Trouble Shooting Microprocessors With ''Free Run'' Fixtures

By Mark Stirling — Coastal Games Ltd.
Richmond, B.C.

This is the first of a series of articles intended to help games service people detect and cure microprocessor system faults quickly and easily. This will be accomplished with the aid of some inexpensive and easily constructed tools.

When a microprocessor system fails to the point where the self-test will not run, troubleshooting the fault can quickly become extremely difficult. This is due to a simple characteristic common to all microprocessor sytems. That is; the data bus forms part of a complex feedback loop and any error in the loop compounds itself causing all components in the loop to act erractically,whether they are good or not. Signature analysis has been put forth as a sophisticated technique to aid in trouble shooting but even this, requires a predictable core of the system to function in order to be of any use. Free-run fixtures will give you a predictable pattern from the microprocessor allowing orderly troubleshooting.

What exactly does this free run fixture do? A free run fixture forces the processor to cycle continuously through all its addresses. This in turn means that all chip selects are activated with a stable regular signal that you can now use your oscillosope to check. In this manner for instance address decoding can be located. Shorted or open address lines can also be easily detected Address buffers can be checked because we have a predictable signal passing through. The nature of the signal is similar to the output of a 16 stage counter. Address line A0will cycle fastest with Address line A1 at half the speed of A0and so on to address line A15. This regular pattern also will generate a stable data stream for those of you with access to signature analysis.

How does the fixture work? The free run fixture works by isolating the data bus, thereby opening the feedback loop and forcing a hard wired no operation instruction on the data bus at the processor. No matter what address is output by the procesor the instruction seen is the same. This forces the program counter to step in a regular fashion through the complete memory space of the system.

Fig 1 shows how a typical free run test fixture is constructed. Figs 2, 3, 4, 5 show the actual connection for 6800, Z80, 6502 and 8080A processors; these being the most common processors at the moment.

To design your own fixture for another processor, all that is required is to wire the no operation (NOP) instruction to the processor and ensure that interrupt and halt and BUS request lines cannot be activated.

Next month I will explain a simple address mapper which can help detect some types of intermittent faults.

Feel free to drop me a line. Your comment will be appreciated along with any questions. I cannot promise to answer each letter individually but will answer those of merit in upcoming articles.

Coastal Games Ltd.,
#105 — 3860 Jacombs Road,
Richmond, B.C.
V6V 1Y6
(604) 270-9346

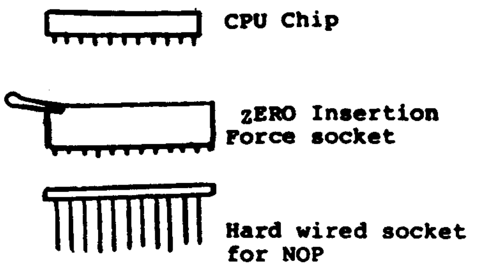
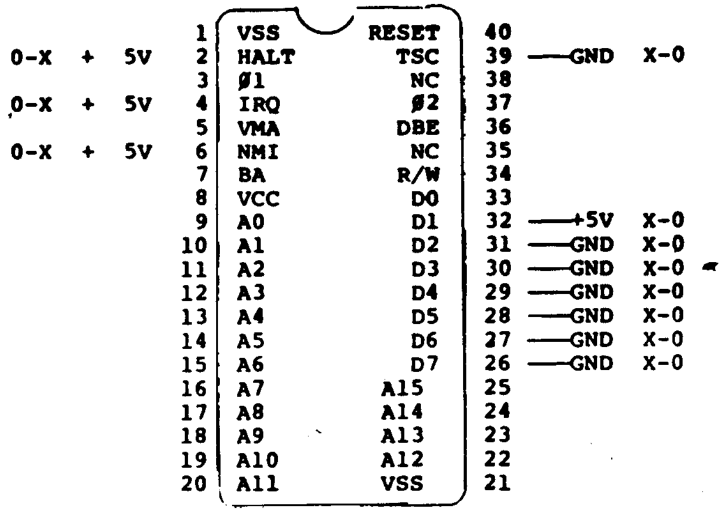
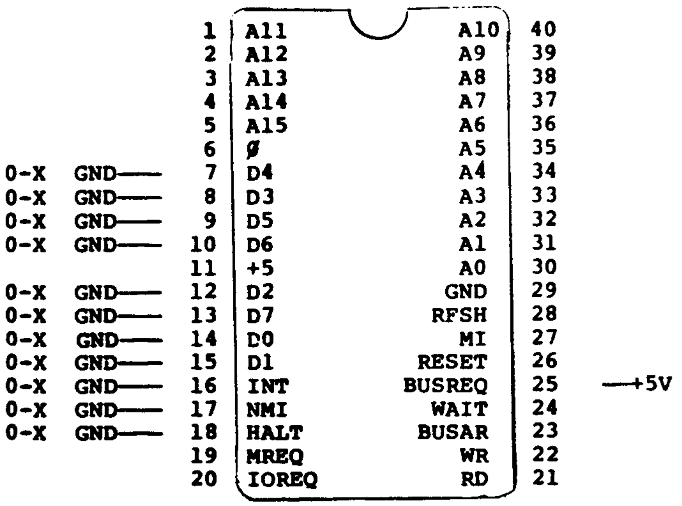


FIG. 1 "FREE RUN" FIXTURE PLUGS INTO GAME CPO SOCKET.



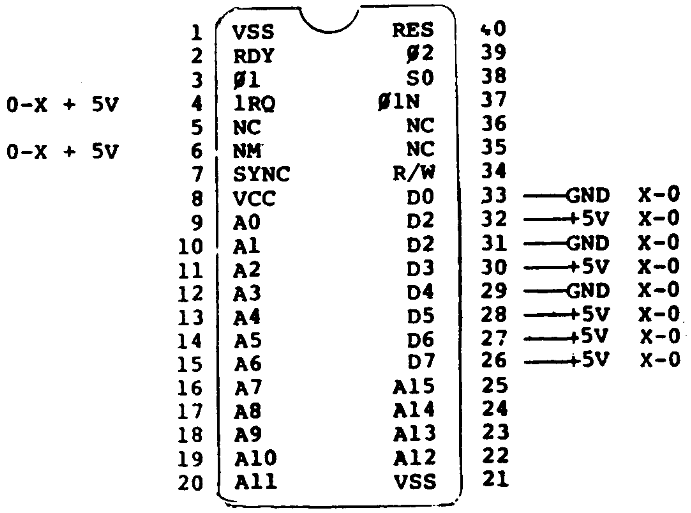
	1	VSS	RESET	40	
0-X + 5V	2	HALT	TSC	39	——GND X-0
	3	Ø1	NC	38	
0-X + 5V	4	IRQ	Ø2	37	
	5	VMA	DBE	36	
0-X + 5V	6	NMI	NC	35	
	7	BA	R/W	34	
	8	VCC	DO	33	
	9	A0	D1	32	——+5V X-0
	10	A1	D2	31	——GND X-0
	11	A2	D3	30	——GND X-0
	12	A3	D4	29	——GND X-0
	13	A4	D5	28	——GND X-0
	14	A5	D6	27	——GND X-0
	15	A6	D7	26	——GND X-0
	16	A7	A15	25	
	17	A8	A14	24	
	18	A9	A13	23	
	19	A10	A12	22	
	20	A11	VSS	21	

FIG. 2 6800



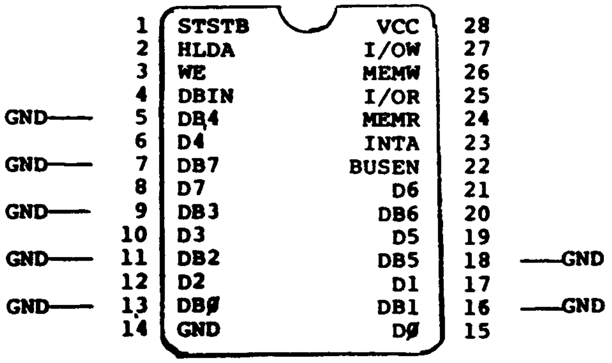
	1	A11	A10	40	
	2	A12	A9	39	
	3	A13	A8	38	
	4	A14	A7	37	
	5	A15	A6	36	
	6	Ø	A5	35	
0-X GND——	7	D4	A4	34	
0-X GND——	8	D3	A3	33	
0-X GND——	9	D5	A2	32	
0-X GND——	10	D6	A1	31	
	11	+5	A0	30	
0-X GND——	12	D2	GND	29	
0-X GND——	13	D7	RFSH	28	
0-X GND——	14	D0	MI	27	
0-X GND——	15	D1	RESET	26	
0-X GND——	16	INT	BUSREQ	25	——+5V
0-X GND——	17	NMI	WAIT	24	
0-X GND——	18	HALT	BUSAR	23	
	19	MREQ	WR	22	
	20	IOREQ	RD	21	

FIG. 3 Z80



	1	VSS	RES	40	
	2	RDY	Ø2	39	
	3	Ø1	SO	38	
0-X + 5V	4	1RQ	Ø1N	37	
	5	NC	NC	36	
0-X + 5V	6	NM	NC	35	
	7	SYNC	R/W	34	
	8	VCC	D0	33	——GND X-0
	9	A0	D2	32	——+5V X-0
	10	A1	D2	31	——GND X-0
	11	A2	D3	30	——+5V X-0
	12	A3	D4	29	——GND X-0
	13	A4	D5	28	——+5V X-0
	14	A5	D6	27	——+5V X-0
	15	A6	D7	26	——+5V X-0
	16	A7	A15	25	
	17	A8	A14	24	
	18	A9	A13	23	
	19	A10	A12	22	
	20	A11	VSS	21	

FIG. 4 6502



	1	STSTB	VCC	28	
	2	HLDA	I/OW	27	
	3	WE	MEMW	26	
	4	DBIN	I/OR	25	
GND——	5	DB4	MEMR	24	
	6	D4	INTA	23	
GND——	7	DB7	BUSEN	22	
	8	D7	D6	21	
GND——	9	DB3	DB6	20	
	10	D3	D5	19	
GND——	11	DB2	DB5	18	——GND
	12	D2	D1	17	
GND——	13	DBØ	DB1	16	——GND
	14	GND	DØ	15	

FIG. 5 8228/38 SYSTEM CONTROLLER FOR 8080A

NOTE: Due to the fact that its status signals are multiplexed with data merely cutting the data lines would completely stop its operation. The data lines must therefore be broken "outside" the system controller chip or chips that demultiplex the status and data lines.

KURZ-KASCH

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.

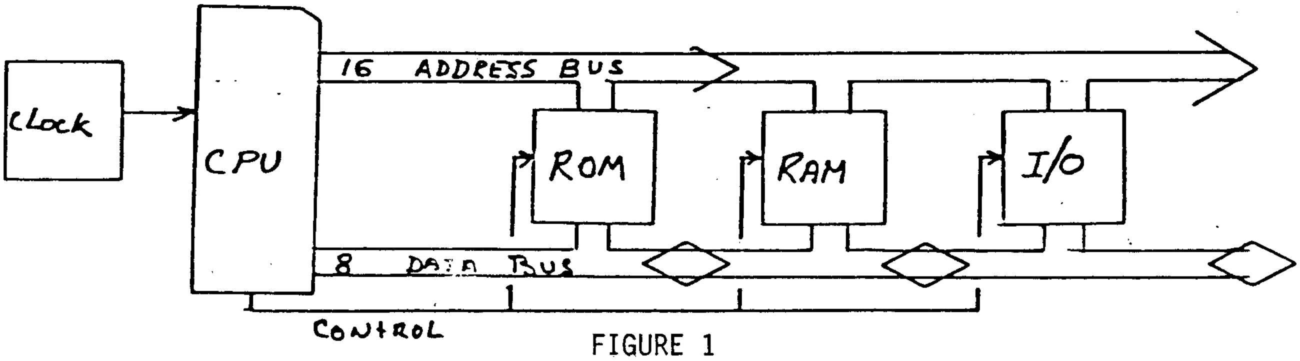


FIGURE 1

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.

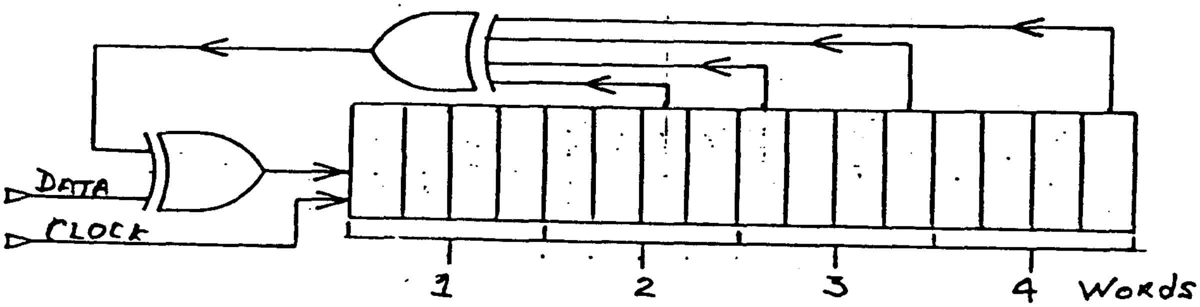
ROM is the permanent memory wherein resides the basic instructions for the CPU. Without ROM the CPU can do nothing. (ROM here is defined as CPU instructions).

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.

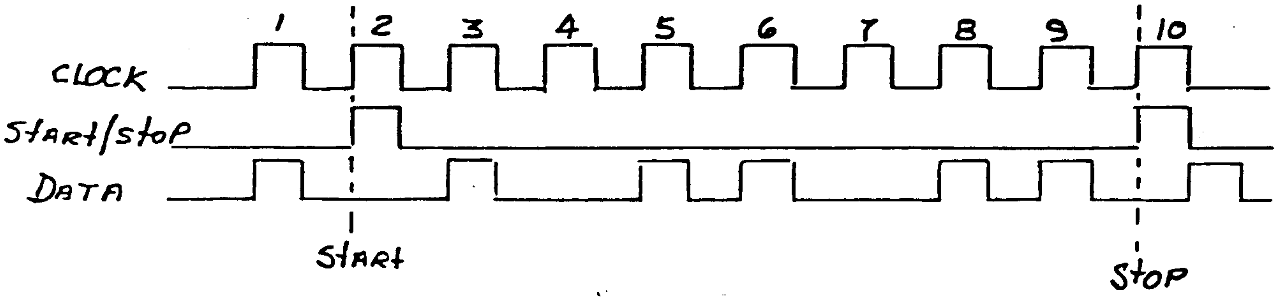


FIGURE 3

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: 1111111100000111111 = D953
1111111100000011111 = 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

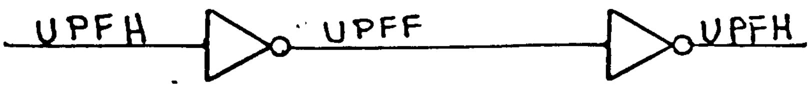


Figure 1 shows a normal inverter chain with signatures you might expect.

Figure 2

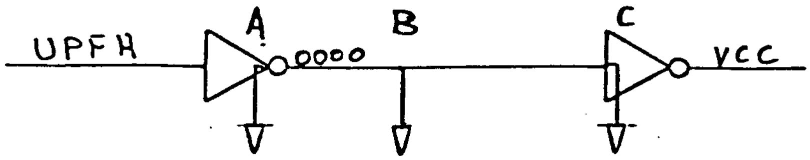


Figure 2 we see a proper signature on the first inverter's input and a "stuck at 0" at the output. As shown at points A, B & C we can have 3 failure modes. Point A is a failed output on the inverter, Point B would be a physical short to ground and C, a failed inverter input.

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

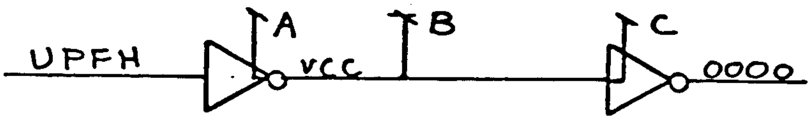


Figure 3 shows a gate or line with a failure mode "stuck at 1". This is fault isolated the same for a "stuck at 0".

Figure 4

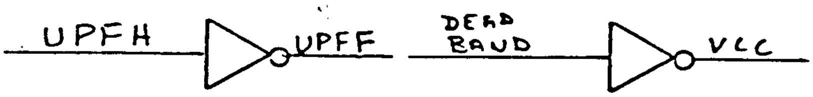


Figure 4 shows the signatures (or probes) indications for a broken trace.

Figure 5

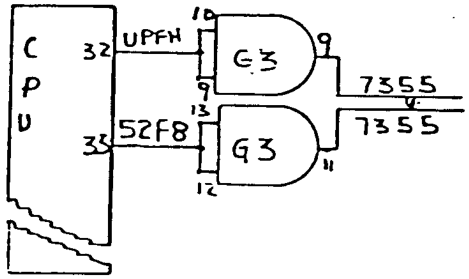


Figure 5 shows address lines from a CPU to buffers (G3) to the address bus. Proper signatures at the CPU and input of buffers but incorrect at buffer output is a fault mode. Where you find 2 or more incorrect but some signatures this indicated two bus lines shorted togeter.

Figure 6

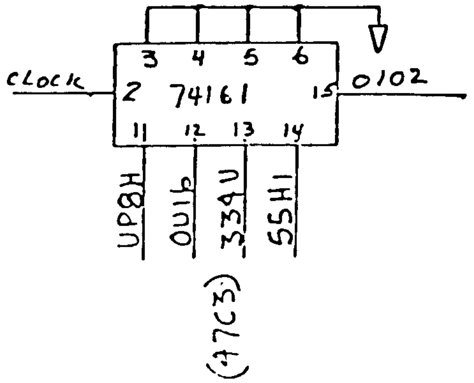
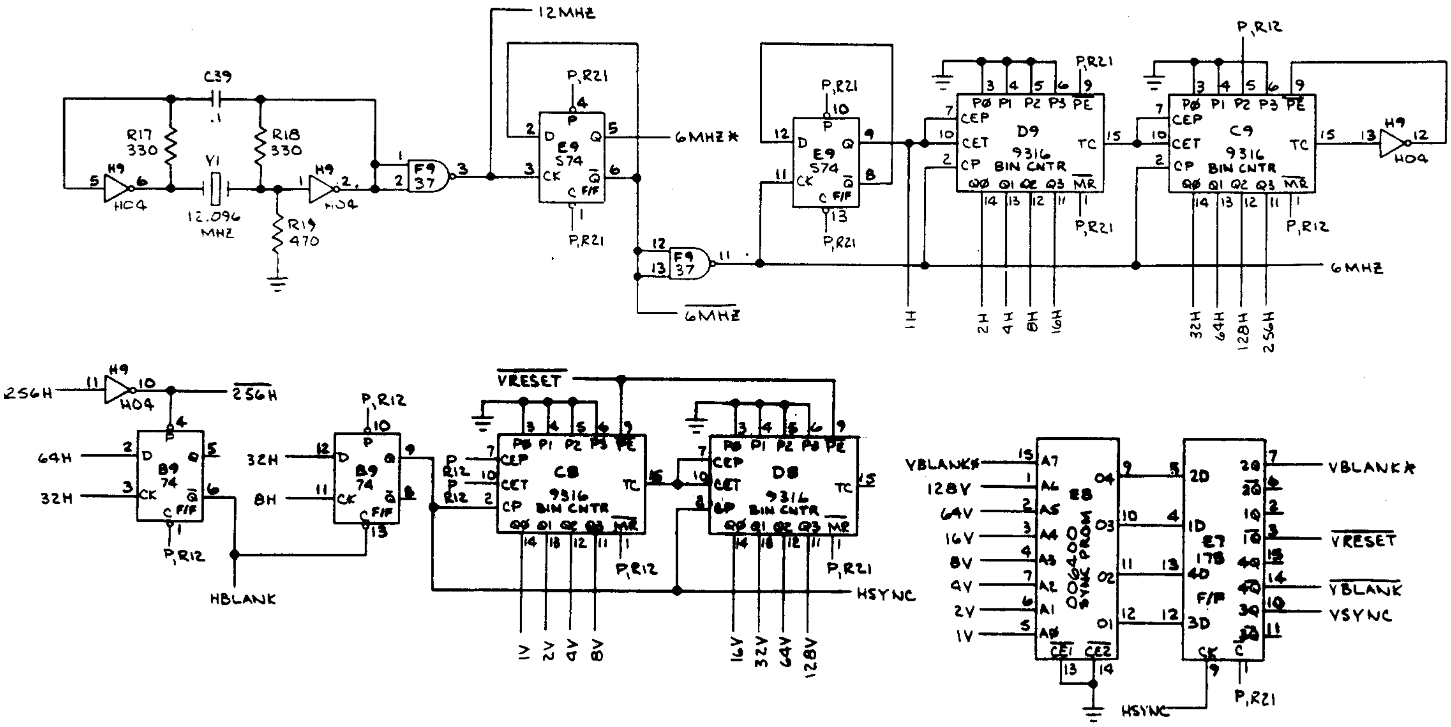


Figure 6 shows a counter which has an incorrect signature (47C3) on pin 13 but correct signature on the ohter pins. To determine if the IC is bad, lift or clip pin 13 and take the signature from the pin. If it is correct you will find another signature on the trace showing a short trace. Incorrect signatures 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 $Ø_1$ & $Ø_2$, 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 docummented 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

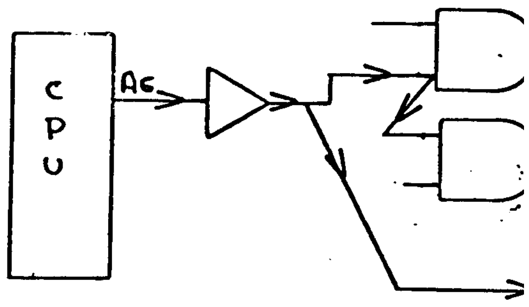
To test a CPU we must give it an instruction which it can execute in a repetitive fashion and generate signatures which are stable and the same for all like CPU's (8080's, Z-80's, 6502's, etc.). This instruction is "NOP". In order that we can be assured the CPU will only see this instruction we must 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
address lines being the outputs. If the address outputs
from all "0's" to all "1's", then the address bus has be
possible combinations and it does this repeatedly. Each
$A_\emptyset$ - A_{15} have a unique signature unless there is a fault
a trace or a device connected to that address line.

To obtain the signature for the address lines we us
and as in all counters (remember the CPU is now a counte
go to the most significant bit. Since we wish to input
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 wi
on each address pin, each address line can be traced to
figure below shows the logical test progression for the
inputs of the bus drivers and then the outputs assure yo
bus is clear. If it isn't, use the fault isolation inst
Section III.



All devices are now being properly addressed so we a
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	0AFA	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

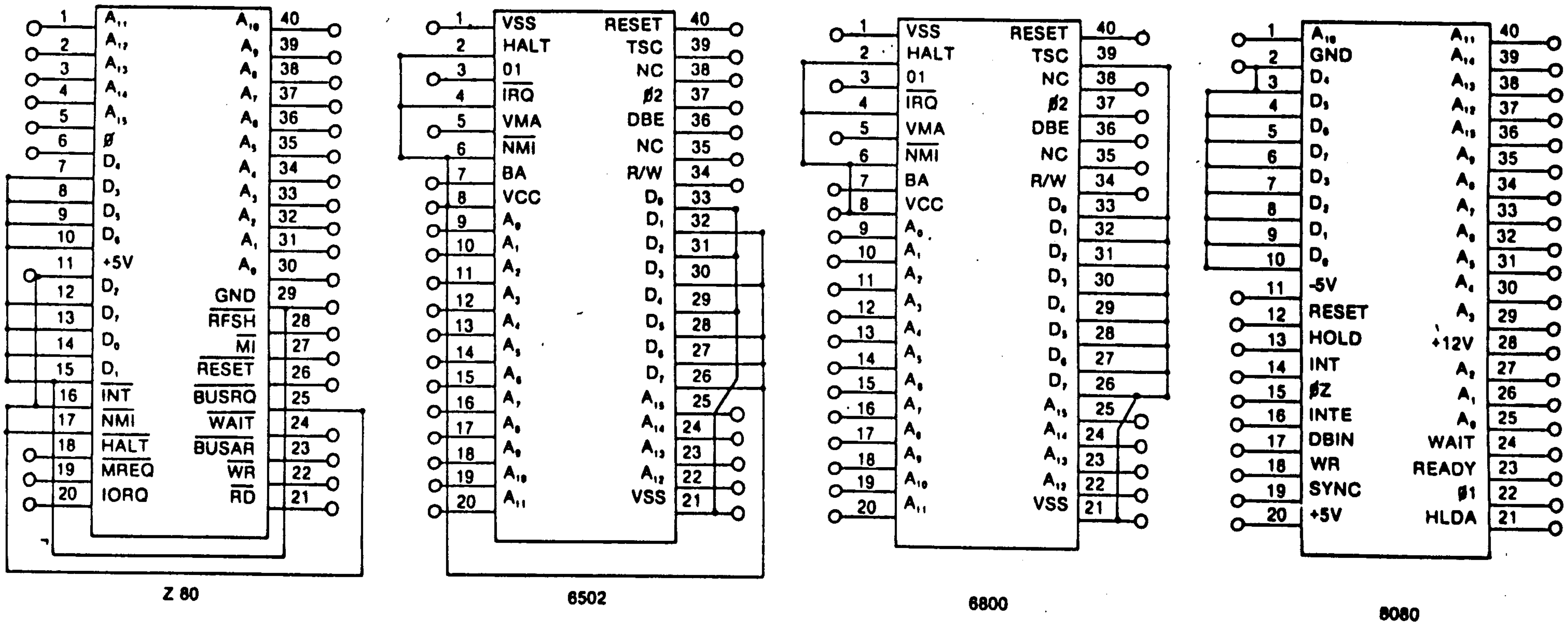


FIGURE
4

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.

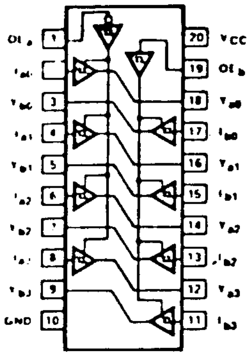
VII

BUS DRIVERS

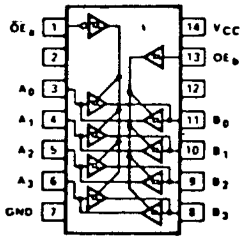
Bus drivers are devices which isolate sections of the various buses. The drivers are directional and have the capability of being removed electrically from the bus by proper logic levels on the OE (output enable) inputs.

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.

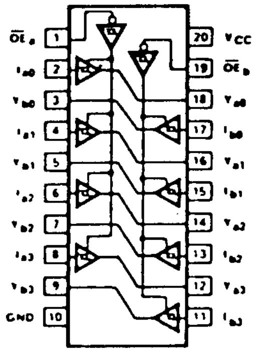
The 74LS241 may be forced by applying a low on Pin 1 and a high on Pin 19. Pin 19 cannot be pulled high. This must.be done by a previous inverting gate whose input can be taken low or by clipping Pin 19. Pin 1 can just be pulled low.



74LS241



74LS243



74LS244

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)

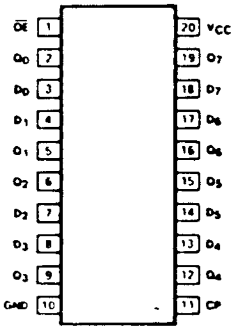
The 74LS244 operates much like the LS241 except the levels on pins 1 and 19 must be low to enable the devices.

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.



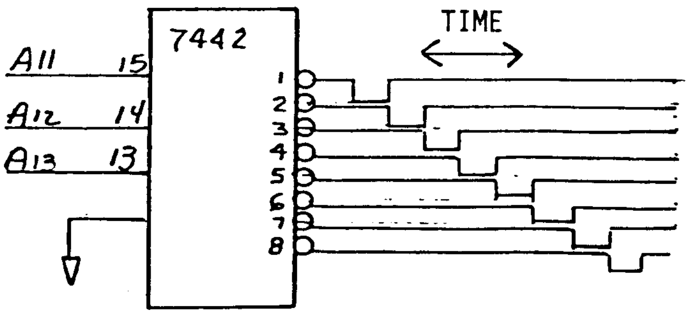
74LS374

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.



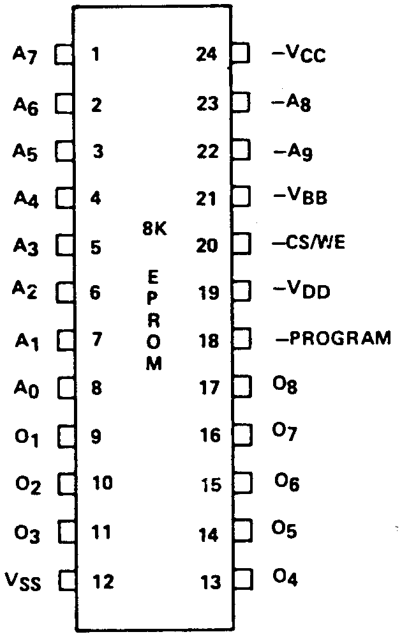
The decoder is addressed by A11, 12, & 13. Three address lines when decoded will cause 8 outputs to sequence.

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



64K ROM 32K ROM 16K ROM 8K ROM					8K ROM	16K ROM	32K ROM	64K ROM
SAME	A_7	1	24	–V_{CC}	SAME	SAME	SAME	SAME
SAME	A_6	2	23	–A_8	SAME	SAME	SAME	SAME
SAME	A_5	3	22	–A_9	SAME	SAME	SAME	SAME
SAME	A_4	4	21	–V_{BB}	CS/NC	CS/NC	CS/NC	A_{12}
SAME	A_3	5	20	–CS/WE	CS/NC	CS/NC	CE	CE
SAME	A_2	6	19	–V_{DD}	SAME/NC	A_{10}	A10	A_{10}
SAME	A_1	7	18	–PROGRAM	CS/NC	CS/NC	A_{11}	A_{11}
SAME	A_0	8	17	O_8	SAME	SAME	SAME	SAME
SAME	O_1	9	16	O_7	SAME	SAME	SAME	SAME
SAME	O_2	10	15	O_6	SAME	SAME	SAME	SAME
SAME	O_3	11	14	O_5	SAME	SAME	SAME	SAME
SAME	V_{SS}	12	13	O_4	SAME	SAME	SAME	SAME

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.

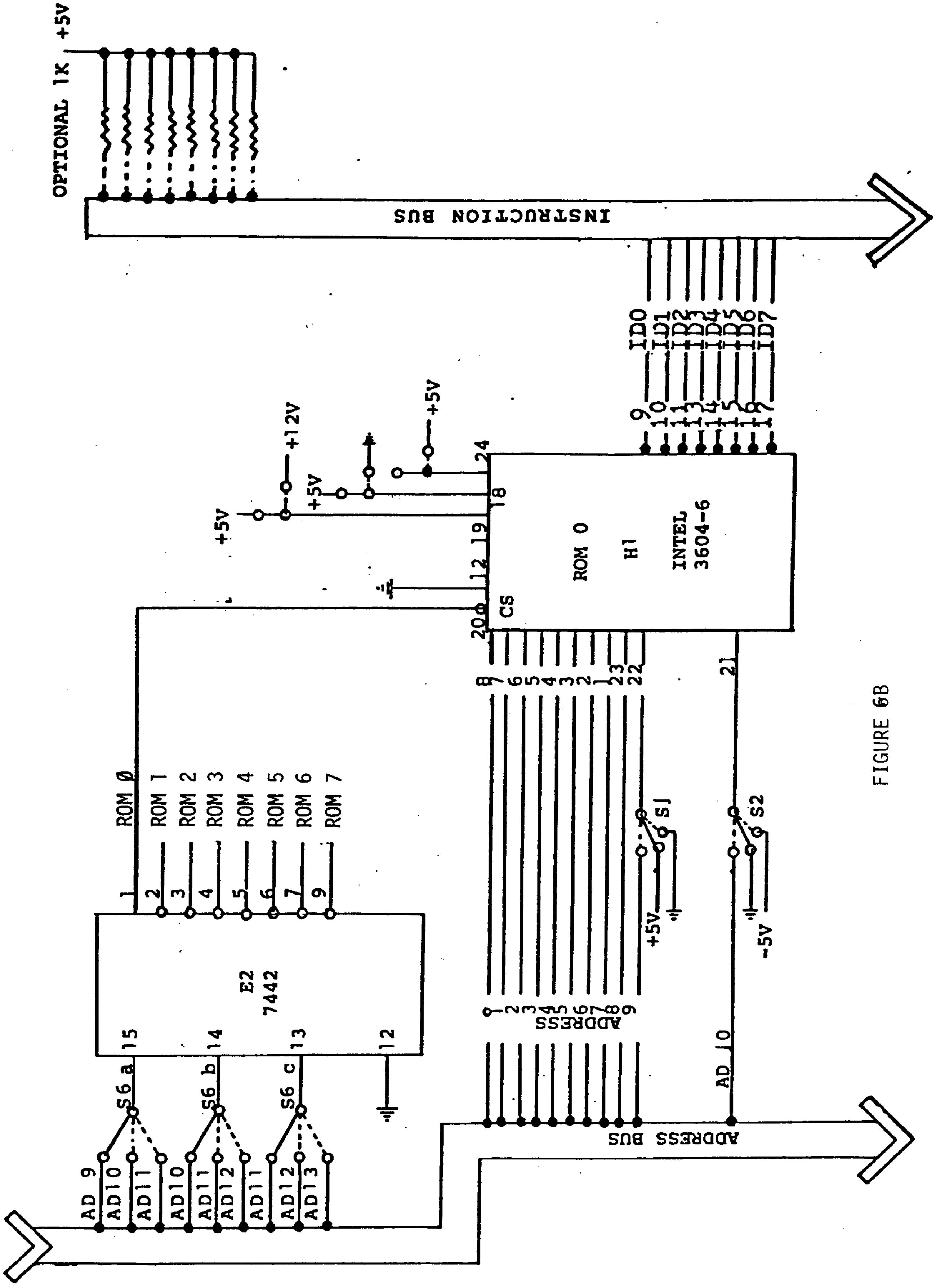
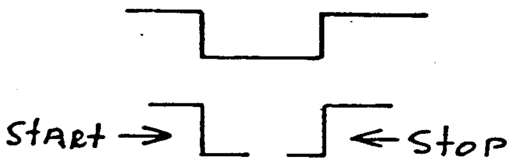

OPTIONAL 1K
+5V
INSTRUCTION BUS
ID0
ID1
ID2
ID3
ID4
ID5
ID6
ID7
9
10
11
13
14
15
16
17
+5V
+12V
+5V
+5V
24
18
19
12
CS
20
ROM 0
H1
INTEL
3604-6
8
7
6
5
4
3
2
1
23
22
21
ROM Ø
ROM 1
ROM 2
ROM 3
ROM 4
ROM 5
ROM 6
ROM 7
1
2
3
4
5
6
7
9
E2
7442
15
14
13
12
S6 a
S6 b
S6 c
AD 9
AD10
AD11
AD10
AD11
AD12
AD11
AD12
AD13
SJ
+5V
S2
-5V
0
1
2
3
4
5
6
7
8
9
ADDRESS
AD 10
ADDRESS BUS

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.

