THIS MONTH 156 Pages

S100 COMPUTER BOARDS—WHO MAKES WHAT

$1.00 • APRIL 1978

Radio-Electronics®

THE MAGAZINE FOR NEW IDEAS IN ELECTRONICS

pendulums, chimes, displays
UNUSUAL CLOCKS
for your wall and shelf

build for your car
DIGITAL TACHOMETER
with LED display

all about the
S100 BUS
and your microcomputer

for the hobbyist
MODIFY HARDWARE
to suit your needs

how to select a
NEW DMM
your money

Video Tape Recorder Update
New RIAA Equalization Curves
Two Hi-Fi Lab Test Reports
Z-80 Computer Corner
Hobby Corner
State-Of-Solid-State
Service Clinic
XR-2208 Application Note

A GERNSBACK PUBLICATION

All About The S-100 Bus

The 100-line bus that is almost standard equipment in 8080 computer systems. Here we look at the bus and how those 100 lines are used.

WILLIAM BARDEN, JR.

ONE OF THE MOST BENEFICIAL EVENTS IN the microcomputer explosion was the establishment of the S-100 microcomputer bus by MITS, Inc. MITS defined the S-100 bus by the design of the Altair 8800 in 1975. Although they did not intend it as a standard, it soon became one as IMSAI, Polymorphic Systems, Processor Technology, and others brought out microcomputers compatible with the S-100 bus structure. In addition to many microcomputers that use the S-100 bus, there are dozens of manufacturers producing memory boards, I/O boards, speech synthesizers, and other hardware compatible with the S-100 bus structure. This article will explain the basis for the S-100 structure in terms of the microprocessor for which it was designed, the 8080A, discuss the signals of the bus, and describe basic interfacing to the bus.

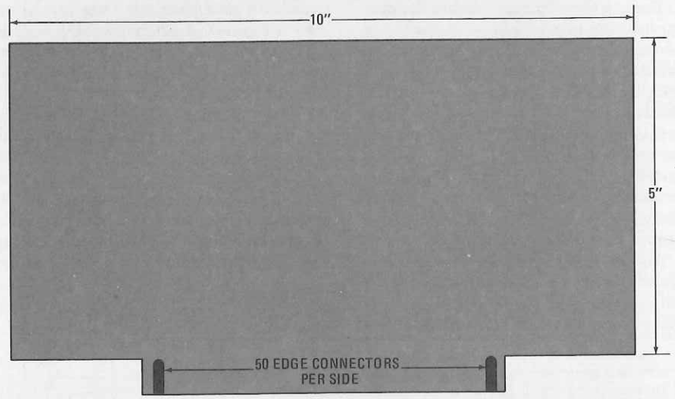

FIG. 1—TYPICAL S-100 BOARD has 100 edge connectors; 50 on each side of the board. These boards are intended to plug into the S-100 mother board.

Physical characteristics

The S-100 bus is a collection of 100 logic and power signals developed from the microprocessor signals. Some signals are logically identical to the signals from the 8080, while others are related, and still others are signals defined by MITS. Physically the bus is represented by a printed-circuit board called a motherboard with 100 parallel foil strips and several 100-pin connectors that are soldered to the foil of the PC board. The typical PC boards that plug into the connectors on the motherboard have 100-pin edge connectors and are the approximate size shown in Fig. 1. A complete S-100 type microcomputer system could consist of the motherboard, power supplies, and a number of S-100 plug-in modules, such as a CPU (*C*entral *P*rocessor *U*nit) module, memory modules, and I/O (*I*nput/*O*utput) modules.

8080 and the S-100

The 8080 microprocessor IC is the heart of most S-100 bus systems, although other microprocessors could be and are being used. Let's first describe the 8080 signals and then see how they relate to the S-100 bus. The pinout of the 8080 is shown in Fig. 2. Most signals are TTL compatible and most are active high.

The 8080 requires three voltages, +5V, −5V, and +12V as indicated. In addition, timing within the 8080 is controlled by a two-phase *non-overlapping* clock represented by ∅1 and ∅2.

