MICROSYSTEMS REVIEWS

CompuPro MPX-1 Multiplexer Channel

An S-100 multiplexer board that relieves the system CPU of having to spend any time handling I/O

by Dennis Thovson

Most microcomputers use the system central processing unit (CPU) to handle all input/output (I/O) to peripheral devices such as printers and CRT terminals. This is usually accomplished in either of two ways: programmed I/O or interrupt I/O. Programmed I/O is the less efficient because the system CPU, not knowing when the peripheral device is ready, is required to wait in a status check program loop, or make repeated calls to the peripheral device until it sets its status to ready. Interrupt I/O is usually more efficient in using CPU time (assuming the plication program or operating system is designed to take advantage of the interrupt capability) because the CPU does not have to continually check the I/O device status; therefore the CPU needs to service an I/O device only when that device requests attention via an interrupt.

For example, consider a multiuser system with a number of active consoles attached. Using program I/O, the CPU will have to poll all consoles periodically (every few milliseconds) to see if any key has been pressed. With interrupt I/O, the CPU can execute the various user application programs until an interrupt is sensed from one of the consoles. Only then will the CPU stop execution of the application program, jump to an interrupt handling program routine, get the keyboard character, store it in a buffer, then return to the application program and resume execution.

No time is wasted in polling the console keyboards looking for a key press, which happens very infrequently in terms of CPU execution time. Is there a still more efficient way to handle I/O without wasting main CPU time? Godbout Electronics has taken a page from the large mainframe computer book and developed a multiplexer board that very nearly relieves the system CPU of having to spend any time handling I/O chores. This board is the MPX-1, for the IEEE-696/S-100 bus.

Overview of the MPX-1

The MPX-1 contains a 6MHz 8085 processor, 16K of RAM, 2 to 8K of EPROM, and an 8259A interrupt controller. The RAM and EPROM are local to the MPX-1 and thus do not occupy any address space in the main system memory on the S-100 bus. The MPX-1 is a complete computer that can run independently of, and in parallel with, the main system CPU. However, it does not itself have any I/O capability; that is, it does not have on-board USARTs, PIAs, or the like with which to communicate with the I/O devices. So, you ask, what good is a computer that can't talk to anything? Pay attention now, I didn't say the MPX-1 couldn't perform any I/O—only that it couldn't perform I/O by itself.

What the MPX-1 does is to steal the bus for a cycle or two when it needs to access I/O or other devices on the system bus. It does this by becoming a temporary master and executing a direct memory access (DMA) cycle on the system bus in accordance with protocol defined in the IEEE-696/S-100 specification. Thus, the MPX-1 has access to the system bus and all the attached resources such as main memory and I/O ports. Only one problem remains: How do the main system CPU and the I/O devices get the attention of the MPX-1? It's really quite simple: they interrupt it.

When the main system CPU needs to get the attention of the MPX-1—to output a character to a console or printer for example—it can place the character in a selected location in system memory and cause an interrupt to the 8085 by executing an OUT instruction to a specified port called the ATTN port. The OUT instruction triggers a hardware interrupt to the 8085 (the MPX-1 uses the restart 7.5 interrupt input unique to the 8085). The 8085, upon acknowledging the interrupt, executes a program which, in this example, initiates a DMA cycle and reads the character to be output from system memory into local memory. When the console or printer is ready, it issues an interrupt, and the MPX-1 initiates another DMA cycle and outputs the character to the I/O device port on the system bus. All of this happens independently of the system CPU except for the bus cycles "stolen" by the MPX-1 when it needs to access the system bus.

The 8259A interrupt controller on the MPX-1 is connected to the eight vectored interrupt lines defined on the S-100 bus. This allows any device that can generate an interrupt, such as a video terminal or printer, to directly get the attention of the 8259A and subsequently the 8085 on the MPX-1 board. A console keyboard may, for example, generate an interrupt on one of the vectored interrupt lines that will cause the MPX-1 to input the character and store it in a buffer until the system CPU reads it out. Note that the main system CPU does not need to know about the interrupt being processed by the MPX-1 board.

In some applications it may be necessary for the MPX-1 to get the immediate attention of the main

Dennis Thovson, 243 McMane Ave., Berkeley Heights, NJ 07933

system CPU. This can be accomplished by connecting one of the eight vectored interrupt lines to the interrupt input of the system CPU and configuring the MPX-1 to generate an interrupt on that line. Also, as described above, the MPX-1 can pass status or other types of information to the system CPU, on a program basis, by writing to system memory.