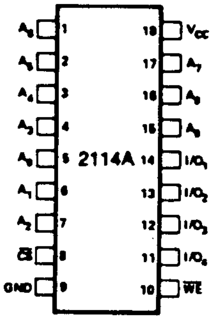
A simple program which will write a "0" and a "1" into alternate cells in the RAMs then compliment this pattern ("1" where there was a "0", and a "0" where there was a "1"). This forms a checkerboard pattern.

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

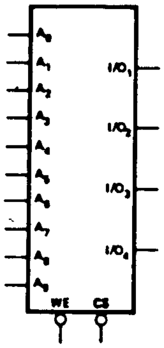
Assume we have software (a ROM) with a RAM TEST pattern in it, the Signature II can tell you which one of a multi-ram bank is defective.

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.

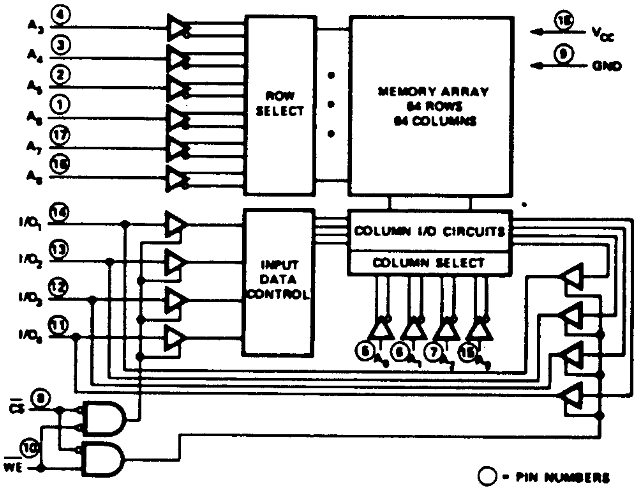
PIN CONFIGURATION



LOGIC SYMBOL



BLOCK DIAGRAM



PIN NAMES

A_0–A_9	ADDRESS INPUTS	V_{CC} POWER (+5V)
$\overline{WE}$	WRITE ENABLE	GND GROUND
$\overline{CS}$	CHIP SELECT	
I/O_1–I/O_4	DATA INPUT/OUTPUT	

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.

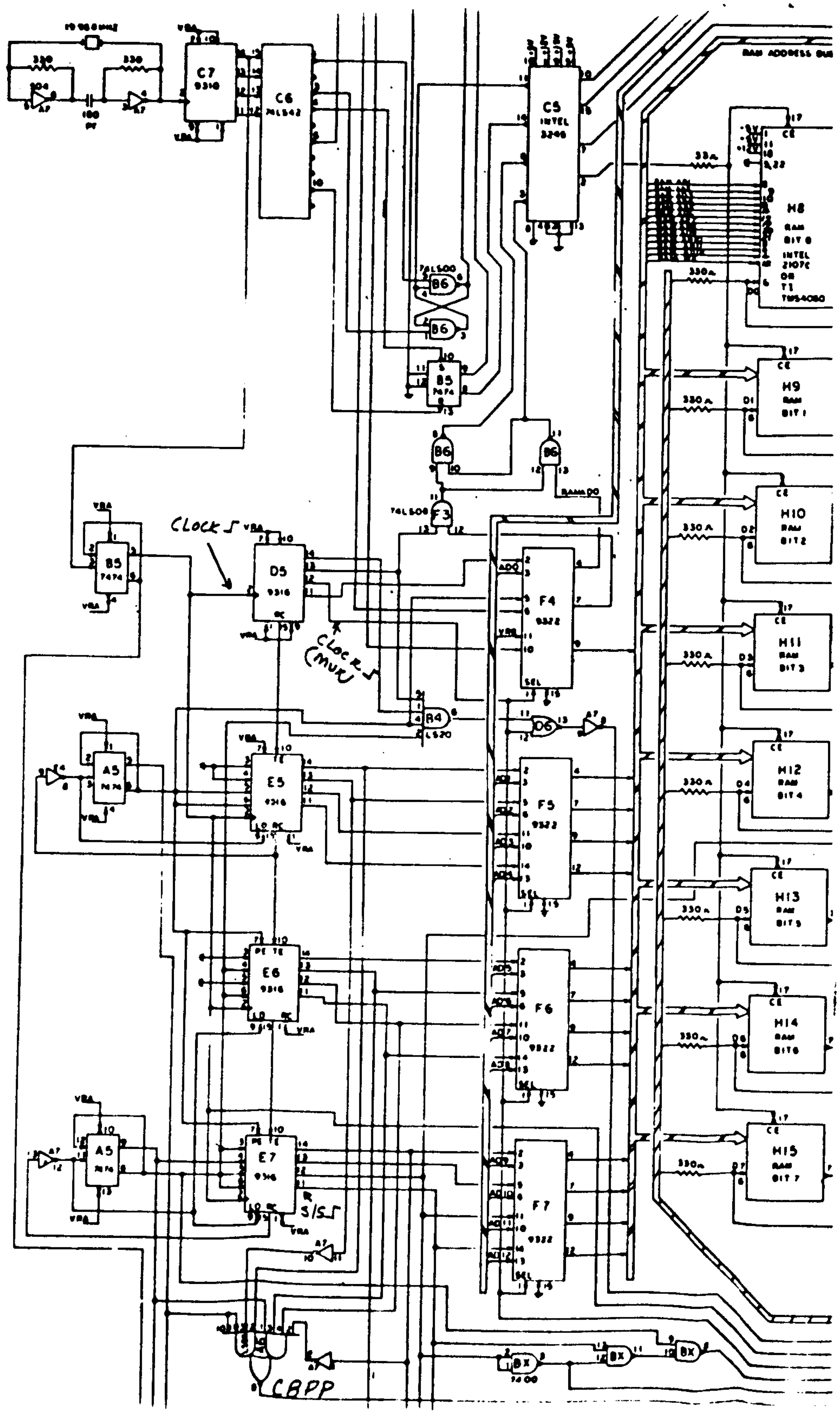
The schematic on page 22 shows one of the two inputs orginate at the CPU and the other from the counter chain. In other words data from two systems can appear at the mux output. This normally would cause the signatures to be unstable. The Signature II clock when in the rising _⌐ mode allows us to look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

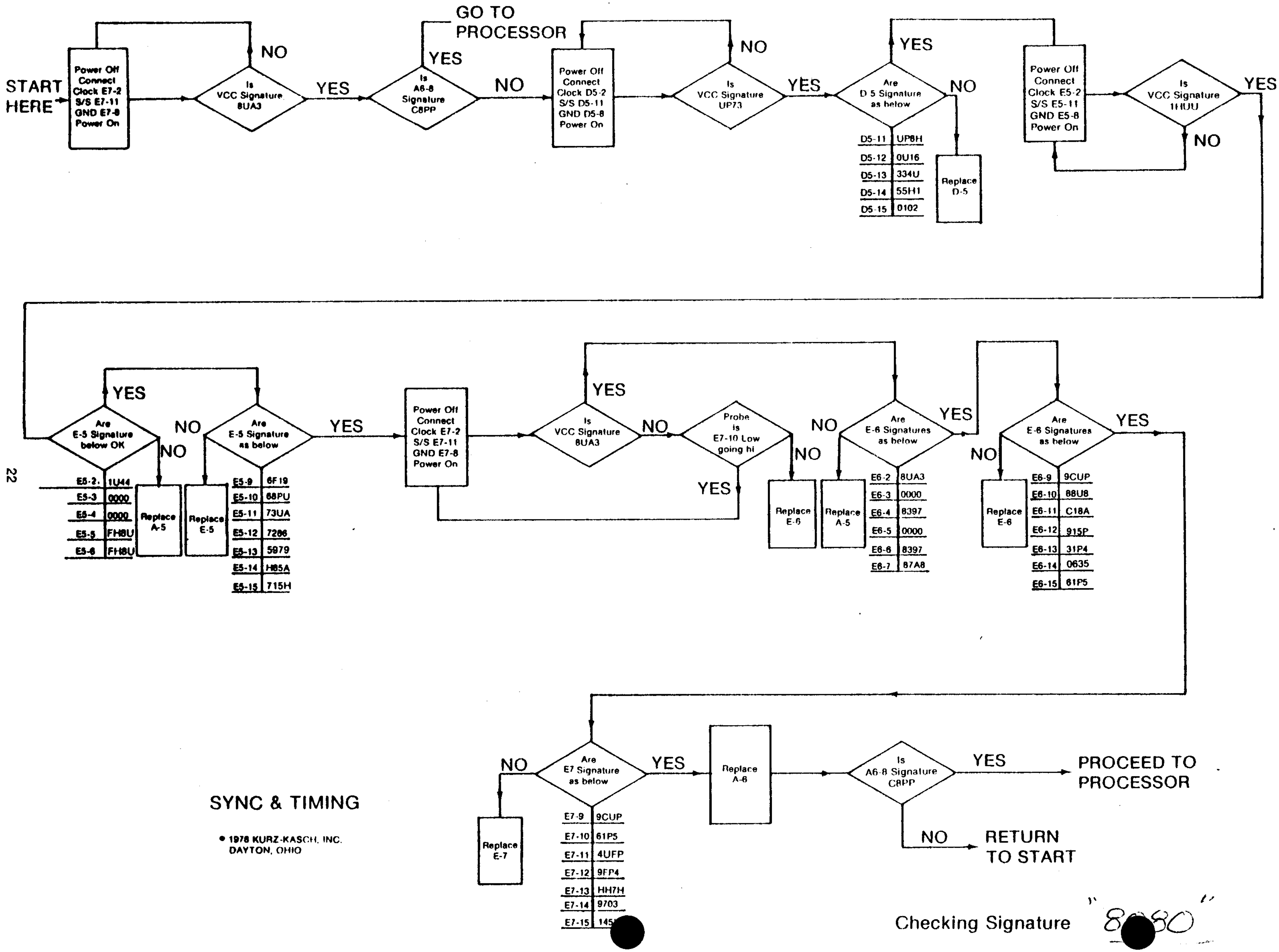
To test the multiplexes with the Signature II connect as follows and read the signatures:

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
9316
C6
74LS42
C5
INTEL
3246
RAM ADDRESS BUS
330
330
100
PF
VRA
74LS00
B6
B5
7474
F3
74LS08
B4
LS20
CLOCK ⎍
CLOCK ⎍
(MUX)
D5
9316
E5
9316
E6
9316
E7
9316
S/S ⎍
A5
7474
F4
9322
F5
9322
F6
9322
F7
9322
SEL
D6
A7
RAM DO
H8
RAM
BIT 0
INTEL
2107C
OR
TI
TMS4060
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
CE
330Ω
33Ω
BX
7400
CBPP


START HERE
Power Off Connect Clock E7-2 S/S E7-11 GND E7-8 Power On
Is VCC Signature 8UA3
NO
YES
Is A6-8 Signature C8PP
YES
GO TO PROCESSOR
NO
Power Off Connect Clock D5-2 S/S D5-11 GND D5-8 Power On
Is VCC Signature UP73
NO
YES
Are D 5 Signature as below
YES
NO
Replace D-5
Power Off Connect Clock E5-2 S/S E5-11 GND E5-8 Power On
Is VCC Signature 1HUU
YES
NO
Are E-5 Signature below OK
YES
NO
Replace A-5
NO
Replace E-5
Are E-5 Signature as below
YES
Power Off Connect Clock E7-2 S/S E7-11 GND E7-8 Power On
Is VCC Signature 8UA3
YES
NO
Probe is E7-10 Low going hi
YES
NO
Replace E-6
NO
Replace A-5
Are E-6 Signatures as below
YES
NO
Replace E-6
Are E-6 Signatures as below
YES
NO
Replace E-7
Are E7 Signature as below
YES
Replace A-6
Is A6-8 Signature C8PP
YES
PROCEED TO PROCESSOR
NO
RETURN TO START

D5-11	UP8H
D5-12	0U16
D5-13	334U
D5-14	55H1
D5-15	0102

E5-2.	1U44
E5-3	0000
E5-4	0000
E5-5	FH8U
E5-6	FH8U

E5-9	6F19
E5-10	68PU
E5-11	73UA
E5-12	7266
E5-13	5979
E5-14	H85A
E5-15	715H

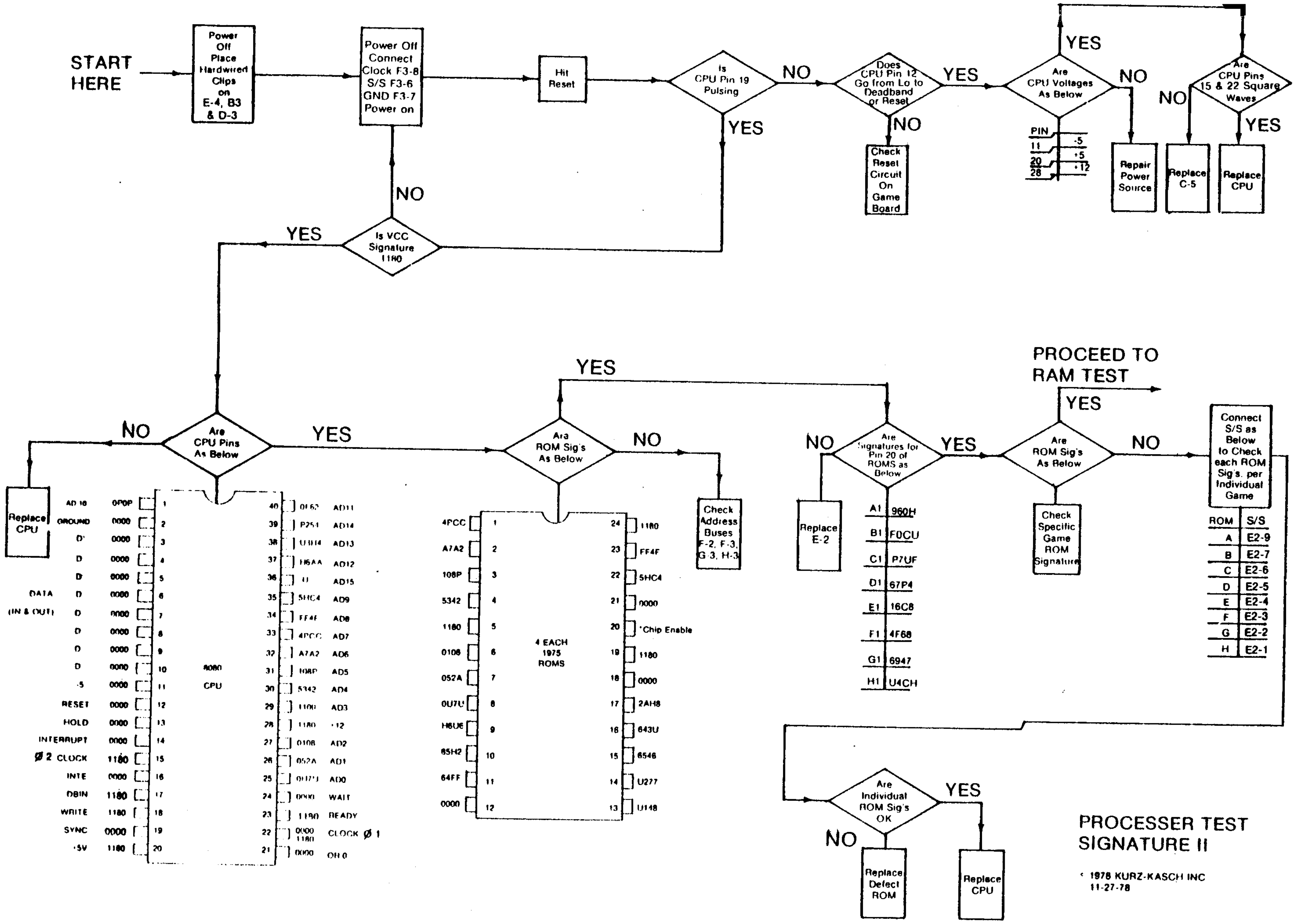
E6-2	8UA3
E6-3	0000
E6-4	8397
E6-5	0000
E6-6	8397
E6-7	87A8

E6-9	9CUP
E6-10	88U8
E6-11	C18A
E6-12	915P
E6-13	31P4
E6-14	0635
E6-15	61P5

E7-9	9CUP
E7-10	61P5
E7-11	4UFP
E7-12	9FP4
E7-13	HH7H
E7-14	9703
E7-15	145

SYNC & TIMING

Checking Signature "8080"



PROCESSER TEST
SIGNATURE II

© 1978 KURZ-KASCH INC
11-27-78

Checking Signature "8080"

Signature II

Kurz·Kasch, Inc.

KURZ-KASCH, INC., ELECTRONICS DIVISION
2271 Arbor Blvd.
Dayton, Ohio 45439

<u>CLOCK ENABLE--------SIGNATURE II MOD B</u>

The red lead on your pod is the "Clock Enable". This enable is selectable either active high (⎍) or low (⊔).

WARNING: The enable lead must be tied either high (⎍) or low (⊔) if not used.

This added feature of the Signature II allows a much more flexible use of available nodes for data entry.

The "Clock Enable" is just what it says it is---the red lead will enable or disable the clock input to the Signature II. It is in effect a 3rd control added to Start and Stop.

There is 1 bit of data entered through the data probe for each clock pulse between the Start and Stop. Now the number of data bits can be further controlled by the time the clock is enabled.

Kurz·Kasch, Inc.
ELECTRONICS DIVISION
2271 Arbor Blvd.
Dayton, Ohio 45439
Tel. (513) 299-0990

June 1, 1979

Dear Signature II Owner:

The Signature II Data probe has a "dual threshold pulse circuit*," incorporated. This feature is exclusive to the Kurz-Kasch, Inc., logic probe and the Signature II Data Probe. It will give you another dimension in trouble-shooting. It functions differently than our previous probes and everyone elses, too. It is included at no additional cost because we are constantly striving to make our product more useful to you.

You will notice a switch at the rear of the label. This switch in the "single" position allows your probe to operate as a "single threshold" probe (just like the old ones). When the switch is in the "dual" position, your probe becomes a "dual-threshold" probe.

To explain the operation of this feature consider Figure 1, and how all previous logic probes operated.

All probes prior to this unit will illuminate the pulse indicator when either threshold (0.8V or 2.4V) is crossed.

FIG. 1

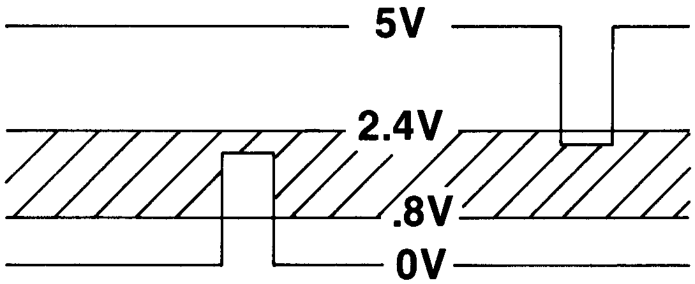
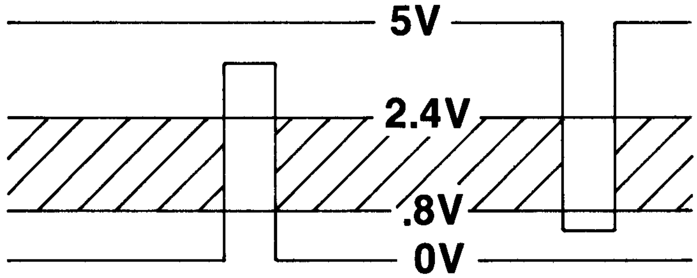


FIG.2



Now refer to Figure 2, to see how the new probes operate.

You must cross both thresholds (0.8V and 2.4V) to cause the pulse indicator to light. The obvious benefit to you, the pulse must be of a VALID logic level in order for the pulse lamp to illuminate. No longer can glitches or small pulses or noise cause the pulse indicator to light. Only VALID pulses will properly operate your probe and cause the pulse indicator to light.

The Dual Threshold Mode has another benefit, and that is the transition across both thresholds must be accomplished within 35 NS. A rise or fall time out of spec will be indicated by a lack of pulse light in the D.T. Mode. When you have a "0" and "1" indication and no pulse indicating then you know the transition time is greater than 35 NS. Since Cmos is generally much slower than this it should be tested in the "single" mode.

*Pat. No. 4110687

For permission request,

Write

KURZ-KASCH, INC., ELECTRONICS DIVISION
2271 Arbor Blvd.
Dayton, Ohio 45439

CONTENTS

I. — SPECIFICATIONS

SIGNATURE II

Data Probe

Indicators:	3 LEDS "0", "1", "P"
Duty Cycle:	DC to 100 MHZ
Pulse Stretch:	50 Ms
Threshold:	Single or Dual Selectable
Input Z:	"0">22k "1">83k
Tip Protection	150V, DC

Control Probe

Inputs:	Clock — White Start — Blue Stop — Yellow V. REF — Orange Ground — Black
Input Z:	>2 MEG For clock, start, stop
Input Protection:	150V, DC
Input FREQ:	>50 MHZ each channel
Logic Levels:	TTL — CMOS/TTL Compatible
Pulse Edge:	Switch Selectable All Channels
Enable:	Clock Enable, Rising or Falling Edge

Data Processor

Clock:	Up to 20 MHZ
Data:	Up to clock speed
Set-up Time:	15NS (data required to be valid at least 15NS before selected clock edge)
Hold Time:	0 NS
Accuracy:	100% probability on validating a data system $1.00 - \frac{\text{BIT ERRORS}}{\text{Bits Stream Length}} =$ probability of detecting a faulty data stream
Min Gate:	1 clock cycle
Min time between Last Stop & Next Start:	1 clock cycle

Power

120V 50/60 HZ @ 11.5VA
220V 50/60 HZ @ 9.5VA

Weight

Net: 5 lbs.

Dimensions

Case Closed: 5.5" H x 9.0" W x 9.0" D (max)

Environmental:

Temperature:	0 to 55 degrees C
Relative Humidity:	95% @ 40 degrees C
Case:	Drip proof

II. — SIGNATURE II
UNDERSTANDING ITS OPERATION

ROMs, RAMs, CPU's, data buses, all have one thing in common — they are difficult for the average technician to troubleshoot. Kurz-Kasch, Electronic Division, developed the Signature II system to solve this problem.

Understanding how it works is essential before using the Signature II system. (Figure 1 shows the system and its controls).

Main Logic Unit

The main logic unit contains the power supplies; on-off switch, 5V power LED, hexidecimal readouts, gate LED, error LED, self test button, glitch switch, and hold switch (model A only). (Figure 2 shows the circuits contained in the main unit).

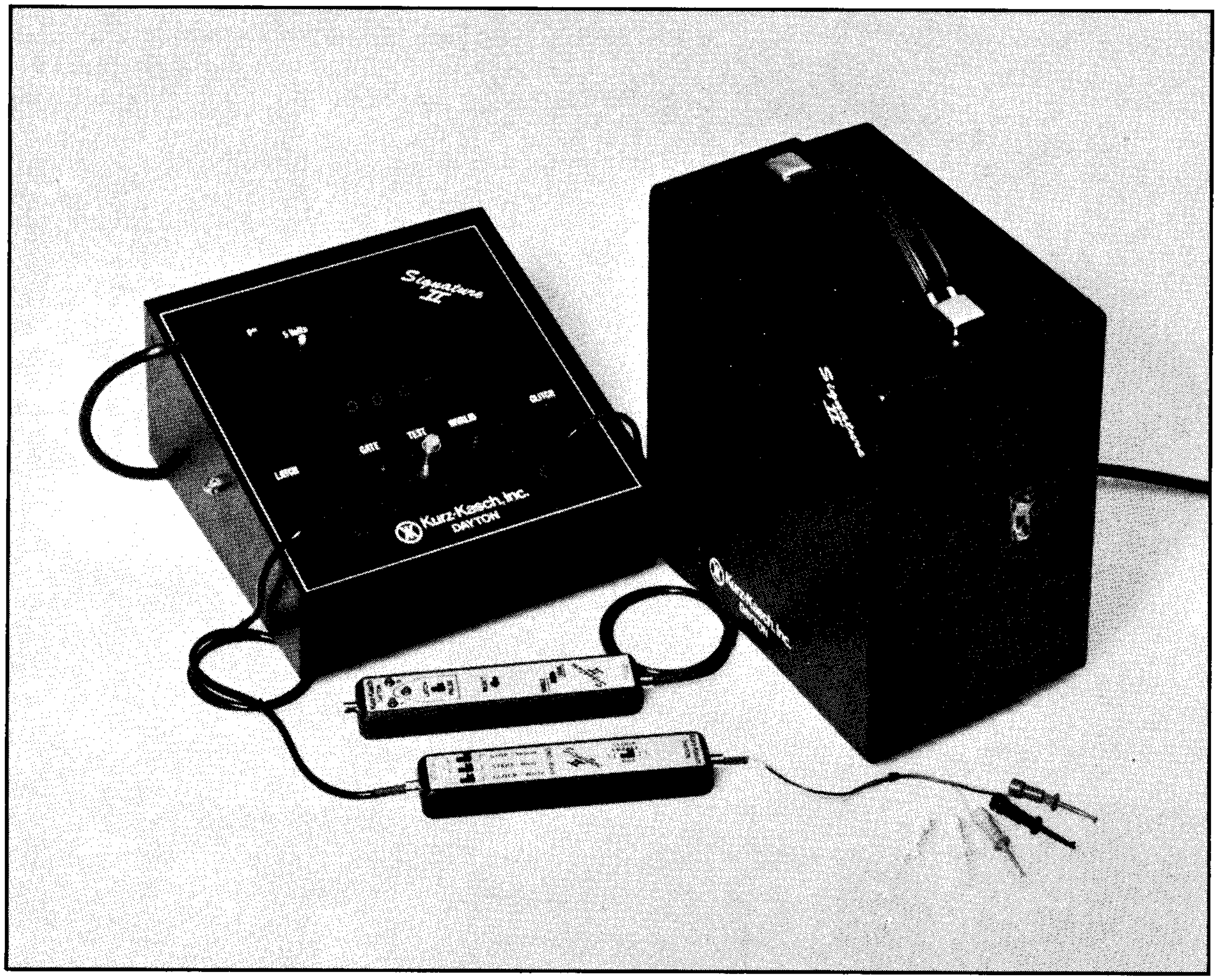


Figure 1

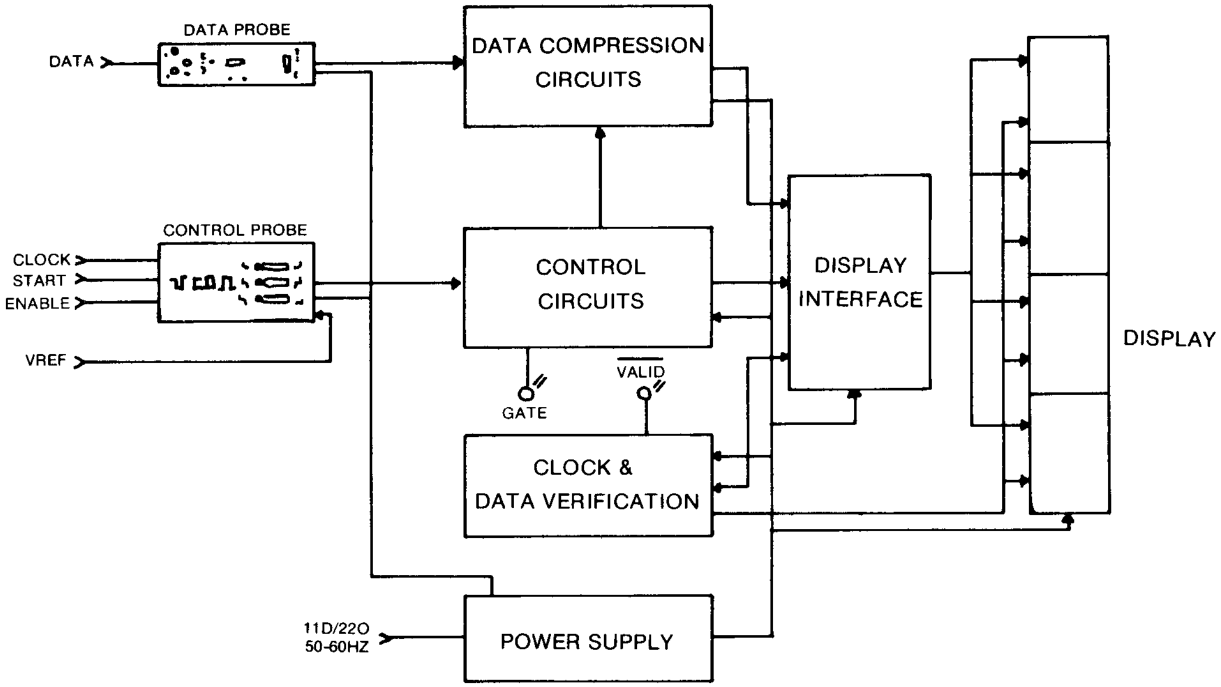
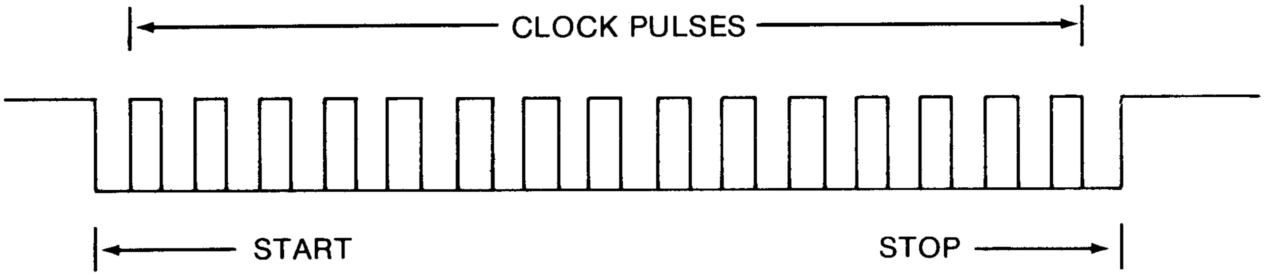


Figure 2



Here's how it works:

The control probe is connected to the start/stop and clock sources on the board to be tested. The start/stop pulse creates a window which enables internal circuits to accept clock pulses. These clock pulses allow acceptance of data bits (from the data probe) by the data compression circuits. A clock enable allows clock pulses to be gated by the circuit under test.

In the above sketch the start/stop control allows 15 clock pulses to be accepted. (In reality the system can operate with from 1 to in excess of 1 megabits of clock).

