KURZ-KASCH, INC., ELECTRONICS DIVISION

Presents

SIGNATURE ANALYSIS FOR MICRO-PROCESSORS

---

by: J.W. Sneed, Jr.

Kurz-Kasch, Inc.
Electronics Division
P.O. Box 1246
Dayton, OH 45401
(513) 299-0990

SIGNATURE ANALYSIS FOR MICROPROCESSORS

I

SIGNATURE ANALYSIS

A means of isolating digital logic faults at the components level. Although considered most useful in servicing micro-processor based products, the technique is applicable to all digital systems. Basically, the technique involves the tracing of signals and the conversion of lengthy bit streams into four-digit hexidecimal "signatures". Using logic diagrams and schematics specially annotated with correct signatures at each node and guided by troubleshooting trees, the technician traces back until he finds a point in the circuit which has a correct input signature and incorrect output signature.

NODE

A point of convergence on a diagram, chart or graph. Nodes can be used to designate a state, event, time convergence, or a coincidence of paths or flows.

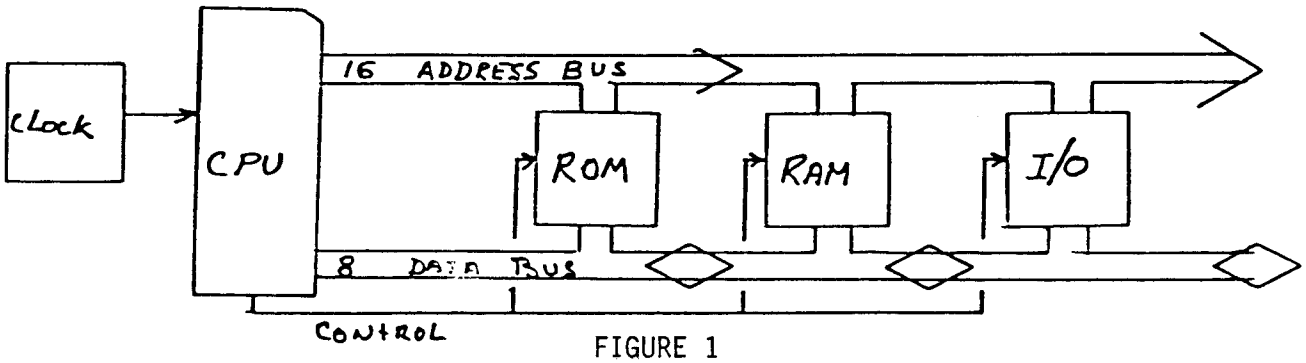


FIGURE 1

Here is a block diagram of a common bus oriented processor. The clock is the heart of the system and is the timing for the entire processor.

A CPU accepts timing information from the clock, generates addressing information, controls functions of various devices in the system, accepts information from memory and/or I. O. devices. This information or data tells the CPU what to do, it does what it is instructed and generates a response to the instructions.

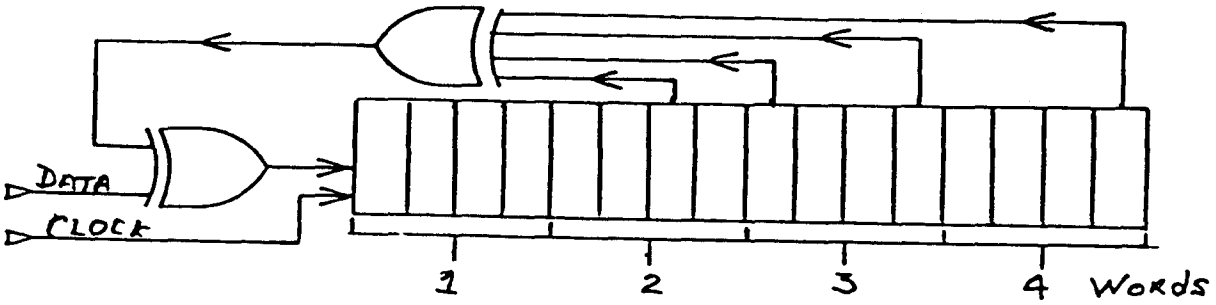
ROM is the permanent memory wherein resides the basic instructions for the CPU. Without ROM the CPU can do nothing. (ROM here is defined as CPU instructions).

RAM is the temporary storage mediam where information is temporarily stored until acted upon by the processor and moved to its final destination. RAMS also hold partially processed data.

I. O. (input-output) devices are the interfaces between the computor system and the outside world.

Throughout the entire processor system, past each node, there are streams of data. These data streams have two forms or characteristics which make them unique-quantity of "0's" & "1's" and the placement in time of them. Polarity of the pulse is important but its placement in time is equally important. The most practical and most accurate method of determining if the time and polarity criteria is met is with Signature Analysis.

Signature Analysis is a technique based on data compression to provide a unique fingerprint of each interconnection or test node in the unit under test (UUT) The Signature II provides the technician with a test probe that can be used to enter data to be recorded and read out for the test node.



Signature Analyzers convert serial bit streams into a 4 hexdigit "Signature "

FIGURE 2

Since a unique signature is generated for each data stream a prime requirement is that the data stream passing a node be indentical (unless faulty) for the same node on each identical board. The second requirement is that this data stream repeat itself. This repetition is assured by having a start and stop pulse which is time and polarity related to the data stream.

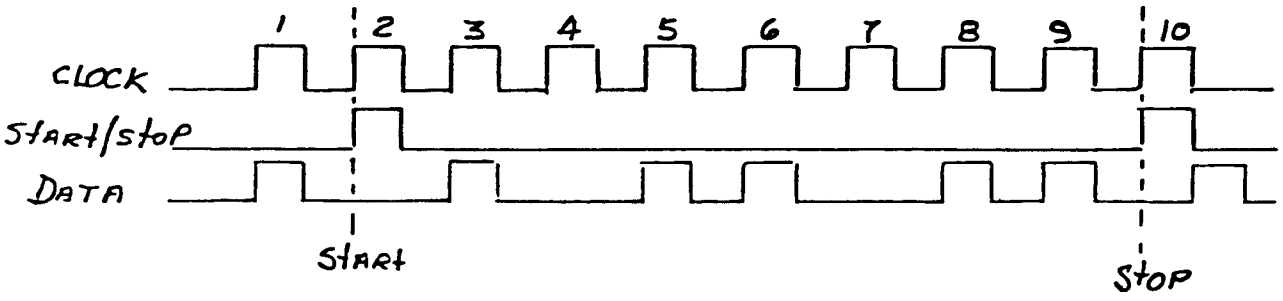


FIGURE 3

If we start the Signature Analyzer by placing "Start" on the rising edge (⎍⎺) of the first pulse on line 2 above and the "Stop" on the rising edge (⎺⎍) of the 2nd pulse labeled "Stop" we will allow clock pulses 2 thru 9 to enter the system. There is one data bit entered into the compression circuit for each clock pulse.