Data is transferred bidirectionally between the 8080 and external devices by the data bus, shown as D7 through DO. Data may be instruction data, memory reference data, or input/output data. The 8080 addresses external memory to get the 8 bits of data by means of 16 address lines A15 through A0. Since binary values from 0000000000000000 through 1111111111111111 may be contained on the lines, 65,536 different memory locations can be addressed.

When the 8080 *executes* an instruction, it goes through a predefined *instruction cycle* controlled by the internal logic of 8080. In the course of the cycle, the 8080 first outputs the address of the current instruction on the address bus. It knows the current address from the content of an internal register called a *program counter*. External memory decodes the 16-bit address from the bus and also detects another 8080 signal, DBIN, that indicates that an input (*to* the CPU) operation is to take place. External memory gates the 8 bits of the selected memory location onto the data bus and the CPU strobes in the data some time later in the cycle. During this *fetch* cycle, the

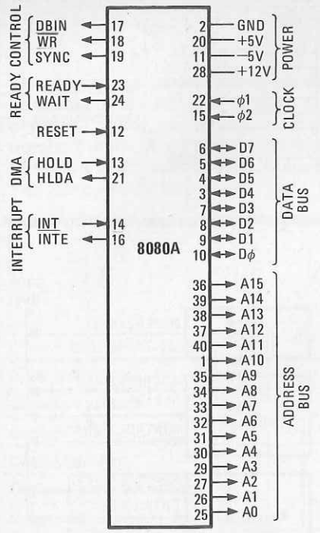

FIG. 2—THE 8080A MICROPROCESSOR IC has the pinout arrangement shown here. Most signals from the IC are TTL compatible.

first byte input represents the complete instruction if the instruction is a one-byte instruction. If the instruction is a two- or three-byte instruction, this first portion of the fetch cycle picks up on the first byte of the instruction. By decoding this first byte (the *operation code*), the CPU knows whether or not to make none, one or two more memory reads to obtain the remainder of the instruction.

When the CPU has the complete instruction, the fetch cycle is completed and the *execution* cycle begins. There are a wide variety of instructions the CPU may execute, but they essentially involve internal functions, reading (again) from memory, writing to memory, reading from an I/O device, or writing to an I/O device.

Reading from memory is similar to the fetch cycle. The address bus holds the memory address of the 8 bits of data to be read and signal DBIN is enabled. If data is to be written to memory, the address bus still holds the memory address of the data to be read, but signal DBIN remains low, and at the proper time signal WR is brought low to strobe the data into external memory. When the CPU executes a read or write (I/O) instruction, the sequence is similar to a memory read or write. The address bus contains the I/O address on lines A7 through A0, as there are only 8 bits available for an I/O address in an I/O instruction.

If data is to be input to the CPU from an external I/O device, signal DBIN is once again high and if an output operation is taking place, DBIN is low and WR is low. How does the I/O device know whether the input or output is from memory or from itself, though? For example, the CPU could read data from memory location 55 and immediately follow that instruction with an input from I/O device 55! To differentiate between I/O addresses and memory addresses, additional signals called status signals are used. There are eight status signals and they are output on the data bus during the beginning of each machine cycle during the time when signal SYNC is high. The status bits and what they represent are shown in Table 1.

TABLE 1

Data Bus Bit	Symbol	Description
D0	INTA	Interrupt acknowledge
D1	WO	Indicates a write to memory or output is about to occur
D2	STACK	Indicates the address bus holds stack address
D3	HLTA	Acknowledge for HALT instruction
D4	OUT	Output device address on address bus; data bus will hold output data when WR active
D5	M1	CPU in fetch cycle for first byte of instruction
D6	INP	Input device address on address bus; data bus will accept input data when DBIN active
D7	MEMR	Data bus will be used for memory read data

The READY signal of the 8080 is an input signal that enables the 8080 to interface with slow memory or I/O devices. If the memory cycle time is not fast enough to allow the memory to respond with data for the CPU, for example, the memory logic may bring down signal READY to a logic 0 level. This causes the CPU to insert an extra clock period in the instruction cycle for as long as READY is low. When in a "not ready" condition, the CPU responds with a WAIT signal that is output to external devices.