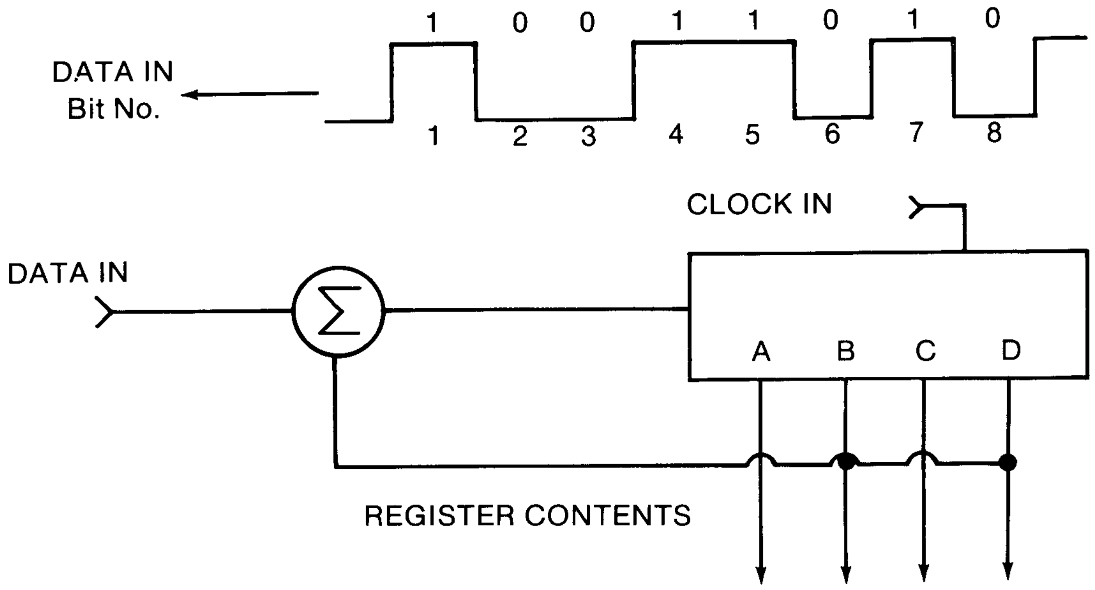
These 15 clock pulses allow 15 data bits to be clocked into the compression circuits. The compression circuits only accept data between the start/stop pulses. This being the case it can be said that 1 data bit or as many as a megabit can be clocked in. These are compressed to 16 bits. These 16 bits are further compressed to 4 bits. The bits are used to address a decoder ROM. The ROM Data output is further decoded to give 4 four bit hexidecimal words. The 4 words have **NO SIGNIFICANCE AS TO THE DATA WORDS CLOCKED IN.** In other words, no displayed word is the computer word or word the Signature II system saw.

To simplify the explanation, lets look at a 4 bit compression circuit. We will clock in 8 data bits and see what is left in the register to display. (Figure 3).

As you can see, after we clocked in 8 bits, two words 1001 (9) & 1010 (A), there remained 4 bits 0011 the word 3.

The only difference between the above example and the Signature II is the quantity of bits of both clock and data that are processed.

The modified hexidecimal words used by the Signature II are:
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, C, F, H, P, & U.



DATA	CLOCK PULSE #	A	B	C	D
—	0	0	0	0	0
1	1	1	0	0	0
0	2	0	1	0	0
0	3	1	0	1	0
1	4	0	0	0	1
1	5	0	1	0	0
0	6	1	0	1	0
1	7	1	0	0	1
0	8	1	1	0	0

Figure 3

There are a couple of other circuits which are interesting. One is the Gate. The Gate circuit LED indicates that clock and start/stop pulses ARE VALID, and that the clock circuit is enabled.

The second circuit is the DATA INVALID. This is a bit more complicated, but easily understood. It works this way — When the stop pulse is received by the compression circuits all activity within these circuits cease. The compressed data stream, now 4 bits in length are stored in a RAM and at the same time they are decoded and displayed. When a new start pulse is received, the data stream is again compressed until another stop pulse is received. At this time the same (or new words) are displayed and moved to storage. The first words and the last words are taken from storage and compared. If they are in any way different, the INVALID LED will flash. Remember that 1 data bit error will cause the 4 displayed words to be entirely different. When the glitch switch is activated a single or multiple bit error will cause the INVALID LED to remain on.

Control Probe

The control probe consists of 4 level detectors automatically toggling TTL or MOS compatible levels set by the internal circuits.

Each of the 4 channels will accept pulses as fast as 20 NS.

Leading edge or trailing edge pulse selection located on the control probe, can be made by moving the appropriate selector switch for either start, stop, clock, or enable.

Data Probe

The data probe, which is a part of Signature II, is a 10 NS logic probe. This is a dual threshold probe — has memory (latch) and independent LEDS for "0", "1", & "P". All three LEDS work full time and display hi's, lows, & pulses from DC to 100 MHZ. This is truly a 10 NS probe and can be used independently if the control probe ground is connected to logic ground.

The reset button on the probe is to reset the registers, latches and displays in the main logic unit to all Zero's.

Duty cycle is displayed by the LEDS from DC to 100 MHZ. Duty cycle is the relative intensity of the "0" & "1" LEDS. Equal intensity is a 50% duty cycle. A "0" or "1" LED will begin to glow at a 20:1 duty cycle. You can see 10 NS pulses spaced 200 NS apart. At a 20:1 duty cycle one LED will be full brilliance and the other one will just begin to glow.

Self Test

To self test the Signature II, connect clock, start, stop and ground to the proper places on the logic board to be tested per software. **NOTE:** "Gate" LED must be on.

Touch data probe tip to ground — Signature II should read **0000**. Touch data probe tip to VCC. Signature II should read the VCC notation on software of board to be tested. While touching VCC push "TEST BUTTON" and readout should be **UUUU**.

During all tests the INVALID LED should only flash on touch, unless the Signature is unstable or remain on if glitch switch is on.

If Signature II passes all the above tests you can be sure the system is working.

Some Signature Indications & What They Can Mean:

A 0000 Signature at a logic node means a short to ground or a stuck at "0" device.

A VCC Signature at a logic node means a short to VCC or a stuck at "1" device.

Two address or data buses with the same but incorrect Signature indicates these two lines are shorted together.

Unstable Signatures where they should be stable, could be improper start, stop, clock or ground connections. "Dirty" power supplies or ground loops can cause instability.

Whenever wrong Signatures or unstable ones are observed, recheck the 4 switches on the pod. They should be in the position indicated by the software.

Remember Murphy's Law — "The least suspected error is probably the fault".

With the threshold switch in "Dual" mode absence of a pulse light on the probe indicates a faulty or tri-state mode if the Signature is other than 0000 or VCC.

III. — MAINTENANCE

Signature II is a very high performance piece of equipment. For this reason the original configuration, both electrical and mechanical, must be preserved.

Should a failure occur it would be best to return the Signature II to the factory. If this is not practical, parts replacement in the field can be made. There are a few components which affect the operation and these must be replaced with factory produced or selected ones. These critical components are listed below.

Under no circumstance should the cables be either shortened or lengthened. All other parts are standard and should be available from your normal supplier.

SYMBOL	DESCRIPTION	KK PART NO.
U-1	IC	0500016
U-4	IC	0500023
U-10, 19	IC	0500017
U-12	IC	0500018
U-14	IC	0500019
U-17	IC	0500020
None	Cable (POD)	0860014
	Cable (PROBE)	0860014
T1	Transformer	0570010
None	Operating Instructions	1670038A

IV. — ASYNCHRONOUS COUNTERS

Any system which can be made repetitive is a candidate for Signature II.

Bit errors, glitches, or other digital ills can be isolated with Signature II in a minimum of time using technicians without extensive training. Usually, an hour or less to familiarize oneself with this new system is all that is required. The technician or engineer can isolate the failure mode without the use of schematics, flow charts, or look-up tables.

The only requirement for use of Signature II is that the data be repetitive. A system designed with Signature II in mind is obviously more flexible but almost all existing systems are adaptable.

An asynchronous counter or system is one in which the logic circuits changes states at arbitrary instants of time determined by the propagation delays through the system components.

The main thrust of this paper addresses itself to the many systems already in existence and crying for solutions to laboratory and field problems.

If one can understand the methods of causing repetitive bit streams in existing equipment then new designs will come easily.

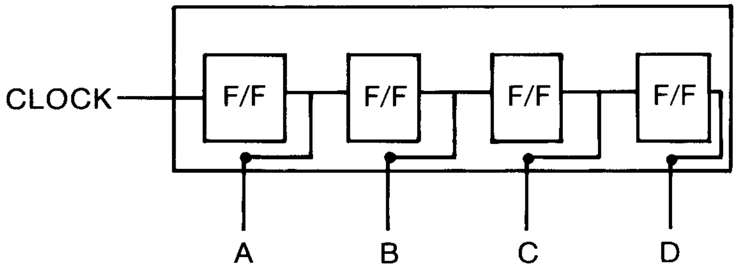


FIGURE 1

An asynchronous type counter is one shown in Figure 1. These are typically the 7490, 7492, 7493 types wherein the first flip-flop is clocked which in turn clocks flip-flop #2, which clocks #3, etc. A typical 7493 which is clocking can have a "D" (most significant bit) output delayed greater than the width of the clock pulse. For this reason, asynchronous counter chains should be checked not over two counters at a time.

How do we check them? A simple rule applied here and in all future Signature II set-ups apply.

1. Signature II clock-to-counter clock input.
2. Start/Stop to **most significant bit** of counter **that is used.**

We emphasize "most significant that is used" because an unused output may be noisy or not terminated (no pull-up). Eratic or unstable signatures can result.

To obtain signatures or to test an asynchronous counter, Figure 2 shows the connections and signatures to be expected.

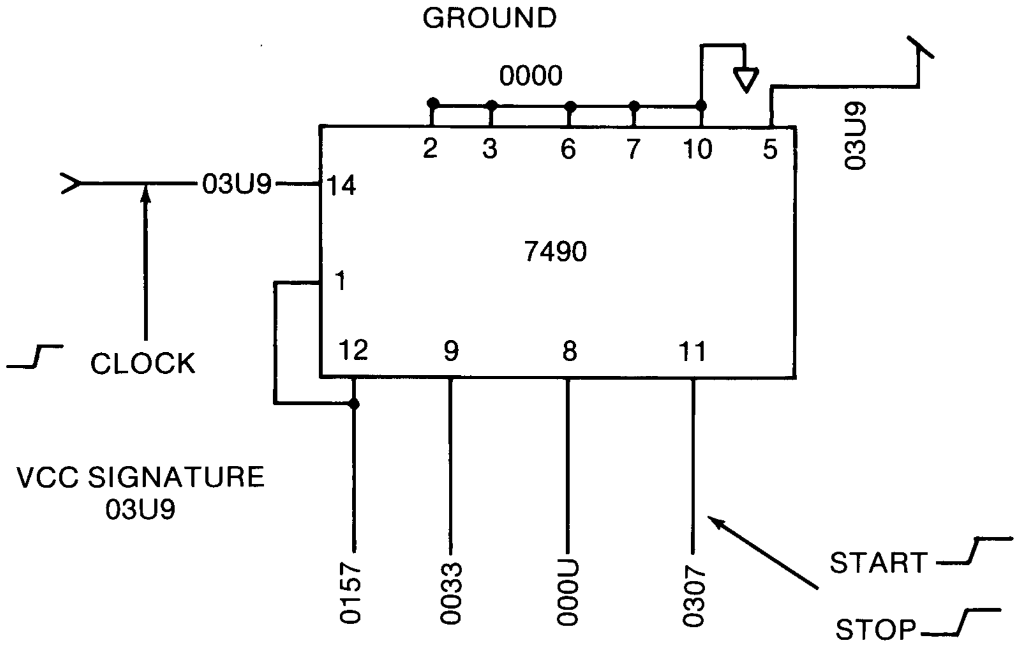


FIGURE 2

As shown in Figure 2, the Signature II clock is connected to the counter clock input, start and stop to the most significant bit of the counter, pin 11, "D" output.

When obtaining signatures to be later used, the first signature to be taken after the Signature II connections are determined is VCC. Touch the data probe tip to VCC. Recording this signature along with the connection information will insure later connections are correct.

SUMMARY

To obtain signatures from asynchronous counter systems follow these rules:

1. Clock input of Signature II connected to clock input of counter under test.
2. Start and stop of Signature II to most significant bit that is terminated. Do not have clock & start/stop separated by more than 2 IC's.
3. Ground of Signature II at point as close to clock input as is practical.
4. Select and record pulse edge used for clock, start, stop, & enable.
5. Record VCC signature.
6. Record signatures at counter or any point in system fed by this counter.

V. — SYNCHRONOUS COUNTERS

A synchronous counter or system is one in which the logic circuits changes state only at certain instances determined by a common clock.

Figure 3 is a typical synchronous counter. A 74161 or 9316 is an example of this type of counter.

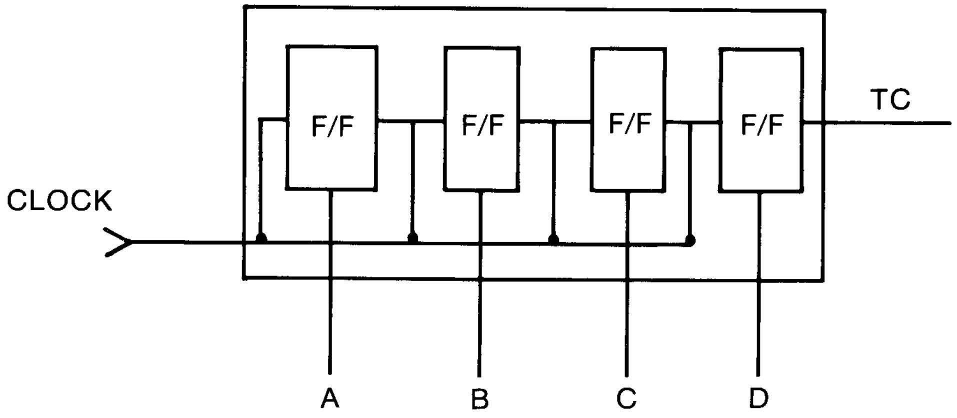


FIGURE 3

The significant difference between an asynchronous and a synchronous counter is the clocking. In Figure 1, in an asynchronous counter, the clock is applied to the first flip-flop which in turn clocks #2, etc. Figure 3 shows the clock being applied simultaneously to each of the flip-flops (synchronous).

Signature II connections to this type of counter or counter chain is no different than with an asynchronous counter. Clock goes to Signature II clock input and the most significant bit — That Is Used — is the start/stop input to Signature II. The connections and signatures to be expected for a single 9316 are shown in Figure 4.

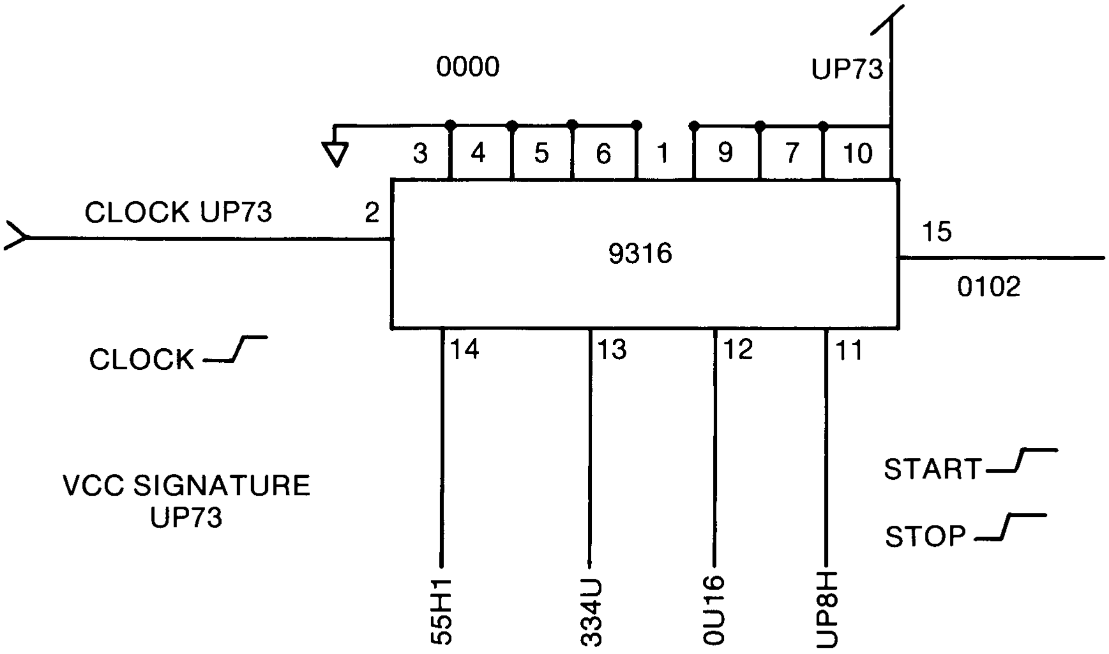


FIGURE 4

Again as with the asynchronous counter and with any initial signature recording, the first signature to be taken and recorded is VCC. This assures that the counter is operating and that the start, stop, and clock inputs to the Signature II are properly connected.

As you will have noticed nothing has been said about clock frequency. As long as the clock is a relatively clean square wave the Signature II is independent of clock frequency within the clock specifications. The limiting factor generally will be the logic speed.

Before we leave the counters let us look at a counter chain consisting of 3 each 9316's. The schematic in Figure 5 is the one used in the final test for Signature II. The clock is 15 MHZ.

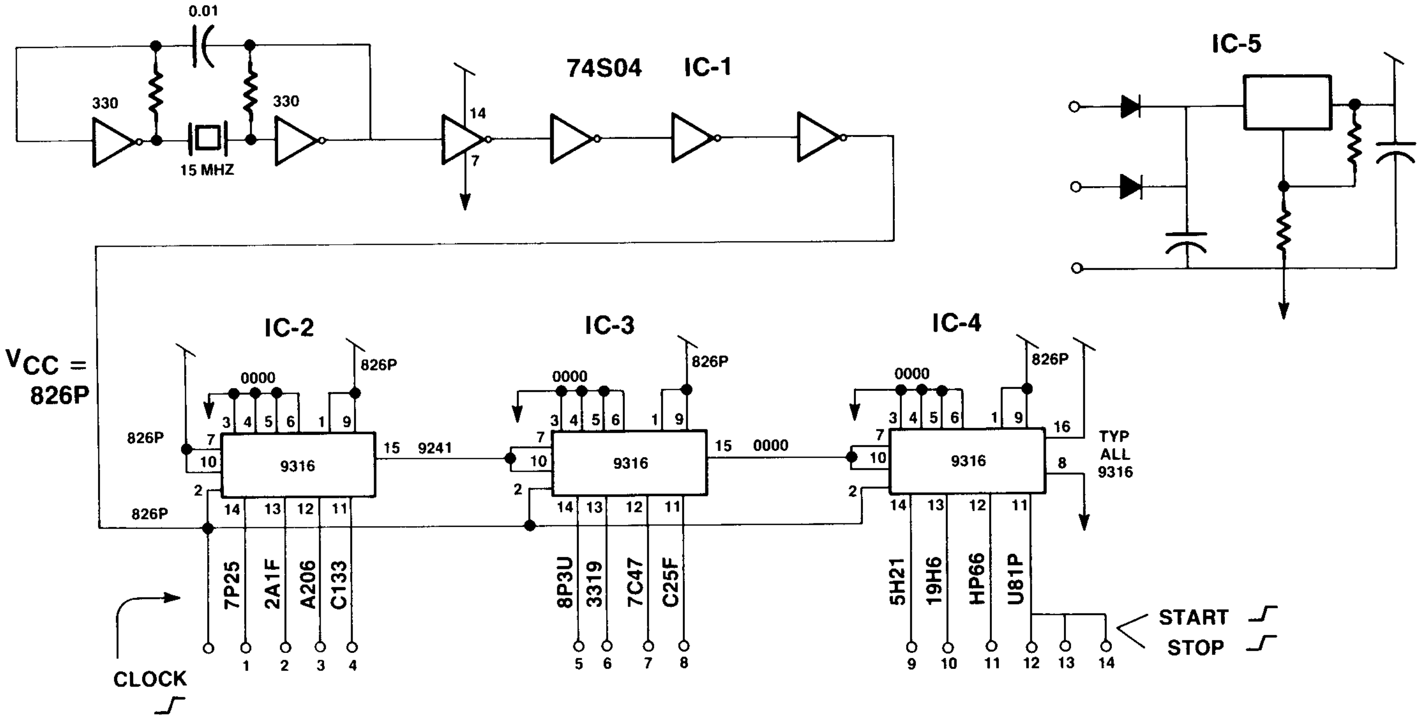


FIGURE 5

Clock is the 15 MHZ square wave from the 74S04 connected to all pin 2's of the 9316's. Start/stop is connected to pin 11 of IC-4. This is the most significant bit of the counter chain.

VI. — VCC SIGNATURES FOR COUNTER CHAIN

There is another technique, other than shown in sections I and II, to check a counter chain. Figure 6 shows how this is done and the signatures to be expected.

Clock input to the Signature II is the same — first counter or system clock. The data probe tip is connected to VCC and the start/stop inputs are connected together using a probe shown in Figure 7. The new probe tip (Figure 7) then is touched to the counter outputs (pins 14, 13, 12, 11, & 15). The VCC Signatures will be as shown in Figure 6.

This system is very useful if a counter output is stuck high, low, or to another line on the logic board.

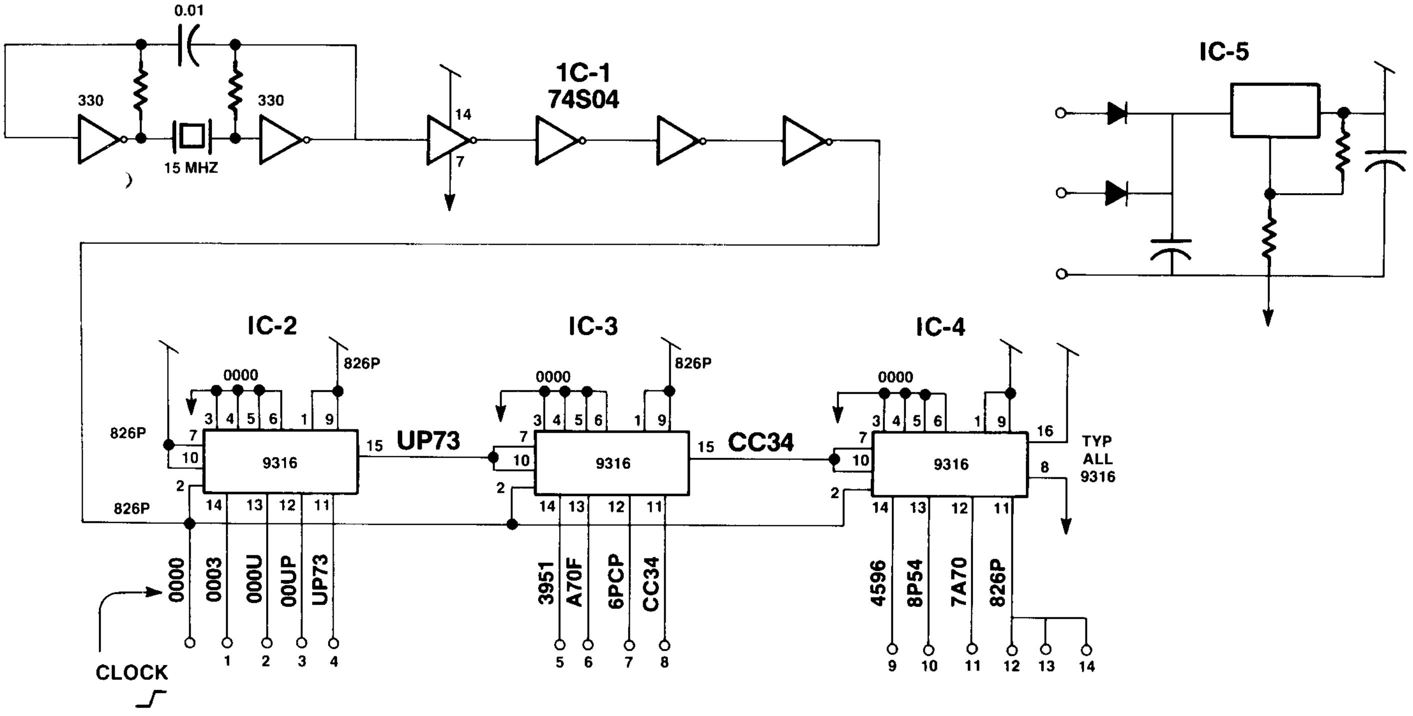


FIGURE 6

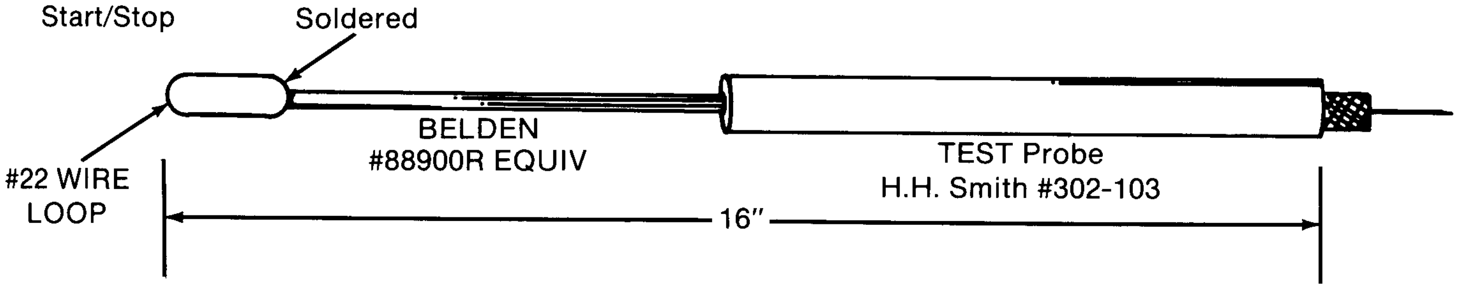


FIGURE 7

VII. — PROCESSORS (CPU)

If a CPU has an instruction forced into its data bus called a "NOP" (No operation) the CPU will become a 16 bit synchronous counter. The 16 counter outputs are the address lines A_0 thru A_{15}.

The instruction sets for the various CPU's will give the machine language for the NOP. Here are the ones for the most popular CPU's:

CPU	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
6502	0	1	0	1	0	1	1	1
6800	1	0	0	0	0	0	0	0
Z-80	0	0	0	0	0	0	0	0
8080	0	0	0	0	0	0	0	0

This NOP instruction causes the program counter (PC) to be advanced (incremented) 1 count each instruction fetch.

If a processor system were designed for Signature II the system would look something like Figure 8.

In computer terminology the largest numbered bus lines are the most significant ie: D^7 is the most significant of D^0 to D_7, A_{15} is the most significant of the address lines A_0 to A_{15}.

As in Figure 8 we can cause a processor to "free-run" or to act as a 16 bit counter by use of an 8 pole double throw switch. The same results could be obtained by installing an IC socket in the data bus. A header could be inserted with either a wired feed through or a wired NOP instruction.

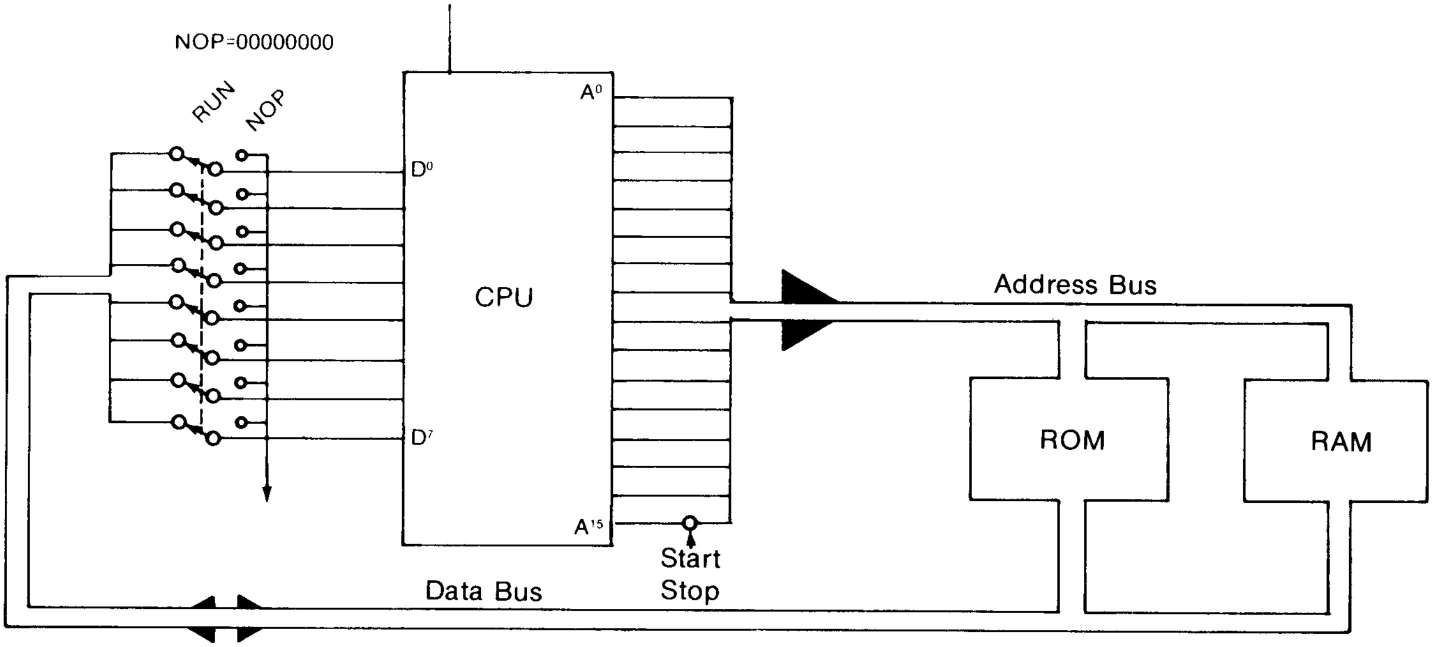


FIGURE 8

Unfortunately all bus oriented systems are not designed for Signature II systems. Let this not deter you. There are ways to accomplish the task.

1. NOP test fixture. (See Figure)
2. Where tri-state bus drivers are used the data bus can be opened by forcing the enable input to the proper logic level.

For example a 74LS245 pin 19 is the "ENABLE G". This means that pin 19 must be low for the drivers to operate. If pin 19 is taken high then the data bus is effectively "cut loose" from the CPU.

A word of warning — NO LOGIC GATE OUTPUT CAN BE TIED HIGH. The enable comes from some gate output. Since this input cannot be tied high check the source of this logic level. The chances are good that there is an invertor feeding this enable line. If it is, then the invertor (or NAND/or NOR) input can be pulled low making the "ENABLE G" go high.

Once the buffer or bus driver has been forced to the isolation mode then a test clip such as the Kurz Kasch TC-590-89 or one of the AP Product "TC" series is placed on the bus driver or drivers and the CPU side of the bus can be hardwired for the proper NOP instruction.

A WORD OF WARNING

NEVER POWER UP A LOGIC BOARD WITH HARDWIRED TEST CLIPS ON THE DRIVERS WITHOUT THE DRIVERS FORCED TO THE ISOLATION STATE.

