LES SOLOMON ON COMPUTER HARDWARE

Talking About Standards

IN THE beginning there was Ed Roberts, the PE-8 computer, and a company called MITS in Albuquerque, NM. The PE-8 was renamed "Altair" and the 100-pin bus structure it used became the Altair bus. An article about it was published in our January 1975 issue. Soon after, entrepeneurs jumped on the bandwagon. The Altair bus was appropriated and soon was called the "Altair/Imsai" bus. (Imsai began marketing a similar type of computer a while later.) As more manufacturers joined the parade, the bus became known as the "Altair/Imsai/Processor Technology/Whatever" bus, a name far too unwieldy to use.

It was at the first Atlantic City Computer Show in 1976 that Roger Melen of Cromemco came up with the name "the Standard-100." Some liked the name, some didn't. Some suggested as an alternative the name "Roberts Bus," after its original designer. In any case, references to the Standard-100 were shortened to S-100 and the name stuck. Thus a new add-on industry was spawned.

Soon, however, the importance of the S-100 became undermined. Not all 100 pins were used in the original bus, and some manufacturers used the extra ones for their "private" signals—ones that would work only with their computers and boards. It quickly got to the point where very few so-called "S-100" boards would work with very few so-called "S-100" computers. For a few years, intermixing S-100 boards and machines became an adventure in frustration.

In 1979, to clean up this mess, several manufacturers got together to try to develop a real S-100 standard spurred by an S-100 standards proposal by Ithaca Audio. After $4^{1}/_{2}$ years of meetings, arguments, and minor hassles, the IEEE finally gave it its blessing and the IEEE-696 (S-100) bus standard was born.

The major changes included 16-bit devices on the bus (the original S-100 was designed to accommodate the 8-bit 8080) and clearing up any ambiguities still attached to the S-100 after it was

IEEE-696 BUS LAYOUT

Pin					
1	+8 Volts (B)		51	+8 Volts (B)	
2	+16 Volts (B)		52	−16 Volts (B)	
3	XRDY (S)	H	53	0 volts	
4	VI0* (S)	L	54	SLAV CLR* (B)	L
5	VI1* (S)	L	55	TMA0* (M)	L
6	VI2* (S)	L	56	TMA1* (M)	L
7	VI3* (S)	L	57	TMA2* (M)	L
8	VI4* (S)	L	58	sXTRQ* (M)	L
9	VI5* (S)	L	59	A19	H
10	VI6* (S)	L	60	SIXTN* (S)	L
11	VI7* (S)	L	61	A20 (M)	H
12	NMI* (S)	L	62	A21 (M)	H
13	PWRFAIL* (B)	L	63	A22 (M)	H
14	TMA3* (M)	L	64	A23 (M)	H
15	A18 (M)	H	65	NDEF	
16	A16 (M)	H	66	NDEF	
17	A17 (M)	H	67	PHANTOM* (M/S)	L
18	SDSB* (M)	L	68	MWRT (B)	H
19	CDSB* (M)	L	69	RFU	
20	0 volts		70	0 volts	
21	NDEF		71	RFU	
22	ADSB* (M)	L	72	RDY (S)	H
23	DODSB* (M)	L	73	INT* (S)	L
24	ϕ (B)	H	74	HOLD* (M)	L
25	pSTVAL* (M)	L	75	RESET* (B)	L
26	pHLDA (M)	H	76	pSYNC (M)	H
27	RFU		77	pWR* (M)	L
28	RFU		78	pDBIN (M)	H
29	A5 (M)	H	79	A0 (M)	H
30	A4 (M)	H	80	A1 (M)	H
31	A3 (M)	H	81	A2 (M)	H
32	A15 (M)	H	82	A6 (M)	H
33	A12 (M)	H	83	A7 (M)	H
34	A9 (M)	H	84	A8 (M)	H
35	DO1 (M)/ED1 (M/S)	H	85	A13 (M)	H
36	DO0 (M)/ED0 (M/S)	H	86	A14 (M)	H
37	A10 (M)	H	87	A11 (M)	H
38	DO4 (M)/ED4 (M/S)	H	88	DO2 (M)/ED2 (M/S)	H
39	DO5 (M)/ED5 (M/S)	H	89	DO3 (M)/ED3 (M/S)	H
40	DO6 (M)/ED6 (M/S)	H	90	DO7 (M)/ED7 (M/S)	H
41	DI2 (S)/OD2 (M/S)	H	91	DI4 (S)/OD4 (M/S)	H
42	DI3 (S)/OD3 (M/S)	H	92	DI5 (S)/OD5 (M/S)	H
43	DI7 (S)/OD7 (M/S)	H	93	DI6 (S)/OD6 (M/S)	H
44	sM1 (M)	H	94	DI1 (S)/OD1 (M/S)	H
45	sOUT (M)	H	95	DIO (S)/OD1 (M/S)	H
46	sINP (M)	H	96	sINTA (M)	H
47	sMEMR (M)	H	97	sWO* (M)	L
48	sHLTA (M)	H	98	ERROR* (S)	L
49	CLOCK (B)		99	POC* (B)	L
50	0 volts		100	0 volts	

used as a sort of standard for the preceding years.

Although the original S-100 "standard" gave some thought to 16-bit CPUs (and, in fact, called the data lines DATA 0 -DATA 16) a problem was created because some 16-bit microprocessor manufacturers wanted to transfer the high byte on the lower half of the data bus, while others wanted to do just the opposite.