In figure 3, the data stream entered would be: 01011011

This stream would be entered each time the window is generated.

To demonstrate how a single bit difference between two data streams affect the "Signature" consider these two streams: 1111111110000011111 = D953

1111111110000001111 = 99F6

Either a bit difference or displacement by time (clock pulses) gives a totally different signature - not one digit or segment difference.

Certain portions of a processor system not designed for Signature Analysis can cause some problems but by and large following a few simple rules will overcome these problems.

Let us take a processor system from the beginning and follow through to develop signatures.

II

ALL VOLTAGES

For any electronic system to operate properly the required voltages must be present and within tolerances.

Tolerances for voltage regulators commonly used are usually ±5%. This will allow a 5 volt regulator to be between 4.75 and 5.25 volts. A 12 volt regulator can vary between 11.4 and 12.6 volts.

A regulated voltage less than the allowed minimum could be caused by:

A. Bad diodes
B. Bad filter
C. Overloaded regulator
D. Bad regulator

in that order.

Of course a power supply not plugged in, turned on, or one with blown fuses does not have voltages within tolerance. Usually the voltage is extremely low--say 0 volts. Don't laugh, it has happened to you--or will some day.

III

SOME COMMON FAULT SIGNATURES & HOW TO DEAL WITH THEM

Figure 1

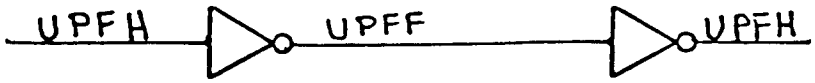


Figure 1 shows a normal inverter chain with signatures you might expect.

Figure 2

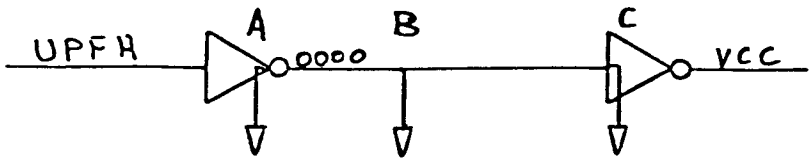


Figure 2 we see a proper signature on the first inverter's input and a "stuck at 0" at the output. As shown at points A, B & C we can have 3 failure modes. Point A is a failed output on the inverter, Point B would be a physical short to ground and C, a failed inverter input.

To determine which failure mode we have:

1--Connect Signature II Pod ground-board ground.
2--Connect HL-480 power leads to board (+5 & Ground)
3--Put Signature II Probe to inverter output.
4--Place HL-480 tip to inverter output.
5--Activate pulser, either 1 shot or 5 HZ mode.
6--If the Signature II Probe pulse light pulses the fault is at point A or C. To isolate the fault further, lift or clip the pin at the output of the inverter and if the output gives the proper signature, UPFF, then the input of the second inverter is bad. If you still have 0000 then the first inverter is bad.
7--If in 6 above the pulse light does not pulse then you have a physical short.

Figure 3

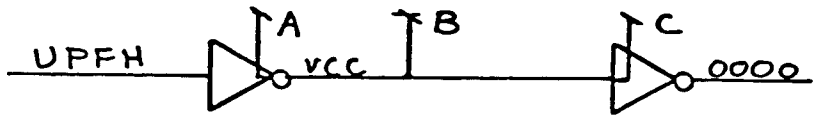


Figure 3 shows a gate or line with a failure mode "stuck at 1". This is fault isolated the same for a "stuck at 0".

Figure 4

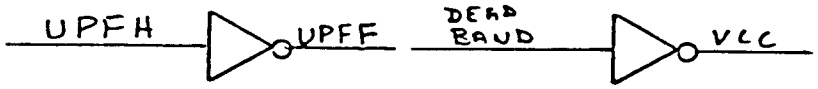


Figure 4 shows the signatures (or probes) indications for a broken trace.

Figure 5

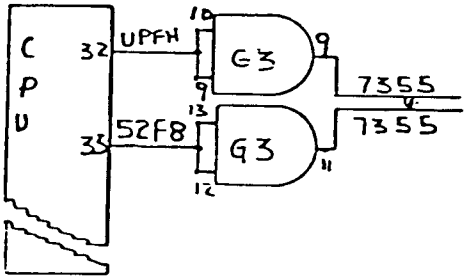


Figure 5 shows address lines from a CPU to buffers (G3) to the address bus. Proper signatures at the CPU and input of buffers but incorrect at buffer output is a fault mode. Where you find 2 or more incorrect but some signatures this indicated two bus lines shorted togeter.