THE INOP (INTELLIGENT NOP): is automatically capable of exercising RAM and/or I/O ports depending upon which of the eight internal programs selected. After the completion of the program, the device automatically switches into the "NOP MODE" of operation enabling the user to take signatures on the target board including RAM Signatures.*

PROCESSOR: Target Type

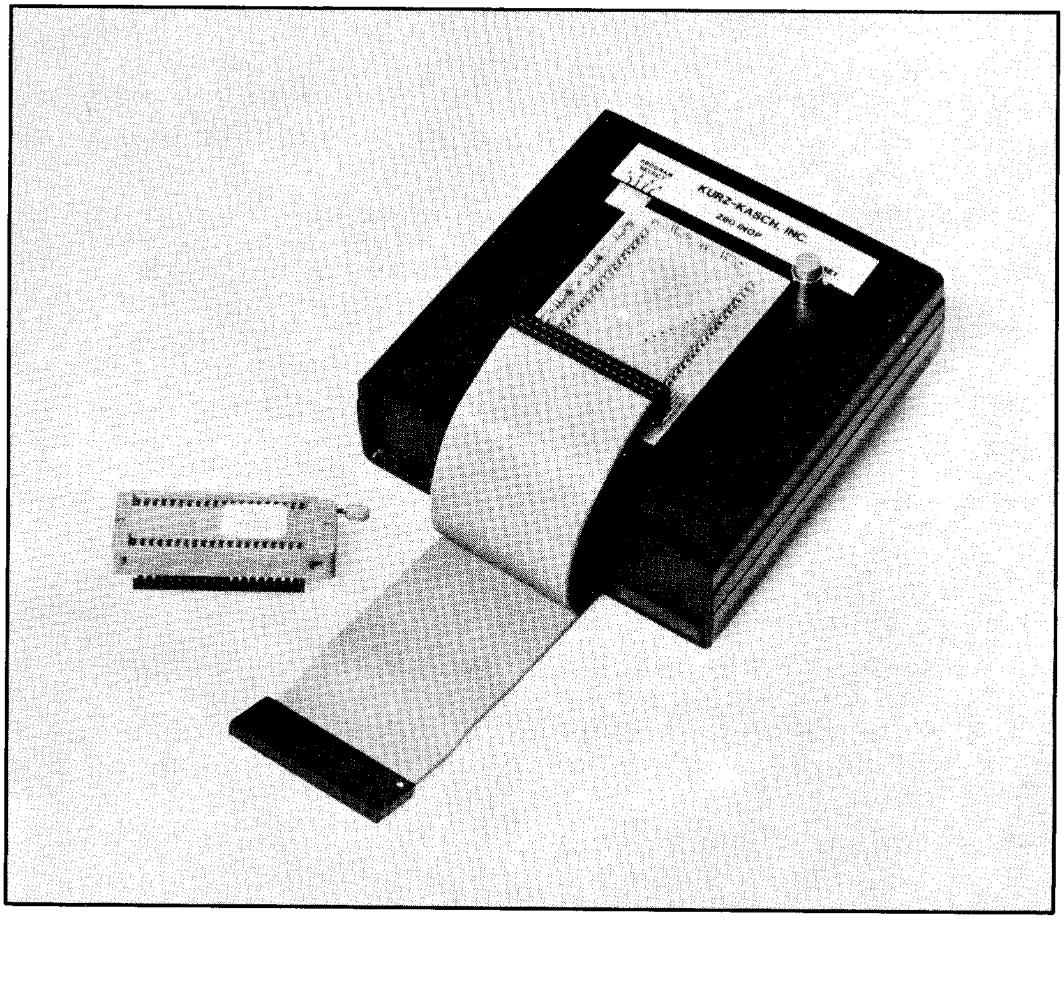
CLOCK: From Target Board

INPUTS: Buffered

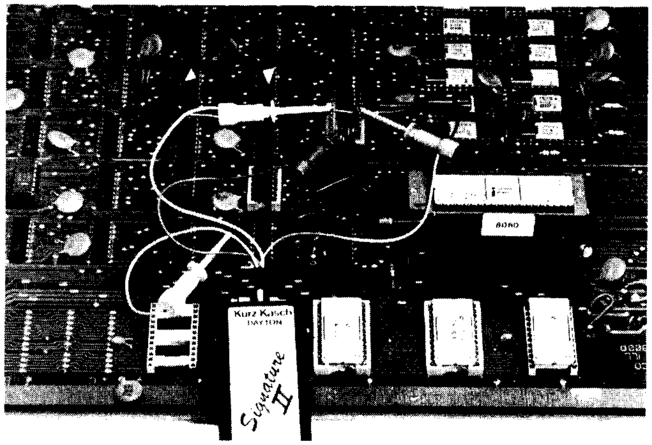
OUTPUTS: BUFFERED

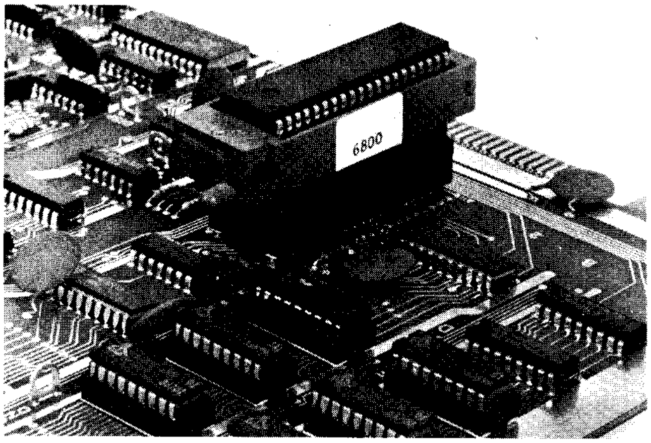
POWER
REQUIRED: 5V @500 MA max (includes processor)

*To exercise I/O ports, the user may program the internal 2716, in the space provided, to his own hardware configuration.



The second way to isolate the CPU data bus from the outside world is for a NOP test fixture be made. Figures 11 & 12 show how this can be done.





CLOCK FOR CPU

We now know how to cause the CPU to become a counter and know to connect start/stop to the highest order address line. (Most significant bit of the CPU counter A-15). Be sure to use the highest order address line that is used.

The clock to be used for the clock input to the Signature II is a bit more difficult to select. With the NOP fixture, clock is designated in Fig. 14 for the four CPU's most commonly used.

The 8080 uses a 2 phase clock neither phase alone is suitable for the system clock. From the timing diagrams for the 8080 we find the address bus and the data bus are both stable when Ø (phase 1) and sync (pin 19 of the CPU) are both high. If we AND both Ø & sync we will have what we need. Some processor systems perform this AND function as the **system time base.** If it is not already done for you, an AND gate can be added to a NOP test fixture or a 74LS08 could be piggy-backed on an IC that has either Ø or sync as an input or output.

The 6800 CPU, pin 37, Ø 2 is stable for the Signature II clock if the trailing edge of the clock is used.

For Z 80 use pin 21, RD, leading edge for clock and start/stop trailing edge for addresses.

The 6502 clock Ø 2 is used as clock input to Signature II.

For other microprocessors consult the CPU timing diagrams or contact Kurz-Kasch.

Since the NOP (PC+1) repetitively runs through all possible addresses a signature from each address line can be obtained. These addresses will follow paths throughout the system. In other words A12 signature at the processor can be traced with the data probe from the processor to its termination, including all branches. ROMS, RAMS, MUX, IO'S, etc. can be tested both inputs and outputs as long as they are addressed by the CPU.

A known good board or system must be probed and VCC signature along with signatures at chosen points be recorded. All future checks on like boards or systems must produce the same signatures. A different signature is indicative of a failure node. Figure shows signatures for the four most common CPU's.

CPU NOP FIXTURE CONNECTIONS

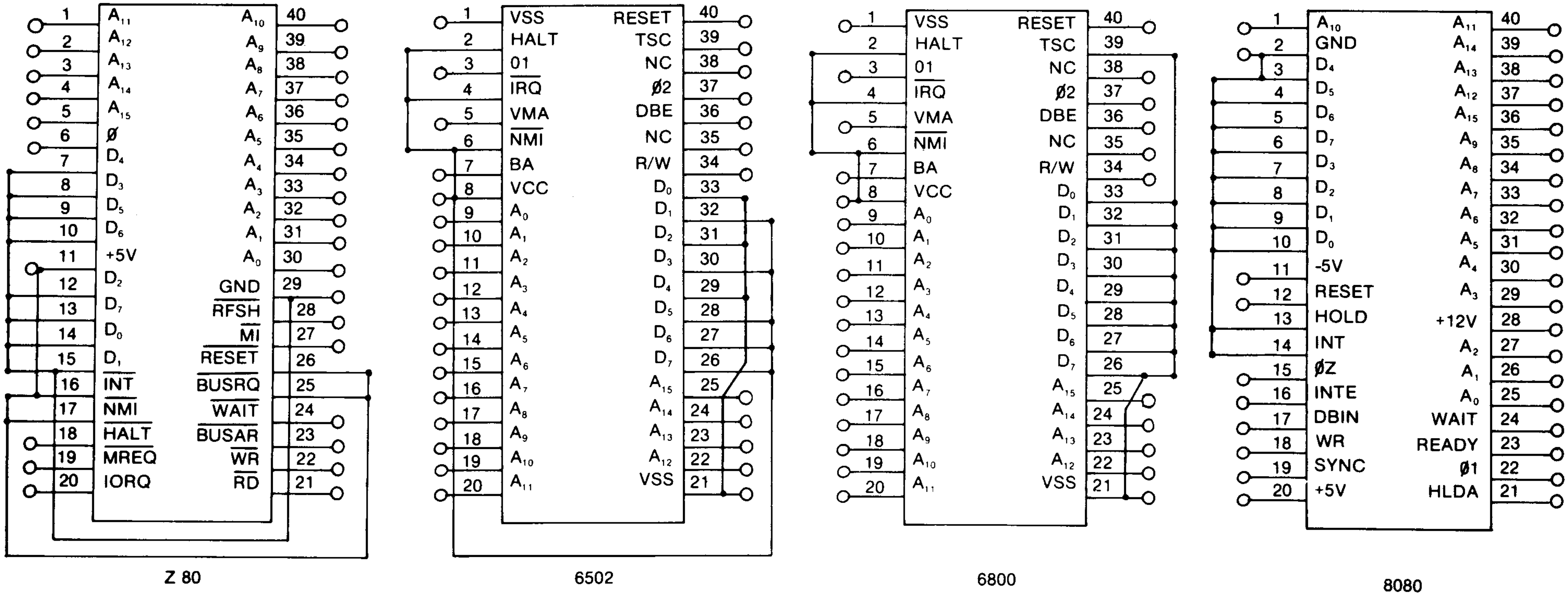


FIGURE 13

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.

NOP FIXTURE SIGNATURES
FOR PROCESSORS

PROCESSOR	Z-80		8080	6502A	6502		6800
CLOCK	PIN 21		PIN 17	PIN 37	PIN 37		PIN 37
START/STOP	5		36	25	25		25
GROUND	29		2	21	21		21
A0	30	UUUU	25	9	9	UUUU	9
A1	31	5555	26	10	10	FFFF	10
A2	32	CCCC	27	11	11	8484	11
A3	33	7F7F	29	12	12	P763	12
A4	34	5H21	30	13	13	1U5P	13
A5	35	0AFA	31	14	14	0356	14
A6	36	UPFH	32	15	15	U759	15
A7	37	52F8	33	16	16	6F9A	16
A8	38	HC89	34	17	17	7791	17
A9	39	2H70	35	18	18	6321	18
A10	40	HPP0	1	19	19	37C5	19
A11	1	1293	40	20	20	6U28	20
A12	2	HAP7	37	22	22	4FCA	22
A13	3	3C96	38	23	23	4868	23
A14	4	3827	39	24	24	9UP1	24
A15	5	755P	36	25	25	0002	25
VCC		0001				0003	

FIGURE 14

VIII. — ROM TESTS

Once the processor is in the NOP condition the address bus starts with all address lines at "0" at reset. The 16 addresses count binarily to all "1's". Any ROM on the address bus will have all its addresses exercised repetitively. ROM data will be output in a serial stream on each of the ROM data lines. A signature will be obtained for each data line and will be repeatable. Any deviation from the signature obtained from a good ROM indicates a defective device.

Chip enables, in multiple ROM banks are extremely useful in isolating which device is outputting bad data.

The system stable clock is used for Signature II clock. Start/Stop comes from the chip enable pin of the particular ROM you wish to check.

If the particular chip enable ($\overline{CE}$) is an active low then that ROM will only output data during the time this input is low. If you visualize the $\overline{CE}$ waveform it becomes clear that the Signature II should begin to take data on the falling edge of this line and cease to accept data on the rising edge. Conversely, if the CE is an active high then select the rising edge for start and trailing edge for stop.

ROM signatures should be taken collectively (clock-system time base, start & stop on most significant address line used). Any one ROM which is defective will cause the collective signatures to be wrong. Should this occur then individual ROM signatures, using CE as start/stop, previously recorded from a good system will pinpoint which device is defective.

IX. — RAM TESTS

These are the most difficult devices to test in any system. Software must be developed which will write into the RAM a pattern of "0" & "1" 's and read out this pattern as a signature. A ROM with a RAM exercise pattern on the address and instruction bus must be developed.

There are several RAM cell test patterns, the best being an individual preference. Kurz-Kasch will assist you in developing a RAM test if you should desire.

The start/stop and clock inputs to the Signature II are generally the same ones used for the ROM test. KK PROM part #0500021 is used with Z80 & 8080 Systems.

X. — INOP OPERATION

1. Remove Processor from target board.
2. Plug in INOP ribbon cable into socket, making sure pin 1 goes to pin 1 of the socket.
3. Disable automatic reset circuitry ("watchdog").
4. Procedures for taking signatures:
   a. To take address signatures, connect S/S, CLK, and GND to the pins on the board labeled S/S, CLK, and GND.The resulting signatures are the same as those listed on page 18 (CLK,START, STOP all rising edge _|‾).
   b. To take RAM Signatures, leave CLK and GND connected to the INOP and connect S/S to the chip select or the chip enable of the RAM being tested. The following table shows the resulting signatures that should be obtained. CLK _|‾, START ‾|_ , STOP _|‾).

	PROGRAM 1*	PROGRAM 2*
D_0	0001	0000
D_1	0000	0001
D_2	0001	0000
D_3	0000	0001
D_4	0001	0000
D_5	0000	0001
D_6	0001	0000
D_7	0000	0001

Program 1 is selected when all of the program switches are closed. Program 2 is selected when switches 1, 3, & 4 are closed. There are six other switch combinations which enable the User to program in any dedicated software for a particular hardware configuration. The following table shows the memory locations for the other possible switch combinations.

*Valid only for program EPROM furnished by Kurz-Kasch and not modified.

SWITCH SETTINGS		PROGRAM SELECTED	MEMORY LOCATION
Open	Closed		
3	2,4	3	0200H to 02FFH
2, 3	4	4	0300H to 03FFH
4	2, 3	5	0400H to 04FFH
2, 4	3	6	0500H to 05FFH
3, 4	2	7	0600H to 06FFH
2, 3, 4	-	8	0700H to 07FFH

5. After the program has been selected, power up target board.
6. Depress reset button on the INOP.
7. When the gate light on the Signature II begins flashing, the signatures on page 1 are ready to be taken.

*NOTE: The 0001 Signature is a VCC signature.
The signature may vary depending upon the target board being tested.
VCC signature for the board under test can be determined by touching the data probe tip to the 5V supply after the gate light begins to flash.
The following signatures show possible, but not all, VCC signatures

VCC Signatures

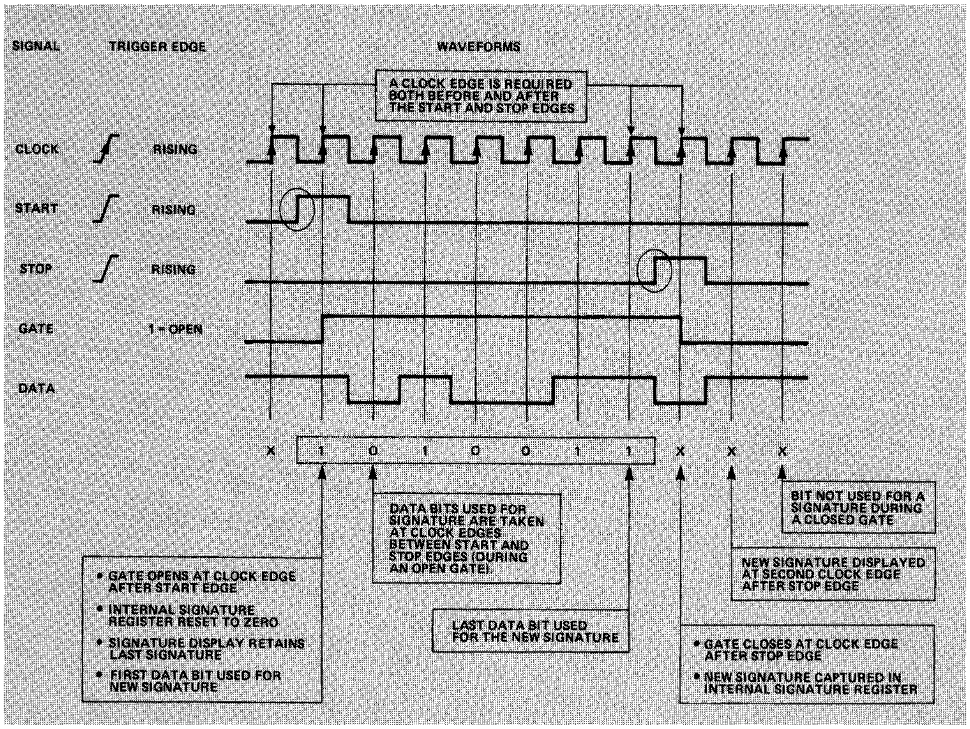
7P25
2A1F
A206
C133
8P3U
3319
7C47
C25F
5H21
19H6
HP66
U81P
003U
000U

00UP
UP73
3951
A70F
6PCP
CC34
4596
8P54
7A70
826P
P254
1180
755U
0001

Application Note 222-4

GUIDELINES FOR SIGNATURE ANALYSIS

Understanding the Signature Measurement



SIGNAL
TRIGGER EDGE
WAVEFORMS
A CLOCK EDGE IS REQUIRED BOTH BEFORE AND AFTER THE START AND STOP EDGES
CLOCK
RISING
START
RISING
STOP
RISING
GATE
1 = OPEN
DATA
X
1
0
1
0
0
1
1
X
X
X
DATA BITS USED FOR SIGNATURE ARE TAKEN AT CLOCK EDGES BETWEEN START AND STOP EDGES (DURING AN OPEN GATE).
BIT NOT USED FOR A SIGNATURE DURING A CLOSED GATE
NEW SIGNATURE DISPLAYED AT SECOND CLOCK EDGE AFTER STOP EDGE
• GATE OPENS AT CLOCK EDGE AFTER START EDGE
• INTERNAL SIGNATURE REGISTER RESET TO ZERO
• SIGNATURE DISPLAY RETAINS LAST SIGNATURE
• FIRST DATA BIT USED FOR NEW SIGNATURE
LAST DATA BIT USED FOR THE NEW SIGNATURE
• GATE CLOSES AT CLOCK EDGE AFTER STOP EDGE
• NEW SIGNATURE CAPTURED IN INTERNAL SIGNATURE REGISTER

HEWLETT PACKARD

APPLICATION NOTE 222-4

GUIDELINES FOR SIGNATURE ANALYSIS
Understanding the Signature Measurement

This Application Note shows how Hewlett-Packard Signature Analyzers take signature measurements. It contains guidelines for controlling the gate through the Start, Stop, and Clock inputs. It shows how measurements of three-state nodes are treated, and more. This information applies to both design and retrofit situations.

FORWARD

ABOUT DIGITAL TROUBLESHOOTING

Microprocessors have revolutionized your product line. Your products are smarter, faster, friendlier and more competitive because they take advantage of μP-based control and computation. They are also harder to build, harder to test and harder to fix when they fail. Complex bus structures and timing relationships have practically obsoleted the scope/voltmeter signal tracing techniques so effective on analog products. The need to enhance the testability and serviceability of your digital products is acute. So is the need for specialized digital troubleshooting equipment.

ABOUT SIGNATURE ANALYSIS

To address these needs, Hewlett-Packard has developed the Signature Analysis technique, as well as a Signature Analyzer product line, for component-level troubleshooting of microprocessor-based products. A Signature Analyzer detects and displays the unique digital signatures associated with the data nodes in a circuit under test. By comparing these actual signatures to the correct ones, a troubleshooter can back-trace to a faulty node. By designing or retrofitting S.A. into digital products, a manufacturer can provide manufacturing test and field service procedures for component-level repair, without dependence on expensive board-exchange programs.

ABOUT THIS PUBLICATION

This application note is aimed at assisting designers, test engineers and others in designing or retrofitting their digital products for Signature Analysis testability and serviceability. It shows how Hewlett-Packard Signature Analyzers take signature measurements. While there are many different ways to control the signature measurement, there are a few common ways that apply to most microprocessor-based products. We've compiled a list of these common measurement control methods in the form of guidelines. These guidelines resulted from suggestions of some of the hundreds who have used signature analysis for the past few years. They are designed to show how to create measurements that result in stable and repeatable signatures, so that correct signatures can be documented in a troubleshooting procedure. When that goal is met, then incorrect, unrepeatable, or unstable signatures measured during troubleshooting, will accurately indicate incorrect or intermittant circuit behavior, allowing fast fault analysis to the component level.

ABOUT OTHER PUBLICATIONS

Application Note 222-0, HP Publication 02-5952-7593, is a complete index to the latest Signature Analysis publications. It lists all other application notes currently available in the AN 222 series about Signature Analysis. They cover a wide range of interests, from how to design or retrofit Signature Analysis into digital systems, to the cost reductions that can be expected in production test and field service by doing so. It also lists all data sheets for the complete line of Hewlett-Packard Signature Analysis products, plus other related publications about digital troubleshooting.

CONTENTS

SECTION 1

Introduction

Contents

This application note explains how Hewlett-Packard Signature Analyzers take signature measurements. Here is what is covered in this note:

1. How the GATE, or measurement cycle, is controlled, including simplified examples that demonstrate basic gating concepts. Some application examples are also given to demonstrate the concepts in a real situation.
2. How data on three-state nodes is interpreted by the signature analyzer during a measurement cycle.
3. How the signature analyzer responds to a reset, and how the HOLD feature works in conjunction with reset.
4. How the signature analyzer is affected by noise on any of its inputs.
5. The signature analyzer specifications.
6. Ways the measurement cycle can be controlled by the product under test, through a combination of hardware and software, during the execution of a circuit stimulus program.

Examples of typical circuit stimulus programs are the subject of additional Case Study notes in this Application Note series. They show typical examples of stimulus program software and the circuits that get stimulated. The Case Studies also contain additional examples of how the GATE is controlled. See Application Note 222-O, "An Index to Signature Analysis Publications," HP Publication 02-5952-7593, for a complete list of current application notes in the AN 222 series.

Organization

Each section of this application note covers a measurement concept by showing simple examples of START, STOP, CLOCK, and DATA waveforms. In some cases, actual application examples will also be shown. The title for each section reflects the way in which most people phrase questions about how the signature analyzer takes a measurement. This organization allows the note to be used as a quick reference guide when a specific question comes up about a measurement concept. But the organization also allows someone unfamiliar with the signature measurement process, to read through it, without having to refer back and forth to different sections.

The simplified examples are intended only to demonstrate and clarify a measurement concept. They do not necessarily reflect an actual measurement. For example, the START, STOP, CLOCK and DATA waveforms are simplified so that they could easily be drawn in diagrams. GATE times were shortened, and the CLOCK is shown with a fixed frequency. However, GATES can be much longer than shown (GATE length is a subject covered in a section of the same name), and the CLOCK need not be constant. The CLOCK can be symmetrical as shown in most of the examples, but it need not be.

SECTION 2

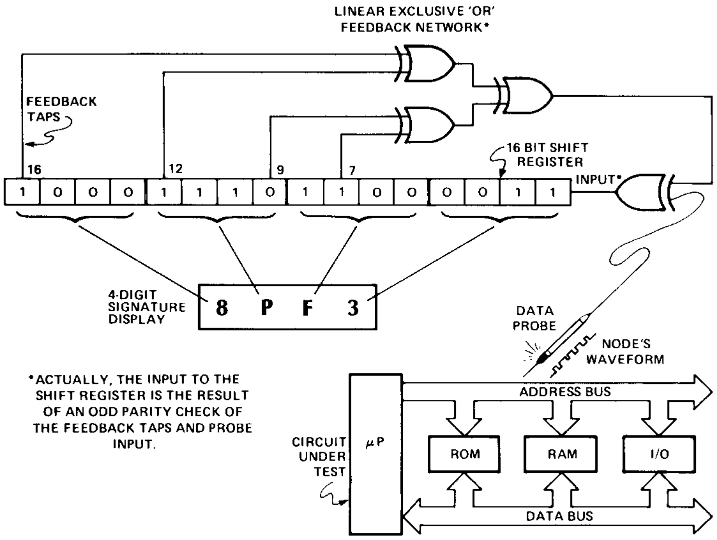
Signature Analyzer Characteristics

Here is a list of the four major characteristics of Hewlett-Packard Signature Analyzers that determine the signature value resulting from a signature measurement cycle. Other signature analyzers may or may not get the same signatures, depending on how they treat these and other characteristics. The first two characteristics are shown in the Figures 2.1 and 2.2. The third and fourth characteristics are covered in the remaining sections of this application note.

1. The internal 16-bit signature register feedback taps.
2. The signature display characters.
3. The gating characteristics.
4. The input logic thresholds.

The Signature Register

Figure 2.1 shows the model for the 16-bit internal signature register including the four feedback tap positions. It also shows how the signature analyzer compresses a node's waveform into a four digit signature display. More on the operation of the shift register and its accuracy of error detection can be found in Application Note 222-2.

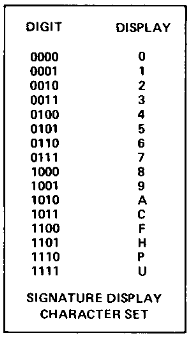

LINEAR EXCLUSIVE 'OR' FEEDBACK NETWORK*
FEEDBACK TAPS
16
12
9
7
16 BIT SHIFT REGISTER
INPUT*
1 0 0 0 1 1 1 0 1 1 0 0 0 0 1 1
4-DIGIT SIGNATURE DISPLAY
8 P F 3
DATA PROBE
NODE'S WAVEFORM
*ACTUALLY, THE INPUT TO THE SHIFT REGISTER IS THE RESULT OF AN ODD PARITY CHECK OF THE FEEDBACK TAPS AND PROBE INPUT.
CIRCUIT UNDER TEST
µP
ADDRESS BUS
ROM
RAM
I/O
DATA BUS

Model Signature Analyzer

Figure 2.1

The Signature Display Character Set

Figure 2.2 shows the sixteen characters that form each of the four digits of the signature display. The character set has been changed from standard hexidecimal, becasue the signature analyzer uses a seven-segment display. The seven-segment display was chosen to make the characters as large as economically possible, for easy readability during troubleshooting. The new character set eliminates the confusion between the small letter b and the number 6, or the capital letter B and the number 8. During troubleshooting, measured signatures are compared against documented ones. There is no diagnostic information in the signature itself. So it doesn't matter WHAT the characters are, as long as they are EASY to compare. The new character set makes the signature instantly recognizable for fast troubleshooting.



DIGIT	DISPLAY
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	C
1100	F
1101	H
1110	P
1111	U

SIGNATURE DISPLAY CHARACTER SET

Figure 2.2

Gating Characteristics and Input Thresholds

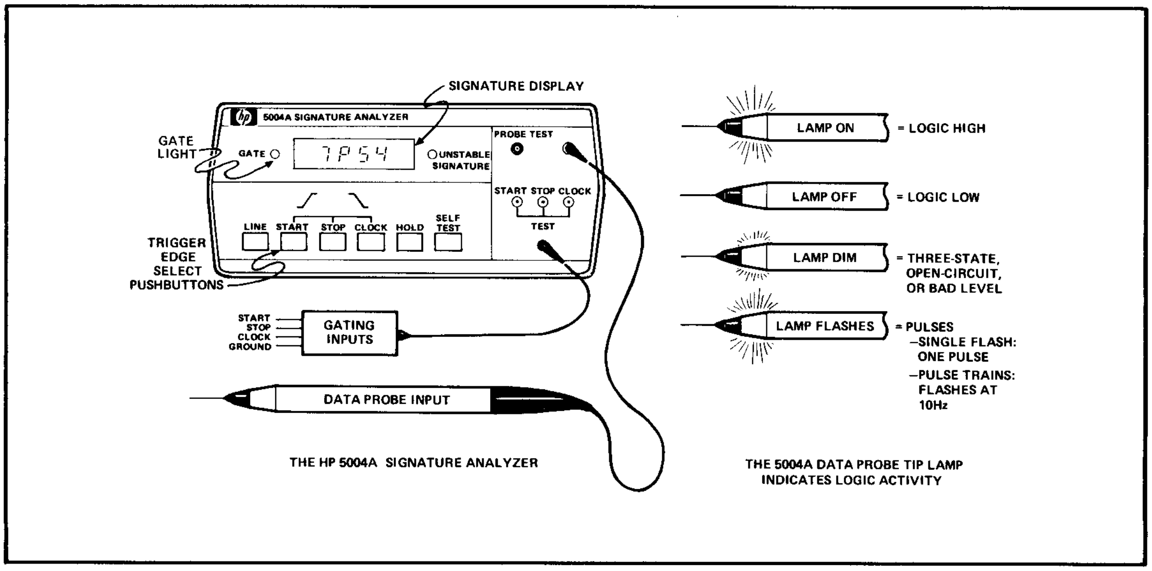
The gating characteristics are covered in sections three through ten of this application note. Input threshold is the subject of section 11 and Appendix A.

Basic Gate Operation

Definition of the Gate

The measurement cycle of the signature analyzer is controlled through the GATE. When the GATE is open, the analyzer is taking a new signature measurement through the DATA probe input. The signature from the previous measurement cycle remains displayed. When the GATE closes, the analyzer stops taking the new signature and then displays it.

The GATE is not an input to the signature analyzer. It is an internal state whose operation is shown by the GATE LIGHT. The GATE LIGHT is shown in Figure 3.1. It turns on when the GATE opens, and turns off when it closes. The GATE LIGHT will blink at about 10 Hz if the GATE cycles faster than that.



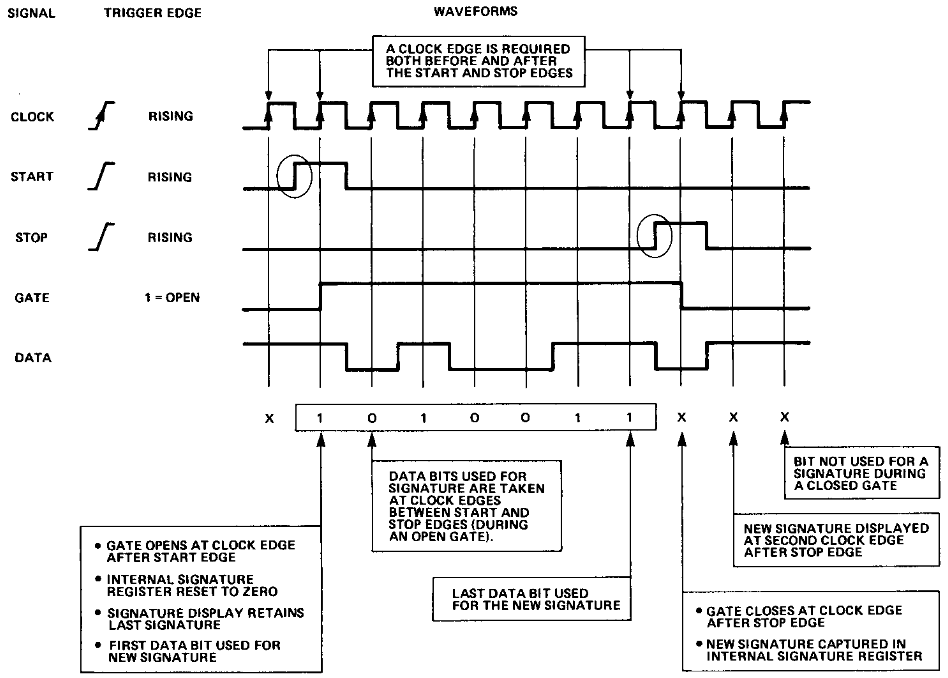
THE HP 5004A SIGNATURE ANALYZER

THE 5004A DATA PROBE TIP LAMP INDICATES LOGIC ACTIVITY

Figure 3.1

The START, STOP, CLOCK and DATA Inputs

The GATE is controlled by the three gating inputs: START, STOP and CLOCK. Basic GATE control is shown in Figure 3.2. START and STOP are used to open and close the GATE at selected trigger edges. The trigger edges for START, STOP and CLOCK are selected by pushbuttons on the instrument, and can be either the rising or falling edge of the input signal, in any combination. From now on, any reference to a START, STOP or CLOCK edge will mean the edge as selected by these pushbuttons.


SIGNAL
TRIGGER EDGE
WAVEFORMS
A CLOCK EDGE IS REQUIRED BOTH BEFORE AND AFTER THE START AND STOP EDGES
CLOCK
RISING
START
RISING
STOP
RISING
GATE
1 = OPEN
DATA
X 1 0 1 0 0 1 1 X X X
BIT NOT USED FOR A SIGNATURE DURING A CLOSED GATE
DATA BITS USED FOR SIGNATURE ARE TAKEN AT CLOCK EDGES BETWEEN START AND STOP EDGES (DURING AN OPEN GATE).
NEW SIGNATURE DISPLAYED AT SECOND CLOCK EDGE AFTER STOP EDGE
• GATE OPENS AT CLOCK EDGE AFTER START EDGE
• INTERNAL SIGNATURE REGISTER RESET TO ZERO
• SIGNATURE DISPLAY RETAINS LAST SIGNATURE
• FIRST DATA BIT USED FOR NEW SIGNATURE
LAST DATA BIT USED FOR THE NEW SIGNATURE
• GATE CLOSES AT CLOCK EDGE AFTER STOP EDGE
• NEW SIGNATURE CAPTURED IN INTERNAL SIGNATURE REGISTER

Basic Gate Operation

Figure 3.2

The CLOCK synchronizes the analyzer to the device being tested. The logic level of the waveform measured by the DATA probe input is sampled at the CLOCK edge and is ignored at all other times. These logic levels are used as data bits that are compressed into the signature. The START and STOP inputs are also synchronously detected by the CLOCK. The GATE opens at the CLOCK edge following the START edge, and not at the START edge itself. Similarly, the GATE closes at the CLOCK edge following the STOP edge, instead of directly at the STOP edge.

