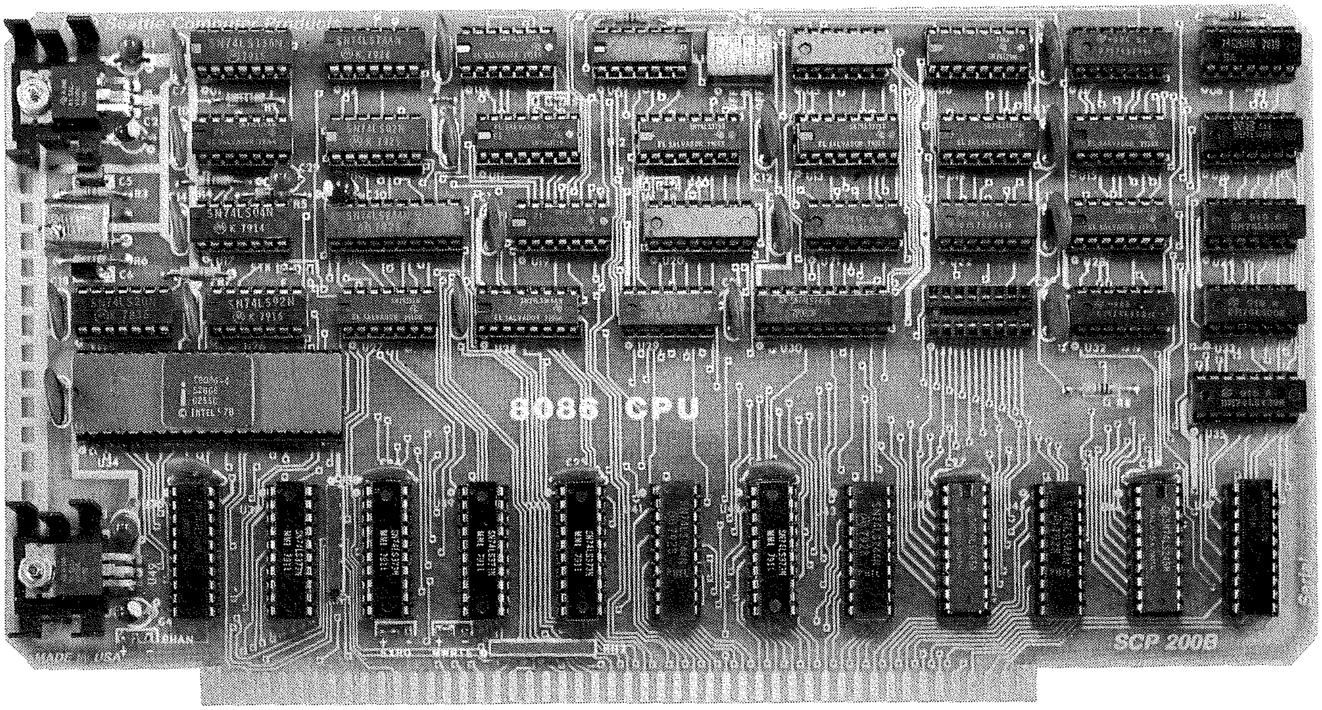
Instruction Manual

Model SCP 200B



8087 adapter board
data at end of manual

8086 CPU

16-Bit Processor for the S-100 Bus

Rev. B

10-1-80

Seattle Computer Products, Inc.
1114 Industry Drive, Seattle, WA. 98188
(206) 575-1830

TABLE OF CONTENTS

Features

> Advanced 16-bit architecture of the 8086 includes a versatile range of addressing modes, hardware multiply and divide, and high-speed string operations.

> Fully meets IEEE-696 standard for 16-bit operation on the S-100 bus, allowing maximum throughput.

> Completely compatible with 8-bit memory and I/O devices with reduced performance.

> Directly addresses up to 1 megabyte of memory, eliminating the need for "bank select".

> Has a minimum instruction execution time of only 250 nanoseconds, yet needs no "wait states" with static memory of 250 nsec. or faster.

> Entire design conforms to IEEE-696 standard for the S-100 bus.

> Configurable to meet a wide range of system requirements through its numerous switches and jumpers.

Configuration Options

SWITCH SELECTED OPTIONS

S-1, located near the top of the board between U4 and U5 has three active segments. The segments of the switch and their function are:

Segment 1: Not used on this board.

Segment 2: Sixteen acknowledge -- the closed position allows the CPU to look at the sXTN* line to determine if the peripheral being accessed can do a sixteen bit transfer. The open position forces the CPU to do all eight bit transfers. The switch is provided because some of the early eight-bit systems used this line (60) for another purpose. The switch allows the user another capability as well. It allows him to compare program execution time differences while using eight-bit or sixteen-bit memories.

Segment 3: This switch selects between 4MHz and 8MHz operation with 8MHz CPU cards. The closed position selects 4MHz, open selects 8MHz. On 4MHz CPU cards this switch segment is defeated by a jumper and has no effect; the CPU always runs at 4MHz.

Segment 4: Closing this switch adds one wait state to each bus cycle. If a peripheral has it's own wait state generator, closing segment 4 adds one wait state in addition to the one(s) requested by the peripheral. With the 4MHz clock, memory chip speed requirements are in excess of 450nS. At 8MHz, 250nS or faster chips must be used. Thus, with most memory, and a 4MHz clock, no wait states are required and the switch segment should remain in the "open" or off position.

JUMPERS

There are six jumpers on the board to allow configuring your CPU for special situations. The chart below shows their "normal" positions.

NAME	NORMAL POSITION	REMARKS
STV (Status Valid)	"D"	Used with DMA
STK	"E"	Used with DMA
I-O	"8"	Puts lower 8-bit I-O address on upper 8-bits
PHANTOM*	"+"	Connects PHANTOM* line
SXRQ (sXTRQ*)	"+"	Connects Sixteen Request line
MWRITE	"+"	Connects MWRITE line

Each of these jumpers will be discussed in more detail to enable you to determine if you should change them to the alternate position.

STV (Status Valid) -- this jumper controls the function of the pSTVAL* line (S-100 pin 25) during DMA. In the "E" position the CPU always drives the pSTVAL* line. In the "D" position this line is three-stated when CDSB* (control disable, S-100 pin 19) is low allowing the DMA device to drive it.

STK -- this jumper controls the function of the "SPEED" line (S-100 pin 98) during DMA. In the "E" position the CPU always drives this line with a high to indicate 8MHz operation or low to indicate 4MHz operation. In the "D" position, this line is three-stated when SDSB* (status disable, S-100 pin 18) is low allowing the DMA device to drive it.

I-O -- In the normal "8" position, this jumper copies the lower 8 bits of I-O address information onto the next higher 8 address bits. If your I-O devices use a 16-bit address, the jumper should be in the 16-bit position. Most existing I-O devices use addressing in which the lower and middle address bytes are the same. The upper address byte, A16-A23, is never used in I/O addressing.

PHANTOM* -- This CPU board generates a PHANTOM* signal whenever it addresses memory locations located above the lowest 64K. This signal may be used to disable older, 8-bit memory cards which do not have extended addressing. Through the use of this PHANTOM* signal, a system may be composed of up to 64K of the older, non-extended addressable memory which would be located in the lowest 64K and newer, extended addressable memory, located in the higher addresses. The PHANTOM* signal would prevent the older memory from interfering with the memory located at the higher addresses. If your system uses this signal for some other purpose, the PHANTOM* generated on the CPU board should be disconnected from the bus by putting the PHANTOM* jumper in the "-" position.

sXTRQ* -- The Sixteen Request signal is generated by the CPU whenever it wants to do a 16-bit transfer. In our system, using 16-bit memory, this jumper should be connected in the "+" postion. If you do not have 16-bit memory in your system, the signal would serve no purpose and the jumper could be in either position. If your system uses bus line 58 for some other purpose, the sXTRQ* line should be disconnected between the CPU and the bus by placing the jumper in the "-" position.

MWRITE -- This jumper connects the MWRITE line of the CPU board to the bus. It is normally connected and used by most systems as the primary write line. It should be disconnected if MWRITE is generated somewhere else in the system, such as on a control panel or the motherboard. Disconnect by placing the jumper in the "-" position.

Technical Description

1.0 General

The SCP-200 CPU card is an Intel 8086 CPU chip with the electrical and mechanical hardware to interface it to the S-100 bus in accordance with the proposed IEEE standard. It may be used to upgrade the performance of present S-100 systems by swapping CPU cards or it may form the foundation of a high performance 16-bit computer.

Some prior knowledge of the fundamentals of the S-100 bus is essential to understanding this document. A thorough understanding requires familiarity with the IEEE-696 standard for the S-100 bus.

The asterisk (*) denotes an active-low signal.

2.0 Features and Comparisons to Proposed Standard

2.1 Clock

All timing signals are derived from an on-board 24 MHz oscillator. The clock rate of the 8086 chip itself is either 4 or 8 MHz. A lower-priced version of the board is capable of 4 MHz operation only, while the higher-priced version has switch selection of either a 4 or 8 MHz clock. All 4 MHz-only CPU cards are factory upgradable to 4/8 MHz.

