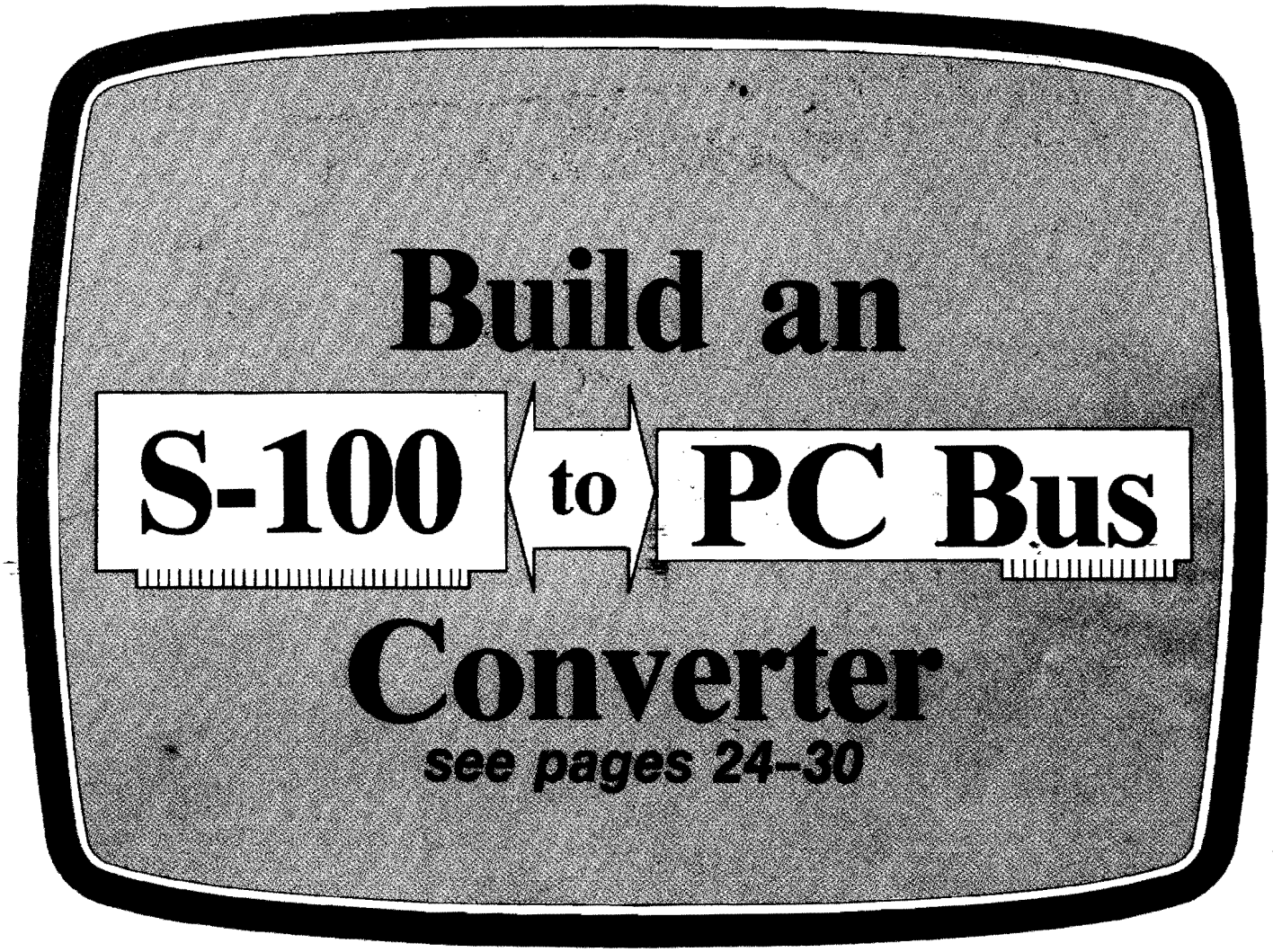
For the Advanced Computer User

Micro/Systems Journal™



Also in this Issue

May/June 1985 Vol. 1/No. 2 $4.00 U.S.A.

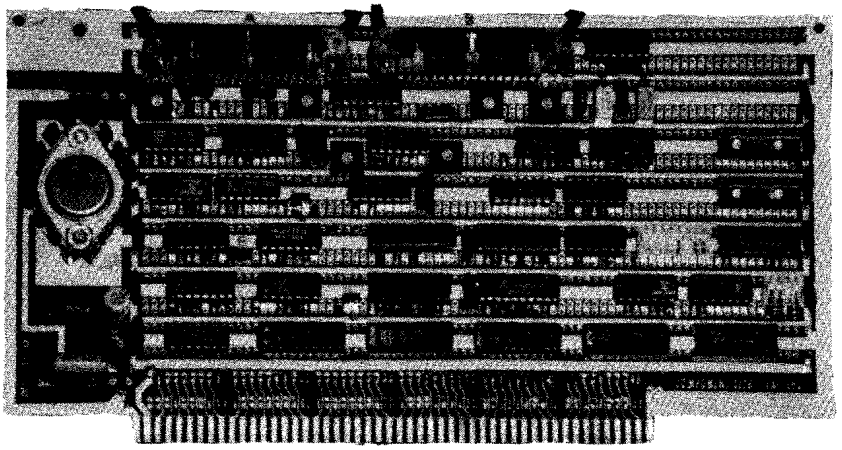
Build an S-100 to PC Bus Converter

by John Monahan

Build an interface so you can use PC video, disk controller and I/O boards with your S-100 system.

