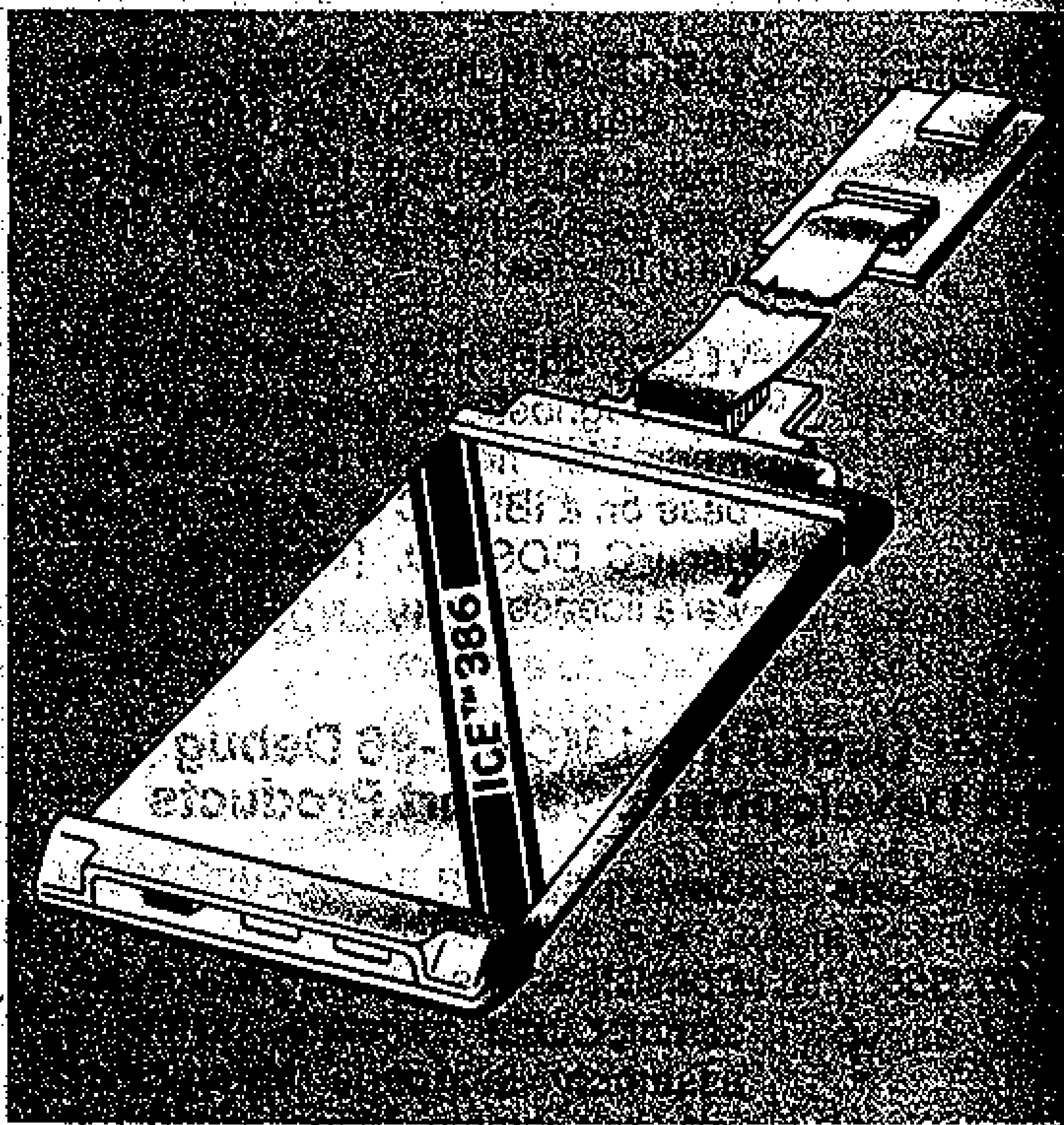
ICE™-386 In-Circuit Emulator For the 80386