On the bus, CLOCK (pin 49) is always 2 MHz, while pSTVAL* (formerly Φ1, pin 25) and Φ (formerly Φ2, pin 24) are equal to the clock frequency of the CPU (4 or 8 MHz). All three are approximately square waves. pSTVAL* and Φ are equal in phase, but pSTVAL* may be configured to be disabled (tristate) by CDSB* (pin 19) while Φ is never disabled.

The 8086 allows 3 clock cycles for memory access (minus delays and setup time). At 8 MHz, 250 ns memory is required while at 4 MHz, over 500 ns access time is allowed. (See section 4.) If more time is required, the card may be switched to insert a wait state in every bus cycle. In addition, the line formerly named sSTACK (pin 98) is driven with clock speed status--high for 8 MHz, low for 4 MHz. This may be used to control wait state generators of peripherals.

2.2 Data Bus

The 8086 CPU chip is a 16-bit processor which is interfaced to the S-100 bus by the support logic in either of two modes: 8-bit mode or 16-bit mode. The mode is selected dynamically at the beginning of each bus cycle, depending on CPU status and bus response, as follows:

2.2.1 sXTRQ*, SIXTN*

If the processor wishes to read or write 16 bits, sXTRQ* (pin 58) on the bus is driven low. The addressed device may then respond by pulling low SIXTN* (pin 60) if it can perform a 16-bit parallel transfer. Timing for SIXTN* is the same as the READY lines (RDY/XRDY)--it must be valid 50 ns before the rising edge of Φ that ends pSYNC. An exception to this is that if a wait state is requested (externally or on-board), SIXTN* may be delayed until near the end of the last wait state. SIXTN* is latched on the rising edge of Φ, either near the end of pSYNC or near the end of the last wait state, and changes after this time are irrelevant.

If, at the time SIXTN* is latched, it is high (inactive), then sXTRQ* is removed and double 8-bit mode is selected. Otherwise, sXTRQ* is kept low and 16-bit mode is selected.

Should the 8086 CPU card be used in a system which previously defined a conflicting use for bus pins 58 or 60, either or both of these lines may be disconnected from CPU logic. In this case, 16-bit mode is not possible.

2.2.2 8-Bit Mode

2.2.2.1 Simple 8-Bit Mode. This mode is used when the processor wishes to transfer only 1 byte. Logic on the card routes data 1) from the DI bus to the processor's low 8 data lines (read even location); 2) from the DI bus to the processor's high 8 data lines (read odd location); 3) from the processor's low 8 data lines to the DO bus (write even location); 4) from the processor's high 8 data lines to the DO bus (write odd location).

2.2.2.2 Double 8-Bit Mode. This mode is used when the processor wishes to transfer a word on an even address but the memory or I/O device is not capable of 16-bit operation. The processor is put into a wait state for at least 4 clock periods while on-board logic runs two bus cycles back-to-back, first at the even address, then at the odd. Each bus cycle is essentially the same as in simple 8-bit mode above, except the data resulting from the first of two read cycles is latched until the second cycle is complete.

2.2.3 16-Bit Mode

In this mode, 16 bits of data are transferred in parallel in one bus cycle. The DO lines carry data to/from the even addressed location (processor's low 8 data lines), and the DI lines carry data to/from the odd addressed location (processor's high 8 data lines).

2.2.4 Data Driver Disable

Bringing low DODSB* (pin 23) will disable both the data input and data output drivers.

2.2.5 Compatibility

The techniques for mode selection and 16-bit transfer are in strict accordance with the proposed IEEE standard. They allow the CPU to run without modification with ordinary 8-bit memory (for easy upgrade), with new 16-bit memory (for high performance), or even in a mixed environment (e.g., 16-bit memories in time-critical code areas).

2.2.6 Data Line Connector

Connector J1, a 16-pin IC socket, is provided to allow monitoring the data lines of the 8086 CPU chip. Pins 1 to 16 of this socket are connected to AD0 to AD15 of the 8086, respectively. Further, by using DODSB* to disable the data drivers when the board is reading (pDBIN active), arbitrary instructions or data may be fed to 8086 through this connector.

2.3 Address Bus

2.3.1 Memory Address

The proposed IEEE standard dedicates 24 pins on the bus for memory address information. All 24 are driven by the CPU card; however since the 8086 chip itself generates only 20 bits of address, A20-A23 are always driven low.

2.3.1.1 Phantom

Traditionally, S-100 memory cards have decoded only the lowest 16 address lines, limiting their address space to 64K bytes. Ordinarily, use of such cards in a system with more than 64K is not possible because these cards will appear in the same relative position in each 64K block. However, special provision has been made on the 8086 CPU card, selected by a jumper: Whenever a memory location ABOVE the lowest 64K is addressed, PHANTOM* (pin 67) is driven low, which may be used to disable memories with a 16-bit address. Thus these memories appear only in the lowest 64K, when PHANTOM* is not being driven.

The PHANTOM*-driving circutiry on the CPU card continues to function during DMA. Thus DMA controllers which use a 24-bit address need not be concerned with the presence of memories using only a 16-bit address.

2.3.2 Input/Output Address

The proposed standard allows and the 8086 CPU chip supports a 16-bit I/O address. However, traditionally the lower 8 bits and upper 8 bits of the address have been the same, and unfortunately many devices have been designed which decode their I/O address from the upper 8 lines. To maintain compatibility with such devices the support logic can be jumpered to throw away the high 8 bits of the 8086's I/O address and substitute the low 8 bits.

2.3.3 Address Driver Disable

Bringing low ADSB* (pin 22) will disable the address drivers.

2.4 Status Bus

The CPU card drives all status lines in accordance with the proposed standard. They are:

NAME	PIN
sMEMR	47
sM1	44
sINP	46
sOUT	45
sWO*	97
sINTA	96
sHLTA	48
sXTRQ*	58

2.4.1 Status Driver Disable

Bringing low SDSB* (pin 18) will disable the status drivers. Clock speed status, pin 98, may optionally be disabled by this signal if necessary, to be consistent with its traditional definition as the sSTACK status line.

2.5 Control Bus

All control lines are in accordance with the proposed standard. They are:

OUTPUTS	PIN	INPUTS	PIN
pSYNC	76	RDY	72
pSTVAL*	25	XRDY	3
pDBIN	78	INT*	73
pWR*	77	NMI*	12
pHLDA	26	HOLD*	74
		SIXTN*	60

Traditionally, another control line has been pWAIT (pin 27). This line is driven by the CPU card and goes high as pSYNC goes low if the bus is put into a wait state.

2.5.1 Control Driver Disable

Bringing low CDSB* (pin 19) will disable the control drivers. Traditionally, pin 25, now pSTVAL*, has not been disabled by this signal. A jumper is provided to select whether or not it will be disabled.

2.6 Power Supply

Only +8V (pins 1 & 51) and GROUND (pins 50 & 100) are used for power. Two on-card regulators provide +5V at 1.2 A (typical) to all circuitry.

2.7 Other lines

2.7.1 Reset

POC* (pin 99) is a totem-pole output driven low momentarily when power is first applied to the card. During this time RESET* (pin 75), normally an input, is also driven low. The CPU may be reset at any time by driving RESET* low for 5 or more Φ cycles.

After reset, the 8086 will begin program execution at 0FFFF0H. See Intel's 8086 Family User's Manual for details.

2.7.2 MWRITE

The CPU card normally generates MWRITE (pin 68). If this signal is generated elsewhere in the system, this function may be defeated by removing a jumper.

2.8 Front Panel Compatibility

There is currently no hardware front panel for the 8086 on the S-100 bus. With one simple modification, an IMSAI front panel may be used to run, stop, single step, and observe address and data. Note that EXAMINE and DEPOSIT functions will not work. A 16-pin socket directly connected to the 8086's Address/Data lines is provided, with pins 1-16 wired to AD0-AD15, respectively. This may be used with an IMSAI front panel to observe 8 data bits at a time. Only pins 9 through 16 of the IMSAI front panel connector are used. To observe the low (even) data byte, plug pins 16 to 9 into pins 1 to 8 of the socket (the cable should extend downward). For the high (odd) data byte, pins 9 to 16 of the cable plug into pins 16 to 9 of the socket (the cable should extend upward). When the processor does a "double gulp"--two 8-bit reads when 16 bits are needed--both halves of the word will be displayed sequentially on the high data lines.

The modification to the IMSAI front panel to allow single stepping is made on the front of the circuit board, near the edge connector below address switch 6. Cut the trace to connector pin 39 (this trace comes in at 45 degrees). Install a jumper from this trace to the trace connected to pin 44 (the next trace to the right). It will be necessary to scrape away a portion of the solder mask from both traces. This change will not affect operation of the front panel with 8080 CPUs.