Having been an S-100 computer user for the past 8 years I have always taken joy in the fact that I had a better computer system than many "ready made" computers that appeared and later disappeared over the years. As more memory, floppy disk drives, and hard disks began to appear as essential components for a good system I simply shopped around for the appropiate S-100 boards. During that time I had written my own personalized monitor to do all those things one wants a computer to do exactly as I wanted them done. Over the years in fact I have built up a complicated web of hardware units that involve: a second S-100 slave computer to handle multi-printer I/O, speech synthesis, keyboard single key to multikey translations, numerous video boards as well as a master/slave arrangement for Z80 and 8086/8087/8089 CPU control transfer. This rather complicated system is driven by CPM80+, CPM86, and MSDOS disk operating systems. There is still no doubt in my mind that the S-100 bus is the best suited for my needs. Because it is a 16-bit bus it is faster than the IBM-PC.

As time went by a serious limitation of the S-100 bus became apparent. Computer board manufacturers were no longer supporting the bus as much as they did in the past. Perhaps the best example of this was the long delay in bringing out an S-100 video board that was both software and hardware compatable with the PC. The cruel hard fact of life is that the PC market is larger than the S-100 market, so many board manufacturers rushed into that arena. This has led to a good supply of many types of PC bus boards (memory, I/O, disk controllers, video, clock boards to name only a few). A second result of this interest in the PC bus is that the almost oversupply of manufacturers has lead to a supply of good boards at very low prices.



The S-100 to PC Bus Interface board.

Why build a converter?

As I watched this drift away from the S-100 bus, I wondered how many companies would be smart enough to avoid the oversupplied PC market. I waited for them to expand their S-100 bus board set. In particular I was looking for an PC compatible video board. I waited and waited and waited.... Nothing happened. In desperation I considered putting together my own S-100 video board. A careful examination of IBM's technical manuals revealed that this would not be all that simple. All those clock circuits scared me off. Besides I would have to really build two boards. One for monochrome and one for graphics.

Then I had an idea. If I could not get an PC compatible video board for the S-100 bus, why not make a board that would convert S-100 bus signals into a form that could be utilized by PC's boards. In other words make an S-100 to PC-bus converter board.

The crude outline for the project started to take shape in my mind. I would build one S-100 board that would connect the S-100 bus over a buffered ribbon cable to a second 62-

pin bus (on its own motherboard) that is compatible with the IBM-PC bus. In this way my S-100 system would talk to PC cards as if they are on the S-100 bus itself. In fact, from a software point of view the 8086 would not know on which hardware bus the board resides.

As new PC boards arrive, I simply plug them into my extended bus. I now have the best of both worlds. My old S-100 system with most of its memory in fast static RAM (16 bit), its hard disk and memory disk etc. all well oiled and at the same time it has the capability to take on PC boards. While I have many boards in my system that I have constructed and would like to talk about, it is the S-100/PC converter board I would like to describe here.

Building the converter

To begin with, this is not a construction project for a novice. There are bus drivers on both bus interfaces of this board. Incorrect activation of these chips could severly damage your system. Only take on this project if you are sure you understand what you are doing. Perhaps this article may interest a hardware manufacturer to contact me to mass produce the board in which case you will not have to put in the hours of work required to carry off such a project.

Finally, dynamic memory on some PC memory boards is not refreshed by a controller on the memory board itself but rather by the cycling of a 8237 DMA controller on the system board. This continuous reading of RAM once every 72 clock cycles (7% of bus bandwidth) keeps the dynamic RAM refreshed. At present I do not have a DMA controller on my PC motherboard to refresh RAM since I have no need for one. Should you wish to interface (slow) PC compatible memory boards with your S-100 system you will have to address this problem. The main use of this converter board is to interface PC video boards, disk controllers, I/O boards and the like.

The PC bus signals

To understand how this board works we will have to examine all the PC signals and describe how this board simulates them via their S-100 counterparts. The S-100 signals are well described in Sol Libes & Mark Garetz's book "Interfacing to S-100/IEEE 696 Microcomputers". The PC signals are well described in the IBM-PC Technical Reference Manual (In my 1981 edition, pages 2-9 to 2-12). The PC bus consists of 62 pin connectors arranged as 2 rows of 31 pins each. One side is numbered A1 to A31. The other is B1 to B31. These are:

