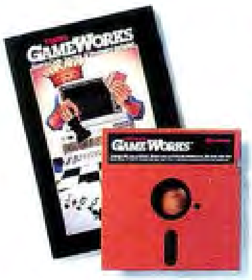
Extra All–IBM Edition

BYTE
THE SMALL SYSTEMS JOURNAL®

Inside the IBM PCs

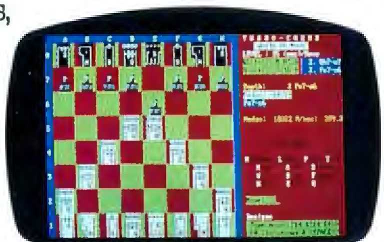
Third Annual Special Issue

EXTRA EDITION BYTE INSIDE THE IBM PCs Volume 10, Number 11



Turbo GameWorks™

Also recently released, Turbo GameWorks is what you think it is: "Games" and "Works." Games you can play right away (like Chess, Bridge and Go-Moku), plus the Works—which is how computer games work. All the secrets and strategies of game theory are there for you to learn. You can play the games "as is" or modify them any which way you want. Source code is included to let you do that, and whether you want to write your own games or simply play the off-the-shelf games, Turbo GameWorks will give hours of diversion, education, and intrigue. George Koltanowski, Dean of American Chess, and former President, United States Chess Federation, reacted to Turbo GameWorks like this, "*With Turbo GameWorks, you're on your way to becoming a master chess player*," and Kit Woolsey, writer, author, and twice Champion of the Blue Ribbon Pairs, wrote, "*Now play the world's most popular card game—Bridge . . . even program your own bidding or scoring conventions.*" Suggested retail: $69.95.



Turbo Graphix Toolbox™

It includes a library of graphics routines for Turbo Pascal programs. Lets even beginning programmers create high-resolution graphics with an IBM, Hercules,™ or compatible graphics adapter. Our Turbo Graphix Toolbox includes all the tools you'll ever need for complex business graphics, easy windowing, and storing screen images to memory. It comes complete with source code, ready to compile. Suggested retail: $69.95.

Recognition for Borland International has come from business, trade, and media, and includes both product awards and awards for technical excellence and marketing. Borland was named "Company of the Year" by PC Magazine; Sidekick, the #1 best seller for the IBM PC, was named "Product of the Year" by InfoWorld; Turbo Pascal was selected one of PC Week's Top 10 Products for 1984; SuperKey won one of PC Magazine's "Best of 1985" awards; Reflex, The Analyst was recognized in the "Software Products of the Year" awards by InfoWorld; and Reflex and SideKick were both nominated for British Micro Awards in 1986.

Turbo Database Toolbox™

A perfect complement to Turbo Pascal, because it contains a complete library of Pascal procedures that allows you to search and sort data and build powerful database applications. Having Turbo Database Toolbox means you don't have to reinvent the wheel each time you write a Turbo Pascal program. It comes with source code for a free sample database—right on disk. The database can be searched by keywords or numbers. Update, add, or delete records as needed. Just compile it and it's ready to go to work for you. Suggested retail: $69.95.

Technical Specifications:

TURBO PASCAL 3.0 Minimum memory:128K; includes 8087 and BCD features for 16-bit MS-DOS and CP/M-86 systems. CP/M-80 version minimum memory: 48K; 8087 and BCD features not available

TURBO DATABASE TOOLBOX Minimum memory: 128K. CP/M-80 minimum memory: 48K. Requires Turbo Pascal 2.0 or later.

TURBO GRAPHIX TOOLBOX* Minimum memory: 192K. Requires PC/MS-DOS 2.0 or later, Turbo Pascal 3.0, and IBM CGA, Hercules Monochrome Card or equivalent.

TURBO TUTOR 2.0 Minimum memory: 192K. CP/M-80 version minimum memory 48K. Requires PC/MS-DOS 2.0 or later and Turbo Pascal 3.0.

TURBO EDITOR TOOLBOX* Minimum memory: 192K. Requires PC/MS-DOS 2.0 or later and Turbo Pascal 3.0.

TURBO GAMEWORKS* Minimum memory: 192K. Requires PC/MS-DOS 2.0 or later and Turbo Pascal 3.0.

TURBO PROLOG* Minimum memory: 384K.

REFLEX: THE ANALYST* Minimum memory: 384K. Requires IBM CGA, Hercules Monochrome Card or equivalent. Works with Intel's AboveBoard-PC and -AT; AST's RAMpage! and RAMpage! AT; Quadram's Liberty-PC and -AT; Tecmar's 640 Plus; IBM's EGA and 3270/PC; AT&T's 6300 and many others.

REFLEX WORKSHOP* Minimum memory: 384K. Requires Reflex: The Analyst. Two disk drives or hard disk recommended.

TURBO LIGHTNING* Minimum memory: 256K. Two disk drives required. Hard disk recommended.

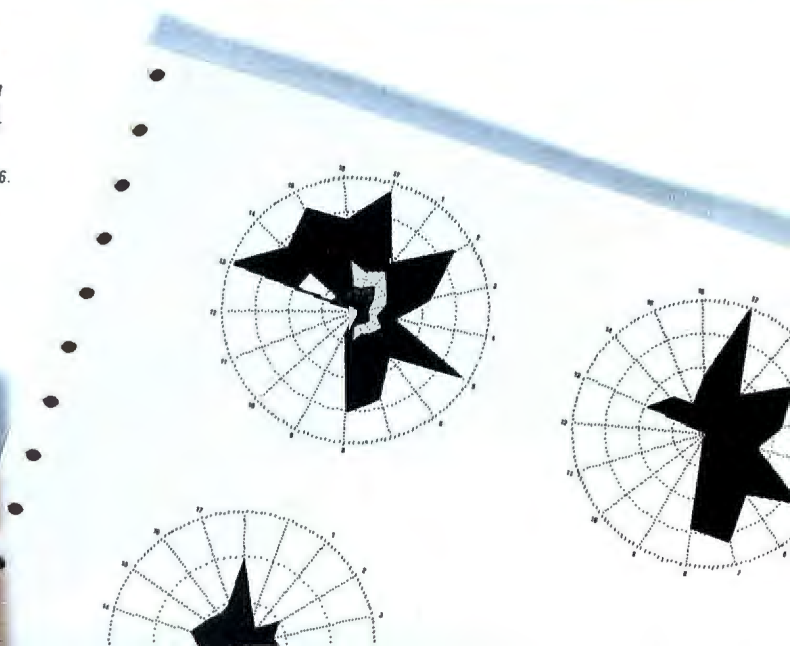
LIGHTNING WORD WIZARD* Minimum memory: 256K. Requires Turbo Lightning. Turbo Pascal 3.0 required to edit source code.

SIDEKICK* Minimum memory: 128K.

TRAVELING SIDEKICK* Minimum memory: 256K.

SUPERKEY* Minimum memory: 128K.

*For IBM PC, AT, XT, PCjr and true compatibles only, running PC/MS-DOS 2.0 or later.



COMPLETELY NEW VERSION!

Turbo Tutor® 2.0

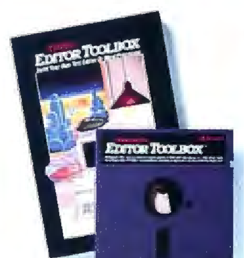
Just released (July '86), the new Turbo Tutor can take you from "What's a computer?" on through to complex data structures, assembly languages, trees, tips on writing long programs in Turbo Pascal, and a high level of expertise. Source code for everything is included. New split screens allow you to put source text in the bottom half of the screen and run the examples in the top half. There are quizzes that ask you, show you, tell you, teach you. You get a 450-page manual—which is not as daunting as it sounds, because unlike many software manuals, it was not written by orangutans. (With our all "almost-free" upgrade, you can upgrade to Turbo Tutor 2.0 by sending us your master diskettes, proof of purchase, and $10.00, which covers shipping and handling.) Suggested retail: $39.95.

NEW SPECIAL!

Turbo Pascal® 3.0

"*For the IBM PC, the benchmark Pascal compiler is undoubtedly Borland International's Turbo Pascal,*" says Gary Ray of *PC Week.* We and more than 500,000 other people around the world think Mr. Ray got that right.

Since launch, Turbo Pascal has become the *de facto* worldwide standard in high-speed Pascal compilers. Described by Jeff Duntemann of *PC Magazine* as the "*Language deal of the century,*" Turbo Pascal is now an even better deal than that—because we've *included* the most popular options (BCD reals and 8087 support). What used to cost $124.95 is now only $99.95! You now get a lot more for a lot less: the compiler, a completely integrated programming environment, and BCD reals and 8087 support—all for a suggested retail of only $99.95.



Turbo Editor Toolbox™

Recently released, we called our new Turbo Editor Toolbox a "construction set to write your own word processor." Peter Feldmann of *PC Magazine* covered it pretty well with, "*A 'write your own word processor' program for intermediate level programmers, with lots of help in the form of prewritten procedures covering everything from word wrap to pull-down windows.*" Source code is included, and we also include MicroStar, a full-blown text editor with pull-down menus and windowing. It interfaces directly with Turbo Lightning to let you spell-check your MicroStar files. Jerry Pournelle of *BYTE* magazine said, "*The new Turbo Editor Toolbox is the Turbo Pascal source code to just about anything you ever wanted a PC-compatible text editor to do.*" Suggested retail: $69.95.

Borland's Business Productivity Programs:

Reflex: The Analyst™ Analytical database manager. Provides complete new look at data normally hidden by programs like 1-2-3® and dBASE.® Best report generator for 1-2-3.

NEW! **Reflex Workshop™** Important new addition to Reflex: The Analyst. Gives you 22 different templates to run your business right.

SideKick® Complete RAM-resident desktop management includes notepad, dialer, calculator and more.

Travelling SideKick™ Electronic version of business/personal diaries, daytime organizers; works with your SideKick files; important professional tool.

SuperKey® Keyboard enhancer. Simple macros turn 1000 keystrokes into 1. Also encrypts your files to keep confidential files confidential.

Borland's Electronic Reference Programs:

Turbo Lightning™ Works with all your programs and checks your spelling *while you type!* Includes 80,000-word Random House® Concise Dictionary and 50,000-word Random House Thesaurus. Forerunner of Turbo Lightning Library.

Lightning Word Wizard™ Includes ingenious crossword solver and six other word challenges. If you're into programming, Lightning Word Wizard is also a development toolbox and the technical reference manual for Turbo Lightning.

We frequently surprise people with inventive, imaginative software, and people frequently surprise us with the way *they* use it.

For example, you'll read on this page how Michael J. Watkins of the Petroleum Technology Center in Houston, Texas, used Turbo Pascal® (and Turbo Graphix Toolbox™ and Turbo Tutor®) to cut down the tedium and time in creating Circular Performance Profile Charts (CPPCs).

We didn't know they existed, but you learn something new every day!

Applications like CPPCs might not fit your exact needs, but at the same time they might stimulate fresh ideas in your mind about how you can put Turbo Pascal and the Turbo Pascal family to work for you.

And thank you for your interest in and support for Borland International.

Philippe Kahn,
President, Borland International

INSIDE STORIES!

■ Turbo Pascal 3.0, already described by *PC Magazine* as "*Language deal of the century*," is now an even better deal than that, because we've included the most popular options (BCD reals and 8087 support). What used to cost $124.95 is now only $99.95!

■ Completely new Turbo Tutor 2.0 now available. New software. New manual. New split screens. New quizzes. Only $39.95. Upgrades available under Borland's "Almost-Free" upgrade plan. Details inside.

LATE NEWS!

■ June/July Special Artificial Intelligence Issue of The *Micro Technical Journal* says, "*Turbo Prolog looks like it's going to be a winner, for both the beginner and professional programmer.*"

Turbo Pascal deliberately programmed to go around in circles

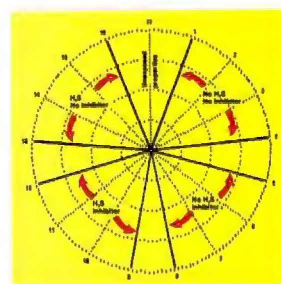
Circular charts (or CPPCs) are used by Michael J. Watkins of the Petroleum Technology Center in Houston, Texas, to plot a single performance property for a large number of elastomers, which have elastic, rubber-like properties.

Mr. Watkins wrote us saying, "*Because CPPCs condense a lot of data in one graphic, they can be very tedious and time-consuming to draw.*"

What he did to solve those problems was to write a Turbo Pascal program for IBM® personal computers to "*generate these charts quickly and easily.*"

He used Turbo Pascal "*because it has a companion set of very powerful graphics programs (Turbo Graphix Toolbox) which greatly simplifies the required programming.*

Turbo Pascal is not a difficult language to use and can be easily learned by persons who can program in FORTRAN or BASIC. An excellent tutorial (Turbo Tutor) is available for the novice or experienced programmer. The Turbo Pascal products are also very moderately priced."



"The computer is no better than its program."
Elting Morison, author of "Men, Machines and Modern Times"

Step-by-step tutorial, demo programs with source code included!

Borland's new Turbo Prolog is the powerful, completely natural introduction to Artificial Intelligence

Prolog is probably one of the most powerful computer programming languages ever conceived, which is why we've made it our *second* language—and "turbo-charged" it to create Turbo Prolog.

Our new Turbo Prolog, the natural language of Artificial Intelligence, brings supercomputer power to your IBM® PC and introduces you step-by-step to the fascinating new world of Artificial Intelligence. And does all this for an astounding $99.95.

Turbo Prolog is to Prolog what Turbo Pascal® is to Pascal!

Our Turbo Pascal astonished everyone who thought of Pascal as "just another language." We changed all that—and now Turbo Pascal is the de facto worldwide standard, with hundreds of thousands of enthusiasts and users in universities, research centers, schools, and with professional programmers, students, and hobbyists.

You can expect at least the same impact from Turbo Prolog, because while Turbo Prolog is the most revolutionary and natural programming language, it is also a complete development environment—just like Turbo Pascal.

Minimum memory: 384K

> "Turbo Prolog offers generally the fastest and most approachable implementation of Prolog."
> Darryl Rubin, AI Expert

Even if you've never programmed before, our free tutorial will get you started right away

You'll get started right away because we have included a complete step-by-step tutorial as part of the 200-page Turbo Prolog Reference Manual. Our tutorial will take you by the hand and teach you everything you're likely to need to know about Turbo Prolog and artificial intelligence.

For example: once you've completed the tutorial, you'll be able to design your own expert systems utilizing Turbo Prolog's powerful problem-solving capabilities.

Think of Turbo Prolog as a high-speed electronic detective. First you feed it information and teach it rules. Then Turbo Prolog "thinks" the problem through and comes up with all the reasonable answers—almost instantly.

If you think that this is amazing, you just need to remember that Turbo Prolog is a 5th-generation language—and the kind of language that 21st century computers will use routinely. In fact, you can compare Turbo Prolog to Turbo Pascal the way you could compare Turbo Pascal to machine language.

You get the complete Turbo Prolog programming system for only $99.95

You get a complete Turbo Prolog development system including:

- The lightning-fast Turbo Prolog incremental compiler and the interactive Turbo Prolog editor.
- The 200-page reference manual which includes the step-by-step Turbo Prolog tutorial.
- The free GeoBase™ natural query language database including commented source code on disk—ready to compile. GeoBase is a complete database designed and developed around U.S. geography. It includes cities, mountains, rivers, and highways, and comes complete with natural query language. Use GeoBase immediately "as is," or modify it to fit your own interests.

So don't delay—don't waste a second—get Turbo Prolog now. $99.95 is an amazingly small price to pay to become an immediate authority—an instant expert on artificial intelligence! The 21st century is only one phone call away.

BORLAND INTERNATIONAL

4585 SCOTTS VALLEY DRIVE
SCOTTS VALLEY, CA 95066
(408) 438-8400 TELEX: 172373

Vive la différence

YES! I want the best!

For credit card orders or the dealer nearest you

call (800) 255-8008

in CA call (800) 742-1133

Copies	Product	Price	Totals
__	Turbo Prolog	$99.95	$______
__	Reflex: The Analyst	149.95*	$______
__	Reflex Workshop	69.95*	$______
__	Reflex, Reflex Workshop	199.95*	$______
__	Turbo Pascal 3.0 w/8087 & BCD	99.95	$______
__	Turbo Pascal for CP/M-80	69.95	$______
__	Turbo Database Toolbox	69.95	$______
__	Turbo Graphix Toolbox	69.95	$______
__	Turbo Tutor 2.0	39.95	$______
__	Turbo Editor Toolbox	69.95	$______
__	Turbo GameWorks	69.95	$______
__	Turbo Lightning	99.95	$______
__	Lightning Word Wizard	69.95	$______
__	Turbo Lightning, Lightning Word Wizard	149.95	$______
__	SideKick	84.95	$______
__	Traveling SideKick	69.95*	$______
__	SideKick, Traveling SideKick	125.00*	$______
__	SuperKey	69.95	$______
	Outside USA add $10 per copy CA and MA res. add sales tax		$______
	Amount enclosed		$______

Prices include shipping to all US cities.

Carefully describe your computer system:

Mine is: __ 8-bit __ 16-bit

I use: __ PC-DOS __ CP/M-80 __ MS-DOS __ CP/M-86

My computer's name and model is: ______

The disk size I use is: ☐ 3½" ☐ 5¼" ☐ 8"

Payment: VISA MC Money order Check

Credit card expiration date ___/___

Card # ______

NOT COPY PROTECTED BG15

****60-DAY MONEY-BACK GUARANTEE**

Name: ______

Shipping Address: ______

City: ______

State: ______ Zip: ______

Telephone: ______

CODs and purchase orders WILL NOT be accepted by Borland. Outside USA make payment by bank draft, payable in US dollars drawn on a US bank.

*Limited Time Offer until September 1, 1986.

**YES, if within 60 days of purchase you find that this product does not perform in accordance with our claims, call our customer service department and we will gladly arrange a refund.

Prices subject to change without notice.

Turbo Prolog 1.0 Technical Specifications

Compiler: Incremental compiler generating native in-line code and linkable object modules. The linking format includes a linker and is compatible with the PC-DOS linker. Large memory model support. Compiles over 2500 lines per minute on a standard IBM PC.

Interactive Editor: The system includes a powerful interactive text editor. If the compiler detects an error, the editor automatically positions the cursor appropriately in the source code. At run-time, Turbo Prolog programs can call the editor, and view the running program's source code.

Type System: A flexible object-oriented type system is supported.

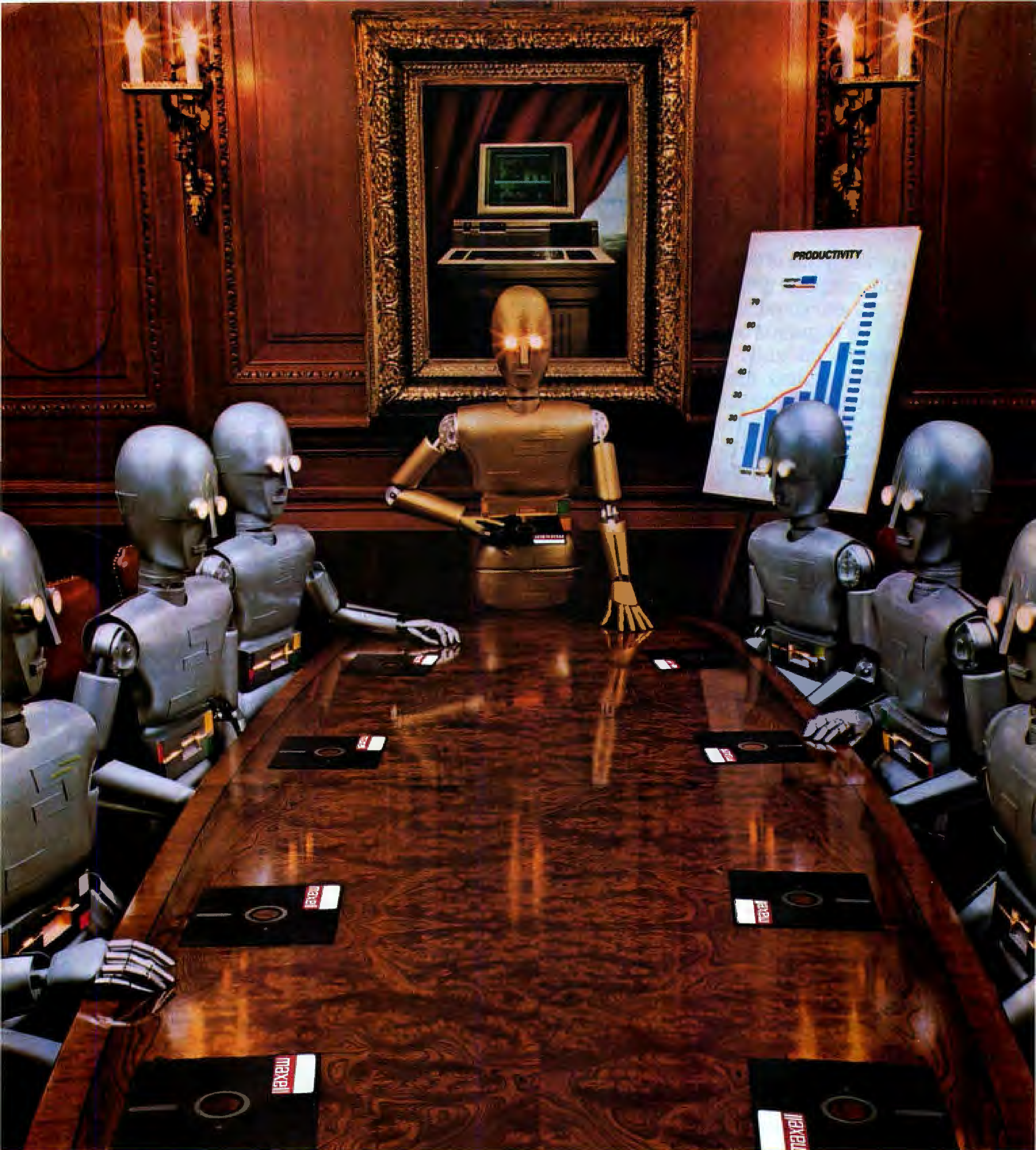
Windowing Support: The system supports both graphic and text windows.

Input/Output: Full I/O facilities, including formatted I/O, streams, and random access files.

Numeric Ranges: Integers: −32767 to 32T67; Reals: 1E−307 to 1E+308.

Debugging: Complete built-in trace debugging capabilities allowing single stepping of programs.

Inquiry 42 for End-Users. Inquiry 43 for DEALERS ONLY.


PRODUCTIVITY
maxell

COVER ILLUSTRATED BY ROBERT TINNEY

VOLUME 11, NUMBER 11, 1986

BYTE

Inside the IBM PCs

LISTINGS

ABC

BYTE (ISSN 0360-5280) is published monthly with one extra issue per year by McGraw-Hill Inc. Founder James H. McGraw (1860-1948). Executive, editorial, circulation, and advertising offices: One Phoenix Mill Lane, Peterborough, NH 03458, phone (603) 924-9281. Office hours: Mon–Thur 8:30 AM – 4:30 PM, Friday 8:30 AM – 1:00 PM, Eastern Time. Address subscriptions to BYTE Subscriptions, P.O. Box 590, Martinsville, NJ 08836. Postmaster: send address changes, USPS Form 3579, undeliverable copies, and fulfillment questions to BYTE Subscriptions, P.O. Box 596, Martinsville, NJ 08836. Second-class postage paid at Peterborough, NH 03458 and additional mailing offices. Postage paid at Winnipeg, Manitoba. Registration number 9321. Subscriptions are $21 for one year, $38 for two years, and $55 for three years in the USA and its possessions. In Canada and Mexico, $23 for one year, $42 for two years, $61 for three years. $69 for one year air delivery to Europe. 31,000 yen for one year air delivery to Japan, 15,600 yen for one year surface delivery to Japan. $37 surface delivery elsewhere. Air delivery to selected areas at additional rates upon request. Single copy price is $3.50 in the US and its possessions, $4.25 in Canada and Mexico, $4.50 in Europe, and $5 elsewhere. Foreign subscriptions and sales should be remitted in U.S. funds drawn on a U.S. bank. Please allow six to eight weeks for delivery of first issue. Printed in the United States of America.

Address all editorial correspondence to the Editor, BYTE, One Phoenix Mill Lane, Peterborough, NH 03458. Unacceptable manuscripts will be returned if accompanied by sufficient first-class postage. Not responsible for lost manuscripts or photos. Opinions expressed by the authors are not necessarily those of BYTE.

Subscription questions or problems should be addressed to: BYTE Subscriber Service, P.O. Box 328, Hancock, NH 03449

Officers of McGraw-Hill Information Systems Company: President: Richard B. Miller. Executive Vice Presidents: Frederick P. Jannott, Construction Information Group; Russell C. White, Computers and Communications Information Group; J. Thomas Ryan, Marketing and International. Senior Vice Presidents: Francis A. Shinal, Controller; Robert C. Violette, Manufacturing and Technology. Senior Vice Presidents and Publishers: Laurence Altman, Electronics Week; Harry L. Brown, BYTE; David J. McGrath, Construction Publications. Group Vice President: Peter B. McCuen, Communications Information. Vice President: Fred O. Jensen, Planning and Development.

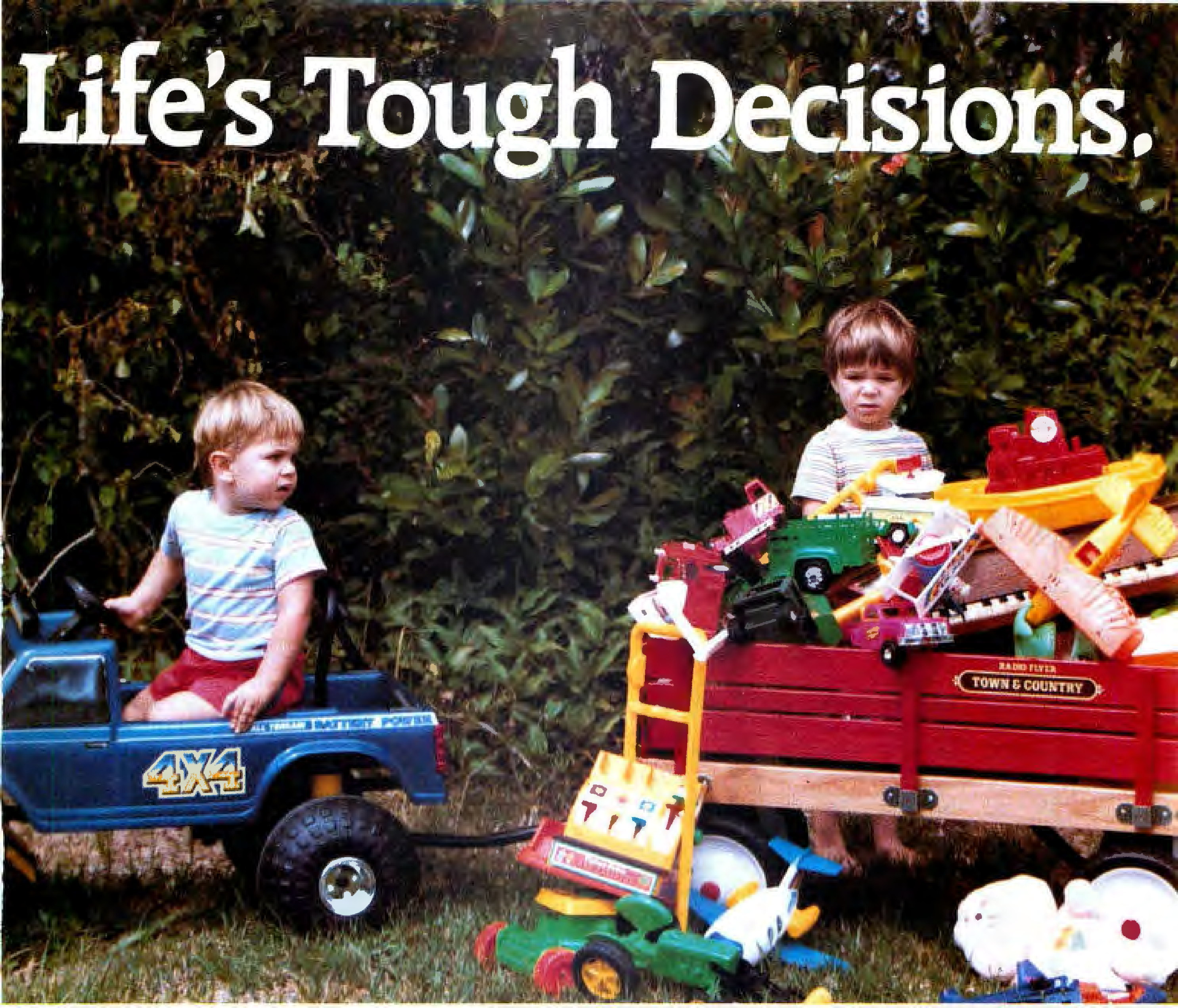
Officers of McGraw-Hill, Inc.: Harold W. McGraw, Jr., Chairman; Joseph L. Dionne, President and Chief Executive Officer; Robert N. Landes, Executive Vice President and Secretary; Walter D. Serwatka, Executive Vice President and Chief Financial Officer; Shel F. Asen, Senior Vice President, Manufacturing; Robert J. Bahash, Senior Vice President, Finance and Manufacturing; Ralph R. Schulz, Senior Vice President, Editorial; George R. Elsinger, Vice President, Circulation; Ralph J. Webb, Vice President and Treasurer.

EDITORIAL AND BUSINESS OFFICE: One Phoenix Mill Lane, Peterborough, New Hampshire 03458, (603) 924-9281.

West Coast Offices: 425 Battery St., San Francisco, CA 94111, (415) 954-9718; 3001 Red Hill Ave., Building #1, Suite 222, Costa Mesa, CA 92626, (714) 557-6292. **New York Editorial Office:** 1221 Avenue of the Americas, New York, NY 10020, (212) 512-2000.

BYTEnet: (617) 861-9764 (set modem at 8-1-N or 7-1-E; 300 or 1200 baud).

Life's Tough Decisions.



THE CHOICES.

Remember when your biggest problem was no storage space in your wagon? No room for your boats, trains, cars, and planes. Your choice was to either get rid of some toys or get a bigger wagon.

Now your *wagon* has turned into an IBM PC-XT. Your boats, trains, and cars are now database, spreadsheet, and word processing files. But your choice for more storage is still about the same: either get rid of some "toys" or get a bigger "wagon."

GETTING A BIGGER WAGON.

Since every bit counts, you can now upgrade your computer with a CORE 43 MB half height drive and still give yourself room to grow. IBM's standard XT contains only a 10 or 20 MB full height drive, which filled up faster than you thought it would. Your PC-XT may even accomodate 2 CORE drives. That's 86 MB and a floppy all inside the box.

IBM PC-XT and IBM PC-AT are registered trademarks of International Business Machines, Inc.

A SIMPLE TRANSITION.

Installation is easy. Everything is supplied for you to install in less than 30 minutes. Have a drive already? Simply remove the old and slide in the new.

SHIFT INTO HIGH GEAR.

Our half height drives are the fastest in the world. With an average access time of less than 26 ms, these drives are 4-5 times faster than the standard XT (and more than 30% faster than an IBM PC-AT). Load, save, sort, and retrieve data in record time. Plus, they're one half the size! Ideal for networks and file servers, and they support both DOS and XENIX.

CORE
INTERNATIONAL

7171 N. Federal Hwy. Boca Raton, FL 33431
(305) 997-6055
Telex: 315 809 CORE INTL DEBE

Vectra is a registered trademark of Hewlett Packard.

Inquiry 76

NEVER GET STRANDED.

Our read/write heads unconditionally and automatically go to a safe landing zone and lock there on power off. Unlike the factory issue XT drive (and most others), our drives protect you from head crashes and lost data. They'll never leave you crying.

They're also fully shock mounted for your XT or even a portable computer.

THE ROAD TO SUCCESS.

CORE International is the world's largest supplier of high performance IBM PC-AT compatible drives. This high performance is now available in a half high form factor for other compatibles including HP Vectra, Zenith, AT&T, and the entire Compaq line. All drives are backed by a one-year warranty in addition to a choice of on-site, local, or exchange maintenance services supported by CORE and other major maintenance companies.

Contact us directly or see your local CORE Authorized Dealer for details.

XENIX is a trademark of Microsoft, Inc.

EDITORIAL

The Struggle for Compatibility

The IBM Personal Computer, introduced in the fall of 1981, celebrates its fifth birthday this year. With the flattery quotient nearing 1.00, the folks in Boca Raton should be proud of their progeny as the homes and offices of customers around the world slowly fill with imitation PCs. 1986 is proving to be the Year of the Clone.

What makes building clones so easy? Why can anybody who feels like it put together a clone, rent a mailbox, and rake in a pile of cash?

The key to clone building is the use of standard parts and the availability of crucial system software, like Phoenix Technology's IBM PC–compatible BIOS and Microsoft's MS-DOS. While there are subtle incompatibilities from clone to clone, there are usually enough similarities from machine to machine to make most clones perform acceptably as copies of the PC. This is particularly true for application software written to make DOS and BIOS calls instead of directly addressing a machine's hardware.

Compatibility with the IBM PC has been a contraposition for IBM. Wide availability of machines that can all run the same software has prompted the creation of thousands of programs. The availability of this software has assured the sale of machines that will run all these great programs. IBM's problem has been that other companies can provide this now-standard computer more cheaply than IBM.

So the IBM PC created an archetype around which a flourishing industry has sprouted. Could this industry-building scenario work in reverse?

Toward a 32-Bit Bus Standard

This past spring, a group of hardware and software makers got together to talk about creating a bus standard for extending the IBM PC AT's 16-bit bus to 32 bits. The extra 16 bits would accommodate the Intel 80386 microprocessor as the engine for new computers and peripheral devices designed to work with the new 32-bit bus. The name chosen for this new bus is Personal Computer Extended Technology (PCET).

The PCET bus makes possible the construction of personal computers and peripheral devices based on the Intel 80386 processor in the absence of a standard 80386 machine from IBM. Many people speculate that IBM will either be late with an 80386-based machine or build an 80386 machine that uses a proprietary operating system (or maybe both). In any case, the PCET bus circumvents IBM by establishing a standard bus around which other manufacturers can build 80386 machines. A half dozen 80386 machines using the PCET bus would establish a standard that might survive, regardless of actions taken by IBM.

Since the PCET bus is an extension of the PC AT bus, machines built around this bus could use existing graphics and memory-expansion cards and existing versions of MS-DOS—and thousands of existing application programs. The PCET bus provides a painless way to build "turbo" ATs—machines that function like an AT but at two to three times the speed.

Once an existing base of turbo ATs develops, software houses can create new operating systems—or adapt existing ones like UNIX—to tap the potential of the 80386. Hardware companies can build graphics and memory boards for a new class of workstations that are based on the 80386.

Thus, a whole new genre of clones can be built, using standard, off-the-shelf parts along with adapted BIOS software and existing operating system and application programs. These new clones will actually predate the machine they are designed to replace! It seems that the personal computer world may take on an odd, ironic tilt.

PCET-bus-based machines may survive as a group only if IBM's 80386 machine is *not* an open-architecture machine. Tandy is gambling that IBM will make an 80386 machine that can be cloned. If this turns out to be the case, any company selling a PCET-bus computer should get in touch with an auctioneer.

The obvious question arises about whether or not people really need turbo ATs. Only 13 percent of the owners of IBM computers require the power of an AT. Maybe only 13 percent of AT owners will benefit from the power of a turbo AT. Even if this were the case, that would still create a niche for approximately 40,000 80386-based computers. But another problem is the issue of affordability. Is using an 80386 in a single-user machine a waste of hardware resources? Is it cost-effective?

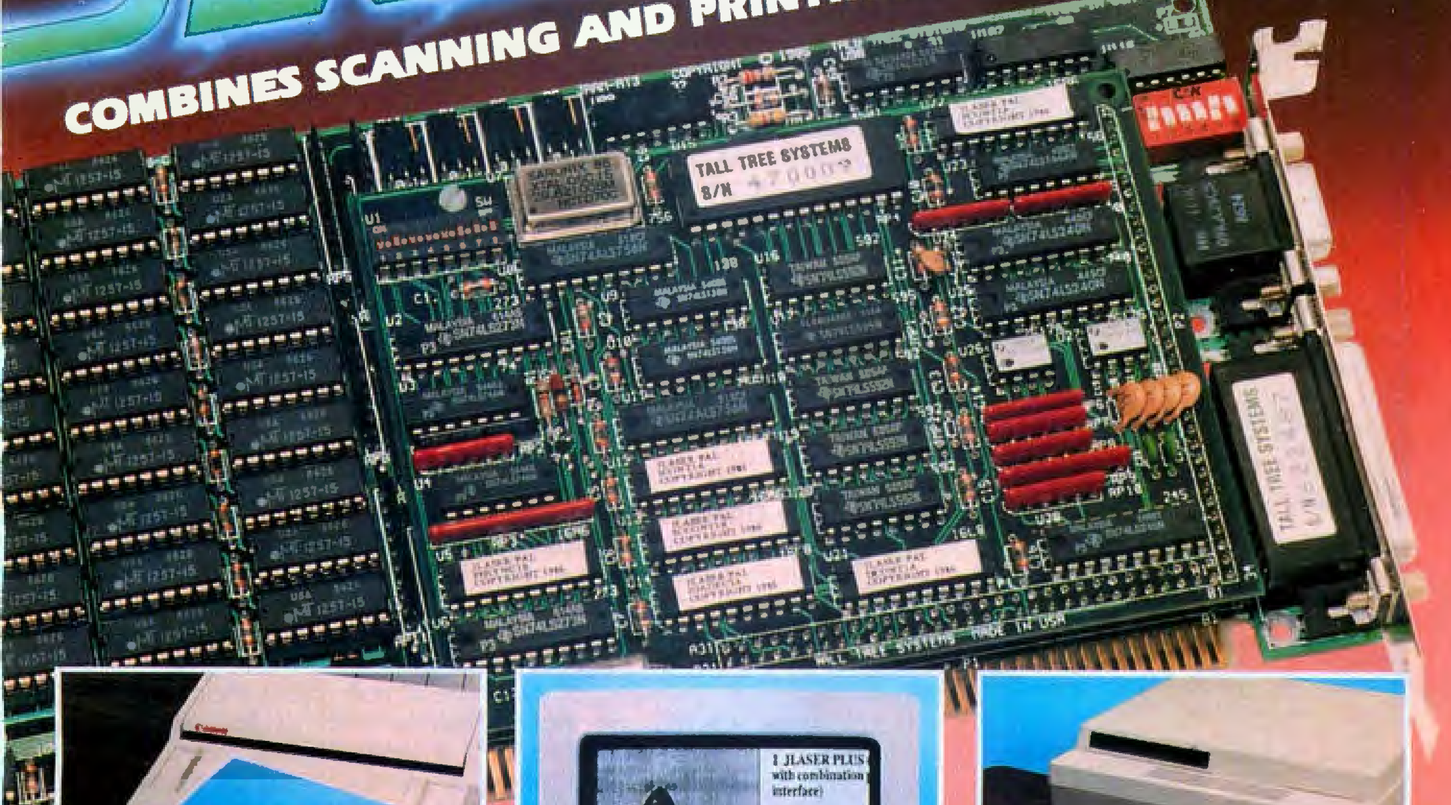
Modifications are already available for Apple's Macintosh that permit the use of an MC68020 processor and MC68881 math coprocessor and the addition of up to 4 megabytes of RAM. These modifications, when coupled with a 1K- by 1K-pixel monitor, transform a Mac into a high-performance workstation. But the cost is high—$7000 to $9000.

Building similarly equipped turbo ATs may turn out to be just as expensive. At that price, buyers who need a full-fledged workstation (in contrast to the folks who want to run Lotus 1-2-3 faster) might look at Sun, Apollo,

(continued)

JLASER PLUS

COMBINES SCANNING AND PRINTING IN A SINGLE BOARD!



It makes desktop publishing a piece of cake!

Tall Tree Systems introduces another breakthrough in desktop publishing with JLASER PLUS. We've combined a 2 MB EMS memory board and an interface to both a Canon®-based laser printer and scanner. JLASER PLUS increases the performance of both devices and gives you a low-cost solution to the limitations you've been experiencing with them.

Furthermore, the same memory that is made available to your printer and scanner is also available for all your other conventional applications. You get system memory, expanded LIM memory, extended memory in an AT-type machine, RAM Disk and print spooler — all in a single slot!

Supporting JLASER PLUS is a host of software packages, such as PC Paintbrush + from ZSoft, Dr. Halo D.P.E. from Media Cybernetics, LaserGL from Software Express, Ventura Publisher from Xerox, Page Builder from White Sciences, Le Print from Le Baugh Software, Fancy Font and Fancy Word from SoftCraft, Inc., and many more to be announced.

It takes a technological innovator like Tall Tree Systems to provide a major advancement like JLASER PLUS. And we don't stop at performance. We also deliver value, which is truly icing on the cake.

TALL TREE SYSTEMS
1120 San Antonio Road
Palo Alto, CA 94303
(415) 964-1980

TALL TREE SYSTEMS

1-2-3 faster) might look to Sun, Apollo, DEC, or IBM's RT PC for more stable hardware and software.

The one application area that can potentially benefit most from 80386 performance is graphics. Intel's new microprocessor can permit the creation of workstations with a great deal of graphics power. But the trend in graphics processing is to offload graphics to its own processor, either custom-built, as in the Amiga, or off-the-shelf parts, like the Texas Instruments TMS34010. There is also the problem of the 16-bit EGA standard and how it can be adapted to a 32-bit environment.

Technology Driven

Regardless of the *need* for 80386-based machines, technology has a way of evolving for its own sake. People need to find out what is possible, whether it be with airplanes or computers.

Therefore, the PCET-bus specification effort makes sense from a research and development point of view. It establishes a common starting ground for machine designs using a new processor.

The PCET-bus design goals seek to provide a high-speed bus for 32-bit memory and memory-mapped data transactions. The bus also needs to be flexible enough to permit 32-bit coprocessor cards to become alternate bus masters.

The PCET bus pipelines address and control information, making this information for a subsequent bus cycle available in the latter stages of the current bus cycle. The bus design calls for a default to two wait states but can be operated with one or no wait states by activating additional control signals.

Compatibility with the existing 16-bit PC AT bus is a must for the PCET bus to take advantage of existing graphics and memory standards. (An interesting outgrowth of the PCET-bus specification effort is that the committee first had to codify the AT bus.) A mix of 16- and 32-bit bus masters can drive the PCET bus to accommodate refresh, direct memory access, and other tasks.

The memory map for the PCET bus shows the entire 4-gigabyte address space of the 80386 as being available to the bus, except the 640K-byte to 1-megabyte area reserved for ROM and video devices.

Mechanically, the PCET bus extends the data lines of the PC AT bus by 40 pins. The data lines on the RT PC bus, already 40 pins instead of 36 as in the AT, also get another 40 pins.

We still need to ask some legitimate questions about the PCET bus. Is it possible to achieve nearly the same 80386 performance on a 16-bit bus feeding a 32-bit cache? Since 85 percent of all CPU operations are read operations, caching operands may allow an 80386 to operate almost as fast with a 16-bit bus as it would with a 32-bit bus for some applications.

Another question about a 32-bit standard bus: What if companies ignore it and create their own bus? There have been widespread reports of 80386-machine projects at Compaq and Zenith that will feature proprietary 32-bit bus designs. The PCET bus, no matter how elegant its design, will fade away like the dodo if everyone ignores it.

Another Round

Regardless of the creation of the PCET bus, there will be new machines built around the 80386. The early machines will be expensive and will have no software to fully exploit their power. They may serve as a bridge from the current PC world to a future 80386-machine-based environment. These new machines will undoubtedly raise a different set of compatibility problems with the machines and software that preceded them.

But we'll buy them anyway.

—G. *Michael Vose*
Senior Technical Editor

Look for IBM-related conferences and special events on BIX starting in late October. Current plans call for a special event on "System Software for the 80386," plus conferences on the RT PC and the PC Convertible.

The Most Powerful LAN Fits on a Disk.

Network Power. You knew that someday there would be a powerful LAN that didn't need old-technology network boards. It would be fast, easy to install, and run 99% of PC-DOS software. It would be expandable, provide remote access, password-protection, and enable you to use inexpensive terminals as workstations in a PC-DOS environment.

Dream no more, because the power is here.

Its name is LANLink™.

A Software-Driven LAN Powerful Enough To Use RS-232 Ports for Network Communications. In development for over three years, LANLink™ represents the next generation of local area networks. All of the logic which has traditionally resided on network boards is on LANLink's Satellite and Server Diskettes.

No additional hardware is required. Inexpensive serial ports replace "Kilobuck" Network Interface Boards making installation costs one-third that of a board-driven network.

How To Configure a Smart Network...With Dumb Terminals, But Without Dedicated Servers. Boasting a wide variety of configurations, LANLink™ is most often set up as a "Star" having up to eight satellites connected to a central, nondedicated server. Larger networks can have multiple servers, supporting a total of 73 or more network users.

R-LAN™ (Remote-LAN) gives users the ability to interact with a LANLink™ network in real time via modem. Plus, if MultiLink Advanced™ is run on a Satellite, inexpensive dumb terminals can be used to access network disks, files, and programs.

99% of PC-DOS Applications Run in a Totally-Transparent Network Environment. If you know DOS, you already know how to use LANLink.™ COPY transfers files among users, and a 2-drive PC Satellite boots 1-2-3 from the Server's hard disk with the entry c:lotus. Each satellite's access can be limited to specific disks, printers, and subdirectories. A wide variety of software including Lotus 1-2-3, dBASE III, and WordStar 2000 is fully compatible. LANLink™ has a collision-free data transfer rate which exceeds 115,000 BPS.

Power Up Your PCs Today. For complete details and the authorized dealer nearest you, call The Software Link TODAY. The LANLink™ Starter Kit is $495 and includes modules for both a Server and a Satellite. For a limited time, 50 feet of shielded RS-232 cable will be included free of charge. Additional Satellite Modules are only $195, each.

LANLink™ is immediately available and comes with a money-back guarantee. VISA, MC, AMEX accepted.

LANLink™

THE SOFTWARE LINK, INC.

Developers of LANLink™ and MultiLink Advanced™

8601 Dunwoody Place, Suite 632, Atlanta, GA 30338 Telex 4996147 SWLINK

CALL: 404/998-0700

Dealer Inquiries Invited

THE SOFTWARE LINK, INC./CANADA 250 Cochrane Drive, Suite 12
Markham, Ontario L3R 6B7 **CALL: 416/477-5480**

Inquiry 263

USE THE BRAINS YOUR IBM WASN'T BORN WITH.

Right at your fingertips in CompuServe's IBM® Forums.

In the **IBM New Users Forum** you'll swap ideas with other new PC users, learn to use Forum features, and pose even basic questions to PC experts.

Our **IBM Junior Forum** gives PCjr® users a reliable source for tips on software, hardware, telecommunications, games and other interests.

In the **IBM Software Forum** you'll trade tips with other IBM PC and AT users on utility software, word processing, DOS and other operating systems.

Visit the **IBM Communications Forum** for advice on the features and compatibility of communications software and hardware, PC Bulletin Boards, micro-mainframe interfaces and more.

The **IBM Hardware Forum** addresses hardware topics of all types, plus product updates and announcements.

Easy access to free software.

- *Download first-rate, non-commercial, user-supported software and utility programs.*
- *Take advantage of CompuServe's inexpensive weeknight and weekend rates (when Forums are most active, and standard online charges are just 10¢ a minute).*
- *Go online in most major metropolitan areas with a local phone call.*
- *And receive a* ***$25.00 Introductory Usage Credit*** *with purchase of your CompuServe Subscription Kit.*

Information you simply can't find anywhere else.

Use the Forum *Message Board* to send and receive electronic messages, and pose specific questions to other IBM and compatible owners.

Join ongoing, real-time discussions in a Forum *Conference.*

Search our unparalleled Forum *Data Libraries* for free software, user tips, transcripts of online conferences and more.

Enjoy other useful services like:

- ***Popular Computer Magazines****—electronic editions, for your reading pleasure. Including Dr. Dobb's Journal and Computer Language.*
- ***Other CompuServe Forums****—supporting LOTUS® products like Symphony™ and 1-2-3.™ Borland International,® Ashton-Tate,® Digital Research,® MicroPro,® Microsoft® and other software. Also Pascal, Basic, C, Forth, Assembly and other programming languages.*

All you need is your IBM or IBM-compatible computer and a modem ...or almost any other computer.

To buy your Subscription Kit, see your nearest computer dealer. Suggested retail price is $39.95. To receive our free brochure, or to order direct, call 800-848-8199 (in Ohio, call 614-457-0802). If you're already a CompuServe subscriber, type GO IBMNET (the IBM Users Network) at any ! prompt to see what you've been missing.

CompuServe®

Information Services, P.O. Box 20212
5000 Arlington Centre Blvd., Columbus, Ohio 43220

800-848-8199

In Ohio, Call 614-457-0802
An H&R Block Company

Inquiry 66

IBM's PC XT *Model* 286 *runs on a* 6-MHz 80286.

IBM Introduces 80286-based XT

IBM's latest version of its PC XT, the Model 286, is based on an 80286 processor running at 6 MHz with no wait states and features more memory than earlier models of the XT and up to three internal drives.

Priced at $3995, the Model 286 comes standard with 640K bytes of memory that can be expanded to 12.6 megabytes, a half-height 1.2-megabyte 5¼-inch floppy disk drive, a 20-megabyte hard disk drive, and a serial/parallel adapter. The machine has eight expansion slots, five of which can accommodate 16-bit boards. Internal storage options include a 3½-inch 720K-byte floppy disk drive ($190), a second 5¼-inch floppy drive with a capacity of 1.2 megabytes ($275), and a 5¼-inch floppy drive that can handle 360K bytes ($225). The Model 286 also supports IBM's external 3½-inch floppy disk drive.

The computer uses a number of custom gate arrays to reduce chip count and board space requirements. Some of the chips are SIP (single in-line package), arranged vertically on the motherboard, thus further reducing the space requirements of the motherboard. The Model 286 has the same footprint as the IBM PC and XT.

According to the company, the machine's 80286 operates in real address mode for compatibility with the 8088 and in protected virtual mode, which supports multitasking and enables the processor to address up to 16 megabytes of real memory and 1 gigabyte of virtual memory. The computer requires DOS version 3.2 or XENIX version 2.0.

The older 6-MHz PC AT will be phased out, the company says, but IBM will continue to sell the 8-MHz version of the AT. Contact IBM Corp., Information Systems Group, 900 King St., Rye Brook, NY 10573, (914) 934-4488.

Inquiry 550.

AT&T's 32-bit Digital Signal Processor

AT&T has developed the WE DSP32, a 32-bit single-chip digital signal processor with floating-point capabilities. Applications for the chip include communications functions such as filtering, adaptive equalization, and echo canceling. The device can also be used for speech and image processing, as well as graphics simulation.

The chip has a clock rate of 16 MHz and executes its instruction cycle in 250 nanoseconds, or 8 million floating-point operations per second. On-chip memory consists of 2K bytes of ROM and 4K of RAM.

Software and hardware development tools that operate under MS-DOS and UNIX System V are available for use with the chip. A Support Software Library includes a high-level "C-like" syntax assembler and a simulator. A single-board development system permits real-time testing, debugging, and evaluation of hardware and software.

Sample OEM quantities of the WE DSP32 chip cost $175. Contact AT&T Technology Systems, 555 Union Blvd., Dept. 50AL203140, Allentown, PA 18103, (800) 372-2447.

Inquiry 551.

Object-oriented Programming Language

Actor is an object-oriented language that makes use of the Microsoft Windows interface. Because it's interactive, you can experiment at the keyboard with your programming ideas and watch the results.

The environment includes a workspace for communicating with the interpreter, a browser, an error manager, a window-based debugger, and a save facility. You can send messages to any object in the system to check status, change parameters, or learn about inherited behavior. A concurrent garbage collector is designed to free you from memory management duties. Actor contains classes that support list processing, pattern matching, parsing, and predicate logic.

The package sells for $495 and comes with source code (except for the assembly language kernel), tutorial, reference manual, and demo program. Contact The Whitewater Group Inc., 906 University Place, Evanston, IL 60201, (312) 491-2370.

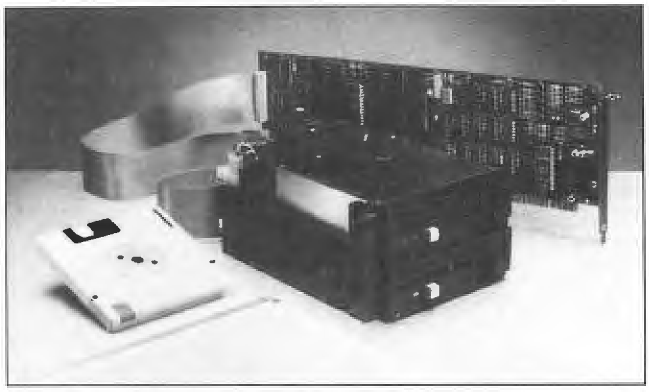
Inquiry 552.

(continued)

Two Removable-Cartridge Systems

Sysgen introduced the DuraPak, an internal storage subsystem with removable hard disk cartridges for IBM PCs, XTs, ATs, and compatibles. A single-drive, 15-megabyte DuraPak system sells for $1295, which includes a controller and installation hardware. A dual-drive, 30-megabyte DuraPak sells for $2095.

Each removable cartridge has a 15-megabyte capacity and measures 4¼ inches by 4⅜ inches. For more information, contact Sysgen Inc., 47853 Warm Springs Blvd., Fremont, CA 94539, (415) 490-6770.
Inquiry 553.



Sysgen's DuraPak uses 15-megabyte removable cartridges.

Data Technology's subsystem features 5¼-inch removable cartridges that can store 10 megabytes of data. The drive's average access time is 75 milliseconds, and its data transfer rate is 275K bytes per second. The cartridges consist of a flexible disk encased in a hard disk cartridge.

A single half-height subsystem retails for $1195; a dual configuration sells for $1995. The disks cost about $39 each. For more information, contact Data Technology Corp., 2775 Northwestern Parkway, Santa Clara, CA 95051, (408) 496-0434.
Inquiry 554.

System for Visual Artists

Time Arts, producers of the Lumena software for visual arts, has released another graphics package. EVA (Environment for Visual Art), based on a 512K-byte PC AT, combines raster and vector display technology and uses 32,000 on-screen colors. You can use the program to digitize images and then manipulate them with the AT. Time Arts says its package can also duplicate many of the video and film effects produced by more expensive systems.

With the digitizing tablet and pen, you have a selection of electronic brushes, colors, typefaces, and such capabilities as masking, automatic edge smoothing, texture rendering, and instant full-color tinting.

EVA uses two monitors: color for working with images, monochrome for keeping track of tools and system status. Images you develop on the 512 by 512 RGB screen can be increased to 2048 by 4096 resolution for direct transfer to a digital film recorder. With appropriate output devices, the system can also make its own color separations, 35mm slides, transparencies, video frames, tape sequences, and ink-jet, dot-matrix, and laser prints. Images can also be sent directly to Scitex graphics systems.

Pricing depends on output requirements, but the company says a complete system can cost $15,000. For more information, contact Time Arts Inc., P.O. Box 6476, 3436 Mendocino Ave., Santa Rosa, CA 95406-0476, (707) 576-7722.
Inquiry 555.

3-D Engineering and Animation on PC AT

Renaissance Graphics unveiled real-time three-dimensional solid-modeling and animation add-on boards for the IBM PC AT. The Renaissance Graphics System (RGS) boards allow for fully animated 3-D shaded objects at a resolution of 512 by 512 pixels, 256 simultaneous colors from a palette of 16.8 million, and flicker-free 60-Hz display on the PC AT.

The system includes three 68000 microprocessors operating at 12.5 MHz, a 68881 math coprocessor, and two video frame buffers. The boards also have a TI 34061 video controller and a 192K-byte ROM with firmware routines. Among the system's capabilities are hidden line and surface removal, flat or smooth shading, selectable light source, and anti-aliasing. Its z-axis clipping, infinite zoom, and roam enable the viewer to travel around in 3-D space, going behind or inside moving objects.

The RGS is designed for use with a high-resolution analog RGB monitor and will be bundled with a 3-D paint system. Standard NTSC, PAL, and SECAM video signals that can be genlocked (synchronized) to an external video source, along with film and slide recording, are also planned and will require additional hardware.

PC AT–based workstations using RGS boards will cost under $20,000 and, according to the company, will run five times faster than competing systems costing $60,000 or more. The recommended system includes an IBM PC AT with a 40-megabyte hard disk, high-resolution analog RGB monitor, and graphics tablet. The RGS cards, which are slated to be available in early 1987, occupy three slots in the PC AT. Contact Renaissance Graphics Inc., 1050 Walnut, Suite 325, Boulder, CO 80302, (303) 443-0191.
Inquiry 556.

CP/M on IBM PCs

Micro Solutions' UniDOS software lets IBM PCs and compatibles run 8-bit CP/M programs without additional hardware. The package emulates a Z80 microprocessor and provides an environment compatible with CP/M 2.2. You can run a CP/M program from DOS by typing its name. The two types of programs can reside on the same disk.

UniDOS retails for $69.95. Contact Micro Solutions Inc., Software Division, 125 South Fourth St., DeKalb, IL 60115, (815) 756-3411.
Inquiry 557.

(continued)

Is the cost of presentation slides catching up with you?

Slides are a vital part of professional presentations. Unfortunately, the cost of having them prepared at slide houses is becoming prohibitive.

Bell & Howell has an alternative. We call it the Bell & Howell Color Digital Imager. It's a complete, in-house system that teams up with your PC to deliver professional quality slides quickly and confidentially. And because it's designed to work directly with Lotus 1-2-3,® WordStar,® and most popular business programs, there's no need to learn any new software.

But even more important, you can prepare slides inexpensively. In fact, compare us to slide houses and you'll find we can save you over 90% per slide. That means you can use slides in almost any presentation.

The Color Digital Imager includes software and film recorder. It lets you compose word slides, charts and graphs. Add a graphics tablet and you can draw free form graphics. And with an optional printer or plotter, you can even generate overhead transparencies or hard copies for pass-outs or leave behinds.

To find out how the Bell & Howell Color Digital Imager can put you way ahead of the game, write Bell & Howell, Quintar Division, 411 Amapola Ave., Torrance, California 90501. Or call (213) 320-5700. We'll send you sample slides, complete information and the name of your nearest dealer.

BELL + HOWELL

Inquiry 243

Seen at great presentations everywhere.

PC Convertible Add-ons

Alloy announced the StarBase subsystem, one of the first add-ons for the IBM PC Convertible. Priced at $1995, StarBase-II is a combination 20-megabyte hard disk and 5¼-inch floppy drive with a parallel printer port. StarBase-III, a second model of the subsystem, also includes a 40-megabyte tape drive and sells for $2495. Both units attach to the expansion connector on the rear of the computer. For more information, contact Alloy Computer Products Inc., 100 Pennsylvania Ave., Framingham, MA 01701, (617) 875-6100.
Inquiry 558.

STB released a memory board, called the C-Ram 384, for the IBM PC Convertible. The C-Ram 384 plugs into a single slot and offers 384K bytes of CMOS RAM, expanding the portable's 256K base memory to 640K. The board connects directly to the IBM memory cards that come as standard equipment in the computer. The C-Ram 384 sells for $495. Contact STB Systems Inc., 601 North Glenville, Suite 125, Richardson, TX 75081, (214) 234-8750.
Inquiry 559.


The StarBase subsystem for the PC Convertible.

Desktop Publisher

Studio Software has released a desktop publishing package called FrontPage for the IBM PC XT and AT. The program uses an icon-oriented display and comes with a font library and preformatted document layouts. Features include full hyphenation and justification, capability to mix type styles and sizes, and on-screen editing. You can import graphics files from spreadsheet, drafting, and drawing programs.

To use the software, you need at least 512K bytes of memory (640K is recommended), a math coprocessor, graphics card and monitor, and a laser printer (Apple LaserWriter, AST TurboLaser, HP Laserjet, or QMS PS 800). FrontPage has a suggested retail price of $695. Contact Studio Software, 17862-C Fitch Ave., Irvine, CA 92714, (800) 821-7816; in California, (714) 474-0131.
Inquiry 560.

Five Graphics Modes on One Chip

NSI Logic announced a chip that emulates five graphics display adapters for the IBM PC. These include IBM's Monochrome Display Adapter (MDA), Color Graphics Adapter (CGA), Enhanced Graphics Adapter (EGA), and Professional Graphics Adapter (PGA), as well as the Hercules Graphics Card. Although several graphics boards are available that can emulate a variety of standards, the chip, called the EVC-315, is one of the first that features PGA emulation. In addition to these five modes, the chip supports a double-scanning mode with a resolution of 640 by 400 pixels, which is compatible with software using the CGA's resolution of 640 by 200.

The EVC-315 chip also has a RAM-resident character generator that can store 256 characters (expandable to 512 or 1024 characters) with a maximum size of 32 by 8 pixels. It can drive monitors at bandwidths from 15.75 kHz to 38 kHz and, the company claims, it has a high graphics throughput.

Available to OEMs, the chip costs $65 per unit in a minimum quantity of 10,000. Contact NSI Logic Inc., Cedar Hill Business Park, 257-B Cedar Hill Rd., Marlboro, MA 01752, (617) 460-0717.
Inquiry 561.

Internal CD-ROM Drive from Panasonic

Panasonic announced a half-height CD-ROM disk drive that fits internally in IBM PC–compatible computers. The Model SQ-D1 drive can be installed in the computer's floppy or hard disk slot and can access up to 540 megabytes of data from a CD-ROM disk. With the addition of an SCSI port, a single computer can access as many as eight CD-ROM drives.

The drive provides a maximum access time of 650 milliseconds and a data transfer rate of 1 megabyte per second. The unit also has a built-in audio interface; the interface card for the IBM PC has a stereo port that can be connected to a standard stereo set.

The internal drive is priced at $1495; an external standalone version (SQ-D101) is also available and costs $1795. Contact Panasonic Industrial Co., One Panasonic Way, Secaucus, NJ 07094, (201) 348-7000.
Inquiry 562.

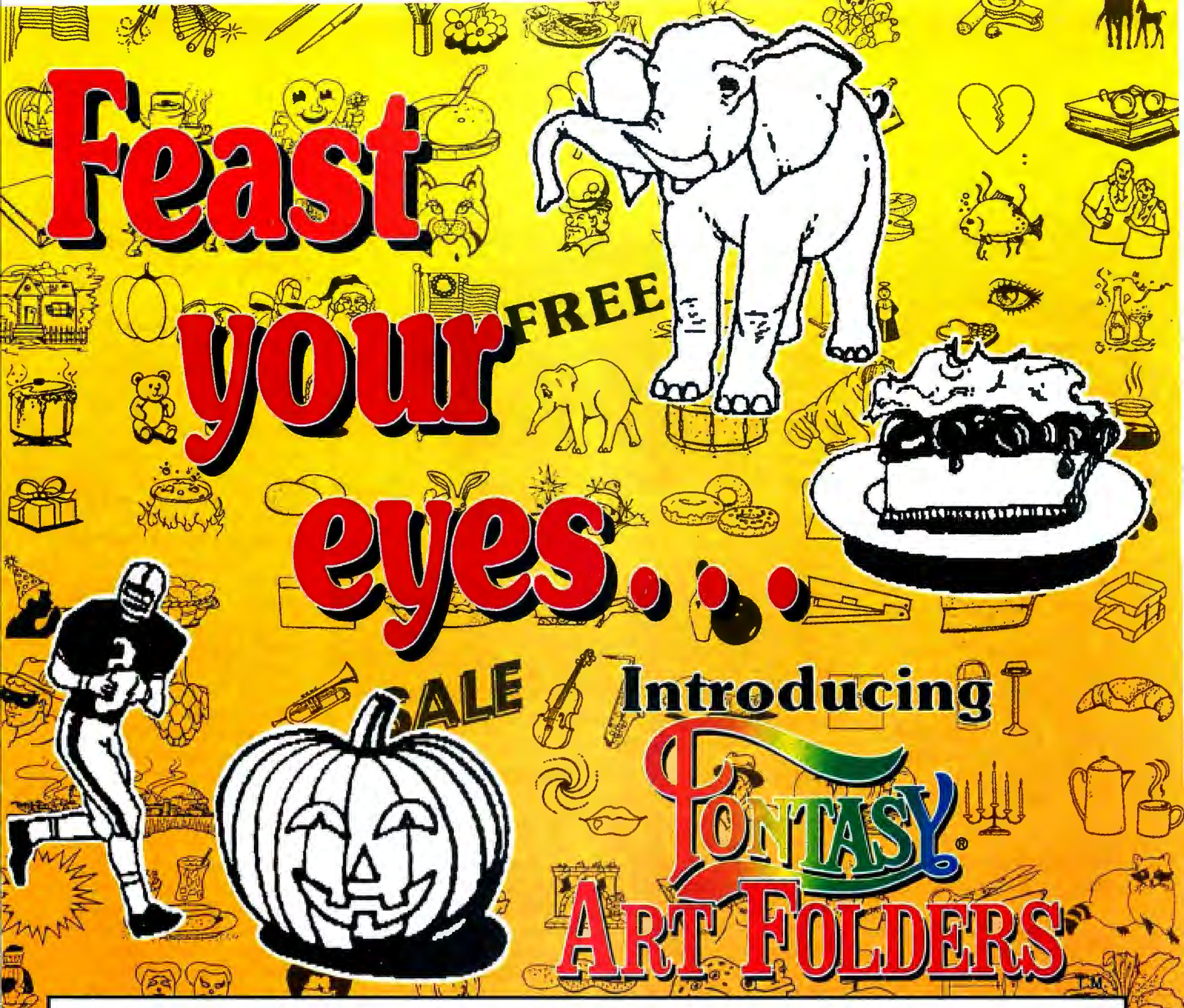
LAN Tester

The Smart LAN Performance Test program is a standardized test package that enables users to compare LANs in a simulated working environment. Innovative Software says test results measure throughput as well as the productivity levels you can anticipate in actual usage. The menu-driven package can be used on any LAN operating system compatible with IBM NETBIOS and DOS CALL 3.1 or later.

The tester chooses the application to be run on each workstation and the level of usage. The application runs in a loop on each station until stopped by the tester. After each workstation is "busied out," the tester can have Smart LPT perform timed tests. Smart LPT can be run on a single PC to establish a baseline timing for comparison. Results can be printed in a bar chart or spreadsheet.

The Smart LPT costs $49.95. Contact Innovative Software Inc., 9875 Widmer Rd., Lenexa, KS 66215, (800) 438-7627; in Kansas, (913) 492-3800.
Inquiry 563.

(continued)



Feast your eyes...
FREE
SALE
Introducing
FONTASY®
ART FOLDERS
T.M.

from the QINT/SQL database.

QINT/SQL needs 640K and a hard disk. Single-user price is $995; multiuser versions start at $1395.

QINT also sells a set of fifth-generation tools to work with QINT/SQL: QINT/TINA, a package that automates Nijssen's Information Analysis Methodology, a stepwise approach to data analysis based on semantic modeling concepts. TINA takes you through the necessary steps with a graphical interface and integrates and validates the components of the final Information Structure Diagram. TINA for micros is $5000.

Contact QINT Database Systems Corp., 125 Roberts Rd., Waltham, MA 02154, (617) 891-3377.
Inquiry 568.

Tandon Announces Line of PCs

After years of selling personal computers under other manufacturers' labels, Tandon has announced a line of computers that it will sell through retailers under its own label.

The XT-compatible models, called the PCX series, share the same capabilities and features as the IBM PC XT except that they include a parallel port as standard. A system with two floppy disk drives and 256K bytes of memory costs $1295. A 14-inch amber monochrome monitor lists for $190, and a display adapter card, which can display either text but no graphics on a monochrome display or color graphics on an RGB monitor, lists for $90. A serial port costs about $100.

The AT-compatible models, called the PCA series, are identical to the IBM PC AT except that they feature a dual-speed 80286 processor that runs at 6 or 8 MHz with one wait state. In addition, the operating system for these models can access disk drives that are larger than 32 megabytes, the limit for most MS-DOS-based systems. The PCA with a 1.2-megabyte floppy and 512K bytes of memory costs $2195, without a monitor. The PCA is also available with 20- and 40-megabyte hard disk drives for $2695 and $3595, respectively. Total system cost for a 20-megabyte unit with monitor and serial port is $3075. For more information, contact Tandon Corp., 20320 Prairie St., Chatsworth, CA 91311, (818) 993-6644.
Inquiry 569.

The Transpac portable hard disk drive for IBM PCs.

Portable Hard Disk

The Transpac from Microcomputer Memories is a portable 3½-inch hard disk drive that connects via cable to IBM and compatible computers. The unit has its own controller and uses the computer's power supply. Its average access time is 65 milliseconds.

The drive weighs less than 3½ pounds and measures about 2½ by 4 by 6 inches. Prices are $595 for a 10-megabyte model and $895 for a 20-megabyte model. For more information, contact Microcomputer Memories Inc., 6855 Vineland Ave., Chatsworth, CA 91605, (818) 700-8000.
Inquiry 570.

WHERE DO NEW PRODUCT ITEMS COME FROM?
The new products listed in this section of BYTE *are chosen from the thousands of press releases, letters, and telephone calls we receive each month from manufacturers, distributors, designers, and readers. The basic criteria for selection for publication are: (a) does a product match our readers' interests? and (b) is it new or is it simply a reintroduction of an old item? Because of the volume of submissions we must sort through every month, the items we publish are based on vendors' statements and are not individually verified. If you want your product to be considered for publication (at no charge), send full information about it, including its price and an address and telephone number where a reader can get further information, to* New Products Editor, BYTE, *One Phoenix Mill Lane, Peterborough,* NH 03458.

Fiber-Optic Ethernet Connections

Codenoll Technology introduced two fiber-optic Ethernet connections for the IBM PC, XT, AT, and compatibles that operate at a rate of 10 megabits per second. According to the company, the use of fiber optics permits throughput five times greater than that of broadband systems and two and a half times greater than that of twisted-pair systems. The connections, each of which comes on a plug-in board, contain a 2000-meter fiber-optic Ethernet Transceiver.

The Codenet-3051 Fiber Optic Ethernet PC Connection is compatible with 3Com products and costs $995. The Codenet-3061 Fiber Optic PC Network Adapter Card, which sells for $1195, is compatible with Sytek products and IBM's NETBIOS. For more information, contact Codenoll Technology Corp., 1086 North Broadway, Yonkers, NY 10701, (914) 965-6300.
Inquiry 571.

PC-based Protocol Analyzer

FELINE (Frederick Engineering's Dataline Monitor/Protocol Analyzer) is a general-purpose data communications protocol analyzer on a board that plugs into an IBM PC or compatible computer. Priced at $1595, the board features multiple protocols, bit error rate testing, full-screen graphics presentation, and data analysis. FELINE comes with software, an RS-232C interface pod, and a carrying case. Contact Frederick Engineering Inc., 54 Cessna Court, Gaithersburg, MD 20879, (301) 926-6772.
Inquiry 572.

LOOKING FOR THE BEST VALUE IN PC/AT COMPATIBLES?

INTRODUCING THE $1495 A★STAR™

Looking for a lower priced, higher quality PC/AT compatible than our new A★Star™ is like looking for a needle in a haystack. You're more apt to come up with a case of hayfever.

A★Star gives you features you won't get from *any* other vendor. Not IBM. Not Compaq. Not even those foreign manufacturers. Features like a 220 watt power supply, eight *available* expansion slots and "network ready" multi-user operation. All for only $1495. And that price includes a 1.2MB diskette drive, 512KB memory, a fixed disk/diskette drive controller, a tactile feedback keyboard (you'll love it!) and a 6/8MHz switch/software selectable 80286 CPU.

Best of all, A★Star is quality built right here in America by Wells American Corporation - an American Stock Exchange manufacturer. So now you can buy with confidence from a vendor you can trust, just like you've trusted IBM. And you'll get a genuine, top quality product (not a cheap imitation) for a price even less than those "questionable" mail order clones. And if that's not enough, it's all money-back guaranteed!

The $1495 Wells American A★Star.™

A PC/AT FOR ONLY $1495?
THAT'S NOTHING TO SNEEZE AT!

[Hurry! This offer *is* limited.]

☐ I'm tired of sneezing! Have someone call me immediately to take my order.

☐ A★Star sounds terrific. Tell me more.

Name: ____________________

Company: ____________________

Phone: ____________________

Address: ____________________

City: ____________________

State: ____________ Zip: ________

Wells American
803/796-7800•TWX 510-601-2645
Sunset Boulevard • West Columbia, SC 29169

IBM, Personal Computer AT and AT are trademarks of International Business Machines Corporation

Inquiry 301

A screen from Profit Technology's Breakthrough! package.

Two Megabytes of Memory for $398

The Breakthru/PC, a plug-in board for IBM PCs and compatibles, provides 2 megabytes of memory and conforms to the Lotus/Intel/Microsoft Expanded Memory Specification. The board sells for $398 with 2 megabytes of RAM or $150 without RAM.

Bundled with the board are RAM disk and print spooler utilities, as well as diagnostic software and a setup program for changing the configuration of expanded memory. The company claims that the board performs bank switching 15 percent faster than Intel's Above Board and 25 percent faster than AST's RAMpage. Contact Vericomp, 8825 Aero Dr., Suite 210, San Diego, CA 92123, (619) 277-0400.

Inquiry 564.

Productivity Tool

Profit Technology announced Breakthrough!, a productivity package that has knowledge databases and idea stimulants.

The program is divided into three sections: Interface and Work, Thought Triggering and Amplification, and Tools and Utilities. In the Interface and Work section, the user can use eight different text-editing environments. The section includes a 220,000-word thesaurus. In the Thought Triggering and Amplification section, you can access a preprogrammed database, which includes menu selection, word triggers, and five flashing windows. The Tools and Utilities section includes a biorhythm generator, a program customizer, and a report generator.

ProView, the windowing/applications manager, is a clone of IBM's TopView and enables you to run 250 programs simultaneously through its multitasking capability.

Breakthrough! costs $822 and requires an IBM PC, XT, AT, or compatible with 512K bytes of RAM, two floppy disk drives, a monochrome monitor, and PC-DOS or MS-DOS 2.0 or higher. Profit Technology recommends using an IBM EGA, a color monitor, and a hard disk. For more information, contact Profit Technology Inc., 39 Broadway, New York, NY 10006, (212) 809-3500.

Inquiry 565.

Hewlett-Packard's DraftPro Plotter

The DraftPro from Hewlett-Packard is an eight-pen plotter designed for PC-based computer-aided design. The plotter, which sells for $5400, produces color output on architectural and engineering C- and D-size and metric A1- and A2-size paper, vellum, or film.

Compatible with most computers, the DraftPro is supported by CAD packages such as AutoCAD and VersaCAD. It features a pen speed of 15 inches per second, a 0.001-inch addressable resolution, and 0.2 percent linear accuracy. The plotter's pen-sorting capability enables the device to draw a full buffer of vectors for one color before proceeding to another color, which reduces plot time.

An RS-232C interface and HP-GL support are standard; an HP-IB interface and Japanese kanji character set are available as options. For more information, contact Inquiries Manager, Hewlett-Packard Co., 1820 Embarcadero Rd., Palo Alto, CA 94303, (800) 367-4772.

Inquiry 566.

Pop-up Outliner, Pop-up Calculator

Popular Programs has two new RAM-resident packages for IBM PCs and work-alikes: an outline processor and an equivalent of Hewlett-Packard's 12C calculator.

Pop-Up Partner, an idea/outline processor, is designed to help you organize information as lists. It uses pull-down menus and comes with outlines that serve as desktop tools (e.g., a phone book, calendar, and expense tracker). The "paragraph" feature lets you write while in the middle of an outline; anything you've written in Partner can be copied in a word-processor or spreadsheet document with a couple of keystrokes.

Partner takes up 99K bytes. It sells for $99.97.

Pop-Up PC-12C Calculator emulates HP's hand-held financial calculator and has the same programming features. It costs $69.95.

Both Pop-Ups run on the IBM PC line under any version of DOS. For more information, contact Popular Programs Inc., 135 Lake St., Suite 180, Kirkland, WA 98033, (206) 822-7065.

Inquiry 567.

Relational Database Compatible with SQL

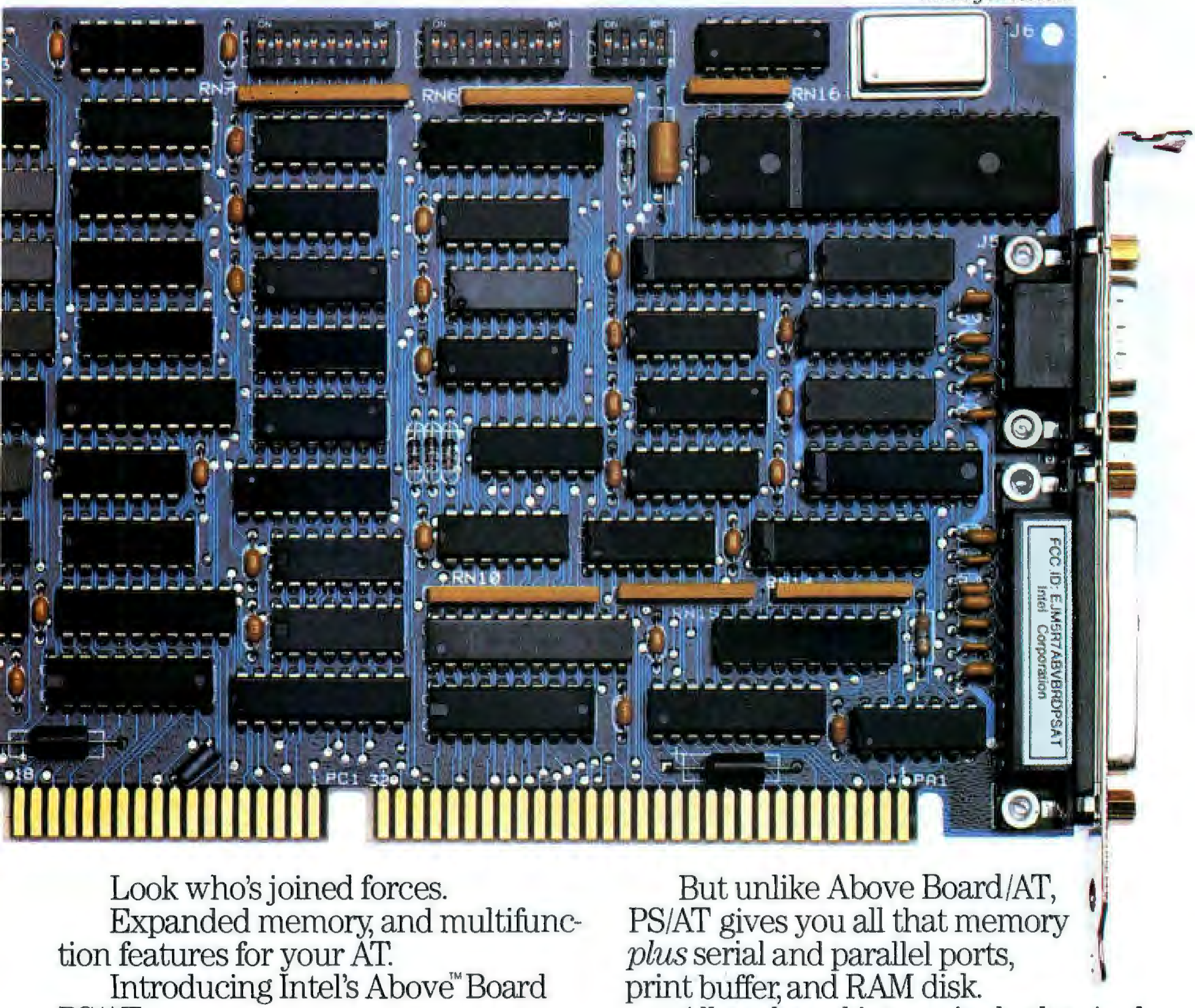
QINT/SQL, a relational database system compatible with IBM's Structured Query Language/Data System and Database 2, is designed primarily for users of PCs in an environment where SQL/DS or DB2 is installed on IBM mainframes. The package enables you to access and manipulate mainframe data from a PC and to prototype systems on the micro and then migrate them to the bigger machine.

With the QINT/Application Development Tool, you can build customized end-user applications that utilize a database. ADT can work on several sets of rows of a table simultaneously. Calculations and system values can be shown on the same screen as data retrieved

(continued)

Intel announ

Multifunction



Look who's joined forces.

Expanded memory, and multifunction features for your AT.

Introducing Intel's Above™ Board PS/AT.

Like our Above Board/AT, PS/AT gives you expanded memory—which meets the Lotus®/Intel/Microsoft® spec, conventional memory—up to 640K, and extended memory—capable of supporting protected mode operating systems.

But unlike Above Board/AT, PS/AT gives you all that memory *plus* serial and parallel ports, print buffer, and RAM disk.

All packaged into a single slot. And all guaranteed to run at 8MHz.

PS/AT not only supplies all the memory you need, it's supported by some of the most popular software. Including 1-2-3,® Symphony,™ Framework II,™ SuperCalc,® Ready!,™ and Auto CAD.™

VERSANET.
VERY, VERY VERSATILE.

CTC SYSTEM'S VERSANET™ PRODUCT FAMILY ENABLES APPLICATIONS FROM MULTI-USER, MULTI-TASKING TO PC LAN NETWORKING AND GENERAL PURPOSE I/O CONTROL FOR IBM PC, XT AND AT.®

Our Versanet™ family of data communications products can fill your needs today and tomorrow. Versanet products support a large number of multi-user/multi-tasking operating systems including Concurrent PCDOS,® PC-DOS,® Xenix® and Unix.® They can let you port up to eight RS232C devices, or as few as four. They can deliver user-programmable on-board intelligence to free microprocessor resources. They can even increase RS232C device transmission distance up to 100% at any baud rate.

And our support of RS232C protocols is complete. For example, we provide TXD, RXD, DTR, DSR, CTS, RTS, RI and DCD. That means full modem handshaking and support of virtually all serial I/O devices.

The Versanet family was designed intelligently, too. Like requiring only 8 I/O addresses *per board* vs. the 8 *per port* other boards require. Like using VLSI Technology to reduce cost and raise reliability. And like offering a hardware fingerprint mechanism to prevent software piracy and allow polling the number of Versanet boards in the chassis.

Versanet boards also come with a connector housing for easy interface to I/O devices external to the PC.

Best of all, these products are proven. We've been shipping for over a year to companies, as big as AT&T and as small as a dream.

Let Versanet products help make your dream. They're very, very versatile. For more information call (408) 725-0501. Or write CTC Systems Inc., 10311 S. De Anza Blvd., Suite 4, Cupertino, CA 95014.

CTC SYSTEMS, INC.

Dealer inquiries welcome.

Inquiry 78

ADVANCED MS-DOS
Ray Duncan
Microsoft Press
Redmond, WA: 1986
468 pages, $22.95

TECHNICAL TOPICS FOR THE IBM PC FAMILY OF MICROCOMPUTERS: An Annotated Bibliography of Recent Books

---

ADVANCED MS-DOS
Reviewed by John D. Unger

If you are a programmer who needs to interface software and hardware with MS-DOS, *Advanced MS-DOS* by Ray Duncan will guide you. You can learn how to use MS-DOS interrupts and their associated functions so your application programs can do file I/O and write to the screen, or you can figure out how to access peripherals quickly and efficiently.

Given the title and subject matter of the book and its author's expertise, you might expect *Advanced MS-DOS* to be difficult to read, but that's not so. Duncan's informal style is carried over from his regular columns in *Dr. Dobb's Journal of Software Tools*. The opening chapter on the history of MS-DOS, for instance, reads like a good biography. *Advanced MS-DOS* exemplifies how a highly technical book can be both informative and readable.

READER REQUIREMENTS

To benefit from this book, you should be able to understand the 8086 assembly language programs that are liberally sprinkled throughout the text as examples. If you have already worked through a good introductory book on 8086 assembly language programming—such as *Assembly Language Primer for the IBM PC & XT* by Robert Lafore (New American Library, 1984)—you will be able to launch yourself into a higher level of assembly language programming with *Advanced MS-DOS*.

Duncan uses C-language programs as examples to explain how to interface high-level programs with MS-DOS functions. But because he is basically an assembly language programmer, he emphasizes writing efficient assembly language code either as stand-alone programs or as functions or procedures to be linked to other high-level language programs. Many of the programs are useful not only as examples but also as the basis of a library of valuable MS-DOS utility and programming tools.

Unless you enjoy typing in endless lines of source code and then searching for syntax errors and other self-induced bugs, you may want to take advantage of the companion disk to *Advanced MS-DOS*. This disk contains the assembly and C-language source code for all the example programs described in the book. Also included are ready-to-run, executable versions of these same examples. Although Duncan recommends using the latest versions of the Microsoft Macro Assembler, MASM version 4.0, and Linker, LINK version 3.05, to assemble and link the source code, I had no problems using MASM version 1.25 and LINK version 2.01 on an AT&T 6300. The .EXE and .COM programs ran perfectly.

The disk also includes utility and example programs that do not have source code listings in the book but that illustrate topics the author discusses. This disk is a great timesaver for those who will actually use the example programs; an order blank for the disk is included in the book.

APPEARANCE AND ORGANIZATION

The layout of *Advanced MS-DOS* is both handsome and helpful. The example programs and fragments of source code are printed in a reverse-video effect with black characters on a light brown background. This makes the code listings stand apart from the text without interfering with their readability.

The material is divided into four sections. The first and largest describes the MS-DOS system services on a category-by-category basis. Section 2, a complete reference guide to all the MS-DOS interrupts, is more than you would expect from a typical programmer's reference manual. Included with the description of how to load the proper registers to call a specific function are notes about the "nonstandard" behavior of the function and how it behaves under different versions of MS-DOS. This is clearly an added bonus of the author's expertise. Every function's description includes a short assembly language example program that can serve as invaluable code fragments when you're writing your own programs to call functions. Two final, shorter sections are reference guides to the special interrupts in the IBM PC ROM BIOS and to the Lotus/Intel/Microsoft expanded memory specifications.

HIGHLIGHTS

One factor that distinguishes *Advanced MS-DOS* from a technical reference manual is the way in which the author treats nonstandard methods of accessing system services. Duncan explains the "correct" and most portable method for accessing an operating system service. He also demonstrates how to use less portable, hardware-dependent methods when speed (rather than portability or compatibility with future versions of the operating system) is the driving force. This approach makes the book more valuable than a mere transliteration of a manual. The author's effectiveness as a programmer adds real merit to these explanations, providing insights into how to accomplish your programming tasks most efficiently.

A chapter on accessing MS-DOS's file- and record-manipulation functions is one of the book's highlights. Because of the historical ties between CP/M and MS-DOS, two methods of controlling files exist in MS-DOS. Duncan provides an in-depth discussion of file management first by comparing the file control block (FCB) and file han-

(continued)

dle methods of file access and then by explaining why he recommends the latter method.

The author devotes an entire chapter to programming for the character I/O devices, such as the video display, keyboard, printer, and serial port. He discusses the pros and cons of using three distinct methods of video display control on MS-DOS and PC-DOS microcomputers.

One example, from the chapter that discusses the MS-DOS EXEC function, presents a program for a simple, table-driven command interpreter shell that overlies COMMAND.COM. This program is designed as a model that can be expanded to provide a customized user interface for special applications. Two parallel listings of the program are presented, one written in C and the other in assembly language. The functions and procedures in the two listings are similar enough to enable the reader to compare and contrast the two versions of the program.

A Worthy Reference

Advanced MS-DOS is a good example of what a technical reference manual should be. Because my interest is in graphics programming, however, I wish that the author had addressed efficient use of the Enhanced Graphics Adapter functions. But because that is mainly a hardware-dependent problem, its lack of coverage does not detract from the book or its intention.

Clearly, Duncan's strengths include a style that is at once easily read, a thorough coverage of the subject matter heretofore unknown, and the frequent use of examples in the form of assembly language programs and code fragments. *Advanced MS-DOS* contains a great deal of valuable information that I know I will frequently refer to.

John D. Unger (P. O. Box 95, Hamilton, VA 22068) *is a geophysicist for the U.S. Government who also writes graphics programs in assembly language and C for microcomputers.*

TECHNICAL TOPICS FOR THE IBM PC FAMILY:
An Annotated Bibliography
Compiled by Donald Evan Crabb

This collection of book annotations, intended as a follow-up to those that appeared in BYTE's last IBM Special Issue (Fall 1985), covers several recently published books on technical topics addressing the IBM PC, XT, PCjr, AT, and compatible computers. The books, all softcover, are divided into three sections: hardware issues, PC-/MS-DOS and related operating systems issues, and programming. This bibliography reflects availability and bias and cannot be considered inclusive.

Hardware Issues

This topic is a catchall for books that cover the hardware design and architecture of the PC, including such issues

(continued)

as connecting your PC to a variety of external devices, how to design the hardware and software interfaces that are necessary, how to network PCs, and discussions of the hardware and software that make up the PC.

Berry, Paul. *Operating the IBM PC Networks*. Berkeley, CA: Sybex, 1986. $19.95. 363 pages.

As local area networks become more important to PC users, the plethora of conflicting network topologies, protocols, and systems becomes more and more mind-boggling. Paul Berry's *Operating the* IBM PC *Networks* helps to sort out some of this information for both the newly released IBM Token Ring network (a baseband configuration) and the older broadband PC Network.

This handbook should prove very useful to LAN users of either network, as well as to potential network customers. It covers information for network administrators, end users, and network consultants. The introduction to LANs is helpful, and the 14 chapters that follow cover most of the important aspects of the two networks in detail. Topics include how to set up and run the equipment, how to automate network usage for end users, how to start and restart the networks, how to use electronic mail over the LAN, how to manage shared resources and disks, and how to optimize the network performance.

The book's only drawback is that it limits coverage to PC Network and IBM Token Ring. I'd like to see similar handbooks for some of the other network systems crowding the market.

Byers, TJ. *Inside the IBM PC AT*. New York: McGraw-Hill, 1985. $19.95. 309 pages.

If you can ignore its enthusiastic narrative style, TJ Byers's *Inside the IBM PC AT* will reward you with comprehensive technical information about the AT. The author covers all the important hardware and software aspects of the AT in detail for experienced PC users and programmers.

Byers provides descriptions and illustrations to help novice AT users get their machines up and running, but the heart of the book lies in the technical discussions of PC-DOS, custom programming, and AT hardware.

Eggebrecht, Lewis C. *Interfacing to the* IBM *Personal Computer*. Indianapolis, IN: Howard W. Sams & Co., 1985. $16.95. 246 pages.

If you're about to attempt to interface a variety of devices to your PC and want to find out just what general issues are entailed, *Interfacing to the* IBM *Personal Computer* is a good place to start. Unfortunately, this reference handbook doesn't present specific interface designs, so much of its practical utility is lost. However, it does provide general information about interfacing the PC, details about the PC

(continued)

A complete list of things to know about 2400 bps modems.

Now that you've memorized that, here's a partial list of why a Hayes® Smartmodem 2400™ is best for you.

1. The Hayes Smartmodem 2400 allows you to communicate with the vast installed-base of 300,1200 and 2400 bps "Hayes-compatible" modems. The Hayes Standard "AT" Command Set allows you to use Smartcom II® and other software that communicates.

2. Through synchronous/asynchronous technologies, the Smartmodem 2400 permits your PC to access mainframes, minis, and on-line services previously inaccessible through asynchronous-only modems.

3. The Hayes Smartmodem 2400 is efficient...it pays for itself in just 4 hours of annual use over long distance.

4. The technology of the Smartmodem 2400 allows you to transfer volumes of files with confidence across the city or across the ocean using Bell and CCITT standards.

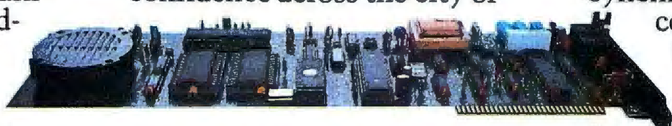
5. The new Smartmodem 2400B™—a plug-in board for the IBM PC and compatibles—allows synchronous and asynchronous communication through the same Com port.

6. You will also get the Hayes standard 2-year limited warranty and the opportunity to extend the warranty to 4 years.

Best of all...you get Hayes. And that's all you ever really have to know!

For more information or technical specs, contact your authorized Hayes dealer. Or Hayes directly at (404) 441-1617.

Hayes Microcomputer Products, Inc., P.O. Box 105203, Atlanta, Georgia 30348.



Hayes

Say yes to the future with Hayes.

bus, system interrupts, the 8088 microprocessor, memory, and interface signal conditioning. The book would have been much more useful had it discussed some sample interfaces in detail.

Foster, Dennis L. *The Practical Guide to the IBM Personal Computer AT*. Reading, MA: Addison-Wesley, 1985. $19.95. 455 pages.

Of the two handbooks annotated here that discuss the IBM PC AT (the other being Byers's book, above), this one is by far the better book for the novice. The architecture, design, and operation of the AT are covered in prose that users new to the AT will find easy to digest. Dennis Foster covers a number of software issues, including TopView, XENIX, PC-DOS (3.0 and 3.1), and the basics of networking. Hard disks and the new quad-density floppy disk drives are also examined.

Despite its 455 pages, *The Practical Guide to the* IBM *Personal Computer AT* seems a little thin, especially since its design leaves much of each page blank. The illustrations are also overly simplistic. For technical information about the AT, I preferred the Byers book.

Morse, Stephen P., and Douglas J. Alpert. *The 80286 Architecture*. New York: John Wiley and Sons Inc., 1986. $24.95. 279 pages.

Stephen Morse, the principal architect of the Intel 8086, has previously written about the 8086 and 8087. Douglas Alpert was formerly a staff engineer at Intel Corporation and along with Morse was instrumental in the development of the Intel 86 family of microprocessors. These men share impressive credentials for writing about the 80286 architecture, and they know what they're talking about. In a nutshell, their book is authoritative yet accessible to programmers who don't have a lot of assembly language background.

Morse and Alpert provide more than a detailed analysis and description of the Intel 80286 and 80287 chips. They discuss the chips' instruction sets; how their multitasking, virtual memory, and system security features work; and how you can build a computer around them. Further, the authors show how to program the 80286 and 80287 for scientific, business, and industrial purposes.

Morse and Alpert also offer us a brief history of the development of the modern microprocessor from the point of view of Intel Corporation. If you want the inside information about the 286/287, this is the book to read.

PC-/MS-DOS

More capabilities have been added to DOS as the PC has matured from an 8088-based machine to one using the

(continued)

80286 chip (DOS versions 1.1, 2.0, 2.1, 3.0, 3.1, 3.2, etc.). The books in this section are user's guides for DOS, written from the point of view of the serious PC user who wants more technical and practical information about the operating system and a DOS reference that's easy to use.

DeVoney, Chris. *Using PC-DOS*. Indianapolis, IN: Que Corporation, 1986. $21.95. 519 pages.

Chris DeVoney's *Using PC-DOS* is an interesting combination of a PC-DOS tutorial, a PC-DOS advanced guide, and a PC-DOS reference handbook. The material is familiar, of course, but the structure and explanations of this book help set it a step above the ordinary. The book could also serve as a textbook for a course on using DOS.

It's easy to read, and it covers all the DOS commands and usage that it should. The command reference section is particularly handy. Along with Wolverton's book (below), it's my favorite DOS handbook, primarily for its clarity and writing style.

Kamin, Jonathan. *MS-DOS Power User's Guide*. Berkeley, CA: Sybex, 1986. $19.95. 397 pages.

If you use your PC for more than a few hours a day, this is a useful book. Jonathan Kamin lists batch programs, BASIC programs, display screens, macro files, and assembly language programs, all of which are carefully designed to help you get more out of your PC. For example, you can configure your system using a chapter on writing CONFIG.SYS and AUTOEXEC.BAT files. If you want to customize the screen colors on your display, just refer to the chapter on customizing your screen and keyboard.

MS-DOS Power User's Guide contains valuable information about the PC and DOS. Undocumented commands and hard-to-find technical details abound. Much of what Kamin presents is available elsewhere, especially in the Peter Norton books (see last year's IBM bibliography), but the organization here is half the reason for buying the book. Kamin has made it easy to find the technical trickery you need to make your PC jump through hoops.

King, Richard Allen. *The MS-DOS Handbook*, 2nd edition. Berkeley, CA: Sybex, 1986. $19.95. 339 pages.

The MS-DOS Handbook is really two reference books in one. Richard King divides his text into a programmer's handbook and a user's handbook. The programmer's handbook discusses the detailed specifications of disk and file operations, keyboard control, monitor and port control, and memory management of MS-DOS. King also includes full descriptions of all operating system functions needed to write MS-DOS programs, including good programming examples.

The user's handbook includes information on EDLIN,

(continued)

DOS files, how to configure the system, batch processing, monitors, serial ports, and more.

The MS-DOS Handbook contains a good discussion on the issues of software compatibility, offers a guide to the different versions of MS-DOS. This is a useful book, especially for its programmer's handbook discussions. The novice material is neither as useful nor as detailed.

Siechert, Carl, and Chris Wood. ***The Power of Running PC/DOS***. Portland, OR: Management Information Source, 1986. $22.95. 386 pages.

The Power of Running PC/DOS is really a handbook for the latest versions of DOS (3.0, 3.1, and 3.2). It can be used equally well by novices and experienced PC users. Siechert and Wood cover familiar ground in their PC setup instructions, DOS command references, directory structures, and DOS utility programs. The book also includes a number of useful BASIC programs that you can type in and run.

It's clearly designed, easy to read, and contains useful examples. Overall, though, it's a bit thin; other books mentioned cover the same material in greater detail.

Wolverton, Van. ***Running MS-DOS***, 2nd edition. Redmond, WA: Microsoft Press, 1985. $21.95. 423 pages.

This book is simply the definitive handbook of PC-/MS-DOS. It has been expanded and improved to cover the newer releases of MS-DOS. *Running MS-DOS* is written for both novices and experienced users. Carefully structured, it contains copious examples and is a model of expository clarity. Van Wolverton's new edition fixes some mistakes of the first edition, adds new material, and explains all the commands and techniques users need to know about managing files and disks, creating custom commands, automating the execution of frequently used commands, managing a hard disk, and manipulating the screen display, printer, and telecommunication devices.

PROGRAMMING

The IBM PC and its compatibles work with a variety of high-level programming languages (through PC-/MS-DOS and CP/M-86): BASIC (interpreted and compiled), Pascal, FORTRAN, COBOL, LISP, and C. Also, the PC can be programmed effectively with 80xxx assembly language. The books listed here teach some technical aspects of high-level or assembly language programming on the PC.

Busch, David D. ***Program Your IBM PC to Program Itself!*** Blue Ridge Summit, PA: Tab Books, 1986. $12.95. 145 pages.

Though the title sounds strange, *Program Your IBM PC to Program Itself!* delivers on its promise. David Busch has

(continued)

MicroComputer Accessories, Inc.

MASTERING THE SYSTEM.

Announcing the next generation of MicroComputer Accessories!

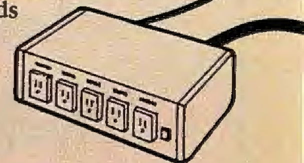
The MicroConsole Control Module

THE MICROCONSOLE™ REMOTE CONTROL CENTER

Complete power control and protection—and more!—with a remote control panel compact enough to rest on the top edge of a keyboard—or anywhere you want it!

This isn't another big switch box under the monitor. We got it all together by taking it apart. The MicroConsole actually consists of two separate wire-connected modules—the Control Module and the Power Module.

The Control Module, a wired-remote control panel, stays close at hand on the keyboard or desktop. It features membrane-type on/off switches for five components and peripherals, with fail-safe Master and CPU switches—plus a calculator and digital clock! The slim, compact Control Module sits on the keyboard or stands alone on its own detachable feet.



The Power Module stays on the floor and keeps messy cables out of the way. It provides outlets for five components and peripherals, with full three-path surge protection and telephone line conditioning for your modem.

The Static Clear

The Surge Protector

THE STATIC CLEAR Dissipator w/LCD Display

Seeing-is-believing static protection. Before touching your computer, touch the Static Clear. When the lightning bolt on the LCD display fades, you know *for sure* that harmful static has been dissipated.

THE SURGE PROTECTOR Power Strip

The best-designed and best-priced power strip available! Six outlets with full three-path surge protection and a circuit breaker switch with an indicator light. There's a slot for wall hanging and a retaining clip for organizing cables. Neat!

SA: 5405 Jandy Place, P.O. Box 66911 • Los Angeles, CA 90066-0911 • Phone 213/301-9400 or 800/521-8270 • Telex 4993614 MICRO-AC • FAX 213/3068379
UROPE: N.V. MicroComputer Accessories Europe S.A. • Rue de Florence • 37 • 1050 Bruxelles, Belgique • Phone 02/538.61.73 • Telex 24088 CETREL
ESTERN CANADA: Phone 800/361-3156 • CENTRAL & EASTERN CANADA: Phone 800/361-2418

assembled a collection of 19 BASIC programs that will save you a lot of time and effort. Included are programs that display graphics screens for presentations, create the skeleton for database programs, check programs for spelling mistakes or incorrect punctuation, provide screen editing, count the words in your document or program, redefine function keys, and design custom screens.

Each program is listed in full (a disk containing the programs is available for $24.95), and a useful description is also provided. The book assumes that you are a PC BASIC programmer and are familiar with PC-DOS.

Campbell, Joe. *Crafting C Tools for the IBM PCs*. Englewood Cliffs, NJ: Prentice-Hall, 1986. $21.95. 434 pages.

Just like Stevens's book discussed below, *Crafting C Tools for the IBM PCs* provides examples and tips for making programming tools. This book is also a C-language reference handbook, as well as a handbook for Intel 8088 assembly language and for PC-DOS. The emphasis here is more on using a combination of C and assembly language to perform such tasks as calling DOS and ROM BIOS routines, handling interrupts and screen cursor management, processing the PC's function keys, reading and setting the default drive and directory path, changing a file's time and date stamp and reading the system clock, and creating screen windows and pop-up menus. Although Joe Campbell's book has a wider reach and is easier for less experienced programmers to read than Al Stevens's *C Development Tools for the* IBM PC, novices should probably stay away. The book has a solid structure, with good use of examples. If you know C programming well and are going to use it frequently, put this book on your shelf, along with Stevens's.

Miller, Alan R. *Assembly Language Techniques for the IBM PC*. Berkeley, CA: Sybex, 1986. $22.95. 358 pages.

Assembly Language Techniques for the IBM PC helps you use and adapt assembly language utility programs for your PC. It is structured around a powerful library of assembly language macros that are discussed, explained, and dissected. Alan Miller has developed programs utilizing these macros and he teaches readers how to use them. Miller's book is written in a manner designed to teach you how to use the material presented—and it does a good job.

Miller's treatment of the subject is fairly comprehensive: He covers the ROM BIOS (both versions 2 and 3); PC-DOS (versions 2 and 3); and the system debugging program, DEBUG. The primary focus, however, is on the collection of macros and programs he has assembled that should run on PCs and ATs. These programs handle screen, port, and keyboard control, as well as memory management and disk I/O. A particularly useful program example shows

(continued)

Interoffice Memo.

A single MaynStream™ backs up *all* the PCs in your office, file by file!

Protecting Files: Top Priority.

Hard drive users learn fast how indispensable tape backup really is. But equipping each of your PCs with its own backup system is a costly way to go. That's why we developed the MaynStream, our lightning-quick, *file-by-file* tape backup. When you've backed up one PC, just detach the MaynStream's Quick-Release Cable and snap it onto any other PC equipped with a low-cost MaynStream controller. Simple menus will guide you through another backup in seconds. And restoring data is just as easy. So when your hard drive forgets to treat your files like your business depends on them, the MaynStream remembers.

Backup That Won't Back Down.

The MaynStream is small, lightweight, and big on performance features. Its rugged design and convenient carrying handle make multi-PC backup a breeze. And talk about fast! Backup takes just 3 minutes* for the average 10MB drive (that's file-by-file, and it's still faster than the now antiquated image mode technology)! The MaynStream uses handy data cassettes small enough to slip into your shirt pocket. Available in 20 and 60 MB models.

Other Powerful Systems for the PC.

MaynStream internal, available in both 20MB and 60MB models.

MaynStream 60Q internal, available in 60MB model.

MaynStream portable, available in both 20MB and 60MB models.

MaynStream 60Q portable, available in 60MB model.

Transport Plus™ portable combination 20MB hard drive and 20MB tape backup.

Transport™ 20MB portable hard drive.

All Maynard systems are backed by our 1-year warranty.

Sold at the finest computer stores. Contact your local dealer or write to us today for product information.

Maynard Electronics

Shaping tomorrow's technology.

460 E. Semoran Blvd., Casselberry, FL 32707 305/331-6402 Inquiry 186

*Actual time: 2min/57sec. Total files backed up: 376 consisting of 10,002,432 bytes. Computed on IBM PC with 512K memory and one 10MB hard disk drive. Your actual backup time may vary. IBM PC is a trademark of International Business Machines.

you how to expand the PC's usable memory beyond 640K (to 704K) by using the 64K-byte block of memory normally left unaddressed by DOS.

To use the material presented by Miller, you should have experience as a high-level and assembly language programmer. You'll also need a PC with at least two disk drives, 256K bytes of RAM, the Microsoft or IBM assembler, a linker program, and a debugger. If you have the requisite hardware, software, and programming experience, *Assembly Language Techniques for the IBM PC* will reward you with a straightforward discussion of the technical problems and solutions involved in assembly language macro programming on the PC.

Scanlon, Leo J. *80286 Assembly Language on MS-DOS Computers*. New York: Prentice-Hall Press, 1986. $21.95. 316 pages.

Leo Scanlon's *80286 Assembly Language on MS-DOS Computers* can be used by both novice and experienced high-level language programmers who want to learn the Intel 80286 instruction set and assembly language programming on computers using it. The book starts with a brief introduction to binary and hexadecimal arithmetic, moves quickly to the fundamentals of assembly language programming, and then covers macro assemblers and how to build macro libraries.

The author uses plain English. His discussion includes useful programs and segments as illustrative tools. He also covers the SYMDEB debugger, the EDLIN line editor, and all the LINK utilities. He even throws in a discussion of how to program the Intel 80287 math coprocessor chip.

The program examples are so useful that you may well use them after you've finished the book. They include high-precision arithmetic programs, sorting programs, and code conversions.

Stevens, Al. *C Development Tools for the IBM PC*. New York: Prentice Hall Press, 1986. $22.95. 240 pages.

Not designed for the novice C programmer, *C Development Tools for the IBM PC* provides the experienced C programmer with a group of important development tools designed to assist in creating real applications. C language tools that support the development of on-line, interactive software systems on the PC are the province of Stevens's book. The book is really a tutorial workshop for tools such as file management, sorting, indexed data management (using B-trees), screen forms management, and menu management.

The author supplies the examples themselves, with thorough discussions of their purpose and use and the history of their development. It is full of technical details that will help you design and code programs of sound structure and proper syntax. If you are going to be doing

(continued)

How to put Ferrari speed and Mack Truck capacity into a PC without available slots.

Install the Dream Board in place of your multifunction card.

Chugging along in an overloaded, underpowered PC without room to expand? Need the increased power and memory of an AT™ without sacrificing your current software library or DP budget?

Problem solved: The Univation Dream Board!

- A PC accelerator board that triples the speed of your PC,
- *Plus* an EMS RAM board with up to 2 megabytes of high-speed RAM for your most complex Lotus™ spreadsheets and CAD operations,
- *Plus* a full-featured multifunction card with serial and parallel ports, clock/calendar, RAM Disk, Print Spooler, Disk Cache, and a speed selection switch that maintains 100% PC software compatibility.

All in ONE SLOT

That's the dream come true. If your PC, XT, Compaq™ Portable, or 100% compatible is full, simply replace your single application multifunction card with the Dream Board. You'll retain all your multifunction features while adding incredible speed and power. **With slots to spare!**

Planning to buy a new PC? The Dream Board is the ideal multifunction board. It provides the speed and power of an AT at a fraction of the cost.

Act While the Dream Board is fresh in your mind!

You need the power and the slots. The Dream Board is the answer! Call Univation today. Representatives are waiting to answer your questions and fill your order. Stop traveling in the slow lane. ***Move into the passing lane today!***

Outside California Call:

(800) 221-5842

UNIVATION

System Solutions for Business

1231 California Circle
Milpitas, CA 95035
(408) 263-1200

Inquiry 289

TRADEMARKS and REGISTERED TRADEMARKS: PC/AT—International Business Machines Corporation; Compaq—Compaq Computer; Lotus—Lotus Development Corporation; Dream Board—Univation.

serious (e.g., production) programming on the PC, XT, or AT in C, keep this handbook nearby.

Troy, Douglas A. ***Complete C Language Programming for the IBM PC***. Boston, MA: Little, Brown and Company, 1986. $24.95. 505 pages.

When you start to read *Complete C Language Programming for the IBM PC*, you may forget that you are reading a book about programming a PC. Not because there aren't copious references to the PC—there are. Not because there aren't examples of specific PC implementations—there are. Not because there aren't discussions of C language compilers for the PC—they are here also. It's the tone and orientation of the book that might throw you. There's no doubt that this book is written by a systems analyst who's had a lot of experience designing and coding C programs on larger computers. Discussions of program development, design, and coding are written in a way that bespeaks a large systems analysis orientation.

Is this approach effective? Surprisingly, yes. In fact, the book does an excellent job of teaching C programming on the PC. I would have liked to see specific discussions of C compilers incorporated into examples in the text (rather than relegated to an all-too-brief appendix), since the differences in performance and implementation are much greater among compilers than Troy admits. But that's a small complaint. This book will teach you the C language with the proper emphasis on good programming style, including top-down design and structured development methods. Troy uses two application programs, a calculator and a stock-portfolio system, to teach C. They are developed and refined with each chapter as more of the C language is revealed. *Complete C Language Programming for the IBM PC* assumes that you have solid programming experience on the PC with another language—Pascal would be helpful.

OTHER SOURCES

Several sources exist for readers to research for further information relating to specific topics on IBM PCs. Although the sources that follow are general and include other aspects and machine-specifics, they can be very instructive.

Bowker, R. R., and Company. ***Bowker's 1985 Complete Sourcebook of Personal Computing***. New York: R. R. Bowker Company, 1985. $19.95. 1050 pages..

Brand, Stewart, ed. ***The Whole Earth Software Catalog for 1986.*** Garden City, NY: Quantum Press/Doubleday, 1985. $17.50. 224 pages.

Char, Carlene, ed. ***Computer Book Review.*** Honolulu, HI: Maeventec, 1986. $20 for 12 issues.

Couger, J. Daniel. ***Computing Newsletter for Schools of Business: Bibliography***. Colorado Springs, CO: University of Colorado, 1986. $4 annual issue.

Sammett, Jean E., ed. ***ACM Guide to Computing Literature.*** New York: Association for Computing Machinery, 1986. $60 annual issue.

Sammett, Jean E., ed. ***Computing Reviews.*** New York: Association for Computing Machinery, 1986. $60 for 12 issues. ■

Donald Evan Crabb is the Director of Instructional Laboratories and a lecturer (The University of Chicago, Department of Computer Science, Ryerson Hall 260, 1100 East 58th St., Chicago, IL 60637). His articles and reviews have appeared in several magazines.

OFFICE AUTOMATION

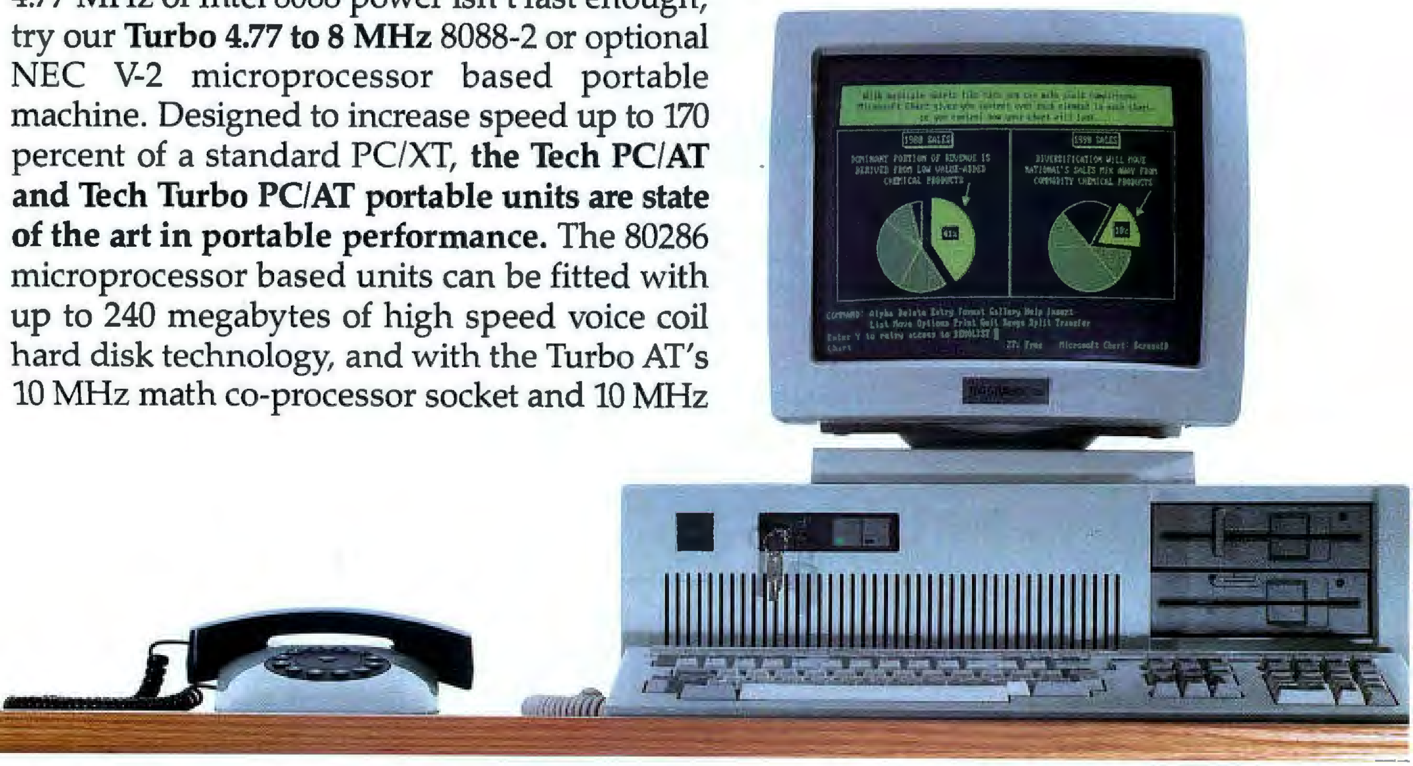
Tech PC, we're out to capture your microcomputer business. No one matches our diverse line of high performance products, or our comprehensive system of support, warranty, and third party maintenance service options. Tech PC is a large Southern California computer design and manufacturing firm with computer research and design specialists, production engineers, technical assistance staff, qualified sales personnel, and a strong management and investment team to provide you with state of the art products, state of the art reliability, and state of the art economy.

PORTABLES

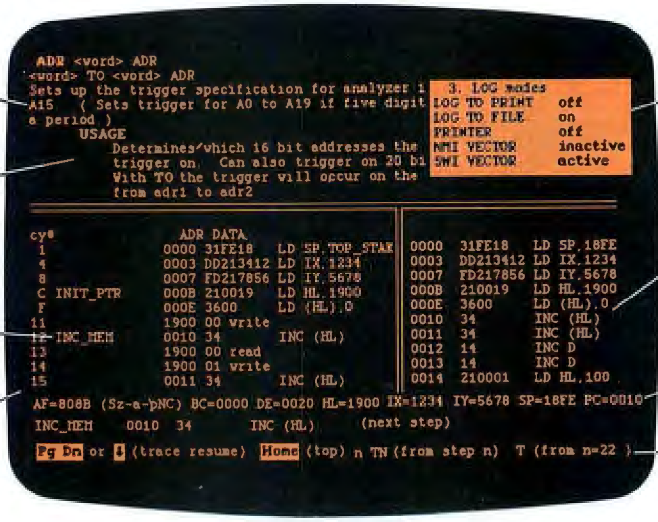
Tech PC Portable Computers come in four different base models to match your computing needs. Our entry level **Tech PC/XT** portable computer is designed to provide a non-obstructive profile, and has a capacity to hold two thirty megabyte hard disks, a high resolution monochrome monitor with a Hercules compatible graphics card to drive it with 800 by 400 lines resolution, and never before heard of 6 expansion slots in a portable computer. If 4.77 MHz of Intel 8088 power isn't fast enough, try our **Turbo 4.77 to 8 MHz** 8088-2 or optional NEC V-2 microprocessor based portable machine. Designed to increase speed up to 170 percent of a standard PC/XT, **the Tech PC/AT and Tech Turbo PC/AT portable units are state of the art in portable performance.** The 80286 microprocessor based units can be fitted with up to 240 megabytes of high speed voice coil hard disk technology, and with the Turbo AT's 10 MHz math co-processor socket and 10 MHz 80286 microprocessor the unit can manipulate large spreadsheets with unparalleled performance.

DESKTOPS

Tech PC Desktop Computers are designed for survival. The **XT's** leave your desktop looking comfortably empty with their slim non-obtrusive profile; The **AT's** give you a powerful presence with a clean lined sturdy chassis. Combined with a 14 inch, 800 by 400 line amber monochrome monitor or a 14 inch, 1280 by 800 black and white monochrome monitor you've got **State of the Art** staring straight back at you. If 8 MHz 80286 power isn't fast enough for you... try the Tech **Turbo PC/AT** with its 10MHz 80286 and 10 MHz 80287 math co-processor socket. The Tech Turbo PC/AT gives you the **strongest, most powerful** 80286 performance available from anyone, anywhere... available with up to 16 megabytes of memory and hard disks up to 1 gigabyte with our new optical storage disk, the Tech Personal



Microprocessor Development Dreams Come True!



On-Line Help menus, Command Glossary, and Word List.

Windows can be used to view source files, previous traces, and more.

Symbol translation or source code line display.

Screen displays scroll off into history buffer – can be viewed later.

Pop-up Mode Selection panel called by soft key.

Disassembly of code in memory can be compared with trace in adjacent window.

Symbolic debug register display.

Context sensitive prompt line.

NEW UniLab II™: FOUR INSTRUMENTS IN ONE!

Here are all the development tools you ever dreamed of integrated into one PC-controlled system:

- An Advanced 48-Channel Bus State Analyzer
- An 8/16-Bit Universal Emulator
- A Built-In EPROM Programmer
- An Input Stimulus Generator

The synergy of these instruments that were designed together to work together saves you time and money. All UniLab II commands and menus are seamlessly integrated into a single, super-efficient working environment.

An Integrated Software Environment, too!

Imagine being able to split your screen and look at real-time program traces and the source code that produced them at the same time! Then go to the On-Line Help or pop-up Mode Panels instantly.

If you see something on a trace that doesn't look like last time, you can hold it in one window while you scroll back through your previous displays.

If you set a breakpoint and single-step you can then go back to using the analyzer without missing a beat. You can even execute a DOS batch file from UniLab to edit, assemble, and link, then automatically load the new program and symbol table. UniLab uses the full power of the PC.

Find bugs fast with Hardware-assisted Debugging

The traditional way to look for bugs is to single-step through suspect parts of the code until you catch it in the act. This requires a lot of guessing and wasted time.

With UniLab's built-in analyzer you eliminate the guesswork. Just describe the bug symptom as a trigger, and let the UniLab hardware search for it as your program runs in real time. UniLab will show you a trace of the program steps leading up to the symptom, almost like magic.

A friendly user interface

UniLab lets you use commands or menus – or a mixture of both. An on-line manual, soft-key help screens, a glossary of commands and their parameters, with full-screen writeups are also at the ready.

Reconfigure for any 8 or 16-bit processor in seconds

Thanks to our unique approach to emulation, changes between processor types require only cable and diskette changes. At last count, we specifically support over 120 microprocessors.

Bonus! The built-in EPROM Programmer and Stimulus Generator are simply icing on the cake.

Affordable capability

How much does all this superior capability cost? A lot less than our less able competitors, and probably a lot less than you expect. Our products are sold with a Money Back Guarantee, and our crack team of Applications Engineers is standing by if you need help. Get the full story on the amazing UniLab II and how it can liberate your development projects, today.

ACTION COUPON

Send me info on UniLab II and your No-Risk 10-Day Evaluation!

Name: ______________ Title: ________

Company: ______________________

Address: ______________________

City: __________________________

State/ZIP: _____________________

Tel.: ______________ Ext. ______

UniLab II™
Universal Development Laboratory

Orion Instruments, Inc.

ORION

702 Marshall Street
Redwood City, California 94063

CALL TOLL FREE: 1-800-245-8500

In California (415) 361-8883

The IBM RT Personal Computer

Inside IBM's RISC *processor and memory management unit*

BY RICHARD O. SIMPSON

BACK IN 1975 the IBM Research Division began a project to investigate the utility of simple but very high speed computer hardware in a practical computer system. This system became known as the 801 Minicomputer—"801" because the project was located in Building 801 of the IBM T. J. Watson Research Center in Yorktown Heights, New York, and "minicomputer" because the machine was envisioned as being less expensive and physically much smaller than the mainframes of the time.

People who observed noticed that programs running on complex high-performance machines spent most of their time performing only very simple instructions, such as Load, Store, Branch On Condition, and Add. A good deal of the control logic of a complex machine exists for the instructions that require multiple cycles and for the complexities of memory operands that cross page boundaries and therefore could cause a page fault partway through an instruction's execution.

A stripped-down machine, one that is simpler than the IBM System/370, for example, could be made faster if it didn't have to deal with complex but statistically seldom-used operations. The goal of the 801 computer project was to produce a machine that executed programs at a sustained rate of one instruction per machine cycle. To achieve this rate, the instructions could not be very complex. All arithmetic and logical instructions were defined to operate on the contents of registers; there were no instructions like Add To Memory or Move Memory To Memory. Only simple addressing modes such as base plus displacement and base plus index were provided so that effective address calculation could be executed in a single cycle.

Operations that would inherently take multiple cycles—such as Multiply—were not provided. Instead, a single-cycle Multiply Step instruction was provided. In addition, provisions were made for very efficient linkage to a subroutine that would build the complex Multiply function out of the primitive Multiply Step instructions.

Optimizing the Compiler

The level of the instruction set of the 801 was comparable to the microcode level of more traditional machines. As a result, more 801 instructions were needed to perform a given function than, for example, IBM System/370 instructions. The size of the programs was not a problem. Memory costs were decreasing, and any product based on the 801 would more than likely be a virtual memory system. However, if an 801 program were a great deal longer than the equivalent program for a traditional machine, it would take longer to execute even at only one cycle per instruction.

The solution was a highly optimizing compiler to shorten an 801 program by eliminating redundant calculations, moving calculations out of

(continued)

Richard O. Simpson holds B.S. and M.E.E. degrees from Rice University and is a member of IBM's Advanced Engineering/Scientific Development organization in Austin, Texas. Since 1981 he has been involved in the architectural definition of ROMP and its memory management unit and more recently in the definition of the IBM RT Personal Computer. He can be contacted at IBM Corporation, Advanced Engineering/Scientific Development, 11400 Burnet Rd., Austin, TX 78758.

loops, performing calculations at compile time, and eliminating unnecessary loads and stores. The compiler and the language it compiled were named PL.8; the language is a system-programming dialect of PL/I, and the ".8" refers both to 801 and to the fact that the language was originally PL/I with those features not useful for system programming (about 20 percent) removed.

The design of the 801 went through several iterations. An early version had sixteen 24-bit registers and a mixture of 2-byte and 4-byte instructions. (The registers were 24 bits wide to simplify construction of a prototype from discrete logic.) The final design had thirty-two 32-bit registers and only 4-byte instructions. Restricting the instruction format to only one size simplified instruction decoding enough so that the machine's cycle time could be reduced.

The instruction set of 801 was specified in conjunction with the design of the PL.8 compiler. Instructions were included based on the needs of the compiler rather than the perceived needs of an assembly language hand-coder. The high-level language was to be the programming interface, through the PL.8 compiler or other compilers.

The designers provided many general-purpose registers because it is easy for a compiler to manage such a large set and it makes for more efficient code by eliminating most "spill" code requirements. The shift in design from 16 to 32 registers was based on compiler studies, which showed that the extra registers were worth the extra hardware cost.

MEMORY BANDWIDTH

Each 801 instruction was one word (32 bits) long, aligned on a word boundary in memory. The width of the data path to and from memory was also 32 bits. Since about 30 percent of the dynamic instruction mix was Load and Store instructions, the memory interface had to supply an average of 1.3 words per machine cycle. To maintain this rate the 801 had separate caches for instructions and data, each of which could handle one word per cycle.

A load from the cache took two cycles altogether. In a form of delayed branch called Branch With Execute, the instruction physically following the branch was always executed regardless of the branch condition. This gave the instruction cache an additional cycle in which to supply the branch target.

These efforts to keep throughput high were successful. Executing instructions on the 801 was somewhat like drinking from a fire hose. The sustained execution rate was around 15.1 million instructions per second at a

(continued)

Aztec C ... The Best C
Frees the genius in you

You've got a great idea . . .
. . . you're ready to write your programs.

You don't want to be sidetracked by all the paperwork. With Manx Aztec C and the ingenious **make** function, your creative processes won't get bogged down in program administration and housekeeping. Manx Aztec C has the most sophisticated, hardworking program administrator available to you. Once you've described your project, adding new features or enhancements is simple. You never have to concern yourself with the repetitive, tedious task of rebuilding your systems.

The development process moves quickly. Compiles, assemblies, link edits . . . all finish in record time.

Manx Aztec C is the fastest, most efficient C development system in the industry. Benchmarks show it . . . reviews commend it . . . users praise it.

You're ready to test the program. You're ahead of schedule. The Manx Aztec C Source Level Debugger shows you the exact C language statement giving you a problem. You fix the problem quickly . . . you're still ahead of schedule.

You've got some time for fine tuning. The Manx Aztec C Profiler examines your program, tells you where the slow spots are and validates your test procedure. A few changes and it's exactly what you wanted.

You've made it!

Aztec C is available for MS-DOS/PC DOS. Call for details on Macintosh, Amiga, Apple II, CP/M-80, CP/M-86, TRS-80, ROM and others.

To order, or, for information

Call Today

1-800-221-0440

In NJ or outside the USA call (201) 542-2121

30-day satisfaction guarantee. Special Discounts are available to professors, students, independent developers, and on a "trade-in" basis. Site licenses.

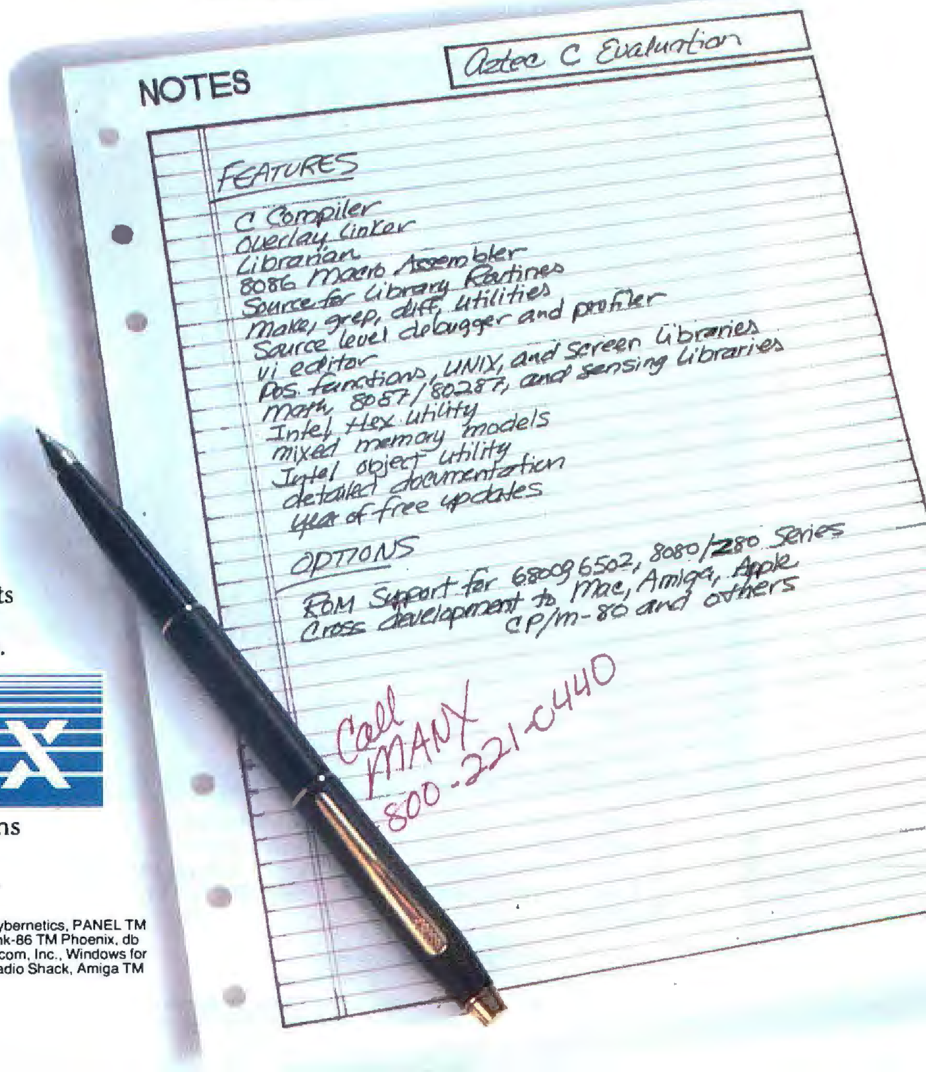
" . . . a superb linker, a profiler, an assembler, and a set of development utilities are only the beginning of this package . . . performed admirably on the benchmarks, with short compile times and the best link times in this review . . . includes the most professional make utility . . . documentation is clear and complete. There is no doubt that this is a valuable and powerful programming environment." **Computer Languages Feb. '86**

" . . . execution times are very good, close to the best on most tests . . . " **PC Tech Journal Jan. '86**

" Easily one of the fastest compilers overall . . . library provides a lot of flexibility . . . generates small .EXE files." **Dr. Dobbs Journal Aug. '85**

C'Prime (Compiler, Assembler, Linker)	**$ 99.**
Aztec C 86-d Developer's System	**$299.**
Aztec C 86-c Commercial System	**$499.**
PC ROM (8086, 68000, 8080, or 6502)	**$750.**

Third Party Software for Aztec C: HALO, PHACT, C-tree, PRE-C. Windows for C, PC-lint, PANEL, Greenleaf, db Vista, C-terp, Plink-86, FirsTime, C Util Lib, and others.



MANX

Manx Software Systems
One Industrial Way
Eatontown, NJ 07724

MS is a registered TM of Microsoft, Inc., CP/M TM DRI, HALO TM Media Cybernetics, PANEL TM Roundhill Computer Systems, Ltd., PHACT TM PHACT Assoc., PRE-C, Plink-86 TM Phoenix, db Vista TM Raima Corp., C-terp, PC-lint, TM Gimpel Software, C-tree TM Faircom, Inc., Windows for C TM Creative Solutions, Apple II, Macintosh TM Apple, Inc., TRS-80 TM Radio Shack, Amiga TM Commodore Int'l.

cycle time of 63 nanoseconds, or about 1.1 cycles per instruction.

Other research groups outside IBM reached the same conclusions regarding rapid execution of simple instructions. The term RISC (reduced instruction set computer) was coined by a group at Berkeley that built a machine with a very simple instruction set. Another group at Stanford devised a machine (MIPS) in which simple operations were executed in parallel in a pipeline, with the compiler responsible for keeping conflicting operations from being scheduled simultaneously.

EVOLUTION OF ROMP FROM THE 801

After the definition of the 801, a project began in the IBM Office Products Division. Its goal was to apply what had been learned on the 801 project to a new microprocessor for use in IBM products. This processor became ROMP, an acronym for Research/OPD Micro Processor.

Most project code names in IBM are changed each year or two, but ROMP was appealing enough that it stuck with the project through the announcement and release of the IBM RT Personal Computer in early 1986. [*Editor's note: In the interest of consistency, the term* RT *processor is used throughout this article to describe the* RISC *chip. The acronym* ROMP *is used only where reference is made to the forerunner of the* RT *processor, the* Research OPD/Micro Processor.] As a microprocessor for a product rather than a minicomputer in a research project, from the start ROMP had different goals from the 801. The 801 strove to maintain the maximum execution speed consistent with its single-cycle-per-instruction design philosophy regardless of hardware cost. Thus the 801 had simpler hardware than, for example, the IBM System/370, but it still had two high-speed caches and multiple internal buses. These were not appropriate for the RT processor, which was to be a single-chip microprocessor.

The cost/performance ratio was important for the RT processor, and two caches were out of the question for a single-chip computer under design in the late 1970s. The lack of cache could not be made up by simply providing very fast memory, again for cost reasons. The RT processor design evolved from a complex set of tradeoffs involving hardware cost, machine cycle time, overall throughput, and the amount of logic that could be fitted onto one VLSI chip.

COMMON PHILOSOPHY

The RT processor and the 801 have the same architectural foundation. Both are true general-register ma-

(continued)

chines in which all computation is done on quantities in registers, and no distinction is made between computation on data and addressing arithmetic. Any register can hold a data quantity or a base address. In both machines the movement of data to and from memory is separated from computation by the use of explicit Load and Store instructions. Most instructions execute in a single cycle. The memory addressing model for each is a simple, linear, byte-space addressed by a 32-bit quantity.

DESIGN DIFFERENCES

The two machines could not be the same, however, because of the tradeoffs discussed above. The most pervasive change was the removal of the two caches. The lack of cache affected everything from the instruction set to the number of registers to page-fault handling.

All 801 instructions were one word long, which made for simple instruction decoding. The 32-bit format allowed room for multiple fields used to hold register numbers (each 5 bits long). The typical instruction specified two source registers and a destination register.

These three-address instructions made for efficient code because it was not necessary to copy one register to another to avoid destruction of one of the operands. The use of all 4-byte instructions means that the CPU's memory bandwidth requirements are greater than one word per cycle. Thus if the RT processor, without caches and with a memory bus width limited to 4 bytes, had implemented the 801 instruction set directly, it could never have achieved a rate of one instruction per cycle.

To reduce the RT processor's memory bandwidth requirements, a 2-byte instruction format was introduced. The instruction set was a mixture of 2-byte and 4-byte instructions with the short format predominating in the dynamic mix. The average instruction length is about 2.5 bytes (versus exactly 4 for the 801). Because about 30 percent of instructions are for loading or storing, the overall bandwidth requirement is about 3.7 bytes per instruction. A memory subsystem that could supply 4 bytes per cycle could keep up with the RT processor's bandwidth requirement.

The 2-byte format couldn't accommodate three 5-bit register fields plus a reasonable size op code, so the instructions were made two-address and the register field size was cut from 5 bits to 4. Consequently, the RT processor uses only 16 general registers (the 801 uses 32).

PRACTICAL COMPLEXITY

To make subroutine linkage as efficient as possible for the RT processor,

(continued)

instructions that load and store a series of registers were introduced. These were not needed on the 801, where a string of individual Load or Store instructions would execute at maximum speed, but they are handy on the RT processor because they allow the entire memory bandwidth to be used for transferring data.

These instructions are complex in comparison to the other single-cycle instructions. They require multiple cycles and must have the ability to handle cases in which a page fault occurs in the middle of the sequence of loads or stores. To provide for this type of instruction and allow other instructions to take multiple cycles (such as the infrequently used Divide Step), the RT processor was designed as a microcoded engine. (By contrast, the 801 was hard-wired.)

A microprogram is executed for each RT processor instruction, but for most of the instructions the microprogram is exactly one step long (most instructions take only one cycle). The microprograms for the more complex instructions take as many cycles as necessary. Microcode is also used for such housekeeping tasks as CPU checkout and register initialization after power-on reset.

RT Processor Programming Model

The RT processor registers visible to the programmer are shown in figure 1. There are sixteen 32-bit general-purpose registers, numbered 0 through 15. Bits are numbered left to right (high-order to low-order), 0 to 31. This left-to-right numbering scheme is used consistently throughout the RT processor architecture, including memory addressing.