Figure 6

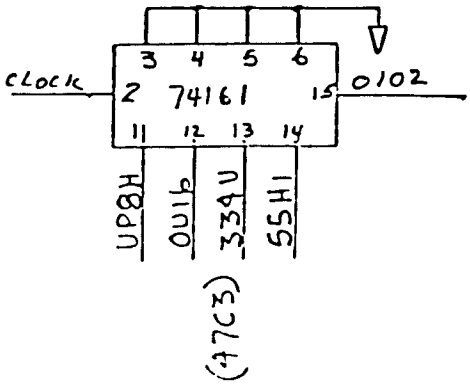
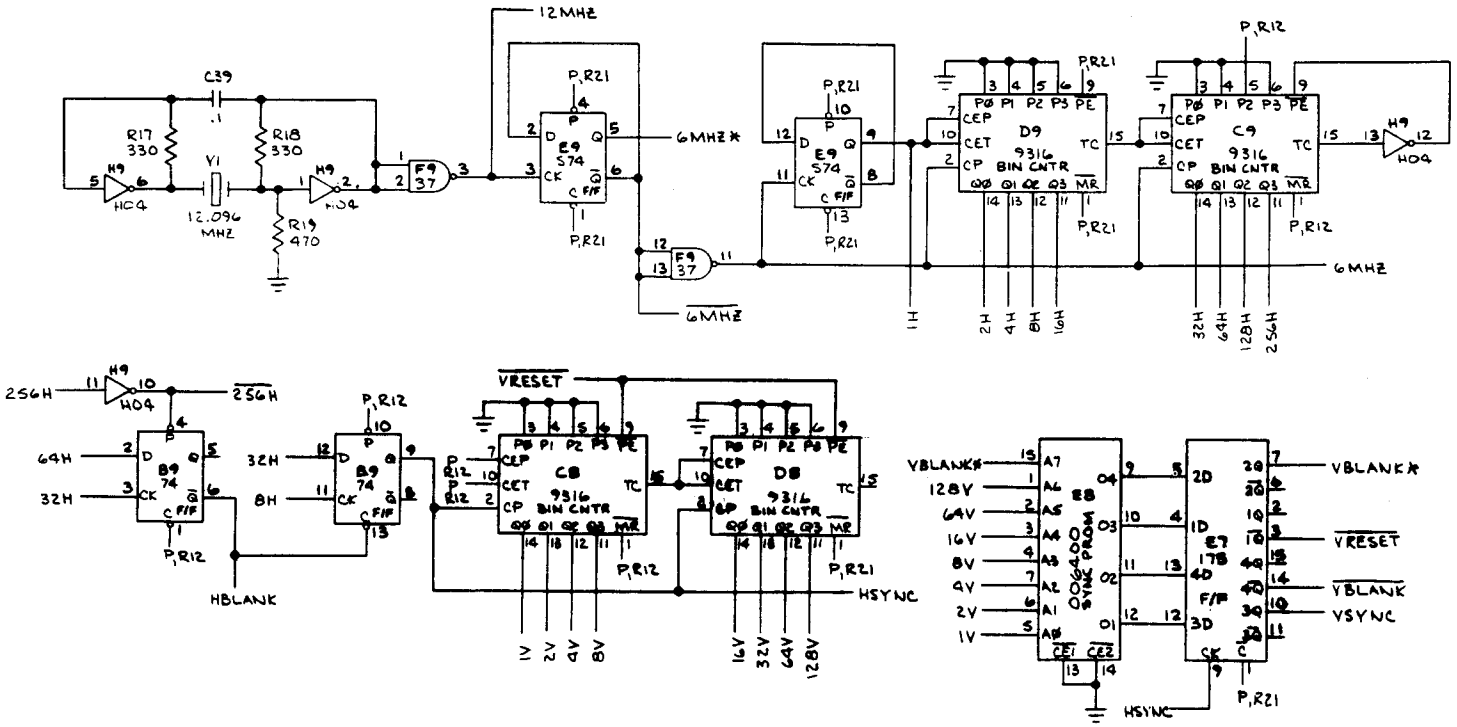


Figure 6 shows a counter which has an incorrect signature (47C3) on pin 13 but correct signature on the ohter pins. To determine if the IC is bad, lift or clip pin 13 and take the signature from the pin. If it is correct you will find another signature on the trace showing a short trace. Incorrect signatures on the lifted or slipped pin indicates a bad IC.

In any test we must check to be sure the device has proper voltages. For the CPU we must also be certain we have clock both $\emptyset_1$ & $\emptyset_2$, if required. A properly designed NOP will eliminate the possibility of an interrupt causing a CPU lockup.

IV

CLOCK & SYNC CIRCUITS



In the above circuit we notice a 12 MHZ clock which is divided by 2 by E9. This 6 MHZ square wave is used to clock the horizontal divider chain (E9, D9, C9, and B9). Derived from this chain is H blank (B9-6) and H SYNC (B9-9).

H SYNC is used to clock the vertical SYNC Divider Chain (C8, D9, E8, and E7). V Blank, V Reset, V Blank, and V SYNC are outputs of E7.

These SYNC circuits are a "piece of cake" to obtain signatures from and to test using SA. Here's all you have to do:

HORIZONTAL SYNC

A. Use D9-2 as CLOCK ⎍⎍
Use 9-11 as START/STOP ⎍⎍

Read and record signatures on D9 pins 7, 14, 13, 12, 11 and 15. If these signatures are correct, proceed.

B. Use C9-2 as CLOCK ⎍⎍
Use C9-11 as START/STOP ⎍⎍

Read and record signatures on C9 pins 10, 9, 14, 13, 12, 11 and 15, and E7-9.

C. After the signatures are obtained on all above pins in steps A & B leave SA connected as in step B and read and record signatures from B9, 2, 3, 4, 5 & 6. Pin 6 gives you H blank.

Now read and record signatures from B9-8, 9, 11, 12 and 13. Pin 9 gives you H SYNC.

If H SYNC is OK, you can proceed to:

VERTICAL SYNC

A. CLOCK C8-2 ⎍⎍
START/STOP C8-11 ⎍⎍

Read and record signatures on C8 pins 14, 13, 12, 11, 15 and 9. If these signatures are OK, proceed.

B. CLOCK D8-2 ⎍⎍
START/STOP D8-11 ⎍⎍

Read and record signatures on D8-9, 10, 14, 13, 12, 11, 15
E8-pins 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12 & 15
E7-pins 2*, 3, 4, 5, 6*, 7, 10, 11*, 12, 13, 14 & 15

You have fully documented this SYNC Chain.

The sequence of the SYNC circuit is CLOCK-H SYNC-V SYNC, all in series. Bad clock can kill everything, H SYNC can kill V SYNC. So start at the beginning.

A quick check can be made by connecting the Signature II to:

C9-2 CLOCK ⎍⎍
C9-11 START/STOP ⎍⎍

Read signature at B9-9, if correct you have horizontal SYNC.

Connect Signature II

D8-2 CLOCK ⎍⎍
D8-11 START/STOP ⎍⎍

Read signature at E7-10, if correct you have V SYNC.

*NOTE: On pins so marked an unstable or no signature may be noticed....

V
MICROPORCESSOR RESET

One of the basic system control functions is the system RESET signal. Whether this signal is generated automatically by external power-on circuitry or manually from a push-button switch, the system components must obey a fixed set of rules to assure proper system operation.

In 6500, 6800 series microprocessor systems the RESET pin must be held low during power-on until the supply voltages and clocks have stabilized. Usually the RESET pin is held low for a minimum of 8 clock pulses. The 8080, Z80, etc., series is just the opposite. RESET is normally low and is taken high for a minimum of 3 clock pulses.

Checking of the reset circuit is best done with the Signature II probe. (Be sure the pod black wire is connected to PC board ground.)

On microprocessors which have watchdog or automatic data resets a continuing pulsing of the reset is an indication of a memory problem. This problem can be located by Signature II using instructions in sections 6-8 of this paper.

