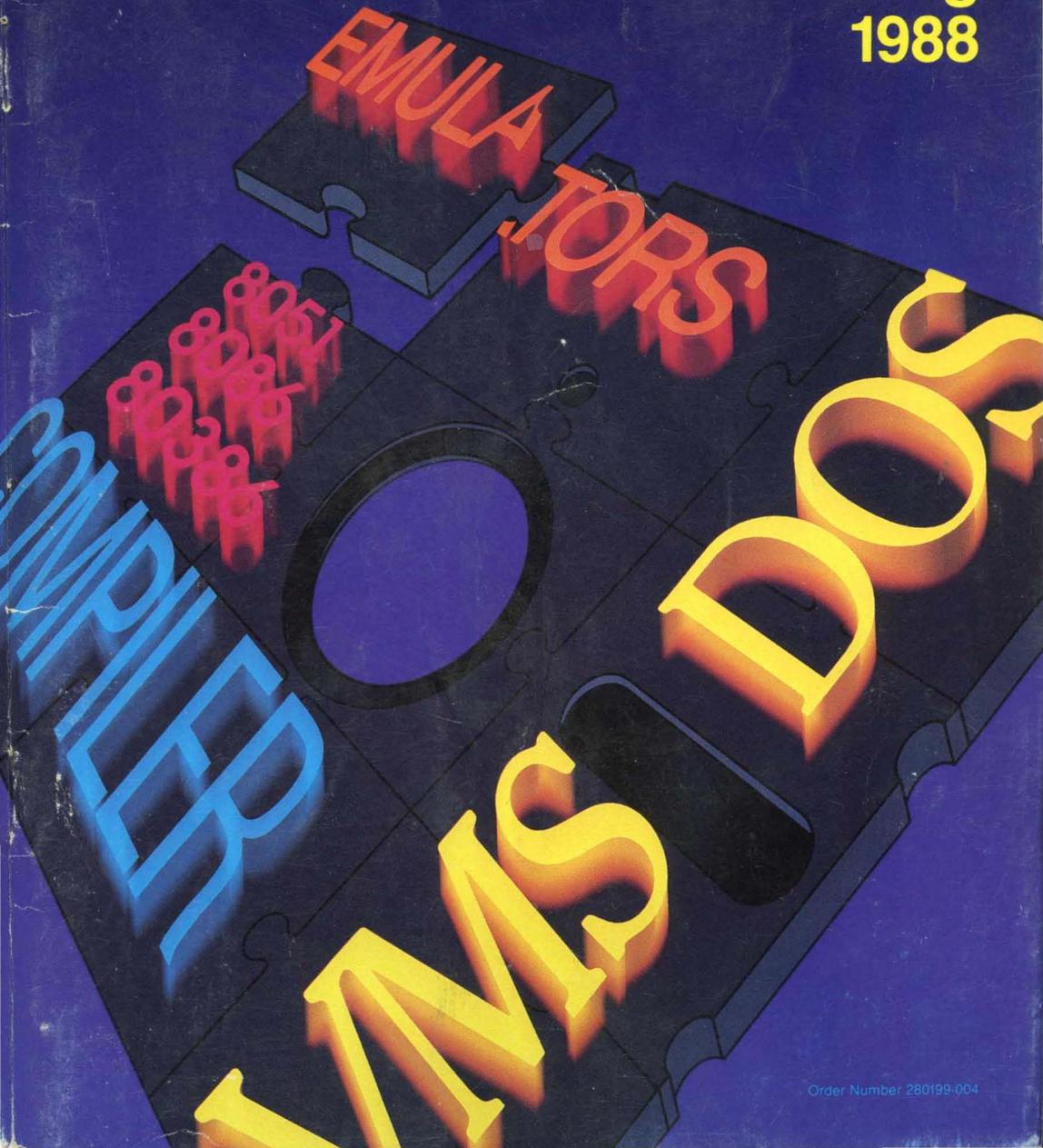


intel<sup>®</sup>

Development  
Tools  
Catalog  
1988



Order Number 280199-004

# Table of Contents

**NOTES:** This Spring 1988 reprint identifies price changes with an "\*" where applicable. Changes in product descriptions are noted by (UPDATED) here and on the page corner.

The notation "(SEE ADDENDUM)" refers to additional products offered within the particular architecture. This reference occurs both on this page and the following Product/Environment Reference page.

<b>Introduction</b>		
Product/Environment Reference		3
An Introduction To Software Development		4
System Requirements		8
<b>80960 Microprocessor Tools (SEE ADDENDUM)</b>		
<b>8086, 80286, 80386 Microprocessor Tools</b>		
Assemblers and Relocation/Linkage Packages		9
PL/M Compilers		13
C Compilers	(UPDATED)	15
FORTRAN Compiler		17
Pascal Compilers		18
Software Debugger		19
Performance Analysis Tools		20
Emulators	(SEE ADDENDUM)	22
Above Board Kits	(SEE ADDENDUM)	24
<b>80376 Microprocessor Tools (SEE ADDENDUM)</b>		
<b>8051 Microcontroller Tools</b>		
Assembler and Relocation/Linkage Package		25
PL/M Compiler and Relocation/Linkage Package		26
Micro/C-51 Compiler (Micro Computer Control Corporation)		27
Emulators		28
<b>8096 Microcontroller Tools (SEE ADDENDUM)</b>		
Assembler and Relocation/Linkage Package		30
PL/M Compiler and Relocation/Linkage Package		31
C Compiler and Relocation/Linkage Package		32
<b>Productivity Tools</b>		
Text Editor		33
Program Management Tools		34
Literature		36
<b>Support Services</b>		
Software	(UPDATED)	37
Hardware		38
Training Workshops	(UPDATED)	39
<b>How to Order</b>		40
<b>Response Subscription Card</b>		

This catalog provides a convenient way to order many of Intel's development tools. It lists programming tools for the PC, VAX, and MicroVAX hosts, as well as debugging and performance analysis tools. For information on other Intel products ask us for the telephone number of your nearest Intel sales office or distributor.

©INTEL, 1988

Intel assumes no responsibility for the use of any circuitry other than circuitry embodied in an Intel product. No other circuit patent licenses are implied. Intel retains the right to make changes to specifications at any time without notice. Intel Corporation makes no warranty for the use of its products and assumes no responsibility for any errors which may appear in this document nor does it make a commitment to update the information contained herein.

It is assumed that each software title listed in this book is a trademark. The following trademarks from other companies were explicitly used: Compaq, Deskpro PVCS, TIPS, VAX, VMS, MicroVAX, REALMATH, IBM PC/AT, IBM PC/XT, MS-DOS, IBM, Relocation, Polytron, Polymake, Microsoft, MS-LINK, DOS PAK, MICRO/C-51.

The following are trademarks of Intel and its affiliates and must be used only to identify Intel products: OpenNET, Above, Intellec, ICE, I<sup>2</sup>ICE, iRMX, RUPI, UPU, MULTIBUS, MCS, 386, 376, and 960.

See Addendum for 80960 and 80376 DOS-hosted development tools.

## Product/Environment Reference

Product	IBM PC/XT/AT (DOS)	VAX (VMS)	MicroVAX (VMS)
<b>8086/88/186/188</b>	Page	Page	Page
ASM/Utilities	9	9	9
PL/M	13	13	13
C	15	15	15
FORTTRAN	17	17	17
Pascal	18		
Text Editor (AEDIT)	33		
PSCOPE	19		
Performance Analysis Tools (iPAT)	20		
Emulators (I <sup>2</sup> ICE)	22		
Emulator (ICE-186)	addendum		
Above Board Kits	24		
Inboard Kits	addendum		
<b>80286</b>			
ASM/Utilities	10	10	10
PL/M	13	13	13
C	15	15	15
Pascal	18		
Text Editor (AEDIT)	33		
Performance Analysis Tools (iPAT)	20		
Emulators (I <sup>2</sup> ICE)	22		
Emulators (ICE-286)	addendum		
Above Board Kits	24		
Inboard Kits	addendum		
<b>80386</b>			
ASM	11	11	11
RLL (Utilities)	11	11	11
PL/M	14	14	14
C	16	16	16
Text Editor (AEDIT)	33		
Emulators (ICE-386)	addendum		
Above Board Kits	24		
Inboard Kits	addendum		
<b>8051</b>			
ASM/Utilities	25		
PL/M/Utilities	26		
Micro/C-51	27		
Text Editor (AEDIT)	33		
Emulators (ICE-5100)	28		
<b>8096</b>			
ASM/Utilities	30		
PL/M/Utilities	31		
C/Utilities	32		
Text Editor (AEDIT)	33		
Emulators (VLSiICE-96, ICE-196 PC)	addendum		
<b>Productivity Tools</b>			
Program Management Tools	34		
Literature	36	36	36
<b>Support Services</b>			
Software	37	37	37
Hardware	38	38	38
Training Workshops	39	39	39

# An Introduction To Software Development

## **What tools should you use for an embedded application?**

Perhaps this is a question you are struggling with right now. You're a professional whose livelihood depends on developing high-quality products using embedded microprocessors or microcontrollers—working within budget and on schedule. You expect your tools to help you get the maximum performance from your product. You need to be able to symbolically test, analyze, and debug your code executing in real-time in a prototype system. You demand high-quality tools. And you depend on support from trained applications specialists. Who can you turn to? Intel, inventor of the microprocessor, provides development tools that can satisfy all of these requirements and more.

Intel's development tools include compilers, assemblers, linkers, emulators, and software analyzers and debuggers. All of these products have been **designed** to work together to maximize your productivity. We don't just want to sell you development tools—we want you to create a higher-quality product ahead of schedule. Many companies offer partial solutions to the embedded application puzzle, but nobody offers as complete a solution as Intel, and no other vendor has as big a stake in your success as Intel.

## **Use Intel microcomputers and microcontrollers to their full potential.**

It makes sense to build systems with state-of-the-art microprocessors and microcontrollers using premier development tools. Intel's development tools will allow you to unleash the full power of the silicon in applications built around Intel microprocessors and microcontrollers, because Intel designs both the components and the tools. The tools are stable and reliable, because Intel has been in the development tool business

since it marketed the world's first microprocessors.

## **Place your code easily into ROM**

Developing an embedded system means that eventually you'll need to transfer the code you write into ROM for shipment with your systems. Intel's development tools are designed so that you can make your code ready for ROM and then easily transfer to ROM programming devices.

## **Produce object code with full symbolic information for state-of-the-art debugging.**

Unlike many compilers, Intel compilers produce object code that contains a complete set of object records. In addition to the object records required for program execution, Intel object code contains debugging records that describe each line and symbol in your source program. This allows Intel's debugging tools, such as PSCOPE, I<sup>2</sup>ICE, ICE (-186, -286, -386, -376, -5100, and -196PC), and VLSiCE-96 to track down bugs in your code while showing you the actual symbols and variable names you defined in your source code. The debugging records also let the iPAT performance analysis tools identify real-time performance bottlenecks and code coverage symbolically.

Once your program is fully debugged and optimized, you can recompile your program without the debugging records to minimize the program's file size.

## **Gain high speed with numeric coprocessors.**

Intel translators support the REALMATH IEEE floating-point standard for consistent and reliable floating-point operations. These operations can be performed with software, or with Intel's high-speed 8087, 80287, or 80387 numeric coprocessors.

## **Gain flexibility with separate compilation, incremental linking, and a complete range of tools.**

With Intel tools, you can work the way you want to work, not the way some program forces you to work.

- All Intel compilers allow you to break your programs into separately-compiled modules, so that you can debug individual modules, and so that minor code changes don't imply lengthy recompilations.
- Intel program linkage tools let you build your application programs incrementally, combining and testing modules as you create them. In this way, you can link new modules to a stable program base as those modules become available.
- Intel offers many different programming languages, all of which produce compatible **object** code. For example, to support the 8086 family, Intel offers assembly language, C, PL/M, and FORTRAN for development under both DOS and VAX/VMS. Code from different languages can be mixed, and you can even mix memory models. This allows different members of your design team to choose the most effective combination of languages and systems to get the job done faster.
- Intel offers a complete set of development tools. In addition to programming languages, Intel offers editors, debuggers, in-circuit emulators, and performance analysis tools.

## **Ensure upward compatibility with future processors.**

Intel development tools provide an easy migration path to future generations of Intel processors. For example, Intel's 8086, 80286, and 80386 languages are upwardly source compatible. In addition, Intel's 80286 and 80386 languages are upwardly **object** code compatible. Therefore, you can link existing 80286 modules directly with 80386 modules by using Intel's 386 Relocation and Linkage

tools. This type of upward compatibility preserves your investment in existing software, improves your productivity, and helps you beat your development schedules.

## **Do your development on the host of your choice.**

All of the products in this catalog are available in versions that run on DOS-based personal computers (IBM PC XT's, AT's, and 100% compatible computers). Many VAX/VMS and MicroVAX/VMS versions are also listed. Although not described in this catalog, many of the tools are also available in iRMX versions as well. Because the tools are available for a variety of hosts, you can choose the host environment that best fits the size of your project and the experience of the developers. Compatible versions of software on different hosts also means that you can use a mixed-host development environment to more fully utilize your existing computer equipment or ease the transition to a new environment.