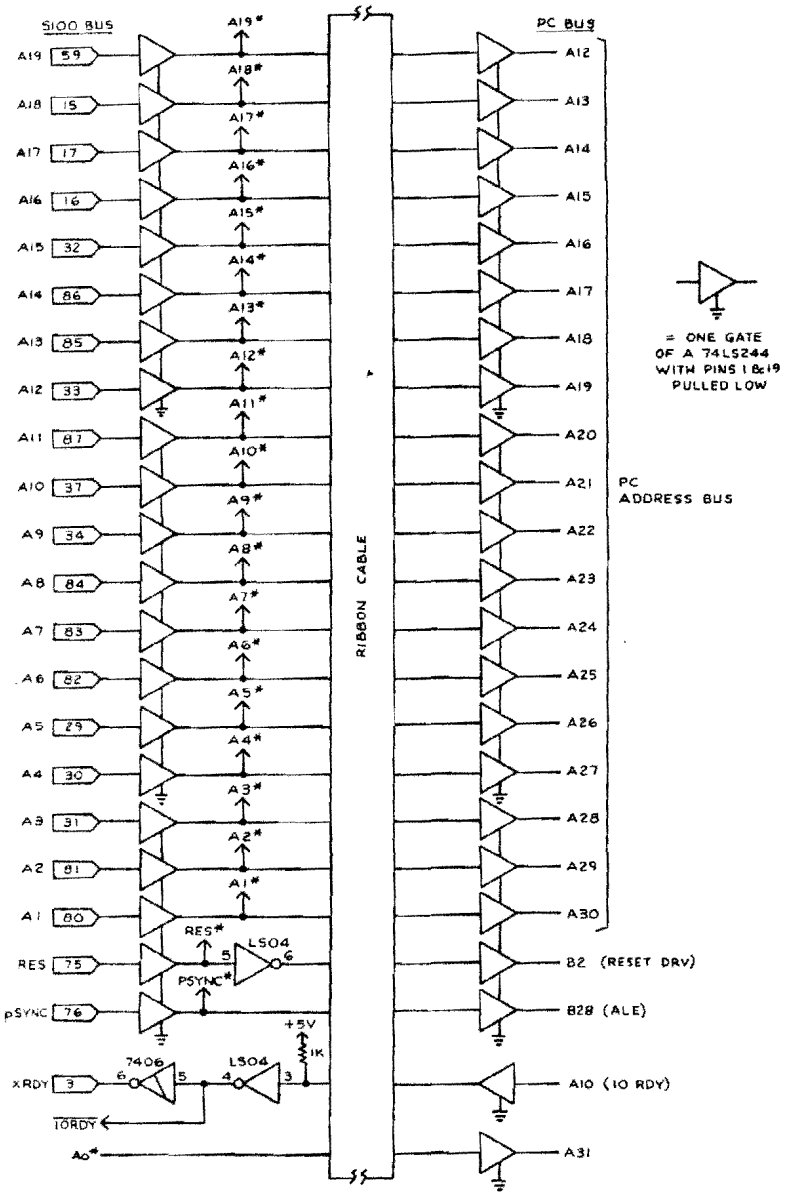


FIGURE 1

A1 -I/O CH CK Provides the CPU with parity error information if an error is detected on any IBM card. It is normally high and is only pulled low when an error is detected. This signal is simply passed directly on to S-100 bus via NMI int line (pin 12) using an open collector output (figure 4).

A2 - A9 Data bits 0 to 7 These lines pass 8 bit data to and from PC cards. Because PC uses an 8088, all data is 8 bits wide. 16 bit transfers are sent or received as two 8 bit data units by the processor. Our first complication is that I use an 8086 on my S-100 bus. For reasons I do not want to go into here, I do not have the capability on the 8086 board of talking to 8 bit memories. It will be clearly necessary on our converter board to convert 16 bit data to 8 bit units to interface to the PC data lines. I also have an old (backup) S-100 8088 board, the converter will also work with this board in 8 bit mode (ie. the S-100 sXTRQ* is not used).

A10 +I/O CH RDY This line (normally high or "ready") is pulled low by a memory or I/O device to lengthen CPU access time to the device. In particular it is used by PC dynamic memory boards to refresh their memory cells. It turns out that this is perhaps the most critical line to be interfaced to the S-100 bus. I will discuss it in more detail later. Suffice to say that the S-100 bus must never access or leave the PC bus when this line is low (i.e. not ready).

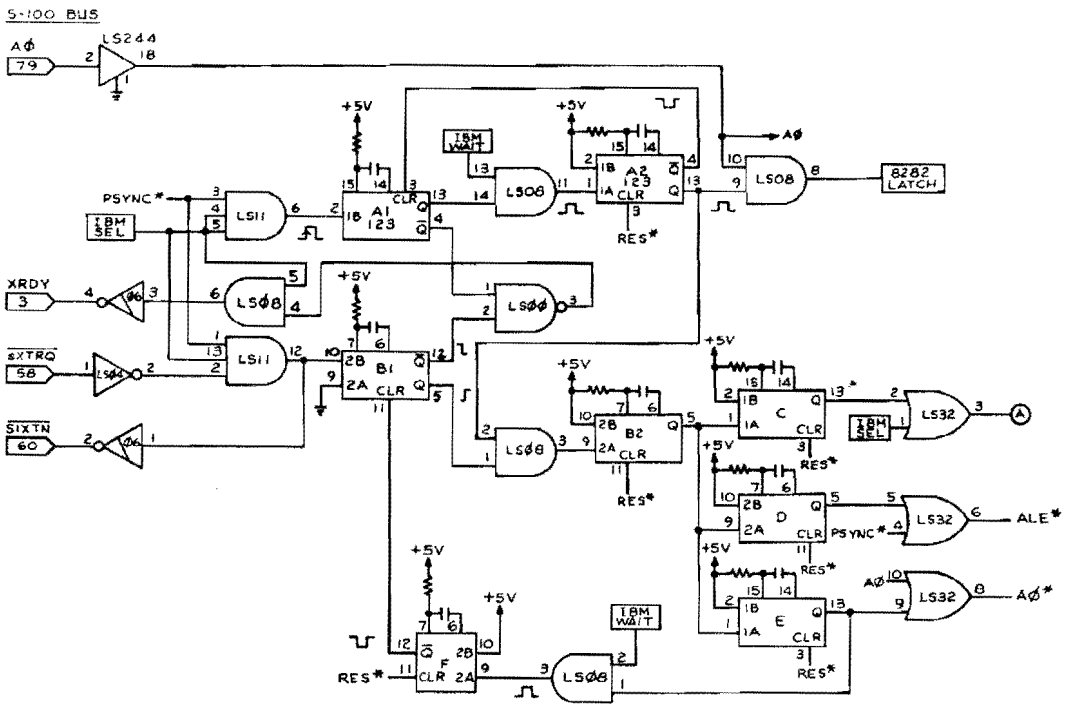


FIGURE 2

A11 +AEN Indicate when a DMA controller has control of bus. Because I do not implement DMA that originates on the PC bus, line is not used and is tied low.