The reset circuits are rather simple so we need not spend too much time on them.

VI
CPU AND ADDRESS BUS

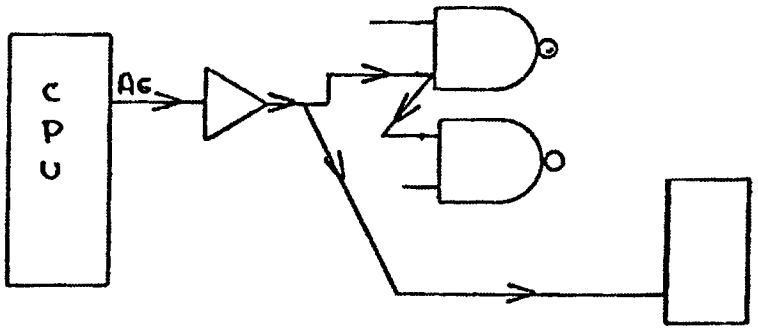
To test a CPU we must give it an instruction which it can execute in a repetitive fashion and generate signatures which are stable and the same for all like CPU's (8080's, Z-80's, 6502's, etc.). This instruction is "NOP". In order that we can be assured the CPU will only see this instruction we must break the data bus and "hardwire" the NOP into the CPU. This is done by a Kurz-Kasch NOP fixture.

This NOP fixture does a few other things such as disable the interrupts so the processor will run even if board faults would tend to "lock up" the processor.

The NOP instruction causes the CPU to become a 65K counter with its 16 address lines being the outputs. If the address outputs are counted through from all "0's" to all "1's", then the address bus has been exercised for all possible combinations and it does this repeatedly. Each of the address lines $A_{\emptyset}$ - A_{15} have a unique signature unless there is a fault with the processor or a trace or a device connected to that address line.

To obtain the signature for the address lines we use the system clock and as in all counters (remember the CPU is now a counter) the start and stop go to the most significant bit. Since we wish to input the maximum number of clock pulses we start and stop on the A-15 leading edge.
Page 11 gives the address bus signatures.

Once we have determined we have a operational CPU with proper signatures on each address pin, each address line can be traced to its termination. The figure below shows the logical test progression for the address bus. Checking inputs of the bus drivers and then the outputs assure you that the address bus is clear. If it isn't, use the fault isolation instructions give you in Section III.



All devices are now being properly addressed so we are ready to begin to measure their response.

NOP FIXTURE SIGNATURES - FOR PROCESSORS

PROCESSOR	Z-80	*6502-A	8080	6502		6800
CLOCK	PIN 21	Pin 37	PIN 17	PIN 37		PIN 37
START/STOP	5	25	36	25		25
GROUND	29	21	2	21		21
A0	30	UUUU	25	9	UUUU	9
A1	31	5555	26	10	FFFF	10
A2	32	CCCC	27	11	8484	11
A3	33	7F7F	29	12	P763	12
A4	34	5H21	30	13	1U5P	13
A5	35	OAFA	31	14	0356	14
A6	36	UPFH	32	15	U759	15
A7	37	52F8	33	16	6F9A	16
A8	38	HC89	34	17	7791	17
A9	39	2H70	35	18	6321	18
A10	40	HPP0	1	19	37C5	19
A11	1	1293	40	20	6U28	20
A12	2	HAP7	37	22	4FCA	22
A13	3	3C96	38	23	4868	23
A14	4	3827	39	24	9UP1	24
A15	5	755P	36	25	0002	25
VCC		0001			0003	

*Atari games will use a KK 6502-A NOP. The signatures will be the same as for an 8080.

Signature II connections to 6502 does not change.

THIS IS CORRECTED COPY
FOR PAGE 17 OF
SIGNATURE II MANUAL

CPU NOP FIXTURE CONNECTIONS

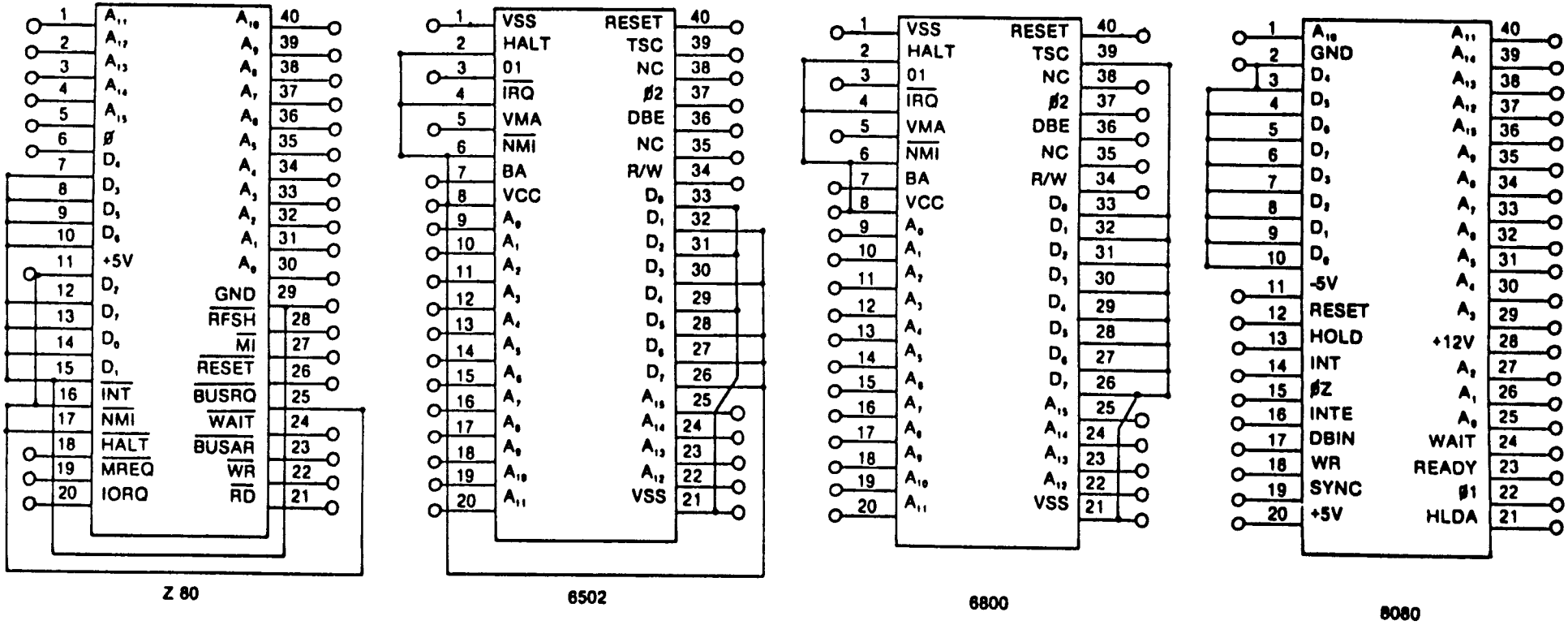


FIGURE
4

O- THESE PINS CONNECT DIRECT FROM LOGIC BOARD CPU SOCKET TO FIXTURE SOCKET PINS WITHOUT O- ARE INTERCONNECTED AS SHOWN. THESE PINS ARE NOT CONNECTED TO LOGIC BOARD CPU SOCKET.