All the general registers are equivalent and can be used for any purpose, addressing or data, with the exception of register 0. When register 0 is specified as a base register, the value 0 is used as the base address rather than the contents of register 0. This provides a means of addressing absolute locations in low memory without requiring that a base register be loaded or that some special direct-addressing mode be provided. This use of register 0 is exactly like that of the IBM System/370.

Each register holds a 32-bit word. Arithmetic is performed using two's-complement notation on the full 32-bit register contents. There are provisions for dealing with short integers (halfwords) and single bytes, both in memory and in the general registers. Some instructions treat the upper half (bits 0 through 15) or lower half (bits 16 through 31) of a register individually, and some deal with the individual bytes within a register (bits 0 through 7, 8 through 15, 16 through 23, and 24 through 31).

In addition to the general registers, the RT processor architecture provides for 16 system control registers (SCRs). As the name implies, most of these are used for controlling the state of the RT processor, but some are directly useful to the programmer. SCRs 6 through 8 are used to control the on-chip interval timer. SCR 10 is the Multiplier/Quotient (MQ) register and is the implied third operand of the Multiply Step and Divide Step instructions. SCR 11 holds Machine Check Status, which reports hardware malfunctions, and Program Check Status, which reports exceptions such as traps, page faults, and illegal operations.

The Interrupt Request Buffer, SCR 12, controls interrupts generated directly by software. SCR 13 is the Instruction Address register (IAR), or program counter. Machine state information, such as virtual address translation on/off, privileged versus unprivileged state, and interrupts enabled/disabled, is kept in SCR 14, the Interrupt Control Status register.

The Condition Status (CS) register (SCR 15) reflects the results of arithmetic operations, comparisons, and other tests. Not all of the 16 SCRs in the architectural scheme are implemented, and of those that are, only SCRs 10 (the MQ) and 15 (the Condition Status bits) are of interest to the

(continued)

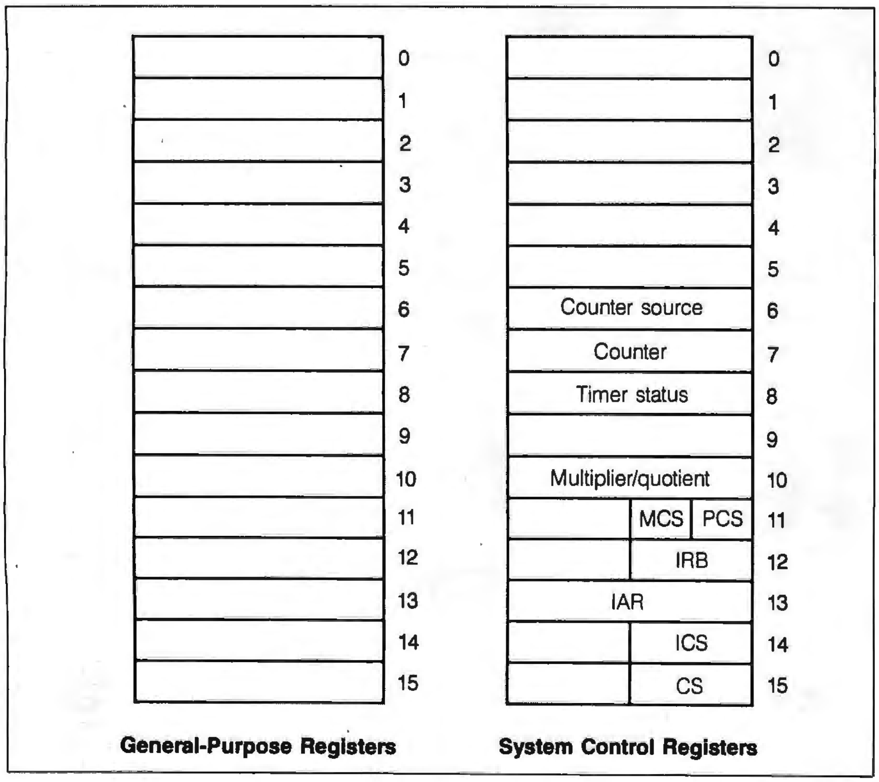


Figure 1: *A model of the RT processor registers. All figures in this article are © copyright International Business Machines Corporation, 1986.*

C FEVER *catch it!*

It's becoming an epidemic . . . everyone is switching to C! First there were a few hackers, then came the college students, next the major software houses, and now the rest of the programming world. Programmers everywhere are infected with the desire for SPEED, POWER, and PORTABILITY.

It's time to face the inevitable. You're going to catch the fever too! When you do, give us a call. We've got the best cure—an illustrated guide to the C language, plus a complete program development system. Everything you need to master the C programming language . . . all at a price that's less than the cost of a book!

But don't let this price fool you. Our system is powerful; it compiles twice as fast as the others, is completely standard, and it's very easy to use. Most C compilers are designed for wizards. We have designed ours for you!

What do you get for a mere $39.95?

- **A 450 Page book filled with sample programs, plus . . .**
- **A fast, standard, full featured C compiler** that supports all data types and the latest features like bit fields, enumerations, structure assignment, and passing/returning structures.
- **A fast linker** that loads separately compiled files, searches libraries, and builds an executable program.
- **An extensive library of more than 170 functions** (including the standard C functions and the computer specific functions that provide direct access to the operating system and BIOS).
- **Tools that allow you to optimize** your programs for minimal space or maximum speed.

Incredible Value
AT ONLY **$39.95** With 30 Day Money-Back Guarantee

Operators are standing by . . . Please use this Number for ORDERS ONLY!

CALL TOLL FREE FOR RUSH ORDER DELIVERY!

1-800-523-9520

IN TEXAS, PLEASE CALL TOLL FREE 1-800-622-4070

For Technical Support Please call 1-214-783-6001

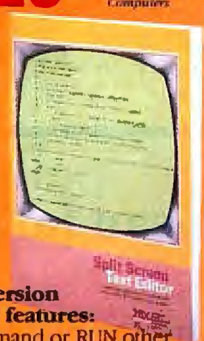
MIX Software, Inc. / 2116 E. Arapaho / Suite 363 / Richardson, Texas 75081

Or contact our Worldwide Distributors direct in:

Canada:	Saraguay Software	1-416-923-1500	Switzerland:	DMB Communication	CH-1-825-53-29
Australia:	Techflow	047-586924	France:	Info/Tech	1-43-44-06-48

Split Screen Text Editor

an Incredible Value AT ONLY $29.95 *(Not Copy Protected!) Works on all MSDOS/PCDOS and CP/M Z80 Computers*



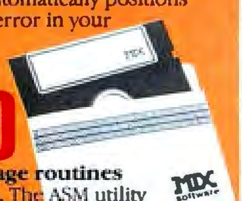
Our high powered editor is great for editing high level languages. **It works just like Micropro's Wordstar,**® but macros allow you to create your own custom editor, and the split-screen feature lets you edit two files at the same time.

The MSDOS/PCDOS version is loaded with special features:

- Execute any DOS command or RUN other programs from the editor.
- Quickly edit files as large as 300,000 characters.
- Compile MIX C programs directly from memory. The editor automatically positions the cursor to the first error in your program.

ASM UTILITY $10

an Incredible Value AT ONLY



Call assembly language routines from your C programs. The ASM utility works with Microsoft's MASM or M80 assembler. Macros make it easy! Works just as if you were calling a C function, and you can even call C functions from assembly language. Lots of useful assembly language functions are included as examples.

Inquiry 8

RUSH REPLY ORDER FORM!

Description	Quantity	PRICE	Total Order
Split-Screen Text Editor	______	$29.95	$______
C Compiler	______	$39.95	$______
C and Text Editor (Special)	______	$54.95	$______
ASM Utility	______	$10.00	$______
Texas Residents Add 6.125% Sales TAX			$
Shipping Charges (See at Right)			$
TOTAL OF YOUR ORDER:			$

Please check method of payment:
☐ Check ☐ Money Order ☐ MasterCard/VISA

Your Card #: ______________

Expires ______________

Shipping Charges: (No charge for ASM Utility)
In the U.S.A.: Add $5.00 per Order.
In CANADA: Add $10.00 per Order.
OVERSEAS: Add $10.00 per Text Editor. Add $20.00 per C Compiler. Add $30.00 for combined C Compiler and Text Editor.

Operating System: ***(Check one)***
☐ CP/M Z80 ☐ MSDOS/PCDOS

Specify Your Computer Name ______________

Specify Disk Format ______________

NAME ______________

Telephone A/C (______) ______________

Street ______________

City ______________

State ______________

Country ______________ ZIP ______________

MIX software
2116 East Arapaho
Suite 363
Richardson, Texas, 75081

Ask about our Volume Discounts!
Call 1-214-783-6001

B

application programmer. The value in the program counter (IAR) is not normally needed directly. Most branches are implicit relative to the IAR, and the value in the IAR is supplied in a general register automatically by the Branch And Link instructions.

MACHINE STATES

The RT processor has both a privileged state and an unprivileged state. All instructions are valid in privileged state. Instructions that deal with protected SCRs or that can directly modify the machine state or the state of the virtual addressing or memory protection hardware are not allowed in unprivileged state. In either privileged or unprivileged state, the virtual addressing and memory protection mechanism can be enabled (virtual mode) or disabled (real addressing mode).

Normally, the operating system runs all application code in unprivileged state with virtual addressing and protection turned on, while parts of the operating system itself run in real addressing mode.

When an interrupt occurs, the current state of the processor (the contents of SCRs 13, 14, and 15) is saved at a specified address in low memory. The saved data is referred to as PS, for processor status. The address at

(continued)

GET SERIOUS

...ABOUT ANALYZING YOUR DATA.

You might be spreading your spreadsheet a little too thin. Or maybe you're starting from scratch. But if you're serious about data analysis, you're ready for SPSS/PC+™—a full software family that brings you six high-powered ways to complete any data analysis task.

Enter it. SPSS/PC+ Data Entry™ takes the effort out of entering and correcting data.

Analyze it. The SPSS/PC+ Base Package provides a powerful array of statistical and reporting procedures.

Examine it. SPSS/PC+ Advanced Statistics™ lets you get more serious with your data.

Table it. SPSS/PC+ Tables™ produces presentation-ready tables instantly.

Chart it. SPSS/PC+ Graphics™ featuring Microsoft® Chart creates show-stopping graphs and charts.

Map it. SPSS/PC+ Mapping™ featuring MAP-MASTER™—our latest option—creates maps where vast amounts of data can be summarized and presented in one, simple picture.

SPSS/PC+ products are being put to productive use by serious fact finders in business, government and education. For countless purposes such as market research. Wage and salary studies. Survey analysis. And quality control. Plus each product is superbly documented and supported by SPSS Inc., a leader in statistical software for nearly 20 years.

So if you're serious about data analysis, step up to SPSS/PC+. For details, contact our Marketing Department. **CALL 1/312/329-3630**

SPSS inc.

SPSS Inc. • 444 North Michigan Avenue, Suite 3000 • Chicago, Illinois 60611
In Europe: SPSS Europe B.V. • P.O. Box 115 • 4200 AC Gorinchem, The Netherlands • Telephone: +31183036711 • TWX: 21019

SPSS/PC+ runs on IBM PC/XT/AT's with hard disk. Contact SPSS Inc. for compatible microcomputers. SPSS/PC+, SPSS/PC+ Data Entry, SPSS/PC+ Advanced Statistics, SPSS/PC+ Tables, SPSS/PC+ Graphics and SPSS/PC+ Mapping are trademarks of SPSS Inc. for its proprietary computer software. Chart and Microsoft are trademarks of Microsoft Corporation. MAP-MASTER is a trademark of Decision Resources.

Inquiry 271

Table 1: *The RT processor's complete instruction set.*

LOADS, STORES

Mnemonic	Bytes	Cycles	Instruction
L	4	1–6	Load
LS	2	1–6	Load Short
LH	4	1–6	Load Half
LHS	2	1–6	Load Half Short
LHA	4	1–6	Load Half Algebraic
LHAS	2	1–6	Load Half Algebraic Short
LC	4	1–6	Load Character
LCS	2	1–6	Load Character Short
LM	4		Load Multiple
ST	4	2–6	Store
STS	2	2–6	Store Short
STH	4	2–6	Store Half
STHS	2	2–6	Store Half Short
STC	4	2–6	Store Character
STCS	2	2–6	Store Character Short
STM	4		Store Multiple
TSH	4	1–6	Test And Set Half

COMPUTATION, REGISTER TRANSFER

Mnemonic	Bytes	Cycles	Instruction
ABS	2*	2	Absolute
A	2	1	Add
AI	4	1	Add Immediate
AIS	2	1	Add Immediate Short
AE	2	1	Add Extended
AEI	4	1	Add Extended Immediate
CAL	4	1	Compute Address Lower
CAL16	4	1	Compute Address Lower, 16-bit
CAS	2	1	Compute Address Short
CAU	4	1	Compute Address Upper
CA16	2	1	Compute Address, 16-bit
C	2	1	Compare
CI	4	1	Compare Immediate
CIS	2	1	Compare Immediate Short
CL	2	1	Compare Logical
CLI	4	1	Compare Logical Immediate
CLRBL	2	1	Clear Bit Lower
CLRBU	2	1	Clear Bit Upper
CLZ	2	1	Count Leading Zeros
D	2	3	Divide Step
DEC	2	1	Decrement
EXTS	2	1	Extend Sign
INC	2	1	Increment
LIS	2	1	Load Immediate Short
M	2	4	Multiply Step
MC03	2	1	Move Character Zero From Three
MC13	2	1	Move Character One From Three
MC23	2	1	Move Character Two From Three
MC33	2	1	Move Character Three From Three
MC30	2	1	Move Character Three From Zero
MC31	2	1	Move Character Three From One
MC32	2	1	Move Character Three From Two

(continued)

which the PS is stored depends on the type (machine check, program check), or I/O level number, of the interrupt. A new PS is then automatically loaded from a location specific to the type or level of interrupt, and processing continues. This new PS specifies (presumably) privileged state, real addressing mode, interrupts disabled, and the address of the interrupt handling routine.

At the conclusion of an interrupt handler, the interrupted program is resumed via a Load Program Status instruction that reloads the saved PS and thus restores the original state. General registers and SCRs other than 13 through 15 are not saved automatically; such saving and restoring must be done by the interrupt handler.

INSTRUCTION SET

The term RISC applies to the RT processor more in the concept of simple one-cycle instructions than in the actual size of the instruction set. The RT processor has 118 instructions (the Berkeley RISC-I had 31). The concept is that if a machine has a certain data flow and an ALU that can perform certain functions, then all the operations those resources can perform that are useful to the compiler are exposed as instructions. This results in some instructions that might not be the first choice in a machine designed for assembly language programming (example: shift left paired immediate plus 16), but the functions they perform are called for by the compiler often enough to warrant their inclusion.

The RT processor's 118 instructions can be divided into six major categories. There are 17 load/store variations, 73 computation and register transfer instructions, 16 branching instructions, 3 traps, 7 system control instructions, and 2 I/O instructions. The complete instruction set is listed in table 1.

There are load instructions for fullwords (32 bits), halfwords (16 bits, padded on the left with 0s to 32 bits when loaded), algebraic halfwords (16 bits, sign-extended to 32 bits when

(continued)

Mnemonic	Bytes	Cycles	Instruction
MFTB	2	1	Move From Test Bit
MFTBIL	2	1	Move From Test Bit Immediate Lower
MFTBIU	2	1	Move From Test Bit Immediate Upper
MTTB	2	1	Move To Test Bit
MTTBIL	2	1	Move To Test Bit Immediate Lower
MTTBIU	2	1	Move To Test Bit Immediate Upper
N	2	1	AND
NILO	4	1	AND Immediate Lower Extended Ones
NILZ	4	1	AND Immediate Lower Extended Zeros
NIUO	4	1	AND Immediate Upper Extended Ones
NIUZ	4	1	AND Immediate Upper Extended Zeros
O	2	1	OR
OIL	4	1	OR Immediate Lower
OIU	4	1	OR Immediate Upper
ONEC	2	1	One's Complement
S	2	1	Subtract
SIS	2	1	Subtract Immediate Short
SE	2	1	Subtract Extended
SF	2	1	Subtract From
SFI	4	1	Subtract From Immediate
SETBL	2	1	Set Bit Lower
SETBU	2	1	Set Bit Upper
SL	2	1	Shift Left
SLI	2	1	Shift Left Immediate
SLI16	2	1	Shift Left Immediate Plus 16
SLP	2	1	Shift Left Paired
SLPI	2	1	Shift Left Paired Immediate
SLPI16	2	1	Shift Left Paired Immediate Plus 16
SR	2	1	Shift Right
SRI	2	1	Shift Right Immediate
SRI16	2	1	Shift Right Immediate Plus 16
SRP	2	1	Shift Right Paired
SRPI	2	1	Shift Right Paired Immediate
SRPI16	2	1	Shift Right Paired Immediate Plus 16
SAR	2	1	Shift Algebraic Right
SARI	2	1	Shift Algebraic Right Immediate
SARI16	2	1	Shift Algebraic Right Immediate Plus 16
TWOC	2	1	Two's Complement
X	2	1	Exclusive-OR
XIL	4	1	Exclusive-OR Immediate Lower
XIU	4	1	Exclusive-OR Immediate Upper

BRANCHING

Mnemonic	Bytes	Cycles	Instruction
BALA	4	5–7	Branch And Link Absolute
BALAX	4	5–7	Branch And Link Absolute With Execute
BALI	4	5–7	Branch And Link Immediate
BALIX	4	5–7	Branch And Link Immediate With Execute
BALR	2	5–7	Branch And Link Register
BALRX	2	5–7	Branch And Link Register With Execute
BB	4	*	Branch On Condition Bit Immediate
BBX	4	*	Branch On Condition Bit Immediate With Execute
BNB	4	*	Branch On Not Condition Bit Immediate
BNBX	4	*	Branch On Not Condition Bit Immediate With Execute
BBR	2	*	Branch On Condition Bit Register
BBRX	2	*	Branch On Condition Bit Register With Execute

(continued)

loaded), and bytes (8 bits, padded on the left with 0s to 32 bits when loaded). A Load Multiple instruction loads from 1 to 16 registers from consecutive words in memory. For each of these there is a corresponding store instruction.

Simple base-register plus constant-displacement addressing modes are provided. More complex modes (base plus index register, for example) are constructed by the compiler from the primitive instructions. The 73 computational instructions include all of the ordinary integer arithmetic, logical, and shift instructions, plus a group of address computation and register transfer instructions.

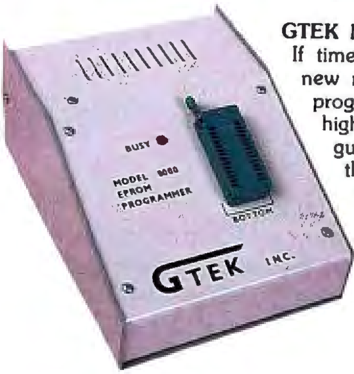
In general, the arithmetic instructions set the Condition Status bits according to their result, while the address computation instructions do not. In some cases the only difference between an arithmetic instruction (Add, Add Immediate) and an address computation instruction (Compute Address, Increment) is whether the Condition Status is affected. This gives the compiler more freedom to rearrange and optimize computations.

There are two classes of conditional branches: jumps, which are 2-byte branches, relative to the contents of the IAR; and branches, which are generally 4 bytes and can be either branch absolute, branch relative (to the IAR), or branch register (via an address in a register). The branches have a traditional form and a delayed-branch form, which is referred to as Branch With Execute. Conditional branches, or jumps, test the value of one of the bits in the Condition Status register. Unconditional branches are done by testing a particular bit in the CS register that is permanently zero. Subroutines are called using Branch And Link instructions that place the address of the instruction following the branch in a general register. Subroutine returns are via a Branch Register instruction that branches to the address saved by the call.

The trap instructions provide an efficient way to test array indexes, case-statement values, and the like, against

(continued)

THIRD GENERATION PROGRAMMERS FROM GTEK



GTEK MODEL 9000 (E)(E)PROM/MPU PROGRAMMER
If time is money, then let us save some for you. The new model 9000, using its quick pulse algorithms, can program a 2764 in 10 seconds. The 9000 offers higher performance than the competition has even begun to think about. Baud rates to 57,600. Supports thru 1024K standard, 8 meg wordwide parts with adapter, Cypress proms, mpu's. NO personality modules. As with our other programmers, RS-232, ASCII data formats, and flexible handshaking make the model 9000 compatible with virtually any computer. Introductory price **$749.**

MODEL 7228 (E)EPROM/MPU PROGRAMMER The standard by which other other programmers are judged. With thousands in the field, the time proven 7228 has become our most popular programmer. Intelligent algorithms standard. Programs a 2764 in one minute. Supports devices thru 512K, mpu's, eeproms, cmos. NO personality modules. Price **$599**

MODEL 7956 GANG (E)EPROM/MPU PROGRAMMER
The 7956 can program eight 2764's in one minute using intelligent algorithms. Standard support thru 512K, eeproms, cmos, mpu's. A must for production emvironments. Available in stand alone only configuration for **$979**
or with RS-232 interface for **$1099**

MODEL 7128 (E)EPROM/MPU PROGRAMMER The programmer that won't die. Thousands manufactured since 1982. Supports thru 256K. PRICE REDUCED to **$389 including free PGX communications software, a $95 value.**

MODEL 705 Motorola 68705 family programmer. Transfers object code to 28 and 40 pin mpu's. Single key stroke operation. **$299**

MODEL 7324 PAL PROGRAMMER The 7324 has a built-in compiler. It supports 20 and 24 pin pals by MMI, NATIONAL, and TI, including the new shared product parts 20S10, 20RS10, 20RS8, 20RS4. It operates stand alone or via RS-232 with PALX communications package. Functionally tests parts after programming and securing. External compilers are supported thru JEDEC and AHS object formats. **$1499**

MODEL 7322 Same as 7324 but no stand alone capability. **$1249**

UTILITY PACKAGES

PGMX High Speed Communications Package. For IBM PC, AT and compatibles. Bi-directional data rates to a scorching 57600 bps. Provides flexible manipulation of Intel Hex files, both 8 and 16 bit formats, and binary files. Use with model 9000. .... **$149**

PGX Utility Package. For PCDOS, CPM, TRSDOS, ISIS, MSDOS. Use with 9000, 7956, 7228, 7128. .... **$95**

PALX Utility Package. Provides communication, download of Palasm source, JEDEC and AHS object files to models 7324, 7322, 7316 pal programmers. **$95**

CROSS ASSEMBLERS and SIMULATORS

You name the cpu, we've got the assembler. Simulators allow target software development and testing on your computer. Versions available for PCDOS, MSDOS, CPM. . Start at **$250**

ERASERS—We stock both Ultra Violet Products and Spectronics for your selection.

MODEL	CHIPS	TIMER	LAMP	PRICE
DE4	8	N	Tube	**$80**
PE14T	9	Y	Tube	**$139**
C25	25	Y	Grid	**$350**
C50	50	Y	Grid	**$600**

ACCESSORIES

CABLES—serial **$30**, parallel **$30**, custom Call For Quote
MODEL 481/482/483 8048 family adapters **$98**
MODEL 511/512 8051 family adapter **$174**
MODEL 755/756 8755 adapter . . . **$135**

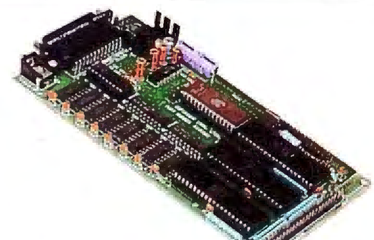
SPECIAL REQUIREMENTS? Call Us For A Quote On Custom OEM Programmers.

Call GTEK's CPM Hotline
1-601-467-9019

If you are a CPM user who wants to convert to MSDOS or PCDOS or a PC user who would like access to the thousands of CPM programs in existance, here is the perfect solution to the problem: CPMulator from GTEK.
CPMulator is the only field proven emulator which emulates the complete Z-80 instruction set. Terminal attribute emulation is Televideo, Lear Siegler, or the ANSI standard. CPM .COM programs reside under PCDOS and are executable under PCDOS as well as 8086 programs.
Comes complete with COPYCPM, a disk conversion utility which allows transferring data and programs to and from 54 different CPM disk formats. Free CPM utility programs are also included.
For speed use the 8080 CPMulator along with the NEC V20 or V30 chip.

PRICE REDUCED

Z80 CPMulator, DR5PC-EMZ80 . . . **$ 99.00**
8080 CPMulator, DR5PC-EMV205 **$ 99.00***
Z80, 8080 CPMulator, DR5PC-EMZV205 .... **$149.00***
*Requires NEC V20 or V30.
NEC V20 (5Mhz) .... **$ 12.00**



Single Board Computer Model 2010

- 64 to 256K bytes data or program ram.
- Use 2764 or 27512 eprom or eeprom program memory.
- Uses single ended 9V @ 500 ma supply.
- Use 8031, 8032, 8751, 8752 mpu.
- 40 bit programmable I/O lines.
- Expansion interface connector.
- Baud rates to 57,600 baud.
- Built in 5 volt regulator.
- Built in rom monitor.
- Small — 3.5 x 7 inches.
- 8052AH basic option.

ONLY $299.00 WITH MANUAL

MODEL 8014 PRINTER BUFFER/SWITCH
0K is **$99**, 128K is **$339**, and 256K is **$399**

GTEK, INC.
Development Hardware & Software
P.O. Box 1346, Bay St. Louis, MS 39520 U.S.A.
(601) 467-8048; TELEX 315814 (GTEK UD)

GTEK, PALASM, CPM, MS-DOS, PC-DOS, ISIS, TRSDOS, & CPEmulator are registered trademarks.

Mnemonic	Bytes	Cycles	Instruction
BNBR	2	*	Branch On Not Condition Bit Register
BNBRX	2	*	Branch On Not Condition Bit Register With Execute
JB	2	*	Jump On Condition Bit
JNB	2	*	Jump On Not Condition Bit

* Conditional branches and jumps take
1 cycle if unsuccessful (fall through)
5–7 cycles if successful

TRAPS

Mnemonic	Bytes	Cycles	Instruction
TGTE	2	2	Trap If Register Greater Than Or Equal
TLT	2	2	Trap If Register Less Than
TI	4	2	Trap On Condition Immediate

SYSTEM CONTROL

Mnemonic	Bytes	Cycles	Instruction
LPS	4	12–16	Load Program Status
CLRSB	2	4	Clear SCR Bit
MFS	2	2	Move From SCR
MTS	2	3	Move To SCR
SETSB	2	4	Set SCR Bit
SVC	2	16–20	Supervisor Call
WAIT	2		Wait

INPUT/OUTPUT

Mnemonic	Bytes	Cycles	Instruction
IOR	4	1	Input/Output Read
IOW	4	2	Input/Output Write

upper and lower bounds. Such instructions perform a logical (unsigned) comparison of a register against an immediate value, or against another register, and cause an interrupt if the specified trap condition "springs."

For arrays with a lower bound of zero, a single trap instruction will catch positive indexes greater than the upper bound and all negative indexes (negative values appear to be large positive values in unsigned comparisons). For nonzero lower bounds an additional add or subtract instruction must precede the trap. The compiler subjects such trap code to the same optimizations as all the rest of a program's computations. As a result, bounds checking for a language like Pascal can be done very efficiently.

The system control instructions perform low-level hardware operations, such as resetting interrupt requests, manipulating the contents of the privileged SCRs, and loading new program status. One system control instruction, Supervisor Call (SVC), is provided for use by unprivileged state code, to switch to privileged state in a controlled manner. SVC causes an interrupt and passes a 16-bit code to the Supervisor Call Interrupt Handler; the operating system then interprets the code as a request for some specified service, performs the requested action, and resumes the program via Load PS.

There are two I/O instructions that transfer 32-bit values between general registers and the I/O bus, but most I/O is memory-mapped and is done using the normal Load and Store instructions. Control of the address translation hardware in the MMU is done by privileged state programs using the I/O instructions directly.

Basic Data Types

The fundamental unit of memory is the 32-bit word, and all memory operations involve 32-bit words. The words are further subdivided into halfwords and bytes. Memory is addressed by byte address, and words and halfwords are required to be on appropriate boundaries. That is, a word address must be a multiple of four, and a halfword address must be a multiple of two. Such alignment requirements make for faster execution and simplify the handling of page faults.

Loading a 4-byte word that crosses a word boundary would require two memory operations rather than one, while the corresponding Store would require two read-modify-write cycles rather than one simple write. If such a misaligned word also crosses a page boundary, then it's possible that a page fault or protection exception could occur partway through a memory operation. Such complexities are handled for Load Multiple and Store Multiple, but they are not warranted for the very heavily used Load and Store instructions.

Signed two's-complement integers can occupy words or halfwords. These halfwords will be sign-extended to fullword quantities when loaded into a general register. Individual bytes can also be loaded and stored but when loaded are extended on the left with 24 zero bits. Thus bytes can naturally hold small unsigned integers as well as characters.

Individual bits cannot be addressed in memory directly but can be manipulated in the general registers once a byte, halfword, or fullword has been loaded. In addition to the normal complement of logical and shift in-

(continued)

TANDY®... Better Again™

The New Tandy 3000 HL

Four times the power of the IBM® PC/XT ...for less.

The Tandy 3000 HL isn't just compatible with IBM's PC/XT—it's better. Better because of its new 16-bit microprocessor that lets you process twice the data of IBM's older, 8-bit design. Better because it has nearly twice the clock speed of the PC/XT.

Better because you get more for less: only $1699.

The Tandy 3000 HL's 512K RAM lets you run the software that you've grown accustomed to—at a much higher speed for greater efficiency. And you can expand with hard disk drives, modems, and more.

The high power and low price of the Tandy 3000 HL make it perfect for offices requiring several workstations. And the ViaNet local area network can bring your 3000 HLs together with your existing MS-DOS based computers. Users and departments can be more efficient.

Plus, the American-made Tandy 3000 HL is backed with total support, including our leasing, service and training plans.

Get the facts today, and you'll agree: the Tandy 3000 HL (25-4070) is your basic building block for total office automation.

Available at over 1200 Radio Shack Computer Centers and at participating Radio Shack stores and dealers.

Radio Shack® COMPUTER CENTERS

A DIVISION OF TANDY CORPORATION

Send me an RSC-17 Computer Catalog.

Mail To: Radio Shack, Dept. 87-A-976
300 One Tandy Center, Fort Worth, TX 76102

Name ____________________

Company ____________________

Address ____________________

City ____________________

State ____________ ZIP ____________

Phone ____________________

Price applies at Radio Shack Computer Centers and participating stores and dealers. Monitor sold separately. IBM/Registered TM International Business Machines Corp. MS-DOS / Registered TM Microsoft Corp. ViaNet/TM ViaNetix, Inc.

structions, there are two instructions (Move To Test Bit, Move From Test Bit) that allow individual bits to be moved between registers or into the Condition Status register and thus affect a conditional branch directly.

Quantities larger than fullwords, such as character strings, are not operated on directly by the RT processor hardware but are treated as sequences of words or bytes. Character strings do not normally have any boundary alignment requirements.

Floating-point quantities handled by the IBM RT Personal Computer's floating-point unit have alignment restrictions as required by the floating-point hardware. Floating-point values are not processed directly by the RT processor.

MEMORY MODEL

The RT processor's memory model is extremely simple, and intentionally so. Memory is addressed at the byte level, with bytes numbered from left to right. Any quantity (byte, halfword, fullword) loaded from memory into a register appears in the register exactly as it did in memory; bytes are never reversed. The memory address of any instruction, primitive data item, or data structure is always the address of its leftmost (high-order) byte, without exception.

In real memory mode, addresses are treated as byte numbers in the range of 0 to $2^{24}-1$. The maximum real memory size is 16 megabytes. The RT processor begins execution in real memory mode after power-on reset, and this mode is used by some of the low-level operating system code, such as interrupt handlers.

Almost all programs are executed in virtual memory mode, including all application code and most of the operating system. In this mode addresses are again simply byte numbers, but with an extended range: 0 to $2^{32}-1$, thus spanning 4 gigabytes.

A program sees virtual memory as a large, linear byte-space with no arbitrary boundaries or overlapping segments. This is an important simplification and contributes to the ease of generating code for the RT processor. The compiler doesn't have to worry about whether a data structure, a stack frame, or even a program will fit within some restricted addressing range covered by a segment register.

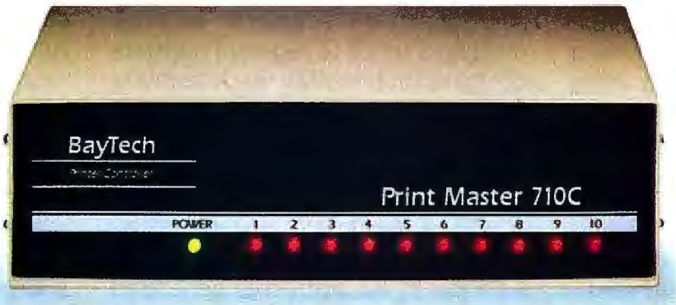
Of course, there are not really 4 gigabytes of main memory behind the 32-bit addresses generated by a program. Address translation hardware in the MMU maps virtual addresses to real addresses, signaling a page fault on references for which a mapping does not currently exist. The fault is resolved by reading in the relevant

(continued)

Now Available!

FIVE NEW MODELS

SHARE PRINTERS AND BUFFER PRINT JOBS WITH ONE VERSATILE UNIT



Print Master from BayTech is an intelligent printer controller that connects between your computers and printers. It allows you to share one printer automatically, contend for multiple printers automatically, or switch between several printers by sending a simple code, not by changing cables. Plus, Print Master's generous built-in buffer spools data until your printers can receive it.

Because Print Master is a very flexible device, you can set it up to fit your application, even if your application changes.

YOU SET UP THE IN-OUT PORTS

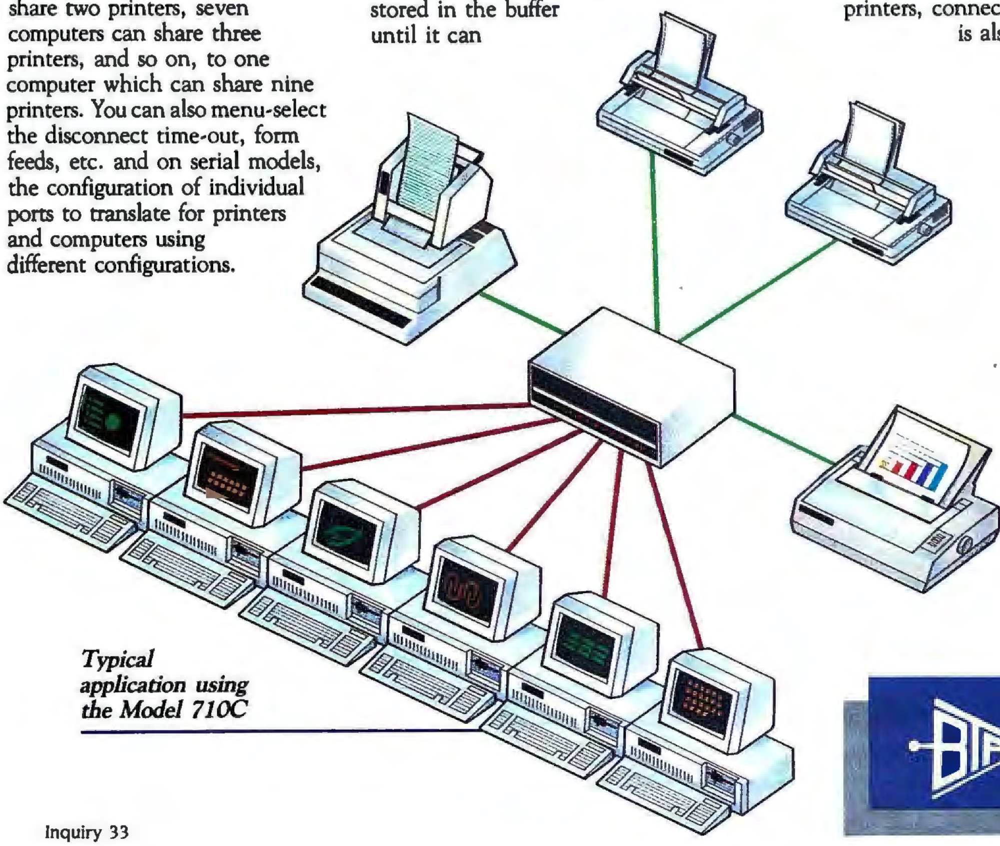
You configure Print Master's ports for any combination of printers and computers by answering questions from easy-to-follow menus. For example, with the ten port Print Master, nine computers can share one printer, eight computers can share two printers, seven computers can share three printers, and so on, to one computer which can share nine printers. You can also menu-select the disconnect time-out, form feeds, etc. and on serial models, the configuration of individual ports to translate for printers and computers using different configurations.

512K OR ONE MEGABYTE BUFFER KEEPS YOU WORKING INSTEAD OF WAITING

Since Print Master can accept data faster than your printer (up to 19.2KB serial or 5,000 characters per second parallel), you can send a print job to Print Master's standard 512K buffer and then go on to another project. All users connected to Print Master can send data to this common pool buffer, and they can be doing it simultaneously, even if no printer is available. Data is stored in the buffer until it can be sent on a first-job-in first-job-out basis to the selected printer. If you need more memory than 512K, Print Master is optionally available with one megabyte buffer.

If several users are sharing one printer, printer sharing via Print Master is completely automatic. There are no codes to send. You simply perform your normal print operation. If you are sharing several identical printers, connection is also automatic. Again, you perform your normal print operation and are connected to the next available printer on a first-come-first-serve basis. Print Master will send data to all printers simultaneously to keep your printers running at full capacity.

If you are sharing several different printers, such as a laser-jet, a dot matrix and a plotter, and you wish to select a specific printer, you do your normal print routine and also send a printer select code (which you can define yourself) before the first characters of your data. The data is then routed to the selected printer. It's that easy.



Typical application using the Model 710C

PARALLEL, RS-232C OR RS-422A MODELS

706A (6 parallel ports), $795
706C (6 serial ports), $795
708C (8 serial ports), $895
710C (10 serial ports), $995
706D (4 parallel/2 serial), $795
708D (6 parallel/2 serial), $895
706E (4 serial/2 parallel), $795
708E (6 serial/2 parallel), $895
710E (8 serial/2 parallel), $995
708F (4 serial/4 parallel), $895
710F (6 serial/4 parallel), $995
All above models have standard 512K buffers.
Additional 500K buffer, $249
RS422A for distances up to 4,000 feet now available on some models.

WANT DETAILS?

Call or write BayTech at P.O. Box 387, Highway 603, Bay Saint Louis, Mississippi 39520, USA. Telex 910-333-1618. Phone 601-467-8231 or

800-523-2702

BAY TECHNICAL ASSOCIATES, INC.
DATA COMMUNICATIONS PRODUCTS

page from disk or creating an empty page on first reference (perhaps first writing out some other page to make room in memory), updating the virtual address mapping information, and restarting the faulting program. Mapping is done on a per-page basis, with page sizes of either 2K bytes or 4K bytes. Page size is fixed at system initialization (the operating system for the IBM RT PC uses 2K-byte pages).

Virtual memory is organized into large segments, and each segment is composed of a collection of pages (see figure 2). Segment sizes can range up to 2^{28} bytes, or 256 megabytes. The pages of a segment are related in some way. A segment may make up a file in the file system, it may be all the computational (local) memory of a running program, or it may be all the executable code loaded for a particular task.

Each segment in active use is assigned a 12-bit number called a segment ID, or SID. A program can access up to 16 segments at once through a set of 16 segment registers contained in the MMU. Each segment register contains an SID, and the contents of the segment registers (i.e., the collection of names of segments to which a task has been given access) is part of the "state," maintained by the operating system, for each task.

Of the 32-bit addresses generated by a program, bits 0 through 3 (the high-order 4 bits) select one of the segment registers (0 through 15), and bits 4 through 31 specify an offset within the segment. Address translation is performed on the combination of the 12-bit SID from the selected segment register and the 28-bit offset. A 40-bit virtual address is formed by concatenating the SID and the offset. The virtual addressing range is therefore 2^{40} bytes, or 1 terabyte (one terabyte = 1024 gigabytes; tera- is the standard prefix denoting a factor of 10^{12}, from the Greek *teras*, which means "monster").

The loading of the segment registers is restricted to privileged state programs, but via an SVC an application can cause any segment, to which it has been granted access, to be mapped using any segment register. A task may access many more than 16 segments by reloading the segment registers. Segments can be shared among tasks merely by loading the same SID value into a segment register for each task. Since the translation is done on the 40-bit address rather than the 32-bit address, it does not matter that a shared segment is mapped using, say, segment

(continued)

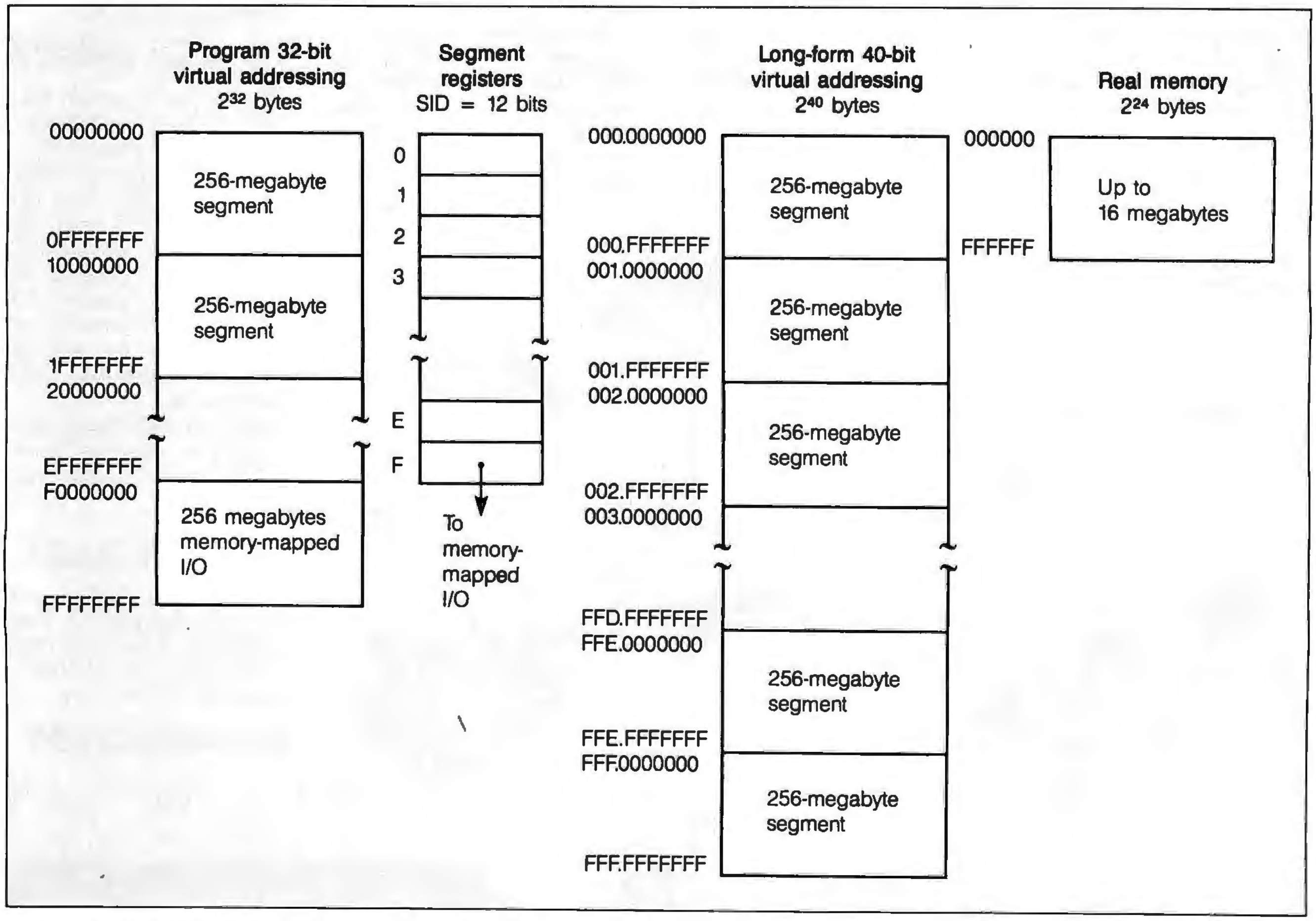


Figure 2: A *model of* RT *processor memory addressing.*

Mark Williams

Now the biggest name in C compilers comes in a size everybody can afford.

Let's C.™

$75

Introducing Mark Williams' $75 C compiler. Want to explore C programming for the first time? Or just on your own time? Now you can do it in a big way without spending that way. With Let's C.

This is no little beginner's model. Let's C is a powerful programming tool, packed with all the essentials of the famous Mark Williams C Programming System. The one chosen by Intel, DEC, Wang and thousands of professional programmers. The one that wins the benchmarks and the reviewers' praise:

"Let's C is a thoroughly professional C environment loaded with tools and programming utilities. It is also another fine Mark Williams Co. product."
—Christopher Skelly, Computer Language Magazine, Feb. 1986

And now for more big news. Get our revolutionary csd C Source Debugger for just $75, too. You can breeze through debugging at the C source level ignoring clunky assembler code.

Affordable, powerful, debuggable. Mark Williams Let's C is the big name C compiler at a price you can handle. Get your hands on it now.

Mark Williams Let's C

- For the IBM-PC and MS-DOS
- Fast compact code plus register variables
- Full Kernighan & Ritchie C and extensions
- Full UNIX™ compatibility and complete libraries
- Small memory model
- Many powerful utilities including linker, assembler, archiver, cc one-step compiling, egrep, pr, tail, wc
- MicroEMACS full screen editor with source
- Supported by dozens of third party libraries
- Upgradeable to C Programming System for large scale applications development

Let's C Benchmark Done on an IBM-PC/XT, no 8087. Program: Floating Point from BYTE, August, 1983.

	Exec Time in Seconds
Let's C	134.20
MS 3.0	347.45

Use this coupon or charge by calling toll-free: 1-800-MWC-1700. In Ill. call 312-472-6659.

ORDER NOW! 60-DAY MONEY BACK GUARANTEE!

Mark Williams Let's C

Please send me:

___ copies of Let's C and ___ copies of csd (C Source Debugger) at $75 each. (Ill. residents add 7% sales tax.)

☐ Check ☐ Money Order ☐ Visa, MasterCard or American Express

Name___

Address___

City___ State___ Zip___

Card #___ Exp. Date___

BYIBM

Signature___

Mark Williams Company
1430 West Wrightwood
Chicago, Illinois 60614

Inquiry 180

Table 2: *Protection conditions associated with a virtual page.*

Page Key	Type of Page	SR Key	Load	Store
00	Key 0 fetch-protected	0	Yes	Yes
		1	No	No
01	Key 0 read/write	0	Yes	Yes
		1	Yes	No
10	Public read/write	0	Yes	Yes
		1	Yes	Yes
11	Public read-only	0	Yes	No
		1	Yes	No

register 12 in one task and segment register 7 in another, as long as both tasks use the same SID.

For each segment, one of two types of memory protection can be specified. A bit in each segment register tells whether the associated segment is normal or special. For normal segments, protection is at the page level. Access rights to each page can be specified as read/write, read-only, or no-access, depending on a segment-key bit in the segment register and two page-key bits in the page table entry. Operating system programs normally run with the segment-key bit = 0, while applications run with the segment-key bit = 1. (Table 2 shows the various protection conditions associated with a virtual page.)

A finer-grained type of protection is available for special segments. Pages of special segments are divided into 16 "lines" of 128 bytes each (for 2K-byte pages) or 256 bytes each (for 4K-byte pages). Read-only or read/write access can be granted on an individual "line" basis, per task.

This type of protection is intended for use by sets of tasks that share a data segment and that need hardware assistance in controlling access to shared data items smaller than a page. Special segments require more effort on the part of the operating system because more manipulation of the page table is involved.

The Memory Management Unit

The MMU performs the address translation and memory protection operations described above. It is the "physical memory" controller for the RT processor as well. All requests for memory read or write go through the MMU, including requests from the RT processor proper and from direct memory access I/O devices. Each request is distinguished (by a control line during the bus transfer) as real, which requires no translation, or virtual, which requires translation and memory protection checking.

Segment Registers

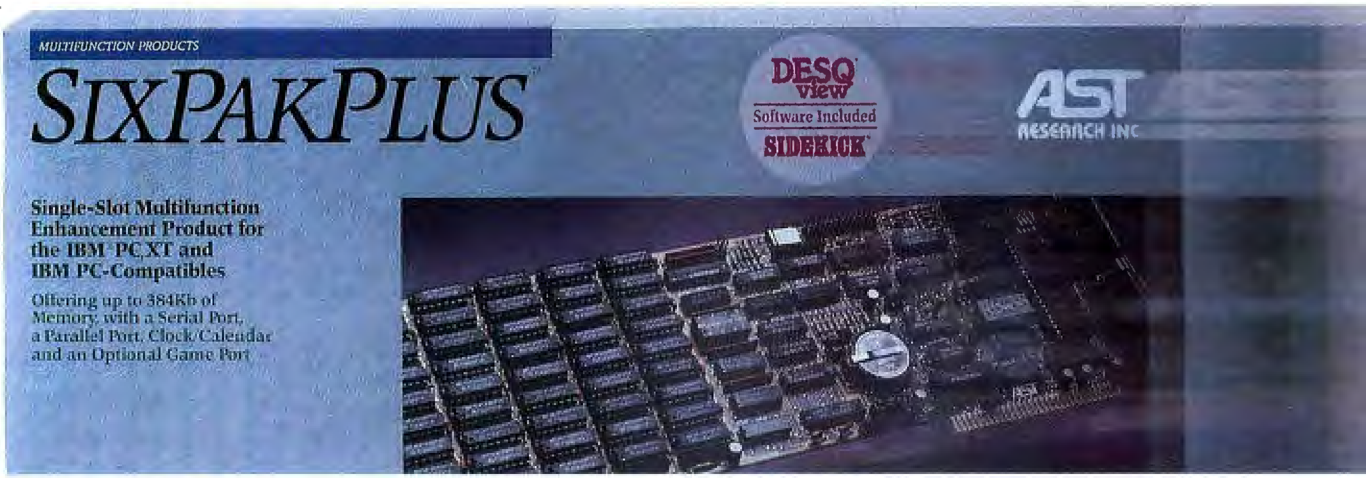
The 16 segment registers described in the memory model reside in the MMU. Address and data transfers to and from the MMU are 32 bits wide. The 32-bit virtual addresses are expanded to 40 bits in the MMU by stripping off bits 0 through 3 and using them to select one of the segment registers. The resulting 40-bit address is then translated to real, and the real address is used to drive the memory arrays. The contents of the segment registers can be read and written by the operating system using the RT processor's I/O instructions.

Inverted Page Table

Conceptually, address translation is performed by looking up the virtual page number (the high-order bits of the 40-bit virtual address) in a table, finding the associated real page number; and substituting the real value for the virtual one. Since the pages are fixed in size (all 2K bytes or all 4K bytes) and must reside in memory on a boundary that is a

(continued)

AST Sets The Multifunction Standard.



First there was SixPakPlus®...the single most popular PC enhancement product in history. Now AST makes the best even better. SixPakPremium™ sets the new standard for multifunction PC enhancement.

Expanded Memory and More I/O. The new standard includes up to two serial ports, a parallel port, optional game port and battery-backed clock/calendar. And we've beefed up the memory, giving you up to 2 megabytes of conventional and/or enhanced expanded memory.

Intelligent Memory.™ We dubbed our enhanced expanded memory *Intelligent Memory.* Intelligent, because it combines full Lotus®/Intel®/Microsoft® EMS compatibility with new, exciting capabilities. Capabilites no ordinary EMS board can match!

Guaranteed EMS Compatible. Test our Premium Guarantee. If within ninety days of purchasing a SixPakPremium, you find an EMS software package incompatible with it, we'll either establish compatibility or buy you a comparable EMS board of your choice.*

THE AST PREMIUM GUARANTEE

More Capability Than Ordinary EMS. SixPakPremium, together with its DESQview™ software—included at no extra cost, lets you organize up to nine standard PC applications at a time in your Intelligent Memory. And DESQview is compatible with hundreds of the most popular PC applications.

FastTask:™ The End Of Workflow Interruptions. Simply load all the applications you use during the course of the day into your Intelligent Memory once in the morning. It's that easy.

Now you're ready to FastTask. That's what we call the ability to move instantly from application to application. There's no fussing with diskettes or closing

*Applies to boards purchased after 9/1/86. Valid only in U.S.A. Some additional restrictions apply. For complete details write to AST Research, Attn. M.C.G. *SixPakPlus registered and SixPakPremium, Intelligent Memory, FastTask, Premium Guarantee and SuperPak trademarks AST Research, Inc. IBM registered trademark International Business Machines Corp. DESQview trademark Quarterdeck Office*

Future technologies will permit the addition of architectural enhancements.

multiple of that size, the low-order bits of the virtual address reflect an offset within the page and are not used in the translation. The virtual page number is the high-order 28 or 29 bits of the 40-bit address, depending on page size.

In traditional translation schemes such as that of the IBM System/370, the page table is organized by virtual address. That is, the virtual page number is used directly as an index into a one- or two-level table to obtain the real page number. As virtual storage becomes very large and as the number of independent tasks supported increases, such tables can become unwieldy because the overall table size is proportional to the amount of virtual storage supported by the operating system.

The RT processor uses an inverted page table instead, an approach first used on the IBM System/38. The table is organized by real address with the contents of each entry being the corresponding virtual address. The table's maximum size is therefore one entry per real page rather than one entry per virtual page. Only a fixed percentage of the machine's real memory is required for the inverted page table regardless of the amount of virtual storage in use.

If the page table entries are in order by real address, how can a virtual-to-real translation be done quickly? It would seem that a sequential scan of the page table would be required, comparing the virtual address with each table entry until a match is found. On the average, half the table entries would have to be checked.

The RT processor avoids this sequential search by using a hashing technique to quickly find an entry for a given virtual address (see figure 3). The 28- or 29-bit virtual page number is "hashed" into a number in the range of 0 to $n-1$, where n is the number of pages of real memory. Thus the range of 2^{28} or 2^{29} virtual page numbers is quickly reduced to one-of-n congruence classes. For each congruence class there is a chain of entries threaded through the inverted page table. Each entry on a given chain is for a real page whose virtual page number hashes to the same congruence class (i.e., a "hash collision").

Translating from virtual to real, then, requires only a simple hash operation (just an exclusive-OR) followed by a sequential search of the entries on the hash chain. This technique works well if the hash chains are very short compared to half the total number of entries in the table. In fact, the average chain length is between one and two entries, so that only a few memory accesses are required to find the real entry for a virtual page. The worst case, of course, would occur if all the virtual pages in the system wound up on the same hash chain, but the chances of that happening are exceedingly remote.

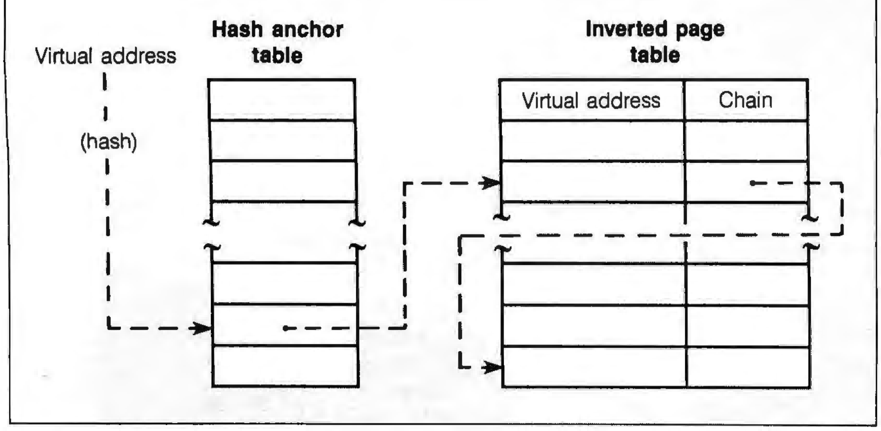


Figure 3: RT *processor hash anchor table and inverted page table.*

TRANSLATION LOOK-ASIDE BUFFER

Regardless of page table organization, no system in which performance is a goal can afford to go to the page table in main memory for each and every translation. In the MMU, a cache of recently used translations is kept in a translation look-aside buffer, or TLB. This is a two-way, set-associative table with 32 total entries. Each entry contains a virtual page number, its corresponding real page number, and memory protection information for the page.

One of 16 congruence classes is selected by using the low-order 4 bits of the virtual page number. For that class, two comparisons are made, in parallel, against the remainder of the virtual page number. If a match is found, the translation can be completed immediately. If no match is found in the TLB, the page table must be searched and the TLB updated.

Hit ratios of address translation caches are typically very high, much higher than those of data caches, for example. The hit ratio for the TLB is about 99.5 percent. As a result, address translation is normally completed within a single cycle. When the TLB does miss, translation is usually extended by only 8 to 11 cycles while the TLB is reloaded.

HARDWARE IMPLEMENTATION

The RT processor and its MMU are each implemented as single VLSI chips. With a cycle time of 170 nanoseconds, the sustained execution rate of the RT processor on a typical mix of instructions in the IBM RT PC is ap-

(continued)

Speed is of the essence.
The C. Itoh ProWriter® C-310/C-315.

If there's anything to that old "time is money" adage, the C-310 and wide-carriage C-315 are the smart buys. Because at up to 300 cps, they're a third faster than other printers in their class—printers like the Epson FX-85,® Okidata 292® and IBM ProPrinter.®

And with features like Auto Paper Loading, a built-in push/pull tractor, and bottom paper feed, you'll spend more time printing and less time fussing with your printer.

The Maserati of printers.
Nationally syndicated computer columnist Peter McWilliams is a man who knows his printers. And when he called the C-310 "... the Maserati of printers," he meant there was a lot more to it than just speed.

Of course, the C-310/315 smokes through draft printing at about 100 cps faster than the competition. But with a graphics resolution of 240 x 216, it's exceptionally crisp in the curves and corners of pie charts and bar graphs. And because it's switch selectable between Epson FX-85 and IBM ProPrinter control codes, it will run on the most popular business software available.

Yet despite all this performance, the C-310/315 features advanced sound insulation to make printing as smooth and quiet as riding in a Rolls-Royce.

True letter quality.
If you're in business—and who isn't?—the C-310/315's best feature may be the one-button, front-panel selection of letter quality print with up to 27 dots of vertical makeup. You get quality that rivals expensive office typewriters. And with the C-310/315's color option and plug-in font cartridges, you can express yourself in seven vivid colors and a variety of different typefaces.

Total support.
Because it's from C. Itoh Digital Products, the C-310/315 is backed by a full one-year warranty on parts and labor, with Authorized Service Centers available nationwide.

More importantly, it's backed by the $90 billion worldwide trading company C. Itoh & Co., Ltd.

So you can depend on C. Itoh to be there when you need them.

Fast track.
As you can see, there's a lot more to the C-310 and wide-track C-315 than unbridled speed. They are also exceptionally smooth, efficient, intelligently designed and, frankly, made for the discriminating business that knows the difference between "cost" and "value."

We can't promise that buying a C. Itoh C-310/315 will shorten your climb to the top. But it will get you out of the office faster.

Upgrade.

C.ITOH

C. Itoh Digital Products, Inc.
19750 South Vermont Avenue, Suite 220
Torrance, California 90502
Telephone: (213) 327-2110 or (800) 423-0300

Epson FX-85 is a registered trademark of Epson America, Inc. Okidata 292 is a registered trademark of Oki America, Inc. IBM ProPrinter is a registered trademark of International Business Machines Corporation.

Inquiry 51 for End-Users.
Inquiry 52 for DEALERS ONLY.

proximately 2 million instructions per second. The peak execution rate, which is reached in sections of normal compiled code where register-to-register operations predominate, is one instruction per cycle:

$$\frac{1 \text{ instruction}}{1.7 \times 10^{-7} \text{ seconds}} = 5.88 \text{ MIPS}$$

PIPELINES

There are two levels of pipelining in the implementation of the RT processor (see figure 4). First, the execution of instructions is done by a four-stage pipeline that overlaps instruction fetch, instruction decode and operand fetch, execution, and result storeback. At a higher level, memory operations (Load, Store, Instruction-Fetch) are pipelined so that several memory operations can be in progress at once and so that the processor need not wait for the completion of a memory operation before executing subsequent instructions. In the current version of the RT processor, instruction fetching is pipelined, but loads and stores are not (in virtual memory mode) so that precise page-fault interrupts can be presented.

In the low-level pipeline, the first stage is unique in that it proceeds asynchronously with the rest of the stages. That is, the instruction fetch mechanism is usually fetching a word that is several instructions beyond the one currently entering decode.

The instruction fetch unit in the RT processor tries to keep its 4-word buffer full by fetching the word at the next sequential address, incrementing its pointer by 4, and repeating this until all 4 words are full. If a branch occurs, the execution unit resets the instruction fetch unit's pointer to the branch target address and marks the four buffer entries "empty." The instruction fetch unit then starts fetching the new instruction stream.

The other three stages of the pipeline run in lockstep. In the first stage, a prefetched instruction is decoded and the contents of registers that it uses as source operands are fetched from the register file. Two registers can be read from the file and two written simultaneously. The first (usually the only) microcode word for the instruc-

(continued)

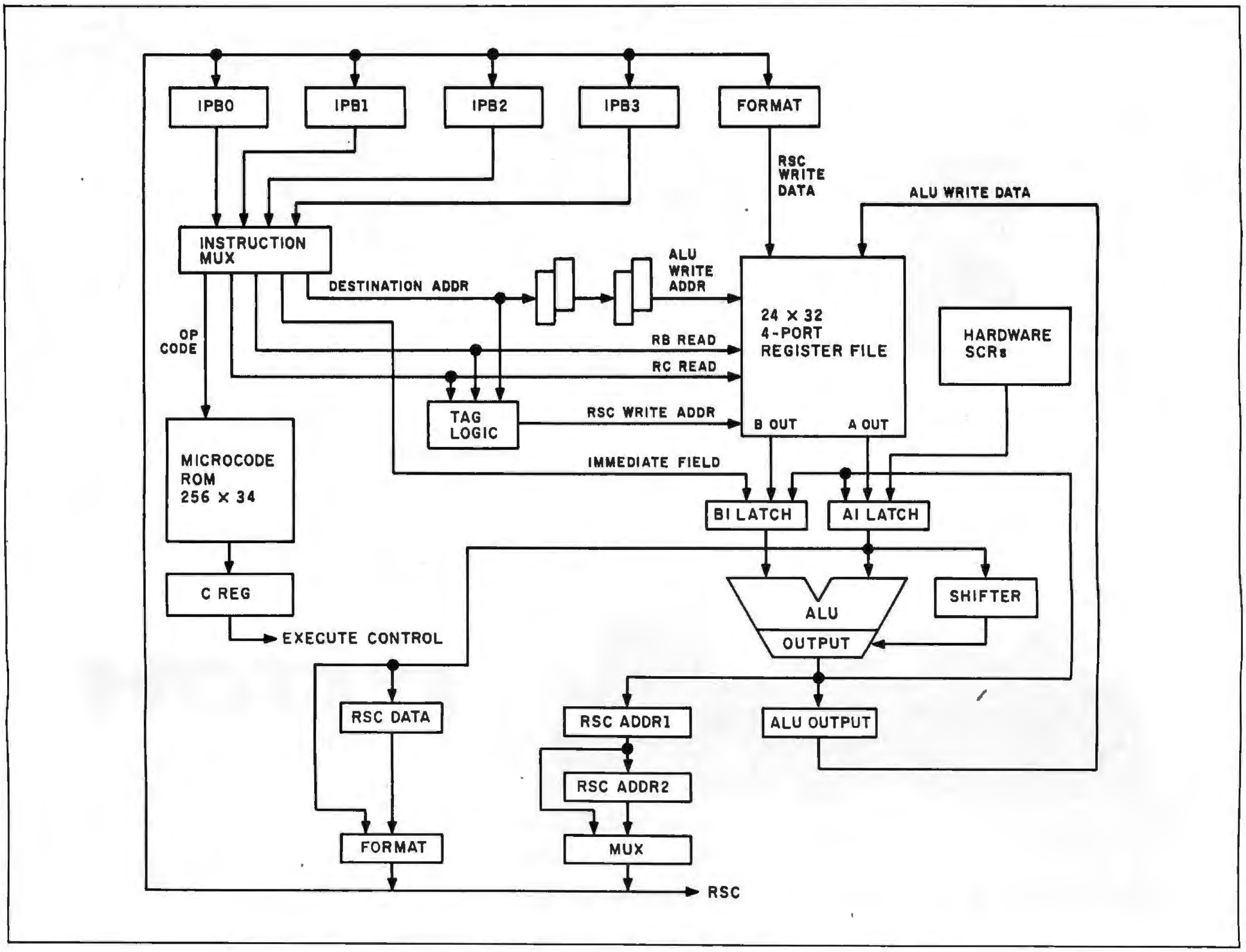


Figure 4: A *data flow block diagram of the* RT *processor.*

AT jr. NOW $599.

FREE! A $185.00 VALUE. With any order of $1,000 or more, AMT will send you, absolutely FREE, a *Micro* brand dress wristwatch. Finest precision Japanese quartz movement. Specify male or female size.

LAUGH, CLONE, LAUGH! Who would guess that our AT jr. has more on the ball than his big brother?

AT jr. not only delivers fantastic features at fabulous prices, but gives you **top quality** in design, components, materials, and construction!

And for reliability, every AT jr. is burned-in at the factory prior to shipment.

No IBM® compatible system gives you more capability and value for your dollar than our incredible AT jr.

AT jr. $599. This feature-packed model is based on an Intel 8088-2 (or NEC V20-8) CPU, running at 4.77 MHz or 8 MHz, selectable; handles PC DOS,® MS DOS,® XENIX,® and other operating systems, plus a full range of IBM-compatible software.

The standard system has a "0" wait state on memory, and operates many times faster than the IBM PC/XT.® Offers 640 KB of RAM on the motherboard; 360 KB floppy disk drive; eight expansion slots; LED indicator for the turbo mode; AT-style keyboard; 135-watt integral power supply; and security key lock. Optional features include a 40 MB hard disk (38 ms), EGA, and an 8087-2 math co-processor.

LAP TOP. $1699. AMT's compact Lap Top system gives you full IBM compatibility in less than 12.2 pounds of high-powered computer.

Features an 80C86 microprocessor, giving you twice the speed of IBM's lap top; 640 KB of memory; two 720 KB, 3.5-inch disk drives; an 80-by-25 LCD display with 640 x 200 resolution; RGB and printer ports; and a 1200 baud internal modem, optional.

AMT-286. $1199. AMT's top-of-the-line AT system outperforms the IBM PC AT® at a fraction of the price. Tagged "Best Priced" by **PC Week**, March 1986 issue.

Our phenomenal AT has a 16-bit Intel 80286 CPU running at 8 MHz (10 MHz or 12 MHz optional); 640 KB of RAM, expandable to 1 MB; 1.2 MB floppy disk drive; accommodations for 20, 30, or 40 MB hard disk drive; a socket for an Intel 80287 math co-processor; and a "0" wait state on memory, optional.

DON'T WAIT—CALL NOW

Pick up your phone now and get full details or place your order.

AMT

AMERICAN MICRO TECHNOLOGY
14751-B Franklin Avenue
Tustin, California 92680
Phones: (714) 731-6800;
in LA call (213) 477-6320;
in Bay area call (415) 490-7967
TWX 5106003265 AMT USA

*Prices and availability subject to change without notice. Dealer inquiries accepted.

®Registered Trademarks—IBM, IBM PC, IBM PC/XT, IBM PC AT, PC DOS: International Business Machines Corporation; MS DOS, XENIX: Microsoft.

tion is fetched simultaneously with the operand registers. The microcoded design of the RT processor does not slow down execution because the microword fetch is overlapped with the fetching of the operands.

In the next stage the decoded instruction is executed using the previously fetched register contents. In the last stage the results of the execution are stored back into the register file or are sent out to memory if this is a Store instruction. In addition, the register file is bypassed when an instruction uses the results of the preceding instruction. The output of the ALU is fed back around to be the input on the next cycle. In the normal course of events the instruction fetch unit will have fetched instructions far enough ahead so that the decoder does not have to wait. There will then be three instructions in process: one in decode, one in execution, and another in storeback so that the throughput is one instruction per cycle (see figure 5).

Multicycle instructions such as Divide Step spend several cycles in execution, thus delaying the instructions that follow them. In addition, it may not be possible to start execution of an instruction if one of the registers it uses is the target of a previous Load instruction that is not yet complete.

PACKET-SWITCHED MEMORY BUS

A Load instruction may not have been completed due to the pipelining of the memory interface. Since the RT

(continued)

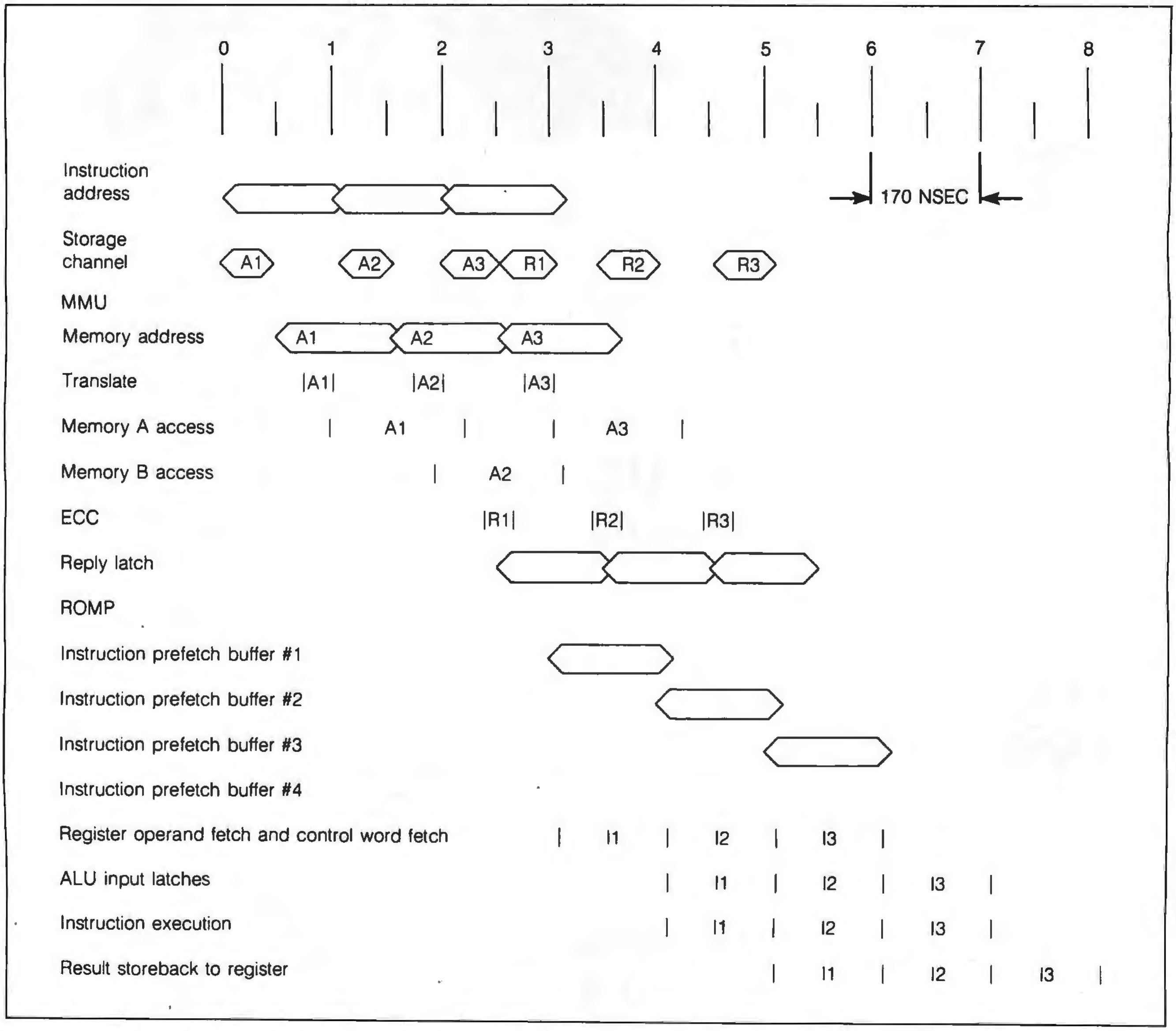
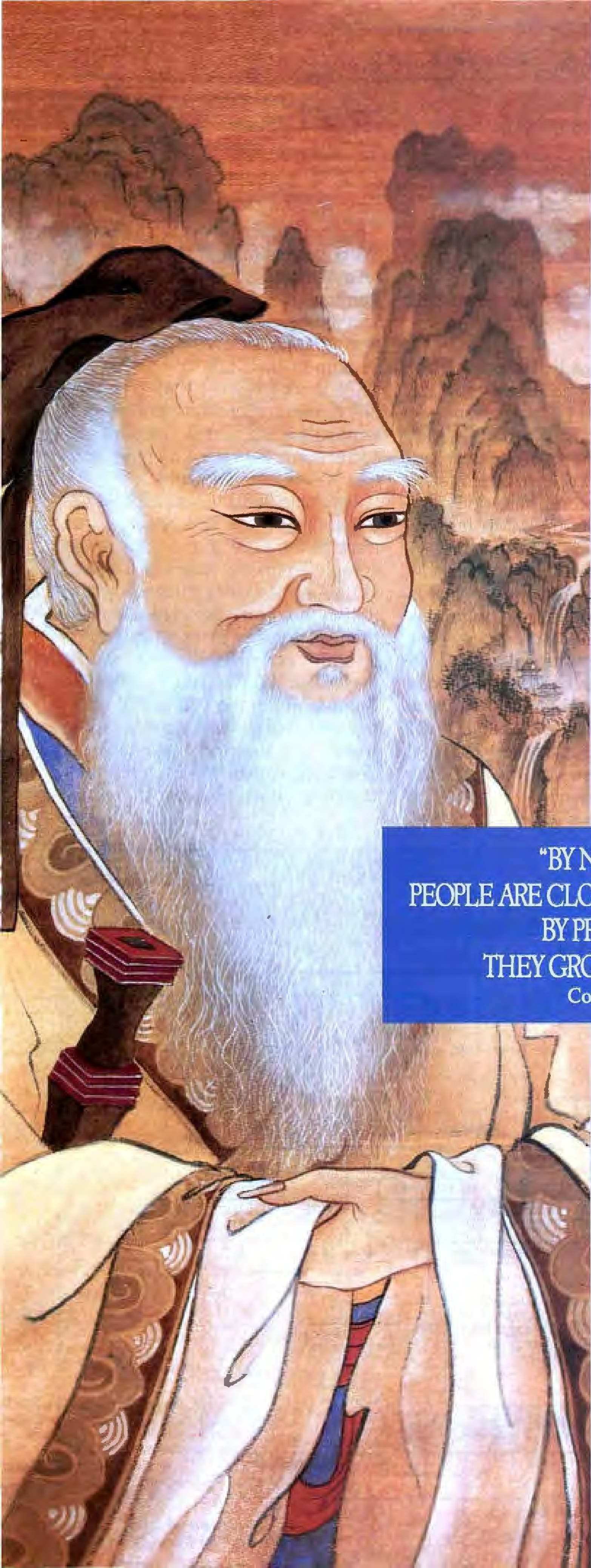


Figure 5: *Three 32-bit instructions. Note the timing overlap in memory access, instruction prefetch, operand fetch, and storeback to registers.*



ITT helps a growing company work together.

Today's leading business computers deliver tremendous power to individuals. One business microcomputer—the ITT XTRA™ XL—can join those individuals together in high-performance multi-user or network office environments.

ITT XTRA XL. The missing link.

Link multiple ITT XTRA XL computers together in a powerful network or use one ITT XTRA XL as a shared processor licensed for up to 16 users with 32 available ports. Connect your existing IBM-compatibles to a cost-effective LAN. Take full advantage of multi-user and networking technology today and far into the future with ITT XTRA XL's versatile, modular design.

Today's connectivity tool.

Other computers claiming connectivity do so at the cost of performance and speed. Not the ITT XTRA XL.

It combines the powerful Intel 80286 main processor with a separate processor for terminal management. It runs DOS or XENIX operating systems, features zero-wait-state memory access, and up to 144 megabytes of very fast disk storage.

And you get all this astonishing performance for far less than the cost of the ITT XTRA XL's nearest competitors.

A family of solutions.

The ITT family of microcomputers provides high performance, compatibility and uncompromising answers for your business applications. The ITT XTRA for entry-level efficiency. The ITT XTRA XP for demanding single-users. The ITT XTRA XL for multi-users and networks. **Your ultimate computer solution is just a phone call away. Call (800) 321-7661. In California, (800) 368-7300.**

ITT
BUSINESS COMPUTERS

processor's main memory does not run as fast as a cache, it would be time-consuming to wait for each Load or Store operation to end before beginning the next operation. To allow multiple memory operations to be in progress at once and to gain maximum use of interleaved (multibanked) memory, a packet-switched bus was defined between the RT processor, the MMU, and the I/O controller.

This bus is known as the ROMP storage channel, or RSC (unfortunately, a nested acronym). The RSC runs synchronously with the RT processor cycle, with two RSC cycles per processor cycle. The standard RT processor cycle time in the IBM RT PC is 170 nanoseconds (the RSC runs at 85 ns/cycle). Even-numbered RSC cycles are used to transmit addresses from the RT processor or from DMA I/O to the MMU, while odd-numbered cycles are used to transmit data.

Each transfer is 32 bits wide. In each processor cycle, therefore, both a memory request (address) and a unit of data (a word loaded or stored) can be transferred, for a net throughput of

$$\frac{4 \text{ bytes}}{1.7 \times 10^{-7} \text{ sec.}} = 23.5 \text{ megabytes/sec.}$$

The address and data portions of the cycle are not necessarily related. In the first half of a processor cycle (i.e., in the even-numbered RSC cycle), the processor may send out an address for a load or instruction fetch, while in the second half (odd-numbered RSC cycle) the MMU may respond with the data for an earlier request. Requests are not necessarily responded to in order, depending on whether the required memory bank is busy when the MMU processes the request. To keep the outstanding requests straight, each request (address) has an associated tag, a 5-bit value that tells who owns the request (RT processor or I/O controller) and what should be done with the response when received (sent to the instruction fetch unit or to a general register).

Further pipelining occurs within the MMU (see figure 6). We have seen that several memory requests can be outstanding at once. The MMU overlaps operations for the various requests—performing address translation (lookup in the TLB) for one request while driving the memory arrays for one or two earlier requests while performing an ECC (error checking and correction) check for a memory read that has been completed.

Assuming that a virtual address has a mapping in the TLB, which is true most of the time, translation can be completed in one cycle. For a typical virtual mode Load instruction in the RT PC, the total time from execution of the instruction in the RT processor to return of the data by the MMU is five cycles. If the required translation is not found in the TLB (a TLB miss), the MMU takes additional cycles to search the inverted page table as described earlier. The hash chain for the congruence class for the virtual

(continued)

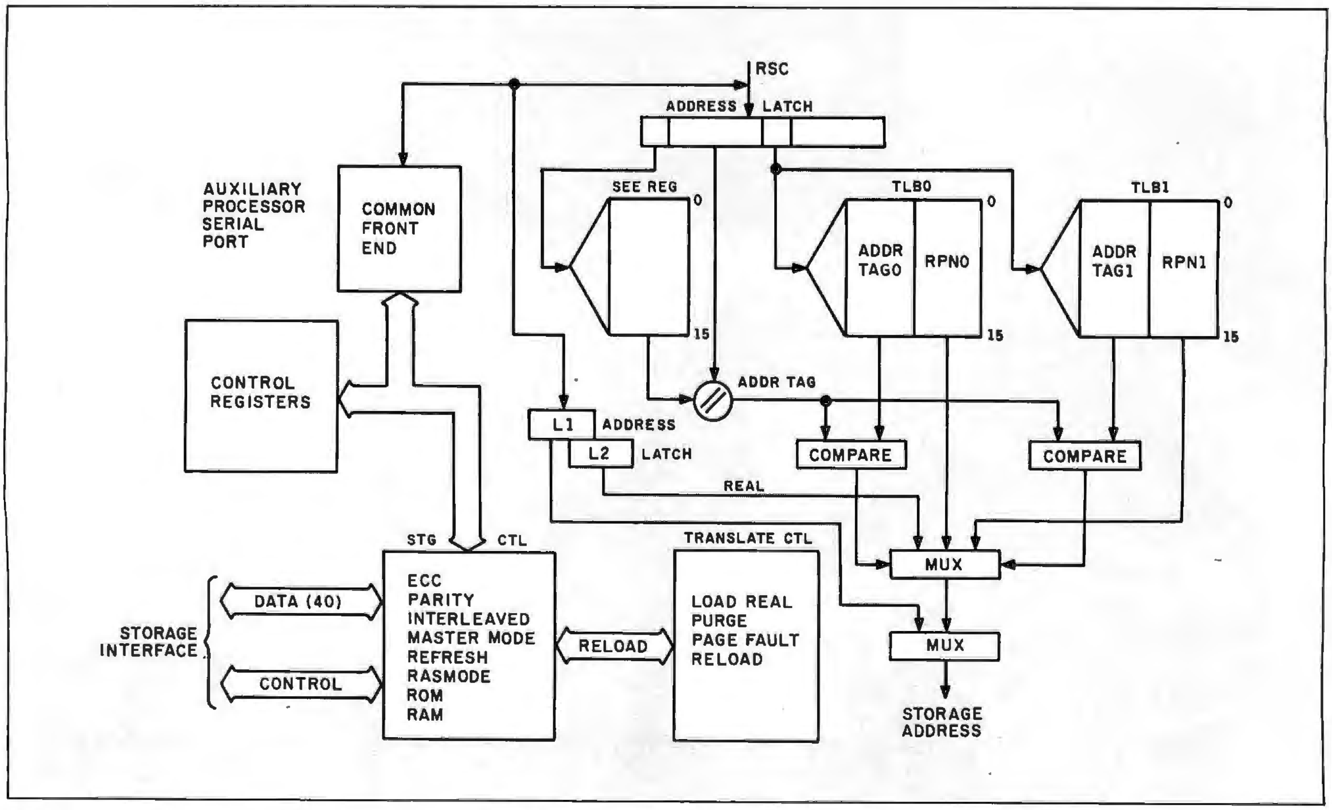


Figure 6: RT *processor memory management unit data flow.*


Clipper is the fastest dBase III and dBase III Plus™ compiler available. Nothing else comes close. When performance counts, experts rely on Clipper for more speed, more power, and more creative freedom. You can, too. Call for details.
• Clipper compiled programs run 2 to 20 times faster.
• No royalties...no runtime fees.
• Source code security.
• User defined functions.
• Arrays.
• Simple menu commands.
• Context sensitive help can be included with programs.
• More fields; more memory variables.
• Call C and Assembly programs.
• Complete debugging facilities.
• Multiple file relationships.
• IBM PC, XT, AT, 3270 compatible™.
• Multi-user capability.
Clipper™
CLIPPER. THE dBASE COMPILER.
A WINNING PERFORMANCE EVERY TIME.
Nantucket™
Nantucket Corporation
5995 South Sepulveda Boulevard
Culver City, California 90230
(213) 390-7923
Outside California call toll-free:
1-800-251-8438
dBase, dBase III, and dBase III Plus are trademarks of Ashton-Tate, Inc.
IBM PC, XT, AT, and 3270 are trademarks of International Business Machines Corporation.
Clipper and Nantucket are trademarks of Nantucket Corporation.

page number is searched, and if a matching page table entry is found, it is automatically loaded into the TLB in place of the least recently used TLB entry for its TLB congruence class, and the translation is completed.

Given that the hash chains are short, this typically adds 8 to 11 cycles to the translation operation. With a hit ratio in the 99.5 percent range, this automatic page table search only degrades the overall RT processor performance by a few percent. By contrast, if the TLB miss resulted in an interrupt to the processor and a table search by software, the overall performance could be degraded by 30 percent or more.

When both the TLB and page table searches fail, the result is a page fault. In this case the MMU does not complete the memory request and causes a page-fault processor interrupt. The operating system then resolves the fault by reading in or creating a page and restarting the faulting program (or by terminating the task, if this access is really an error).

Instruction fetch requests, which are interleaved with Load/Store requests from the instruction stream, can also cause page faults. The MMU translates these requests in the same way and reports page faults to the processor, but the processor does not generate an interrupt unless the faulting word is actually needed by the instruction stream. The RT processor therefore does not process spurious page faults caused by the eager instruction fetch unit fetching ahead beyond the end of a page, when the instruction stream will actually branch before crossing the page boundary.

If the translation is successful but the memory protection bits for the page don't match the type of request, then an interrupt similar to a page fault is generated. The operating system can change the protection status of the page and restart the program, or it can terminate the task in the case of an addressing error.

To assist the operating system in selecting pages for replacement, the MMU maintains a table of reference and change bits for each real page. This table is stored in a small static RAM connected to the MMU and is updated in parallel with the actual memory operations. The operating system can read these "flags" for each page to determine which pages are "dirty" and must be written to disk before being reused and which pages have not been referenced at all and are therefore good candidates for reuse.

As the physical memory controller in an RT processor–based system, the MMU is also in charge of such mundane matters as hidden refresh for dynamic RAMs and performing ECC or parity checks on data read from memory. Either ECC (40 bits wide, on a 32-bit word) or parity (1 parity bit per byte) can be used with the RT processor. (The IBM RT PC uses the ECC mode.)

Timing values that control access to the dynamic RAM arrays are set by components external to the MMU, so that a wide variety of RAMs can be used with an RT processor–based system. The IBM RT PC uses 150-ns access time memory chips interleaved in two banks. The MMU also supports the use of read-only memory in addition to RAM. On the IBM RT PC the initial RT processor checkout program begins executing from ROM. This program initializes the RAM, reads in a larger initialization program from disk, branches to the program in RAM, and then switches the ROM out of the addressing space, causing all further execution to be from RAM.

THE FUTURE

Improvements to the RT processor system and RT processor's MMU could come from two sources: technological advances over time and architectural enhancements. RISC architectures like that of the RT processor scale well as technology improves. With smaller dimensions of VLSI geometry come both increased density and faster clock times. Since most RT processor instructions require only one cycle, a reduction in the cycle time translates directly into increased execution speed.

(continued)

Slash development time

Easy does it with your PC and our HP PC Instruments.

Easy to set up — HP PC Instruments are easy to install. Simply connect instruments and accessories and load the software; you're up and running in a few hours. They're easily controlled with the HP Vectra PC (below); the IBM PC, PC/XT or PC/AT; the AT&T PC 6300; or other selected compatibles over the HP PC Instruments Bus (PCIB).

Easy to debug — Soft front panels let you see the status of all instruments in the system at a glance. And you can switch easily between panels and your BASIC program for simple program debugging.

Easy to use — Soft front panels (oscilloscope display shown below) let you enter much of your program with a mouse or touchscreen which means less data entry for you. And the system software generates the major portion of your program at the touch of a key. Add your own touches to customize and you're done. Result: You can generate complete test programs in days instead of weeks.

Easy to afford* — PC Instruments are inexpensive. You can buy and use as many, or as few, as you need.

4½ Digit Digital Multimeter: $700
50MHz Digital Oscilloscope: $1500
5MHz Function Generator: $1500
100MHz Universal Counter: $900
Relay Multiplexer: $700
12 bit Dual Voltage DAC: $800
16 bit Digital I/O: $700
Relay Actuator: $700
20 Volt Power Supply: $900
PCIB Interface: $500
HP-IB Interface: $400
Optional System Power Unit: $400
ASYST™ Scientific Software+: $1695-$2495

Easy to combine with HP-IB — For optimum price and performance, mix and match traditional HP-IB instruments, like the voltmeter below, with any of the PC modules. Add PC modules to traditional instruments, and vice-versa. And control everything with the same BASIC program.

Free technical assistance — Call the PC Instruments Support Hot-Line at (201) 627-2500. Talk to an engineer about how PC Instruments can automate your testing. And request technical literature and a free demo disc that simulates system operation. Or call your local HP sales office listed in the white pages of your telephone directory. Ask for the Electronic Instruments Department.

*Prices are U.S.A. list prices only.
+ASYST™ Scientific Software is a trademark of Macmillan Software Co., a div. of Macmillan, Inc.

HEWLETT PACKARD

Inquiry 125

HP—IB 3478 Voltmeter

D 2115651

Normal throughput is one instruction per processor cycle.

The greater density available in future technologies will permit the addition of architectural features that had to be omitted from the current RT processor and MMU. Probably the most obvious enhancement would be the restoration of one or both of the caches from the original 801 design. Additional silicon space would also permit the inclusion of functions that do not fit the RISC paradigm of one cycle per instruction but that have proven (by measurement of significant running code) to be very useful for enhanced performance.

The MMU's virtual memory mapping functions permit the operating system to create a one-level store in which objects traditionally considered to be disk files become simply arrays of data structures in virtual memory. I/O operations on such a file, instead of being explicit reads and writes, are implicit and are handled by the Paging Supervisor. This can be much more efficient as well as being far easier to program than the traditional access methods. Increases in memory size and cache size will mean that programs traditionally thought of as I/O-bound may become processor-bound. These programs will benefit even more from a faster processor.

For both areas, pure technological speedup and additional functions, the architecture of the RT processor provides an extensible base for the future.

BIBLIOGRAPHY

IBM Corporation (various authors). *IBM RT Personal Computer Technology*, form number SA23-1057. Available through IBM branch offices. This book consists of more than two dozen papers on various aspects of the RT processor, MMU, and the IBM RT PC. "RT Processor/MMU Implementation" on pages 57–65 is of particular interest.

Hennessy, John, Norman Jouppi, Steven Przybylski, Christopher Rowen, Thomas Gross, Forest Baskett, and John Gill. "MIPS: A Microprocessor Architecture." *Proceedings of the 15th Annual Microprogramming Workshop*, SIGMICRO '82, 1982, pages 17–22.

Markoff, John. "RISC Chips." BYTE, November 1984, pages 191–206.

Patterson, David A., and David R. Ditzel. "The Case for the Reduced Instruction Set Computer." *Computer Architecture News*, October 15, 1980, pages 25–33.

Patterson, David A. "Reduced Instruction Set Computers." *Communications of the ACM*, January 1985, pages 8–21.

Radin, George. "The 801 Minicomputer." *SIGPLAN Notices*, April 1982, pages 39–46. ■

Why Kaypro?

Here are eight reasons to choose the KAYPRO PC.

1. **Non-Obsolete Design**
   It's a snap to update vital system components – right down to the microprocessor.

2. **IBM Compatibility Guaranteed***
   Only Kaypro guarantees IBM compatibility – and gives you three times the IBM standard memory – with the KAYPRO PC's 768KB RAM.

3. **12-Month Warranty**
   A one year parts and labor warranty is standard.

4. **Included Software: WordStar... Plus**
   Every KAYPRO PC comes standard with WordStar, MS-DOS 3.2, GW BASIC, K-Desk, PolyWindows, CorrectStar, MailMerge, and more.

5. **Two Speeds For Fast Processing**
   Zip along at 4.77MHz – or zoom with 8.0MHz.

6. **Both IBM Video Standards... Plus**
   Get maximum software compatibility with IBM monochrome and CGA video standards – and Hercules emulation – built in.

7. **IBM PC AT-style Keyboard**
   Just like the expensive IBM PC AT – complete with tamper-proof keylock.

8. **Made in the USA**
   Few IBM PC or XT compatibles can make this claim.

Suggested retail price: $1595.

Visit us at Comdex

Booth #1280

KAYPRO® CORPORATION
Innovators of Electronic Products for Over 34 Years

For the dealer in your area, call tollfree 1-800-4-KAYPRO (1-800-452-9776) or 1-619-481-4300.

Trademarks: IBM and AT are trademarks of International Business Machines, Inc.; Hercules, Hercules Computer Technology; WordStar, CorrectStar, MailMerge, MicroPro International; GW BASIC, Microsoft, Inc.; PolyWindows, POLYTRON; K-Desk, Kaypro Corporation. *For the first 90 days after purchase, Kaypro assures that any major IBM-compatible business software will run on the KAYPRO PC or they will grant a trade-in credit toward the purchase of an IBM PC. Trade-in for IBM PC is available only through original Kaypro dealer, and IBM suggested retail price applies.

Simply put: THE™ highest quality and best value in computer products anywhere.

THE™ commitment to our customers is compatibility, quality, value, service and support

THE™ backs every product with a

45 DAY MONEY BACK GUARANTEE

and a

FULL 1 YEAR PARTS AND LABOR WARRANTY

THE™ MULTI 384

This multi-function card features 0-384K memory, a parallel printer port, a serial port for communications, a clock/calendar with battery backup, and a software bonus that includes RAMdisk and other utilities. Retail $90.00

$67.00* Special Wholesale Price

BEST BUY Functional Clone of AST™ SixPackPlus

THE™ COLOR CARD

100% compatible with the IBM™ colorcard with display modes of 80x25 alphanumeric and 320x200 graphic. Retail Price $75.00

Special Wholesale Price **$60.00***

THE™ 576K MEMORY +

Supports 0-576K of available memory and is compatible with all IBM™ PC's. Retail $46.25

Special Wholesale Price **$35.00***

THE™ PRINTERFACE

Supports all text and graphics features and is fully compatible with third party software. Retail Price $23.13

Special Wholesale Price **$18.50***

THE™ MODEMS

100% Hayes compatible

THE™ 1200 COM EXTERNAL

This self-testing 1200 BPS modem comes with auto answer, auto dial, auto redial and a built-in speaker. Retail Price $161.25*

Special Wholesale Price **$129.00***

THE™ 1200 COM INTERNAL

Features auto answer, dial and redial, with a built-in speaker, RS 232-C serial port and PC Talk III. Retail Price $148.75

Special Wholesale Price **$109.00***

THE™ 2400 COM EXTERNAL

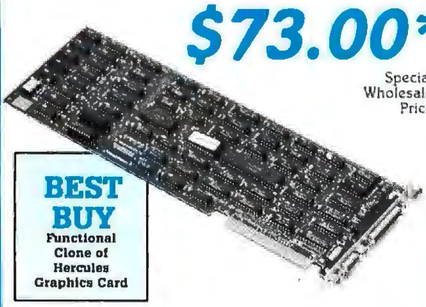
Eight times the speed of a 300 BPS unit makes this 2400 Baud modem truly affordable, and we've made it truly compatible. With auto answer, auto dial, auto redial, and a built-in speaker with volume control. Retail Price $346.25

Special Wholesale Price **$259.00***

THE™ H720 MONO

This monochrome graphics card with printer port features 100% IBM™ and Hercules™ compatibility, at a fraction of their prices. Retail Price $91.25

$73.00* Special Wholesale Price



BEST BUY Functional Clone of Hercules Graphics Card

A $120.00 VALUE!

PC + STARTER KIT

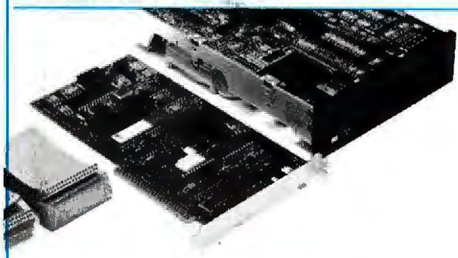
ADAM OSBORN'S PAPERBACK SOFTWARE

- Paperback Writer
- Paperback Speller
- Numberworks Spreadsheet

All three for only...

$32.00*

With the purchase of a PC +



THE™ 20MB HARD DISK

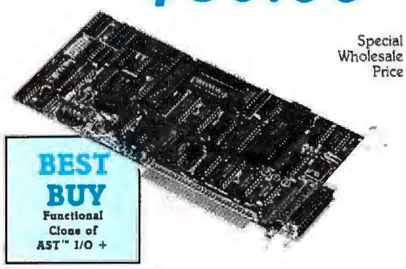
Featuring a half-height IBM™ PC compatible internal 20MB disk drive, controller card, connecting cables and installation manual. Retail $458.75

Special Wholesale Price **$357.00***

THE™ MULTI I/O

Feature for feature, **THE**™ matches the AST™ I/O + and comes up a winner, with serial/parallel and game ports standard, (a second serial port optional), a clock/calendar, RAMdisk and print spooler. Retail Price $75.00

$60.00* Special Wholesale Price



BEST BUY Functional Clone of AST™ I/O +

THE™ EGA PLUS

With 256K standard this 100% IBM™ EGA and Hercules Graphics compatible card is ideal for high resolution graphics display of Microsoft™ Windows, Lotus™ and AutoCAD™. Retail Price $283.75*

Special Wholesale Price **$220.00***

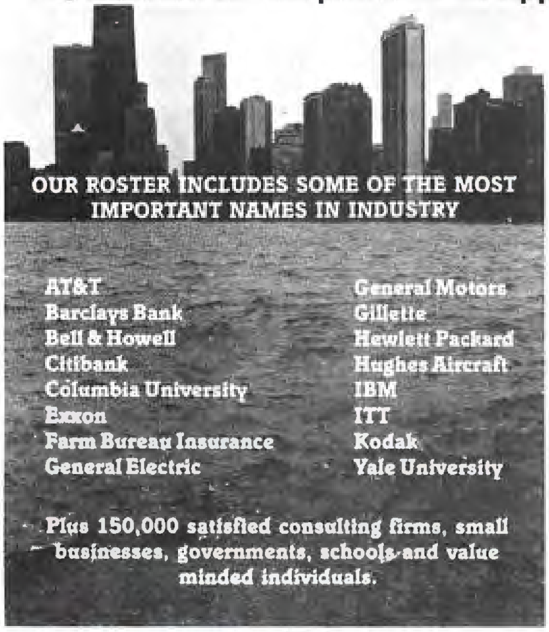
*All Items Subject to Availability—Prices Subject to Change
TM—Registered Trademarks of IBM/Ast Research/Microsoft/Hercules/Hayes

Thompson, Harriman and Edwards Computer

PC NETWORK

BUY HARDWARE AND SOFTWARE AT WHOLESALE +8%, AND GET 14-30 DAY SOFTWARE RENTALS†...

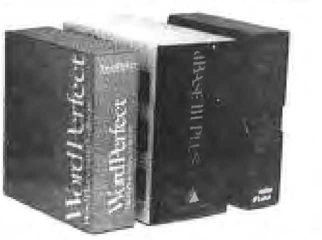
Listed below are just a few of the over 30,000 products available at our EVERYDAY LOW PRICES!
The Network **carries products for Apple, IBM, CP/M and most other popular computer families.**



PRINTERS

Epson
FX 286
132CPS/200CPS Letter Quality Printer
$440.00*

FX-85	*80COL/160CPS*	355.00*
LQ-800	*180COL/60CPS Par*	519.00*
LQ-1000	*180COL/60CPS Par/Ser NLQ*	699.00*



TOP BUSINESS PROGRAMS
FOR THE IBM™ PC/XT/AT

LOTUS 1-2-3
$280.00*
ASHTON TATE DBASE III
$365.00*
SATELITE SOFTWARE WORD PERFECT
$180.00*

GAMES & EDUCATIONAL SOFTWARE
(Please add $1 shipping and handling for each title ordered from below.)

	Wholesale
Bluechip *Millionaire, Oil Baron or Tycoon*	$28.99*
Broderbund *Ancient Art of War*	22.75*
Broderbund *Lode Runner*	17.25*
Comprehensive *Intro to Personal Computing*	30.00*
Davidson *Math Blaster, Word Attack!*	21.50*
Epyx *Winter Games/Temple Apshai/Summer Games II*	20.00*
Hayden *Sargon III*	24.75*
Individual *The Instructor*	24.50*
▶ **Infocom** *Zork I or Witness, Trinity*	20.00*
▶ **Infocom** *Deadline or Suspended*	24.00*
Microsoft *Flight Simulator*	$27.00*
Mindscape *The Mist, First Blood Pt. 2/or View to a Kill*	21.00*
Mouse Systems *PC Paint-Turn your PC into a Color Macintosh!*	$59.95*
Pinpoint Publishing *Micro Cookbook*	21.00*
Scarborough *MasterType*	19.75*
Sierra On-Line *King's Quest II*	24.75*
Spectrum Holobyte *GATO*	23.99*
▶ **Spinnaker** *Alphabet Zoo, Kinder Comp Story Machine, Face Maker, Hey Diddle Diddle, Rhymes & Riddles*	15.25*
▶ **Spinnaker** *Delta Drawing or Most Amazing Thing*	20.75*
Sublogic *Night Mission Pinball*	20.00*
Sublogic *Jet*	27.50*
Timeworks *Evelyn Wood Dynamic Reader*	41.00*

BUSINESS SOFTWARE
(Please add $2.50 shipping and handling for each title ordered from below.)

ANSA *Paradox*	$445.00*
▶ **Ashton-Tate** *DBase III+*	365.00*
▶ **Ashton-Tate** *Framework II*	365.00*
Borland *Reflex*	54.00*
Borland *SideKick (Protected)*	26.00*
Borland *Turbo Pascal*	35.00*
Borland *Turbo Pascal 8087*	53.00*
Borland *SuperKey*	34.50*
Breakthrough Software *Time Line 2.0*	203.00*
Central Point *Copy II PC*	16.25*
Computer Associates *Supercalc IV*	239.95*
Computer Associates *Super Project Plus*	239.95*
Digital Research *Gem Draw*	73.00*
Enertronics *Energraphics*	145.00*
FGS *Fastback*	85.00*
Funk Software *Sideways*	32.00*
Harvard *Harvard Total Project Manager*	245.00*
Hayes *Smartcom II—New VT 100 Emulator*	68.00*
Human Edge *The Management Edge*	45.00*
▶ **Human Edge** *The Sales Edge or Mind Prober*	24.00*
Infocom *Cornerstone*	70.50*
Lifetree *Volkswriter III*	126.00*
Living Videotext *Think Tank*	87.00*
Lotus *Lotus 1-2-3*	280.00*
MicroPro *Wordstar 2000*	$209.00*
MicroPro *Wordstar 2000+*	259.00*
MicroRim *R Base 5000*	215.00*
MicroRim *R Base System V*	339.00*
Microsoft *C Compiler*	215.00*
▶ **Microsoft** *Word—Latest Version 3.0*	202.00*
Microsoft *Multiplan*	98.00*
Migent *Ability*	55.00*
Monogram *Dollars & Sense*	85.00*
Mosaic *The Twin*	75.50*
▶ **Multimate** *Multimate (Latest Version)*	180.00*
Multimate *Multimate Advantage*	270.00*
▶ **Meca** *Managing Your Money*	92.00*
Nantucket *Clipper DBase III Compiler*	350.00*
Norton *Commander*	36.00*
Norton *Norton Utilities 3.1*	43.00*
Open Systems *A/R INV G/L A/P Ver.3.X*	ea.395.00*
Power Base *Power Base*	200.00*
Rosesoft *ProKey Version 4*	65.00*
Satellite Software *Word Perfect*	180.00*
Software Group *Enable*	293.00*
Software Publishing *PFS: File, Write, Graph*	68.00*
Software Publishing *PFS: Report*	64.00*

HARDWARE
(Please add shipping and handling charges found in italics next to price.)

COMPLETE SYSTEMS

Apple *Macintosh/Apple IIc/IIe*	CALL	
AT&T *6300 PC CPU2 640K/2 Floppy/1 Serial/1 Parallel/Mono Monitor*	$1,775.00*	*(38.34)*
COMPAQ *Hard Disk20MB Portable/1 Floppy w/256K*	1,910.00*	*(41.26)*
COMPAQ *DeskPro Hard Disk 20MB w/256K/ 1 Floppy*	1,807.00*	*(39.03)*
▶ **IBM** *PC Base System 2 DS/DD/FDC w/256K*	1,025.00*	*(22 14)*
▶ **IBM** *PC Professional 20MB Hard Disk*	1,337.00*	*(28.88)*
IBM *PC/AT All Configs*	CALL	
IBM *PC/AT Base Unit 1.2MB Floppy w/256K*	2,875.00*	*(60.37)*

DISK DRIVES & CONTROLLERS

IOmega *Bernoulli Box/Dual MP Drives*	$1,710.00*	*(38.94)*
▶ **PC Network** *10MB Internal 1/2 Height Autoboot Drive: by Seagate or Shugart*	299.00*	*(6.46)*
PC Network *10MB Tape Backup Same Unit used in Compaq's DeskPro*	340.00*	*(9.48)*
PC Network *20MB Tape Backup Half Height w/controller*	495.00*	*(10.69)*
▶ **PC Network** *Half Height DS/DD Drive*	75.00*	*(1.68)*
Tandon *20MB Hard Card*	435.00*	*(9.40)*
▶ **THE™20MB Hard Disk** *Hard Disk w/20MB Tape Backup - Combo Kit*	735.00*	*(15.88)*
▶ **THE™20MB** *20MB Internal 1/2 Height Autoboot Drives: by Seagate or Microscience*	357.00*	*(7.93)*

MEMORY CHIPS
All chips guaranteed for life.

▶ **64K Memory Upgrade Kits** *(9 chips)*	$7.75*	*(1.00)*
▶ **64K Dynamic Ram Chips** *(Each)*	1.00*	*(1.00)*
▶ **256K Dynamic Ram Chips** *(Each)*	2.80*	*(1.00)*
▶ **128K AT Mother Board Chips** *(Each)*	5.60*	*(1.00)*
▶ **Intel** *8087 Co-Processor*	99.50*	*(2.15)*

MODEMS

▶ **Hayes** *Smartmodem 2400*	$525.00*	*(5.00)*
Hayes *Smartmodem 1200B with new Smartcom II VT 100 Emulator*	315.00*	*(2.50)*
Hayes *Smartmodem 1200*	340.00*	*(5.00)*
Prometheus *Promodem 1200*	247.00*	*(6.00)*
U.S. Robotics *Courier 2400BPS Modem*	345.00*	*(5.00)*
U.S. Robotics *Password*	172.00*	*(2.50)*
ALL THE™ MODEMS ARE 100% HAYES COMPATIBLE		
▶ **THE™1200 Com External**	119.00*	*(2.58)*
▶ **THE™2400 Com External**	259.00*	*(5 54)*
▶ **THE™1200 Com Internal** *w/Free PC Talk III*	109.00*	*(2.50)*

MONITORS

Amdek *Video 310A Amber TTL Monitor*	$130.00*	*(3.00)*
Amdek *Color 600 High Res RGB Monitor*	365.00*	*(7.88)*
Amdek *Color 722 NEW! Enhanced*	424.00*	*(9.16)*
Magnavox *Amber IBM Type TTL Monitor*	79.00*	*(5.00)*
Magnavox *14" RGB Color Monitor 640 x 240*	275.00*	*(5.00)*
NEC *JC-1401 Multisync RGB ColorMonitor*	480.00*	*(10.37)*
Princeton *HX-12 RGB Monitor*	369.50*	*(7.98)*
Princeton *HX-12E RGA or EGA Monitor*	430.00*	*(9.29)*
Princeton *MAX-12E Supports Either Mono or RGB Cards*	138.00*	*(5.00)*
Taxan *630 High Res RGB Monitor*	383.00*	*(8.27)*
Taxan *640 Highest Res (720x400) Currently Available—Works with Persyst BOB Card*	439.00*	*(8.27)*

***PC NETWORK Members pay just 8% above the wholesale price, plus shipping.** All prices reflect a 3% cash discount. Minimum shipping $2.50 per order. International orders call for shipping & handling charges. Money Orders, personal and company checks please allow 10 working days to clear. All prices subject to change.

MULTIFUNCTION CARDS

AST *SixPakPremium™ w/0K*	$315.00*	*(2.50)*
▶ **AST** *SixPakPlus™ with 64K*	139.00*	*(2.50)*
AST *I/O Plus II™*	120.00*	*(2.50)*
▶ **AST** *Advantage for AT*	325.00*	*(2.50)*
Orchid *Conquest*	220.00*	*(2.50)*
Orchid *Tiny Turbo 286*	420.00*	*(2.50)*
Tecmar *Captain Multifunction Card 0/K*	109.00*	*(2.50)*
▶ **THE™576K Memory +** *Board w/OK*	35.00*	*(2.50)*
▶ **THE™Multi 384** *A functional clone of AST's™ SixPakPlus.™ Includes up to 384K of expansion memory, 1 serial, 1 parallel, 1 game port, a clock/calendar and 3 software standard*	67.00*	*(1 62)*
▶ **THE™Multi I/O** *I/O Plus Clone Ser/Par/Game/Clock Standard*	60.00*	*(2.50)*

PRINTERS

Citizen *MSP-10 NEW! 160CPS/80COL/Fric + Trac*	$259.00*	*(4.86)*
Citizen *MSP-15 NEW! 160CPS/132COL/Fric + Trac*	340.00*	*(7.34)*
Citizen *MSP-25 NEW! 200CPS/132COL/Fric + Trac*	435.00*	*(9.40)*
Citizen *Premiere 35 NEW! 35CPS Daisywheel/132COL/Fric + Trac*	420.00*	*(9.07)*
Citizen *120-D 120/25CPS 10" w/Tractor*	169.00*	*(3.65)*
Epson *LX-80 100CPS 80COL LQ Model*	225.00*	*(4.86)*
Epson *FX-85*	355.00*	*(7.45)*
▶ **Epson** *FX-286 NEW! 200CPS/40CPS NLQ*	440.00*	*(10.91)*
▶ **Epson** *LQ-1000 NEW! 180CPS/60CPS NLQ*	699.00*	*(15.01)*
NEC *2050 20CPS Letter Quality Printer*	570.00*	*(11.88)*
NEC *3550 33CPS Letter Quality Printer*	690.00*	*(14.90)*
NEC *8850 55CPS Printer*	949.00*	*(20.50)*
NEC *Pin Writer P-7 180/65CPS/132COL*	535.00*	*(11.56)*
▶ **Okidata** *ML 182 NEW! 120CPS LQ Model/Fric/IBM Graphics and more*	188.00*	*(4.06)*
▶ **Okidata** *ML 192 NEW! 160CPS LQ Sleek Design/Fric/IBM Graphics*	315.00*	*(6.80)*
▶ **Okidata** *ML 193 NEW! 160CPS LQ Sleek Design/Fric/IBM Graphics*	425.00*	*(9.18)*
Panasonic *KX-P1091 120/75/22CPS*	220.00*	*(4.75)*
Panasonic *KX-P1092 180/CPS w/Tractor*	329.00*	*(7.11)*
Panasonic *KX-P1592 180/CPS 15" w/Trac*	406.00*	*(8.77)*
Toshiba *NEW! P341 210CPS/132COL*	699.00*	*(15.01)*
Toshiba *351 Par/Ser 288CPS/132COL*	959.00*	*(20.71)*

VIDEO CARDS

Hercules *Color Card w/Parallel Port*	$126.00*	*(2.50)*
Hercules *Monochrome Graphics Cards*	162.00*	*(2.50)*
Paradise *Autoswitch EGA*	336.00*	*(2.50)*
Orchid *Turbo EGA Card*	610.00*	*(2.50)*
Quadram *Quad EGA +*	323.00*	*(2.50)*
▶ **THE™EGA Plus** *w/256K and printer port*	227.00*	*(2.50)*
▶ **THE™H720** *Mono Graphics w/parallel port 100% Hercules Compatible!*	73.00*	*(2.50)*
▶ **THE™Color Card** *100% IBM Compatible*	60.00*	*(2.50)*

ACCESSORIES AND SUPPLIES

▶ **Brand Name DS/DD Diskettes** *Guaranteed for Life! Not Generic*	$7.95*	*(1.00)*
▶ **DS/DD Bulk Pack Special** *Packaged in 50 with sleeves and labels Guaranteed for Life!!*	ea. .54*	
▶ **PC Network** *Replacement 130 Watt IBM-PC Power Supply—Gives you PC (Old or New) the same capacity as an XT. Good for add in tape drives (without need for a piggyback unit) and large capacity disk drives.*	79.00*	*(2.70)*
SMA *PCDocumate:Keyboard Templates for Lotus/DBase/Multimate and others*	ea. 9.99*	*(1.00)*
Sony *DS/DD Diskettes*	15.95*	*(1.00)*
WP Printer Paper *2600 Sheets Microfine Perfs (invisible when torn)*	17.00*	*(10.00)*

†**RENT BEFORE YOU BUY**—Members are eligible to join the NETWORK'S Business and Game Software Rental Libraries to evaluate products for a full 14 (Regular) or 30 (V.I.P.) days to see if they meet your needs. And the NETWORK'S rental charges are far less than other software rental services—JUST 20% OF THE MEMBER WHOLESALE PRICE. **We feature over 1,000 available titles in IBM/Apple/Mac and CP/M.**

Prices subject to change without notice.

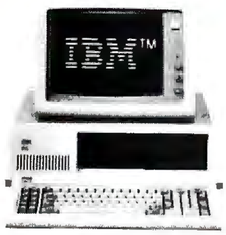
COMPLETE SYSTEMS
(Optional local warranty service available in most major metro areas)

PC NETWORK

IBM PC BASE SYSTEM
IBM PC w/256K
Floppy Drive Controller
2 Double Sided Double Density Disk Drives
Mix and Match with your Favorite Monitor and Printer!

$1,025.00*

CUSTOM CONFIGURATIONS WELCOME



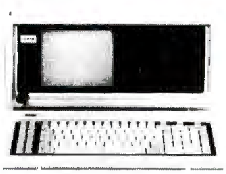
IBM PC HARD DISK SYSTEM
IBM PC w/256K
Floppy Drive Controller
1 Double Sided Double Density Disk Drive
Half Height Hard Disk

w/20MB

$1,337.00*

CALL FOR LATEST IBM ANNOUNCEMENTS

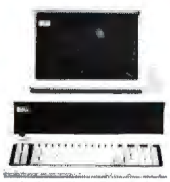
PORTABLE HARD DISK SYSTEM
256K/1 Floppy/20MB Hard Disk



COMPAQ™

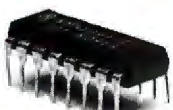
$1,910.00*

DESKPRO SYSTEM
256K CPU/1 Floppy/Hard Disk



$1,999.00* w/30MB
$1,807.00* w/20MB

64K IBM PC EXPANSION KIT
Lifetime Guarantee!
Set of 9
150/200ns



$7.75*

256K	Set of 9	$26.10*
128K	For the AT	35.10*

INTEL™ Math Co-Processors

8087	5Mhz	$99.50*
8087	8Mhz	161.00*
80287	5Mhz/8Mhz	197.00*

DISKETTES
Guaranteed for Life!
Brand Name Diskettes available in boxes of 10, or in bulk packs of 50

5¼" DS/DD
$7.95* Bx. 10
Bx. 50 ea. .54*

5¼" SS/DD
$6.95* Bx. 10
Box 50 ea. .47*

New from THE™

THE™ 20MB HARD DISK CARD
$375.00*

1/2 Height 360K DS/DD Floppy Disk Drive
$75.00*

THE™ 20 + 20
20MB Hard Disk
20MB Tape Backup

$735.00*

THE™ 20MB Hard Disk Kit
$357.00*

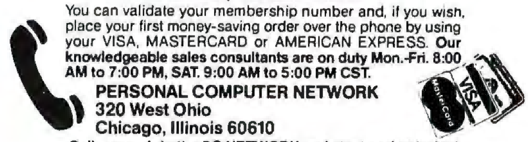
***Members pay 8% above this wholesale price plus shipping.**
CALL TOLL FREE 1-800-621-S A V E (orders and memberships)
In Illinois call (312) 280-0002 Validation Code: B3Y6
Customer Service and Order Status (312) 280-1567

TM & R—Registered Trademarks of IBM/COMPAQ/APPLE/AST Research/Lotus/Multimate/IOmega/Hayes/Hercules

SEE WHY OVER 150,000 HAVE JOINED MAKING US THE NATION'S #1 SOURCE FOR EVERYTHING IN COMPUTING

- ***Our 600 Page Wholesale Catalog***
  Over 30,000 products priced at Wholesale + 8%. Anything you will ever need at a Consistent low price...
- ***Quarterly Catalog Updates***
  Your Catalog is never obsolete! Keep on top of the newest products and latest price changes.
- ***The Printout***
  Our newsletter gives you fantastic specials alog with unbiased analysis of new products and industry trends.
- ***10 Day Returns on any Hardware!!***
  If you don't like any hardware product—for any reason—return it for a refund.
- ***1000 + Title Rental Library***
  14 to 30 day rentals on over 1000 different titles. Try before you buy!!
- ***Size, Strength and Stability***
  The Network has over 150 employees, 45,00 square feet of office and warehouse space, inventory valued in excess of $15,000,000 and is ranked the largest computer product supplier in the nation! Our commitment is to serve our customers and our 90% repeat business rate is proof!!
- ***OnLine™ Bulletin Board***
  Download "Freeware" from the nation's largest single concentration of Public Domain software available! Get tied into the nation's largest technical information network! Place orders, get tech support or contact customer service from the Network's OnLine,™ not just a bulletin board but a complete customer information network.

CALL TOLL FREE 1-800-621-S-A-V-E
(Orders—Membership and Advice!)
In Illinois call (312) 280-0002
Your Membership Validation Number **B3Y6**



You can validate your membership number and, if you wish, place your first money-saving order over the phone by using your VISA, MASTERCARD or AMERICAN EXPRESS. **Our knowledgeable sales consultants are on duty Mon.-Fri. 8:00 AM to 7:00 PM, SAT. 9:00 AM to 5:00 PM CST.**

PERSONAL COMPUTER NETWORK
320 West Ohio
Chicago, Illinois 60610

Call now...Join the PC NETWORK and start saving today!
Customer Service and Order Status (312) 280-1567
8:30 AM to 4:30 PM, Mon.-Fri. CST

PC NETWORK • MEMBERSHIP APPLICATION
YES! Please enroll me as a member in the PC Network™ and send my catalog featuring thousands of computer products, all at just 8% above DEALER WHOLESALE PRICES. I will also periodically receive "THE PRINTOUT," a special up-date on merchandise at prices BELOW even those in my wholesale catalog, and all the other exclusive money-saving services available to Members.

3Y6 ☐

I am under no obligation to buy anything. My complete satisfaction is guaranteed. Please check (✓) all boxes that apply:

Basic Membership
with optional 14 Day Evaluation.
☐ One-year membership for $8
☐ Two-year membership for $15 (Save $1)
☐ Business Software Evaluation Library for $25 add'l. per year—with 14-day returns
☐ Games Software Evaluation Library for $10 add'l. per year

Special V.I.P. Membership*
with optional 30 Day Evaluation.
☐ One-year membership for $15
☐ Two-year membership for $25 (Save $5)
☐ BOTH Business and Game Software Evaluation Libraries for $30 add'l. per year—with 30 day returns
*V.I.P. members may receive advance notice on limited quantity merchandise specials

☐ Bill my credit card: ☐ VISA ☐ MasterCard ☐ American Express

Account Number: ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Exp. Date ________ mo. year

☐ Check or money order enclosed for $________

Name________

Address________

City________ State________ Zip________

Telephone (   )________

My computer(s) is: ☐ IBM PC ☐ IBM-XT ☐ IBM—AT ☐ Apple II ☐ Macintosh ☐ Other________

Signature________
(Signature required to validate membership)

This Breakthrough Just Broke Through. Again.

When the Leading Edge® Model "D"™ Personal Computer was first introduced, it stunned the world by being the first and only IBM®-compatible to come with more standard features than the IBM PC at about half the price.

It's been over a year since then. We think it's time to stun the world again.

Since the Model "D" already comes with dual-drives, high-resolution monochrome monitor, monochrome and color graphics output, parallel and serial ports, and free word processing software, there was only one thing to add: memory. So we beefed it up to **512K** (expandable to 768K on the motherboard).

But there was something to drop. The price. We reduced it to **$1295.**

Then we added a 30MB fixed-disk version of the Model "D" and priced it at **$1995,** including word processing with spelling correction and a Lotus 1-2-3® look-alike spreadsheet.

And to make sure nobody would be left unstunned this time around, we created the Leading Edge Information System™ — a Model "D" Personal Computer with a built-in 1200 baud Hayes®-compatible modem and communications software — and priced it at an incredible **$1495.** It also comes with word processing with spelling correction and a Lotus 1-2-3 look-alike spreadsheet.

Of course, every system comes with a 15 month warranty, a full-year longer than IBM's, and easy-to-use instruction manuals.

So if you missed it last time, don't this time. Breakthroughs like this don't happen too often. Call **1-800-USA-LEAD** for more information and a dealer near you.

Leading Edge Hardware Products, Inc.
225 Turnpike Street, Canton, MA 02021
1-800-USA-LEAD in Massachusetts (617) 828-8150

LEADING EDGE

THE IBM PC CONVERTIBLE

A portable that offers many different options and up to 512K bytes of memory

BY G. MICHAEL VOSE

Editor's note: The following is a BYTE product description. It is not a review. We provide an advance look at this product because we feel that it is significant. We plan to provide a complete review in a subsequent issue.

IBM'S PC Convertible looks like a portable computer complete with a movable handle but feels more like a transportable computer, such as the Compaq. Packed in its optional carrying case, the PC Convertible and battery pack weigh over 15 pounds, a little over one-half the weight of a Compaq. The machine offers up to 512K bytes of memory, 1.4 megabytes of on-line disk storage, and a readable 80-column by 25-line liquid-crystal display, as well as IBM PC compatibility.

The Convertible's many options include a parallel/serial adapter, a portable printer, a monochrome and color display adapter, a 9-inch monochrome or 13-inch color monitor on its own pedestal, 128K-byte memory upgrades, and an internal 110/300/1200-baud modem. These options can increase the Convertible's base price from $1995 to as much as $3000.

IBM supplies some bundled software with the Convertible, including several notebook-like applications for note-taking, phone-number files, and the like. However, the Convertible uses 3½-inch disks, so obtaining other software for the machine presents a problem.

MACHINE INTERNALS

A CMOS version of Intel's 8088 microprocessor powers the PC Convertible. Driven by a 33 percent duty-cycle crystal similar to the IBM PC's, this 14.31818-MHz oscillator's frequency is divided by three to drive the 80C88 at 4.77 MHz. The 80C88's internal circuitry is static, eliminating the need for refresh clock cycles for its on-chip registers, counters, and latches. The 80C88's clock-cycle time is 210 nanoseconds and the Convertible's main-memory cycle is 840 ns, while the I/O cycle is 1.05 microseconds.

The Convertible uses a system timer compatible with the Intel 8253 Programmable Interval Timer, and it provides two programmable channels. The base timing resolution for a timer channel is 840 ns. The interrupt controller on the Convertible duplicates the functions of the Intel 8259 Programmable Interrupt Controller. This interrupt controller provides nine interrupt levels, one of which is nonmaskable. The system clock, interrupt controller, and other devices (keyboard controller, audio controller, and I/O controller) reside on the motherboard in a VLSI gate-array chip. Another gate array houses the disk and printer interface circuitry, time-of-day clock, and system timer. All the Convertible's circuitry, including a total of five gate arrays, resides in just 19 surface-mounted chips on the machine's motherboard (see photo 1).

The Convertible's standard 256K bytes of nonparity-checked RAM resides on two small removable circuit boards located just under the machine's keyboard. To add an additional 256K bytes of RAM to the machine, you plug two similar 128K-byte boards alongside the existing ones for a total of four boards. The Convertible's static RAM features 256K by 1-bit chips. [*Editor's note: Peripheral manufacturer* STB *Systems has already announced a* 384K-*byte*

(continued)

G. Michael Vose is a senior technical editor for BYTE *(One Phoenix Mill Lane, Peterborough,* NH 03458).

PHOTOGRAPHED BY PAUL AVIS

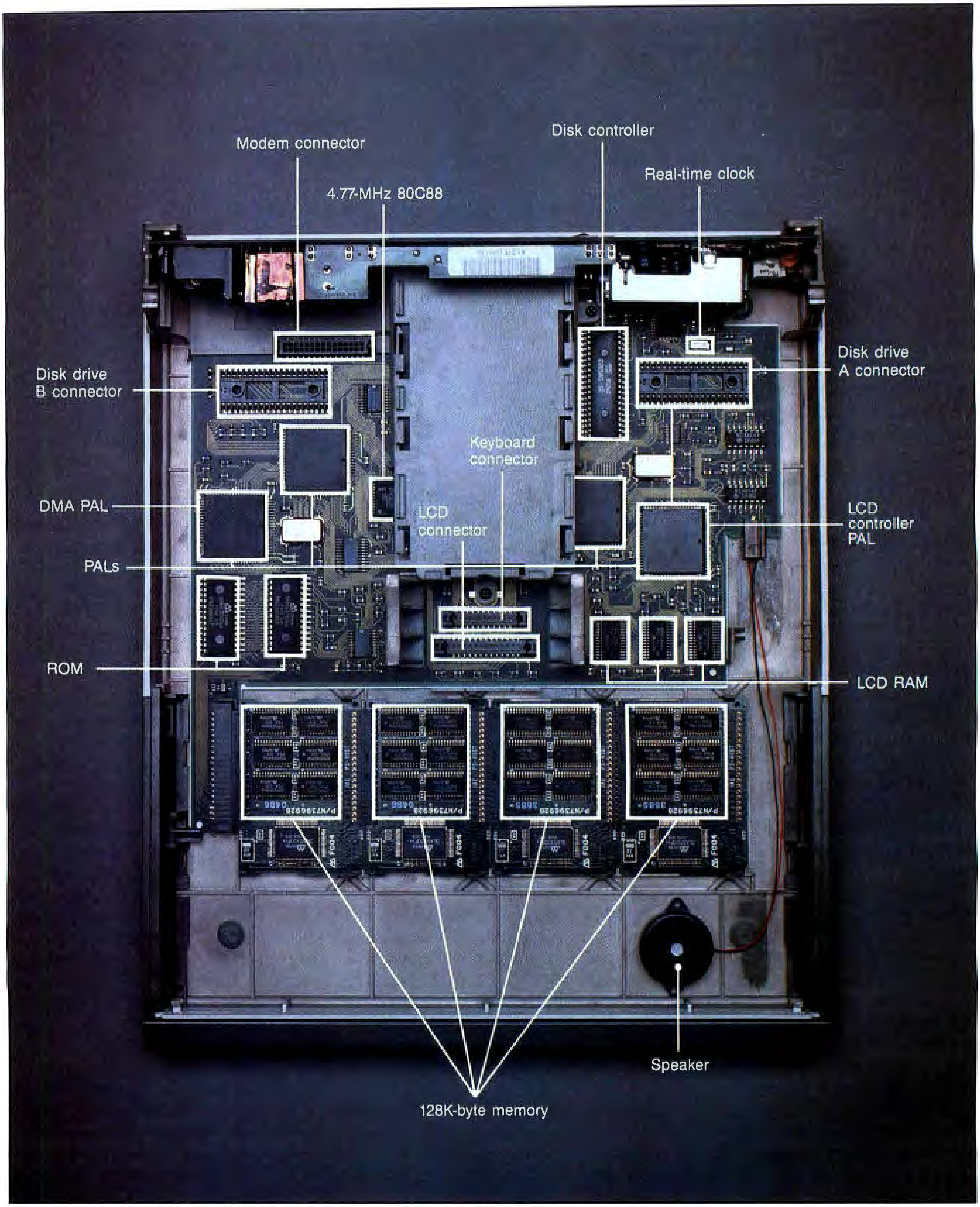


Photo 1: *The sparsely populated PC Convertible main circuit board and 128K-byte memory boards.*

expansion board that lets the Convertible use a full 640K *bytes of* RAM; *see the* What's *New item on page* 14.]

Two 32K-byte ROMs contain the PC Convertible's on-board system software. The machine's resident BASIC uses 32K bytes and the POST (power-on self-test) and BIOS routines occupy the other 32K bytes in separate 24K and 8K segments.

The two 3½-inch disk drives in the PC Convertible offer formatted storage of 720K bytes per disk. The disks used by these drives store data at 135 tracks per inch (80 tracks per side) and can be accessed by the drives at 6 milliseconds track to track. The maximum data-transfer rate is 250K bytes per second. The NEC μPD765 floppy-disk controller manages the I/O control of the disk drives.

The Convertible's DMA (direct memory access) controller is compatible with the Intel 8237 programmable DMA controller and regulates the movement of data from memory to I/O devices without involving the microprocessor. Since the Convertible's static RAM requires no refreshing, the DMA controller does not support refresh; therefore, it supports three DMA channels instead of four (as in the IBM PC) and uses five clock cycles (1.05 μs per byte) for every DMA data transfer. The 16-bit control registers in the DMA controller require the use of a page register to develop the 20-bit addresses possible within the Convertible's 1-megabyte addressable memory. The DMA controller does not support memory-to-memory data transfer, cascading, or automatic initialization.

To beep error messages or to play music, the Convertible uses a small speaker on the motherboard controlled by an audio controller that receives its signals from the system timer and the I/O controller.

The LCD measures 3¾ inches high by 10¾ inches wide (see photo 2). It displays 25 lines of 80 characters at a resolution of 640 by 200 pixels. The LCD disconnects from the Convertible's system unit via a 20-pin male/female edge connector for the convenience of users who elect to use an optional monochrome or color display. The LCD is not backlit but is adjustable for accommodating ambient room light. The aspect ratio of the LCD differs from standard CRT devices by about 20 percent.

The LCD controller uses an interface similar to that of the Motorola 6845 CRT controller, and thus allows the operation of programs written to use the IBM monochrome display adapter or the IBM color/graphics display adapter. The LCD controller has its own 16K bytes of video RAM and uses either the main font (the IBM PC character set) or an alternate font to display up to 256 characters per font.

The keyboard on the Convertible is similar at first glance to the PCjr keyboard—function keys in a row across the top, a small function key on the bottom left, and a cluster of cursor keys on the bottom right. However, the similarity to the PCjr ends here. The Convertible has 78 keys on a keyboard with a surprisingly solid feel for such a small device (see photo 3). The Convertible keyboard generates all the scan codes recognized by the IBM PC BIOS and includes an Alternate key and a Control key.

Supplying Power

There are four ways that a PC Convertible can get electricity: an AC adapter, a battery pack, a battery charger, or an automobile adapter that plugs into an automobile cigarette lighter. You can use the battery charger only to charge the batteries, not to run the computer. The AC adapter and the automobile adapter will run the Convertible and charge the battery pack at the same time. The AC adapter uses 90- to 265-volt, 50/60-Hz electrical-source power.

The battery pack supplied with the PC Convertible has eight nickel-cadmium cells and, when fully charged, can power the machine for 6 to 10 hours depending on the installed memory and the use of disk drives and attached peripherals. If the battery-pack power gets low, an icon that looks like a flashlight battery blinks in the upper right corner of the screen to warn you to correct the problem. Recharging the Convertible's battery is a 12- to 14-hour process after you see a low-power warning.

To preserve power, the Convertible will power down if there is no disk or keyboard activity for two minutes. During its power-down operation, it saves a variety of information about the state of the machine at the time of shutdown.

Optional Hardware

Since the PC Convertible has no external connectors for serial or parallel printers, the serial/parallel adapter is an important option. IBM gets away with making these ordinarily indispensable outputs optional by offering

(continued)

Photo 2: *The* PC *Convertible's* LCD *screen.*

a low-power, thermal-transfer, serial dot-matrix printer that attaches to the Convertible's system bus. However, many users will elect to use their own printers rather than buy another one, and those users must buy the adapter. The serial/parallel adapter attaches to the back of the Convertible, connecting to the 72-pin system bus. The system bus is brought out on the other side of the adapter so that you can still attach other peripherals to the computer.

The serial/parallel adapter differs slightly from the IBM PC's in the way that it implements the functions of the PC's asynchronous communications adapter. The Convertible's BIOS contains routines that allow control of local power on the serial adapter. Additionally, the maximum data-transfer rate of the serial/parallel adapter's serial port is 9600 baud.

Another important Convertible option is the internal modem. This modem was not available for my use, so I have to take IBM's word that it actually exists. According to IBM, this modem is an auto-answer, auto-dial device that supports tone or pulse dialing. It supports full-duplex operation at 110, 300, or 1200 baud, conforming to the frequency specifications of the Bell 103A or 212A standards. The modem is not Hayes-compatible.

The PC Convertible printer runs off either the computer's battery pack or separate AC power. Weighing in at 3½ pounds, it prints unidirectionally at 40 characters per second in one pitch (10). The printer uses a 24-pin, thermal-transfer, nonimpact, dot-matrix print head that requires either thermal paper, a thermal-transfer ribbon and thermal-transfer paper, or high-quality (20-pound or less) plain paper. Supporting condensed, double-wide, and emphasized print modes, the printer will underline text and can print sub- and superscripts.

Photo 3: *The PC Convertible case and keyboard.*

OTHER OPTIONS

For LCD haters, IBM offers several display options including monochrome, color, and TV (RF modulator) adapters. IBM also sells a 9-inch monochrome or 13-inch color monitor attached to its own tiltable pedestal for use with the Convertible. The distance between the bottom of the pedestal and the CRT is sufficient to slide the Convertible neatly between the two. The monochrome and color monitors run only on AC power, so you'll be stuck with the LCD at the beach unless you have a long extension cord.

The CRT display adapter supports these direct-drive or composite monitors: the PC Convertible monochrome and color displays, the IBM PCjr color display, the IBM PC color display, or a standard TV set (using the IBM PCjr connector for TV). You can also use any IBM-compatible monitor with the Convertible. Since you'll have to be close to an AC outlet to use a monitor, IBM specifies that you must use the AC adapter to operate the Convertible when the CRT display adapter is connected to the computer. I used the machine without the AC adapter with no apparent ill effects, but I'm sure IBM advises against this practice to avoid quickly draining the battery.

My favorite option for the PC Convertible is the carrying case. It is made of canvas with lots of foam padding, and its design reminds me of a military flak jacket (its color is a conservative gray.) This carrying case will lend an air of strategic importance to your portable computing.

PC CONVERTIBLE SOFTWARE

The PC Convertible uses PC-DOS 3.2 to manage its hardware resources. Since it is a fully IBM PC–compatible computer (assuming, as IBM insists, that applications software makes BIOS and DOS calls only), the Convertible will run any IBM PC software.

However, the availability of your favorite software on 3½-inch disks could be a problem. At BYTE we use the Manzana MDP3A disk drive with accessory package software ($495 from Manzana, 935 Camino del Sur, Isla Vista, CA 93117, (805) 968-1387) to transfer software to the 3½-inch format for test purposes. I copied my favorite word-processing program onto a formatted 3½-inch PC Convertible disk and used it to write this article. I reformatted the disk after I copied the article back to a 5¼-inch disk.

Concurrent with the Convertible's debut, IBM announced two versions of a 3½-inch stand-alone disk drive for use with all the computers in the PC family. The availability of this drive (which requires PC-DOS 3.2) makes porting non-copy-protected software to the 3½-inch format less of a problem.

If you cannot acquire software on 3½-inch disks, you can still use a PC Convertible. The Convertible comes with an applications disk that contains a note writer, a scheduler, a phone list, a calculator, and some help files and utilities.

The note writer lets you create, save, and print notes up to six pages (12,000 characters) in length. You can adjust the left and right margins, do block moves or copies, and display a directory of existing notes. You cannot adjust line spacing or delete blocks.

The scheduler lets you set up a calendar of daily, weekly, or monthly appointments. You can set an audible alarm and mark appointments with an asterisk to remind you of important dates. You can even print out your weekly or daily schedule. You can set up the schedule profile with any start and end time and specify the time increment from 5 to 60 minutes.

The phone list lets you create and store name, address, and phone-number files. The program will even auto-dial numbers in the file if you have an internal or external modem attached to your Convertible.

The calculator adds, subtracts, multiplies, divides, figures percentages, and prints a log of your calculator operations.

The utilities supplied with the Convertible, called tools, mostly assign standard DOS functions such as FORMAT, ERASE, and DISKCOPY to the function keys for single-keystroke invocation.