Selected details

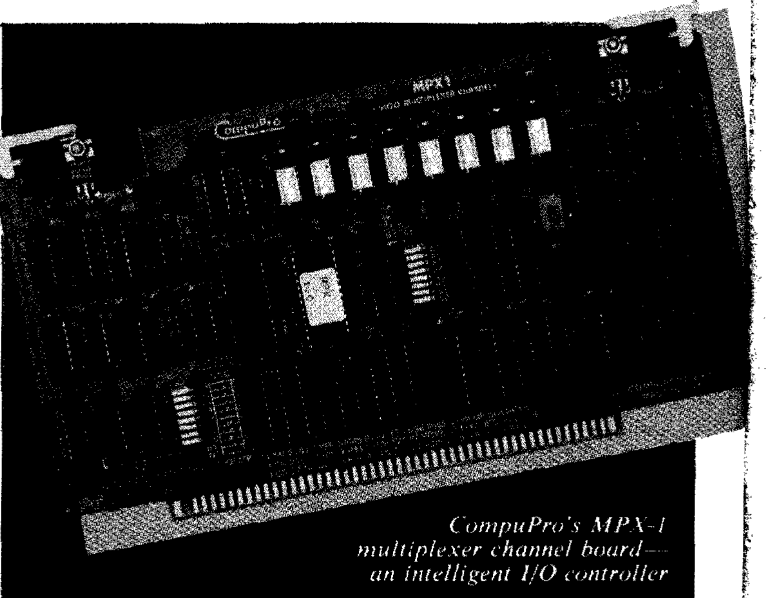
Communication between the MPX-1 and the system CPU takes place through a 100H byte window which the MPX-1 can place anywhere in the main system address space, including extended address memory. Through this window, the MPX-1 can read from, write to, or even execute code resident in the system memory. To understand how this is accomplished by the MPX-1, let us first take a look at its local memory map:

0000H	to	3FFFH	RAM
4000H	to	7FFFH	EPROM
8000H	to	8001H	8259A Registers
8002H			Set Interrupts Latch
8004H			DMA Address Bits 8-15
8005H			DMA Address Bits 16-23
8007H			Interrupt Response Byte
C000H	to	FFFFH	External window

Note that any local address above C000H accesses the external window. But if the window is only 100H, which 100H page within the external window is accessed? The 100H page accessed is that page selected by the DMA address bytes stored at 8004H and 8005H, which represent address bits 8–15 and bits 16–23, respectively. Any memory above C000H addressed by the 8085 will be within the 100H page, starting at at the previously selected DMA address. The careful reader will note that only the low byte of the local address above C000H has any meaning for external memory address selection. The MPX-1 uses the high byte as an indication to trigger a DMA cycle for access to the system memory addressed by the DMA address (bits 8–23) and the low order byte (bits 0–7) of the local address.

Since the MPX-1 does not have any I/O ports itself, an 'IN' or 'OUT' instruction executed by the 8085 will cause the MPX-1 to trigger a DMA cycle to access any ports available on the system bus. Thus the MPX-1 has full access to all the I/O resources on the system bus at a very small time penalty to the system CPU (a stolen bus cycle).

It is beyond the scope of this review to discuss the 8259A interrupt controller in any depth. However, a few remarks may be of assistance to anyone considering integrating the MPX-1 into their system. The 8259A can be connected directly to any or all of the eight vectored interrupt lines defined on the S-100 bus via user-configurable jumpers on a DIP header. The 8259A must be initialized to the desired configuration before it can be used. This is accomplished by writing a sequence of "initialization" and "control" words to the 8259A registers memory-mapped to 8000H and 8001H.


CompuPro's MPX-1 multiplexer channel board— an intelligent I/O controller

An interrupt detected by the 8259A will, after resolving any priority disputes caused by simultaneous interrupts, provide an interrupt to the 8085. It will also provide, during the 8085 interrupt acknowledge cycle, a vector (memory address) to a jump table containing pointers to interrupt handling routines. The 8085 executes the interrupt program routine, tells the 8259A when it is done by writing a byte to the appropriate 8259A register, and then returns to whatever program it was executing prior to the interrupt. All of the above takes place locally on the MPX-1, independent of the system CPU.

As previously mentioned, the MPX-1 also has the capability of providing an interrupt to the system CPU. This is accomplished by jumpering the serial output data (SOD) lead from the 8085 to one of the eight vectored interrupt lines. A "1" written to the SOD port will then cause an interrupt input to the system CPU, assuming it is connected to the proper vectored interrupt line. The MPX-1 will respond to a system bus interrupt acknowledge cycle by writing the byte previously stored at local address 8007H to the system data bus.

