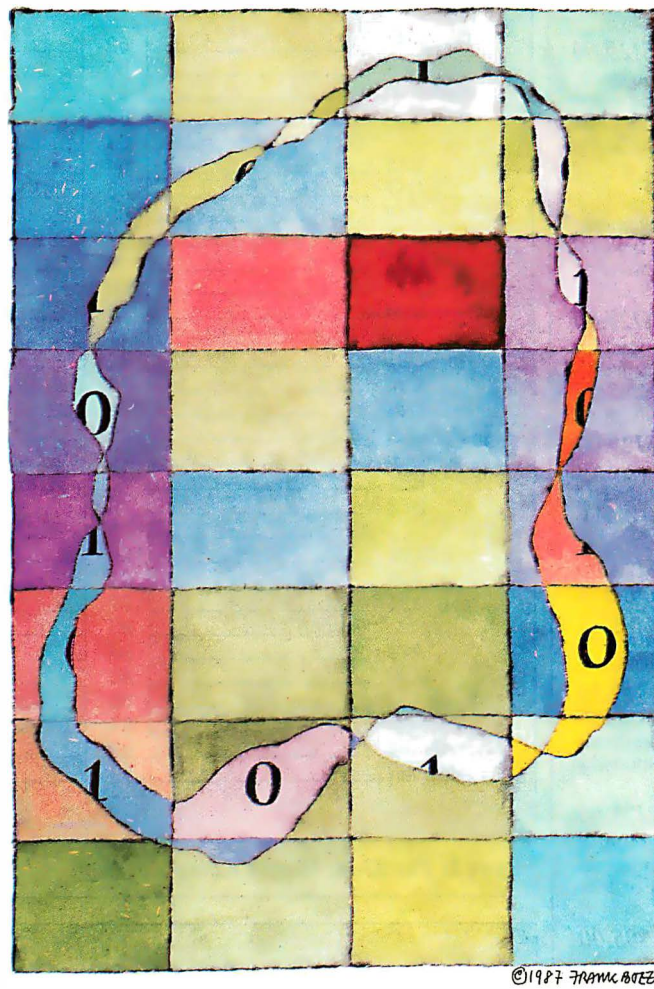
The Personal System/2 Model 80 takes the 32-bit road to more power, more memory, more speed

The 32-bit Micro Channel

Jon Shiell

The 32-bit Micro Channel opens a window on IBM's future. The three 32-bit bus slots in the Personal System/2 Model 80 support the 80386 processor with 32-bit addressing and 32-bit data transfers. The direction in which IBM's personal computers are headed is no longer implied; it is defined: the 32-bit road to more power, more memory, more speed. No big surprise—but the question of which way IBM will go and how it will get there is always of major interest to anyone seriously involved with personal computers.



The Personal System/2

As I write this, the new IBM PS/2 family consists of five basic machines, three of which are based on a proprietary new bus called the Micro Channel. The Models 25 and 30 are 8-megahertz 8086-based desktop machines that use an IBM PC XT bus with three slots. The next two are 10-MHz 80286-based machines: the Model 50, a desktop model with four 16-bit Micro Channel slots, and the Model 60, a floor-standing model with seven 16-bit Micro Channel slots. Finally, the Model 80 is an 80386-based floor-standing machine available in both 16-MHz and 20-MHz versions with three 32-bit and four 16-bit Micro Channel slots.

Previous BYTE articles have covered the Model 30 (see "The IBM PS/2 Model 30" by Curtis Franklin Jr., July); the Models 50 and 60 (see "The IBM PS/2 Model 50" and its accompanying text box on the Model 60, by Richard Grehan, July); the 16-bit Micro Channel (see "Under the Covers" by Steve Ciarcia, August); and information on the PS/2 in general (see "First Impressions: The IBM PS/2 Computers," June).

Because of this heavy coverage, I'll focus on the unique features of the Model 80's 32-bit bus and not attempt a comprehensive overview of the Micro Channel. Information for this article came from a variety of sources including the *IBM Personal System/2 Model 80 Technical Reference*.

The 32-bit Difference

The Model 80's Micro Channel differs from that of the Model 50 and 60 in that, in addition to 16-bit slots, it also has 32-bit slots. The Micro Channel connector for the 32-bit extension extends the 16-bit Micro Channel connector to accommodate 32-bit addressing and 32-bit data transfers (see figure 1). It connects to the "bottom" of the 16-bit extension (see figure 1 in "Under the Covers," August, page 104).