3.0 S-100 Connector Layout

PIN	NAME	PIN	NAME
1	+8 Volts	51	+8 Volts
2	+16 Volts	52	-16 Volts
3	XRDY	53	
4		54	
5		55	
6		56	
7		57	
8		58	sXTRQ*
9		59	A19
10		60	SIXTN*
11		61	A20
12	NMI*	62	A21
13		63	A22
14		64	A23
15	A18	65	
16	A16	66	
17	A17	67	PHANTOM*
18	SDSB*	68	MWRITE
19	CDSB*	69	
20		70	
21		71	
22	ADSB*	72	RDY
23	DODSB*	73	INT*
24	Φ	74	HOLD*
25	pSTVAL*	75	RESET*
26	pHLDA	76	pSYNC
27	pWAIT	77	pWR*
28		78	pDBIN
29	A5	79	A0
30	A4	80	A1
31	A3	81	A2
32	A15	82	A6
33	A12	83	A7
34	A9	84	A8
35	DO1	85	A13
36	DO0	86	A14
37	A10	87	A11
38	DO4	88	DO2
39	DO5	89	DO3
40	DO6	90	DO7
41	DI2	91	DI4
42	DI3	92	DI5
43	DI7	93	DI6
44	sM1	94	DI1
45	sOUT	95	DI0
46	sINP	96	sINTA
47	sMEMR	97	sWO*
48	sHLTA	98	SPEED
49	CLOCK	99	POC*
50	GROUND	100	GROUND

4.0 A.C. Characteristics

The symbol in the left column is of the form

TAYBZ

where T means a timing measurement;
A,B are abbreviations of signal names;
Y,Z are abbreviations of signal levels.
It means the time from the first signal reaching the first level until the second signal reaches the second level.

The following abbreviations are used:

Signal Names

A - Address bus
C - Clock (Φ)
D - pDBIN
I - Data-in
M - MWRITE
O - Data-out
R - Ready (RDY/XRDY/SIXTN*)
S - Status bus
V - pSTVAL*
W - pWR*
Y - pSync

Signal Levels

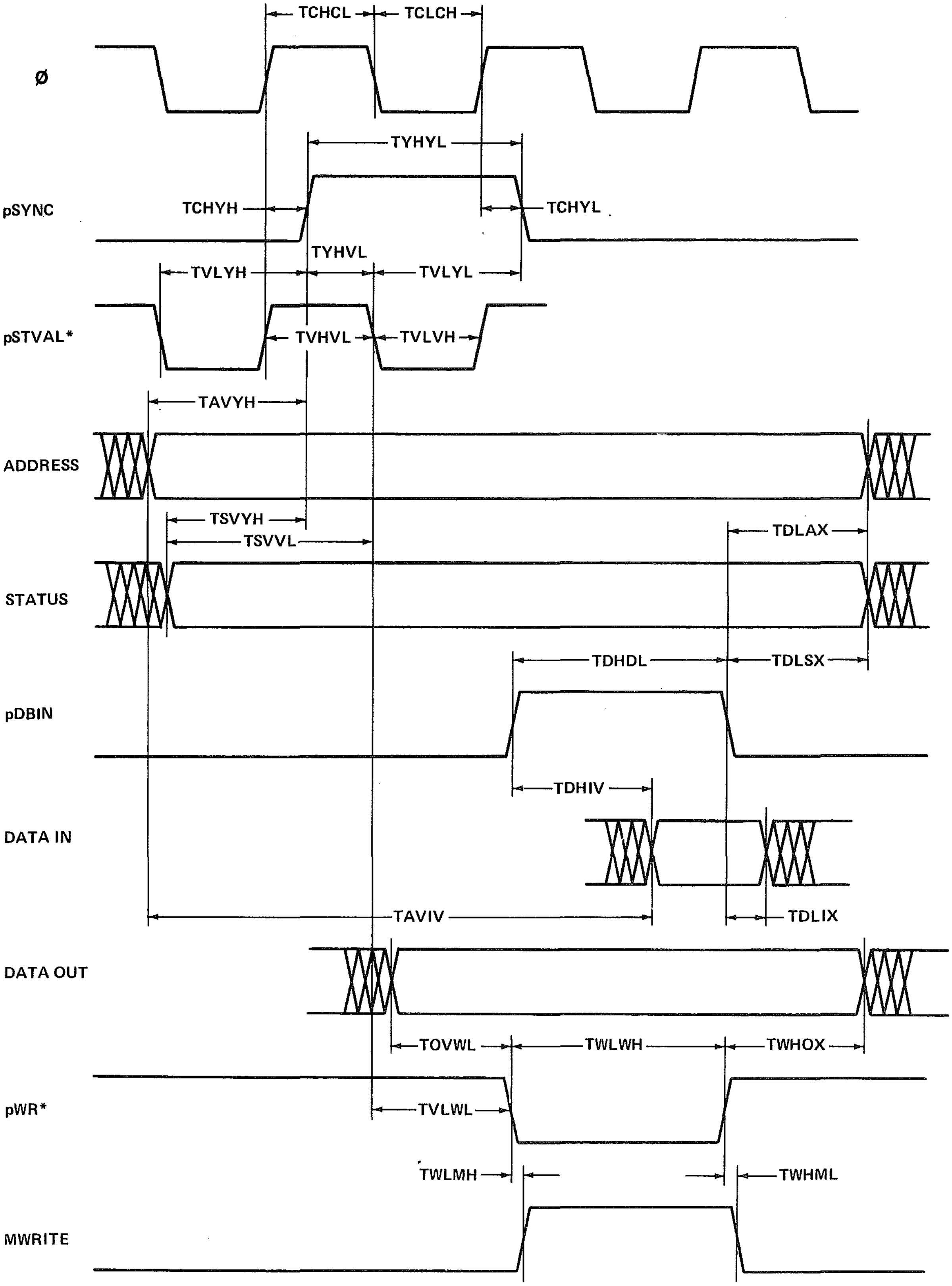
H - High
L - Low
V - Valid
X - Invalid

For example, TRVCH means "Time from Ready becomes Valid until Clock goes High."

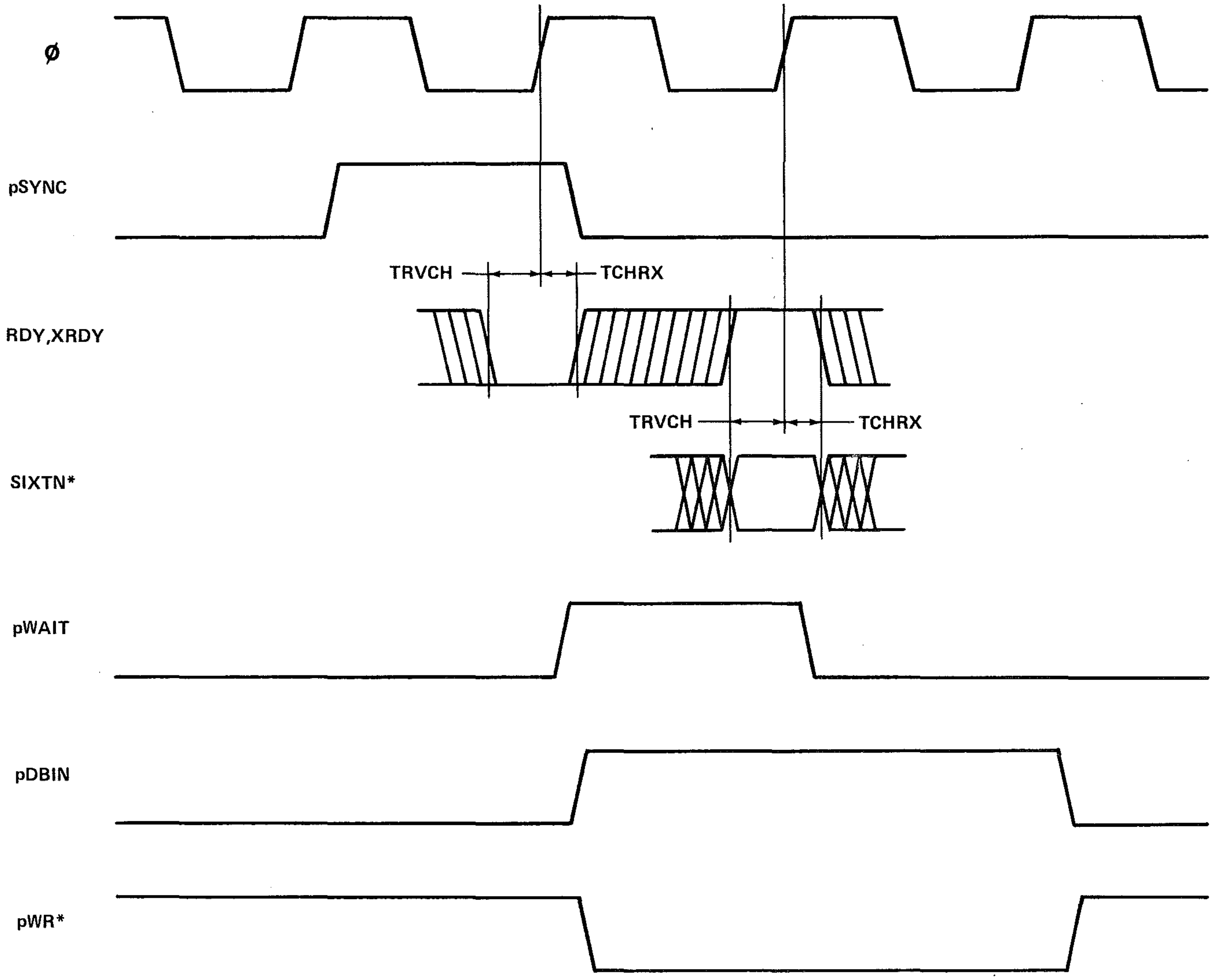
Times specified are in nanoseconds, either minimum or maximum as appropriate for a worst-case specification. Times listed are for a 4 MHz clock, for an 8 MHz clock, and for comparison purposes, the requirements of the proposed S-100 standard at 4 MHz. The maximum clock speed covered by the standard is 6 MHz.