## **Get thorough, professional support to back up these high-quality products.**

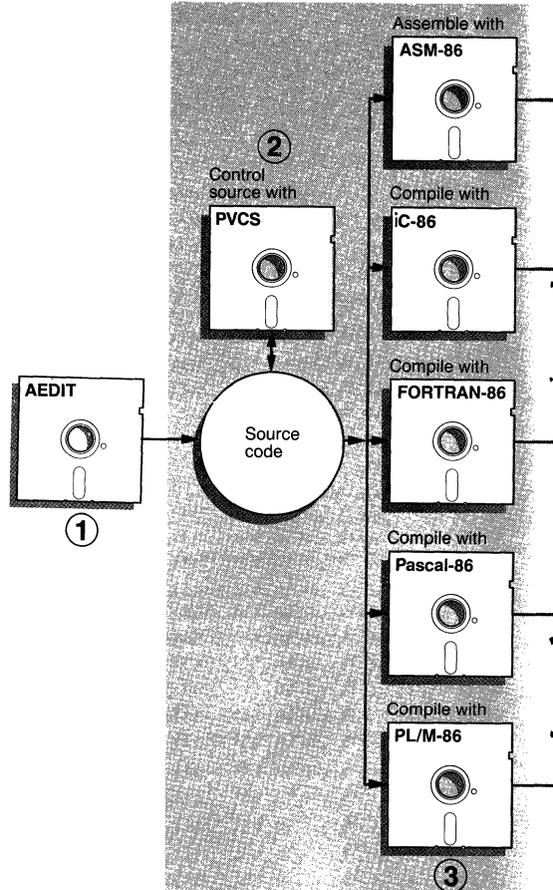
All Intel products are backed by a professional support organization including 90 days post-purchase support and a variety of extended support agreements. This software support is available in multiple configurations to best fit your development environment and covers technical publications, special troubleshooting guides, and technical information telephone hotline service. Extended hardware support is available to cover your development equipment with complete on-site maintenance in most cases.

Intel also offers comprehensive training workshops dealing with both software and hardware aspects of Intel microcomputers. These workshops are given regularly in major cities across the country by expert technical instructors.

## Here's how the development process works using the tools in this catalog.

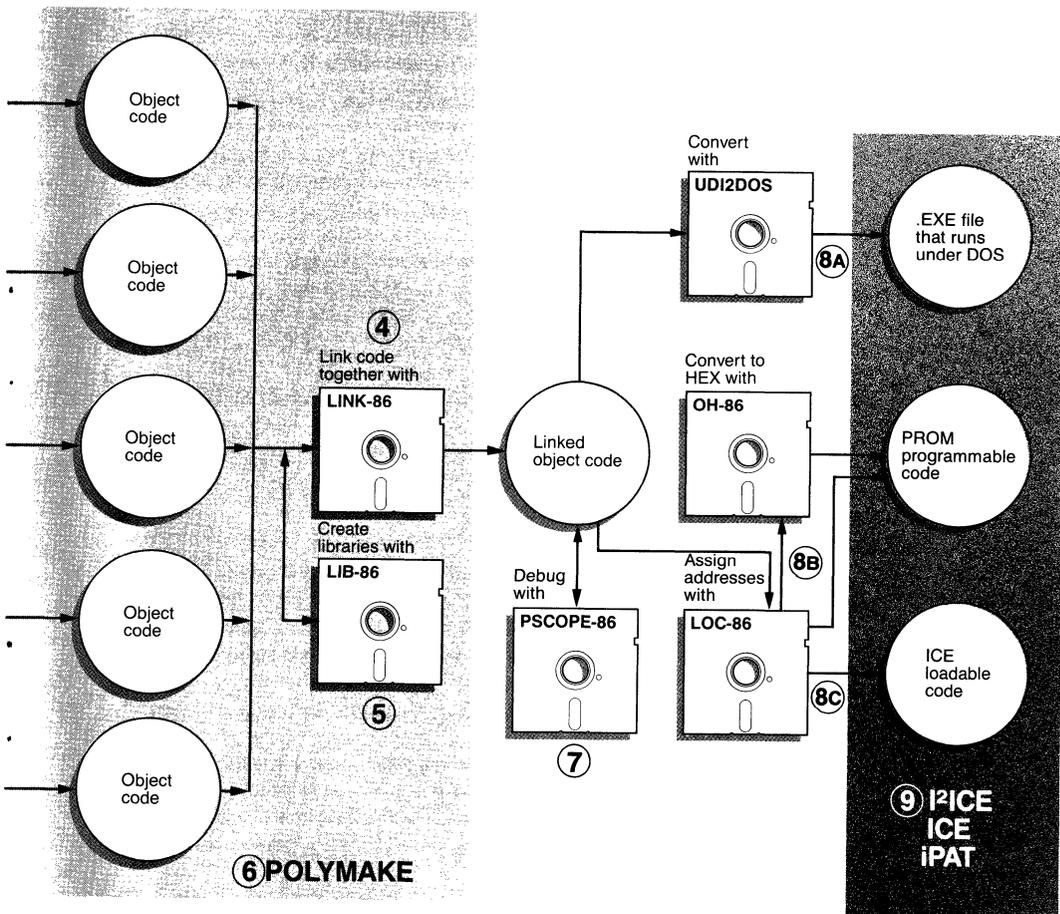
To give you an idea of how easy it is to develop programs using the tools listed in this catalog, let's run through the process step by step. These steps, illustrated in the accompanying figure, assume you want to develop 8086 code for an embedded component application using an IBM PC/XT or PC/AT for code development.

1. The first step is creating the source code. When doing this, you can use any editor that generates pure ASCII files. Intel's AEDIT program editor is a good choice because it is screen-oriented. It lets you concentrate on what you are writing, not on how to use the editor. Because Intel tools allow separately-compiled code, you can break up your programming application into several source files. If you wish, you can code parts of your application in different languages.
2. Use the POLYTRON Version Control System to store and track multiple versions of source code so that you always have a complete product history.
3. Invoke the assembler or the appropriate compiler to translate the source code into object code. Intel has 8086 compilers for C, FORTRAN, Pascal, and PL/M.
4. Once you finish assembling and/or compiling your code, use LINK-86 to link your 8086 program modules together. LINK-86 can produce a relocatable module, a module for incremental linking, or it can simply link code together in preparation for placing the code into ROM.
5. Instead of linking all your code together immediately, you might want to use LIB-86 to organize your object modules into libraries. When you link a program to a library, the linker extracts only those library modules that the program needs. Libraries are good places to store commonly-used procedures.
6. To automate your program development process, use POLYTRON's PolyMake to



easily and quickly generate the latest version of your program. Or use an older version of the source stored in the POLYTRON Version Control System to generate linked object code according to the rules that you specify.

7. After producing linked modules, invoke the PSCOPE 86 debugger to test and debug your program. PSCOPE 86 helps you locate obvious and subtle bugs by letting you step through and examine your code in the high-level language you used originally.
8. You will now want to load the code into system memory for execution and final debug. There are three ways to do this,



depending on your situation. You can convert it to DOS format, load it with an in-circuit emulator, or program it into PROM.

- 8a. You can choose to make your 8086 code ready to run under DOS by invoking the UDI2DOS conversion program. This program simply adds the DOS runtime support to the relocatable object module produced by LINK-86. The resulting file is an .EXE file that you can invoke directly from DOS.
- 8b. Instead of converting the module to DOS format, invoke LOC-86 to assign absolute addresses to the object module generated by LINK-86. This absolute code can be burned into

PROM using Intel PROM Programmers, or you can invoke OH86 to convert the absolute code into hexadecimal format as required by some PROM programming equipment.

- 8c. The code generated by LOC-86 can also be downloaded to your target system by using an in-circuit emulator such as I<sup>2</sup>ICE (Integrated Instrumentation and In-Circuit Emulation System).
- 9. Use Intel's I<sup>2</sup>ICE or ICE to do emulation and symbolic debugging. Use Intel's Performance Analysis Tool (IPAT) for symbolic performance and code coverage analysis. Both are real-time, non-intrusive, transparent tools for software development.

# System Requirements

Intel supplies development tools for several different development environments, including DOS, VAX/VMS and MicroVAX/VMS. To run these tools, the development system must be appropriately configured.

## DOS

Running the DOS versions of Intel's development tools requires an IBM Personal Computer XT, AT, or 100%-compatible computer, with at least 512K bytes of RAM and DOS 3.0 or later.

Programs generated by the 8086 development tools can run directly under DOS. A utility called UDI2DOS adds the DOS runtime support to linked 8086 object files, transforming them into .EXE files that are suitable for execution on the PC. The 8086 development tools can also be used to develop code for an 80286 or 80386 processor running in real mode. For example, Intel's 8086 languages produce code which can execute on an IBM AT (which contains an 80286) or a Compaq Deskpro 386 (which contains an 80386) running DOS 3.0.

Intel's 80286 and 80386 development tools can be used to develop software for the 80286 and 80386 processors running in protected mode. The 80286, 80386, and 80376 assembler, compilers, and relocation and linkage packages can be invoked from DOS, allowing you to use your PC as a program development station. Executing and debugging your 80286, 80386, and 80376 programs requires an appropriate target environment or one of Intel's real-time emulators, such as I<sup>2</sup>ICE and probes or an ICE emulation system.

Programs generated by the other development tools (the 8051 and 8096 packages) run only in an environment that contains those microcontrollers. The development tools will run on your PC, but the code they generate will not. Emulators for these microcontrollers, such as the ICE-5100, VLSiCE 96, or ICE-196 PC enable you to debug your microcontroller applications.

## VAX/VMS and MicroVAX/VMS

The VAX versions of Intel's development tools require a Digital Equipment Corporation VAX (11/7xx or 8xxx) and VMS version 4.4 or later.

The MicroVAX versions of Intel's development tools require a Digital Equipment Corporation MicroVAXII and VMS version 4.4 or later.

Code generated on a VAX uses Intel object module format. This object code must be executed in an environment containing the appropriate Intel microprocessors. Downloading can be accomplished in numerous ways, including using Intel's high-performance OpenNet™ network to connect VAXs with PCs. The OpenNet network permits code to be downloaded to the In-Circuit Emulator and Performance Analysis Tools hosted on PCs.



The Assemblers and Relocation/Linkage packages provide the tools that assembly-language programmers need to maintain complete control over the 8086, 80186, 80286, and 80386 microprocessors. The assembly languages are strongly typed, providing extensive checks on variables and labels. This helps catch many programming errors long before the debugging cycle. Macro facilities are also available to speed and simplify your work.

The relocation and linkage packages make your programs ready to run. They link programs together, assign absolute addresses, gather modules into libraries, and perform other system functions.

Assembler 86, 286 and 386:

- Are highly mnemonic and compact, and are strongly typed to detect errors at assembly time
- Place high-level symbolic information in object modules to enable symbolic debugging
- Have powerful text macro facility with three macro listing options, including string functions, and can expand conditional assembly pseudo-ops

The Relocation/Linkage Packages:

- Resolve PUBLIC/EXTERNAL references and perform intermodule type checking
- Select required modules from libraries to satisfy symbolic references and provide fast, easy management of object module libraries

- Simplify debugging by producing detailed maps that show references between program modules

---

## ASSEMBLER 86, RELOCATION/ LINKAGE 86 PACKAGE

---

This package is a complete development package for assembly-language programmers. It contains the 8086 Assembler and an extremely rich set of linking/locating utilities known as R & L 86.

---

### ASSEMBLER 86

---

- Produces standard Intel object modules that can be compiled separately, linked to programs written in any Intel 8086 language, and debugged with the PSCOPE 86, I2ICE, and ICE debuggers

---

### RELOCATION/LINKAGE 86 (R & L 86)

---

- Combines modules produced by Intel 8086 assembler, compilers, and librarian into a single relocatable object module and supports incremental linking
- Offers controls to generate the appropriate modules for loading into PSCOPE 86, I2ICE, or ICE for debugging, or for loading into ROM as absolute code
- Manipulates debug symbol information to enable line numbers, local symbols, and public symbols to be purged and listed selectively
- Provides library of routines that emulate 8087 floating-point operations
- Supports multiple overlays, up to the physical memory limitation of your target system
- Transforms linked object files into .EXE files for execution under DOS (DOS version only)
- Converts 8086 absolute object modules into hexadecimal format to facilitate file-loading by hexadecimal loaders (such as PROM programmers)

Assembler 86 produces code for the complete family of 8086 microprocessors, including the 80286, 80386, and 80376.

However, to take advantage of the protected mode features of the 80286 or 80386, you should use Assembler 286 or Assembler 386.

The Relocation/Linkage 86 package includes several tools. LINK-86 combines program modules and provides incremental linking, optional inclusion of symbolic debugging information, and generation of summary maps. LIB-86 gathers modules into libraries where they can be accessed individually by LINK-86. CREF-86 is a cross-reference utility that shows references between modules.

The 8087 emulation library enables programs to perform floating-point operations without requiring an 8087 or 80287 coprocessor. Library functions convert to and from decimal or binary, and they provide several error-handling features. Common elementary functions include complex numbers, logarithmic and exponential functions, trigonometric and hyperbolic functions, and more.

The UDI2DOS tool is included with the DOS version of the package and adds the DOS runtime support to linked modules, converting them into .EXE files. Two other tools, LOC-86 and OH-86, facilitate placing programs into ROM by assigning program addresses and converting the resultant object code into hexadecimal form.

#### ORDERING & PRICE INFORMATION

DOS	D86ASM86NL	\$750
	Quantity 2-4	\$650
	Quantity 5-9	\$575
	Quantity 10 +	\$495
MicroVAX/VMS	MVVSASM86*	\$2,500
VAX/VMS	VVSASM86*	\$5,000
DOS Kits	D86PAK86NL	\$1,395

Assembler 86 and Relocation/Linkage Package, PSCOPE 86 debugger (pg. 26), and Aedit text editor (pg. 33)

Assembler 86 and Relocation/Linkage Package with Above Board (pg. 24).

\*New pricing data.

## ASSEMBLER 286, RELOCATION/ LINKAGE 286 PACKAGE

This package is a complete development package for assembly-language programmers who need to produce system software that runs in protected mode of the 80286. It includes the 80286 Assembler and an extremely rich set of linking/locating utilities known as R & L 286.

