Distributed processing with iAPX 186 microprocessor systems

by TONY ZINGALE

Intel Corporation
Santa Clara, California

ABSTRACT

In most early computer systems, large central computers, minicomputers, or microcomputers were used to perform all the necessary data processing activities in the system. The overall performance of the entire system was limited by the ability of the central CPU to bring in data, process it, and output it in some usable format. This often resulted in large data input/output bottlenecks in I/O-intensive applications where the CPU time required to service I/O functions left little time for data processing. The obvious result is a slow, non-optimum data processing system. Now many applications are moving in the direction of simpler and easier-to-use distributed systems, where a central CPU delegates some of the processing tasks to distributed processing subsystems. Not only are costs lower with the distributed system approach, but the time needed to implement such systems is substantially less.

For example, a network transaction processing system can now use numerous automatic tellers to process data at a variety of dispersed geographic locations. The tellers, or distributed nodes, can then collect, process, output, and eventually pass on necessary information to the central host computer, located at some detached location, without burdening the central computer with handling each simple transaction. The host becomes involved with an individual node only when the intelligent node requires the timely interaction. The heart of a distributed node itself must be an intelligent processing device capable of handling all the processing and I/O requirements needed by the node.

The iAPX 186 is a new highly integrated 16-bit microprocessor. It combines 10 of the most common microprocessor system components onto one. The 80186 is essentially a 16-bit CPU board integrated onto a single silicon chip. By combining a limited number of peripheral support components with memory together with an iAPX 186, one can achieve a condensed, cost-effective system on one board, making the 80186 an optimal microprocessor for distributed processing nodes. This high level of integration is accomplished through an advanced HMOS II silicon gate technology. For the first time it provides a system cost saving significantly greater than that of the previous 16-bit microprocessor design alternatives. The 80186, an upgrade from the industry standard iAPX 86 and 88, offers two to three times the system throughput of a standard iAPX 86. The iAPX 186 adds 10 new instruction types to optimize existing iAPX 86 or 88 application code or streamline new iAPX 186 application code. All these hardware and software attributes make distributed processing with iAPX 186 systems a cost-effective, easy 16-bit microprocessor solution.
CLASSICAL DISTRIBUTED PROCESSING

The concept allows the use of dispersed processing sites or nodes to offload a sophisticated central computer, minicomputer, or microcomputer. The real payoff from the distributed processing approach is the increased responsiveness to the user's needs of the data processing function, achieved by providing an effective, fast, powerful processing mechanism at the lower levels. The declining costs of microcomputers and memories have provided the economic justification for distributed computing. The approach now is to let microcomputers located near the data do much of the real-time processing and send only a summary to the host computer (Figure 1).

Distributed Processing Node Requirements

The data processing node must be

1. More cost-effective than similar approaches
2. Easy to implement, thus making possible a fast end-product time to market
3. Compatible with existing software, if any
4. Capable of high-speed execution rates

In addition to the general requirements stated above, there are a set of hardware requirements to be satisfied.

1. High-speed, flexible DMA is needed by any I/O subsystem to accomplish data transfers between I/O devices connected to the distributed node (i.e., keyboards, disks, printers, modems) and local system memory or vice versa. This is a key requirement for moving blocks of data in and out of a distributed node that can improve system performance and execution time.
2. Flexible hardware timers are always required to time external events occurring in the system. Timed external events usually correspond to some synchronized system activity. For example, the number of words that have been printed may signify to the CPU that it needs to start a new page or generate some type of interrupt to the CPU to stop printing.
3. In a time-sensitive distributed system there is a definite need for the handling of a large number of real-time interrupts. For example, if several intelligent terminals are connected to a single distributed node in addition to the standard I/O devices, multiple interrupts will appear at the node simultaneously. These interrupts must be acknowledged, prioritized, and handled cleanly and rapidly.
4. Address decoding hardware is needed to provide the system with a systematic convention for selecting memory spaces and peripheral devices; wait-state generating circuitry is required to insure timing compatibility with memories and peripherals at the proper speeds. This hardware can require an appreciable portion of the board space of the distributed node.

This feature set is optimal in that it provides all the basic requirements of a distributed processing node. The iAPX 186 integrates these common system functions into a single silicon chip.

Cost-Effective, Optimal Integrated Feature Set

A block diagram of the iAPX 186 integrated hardware feature set is shown in Figure 2, followed by a summary of each on-chip feature.

Clock generator: The 80186 provides an internal clock oscillator, which requires a single external crystal or TTL-level frequency source. The system clock output is a standard 8-MHz, 50% duty cycle clock at half the crystal frequency, or 16 MHz. This output can be used to drive the clock inputs of other system components and hence make additional clock generation devices unnecessary. Synchronous and asynchronous ready inputs are supplied for flexible peripheral-device synchronization.
Timers: Two independent 16-bit programmable timer/counters are provided to count time external events, external events, and generate nonrepetitive waveforms. A third 16-bit programmable timer, not connected externally, is useful for implementing time delays and as a prescaler for the two externally connected timers. The iAPX 186 integrated timers are very flexible and can be configured to time/count a variety of distributed I/O types of activities.

