iAPX 386
HIGH PERFORMANCE 32-BIT MICROPROCESSOR
PRODUCT PREVIEW

- High Performance CPU
  - 2 to 3 times iAPX 286 performance
  - Pipelined implementation
  - Multiple on-chip caches
  - High bandwidth 32-bit bus
  - 12 and 16 MHz clock

- Complete Support for 32-bit Addressing
  - 2^{32} bytes of physical address space
  - 2^{32} bytes per segment
  - 2^{46} bytes of virtual address space per task

- Software Compatible with iAPX 86, iAPX 88, iAPX 186, iAPX 188, and iAPX 286 microsystems

- Multiple Coprocessor Interface

- Complete 32-bit Supermicro
  - 8-, 16-, and 32-bit data
  - Integrated Multitasking support
  - On-chip Virtual Memory support

- On-chip Memory Management and Protection
  - Fully compatible with iAPX 286

- Optional On-Chip Paging

- CHMOS III Technology

The iAPX 386 is the first 32-bit member of the iAPX 86 family. It maintains software compatibility with the entire iAPX 86 family (8086, 8088, 80186, 80188, and 80286), while adding full 32-bit and Supermicro capabilities and significantly enhanced performance.

The iAPX 386 brings new performance standards to the microprocessor world. The combination of advanced process technology and architectural concepts which were previously found only on mainframe and large minicomputers make the iAPX 386 the highest performance microprocessor in its category. Concepts such as caching, pipelining, high performance bus, and a high-speed execution unit provide this level of performance at both CPU and system level.

![Figure 1. iAPX 386 Pipeline](image_url)
The iAPX 386 provides access to the large base of software developed for the 8086, 8088, 80186, 80188, and 80286 microprocessors. It maintains binary-level-code compatibility with these earlier generations to allow execution of existing applications without recompilation or reassembly. The user preserves his software investment and can reduce the time-to-market for new products.

The iAPX 386 provides full 32-bit Supermicro support. 32-bit architecture and internal implementation, including registers, arithmetic-and-logic unit, instruction set, addresses, address bus and data bus, provide the functionality needed to build a full, 386-based, 32-bit system.

Supermicro capabilities, such as hardware-supported multitasking and virtual memory support, provide the foundations necessary to build advanced multitasking and multiuser systems. The hardware multitasking support of the 386 automatically stores the task state and loads the new task during task switch operations. The on-chip memory management and protection mechanism performs the mapping from virtual to physical memory and provides the protection necessary for maintaining task integrity in a multitasking environment. Optional paging allows for a finer granularity of physical memory management.

The iAPX 386 supports a full family of specialized coprocessors. Coprocessors allow separation of specialized functions at the component level, so that the user can tailor the system’s functionality and performance to his application while optimizing the design for cost. Multiple coprocessors are supported by the 386, such as the new numerics coprocessor for the 386, the 80387, the Ethernet Local Area Network coprocessor, the 82586, and the text coprocessor, the 82730.

**PERFORMANCE**

The iAPX 386 provides two to three times the performance of the industry performance leader, the iAPX 286. A combination of advanced technology, pipelined architecture, on-chip code and descriptor caches, high performance bus and high speed coprocessors provide this improvement. With the higher execution speed, the designer can develop more complex systems and applications, increasing the appeal of his end product to the market.

**CHMOS**

Intel’s advanced CHMOS III process (Complementary High Speed Metal Oxide Semiconductor) eliminates the frequency and reliability limitations of traditional CMOS processes and opens a new era in microprocessor performance. It combines the high performance capabilities of Intel’s leading HMOS III technology with the high density and low power characteristics of CMOS. Using this technology, the iAPX 386 is designed to operate at 12 and 16 MHz.