Symbol	Description	4 MHz	8 MHz	S-100
				(4 MHz)
INPUTS				
TAVIV	Allowed address access time	575	260	-
TVLIV	Access time from pSTVAL*	310	130	-
TRVCH	RDY/XRDY/SIXTN* setup time	50	50	80
TCHRX	RDY/XRDY/SIXTN* hold time	0	0	70
TDLIX	Data-in hold time	0	0	50
TDHIV	Driver enable time for slaves	170	50	50
OUTPUTS				
TCLCL,TCHCH	Clock period	250	125	250
TCHCL,TCLCH	Clock pulse width	120	58	113
TCHYH,TCHYL	Clock to pSYNC delay	20	20	100
TYHYL	pSYNC pulse width	240	115	175
TVLYL	pSTVAL* low before pSYNC low	125	62	0
TVHVL	pSTVAL* pulse width high	120	58	50
TVLVH	pSTVAL* pulse width low	120	58	50
TVLYH	pSTVAL* low before pSYNC high	125	62	0
TYHVL	pSYNC high before pSTVAL* low	100	45	50
TAVYH	Address valid before pSYNC	140	60	-
TSVYH	Status valid before pSYNC	160	35	-
except sXTRQ*		70	0	-
TSVVL	Status valid before pSTVAL* low	180	50	40
TDLAX,TDLSX	Address/status hold after pDBIN	160	80	50
TDHDL	pDBIN pulse width	240	115	225
TWLWH	pWR* pulse width	240	115	225
TVLWL	pSTVAL* low before pWR* low	125	62	30
TOVWL	Data-out setup time before pWR*	170	45	25
TWHOX,TWHAX, TWHSX	Data-out/address/status hold	160	80	50
TWLMH,TWHML	MWRITE delay from pWR*	30	30	30

STANDARD BUS CYCLE



BUS CYCLE WITH ONE WAIT STATE



Theory of Operation

Most of the signals described below are shown in the timing diagram accompanying this Theory of Operation. The diagram shows a typical double 8-bit transfer or "double gulp." The names of various signals followed by an asterisk signify an active low signal and correspond to "barred" names on the schematic. For example, ALE* is the same as $\overline{\text{ALE}}$.

Clock Circuitry

The main timing source for the 8086 CPU Card (and thus the entire system) is a 24 MHz crystal clock. The 24 MHz clock is divided by two by half of U9. The other half of U9 synchronizes the output of the SP4/8 switch. 3/4 of U10 is used as a multiplexer to select either 24 MHz or 12 MHz. The 24 MHz clock is also divided by 12 by U26 to provide the 2 MHz "CLOCK" line, S-100 pin 49. The 12 or 24 MHz output from U10 (pin 13) drives a three bit ring counter which divides the 12 or 24 MHz signal by three to provide three one-third/two-thirds duty-cycle 4 or 8 MHz clocks ΦA, ΦB, and ΦC. ΦC is shifted one-half clock by half of U3, the result on pin 2 is buffered by one section of U25 and fed to the 8086 as its clock. ΦC and the 8086 clock are combined by a NOR gate to form Φ2, which in turn is driven onto the bus as Φ (pin 24) and pSTVAL* (pin 25). The timing relationships between these various clocks are shown in the accompanying diagram.

Bus Cycle Control Circuitry

The 8086 itself is operated in the "MAX" mode therefore the start of a bus cycle is indicated by one or more of the three status lines S0*, S1*, or S2* going low. The circuit made up of a NAND gate from U4, an AND gate from U5, and a J-K flip flop from U14 provides the LOAD signal at pin 6 of U5. As shown in the diagram, LOAD goes high when status goes active and is clocked low on the next falling edge of Φ2. LOAD is used primarily to load data into U6, the bus cycle control shift register. LOAD is also used to latch the three 8086 status lines and to set ALE high. ALE (Address Latch Enable) is generated by half of U22, a D flip flop. LOAD is used to set ALE high, ALE clocks low on the next falling edge of Φ, the 8086 clock which is roughly the same as the rising edge of Φ2. ALE is used to latch the address output from the 8086 AD pins. Since the AD pins are multiplexed address/data pins the address is soon replaced by data and must be latched in order to last through the bus cycle.

U6 is a four bit shift register which is used to time the four-clock bus cycles. As a result of the LOAD signal going high, U6 is loaded so that T1 is high and T0, T2, and T3 are low. At the falling edge of Φ2, the "one" loaded into T1 is shifted to the next output. T1 is delayed one-half clock by one section of U20 to form pSYNC (S-100 pin 76). T2 is gated with various status information and then delayed one-half clock by other sections of U20 to provide pDBIN (S-100 pin 78) and pWR* (S-100 pin 77). The diagram shows the timing relationships between T0, T1, T2, T3, pSYNC, pDBIN, and pWR*.

sXTRQ*/SIXTN* Circuitry

When ALE goes high it sets pin 6 of U22 (a D flip flop) low. Pin 6 of U22 is the WAIT signal shown in the diagram, its function is to put the CPU into a wait state if the CPU is attempting to perform a sixteen bit transfer but the device to be transfered to or from can't do sixteen bit transfers. If such a situation occurs the output of the NAND gate U32, pin 6 will be low because SXT will be high indicating that the CPU wishes to do a sixteen bit transfer and SIXTN* will be high indicating that the device to be transfered to or from can't do it. Since the output of the NAND gate is low, WAIT will clock high on the next rising edge of Φ2 after ALE goes away. WAIT connects to pin 3 of U27, an input to a NOR gate. The output will thus go low which drives the READY input to the 8086 low putting it in a wait state (four to be exact). WAIT is connected to the J-K inputs of both sections of

U23, a dual J-K flip flop. Both sections are clocked on the falling edge of T2. If WAIT is high, then DONE (pin 3 of U23) will remain low indicating that we are not done yet. U23 pin 5 (called U23-5 in the diagram) will go high which removes the sXTRQ* signal from the bus since we're not doing a sixteen-bit transfer.

Meanwhile, the bus cycle control shift register clocks T3 high. T3 connects to both the J and K inputs (pins 1 and 4) of U14, a J-K flip flop. When Φ2 falls with T3 high the output at pin 3 of U14 toggles which sets ODDBT high indicating that we've done the even byte and now we're going on to do the odd byte. Notice that ODDBT and T2 go into pins 9 and 10 of U21 (a NAND gate) to provide a signal called RDY* but since T2 has already gone low, RDY* stays high when ODDBT goes high. T3 is connected to the SERIAL IN pin of the bus cycle control shift register so that when T3 clocks low, T0 clocks high. T0 and DONE go into pins 4 and 5 of U21 (a NAND gate) the output of which is connected to the CLEAR pin of the shift register. But since DONE is still low as mentioned above, the shift register isn't cleared. As the shift register clocks two more times and first T1 then T2 go high as a second bus cycle is performed, this time with ODDBT high.

When T2 goes high the second time, ODDBT is high also and RDY* (pin 8 of U21) goes low. RDY* is connected to one of the inputs (pin 5 of U32) of the four-input NAND gate which has SXT and SIXTN* connected to it. The output of the NAND gate (pin 6 of U32) now goes high since one of the inputs went low. Therefore, on the next rising edge of Φ2, WAIT goes low. On the falling edge of T2, DONE goes high since WAIT has gone low. When T2 went low, T3 went high raising the J and K inputs (pins 1 and 4 of U14) putting the ODDBT flip flop in the toggle mode. On the next falling edge of Φ2 ODDBT goes low again indicating that the odd byte has been transfered. When T3 falls, T0 goes high once more and this time DONE is high so the clear input to the bus cycle control shift register (pin 1 of U6) goes high. Since the clear input is asynchronous the clear pulse is self-annihilating: as soon as clear goes low, so does T0. When T0 goes low, so does clear.

If SIXTN* is low (indicating that the device can do a sixteen bit transfer), things are a lot simpler. The D input to the flip flop which generates WAIT (pin 2 of U22) will be high due to SIXTN* being low. Therefore WAIT will never go high, DONE will come on when T2 falls and U23-5 (pin 5 of U23) will stay low keeping sXTRQ* active during the whole cycle. ODDBT will still go high at the end of T3 but will be cleared next time ALE happens.

If the 8086 wasn't doing a sixteen bit transfer SXT would be low and this would have the same effect as SIXTN* being low: no second bus cycle.

