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

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

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

- 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

- Multiple Coprocessor Interface
- 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 Superscalar 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.

Superscalar capabilities, such as hardware-supported multitasking and virtual memory support, provide the foundations necessary to build advanced multitasking and multiuser systems. The hardware mult-tasking 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 cache 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
rest of the CPU execution adds to the overall performance improvement.

**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><strong>Table 1 - Data Types</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Arithmetic 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 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 Variable length: 0 to 4 Gigabytes</td>
</tr>
<tr>
<td>Bit Field 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>Table 2 - Memory Addressability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical Memory</td>
</tr>
<tr>
<td>Segment Offset</td>
</tr>
<tr>
<td>Virtual Memory per task</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.
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>CS, SS, DS, ES, FS, GS</td>
</tr>
<tr>
<td>None, or 16-bit Base Register</td>
<td>None, or 32-bit Base Register</td>
</tr>
<tr>
<td>+ Optional BASE</td>
<td></td>
</tr>
<tr>
<td>None, or 16-bit Index Register</td>
<td>None, or 32-bit Index Register</td>
</tr>
<tr>
<td>+ Optional INDEX</td>
<td></td>
</tr>
<tr>
<td>None, 8, or 16-bit</td>
<td>None, 8, or 32-bit</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>— Extra Software</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 multimegabyte 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></td>
<td>— Many memory resident tasks</td>
</tr>
<tr>
<td>• One segment may represent entire code or data structure</td>
<td>• Swapping and physical memory management tool</td>
</tr>
<tr>
<td></td>
<td>— Invisible to applications programmer</td>
</tr>
<tr>
<td>• Tool for applications programmer</td>
<td></td>
</tr>
</tbody>
</table>

231052-001
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.

DOCUMENTATION
Further information is provided in the following documents:
Introduction to the iAPX 286, #210308-001
iAPX 286 Programmer’s Reference Manual, #210498-002
iAPX 286 Operating System Writer’s Guide, #121960-001

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 Super-micro 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.
# U.S. SALES OFFICES