A12-A31 A0 - A19 The address lines on bus to address memory and I/O devices. They are directly comparable with the S-100 address lines and are connected directly to both buses via drivers.

B1 This is one of the bus ground lines and is connected directly to the S-100 ground lines (pins 20,50,53,100).

B2 RESET DRV Used to reset or initialize system logic on the bus. It is active high. The S-100 RESET line (pin 75) is active low, so we must invert the S-100 signal on the converter board before connecting it to the PC bus.

B3 +5V Unlike the S-100 bus the PC bus provides filtered 5 volts directly to its cards. Because a number of cards may be on the bus, the supply current may be considerable. I supply 5 volts to the PC bus from a 5 Amp +5 voltage regulator placed on the PC motherboard. This is enough for a few cards but may need to be increased later.

B4 IRQ2 This is one of 6 Interrupt request lines used to signal the CPU that an I/O device needs some sort of attention. An interrupt is generated by raising IRQ2 line and holding it high. The S-100 bus has interrupt request lines also. However they are active low. These signals are jumperable on converter card and inverted.

B5 -5VDC A -5 volts supply provided to PC cards that require this voltage. A simple 1 amp voltage regulator on PC motherboard is sufficient.

B6 +DRQ2 The DRQ lines on PC bus are used by devices such as a DMA controller to request control of bus away from 8088. The S-100 bus also has DMA request lines. However a DMA request originating on PC side of converter bus could become a can of worms. For the moment I choose to ignore these lines.

B7 -12VDC A -12 volt supply provided to PC cards that require this voltage. A simple 1 amp voltage regulator on the PC motherboard is sufficient.