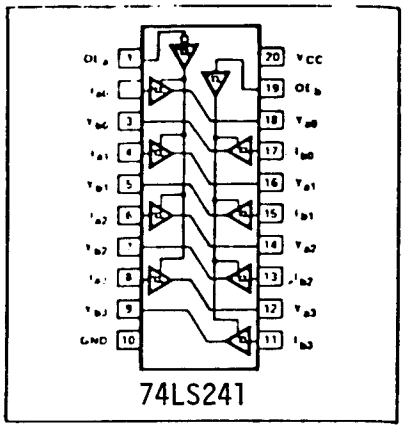
VII

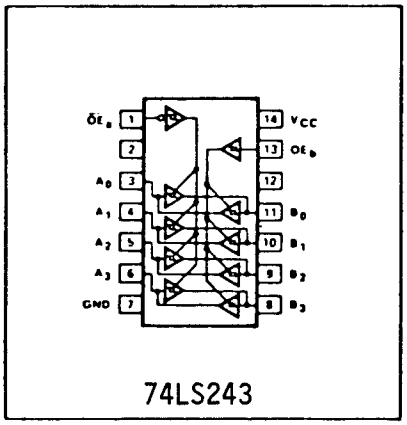
BUS DRIVERS

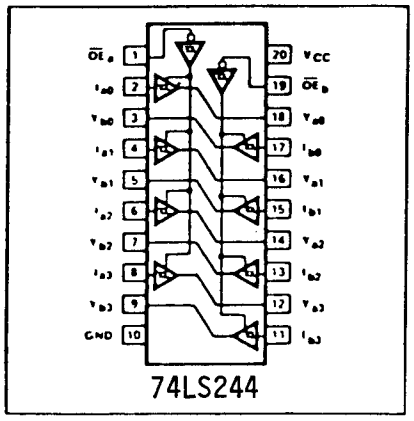
Bus drivers are devices which isolate sections of the various buses. The drivers are directional and have the capability of being removed electrically from the bus by proper logic levels on the OE (output enable) inputs.

When testing a computor board with signature analysis often the drivers are disabled. In order to take signatues on the inputs and outputs they often must be force enabled.

The 74LS241 may be forced by applying a low on Pin 1 and a high on Pin 19. Pin 19 cannot be pulled high. This must be done by a previous inverting gate whose input can be taken low or by clipping Pin 19. Pin 1 can just be pulled low.







A 74LS243 is a bi-directional driver. If pin 1 and 13 are low, data will flow from A to B. A high on pins 1 and 13 will cause data to flow from B to A. Under no circumstances take 1 low and 13 high at the same time or pull either pin high. (see 74LS241 above)

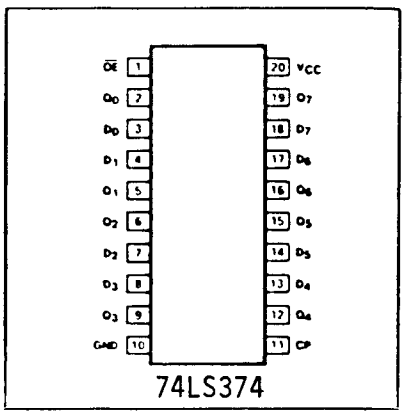
The 74LS244 operates much like the LS241 except the levels on pins 1 and 19 must be low to enable the devices.

On non-inverting devices like the ones above, signatures on input and output will be the same when enabled. Inverting drivers will have different input-out signatures.

A latch such as the 74LS374 must have two conditions satisfied to pass data.

1. pin 1 $\overline{OE}$ must be low.
2. pin 11 must clock high.

Pin 1 can be taken low and held while a pulser (KK/HL-480) is touched to pin 11 and "one shot" pulsed. This will transfer whatever data on D to Q.



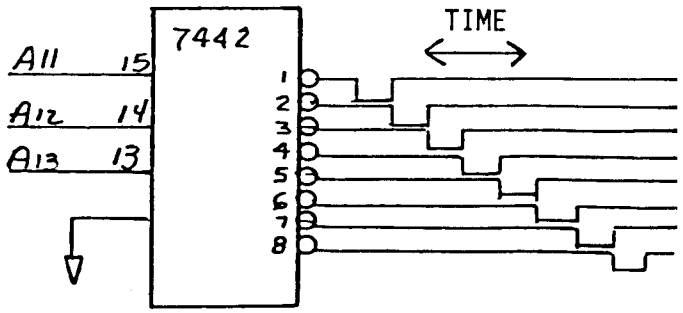
VIII

ROMs, RAMs, DECODERs

The ROM is the permanent memory which causes the processor system to begin and continue to perform in a predetermined manner. Bad ROMs can and will cause the system to do crazy things. Fortunately ROMs are very easily tested by Signature Analysis. The easiest way is whth a Kurz-Kasch ROM Test I or ROM TEST II, but equally as accurate is with the Signature II.

For the ROM Test we still use the system clock for the Signature II, but for start/stop we must select a point which will allow us to look at an individual device.

This point is the ROM enable (decoder). The figure below shows the address decoder and what its outputs look like.