**Pipelined Microarchitecture**

The iAPX 386’s pipelined architecture provides parallel fetching, decoding, execution and address translation inside the CPU. Like the 286, four units are included in the pipeline. The Bus Interface Unit generates the bus transactions and keeps the prefetch queue and the code cache full and ready for access by the Decode Unit. The Decode Unit breaks each instruction into a fully decoded equivalent and places it in the instruction queue, ready for fast and efficient execution by the Execution Unit. The Address Unit performs the address calculation and the protection checks for virtual and physical memory management (see Figure 1). This highly pipelined organization eradicates instruction fetch and decode times as consumers of execution time, providing a radical improvement in performance.

Inside the pipeline, a code cache, a segment descriptor cache, and a page descriptor cache maintain instructions and memory information ready for immediate use by the CPU.

**On-Chip Code Cache**

The instruction cache improves processor performance by providing an instruction fetch path that is much faster than an external bus, and which decreases the consumption of bus bandwidth for instruction fetch cycles. The instruction cache maps the most recently referenced memory locations to high-speed on-chip memory. Most programs exhibit a high degree of locality of reference, so that even though the cache memory is much smaller than the system memory, the mapping of the most recently referenced locations ensures a high hit rate in the cache. Overall system performance is typically increased by 20 to 30%.

**Segment and Page Descriptor Caches**

The segment descriptor and page descriptor caches maintain information on the working set inside the CPU. This information is available for immediate access, eliminating references to memory-based segment or page tables when accessing the task’s address space.

These descriptors are in the on-chip memory management unit, which provides a further increase in performance. By performing memory management on-chip, the 386 eliminates any memory access delays required by other implementations that use off-chip memory management units. The result is high performance and relaxed memory access time requirements. Address translation in parallel with the
High Speed Bus
The iAPX 386 implements a 32-bit data path and 12 MHz or 16 MHz bus clock to provide a throughput rate 2 to 4 times that of current microprocessor systems. This high speed bus increases overall system performance by insuring prompt transfers between the CPU, memory and peripherals without becoming a bottleneck for the system. An estimated system performance increment of about 20% is achieved via this high speed bus.

Coprocessors
To provide even higher system performance, the iAPX 386 will support advanced coprocessors. The 80387, the numerics coprocessor for the iAPX 386, is software compatible with the 8087 and 80287, and will perform numeric calculations at more than 4 times the speed of the 287. Numeric intensive applications in a 386/387 system will run significantly faster than even the performance leading 286/287 systems.

Other coprocessors will also support the iAPX 386. For example, the 82586 Ethernet LAN and the 82730 Text coprocessors will run with the 386, offloading the CPU of the functions for which these two coprocessors are specially designed, further increasing system performance.

SUPERMICRO SUPPORT
The iAPX 386 provides complete 32-bit support as well as Supermicro support. Data types, registers and instructions have been expanded to support 32-bit systems.

Data Types
The data types are the same as in the 286: arithmetic, floating point, byte string, and bit field. Table 1 presents an overview of the data types available under each category.

<table>
<thead>
<tr>
<th>Table 1 - Data Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arithmetic</td>
</tr>
<tr>
<td>8-, 16-, 32-bit integers</td>
</tr>
<tr>
<td>8-, 16-, 32-bit ordinals</td>
</tr>
<tr>
<td>Unsigned packed decimal - BCD</td>
</tr>
<tr>
<td>Unsigned unpacked decimal - BCD</td>
</tr>
<tr>
<td>Floating Point</td>
</tr>
<tr>
<td>Supported in 80387 coprocessor</td>
</tr>
<tr>
<td>16-, 32-, 64-bit integer</td>
</tr>
<tr>
<td>32-, 64-, 80-bit IEEE P754 standard</td>
</tr>
<tr>
<td>80-bit decimal format</td>
</tr>
<tr>
<td>Byte String</td>
</tr>
<tr>
<td>Variable length: 0 to 4 Gigabytes</td>
</tr>
<tr>
<td>Bit Field</td>
</tr>
<tr>
<td>1 or more bits</td>
</tr>
</tbody>
</table>
Instruction Set

The iAPX 386’s instruction set provides for full 32-bit data manipulation and addressing, in addition to 8- and 16-bit data, and 16-bit compact addressing. The 386’s instruction set is a superset of the 286’s, and allows full compatibility with the iAPX 86, 186, and 286.

