MAR, INC. • 23600 Mercantile Road • Cleveland, Ohio 44122 • Phone: (216) 464-7410

8086/S100 CPU - INTERRUPT

INTRODUCTION

8086/S-100 CPU - INTERRUPT

The Tecmar 16 bit S-100 CPU-Interrupt board is based on the INTEL 8086 microprocessor and consists of the CPU with 8 levels of vectored interrupt and S-100 interface with extended addressing. The board complies with the proposed IEEE 16 bit S-100 protocol providing the ability to fetch and store two bytes at a time which effectively doubles the previous capacity of the bus. This board can access I/O devices (byte-at-a-time or word-at-a-time) through memory locations or I/O ports providing compatibility with conventional S-100 8080/Z80 systems. The board permits a one megabyte address space by utilizing a 20 bit address which is formed using the extended address protocol in the proposed standard for the S-100 bus allowing full use of the 8086.

The INTEL 8086 microprocessor is fully described in INTEL`s MCS-86$^{TM}$ User`s Manual. (Please refer to this since the information on the 8086 integrated circuit will not be repeated in this manual.) The primary features include eight 16 bit registers, hardware multiply and divide, hardware trace mode, and inherently position independent code via base registers. The 8086 machine language provides a 5 to 10 fold base increase in performance over the 8080. Assembly language programs for the 8080 can be easily translated into 8086 programs with approximately a factor of two increase in performance.

The CPU-Interrupt board can be used with 8-bit memory boards, but memory capable of 16-bit transfers will allow faster operation. The board has a power-on jump feature to $FFFF0_H$ where a monitor is provided (on the PROM-I/O board).

Depending upon the crystal installed, the system can be set up to operate at 4, 5, or 8 MHz.

The 8086 uses a full 16-bit address for I/O devices unlike the 8080 which duplicates the 8-bit I/O address on A15-A8 and A7-A0.

The board is shipped with all jumpers in a standard configuration but can easily be changed to take advantage of the optional features, such as the wait state generator.

FEATURES

8086/S-100 CPU - INTERRUPT

1) S-100 bus compatible

2) 16 bit microcomputer (8086)

3) 8 levels of vectored interrupt

4) Designed for operation at 4, 5, or 8 MHz

5) Compatible with conventional S-100 8080/Z80 boards

6) One megabyte address space

7) Full 16 bit address for I/O device

8) On-board regulators

9) Operates with 8259A vectored interrupt chip or from $\overline{\text{PINT}}$ (line 73) of S-100 bus.

10) Operates with 8 bit wide memory or 16 bit wide memory

REQUIREMENTS

1) S-100 bus

Processor Timing: using a crystal oscillator, supply the processor clock (4,5, or 8 MHz) to the processor and the bus ϕ_2 line. Supply a 2 MHz peripheral timing clock to the bus. Provide power-on clear to the processor and bus and synchronize the reset to the processor. Synchronize the wait state generation with the clock (XRDY, PRDY AND CPU READY). This circuit is centered around the Intel 8284 clock generator chip.

Bus Control: From the processor clock and the S0, S1, S2 signals provided by the processor, generate the S-100 bus status lines and the S-100 bus command/control lines, disabling them on the assertion of $\overline{\text{STAT DSB}}$ and $\overline{\text{C/C DSB}}$ respectively. Handle DMA request/grant timing. Provide timing and control signals to the rest of the board (ALE, PSYNC, DT/$\overline{\text{R}}$, $\overline{\text{DEN}}$, $\overline{\text{INTA}}$, $\overline{\text{IORDC}}$, $\overline{\text{AIOWC}}$). Centered on an Intel 8288 bus controller chip.

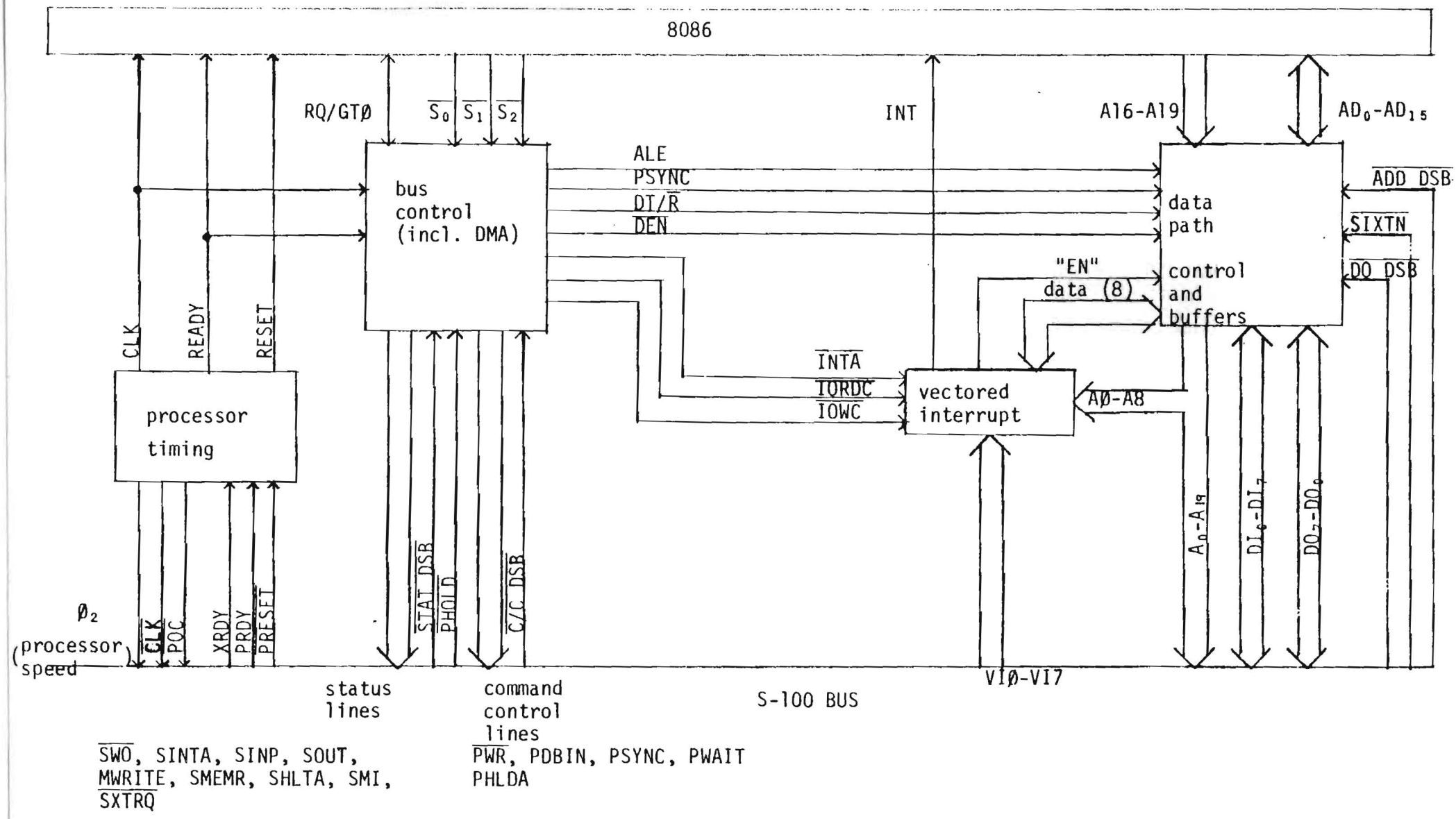
Vectored Interrupt: Based on the $\overline{\text{VI0}}$-$\overline{\text{VI7}}$ lines on the bus generate and vector interrupts to the 8086 processor (INT). Based on an 8259A priority interrupt chip.

Data Path Control and Buffers: Using signals generated by the bus control section: Latch the address from AD0-A19 and drive the address lines A0-A19 on the bus, drive the 8086 AD inputs from the appropriate S100 bus lines at the appropriate times, drive the bus from the 8086 AD outputs at the appropriate times. The address and data outputs to the bus may be disabled by $\overline{\text{ADD DSB}}$ and $\overline{\text{DO DSB}}$ respectively. The output depends on $\overline{\text{SIXTN}}$ rather than on the internally generated $\overline{\text{SXTRQ}}$ to avoid any problems which might arise from bus contention in the event of the failure of a 16-bit operation. This block is centered around a set of bidirectional bus transceivers and address latches.

I.C.`s associated with block diagram (See Component layout).

Block	Type	I.C. #
8086	8086-4	32
Processor Timing	74LS05	2
	74LS04	3,5
	74367	17
	74LS161	29
	8284	31
Bus Control	74LS08	30
	8288	24
	74368	9, 10
	74LS04	4, 26 (DMA)
	74LS100	16, 36 (DMA)
	74LS10	33
	74LS02	35 (DMA)
	74LS175	34, 27 (DMA)
	74367	17
	74LS74	25 (DMA)
	74LS136	11 (DMA)
Vectored Interrupt	74LS05	2, 6
	74LS04	3, 4, 5
	8259A	13
	make sure strap for interrupt source is correct	
Data Path	8282	7, 14, 15
	74LS10	33
	74LS04	4, 5, 18
	74LS245	8, 39
	74LS32	1, 23
	74LS11	12
	74164	19
	74125	20
	74LS00	21
	74LS74	22
	74LS08	28
	74LS173	37
	81LS95	38

BLOCK DIAGRAM OF 8086 CPU CARD



8086/S-100 CPU - INTERRUPT

The reader of this section is cautioned that he is expected to be reasonably familiar with use of the 7400 series TTL and have reviewed the data sheets for the Intel 8086, 8282, 8284, 8286, 8288, and 8259A ICs in chapter 5 of the MCS-86 users` manual. He is also expected to have reviewed a copy of the proposed IEEE standard for the S-100 bus.

The first step in interfacing a processor to the S-100 bus is defining the exact meaning of each bus signal to be generated under all possible conditions. The second step is establishing what signals are available from the processor. A mapping is then defined to create those signals required at the processor and at the bus from those available at the bus and at the processor. The mapping must then be examined to ensure tha it places no unnecessary restrictions upon the operation of the processor or the bus and that it will in fact work even under worst-case conditions.

In order to conform with the proposed standard for the S-100 bus*, the CPU card must generate the following signals:

A19	DI7/DATA15	
A18	DI6/DATA14	
A17	DI5/DATA13	PHLDA
A16	DI4/DATA12	SM1
A15	DI3/DATA11	SOUT
A14	DI2/DATA10	SINP
A13	DI1/DATA9	SMEMR
A12	DI0/DATA8	SHLTA
A11	DO7/DATA7	SXTRQ
A10	DO6/DATA6	PSYNC
A9	DO5/DATA5	$\overline{\text{PWR}}$
A8	DO4/DATA4	PDBIN
A7	DO3/DATA3	SINTA
A6	DO2/DATA2	$\overline{\text{SWO}}$
A5	DO1/DATA1	
A4	DO0/DATA0	
A3		
A2		
A1		
A0		

---

*See IEEE "Computer" magazine, March 1979, pp. 20-44.

In addition to those lines, the CPU card or front panel must generate certain timing signals. Since there is no front panel, the CPU must generate

ϕ_2	bus timing
CLOCK	peripheral timing
MWRITE	memoy write
$\overline{\text{POC}}$	power-on clear

Besides the data lines (DO_7-DO_0 and DI_7-DI_0), the inputs to the CPU board from the S-100 bus are as follows:

XRDY	$\overline{\text{PRESET}}$	$\overline{\text{VI0}}$
$\overline{\text{STAT DSB}}$		$\overline{\text{VI1}}$
$\overline{\text{C/C DSB}}$		$\overline{\text{VI2}}$
$\overline{\text{ADD DSB}}$		$\overline{\text{VI3}}$
$\overline{\text{DO DSB}}$		$\overline{\text{VI4}}$
$\overline{\text{SIXTN}}$		$\overline{\text{VI5}}$
PRDY		$\overline{\text{VI6}}$
$\overline{\text{PINT}}$		$\overline{\text{VI7}}$
$\overline{\text{PHOLD}}$		

in addition to the +8V and ground lines on the bus.