CONCLUSIONS

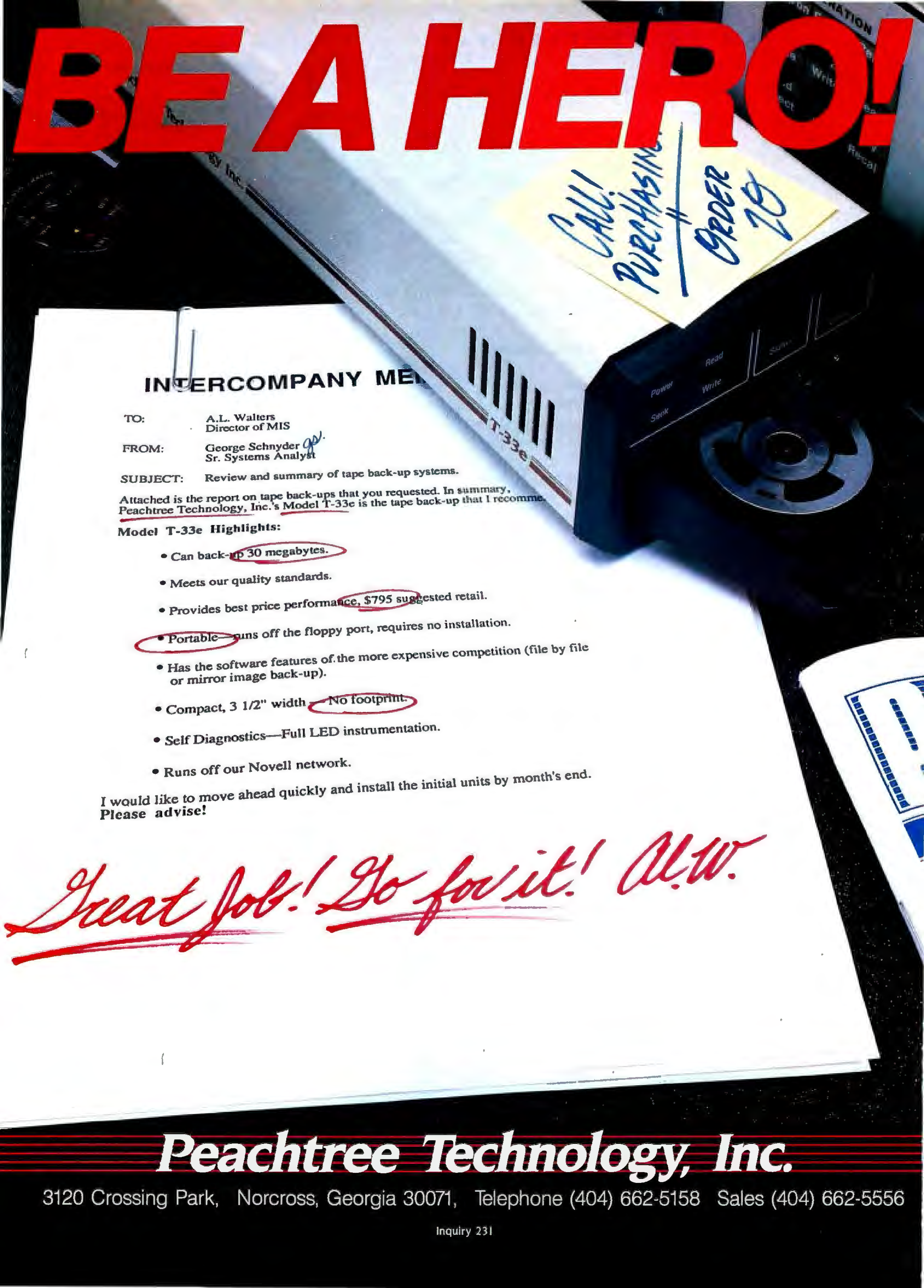
In BYTE, product descriptions give readers a first look at hardware or software prior to an in-depth product review. Therefore, I will make no judgments about the PC Convertible other than to say that it was easy to set up, add optional equipment to, and use to write this article. The machine worked without any failures during the few days I used it. I found the LCD to be readable for short stretches; however, I used the monochrome display for my longer text-composition stints. I like the keyboard as well as that of any portable computer I've tried. The layout of the Convertible's disk drives is attractive and useful.

How compatible is the PC Convertible? IBM makes no secret of the fact that the BIOS microcode is not identical to that of other IBM PC models. Programs that bypass the BIOS and directly access the hardware of IBM PCs might not run correctly on the Convertible as a result.

Many of the Convertible's incompatibilities are subtle. For example, the LCD can display reverse video but not high-intensity text. Another subtle incompatibility is the machine's static RAM; since the RAM needs no refresh, there are more microprocessor cycles available for any set period of time, making the processor appear to run faster. This apparent speed increase can affect programs that use timing loops for delays.

If you plan to write software for the PC family and want to have your program check to see if it is running on a Convertible, you can read the value in memory location FFFFE—a hexadecimal value of F9 means the machine is a Convertible.

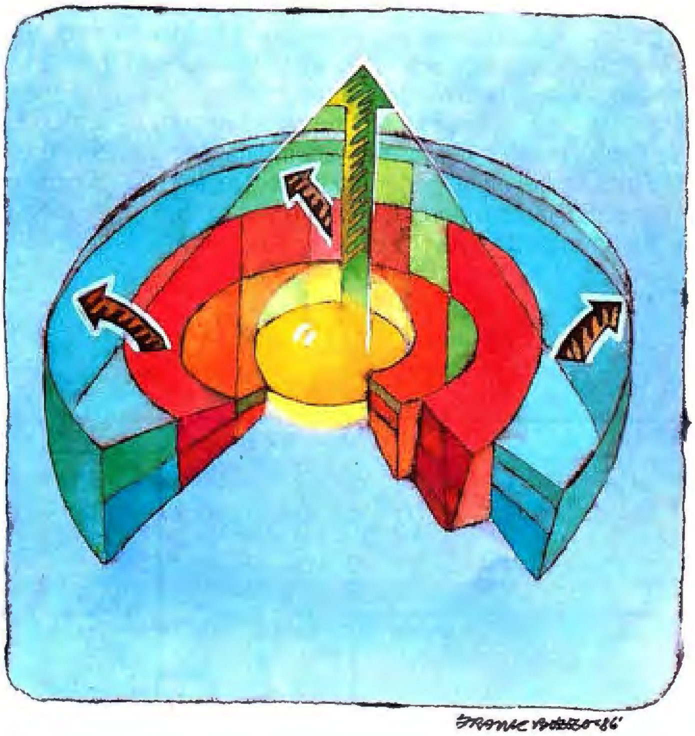
Editor's note: See the editorial on page 6 for a discussion of some other IBM PC*–family compatibility issues.* ■


BE A HERO!
CALL! PURCHASING
ORDER 20
T-33e
Power
Seek
Read
Write
INTERCOMPANY ME
TO: A.L. Walters
Director of MIS
FROM: George Schnyder
Sr. Systems Analyst
SUBJECT: Review and summary of tape back-up systems.
Attached is the report on tape back-ups that you requested. In summary, Peachtree Technology, Inc.'s Model T-33e is the tape back-up that I recomme
Model T-33e Highlights:
• Can back-up 30 megabytes.
• Meets our quality standards.
• Provides best price performance, $795 suggested retail.
• Portable—runs off the floppy port, requires no installation.
• Has the software features of the more expensive competition (file by file or mirror image back-up).
• Compact, 3 1/2" width—No footprint.
• Self Diagnostics—Full LED instrumentation.
• Runs off our Novell network.
I would like to move ahead quickly and install the initial units by month's end.
Please advise!
Great Job! Go for it! A.L.W.
Peachtree Technology, Inc.
3120 Crossing Park, Norcross, Georgia 30071, Telephone (404) 662-5158 Sales (404) 662-5556
Inquiry 231

INTEL'S 80386 ARCHITECTURE

The 80386 combines pipelined architecture, segmentation, and paging in a multitasking environment

BY PAUL WELLS



WITH A SUSTAINED performance of 3 to 4 million instructions per second, the 80386 CPU surpasses the speed of many current superminicomputers and matches the mainframes of only 10 years ago. The chip's full 32-bit architecture furnishes a high degree of parallelism and an extremely fast local bus. Hardware-supported multitasking and protection mechanisms push system performance and integrity further into the mainframe range.

The 80386 is designed to handle mainframe-size applications with typically large integers, data structures, and large numbers of programs. The 80386 can address 4 gigabytes of physical memory and supports virtual memory, a capability pioneered on mainframes.

MEMORY CAPACITY

Virtual memory allows the use of programs and data structures larger than the actual physical memory, and the 80386's 64-terabyte (2^{46} bytes) virtual memory capacity lets the processor accommodate even the largest programs and data structures.

The 80386 gives designers a wide choice of memory models from which to choose: flat physical addressing, flat demand-paged, or a fully segmented protected model. This versatility and combination of both segmentation and paging traces its origins to mainframe systems like the Control Data Cyber 180 and the IBM 360/67.

The 80386 provides object code compatibility with programs created for Intel's iAPX processor family: 8088, 8086, 80186, 80188, and 80286. Moreover, the 80386's virtual 8086 facilities let MS-DOS–based 8086 programs run concurrently with programs written for 32-bit operating systems, with each program operating as a separate task in a protected multitasking framework.

HARDWARE IMPLEMENTATION

The 80386 is manufactured using Intel's CHMOS III process, combining the high performance of HMOS with the low power dissipation of CMOS. Double metal layers and 1.5-micron lithography allow a chip density of more than 275,000 transistors. The chip is available in 12- and 16-megahertz versions and is housed in a 132-lead ceramic pin-grid array (PGA), a package designed for compactness and high reliability (see figure 1).

32-BIT ARCHITECTURE

The 80386 includes a full complement of 32-bit registers and 32-bit-wide internal data paths. You can use eight general registers interchangeably as accumulators, for arithmetic and logical operations, or as addressing registers or data registers. Each register can be used for 32- or 16-bit values, and four of them can be split into pairs of 8-bit registers. A 32-bit flags register contains processor-control flags and records the results of logical and arithmetic instructions. A 32-bit instruction pointer controls instruction fetching.

Six 16-bit segment registers implement memory segmentation. When you add the 80387 numeric coprocessor, eight 80-bit floating-point registers are available for numeric computations. The operating system uses a number of system registers, which are invisible to the application pro-

(continued)

Paul Wells is the technical marketing manager for Intel and can be contacted at Intel Corporation, 3065 Bowers Ave., Santa Clara, CA 95015, (408) 987-8080.

ILLUSTRATED BY FRANK BOZZO

The 80386 is partitioned into a six-stage pipelined architecture.

grammer, for initialization, system management, exception handling, and multitasking. Six debug registers allow the setting of code or data breakpoints for previously unavailable debug capabilities. An additional register supports the 80386 self-test capability, which lets the chip test more than half its transistors. (Figures 2a, 2b, 2c, and 2d illustrate the 80386 CPU registers in detail.)

The principal data types are 8-, 16-, and 32-bit integers and ordinals; packed and unpacked decimals; near and far pointers; and strings of bits, bytes, words, and dwords.

A complete set of instructions, including bit-manipulation instructions, implements the processing of these data types. The 80386 instruction set is a superset of 8086 and 80286 instructions. Instructions average 3.1 bytes in length and require 4.4 clock cycles for execution when measured in a typical application mix.

The 80386's symmetric addressing modes can use any combination of the eight general-purpose registers and support efficient access to standard data structures like arrays, records, arrays of records, and records containing arrays.

OPTIMIZING FOR SPEED

The 80386 is partitioned into a six-stage pipelined architecture (see figure 3). Each processor's six functional units can work independently and in parallel with the others, enabling the chip to overlap execution of multiple instructions. The bus unit carries out bus transactions for other units, while the prefetch unit reads the next 4 bytes of the instruction stream into the prefetch queue when idle bus cycles are available. The decode unit converts each instruction to its component microinstructions, and the execution unit executes these microinstructions.

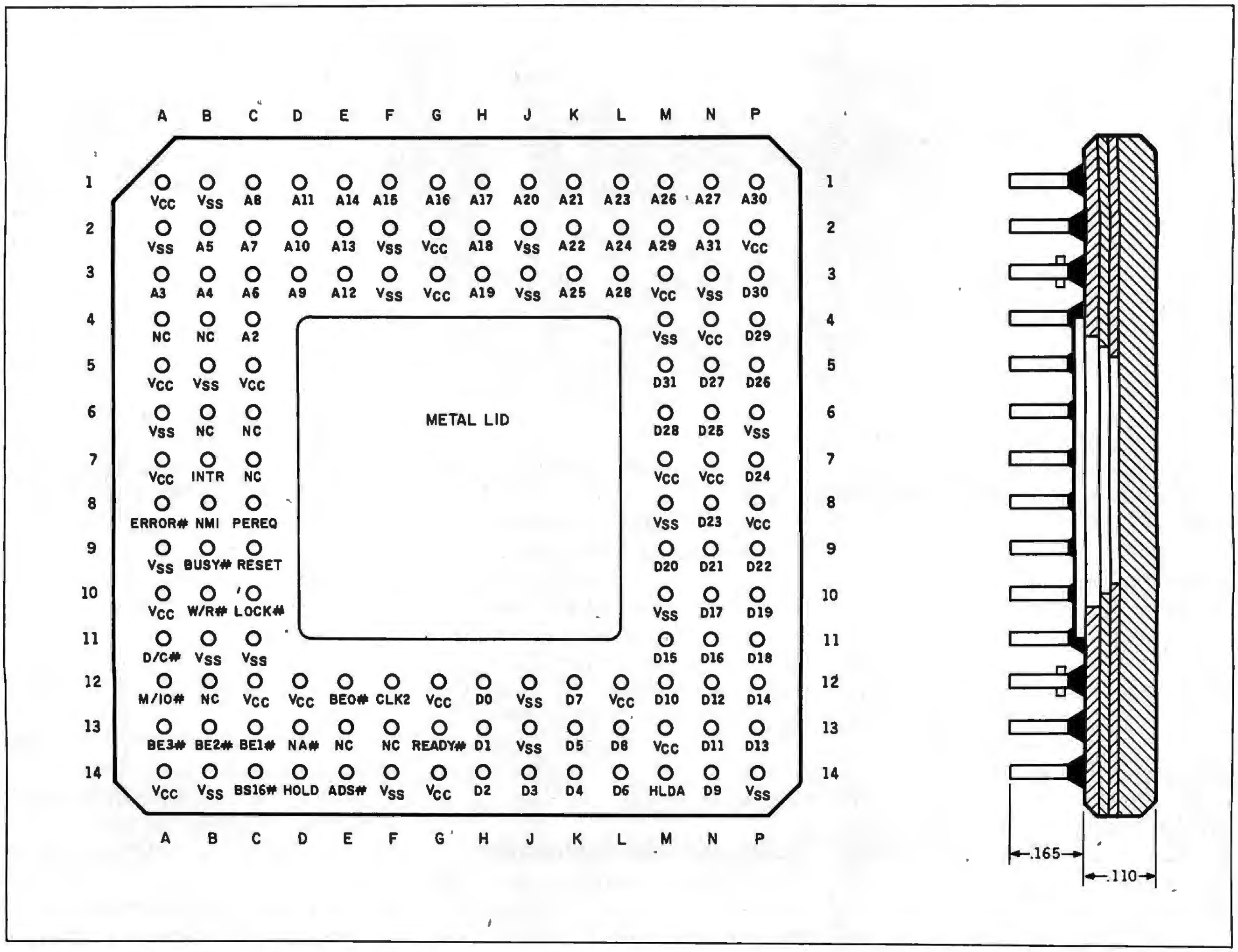


Figure 1: PGA *package and pin-out of the* 80386 *processor.*

The paging and segmentation units are also part of the pipeline. Together, these constitute the processor's memory management unit, which speeds up the address-translation process. By operating in parallel with other CPU activities, the on-chip MMU avoids the wait states associated with off-chip memory management and allows virtual-to-physical address translation with little effective delay.

The 80386 execution unit can sum two 32-bit registers in 2 clock cycles, while the multiply/divide hardware uses an advanced early-out algorithm to perform 32-bit multiplications in from 9 to 42 clocks (the exact number of clocks depends on the number of significant digits). A 64-bit barrel shifter can perform shifts of 1 to 64 bits in a single clock cycle.

The 80386 local bus achieves an overall throughput of 32 megabytes per second at 16 MHz, employing separate 32-bit data and address lines. The bus can complete an address-to-data-bus cycle in only two clocks, enabling very fast access to cache memory. Alternatively, a control feature can be used to pipeline the address from one cycle with data from the previous cycle, giving a three-clock latency appropriate to inexpensive dynamic RAMs. While the address-to-data-bus cycle increases to three clocks, address outputs and data inputs continue at a frequency of two clocks, avoiding any degradation of the processor's high bandwidth.

CACHE STRATEGY

Although a number of microprocessors feature on-chip code and data caches, their average size is about 256 bytes (too small to provide significant performance benefits). The 80386 is optimized for off-chip caching, providing a two-clock access to arbitrarily large caches. Such caches can be designed for hit rates in the 90 to 99 percent range, whereas on-chip code and data caches typically reach hit rates of only 20 to 30 percent, thus providing minimal performance benefits.

Dynamic bus sizing allows transfers of 16- or 32-bit data. This allows arbitrary combinations of 16- and 32-bit memory subsystems, connection to 16-bit buses like Intel's Multibus I, and compatibility with 16-bit peripheral devices. The processor can change the bus size during any bus cycle, and the mechanism is completely transparent to the application programmer.

PERFORMANCE

Pipelining, the on-chip MMU, a fast local bus, and the processor's compact code contribute to a fast

(continued)

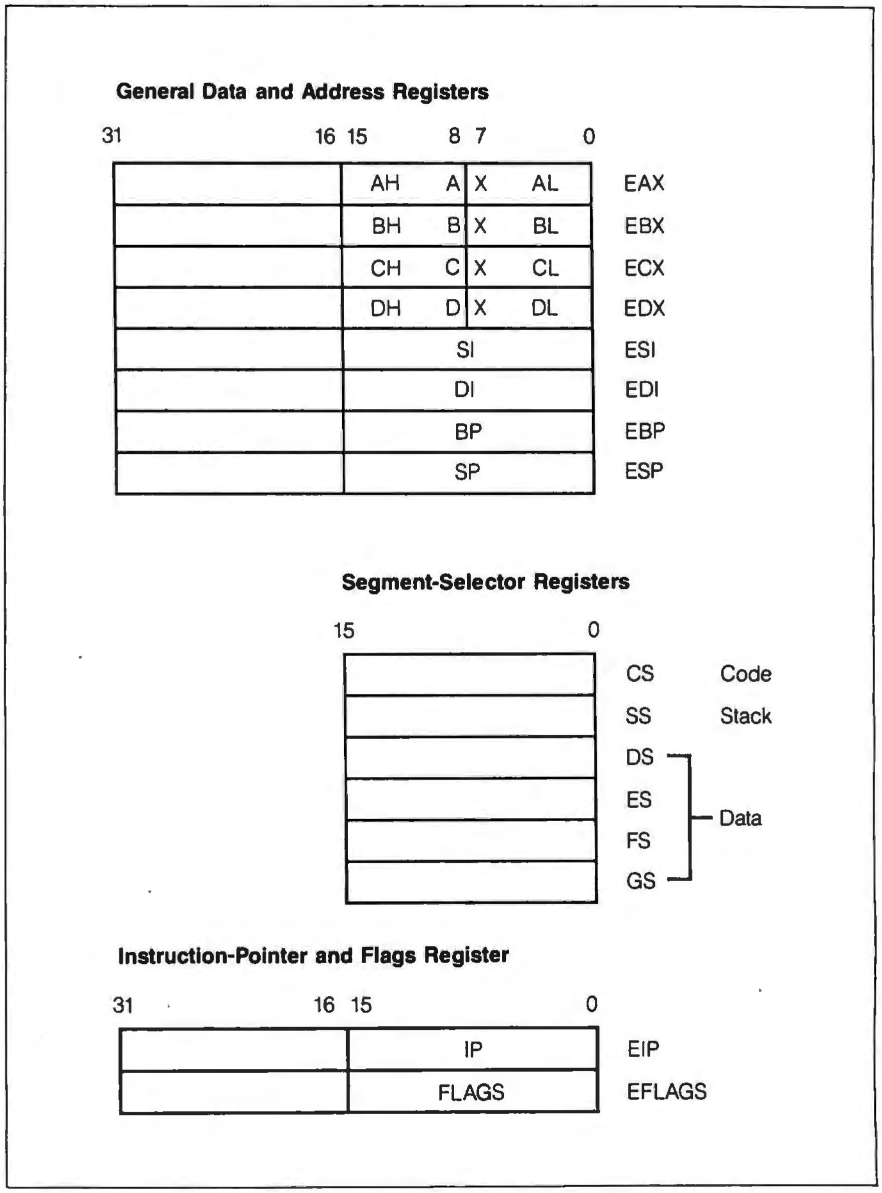


Figure 2a: *The* 80386 CPU *general data and address registers. The least significant* 16 *bits of the registers can be accessed separately using the two-letter register names* (AX, BX, *etc.*). *The* 8-*bit operations can access the individual bytes of the* 16-*bit registers using the byte names* (AL, AH, BL, BH, *etc.*). *The lower* 16 *bits of the instruction-pointer register and flags register* (EIP *and* EFLAGS) *are named* IP *and* FLAGS *and are used for* 8086 *and* 80286 *emulation.*

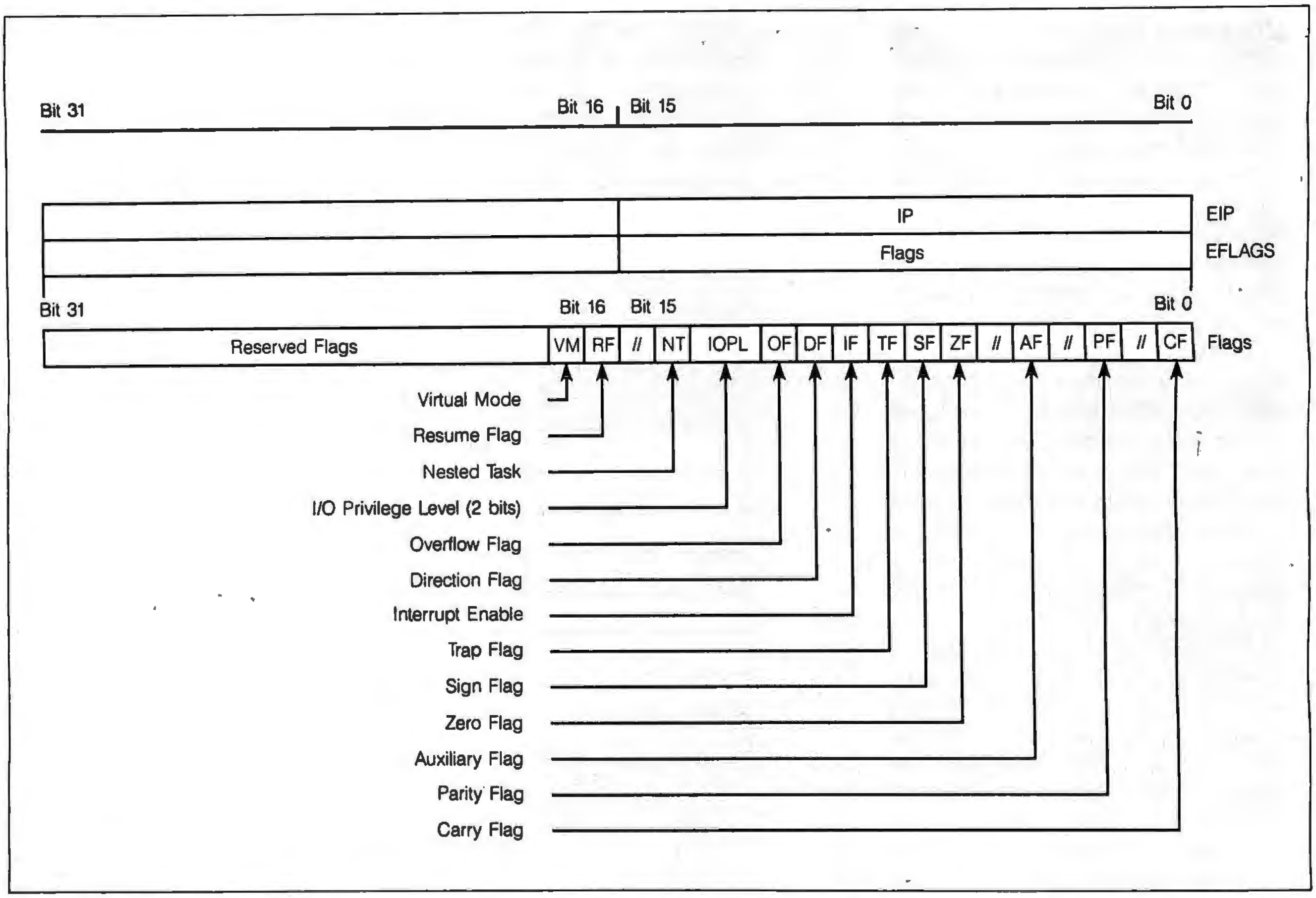


Figure 2b: *The 80386 instruction-pointer and EFLAGS registers. The 32-bit instruction-pointer register holds the offset (to be added to the base address of the current segment) for the next instruction to be executed. EFLAGS-register bit VM enables 8086 emulation when the CPU is in protected mode. The RF (resume flag) bit is used in conjunction with debug-register breakpoints. It causes the bug fault to be ignored on the next instruction. The other 16 flags are common to 8086 and 80286 modes.*

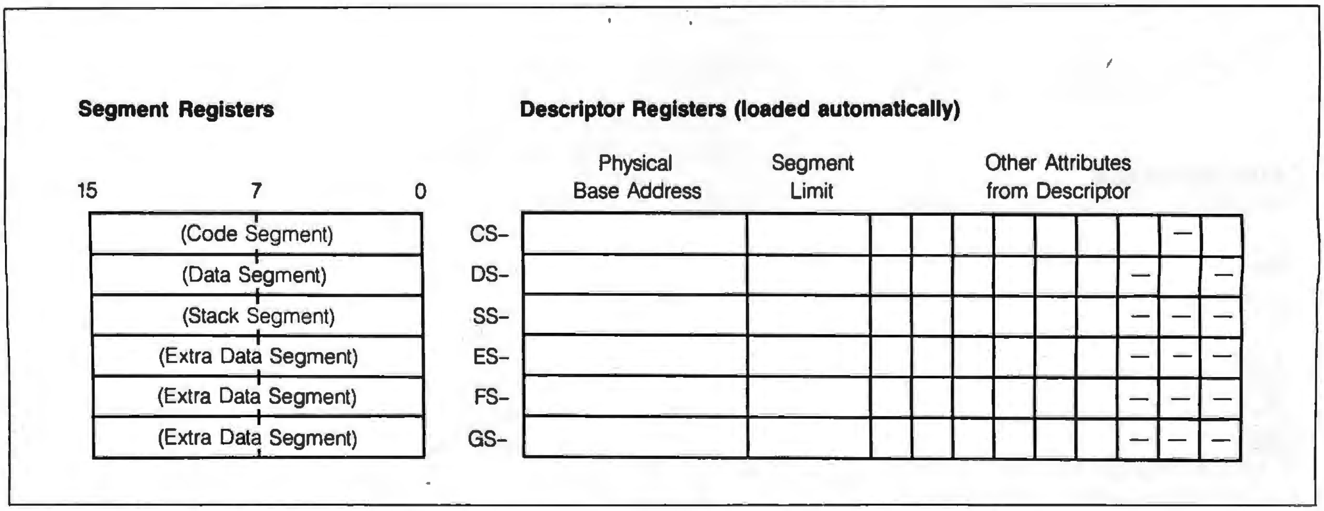


Figure 2c: *The 80386 segment registers. The 16-bit segment registers hold the selector values to the currently addressable memory locations. Real address mode segments can be 1 byte to 2^{16}, or 64K bytes in size. Protected-mode addressing allows segments of 1 byte to 2^{32} bytes, or 4 gigabytes, in size.*

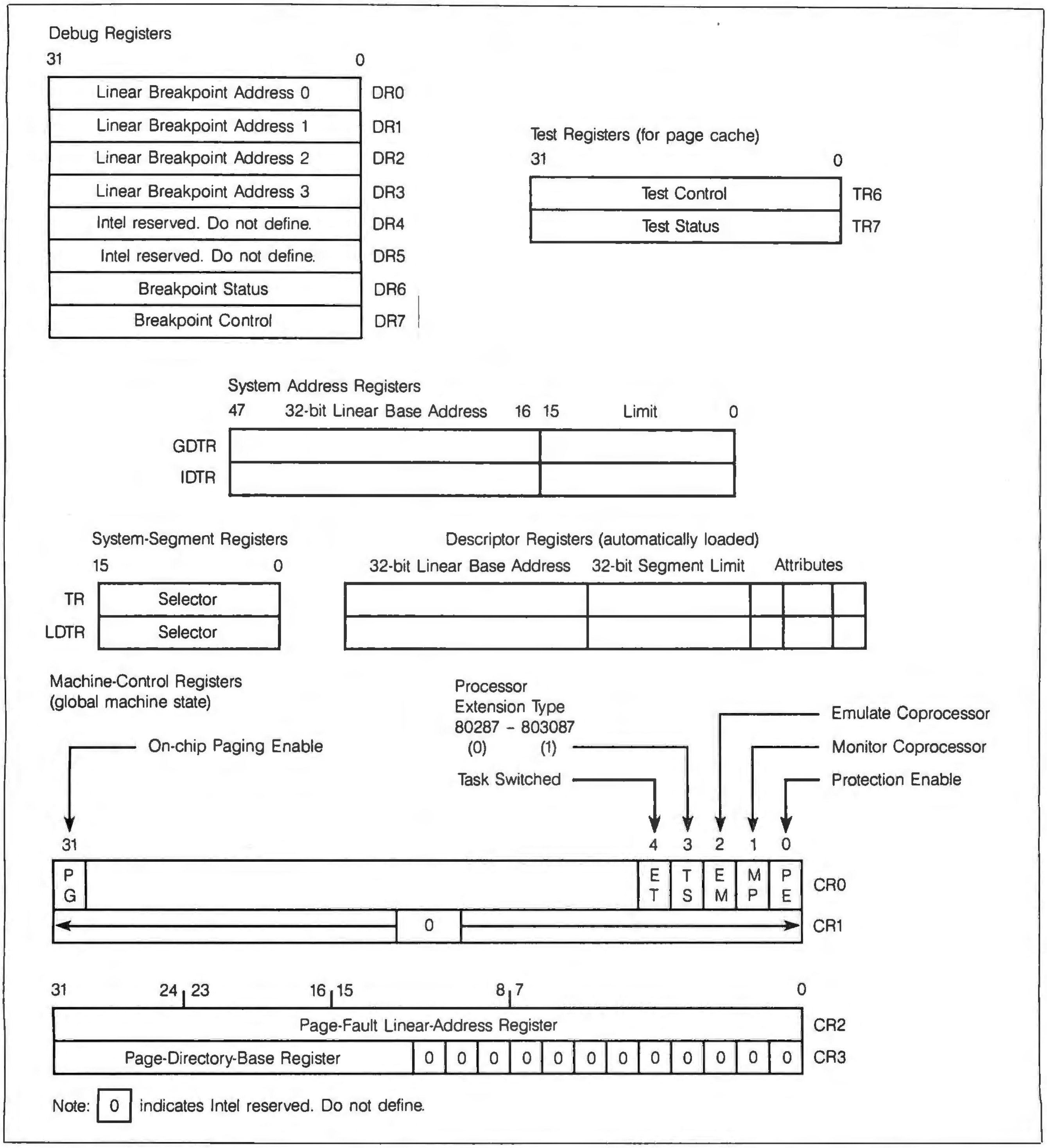


Figure 2d: *Ancillary 80386 registers. Debug-control register DR7 is used to set the breakpoints. Debug-status register DR6 holds the current state of the breakpoints. The four special system-address registers are used to reference tables or segments required by the 80286/80386 protected mode. They are named GDTR (global-descriptor-table register), IDT (interrupt descriptor table), LDT (local descriptor table), and TSS (task-state segment). Control register 0 (CR0) holds the machine-status word. Control register 2 (CR2), the page-fault linear-address register, holds the 32-bit linear address that caused the last page fault. Control register 3 (CR3), the page-directory base-address register, holds the physical base address of the page-directory table.*

Logical addresses pass through translation hardware to yield the physical address.

throughput. Recent benchmark tests set the 80386's performance at 3.64 MIPS, averaged over 14 applications and 31 million instructions.

The chip's interrupt response has been clocked at 3.7 microseconds from event to first handler instruction. Other benchmarks show the 80386 executing at 6133 Dhrystones per second. This compares favorably with the VAX-11/780's rate of 1662 Dhrystones per second and the VAX 8600 at 7000. The 80386's performance can be further boosted with an 80387 coprocessor (attaining 1.8 million Whetstones per second) or a Weitek 1167 floating-point coprocessor (4.0 million Whetstones per second).

All Intel results were obtained using a 16-MHz 80386 CPU and zero-wait-state memory. Whetstone and Dhrystone measurements used Intel's revision X.02 C compiler.

MEMORY MANAGEMENT

Virtual memory defines a logical address space that can exceed the size of the actual physical address space, letting you use mass storage devices in place of a large RAM array.

Logical addresses pass through translation hardware to yield the physical address (see figure 4). The large logical address space is subdivided into units of allocation, segments or pages, that can be swapped into and out of main memory from secondary storage devices to provide a task with necessary code or data. Swapping is performed under the management of the operating system and is transparent to the application.

Both paging and segmentation schemes have particular advantages. The 80386 provides a choice of these

(continued)

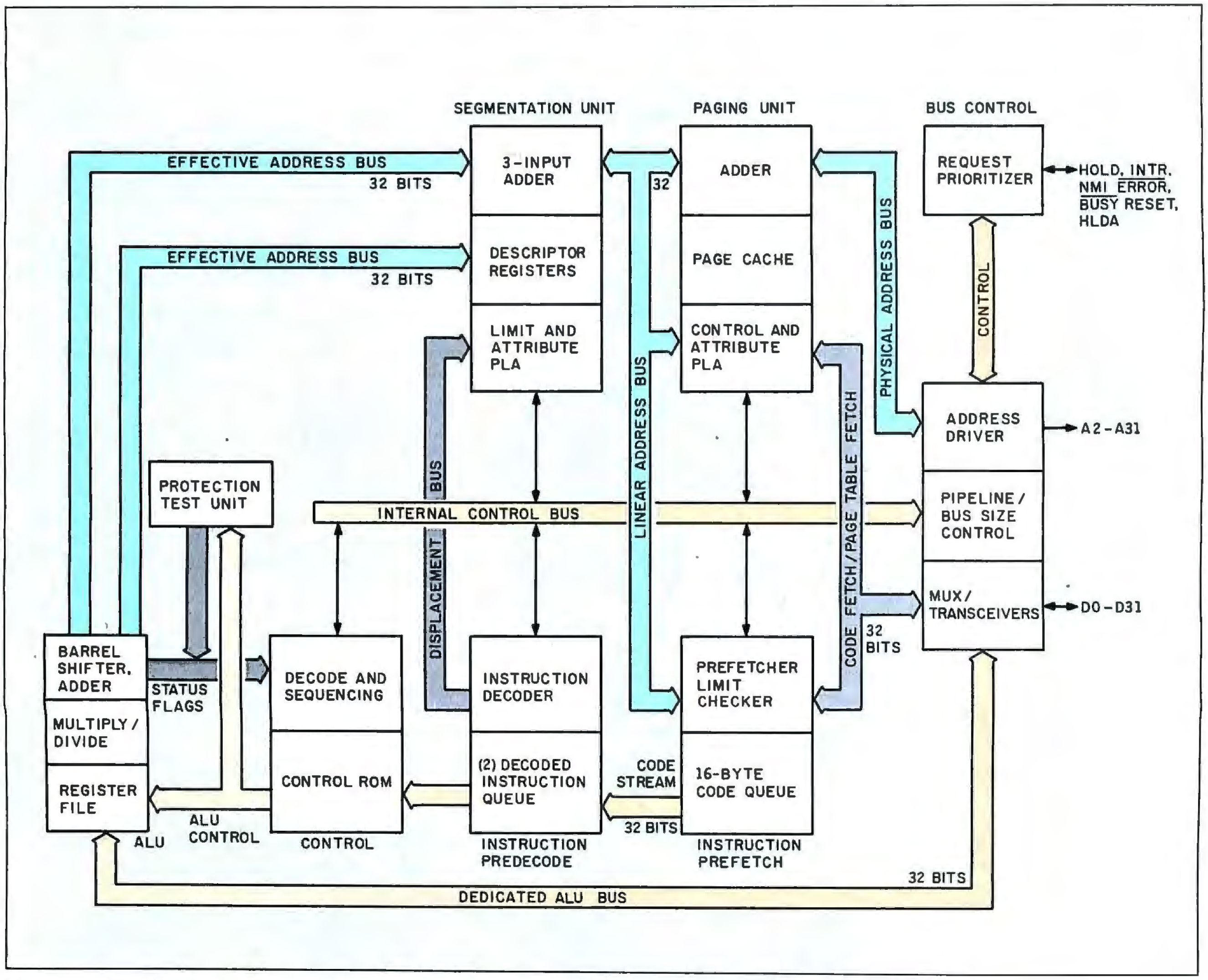


Figure 3: *The* 80386 CPU *pipeline architecture.*

For $295 you can draw your own conclusions.

Drafix™ 1. The first high performance CAD software everyone can afford.

Low-cost, high performance computer-aided design & drafting (CAD) software is finally available for use on your PC.

There's never been a professional CAD package so complete at such a low cost.

Drafix 1 from Foresight Resources. Only $295.

Powerful, full-featured second generation CAD software.

Drafix 1 offers all of the capabilities you expect to find in packages costing $2,000 or more.

This is not simply a souped-up paint package. Drafix 1 is a breakthrough in design that organizes sophisticated CAD functions into smooth, fast operations. All of the drawing, designing and editing functions that designers demand from a serious CAD tool are included in Drafix 1.

A breakthrough in screen design and visual user interface.

Drafix 1 features the best organized screen design ever devised. Very simply, it displays all the information you need, all of the time.

The entire menu hierarchy is displayed constantly. There's no need to memorize commands or search for menus.

Snap-mode options are continually shown on the left screen border and can be selected "on the fly" by either pointer device or a single keystroke. Roll-down screens provide quick access to the virtually unlimited drawing, display and editing options.

And it's all controlled by a versatile three-button mouse, or digitizer, with on-screen prompts so each button function is clearly defined.

AutoCAD® compatible for easy expansion.

If you or someone in your company already uses AutoCAD, Drafix 1 offers an inexpensive way to add to your capability.

The optional Drafix 1/AutoCAD file exchange utility permits transfer of drawings between Drafix 1 and AutoCAD.

It's the perfect, low-cost alternative for increasing your drafting and design capacity.

Unbelievable low bundle prices.

To get you up and running we offer two special hardware bundles — all the equipment you need at prices you won't believe.

Get Drafix 1 with your choice of Torrington's Manager Mouse or the LOGIMOUSE® C7 for just $395.00. Or get Drafix 1 with the SummaSketch 12" × 12" digitizer tablet with stylus for just $650.00.

Order today. Take advantage of our 30-day, money back guarantee.

We're so certain you'll like the performance and versatility of Drafix 1 you're welcome to try it risk free for 30 days. If you're not completely satisfied, return it to us for a full refund.

Find out just how good CAD software can be. For only $295.00.

Use our toll-free number and any major credit card to order your copy of Drafix 1 today.

SPECIFICATIONS

Code specifications
- Lattice "C" language
- Halo graphics
- Floating point database
- Coprocessor support
- PC-DOS/MS-DOS 2.1 or later

Drafix 1 Features

Items
- Lines
- Pt markers
- Arcs & circles
- Note text
- Polygons & ellipses
- Symbols/individual & nested

Item attributes
- 16 pen colors
- 255 Layers
- 8 Linetypes
- 12 Text fonts
- 32 Pt marker types

Screen Display
- Zoom/Pan/Full
- 8 Save Views
- Slide save/view
- Grids on/off

Metric & English Standards
- Engineering (decimal/fractional)
- Architectural (ft in)

Project drawing info.

Numeric input
- Keyboard and cursor
- Absolute, Relative, Polar

Snap Modes
- Gridpoint
- Endpoint
- Midpoint
- Intersect
- On item
- Quadrant
- Tangent
- Arc center

Transform/Copy
- Move
- Rotate
- Scale
- Mirror
- Align

Modify/Edit
- Attributes
- Break/Divide
- Fillet/chamfer
- Trim
- Stretch
- Erase
- Explode symbols & polygons
- Mask
- Replace
- Merge
- Region select
- Workgroup

Auto hatching, polygon fill

Auto Dimensions
- Linear Hor/Ver/Aligned
- Angular
- Dia/Radius
- Leader Notes
- Ordinate
- Chain & Baseline

Check calculations
- Coordinates
- Distance & angle
- Area & perimeter

Item Masking

Hardware

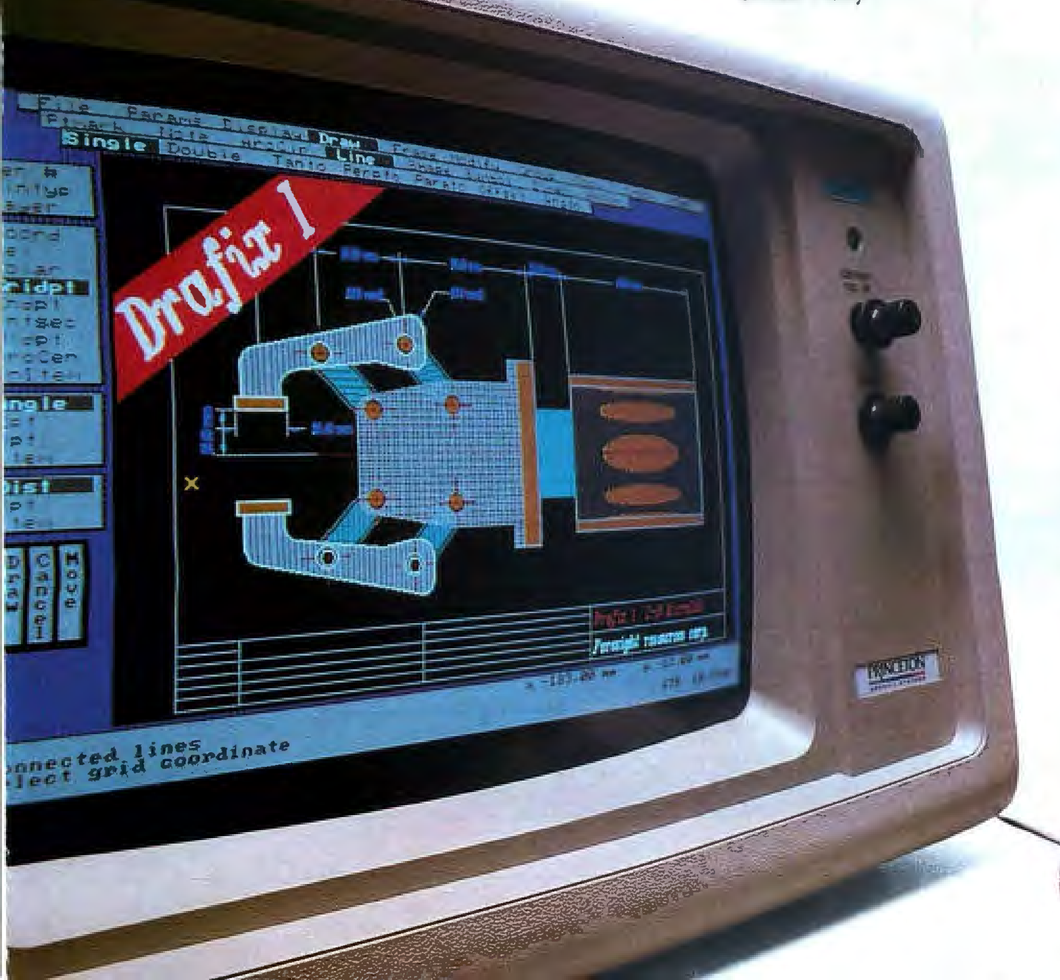
Computers
- IBM PC XT AT and compatibles
- AT&T 6300 Plus
- Zenith
- System Requirements
  - 512K RAM
  - RS 232 Comm port
  - Mouse or digitizer
  - Coprocessor recommended

Graphics Display boards
- IBM CGA/EGA and compatibles
- Hercules and a wide range of graphics boards

Digitizers & Locators
- All popular locator devices supported

Plotters
- All popular plotters ("C" size) supported
- "E" size plotter option available

Printers
- All popular laser and dot matrix printers supported



Order Now! Call Toll-Free!

1-800-231-8574, Ext. 150

- ☐ Drafix I CAD Package $295
- ☐ Drafix I with Mouse $395
  - Torrington ____
  - LOGIMOUSE ____
- ☐ Drafix I with SummaSketch tablet $650
- ☐ AutoCAD File Exchange Utility $ 95

Dealer inquiries welcome

Check, money order, Visa and MasterCard accepted

FORESIGHT RESOURCES CORP.™
932 Massachusetts
Lawrence, KS 66044
913/841-1121

AutoCAD is a trademark of Autodesk Inc.
IBM PC/XT/AT are trademarks of the IBM Corp
SummaSketch is a trademark of Summagraphics Inc
Manager Mouse is a trademark of Torrington Company

Inquiry 320 for End-Users.
Inquiry 321 for DEALERS ONLY.

two major memory models, to be used alone or in combination. System designers can choose the model best suited to their specific application.

In the simplest case, designers can elect not to employ virtual memory. They can group all addresses in a single segment, equivalent in size to the entire physical memory (up to 4 gigabytes). Addresses are organized in a linear, unstructured array. No MMU translation is required, and the logical address is precisely equivalent to the physical address. Such flat memory organizations are often preferred in embedded real-time systems like industrial controllers and private branch exchanges.

SEGMENTATION

In segmented models, logical memory is subdivided into segments of variable length (see figure 4). Each code

(continued)

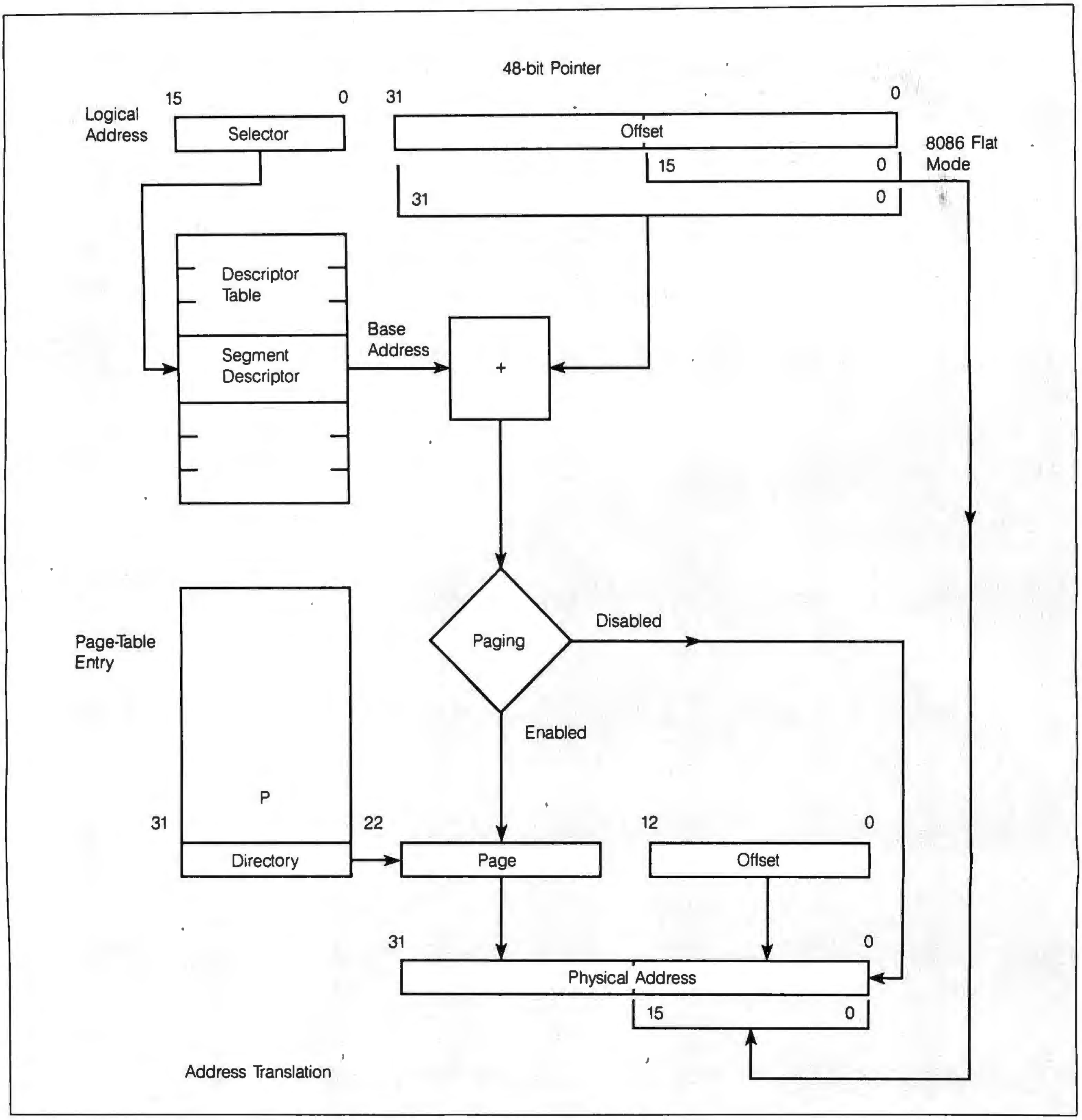


Figure 4: 80386 *address translation.*

RAM-Resident Program Users...
IT'S EASY TO WIN WHEN YOU BUY THE REFEREE™
SUPERKEY
Prokey
Lotus 1-2-3
SIDEKICK
With Referee, you make the rules.
If you use desktop organizers, spell checkers, keyboard enhancers or other RAM-resident programs, you may have already discovered the horror of "RAM Cram."
RAM Cram occurs when memory-resident programs compete with each other (and with applications programs) for control of your keyboard or other computer resources. It's a fierce competition that can cause your computer to lock up completely. Then *you* pay the penalty—in lost time and lost data.
Referee, by Persoft, is a new type of software that puts you in total control of your RAM-resident programs.
And that puts Referee in a league all its own. You can create your own RAM Teams™ for specific applications programs. Team Superkey™ and Sidekick™ with Lotus 1-2-3™. Or call in Prokey™ and bench the others when you switch to dBASE III™.
Load all the programs you need at the beginning of your workday. Referee automatically activates and deactivates the ones you need according to *your* set of rules. Instantly. Invisibly. You can also use Referee to unload programs from memory—even those with no unload option of their own.
Referee's Sideline™ menu enables you to control RAM-resident programs from within an applications program!
It's ideal for integrated packages like Symphony™. You can use a keyboard enhancer with the spreadsheet module. Or deactivate it, enter the word processing module and activate your favorite spell checker. It's easy. And you never have to back all the way out of the program!
Referee puts an end to RAM-resident program conflicts. At $69.95, it solves a very big problem for a very small price.
It's easier to win with the Referee on your side. For more information, and for the dealer nearest you, contact: Persoft, Inc., 465 Science Drive, Madison, WI 53711, (608) 273-6000—Telex 759491.
Referee™
persoft
© Copyright 1986 Persoft, Inc. All rights reserved.
Referee, Sideline, and RAM Teams are trademarks of Persoft, Inc.
Sidekick and Superkey are registered trademarks of Borland International, Inc.
Lotus 1-2-3 and Symphony are registered trademarks of Lotus Development Corp.
Prokey is a trademark of RoseSoft, Inc.
dBASE III is a registered trademark of Ashton-Tate.
Visit Booth #8624
Hilton Pavillion West
COMDEX/Las Vegas

SYSTEMS IMPLICATIONS OF THE INTEL 80386

BY BILL NICHOLLS

What is a "systems" point of view? Briefly stated, it is the evaluation of a component's performance while it is part of a functioning system. This is a real-world view of performance, since no component operates without some interface to other equipment. Evaluation of a new processor must therefore be done in light of current and future demands on the whole computer system.

CLASSIFYING THE 80386

Classification of a processor is not one-dimensional. The primary dimensions are speed, architecture, and technology. In these broad terms, the 80386 is fast in operation (12- to 16-megahertz processor timing), complex in architecture (large instruction set, levels of protection, and virtual memory support), and current in technology (CHMOS).

How does the 80386 compare to other processors, small and large? Clearly, the 80386 is a microprocessor because of its implementation on a single chip. But do its characteristics qualify it as a mainframe? Certainly its speed is more than adequate for that classification. But while the architecture is as complex as some mainframes, it is incomplete in the area of I/O paths.

The 80386 can perform a number of direct I/O functions of various types and sizes. However, each element transferred requires the execution of one or more instructions to perform the transfer. Low-speed I/O can be done this way using interrupts, but high-speed I/O requires interrupts to be locked out and the chip dedicated to one peripheral device. This is acceptable only in some dedicated systems, not in general-purpose systems.

The inability to perform I/O while continuing to compute separates the 80386 from a full mainframe classification. Clearly, other chips can perform that function, and the combination of those chips and an 80386 is functionally equivalent to a mainframe.

Given a proper I/O design and adequate memory size and speed, the 80386 would compare closely in performance to an IBM Model 4341-1 or 4341-2, depending on a number of factors. The higher instruction-execution rate of the 80386 would be offset by the more powerful instructions of the 4341. Memory and I/O bandwidth would also have to be evaluated for specific designs. In capability, however, the 80386 could perform at the level of an IBM 4341. The major differences in performance would depend on how the 80386's I/O subsystem compared to the 4341's I/O subsystem and the use of special 4341 instructions.

HIGH-PERFORMANCE HARDWARE

In this context, high performance means maximizing system performance relative to the design objective without excessive cost. For systems with the same processor, different design objectives will lead to different performance levels due to the economic choices made.

The primary requirement for a high-performance 80386 system is adequate memory bandwidth. Bandwidth as measured in megabytes per second is an important number. However, bandwidth that is not carefully matched to the processor cycle is usually wasted.

Some additional bandwidth is required for I/O. For example, an 80386 with a 12.5-MHz clock tick (25-MHz crystal) requires an 80-nanosecond memory access for zero wait states and 160 ns for one wait state. With a zero-wait-state memory, the processor is internally (CPU) limited; with a one-wait-state memory, it is bus-limited. If a processor is bus-limited, I/O cycles on the bus will interfere with processor execution, causing some reduction in processing capacity. For designs like single-user computers, this is probably acceptable. For large multiuser systems, it means trading off memory cost against performance gains that are important but hard to justify.

Until recently, there was no easy way to match a fast processor to slower memory speeds without loss of performance. The typical solution was to use a small but fast cache memory and a larger but slower main memory. The added complexity and cost removed this technique from the personal computer market. The 80386 offers a method for adding this performance boost.

Using an external pin called *next address*, the designer can have the processor output the next address before the current memory cycle is finished. This enables the address to be decoded for the next memory access to happen concurrently with the current memory data cycle. This is called pipelined mode access.

Table A shows the trade-off between processor clock speed, wait states, and required memory-access time. The Bus Cycles Available column shows the percentage of the bus that can be used

Table A: *The differences in processor clock speed, wait states, and memory-access times for the Intel* 80386.

Processor Clock	Access Mode	Memory 0 ws	Access 1 ws	Bus Cycles Available 0 ws	Bus Cycles Available 1 ws
12.5 MHz	2-cycle	80 ns	160 ns	27%	14%
	2-pipe	160 ns	240 ns	21%	11%
16 MHz	2-cycle	45 ns	107 ns	27%	14%
	2-pipe	108 ns	170 ns	21%	11%

Note: 2-cycle is the standard memory access; 2-pipe is the 2-cycle pipelined mode of access.

by I/O before interference with the processor is certain. Below that level, I/O interference is small.

Another way to enhance large-system performance is with banked memory, a technique used in large mainframes. The basic system can be delivered with a single bank of memory. As the system grows in the field, a second bank of memory could be added so that sequential addresses would then alternate banks. This bank alternation works for both I/O and processor access. Large mainframes may use up to 16-way bank interleave, but multiprocessor microcomputers will probably need no more than 4-way banked memory for higher performance.

FAST I/O

Fast I/O is not well supported in the 80386. An external direct memory access is required. With the complexity of addressing in the 80386, even simple DMA can be awkward. A single I/O command can refer to data that spans over page boundaries and is relocated in noncontiguous real memory addresses. Thus, the operating system must first test for this condition and split one logical I/O operation into more than one physical I/O operation, then schedule and coordinate the physical I/O operations, and finally notify the calling program when done. In addition to adding significant overhead, it also complicates error recovery.

A second type of DMA, "smart" DMA, is now available. It can perform multiple operations as a single logical I/O operation and relieve the processor of substantial overhead. Large-scale systems have had this feature, *scatter read/gather write*, for some time. This means the DMA chip can read a physical record from scattered locations and gather scattered data to write as a single record. These features are incorporated in the Intel 82258 DMA chip.

In addition to a powerful DMA capability, the hardware must support higher transfer rates than the current 5 megabits per second. Multiple transfers on different channels at the same time must also be supported. Both of these requirements can be met with the 82258, which can support four high-speed channels with a maximum throughput of 16 megabytes per second. The 82258 can also operate as three high-speed channels and one multiplexed channel supporting 32 subchannels at up to 30k bits per second each.

Few programs will be able to fully utilize an 80386, even with fast I/O. One way to use the available time is to run multiple programs or tasks. Multitasking divides the processor's time in slices and allocates one slice to each task, switching rapidly between tasks to give the illusion of continuous operation. Some tasks may slow down compared to running alone, but overall throughput increases up to a point.

That point is determined primarily by the operating system overhead in performing its primary services: I/O control and task switching. More tasks, more I/O, or shorter time slices increase the overhead. When the increase in overhead exceeds the improvement in throughput, system performance begins to degrade. If the processor provides fast methods for I/O processing and task switching, a system with a well-designed operating system will reach a higher level of throughput before system performance degrades.

The Intel 80386 processor has fast task switching in its instruction set. A single instruction will totally switch tasks including registers and environments in 17 to 20 microseconds. This is an order of magnitude faster than processors without this instruction and significantly reduces overhead in support of multitasking. Interrupts can use task switches or, for fastest response, calls that execute in 3 microseconds but do not switch task environments.

Bill Nicholls is president of, and a software engineer at, BGW Systems Inc. (16714 Meridian S, Suite 200, Puyallup, WA 98373).

and data module (such as file buffers, program variables, and so on) can be assigned a different segment. Since each module has its own logical memory address space, it is easy to implement schemes that protect individual modules, share them selectively between tasks, and manipulate them separately or together. Segments in the 80386 can be any size up to 4 gigabytes (enough for the largest code and data modules), with a maximum of 16,000 segments per task.

This is how the 80386 implements segmentation. The logical address used by the application consists of a 16-bit segment selector and a 32-bit offset into the selected segment. The selector references a segment descriptor held in a table that the operating system maintains. The 64-bit descriptor contains the segment's 32-bit base address and limit (length), together with swapping information and protection parameters like privilege level, access rights, and segment type. In addition, the descriptor's present bit tells the processor whether the desired segment is present in memory or needs to be swapped in from disk.

The segment's base address, obtained from the descriptor, is added to the offset to give a linear address. The linear address is then passed through the paging unit to yield the physical address unless paging is disabled, in which case the linear address is the physical address.

The processor automatically maintains descriptors for the current segment in an on-chip cache, minimizing the need to index memory-based descriptor tables. The cache-resident segment descriptor is replaced automatically with a new one whenever a new segment selector is loaded into a segment register. Programs can address up to six segments at a time: a code segment, a stack segment, and up to four data segments.

PAGING

While segments vary in size and map well to programming constructs, pages are small blocks of fixed size

(continued)

The page-table entry includes attributes like present, rights, accessed, and dirty.

(4K bytes in the 80386). Paging simplifies swapping algorithms by providing uniform units of allocation and relocation. Thus, while segmentation can be used to structure the logical address space, paging can be used to manage the physical memory.

The paging system uses a 32-bit linear address. The upper 10 bits of the address index a 1024-entry page-table directory to obtain a page-table address (see figure 5). The middle 10 bits of the linear address are added to the page-table address to get a page-table entry that describes the target page. The final 12 bits of the linear address are added to the page address to generate a 32-bit physical address.

Just as the processor's segmentation mechanism caches segment descriptors, the paging unit supplies an on-chip associative cache containing mapping information for the 32 most recently used pages. The cache, called a translation lookaside buffer (TLB), has a hit rate of 98 to 99 percent, minimizing the performance impact of enabling the paging system.

The page-table entry includes attributes like present, rights, privilege level, accessed, and dirty. The accessed bit is set automatically whenever a page is read or written. The dirty bit is set whenever a page is written. By monitoring accessed and dirty bits, the operating system can implement a variety of page-replacement algorithms, like least recently used (LRU) approximation algorithms. No other microprocessor provides both dirty and accessed bits.

COMBINING PAGING AND SEGMENTATION

Although it is possible to use paging or segmentation alone, the 80386 also lets designers combine the advantages of both models, providing a powerful capability.

The application programmer employs segmentation to implement data sharing, software protection, and multitasking, while paging provides the underlying physical memory management. The segmentation/paging combination was first used by Control Data and IBM mainframes and is also employed by UNIX System V, which allots each task a segment for code and a segment for data and the stack.

When the two memory management organizations are used together in the 80386, linear addresses pro-

(continued)

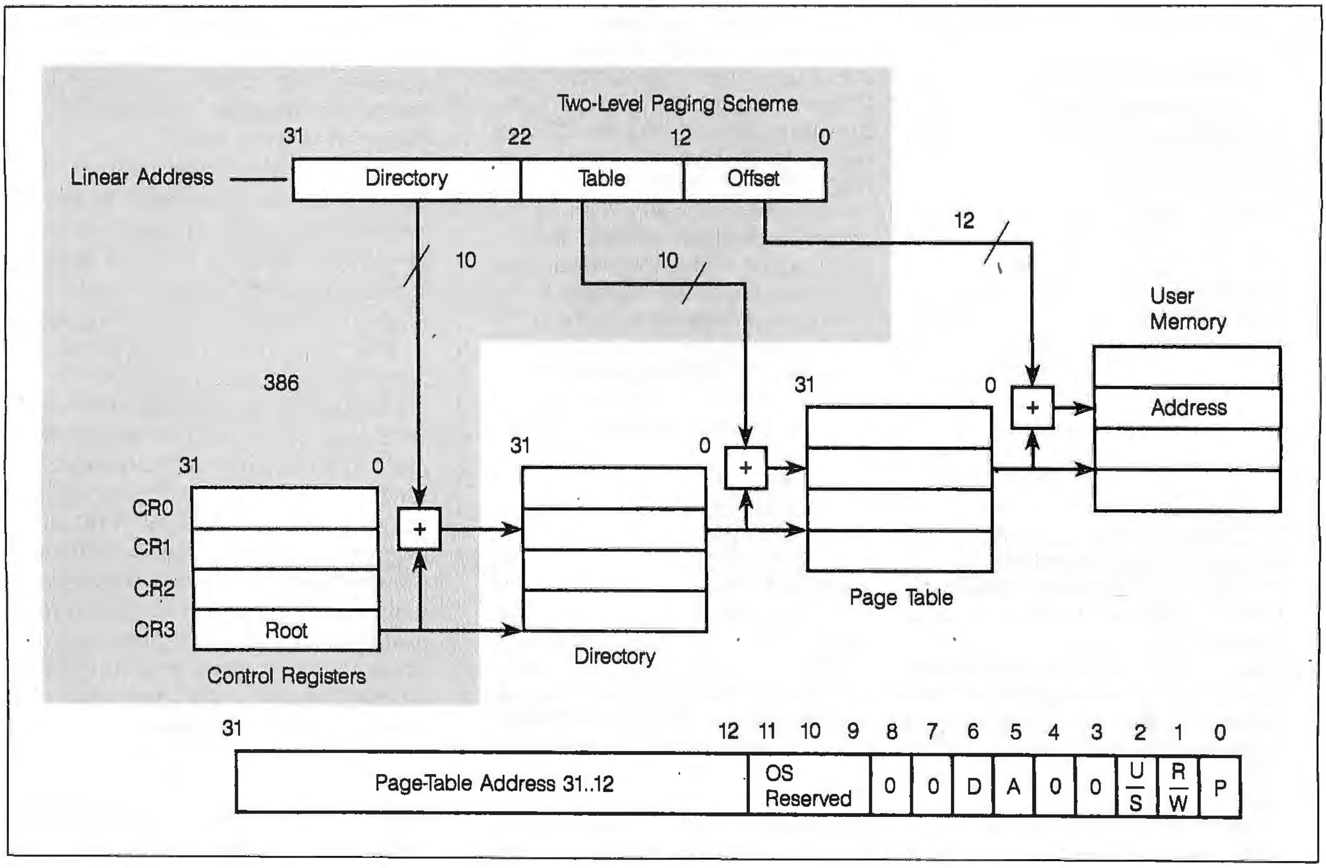
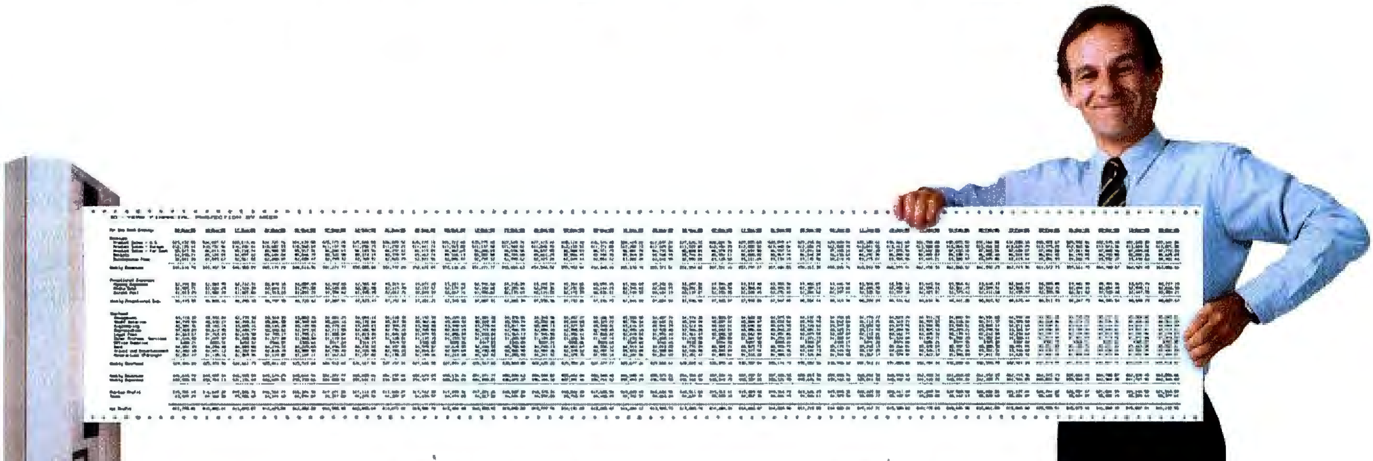


Figure 5: 80386 *paging system.*

We've gone to great widths to make you look good.



Introducing New SIDEWAYS™ Version 3.

Sideways has always been the favorite way to print nearly everything that's too wide for a printer. Not surprising.

It's quick. It's easy. And it makes clean, one-piece printouts as wide as you like. So you get spreadsheets that really spread out. Pert charts that peer far into the future. All kinds of extra-wide reports without staples, glue or tape.

But now Sideways Version 3 goes even further.

Especially for 1-2-3™ and Symphony.®

Now you can make short work of those wide 1-2-3 spreadsheets. No need to make print files. Just highlight a range and print right from your worksheet file using familiar Lotus-like menus.

Mix bold, underlined and expanded type, all on the same page. Print through or skip over perforations. Make the finest extra-wide printouts this side of typesetting.

Using Symphony? Sideways chimes right in to print any worksheet you've loaded without ever leaving your Symphony session.

No matter what software you use.

Spreadsheets, databases, word processors, project planners. Whatever you use, you'll get more out of new Sideways.

Choose from nine different type sizes, from "minuscule" to "mammoth." Print bold, underlined or expanded – whatever your software can create. Even foreign language and line-drawing characters. Plus, there are so many ways to adjust margins, spacing and page size, it stretches the imagination.

Sideways Version 3 for the IBM® PC and all popular dot-matrix printers. Available now at computer dealers nationwide. Or directly from Funk Software with your check, Visa or Mastercard.

At just $69.95 a copy, you'll never want for width again.

Funk Software, 222 Third Street, Cambridge, MA 02142. Phone (617) 497-6339.

duced by the segmentation unit are passed through the paging mechanism to generate the final physical address. Overall, this multistage address-translation facility is fast, due to the on-chip translation caches and the pipelining of memory management functions. Translation from logical to physical address typically requires only one and a half clocks. From the user's perspective, the effective delay is zero clocks since translation is performed on-chip, overlapped with other processor activities.

PROTECTION HARDWARE AND SOFTWARE

An additional benefit of the 80386's on-chip MMU is an array of protection mechanisms that operating systems can selectively employ, including separation of task address spaces, typed segments, access rights for segments and pages, and segment-limit checking.

A frequent and insidious problem occurs when an erroneous task violates the address space of another task and overwrites code or data. In this way, bugs can propagate through a system and be nearly impossible to track. The 80386 protects task address spaces by assigning them to different segments and/or separate page directories and page tables. Segment-descriptor tables prevent unwarranted access by other tasks.

In addition to isolating tasks from one another, the processor provides a ring-protection mechanism that can guard low-level software both from bugs in overlying layers and from intentional tampering at the application level. Each segment descriptor and page-table entry has a field that defines the segment's and page's privilege level, ranked from 0, the most privileged level, to 3, the least trusted. Programs at a given level can access data and code only at the same or a numerically higher level (see figure 6).

Segment descriptors hold not only privilege information but also the segment limit (references to the segment must fall within the limit, preventing

(continued)

31	24	16	8	5	0	
Selector		Offset 15...0				0

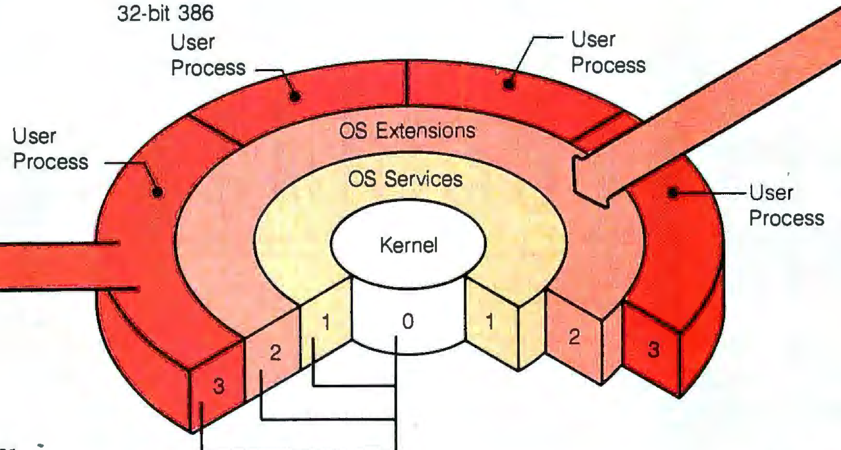
Offset 31...16	P	DPL	0	Type	0	0	0	Word Count 4...0	+4

Gate Descriptor Field

Name	Value	Description
Type	4	286 call gate
	5	Task gate (for 286 or 386 task)
	6	286 interrupt gate
	7	286 trap gate
	C	386 call gate
	E	386 interrupt gate
	F	386 trap gate
P	0	Descriptor contents are not valid
	1	Descriptor contents are valid

DPL—least privileged level at which a task may access the gate. WORD COUNT 0-31—the number of parameters to copy from caller's stack to the called procedure's stack. The parameters are 32-bit quantities for 386 gates and 16-bit quantities for 286 gates.

Destination Selector	16-bit selector	Selector to the target code segment or Selector to the target task-state segment for task gate
Destination Offset	offset 16-bit 286 32-bit 386	Entry point within the target code segment



Segment Descriptor

31											0	Byte Address
Segment Base 15...0						Segment Limit 15...0						0
Base 31...24	G	D	0	0	Limit 19...16	P	DPL	S	Type	A	Base 23...16	+4

Base	Base address of the segment
Limit	The length of the segment
P	Present Bit 1=Present 0=Not Present
DPL	Descriptor Privilege Level 0-3
S	Segment Descriptor 0=System Descriptor 1=Code or Data Segment Descriptor
Type	Type of Segment
A	Accessed Bit
G	Granularity Bit 1=Segment length is page granular 0=Segment length is byte granular
D	Default Operation Size (recognized in code segment descriptors only) 1=32-bit segment 0=16-bit segment
0	Bit must be zero (0) for compatibility with future processors

Figure 6: *Four levels of protection.*

Segment Descriptor-Cache Register Contents

	32-bit Base (Updated during Selector Load into Segment Register)	32-bit Limit (Fixed)	Other Attributes (Fixed)									
	Base	Limit	Present	Privilege Level	Accessed	Granularity	Expansion Direction	Readable	Writable	Executable	Stack Size	Conforming Privilege

Real Address Mode (Segment Limit and Attributes are Fixed)

	Base	Limit	Present	Privilege Level	Accessed	Granularity	Expansion Direction	Readable	Writable	Executable	Stack Size	Conforming Privilege
CS	16X Current CS Selector*	0000FFFFH	Y	0	Y	B	U	Y	Y	Y	–	N
SS	16X Current SS Selector	0000FFFFH	Y	0	Y	B	U	Y	Y	N	W	–
DS	16X Current DS Selector	0000FFFFH	Y	0	Y	B	U	Y	Y	N	–	–
ES	16X Current ES Selector	0000FFFFH	Y	0	Y	B	U	Y	Y	N	–	–
FS	16X Current FS Selector	0000FFFFH	Y	0	Y	B	U	Y	Y	N	–	–
GS	16X Current GS Selector	0000FFFFH	Y	0	Y	B	U	Y	Y	N	–	–

Key:
- Y = yes
- N = no
- 0 = privilege level 0
- 1 = privilege level 1
- 2 = privilege level 2
- 3 = privilege level 3
- U = expand up
- D = expand down
- B = byte granularity
- P = page granularity
- W = push/pop 16-bit words
- F = push/pop 32-bit dwords
- – = does not apply to that segment-cache register

*Except the 32-bit CS base is initialized to FFFFF000H after reset intersegment CALL or intersegment JMP, or INT).

Protected Mode (Loaded per Descriptor)

	Base	Limit	Present	Privilege Level	Accessed	Granularity	Expansion Direction	Readable	Writable	Executable	Stack Size	Conforming Privilege
CS	Base per Seg Descr	Limit per Seg Descr	p	d	d	d	d	d	N	Y	–	d
SS	Base per Seg Descr	Limit per Seg Descr	p	d	d	d	d	r	w	N	d	–
DS	Base per Seg Descr	Limit per Seg Descr	p	d	d	d	d	d	d	N	–	–
ES	Base per Seg Descr	Limit per Seg Descr	p	d	d	d	d	d	d	N	–	–
FS	Base per Seg Descr	Limit per Seg Descr	p	d	d	d	d	d	d	N	–	–
GS	Base per Seg Descr	Limit per Seg Descr	p	d	d	d	d	d	d	N	–	–

Key:
- Y = fixed yes
- N = fixed no
- d = per segment descriptor
- p = per segment descriptor; descriptor must indicate "present" to avoid exception 11 (exception 12 in case of SS)
- r = per segment descriptor, but descriptor must indicate "readable" to avoid exception 13 (special case for SS)
- w = per segment descriptor, but descriptor must indicate "writable" to avoid exception 13 (special case for SS)
- – = does not apply to that segment-cache register

Virtual 8086 Mode within Protected Mode (Segment Limit and Attributes are Fixed)

	Base	Limit	Present	Privilege Level	Accessed	Granularity	Expansion Direction	Readable	Writable	Executable	Stack Size	Conforming Privilege
CS	16X Current CS Selector	0000FFFFH	Y	3	Y	B	U	Y	Y	Y	–	N
SS	16X Current SS Selector	0000FFFFH	Y	3	Y	B	U	Y	Y	N	W	–
DS	16X Current DS Selector	0000FFFFH	Y	3	Y	B	U	Y	Y	N	–	–
ES	16X Current ES Selector	0000FFFFH	Y	3	Y	B	U	Y	Y	N	–	–
FS	16X Current FS Selector	0000FFFFH	Y	3	Y	B	U	Y	Y	N	–	–
GS	16X Current GS Selector	0000FFFFH	Y	3	Y	B	U	Y	Y	N	–	–

Key:
- Y = yes
- N = no
- 0 = privilege level 0
- 1 = privilege level 1
- 2 = privilege level 2
- 3 = privilege level 3
- U = expand up
- D = expand down
- B = byte granularity
- P = page granularity
- W = push/pop 16-bit words
- F = push/pop 32-bit dwords
- – = does not apply to that segment-cache register

Figure 7: *Segment-descriptor registers.*

such common errors as stack overflow and array subscript errors); segment type, such as code or data (thus catching an attempt to overwrite code, for example); and access rights, such as read-only.

Thus, a spectrum of protection schemes is possible, ranging from no protection to a four-level hierarchy that separates the operating system kernel, residing at level 0, from the operating system services at level 1, custom operating system extensions at level 2, and applications at level 3. (Figure 7 shows segment-descriptor-register contents for real address, protected, and virtual 8086 modes of operation.)

Of course, it is necessary to let applications at low privilege levels make occasional calls, like disk I/O requests, to more protected levels. To handle cases such as this, the operating system can define entry points called gates. Gates contain the logical addresses of entry points and a set of attributes.

MULTITASKING

The 80386 was designed for high-throughput multitasking, in which each task sees only its own execution context and is unaware of other tasks running concurrently. The operating system switches the processor from one context to another, interleaving tasks in accordance with a scheduling policy.

Several data structures, transparent to applications, are defined to support the operating system, principally the task-state segment (TSS). The TSS holds the state of tasks, including processor-register values and data such as scheduling priority, address-space definitions, descriptors, and page-table-directory base. Values held in the 80386 task register point to the running task's TSS.

Because task switching can be frequent, the 80386 provides special high-speed hardware to support this operation. In response to a jump instruction with a special address, the processor automatically stores the state of one task and loads the state of another (at a speed roughly 10 times as fast as software-controlled multitasking).

SOFTWARE COMPATIBILITY

The 80386 supplies binary code compatibility with all previous 8086-family software applications. You can port operating systems and programming tools directly to the new processor without recompilation or assembly.

For 8086 programs, the processor provides virtual 8086 mode, which establishes a task-selectable protected 8086 environment within the multitasking framework. In this environment, 8086 tasks can take advantage of the 80386's protection and paging

(continued)

The I/O privilege level is used to control interrupt enabling, disabling, and system calls.

mechanisms, while running nonprivileged 8086 instructions directly at roughly 16 times the speed of an 8086. If the 80386 operating system you are using supports virtual memory, you can swap 8086 tasks just like any other tasks.

VIRTUAL 8086 MODE

In virtual 8086 mode (activated by the VM86 flag in the flags register), the 80386 presents 8086 operating systems and applications with a virtual machine (the 80386 looks and acts just like an 8086 to the application program). Segment-register semantics are the same as in the 8086.

The total address space of the 8086 is 1 megabyte. Addresses generated by virtual 8086 tasks will all fall within the first megabyte of the 80386 linear address space. In order to assign each task a separate address space, paging is used to relocate the linear address spaces to different areas of the physical address space.

Many 8086 programs were designed. for single-tasking environments, and many use I/O and interrupt control in a manner inappropriate for multitasking systems. The 80386 therefore provides mechanisms to selectively trap and manage this I/O and interrupt activity.

I/O PRIVILEGE LEVEL

The I/O privilege level (IOPL) is used to control interrupt enabling, disabling, and system calls. The I/O-port permission map selectively controls access to I/O ports. Memory-mapped devices can be managed using page-level protection attributes. The software module that manages these operations is called a virtual machine monitor.

With the exception of interrupts and system calls, all 8086 application instructions execute directly on the 80386 at full CPU speed. And since the 80386 is a proper superset of the 80286 architecture, the 80386 runs 80286 applications unchanged. Because the virtual 8086 mode is selectable on a per-task basis and the execution of 16-bit 80286 code is selectable on a per-segment basis, the 80386 can run 8086 applications, 80286 applications, and 32-bit 80386 applications concurrently in a protected multitasking environment.

MULTIPLE EXECUTION ENVIRONMENTS

Taken together, the 80386's flexible memory management architecture, multitasking, protection, and virtual 8086 mode open a vast range of design options. Most striking is the ability to host multiple concurrent-execution environments. Numerous applications, written for different operating systems and different memory management schemes, can run under a 32-bit operating system. An 80386-based system using a UNIX System V kernel could simultaneously support 16-bit 8086 MS-DOS applications running in virtual 8086 mode; XENIX-based 16-bit 80286 applications using segmentation and paging; code ported from linear UNIX environments, like Berkeley 4.2, running in a paged linear model; and UNIX System V applications using segmentation and paging. Despite their diversity, such tasks could coexist, each in its own protected address space. The operating system could switch between environments in the space of a single instruction.

Future 80386 applications might include multiuser microcomputers running mainframe-size programs at mainframe speeds, engineering workstations executing sophisticated UNIX CAE programs concurrently with popular MS-DOS office-automation packages, and 80386-based superminicomputers for on-line transaction processing. ■

Canon presents your personal print shop.

Here's the laser beam printer from the company that pioneered desktop laser beam printing.

From the shortest memo to the longest report, the Canon LBP-8 A1 makes everything you print look hot off the presses. Your output will look so professional you'll think it came from a print shop.

The LBP-8 A1 is incredibly fast, producing copy at a rate of 8 pages per minute, and because it's non-impact, it prints so quietly that you'll hardly know it's working.

This versatile printer lets you use a wide range of plug-in fonts. And Canon's replaceable cartridge system makes it virtually maintenance-free.

To set up your own personal print shop, get the Canon LBP-8 A1, or the LBP-8 A2 with full graphics capability. You're sure to make a favorable impression with everything you print. To learn more about Canon's advanced laser beam printers as well as the complete line of wire dot matrix and bubble-jet printers, call 1-800-453-3307. (In Utah, 1-800-662-2500.)

Canon®
PRINTERS
Printouts that stand out.

Canon U.S.A., Inc., Printer Division, P.O. Box 619865, Dallas/Fort Worth Airport, Texas 75261. © 1986 Canon U.S.A., Inc. Printility is a product of Metro Software Inc.

Complete Systems Shipped Overnight

TURBO XT $799

• 4.77 & 6.77 mhz • 2 Drives
• 640K • Serial Parallel Clock
• Keyboard

BOARDS

MULTIFUNCTION

Quadram Quadport AT	CALL
Quadram Quadboard 384K	$179
Quadram Liberty	CALL
Quadram EMS Plus	$329
AST Sixpak 384K	$219
AST Sixpak Premium	CALL
AST Rampage XT/AT	CALL
AST Ramvantage PC/AT	Call New, Low Price
Thesis Fastcard V OK	New!
Ram Board 512K half card	$169

VIDEO/GRAPHICS

AST Preview	$289
Hercules Color	$179
Hercules Graphics Plus	CALL
IBM Color	$198
IBM Monochrome	$198
Quadram EGA	$375
Color Card w/par.	$129
Monochrome Graphics Herc. comp. w/parallel	$149

NETWORK

Orchid PCNET	CALL
Quadram Quadnet IX	CALL

EMULATION

AST 5251 mod 11/12	CALL
DCA IRMA	CALL
Forte Data 3278	CALL

MASS STORAGE

FLOPPY DRIVES

Half Height 360K (bl/grey)	$98
Half Height 1.2MB (gry)	$147
Tandon TM100-2	$125
Toshiba Half Height	CALL

INTERNAL HARD DRIVES

All hardware drives complete with controllers, mounting hardware & instructions. All are formatted, tested & have 1 year warranty.

Seagate 10MB	$399
Seagate 20MB	$499
Seagate 30MB	CALL
Seagate 20MB for AT	$599
Seagate 30MB for AT	$729
Priam 40MB/60MB for AT	$1395/1595
Priam 40MB/60MB for AT	$1425/1625

DISK CARDS

CMS 10MB/20MB	$499/699
Hardcard 10MB/20MB	CALL
Mountain 10MB/20MB	CALL

DISK SUBSYSTEMS

Bernoulli Box 20MB/40MB	New Low Prices
Sysgen Flatpak 20/20 & 20/60	CALL
Tallgrass all models	CALL

TAPE DRIVES

Irwin 10MB/20MB inter.	from$499
Irwin 10MB/20MB exter.	CALL
Tallgrass 4060	$1349
Sysgen Smart Image 20 & 60	CALL

MODEMS

Hayes 1200/1200B	$395/$329
Hayes 2400/2400B	$596/526
Migent 1200 Internal	New $109
US Robotics Password 1200/2400	CALL
US Robotics Courier	CALL
Ventel 1200 w/XTALK	CALL

AT&T 6300

$1899

• AT&T Monochrome Display
• 2 Drive • 640K • Complete

PC MART™

CHIPS

64K/128K/256K
8087/80287 6 HMZ & 8HMZ
CALL FOR MARKET PRICE

MONITORS

MONOCHROME

AMDEK 310A	$168
IBM Monochrome	CALL
Princeton Graphics MAX-12[(E)]	$175
Quadram Amberchrome	CALL

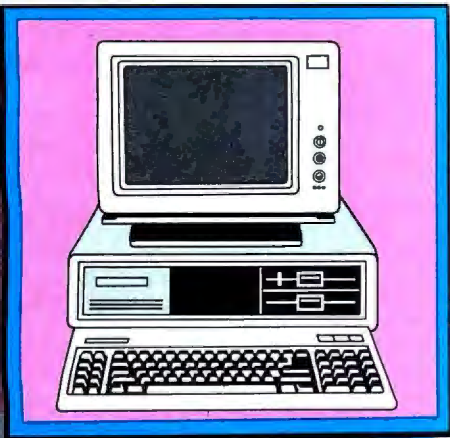
COLOR

Amdek 600/722	$429/528
IBM Enhanced Color	CALL
Mitsubishi Color Monitors	from $349
Princeton Graphics HX12/HX12E	$436/526
Quadram Enhanced color	NEW!

COMPAQ AT&T

Compaq products available in retail stores only
Visit our showrooms in ATLANTA, CHARLOTTE, DURHAM & RALEIGH.

The Oldest Computer Store in the South • Since 1975



*TURBO AT 286 $1899

- 6/8 mhz • 640K • 1-2 MB Floppy
- Serial Parallelport Clock
- AT Keyboard
- 20MB drive

1-800-241-0286

SUPPLY HOUSE

PC Mart Diskettes box	$18
case	$159
Maxell Disckettes box	$28
case	$258

PC MART™

Personal Computer Centers
1485 Northeast Express NE
Atlanta, GA 30329
INFORMATION
1-404-634-5995

IBM®

AT Enhanced $3495

• 6 mhz • 1-2 MB Floppy • 512K
• 20 MB Drive • Serial Parallel Clock

SOFTWARE

Ashton Tate Framework II	Call
Andrew Tobias Managing Your Money	$129
Ashton Tate dBase III Plus	Call
Borland Sidekick	$39
Borland Turbo Pascal w/BCD & 8087	$79
Borland Turbo Lighting	$65
Borland Prolog	Call
Borland Superkey	$49
Borland Reflex	$69
Clipper dBase Compiler	CALL
Computer Associates IUS	
Accting Series	from$299
Decision Resources Chartmaster	$239
Decision Resources Signmaster	$189
Decision Resources Diagrammaster	$259
Digital Research Gem Draw Collection	CALL
Digital Research Gem Colllection	CALL
Easy CADD	CALL
Funk Software Sideways	$48
Harvard Total Project Manager	CALL
Hayes Smartcom II	$89
IBM Displaywriter III	$369
Lattice C-Complier	$298
Llfetree Volkswriter Deluxe	$169
Microsoft Windows	$79
Microsoft Word 3.0	CALL
Microstuff XTalk XVI	$109
MultiMate latest	CALL
Multimate Advantage	CALL
Norton Utilities	$59
Open Systems Accounting Series	CALL
Prodesign II CAD software	CALL
Rosesoft Prokey	$99
Satellite Software Wordperfect 4.1	CALL
Software Publishing Co PFS Series	from $79
Sorcim Supercalc III	$219

PRINTERS

DOT MATRIX

Brother HR1509	$398
Brother 2024L	CALL
Brother Twinwriter 5 Dot Matrix & Letter Quality	CALL
Epson LX-86	CALL
Epson FX-85/FX-286	CALL
Epson LQ-800/LQ-1000	CALL
IBM ProPrinter	$425
Okidata 182	$239
Okidata 192/193	$385/525
Okidata 292/293	CALL
Okidata 2410	CALL
Toshiba 321/341/351	New Low Price!
Toshiba 351 Color	CALL

LASER/LETTER QUALITY

Brother HR15 XL	CALL
Brother HR25/HR35	$568/$798
Corona LP300 Laser	$2395
Epson DX20/DX35	CALL
IBM Quietwriter II	$1,259
Quadram Quadlaser	In Stock

PLOTTERS

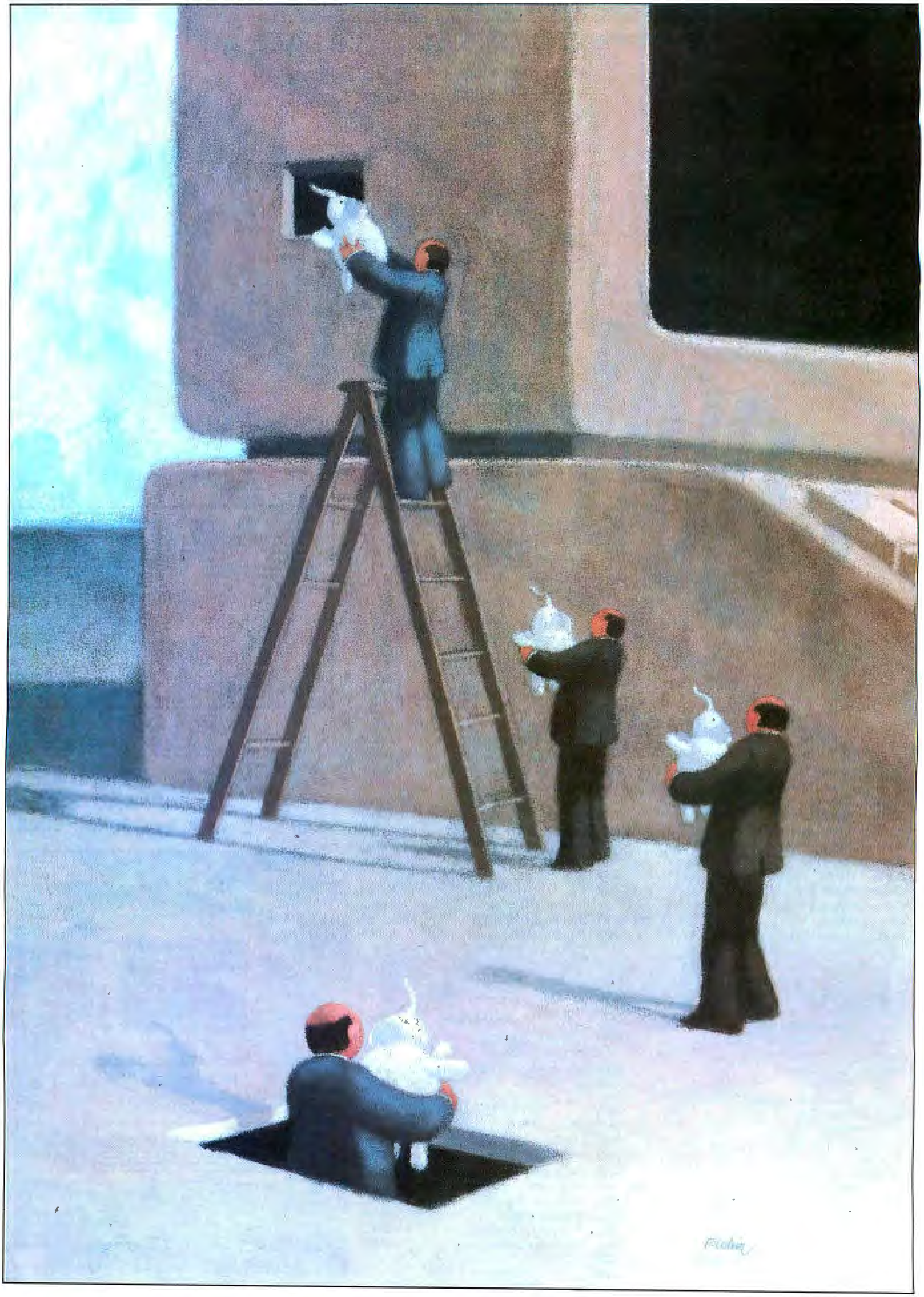
Houston Instruments	CALL
IO Line Plotters	CALL

all models available for quick delivery

Miscellaneous

Mousesystems PC Mouse w/paint	$139
Microsoft Mouse serial/bus	from$129
Keytronics 5151/5153	CALL
Keyboards for PC and AT	from $69
Quadram Microfazer 8K	$129
Quadram Microfazer II	CALL
Tripplite Isobar	from $45
Tripplite UPS 400	CALL
Curtis Protectors	from $29

**Monitor Not Included*



ILLUSTRATED BY ROB COLVIN

VIRTUAL MEMORY, VIRTUAL MACHINES

The 80386 offers virtual memory capabilities with memory protection

BY JON SHIELL

VIRTUAL MEMORY is a method for making the amount of memory that a program uses independent of the actual amount of RAM in the machine. In virtual memory systems, the disk memory effectively becomes the main memory, and the RAM becomes a temporary holding area for code and data that the processor is currently using. With virtual memory, the programmer doesn't care how much RAM (hereafter called *real memory*) the machine has. In fact, the program has no way of knowing how much real memory the system has. Instead, the program cares only about how big the virtual address space is.

The 8086 (which does not support virtual memory), the 80286, and the 80386 in real mode have a real address space of only 1 megabyte. The 80286 in protected mode has a virtual address space of 1 gigabyte and a real address space of 16 megabytes. The 80386 in protected mode has a virtual address space of 64 terabytes and a 4-gigabyte real address space. For comparison, IBM's latest mainframe architecture, 370/XA, can have virtual address spaces that add up to 128 terabytes and a real address space of 2 gigabytes. The 80286's 1-gigabyte virtual address space is composed of 16,000 segments of 64K bytes each. The 80386's 64-terabyte virtual address space is composed of 16,000 segments of 4 gigabytes each. [*Editor's note: The latest version of the* 80386 *is* B1.]

Virtual memory systems must provide hardware support for the translation of the virtual (or logical) addresses into real (or physical) addresses. During the translation, the hardware must be able to recognize those portions of code and data that are not resident in real memory. Such recognition is commonly called a *fault*. An interrupt must notify the operating system that such portions are needed and must be brought into real memory from the disk (see figure 1).

A virtual memory system must also be able to handle the situation where either the next instruction that is to execute or data that the current instruction needs is not resident. In such cases, the hardware has to fault to the operating system, which then brings in the required code or data. Then the operating system must be able to resume execution of the program at the instruction where the fault occurred without the fault being visible to the program. In other words, the instructions must be *restartable*.

A couple of ways exist to make an instruction restartable. For those instructions that write a single result, you can have the instruction issue all its accesses before it attempts to write the result. For more complex instructions that store multiple results, you have two options. The instruction has to either test all result locations to make sure they are resident before it actually stores to any of them or it must be interruptible: It must save enough state information to be able to resume execution at the point at which it was stopped.

VIRTUAL MEMORY SCHEMES

There are three kinds of virtual memory (VM) schemes. Segmenting schemes break memory into variable-length units called *segments*. Paging schemes divide memory into fixed-length units called *pages*. Finally, the hybrid scheme (like that in the 80386) uses variable-length segments that are divided into pages.

In virtual memory schemes, you have a trade-off between the amount of real memory in the system and the amount of virtual memory that it can support with reasonable performance. Typically, what you need is enough real memory to hold the working sets of a significant fraction of the programs that you want to run. Having less than this amount of real memory causes a tremendous increase in disk I/O due to paging or segment swapping. (Swapping of pages is called *paging*. Bringing something from a disk to real memory is a page-in operation; moving something from real memory to a disk is a page-out operation.) When the system reaches the point where a large amount of its time is spent paging in and paging out instead of executing programs, it is said to be *thrashing*.

The 8086/8088 microprocessors cannot implement virtual memory because they don't have the required address-translation hardware on-chip and they don't support an off-chip memory management unit. The only memory management that they can perform is overlays, which require that the program explicitly manage its own swapping of resident and disk code and data. Almost all big MS-DOS programs use overlays: Lotus 1-2-3 and WordStar are two examples. DOS provides explicit support for overlays via a DOS function call. Overlays have certain disadvantages. First, an individual section of an overlay can be

(continued)

Jon Shiell is a system architect and microprogrammer. He can be reached at P.O. Box 61195, *Sunnyvale*, CA 94088.

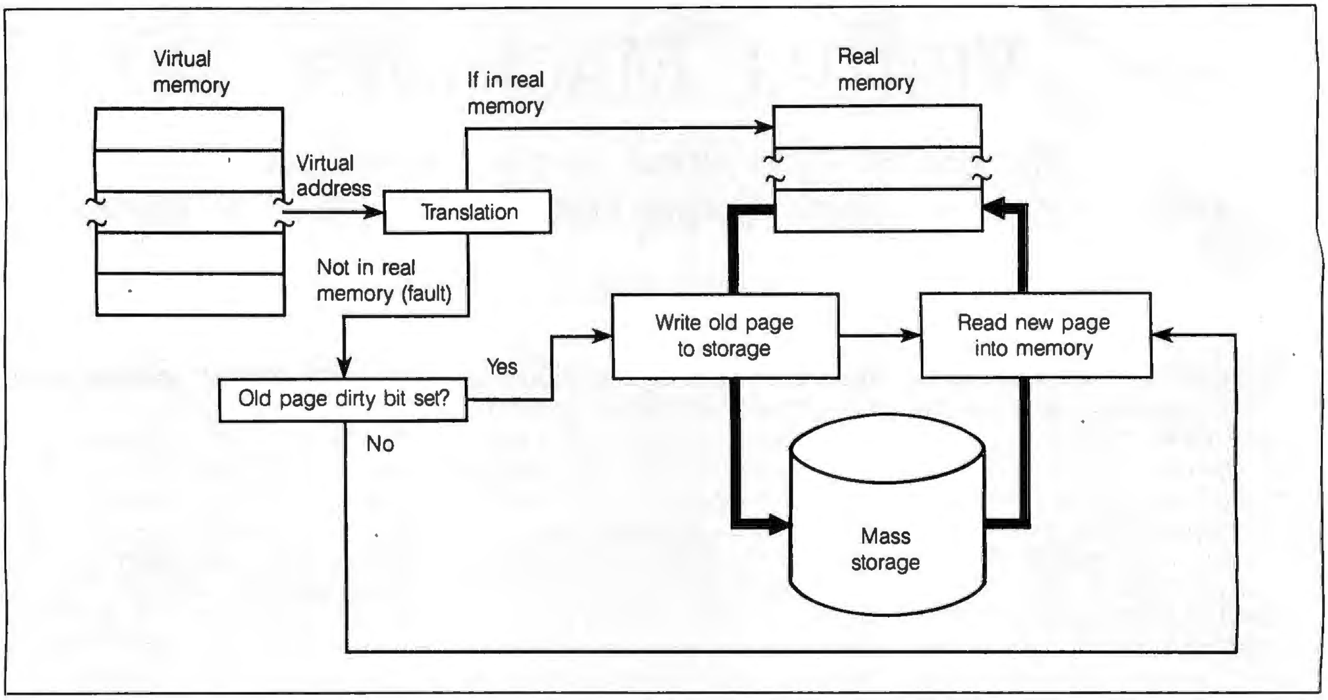


Figure 1: A *virtual memory system.*

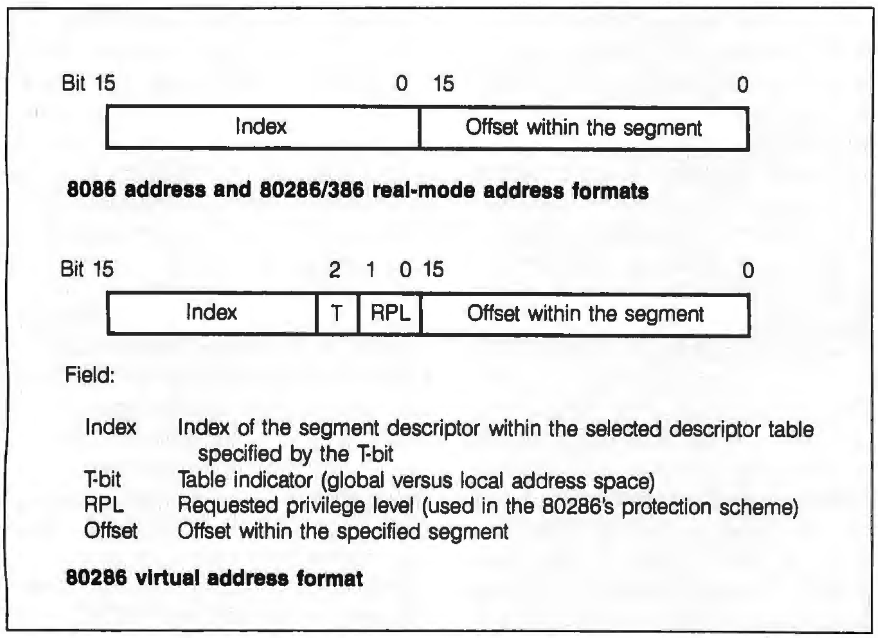


Figure 2: *Addressing formats for* iAPX *microprocessors.*

no larger than some fraction of the available real memory. Second, the programmer is responsible for managing the overlays.

The IBM PC multitasking environments (like TopView, CCP/M, and DESQview) do not fully implement virtual memory in that each program can use only what real memory is available to it. But by using time-slicing and swapping, such systems make memory appear large enough to hold multiple programs. Because the 8086/8088 doesn't provide any memory protection, it is possible for a program to corrupt the operating system or anything else in memory.

The 80286 VM Scheme

One of Intel's major enhancements in defining the 80286 was a protected mode that includes the logic required to implement VM. (From now on, references to the 80286 or the 80386 are to their protected modes.) The virtual addresses in the 80286 are split into two 16-bit segments. The first half is a segment selector that specifies the segment to be used. The other half is the offset within the segment, thus permitting access to any byte within a 64K-byte address space. The format of these two halves is shown in figure 2, along with the 8086/8088 addressing scheme.

Note that the format of an 80286 segment selector differs from that of the 8086. This is the major reason that many programs written for MS-DOS or other 8086 operating systems won't run in 80286 protected mode. The problem is that addresses in protected mode cannot be treated as 32-bit integers because of the T and RPL fields in the segment selector.

The presence of these fields means that the 8086 method of doing simple 32-bit arithmetic for addressing structures bigger than 64K bytes doesn't work in protected mode. A couple of methods that will work are detailed in listing 1. Also note that the 80286's segments—like the 8086's—can be from 1K to 64K bytes long. However, the 8086 segments start on a 16-byte boundary; thus, the segment number can overlap the apparent offset (see figure 3). The 80286's segment numbers do not overlap the offset values, so they appear on 64K-byte boundaries in the virtual address space.

Intel has solved this 8086 addressing compatibility problem in the 80386 by providing an 8086 virtual machine mode that I will discuss in more detail later. In this mode, even though the machine is running in protected mode, the 80386 uses 8086-type selector and addressing.

The 80286 (used in the IBM PC AT and its clones) implements virtual memory at the segment level, that is, by swapping whole segments. Thus, address translation in the 80286 consists of remapping segments from one spot within the virtual address space into another within the real memory (see figure 4). This is done by providing a P-bit (present bit) in the segment descriptor that indicates whether or not the segment is present in real memory. When software attempts to load a segment register with a descriptor whose P-bit is 0, the hardware takes an interrupt of type 11 or, for stacks, interrupt type 12 (see table 1).

This interrupt lets the operating system fault handler load this segment into real memory, possibly overwriting a current segment. If the segment to be overwritten has been modified since being brought in from disk, the operating system will write the modified segment out to disk before overwriting it. Once the required segment has been brought in, the fault handler can return to the interrupted program, letting the program continue operation without knowing anything happened. The instruction pointer that is saved on the stack points to the first byte of the faulted instruction. This facilitates restarting the interrupted program from the fault handler via the IRET (interrupt return) instruction.

THE 80386 VM SCHEME

The 80386 overcomes the addressing compatibility problems between the 80286 and 8086 by implementing both of their addressing modes as well as its own 32-bit native addressing mode. There is no specific 80286 compatibility mode. Instead, each time the 80386 switches between tasks, it specifies the mode of its target program (see figure 5) via the TYPE and D bit fields of the code-segment descriptor. Unlike the method used for 80286 compatibility, 8086 mode is explicitly specified by the VM bit in the processor's FLAGS register.

In addition to the segment-level protection provided by the 80286, the 80386 provides page-level protection when paging is used. Since the 80386 first performs segment translation, followed by page translation, it is possible to have two or more small segments share a single page. In this

(continued)

Listing 1: *Pseudocode example of address arithmetic in 80286 protected mode.*

```
Adding a displacement of 16 bits or less:  (offset in DX, displacement in AX)
    Add new displacement to DX.
    If carry-out then
       Move DS into BX.
       Add '0008'x to BX.     (add 1 to index field)
       Move BX back into DS.

Adding a displacement of more than 16 bits: (offset in DX, displacement in BX:AX)
    Add new displacement to DX.  (DX <- DX + AX)
    Add carry-out to BX.
    Left-shift BX by 3.  (align with index field)
    Move DS into CX.
    Add CX to BX.     (add displacement high to index field)
    Move CX back into DS.

80386 handling of displacements of up to 32 bits
    in native mode: (offset in EDX, displacement in EAX)

    Add EDX, EAX
```

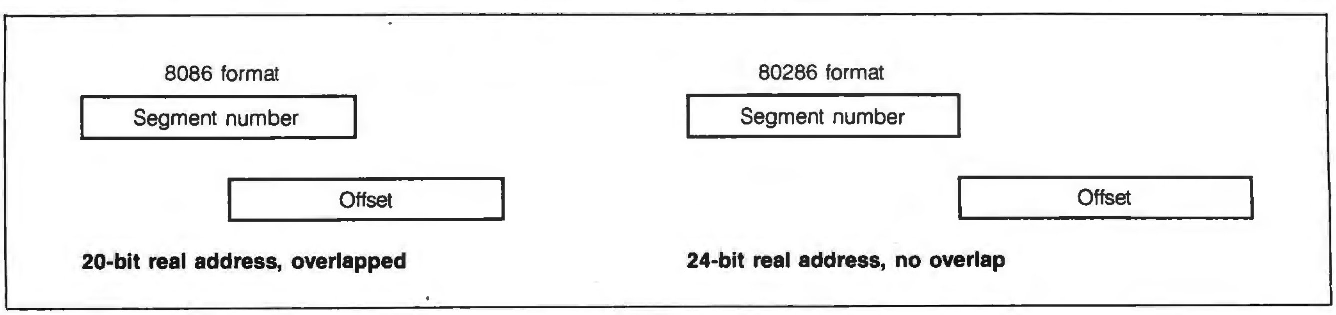


Figure 3: *Differences in address translation between the 8086 and 80286.*

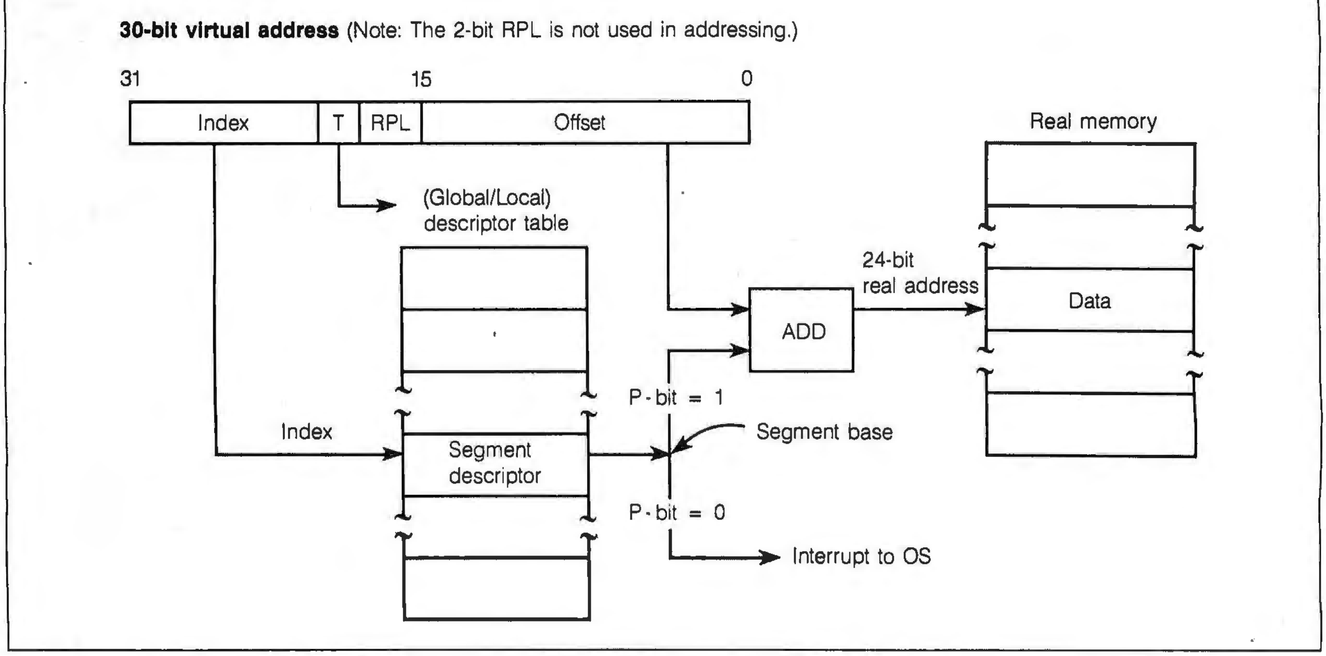


Figure 4: *80286 virtual address translation.*

Table 1: *80386 interrupt vectors.*

Function	Interrupt Number	Instruction That Can Cause Exception	Return Address Points to Faulting Instruction
Divide error	0	DIV, IDIV	Yes
Debug exception	1	Any instruction	Yes
NMI interrupt	2	INT 2 or NMI	No
1-byte interrupt	3	INT	No
Interrupt on overflow	4	INTO	No
Array bounds check	5	BOUND	Yes
Invalid op code	6	Any illegal instruction	Yes
Device not available	7	ESC, WAIT	Yes
Double fault	8	Any instruction that can generate an exception	
Invalid TSS	10	JMP, CALL, IRET, INT	Yes
Segment not present	11	Segment-register instructions	Yes
Stack fault	12	Stack references	Yes
General protection fault	13	Any memory reference	Yes
Page fault	14	Any memory access or code fetch	Yes
Coprocessor error	16	ESC, WAIT	Yes
Intel reserved	17–32		
2-byte interrupt	0–255	INT n	No

case, the page-level protection would not be used. Note that the operating system can still use the paging scheme for memory management; however, care must be taken to ensure that the page is swapped out only when all the segments in the page are not in use.

The 80386 VM scheme implements virtual memory via mainframe-style paging schemes using two-level translation tables. The first level, pointed to by CR3 (control register 3), is the page-directory table containing from 1 to 1024 page-directory entries of 4 bytes each. Each of these PDEs points to a page containing page-table entries of 4 bytes each, for up to 1024 pages. Each PTE contains the address of a 4K-byte block in real memory called a *page frame.* Thus, a PDE points to PTEs describing up to 4 megabytes of real memory. Both the PDT and each page table will not exceed one page (4K bytes) in length and are aligned on page (4K-byte) boundaries. Also, the pointer in CR3 and the pointers within the PDEs and PTEs are real addresses, not virtual addresses. If they were not real addresses, you would run the risk of your page tables being paged out, effectively hanging the system.

Figure 6 shows the format and contents of a PTE. The accessed and dirty bits of a PTE are used for memory management. The OS will use the accessed bit to keep track of how often a program uses a page and the dirty bit to tell whether the page must be copied out to the disk before the new page overwrites it. The standard method of tracking a program's page use is to have the operating system periodically clear the accessed bit of all pages a program has in real memory and then check, after a certain period, which pages have their accessed bits set. This information, collected over a period of time, tells which pages a program is currently using. This set of pages is called the program's *current working set.* Also, a P-bit indicates to the processor whether this page is present in real memory. If the bit is off, the processor takes a page fault to the operating system so

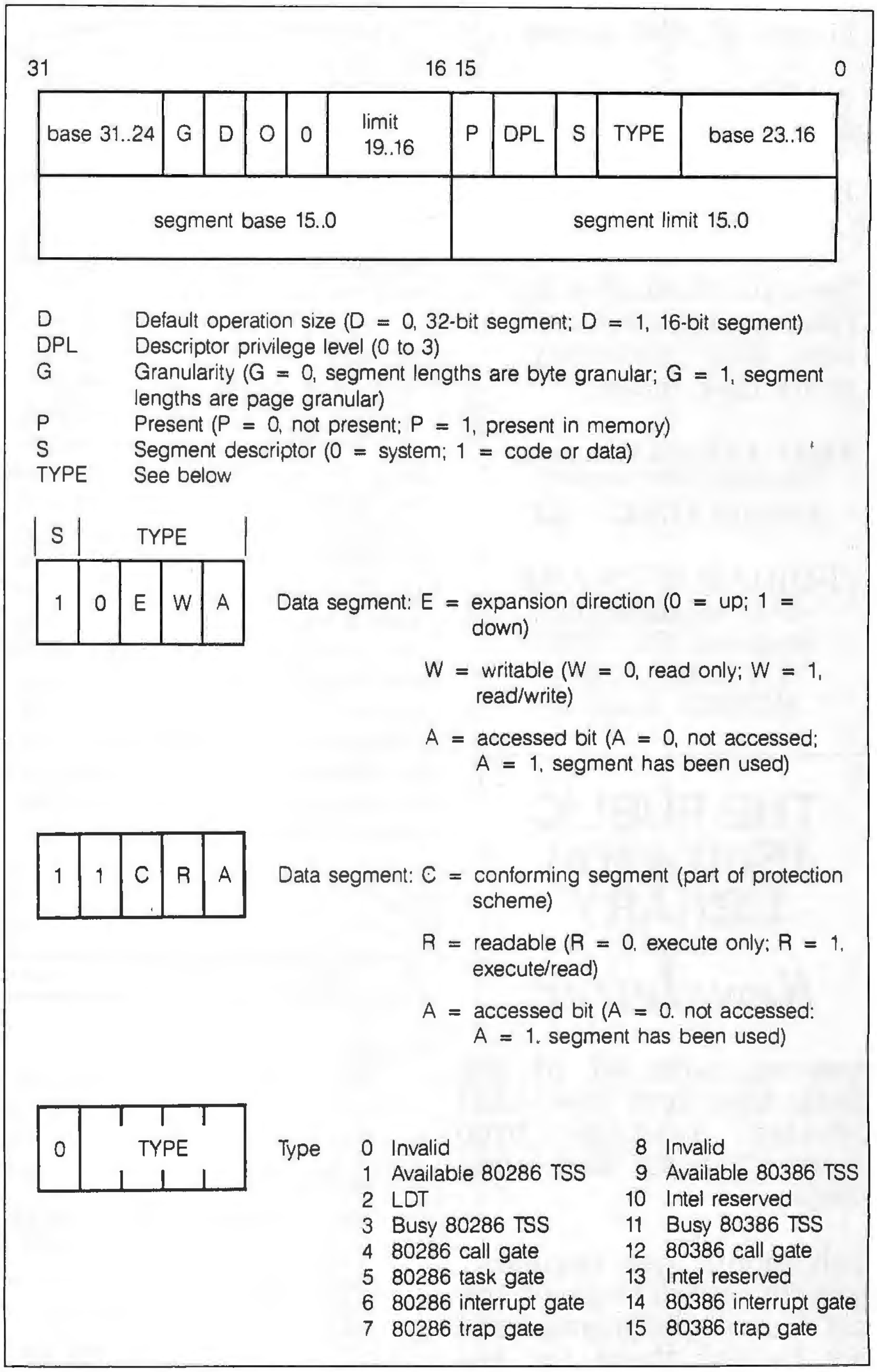


Figure 5: 80286/80386 *segment-descriptor format.*

it can bring in the page from disk. The process of bringing in a page when a program faults on it is called *demand paging.*

The translation of virtual addresses requires two accesses—the first to get the PDE and the second to get the PTE—for each access done by the program. Rather than run at less than one-third of its potential speed, the 80386 uses an on-chip address cache called a translation-lookaside buffer. This cache, unlike a "normal" cache, returns a PTE—instead of a data word—when given an address. The

(continued)

80386 has a 32-entry, four-way set (that is, four sets of eight PTEs each) associative TLB, which should give a hit rate of about 98 percent (see figure 7). In other words, 98 percent of the time the TLB will contain the PTE that you need. The other 2 percent of the time, the processor must actually do the translation and then load the new PTE into the TLB.

The dirty bit indicates that the page has been altered by the processor since being brought into real memory. When a program does a write to memory, the processor sets the dirty bit in the PTE. What actually occurs is that the 80386 sets the real dirty bit on the first write, because a copy of this bit is kept in the TLB. Note that the 80286 segment descriptor doesn't have a hardware dirty bit. The way that an 80286 operating system tells that a segment is dirty is to initially make all the program's segment descriptors write-protected when they are brought in from disk. When the program attempts to write (first time only) to the segment, an exception occurs and the operating system takes control. The operating system sets a software dirty bit, removes the write-protection from the segment, and restarts the task.

Effects of Virtual Memory

For a system based on segment swapping (the 80286), the only requirement is that I/O can't cross a segment boundary. The OS must load the starting real address into the direct-memory-access unit instead of the starting virtual address, unless the DMA unit is capable of handling the translation and knows where the tables are. In either case, the OS must not swap out the segment until I/O finishes. Thus, the operating system must "lock" a segment into memory. A software-only bit in the descriptor is normally used for this purpose.

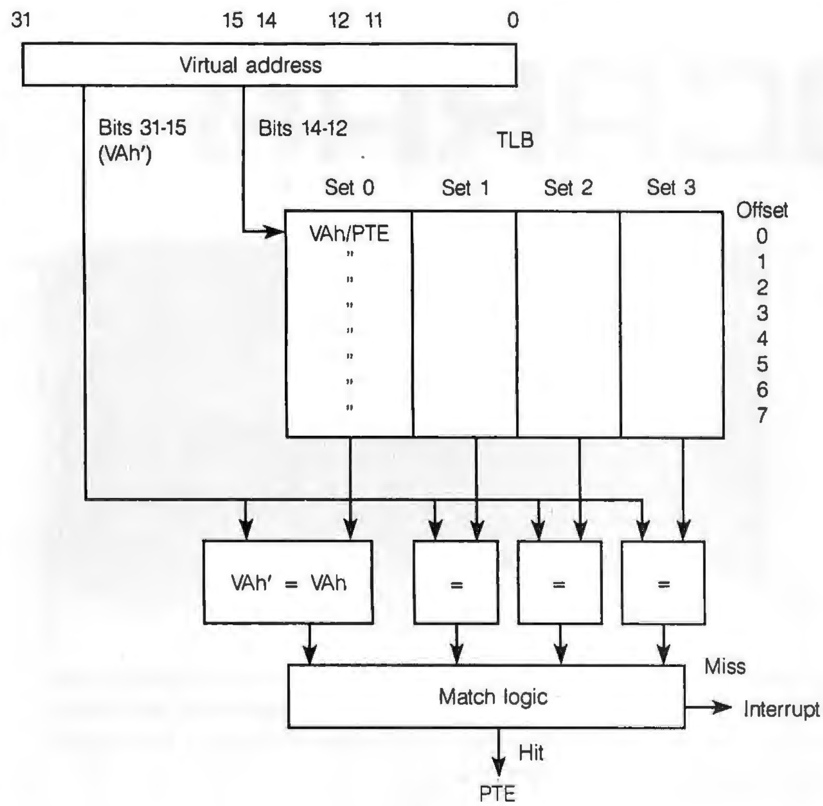
For a paged system, things are further complicated by the fact that a segment can span more than one page. This leads to three possible ways to handle DMA. In the first and simplest (this is the case for PCs so

Bit 31–12	11–9	8	7	6	5	4	3	2	1	0
Page-frame address (PFA)	OS Res	0	0	D	A	0	0	U	R	P

Field:

PFA	Real address bits 31-12. If it's a directory entry, this is the real address of a page frame containing the page table. If it's a page-table entry, this is the address of a page frame for program use.
OS Res	Reserved for operating system use.
D-bit	Dirty page bit. If a 1, the page has been modified; it's set when a write to an address in the page occurs.
A-bit	Accessed bit is set to 1 when any kind of access (read/write/execute) occurs to an address in the page.
U-bit	User access permitted. If a 1, access is permitted by level-3 programs. If a level-3 program attempts to access a page with a U-bit of 0, an exception occurs.
R-bit	Level-3 access control (assuming U-bit is 1). If a 0, only read/execute accesses are allowed; if the R-bit is a 1, writes are also permitted.
P-bit	Present bit. If P=1, page is present in real memory; if P=0, page is swapped out. An interrupt is generated when a program attempts to use this page. If P=0, the rest of the word is undefined from hardware's point of view.

Figure 6: 80386 *page-directory/table-entry format.*



How a TLB works:

Use a portion of the VA, bits 14-12 in this case, as the index into the TLB arrays. Read out one entry per set in the TLB, four entries in this case. The entries that were read out consist of three parts: VA entry, PTE, and control information. Compare the VA in each entry with the VA you are looking up. If any matches, you have your translation (unless control indicates this entry is invalid). Otherwise, signal a TLB miss to the processor so it can get the correct translation.

Figure 7: *Sample diagram for a 32-entry translation-lookaside buffer.*

far), the DMA unit doesn't know about paging and the processor must set up each page transfer separately. In the second way, the DMA unit knows how to handle a string of real addresses and counts that point to each page that I/O is to address and specify the number of bytes to be moved. Finally, the DMA unit itself can use virtual addressing, in which case it must be given a translation table address in addition to a starting address and count. Of course, the pages that I/O will be addressing should be locked.

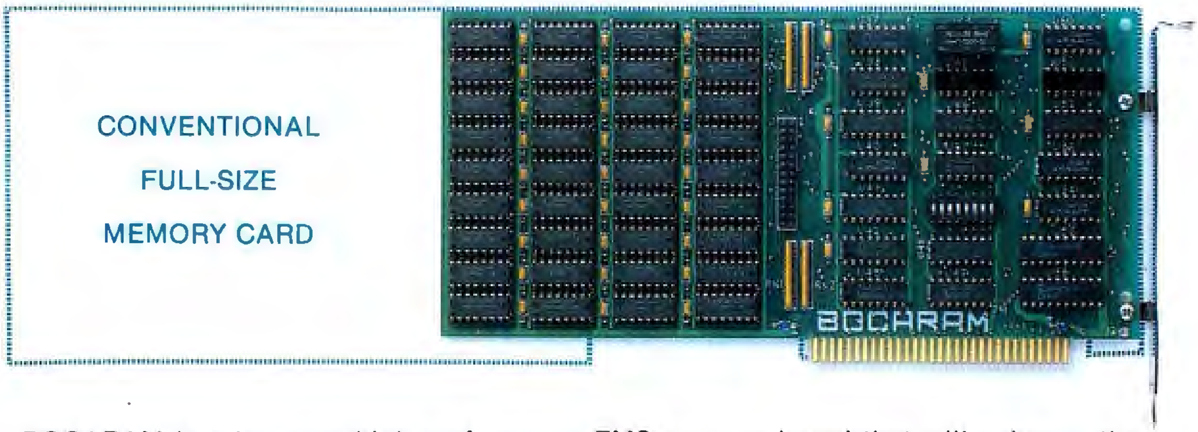
An 80386 running in 8086 virtual machine mode treats segment descriptors as the 8086 does: They are simple 16-bit integers that are multiplied by 16, then added to the 16-bit offset to get a 20-bit virtual address. Unlike the 8086, the 80386 then performs page translation on the address. Thus, the 1-megabyte address space of an 8086 program is split into 256 4K-byte pages. Using paging to remap the address space permits the sharing of pages that are not written to. This reduces the real memory required to support multiple 8086 virtual machines. Also, the host operating system can use the remapping function to provide page-level protection for the guest operating system.

(continued)

EMS
ABOVE BOARD
COMPATIBLE

A low cost mid-size card for oversize jobs

BOCARAM™



BOCARAM is a low-cost high performance EMS memory board that will enhance the power of your PC. BOCARAM conforms to the Lotus/Intel/Microsoft Expanded Memory Specification whose growing list of supporters includes over twenty leading software manufacturers like Lotus 1-2-3 (Release 2.0) and Symphony.

BOCARAM is designed to connect to a PC/XT 8 bit bus, but works equally well in the PC-AT and compatibles with clock speeds up to 12 MHz.

A versatile memory management software package speeds installation and provides a RAMdisk and print spooler. BOCARAM will co-reside with other memory boards such as Intel's Above Board or Boca Research's MEMEK.

The Boca Research family of memory and graphic add-on products is available from a growing network of dealers. All products are supported by a qualified technical support staff. Boca Research products are designed and manufactured in the U.S. to meet the highest standards of quality control. All products are protected with a one year warranty.

Suggested Retail Prices	
256K	$245.00
1Meg	395.00
2Meg	740.00

For additional information call your Boca Research dealer or contact us directly at:

Specifications

Board Size:	4.25 X 8.5 inches
CPU Speed:	4.77 to 12 MHz.
RAM Chips:	150 NS 256K

LOTUS INTEL MICROSOFT
EMS COMPATIBLE

BOCA RESEARCH, INC.

6401 CONGRESS AVENUE, BOCA RATON, FLORIDA 33431 (305) 997-6227

Registered Trademarks:
IBM PC/XT/AT, INTEL, ABOVE BOARD, LOTUS, MICROSOFT

WHY NO 80386 VIRTUAL MACHINE?

One reason the 80386 cannot "virtualize" itself is the asymmetric behavior in its instruction pairs, such as PUSHF and POPF. PUSHF saves the REAL flag values in protected mode. POPF will not cause a general-protection exception in protected mode. Even if an attempt is made to alter the I/O privilege level or interrupt flag bits without sufficient privilege level, no interrupt will occur for an unauthorized alteration attempt; it is simply ignored.

Consider the attempt to virtualize XENIX-286 under the current 386 architecture. The XENIX kernel may use the POPF instruction to alter the current IF bit. This works in native 286 mode (no VM). Under VM, the hardware will not allow the change, and no notification will occur (via a general-protection exception) that an attempt to alter the IF bit was made. As far as the host CP is concerned, the guest XENIX OS is still able to receive interrupts from the outside. If XENIX is in the middle of altering a critical area where interrupts can't be tolerated (such as stack changes or modifying the global-descriptor table) and the host CP has no way of knowing its guest's state, the interrupt is passed to XENIX at the wrong time, causing the guest to crash.

A similar problem exists with the task-register load and store instructions (LTR/STR). Again, consider the attempt to virtualize XENIX-286. It's normal for XENIX to store its current task register during preparation for a task switch (many operating systems do this). In the current 386 architecture, if XENIX attempted to load the task register, this would cause a GP exception and appropriate simulation by the CP could be done. However, when XENIX stores its task register (which has been virtualized), no GP exception occurs, and the real (CP) task register is stored. If XENIX had stored the TR with a value, say, of 21F (a XENIX task-register value), the LOAD might return a value of 1C (the real CP value). Any internal bookkeeping XENIX was doing that depended on LTR/STR being a matched set is destroyed, and bugs will result.

The final problem with virtualizing the 80386 relates to addressing problems. Part of the difficulty is that the CP must share the same address space with the guest. The other part of this problem deals with interrupts: For a virtual machine interrupt, the address that you transfer to is a virtual address. This causes the following problem when attempting to run an 80386 virtual machine. Say that a guest OS is using linear address 8000000 for some control information. Suppose that the host CP also uses that address for some purpose. Due to the nature of the 80386, the host CP cannot set up the page tables to prevent this, and data is overwritten.

It is even possible to support virtual expanded memory.

VIRTUAL MACHINES

A virtual machine is the extension of the concept of virtual memory to an entire machine, including I/O devices and privilege levels, so that a program can't tell that it doesn't have a real machine—in this case an 8086—to itself. A virtual machine lets multiple operating systems (which may be incompatible) run concurrently with no modification or conflict. A program running in a virtual machine sees whatever memory and I/O the host operating system desires, not the amount of memory in or the I/O devices of the real machine. It is important to note that a virtual machine is not an emulator, nor is a virtual machine as efficient as a real machine.

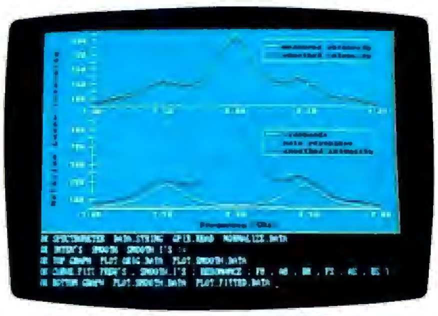
Why use virtual machines in the first place? Because this permits an 80386 system to run multiple PC-DOS programs unmodified in what appears to be their native environment. Thus, the

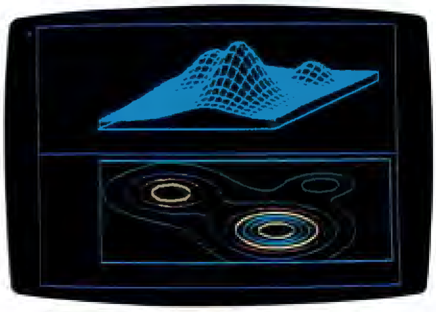
(continued)

Still the only integrated scientific analysis, graphics, and data acquisition software.

ASYST.™

Macmillan's widely-acclaimed ASYST™ Scientific Software still offers more built-in capabilities than any other PC software. Its unique interactive programming environment includes such functions as FFT, smoothing, integration, differentiation, curve fitting, statistics, differential equations, and matrix and polynomial operations. And all functions are supported by sophisticated graphics, such as error bars, scatter plots, and axonometric plotting.





ASYST's extensive function set provides the building-blocks that let you assemble scientific applications – fast. RS-232 and optional A/D and GPIB/IEEE-488 support offer interface versatility you just can't get from other products. And all hardware support integrates seamlessly with the full range of ASYST capabilities.

Free technical support. 60 days of free technical support help you design, set up, and fine-tune the perfect system. Ongoing support is available at a nominal cost.

30-Day No-Risk Offer.
CALL 1-800-348-0033.

In New York state, (212) 702-3241.

Ask for technical literature, including performance benchmarks.

Macmillan Software Co.

An affiliate of Macmillan Publishing Company
630 Third Avenue, New York, NY 10017

ASYST is a trademark of Macmillan Software Co. ASYST was developed by Adaptable Laboratories Software, P.O. Box 18448, Rochester, NY 14618 need to modify existing programs so that they run in protected mode is avoided. This ability to run multiple operating systems lets the 80386 run an 8086-type UNIX or CP/M-86 at the same time, in addition to the native-mode user interface.

To support multiple virtual machines, a control program (CP) actually manages the real machine's resources (real memory, I/O, privilege levels). The CP creates and handles multiple virtual machines, distributing the real machine resources in an orderly fashion. The CP is not an operating system: It is a layer of software that mediates requests from the operating system to the hardware.

For a processor to support virtual machines, all instructions that would let a program be able to tell that it is running in user mode must be privileged. Access to some facilities, such as I/O and timers, must be restricted. There must be no possibility of a conflict between a guest address space and the host operating system's address space. This becomes especially evident when you attempt to run a copy of the CP in a virtual machine.

While the current 80386 supports virtual 8086 machines, it does not support virtual 80286 and 80386 machines. This is due to problems in handling the PUSHF and POPF instructions and the store-system-register instructions (STR, SGDT, SLDT, SMSW) and to the fact that the address to which an interrupt transfers control is a virtual address using the local-descriptor tables and global-descriptor tables at the time the interrupt occurred. (For additional information, see the text box "Why No 80386 Virtual Machine?" on page 119.) While the lack of ability to run a virtual machine other than an 8086 is a real limitation, it doesn't seem to be a fatal flaw. Intel might fix this in a later release of the chip.

The 80386 CP invokes an 8086 virtual machine by performing a task switch where a program's task-state segment has the 8086 virtual mode bit (bit 17) set in its copy of the EFLAGS register. Virtual 8086 programs run at the lowest privilege level, so to utilize specific privileged instructions, the I/O permission-level field in the EFLAGS register and the I/O permission bit map (located in the task-state segment) must also be set up. Now any interrupt or exception causes the 80386 to switch back into protected mode (native 80386 mode) before invoking the handler pointed to by the interrupt-descriptor table. Thus, the routines that are invoked must be 80386 native-mode routines. In most cases, these routines will emulate the operation requested by the virtual machine—for example, an OUT instruction—then return to 8086 virtual mode via an IRET instruction.

When the 80386 runs in virtual 8086 mode, it does address generation like an 8086 but the 20-bit address that is produced is a virtual address. The page-translation hardware then translates the virtual address into a page number and a starting offset within that page. This lets the 80386 CP use a page-level protection "below" the guest operating system, so that elements such as ROM can be emulated (just mark the page "read-only"). This same method lets the host share portions of memory between multiple virtual machines. So while the 8086 virtual machine can't use the segment-level protection that is available to both 80286 and 80386 mode programs, the 80386 CP can use protection at the page level to enhance real memory utilization of the guest machines.

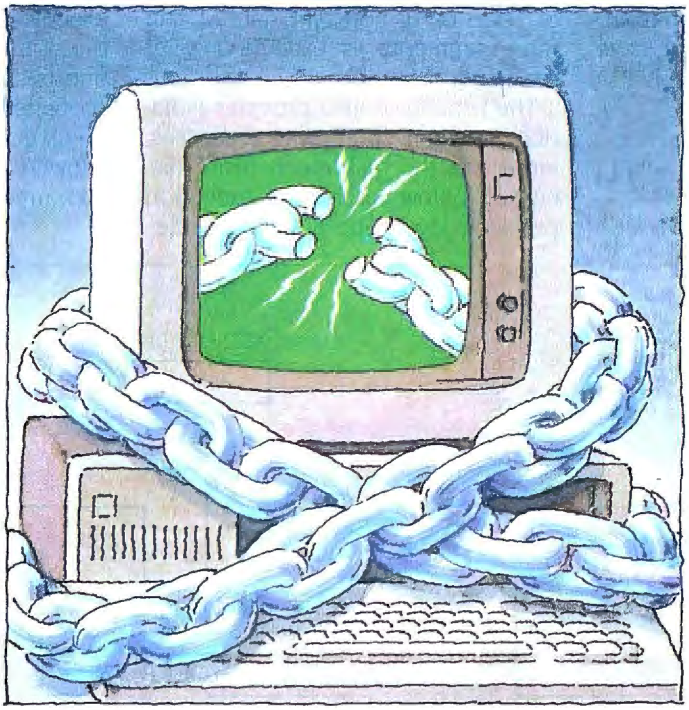
THE WRAP-UP

Virtual memory lets you run a program even if your system doesn't have enough real memory. It also makes multitasking much safer by providing the ability to isolate tasks from one another. The virtual machine mode will protect your investment in 8086 software while providing a relatively painless method of entrance to virtual memory, multitasking, and the 80386 itself. If Intel corrects the inability of the 80386 to "virtualize" itself and the 80286, the same path provided for 8086 software will be available for 80286 software such as XENIX and protected-mode DOS. ■

A PROTECTED-MODE PROGRAM FOR THE PC AT

A program that lets you tap the 80286's power by using its native mode

BY ROSS P. NELSON



THE COMPUTER industry usually ties new-generation hardware to the older hardware by making it compatible with existing software. To take advantage of the machine's newer features, you must write software for its native mode. Such is the case with the IBM PC AT's central processor, the Intel 80286, which runs software written for the 8086/8088 processors.

Due to lack of operating system support, very little software has appeared for the 80286's native mode, called the protected virtual address mode, leaving the processor's power largely untapped in current microcomputer systems. Ironically, applications programs that adhere to a few simple rules will run unmodified in the native mode, but only if you dramatically modify the operating system. I've written a program, PM_AT.EXE, that places an IBM PC AT in its protected mode and provides a base for future expansion and experimentation with the protected mode in the 80286. [*Editor's note*: PM_AT.EXE, PM_AT.ASM (*the source code for* PM_AT.EXE), *and* PROTECT.INC, *a module for the program, are available on disk, in print, and on* BIX. *See the insert card following page* 176 *for details. Listings are also available on* BYTE*net*. *See page* 4.]