### ASSEMBLER 286

- Produces Intel 80286 object modules that can be assembled separately and linked to programs written in any Intel 80286 language
- Supports the full instruction set of the 80286, including memory protection and numerics
- Supports addressing in the full 16 Megabyte physical address space provided by the 80286 processor
- Has instruction set and assembler mnemonics that are upward compatible with Assembler 86

### RELOCATION/LINKAGE 286 (R & L 286)

- Combines modules produced by Intel 80286 assembler, compilers, and librarian into executable tasks and supports incremental linking
- Assigns addresses to tasks in the 16 Megabyte physical address space
- Facilitates creation of protected, multitasking systems
- Supports memory protection by building system tables, initializing tasks, and assigning protection rights to segments
- Generates print files containing system maps and listings of errors and warnings

Assembler 286 creates individual program modules that can be linked with other OMF-286 compatible modules and debugged with powerful Intel debuggers such as I<sup>2</sup>ICE or ICE. To preserve your

software investment, you can also link OMF-286 code with OMF-386 compatible modules by using Intel's 386 Relocation, Linkage, and Library tools.

The Relocation/Linkage 286 package contains several tools. BIND-286 is a program linker that also can assign addresses in the 16 Megabyte physical address space or the 1 Gigabyte virtual address space. MAP-286 is a cross-reference utility that shows references between modules. LIBRARIAN-286 gathers modules into a library where they can be accessed individually by BIND-286. And finally, BUILD-286 configures multitasking, protected systems by defining system tables, initializing tasks, assigning protection rights, and configuring the memory image. The modules produced by BUILD-286 can be placed into ROM by any ROM programmer that understands the OMF-286 format.

#### ORDERING & PRICE INFORMATION

<b>DOS</b>	<b>D86ASM286NL</b>	<b>\$750</b>
	Quantity 2-4	<b>\$650</b>
	Quantity 5-9	<b>\$575</b>
	Quantity 10 +	<b>\$495</b>
<b>MicroVAX/VMS</b>	<b>MVVSASM286*</b>	<b>\$2,500</b>
<b>VAX</b>	<b>VVSASM286*</b>	<b>\$5,000</b>
<b>DOS Kits</b>		
Assembler 286 and Relocation/ Linkage Package with Above Board (pg. 24).		

\*New pricing data.

#### ASSEMBLER 386

This package is the macro assembler for the 80386 assembly language.

- Produces Intel 80386 object modules that can be assembled separately and linked to programs written in any Intel 80386 language and debugged at the source level with Intel's ICE 386 debugging system
- Supports the full instruction set of the 80386 and 80376, including memory protection and numerics

- Supports addressing in the full 4 Gigabyte physical address space provided by the 80386 and 80376 processors
- Has instruction set and assembler mnemonics that are upward compatible with ASM 286 and ASM 86

The Assembler 386 package gives programmers complete control over the powerful 80386 and 80376 microprocessors. The assembler mnemonics are a superset of the Assembler 286/86 mnemonics, with new mnemonics added to support the new 80386 instructions. The segmentation directives have been greatly simplified.

#### ORDERING & PRICE INFORMATION

<b>DOS</b>	<b>D86ASM386NL*</b>	<b>\$600</b>
<b>MicroVAX/VMS</b>	<b>MVVSASM386</b>	<b>\$2,295</b>
<b>VAX/VMS</b>	<b>VVSASM386*</b>	<b>\$4,500</b>
<b>DOS Kits</b>		
Assembler 386 with Above Board (pg. 24).		

\*New pricing data.

#### RELOCATION/LINKAGE 386 PACKAGE

This package is a comprehensive set of software design aids for programming the 80386 and 80376 microprocessor systems.

- Combines modules produced by Intel 80386 assembler, compilers, and librarian into executable tasks and supports incremental linking
- Assigns addresses to tasks in the 4 Gigabyte physical address space
- Facilitates creation of protected, multitasking systems
- Supports memory protection by building system tables, initializing tasks, and assigning protection rights to segments
- Generates print files containing system maps and listings of errors and warnings

The Relocation/Linkage 386 package enables system programmers to design protected, multiuser and multitasking operating system software, and it enables application programmers to develop tasks to run on protected systems.

The package consists of several tools: BIND-386, BUILD-386, MAP-386, and LIBRARIAN-386.

BIND-386 combines 80386 object modules into executable tasks, resolves external symbol references, combines segments, and performs address fixups on code and data. It can generate a loadable module (for execution or debugging), or a linkable module that can be input into BIND-386 later (a process called incremental binding). BIND-386 also accepts library modules, linking only those modules that resolve external references. In addition, BIND-386 can combine 80286 object modules with 80386 object modules, preserving software investment and ensuring upward migration.

BUILD-386 configures multitasking, protected systems from an operating system and discrete tasks. It accepts input modules from 80386 language translators and from BIND-286 and BIND-386. It also accepts a build file containing definitions and initial values for descriptor tables, gates, segments, and tasks. BUILD-386 generates a loadable or bootloadable output module by assigning absolute addresses. These output modules can be placed into ROM by any ROM programmer that understands the OMF-386 format. MAP-386 is a cross-reference utility that lists symbolic information in one easy-to-read place. It simplifies debugging by showing references between program modules.

LIBRARIAN-386 gathers modules into libraries where they can be accessed individually by BIND-386. LIBRARIAN-386 contains commands for creating libraries, as well as adding, replacing, copying, saving, and displaying module information.

#### ORDERING & PRICE INFORMATION

DOS	D86RLL386NL	\$600
MicroVAX/VMS	MVVSRL386	\$2,295
VAX/VMS	VVSRL386*	\$4,500
DOS Kits		
Relocation/Linkage 386 with Above Board (pg. 24).		

\*New pricing data.



The PL/M language is a structured, medium-level language created specifically as a system development language for Intel microcomputers. It provides the advantages of a high-level language with the power of assembly language. PL/M is an excellent alternative to C in, for example, I/O-intensive applications. Like the Assembler, PL/M does not require a run-time environment and thus can produce highly optimized code. In 80286 and 80386 systems, PL/M is ideal for developing multi-user, multitasking, virtual-memory operating systems to run in protected mode.

PL/M programs consist of a series of modules, procedures, and blocks. The program statements naturally express the program algorithm, freeing the programmer to concentrate on the logic of the program without concern for the details of assembly language. It is easy to learn and use, yet it allows complete access to the processor and it produces code whose efficiency rivals that of assembly language.

The following characteristics are common to each PL/M package listed here:

- Produces code whose efficiency rivals that of assembly language
- Has a block-structured syntax that encourages program modularity
- Requires fewer source statements than any other high-level language
- Has built-in syntax checker
- Allows foreign character sets in comments and strings

- Object code across multiple hosts is identical

---

## PL/M 86

---

- Can fully access the 8086 architecture
- Produces standard Intel 8086 object modules that can be compiled separately, linked to programs written in any Intel 8086 language, and debugged at the source level with the PSCOPE 86, I<sup>2</sup>ICE, and ICE debuggers
- Supports IEEE floating-point math with 8087 and 80287 coprocessors
- Supports SMALL, COMPACT, MEDIUM, and LARGE models of segmentation
- Allows nested structures
- Has type-casting capabilities

The PL/M 86 compiler can separately compile individual program modules, producing code that can be linked with other OMF-86 compatible modules. Because the object code can contain the full complement of Intel object records, you can use powerful Intel high-level-language debuggers such as PSCOPE 86, I<sup>2</sup>ICE, and ICE for source-level debugging.

### ORDERING & PRICE INFORMATION

DOS	D86PLM86NL	\$750
	Quantity 2-4	\$650
	Quantity 5-9	\$575
	Quantity 10 +	\$495

MicroVAX	MVVSPML86*	\$3,000
VAX	VVSPLM86*	\$6,000

#### DOS Kits

PL/M 86 with Above Board (pg. 24).

\*New pricing data

---

## PL/M 286

---

- Provides an advanced, structured system implementation language for protected mode of the 80286
- Is upward compatible with PL/M 86, assuring software portability

- Supports the 16 MB physical address space of the 80286 and enables programs to use other 80286 features, such as protection and virtual memory
- Produces Intel 80286 object modules that can be compiled separately, linked to programs written in any Intel 80286 language, and debugged at the source level with Intel's I<sup>2</sup>ICE and ICE debugging system
- Supports IEEE floating-point math with the 80287 coprocessor

PL/M 286 is upward compatible with PL/M 86, so existing system software can be recompiled with PL/M 286 to execute in protected virtual address mode on the 80286. In addition, the language has been enhanced to support the 80286 memory management and protection features.

The PL/M 286 compiler can separately compile individual program modules, producing code that can be linked with other OMF-286 compatible modules. To preserve your software investment, you can also link OMF-286 code with OMF-386 compatible modules by using Intel's 386 Relocation, Linkage, and Library tools.

#### ORDERING & PRICE INFORMATION

<b>DOS</b>	<b>D86PLM286NL</b>	<b>\$750</b>
	<b>Quantity 2-4</b>	<b>\$650</b>
	<b>Quantity 5-9</b>	<b>\$575</b>
	<b>Quantity 10 +</b>	<b>\$495</b>
<b>MicroVAX</b>	<b>MVVSPLM286*</b>	<b>\$3,750</b>
<b>VAX</b>	<b>VVSPLM286*</b>	<b>\$7,500</b>

#### DOS Kits

PL/M 286 with Above Board (pg. 24).

\*New pricing data.

### PL/M 386

- Supports 80386 microprocessor
- Provides an advanced, structured system implementation language for protected mode of the 80386
- Is upward compatible with PL/M 286 and PL/M 86, assuring software portability

- Supports the full 4 Gigabyte physical address space of the 80386 and enables programs to use new 80386 features, such as memory paging
- Produces Intel 80386 object modules that can be compiled separately and linked to programs written in any Intel 80386 language and debugged at the source level with Intel's ICE-386 debugging system

PL/M 386 is upward compatible with PL/M 286 and PL/M 86, so existing system software can be recompiled with PL/M 386 to execute in protected mode on the 80386. In addition, the language has been enhanced to support the unique 80386 memory management and protection features.

The PL/M 386 compiler can separately compile individual program modules, producing code that can be linked with other OMF-386 compatible modules.

#### ORDERING & PRICE INFORMATION

<b>DOS</b>	<b>D86PLM386NL</b>	<b>\$900</b>
<b>VAX</b>	<b>VVSPLM386*</b>	<b>\$9,950</b>
<b>MicroVAX</b>	<b>MVVSPLM386*</b>	<b>\$5,000</b>

#### DOS Kits

PL/M 386 with Above Board (pg. 24).

\*New pricing data.

# C COMPILERS 8086, 80286, 80386

The C programming language is known for its flexibility and portability. It is a block-structured, high-level language that is ideal for developing multi-user, multitasking, virtual memory operating systems to run in protected mode of the 80286 and 80386 processors.

Intel C compilers are unique from other C compilers in that they have been designed specifically for developing embedded component applications. Intel C compilers provide three unique benefits to software developers:

1. **Built-in Functions.** Allow highly optimized code and eliminate the need for in-line assembly. With built-ins you can enable interrupts or directly control hardware I/O from the high level language.
2. **Symbolic debug information.** Intel C compilers provide extensive symbolic debug information to speed development with an ICE.
3. **ROMable code.** Loading code into ROM is a simple, one-step process with Intel C.

In addition, runtime libraries include the STDIO library, conversion routines, string manipulation routines, routines for performing 32-bit arithmetic and floating-point operations, and routines that provide an interface to the operating system.

## iC 86

- Built-in functions for automatic machine code generation
- Compliance with draft proposed ANSI C standard

- Produces Intel OMF-86 object modules that can be compiled separately, linked to programs written in other Intel languages, and debugged at the source level with the PSCOPE 86, I<sup>2</sup>ICE, or ICE debuggers
- Supports the IEEE floating-point math standard with the 8087 and 80287 coprocessors
- Supports multiple memory models such as Small, Medium, Compact, and Large

The iC 86 compiler now available is a new generation C compiler providing unparalleled performance for embedded microprocessor designs. In addition to the features above, this compiler uses state-of-the-art code generation technology. It has the ability to mix memory models with "near" and "far" pointers. iC 86 is compatible with other C compilers and PL/M providing both standard C and PL/M calling conventions and has four optimization levels.

Because the object code can contain the full complement of Intel object records, high-level-language debuggers such as PSCOPE 86, I<sup>2</sup>ICE, and ICE can be used for source-level debugging.

## ORDERING & PRICE INFORMATION

DOS	D86C86NL	\$750
	Quantity 2-4	\$650
	Quantity 5-9	\$575
	Quantity 10 +	\$495
MicroVAX	MVVSC86*	\$3,000
VAX	VVSC86*	\$6,000
DOS Kits		
iC 86 with Above Board (pg. 24).		

\*New pricing data.

## iC 286

- Built-in functions for automatic machine code generation
- Compliance with draft proposed ANSI C standard
- Supports the 16 MB physical address space of the 80286 and enables programs to use 80286 features such as protection and virtual memory

- Produces Intel 80286 object modules that can be compiled separately, linked to programs written in any Intel 80286 language, and debugged at the source level with Intel's I<sup>2</sup>ICE or ICE emulators
- Supports IEEE floating-point math with the 80287 coprocessor
- Supports multiple memory models such as Small, Medium, Compact and Large

iC 286 is upward compatible with iC 86. It can separately compile individual program modules, producing code that can be linked with other OMF-286 compatible modules. To preserve your software investment, you can also link OMF-286 code with OMF-386 compatible modules by using Intel's 386 Relocation, Linkage, and Library tools. Because the object code can contain the full complement of Intel object records, Intel high-level-language debuggers such as I<sup>2</sup>ICE and ICE can be used for source-level debugging.