All existing instructions have been extended to support 32-bit addresses and operands. New bit manipulation and other instructions have been added for extra flexibility in designing complex software. Operating systems, compilers and graphics applications will benefit from these new instructions.

Register Sets

The general registers of the 386 support 32-bit data and addressing. They also provide for 8- and 16-bit data and 16-bit compact addressing, thereby offering total software compatibility. Figure 3 shows a model of the general register set.

The segment registers of the iAPX 86 family are identical in the 386 with the addition of non-dedicated segment registers, FS and GS, for increased flexibility in simultaneously manipulating multiple data structures. Figure 4 provides a model of the segment register set.

Memory Addressability

The iAPX 386 provides complete 32-bit addressability to very large memories. Table 2 indicates the memory addressability of the 386.

The 386’s large addressability allows for direct mapping of the large mass storage devices that will be available in the future, such as advanced winchester and video disks, into virtual memory.

<table>
<thead>
<tr>
<th>Physical Memory</th>
<th>4 Gigabytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segment Offset</td>
<td>4 Gigabytes</td>
</tr>
<tr>
<td>Virtual Memory per task</td>
<td>64 Terabytes</td>
</tr>
</tbody>
</table>

Effective Addressing

The iAPX 386’s effective addressing is an expansion of the 286 model. Just like in the 286, an optional base is added to an optional index and then to an optional displacement to provide the offset. The 386’s mechanism is fully compatible with the rest of the iAPX 86 family. Table 3 indicates the choices available for address calculation.

---

Figure 3. General Register Set
Memory Management Model — Segmentation and Paging

Memory management and protection allows designers to use system memory efficiently, while at the same time providing a mechanism to use virtual memory. Product development is simplified, both hardware and software.

The iAPX 386 offers on-chip memory management and protection to provide for the management and protection of the entire system memory. Not only does it increase performance by being on-chip, but it reduces system costs by eliminating external memory management units; by simplifying development of memory management software; by reducing memory access requirements, and by providing complete compatibility from a 286 to a 386 system and from a 386 system to other 386 systems. Table 4 summarizes the benefits of the on-chip memory management unit.

The iAPX 386 offers segmented virtual memory and optional paging for physical memory management and swapping.

SEGMENTATION

Segmentation provides the management of virtual memory. To the programmer, it is a natural way of organizing programs. He can set up various segments for different types of data (file buffers, program

Table 3 - Address Calculation

<table>
<thead>
<tr>
<th>16-Bit</th>
<th>32-Bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>(86/186/286 Compatible)</td>
<td></td>
</tr>
<tr>
<td>CS, SS, DS, ES</td>
<td>SEGMENT</td>
</tr>
<tr>
<td>None, or 16-bit Base Register</td>
<td>CS, SS, DS, ES, FS, GS</td>
</tr>
<tr>
<td>+ OPTIONAL BASE</td>
<td>None, or 32-bit Base Register</td>
</tr>
<tr>
<td>None, or 16-bit Index Register</td>
<td>+ OPTIONAL INDEX</td>
</tr>
<tr>
<td>None, 8, or 16-bit</td>
<td>None, 8, or 32-bit</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 4 - Elimination of an External MMU

<table>
<thead>
<tr>
<th>Increases</th>
<th>Reduces</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Performance</td>
<td>System Cost</td>
</tr>
<tr>
<td>Compatibility and Upgrade Path</td>
<td>Extra Hardware</td>
</tr>
<tr>
<td>Overall Memory Bandwidth</td>
<td>Faster Memories</td>
</tr>
<tr>
<td>System Reliability</td>
<td>Development time</td>
</tr>
<tr>
<td></td>
<td>System Complexity</td>
</tr>
<tr>
<td></td>
<td>Risk of Incompatibility</td>
</tr>
</tbody>
</table>

variables, work areas, etc.). He can also break up the virtual memory according to the organization of the program routines, permitting modular compilation of the different routines at separate times.