The 32-bit extension itself consists of the seven control lines for 32-bit data (-BE0 through -BE3, -CD DS 32, -DS 32 RTN, and TR 32), eight additional address lines (A24 through A31), and 16 data lines (D16 through D31).

Lines -BE0 through -BE3 ("byte enable" 0 through 3) are used during 32-bit slave data transfers to tell the bus which bytes are to go on it. Line -CD DS 32 (card data size 32) indicates that the data port at the location addressed is a 32-bit data port. Line -DS 32 RTN (data size 32 return) is a negative OR of -CD DS 32 and provides a check for the channel on data-size information. TR 32 (translate 32) provides an indicator as to what logic is driving -BE0 through -BE3. If TR 32 is inactive, the 32-bit bus master is in charge of these lines; if it is active, the central-translator logic is driving them.

continued

Jon Shiell is a contributing editor for BYTE (One Phoenix Mill Lane, Peterborough, NH 03458). He can also be reached on BIX as jshiell.

Illustration: Frank Bozzo © 1987

There is also a Micro Channel connector for the matched-memory extension. This connector extends the 32-bit Micro Channel connector to accommodate matched-memory cycles (see figure 2). It connects to the "top" of the 8-bit section of the Micro Channel in the same position that an auxiliary video extension might occupy on a 16-bit Micro Channel.

The matched-memory-cycle section consists of three signals, -MMC, MMCR, and -MMC CMD. Signal -MMC (matched-memory cycle), driven by system-board logic, indicates that the CPU is in control of the bus and can run a matched-memory cycle. Signal -MMCR (matched-memory-cycle request) is driven by a slave on either the 16-bit or 32-bit channel to request a faster cycle. Since the 80386 is the only controlling device allowed to run matched-memory cycles, if an 8-bit or 16-bit channel slave—or a 32-bit slave in a nonmicroprocessor bus cycle—requests -MMCR, the system will run a basic-transfer cycle. Signal -MMC CMD (matched-memory-cycle command) defines when the data on