The ICE™-386 is the In-Circuit Emulator for the 80386 processor. It helps a design team meet its most critical schedules by increasing engineering productivity and by allowing parallel hardware and software development. The ICE-386 emulator increases engineering productivity by leveraging Intel's bondout technology to provide unique access to the 386 internals. This window into the microprocessor becomes even more important for the 32-bit architectures because of the high degree of pipelining and caching that is invisible to the outside world. And before the prototype hardware is stabilized or even functional, one can utilize the ICE-386 emulator's loopback mode to start debugging software.



Product Highlights

- —Up to 16 MHz emulation allowing detection of timing problems at 16 MHz
- —Four on-chip breakpoints which can halt a user program on specific execution addresses or data acceses to permit the user to examine the processor status
- —128K bytes of mappable memory to make working memory space available to the user prototype
- —Execution address trace buffer to store over 2000 frames of program execution history
- —Ability to interrogate and modify processor objects and system memory
- —High-level language debug with symbolics to aid in program development. The emulator is integrated with Intel compilers to provide sophisticated debug information
- —Non-intrusive to user prototype memory and interrupt structure
- —Emulation timer and execution trace time stamps to analyze time taken by user code
- —Loaders for 8086, 80286, and 80386 Object Modules
- —Complete coprocessor support for both the 80287 and the 80387
- —Advanced human interface with command syntax guidance and on-line help

Product Description

The Intel ICE-386 In-Circuit-Emulator provides sophisticated hardware and software debug capabilities for 80386 based designs. These capabilities include In-Circuit-Emulation for the 80386 central processing unit, and the 80287 and the 80387 numeric coprocessors. Using ICE-386 emulator, 80386 software can be executed in continuous or single-step modes. Breakpoints allow stopping emulation on specified instruction execution or data access, and the trace capability records the program execution history prior to the break. Through ICE-386 emulation, access to user software and user data structures may be done symbolically by referring to the program variables.

16 MHz Emulation

ICE-386 allows up to 16 MHz emulation of the 80386, thus providing early detection of the subtle timing problems that may arise at full speed. Intel's bondout technology also stresses the tightest possible conformance between timing parameters of the emulator and of the target processor.

On-Chip Breakpoints

The 80386 processor has four on-chip word recognizers that can be programmed to halt processor emulation at either instruction execution or data access at specified locations. Since these word recognizers are on the 80386 chip, the processor can continue real-time execution up to the last instruction.

128K Byte Map Memory
The ICE-386 In-Circuit-Emulator can allocate 128K bytes of its memory in 4K-byte blocks to overlay to the user address space. It allows the prototype system to borrow memory resources from the ICE-386 emulator. If prototype memory is not yet available, the user program may reside in the emulator's RAM memory. Because this memory is RAM, changes can be made quickly and easily. Also, if the prototype contains EPROM, ICE-386 memory can overlay the EPROM memory space so that the EPROM need not be programmed during development.

Execution Trace
ICE-386 emulation has the capacity of storing over two thousand frames of program execution history.

Interrogate/Modify Processor and Memory
The ICE-386 emulator lets the software developer interrogate and modify all processor data structures in the symbolic form. The user memory can be read disassembled and can be modified using the single-line instruction assembler.

Symbolic Debugging
The ICE-386 emulator also allows the software developer to examine or modify memory locations using symbolic references. A symbolic reference is a procedure name, variable name, line number, or program label that corresponds to a location in the program space. Variables contain type information such as byte, word, record, array, array of records, etc. Symbolic debugging lets the user work in the context of the original program, helping meet the most critical schedules.

Non-Intrusive
The Intel ICE-386 In-Circuit-Emulator employs the special bondout technology to make emulation possible without invading the prototype's RAM space and interrupt structure. Your debugged prototype will confidently lead you to a reliable product in the market.