Code and data, by nature, are of variable size. Segmentation provides the right virtual memory fit, assigning only as much memory as needed for the individual routines and data. It is a way of directly relating memory organization and usage to the code and data structures.

Segmentation also simplifies the sharing of modules. A single descriptor applies to an entire data structure to allow fast, efficient control and sharing of programs. For the operating system, it is very easy to transfer the module descriptors to another task to share a program.

Finally, segmentation provides for protection. Since an entire code module or data structure is defined by a single descriptor, this descriptor can also uniquely specify the usage and attributes of the segment. Segment limits help detect limit violations or stack overflow, allowing these errors to be automatically detected, prevented and reported to the operating system. Segmentation provides the mechanism to protect the programmer’s code and data directly as he sees it: in variable size chunks, large or small.

**PAGING**

With the advent of the potential for large multi-megabyte segments, a more granular mechanism to manage physical memory becomes necessary in swapping systems. While this is not an issue for static systems, loading and unloading segments of hundreds of kilobytes to megabytes into physical memory can be slow and cumbersome. To avoid this, the iAPX 386 offers paging to manage physical memory. Placed under segmentation, it breaks up large segments into small fixed size blocks. Instead of having to load the entire segment, the operating system can load the individual pages as needed. This simplifies the management of physical memory. It also provides for small physical memory allocation for each task. Rather than having many megabytes per task present in physical memory, the 386’s operating system only needs to keep the current pages. This feature takes advantage of the locality of reference displayed by most programs and takes advantage of swapping.

Paging is optional and completely transparent to the application programmer. Its presence does not affect his programming at all; he can assume that it does not exist. Paging is a tool for the developer of the swapping system. Paging by itself, though, cannot easily provide the protection, modularity, and the sharing of programs that segmentation allows.

Paging and segmentation are complementary and together provide the best memory management and protection solution. Segmentation is optimal for the way a programmer sees and protects his program. Managing large physical memories for large code and data structures in swapping systems makes good use of paging. Table 5 summarizes this complementary nature of segmentation and paging.

Table 5 - Segmentation and Paging
A Complementary Pair

<table>
<thead>
<tr>
<th>SEGMENTATION</th>
<th>PAGING</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Large virtual space per task</td>
<td>• Fine and regular granularity for physical memory management</td>
</tr>
<tr>
<td>• Variable size to match code and data structure</td>
<td>• Small working sets for large tasks</td>
</tr>
<tr>
<td>• One segment may represent entire code or data structure</td>
<td>— Many memory resident tasks</td>
</tr>
<tr>
<td>• Tool for applications programmer</td>
<td>• Swapping and physical memory management tool</td>
</tr>
<tr>
<td></td>
<td>— Invisible to applications programmer</td>
</tr>
</tbody>
</table>
Segmentation Description
The segmentation mechanism of the iAPX 386 is identical to the iAPX 286's. This mechanism, very similar to the one used in Multics and the Burroughs B5000 computers, uses a selector held in a segment register to point to a segment of the user's address space via a descriptor. This descriptor holds the base address, limit, access rights and protection information. The base address is added to the effective address to obtain the 32-bit address. See figure 5 for details.

Paging Description
If paging is used, the linear address generated by the segmentation mechanism is mapped through page tables to obtain the physical memory address of the desired information. See figure 6 for more details.

Paging is an optional feature. Disabling it provides a memory mechanism identical to the 286's. This means that 286 operating systems can easily migrate directly into the 386 with complete assurance that the memory management and protection will work the same. This allows simple migration of 286-based operating systems to very high performance 386 systems and quick time to market.

Protection
As mentioned before, the iAPX 386 offers hardware enforced protection. The mechanism is the same as the 286's, insuring software compatibility. Tasks are isolated and protected from each other. This guarantees that code and data will not be damaged by bugs in other programs.