continued

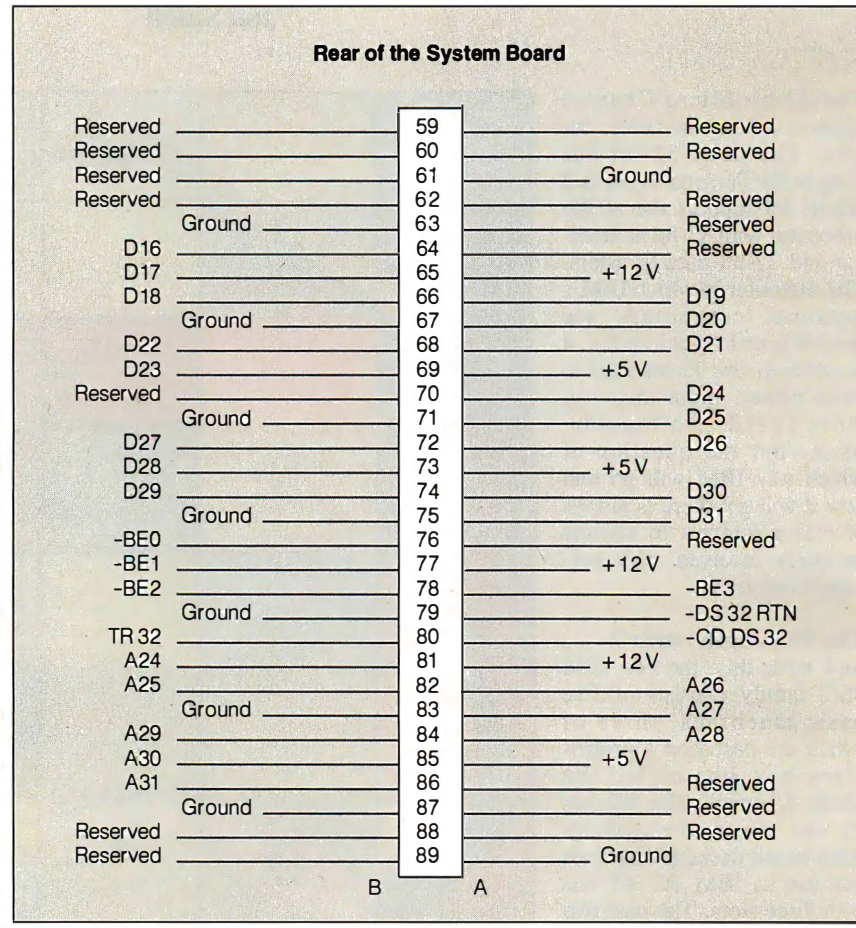


Figure 1: *The 32-bit Micro Channel extension.*

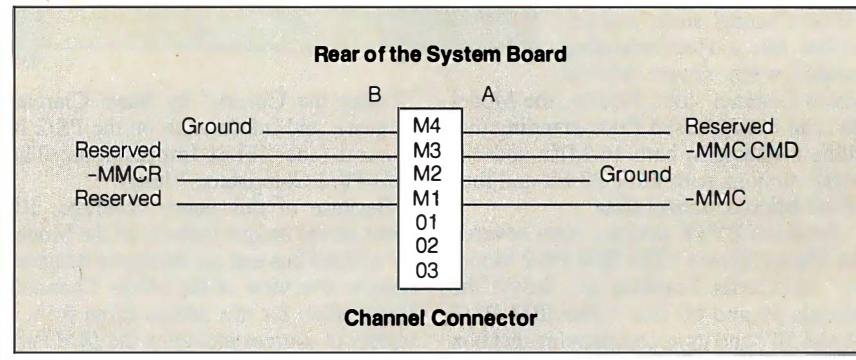


Figure 2: *The matched-memory extension to the 32-bit Micro Channel.*

The Micro Channel uses the 80286's addressing scheme for 8-bit and 16-bit transfers and the 80386's scheme for 32-bit transfers.

the bus is valid, but only during a matched-memory cycle. Together, these signals extend the Micro Channel to use full 32-bit addresses and data.

Photos 1a through 1c provide a telescopic view of the Micro Channel starting with an inside view of the Model 80 (photo 1a), zooming in to a closer view that contains the Micro Channel on the left and shows the 80386 and optional 80387 chips on the right (photo 1b), and ending with a close-up of the Micro Channel (photo 1c). Photo 1c shows the three 32-bit channel connectors, 1, 2, and 4 (counting from the bottom); notice the short extension on the left end of each, the matched-memory extension. Also note the longer extension to the left in slot 6; this is a 16-bit channel connector with the auxiliary video extension.

How the 32-bit Bus Works

The PS/2 Model 80 has special logic, called the address-bus translator, that lets 16-bit devices communicate with 32-bit slaves, and vice versa. The 32-bit slaves (and 32-bit devices) use -BE0 through -BE3 instead of the A0 (address bit 0, the least-significant address bit) and -SBHE (system byte high enable, which indicates and enables data transfer on the high byte of the 16-bit data bus—i.e., D8 through D15) bus lines of the 16-bit bus. A0 and -SBHE are used together to distinguish between high-byte (D8 through D15) and low-byte (D0 through D7) data transfers on the 16-bit bus.