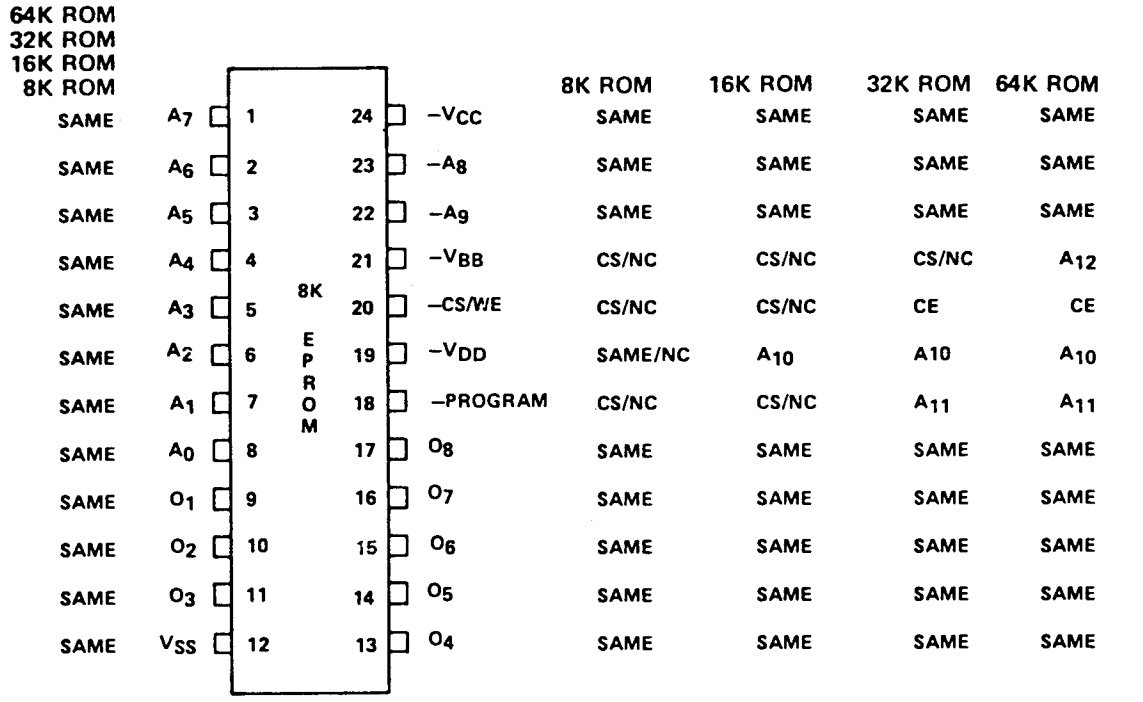
The decoder is addressed by A11, 12, & 13. Three address lines when decoded will cause 8 outputs to sequence.

A13	A12	A11	7442
0	0	0	Pin 1 goes low
0	0	1	Pin 2 goes low
0	1	0	Pin 3 goes low
0	1	1	Pin 4 goes low
1	0	0	Pin 5 goes low
1	0	1	Pin 6 goes low
1	1	0	Pin 7 goes low
1	1	1	Pin 8 goes low

With the Signature II connected in the address signature mode (start ⎍⎍, stop ⎍⎍ to A15) you will get the signatures for the 7442 outputs. These signatures will assure you that the ROMs will be properly enabled.

If we look at a ROM we can see what needs to transpire in order to get it to output the data it holds.

EPROM AND ROM PIN CONNECTION COMPATIBILITY
Figure 6



Virtually all pin connections are same for an 8K EPROM through a 64K ROM. Significant difference occurs on the EPROM V_{BB}, V_{DD}, and program pins. These pins in ROM are either CS/$\overline{CS}$ functions or address inputs with some manufacturers allowing no-connect options. With N/C option, EPROM can be directly replaced by ROM with no circuit change, except when using 16K or larger.

We see that this ERPOM (2716) is to be addressed by AØ - A10, that it needs ground & +5V for its operation. We also note there are two pins, 18 & 20, $\overline{CE}_1$ and $\overline{CE}_2$ which may be used as enables. If we tie pin 20 low we have a single enable, pin 18 ($\overline{CE}$). The bar over the CE indicates that this pin must go low to enable the EPROM.

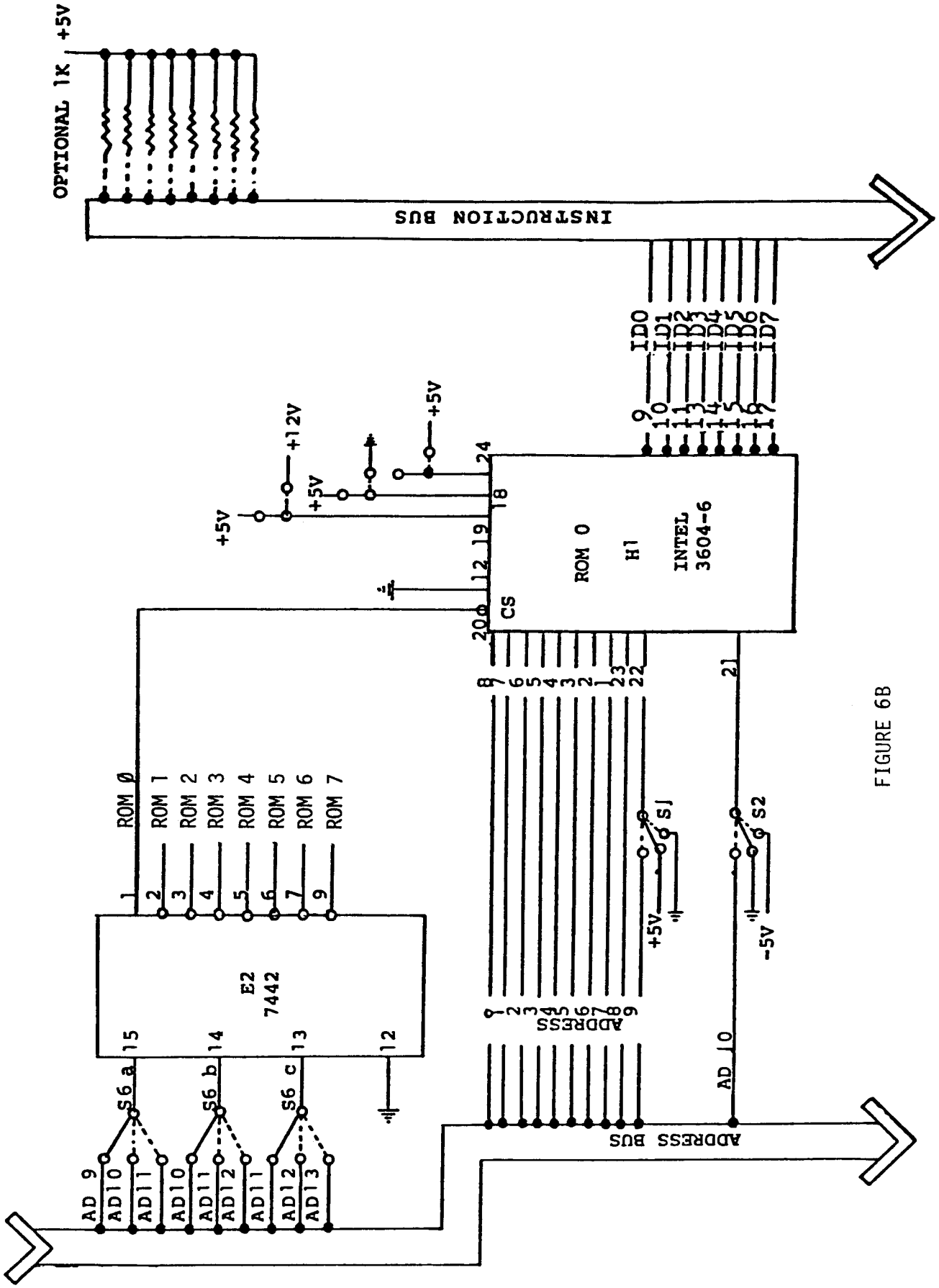
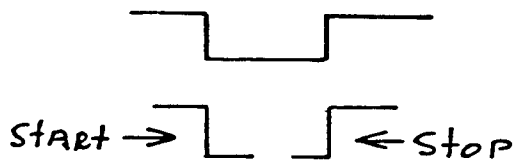