Each of the three timers is equipped with a 16-bit timer register that contains the current value of the timer. It can be read or written at any time, independent of whether the timer is running. Each timer is also equipped with a 16-bit max count register containing the maximum value the timer will reach. In addition, the two externally connected timers each have a second 16-bit max count register, which enables the timers to alternate their count between two different max count values as programmed by the user. When a terminal count is reached, an interrupt may be generated, and the timer value is reset to zero.

The timers have several flexible programmable options in their mode of operation. All three timers can be set to halt or continue on a terminal count value, so no external event or device need wait for a timer reset. The two externally connected timers can select between internal and external clocks, alternate between max count registers or use only one, and be set to retrigger on external events.

DMA channels: The on-chip DMA controller unit in the iAPX 186 contains two independent high-speed DMA channels. DMA transfers can occur between memory and I/O spaces (i.e. M-I/O) or within the same space (i.e. M-M, I/O-I/O). The latter feature allows I/O devices and memory buffers to be freely located anywhere in the distributed system. For example, memory-mapped I/O can be handled without any external decode logic to select the required I/O space or device. Each DMA channel maintains two 20-bit source and destination pointers that can be incremented, decremented, or left unchanged after each transfer. Data transfers are programmed by the user to be either byte or word transfers and can occur anywhere in the 1 megabyte of directly addressable memory space. This allows a maximum transfer rate of 1 MWord/second or 2 MBytes/second. The user can specify several different modes of DMA operation via the on-chip 16-bit DMA channel control word.

By using the 80186 DMA facilities, data can be input onto local system memory, processed, passed on to the host computer (if needed), and output to another I/O device, all by the use of the two independent, high-speed, on-chip DMA channels.

Interrupt Controller: The 80186 interrupt controller resolves priority among interrupt requests that arrive simultaneously. It can accept interrupts from up to five external hardware sources (NMI + 4) and internal sources as well (timers, DMA channels). Each interrupt source has a programmable priority level and a preassigned interrupt vector type, used in deriving an address to a table in memory where interrupt service routine addresses are located. This enhancement of predefined vector types makes the interrupt response time about 1.5 times faster than the typical iAPX 86 response time. The 8259A programmable interrupt controller (PIC) interrupt modes, like fully nested and specially fully nested, are provided by the 80186 as well. In addition, multiple 8259As can be cascaded to provide the system with up to 128 external interrupts. There is also an RMX-86 real-time operating system mode of operation for maximum user flexibility that provides many of the same interrupt features described here.

Chip select/ready generation: The iAPX 186 contains programmable chip select logic to provide chip select signals for memory components, peripheral components, and programmable ready (wait states) generation logic. The result of this integrated logic is a lower system part count, since as many as 11 TTL packs will be saved. In addition to a lower system cost, the speed/timing performance of the system will improve as a result of the elimination of external propagation delays. Another advantage involves flexibility in the choice of memory component size and speed. Three memory ranges (lower, middle, upper) can be programmed to variable lengths (1K, 2K, 4K, ..., 256K) so that a variety of memory chip sizes can be used. Further, anywhere from zero to three wait states can be programmed so either high-speed or low-cost, slower memories can be used. With respect to the peripheral chip selects, as many as seven different peripheral components can be addressed via I/O or memory space. Again, programmable wait states may be injected to synchronize slower peripherals with the 80186 itself or memory.

The chip select/ready logic contributes heavily to making the iAPX 186 an optimum, low-cost choice for a distributed processing node. In the past, this necessary logic had to be designed, debugged, and programmed. Now, with the 80186, the design, debug, and programming are done by initializing the associated 16-bit on-chip control registers.

CPU internal registers: The added functionality of the iAPX 186 (i.e., timers, DMA, interrupt controller, and chip selects) uses on-chip 16-bit control registers for each integrated device. They are contained in a 256-byte control block (see Figure 3) included in the 80186 CPU register architecture. The control register block may be either I/O or memory-mapped, based on initialization for a new control block pointer in the CPU. Except for these additions, the register architecture of the iAPX 186 is identical to the iAPX 86.
Software Compatibility