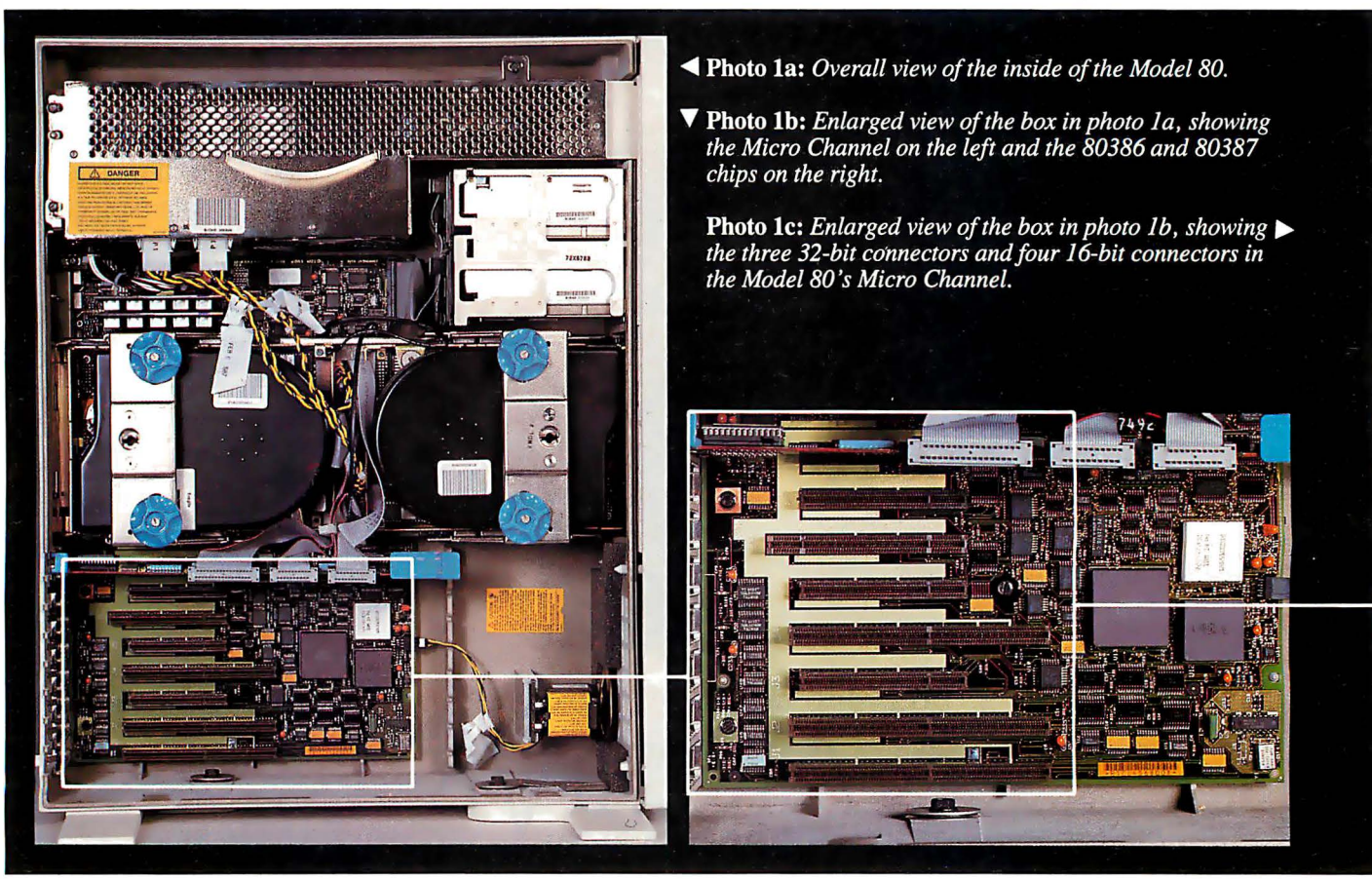
Sixteen-bit and 32-bit transfers use different signals because the 80286 (16-bit) and 80386 (32-bit) system microprocessors address memory differently. The Micro Channel uses the 80286's addressing scheme for 8-bit and 16-bit accesses and the 80386's scheme for 32-bit transfers. Thus, no translation is required when a processor accesses native memory; however, when an 80386 accesses 16-bit memory or I/O, translation is needed.

The signal TR 32 is driven inactive by 32-bit devices; this signal is used by the address-bus translator along with the -CD DS 16 (card data size 16) and -CD DS 32 (card data size 32) signals (which indicate the slave's data size) to determine if translation is required and which party is 32-bit. In addition to the address-bus translator, data-bus-steering logic is required to cross data between D16 through D31 and D0 through D15 because 16-bit devices (and slaves) don't use the high-order 16 data lines.

Four different types of bus cycles are defined for the 32-bit Micro Channel. In order from fastest to slowest, they are matched-memory cycles, basic-transfer cycles, synchronous extended-transfer cycles, and asynchronous extended-transfer cycles. While the Micro Channel is defined as an asynchronous bus, the first three of these cycles are synchronous special cases.

The Matched-Memory Cycle

The matched-memory cycle is a synchronous cycle supported only by the Model 80. It provides the most efficient data transfer between the 80386 and the Micro Channel and is this bus's equivalent to the PC AT's zero-wait-state memory. The system board's ROM, the 80386's RAM, and the 32-bit memory-expansion adapter follow the matched-memory-cycle



◀ **Photo 1a:** *Overall view of the inside of the Model 80.*

▼ **Photo 1b:** *Enlarged view of the box in photo 1a, showing the Micro Channel on the left and the 80386 and 80387 chips on the right.*

Photo 1c: *Enlarged view of the box in photo 1b, showing the three 32-bit connectors and four 16-bit connectors in the Model 80's Micro Channel.* ▶

protocol (i.e., -MMC, -MMCR, and -MMC CMD). A channel slave can be either a memory slave or an I/O slave with either a 16-bit or 32-bit data bus width. However, 8-bit devices cannot run matched-memory cycles, nor can DMA.