In addition, the protection mechanism offers four privilege levels to provide protection of sensitive code and data within a task. This allows a combination of operating system, system services, shared libraries and application programs to reside in a common virtual address space, yet be protected. The highest privilege level is typically reserved for the operating system kernel, the most trusted program in the system. Other levels below the kernel can be configured to include system and application services code, as well as the application itself. Structured operating system design is encouraged by this mechanism, making operating systems simpler to implement and easier to maintain. Figure 7 represents this protection model.

Quick, controlled access to the data and routines at each level is directly provided by the 386 integrated protection. This preserves system integrity as well as the high performance of the iAPX 386.

Multitasking
Many advanced applications require multiple activities to be in process at any given instance. This requires systems to perform task switches frequently. To improve the performance of these systems, the 386, just like the 286, offers hardware integrated multitasking. What this means is that the traditional steps involved in task switching, storing the task context, loading the next task and giving it control, are automatically performed in hardware. This simplifies operating system development by moving most of the task switch chores to the hardware, while at the same time improving performance. Hardware supported multitasking execution takes a lot less time than a software implementation. The system can switch among many tasks quickly and often, without incurring performance degradation.

![Figure 5. Segmentation Mechanism](image-url)
SOFTWARE COMPATIBILITY
The iAPX 386 continues the evolution of the iAPX 86 family by maintaining full compatibility at the object-code level with the 8086, 8088, 80186, 80188, and 80286 microprocessors. This compatibility provides immediate access to the largest base of 16-bit
microprocessor software in existence. The designer preserves his investment in software, both applications and operating systems, and can add new software to his system quickly and efficiently.

To utilize the majority of the application software, no recompilation or reassembly is necessary. Exceptions might exist for speed or input/output configuration-dependent software, but the majority of the software will run unchanged.

The same is true of operating systems. Not only is the software compatible at the object-code level, but since the segmented memory management and protection mechanism of the 286 and 386 are identical, no changes are necessary to the operating system routines in charge of managing the memory. If paging is going to be used, routines to manage the pages can be added under the segmentation software.

In other words, iAPX 86 family software, both operating systems and applications, will run on the iAPX 386 without changes and at significantly higher speeds. Not only does the designer protect his software investment, but he can reduce his time to market by using existing 86 software in his new 386 system.

**SUMMARY**
The iAPX 386 is the most advanced 32-bit microprocessor. It not only provides the highest performance in the industry, but with its 32-bit Supermicro capabilities of virtual memory and multitasking support, it will serve as the standard in 32-bit processing. A full line of coprocessors insures additional performance and functionality to meet the designer’s needs.

With its complete software compatibility with the iAPX 86 family, the iAPX 386 allows the designer to develop a very powerful and advanced system, while reducing his time to market. The software base available for the 386 is the largest in the industry, based on the industry standard, the iAPX 86 architecture.

With such advanced capabilities and software base, the iAPX 386 provides the designer with the most comprehensive and best supported 32-bit solution.

**DOCUMENTATION**
Further information is provided in the following documents:

- Introduction to the iAPX 286, #210308-001
- iAPX 286 Operating System Writer’s Guide, #121960-001

![Figure 8. Software Compatibility in iAPX 86 Family](image-url)
ALABAMA
Inte! Corp.
303 Williams Avenue, S.W.
Suite 1422
Huntsville 35801
Tel: (205) 533-9353

ARIZONA
Inte! Corp.
1125 N. 20th Drive
Suite 2140
Phoenix 85029
Tel: (602) 895-4180

CALIFORNIA
Inte! Corp.
1010 Hunley Way
Suite 300
Sacramento 95825
Tel: (916) 929-4078
Inte! Corp.
7607 Opportunity Road
Suite 135
San Diego 92111
Tel: (714) 266-3363