B8 We get off light for this one

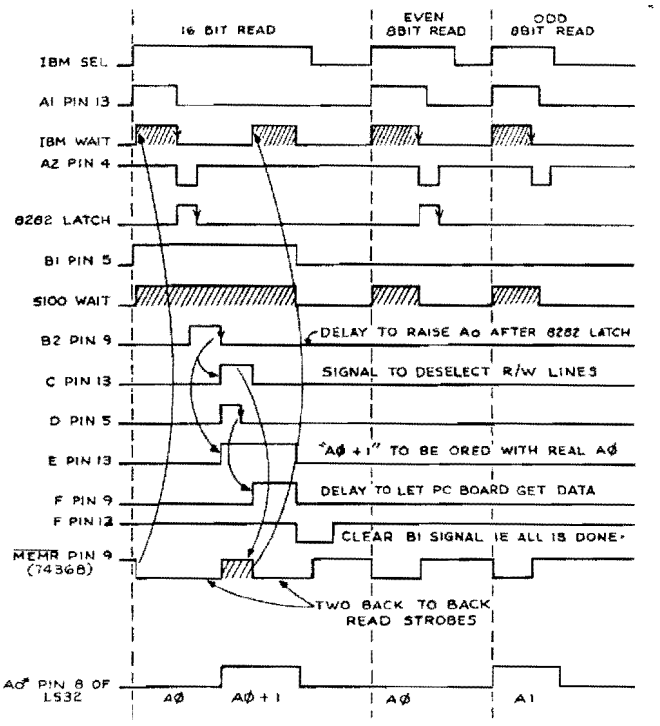


FIGURE 6

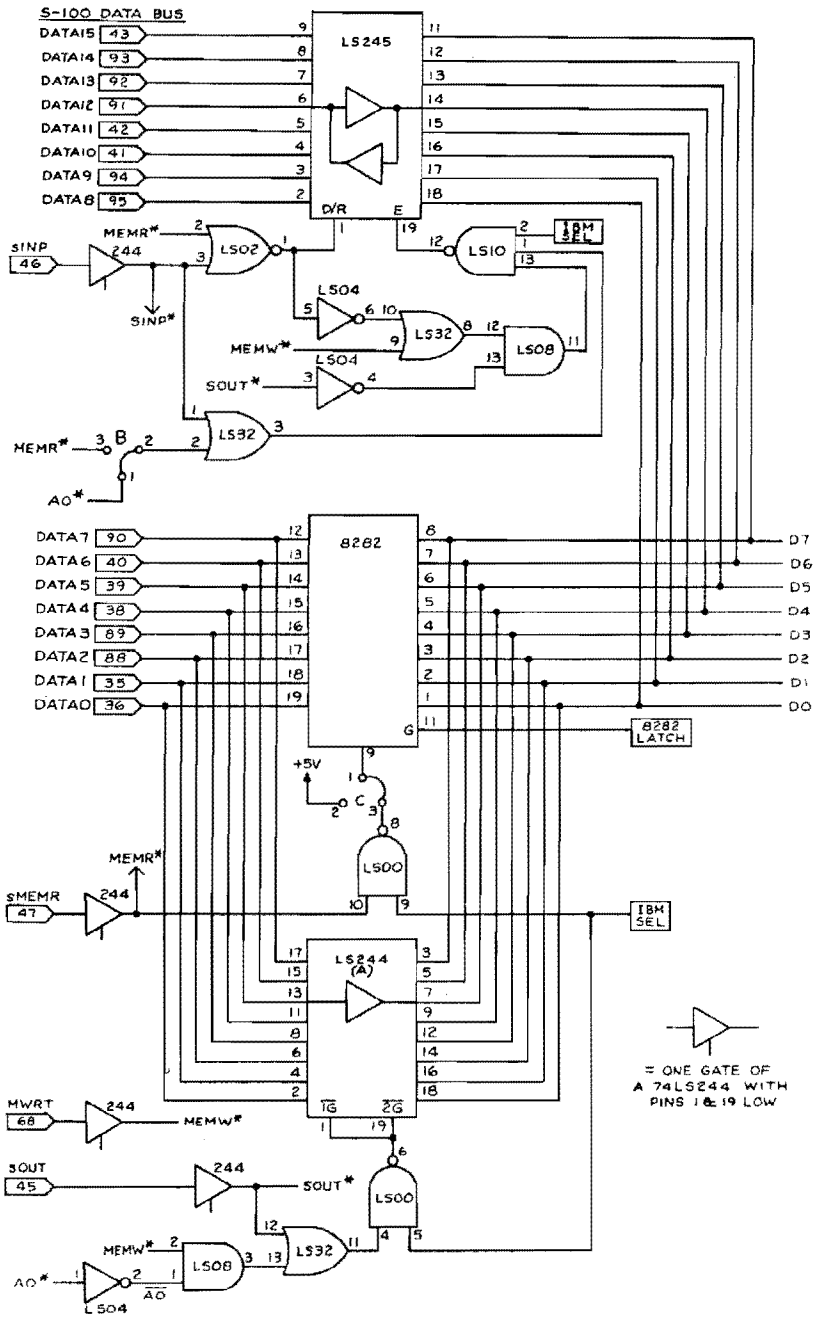


FIGURE 3

since no function has yet been assigned to this pin by IBM.

B9 +12VDC A +12 volt supply provided to PC cards that require this voltage. A simple 1 amp voltage regulator on the PC motherboard is sufficient.

B10 GND Same as B1.

B11 -MEMW Instructs memory devices to store data that is on data bus (active low). Corresponding S-100 signal is MEMW (pin 68). Since it is active high it must be inverted.

B12 -MEMR Instructs memory devices to read data that is on data bus (active low). The corresponding S-100 signal is sMEMR (pin 47). Since it is active high it must be inverted.

B13 -IOW Instructs I/O devices to store data that is on data bus (active low). Corresponding S-100 signal is sOUT (pin 45). Since it is active high it must be inverted.

B14 -IOR Instructs I/O devices to read data that is on data bus (active low). Corresponding S-100 signal is sINP (pin 46). Since it is active high it must be inverted.

B15 -DACK3 One of the 3 bus DMA acknowlege lines. Goes low when 8088 grants a DMA request. The S-100 bus has DMA acknowledge lines also. However I have not utilized them.

B16 +DRQ3 Same as B6.

B17 -DACK1 Same as B15.

B18 -DRQ1 Same as B6.

B19 -DACK0 Same as B15. Used by PC to refresh system dynamic memory.

B20 CLK This is the IBM-PC bus system clock (4.77 MHz , 33% duty cycle). Since the system clock on my S-100 system will normally be running at a very different frequency (8 MHz at present), it is clearly necessary to generate a separate system clock on the PC motherboard. The simplest way to do this is copy the PC directly by using an 8284A clock generator (figure 4), insuring that timing on the bus will be exactly as in a PC.

B21-B25 IRQ7-IRQ3 More interrupt lines; see B4.

B26 -DACK2 Same as B15.

B27 +T/C Provides a pulse when terminal count for any DMA controller is reached; not used in my system.

A28 +AEN The address latch enable signal produced by the PC 8288 Bus Controller chip. The falling edge of this signal indicates that a valid address is on the bus. The S-100 pSYNC (pin76) signal is quite compatible with this signal.

A29 +5VDC Same as line B3.

A30 +OSC A high speed clock with a 70 nsec. period (14.31818 MHz) and a 50% duty cycle. Since this signal is used for many video boards it must be accurate. It is generated exactly as IBM does using an 8284A clock generator directly on the PC motherboard.

A31 +GND Another ground same as B1.

The above describes all PC bus signals. Now let us see how we connect them to the S-100 bus.

Figure 1 shows the connections for the S-100 address lines. Because all signals are being sent over a ribbon cable (in my case 2 feet long), it is necessary to buffer signals at both ends of the cable. I use 74LS244 line drivers with pins 1 and 19 tied to ground to enable the devices. Also, just to be on the safe side I run a ground line between each address line in the cable. The address lines with stars above them (eg. A19*) are simply S-100 signals buffered through the 74LS244. These are used elsewhere on the board. This insures that each S-100 bus signal has only a single gate load. It is very confusing at first glance to see the PC address pins A12 to A30 refer to the pin numbering system not to the value of the address line itself.S-100 pin 75, RESET is inverted through a 74LS04 before becoming RESET DRV on the IBM bus. pSYNC becomes ALE. IO RDY on the IBM bus is inverted and