The Data Buffer Enable (GA* - GE*) Circuitry and Data Buffers

This circuit takes as its inputs ODDBT, LA0 (Latched A0), INTA (Interrupt Acknowledge), WO (Write & Output), sXTRQ*, DODSB* (Data Out Disable, S-100 pin 23), and a strobe made up of SYNC, T1, and T2, and generates the necessary output enable and gating signals for the address and data buffers and latches.

The strobe referred to is generated at pin 5 of U8 and is low whenever pSYNC, T1, or T2 are high. The strobe is combined with DODSB* such that the strobe is gated off when DODSB* is low. The gated strobe comes from pin 4 of U15 and is active high. A list of the various kinds of cycles and which "enables" (GA*, GB*, etc.) are active and why is given below:

Type of cycle	active enables	reason enable is active
8 bit even read	GA*	none (enables U46 to drive AD8-AD15 with data from DI bus)
	GC*	enables U45 to drive AD0-AD7 with data from DI bus
8 bit even write	GA*	none (enables U46 to drive data from AD8-AD15 onto DI bus)
	GB*	enables U44 to drive data from AD0-AD7 onto DO bus
8 bit odd read	GA*	enables U46 to drive data AD8-AD15 with data from DI bus
	GE*	none (enables U38 to drive AD0-AD7)
8 bit odd write	GA*	none (enables U46 to drive data from AD8-AD15 onto DI bus)
	GD*	enables U43 to drive data from AD8-AD15 onto DO bus
16 bit read	GA*	enables U46 to drive AD8-AD15 with data from DI bus (odd byte)
	GB*	enables U44 to drive AD0-AD7 with data from DO bus (even byte)
16 bit write	GA*	enables U46 to drive data from AD8-AD15 onto DI bus (odd byte)
	GB*	enables U44 to drive data from AD0-AD7 onto DO bus (even byte)
1st byte double-gulp read	GA*	none (enables U46 to drive AD8-AD15 with data from DI bus)
	GC*	enables U45 to drive AD0-AD7 with data from DI bus opens latch U38 to store data on AD0-AD7
2nd byte double-gulp read	GA*	enables U46 to drive AD8-AD15 with data from DI bus
	GE*	enables U38 to drive AD0-AD7 with data stored from previous cycle
1st byte double-gulp write	GA*	none (enables U46 to drive data from AD8-AD15 onto DI bus)
	GB*	enables U44 to drive data from AD0-AD7 onto DO bus (even byte)
2nd byte double-gulp write	GA*	none (enables U46 to drive data from AD8-AD15 onto DI bus)
	GD*	enables U43 to drive data from AD8-AD15 onto DO bus (odd byte,

The DIR input to transceivers U44 and U46 connects to LS1* which is an 8086 status line indicating which direction data is going: LS1* = 0 is read, LS1* = 1 is write. The operation of latch U38 isn't obvious. During the first byte of a double-gulp read GC* is enabled to latch the data on AD0-AD7 from the DI bus. During the second byte of the double-gulp read, data from the DI bus is sent to AD8-AD15 by U46 and U38 is enabled by GE* to drive AD0-AD7 with data saved from the first cycle.

When the CPU does a double-gulp read or write things are actually more complex than they seem. Referring to the bus cycle timing diagram we see that sXTRQ* doesn't go high till the end of pSYNC in the first cycle. Up until this time the CPU "assumed" it was doing a 16 bit transfer. When sXTRQ* goes inactive the cycle changes from a 16-bit transfer to a double-gulp. In the case of a read, GB* turns off and GC* turns on so that the CPU reads the DI bus rather than the DO bus. In the case of a write GB* doesn't turn off but the circuitry that generates GB* switches. In order to generate GB*, one or more of the inputs (pins 1, 2, or 13 of U5) to the AND gate which generates GB* must be low. When sXTRQ* goes low, pins 2 and 13 of U5 go high but pin 1 goes low to keep GB* low. The non-inverting buffer driving pin 13 of U5 is used to keep pin 13 from going high before pin 1 has had a chance to go low.

The Wait Circuitry

Wait states can be added to any bus cycle by pulling either RDY* (S-100 pin 72) or XRDY* (S-100 pin 3) low. These two signals are fed into a NAND gate (pins 9 and 13, resp. of U32) the output of which (pin 8) goes high when a wait state is requested. The output of the NAND gate goes to an input of a NOR gate (pin 11 of U15) the output of which (pin 13) goes low when a wait state is requested. This signal is connected to the D input (pin 12 of U7) of a flip flop which synchronizes the RDY* and XRDY* inputs so the board "looks" at them on the rising edge of Φ (the system clock, S-100 pin 24). The output of the flip flop is called ADV and is available in both polarities. ADV* connects to the SHIFT input of the bus cycle control shift register through a NOR gate so that a wait state is added whenever ADV* is low (active). The T2 output from the shift register sets the ADV flip flop to the inactive state whenever T2 is low, thus wait states can only be added to lengthen the T2 state of the bus cycle. Since pDBIN and pWR* are derived from T2, as T2 is lengthened, so are pDBIN and pWR*. T2 is also used in the sXTRQ*/SIXTN* circuit as described above so that SIXTN* is always sampled on the rising edge of Φ just before T2 falls which always corresponds to one clock before either pDBIN or pWR* go inactive (low or high, respectively).

A switch on the board allows one wait state to be added to each cycle. It works as follows: when the WA position of the dip-switch (pins 4 and 5) is closed, a wait state is generated as long as the output (pin 5 of U7) of the on-board wait state generator flip flop is low. T2 clears the flip flop so that the output is initially low when T2 goes high. If a wait state is requested from the bus either through RDY* or XRDY*, the D input to the flip flop (pin 2) will be low for as long as RDY* or XRDY* are low. When RDY* and XRDY* both go high, the D input will go high and since the flip flop is clocked by O2, the output will go high on the next rising edge. This holds the D input (pin 12 of U7) low till that rising edge and adds one wait state in addition to those requested by RDY* and XRDY*. If RDY* and XRDY* never requested wait states then the input to the synchronizer flip flop will go high on the first available rising edge of Φ2 during T2 and only one wait state will be added.

The Hold Circuitry

In order to understand how this circuit works, it is necessary to understand how the 8086 CPU handles hold requests on its RQ/GT* inputs. To do a hold request from the 8086 it is necessary to pull the RQ/GT* line low for one clock. When the 8086 grants the hold request it will pull the RQ/GT* line low for one clock. When the device requesting the hold wishes to release it, it pulls RQ/GT* low for one clock a second time. Refer to page 5-11 of the MCS-86 User's Manual or page B-10 of The 8086 Family User's Manual for details.

When HOLD* (S-100 pin 74) is high the two inputs (pins 4 and 5 of U1) to the XOR gate are of opposite polarity so that the output is not pulling low. When HOLD* goes low, pin 4 of the XOR gate goes high on the next rising edge of ΦA making both inputs high. Therefore the output goes low bringing RG/GT* low. On the next edge of ΦA, pin 5 of the XOR gate goes low making the inputs of opposite polarity again, thus the output goes high once more and so does RQ/GT*. This action provides the request pulse.

Until the pin 5 input to the XOR gate went low, the clear input (pin 10 of U3) of the hold acknowledge flip flop was held low keeping pHLDA (S-100 pin 26) low. When the pin 5 input to the XOR gate went low, the clear input went high allowing the flip flop to change state. When the grant pulse from the 8086 occures, the RQ/GT* line goes low and the hold acknowledge flip flop will toggle on the next falling edge of ΦC bringing pHLDA high.

When HOLD* returns high the two flip flops of U2 clock back to their original states one by one causing RQ/GT* go be pulled low once again by the XOR gate. The hold acknowledge flip flop is toggled off by the falling edge of ΦC at the second rising edge of ΦA due to the release pulse on RQ/GT*. When the hold acknowledge flip flop toggles off, pHLDA goes low again.

The Status Decoder Circuitry

The three status lines from the 8086 CPU chip S0*, S1*, and S2* are latched by U12 so that status remains correct throughout the bus cycle. The latch (U12) provides both inverting and non-inverting versions of the latched status. The status decoder circuitry consists of two two-input NAND gates (pins 8-13 of U33), one two-input NOR gate (pins 4-6 of U16) and a two-line to four-line demultiplexer (half of U35). The table below shows the eight possible combinations of the three 8086 status lines, the type of cycle involved, and the decoded S-100 status.

S2*	S1*	S0*	type of cycle	sMEMR	sINP	sM1	sOUT	sHLTA	sWO*	sINTA
0	0	0	int. ack.	0	0	1	0	0	1	1
0	0	1	I/O read	0	1	0	0	0	1	0
0	1	0	I/O write	0	0	0	1	0	0	0
0	1	1	halt	0	0	0	0	1	1	0
1	0	0	code access	1	0	1	0	0	1	0
1	0	1	memory read	1	0	0	0	0	1	0
1	1	0	memory write	0	0	0	0	0	0	0
1	1	1	passive	-	-	-	-	-	-	-

The address latches and buffers.