Two types of 80386 bus cycles are supported on the Micro Channel for the Model 80: matched-memory cycles and basic-transfer cycles. A matched-memory cycle occupies at least three 16-MHz clock cycles, or 187.5 nanoseconds. A basic-transfer cycle takes at least four 16-MHz clock cycles, or 250 ns. The channel slave must issue an -MMCR request for each bus cycle if it wants a matched-memory cycle; if it doesn't, it will receive a basic-transfer cycle as the default.

If the channel slave issues an -MMCR request, then the system microprocessor responds with -MMC CMD. If the channel slave doesn't return the -MMCR request, the system runs a basic-transfer cycle. These two 80386 bus cycles can be mixed and matched any way you wish; the process is totally dynamic and is determined on a cycle-by-cycle basis.

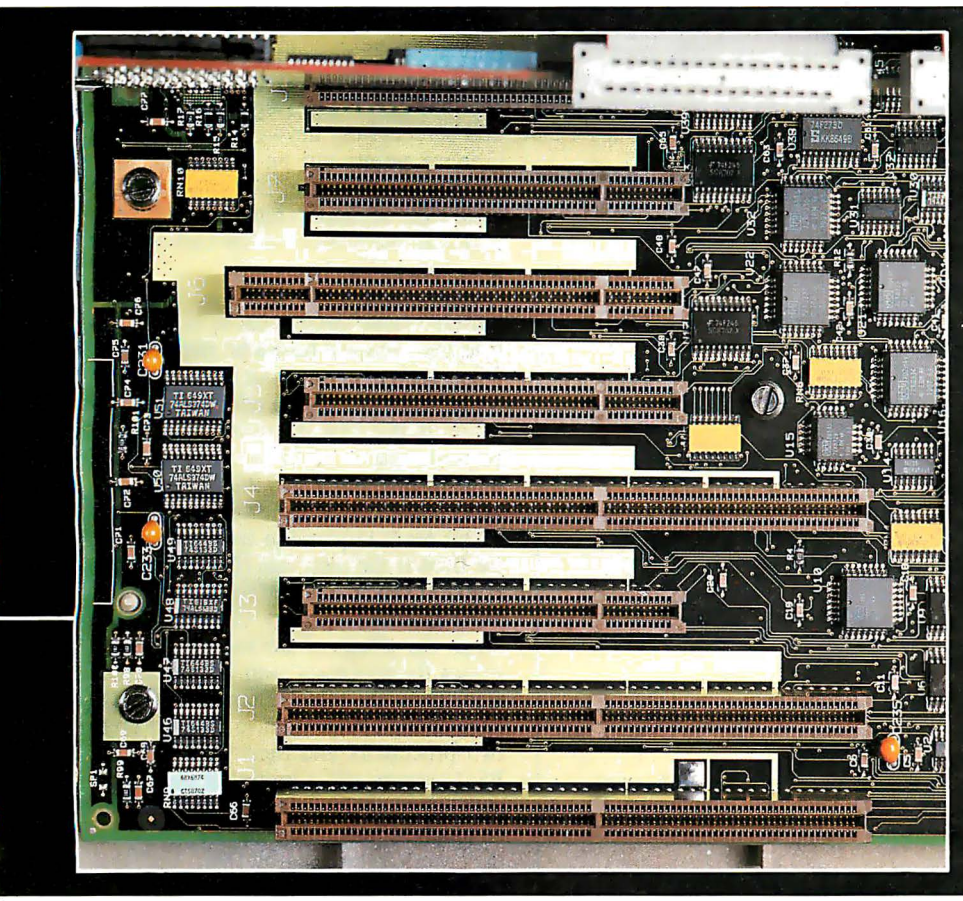
You can extend bus cycles until -CD ChRdy (channel ready) is found active. A memory or channel slave can set -CD ChRdy inactive to allow more time to complete a matched-memory cycle or a basic-transfer cycle when the default length of the cycle is not long enough.

A warning from the technical reference: "When MMC is active, matched-memory-cycle 32-bit and 16-bit devices should not use -MADE 24, A0, A1, or -SBHE in logic that generates -MMCR, -CD DS 16/32, -SEL FBK, and -CD ChRdy."