MPX-1 programming

As delivered, the MPX-1 comes equipped with a 2716 EPROM programmed with an initializing routine and a number of utilities. Since the 8085 starts execution at address 0 (unitialized RAM), the MPX-1 uses a hardware trick which in effect exchanges the EPROM located at 4000H with RAM at 0 during a RESET or SLAVE CLR. After the first 3 bytes of the EPROM are read by the 8085 (which contain a jump to the starting address of the initializing routine), the EPROM is restored to its normal base address of 4000H.

Godbout Electronics has taken a page from the large mainframe computer book and developed a multiplexer board, the MPX-1, which very nearly relieves the CPU of handling I/O chores.

The MPX-1 utilities furnished include 8259A initializing routines as previously discussed and routines dealing with 8259A status and control, loading MPX-1 RAM with a program from system RAM, executing a program in MPX-1 RAM, and moving a block of memory from one location to another in system RAM. The system CPU can cause the MPX-1 to execute any of the utilities by issuing an OUT instruction to the ATTN port and passing a corresponding command byte and any required parameters through the system "window." The manual says that these utilities are ". . . partly tutorial and partly a useful way to get 'up and running' with the MPX in a minimum amount of time." While I might argue with the "minimum amount of time" statement, the utilities are certainly useful and do help the programmer to understand the functioning of the MPX-1 board. Full source code for the utilities is furnished in the manual.

The ability to execute programs in MPX-1 local RAM loaded from system RAM enables the user to configure the MPX-1 for a specific application at the time of system initialization. It also enables a programmer to try out and debug routines in RAM before they are burned into EPROM. It is in this latter mode that I have spent many hours alternately praising and cursing the MPX-1.

Practical experience with the MPX-1

The basic idea was to program the MXP-1 to handle all I/O chores for my single user 6MHz CompuPro Z80 system. The I/O devices consist of an old TDL video board that requires a lot of software but can emulate most any terminal, a Visual 50 video terminal, a Diablo 1620 printer, which also requires a lot of software to print bidirectionally at 1200 baud, and a 1200 baud 212 type modem. The last three devices are driven by a CompuPro Interfacer 4 I/O board.

The most difficult part of programming the MPX-1 arises because all debugging must be done indirectly—there is no direct access to the 8085 or the MPX-1 local memory. The 8085 runs its programs in splendid isolation from the system (as it is supposed to do). The technique I used was to first develop and run the programs on the main system to uncover and correct any logic errors. This is only partially useful, since one cannot normally simulate all of the hardware features and software interactions of the MPX-1, to say nothing about the timing interactions of a full interrupt system.

The timing interactions are particularly troublesome because they are completely asynchronous to the system—even more so than with a conventional interrupt system. There are two levels of synchronization to contend with: the interrupt programs on the MPX-1 board itself, and the communication between the system CPU and the MPX-1. The I/O routines written for the MPX-1 require a thorough understanding of the 8259A interrupt controller and the particular features of the ATTN port implementation. Since interrupts generated by the ATTN port are directed to the 8085 RST 7.5 input, and those generated by the 8259A are directed to the main 8085 INTR input, these interrupts are independent of one another. (Remember that the 8259A by itself can handle up to eight interrupt requests all occurring independently of one another.)

It is necessary to thoroughly think through the logic of what happens when each interrupt occurs, as well as when and when not to re-enable interrupts during the processing of any given interrupt. I learned this lesson the hard way! At various times during MPX-1 program development for the Diablo 1620 driver, it would print alternate characters, or print every character twice, or go into hyperspace without any clue as to why. This was after the program had been tested and worked perfectly on the main system!

Most of these problems occurred when passing a character to output from the system CPU to the MPX-1 via the ATTN port. I used a software handshake to tell the system CPU when the MPX-1 program had accepted a character and was ready for another one; i.e., the MPX-1 program placed a status byte in system memory. The program timing of this handshake is critical, since the ATTN port interrupt to the 8085, which is disabled whenever an OUT instruction from the system CPU is processed, must be re-enabled.