First and Last Data Bits Sampled For a Signature

The first DATA bit sampled for use in a signature occurs coincident with the GATE opening at the CLOCK edge following the START edge. The last DATA bit used occurs at the CLOCK edge PRIOR to the STOP edge. In other words, DATA is sampled at every CLOCK edge BETWEEN the START and STOP edges. DATA is not used for a signature when the GATE closes at the CLOCK edge following the STOP edge. The signature is displayed at the SECOND CLOCK edge after the STOP edge.

There are many different ways to control the GATE of the signature analyzer. For instance, both START and STOP can be connected to the same signal or different ones. And they can be triggered on the same or different edges. START usually opens the GATE and STOP closes it, but the GATE can toggle open and closed as well. The GATE can be as long or as short as required. START and STOP can even have multiple edges. The remaining guidelines explain each one of these characteristics and more.

Getting the Gate To Open and Close

The Interaction of START, STOP and CLOCK

There must be a CLOCK edge both before and after the START and STOP edges, as shown in Figure 3.2. The GATE opens at the CLOCK edge following the START edge, and not at the START edge itself. Similarly, the GATE closes at the CLOCK edge following the STOP edge, instead of directly at the STOP edge. A CLOCK edge is also required before both the START and STOP edges, because the signature analyzer compares the logic states of START and STOP from one CLOCK edge to the next, and detects a change in state as an edge. The result of that comparison determines whether the GATE will open or close. For example, consider START selected for a rising edge trigger, the GATE initially closed, and START initially at logic low. At each CLOCK edge, START remains low, so a trigger edge is not detected, and the GATE remains closed. Now, before the next CLOCK edge, suppose START goes high. At the next CLOCK edge, START will be detected as high. The signature analyzer detects the change in state from low (at one CLOCK edge) to high (at the next CLOCK edge) as a rising edge, and opens the GATE at the CLOCK edge. See Figure 3.2.

Situations That Cause the Gate Not To Open or Close

If the GATE does not open or close as expected, it's probably because there is not a CLOCK edge both before and after the START and STOP edges. Here are four situations that demonstrate this, along with ways to get the GATE to open and close as required.

1: No CLOCK before or after the START or STOP edge.

In Figure 4.1, START and STOP are connected to the same signal and trigger on opposite edges. There is no CLOCK edge to detect the logic LOW level of START/STOP, so the rising edge for START does not open the GATE. Similarly the falling edge for STOP won't close the GATE if it starts open.

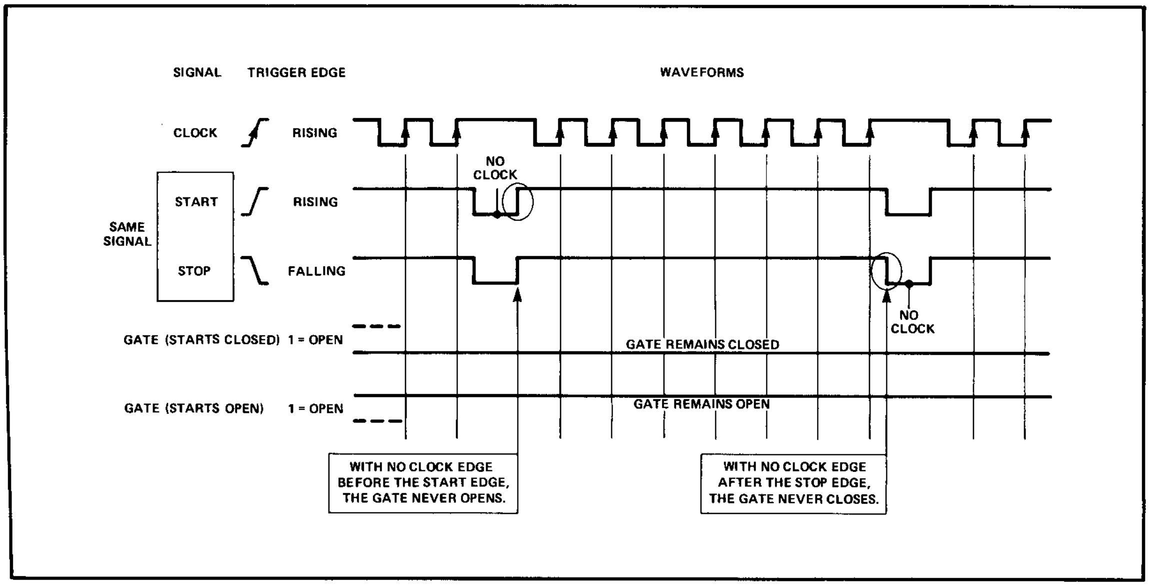
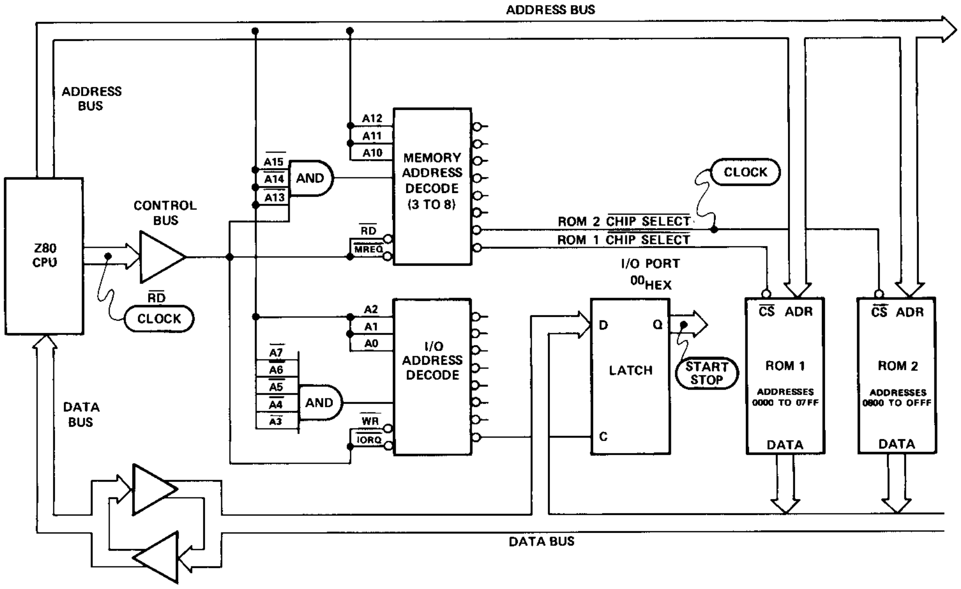

SIGNAL
TRIGGER EDGE
WAVEFORMS
CLOCK
RISING
NO CLOCK
START
RISING
SAME SIGNAL
STOP
FALLING
NO CLOCK
GATE (STARTS CLOSED) 1 = OPEN
GATE REMAINS CLOSED
GATE (STARTS OPEN) 1 = OPEN
GATE REMAINS OPEN
WITH NO CLOCK EDGE BEFORE THE START EDGE, THE GATE NEVER OPENS.
WITH NO CLOCK EDGE AFTER THE STOP EDGE, THE GATE NEVER CLOSES.

Figure 4.1

Here's an example of this situation from an actual application. In the Zilog Z80 microprocessor-based system of Figure 4.2, START and STOP are connected to a bit in a latch addressed as an I/O port. START is selected to trigger on a rising edge, and STOP triggers on a falling edge. The START and STOP edges are generated by a combination of this hardware, and the execution of the ROM stimulus program of Figure 4.3. The bit in the latch is set to logic one at the beginning of the program to open the GATE, and reset to logic zero at the end to close the GATE.



Simplified Z80 Microprocessor-Based System

Figure 4.2

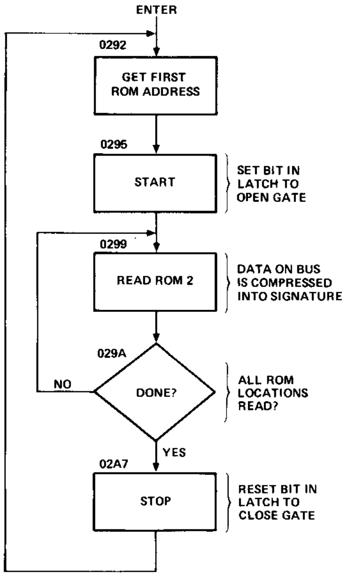
The ROM stimulus program of Figure 4.3 is stored in ROM #1, and is used to exercise ROM #2. This allows signature analysis to be used to troubleshoot ROM #2 and associated circuits such as address decoders. The program simply reads all locations of ROM #2 onto the data bus. Before the first ROM #2 location is read, the program sets the bit in the latch to open the GATE. Now all ROM #2 data that is read onto the bus will be used for a signature. After reading all ROM #2 locations, the program resets the bit to close the GATE, then returns to the beginning to open the GATE and start over.

Z80 ASSEMBLY CODE

```
HEX
ADRS   CONTENTS   LABEL    INSTRUCTION          COMMENTS

0292   210008     START:   LD    HL, 0800   ; H-L POINTS TO START OF ROM.
0295   AF                  XOR   A          ; A=0
0296   3C                  INC   A          ; A=1
0297   D300                OUT   00H, A     ; OPEN GATE BY SETTING LATCH.
0299   7E         ROM:     LD    A, (HL)    ; READ ROM.
029A   23                  INC   HL         ; INCREMENT ROM POINTER.
029B   7D                  LD    A, L       ; HAVE 256 BYTES OF ROM.
029C   FE00                CPI   00H        ; BEEN READ?
029E   C29902              JP    NZ, ROM    ; NO. READ MORE ROM.
02A1   7C                  LD    A, H       ; YES, BUT HAVE 2K BYTES.
02A2   FE10                CPI   10H        ; OF ROM BEEN READ?
02A4   C29902              JP    NZ, ROM    ; NO. READ MORE ROM.
02A7   AF         STOP:    XOR   A          ; YES. A=0
02A8   D300                OUT   00H, A     ; CLOSE GATE BY RESETTING LATCH.
02AA   C39202              JP    START      ; JUMP TO BEGINNING OF LOOP.
```

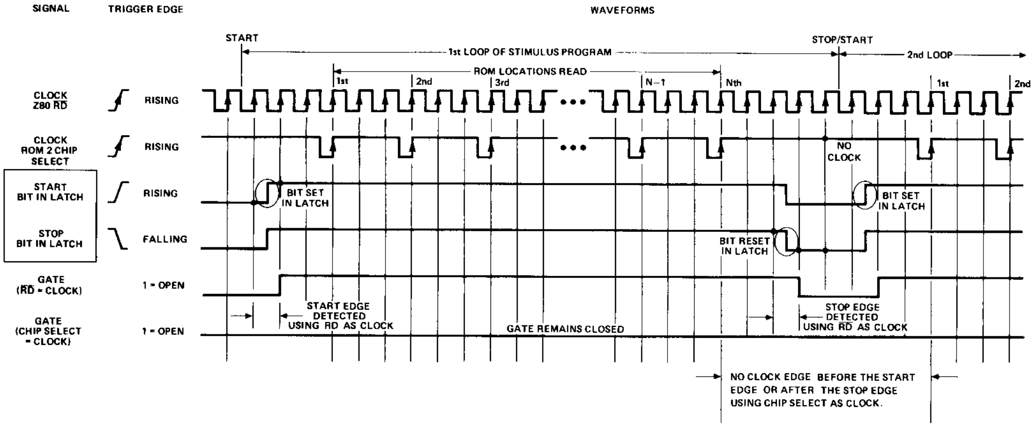
NOTE: THE LATCH THAT IS SET AND RESET TO OPEN AND CLOSE THE GATE IS LOCATED AT I/O PORT ADDRESS 00_H.


ENTER
0292
GET FIRST ROM ADDRESS
0295
START
SET BIT IN LATCH TO OPEN GATE
0299
READ ROM 2
DATA ON BUS IS COMPRESSED INTO SIGNATURE
029A
NO
DONE?
ALL ROM LOCATIONS READ?
YES
02A7
STOP
RESET BIT IN LATCH TO CLOSE GATE

ROM Stimulus Program

Figure 4.3

If $\overline{\text{RD}}$ from the Z80 is used as a CLOCK for the signature analyzer, then the START and STOP edges will be detected, and the GATE will open and close as shown in Figure 4.4. But if the chip select for ROM #2 is used as the CLOCK (which combines the address decode for ROM #2 and the $\overline{\text{RD}}$ line from the Z80), the GATE will not open. This is because a CLOCK edge does not occur between the program steps where the bit is reset and set again. ROM #2 is not being read during that time, so the chip select line, and therefore the CLOCK, remains inactive.



ROM Stimulus Program Timing Diagram Showing Start and Stop Edge Detection

Figure 4.4

Why not connect CLOCK to RD instead of the chip select for ROM #2 and be satisfied? The answer depends on the data that should, and should not be, clocked into the signature analyzer. In this example, a RD CLOCK occurs every time the processor reads data from ROM #2. Therefore, if a signature is taken on the data bus, then it will be composed of data from ROM #2 as intended. But a RD CLOCK also occurs every time the processor fetches instructions from ROM #1 during program execution. Therefore, the signature is also composed of data from ROM #1 as well as ROM #2. This is satisfactory as long as the contents of ROM #1 is known to be good, or can be verified by some other means such as FREERUN. (The contents of ROM #1 had better be good or else the program will not execute properly, and will not exercise ROM #2.)

What if it's necessary to clock data into the signature analyzer **only** when a device is exercised by the program (in order to keep data from being entered into the signature analyzer during any other time)? For instance, imagine that this example program was used to stimulate RAM, instead of ROM, and that the RAM was accessed by both the processor and another device. This occurs in a CRT terminal where the processor stores characters in RAM, and the CRT regularly accesses the RAM to put the characters on the screen. This interaction of the processor and the CRT with RAM, is random with respect to each other. If the CLOCK is connected to RD, then the data sampled by the signature analyzer will be a combination of data as accessed by the processor (during the stimulus program), and data as accessed by the CRT (during refresh of the screen). Since this combination of data is random, the signature will be unstable. To get a stable signature, the CLOCK is connected to the chip select of the RAM. The chip select combines the RD line with the processor's address decode for the RAM. The chip select line is active only when the RAM is accessed by the processor. Now the signature analyzer will sample data from the RAM only while the RAM is read during the stimulus program.

In the original ROM example, if the chip select for ROM #2 is used as a CLOCK, then only data from ROM #2 will be used for a signature. Data from ROM #1 will be ignored. However, when the chip select for ROM #2 is used as a CLOCK, then the START and STOP edges are not detected. This is because the chip select is not active, and therefore there's no CLOCK edge, between the START and STOP edge generation. To create the required CLOCK edge, a minor step could be added to the program that reads the first ROM location as shown in Figure 4.5.

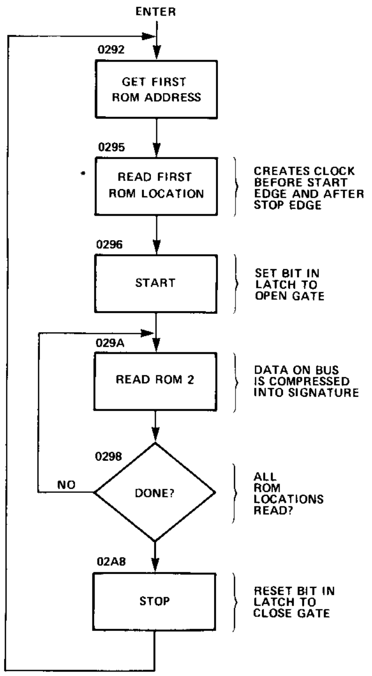
Z80 ASSEMBLY CODE

```
   HEX
   ADRS   CONTENTS   LABEL       INSTRUCTION        COMMENTS

   0292   210008     START:      LD    HL, 0800   ; H-L POINTS TO START OF ROM.
*  0295   7E                     LD    A, (HL)    ; READ FIRST ROM LOCATION.
   0296   AF                     XOR   A          ; A=0
   0297   3C                     INC   A          ; A=1
   0298   D300                   OUT   00H, A     ; OPEN GATE BY SETTING LATCH.
   029A   7E         ROM:        LD    A, (HL)    ; READ ROM.
   029B   23                     INC   HL         ; INCREMENT ROM POINTER.
   029C   7D                     LD    A, L       ; HAVE 256 BYTES OF ROM.
   029D   FE00                   CPI   00H        ; BEEN READ?
   029F   C29A02                 JP    NZ, ROM    ; NO. READ MORE ROM.
   02A2   7C                     LD    A, H       ; YES, BUT HAVE 2K BYTES.
   02A3   FE10                   CPI   10H        ; OF ROM BEEN READ?
   02A5   C29A02                 JP    NZ, ROM    ; NO. READ MORE ROM.
   02A8   AF         STOP:       XOR   A          ; YES. A=0
   02A9   D300                   OUT   00H, A     ; CLOSE GATE BY RESETTING LATCH.
   02AB   C39202                 JP    START      ; JUMP TO BEGINNING OF LOOP.
```

NOTE: THE LATCH THAT IS SET AND RESET TO OPEN AND CLOSE THE GATE IS LOCATED AT I/O PORT ADDRESS 00_H.

• EXTRA PROGRAM STEP

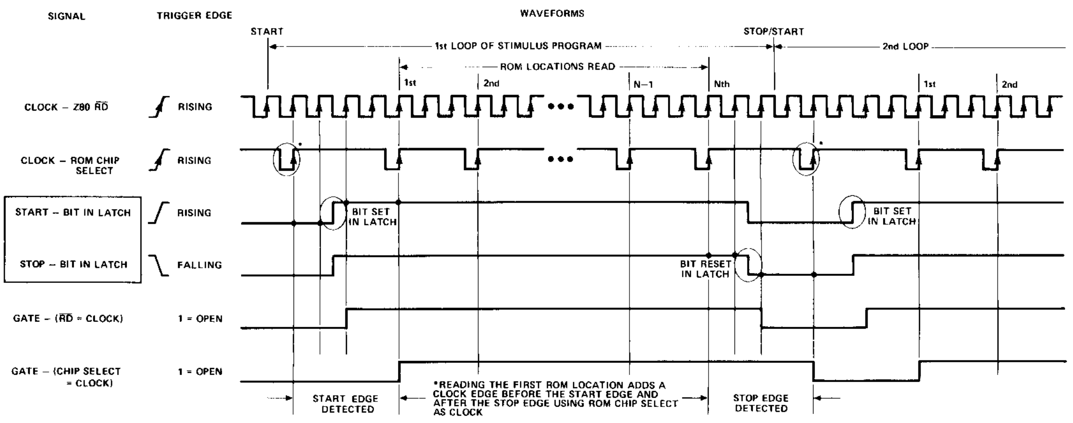


ENTER
0292
GET FIRST ROM ADDRESS
0295
•
READ FIRST ROM LOCATION
CREATES CLOCK BEFORE START EDGE AND AFTER STOP EDGE
0296
START
SET BIT IN LATCH TO OPEN GATE
029A
READ ROM 2
DATA ON BUS IS COMPRESSED INTO SIGNATURE
0298
NO
DONE?
ALL ROM LOCATIONS READ?
02A8
STOP
RESET BIT IN LATCH TO CLOSE GATE

ROM Stimulus Program With Step Added
To Detect the Start and Stop Edges

Figure 4.5

This inserts a clock pulse into the CLOCK waveform as shown in Figure 4.6. Now a CLOCK edge occurs both before and after the START and STOP edges, and the GATE opens and closes as required.

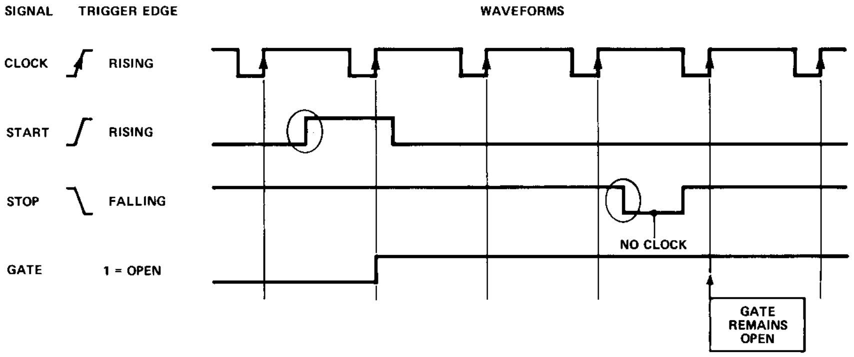


ROM Stimulus Program Timing Diagram Showing Start and Stop Edge Detection

Figure 4.6

2: The START or STOP pulse is too short for the CLOCK frequency.

In the example of Figure 4.7, the STOP pulse is too short for the slower CLOCK frequency. The GATE remains open because the STOP edge is not detected. This can occur when START and STOP are connected to address decodes, and pulsed at the beginning and end of the stimulus program. If a CLOCK is chosen that is slower than the pulses, such as a UART's transmitter clock input, then the pulses will remain undetected. To get the GATE to open and close, choose a higher frequency CLOCK, or create new START and STOP edges by controlling a bit in a latch as shown in the previous example, or using the address decode pulses to set and reset a flip-flop or latch directly.

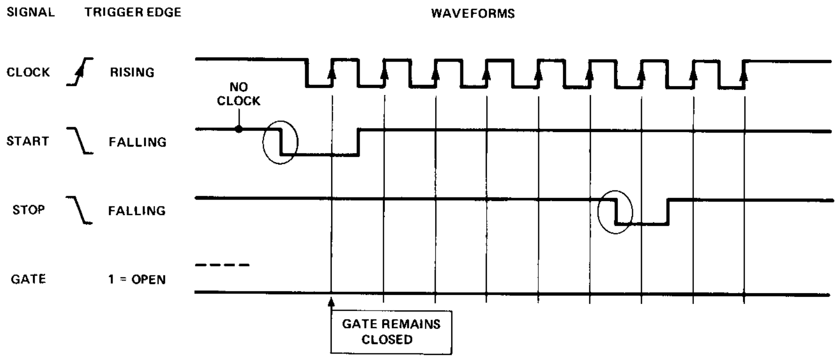


The STOP Edge Is Not Detected, Leaving the GATE Open,
Because the STOP Pulse Is Too Short For the CLOCK Frequency

Figure 4.7

3: The CLOCK turns on after the START edge.

In figure 4.8, the START edge is not detected because it occurs before the first CLOCK edge. This can happen in board test systems, where START is the signal from the board tester that turns on the CLOCK to the board under test. Here, selecting the other edge for START will allow the GATE to open. Choosing a CLOCK that is running before the START edge will also ensure its detection.

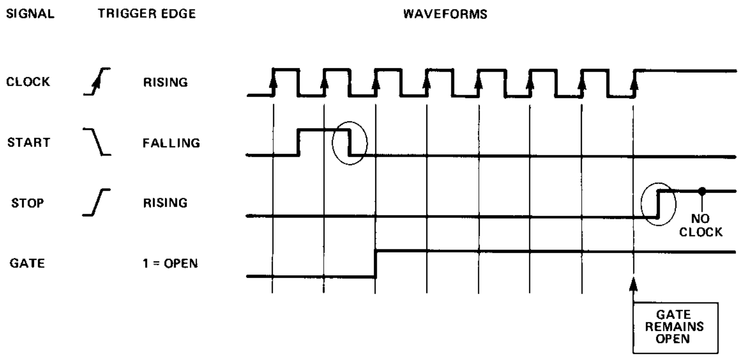


The START Edge Is Not Detected, and the GATE Never Opens, Because the CLOCK Turns On After the START Edge

Figure 4.8

4: The CLOCK turns off before the STOP edge.

In Figure 4.9, the STOP edge is not detected because it occurs after the last CLOCK edge. In this example, STOP is the terminal count output of a counter. When the terminal count occurs, the counter's clock is turned off. Connecting CLOCK to the counter's clock will cause the STOP edge to remain undetected. To ensure detection, the CLOCK should be moved to a clock that runs after the STOP edge.



The STOP Edge Is Not Detected, Leaving the GATE Open, Because the CLOCK Turns Off Before the STOP Edge

Figure 4.9

Two Ways To Control The Gate

There are two basic ways to control the GATE. One way is to have the START edge open the GATE and a separate STOP edge close it. Another way is to toggle the GATE open and closed.

START Opens the GATE, STOP Closes It

Here are two examples where the START edge always opens the GATE, and the STOP edge always closes it. In the first example of Figure 5.1, this happens when:

1. START and STOP are connected to DIFFERENT signals.
2. START and STOP trigger on EITHER edge.
3. The START and STOP edges DO NOT occur between the same two CLOCK edges.

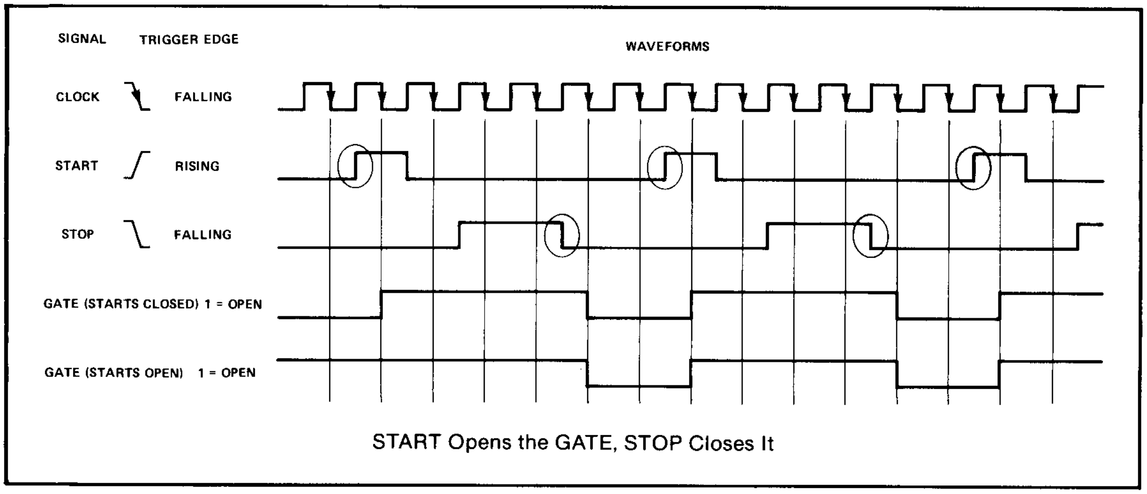

SIGNAL
TRIGGER EDGE
WAVEFORMS
CLOCK
FALLING
START
RISING
STOP
FALLING
GATE (STARTS CLOSED) 1 = OPEN
GATE (STARTS OPEN) 1 = OPEN
START Opens the GATE, STOP Closes It

Figure 5.1

In the second example of Figure 5.2:

1. START and STOP are connected to the SAME signal.
2. START and STOP trigger on OPPOSITE edges.

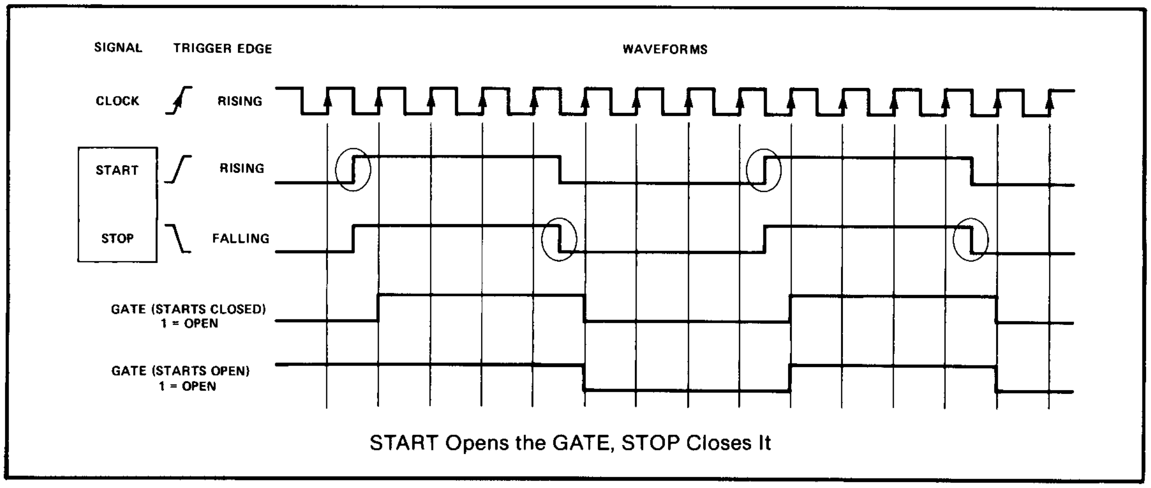
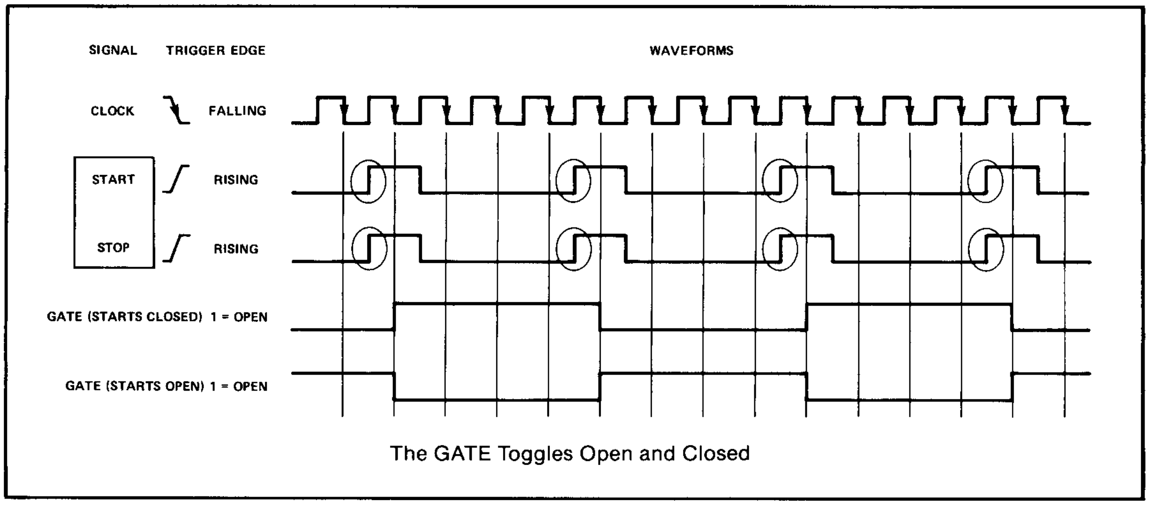

SIGNAL
TRIGGER EDGE
WAVEFORMS
CLOCK
RISING
START
RISING
STOP
FALLING
GATE (STARTS CLOSED) 1 = OPEN
GATE (STARTS OPEN) 1 = OPEN
START Opens the GATE, STOP Closes It

Figure 5.2

The GATE Toggles Open and Closed

In these examples, the GATE toggles because the START and STOP edges occur between the same two CLOCK edges. In the first example of Figure 5.3:

1. START and STOP are connected to the SAME signal.
2. START and STOP both trigger on the SAME edge.

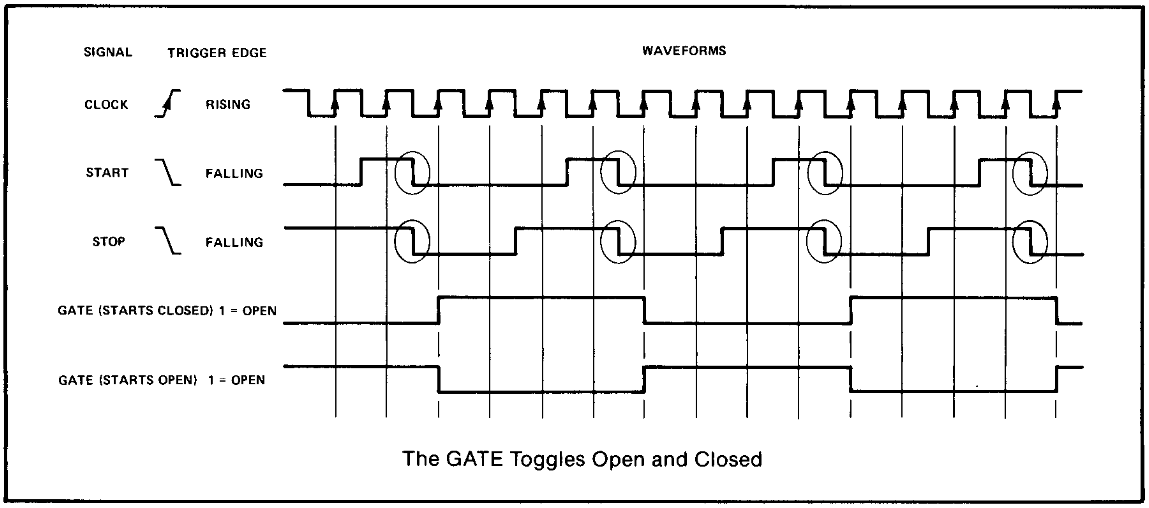


The GATE Toggles Open and Closed

Figure 5.3

In the second example of Figure 5.4:

1. START and STOP are connected to DIFFERENT signals.
2. START and STOP trigger on EITHER edge.
3. The START and STOP edges occur between the SAME two CLOCK edges.