COLORADO
Inte! Corp.
2000 East 4th Street
Suite 100
Santa Ana 92705
Tel: (714) 835-3943
TWX: 910-959-1114
Inte! Corp.
1320 Sheridan Way
Mt. View 94043
Tel: (415) 968-8866
TWX: 910-339-9279
910-339-2355
Inte! Corp.
5530 Corbin Avenue
Suite 120
Tarzana 91356
Tel: (213) 708-2333
TWX: 910-495-2045

CONNECTICUT
Inte! Corp.
4445 Northpark Drive
Suite 100
Colorado Springs 80907
Tel: (303) 954-8622
Inte! Corp.
600 S. Cherry Street
Suite 700
Denver 80202
Tel: (303) 321-8065
TWX: 910-321-2289

FLORIDA
Inte! Corp.
35 Padarana Road
Ft. Lauderdale 33309
Tel: (305) 771-0600
TWX: 710-455-1199

GEORGIA
Inte! Corp.
3300 Holcombe Bridge Road
Suite 225
Baltimore 21218
Tel: (301) 445-0541

ILLINOIS
Inte! Corp.
250 Golf Road,
Suite 815
Rolling Meadows 60008
Tel: (708) 971-7200
TWX: 910-651-2388

INDIANA
Inte! Corp.
9100 Purgue 2099
Suite 100
Indianapolis 46258
Tel: (317) 876-0620

IOWA
Inte! Corp.
St. Andrews Building
1300 St. Andrews Drive N.E.
Cedar Rapids 52402
Tel: (319) 393-5510

KANSAS
Inte! Corp.
8409 W. 11th Street
Suite 100
Overland Park 66210
Tel: (913) 542-8080

LOUISIANA
Inte! Corp.
Industrial Digital Systems Corp.
2323 Severin Avenue
Suite 200
Metairie, LA 70001
Tel: (504) 831-8492

MARYLAND
Inte! Corp.
7257 Parkway Drive
Hanover 21076
Tel: (410) 796-7500
TWX: 710-662-1944
Inte! Corp.
1620 Elson Road
Silver Spring 20903
Tel: (301) 431-1200

MASSACHUSETTS
Inte! Corp.
27 Industrial Avenue
Chelmsford 01824
Tel: (617) 254-1800
TWX: 710-343-6333
EMC Corp.
385 Elcott Road
Newton 21284
Tel: (617) 264-4744
TWX: 922531

MICHIGAN
Inte! Corp.
2650 Northwestern Hwy.
Suite 401
Southfield 48075
Tel: (313) 352-0920
TWX: 810-244-9415

MINNESOTA
Inte! Corp.
3500 W. 80th Street
Suite 360
Minneapolis 55431
Tel: (612) 836-8722
TWX: 910-576-2867

MISSOURI
Inte! Corp.
4203 East City Expressway
Suite 131
Earth City 63045
Tel: (314) 291-1990

NEW JERSEY
Inte! Corp.
Randi Plaza III
Randi Center
Ewickon 08837
Tel: (201) 233-3000
TWX: 710-480-9338

NEW MEXICO
Inte! Corp.
1129 Juan Tabo N.E.
Albuquerque 87121
Tel: (505) 282-8586

NEW YORK
Inte! Corp.
300 Vanderbilt Motor Parkway
Hauppauge 11788
Tel: (516) 231-3300
TWX: 510-227-6236
Inte! Corp.
80 Washington Street
Rochester 14601
Tel: (716) 473-2300
TWX: 510-248-0060
Inte! Corp.
211 W. Spruce Boulevard
Rochester 14623
Tel: (716) 242-6550
TWX: 510-253-7391

NORTH CAROLINA
Inte! Corp.
2306 W. Meadowview Road
Suite 206
Greenboro 27407
Tel: (919) 294-1514

OHIO
Inte! Corp.
6550 Poe Avenue
Dayton 45414
Tel: (909) 252-4455
TWX: 810-450-2528

Inte! Corp.
Chagrin-Breckenridge Blvd.
Suite 300
Cleveland 44122
Tel: (216) 464-6915
TWX: 810-427-9298