The IEEE-696 solved this by considering the data bits as two bytes—one odd and the other even. These are now called OD7-0 (OD for Odd Data) and ED7-0 (Even Data), thus replacing the older designations. The "rule" is: byte data written or read with A0 = 1 appears on the OD7-0 lines during a 16-bit transfer, while byte data written or read with A0 = 0 appears on the ED7-0 lines during a 16-bit transfer. All you have to remember is that any address where A0 = 1 is odd, and any address where A0 = 0 is even.

Generally, 16-bit processors do not produce address line A0. This is because these processors have a basic data path of two bytes. If 0 is the lowest address, then the next word address must be 2, the next 4, etc. Since only odd addresses could use A0 explicitly, it can be eliminated. The fact the address line is missing does little harm since all processors capable of byte addressing provide some means of generating a "fake" address, A0. For example, the MC68000 uses its UDS and LDS lines for this operation. Address A0 is user definable as even or odd for maximum flexibility to allow older "S-100" I/O and memory boards to work.

The PHANTOM line (pin 67) is used to remove a slave from the system address space, and is considered part of the address bus. This signal allows "shadow" ROM to exist and overlay a portion of memory. When the PHANTOM operates (strikes?) the shadow ROM "disappears" from the system. This allows a bootstrap ROM on power up to disappear when the system is running.

The IEEE-696 data bus includes the original S-100 data in/out unidirectional 8-bit data buses, but allows them to be combined into a single bidirectional 16-bit bus.

One other change was the re-naming of DMA (Direct Memory Access) to TMA (Temporary Master Access). When a temporary master is accessing the bus, it may execute either a memory or an I/O cycle. The term DMA implies a memory access only, while the term TMA does not imply any particular cycle, thus more realistically describing this signal.

Two new terms have been added to the S-100 bus. These are NDEF for "Not Defined" (the manufacturer must specify any use in detail), which appear at pins 21, 65, and 66; and RFU for "Reserved for Future Use" that appears at pins 27,28,69, and 71.

The IEEE-696 bus is organized into eight sets of signal lines, one set of power lines, and is shown in the accompanying Table.

The signal lines are formed from 16 lines that make up the data bus, 16, or 24 lines forming the address bus, eight lines in the status bus, five in the control output bus, six for the control input bus, eight for TMA, eight for vectored interrupt, 16 utility signals, and nine for the power bus.

The mnemonics for status lines are preceded by a lower-case "s," while the mnemonics for the control output lines are preceded by a lower-case "p." The suffix asterisk "*" expresses the relationship between the truth state and the electrical state. That is, the variable so indicated is true when its bus line is low. (Note that the vinculum—a bar over a name indicating active low—is not used.) The M or S refer to Master or Slave operation.

Other Standards. There are several other places where standards are needed. In a classical example of serendipity, the lack of standards in the printer industry produced the parallel printer port we call "Centronics." It looks like we lucked into a good thing here. Since it is so widely accepted, why is it not a standard?

Now let us take a look at the pin designations on a typical DB-25 connector. Why do some companies put "private" signals on this connector? Should we not conceive a DB-25 connector standard before the European community takes over with their DIN plugs? Take a close look at many new systems for the appearance of this round connector.

Let us look at floppy disks. Once you leave the security of the IBM format for 8" drives, you are in a wilderness. In the 8" disk business alone, there are five different sizes and 21 different interfaces. Some diskettes differ not only in their overall diameter, but also in the size of their center holes. The 5¼" diskette is burdened with hard and soft sectoring, with the latter offering the choice of 10 or 16 holes (even this is not true since there is an extra index hole). In fact, just buying a package of 5¼" diskettes can be exciting, especially if you have to make a long trip back to the dealer. Also, I don't think many of us really know if there is a standard for single/double/quad-density diskettes without even worrying about single- or double-sided. I am afraid to ask.

Another place where a problem has loomed is in terminals. Why is it when we change terminals from, say, a DEC or Health/Zenith to, for example, a Hazeltine-1500 (Hazeltine has announced it is going out of the terminal business by the way), the software won't work properly? Is this because the bulk of software is written not only for a particular operating system and computer, but also for a particular terminal? There is a standard for terminals (ANSI 3.64), so why do we have a battle between those using this standard and those that abide with ASCII?

Software has "standards" problems, too. For instance, how many BASICs do you know of that can actually communicate with another BASIC? How many books have been written explaining how each version of BASIC works? And why do we have to have BASIC-to-BASIC translation manuals?

Even our schematics need clarification. Back in 1973 (remember that microcomputers didn't come along until 1975), there was ANSI Standard Y32.14 that spelled out how logic symbols should be used. But, outside of HP and TI, who else uses or has even heard of this standard? If it exists, and it is a standard, then why don't we use it?

However, there are some gleams of light down the tunnel. Seagate Technology (it pioneered the 5¼" hard disk) very quickly established a de-facto standard for this storage approach. The Network Users Associaton is pushing for standards to make sure that equipment from different manufacturers interface properly over a communications link. The IEEE is currently working toward creation of guidelines for software standards, while half-inch tape systems are the bailiwick of the California Tape Standards Association.

I've posed a number of questions here. Calling several industry people looking for some answers or even good suggestions, I gleaned very little in the way of answers. Most responses were similar: "We've always done it that way with no complaints." "Our customers are happy so why should we change?" "Ask our competition why they do it their way." And so it went.

The importance of standards cannot be underestimated. They make possible easy implementation, which results in faster growth and lower costs. For its own good, therefore, the microcomputer industry should develop as many as possible with dispatch. ◇