One must be very careful that the sequence and timing of setting the status byte to ready and rearming the ATTN port interrupt does not permit some unexpected event or time delay to occur in between the two, which would allow the system CPU to send another character or command to the MPX-1 when it should not. This condition arises because the action of setting the status byte and rearming the ATTN interrupt are separate program steps that cannot occur simultaneously. The most foolproof method may be to use the output of the MPX-1 8085 SOD port to interrupt the system CPU as an indication that the MPX-1 is ready. This would, in effect, amount to a hardware handshake and get around the problem of the system CPU knowing when the MPX-1 is ready to accept another character or command. However, with careful attention to program timing detail, I have not found this to be necessary in the programs that have been developed so far.

Summary

The MPX-1 is an elegant concept, flexible in the extreme and a pain in the posterior to program—but then a good challenge is always rewarding when finally met. The first page of the manual states: "The manual is intended to guide the sophisticated systems integrator or OEM through hardware features of the MPX-1. This manual is not intended for novice or inexperienced users." Amen! It further goes on to say

The MPX-1 is an elegant concept, flexible in the extreme and a pain in the posterior to program—but then a good challenge is always rewarding when finally met.

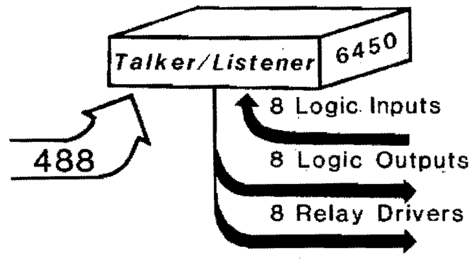
that you should not expect any applications assistance from either CompuPro or your dealer beyond the contents of the manual. The manual itself is typically CompuPro: complete but terse—you have to read it very carefully to extract the nuggets from the ore. The message is clear: Don't buy this board unless you are an experienced assembly language programmer and can understand the hardware concepts involved.

This product will be of most value to the system integrator who can afford the time and effort necessary to adapt the MPX-1 to one particular hardware configuration and then sell many of them. It should be ideal for handling I/O chores in a multiuser system. Another possible application might be a print spooler—the capability exists on the board; all that's missing is the software. And, for very large systems, more than one MPX-1 can be used, since the necessary DMA priority and arbitration hardware are included on the board. Even the dyed-in-the-wool hardware/software hacker may find the MPX-1 board useful to improve the performance of a single-user system. I have concluded, after spending two months developing programs and using the MPX-1, that my system would be incomplete without this board.

The MPX-1 is available in 16K only; for information contact CompuPro Div., Godbout Electronics (Oakland Airport, CA 94614; (415) 562-0636). List price is $649 A&T, $749 SCS (Certified System Component high reliability); however, some dealers are advertising substantial discounts.

IEEE-488 Low Cost Interfacing

FOR: Auto-test / Bread Boarding



- Provides convenient interfacing to the 488 Bus
- Supports all Talk/Listen functions
- Available in two configurations:
  - 6450A Interface Unit . . . $349
  - 6450P PC Board $249

SEITZ *SEITZ TECHNICAL PRODUCTS, INC.*
P.O. BOX 76, NEW LONDON, PA. 19360
PHONE: (215) 255-5111

CIRCLE 260 ON READER SERVICE CARD

NEW!

Aztec C][
• generates FAST 6502 code for APPLE, ATARI, and COMMODORE

Aztec C86
• PC DOS, MSDOS, CP/M-86

CROSS COMPILERS
• PDP 11, 8088/8086, 8080/Z80, 6502

PORTABLE SOFTWARE
APPLE CP/M IBM

Leap into a new dimension with Aztec C!

C COMPILERS — COMMON FEATURES:
• UNIX VER 7 compatibility • standard float, double, and long support • run time library with full I/O and source • fast compilation and execution • full language • cross compilers available

AZTEC C II CP/M (MP/M) $199
• produces relocatable 8080 source code • assembler and linker supplied • M80 interface • SID/ZSID debugger interface • library utility • APPLE requires Z80 and 16K card

AZTEC C][APPLE DOS $199
• produces *fast* relocatable 6502 source code • relocating assembler supplied • APPLE SHELL • VED editor • library and other utilities • requires 16K card

AZTEC C86 IBM PC MSDOS CP/M-86 $249
• produces relocatable 8088/8086 source code • assembler and linker supplied

ORDER BY PHONE OR BY MAIL — Specify products and disk format — Manuals only $30

MANX
software systems
Box 55, Shrewsbury, N.J. 07701 (201) 780-4004

MasterCard VISA

CP/M FORMATS: 8" STD, HEATH, APPLE, OSBORNE, NORTHSTAR OUTSIDE USA — Add $10
In N.J. add 6% sales tax

CIRCLE 245 ON READER SERVICE CARD