PROCESSOR OVERVIEW

In the 8086 emulation mode (which is equivalent to the standard operating modes of the 8088, 8086, 80188, and 80186), the 80286 has one type of system object, the segment. The processor grants access to a segment when a value is loaded into a segment register. The processor then allows direct access to the 64K-byte block of memory beginning at location <value> × 16.

In protected mode, the processor views the system much the same as an operating system would. The processor "knows" about objects other than memory segments. Each object is referenced by a descriptor, which contains information about the object. When the system is in protected mode, the value loaded into the segment register does not point to the object itself, but to a table of descriptors where the object is described. From that table, the physical address and size of the object (assuming it is a memory segment) are extracted. Memory segments can range from 1K to 64K bytes in size, and the processor verifies that every memory access is within the bounds of the segment size (see figure 1).

In addition to segments, the 80286 works with descriptor tables, task segments, and gates. A particular type of descriptor is used to reference each object (see figure 2). The processor uses descriptors to ensure that system security is not violated. To prevent a valuable program from being disassembled, the processor would use an execute-only descriptor for the code segment; any program that tried to read that segment as data would fail.

The 80286 requires two tables of descriptors whenever the system is running in protected mode: the global descriptor table (GDT) and the interrupt descriptor table (IDT). A number of local descriptor tables (LDTs) can also exist. LDTs are associated with the 80286 task implementation. The GDT and LDTs contain descriptors for code and data segments. The IDT is analogous to the 8086 interrupt vector table and contains special code descriptors known as gates.

(continued)

Ross P. Nelson (1821 Ashmeade Ct., San Jose, CA 95125) is a systems programmer.

ILLUSTRATED BY BENJAMIN PEREZ

In 80286 parlance, the 16-bit value loaded into the segment register is called a selector. A selector accesses a system object by selecting the descriptor for the object. A selector has three components: the table indicator (TI), which selects either the GDT or the currently active LDT; the desired descriptor within the table (Index); and a requested privilege level (RPL). Normally, the RPL is the same as the operating privilege of the currently executing code. The 16 bits that are loaded into the segment register have the following format:

```
15 3                2    1 0
-----------------------------
|  INDEX            |TI| RPL|
-----------------------------
TI = 0 / GDT,  TI = 1 / LDT
```

For example, the processor interprets the value 13A1 hexadecimal in a segment register as a request to access descriptor 274 hexadecimal in the GDT with a privilege of 1.

PROTECTION

In protected mode, as the name implies, the 80286 is concerned with system integrity. It makes every effort to ensure that the flow of execution is well behaved. To prevent corruption of code or of constant data, it can mark segments as execute-only or read-only.

The processor also provides isolation between coresident processes, either via the task mechanism (discussed below) or by the execution privilege level. The 80286 provides four levels of security, numbered from 0 (most secure) to 3 (least secure). A program may access any descriptor with a descriptor privilege level that is numerically equal to or higher (less secure) than its own execution level. Typically, operating system code executes at the most privileged levels, and the applications execute at less privileged levels.

This arrangement presents a problem when an application attempts to execute subroutines that are part of the operating system, such as I/O requests. The code segments should be protected from corruption, but less privileged applications should still be allowed access. The 80286 provides two mechanisms to deal with this

(continued)

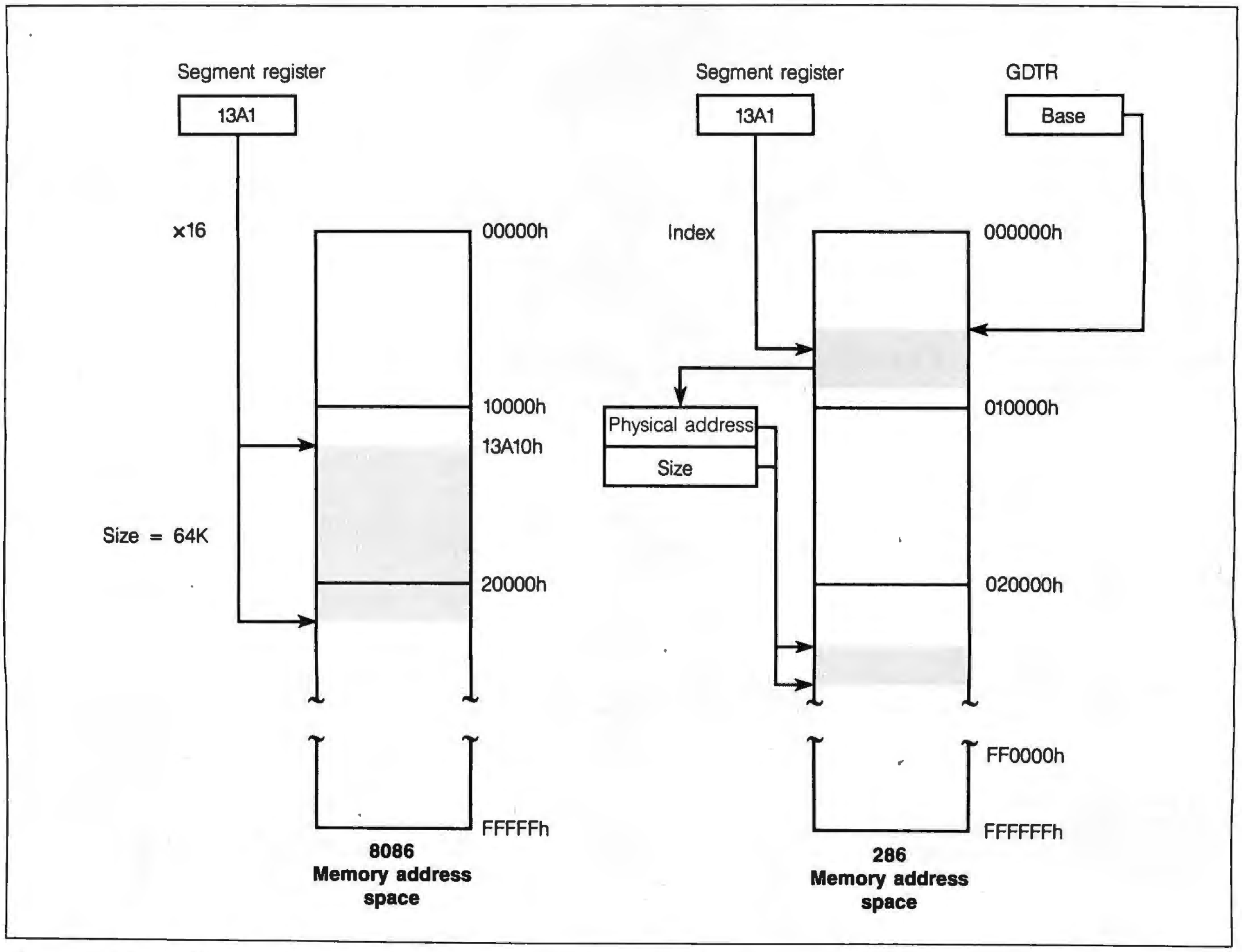


Figure 1: *Direct addressing of real-address mode versus the indirect addressing of protected mode.*

QUALITY: an American achievement.

Ask your dealer about FREE 90 day On-site Service*

AMERICAN 286-A™ PERFECT SYSTEM

Purchasing a compatible is a wise decision. You can find powerful systems at competitive prices without compromising quality.

The American 286-A™ (formerly American AT) is an IBM AT® compatible with a proven record in reliability. The four-layer motherboard ensures low noise and minimal interference. It passes IBM's diagnostics test with flying colors. Supported by only the best, the American 286-A uses major national brand components for data storage units.

The American 286-A offers unprecedented compatibility. Supporting all popular business and education applications and networking systems, the American 286-A emulates IBM operations.

And, the American 286-A won't leave you stranded. We back the American 286-A with a 12 month warranty on parts and labor.

System characteristics:

- 6MHz/8MHz switchable clock speeds.
- Up to 3.5 Mb memory expandable.
- Includes MS-DOS® 3.1 software and system manual.
- Enhanced chassis design.
- 12 month limited warranty on parts and labor.
- Reliable 4-layer motherboard.
- Socket to support the 80287 math co-processor.
- Eight I/O expansion slots.
- AT/XT switchable keyboard.
- 1.2 Mb diskette drive with controller.
- Color or monochrome graphics card.
- Serial/parallel interface card.
- 33 Mb (formatted) fixed disk drive with 40 ms access time.
- 20 Mb tape backup.
- American Mouse with free Compu-Brush graphics program.
- FCC approved; class 15J, part B.

AUTHORIZED DISTRIBUTOR AND SERVICE CENTERS:

California Micro, Inc.
Los Angeles, CA (800) 792-6500

Imagine Computers
Goleta, CA (800) 344-2964

PC Land, Inc.
Tustin, CA (714) 730-6723

Computer Professionals
Lakewood, CO (303) 232-4009

Computer Peripheral Warehouse, Inc.
Deerfield Beach, Fl (305) 481-2170

PC Distributor
Orlando, FL (800) 624-3250

The Super Source
Norcross, GA (800) 241-8579

Mid America
Carmel, IN (317) 846-3101

CPU Distribution, Inc.
Burnsville, MN (612) 894-9153

Asibem, Inc.
Bluespring, MO (816) 229-2442

Computer Wholesalers
Lincoln, NE (402) 466-1962

Micro Configuration East, Inc.
Brooklyn, NY (718) 941-2512

Microsel
Oklahoma City, OK (405) 787-4354

Power House Sales
Sioux Falls, SD (605) 335-7181

American Computer Distributing
Chattanooga, TN (615) 870-1073

Columbia Data Systems, Inc.
Columbia, TN (615) 381-4650

Inter-Micro Distributor, Inc.
Alberta, Canada (403) 438-3997

Paris Sud Electronique Composant
Paris, France (1) 69.20.66.99

MAJOR DEALERS:

Computerland of Whittier
Whittier, CA (213) 945-8321

Elek-tek, Inc.
Chicago, IL (312) 677-7660

Inacomp Computer Center
Columbus, OH (614) 431-2230

Inacomp Computer Center
Saginaw, MI (517) 790-1360

Executive Business Machines
Hickory, NC (704) 324-9035

See us at COMDEX/Fall '86 November 10-14, 1986

American
COMPUTER & PERIPHERAL, INC.

Corporate Office: 2720 Croddy Way, Santa Ana, CA 92704 USA • Tel: (714) 545-2004 • Fax: (714) 545-2146
Northeastern Office: 830 Busch Court, Columbus, OH 43229 USA • Tel: (614) 846-5433 • Fax: (614) 846-7656

®IBM AT and MS-DOS are trademarks of International Business Machines and Microsoft Corporations, respectively.

*Effective October 1, 1986. Subject to Intelogic Trace geographic restrictions and regulations.

Inquiry 14 for End-Users. Inquiry 15 for DEALERS ONLY.

problem. The first is the conforming code segment, used when the shared code does not need to access restricted elements of the system, such as hardware I/O ports. Libraries of routines that perform data conversion between ASCII and binary or floating-point, for example, could take advantage of conforming segments.

The other mechanism is a descriptor called the gate, which provides a

(continued)

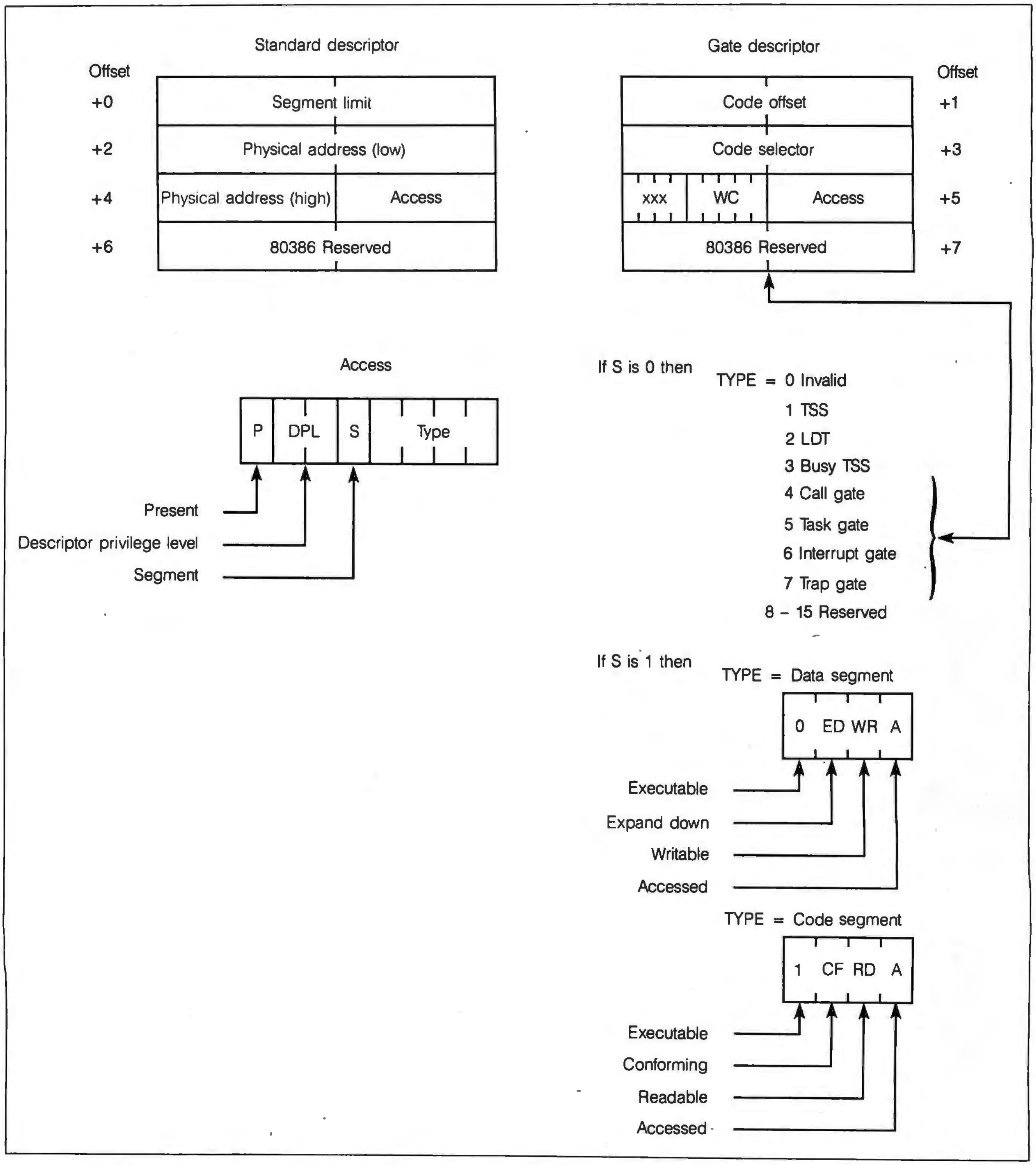


Figure 2: *The format of a descriptor.*

DD: Two *Ctrl* and *Alt* keys for ambidextrous access.
IBM: Same

DD: Positive-tactile, firm-feel keys.
IBM: Same.

DD: 12 *Function* keys for increased automatic operation.
IBM: Same.

DD: Extra-wide, easy-to-find *Return* key.
IBM: Changed sizes again by shrinking their Return key.

DD: Dedicated *Function Control* keys.
IBM: Same.

DD: *Caps, Num* and *Scroll* Lock indicator lights.
IBM: No lights on their XT.

DD: Isolated *Escape* key.
IBM: Same.

DD: Enlarged *Backspace* and *Shift* keys
IBM: Same

DD: Separate dedicated *Numeric Pad* with enlarged *Enter key,* four *Arithmetic Function* keys.
IBM: Same.

DD: Switch-selectable compatibility with IBM, PC, XT, AT, PCjr, AT&T and all compatibles.
IBM: Enhanced keyboard runs only on their new XT's, AT's and ignores their installed base which probably includes you.

DD: Separate *Cursor* and *Screen Control* keys.
IBM: Same.

Here it is... DATADESK'S
TURBO-101 ENHANCED KEYBOARD
for the 10 million PC users IBM just ignored!

IBM just announced their new redesigned "standard" keyboard for personal computers. There's only one problem: *it won't work* on your IBM computer if it was purchased prior to June 1986 or on *any* PC compatible purchased at any time!

Not to worry. Our new *Turbo-101* Enhanced Keyboard gives you the layout and enhancements of the IBM with some logical improvements (see above photo). And it works on your existing PC, XT, AT, PCjr, AT&T, Epson and virtually all compatibles!

Get Borland's Turbo Lightning for Free!

To really turbocharge your productivity, we are including, free-of-charge, Borland's red-hot Turbo Lightning software with each keyboard. Now, when using SideKick, WordPerfect, Microsoft Word, 1-2-3 or most popular programs, our Turbo-101 Keyboard will check your spelling *as you type,* gives you instant access to Random House's 80,000-word Concise Dictionary and 60,000-word Thesaurus and much, much more! *"Lightning's good enough to make programmers and users cheer, executives of other software companies weep,"* says Jim Seymour of PC Week. Sold separately, Turbo Lightning retails for $99.95!

The Turbo-101 is the best data entry tool since the pencil!

For users of spreadsheets like 1-2-3, the Turbo-101's *separate* cursor controls and numeric keypad makes entering numeric data into cells and moving from cell to cell as natural as moving your fingers. And for *word-processing,* the *'Selectric'* typewriter layout makes the Turbo-101 as easy to use as a pencil; and with the extra large Enter, Shift & Control Keys, you'll make so few mistakes, you wont even need an eraser!

Up to now, DataDesk International may be one of the best kept secrets, but here's what's being said about our first *end-user* Keyboard/Borland software bundle:

"DataDesk Intl. has designed a sturdy and handsome keyboard that has tactile response...is the hardware bargain of the year" says Charles Humble, Oregonian Jan 86

"In addition to incorporating every improvement IBM designed into their advanced AT keyboard, it solves all of the problems exhibited by their regular PC/XT keyboard...It's a great bargain!" PC Productivity Digest May 86

"This keyboard is neat to type on and feels solid. It has tactile feedback keys...I can type much faster on it." Test Drive Scorecard: DataDesk-10 Key Tronics-9 Teleconnect Magazine May 86

"The best part of the keyboard is the way it feels. It's ideal! And fast. I've never worked on a keyboard with a nicer touch." Business Computer Digest Aug 86

"Who Can Pass Up a Deal? Department. Talk about an aggressive product!" John C. Dvorak, InfoWorld Mar 86

"It's a good keyboard. Good feel: the keys have tactile feedback. No mush at all. This is about as good a keyboard deal as you're likely to find. . . I have absolutely no hesitation in recommending the model PC8700." Jerry Pournelle Byte Magazine, Sept 86.

SPECIAL OFFER!
ONLY $149.95* *FOR BOTH KEYBOARD & SOFTWARE*

Includes 30-day money back guarantee and 1 year full warranty.

To prove that we don't ignore you or your pocketbook, you get our Turbo-101 Enhanced Keyboard and Borland's Turbo Lightning for an astounding $149.95* No, you didn't read it wrong. During this amazing **Introductory Offer** you get both keyboard and software for less than most software programs by themselves! Now, if you're still feeling ignored, you can always do what you-know-who wants you to do...*and buy a new computer to get their keyboard!*

*Price does not include adaptor cables required by certain compatibles • A limited offer—price subject to change without notification.

credit card orders
call (800) 826-5398
in CA call (800) 592-9602

BOTH TURBO-101 ENHANCED™ KEYBOARD AND BORLAND'S TURBO LIGHTNING™ SOFTWARE FOR ONLY:
$149.95* LIMITED OFFER
*Plus $10 Shipping & Handling. California Residents Add $9.75 Per Unit Sales Tax
NO. UNITS: ______
PAYMENT: ☐VISA ☐MC ☐CHECK
AMOUNT ENCLOSED: $______

Enter
NAME: ______
ADDRESS: ______ STATE: ______ ZIP: ______
CITY: ______ COMPUTER TYPE: ______
PHONE: ______ EXP: ______
VISA☐ MC☐ NO.: ______

The Only Alternative
datadesk™
INTERNATIONAL

7650 Haskell Ave., Van Nuys, CA 91406 (818) 780-1673

Turbo-101 is a trademark of DataDesk International. Turbo Lightning is a trademark of Borland International. IBM and IBM AT are registered trade-marks of International Business Machines, Inc.

passageway to a more privileged execution level. It does this with an additional level of indirection. A gate has a privilege level that is separate from the privilege level of the code segment to which it points (see figure 3). For example, a call gate that has a descriptor privilege level of 3 could point to a code segment with a descriptor privilege level of 0. This means that any executing code segment could issue a call through the gate because 3 is the least privileged level. Since the gate points to a level-0 segment, the 80286 would change the privilege of the currently executing task to level 0 to match the new segment. Thus, the application runs at a higher privilege level, but only while executing the operating system's secure code. It then returns to its original level after the return instruction.

The 80286 architecture also provides interrupt and trap gates for handling hardware and software interrupts.

The notion of a task is built into the 80286 architecture. Loosely defined, a task is a set of segments required to perform a particular series of operations. The 80286 creates this union of segments with the help of two descriptors, the task-state segment and the LDT.

It is often helpful to think of tasks as separate programs executing on their own individual microprocessors. The task-state segment provides a good simulation of this notion, since each task-state segment holds a copy of all the machine registers used by the task. Because in most cases a single processor handles all the tasks in a system, the operating system switches between the tasks, executing one for a period of time, then another. In this way, a multitasking system appears to be running many programs concurrently. When a task-switch operation is performed on the 80286, the current values of all the registers are stored in the task-state segment of the executing segment; the registers are then loaded from the task-state segment of a new task, which begins execution based on the new value of the instruction pointer.

The segment descriptors in the task's LDT are available only to that task; another task cannot access them. This isolation provides an excellent form of protection.

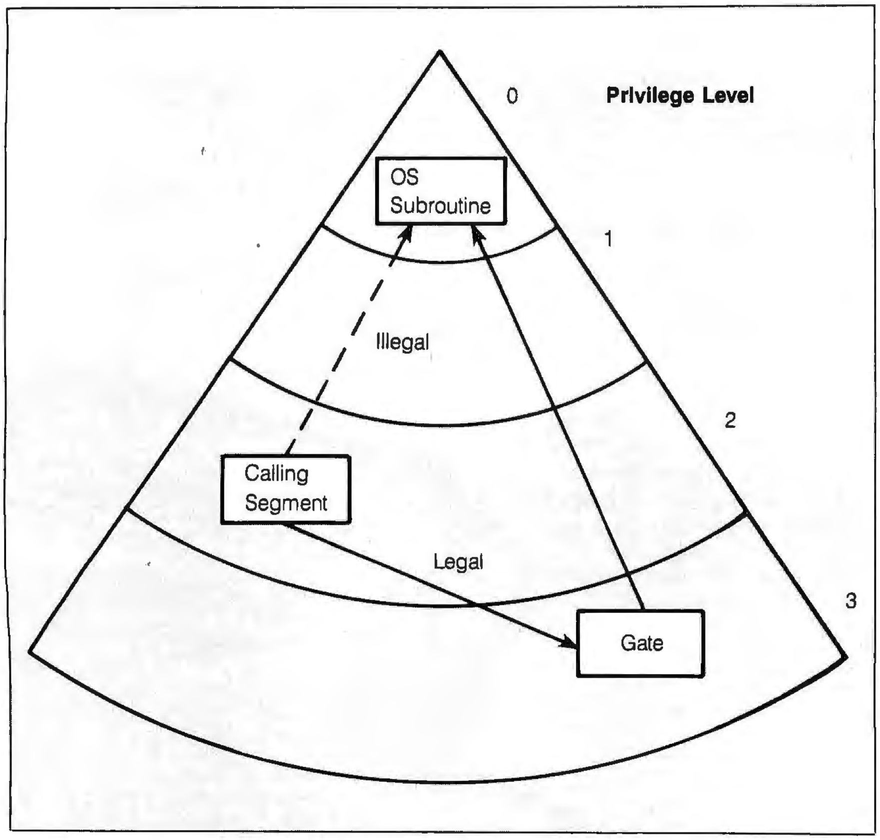


Figure 3: *A call gate that has a descriptor privilege level of 3 but points to a code segment with a descriptor privilege level of 0.*

THE PROTECTED-MODE PROGRAM

My program, PM_AT, places the IBM PC AT into protected mode and provides support scaffolding for further experimentation with protected mode. The program illustrates various features of the 80286's native mode and includes examples of fault handlers, conforming code segments, gates, and task switching.

I wrote the program using the IBM Macro Assembler version 2.0. Since this product was designed for the 8086 family of processors, it does not support some features of protected mode. In some cases, I have resorted to programming tricks to provide a reasonable approximation of the missing features.

The PROTECT.INC listing contains a set of macros that let the user assemble 80286 protected-mode instructions. The 80286's instruction set contains two kinds of enhancements over the 8086 family: the addition of new instructions such as PUSH <data> and ENTER, and the protected-mode instruction set. The IBM Macro Assembler version 2.0 supports the first group of new instructions but not the second. When you include PROTECT.INC, macros simulate the unsupported op codes such as LGDT and ARPL.

TABLES

Three macros help build the descriptor tables. Each segment in the program is defined with either a memory-segment (MSEG) or a system-segment (SSEG) macro. These two macros

(continued)

- GOOD Product Knowledge
- FREE Technical Support
- Warranty & Return Policy

800 382-3888

STOP

YOU'VE FOUND US

NITRO MICRO SM

THE LOGICAL ALTERNATIVE To Your Mail Order NEEDS....

ORDER TOLL FREE 800-382-3888

COMPUTERS

AT&T

6300 128k 1 Drive $1589
6300 256k 2 Drives $1699
6300 640k 20MB $2298
6300 PLUS 512k 20MB $3790
All AT&T Systems Include: CPU, Monochrome Monitor, Keyboard and MS-Dos

NITRO MICRO

PC II 640k 1 Drive 4.77/8MHz $ 695
286 640k 1.2MB Drive 8MHz $1395
Enhanced 286 640k 30MB $1995

Panasonic

Business Partner 256k 1 Drive $ 978
Business Partner 256k 2 Drives $1098
Business Partner 286 512k $1978
Sr. Partner 256k 1 Drive $1098
Sr. Partner 256k 2 Drives $1329
Sr. Partner 256k 10MB $1998
Executive Partner 1 Drive $1798
Executive Partner 2 Drives $1870
Executive Partner 10MB $2695

QUADRAM

Quadview 640k 1 Drive $1396

AMIGA

AMIGA Computer System $1039

SHARP

PC-7000 384k 2 Drive Portable $1139

TELEVIDEO

Telecat 512k 20MB Harddisk $1979
Tele286 Model 1 512k 1.2MB Drive $1949
Tele286 Model 2 512k 41MB Fixed $2829

TOSHIBA

1100 512k 3.5" Drive $1548
1100 Plus 640k Dual 3.5" Drives $1749
3100 640k 3.5" Drive 10MB Fixed $3369

ZENITH

ZFL-171-42 256k 2 Drive Port. $1688
ZF-148-41 256k 1 Drive $ 935
ZF-148-42 256k 2 Drives $1094
ZF-158-41 256k 1 Drive $1289
ZF-158-42 256k 2 Drives $1399
ZF-158-43 256k 20MB Harddisk $1849
ZF-241-81 AT 512k 1.2MB Drive $2396
ZF-241-82 AT 512k 20MB Fixed $3079

MODEMS

AST

Reach 1200 Card w/Crosstalk $ 359

AT&T

4112 1200 Card w/Softcall $ 329
4000 1200 External $ 314

EVEREX

300/1200 Internal $ 139

Hayes

1200B Internal w/Smartcom II $ 339
2400B Internal w/Smartcom II $ 539
Smartmodem 300 External $ 136
Smartmodem 1200 External $ 355
Smartmodem 2400 External $ 589

PROMETHEUS

Promodem 1200B Internal $ 228
Promodem 1200B w/Mite $ 254

QUADRAM

1200 Internal w/Crosstalk $ 289
2400 Internal w/Crosstalk $ 448

VENTEL

1200 Halfcard w/Crosstalk $ 326
2400 Halfcard w/Crosstalk $ 428
PC 1200 External $ 319
PC 2400 External $ 428

WATSON

Watson 1200 Data/Voice $ 344

INTERNAL BOARDS

AST

SIXPACK 64k w/ Sidekick $ 167
SIXPACK 384k w/Sidekick $ 199
SIXPACK PREMIUM 512k $ 358
Advantage 1.5MB $ 678
Advantage 3MB $ 997
5251-11 Package $ 588
5251-12 Package $ 525

DCA

IRMA $ 759

DIGIGRAPHICS

Extend Series 384k $ 149

EPSON

Monochrome Board $ 95
Color Video Board $ 109

EVEREX

Edge $ 269
Graphics Edge $ 298
Mini-Magic Card 384k $ 118

HERCULES

MonoGraphics Card $ 189
Color Card $ 149

INTEL

PC Above Board 2MB $ 638
AT Above Board 2MB $ 649

MOUNTAIN

RACECARD 286 Halfcard $ 469

NITRO MICRO

Monographic Card w/Parallel $ 99
Color Graphics Card w/Parallel $ 98
Parallel Printer Adapter $ 39
Parallel/Serial Adapter $ 59
7-PAK 0k Par/Ser/Clock/Game $ 99

ORCHID

Tiny Turbo 286 $ 439

QUADRAM

EGA+ Video Card $ 384
Quadcolor I $ 208
Quadboard 384K $ 229
QuadSprint $ 436
QuadLink $ 349

SIGMA

Color 400 Video Card $ 462

STB

Chauffeur $ 259

TSENG LABS

EVA EGA Adapter $ 319

UNIVATION

TurboCharger 9.57 MHz 8086 Card $ 379
Dream Board 512K EMS Turbo $ 539

MONITORS

AMDEK

300G Green $ 119
300A Amber $ 128
310A Monochrome Amber $ 155

EPSON

Monochrome Monitor $ 109
RGB Color Display $ 388

NEC

MultiSync HiRes Monitor $ 538

PGS Princeton Graphic Systems

Max 12E Monochrome $ 159
HX-12 Color Display $ 419
HX-12E HiRes Color $ 525
SR-12 HiRes Color $ 539

TAXAN

121 Green monochrome $ 139
122 Amber monochrome $ 148
630 RGB Color $ 428
640 RGB Color $ 488

ZENITH

1230 Green composite $ 99
1220 Amber composite $ 99

DRIVES

I-MEGA

Half Height 10MB $1319
Half Height 10-10 $1966
Half Height 20MB $1752
Half Height 20-20 $2498

IRWIN MAGNETICS

10MB Tape Backup Internal $ 462
20MB Backup Internal AT $ 566
10MB Tape Backup External $ 655

MAYNARD

10MB ONBOARD Hardcard $ 588
20MB ONBOARD Hardcard $ 689

MOUNTAIN

20MB Drivecard $ 769

PLUS DEVELOPMENT

10MB Hardcard $ 499
20MB Hardcard $ 639

SEAGATE

10MB Internal w/cont $ 409
20MB Internal w/cont $ 439
30MB Internal AT $ 668
20MB External w/cont $ 538

PRINTERS

CITIZEN

MSP-10 160 cps $ 289
MSP-15 160 cps 132 col $ 389
MSP-20 200 cps $ 329
MSP-25 200 cps 132 col $ 488
Premiere 35 Daisywheel 35 cps $ 469

DATA PRODUCTS

8012 180 cps $ 288
8022 180 cps wide $ 454
8052 200 cps color $1079
8072 200 cps color $1432

EPSON

LX-86 120 cps $ 218
FX-85 160 cps $ 349
FX-286 200 cps 132 col $ 549
LQ-800 NLQ 200 cps $ 528
LQ-1000 NLQ 200 cps 132 col $ 688
LQ-2500 324 cps $1069
DX-10 LQ 10 cps $ 228
DX-20 LQ 20 cps $ 312
DX-35 LQ 35 cps $ 559
EX-800 30 cps $ 529

HEWLETT PACKARD

LaserJet $2298
LaserJet Plus $2919
ThinkJet $ 398
7475A Plotter 6 Pen $1348

OKIDATA

182 IBM Parallel $ 219
183 Wide IBM Parallel $ 369
192 IBM Parallel $ 349
193 Wide IBM Parallel $ 489

OTC

700 700 cps $1419

Panasonic

1091 120 cps $ 248
1092 180 cps $ 329
1592 180 cps $ 439
1595 240 cps $ 588

QMS

Kiss Laser Printer $1698
BIG KISS Laser printer $2389
Post Script 800 Laser Printer $4688

QUADRAM

Buffer 8K Par-Par $ 138
Buffer 64K Par-Par $ 165
Buffer 128K Par-Par $ 198

TOSHIBA

P321 Parallel $ 479
P341 Parallel $ 788
P351 Parallel & Serial $1032

Add 2% shipping and handling on all orders.
NO SALES TAX
No surcharge on Visa or MasterCard. All items subject to availability and price change. All hardware and software will be repaired or replaced at our discretion within warranty period of manufacturer.

VISA MasterCard

Hours: Eastern
Mon.-Fri. 10 a.m.-7 p.m.
Saturday 11 a.m.-5 p.m.

QUICK SHIP FAST DELIVERY SERVICE

Inquiry 138

ORDER TOLL FREE 800-382-3888

P.O. Box 90200
Indianapolis, IN
46290-0200

NITRO MICRO SM

create assembler variables that contain information necessary for building a descriptor. A third macro, the descriptor (DSCRP) macro, builds a descriptor from the variables when given the name of a segment. The DSCRP macro also accepts an optional export name that is used as a selector for the descriptor. This trick works because the format of a level-0 GDT selector is equivalent to the index multiplied by eight. Since each descriptor takes up 8 bytes, the byte offset of the descriptor equals the index times eight.

The GDT begins with the line

```
DESCRIP <0, 0, 0, 0>
```

which builds the first GDT entry via the DESCRIP data structure. The system can never access descriptor 0 in the GDT; therefore, it is null. Descriptors 1 through 7 are set up to meet the requirements of the INT 15h function call. Descriptor 8 provides a task-state segment for the initialization code. The INT 15h protocol does not require the creation of a task, but a task fault will occur at the first task switch if the task register has not been loaded.

The next four descriptors are for the portion of code that I call the MiniBIOS. The MiniBIOS handles the hardware interrupts and provides screen-handling routines similar to those normally found in the standard ROM BIOS; however, the ROM functions will not execute correctly when the processor is in protected mode. When you invoke the MiniBIOS functions (via a trap gate at INT 30h), the privilege of the executing task is set to 0 for the duration of the function.

The next nine descriptors point to the fault handlers I have implemented. As you add new code to the system, it is inevitable that at some point a routine will fail. The handlers that are a part of PM__AT will indicate the type of fault and the location in the program where the fault occurred. The descriptors provide code, data, and task segments for the fault-handling code. The last of these descriptors is initially unused but available for the fault-handling routines to modify as needed.

The next two descriptors provide access to a library of ASCII/binary-conversion routines. The first descriptor, shlib__code, is a level-0 conforming code segment. Programs with a lower privilege can gain access to this code via the CALL__ GATE descriptor, which has a privilege level of 3. Since the code segment is conforming, the privilege of the calling task does not change when executing the code in shlib__ code.

The next descriptors provide the system-level descriptors for a second task to illustrate the 80286's task-switching capabilities.

(continued)

The final four descriptors are null and are available for future expansion. Normally, an operating system will provide functions such as CREATE_TASK, ALLOC_SEGMENT, and FREE_SEGMENT, which create new descriptors or free up entries in the GDT for later use. The operating system would have access to the GDT as if it were a standard data segment (via descriptor 1). Obviously, this sort of access to the GDT must be limited to only the most secure portions of the operating system.

The GDT requires the physical starting address of the segment as a part of segment descriptors. The assembler places the 16-bit (real address mode) segment address in the prototype GDT. A subroutine in the program's initialization portion adjusts all the addresses, multiplying them by 16. It assumes all data resides in the first 1 megabyte of memory.

The IDT contains a set of gates that point to the routines that handle interrupts and faults. It contains only trap, interrupt, or task gates. Trap gates handle software-interrupt requests, and interrupt gates handle hardware interrupts. You can invoke a task gate by hardware or software; it causes an immediate task switch.

As defined under DOS, the interrupt structure of the PC AT resembles that of the standard PC or XT. Unfortunately, IBM chose to use some of the interrupts that Intel designated as reserved. In the AT, this causes a conflict between the way DOS wants to use the interrupts and the way the processor wants to use them. In my program, I followed the Intel specification and allocated the first 32 interrupts for exceptions and fault handling, mapping the hardware interrupts to vectors 32 (20 hexadecimal) and above (see table 1).

Descriptors 0 through 31 handle processor faults. All these gates are interrupt gates except for 8, 10, and 12, which are task gates. Task gates are required because the failures that cause these interrupts are so serious that the processor must load an entirely new machine state to continue running.

Table 1: *The interrupt vector mapping used in* PM_AT *contrasted with* DOS.

Vector	Function	DOS Equivalent
00	Divide fault	0
01	Trap (single step)	1
02	NMI	2
03	BRKPT (INT 3)	3
04	Interrupt on overflow	4
05	Bound interrupt	
07	80287 not available	
08	Double fault	
09	80287 segment overrun	
10	Invalid TSS fault	
11	Not present fault	
12	Stack fault	
13	General protection fault	
16	80287 error	
17 . . . 31	Reserved	
32	Timer	08
33	Keyboard interrupt	09
35	Com port 1	11
36	Com port 2	12
37	Printer 2	13
38	Floppy disk	14
39	Printer 1	15
40	Real-time clock	
45	80287 request	
46	Hard disk	

The hardware interrupts are found at vectors 20 to 2F hexadecimal. The change from the DOS vector locations is accomplished by programming the 8259A interrupt controller. In the PC AT, the BIOS call INT 15h takes care of this detail.

Interrupt vectors 30 and 31 hexadecimal have been allocated to the MiniBIOS. Since INT 30h provides display services, its privilege level is set to 3 so that any task may request its services.

The last page of the program contains an LDT for the second task. The code and data segments for the second task are in the LDT rather than the GDT. When an LDT is used for each task, the GDT will contain only a task-state segment and an LDT descriptor for the task. The descriptors for the code and data segments need not appear in the GDT.

INITIALIZATION

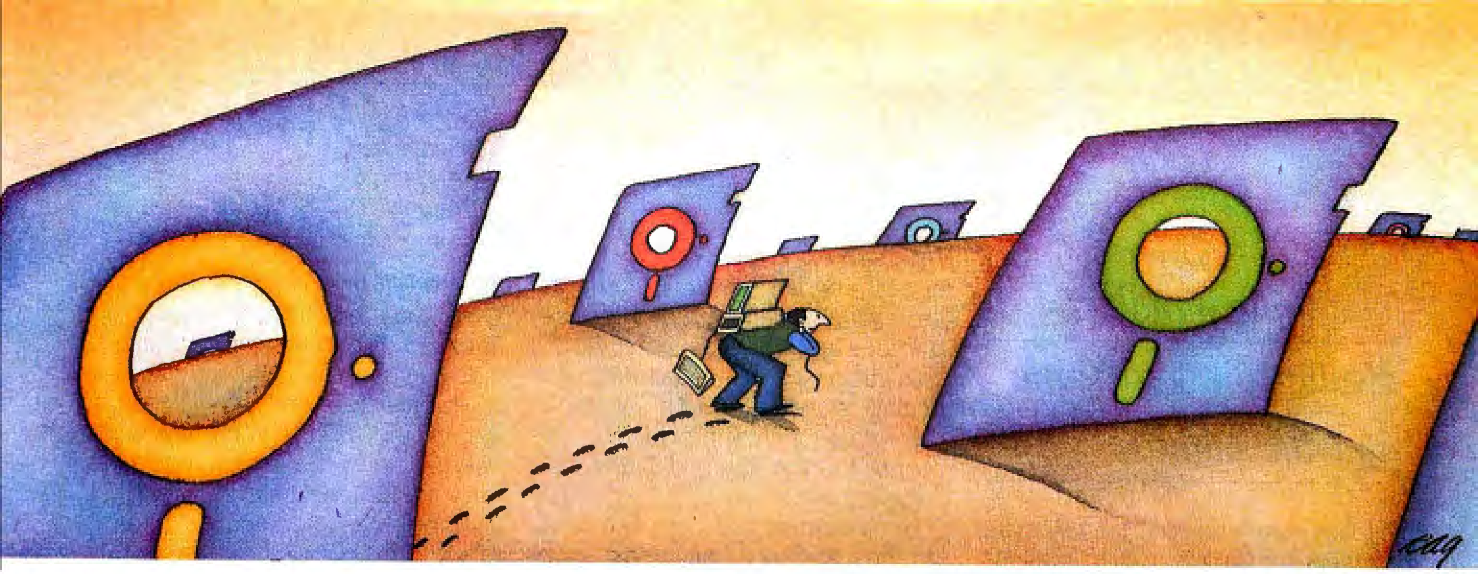
After MS-DOS loads PM_AT, execution begins at the label START. The program then calls the subroutine ADJUST_ADDR to fix up the physical addresses in the GDT and LDT, converting them from 16-bit real address mode segments to 24-bit physical addresses. It then calls the ROM BIOS to request protected mode.

The PC AT BIOS provides a function (code 89 hexadecimal of INT 15h) that places the processor into protected mode. To do this, it loads the GDT and IDT base registers, programs the 8259A interrupt controller with the hardware vector locations, and frees up the A20 address bit. (Due to a design oversight at Intel, the 80286 occasionally generates addresses beyond 1 megabyte. The PC AT includes hardware to force the A20 bit to 0 while running in emulation mode.) It then sets the protected-mode bit in the machine status word (MSW) register of the processor and returns to the calling program.

PROTECTED-MODE EXECUTION

The only setup required after the program returns from the INT 15h call is

(continued)



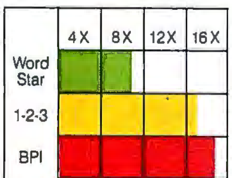
PROBLEM: There's just no easy way to move from one software program to another.

THE SOFTLOGIC SOLUTION: Software Carousel

Now you can keep up to 10 programs loaded and ready to run.

Hard to believe, but some people are happy with just one kind of PC software. Well, this is not a product for them.

But if you're someone who depends on many packages, all the time—someone who'd use several programs at once if you could, well now you can. With Software Carousel.



With Software Carousel running in RAM, you can load a program and retrieve a file up to 15 times faster. Test conducted on an IBM XT

Why call it "Software Carousel"?

In some ways, Software Carousel works like the slide projector you're used to. You load a handful of pictures, view one at a time, then quickly switch to another. A simple idea, with powerful possibilities for computing.

Here's how it works. When you start Software Carousel, just tell it how much memory you have, load your software and go to work.

Need to crunch numbers? Switch to your spreadsheet. Need your word processor? Don't bother saving your spreadsheet file. Just whip over to your document and do your work. Snap back to your spreadsheet, and it's just like you left it.

With up to ten different programs at your fingertips, you'll have instant access to your database, communications, spelling checker, spreadsheet, word processor, RAM resident utilities, languages, anything you like.

Reach deep into expanded memory.

This could be the best reason ever for owning an expanded memory card, like the Intel Above Board, AST RAMpage, or any card compatible with the L/I/M Extended Memory Standard.

Software Carousel puts programs into this "high-end" memory for temporary storage when they're not in use. And switches them back out when you want them. It's fast, efficient, and easy.

If you want, Software Carousel will even use your hard drive for swapping. Just allocate a portion for storage, and go to work.

Sidekick, Superkey and Ready. All at the same time.

You know what happens if you try loading two or more RAM resident utilities at once. You get crashed keyboards, frozen screens, all kinds of interference between programs fighting for control.

With Software Carousel, you can have as many accessories and utilities on-tap as you want. Just load different ones in different Carousel partitions. Since they can't see each other, they can't fight.

The easy way to maximize PC power.

With all this power, you might think Software Carousel is complicated and difficult to use. Not so. Set it up once, and it will remember forever. Better still, Carousel will look for the programs you use most often, and optimize them for the quickest access.

You can spend a lot more money, and still not get the convenience and productivity increase of Software Carousel.

The way we see it, there are certain things you have the right to expect from your computer. Access to your software is one of them. And at our special introductory price of just $49.95*, Software Carousel is the best way to get it.

But hurry. This price won't last long.

Order today at 800-272-9900 (603-627-9900 in NH) or send the coupon below.

Special combination pricing is available for the purchase of Software Carousel and other SoftLogic products, including Cubit, DoubleDOS and Disk Optimizer.

Software Carousel $49.95*

YES! Send me_____copies of Software Carousel at the special introductory price of just $49.95*

Name________________________

Company________________________

Address________________________

City____________ State/Zip____________

Check Enclosed ☐ VISA ☐ MC ☐ AMEX ☐

Card # Exp. Date

Signature________________________

SoftLogic Solutions, Inc.
530 Chestnut Street
Manchester, NH 03101
800-272-9900
(603-627-9900 in NH)

SOFTLOGIC SOLUTIONS

Call today: 800-272-9900

*plus $5.00 shipping and handling.

The fault handlers display the fault type and the address.

to load the task register. The task-state segment for the initial task contains no data but must be present for a task switch to occur.

Program execution is very simple. The program beeps once to let you know all is well, clears the interrupt-register masks, enables interrupts, and goes into a loop. The main task prints the number of clock ticks that have occurred since interrupts were enabled. It then invokes a second task, which merely displays a message and returns. The program loops until you generate a keyboard interrupt; it will then halt and reset the system.

The second task runs at privilege level 3, the way most applications would in a standard operating system. Its execution privilege level is determined by the privilege of its code-segment descriptor found in the task's LDT. When it invokes the display services via INT 30h, however, its execution privilege is changed to level 0 for the duration of the call. To prevent an errant program from corrupting the local variables and return addresses of the level-0 code, the 80286 changes to a level-0 stack before it begins level-0 execution. A level-0 memory segment is provided for this purpose in the task's LDT and task-state segment.

The executing tasks use two small libraries of code. One is the set of routines that does ASCII/binary conversion. They are located in a conforming code segment, and any task can call them via the CALL_EX macro. This macro takes an export name and a privilege level and creates a FAR call to the object. It can be used with any export segment or gate. The other set of routines provides display services and is part of the Mini-BIOS. Since these routines require access to hardware, they will execute at privilege level 0 and are invoked via the gate at interrupt 30h.

The ASCII/binary-conversion library is located in the shlib_code segment. This library provides the following 16-bit conversion functions: ASCII hexadecimal to binary, binary to ASCII hexadecimal, ASCII signed integer to binary, binary to ASCII signed integer, ASCII unsigned integer to binary, and binary to ASCII unsigned integer. The actual calling sequences are documented in the code.

PM_AT provides a set of rudimentary display utilities for character and line output and cursor manipulation. These are invoked via the trap gate at INT 30h. The current implementation of the program supports only the IBM monochrome display. To support the color adapter, you replace the DEV_MONO literal with DEV_COLOR. To prevent snow on the display, a color implementation should also loop and wait for a retrace before writing any characters to the display.

Fault Handlers

If you use this program as a basis for your own experiments with protected mode, the fault handlers are important. Whenever the 80286 detects a violation of its protection rules, it issues an interrupt that will vector through the IDT to the appropriate fault-handling routine. My fault handlers display the fault type and the address of the instruction that caused the fault. They will also display the fault code for the "general protection" fault and other faults that provide an error code. The format of the address is <selector>:<offset>. You will have to extract the selector's index

(continued)

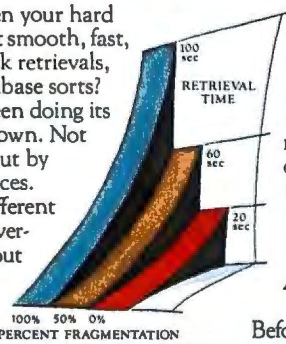
PROBLEM: The more experience your hard disk has, the harder it has to work.

THE SOFTLOGIC SOLUTION: Disk Optimizer™

Your hard disk will run faster when it's not chasing around after files.

Remember the old days when your hard drive was new? Remember that smooth, fast, slick performance? Those quick retrievals, rapid saves, lightning-like database sorts?

Well ever since, DOS has been doing its best to slow your hard drive down. Not by slowing down the motor, but by breaking your files up into pieces. Storing different chunks in different places. Data files, programs, overlays and batches that started out in one seamless piece are now scattered all over.



The more fragmented your files get, the longer they take to retrieve.

Loading is slower.
Sorting is slower.
Retrieving, backing-up.
Everything takes longer because your disk has to work harder.

Problem is, it's something that happens so gradually you may not notice the difference. At least, not until you see the dramatic improvement after using Disk Optimizer.

File fragmentation—It's a problem you can see.

Watch your hard drive the next time it reads or writes a file. Each "blip" of the LED means the drive-head is moving to another place on the disk—either to pick up or lay down another chunk of data.

And the truth is, head movement takes time. Far more time than actual reading and writing. What's worse, all this head movement causes extra wear and tear that can shorten the life of your drive.

Disk Optimizer—Tunes up your disk by cleaning up your files.

Disk Optimizer works by finding all the scattered pieces of your files and putting them back together where they belong. Next time your drive reads it, there's just one place to look.

And the results are often dramatic. Reading and writing times may be cut by as much as two thirds. Database sorts that used to take hundreds of head moves now proceed quickly and efficiently. And since head movement is now at an absolute minimum, your disk drive will lead a longer, more productive life.

Analyze, scrutinize, optimize.

Before you optimize, you'll probably want to analyze. So Disk Optimizer shows you, in percentages, how much fragmentation has taken place—on the entire disk, in individual directories, or for groups of files you specify using global or wildcard names.

Plus, there's built-in data security that lets you assign passwords to as many files or file groups as you want.

And the File Peeker gives you an inside look at the structure of files. It's a great way for non-programmers to learn more about computers, and a powerful tool for professionals who want to analyze the contents of their disks.

Get your hard drive back in shape—at a special low price.

When you think about it, it's simple. The longer you own your hard drive, the more you come to depend on it. But the longer you wait to get Disk Optimizer, the less performance you'll have.

And the less chance you'll have to buy Disk Optimizer at the special introductory price of just $49.95*

That's a small price to pay to get back the speed you depend on. But it's a price that won't last long.

Ask for Disk Optimizer at your computer dealer.

Or order today by calling SoftLogic Solutions at 800-272-9900 (603-627-9900 in NH), or send the coupon below.

Special combination pricing is available when you buy Disk Optimizer along with other SoftLogic products like DoubleDOS, Software Carousel and Cubit.

Ask for details.

Disk Optimizer 49^{95}*

YES! Please send me______copies of Disk Optimizer at this special introductory price.

Name____________________

Company____________________

Address____________________

City__________State/Zip__________

Check Enclosed ☐ VISA ☐ MC ☐ AMEX ☐

Card #__________Exp. Date__________

Signature____________________

SoftLogic Solutions, Inc.
530 Chestnut Street
Manchester, NH 03101
800-272-9900
(603-627-9900 in NH)

SOFTLOGIC SOLUTIONS

Call today:800-272-9900

*plus $5.00 shipping and handling.

portion to determine which code segment was executing when the fault occurred.

In a complete operating system, the different faults would be handled differently. For example, the "not present" fault (INT 11h) would probably cause the operating system to load the referenced segment from virtual memory and restart the instruction that caused the fault; the "general protection" fault (INT 13h) would normally cause termination of the offending task and the logging of a message to the appropriate terminal.

My program assumes that any fault is a programmer error. It will display the location and type of fault on the display, pause, and reboot the system. Should you attempt to make the fault handlers more sophisticated, you might run across two other conditions: double fault and shutdown. Double fault (INT 8) occurs when the system is attempting to process a fault but is prevented from doing so by another protection violation. The double-fault handler, therefore, should be an isolated task that makes as few assumptions as possible. Should the processor detect another violation while attempting to handle the double-fault condition, it will give up entirely and shut down. The PC AT includes hardware that detects the shutdown status on the bus and resets the processor.

SUGGESTIONS FOR EXPANSION

To understand all the complex interactions of the protected mode, you will almost certainly need Intel's *iAPX 286 Programmer's Reference Manual.* This manual includes a pseudocode description of the processor microcode for instructions such as INT or CALL FAR. Unfortunately, in the latest edition (1985), some of the steps are missing. However, after you've worked with the processor for a while, you should get a feel for the types of operations being carried out.

The next logical step in developing the code is adding a keyboard driver. After you get this working, you will have a small interactive system to build on and could add a debugger/monitor. This would let you recover more gracefully from certain errors rather than reset the system.

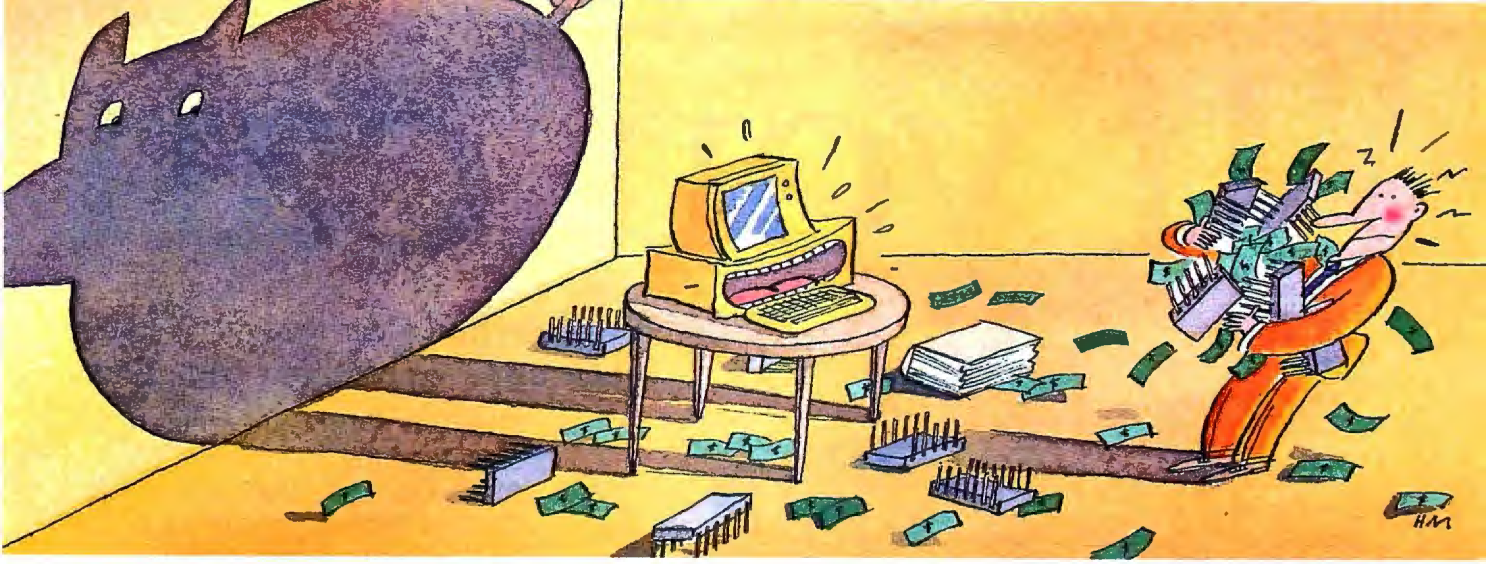
The complexity of the native-mode 80286 is an order of magnitude greater than that of the previous generation of microprocessors. But as powerful support for secure multitasking operating systems, it will be much appreciated by designers of the next generation of operating systems. ■

BIBLIOGRAPHY

iAPX 286 Operating Systems Writer's Guide. Intel Corporation, 1983.

iAPX 286 Programmer's Reference Manual. Intel Corporation, 1985.

IBM PC AT Technical Reference. IBM Corporation, 1984.



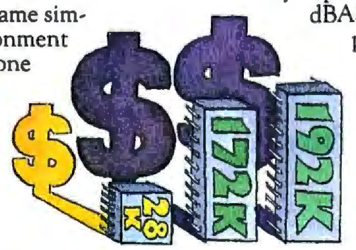
PROBLEM: Most concurrent processing systems take everything you've got, and give back very little.

THE SOFTLOGIC SOLUTION: DoubleDOS™

True multitasking that's quick, painless and powerful.

A lot of people had the same simple idea – make a PC environment where your computer does one thing in the background while you work on something else in the foreground. Problem is, the way some people designed their systems, they're just not simple at all.

Look at products like TopView™ or Windows.™ Not only do they cost a fortune, they take gobs of RAM and half a lifetime to learn. Worse yet, they don't even run much software. And what they do run is slow.



$49.95	$99.00	$175.00
DoubleDOS	Windows	TopView

DoubleDOS costs less and consumes less RAM than other well-known systems.

So why not get DoubleDOS instead? It's the easy, economical, memory-efficient system that runs all your PC programs, and runs them two at a time.

With DoubleDOS, you can sort a database at the same time you write a memo. Print a spreadsheet while you update accounting. In fact, do almost any two things you want with the software you already have. Best of all, DoubleDOS is fast and simple to use. So you'll be running it in a matter of minutes.

Here's all you need to know to run DoubleDOS.

Unless you know all about .PIF files, interrupt vector swaps and the like, you won't exactly breeze through the set-up procedures for TopView and Windows. Not like DoubleDOS. Just tell it how to divide your memory, and that's it. You're ready to load and run your software right from that familiar DOS prompt.

Say you've started a sort routine in dBASE. Instead of waiting for it to finish, just press "Alt-Esc." Your dBASE screen instantly disappears, and up comes another DOS prompt. Simply load another program and go to work. Meanwhile, dBASE crunches away off-screen.

Press "Alt-Esc" again and dBASE pops back on while your other program continues in the background. Switch back and forth all you want.

Your programs load and run exactly as they would ordinarily. No lost features. No cramped, clumsy windows. No complex menus and new commands. Just "Alt-Esc."

Multitasking in the fast lane with DoubleDOS.

Aside from what it does, the best part of DoubleDOS is what it doesn't do. And that's slow your computer to a crawl.

In contrast to systems like TopView that permanently divide CPU time between the foreground and background, DoubleDOS constantly evaluates the processing requirements of each program it's running. With DoubleDOS, the program that needs more power, gets it. So all your work gets done faster.

Can over 50,000 DoubleDOS owners be wrong?

If you're too busy trying to get work done to suffer through the debugs, updates and fuzzy standards of other concurrent software systems, try DoubleDOS. It comes to you thoroughly field-tested and work-proven by thousands of satisfied users.

So why pay more? Especially when you can have the full power of proven multitasking software for just $49.95.* Look for DoubleDOS at better computer dealers everywhere, or order direct by calling SoftLogic Solutions today at 800-272-9900 (603-627-9900 in New Hampshire). Or send the coupon below.

You'll save even more when you buy DoubleDOS along with other SoftLogic products, including Disk Optimizer, Software Carousel and Cubit. Ask for details.

DoubleDOS $49.95*

YES! Please send me ______ copies of DoubleDOS at just $49.95* each.

Name ______

Company ______

Address ______

City ______ State/Zip ______

Check Enclosed ☐ VISA ☐ MC ☐ AMEX ☐

Card # ______ Exp. Date ______

Signature ______

SoftLogic Solutions, Inc.
530 Chestnut Street
Manchester, NH 03101
800-272-9900
(603-627-9900 in NH)

SOFTLOGIC SOLUTIONS

Call today: 800-272-9900

*Plus $5.00 shipping and handling.
TM signifies manufacturer's trademark.

"If you can

EVEREX Streaming Tape Family

If you can find a company with better tape achievements, buy from them:

- Over 30,000 streaming tape sub-systems, ½ height, 5¼" form factor industry standard QIC 24 recording format, QIC-36 hardware interface and largest installed customer base worldwide.
- Over 100,000 industry standard streaming tape controllers in the field.
- The first company to release QIC-02 IBM-compatible software drivers to the OEM community.
- The first company to release an internal 5¼" half-height IBM compatible tape subsystem.
- The first company to release file-by-file retrieval from a mirror-image backup.

If you can find better streaming tape software, buy it:

User Interface

- Menu driven.
- Multiple windows (nestable and movable).
- Context sensitive on-line help.
- File selection from directory listing or command line.
- Constant tape status display.
- Menu driven hardware configuration guide.

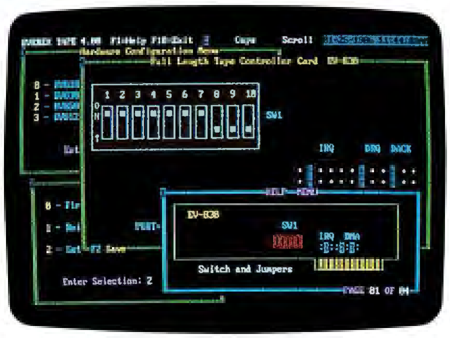
Tape Functions

- Multiple tapes allowing unlimited backup capacity.
- Multiple datasets for multiple backups on the same tape.
- Tape sequence management recorded on the tape.
- Programmable messages.
- File-by-file and image backup on the same tape.
- Installation program.

Functionality

- Physical image backup/restore of local disks.
- Logical image backup/restore of local or remote disks.
- File backup from any local or remote disk of any file size.
- File restore from image backup or file backup.
- Allows wildcards for selection of files.
- Automatic scheduler for unattended backups.
- Backup across networks (using logical image or file-by-file).
- Real time clock display.
- Backup any disk whether formatted for DOS or not
- Escape to DOS while in backup/restore operation.

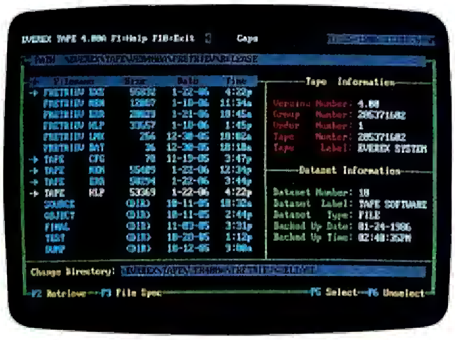
Automatic Scheduler



Nestable, Movable Windows

find...

Backup with Multiple Datasets



Multiple Tape Restores

If you can find a streaming tape with better performance and system characteristics, buy it:

Performance

- File-by-file operations near streaming speed.
- Dynamic adjustment to disk and system speeds.

System Characteristics

- Hardware identification provides software integrity.
- External systems can be shared with many users.
- Same software for all types of tape drive, controller, and computer system configurations.
- External systems uses a shielded round cable with a D-connector for tangle-free, easy connect and disconnect.
- Hardware read-after-write data verification.
- Track 0 seek to insure tape interchangeability.
- Original controller manufacturer.
- Industry standard QIC-24 tape format.
- Industry standard QIC-36 interface.
- Industry standard QIC-02 commands.
- Tape does not require pre-formatting.
- Available in 20MB, 60MB, and 100MB drive capacities.

...buy it!"

EVEREX Excel Streaming Tape.
For the name of your nearest EVEREX dealer or more information, call us. Remember... We are EVER for EXcellence.

Excel Stream 20 Internal

Excel Stream 60 Internal

Excel Stream 60 External ½ Height

Excel Stream 20 External

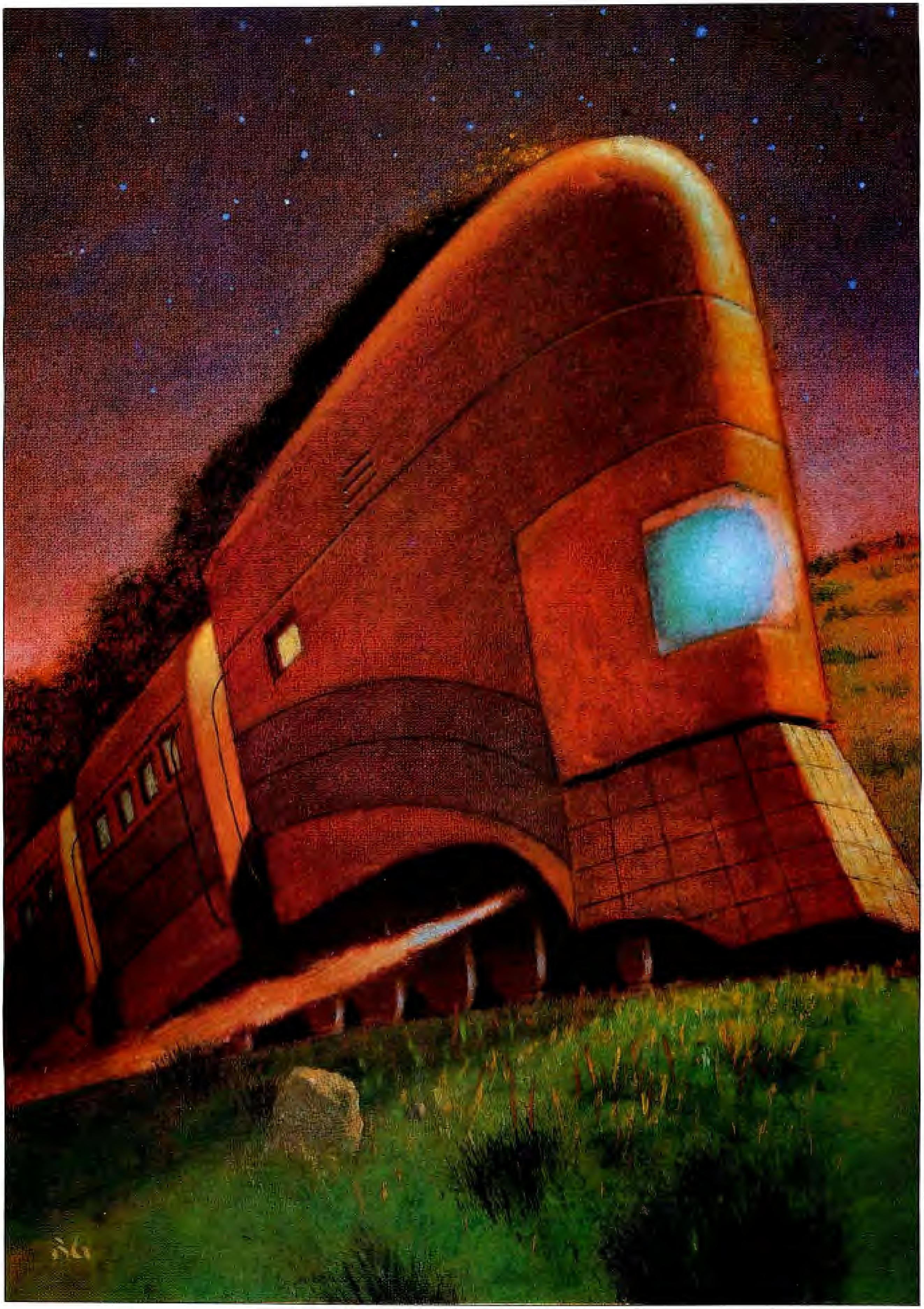
Excel Stream 60 External

Excel Stream 60-8 External

EVEREX
EVER for EXcellence ™

1-800-821-0806 1-800-821-0807 (in CA)
48431 Milmont Dr., Fremont, CA 94538

Inquiry 92 for End-Users. Inquiry 93 for DEALERS ONLY.



ILLUSTRATED BY BART GOLDMAN

IBM PC ACCELERATORS

The theory of operation of three kinds of speedup boards for PCs

BY STEPHEN S. FRIED

EXPANSION BOARDS are available that can speed up your IBM PC or XT. These accelerator boards cost from $149 to $2000 and can result in increased system throughput of 50 to 350 percent (i.e., system speedups of 1.5 to 4.5 times). Some accelerators claim to be better than this, but, for average programs, a speedup factor of 4.5 is the best that can be expected with the average IBM PC. Since a number of accelerators convert your machine into a device that performs as well as or better than the AT, I have included the AT in this article for comparison purposes. Also, the AT has the advantage of running at a number of different speeds. However, in many ways, accelerators are clearly superior to the AT. For example, many of them will be more compatible with your existing hardware and software than the AT, many have faster numerics than the AT, and most eliminate the need to upgrade to 16-bit expansion boards.

Unfortunately, deciding which accelerator to buy can be difficult. For the most part, you must make the decision on your own or with the advice of a friend who has already made such a purchase. Accelerator technology is over the heads of most retailers and does not help them sell new machines. Speeding up an existing machine can involve the use of motherboard boosters, coprocessor accelerators, or emulator accelerators with or without cached memory.

This article will clarify how accelerators work, help you understand how to compare accelerator speed and compatibility, and give you an idea about what type of accelerator does or does not suit your needs. I'll start off with an intuitive discussion of what an IBM PC–compatible accelerator is, followed by a description of the system constraints within which any accelerator must work and of the three classes of accelerators in today's market.

WHAT IS AN ACCELERATOR?

An IBM PC–compatible accelerator is a hardware product that runs at least MS-DOS applications faster than your original IBM PC. Some, but not all, accelerators will run other operating systems and applications that use the hardware directly. I have already suggested that at least two classes of accelerators exist: those that run MS-DOS only and those that run all PC applications. I will show that there are actually three classes and that the division is dictated by the hardware interface between the CPU on the accelerator and the PC's motherboard.

To perform in the total system environment, the accelerator must provide some mechanism for the CPU on the accelerator to communicate and manage its own devices and those on the system board. These devices include memory on the accelerator, motherboard, or in the I/O channel; the DMA controller on the motherboard; and cards in the I/O channel, such as hard disk controllers and graphics adapters.

The typical accelerator contains a clear dichotomy: Most accelerators speed up because they have internal 16-bit data buses instead of the 8-bit bus that is standard on the PC. This means that the accelerator must be able to operate in two modes: 16-bit and 8-bit. For the devices on the motherboard or in the I/O channel to function properly, the accelerator's CPU must access them in a manner that makes the motherboard think they are being controlled by an 8088 running at 4.77 MHz. Three approaches will placate the motherboard: leaving an 8088 behind to manage the motherboard, emulating the 8088 with a faster CPU on the accelerator, or ignoring the rules and speeding up the motherboard in the hope that your motherboard is much better than the one IBM designed in 1981. I will refer to these three approaches as coprocessing, emulating, and motherboard boosting.

COMPATIBILITY

Compatibility is a more important issue than speed, and I will treat it first. The accelerator that runs 99 percent of your software three times faster will always be a better buy than the accelerator that runs 85 percent of your software four times faster. This is because the cost of a system's software is often much greater than the cost of the hardware and often cannot be replaced at a reasonable price. Having made a large investment in software that works, people are normally reluctant to sacrifice an old machine for a newer one unless the newer one will run the installed software base.

An accelerator can attempt to achieve four levels of compatibility: source code level, operating system level, object code level, and PC hardware level. The first two levels, source

(continued)

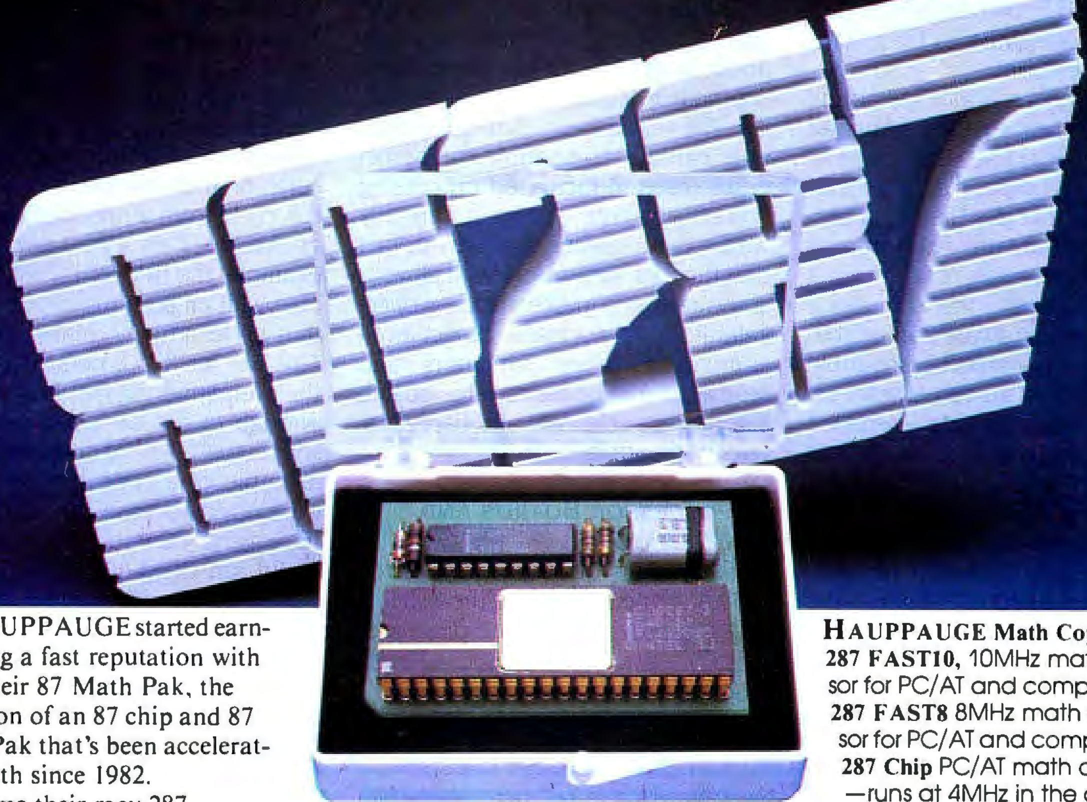
Stephen S. Fried is principal scientist and vice president of R&D at MicroWay Inc. (POB 79, Kingston, MA 02364), which produces 87BASIC and 87BASIC Inline.

and operating system, normally go together and do not require a particular hardware configuration to achieve. For example, it is easy to port UNIX applications written in C among different processors that run UNIX. As a result, many of the minicomputer vendors who have mature products that run under UNIX have chosen coprocessor boards that run UNIX (better than the 80286) for their ports to the PC market. This strategy is probably based as much on greed as speed, since the code that runs on these boards is worthless without the hardware and the vendors don't sell you their personalized accelerator without the software. As another example, in the early days of the PC, I easily ported applications written in Microsoft BASIC that ran on my TRS-80 over to my PC. While this definitely increased the speed of my applications, I could not move all of them because many of the best were written in assembly language.

To run programs written in assembly language, an accelerator has to use a CPU that is object-level-compatible with the original used in the PC. It turns out that it is possible to translate 8080 (Z80 is a superset of the 8080) assembly code to run on the 8088, but it is not possible to run 8080 object code on an 8088 (the machine-level encoding of the instructions is different). The object-level differences between the 8088 and the 8080, 6502, and 68000 have spawned a market for coprocessor boards that run your old programs on your new machine. For an accelerator to run 8088 code, it must contain an object-level-compatible (or machine-level-compatible) processor. Examples of processors that are object-level upward-compatible with the 8088 include the Intel 80186, 8086, 80286, and 80386, and the NEC V20 and V30.

The NEC V20 and V30 can switch back into modes that can also run Z80 object code. Had they appeared in 1981, they could have changed the course of microcomputer history and extended the economic life cycle of CP/M-80. The 80386 does some similar tricks within the Intel family. Like the 80286, the 80386 boots up in an 8086 object-compatible mode called real address mode. Unlike the 80286, it can switch over to a virtual 8086 mode in which it can run one or more 8088 or 8086 programs that have their own private virtual address space. However, in its pure 32-bit mode, the 80386 is not object-compatible with the 8088. This feature indicates recognition of the importance of object-level upward compatibility from the microprocessor vendors themselves. As VLSI designs get more dense, it becomes possible to guarantee the future suc-

(continued)



cess of new products by incorporating older designs as subsets. Thus, an activity that used to be relegated to the OEM is now relegated to the microprocessor merchant.

Object-level compatibility is not all that is needed to guarantee that all programs will work. Accelerators exist that are object-compatible but will not run programs that use the PC's hardware directly; they are not compatible at the PC-hardware level. From a practical viewpoint, the difference between a PC-hardware-compatible accelerator and an object-compatible accelerator is that an object-level accelerator will run any well-behaved MS-DOS application, while a PC-hardware-compatible accelerator will run any operating system or application that controls the hardware in the machine directly. Clearly, the PC-hardware level of compatibility is the ultimate and will run the broadest range of your existing applications.

Because there are special situations where object-level compatibility can have big payoffs, I must be cautious about simply blacklisting object-level-compatible accelerators as incompatible. The very aspects that make them only partially compatible can sometimes result in increased processing power. However, you must carefully weigh this increase against the fact that they will never work with all your software. The coprocessor class of accelerators is normally object-level-compatible, while emulators and motherboard boosters are object-level-compatible *and* PC-hardware-level-compatible.

Coprocessor Boards and Accelerators

The first class of accelerators is the oldest. Its members leave the 8088 intact and provide a second processor on an expansion card that fits into the I/O channel. Because those accelerators have two active processors, they occasionally exhibit some parallel execution or coprocessing, hence the name. "Co" can also imply that either processor can control the system bus.

Coprocessors don't necessarily accelerate; their most common job is to run code from a previous generation of machines on a new machine. For example, the IBM RT PC will be available with an 80286 coprocessor card that will let it run normal PC programs. This will let RT owners break out of their engineering applications long enough to run off a few worksheets with Lotus 1-2-3, for example. Another example of a coprocessor that was developed to ease the transition to the original PC is the Baby Blue, which could run CP/M-80 programs.

Today the most popular PC copro-

(continued)

cessor products are those that let alternate sophisticated operating systems and processors run in the AT. The operating system that drives most of these coprocessor boards is UNIX, and the processors of choice for this job include the IBM 370 on a board, the Motorola 68000/68020, and the National Semiconductor 32032. The key to three of these coprocessor boards is that they let minicomputer software developers move their UNIX applications to the AT. At a recent CAD/CAE show, I saw at least five $30,000 electronic-design packages, each of which ran on the AT and came with its own $6000 accelerator. While some argument can be made that the C implementations on some of the minis are more powerful than those available on a PC, I don't think the work involved in porting these applications to UNIX on the coprocessor boards was much more involved than a C port straight to an AT.

When a coprocessor board uses a processor that is object-level-compatible with the 8088, I call it a coprocessor accelerator. This type of accelerator offers two advantages over a conventional PC. First, the boards have two processors; the original 8088 stays in place to handle I/O tasks, while the on-board coprocessor is available for CPU-bound tasks. Because the 8088 is not removed, a coprocessor accelerator is slightly easier to install than an emulator. Second, since the I/O interface between the CPU on the accelerator and the 8088 is nonstandard, the cards will usually work only with MS-DOS–compatible software that does not use the hardware directly. Since your nonstandard architecture is going to force you to write an operating-system-interface layer, it pays to add other nonstandard features such as increased system memory. I am not familiar with all the benefits of all the boards on the market, but these nonstandard benefits appear to include increases in DOS memory and the ability to run several coprocessor cards in the same PC.

Faster I/O is not usually a result of installing a coprocessor accelerator. I/O ought to be faster in this type of accelerator and is when the bottleneck in an application is the time spent by the operating system handling I/O calls. However, this is rarely the case. Operating systems such as MS-DOS spend a small proportion of their time in the OS as opposed to the hardware driver. On the other hand, sophisticated multiuser and real-time operating systems often bog down in the layers of the OS. Unfortunately, these operating systems rarely run on coprocessor accelerators, so this

(continued)

benefit is not realized in practice. The usual reason that I/O is slow is slow I/O devices. I will show this in the benchmarks, where the range in I/O throughput varies over a factor of 60 to 1 in going from RAM disks to floppies.

When really fast I/O devices are used, coprocessors lose ground with respect to emulators. This is the result of the fact that they use the relatively slow 8088 for I/O, whereas the emulator uses its on-board 8086 or 80286 for I/O. A novel installation I was involved in recently used an emulator accelerator to speed up motherboard I/O by effectively replacing a PC's 8088 with an 8086. It also used an Orchid Turbo board with an 80186 that accelerates CPU-intensive activities. While this installation worked, I am not convinced that the combination is any better than a straight 8086 board running with a RAM BIOS that speeds up the ROM I/O routines by moving them to 16-bit RAM. A simple analysis of the situation reveals that, since MS-DOS is single-threaded, there would be no parallel processing of I/O with CPU-bound activities. Consequently, the two boards would alternate back and forth, and the total time to do a job would be the sum of the time taken by each board. Since the 9.54-MHz 8086 board has a higher throughput than the 8-MHz 80186 on the Orchid Turbo, my guess is that an 8086 board by itself would have run as fast or faster than the combined installation.

The main benefit of coprocessor accelerators, the possibility of achieving concurrent I/O, is also their biggest drawback. The key to leaving the 8088 behind on the system board is a software layer that translates calls that would normally go to the ROM BIOS into commands that now have to be shipped to the system board's 8088. As long as you are running MS-DOS applications that avoid direct manipulation of the hardware, all is fine. However, any time an application or new operating system attempts to directly control the hardware, a problem arises: The coprocessor board has a different architecture than the original machine and responds differently to hardware commands. Operating systems such as XENIX, RTOS, UCSD p-System, and Concurrent DOS, and operating environments such as Window, DESQview, DoubleDOS, and GEM are all examples of programs that have to directly control the hardware and will not work with a coprocessor unless the interface layer between the board and the 8088 is rewritten for each application. Reworking this interface turns out to be an incredibly expensive task and is the reason why popular coprocessor boards such as the Orchid Turbo, Pfaster-286, and

(continued)

PCelevator work only with MS-DOS and applications that avoid controlling the hardware.

EMULATOR ACCELERATORS

The second class of accelerators requires that you remove the 8088 on your motherboard using a simple tool and replace it with a 40-pin plug that is connected to the accelerator, which you then install in the I/O channel. This type of board normally includes a 16-bit CPU and the circuitry required to let the 16-bit CPU mimic the 8088 using the 40-lead ribbon cable that plugs into the 8088 socket. Since this process is usually called processor emulation, I refer to this class as emulator accelerators.

There are at least three types of emulators on the market. These can be distinguished by the type of memory that resides on the accelerator and include no memory, 16-bit cached memory, and 16-bit true memory. Cards are available that provide a combination of cache and pure memory. From the viewpoint of speed, the more pure memory a board has, the faster it will run. Whatever type of memory is used, the processor on an emulator has to be an 8088 object-compatible (8088, 8086, 80186, 80286, V20, or V30) device that is capable of running at a clock speed greater than 4.77 MHz.

The classic emulator accelerator gets most of its speed by the increase in data-bus bandwidth between the CPU and the memory resident on the accelerator. Simply stated, the 8086 is hooked directly to a 16-bit memory bank on the accelerator via a 16-bit data bus. This bus is twice as wide as the 8-bit data bus on the motherboard and is running at twice the clock speed. Multiplying the increase in width (two) by the increase in frequency (two) gives a data-bus bandwidth increase of four.

Unfortunately, the accelerator has to communicate occasionally with the motherboard through the 40-lead cable that plugs into the 8088 socket. All communication with the motherboard requires that the 8086's 16-bit

(continued)

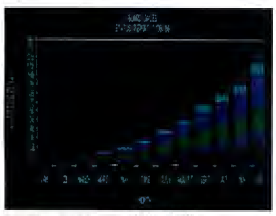
Seal of Approval

Lotus certifies ATI's Graphics Solution

ATI's **"Small Wonder" Graphics Solution** has never been slow to pick up compliments, particularly in the area of multi-mode flexibility. Much of the credit for this is due to ATI's use of advanced technology. Therefore it's not very surprising that Lotus – the world's leader in software – has certified the "Graphics Solution" for 132 x 44 monochrome spreadsheets and 640 x 200 7 color graphics so you can see more spreadsheet and sharper, more colorful graphs on 1-2-3 (2.0) and Symphony (1.1).

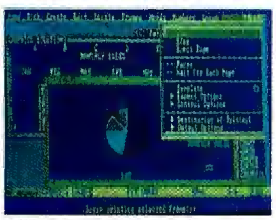
At ATI, we think we've developed a better graphics card. It just might surprise you how much better. Call us at **(416) 477-8804** for more information.

LOTUS CERTIFIED 132 x 44 MONOCHROME SPREADSHEETS



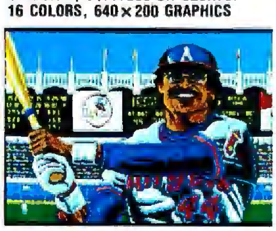
LOTUS CERTIFIED 640 x 200 HIGH RESOLUTION COLOR/GRAPHICS

COLOR SOFTWARE ON MONOCHROME MONITOR IN 16 SHADES



ASHTON-TATE FRAMEWORK II
4 COLORS, 640 x 200 ON DESKTOP
16 COLORS, 640 x 200 GRAPHICS

HERCULES COMPATIBLE MONOCHROME GRAPHICS



PLANTRONICS COMPATIBLE HIGH RESOLUTION 16 COLORS

THE GRAPHICS SOLUTION

FEATURES

- **Hercules Compatible Monochrome Graphics**
- **Runs Color/Graphics Software on Monochrome Monitor in 16 Shades Without Pre-Boot Drivers**
- **High Resolution Color/Graphics — 320 x 200 16 Colors**
  — 640 x 200 4 Colors
  — 640 x 200 16 Colors
- **IBM Compatible Color/Graphics**
- **132 Columns in Either Color or Monochrome**
- **Lotus Certified Drivers for 132 Columns and 640 x 200 High Resolution Color/Graphics**
- **Drivers for 640 x 200 High Resolution 16 Colors on Framework II**
- **Parallel Port and Serial Port Optional**
- **2 Year Warranty**
- **$299 Suggested Retail**

DISTRIBUTORS

F.A. COMPONENTS
NY (718) 507-1444, outside NY (800) 847-4148
IN (219) 432-8540, outside IN (800) 331-7567
SC (803) 288-2422, outside SC (800) 845-2747
WA (800) 321-6012, outside WA (800) 426-1412

SAI America – MD (301) 459-2100, outside MD (800) 638-9579
Compuserve – CAN (416) 477-8088
EMJ Data – ONT (800) 265-7212, outside ONT (800) 837-2444
Electronic Marketing Specialists
S.CA (714) 832-9920, (818) 708-2055, (619) 560-5133
N.CA (408) 245-9291, UT (801) 268-9866
CO (303) 745-0745, AZ (602) 840-8855
WA (206) 575-0223

NOW! ATI ALSO HAS DRIVERS FOR ASHTON·TATE* FRAMEWORK II

Small Wonder From ® ATi TECHNOLOGIES INC.

450 Esna Park Dr.
Markham, Ontario, Canada L3R 1H5
TELEX 06-966640 (CMC – MKHM)

Technology you can Trust.

*Trademarks: Lotus, 1-2-3, Symphony, — Lotus Development Corporation; Framework II — Ashton-Tate Inc.; PC Paintbrush + — Z-Soft Inc.; IBM — International Business Machines Inc.; Hercules — Hercules Computer Technology Inc.; Plantronics — Plantronics Enhanced Graphics Corp.

*This Lotus logo is used by permission of Lotus Development Corporation.

The Graphics Solution has successfully passed compatibility testing with 1-2-3 (2.0) and Symphony (1.1) on an IBM PC-XT with PC-DOS 3.1. The Graphics Solution must be installed.

Inquiry 25

requests be converted into two 8-bit requests to the motherboard. This is done by a combination of buffers and latches—controlled by the main accelerator logic—that idles the 8086 for sixteen 9.54-MHz clock cycles while it generates two 8-cycle I/O requests to the motherboard. These requests are synchronized with the PC 4.77-MHz clock, which interprets them as two normal 4-cycle requests. The control circuit must also generate 8088 status signals whose job is to keep the PC motherboard asleep during 9.54-MHz operation and turn it on during emulation cycles.

All normal I/O cycles need servicing and include memory reads and writes, I/O reads and writes (to ports), and DMA requests made by the DMA controller on the PC's motherboard. DMA requests can be complicated. You can't boot a PC unless it has memory on its motherboard, so you have to leave some memory in the PC's motherboard. Also, the DMA controller on the motherboard will see this memory in the normal course of events. This brings up an interesting situation.

What happens if the DMA controller does a read of memory in the case where that memory resides both on the motherboard and on the accelerator? The answer is that, unless both the 8- and 16-bit memories contain the same information, there will be a battle over the bus and the board that has the stronger drivers will win. Two techniques exist for eliminating this conflict. Whenever you write to an address that is populated on the motherboard and the accelerator, you do a simultaneous write to both. This will result in a small reduction in speed for programs that write frequently to low memory; you can almost eliminate it if you limit low memory to 64K bytes or fill it up with a utility such as a RAM disk.

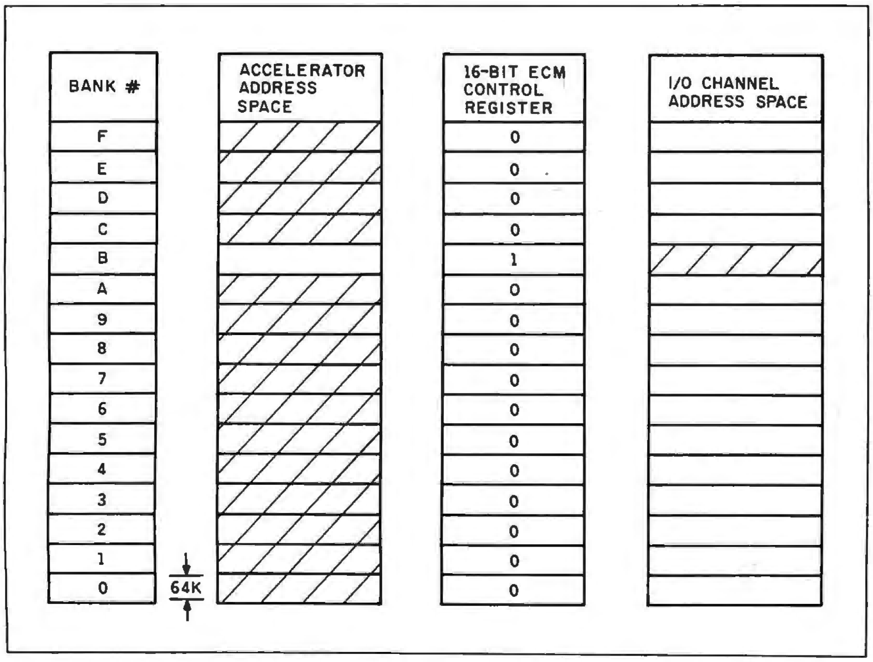


Figure 1: *Extended Conventional Memory uses a 16-bit register to partition the 8086's address space from that of the I/O channel. Each register bit controls whether a corresponding 64K-byte bank of the address space is on board the accelerator or in the I/O channel. The register in this figure shows the state of the machine when a 1-megabyte DOS partition has been interrupted to read or write to a graphics adapter that is addressed through bank B. If an Enhanced Graphics Adapter were being written to, the driver would have to switch both banks A and B. (The cross-hatched areas indicate currently active banks.)*

The other solution is to duplicate the four DMA high address bit registers that the PC has on its motherboard with a similar set of registers on the accelerator and then write a phony value into the registers on the motherboard that points to an address for which the motherboard has no RAM. Without proper DMA control, neither your floppies nor your hard disk will work. If the emulator accelerator you purchase does not have a DMA test program, you could be asking for trouble. Make sure you test system DMA if possible and, if you cannot, start off with floppies only or back up your hard disk.

Whenever the speedup board reads or writes to the motherboard or I/O channel, the data-bus bandwidth falls back to that of the PC's original 8088. However, in most programs the number of emulation cycles that have to be run is very small. To help reduce this count, many emulators have a feature that maps the highest 64K-byte bank of memory on the board into bank F of the 8086 address space. After this is done, a software routine moves the contents of the motherboard ROMs to RAM. This typically results in a factor of three speed increase in interpreted BASIC programs and some I/O, especially screen I/O. Of course, this technique also reduces the amount of 16-bit memory available for DOS from 640K to 576K bytes, but it is usually worth it. In cases where memory is dear, it is possible to place a bank on another board in the I/O channel or buy an emulator that can be populated over 640K bytes.

Figure 1 shows the memory-mapping scheme for a 1-megabyte accelerator that I designed. This board uses a 16-bit register to control how memory is mapped. When a bit in the register is on, the corresponding bank will reside on the 8-bit side of the machine. When a bit is off, the bank is on the accelerator. For example, if all the bits are off, all the memory in the system is on the accelerator.

This brings up an intriguing question: What prevents this system from

(continued)

A GREAT DEAL JUST FELL INTO YOUR LAP

This is the one you've been waiting for.

The remarkable new BONDWELL 8.™ At $1595.00 retail, it is the lowest priced IBM compatible lap-sized portable computer on the market today.

It's a great deal for a small price. With built-in features that include 512K RAM, 720K 3½" drive, modem, back-lit LCD display, parallel and serial ports, composite and RGB outputs, real time clock and AC adapter.

High performance value. It's what you expect from Bondwell's entire line of desktop, transportable, and lap portable computers.

Order your BONDWELL 8 today at (415) 490-4300. Don't let a great deal slip through your fingers.

Bondwell™

ONE BETTER

 Inquiry 316

running with a DOS that has 1 megabyte available to it? The answer is: not very much. The key is a little software routine that traps screen I/O requests to the ROM BIOS and swaps the graphics-adapter memory in and out as needed (i.e., turns on bank B by writing to the memory-control register with bit B set on). You also must change the value of the DOS memory size byte (in low memory) and then call for reboot with INT 19h. I make this sound a little easier than it is. You also have to worry about resizing DOS back to 640K or 704K bytes, moving hard disk drivers out of the way, overwriting BASIC, and so on. However, all these problems are surmountable and result in a DOS that has 1016K bytes available to it. In fact, the only memory not available is the 8K bytes taken up by the ROM BIOS in high memory.

I call this technique Extended Conventional Memory (ECM). (Figure 2 shows the block diagram of an accelerator incorporating ECM.) It works well with applications generated by existing FORTRAN, BASIC, and C compilers that do not address the screen directly (i.e., that use DOS or INT 10h). Since most applications compiled by end users fall into this category, this board is a useful partial solution to the memory squeeze play. However, ECM will work only with applications like Lotus 1-2-3 and large memories if the screen drivers are modified to swap bank B before and after every screen operation. In the case of Lotus 1-2-3 and most products that have separate screen drivers, this job is not a big one. Fortunately, if a product does not work with ECM and it does not pay to patch it, you can

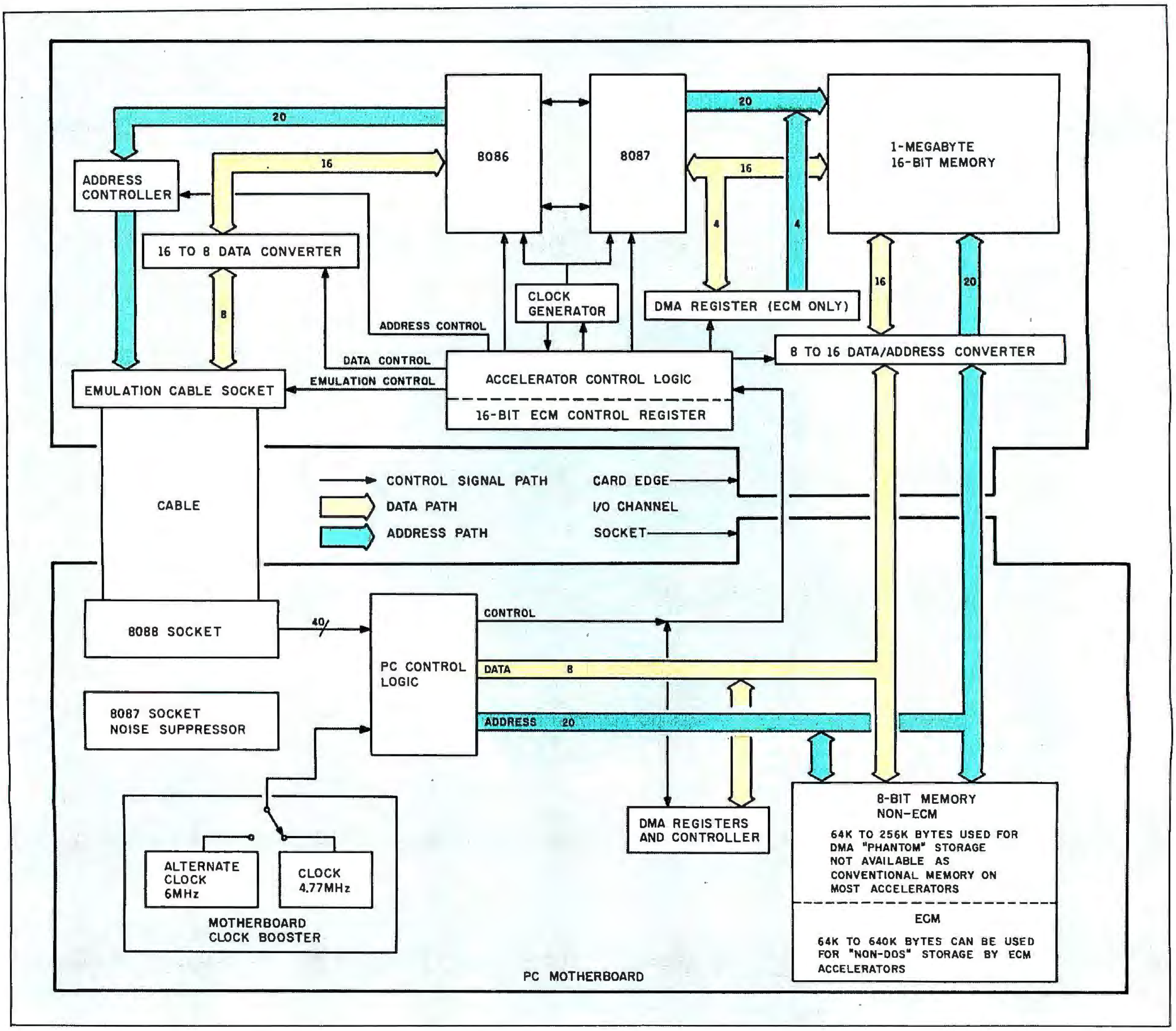


Figure 2: *The block diagram for an accelerator board design incorporating the Extended Conventional Memory scheme.*

always reboot the board to a 640K- or 704K-byte machine. The memory-control register also makes it possible to use the memory on your motherboard and in your I/O channel for I/O utilities such as RAM disks, print spoolers, and disk caches.

Another popular form of emulator uses cached memory instead of real memory. Cached memory is composed of fast static RAMs and is usually small. The theory behind cached memory is to save money on fast RAM by using a small, very high speed cache for tight program loops. In normal practice, this means placing a small cache inside the processor itself. For normal accelerators, there is no cost issue involved in buying faster RAM. The same 150-nanosecond devices that work on your motherboard at 4.77-MHz work fine on a 9.54-MHz 16-bit accelerator.

What is the issue then? First, duplicating existing RAM. Some people like to think that the 256K to 640K bytes that they paid dearly for two years ago is still worth something. The second issue is space. It is possible to build a half-size accelerator without going to semicustom chips if you use a cache instead of true RAM. As an advocate of true RAM, I believe that having all your memory set up as 16 bits instead of 8 guarantees that all CPU operations speed up at a reasonable incremental cost. The 36 chips that make up a megabyte these days cost about $80 as compared with six 55-ns static RAMs that cost $42. The other benefit of a megabyte of RAM on an accelerator is that you can do other things with it, like ECM.

I will now examine how cache is used in an emulator. The theory is simple: Well-written programs spend 90 percent of their time in 10 percent of the code. If this code is mostly in a small fast memory that has to be updated only when you pass out of the hot code (the 10 percent that runs the most), you have a cost-effective way of building a system that minimizes the use of expensive fast RAM. The 90 percent/10 percent principle for programs reads a little differently at the machine level. What typically happens is that a small number of routines that are several hundred bytes long become the hot areas. The key to designing a cache is to figure out where these areas are, keep them in the cache, and go to slower system memory only when other information needs to enter the processor.

This is easier said than done. In practice, you don't examine a program to decide what to cache and what not, but you build a machine that automatically brings information into the cache whenever it is needed but is not already in the cache. In this way, the sections of a program that are hot will end up in the cache because they repeat (i.e., if the cache is long enough, they will still be in the cache from the previous access).

Caching is accomplished via elegant schemes, one of which is called a write-through cache. An 8K-byte write-through cache is composed of three parts: a linear array of static RAM that is organized as 4096 sixteen-bit words, 4096 tag bytes, and logic that is capable of comparing the tag bytes with the address bus to determine whether the cache contains the data of interest. The entire 1-megabyte address space of the processor is mapped onto this 8K-byte area using a simple technique.

When the 16-bit processor on the accelerator accesses a word, the lowest 12 bits are used to index into the tag array, and the 8-bit value of the tag is then compared against the highest 8 bits on the address bus. A match means that the 16 bits of data in the cache at the index location are valid and the processor is passed these bytes. This is called a cache hit. If the tag does not agree with the address bus, the value in the cache is incorrect and that causes an 8088-emulation cycle that reads 16 bits of data from the motherboard. When this information comes up, it is simultaneously read into the processor and the cache, and the tag for the address in question is changed. Any write from the processor to memory automatically writes to both the cache and the motherboard; this is the write-through. In this scheme, the data-bus bandwidth between the processor and cache memory is 100 percent for repeat reads of the cache. Memory reads that miss the cache and writes run at the speed of the 8088.

For the 7.2-MHz 80286 cache board used in the benchmarks, the data-bus bandwidth between the processor and the cache was 6 times that of the 8088 in a PC. Assuming that the cache were to be hit 5/6 of the time (82 percent), this would result in an overall bandwidth that was three times that of a PC. Simply stated, the misses that happen 1/6 of the time take as long to execute as the hits, since the misses run 6 times slower. This doubles the execution time, which effectively halves the data-bus bandwidth. As I will show below, the cache's no-wait-state memory is so fast that it is mostly wasted. Only about 5 percent of the instructions in a no-wait 80286 system are limited by the data bus, while nearly 66 percent are limited in a one-wait-state 80286 or 8086 system. What this means is that the high-speed data-bus bandwidth is less than 6.0, which implies that programs that hit only 82 percent of the time speed up less than 3 to 1.

The final emulator-type accelerator has no memory on it at all. The idea is simple: Increase the speed of register instructions only. This type of device is best described as an 8087 accelerator, as the 8087 is the only component that spends many more cycles on the average executing than it does reading the bus. These devices involve little logic and a small increase in speed; they plug directly into the 8088 only.

Most emulators are PC-hardware-compatible. The main compatibility problems are the result of hardware that is being driven with software that contains timing loops. For these devices, it is necessary to either slow the board up or patch the drivers. The other problem is software that knows it is on a PC and can't find an 8087 (because the system has an 80287) but knows the 8087 toggle is set to on. The best example of this problem was an early version of the Intel

(continued)

Above Board that did not take kindly to 80287s on accelerators in PCs. Another incompatible accelerator is the ECM. As long as it is running in the ECM mode, it works with all software that obeys the rules IBM set down with PC-DOS. When an application disobeys those rules, the board must be reset with software to the 640K bytes that most software expects. Clearly, it helps to have a firm grounding in the operation of both PCs and DOS to overcome the challenges that running at high speed produces.

MOTHERBOARD ACCELERATORS

The third class of accelerators is the simplest. Its members provide an alternate clock frequency for the 8088 on your motherboard (or your AT), upgrade the 8088 to an 8088-2 or NEC V20, or do both. Lately, this group of products has become more sophisticated. One product lets you control the processor's speed by a switch and software, while another inserts wait states during I/O cycles, thus discriminating between I/O cycles and internal CPU cycles. Without wait states, it is possible to run some motherboards up to 8 MHz.

The motherboard boosters come in a number of forms. The key to their use is the PC's 8284 clock chip that is socketed in about 60 percent of the machines that IBM built. The principle is simple: Remove the chip and plug in a board that pipes new clock signals down onto your motherboard. Of course, you might not have a socket. The solution is to have a reputable repair person replace your chip with a socket. The latter is easy to do yourself, if you know what you are doing. The risk is that you might have to replace your motherboard, a $200 expense. If you want to get the best performance from your motherboard, buy one that is guaranteed to work at 8 MHz and is socketed. This eliminates having to remove 200-ns RAM chips from your old PC motherboard, and motherboard accelerators run faster with clones than they do with IBM motherboards.

The main problem with the inexpensive clock-upgrade approach is that it can extend the system components out of their rated timing specifications. While it is possible for the system to run at speeds above specification, it will probably be more prone to intermittent failures and crashes. In addition, it is often a hit-or-miss business figuring out exactly how far above specification a particular PC is capable of running. The uncertainty results from the fact that a PC is a real-time system, and it is possible to have rare combinations of events occur that overextend the system. On the positive side, a PC performs 200,000 to 500,000 operations per second and, except for unusual events, it should not take it long to explore its operating-phase space.

The motherboard accelerator route, if you pursue it with care, can be the most cost-effective form of acceleration. It is important to buy an accelerator that you can slow down with software or that comes with its own device drivers for the floppy and hard disks. My experience shows that, up to 6.7 MHz, your machine will either run or crash. If it crashes, you have an early PC or one built with marginal components. Above 6.7 MHz, the speed of RAMs and the *quality* of your floppy and hard disk controllers starts to become critical. Above 7.4 MHz, you will definitely have to slow up when you do I/O to your storage systems. This is most easily accomplished with a resident driver that traps requests to INT 13h, slows the accelerator to 4.77 MHz, waits for the I/O to complete, and then speeds back up. With such a driver, it is possible to run some motherboards up to 8 MHz. My experience shows that

Table 1: *Results from runs of the Performance Index benchmark on several different accelerator boards.*

Components of the PI Benchmark

Machine	Numeric Data Processor		Data Bus		Register			Disk	
	1	2	3	4	5	6	7	8	9
PC 4.77	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
PC V20/8	1.71	1.71	1.71	1.70	3.59	5.51	1.77	1.0	1.0
PC2	2.06	2.08	4.05	4.02	2.00	2.03	2.74	.72	.76
PC2 V30	2.11	2.26	4.09	4.06	4.25	6.70	2.79	.77	.77
PC2 12	2.65	2.80	5.17	5.17	2.56	2.54	3.53	.74	.74
AT 6/4	.72	.94	3.27	3.34	3.46	7.74	2.53	3.03	1.1
AT 6/8	1.06	1.54	3.27	3.34	3.46	7.75	2.54	3.03	1.1
AT 6/10	1.15	1.77	3.27	3.34	3.46	7.75	2.54	3.03	1.1
AT 8/8	1.23	1.68	4.50	4.54	4.73	10.9	3.41	3.14	1.1
AT 8/10	1.37	1.98	4.50	4.46	4.78	10.9	3.44	3.90	1.1
PC 7/8	1.2	1.61	4.58	.99	3.19	9.1	2.18	3.0	1.0
					6.00(b)		1.92(b)		
PC 7/10	1.47	2.00	4.57	.98	3.24	9.6	2.15	3.2	3.9
					6.00(b)		1.92(b)		

Legend:

Machine	Processor	MHz	NDP	MHz
PC 4.77	8088	4.77	8087	4.77
PC V20/8	V20	8.0	8087	8.00
PC2	8086	9.54	8087	9.54
PC2 V30	V30	9.54	8087	9.54
PC2 12	8086	12.00	8087	12.00
AT 6/4	80286	6.00	80287	4.00
AT 6/8	80286	6.00	80287	8.00
AT 6/10	80286	6.00	80287	10.00
AT 8/8	80286	8.00	80287	8.00
AT 8/10	80286	8.00	80287	10.00
PC 7/8	80286	7.20	80287	8.00
PC 7/10	80286	7.20	80287	10.00

IBM motherboards top out at 7.4 MHz and good-quality clones run up to 8 MHz.

The motherboard-acceleration technique has one advantage over the coprocessor and emulator techniques: It speeds up motherboard operations, including DMA, which is not affected by the other two classes of accelerator. No matter what technique you use for accelerating, you will have some motherboard operations that detract from overall speed, so picking up the motherboard speed is a good way to bring up the system's slowest activities. When combined with an emulator or coprocessor, the motherboard accelerator forms a hybrid system that improves the performance of all the components, including the accelerator.

Using this technique, I have boosted a motherboard to 6 MHz and run a 9.54-MHz 8086 accelerator at 12 MHz. At this speed, the 8086 outperforms an 8-MHz 80286 in all but two benchmarks (see table 1). The 12-MHz 8087 on this board beats an 8-MHz 80287 by over a factor of two for the most common operations. Not all accelerators can be boosted with this technique. The two that I have successfully boosted required design changes to run at the higher speed.

For speeding up word processors or running 8087-bound applications, the motherboard accelerator is an exceptional buy. An 8-MHz V20 system does numerics faster than an 8-MHz AT and generally increases throughput by 70 percent for only $149 ($298 if you need to replace your old 8087).

Benchmarking Accelerators

People buy accelerators because they want to spend less time waiting for results. Unfortunately, most are not in a position to try a number of accelerators running with their specific applications before they buy. At this point, you might be wondering if there is a simple universal benchmark comparing all accelerators and predicting the relative speed of these boards running any application. Such a benchmark cannot exist because most PC applications fall into one of at least five different categories and, since it takes at least four measurements to gather even basic information, one benchmark could never do.

The five categories are number-bound, mass-storage-bound, data-bus-bandwidth-bound, integer-register-bound, and bound by two or more of the four processes. A sixth type of instruction, which grossly distorts many of the published benchmarks, is dominated by integer multiplies and divides; I have not included it as a category because it plays such a limited role in real programs. This is documented in my BYTE article "The 8087/80287 Performance Curve" (*Inside the IBM PCs*, Fall 1985).

In that article, I showed that for a particular accelerator running an 8086, one-third of the instructions were bound by the internal processor throughput, which is proportional to clock speed (and which I therefore call clock-bound or register-bound), and two-thirds were bound by how fast the processor could fetch new instructions (data-bus-bandwidth-bound). In particular, I found that a 9.54-MHz 8086 runs three times faster than a 4.77-MHz 8088 and that this speed could be derived using a simple theory that can be summarized as follows: The average of the speedups (two for clock-bound instructions and four for data-bus-bandwith-bound instructions) equals the average speedup, three. In this article, a similarly interesting principle applies when there are more than two processes determining the speedup of an application: The overall increase in speed is equal to the average of the slowest and fastest processes that comprise 10 percent or more of the total instructions. This principle could be thought of as taking the average of significant extremes and throwing out processes that rarely occur.

Before I present my benchmark and findings, I want to examine a famous benchmark that reviewers and manufacturers everywhere use but which often yields anomalous results: SI by Peter Norton. This benchmark, which is part of Norton's SysInfo utility, can over- or underestimate the average speed of a board by as much as 100 percent. The best example is the 9.54-MHz 8086 accelerator. SI usually yields a value of 2 for this processor, which, to take the benchmark literally, would imply that a 9.54-MHz 8086 is only twice as fast as a 4.77-MHz 8088. Since that is exactly equal to the increase in clock speed, an innocent application of SI generates the false conclusion that there is no benefit in going from an 8-bit to a 16-bit data bus. I know for a fact that a 9.54-MHz 8086 runs three times faster than a 4.77-MHz 8088 when executing typical applications and that all this fuss over 16- and 32-bit data buses is for real. This also tells me that SI is composed of 100 percent clock-bound operations, which is why it scales with clock speed when run on an 8086.

The NEC V20 and V30 speed up a number of clock-bound operations, such as integer multiplies, divides, and string operations. For typical applications, it is reasonable to expect a 5 to 15 percent increase in speed from the NEC V-series processors over their Intel counterparts; you can occasionally get up to a 40 percent increase when performing floating-point arithmetic without an 8087.

So what happens when I replace the 9.54-MHz 8086 with a 9.54-MHz V30? I get a jump in SI results from 2.0 to 4.0. Where SI underestimated the 8086's performance, it now overestimates the V30's performance. If SI were a true index of system speed, it would read about 3.1, not 4.0, for a V30 running at 9.54 MHz. The interesting fact that I have uncovered in SI is that it is bimodal: On the 8086 it is clock-bound, while on the V30 it becomes data-bus-bandwidth-bound. I came to this conclusion without the aid of a debugger, but I suspect from an examination of the pure benchmarks that make up PI benchmarks that SI is partially made up of block moves and integer arithmetic, two rarely used operations in compiler-generated code.

Finally, SI acts strangely on some compatibles, and it is possible to get

(continued)

speedups like 101 or different values every time you run the benchmark. In general, I would divide any advertised SI claim for an 80286 machine in half, as the benchmark was inflated by a factor of two in the first place by its sole use of operations that are rarely used and are among the slowest in the 8088 repertoire. If the manufacturer of an 80286 board claims it is 6.6 times faster than a PC, the estimate is probably based on SI, and 2.7 to 3.3 is a much better estimate. This rule does not hold for 8086 boards, which actually run faster than SI by 50 percent. In conclusion, for a benchmark to be meaningful, it must sample the mainstream of processor activities in the user's applications and it must be documented.

THE ACCELEROMETER PI

By now, I hope that you are convinced of at least three things. First, a number of underlying processes come into play in a system. Also, an understanding of what those processes are is fundamental to using benchmarks for making system measurements. Finally, benchmarks that measure pure processes can be useful in predicting what happens in complex real-world situations. Toward that end, I designed a suite of benchmarks to measure the pure processes that go on in object-compatible Intel accelerators.

My accelerometer is called PI (see listing 1), which stands for Performance Index. It was my intention to create a program that would let users design their own complex benchmarks from a number of pure ingredients that I built into PI. The results of running PI on a number of different machines are in table 1. [*Editor's note: The Performance Index benchmark,* PI.BAS, *and several associated files are available on disk, in print, and on* BIX. *See the insert card following page* 176 *for details. Listings are available on* BYTEnet; *see page* 4.]

PI is composed of nine standard benchmarks and two alternates that measure interesting properties of cache cards or 80286 cards with no wait states. There are four groups of benchmarks: numeric, data-bus-bandwidth-limited, register- or clock-bound, and disk-bound.

INTERPRETING THE RESULTS

Benchmarking accelerators using the PI benchmarks resulted in two points of interest: the data-bus bandwidth of the PC and that of the AT. At 4.77 MHz, an average PC clock cycle takes 210 ns, and it takes four cycles to perform a memory read or write; that is, it takes 840 ns to move a byte over the 8-bit-wide bus that results in a data-bus bandwidth of 1.19 megabytes per second. This pure band-

(continued)

"USER-FRIENDLY" REDEFINED.

Now you can Control and Protect your PC Resources and make them a lot friendlier at the same time.

TheEMCEE will reduce your cost of support, increase productivity and protect your sensitive data from being viewed, altered or deleted by unauthorized users.

It will not allow access to your hard disk via a control break or a boot from A.

When you engage TheEMCEE, it brings a host of feature presentations to the show:

- **Advanced Menus**
- **System Log**
- **Network Compatibility**
- **Encryption Security**
- **Password Security**
- **Enhanced DOS**
- **Optional File Manager**

TheEMCEE takes the "oops" out of personal computing.

Instead of getting a blank screen with a DOS prompt when you turn on your machine, you get your own system of menus.

You define the menus, the user prompts, passwords, system log and encryption features, even the color, with lots of on line help there to assist you.

And you can control user access to any item on any menu using up to 99 levels of password protection for any selection.

Then, with a few keystrokes you can run any program, command, batch file, menu or hidden item.

Access files and programs in other directories or on other directories or on other drives.

If you are a user, TheEMCEE automates and simplifies your system.

If you have the responsibility for corporate PC's, TheEMCEE is the applications manager you have been looking for.

It lets you set up a friendly interface that reduces support, and at the same time provides the control and protection you need.

Now, wouldn't you like to have TheEMCEE direct your show?

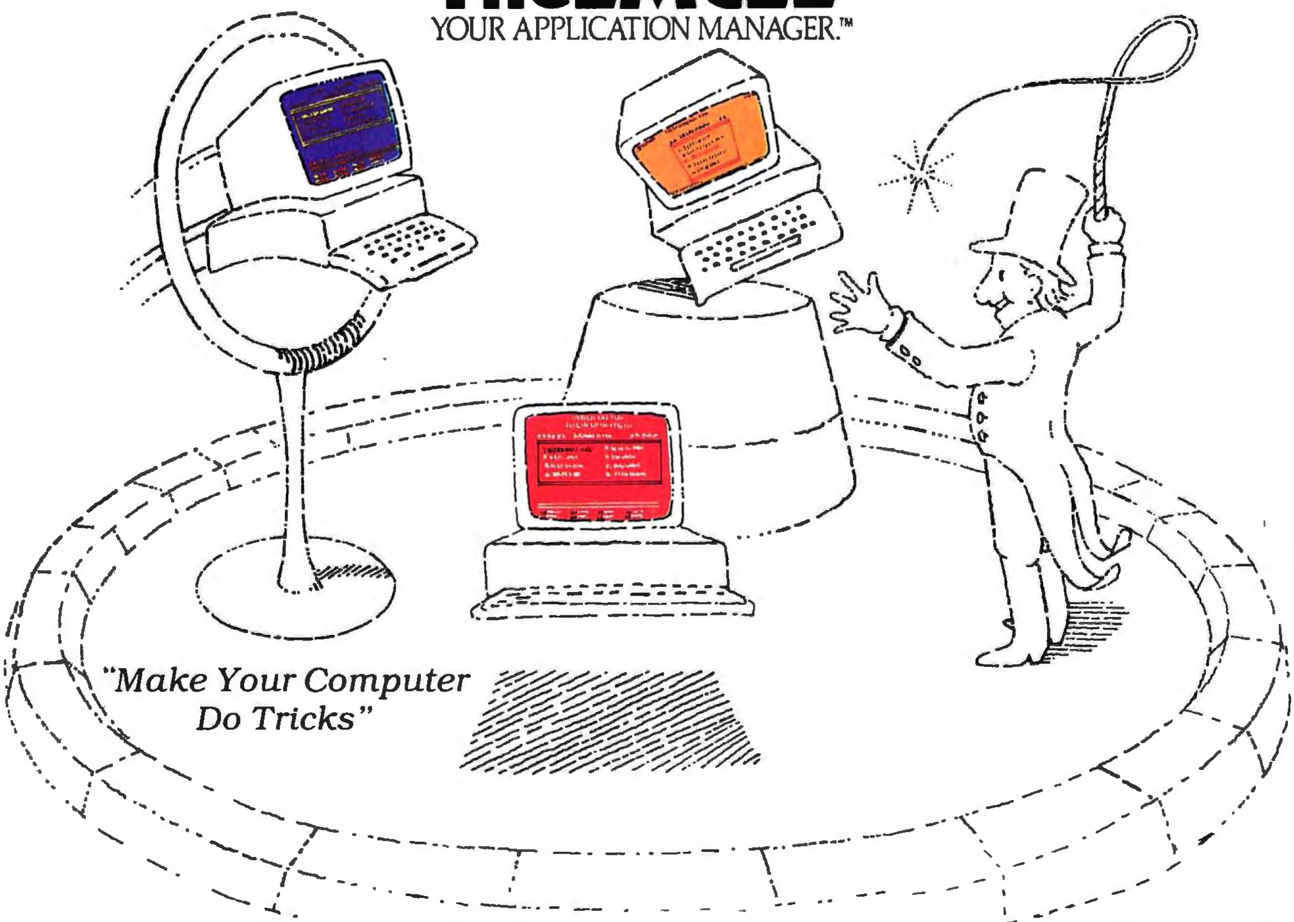
And it's all yours for just $49.95 with a money-back guarantee—what could be friendlier than that?

TheEMCEE, because it's a three ring circus in there.

$49^95

(800) 423-9147
(213) 457-1789

TheEMCEE™
YOUR APPLICATION MANAGER.™



COMMAND
Software Systems
Malibu, CA

Listing 1: The Performance Index benchmark main program. This IBM BASIC Compiler (version 2.00) program calls several assembly language routines to perform the actual benchmarks. The program uses the object module TIME100 to perform the timing function.

```
10 REM  PI.BAS -- Performance Index Benchmark Program
20 REM Copyright (C) MicroWay, Inc., 1986
30 REM MicroWay, Inc.    PO Box 79    Kingston MA 02364    (617) 746-7341
40 REM
50 REM Does not use Numeric Coprocessor (8087 or 80287).  See PI87.BAS for
60 REM a similar program which does use the Numeric Coprocessor.
70 REM Compiled with IBM BASIC Compiler V2.00 (/O switch).  Linked with
80 REM assembly language subroutines:
90 REM DBB4 DBBCACHE DBB16 CB4 CB16 IA XFER TIME100
100 REM
110 DEFINT A-Y
120 DIM NAME$ (11), SECTION$ (11), ZF (11), ZP (11)
130 DIM DSRC (10249), DDEST (10249)
140 COMMON DSRC (), DDEST (), SPC1!, SPC2!, SPC3!, SVA!, SVB!
150 DEF FNS# (X$) = 60# * (60# * VAL (MID$ (X$, 1, 2)) + VAL (MID$ (X$, 4,
2))) + VAL (MID$ (X$, 7, 2)) + .01# * VAL (MID$ (X$, 10, 2))
160 LINES$ = "" : FOR I = 1 TO 69 : LINES$ = LINES$ + CHR$ (205) : NEXT I
170 PRINT CHR$ (201); LINES$; CHR$ (187)
180 PRINT CHR$ (186); " M i c r o W a y   P e r f o r m a n c e   I n d e x
P r o g r a m "; CHR$ (186)
190 PRINT CHR$ (200); LINES$; CHR$ (188)
200 PRINT "                          Copyright (C) MicroWay, Inc., 1986."
210 PRINT "                  (This version does not use the Numeric Coprocessor)"
220 PRINT
230 DATA 0.6013,0.5607,0.6,0.6,0.595,0.589,0.62,0.6,0.6,0.6064,0.6064
240 DATA " 1"," 2"," 3"," 4"," 5"," 6"," 7"," 8"," 9","10","11"
250 DATA "Floating point -- arithmetic                       "
260 DATA "Floating point -- Savage (transcendentals)         "
270 DATA "Data bus bound -- 4K bytes of instructions         "
280 DATA "Data bus bound -- 4K bytes of register moves       "
290 DATA "Data bus bound -- 16K bytes of instructions        "
300 DATA "Clock bound -- 4K byte block moves                 "
310 DATA "Clock bound -- 16K byte block moves                "
320 DATA "Integer multiply and divide                        "
330 DATA "Subroutine calls and stack operations              "
340 DATA "Disk I/O -- sequential read from random file       "
350 DATA "Disk I/O -- sequential write to random file        "
360 DATA 25,25,11.111111,11.111111,11.111111,8.333333,8.333333,0,0,0,0
370 CALL TIME100
380 FIRSTTIME = 1
390 FOR I = 1 TO 11 : READ ZF (I)        : NEXT I
400 FOR I = 1 TO 11 : READ SECTION$ (I) : NEXT I
410 FOR I = 1 TO 11 : READ NAME$ (I)     : NEXT I
420 FOR I = 1 TO 11 : READ ZP (I)        : NEXT I
430 MAXWRITES = 20 : MAXREADS = 20
440 GOTO 570
450 REM *** Create file so that reads will have something to work on
460 REM *** and so that writes will not have to create new records.
470 IF MAXWRITES > MAXREADS THEN MPUT = MAXWRITES : ELSE MPUT = MAXREADS
480 OPEN "TEMP.TMP" FOR RANDOM AS #1 LEN=2048
490 FIELD #1, 2048 AS S$
500 T$ = "" : FOR I=0 TO 1023 : T$ = T$ + MKI$ (256 * RND (I)) : NEXT I
510 LSET S$ = T$
520 FOR I = 1 TO MPUT : PUT #1, I : NEXT I : CLOSE #1
530 PRINT "Please enter percentage of benchmark devoted to each operation."
540 FOR I = 1 TO 11
550 PRINT "Section ";SECTION$ (I);"   ";NAME$ (I);": "; : INPUT "", ZP (I)
560 NEXT I
570 ZT = 0 : FOR I = 1 TO 11 : ZT = ZT + ZP (I) : NEXT I
580 IF ZT > 99.5 AND ZT < 100.5 THEN GOTO 640
590 PRINT : PRINT "Caution: percentages total "; ZT;
600 INPUT " Adjust to 100% ? ", ADJ$ : PRINT : ADJ$ = LEFT$ (ADJ$, 1)
```

(continued)


BASF
10 TWO SIDE-DOUBLE DENSITY FLEXYDISKS
FlexyDisk
2D
The data you'll have to depend on tomorrow could depend on the disk you choose today.
BASF
BASF QUALIMETRIC™ 5.25" FLEXYDISKS® CERTIFIED 100% ERROR-FREE AND WARRANTED...
FOREVER

```
610 IF ADJ$ = "Y" OR ADJ$ = "y" THEN GOTO 630
620 IF ADJ$ = "N" OR ADJ$ = "n" THEN GOTO 660 : ELSE GOTO 600
630 IF ZT = 0! THEN GOTO 840
640 Z100 = 1# / ZT : FOR I = 1 TO 11 : ZP (I) = ZP (I) * Z100 : NEXT I
650 ZT = 100# : GOTO 670
660 FOR I = 1 TO 11 : ZP (I) = .01 * ZP (I) : NEXT I
670 PRINT "Section                    Description                    Performance
Index"
680 PRINT "======= =============================================
================="
690 ZTET = 0# : FOR J = 1 TO 11
700 ZITER = 60# * ZP (J) / ZF (J)
710 ITER = INT (ZITER) : IF ZITER - ITER > .1 THEN ITER = ITER + 1
730 ON J GOSUB 930,970,1000,1010,1020,1030,1040,1050,1060,1070,1140
740 ZET = FNS# (B$) - FNS# (A$)
750 IF ITER <> 0 THEN ZET = ZET * ZITER / ITER : ELSE ZET = 0#
760 PRINT "  ";SECTION$ (J);"    ";NAME$ (J);"      ";
770 IF ZET = 0 THEN PRINT "  ----" : GOTO 790
780 PRINT USING "###.##"; ZF (J) * ZITER / ZET
790 ZTET = ZTET + ZET
800 NEXT J
810 PRINT : PRINT "Performance index for entire benchmark is ";
820 IF ZTET = 0 THEN PRINT "not meaningful." : GOTO 840
830 PRINT USING "###.##"; .6# * ZT / ZTET
840 PRINT
850 INPUT "Would you like to run again, specifying the section weights
yourself? ", MORE$ : MORE$ = LEFT$ (MORE$, 1)
860 IF MORE$ = "Y" OR MORE$ = "y" THEN GOTO 900
870 IF MORE$ = "N" OR MORE$ = "n" THEN GOTO 880 : ELSE GOTO 840
880 IF FIRSTTIME > 1 THEN KILL "TEMP.TMP"
890 END
900 PRINT : FIRSTTIME = FIRSTTIME + 1
910 IF FIRSTTIME = 2 THEN GOTO 450 : ELSE GOTO 530
930 A$ = TIME$ : FOR I = 1 TO ITER : ZA# = 0# : FOR K = 1 TO 114
940 ZB# = (ZA# + 1#) * (ZA# + 2#) / (ZA# + 3#) + (ZA# + 2#) * (ZA# + 3#) /
(ZA# + 1#) + (ZA# + 3#) * (ZA# + 1#) / (ZA# + 2#)
950 NEXT K : NEXT I : B$ = TIME$ : RETURN
970 A$ = TIME$ : FOR I = 1 TO ITER : ZA# = 1# : FOR K = 1 TO 9
980 ZA# = TAN (ATN (EXP (LOG (SQR (ZA# * ZA#))))) + 1#
990 NEXT K : NEXT I : B$ = TIME$ : RETURN
1000 A$ = TIME$ : FOR I = 1 TO ITER : CALL DBB4  : NEXT I : B$ = TIME$ :
RETURN
1010 A$ = TIME$ : FOR I = 1 TO ITER : CALL DBBCACHE:NEXT I: B$ = TIME$ :
RETURN
1020 A$ = TIME$ : FOR I = 1 TO ITER : CALL DBB16 : NEXT I : B$ = TIME$ :
RETURN
1030 A$ = TIME$ : FOR I = 1 TO ITER : CALL CB4   : NEXT I : B$ = TIME$ :
RETURN
1040 A$ = TIME$ : FOR I = 1 TO ITER : CALL CB16  : NEXT I : B$ = TIME$ :
RETURN
1050 A$ = TIME$ : FOR I = 1 TO ITER : CALL IA    : NEXT I : B$ = TIME$ :
RETURN
1060 A$ = TIME$ : FOR I = 1 TO ITER : CALL XFER  : NEXT I : B$ = TIME$ :
RETURN
1080 IF ITER = 0 THEN A$=TIME$ : B$=A$ : RETURN
1090 OPEN "TEMP.TMP" FOR RANDOM AS #1 LEN=2048
1100 FIELD #1, 2048 AS S$
1110 A$ = TIME$
1120 FOR K = 1 TO ITER : FOR I = 1 TO MAXREADS : GET #1, I : NEXT I : NEXT K
1130 B$ = TIME$ : CLOSE #1 : RETURN
1150 IF ITER = 0 THEN A$=TIME$ : B$=A$ : RETURN
1160 OPEN "TEMP.TMP" FOR RANDOM AS #1 LEN=2048
1170 FIELD #1, 2048 AS S$ : LSET S$ = T$
1180 A$ = TIME$
1190 FOR K = 1 TO ITER : FOR I = 1 TO MAXWRITES : PUT #1, I : NEXT I : NEXT K
1200 B$ = TIME$ : CLOSE #1 : RETURN
```


The data that can fit in your
pocket today can have an
impact on what s into
your pocket tomorrow.
BASF
BASF QUALIMETRIC™ 3.5" MICRO FLEXYDISKS.® CERTIFIED 100% ERROR-FREE AND WARRANTED...
FOREVER

width is degraded by the DMA controller, which runs 7 percent of the time refreshing memory, resulting in an actual bandwidth of 1.10 megabytes per second.

On the other hand, the AT has a 16-bit bus and a cycle time at 6 MHz of 167 ns. Its 80286 normally takes only two cycles to perform a memory operation, but in the AT it was set up with a single wait state, resulting in a total memory-fetch time of 500 ns. The rate of 2 bytes every 500 ns is the same as 4 bytes every microsecond or 4 megabytes per second. In the case of the AT, only 5.3 percent is lost to refresh, resulting in a net bandwidth of 3.78 megabytes per second. Dividing these rates yields a PI for the AT in this department of 3.44, which is close to the 3.27 actually observed for the AT running at 6 MHz in table 1. The experimental difference could be attributed to inaccurate information from IBM on the refresh overhead, loop overhead, or system-background operations, such as timer-interrupt routines. Using the same benchmark on the 9.54-MHz 8086 accelerator running on a PC, I got 4.02, which agrees well with the 4.0 I would expect upon doubling the data-bus width and frequency.

I obtained this benchmark using an assembly language subroutine that did a large number of fast-running 8086 operations. By definition, any operation whose binary machine code takes longer to fetch from memory into the processor than it does to run will end up producing a data-bus-bandwidth-limited situation (i.e., the speed of the sequence is controlled by how fast you can fetch in op codes). The criterion of which operations are data-bus-bandwidth-limited is an easy one, but it is also machine-dependent. The PC takes four cycles to fetch a byte; therefore, any operation whose instruction is a byte long must execute in less than four cycles. Two byte-long instructions must execute in less than eight cycles, etc. The AT takes three cycles to fetch a word, and the criterion for a 2-byte instruction is that it take three cycles to execute to remain under the fetch time. When I apply this reasoning to a cached board with no wait states, I come to the conclusion that very few of the instructions are bus-limited, for I must be able to execute byte-long instructions in one cycle (there are none) and word-long instructions in two cycles. As a result, I rewrote the data-bus-limited benchmark so it would work with no-wait-state 80286 boards. Listing 2 shows the code stream of the original benchmark.

Listing 2's sequence of instructions was repeated to build a 4000-byte-long sequence that was then called as a subroutine. These instructions are all bus-limited for an 8088, 8086, or

(continued)

This is BEFORE LETTRIX.
This is AFTER LETTRIX.

Teach Your Printer Some Manners with

LETTRIX

> "Most users will be on their knees giving thanks for all of the new print features Lettrix provides for less than $100."
>
> *Peter Lewis*
> *The New York Times*

Are you embarrassed to have your dot-matrix print seen in public? With LETTRIX you get well-groomed, letter-quality text directly from your current software. Wiithout buying a new printer. LETTRIX gives your software the social graces of five size options, the flexibility of 21 sharper fonts (including Greek, Cyrillic, and Hebrew), a plethora of formatting options, even the ability to design your own types.
LETTRIX resides in memory and can be used at will, without the pampering needed to make "Font," "Print" or "Paint" programs work. Equally at ease with 1-2-3 or dBase, WordStar or WordPerfect, Multimate or Microsoft Word, LETTRIX mingles comfortably with most DOS software. Best of all, LETTRIX is a pleasure to talk to, since it understands your own software's formatting commands -- and lets you add new ones as well. For instance LETTRIX can add boldfacing, italics, underlining, and the typefaces below to a 1-2-3 spreadsheet or a dBase report. LETTRIX requires an IBM PC, PC Jr., XT, AT or compatible running PC-DOS (2.0 or later) or MS-DOS (2.11 or later) and occupies 60kb of RAM memory.

LETTRIX sharpens dot printing from most software in 20 typefaces:

Roman	Broadway
Park Avenue	WESTERN
SHADOW	Old English
Courier	BLOCK
BANKER	Art Deco
Gothic	Engraved
Español	OUTLINE
Français	Prestige
Русски	Folio
ʽΕλληνικός	ORATOR
עברית	OCR-A

Yes, I want to teach my printer some manners. Please send me ____ copies of LETTRIX at $98.50 each with your ten-day money-back guarantee. My printers are:

- Star
- Panasonic
- AT&T 473, 474
- Okidata (IBM Graphics Configuration)
- Epson MX, RX, FX, LX, JX
- IBM Graphics Printer, Proprinter

I am paying by: • VISA • MasterCard • Check • C.O.D.

Call toll-free:
(800) 351-4500

In Connecticut:
(203) 624-0000

to order or to request our free print sample.

Hammerlab Corporation
938 Chapel Street
New Haven, CT 06510

Hammerlab accepts corporate, institutional, and governmental purchase orders by mail.

Listing 2: *The assembly language code for the data-bus-bandwidth benchmark.*

Op code		Number of bytes	Number of cycles
XOR	AX,AX	2	3
XOR	DX,DX	2	3
ADD	AX,DX	2	3
ADD	AX,DX	2	3
SUB	AX,DX	2	3
SUB	AX,DX	2	3
CLC		1	2
CLD		1	2
NOP		1	3
NOP		1	3

80286 with one wait state but are not bus-limited for a no-wait-state 80286. To reexamine the bus limit in this card, I replaced this 10-byte sequence with a 10-byte sequence of

MOV AX,BX

which obeys the criterion. Then I labeled this benchmark 3b. It takes a cached board at 7.2-MHz two cycles or 277 ns to read in a word. The same process in a PC takes eight cycles or 1680 ns. The ratio of these times is 6.06, which agrees very well with the value of 6.0 for the cache board. In the process of creating this benchmark, I have also learned that only one kind of instruction, interregister moves, is bus-limited in the cached board.

The data-bus-bandwidth benchmark teaches us a lot about 80286 machines. As the data bus goes from one wait state to no wait states, the data-bus activity starts to dry up. Watched on a scope in a six-line loop, the processor performs a flurry of fetches and then goes into an idle state for quite a while as the program works its way back to the bottom of the loop. From a designer's point of view, this means that a no-wait-state board does not exhibit the optimal distribution of data-bus speed to register speed. If memory fetches are done consistently faster than register-bound operations and fast memories cost a lot of money, you would do just as well with slower-speed memories that provide the needed instructions just as the register-bound operations complete. Put in other terms, I think the IBM one-wait-state approach might be near optimal for 80286 machines that are not bogged down with DMA cycles, and the hoopla about no-wait-state memories could very well be a lot of nonsense.

This idea is also highlighted in the *iAPX 286 Hardware Design Manual*, which points out that with no wait states the bus is idle 75 percent of the time, versus 90 percent with one wait state. This document also states that the main beneficiary of no-wait-state memories is block moves. Block moves get used primarily in I/O-intensive applications such as RAM disks and disk caches. It is unlikely that the added speed they represent would make much of an impact on typical single-user applications, although they might make a difference in multiuser operating systems. Unfortunately, when run in protected mode on an 80286, these operating systems will become more register-speed-dependent (a feature of protected mode); that would again argue against ultra-fast memories to improve overall throughput.