The RESET signal is an input that accomplishes what the name describes. It is used before program execution to reset the program counter to zero. Program execution then proceeds from memory location to zero, as previously described for the fetch operation.

The 8080 has the capability, as do most microprocessors, of temporarily suspending instruction execution for *direct-memory-access*, or DMA. DMA permits external memory to be accessed independently of the CPU for high-speed I/O that cannot afford the time required for the CPU to issue a simple byte at a time. When an external I/O device *controller* makes a HOLD request, the CPU responds with an HLDA (Hold Acknowledge), indicating that it has released the address and data buses to the external device. (This is important to avoid the conflict of use of the buses by both the CPU and the external device controller simultaneously.) Normal CPU operation resumes when the external device controller brings the HOLD signal to a logic 0 level.

The remaining two 8080 signals, INT and INTE, are associated with CPU *interrupts*. An interrupt is an external signal that forces the CPU to suspend program execution at the current instruction location and transfer control (or jump) to a predefined address that contains an interrupt program. An example of this might be a CPU dedicated to running a control program that is interrupted when a Teletype key is depressed. The external signal causing the interrupt is INT, which causes an interrupt only if the control program has enabled the interrupt condition by setting an internal interrupt enable flip-flop. The state of the interrupt enable flip-flop is brought out on line INTE.

The above description outlines the 8080 signals necessary to interface external devices to the microprocessor. Now let's see what MITS did with the microprocessor signals to construct a working microcomputer with a control panel, memory and I/O. What they did in the 8800 design defines the S-100 bus.

S-100 bus signals

Power signals: the +8V, +18V and −18V unregulated lines are provided on the S-100 bus. These voltages are regulated to +5 and other required voltages by on-board regulators for each S-100 module.

Data bus: The 8080 data bus is buffered in the S-100 configuration to provide a greater driving capacity. In addition, the data bus is converted from a bidirectional bus to two unidirectional buses. Lines DO7 through DO∅ is the data bus coming out from an S-100 CPU while lines DI7 through DI∅ is the data bus going into an S-100 CPU. The ouput lines are enabled by S-100 bus signals DO DBS that connects to tri-state buffers (see Fig. 3).

Address bus: Lines A15 through AO of the 8080 are buffered in the S-100 bus system; the tri-state enable signal is ADDR DSBL (see Fig. 3).

The six command and control ouputs described for the 8080—SYNC, DBIN, WAIT, WR, HLDA and INTE—are logically unchanged, but buffered in the S-100 bus system. Their tri-state buffer enable signal is C/C DBS. The six signals are renamed PSYNC, PDBIN, PWAIT, PWR, PHLDA and PINTE (see Fig. 4).

The eight status bits of the 8080 are latched into a status latch, as shown in Fig. 5. The status bits listed in Table 1

now become SINTA, $\overline{SWO}$, SSTACK, SHLTA, SOUT, SM1, SINP and $\overline{SMEMR}$. Status is disabled by signal $\overline{STATUS\ DSBL}$. A memory write signal MWRITE is developed from $\overline{SMEMR}$.

Four inputs to the 8080, READY, HOLD, INT and RESET are either buffered or latched from the S-100 bus signals PRDY/XRDY, PHOLD, PINT and PRESET.

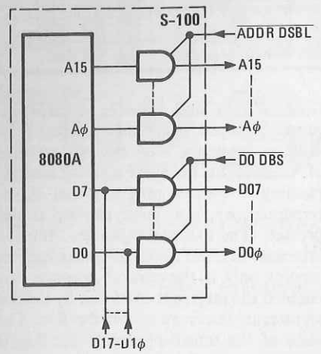

FIG. 3—IN THE S-100 BUS, the data lines from the 8080 are buffered by tri-state buffers as shown in this diagram.