<b>ORDERING &amp; PRICE INFORMATION</b>		
<b>DOS</b>	<b>D86C286NL</b>	<b>\$750</b>
	Quantity 2-4	\$650
	Quantity 5-9	\$575
	Quantity 10 +	\$495
<b>MicroVAX</b>	<b>MVVSC286*</b>	<b>\$3,750</b>
<b>VAX</b>	<b>VVSC286*</b>	<b>\$7,500</b>
<b>DOS Kits</b>		
iC 286 with Above Board (pg. 24).		

\*New pricing data.

## **iC 386**

- Supports 80386 and 80376 microprocessors
- Produces high-density code rivaling an assembler
- Supports the full 4 Gigabyte physical address space of the 80386 and enables programs to use new 80386 features, such as memory paging

- Manipulates bit fields, pointers, addresses, and registers, enabling programs to take full advantage of the fundamental concepts of the 80386
- Produces Intel OMF-386 object modules that can be compiled separately and linked to programs written in any Intel 80386 language and debugged at the source level with Intel's ICE-386 debugging system

The iC 386 compiler supports the proposed draft ANSI C standard and the full Kernighan and Ritchie C. It can separately compile individual program modules, producing code that can be linked with other OMF-386 compatible modules. Compilation is faster than with standard C compilers because iC 386 produces object code directly, without generating intermediate assembly code.

<b>ORDERING &amp; PRICE INFORMATION</b>		
<b>DOS</b>	<b>D86C386NL</b>	<b>\$900</b>
<b>MicroVAX</b>	<b>MVVSC386*</b>	<b>\$5,000</b>
<b>VAX</b>	<b>VVSC386*</b>	<b>\$9,950</b>
<b>DOS Kits</b>		
iC 386 with Above Board (pg. 24).		

\*New pricing data.



- Features high-level support for floating-point calculations, transcendentals, interrupt procedures, and run-time exception handling
- Meets ANSI FORTRAN 77 subset language specifications
- Produces standard Intel 8086 object modules that can be compiled separately, linked to programs written in any Intel 8086 language, and debugged with the PSCOPE 86, I<sup>2</sup>ICE, and ICE debuggers
- Supports the IEEE floating-point math standard with 8087 and 80287 coprocessors
- Supports arrays larger than 64K bytes

FORTRAN has long been the industry-standard programming language for numerical processing applications. FORTRAN 86 meets the ANSI FORTRAN 77 Language Subset Specification and includes many features of the full standard. It supports single-precision (32-bit), double-precision (64-bit), double-extended-precision (80-bit), complex (two 32-bit), and double-complex (two 64-bit) floating-point data types. Floating-point operations can be performed with software or with numeric coprocessors, such as the 8087 and 80287. In addition, FORTRAN 86 has microprocessor extensions for performing direct byte- or word-oriented port I/O, developing reentrant procedures, and creating interrupt procedures.

Like the other Intel 8086 languages, the FORTRAN 86 compiler can separately compile individual program modules, producing code in the Intel OMF-86 format. This code can be linked with other OMF-86 compatible modules. Because the object code can contain the full complement of Intel object records, you can use powerful Intel high-level-language debuggers such as PSCOPE 86, I<sup>2</sup>ICE, and ICE for source-level debugging.

#### ORDERING & PRICE INFORMATION

DOS	D86FOR86NL	\$750
	Quantity 2-4	\$650
	Quantity 5-9	\$575
	Quantity 10 +	\$495
MicroVAX	MVVSFORT86*	\$3,000
VAX	VVSFORT86*	\$6,000

\*New pricing data.



- Supports IEEE floating-point math with 8087 and 80287 coprocessors

The Pascal 86 compiler can separately compile individual program modules, producing code that can be linked with other OMF-86 compatible modules.

ORDERING & PRICE INFORMATION		
DOS	D86PAS86NL	\$750
	Quantity 2-4	\$650
	Quantity 5-9	\$575
	Quantity 10 +	\$495

The Pascal compilers provide a complete implementation of the ISO proposed standard Pascal for 8086, 80186, and 80286 microprocessors. In addition, the Intel Pascal compilers contain extensions to standard Pascal that tailor the resulting code to fit microcomputer applications. There are extensions for interrupt handling and port I/O. Predefined type extensions also allow you to specify the precision of real, integer, and unsigned calculations; check errors on 8087 or 80287 operations; and circumvent the type checking on calls to non-Pascal routines.

The following characteristics are common to each Pascal package listed here:

- Offers strict implementation of ISO standard Pascal
- Contains extensions to the ISO standard that are essential for microcomputer applications
- Allows separate compilation with type-checking enforced between modules
- Has compiler option to support full run-time range-checking
- Supports large array operation
- Can use powerful Intel debuggers such as I<sup>2</sup>ICE or ICE for source level debugging

### PASCAL 86

- Produces Intel OMF-86 object modules that can be compiled separately, linked to programs written in other Intel languages, and debugged with the PSCOPE 86, I<sup>2</sup>ICE, and ICE debuggers

### PASCAL 286

- Provides a powerful, structured, applications programming language for protected mode of the 80286
- Is upward compatible with Pascal 86, assuring software portability
- Supports the 16 MB physical address space of the 80286 and enables programs to use other 80286 features, such as protection and virtual memory
- Produces OMF-286 object code that can be compiled separately, linked to programs written in any Intel 80286 language, and debugged at the source level with Intel's I<sup>2</sup>ICE or ICE debugging system
- Supports IEEE floating-point math with 80287 coprocessor

The Pascal 286 compiler can separately compile individual program modules, producing code that can be linked with other OMF-286 compatible modules. To preserve your software investment, you can also link OMF-286 code with OMF-386 compatible modules by using Intel's 386 Relocation, Linkage, and Library tools.

ORDERING & PRICE INFORMATION		
DOS	D86PAS286NL	\$750
	Quantity 2-4	\$650
	Quantity 5-9	\$575
	Quantity 10 +	\$495

# SOFTWARE DEBUGGER

## 8086

- Displays source code on the screen during program step, at execution breakpoints, or on user request
- Sets breakpoints and traces program execution
- Displays and modifies dynamic and static variables symbolically
- Maintains type information about variables by interpreting object records
- Permits creation of debug macros to extend the command language
- Permits creation of program patches using high-level debug constructs
- Disassembles memory and provides an interactive assembler
- Permits access to any 8087 register by name, either through a built-in 8087 math coprocessor emulator, or through the 8087 itself
- Provides an on-line help facility
- Single-steps through assembly-language instructions, high-level language statements, or procedures
- Provides a shell escape for access to DOS commands

In describing PSCOPE 86, the October, 1986 issue of *Computer Language* said, "PSCOPE 86 is an extremely powerful debugger that allows full access to all parts of programs written in high-level languages." This aptly describes a high-level language debugger that can debug any 8086 program written in assembly language, C, FORTRAN, Pascal, or PL/M. It uses the symbolic debug information in Intel object records, in conjunction with the program listing file, to show you the actual

source statements you used when creating your program. You can view any part of your program's source code, step through the program a single source statement at a time, and set execution breakpoints at source statements.

PSCOPE 86 contains a complete set of descriptive debugging statements and a command language that includes block constructs such as DO-END, COUNT-END, REPEAT-END, and IF-THEN-ELSE. With a syntax that is much like a high-level language, PSCOPE 86 lets you refer to memory locations symbolically, create debugging symbols, create and label groups of statements, and invoke your statements as separate debugging procedures.

You can also use these debugging statements to patch program errors without recoding, recompiling, and relinking your application. Patching is a powerful technique that lets you make nontrivial changes quickly, but lets you make them temporarily.

### ORDERING & PRICE INFORMATION

DOS	D86PSC86NL	\$995
	Quantity 2-4	\$795
	Quantity 5-9	\$750
	Quantity 10 +	\$700

DOS Kits	D86PAK86NL	\$1,395
PSCOPE 86, Assembler 86 and Relocation/Linkage Package (pg. 9), and AEDIT text editor (pg. 33)		

PSCOPE 86 with Above Board (pg. 24).



# PERFORMANCE ANALYSIS TOOLS

## 8086, 80186, 80286

- Provides real-time performance and code coverage analysis non-intrusively with 100% sampling
- Displays information using histograms or analysis tables
- Accepts specification of ranges with addresses, program symbolic names, or user-defined symbolic names
- Can disarm/arm analysis on called subroutines, external interrupts, interrupt routines, operating system functions, or any execution address or range
- Supports the 8086, 8088, 80186, and 80188 processors, as well as the 80286 processor in both real and protected modes
- Hosted serially on PC/XT and PC/AT systems
- Presents an easy-to-use human interface, including function keys and color/monochrome graphics

iPAT, Intel's Performance Analysis Tool, helps engineers control the performance and reliability of a software-driven system by showing, via histograms and tables, the real-time execution activity of software in terms of range names or addresses. iPAT is used to analyze real-time software execution in component-, board-, and system-level products based on 8086 and 80286 family of microprocessors. These include the 8086, 8088, 80186, and 80188 processors, plus the 80286 processor in both Real (86 Emulation) and Virtual (286 protected) modes.

iPAT is an instrument with which software engineers passively monitor real-time software activity non-intrusively and with 100% sampling. In some development environments, ability to interact with a component-level prototype is required. For such environments, an iPAT interface version that is integrated with either the I<sup>2</sup>CICE emulator system and probes or ICE emulator system is available. This need arises, for example, if object code must be externally downloaded through an emulator into partially-built hardware, or if inputs to drive test cases can only be created in target memory through the use of an emulator.

---

### APPLICATION EXAMPLES

---

An iPAT-286 probe, for example, can be used with an iPAT Core system to profile application performance characteristics in an 80286 protected-mode operating system environment. Developers use iPAT and an iPAT-286 probe to obtain histograms of overall performance, subroutine characteristics, and interactions with interrupt routines, external hardware, or with the operating system. They can then speed-tune the product's execution. At the test and acceptance phase, iPAT code coverage analysis enables developers to achieve a higher level of testing in less time by enabling them to determine incremental testing requirements and to document testing results.

iPAT analysis of a partially-built, 80186-based embedded control system, in contrast, requires an iPAT version with an interface to an I<sup>2</sup>CICE emulator and an I<sup>2</sup>CICE-186 emulation probe or an ICE-186 emulator. With the emulator, code is downloaded into the target, target resources are defined or stubbed, and the system is set into execution. iPAT can then profile the system in real time, testing and improving performance and reliability characteristics.

---

### iPAT CAPABILITIES

---

iPAT supports symbolic information from Intel 8086 and 80286 languages and

assemblers (OMF-86 and OMF-286), enabling use of module and procedure names, as well as line numbers, to specify and display ranges for analysis. All real- and virtual-address software can be analyzed, except software whose location in memory is dynamic during analysis.

iPAT also provides utilities for converting symbol files for logical range maps from other vendors' software tools (such as .MAP files from Microsoft's LINK) to Intel OMF-86 compatible symbol files for loading into iPAT. In addition, symbolic ranges can be defined by the user from within the iPAT environment.

Up to 125 modules, procedures, lines, or address partitions can be profiled at one time. Users can identify time spent in such ranges either with or without accruing to them time spent in some or all other ranges, such as subroutines, interrupt handlers, or an entire operating system. Code coverage information can be displayed for up to 252K of object code at once with no fundamental limit to the number of modules, procedures, lines, or address ranges. A synch line can be attached to external sources, including the output line of a logic analyzer.

Additional modes show the relationship between blocks of code, such as how specified procedures are used by a single calling procedure, how multiple ranges interact, and how often events such as interrupts occur while executing a range of code.

Function keys and context-sensitive prompt lines for command input provide the iPAT user with a tool environment that is easily learned and understood. Data collected by iPAT is accumulated until cleared, can be updated and displayed during system execution, and is easily saved and printed for documentation.

### SYSTEM DESCRIPTION/ CONFIGURATION

A complete iPAT system consists of the IPATCORE plus a probe package or an emulator system interface. The latter allows the iPATCORE to use the emulator

probe with the I<sup>2</sup>ICE or ICE as the target connection. Each requires the appropriate emulator system and PC host. Packages are described in the Ordering and Price Information box below.

<b>ORDERING &amp; PRICE INFORMATION</b>		
<b>DOS</b>	<b>iPATCORE</b>	<b>\$4,995</b>
	iPAT core system	
	<b>iPAT88DOS</b>	<b>\$2,995</b>
	8088/8086 probe package Requires an iPATCORE	
	<b>iPAT286DOS</b>	<b>\$3,995</b>
	80286 probe package Requires an iPATCORE	
	<b>IIIPATD</b>	<b>\$4,995</b>
	I <sup>2</sup> ICE emulator system interface. Requires I <sup>2</sup> ICE emulator system (pg. 22). Requires an iPATCORE and I <sup>2</sup> ICE emulator	
<b>DOS Kits</b>	<b>ICEPATKIT</b>	<b>\$6,125</b>
	Complete iPAT system including iPATCORE and software, PC cables, and ICE interface cables. Requires appropriate ICE emulator system (See Addendum).	
	<b>I<sup>2</sup>ICEPATKIT</b>	<b>\$9,990</b>
	Complete iPAT system including iPATCORE and software, PC cables, and I <sup>2</sup> ICE interface board and cables. Requires I <sup>2</sup> ICE emulator system (pg. 22).	
	<b>III010PATC86D*</b>	<b>\$17,995</b>
	Complete iPAT system with 8086/88 I <sup>2</sup> ICE emulator system (pg. 22), iC 86 (pg. 15), Assembler 86 and Relocation/Linkage 86 Package (pg. 9), and AEDIT text editor (pg. 33)	
	<b>III111PATC86D*</b>	<b>\$17,995</b>
	As above except with 80186/188 I <sup>2</sup> ICE emulator system	
	<b>III212PATC86D*</b>	<b>\$19,495</b>
	As above except with 10 MHz 80286 I <sup>2</sup> ICE emulator system	

\*New pricing data.