Variations on a Theme

The line -CD ChRdy is used by slave programs to tell the Micro Channel when their data is ready. It may take as much as 3 microseconds for a slave to make -CD ChRdy active. Data transfers other than matched-memory transfers generally use the same control sequence.

1. The address bus, -MADE 24 (memory-address enable 24, which indicates when an extended address is used on the bus), M/-IO (memory/input output, which distinguishes a memory cycle from an I/O cycle), and -Refresh (if applicable) become active. The cycle begins.
2. The status signals, -S0 and -S1, become active. (Bus addresses must be valid before either -S0 or -S1 becomes active.)
3. The -ADL (address-decode latch) signal becomes active.
4. In response to active -ADL, -MADE 24, and M/-IO, the adapter returns -CD Sfdbk (card-selected feedback, which indicates, when active, that the slave addressed by the system microprocessor is present at the address specified), -CD DS 16 if the attachment can handle 16-bit operation, and both -CD DS 16 and -CD DS 32 if it can handle 32-bit operation.
5. In response to active -ADL, -MADE 24, M/-IO, -S0, and -S1, the adapter drives -CD ChRdy inactive if the bus cycle is too short for the transfer and needs to be extended.
6. For a Write cycle, the Write data appears on the Micro Channel. (Various combinations of M/-IO, -S0, and -S1 indicate whether a transfer is a read or write operation and whether it is to or from memory or I/O.)
7. -CMD (command, which is used to determine when data on the data bus is valid) becomes active, and -ADL becomes inactive.
8. -S0 and -S1 become inactive.
9. Address signals become inactive in preparation for the next bus cycle.
10. In response to an address change, the attachment sets -CD Sfdbk, -CD DS 16, and -CD DS 32 inactive.
11. If -CD ChRdy has been set inactive, the system remains in this state until -CD ChRdy is set active. (This interval should not exceed 3 ms.)
12. For a read cycle, the attachment places read data on the Micro Channel to be ready for the trailing edge of -CMD.
13. The address, -S0, -S1, and M/-IO for the next cycle may become active. (You can overlap activating the address and status indicators with the preceding cycle to minimize the impact that memory-access time makes on performance.)
14. -CMD goes inactive. The cycle ends.



The remaining three bus cycles (the basic-transfer cycle, the asynchronous extended-transfer cycle, and the synchronous extended-transfer cycle) are variations on this theme. The basic-transfer cycle is the default bus cycle and is synchronous. While it requires a minimum of 200 ns, it requires four clock cycles of 62.5 ns each, or 250 ns, in the 16-MHz Model 80. The asynchronous extended-transfer cycle and the synchronous extended-transfer cycle are differentiated by the slave's use of -CD ChRdy. If it is an asynchronous cycle, the slave releases -CD ChRdy asynchronously and provides the read data within 60 ns of the release. Similarly, if the cycle is synchronous, the release is synchronous and occurs within 30 ns of the leading edge of -CMD. While the asynchronous extended-transfer cycle requires a minimum of more than 300 ns, the synchronous extended-transfer cycle is only slightly faster, requiring a minimum of 300 ns.

continued

You could have an 18-megabyte system with one 32-bit slot still available.

A special note: If activation of the status indicators overlaps with the previous -CMD cycle in the two extended cycles, -CD ChRdy is invalid during the overlap. This varies from the control sequence described above.

Details to Remember

The 32-bit Micro Channel is a superset of the 16-bit Micro Channel; thus, POS (Programmable Option Select), arbitration, and timing are the same (except in the matched-memory cycle). One point worth remembering: While bus masters on the Micro Channel can access all *memory* addresses on the system board, they can't access *I/O* addresses less than 100 hexadecimal on the system board; this is true of all Micro Channel systems, not just the Model 80.

While the Model 80's Micro Channel supports 32-bit addresses for the 32-bit slots, the system DMA channels on the Model 80 support only 24-bit addresses. This can be a problem if you want to use more than 16 megabytes of memory; all DMA (i.e., disk I/O and so forth) must be moved to low storage, then moved by the processor to high storage. The memory-remapping facility may provide a way around this by letting you remap banks of memory between high and low addresses. That is, the operating system might reserve, for example, four banks of memory between 8 megabytes and 12 megabytes for remapping; memory remapping in the Model 80 is in units of 1 megabyte. Then, when the operating system needs to perform DMA I/O at a high memory location—the bank at physical address 22 megabytes, for example—it could remap that bank to one of the four banks reserved for remapping, perhaps the one at location 10 megabytes. After the I/O completes, the operating system moves the relocated bank of memory back into its original high-memory position.