OPTIONAL 1K
+5V
INSTRUCTION BUS
ID0
ID1
ID2
ID3
ID4
ID5
ID6
ID7
+12V
+5V
+5V
+5V
ROM 0
H1
INTEL
3604-6
CS
ROM Ø
ROM 1
ROM 2
ROM 3
ROM 4
ROM 5
ROM 6
ROM 7
E2
7442
S6 a
S6 b
S6 c
AD 9
AD10
AD11
AD10
AD11
AD12
AD11
AD12
AD13
ADDRESS
AD 10
ADDRESS BUS
SJ
S2
+5V
-5V

FIGURE 6B

Going back to page 14 we see that the 7442 outputs go low in sequence starting with the address lines all at "0". First pin 1 goes low. If we tied pin 18 of the EPROM to pin 1 of the decoder, for a period of time the 2716 would be low thereby enabled and would output data. The data outputted will be dependant upon the addresses applied, in practice the EPROM would be totally addressed and therefore all data contained therin would appear at the data outputs ($D_{\emptyset}$ - D_7). Further there will be an address change and a resulting data change with each clock pulse.

Since we wish to have the Signature II only look at the data bus (EPROM output) during the time the EPROM is enabled we set the start-stop switches in a manner to portray the enable pulse.



The Signature II will give a signature for each of the 8 data output data streams which would occur between the start and stop times.

Should there be more than 1 ROM or other memory device necessary to perform the computor's mission one would only need to connect the other ROM (s) to the 7442 enables in the order you would want them to operate. (see page 16)

To read signatures for the additional ROM (s) one would only need to connect the Signature II to the enable pin (18) of the device you wish to check.

If when checking a multi ROM bank the signatures will not stabilize, remove all ROMs except the one you are checking, or connect a 4.7 K pull-up to data probe tip.

Here again faults such as outlined in the common fault section will be effective for isolating a problem.

Les us stop at this point and reflect where we are:

1. CPU is running
2. Address bus including drivers (buffers) is clear and healthy.
3. ROMs are being addressed and enabled properly.
4. ROMs are outputting proper data and the data bus is clear.

We move on to the next major item in the computor:

RAM

The RAMs are the most difficult components to test in a processor. It is necessary to be able to write into them a known program which can be repeated. This must be done with software.

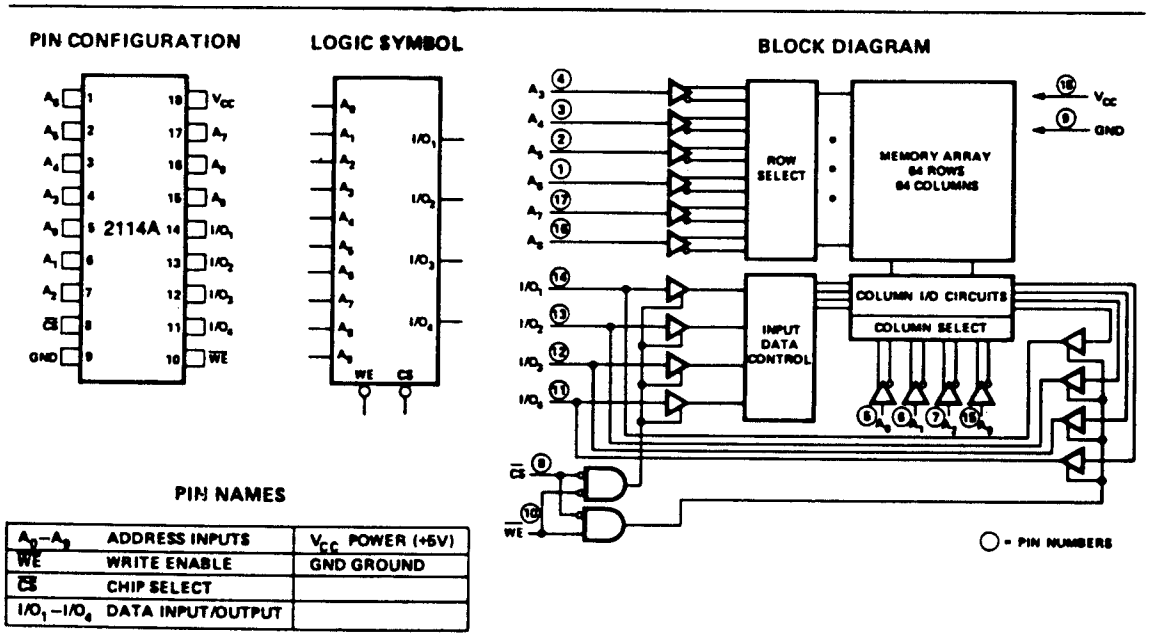
If the board you are testing has RAM-TEST program in its memory or can be loaded in via a PIA then the processor system at this point can execute the program and dynamically test the RAMs. This is a good deal, but, if you don't have a built-in RAM test they can still be tested.

A simple program which will write a "0" and a "1" into alternate cells in the RAMs then compliment this pattern ("1" where there was a "0", and a "0" where there was a "1"). This forms a checkerboard pattern.

Kurz-Kasch has programs for several of the processors and will assist in programs for specific applications.