# EMULATORS

## 8086, 80186, 80286

- Provides full-speed, real-time, transparent emulation
- Uses common emulation base and buffer box for low-cost conversion among 8086, 8088, 80186, 80188, and 80286 processors
- Includes symbolic and source level debugging capabilities
- Provides built-in editor
- Supports interfacing to Intel's performance analysis tool to optimize code and improve software reliability
- Provides multi-condition, multi-level, multi-probe break and trace capability
- Simultaneously controls up to four microprocessors for debugging multiprocessor systems with a single workstation
- Maps user programs into a maximum of 288K bytes of zero-wait-state RAM
- Provides disassembly and single line assembly to help with on-line code patching
- Displays context-sensitive command menus as a quick reminder of the available commands

The Intel Integrated Instrumentation and In-Circuit Emulation (I<sup>2</sup>ICE) system is a hardware/software combination that aids the real-time debugging of systems using the 8086, 8088, 80186, 80188, and 80286 microprocessors. Support for the 8087 and 80287 coprocessors is also provided. The I<sup>2</sup>ICE emulator system supports

source-level debugging of programs built with Intel's software tools described in this catalog. As many as four I<sup>2</sup>ICE system instrumentation chassis can be hosted by an IBM PC/XT or PC/AT development host.

The base I<sup>2</sup>ICE system consists of an interface board that fits into the PC, an instrumentation chassis that can hold the common emulation boards and optional memory, optional iPAT interface boards, and cables. A separate personality module is available for each microprocessor group, enabling a single system to support the entire Intel 16-bit microprocessor family.

I<sup>2</sup>ICE software includes the host and probe software and confidence tests.

The I<sup>2</sup>ICE emulator enables hardware and software development of microprocessor applications to proceed simultaneously. While hardware development is in progress, programmers can test their software using the emulator hardware, which includes up to 288K bytes of programmable wait-state RAM (0 wait states at 10 MHz). The processor's I/O space can also be mapped to the development system for testing. Prototype hardware can then be added to the system as it is designed to provide early integration of hardware and software.

I<sup>2</sup>ICE provides the ideal debugging environment for any microprocessor system. It runs in real-time without intruding into the prototype's memory or interrupt structure. It offers a powerful break/trace state machine to isolate complex events, and it includes system emulation clips to sample eight external input lines during each processor bus cycle. It provides symbolic and source-level displays so that you can conduct your debugging operations in the language of your source code. When interfaced to the performance analysis tool (iPAT) you can closely couple performance analysis with emulation.

Despite the power offered by I<sup>2</sup>ICE, the product is easy to use. It includes an on-line syntax menu to help create syntactically correct commands. It enables you to combine debugging commands into procedures and save those procedures on the development system for later use. The built-in editor makes creating and modifying those files extremely easy.

### HIGHLIGHTS OF THE 8086/8088 PERSONALITY MODULE AND PROBE

- Provides up to 10 MHz real-time emulation
- Supports one megabyte addressing
- Emulates both minimum and maximum modes
- Provides 8087 math coprocessor support

### HIGHLIGHTS OF THE 80186/80188 PERSONALITY MODULE AND PROBE

- Supports one megabyte addressing
- Supports standard and queue status modes
- Provides keyword access to the internal peripheral control block and the relocation register

### HIGHLIGHTS OF THE 80286 PERSONALITY MODULE AND PROBE

- Provides up to 10 MHz real-time emulation
- Supports real and protected mode
- Includes an object code loader for both 8086 and 80286 object files
- Supports multiprocessing and multitasking
- Supports local descriptor tables (LDTs) and can read and write the normally invisible portions of segment and table registers
- Provides full 24-bit address mapping
- Supports DMA (Hold/Hold Acknowledge) in both emulation and interrogation modes

### ORDERING & PRICE INFORMATION

<b>DOS</b>	<b>pIII010KITD*</b>	<b>\$7,995</b>
	8086 emulator system—10 MHz	
	<b>pIII111KITD*</b>	<b>\$7,995</b>
	80186/80188 emulator system—10 MHz	
	<b>pIII212KITD*</b>	<b>\$9,995</b>
	80286 emulator system—10 MHz	
	<b>pIII086A904D</b>	<b>\$2,995</b>
	Conversion kit—converts 80186/188 or 80286 I <sup>2</sup> ICE emulator system to support 8086/88	
	<b>pIII186C914D</b>	<b>\$2,995</b>
	Conversion kit—converts 8086/88 or 80286 I <sup>2</sup> ICE emulator system to support 80186/188	
	<b>pIII286C924D*</b>	<b>\$4,995</b>
	Conversion kit—converts 8086/88 or 80186/188 I <sup>2</sup> ICE emulator system to support 10 MHz 80286	
	<b>pIII707</b>	<b>\$3,950</b>
	A 128K-byte optional high-speed memory board for the instrumentation chassis.	
	<b>IIIPLCC68</b>	<b>\$400</b>
	A PLCC interface for 80186/80188 and 80286 probes.	
	<b>IIIPLCC44</b>	<b>\$400</b>
	A PLCC interface for an 8086/8088 probe.	
<b>DOS Kits</b>	<b>pIII010PATC86D*</b>	<b>\$17,995</b>
	I <sup>2</sup> ICE emulator system, complete iPAT system (pg. 20) with 8086/88, iC 86 (pg. 15), Assembler 86 and Relocation/Linkage 86 Package (pg. 9), and AEDIT text editor (pg. 33)	
	<b>pIII111PATC86D*</b>	<b>\$17,995</b>
	As above except with 80186/188 I <sup>2</sup> ICE emulator system	
	<b>pIII212PATC86D*</b>	<b>\$19,495</b>
	As above except with 10 MHz 80286 I <sup>2</sup> ICE emulator system	

\*New pricing data.



Above Board kits bundle Intel's popular Above Board memory expansion boards with Intel program development software. There are several kits to choose from, each containing an Above Board, an assembler, the appropriate relocation and linkage tools, the AEDIT editor, and one compiler (PL/M or C). Versions of the Above Board are available for both the IBM PC/XT and compatible family (PS/PC) and the IBM AT and compatible family (PS/AT).

Each PC/XT Above Board is configured with 1.5 Megabytes of memory and each AT Above Board is configured with 2.0 Megabytes of memory. Both have a battery-powered clock/calendar, a serial port, and a parallel port. The extra memory can be used as a RAM disk for your program development software. You can decrease development time by running your assembles and compiles from this RAM disk.

#### **KIT CONTENTS**

The kits listed here are configured for either a PC/XT or /AT and consist of a code development kit for the 8086, 80286 or 80386. Each code development kit will include an Above Board, the appropriate Assembler and Relocation/Linkage Package (pg. 9), the AEDIT text editor (pg. 33), and either the C (pg. 15) or PL/M (pg. 13) compiler.

**See addendum for Inboard Kits.**

#### **ORDERING & PRICE INFORMATION**

**DOS XT Kits ABXTPLM86NL \$2,195**  
8086 code development kit with the PL/M compiler. The PSCOPE debugger (pg. 19) is also included.

**ABXTC86NL \$2,195**  
8086 code development kit with the C compiler. The PSCOPE debugger (pg. 19) is also included.

**ABXTPLM286NL \$2,195**  
80286 code development kit with the PL/M compiler.

**ABXTC286NL \$2,195**  
80286 code development kit with the C compiler.

**ABXTPLM386NL\* \$2,795**  
80386 code development kit with the PL/M compiler.

**ABXTC386NL\* \$2,795**  
80386 code development kit with the C compiler.

**DOS AT Kits ABATPLM86NL \$2,295**  
8086 code development kit with the PL/M compiler. The PSCOPE debugger (pg. 19) is also included.

**ABATC86NL \$2,295**  
8086 code development kit with the C compiler. The PSCOPE debugger (pg. 19) is also included.

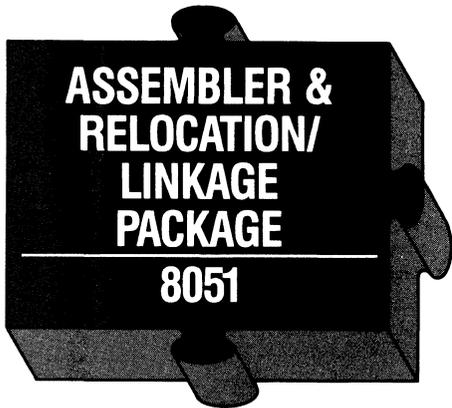
**ABATPLM286NL \$2,295**  
80286 code development kit with the PL/M compiler.

**ABATC286NL \$2,295**  
80286 code development kit with the C compiler.

**ABATPLM386NL\* \$2,895**  
80386 code development kit with the PL/M compiler.

**ABATC386NL\* \$2,895**  
80386 code development kit with the C compiler.

\*New pricing data.



The 8051 family of microcontrollers has been optimized to deliver high performance in real-time control applications. Assembler 51 is useful for implementing extremely time- or memory-critical applications. It can easily be mixed with PL/M 51 modules to enhance maintainability and portability.

---

#### ASSEMBLER 51

---

- Gives symbolic access to powerful 8051 hardware features
- Provides software support for many addressing and data allocation capabilities
- Provides symbol table, cross-reference table, macro capabilities, and conditional assembly
- Produces object files that can be linked together and located at absolute addresses or symbolically debugged with Intel's ICE-5100 emulator

---

#### RELOCATION/LINKAGE 51 PACKAGE

---

- Links modules generated by Assembler 51 and PL/M 51
- Locates linked object modules at absolute addresses
- Creates libraries of object modules and can add and delete library modules
- Permits modules to be selectively linked from libraries

- Converts 8051 object modules into symbolic hexadecimal format to facilitate file-loading by symbolic hexadecimal loaders (such as PROM programmers)

The Assembler 51 and Relocation/Linkage 51 Package is a complete package that supports all the latest proliferations of the 8051 family, including the 80C252 and the UPI-452 family of programmable I/O processors. The package includes Assembler 51, plus a relocation and linkage package that contains a librarian and an object-to-hex converter.

Assembler 51 is a powerful assembly language that gives you complete control over any microprocessor in the 8051 family, enabling you to produce the most efficient code possible. With Assembler 51, you can refer symbolically to many of the useful addressing features of the 8051. For example, you can use symbolic references for bit and byte locations, for 4-bit BCD arithmetic operations, for hardware registers, for I/O ports, for control bits, and for RAM addresses.

In addition, the assembler lets you break up code into separately-assembled modules, it provides conditional-assembly capabilities, and it supports macros to automate frequently-used code sequences.

The Relocation/Linkage 51 package supports the development of modular programs in a flexible manner. RL51 combines PL/M 51 and Assembler 51 modules and creates loadable code by assigning absolute memory addresses. LIB51 gathers modules into a library, where they can be accessed individually by RL51. OH51 converts 8051 object modules into hexadecimal form in preparation for loading into ROM.

#### ORDERING & PRICE INFORMATION

DOS	D86ASM51NL	\$750
	Quantity 2-4	\$650
	Quantity 5-9	\$575
	Quantity 10 +	\$495



# PL/M COMPILER & RELOCATION/ LINKAGE PACKAGE

## 8051

The 8051 microcontroller imposes unique constraints on a high-level language with its memory-mapped register sets, flexible operand addressing modes, large instruction set, programmable interrupt structure, configurable I/O, and limited on-chip memory. PL/M 51 has been tailor-made to deliver optimal performance while providing the maintainability, reliability, and portability of a high-level language.

### PL/M 51

- Gives high-level access to the powerful 8051 single-chip microcontroller family
- Gives programmers complete control of microcontroller resources by providing architecture-specific built-ins such as data type representation at run time, data type conversions, and support for task switching among register banks
- Provides optimal use of scarce on-chip 8051 RAM by overlaying variables
- Provides optimum balance among on-chip RAM usage, code size, and code execution time
- Produces object files that can be linked and assigned absolute addresses or symbolically debugged with Intel's ICE-5100 emulator

### RELOCATION/LINKAGE 51 PACKAGE

- Links modules generated by Assembler 51 and PL/M 51
- Locates linked object modules at absolute addresses

- Creates libraries of object modules and can add and delete library modules
- Permits modules to be selectively linked from libraries
- Converts 8051 object modules into symbolic hexadecimal format to facilitate file-loading by symbolic hexadecimal loaders (such as PROM programmers)

The PL/M 51 Compiler and Relocation/Linkage 51 Package is a complete package that supports all the latest proliferations of the 8051 family, including the 80C252 and the UPI-452 family of programmable I/O processors.

PL/M 51 is a genuine alternative to assembly language. Because it is an architecture-specific block-structured language, it satisfies the conflicting requirements of high performance and maintainable code. PL/M 51 provides control-flow statements for structured programming, while also supporting interrupt handling and architecture-specific attributes (such as the REGISTER and AUXILIARY variable attributes).

PL/M 51 also offers extra features such as a syntax checker, multiple levels of optimization, cross-reference generation, and debug record generation.

The Relocation/Linkage 51 package supports the development of modular programs in a flexible manner. The RL51 linker and relocater combines PL/M 51 and Assembler 51 modules and creates loadable code by assigning absolute memory addresses. LIB51 gathers modules into a library, where they can be accessed individually by RL51. OH51 converts 8051 object modules into hexadecimal form in preparation for loading into ROM.

### ORDERING & PRICE INFORMATION

DOS	D86PLM51NL	\$750
	Quantity 2-4	\$650
	Quantity 5-9	\$575
	Quantity 10 +	\$495



**MICRO/C-51  
COMPILER  
MICRO COMPUTER  
CONTROL  
CORPORATION**

**8051**

Because all of your C code is eventually assembled with Assembler 51, your C programs can be linked with Assembler 51 and PL/M 51 routines, and debugged symbolically with ICE-5100.

<b>ORDERING &amp; PRICE INFORMATION</b>		
DOS	MCC-51 *	\$595

\*New pricing data.