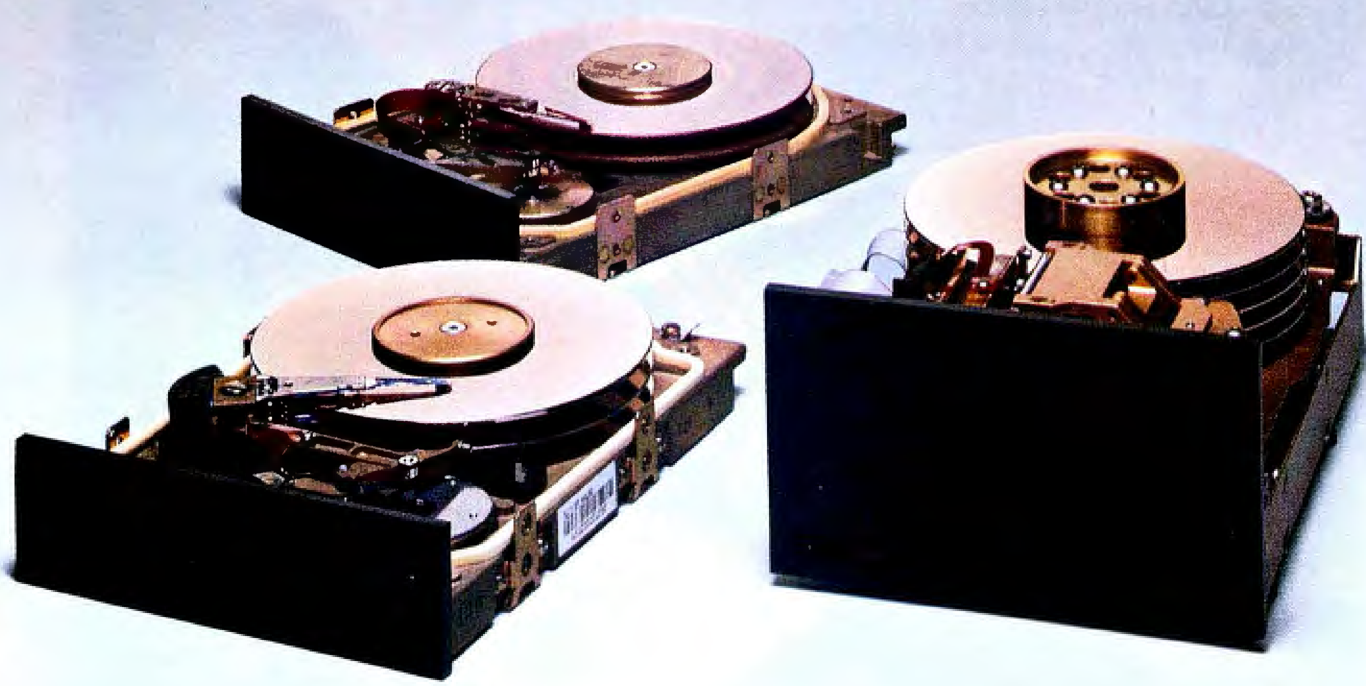
CONCLUSION

Accelerators have been available for the IBM PC for three years. They are a cost-effective method to either speed up or extend the architecture of the computer. Extended architectures range from adding new CPUs to increasing the memory available for MS-DOS 3.0. Three types of accelerators were discussed in this article. Of these, the emulator approach is the most compatible and reliable and the route that provides the largest benefit-to-cost ratio for the average user. Speedup factors of two to three can be obtained using emulators for as little as $500, depending on the amount of 16-bit memory that the board contains. For most users, a $500 accelerator that triples the speed of everything from word processing to FORTRAN applications is more than a good deal, it's addicting; users get mad if they have to put the original 8088 back in.

Accelerators that give more than a 3 to 1 increase in speed also suffer from increased incompatibility and installation problems. I found that going from 9.54 MHz to 12 MHz quadrupled the number of problems I had with emulator installations. If you are contemplating an 8086 emulator that runs faster than 9.54 MHz or an 80286 cached emulator that runs faster than 7.2 MHz, it's a good idea to make sure that the product in question has a provision for running at slower speeds as well. Software-controlled speed is a real benefit at the highest speeds: It is easy to trap the hardware interrupts that control specific devices in the PC and thereby tailor the accelerator to the slow I/O boards or a poor DMA controller on an older motherboard.

The users of accelerators have a tendency to cluster around certain type of products. Individuals running intensive, floating-point-bound problems end up with 8086/8087-based emulators, and these have a tendency to get a reputation for being math-only accelerators, which they are not. Individuals who are doing word processing or program development have recently been heading toward the 80286 cache boards. For I/O-bound applications such as code development, the I/O software utilities that come with an emulator often contribute as much to the speed as the increase in CPU throughput. The factor of two to four that a disk cache adds to hard disk access speed drowns out any difference between the 8086 and the 80286. Finally, there is the support issue. Good support direct from the manufacturer is more critical with accelerators than with any other type of add-in card. ■

Unforgettable.



Disc drives should never forget one thing: your important information.

At Seagate, we build our drives to protect your data against all mishaps. And we put each drive through a series of grueling tests to make sure it doesn't forget.

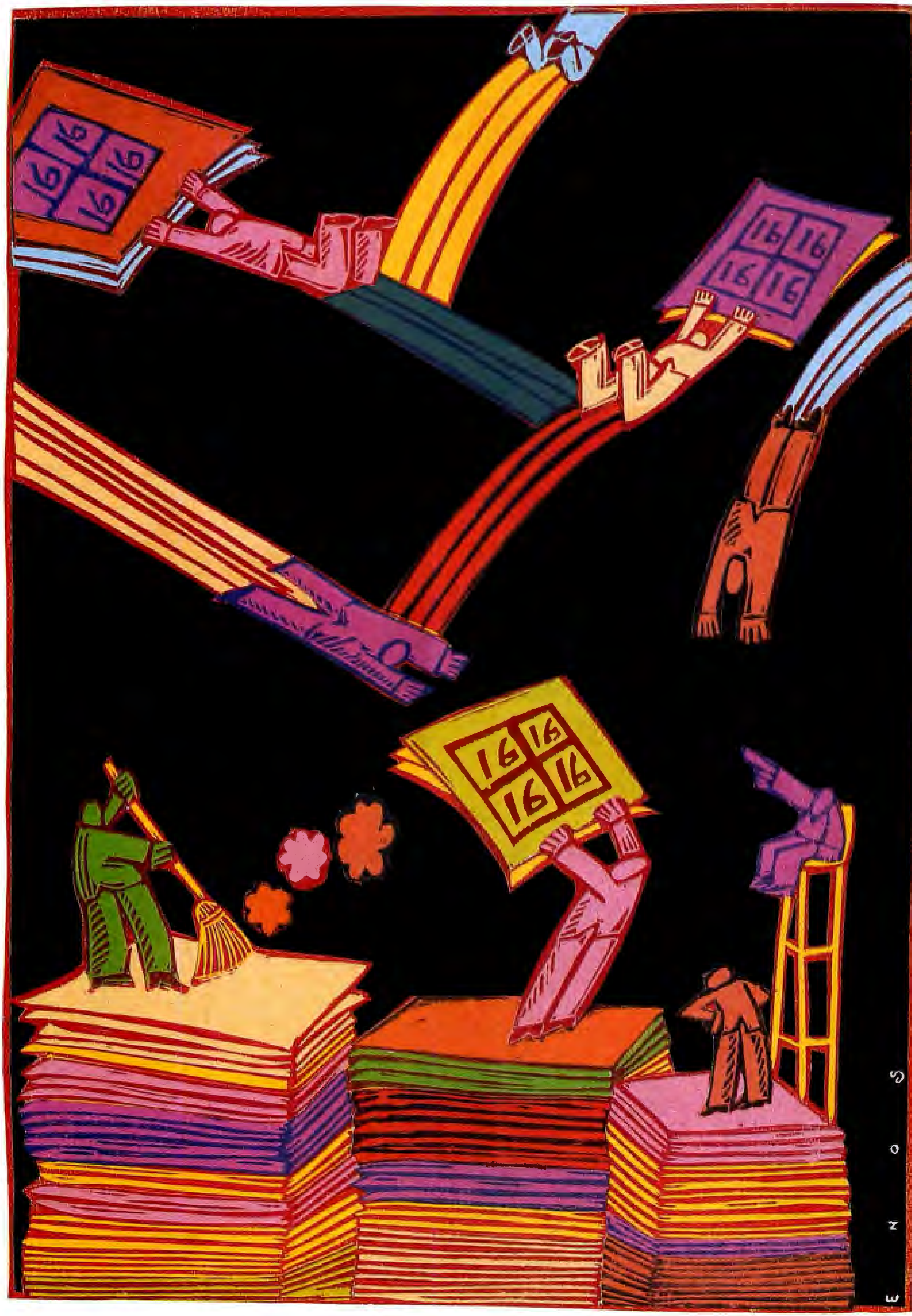
That's why quality-conscious customers have bought more than 3 million of our 5¼″ hard disc drives.

So the next time you're looking for a new computer, disc memory upgrade or add-on, ask for Seagate by name.

It's all you need to remember.

Seagate Technology, 920 Disc Drive, Scotts Valley, CA 95066. 800-468-DISC. In California, 800-468-DISK.

Seagate



ILLUSTRATED BY RANDALL ENOS

LOTUS/INTEL/MICROSOFT EXPANDED MEMORY

This version of bank-switched memory provides a uniform software interface to all applications regardless of hardware differences

BY RAY DUNCAN

THE LOTUS/INTEL/MICROSOFT Expanded Memory Specification (EMS) 3.2 is a functional definition of a bank-switched memory-expansion subsystem made up of hardware expansion modules and a user-installable, resident driver program specific for those modules.

Bank switching is a technique whereby the central processor can make one of many *logical* memory pages available for access in a window at a predetermined physical address. It's somewhat like bringing one card to the top of the deck where you can read it.

Bank-switched memory is hardly a new idea. It has been used on Apple and 8080/Z80 S-100 bus computers since the early days of microcomputing to overcome the addressing limitations of the 8-bit processor. For the IBM PC, Tall Tree Systems has been selling a bank-switched memory board called JRAM with electronic disk and print spooler software for several years. But the LIM EMS version of bank-switched memory is a step ahead: It provides a uniform software interface to all applications, regardless of the differences in the underlying hardware. LIM EMS is not, however, the only memory-expansion subsystem available. See the text box "The AST Enhanced EMS" on page 170.

From a hardware standpoint, EMS allows you to install as much as 8 megabytes of expanded memory in a system. This expanded memory is available to application software in 16K-byte logical pages, which are mapped for access onto one of four 16K-byte physical pages. These physical pages exist in a contiguous 64K-byte area called the *page frame*. You can configure the location of the page frame to avoid conflict with other hardware options, but it always resides above the MS-DOS/PC-DOS main memory area (0K to 640K bytes).

Expanded memory should not be confused with *extended* memory. Extended memory is IBM's term for the memory at physical addresses above 1 megabyte that an 80286 microprocessor can access in protected mode. Current versions of MS-DOS run the 80286 in real mode (8086-emulation mode), where extended memory is not directly accessible.

The EMS installable driver, called the Expanded Memory Manager (EMM), provides the hardware-independent interface between application software and expanded-memory hardware. The EMM comes as an installable character device driver that links into the operating system by adding a DEVICE line to the CONFIG.SYS file.

Internally, the EMM is divided into two major sections: the driver and the manager. The driver mimics some of the actions of a real device driver; it includes initialization and output-status functions and a valid device header. The manager is the true interface between application software and the expanded-memory hardware. The EMM provides several classes of services:

- Status of hardware and software modules
- Allocation of expanded-memory pages
- Mapping logical pages into the physical page frame
- Deallocation of expanded-memory pages
- Support for multitasking operating systems
- Diagnostic routines

Application programs communicate with the EMM directly via software interrupt 67h. MS-DOS is not involved. However, Microsoft Windows (which runs on top of MS-DOS) makes heavy use of expanded memory, if it is present, for program swapping. It seems reasonable to expect that future multitasking versions of MS-DOS and other operating systems for the 8086/88 family may do the same.

USING EXPANDED MEMORY

Expanded memory is a system resource and must be treated like one. Application programs must acquire, use, and release expanded memory in a manner that doesn't cause problems for other programs that are trying to use it.

Before you use expanded memory, you must determine that the EMM is present in the system. There are two approved methods of testing for the existence of the EMM.

The first method is to issue an open request (interrupt 21h, function 3Dh) using the device name of the EMM driver: EMMXXXX0. If the open succeeds, either the drive is present or there is a file on the default drive by that name. To rule out a duplicate file-

(continued)

Ray Duncan is the founder and president of Laboratory Microsystems Inc. (P. O. Box 10430, Marina del Rey, CA 90295). An M.D., he is also a columnist for Dr. Dobb's Journal *and the author of* Advanced MS-DOS *(Microsoft Press).*

THE AST ENHANCED EMS

AST Research has also formulated and announced a bank-switched memory subsystem specification. The Enhanced Expanded Memory Specification (EEMS) is upward-compatible with LIM EMS version 3.2 but technically more complex.

EEMS differs from EMS in two important ways:

- In EMS, the 64K-byte page frame is located *above* the 640K-byte MS-DOS area and is used to map four different 16K-byte logical pages into physical memory simultaneously. In EEMS, as many as 64 logical pages can theoretically be mapped into physical memory at once. Practically speaking, on a typical IBM PC under MS-DOS, about 15 logical pages simultaneously mapped into memory is probably the upper limit. (This is the memory above the 640K-byte MS-DOS area minus the addresses occupied by system ROMs and video buffers.)
- In EMS, logical pages cannot be bank-swapped into the 640K-byte memory area controlled by MS-DOS, making it difficult to use expanded memory for paging executable code. In EEMS, logical pages can be mapped into physical memory anywhere in the microprocessor's address space, making it possible for specially designed multitasking operating systems to use expanded memory for very fast switching between processes.

AST's EEMS redefines the software interfaces between the EMM and the application program (see table 2) only slightly. It extends function 5 (Map Memory) to allow physical page numbers larger than 3 and adds function 33 to allow applications to obtain a list of all the physical page addresses. For upward compatibility with EMS, EEMS requires that the first four physical 16K-byte pages be contiguous.

Although EEMS is more sophisticated than EMS and has considerably more potential if properly used by an operating system designed to exploit all its features, to date it has received little support from other PC software and hardware developers. Currently, the only commonly available programs that use EEMS are Quarterdeck Systems' DESQview and Digital Research's Concurrent DOS XM. The only other manufacturer of EEMS hardware, Quadram, has reported delays in designing a board that is simultaneously EMS- and EEMS-compatible.

name on another drive, you issue a get-output-status request via the IOCTL (input/output control) function (interrupt 21h, function 44h); the status returned in register AL will be 0FFh if the driver is present and 0h if not. Either way, you should then close (interrupt 21h, function 3Eh) the handle obtained from the open so the system can reuse it. (See listing 1.)

The second method is to use the address found in the vector for interrupt 67h to inspect the device header of the presumed EMM. If the EMM is present, the name field at offset 0Ah of the device header will contain the string EMMXXXX0. This approach is nearly foolproof and avoids the relatively high overhead of an open. However, it is somewhat less "well behaved" since it involves inspecting memory that does not belong to the application. (See listing 2.)

After establishing that the EMM is present, a typical application follows this general sequence to use expanded-memory resources:

1. Check the operational status of the EMM (EMS function 1).
2. Check the version number of the EMM (EMS function 7) to make sure that all the services that the application needs are available.
3. Obtain the segment of the page frame that the EMM uses (EMS function 2).
4. Allocate the desired number of expanded-memory pages (EMS function 4). If the allocation is successful, the EMM returns a *handle* that the application then uses to refer to the expanded-memory pages that it "owns." This is analogous to opening a file and using a handle for any

(continued)

For People Who'd Rather Switch than Fight!

The ultimate printer-sharing switch has arrived...

Here's the Logical Connection™!

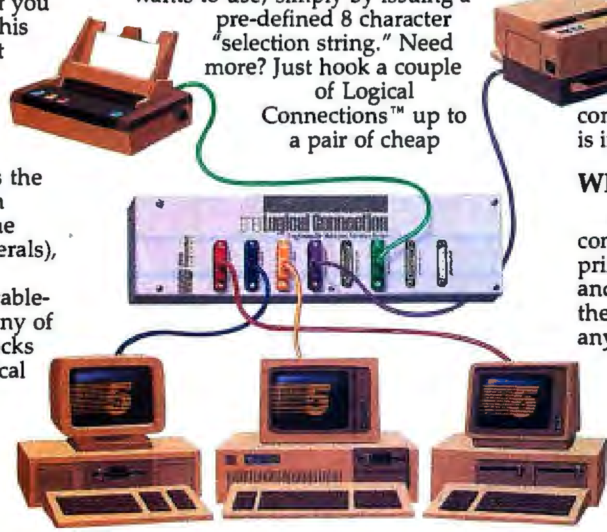
It happens all the time: your entire office staff is having a "get-together" by the printer. And if you haven't found the solution to this bottleneck yet, it's time you got the connection: The Logical Connection.™

No more slow-downs.

The Logical Connection™ is the ultimate printer-sharing switch because it lets *everyone* share the same printers (or other peripherals), at the same time! With no downloading, disk-shuffling, cable-swapping, switch-flipping or any of the other thousand natural shocks that sharing's heir to. The Logical Connection™'s big, smart buffer (256K, 512K model also available) will spool each document for sequential printing, so your computers can keep right on working. But wait, there's more . . .

Long-distance switching.

What if your printers and computers are strung out all over the building? No problem, because you can "daisy-chain" up to 45 Logical Connections™ together. Over a single twisted-pair wire like telephone wire (which, by the way, you probably already have *wired* all over the building). And each computer can choose which printer it wants to use, simply by issuing a pre-defined 8 character "selection string." Need more? Just hook a couple of Logical Connections™ up to a pair of cheap modems and let a roomful of computers talk to a roomful of printers across the country . . . or across the world.



Automatic Data Conversion

If you want to run a parallel daisy-wheel *and* a serial laser printer from the same computer port, no problem. And if two devices have different handshaking, baud rates, word lengths, or parities, The Logical Connection™ will do all the conversions for you *automatically*, whenever they're needed. And it will never forget your configurations, because its memory is in battery-powered RAM.

Why fight it?

So if you'd like to give *all* your computers instant access to *all* your printers (and plotters and modems and whatever else you'd like to hook them up to), *all* the time, why fight it any longer? Make the switch. Get The Logical Connection™ at your leading computer dealer today, or order direct from Fifth Generation Systems.

the Logical Connection
Programmable Multi-port Interface Device

$495
Suggested retail price

Fifth Generation SYSTEMS
Innovative Products Using Today's Technology

7942 Picardy Avenue, Suite B350, Baton Rouge, Louisiana 70809 • Sales: 909 Electric Avenue, Suite 308, Seal Beach, CA 90740 • (213) 493-4483

Inquiry 96 for End-Users. Inquiry 97 for DEALERS ONLY.

Listing 1: *Testing for the* EMM *via the* MS-DOS open *and* IOCTL *functions.*

```
              mov   ah,3dh              ;function 3Dh = open.
              mov   al,0                ;mode = read-only.
              mov   dx,seg emm_name     ;DS:DX = addr of name
              mov   ds,dx               ;of Expanded Memory Manager.
              mov   dx,offset emm_name
              int   21h                 ;transfer to MS-DOS.
              jc    emm_absent          ;open failed, driver absent
                                        ;or no more handles.

                                        ;open succeeded, make sure
                                        ;it was not a file.
              mov   bx,ax               ;BX = handle for EMM.
              mov   ah,44h              ;function 44h = IOCTL.
              mov   al,7                ;subfxn 7 = get outp. stat.
              mov   cx,0                ;CX = bytes to read.
              mov   dx,offset emm_buff  ;DS:DX = dummy buffer addr.
              int   21h                 ;transfer to MS-DOS.
              push  ax                  ;save code from IOCTL.
                                        ;close file or device
                                        ;to reclaim handle.
              mov   ah,3eh              ;function 3Eh = close.
              int   21h                 ;BX still contains handle.
              jc    close_failed        ;jump if error on close.
              pop   ax                  ;now look at status code
              or    al,al               ;returned from IOCTL.
              jz    emm_absent          ;if status = "not ready"
                                        ;then it was a file, not
                                        ;the expected EMM driver.
              .
emm_present:  .                         ;driver is present.
              .
emm_absent:   .                         ;driver absent, or no
              .                         ;more handles available.
              .
close_failed: .                         ;error on close.
              .

emm_name      db    'EMMXXXX0',0        ;device name for Expanded
                                        ;Memory Manager

emm_buff      dw    0                   ;dummy buffer for IOCTL call
```

Listing 2: *Testing for the* EMM *by inspecting the name field in the driver's device header.*

```
emm_int       equ   67h                 ;Expanded Memory Manager
                                        ;interrupt vector.
              .
              .
              .
              mov   ah,35h              ;DOS function 35h =
              mov   al,emm_int          ;get interrupt vector
              int   21h                 ;into ES:BX.
                                        ;assume ES:0000 points
                                        ;to the base of the EMM.
              mov   di,10               ;ES:DI = addr of name
                                        ;field in Device Header.
              mov   si,seg emm_name     ;let DS:SI = addr of
              mov   ds,si               ;guaranteed driver name
              mov   si,offset emm_name  ;for EMM.
              mov   cx,8                ;length of name field.
```

```
                cld
                repz cmpsb              ;compare strings.
                jnz  emm_absent         ;strings didn't match.
                .
                .
emm_present:    .                       ;driver present.
                .
emm_absent:     .                       ;driver absent.
                .

emm_name        db    'EMMXXXX0',0      ;device name for Expanded
                                        ;Memory Manager.
```

Listing 3: *The sector mapping routine from the* EMSDISK.ASM *program. This is an example of a program mapping its internal records onto the* EMS 16K-*byte logical pages.*

```
;
; Map into memory a logical "disk" sector from the EMS
; pages allocated to the EMSDISK.
;
; Called with:  AX    = logical sector number
;
; Returns:      CY    = clear if no error
;               AX    = offset within EMS page frame
;               AX,CX,DX destroyed
;
;               CY    = set if EMM mapping error
;               AX,CX,DX destroyed
;
map_sec proc    near

        mov     dx,0            ; divide sector no. by sectors
        mov     cx,sec_per_page ; per page, to get EMS page number
        div     cx              ; now AX=EMS page,DX=rel. sector
        push    dx              ; save remainder
        mov     bx,ax           ; BX <- EMS page number
        mov     ax,4400h        ; map function, phys. page=0
        mov     dx,emm_handle   ; process ID for EMSDISK
        int     67h

        or      ah,ah           ; if EMM error, jump to return flag
        jnz     map_sec1

        pop     ax              ; get remainder (relative sector)
        mov     cx,sec_size     ; remainder*sec_size to get offset
        mul     cx              ; into EMS logical page
        clc                     ; return CY=clear for no error
        ret                     ; back to caller

map_sec1:                       ; come here if EMM mapping error
        add     sp,2            ; clear stack
        stc                     ; return CY=set for error
        ret

map_sec endp
;
```

subsequent read or write operations on it.

5. If the requested number of pages is not available, the application can query the EMM to find out how many pages are available (EMS function 3). Then the application can determine if it can continue processing in a degraded fashion.

6. Perform mapping requests (EMS function 5) as needed to gain access through the page frame to the expanded-memory logical pages that the application owns. Any further

(continued)

blocking, deblocking, or imposing a record structure on the expanded memory must be done at the application level. (The routine map__sec, which is extracted from EMSDISK.ASM, is an example of such mapping. See listing 3.)

7. Deallocate the expanded-memory pages and release the EMM handle obtained earlier (EMS function 6) when it has finished using them. If the application doesn't perform this housekeeping, those pages will be unavailable to other programs until the system is rebooted.

Application programs communicate with the EMM directly via interrupt 67h. The basic calling sequence for the EMM is shown in listing 4. In general, registers ES:DI are used to pass the address of a buffer or an array, and register DX is used to hold an EMM handle. Upon return from an EMM function call, register AH will contain zeros if the function was successful or an error code with the most significant bit set if the function failed (see table 1). Other values are typically returned in registers AL and BX or in a user-specified buffer. Table 2 contains a summary of the services available from the EMM.

An interrupt handler or a resident driver that uses EMS follows the same general procedure that an application does, with a few minor variations. It may need to acquire an EMM handle and allocate pages before the operating system is fully functional. Thus, since it may or may not have MS-DOS file open, close, and IOCTL functions available, it must use a modified version of the get-interrupt-vector technique to test for the existence of the EMM.

A resident driver or an interrupt handler typically owns its expanded-memory pages on a permanent basis (until the system is rebooted) and never deallocates them. It must also save and restore the EMM's page mapping context (EMS functions 8 and 9, respectively), so it won't disturb a foreground program's use of EMS.

The EMM relies heavily on the good behavior of other software to avoid the corruption of expanded memory. If several applications that use expanded memory run under a multitasking manager, such as Windows or TopView, and one of them does not abide strictly by the EMM's conventions, the data of some or all of those applications may be destroyed.

Listing 4: *The calling sequence for the* EMM.

```
mov   ah,function          ;AH always determines service type
.
.                          ;load other registers with values
.                          ;specific to the EMS function
.
int   67h
```

Table 1: EMM *error codes. After a call to the* EMM, *register* AH *will contain a zero if the function was successful or an error code in the range* 80h–8Fh.

Error Code	Significance
00	Function was successful.
80h	Internal error in the EMM software.
81h	Malfunction in the expanded-memory hardware.
82h	Memory manager is busy.
83h	Invalid handle.
84h	Function requested by the application is not defined.
85h	No more EMM handles available.
86h	Error in save or restore of mapping context.
87h	An allocation request specified more logical pages than are physically available in the system; no pages were allocated.
88h	An allocation request specified more logical pages than are currently available in the system (the request does not exceed the physical pages that exist, but some are already allocated to other handles); no pages were allocated.
89h	You cannot allocate zero pages.
8Ah	The logical page that was requested is outside the range of logical pages allocated to the handle.
8Bh	Illegal physical page number in mapping request (not in the range 0 to 3).
8Ch	The save area for mapping contexts is full.
8Dh	Save of mapping context failed because save area already contains a context associated with the requested handle.
8Eh	Restore of mapping context failed because save area does not contain a context for the requested handle.
8Fh	Subfunction parameter not defined.

THE EMSDISK PROGRAM

EMSDISK is an installable RAM disk driver for MS-DOS 2.0 or higher that uses expanded memory and the EMM services. Written in assembly language, it can emulate a fixed disk as large as 8 megabytes and store the files in expanded-memory pages for extremely fast access. It was tested on a Compaq running PC-DOS 3.1, using the Intel Above Board and its EMM version 3.3. [*Editor's note:* EMSDISK.ASM *is available on disk, in print, and on* BIX. *See the insert card following page* 176 *for details. Listings are also available on* BYTE*net. See page* 4.]

Don't confuse the EMSDISK driver with the EMM driver that is provided with the expanded-memory board. The EMM is a proprietary software product of the board manufacturer. EMSDISK requires the EMM in order to operate, but the only resemblance between the two programs is that they are both user-installable device drivers. (To implement EMSDISK on your system, see the text box "Putting EMSDISK to Work" below.)

EMSDISK follows the structure required of an installable block device driver under MS-DOS. For more information about the structure or programming of installable drivers, see the bibliography at the end of this article.

The EMSDISK driver contains two critical data structures: the device header and the BIOS parameter block (BPB). The device header contains linkages to the driver's routines, a device-attribute word that describes its nature and capabilities, and a link to the next installed driver in the chain. The BPB is supplied by block device drivers for use by MS-DOS and describes the size and location of a disk's control and data areas.

EMSDISK breaks down logically into two major modules: the initialization module and the driver proper. MS-DOS calls the initialization module when it first loads EMSDISK. This module is responsible for determining whether the EMM is present, checking its version, getting the segment address of the EMM's page frame, allocating memory pages for EMSDISK, setting up the BPB, printing the program's identification message, and notifying the operating system of the amount of memory the program occupies and the number of logical disk units it supports. The initialization routine is used only once, and the

(continued)

PUTTING EMSDISK TO WORK

To assemble the source code into a usable EMSDISK driver module, you need the Microsoft Macro Assembler (MASM.EXE), the Microsoft Linker (LINK.EXE), and the EXE-to-BIN conversion utility (EXE2BIN.EXE). Put these three programs and the source file, EMSDISK.ASM, on a working disk, then enter the following sequence of instructions:

```
MASM EMSDISK;
LINK EMSDISK;
EXE2BIN EMSDISK.EXE
  EMSDISK.BIN
DEL EMSDISK.EXE
```

Now copy the resulting EMSDISK.BIN file to the root directory of your system boot disk. Finally, to link the EMSDISK driver into the operating system, you need to add the line DEVICE= EMSDISK.BIN to the CONFIG.SYS file on your system boot disk. This line must occur after the line that loads the EMM (DEVICE=EMM.SYS in the case of the Intel Above Board).

If you enter the DEVICE line as given, all the available expanded-memory pages will be allocated to the EMSDISK. If you want a smaller size EMSDISK, enter the line DEVICE= EMSDISK.BIN *nnn*K. For example, to create an EMSDISK of 512K bytes, you would enter DEVICE= EMSDISK.BIN 512K in your CONFIG.SYS file. If you omit the *nnn*K parameter, or specify more expanded memory than actually exists or is available, the EMSDISK driver will simply allocate all of the available expanded-memory pages.

When the system is booting, EMSDISK will display a message on the screen to let you know that it has installed successfully, what drive letter is assigned to it, and how much memory is allocated to it. The drive letter that is assigned to the EMSDISK depends on the number of other block devices in your system and the position of the DEVICE=EMSDISK.BIN line in the CONFIG.SYS file. Under DOS 3.x, the drive letter is made available to the EMSDISK driver in the initialization process and is printed out in the driver identification message; under DOS 2.x, you have to figure out which drive letter it is.

Finally, you can create multiple EMSDISK drives by placing multiple DEVICE=EMSDISK.BIN *nnn*K lines in your CONFIG.SYS file, allocating a portion of the available expanded memory on each line. Realize, however, that if you omit the *nnn*K parameter, the first DEVICE=EMSDISK.BIN line will allocate all the available expanded memory, and none will be left for the others.

Table 2: A *summary of software interfaces to application programs provided by the* EMM.

Function Name	Action	Called With	Returns	Comments
Function 1: Get Manager Status	Tests whether the expanded-memory software and hardware are functional.	AH = 40h	AH = status	This call is used after the program has established that the EMM is present, using one of the techniques presented in listings 1 and 2.
Function 2: Get Page Frame Segment	Obtains the segment address of the EMM page frame.	AH = 41h	AH = status BX = segment of page frame, if AH = 0	The page frame is divided into four 16K-byte pages, which are used to map logical expanded-memory pages into the physical memory space of the 8086/88.
Function 3: Get Number of Pages	Obtains the total number of logical expanded-memory pages present in the system and the number not already allocated.	AH = 42h	AH = status BX = unallocated EMS pages, if AH = 0 DX = total EMS pages in system	The application need not have already acquired an EMM handle to use this function.
Function 4: Get Handle and Allocate Memory	Obtains an EMM handle and allocates logical EMS pages to be controlled by that handle.	AH = 43h BX = logical pages to allocate	AH = status DX = handle, if AH = 0	Equivalent of a file open function for the EMM. The handle that is returned is analogous to a file handle and "owns" a certain number of EMM pages. The handle must be used with every subsequent request to map memory and must be released by a close operation when the application is finished. The function may fail because available EMM handles or EMS pages have been exhausted.
Function 5: Map Memory	Maps one of the logical pages of expanded memory assigned to a handle onto one of the four physical pages within the EMM's page frame.	AH = 44h AL = physical page (0 to 3) BX = logical page (0. . .[*n*−1]) DX = handle	AH = status	The logical page number must be in the range (0. . .[*n*−1]), where *n* is the number of logical pages previously allocated to the EMM handle with function 4. To access the memory once it has been mapped to a physical page, the application also needs the segment of the EMM's page frame, obtained with function 2.
Function 6: Release Handle and Memory	Deallocates the logical pages of expanded memory currently assigned to a handle and then releases the handle for reuse.	AH = 45h DX = EMM handle	AH = status	This is the equivalent of a close operation on a file. It notifies the EMM that the application will not be making further use of the data it may have stored within expanded-memory pages.
Function 7: Get EMM Version	Returns the version number of the EMM software.	AH = 46h	AH = status AL = EMM version, if AH = 0	The returned value is the version of the EMS that the EMM driver complies with. It is returned encoded as BCD, with the integer part in the upper 4 bits and the fractional part in the lower 4 bits.
Function 8: Save Mapping Context	Saves the contents of the expanded-memory page mapping registers on the expanded-memory boards, associating those contents with a specific EMM handle.	AH = 47h DX = handle	AH = status	This function is designed for use by interrupt handlers and resident drivers or utilities that must access expanded memory. The handle supplied to the function is the handle that was assigned to the interrupt handler during its initialization sequence, not the one assigned to the program that was interrupted.

LISTINGS YOUR WAY!

It's easy. Now there are three ways to get source-code listings that go with BYTE articles—on disk, in print or on line.

LISTINGS ON DISK

BYTE listings on disk are right for you if you want to compile or run programs quickly. Disks are now available by annual subscription as well as individual orders. BYTE offers listings in 11 popular disk formats. See the order card below.

LISTINGS IN PRINT

This print supplement contains source-code listings described in the issues of BYTE. See order card below.

LISTINGS FROM BIX

(BYTE Information Exchange) By joining BIX, you can download BYTE source-code listings for no additional connect-time charges. See BIX informational ad listed in Reader Service index in this issue.

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS MAIL PERMIT NO. 10 PETERBOROUGH, NH

POSTAGE WILL BE PAID BY ADDRESSEE

BYTE Listings
One Phoenix Mill Lane
Peterborough, NH 03458-9990

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS MAIL PERMIT NO. 10 PETERBOROUGH, NH

POSTAGE WILL BE PAID BY ADDRESSEE

BYTE Listings
One Phoenix Mill Lane
Peterborough, NH 03458-9990

FOR DIRECT ORDERING CALL TOLL FREE: 800-258-5485

Call: M–F, 8:30 a.m. to 4:30 p.m. Eastern Time

(603-924-9281 for New Hampshire residents)

For credit card orders only.

Function Name	Action	Called With	Returns	Comments
Function 9: Restore Mapping Context	Restores the contents of all expanded-memory hardware page mapping registers to the values associated with the given handle.	AH = 48h DX = EMM handle	AH = status	Use of this function must be balanced with a previous call to EMS function 8. It allows an interrupt handler or resident driver that used expanded memory to restore the mapping context to its state at the point of interrupt.
Function 10 (0Ah): Reserved				This function was defined in the EMS 3.0 specification, but it is no longer documented in EMS version 3.2.
Function 11 (0Bh): Reserved				This function was defined in the EMS 3.0 specification, but it is no longer documented in EMS version 3.2.
Function 12 (0Ch): Get Number of EMM Handles	Returns the number of active EMM handles.	AH = 4Bh	AH = status BX = number of EMM handles, if AH = 0	If the returned number of handles is zero, none of the expanded memory is in use. The number of active EMM handles will never exceed 255.
Function 13 (0Dh): Get Pages Owned by Handle	Returns the number of logical expanded-memory pages allocated to a specific handle.	AH = 4Ch DX = EMM handle	AH = status BX = logical pages, if AH = 0	A single program can make several allocation requests and therefore "own" several EMM handles. The number of pages returned will always be in the range of 1 to 512—never zero if the function is successful.
Function 14 (0Eh): Get Pages for All Handles	Returns an array that contains all the active handles and the number of logical expanded-memory pages associated with each handle.	AH = 4Dh DI = offset of array to receive information ES = array segment	AH = status BX = number of active EMM handles If AH = 0, array is filled in as described in the Comments column.	The array is filled in with 2-word entries. The first word of each entry contains a handle; the next word, the number of pages associated with that handle. The value returned in BX gives the number of valid 2-word entries in the array. Since the maximum number of active EMM handles is 255, the array need not be larger than 1024 bytes.
Function 15 (0Fh): Get/Set Page Map	Saves or sets the contents of the EMS page mapping registers on the expanded-memory boards.	AH = 4Eh AL = subfunction number DS:SI = array holding mapping information (subfunctions 1 and 2) ES:DI = array to receive information (subfunctions 0 and 2)	AH = status AL = bytes in page mapping array (subfunction 3) Array pointed to by ES:DI receives mapping information for subfunctions 0 and 2.	Subfunctions: 0 = get mapping registers into array 1 = set mapping registers from array 2 = get and set mapping registers in one operation 3 = return needed size of page mapping array This function was added in EMS version 3.2 and is designed to support multitasking operating systems. It should not ordinarily be used by application software. The array contains hardware- and EMM-software-dependent information. It may contain other information that is necessary to restore the expanded-memory subsystem to its previous state, as well as the contents of the mapping registers themselves.

Listing 5: *An extract from the initialization module of the* EMSDISK *program.*

```
init2:  xor     ax,ax               ; if EMM driver present,
        mov     es,ax               ; address in vector
        mov     bx,emm_int*4        ; points to EMM driver.
        mov     es,es:[bx+2]        ; ES:0000 points to EMM hdr,
        mov     di,10               ; ES:DI = addr of device
                                    ; name field, DS:SI = addr
                                    ; of EMM driver name
        mov     si,offset emm_name
        mov     cx,8                ; length of device name fld
        cld
        repz    cmpsb               ; if EMM name matches
        jz      init3               ; driver header, jump.
        mov     dx,offset msg1      ; if strings didn't match,
        jmp     init_err            ; driver is absent, exit.

init3:  mov     ah,40h              ; EMM driver is present,
        int     67h                 ; test EM Manager status.
        or      ah,ah
        jz      init4               ; jump, driver is OK
        mov     dx,offset msg2      ; EMM non-functional, print
        jmp     init_err            ; error message and exit

init4:  mov     ah,46h              ; check EM Manager version
        int     67h
        or      ah,ah
        jz      init5               ; jump, got version ok
init45: mov     dx,offset msg3      ; print EMM error message
        jmp     init_err            ; exit, discarding driver

init5:  cmp     al,030h             ; check at least ver. 3.0
        jae     init6               ; jump if EMM version OK
        mov     dx,offset msg6
        jmp     init_err            ; EMM version too low, exit

init6:  mov     ah,41h              ; get page frame segment
        int     67h
        or      ah,ah
        jnz     init45              ; jump if didn't get frame
        mov     page_frame,bx       ; save segment of page frame

        mov     ah,42h              ; get number of pages avail
        int     67h
        or      ah,ah
        jnz     init45              ; jump if error on get pages
        mov     total_pages,dx      ; save total EMM pages
        mov     avail_pages,bx      ; save available EMM pages
        or      bx,bx
        jnz     init7               ; proceed if some pages
                                    ; available
        mov     dx,offset msg4
        jmp     init_err            ; no EMS pages left, exit

init7:  call    get_kb              ; convert desired size
                                    ; of EMSDISK
        mov     ah,43h              ; try and allocate EMM pages
        mov     bx,owned_pages
        int     67h                 ; if allocation is
        or      ah,ah
        jz      init8               ; successful, jump
        mov     dx,offset msg5      ; if it failed, print
        jmp     init_err            ; error message and exit

init8:  mov     emm_handle,dx       ; save EMM handle
                                    ; for this driver
```

(continued)

```
         call    setup              ; set up BPB

         call    format             ; format the EMSDISK
         jnc     init9              ; jump if no format error
         mov     dx,offset msg7     ; formatting error, exit
         jmp     init_err

init9:   call    signon             ; format and display driver
         .                          ; identification
         .
         .
```

memory it occupies is discarded afterward. If it can't find the EMM or no pages are available, it aborts the installation of the entire program so that no system memory is wasted. (See listing 5.)

The driver proper is composed mainly of the read and write subroutines that transfer logical sectors to and from expanded-memory pages. Two other minor subroutines, media_chk and build_bpb, provide MS-DOS with information about the RAM disk's logical structure.

EMSDISK contains a fair amount of code that is not strictly necessary for its operation but is provided as an example of good behavior. This includes checking the DOS and EMM versions, testing the maximum command code passed to the driver in the request header, discarding the EMSDISK driver if the EMM cannot be found or if inadequate expanded memory is available, and using a local stack (to avoid overflowing the DOS stack) in the interrupt routine. On the other hand, I have omitted many optimizations from EMSDISK that could increase performance, such as conserving page mappings when multisector transfers are requested by DOS.

There are three values in the EMSDISK program that you may wish to fine-tune to change its behavior. Two of these are equates at the beginning of the program: sec_size, which sets the logical sector size (I have used 512 bytes for symmetry with other IBM media), and dir_size, which specifies the maximum number of entries in EMSDISK's root directory. The sector size should be an even multiple of 512 bytes (512, 1024, 2048, . . . , 16384). If you modify sec_size, you will need to set the non-IBM-format bit in the device-attribute word of the device header.

The third value that you can alter is the number of sectors per cluster. This value is calculated in the setup subroutine according to the number of logical sectors that will fit into the allocated expanded-memory pages. Raising the number of sectors per cluster trades wasted disk space for a faster access speed. On average, one-half cluster per file is wasted space on the disk.

A Lasting Effect

The RAM disk program EMSDISK is only one way in which you can use expanded memory. Several of the most popular spreadsheet and integrated data management programs have been released in new versions to take advantage of the additional fast storage that is provided by expanded memory. ■

BIBLIOGRAPHY

Angermeyer, John, and Kevin Jaeger. *MS-DOS Developer's Guide.* Indianapolis, IN: The Waite Group, Howard K. Sams and Co., 1986.

The AST Enhanced Expanded Memory Specification. Product Marketing, AST Research, 2121 Alton Ave., Irvine, CA 92714. Hotline number: (714) 863- 1333, ext. 5906.

Duncan, Ray. *Advanced MS-DOS.* Redmond, WA: Microsoft Press, 1986.

The Lotus/Intel/Microsoft Expanded Memory Specification, original document (part no. 300275-003, dated September 1985). Intel Corp., 3065 Bowers Ave., Santa Clara, CA 95051. Hotline number for software developers: (800) 538-3373.

Microsoft MS-DOS Programmer's Reference. Document 410630014-320-003-1285. Redmond, WA: Microsoft Corp., 1986.

Inquiry 16

Performance Programming

Developing a single program for a variety of IBM PC compatibles

BY JOEL ROSENBLUM AND DAN JACOBS

IF YOU CREATE programs for the IBM PC, you often need to sell separate versions of your software for particular "compatible" machines. The wide variety of so-called PC compatibles creates problems for you as a software engineer trying to write one version of your program that operates on as many machines as possible.

As many buyers have already discovered, some IBM PC clones are not true clones at all. These machines either are enhanced with features that give them a marketing edge over the original IBM machine or are so restricted that they have trouble using common existing software such as Lotus 1-2-3 and dBASE III. All this hardware incompatibility makes it more difficult to design and sell your programs to everyone who can use them. Your customers send you back to the salt mines to develop 16 different versions of the same software product.

Defining Performance Programming

We have developed a methodology that lets you design "generic" software; it will operate with most IBM-compatible machines that use MS-DOS.

Best of all, our procedures will let your programs automatically take advantage of PC hardware features. Your programs will know when they can write directly to video memory or access the IBM BIOS low-memory variables. If your program encounters a machine it doesn't know, it can decide whether certain features exist for it to use or whether it should treat the new machine as generic until you can incorporate that machine's features into the next release of your software. We will also describe some methods that allow user-selectable compatibility so your customers won't have to wait for your next software release. Now you can sell only one version of your program. However, that version is special because you create it using a method we call performance programming.

We will explain how to program for performance while maintaining compatibility. Our approach is twofold. First, we define a generic PC as a model on which to base your program. Then we define and use a features table that automatically determines IBM PC compatibility by reading the manufacturer's copyright notice contained in the machine's BIOS ROMs. We next look at how you can use the presence of other equipment within your machine to set "performance bits" and how you can incorporate that information into your overall program design. Finally, we will give some examples relating to processor detection and speed determination, video capability, extended character set presence, ANSI cursor addressing, and use of other set bits in your program.

Defining the Generic PC

We define a generic PC as a computer that is only minimally compatible with the IBM PC. The requirements of our theoretical generic PC design are the presence of an Intel 8088 or equivalent processor, the ability to read/write either IBM PC 5¼-inch or 3½-inch disks, MS-DOS, the ability to

(continued)

Joel Rosenblum (609 East Speer Blvd., Denver, CO 80203) is president of Data Pacific Inc. and codeveloper of the Apple Macintosh emulator for the Atari ST. Dan Jacobs is a member of the technical staff at Quest Analysis (11965 Venice Blvd., #305, Los Angeles, CA 90066).

ILLUSTRATED BY ALEXANDRA BOIES

display 25 lines of either 40- or 80-column text, and an ASCII keyboard without cursor and function keys.

Our generic PC does not include the following features present in a true IBM PC: an IBM extended character set, direct cursor-key positioning, bit-mapped screen graphics, the ability to use BIOS calls and corresponding access to low-memory variables, and the ability to access hardware directly.

We assume that a generic PC has a text-only screen display that supports only the standard ASCII character set and has a keyboard without cursor-positioning or function keys. You might note that the Zenith Z-100 is close to qualifying as our theoretical generic PC. Also, the generic system does not have an IBM- compatible BIOS even though it supports MS-DOS. This implies that all operating system function calls for a generic PC must occur at the highest level, which in this case is by using DOS system interrupt calls 20h through 27h. More information about these OS interrupt calls appears in IBM's *Disk Operating System Technical Reference*, available through IBM dealers or directly from IBM (write IBM Technical Directory, P.O. Box 2009, Racine, WI 53404-3336 or call (800) 426-7282).

The advantages of generic PC software are threefold. From your standpoint as a programmer, your software will work on most IBM PCs and compatibles. As a software publisher and dealer, you have to stock and sell only one version of a program. Finally, as a user, you do not need to be concerned about program compatibility or special steps to install the software on your IBM PC–compatible hardware.

On the other hand, generic software cannot take advantage of the advanced features present in the original IBM PC and some compatibles. Software with only scrolling generic screens and no graphics capability is not acceptable for some users. Here the second part of our methodology, the features table, comes into play.

DEFINING A FEATURES TABLE

A features table lets your programs take advantage of a particular compatible's features by determining the equipment your program is operating on. Your software automatically enhances itself for certain compatibles, so your program runs faster and more efficiently and has an improved user interface.

If your program is to operate with this type of intelligence, it must reliably determine the equipment. Your software can't read the label on the outside of your compatible, so we rely primarily on the copyright notice contained inside the machine's BIOS ROMs.

When an engineer designs a PC compatible, he or she incorporates into that machine's ROMs a set of routines that directly control the hardware. Those routines are the BIOS. A manufacturer of PC compatibles can write his or her own BIOS or purchase a BIOS from another manufacturer such as Phoenix Software Associates or Faraday Electronics.

The manufacturer can also market a machine that is made by an OEM. The OEM usually brands its machines by placing a copyright notice within the BIOS code. An example is the AT&T PC 6300, which is manufactured by Olivetti. Olivetti purchases the BIOS for its machine from Phoenix Software Associates. So what is the resulting copyright notice inside the user's 6300? It is the string "OLIVETTI," followed by some text.

Copyright notices thus far have been an unofficial method of determining a machine's manufacturer. The method is not 100 percent reliable, however, since the manufacturer might decide, for whatever reason, to change the location or actual string of the copyright notice. To prevent problems with your users, we have provided some software escape hatches. When your program cannot locate the copyright, the software will default to its generic-machine mode. However, some of your users might not be satisfied with the resulting loss of performance, so we also suggest that you incorporate into your program a method for the user to set the machine's compatibility level. Here are three examples of how to do that.

First, you can let your user input a slash option such as /IBM when invoking your software with its command string, as in

```
yourprm /IBM
```

A good example of a program that uses the slash-option approach is Microsoft's SYMDEB.

Second, you can incorporate an "It's a compatible" override as part of an optional installation process. Finally, you can let your user set compatibility as part of the DOS environment. For example,

```
set compatibility = IBM
```

The use of any of these methods in your program lets those users who need more performance obtain it. When a new machine appears on the market, advanced users will be able to use that machine to its fullest potential.

An example of a C-language implementation of a features table is presented in listing 1. [*Editor's note: Only listings 1, 2, and 3 are reproduced here. The entire set of listings* (1 *through* 9) *is available for downloading. See the note at the end of this article.*] In that example, machine characteristics are defined as bit flags in the variables FEATURES and VIDEO_FEATURES.

You should develop your own set of bit flags for the features your program needs to use. These bit flags are bits that you can set either off or on and that exist within a word. For example, if we set bits 1, 4, and 8 of a word "on" while leaving the remaining bits "off," the value of that word is 13 hexadecimal. Since we are interested in determining only if particular bits within the word are set, the value of the word is not important. Instead, we extract the bits that we are interested in by using a mask. Our sample programs in the listings illustrate how the mask is used.

Some of your programs might require information about the compatible's operating system version or the availability of certain BIOS calls. We will not discuss the detection of this additional information here, but we

(continued)

Listing 1: A *sample features table, as described in the article.*

```
/*

*/

/*
   This code contains a C language Features Table which contains the location
   and copyright string that uniquely identify a machine.  In addition to the
   copyrights, the bit fields for the  FEATURES and  VIDEO_FEATURES variables
   are defined. Note that this code is compiled with the Lattice  C  Compiler
   version 2.15, using -md -n options.  Copyright  (c)  1986  Dan Jacobs  and
   Joel Rosenblum for public, unrestricted use.
*/

/* bit flags for FEATURES */

#define IBMPC          0x0100  /* IBM PC, XT, Portable, */
#define IBMPCAT        0x0200  /* IBM AT */
#define IBMCOMPAT      0x0400  /* IBM PC BIOS Compatible */
#define IBM_CONVERT    0x0800  /* IBM Convertible */
#define GENERIC        0x1000  /* Assumed generic PC */
#define NO_DMA         0x2000  /* Machine has no DMA */
#define WANG           0x4000  /* Wang PC special case*/
#define TIPROF         0x8000  /* TI Professional PC special case */
#define CPU_88         0x0001  /* 8088, 8086 processor */
#define CPU_186        0x0002  /* 80188, 80186 processor */
#define CPU_286        0x0004  /* 80286 processor */
#define CPU_V20        0x0008  /* V20, V30 processor */
#define NDP            0x0010  /* 8087, 80287 math coprocessor */

/* bit flags for VIDEO_FEATURES */

#define CGA            0x0001  /* IBM Color Graphics Adapter */
#define MONO           0x0002  /* IBM Monochrome Adapter */
#define HERCULES       0x0004  /* Hercules Monochrome Adapter Card */
#define PGA            0x0008  /* IBM Professional Graphics Adapter */
#define EGA_MONO       0x0010  /*  w/Monochrome Monitor */
#define EGA_COLOR      0x0020  /*  w/Color Monitor */
#define EGA_HIGH       0x0040  /*  w/High Resolution Color Monitor */
#define UNKNOWN        0x0080  /* Unknown graphics type */
#define ANSI           0x0100  /* ANSI.SYS installed */

/* Additional bit fields may also be defined, please see TABLE 1 for suggestions */

struct machine_info {
    char    *logo;  /* Unique copyright string */
    long    addr;   /* String memory location  */
            /* Note: actually (char *) must be (long) to pass Lattice C */
    int type;    /* Machine attributes */
} feature_table[] = {

/*      copyright notice, physical address, pctype */

    { "IBM", 0xFE00EL, IBMPC|IBMCOMPAT},      /* All IBMs */
    { "COMPAQ", 0xFFFEAL, IBMCOMPAT},         /* All Compaqs */
    { "Corona", 0xFE00F, IBMCOMPAT},          /* Old version Corona */
    { "Corona", 0xFE01A, IBMCOMPAT},          /* Version 3.10 ROM Corona (also Phillips) */
    { "M.P.C.", 0xFDB2D, IBMCOMPAT},          /* New Columbia */
    { "Columbia", 0xFF768, IBMCOMPAT},        /* Old Columbia */
    { "Eagle PC", 0xFFFAA, IBMCOMPAT},        /* Eagle PC */
    { "Eagle PC", 0xFF810, IBMCOMPAT},        /* Eagle PC Plus */
    { "Zenith", 0xFB000, IBMCOMPAT},          /* Zenith Data Systems */
    { "Zenith", 0xFC2FF, GENERIC},            /* Zenith 100 */
    { "MITSUBISHI", 0xFC02A, IBMCOMPAT},      /* Sperry PC and Leading Edge */
    { "TVS", 0xFE003, IBMCOMPAT},             /* TeleVideo */
    { "OSM", 0xFFFF5, IBMCOMPAT},             /* OSM Rom Version 3.6 or later */
    { "OLIVETTI", 0xFC050, IBMCOMPAT},        /* AT&T-IS PC 300 and Xerox  */
```

(continued)

```
                                              /* NOTE: AT&T is not supporting V1.0 ROMs */
    { "WANG", 0xFFFC2, WANG|GENERIC},         /* WANG Professional */
    { "ADDS", 0xFC050, IBMCOMPAT},            /* Applied Data Digital Systems Model PC/I */
    { "CROS", 0xFE000, IBMCOMPAT},            /* Seattle Telecom Turbo Boards */
    { "PCPI", 0xFE00F, IBMCOMPAT},            /* Personal Computer Products */
    { "TAVA", 0xFE018, IBMCOMPAT},            /* Tava PC */
    { "Tandy", 0xFC02B, IBMCOMPAT|NO_DMA},    /* Tandy 1000 has no DMA without memory upgrade */
    { "Tandon", 0xFC013, IBMCOMPAT},          /* Tandy 1200 */
    { "Texas", 0xFE022, TIPROF},              /* TI Professional V1.23 & V2.11 SYSROM */
    { "American", 0xFE004, IBMCOMPAT},        /* American PC lookalike */
    { "STD", 0xFE00EL, IBMCOMPAT},            /* STD Turbo Boards in IBMs */
    { "STD", 0xFFFEAL, IBMCOMPAT},            /* STD Turbo Boards in COMPAQs */
    { "TOMCAT", 0xFE028, IBMCOMPAT},          /* Tomcat AT clone */
    { "WYSE", 0xFC003, IBMCOMPAT},            /* WYSE PC BIOS Version 1.08 */
    { "Hewlett-Packard", 0xF0024, IBMCOMPAT}, /* Hewlett-Packard Vectra AT Compatible */
    { "Morrow", 0xFE073, IBMCOMPAT},          /* Morrow Pivot II & Zenith 171 Desktops */

    /* Additional machines are added to the list starting here */

    { NULL, NULL, 0}
};
```

Table 1: A *summary of some bit-field considerations, information that programs might require before they can run on* IBM PC–*compatible systems.*

CPU-related features:
- detection of the primary processor in the machine
- detection of the processor's speed
- detection and use of coprocessors

Video-related features:
- present video type(s)
- present video mode
- installation of ANSI.SYS video device driver
- ability to use IBM expanded-video character set

Operating system–related features:
- availability of certain BIOS-level calls
- DOS version check

Disk-related use and detection:
- floppy and/or hard disk drive type
- floppy and/or hard disk drive size
- detection of floppy disk drive mode
- ability to use interrupt 13h calls
- DMA detection

have provided a summary of some bit-field considerations for use in the FEATURES variable in table 1.

The definitions are then followed by the copyright notices that appear in a structure containing the location and unique identifying string for some sample machine BIOS ROMs. Note that each defined machine in the table contains its own set of characteristics that are then placed in the bit-flag variable.

Two machines, the Wang PC and the TI Professional, are important examples of how you can use the table to deal with some unusual compatibility considerations. For example, the Wang PC requires clearing the direction flag before making disk I/O calls from assembly language routines. This problem does not occur when making the I/O calls from most high-level languages since, in these languages, the direction flag is usually left in the increment position. The TI Professional requires a substitution of interrupt 4Eh in place of any assembly language interrupt 13h direct disk I/O calls. Thus, listing 1 defines the special bit flags WANG and TIPROF. If your program needed to use any of these calls on one of these two machines, the defined bit flag would handle the branch within your program. Some examples of use of the bit-flag branch to control program performance optimization are contained in the Detecting Video Type section of this article.

The sample features table in listing 1 also shows how you might determine the presence of a turbo board. The copyright notice was given for an STD turbo board used in an IBM or Compaq PC. Again, once it detects the board, your program could act accordingly. Of course, your program could also incorporate detection for other installed circuit cards using their copyright notices.

In summary, the equipment features table places all the burden of compatibility and machine optimization in one convenient place. If a computer is not on the list, the program assumes it is generic. Otherwise, the capabilities of the machine are already "known" and your program would automatically use them. For new or previously unknown machines, your user must either accept the automatic generic-machine mode or override it by providing the program with performance optimization. Therefore, as a service to your users who prefer to use software as it comes off the shelf, you have a responsibility to keep your features table as up-to-date as possible. Perhaps the features table will become a public domain standard for use by all programmers.

USING THE FEATURES TABLE

Now that you have developed a features table, how do you use it in your code? Listing 2 contains a C-language routine that uses the features-table structure defined in listing 1 to determine machine compatibilities. The routine reads the copyright notice, if it can locate one, and sets the bits within the variable FEATURES. The routine also calls separate functions that set additional bits within FEATURES and VIDEO_FEATURES. Your program's code then checks the appropriate bit within the variable to determine if a particular ability exists for the machine on which your software is currently operating.

It is important that the routines in your program that use the features variables exist at the lowest level of your code. The reason for this is so that the rest of your program will sit above the machine's hardware and be oblivious to the equipment present. Your program will execute the appropriate routines based on the information contained within the variables. For example, suppose your software could speed up its disk access by using BIOS interrupt 13h calls. If FEATURES indicated that the machine your software is currently operating on is 100 percent IBM-compatible, then you would use the BIOS interrupt; otherwise, you would use DOS interrupt calls 25h and 26h to access the disk. In effect, you create a device-independent interface for your program. If you later find that you need to define another FEATURES bit or you need to extend the length of the FEATURES variable, you will be able to make the change with a minimum of recoding.

Now that we have defined the bits

(continued)

Listing 2: *C-language routine that uses the features table of listing 1 to determine machine compatibility.*

```
/*

*/

/*
   This code shows  how the Features Table is used to set up
   the bit flags  in the FEATURES  variable.  Once  FEATURES
   is  set  for  the  current  machine,  your  other program
   modules can  use it to  determine what  section  of  code
   should be executed to yield the best program performance.

   Copyright (c) 1986 Dan Jacobs and Joel Rosenblum

   Compiled using Lattice C ver 2.15, using -md -n options
*/

unsigned int video_features;   /* global variable which holds video attributes */

unsigned int set_features()
{
     char *cp;                    /* Pointer */
     struct machine_info *p;      /* Features Table Structure */
     unsigned int feature;        /* FEATURES bit field variable */

     /* Assume the PC is generic to start */
     feature = GENERIC;           /* default to generic */

/*
   A pseudo code example of how to override the automatic set of FEATURES
   assuming that either a slash option  or a DOS environment variable is
   set.  Your actual code will depend upon how you do the parsing.  See
   the  section  on using the  machine's copyright notice to determine
   compatibility  in our text.

begin pseudo code example --

     if ((/IBM entered on command line) |
         (compatibility = IBM is in the environment)) {
         feature = IBMCOMPAT;
         goto cpu_test;
     }
-- end pseudo code example */

     for (p = feature_table; p->addr != NULL; p++) /* If next table entry is NULL, end loop */
         if (strncmp((char *) p->addr, p->logo, strlen(p->logo)) == 0) {
             feature = p->type;
             break;
         }

     if (feature & IBMPC) {           /* IBM Personal Computers */
         cp = (char *) 0xFFFFE;        /* physical address 0FFFF:E */
         switch (*cp) {
             case 0xF9:  /* IBM Convertible */
                 feature |= IBM_CONVERT;
                 break;
             case 0xFC:       /* IBM AT */
                 feature |= IBMPCAT | CPU_286;
                 break;
             case 0xFD:  /* IBM PC JR */
                 feature |= NO_DMA;
                 break;
             case 0xFE:  /* IBM XT or Portable */
             case 0xFF:  /* IBM PC */
                 break;
             default:
                 break;  /* unknown IBM type */
```

(continued on page 190)

that make up the variables, the next sections will show you the details behind setting those flags. These examples illustrate how your programs can detect such features of your machine as its processor, effective clock speed, and video display. These routines are the foundation for the creation of your own hardware-independent modules.

DETERMINING CPU TYPE

Software that is dependent upon processor type needs to determine under which processor it is operating. Examples include operating system software that might require the use of certain processor modes such as the Intel 80286's protected mode or the use of the PUSHA instruction on the 80186.

Listing 3 gives a C-callable assembly language routine that you can use to determine the presence of the Intel 8088/8086, 80186, and 80286 processors and the NEC V20 and V30 processors, so you can set the corresponding bit in FEATURES.

Your program's performance might improve by detecting and using a math coprocessor. Examples include programs with heavy trigonometric or floating-point operations such as statistical analysis and speech-processing software. Two methods for detecting a math coprocessor are in listing 4.

Finally, it might be important to determine the processor's clock speed. Timing loops are an important part of communications software, and the assembly language procedure in listing 5a lets you set processor-speed-independent timing loops. Listing 5b shows an example of how to use the timing routine.

DETECTING VIDEO TYPE

Automatic video detection should be an almost universal part of your software if you want to display any of the IBM extended character set or bit-mapped graphics on the video display. Further, automatic display-type determination eliminates placing the burden of video-display installation

(continued)

on the user. Of course, a feature that lets users override this display determination, similar to the ones described in the copyright-notice section of our article, could be incorporated into your program.

Video detection starts with determination of the active display-adapter type. The five major adapter types are IBM Monochrome Adapter, Hercules Graphics Card, IBM Color Graphics Adapter (CGA), IBM Enhanced Graphics Adapter (EGA), and IBM Professional Graphics Adapters (PGA). The procedure in listing 6 detects these adapter types and sets the appropriate performance bits in the variable VIDEO_FEATURES.

The performance bits in VIDEO_FEATURES activate code within your program, which can then take advantage of the selected display adapter's features. Programs that allow use of the EGA's colors, for instance, could be set active when that adapter type is detected. You could also use VIDEO_FEATURES bit flags to politely notify your user that a particular display is not present if, for instance, your software absolutely requires the use of that display.

If none of the above display adapters is in the compatible's hardware, the display might still have the ability to position the cursor directly on the screen. Direct cursor positioning lets screen displays appear without scrolling. Displays with this capability are ANSI-compatible. ANSI compatibility implies that the ANSI.SYS device driver has been installed by the user as a part of the CONFIG.SYS file on the compatible machine. If you incorporate the code contained in listing 7 into your program, the program will be able to directly check whether ANSI.SYS has been installed on the user's machine.

In addition to determining ANSI compatibility, you might want to take advantage of the IBM extended character set so that you can design screen displays and menus to look better. Screen enhancements such as character blinking, underlining, and boldface might also be possible. We

(continued)

```
        }
    }

    /* add the cpu type to FEATURES -- see listing 3 */
    feature = cputest(feature);

    /* check for numeric data processor -- see listing 4 */
    feature = test_ndp(feature);

    /* add the video display type to VIDEO_FEATURES -- see listing 6 */
    video_features = video_test(feature);

    /* check if  ANSI.SYS is present -- see listing 7 */
    if (check_ansi()) {
            video_features |= ANSI;
        } else {
            putch(0x0D);  /* This cleans up any garbage left on the screen */
            putch(' ');   /* by the ANSI check if ANSI.SYS  is not present */
            putch(' ');
            putch(' ');
            putch(' ');
            putch(0x0A);
            putch(0x0D);
        }

        return (feature);
    }
```

Listing 3: A *C-callable assembly language routine for determining the type of microprocessor used in an* IBM PC *compatible.*

```
; Cputest.asm is the Lattice C-callable assembly language routine that determines
; the machine's processor type. Copyright (c) 1986 Dan Jacobs and Joel Rosenblum

        name    cputest         ;determine CPU type
        include dos.mac         ; Lattice C memory model configuration macro
                                ; In this case it is a copy of dm8086.mac

; processor type equates
CPU_88     equ 01H        ; Intel 8088 - 8086
CPU_186    equ 02H        ; Intel 80188 - 80186
CPU_286    equ 04H        ; Intel 80286
CPU_V20    equ 08H        ; NEC V20 - V30

        PSEG

comment\**********************************************************************

NAME
        cputest
SYNOPSIS
        unsigned int cputest (features)
        unsigned int features;          see definition of machine type
DESCRIPTION
        returns features with the proper active CPU type or'ed in

***************************************************************************\

                public  cputest
cputest         proc    near

                push    BP              ; save the frame pointer (if called from C)
                mov     BP, SP
                                        ; next, save the passed existing features
                mov     AX, 4[BP]
```

(continued)

PC-DESK
TAKES YOU BEYOND SIDEKICK

PC-DESK. The Ultimate in Desk-Top Management

Now, finally at your command. A completely integrated word processor, calendar/reminder, name/address/phone database and directory. Letter/label/report printer, mailmerge, invoice system and printing calculator.

PC-Desk does just about everything you need. It's a powerful and easy-to-use tool that reduces your work and makes your business life easier. And, it does more than hundreds of higher-priced programs.

The Most Powerful Feature

PC-Desk automatically prepares your letters, invoices and memos.

It's fast and easy to send individually typed, personalized letters, either singly or by the hundreds. Just select the names you want. Tell PC-Desk which letter to send, and it does the rest.

It puts in the special greeting you use for each person. It adds the variable information you want. . .the shipment date, company or product name, or whatever you choose. All at the touch of a button.

PC-Desk is simply spectacular. Users say, "This is the way software ought to be!" Try it for 60 days. If you're not completely satisfied, return it for a full refund.

Free Updates, Free Support.

That's right. Free updates. (Remember, if you already own PC-Desk, send for your free update to version 6.5.)

And, if you ever need help, just call us. We'll talk you through it.

Look At Everything You Get!

Easy Instructions
Step by step directions on the screen. You'll always know what codes to use. Even if you're new to computers, you'll be efficient right away.

Calendar/Reminder
24 reminders per day. Reminder days are highlighted. Put the cursor on any day. . . all your reminders pop up. See a full month at a time. Flick back and forth between months. Print reminders with a single keystroke. You're organized and you'll stay organized.

To Do List
Tracks up to 20 projects. You'll never forget what you need to do.

Word Processor
Full-featured. . .why use any other word processor? PC-Desk gives you professional-looking documents. . .with bold, underlining, centering, and justification. Word wrap, block move, search, margin settings. And no waiting for screens to come up with Turboscreen.

Personal Address Book
Automatically printed in a pocket-sized format. Cross-indexed by company name. Breast-pocket holder free when you order PC-Desk.

Database
Outstanding organization and flexibility. Easy to access and maintain. Name, address, phone number (work and home), salutation, title, date, variables for automatic printing, comments, and more. Sort on zip, or alphabetically by name or company. Search, scroll through, change or delete. Select names for group mailings. Make notes on client contact. Print your rolodex, envelopes, (with return address too), letterheads, invoices, labels, entire name and address file.

Mail Merge
Fully integrated with database. Letters with up to four variables.

Calculator
Print your calculations. Pops up into word processor. Hex, binary, and octal for programmers.

Memory Partitioning
Lets you flip back and forth between PC-Desk and other programs. Cut and paste.

Logs
Client billing, computer use, telephone, and expense logs. Great for tax records! Print monthly statements with total time and expenses. Edit your logs. With auto-dialer calls are logged automatically.

Auto-Dial
Save time! Auto-dial any phone number in your database. Choose work or home number.

Encrypt
Protect your files from prying eyes.

DOS Functions
Copy, delete, print, view, renamefiles easily.

On-Screen Clock

Manual
Full instructions.

Not Copy Protected

Software STUDIOS
8516 Sugarbush, Suite 104
Annandale, VA 22003
703/978-2339

PC-DESK
SIMPLY THE BEST.

To order, or for more information:
Call Toll Free 1-800-A-PC-DESK

YES! Rush me the best—PC-Desk.
60-Day Money-Back Guarantee.

★ SPECIAL OFFER ★
Only $79
(plus $5.00 Shipping & Handling).
You save $46 on the regular $125 price.
Offer Expires in 30 Days

☐ Check ☐ Money order
☐ MasterCard ☐ Visa

Card # ______________________
Expiration date ______/ ____________
Name ______________________
Firm ______________________
Address ______________________
City ______________________
State/Zip ______________________

Requirements: IBM-PC, AT, XT, or compatible. 128K for program, 272K for memory partitioning, MS-DOS 2.0. Modem required for Auto-Dial only.

```
            push    AX
        ; check for  8088 or 8086  by using the  SHR instruction  since the
        ; 8088 and 8086 do not mask cl with 07H before executing the shift.
            mov     CL, 20H
            mov     AX, 1
            shr     AX, CL
            test    AX, AX          ; if after the shift AX is the same
                                    ; as before, it's an 8088 - 8086 or V20 - V30
            jnz     check_80186     ; else, continue checking other Intel CPUs
        ; check for V20 or V30 by detecting if PUSHA is a valid instruction
        ; on the NEC CPUs
            mov     BX, SP          ; save SP
            pusha
            cmp     BX, SP          ; if SP has not been decremented, then
            je      is_88           ; it's an 8088 - 8086
            popa                    ; else, we restore registers
            mov     AX, CPU_V20
            jmp     return
is_88:      mov     AX, CPU_88
            jmp     return
check_80186:
        ; check for the 80188 or 80186 by detecting if SP is updated
        ; before or after it is pushed.
            push    SP
            pop     BX
            cmp     BX, SP
            je      is_286          ; if updated after, it's an 80286
            mov     AX, CPU_186     ; else, it's an 80186 or 80188
            jmp     short return
is_286:     mov     AX, CPU_286

return:     pop     BX              ; recall saved features
            or      AX, BX          ; and or cputype into other features bits
            pop     BP
            ret
cputest     endp
            ENDPS
            end
```

```
 ┌──────────────────────────┐        +---------------------------+
 │ IBM extended characters  │        | Standard ASCII characters |
 └──────────────────────────┘        +---------------------------+

     100 percent IBM PC                   Generic MS-DOS
   compatible display box                  display box
```

Figure 1: *Two boxes drawn by the routine in listing 8, on a system that has the IBM extended character set available (left) and on a system that does not and uses only ASCII characters (right).*

assume that a machine has an extended character set if it has the IBMCOMPAT bit set in the variable FEATURES.

The sample program in listing 8 shows how to draw a simple box that could take advantage of the machine's ability to use the IBM extended character set. The draw_box routine uses the IBMCOMPAT bit flag contained in FEATURES to determine which character set can be displayed. Figure 1 shows the difference between boxes drawn with the two sets.

The draw_box routine uses a routine called move_cursor, which is contained in listing 9. Whereas draw_box uses a bit flag set in FEATURES, move_cursor relies more directly upon the video hardware. Thus, it uses bit flags set in VIDEO_FEATURES, as well as a bit set in FEATURES. Move_cursor then moves the cursor to the position row, column. It decides which method is needed to move the cursor on your particular machine without burdening the calling program with the task.

Note that if your program detects the presence of ANSI.SYS operating on a machine that is IBMCOMPAT and sets screen attributes using BIOS-level video calls, then it should not use the low-level video DOS calls to display additional text. To do so will overwrite the attributes you have just set with what ANSI.SYS believes the attributes to be. Complicated prob-

(continued)

NEW FROM

Compuclassics

SOFTWARE

WORD PROCESSORS

LEADING EDGE
Word Processor with Merge & Spell 79.00
LIFETREE
Volkswriter 3 159.00
MICROPRO
Wordstar 179.00
Wordstar Propack 249.00
Wordstar 2000 245.00
MICROSOFT
Word (Version 3) 279.00
MULTIMATE
Multimate 235.00
Multimate Advantage 319.00
SATELLITE SOFTWARE
Word Perfect 219.00
Word Perfect Library 59.00

WORD PROCESSING UTILITIES

BORLAND INTERNATIONAL
Turbo Lightning 62.00
LIVING VIDEOTEXT
Thinktank 99.00
MAX THINK
Max Think 49.00
WRITING CONSULTANTS
Word Finder 60.00

DATA BASE MANAGEMENT

ASHTON TATE
dBase III Plus 419.00
BORLAND INTERNATIONAL
Reflex 85.00
LEADING EDGE
Nutshell 69.00
M.D.B.S.
Kman II 299.00
Kman Bundle 369.00
MICRORIM
RBase 5000 269.00
Clout II 145.00
RBase System V CALL

DATABASE UTILITIES

FOX & GELLER
Quickreport 145.00
NANTUCKET
Clipper 359.00
SOFTWARE TOOLS
Viewgen 49.95

SPREADSHEETS/ INTEGRATED SOFTWARE

ASHTON TATE
Framework II 429.00
COMPUTER ASSOCIATES/I.U.S.
Supercalc 4 295.00
DIGITAL RESEARCH
GEM Draw 159.00
GEM Collection 119.00
GEM Desktop 32.00
LOTUS
Lotus 1-2-3 Version 2 339.00
Symphony 449.00
MICROSOFT
Multiplan 129.00
MIGENT SOFTWARE
Ability 59.00
PAPERBACK SOFTWARE
VP Planner 59.00
SOFTWARE GROUP
Enable 359.00
SOFTWARE PUBLISHING
PFS File, Write, Plan 80.00 each
PFS Report 69.00
SYMANTEC
Q & A 225.00

SPREADSHEET UTILITIES

CAMBRIDGE CO-OPERATIVE
Spreadsheet Analyst 75.00
LOTUS
1-2-3- Report Writer 115.00
TURNER HALL
SQZ 59.00

PRESENTATION GRAPHICS

AMERICAN SMALL BUSINESS
Prodesign II 199.00
BLOC DEVELOPMENT
Form Tool 42.00
DECISION RESOURCES
Chartmaster 209.00
Signmaster 145.00
ENERTRONICS RESEARCH
Energraphics 2.0 329.00
GRAPHICS COMMUNICATIONS
Graphwriter Combo 329.00
Freelance 219.00
MICROGRAFX
PC Draw 159.00
In-A-Vision 199.00
Windows Draw 85.00
MICROSOFT
Chart 189.00
SPRINGBOARD
Newsroom 32.00
T-Maker
ClickArt Personal Publisher 129.00
UNISON WORLD
Printmaster 32.00
Art Gallery I & II 22.00 each

PROJECT MANAGEMENT

BREAKTHROUGH
Timeline 235.00
COMPUTER ASSOCIATES/I.U.S.
Superproject Plus 279.00
HARVARD SOFTWARE
Total Project Manager 279.00
SCITOR
Project Scheduler Network 339.00

ACCOUNTING

COMPUTER ASSOCIATES/I.U.S.
Each Accounting Module 309.00
Payroll 389.00
Easy Plus 89.00
OPEN SYSTEMS
GL/AR/AP/Payroll/Inventory 429.00
Purchase Order/Sales Order/Report Writer 379.00
Resource Manager 239.00
SBT
Accounting Software for dBase III
dLedger/dPayroll/dProject 275.00
dOrder/dInvoice/dPurchase 145.00

PERSONAL MANAGEMENT/ INVESTMENT

MECA
Managing Your Money 99.00
MONOGRAM
Dollars & Sense 99.00

COMMUNICATIONS/ TERMINAL EMULATION

HAYES
Smartcom II 95.00
MERIDIAN TECHNOLOGY
Carbon Copy 109.00
MICROSTUF
Crosstalk XVI 99.00
Remote 99.00
PERSOFT
Smarterm 220 149.00
SOFTKLONE
Mirror (Crosstalk Clone) 29.00

UTILITIES

BORLAND INTERNATIONAL
Superkey 40.00
Sidekick (Unprotected) 48.00
Travelling Sidekick 40.00
CENTRAL POINT
Copy II PC 29.00
Copy II Option Board 90.00
EXECUTIVE SYSTEMS
X-Tree 32.00
FIFTH GENERATION
Fastback 99.00
MICROSOFT
Windows 65.00
PERSONICS
Smart Notes 49.00
PETER NORTON
Norton Utilities 55.00
Norton Commander 37.00
QUAID SOFTWARE
Copywrite 39.00
Zerodisk 55.00
SOFTLOGIC SOLUTIONS
Disk Optimizer 37.00
Double Dos 37.00
Software Carousel 25.00
STERLING SOFTWARE
Intelligent Back-up 79.00

PROGRAMMING LANGUAGES

BORLAND INTERNATIONAL
Turbo Pascal W/8087 & BCD 59.00
Turbo Database Toolbox 40.00
Turbo Tutor 22.00
Turbo Graphix Toolbox 40.00
Turbo Jumbo Pack 139.00
LIFEBOAT ASSOCIATES
Lattice C Compiler 259.00
MICROSOFT
Macro Assembler 95.00
Fortran Compiler 219.00
Quick Basic 65.00
C Compiler 289.00

HARDWARE

MEMORY/MULTIFUNCTION BOARDS

A.S.T. RESEARCH
Six Pack Plus w/64k 170.00
Six Pack Premium w/512k 409.00
Rampage PC with 256k 275.00
Rampage AT with 512k 485.00
PARADISE SYSTEMS
Five Pack with 0 k 120.00
TALLTREE SYSTEMS
JRAM 3 PC with 0 k 195.00
JRAM 3 AT with 0 k 245.00
JLASER 270.00

GRAPHICS CARDS

HERCULES
Monochrome Plus 185.00
Color Card 159.00
PARADISE
Modular Graphics Card 219.00
EGA Auto Switch Card 359.00
SIGMA DESIGNS
Sigma 400 425.00
Sigma EGA 349.00
TECMAR
EGA Master 305.00
ZUCKERBOARD
Color Card (No paralled Port) 59.00
Color Card (Paralled Port) 79.00
Monochrome Graphics (Hercules Compatible) 79.00
I-D Mini w/clock calendar, parallel & serial port 75.00

MODEMS

HAYES
1200 399.00
1200B with Smartcom II 379.00
MIGENT
External Modem–1200 Baud 99.00
PRACTICAL PERIPHERALS
1200 Internal half-card 140.00

MONITORS

AMDEK
310 A 155.00
MAGNAVOX
Monochrome Monitor (Amber/Green) 90.00
Color (320X200) 279.00
NEC
NEC Multisync 575.00
PRINCETON
Max 12E 165.00
HX 12 469.00
SR 12 629.00

PRINTERS

C. ITOH DIGITAL GROUP Call for Prices
CANON
LBP-8A1 Laser Printer 2,285.00
CITIZEN
120D 199.00
Premiere 35 LQ 475.00
EPSON Call for Prices
NEC
Pinwriter P5 1,159.00
Pinwriter P6 529.00
OKIDATA Call for Prices
TOSHIBA
P321 580.00
P351 1205.00
CALL FOR ALL PRINTER ACCESSORIES

PLOTTERS/ PRESENTATION SYSTEMS

POLAROID
Polaroid Palette 1,499.00
SOFTWARE GARDEN
Dan Bricklin's Demo Program 60.00

HARD DISKS/ TAPE BACK-UPS

12 INTERFACE
10 MB External Tape Back-up 579.00
IRWIN MAGNETICS
10 MB Internal Tape Back-up 549.00
20 MB Internal AT Tape Back-up 859.00
MOUNTAIN
20 MB Hard Card 825.00
SEAGATE
20 MB Hard Disk with Controller 469.00

INPUT DEVICES

KEYTRONICS
KB 5151 175.00
MICROSOFT
Mouse with PC Paintbrush 129.00
MOUSE SYSTEMS
Mouse Systems Mouse with OR Halo II 99.00

SURGE PROTECTION/ BACK-UP POWER

COMPUTER ACCESSORIES
P15 Power Director 115.00
EPD
EPD Lemon 6 Outlet 32.00
IBC/TADIRAN
AT Replacement Battery 27.50
KENSINGTON
Masterpiece 89.00
Masterpiece Plus 115.00

RAM CHIPS/ CO-PROCESSORS

Intel, NEC, Toshiba, etc. Call for prices

COMPUTERS

PC/XT Compatible Computers
(FCC & UL Authorized)
Complete PC Systems from under **$850.00**
Complete XT Systems from under **$1,150.00**
Call for price quotation
We can build the system according to your needs

ADVANTAGES

- Orders placed before 3 p.m. (PST) shipped same day.
- We welcome corporate accounts.
- Free technical support.
- Immediate replacement of defective goods.
- Bulk discounts.
- One million dollar inventory.
- Immediate shipment on purchase orders from qualified institutions (schools and universities included).
- No hidden charges.

TERMS:

All prices subject to change without notice. We do not guarantee compatibility. No surcharge for VISA or MasterCard. 2% surcharge American Express. No returns without RA#. Short shipments must be notified within 48 hours. 15% restocking fee on non-defective goods: $3.00 C.O.D. charge. Shipping $4.00 per item, less on bulk orders. ($8.00 Blue Label.) (Higher for some hardware items.)

Order desk & technical support open:
8 a.m. to 6 p.m. Monday–Friday
10 a.m. to 2 p.m. Saturday (PST)
6934 Canby St. Suite #109-110
Reseda, CA 91335
Call for prices for any item not included in this price list.

1 818 705-1895 US 1 800 231-6603 CA 1 800 328 4473 US

lems like this are easier to solve with the availablity of some of the seemingly unrelated bit flags. Parts of your program gain compatibility and performance on some machines but not on others. Here's a scenario.

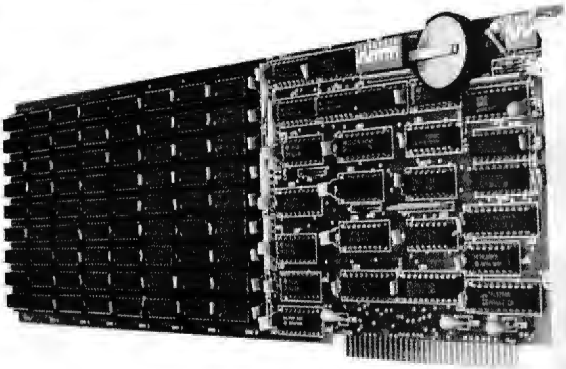
Incorporating processor- and speed-detection procedures into your program allows for run-time performance determination of the PC compatible. Based on that run-time analysis, you could enable sections of optimizing code in the program by setting your program's "performance bit flags." For example, a multiuser spreadsheet program that allows for multiple users only if operating on an 80286-based system could determine if the processor type was an 80286. If so, a flag within the software could be set that allows multiuser capability. Then detection would proceed to determine the presence of a math coprocessor. If an 8087 or 80287 was detected, a bit would enable use of the hardware-based math routines. Finally, your device time-out loops could be adjusted according to the results of the processor-speed check. Combining this analysis with the screen display lets your spreadsheet program allow charts and plots to appear on the screen, if possible.

Didn't we mention the salt mines? We have eliminated some of your work, but good programs come from good design. You now have a program that appears to be hardware-intelligent and contains performance increases to match. This combination keeps your users happy and helps you sell more software. Meanwhile, you can go back to the salt mines and write the next software masterpiece.

Editor's note: The listings from this article are available on disk, in print, and on BIX. *See the insert card following page 176 for details. Listings are also available on* BYTEnet. *See page 4. Listings 1, 2, 5b, 7, 8, and 9 are C source code and are available as the file* PERPROPC.C*; listings 3, 4, 5a, and 6 are assembly language code and are available as the file* PERPROPC.ASM. *You will need the Lattice C compiler or a compatible C compiler to use the source code listings.* ■

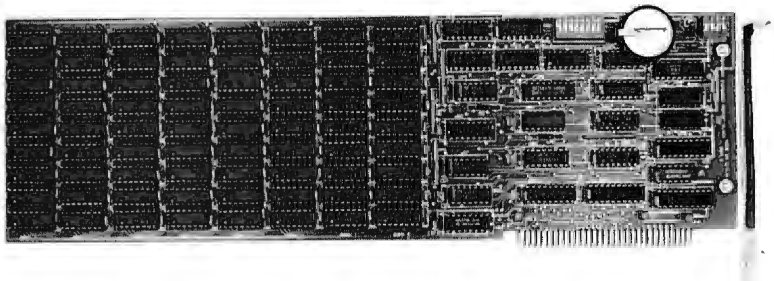
APPARAT INTRODUCES THE...

LIMBO*



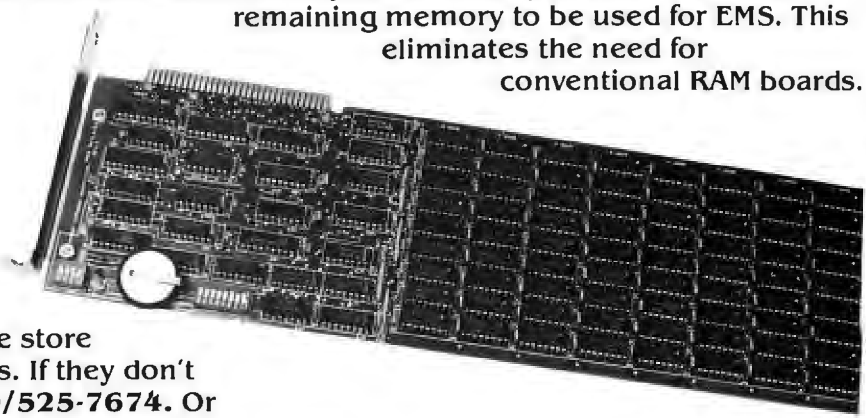
You've got a problem. You have 640K of RAM and you still run out of memory! Go to your favorite computer store. Then ask about expanded memory cards. See the problem? Just too many to pick from. Now, ask to see the **LIMBO™.** No problem. There's only one. Ours. Except you probably don't know a **LIMBO** from a dance step. Maybe you guessed that it is an expanded memory board that supports the Lotus/Intel/Microsoft Expanded Memory Specification (EMS). And that it has a 2 MB capacity and up to 8 MB is supported by the memory management software.

The **LIMBO** includes a clock/calendar and 8 banks for use by either 64K or 256K DRAM. Software includes the EMS memory manager, EMS compatible RAM Disk and print spooler.



The **LIMBO** is manufactured by Apparat in Denver, Colorado, and is covered by Apparat's Exclusive Limited Lifetime Warranty. So you can be confident that you won't find yourself out on a limb. (We've been in business since 1978.)

Need an EMS multifunctional board? Ask to see the **LIMBO II.** The **LIMBO II** has a 1.25 MB capacity with a parallel port, a serial port, a clock/calendar, and game-stick adapter. **LIMBO** and **LIMBO II** have flexible starting addresses and can be set to fill system memory to 640K, with the remaining memory to be used for EMS. This eliminates the need for conventional RAM boards.



The **LIMBO** is only **$199** with 64K of RAM installed or **$549** with full 2 MB of RAM installed.

The **LIMBO II** is only **$299** with 256K of RAM installed or **$499** with full 1.25 MB of RAM installed.

Go check out the **LIMBO** at the same store carrying other expanded memory boards. If they don't have the **LIMBO,** have them call us: **800/525-7674.** Or write: **Apparat, Inc.,** 6801 South Dayton, Englewood, Colorado 80112, 303/799-0819.

NOTE: The LIMBO boards work great with Lotus 1-2-3 Release 2.

Apparat, Inc.

Inquiry 21

6801 South Dayton, Englewood, CO 80112 — Customer Service 303/799-0819
Retail Outlet in the Chicago area 312/640-0322

* Lotus is a registered trademark of the Lotus Development Corporation. * Intel is a registered trademark of the Intel Corporation. * Microsoft is a registered trademark of the Microsoft Corporation.

* Lotus Intel Microsoft EMS BOard

Inquiry 55

RAM-LOADABLE CHARACTER SETS FOR THE IBM PC

How to create and use RAM-based character fonts

BY RICHARD WILTON

MOST IBM PC users take for granted the size and shape of the characters on their video displays. However, the newer video display adapters from IBM and Hercules let you change the font of the displayed characters, as well as the set of displayable characters, without compromising the display's speed or clarity.

The ability to display characters of arbitrary size and shape helps programmers make the user interface more natural. For example, a word-processing program can display characters exactly as they will appear on the printed page. Foreign-language alphabets, mathematical symbols, and simple graphics elements can all appear on a single screen.

Almost all the video display adapters available for the IBM PC can display some sort of customized character set (see table 1). Some adapters provide more built-in support for custom character sets than others. In general, a greater amount of hardware and firmware support simplifies character-set software and delivers greater speed.

There are two general approaches to displaying customized character sets on an IBM PC. One is to define and display each character with special software routines. The other is to load a table containing formatted descriptions of each character into RAM where a hardware character generator can display them. I will focus on programming for such RAM-loadable fonts. I'll explain how to create them and how to apply them on commonly available IBM PC video adapters.

ALPHANUMERIC CHARACTERS

A close look at the characters on your video display reveals that each one is composed of a group of dots. A hardware or software character generator puts the dots on the screen in the patterns that you see as individual characters.

To display alphanumeric characters using an IBM PC video adapter, you must store the code for each character in a buffer in the adapter's video display RAM. Data bytes in the video display buffer represent these character codes. Each 8-bit byte from 0 to 255 (0 to 0FF hexadecimal) is interpreted as a character code that corresponds to a single character in an extended ASCII character set.

Every IBM PC video adapter includes a hardware character generator that displays the dot pattern corresponding to the character codes in the video display buffer. The dot patterns are stored in an 8K-byte table in ROM and are indexed by the character code. Thus, for each character code in the video display buffer, the character generator copies the corresponding dot pattern to the video display screen (see figure 1).

SOFTWARE CHARACTER SETS

Many IBM PC video adapters support one or more *all-points-addressable* (APA) display modes that let you place individual dots directly on the display. Such adapters can interpret the data in the video display buffer as a bit stream that corresponds to the appearance of points on the screen. You can draw characters on the display by setting the appropriate bits in the video display buffer.

[*Editor's note: There are 7 programs that illustrate many of the concepts explained in*

(continued)

Richard Wilton is a software developer with Laboratory Microsystems Inc., 3007 *Washington Blvd., Marina del Rey*, CA 90295.

Table 1: *Support for custom character sets on video display adapters.*

	APA graphics	BIOS support for APA graphics characters	RAM-loadable character sets
MDA	no	no	no
CGA	yes	yes	no
PCjr	yes	yes	no
EGA	yes	yes	yes
HGC	yes	no	no
HGC Plus	yes	no	yes

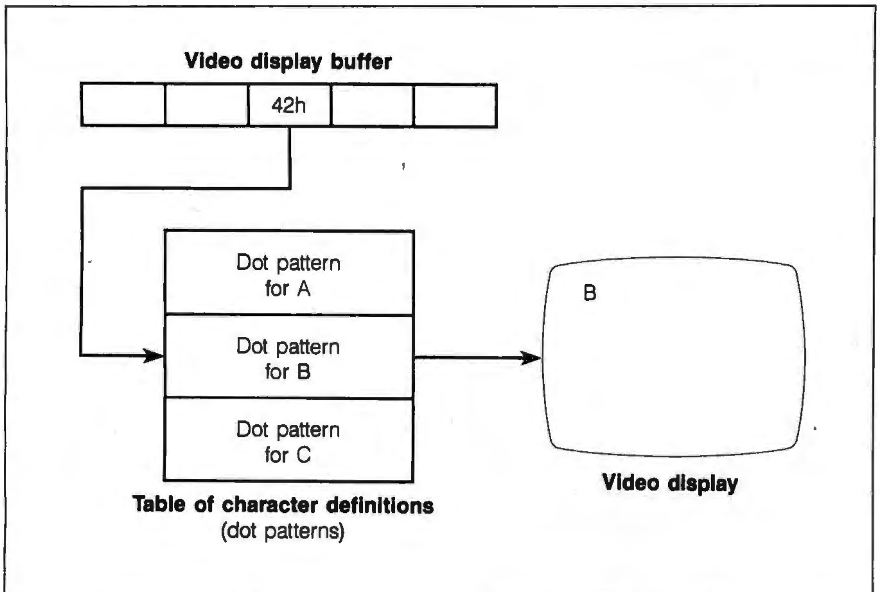


Figure 1: A *block diagram for an alphanumeric character generator.*

this article. They are available on disk, in print, and on BIX. *See the order card following page* 176 *for details. Listings are also available on* BYTEnet. *See page* 4.]

Because it is under software control, this table-driven technique is extremely flexible. The characters you draw can be virtually any size and shape. Using a software character generator is also the best way to mix text with graphics images because you have point-by-point control over arcs, lines, and other graphics elements as well as text characters.

IBM's Color Graphics Adapter supports a software character generator in its ROM BIOS. The routine uses the address in interrupt vector 1Fh [*Editor's note: For the remainder of the article, all interrupt vectors and addresses will be in hexadecimal.*] to locate a character-definition table for ASCII characters 80 through FF. If you store the address of your own character-definition table in this interrupt vector, the BIOS character-generator routine will use your character set for ASCII characters 80 through FF hexadecimal (see the program CSLS2.ASM for an example).

Unfortunately, the BIOS definitions for ASCII characters 00 through 7F hexadecimal are always taken from a table in ROM at address F000:FA6E. Thus, you still need to do some special programming if you want to customize the first 128 ASCII characters, 00 through 7F.

However, IBM's Enhanced Graphics Adapter lets you define all 256 characters for APA graphics. The address of your character-definition table is passed to the BIOS using interrupt 10h, function 10h. When the BIOS is subsequently called upon to display graphics characters, the BIOS routine uses your character-definition table in RAM rather than its default table in ROM.

Whether you write them yourself or use the ones provided in IBM's ROM BIOS, such APA software character generators are much slower than an alphanumeric hardware character generator. One reason is that you might need to write a considerable amount of code to implement a practical character-set generator. Even in assembly language, executing a sizable subroutine just to display a character of text can significantly slow down an application. Also, you have to manipulate much more data in the video display buffer when you draw each character explicitly. Lastly, software runs much more slowly than hardware.

RAM-LOADABLE CHARACTER SETS

A reasonable compromise adopted by IBM on the EGA and by Hercules on the Graphics Card Plus is to implement a hardware character-set generator that can use a table of character definitions that you load into RAM rather than store permanently in ROM. This approach does not provide as much flexibility as the use of a software character generator but preserves the speed advantage of using a hardware character generator.

Using a RAM-loadable character set requires that you accomplish three programming tasks. First, you must create a table containing the dot-pattern definitions for your character set. Second, you must load the character-definition table into RAM. Third, you have to program the video adapter's CRT controller to display your characters.

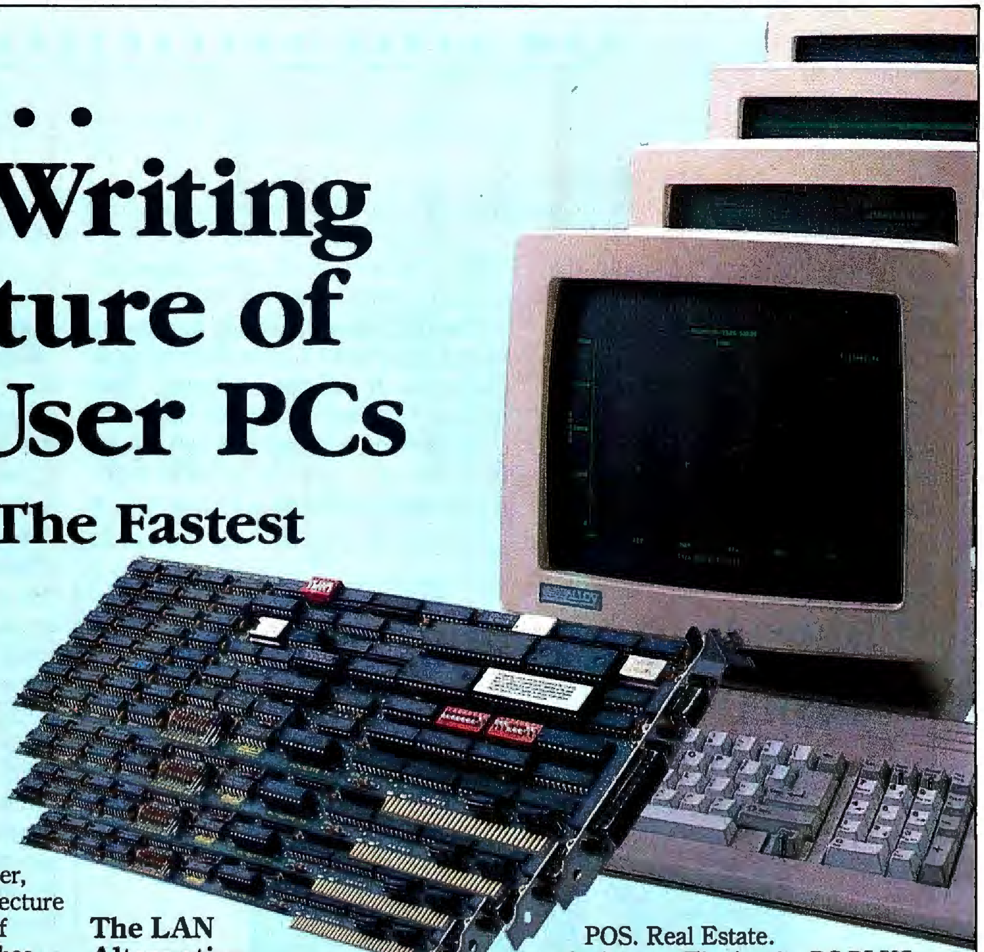
CHARACTER-SET DEFINITION

To define a character set, you create an ordered list of the dot patterns that

(continued)

Alloy . . .
We're Writing the Future of Multi-User PCs

PC-PLUS — The Fastest Growing Multi-user PC Solution



Alloy introduces PC-PLUS — a revolutionary multiuser, multiprocessor architecture for PCs. Amidst the confusion of LANs and time-sharing approaches, RTNX and PC-SLAVE/16 provides a unique, high-performance solution. What LANs promised, PC-PLUS delivered. Faster. Easier to install and use. Less Expensive.

ATNX with its lightning-fast disk caching supports AT-class machines. File and record locking — plus controls to manage file access. Disk volumes greater than 32 Mb. DOS 3.X machines can reach the peak of multiuser performance.

NTNX and PCST/G graphics terminal continue the tradition. NTNX lets PC-PLUS run multiuser applications written for Novell NetWare and MS-NET. PCST/G brings high-resolution, Hercules-compatible graphics to PC-PLUS workstations.

We've written the history of multiuser PCs, and we're writing the future . . . with PC-compatible multiuser systems.

The LAN Alternative

PC-PLUS . . . the perfect solution for sharing data among users. The 8MHz SLAVE/16 card operates at over twice the speed of typical PCs. Advanced bank-switched memory delivers data at bus speed. Many times faster than LANs. Engineering ingenuity makes PC-PLUS simpler and much easier to install than LANs or UNIX multiuser systems. While sharing disks, printers and other peripherals with single-user ease of operation.

The Optimum IBM Multi-user Solution . . . Made Possible by Alloy

The PC-PLUS family turns your single IBM personal computer or compatible into a fully integrated office solution. Up to 31 users can be added to the host — while maintaining full functionality of the PC. As needs grow, add more PC-SLAVE/16 cards and low-cost terminals.

The Choice of Vertical Market Resellers

PC-PLUS expands as business grows. In any vertical market. Law. Finance. Medicine. POS. Real Estate.

And more. That's why PC-PLUS is the fastest-growing multiuser solution among vertical market resellers. Today with a strong commitment to tomorrow.

The PC-PLUS is available from:

ALLOY Computer Products, Inc.

Alloy Computer Products, Inc., 100 Pennsylvania Avenue, Framingham, Massachusetts 01701. (617) 875-6100, TWX: 710-346-0394
Alloy Computer Products, Inc., 9 Executive Circle, Suite 240, Irvine, California 92714. (714) 261-7661
In Europe: Alloy Computer Products (Europe) Ltd., Cirencester, Gloucestershire, England. Tel. 0285-69571, Tlx: 43340
In Australia: Alloy Australia Pty. Ltd., 5 Enterprise Court, Mulgrave, Victoria, Australia 3170. Tel. 011-613-561-4988

make up the characters. Usually, the order in which the characters appear corresponds to the ASCII character sequence. Of course, many applications demand the use of other character-set sequences. For example, you might want to use the EBCDIC character sequence, a foreign-language character set, or even a set of graphics "tiles" that you could piece together to form larger graphics images.

The HGC Plus character generator uses a table in RAM that defines each character in exactly 16 bytes. This means that the maximum size of a character in the table is 8 dots wide (the size of 1 byte) by 16 dots high. If you wish to define a character set in which the characters are less than 16 dots high, you need to pad each character definition out to a total of 16 bytes. When you program the HGC Plus's video controller to display characters that are less than 16 rows high, the extra padding is ignored.

The EGA's character-generator table is similar except that it has a 32-byte-per-character format. This means that character sets that utilize a small dot matrix contain a significant amount of padding. For instance, a table that defines an 8- by 8-dot character set would contain 75 percent padding.

For this reason, the EGA's ROM BIOS provides a routine that loads a table that includes no padding between characters. If your characters are defined in an 8- by 6-dot matrix, for example, each character definition is exactly 6 bytes long. If you save character-set definitions in disk files, this "compressed" format can save a fair amount of space. Also, since the ROM BIOS character-definition tables are stored in a compressed format, you can use the same EGA BIOS routine to load them into RAM as well. (The programs CSLS4.ASM and CSLS5.ASM contain examples of compressed and padded character-definition tables.)

Mapping Character-Generator RAM

You have to know exactly where in RAM to store your character-set definition table. Because the architecture of RAM differs among video display adapters, the way you load your character set into RAM is hardware-dependent.

All IBM PC video display adapters are designed with the video display refresh buffer located in RAM, built onto the adapter itself. When you use RAM-loadable character sets, this on-board RAM is partitioned so that a portion is used to store the character-set definition table. The method of partitioning reflects the underlying architecture of the adapter's on-board RAM.

The HGC Plus can use the 64K bytes of RAM from B000:0000 through

B000:FFFF as a video display buffer. However, when you use a RAM-loadable character set, only the first 16K bytes serve as the display buffer. The remaining 48K bytes, starting at B000:4000, can contain your character-set definition table.

IBM's EGA has a different RAM layout. Individual pixels are represented by corresponding bits in four parallel bit planes. All four bit planes occupy the same range of addresses in the 8088's address space. Each plane is individually addressable by modifying the contents of special registers on the EGA.

The EGA's alphanumeric character generator uses only one of the four parallel bit planes as a display refresh buffer. One of the other three bit planes is then used to store up to 32K bytes of character definitions for the character generator.

LOADING A CHARACTER-SET DEFINITION

Since different video adapters have different hardware configurations of video RAM, you must tailor your font-loading routines to the hardware you use. This task is simpler on the EGA with its fairly complete ROM BIOS support for loading character-definition tables into video RAM. On the HGC Plus, you must write your own software routines to load a font into RAM.

As you have already seen, the EGA and the HGC require that your character-definition table reside at a particular location in video RAM. On the HGC Plus, this location is at B000:4000 (see figure 2a). You can copy the table to this location from somewhere else in memory. (The program CSLS5.ASM shows an example of the copy operation.) If your character-definition data is stored in a disk file, however, you can read the data directly from the file to B000:4000. You can use MS-DOS function 3Fh of interrupt 21h to accomplish this.

Since the HGC Plus lets you use all 48K bytes of the on-board RAM between B000:4000 and B000:FFFF for your table, you can define up to 3072

(continued)

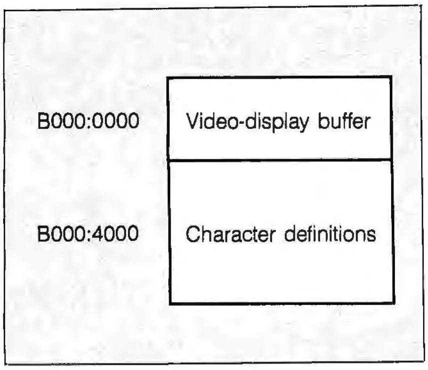


Figure 2a: *Video RAM configuration for the Hercules Graphics Card Plus.*

(49152/16) characters in a single table. However, the IBM PC BIOS and PC-DOS are capable of displaying only the first 256 characters. You can display all 3072 characters at one time only by changing the way character codes are stored in the video display buffer (this will be discussed further under "Extended Character Sets").

You must restrict the size of your character-definition table if your HGC Plus is installed in a system with another video adapter such as the IBM CGA, which uses RAM beginning at B800:0000 (B000:8000). In this case, you must limit your character-definition table to the 16K bytes between B000:4000 and B000:8000. This is done by resetting bit 1 of the HGC Plus's configuration switch (I/O port 3BF). Using only 16K bytes lets you define up to 1024 characters in your table.

On the EGA, you can load your character definitions in up to four RAM locations in bit plane 2 of the EGA's on-board RAM (see figure 2b). The maximum size of each character-definition table is 8K bytes (256 characters times 32 bytes per character). The actual number of character tables you can load at one time depends on the amount of RAM on your EGA: One RAM-loadable character set is supported for every 64K bytes of RAM on the card. Thus, the basic 64K-byte EGA provides enough RAM for only one table, while the fully loaded 256K-byte card supports four character-definition tables at once (see figure 3).

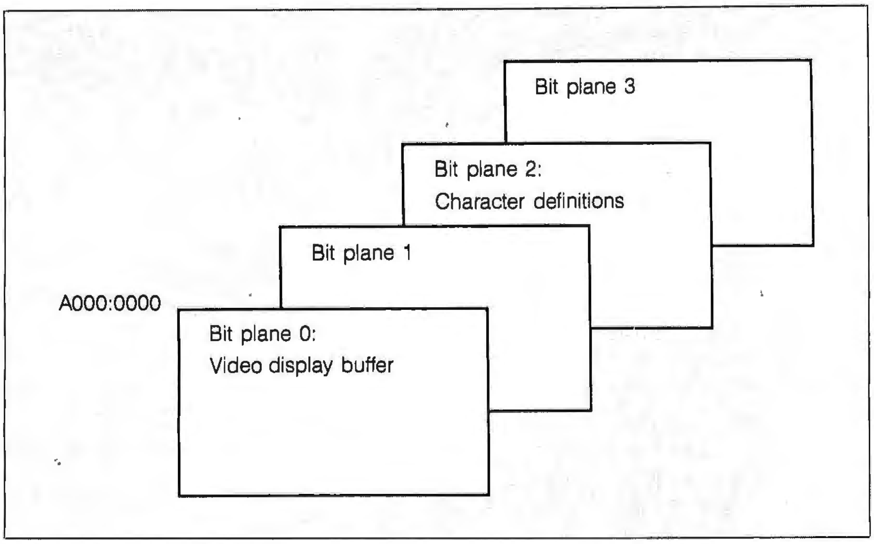


Figure 2b: *Video RAM configuration for IBM's Enhanced Graphics Adapter.*

Starting address		Total EGA RAM
A000:0000	8K character definitions	64K
	(Unused)	
A000:4000	8K character definitions	128K
	(Unused)	
A000:8000	8K character definitions	192K
	(Unused)	
A000:C000	8K character definitions	256K
	(Unused)	

Figure 3: *Bit plane 2 of the EGA's character-set RAM.*

Although you can define only 256 characters in each table, you can maintain up to four such tables in EGA RAM at once and display characters from any two of them at the same time. In other words, you can load as many as 1024 characters into RAM and display 512 of them at once.

The addresses of the 8K-byte tables are at 16K-byte increments in segment A000 (i.e., A000:0000, A000:4000, A000:8000, and A000:C000). Of course, if you want to write to these locations directly, you must selectively enable only bit plane 2 and disable the other three bit planes by programming the adapter's Sequencer Map Mask register. However, you may use EGA BIOS interrupt 10h, function 11h, to copy a character-definition table properly from any arbitrary address in RAM to any of the four available locations in bit plane 2.

PROGRAMMING THE VIDEO CONTROLLER

The third task you must accomplish in order to use a RAM-loadable character set is to program your video dis-

(continued)

Does your laser printer have enough power to produce a stunning document?

Only SoftCraft offers you: 1. Font variety, 2. Formatting power, 3. The ability to create custom characters and symbols.

Armed with Fancy Word, Fancy Font or Laser Fonts your laser printer becomes a high-quality alternative to expensive, time consuming typesetting services. SoftCraft products maximize your laser printer's potential by easily turning your newsletters, presentation materials, documents and invitations into works of art.

In fact, this ad copy was printed using Laser Fonts on an HP LaserJet printer!

1. Font Variety

SoftCraft's complete library contains hundreds of fonts in various styles and sizes from 4 to 72 points. Each product comes with fonts in sizes from 8 to 24 points, including Roman, **Sans Serif**, *Italic*, **Bold**, *Script*, Old English and more.

2. Formatting Power

In addition to all of the conventional formatting commands, *Fancy Font* includes commands that deal with variable-width (proportional) fonts and different height fonts. This is especially useful in setting tables, math formulas and perfectly justified text.

Fancy Word and *Laser Fonts* support all of Microsoft Word's formatting features; no other codes or commands are required.

3. Create Custom Symbols and New Fonts

An included font editing program (optional with Fancy Word) allows you to modify any character and create new characters and logos, up to one inch by one inch. You can also create new fonts, scaled to size, from our database of character outlines.

Choose the Product That's Right For You

Fancy Font: A powerful text formatting and printing program that works with any word processor to provide a wide range of font styles and sizes. You embed simple commands in your document to control Fancy Font's extensive formatting capabilities.

Fancy Word: For Microsoft Word users. You use Microsoft Word like you always have, including the paragraph and character format commands, but you now print all the special characters, styles and sizes.

Laser Fonts: Downloads a large variety of fonts, up to 30 points in size, into laser printers at a significantly reduced price over HP. Laser Fonts automatically configures Microsoft Word to fully take advantage of the downloaded fonts and can be used with any other word processor that can access soft fonts. Laser Fonts can also be used with Fancy Font or Fancy Word to dramatically speed printing on laser printers.

SoftCraft software runs on PCDOS and MSDOS computers with HP, Canon, NCR, Tall Tree and compatible laser printers (Laser Fonts requires a printer with download capability). Fancy Font and Fancy Word are also available for dot matrix printers.

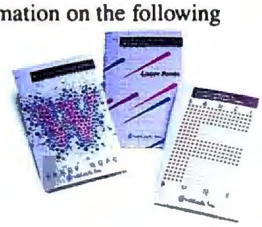
Call TOLL FREE **(800) 351-0500**

SoftCraft, Inc.

222 State Street, Madison, WI 53703 (608) 257-3300

Fancy Font is a registered trademark of SoftCraft, Inc. Fancy Word and Laser Fonts are trademarks of SoftCraft, Inc. Microsoft is a registered trademark of Microsoft Corp.

Please send me information on the following SoftCraft products:



- ☐ Laser Fonts
- ☐ Fancy Word
- ☐ Fancy Font

NAME ____________________

ADDRESS ____________________

CITY, STATE, ZIP ____________________

PHONE ____________________

PRINTER MAKE ____________________

MAIL TO:
SoftCraft, Inc., 222 State St., Madison, WI 53703

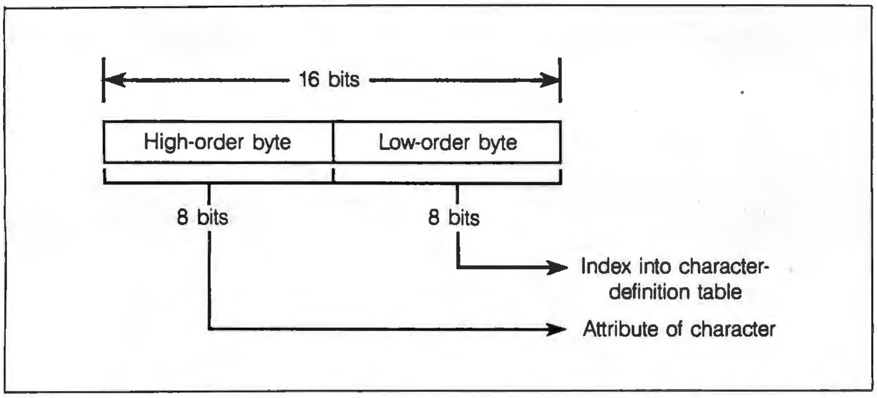


Figure 4a: *The video display buffer mapping of the alphanumeric modes of* IBM's Monochrome Display Adapter, CGA, *and* EGA, *and the Hercules Graphics Card Plus.*

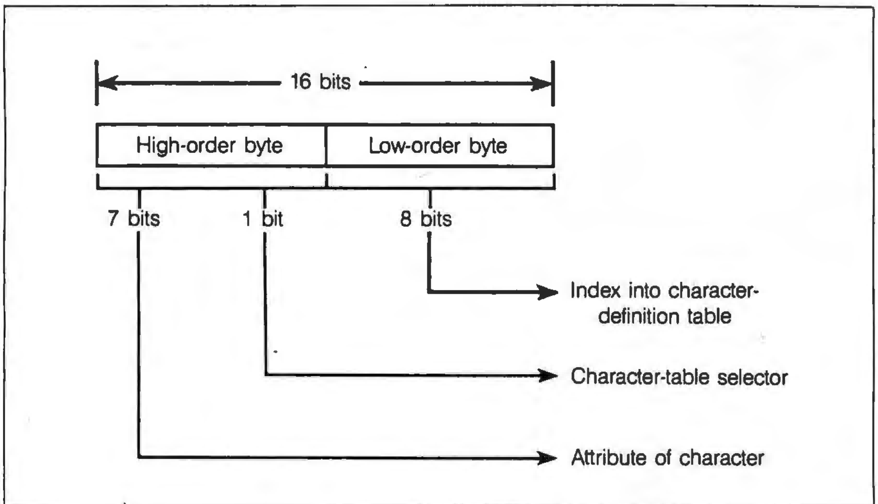


Figure 4b: *The video display buffer mapping of* IBM's EGA *extended character set. Bit 3 of the high-order byte selects one of two 256-character tables.*

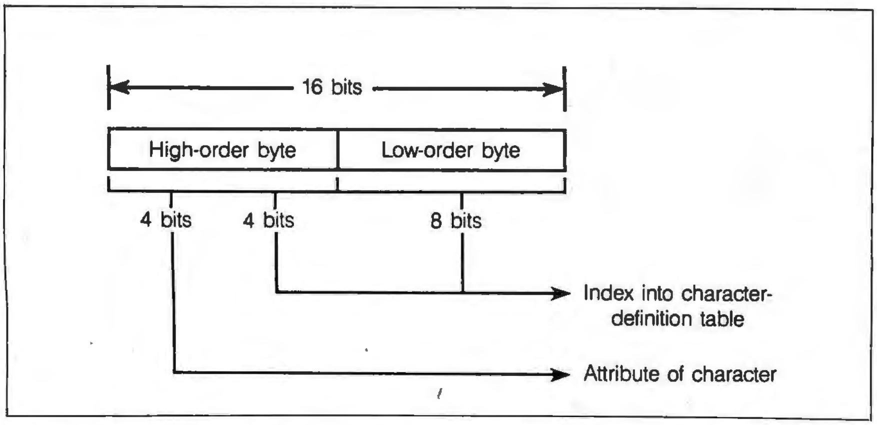


Figure 4c: *The extended character set of the* HGC Plus *video display buffer map. Bits 0 to 7 of the low-order byte are appended to bits 0 to 3 of the high-order byte to form a 12-bit index into the character-definition table.*

play adapter's CRT controller. On the EGA, the CRT controller circuitry resides in a special LSI chip with 27 internal registers. The HGC Plus uses a Motorola 6845 plus a proprietary chip that adds an additional 3 registers for controlling RAM-loadable font configurations.

The way you configure the CRT controller on the EGA and the HGC Plus depends primarily on the height of the characters in your character set. Hercules provides a table of register values for its CRT controller. Your program must explicitly load each of these values into the proper CRT controller registers.

Similarly, the EGA's CRT controller must be programmed to reflect the character height. In particular, the Maximum Scan Line, Cursor Start, Cursor End, Vertical Display Enable End, and Underline Location registers must be updated. Again, your task is simplified by an EGA ROM BIOS routine (interrupt 10h, function 11h). The BIOS routine calculates the correct register values given the number of bytes per character in your character-definition table and programs the CRT controller accordingly.

NINE-DOT-WIDE CHARACTERS

Although the height of the characters you define may vary, you might expect their width to be restricted to eight dots, since this is the number of bits in each byte of your character-definition table. However, both the HGC Plus and the EGA can be programmed to put 8-dot-wide characters on the screen in a 9-dot-wide matrix. The extra space between characters improves their appearance on a green IBM monochrome display.

To do this on the HGC Plus, you must set bit 1 in CRT controller register 14 hexadecimal. On the EGA, you reset bit 0 of the Sequencer Clocking Mode register. The EGA BIOS routines set this bit for alphanumeric display modes on a color monitor (8-dot-wide characters) and reset the bit for a monochrome monitor (9-dot-wide characters).

A subtle built-in function of both

(continued)

adapters relates to the use of the block-graphics characters, specified with character codes B0 through DF hexadecimal in IBM's PC character set. You can concatenate these characters to draw horizontal lines on the display. When they display characters in a 9-dot-wide matrix, the EGA and the HGC Plus copy the rightmost dot of all block-graphics characters into the extra space between characters. This ensures the continuity of horizontal lines.

EXTENDED CHARACTER SETS

When 8-bit codes represent characters in the video display buffer, a maximum of 256 (2^8) different characters can be displayed at one time. To display more characters than this, you must find a way to use character codes that are larger than 8 bits.

When IBM PC video adapters display text, 16 bits (one word) of data are actually stored in the video display buffer for each character on the screen. Usually, 8 of these bits encode the ASCII character number, while the other 8 bits specify attributes such as the character's color, intensity, blinking, or underlining (see figure 4a). The HGC Plus and the EGA use one or more of the 8 attribute bits to extend the range of character codes and thus increase the number of displayable characters.

On the EGA, the character-generator hardware can use bit 3 of the high-order byte to toggle two 256-character tables (see figure 4b). Which two of the four possible tables are actually used is determined by programming the EGA's Sequencer Character Map Selector register. Although it is possible to program this register directly, EGA ROM BIOS function 11h will perform this function for you.

On the HGC Plus, the character-generator hardware can use the low-order 12 bits of the 16-bit word as an index into the character-definition table at B000:4000 (see figure 4c). The character generator interprets character codes in this way when you set bit 2 of CRT controller register 14 hexadecimal. (The program CSLS7.ASM demonstrates this technique.)

In order to use the extended character sets on either adapter, you must write software routines to store character codes and attributes in the display buffer in the appropriate format. Unfortunately, software that uses extended character sets on the EGA will not format character codes and attributes correctly for the HGC Plus, and vice versa. Furthermore, all the IBM PC's BIOS routines assume that the video buffer is formatted for only 256 character codes (as in figure 4a). Programs that perform video output with BIOS calls or with PC-DOS calls (which in turn call the BIOS) will not

(continued)

run properly if your adapter is configured to display an extended character set.

UPDATING BIOS RAM

When you change the height or width of your character set, you implicitly change the screen's logical dimensions. The usual screen dimensions are 80 characters across by 25 characters down. Consider what happens when you use 8 by 8 characters on a screen that usually displays 9 by 14 characters, such as the green monochrome display: Your screen now displays 90 characters across and 43 characters down.

Although the ROM BIOS and PC-DOS are not always flexible enough to function correctly with screen dimensions other than 80 by 25 or 40 by 25, you can preserve a great deal of function by updating the status values stored by the BIOS in segment 40. In particular, several different BIOS routines use the words at 0040:004A (CRT_COLS) and 0040:004C (CRT_LEN).

PROBLEMS AND PITFALLS

I've alluded to one obvious problem with the use of RAM-loadable character sets: namely, that neither PC-DOS nor the ROM BIOS was designed to handle character sets of different-size characters. Also, neither DOS nor the BIOS can support the modified video buffer format required for extended character sets. When the usual DOS and BIOS video output routines are used with an extended character-set configuration, garbage characters appear on the display. The only solution is to be certain that an appropriate character set is available when control is transferred out of your application to DOS or to a BIOS video output routine.

A related problem crops up in a windowing or multitasking environment when a coresident program or an alternate task takes control of the display. Your program's character set and CRT controller configuration might be incompatible with those expected by some other task. At present there is no agreed-upon way to coordinate character set and video status between tasks. The problem might be solved with some type of global video status table or by a memory-resident display manager, but any such approach would have to be agreed upon by software developers to become usable.

Despite such potential pitfalls, the availability of RAM-loadable character sets in moderately priced video hardware should encourage their use by software developers. The reasonable compromise between the flexibility of dot-addressable graphics and the speed of a hardware character generator provides a way to improve the user interface in many applications. ■

CMO. Doing business-to-b

Person to person.

Not too long ago, I got a call from a new customer with an unexpected problem—we had delivered his order too quickly. He had placed a COD order with one of our sales consultants on the 14th. It went through our system and was shipped the 15th. When the shipment arrived, the customer called. He said he hadn't expected it so soon and hadn't had time to arrange for payment.

It turned out the customer had worked as a sales rep for a competitor of ours. He said it takes them 3 or 4 weeks to complete most orders. Consequently, he didn't think there was any hurry to get a check ready for the CMO order.

"They must have a lot of cancelled orders," I said.

He said yes, that in a typical month they only ship 40% of what's ordered.

That surprised me. I told him we ship 90% of our orders the next day. And we have very few cancellations.

Well, everything worked out. I apologized for our promptness. He laughed and said he'd be ready for us the next time he ordered.

And we'll be ready for him.

CMO is committed to its customers. We operate an IBM System 38 with state-of-the-art software to help expedite the tens of thousands of orders we receive. Thanks to this system CMO is able to provide customers with up-to-the-minute information about their orders.

Thomas A. Penfield
Sales Manager, CMO

SOFTWARE.

Ansa	
Paradox	$499.00
Ashton-Tate	
d-Base III +	429.00
Framework II	429.00
Borland	
Reflex	59.99
Lightening/Word Wizard	99.99
Central Point Software	
Copy II PC	24.99
5th Generation	
Fastback	see special
IMSI	
Mouse w/Dr. Halo	119.00
IUS-Sorcim	
General Ledger	299.00
SuperCalc III	329.00
Super Project Plus	299.00
Lifetree	
Volkswriter III	149.00
Lotus	
Lotus 1•2•3	329.00
Symphony	449.00
MECA	
Managing Your Money	119.00
MicroPro	
Wordstar 2000 +	299.00
Wordstar Professional w/GL Demo	189.00
Microrim	
R:Base 5000	359.00
Micro Soft	
Micro Soft Word	289.00
Micro Soft Mouse/Serial	139.00
Micro Soft Mouse/Bus	139.00
Microstuf	
Crosstalk XVI	89.99
Multimate	
Multimate 3.3	229.00
Multimate Advantage	289.00
Norton Software	
Norton Utilities	57.99
PFS	
First Choice	99.99
Professional Write	129.00
Professional File	159.00
Satellite Systems	
Word Perfect 4.1	219.00
The Software Group	
Enable	369.00

MONITORS.

Amdek	
Color 600 RGB, IBM CGA	$399.00
Color 722 RGB, CGA/EGA	499.00
Color 730 RGB Analog	699.00
Magnavox	
8CM515 RGB Monitor 80	289.00
7BM623 PC Monitor 80	99.99
NEC	
JC-1401P3A Multi-Sync	599.00
JB-1280 TTL Green	139.00
JB-1285 TTL Amber	139.00
Princeton Graphics	
MAX-12 Amber	169.00
HX-12 16 Color	429.00
HX12e 64 Color	509.00
SR-12e Super Hi-Res	589.00
SR-12e RGB Analog	649.00
Taxan	
640 RGB Hi-Res	539.00

MODEMS.

Anchor	
Expressi 1200 Half Card	$149.00
Everex	
Evercom 920 1200 Baud	139.00
Hayes	
Smartmodem 1200B	359.00
Smartmodem 2400	599.00
Smartmodem 2400B	539.00
Practical Peripherals	
Practical Modem 1200SA	179.00

DRIVES.

CMS	
Drive Plus 20 MB Card	$499.00
T-120 20 MB Tape-Backup	469.00
T-160 60 MB Tape-Backup	699.00
Iomega	
A210H 10+10 Bernoulli Box	1899.00
A220H 20+20 Bernoulli Box	2499.00
Mountain	
Drive Card-20	689.00
Drive Card-30	949.00
Priam	
Innerspace ID-60AT	1349.00
Seagate	
ST-4026 20 MB Hard Drive	689.00
ST-4038 30 MB Hard Drive	739.00

CMO

GET A TAILOR-MADE, 20 MEG, XT-COMPATIBLE FOR AS LITTLE AS $999.

Presenting PC TOO.™ Only from CMO.

When you think of tailor-made, you think expensive, right? Think again.

Now CMO can offer you a 20 Mb PC-compatible—one we assemble in our Pennsylvania lab just for you. To your specifications. With the attention to quality and performance you've come to expect from CMO. All for just $999.

Every tailor-made clone we make is backed with a full 90-day warranty on labor and parts, a 30 day money back guarantee and easy access to CMO technical support after the sale—the kind that keeps you up, running and satisfied.

For $999, you get: A 20 Mb, PC-XT compatible that runs thousands of programs, including Lotus 1-2-3™ ■ A 4.77 Mh chip speed ■ An XT chassis with 256K (expandable to 640K) ■ A 20 Mb hard drive ■ A 360K floppy drive ■ A floppy disk and hard drive controller ■ A 150 watt power supply (not just 130) ■ Open architecture with 8 expansion slots (not just 7) ■ And an 84-key, XT-style keyboard.

Need more power and more flexibility? Call us. Let's talk about your tailor-made PC TOO.

Your options are unlimited.

CMO. Policies and particulars.

Next day shipping on all in-stock items. • Free technical support from our own factory-trained staff. • Toll-free order inquiry. • No surcharge on VISA or MasterCard orders. • Credit cards will not be charged until the order is shipped. • No limit and no deposit on COD orders. • No sales tax on orders shipped outside PA. • No waiting period for cashier's checks.

SHIPPING. Add 3% (minimum $7.00) shipping and handling on all orders. • Larger shipments may require additional charges. • Returned items may be subject to a restocking fee. • All items subject to availability and price changes.

EDUCATIONAL INSTITUTIONS: 1 800 221-4283

CMO

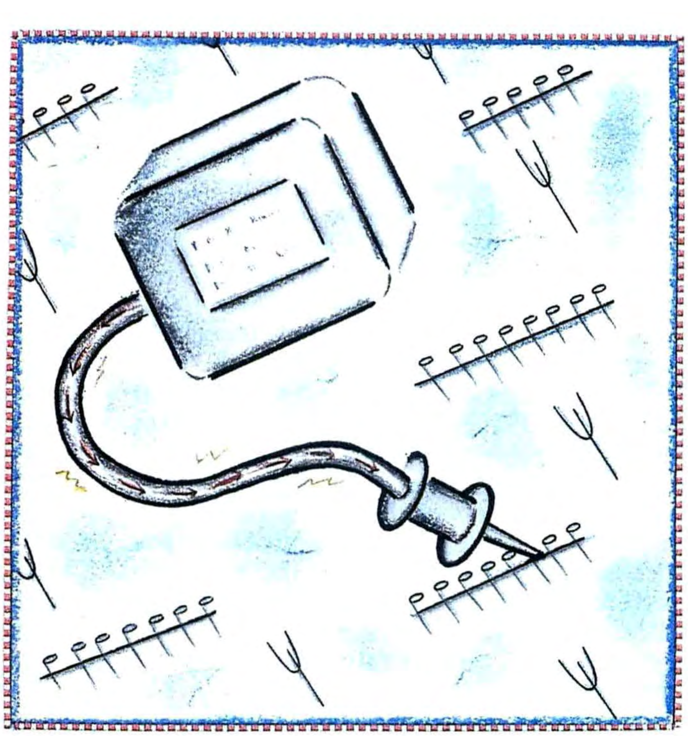
When you want to talk computers.
When you want to talk price.
Call toll-free: 1 800 233-8950.

Outside the U.S.A. 717 327-9575 Telex 5106017898
CMO. 477 East Third Street, Dept. A113, Williamsport, PA 17701
All major credit cards accepted

SPEEDING UP THE PC AT

Without spending a fortune, you can dramatically increase the AT's potential

BY BRIAN K. ROEMMELE



YOU CAN INCREASE the standard 6-MHz CPU speed of the IBM PC AT to 8 MHz with relatively low-cost modifications and replacement parts. All the critical components are socketed and easy to replace. You can obtain speeds faster than 8 MHz after testing and identifying the components that are not capable of higher speeds. How involved the speedup is depends on how fast you want your AT to go. The modifications can range from a simple crystal change to changing the CPU, RAM, and clock-generator chip as well as the crystal. Also, you will need to consider the speed limitations of any expansion boards you have in the system. I will explain how you can speed up your AT and describe the possible negative effects. [*Editor's note: See the "Benchmarks" text box for the results we got when speeding up one of* BYTE's ATs.]

It is rare to find a system that has so much upgrade potential with so little cost investment. The generous performance margins that Intel built into its chips, along with the predictable quality of the components on the AT's motherboard, let you increase performance without causing problems. Also, evidence suggests that the AT was originally designed for an 8-MHz CPU speed. First, the crystal is socketed rather that soldered, which is unusual for microcomputers. Second, in the schematics in the IBM *Technical Reference Manual* (Part #1502494, first edition, March 1984) the SysClk, ProcClk, and PClk traces all reference an 8-MHz bus, system frequency, and a 16-MHz crystal.

CRYSTALS

With a new crystal, you can increase the speed of your AT to 8 MHz. But the crystal you buy must be of high quality. The socket used to hold the AT's crystal requires an HC 25– or HC 33–type container. It is tempting to buy a low-cost wire-leaded HC 18 type and solder on nonpliable pins so you can mount it into the socket. In my tests, this method proved to be unreliable over the long term.

Standard HC 18– and HC 25–type crystals that you can find at local electronics stores might not be able to produce the stable and clean output many chips need to operate above the manufacturer's specifications. This is important for reliable operation when running the AT above 8 MHz.

You should buy an HC 25 military-specification-type crystal because of its stable frequency over temperature —better than 0.0005 percent +/– frequency deviation. The mil-spec crystals can cost twice as much as the lower-quality crystals and might be more stable than necessary, but the more stable frequency source will prevent damage to system components.

80286 MICROPROCESSOR

The 80286 is one of the most critical components when you are increasing the AT's system frequency. Intel's testing system works with a 35 percent upper performance margin when related to temperature, current, and frequency capabilities. Intel designed the 80286 to ultimately yield 10-MHz devices in volume production, but in early production runs Intel was able to produce only 6-MHz devices; this might be why the AT came out as a 6-MHz machine. Intel rates the 80286 by inducing an 85°C case

(continued)

Brian K. Roemmele is the president of Ariel Computer Corporation (Box 866, Flemington, NJ 08822, (201) 788-9002), *maker of an* IBM PC AT *speed-up kit.*

ILLUSTRATION BY ALEXA GRACE

temperature with a stable and clean power source. The chip's resulting coefficient at 3.3 power dissipation will qualify its frequency capabilities. The AT's power supply produces a relatively stable, clean current. The surrounding ambient temperature (35 to 52°C) within the AT's case is rarely close to the 85°C case temperature used to rate the 80286.

Considering Intel's test safety margin and the CPU's relative temperature, 8-MHz operation from the 80286-6 is safe. On average, consider 9.5 MHz to be the maximum frequency of the 6-MHz device. The radiated temperature from the 80286-6 running with an 8- to 9-MHz clock stabilizes well below the chip's maximum specifications. Although I successfully ran some ATs with stock 80286-6s at 10 MHz, I would not recommend this unless you monitor the CPU's temperature and check the system regularly.

In the tests I ran, I saw no evidence of lower mortality of the 80286-6, provided it receives a reliable, stable clock frequency. The gates that directly see the input frequency could theoretically suffer unpredictable performance or damage from wide and continuous frequency shifts. The damage is usually to the gates of the chip. If you don't want to worry about running in the safety margins of an 80286-6, you can replace the CPU with an 80286-8 or 80286-10. Since this chip is socketed, it is a simple upgrade.

82284 Clock Generator

The Intel 82284 clock generator divides the raw frequency from the crystal or frequency synthesizer in half, which stabilizes the frequency. The AT's standard 12-MHz crystal yields a 6-MHz processor and system frequency. The same test rating performed on the 80286 is performed on the 82284, resulting in the same coefficients. Intel designed this device to yield 8- to 10-MHz devices in volume production. In the AT, you can easily expect 10-MHz operation from this chip. In some ATs this chip is socketed, and you can replace it with a faster device. The chip will need replacing if you want to bring the AT above 11.5 MHz.

RAM Chips

The AT waits one CPU cycle before it performs a read or write to the RAM or a read to ROM. This wait state is wired in by the hardware and does not increase when you increase the CPU frequency. In fact, the period of wait decreases proportionally to the increase in frequency. You can approximately calculate the needed RAM nanosecond rating by dividing two into the CPU frequency (e.g., 2/6 = .3333, or 333 nanoseconds).

To arrive at an accurate result, you need to include the wait state (1/CPU frequency) and overall capacitance and propagation delays. IBM's *Technical Reference Manual* lists all RAM as 150 ns or better for the system board. The standard 128K-bit piggy-back 150-ns RAM is adequate up to a 9.5-MHz CPU speed. At 10 to 11 MHz, 120 ns is required; above that, 100-ns RAM should cover any other frequency that the rest of the system can handle. Because of increased capacitance and propagation delays, the RAM that populates memory-expansion cards should be at least 120 ns at frequencies above 8 MHz.

80287 Math Coprocessor

The standard 80287 math coprocessor receives one-third of the actual crystal frequency. On the standard 6-MHz AT, this works out to 4 MHz. For system frequencies below 10 MHz, the normally heat-sensitive 80287 should not experience any dramatic temperature increases. With a system speed of 10 MHz, the 80287 is running at 6.6 MHz, which is far enough beyond the Intel 5-MHz specification that an 80287-8 should be used.

Various Chips

The 74ALSxxx-series TTL logic chips covering approximately 60 percent of the system board and the Intel 82288 bus controller are the final limiting factors after you have considered or replaced all other components. They are both capable of approximately 11.5-MHz operation, but beyond that you should replace them. This is difficult because they are soldered into place. The extra 1 or 2 MHz you could gain might not be worth the heat and circuit-board damage that you could cause during desoldering.

RAM Boards

The quality control enforced by IBM on the chips that form the system board is predictable. After months of observing and testing, I have arrived at a reliable success rate in speeding up ATs. This is not the case for boards supplied by outside vendors. Some slow boards, just by being connected to the system bus, could cause the whole machine to lock up. RAM boards are particularly sensitive. To date there are no Lotus/Intel/Microsoft Expanded Memory Specification memory boards that run at or above 9 MHz. Some are not even capable of running at 8 MHz.

Most memory-board limitations can be traced to PAL (programmable array logic) chips, slow delay lines, and slow 74LSxxx-series chips. Standard RAM boards are more likely to be compatible. For example, the IBM 128K-byte memory board and the AST Advantage can run at 10 MHz, although the manufacturers don't support these capabilities. I have used the AST Advantage in ATs running at 12 MHz populated with 100-ns RAM chips. Other RAM boards are capable of fast CPU speeds. Working with the manufacturer or dealer in a trial-and-error process to find a suitable board would be the best way to identify boards capable of operating at high speed.

Monitor Boards

Monitor boards, like memory boards, suffer from wide variations in the system frequencies at which they can operate. Some older boards designed specifically for the PC might not work correctly in even a 6-MHz AT. Most monitor boards now manufactured specify their maximum system frequency capabilities. All of IBM's display cards can operate at up to

10-MHz CPU speeds. Monitor boards rarely reference the CPU frequency.

OTHER BOARDS

Printer and RS-232C ports, if designed to IBM's recommended specification, do not limit the AT's maximum frequency capabilities. A few low-priced imported boards have failed in 8-MHz systems. They interface to the system bus with slow 74LSxxx TTL chips. Timing (data rates) for the printer and RS-232C ports is usually derived from an autonomous frequency source unrelated to the CPU frequency.

Modem boards are a little more sensitive to a faster system. All Hayes internal modem products perform correctly at 10 MHz. Older or low-quality modem boards seem to fail above 8-MHz CPU speeds. Most boards designed for AT-class systems should not show any major problems; when in doubt, contact a knowledgeable engineer at the manufacturer.

FLOPPY AND HARD DISK DRIVES

The fixed disk/floppy disk controller card that comes with the AT can run at 12.5-MHz. The hard disk drives I have tested work fine at faster CPU speeds. However, the 1.2-megabyte floppy disk drive experiences some errors at or above 9 MHz.

Approximately 45 percent of the systems I tested reported a "Drive not ready error" during first-time drive access or during a copy or format. In many cases this error can be traced to an optimization for 8-MHz CPU and bus frequencies in the BIOS and PC-DOS 3.1 and 3.2.

PC-DOS and the BIOS set up a timing loop to wait for the drive to respond after being prompted with a signal. The loop time decreases while the drive's reaction time stays the same, causing the drive to sometimes fail to report "Drive ready" to the DOS/BIOS. Fortunately, you can press R for retry until the drive is ready. [*Editor's note: Ariel sells a patch to the* BIOS *to watch for the drive log-on.*]

HARD DISK INTERLEAVE

If you increase the CPU speed above 8 MHz, you can change the hard disk interleave. The 3:1 interleave on the standard IBM-supplied 20-megabyte drive is optimized for 6- to 8-MHz CPU speeds. At a CPU speed of 9 MHz or faster, a 2:1 interleave is better. You can change the hard disk interleave with IBM's Advanced Diagnostics found in the *Hardware Maintenance and Service Manual* (Part #1502242). You can expect a performance increase of 30 to 50 percent from a drive with a 2:1 interleave and a CPU speed of 9 MHz or above.