**ALABAMA**
Intel Corp.
303 Williams Avenue, S.W.
Suite 1423
Huntsville, 35801
Tel: (205) 333-9353

**ARIZONA**
Intel Corp.
1122 S. 28th Drive
Suite 2140
Phoenix 85029
Tel: (602) 899-4980

**CALIFORNIA**
Intel Corp.
1010 Hurley Way
Suite 300
Sacramento 95825
Tel: (916) 929-4078

Intel Corp.
1500 Opportunity Road
Suite 125
San Diego 92111
Tel: (714) 268-3563

**COLORADO**
Intel Corp.
4445 Northpark Drive
Suite 100
Colorado Springs 80907
Tel: (303) 594-6622

Intel Corp.
600 S. Cherry Street
Suite 720
Denver 80202
Tel: (303) 321-8086

Tel: (303) 931-2289

**CONNECTICUT**
Intel Corp.
38 Paddock Road
Danbury 06810
Tel: (203) 792-8966

Tel: (203) 710-455-1199

EMC Corp.
393 Center Street
Wallingford 06492
Tel: (203) 265-5691

**FLORIDA**
Intel Corp.
1500 N. W. 62nd Street
Suite 104
Ft. Lauderdale 33309
Tel: (305) 771-9600

Tel: (305) 955-9407

Intel Corp.
500 N. Maitland
Suite 205
Maitland 32711
Tel: (305) 628-2393

Tel: (813) 853-9219

**GEORGIA**
Intel Corp.
3200 Holcomb Bridge Road
Suite 225
Norcross 30092
Tel: (404) 449-0541

**ILLINOIS**
Intel Corp.
2550 Golf Road,
Suite 815
Rolling Meadows 60008
Tel: (217) 891-7200

Tel: (815) 651-5881

**INDIANA**
Intel Corp.
9100 Purdue 2090
Suite 400
Indianapolis 46268
Tel: (317) 876-0550

**IOWA**
Intel Corp.
St. Andrews Building
1930 St. Andrews Drive N.E.
Center Rapids 52402
Tel: (319) 933-5510

**KANSAS**
Intel Corp.
8400 W. 11th Street
Suite 175
Overland Park 66210
Tel: (913) 642-8080

**LOUISIANA**
Industrial Systems Corp.
2322 Seventh Avenue
Suite 205
Metairie, LA 70001
Tel: (504) 831-8492

**MAINE**
Intel Corp.
7225 Parkway Drive
Hanover 27075
Tel: (301) 796-7500

Tel: (310) 862-1944

**MASSACHUSETTS**
Intel Corp.
27 Industrial Avenue
Chelmsford 01824
Tel: (978) 284-1800

Tel: (781) 343-6333

EMC Corp.
385 Elston Street
Newton 21184
Tel: (617) 264-4740

Tel: (978) 255-2351

**MICHIGAN**
Intel Corp.
2500 Northwestern Highway
Suite 401
Southfield 48075
Tel: (313) 922-9199

Tel: (734) 244-9195

**MINNESOTA**
Intel Corp.
2350 W. 50th Street
Suite 360
Bloomington 55431
Tel: (612) 835-8722

Tel: (952) 576-2867

**MISSOURI**
Intel Corp.
4203 Earth City Expressway
Suite 131
Earth City 63045
Tel: (314) 291-1990

**NEW JERSEY**
Intel Corp.
30 Princeton Plaza II
Piscataway 08854

Tel: (201) 225-3200

Tel: (732) 461-3258

**NEW MEXICO**
Intel Corp.
1120 Juan Tabo N.E.
Albuquerque 87112
Tel: (505) 253-8092

**NEW YORK**
Intel Corp.
200 Washington Street
Poughkeepsie 12601
Tel: (914) 747-2303

Tel: (518) 246-0060

**OKLAHOMA**
Intel Corp.
4157 S. Harvard Avenue
Suite 22
Oklahoma City 73135
Tel: (405) 717-8688

Tel: (918) 467-8741

**OMAHA**
Intel Corp.
15730 W. Beachwood Drive
Suite 724
Cleveland 44122
Tel: (216) 484-6195

Tel: (810) 471-9298

**OREGON**
Intel Corp.
10101 South W. 50th Street
Suite 360
Bloomington 55431
Tel: (612) 835-8722

Tel: (952) 576-2867

**PENNSYLVANIA**
Intel Corp.
510 Pennsylvania Avenue
Fort Washington 19034
Tel: (215) 461-1000

Tel: (512) 833-8579

**TEXAS**
Intel Corp.
3200 Research Forest Drive
Suite 100
Dallas 75246
Tel: (214) 284-8087

Tel: (713) 988-3800

Tel: (512) 858-9000

**UTAH**
Intel Corp.
256 West 1300 South
Salt Lake City 84101
Tel: (801) 503-8066

**VIRGINIA**
Intel Corp.
16001 Santa Rosa Road
Suite 109
Richmond 23298
Tel: (804) 282-5668

**WASHINGTON**
Intel Corp.
110 11th Avenue N.E.
Suite 510
Bellevue 98004
Tel: (206) 453-8086

Tel: (206) 453-8086

Tel: (910) 443-3002

**WISCONSIN**
Intel Corp.
450 N. Sunnyslope Road
Suite 130
Brookfield 53005
Tel: (414) 784-9060

*Field Application Location*
EUROPEAN DISTRIBUTORS/REPRESENTATIVES

AUSTRIA
Bacher Elektronische Geräte GmbH
Rodtenkirchen 26
A-1120 Vienna
Tel: (222) 83 63 96
TELEX 11522 BASAT A

BELGIUM
Inciso Belgium S.A.
A. de la Hulpe 26
B-1130 Brussels
Tel: (2) 687 40 21
TELEX 170 241

DENMARK
A/S Nykredit Kompagni
D-2900 Gentofte
Tel: (39) 37 57 98
TELEX 05019 NYKED D

FINLAND
Synety Oy
Keskuskatu 2
FIN-00280 Helsinki
Tel: (10) 708 36 54
TELEX 27521 JEF D

FRANCE
Bacchus S.A.
11 rue de la Bourse
75001 Paris
Tel: (1) 55 83 57 75
TELEX 09819 BAC D

GERMANY
Siemens AG
Postfach 20 09 20
D-1000 Berlin 20
Tel: (90) 12 00 00
TELEX 21935 SIEM D

ITALY
Inseco A.S.
P.O. Box 1600
I-41100 Bologna
Tel: (051) 44 01 23
TELEX 41121 INSE D

SWITZERLAND
Norelco SA
Case Postale 309
CH-1950 Horgen
Tel: (1) 33 22 12
TELEX 21935 SIEM D

UNITED KINGDOM
Bacci Ltd.
Shepperton, Middx TW17 7QA
Tel: (01) 789 56 78
TELEX 21935 SIEM D