- Provides a subset of the C language defined by Kernighan and Ritchie, giving high-level access to the powerful 8051 single-chip microcontroller family
- Produces Assembler 51 source file as output, so the resulting code can be linked with other Assembler 51 and PL/M 51 modules and symbolically debugged with Intel's ICE-5100 emulator
- Enables access to all bit- and byte-addressable function registers

The Micro/C-51 compiler is a C compiler for the powerful 8051 family of microcontrollers. It enables programmers to use the familiar C language when developing 8051 applications.

The output of the Micro/C-51 compiler is an assembly language source file that is compatible with Intel's Assembler 51 and Relocation/Linkage 51 package. To generate executable code, you must compile your program with the Micro/C-51 compiler, assemble the resulting assembly language source file with Intel's Assembler 51, and link and locate the object code with Intel's Relocation/Linkage 51 package.



- Provides full-speed, real-time emulation of 8051 family, RUPI-44, and UPI-452 components
- Contains 64K bytes of mappable high-speed emulation memory and 254 frames of trace memory, including execution address and clips
- Includes symbolic debugging capabilities to permit access to memory locations and program variables by name
- Supports four address breakpoints with in-range, out-of-range, and page breaks
- Displays context-sensitive command menus as a quick reminder of the available commands
- Disassembles memory and provides an interactive assembler
- Contains a built-in, screen-oriented text editor

The ICE-5100 in-circuit emulator is a high-level, interactive debugging system that is used to develop and test systems based on 8051 family, RUPI-44, and UPI-452 microcontrollers. The emulator includes all hardware and control software to debug target software in real time before the target hardware becomes available.

The ICE-5100 system consists of a power supply, a controller pod, a crystal power accessory for running in standalone mode, a clip accessory for tracing external vents, and a serial cable to connect the emulator to the host development system. A separate user probe assembly is available for each of the microcontroller products supported. The software, including diagnostics and a tutorial, is available for an IBM PC/XT/AT or a compatible host running DOS 3.0 or later.

The ICE-5100 emulator enables hardware and software development applications to proceed simultaneously. While hardware development is in progress, programmers can execute and test their software using the emulator hardware, which includes 64K bytes of RAM. Prototype hardware can then be added to the system as it is designed to provide early integration of hardware and software. When the prototype hardware is complete, the ICE-5100 emulator provides real-time emulation of the microcontroller to debug the system as a completed unit.

The emulator provides mnemonic access to the microcontroller's function registers and register bits, and it allows the operator to use symbolic references (procedure names, line numbers, variable names, or labels) when changing variables, defining break or trace events, or referring to variables as primitive data types. The ICE-5100 emulator maintains 5K (up to 61K) bytes of host memory for storing the virtual symbol table, but also allows additional symbols to be stored temporarily on the host disk drive.

The ICE-5100 emulator provides four instruction execution address breakpoints that can be set to halt program execution at specific addresses, an address range, or a 256K byte page. Program tracing can be triggered by similar conditions.

To make emulation easier, the emulator allows users to combine commands into user-defined procedures that can be saved in text files and recalled for later use. An on-line syntax menu aids in creating syntactically correct commands, and a help menu provides command assistance.

#### **HIGHLIGHTS OF THE ICE-5100/252 USER PROBE (FOR 8051 SYSTEMS)**

- Provides full emulator support for the 8051 family
- Supports microcontroller speeds up to 16 MHz
- Includes a user probe assembly with a 40-pin DIP adapter
- Provides full CHMOS and NMOS component support

---

**HIGHLIGHTS OF THE ICE-5100/044  
USER PROBE (FOR RUPI-44  
SYSTEMS)**

---

- Provides full emulator support for the RUPI-44 family (8044, 8744, and 8344), including 8044-based BITBUS™ board products
- Supports microcontroller speeds up to 12 MHz
- Includes a user probe assembly with a 40-pin DIP adapter

---

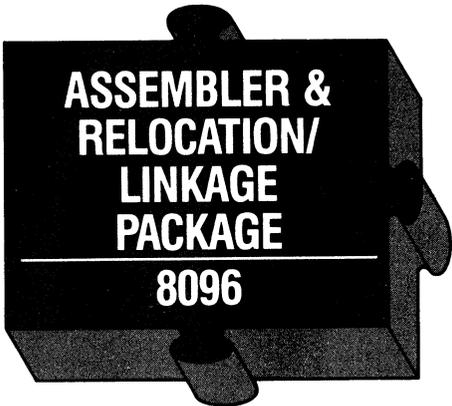
**HIGHLIGHTS OF THE ICE-5100/452  
USER PROBE (FOR UPI-452  
SYSTEMS)**

---

- Provides full emulator support for the UPI-452 family of I/O processors, including the 83452, 87452, and 80452
- Provides full emulation and debug support for the FIFO buffer
- Includes a user probe assembly with a 68-pin PGA adapter

**ORDERING & PRICE INFORMATION**

<b>DOS</b>	<b>pl252KITD</b>	<b>\$6,495</b>
Complete ICE-5100/252 system		
	<b>pl044KITD</b>	<b>\$6,495</b>
Complete ICE-5100/044 system		
	<b>pl452KITD</b>	<b>\$6,495</b>
Complete ICE-5100/452 system		
	<b>pTA252D</b>	<b>\$400</b>
PLCC target adapter for ICE-5100/252 and ICE-5100/044 applications		
	<b>TA452E</b>	<b>\$400</b>
PLCC target adapter for ICE-5100/452 applications		
<b>DOS Kits</b>	<b>pl252KITAD</b>	<b>\$6,995</b>
Complete ICE-5100/252 system with Assembler 51 and Relocation/Linkage 51 Package (pg. 25), and AEDIT text editor (pg. 33)		
	<b>pl044KITAD</b>	<b>\$6,995</b>
Complete ICE-5100/044 system with Assembler 51 and Relocation/Linkage 51 Package (pg. 25), and AEDIT text editor (pg. 33)		
	<b>pl452KITAD</b>	<b>\$6,995</b>
Complete ICE-5100/452 system with Assembler 51 and Relocation/Linkage 51 Package (pg. 25), and AEDIT text editor (pg. 33)		



The 8096 microcontroller family offers the highest level of integration ever achieved on a single-chip controller. The 8096 provides board-level performance by integrating a wide range of I/O facilities and peripherals into a powerful 16-bit CPU.

---

#### ASSEMBLER 96

- Gives symbolic access to powerful 8096 hardware features
- Provides support for many addressing and data allocation capabilities
- Provides symbol table, cross-reference table, macro capabilities, and conditional assembly
- Produces object files that can be linked together and located at absolute addresses or symbolically debugged with Intel's VLSiCE-96 or ICE-196 PC emulators

---

#### RELOCATION/LINKAGE 96 PACKAGE

- Supports single-precision, 32-bit floating-point variables and adheres to IEEE floating-point standard
- Offers base arithmetic operations of addition, subtraction, multiplication, division, Mod, and square root and includes an error-handler library
- Links modules generated by Assembler 96, PL/M 96, and iC 96, and supports incremental linking
- Locates linked object modules at absolute addresses

- Creates libraries of object modules and can add and delete library modules
- Permits modules to be selectively linked from libraries
- Converts 8096 object modules into symbolic hexadecimal format to facilitate file loading by symbolic hexadecimal loaders.

The Assembler 96 and Relocation/Linkage 96 Package is a complete package for the powerful 8096 family of microcontrollers, including the 80C196 and 8098. It includes Assembler 96, plus a relocation and linkage package that also contains a library of IEEE floating-point arithmetic functions and a librarian.

Assembler 96 is a powerful assembly language that gives you complete control over any microprocessor in the 8096 family. Assembler 96 lets you refer symbolically to many of the useful addressing features of the 8096. For example, an "include" file is supplied to define all the 8096 hardware registers.

In addition, the assembler lets you break up code into separately-assembled modules, provides conditional-assembly capabilities, and supports macros to automate frequently-used code sequences.

The Relocation/Linkage 96 package supports the development of modular programs in a flexible manner. FPAL96 is a library of single-precision, 32-bit floating-point functions that adhere to the proposed IEEE floating-point standard for accuracy and reliability. The RL96 linker and relocater combines program modules and assigns absolute addresses. It supports incremental linking, allowing more flexibility in the development environment. LIB96 gathers modules into a library, where they can be accessed individually by RL96.

#### ORDERING & PRICE INFORMATION

DOS	D86ASM96NL	\$750
	Quantity 2-4	\$650
	Quantity 5-9	\$575
	Quantity 10 +	\$495

# PL/M COMPILER & RELOCATION/ LINKAGE PACKAGE

---

## 8096

---

### PL/M 96

- Gives high-level access to the powerful 8096 16-bit microcontroller family
- Has block-structured design to encourage modular programming
- Enables programmers to minimize the use of scarce on-chip RAM
- Allows explicit definition of interrupt subroutines for improved speed
- Provides direct I/O support
- Gives programmers complete control of microcontroller resources
- Produces object files that can be linked together and located at absolute addresses or symbolically debugged with Intel's VLSiCE-96 or ICE-196 PC emulators

---

### RELOCATION/LINKAGE 96 PACKAGE

- Supports single-precision, 32-bit floating-point variables and adheres to IEEE floating-point standard
- Offers base arithmetic operations of addition, subtraction, multiplication, division, Mod, and square root and includes an error-handler library
- Links modules generated by Assembler 96, PL/M 96, and iC 96
- Locates linked object modules at absolute addresses

- Creates libraries of object modules and can add and delete library modules
- Permits modules to be selectively linked from libraries
- Converts 8096 object modules into symbolic hexadecimal format to facilitate file loading by symbolic hexadecimal loaders.

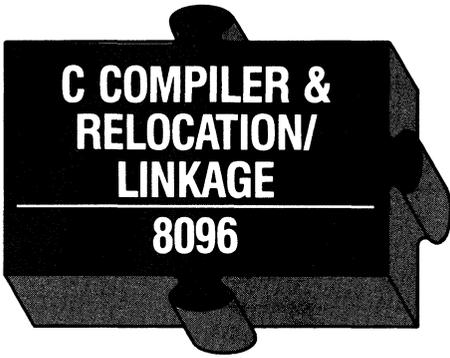
The PL/M 96 and Relocation/Linkage 96 package is a complete package that allows you to write high-level-language programs for the 8096 microcontroller family, including the 80C196 and 8098. It includes the PL/M 96 compiler, plus a relocation and linkage package that also contains a library of IEEE floating-point arithmetic functions and a librarian.

PL/M 96 is a structured, high-level programming language that supports the unique software requirements of advanced 16-bit microcontrollers. It supports seven data types, based variables, and two data-structuring facilities for flexible programming. It provides built-in functions specific to the 8096, such as OVERFLOW, and allows interrupts to be enabled and disabled. These architectural built-ins let you produce highly-optimized code.

The Relocation/Linkage 96 package supports the development of modular programs in a flexible manner. FPAL96 is a library of single-precision, 32-bit floating-point functions that adhere to the proposed IEEE floating-point standard for accuracy and reliability. The RL96 linker and relocater combines program modules and assigns absolute addresses. It supports incremental linking, allowing flexibility in the development environment. LIB96 gathers modules into a library, where they can be accessed individually by RL96.

#### ORDERING & PRICE INFORMATION

DOS	D86PLM96NL	\$750
	Quantity 2-4	\$650
	Quantity 5-9	\$575
	Quantity 10 +	\$495




---

## iC 96

---

- Implements the full C language as defined by the proposed ANSI standard, giving high-level access to the powerful 8096 16-bit, single-chip microcontroller family
- Supports interrupt handling directly from the high level language
- Produces high-density code rivaling an assembler
- Provides data types and structures similar to those provided by PL/M 96
- Produces object files that can be linked together and located at absolute addresses or symbolically debugged with Intel's VLSiCE-96 or ICE-196 PC emulators

---

## RELOCATION/LINKAGE 96 PACKAGE

---

- Supports single-precision, 32-bit floating-point variables and adheres to IEEE floating-point standard
- Offers base arithmetic operations of addition, subtraction, multiplication, division, Mod, and square root and includes an error- handler library
- Links modules generated by Assembler 96, PL/M 96, and iC 96, and supports incremental linking
- Locates linked object modules at absolute addresses
- Creates libraries of object modules and can add and delete library modules

- Permits modules to be selectively linked from libraries
- Converts 8096 object modules into symbolic hexadecimal format to facilitate file loading by symbolic hexadecimal loaders.

The iC 96 Compiler and Relocation/Linkage 96 Package is a complete package that allows you to write high-level-language programs for the advanced 8096 microcontroller family, including the 80C196 and 8098. It includes the iC 96 compiler, plus a relocation and linkage package that also contains a library of IEEE floating-point arithmetic functions and a librarian.

iC 96 is a structured language that supports the proposed ANSI C standard, allowing programs developed on other systems to be ported to the 8096 environment. It includes the latest C enhancements, such as structure assignments, functions taking structure arguments and returning structures, and the "void" and "enum" data types. In addition, a runtime library is provided to support the standard C language I/O library functions.

The Relocation/Linkage 96 package supports the development of modular programs in a flexible manner. FPAL96 is a library of single-precision, 32-bit floating-point functions that adhere to the proposed IEEE floating-point standard for accuracy and reliability. The RL96 linker and relocater combines program modules and assigns absolute addresses. It supports incremental linking, allowing flexibility in the development environment. LIB96 gathers modules into a library, where they can be accessed individually by RL96.