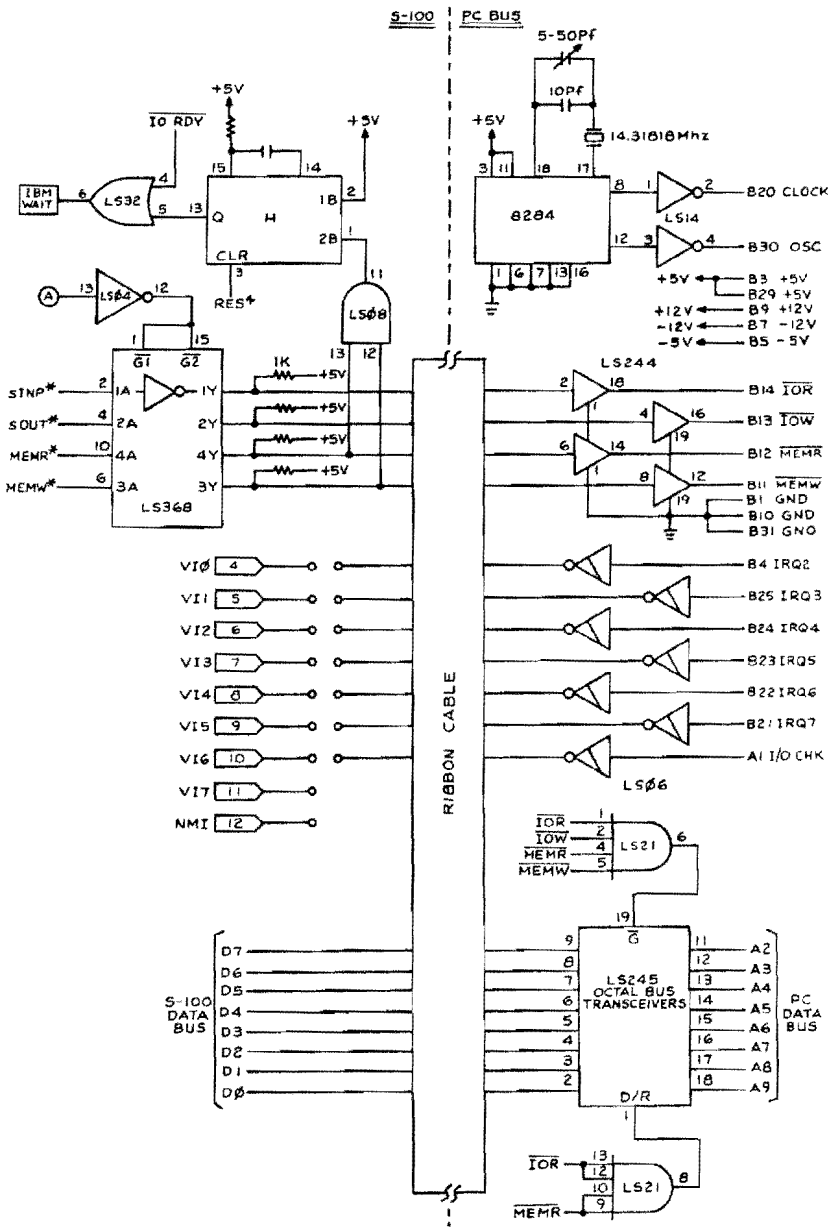


FIGURE 4

then passed on to an open collector inverter 7406 to become XRDY. Address line A0 from the S-100 bus is processed in a special way (see below) to become A0*. This is sent to address line 0 on the PC bus (pin A31).

Figure 2 is the heart of the circuit. Let us consider first 8 bit memory read or write. If the IBM select circuit (described below) determines that the S-100 CPU is addressing a board in one of its PC windows, the Q1 output of one-shot A1 will go high after pSYNC goes high. At the same time the not Q output of A1 will go low holding the S-100 bus in a permanent wait state. The read/write strobe going to the PC board (figure 4) will cause IBM WAIT to go high. When this is over (and it is important to remember that this is highly variable, depending on for example, status of 6845 on IBM video board), the input to A2 will go low. At this point there is valid data on the IBM bus. A2 output Q latches the data for an 8282 (described below) and A2 not-Q output clears single shot A1, which removes the wait state from the S-100 bus.

Things are a bit more complicated for 16 bit memory read or write. We must read/write the lower (A0=0) 8 bits of data, raise A0, and then read/write the odd 8 bits of data. It is necessary to completely remake the ALE and R/W strobes for the IBM bus the second time. The low order R/W cycle for the (A0=0) is exactly the same as described above except now when A2 clears the wait state setup by A1, the S-100 bus is still held in a permanent wait state by the not Q output of B1. This is because now the S-100 signal SIXTN went low triggering B1. After the low order address data is latched into the 8282 (see above) there is a short delay via B2 after which the falling edge of B2 triggers the start of the second cycle. Pins 1 and 15 of the 74LS368 (figure 4), which enable the read/write strobes, temporarily go high. They then return low. The PC sees this as a new R/W strobe. See figure 6 for a detailed timing diagram. We also raise A0 via single shot E and relatch the address lines by a pulse from single shot D. When the R/W strobe comes down, the IBM Boards think a new board access is required. After the (variable length) wait state is generated (via IBM WAIT) the single shot F is triggered. This in turn clears B1, which in turn removes the wait state held on the S-100 bus.