OKLAHOMA
Inte! Corp.
4157 S. Harvard Avenue
Suite 123
Tulsa 74135
Tel: (918) 746-8688

OREGON
Inte! Corp.
10700 SE Beaver Point
Hillsdale Highway
Suite 22
Beaverton 97005
Tel: (503) 241-8086
TWX: 910-467-8741

PENNSYLVANIA
Inte! Corp.
510 Pennsylvania Avenue
Fort Washington 19084
Tel: (215) 861-1000
TWX: 910-661-2077

Inte! Corp.
201 Penn Center Boulevard
Suite 301
Pittsburgh 15235
Tel: (412) 823-4970
Q.E.D. Electronics
300 N York Road
Hatboro 19040
Tel: (215) 747-9060

TEXAS
Inte! Corp.
12300 Ford Road
Suite 380
Dallas 7524
Tel: (214) 241-8087
TWX: 910-860-5617

Inte! Corp.
7322 S. W. Freeway
Suite 1490
Houston 77024
Tel: (713) 968-8086
TWX: 910-881-2490

Industrial Digital Systems Corp.
5925 Sovereign
Suite 101
Houston 77036
Tel: (713) 998-9421

Inte! Corp.
133 E. Anderson Lane
Suite 314
Austin 78752
Tel: (512) 454-3282

UTAH
Inte! Corp.
298 West 400 South
Salt Lake City 84101
Tel: (801) 533-8086

VIRGINIA
Inte! Corp.
1803 State Road
Suite 109
Richmond 23288
Tel: (804) 282-5608

WASHINGTON
Inte! Corp.
111 11th Avenue N.E.
Suite 510
Bellevue 98004
Tel: (206) 453-8086
TWX: 910-443-3002

WISCONSIN
Inte! Corp.
450 N Sunnyslope Road
Suite 130
Brookfield 53005
Tel: (414) 784-5060

*Field Application Location
EUROPEAN DISTRIBUTORS/REPRESENTATIVES

AUTRIA
Bacher Elektronische Gerate GmbH Rottenwegasse 26 A-1120 Vienna Tel: (222) 83 63 96 TELEX 11352 BASAT A

BELGIUM
Inrico Belgium S.A. Rue des Cieux 505 B-1190 Brussels Tel: 02 231 93 93

DENMARK
Multiplex A/S Selsmakergade 30 DK-2600 Glostrup Tel: 08 265 66 45 TX 33358

FINLAND
Mikro Instruments AB Mekononkatu 24 A SF 00210 Helsinki 21 Tel: 09 692 60 22 TELEX 124 224 FIN

FRANCE
GÉRÉRIM 21, des Courtois à 95700 La Faye Tel: 01 39 92 71 00

GERMANY
Electronic 2000 Vertriebs A.G Neumannstrasse 75 D-8000 Munich 80 Tel: 089 43 40 61 TELEX 525561 EIEC D
Jenoptik GmbH Postfach 1180 D-28227 Bremen Tel: (0421) 253 92 93

NETHERLANDS
Koning & Harman Koperweerstraat 30 B-1050 Brussels Tel: 02 547 52 30 TELEX 168901 KONH NL

ITALY
Intersil Corporation Italia Spa Via Nazionale 30094 Ascoli Piceno Tel: 073 82 00 30 TELEX 130151 ITALI

SWEDEN
Intersil Sweden A B Box 20902 S-161 87 Bromma Tel: 08 98 53 84 TELEX 126015 SWI

SWITZERLAND
Intersil Corporation S.A. Kornhausstrasse 10 CH-8002 Zurich Tel: 01 372 29 00 TELEX 502377 STS

UNITED KINGDOM
Intersil Corporation (U.K.) Ltd. 5 Tolworth Park Road Surbiton, Surrey KT5 0ST Tel: 01 871 5500 TELEX 648022

YUGOSLAVIA
H. R. Microelectronics Enterprises P.O. Box 56017 San Jose, California 95150 Tel: 408 488 0200 TELEX 278559