### ORDERING & PRICE INFORMATION

DOS	D86C96NL	\$750
	Quantity 2-4	\$650
	Quantity 5-9	\$575
	Quantity 10 +	\$495



## TEXT EDITOR PRODUCTIVITY TOOLS

- Allows full screen editing of source code and documentation
- Provides full range of editing support, from document processing to hexadecimal code entry
- Supports macros for repetitive or complex editing tasks
- Provides a powerful macro language for developing "smart" macros
- Supports dual-file editing with optional split-screen windowing
- Allows unlimited file size and line length
- Offers quick response with easy-to-use, context-sensitive command menus
- Is configurable and extensible for complete control of the editing process, yet remains easy to learn and use
- Supports documentation preparation with paragraph filling and justification options.
- Provides shell escape function for access to DOS commands

AEDIT is a full-screen editor designed specifically for software engineers and writers. It has many features that make it ideal for program editing. For example, it lets you switch between files instantly, and its split-screen windowing capability enables you to view two files at once. It has a macro facility that you can use to combine several functions into a single command. These macros can be created in two ways: by using AEDIT's "learning"

mode to store your keystrokes or by using the powerful macro language. You can also use and modify the extensive library of macros provided with the editor.

With these and other features, such as context-sensitive command menus and shell escape to the operating system, AEDIT is the complete program-editing tool.

### ORDERING & PRICE INFORMATION

DOS	D86EDINL	\$350
	Quantity 2-4	\$325
	Quantity 5-9	\$300
	Quantity 10 +	\$250
DOS Kits	D86PAK86NL	\$1,395
AEDIT, PSCOPE 86 (pg. 19) and the Assembler 86 and Relocation/Linkage 86 package (pg. 9).		



---

## **POLYTRON VERSION CONTROL SYSTEM (PVCS)**

---

POLYTRON Version Control System (PVCS) is a state-of-the-art source code control system for personal computers. Project administrators, managers, and librarians will find this tool indispensable for controlling the proliferation of revisions and versions of source code written in any language.

PVCS consists of several independent programs that allow users to maintain a history of the revisions to a source document. This source document can be any type of text file: program source code, correspondence, specifications, technical manuals, etc.

- Stores and retrieves multiple versions of source code, saving old versions in a space-efficient manner
- Automatically logs changes to maintain a complete product history
- Keeps an activity journal for a group of files to monitor the evolution of a software project
- Allows separate lines of development so that module changes can take several simultaneous paths
- Avoids access conflicts by alerting programmers when two or more of them wish to modify the same module
- Network versions also available

- Merges simultaneous changes into a single new version, flagging conflicts so the user can decide which change to incorporate
- Provides release and configuration control, enabling modules to be retrieved by their own revision numbers or by the system version label
- Works with PolyMake

There are two primary types of files used by PVCS: the source document (called the workfile) and archives of the revisions to the document (called logfiles). PVCS traces the evolution of a product by maintaining the historical sequence of changes made.

PVCS always keeps the latest revision of a document intact for quick retrieval, but for older revisions it minimizes storage requirements by saving just the differences between one revision and the previous one. This "reverse delta" technique makes PVCS the fastest version control system available.

PVCS provides the flexibility to fit into almost any programming environment. It permits simultaneous changes to the same module, it flags conflicting entries, and it enables branching so that multiple projects can be based on a single common source file. It has sensible defaults to simplify the first use, but it enables users to configure it to fit almost any project management strategy.

<b>ORDERING &amp; PRICE INFORMATION</b>		
<b>DOS</b>	<b>D86PVCS</b>	<b>\$395</b>

---

## POLYMAKE

---

PolyMake is a simple but powerful productivity tool for anyone who develops programs on personal computers running DOS. PolyMake is an intelligent assistant that remembers how to rebuild a program whenever any part of the program changes. It remembers which files depend on others and automatically updates the entire program whenever you type one simple command.

- Provides automated, intelligent software maintenance
- Enables you to rebuild an entire program simply by typing "make"
- Compatible with UNIX makefiles and uses UNIX syntax
- Operates on any DOS files (ASCII, binary, etc.)
- Works with any compiler, linker, librarian, version control system, or other system utilities that have a command driven interface.
- Uses modifiable and expandable operating rules, allows fully recursive makefile processing, and has an iterative macro expansion and command capability
- Supports generic operations that enable a single set of commands to describe how to rebuild all files of a given type
- Has a debug mode to help build correct makefiles
- Network versions also available

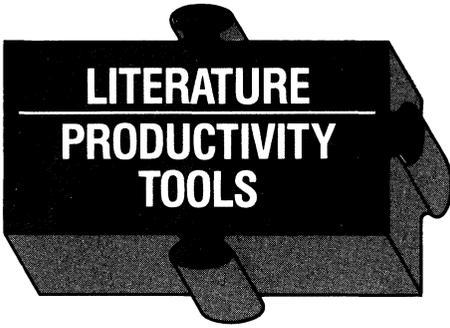
PolyMake knows how to rebuild programs because you specify the rules in an ASCII file called a makefile. The makefile consists of macro definitions, dependency descriptions, and executable commands. The executable commands can be any that your system supports, including compiling, linking, updating a database, or running any other utility. Once you specify the rules, no matter how complicated, you can forget them and let PolyMake regenerate your program whenever you make changes.

PolyMake saves you time because it regenerates only those parts of a program that need regenerating. To do this, it checks the date and time of each relevant target file and compares it with the date and time of each file on which the target depends. If any of the source files has a more recent date and time, PolyMake automatically remakes the target file.

The makefiles use generalized rules based on file extensions to reduce the need for repetition in the makefiles. In addition, special macros are available to simplify the expression of operations. The makefiles use the UNIX syntax, so UNIX programmers will feel right at home immediately.

PolyMake can work together with PVCS to provide added productivity. PolyMake reads the date and time stamps in PVCS logfiles. This further increases the productivity of programmers and system administrators by automating and adding precision to the configuration management process.

ORDERING & PRICE INFORMATION		
DOS	D86MAKE	\$149



Below are several publications we have reviewed and offer to supplement your development tools library. Also ask about Intel's complete Literature Guide.

**#555094 \$27**  
**8086/8088 16-Bit Microprocessor Primer, by Morgan, Waite. 1982.** Covers graphic peripherals, data types and numbers, physical memory organization, memory management, multi-processing and coprocessing. 355 pages.

**#555104 \$32**  
**The 16-Bit Microprocessor, by Goody. 1986.** Presents hardware, software, interfacing, and troubleshooting in an application-oriented product development environment. Emphasizes machine, assembly, and high level programming. 482 pages.

**#555694 \$18**  
**The 8087 Primer, by Palmer, Morse. 1984.** 8087 architecture and benefits. Includes interfaces with 8086. Programming in FORTRAN and Assembly language. 182 pages.

**#555859 \$26**  
**The 80286 Architecture, by Morse and Albert. 1986.** Describes the 80286 instruction set, registers and operating system features, programming, multi-tasking, virtual memory, and system security. Explains how to write programs requiring larger than available memory and interface hardware to 80286/80287 systems. 279 pages.

**#555108 \$26**  
**Programming the 80386, by Crawford and Gelsing. 1987.** Complete description of the assembly language programmer's view of the 80386/80387. Includes basic machine model, multitasking and memory management, plus 8086 and 80286 programs on the 80386. 750 pages.

**#555706 \$30**  
**The 8051: Programming Interfacing Applications, by Katz, Boyet. 1982.** 81 experiments with the 8051 single-chip microcomputer. Covers 16-bit timer-counters, serial I/O and arithmetic. 396 pages.

**#555954 \$45**  
**The 8096: Programming, Interfacing, Applications, by Katz, Boyet. 1986.** 122 hands-on experiments examples and applications using iSBE-96 emulator. 638 pages.

**#555106 \$24**  
**The C Companion, by Holub. 1987.** Provides background material for advanced understanding of the C language, and examines advanced topics such as uses of pointers and subroutines with a variable number of arguments. 284 pages.

**#555107 \$26**  
**C: A Reference Manual, by Harbison-Steele. 1987.** Complete description of C language, including recent extensions, the Draft Proposed ANSI C language, and over 180 standard run-time library functions. 404 pages. (Also included with D86C86NL and D86C386NL.)

**#555109 \$24**  
**A Guide to PL/M Programming for Microcomputer Applications, by McCracken. 1978.** Microcomputer programming using PL/M. Emphasizes effective use of procedures and block structuring. Includes many example programs. 247 pages.



**SOFTWARE  
SUPPORT  
SERVICES**

Below are the highlights of Intel's customized support services. There are multiple configurations of support to fit your specific development focus.

- Software Support Agreements are for a minimum term of one year and continue on a month to month basis thereafter until canceled by either party with a 30-day prior written notice to Intel Customer Support. A 20% discount will be applied if you purchase your extended software support at the time of your Intel software purchase and is extended to the second year with an initial 2-year software support purchase.
- Included in our Standard software support agreements are:
  - Technical Information Phone Service (TIPS): By calling a special 800 number you can communicate directly with a technical person for assistance.
  - Subscriptions:
    - a) Monthly publication, entitled ;COMMENTS;
    - b) Quarterly publications, Trouble Shooting Guides (TSG);
    - c) Intel response to written questions on standard SPR form, Software Problem Report (SPR).
  - Updates: You will have access to periodic updates and related documentation changes for a nominal charge.
  - Insite Users Library: Membership to this resource entitles you to purchase a variety of peripheral technical software.

- Bulletin Board Service (if available).
- Multiple system discounts can be negotiated.
- Consulting services are available upon request as a separate service outside the above agreement guidelines. These are billed on a time plus material basis for on-site field Systems Engineering Services. Travel and expenses are billed separately.
- Software support services are also available for other environments, including network software.

**ORDERING & PRICE INFORMATION**

The 1st price applies to single PC systems. The 2nd price applies to networked PC systems.

<b>DOS</b>	<b>SSCD51KIT</b>	<b>\$540</b>
	Covers all 8051 DOS development software	<b>\$1,440</b>
	<b>SSCD96KIT</b>	<b>\$540</b>
	Covers all 8096 DOS development software	<b>\$1,440</b>
	<b>SSCD86KIT</b>	<b>\$660</b>
	Covers all 8086 DOS development software	<b>\$1,680</b>
	<b>SSCD286KIT</b>	<b>\$660</b>
	Covers all 80286 DOS development software	<b>\$1,680</b>
	<b>SSCD386KIT</b>	<b>\$780</b>
	Covers all 80386 DOS development software	<b>\$1,980</b>
	<b>SSCD700KIT</b>	<b>\$1,140</b>
	Covers all DOS architecture development software	<b>\$2,940</b>
<b>VAX/VMS</b>		
<b>MicroVAX/VMS SC300</b>		<b>\$1,080</b>
	Covers all VMS development software for VAX and/or MicroVAX	

**NOTE:** Prices are for one year's service and are billed to your company purchase order.



## HARDWARE SUPPORT SERVICES

Below are the highlights of Intel's hardware support services.

- Standard Hardware Maintenance Agreements are available within a 50-mile radius of most major cities.
- Maintenance Agreement terms are one year minimum, and include on-site service, parts, and labor. This continues on a month to month basis thereafter until canceled by either party with a 30-day prior written notice to Intel Customer Support.
- Agreements are designed to cover entire systems, including the host IBM PC XTs/ATs and peripherals. Pricing is available on specific parts so that additions to your development system can be easily added to your Agreement.
- Coverage is for all parts, materials, and labor unless services are requested outside the covered hours/terms, and then only the labor is billable. Standard coverage is for 9-hour-day/5-day-week coverage. Extended coverage is available.
- Discount pricing will apply when 5 + systems are covered.
- Areas not included in our Standard Maintenance Agreements may be eligible for our Country Doctor Service at 150% current Standard Maintenance Agreement prices.
- As a separate service, certain products may be supported under our economic Carry-In Service.

- If equipment you wish to support is not listed below, be sure and inquire when you call 1-800-874-6835.

### ORDERING & PRICE INFORMATION

IBM XT/AT	\$ 500
iPAT	600
ICE-5100	600
I <sup>2</sup> ICE Systems	2000

**NOTE:** These prices are approximations. Specific pricing will be determined by configuration and quantity of host computer and peripherals. Our Catalog Sales Specialist will assist you with this when your equipment is identified.

Prices are for one year's service and are billed to your company purchase order.



# TRAINING WORKSHOPS

---

# SUPPORT SERVICES

In today's highly competitive marketplace, productivity is all-important. At Intel, our comprehensive training workshops are designed to substantially increase customer productivity by enhancing product knowledge. Technical instruction provides in-depth exposure to our microcomputer products, promoting a greater understanding of what Intel products can do.

- Classes include quality instructional materials and hands-on laboratory sessions.
- The workshops listed are 2-5 days in length, include all materials, and run from 8:30 AM to approximately 5:00 PM.
- Costs range from approximately \$695 to \$1195 per student. Group discounts are available.
- We will work with you and the closest Intel Training Center to identify the solution best suiting your needs. Obtain detailed information when you call 1-800-874-6835.
- Customized workshops are possible by special arrangements with an Intel Training Specialist.

## ■ Microprocessor/Microcontroller Workshops

- 80960 Architecture Development Workshop (begins July 1988)
- 80376/80386 Embedded Processor Systems Programming
- 80386 ASM for 8086 Programmers
- 80386 Hardware System Design
- 80386 Programming Using ASM 386
- 80386 System Software
- Introduction to Microprocessors using the 8086
- 8086 and 80286 Real Mode Programming
- 80186 Microprocessors
- 80286 Microprocessors
- Operating System Concepts Using the 80286
- Numeric Coprocessors
- I<sup>2</sup>CICE User's Workshop
- ICE 386
- 8051 Microcontrollers
- 8096 16-Bit Microcontrollers

## ■ Programming Workshops

- PL/M Programming
- PL/M 86/286/386 Programming

## ■ Intel Worldwide Training Centers

Boston  
Chicago  
Florida  
Portland  
Santa Ana  
Santa Clara  
Washington, D.C.  
Copenhagen  
Milan  
Munich  
Paris  
Rotterdam  
Stockholm  
Swindon  
Tel Aviv  
Hong Kong  
Osaka  
Seoul  
Singapore  
Taiwan  
Tokyo

# How to Order through the 800 #

Ordering is as easy as 1, 2, 3. Our Catalog Sales Specialist can either take your order directly through the 800# or assist you in reaching the closest Intel distributor or sales office.