The address latches consist of four 74LS373 type latches. U37 latches A16-A19 and drives A20-A23 low since these top four address lines are not supplied by the 8086 CPU chip. U42 latches A8-A15. U39 and U40 latch A0-A7. U40 performs exactly the same function as U39 except it always sets A0 high and is only used during the second byte of a double-gulp. The gate inputs to each latch are connected to ALE to latch the address from the address/data lines. The four sections of U29 (quad NAND) provide the four output enables for the latches.

output enable	when active
ODD*	second byte of a double-gulp
EVEN*	all cycles except those when ODD* is active
IO*	all I/O cycles when the "I/O" jumper is in the "8" position
MEM*	all cycles except those when IO* is active

When IO* is active during an I/O cycle with the "I/O" jumper in the "8" position, U41 is enabled to drive the address from A0-A7 onto A8-A15. This provides the "address mirror" feature for I/O boards which use A8-A15 for I/O port address decoding.

Address, Data, Status, and Control Disables

The address latches and buffers are enabled by four sections of U29, a quad NAND gate. One input to each gate is driven by ADSB* (S-100 pin 22) through a non-inverting buffer. If ADSB* goes low, all address enables are turned off.

The data buffers are enabled as described in "The Data Buffer Enable Circuitry" section. The DODSB* (S-100 pin 23) input disables the "strobe" which is necessary for any of the enables to be active thus deactivating all enables.

The status buffer, U47, has its output enable connected to SDSB* (S-100 pin 18) through an inverting buffer so that when SDSB* goes low, the status buffer is disabled.

The control disable (CDSB*, S-100 pin 19) works on the control buffer, U28, in exactly the same manner as the status disable works on the status buffer.

MWRITE Circuit

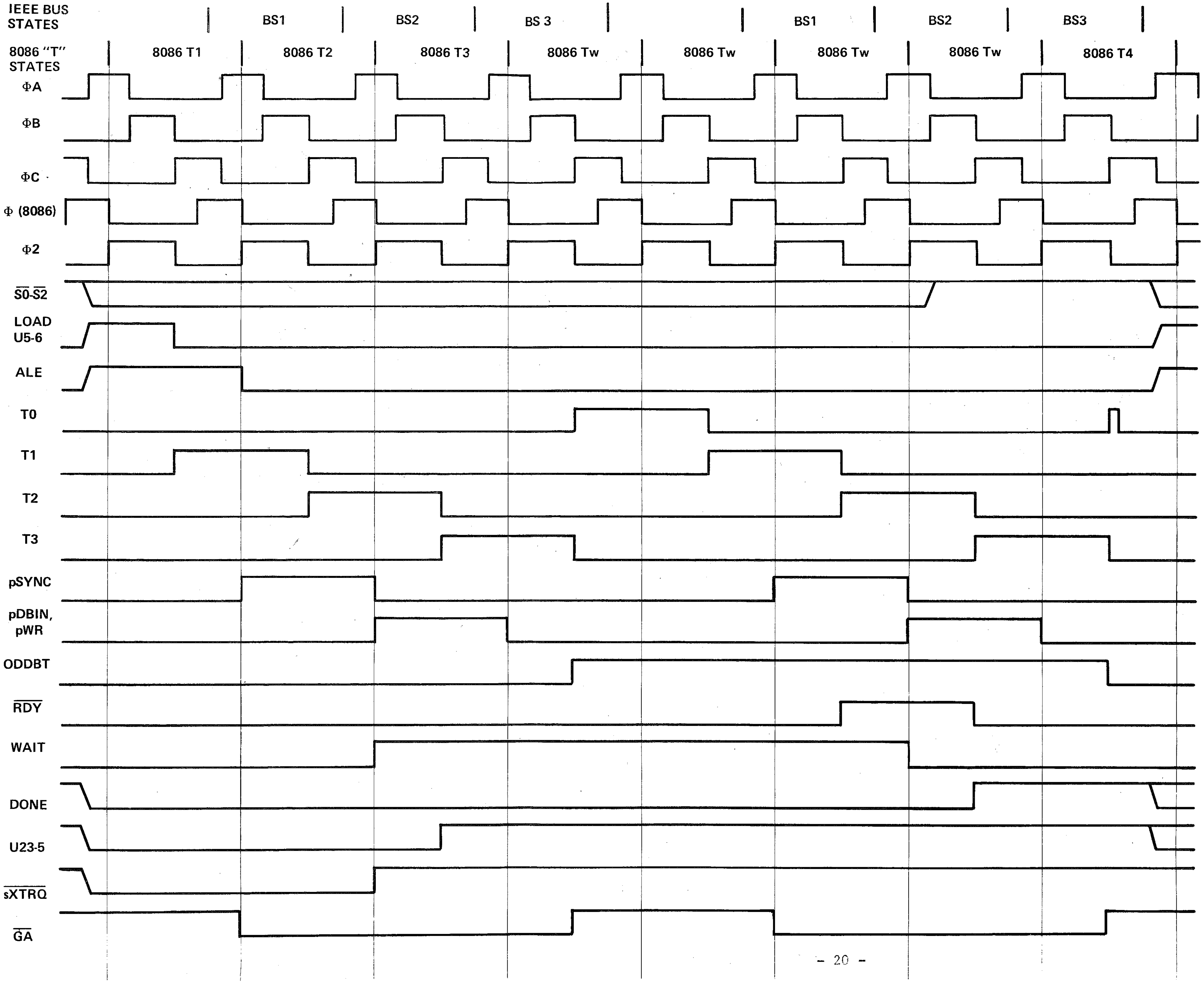
The sOUT and pWR* signals on the bus (pins 45 and 77, resp.) are NORed together to form MWRITE. MWRITE is driven onto pin 68 of the bus and goes through the "MWRTE" jumper so that it can be disabled if another device in the system provides MWRITE. Notice that since sOUT and pWR* are read from the bus, MWRITE will be valid no matter what bus master provides sOUT and pWR*.

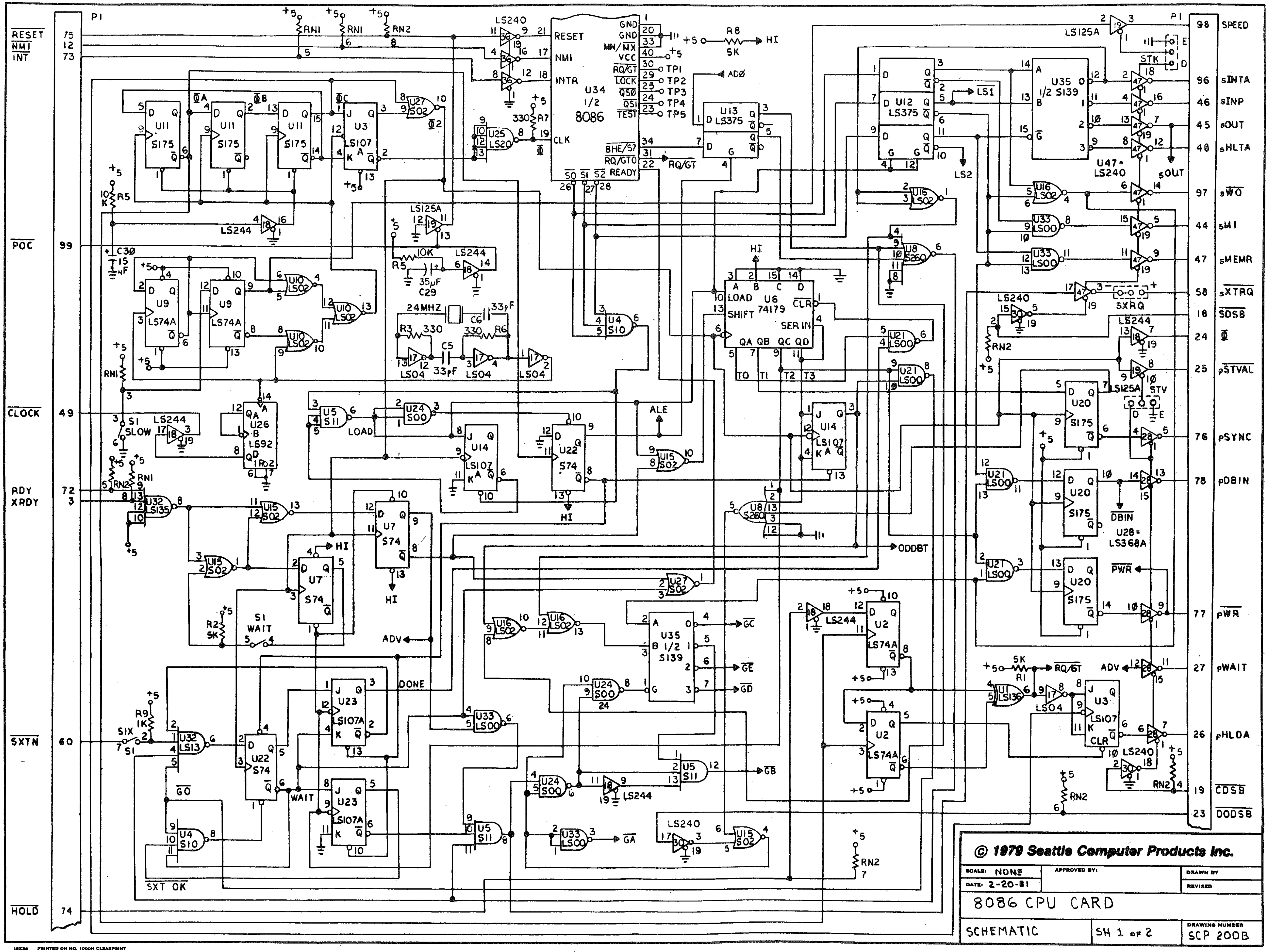
The PHANTOM* Circuit

The four top address lines provided by the 8086 (A16-A19) are NORed together and can drive PHANTOM* if desired by placing the "PHAN" jumper in the "+" position. In this way, whenever the CPU addresses memory above the first 64K the PHANTOM* line will be activated to disable nonextended-address memory in the first 64K. Notice that the source for A16-A19 is the bus itself so that this circuit will operate for any bus master. Also notice that it is NOT an open collector output and therefore if used should be the only PHANTOM* driver in the system.