Based on the 8086 documentation, operating in "MAX" mode, the following signals are available (see pages 5-9 and 5-11 of the MCS-86 User`s Manual).

AD_{15}-AD_0

A_{19}, A_{18}, A_{17}, A_{16}

$\overline{\text{BHE}}$

$\overline{\text{RD}}$

$\overline{S}_2$, $\overline{S}_1$, $\overline{S}_0$

$\overline{\text{RQ}}/\overline{\text{GT}}_0$, $\overline{\text{RQ}}/\overline{\text{GT}}_1$

$\overline{\text{LOCK}}$

QS_1, QS_0

And the 8086 requires the following inputs:

READY
INTR
TEST (tie low)
RESET
CLK

Adding an 8284 (I.C. 31) clock generator provides the following outputs:

RESET	(8086 input)
OSC	
CLK	(8086 input)
PCLK	
READY	(8086 input)

and, the following inputs are required:

X1, X2	connect to 12, 15, or 24 MHz fundamental frequency crystal
TANK	leave open for use with overtone mode crystals).
F/$\overline{C}$	tie low (input is from crystal, not EFI)
EFI	tie low (unused input)
CSYNC	tie low (unused multiprocessor sync facility)
$\underline{RDY2}$	
$\underline{\overline{AEN2}}$	tie low
$\overline{AEN1}$	tie high
$\underline{RDY1}$	tie low unused "alternate bus" ready line
$\overline{RES}$	

Adding an 8288 (I.C. 44) bus controller provides the outputs (see MCS-86 User`s Manual, pages 5-33).

$\overline{MRDC}$

$\overline{MWTC}$

$\overline{AMWC}$

$\overline{IORDC}$

$\overline{IOWC}$

$\overline{AIOWC}$

$\overline{INTA}$

DT/$\overline{R}$

DEN

MCE

ALE

requiring the inputs

$\overline{S}_0$ (from 8086)

$\overline{S}_1$ (from 8086)

$\overline{S}_2$ (from 8086)

CLK (from 8284)

$\overline{AEN}$ (tie low)

CEN (tie high)

IOB (tie low)

Summarizing the required inputs, we have

INTR
RDY2
$\overline{RES}$

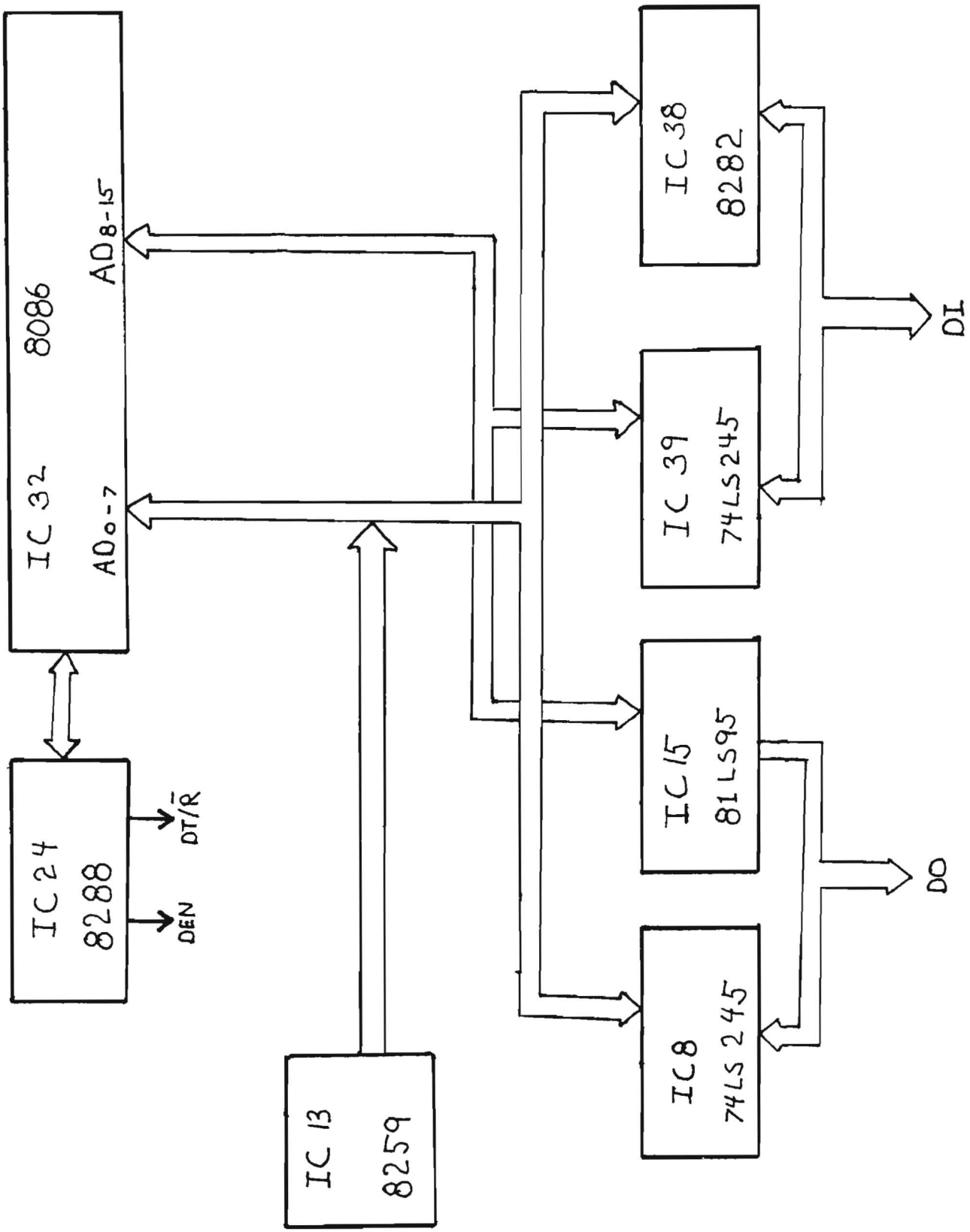
and the data input lines time multiplexed into AD_{15}-AD_0.

The 8086's complex scheme of addressing and intstruction look-ahead prohibits the use of a conventional front panel for altering memory, examining memory, or starting program execution at an arbitrary location as is done with IMSAI or Altair 8080-based S-100 systems. The convention used in the systems which have no front panel is a simple switch called "RESET" which when pressed connects line 75 of the bus ("$\overline{PRESET}$") to ground and no other lights or switches. This scheme seems to be most reasonable for use in an 8086 system, and thus it is adopted. A software-based front panel could of course be designed with lights and switches as I/O devices under program control.

The most complicated part of the mapping between the 8086 and the S-100 bus is in the data paths. There are so many functions that the only reasonable way to list them is in a table.

function	data source	data destnation
8-bit memory write or 8-bit I/O output to an even numbered port or location.	AD_7-AD_0	DO_7-DO_0
8-bit memory write or 8-bit I/O output to an odd numbered port or location	AD_{15}-AD_8	DO_7-DO_0
16-bit memory read or 16-bit I/O output	AD_{15}-AD_8 AD_7-AD_0	DI_7-DI_0 DO_7-DO_0
8-bit memory read or 8-bit I/O input from even-numbered port or location	DI_7-DI_0	AD_7-AD_0
8-bit memory read or 8-bit I/O input from odd-numbered port or location	DI_7-DI_0	AD_{15}-AD_8
16-bit memory read or 16-bit I/O input	DI_7-DI_0 DO_7-DO_0	AD_{15}-AD_8 AD_7-AD_0
16-bit read from 8-bit memory	DI_7-DI_0	AD_7-AD_0 AD_{15}-AD_8
16-bit write to 8-bit memory	AD_7-AD_0 AD_{15}-AD_8	DO_7-DO_0
interrupt acknowledge 8259A chip not installed	DI_7-DI_0	AD_7-AD_0
interrupt acknowledge 8259A chip installed	8259A data lines	AD_7-AD_0

These transfers are conceptually implemented as follows:



The transfers needed may be organized by destination.

destination	Source
DO	$AD_7 \ldots AD_0$ $AD_{15} \ldots AD_8$
DI	$AD_{15} \ldots AD_8$
AD_7-AD_0	DI DO on board 8259A
AD_{15}-AD_8	DI

The bus buffers will all be disabled when DO DSB is pulled low (for DMA), or when the 8259A is acknowledging an interrupt. The DIR inputs of the two transceivers are driven by the DT/$\overline{R}$ line of the bus controller (8288).

The control lines for the drivers must now be considered. The IC's at the data bus interface are enabled as follows:

IC 80 (DO_0-DO_7):

Enabled for	Direction
8-bit write to even byte	8086 to bus
16-bit read (even byte)	bus to 8086
16-bit write (even byte)	8086 to bus

IC 39 (DI_0-DI_7):

Enabled for	Direction
8-bit read from odd byte	bus to 8086
16-bit read (odd byte)	bus to 8086
16-bit write (odd byte)	8086 to bus

IC 15 (DO_0-DO_7):

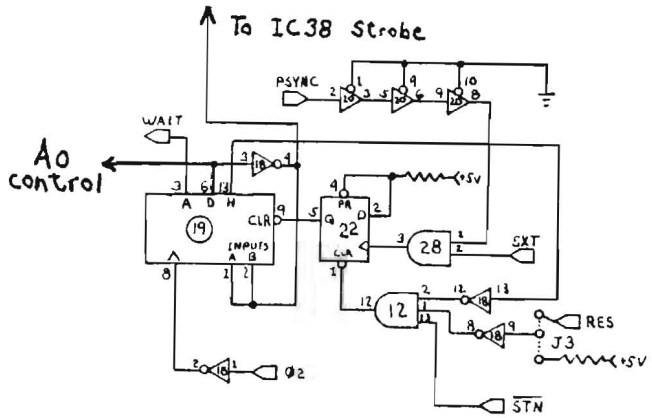
8-bit write to odd byte

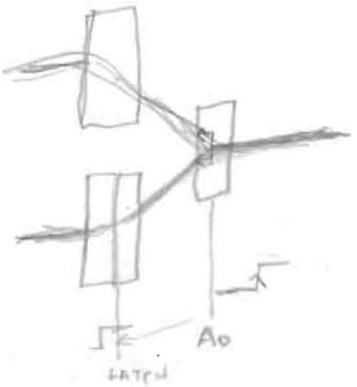
IC 38 (DI_0-DI_7):