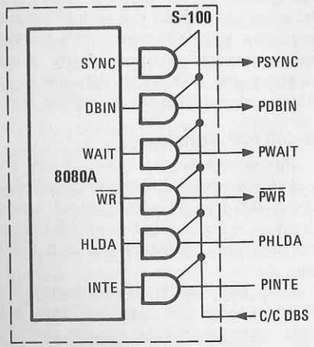

FIG. 4—SIX COMMAND AND CONTROL outputs are also buffered for the S-100 bus. These signals and the buffering can be seen here.

The Ø1 and Ø2 clocks are developed in the S-100 CPU circuitry and routed to the system via S-100 bus outputs Ø1, Ø2 and $\overline{CLOCK}$. The latter is $\overline{Ø2}$ inverted. Signal $\overline{POC}$, Power on Clear, is developed in the power supply logic and indicates when system power is on.

The above signals are S-100 bus signals intimately associated with the 8080, and together with unused pins cover about 80% of the S-100 lines. The remainder of the lines are for the most part associated with *vectored interrupts* and *control panel functions* in the microcomputer.

Signals VI0 through V17 are eight vectored interrupt lines. The 8080 provides eight *vectors* or pointers to interrupt locations with proper external logic. Signals V10 through V17 are S-100 system signals fed to an interrupt board that would implement these functions.

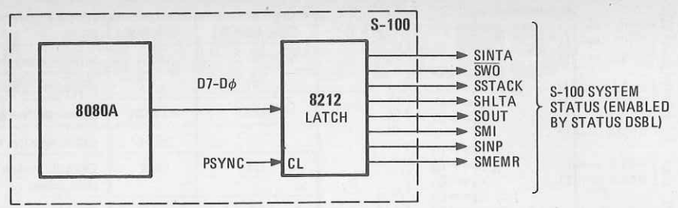

FIG. 5—EIGHT STATUS BITS of the 8080 are latched into a status latch as shown here. When desired, status is disabled by signal STATUS DSBL.

Control panel signals in an S-100 microcomputer system may be present on the S-100 motherboard that *does* have a control panel or they may be strapped to the proper logic level. Some control panel signals may be simply deprived from switch settings. Signals PROT and UNPROT would typically be derived from a switch on a front panel that controls alteration of data in memory. Signals $\overline{PRESET}$ or $\overline{EXT\ CLR}$ may be derived from a momentary switch for system reset and clear of external devices. Signals RUN and SS indicate that the system is running or that a single-step switch is being used to step through a program, respectively. Signal $\overline{SSWI}$ indicates that a data transfer from the control panel sense switches is to take place, for example, altering memory contents.

Although most manufacturers that make S-100 motherboards have made their boards consistent with the above signals, some incompatabilities do exist, especially in cases where undefined pins have been used to carry required new signals. Boards of this kind are not completely compatible with the S-100 bus and may not be used without some modification.

An S-100 system

Now that we've seen the 8080 signals and their relation to S-100 bus signals, let's look at a typical S-100 bus system. We'll assume that the system uses an 8080 CPU card that contains only the 8080 microprocessor and related logic. This will probably consist of buffering and an 8212 status latch. Many of the S-100 bus signals are generated on this board, such as A15 through A0, the status signals and clock signals. Alternatively, the CPU board might contain a Z-80 or even a 6800 microprocessor. However, if a different microprocessor is used it must generate compatible S-100 signals. In many cases, additional logic will have to be added to the CPU board to create S-100 signals. See Fig. 6 for a look at the kinds of signals that may have to be added.

The memory boards in the system use the 16 address lines as inputs to select the memory location being accessed. If the board is an 8K (or 8192) byte board, 13 lines, A12 through A0, are used to select the specific memory location of the 8K while A15 through A13 are decoded to

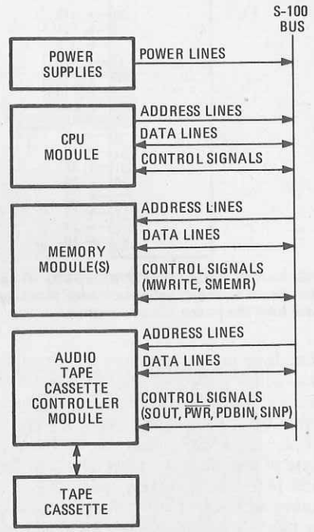

FIG. 6—IF AN 8080 IS NOT USED as the CPU, additional logic may have to be added to create S-100 signals.

select which of the 8K boards is being accessed. Signal MWRITE is used to select the read or write function for the memory IC's on the board, while SMEMR gates the data out to lines DI7 through DIØ on a memory read.