The POC* and RESET* Circuitry

The power-on-clear circuitry actually consists of two seperate R-C networks to provide two power-on-clear signals. The first comes from R5 & C30 and resets the ring counter described in the "clock circuitry" section. The second uses R4 & C29 and provides the standard system power-on-clear. C30 is only half as big as C29 while R4 and R5 are the same size, thus the clock power-on-clear releases before the other one so that the clock is up and running before the standard power-on-clear releases. POC* also drives RESET* through one section of a 74LS125A wired as a "pseudo open-collector" driver. RESET* drives the RESET input to the 8086 CPU chip through a schmitt-trigger buffer.


IEEE BUS STATES
BS1
BS2
BS 3
BS1
BS2
BS3
8086 "T" STATES
8086 T1
8086 T2
8086 T3
8086 Tw
8086 Tw
8086 Tw
8086 Tw
8086 T4
ΦA
ΦB
ΦC
Φ (8086)
Φ2
$\overline{S0}$-$\overline{S2}$
LOAD U5-6
ALE
T0
T1
T2
T3
pSYNC
pDBIN, pWR
ODDBT
$\overline{RDY}$
WAIT
DONE
U23-5
$\overline{sXTRQ}$
$\overline{GA}$



© 1979 Seattle Computer Products Inc.

SCALE: NONE
APPROVED BY:
DRAWN BY
DATE: 2-20-81
REVISED

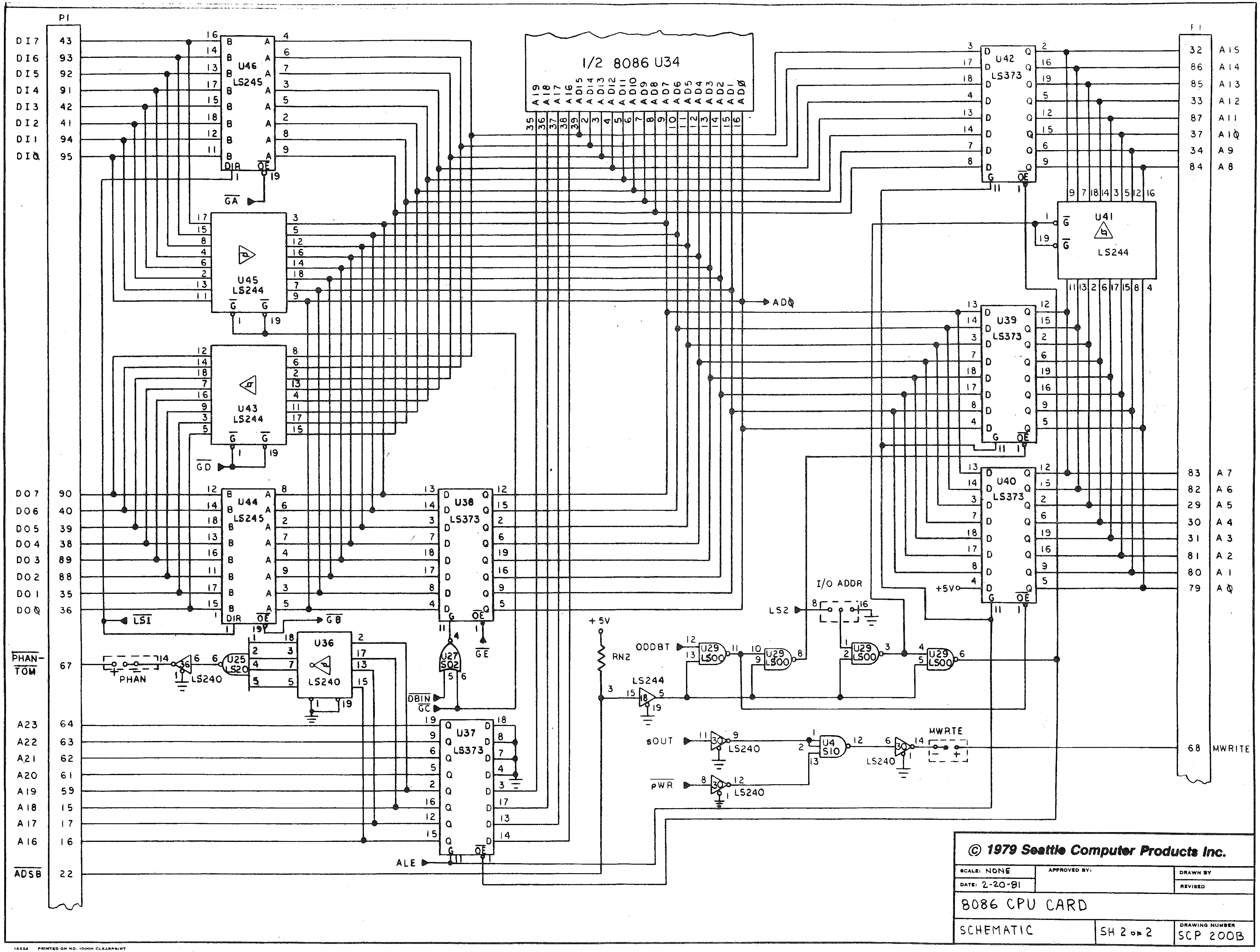
8086 CPU CARD

SCHEMATIC

SH 1 OF 2

DRAWING NUMBER SCP 200B

18X24 PRINTED ON NO. 1000H CLEARPRINT



© 1979 Seattle Computer Products Inc.

SCALE: NONE
APPROVED BY:
DRAWN BY
DATE: 2-20-81
REVISED

8086 CPU CARD

SCHEMATIC SH 2 of 2

DRAWING NUMBER SCP 200B

18334 PRINTED ON NO. 1000H CLEARPRINT

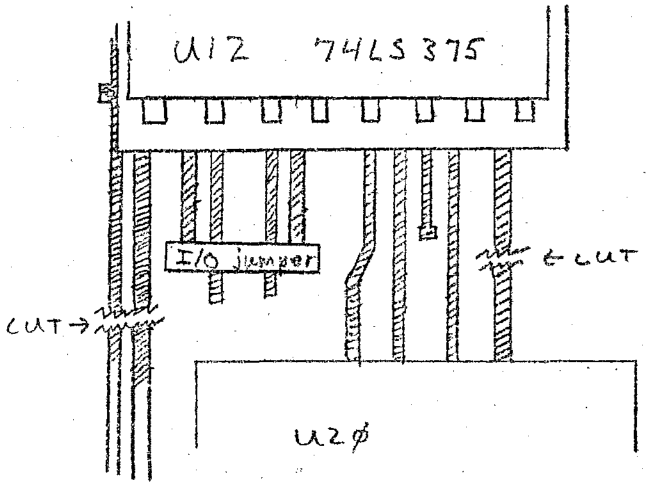
How to install the 8087 adaptor on a SCP 8086 CPU board.

Note: Read over these instructions carefully before you start installing the adaptor. This installation requires cutting traces, soldering wires, and drilling holes, if you're not sure you're up to it, please send the CPU and adaptor back to SCP and we'll install it for you at no charge. (We will charge you if you botch up the installation and return it to us to fix).

Step 1 Carefully remove the 8086 chip from its socket on the 8086 CPU and place it someplace safe. Also remove U12 (74LS375).

Step 2 Cut the copper traces connecting to pins 26, 27, and 28 of the 8086 on the back side of the 8086 CPU board using a razor blade or Exacto knife. Cut as close to these pins as possible.

Step 3 Cut the three traces below U12 on the front side of the board shown in the diagram below. Two of the traces pass just to the left of the I-O jumper, the other trace comes out from between pins 7 and 8 of U12. Make these cuts below U12 and above U20.



Step 4 Solder a wire jumper between pin 26 of U34 and pin 1 of U12. Kynar wire-wrap wire is very good for this purpose (you can even get it at Radio Shack). Make sure that the wire is fairly tight so it won't get snagged on things and break and that the insulation isn't damaged.

Step 5 Solder a wire jumper between pin 27 of U34 and pin 7 of U12.

Step 6 Solder a wire jumper between pin 28 of U34 and pin 9 of U12.