Since software costs are influencing most microcomputer decisions today, system designers must take this enormous investment seriously when choosing microcomputers for future product upgrades. This is especially true in the cost-sensitive distributed processing area, where virtually hundreds of nodes will be designed and programmed to interface to a central host computer. Software compatibility between the nodes and the central host makes the overall system easier to use and will shorten the design cycle considerably. For future product upgrades, software compatibility must be a decision variable in today's product. If not, when bringing a new product to market, engineers may spend all of their time rewriting hundreds of lines of general-purpose software rather than writing new streamlined application code. All this can be saved by using the iAPX 186. Since the 80186 is completely object-code-compatible with the iAPX 86 or 88 or 286, software investments are intact for future product offerings. Not only is the 80186 totally software-compatible with the 8086 or 8088; it adds 10 new instruction types as well. Instructions like block move (running at bus bandwidth or 2MBytes/sec), push or pop all the registers (push/pop all), and multiply immediate are all new to the basic iAPX 86, 88 instruction set. These instructions help enhance existing iAPX 86 or 88 application code, if needed, or produce optimum, high-speed iAPX 186 code.

iAPX 186 Performance Comparisons

The iAPX 186 overall performance speed is two to three times faster than the 5MHz iAPX 86 and 30% faster than the 8MHz iAPX 86. Many instructions, specifically those for integer arithmetic (i.e., multiply and divide), execute 5 to 6 times faster than on a 5MHz iAPX 86 (see Table I). In benchmarks based on Intel standard applications, operations like block translation, bubble sort, and automated parts inspection show that the iAPX 186 yields a 1.66 times performance increase over the 8MHz iAPX 86 (see Table II). These benchmarks were selected to evaluate the performance of 16-bit microprocessors and demonstrate the capabilities necessary for intensive I/O operations, general integer arithmetic, and data manipulation operations necessary for real-time business and EDP applications. Naturally the most likely environment for finding a distributed processing system lies in these application areas. The iAPX 186 satisfies the high-speed execution requirement for a distributed node by surpassing the existing high-performance standards set by the iAPX 86 and at the same time is totally software-compatible to the iAPX 86, 88, and 286.
TYPICAL DISTRIBUTED SYSTEM CONFIGURATION

A sophisticated central host computer capable of handling multiple users in a real-time environment is obviously a major need for an effective distributed processing system. This device is responsible for controlling all the distributed nodes in the system. This requires an extremely large memory space to handle the multiple-nodes memory and I/O space requirements and also requires some form of system integrity mechanism that would insure that each node executes independent of the others. The microcomputer that fits this requirement best is the iAPX 286 (see Figure 4). Not only is the 80286 software compatible with the iAPX 86, 88, 186, providing six times the performance of an 5 MHz iAPX 86; it also offers on-chip memory management and memory protection. The iAPX 286 is capable of directly managing up to 16 megabytes of real memory and up to 1 gigabyte (2^30 bytes) of virtual memory. It can provide memory protection for each distributed node by verifying each specific task's address range and access rights for every memory access. These integrated features of the 80286 satisfy the requirements of a central host and of controlling the distributed nodes in a system, since each will require some independent memory space and also some form of protection from the other nodes in the system.

As Figure 4 shows, communications between the iAPX 286 host computer and the iAPX 186 distributed nodes takes place by passing messages and data through a dual-port RAM. The dual port is used to isolate the iAPX 186 systems or nodes from the protected bus structure of the iAPX 286, maintaining full system integrity.

One design variable to consider in a distributed node scheme is error detection and correction in and out of the dual-port RAM. The Intel 8206 Error Detection and Correction unit performs this function with one device. The 8206 serves as an interface between large memory systems (i.e., iAPX 286 systems) and the system bus of the iAPX 186. The EDC unit will internally detect all one-bit errors and most multiple-bit errors and automatically make corrections. Obviously, errors can occur in any system configuration when data are written incorrectly to memory, a memory cell loses data, or a complete memory component is missing or dead. These errors can be carried throughout the system and affect end results unless detected and corrected. In Figure 4 a dual-port RAM scheme is used to interface the iAPX 286 protected system bus to the iAPX 186 local bus. The 8207 Advanced Dynamic RAM Controller is capable of controlling two memory ports at the 8-MHz speed for both microcomputers and supporting a megabyte of address space. The 8207 provides the necessary control and timing signals to interface memory to the 8206 EDC component as well (see Figure 5). Previously this mechanism, the combined 8206 and 8207 components, took as many as 50 TTL components. Together the two peripheral devices provide a cost-effective, error-free, highly reliable memory subsystem for a distributed processing node.

CONCLUSIONS

The iAPX 186 exceeds all the stated requirements for use as an effective distributed processing node. This optimal integrated feature set of the 80186 is streamlined to manage the necessary I/O hardware and real-time/high-speed software needs of a distributed system. It is very cost-effective, easy to use, high-performance, and compatible with any iAPX 86 or 88 existing software; and it can also be tightly coupled with an iAPX 286 central host and provide highly reliable memory subsystems through the use of the 8206 EDC and the 8207 peripheral devices.

REFERENCES


From the collection of the Computer History Museum (www.computerhistory.org)