1. **Approved company Purchase Order number or check information for prepaid orders.**
2. **Shipping and billing addresses.**
3. **Product codes and required quantities.**

## **Price/Sales Tax**

Product prices are listed on individual pages describing the products. Add the sales tax applicable to your state.

## **Licensing Requirements**

Most DOS-hosted products ship under shrink-wrap license. Some products may also require a Unix pass-thru license, or a signed Software License Agreement. Where required, these documents will be mailed to you when you place your order. Your order will be ready for processing upon receipt of the completed License Agreement(s).

## **Shipping Charges**

Intel ships UPS Blue Label unless otherwise requested. Freight is prepaid and billed with your product. Your Purchase Order can indicate "Plus Freight, UPS Blue."

## **DOS Software Guarantee**

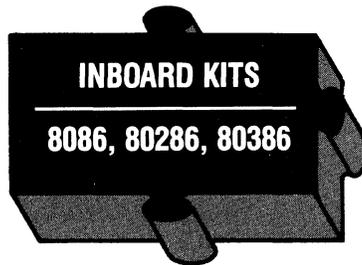
All Intel DOS hosted languages\* include a 30 day return policy. Individual return authorization paperwork is handled through the 800#, your distributor, or your Intel representative.

\*This applies to Intel's DOS version assemblers, compilers, utilities, editor, and debugger.

**NOTE:** The Product/Environment Reference provides an easy guide for finding a complete development solution that meets your needs.

# DEVELOPMENT TOOLS CATALOG

# ADDENDUM



## INBOARD KITS

8086, 80286, 80386



## EMULATORS

80186, 80286, 80386

Inboard Kits bundle the powerful 386 Inboard with Intel program development software. Each kit has an Inboard 386, assembler, relocation and linkage tools, AEDIT editor, and either PL/M or C compiler. Inboard 386 provides your PC/AT with the speed of an 80386 machine plus access to many of the 80386 capabilities. The Inboard 386 fits into a 16-bit slot with a cable connecting to the 80286 socket after removing the 80286. The 1M byte onboard memory can be configured as either conventional or extended memory. The Inboard 386 comes with various configurations and control software, including support for the Lotus/Intel/Microsoft (LIM) expanded memory specification.

### ORDERING & PRICE INFORMATION

Each kit requires an IBM PC/AT or compatible 80286 based PC and appropriate cable (listed below).

<b>DOS AT Kits</b>	<b>INBRDC86NL</b>	<b>\$3,595</b>
	8086 kit with C compiler and PSCOPE debugger.	
	<b>INBRDPLM86NL</b>	<b>\$3,595</b>
	Same as above with PL/M.	
	<b>INBRDC286NL</b>	<b>\$3,595</b>
	80286 kit with C.	
	<b>INBRDPLM286NL</b>	<b>\$3,595</b>
	Same as above with PL/M.	
	<b>INBRDC386NL</b>	<b>\$3,995</b>
	80386 kit with C.	
	<b>INBRDPLM386NL</b>	<b>\$3,995</b>
	Same as above with PL/M.	
<b>Cables:</b>	<b>PCIB3001</b>	<b>\$200</b>
	Cable for IBM PC/AT.	
	<b>PCIB3002</b>	<b>\$200</b>
	Cable for AT compatible PC.	

### IN-CIRCUIT EMULATORS (ICE)

- Provides full-speed, real-time, transparent emulation
- Includes symbolic and source-level debugging capabilities
- Supports interfacing to Intel's Performance Analysis Tool
- Supports interfacing to standard logic analyzers
- These emulators are hosted on the popular IBM PC/AT with 640K memory, Intel Above Board with 1M byte memory, and PC DOS 3.2 or later

The Intel In-Circuit Emulator (ICE) system is a hardware/software solution that aids real-time debugging of microprocessor applications. Intel's ICE system is designed so that software development can proceed and be debugged even when the user's hardware prototype is unavailable. ICE software takes advantage of the special debug information provided by Intel compilers and assemblers, providing superior symbolic debug ability.

### ICE-186 EMULATOR

- Supports 80C186 component designs
- Real-time emulation at 12.5 MHz
- Supports DRAM Refresh
- Supports Fast Breaks
- Single step capability
- 128K Bytes ICE memory

The ICE-186 emulator is ideally suited for developing real-time applications requiring time-critical hardware and software for 80C186 microprocessors. It captures up to 2,048 frames of processor activity, including both execution and databus activity. DRAM refresh signals are continued even when emulation has been halted so DRAM memory will not be lost. A complete ICE-186 system includes the control unit, power supply, probe, the crystal power accessory, and necessary cables.

### ORDERING & PRICE INFORMATION

<b>DOS</b>	<b>ICE186</b>	<b>\$ 9,995</b>
	<b>ICE186AB</b>	<b>\$11,140</b>
	Includes 2 MB Above Board	
<b>DOS Kits</b>	<b>ICE186PAT</b>	<b>\$16,120</b>
	ICE-186 emulator system with iPATCORE and software	

## ICE-286 EMULATOR

- Realtime emulation at 12.5 MHz
- Supports both protected and real modes
- Supports fast breaks

ICE-286 emulator is ideally suited for developing real time applications such as process control, machine control, communications, and other applications requiring full power of the 80286 microprocessor. It has full access to all protected mode registers and permits modification of register contents. The protected mode of execution is beneficial for secure, multitasking applications. A complete ICE-186 system includes the control unit, power supply, probe, the crystal power accessory, and necessary cables.

ORDERING & PRICE INFORMATION		
DOS	ICE286	\$12,495
	ICE286AB	\$13,640
Includes 2 MB Above Board		
DOS Kits	ICE286PAT	\$18,620
	ICE-286 emulator system with iPATCORE and software	

## ICE-386 EMULATOR

- Two versions supporting speeds up to 16 MHz or up to 20 MHz
- Loaders for 8086, 80286, and 80386 object modules
- 80387 numeric coprocessor support
- Includes optional isolation board

ICE-386 emulator is designed to provide unique access to the 80386 architecture. Your team can do parallel hardware and software development by using the emulator's loopback mode to start debugging software prior to prototype availability. With support for speeds up to 16 MHz or up to 20 MHz, it provides the fastest emulation available. ICE-386 can take advantage of the special debug information provided by Intel's complete set of software tools such as Assembler, C, and PL/M. The Optional Isolation Board is a board installed between the processor module and target system to protect your 80386 emulation from an untested target system.

ORDERING & PRICE INFORMATION		
DOS	pICE386DP	\$13,500
	Complete emulator system to 16 MHz	
	pICE386DPAB	\$14,645
	Above system plus 2 MB Above Board	
	ICE38620D	\$16,995
	Complete emulator system at 20 MHz	
	ICE38620DAB	\$18,140
	20 MHz system with 2 MB Above Board	



Intel's VLSiCE-96 and ICE-196PC in-circuit emulators offer powerful debugging solutions for microcontroller design needs. They each can operate independent of a prototype target system so you can begin immediate program development and debug, integrating hardware and software as your product is being developed. These in-circuit emulators are hosted on the standard IBM PC/XT or PC/AT and are supported by Intel's 8096 software development tools, such as the macro-assembler and high-level languages iC 96 and PL/M 96.

## VLSiCE-96 EMULATOR

- Supports fasbreaks and dynamic trace
- 4K bytes trace buffer
- Provides on-line help and editor
- Shadow registers allow reading many 8096 write-only and writing many read-only registers

VLSiCE-96 in-circuit emulator is designed so that its precise characteristics match the full speed of the component and make it a valuable tool for debugging your 8096 family of microcontroller designs. MCS-96 components supported by the VLSiCE-96 include the 8096, 8095, 8097, and 8098, as well as the ROM and EPROM versions. A significant benefit of this emulator is its ability to read many of the write-only registers and write many of the read-only registers. With the VLSiCE-96 emulator, emulation occurs in real-time without sacrificing microcontroller resources. To speed your learning curve, an on-line tutorial is included. You are able to interact with and use the emulator while executing the tutorial.

ORDERING & PRICE INFORMATION		
DOS	VO96KITD	\$8,495
Complete VLSiCE96 system		
	VO96KITAD	\$8,995
Complete VLSiCE96 system plus ASM 96 and AEDIT		

## ICE-196PC EMULATOR

- 2K-entry trace buffer
- Multiple Breakpoints
- Symbolic support and source code display

ICE-196PC in-circuit emulator delivers real-time high-level debugging capabilities for developing, integrating and testing 80C196-based microcontroller designs. This emulator represents a low-cost development environment for designing real-time microcontroller-based applications. The 2K byte trace buffer keeps a history of actual instruction execution. It can be displayed as disassembled instructions or original high-level language source code such as iC 96 or PL/M 96. ICE-196PC also includes an on-line help facility, dynamic command entry and syntax guide plus a built-in editor.

ORDERING & PRICE INFORMATION		
DOS	<b>ICE196PCB</b>	<b>\$2,495</b>
	Complete emulator system but minus the Crystal Power Accessory (CPA)	
	<b>ICE196PC</b>	<b>\$2,995</b>
	Complete emulator system including the CPA	
	<b>ICE196PC03A</b>	<b>\$4,300</b>
	Complete emulator system with guaranteed conversion to support 80C196KB component when available	
	<b>CPA 196</b>	<b>\$750</b>
	Crystal Power Accessory (Optional)	



- Assembler and utilities
- Full C compiler
- Execution vehicle for architecture evaluation, program execution, and software debugging
- Program development kit with Assembler, C compiler, and utilities
- Full speed in-circuit emulator
- DOS-hosted on popular IBM PC/AT and compatible computers

A complete development solution is being offered for Intel 80376 embedded microprocessor design and development. These tools include a high-level language compiler, assembler, software utilities, and an emulator. All work together to improve development productivity and allow extensive debug by providing full access to the 80376 architecture to maximize system performance and speed time-to-market.

## SOFTWARE DEVELOPMENT KIT

- Easy-to-order single kit for code development
- 80386 Assembler, C compiler, and utilities provide base for system programmers to develop for 80376 embedded applications
- Each element includes 80376 specific notes

ORDERING & PRICE INFORMATION		
DOS	<b>SKIT2</b>	<b>(call for avail.)</b>
	Includes ASM 386, iC 386, RLL 386, and a Help tool that has Loader, Debugger, and template	

## IN-CIRCUIT EMULATOR KITS

- Two differently configured kits to meet your specific development needs
- Each contains an 80376 specific converter card

These kits offer immediate symbolic debug capability for your 80376 embedded designs. Both use the existing ICE-386 as the emulation base.

ORDERING & PRICE INFORMATION		
DOS	<b>SKIT1</b>	<b>(call for avail.)</b>
	Includes 16 MHz ICE-386, Converter Card, and then full ICE-376 when available.	
	<b>SKIT3</b>	<b>(call for avail.)</b>
	Includes 80376 Converter Card and full ICE-376 when available (Requires existing ICE-376 emulator)	



- Products available individually and in complete easy-to-order starter kits in multiple combinations
- DOS-hosted on popular IBM PC/AT and compatible computers

The initial development toolset for this revolutionary 80960 high-performance embedded 32-bit processor family is available today for immediate design and development. Because Intel began work on these 80960 development tools early in the 80960 design, the initial support products are unusually complete and mature. A growing family of hardware and software development tools will be available concurrently with the release of new devices that evolve from the 80960 flexible "core" architecture.

### IC960DP C COMPILER

- Full C with ANSI extensions
- Optimizing compiler
- Retargetable STDIO library
- Floating point library
- In-line assembly coding

A full C language programming environment optimized for the 80960 family. It is specially tuned for the requirements of embedded 80960KB and 80960MC designs. This compiler requires an Intel Above Board with 1M byte memory. A Unix pass-thru license agreement is needed.

ORDERING & PRICE INFORMATION		
DOS	IC960DP	\$700
DOS Kits	Described below	

### EVA-960KB EXECUTION VEHICLE

- Supports speeds up to 20 MHz for 80960KB
- 64K bytes zero wait-state static RAM for benchmarking and performance evaluation
- 1M byte DRAM for code development and debug
- DOS access libraries
- Debug monitor

This evaluation board plugs into an IBM PC/AT expansion slot and provides a complete 80960KB subsystem for architecture evaluation, program execution, and software debugging. Requires a Unix pass-thru license.

ORDERING & PRICE INFORMATION		
DOS	EVA960KB	\$6,000
DOS Kits	Described below	

### ASM 960 ASSEMBLER

- Assembler
- Linker/Loader
- Macro Preprocessor
- Archiver
- Object Module Utilities
- COFF to HEX Converter
- PROM Builder

A complete assembly language development environment to support speed-critical programming and debugging with first release supporting 80960KB and 80960MC. All 80960 tools require a Unix pass-thru license agreement.

ORDERING & PRICE INFORMATION		
DOS	ASM960D	\$900
DOS Kits	Described below.	

### 80960 DEVELOPMENT KITS

The kits listed here are configured for your specific development focus with 80960 embedded designs. The contents of each easy-to-order kit is described under the product ordering code.

ORDERING & PRICE INFORMATION		
DOS Kits	960SKIT3	\$1,440
	Includes ASM 960D Assembler and iC 960DP compiler	
	960SKIT3AB	\$2,590
	Same as above plus Intel Above Board with 2M byte memory	
	960SKIT1	\$6,200
	Includes EVA960KB execution vehicle and ASM 960D	
	960SKIT2	\$6,800
	Same as above plus iC 960DP compiler (Requires Above Board)	
	960SKIT2AB	\$7,950
	Same as SKIT2 plus Intel Above Board with 2M byte memory	



5200 N.E. Elam Young Parkway  
Hillsboro, OR 97124-9990