Step 7 The male quick-disconnect connector supplied with the 8087 adaptor will be used to tap the +8 volt power coming into the 8086 CPU board to go to the 8087 adaptor. Locate the +8 volt trace coming from bus pins 1 and 51 under the "MADE IN USA" marking on the front of the board. Using the two solder pins of the connector as a guide for spacing, scratch two marks in the solder mask over the +8 volt trace on the back of the board. These should be in the center of the +8 volt trace and just under the "E" and the "U" in "MADE IN USA". Use a drill bit slightly larger than the solder pins of the male quick-disconnect connector (about 1/16 to 5/64 inch) to drill two holes in the board where you have marked.

✓ Step 8 Scratch the solder mask off the +8 volt trace around the two holes so solder will stick to the bare metal. Put the two pins of the male quick-disconnect connector through the holes so the pins stick out on the back and solder them to the +8 volt trace. The +8 volt trace is fairly thick and requires lots of heat to get the solder to flow well. Be sure you have made a good connection because this connection is required to carry quite a bit of current.

✓ Step 9 Solder a short piece of wire between pins 30 and 31 of U34 on the 8086 CPU board.

✓ Step 10 Remove the nuts attaching the two voltage regulators to the 8086 CPU board but leave the lockwashers on. Add a second lockwasher to each and screw on the metal spacers provided in place of the nuts. Screw these down tight so they provide a ground path to the 8087 adaptor.

✓ Step 11 Check to make sure that pins 31, 33, and 40 of the pins sticking out of the 8087 adaptor have been cut off. Plug the pins of the 8087 adaptor into the 8086 socket on the CPU Support board. Be absolutely certain that each pin goes into the proper hole in the socket and that none are bent.

✓ Step 12 Put a lockwasher between the top of each metal spacer and the 8087 adaptor's printed circuit board. Put a screw in each of the two holes in the 8087 adaptor and tighten them down well. Don't put a lockwasher under the screw-head because it will damage the printed circuit board.

✓ Step 13 Double check the order of assembly of the voltage regulators and spacers. From the bottom up: screw-head, 8086 CPU board, heatsink, voltage regulator, lockwasher, lockwasher, threaded metal spacer, lockwasher, 8087 adaptor board, screw-head.

✓ Step 14 Plug the connector from the 8087 adaptor onto the male quick-disonnect you installed earlier to provide power to the adaptor.

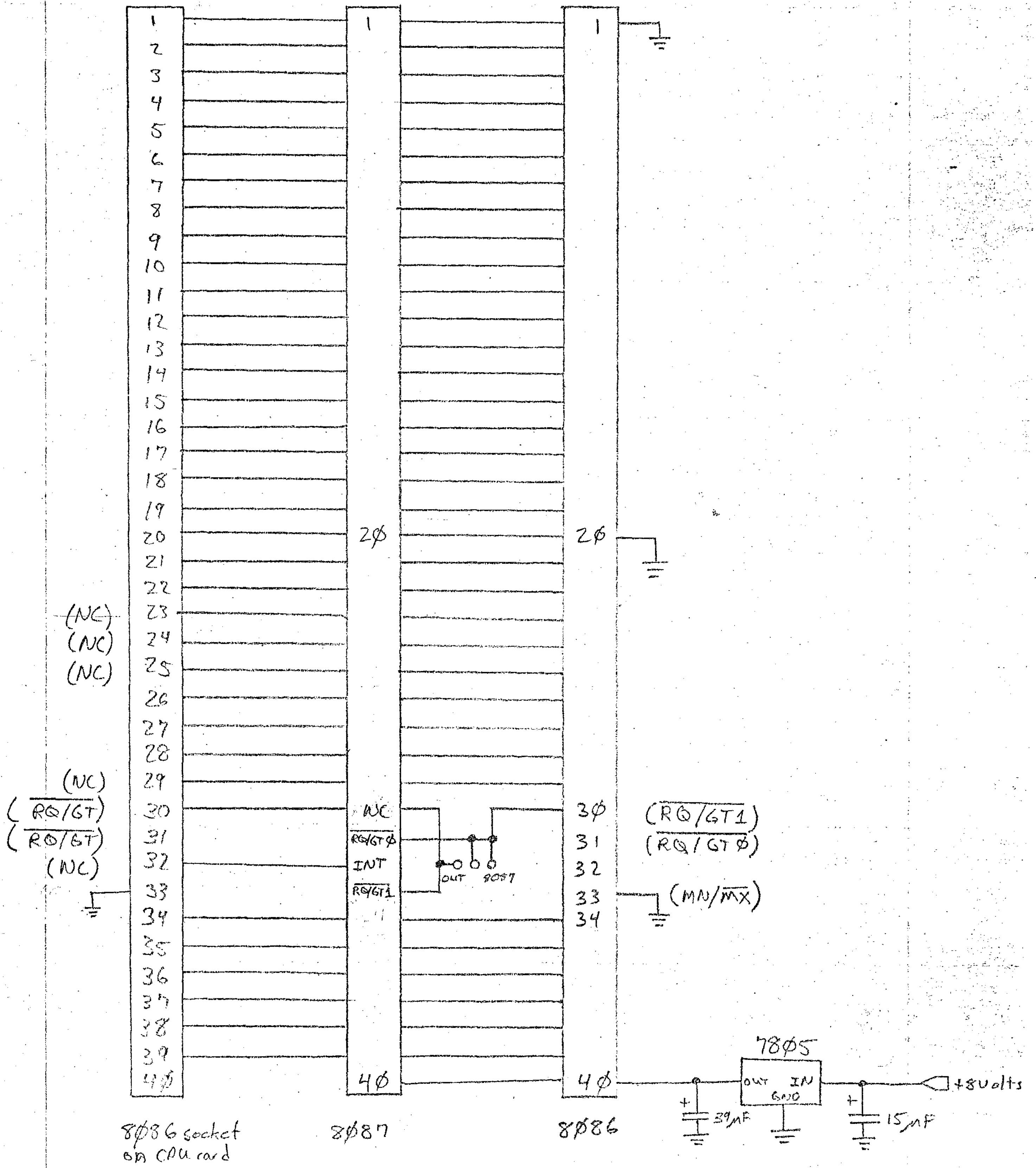
Step 15 Put U12 (74LS375) back in its socket, making sure the notch points to the left.

✓ Step 16 Before installing either the 8086 or the 8087, put the CPU board into an S-100 bus and measure the output voltage of all three regulators. The top pin should have between 4.8 and 5.2 volts on it. The bottom regulator on the 8086 CPU board can best be measured by measuring pin 20 of one of the 20-pin I.C.'s along the bottom row.

Step 17 If the above test passed, remove the board from the S-100 bus and install the 8086 chip, with pin 1 facing left, in the top socket of the 8087 adaptor (the one labeled "8086"). If you have an 8087, put it in the other socket (the one labeled "8087").

Step 18 If an 8087 is installed, put the jumper on the adaptor in the "8087" position, if no 8087 is installed, put the jumper in the "OUT" position. This jumper controls DMA handshaking. If the 8087 is installed, DMA requests go directly to it; if the 8087 is not installed, DMA requests are passed on to the 8086.

8Ø87 piggy back adaptor



One-Year Limited Warranty

WARRANTEE AND WARRANTY PERIOD

The Seattle Computer Products (hereinafter referred to as SCP) warranty for this product extends to the original purchaser and all subsequent owners of the product for a period of one year from the time the product is first sold at retail and for such additional time as the product may be out of the owner's possession for the purpose of receiving warranty service at the factory.

WARRANTY COVERAGE

This product is warranted to be free from defects of material and workmanship and to perform within its specifications as detailed in the instruction or operating manual during the period of the warranty.

This warranty does not cover damage and is void if the product has been damaged by neglect, accident, unreasonable use, improper repair, or other causes not arising out of defects in material or workmanship.

WARRANTY PERFORMANCE

During the warranty period, SCP will repair or replace defective boards or products or components of boards or products upon written notice that a defect exists. Certain high value parts may have to be returned to SCP prior to replacement. Other components will be replaced without the part having to be returned to the factory with the exception the SCP retains the right in all cases to examine the defective board or other products prior to the items replacement under the warranty. In the event the return of the board, product, or component is requested by SCP under this warranty, the owner shall ship the item prepaid to the SCP factory. SCP will pay for shipment of replacement items back to the owner. All repairs or replacements under this warranty will be performed by SCP within five working days of receipt of notice of defect or return of components as called for under this warranty.

WARRANTY DISCLAIMERS

While high reliability was a major design factor for this product and care was used in its manufacture, no certainty can be achieved than any particular product will operate correctly for any specific time. No representation is made by SCP that this product will not fail in normal use. Because of the inability to guarantee 100% reliability, SCP shall not be liable for any consequencial damange the user may suffer because the products fails to function reliably 100% of the time. Any implied warranties arising from the sale of this product are limited in duration to the warranty period defined above.

LEGAL REMEDIES

This warranty gives the purchaser specific legal rights. He may have additional rights which vary from state to state.

SHIPPING INSTRUCTIONS

In the event it becomes necessary toreturn the product or component to SCP, also return a written explaination of the difficulty encountered along with your name, address and phone number. Package the items in a crushproof container with adequate packing material to prevent damage and ship prepaid to:

Seattle Computer Products
1114 Industry Drive
Seattle, Washington 98188