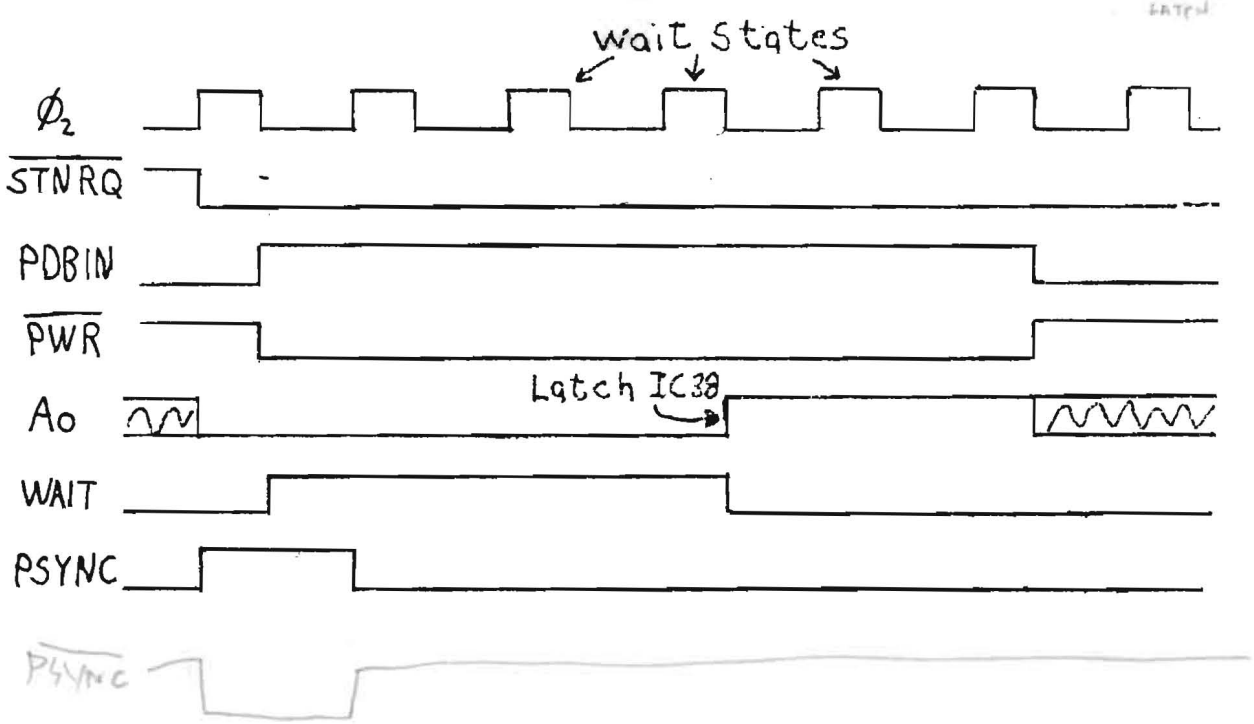
8-bit read from even byte (data is latched when A0 goes high on second half of 16 bit read from 8-bit memory.)

All of these IC's also require that DEN (from 8288) and $\overline{EN}$ (from 8259A) both be high for a transfer to take place.

The 16-bit transfer to/from 8-bit memory is accomplished by a circuit that detects the abscence of SIXTN when SXTRQ is active. If this happens, wait states are inserted while the circuit holds AO low for three clock periods and then holds AO high for the rest of the bus cycle (3 more clock periods). The transfer proceeds as if two memory operations were done back-to-back at the even and odd byte locations. The only difference is that the data from the even byte (IC 38) is latched while the odd byte is read and IC 38 is still enabled.

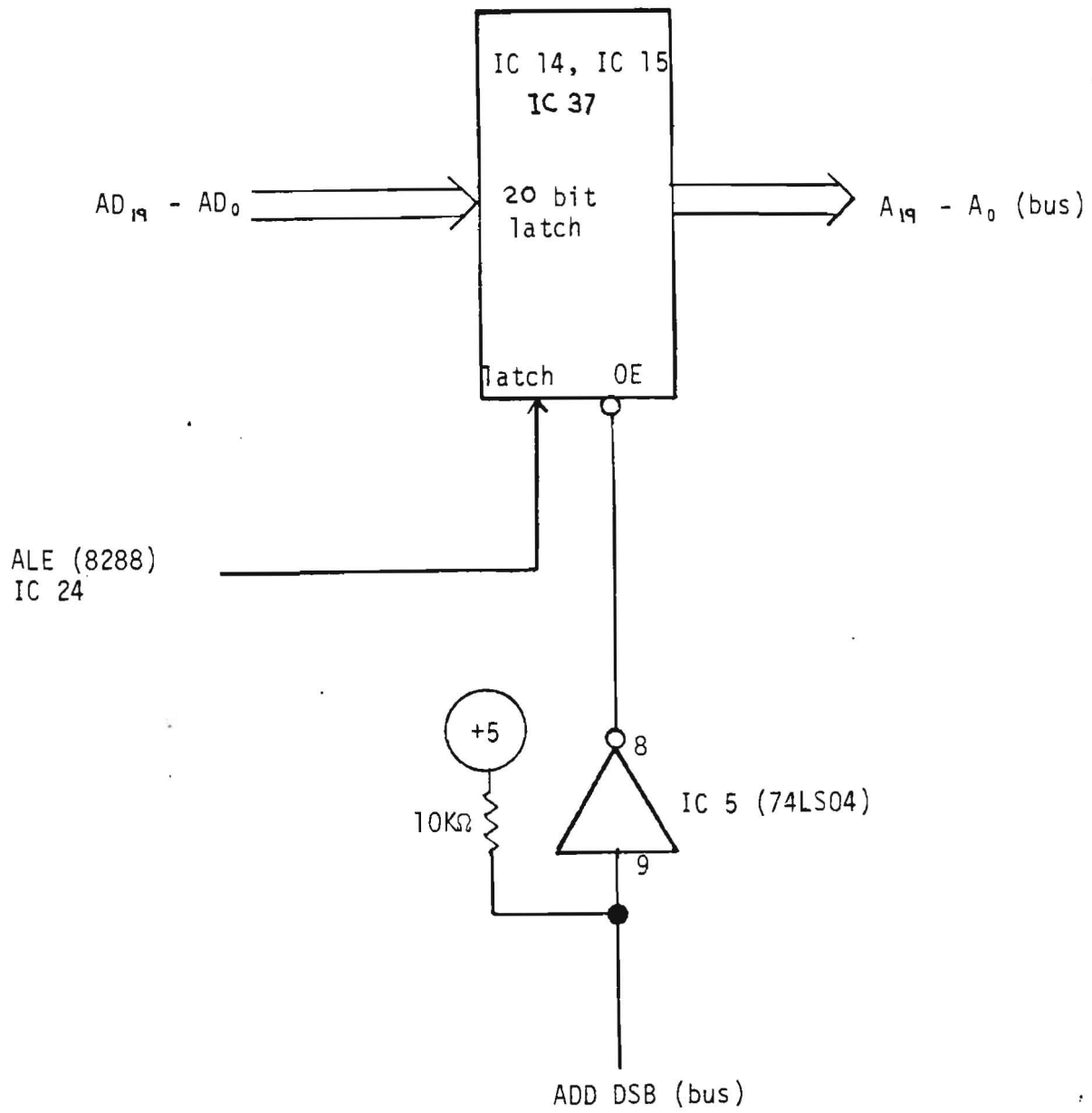






The details on the 8259 interrupt chip are given in its section later.

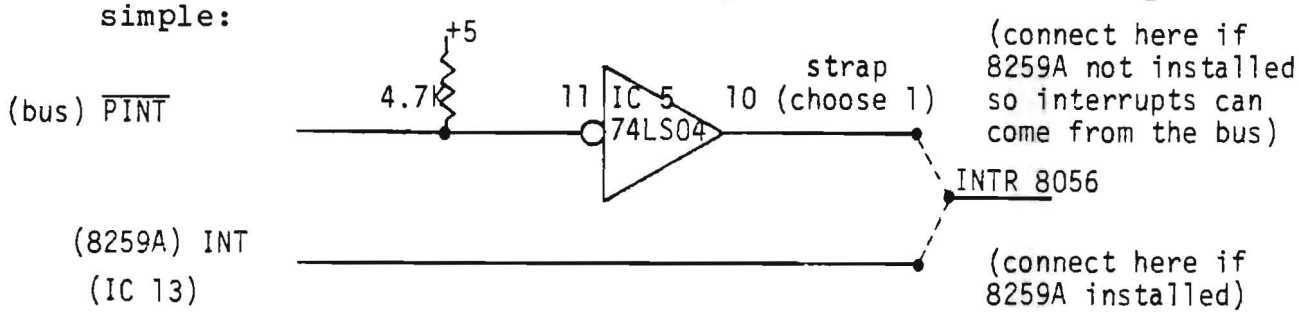
Now that the data paths are taken care of, the address path (naturally) comes next. Rather conventionally, this is:



Associated with this is the PHANTOM line circuitry. The extended address lines A16-A19 are or-ed and a true output causes the PHANTOM line (S-100 pin 67) to be pulled low (if PHANTOM jumper is in place) whenever memory is addressed above the lower 64K. This allows the use of RAM cards that only decode the A0-A15 address lines in the lower 64K without conflict above this space.

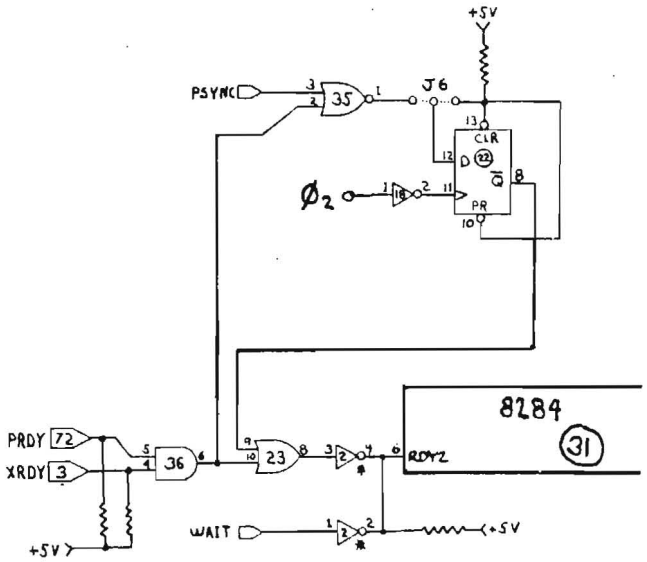
Next, the required 8086/8284 inputs INTR, RDY1, and RES will be considered. (the 8259A`s interrupt inputs are considered separately as well as the DMA request/grant sequence).

The interrupt line, PINT, can indirectly drive the INTR interrupt input of the 8086. However, if the 8259A interrupt chip is used, it must drive the INTR. To avoid noise problems (from disconnected inputs), a wired strap is used rather than an "OR" gate. The circuitry is simple:

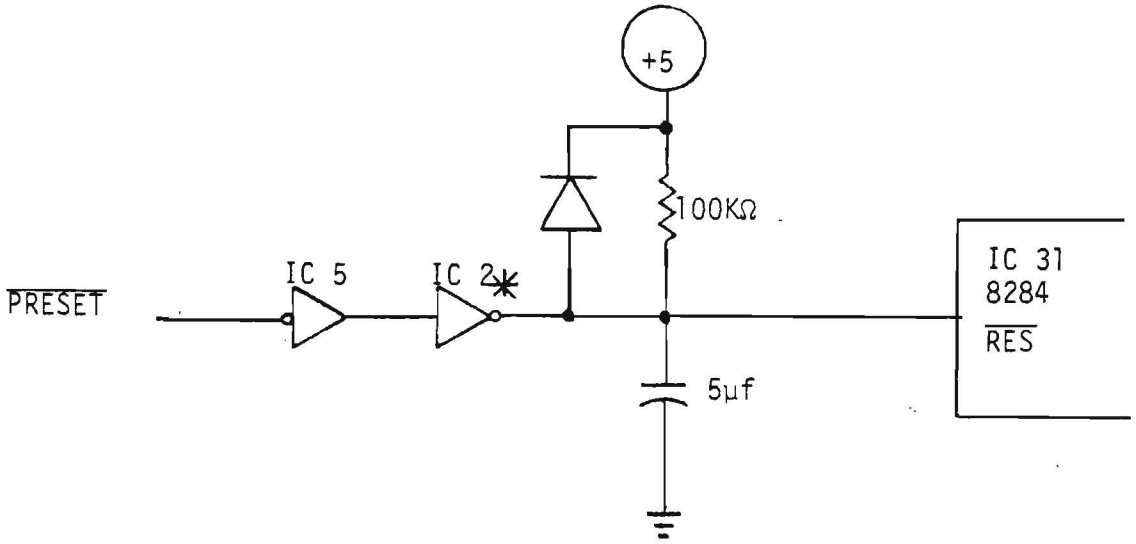


The "ready" inputs from the bus drive the 8284`s (IC 31) RDY2 input. Both lines must be "ready" for the RDY1 line to be ready. Note that the 8284 has an "OR" between RDY1 and RDY2 so both inputs cannot be used.