Assume we have software (a ROM) with a RAM TEST pattern in it, the Signature II can tell you which one of a multi-ram bank is defective.

Refer back to pages 14 & 16, and we see a decoder, RAMs are enabled in the same manner as ROMs with one exception---a read-write enable. Look at the figure below, which is a pinout of a 2114 RAM.



Pins 1-7 and 15-17 are the address lines which define the cells into which data is to be written or read. Pin 8 is the chip enable just like the ROM. When this pin goes low the RAM is enabled but---it can't input data until pin 10 (write) goes low. This pin is controlled by the CPU which commands that it write onto or read from the data bus. Remember a command must be always viewed from the CPU. The command is what it is doing---not other devices.

If the Signature II start and stop are connected to the CE (just as in the case of ROM) and set the switches to read signatures where CE is low you will get signatures for the data being written into the RAM. By using the write line as start---stop we can read signatures for the contents of the RAM.

IX
MULTIPLEXERS

A multiplexer is a device which has two (2) or more inputs and selects one input at a time to be outputted.

For a practical application the Midway 8080 motherboard uses four 9322 muxes. Pin 1, when low allows the data on input 1 to appear at the output. When pin 1 is high data on input 2 appears at the output.

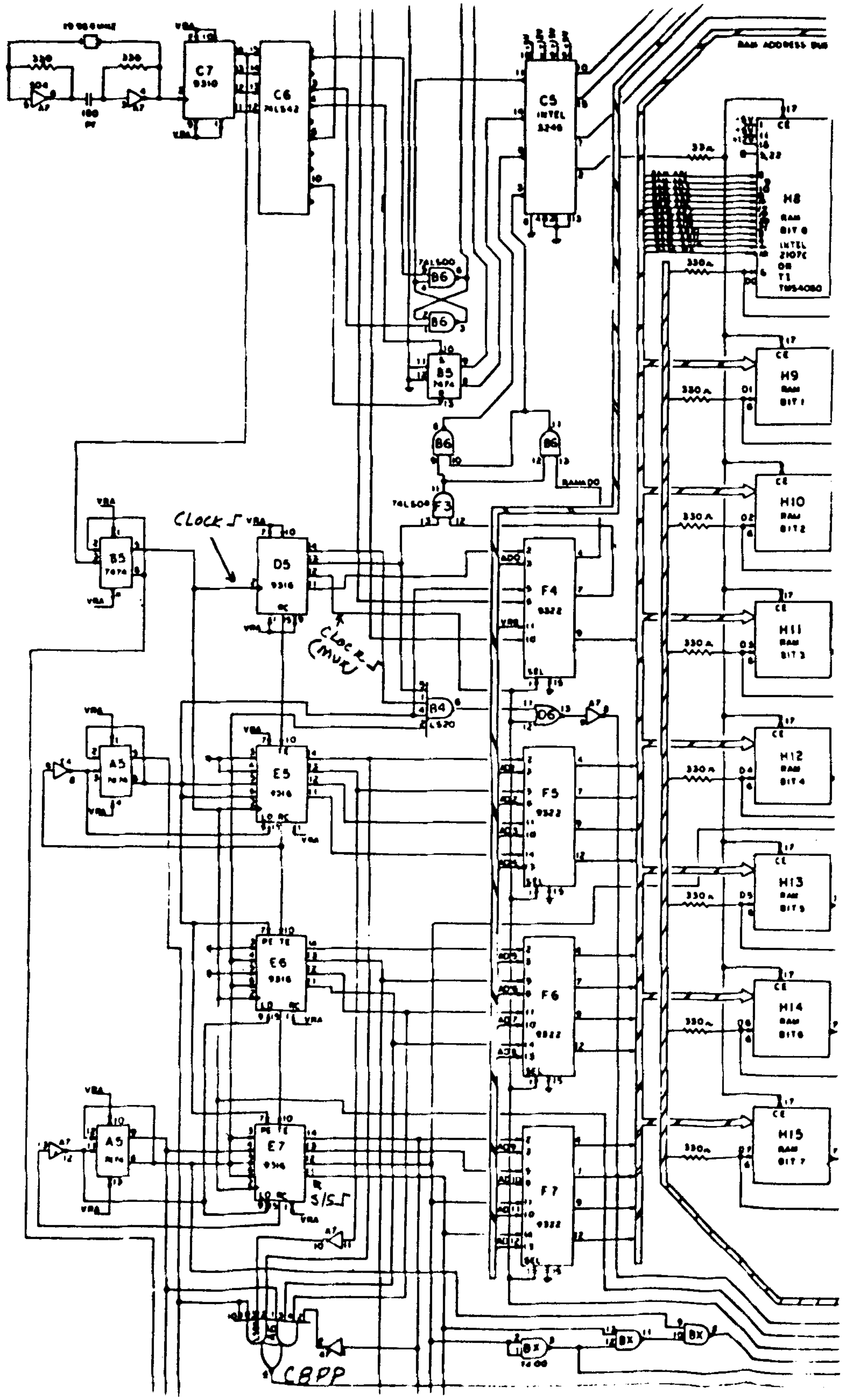
The schematic on page 22 shows one of the two inputs orginate at the CPU and the other from the counter chain. In other words data from two systems can appear at the mux output. This normally would cause the signatures to be unstable. The Signature II clock when in the rising ⎍⎾ mode allows us to look at input 1 through the mux and gives a stable signature.

To test the multiplexes with the Signature II connect as follows and read the signatures:

8080 MIDWAY RAM ADDRESS MULTIPLEXERS

CLOCK D5 - 12⎍⎾
START/STOP E7 - 11⎍⎾
VCC H6HU

PIN	F4	F5	F6	F7
2	4HC5	3C6F	43CF	6172
5	CCC6	U249	7U95	599P
11	H6HU	8F33	C674	067U
14	Dead Band	6481	3669	07HA
4	4HC5	3C6F	43CF	6172
7	CCC6	U249	7U95	599P
9	H6HU	8F33	C674	067U
12	H6HU	6481	3669	07HA


C7
C6
C5
RAM ADDRESS BUS
H8
RAM
BIT 0
H9
RAM
BIT 1
H10
RAM
BIT 2
H11
RAM
BIT 3
H12
RAM
BIT 4
H13
RAM
BIT 5
H14
RAM
BIT 6
H15
RAM
BIT 7
B6
B5
F3
A4
D5
D6
F4
F5
F6
F7
E5
E6
E7
A5
BX
CLOCK
CLOCK (MUX)
S/S
CBPP