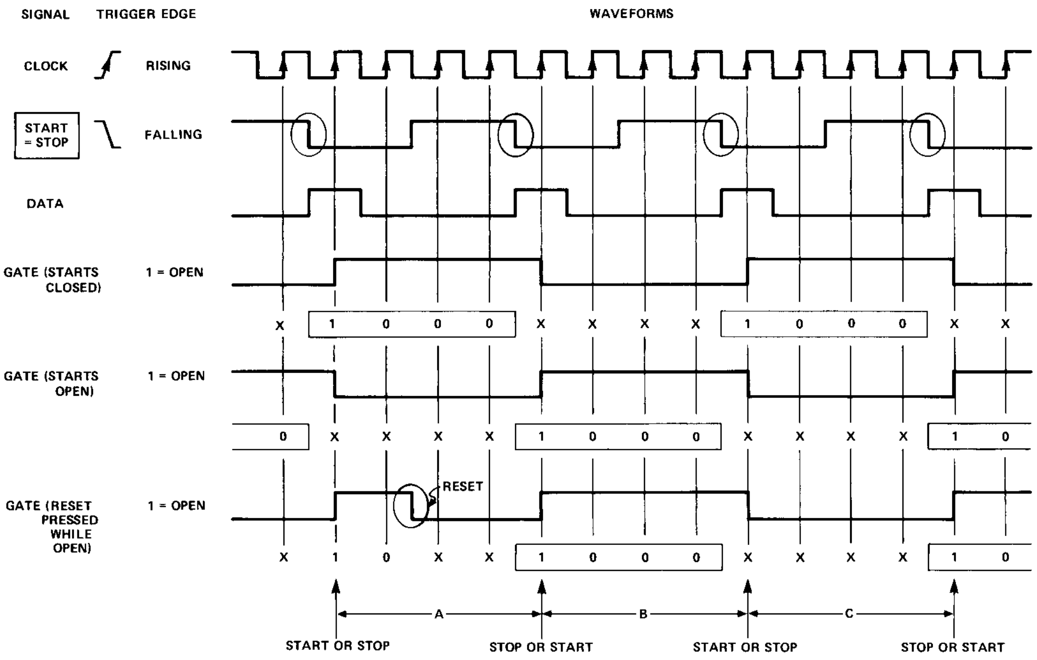


The GATE Toggles Open and Closed

Figure 5.4

When toggling the GATE, keep the same number of CLOCK edges, and the same pattern of DATA bits, between alternate trigger edges, so that the GATE can open arbitrarily at either edge. This insures repeatable signatures. Figures 5.5 through 5.7 show this.

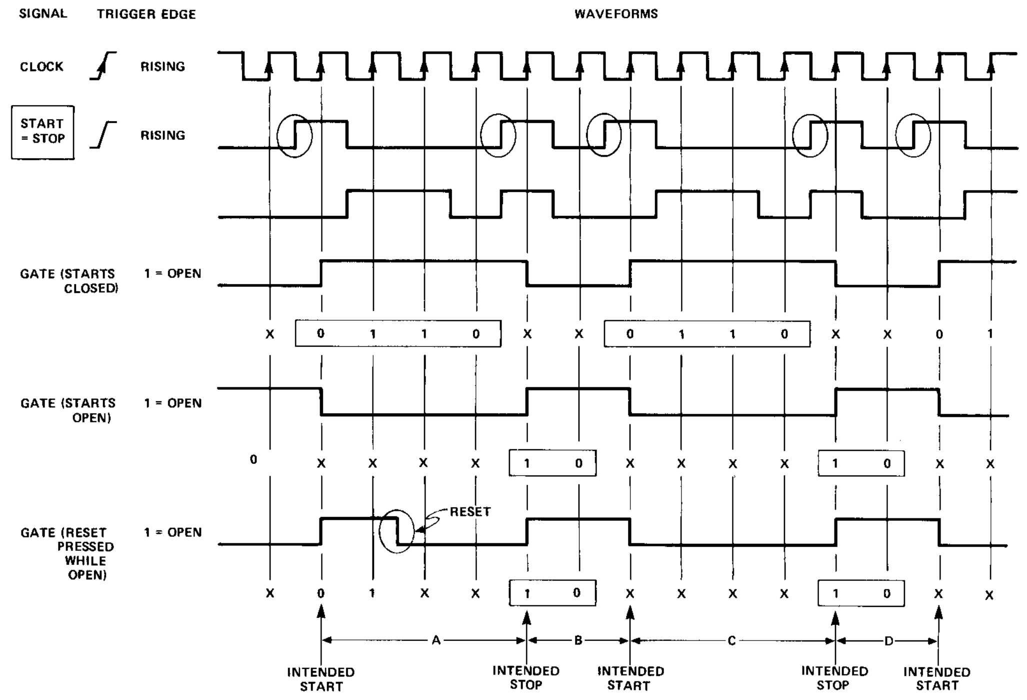
In Figure 5.5, the intended signature is always measured, regardless of the initial GATE state, even if the signature analyzer is RESET while the GATE is open. The reason? The number of CLOCK edges between any two trigger edges is identical, so that the GATE opens at any edge and still measures the same data (e.g., potential GATE times A, B, and C are identical).



The GATE Can Toggle Open Or Closed At Either Edge and Result In a Repeatable Signature, Because the Number Of CLOCK Edges and the Patterns Of Data Bits Are Always the Same Between START/STOP Edges

Figure 5.5

In Figure 5.6, the GATE is intended to be open during times A and C. However, if the GATE starts open (which can occur if RESET is pressed during A or C), then the GATE will continue to toggle at each new trigger edge, but will be open when it should be closed and vice versa. This will result in signatures taken during times B and D when the GATE is intended to be closed. The reason? The times between any two trigger edges are not the same. The GATE cannot open arbitrarily at any trigger edge, as it could in Figure 5.5, and still measure the same data.

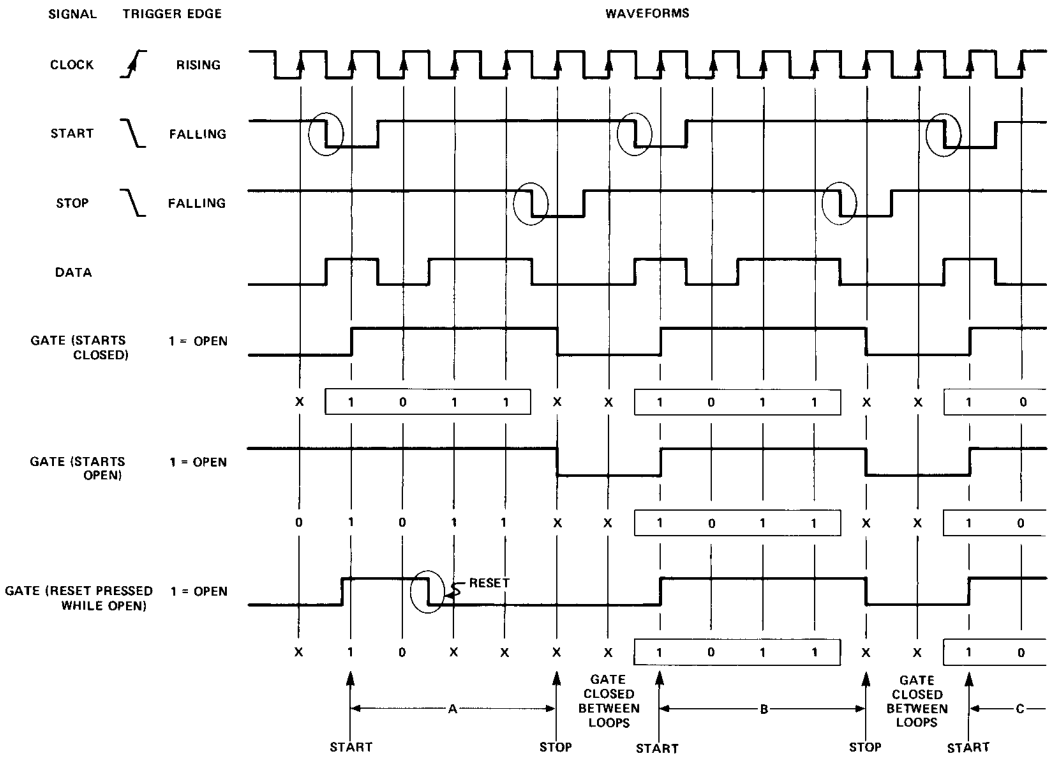


Signature Differs Depending On When GATE Opens

Figure 5.6

There are at least two ways to make the GATE open and close when intended. One way is to keep START and STOP on the same signal and continue to toggle the GATE. However, create an edge at the beginning of the stimulus program or time period of interest to open the GATE. Do not create another edge (for STOP) at the end of the program as in Figure 5.6. Instead, cause the program to return to the beginning where the same edge will then close the GATE for the next execution of the loop. The GATE will then continue to toggle as shown in Figure 5.5.

Another way is to have START open the GATE and STOP close it, as shown in Figure 5.7. Do this by connecting START and STOP to separate signals. Create one START edge at the beginning of the stimulus program or time period of interest to open the GATE. Create one STOP edge to close the GATE at the end of the program. Even pressing RESET will not change the times the GATE will open. This is especially useful if the GATE must be closed for an extended period between loops of the stimulus.


SIGNAL
TRIGGER EDGE
WAVEFORMS
CLOCK
RISING
START
FALLING
STOP
FALLING
DATA
GATE (STARTS CLOSED)
1 = OPEN
X 1 0 1 1 X X 1 0 1 1 X X 1 0
GATE (STARTS OPEN)
1 = OPEN
0 1 0 1 1 X X 1 0 1 1 X X 1 0
GATE (RESET PRESSED WHILE OPEN)
1 = OPEN
RESET
X 1 0 X X X X 1 0 1 1 X X 1 0
GATE CLOSED BETWEEN LOOPS
GATE CLOSED BETWEEN LOOPS
A
B
C
START
STOP
START
STOP
START

START Opens the GATE, STOP Closes It

Figure 5.7

Gate Lengths

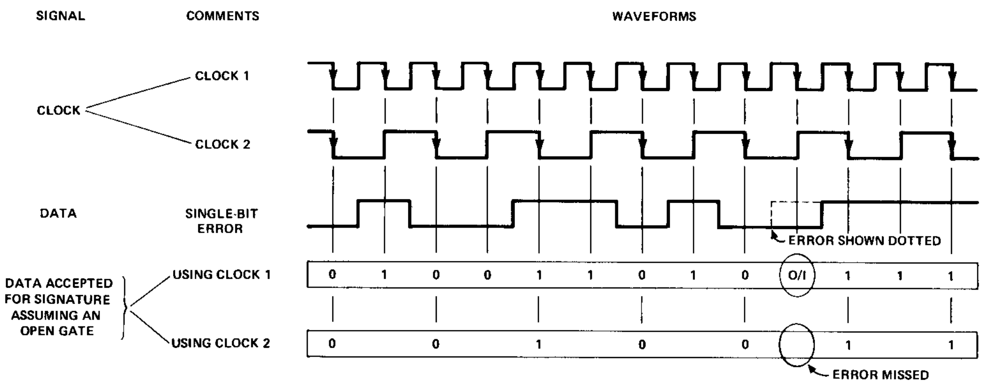
Short GATES Mean Fast Troubleshooting

Keeping GATE cycle times under about ½ second allows the troubleshooter to quickly jump from node to node, making the total troubleshooting time very fast. Even if the probe should accidently slip from the node, then when the probe is again solidly on the node, it will be a maximum wait of one second for the correct signature. When the GATE is two seconds or longer, it could seem like a long wait for the signature. If GATES are longer than two seconds, consider splitting the stimulus program into smaller sections, so that each one runs with shorter GATES.

The easiest way to determine the GATE length is to measure it empirically. That is, run the stimulus and observe the GATE light. For GATE cycle times longer than 100 msec, the light is a direct indication of GATE length. For cycle times less than 100 msec, the light will flash at about 10 Hz.

How Long Does The GATE Need To Be Open?

The GATE needs to be open only as long as it takes to stimulate the measured node through all of its functional states. This may require stimulating the node through more than both logic states. If this is done, then the signature analyzer will compress all the data on a node into a compact signature. The signature then contains all the information about the correct functioning of the device driving the node. Be sure that the CLOCK will sample all of the functional states of the node, otherwise a functional error, such as a single bit error, could go undetected. See Figure 6.1.



Choose a GATE and a CLOCK to Sample All Functional States, and All Functional Errors Will Be Detected

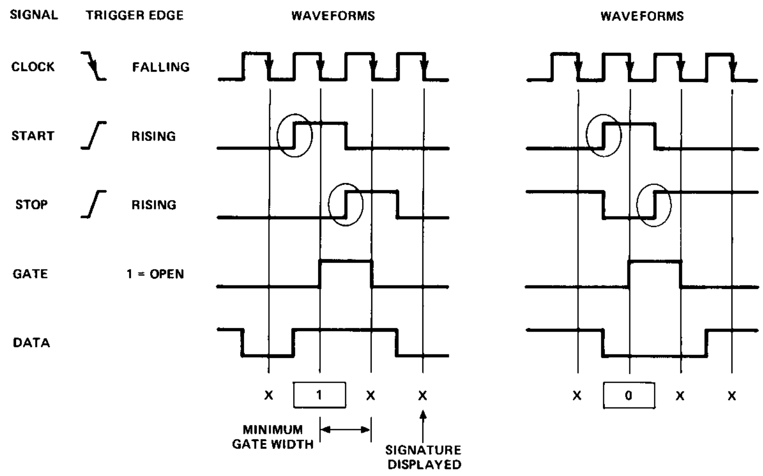
Figure 6.1

Single bit errors are guaranteed to result in a signature change for that node, as long as the error bit has been clocked into the signature analyzer. Multiple bit errors will result in a signature change for that node with a probability of 99.998%. In other words, functional failures in the device driving the node will be detected with a probability of 99.998%, worst case, assuming the error bits have been clocked into the signature analyzer, and that the device has been exercised through all of its functions.

It's unnecessary to artificially lengthen the GATE to increase the accuracy of error detection. The signature analyzer's accuracy of error detection is 100% for 16 or less data bits clocked into the analyzer. For more than 16 bits, the accuracy remains at 99.998%, worst case. This means that GATES can be as short or as long as necessary. Application Note 222-2 contains a complete discussion of the accuracy of error detection.

Minimum and Maximum Timing For An Open GATE

At least one CLOCK edge must occur between the START edge and the STOP edge, in order to open then close the GATE. The GATE opens for one CLOCK edge, while one data bit is clocked into the signature analyzer. The new signature will be displayed at the SECOND CLOCK edge after the STOP edge (i.e., after the GATE closes). See Figure 6.2.



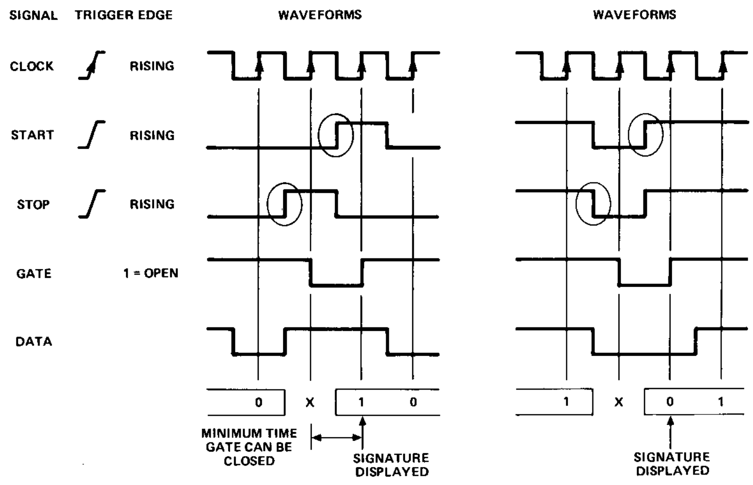
Two Examples Of GATE Open For Minimum Time

Figure 6.2

Any number of CLOCK edges can occur between a START edge and a STOP edge. There is no upper limit. Even the number of possible signatures (65,536) is not an upper limit. The signature analyzer will continue to gather a signature as long as the GATE is open. However, the new signature will not be displayed until the SECOND CLOCK edge after the STOP edge (i.e., not until the GATE closes).

Minimum and Maximum Timing For A Closed GATE

At least one CLOCK edge must occur between the last STOP edge and the next START edge, in order to close then open the GATE. The GATE will close for that CLOCK edge, and the data bit at that edge will not be used for the signature. The signature will be displayed at the SECOND CLOCK edge after the STOP edge (i.e., when the GATE opens again). See Figure 6.3.



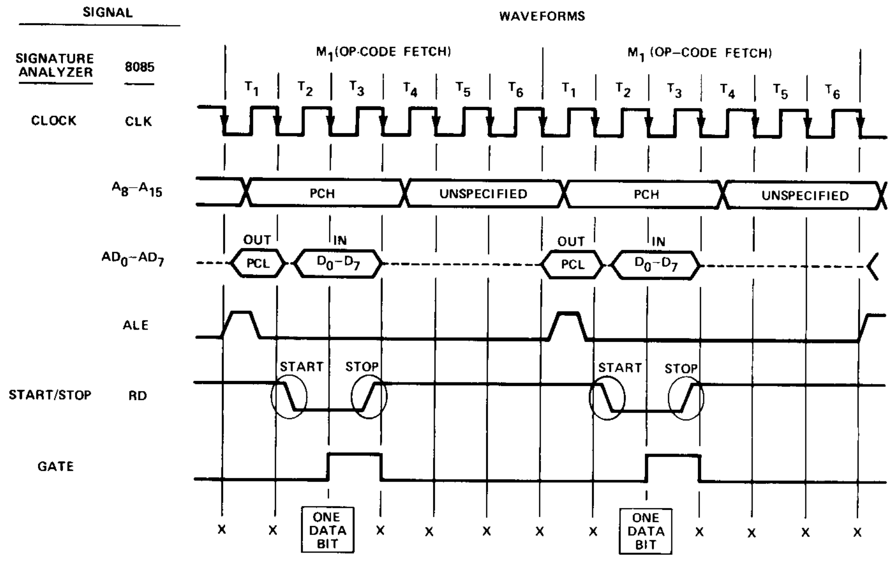
Two Examples Of GATE Closed For Minimum Time

Figure 6.3

Any number of CLOCK edges can occur between the last STOP edge and the next START edge. There is no upper limit. The signature analyzer will not gather a signature as long as the GATE is closed. The last signature will be displayed at the SECOND CLOCK edge after the STOP edge (i.e., after the GATE closed), and will continue to be displayed as long as the GATE is closed.

Creating A Synchronous Logic Probe

A minimum GATE of one CLOCK cycle allows the signature analyzer to be used as a synchronous logic probe. This can be very useful in determining the logic state of a node at an exact instant in time. The need for a synchronous logic probe measurement was discovered when trying to FREERUN microprocessors with multiplexed address and data buses, such as the Intel 8085. When the 8085 FREERUNS, an address is placed onto the multiplexed bus by the 8085 during the first half of each instruction fetch cycle. When the 8085 switches the bus to read the instruction, a NOP or similar instruction is forced onto the bus, so that the microprocessor simply increments to the next memory address, and so on. The address incrementation acts as a stimulus to memory devices within the system, so that signature analysis can be used for troubleshooting. (For a thorough discussion of FREERUN, see Application Note 222.) If there is a fault on the multiplexed bus, such as a short to ground, or short to another line, then two things could happen that would cause the 8085 not to stimulate the memory devices properly. First, the instruction could be changed by the fault, so that the microprocessor no longer increments the address bus sequentially. Second, the address lines could be changed by the fault such that they would not stimulate all memory devices. It would be easy to find the fault if it could be determined what instruction was being read into the microprocessor. To find out, it's necessary to separate address output, from data input, on the multiplexed bus. This can be done by sampling the bus only during the time the 8085 is reading it. Figure 6.4 shows how the signature analyzer can be used to display each bit of the instruction as if it were static on the bus. The signature will either be 0000 or 0001, corresponding to the logic state of the bus at the moment it was sampled.



The signature analyzer can be used as a synchronous logic probe to display each bit of the instruction as if it were static on the bus. Here, the 8085 is FREERUNNING, so the instruction is the same for each M_1 cycle.

Figure 6.4

For most other cases, GATE times will be longer than one bit. Otherwise, the node has not been allowed to cycle through all of it's possible functional states.

SECTION 7

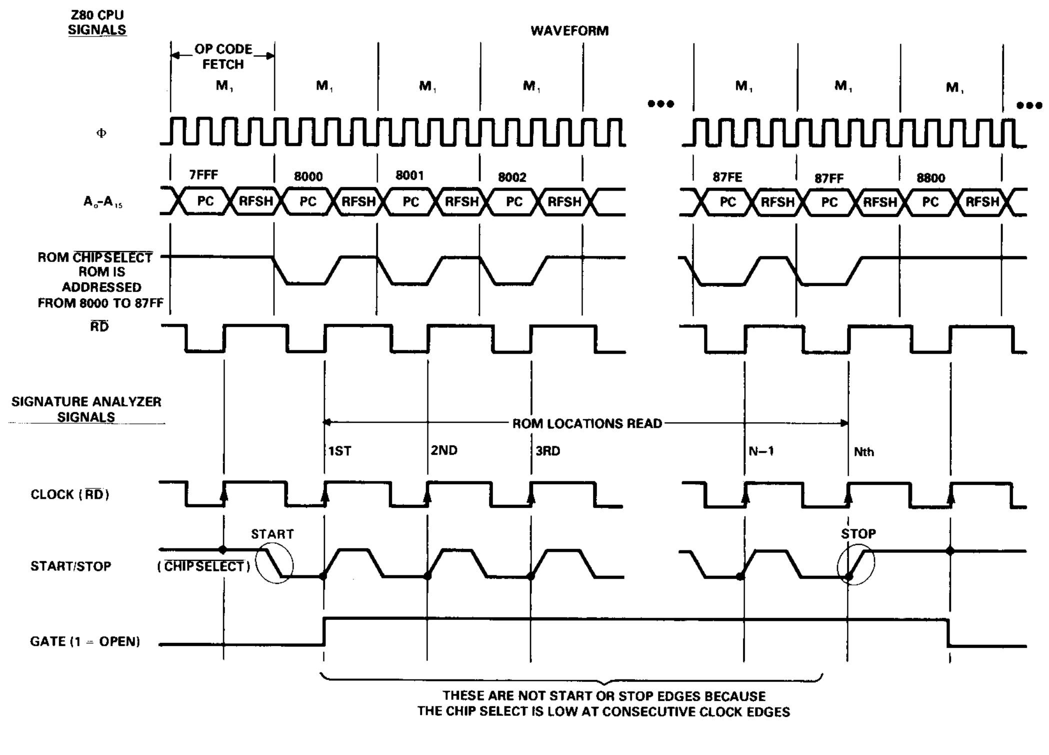
Multiple START and STOP Edges

If there is another START edge between any START edge and a STOP edge, then there are multiple START edges. Similarly, if there is at least one additional STOP edge between any STOP edge and a START edge, then there are multiple STOP edges. This assumes two things. First, that the START and STOP inputs are connected to separate signals. Second, that the START and STOP edges are detected by the CLOCK. The next paragraph shows how to determine if the second condition is true. If both conditions are true, then the first START edge after the GATE closes will open the GATE. Further START edges have no effect. The first STOP edge after the GATE opens will close the GATE. Further STOP edges have no effect. The GATE opens again at the first (next) START edge after the GATE closes. Resetting the signature analyzer, or using the HOLD feature, may affect the operation of the GATE. See Section Eight.

Are There Really Multiple START or STOP Edges?

Signals sometimes seem to have more than one edge when they really don't. For example, when FREERUNNING a microprocessor such as the Zilog Z80, START and STOP are frequently connected to ROM chip selects or their equivalents. This creates a GATE that's open only while ROM data is on the bus, allowing ROM's and associated circuits to be verified and troubleshot.

The chip selects become active at each address, and then inactive again during address changes. This creates multiple edges that at first seem to be multiple START and STOP edges. But when $\overline{RD}$ is used as a CLOCK, only one START and STOP edge actually occur. See Figure 7.1.



What Sometimes Seem To Be Multiple START/STOP Edges Really Aren't

Figure 7.1

SECTION 8

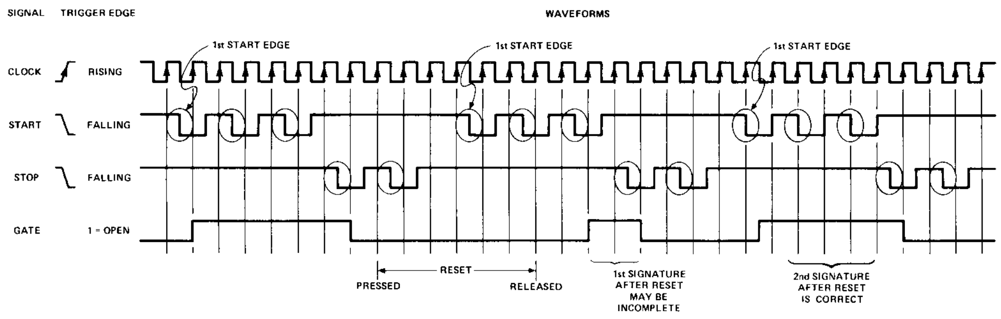
Reset and Hold

Resetting The Signature Analyzer

The HP Model 5004A Signature Analyzer is reset by pressing the RESET button on the DATA probe. The signature analyzer is usually reset for one of the following reasons:

1. It's a convenient way to cause the signature analyzer to display the V_{CC} (all ones) signature without touching the DATA probe to V_{CC}.
2. It's the only way to get the signature analyzer to take another signature while it's in the HOLD mode.
3. It's a way to reset the display and close the GATE, just after placing the DATA probe on a node, so that the next signature displayed is the correct one. This is usually done when GATE cycle times are long.

When there are multiple START edges, the first signature displayed after reset may not be the correct one. As shown in Figure 8.1, when the signature analyzer is RESET, the GATE closes, and will remain closed, until the next START edge. If the GATE opens again at the first of multiple START edges, then the first signature will be a correct (complete) one. If the GATE reopens at one of the multiple START edges other than the first one, then the first signature will be a partial (incorrect) one. In either case, the second signature after RESET will always be correct (complete), assuming the GATE repeats and the signature analyzer is not in the HOLD mode.



The First Signature After RESET May Be Incorrect
If There Are Multiple START Edges

Figure 8.1

The HOLD Feature

The HOLD feature causes the HP 5004A Signature Analyzer to take only ONE signature after RESET. To use the HOLD feature, press the HOLD button IN. Place the DATA probe on the node, and then press RESET for a new signature. The next signature displayed will be held in the display, even if the probe is removed from the node. This feature is handy when the probe must be removed from the node temporarily to see the display. This can happen when troubleshooting a product that's in an awkward location. The HOLD feature also makes it convenient to document signatures in the troubleshooting procedure. After the signature is captured in the display, the probe can be set down so that the signature can be written. HOLD can even be used to capture signatures of single-cycle events that occur when the product is first turned on. Note that a second signature is not taken in the HOLD mode. If the HOLD feature is intended to be used, then START signals must have only one edge. If this is done, then the "held" signature will always be correct (complete).

Getting Correct, Repeatable, and Stable Signatures

While creating an SA circuit stimulus, a few circuit conditions can cause unstable or unrepeatable signatures. This section outlines their causes and how to recognize them. The remaining guidelines show how to eliminate them from the SA stimulus design so that incorrect signatures measured during troubleshooting accurately indicate a fault.

Correct and Incorrect Signatures

Correct signatures are defined as the signatures documented in the troubleshooting procedure for a product. An incorrect signature is defined as a signature displayed on the signature analyzer, that doesn't match the documented one for the node being probed. If the correct signatures are repeatable, then measuring an incorrect signature during troubleshooting will accurately indicate an incorrect waveform for that node. This allows the troubleshooter to quickly back-trace through the circuit following incorrect waveforms to the faulty node. The fault is found at the point where back-tracing further results in correct signatures again. For instance, if the signature for an output of a device is incorrect, signatures are taken on the inputs for that device. If the input signatures are correct, then the fault has been isolated to the output node. If any input signature is incorrect, back-tracing continues along that signal path.

A troubleshooter is concerned only if the signatures are correct or incorrect. He must believe that the signatures documented for the product are correct. That is, he assumes that a known good product will yield all of the signatures as documented in the troubleshooting procedure. The person that creates and checks the signature analysis stimulus is concerned about assuring that signatures to be documented in the troubleshooting procedure are really correct. Correct signatures must be repeatable.

The creator of the stimulus also is concerned about whether all possible product faults will yield incorrect signatures. Two things determine this. First, the accuracy of error detection of the measurement, and second, the extent of the functional stimulus of the product. The signature analysis measurement gives a 99.998% worst-case probability of detecting an error, as long as the error appears in the waveform for that node, and the error bit(s) have been clocked into the signature analyzer. The error will appear in the waveform only if the functional stimulus causes it to be there.

Repeatable and Unrepeatable Signatures

Repeatable signatures are defined as getting the same signature each and every time the signature measurement is made, no matter when it is made, if:

1. Identical nodes in the product are measured.
2. Identical known good products are tried.
3. Identical signature measurements are performed.
4. Identical signature analysis stimulus is run.
5. Identical signature analyzers are used (i.e., signature analyzers with compatible characteristics. See Section 2 for a description of the characteristics.)

Unrepeatable signatures are defined as getting a different signature any time the same node in a known good product is measured, in the same way, when running the same stimulus. One of the goals of the signature analysis circuit stimulus creator, is to eliminate unrepeatable signatures from the product.

Making Sure Signatures Are Repeatable

Unrepeatable signatures usually occur because some circuit element, such as a flip-flop or RAM, has not been initialized before the SA stimulus begins. They also can occur if the GATE toggles open at the wrong time. The following steps will help uncover repeatable signatures in a product. They should be performed to verify ALL signatures in a product BEFORE they are considered correct.

1. Turn the product under test off then on again.
2. Reset the signature analyzer, or touch the DATA probe to ground or V_{CC} before measuring a node.
3. Take signatures in as many other known good products as time and effort permits.

Turning the product under test off then on again determines if any devices are not being initialized before the SA stimulus begins. For example, a flip-flop that is not set or reset at power-on or at the beginning of the SA stimulus loop can cause signatures to differ. RAM also must be initialized with some known pattern before the SA stimulus begins if any nodes associated with that RAM are measured during the execution of the stimulus.

Resetting the signature analyzer determines two things. First, that the GATE toggles open at the intended time. See Section Five. Second, that the node being sampled is always in the third state. If a node is always in the third state, the signature will either be all zeroes, or the V_{CC} signature, depending on the last valid logic state of the previous node measured. If an all zeroes signature has been documented for the three-state node, WITHOUT a notation that the signature could ALSO be the V_{CC} signature, then the signature will appear to be unrepeatable. (This is a special case where the signature is unrepeatable, but can only be one of two possible signatures.) Reset causes the signature analyzer to use a logic one as the last valid logic state. This is the same as if the probe had been touched to V_{CC} before measuring the three-state node. See Sections Eight and Eleven.

Taking signatures in as many known good products as possible determines if all uninitialized devices have been eliminated from the signature measurement. It also determines if all recommended operating conditions of the signature analyzer have been met. Using another signature analyzer can also help determine this. For example, if the specification for the setup time is being violated, but the signature analyzer actually is performing better, then another signature analyzer may help uncover this. See Appendix A for setup and hold times.