Added to this is the optional wait state generator. This adds one wait state to every bus cycle in addition to any requested by the device being accessed. The circuit adds a one clock duration low on the RD2 line at the end of a low caused by another device (slow memory) or right after the PSYNC pulse if no other wait state is requested. The circuit is:

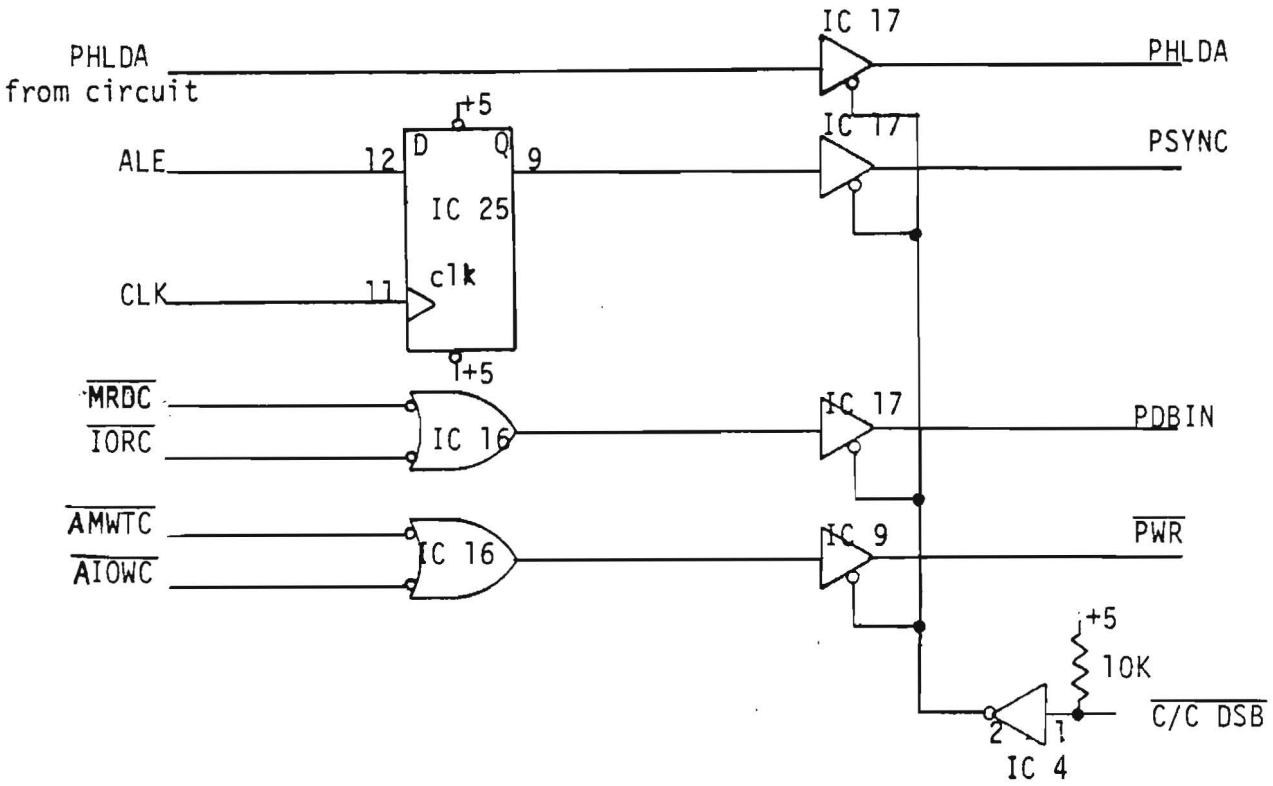


The $\overline{RES}$ input of the 8284 (I.C. 31) remains. This input should provide power-on reset, reset in the event of momentary power failure, and reset when the bus $\overline{PRESET}$ line is brought low. A Schmitt trigger is provided on the 8284, so an R-C network is adequate for power-on detection with a delay of 0.2-0.5 sec. to allow the power supply to settle. A diode is added to discharge the capacitor in the event of momentary power failure. An open-collector gate serves to discharge (and hold low) the capacitor as $\overline{PRESET}$ is brought low. The complete circuit is:



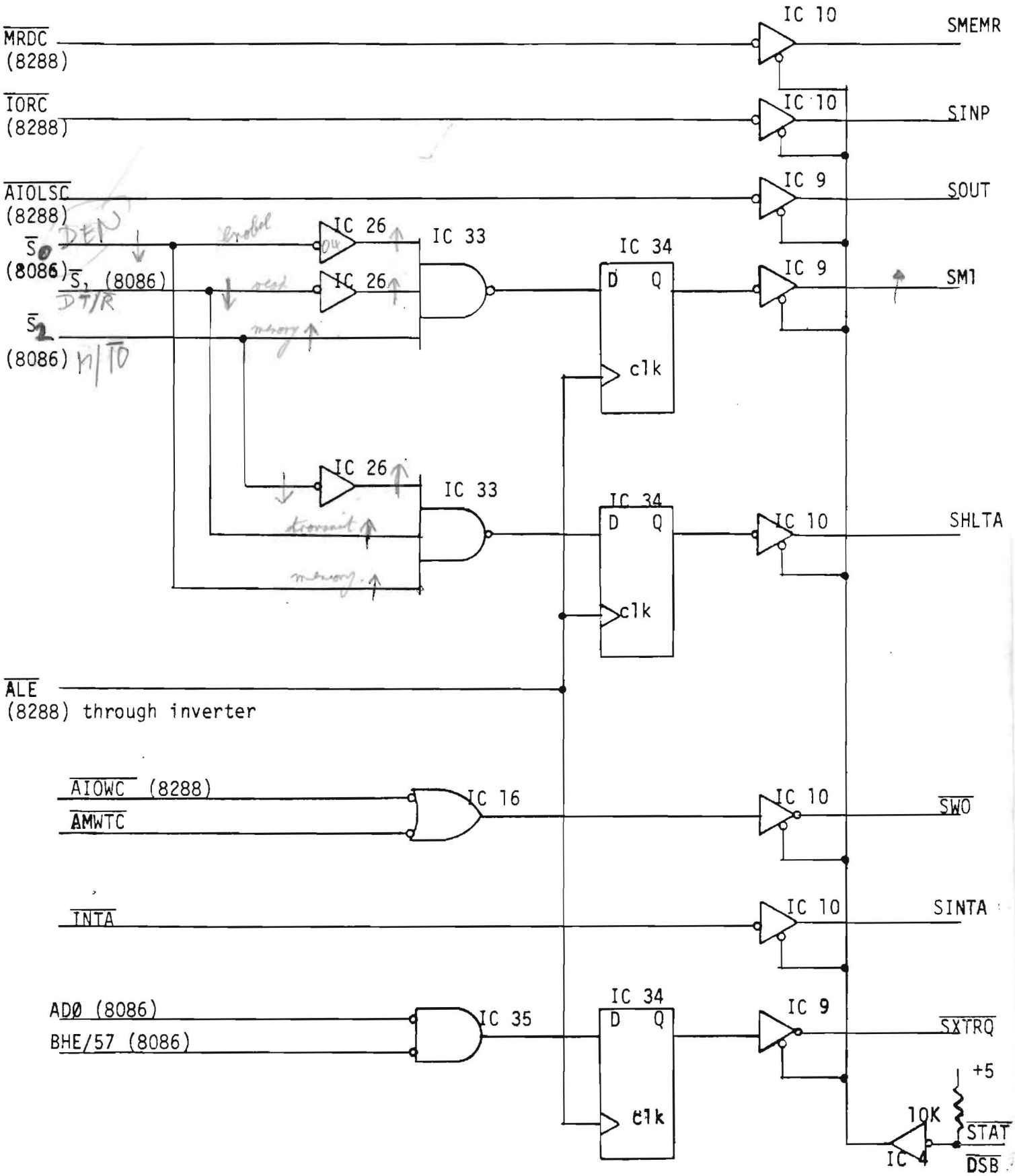
The required output signals are divided into three groups, the command/control lines (PHLDA, PSYNC, PDBIN, $\overline{PWR}$), the status lines (SMEMR, SINP, SMI, SOUT, SHLTA, $\overline{SWO}$, SINTA, and $\overline{SXTRQ}$), and the other lines (ϕ_2, CLOCK, MWRITE, $\overline{POC}$).

The command/control lines will be discussed first. The discussion of PHLDA is reserved for the DMA request/grant paragraph later. PSYNC is a positive-going pulse of one clock period duration at the beginning of a bus cycle. This is accomplished by latching ALE (from the 8282) on the rising edge of the system clock CLK (from one 8284). ALE is high for one rising CLK edge at the beginning of each cycle. PDBIN is asserted when either $\overline{MRDC}$ (8288) or $\overline{IORC}$ (8288) is asserted. $\overline{PWR}$ occurs with either $\overline{AIOWC}$ or $\overline{AMWTC}$ (8288). The schematic is shown on the next page.

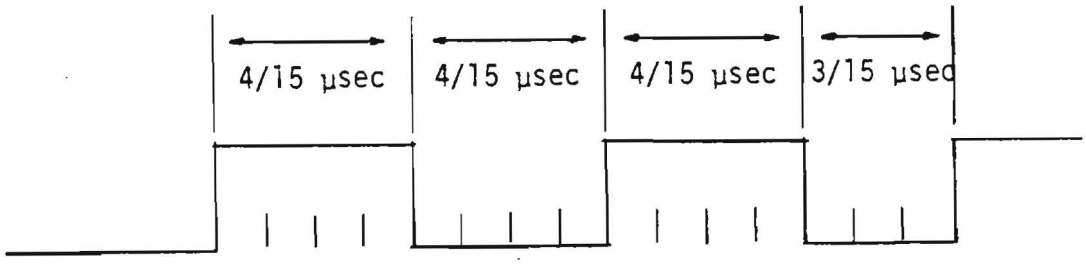


The status lines are next. SMEMR (bus) is MRDC (8288). SINP (bus) is IORC (8288). SM 1 (bus) is decoded from $\overline{S}_0$, $\overline{S}_1$, $\overline{S}_2$ (8086) (and this could not be decoded unless the 8086 were operating in "MAX" mode). Since $\overline{S}_0$, $\overline{S}_1$,and $\overline{S}_2$ are valid only at the falling edge ofALE, this output must be latched. SOUT (bus) is $\overline{\text{AIOWC}}$ (8288). SHLTA (bus) is derived from $\overline{S}_0$, $\overline{S}_1$, and $\overline{S}_2$ in much the same manner as SM1. $\overline{\text{SWO}}$ (bus) is asserted with either $\overline{\text{AIOWC}}$ (8288) or $\overline{\text{AMWTC}}$ (8288). SINTA (bus) is $\overline{\text{INTA}}$ (8288). $\overline{\text{SXTRQ}}$ (bus) is asserted for a cycle whenever both AD0 and BHE are low when ALE falls. The schematic is thus:

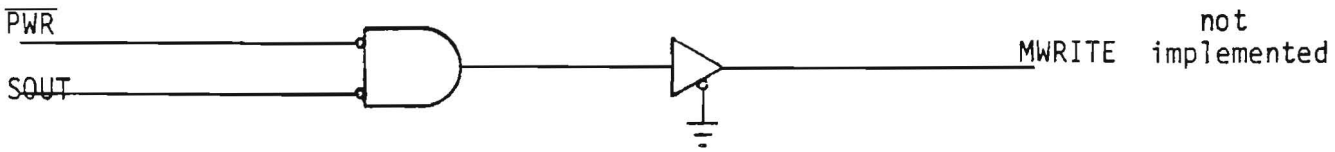
10/17 for 88



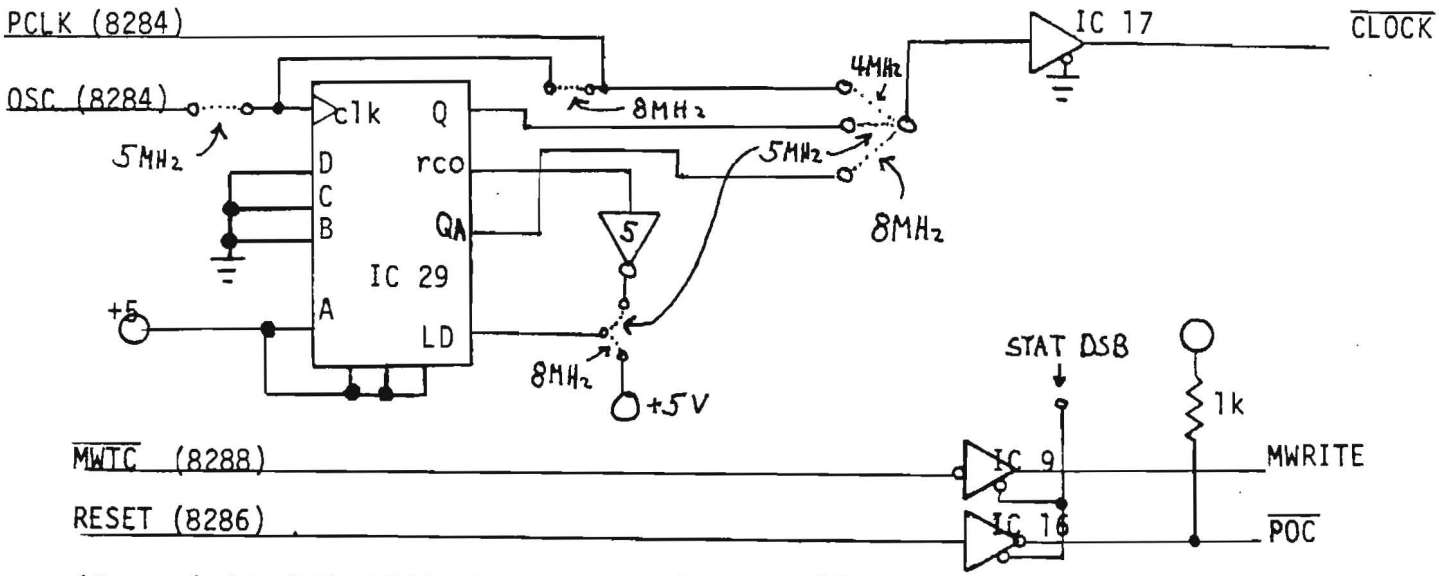
This leaves the other signals; ϕ_2, CLOCK, MWRITE, and $\overline{\text{POC}}$. ϕ_2 is the processor clock, CLK. CLOCK is required to be a 2 MHz signal of 40-60% duty cycle. When a 4MHz 8086 is used, the PCLK output of the 8284 provides a suitable signal. When 5 MHz operation is desired, the OSC output of the 8284 (at 15 MHz) must somehow be divided to give 2 MHz. Clearly it is <u>not</u> possible to produce a symmetric waveform, but dividing by 8 with a reset every 15 pulses will produce a waveform which is shaped like this.



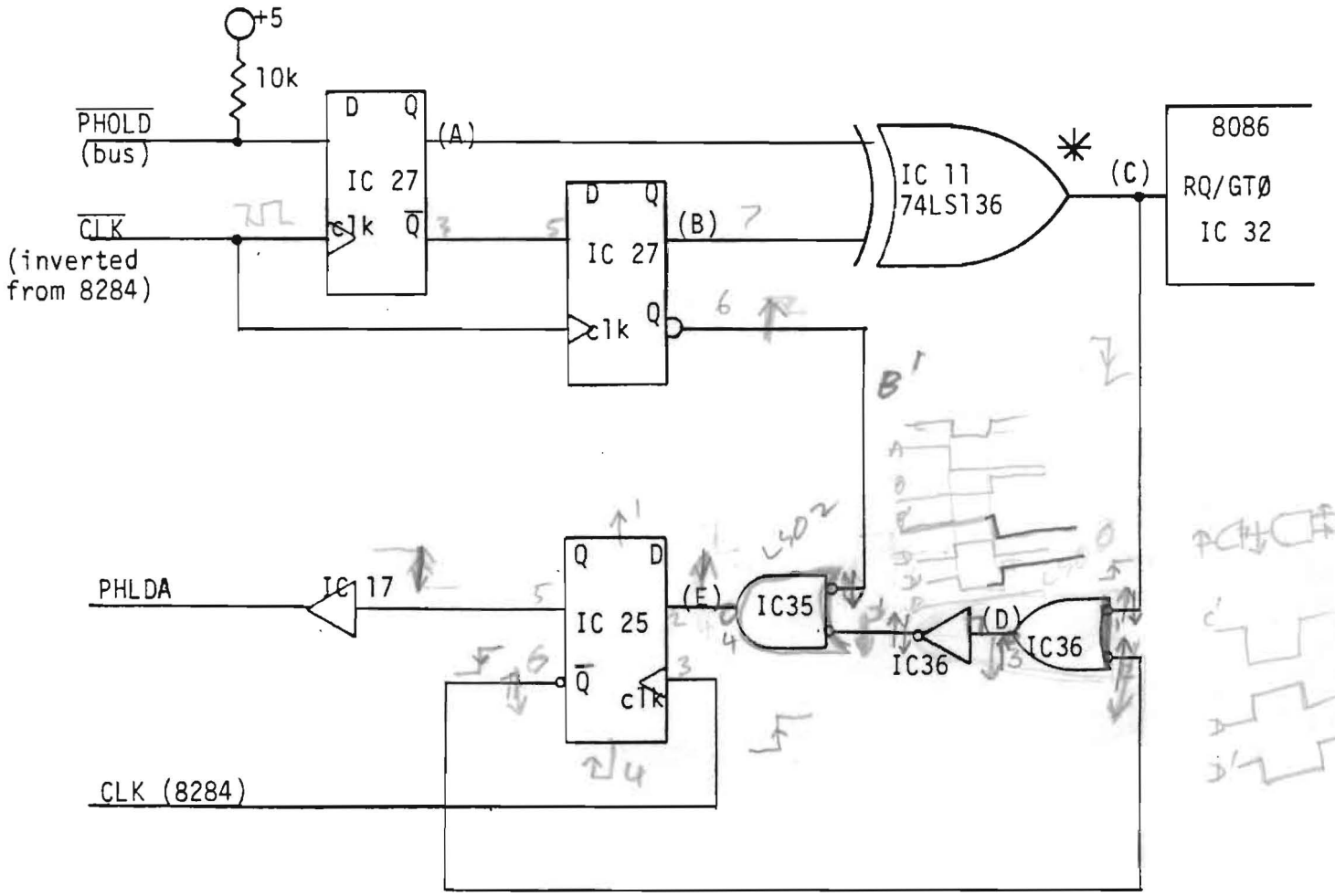
This can be done with one 74LS161 (or 163) counter as shown in the schematic . For 8MHz operation, the 74LS161 is used to divide the PCLK output by two. MWRITE poses a serious design problem. Either it can be treated in the older fashion and left always driving the bus as:



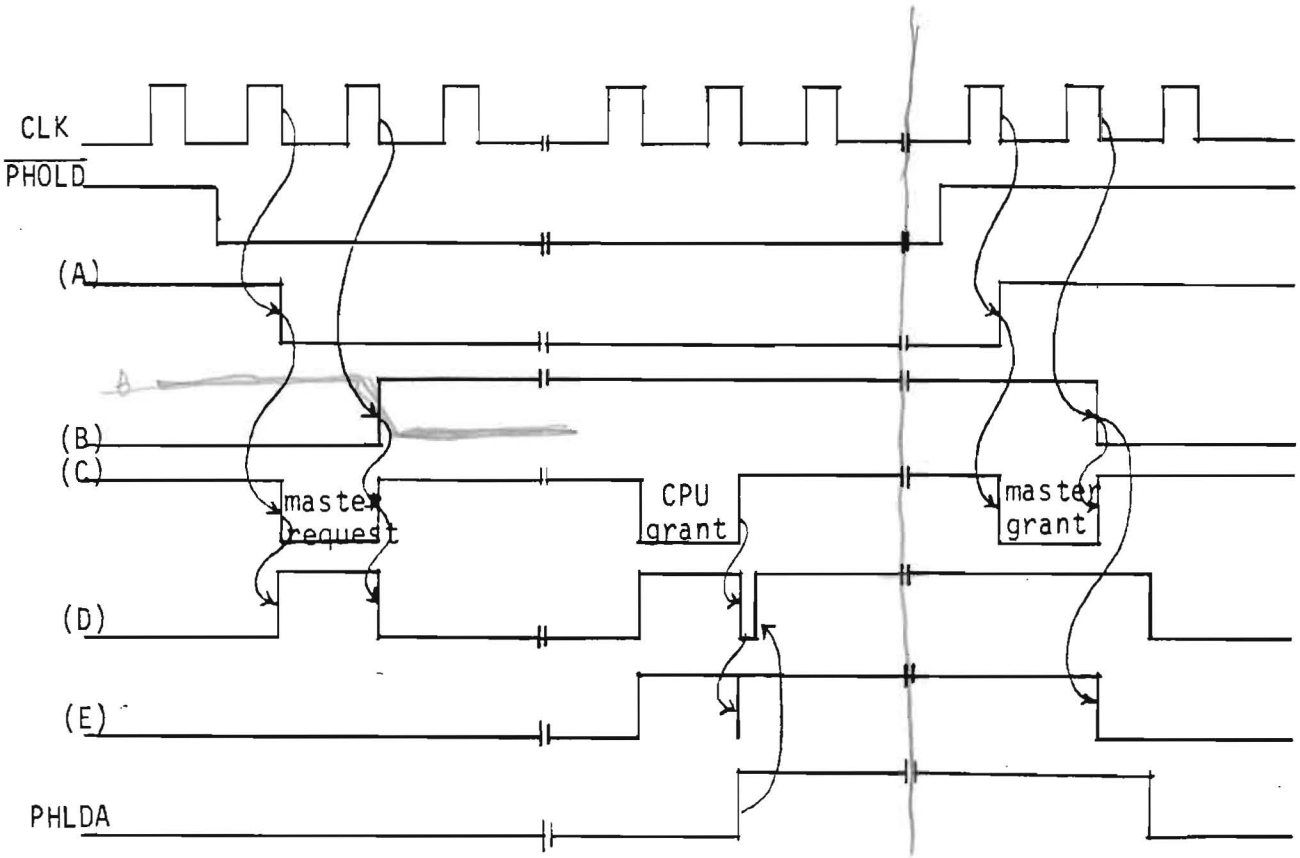
Or it can be treated as a status signal and disabled during a DMA hold sequence, as some later designs have done. If the memory (and memory-mapped I/O) designer avoids using this line, no problem will result. If no DMA devices are used the designs are equivalent. The latter design is selected for ease of implementation, and MWRITE is thus AMWTC. POC is the RESET (8284) line, which conveniently mirrors PRESET. We now have the schematic:



Next comes the DMA request/grant circuitry. It is necessary to generate a low-true pulse of one CLK cycle duration into the $\overline{RQ}/\overline{GT0}$ pin of the 8086 upon receiving a $\overline{PHOLD}$ signal. When the processor responds with a low-true pulse through the same pin, PHLDA is asserted on the bus. When $\overline{PHOLD}$ is released, another low-true pulse must be generated into the pin. This circuitry will accomplish the task:



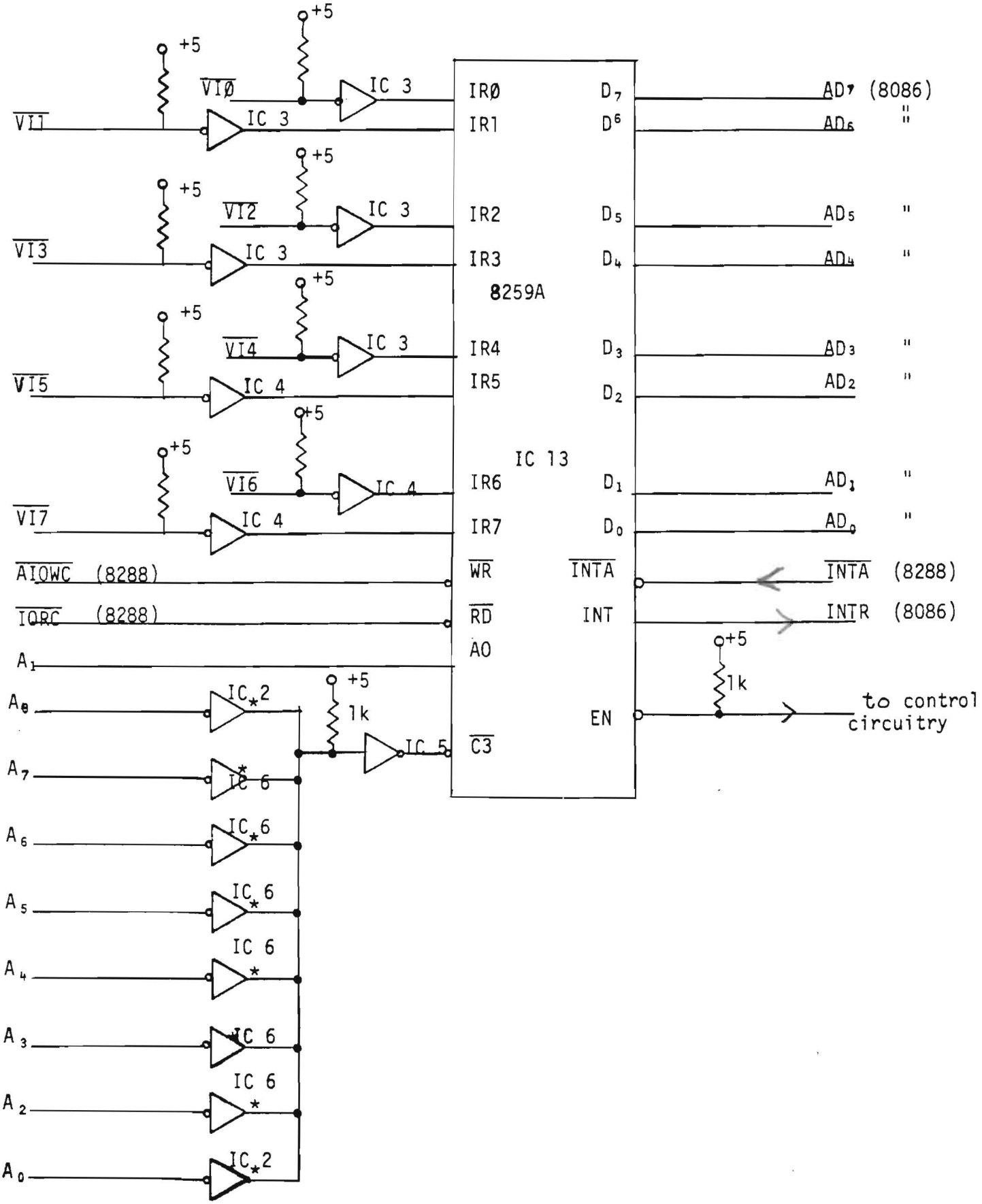
(A) is $\overline{PHOLD}$ synchronized on the falling edge of the processor clock.
(B) is (A) inverted and delayed one clock period; when high the master request pulse is over.
(C) is the $\overline{RQ}/\overline{GT0}$ pin of the 8086
(D) is high when PHLDA is asserted or $\overline{RQ}/\overline{GT0}$ is low.
(E) is high when the master request pulse is done and either the CPU grant is being pulsed on $\overline{RQ}/\overline{GT0}$ or PHLDA is already asserted.



See the relevant timing diagram on p. 5-17 of Intel's MCS-86™ User's Manual

Finally there is the optional 8259A priority interrupt controller chip and its associated circuitry. First, what happens if the chip is not present? The $\overline{EN}$ (8259A) line is pulled high by a pullup resistor to +5 volts and during the second INTA pulse the interrupt address is expected to be on the DI lines on the bus. The strap option for INTR (8086 output) was discussed earlier. The interrupt lines are derived from $\overline{VI0}$-$\overline{VI7}$ on the bus; the data lines are connected to AD_7-AD_0 on the 8086. $\overline{AIOWC}$ (8288) drives $\overline{WR}$ (8259A), $\overline{IORC}$ (8288) drives $\overline{RD}$ (8259A). $\overline{CS}$ is asserted when A0 and A2-A8 are all low (I/O ports 0 and 2). A1 (bus from 8282) drives A0 (8259A). Two consecutive even ports are chosen so that the data lines used by the 8086 will be the same for both. (These particular ports are those used by the INTEL DEMO-86 monitor ROM.) $\overline{INTA}$ (8288) drives $\overline{INTA}$ (8259A) directly. INT (8259A) drives INTR (8086) directly, as discussed before. EN (8259A) is used to disable the outputs of the multiplexers to AD_7-AD_0 as discussed under data paths. The CAS lines are left open (consigning the 8259A to operate in buffered, non-cascaded mode) because their use would require the definition of additional bus lines.

The schematic is thus:

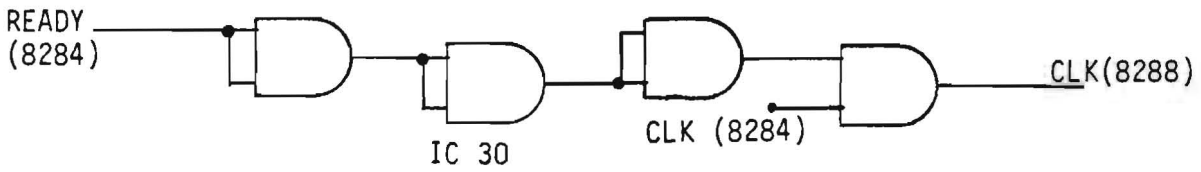


Outside of mapping a series of logic equations into SSI logic and selecting bus drivers for the control signals, the design is now complete. Remaining is a check on power consumption and design of the on-board power supply, a timing check of the circuit to make sure that there are no ridiculously long delays in sending data to or taking data from the bus, and a system check to make sure that all of the individually designed circuits will work together.

Without going into the details of gate counts and individual worst case supply currents, the board will consume at most about 1,800 ma. This means that two conventional 7805 regulators will have to be used if they are to regulate for the board. Adequate bypassing and decoupling are also necessary. Decoupling is accomplished by an 0.1 f ceramic disk capacitor across the power supply at the Vcc terminal of each package. Bypassing is inherent in the supply.

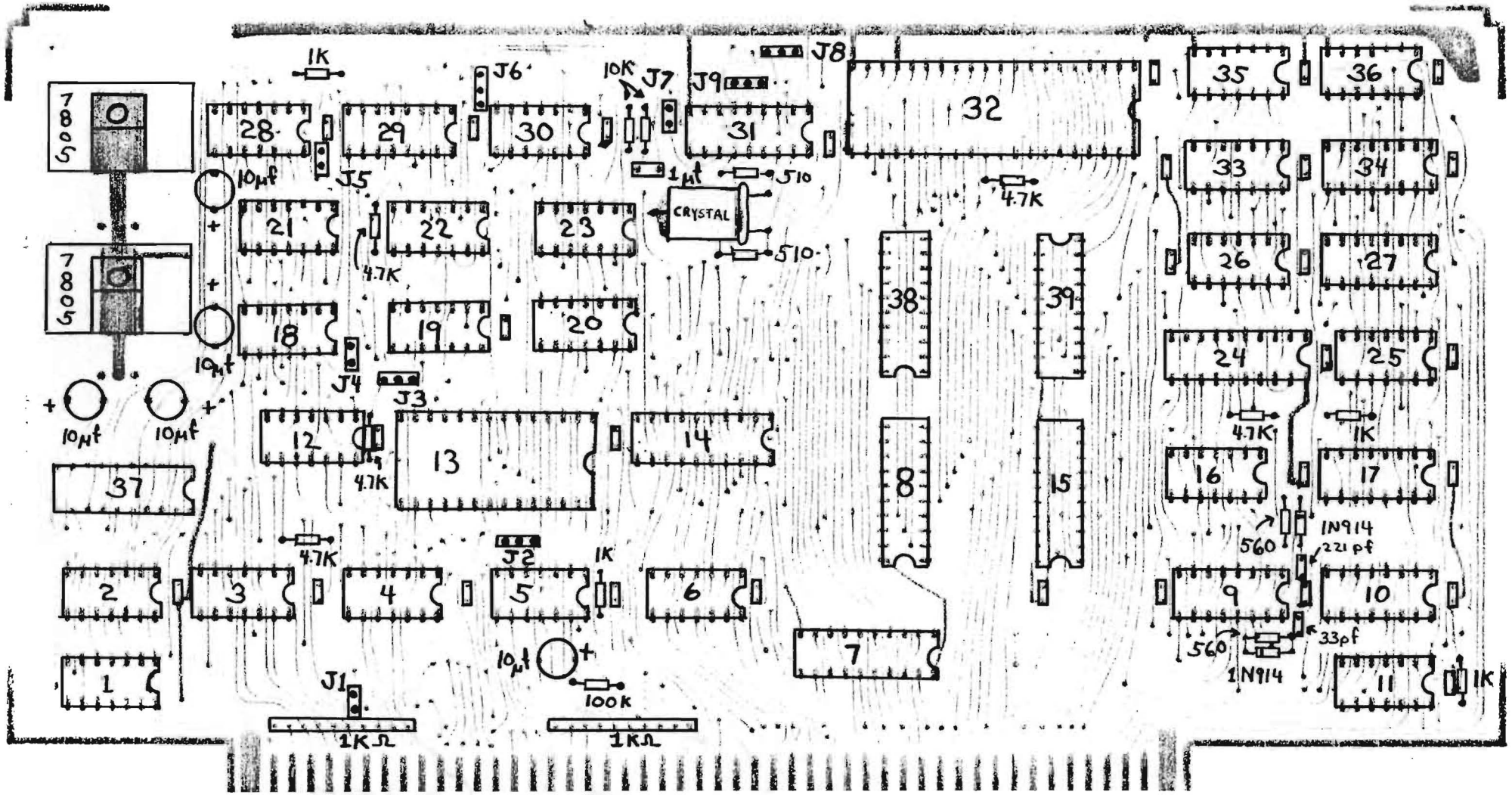
The timing check shows that at worst a 20 ns. delay is possible in read and write operations due to bus drivers on the card. This must be accounted for in specifying memory and I/O speed. (This assumes all 74LS parts).

The system check turns up an anomaly in the use of the 8288 bus controller in systems having "wait states". The 8288 bus controller cannot detect the occurence of a not ready condition (wait state) and will proceed as if said condition had not occurred. Since the 8288 is a static device, the clock into it can be gated to solve this problem as follows:



The gate delays are necessary to prevent glitching on the 8288's CLK input. Experimentally this circuit offers the largest margin against varying gate delays in either direction so that aging and temperature will have a minimum effect on its operation.