SOFTWARE

Software for MS-/PC-DOS systems rarely intentionally uses the CPU frequency as a critical reference. Theoretically, software that assumes a particular CPU speed could pose an uncorrectable problem, although you can readjust some software defaults that assume a particular CPU speed. For example, in some terminal-emulation programs you might need to increase the time-out default before carrier detect in auto-dial mode.

Timed screen sequences and timed screen menus might need adjustment, if it's possible. Some copy-protection techniques assume a particular system speed and will create a problem. Ashton-Tate seems to be the only company using this form of time-based copy protection.

ERRORS

The AT's POST (power-on self-test) investigates all the critical points of concern when increasing the CPU frequency. Running the standard IBM AT diagnostics disk continuously for the first 5 to 10 minutes after finalizing on a frequency should magnify conditions that the POST might have missed. During the testing of the monitor card in some systems, diagnostics might report a 501 error. This is due to IBM's technique for testing a specific function on the monitor board and does not represent any failure of the hardware.

Table 1 lists some of the errors that POST and diagnostics report, with a brief explanation and recommendation for correction. If the system fails to respond at less than 8-MHz CPU speeds, it is most often traced to a

(continued)

Table 1: *Some errors that POST and diagnostics report if some component fails when you increase the* IBM AT's *processor speed.*

Error	Description
Parity check 1 and 2 error	The RAM is too slow in responding to the CPU access cycle (install 120- or 100-ns RAM chips).
164 memory-size error	Usually attributed to an expansion RAM board that cannot handle the current CPU frequency.
201 system-board error	A general error—it usually means that under this system setup the current CPU speed is too fast for various chips on the system board (difficult to correct). It might also be an 80286 that is borderline failing.
System activity but no information on your screen	Generally relates to an incompatible display board.
No system activity or display	This is usually an 80286 failure. It can also represent a failure of the various chips on the system board.
Drive not ready error (abort, ignore, retry)	This can be traced to the BIOS, DOS, and the drive itself. You can retry until the drive is ready.

BENCHMARKS

BY EVA WHITE

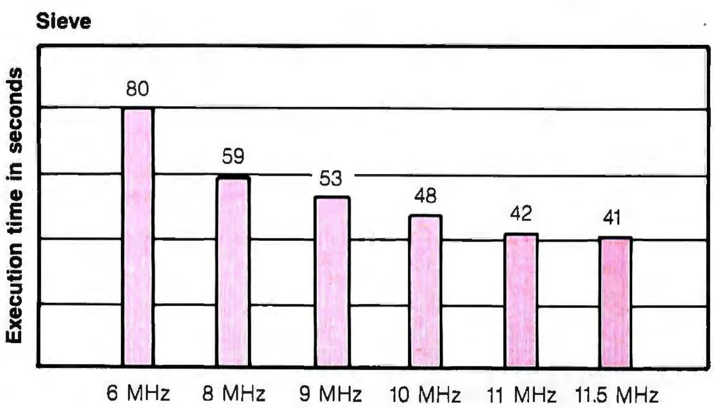
I tried some of Brain Roemmele's techniques to speed up one of BYTE's IBM PC ATs. I wanted to run benchmarks at six different speeds, so rather than buy six crystals, I opted for Ariel Corporation's variable frequency synthesizer (VFS), XCELX. It replaces the 12-MHz crystal in the AT.

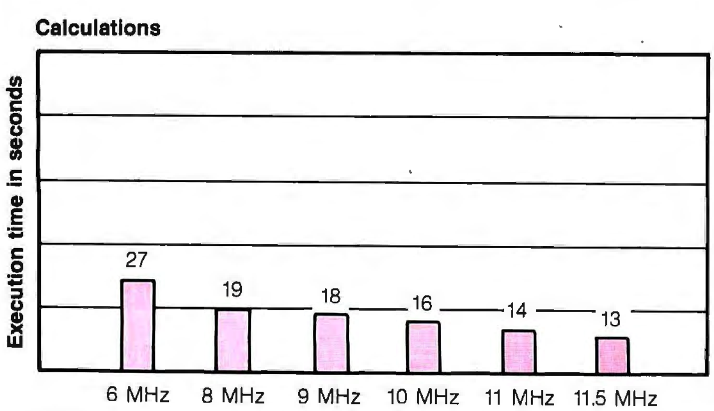
The VFS is a silver box (about 2 by 1 by 1 inch) that you mount with Velcro to the AT's power-supply case. Three wires and a control panel with variable resistor and toggle switch come out of the frequency synthesizer box; the red wire plugs in beside the wire in pin 1 of the P58 power connector, the black wire plugs in beside the wire in pin 1 of P59, and a wire with a hook-clip is attached to one of the leads of the empty crystal socket. The variable resistor lets you adjust the CPU speed, and the toggle switch engages either the standard (6 MHz) operation or the variable-speed CPU operation. The panel mounts in the rear of the AT (between the card slots and the fan). The control panel is bolted in place of the utility plate. The control knob and toggle switch stick out the back of the AT's case, preventing you from replacing the plastic rear cover. An adhesive label goes over the control panel to give you approximate CPU speed. A program included with the XCELX software called Tuneup gives a more accurate value for the CPU speed. I used it to set the CPU speed for the benchmark tests.

The CPU speed knob was too easy to turn. If you had to reach behind the machine, you could accidentally brush against the knob, turn it past the speed that the machine could handle, and cause a crash. To be on the safe side you can flip the toggle switch to standard mode, adjusting any cables behind the machine.

The Benchmarks for a standard AT were identical to those I ran with the synthesizer in standard mode (6 MHz). I flipped the toggle switch and was able to crank up the standard AT to 9 MHz before the machine froze. Then I replaced the CPU with an 80286 rated at 12.5 MHz and replaced the standard RAM with 100-ns RAM, both supplied by Ariel Corp. I ran the Sieve and Calculations programs at different speeds; the results are shown in table A. (See *Inside the IBM PCs*, Fall 1985, pages 200–201 for the listings of the Sieve and Calculations programs.) The speed increase going from 6 to 8 MHz was dramatic. While speeding up the machine beyond 8 MHz results in a noticeable decrease in execution times, it becomes less attractive when you have to replace the CPU and memory. Still, it's neat to see the AT running at 11.5 MHz. So far our speeded-up AT has been running the Sieve continuously for four days and is still working fine.

Table A: *Sieve and Calculations benchmarks run on an* IBM PC AT *at various* CPU *speeds. The benchmarks at* 9 MHz *and below were performed on an* AT *with all stock equipment except for a frequency synthesizer. For those tests at* 10 MHz *and above,* 512K *bytes of* 100-*ns* RAM *and a* 12.5-MHz 80826 *were installed. The* 80287 *was not accessed during these tests.*





Eva White is a BYTE *technical editor. She can be reached at* BYTE, *One Phoenix Mill Lane, Peterborough,* NH 03458.

memory or monitor board not designed for anything above 6 MHz.

DAMAGE

With all the components I have mentioned, permanent circuit or chip damage is unlikely when you are increasing the CPU frequency. Provided the frequency is reliable and stable, the worst that you can expect is a reported error or the system refusing to come up altogether. A chip either fails entirely or does not work normally. Heat-related damage might occur from prolonged use of a borderline device or use of an unstable frequency reference. The borderline problem is possible but unlikely. By selecting a lower frequency when you suspect this, you will make damage less likely.

IBM'S WARRANTY

As of this writing, IBM has made no official statement on the effect that increasing the system frequency will have on the company's one-year limited warranty. Officially, IBM doesn't encourage any activity of this type; unofficially, many IBM engineers and salespeople have been performing and recommending the upgrade. The dealer seems to have the final word on the warranty question.

IBM'S NEW ROM

The Model 239 30-megabyte hard disk system is not capable of increased CPU speeds with just a crystal replacement. Because of the popularity of increasing the CPU frequency, IBM chose to alter the ROMs in the new system. The affected ROMs have date codes beginning in June and July 1985. The alteration is a time-based loop in the first 500 lines of code in the BIOS's POST portion. The loop will determine if the CPU speed is greater than 6.6 MHz. If it is greater, the code will perform a branch that will purposely seize the system. The system will not beep or report any errors. (Newer ROM releases will produce two beeps.) I had considered a sophisticated ROM patch using mapped-in RAM or ROM to the code area. However, such a solution is complicated, inelegant, and would have introduced more vulnerability to the system as a whole.

I have come up with an easy fix for the ROM check. If you maintain a 6-MHz frequency to the 80286 for at least 10 seconds after power-up, the POST sees the standard AT frequency. Slowly and uniformly increasing the frequency to the system's maximum speed circumvents the ROM's investigations. A crystal switch selector would lock up the system if switched while the system was running. Debouncing only marginally im-

(continued)

proves the crystal switch design. A variable-frequency synthesizer proved to be the only reliable option.

It was relatively easy to design a frequency synthesizer that will automatically handle the procedures for getting around the ROM check. The timing loop for detecting an increase in system frequency is activated only during power-up and during a Ctrl-Alt-Del key combination. If you add an extra connector to a chip that receives this interrupt, the frequency synthesizer resets so that the required 6-MHz CPU speed is maintained during restart of POST.

Unlike crystals, which are passive devices, the frequency synthesizer is an active device that produces a continuum of frequencies; it allows increasing and decreasing of frequency while the system is running. This makes it easy to adjust the system for top performance. A variable resistor-based circuit offers the widest possible range of frequencies from which to choose.

NEW 8-MHZ AT

As I write this, IBM is beginning to ship the new Model 339 and 319 8-MHz system. It uses the same Type 2 system board found on most Model 239s. This board requires 256K-bit memory chips rather than the 128K-bit devices the Type 1 board used.

The differences are that the CPU is an 8-MHz marked part, the crystal is a 16-MHz device, and the ROMs (date beginning November 1985) have a loop that senses the CPU speed and causes them to seize on frequencies above 8.6 MHz. I have used a frequency synthesizer in an unmodified system up to and above a 10-MHz CPU speed.

CONCLUSIONS

If you are using an IBM PC AT with 512K bytes of RAM, a 20-megabyte fixed disk, and an IBM monochrome adapter, you can replace the 12-MHz crystal with a 16-MHz crystal without replacing other components. You might even be able to push the 80286-6 to 9 MHz with an 18-MHz crystal. Using a 20-MHz crystal, you would need to replace the CPU. With a 22-MHz crystal you would need to replace the CPU and RAM. At 24 MHz, you would need to replace the CPU, RAM, 82284 and 82288 chips, and possibly some 74ALS TTL chips.

Electronics manufacturers test and rate their products to perform predictably and reliably within their stated guidelines. Normally you should not design products that use components close to or above the manufacturer's specifications. Under most circumstances, I believe this is a good rule to follow, but the AT has a large safety margin; I have found that the benefits of improving its speed greatly outweigh the potential drawbacks. ■

"...masterpiece of IBM imitation...minor masterpiece of the circuit designer's art... impressed by its high standards of quality and workmanship."
Winn L. Rosch, Cloning Your Own PC, PC Magazine, July 10, 1984.

"The most compatible IBM clone I've ever worked with."
Lee Konowe, American Software Club, Ridgefield, CT.

"Fullfills our every need...and leaves us money for software."
John Glenn, Electric Engineer, U.S. Navy.

Display Telecommunications Corporation was named *Best Small Business Exporter in Texas for 1986* because we offer the best IBM® PC clone available.

COMPATIBILITY IBM®

DISPLAY TELECOMMUNICATIONS QUALITY

$750

MEGA-KIT INCLUDES: System Unit • 256K Mega-Board-XT.™ (2) 360K Floppy Drives and Controller • Enhanced Keyboard • 130W Switching Power Supply–**FCC APPROVED**

- MEGA-KIT™ with Monochrome Monitor and Controller–$909.95
- MEGA-KIT™ with 20 meg Hard Drive, Monochrome Monitor and Controller–$1269.90

IBM is a registered trademark of International Business Machines Corporation.

Mega Board-XT™	289.95	Power Supply	89.95	Disk Drive Controller	59.95
Mega-Case™	69.95	Keyboard	79.95	Floppy Disk Drive	139.95

CALL FOR PRICES ON OTHER COMPONENTS

INTRODUCING: **MEGA-NET™**–Display Telecommunication's NET BIOS Compatible Token-Ring Twisted Pair Local Area Network
- *Benchmarked 20% faster than IBM*
- *Uses 40% less memory than IBM*

Display Telecommunications Corporation

8445 Freeport Parkway • Suite 445 • Irving, TX 75063
1-800-227-8383 • For Technical Calls Only: 1-214-929-1382
TELEX 5106000176 DTC UD

Inquiry 88

Best Small Business Exporter in Texas for 1986

Using Assembly Routines in MS-FORTRAN Programs

Improve the speed of your FORTRAN code

BY MARK DAHMKE

IN THE COURSE of developing a large program, it is often necessary to call assembly language routines to make a critical task efficient or to gain access to features that are machine-specific or not implemented well in a high-level language. To that end, I'll present some mechanisms for dealing with the problem of passing parameters and managing the programming environment when working with Microsoft FORTRAN and 8086-family assembly language. The examples presented here are written in Microsoft assembly language and should work correctly with any version of the assembler.

The FORTRAN Environment

Microsoft FORTRAN, Pascal, and C have many common features in terms of their environment, parameter passing, and data structures. Many of the examples shown here will work well with all these languages, but the emphasis in this article will be on Microsoft FORTRAN.

The Microsoft FORTRAN run-time environment is shown in figure 1. Depending on which version of FORTRAN you are using, there may be some minor differences. FORTRAN begins by allocating memory at the first available low address and places the code segment (all user-generated code and library routines) there. All segments that are part of a group must reside in the same 64K-byte block of memory, due to the segmentation scheme of the 8088 and the 8086. For example, all FORTRAN subroutines compiled together (from one source file) are treated as a group. Thus, you could create a program with 256K bytes of code by compiling the main program and subroutines in four pieces, then linking them.

Similarly, blocks of constants and variables are put in groups that cannot exceed 64K bytes each. The most frequently used of these is the DGROUP, which includes all constants, variables, and unnamed common blocks and strings. If a program becomes so large that the data space exceeds 64K bytes, you must move variables and arrays into common blocks; unfortunately, common blocks are less efficient to address. The DGROUP also contains stack space and heap space used by FORTRAN.

Due to the 8088 and 8086 family's segmented address space scheme, you must establish addressability before reading or writing memory or you will likely cause your system to crash. "Establishing addressability" means loading one of the segment registers with a 16-bit pointer to the base of the block of memory you want access to. Figure 2 shows how this works. The CS, DS, and SS registers establish addressability for code, data, and stack segments, respectively. The ES register is for an "extra segment" that allows you to copy data from one segment to another. The segment registers hold the most significant 16 bits of a 20-bit address. This is added to the "offset" address specified in the instruction. For example, the CS and IP (instruction pointer) registers always point to the current instruction. Addresses of this form are denoted as CS:IP or "segment plus offset."

Parameters passed to subroutines must include the segment portion of the address if the constant or variable

(continued)

A contributing editor for BYTE, *Mark Dahmke owns a consulting business and has had three books published by* BYTE Books/McGraw-Hill. *He can be reached at* P. O. Box 80266, Lincoln, NE 68501.

ILLUSTRATED BY MARY ANN SMITH

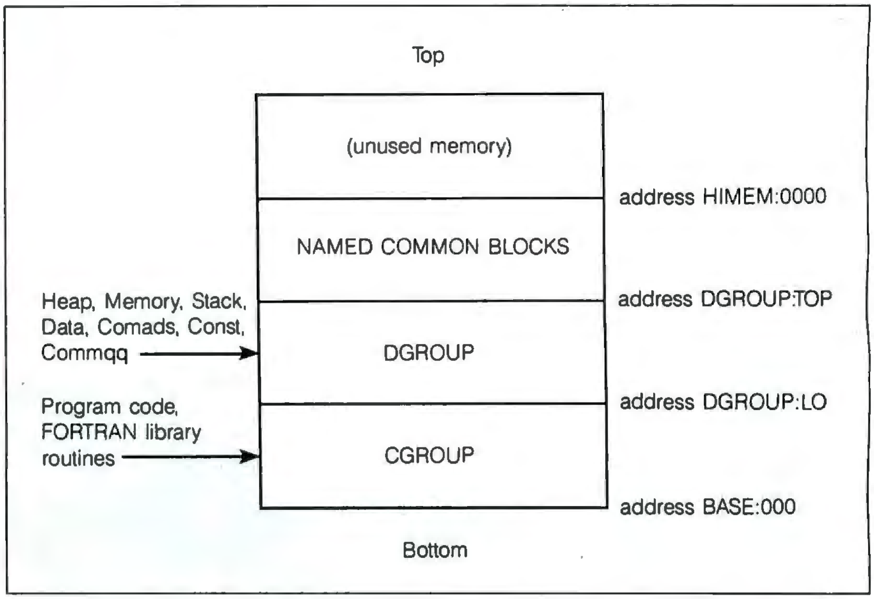


Figure 1: MS-FORTRAN *run-time memory organization.* FORTRAN *sets up two major areas, the code group and the data group. You may need additional groups for named common blocks. Due to the segmented address scheme of the* 8086 *family,* 4*-byte addresses (segment plus offset) are required for most parameter passing to subroutines.*

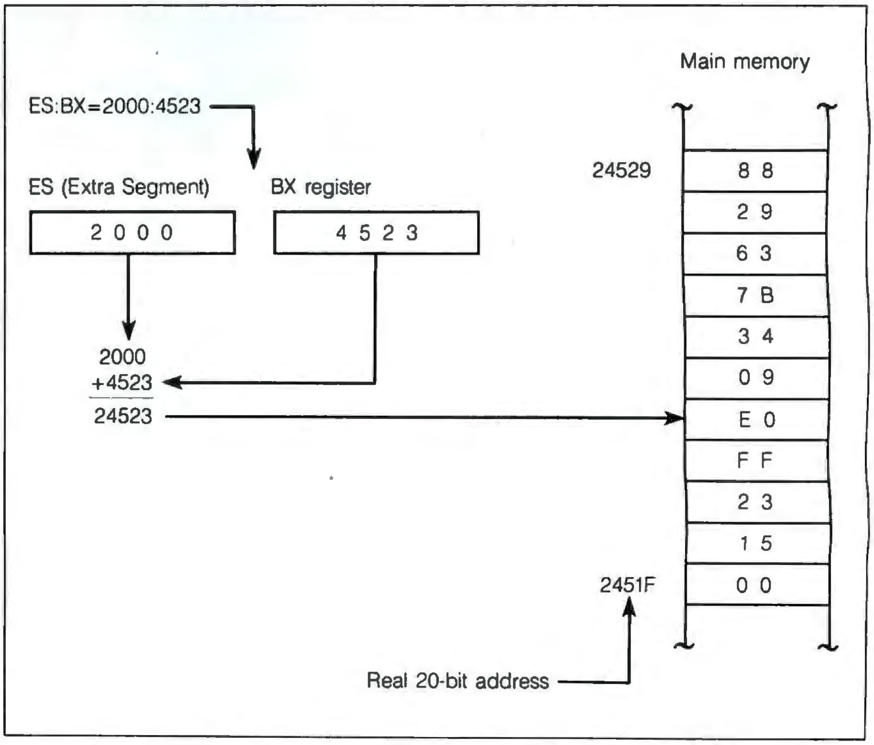


Figure 2: *Establishing addressability. Assuming that* ES *and* BX *have been loaded with* 2000 *and* 4523*, respectively, the instruction* MOV AX,ES:[BX] *would cause the value* 09E0 *to be loaded into* AX. *If interpreted as an* INTEGER*2 *number,* 09E0 *would equal* 2528 *in decimal.*

NETWORK VERSION AVAILABLE

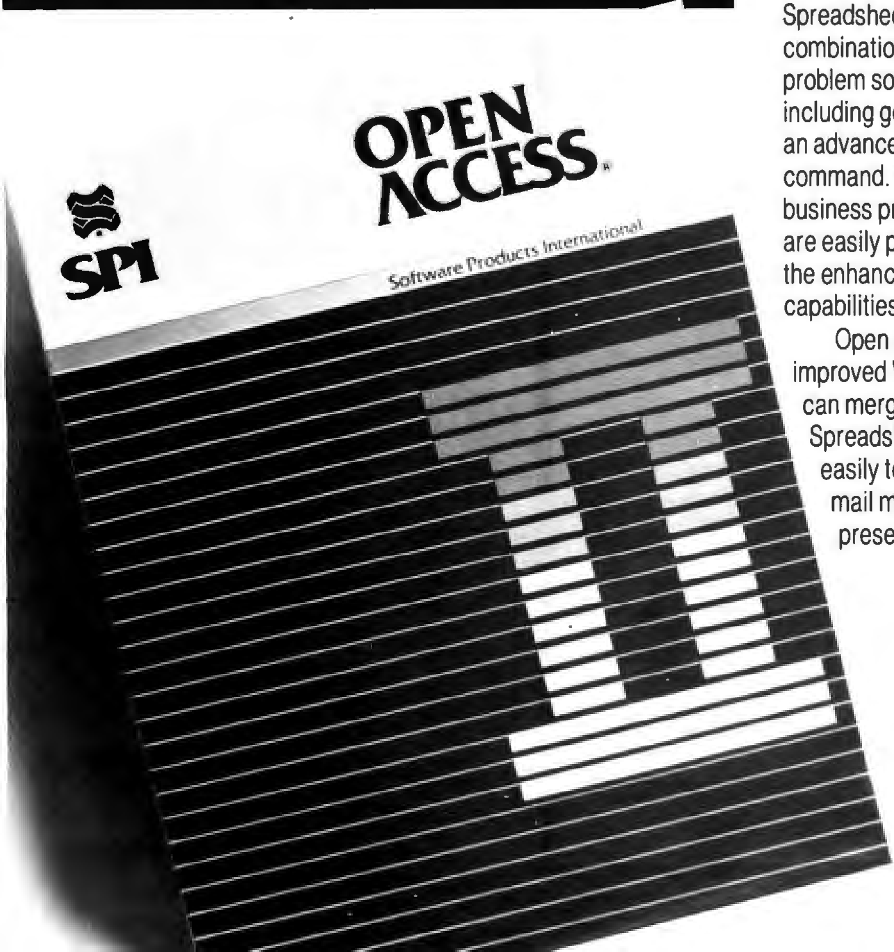
INTRODUCING OPEN ACCESS® II™

THE FULLY INTEGRATED BUSINESS SYSTEM WITH THE WORLD AT YOUR FINGERTIPS

Open Access II is a new, super-program which can perform virtually every managerial and business task you'll ever need with performance and ease-of-use unheard of in the industry.

Open Access II
Comparison Chart

	Database	Form Query	Report Generator	Query Processor	Relational Database	Graphics	3-D Graphics	Spreadsheet	Goal Seeking	Word Processor	Communications	Time Management	Price
Open Access II	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	**595⁰⁰**
Lotus 1•2•3	✓					✓		✓					495⁰⁰
Symphony	✓	✓	✓			✓		✓		✓	✓		695⁰⁰
Framework	✓		✓	✓		✓		✓		✓	✓		695⁰⁰
Enable	✓		✓			✓	✓	✓		✓	✓		695⁰⁰
Smart	✓		✓			✓	✓	✓		✓	✓	✓	895⁰⁰



Open Access II combines an extremely powerful relational database and superior spreadsheet with data communications, word processing, 3-dimensional graphics, and time management. Information is conveniently changed from module to module.

Open Access II's Database features an effective and flexible report generator and user definable screens. "Programmer," our new Database language, features simple interacting English-like commands to produce your customized applications.

Open Access II's Spreadsheet offers a unique combination of business problem solving capabilities; including goal-seeking, and an advanced table look-up command. Professional business presentations are easily produced by the enhanced graphic capabilities.

Open Access II's improved Word Processor can merge data from Spreadsheet or Database easily to produce reports, mail merge and graphic presentations.

Open Access II's data communications provides you with the capabilities to handle any of your micro computer communications needs. All of Open Access II's modules work together and data is easily communicated to the rest of the world.

BEST OF ALL, THE PRICE

$595.

Open Access II's super program is also available in the network version. Ask for details. Call (619) 450-1526 in California, or (800) 521-3511 if outside California for your nearest authorized dealer or distributor and expand your business base with Open Access II, the fully integrated business system.

SPI

SOFTWARE PRODUCTS INTERNATIONAL

10240 Sorrento Valley Road
San Diego, California 92121
(619) 450-1526

For Your Authorized Dealer or Distributor Outside California
(800) 521-3511

Inquiry 265 for End-Users.
Inquiry 266 for DEALERS ONLY.

is not considered local to the subroutine. Since it is almost impossible to tell in advance whether the DS or ES register will be correctly loaded before calling the routine, FORTRAN always passes the segment and offset to a subroutine.

PASSING PARAMETERS

Two cases must be dealt with in FORTRAN: functions and subroutines. In both cases, FORTRAN uses the call-by-reference method for passing parameters, even for constants and expressions. This method makes it easier to write assembly language programs, since the values being passed are always addressed in the segment-plus-offset format.

For both functions and subroutines, the parameters are placed on the stack in the order in which they appear in the parameter list of the calling program. Figure 3 shows the stack before a call to a sample function. In the case of functions, results are returned in registers if the result is defined as an integer or single-precision real number. If the result is 16 or fewer bits long, it is assumed to be in the AX register upon return. If the result is declared INTEGER*4 or REAL*4, it will be in the AX,DX pair. If a number is double-precision (REAL*8), the result must be returned in a dummy variable created by the calling program. The offset address of this variable is in the stack at location BP+6.

SUBROUTINES

All variables passed to subroutines and all results returned from subroutines are passed as parameters in the call list. Figure 4 shows how the stack is loaded prior to calling the sample subroutine shown.

ENTRY AND EXIT REQUIREMENTS

As shown earlier, the addresses must be obtained from the stack and placed in segment and offset registers to gain access to the value of the variable. This can be confusing, especially if the stack contains many parameters. All addressing is done on the basis of the base pointer (BP) register.

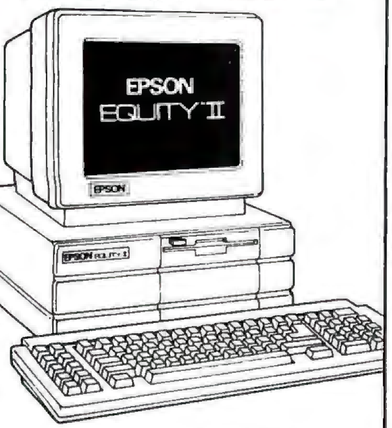
(continued)

S-100 "THE PC PEOPLE"

800-528-3138 Orders Only
602-991-7870 Customer Service

S-100 DIV./696 CORP.
14455 NORTH 79TH ST.
SCOTTSDALE AZ 85260
TELEX 9103806778 SONE HUND

THE CURE FOR THE COMMON CLONE!

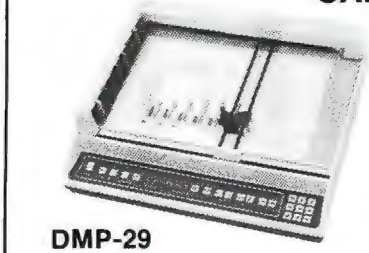


EPSON EQUITY II

IBM-PC compatibility mated to the new NEC V30 microprocessor gives you 8086 CPU power at speeds of 4.77 and 7.16 MHz. On board you'll find 640K RAM, color and Hercules compatible graphics, clock/calendar and 5 full size option slots - all standard equipment! Add to this two 360K drives, an AT-type keyboard and a one year warranty - well, you can see it makes sense to start building some EQUITY into your business.

The Epson EQUITY II......CALL

CAD



DMP-29

houston instrument

DMP 41/42	$2449
DMP 51/52	3699
DMP 56	4399
DMP 29	1699
DMP 40	889

True Grid Digitizers

TG 1005	$289
TG 1011	509
TG 1017	619

Also: cursors, styli, etc.

HANZON LP-3000+ LASER .... **CALL**
HPGL PLOTTER COMMANDS!!!

NEC *MULTISYNC* **$529**

ACS *GRAPHAX 20/20* 2K by 2K Graphics Resolution Single Card Design, up to 32 Colors, Supports AutoCAD.................................. **CALL**

ACS

EVERYONE'S PHONING HOME ABOUT THIS ADVANCED "AT" COMPATIBLE SYSTEM

"ET-286" BASIC SYSTEM

12MHZ ONE WAIT STATE OF 10MHZ NO WAIT STATE SPECIAL ORDER - ALLOW 5 WEEKS. 10 & 6 MHZ 80286 CPU ONE WAIT STATE, 1 - 4MB RAM ON MOTHER BOARD.

ALSO ON MOTHERBOARD:
- 6 AT COMPATIBLE SLOTS
- 2 XT COMPATIBLE SLOTS
- 3 RS232 OR RS422 SWITCHABLE S PORTS
- 2 CENTRONICS P PORTS
- CLK/CAL With BATT BACK-UP
- 100% IBM-AT COMPATIBLE BIOS
- CABLE, 200 WATT P/S, & KEYBD..$1595

Printers

CORDATA *LP300* Laser w/ WordStar 2000	2175
TOSHIBA *P321* Para & Serial	449
TOSHIBA *P341*	789
CITIZEN *MSP25*	499
CITIZEN *MSP 20*	349
CITIZEN *120D* 120CPS, TRAC	189
OKIDATA *292* Color Printer	459
OKIDATA *293* Color W/ Trac, 132 Col	589
OKIDATA *Personality Boards*	88
XEROX *Diablo D25*.......$484 *Diablo 635*	949
XEROX *4045 Laser*	4585
STAR MICRONICS *SR15*	595

BROTHER M-1509

~~$399~~ CALL

180 CPS Draft - 45 CPS NLQ - Push Tractor
Auto Feed - S&P Ports - Wide Carriage
EPSON Compatible - More!

1409	**CALL**
TWINWRITER 5 Daisy & Dot Matrix	**CALL**
HR35 36 CPS Daisy, 132 Col	**$729**
HR20 20 CPS	**$349**
HR15XL Daisy	**$299**

Hard Disk & Tape Subsystems

20 MB HARD DRIVES
with CONTROLLER
$379

MICROPOLIS *MC-1325* 85MB 30mSEC 5¼"	1299
QUANTUM *Q540* 43MB, 45mSEC 5¼"	939
MAXTOR *XT1140* 140MB, 30mSEC 5¼"	3299
AMCODYNE *ARAPAHOE 7110*, 25MB Fixed & 25MB Removable CDC Lark Plug Compatible	CALL
IRWIN 10MB Tape B/U	369
IRWIN 20MB Tape B/U	459
WANGTEC *PC-36* 60MB INT Tape *FAST!!!*	785

Software

WE SELL ALL WELL KNOWN BRANDS. **ORDER CORRECTLY! OPENED SW IS NOT RETURNABLE!**

PC Boards, Etc.

TECMAR *WAVE* 64 - 256K..$149 *BOSUN* S,P,CLK.	149
TECMAR *Expansion Chassis* - 11 Slots, Etc.	595
TECMAR *Graphics Master*	429
TECMAR *IEEE 488*	235
AST *6 Pack Plus*	165
AST *6 Pack Premium*	385
AST *Rampage AT*	512
AST *Advantage*	385
INTEL *Above Board* PCMB 1010.... List **$345** Sell	199
INTEL *Above Board* - AT 128K to 4MB	399
STB *Rio Grande* 128K to 5MB	269
MACROTECH *MSRAT* 6,8,10,12MHz, to 3MB..From	185

PC & AT Multi User Cards

ALLOY *PC-SLAVE/16* 1MB 8MHz V-20	**$655**
RTNX *MSDOS 2.1* / **ATNX** *MSDOS 3.1*	119/189
PC PLUS PKG! PC Slave / 16, Link 125 Term, 20' Cbl	1099

Scientific / Indust. Bds.

Industrial Computer Designs:
A/D 64-100..........$315 A/D 64-PC..........$396
D/A 64-100..........$236 D/A 64-PC..........$468
1018-PC 96 TTL INPUT LINES **$446**
Scientific Solutions (TECMAR)
IEEE-488..........$239 LAB MASTER..........$659

Power Systems & Acces

SAFE (SAFT) *SPS1000* WATT SINE WAVE	**$989**
SOLA *750 WATT* With 4500 WATT HI-INRUSH	1495

Computer Accessories:

ALL CABLES SOLD AT DEEP DISCOUNT	**CALL**
S4 4 CIRCUIT SURGE SUPPRESSOR	**$19**
S10 6 CIRCUIT SURGE SUPPRESSOR	29
P15 5 CIRCUIT MON BASE W/ MODEM PROT	98
U1000 POWER SAVER 1000W U.P.S. (TOPAZ)	729

Modems

CTS Hayes Comp. *212 AHC*...$149 *224 AD×1*	**$295**
PROMETHEUS *PROMODEM 1200*	259
PROMODEM *1200G*........$179 *2400G*	339
MULTITECH *MT224EH* MNP	529
MULTITECH *MT224ER* Rack Mount	495

Floppy Disk Drives

MITSUBISHI *2894* 8" **$475** *2896* 1/2 HT,8" **$435**
MITSUBISHI *MF501* 48TPI..**$95** *MF504* 96TPI..**$129**

TEAC *FD55BV* 48TPI	**$89**
TANDON *TN-50* DSDD For Compatibles	**CALL**

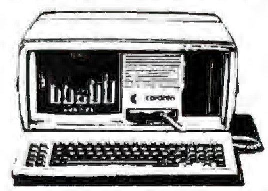
Computer Systems

BIG BLUE CLONE™

When you need performance without the Name Brand cost, the BIG BLUE CLONE from S-100 will fit the bill at a price that will let you foot the bill.

THE BASIC B.B.C.:
256K turbo with 8 Slots, 150 W P/S, Slide Case, Speaker, At Style Keybd. F.C.C. Approved & a 1 Yr. Warr...... **$424**
CALL 1-528-2138 NOW FOR ADDITIONAL OPTIONS!

CORDATA AT SERIES:



NEW LOW PRICES

AT PORTABLE 8MHz, 640K, 1.2 Flpy, S,P, Mon	$1649
AT DESKTOP 8MHz, 640K, 1.2 Flpy, S,P	1589
AT DESKTOP 20MB as above w/ H.D.	1999
PC-400 512K, 2 Flpys, 14" 6740x400 Mon	979
PPC-400 as above but Portable w/ 9" Mon	979
20MB Version of Desktop or Portable	1369

ALSO:

SPERRY *PC/IT* Enhanced	**$2899**
ZENITH *148, 158, 171, 200 AT*	CALL
ALTOS *1086-50* 1MB RAM, 50MB HARD DISK, 50MB TAPE DRIVE, XENIX	11399
ALTOS *1086-80* AS ABOVE WITH 80MB HD & TAPE	12999
ALTOS *686-25* 6 USER, 25MB HD, XENIX	4099

Monitors & Terminals

NEC *MULTISYNC*	**$529**
ZENITH *1240* Flat Screen, Amber	149
PRINCETON GRAPHICS *All Models*	CALL
TATUNG *1422A* Dual Freq, Tilt/Swival 14" G/A	139
TATUNG *1360* 640x200 RGB, 13" G/A Switch	339
WYSE TERMINALS *ALL MODELS*	CALL
LINK *125* Emulates WY-50 1 Year Warranty	399

All merchandise new. Advertised prices are cash prepaid only. PO's from qualified firms & AM. Express: ÷.95.MC & Visa: ÷.968. COD's (99, $5 min. fee) with Cashiers Check/MO. Bank Wires & APO's accepted. Shipping: minimum $4 first 5lbs. Tax: AZ RES. ONLY add 6 5% sales tax. All returns subject to 20% restocking fee or credit towards future purchases. All prices & availability subject to change without notice.

S-100™ ESTABLISHED 1977

Inquiry 247 for End-Users. Inquiry 248 for DEALERS ONLY.

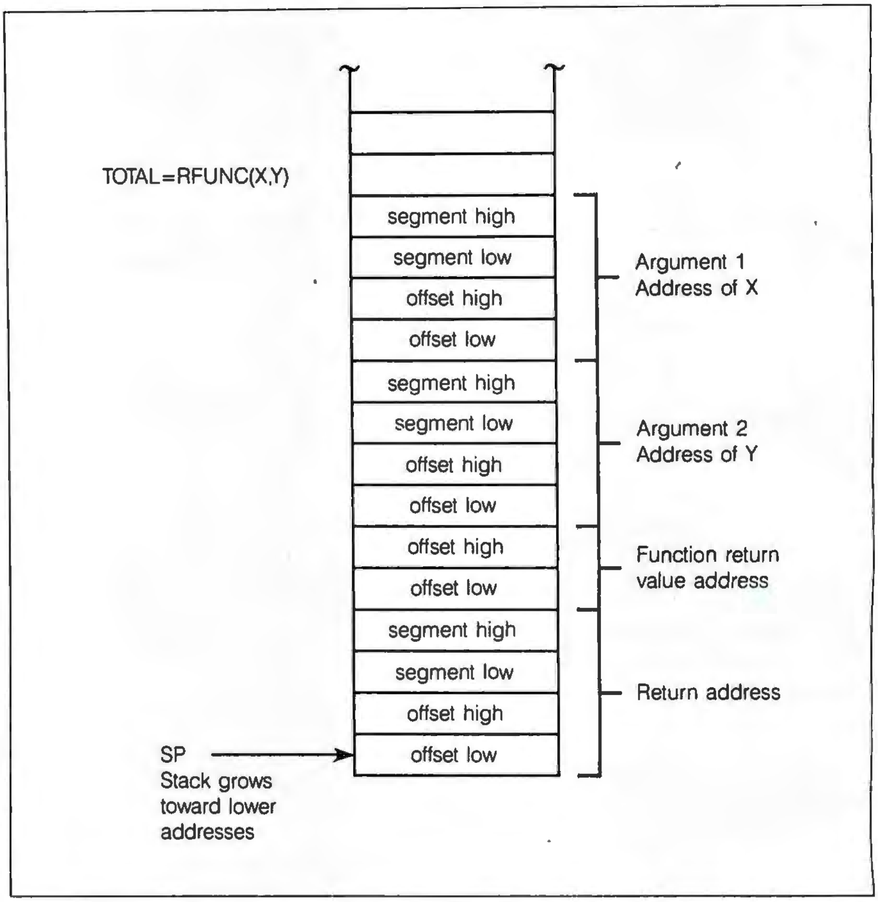


Figure 3: *The stack segment before a function call. All argument addresses are pushed onto the stack prior to a subroutine or function call. Four bytes are required to store each address in the segment-plus-offset format.*

To simplify the process of computing offsets, I wrote the macros shown in figure 5.

SETFRAME (see listing 1) performs the housekeeping tasks needed to gain access to the stack. The frame pointer must be saved, so it is in the same location when returning to the calling program. Also, the stack pointer (SP) must be loaded into the BP so that the offset address to the parameters will be correct.

The GETPARM macro solves the problem of getting an address and putting it in the ES:BX register pair. The first argument of GETPARM is the number (counting from the left) of the parameter you want to retrieve. The second argument specifies the total number of parameters in the call list. For example, if the call CALL SUBR(A,B,C) were made from a program, and you wanted to retrieve the address of the variable C, you would enter GETPARM 3,3, which would return the address of the third of three parameters. Listing 1 shows the instructions generated for all three parameters in the call list of this sample program. [*Editor's note: Listing 1 and the sample assembly language subroutines discussed below are available on disk, in print, and on* BIX. *See the insert card following page* 176 *for details. Listings are also available on* BYTE*net. See page* 4.]

The third macro shown in figure 5 is for housekeeping at the end of a subroutine. POPRET will make sure the SP and BP registers are returned to their starting values and that the correct number of bytes is popped off the stack before returning. This is necessary because the calling program

(continued)

If you pay the price of a network, you're paying too much.

The Classic Multiuser System delivers full PC performance to multiple users for half the price (or less) of a network.

Which helps explain why it's selling so well. And why ComputerLand chose it.

You can select from a virtually unlimited selection of software. The Classic Multiuser System runs all DOS 3.1 and 3.2 programs, in addition to most programs designed to run on the IBM and 3Com 3+ networks.

You get all the advantages of a network in a true multiuser solution, complete with record locking and file sharing.

It just doesn't make sense anymore to add a PC for each new user when you need only monitors and keyboards.

Especially since the Classic Multiuser System is faster to install than a network. And much easier to work with, because all users see are the familiar DOS commands they already know. So you not only save on hardware, you save on training and support.

Simply insert our solution in an AT (or in a PC with our 286 Speed Pak) and you're ready to grow your own economical Classic Multiuser System.

Ask for the Classic Multiuser System at participating ComputerLand stores nationwide. Or call us at 800/392-8324 for the dealer nearest you. (408/434-9333 in CA.)

Or pay the price.

CLASSIC

Inquiry 60 for End-Users.
Inquiry 61 for DEALERS ONLY.

The Classic Multiuser System. Now at ComputerLand.

Atlanta Office: 404-587-3237
Chicago Office: 312-695-8233
New York Office: 516-434-1616

Classic Technology Ltd.–EUROPE
Jupiterstraat 98
2132 HE Hoofddorp
Netherlands
31-2503-32074

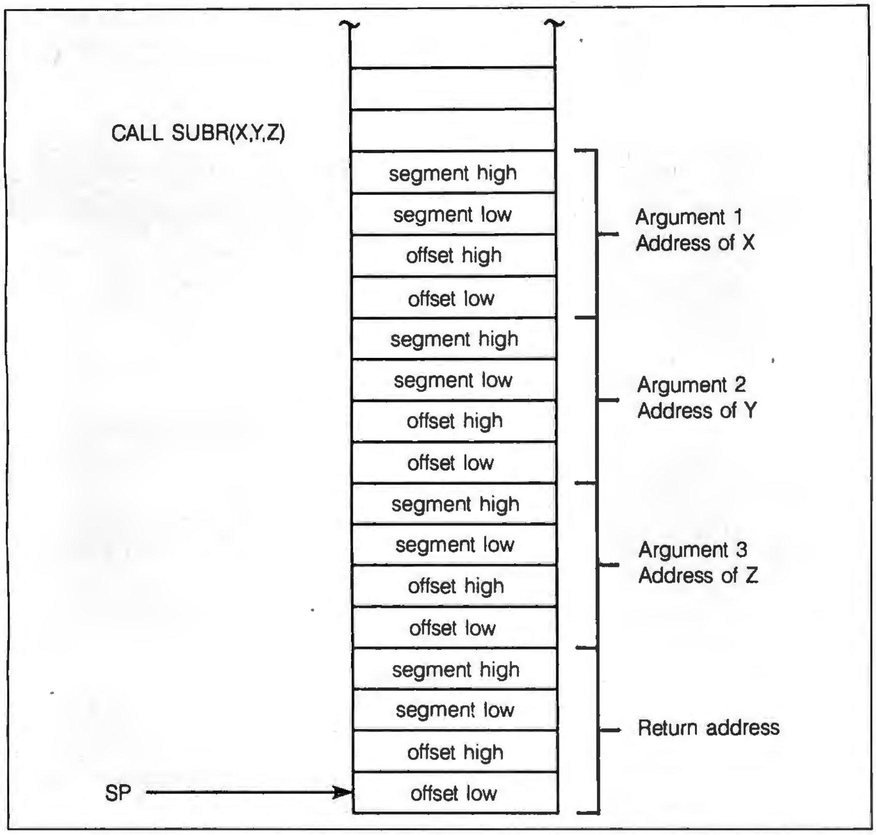


Figure 4: *The stack segment before a subroutine call.*

assumes that the stack is the same as it was before setting up for the subroutine call. The only argument given to POPRET is the number of parameters in the call list. In this example, four parameters were passed, so you would code POPRET 4. After 16 bytes are popped off the stack, a FAR return instruction is executed, which passes control back to the caller.

EXAMPLES

I have written four sample subroutines. The first two, SRCHF and SRCHN, allow a calling FORTRAN program to scan the current directory for filenames. These subroutines use the old DOS function calls that use FCBs (file control blocks) instead of the newer DOS functions, which use file handles.

SRCHF and SRCHN are almost identical, except that SRCHF locates the first matching directory entry and SRCHN finds subsequent matching entries. SRCHF is called just once, while SRCHN can be called repeatedly to get subsequent filenames that may or may not be identical to the first one returned.

(continued)

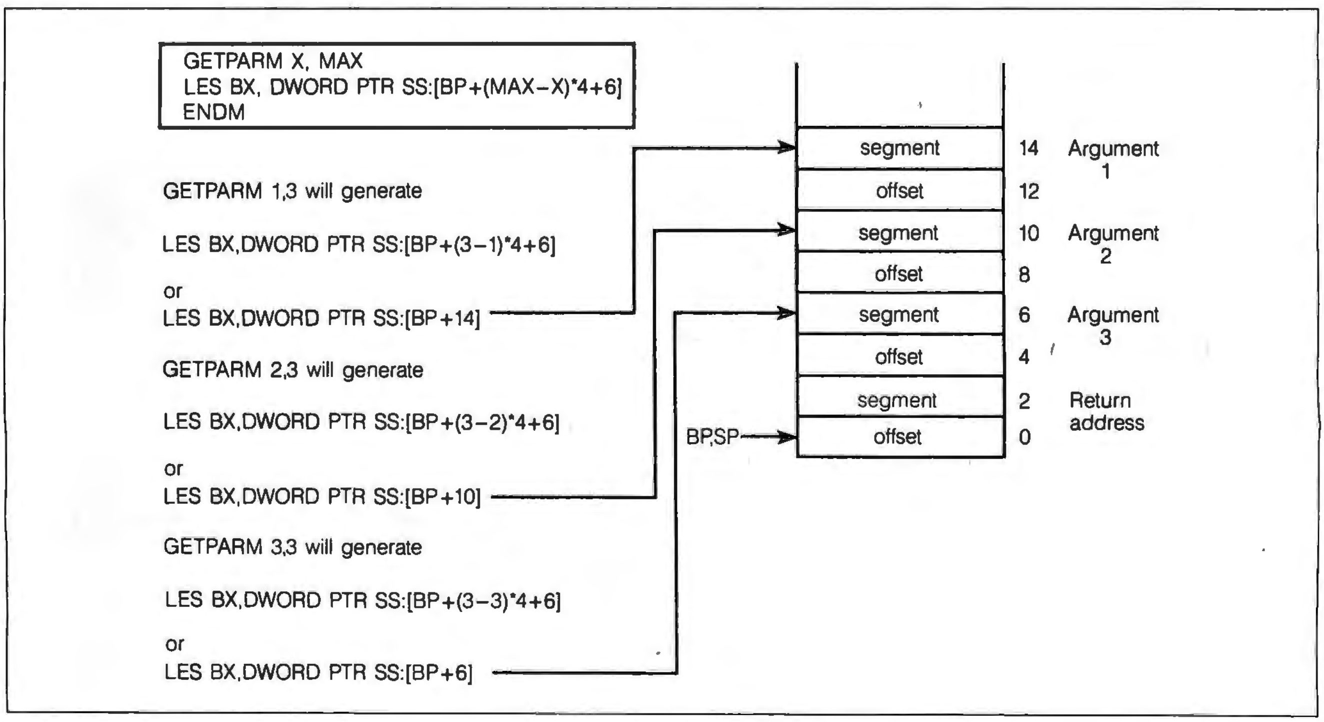


Figure 5: *Utility macros that simplify parameter handling in assembly language subroutines.*

10 Mhz
Model Available Now!
Increases Speed 10 Times

ComputerLand chooses the Classic 286 Speed Pak. Here's why...

Speed. Compatibility. Upgradeability. You need all three in a turbo board for your PC/XT.

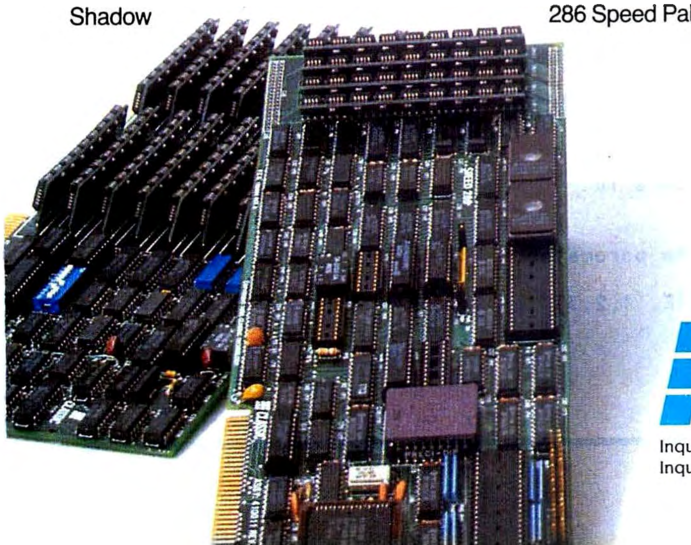
And no one but Classic delivers all three.

The Classic 286 Speed Pak makes your PC/XT blaze-up to **7 times** faster with the 8Mhz model, and **10 times** faster with the 10Mhz model. That's up to 100% faster than turbo boards that use cache.

And it's the only **fully** compatible turbo board. Simply plug in and run. No software patches needed. It works with the IBM EGA, Lotus/Intel Above Board and all popular networks.

What's more, the Classic 286 Speed Pak is the **only** turbo board that grows with your needs...

TURBO BOARD COMPARISON	CLASSIC 286 SPEED PAK	ORCHID TURBO 286E	VICTOR/PCT MOUNTAIN
SPEED:			
▲ INCREASES PC/XT SPEED*	4-5X	5X	2-3X
▲ 16-BIT PROCESSING-20-30% FASTER THAN CACHE	YES	YES	NO
COMPATIBILITY:			
▲ FULLY SOFTWARE/ HARDWARE COMPATIBLE	YES	NO	NO
▲ NO SOFTWARE PATCHES NEEDED	YES	NO	NO
▲ IBM EGA COMPATIBLE	YES	NO	YES
▲ LOTUS/INTEL EMS COMPATIBLE	YES	NO	YES
▲ NETWORK COMPATIBLE	YES	NO	YES
▲ STANDARD 8088 MODE	YES	YES	NO
UPGRADEABILITY:			
▲ 16-BIT MEMORY EXPANSION UP TO 16MB	YES	NO	NO
▲ 16-BIT EMS TO 5MB	YES	NO	NO
▲ MULTIPLE VDISKS	YES	NO	NO
▲ MULTIUSER OPTION	YES	NO	NO
PERFORMANCE:			
▲ FULL 16-BIT BIOS/RAM	YES	YES	NO
▲ INCLUDES 1 MB MEMORY	YES	YES	NO
3-YR. WARRANTY:	YES	NO	NO



The "DYNAMIC DUO"

Add Classic SHADOW 2.5MB memory modules to run high speed EMS memory. You'll move through huge spreadsheets with dazzling speed, thanks to SHADOW's 16-Bit memory.

And your disk-hungry programs run lightning fast with SHADOW's high capacity RAM disks. You have the flexibility to use both EMS and RAM disks for more power and speed.

You can even turn your PC or XT into a complete multiuser, multitasking system.

The 286 Speed Pak and the SHADOW are the perfect pair to make your PC or XT a *real powerhouse.*

So visit your local ComputerLand store today or call 800-392-3224 for the name of your local Classic dealer [In CA, call 800-338-0777].

Because no one else gives you everything in a turbo board. No one, no way, no how.

Classic Technology Corporation. 2090 Concourse Drive, San Jose, CA 95131.

Atlanta Office: 404-587-3237
Chicago Office: 312-695-8233
New York Office: 516-434-1616

Classic Technology Ltd.–EUROPE
Jupiterstraat 98
2132 HE Hoofddorp
Netherlands
31-2503-32074

CLASSIC

Everything you always wanted in an IBM.

Inquiry 62 for End-Users.
Inquiry 63 for DEALERS ONLY.

*Average benchmark using popular programs
IBM is a registered trademark of International Business Machines Corporation
CLASSIC, 286 SPEED PAK and SHADOW are trademarks of Classic Technology Corporation
COMPUTERLAND is a registered trademark of ComputerLand Corporation

Because these two routines are nearly identical, I will describe only SRCHF in detail. The calling FORTRAN routine must load a string with the filename to be matched. This name may include question mark (?) as a wildcard, but not an asterisk (*). While the asterisk is a valid wildcard in many DOS commands, this low-level function call doesn't recognize it. Normally the asterisk is parsed by the calling program and converted into question marks.

The calling program must pass two other parameters: the drive code and an empty string in which the requested filename is returned. The drive code is an integer that corresponds to the disk drive letter (1 = A:, 2 = B:, 3 = C:, and 0 = default).

Upon receiving control, the SRCHF routine retrieves the drive code value. The GETPARM 1,3 macro looks into the stack and finds the double word pointer for the drive code, which is returned in the ES:BX register pair. The next instruction gets the value of the drive code and stores it in the AX register. The FCB template must be initialized with the drive code and filename. The next instruction copies the least significant byte of the drive code (all that really matters anyway) into the first byte of the FCB.

Next, a GETPARM 2,3 macro retrieves the pointer to the second of three parameters passed to SRCHF—the filename itself. The filename must be stored as 11 characters with no period separating the filename from the file type. For example, the filename SAMPLE.DAT would be stored as SAMPLE_ _DAT. A second pointer is set up to copy these 11 bytes from the FORTRAN string to the filename area of the FCB. The length is loaded into CX, and a REP MOVSB instruction performs the copy automatically.

After the FCB is loaded, DOS must be told where to find the buffer that will hold the resulting FCB. This is done through the SET DMA ADDRESS function call. Finally, a call to the SEARCH FIRST function will cause DOS to check the directory and return either a matching filename or an error code. If a filename is found, an error code of zero will be returned, along with the filename in the same 11-byte format as described earlier. If no files are found, the subroutine returns a single question mark in the first character of the returned filename string.

GETDIR AND CHDIR

The subroutine GETDIR returns the current directory name. Three parameters are required: an empty string to store the resulting directory name, the drive code, and an error code variable.

This routine first gets a pointer to the character string and stores the segment and offset in STRING_SEG and STRING, respectively. After load-

(continued)

Listing 1: *The* GETPARM *macro loads the* ES:BX *pair with the address of a parameter from the subroutine call list. The first argument of the macro is the number (counting from the left) of the desired parameter in the call list. The second argument* (X,MAX) *is the maximum number of parameters in the call list.*

```
;  SETFRAME:  Sets the environment upon entry to a subroutine.
;
SETFRAME MACRO
        PUSH    BP                  ;SAVE FRAMEPOINTER ON STACK
        MOV     BP,SP
        ENDM
;
;  POPRET:    Restores the BP register and returns to the
;             calling FORTRAN routine after cleaning up the
;             stack.
;
POPRET  MACRO   NPARMS              ;RETURN FROM SUBR.  NPARMS=NUMBER OF PARMS
        POP     BP
        RET     NPARMS*4
        ENDM
;
; GETPARM: returns a pointer to a parameter in the call list.
;
;       Operands:  X = the number of the desired parameter
;                MAX = the maximum number of parameters in
;                      the call list.
;
;       Result: The ES:BX register pair points to the parameter.
;
GETPARM MACRO   X,MAX               ;PARAMETER NUMBER (IE. 1,2,3)
        LES     BX,DWORD PTR SS:[BP+(MAX-X)*4+6]
        ENDM
;
;
```

AVOCET INTRODUCES MACRO ASSEMBLERS.

Put your design ideas into more chips.

New AVMAC™ Macro cross-assemblers, combined with our simulators, emulators and EPROM programmers, turn your PC or VAX into a complete development system.

AVMAC is fast. Its transparent buffering scheme provides the speed of memory-based assembly without restricting file size. And, AVMAC has one of the most comprehensive macro facilities available.

AVMAC is loaded with features that make your development job easier. You get source code flexibility; AVMAC supports over 1000 relocatable segments. AVLINK™ linker and AVLIB™ librarian support modular programming with full control over segment combination and placement. AVREF™ generates cross-references by line and procedure. And, AVMAC gives you informative, plain English error messages.

AVMAC offers superior compatibility with manufacturers' assembly languages. And it's designed to work hand-in-hand with AVSIM™, Avocet's software simulator/debugger.

Every company talks about service. But at Avocet we're committed to satisfying customer requests for in-stock items in 2-3 working days. And that's a fact!

For easy access, call our 800 number for ordering, product information and technical assistance. We're also interested in *your* ideas on how we can better serve your needs.

AVMAC Macro Cross-assemblers

MSDOS™, PCDOS™	$349
XMAC68K	$595
VAX™/UNIX™	$995

Target Microprocessor Families Supported:

6804	6502/65C02	Z8
6805	6800/01,6301	Z80
6809	NEC 7500	68H11
1802/1805	8085	HD64180
8048/8041	COP400	68020
8051	F8/3870	68000/68010
NEW!	TMS32010	TMS32020

In addition to AVMAC, Avocet provides everything you need to develop microprocessor software.

AVSIM full-screen Simulator/Debuggers make software simulation practical. Test your code in a crashproof, interactive environment, without additional hardware. Price $299.

TRICE™ self-powered In-circuit Emulators recognize 34 different commands; set break points, single-step, trace and more. Priced from $498.

AVPROM™ Programmers work with any PC. Program over 37 different devices, including EPROMs through 27512, CMOS and E²PROMS, and MPU/EPROM combos, using fast, "adaptive" algorithms. From $429. Gang programmers from $979.

To find out more about Avocet software development tools, call us toll-free:

1-800-448-8500

(In the U.S. except Alaska.)

VISA and Mastercard accepted. Most popular disk formats now available—please specify. Software shipment by second-day air within the continental U.S. included. Special handling and other shipping additional; call for exact quotes. OEM INQUIRIES INVITED. Avocet Systems Inc., P.O. Box 490-B16, Rockport, Maine 04856, (207) 236-9055. Telex: 467210 AVOCET CI.

AVOCET SYSTEMS INC.®

™ signifies manufacturer's trademark

ing the drive code into the AX register, the string segment and offset addresses are loaded into the DS:DX pair, and DOS function 47h is called. This function stores the full directory name in the string space but will not add a leading backslash (\ ) or drive letter and colon to the name. Thus, if you are currently in the directory C: \ MAIN \ SAMPLE, this subroutine will return the string MAIN \ SAMPLE. Also, the string will be terminated with a null or zero byte. After the DOS function call, the return code is stored in the third variable and returned to the calling program.

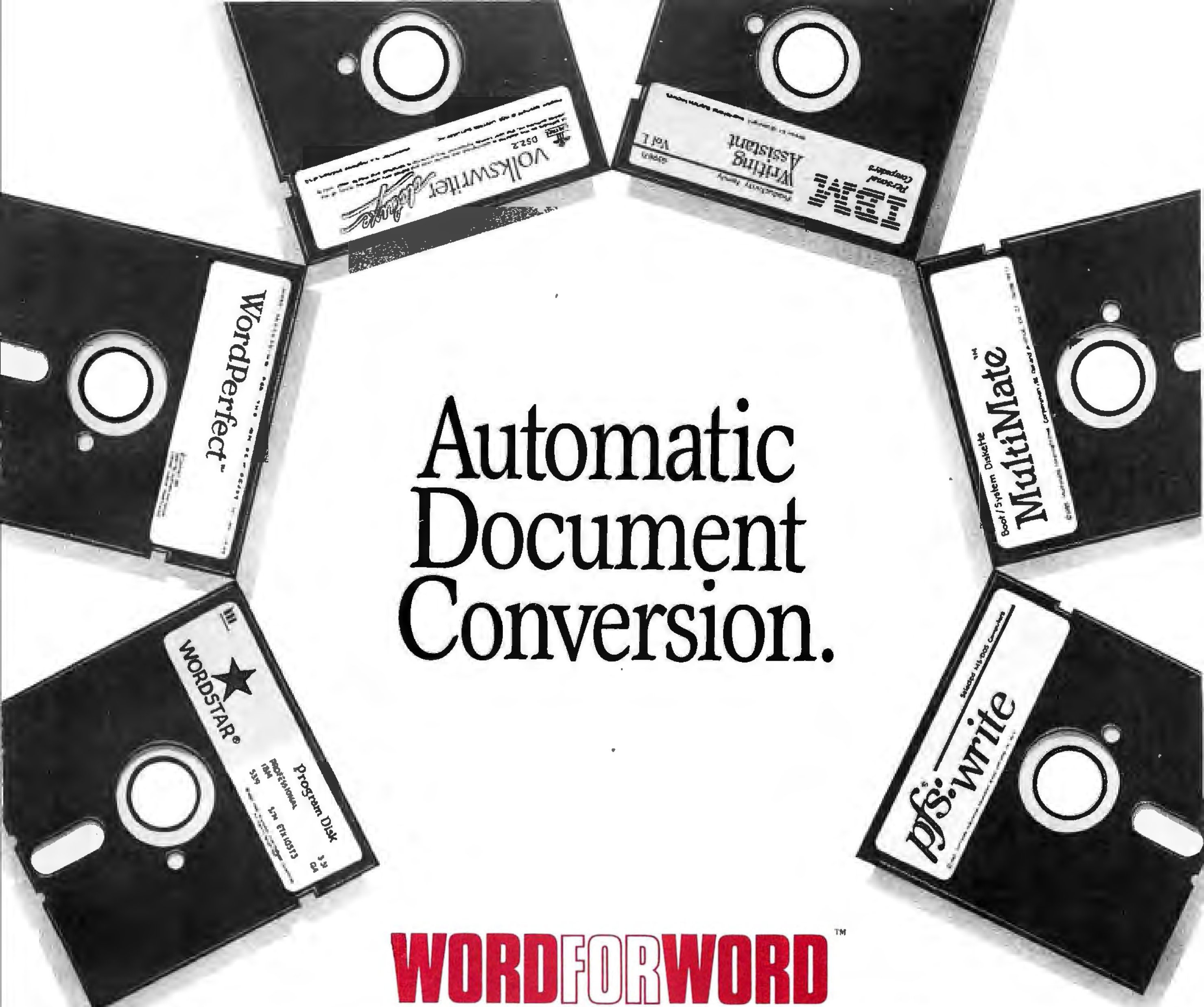
CHDIR (change directory) is almost identical to GETDIR, but the input directory name can be fully qualified, including a drive code, colon, and leading backslash character, just as you would enter it in the DOS CD command. The string must be terminated with a null or zero byte to be accepted, though.

THE GETDFS SUBROUTINE

The GETDFS (get disk free space) routine returns information about the disk drive. Three values are returned, indicating the number of bytes per sector, sectors per cluster, and the number of free clusters on a disk. With these three values, you can calculate the number of free bytes easily. The values are returned to the calling program in INTEGER*2 variables, but if the three are multiplied to get free space in bytes, you must use an INTEGER*4 variable to hold the result. If the value returned in the fourth argument of the call list (ICLUST) is equal to −1 (0FFFF hexadecimal), an error condition has resulted.

A USEFUL MODEL

Many DOS functions, graphics, screen formatting, and resident modules cannot be handled through native FORTRAN code. By coding these functions in assembly language, you can extend the versatility of a program without forcing a complete rewrite in a different language. These sample routines can provide a useful model for creating your own extensions to FORTRAN. ■



Automatic Document Conversion.

WORDFORWORD™

by Mastersoft,® Inc.

Word For Word is the software document converter that converts files and documents from one PC-compatible word processing system to another.

Producing **"mirror-image"** conversions that preserve ALL features and functions, **Word For Word** can transmit documents over standard communication links between like or unlike word processing systems. It also allows uploading to or downloading from mini and mainframe computers in ASCII and EBCDIC formats.

Fast—converts a one-page document in five seconds.

Easy-to-use, menu driven operation.

For further information call (800) 654-5301 or (213) 493-2471 and ask about the availability of Microsoft Word,™ Displaywrite III,™ DCA and Wordstar 2000.™

Dealer Inquiries Invited

Marketing and Sales
by CSSL, Inc.
909 Electric Avenue, Suite 202
Seal Beach, CA 90740
(800) 654-5301 or (213) 493-2471

SPECIAL INTRODUCTORY OFFER — $149.

WORDFORWORD

c/o CSSL Inc.
909 Electric Avenue, Suite 202
Seal Beach, CA 90740

☐ YES, send me Word For Word for $149.00. (Add $5.00 for shipping and handling. Arizona residents add an additional $9.98 for sales tax.)

☐ My check is enclosed. Or charge my

☐ VISA ☐ MasterCard ☐ Am.Ex.

Card # ______________

Exp. ______________

Name ______________

Address ______________

City ________ State ____ ZIP ____

Telephone (   )

The following are trademarks of their respective companies: WORD FOR WORD and Mastersoft: Mastersoft, Inc.; Wordstar and Wordstar 2000; Micropro International Corp.; Multimate: Ashton-Tate, Inc.; Displaywrite III and IBM Writing Assistant: International Business Machines Corp.; WordPerfect: Satellite Software, Inc.; Microsoft Word: Microsoft, Inc.; Volkswriter: Lifetree Software, Inc.; PFS: Write: Software Publishing Corp.

9:08: *"Hit any key. Where's the 'any' key?"*

9:47: *"So that's how you create a file."*

10:26: *"And they all laughed when I sat down at the computer."*

DataEase. If it was hard to use we would have called it DataHard.

There are many jobs a database manager is perfect for. It can significantly reduce the hours people spend keeping records, filing and creating reports. The trouble is most database managers aren't perfect for the people who have to do those jobs. They're just too difficult for them to learn.

So a lot of people end up using a spreadsheet program to try to do the job of a database manager. A job the spreadsheet wasn't designed for and doesn't do very well.

That's why many companies that had already standardized on complex systems like dBASE™ have decided to re-standardize on DataEase.™

DataEase. No database manager that does as much is easier.

People become productive immediately with DataEase. Even people without any PC experience. In just a couple of hours they'll be setting up forms, files and reports by themselves. Because DataEase is menu driven and leads them through the process.

And DataEase is just as powerful as much more complex systems. But the power doesn't get in the way of a beginner using DataEase. Instead it becomes available as it's needed. So as they become more experienced, the transition from building simple applications to building complex ones is seamless. And easy.

And even though DataEase is easy, we support it as if it weren't. We have customer support people to answer your questions. Seminars and user groups around the country to help you make more productive use of DataEase. And we offer free templates to help make developing applications even easier.

DataEase. What it can do is hard. How it does it is easy.

Send us the coupon for a free DataEase demonstration disk. Or call 1-800-334-**EASE**, or 203-374-8000.

Byte 10/15/86

Software Solutions, Inc.
12 Cambridge Drive, Trumbull, CT 06611
1-800-334-**EASE**

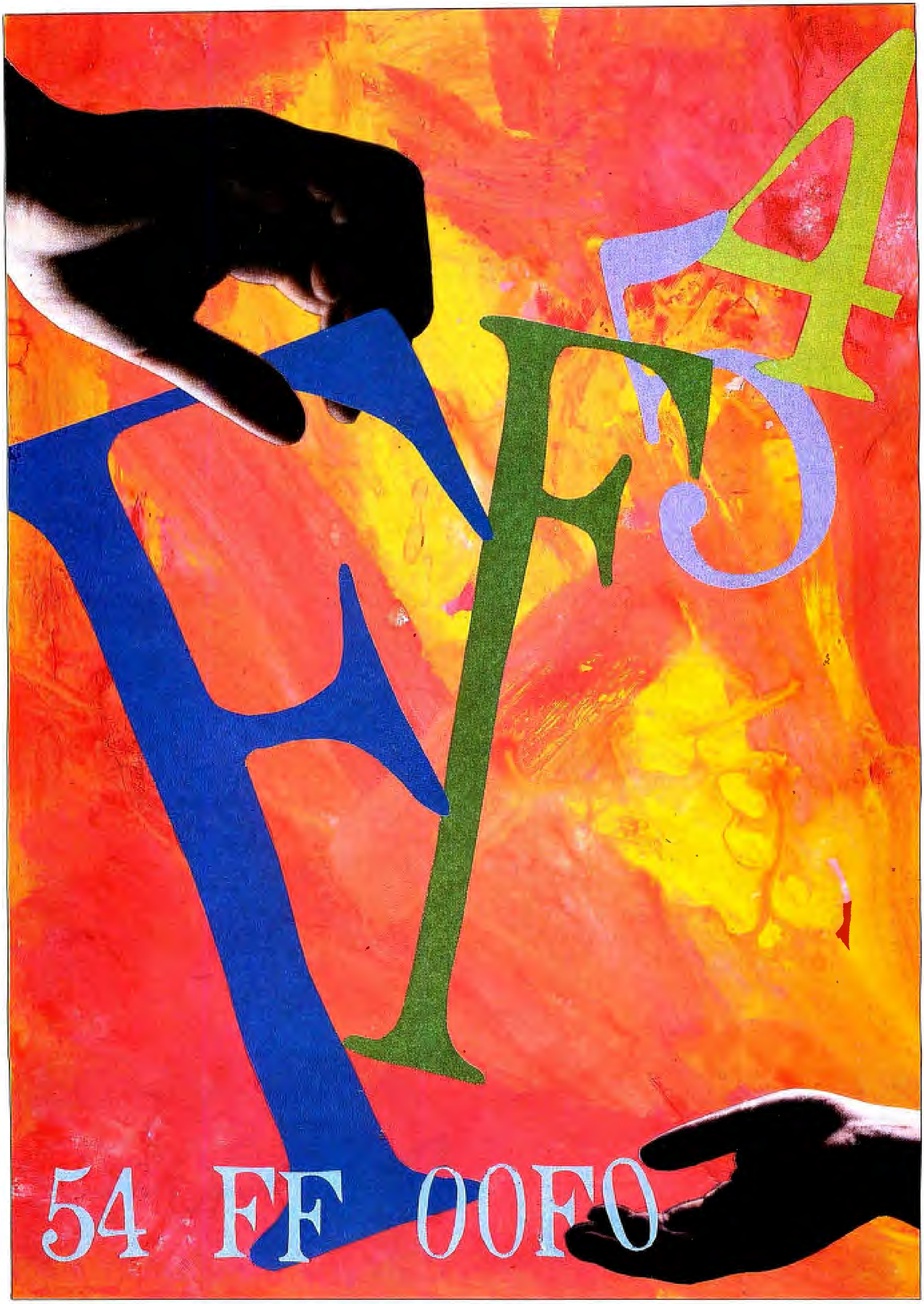
☐ Please send me a free DataEase™ demonstration disk.
☐ Please have a Software Solutions representative call.

Name__________ Title__________
Company__________ Phone__________
Address__________
City__________ State__________ Zip__________
No. of PCs on site ☐ 1-50 ☐ 51-250 ☐ 251+
☐IBM PC/XT/AT ☐IBM COMPATIBLES ☐WANG ☐DEC ☐OTHER

DATAEASE™

FROM SOFTWARE SOLUTIONS, INC.

dBASE is a trademark of Ashton-Tate



 ILLUSTRATED BY MIRIAM SCHAER

MEMORY MANIPULATIONS

PART 1: ARABIC VERSUS ROMAN

Making sense out of FATs and other reversed addresses

BY ALAN R. MILLER

Editor's note: Alan Miller explains two ways to manipulate the memory of the IBM PC. Part 1 clarifies the ordering of addresses. Part 2 discusses how to adjust the size of memory.

READING HEXADECIMAL numbers larger than one byte presents a problem that is rooted in the difference between the Arabic language, which is written from right to left and from which our numbers come, and the Roman language, which is written from left to right and from which our letters come. In our number systems—whether binary, hexadecimal, or decimal—the value of each digit position increases as you move from right to left, in the Arabic style. For example, in the decimal system, the tens are positioned to the left of the units, and the hundreds to the left of the tens.

Unfortunately, it has become the custom to display successive hexadecimal bytes in the Roman fashion, from left to right. Thus, although the 16-bit binary number 1010 1101 0101 0111 is equivalent to the hexadecimal number AD57, when this number is written as two bytes, the order is reversed to give 57 AD. Now the low byte, 57, is on the left, and the high byte, AD, is on the right. Notice the inconsistency of this form. The two characters of *each* byte are shown with the high nybble on the left and the low nybble on the right in Arabic style. However, *successive* bytes of the number are given with the low byte on the left and the high byte on the right in Roman style.

FINDING THE PORT ADDRESSES

You can observe the reversed display on the IBM PC by using the DOS program DEBUG. Execute DEBUG by typing its name and then pressing the Return key. When you see the minus-sign prompt, display two lines in hexadecimal notation by entering the command D0:400L20, then press the Return key. You should see something like the display in figure 1, corresponding to the first 32 bytes of the BIOS data area. The first symbols give the beginning address of 400 hexadecimal that was requested.

The first item at address 400 is the 16-bit address of the serial port. The two bytes are shown as F8 03. However, you need to reverse these two bytes to find the real address of the serial port, which is 3F8 hexadecimal. Similarly, the addresses of the two parallel ports start at address 408. After you reverse the byte pairs, the values BC 03 78 03 give the port addresses 3BC and 378. While this might seem confusing, the comforting rule is that the higher, and therefore more significant, half of the address resides higher in memory than the lower half.

Consider the next line at address 413. The two bytes at this location give the maximum amount of available RAM in K bytes. The reversed bytes 80 and 02 become 280 hexadecimal or 640K bytes.

THE INTERRUPT VECTORS

The first 1024 bytes of IBM PC memory (0 to 400 hexadecimal) contain 256 four-byte addresses that point to subroutines or data located somewhere in memory. These addresses are numbered from 0 through FF hexadecimal and are known as interrupt vectors. The hardware directly accesses some of these vectors. For example, when you type information at the keyboard, an electrical signal interrupts the microprocessor and directs it to the address given at 24 hexadecimal, which is for interrupt 9. The address there points to a keyboard-service routine in the ROM BIOS near the top of memory. After the keyboard routine completes its work, it returns control to the interrupted program.

If you use DEBUG to look at the first 12 interrupt vectors by giving the command D0:0L30, you will see something similar to figure 2. Each group of four bytes represents one interrupt vector. For example, interrupt 5 activates the PrtSc routine. When you hold down the Shift key and press the PrtSc key, the microprocessor branches to the address given at location 14 hexadecimal. The data there is shown as 54 FF 00 F0, which after reversal represents the address F000:FF54. Again, the display is inconsistent. The two characters in each byte have the higher half on the left and the lower half on the right in Arabic fashion. However, successive bytes are arranged with the higher bytes on the right and the lower bytes on the left in Roman fashion. Notice

(continued)

Alan R. Miller is a BYTE contributing editor and the author of Assembly Language Techniques for the IBM PC (Sybex, 1986). *These two articles are adaptations of information from that book. Alan is a professor at New Mexico Institute of Mining and Technology (Metallurgy Department, Socorro, NM 87801), where he has taught materials science, thermodynamics, electrical engineering, and programming methods. He holds a Ph.D. in engineering from the University of California at Berkeley.*

that if the display's second line were in reverse order (i.e., in Arabic fashion), you would see the contents of figure 3a. Then you could more easily read the four interrupt vectors shown in figure 3b. The address for interrupt 5 is now the second one from the right.

```
0000:0400  F8 03 F8 02 00 00 00 00-BC 03 78 03 00 00 00 00
0000:0410  7F 82 00 80 02 40 02 00-00 00 32 00 32 00 64 20
```

Figure 1: *An example of DEBUG's display of the first 32 bytes of the BIOS data area.*

```
0000:0000  E8 4E 2E 01 F0 01 70 00-C3 E2 00 F0 F0 01 70 00
0000:0010  F0 01 70 00 54 FF 00 F0-47 FF 00 F0 47 FF 00 F0
0000:0020  DF 01 6F 0B 2F 01 4A 10-DD E6 00 F0 DD E6 00 F0
```

Figure 2: *An example of DEBUG's display of the first 12 interrupt vectors.*

```
F0 00 FF 47 F0 00 FF 47-F0 00 FF 54 00 70 01 F0 0000:0010
```

Figure 3a: *The second line of figure 2 shown in reverse order (i.e., in Arabic fashion).*

```
F000:FF47  F000:FF47 – F000:FF54  0070:01F0
```

Figure 3b: *The four interrupt vectors described in figure 3a.*

```
xxxx:0170  C0 04 4D E0 04 4F 00 05-51 20 05 53 40 05 55 60
```

Figure 4a: *An example of DEBUG's display of the FAT. (The xxxx value is unimportant.)*

```
4C 4D 4E 4F 50 51 52 53 54 55
```

Figure 4b: *The actual FAT numbers shown in figure 4a.*

```
60 55 05 40 53 05 20 51-05 00 4F 04 E0 4D 04 C0
```

Figure 4c: *The FAT display from figure 4a shown backward.*

```
6 0 55 0 54 0 53 0 52 0 51-0 50 0 4F 0 4E 0 4D 0 4C 0
```

Figure 4d: *The display in figure 4c with the nybbles split. Note that the FAT for a floppy disk uses three nybbles per entry.*

THE FILE ALLOCATION TABLE

A more serious problem occurs with the file allocation table (FAT), a set of numbers that shows where each part of a file is located on the disk. Floppy disks and hard disks smaller than 20 megabytes use a 12-bit FAT, while larger disks use a 16-bit FAT. (A 20-megabyte hard disk uses a 12-bit FAT with DOS 2.x and a 16-bit FAT with DOS 3.x.) You can inspect the FAT for the floppy disk in drive A by giving the DEBUG command LDS:100 0 1 2. The first parameter to the load command is the offset address in memory where the FAT will be loaded (100 hexadecimal). The next parameter, 0, requests drive A; 1 is starting sector number, and 2 is the number of sectors.

After the FAT is loaded, you can examine it with the command D170L10. You will see something like the contents of figure 4a. Although it is not readily apparent, the FAT numbers in figure 4a are those in figure 4b. It is more obvious if you display the line backward (figure 4c) and then split the nybbles (figure 4d). Remember, the FAT for a floppy disk uses 12 bits, or three nybbles, per entry. When read from right to left (Arabic style), the bytes appear as in figure 4b.

Several authors have criticized the FAT's backward design. However, the design is correct; the problem lies in mixing Arabic and Roman notation. We have accepted the system of Arabic numerals, which is backward to us, without question. However, we insist on displaying successive bytes in Roman fashion from left to right. In this way, we scramble 12-bit, 16-bit, and 32-bit numbers in our displays.

Microsoft compromises by reversing the 16-bit addresses but then runs the bytes together to indicate that they are reversed. Consider a typical assembly listing that shows

```
E9 02DC    JMP done
```

The value E9 hexadecimal is the 8088 code for JMP, and the number 02DC hexadecimal is the relative distance to be jumped. However, if you look at a hexadecimal display of this instruction, you would see E9 DC 02. The DC

(continued)

AMERICA'S NEXT FAVORITE MOUSE.

The best price: $99

THE BEST HARDWARE:
- PC/XT/AT compatible through your serial port
- NO pad. NO power supply
- High (200 DPI) resolution
- Tactile feedback switches

THE BEST SOFTWARE:
LOGIMOUSE with Plus Package Software
$119

LOGIMOUSE® C7

- Driver 3.0: makes LOGIMOUSE run with all mouse based software, including Microsoft.
- LOGIMENU: a programmable Pop-Up Menu System that customizes LOGIMOUSE for each of your keyboard based applications.
- CLICK: a new concept in mouse software! It resides in memory, detects what application you're running and automatically sets the mouse to your predefined setting.
- POINT-AND-CLICK SHELL for Lotus 1-2-3: the only mouse interface for 1-2-3 that goes beyond keyboard emulation. With context sensitive pop-up menus and an independent mouse cursor that changes shape and meaning in different areas of the 1-2-3 screen. Makes it faster and easier to select a cell, invoke a command or scroll.
- POINT EDITOR: a mouse based program editor featuring pop-up menus and overlapping, color windows for faster, easier file editing. Use it instead of EDLIN or any other ASCII file editor.

THE BEST DEALS:

NEW! Introductory Price!
LOGIMOUSE with Plus Pkg. & Reflex ~~$268~~ $199
A winning combination! LOGIMOUSE enhances the power of Reflex, Borland's amazing database management system, to probe relationships in your data and display them in graphic form.

NEW! Introductory Price!
LOGICADD-LOGIMOUSE with Plus Pkg. & Generic CADD 2.0 with Dot Plot ~~$244~~ $189
Everything you need to turn your PC into a CAD workstation. Generic CADD has the features and performance of high-priced CAD, DotPlot turns your dot matrix printer into a plotter, and LOGIMOUSE is the ultimate input device.

NEW Reflex & LOGICADD Packages!

LOGIPAINT SET—LOGIMOUSE with Plus Pkg. & PC Paintbrush 3.0 ~~$258~~ $169
LOGIMOUSE plus PC Paintbrush 3.0 is the most advanced paint set available for the PC. Use it for designing a logo or painting a picture. You won't believe its power with either freehand drawing or graphics!

To place a credit card order call our special toll-free number:
800-231-7717

Call toll-free in California:
800-552-8885

YES! I want America's next favorite mouse! Please send me:

☐ LOGIMOUSE with Driver 3.0	$ 99	**30-Day Money-Back Guarantee/ New 3-Year Warranty**
☐ LOGIMOUSE with Plus Software	$119	
☐ Plus Package Software	$ 29	
☐ LOGIMOUSE with Reflex	$199	Add $6.50 for shipping & handling. Calif. residents add applicable sales tax.
☐ LOGICADD	$189	
☐ LOGIPAINT	$169	

For your computer model: ______

☐ VISA ☐ MASTERCARD ☐ CHECK ENCLOSED

Card Number ______ Expiration Date

SIGNATURE ______

NAME ______

ADDRESS ______

CITY, STATE ______

ZIP ______ PHONE ______

DEALER INQUIRIES WELCOME

LOGITECH

United States:
LOGITECH, Inc.
805 Veterans Blvd.
Redwood City, CA 94063
Tel: 415-365-9852

In Europe:
LOGITECH SA
CH-1143 Apples
Switzerland
Tel: 41-21-774545

Listing 1: ADUMP.ASM, *a program designed to display bytes in Arabic fashion from right to left. (Note that this is written to be executed in a* .COM *file.)*

```
        PAGE    ,132
        TITLE   ADUMP

COMMENT *
  Display a portion of memory in hex and ASCII
  Successive hex bytes are reversed in Arabic fashion
  Usage:
        ADUMP
        ADUMP  400      ;offset
        ADUMP  5300 0   ;segment, offset
  Macros: @d_char, @ucase, @write_r
  INT 21h function 8
 *
        IF1
        INCLUDE mymac.lib
        ENDIF

blank   EQU     32
cr      EQU     13
esc     EQU     1Bh
lf      EQU     10
period  EQU     46
up_arr  EQU     72                  ;up cursor
down_ar EQU     80                  ;down cursor
pg_up   EQU     73                  ;previous screen
pg_dn   EQU     81                  ;next screen

code    SEGMENT
        ASSUME  CS:code, DS:code
        ORG     5Dh
param1  DB      ?                   ;parameter 1
        ORG     6Dh
param2  DB      ?                   ;parameter 2

        ORG     100h
strt:
        @write_r <OFFSET mes1>
; set segment and offset to zero
        XOR     SI,SI               ;zero offset
        MOV     ES,SI               ;segment
; check if parameter was entered
        MOV     DI,OFFSET param1
        CMP     BYTE PTR [DI],blank ;anything?
        JZ      t_lines
; convert parameter from ASCII hex to binary in SI
        CALL    get_hex
        MOV     SI,DX
; check if second parameter was entered
        MOV     DI,OFFSET param2
        CMP     BYTE PTR [DI],blank ;anything?
        JZ      t_lines
        MOV     ES,SI
; convert parameter from ASCII hex to binary in SI
        CALL    get_hex
        MOV     SI,DX
;-----------------------------------------------
; start display of 16 lines
; ES has segment, SI has offset
;-----------------------------------------------
t_lines:
        @write_r <OFFSET mes2>
        MOV     CX,21               ;lines
```

(continued)

byte is located between the E9 and 02 bytes. This would be much clearer if we displayed the data with the order of the bytes reversed, as in

02 DC E9   JMP done

or

JMP done   02 DC E9

AN ARABIC DISPLAY

The assembly language program ADUMP.ASM (listing 1) is designed to make your hexadecimal displays easier to read by giving the bytes in Arabic fashion from right to left. To run ADUMP.ASM, you need the macros given in MYMAC.LIB (listing 2); notice that the macro names begin with @. (If you don't have a macro assembler, you must put the corresponding code at each place in the program. But then be careful to change the dummy variables and to remove the INCLUDE directive near the beginning. ADUMP.ASM is described in my book *Assembly Language Techniques for the IBM PC*, Sybex, 1986.) [*Editor's note*: ADUMP.ASM *and* MYMAC.LIB *are available on disk, in print, and on* BIX. *See the insert card following page 176 for details. Listings are also available on* BYTE*net. See page 4.*]

After you assemble ADUMP.ASM, if you execute it without parameters, you will see a listing similar to the one in table 1. The display begins at address 0, and so it includes the BIOS interrupt vectors. For example, the PrtSc vector at address 14 hexadecimal can now be clearly seen as F0 00 FF 54 or F000:FF54.

You can press the down arrow key to see the next line or the PgDn key to see the next page. To view another area of memory, execute this program with one or two parameters. If you provide only one parameter, the display begins at that hexadecimal offset from the beginning of memory. For example, to see the BIOS data area, you use the parameter 400. If you provide two parameters, the first is the segment address and the second is the offset within that segment.

ADUMP.ASM contains routines for

(continued)

Business Tools Introduces Reality Pricing.©

- Relational Database
- 4th Generation Language
- Screen Painter
- Program Generator
- Report Writer
- Source Code Editor
- Runtime Compiler

Reality Pricing means real features + real savings.

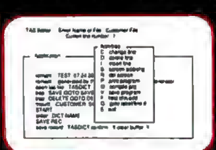
TAS-Plus just made it faster, easier and cheaper to build database applications.

TAS-Plus combines the power of a Relational Database with the ease of a Program Generator. Then TAS-Plus adds a Runtime Compiler to produce lightning-fast finished code.

Look at what TAS-Plus gives you:

Relational Database
4th Generation Language
Screen Painter
Program Generator
Report Writer
Source Code Editor
Runtime Compiler

TAS-Plus gives you power where it counts. You can store up to 65,000 records, open up to 16 files at a time and enter up to 10,000 characters per record. TAS-Plus even reads your old DBase files.



TAS-Plus writes the program for you

With TAS-Plus, you can start building professional database applications on day one. Just "paint" the screen the way you want and TAS-Plus writes the program for you. You can even paint using different colors or graphic characters. And custom reports are just as easy.

TAS-Plus has over 86 commands and 200 options available in its Source Code Editor, so you won't run out of room to grow.

Easy to use features

Add new databases quickly and easily.
Add, change or delete records without any programming at all.
Browse through your database and see multiple records on the screen at the same time.
Restructure capability allows you to change existing databases without loss of any data.

All this for just $69

TAS-Plus would be respectable at any price, at $69 it's awesome.

30 day Money Back Deal

TAS-Plus comes with a 30 day money back guarantee (less $15 handling fee). TAS-Plus is available for the IBM PC/XT/AT and fully compatible computers. TAS—Plus Accounting applications available.

Order Today 1-800-648-6258

Call our Toll-Free Hotline. Use your VISA, MasterCard or American Express to order today. For information or Washington residents call 1-206-644-2015.

The following are registered trademarks of these companies: TAS-Plus, The Accounting Solution, Business Tools, Inc; IBM PC/XT/AT, International Business Machines Corp;

Inquiry 48

TAS+™

NOT COPY-PROTECTED

YES! Rush me the following items.

Qty.	Item	Price	Subtotal
______	TAS+	$69	______
Shipping add $8 USA, $25 outside USA		Shipping	______
WA res add $5.59 Tax		Tax	______
		Amount Enclosed (US funds only)	______

Name: ______
Shipping Address: ______
City: ______
State: ______ Zip: ______
Telephone: ______
Payment: VISA MC AMX Check
Credit Card Expiration Date: ______
Card Number: ______
Name on Card: ______

BUSINESS TOOLS INC.

4038-B 128th Ave. S.E., Suite 266
Bellevue, Washington 98006
(206) 644-2015

VISA MasterCard

```
n_line:
        @d_char cr
        @d_char lf
; display address and 16 hex bytes
        CALL    do_hex
; display ASCII
        CALL    do_ascii
        ADD     SI,16           ;next line
        LOOP    n_line
; wait for keyboard input
        CALL    get_in
        CMP     AL,esc          ;quit?
        JZ      done            ;yes
        CMP     AL,cr           ;next page?
        JNZ     not_cr          ;no
        JMP     t_lines
not_cr:
        CMP     AL,pg_dn
        JNZ     not_pd
        JMP     t_lines         ;next page
not_pd:
        CMP     AL,up_arr
        JNZ     not_ua          ;prev line
; back up pointer one line and redisplay
        SUB     SI,22*16
        JMP     t_lines
not_ua:
        CMP     AL,pg_up
        JNZ     not_tl          ;prev page
; back up pointer one page and redisplay
        SUB     SI,42*16
        JMP     t_lines
not_tl:
        MOV     CX,1
        CMP     AL,blank
        JZ      n_line          ;next line
        CMP     AL,down_ar
        JZ      n_line          ;next line
done:
        INT     20h
; translation table binary to ASCII hex
hex_tab DB      '0123456789ABCDEF'
;*******************************************
; Convert binary number in AL to hex and
;  and display on screen
; Input:        AL = byte
;*******************************************
outhex  PROC
        PUSH    DX
        MOV     DL,AL           ;save
        REPT    4               ;times 16
        SHR     AL,1
        ENDM
        CALL    outhx           ;high half
        MOV     AL,DL           ;get back
        CALL    outhx           ;low half
        MOV     AL,DL           ;restore
        POP     DX
        RET
outhex  ENDP
;********  End outhex  ******************
;******************************************
; convert four bits in AL to hex and
; display on screen
;******************************************
```

```
outhx   PROC
        AND     AL,0Fh              ;low 4 bits
        PUSH    BX
        MOV     BX,OFFSET hex_tab
        XLAT    hex_tab
        @d_char AL                  ;display
        POP     BX
        RET
outhx   ENDP
;********  End outhx  ********************
;*****************************************
; Convert ASCII hex to binary
; Input:        DI points to string
; Output:       DX = binary value
;*****************************************
get_hex PROC
        PUSH    AX
        XOR     AX,AX               ;zero
        MOV     DX,AX               ;DX too
g_hex2:
        MOV     AL,[DI]
        SUB     AL,'0'              ;make binary
        JL      g_hex_b             ;too low
        CMP     AL,10
        JL      g_hex_g             ;ok
        @ucase                      ;make upper
        SUB     AL,'A'-'9'-1        ;adjust for A-F
        JL      g_hex_b             ;too low
        CMP     AL,15
        JG      g_hex_b             ;too large
; add new character to sum
g_hex_g:
        REPT    4
        SHL     DX,1
        ENDM
        ADD     DX,AX
        INC     DI
        JMP     g_hex2              ;next char
; end of string
g_hex_b:
        POP     AX
        RET

get_hex ENDP
;*************  End of get_hex  **********
;*****************************************
; display address and 16 hex bytes
; Input:        SI = pointer
; Output:       SI is 16 larger
;*****************************************
do_hex  PROC
        MOV     DI,SI
        ADD     DI,16
        PUSH    CX
        MOV     CX,16               ;one line
next_ch:
        DEC     DI                  ;next
        MOV     AL,ES:[DI]
        CALL    outhex
        PUSH    BX
        MOV     BL,blank
; put a minus sign at quarter points
        MOV     AX,DI
        AND     AL,0fh
        JZ      not_min             ;line end
        AND     AL,3
```

(continued)

```
        JNZ     not_min
        MOV     BL,'-'              ;minus
not_min:
        @d_char BL
        POP     BX
        LOOP    next_ch
        POP     CX
        @d_char blank
; display address at right side
        PUSH    BX
; display segment address
        MOV     BX,ES
        MOV     AL,BH
        CALL    outhex              ;high half
        MOV     AL,BL
        CALL    outhex              ;low half
        @d_char ':'                 ;colon
; display offset address
        MOV     BX,SI               ;offset
        MOV     AL,BH
        CALL    outhex              ;high half
        MOV     AL,BL
        POP     BX
        CALL    outhex              ;low half
        RET
do_hex  ENDP
;************* End of do_hex ************
;*****************************************
; display 16 ASCII bytes
; control characters shown as period
; extended character translated to ASCII
;
; Input:        SI = pointer
;*****************************************
do_ascii PROC
        @d_char blank
        @d_char blank
        MOV     DI,SI
        PUSH    CX
        MOV     CX,16               ;one line
next_as:
        MOV     AL,ES:[DI]
        AND     AL,7Fh              ;make ASCII
        CMP     AL,7Fh              ;DEL?
        JZ      d_per               ;yes
        CMP     AL,blank            ;too small?
        JNB     nbl
d_per:
; change to period
        MOV     AL,period
nbl:
        @d_char AL
        INC     DI                  ;next
        LOOP    next_as
        POP     CX
        RET
do_ascii ENDP
;************* End of do_ascii **********
;*****************************************
; wait for character from keyboard
; don't echo on screen
; if extended code, two calls are made
;
; Output:       AL = character
;*****************************************
```

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS MAIL PERMIT NO. 39 MARTINSVILLE, NJ

POSTAGE WILL BE PAID BY ADDRESSEE

BYTE Subscriptions
PO Box 597
Martinsville, NJ 08836-9956

```
get_in  PROC
        MOV     AH,8
        INT     21h
        OR      AL,AL              ;zero?
        JNZ     get_2              ;no
; get extended code
        INT     21h
get_2:
        RET
get_in  ENDP
;************* End of get_in  **********
mes1    DB      '    Arabic display--Hex goes from'
        DB      ' right to left',cr,lf
        DB      ' Active keys are: down and up'
        DB      ' arrows, PgDn, PgUp, and Esc$'
mes2    DB      cr,lf,lf,' F  E  D  C  B  A  9  8'
        DB      '  7  6  5  4  3  2  1  0  SEGM'
        DB      ' ADDR  0123456789ABCDEF$'

code    ENDS
        END     strt
```

Table 1: *Arabic display of* BIOS *interrupt vectors. If you execute the* ADUMP *program without parameters, you will see an output listing similar to this.*

Arabic display—Hex goes from right to left
Active keys are: down and up arrows, PgDn, PgUp, and Esc

F	E	D	C B	A	9	8 7	6	5	4 3	2	1	0	SEGM ADDR
00	70	01	F0-F0	00	E2	C3-00	70	01	F0-01	2E	4E	E8	0000:0000
F0	00	FF	47-F0	00	FF	47-F0	00	FF	54-00	70	01	F0	0000:0010
F0	00	E6	DD-F0	00	E6	DD-10	4A	01	2F-0B	6F	01	DF	0000:0020
00	70	01	F0-F0	00	EF	57-C8	00	08	49-F0	00	E6	DD	0000:0030
00	70	09	AB-F0	00	F8	41-F0	00	F8	4D-F0	00	F0	65	0000:0040
0B	6F	01	64-F0	00	E8	2E-F0	00	F8	59-F0	00	E7	39	0000:0050
00	70	01	EA-F0	00	FE	6E-00	70	0A	DD-F6	00	00	00	0000:0060
F0	00	00	00-00	00	05	22-F0	00	F0	A4-F0	00	FF	53	0000:0070
0A	9A	01	3C-0A	9A	01	2F-10	67	03	5C-01	2E	12	C3	0000:0080

Listing 2: MYMAC.LIB, *a macro-definition library.*

```
@d_char MACRO   par?
;; display par? on video screen
;; Usage:   @d_char  AL          ;;from register
;;          @d_char  '*'         ;;constant
        PUSH    DX
        MOV     DL,par?
        MOV     AH,2
        INT     21h
        POP     DX
        ENDM
@ucase  MACRO
        LOCAL   notup?
;; Macro to change AL to upper case
;; Usage:  @ucase
        CMP     AL,'a'           ;;uppercase?
        JB      notup?           ;;yes
        AND     AL,5Fh           ;;make upper
notup?:
        ENDM
```

(continued)

```
@write  MACRO    text?
        LOCAL    around,mesg
;; Macro to embed version number
;; Usage:        @write  'title, date'
        PUSH     DX
        PUSH     AX
        MOV      AH,9             ;;write string
        MOV      DX,OFFSET mesg
        INT      21h
        POP      AX
        POP      DX
        JMP      SHORT around

mesg    DB       text?,'$'
around:                   ;;write
        ENDM
@write_r MACRO   addr?
;; display text at location addr?
;; text must end with $
;; Usage:    @write_r <OFFSET p_ver>

        PUSH     DX
        MOV      DX,addr?
        MOV      AH,9
        INT      21h
        POP      DX
        ENDM
```

converting binary to hexadecimal and hexadecimal to binary. The binary-to-hexadecimal routine uses the 8088 XLAT instruction and a simple table of hexadecimal values for the conversion.

You can back the display up one line by pressing the up arrow key and back it up one page by pressing PgUp. Unfortunately, these operations refresh the entire screen, so they are time-consuming. It would be faster if the information were written directly to the video screen; then the screen would be instantly updated. This technique is also described in my book.

FINALLY

There is considerable confusion between Arabic and Roman styles in reading hexadecimal addresses. However, with the help of ADUMP.ASM or a similar program, you can make such addresses more logically readable.

PART 2: ADJUSTING MEMORY SIZE

Making it smaller—or larger

IT IS POSSIBLE to decrease or increase the amount of memory available for program execution on the IBM PC. Everyone can find reasons for increasing memory, but there are also reasons why you might want to decrease it.

REDUCING THE MEMORY SIZE

Although the IBM PC can, in principle, address about 1 megabyte of memory, DOS versions 2.x and 3.x are limited to a maximum of 640K bytes for program execution. Many programs can use all 640K bytes. However, while several commercial programs won't operate with this much memory, they will run correctly with less, usually 512K bytes or fewer. Furthermore, if you attempt to run one of these programs with the full 640K bytes, you will get the confusing error message "Insufficient memory." You can solve this problem by reducing the effective memory size of your PC temporarily while you run these programs.

One way to reduce memory size is to install a RAM disk. However, if you have a PC AT, it would be better to install a RAM disk in addition to the regular 1 megabyte of memory using a board such as the AST Advantage. Then you can still have 640K bytes of RAM for program execution.

You can also reduce the memory size by creating a resident program. When a program begins execution, DOS gives it all the remaining RAM. If that program terminates normally through interrupt 20h or function 4Ch of interrupt 21h, DOS recovers the allocated RAM for the next program's use. However, if a program terminates with DOS interrupt 27h, DOS reserves a portion of memory, and the effective memory available for the next program is reduced accordingly. This technique is commonly utilized for installing device drivers such as a print spooler or a RAM disk program.

The program in listing 3, FILL.ASM, allocates a 64K-byte block of memory; it does not execute any code in this space. You also need the macro library, MYMAC.LIB (see listing 2 in "Arabic versus Roman" on page 241), since the program uses the macro @write. [*Editor's note*: FILL.ASM, MYMAC.LIB, *and* MEM704.ASM *are available in a variety of formats. See the insert card following page* 176 *for details.*] If you have 640K bytes of memory available, you might occasionally need to run this program.

You can run the DOS CHKDSK.COM program to determine the amount of available memory. For example, if you have 640K bytes, you will see a display such as

```
655360 bytes total
  memory
618464 bytes free
```

If you assemble and execute FILL.ASM, you will see the expression "Memory reduced 64K bytes" on the screen. You should run the CHKDSK.COM program again to verify that the memory has been reduced:

```
655360 bytes total
  memory
552912 bytes free
```

FILL.ASM begins with assembler directives and comments; the INCLUDE directive tells the assembler to look in the library you created, MYMAC.LIB, for the macros. Since FILL.ASM assembles into a .COM file, you set the instruction pointer to 100 hexadecimal with the ORG statement. The executable instructions begin with the @write macro, which displays the "Memory reduced . . ." message.

The program contains only two more microprocessor instructions: MOV and INT. MOV places the resident code length in the DX register; INT terminates the program with DOS interrupt 27h, leaving 64K bytes resident. The assembler calculates the resident length by using the labels pg_len and strt.

INCREASING THE MEMORY TO 704K BYTES

The maximum amount of RAM that DOS can normally use is 640K bytes, ending with segment 9000 hexadecimal. Since the video-screen memory

(continued)

ILLUSTRATED BY MIRIAM SCHAER

begins at segment B000, an unused 64K-byte block of memory is at segment A000. The program MEM704.ASM (listing 4) lets you expand DOS into this segment so that 704K bytes are available to your programs. Of course, you can't use this program unless you have memory in segment A000. Therefore, you might need to add another memory board. Some memory boards, such as the AST SixPak, can't be addressed in this region, while others, like the IBM 64/256 board, can. After adding the new memory and addressing it to the A000 segment, you must run MEM704.ASM to reset the internal indicators for maximum RAM memory to 704K bytes. You don't need to change the switches on your system board that specify memory size.

MEM704.ASM does several things. It increases the maximum memory size, stored in the BIOS data area at 413 hexadecimal (or 40:13), and sets the new memory to zero to avoid parity errors. After making these changes, the program initiates interrupt 19h so the PC will reload DOS. Finally, it checks to make sure that it doesn't perform these services a second time.

First, MEM704.ASM establishes the segment in the BIOS data area at 40 hexadecimal (bios_d). Then it defines the offset (in this segment) of the maximum memory size stored at location 13 hexadecimal (mem_siz). The code segment begins by checking the word at 413 to see if the value stored there is already 704K (2C0 hexadecimal). (This will be the case after the program has reset the computer with interrupt 19h.) If so, the program terminates because this is the second time it has been run. However, if this is the first time the program has executed, it checks to see if the computer has 704K bytes of memory installed.

To determine whether any memory exists at the A000 segment, the program writes the pattern 55 hexadecimal, an alternating sequence of zeros and ones, at the beginning of the A000 segment. Then it reads back the value to see if it is correct. Next, it writes the value AA, an alternating se-

Listing 3: FILL.ASM, *a program to reduce the size of addressable memory. (Note that this is written to be executed in a .COM file.)*

```
        PAGE    ,132
        TITLE   FILL part of memory

COMMENT *
  Program to fill part of memory with
  resident program (some programs will
  not run when there is too much memory).
  Macros: @write
  INT 27h
 *
cr      EQU     13
lf      EQU     10

        IF1
        INCLUDE mymac.lib
        ENDIF

code    SEGMENT
        ASSUME  CS:code, DS:code
        ORG     100h

strt:
   @write    <cr,lf,' Memory reduced 64K bytes'>
;-------------------------------------------------
; quit and stay resident
;-------------------------------------------------
        MOV     DX,OFFSET strt-OFFSET pg_len
        INT     27h                 ;done
pg_len  LABEL   BYTE

code    ENDS
        END     strt
```

Listing 4: MEM704.ASM, *a program to increase addressable memory from 640K to 704K bytes. (Note that this is written to be executed in a .COM file.)*

```
        PAGE    ,132
        TITLE   704K - set memory to 704K bytes
COMMENT *
   Increase memory to 704K bytes by changing
   BIOS data word at 40:13 hex
   Set new memory to zero
 *
cr      EQU     13
lf      EQU     10

        IF1
        INCLUDE mymac.lib
        ENDIF

old_siz EQU     640             ;old mem size
new_siz EQU     704             ;new mem size
new_seg EQU     0A000h          ;704K segment

; BIOS data area
bios_d  SEGMENT AT 40h
        ORG     13h
mem_siz LABEL   WORD            ;memory size
bios_d  ENDS
```

You can increase or decrease the PC's memory to do what you want.

```
code     SEGMENT
         ASSUME  CS:code,DS:code, ES:bios_d
         ORG     100h
strt:
; point to memory size in BIOS data area
         MOV     AX,bios_d            ;zero
         MOV     ES,AX
         MOV     SI,OFFSET mem_siz
; check if already run
         CMP     WORD PTR ES:[SI],new_siz
         JNE     ok
         INT     20h                  ;quit
; check for memory at A000 hex
ok:
         MOV     BX,ES                ;save ES
         MOV     AX,new_seg
         MOV     ES,AX                ;ES=A000
         XOR     DI,DI                ;offset
         MOV     AX,55AAh             ;01010101
         MOV     ES:[DI],AH           ;55 hex
         CMP     AH,ES:[DI]           ;ok?
         JNZ     no_mem               ;no
         MOV     ES:[DI],AL           ;AA hex
         CMP     AL,ES:[DI]           ;ok?
         JNZ     no_mem               ;no
; set memory size for 704K bytes
         MOV     CX,ES                ;save ES
         MOV     ES,BX                ;ES=40
         MOV     WORD PTR ES:[SI],new_siz
         MOV     ES,CX                ;ES=A000
; fill new memory with zeros
         XOR     AX,AX                ;zero word
         MOV     DI,AX                ;zero location
         MOV     CX,8000h             ;32K words
         REP     STOSW                ;do it
         @write  <' Memory now 704K bytes',cr,lf>
         INT     19h                  ;reboot system
; no memory at A000 hex, terminate without change
no_mem:
         @write  'No memory at A000 hex, aborting'
         INT     20h

code     ENDS
         END     strt
```

quence of ones and zeros, and again checks for the correct value. If the program can't read back the values it stored, it concludes that there is no memory at the A000 segment. The program displays an error message and aborts.

If 704K bytes of memory are present, the program changes the BIOS word at 413 to 2C0, sets the memory in segment A000 to zero, writes the message "Memory now 704K bytes" on the screen, and reboots with interrupt 19h.

This "new" memory needs to be set to zero because the IBM PC uses parity-checked memory. Each 8-bit byte has an associated ninth bit that is used for a parity check. The sum of the 9 bits *must* have zero parity. When any byte is read, the parity bit is checked to see that there is no error. However, this scheme works properly only if the parity bits are set correctly on start-up. You can accomplish this by writing a zero in each memory location.

The system sets memory parity only up through 640K bytes. Therefore, the new memory in the A000 segment will not automatically be set to even parity. If you try to read from the new memory before setting its parity, you will get a parity error and your computer will lock up. However, if you first fill the new memory with zeros, it will pass subsequent parity checks.

MEM704.ASM sets the new 64K bytes of memory to zero with the store-string instruction (STOSW). It points the ES:DI registers to the beginning of segment A000, sets the AX register to zero, sets the CX register to 64K bytes (8000 hexadecimal, or 32K, words), and combines the REP instruction with the STOSW string instruction. After you run MEM704.ASM, your programs can use as much as 704K bytes of memory. You might want to establish a 360K-byte RAM disk with the DOS VDISK program. If you do, you will still have 344K bytes of usable memory.

If you run CHKDSK.COM after establishing a 360K-byte RAM disk, you will see a display such as

```
720896 bytes total memory
281472 bytes free
```

Since 1K bytes equals 1024 or 2^{10} bytes, 704K bytes equals 720,896 bytes. You can also think of this as 11 segments (0 through A) of 64K bytes, or 65,536 bytes, each.

While this technique works properly with both the monochrome and the graphics adapter, there is a potential problem. The IBM Extended Graphics Adapter uses part of the A000 segment. Therefore, you cannot use this technique when you have an EGA installed in your computer.

In Conclusion

As you can see, you can adjust the amount of memory on the IBM PC to fill your needs. Within limits, you can increase it—or decrease it—to do what you want to do. You aren't necessarily stuck with what comes on the system. ■

Before you consider the new Hercules Graphics Card Plus, consider the technology behind it.

On July 1, Hercules™ introduced a product that will forever change the way information is displayed on a PC.

The product is called the Hercules Graphics Card Plus.

We gave it that name because it gives you the same hi-resolution text and 720x348 graphics that made the original Hercules Graphics Card famous.

Plus it gives you RamFont.™

RamFont is a radical new hardware mode that combines the speed of text mode with the flexibility of graphics mode.

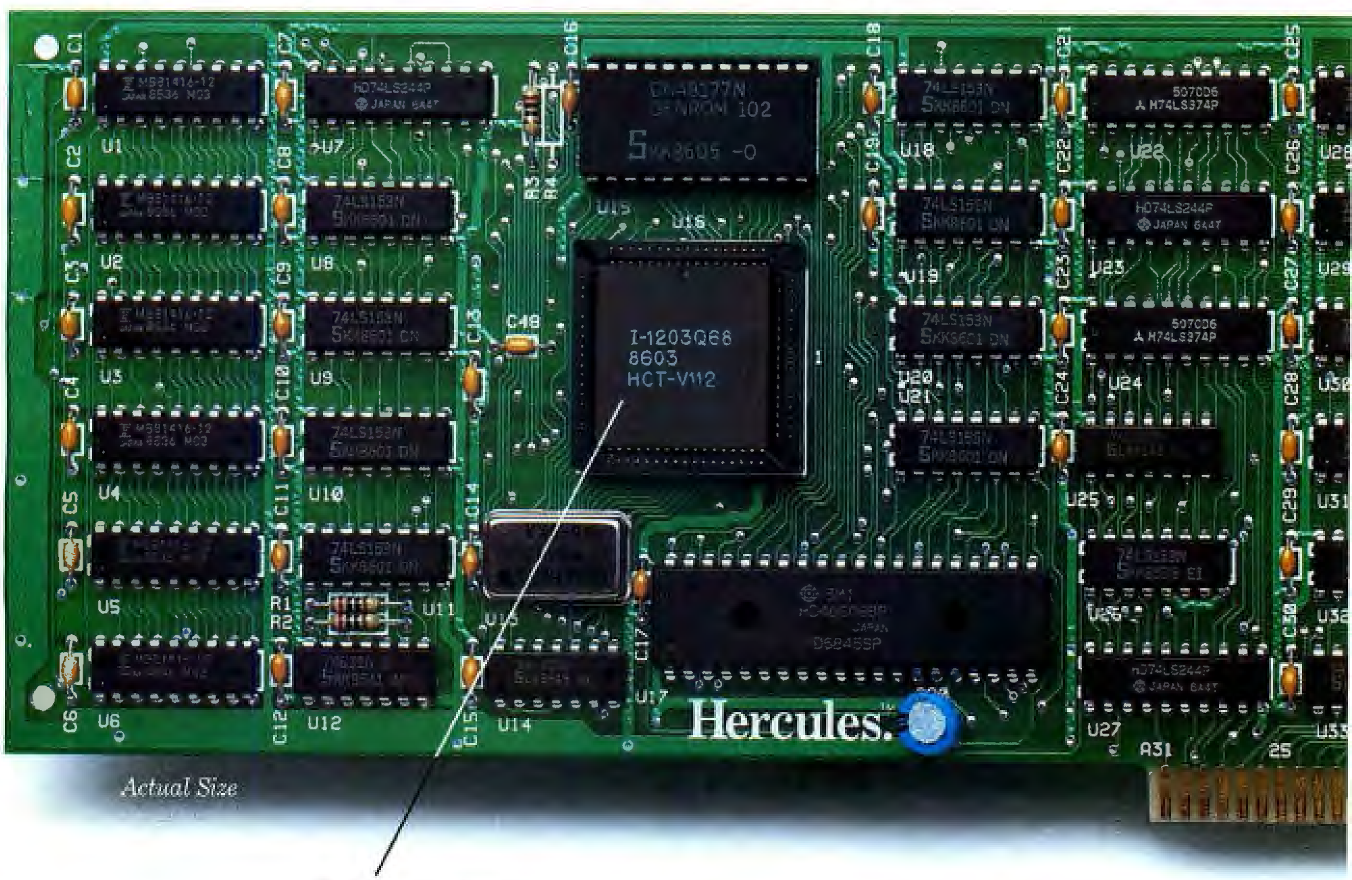
And opens up a whole new world for software.

The world according to RamFont.

In the old days (before July 1), programs like Lotus® 1-2-3®, Symphony,™ Framework™ and Microsoft® Word had to use graphics mode to display multiple fonts and variable text sizes, or to mix text with graphics.

But graphics mode is a whole lot slower than text mode. Up to eight times slower.

Enter RamFont.



The heart of the Graphics Card Plus: the V112 microchip, Hercules' next generation video processor that makes the RamFont mode possible.

It lets all the programs we just mentioned (and plenty more in the future) do all the things we just described, all on one screen.

At precisely the same speed as text mode.

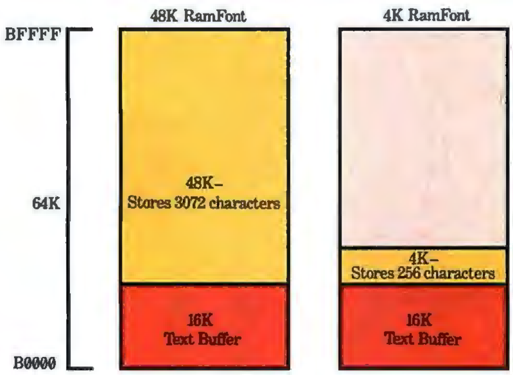
How RamFont works.

Like text mode, RamFont uses a 16-bit word to represent a character on the display.

Unlike text mode, however, the 48K RamFont mode uses a 12-bit character code instead of an 8-bit code.

Which allows you to choose from an astonishing 3072 different characters.

While setting the size of your screen cells from eight to nine pixels wide and from four to 16 scan lines tall.



The new RamFont mode displays 3072 programmable characters at the speed of text mode, or replaces the standard character set with one of your choice in 4K RamFont mode.

Another Hercules innovation: the new LPT112. It's a parallel printer port on a chip.

To help you design your own RamFont characters and symbols, we've included a font editor called FontMan.™

Along with a set of 25 sample fonts to start your library.

And since the 4K RamFont mode can accept 8-bit character codes, running your text mode software with your favorite font is as simple as loading it into RAM.

Just for the record, RamFont supports the standard character attributes of reverse, high-intensity, blink and underline.

Plus two new RamFont attributes: boldface and strike-through.

What price success?

While we're on the subject of technological breakthroughs, let's not forget the suggested retail price.

$299.
Suggested retail price.

Think of it. You get everything the original card gave you.

Plus RamFont.

Plus FontMan.

Plus a free printer cable.

Plus a parallel printer port that you can now disable if there's a conflict with another port in your system.

All for just $299—about half the price of the original Graphics Card.

If you haven't gone into shock, call 1-800-532-0600 ext. 212 for the name of an Authorized Hercules Dealer near you and we'll rush you our free info kit.

Hercules.™

We're strong on graphics.

Address: 2550 Ninth Street, Berkeley, California 94710 Ph: 415 540-6000
Telex: 754063

Trademarks/Owners: Lotus, 1-2-3, Symphony/Lotus; Framework/Ashton-Tate; Microsoft/Microsoft; Hercules, RamFont, FontMan/Hercules

Inquiry 123

Inquiry 283 for End-Users. Inquiry 284 for DEALERS ONLY.

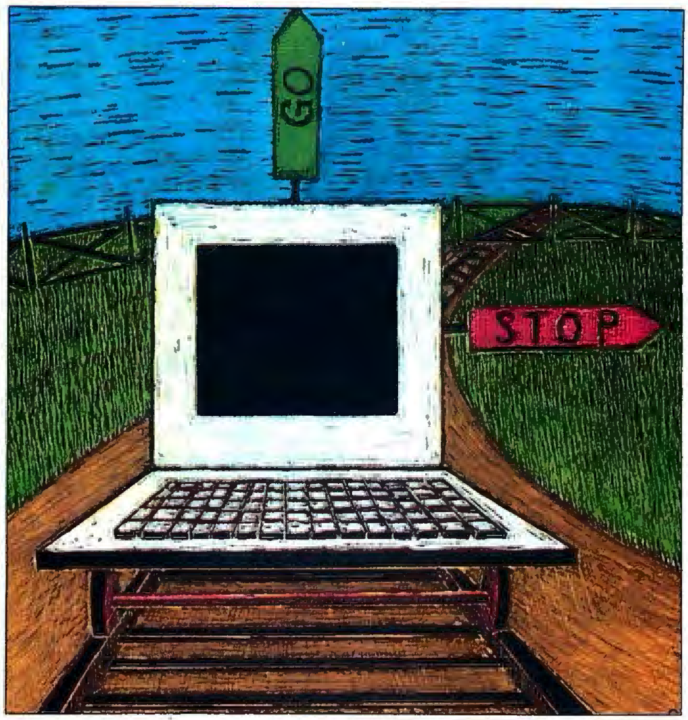
WRITING ASSEMBLY LANGUAGE INTERRUPT ROUTINES

A description of interrupts on the PC and examples of some interrupt handlers

BY WILLIAM J. CLAFF

THIS IS A GUIDE for IBM PC programmers interested in writing assembly language interrupt routines. I will briefly discuss the concept of an interrupt, followed by some programming examples.

You can think of interrupt routines as subroutines called by the hardware or the operating system. Implementing interrupt routines is an advanced topic in programming that requires a high degree of understanding of the computer's existing interrupt structure and hardware. I am assuming that you are familiar with the basics of assembly language programming for the IBM PC.



INTERRUPTS ON THE 8088

The Intel 8088 microprocessor has 256 interrupts that are dispatched through the interrupt-vector table located at 0000:0000 hexadecimal. This table is treated as a zero-based array of the interrupt routines' double-word addresses. The two types of interrupts are hardware and software.

The hardware interrupt is aptly named. The CPU, or external circuitry attached to the CPU, generates hardware interrupts. When a hardware interrupt occurs, it suspends the code that is currently executing so the interrupt routine can be called.

The software interrupt is not so aptly named. The execution of the INT instruction generates software interrupts. Since the code is not suspended at an arbitrary location as it is during a hardware interrupt, the term "interrupt" is a misnomer. In effect, INT behaves much like a subroutine call.

Many early microcomputer systems had no software-interrupt capability. This concept was first introduced in the Intel 8080 with the RST (restart) instruction. The 8080's interrupts are usually controlled by an Intel 8259 Programmable Interrupt Controller (PIC) that lets you associate a hardware interrupt with each of the eight RST interrupts. Intel took a different approach with the 8085; it included three hardware interrupts with no corresponding software interrupt. It wasn't until the 8086 that Intel recognized the value of stand-alone software interrupts.

HARDWARE INTERRUPTS

Hardware interrupts are either internal or external. Internal hardware interrupts are generated from within the CPU; external hardware interrupts are generated by circuitry outside the CPU. On the IBM PC, the 8259 PIC manages external interrupts.

SOFTWARE INTERRUPTS

Software interrupts have three levels: BIOS interrupts, DOS interrupts, and application-program interrupts.

IBM intended the BIOS interrupts to be the sole means of accessing the hardware. Unfortunately, due to incomplete functionality and speed considerations, many programs manipulate the hardware directly. Such programs are less likely to work with "IBM-compatible" computers.

DOS interrupts are intended as the sole means of accessing the DOS

(continued)

William J. Claff (7 Roberts Rd., Wellesley, MA 02181) is a senior technical engineer with SoftSet Associates. He has run his own microcomputer retail and consulting business since 1978. Mr. Claff, who holds an M.S. in applied mathematics from Harvard University, publishes an IBM technical newsletter and is the leader of the IBM PC Technical Subgroup of the Boston Computer Society's IBM PC Users Group.

ILLUSTRATED BY CLAUDIA TANTILLO

The interrupt-intercept routine remembers the original interrupt routine.

resources. Fortunately, most programs abide by the intent of the DOS software-interrupt interface.

The application-program interrupts are those used by programs executed by the DOS (e.g., BASIC). You must be aware of these interrupts in order to permanently install a new interrupt routine. Although many interrupts are unused, only interrupts 60h through 67h are officially available.

INTERRUPT LEVELS

To illustrate the various interrupt levels, I want to consider the PC's keyboard. The three keyboard-related interrupts are the external hardware interrupt, the BIOS software interrupt, and the DOS software interrupt.

A DOS software interrupt that involves the keyboard (e.g., function 1h) will use the BIOS software interrupt 16h to interface with the hardware. The INT 016H routine polls memory locations it shares with hardware interrupt 9h. When you press a key, you generate an interrupt 9h. The INT 009H routine stores the keystroke in the shared buffer; the INT 016H routine senses the arrival of the keystroke and returns the key to DOS.

REPLACEMENT VS. INTERCEPT

An interrupt-replacement routine is one that replaces an existing interrupt routine. An INT 023H (control/break) handler would be of this type.

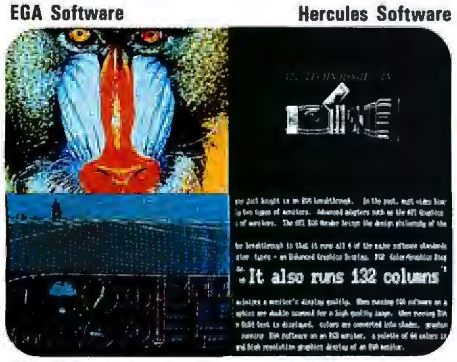
Rather than replacing an existing interrupt routine, the interrupt-intercept routine remembers the original interrupt routine. It might or might not pass control through to the original routine. Most keyboard macro programs are of this type. They do not replace the BIOS INT 009H routine (the keyboard-interrupt handler) but field the hardware interrupt first and decide whether or not to use the

(continued)

EGAWONDER™

Any Software. Any Monitor. Any Time.

EGA Software — Hercules Software



CGA Software — 132 Column Software

EGA Monitor

EGA Wonder runs EGA, CGA, MDA, Hercules and 132 Columns on an Enhanced Graphics Display. CGA text is improved to 8 × 14 and graphics are double scanned for a high quality display.

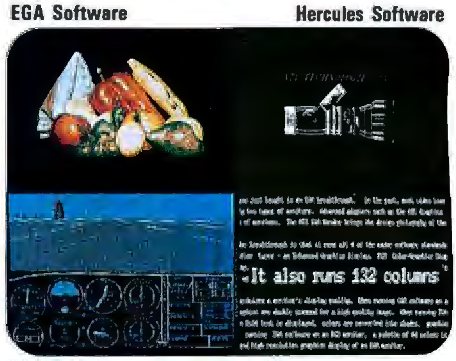
EGA Software — Hercules Software

CGA Software — 132 Column Software

TTL Monochrome Monitor

EGA Wonder runs EGA, CGA, MDA, Hercules and 132 Columns on a TTL Monochrome Monitor. Colors of EGA and CGA are converted into shades, graphics are full screen and no pre-boot drivers are required.

EGA Software — Hercules Software



CGA Software — 132 Column Software

RGB, *Composite Monitors, and the IBM Portable P.C.

EGA Wonder also runs EGA, CGA, MDA, Hercules and 132 Columns on an RGB Monitor, *Composite Monitor, and the IBM Portable P.C. in 64 colors (shades). EGA and Hercules software are interlaced for high resolution text and graphics.

* Composite Monitor not shown

Features	ATI EGA Wonder	Paradise Auto Switch	Quadram and Video 7
Compatible to EGA, CGA, MDA, Hercules	✓	✓	✓
256K Video Memory	✓	✓	✓
Automatic Switching Between EGA and CGA Color Modes and Among EGA, MDA, and Hercules Modes	✓	✓	
Runs EGA, CGA, MDA, Hercules and 132 Columns on an EGA Monitor	✓		
Runs EGA, CGA, MDA, Hercules and 132 Columns on an RGB Color Monitor	✓		
Runs EGA, CGA, MDA, Hercules and 132 Columns on a TTL Monochrome Monitor	✓		
Runs EGA, CGA, MDA, Hercules and 132 Columns on a Composite Monitor	✓		
Runs EGA, CGA, MDA, Hercules and 132 Columns on an IBM Portable P.C.	✓		
Warranty	2 Yrs	1 Yr	1 Yr
Suggested List Price	$399	$599	$595

EGA — Enhanced Graphics Adapter MDA — Monochrome Display Adapter
CGA — Color Graphics Adapter

*Trademarks: Paradise, Auto Switch — Paradise Systems Inc; Quad EGA+ is a trademark of Quadram Corp.; Video-7-Video-7, Incorporated. IBM Portable P.C.— IBM — International Business Machines

Upgrade to EGA Without An EGA Monitor

Only EGA Wonder transcends normal hardware compatibility by displaying EGA, MDA, CGA, Hercules and/or 132 columns, on any of the four major monitor types.

True Hardware Level Compatibility

No incompatibilities, no limitations — EGA Wonder is compatible at the hardware level to EGA, CGA, MDA, and Hercules.

Automatic Mode Switching

No mode commands, no special software — EGA Wonder can sense and automatically switch between EGA and CGA color modes or between MDA, Hercules and EGA monochrome modes.

The Unique Solution

Only EGA Wonder runs any software, any monitor, anytime with true hardware level compatibility and automatic mode switching. Call us today at *(416) 477-8804* for more information.

ATi® TECHNOLOGIES INC.

Technology you can Trust.

450 Esna Park Dr. Markham, Ontario Canada L3R 1H5 TLX. 06-966640

Inquiry 318

Listing 1: *Source code for the* PRTSC *program. The include file,* PSP.INC, *is in listing* 2.

```
;=======INTERRUPT VECTOR STRUCTURE
VECTOR  STRUC
REGIP   DW      ?
REGCS   DW      ?
VECTOR  ENDS
;=======KEYBOARD SHIFT FLAG RECORD
RIGHT   EQU     00000001B
;=======CODE SEGMENT
CODE    SEGMENT
;*******EXECUTION STARTS WITH THIS PIECE
        ASSUME  CS:CODE,DS:CODE,ES:CODE,SS:CODE
; PROGRAM SEGMENT PREFIX
        INCLUDE PSP.INC
; LABEL FOR END STATEMENT
IP      LABEL   NEAR
        JMP     SHORT START
;*******INTERRUPT INTERCEPT PIECE
        ASSUME  CS:CODE,DS:NOTHING,ES:NOTHING
        ASSUME  SS:NOTHING
ORG05   VECTOR  <>              ;ORIGINAL INT
                                ; 005H VECTOR
;-------INTERCEPT ROUTINE
INT05   PROC    FAR
        PUSH    AX
        MOV     AH,002H         ;GET KEYBOARD
                                ; FLAGS FROM BIOS
        INT     016H
; TEST FOR RIGHT SHIFT KEY
        TEST    AL,RIGHT
        POP     AX
        JNZ     INT05X
; NOT DOWN, DO ORIGINAL INT 005H
        JMP     ORG05
INT05X  LABEL   NEAR
        IRET
INT05   ENDP
;*******INITIALIZATION PIECE
        ASSUME  CS:CODE,DS:CODE,ES:CODE,SS:CODE
START   LABEL   NEAR
;-------GET INT 5 VECTOR AND SAVE IN ORG05
        PUSH    ES
        MOV     AH,035H
        MOV     AL,005H
        INT     021H
        ASSUME  ES:NOTHING
        MOV     ORG05.REGIP,BX
        MOV     ORG05.REGCS,ES
        POP     ES
        ASSUME  ES:CODE
;-------SET INT 5 VECTOR TO INT05
        MOV     AH,025H
        MOV     AL,005H
        MOV     DX,OFFSET INT05
        INT     021H
;-------FREE MEMORY ALLOCATED TO THE ENVIRONMENT
        PUSH    ES
        MOV     AH,049H
        MOV     ES,ENVIRONMENT
        ASSUME  ES:NOTHING
        INT     021H
```

BIOS INT 009H routine as well.

In fact, most interrupt routines are (or should be) interrupt-intercept routines. For example, some word processors will replace the keyboard interrupt and not pass control through to other programs "watching" the keyboard.

So far, I have been careful to differentiate hardware from software, internal from external, and BIOS from DOS interrupts. However, the type of interrupt routine is usually implied in the interrupt number. For the balance of this article, I will not distinguish between interrupt types unless there is an ambiguity.

PRTSC

Listing 1, PRTSC.ASM, is a handy interrupt-intercept routine. [*Editor's note: The listings in this article are available on disk, in print, and on* BIX. *See the insert card following page* 176 *for details. Listings are also available on* BYTEnet. *See page* 4.] It intercepts interrupt 5h (the print-screen interrupt) and passes control through to the original print-screen routine only if you are not holding down the right Shift key. This eliminates accidental print-screen functions caused by careless typists and abetted by the adjacent placement of the Shift and Print Screen keys on the IBM PC and PC XT keyboards.

PRTSC begins by defining VECTOR and RIGHT. VECTOR is a structure for holding an interrupt-vector address. RIGHT is a bit in KB_FLAG that is described in the ROM BIOS data area.

The CODE segment is divided into three sections: the entry code, the intercept-routine code, and the initialization code. Note that the ASSUME statements at the top of each piece properly convey the segment-register values. In particular, the only known segment-register value at the top of the intercept piece is CS= SEG CODE. I will describe the sections of the CODE segment in order of execution.

ENTRY CODE

The entry code starts with an INCLUDE of the file PSP.INC, as shown in listing 2. This file represents the layout of the program segment prefix (PSP). PRTSC uses the value of ENVIRONMENT (it needs this to free the memory allocated to the environment). IP is a LABEL for the END statement and must be at offset 00100 hexadecimal. There is no need for an ORG 00100H statement, since the PSP occupies 00100 hexadecimal bytes. The only executable code in the entry code is a jump to the START LABEL of the initialization code. Since you jump over the interrupt-routine code, the initialization code will be highest in memory.

```
        POP     ES
        ASSUME  ES:CODE
;-------TERMINATE PROTECTING MEMORY BELOW START
        MOV     DX,OFFSET START
        INT     027H
CODE    ENDS
        END     IP        ;NOTE IP FOR EXE2BIN
```

Listing 2: *Layout of the program segment prefix.*

```
;=======PROGRAM SEGMENT PREFIX
                    INT     020H          ;INT 020H
TOP_OF_MEMORY       DW      ?
                    DB      ?             ;IOCTL
                    DB      5 DUP(?)      ;DOS_JMP
TERMINATE           VECTOR  <>
CTRL_BREAK          VECTOR  <>
CRITICAL_ERROR      VECTOR  <>
                    DW      ?             ;USED BY DOS
                    DB      20 DUP(?)     ;USED BY DOS
ENVIRONMENT         DW      ?
                    DW      7 DUP(?)      ;USED BY DOS
                    DB      20 DUP(?)     ;UNUSED ?
DOS_CALL            PROC    FAR
                    INT     021H          ;INT 021H
                    RET                   ;RET
DOS_CALL            ENDP
                    DB      9 DUP(?)      ;UNUSED ?
FORMATTED_AREA_1    DB      16 DUP(?)
FORMATTED_AREA_2    DB      16 DUP(?)
                    DB      4 DUP(?)      ;UNUSED
UNFORMATTED_LENGTH  DB      ?
UNFORMATTED_AREA    DB      127 DUP(?)
```

INITIALIZATION CODE

This section uses DOS functions 35h (GET) and 25h (SET) to retrieve and modify the interrupt 5h vector. The original interrupt 5h vector is saved for use by the intercept routine.

Note the careful use of ASSUME statements. Good assembly language programming requires strict adherence to this rule: Any statement changing the value of a segment register must be immediately followed by the appropriate ASSUME statement.

The next piece of code frees the memory that is allocated to the environment. This memory is usually freed when the program terminates; however, INT 027H (the terminate-and-stay-resident interrupt) does not free this memory. Finally, the termination interrupt is executed, returning to DOS. The memory before the LABEL START is reserved so that subsequent programs do not overlay the interrupt routine. The IP on the END statement is required for EXE2BIN to work properly. As noted earlier, IP must have a value of 00100 hexadecimal.

INTERCEPT-ROUTINE CODE

Control passes to the code at location INT05 whenever an INT 005H is executed. (Interrupt 5 is a BIOS interrupt that is often replaced by executing the DOS program GRAPHICS.) INT05 checks the keyboard-shift flags using a BIOS interrupt. If the right Shift key is held down, the routine simply exits with an IRET. If the right Shift key is not held down, control is passed to the previous interrupt 5h routine.

ASSEMBLING PRTSC

To assemble PRTSC and produce a usable program file, execute the following commands:

```
MASM PRTSC;
LINK PRTSC;
EXE2BIN PRTSC PRTSC.COM
```

Do not be alarmed by any message from the linker concerning the stack segment, since .COM files cannot have a stack segment. To use PRTSC, type PRTSC on a line by itself in your AUTOEXEC.BAT file.

NESTED INTERCEPT ROUTINES

PRTSC does not detect whether or not you have already installed the intercept routine. The installation of additional PRTSC routines causes no problems, except that each installation uses memory. There is no standard for the identification of intercept routines. Therefore, detection of reinstallation is not a trivial matter. Although it is probably too late to take hold, I propose the following interrupt-routine standard:

```
NAME      DB      '   '
NEXTRTN   VECTOR  <>
ROUTINE   PROC    FAR
          ...
ROUTINE   ENDP
```

The ROUTINE is preceded by the VECTOR NEXTRTN, where VECTOR defines a structure consisting of a segment/offset double word. NEXTRTN

(continued)

Listing 3: *Source code for the* CLK *program.*

```
;=======VECTOR STRUCTURE
VECTOR   STRUC
REGIP    DW      ?
REGCS    DW      ?
VECTOR   ENDS
;=======DATA SEGMENT
DATA     SEGMENT PUBLIC 'DATA'
CLKDIV   DW      ?         ;<- USER PROVIDED DIVISOR
         PUBLIC  CLKDIV
                           ; ACTUAL DIVISOR ...
CLKDIVH DW      ?         ;...(HIGH WORD)
CLKDIVL DW      ?         ;...(LOW WORD)
CLKMOD   DW      ?         ; DIVISOR MODULUS
; USER-PROVIDED NEAR PROCEDURE
CLKRTN   DW      ?
         PUBLIC  CLKRTN
INT08    VECTOR  <>        ;INTERCEPTED INT 008H VECTOR
DATA     ENDS
;=======STACK SEGMENT
STACK    SEGMENT STACK 'STACK'
STACK    ENDS
;=======CODE SEGMENT
CODE     SEGMENT PUBLIC 'CODE'
         ASSUME  CS:CODE,DS:DATA,ES:DATA,SS:STACK
         EXTRN   SEGCODE:WORD     ;<- CODE SEGMENT
         EXTRN   SEGDATA:WORD     ;<- DATA SEGMENT
;*******CLKRATE CLOCK RATE SETTING ROUTINE
CLKRATE PROC     NEAR
;-------LOAD COUNTER 0 OF THE 8259
         PUSH    AX
         MOV     AL,00110110B
         OUT     043H,AL
         POP     AX
         OUT     040H,AL
         XCHG    AH,AL
         OUT     040H,AL
         XCHG    AH,AL
         RET
CLKRATE ENDP
;*******CLKPRO  CLOCK PROLOGUE
CLKPRO   PROC    NEAR
         PUBLIC  CLKPRO
         PUSH    AX
         PUSH    DX
;-------INITIALIZE CLK VARIABLES
         XOR     AX,AX
         CALL    CLKRATE
         MOV     CLKDIV,AX
         MOV     CLKDIVH,1
         MOV     CLKDIVL,AX
         MOV     CLKMOD,AX
         MOV     CLKRTN,OFFSET CLKNUL
;-------SAVE CURRENT INTERRUPT VECTOR
         PUSH    ES
         MOV     AH,035H
         MOV     AL,008H
         INT     021H
         ASSUME  ES:NOTHING
         MOV     INT08.REGIP,BX
         MOV     INT08.REGCS,ES
         POP     ES
         ASSUME  ES:DATA
;-------INSTALL INTERRUPT INTERCEPT VECTOR
```

(continued)

points to the next routine in the intercept-routine chain. If there is no next routine, NEXTRTN is VECTOR <–1,–1>. NEXTRTN is preceded by an eight-character name.

If this standard were in place, the installation code could start at the original interrupt-vector location, follow the interrupt-routine chain, and determine whether or not to install the intercept. The linked list could also be manipulated to modify the priorities of routines vying for a common resource.

CLK

Listing 3, CLK.ASM, is a module containing code to assist in managing the Intel 8253 Programmable Interval Timer. This chip has three counters. Counter 0 generates interrupt 8h, counter 1 triggers dynamic memory refreshes, and counter 2 drives the cassette port or speaker. They count at a rate determined by the system crystal frequency of 14.31818 megahertz.

Designers of the PC chose this value because, when divided by 4, it produces a 3.58-MHz rate for television color burst. When divided by 3, it produces a 4.77-MHz rate for the Intel 8088. When divided by 12, it produces a 1.19-MHz rate for the Intel 8253. The counter chip has 16-bit counters that count down to 0 (a 0 value is equivalent to 65,536 decimal). The 1.19-MHz rate divided by 65,536 produces the 18.2-Hz signal that triggers the interrupt 8h handler.

Interrupt 8h

Interrupt 8h has three responsibilities. First, it updates a count that represents the system time. Second, it decrements the count for the floppy disk head load. This controls when the head is retracted and the LED light goes off; the head is kept against the disk until this count reaches 0. The count is reset on every disk access, thus reducing disk access time if a new read or write request arrives while the head is still loaded.