Here's an example of a situation that can cause unrepeatable signatures. On the data bus of a FREERUNNING microprocessor system, there may be data from ROM, RAM and I/O devices. The ROM data is always the same, but RAM data will differ each time the system is turned off then on again. This is because the processor has no way to initialize RAM by storing a pattern in it. Therefore, the data from RAM and other uninitialized devices must be eliminated from a signature measurement of the data bus. This will result in the same DATA (ROM data only) being measured each time the GATE is open and therefore results in repeatable signatures. This can be done in two ways. One way is to move START and STOP to ROM chip selects (or their equivalents), to create a GATE that is open only while ROM data is on the bus. Another way to do this is to choose a CLOCK that samples data on the bus only when ROM data is present.

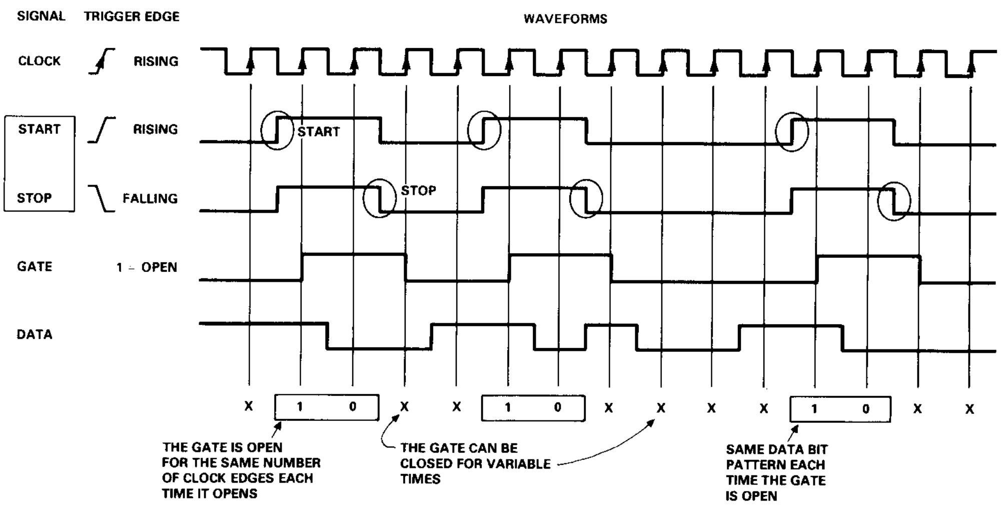
Stable and Unstable Signatures

The definition of stable and unstable signatures depends on the measurement of two signatures in a row. Stable signatures occur when two adjacent signature measurements result in the same signature for all measurements. Unstable signatures occur when the signatures of any two adjacent measurement cycles differ. An unstable signature is indicated by the UNSTABLE SIGNATURE LIGHT. The light turns on for approximately 100 milliseconds whenever a signature taken during ANY GATE cycle differs from the signature taken during the immediately previous GATE cycle. If unstable signatures occur for every GATE cycle, the light will blink at $\leq$10 Hz. Unstable signatures are not of concern while the signature analyzer is in the HOLD mode. This is because only one signature is taken after RESET is pressed. However, the UNSTABLE LIGHT will flash as the signature display changes from 0000 after RESET, to a captured signature other than 0000.

Sometimes the four-digit signature display will indicate an unstable signature by slowly and/or constantly changing, or changing so fast that it becomes unintelligible. However, with fast GATES and an intermittant circuit fault, intermittantly unstable signatures can occur without a noticeable signature display change. In those cases, the UNSTABLE SIGNATURE LIGHT will be the only indication of an unstable signature and the intermittant circuit.

Eliminating Unstable Signatures

Unstable signatures usually occur because the GATE length created by the stimulus is not the same from loop to loop, or the response of the circuit to the stimulus is not the same from loop to loop. For stable and repeatable signatures, the GATE must be open for the same number of CLOCK edges each time it opens. The DATA bits sampled by CLOCK edges during an open GATE must also be identical (same logic levels and time relationship) from one open GATE to the next. While the GATE is closed, the number of CLOCK edges can vary as well as the DATA pattern. See Figure 9.1. To check DMA cycles that transfer the same data each time, connect START to a line that signals when DMA begins. In a Zilog Z80-based system, this might be HOLD ACKNOWLEDGE. This control line from the Z80 signals to an external device, such as a DMA controller, that it now can take control of the bus. Connect STOP to a similar signal that occurs at the end of the DMA cycle. See Figure 9.1.


SIGNAL
TRIGGER EDGE
WAVEFORMS
CLOCK
RISING
START
RISING
START
STOP
FALLING
STOP
GATE
1 = OPEN
DATA
X
1
0
X
X
1
0
X
X
X
X
1
0
X
X
THE GATE IS OPEN FOR THE SAME NUMBER OF CLOCK EDGES EACH TIME IT OPENS
THE GATE CAN BE CLOSED FOR VARIABLE TIMES
SAME DATA BIT PATTERN EACH TIME THE GATE IS OPEN

Some Conditions For Stable and Repeatable Signatures

Figure 9.1

Unstable signatures can sometimes be documented for nodes if the signature changes by only a few numbers. For instance, if a signature is unstable, but only changes between two numbers, then both can be documented. Then, either one measured during troubleshooting will indicate correct circuit function, while any other signature will indicate a fault.

SECTION 10

About Noise

Most signal noise does NOT result in incorrect, unstable, or unrepeatable signatures, because the signature analyzer synchronously detects logic state changes on START, STOP and DATA at CLOCK edges. This section shows the effects of two different types of noise, defined below, on each of these four inputs.

Synchronous Noise Definition

Synchronous noise is any noise on a signal that can be predicted with respect to a CLOCK edge. It is usually caused by clocking data synchronously, through a digital system. It consists of momentary transitions, or multiple logic state changes, through the logic threshold opposite to the signal's defined stable state, between two CLOCK edges. These transitions occur after the CLOCK edge that caused them, but are absent at the next CLOCK edge. Here, "at the CLOCK edge" means the signal is at its defined state during the data setup time of the signature analyzer.

Asynchronous Noise Definition

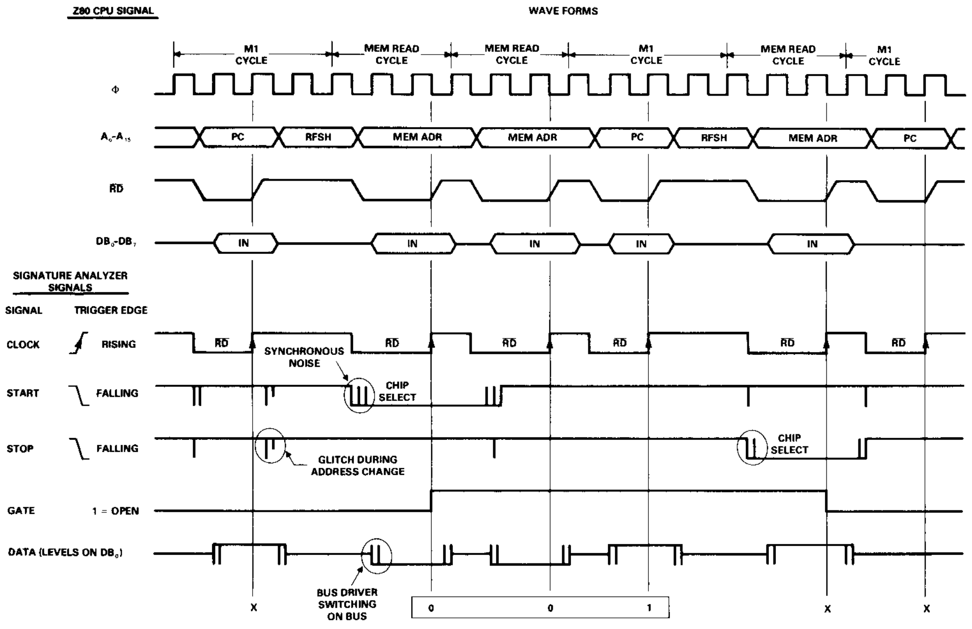
Asynchronous noise is any noise on a signal that occurs randomly with respect to the CLOCK. This noise is usually referred to as a "glitch" and can occur even at a CLOCK edge. It consists of momentary transitions, or multiple logic state changes, through the logic threshold opposite to the signal's defined stable state, between two CLOCK edges. These transitions will be detected as an incorrect logic state only if they occur exactly at the CLOCK edge of the signature analyzer.

Noise On START, STOP and DATA

Synchronous noise on START and STOP is ignored by the signature analyzer, as shown in Figure 10.1. This is a real advantage when using address decoders or chip selects as START and STOP, even though they frequently are noisy during address changes. But since a CLOCK can be used that occurs only after the address bus is stable, the noise does not get detected and will not open or close the gate incorrectly at the noise edges. The gate opens or closes only at a CLOCK edge when the START or STOP edge has been synchronously detected.

Synchronous noise on the DATA input is also ignored. This is an advantage when taking signatures on high-current data bus lines which are noisy while drivers switch on and off the bus. Only valid logic states at CLOCK edges will be used as DATA bits for the signatures.

Asynchronous noise on START, STOP and DATA can cause incorrect, unstable, or unrepeatable signatures if it occurs at the CLOCK edge. This is because it can violate the setup time of the CLOCK as defined in the Recommended Operating Conditions of Appendix A.



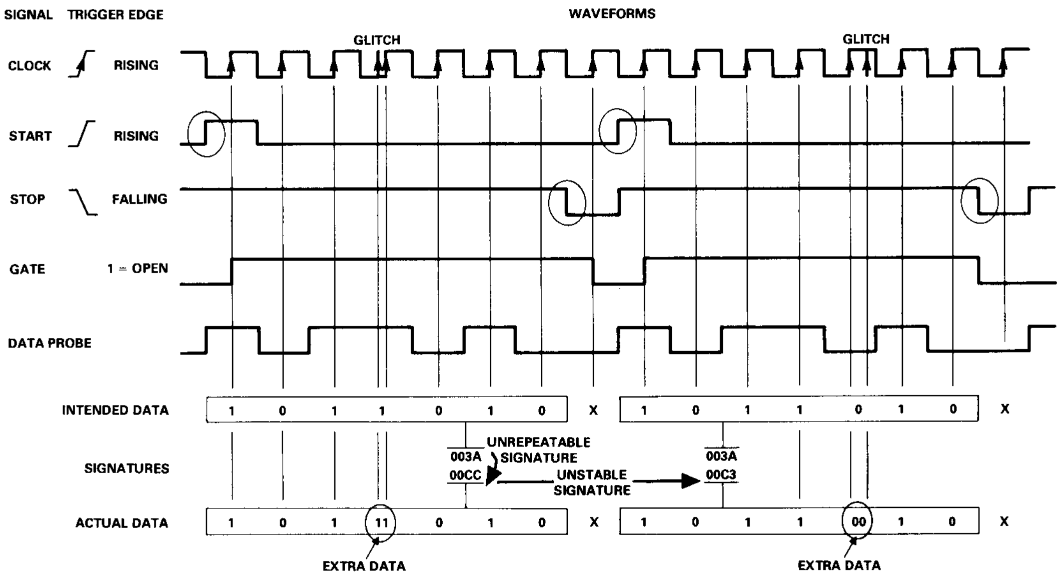
Synchronous Noise Is Ignored In This Z80-Based Microprocessor System

Figure 10.1

CLOCK Noise

Any transition through the single logic threshold of the CLOCK input is defined as a CLOCK edge. Synchronous or asynchronous noise on the CLOCK results in extra CLOCK edges to the signature analyzer, as shown in Figure 10.2.

If the extra CLOCK edges occur while the gate is closed, then the signature may or may not be affected depending on the activity on START. If the extra CLOCK edges do not detect a START edge, then the gate will remain closed and the signature will be correct. However, if the extra CLOCK edges do accidentally detect a START edge, then the gate will open early. The gate length will then vary resulting in an unstable or unrepeatable signature. If the extra CLOCK edges occur while the gate is open, the length of the gate will change, resulting in unstable or unrepeatable signatures.



Noise On The Clock Input Results In Unstable and Unrepeatable Signatures

Figure 10.2

Ground Noise

The ground leads of the signature analyzer should be connected to circuit grounds that are free of excessive noise. Normally only the GATING POD ground lead need be connected to the circuit under test. However, when measuring high-frequency DATA (usually above 5 MHz), an additional ground lead connected to the DATA probe ground point will help reduce any noise induced through a long ground lead path.

About Three-State Nodes

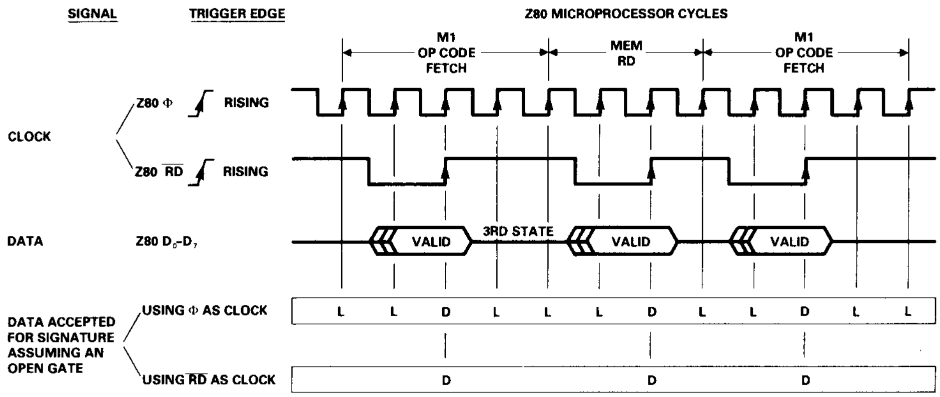
When three-state nodes are probed with a Hewlett-Packard Signature Analyzer, stable and repeatable signatures can be obtained, even though the TTL level of the node is not defined during the third state. Two things allow this to happen:

1. A CLOCK can be chosen to sample data on a three-state node only when it is in a defined state. When using such a CLOCK, the node is NOT sampled during the third state.
2. If such a CLOCK is not used, then the DATA probe and the front-end logic of the signature analyzer will define a third state sampled by the CLOCK as the last valid logic level on the node before it went into the third state.

Here's the details.

Using A CLOCK That Samples Only Defined DATA

In most applications, it's easy to choose a CLOCK that samples a three-state node only when valid data is present, and not when the node is in the third state. For example, a microprocessor's data bus can enter the third state between op code fetches, memory references, and I/O operations as the bus direction is turned around from writing to reading a device. Usually there is a control output from the microprocessor (e.g., a read or write line), that defines when data is valid on the bus. Using this control line as the CLOCK synchronizes the signature analyzer, so that only valid logic levels on the bus are sampled and third-state levels are ignored. See Figure 11.1.



A "D" INDICATES VALID DATA (EITHER A LOGIC HIGH OR LOGIC LOW) IS PRESENT ON THE DATA BUS AND IS SAMPLED BY THE SIGNATURE ANALYZER.

AN "L" INDICATES DATA WILL BE THE LAST VALID LOGIC STATE WITH NO PULLUPS ON THE NODE, OR A LOGIC HIGH IF PULLUPS ARE ON THE NODE.

A Clock Can Be Chosen To Sample DATA On a Three-State Node Only When It Is In a Defined State

Figure 11.1

Using A CLOCK That Samples A Node During The Third State

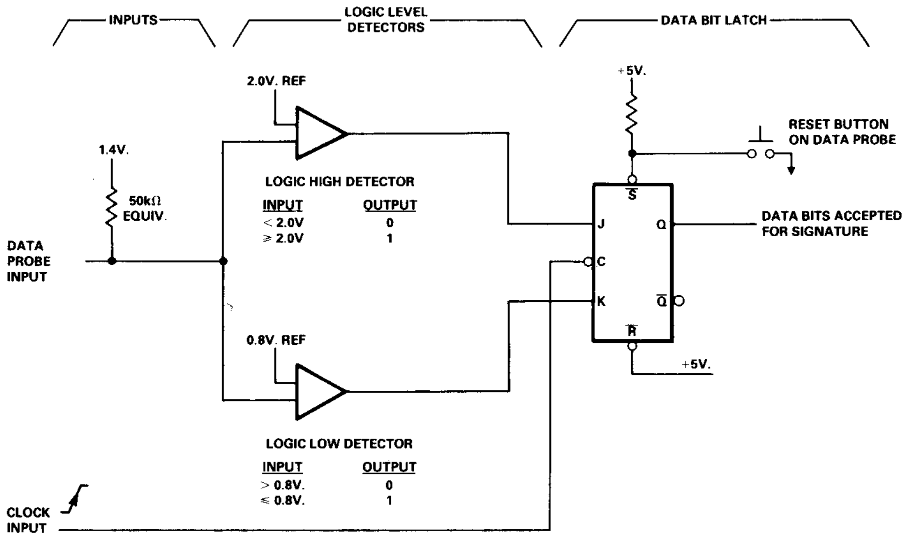
Sometimes it's necessary to use a CLOCK that samples a node when it's in the third state. This can happen when a synchronous CLOCK is not available or when a node remains in the third state for a long time. The signature then depends on whether or not there's an external pullup (or pulldown), on the node.

Signatures For Three-State Nodes Without Pullups

Stable and repeatable signatures can be obtained on three-state nodes without pullups using a Hewlett-Packard Signature Analyzer. A pullup does not need to be added to the node during system design, nor does a pullup need to be added to the DATA probe during troubleshooting. This is because the signature analyzer will continue to use the last valid logic state as a DATA bit when the node enters the third state. There are three parts to the signature analyzer that allow this to happen:

1. An internal reference inside the DATA probe pulls the node to 1.4 volts during the third state.
2. Dual threshold logic level detectors that define logic LOW as 0.8 volts or less, logic HIGH as 2.0 volts or greater, and anything in between LOW and HIGH as the third state.
3. A clocked DATA bit latch.

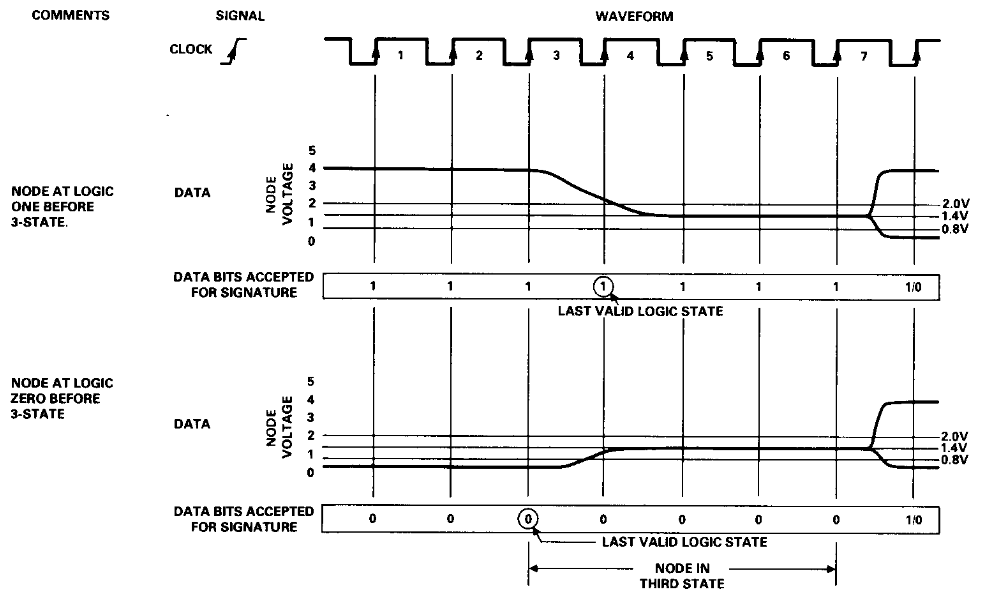
Figure 11.2 shows a simplified diagram of the input circuit of the HP Model 5004A Signature Analyzer. The DATA bits accepted for a signature are latched into a flip-flop at each CLOCK edge. If the voltage at the DATA probe input is 2.0 volts or greater, the logic level converter on the J input to the flip-flop causes the flip-flop to set to logic one at the next CLOCK edge. If the probe voltage is 0.8 volts or less, the converter on the K input causes the flip-flop to reset to logic zero at the next CLOCK edge. If the probe voltage is between 0.8 volts and 2.0 volts, neither logic level converter is active and the flip-flop neither sets nor resets.



Front-End For the HP Model 5004A Signature Analyzer

Figure 11.2

When the node enters the third state, the DATA probe's internal reference pulls the node to 1.4 volts. If the last valid logic state of the node was a high, then the reference pulls the node from logic high to 1.4 volts during the third state and does not allow the node's voltage to drop into the logic zero detection level of 0.8 volts or less. Similarly, a logic low is pulled up to 1.4 volts and a logic high is never detected. This is shown in Figure 11.3. In other words, the last valid logic state remains in the DATA latch and continues to be used for a signature while the node is in the third state. If a node remains in the third state during the entire measurement, then the signature will be all zeroes or the V_{CC} signature depending on whether the last bit on the last node measured was a one or a zero.



Repeatable Signatures Result On Three-State Nodes Without Pullup Resistors

Figure 11.3

Getting Vcc Signatures Without Probing Vcc

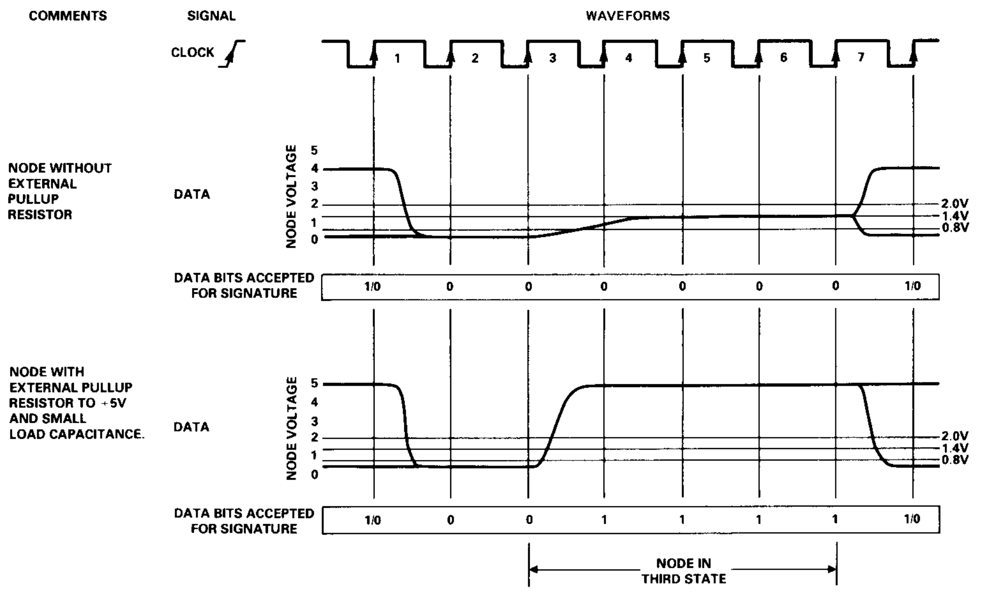
The RESET button on the DATA probe sets the DATA latch to a logic one so that if the probe is left floating (no node is being probed or the node is always in the third state), then all ones are clocked into the signature analyzer as if the probe were placed on +5VDC. This results in a Vcc (or +5V, or all ones) signature.

Noise On A Three-State Node

The signature analyzer's treatment of three-state nodes without pullups has been optimized, and proven with several years' experience. The DATA probe input has high enough impedance to reduce circuit loading of the node during the active state and low enough impedance to reduce most noise induced on the node during the third state. This treatment results in stable and repeatable signatures in most cases. However, because of their high impedance, three-state nodes are susceptible to excessive noise induced through crosstalk with high current carrying lines. Although unlikely, excessive noise could cause the logic threshold detectors to switch states exactly at a CLOCK edge and cause an erroneous DATA bit to be sampled resulting in an incorrect, unstable or unrepeatable signature. Section Ten shows how noise on any input affects the signature measurement.

Signatures For Three-State Nodes With Pullups

Stable and repeatable signatures can also be obtained for three-state nodes with pullup resistors in most cases. This is because the pullups on three-state nodes will override the signature analyzer's internal reference and pull the node up to logic high during the third state. This results in logic ones being used as DATA bits during the third state instead of the last valid logic state, as was the case for a node without pullups. See Figure 11.4.


COMMENTS
SIGNAL
WAVEFORMS
CLOCK
1
2
3
4
5
6
7
NODE WITHOUT EXTERNAL PULLUP RESISTOR
DATA
NODE VOLTAGE
5
4
3
2
1
0
2.0V
1.4V
0.8V
DATA BITS ACCEPTED FOR SIGNATURE
1/0
0
0
0
0
0
0
1/0
NODE WITH EXTERNAL PULLUP RESISTOR TO +5V AND SMALL LOAD CAPACITANCE.
DATA
NODE VOLTAGE
5
4
3
2
1
0
2.0V
1.4V
0.8V
DATA BITS ACCEPTED FOR SIGNATURE
1/0
0
0
1
1
1
1
1/0
NODE IN THIRD STATE

Repeatable Signatures Can Be Obtained On Three-State Nodes With Pullups and Small Load Capacitance

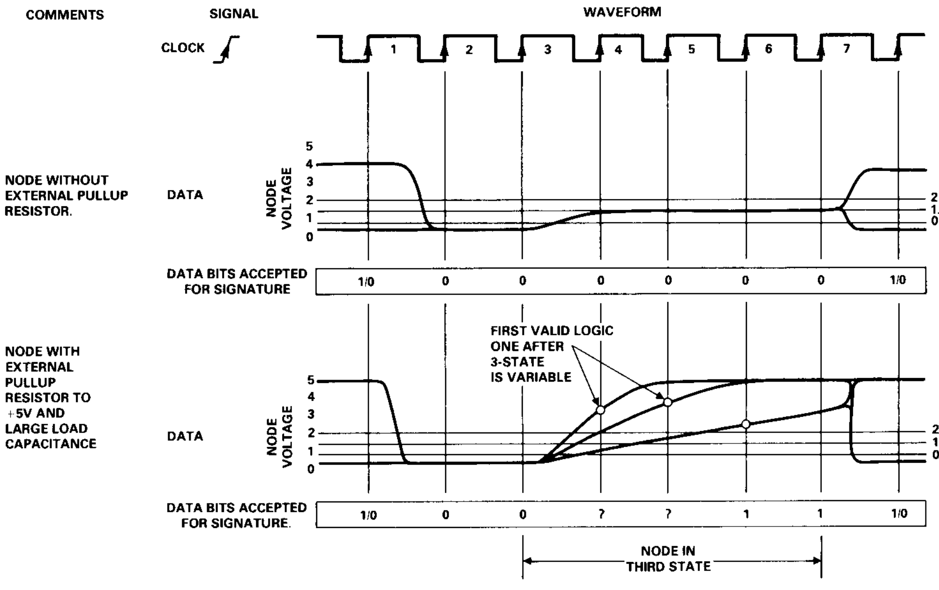
Figure 11.4

However, it's not always possible to add pullups to a three-state node. This may be because the pullups may load the circuit too much when the node is active. Or perhaps the pullups were not designed into the circuit and they cannot be added later. If there are pullups on the node, or if an external pullup is placed on the DATA probe of the signature analyzer, be careful that the signal risetimes remain fast enough.

Nodal Capacitance And Pullups Can Cause Slow Risetimes

Without pullups on the node, it doesn't matter how long it takes for the DATA probe to pull the node to 1.4 volts. Risetimes are not a concern since the DATA bit latch will not change state during the third state. However, external pullups or pulldowns on the node, combined with large nodal capacitance, can cause slow signal risetimes to logic one (or fall times to zero), resulting in erroneous DATA being detected. This can cause incorrect, unstable, or unrepeatable signatures in some cases.

In the case of pullups on the node, the node's voltage will eventually be detected as a logic high during the third state, but the point at which the data will be ones instead of zeroes depends on how fast the node's voltage rises to +5VDC. This is shown in Figure 11.5. The node's RC time constant will determine the rise time. If the risetime is always the same, or if the signal is guaranteed to rise above the logic high threshold before the next CLOCK edge (including time for setup), then a stable signature will result. However, the RC time constant depends on a fixed pullup resistor (if it is designed into the board), or a variable one (if it is placed on the DATA probe during troubleshooting). The RC time constant also depends on the node capacitance that could vary from unit to unit, or even on the same unit over time, or if a part is replaced. The capacitance can also vary, particularly on a large bus, due to the number of devices on the bus, the variable capacitance of each device due to manufacturing variations, the bus length, including backplanes and optional PC boards, and so on.


COMMENTS
SIGNAL
WAVEFORM
CLOCK
1
2
3
4
5
6
7
NODE WITHOUT EXTERNAL PULLUP RESISTOR.
DATA
NODE VOLTAGE
5
4
3
2
1
0
2.0V
1.4V
0.8V
DATA BITS ACCEPTED FOR SIGNATURE
1/0
0
0
0
0
0
0
1/0
NODE WITH EXTERNAL PULLUP RESISTOR TO +5V AND LARGE LOAD CAPACITANCE
FIRST VALID LOGIC ONE AFTER 3-STATE IS VARIABLE
DATA
NODE VOLTAGE
5
4
3
2
1
0
2.0V
1.4V
0.8V
DATA BITS ACCEPTED FOR SIGNATURE.
1/0
0
0
?
?
1
1
1/0
NODE IN THIRD STATE

It's Harder To Get Repeatable, Stable Signatures On Three-State Nodes With Pullups and Large Node Capacitance

Figure 11.5

If unstable or unrepeatable signatures result on a node with a pullup resistor, first verify that a slow risetime is the problem. If it is, then try to find a CLOCK that will sample only valid DATA on the node and will ignore the third state. If this is not possible, then calculate the node's worst case capacitance and choose a pullup resistor that will insure the node's voltage will be above 2.0 volts before the next CLOCK. Be sure all devices on the bus have sufficient drive capability with the new resistor.

If risetime does not seem to be a problem, then check if there is excessive noise synchronous to the CLOCK. If so, eliminate the source of the noise, or choose a CLOCK that will ignore it.

APPENDIX A

Hewlett-Packard Model 5004A Signature Analyzer

Any edge for CLOCK, with any trigger edge selected for START or STOP, in any combination			Min.	Nom.	Max.	Unit	See Fig.
Setup time, t_{s-}	t_{ss}	START	25			ns	A.1
		STOP	25				
	t_{sd}	DATA	. 15				
Hold time, t_{h-}	t_{hs}	START	0				
		STOP	0				
	t_{hd}	DATA	0				
Clock frequency			0		10	MHz	
Width of CLOCK pulse, t_w high or low			50			ns	
Input impedance to 1.4V		START					
		STOP		50		kΩ	
		CLOCK		7		pf	
		DATA					
V_{ih} High-level input voltage		DATA	1.6	2.0		V	A.2
V_{il} Low-level input voltage				0.8	1.2		
V_{ref} Single logic threshold range		START					A.3
		STOP	0.8	1.4	2.0	V	
		CLOCK					
Hysteresis band around single logic threshold		START					
		STOP		.1		V	
		CLOCK					
Overload protection, all inputs		continuous	−150		150		
		intermittent	−250		+250	V	
		≤1 min.		250		VAC	

Table A.1—Recommended Operating Conditions

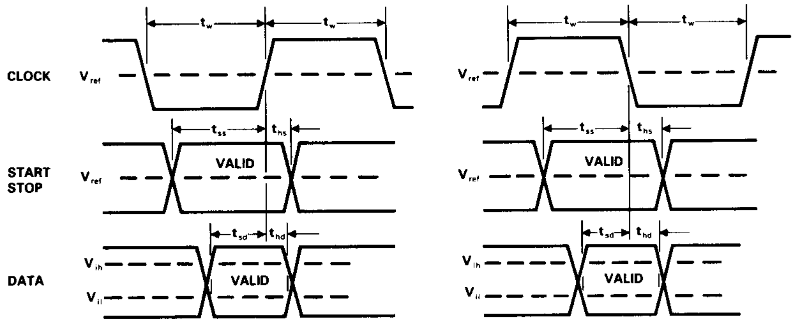
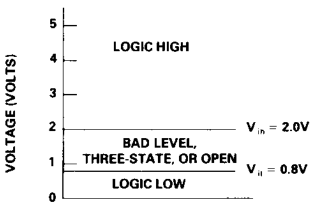


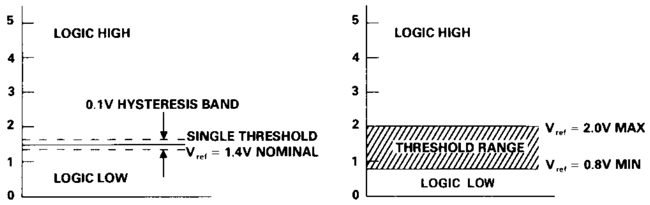
Figure A.1—Voltage Waveforms



THE DATA PROBE INPUT DETECTS TTL LOGIC LEVELS AS FOLLOWS

LOGIC LEVEL	VOLTAGE RANGE	PROBE LAMP	BIT FOR SIGNATURE
LOW	$V \leq 0.8$	OFF	0
HIGH	$V \geq 2.0$	BRIGHT	1
THREE-STATE, BAD LEVEL OR OPEN CIRCUIT	$0.8 < V < 2.0$	DIM	LAST VALID DATA BIT

Figure A.2—Data Probe Dual Thresholds (Nominal)



THE NOMINAL SINGLE THRESHOLD LEVEL IS 1.4V, BUT THE LEVEL CAN RANGE FROM 0.8V TO 2.0V. THERE IS A 0.1V HYSTERESIS BAND AROUND THE ACTUAL THRESHOLD LEVEL.