Both I/O read and write are assumed to be carried out as 8 bit transfers by the S-100 CPU. sINP generated inputs will arrive from the IBM bus (if the correct port is mapped, see below) to the "data in" S-100 bus lines via the 74LS245 shown in figure 3. A careful analysis of the diagram will show that pin 19 of this chip (the output enable pin), will always go low when sINP goes high and IBM SEL is high. sOUT generated outputs will travel to the IBM bus via the 74LS244(A) shown in figure 3. Pins 1 and 19 of this chip go low when sOUT and IBM SEL are high.

If you are using a CPU that has an 8088 (i.e. no 16 bit requests) you can simplify figure 3 by changing jumper b1-2 to position b2-3 and c1-3 to c1-2. This effectively removes the 8282 latch forcing all "data out" to go through the 74LS244(A) and all "data in" to arrive via the 74LS245.

Figure 4 shows how the read and write strobes are connected to the IBM bus. Because some static memory boards may not put a wait state on the IBM bus (ie. lower IO RDY) when they are read or written to, and the circuit in figure 2 requires at least a minute pulse of a wait state, a minimal IBM WAIT state signal is generated via single shot H.

The 74LS245 in figure 4 is a bidirectional transceiver for the 8 data lines going to the IBM bus. It is enabled when IOR,IOW,MEMR or MEMW goes low. Pin 1 determines the direction. You may jumper the IBM bus IRQ lines as you wish.

The 8284 clock generator circuit is exactly as utilized by IBM. It is necessary to place this circuit on the IBM motherboard to have sharp signals.

The only remaining section is the mapping system (figure 5). Because

one will normally only want a certain portion of the S-100 16M bytes of memory to map into the IBM bus hardware we must enable the board only when this occurs. The 74LS684 can be jumpered to go low only when the correct memory range is addressed. The jumpering is done so that a high or low address line on one side of the chip matches a high or low on the other. This should be familiar to anybody putting memory boards together. If the correct memory range is being addressed, IBM SEL will go high and be used as described above.

Mapping of I/O ports is essentially the same. Up to 4 blocks of I/O ports (figure 5) can be used. Clearly this could be expanded. For example the PC uses ports 3B0 to 3BF for its monochrome display. The jumpers shown are addressed for this range. Note that unconnected inputs on a 74LS682 are internally pulled high.

Table I shows the capacitor and resistor values for the one-shots used in figures 2 and 4. While putting the board together I used variable resistors to fine tune the system. There is much room for upward variation in the values. Figure 6 shows the scope traces at various points on the board when the small test program shown in figure 7 is run. This test can be used when adjusting the R-C values of the one shots. In the program a continuous series of 16 bit, 8 bit even and 8 bit odd reads to the PC video board are made. The scope is triggered on pin 12 of the 74LS11 in figure 2. The scan rate is 1 micro-sec/cm.

Table 1

Resistor-Capacitor Values
For 74LS123's

IC	Resistor	Capacitor
A1	9Kohms	510pF
A2	14Kohms	50pF
B1	24Kohms	510pF
B2	18.8Kohms	47pF
C	10.7Kohms	56pF
D	6.1Kohms	56pF
E	16.3Kohms	330pF
F	5.8Kohms	30pF
H	1Kohm	180pF

This is not a simple project. However the advantage of having the ability to use PC boards in your S-100 system makes this project worthwhile. There are many ways this board could be improved. I intend, at a later date, to upgrade it to be compatible with the IBM-AT bus system, and to an on-board DMA controller on the motherboard. I will be pleased to collect any