Third, interrupt 8h does an INT 1CH (timer tick), letting you provide a

(continued)

YES! I want to subscribe to BYTE and save *50% off** the cover price.

U.S. ☐ $21.00 (1 yr.) *Basic subscription rate ☐ $38.00 (2 yrs.) ☐ $55.00 (3 yrs.)

☐ Canada/Mexico $23.00 U.S. (1 yr.) ☐ Europe-Air Delivery $69.00 U.S. (1 yr.) ☐ Worldwide-Surface Delivery $37.00 U.S. (1 yr.)

Enclose payment (check or credit card information) and get one extra issue for each year you subscribe (North America only).

☐ Bill me (North America only)
☐ Payment enclosed (U.S. funds drawn on a U.S. bank only)
☐ Charge to my ☐ VISA ☐ MasterCard

46STSX

Card # ________ Expires ________ Signature ________

Name ________

Address ________

City/State ________ Country ________ Code ________

Please allow 6–8 weeks for processing your subscription

YES! I want to subscribe to BYTE and save *50% off** the cover price.

U.S. ☐ $21.00 (1 yr.) *Basic subscription rate ☐ $38.00 (2 yrs.) ☐ $55.00 (3 yrs.)

☐ Canada/Mexico $23.00 U.S. (1 yr.) ☐ Europe-Air Delivery $69.00 U.S. (1 yr.) ☐ Worldwide-Surface Delivery $37.00 U.S. (1 yr.)

Enclose payment (check or credit card information) and get one extra issue for each year you subscribe (North America only).

☐ Bill me (North America only)
☐ Payment enclosed (U.S. funds drawn on a U.S. bank only)
☐ Charge to my ☐ VISA ☐ MasterCard

46STSX

Card # ________ Expires ________ Signature ________

Name ________

Address ________

City/State ________ Country ________ Code ________

Please allow 6–8 weeks for processing your subscription

The CLK module lets you run counter 0 at almost any rate.

routine to be executed 18.2 times per second. To run counter 0 at a rate other than 65,536, you need to write an intercept routine that calls the original interrupt routine at the rate of 18.2 times per second. The CLK module lets you run counter 0 at almost any rate.

CLK WALK-THROUGH

The CLK module starts with my ubiquitous VECTOR structure. The DATA segment contains several values. CLKDIV is the interrupt rate that the user desires. CLKDIVH and CLKDIVL are the high and low portions of the current divisor. CLKMOD is the number of 1.19-MHz clock ticks since the last true interrupt 8h. CLKRTN is the user-provided NEAR procedure, to be called on every interrupt 8h. INT08 is the original interrupt 8h vector.

I have defined a STACK segment, even though the SS register is not used, so that the ASSUME SS:STACK statement can be coded. Defining all known segments is a good programming practice.

The CODE segment begins with the proper ASSUMEs and with two EXTRN (external) definitions. In particular, the interrupt routine needs SEGDATA to locate the DATA segment.

CLKRATE is a utility routine that sets the counter 0 divisor value.

CLKPRO is the CLK-module prologue routine. It calls CLKRATE to ensure that counter 0's divisor is set to 65,536. The DATA segment values are initialized, and the current interrupt 8h VECTOR is saved in INT08. Finally, CLKPRO installs CLKINT as the interrupt 8h intercept routine.

CLKEPI is the CLK-module epilogue

(continued)

```
        PUSH    DS
        MOV     AH,025H
        MOV     AL,008H
        MOV     DX,OFFSET CLKINT
        MOV     DS,SEGCODE
        ASSUME  DS:CODE
        INT     021H
        POP     DS
        ASSUME  DS:DATA
        POP     DX
        POP     AX
        RET
CLKPRO  ENDP
;*******CLKEPI  CLOCK EPILOGUE
CLKEPI  PROC    NEAR
        PUBLIC  CLKEPI
        PUSH    AX
        PUSH    DX
;-------RESET CLOCK DIVISOR TO 65536
        XOR     AX,AX
        CALL    CLKRATE
;-------RESET INTERRUPT VECTOR
        PUSH    DS
        MOV     AH,025H
        MOV     AL,008H
        LDS     DX,INT08
        ASSUME  DS:NOTHING
        INT     021H
        POP     DS
        ASSUME  DS:DATA
        POP     DX
        POP     AX
        RET
CLKEPI  ENDP
;*******INTERRUPT ROUTINES
        ASSUME  CS:CODE,DS:NOTHING,ES:NOTHING
        ASSUME  SS:NOTHING
;*******CLKINT  CLOCK INTERRUPT INTERCEPT ROUTINE
CLKINT  PROC    FAR
        PUSH    AX
        PUSH    DS
        PUSH    ES
;-------ESTABLISH ADDRESSABILITY
        MOV     DS,SEGDATA
        ASSUME  DS:DATA
        MOV     ES,SEGDATA
        ASSUME  ES:DATA
;-------DO USER ROUTINE.
        CALL    CLKRTN
;-------CHECK FOR ROLL-OVER OF 65536 CYCLES
        MOV     AX,CLKDIVL
        ADD     CLKMOD,AX
        MOV     AX,CLKDIVH
        ADC     AX,0
        JNZ     CLKINT8
;-------NOT TIME YET, SKIP ORIGINAL INTERRUPT
        MOV     AL,00100000B
        OUT     020H,AL
        JMP     CLKINT7
;-------DO THE ORIGINAL INTERRUPT.
CLKINT8 LABEL   NEAR
        PUSHF
        CALL    INT08
CLKINT7 LABEL   NEAR
;-------CHANGE DIVISOR IF SO REQUESTED
        MOV     AX,CLKDIV
```

(continued)

```
         CMP     AX,CLKDIVL
         JE      CLKINTX
         CALL    CLKRATE
         MOV     CLKDIVL,AX
         CMP     AX,1
         MOV     CLKDIVH,0
         ADC     CLKDIVH,0
CLKINTX  LABEL   NEAR
         POP     ES
         ASSUME  ES:NOTHING
         POP     DS
         ASSUME  DS:NOTHING
         POP     AX
         IRET
CLKINT   ENDP
;*******CLKNUL   INITIAL USER PROVIDED ROUTINE
CLKNUL   PROC    NEAR
         RET
CLKNUL   ENDP
CODE     ENDS
         END
```

Listing 4: *Source code for the* DEMO *program.*

```
;=======VECTOR STRUCTURE
VECTOR   STRUC
REGIP    DW      ?
REGCS    DW      ?
VECTOR   ENDS
;=======PSP SEGMENT
PSP      SEGMENT AT 0FFFFH
; PROGRAM SEGMENT PREFIX
         INCLUDE PSP.INC
PSP      ENDS
;=======DATA SEGMENT
DATA     SEGMENT PUBLIC 'DATA'
         EXTRN   CLKDIV:WORD
         EXTRN   CLKRTN:WORD
COL      DB      0          ;COLUMN
DIR      DB      1          ;DIRECTION
LEFT     DB      ?          ;NUMBER LEFT
COLS     DB      ?          ;LAST COLUMN (0 RELATIVE)
SPEED    DW      1000000000000000B      ;SPEED MASK
DEEPS    DW      0000000000000001B      ;BACKWARDS!
DATA     ENDS
;=======STACK SEGMENT
STACK    SEGMENT STACK 'STACK'
         DW      128 DUP(?)
STACK    ENDS
;=======CODE SEGMENT
CODE     SEGMENT PUBLIC 'CODE'
         ASSUME  CS:CODE,DS:PSP,ES:PSP,SS:STACK
         EXTRN   CLKEPI:NEAR
         EXTRN   CLKPRO:NEAR
DOSCALL  VECTOR  <>         ;WILL POINT TO
                            ;  PSP:DOS_CALL

SEGCODE  DW      CODE       ;CODE SEGMENT
         PUBLIC  SEGCODE
SEGDATA  DW      DATA       ;DATA SEGMENT
         PUBLIC  SEGDATA
IP       LABEL   NEAR       ;EXECUTION STARTS HERE
```

(continued)

routine. It also calls CLKRATE to ensure a divisor of 65,536. Then, CLKEPI resets the interrupt 8h interrupt vector to its original value.

CLKINT

CLKINT is the CLK-module interrupt-intercept routine where the work is done. Note again that the program initially assumes nothing except CS= CODE. CLKINT establishes the location of the DATA segment and the extra segment by loading DS and ES with SEGDATA. It then calls the user-provided NEAR procedure, whose address is in CLKRTN. If CLKRTN does not finish before the next timer interrupt, an interrupt might be lost because the hardware keeps track of an interrupt's occurrence only—not how many times it has occurred. The current counter's divisor represents the number of 1.19-MHz ticks that have passed since the last interrupt. This value is added to CLKMOD.

A carry indicates a rollover past 65,536. If there is no rollover, the 8259 is notified that the interrupt has been processed. If there is a rollover, the intercepted interrupt 8h is called. In order to call an interrupt routine, you must precede that call with a PUSHF instruction. Finally, the routine checks to see if the user has requested a different divisor since the previous interrupt and, if so, sets the new divisor.

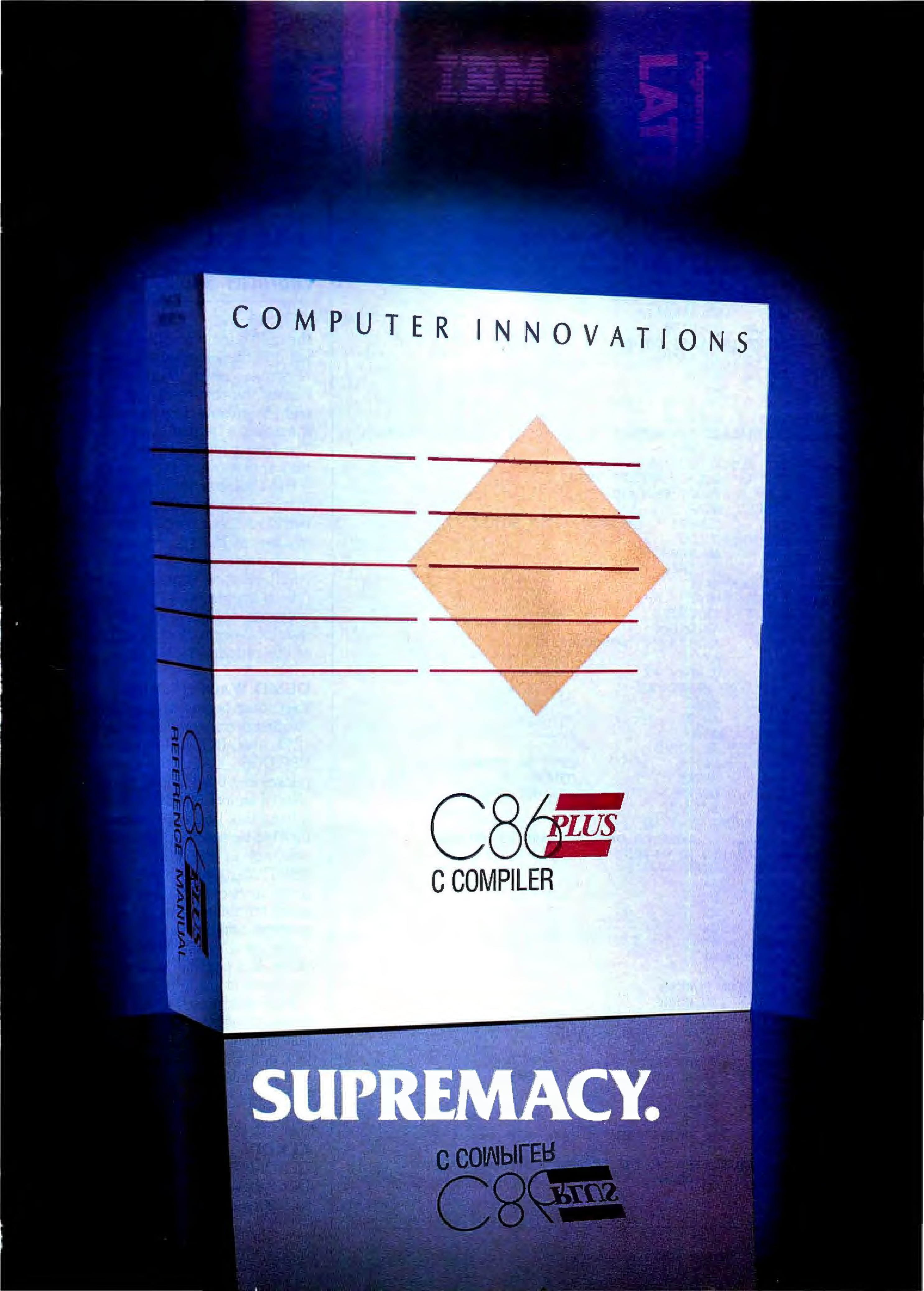
CLKNUL is a null interrupt routine used as the initial value of CLKRTN.

I have used the CLK module for interrupt-driven music at varying tempos and to invoke C procedures as background tasks. Unfortunately, a full-blown example of its use is beyond the scope of this article. However, the DEMO program described below hints at some of its potential uses.

DEMO

Listing 4, DEMO.ASM, is a program that graphically verifies the operation of the CLK module. It writes characters to the top line of the screen until you press the Escape key. The characters are displayed left to right until

(continued)


COMPUTER INNOVATIONS
C86 PLUS
C COMPILER
C86 PLUS REFERENCE MANUAL
SUPREMACY.

```
;-------SETUP DOSCALL VECTOR
        MOV     DOSCALL.REGIP,OFFSET DOS_CALL
        MOV     DOSCALL.REGCS,DS
;-------SETUP DS AND ES REGISTERS
        MOV     ES,SEGDATA
        ASSUME  ES:DATA
        MOV     DS,SEGDATA
        ASSUME  DS:DATA
;-------DETERMINE LAST COLUMN
        MOV     AH,00FH
        INT     010H
        DEC     AH
        MOV     COLS,AH
        MOV     LEFT,AH
;-------INITIALIZE CLK MODULE
        CALL    CLKPRO
;-------SET CLOCK ROUTINE
        MOV     CLKRTN,OFFSET INT08
        JMP     SHORT SETRATE
MAINX   LABEL   NEAR
        CALL    CLKEPI   ;DEINSTALL CLK MODULE
;-------RETURN TO DOS
        MOV     AH,000H
        CALL    DOSCALL
;-------SET CLK RATE
SETRATE LABEL   NEAR
        MOV     AX,SPEED
        OR      AX,DEEPS
        AND     AX,1111111100000000B
        SHL     AX,1
        MOV     CLKDIV,AX
;-------WAIT FOR KEYSTROKE
KEY     LABEL   NEAR
        MOV     AH,0
        INT     016H
        CMP     AL,01BH
        JE      MAINX              ;EXIT ON ESCAPE
        ROR     SPEED,1            ;OTHERWISE,
        ROL     DEEPS,1            ; CHANGE SPEED
        JMP     SETRATE
;*******INTERRUPT ROUTINE
        ASSUME  CS:CODE,DS:DATA,ES:DATA,SS:NOTHING
; NEAR PROC CALLED BY INT08
INT08   PROC    NEAR
        PUSH    AX
        PUSH    BX
        PUSH    CX
        PUSH    DX
        PUSH    BP
        PUSH    SI
        PUSH    DI
;-------POSITION CURSOR
        MOV     AH,002H
        MOV     BH,0
        MOV     DH,0
        MOV     DL,COL
        INT     010H
;-------DERIVE CHARACTER FROM TIME OF DAY
        MOV     AH,0
        INT     01AH
        MOV     AL,DL
        AND     AL,00111111B
        ADD     AL,' '
;-------PUT CHARACTER ON THE SCREEN
        MOV     AH,00AH
```

(continued)

When the clock rate is 36.4 Hz, each character shows twice.

the right edge is reached and right to left until the left edge is reached. Any keystroke other than an Escape causes the rate to change. The first eight keystrokes double the rate until it reaches a frequency of 2329.6 Hz. The next eight keystrokes halve the rate until it returns to 18.2 Hz.

The character to display is chosen from the lower-order bits of the system clock. Since the system clock runs at a rate of 18.2 Hz, the initial output is of consecutive characters in the ASCII collating sequence. When the rate is increased to 36.4 Hz, each character shows twice. This verifies that the system clock is still running at the proper 18.2-Hz rate.

DEMO WALK-THROUGH

Execution begins at the LABEL IP. The first two statements set up a VECTOR called DOSCALL that points to PSP:DOS_CALL. This documented but seldom-used field in the PSP consists of an INT 021H and a FAR RET. Calling this VECTOR causes an INT 021H to be executed with the CS register set to the PSP. Since calling PSP:DOS_CALL meets all the requirements of DOS functions 0h (program terminate) and 31h (terminate process and remain resident), you never need to use INT 020H or INT 027H. It is possible to make an .EXE file resident using this technique.

At initialization, DEMO sets up the segment registers and determines the last display column. It then calls the CLK prologue routine and informs the CLK module that INT08 is the routine to be called at the rate determined by CLKDIV.

CLKDIV

CLKDIV is derived from SPEED and DEEPS using some unnecessary but fun bit-twiddling. SPEED contains a

(continued)

Fast ANSI Console Driver


```
          MOV     BH,0
          MOV     CX,1
          INT     010H
;--------UPDATE COLUMN
          MOV     AL,DIR
          ADD     COL,AL
          DEC     LEFT
          JNZ     SHOWX
          NEG     DIR
          MOV     AL,COLS
          MOV     LEFT,AL
SHOWX     LABEL   NEAR
          POP     DI
          POP     SI
          POP     BP
          POP     DX
          POP     CX
          POP     BX
          POP     AX
          RET
INT08     ENDP
CODE      ENDS
          END     IP
```

single bit and is rotated right on each keystroke. DEEPS is the reverse of SPEED; it contains a single bit (which is the mirror image of SPEED) that is rotated left on each keystroke. When SPEED and DEEPS are combined via an OR operation, a single bit is in each byte. To set CLKDIV, the program isolates the upper byte and doubles its value. This produces divisors from 10000 down to 02000 hexadecimal in powers of two.

Main Loop

The main loop processes keystrokes. If the keystroke is not an Escape, it cycles CLKDIV as described above. If the keystroke is an Escape, it calls the CLK prologue routine and the program is terminated.

INT08

When INT08 is called, it first pushes all registers. BP, SI, and DI are pushed because INT 010H does not preserve these registers. Next, the routine positions the cursor and derives the character to be displayed from the time of day. It sends this character to the CRT and updates the column number. Finally, INT08 restores all the registers and returns control to the CLK module.

I have used only BIOS interrupts in this program because it is difficult to use DOS interrupts within a hardware-interrupt routine. This is because DOS is not reentrant.

Assembling DEMO

The steps in assembling DEMO are

```
MASM   DEMO;
LINK   DEMO+CLK;
```

To execute the DEMO program, simply enter DEMO.

Conclusion

In this article, I have covered the major aspects of writing interrupt routines for the IBM PC family. You should be able to successfully tackle writing your own interrupt routines by building on this foundation. ■

BIBLIOGRAPHY

Disk Operating System Technical Reference. Boca Raton, FL: IBM Corporation, 1985.

iAPX 86,88 User's Manual. Santa Clara, CA: Intel Corporation, 1981.

MCS-85 User's Manual. Santa Clara, CA: Intel Corporation, 1977.

Technical Reference Personal Computer AT. Boca Raton, FL: IBM Corporation, 1984.

HOW FAST CAN THE 80386 MAKE YOUR WORKSTATION GO?

Faster than any other 32-bit microprocessor can.

In technical terms that's 4 MIPS. 4 MegaWhetstones. With 4 Gigabytes.

Fast enough to have a sudden and lethal impact on your competition.

To achieve its blazing speed, the 80386 has six-stage pipelining, which allows the standard 16 MHz part to deliver over 6100 Dhrystones per second.

That's nearly four times the Dhrystone performance of a VAX* 11/780, or 4 VAX MIPS.

And with the 80386 you don't sacrifice all that performance for demand-paged virtual memory. Because of its on-chip paging unit and transparent paging cache, the 80386 can do a full virtual to physical address translation without the performance penalty of non-pipelined, off-chip designs. Giving you 4 MIPS of deliverable UNIX** performance.

Of course, to achieve performance like that, you need a bus that's faster than a speeding bullet.

Like our 2 clock bus that provides a 32 Megabytes per second transfer rate.

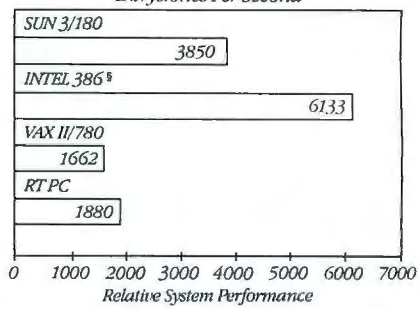
That's the highest performance of any microprocessor bus.

So what more could advanced system designers ask for?

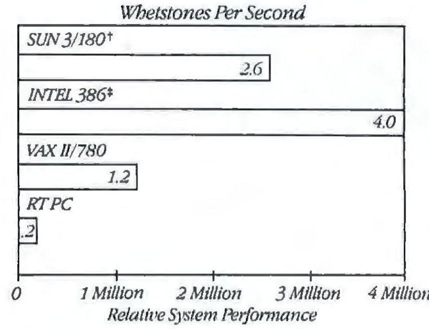
Blistering 4 MegaWhetstones per second performance for superb graphics and number crunching. Only the 80386 uniquely teams with Weitek's 1167 floating point coprocessor to deliver the highest numerics performance on the market.

And of course all this comes in a 64 Terabyte virtual address space with a 4 Gigabyte linear address space.

Dhrystones Per Second



Whetstones Per Second



§ Intel 386 MULTIBUS* I 16 MHz Starter Kit.
† With Weitek 1164, 1165 floating point chip set.
‡ With Weitek 1167 floating point coprocessor.
Source: Refer to Intel 80386 Benchmark Report.

Finally, for your silver bullet, add to all this $6 billion worth of DOS software. And watch how fast it runs on your 4 MIPS, 4 MegaWhetstones UNIX machine.

To find out more, attend one of our 80386 Seminars. Call your local Intel sales office for details.

Or call (800) 548-4725 for complete technical information.

Get all the ammunition you need to make your next system a success.

intel®

*VAX is a registered trademark of Digital Equipment Corp.
**UNIX is a trademark of AT&T Bell Laboratories.

NEC'S PINWRITER P5XL HAS MADE BLACK A PRIMARY COLOR.

Our Pinwriter® P5XL printer has changed forever the way people look at dot matrix printing.

It's the first 24-pin dot matrix printer to use a letter-quality multistrike film ribbon—the same ribbon used in typewriters and letter-quality printers, such as our Spinwriter.® So for the first time in computer history there is a printer that honestly does everything. A printer that produces important letters and documents with crisp, black, true letter-quality printing. But with all the speed and graphics capability dot matrix printers are known for.

```
Dear Mr. Black:
```

Actual line printed with a Pinwriter P5XL printer.

Fast, black letter-quality printing will be the primary reason many people will buy a P5XL printer. But there are plenty of other good reasons. In fact, it's the most versatile printer ever created for personal computers.

It can use an optional ribbon to print seven other colors plus black. And it has the best graphics resolution of any impact printer you can buy, due in part to our advanced 24-pin printhead. Plus it can print more type faces automatically than any other dot matrix printer. And it's quiet and fast.

You can also expect a P5XL printer to turn out millions of characters before it will need service because it has the highest reliability rating in the industry. And there's a nationwide network of NEC Customer Service Centers to take care of maintenance.

Now, while the Pinwriter P5XL performs a little black magic, you won't have to go in the red to buy it.

The Pinwriter P5XL is the latest addition to the most advanced and extensive family of 24-pin printers available. See it at your dealer or for an information package that includes actual print samples, call 1-800-343-4418 (in MA 617-264-8635). Or write: NEC Information Systems, Dept. 1610, 1414 Massachusetts Ave., Boxborough, MA 01719.

NEC PRINTERS. THEY ONLY STOP WHEN YOU WANT THEM TO.

C&C Computers and Communications

The chevrons and serpentine designs are registered trademarks of Binney and Smith Inc., used with permission.

NEC

NEC Information Systems, Inc.

Inquiry 207

C VERSUS ASSEMBLY—C PLUS ASSEMBLY

C provides portability and assembly functionality without the pitfalls of assembly code

BY TOM HOGAN

IS C A BETTER choice than assembly language for professional programming tasks? C has a reputation for being portable, easy to write, very fast, and difficult to maintain. Assembly has a reputation for being compact, difficult to write, lightning fast, and impossible to maintain.

While the reputations of C and assembly language have some factual basis, it is easy to find examples that don't fit them. Some C programs are definitely not portable, and some C applications run faster than others written in assembly. Assembly can be written to be readable, as can C (see "Easy C" by Pete Orlin and John Heath, May BYTE). This discussion should clarify matters.

My bias is in favor of C because applications can be written more quickly with C than with assembly. My experience is in developing a C graphics library for the IBM PC. Since the hardware is barely adequate for real-time low-resolution graphics applications, I had to write some assembly code to get acceptable performance. In doing this, I became familiar with the C-to-assembly language interface.

I will compare C with assembly by discussing their relative speed, the suitability of one versus the other for complex applications, real-world products that were written in each language, the comparative popularity of products written in each, which applications absolutely require assembly language, and techniques to help you write assembly functions to work with your C programs.

C's portability is one of its supposed advantages over other languages. Assembly language is not portable, although cross-assemblers can aid in moving code to new architectures. High-level languages like FORTRAN and Pascal have built-in I/O capabilities that limit their portability. C lacks any I/O functions, but every C compiler I am aware of comes with a library that supplies these.

A bigger problem with high-level languages is that most compilers are supersets of the language. If a program relies heavily on a compiler's nonstandard features, it is not going to port easily to another compiler. C avoids this by not including much in the language. If a programmer needs a special function, he writes it himself or buys a commercial library package.

One dilemma C programmers often face is whether to sacrifice speed or portability. If the application must run independently of the hardware (under UNIX, for example), portability is more important. If the application is written for a single architecture, speed is the priority.

C programs are not necessarily portable, but you can write them to be portable, and you must take great care to ensure this. Generally, you reduce a program's portability if you introduce assembly language to improve its speed. The resources and overhead of MS-DOS, for example, are too limiting for highly interactive commercial applications to rely on them (or on the standard library that relies on them), so assembly language is used for video output instead of MS-DOS routines. For example, writing text through DOS is much too slow when you have to update an entire screen for a full-screen editor.

(continued)

Tom Hogan (C Source Inc., 12801 Frost Rd., Kansas City, MO 64138) *is a senior programmer using assembly and various C compilers. His interests are war and fantasy games and computer-game design.*

ILLUSTRATED BY KAREN STOLPER

C VERSUS ASSEMBLY

Comparing C programs written for speed with assembly programs produces mixed results.

If you compare well-behaved C programs (such as UNIX applications) with programs written in assembly, you might conclude that assembly is much faster than C. However, comparing C programs written for speed with assembly programs produces mixed results.

An article written by Mark Edwards and titled "Programming Editors, Programmable Editors" in the November 1985 issue of Dr. *Dobb's Journal* rated the speed of different editors for various applications. The editors written in C included BRIEF, EC, EMACS, and Epsilon. [*Editor's note*: BRIEF *and* Epsilon *are written primarily in* C *with assembly routines to optimize performance.* EC *and* EMACS *are written entirely in* C.] Those written in assembly were Pmate, VEDIT PLUS, and XyWrite II Plus. Editors written in C had by far the fastest times in the following categories: search and replace (EC), change ASM comments to C (EMACS), and brace count macro (BRIEF). In file I/O, XyWrite came in first, followed closely by Epsilon, with the rest of the editors quite far behind. Pmate had the fastest load time, although EC was a not-too-distant second.

In the macro benchmark, editors written in assembly were at least an order of magnitude slower than the editors written in C that took first and second place. Screen handling had a three-way tie for first place with a C editor and two assembly editors.

The upshot of all this is that the language in which a program is written matters less than its implementation where speed is concerned. The

(continued)

Amdek challenges you to read between the lines.

All 350 of 'em.

Not all monitors are created equal. And no monitor in this price range can equal the new Amdek Color 722.

What makes the 722 RGB monitor so distinctive? For one, a dual frequency output that is capable of supporting IBM's Enhanced Graphics Adaptor. The result is 350 lines of resolution, assuring you of a sharper, crisper image that makes your graphs and charts look more like a work of art, and less like a rough sketch. For another, the fact that the 722 is made by Amdek —where monitors are our only line, not just a sideline.

The high performance 722 features an etched glass, non-glare screen that's easy on the eyes, plus front-mounted controls that are easy on the operator. This means less eye fatigue and greater productivity.

What's more, the 3-position text switch enables you to choose green, amber or full-color type — up to 80 characters in width x 25 lines of text. There's even an optional tilt/swivel stand that allows you to select a viewing angle that's most comfortable for you.

It all adds up to more monitor for the money, and is backed by more warranty for your peace of mind. Amdek's warranty protects your investment for three full years on the CRT, and two years on all other parts and labor. No other monitor warranty offers you more.

For high resolution performance, see the new Amdek Color 722. We know you'll be impressed. After all, everytime an Amdek monitor is sold, we know that we are putting our image on the line.

AMDEK

Inquiry 13

Clearly the finest in monitors.

IBM is a registered trademark of International Business Machine Corp.

2201 Lively Boulevard, Elk Grove Village, IL 60007. Phone: 312/364-1180. TLX: 280-803 editors written in C were generally faster than those written in assembly. There were two reasons for the performance advantage: The editors written in C could be written faster initially, so the author could spend more time improving performance; and a complex design can be implemented in C much more easily than it can in assembly.

You might argue that C programs cannot be fast because they have too much high-level language overhead compared with assembly. Who cares whether or not C has overhead? The question is, "Can I make my C program run like greased lightning?" The answer is, "Yes, if you know what you are doing." The whole question hinges on program design and how well you can write code—not on how efficient your compiler is (although more efficient code is without question desirable).

Let's look at how C programs can be made about as fast as assembly programs. Assume that both the C program and the assembly program use the same algorithm, the algorithm causes the C program to spend 95 percent of its time in 5 percent of the code, and the C program is unacceptably slow. An experienced C programmer can rewrite the time-intensive 5 percent of the C code in assembly, often achieving a several-fold improvement in performance. Thus, the final program will run about as fast as if it had been written entirely in assembly.

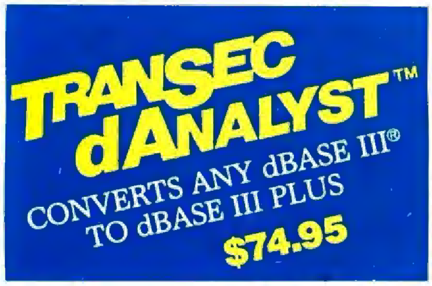
In the world of commercial applications, speed is often crucial to success. Part of the reason for the success of Turbo Pascal (written in assembly) is that it is more fun to use than competitive products. Today, people complain that WordStar 2000 (written in C) is slower than the original WordStar (cross-assembled from 8080 assembly), although WordStar 2000 does a lot more.

Assembly advocates tout early successes like dBASE II and WordStar to show that the most popular programs are written in assembly. I submit that assembly was used because it was the

(continued)

TRANSEC NEW PRODUCTS

Program Review:



Makes dBASE Easier to Program, Reduces Logic Errors, Bugs

TranSec dANALYST™ automatically converts any dBASE III program to dBASE III PLUS with true record and file locking for multi-user operation. Features include syntax/blocking analysis, tabbed file output, variable cross reference. Framework-type interface command windows, pop-up menus, graphic analysis REPORTS menu. Saves programmer's time to analyze and debug dBASE II, III, or PLUS. Supports multi-user Clipper and network version of WordTech. Not copy protected. Rated a "best buy."

TranSec dANALYST™ $74.95 plus $5 s/h
TO ORDER: 1-800-423-0772 VISA/MC
IN FL: 1-305-474-7548

Program Review:

Simplify Naming, Finding Your Files with 32-Character File Names

TranSec PCEasy™ is a new easyto-use "office environment" file management system that allows anyone to easily organize large amounts of data on a hard disk. Expands DOS file name limit to 32 characters to name files naturally, the way you think. Use easily remembered plain words or numbers. The interface structure mimics the paper world, only it's faster, better and easier to use . . . by programmers, managers, secretaries, clerks. PCEasy immediately finds file cabinets, file drawers, file folders, spread sheets, reports in their own logical location. Automatic menu access to application programs. Works with all popular applications and utilities. Not copy protected. Rated "★★★★ for ease of use."

TranSec PCEasy™ $74.95 plus $5 s/h
TO ORDER: 1-800-423-0772 VISA/MC
IN FL: 1-305-474-7548

UNLOCK™ Removes Copy Protection

Produces Unprotected Backup Copies

RUNS YOUR SOFTWARE ON ANY HARD DISK

UNlock "copying" disk allows you to make "unprotected" DOS copies of popular original program disks. **Unprotected backup copies** perform perfectly, as do copies of these copies. UNlock copies **run on any hard disk,** including Bernoulli Boxes. No original required in drive "A." **Run on a RAM disk** simply and conveniently. **Also, copy DOS 5¼" programs to 3½" diskettes.** For IBM® PC, XT, AT, compatibles, 256K or more, DOS 2.1 or higher. Choice of the critics!

Guaranteed to work only with programs below:

UNLOCK™ ALBUM "A" PLUS — $49.95
(Plus $4 ship/handling. Foreign orders $10)

- dBASE III & dBASE III PLUS™
- FRAMEWORK I & II™
- CHARTMASTER™ (6.1)
- SIGNMASTER™ (5.1)
- FASTBACK™ (5.3)
- HARVARD TOTAL PROJECT MGR.™ (1.10)
- THINKTANK™ (2.0, 2.1)
- LOTUS 1-2-3™ (1.A-2.0)
- IBM WRITING ASSISTANT™ (1.01)
- IBM FILING ASSISTANT™ (1.0)
- IBM REPORTING ASSISTANT™ (1.0)
- GRAPHWRITER™ (4.3, 4.31)
- REALIA COBOL™ (1.2, 2.0)
- MULTILINK ADVANCED™ (3.02, 3.03)
- DOLLARS & SENSE™ (2.0)

UNLOCK™ ALBUM "B" PLUS — $49.95
(Plus $4 ship/handling. Foreign orders $10)

- SYMPHONY™ (1.1)
- CLIPPER™
- LOTUS 1-2-3 REPORT WRITER™ (1.0)
- DOUBLEDOS™
- smARTWORK™ (1.0 Rev. 8 thru 10 & 1.1)
- DISK OPTIMIZER™
- MANAGING YOUR MONEY™ (1.5, 1.51, 2.0)
- MICROSOFT WORD™ (1.15, 2.0, 2.01)
- PFS: ACCESS™ (1984 Ed.)
- PFS: PLAN™ (B)
- PFS: GRAPH™ (B)
- PFS: WRITE™ (1.01, C)
- PFS: REPORT™ (B, C)
- PFS: FILE™ (B, C)
- DATABASE MANAGER II—THE INTEGRATOR™ (2.0, 2.02)

SPECIAL OFFER — "Best of UNLOCK" ALBUM "D" PLUS — $74.95
(Plus $4 ship/handling. Foreign orders $10)

- dBASE III & dBASE III PLUS™
- FRAMEWORK I & II™
- CLIPPER™
- FASTBACK™ (5.3)
- CHARTMASTER™ (6.1)
- SIGNMASTER™ (5. 1)
- DOLLARS & SENSE™ (2.0)
- LOTUS 1-2-3™ (1.A-2.0)
- SYMPHONY™ (1.1)
- LOTUS 1-2-3 REPORT WRITER™ (1.0)
- DOUBLEDOS™
- HARVARD TOTAL PROJECT MGR.™ (1.10)
- MANAGING YOUR MONEY™ (1.5, 1.51, 2.0)
- THINKTANK™ (2.0, 2.1)
- MICROSOFT WORD™ (1.15, 2.0, 2.01)

UNLOCK™ FLIGHT/JET #201 — $14.95
- FLIGHT SIMULATOR™ • JET™

(Plus $4 ship/handling. Foreign orders $10)

CHOICE OF THE CRITICS!

PERSONAL COMPUTING **"UNlock has two particularly endearing characteristics: it works, and works simply. I was able to quickly produce unprotected copies of Lotus 1-2-3 release 2, Symphony 1.1, Microsoft Word 2.0, dBase III 1.1, and Framework II. These copies performed flawlessly, as did copies of these copies."** Christopher O'Malley, PERSONAL COMPUTING, April '86

PC MAGAZINE **"Because copy protection can interfere with the ability to back up a hard disk, business-oriented users may prefer programs like TranSec's UNlock series."** Winn L. Rosch, **PC MAGAZINE, May 27, 1986**

BYTE **"UNlock 4.7 defeats the latest Prolok and SuperLock type of copy protection scheme. It's menu-driven and works fine on the programs it's supposed to work on: Lotus 1-2-3, dBase III, Framework, Symphony, Paradox, and several others."** Jerry Pournelle, **BYTE, Feb. '86**

Recommended by the editors of: PERSONAL COMPUTING "The Best Software Utilities For Under $100"

TRANSEC UNLOCK™

ORDER TODAY TOLL FREE:
1-800-423-0772
IN FLORIDA: 1-305-474-7548

TranSec Systems, Inc., 1802 North University Drive, Plantation, FL 33322

Trademarks are the sole property of their respective owners. UNlock is for use only to improve the useability of legally acquired and operated software.

One overlooked factor is the inverse relation between speed and the time required to write a program.

best tool available at the time. It offered very good speed in the finished applications and produced compact code (important on 8080 systems where memory was limited). Also, users were willing to put up with buggy software because there wasn't a lot of software available.

Now that excellent implementations of C are available, many programmers are choosing them over assembly. Another important reason for using assembly has vanished—compact size. Personal computers tend to have at least 512K bytes of memory, so a program's compactness is less important than it was on 8080 systems. Compactness will become less important as systems have increasing amounts of memory.

Assembly is less attractive as a programming language because programs written in it tend to have more bugs than those written in other languages. Fast programs don't help you get a job done more quickly if they are buggy. Programs written in C help you get the job done faster because the program is more robust (that is, less likely to crash), resulting in less downtime. So a program's useful speed can be as important as its execution speed.

One overlooked factor is the inverse relation between speed and the time necessary to write a program. This is hard to quantify, because it assumes that if you have more time to work on a program you will find a faster algorithm. It is this factor that lets some C programs run faster than their as-

(continued)

Post Office
Will Not
Deliver Without
First Class
Stamp

McGraw-Hill Continuing Education Center
3939 Wisconsin Avenue
Washington, D.C. 20016-2876

Make Any Computer Do Exactly What You Want With McGraw-Hill's

Contemporary Programming & Software Design Series

From Writing Your Own Programs to Modifying Existing Software, Here's the New, Easy, and Low Cost Way to Unlock the Secrets of Your Computer

Whether you use computers for business, for personal applications, or for fun, off-the-shelf programs will never do everything you want them to do for you. That's because they were written by programmers to satisfy what they perceived as the needs of the greatest number of potential users—often missing some or many of your specific needs.

That's why McGraw-Hill's new Contemporary Programming and Software Design Series teaches you how to create your own software . . . either from scratch or by making key modifications to existing programs.

There is nothing magical about it. You learn the process of building a computer program step-by-step with McGraw-Hill *Concept Modules* sent to you one at a time, once a month. Each of the ten modules in the Series takes you through an important step in the development of the structure and detailed logic of a program, including testing, debugging, and documentation.

Unique Interactive Hands-On Instruction

Each module includes an easy-to-understand guide PLUS a 5¼" floppy disk containing typical programs and interactive instruction that you can run on IBM PCs, PC compatibles and Commodore 64 and 128 computers for hands-on experience.

In the first Module, for example, when your sample program (Declining Interest Loans) appears on your screen, you'll find errors on certain program lines. You'll also see that the program is only three-quarters completed.

Now comes the fun part. You'll discover how this program is built, and in the process you'll learn how to identify and correct errors. And by the end of Module 1, you'll actually have completed this program yourself.

But there's more. Special graphics on your screen work in conjunction with the accompanying guide to amplify, illustrate, and deepen your understanding of software design principles.

The Crucial 95%—Learn the Foundation of Computer Programming

While the Series includes interactive disks that run on specific computers, everything you learn you can apply to any language or machine. Why is this possible? Because McGraw-Hill knows programming is far more than coding a program into the computer using a specific language. In the real world of computers, 95% of the programming process is carried out using design techniques that are independent of specific language or machine. It is this crucial 95% that you thoroughly understand and master in the Series.

Make no mistake. Almost all books and courses on "programming" teach you only the final 5% of the total programming process—namely, how to code in a specific language . . . information of little value if you don't know how to reach the point in the programming process when you are ready to code.

With the Series, however, you'll learn to create your own programs from scratch, even modify off-the-shelf programs. You'll learn enough BASIC and machine language to get you started on the remaining 5% of the programming process.

Build Your Own Personal Software Library

The sample programs you work with throughout the Series are excellent learning tools. But they're more than that. By combining the sample programs onto one master disk, you'll have the start of your own personal software library. In addition to the programs you've written and modified throughout the Series, you'll also receive dozens of the most popular public domain and user-supported programs, such as data base manager, word processor, calendar generator, appointments reminder and much, much more.

15-Day No-Risk Trial

To order your first module without risk, send the postage-paid card today. Examine the first module for 15 days and see how the Series will help you make your computer do exactly what you want it to do!

If someone has beaten you to the card, write to us for ordering information about the Contemporary Programming and Software Design Series.

McGraw-Hill
Continuing Education Center
3939 Wisconsin Avenue
Washington, DC 20016

sembly language counterparts. Even programmers who favor assembly language admit that assembly programs take longer to write than C programs.

A general rule of thumb is that it takes about the same amount of time to write a line of source code no matter which language you use. (Scaling factors enter in if you want precise results, but the general rule is adequate for my argument.) Because it typically takes me three lines of assembly code to generate the equivalent of one line of C, I will estimate that it takes about three times as long to write a program in assembly as it does in C.

The cost to a company that uses assembly as its primary programming language is twofold. Assuming salaries for C and assembly programmers are comparable, the company spends three times as much on programmers' salaries as it would if its programmers wrote in C. And competitors who use C will beat it to market and get plenty of time to establish their products.

As far as maintenance goes, programs written in assembly are more difficult (translate "expensive") to maintain for commercial developers. Naturally, the extra cost is passed along to the consumer.

Some people assert that assembly has a more complete set of instructions than C. I disagree—assembly has a much *finer* set of instructions than C. Assembly gives the programmer much tighter control of program flow. However, it interfaces so nicely with C that you might say that the C instruction set includes assembly's instruction set. As far as addressing memory goes, C provides about everything you need. (The only really necessary data types that C lacks are an 8-byte BCD data type for accounting packages and an 8-byte long for high-resolution graphics.)

C has much more powerful software tools than assembly because of all the support products for C programmers. There are more than 15 professional C compilers, plus many source-level debuggers, general C libraries, C

(continued)

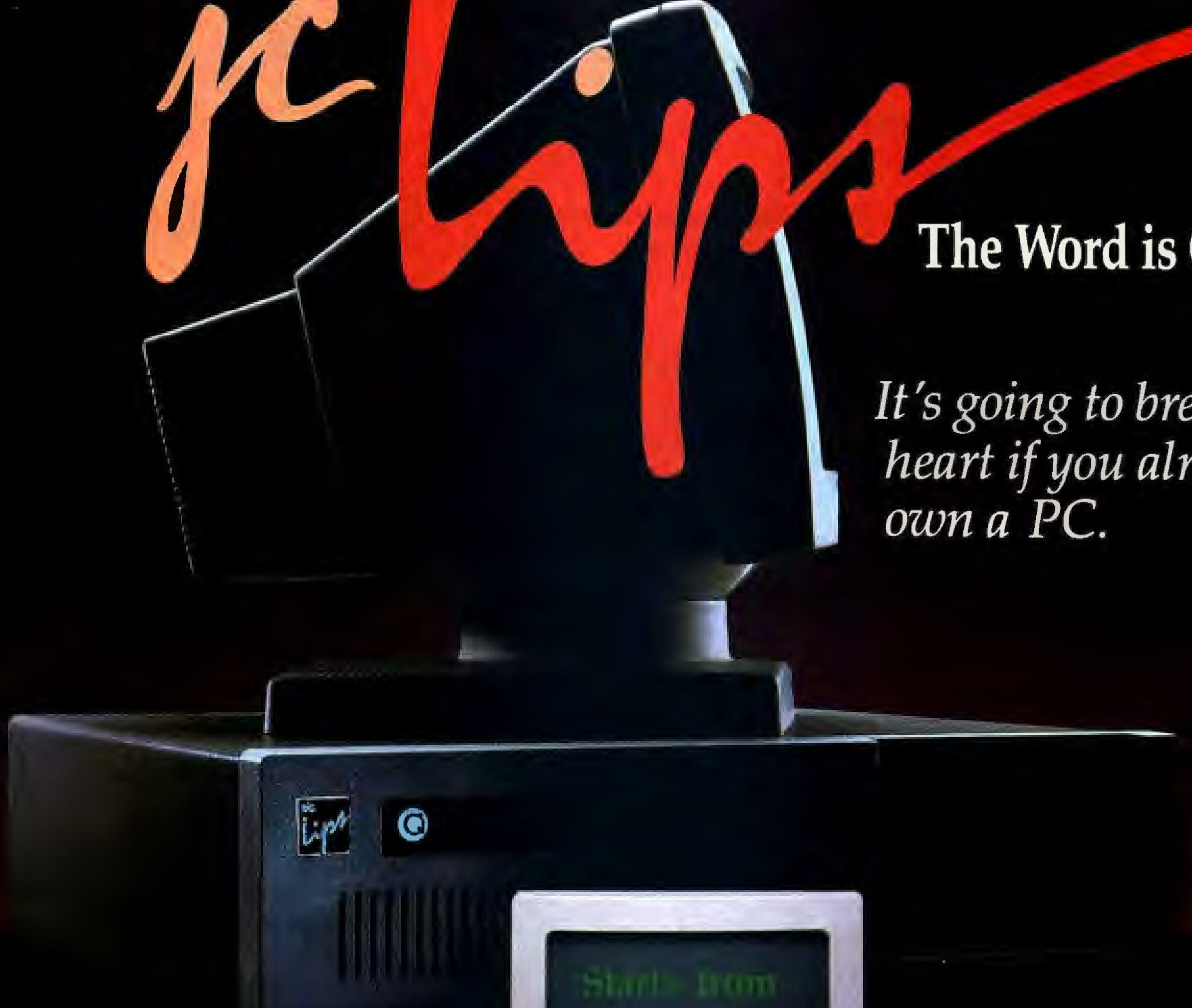
 Inquiry 258

Inquiry 148 for End-Users.
Inquiry 149 for DEALERS ONLY. →

jc Lips

The Word is Out...

It's going to break your heart if you already own a PC.



JC LIPS provides a new experience in personal computing. Our third generation personal computer brings you technological advancement and features which have not been possible in the past. It runs all popular IBM PC software 2 to 4 times faster than other compatibles. It will impress you with unsurpassed performance and cool reliability. Best of all, it is available to you at an affordable price.

JC LIPS features the advanced-technology CMOS NEC V40 processor running at 8 MHz with an equivalent throughput of at least 9.5 MHz. The systemboard has 256K of RAM, expandable to 640K, a built-in floppy disk controller, a socket for a high speed floating point processor, and 8 I/O expansion slots. Also included as standard on systems are 360K floppy disk(s), a serial interface, a parallel printer interface, a battery backed-up clock, a keyboard, and a 135 Watt power supply, capable of handling all of this plus any accessories you may add later.

JC LIPS' display system includes a quality 14" high resolution monochrome monitor, with a dark background non-glare screen and a tilt and swivel base for easy viewing. The multi-function display card provides an IBM PC compatible interface for a monochrome display with Hercules compatible graphics or a RGB color graphics display.

JC LIPS is also available with high speed 20 MB hard disk with 2K cache buffer memory controller, Enhanced Color Graphics display adapter, high resolution RGB color monitor, local area networking controller, and more. These options can be added at any time, and with the features we supply as standard, it will be a difficult system to outgrow.

NOW *the word is out*. Another superb product is available from the high-performance, multi-user systems company, JC Information Systems.

JC INFORMATION SYSTEMS
161 Whitney Place
Fremont, CA 94539
(415) 659-8440 TWX 910-381-7041

C's rich programming environment is probably the most important reason to choose it as a development language.

libraries that emulate commercial database managers, interpreters, lint checkers, editors that offer special C support, translators that convert source code from other languages to C, and teaching tools to help novice C programmers learn C. The rich programming environment for C is probably the most important reason to choose C as a development language.

Assembly just doesn't have the support. Support products include Microsoft's Macro Assembler, other miscellaneous assemblers, cross-assemblers, and a few symbolic debuggers. Its support is sparse when compared with C's.

One assembly writer was surprised to learn that a C programmer went to the trouble to write his own I/O routines. He thought that a C compiler was supposed to do that. His error is instructive in the philosophy behind C: The compiler is deliberately kept sparse so that you can tailor I/O routines to your needs. Many C programmers write their own routines eventually. The standard library just gets you up and running under MS-DOS.

Having covered the disadvantages of assembly, I want to point out two instances where it has distinct advantages over C. Where memory is extremely limited, C might not be able to produce code that will fit. Assembly's compactness may be the deciding criterion. Another case is when very high-performance software is needed and the program doesn't spend most of its time in a small sec-

(continued)

Perfect timing. Because the Maxi-Switch ME-101 is the next-generation AT, PC/XT* compatible keyboard. And you can order it right now. Not next week, not tomorrow. Today.

Three custom layouts. Order the standard 101-key when you must have an exact IBM layout. Select the 102-key European version when you need a keyboard tailored for international markets. To make any operator's data entry life easier, choose the super enhanced model with its oversize L-shaped "Enter" key for quick, accurate striking.

All three switch selectable layouts offer a smaller footprint for extra desktop space. All three give you the option of relocating the "Caps Lock" and "Ctrl" keys to their traditional AT positions. All three feature the patented Maxi-Key rollover for fast, error-free data entry.

The Maxi-Touch is here. And best of all — all three models offer you the Maxi-Touch,™ a clear, consistent tactile response with true breakover keyswitch action. Made possible by the conductive elastomer technology Maxi-Switch introduced to the keyboard world in 1981, proven since then in hundreds of thousands of keyboards to measurably increase typing throughput.

No time to lose. Call or write today for detailed information about our entire line of IBM plug compatible keyboards. If you have a custom requirement, call us or your local Maxi-Switch representative about making your keyboard your way. At Maxi-Switch, we profile our products for tomorrow's technology.

Get in touch with the **Maxi-Switch Company,** 9697 East River Road, Minneapolis, Minnesota 55433, phone (612) 755-7660, TWX 910-576-2690. In Europe, contact **EECO Ltd.,** Bar Hill, Cambridge, CB3 8SQ, England phone 0954-80257, TLX 817303.

EECO Incorporated | Maxi-SWITCH

Experience the Maxi-Touch

*IBM AT and XT are registered trademarks of IBM Corporation.

Inquiry 184 for End-Users. Inquiry 185 for DEALERS ONLY.

tion of the code. In this case it might be easier to write the whole application in assembly.

SUPPLEMENTING C WITH ASSEMBLY

Although I have presented C and assembly in an opposition format, I would like to show how assembly can complement C. Of the C programs written for the PC architecture, most are written in the small-data model. Accordingly, if a program needs to access areas outside of the program's data space, such as video RAM, it needs to use assembly routines.

C doesn't provide access to some of the 8088's hardware resources. If a program needs to access a system interrupt or a port, you must use assembly. Generally, the C compiler library includes assembly functions that do these things. Sometimes these resources are inadequate, requiring you to write assembly code.

If a program's performance is too slow, you will have to write certain portions of the code in assembly. Assembly provides access to the fast 8088 block write/search commands (REPZ, REPNZ with MOVS, SCAS, CMPS, or STOS) that C lacks. Again, the compiler library typically provides C functions using fast block writes/searches, but they might not do what the programmer needs.

HOW ASSEMBLY INTERFACES WITH C

Let's look at what C programmers have to take into consideration when writing assembly routines. The C programmer who is starting to write assembly code for use by C programs will run into several obstacles, one of which deals with segments. C compilers take care of all the linkage information so you don't have to worry about code and data segments. Assembly isn't so nice: You have to make sure the assembly code includes instructions that match the output of the C compiler to tell the linker where code and data are to go.

Some C compilers add an underscore before or after public names. When you write public functions or reference static data in assembly code, be sure to add a leading or trailing underscore (as required by your compiler) to its name. Should you forget the underscore, you will probably get an error message from the linker about unresolved external names.

C compilers assume that the contents of certain registers will be saved. In the small-data model, most compilers require that all segment registers be saved. In the large-data model, the ES register doesn't need to be saved, but the contents of all other segment registers need to be saved. If the compiler implements register

(continued)

The Diconix 150. So quiet, it's the one PC printer you can take lightly. Anywhere.

DICONIX
A Kodak Company

3100 Research Boulevard
Dayton, Ohio 45420.
1-800-DICONIX Telex: 288-280

Inquiry 85

The Common Dilemma

Apparent Solution.

512KB,6Mhz.
$4545*

Intelligent Conclusion.

1MB, 12.5Mhz.
$2549*

The Common Dilemma for you, the microcomputer buyer is deciding which system to purchase to meet the needs of your users, your customers, or yourself.

The Apparent Solution seems to be that you *could* buy IBM®. What you'd get however, is a naked machine with inadequate performance at an astronomical price.

Or, you *could* be enticed by those super-cheap and questionable clones at the expense of sacrificing full compatibility, quality, reliability, and user satisfaction.

The Intelligent Conclusion *is*, of course, to move up to **Proteus™. Proteus™** solves the dilemma with three high quality *American-Designed, Engineered* and *American-Built* powerful microcomputers. **Proteus™ Microcomputers,** which come from a leading electronics corporation, surpass all microcomputers on the market, in technology, speed, reliability, price and after-sale support.

Our basic philosophy:

- ☐ use high quality and stable 8-layer boards.
- ☐ use only genuine components.
- ☐ we are always testing software & hardware products to ensure total compatibility—we spend the time making it right so you don't have to.

And **Proteus™** comes with Operations & Technical Guides that are meant to be used by people, not machines. Of course, we also offer our **free** 24-hour, 7-day on line system for technical support, service inquiries, and just plain questions about **Proteus™**. Fully compatible with UNIX,' ZENIX and Autocad.

The Proteus-XTi™ is our basic business system. As with all **Proteus™** systems, it includes **nation-wide on-site support by a major U.S. maintenance group. Free upgrades** of BIOS, ROM, and systems utility software for all registered users.

- Intel 8088-2CPU, 4.77/8 Mhz
- 640K RAM Expandable to 1MB on Motherboard
- O Memory Wait State
- 360K 5¼" Floppy Drive
- 2 Serial Ports, 2 Parallel Ports
- Clock/Calendar, SASI Interface on Board
- Reset Button
- 135 Watt UL Power Supply
- Hercules Compatible Mono/Graphics Adapter
- High Resolution Monochrome Monitor

Price: $979.

The Proteus-286™, Our Standard Configuration

- Intel 80286-10 CPU, 6/10 Mhz
- 1 Wait State, 0 Wait State Optional
- 1.2 MB 5¼" Floppy Disk Drive
- 360K 5¼" **or** 3.5" Microfloppy
- Combination Hard/Floppy Disk Controller
- 1 MB RAM Expandable to 4MB on Motherboard
- 5 Mhz DMA Bus
- 2 Parallel Ports, 3 Serial Ports on Motherboard Selectable RS232 **or** RS422
- Clock/Calendar with Battery on Board
- A.T. Keyboard
- Reset Button
- 200 Watt UL Power Supply — 110v/220v Switching

Price: $2195.

The Proteus-286GT™, Our Standard Configuration

Same as 286 but with

- Intel 80286-12 CPU, 6/12.5 Mhz
- One Memory Wait State

Price: $2549.

proteus
The Intelligent Conclusion

To Order, Call:

WINTECH DATA PRODUCTS CORP.® U.S.A.
377 ROUTE 17
HASBROUCK HEIGHTS, N.J. 07604

PHONE: 201-288-8629
TELEX: 510 601 0960
FAX: 201-288-8577

Inquiry 307

Special Configurations with Hard Disks, EGA, CGA, Monitors available.
Terms: Visa, M.C., C.O.D., Prepayment, Wire Transfer or L.C.

*(The Fine Print). Suggested Retail Prices. IBM is a registered trademark of International Business Machines. Proteus is a trademark of Wintech Data Products Corp. All prices, terms, specs, and offers, subject to change.

variables, the SI and DI registers might need to be saved. BP needs to be pushed and popped if you use it to address the stack (which you do if you need to pass arguments to the function).

Assembly routines address these arguments as offsets of the base pointer. The difficulty with addressing arguments as numeric offsets is that it is easy to become confused as to just what [BP + 18] is pointing at. The offset is different for the small- and large-code models, because in the large-code model the segment number—as well as the offset—must be in the return sequence. The small-code model requires only the offset. In the large-data model, pointers are 4-byte quantities. To load pointers passed as arguments on the stack into a register requires an LES or LDS instruction. In the small-data model, a simple MOV will suffice.

Sometimes your assembly function will need to return a value. Generally, integers are returned in the AX register. The way pointers are returned is variable, depending on the compiler and whether you are using the small- or large-data model. Longs may be returned in AX and BX or in AX and DX (which is standard DOS convention). Chars are returned in AL. Doubles are returned in AX, BX, CX, and DX. One compiler returns near pointers in a different register than integers—a mistake in my opinion, because many C programmers don't explicitly declare functions to return pointers.

USING A MACRO ASSEMBLER

You can make writing assembly for C a good deal easier when you use a macro assembler. You should combine standard patterns into macros. For example, the prologue to starting a function called FN_INT in the small-code model is

```
PUBLIC FN_INT
FN_INT PROC NEAR
```

As an example, you could write a macro called FN_PROLOGUE (this macro won't work with version 1.0 of

(continued)

MASM because of a bug):

```
FN_PROLOGUE MACRO NAME
PUBLIC NAME
NAME PROC NEAR
ENDM
```

Then, if you wanted to write FN_INT, you could start with

```
FN_PROLOGUE FN_INT
```

Most compilers put the first argument passed to a function at [BP + 4] in the small-code model (remember, the return address and BP register—two words—are commonly pushed onto the stack upon entering a function). Let's define a constant, ARG_BASE, which points at the first argument passed to the assembly function:

```
ARG_BASE EQU BP + 4
```

To make sure you address the stack properly, you can create other constants (assuming the small-data model):

```
INTEGER = 2
POINTER = 2
LONG = 4
LONG_PTR = 4
FLOAT = 4
DOUBLE = 8
CHAR = 2 ; although chars are
         ; 1-byte values, com-
         ; pilers promote
         ; them to integers
         ; before pushing them
         ; on the stack
```

With these defined, you can address the first element on the stack with the expression [ARG_BASE]. If the first argument is a pointer, you can address the second argument with the expression [ARG_BASE + POINTER], and so on.

For those of you who use a compiler that adds an underscore before or after public names, an operator in the Macro Assembler that you will find invaluable is &. For example, if you want to append an underscore to a function called FN_INT, you would say FN_INT&_. This can be handy when you are creating macros. You might want to modify my first example using this trick.

WRITING FAST-EXECUTING ASSEMBLY CODE

If you've decided to use assembly routines to improve the performance of your C program, here are some hints to make these parts of your code as fast as possible.

Keep your variables in registers. The difference between MOV REG, REG and MOV REG, MEMORY is theoretically a factor of eight. In practice, the execution of many small, fast instructions limits the speed because time is spent moving the instructions into the prefetch queue.

You normally have six registers to work with—AX, BX, CX, DX, SI, and DI. Any variables that you can guarantee won't exceed the value of 255 can be kept in AL, AH, BL, BH, and so on. This means you could have up to 10 variables in registers. For most functions, this is more than adequate.

Write large code. This might sound like

(continued)

Listing 1: A *faster way to multiply a byte variable by the value* 80.

```
MOV  AX, VARIABLE    ; 3 bytes, 14 clocks
MOV  CL, 4           ; 2 bytes,  4 clocks
SHL  AX, CL          ; 2 bytes, 24 clocks
MOV  BX, AX          ; 2 bytes,  2 clocks
SHL  AX, 1           ; 2 bytes,  2 clocks
SHL  AX, 1           ; 2 bytes,  6 clocks -- prefetch queue
                     ; is empty before the instruction
                     ; is executed
ADD  AX, BX          ; 2 bytes, 14 clocks -- 4 clocks/byte
                     ; for the BIU to put the instruction
                     ; in the queue plus 2 clocks to
                     ; execute it
```

Listing 2: A *routine to move a row of text into the screen buffer.*

```
      MOV  CX, 40            ; 40 WORDs to search
      MOV  DI, BUFFER        ; DI points at the buffer
      MOV  SI, SCREEN_OFFSET ; SI points at the screen offset
START:
      MOV  AX, [DI]          ; 14 clocks -- get two chars from the buffer
      AND  AX, 07F7FH        ;  4 clocks -- turn off the high bits
      MOV  [SI], AX          ; 14 clocks -- put the chars into screen RAM
      ADD  SI, 2             ;  4 clocks -- the next two chars in the buffer
      ADD  DI, 2             ;  4 clocks -- the next two chars in screen RAM
      LOOP START             ; 17 clocks -- continue until CX is zero
```

Introducing . . .

ATI 6/12 MHz System 286

A Quick Step Forward...

IT'S FAST! The ATI System 286-12 runs at an amazing speed of 12MHz. *That's 100% faster than the original IBM Personal Computer AT.®*

IT'S COMPATIBLE! Even at this amazing speed, it is compatible with most name-brand multifunction, video and peripheral boards. And to protect your existing investment, we've built-in a normal speed mode as well. A few simple keystrokes will switch you to 6 MHz operation mode, when necessary.

IT'S RELIABLE! This ATI system 286-12 is evolved from proven design of ATI-2000 and ATI-1000 mother boards, of which over 10,000 pieces have been shipped in the field. And to back this reputation, on-line technical support is available to you from knowledgeable people who speak your language.

IT'S VERSATILE! Combine the ATI System 286-12 with our MegaGraph Plus EGA board to get superb graphics performance. Or expand your system's main memory with the ATI Multifunction card and benefit from Xenix® applications. Up to five disk drive slots and eight expansion slots give you everything you need to grow from single-user environments to the next generation multi-user systems.

Call or write today for more information on our full line of AT® compatible system boards, graphics peripherals, and system accessories. And take a quick step forward!

ATI welcomes our current customers to inquire regarding our special offers.

Specifications:

- IBM PC-AT compatible
- 12 MHz 80286 microprocessor
- Dual speed, 6/12 MHz
- Keyboard switchable
- Norton Sys Info rating of 11.5-13.3
- 1 MByte on board memory
- One 1.2 MByte Floppy Drive
- 200 watt power supply
- AT style keyboard
- BIOS standard

ATI

ATronics International Inc.

We Deliver Advanced Technology

491 Valley Way, Milpitas CA 95035

(408) 943-6629 TLX: 510-600-6093

Inquiry 27

Listing 3: *A routine to search for a byte using the* compare string block *instruction.*

```
        MOV    CX, 128          ; SEARCH 128 BYTES
        MOV    AL, VALUE        ; VALUE TO BE FOUND
        MOV    DI, OFFSET       ; START OF MEMORY TO SEARCH
REPNZ   SCASB                   ; 15 clocks/repetition,
                                ; plus 2 clocks per
                                ; execution of prefix.
```

Listing 4: *The same search routine without using block instructions.*

```
        MOV    CX, 128          ; SEARCH 128 BYTES
        MOV    AL, VALUE        ; VALUE TO BE FOUND
        MOV    AH, AL           ; SAME VALUE IN AH
        MOV    DI, OFFSET       ; START OF MEMORY TO SEARCH
        DEC    DI
TOP:
        INC    DI               ;  2 clocks
        MOV    AL, AH           ;  2 clocks
        SUB    AL, [DI]         ; 14 clocks
        LOOPNZ TOP              ; 19 clocks
```

heresy, but it's true in many cases. Here's an example in which larger code runs faster than smaller code. Suppose you want to multiply a byte variable by 80 and can guarantee no overflow. One way to do it is

```
MOV AX, VARIABLE ; 3 bytes
                 ; 14 clocks
MOV BL, 80       ; 2 bytes
                 ; 4 clocks
MUL AL, BL       ; 2 bytes
                 ; 70–77 clocks
```

This produces 7 bytes of code that takes 88–95 clocks to execute.

Listing 1 shows a much faster way. The code is 15 bytes long and takes about 66 clocks to execute. It is about twice as much code as the previous example and runs about a third faster. (The comparison assumes that the prefetch queue is full at the beginning of each example.) This finding runs counter to what some experts say—that the larger a program is, the slower it runs.

Michael Abrash had an interesting observation in his article on how using fast shift and rotate instructions can actually slow execution time (see "Bit Rotation Speeds," *PC Tech Journal*, May 1986). Although Intel gives a time of two clocks for the instruction SHL AX, 1, if the instruction isn't already in the prefetch queue, it will take eight clocks to fetch the instruction and two clocks to execute it—five times what the book says. You empty the queue quickly with those fast two or three clock instructions. It might be faster to use the shift or rotate instructions with the CL register.

A corollary of "write large code" is "avoid elegance." Brute-force routines run faster than routines that rely on a lot of decision making. Those conditional jumps aren't cheap. (The idea that more code runs faster also applies to C, although at a higher level. You might have two similar routines that are optimized to be fast at specific tasks instead of one general, slower routine.)

Suppose you want to move a row of text into screen RAM from a buffer and turn off the high bit. You could write the code shown in listing 2. Each iteration of the loop takes 57 clocks. The total time to process 80 characters is 2280 clocks, which doesn't include the overhead from having to refill the prefetch queue after each jump. Look what happens when you use more code:

```
    MOV   CX, 20
    MOV   DI, BUFFER
    MOV   SI, SCREEN__OFFSET
START:
    MOV   AX, [DI]        ; 14 clocks
    AND   AX, 07F7FH      ;  4 clocks
    MOV   [SI], AX        ; 14 clocks
    MOV   AX, [DI + 2]    ; 14 clocks
    AND   AX, 07F7FH      ;  4 clocks
    MOV   [SI + 2], AX    ; 14 clocks
    ADD   SI, 4           ;  4 clocks
    ADD   DI, 4           ;  4 clocks
    LOOP  START           ; 17 clocks
```

The total time for this loop is 1780 clocks. You can continue the process by moving four words at a time instead of two. For four words, the total time is 1530 clocks. The advantage decreases significantly as more time is spent inside the loop. Figure that the overhead is 23 clocks times the number of iterations.

Take advantage of block instructions. The 8088 instruction set includes instructions that can search a block of memory for a byte or word value, or write a byte or word value into a block of memory. These instructions execute much faster than code that uses instructions that search a byte or word at a time. For example, the code in listing 3 searches for VALUE beginning at OFFSET. Assuming that the code searches the entire 128 bytes without finding VALUE, the code will take about 1950 clocks. The equivalent code that doesn't use block instructions is shown in listing 4. The set of instructions in listing 4 takes over twice as long to execute as the previous set (37 clocks per repetition versus the previous 15).

Count the clocks. Use short instructions if possible. Be aware of how instructions modify flags so you don't use a TST or CMP if the flags already contain the information you need to do a conditional jump. Use one-word instruction instead of two byte instructions if possible. The list of tricks to speed up code goes on and on.

One key idea is: Learn the 8088 instruction set. Another is: Don't be afraid to use unconventional tricks. Often, merely setting up a nonintuitive array of data to drive the decision making will reduce code size and increase the speed. This will work in your C code as well. ■

ONLY PANASONIC® GIVES YOU A BUSINESS PARTNER.™

Everyone seems to be jumping on the bandwagon.

Today, companies that weren't there yesterday, and may not be there tomorrow, are building computers from easily assembled components.

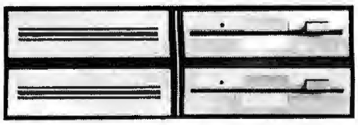
The result is a sea of computer clones with little to distinguish one from another.

THE PANASONIC DIFFERENCE—A POWERFUL BUSINESS ALLY. Our approach to offering you a computer was entirely different. We looked at what would work best for you and your business.

The result is the Panasonic Business Partner and Panasonic Business Partner 286. Two powerful business allies, designed to answer your most demanding computer requirements.

YOU CAN SEE THE DIFFERENCE PANASONIC MAKES. With our long, unparalleled heritage in video technology, you get high-quality, high resolution monitors. An optional Panasonic CGA PLUS board is switchable to let you use a color or monochrome display for both graphics and text.

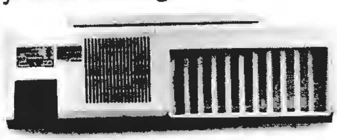
BUILT-IN FLEXIBILITY. No two businesses run exactly the same way. So you can configure the Business Partners to your own individual needs. In addition to their built-in floppy disk drives, you can add a hard disk drive, additional half-height drives—even a magnetic tape drive.



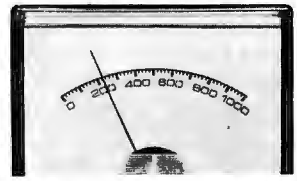
SPEED AND PERFORMANCE. The Business Partner 286 is built with maximum power and speed. Its 80286 processor runs at 6 MHz or 8 MHz, with the ability of addressing up to 15Mb RAM.

The Business Partner, with its 8086-2 processor, runs at 4.77 MHz, and for projects that need added acceleration, it will run up to 7.16 MHz.

FUTURE GROWTH. Both Business Partners have been designed to grow as your business grows. Extra slots are built in to offer you the flexibility—now and in the future—to add on cards for high resolution graphics, communications accessories and other peripherals.

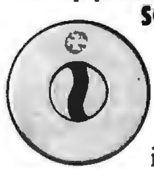


POWER TO SPARE. As your business grows, you'll put greater demands on your Business Partner. Don't worry, it can handle it. With 130 watts on the Business Partner and 220 watts on the Business Partner 286, you'll be able to drive additional peripherals and expansion boards.



IMPORTANT DESIGN CONSIDERATIONS. The deluxe, detachable keyboards on both models have been designed for comfort and ease of use. We've separated the cursor keys and numeric keys for quick, convenient access. Swivel monitors adjust for each operator. On the Business Partner 286, you can lock the cabinet with a key to help prevent tampering and prying.

SOME REASSURING LAST WORDS. You'll find the Business Partner compatible with the IBM* PC/XT. And the Business Partner 286, with the PC/AT. So you'll be able to run the most popular software. You'll also get compatibility with Panasonic's wide range of quality peripherals, for the complete Panasonic office.



And more importantly, both Business Partners are from Panasonic, the company firmly committed to building a high-quality and dependable reputation in computer products. So both are backed with a full-year limited warranty.

For more information, and the location of the dealer nearest you, call **1-800-PIC-8086.** Or write: Panasonic Industrial Company, Computer Products Division, Two Panasonic Way, Secaucus, NJ 07094. And discover how the right Business Partner can make running your business easier. You'll see: **THE EASIER, THE BETTER.**

Panasonic
Office Automation OA

*IBM is a registered trademark of the International Business Machines Corp.

pascal/turbo #244, from billn (Bill Nicholls)
a comment to message 221

My company is using TP for production programming and we have found the screen differences (and others) to cause some compatibility problems. The best solution to the screen compatibility issue is to use the IBM version 3 compiler for PC and 100% compatibles, but use generic TP version 2 for MS-DOS compatibles, as it still supports reverse video, etc., which was removed from generic v.3. This has not been exhaustively tested but works for our code. The v.2 generic works on a PC also.
BillN.

pascal/turbo #268, from jdow (Joanne Dow)
a comment to message 262

At the risk of getting fried for the "C"isms here, this is a very partial listing of the ANSI x3.64 commands. Hope they help.
<^_^>
==========
c.language/tips #48, from jdow

And printf("\0x1b[%dJ", parameter) will:

```
parameter  action
0          erase to end of screen
1          erase from start to current (including
           current)
2          as noted clear screen leaving cursor in
           place
```

And printf("\0x1b[%dK", parameter); is the same with line replacing screen.
printf("\0x1b[%d;%dH); can be: printf("0x1b[H"); for home cursor or printf("\0x1b[%dH", line); works, as does printf("\0x1b[;%dH", column);

```
cursor     sequence
  up       \0x1b[A
  dn       \0x1b[B
  rt       \0x1b[C    (make last two lines "\x1b" into
                      "\0x1b")
  lt       \0x1b[D
```

Note that a parameter can squeeze into the string just after the "[" for a rep count.

index = \0x1bM
reverse index = \0x1bD

And then we get into the attributes that I won't get into because I am not sure what they'd be implementing. I believe the general sequence would be "\0x1b[%d;%d;...%dm" (any sensible number of %d's in there separated by ";" will work). If the parameter is "0" then all attributes are canceled.
These are ANSI x3.64 defns that, hopefully, they followed in ANSI.SYS but I have no PC to try them on. I'm just reading my terminal defns.
<^_^>

THE EOF( ) FUNCTION

pascal/turbo #1036, from mwelch (Mark Welch)

I've been fiddling with a program that will invert a file or set of files (text files) into a binary tree of the words, so I can easily search and locate all instances of a word *within* the file set. (I know *it's* been done before, and yes, I'll post it when it's ready.)

To try to cut down execution time, I've been using

(continued)

ONLY PANASONIC® GIVES YOU A BUSINESS PARTNER.™

Everyone seems to be jumping on the bandwagon.

Today, companies that weren't there yesterday, and may not be there tomorrow, are building computers from easily assembled components.

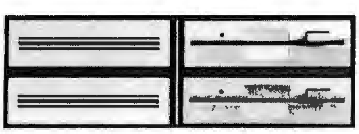
The result is a sea of computer clones with little to distinguish one from another.

THE PANASONIC DIFFERENCE—A POWERFUL BUSINESS ALLY. Our approach to offering you a computer was entirely different. We looked at what would work best for you and your business.

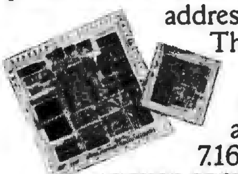
The result is the Panasonic Business Partner and Panasonic Business Partner 286. Two powerful business allies, designed to answer your most demanding computer requirements.

YOU CAN SEE THE DIFFERENCE PANASONIC MAKES. With our long, unparalleled heritage in video technology, you get high-quality, high resolution monitors. An optional Panasonic CGA PLUS board is switchable to let you use a color or monochrome display for both graphics and text.

BUILT-IN FLEXIBILITY. No two businesses run exactly the same way. So you can configure the Business Partners to your own individual needs. In addition to their built-in floppy disk drives, you can add a hard disk drive, additional half-height drives—even a magnetic tape drive.

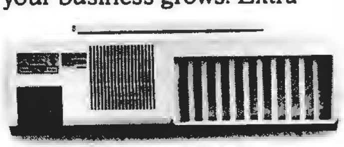


SPEED AND PERFORMANCE. The Business Partner 286 is built with maximum power and speed. Its 80286 processor runs at 6 MHz or 8 MHz, with the ability of addressing up to 15Mb RAM.

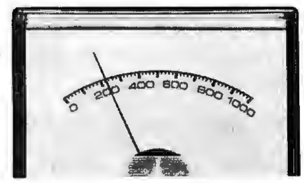


The Business Partner, with its 8086-2 processor, runs at 4.77 MHz, and for projects that need added acceleration, it will run up to 7.16 MHz.

FUTURE GROWTH. Both Business Partners have been designed to grow as your business grows. Extra slots are built in to offer you the flexibility—now and in the future—to add on cards for high resolution graphics, communications accessories and other peripherals.

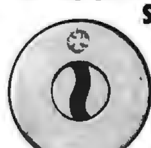


POWER TO SPARE. As your business grows, you'll put greater demands on your Business Partner. Don't worry, it can handle it. With 130 watts on the Business Partner and 220 watts on the Business Partner 286, you'll be able to drive additional peripherals and expansion boards.



IMPORTANT DESIGN CONSIDERATIONS. The deluxe, detachable keyboards on both models have been designed for comfort and ease of use. We've separated the cursor keys and numeric keys for quick, convenient access. Swivel monitors adjust for each operator. On the Business Partner 286, you can lock the cabinet with a key to help prevent tampering and prying.

SOME REASSURING LAST WORDS. You'll find the Business Partner compatible with the IBM* PC/XT. And the Business Partner 286, with the PC/AT. So you'll be able to run the most popular software. You'll also get compatibility with Panasonic's wide range of quality peripherals, for the complete Panasonic office.



And more importantly, both Business Partners are from Panasonic, the company firmly committed to building a high-quality and dependable reputation in computer products. So both are backed with a full-year limited warranty.

For more information, and the location of the dealer nearest you, call **1-800-PIC-8086.** Or write: Panasonic Industrial Company, Computer Products Division, Two Panasonic Way, Secaucus, NJ 07094. And discover how the right Business Partner can make running your business easier. You'll see: **THE EASIER, THE BETTER.**

Panasonic
Office Automation OA

*IBM is a registered trademark of the International Business Machines Corp.

BIX is the BYTE Information Exchange, BYTE's continuous electronic conferencing system. The following excerpts represent conversations taking place on BIX over a number of months and in a number of different conferences. The common ground for all is, of course, the IBM PC. The conferences represented this month are IBM AT, Pascal, Assembler, and BASIC. For information on how to join BIX, see the advertisement on page 320.

IBM PC AND COMPATIBLES

As the most powerful member of the IBM PC family, the PC AT receives a good deal of attention on BIX. In the following excerpts, an address space is explored.

ADDRESS SPACE E000:0000

ibm.at/at.hardware #353, from tanstaafl (Dana Cline)

Does anyone know if the 64K bytes starting at E000:0000 is used for anything? On my AT, Debug reads changing garbage there. I think it is mapped to an empty ROM socket. I need to use it for a video board and don't know if it will work reliably. Any ideas?

ibm.at/at.hardware #354, from cjackson (Craig Jackson)
a comment to message 353

The only uses I've ever heard of for that area are the PCjr ROM cartridge slots. One is at D000 and the other at E000. IBM presumably will evaluate new uses for this space as the PCjr becomes less of an issue.

ibm.at/at.hardware #355, from rschnapp (Russell L. Schnapp)
a comment to message 353

This area can be used for onboard BIOS ROM. I don't think the AT has any dedicated purpose for E000:0.

ibm.at/at.hardware #369, from tanstaafl
a comment to message 353

I've investigated the E000:xxxx address space on my AT and found some interesting things. The results make me wonder about the sanity of Debug. I'm using a video board that looks like RAM to the system. I jumpered it to fit in segment E. When using the Debug fill command, no matter what I fill with, it sets the RAM to FFs. When I display that area, it shows garbage. However, the garbage changes. Not only that, a Debug display does not match the hex to the ASCII. The left side has FEs and 5As, while the characters are "<" and "s" (and periods). 5A is "Z", so why does Debug display it wrong?

As far as E000 reserved for BIOS ROM, there is none there that I can find. Does it map to the empty sockets on the motherboard? A Debug dump of D000 shows different stuff, and D000 works for the video board. According to all the manuals, both D and E segments are reserved, unused. AST Research uses D and E to page in their Enhanced Memory Spec memory. What do they know about this area?

ibm.at/at.hardware #383, from rschnapp
a comment to message 369

It appears that E000:0 does, in fact, map to the spare ROM sockets on the AT motherboard. We have Golden Bow disk map chips in our ATs, and when I looked at E000:0, I found their copyright notice (doubled up, of course).

I haven't checked the tech specs yet, to find out if E000 is still available if you don't plug in ROM chips in those sockets.

PASCAL

The Pascal conference is consistently one of the busiest on the system. The following conversations cover a number of topics including setting screen attributes, errant EOF markers, a data compression technique (this is a simple technique in the best hacker tradition), and replacing console output drivers.

SCREEN ATTRIBUTES FROM PASCAL

pascal/turbo #221, from rlang (Robert Lang)

This question breaks down into two parts. First, since Turbo Pascal doesn't have an Inverse or Blink command similar to LowVideo and NormVideo, I have been playing around with attribute bytes. The way I currently do it is to create an absolute variable:

```
  MonoScreen : array [ 1..25, 1..80, 1..2 ] of char
absolute $B000:0000;
```

And one for Color screen at $B800. Then I write into Row, Col, 2 for the attribute byte, the code I want. It works but violates every rule on writing generic software. Is there a way I can use an interrupt or ms-dos call to set an attribute byte that will work on all MS-DOS machines? One that is as simple to use as assigning to an array the Row, Col and either char or attribute byte? (Perhaps a procedure with arguments?)

Second, Microsoft Word does something on a monochrome screen I didn't think could be done. When a bold letter, word, etc., is highlighted it appears in inverse, but with a bold background. (Bold in this case implies intensity on.) Whenever I set the intensity bit on, it only works for foreground, that is, the character, not the background. I just double-checked this and MS Word does in fact have two different intensity inverse backgrounds. Anyone know how they can set background on intense? Could they be using a different character set where the character is the background and the foreground is the shadow of the character? If I knew how, I would love to get a copy of the screen bytes as in the adapter when I have some bold text highlighted with MS Word. I would really like to be able to get intense inverse in my TP program. Granted it is an aesthetic issue, but if MS Word can do it, then there has to be a way in TP.

Sorry about length of question, but I am way over my head and could use any help anyone might offer.
Rob Lang

pascal/turbo #232, from rlang
a comment to message 221

My objective is to be MS-DOS-specific, not IBM PC-specific. My assumption (!?) is that if I keep to generic TP procedures and functions, then I can run my software on other machines. Inherent in this assumption was the capability of the MS-DOS machine to be able to display on the screen

```
  Normal Text  ( Intense )
  Dim Text     ( LowVideo )
  Inverse
  Blinking
```

and hopefully any one of the permutations (inverse blinking of high intensity text!?).

Perhaps this assumption is incorrect for all MS-DOS machines and I should admit defeat now and stick with PC-specific software?
Rob Lang

(continued)

pascal/turbo #244, from bilin (Bill Nicholls)
a comment to message 221

My company is using TP for production programming and we have found the screen differences (and others) to cause some compatibility problems. The best solution to the screen compatibility issue is to use the IBM version 3 compiler for PC and 100% compatibles, but use generic TP version 2 for MS-DOS compatibles, as it still supports reverse video, etc., which was removed from generic v.3. This has not been exhaustively tested but works for our code. The v.2 generic works on a PC also.
BilIN.

pascal/turbo #268, from jdow (Joanne Dow)
a comment to message 262

At the risk of getting fried for the "C"isms here, this is a very partial listing of the ANSI x3.64 commands. Hope they help.
<^_^>
===========
c.language/tips #48, from jdow

And printf("\0x1b[%dJ", parameter) will:

```
parameter  action
0          erase to end of screen
1          erase from start to current (including
           current)
2          as noted clear screen leaving cursor in
           place
```

And printf("\0x1b[%dK", parameter); is the same with line replacing screen.
printf("\0x1b[%d;%dH); can be: printf("0x1b[H"); for home cursor or printf("\0x1b[%dH", line); works, as does printf("\0x1b[;%dH", column);

```
cursor     sequence
  up       \0x1b[A
  dn       \0x1b[B
  rt       \0x1b[C    (make last two lines "\x1b" into
                      "\0x1b")
  lt       \0x1b[D
```

Note that a parameter can squeeze into the string just after the "[" for a rep count.

index = \0x1bM
reverse index = \0x1bD

And then we get into the attributes that I won't get into because I am not sure what they'd be implementing. I believe the general sequence would be "\0x1b[%d;%d;...%dm" (any sensible number of %d's in there separated by ";" will work). If the parameter is "0" then all attributes are canceled.
These are ANSI x3.64 defns that, hopefully, they followed in ANSI.SYS but I have no PC to try them on. I'm just reading my terminal defns.
<^_^>

THE EOF( ) FUNCTION

pascal/turbo #1036, from mwelch (Mark Welch)

I've been fiddling with a program that will invert a file or set of files (text files) into a binary tree of the words, so I can easily search and locate all instances of a word within the file set. (I know it's been done before, and yes, I'll post it when it's ready.)

To try to cut down execution time, I've been using

(continued)

$9,900 = 563 MegaBytes

For All PC, XT, AT and RT Applications

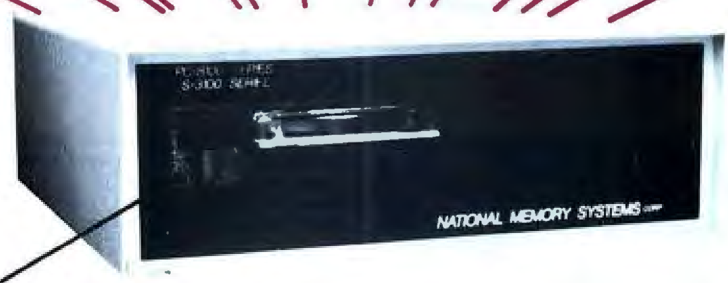
$9,900	$8,500	$5,900
563MB	368MB	168MB

Optional integrated 60MB tape back-up
$1,495

- Data rate to 2.4MB/Sec.
- Average access times as fast as 15 milliseconds
- Breaks DOS 32MB limit

8000 Series 8 inch Winchester technologies intelligently interfaced to microcomputers

$4,500	$4,600	$6,400	$7,500	$10,900
51MB	85MB	140MB	190MB	280MB



Optional integrated 60MB tape back-up

8100 Series 5.25 inch Disk/Tape memory systems

8000A Series

Mainframe: NMS Disk/Tape systems and controllers provide economical, compact, and high performance mass storage that can be formatted as one volume. When you need performance and/or storage on your PC, XT, AT and RT, you need the industry standard NMS 8000A series with Multifunction Controller. The SMD-SCSI single board controller you've been waiting for is available now . . . The NMS 8000A.

Tape Back-Up: NMS provides ANSI-IBM format ½ inch tape systems and 60mb of file-by-file/streamer ¼ inch cartridge tape systems. (NMS 9000 and PC .25 series).

Laser Optical: Complete, 12 inch media, 1000MB memory systems that support DOS applications and format as one volume.

Warranty: Controller . . . 24 months (RTV), and 12 months (RTV) all other system parts.

Call us today!

National
Memory
Systems
CORPORATION

355 Earhart Way
Livermore, CA 94550

(415) 443-1669

8100 Series

Selection: Only NMS provides selection of 8 inch and 5.25 inch technologies. Where speed, power and networking are important . . . NMS is there.

Tape Back-Up: An option you control . . . file by file and streamer back-ups. 60 megabytes per cartridge. Fast, reliable, and easy to use. One half inch ASCII/EBCDIC tape systems are available too.

DOS Limits: They don't exist with NMS memory systems. We were the first to break the barriers for you!

PRICES VALID THROUGH 12/31/86

— When gems lose their luster . . . NMS still shines —
— Compatible with all popular networks —
— One year RTV warranty —

Inquiry 205

the Execution Timer from the Turbo Power Utilities, and after much agony I've found the single largest bottleneck: the Turbo eof() function. While each call to it takes only a *very* short time, my program calls it very often (2716 times for a typical run), and it takes longer than most other functions. As a result, it accounts for about 30% of the program's total execution time.

Is there any way to cut down on the number of calls to eof() or to speed up eof() or, conversely, to use some other routine?

The code segment in question is:

```
...
var  f: file of char; (* can't be text, since I need to
know exact file pos *)
...
repeat
...
until eof(f);
...
```

pascal/turbo #1053, from jimkeo (Jim Keohane)
a comment to message 1036

Invert.pas speedup via blockread

Mark, the following reduced time for inverting a file from 3.40 seconds to 1.27. See "Now" calls for start-stop time displays.conds.   - Jim

PS- If garbage past the ^Z in the last block is a problem, then the "for i:=0 to 127" could be changed to "for i:=0 to last" where last was set by a "scan" for ^Z in the current block. See very fast "scan" function in Turbo #757.

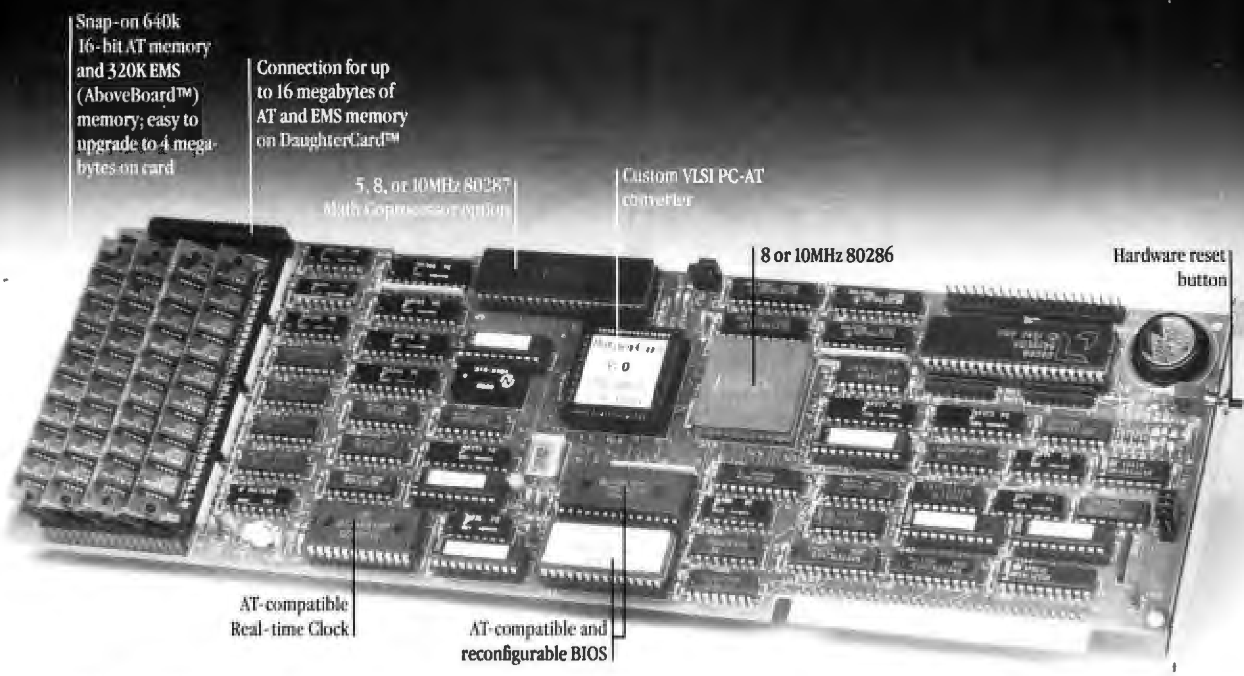
PPS- To reduce mem requirements, I would move "Text:S" to end of record description. Then, instead of New(recptr) I would use GetMem(recptr,sizeof(recptr^)-WordLength+Length(St)); That way, if St="HEAD", you save 10 bytes of memory!

```
{==========added for DOS time call================}
type intregs = record
               case integer of
0:(ax,bx,cx,dx,bp,si,di,ds,es,flags:integer);
               1:(al,ah,bl,bh,cl,ch,dl,dh:byte)
             end;
procedure time(var
hours,minutes,seconds,hundredths:integer);
var r:intregs;
begin
 r.ah:=$2c;
 msdos(r);
 hours:=r.ch;
 minutes:=r.cl;
 seconds:=r.dh;
 hundredths:=r.dl
end;

procedure now;
var h,m,s,hs:integer;
begin
 time(h,m,s,hs);
 writeln(h,':',m,':',s,':',hs)
end;

{================ADDED to InvertFil==============}

{p.s. changed fil to just 'file', not 'file of char'}

var block:array[0..127] of char;
    j,k:integer;

Begin   (* InvertFil *)
```

(continued)

SOTA Technology Announces

▷MotherCard 5.0™◁



Put this board into your old PC and get full AT hardware/software compatibility. Only $995!

Speed is everything

You started using a computer to get your work done faster. Now you'd like your computer to go faster, right?

Easy. SOTA Technology's *MotherCard 5.0* is the quickest and easiest way to upgrade your favorite old PC.

Easy to install: No hardware or software changes required.

Easy to use: simply run your same familiar programs. They just run *seven to ten times faster.*

And it's easy to buy: $995 makes your current IBM PC system fully AT compatible; that's a lot less than you'd have to invest in any off-brand machine, set up the way your current system is configured.

SOTA's *MotherCard 5.0* is the easiest way to *get faster* and still keep your "old reliable" IBM system.

Compatible with all software.

An ingenious new concept of putting the machine BIOS (Basic Input Output System) into battery-backed RAM insures long-term compatibility with all your software.

You will not only be able to run all your tried-and-true PC software, but anything written specifically for the PC AT, as well.

Hardware compatibility, too!

The SOTA Technology *MotherCard 5.0* is compatible with any other board in your PC. EGA, CGA, LAN, internal modems and tape backup systems all work without a hitch.

The megabyte of 16-bit AT and EMS on-board memory can be expanded with our DaughterCard to anywhere from 2 to 16 megabytes.

Our complex VLSI PC-AT converter circuit handles any signal coming through that requires an I/O call on the system bus.

This allows the 286 to do its 16-bit work at full speed and still talk to the outside world over the 8-bit IBM PC bus.

Magic? No. This is the only way to be fully AT-compatible and still keep your reliable old equipment. And we're the only company doing it this way.

If any new "protected mode" DOS comes along, we know we'll be compatible. It's in the nature of the technology. And, as our name suggests, we deal with *State Of The Art Technology.*

Order your MotherCard 5.0 today! Money back guarantee.

While we are still establishing our dealer network, we want to make the *MotherCard 5.0* available to you without delay.

To order, just call SOTA Technology during West Coast business hours. You can use any major credit card or send a company purchase order or check.

SOTA Technology *unconditionally guarantees the MotherCard 5.0* against defects of workmanship or any compatibility problems. We will gladly replace any card or refund your money in full if you return it within 30 days after receipt.

TO ORDER, CALL 800-237-1713
In California, call 408-245-3366.

SOTA™

STATE OF THE ART TECHNOLOGY™
SOTA TECHNOLOGY, INC.
657 N. Pastoria Blvd, Sunnyvale, CA 94086

Trademarks: IBM, PC/XT/AT are trademarks of IBM Corp.
AboveBoard is Intel's. MotherCard, MotherCard 5.0 and DaughterCard are ours.

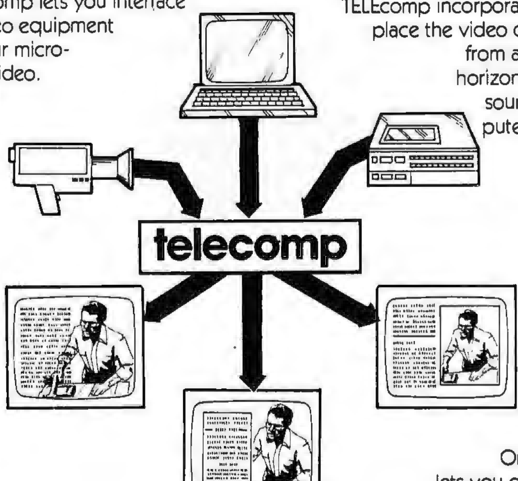
WE'RE TEACHING COMPUTERS AND VIDEO NEW TRICKS!

INTRODUCING TELECOMP COMPUTER/VIDEO COMBINERS

Now, Add Full Color Video From Any Video Source To Your Micro-Computer Display...

VIRTUALLY NO LIMIT TO THE APPLICATIONS

With just a few simple cable connections, TELEcomp lets you interface almost any micro-computer with virtually all video equipment without limitation. Imagine the new horizons your micro-computer can attain with the added power of video. Simulations become real! Demonstrations come alive! Instruction and training can be combined with data displays. Computer graphics and animations mixed with video take on a new dimension. Research and development groups, scientists, doctors, and trainers, to name a few, can generate computer data and make real-time video observations all on a single monitor screen.



RECORD THE COMBINED OUTPUT ON VIDEO TAPE OR DISC

And with the TELEcomp 2000 Series, you can project the signal using any composite or RGB video projector. In fact, you can broadcast the signal or even transmit it by satellite for applications such as teleconferencing, because the TELEcomp 2000 returns an NTSC standard video signal, equal to the quality of the video input—with the computer output combined!

USE WITH MOST MICRO-COMPUTERS... NO SOFTWARE REQUIRED

TELEcomp has been engineered to provide you with the maximum in benefits and versatility. TELEcomp interfaces with virtually all computers with TTL or composite video output. This includes IBM PC/XT/AT and compatibles, Apple, Commodore and countless others. Since TELEcomp does all the work, no software is required. Your computer remains free to run the software you want and to perform the jobs you need. TELEcomp's external peripheral design doesn't use up your computer's valuable board slots or memory.

ADD, COMBINE, MIX OR SUPERIMPOSE VIDEO AND COMPUTER

TELEcomp incorporates full window control which permits you to place the video or computer image anywhere on the screen—from a dot to a full screen image (4 dials control the horizontal/vertical window image size). Overlay and source selector switches let you view either computer, video or combined image and then choose which source (computer, video or mix) is dominant. For more flexibility there's even an overlay intensity control. TELEcomp 2000 Series converts your TTL signal to composite video and provides 2 amplified video outputs along with analog RGB output. It even includes a built-in video sync generator in case you want to record, display or project your micro-computer's output when no other video source is present.

MORE VALUE AND A LOT MORE FEATURES STARTING AT UNDER $400.00

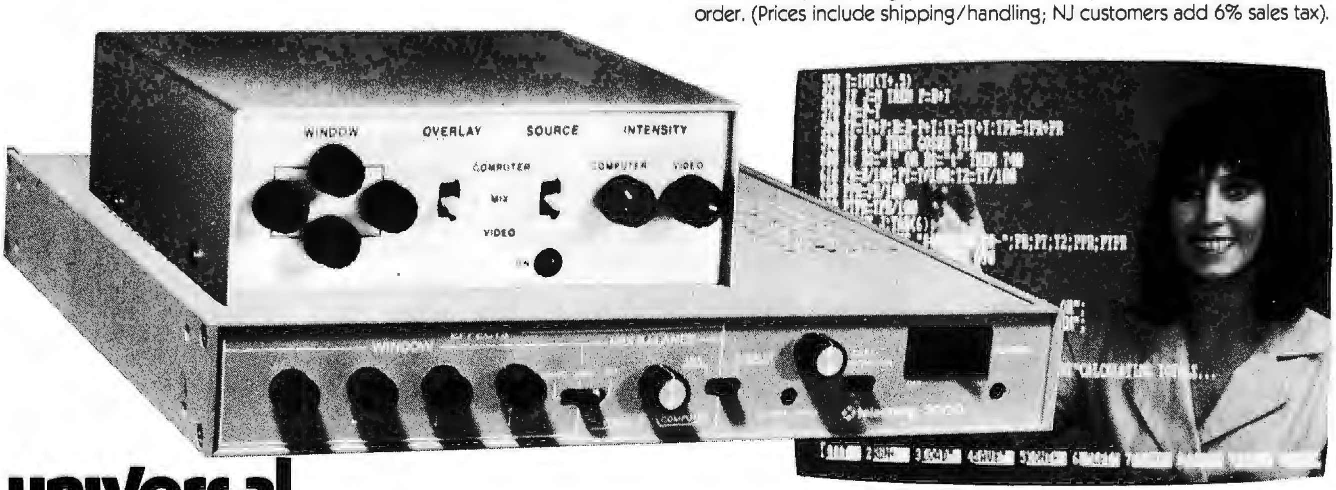
Order the TELEcomp you need. TELEcomp-1000 lets you combine genlockable B+W camera video with computers that have composite video output for only $399.95. Choose the TELEcomp-2000M to combine monochrome computer output with any color video source for $995.00. If your needs require color computer output combined with any color video source, order the TELEcomp-2000C for only $1595.00.

SPECIAL LIMITED TIME OFFER

Order a TELEcomp-1000 with our hi-res. B+W genlockable video camera. Get both for only $599.00. Save over $250.00.

ORDER NOW... AT THESE INCREDIBLE BREAKTHROUGH PRICES!

All of the capabilities of the remarkable TELECOMP product are available now. With our 10 DAY RISK FREE TRIAL, you can order knowing that TELEcomp will perform to your satisfaction or we'll return your money! Don't delay, order now—call toll free 800-431-6000 and have your Visa, Mastercard or American Express card ready. Or order by writing to the address listed below, enclosing your check or money order with your purchase order. (Prices include shipping/handling; NJ customers add 6% sales tax).



universal micro™

195 Bonhomme Street, P.O. Box 488
Hackensack, NJ, USA, 07602.
International telex 219-804
NJ call 201-487-6340
Toll-Free 800-631-0867

telecomp

COMPUTER/VIDEO COMBINERS™

Dealer, distributor and OEM inquires are encouraged.

Inquiry 291

```
  Writeln('Invert: ',FN); (* let user know what file
we're fiddling with *)
  NumFiles := NumFiles + 1;
  If NumFiles > MaxFiles (* remove this when FileList
is a linked list *)
    then begin (* crash impolitely, trashing all work
done so far *)
           writeln('Too many files inverted: maximum is
',MaxFiles);
           Halt;
         end
    else FileList[NumFiles] := fn;
  Assign(fil,fn);
  Reset(fil);
  St := '';
  Now;          {display time to 100ths sec}
  k:=0;                                      {block no}
  blockread(fil,block,1,j);                  {read first
block}
  while (j=1) do
  begin
  for i := 0 to 127 do                       {128 bytes
per block}
  begin
      c:=block[i];
      If ((c >= 'A') and (c < 'z'))
         and ((c >= 'a') or (c <= 'Z'))
        then st := st + UpCase(c)
        else if (Ord(St[0]) > 0)
          then
            begin
              matchKey := false;
              case st[1] of
                'A': matchKey := (st = 'A') or (st =
'AND') or (st = 'AN');
                'I': matchKey := (st = 'IN') or (st =
'IS') or
                                 (st = 'IT') or (st =
'ITS');
                'N': matchKey := (st = 'NOT');
                'O': matchKey := (st = 'OR') or (st =
'ON') or (st = 'OF');
                'T': matchKey := (st = 'THE') or (st =
'TO');
                'Y': matchKey := (st = 'YOU');
              end; (* case *)
              if matchKey
                then
                  St := ''
                else
                  begin
                    New(StPtr);
                    StPtr^.F := NumFiles;
                    StPtr^.FilPosPtr :=k+i-ord(st[0]);
{FilePos out}
                    StPtr^.Next := Nil;
                    AddWord(St,StPtr);
                    St := '';
                  end;
           end;
  end;
  k:=k+128;  {add in 128 bytes per block}
  blockread(fil,block,1,j)    {read next and j=1 if
more}
  end;
  Now;
  Writeln('  Done Inverting ',FN);
End; (* InvertFile *)
```

Data Compression Tricks

pascal/turbo #624, from jimkeo

This is a CORRECTION/REPLACEMENT for msg #619, now withdrawn.

You have a VERY LARGE data file filled with REAL numbers, what can ya do? Well, here's what SOME can do. Turbo REALS take 6 bytes. That's a lot of precision

(continued)

and a big range. Suppose your numbers can be held in an integer. "But my numbers are outside the range -32768..32767," you say. Well, do ya need them kept exactly or just to a certain precision? Like for graphing purposes. Turbo REALS have an 8-bit signed exponent and a 40-bit signed fraction. The following code has 2 routines called "i2r" and "r2i" which swap numbers between integer and real. The integers are really mini-reals with a 5-bit UNSIGNED exponent and an 11-bit UNSIGNED fraction. The example assumes positive so I don't need that 1 bit for exponent sign. The fraction sign only is negative when the value is less than 2. I get around this by adding 2 in "r2i" and subtracting 2 in "i2r" thus saving another bit. The "integers" can now hold values into the BILLIONS (mebbe I shudda called this a sneakie). The precision is to the nearest 16th when number less than 613, nearest 8th when less than 1025, nearest whole number when less than 4094, nearest even number when less than 8190 and so on. Depending on your needs the below routines might save you a lot of disk space, disk I/o time and RAM! You can always modify the routines to return a 3-byte structure (as a proc, not func) and can thus add greater range and/or more precision and still save 50%!

I use the routines all the time in stock market applications and in many graphics applications. Hope they're useful to you. - Jim Keohane

P.S. The routines are in a program that asks you to enter numbers, then shows you what that number becomes after packing and after unpacking. Use the program with the range of numbers your application requires to see if it is adequate. I have other data compression techniques I'll post. Send me mail if you have a particular data type in question (strings, chars, etc.).

```
program tr;
type both = record case integer of
            0:(r:real);
            1:(b:array[0..5] of byte)
            end;
     str3=string[3];

var r:real;i:integer;

function r2i(r:real):integer;
var x:both;
    i:integer;
begin
x.r:=r+2;
i:=(x.b[0] and $1f) shl 11; {5 lsb of exponent}
i:=i or ( (x.b[5] and $7f) shl 4); {hi 7 bits of
precision}
i:=i or (x.b[4] shr 4); {lo 4 bits of precision}
r2i:=i
end;

function i2r(i:integer):real;
var j:integer;x:both;
begin
x.r:=0;
x.b[0]:=$80 or ((i shr 11) and $1f);
x.b[5] := (i shr 4) and $7f; {hi 7 bits precision}
x.b[4] := (i and $000f) shl 4; {lo 4 bits precision}
i2r:=x.r-2
end;

begin
repeat
write('enter real number.....');readln(r);
writeln('orig=',r:12:6,' int=',r2i(r),'
packed/unpacked=',i2r(r2i(r)):12:6);
until r=0;
end.
```

Extra advantage: The mini-reals have the bit pattern eeeeehhhhhhhllll where eeeee is a 5-bit exponent,

(continued)

hhhhhhhllll is the fraction broken into high & low significant parts. This means that for real #s less than 2**15-3 (32765) the mini-real retains the same order as the full real. If you wanna test for >, <, =, <=, or >= you can using the mini-real (integer) and not have to convert back to Turbo REALS before every comparison.

Replacing Console Output Driver

pascal/ms.pascal #36, from mcmahon (Steve McMahon)

I'm working with MS Pascal 3.31 (Microsoft's, not IBM's version) and would like to find out if there is any way to replace the console output mechanism in such a fashion that the full Write library can still be used.

What I'm after is something like Turbo's ability to use the ConOutPtr to replace the built-in, underlying console output procedure with another.

This version of MS Pascal has the console output seemingly inextricably linked with the DOS2 file system functions. That's fine if you're willing to accept slow console output that can be killed (along with the program) by an inadvertant control-c from the keyboard, but not OK for anything else.

I'd like to get it to use DOS function 6 for output for program versions to be used on generic MS-DOS machines and the IBM BIOS functions on program versions for use on IBMs. Do I have to give up the write, writeln procedures to do this?

Any help will be much appreciated.
Steve McMahon

---

pascal/ms.pascal #37, from johnm (John McLaughlin)
a comment to message 36

How about using (or writing) a C-type "printf" function with varying numbers of arg's. "printf" may be in pascal.lib, but if not, it would be fairly easy to write. I'm a heavy MS Pascal user.

johnm, portland, or.

---

pascal/ms.pascal #38, from mcmahon
a comment to message 37

The point of finding a way to replace just the driver is to avoid the work of having to redo the terminal I/O routines. If I have to redo them, I'd probably go with a variety of special-purpose routines (e.g., WriteString, WriteLn, WriteInt ... a la Modula-2) rather than one big, relatively non-portable one.

But, there >>must<< be some way to accomplish the driver replacement. If Microsoft really buried the console I/O inextricably in the file system it seems to me they've compromised the possibility of porting code by forcing the use of non-standard constructs to achieve adequate terminal communication.
Steve McMahon

---

pascal/ms.pascal #41, from jsprowl (Jim Sprowl)
a comment to message 36

Steve, you should write your screen routine in assembler and link it into your Pascal program as an "extern" function. I wrote a full VT100 emulator that also emulates the DOS ANSI.SYS device driver. It includes Pascal callable entries that let me pass in single characters as well as LSTRINGS. I use ANSI standard escape sequences to position the cursor, etc. If you write an assembly language procedure to accept an LSTRING, watch out -- Microsoft Pascal passes an LSTRING with two words, not one, pushed on the stack.

(continued)

The first word is the LSTRING length, and the second word is its address within the DATA segment. This is not documented anywhere that I can find. There are some good examples in the IBM Pascal manual of how to pass parameters to an assembly language program.

Assembly language functions are really simple -- just place the return argument in the AX or AL register, if I remember correctly. Make sure to pop off just the same number of bytes as Pascal pushed onto the stack, or the computer will head off into never-never land.

If you pass a byte to an assembly language routine by value, the byte will be placed on the stack along with a byte of padding so that a word is placed on the stack -- caveat emptor!

Let me know if you would like some sample code.

---

pascal/ms.pascal #43, from mcmahon
a comment to message 41

Thot sounds wonderful, but ...

What I was really after was a way to replace the terminal output driver and retain the ability to use all the compiler magic of the Write/WriteLn routines. Writing assembler EXTERN routines is no particular problem: I've done lots before and would have to do another to do what I'm proposing. Except that what I'm proposing doesn't seem to be possible, judging from the reaction here.

To re-iterate:

In Turbo, you can replace the console output driver with a statement like:

```
ConOutPtr := Ofs( MyReplacementRoutine );
```

where MyReplacementRoutine is a function that takes a character as a value parameter. So, in Turbo, you can write a little inline code to handle an alternative console output technique, initialize the ConOutPtr at the top of the program and forget about it!

I guess Microsoft doesn't believe we'd really like to use the Write/WriteLn routines for anything other than slow, redirectable output!

Steve McMahon

ASSEMBLY

Though high-level languages have gotten the lion's share of publicity in the last few years, assembler remains the language tool of choice for many applications. The Assembler conference on BIX has topics covering most of the popular microprocessors in use today. These excerpts from the 8088 topic cover screen writing attributes and writing custom interrupt handlers.

REPLACING DOS INTERRUPT HANDLERS

assembler/cpu8088 #27, from leroy (Leroy Casterline)

I am an experienced C programmer and have come to the point in my life where learning 8088 ASM is necessary. I need to write a few routines in ASM to be called from (Lattice) C and am looking for pointers (no pun intended) to good sources of learning material or any tips you folks would be willing to share with me.

I need to write a couple of interrupt service routines (pref in C) and must come up with a method of replacing the DOS interrupt handlers with pointers to

(continued)

You Can Get Programming Software Anywhere

Here Are 10 Reasons Why You Should Insist On LIFEBOAT.

Everything the professional programmer needs, from a wide selection to our Tech Support Hotline to our competitive prices! Lifeboat's got the answers ...and the reasons why you'll insist on Lifeboat!

1 Latest Versions of Top-Rated Programs ... An unbeatable selection of up-to-the-minute programs. Including...

ADVANTAGE C++ • Advanced Trace 86 • ASCOM IV • BASIC_C • Bastoc • BetterBASIC, Math Support, Runtime • Brief • Btrieve • C-86 • C-FOOD Smorgasbord • C Power Packs • csd C Source Debugger • C-terp • C-Tree • Essential C Utility Library • Curses • CVUE • dBC III • db_Vista • dBX • Dr. Halo II • EMACS • Essential Graphics • Fast C • FirsTime • Float 87 • Greenleaf Comm Library • Greenleaf Functions • GSS Graphic Development Toolkit • GSS Toolkit — Plotting • Halo • Hammer • Informix • Instant C • Lattice C • Let's C • LMK • LSE • Mark Williams C • Microsoft BASIC Interpreter, C, COBOL, FORTRAN, Macro Assembler, Pascal • QuickBASIC • Norton Utilities • Opt-tech Sort • Panel • Pasm86 • PC-Lint • Periscope I • Periscope II • Pfix86 Plus • PforCe • Phact • Plink86 Plus • Pmaker • Pmate • PowerMake • Pre-C • ProFORTRAN • ProPascal • Ptel • Quelo Linker/Assembler • Rtrieve • RUN/C-The C Interpreter • RUN/C Professional • Text Management Utilities • TopView Toolbasket • VEDIT • VEDIT PLUS, V-PRINT, V-SPELL • Wizard C • WINDOWS FOR C • WINDOWS FOR DATA • Xtrieve • ZView

...and so much more!

2 Wide Selection ... Over 200 Programs to choose from, including C++, C, BASIC, Pascal, RPG II, Translators, Editors, Function Libraries, Language Utilities, Data Base Record Managers, Graphics, Communications and more!

3 Tech Support Hotline ... Immediate help with technical problems for all the software we publish. Lifeboat's there when you need us most!

4 Programming Tools Guide... a comprehensive reference catalog to help you select programs best suited to your needs.

5 Personal, Professional Advisor ... Your own Account Rep to advise you on the best programming software for your specific needs.

6 Automatic Update Service ... The easiest way to stay on top of the software market, with automatic notification of all new and updated programs, pre-tested and reviewed by Lifeboat.

7 Newsletter, Conference & Seminar Programs ... to keep you up-to-date on the newest products, technical developments and technical support relating to the world of software programming.

8 Special Orders... In addition to our extensive selection, Lifeboat offers customers the option to select from an open ended product line.

9 Competitively Priced ... Our prices are competitive with those of other suppliers that offer none of the essential services to which we are committed.

10 Prompt Delivery... Our huge inventory allows Lifeboat to ship most of our products the same day they are ordered, including overnight deliveries.

BRAND NEW FROM LIFEBOAT

ADVANTAGE C++

They say you can't be all things to all people. But Lifeboat's *ADVANTAGE C++* proves that you can be! This exciting new product, developed by AT&T, represents a major programming breakthrough. By introducing the concept of classes, it enables programmers to use object-oriented programming methods. Plus it gives you a host of other major improvements over C.

***ADVANTAGE C++* gives you the ability to create new functionality to solve your applications problems. It allows programmers to more productively build large and sophisticated applications. All the benefits of C, without its limitations.**

***ADVANTAGE C++* is available for the most popular C compilers, Lattice C and Microsoft C.**

So why be limited to just C... when you can have all these pluses! *ADVANTAGE C++* exclusively from *LIFEBOAT.*

1-800-847-7078 In NY: 914-332-1875

LIFEBOAT

The Full-Service Source For Programming Software

Inquiry 164

International Sales Offices
Australia
Fagan Microprocessor Associates
Phone: (6) 3699-9899
Canada: Scantel Systems
Phone: (416) 449-9252

England: Grey Matter, Ltd.
Phone: (44) 364-53499
Italy: Lifeoat Associates, S.p.A.
Phone: (02) 656-841
Japan: Lifeboat Japan
Phone: (03) 293-4711

Spain: Micronet, S.A.
Phone: (34) 1-457-5056
The Netherlands:
GIGA Compter Products
Phone: (31) 10-771846

West Germany:
MEMA Computer GmbH
Phone: (49) (069) 34 72 26
Omnitex
Phone: (49) (076) 23 61820

the C functions that will actually process the interrupt. I need to write a general-purpose routine that will allow me to specify the interrupt number to replace and a pointer to the C function. Anyone had experience with this?

Thanks in advance,
Leroy

---

assembler/cpu8088 #28, from barryn (Barry Nance)
a comment to message 27

Leo Scanlon's "IBM PC & XT Assembly Language: A Guide for Programmers" is a decent text on the subject.

I could put about 60 lines of assembler code here that replaces an interrupt vector (part of a program I wrote). Want it?

---

assembler/cpu8088 #29, from hvanderbilt (Henry Vanderbilt)
a comment to message 28

Somebody's bound to get some good out of it. Put it up!

---

assembler/cpu8088 #30, from barryn
a comment to message 29

OK, you asked for it. (Seriously, it's a code fragment from a GRAPHICS.COM-type replacement that I wrote for the Okidata 92/93 printers. It is about 60 lines long, so, for those not wanting it, you can skip the next message. BTW, pardon that it occurs in uppercase; I apologize in advance.)

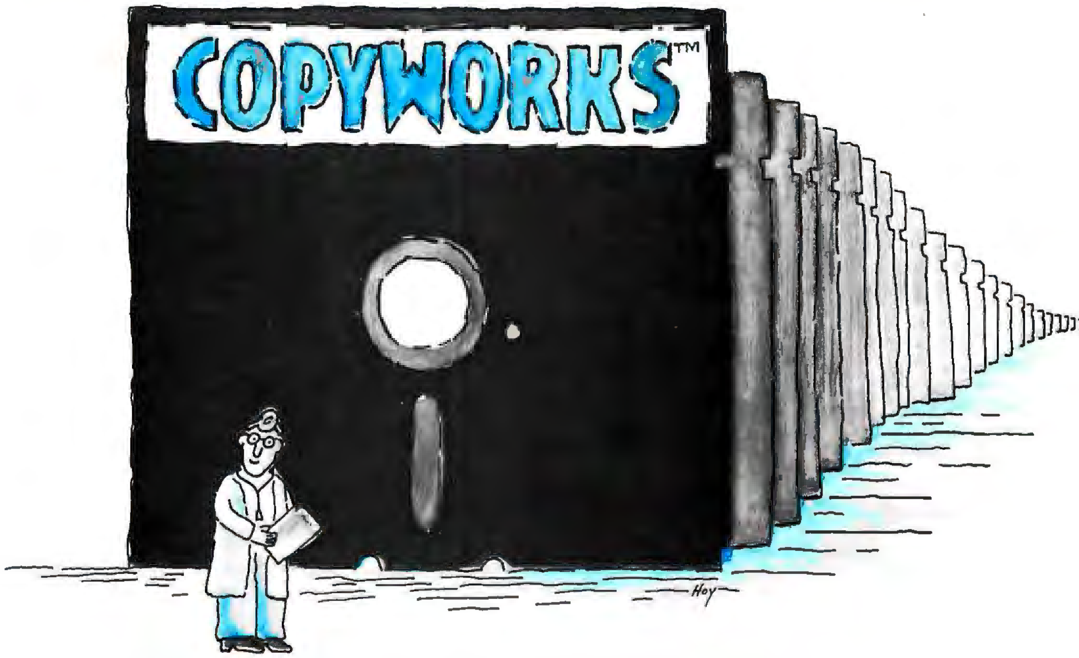
---

assembler/cpu8088 #31, from barryn

```
TITLE: Code fragment for interrupt replacement
;
;        INITIALIZATION CODE (NOT PART OF RESIDENT)
;
         INIT_CODE:
              ASSUME  CS:CODE
              PUSH    DS                ; set up DOS
                                        return
              MOV     AX,0
              PUSH    AX
;
;             IF WE'VE BEEN HERE BEFORE, THEN
;             SAY SO AND EXIT GRACEFULLY.
;             (JUST RETURN TO DOS; WE'RE ALREADY
;             RESIDENT)
;
              ASSUME  DS:INT_5_LOC
              MOV     AX,INT_5_LOC      ; establish
              MOV     DS,AX             ; addressability
;
              MOV     DX,INT_5_OFS
              ASSUME  DS:CODE
              MOV     AX,CS
              MOV     DS,AX
              MOV     AX, OFFSET MAIN_ROUTINE
              CMP     AX,DX             ;BIOS interrupt
                                        in place?
              JNE     P010_CONTINUE ;yes, go ahead &
                                        replace
;
              MOV     DX, OFFSET ERRMSG
                                      ;no,say'already
                                      here'
              MOV     AH,9H
              INT     21H
;
              RET              ;FAR RETURN TO DOS
;
;        outline of initialization logic:
;             1. REPLACE INTERRUPT 5 VECTOR ADDRESS
;                    (WITH ADDRESS OF
;                    MAIN_ROUTINE)
```

(continued)

THE CURE FOR COPY WOES.



COPYWORKS®, is the ultimate software duplication system, unequalled for versatility, power, and customized options.

COPYWORKS is designed by Vault Corporation to maximize your PC/MS-DOS® software production capabilities and will enhance your disk production output whether you hand feed disks or use an autoloading machine such as Vault's PROLOADER®.

COPYWORKS verifies good diskettes and rejects bad ones, and is just as effective at copying unprotected software as it is at producing copy protected diskettes using Vault's award winning, patented PROLOK® protection, or serialized UNILOK® for networking or easy field updating of protected programs.

COPYWORKS uses an IBM® PC as the host computer, but can be configured to work in a variety of environments under many different copy formats.

Multiple slash (/) options allow you to:

- Produce single-sided or double-sided diskettes.
- Create disks with 8 or 9 sectors per track.
- Use Vault fingerprinted diskettes or plain diskettes.
- Create hidden files, system files, and read-only files.
- Assign serial numbers to successfully copied diskettes.
- Assign password for absolute access control.
- Execute custom programs during the duplication process.

Additional data files may be purchased allowing you to produce:

HARD DISK PROLOK and UNILOK diskettes with easy to follow help screens enabling end users to install/uninstall protected programs and data files onto a hard disk and to make executable backup copies on floppy diskettes.

CHRONOLOK® with a limited number of runs of protected programs from 1-32,000 for demo disks or sale-by-situation such as psychological testing, where specified numbers of execution are desired.

No longer do you need to be locked into a single duplication configuration. COPYWORKS gives you the freedom of choice that is every American's birth right.

VAULT™ CORPORATION

For ordering your COPYWORKS package call tollfree:
800-445-0193 (US)
or 800-821-8638 (CA)
or write: Vault Corporation,
2192 Anchor Court, Unit C,
Newbury Park, California 91320.

COPYWORKS.™
THE ULTIMATE DUPLICATION SOFTWARE.

```
;               2. SAY HELLO TO THE PERSON WATCHING THE
                   SCREEN
;               3. SET PRINT SCREEN STATUS FLAG = ZERO
;               4. TERMINATE BUT STAY RESIDENT
;
P010_CONTINUE:
;
;       replace the int 5 vector
;
                MOV     AX, OFFSET MAIN_ROUTINE
                MOV     DX,AX
                MOV     AL,05H
                MOV     AH,25H
                INT     21H
;
;       say hello
;
                MOV     DX, OFFSET HELLO
                MOV     AH,09H
                INT     21H
;
;       set print screen status flag byte to zero
;
                ASSUME  DS:XXDATA
                MOV     AX,XXDATA
                MOV     DS,AX
                MOV     AX,0
                MOV     STATUS_BYTE,AL
;
;       terminate but stay resident
;        (last address of resident portion is
INIT_CODE)
;
                MOV     DX, OFFSET INIT_CODE
                INT     27H
;
```

BASIC

The BASIC programming language remains one of the most popular ways of getting things done on a micro. The BASIC conference on BIX contains questions, answers, and statements covering problem solving, design philosophy, and programming style using BASIC. Two conversations follow, the first on the FIELD statement in Microsoft BASIC and the second on tips for a compact bit-counting program.

THE FIELD STATEMENT

basic/misc #46, from btonkin (Bruce Tonkin)

I have often heard that Microsoft BASIC users are intimidated by the FIELD statement. I wasn't, and most of the people I have talked to have not been, either. They might not be representative, though. Would any of you like to see the FIELD statement abolished? If so, what would you replace it with? I'd like to hear some comment, especially from people with limited knowledge of BASIC.

basic/misc #48, from dmick (Dan Mick)
a comment to message 46

I sure wish one could *easily* and *clearly* FIELD a file buffer with an array, without having to go through the "dummy variable" crap you usually have to, like so:

```
FOR I=1 TO 256:FIELD #1, (I-1) AS DUM$, 1 AS
CHAR$(I):NEXT I
```

Why not a syntax like FIELD #1, 256 AS A$() and let BASIC take care of it? (A minor quibble, but otherwise FIELD makes a lot of sense and is a good way to read other language datafiles, like FORTRAN.)

(continued)

A$(), #1, 256 BY 1? Or FIELD #1; A$(256) BY 1? Maybe even FIELD #1, A$(1-256) BY 1 or FIELD #1, 1 AS A$(256)

Is the horse dead yet? (smirk) Is it obvious I've read Pascal and PL/I? (grin,chuckle)

---

basic/misc #51, from btonkin
a comment to message 50

Your examples would seem ambiguous, syntactically. The nice thing about the CFIELD statement is that you could have an integer array specify the individual FIELD lengths. It's not clear that any of yours could do the same. Further, A$(256) already has a meaning in BASIC; using it the way you did would be, er, confusing at best. How would you like FIELD #1, USING F(), A$() Where F() is an array holding the field lengths, and A$() is the array of separate fields, in order?

No, I don't think the horse is dead yet. Kick 'im again, Dan.

---

basic/misc #52, from dmick
a comment to message 51

I have an even better thought. How about having the FIELD statement continue from the last statement unless you add the option ,CLEAR or ,NEW to the parameter list? No, better would be a CLEAR FIELD or CLEAR FILENUM statement. Although if you close & re-open the file, the field is cleared, right? How about that? Is there really any reason FIELD should not mean continue specifying the map of this buffer from the last FIELDed position? (Except for compatibility with older code, which I admit is not a trivial exception.) Perhaps a new statement like CFIELD *is* the best option. Or maybe I should have just thought about it for 10 minutes before saying *any* of this.

---

basic/misc #53, from btonkin
a comment to message 52

Sounds like a good idea, except . . . I think there may be a problem with the "of this buffer" part. Could make for horrendous problems in compiled programs, especially given the possibilities of a GOTO statement mucking things up. The other problem I see is one involving multiple FIELD statements, all to the same buffer, existing at the same time. I use that one a lot, depending on how I want to look at the data at the time. If FIELD meant "continue from last fielded location," it would screw up multiple allocations for sure. I'm pretty sure CFIELD isn't the optimal solution, though. Someone's bound to think of something better. Any more ideas?

---

basic/misc #54, from barryn
a comment to message 53

(timidly...) I kind of like the way it works now. If I have to redefine part of a record, I can use the FIELD statement the way you've described. If I want to have multiple FIELD statements that want to follow one another across a record, I just put a FillerX$ item at the beginning of each new FIELD statement whose size exactly covers the already-defined portion of the record.

---

basic/misc #55, from btonkin
a comment to message 54

No need to be timid. Your approach is exactly what I've been doing for years. Still, it seems like rather a waste, and in interpreted BASIC you end up having to watch to make sure you don't make the filler longer than 255 characters (boom! string too long). That latter problem means you've got to use multiple dummy strings explicitly in the FIELD statement for long records, and the line length limitation will eat you up pretty soon.

(continued)

In compiled BASIC under CP/M, you have the same problem; no string can be longer than 255 characters. If you want to write transportable programs, you end up with scads of useless dummy variables cluttering up the code, and FIELD statements that look like their nearest ancestor was spaghetti.

So, the question: what should we do instead?

---

basic/misc #59, from bwaldheim (Bill Waldheim)
a comment to message 46

Bruce, I've been programming in BASIC (Microsoft & GW) for about as long as you have and I agree. The FIELD statement didn't cause me any problems once I became comfortable enough in BASIC to start messing around with files. The beauty of BASIC is that I could write little 10-line programs of the "Gee, I wonder if this'll work" type. I think some beginners have the mistaken idea that they must write useful code right from the start, but that ain't the way to really learn anything. The machines may lock up, but they don't blow up. And avoiding the FIELD statement won't make life any easier later on when writing let's say an accounting program that accesses 50 to 100 accounts. Having to scan them all from the beginning via a sequential type file will really spoil what might otherwise be a good program. So to all you novices out there, jump in and get your feet wet and DON'T be afraid of making mistakes. That's part of the learning process!!!

Bill Waldheim

---

basic/misc #60, from bwaldheim

Bruce, I'm putting these comments in as I scan the new messages, so that's why there are two from me. The CFIELD sounds like a sensible solution to me. As far as clearing things out, as was mentioned in an earlier message, I wish BASIC had a dBASE-type CLEAR statement that you could use to DISCREETLY release variables from memory instead of the shotgun it now has. How sweet it would be!

Bill

---

basic/misc #61, from btonkin
a comment to message 60

A possible way to work around the CLEAR statement would be a CHAIN that allowed an EXCEPT for variables you wanted killed. Often, I get programs where CHAIN ALL is too inclusive, and COMMON is far too much of a pain.

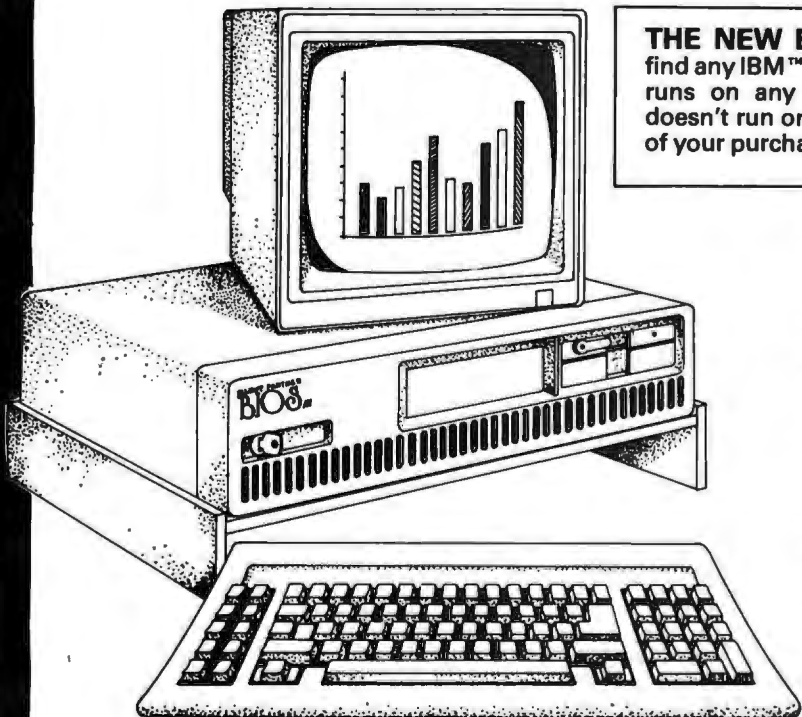
COUNTING "ON" BITS IN A WORD

basic/one.liners #15, from jimkeo

I forget where exactly on BIX this problem was first posed. The problem was to code a routine that calculated the number of one (1) bits in a word. One solution was to use each byte of the word as an index into a 256-byte table preset with the appropriate counts. Another solution that I suggested is the following: Function fnbt(I) returns the number of bits

(continued)

COMPATABILITY GUARANTEED
OR WE'LL EAT IT!



THE NEW BIOS™ GUARANTEE* - If you find any IBM™ Compatible Software Package that runs on any other compatible computer and doesn't run on the BIOS™ AT. We'll refund 100% of your purchase price.

- 640K RAM
- 80286 16 Bit Processor
- 8 Mhz 0 Wait States
  6 Mhz 1 Wait States
- 84 Key AT Style Keyboard
- MS™ DOS (Latest Version)
- GW™ Basic
- Advanced Diagnostics
- 20 MB Fixed Disk Drive Included
- Users Manual
- 8 Expansion slots
  6 16 Bit Slots
  2 8 Bit Slots
- Serial/Parallel Board
- Hard Disk Contoller Supports 2 Floppy & Two Fixed Disk Drives
- FCC Class B, UL, & CSA Approved
- Made in Japan
- 90 Days on site service by Intelogic Trace
- Extended on site 9 month service warranty only $109.00 dealer
- 550 pages of easy to use documentation

NOVELL COMPATIBLE
8 MHZ 0 WAIT STATES

PETER NORTON SYS INFO RATES THE BIOS™ AT 9.2

FREE—
90 DAYS ON SITE SERVICE

Distributed by:

DISTRIBUTION PLUS

Call **USA (800) 638-7654**
MD (800) 492-5316

For Dealer Near You

OFFER LIMITED TO UP TO RETAIL PRICE OF $2950*
Sugg. Retail Price

* Offer limited to up to dealer cost of $1999, 30 days after invoice purchase date. BIOS must be returned complete, as new, in original box with all accessories, manuals, diskettes, cables, etc. to apply. Program must be compatible with DOS 3.2 - must be original purchaser of BIOS™. Offer void where prohibited by law. Purchaser must present original invoice and proof of payment (cancelled check, charge slip, etc.) to Silent Partner Corp. to qualify. Silent Partner has the right to refuse refund for any legitimate reason.

***Individual Retailers Prices May Vary**

SILENT PARTNER
BIOS AT

7313 Grove Road, Frederick, Maryland 21701

BIOS™ is a trademark of Silent Partner Corporation
IBM is a trademark of International Business Machine.
MS™ DOS is a trademark of Microsoft Corp.
GW™ Basic is a trademark of Microsoft Corp.

>Count Bits...
   Touche! Juan. I guess I got a little carried away with functions. - Jim

---

basic/one.liners #18, from btonkin
a comment to message 16

Elegant solution, Juan! All I'd change in that one would be the part that reads B=INT(B/2). Better would be B=B\2; it does the same thing!

---

basic/one.liners #19, from jimkeo
a comment to message 18

>Count Bits...
   Bruce & Juan, I did some checking and the count-bits problem was first posed in soft.eng/algorithms #47. Another cute solution. Though it can't be contained on one line due to a GOTO, it has merit in that it loops one time for every 1 bit rather than 16 times. Statement 50 can be omitted if you have overflow checking disabled. It makes use of the fact that "B=B AND (B-1)" turns OFF the rightmost bit.
   - Jim Keohane
   PS- Taken from C code posted by rfoxmich in soft.eng/algorithms #80

```
10 GOTO 80
20 K=0
30 IF B=0 THEN RETURN
40 K=K+1
50 IF B=&H8000 THEN RETURN
60 B=B AND (B-1)
70 GOTO 30
80 INPUT "Number....",B
90 GOSUB 20
100 PRINT K
```

---

basic/one.liners #20, from juan
a comment to message 18

Now that is interesting. "\" you say? OK, I'll try it. What follows is a THREE liner. Believe it or not, it is a full implementation of the Quicksort algorithm. To call it, fill array LN$() with the elements to be sorted and set N to the maximum element to go up to when sorting. On return LN$(N) will be sorted!

```
1000 S=1:STACK(S)=1:STACK(S+1)=N:WHILE
S>0:L=STACK(S):R=STACK(S+1):S=S-2:WHILE
L<R:I=L:J=R:X$=LN$((L+R)/2):WHILE I<=J:WHILE
LN$(I)<X$:I=I+1:WEND:WHILE X$<LN$(J):J=J-1:WEND:IF I<=J
THEN SWAP LN$(I),LN$(J):I=I+1:J=J-1
1010 WEND:IF I<R THEN S=S+2:STACK(S)=I:STACK(S+1)=R
1020 R=J:WEND:WEND:RETURN
```

   Enjoy.
   |1|
   PS- If LN$() is very big, then dimension STACK() out to about 20 or 30, not much more, since by that point you should write an assembly language version.
|1|

---

basic/one.liners #21, from juan
a comment to message 19

As a matter of fact, I think it can. Try this out:

```
  10 K=0:WHILE (B<>0) and (B<>&H8000):K=K+1:B=B and (B-
1):WEND
```

it should work, but I don't guarantee it. I just winged that one.
|1|

---

basic/one.liners #22, from jimkeo
a comment to message 21

>WENDs...
   Hmmmmmm. Gotta brush up on my WENDs. Forgot about WHILE-WEND. - Jim
   PS- It won't work 'cause K will end up zero if B is &H8000 initially. Mebbe change "K=0" to "K=ABS(B=&H8000)" That should do it!

---

basic/one.liners #23, from juan
a comment to message 22

Question: Why do you need that &H8000 anyways?
|1|

---

basic/one.liners #24, from jimkeo
a comment to message 23

>&H8000 test
Otherwise, ya get OVERFLOW with "B-1" when B=&H8000 (-32768). Of course, you can disable OVERFLOW as I think I mentioned with the example. - Jim

MS-DOS

The activity in the MS-DOS conference reflects the fact that a high percentage of BIXen (and BYTE readers) own or use IBM PCs (or clones). This thread is centered on the very disturbing situation of a destructive program being distributed through public domain channels. The name of this program, ways to recognize other vandal programs, and possible ways to protect a machine from damage are discussed.

An Attack on the Public Domain

ms.dos/other #464, from dondumitru (Donald Dumitru)

TITLE: ARC513.COM is a TROJAN!
WARNING! WARNING! WARNING! WARNING! WARNING! WARNING!

   The program ARC513.COM is a TROJAN! It will alter the boot sector on your hard disk, preventing you from booting from your HD until you do a PHYSICAL FORMAT! Do *not* run this program! If you see it posted on a BBS, tell the sysop about it! I am attempting to track down the source of the copy that I have - wish me luck.

   I had warned the sysop of the board that I got ARC513.COM from, after I had heard previous warnings from BIX and FidoNews, but before I downloaded it to test it myself. He said he ran check4bomb on it, and it passed as "safe". From my (very limited) debugging of ARC513.COM, I'd say that that thing does weird things to make it hard to trace through. It is even possible that it tests whether it is being run under Debug or not.
   Donald
   PS- But it definitely does things to your hard disk. I ran it *twice*, and had to do a physical format after each run.