Electrical compliance with the S-100 standard is accomplished by the use of 74367 and 74368 bus drivers for the control signals and by making pullup resistors on the open collector bus lines 5-10KΩ to be well within the current limits. The 8282 and 74LS245 far exceed the drive capability requirement.



8086 CPU BOARD COMPONENT PLACEMENT

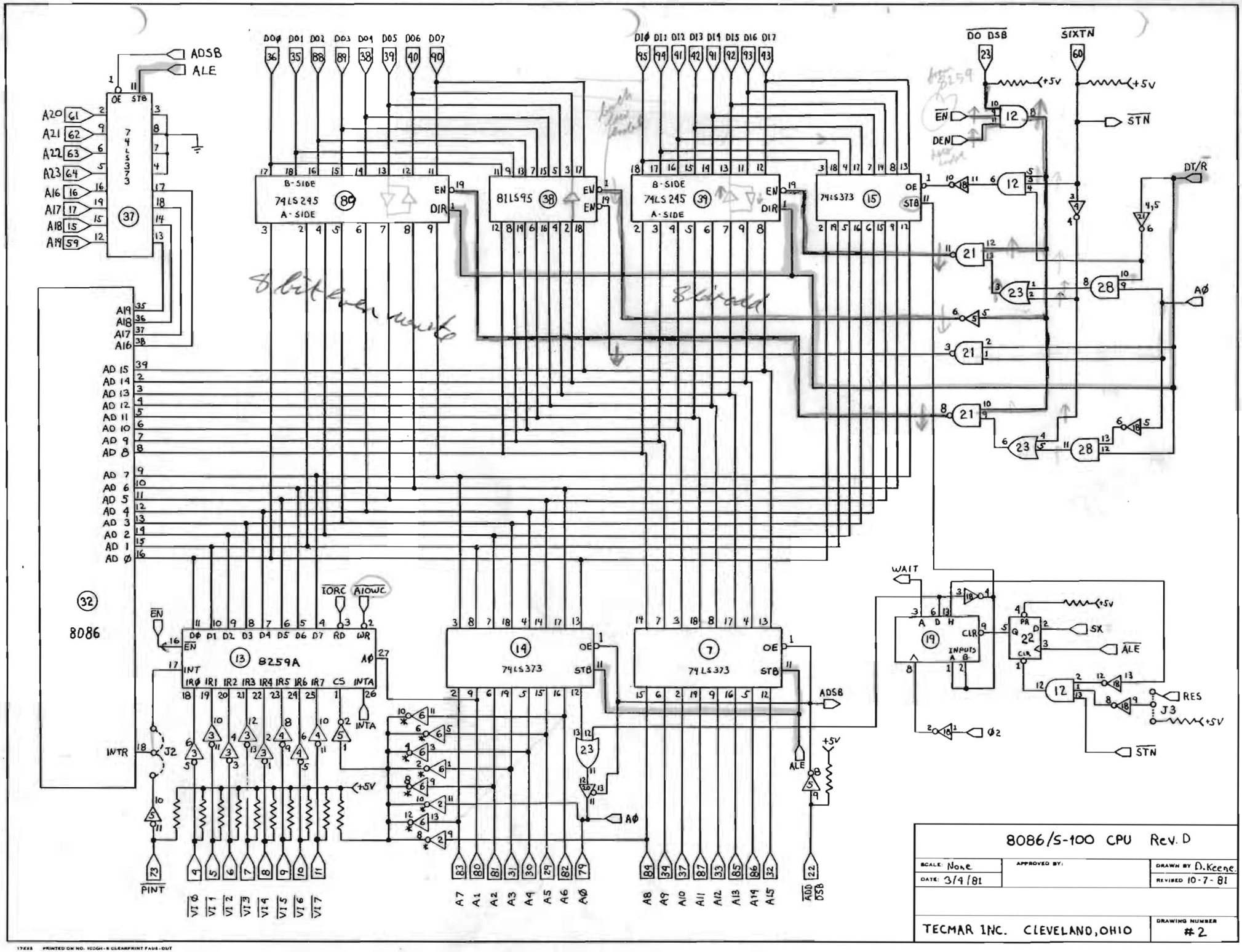
NOTE: 1) ▷ INDICATES SIGNALS THAT ARE CONNECTED TO COMPONENTS WHICH ARE SHOWN ON PART 2 OF THIS DIAGRAM (SHEET #2)

2) ENCIRCLED #'S INDICATE CHIP #. IN GATES, INVERTERS AND BUFFERS THE CIRCLE IS NOT USED, TO IDENTIFY CHIP TYPE - SEE LIST

3) ONLY PART OF THE CPU CHIP IS SHOWN HERE - THE REST OF IT IS DESCRIBED ON SHEET 2 OF THIS BOARD'S DIAGRAM.

IC #	TYPE
9,10	74368
11	74 LS 136
16,36	74 LS 00
17	74 367
24	8288
25	74 LS 74
26	74 LS 04
27,34	74 LS 175
29	74 LS 161
30	74 LS 08
31	8284
32	8086
33	74 LS 10
35	74 LS 02

8086/S-100 CPU Rev. D			
SCALE: NONE	APPROVED BY:	DRAWN BY Rafi Levy	
DATE: May, 17, 1979		REVISED 10-7-81	
TECMAR INC. Cleveland, Ohio		DRAWING NUMBER Sheet #1	


8086/S-100 CPU Rev. D
SCALE: None
DATE: 3/9/81
APPROVED BY:
DRAWN BY D. Keene
REVISED 10-7-81
TECMAR INC. CLEVELAND, OHIO
DRAWING NUMBER #2
8086
8259A
74LS373
74LS245
8T595
74LS374
ADSB
ALE
INTR
INTA
WAIT
RES
STN
SX
+5V

PARTS LIST

8086/S-100 BUS

I.C.#	PART #	QUANTITY	COMPONENTS PART TYPE AND NUMBER
1	74LS32	1	1N915 DIODE
2	74LS05	1	100KΩ RESISTOR
3	74LS04	3	1KΩ RESISTORS
4	74LS04	6	4.7KΩ RESISTORS
5	74LS04	2	4.7KΩ RESISTOR NETWORKS
6	74LS05		BOURNS#4310R-101-472
7	8282	29	0.1µf CERAMIC DECOUPLING
8	74LS245		CAPACITORS
9	74368	5	18µf TANTALUM CAPACITORS
10	74368		15VWDC
11	74LS136	1	CRYSTAL
12	74LS11		12.0 MHz. for 4 MHz. CPU
13	8259A		15.0 MHZ. for 5 MHz. CPU
14	8282		24.0 MHz. for 8 MHz. CPU
15	8282	2	7805 VOLTAGE REGULATORS
16	74LS00	2	HEAT SINKS
17	8T97 or 74367		THERMALLOY#6107-14
18	74LS04		
19	74164		
20	74125	7	JUMPER CONNECTORS
21	74LS00		
22	74LS74		
23	74LS32		
24	8288		
25	74LS74		
26	74LS04		
27	74LS175		
28	74LS08		
29	74LS161 (for 5 MHz. or 8 MHz.)		
30	74LS08		
31	8284		
32	8086		
33	74LS10		
34	74LS175		
35	74LS02		
36	74LS00		
37	74LS173		
38	81LS95		
39	74LS245		

SET-UP

8086/S-100 CPU - INTERRUPT

Boards built and tested by Tecmar Inc. will have all jumpers in place for the processor speed purchased. The options will be set as follows:

1. 16 bit operation only
2. No wait state (except for 8 MHz boards which will have the wait state option selected
3. $\overline{\text{PHANTOM}}$ line driver not connected
4. 8259A vectored interrupt selected

No set-up is needed for these boards. If one desires to change this set-up or if the board was not built and tested by Tecmar, see the Jumper Summary that follows. If changing the processor speed, remember to change the crystal also; 12.000MHz crystal for 4MHz operation, 15.000 MHz for 5MHz or 24.000MHz for 8MHz. Also note that an 8086 chip will run at slower speeds than it is rated but not at faster speeds; i.e. don`t expect to run an 8086-4 (4MHz version) at 8MHz.

JUMPER SUMMARY

8086/S-100 CPU - INTERRUPT

There are nine jumper locations on the board that are used for setting the operating speed and selecting the optional features on the board. All illustrations of jumper placement are in the same orientation as on the parts placement diagram. See the diagram for position of jumpers.

NOTE:  means place the small jumper connector so that it connects (covers) the two jumper pins that are connected by the , the third pin will have nothing on it.

J1: $\overline{\text{PHANTOM}}$ jumper. Put a jumper here to cause the $\overline{\text{PHANTOM}}$ line (S-100 pin 67) to be pulled low whenever the CPU addresses memory above the lower 64K. This allows an older memory card with a 16-bit address to be used in the lower 64K (as long as the memoy cards also have a $\overline{\text{PHANTOM}}$ line connection). If there is no jumper here, the CPU card has no affect on $\overline{\text{PHANTOM}}$.

J2: Selection of 8259A or PINT interrupt operation.
A)  For operation with 8259A vectored interrupt chip
B)  For operation of interrupts with the PINT line(S-100 pin 73)

J3: Selection of the 16-bit transfers from 8-bit memory option.
A)  Jumper this way if all of system memory is 16 bits wide
B)  Jumper this way if any of system memory is 8 bit only memory

J6: Selection of optional wait state.
A) With jumper this way the CPU board will insert a wait state, in addition to any wait state requested by the device being accessed, in every bus cycle. This is needed for 8 MHz operation or for use with very slow memory.
B)  With jumper this way no extra wait states will be added.

All other jumpers are for CPU speed selection and should not be changed unless the crystal is changed also (see system set-up).

J4: This jumper is in for 5 or 4 MHz operation, out for 8 MHz.

J5: This jumper is in for 8 MHz operation, out for 5 or 4 MHz.

J7: This jumper is in for 5 MHz operation, out for 4 or 8 MHz.