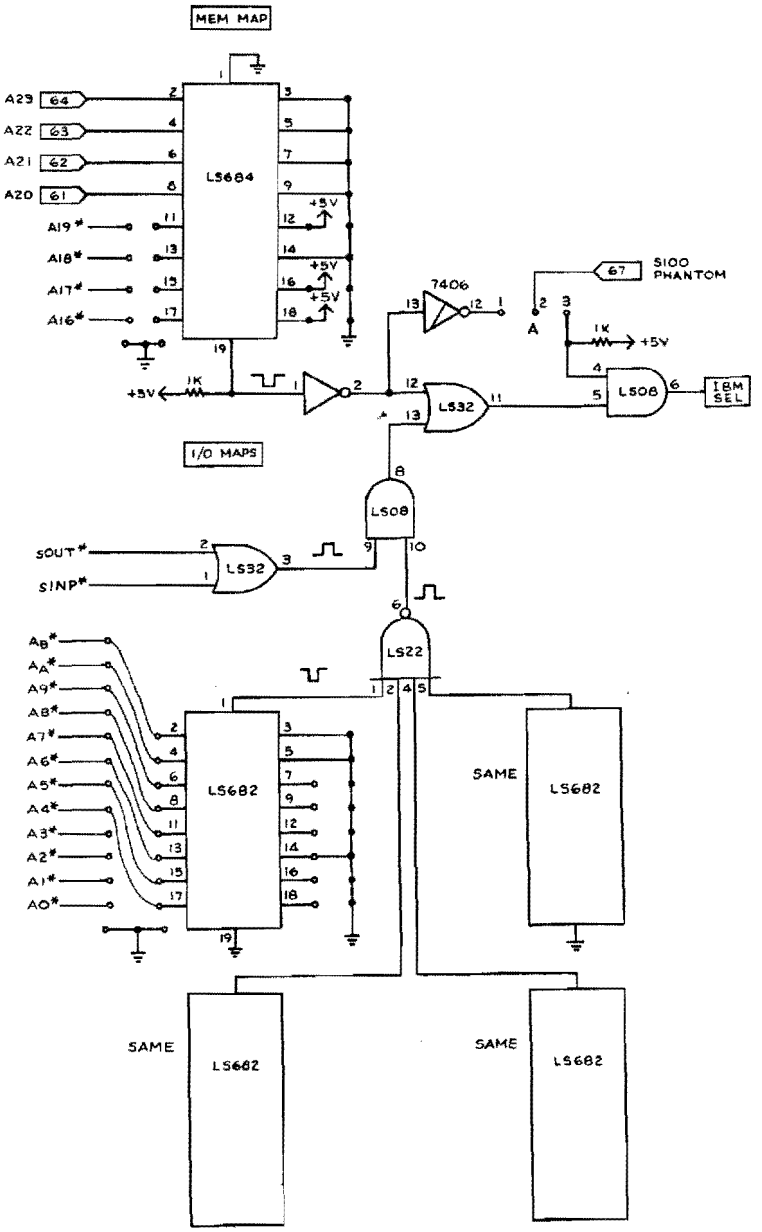


FIGURE 5

circuit improvements you may have.

Finally, it must be pointed out that to utilize the full advantage of this system you must interface it with the correct software. For example, I use IBM's ROM BIOS drivers for their monochrome and color video boards almost word for word out of their technical manual. These form part of a CONIO device driver for my custom MSDOS system.

In future articles I hope to tell you about my S-100/PC keyboard adaptor unit and an S-100 8086 slave CPU (which utilizes an 8087 and 8089) that has the best elements of many commercial CPU boards. These boards when combined with good software (written by a friend of mine that for example utilizes the 8089) yields an S-100 system that today knocks the socks off anything in the non S-100 arena.

```
;Program to generate test pattern
;for IBM-PC converter board. Code
;written for Digital Research's
;ASM86.
        CSEG
        org     0
        mov     ax,0b000h
        mov     ds,ax
        mov     bx,0
;
more:   mov     Word Ptr [BX],ax
        mov     Byte Ptr [BX],al
        inc     bx
        mov     Byte Ptr [BX],al
        dec     bx
        jmps    more

        DSEG
        org     100h
        db      0
        END
```

FIGURE 7

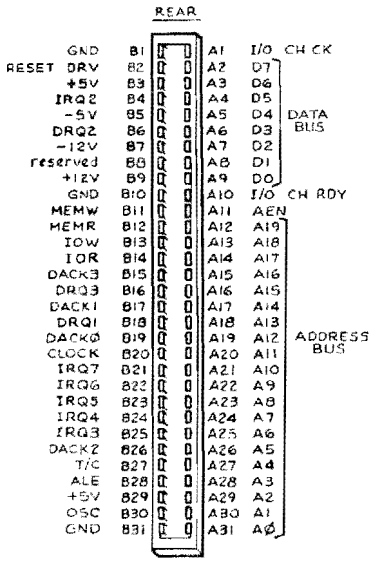


FIGURE 8 — BUS PIN FUNCTIONS

John Monahan is a molecular biologist. He received a Phd in Bio-Chemistry from MacMaster University in Ontario Canada. John has been a computer hobbyist for the past ten years and has built several homebrew systems. He has been a member of the Amateur Computer Group of New Jersey for over 9 years and recently moved to the San Francisco bay area. He can be contacted at: Box 1908, Orinda CA 94563.

Running MS-DOS On S-100 Systems

It should be pointed out that the IBM-PC version of MS-DOS, as well as the versions for most of the PC clones, can be purchased separately of the systems at most computer stores. However, these versions contain system I/O drivers that are not compatible with most S-100 systems. Neither Microsoft nor any of the equipment manufacturers provide the system initializing files necessary to port MS-DOS to another hardware configuration. Seattle Computer Products, did, until a short time ago sell a version of MS-DOS that contained the SYSINT files necessary for the job.

If you own a CompuPro system then you can buy a version of MS-DOS 2.0 already configured for your system from Computer House, 20 Oak Grove Ave, Woodacre CA 94973, tel: (415)897-6387.

If you own a Lomas Data System, then you can buy a preconfigured version of MS-DOS from them (see their ad in this issue).

People using the Tarbell Electronics 816 S-100 board can boot a standard IBM-PC version of MS-DOS. Although they would be better off buying a non-IBM version of MS-DOS which contains disk versions of Basic and BasicA.

Concurrent-DOS from Digital Research is also MS-DOS compatible. The current version (3) is compatible with MS-DOS Version 1. DRI is expected to shortly release Concurrent-DOS 4.1 which is expected to be MS-DOS Version 2 compatible (and also is expected to include the GEM Macintosh-like user interface). Concurrent-DOS V3 is available for several S-100 systems. For example, CompuPro furnishes an implementation for use with their S-100 80286 CPU and new PC-compatible video display cards (a review of which is in the works). And they are promising to furnish Concurrent-DOS V4 as soon as it is available.

DRI, like Microsoft, furnishes Concurrent-DOS strickly as an OEM product. It does sell, via retailers, an IBM-PC (and compatibles) version. However, the software necessary to configure it for a particular hardware configuration is provided only to OEMs. DRI no longer markets any configurable operating systems to non-OEM customers and no longer provides support for its old configurable CP/M operating systems.

We are attempting to find a way in which other S-100 8086 users can implement MS-DOS or Concurrent-DOS on their systems. If any reader can help in this regard please call me.

Sol Libes, editor