---

ms.dos/other #469, from dondumitru
a comment to message 467

These are the details of my experience with ARC513.COM.

   I first became aware of this program from messages posted on BIX by "wheelock," warning that he had heard rumors about ARC513.COM being a trojan.

   The next day, as I cruised past a local BBS, I noted that ARC513.COM was in the "recent uploads" section. I left a message to the sysop, warning him about the program. He left a reply thanking me for the warning, then another reply stating that he had run Check4Bomb, and that the program had passed as "safe" (so he left it posted).

   A couple of days later, in FidoNews #319, there was a warning (unsigned) that stated that ARC513.COM was

(continued)

COMPATABILITY GUARANTEED

OR WE'LL EAT IT!

THE NEW BIOS™ GUARANTEE* - If you find any IBM™ Compatible Software Package that runs on any other compatible computer and doesn't run on the BIOS™ AT. We'll refund 100% of your purchase price.

- 640K RAM
- 80286 16 Bit Processor
- 8 Mhz 0 Wait States
  6 Mhz 1 Wait States
- 84 Key AT Style Keyboard
- MS™ DOS (Latest Version)
- GW™ Basic
- Advanced Diagnostics
- 20 MB Fixed Disk Drive Included
- Users Manual
- 8 Expansion slots
  6 16 Bit Slots
  2 8 Bit Slots
- Serial/Parallel Board
- Hard Disk Contoller Supports 2 Floppy & Two Fixed Disk Drives
- FCC Class B, UL, & CSA Approved
- Made in Japan
- 90 Days on site service by Intelogic Trace
- Extended on site 9 month service warranty only $109.00 dealer
- 550 pages of easy to use documentation

NOVELL COMPATIBLE
8 MHZ 0 WAIT STATES

PETER NORTON SYS INFO RATES THE BIOS™ AT 9.2

FREE—
90 DAYS ON SITE SERVICE

Distributed by:

DISTRIBUTION PLUS

Call **USA (800) 638-7654**
MD (800) 492-5316

For Dealer Near You

BIOS™ is a trademark of Silent Partner Corporation
IBM is a trademark of International Business Machine.
MS™ DOS is a trademark of Microsoft Corp.
GW™ Basic is a trademark of Microsoft Corp.

OFFER LIMITED TO UP TO RETAIL PRICE OF $2950* **Sugg. Retail Price**

* Offer limited to up to dealer cost of $1999, 30 days after invoice purchase date. BIOS must be returned complete, as new, in original box with all accessories, manuals, diskettes, cables, etc. to apply. Program must be compatible with DOS 3.2 - must be original purchaser of BIOS™. Offer void where prohibited by law. Purchaser must present original invoice and proof of payment (cancelled check, charge slip, etc.) to Silent Partner Corp. to qualify. Silent Partner has the right to refuse refund for any legitimate reason.

***Individual Retailers Prices May Vary**

SILENT PARTNER

BIOS AT

7313 Grove Road, Frederick, Maryland 21701 in I that are ON (0 thru 16). It has application where individual bits are used as switches and you wish a count.

— Jim Keohane

```
1 DEF FNB1(I)=(I AND &H5555)+((I\2) AND &H5555)
2 DEF FNB2(I)=(FNB1(I) AND &H3333)+((FNB1(I)\4) AND
&H3333)
3 DEF FNB3(I)=(FNB2(I) AND &HF0F)+((FNB2(I)\16) AND
&HF0F)
4 DEF FNBB(I)=(FNB3(I) AND &HFF)+((FNB3(I)\256) AND
&HFF)
5 DEF FNBT(I)=FNBB(I AND 32767)-(I<0)
6 print fnbt(32767),     fnbt(-1),     fnbt(255),
fnbt(0)
7 REM          15          16          8          0
```

---

basic/one.liners #16, from juan (Juan Orlandini)
a comment to message 15

```
1 CIXI=0:FOR IXI=1 TO 16:CIXI=(B AND
1)+CIXI:B=INT(B/2):NEXT IXI:B=CIXI:RETURN
```

Now that is what I call a ONE-LINER. It counts the number of bits in a word. Just set B equal to the word and then call it. B will return the number of bits on. By the way, this comes directly from a Pascal program that I wrote.

|1|

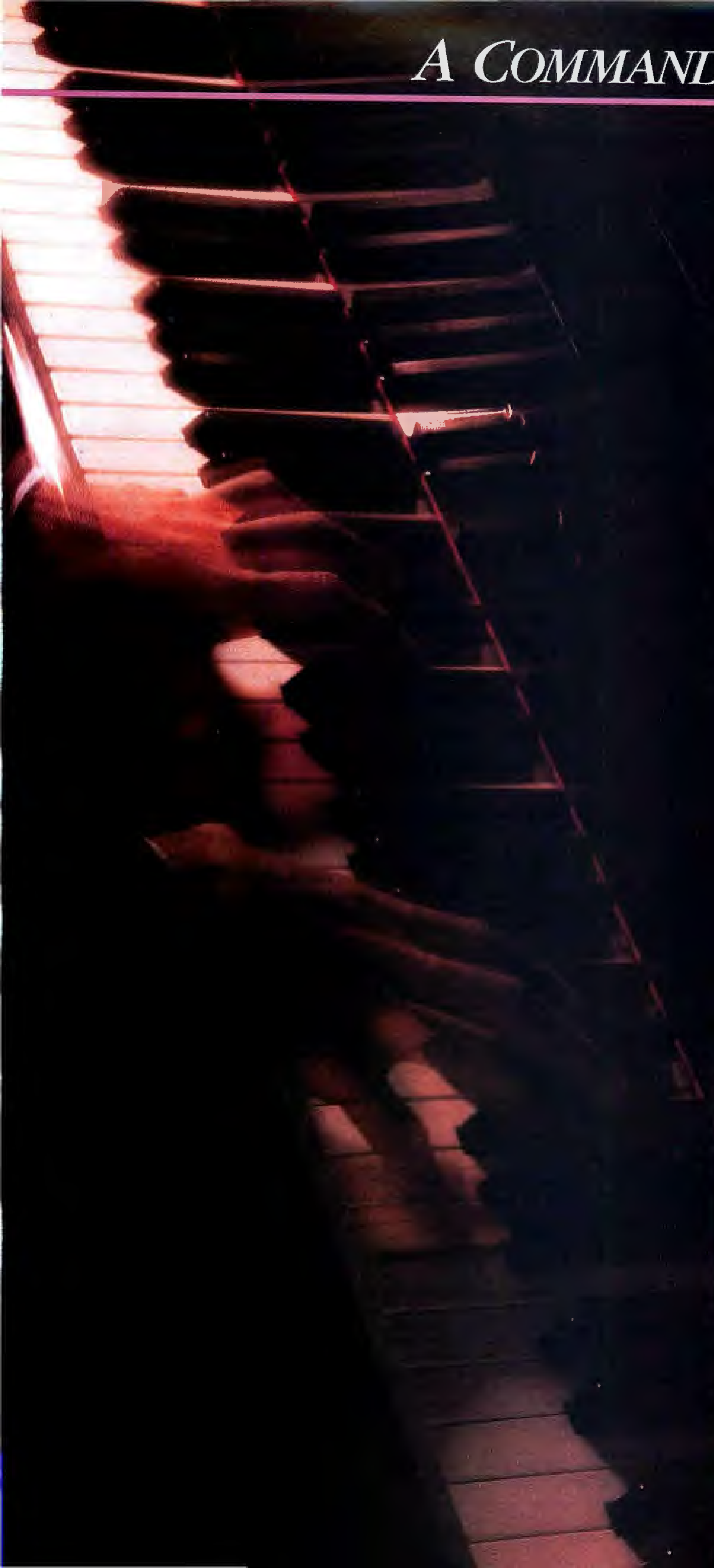
Moderator of the Osborne Conference.

---

basic/one.liners #17, from jimkeo
a comment to message 16

(continued)

A COMMAND PERFORMANCE



Your performance depends as much on your instrument as it does on your ability.

XyWrite III is a critically acclaimed word processor that extends your writing and editing range—on a simple memo or letter, a complex technical or legal document, or a multi-column script. With XyWrite III, you get nine windows, a redefinable keyboard, and many more features to support a truly fine performance.

A pianist couldn't perform if several seconds elapsed between pressing a key and hearing the note. You shouldn't wait precious seconds for your word processor to handle a command. XyWrite III processes your commands instantly, so you get more done in less time.

Enjoy the feel of a superior instrument when you sit down at the keyboard. Call (617) 275-4439 today and order our demo disk.

XyQuest Inquiry 309

XyQuest, Inc.
3 Loomis Street, Bedford, MA 01730

>Count Bits...
   Touche! Juan. I guess I got a little carried away with functions. - Jim

---

basic/one.liners #18, from btonkin
a comment to message 16

Elegant solution, Juan! All I'd change in that one would be the part that reads B=INT(B/2). Better would be B=B\2; it does the same thing!

---

basic/one.liners #19, from jimkeo
a comment to message 18

>Count Bits...
   Bruce & Juan, I did some checking and the count-bits problem was first posed in soft.eng/algorithms #47. Another cute solution. Though it can't be contained on one line due to a GOTO, it has merit in that it loops one time for every 1 bit rather than 16 times. Statement 50 can be omitted if you have overflow checking disabled. It makes use of the fact that "B=B AND (B-1)" turns OFF the rightmost bit.
   - Jim Keohane
   PS- Taken from C code posted by rfoxmich in soft.eng/algorithms #80

```
10 GOTO 80
20 K=0
30 IF B=0 THEN RETURN
40 K=K+1
50 IF B=&H8000 THEN RETURN
60 B=B AND (B-1)
70 GOTO 30
80 INPUT "Number....",B
90 GOSUB 20
100 PRINT K
```

---

basic/one.liners #20, from juan
a comment to message 18

Now that is interesting. "\" you say? OK, I'll try it. What follows is a THREE liner. Believe it or not, it is a full implementation of the Quicksort algorithm. To call it, fill array LN$() with the elements to be sorted and set N to the maximum element to go up to when sorting. On return LN$(N) will be sorted!

```
1000 S=1:STACK(S)=1:STACK(S+1)=N:WHILE
S>0:L=STACK(S):R=STACK(S+1):S=S-2:WHILE
L<R:I=L:J=R:X$=LN$((L+R)/2):WHILE I<=J:WHILE
LN$(I)<X$:I=I+1:WEND:WHILE X$<LN$(J):J=J-1:WEND:IF I<=J
THEN SWAP LN$(I),LN$(J):I=I+1:J=J-1
1010 WEND:IF I<R THEN S=S+2:STACK(S)=I:STACK(S+1)=R
1020 R=J:WEND:WEND:RETURN
```

   Enjoy.
   |1|
   PS- If LN$() is very big, then dimension STACK() out to about 20 or 30, not much more, since by that point you should write an assembly language version.
|1|

---

basic/one.liners #21, from juan
a comment to message 19

As a matter of fact, I think it can. Try this out:

```
   10 K=0:WHILE (B<>0) and (B<>&H8000):K=K+1:B=B and (B-1):WEND
```

it should work, but I don't guarantee it. I just winged that one.
|1|

---

basic/one.liners #22, from jimkeo
a comment to message 21

>WENDs...
   Hmmmmmm. Gotta brush up on my WENDs. Forgot about WHILE-WEND. - Jim
   PS- It won't work 'cause K will end up zero if B is &H8000 initially. Mebbe change "K=0" to "K=ABS(B=&H8000)" That should do it!

---

basic/one.liners #23, from juan
a comment to message 22

Question: Why do you need that &H8000 anyways?
|1|

---

basic/one.liners #24, from jimkeo
a comment to message 23

>&H8000 test
Otherwise, ya get OVERFLOW with "B-1" when B=&H8000 (-32768). Of course, you can disable OVERFLOW as I think I mentioned with the example. - Jim

MS-DOS

The activity in the MS-DOS conference reflects the fact that a high percentage of BIXen (and BYTE readers) own or use IBM PCs (or clones). This thread is centered on the very disturbing situation of a destructive program being distributed through public domain channels. The name of this program, ways to recognize other vandal programs, and possible ways to protect a machine from damage are discussed.

AN ATTACK ON THE PUBLIC DOMAIN

---

ms.dos/other #464, from dondumitru (Donald Dumitru)

TITLE: ARC513.COM is a TROJAN!
WARNING! WARNING! WARNING! WARNING! WARNING! WARNING!

   The program ARC513.COM is a TROJAN! It will alter the boot sector on your hard disk, preventing you from booting from your HD until you do a PHYSICAL FORMAT! Do *not* run this program! If you see it posted on a BBS, tell the sysop about it! I am attempting to track down the source of the copy that I have - wish me luck.

   I had warned the sysop of the board that I got ARC513.COM from, after I had heard previous warnings from BIX and FidoNews, but before I downloaded it to test it myself. He said he ran check4bomb on it, and it passed as "safe". From my (very limited) debugging of ARC513.COM, I'd say that that thing does weird things to make it hard to trace through. It is even possible that it tests whether it is being run under Debug or not.
   Donald
   PS- But it definitely does things to your hard disk. I ran it *twice*, and had to do a physical format after each run.

---

ms.dos/other #469, from dondumitru
a comment to message 467

These are the details of my experience with ARC513.COM.

   I first became aware of this program from messages posted on BIX by "wheelock," warning that he had heard rumors about ARC513.COM being a trojan.

   The next day, as I cruised past a local BBS, I noted that ARC513.COM was in the "recent uploads" section. I left a message to the sysop, warning him about the program. He left a reply thanking me for the warning, then another reply stating that he had run Check4Bomb, and that the program had passed as "safe" (so he left it posted).

   A couple of days later, in FidoNews #319, there was a warning (unsigned) that stated that ARC513.COM was

(continued)

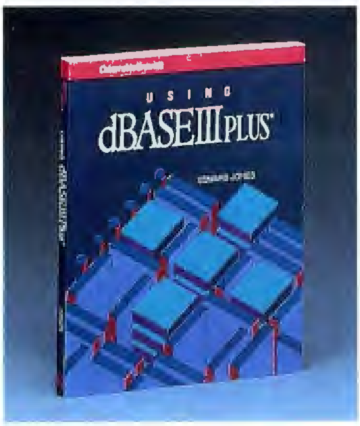
Osborne/McGraw-Hill Computer Books For IBM® PC Productivity

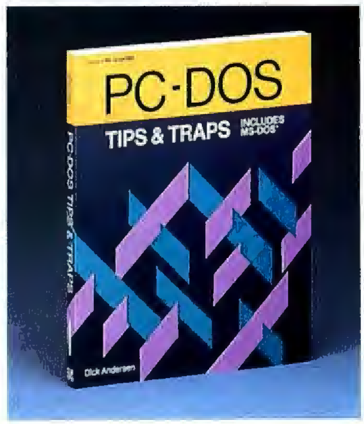
Using dBASE III PLUS™

by **Edward Jones**

You'll be in full command of all the new features of this powerful database software with Jones' expertise. Design, create, and display a dBASE III PLUS database; generate reports; use Query files; and plug into networking. A practical handbook for beginners and experienced users.

$18.95, Order #0-07-881252-6, 516 pp. 7½ x 9¼





PC-DOS Tips & Traps

by **Dick Andersen, Janice M. Gessin, Fred Warren, and Jack Rodgers**

Here's an invaluable collection of helpful tips and clever solutions to troublesome traps for everyone using PC-DOS 2.1 or MS-DOS® 2.11. From initializing your system and formatting disks, to controlling peripherals and managing the DOS environment, you'll find an array of tricks and secrets.

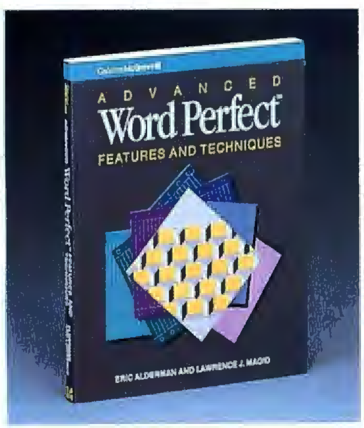
$16.95, Order #0-07-881194-5, 218 pp. 7⅜ x 9¼

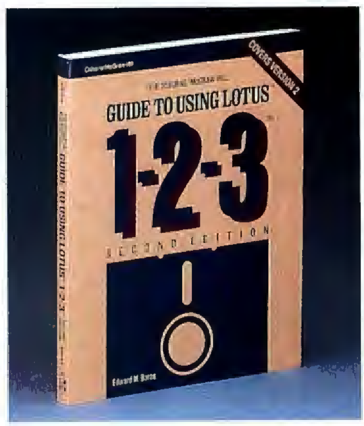
Advanced WordPerfect®: Features & Techniques

by **Eric Alderman and Lawrence J. Magid**

The source for ambitious users. Provides tools and concepts that you can use to take advantage of macros, paragraph numbering and outlining, indexing, and WordPerfect's mathematical capabilities. Covers WordPerfect integration with Lotus® 1-2-3®, dBASE III®, Sidekick®, and ProKey.™

$16.95, Order #0-07-881239-9, 310 pp. 7⅜ x 9¼





The Osborne/McGraw-Hill Guide to Using Lotus® 1-2-3®, Second Edition

by **Edward M. Baras**

Master Lotus 1-2-3 upgrade, Release 2, with this comprehensive guide. Provides detailed descriptions of worksheet, database, and graphics functions for fundamental through advanced applications. Includes ready-to-use models.

$18.95, Order #0-07-881230-5, 412 pp. 7⅜ x 9¼

◆ Osborne/McGraw-Hill books are available at bookstores and computer stores everywhere.

◆ To order by mail, complete the coupon and send it to: Osborne/McGraw-Hill
P. O. Box 400
Hightstown, NJ 08520

◆ All orders must be prepaid and should include local tax.

◆ Checks, money orders, VISA and MasterCard are acceptable for payment. No postage or handling charges are required.

◆ Books will be shipped via UPS. Allow 4-6 weeks for delivery. Books will not be delivered to post office boxes.

This order is subject to acceptance by McGraw-Hill. Offer is good only in the U.S.A.

Osborne McGraw-Hill

NAME ______
ADDRESS (No PO Box) ______
CITY ______
STATE ______ ZIP ______
INDICATE METHOD OF PAYMENT ☐ CHECK/MONEY ORDER
☐ VISA/EXP. DATE ______ ☐ MASTERCARD/EXP. DATE ______
CARD # ______
SIGNATURE ______
QTY ______ ORDER # ______ PRICE ______

Osborne/McGraw-Hill
PO Box 400, Hightstown, NJ 08520
Clip and send.
42-Q-015-7000-1

SUBTOTAL ______
TAX ______
TOTAL ______

30 Day Satisfaction Guarantee*

Not Copy-Protected

BROWN BAG Software®

PC EDITOR'S CHOICE

PC Outline!™

the Best Memory-Resident Outline Processor

Outline! is only $89.95 +$5 shipping* • CA residents add tax

COMPARE FOR YOURSELF

	Outline!	ThinkTank	Ready	MaxThink
OUTLINE FUNCTIONS:				
Outline structure	Y	Y	Y	Y
Autonumbering	Y	N	N	Y
Multi-line entries	Y	Y	N	Y
Sorting	Y	Y	?	Y
Numbering Formats	7	0	0	?
Outline Title	Y	N	N	N
Hide/Unhide	Y	Y	Y	Y
Save previous Hides	Y	N	N	?
Global Hide/Unhides	Y	N	N	N
Hide Paragraphs	Y	Y	N/A	?
Start new numbering	Y	N	N	N
Skip numbering	Y	N	N	N
Place Entry left/right	Y	N	N	N
Xfer between outlines	Y	N	N	N
EDITING FUNCTIONS:				
Full screen editing	Y	N	N	N
Margin Control	Y	N	N	N
Hanging Indents	Y	N	N	N
Justification	Y	N	N	N
Centering	Y	N	N	N
Search/Replace	Y	Y	Y	Y
Block Copy/Move	Y	Y	N	Y
Xfer between windows	Y	N	N	N
On Screen Page Breaks	Y	N	N	N
Insert Page Breaks	Y	N	N	N
Tab Settings	Y	Y	N	N
Printer fonts	Y	N	N	N
Print any part	Y	N	N	?
MISC FUNCTIONS:				
Memory Resident Option	Y	N	Y	N
Pull-Down Menus	Y	N	N	N
Windows	9	1	1	1
Zoom Windows	Y	N	N	N
Macros	Y	N	N	N
Live Time/Date stamp	Y	N	N	N
Min. Memory Consumption	85k	256k	128k	?
Import from screen	Y	N	N	N
Export to application	Y	N	Y	N
Autosave	Y	Y	N	N

To Order PC Outline! Call Now

24 hours a day, 7 days a week

800-523-0764

in California

800-323-5335

C'est la différence!

If lines are busy or calling from out of the country call

408-559-4545 TELEX: 70414 (SBS NV UD)

*call for details

BROWN BAG Software®

Brown Bag Software
2105 So. Bascom Ave.
Suite 164
Campbell, CA 95008

hacked (meaning that it was not a legitimate release of ARC), and that it was possibly a trojan.

That night, I went back by the BBS where I had seen ARC513.COM and downloaded it. After backing up my hard disk, I ran it. Surprise! It looked just like any other ARC and professed to be version 5.13, created in late April. I had it extract some library files, etc., with no problems. So I rebooted. Surprise! The system locked up after attempting to read from the hard disk, and I had to power down. I let it try to boot from the HD twice more (with duplicate results) before stuffing a floppy in to boot from. The hard disk was *still* accessible, with no apparent change. All data was available. I just couldn't boot from it. So I did a DOS (logical) FORMAT and tried to boot from the HD - lockup. I tried FDISK, FORMAT, reboot, lockup. I did a low-level format, FDISK, FORMAT, reboot, success! (I would sure *hope* so!) I ran the bugger again, and again had to resort to low-level formatting to boot from my hard disk.

My best guess is that it alters bootstrap code in the boot sector. FDISK will detect that there *is* a boot record on the HD and so doesn't rewrite the bootstrap code but just updates the partition table. FORMAT doesn't touch the boot sector at all. So you have to do a low-level format to erase the boot sector, forcing FDISK to completely rewrite it, etc. This is just a guess.

Observations: The company that produces ARC distributes .COM files - these .COM files have the executable code as well as documentation in them, and they unpack themselves when run. The executable code is *always* an .EXE file. (They have stated that executable .COM versions of ARC are hacked and thus are against the licensing agreement.) This trojan is an executable .COM file. Its approximate size is 33792 bytes (having been rounded up from XMODEM transfer). The first couple of bytes in the file are -9C 55 56 8C CD 83-. This should help identify any other copies of it that are around.

I still have a copy of it. If anybody *really* wants to have a chance at debugging through it (I don't think I have the skill to get too far), they should BIXmail me.
Donald

---

ms.dos/other #473, from dmick
a comment to message 472

I have that too, and it's short enough so I could upload ASCII .ASM source or Debug instructions. What it does is intercept INT 13 calls to disk units >3 (hard disks) and if it's a write, returns the error that means "write-protected disk in drive." You must reboot to remove, but it works for most things. I don't know if it's foolproof, though. I suppose one could always call C800: at some offset to do the same thing as an INT 13h, and bypass the resident WPTHD. Or maybe some other wizard knows if (besides that) there's another commonly-used way to write to the disks than INT 13? It seems pretty low-level and possibly what the Debug interrupts use for "w" command. I'll investigate that and report back. Meanwhile, here's the hex dump of WPTHD, which you may use to create a .COM file that will do the above.

```
0000H  EB 1A 00 00 00 00 80 FC 03 75 0C 80 FA 03 76 07
0010H  B8 00 03 F9 CA 02 00 2E FF 2E 02 01 BB 4C 00 B8
0020H  00 00 8E C0 26 8B 47 02 26 8B 1F 8E C0 8B FB BE
0030H  06 01 B9 08 00 F3 A6 75 04 B4 4C CD 21 89 1E 02
0040H  01 8C 06 04 01 BA 06 01 B8 13 25 CD 21 BA 1C 01
0050H  B1 04 D3 EA 42 B4 31 CD 21 00 00 00 00 00 00 00
```

Since I've got SideKick up, here also is the Debug unassembly. And I commented it (that's why this is the second try).

```
6237:0100 EB1A      JMP   011C          ;to install
                                        ;routine
6237:0102 0000      ADD   [BX+SI],AL    ;holds old int
                                        ;vector offset
6237:0104 0000      ADD   [BX+SI],AL    ;and segment
6237:0106 80FC03    CMP   AH,03         ;write?
6237:0109 750C      JNZ   0117          ;no, skip
6237:010B 80FA03    CMP   DL,03         ;disk unit >=3?
6237:010E 7607      JBE   0117          ;no, skip
6237:0110 B80003    MOV   AX,0300       ;wrtprot error
6237:0113 F9        STC                 ;cy=error
6237:0114 CA0200    RETF  0002          ;get rid of
                                        ;callers flags
6237:0117 2E        CS:
6237:0118 FF2E0201  JMP   FAR [0102]    ;go to previous
                                        ;int 13
6237:011C BB4C00    MOV   BX,004C       ;install - 0:4c
                                        ;is int 13 vec
6237:011F B80000    MOV   AX,0000
6237:0122 8EC0      MOV   ES,AX
6237:0124 26        ES:
6237:0125 8B4702    MOV   AX,[BX+02]
6237:0128 26        ES:
6237:0129 8B1F      MOV   BX,[BX]
6237:012B 8EC0      MOV   ES,AX
6237:012D 8BFB      MOV   DI,BX
6237:012F BE0601    MOV   SI,0106
6237:0132 B90800    MOV   CX,0008       ;check first 8
                                        ;bytes of isr
6237:0135 F3        REPZ
6237:0136 A6        CMPSB
6237:0137 7504      JNZ   013D          ;if not same,
                                        ;do install
6237:0139 B44C      MOV   AH,4C         ;same,
                                        ;terminate
6237:013B CD21      INT   21
6237:013D 891E0201  MOV   [0102],BX     ;store old
                                        ;vecadd
6237:0141 8C060401  MOV   [0104],ES
6237:0145 BA0601    MOV   DX,0106       ;install ours
6237:0148 B81325    MOV   AX,2513
6237:014B CD21      INT   21
6237:014D BA1C01    MOV   DX,011C       ;term and stay
                                        ;res
6237:0150 B104      MOV   CL,04
6237:0152 D3EA      SHR   DX,CL
6237:0154 42        INC   DX
6237:0155 B431      MOV   AH,31
6237:0157 CD21      INT   21
```

Okay? I know nobody asked, but....

---

ms.dos/other #474, from barryn
a comment to message 473

Gosh, that'll be even faster than a RAM disk for output files! <grin>

Seriously, the only other way besides INT 13 (and DOS function calls, of course, which use INT 13) to access a disk would be to write low-level code to send commands directly to the disk controller.

I have no idea how you'd even begin to protect against that kind of code.

---

ms.dos/other #475, from dondumitru
a comment to message 474

To protect against *all* attempted access to your hard disk, do what I did while I was debugging through ARC513.COM a bit more today - remove the hard disk. Of course this has the disadvantage of preventing you from *reading* from the HD as well, but in some cases...

I'm gonna type in that WPTHD program and see if it works.
Donald

(continued)

ms.dos/other #479, from petewhite (Pete White)
a comment to message 471

I'm trying to find that same program. It doesn't really "write protect," but I guess that's terminology. It gives a warning when any attempt is made to the hard disk and refuses to do so. I KNOW I have it, but on which section of my board? System (the Cul-De-Dac) is getting too big!

ms.dos/other #483, from wgivens (Wayne Givens)
a comment to message 475

A thought, Don - if that ARC513.COM is messing up the boot record (IBMBIO.COM and IBMSYS.COM?), you can unhide them and do a compare to verify/deny this. If bad, just strip all attributes, do a copy, and remark them as wo, sys, and hidden. If this works, then you know where the bugger is attacking. Otherwise....sigh.
Wayne

ms.dos/other #488, from dmick
a comment to message 483

The "boot record" is a reserved sector on the hard disk that contains partition info, which is bootable, size, etc., and is read (by the hard disk BIOS?) at bootup only. If IBMBIO and IBMSYS were the problems, one could do a SYS or something to redo. I assume Don tried that or has some other indication that the boot record was the problem.

ms.dos/other #489, from dondumitru
a comment to message 483

No! No! It's screwing up the hard disk PARTITION RECORD, which is located outside of the DOS partition (outside of *any* partition). Even Norton utilities won't read it!

ms.dos/other #490, from chrisd (Chris Dunford)
a comment to message 489

BIOS will read it. Using Debug, you can set the registers to read track 0, head 0, sector 0 and do an INT 13 to read the sector into memory someplace. You can then fix it in memory and write it back out.

ms.dos/other #491, from dmick
a comment to message 490

Sure, BIOS will (through Debug, I'd think). So will DOS interrupts....are they 25 and 26h? I never can remember.
Surprising that Norton won't. I guess I never tried it.

Actually, IBM calls that the boot record/partition table. They say a fixed-disk boot record (normally either the DOS loader or a code frag that prints the msg "Non-system disk") consists of the normal boot record PLUS a partition table. They both (being contiguous) reside in the first sector of the disk, that is, sector 0, cylinder 0, head 0. Try getting at it with Debug's L command, Don; it works 'cause I've snooped there before. The L command uses that DOS interrupt (it's 25h, absolute disk read) but makes it a lot easier to do than writing your own code fragment. It'd be interesting to see just what happened as a result of running ARC, just to get some clue as to whether it was a deliberate trojan or what.

BTW, I've seen it on several other boards since you told us all about it. Those who are members might take a look and leave a msg for the sysop if you find it.

ms.dos/other #494, from dondumitru
a comment to message 491

The DOS interrupt *won't* read the boot record - they only allow access to the DOS partition. Any sector number you pass to DOS interrupts is the logical sector in the DOS partition - not the physical sector on the disk.

The hard disk is set up as follows:
The first physical sector has the fixed-disk master boot record in it. This boot record has some bootstrap code in the front, as well as the partition table after. It is loaded into memory (by BIOS) and given control. It looks at its own boot record for a bootable partition, loads that partition's boot record into memory, and gives *it* control. This last step is the first step with diskettes. They don't load a master boot record into memory to find the bootable partition, 'cause they only have *one* partition.

Norton utilities won't read the master fixed-disk boot record (probably) because NU uses DOS interrupts for I/O (so it will work on MS-DOS machines, not just PCs).

What ARC513.COM seems to do is screw up the master fixed-disk boot record's bootstrap code. But it leaves enough of it there so that FDISK thinks it's OK and doesn't bother rewriting it - just updates the partition table at the end.

[The last paragraph is an educated guess!]
Donald

ms.dos/other #524, from dmick
a comment to message 523

Well, let's theorize together. If 1) a program has no INT 13s 2) no obvious DOS writes (INT 21 calls or INT 25 or 26h), and 3) doesn't do any IN or OUT inst. or, if it does, none to the disk board ports, then is the only other way to do it to do one of those by modifying code on the fly? One could use one of the debugging utilities that execute code like an interpreter to breakpoint mem accesses, port accesses, specific interrupts, etc., to stop any Trojan if there's no other way to write to the disk. Let me recap in a bit more organized fashion what ways I think you have to stop:

1) INT 13h (or only some...can look at subfunction in AX to check)
2) DOS calls (INT 21h, several functions...INT 26h)
3) IN's and OUT's to disk ports...I guess the important ones are the hard disk ports, since you can run floppies with write-protect tabs and if you lose 'em who cares anyway.

Is there another way to make the hard disk controller write to the disk? Or is there any other way besides writing to the hd to bung it up?

ms.dos/other #525, from dondumitru
a comment to message 524

All three of those methods are (reasonably) easy to check for in small files. In larger files, it might take a little more time to search through (but would still be worth it). The problem is large files with self-modifying code - can you imagine trying to run that 37K+ ARC513.COM through an intepreter?!! That would be a lot worse than "go out for coffee while its running." Maybe if you had an AT (or a 386-based machine) you could put the CPU in protected mode so self-modifying code wouldn't work. Then if the CPU sets a flag when self-modification is tried, you could check it for an indication. But *lots* of programs would be trapped by the protected mode (any program with data in the same segment as code...*lots*). Other ideas? ■

Technological Breakthrough

Number Crunching

THE BUYER'S MART

A Directory of Products and Services

THE BUYER'S MART is a monthly advertising section which enables readers to easily locate suppliers by product category. As a unique feature, each BUYER'S MART ad includes a Reader Service number to assist interested readers in requesting information from participating advertisers.

RATES: 1x—$475 3x—$450 6x—$425 12x—$375
Prepayment must accompany each insertion. VISA/MC Accepted.

AD FORMAT: Each ad will be designed and typeset by BYTE. Advertisers must furnish typewritten copy. Ads can include headline (23 characters maximum), descriptive text (250 characters maximum), plus company name, address and telephone number. Do not send logos or camera-ready artwork.

DEADLINE: Ad copy is due 2 months prior to issue date. For example: October issue closes on August 1. Send your copy and payment to THE BUYER'S MART, BYTE magazine, 70 Main Street, Peterborough, NH 03458. For more information call Karen Burgess at BYTE 603-924-3754.

THE BUYER'S MART

DUPLICATING SERVICES

DISK DUPLICATION
LOWEST PRICES
CALL NOW
401-624-3322
Newline Software
P.O. Box 289, Tiverton, RI 02878

Inquiry 697.

BLANKET SERVICES
Diskette duplication • Packaging • Stocking/Drop shipping • 48 hour delivery • SUPERLoK copy protection • No mastering fee • No charge for standard labels • Place a blanket order with releases as you need them for any quantity at a fixed price.
Star-Byte, Inc.
2564 Industry Lane, Norristown, PA 19403
215-539-4300 **800-243-1515**

Inquiry 715.

ENTERTAINMENT

★ BORED WITH YOUR IBM PC? ★
Turn it into a FLYING SAUCER cockpit simulator for **$2.00!** To introduce new customers to our line of fine software toys, a copy of Tommy's Saucer (**$19.95** value) will be sent to the first 50 zillion who send **$2** and ask for our free electronic catalog on 2s/2d diskette.
TOMMY'S TOYS
Box 11261, Denver, CO 80211

Inquiry 718.

FLOW CHARTS

FLOW CHARTING II HELPS YOU!
Precise flowcharting is fast and simple with Flow Charting II. Draw, edit and print perfect charts: bold and normal fonts, 24 shapes — 95 sizes; fast entry of arrows, bypasses & connectors; Fast Insert Line; shrink screen displays 200-column chart; 40 column edit screen for detail work, much more!
PATTON & PATTON
340 Lassenpark Circle, San Jose, CA 95136
1-800/672-3470, ext. 897 (CA residents)
1-800/538-8157, ext. 897 (Outside CA)
408/629-5044 (Outside the U.S.A.)

Inquiry 701.

HARDWARE

PC WHOLESALE CENTER
F.C.C. APPROVED BASIC KIT

• 640K Mother Board (OK)	$94.00
• 5150 Keyboard	$46.00
• 150W Power Supply	$56.00
• Case	$34.00

ALL FOR $230.00, CALL FOR OTHER ADD-ON CARD.
• TTL Monitor. . . . .$80.00 • AGP. $60.00
SANFORD INC.
10413 Rockley Rd., Houston, TX 77099
(713) 879-0068

Inquiry 710.

M68000-SBC
Expandable single board computer with on board 10 MHZ 68000, 20K SRAM, 32K PROM, two RS-232 serial ports, 16-bit parallel port, 5 counter/timers. Expansion for additional memory/Floppy Disk Controller board. Custom designs available. **Price starts at $89.95.**
Educational Microcomputer Systems
POB 16115, Irvine, CA 92713
(714) 854-8545

Inquiry 724.

HARDWARE

POWERLINE GREMLINS??
POWER FAILURES??
The MEIRICK STANDBY POWER SYSTEM is the TOTAL SOLUTION to your powerline problems.
400 watt system – **$449;**
800 watt system – **$795**
MEIRICK Inc., POWER SYSTEMS DIV.
Box 298, Frisco, CO 80443 **303-668-3251**

HARDWARE ADD-ONS

RS-232 RS-422

• Four Port RS-232 w/software	$349.00
• RS-422/485 Serial	$149.00
• RS-232/422 with 24 Parallel I/O	$199.00
• RS-422 Synchronous	$299.00
• EPROM/RAM	$199.00
• A/D & Digital I/O	$299.00

Call About Custom Designed Boards
Sealevel Systems Inc.
P.O. Box 1808, Easley, SC 29641
(803) 855-1581

Inquiry 711.

GM CAR DIAGNOSTICS TO PC
Connect the IBM PC to your GM car with our 85105I/F and RS232. Our BASIC gives car trouble codes and sensor data from most 1980 ½ & newer cars. Exceptions include Cadillac. **($160)** Store data on audio tape, with our 85104 cassette I/F. **($80)** Play tape to 85104 to 85105 to PC.
EXECUTIVE ELECTRONICS INC.
938 Main St., Dept. B, Yarmouthport, MA 02675
(617) 362-3694

Inquiry 669.

PATCHMAN®
ABC Serial Switching Unit
• 3:1 switching
• 8" × 4" × 2"
• RS-232 connectors
• Null Modem - "Cross-over"
ONLY **$99.95**
U S Software®
United States Software Corporation
14215 N.W. Science Park Dr., Portland, OR 97229
(503) 641-8446

Inquiry 720.

DATA ACQUISITION IN A PC!
• ICIS plug-in cards for PC and compatibles
• 16-64 A-D/frequency inputs; 0-8 D-A outputs
• 24-48 digital I/O incl. 0-24 event counts
• Data converted to engineering units; avg, std dev, max, min computed; results stored on board.
• Ideal for continuous monitoring appl.
Western Telecomputing Corporation
202 E Kagy Blvd, Bozeman, MT 59715
(406) 586-1511 or (touch tone)
1-800-423-0587, then 328 222

Inquiry 722.

MAINTENANCE MGT.

MAINTENANCE & INSPECTION
An easy-to-use computerized method to organize, operate, and control an effective preventive maintenance system. Applicable to any type of business or industry. Full screen input and displays, plus function key help command. Maintains historical data, creates work orders, and provides numerous reports. For info. write to:
Penguin Computer Consultants
P.O. Box 20485, San Jose, CA 95160
(408) 997-7703

Inquiry 704.

PROGRAM GENERATORS

WINDOW.LIB **$99.**
An easy to use complete windowing system for programmers and writing in . . .
• BASIC IBM, M-S, CB86 . . . • C LATTICE, M-S . . .
• COBOL M-S, RMF . . . • PASCAL M-S, TURBO . . .
• FORTRAN M-S, RMF . . . dBase Clipper
Window editor included. Create callable pop-up menus, help screens. The line selector features auto reverse highlighting and cursor control. No royalties. Ask about our BIOS/DOS LIB.VISA/MC.
GLENCO ENGINEERING **(312) 392-2492**
3920 Ridge Ave., Arlington Hts., IL 60004

Inquiry 730.

PUBLIC DOMAIN

THE BEST OF THE BEST!
Public Domain & User Supported Software for IBM PC or compatible. Word Processing, Database, Spreadsheets, Utilities, Games, Accounting, etc! 50 disk package **$205.00,** or rent 2 wks. **$75.00.** Info & Super Sampler Disk **$6.50,** Deluxe Word Processor **$6.50,** both **$12.00.** MC/VISA
BLUE CIRCLE GROUP, INC.
Box 23502, Minneapolis, MN 55423
(612) 823-4111

Inquiry 655.

SOFTWARE A.I.

OPS5+ for PC's & Mac's
OPS5+ by Computer*Thought is a quality implementation of OPS5 for IBM PC, XT, AT's & compatibles and the Apple Macintosh Mouse controlled, menu driven development environment **$1,750** single qty. (Substantial educational, VAR & corporate discounts). Mention BYTE & get a free mouse or **$100** off until 12/31/86.
**** Distributed by JCL Systems ****
2508 Teakwood Ln. # 100, Plano, TX 75075
(214) 867-0962 VISA, MC, MO & Checks accepted

Inquiry 660.

SOFTWARE ACCOUNTING

dBASE BUSINESS TOOLS
• General Ledger • Purch Ord/Invntory
• Accounts Recvbl. • Accounts Payable
• Order Entry • Job Costing
• Sales Analysis • Job Estimating
$99 EA. + s&h w/dBASE 2, 3 or 3+ SOURCE CODE
dATAMAR SYSTEMS™ Cred. Card-Check-COD
6809-B Convoy Court, San Diego, CA 92111
800-223-9963/CA 800-221-9964/(619) 278-8222

Inquiry 661.

► TIME & BILLING ◄
400 clients/20 partners/80 job & 40 out of pocket categories/20 areas of practice/fixed fee or hourly/more! Prints billing/statements/aged rec's/more! Free phone support. $149 (VISA/MC/AMEX). MS-DOS/CPM-80. Other original software. **FREE** catalogue.
MICRO-ART PROGRAMMERS
614 Alamenda Padre Serra, Santa Barbara, CA 93103
(805) 962-0922

Inquiry 728.

SOFTWARE/BUSINESS

SAVE LEGAL FEES
157 time tested legal forms on disk, complete with menu driven system and editing and printing capability. Free copy of PC-Write word processor included. Forms written by a noted attorney and applicable nation-wide. **$129.95 plus $7.50 shipping and handling.**
LegalEase™ HDG Software, Inc.
54 Whitney St., Sherborn, MA 01770
617-651-1881
800-628-2828 x 706 for credit card orders

Inquiry 674.

THE BUYER'S MART

SOFTWARE/BUSINESS

PC-Write™ Shareware
Fast, powerful word processor/text editor for IBM PC. Mailmerge, split screen, ASCII files, macros. Supports proportional printing, laserJet, 300+ printers. **$10** for all software, tutorial/guide on diskette. OK to copy. Register for manual/support, **$75**. Full 90-day guarantee. Version 2.7 with spell check soon: diskette pair, **$16**; register, **$89**.
Quicksoft **(206) 282-0452 Visa/MC**
219 First N. #224J, Seattle, WA 98109

Inquiry 706.

METATEXT
Letter quality from any dot matrix printer that is Epson or IBM Graphics Printer compatible. Memory resident, 10 fonts. COMPLETELY compatible with word processors, spreadsheets, languages. Approx. 1/7th normal printer speed.
ONLY **$89** VISA/MC
For more information call or write:
Image Computer Systems
P.O. Box 647, Avon, CT 06001
(203) 678-8771

Inquiry 678.

dFELLER Inventory
Business inventory programs written in modifiable dBASE source code.
dFELLER Inventory **$150.00**
Requires dBASE II or III, PC-DOS/CPM
dFELLER Plus **$200.00**
with History and Purchase Orders
Requires dBASE III or dBASE III Plus (For Stockrooms)
Feller Associates
550 CR PPA, Route 3, Ishpeming, MI 49849
(906) 486-6024

Inquiry 670.

TRANSPORTATION RESEARCH
TSA88 Transportation Simplex Algorithm (up to 510 sources, sinks or transshipment points)
TNET88 Transportation Network System (networks up to 510 nodes & 16K links)
TPR088 Transportation Problem Solver (shortest path, tours up to 50 stops)
Req. 192K, color graphics adaptor. **$99** each w/8087 support, user's guide. Write or call for our brochure.
EASTERN SOFTWARE PRODUCTS INC.
P.O. Box 15328, Alexandria, VA 22309
(703) 549-5469

Inquiry 665.

DBASE III SOURCE CODE FOR
HUMAN RESOURCES MGR. **$250**
VIDEO MOVIE MGR. **$395**
Runs on IBM PC or compatible. All modifiable. Demo diskette plus manual **$25**.
ORDER NOW!
PC MASTERS, INC.
POB 521, Lapuente, CA 91746
(818) 913-6100

Inquiry 702.

FOR YOUR EYES ONLY* $55
Did your competition underbid you again? Losing company secrets? Fight back! Password protect your files on floppies/HD. Has Corporate Security for employees and Maximum Security for office manager. Memory resident. Best Buy in Data Encryption. **$54.95.** U.S. Requires 256K IBM PC/XT/AT, DOS 2.0 or later.
GREBAR
Box 2926, Winnipeg, Manitoba, Canada R3C 4B5
(204) 942-3301

SOFTWARE/BUSINESS

OPERATIONS RESEARCH
LP88 Linear Programming (up to 510 constraints and 2510 variables). BLP88 Bounded Variables LP (up to 510 by 1510). DLP88 Dual Linear Programming (up to 1510 by 510). MILP88 Mixed-Integer LP (up to 64 integers, 255 by 1255). MZLP88 Mixed-Zero/One LP (up to 255 by 1255). Req. 192K. **$99** each w/8087 support, user's guide. Write or call for our brochure.
EASTERN SOFTWARE PRODUCTS INC.
P.O. Box 15328, Alexandria, VA 22309
(703) 549-5469

Inquiry 728.

SOFTWARE/CHURCH

ROMAR CHURCH SYSTEMS™
Membership-61 fields plus alternate address; labels, letters, reports any field(s). Offering-256 funds; optional pledge; statements; post to 255 x/year. Finance-G/L with budget; up to 500 sub-totals & 99 depts: month & YTD reports anytime for any month. Size 2000 people/floppy; 25000/10 meg. Ad too short! Write!
Romar Church Systems, Attn: BMB
P.O. Box 4211, Elkhart, IN 46514
(219) 262-2188

Inquiry 709.

Church Package
Parishioner Time, Talent and Treasure System program is written in modifiable dBASE source code.
• Contributions • Disbursements • Ledger
• Names with mailing labels
• Personal information database.
Requires dBASE II or III. PC-DOS/CPM-80 $200.
Feller Associates
550 CR PPA, Route 3, Ishpeming, MI 49849
(906) 486-6024

Inquiry 729.

RECORD OF CONTRIBUTIONS
An in depth system to account for contributions to the church - and more. Easy to enter/update/retrieve data. Prepares Reports/Statements on demand. For the IBM PC/XT or AT or compatibles. Excellent Documentation. **$99.00 + $5.00** S&H.
MICRO-DYNAMICS OF TEXAS
P.O. Box 40691, Houston, TX 77240
(713) 896-9957

Inquiry 689.

SOFTWARE/EDUCATION

KID HAS MATH TROUBLE?
Our exciting "EDUCATIONAL GAMES" could make your kid **THE TOP MATH STUDENT** in his class. Send CHK/MO, specify **IBM PC** or **APPLE II/C. $21.50** ea. + **$2** S/H.
BLACK BELT-ADDITION -SUBTRACTION
-DIVISION -MULTIPLICATION
ROBIN SOFTWARE
P.O. Box 1053, Cupertino, CA 95015
(408) 255-7456

Inquiry 708.

SOFTWARE/GAMES

PC/VEGAS II
Poker, Blackjack, Roulette, Keno, Slots—**$26.** Above plus Baccarat and Craps—**$39.** Exact Vegas simulation, includes Help, best bet/odds windows. Cross-game scoring. Graphics not mandatory. Color or Mono. MS/PC DOS, 128K. C Source available, ask. Add **$2** for shipping/CA res. add 6½% tax.
D & H BEAR SOFTWARE
P.O. Box 10793, Marina Del Rey, CA 90295
(213) 374-0358

Inquiry 660.

SOFTWARE/GENERAL

NEW INTEGRATED ENVIRONMENT
AI FOR THE IBM PC
TOPSI IS A FULL VERSION OF **OPS5** WHICH RUNS UNDER MS-DOS, UNIX OR CP/M. A FAST, EFFICIENT EXPERT SYSTEM DEVELOPMENT TOOL.
PROTOTYPING: **$125** PRODUCTION: **$250**
PROFESSIONAL: **$375** SHIPPING, ADD **$5**
DYNAMIC MASTER SYSTEMS
POB 566456, Atlanta, GA 30356
(404) 565-0771 Telex #282923

Inquiry 664.

Label Printing Made Easy
LABLMAKR prints on label or card stock up to 8" wide. 52 standard formats, thousands of variations - including audio/video cassette labels. Works with PC-compatible and dot matrix printers. Only **$49** - money back if not delighted.
ETS Center
35026-A, S. Turtle Trail, P.O. Box 651
Willoughby, OH 44094
(216) 946-8479

Inquiry 668.

NEW Turbo Pascal™ Programs
Outstanding software. Full source code included! KEYMASTER, DESKMASTER: Pop Up desk organizers. Editor, macros, alarms, addr book/dialer, calculator, more! **$49.95** + **$2** s&h each. IBM PC & Comp. DICTIONARY TOOLBOX: Library to produce/manage bit compressed dict. **$39.95.** Any MS-DOS Comp. CHESTER: Full featured Chinese checkers game using heuristic search. **$22.00** IBM-PC & Comp. Visa/MC
A. I. SOFTWARE 512-928-2009
2200 Rogge Ln., Austin, TX 78723

Inquiry 650.

PERSONAL FINANCE STATEMENT
Spreadsheet for Lotus, Symphony or Multiplan on IBM PC's or compatibles. Produces a Personal Finance Statement in a standard form acceptable by all banks and lending institutions. Never fill out another personal statement. Easy to update. Keeps a permanent record of your assets and liabilities. **30-day money-back guarantee. $29.95. MC/VISA.**
NavStar
P.O. Box 1763, Rockville, MD 20850
800-835-2246 x230 In KS 800-362-2421 x230

Inquiry 695.

SOFTWARE/GRAPHICS

GRAPHICS PRINTER SUPPORT
AT LAST! Use the PrtSc key to make quality scaled B&W or color reproductions of your display on any dot matrix, inkjet, or laser printer. GRAFPLUS supports all versions of PC or MS-DOS with IBM, Tecmar, and Hercules graphics boards. $59.95.
Jewell Technologies, Inc.
4302 SW Alaska St., Suite 207, Seattle, WA 98116
(206) 937-1081

Inquiry 681.

FORTRAN PROGRAMMER?
Now you can call 2-D and 3-D graphics routines within your FORTRAN program.
GRAFMATIC: 75 callable routines for screen output. **$135.**
PLOTMATIC: Pen plotter driver. **$135.**
For the IBM PC, XT, AT and compatibles. We support a variety of compilers, graphics boards and plotters.
MICROCOMPATIBLES
301 Prelude Drive, Dept. B
Silver Spring, MD 20901
(301) 593-0683

Inquiry 692.

THE BUYER'S MART

SOFTWARE/GRAPHICS

SCIENTIFIC GRAPHS

SCI-GRAF produces high-res graphs (1680 x 1712 pixels) on Epson or IBM graphics printers. Supports log/linear scales, error bars, overlays, batch-mode operation. Very flexible data entry from disk or keyboard. User customizable. Requires DOS 2 or 3, 256K.
$99.95 (Sorry, no credit cards)

Microcomputer Systems Consultants
32 West Anapamu, Suite 190, Santa Barbara, CA 93101
(805) 963-3412

Inquiry 693.

SOFTWARE/LANGUAGES

Minnesota SNOBOL4 *Language*

Powerful string & data handling facilities. Interpreter compatible with mainframe SNOBOL4. 32K strings, 32 bit integers, 8087 for float & large memory model. Sample pgms include ELIZA. For >128K IBM PC & DOS or equivalent. Definitive "green" book by Griswold et al available.

Guide + 5¼" SSDD diskette $44.95
Guide + diskette + "green" book $59.95
"Green" book only $26.95

Postpaid in USA. In NY add tax. VISA/MC (914) 271-5855

BERSTIS INTERNATIONAL
POB 441, Millwood, NY 10546

Inquiry 654.

SOFTWARE/SCIENTIFIC

ELECTRONIC CIRCUIT ANALYSIS

Has AC, DC, Transient and Fourier. Includes worst-case, Monte-Carlo, component sweeping. Macro models, transmission lines.

Electronic Circuit Analysis .... **$450.**
EC-Ace .... **$95.**

TATUM LABS
POB 698, Sandy Hook, CT 06482
(203) 264-3755

Inquiry 717.

PC-KEY-DRAW TRY IT $15 VER. 3.1

Comprehensive, high-power graphics editor, CAD, paint, slide show program. Over 150 standard functions plus macros, 64 pages/layers, zoom printer, 64 patterns, novice/expert modes. $15 includes manual, software, quick ref. card. Register $100 for support, updates complete printed manuals. Shareware. IBM PC and compatibles.

OEDWARE
P.O. Box 595, Columbia, MD 21045-0595
(301) 997-9333

Inquiry 699.

SOFTWARE/PRINTER

HP LASERJET

LASER FONTS works with *your* font cartridges to make near typeset printing a snap. Simply embed {xxx} into your word processor spreadsheet, etc. Only **$39 + $3** s/h.

- 245 type styles
- Site licences
- Memory-resident
- Network version
- IBM PC DOS 2.0+
- Guarantee

Keller Software
1825 Westcliff Dr., Newport Beach, CA
714/642-5544

Inquiry 683.

FAST FOURIER TRANSFORMS

Best available! Extremely fast and accurate. Two assembly language subroutines: one for integer data, one for floating point (8087 & 80287 supported). Forward transforms, inverse, power spectra, special modes for real-valued data. Callable from MS-FORTRAN. Excellent documentation. **$149** each, or **$249** for both.

J. W. HARTWELL & ASSOCIATES
Route 4, Box 1540, Hillsborough, NC 27278
(919) 732-7951

Inquiry 682.

PRESENTATION GRAPHICS

Executive Picture Show - complete presentation graphics! IDEAL for demos, tutorials, hardcopy, slides. SLIDE-SHOWS, WORD CHARTS, BUSINESS GRAPHICS, FREE-FORM PAINTING & DRAWING and GRAPHICS LIBRARIES. SUPPORTS dot matrix, inkjet, laser printers, and plotters. Complete package **$245** - Trial version **$15.**

PCsoftware
11627 Calamar Court, San Diego, CA 92124
(619) 571-0981

Inquiry 703.

SOFTWARE/SCIENTIFIC

HP-PC HYPER-CALCULATOR

Pop-up MS-DOS programmable scientific calculator emulates HP-11C. Hex/oct/bin arithmetic, two-way data transfer, program & data disk storage, HELP, alpha prompts, 100 registers, 1000 program steps. Free 8087 version, utility programs. **$39.95 + $1 s&h.**

SUNDERLAND SOFTWARE ASSOCIATES
P.O. BOX 7000-64
REDONDO BEACH, CA 90277
CALL TOLL FREE **800-628-2828 ext. 502**

Inquiry 718.

forMath® text-formatter

- Equations, matrices, ratios, integrals, diagrams
- Macros, fonts, Greek/math symbols
- Hyphenation, secn/eqn/ref numbering
- Indexes, table of contents, footnotes
- Dot-matrix, daisywheel, laser printers, all monitors

$400. $50 for demo

SHANTHA SOFTWARE INC.
50 West 97th St. Room 11N, New York City 10025
(212) 222-SNIP
Touchtone toll free: **950-1088-wait-FORMATH**

Inquiry 712.

SOFTWARE/LANGUAGES

SIMULATION LANGUAGE

TurboSim allows large discrete event simulations i.e., manufacturing systems, health-care systems, etc. to be run on a micro. It features a comprehensive final report, histograms, plots and multiple simulation runs in an unattended mode. Source code is written in Turbo Pascal and is included. Borland International's Turbo Pascal and an IBM/PC with 64K are required. $49.95.

Micro Simulation
37 William J. Heights, Framingham, MA 01701
(617) 875-6098

Inquiry 691.

UPGRADE YOUR IBM PC TO A STORAGE OSCILLOSCOPE!

Do Data Acquisition, Frequency Spectrum Analysis, Transfer Functions, Analysis with Lotus 1-2-3, and more—inexpensively and all without programming!

For Info on SNAPSHOT Storage Scope:

HEM Data Corporation
17025 Crescent • Southfield, MI 48076
(313) 559-5607

Inquiry 675.

► SCIENTIFIC CALCULATOR ◄

Memory resident emulation of the HP-11C plus direct insertion of results in foreground program, graphically presented, decimal, hex, binary, & octal operations, boolean functions and more. Mouse support. 8087 or Standard versions $49.95 + 3.00 s&h.

K SOFTWARE HOUSE
Rt. 2, Box 83B1 Unionville, TN 37180
(615) 294-5090

Inquiry 725.

SCREEN PROGRAM GENERATOR

SPG I creates BASIC program code quickly and easily. Design your screen directly and SPG I will generate the BASIC program code to handle the display and input of data. Controls use of color, editing, graphics, cursor movement and function keys. Generated code easy to modify / merge.
IBM PC & compatibles 128K.
$75 + $3.00 S&H (CA+tax)

Treebeard Software
5901 Warner Avenue, Suite 428
Huntington Beach, CA 92649
(714) 840-6939

Inquiry 619.

DIGITAL SIGNAL ANALYSIS

Fourier Perspective II w/graphics. FFT/IFT, convolution, deconvolution, Hanning, Power Spectrum, Cross Correlation, 25 more. Graphics 2D and 3D. 8087 recommended, not required. Easy to use menus. FP II $395 or FP I (FFT w/graphics only) $149. Runs on PC, PC-XT, PC-AT.

Alligator Transforms Scientific Software
P.O. Box 11386, Costa Mesa, CA 92627
(714) 662-0660

Inquiry 651.

CHARACTER CUSTOMIZATION

CHARGEN! 2.1 lets you modify the character set on the EGA. Useful for scientific/engineering symbols on your word processor, special symbols in your programs, and more. Requires DOS 2.x or 3.x, Enhanced Graphics Adapter. **$35 + $2** s/h (MN add 6%).

DK Micro Consultants
P.O. Box 8714, Minneapolis, MN 55406
(612) 722-0931

Inquiry 663.

MUMPS

DTM-PC, fastest ANSI Standard MUMPS for IBM PC XT, AT or compatibles. VA File Manager available. Runs up to 16 crts from one AT.

DataTree, Inc.
318 Bear Hill Rd., Waltham, MA 02154
(617) 890-1620

Inquiry 662.

CROSS ASSEMBLERS for VAX VMS and PC/MS DOS

New Low Prices

Relocatable Macro Cross Assemblers, Linkers, Librarians
Targeted to almost all Microprocessors

ENERTEC, INC.
BOX 1312, Lansdale, PA 19446
215-362-0966 **MC/VISA**

Inquiry 667.

► MATRIX 100 ◄

Perform multiple regression, solve simultaneous equations, invert matrices, etc. in BASIC or Fortran. ". . . very sound numerically . . . very powerful and very easy to use" — **OR/MS Today** 2/85. "Stanford Business Software deserves congratulations" — **PC Mag.** 5/14/85. Price **$80**; 8087 support **$125**; compiler support **$250**; Fortran library **$175**. (**$4** s&h).

STANFORD BUSINESS SOFTWARE, INC.
2672 Bayshore Parkway, Ste. 304
Mountain View, CA 94043
To Order Call **(415) 424-9499**

Inquiry 714.

THE BUYER'S MART

SAVE! — SAVE! — SAVE!

MEMORIES—BOARDS—PRINTERS

- ★ **64K/128K/256K D-RAMs at LOW PRICES**
- ★ **V20-5 or -8...$14.00 / V30-8...16.00**
- ★ **8087....$109.00 / 8087-2....$155.00**
  80287-6.$179.00 / 80287-8...$285.00
- ★ **Memory Boards and Modules**
  - JRAM-2 Parallel or Serial w/ØK...$149.25
  - JRAM-3 /JRAM-AT Par. or Ser. w/ØK...$201.75
  - JRAM AT-3 Par. or Ser. w/ØK...$261.75
  - Ser/Par or Ser/Ser Clock module.......$150.00
  - Ser/Par or Ser/Ser module............$105.00
  - Clock/Calendar module..................$60.00
  - JDISKETTE module....................$150.00
  - JLASER-1 w/ Canon switch & cable.....$300.00
- ★ **BROTHER Printers & Accessories**
  - M-1509 Dot Matrix Printer...........$380.00
  - HR-15XL Daisy Wheel Printer.........$365.00
  - Twinwriter 5 Dot Matrix/Daisy.........$880.00
  - Printer Accessories.............25% off LIST

* WARRANTY: All I.C.s guaranteed for 180 days. * OEM and DEALER discounts available. * Prepaid orders — No shipping charges * Maryland residents must add sales tax.

Advanced Technology Products, Corp.
P.O. Box 2205 / 11141 Georgia Ave., Silver Spring, MD 20902
Phone: 301-933-3523

Inquiry 5

64K ■ 128K ■ 256K DRAMS

EPROMS

80287-10 ■ 80287-8 ■ 80287-3
8087-3 ■ 8087-2 ■ 8087-1
4416 ■ 4464
V-20 ■ V-30

BITTNER ELECTRONICS

MasterCard VISA

(714) 497-6200
CALL NOW FOR FREE CATALOG

Inquiry 35

NEW AN I/O™ Series
Analog Interfaces for your PC/APPLE

Versatile Accurate and Fast

MODELS

AP-8	• 8-bit A/D and D/A to 50Khz. amplifier. anti-aliasing filters. etc.S299.
AP-12	• 12-bit A/D and D/A to 40Khz. analog multiplexer. amplifier. S/H. 8-bit I/O. clock. etc. S450
PC-12	• 12-bit A/D or D/A to 80Khz. analog multiplexer. amplifier. anti-aliasing filter. S/H. on board VCO function generator. 8-bit I/O. etc. S750.
PC-DMA	• Our best available analog interface. 12-bit A/D or D/A to 100Khz. DMA driven to allow real-time processing on board timer for zero timing jetter. input and output filters, amplifier. S/H. multiplexer 500Hz to 100Khz phase-locked (to timer) function generator. fully program mable. S1250.

PC-model designation refers to IBM PC compatible boards. AP for Apple II. Prices include software. Peripheral multi-channel input signal conditioner and filter. and PC-based FFT board available soon. For purchase or info please contact:

CANETICS. Inc. PO Box 70549. Pasadena. Ca 91107
Telephone (818) 584-0438

Inquiry 53

DATA ACQUISITION TO GO
INTERFACE FOR ANY COMPUTER

FREE IBM SOFTWARE

Connects via RS-232. Fully IBM compatible. Built-in BASIC. Stand alone capability. Expandable. Battery Option. Basic system: 16 ch. 12 bit A/D, 2 ch. D/A, 32 bit Digital I/O. Expansion boards available. Direct Bus units for many computers.

(201) 299-1615
P.O. Box 246, Morris Plains, NJ 07950

ELEXOR

Inquiry 91

Why waste time?

ACCELERATE with VCACHE™

HARD DISK ACCELERATOR

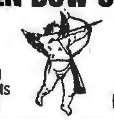
- Caching eliminates repetitive disk accesses
- Use up to 15 Mb of extended/ expanded or 500 Kb of standard memory

DISKETTE ACCELERATOR
SCREEN ACCELERATOR

FAST — FRIENDLY — SAFE

VCACHE
GOLDEN BOW SYSTEMS

$65
Add $3 for shipping/handling
California residents add 6% sales tax



2870 Fifth Avenue
Suite 201
San Diego, CA 92103
619/298-9349

Inquiry 114

Get the whole story on graphics terminal emulation.

To find out more about software that lets your PC emulate TEKTRONIX™ 4105/6/7/9 and DEC VT100™ terminals, call or write:

GRAFPOINT
4340 Stevens Creeks Blvd., Suite 280, San Jose, CA 95129 (408) 249-7951

Inquiry 117

DYNAMIC RAMS

41256	100ns	$4.85
41256	120ns	$3.25
✓41256	150ns	$2.90
4164	120ns	$1.40
✓4164	150ns	$1.10
4464	150ns	$3.95

■ PROCESSORS

8087-2	8mHz	$159.00
8087-3	5mHz	$115.00
80287-3	6mHz	$179.00
80287-8	8mHz	$ CALL
V-20	8mHz	$ 12.75
V-30	8mHz	$ 14.00

■ STATIC RAMS

6264LP-15	150ns	$2.95

■ E. PROMS

27C256	250ns	$6.50
27256	250ns	$5.25
27128	250ns	$3.95
27C64	200ns	$4.50
2764	250ns	$3.25
2732A	250ns	$3.75

■ 8000's
(Parts in stock)

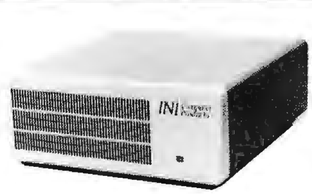
I.C. EXPRESS

15358 Valley Blvd., City of Industry, CA 91746
Phone: 818-369-2688 (Mon-Fri • 8-5)

ORDER TOLL FREE
(800) 892-8889 Outside California • (800) 882-8181 Inside California

CALL FOR CURRENT PRICES & VOLUME DISCOUNTS.
Price Shown for Cash • MasterCard/VISA add 3% more
Prices are subject to change. Minimum order $10.00
California residents must add 6.5% sales tax
Shipping & Handling: UPS Ground $5.00, UPS Air $7.00 (under 1 lb.)
ALL MERCHANDISE IS 100% GUARANTEED.

Inquiry 137



INI-10/AT
PC-AT EXPANSION CHASSIS

- Full 16-bit PC-AT bus
- 10 slots
- 250W power supply
- Two fans
- Direct extension of PC-AT bus
- No software changes needed
- Inexpensive, simple, attractive

Dealer Inquires Invited.
List Price $1,400.00

INI Computer Products
6915 Hightech Drive
Midvale, UT 84047

(801)561-1100
Contact Mark Jenkins

Inquiry 140

GAT-286
COMPLETE SYSTEM—$2195.00

Includes one megabyte memory on board, 6 or 8 MHz, zero wait state, Phoenix BIOS; Disk drive controller with 2 floppy disk drives (1.2 M.B. and 360 K) and one 20 M.B. hard disk; monochrome monitor with adapter, serial/parallel ports; 200 W. power supply, keyboard, case. IBM PC/AT compatible. Certified to comply with FCC class B standards.

PC/XT/AT COMPATIBLE PARTS

Seagate 30 M.B., High Speed Hard Disk	$645.00
20 M.B. Hard Disk Kit for PC/XT	$395.00
30 M.B. Hard Disk Kit for PC/XT	$465.00
135 W. Power Supply	$ 65.00
I/O Card for XT or AT	$ 75.00
LOGIMOUSE	$85.00
8087-3 Co-Processor	$120.00
80287-3 Co-Processor	$215.00
XT Compatible System w/640 K, all ports & clock, 2 drives, monochrome monitor	$825.00

LUCKY COMPUTERS
(214) 690-6110
1701 Greenville, Suite 602, Richardson, TX 75081

IBM PC/XT/AT are Registered Trademarks of International Business Machines Corp.

Inquiry 171

Inquiry 104 for End-Users. Inquiry 105 for DEALERS ONLY.

GEM

MAXIMUM PERFORMANCE.

GEM-ST
Don't be fooled. No hidden cost!

In keeping with industry trends MSO is bringing our customers high performance P.C. compatibles and accessories. The GEM-ST™ features the V20-8 chip which runs at three times the speed of the IBM-PC XT* and also runs CPM 8080 software.

FREE 5 module intergrated software

$1199

22 meg sub-system with tape back-up $599 w/purchase

GEM-ST Specifications:

CPU - V20-8 Chip & Run CPM Software
RAM - 640K
Clock Rate - 4.77 or 8 MHZ
Disk Interface and Drives - (2) - 5-1/4" floppy drives 360K
ROM BIOS - Legal BIOS
Keyboard - AT Style Keyboard
Expansion Slots - 8 IBM* compatible slots
Power Supply - DC Voltage and Max Current
+ 5V @ 15A - - 5V @ .5A
+ 12V @ 6A - 12V @ .5A
Imput Voltage 115/230V
Meets UL/FCC Standards
Standard Interface - (1) - Serial RS-232-Port
(1) - Game Port
(1) - Parallel Printer Port
Clock - Clock calendar battery backed up
Video - TTL Hi-Res Monographics Card and TTL Monitor
Operating System - MS-DOS
Warranty: 1 year all parts. 90 day labor.

GEM-AT Specifications:

CPU - 80286
RAM - 1 Meg
Clock Rate - 8 MHZ
Disk Interface and Drives - (1) - 5-1/4" floppy drive 1.2 Meg
ROM BIOS - Legal BIOS
Keyboard - AT Style Keyboard
Expansion Slots - 8 IBM* compatible slots
Power Supply - 200 Watt
Meets UL/FCC Standards
Standard Interface - (1) - Serial RS-232-Port
(1) - Parallel Printer Port
Clock - Clock calendar battery backed up
Video - TTL Hi-Res Monographics Card and TTL Monitor
Operating System - MS-DOS
Warranty: 1 year all parts. 90 day labor.

EDUCATIONAL DISCOUNTS

At Micro Supply Organization we offer the lowest prices on GEM computers. With prices like these you can afford the convenience of owning and operating more than one computer.

We also offer the User Support Hotline for questions concerning GEM computers. Whether you need one or a dozen computers, Micro Supply Organization is the place to get them.

GEM-AT
Performance for the power user.

The GEM-AT™ runs eight times as fast as an IBM-XT™, for your multi-tasking and multi-user needs.

Including MS-DOS 3.2 with G.W. Basic. The GEM-AT™ is a complete turnkey system with everything to plug and go!

FREE 5 module intergrated software

$1899

32 meg sub system with tape backup $799 w/purchase

MSO
MICRO SUPPLY
ORGANIZATION INC

For our catalog with complete details and prices, send $2 to:
Micro Supply Organization, Inc.
4909 Stockdale Hwy #180
Bakersfield, CA 93309
15% Restocking on Returned Orders

MADE IN AMERICA

805/393-2247

Inquiry 115

Inquiry 217

Inquiry 197

Inquiry 165

Inquiry 225

a message to our subscribers

From time to time we make the BYTE subscriber list available to other companies who wish to send our subscribers material about their products. We take great care to screen these companies, choosing only those who are reputable, and whose products, services, or information we feel would be of interest to you. Direct mail is an efficient medium for presenting the latest personal computer goods and services to our subscribers.

Many BYTE subscribers appreciate this controlled use of our mailing list, and look forward to finding information of interest to them in the mail. Used are our subscribers' names and addresses only (no other information we may have is ever given).

While we believe the distribution of this information is of benefit to our subscribers, we firmly respect the wishes of any subscriber who does not want to receive such promotional literature. Should you wish to restrict the use of your name, simply send your request to the following address.

BYTE Magazine
Attn: Circulation Department
One Phoenix Mill Lane
Peterborough NH
03458

Inquiry 2

NEW DEALER SPECIAL

XT—COMPUTER

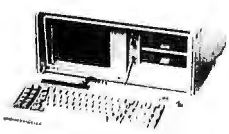
	QTY 2
640K MOTHER BOARD W/OK	89
256K RAM	27
135 WATT POWER SUPPLY	53
FLOPPY DISK CONTROLLER	27
FLIP TOP BOX	38
XT/AT KEYBOARD	59
MONO/GRAPHICS/PRN (HERC)	54
TTL MONITOR W/SWIVEL	96
(1) 360K FLOPPY	92
PARTS	$535
ASSEMBLY & TEST	64
	$599
20 MEG HARD DISK W/CONTROLLER	425

AT COMPUTER

	QTY 2
80286 MOTHER BOARD W 512K RAM	595
200 W POWER SUPPLY	98
AT BOX W/KEY LOCK	79
SERIAL & PARALLEL CARD	65
AD KEYBOARD	59
1.2 M FLOPPY	129
FDD & HDD CONTROLLER CARD	225
PARTS	$1,250
ASSEMBLY & TEST	150
	$1,400
30 MEG 40 MSEC HARD DRIVE	615
EGA CARD	295
EGA MONITOR	449

XT PORTABLE $995

640K MOTHER BOARD
640K RAM
MULTI I/O CARD
CLOCK, CALENDAR,
SERIAL,
PARALLEL
TWO 360K
FLOPPY DRIVES
MONO/GRAPHICS/PRN (HERC)
DETACHED KEYBOARD
135 W POWER SUPPLY
PORTABLE CASE
9" MONITOR, GREEN or AMBER



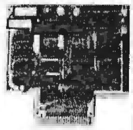
ADD ON BOARDS — PARTIAL LIST

XT-640K TURBO 4.77/8 MHZ W/OK	99
LOCAL AREA NET CARD W/SOFTWARE	249
384K MULTI-FUNCTION (AST 6 COMP) OK	69
MODEM CARD 300/1200 (HAYES COMP)	149
DISK I/O 2FO CTRL, PRN, SER, CLOCK, CAL, GAME	70
AB SWITCH BOX W/64K BUFFER (1:2 or 2:1)	94
AB SWITCH BOX (1:2 or 2:1)	29
COLOR/GRAPHICS/PRINTER CARD RGB	58

ACS IMPORTERS
5311 DERRY AVE., UNIT A
AGOURA HILLS, CA 91301
HOURS 9 AM - 6 PM PST
(818) 889-1092
TELEX: 299353 POST UR

PC DIADAC «» TAKE CONTROL WITH YOUR IBM PC/CLONE
16CHANNEL 12BIT ANALOG INPUT
UP TO 4 CHANNEL ANALOG OUTPUT
$375.00 DIGITAL I/O 24 LINES

DIADACS 1 «» TAKE CONTROL WITH YOUR COMMODORE 64
16 CHANNEL 12 BIT
A/D CONVERTER
D/A CONVERTER 12 BIT
DIGITAL OUTPUT 10LINE
DIGITAL INPUT 12LINE
REAL TIME CLOCK



Comes complete with user's manual and software driver........ $279

CONTROL MODE 1 DATA ACQUISITION SOFTWARE for C64 150

DEALERS/REPRESENTATIVES PLEASE CALL
CORE MICROTECH P.O. Box 102 • LANGEHORNE, PA 19047 • (215) 757-0284

Inquiry 50

MYSTIC PASCAL
640K CODE & DATA

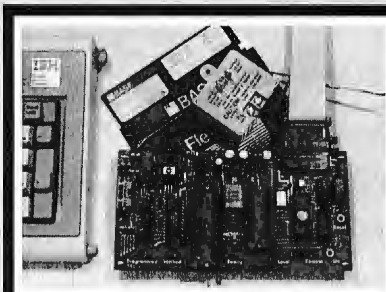
- complete programming environment with editor
- first incremental compiler for Pascal
- 8087 & software math
- true MULTI-TASKING support
- spectacular graphics on 20 screens with pan & zoom
- creates linkable OBJect files
- open 8 to 5 Mtn. time, Mon thru Sat

Disk with condensed manual to print out $16—full printed manual $16—both $32. For IBM PC 320K. Visa/MC/COD—add $2 shipping, COD $3. Overseas add $10 for disk & manual or $5 for disk only.

Mystic Canyon Software
P.O. Box 1010
Pecos, NM 87552 505/757-6344

NEW VERSION 1.6— STILL ONLY $16!!!

Inquiry 202



MOTOROLA 6805 SINGLE CHIP MICROCOMPUTER DEVELOPMENT SYSTEMS

Two systems from TEC allow the IBM PC/XT/AT to be used as a complete development system for the Motorola 6805 series single chip microcomputers. Model MCPM-1 supports the MC68705P3,P5,U3,U5, R3, & R5 chips. Model MCPM-2 supports the MC1468705F2 & G2 cmos versions. Both systems are priced at $495 and include a cross assembler program, a Simulator/Debugger program and a programming circuit board with driver software. One serial port required.

TEC
PO Box 53, West Glover, Vt. 05875
(802) 525-3458

Inquiry 278

MEMORY

Virtual

MicroVMS gives you 4 MBytes of LIM Expanded Memory WITHOUT BOARDS. Uses hard disk or Advantage! RAM for 1-2-3 Rel 2, Symphony 1.1, Reflex 1.1, or SuperCalc 4 data storage. Only $75.

Conventional

XTend puts 640K on the system board of your IBM XT, Compaq Portable or Plus. From $24.50 to $119.00.

Expanded

Breakthru/PC adds 2 MBytes of LIM expanded memory to your PC, XT, AT or compatible. Check our specs before buying Above Board or RAMpage! From $150 (no RAM) to $450 (2 Mbytes).

VERICOMP
8825 Aero Drive (Suite 210)
San Diego, CA 92123
VISA 619-277-0400

Inquiry 294

Beat the limits!

Vfeature™ DELUXE

Software for hard disks

- DOS partitions to 1 GIGABYTE
- Spans two drives in one bootable partition
- Supports big drives on AT and XT
- Secures data

GOLDEN BOW SYSTEMS

$120
$3 shipping/handling
California orders add 6%

2870 Fifth Avenue
Suite 201
San Diego, CA 92103
619/298-9349

Inquiry 116

Money Counts™

COMPLETE MONEY MANAGEMENT FOR HOME AND BUSINESS.

• Checking, Savings and Credit Card Mgmt. • Smart Account Reconciler With Automatic Error Detector • Budgeting • CPA Designed • Financial Statements • Graphics • On-Line Help Screens • High Speed Financial Data Base • Inquiry Reports • Pop-Up Calculator • Tutorial • Not copy protected • Simplifies Tax Preparation • Much More.

Only $69.95
Visa • MC • C.O.D. Free Brochure
319/373-0197

PARSONS TECHNOLOGY
6925 Surrey Drive NE
Cedar Rapids, IA 52402

Inquiry 223

HUNTING FOR A GREAT DEAL?

HARD DISK DRIVE CARDS

Logic Array MegaCard!

- Hard disk drive on a plug-in card gives PC or compatible the storage capacity of an XT
- Uses NEC technology for high reliability
- Fast, easy installation
- **two year guarantee**—parts and labor

21-Megabyte . . . . CALL FOR PRICE
30-Megabyte . . . . CALL FOR PRICE

HARD DISK DRIVE KITS

(Half height, boots from hard disk—includes controller, cables, manual)

20-Megabyte $415
40-Megabyte $599
(with Disk Manager software)
On Track Disk Manager Software sold separately $65

BOARD LEVEL PRODUCTS FROM MYLEX

Multifunction card
0-K $82
384-K $123
Envision I (mono/color/graphics) $147
Envision II (mono/color/graphics, char. RAM enhanced modes) $156
Color Adapter I $77
Color Adapter II $83
Business Master I (mon/par adapter, enhanced text modes) $80
Business Master II (high-res. mon graphics, enhanced text modes) $91
Enhanced Graphics Adapter $279
(EGA, CGA, MDA, HGC compatible)
Advanced Graphics Adapter I $345
(high-res. 800 x 600, CGA compatible)
Advanced Graphics Adapter II $605
(with 80186 multi-processor)

DISK DRIVE CONTROLLERS

From National Computer Ltd., Japan's premier maker of controllers

- These compact AT-compatible controller cards are among the smallest available.
- Co-resident, IBM-COMPATIBLE

NDC 5126 AT Winchester only $179
(a half card—so small you won't believe it's AT-compatible)
NDC 5125 AT Winchester/Floppy $189
NDC 5127 XT Winchester only $79
NDC 3011 SASI Controller $91
Data Technology DTC5150CX $99

FLOPPY DISK DRIVES

360-K $89
1.2 Megabyte, AT-Compatible $129
1.2 Megabyte, PC-compatible
(enables your PC or XT to read and write AT-formatted diskettes)
was $299—clearance price NOW $129

EGA SOFTWARE

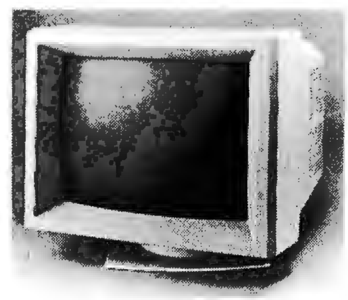
Rix EGA paint $49

CHIPS

256K D-RAM (set of 9) $27
64K D-RAM (set of 9) $9
V-20 (8 MHz) $10.95

INTEL MATH CO-PROCESSORS

8087 . . . . $125
8087-2 . . . . $179
80287-6 . . . . $225



MONITORS

TTL Amber $98
MultiSync . . . . CALL FOR PRICE

HERE'S HOW TO ORDER:

We accept Visa, MasterCard and American Express (no surcharge) and Money Orders. Minimum charge card order $25. No sales tax outside California. (CA residents add 6%.)
Personal or company checks require three weeks to clear.
We insure all orders.
Shipping and handling minimum $5 up to 3 lbs. within continental U.S. Call for shipping info on other orders.

Prices and availability may change.
Returns require authorization and are subject to restocking charge.
Normal manufacturers' warranties apply.
Not responsible for typographical errors.
IBM, MultiSync, MegaCard, Business Card, Mylex EGA Paint and On Track are trademarks of their respective companies.

CALL 800-443-7355
In Calif. call 800-345-4868
Tech. support: 714-533-1651

Hours: Mon.–Fri. 7 a.m.–5 p.m. PDT

HUNTER SALES

P.O. BOX 5008 IRVINE, CA 92715-5008

Inquiry 170

38¢ Each is all you have to pay for the best 5.25" Double Sided, Double Density diskettes to be found!

SuperStar™ and DISK WORLD!™ do it again: super-premium diskettes at the lowest prices to be found anywhere ...without lies, fraud, hokum or crazy deals.

The "Life Is Simple" price list for first-line, top-quality diskettes:

5.25"SSDD (P/N38002)...................38 ea.
5.25" DSDD(P/N38012)...................38 ea.
5.25"SSDD - 96TPI(P/N38022)..........68 ea.
5.25" DSDD - 96TPI(P/N38032).......71 ea.
5.25" DSDD -HD(P/N38042).......$ 1.36 ea.

3.50" SSDD(P/N38052).............$ 1.39 ea.
3.50" DSDD(P/N38062).............$ 1.45 ea.

All prices quoted are for multiples of 100 each. Add 10% for orders of 50 each. All 5.25" diskettes are supplied with Tyvec™ sleeves, user ID labels and write-protect tabs. 3.50" diskettes are supplied with user ID labels.

Don't buy junk.

DISK WORLD!™ has always been the price and quality leader in diskettes.

Our SuperStar™ diskettes are fully certified to at least 65% clipping levels, far in excess of the ANSI and IBM™ standard of 40%.

Furthermore, SuperStar™ diskettes have jackets of at least 8-mils, as opposed to the 6-mil jackets offered by other low-priced brands. (And, believe us, jacket thickness makes a big difference in whether or not you will get read/write errors on your machine.)

SuperStar™ 3.50" diskettes are all licensed by Sony™ Corporation, the inventor of the 3.50" diskette standard, and meet their exacting standards.

All SuperStar™ diskettes carry a FULL LIFETIME WARRANTY against manufacturing and material defects.

Finally, all SuperStar™ diskettes are manufactured in the United States...not in Hong Kong, Taiwan, Pago Pago or other strange and exotic places where the standards of quality might not be quite as high.

In short, SuperStar™ diskettes are equal or superior to such famous brand names as 3m™, TDK™, Fuji™, Maxell™ and just about any other brand you can name!

SuperStar™ diskettes simply cost half as much or less.

The truth about diskette prices, brand names and some other unpleasant facts of life.

There are exactly two kinds of diskettes: good and bad.

The brand name...or absence of a brand name...on a diskette doesn't mean a thing in terms of quality.

Neither does the price.

For example, let's take a hard look at 3M™ diskettes, a brand which we have carried for more than three years.

During that time, we sold more than $ 6,500,000 worth of these diskettes...and that's one hell of a lot of diskettes.

Until very recently, we were proud to market 3M™ brand diskettes. Now we don't advertise them and actively discourage people from buying them.

Why?

Well, here's a quote from the July 21, 1986 Business Week magazine (page 107): "The disks have been plagued by quality problems recently..."according to one industry expert.

And a radical change in 3M's pricing policies has resulted in exactly the same diskette being sold at prices ranging from 29 to 89 cents, depending on who you were.

Not exactly a fair deal for anyone...except 3M™ which, according to Business Week earned a mere $ 73,000,000 in profits on magnetic media in 1985...mostly at your expense and ours while "(t)he diskettes have been plagued by quality problems recently..."

Counterfeiting and The Hong Kong Connection.

If the life of a diskette purchaser weren't bad enough already, there are now two more problems to contend with.

Counterfeiting is the newest game in town for the "let's get rich quick" crowd.

At least two mail order companies are currently operating very comfortable businesses offering "famous brand name" diskettes at unbeatable prices.

The prices ought to be unbeatable, since these two companies simply take an inferior diskette and slip into a Tyvec™ sleeve which they have printed to be an exact duplicate of the "famous brand name", slap on a forged manufacturer's label to the diskette and make a very tidy profit...without regard to the harm they are causing the end user.

Finally, there is now a wave of cheap, poorly made diskettes pouring in from the Orient, mostly Hong Kong. We are not talking about reputable brands such as Sony™ or Fuji™: we are talking about diskettes which are being turned out in sweatshops on outmoded equipment with little or no regard for quality control.

They show up in this country under a variety of names at extremely attractive prices. Prices so attractive, in fact, that your savings on buying these diskettes may just about pay for the cost of replacing your drives after you use them for a while.

90,000 DISK WORLD!™ customers really tell the story.

DISK WORLD!™ serves more than 90,000 customers throughout the United States and now the world. More and more major corporations, universities and governmental bodies have switched over to our SuperStar™ brands because they recognize that SuperStar™...with DISK WORLD!'s integrity standing behind it...offer absolutely the best combination of HIGH-QUALITY and LOW PRICE.

In short, DISK WORLD!™ and SuperStar™ are unbeatable.

Over the past three years, we've sold more than 30,000,000 diskettes.

Our return rate has been less than 1/10,000th of 1%.

Those two facts speak for themselves.

When every bit counts, it's SuperStar™!

We don't have to boast of imaginary factories employing "floppy phreaks", advertise diskettes for 39¢...if you buy 10,000 of them, offer "2 for 1" replacement guaranties or engage in any of the other hokum schemes designed to part you from your money.

Nor do we have to offer you "famous brand name" diskettes at inflated prices that don't offer you any more in quality, but do manage to support fleets of corporate jet aircraft and executive retreats in Minnesota and Canada.

We simply deliver the best for less.

More than a million diskettes a month flow out of our warehouse under the SuperStar™ "non-name". As noted, fewer than 1/10,000th of 1% of these are returned for any reason.

And that should say it all: SuperStar™ means far more for far less.

HOW TO ORDER:
ORDERS ONLY:
1-800-621-6827
(In Illinois: 1-312-256-7140)
INQUIRIES:
1-312-256-7140

FOR FASTEST SERVICE, USE NO -COST MCI MAIL. Our address is DISKORDER. It's a FREE ,MCI MAIL letter. No charge to you. (Situation permitting, we'll ship these orders in 24 hours or less.)

SHIPPING: 5.25" & 3.50" DISKETTES-Add $ 3.00 per each 100 or fewer diskettes. **OTHER ITEMS:** Add shipping charges as shown in addition to other shipping charges. **PAYMENT:** VISA, MASTERCARD and Prepaid orders accepted. **COD ORDERS:** Add $ 5.00 special handling charge. **APO, FPO, AK, HI & PR ORDERS:** Include shipping charges as shown and additional 5% of total order amount to cover PAL and insurance. We ship only to United States addresses, except as shown above. **TAXES:** Illinois residents add 7% sales tax.

DISK WORLD!, Inc.

629 Green Bay Road
Wilmette, Illinois 60091

Mail Order Electronics · Worldwide
Jameco ELECTRONICS

Mail-Order Electronics 415-592-8097

IBM® COMPATIBLE COMPUTER ACCESSORIES

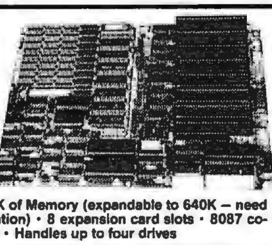
NOW YOU CAN BUILD AN IBM PC/XT COMPATIBLE!

Part No.	Description	Price
IBM-64K (2 sets)	64K RAM CHIPS (18)	$ 24.98
KB-83	83-KEY KEYBOARD	$ 29.95
IBM-FCC	FLOPPY CONTROLLER CARD	$ 39.95
IBM-Case	FLIP-TOP CASE	$ 39.95
IBM-MCC	MONOCHROME CARD	$ 69.95
IBM-PS	POWER SUPPLY	$ 69.95
FD55B	TEAC 5¼" DISK DRIVE	$109.95
IBM-MON	12" MONOCHROME MONITOR	$ 99.95
IBM-MB	MOTHERBOARD (with Zero-K RAM — Includes BIOS-ROM)	$129.95
	Regular List	$614.58

SPECIAL!

IBM-Special (Includes the 9 items listed above)........ **$549.95**

IBM PC XT Compatible Motherboard



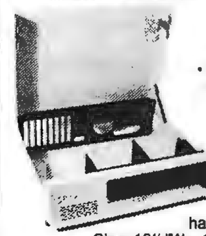
SALE!

• Comes with Zero-K of Memory (expandable to 640K — need 128K RAM for operation) • 8 expansion card slots • 8087 co-processor capability • Handles up to four drives

The whole board comes complete (including BIOS ROM) with system board, specifications and schematics. Ideal for engineer or OEM applications! Size: 12"L x 8½"W x 15/16"H • Wt. 2 lbs.

IBM-MB............ $129.95

IBM PC XT Compatible Case



Expand Your PC!

• Metal housing and chassis with coating & anti-static treatment
• Plastic face plate
• Flip-up lid for easy access
• Back plate set for expansion including 8 card slots and power supply mount
• Color: grey • All necessary hardware for expansion included

Size: 19½"W x 16½"D x 5¾"H • Weight: 17 lbs.

IBM-Case........... $39.95

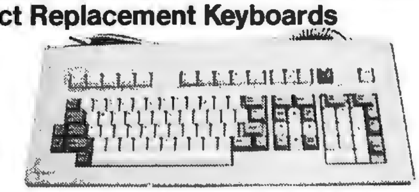
5¼" Disk Drives

SA455	Shugart 5¼" Half-Ht.	$109.95
FD55B	Teac 5¼" Half-Ht.	$109.95
TM100-2	Tandon 5¼" Full-Ht.	$119.95

IBM PC XT Compatible Direct Replacement Keyboards

• Identical layout as original IBM PC keyboard • Illuminated "Caps Lock" and "Num Lock" keys • 84 keys, mechanical switches (uses Intel 8048) • Complete with schematics and instructions • Color: off-white • Size: 17¼"L x 7¼"W x 1¼"H

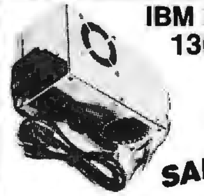
IBM-KB............. $59.95



• Enhanced PC/XT keyboard (equivalent to Keytronics™ 5151) • Separate cursor and numeric keyboard • Typewriter style layout: easier to learn! • LED indicators • Ready-to-go, complete with schematic & instructions • Color: off-white • Size: 20"L x 8½"W x 1½"H

IBM-ENH............ $89.95

IBM PC/XT Equivalent 130W Power Supply



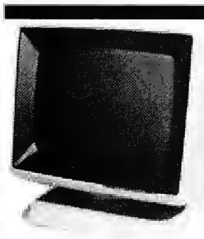
+5VDC @ 15A
−5VDC @ 0.5A
+12VDC @ 4.2A
−12VDC @ 0.5A

SALE!

• Input: 100V-130V @ 47Hz • Output: +5VDC @ 15A, −5VDC @ 0.5A, +12VDC @ 4.2A, −12VDC @ 0.5A • Plug compatible connectors • Fits into IBM PC • Built-in fan • Weight: 6 lbs. • Size: 9½"L x 5½"W x 4⅝"H • Spec. included

IBM-PS............. $69.95

MONITOR

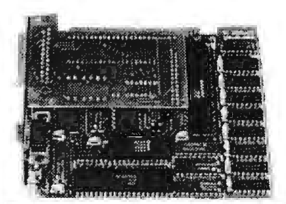


12" Green Monochrome Monitor for IBM PC

FEATURES:
• TTL input • Band width: 20MHz
• Input impedance: 75 Ohm
• Resolution: 800 lines at center
• Power consumption: 30 watts
• Complete with tilt/swivel monitor stand
• Weight: 19 lbs • Size: 14.5"W x 16"D x 16.5"H

IBM-MON............. $99.95

ZUCKERBOARD



MADE IN THE U.S.A.

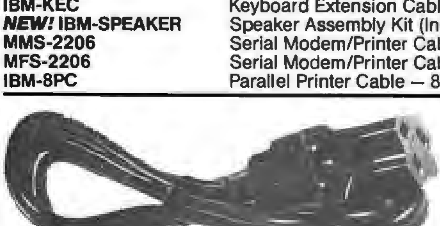
Integrated Color Board for IBM PC, XT, AT and Portable Computers

• Fully compatible with all IBM color software and RGB monitors • Fits the difficult-to-use half card short slot or any long slot • Requires 1/3 the power of conventional color boards • Quality backed by two year warranty

IBM-ICB........... $99.95

CABLES, BEZELS AND SPEAKERS

IBM-BEZEL	Half-Height Bezel (Fills empty space when mounting half-height drives)	$4.95
IBM-KEC	Keyboard Extension Cable for IBM PC & XT Computers — 5' length	$5.95
NEW! IBM-SPEAKER	Speaker Assembly Kit (Includes speaker, wire and 4-pin Molex connector)	$9.95
MMS-2206	Serial Modem/Printer Cable — 6' length — DB25P (Male) to DB25P (Male)	$6.95
MFS-2206	Serial Modem/Printer Cable — 6' length — DB25P (Male) to DB25S (Female)	$7.49
IBM-8PC	Parallel Printer Cable — 8' length — DB25P (Male) to CEN36M Centronics (Male)	$9.95



Monitor Adapter Power Cable

Enables user to plug any monitor into your IBM Power Supply
(IBM-PS — See above)

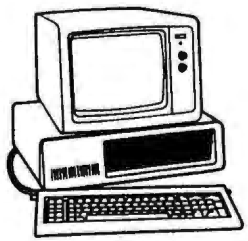
MCable........................ $9.95

Inquiry 147

JADE XPC or XPC-AT

- 640K of RAM
- 4.77 MHz 8088
- 7 MHz turbo mode
- 135 watt power supply
- 360K disk drive
- Floppy disk controller
- 5151-style keyboard
- 8 expansion slots
- PC-style case
- 90 day warranty

$588



- 1 MB of RAM
- 6 MHz 80286
- 8 MHz turbo mode
- 200 watt power supply
- 1.2 MB disk drive
- Floppy/hard disk controller
- AT-style keyboard
- 8 expansion slots
- AT-style case
- 90 day warranty

$1288

OPTION #1
Two disk drives
Mono graphics card
Parallel printer port
Amdek 310A monitor

XPC $898 XPC-AT $1698

OPTION #2
20 MB hard disk
Mono graphics card
Parallel printer port
PGS MAX-12E monitor

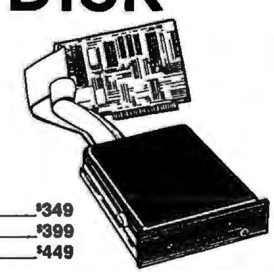
XPC $1288 XPC-AT $1898

OPTION #3
20 MB hard disk
Color graphics card
Parallel printer port
Hitachi hi-res color monitor

XPC $1388 XPC-AT $1998

HARD DISK DRIVE

$349



10 MB Kit	$349
20 MB Kit	$399
20 MB hard disk card	$449

20 MB HI-SPEED HARD DISK FOR YOUR AT

$399

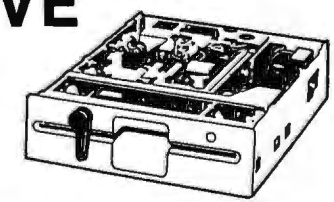
Limited quantity.

30 MB Hi-speed	$695
40 MB Hi-speed	$895
70 MB Hi-speed	$1695
120 MB Hi-speed	$3395

QUME DISK DRIVE

- Double sided
- Double density
- 360K for PC

$69



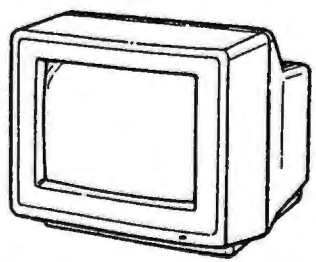
Teac FD55 BV	$109
Tandon TM100-2	$109
1.2 MB for AT	$149

HIGH RESOLUTION COLOR MONITOR

$269

List Price $699

Free PC RGB cable included



- IBM PC color graphics compatible
- High resolution 640 x 240
- Excellent dot pitch .38 mm
- 16 true colors, 2000 characters
- Accepts standard RGB input signals
- Non-glare picture tube
- New designer sculpted case

Manufactured by Hitachi

HITACHI®

MICROSPEED FAST 88
7 MHz TURBO CARD FOR YOUR PC

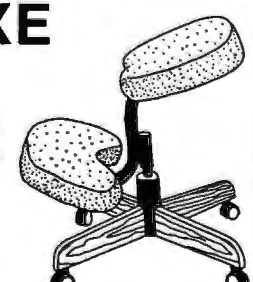
$99

- Uses no slots
- 6.14, 6.67, 7.37 MHz
- No wait states
- External switch
- Reset switch

DELUXE BACK CHAIR

$49

List Price $99



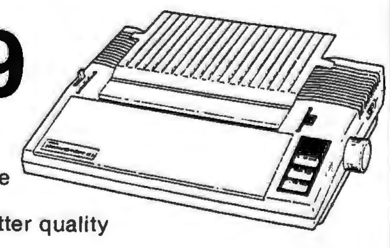
EPSON HOMEWRITER

100 CPS PRINTER

$189

List Price $309

Including free interface & cable



- 16 cps near letter quality
- Graphics
- Bidirectional printing
- SelecType
- Tractor feed
- Optional cut sheet feeder
- 32K buffer

EPSON®

Monochrome Monitors

Jade green	$99
Jade amber	$99
Amdek 310A amber	$149
Quimax 14" green T/S	$139
Quimax 14" amber T/S	$139
Thompson green	$99
Thompson amber	$99
PGS MAX 12E amber	$159

Kaypro Sale

Limited quantity. 10-day money-back guarantee—Call for details.

Kaypro 2000	$995
Kaypro 10	$795
Kaypro 16	$1195
Kaypro 2000 accessories	30% off

PC Power Supply

135 watt power supply	$69
150 watt power supply	$79

PC or AT Video Boards

Hercules Color Graphics	$159
Jade Color Graphics	$89
Hercules Mono Graphics Plus	$199
JADE Monochrome Graphics	$99
Quad EGA+	$359
Tecmar Graphics Master	$439
JADE EGA+	$239

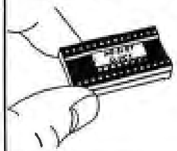
High Speed APU Chips

8087-3 5 MHz	$119
8087-2 8 MHz	$159
80287-3 5 MHz	$179
80287-8 8 MHz	$269
NEC V-20 5 MHz	$11
NEC V-20 8 MHz	$14
NEC V-30 8 MHz	$16

RGB Color Monitors

Hitachi 640 x 262	$269
Magnavox 640 x 240	$319
Quimax EGA 640 x 350	$489
NEC Multisync 800 x 560	CALL
PGS HX-12 640 x 240	$439
PGS HX-12E 640 x 350	$529

NO-SLOT CLOCK $49

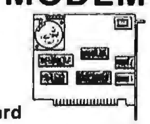


Real time clock/calendar for your PC-XT or clone

Modems

1200 BAUD MODEM

$129



Internal 1200 baud card with software	$129
External 1200 baud	$139
Hayes 1200B internal card w/o software	$329
Hayes 1200B internal card w/Smartcom II	$359
Hayes Smartmodem 1200	$389
Hayes 2400B internal card w/Smartcom II	$549
Hayes Smartmodem 2400	$589

PC Mouse

3 Button Mouse with pop-up software, serial	$99
PC Mouse w/Dr. Halo II	$119
Microsoft Mouse with PC Paintbrush software	$129

POWER/SURGE PROTECTION

6 socket, UL listed, U.S. made Power/surge strip

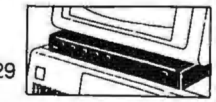
$12.95

List Price $49

6 button, under-monitor, swivel Power/surge control center

$79.95

List Price $129



Iso-Bar

These industrial quality ISO-BAR's contain surge suppression circuitry & built-in noise filters plus a 15 amp circuit breaker.

6 socket, 1 filter ISO-BAR	$44
4 socket, 2 filter ISO-BAR	$59
8 socket, 4 filter ISO-BAR	$69
200W standby power supply	$269
425W standby power supply	$399
675W standby power supply	$599
1000W standby power supply	$899

Apple Accessories

IIc ½ high disk drive	$119
IIe ½ high disk drive	$99
IIe, II+ disk drive	$129
IIe 64K, 80 column card	$49
II+ 80 column card	$119
Grappler printer card	$89
II+ 16K RAM card	$39
II+, IIe printer card & cable	$49
II+, IIe cooling fan	$59

Plotters

Roland DXY-800 8 pen	$399
HP 7470 2 pen	$899
HP 7475 6 pen	$1699
HP 7550A 8 pen	$3399
HP Color Pro 8 pen	$1699
Houston Inst. PC 695	$599
Houston Inst. DMP 40P	$849
Houston Inst. DMP 29	$1699
Houston Inst. DMP 41/42	$2495
Houston Inst. DMP 51/52	$3795
Houston Inst. DMP 56	$4495

PC Multifunction Cards

384K Jade Seven Pak w/0K, p, s, c, c, g, software	$89
384K Jade Seven Pak with 384K installed	$139
AST Six Pak Plus 64K	$159
AST Six Pak Plus 384K	$229
AST Six Pak Premium 512K	$389
AST Rampage 256K	$239
AST Rampage 2 MB	$699
Tall Tree JRAM-3 0K	$189
Tall Tree JRAM-3 2 MB	$599
Intel Above Board 64K	$279
Intel Above Board 2 MB	$599
Quad Board w/0K	$119
Quad Board 384K	$169

AT Multifunction Cards

1.5 MB Jade-AT Pak w/0K, p, s, optional g, s	$199
1.5 MB Jade-AT Pak w/1.5 MB RAM installed	$399
AST Advantage-AT 128K	$349
AST Advantage-AT 3 MB	$989
AST Rampage-AT 512K	$469
AST Rampage-AT 2 MB	$749
Tall Tree JRAM-AT 0K	$169
Tall Tree JRAM-AT 2 MB	$599
Tall Tree JRAM 3-AT 0K	$239
Tall Tree JRAM 3-AT 2 MB	$699
Intel Above Board 128K	$399
Intel Above Board 4 MB	$1299

Printers On Sale!

Citizen 120D 120 cps	$198
Okidata 192 160 cps	$339
Epson LX-86 120 cps	$100 off
Epson FX-85 160 cps	$150 off
Epson FX-286 160 cps	$300 off
Epson LQ-800 24 pin	$230 off
Epson LQ-1000 24 pin	$300 off
Epson EX-800 300 cps	$200 off
Epson EX-1000 300 cps	$200 off
Epson LQ-2500 324 cps	$400 off
14 cps daisywheel	$179
22 cps daisywheel	$199
40 cps daisywheel	$429

64K BUFFER FOR YOUR EPSON

Serial or parallel input. Fits MX, RX, FX printers

Limited Quantity

$99

Microfazer Buffers

Expands to 64K (Parallel to 512K)

8K Parallel in/parallel out	$139
64K Parallel in/parallel out	$164
128K Parallel in/parallel out	$269
512K Parallel in/parallel out	$499

Your choice: serial in/serial out; parallel in/serial out; serial in/parallel out. **8K** $169 **64K** $199

JADE COMPUTER PRODUCTS 1975 - 1986 11 YEARS SATISFACTION GUARANTEED!

JADE COMPUTER

4901 W. Rosecrans Ave. Box 5046 Hawthorne, CA 90251-5046

VISA MasterCard AMERICAN EXPRESS

Prices at our six store locations will be higher.
We accept cash, checks, credit cards or purchase orders from qualified firms and institutions. **Minimum prepaid order $15.00.** California residents add 6½% tax. **Prices & availability subject to change without notice.** Shipping & handling charges via UPS Ground 50¢/lb. UPS Air $1.00/lb. **Minimum charge $3.00.**

PLACE ORDERS TOLL FREE!

Continental U.S.A.

(800)421-5500

Inside California

(800)262-1710

THE LATEST IN PC ENHANCEMENT PRODUCTS

ENHANCED GRAPHICS ADAPTOR

100% IBM COMPATIBLE—PASSES IBM EGA DIAGNOSTICS

* COMPATIBLE WITH IBM EGA, COLOR GRAPHICS ADAPTOR AND MONOCHROME ADAPTOR
* DISPLAYS 16 COLORS OUT OF 64 COLORS
* COMES WITH 256K OF VIDEO RAM
* DUAL SCANNING FREQUENCIES WORKS WITH STANDARD OR EGA TYPE RGB MONITORS
* LIGHT PEN INTERFACE

NOW ONLY $259.95

EGA MONITOR

FULL ONE YEAR WARRANTY

* EGA AND CGA COMPATIBLE
* SCANNING FREQUENCIES: 15.75 KHz / 21.85 KHz
* 14" BLACK MATRIX, NON-GLARE SCREEN
* RESOLUTION: 640 x 200 / 640 x 350
* .31 DOT, 25MHz
* 16 COLORS OUT OF 64
* TEXT SWITCH-GREEN/AMBER

$499.95

BUY BOTH $699!

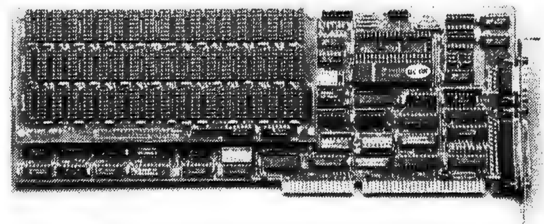
20 MB HARD DISK SYSTEM

INCLUDES HARD DISK CONTROLLER, CABLES AND INSTRUCTIONS. ALL DRIVES ARE PRE-TESTED AND COME WITH A 1 YEAR WARRANTY.

$389.95

AT MULTIFUNCTION CARD $199.95

ADDS UP TO 3 MEGABYTES OF USER EXPANDABLE MEMORY

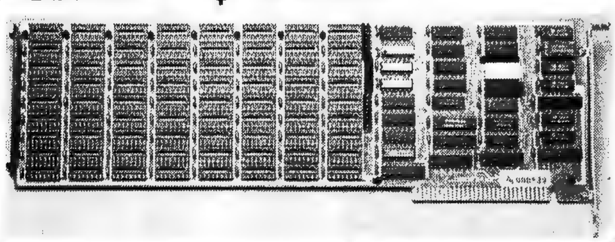


* SHIPPED WITH ZERO K RAM, USER EXPANDABLE TO 1.5 MEGABYTES RAM ON BOARD, UP TO 3 MEGABYTES WITH OPTIONAL PIGGYBACK CARD
* USES 64K OR 256K DYNAMIC RAMS
* PARALLEL PORT & GAME PORT
* SERIAL PORT
* OPTIONAL SECOND SERIAL PORT

PIGGYBACK MEMORY CARD (NO MEMORY INSTALLED) $49.95

EXPANDED MEMORY CARD $199.95

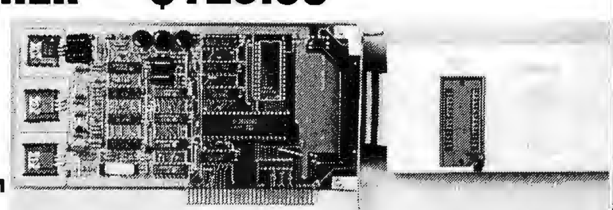
UP TO 2 MEGABYTES OF LOTUS/INTEL COMPATIBLE MEMORY



* CONFORMS TO LOTUS/INTEL EXPANDED MEMORY SPECIFICATIONS (EMS)
* SHIPPED WITH ZERO K RAM, USER EXPANDABLE TO 2 MEGABYTES
* USES 64K OR 256K DYNAMIC RAMS
* USE AS EXPANDED (EMS) OR CONVENTIONAL MEMORY, RAMDISK OR SPOOLER
* SOFTWARE INCLUDES EMS DEVICE DRIVERS, PRINT SPOOLER AND RAMDISK

EPROM PROGRAMMER $129.95

FOR IBM PC/XT/AT AND COMPATIBLES



* PROGRAMS 27xxx SERIES EPROMS UP TO 27512
* MENU DRIVEN SOFTWARE PROVIDED ON DISKETTE
* AUTOMATICALLY SETS PROGRAMMING VOLTAGE
* LOADS AND SAVES EPROM BUFFER TO DISK
* READ, WRITE, COPY, VERIFY OR CHECK BLANK
* DEBUG STYLE EDITOR FOR EASY MODIFICATION OF PROGRAM
* SPLITS OR COMBINES EPROMS OF DIFFERING SIZES
* INTERNAL CARD WITH EXTERNAL CABLE FOR A ZIF SOCKET

JDR Microdevices

110 Knowles Drive, Los Gatos, CA 95030
Toll Free 800-538-5000 • (408) 866-6200 • FAX (408) 378-8927 • Telex 171-110

BYTE's Ongoing Monitor Box

BOMB *Card Explanation*

MAKE YOUR VOTE COUNT

Just as with your regular issues of BYTE, the editors would like to survey how effectively each article in our 1986 BYTE IBM PC special issue serves our readers. Please fill out the card following the Reader Service index, mail it in, and look for the tabulation of your votes in a future issue of BYTE.

First-place winner based on the number of respondents who read it will receive a bonus of $100, and the second-place author will receive $50. The first-place winner for quality will receive an award of $50. Articles written by anyone under the aegis of BYTE automatically forfeits the bonus to the article placing next.

R·E·A·D·E·R S·E·R·V·I·C·E

BYTE ADVERTISING SALES STAFF:

Dennis J. Riley, Advertising Sales Manager, One Phoenix Mill Lane, Peterborough, NH 03458, tel. (603) 924-9281

NEW ENGLAND
ME, NH, VT, MA, RI,
ONTARIO, CANADA & EASTERN CANADA
Paul McPherson Jr. (617) 262-1160
McGraw-Hill Publications
575 Boylston Street
Boston, MA 02116

ATLANTIC
NY, NYC, CT, NJ (NORTH)
Leah G. Rabinowitz (212) 512-2096
McGraw-Hill Publications
1221 Avenue of the Americas—
39th Floor
New York, NY 10020

Dick McGurk (203) 968-7111
McGraw-Hill Publications
Building A—3rd Floor
777 Long Ridge Road
Stamford, CT 06902

EAST
PA, KY, OH, NJ (SOUTH),
MD, VA, W.VA, DE, D.C.
Daniel Ferro (215) 496-3833
McGraw-Hill Publications
Three Parkway
Philadelphia, PA 19102

SOUTHEAST
NC, SC, GA, FL, AL, TN
Maggie M. Dorvee (404) 252-0626
McGraw-Hill Publications
4170 Ashford-Dunwoody Road—
Suite 420
Atlanta, GA 30319

MIDWEST
IL, MO, KS, IA, ND, SD, MN, WI, NB, IN, MI
Bob Denmead (312) 751-3740
McGraw-Hill Publications
Blair Building
645 North Michigan Ave.
Chicago, IL 60611

SOUTHWEST, ROCKY MOUNTAIN
UT, CO, WY, OK, TX, AR, MS, LA
Kevin Harold (214) 458-2400
McGraw-Hill Publications
Prestonwood Tower—Suite 907
5151 Beltline
Dallas, TX 75240

SOUTH PACIFIC
SOUTHERN CA, AZ, NM, LAS VEGAS
Jack Anderson (714) 557-6292
McGraw-Hill Publications
3001 Red Hill Ave.
Building #1—Suite 222
Costa Mesa, CA 92626

Karen Niles (213) 480-5243, 487-1160
McGraw-Hill Publications
3333 Wilshire Boulevard #407
Los Angeles, CA 90010

NORTH PACIFIC
HI, WA, OR, ID, MT, NORTHERN CA,
NV (except LAS VEGAS), W. CANADA
Mike Kisseberth (415) 362-4600
McGraw-Hill Publications
425 Battery Street
San Francisco, CA 94111

Bill McAfee (415) 349-4100
McGraw-Hill Publications
951 Mariner's Island Blvd.—3rd Floor
San Mateo, CA 94404

WEST COAST SURPLUS AND RETAIL ACCOUNTS
Tom Harvey (805) 964-8577
3463 State Street—Suite 256
Santa Barbara, CA 93105

The Buyer's Mart
Karen Burgess (603) 924-3754
BYTE Publications
One Phoenix Mill Lane
Peterborough, NH 03458

BYTE BITS (2x3)
Dan Harper (603) 924-6830
BYTE Publications
One Phoenix Mill Lane
Peterborough, NH 03458

Post Card Mailings National
Ed Ware (603) 924-9281
Bradley Browne (603) 924-6166
BYTE Publications
One Phoenix Mill Lane
Peterborough, NH 03458

International Advertising Sales Staff:

Mr. Hans Csokor
Publimedia
Reisnerstrasse 61
A-1037 Vienna, Austria
222 75 76 84

Mrs. Gurit Gepner
McGraw-Hill Publishing Co.
PO Box 2156
Bat Yam, 59121 Israel
3 866 561 321 39

Mr. Fritz Krusebecker
McGraw-Hill Publishing Co.
Liebigstrasse 19
D-6000 Frankfurt/Main 1
West Germany
69 72 01 81

Mrs. Maria Sarmiento
Pedro Teixeira 8, Off. 320
Iberia Mart 1
Madrid 4, Spain
1 45 52 891

Mr. Andrew Karnig
Andrew Karnig & Associates
Finnbodavagen
S-131 31 Nacka, Sweden
8-44 0005

Mr. Alain Faure
McGraw-Hill Publishing Co.
128 Faubourg Saint Honore
75008 Paris
France
(1) 42-89-03-81

Mr. Arthur Scheffer
McGraw-Hill Publishing Co.
34 Dover St.
London W1X 3RA
England 01 493 1451

Manuela Capuano
McGraw-Hill Publishing Co.
Via Flavio Baracchini 1
20123 Milan, Italy
02 86 90 617

Seavex Ltd.
400 Orchard Road, #10-01
Singapore 0923
Republic of Singapore
Tel: 734-9790
Telex: RS35539 SEAVEX

Seavex Ltd.
503 Wilson House
19-27 Wyndham St.
Central, Hong Kong
Tel: 5-260149
Telex: 60904 SEVEX HX

Hiro Morita
McGraw-Hill Publishing Co.
Overseas Corp.
Room 1528
Kasumigaseki Bldg.
3-2-5 Kasumigaseki,
Chiyoda-Ku
Tokyo 100, Japan
3 581 9811

Mr. Ernest McCrary
Empresa Internacional de
Comunicacoes Ltda.
Rua da Consolacao, 222
Conjunto 103
01302 Sao Paulo, S.P., Brasil
Tel: (11) 259-3811
Telex: (100) 32122 EMBN

* *Correspond directly with company.*

TO GET FURTHER information on the products advertised in BYTE, either pick up your touch-tone telephone and use TIPS (if you are a subscriber), or fill out the reader service card. Either way full instructions are provided following this reader service index which is provided as an additional service by the publisher, who assumes no liability for errors or omissions.

TIPS

SUBSCRIBERS ONLY! *

Use BYTE's Telephone Inquiry Processing System

Using TIPS can bring product information as much as 10 days earlier.

SEND FOR YOUR SUBSCRIBER I.D. CARD

1) If you are a new subscriber or have lost your I.D. card, circle #1 on the Reader Service Card; **attach mailer label**. We will immediately send your personal TIPS subscriber card.

GET PREPARED

2) Write your Subscriber Number, as printed on your Subscriber I.D. Card, in boxes in Step 5 below. (Do not add 0's to fill in blank boxes)

3) Write numbers for information desired in boxes in Step 7b below. (Do not add 0's to fill in blank boxes.)

CALL TIPS

4) Now, on a Touch-Tone telephone dial: (413) 442-2668 and wait for voice commands.

ENTER YOUR SUBSCRIBER AND ISSUE NUMBERS

5) When TIPS says: "Enter Subscriber Number"
(Enter by pushing the numbers and symbols [# or * enclosed in the boxes] on telephone pad ignoring blank boxes)
Enter □□□□□□□□□□ # #

6) When TIPS says "Enter magazine code & issue code"
Enter 1 # 1 3 6 # #

ENTER YOUR INQUIRIES

7a) When TIPS says "Enter (next) Inquiry Number"
Enter one inquiry selection from below (ignore blank boxes)

b) Repeat 7a as needed (maximum 17 inquiry numbers)

1. □□□ # #	6. □□□ # #	10. □□□ # #	14. □□□ # #
2. □□□ # #	7. □□□ # #	11. □□□ # #	15. □□□ # #
3. □□□ # #	8. □□□ # #	12. □□□ # #	16. □□□ # #
4. □□□ # #	9. □□□ # #	13. □□□ # #	17. □□□ # #
5. □□□ # #			

END SESSION

8) End session by entering * * 9 1 # #

9) Hang up after hearing final message
If you are a subscriber and need assistance, call (603) 924-9281.

Available 24 Hours, 7 Days a Week

If you are not a subscriber fill out the subscription card found in this issue or, call BYTE Circulation 800-258-5485.

***Domestic and Canadian Subscribers Only!**

BYTE READER SERVICE

Fill out this coupon carefully. PLEASE PRINT. Requests cannot be honored unless the zip code is included. This card is valid for 6 months from cover date.

I am requesting information for: ☐ Business ☐ Personal ☐ Both

I purchased this copy by ☐ Subscription

☐ Newsstand, computer store, bookstore

Name ________________ IBM 46SRSX

(Title) ________________ (Company) ________________

Address ________________ Telephone ________________

City ________________ State ______ Zip ______

Country ________________

1 23 45 67 89	111 133 155 177 199	221 243 265 287 309	331 353 375 397 419	441 463 485 507 529	551 573 595.617 639	661 683 705 727 749	771 793
2 24 46 68 90	112 134 156 178 200	222 244 266 288 310	332 354 376 398 420	442 464 486 508 530	552 574 596 618 640	662 684 706 728 750	772 794
3 25 47 69 91	113 135 157 179 201	223 245 267 289 311	333 355 377 399 421	443 465 487 509 531	553 575 597 619 641	663 685 707 729 751	773 795
4 26 48 70 92	114 136 158 180 202	224 246 268 290 312	334 356 378 400 422	444 466 488 510 532	554 576 598 620 642	664 686 708 730 752	774 796
5 27 49 71 93	115 137 159 181 203	225 247 269 291 313	335 357 379 401 423	445 467 489 511 533	555 577 599 621 643	665 687 709 731 753	775 797
6 28 50 72 94	116 138 160 182 204	226 248 270 292 314	336 358 380 402 424	446 468 490 512 534	556 578 600 622 644	666 688 710 732 754	776 798
7 29 51 73 95	117 139 161 183 205	227 249 271 293 315	337 359 381 403 425	447 469 491 513 535	557 579 601 623 645	667 689 711 733 755	777 799
8 30 52 74 96	118 140 162 184 206	228 250 272 294 316	338 360 382 404 426	448 470 492 514 536	558 580 602 624 646	668 690 712 734 756	778 800
9 31 53 75 97	119 141 163 185 207	229 251 273 295 317	339 361 383 405 427	449 471 493 515 537	559 581 603 625 647	669 691 713 735 757	779 801
10 32 54 76 98	120 142 164 186 208	230 252 274 296 318	340 362 384 406 428	450 472 494 516 538	560 582 604 626 648	670 692 714 736 758	780 802
11 33 55 77 99	121 143 165 187 209	231 253 275 297 319	341 363 385 407 429	451 473 495 517 539	561 583 605 627 649	671 693 715 737 759	781 803
12 34 56 78 100	122 144 166 188 210	232 254 276 298 330	342 364 386 408 430	452 474 496 518 540	562 584 606 628 650	672 694 716 738 760	782 804
13 35 57 79 101	123 145 167 189 211	233 255 277 299 321	343 365 387 409 431	453 475 497 519 541	563 585 607 629 651	673 695 717 739 761	783 805
14 36 58 80 102	124 146 168 190 212	234 256 278 300 322	344 366 388 410 432	454 476 498 520 542	564 586 608 630 652	674 696 718 740 762	784 806
15 37 59 81 103	125 147 169 191 213	235 257 279 301 323	345 367 389 411 433	455 477 499 521 543	565 587 609 631 653	675 697 719 741 763	785 807
16 38 60 82 104	126 148 170 192 214	236 258 280 302 324	346 368 390 412 434	456 478 500 522 544	566 588 610 632 654	676 698 720 742 764	786 808
17 39 61 83 105	127 149 171 193 215	237 259 281 303 325	347 369 391 413 435	457 479 501 523 545	567 589 611 633 655	677 699 721 743 765	787 809
18 40 62 84 106	128 150 172 194 216	238 260 282 304 326	348 370 392 414 436	458 480 502 524 546	568 590 612 634 656	678 700 722 744 766	788 810
19 41 63 85 107	129 151 173 195 217	239 261 283 305 327	349 371 393 415 437	459 481 503 525 547	569 591 613 635 657	679 701 723 745 767	789 811
20 42 64 86 108	130 152 174 196 218	240 262 284 306 328	350 372 394 416 438	460 482 504 526 548	570 592 614 636 658	680 702 724 746 768	790 812
21 43 65 87 109	131 153 175 197 219	241 263 285 307 329	351 373 395 417 439	461 483 505 527 549	571 593 615 637 659	681 703 725 747 769	791 813
22 44 66 88 110	132 154 176 198 220	242 264 286 308 330	352 374 396 418 440	462 484 506 528 550	572 594 616 638 660	682 704 726 748 770	792 814

BYTE's BOMB is your direct line to the editor's desk. Each month, the two top-rated authors receive bonuses based on your evaluation. First look at the list of this month's articles and corresponding article numbers (located on the page preceding the Reader Service list), then rate each article you've read as Excellent, Good, Fair, or Poor, based on your overall impression of the article, by circling the appropriate number in each column below. Your feedback helps us produce the best possible magazine each month.

Article No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Excellent	1	5	9	13	17	21	25	29	33	37	41	45	49	53	57	61	65	69	73	77	81	85	89	93	97
Good	2	6	10	14	18	22	26	30	34	38	42	46	50	54	58	62	66	70	74	78	82	86	90	94	98
Fair	3	7	11	15	19	23	27	31	35	39	43	47	51	55	59	63	67	71	75	79	83	87	91	95	99
Poor	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72	76	80	84	88	92	96	100

Article No.	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
Excellent	101	105	109	113	117	121	125	129	133	137	141	145	149	153	157	161	165	169	173	177	181	185	189	193	197
Good	102	106	110	114	118	122	126	130	134	138	142	146	150	154	158	162	166	170	174	178	182	186	190	194	198
Fair	103	107	111	115	119	123	127	131	135	139	143	147	151	155	159	163	167	171	175	179	183	187	191	195	199
Poor	104	108	112	116	120	124	128	132	136	140	144	148	152	156	160	164	168	172	176	180	184	188	192	196	200

BYTE READER SERVICE

Fill out this coupon carefully. PLEASE PRINT. Requests cannot be honored unless the zip code is included. This card is valid for 6 months from cover date.

I am requesting information for: ☐ Business ☐ Personal ☐ Both

I purchased this copy by ☐ Subscription

☐ Newsstand, computer store, bookstore

Name ________________ IBM 46SRSX

(Title) ________________ (Company) ________________

Address ________________ Telephone ________________

City ________________ State ______ Zip ______

Country ________________

1 23 45 67 89	111 133 155 177 199	221 243 265 287 309	331 353 375 397 419	441 463 485 507 529	551 573 595 617 639	661 683 705 727 749	771 793
2 24 46 68 90	112 134 156 178 200	222 244 266 288 310	332 354 376 398 420	442 464 486 508 530	552 574 596 618 640	662 684 706 728 750	772 794
3 25 47 69 91	113 135 157 179 201	223 245 267 289 311	333 355 377 399 421	443 465 487 509 531	553 575 597 619 641	663 685 707 729 751	773 795
4 26 48 70 92	114 136 158 180 202	224 246 268 290 312	334 356 378 400 422	444 466 488 510 532	554 576 598 620 642	664 686 708 730 752	774 796
5 27 49 71 93	115 137 159 181 203	225 247 269 291 313	335 357 379 401 423	445 467 489 511 533	555 577 599 621 643	665 687 709 731 753	775 797
6 28 50 72 94	116 138 160 182 204	226 248 270 292 314	336 358 380 402 424	446 468 490 512 534	556 578 600 622 644	666 688 710 732 754	776 798
7 29 51 73 95	117 139 161 183 205	227 249 271 293 315	337 359 381 403 425	447 469 491 513 535	557 579 601 623 645	667 689 711 733 755	777 799
8 30 52 74 96	118 140 162 184 206	228 250 272 294 316	338 360 382 404 426	448 470 492 514 536	558 580 602 624 646	668 690 712 734 756	778 800
9 31 53 75 97	119 141 163 185 207	229 251 273 295 317	339 361 383 405 427	449 471 493 515 537	559 581 603 625 647	669 691 713 735 757	779 801
10 32 54 76 98	120 142 164 186 208	230 252 274 296 318	340 362 384 406 428	450 472 494 516 538	560 582 604 626 648	670 692 714 736 758	780 802
11 33 55 77 99	121 143 165 187 209	231 253 275 297 319	341 363 385 407 429	451 473 495 517 539	561 583 605 627 649	671 693 715 737 759	781 803
12 34 56 78 100	122 144 166 188 210	232 254 276 298 330	342 364 386 408 430	452 474 496 518 540	562 584 606 628 650	672 694 716 738 760	782 804
13 35 57 79 101	123 145 167 189 211	233 255 277 299 321	343 365 387 409 431	453 475 497 519 541	563 585 607 629 651	673 695 717 739 761	783 805
14 36 58 80 102	124 146 168 190 212	234 256 278 300 322	344 366 388 410 432	454 476 498 520 542	564 586 608 630 652	674 696 718 740 762	784 806
15 37 59 81 103	125 147 169 191 213	235 257 279 301 323	345 367 389 411 433	455 477 499 521 543	565 587 609 631 653	675 697 719 741 763	785 807
16 38 60 82 104	126 148 170 192 214	236 258 280 302 324	346 368 390 412 434	456 478 500 522 544	566 588 610 632 654	676 698 720 742 764	786 808
17 39 61 83 105	127 149 171 193 215	237 259 281 303 325	347 369 391 413 435	457 479 501 523 545	567 589 611 633 655	677 699 721 743 765	787 809
18 40 62 84 106	128 150 172 194 216	238 260 282 304 326	348 370 392 414 436	458 480 502 524 546	568 590 612 634 656	678 700 722 744 766	788 810
19 41 63 85 107	129 151 173 195 217	239 261 283 305 327	349 371 393 415 437	459 481 503 525 547	569 591 613 635 657	679 701 723 745 767	789 811
20 42 64 86 108	130 152 174 196 218	240 262 284 306 328	350 372 394 416 438	460 482 504 526 548	570 592 614 636 658	680 702 724 746 768	790 812
21 43 65 87 109	131 153 175 197 219	241 263 285 307 329	351 373 395 417 439	461 483 505 527 549	571 593 615 637 659	681 703 725 747 769	791 813
22 44 66 88 110	132 154 176 198 220	242 264 286 308 330	352 374 396 418 440	462 484 506 528 550	572 594 616 638 660	682 704 726 748 770	792 814

To get further information on the products advertised in BYTE, fill out the reader service card with your name and address. Then circle the appropriate numbers for the advertisers you select from the list. Add a first-class stamp to the card, then drop it in the mail. Not only do you gain information, but our advertisers are encouraged to use the marketplace provided by BYTE. This helps us bring you a bigger BYTE. The index is provided as an additional service by the publisher, who assumes no liability for errors or omissions.

PLACE
POSTAGE
HERE

BYTE

READER SERVICE
PO BOX 298
DALTON, MA 01227-0298
USA

PLACE
POSTAGE
HERE

BYTE

READER SERVICE
PO BOX 298
DALTON, MA 01227-0298
USA

For your own subscription to BYTE, complete this card and mail.

46SRSX

Name ______

Address ______

City ______

State ______ Zip ______ Country ______

☐ Bill me (U.S.A., Canada, Mexico only)

☐ Check enclosed (All checks must be payable in U.S. funds drawn on a U.S. bank.)

Charge to my ☐ MasterCard ☐ VISA

Card # ______ Expiration Date ______

Signature ______

Subscription Rates

	USA	Canada /Mexico
ONE YEAR:	☐$21	☐$23
TWO YEARS:	☐$38	☐$42
THREE YEARS:	☐$55	☐$61

☐$69 Europe (1 year-air delivery)

☐$37 Europe and worldwide (1 year-surface mail)

Air mail rates outside Europe available upon request.

Please allow 6–8 weeks for processing. Thank you.

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS MAIL PERMIT NO. 39 MARTINSVILLE, NJ

POSTAGE WILL BE PAID BY ADDRESSEE

BYTE Subscriptions
PO Box 597
Martinsville, NJ 08836-9956
USA

For your own subscription to BYTE, complete this card and mail.

46SRSX

Name ____________

Address ____________

City ____________

State ________ Zip ________ Country ________

☐ Bill me (U.S.A., Canada, Mexico only)

☐ Check enclosed (All checks must be payable in U.S. funds drawn on a U.S. bank.)

Charge to my ☐ MasterCard ☐ VISA

Card # ____________ Expiration Date ________

Signature ____________

Subscription Rates

	USA	Canada /Mexico
ONE YEAR:	☐$21	☐$23
TWO YEARS:	☐$38	☐$42
THREE YEARS:	☐$55	☐$61

☐$69 Europe (1 year-air delivery)

☐$37 Europe and worldwide (1 year-surface mail)

Air mail rates outside Europe available upon request.

Please allow 6–8 weeks for processing. Thank you.

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS MAIL PERMIT NO. 39 MARTINSVILLE, NJ

POSTAGE WILL BE PAID BY ADDRESSEE

BYTE Subscriptions
PO Box 597
Martinsville, NJ 08836-9956
USA

Make a career move!

Get your *own* copy of BYTE!

Sharing is nice but not practical when you have to stay on the leading edge of microcomputer technology. From each BYTE issue theme to "MICROBYTES" and "Best of BIX", no other magazine—or group of magazines—can deliver you the vital technical information packaged in one issue of BYTE.

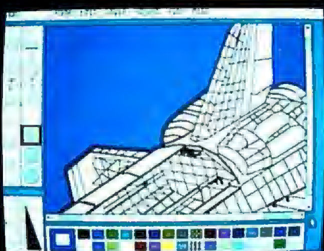
Don't rely on others for BYTE information—waiting on-line can be hazardous to your job. Make a career move! Have BYTE delivered to *your* door in *your* name. Use one of the adjacent cards to ensure your position in microcomputer technology.

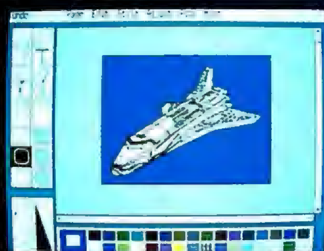
For direct ordering call toll free weekdays 8:30am–4:30pm E.S.T.: 800-258-5485 (603-924-9281 for New Hampshire residents).

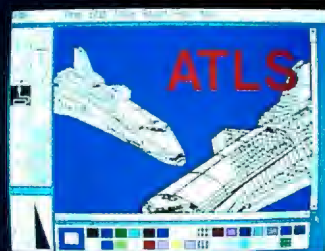
Welcome to BYTE Country!

Unlock your desktop with Publisher's Paintbrush.™

Finally, a paint package designed for desktop publishing! When you create or scan a 300 dpi page, you'll get a 64-screen computer image. That's why Publisher's Paintbrush lets you zoom out and work on the big picture. So you get ultra-sharp resolution without ultra-tedious labor.


One screen of a 64-screen image.

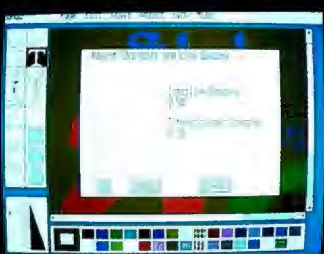

Shrink-down to a manageable size.

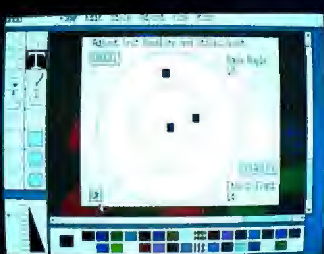


Next, cut-and-paste . . .

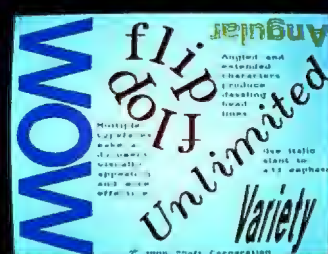
A few edits later . . . VOILA!

New typography frontiers: continuously adjustable point sizes, text slope, line and character spacing, extended and condensed type, and letter slant.


Easy-to-use menus.

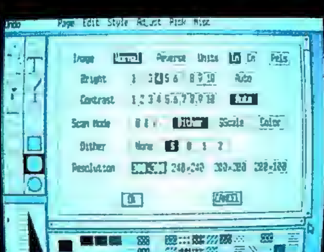

Slant and angle text 1° – 359°.

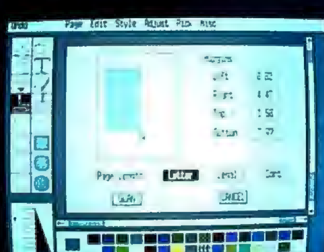
4- to 250-point type.

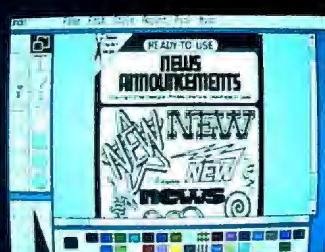


Adjustable everything!

Publisher's Paintbrush adapts to most scanners with absolute simplicity. Scan in existing art, logos and diagrams to save many hours of work!


Full use of scanner features.


Scan all or part of a page.



Combine art and type . . .

. . . from several sources!

Publisher's Paintbrush is a quantum leap in power beyond our top-selling PC Paintbrush®. In fact, it's the dawn of a new era for desktop publishing. With it, you can produce pictures of super-high resolution (many times sharper than your screen!) and marry them into text pages. With it, you can take full advantage of desktop publishing packages, laser printers and image scanners. Without it, you're stuck with ragged edges and tedious multi-screen editing.

Publisher's Paintbrush supports major-brand image scanners and hundreds of printers and video display boards. And most desktop publishing packages have built-in links to Publisher's Paintbrush.

Say goodbye to the old boundaries. Say "Show me" to your ZSOFT dealer. Or call our corporate offices for more information.

Z-SOFT

1950 Spectrum Circle, Suite A495, Marietta, Georgia 30067, (404) 980-1950

Inquiry 310

ARE YOUR DOCUMENTS ALIEN?

If capturing text or images from paper documents is alien to you, our ***SCANNING CONNECTION*** is your best plan of attack.

In less than one minute, an image of your document can be read, stored, retrieved, processed, and output to a laser printer.

A version of Halo DPE® is provided for processing images. Photographs, books, articles, and drawings can be scanned and saved in compressed form.

The saved images can be combined with text in a wide variety of fonts and sizes. The completed document can be reproduced on a laser printer.

Images can be processed with our OCR software to produce text documents. Enhanced matrix matching is used for accurate character recognition. After the system is trained for each new typestyle, the matrix pattern is saved and reused.

A text editor is used to locate and correct scanning errors. The final document is stored in IBM DCA format with tabs, margins, indents, and line ends. The IBM DCA format can be read by most popular word processing programs.

We specialize in conversion systems and can transport your documents or images to most computer, typesetting, or word processing systems.

With our ***DISKETTE CONNECTION*** system you can read and write most 8″, 5¼″, and 3½″ diskettes using a PC. We support systems from CPT, LANIER, NBI, WANG, XEROX, HONEYWELL, DG, DEC, and IBM.

Our ***TAPE CONNECTION*** system can read and write most 9-Track magnetic tapes using your PC. We can supply tape drives in several models at 800, 1600, or 6250 BPI. Tapes can be labeled and multivolume.

The price for a Canon IX-12 scanner, controller card, cable, and image software is **$1195.** The optional OCR software is **$595.** The price for an IBM 3117 flat bed scanner, controller card, and image software is **$1395.** The price for an 8″ diskette drive, controller card, cable, and utility software is **$1195.** The optional software for word processing conversion is **$695.** The price for a 9-Track 1600 BPI tape drive, controller card, cable, and software is **$3495.**

Since 1982, we have installed thousands of diskette and tape conversion systems at customer locations around the world. We have written hundreds of software programs and designed dozens of hardware products to meet our customer's conversion requirements. We support our products because we developed them. Call us today for help in solving your conversion problems.

Flagstaff Engineering

1120 W. Kaibab • Flagstaff, AZ 86001 • 602-779-3341 • telex 705609

In Europe contact Compusol
12 Rue Rosenwald • 75015 Paris
tel 530.07.37 • telex 205431F

Inquiry 98 for End-Users. Inquiry 99 for DEALERS ONLY.