This problem is unlikely to arise in the next few years, because OS/2 and other 80286 operating systems can address up to 16 megabytes, as can the 80286 itself. Also, the performance that the 80386 provides, whether at 16 or 20 MHz, pipelined with one wait state, won't require more than 16 megabytes of memory in most cases.

Future Directions?

The three 32-bit slots in the Model 80 are primarily meant for memory cards; assuming the use of 1-megabit dynamic RAM chips, a normal memory card will contain about 8 megabytes of memory. Using two slots for memory and up to 2 megabytes on the motherboard, you could have an 18-megabyte system with one 32-bit slot still available. What can you do with that third 32-bit slot? Well, you could use it for an expansion chassis or another 8 megabytes of memory. But you could also presumably use it for an additional 32-bit processor (with its own memory cache); however, the current Model 80 doesn't have the bus bandwidth to support this.

Let's assume we have a Model 80-71; that is, a 16-MHz 80386. The 80386 runs pipelined with one wait state; according to Intel, that gives a bus utilization of about 86 percent. RAM Refresh takes less than 5 percent of the bus (see table 1), leaving about 10 percent of the bus bandwidth available for DMA transfers when the processor is 100 percent busy. Ten percent of a second is 100 ms for DMA each second. Since the processor can use the 300-ns arbitration time to access memory, as can RAM Refresh, we will count only DMA transfer time. At 400 ns per transfer, this gives 100 ms/400 ns or 25,000 DMA transfers per second; this would be about 50K bytes per second if we used 16-bit transfers. If you assume an I/O rate of 90K bytes per second as being two-thirds 16-bit transfers and one-third 8-bit transfers, you get 300 ms/400 ns or a total of 75,000 transfers per second. That's 300 ms of each second for DMA transfers, plus 50 ms for RAM Refresh, leaving 650 ms for the processor.

If we have 100 percent processor utilization, the 80386 uses 86 percent of the bandwidth; so if only 65 percent of the bandwidth remains, the 80386 can run at only about 75 percent utilization. Under normal operation, a system microprocessor runs nearer to 75 or 80 percent than 100 percent busy anyway. However, this exercise points out an interesting fact: The Micro Channel doesn't have the bandwidth to support multiple processors unless all the processors use caches to cut the required bus bandwidth.

If you have a 64K-byte direct-mapped buffered store through cache using a 16-byte line, you will have a hit rate of about 96 percent, assuming four reads for every write. The 80386's 86 percent bus utilization in this case decreases to about 33 percent; that is, 0.86 × (4 × 0.04 × 0.80 16-byte reads + 0.20 4-byte writes) = 33 percent, where the 4 equals the four 32-bit reads required to fetch a line, and the 0.04 is the miss rate (1 minus the hit rate of 0.96). I assumed the memory can burst matched-memory cycles. This ignores the effects of device bursting and makes the processor wait till memory is free. Thus, the Micro Channel appears to have enough bandwidth to support up to two processors and their I/O. If future PS/2 systems use a cache for the system processor, multiprocessor systems will be much more viable—and I suspect that they won't have the 24-bit DMA limit that the current Model 80 has. ■

Table 1: *The Model 80's internal timings.*

	Model 80-41, 71	Model 80-111
Processor clock speed	16 MHz	20 MHz
Minimum system board RAM access time	187.5 ns	100 ns
Minimum system board ROM access time	187.5 ns	100 ns
Minimum system board I/O time	500 ns	200 ns
Minimum system board video (8 bits)	2000 ns	700 ns
Basic bus cycle	250 ns	200 ns
Arbitration cycle time (minimum)	375 ns	300 ns
DMA		
Minimum bus burst timing	375+500*n* ns	300+400*n* ns
System board burst timing	375+625*n* ns	300+400*n* ns
Bus master		
Minimum bus burst timing	375+387.5*n* ns	300+200*n* ns
System board burst timing	375+625*n* ns	300+400*n* ns
Refresh cycle	8 MHz, 5%	8 MHz, 5%

Where *n* is the number of doublewords, words, or bytes transferred.