J8:  For 8 MHz operation
 For 4 MHz operation
No jumper for 5 MHz operation

J9:  For 5 MHz operation
 For 8 MHz operation
No jumper for 4 MHz operation

TIMING REQUIREMENTS

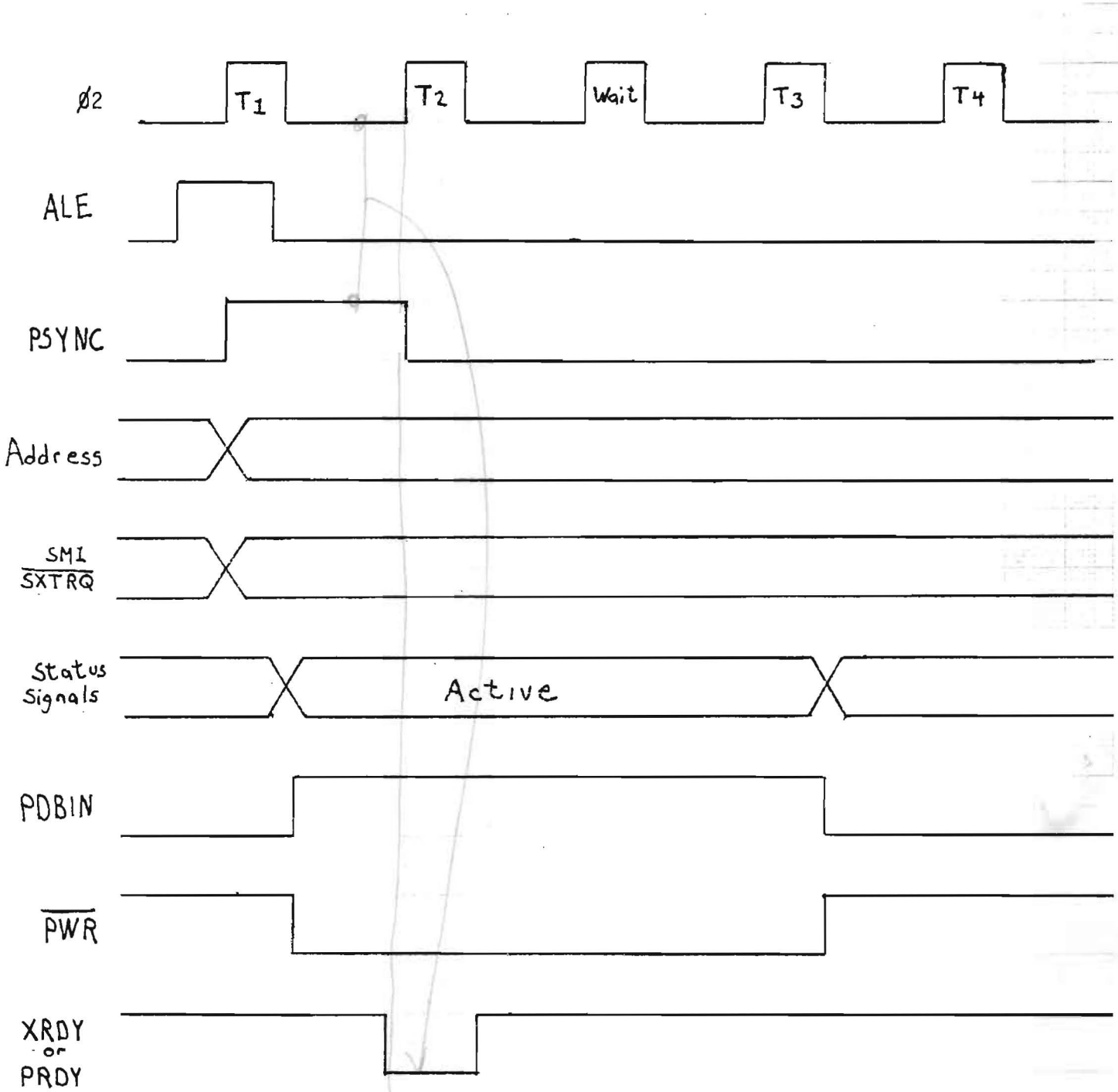
8086/S-100 CPU - INTERRUPT

Parameter	4MHz	5MHz	8MHz
read pulse	500ns	400ns	250ns
address to read puls	80ns	65ns	40ns
write pulse width	500ns	400ns	250ns
address to write pulse	80ns	65ns	40ns
worst-case minimum access time from read strobe	480ns	380ns	230ns
worst-case write cycle time	1000ns	800ns	500ns

These timing requirements refer to memory boards not memory chips. The bus driver delays on the board will require chips to have access times that are around 50ns. faster than the board timing requirement.

These read and write times also apply to I/O devices -- note that standard 8251A requires a read pulse of 250ns and a write pulse of 250ns with data setup of 150ns, so a write state is required on write operations. The standard 8255A requires a read pulse width of 300ns and a write pulse width of 400ns with a 100ns data setup time and 30ns hold time, so it requires a wait state also.

8086 CPU BUS TIMING WITH ONE WAIT STATE



CRITICAL TIMING SIGNALS

8086/S-100 CPU - INTERRUPT

Please observe that all signals are necessary to the oeration of the board; there are very few failures that will not "crash the system".

CLK processor clock - without this nothing will work (4, 5, or 8 MHz) also check this at pin 2 of the 8288 and look for glitches.

data path gating signals:

		I.C.	Pin #	
DI bus:	$\overline{OE}$	39	19	
	DIR	39	1	
	$\overline{EN}$	38	9	
	STB	38	11	
DO bus:	$\overline{OE}$	8	19	
	DIR	8	1	
	$\overline{EN1}$	15	1	
	$\overline{EN2}$	15	19	
Address bus:	$\overline{OE}$	7	9	
		14	9	
		37	1,2	
	ALE	7	11	
		14	11	
		37	7	(ALE inverted)
		24	3	

from bus: $\overline{SIXTN}$ (absence of this signal on 16-bit operations to 16-bit memory will cause the operation to fail).

Proper 8086 operation can be diagnosed by examining the outputs of the 8288 controller chip while single-stepping the processor and comparing them to the operations implied by the software.

FOR EXAMPLE SEE NEXT SECTION.

DIAGNOSTIC AIDS

8086/S-100 CPU - INTERRUPT

The 8086 board set is fully static in that it can be single-cycled from the bus. Note however that only memory or I/O cycles may be extended to yield useful information on the bus. If a known program is being executed (i.e. a ROM monitor or bootstrap upon startup) the sequence of instruction fetches of data accesses as well as I/O operations may be observed with the use of:

1) A device to drive the XRDY (pin 3) line low from the time that PSYNC is high and $\emptyset_2$ (8086 clock) is low until a (debounced) pushbutton is depressed. It should be possible to disable this circuit by means of a double-throw toggle switch. These are the single cycle and run/stop switches respectively.

2) Some means for displaying the states of all the bus lines. If a lot of time is available, a logic probe will do. Otherwise, you will need a display panel which captures <u>every</u> line used on the bus.

3) A logic probe or clip-on display to show the logic states of all 37 signal lines on the 8086 chip itself, and on any other package desired.

The technique consists of resetting the processor while the run/stop switch is at "stop", and then examining the bus data, address, status, and control information for incorrect signals. If an incorrect signal is found, it should be traced back to the I.C. it originates at (for address signals, the 8282; data signals through the 74LS245's or the other two data buffers).

Bus cycles should be repeatedly examined until the program appears to execute correctly, having performed:

1) 16-bit memory read
2) 8-bit memory read from an odd location (A0=1)
3) 8-bit memory read from an even location (A0=0)
4) 16-bit memory write
5) 8-bit memory write to an odd location (A0=1)
6) 8-bit memory write to an even location (A0=0)
7) I/O input operation
8) I/O output operation
9) 16-bit read from 8-bit memory (if 8-bit memory is present)
10) 16-bit write to 8-bit memory

If all those operations appear successful, the problem is probably either

1) bus noise (are you using an active terminator card? Is the bus adequately shielded?)
2) bus signal risetime problem (do all your boards conform to the electrical portion of the proposed S-100 specification?)
3) bus contention (is another device dynamically driving the bus due to faulty address decoding logic or control logic?

In single-stepping a program, do not forget that the 8086 pre-fetches instructions. It is, therefore, necessary to keep track of where it should be going and not decide that you have a defective processor chip when instructions are fetched beyond a jump which should have been taken.

Here is an example taken from the first few instructions of the INTEL Demo-86 monitor:

```
FFFF0   JMP    FF60:00BC          EA   BC00   60FF
FF6BC   CLI                       FA
FF6BD   MOV    SS,CS:00B8         2E   8E  16  00B8
FF6C2   MOV    SP,07C0            BC   07C0
          .
          .
          .
FF6B8   1000
```

```
First cycle:
  FFFF0   DI=BC  DO=EA  SMEMR=1,SXTRQ=0,  SM1=1,  PDBIN=1
Second cycle:
  FFFF2   DI=60  DO=00    "       "         "        "
Third cycle:
  FFFF4   DI=XX  DO=FF    "       "         "        "
insert possible additional fetch cycle at FFFF6
Next cycle:
  FF6BC   DI=2E  DO=FA  SMEMR=1,SXTRQ=0,  SM1=1,  PDBIN=1
```

And so on.

Please note that in order to implement this debugging technique, an intimate knowledge of the hardware and a listing of the software, a well as special hardware is required. Conventional 8080 front panels will not work and may damage the 8086 board set if their use is attempted.

In the example given above, assume that on a memory read operation the correct information is found on the bus, but the processor acts as if it had received incorrect data (i.e. jumps into nonexistent memory). In this cae, reset the processor again and compare the data signals at the processor pins AD15 to AD0 with the apropriate DI_7 to DI_0 and DO_7 to DO_0 signals respectively.

If they are not the same at, say, bit 3 (i.e. DO_3 is not the same as AD_3), trace the DO_3 signal into pin 15 of IC8 (74LS245), verify that it is the same at pin 5 and at pin 13 of the 8086. The location of the failure must then be in the node which first has an incorrect signal (such as a shorted of broken trace) or in the device driving that node (i.e. bad IC) or in a device receiving data from that node (i.e. bad IC shorting input to +5 or ground).

1. Due to the complex nature of the 8086 addressing scheme it is essentially infeasible to implement a front panel in the manner done on the IMSAI and ALTAIR machines.

   The reset, run/stop, and single-step features of those front panels as well as a bus display (the address, data, and status lights) can be implemented in a manner similar to those front panels with the addition of status display lights on $\overline{\text{SXTRQ}}$ and $\overline{\text{SIXTN}}$.

   Altering memory could be accomplished through the use of a DMA scheme, but starting program execution at a particular point would be completely infeasible from a front-panel-like device.

   Therefore, it is recommended that the machine be set up with a terminal-oriented monitor. It has a power-on jump feature (to fixed address FFFF0H) and a simple monitor is available from INTEL in a pair of 2616 PROMs.

   A front panel effect could also be achieved by storing in ROM a program to treat a set of lights and switches as I/O devices and emulate a front panel, but that implementation sacrifices the ability to single-step a program.

2. The 8086 system for the S-100 bus meets all of the requirements for a bus master outlined in the proposed standard published in the March 1979 IEEE Computer magazine, except that it is faster.

   It uses no lines that are specified as undefined or reserved for future use. It also does not use the STVAL (status valid) line which was not well defined in the proposed standard and is not even needed since the status lines are valid immediately following the PSYC pulse.

   It drives the following lines which are not considered type "M" (master) signals:

ϕ_2	24	
CLOCK	49	
MWRITE	68	- disabled by STAT DSB low
$\overline{\text{POC}}$	99	

The CPU card treats $\overline{\text{PRESET}}$ (75) as an open-collector input (or momentary normally-open switch to ground) and emits a low on the $\overline{\text{POC}}$ line when that input is asserted exactly as if the power had just been turned on.

The $\overline{\text{PINT}}$ line is ignored by the CPU if the vectored interrupt option is instaled on the CPU card. (See jumper summary).

Note that the response to the $\overline{\text{INTA}}$ signal by the interrupt controller or interupting device for the 8086 is ENTIRELY different from the response on an 8080 system. (The 8086 requires that a one-byte interrupt type be put on the bus, whereas the 8080 requires than an instruction be put onto the bus.) For this reason the use of the vectored interrupt option on the CPU card is recommended whenever interrupts are to be used.

The system presently runs on a $\emptyset_2$ clock of 5 MHz as the standard. The system can also be used with 4 or 8 MHz 8086`s, but at 8MHz this is faster than the S-100 standard indicates. To run at 8 MHz the wait state option will have to be used as there will probably not be very many devices in the system that can run that fast. The wait state option will slow bus operations down to a reasonable speed while allowing internal 8086 operations to proceed at 8 MHz. Devices which use the $\emptyset_2$ clock as a timing base for external operations assuming that it is 2MHz should be modified to use the CLOCK (49) signal which is 2MHz. Note: in 5 MHz systems the CLOCK signal will not be symmetrical.

Note that unlike the 8080, the 8086 does <u>not</u> duplicate the 8-bit I/O address on A15-A8 and A7-A0, but rather uses a full 16-bit address for I/O devices. I/O mapped boards accessing lines A15 through A8 as substitutes for A7 through A0 will need to be addressed differently in software as opposed to the way they were on the 8080. The extended address bits are all zero for an I/O operation.