A typical I/O board in the system is an audio tape controller. The address of the controller is decoded from address lines A7 through A2. Address lines A1 and A0 are used to decode functions of the controller such as read and write. Data is output from the controller on lines D07 through DOØ when the controller address is selected on lines A7 through A2, when a write function is specified by A1 and A0 and when SOUT and $\overline{PWR}$ are logic 1 levels. Signal SOUT indicates "output address available" and $\overline{PWR}$ indicates "data available." Data is input from the controller from lines DI7 through DI0 by decoding signals PDBIN, SINP and A7 through A0. Signal PDBIN is the 8080 DBIN (input address available).

The above describes some of the rudiments of S-100 bus operation. Although the S-100 bus has lately taken its share of abuse, it does *work*, it's adaptable, it's fairly efficient, and most importantly it is one of the few standards in an otherwise chaotic hobby. **R-E**

ON THE S-100 BUS

This chart is a partial listing of manufacturers of hobby computer products that fit (plug into) the S-100 bus. To conserve space, we have not listed addresses or phone numbers here. **A complete list of addresses and phone numbers is available FREE.** Simply circle **number 120** on the Readers Service Card Inside the rear cover of this issue. If we've left anyone out of this directory we'd like to know about it. Send us data on any missing entries so we can include them in the future. This is the first in a continuing series of directories that will cover all aspects of hobby computers. Look for more directories covering other bus systems later this year.

MANUFACTURER	CPU	FRONT PANEL	MEMORY RAM	MEMORY PROM	INTERFACE SERIAL	INTERFACE PARALLEL	INTERFACE CASSETTE	INTERFACE VIDEO	FLOPPY CONTROLLER	MUSIC SYNTH	VOICE SYNTH	MATH	OTHER
Artec Electronics			●										Prototype; Wirewrap
Canada Systems													Real Time Clock; AC Controller
Central Data	●		●										
CMC Marketing	●		●		●	●			●				A/D Converter; D/A Converter
Computalker Consultants											●		
Cromemco Inc.	●		●	●				●	●				A/D Converter; D/A Converter
Cybercom								●					
Dajen Electronics													2708 Programmer
Databyte													Logic Analyzer
DC Hayes Associates					●								
Digital Group, The						●							AC Controller; DC Controller
Digital Research Group			●	●									
Electronic Control Technology	●		●										Extender; Prototyping
Equinox Div.		●	●										
Extensyn Corp.			●						●				
Godbout			●										Mother, Extender, Control, Terminator Boards
Heuristics										●			
Imsai	●		●		●	●	●	●	●				
International Data Systems													Clock; Data Access; Frequency Counter; Modem; Clock; D/A Converter
I.O.R.					●	●							
Ithica Audio	●		●	●									Prototype; Blank
Micropolis									●				
Micro Systems Development								●	●				
MITS	●	●	●		●	●	●	●					
Mountain Hardware													AC Controller
National Multiplex													System Controller
Peripheral Vision					●	●							
North Star Computers	●	●	●						●			●	
Priority One Electronics													Extender; Universal Plugboard
Processor Technology			●		●	●	●	●					
SD Sales	●		●										
Solid State Music										●			
Tarbell Electronics							●		●				
TDL	●		●				●	●					System Controller
Thinker Toys					●	●	●	●					
Trace Electronics			●										
Vector Graphics			●			●	●	●	●				
Vector													Universal Plugboards
Vista Computer									●				
Xitex Corp.													Video Terminal
Xybek				●									