Figure A.3—START, STOP and CLOCK Single Thresholds

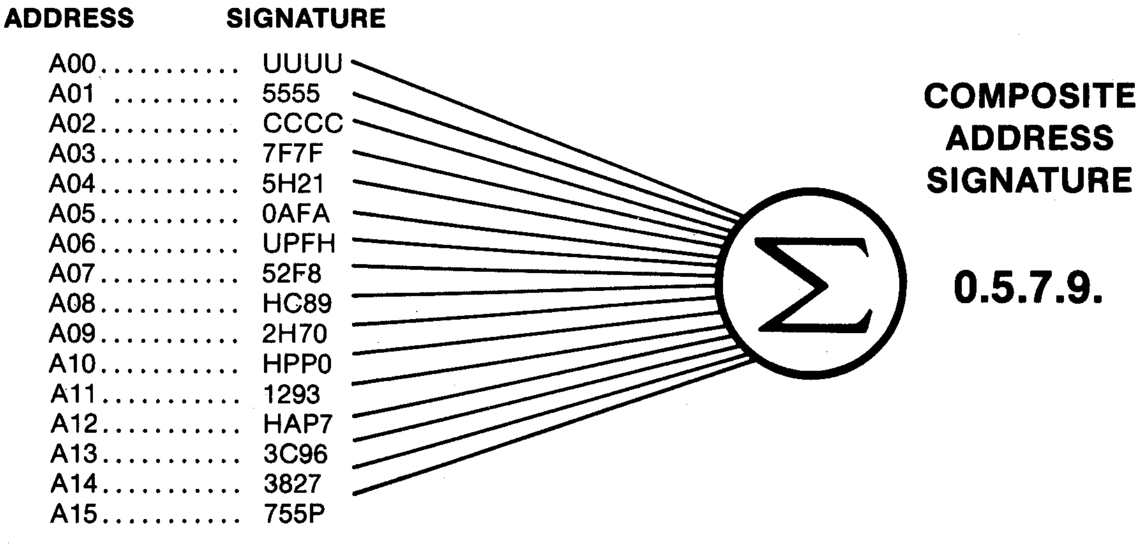
HEWLETT PACKARD

For more information, call your local HP Sales Office or nearest Regional Office: **Eastern** (201) 265-5000; **Midwestern** (312) 255-9800; **Southern** (404) 955-1500; **Western** (213) 970-7500; **Canadian** (416) 678-9430. Ask the operator for instrument sales. Or write Hewlett-Packard, 1501 Page Mill Road, Palo Alto, CA 94304. **In Europe**: Hewlett-Packard S.A., 7, rue du Bois-du-Lan, P.O. Box, CH 1217 Meyrin 2, Geneva, Switzerland. **In Japan**: Yokogawa-Hewlett-Packard Ltd., 29-21, Takaido-Higashi 3-chome, Suginami-ku, Tokyo 168.

5952-7595 **reprinted MARCH 1991** **PRINTED IN USA**

Application Note 222-6

Troubleshooting with Composite Signatures



ADDRESS	SIGNATURE
A00	UUUU
A01	5555
A02	CCCC
A03	7F7F
A04	5H21
A05	0AFA
A06	UPFH
A07	52F8
A08	HC89
A09	2H70
A10	HPP0
A11	1293
A12	HAP7
A13	3C96
A14	3827
A15	755P

COMPOSITE
ADDRESS
SIGNATURE

0.5.7.9.

HEWLETT
PACKARD

Application Note 222-6

"Troubleshooting with Composite Signatures"

Composite Signature is the binary sum of individual signatures. Composite signature saves the troubleshooter time by reducing the amount of visual comparisons of the signature analyzer display to printed signature tables. Any grouping of digital signals can be chosen to form a composite signature. Composite IC or Bus signatures are good examples.

This application note explains how composite signature and the backtracing algorithm can be used to implement a structured troubleshooting procedure without a computer-aided system. The result is time savings for the logic troubleshooter.

FOREWORD

ABOUT DIGITAL TROUBLESHOOTING

Microprocessors have revolutionized your product line. Your products are smarter, faster, friendlier and more competitive because they take advantage of μP-based control and computation. They are also harder to build, harder to test and harder to fix when they fail. Complex bus structures and timing relationships have practically obsoleted the scope/voltmeter and signal tracing techniques so effective on analog products. The need to enhance the testability and serviceability of your digital products is acute. So is the need for specialized digital troubleshooting equipment.

ABOUT SIGNATURE ANALYSIS

To address these needs, Hewlett-Packard has developed the Signature Analysis technique, as well as a Signature Analyzer product line, for component-level troubleshooting of microprocessor-based products. A Signature Analyzer detects and displays the unique digital signatures associated with the data nodes in a circuit under test. By comparing these actual signatures to the correct ones, a troubleshooter can back-trace to a faulty node. By designing S.A. into digital products, or stimulating them externally, a manufacturer can provide manufacturing test and field service procedures for component-level repair, without dependence on expensive board-exchange programs.

ABOUT THIS PUBLICATION

This application note shows how composite signature can be used to implement a structured troubleshooting procedure without a computer-aided system. The backtracing algorithm is explained and flowcharted. Examples show how to measure and calculate composite signature. Potential time savings is estimated. Theory of operation and probability of error detection are derived in the appendices.

ABOUT OTHER PUBLICATIONS

Application Note 222-0, "An Index to Signature Analysis Publications" lists all other application notes currently available in the AN 222 series about Signature Analysis. They cover a wide range of interests, from how to design or retrofit Signature Analysis into digital systems, to the cost reductions that can be expected in production test and field service by doing so. It also lists all data sheets for the complete line of Hewlett-Packard Signature Analysis products, plus other related publications about digital troubleshooting.

TABLE OF CONTENTS

INTRODUCTION

Composite signature **magnifies** the power of Signature Analysis by providing a single signature that verifies the correct operation of an IC, microprocessor bus or digital circuit. Rough estimates show that one can find the first bad signature ten times faster than by taking individual signatures. The time saved is realized when making manual comparisons of measured signatures to the expected ones recorded on paper. One simply probes the individual nodes and then only compares the composite signatures to detect the presence of individual faulty waveforms. Of course, with a computer-aided system there is no advantage because comparisons of signatures to memory are done automatically. Therefore, composite signature is primarily used in manual troubleshooting applications.

Figure 1. Composite signature saves time by reducing the amount of visual comparisons the troubleshooter must make to printed documentation. The HP 5006A Signature Analyzer is the first signature measurement instrument to offer this capability.

A STRUCTURED APPROACH TO FINDING FAULTS

The composite signature function now gives digital troubleshooting a **structured approach**. The objective of digital troubleshooting is to find the **source** of the faulty signals being propagated in a circuit. A fault source could be a bad integrated circuit, component, solder bridge of faulty trace. Figure 2 summarizes the steps taken to find the source of faulty signals on a digital circuit board. Composite signature aids the troubleshooter who does not have access to a computer-aided troubleshooting system by making the process faster and easier. The instrument used as an example in this note is the HP 5006A Signature Analyzer, the first signature measurement instrument to offer composite signature.

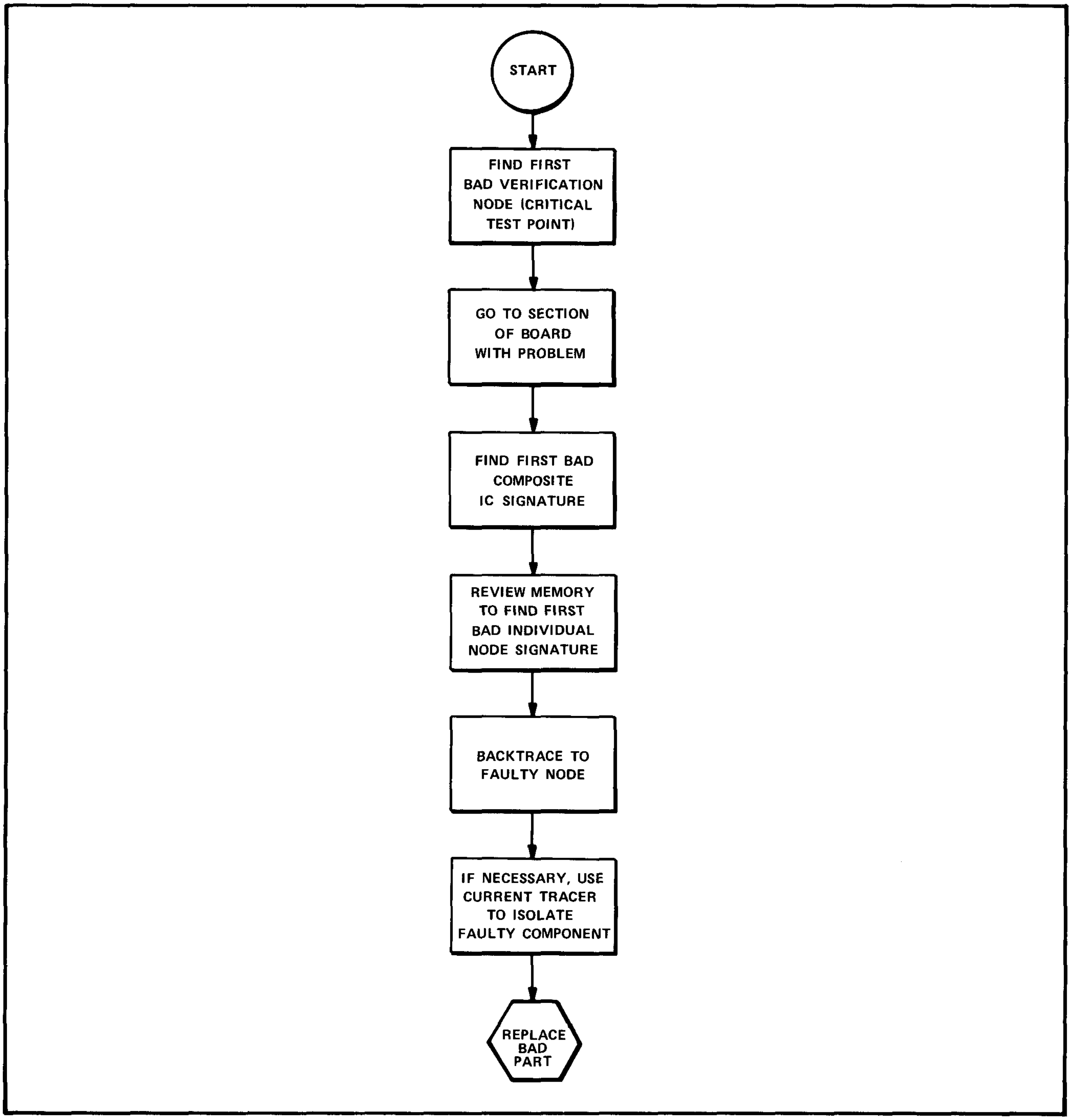

START
FIND FIRST BAD VERIFICATION NODE (CRITICAL TEST POINT)
GO TO SECTION OF BOARD WITH PROBLEM
FIND FIRST BAD COMPOSITE IC SIGNATURE
REVIEW MEMORY TO FIND FIRST BAD INDIVIDUAL NODE SIGNATURE
BACKTRACE TO FAULTY NODE
IF NECESSARY, USE CURRENT TRACER TO ISOLATE FAULTY COMPONENT
REPLACE BAD PART

Figure 2. Identifying a Fault Source in a Digital Circuit

Composite signature is a sum of all triggered signatures since the last CLEAR. A signature is triggered when the probe button is pushed. This causes the signature to be added to the composite. The triggered signature is also stored in the memory stack.

First, when troubleshooting by hand, the correct operation of a circuit board can be tested by taking a composite signature on key test points or "verification nodes". A good composite at this point means the circuit board is operating correctly with high confidence. This assumes the designer has created verification nodes that will not leave undetected faults.

If the composite does not match the signature table, there is a fault somewhere in the circuit. The divide and conquer strategy is now used. Perhaps a bad signature at an individual verification node indicates a certain part of the circuit is the source of the fault. Thus, the next step is to measure individual signatures on verification nodes. Figure 3 details the process used to verify correctly operating circuit boards or find the subsection of the board with the failure if a faulty verification node is found.

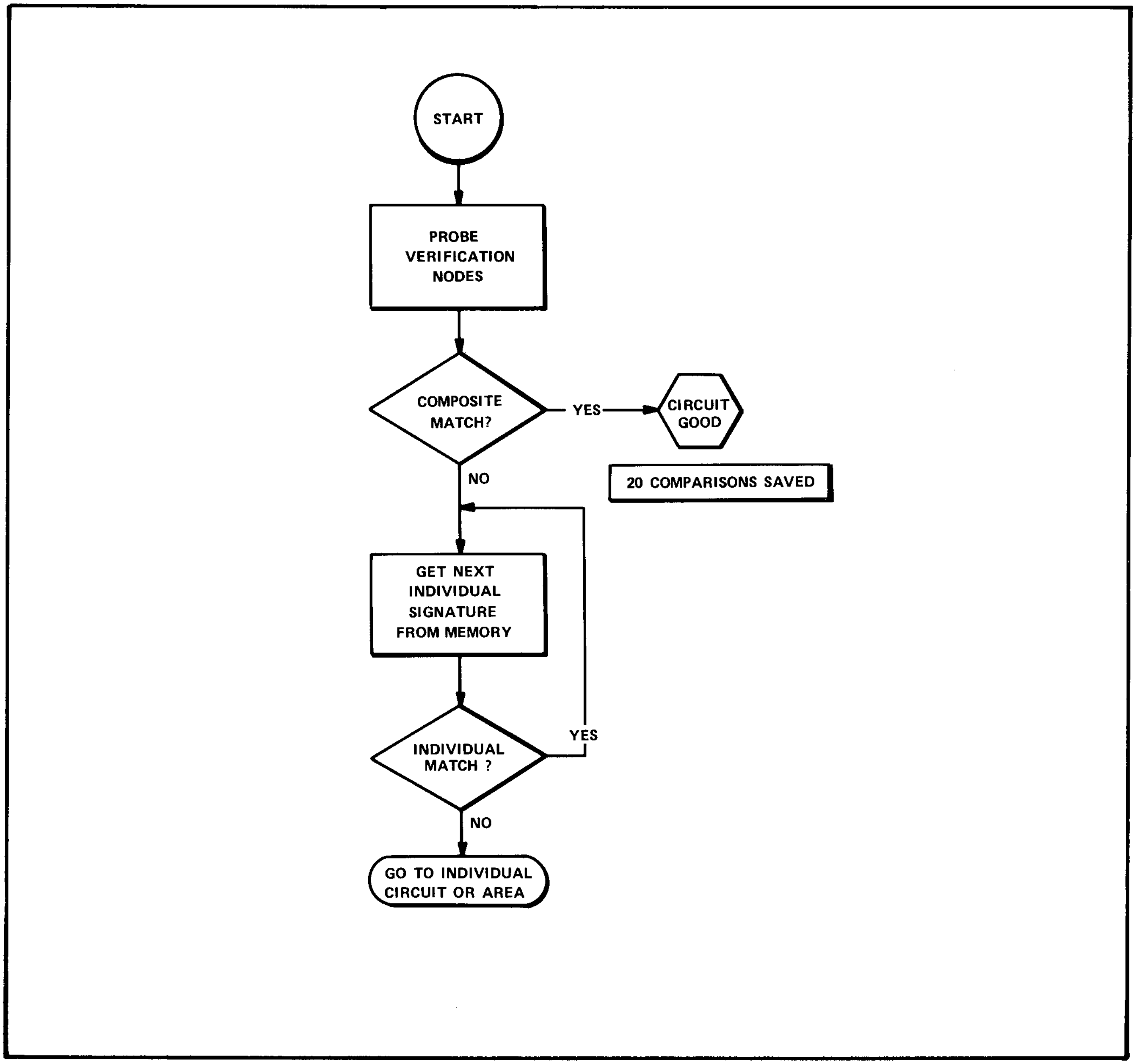


Figure 3. Board Verification — Finding the Subsection of Board with the Fault Source

Composite signature is again used in the faulty part of a circuit to do high level testing and quickly identify a bad signal with only minimal comparisons to documentation. Figure 4 shows the process of finding the first individual signature that does not match. This is necessary to begin backtracing. Composites can be taken on logical groupings of signals. For example, since it is easy to lose your place and tedious to count pins, a composite signature for a single IC saves the troubleshooter time and frustration when probing.

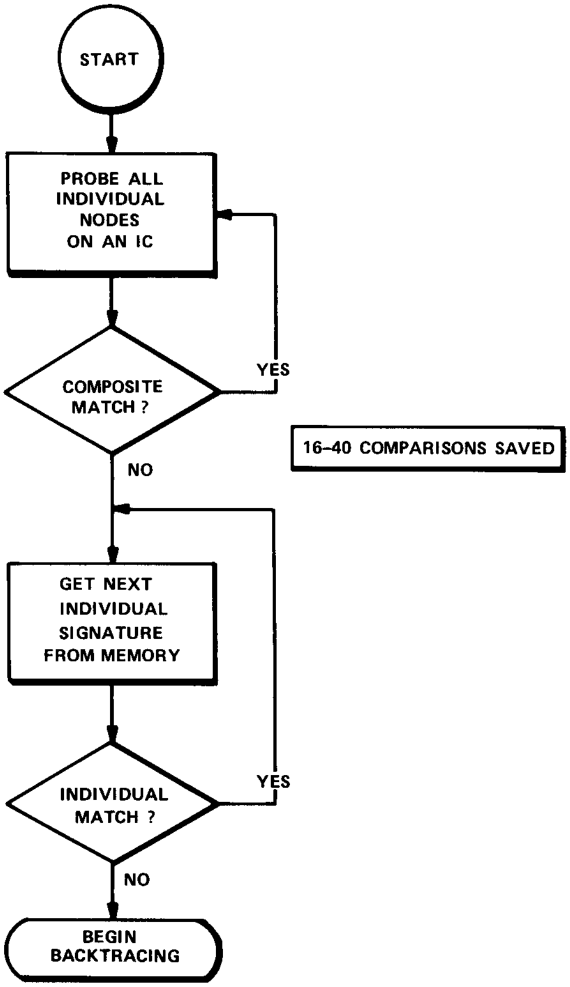


Figure 4. Finding a place to start backtracing by comparing Composite Signature for an Integrated Circuit.

Notice that as long as the composite consists of less than 32 signatures, it will not be necessary to **reprobe** the individual nodes, test points, or pins. The HP 5006 A remembers the last 32 signatures probed. It numbers them in the order probed. Number one is thus the first signature probed. When reviewing the stack in RECALL mode, the HP 5006 first displays the number and then the signature. The first signature displayed is the composite. The individual signatures are displayed in reverse order when the probe button is pushed (RECALL mode). Address and data buses can have composite signatures. Checking each line with the HP 5006A signature memory can be done quickly if there is no match of composite signature.

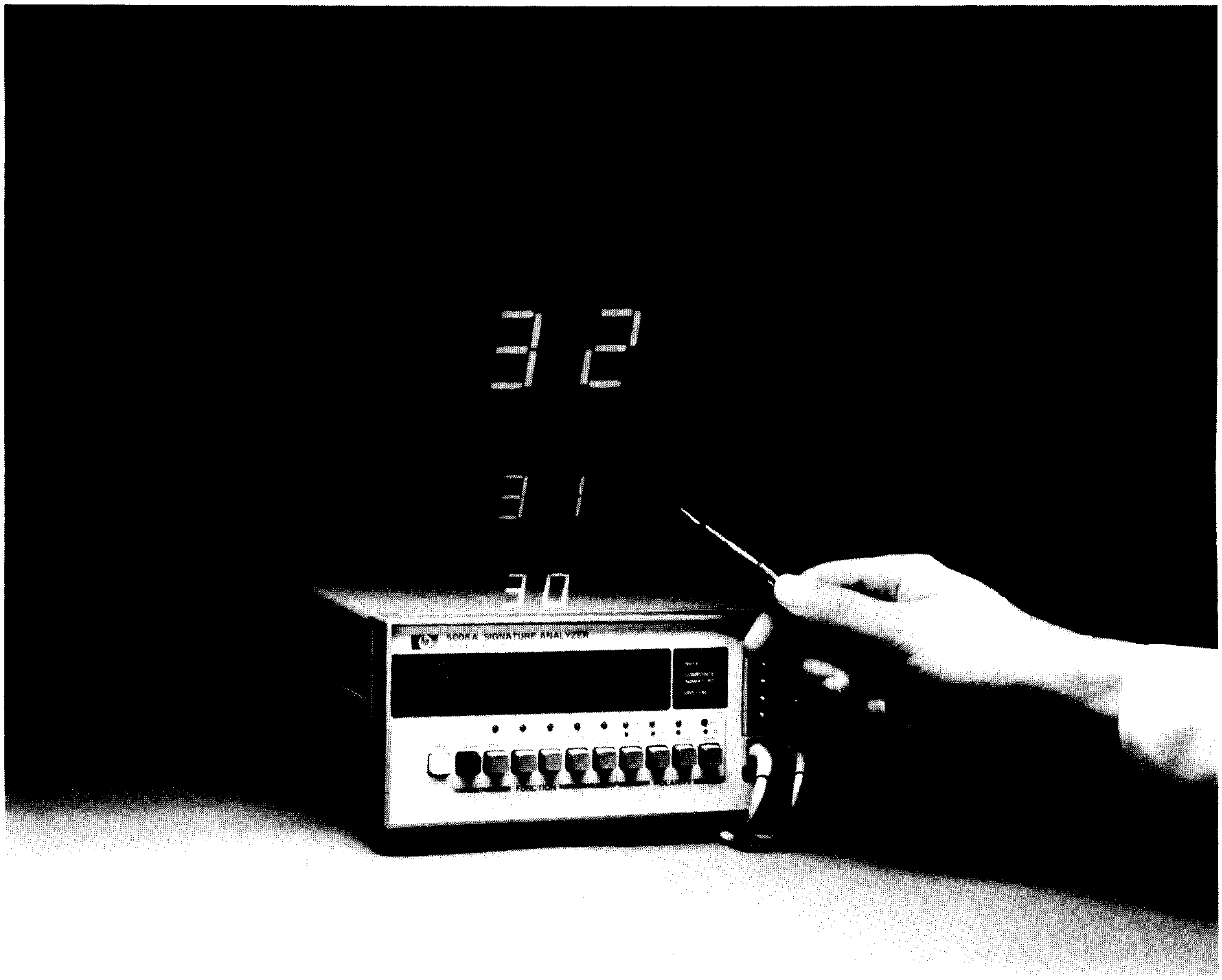


Figure 5. The operator can concentrate on probing without referring to the display using the signature memory in the HP 5006A. This makes misprobes and miscounts less likely. Time is saved when composite signature does not match because individual signatures do not need to be reprobed. The memory can be reviewed in the RECALL mode by pressing the probe switch and then compared to printed tables.

BACKTRACING

Once a bad signal is identified, the backtracing process begins. Backtracing is defined as tracing faulty signals upstream in a circuit to isolate the source of bit stream errors. Because signature analysis is highly accurate, it is well suited for detecting even the most subtle errors downstream from the source. See Figure 6 for a detailed graphical explanation of backtracing. Because a good part will pass faulty waveforms, inputs are checked first. If a faulty waveform is detected at an input, there is no reason to check an output. The troubleshooter uses the schematic and signature table to trace his way back to the source of errors, checking only inputs as he probes. The stopping rule in backtracing is:

"When all inputs on a device are **good**, *stop backtracing."*

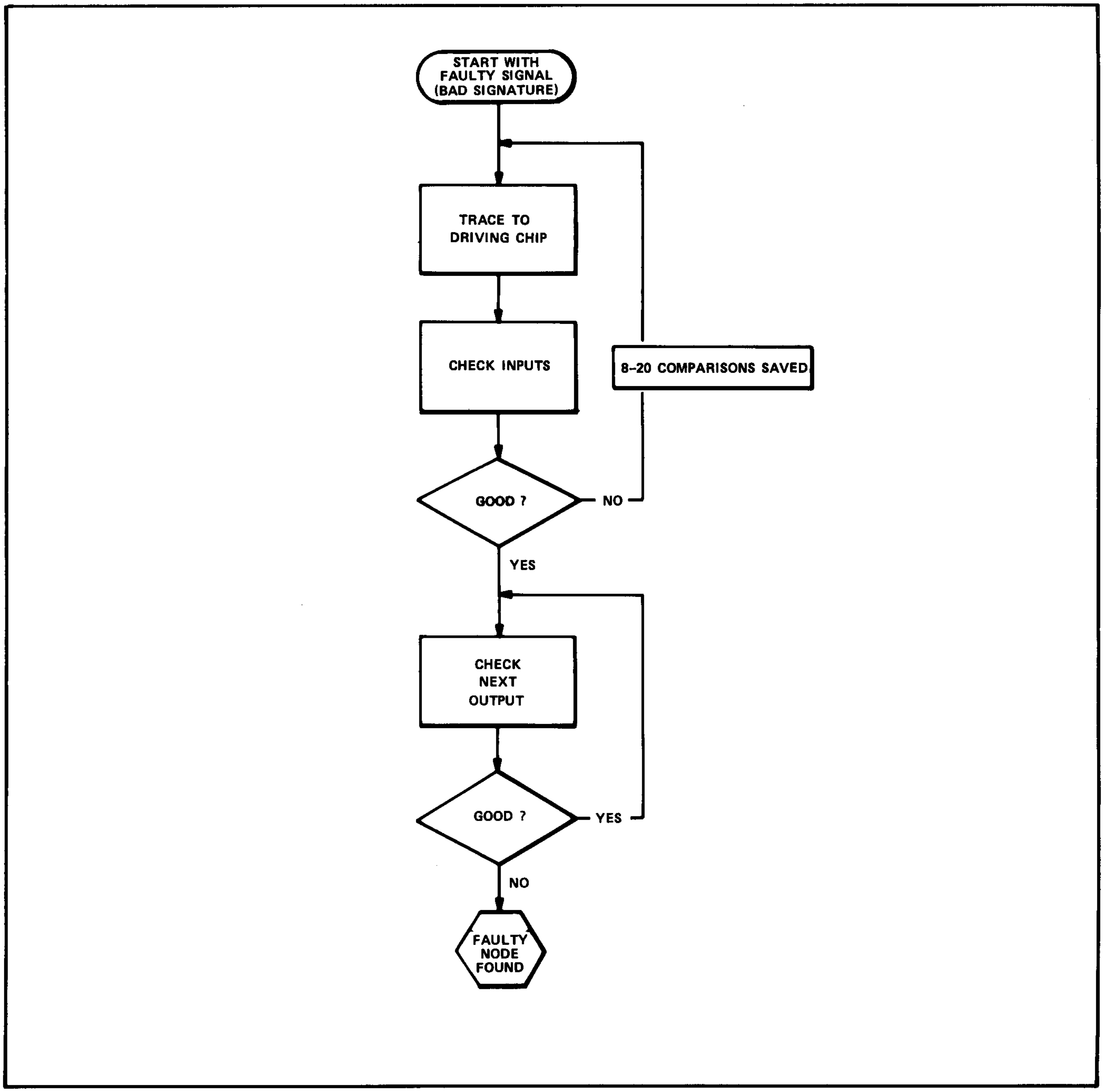


Figure 6. Backtracing

In most cases, the source of waveform errors is generated inside a faulty device. Most likely one of the output signatures will not match the recorded good ones. In all cases, a bad node has been found. Figure 7 shows the conclusions made after all inputs have been found good. If all outputs (and inputs) are good, the fault must be between this **good** chip and the last faulty input probed. Hence, we conclude the problem is a bad trace, in this case.

It is possible to speed up the backtracing process by generating composite signatures for inputs. The troubleshooter then probes all inputs on a device and only compares the composite signature to the expected one. Thus, when a good composite signature is found for inputs, the backtracing process can stop.

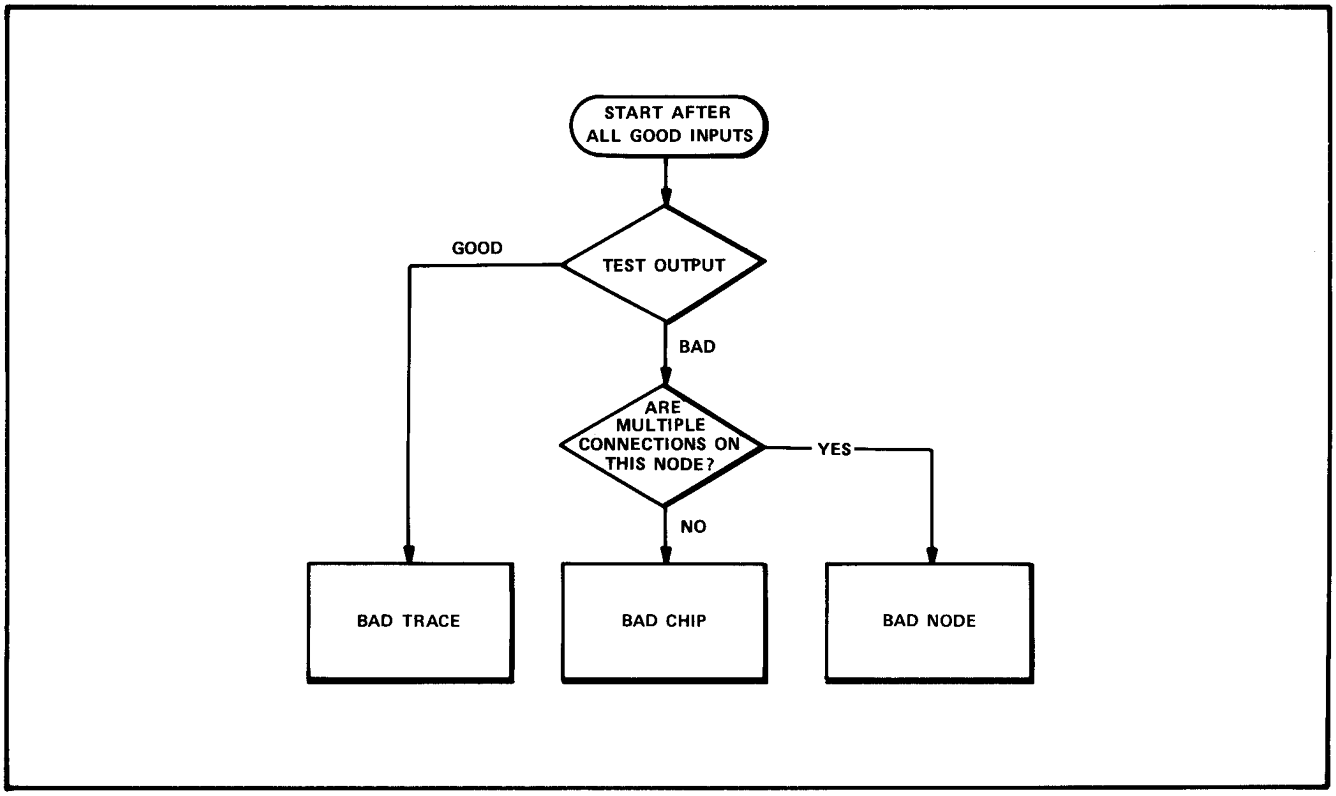

START AFTER ALL GOOD INPUTS
TEST OUTPUT
GOOD
BAD
ARE MULTIPLE CONNECTIONS ON THIS NODE?
YES
NO
BAD TRACE
BAD CHIP
BAD NODE

Figure 7. Resolving outputs after finding a part with all good inputs.

TIME SAVING ESTIMATES

It has been estimated that 5 to 35 minutes can be saved per circuit board using composite signatures when troubleshooting by hand without the aid of a computer to do matching of the individual signatures. These estimates are based on a number of assumptions:

1. The average number of nodes which need to be probed to find the bad parts.
2. The amount of time saved by not having to read the display and make a visual comparison to a signature table or schematic, for each individual signature.

Table 1 details these estimates for the three steps in troubleshooting where signature comparisons occur: probing verification nodes, finding the first bad signal and backtracing. These estimates do not take into consideration the additional advantage of not having to hold the probe on the node while reading the display and comparing to documentation. The benefit is that the probe is less likely to slip off the pin. Thus, the user will not have to recount the pins and the chances of a misprobe are much lower.

TABLE 1 – **Summary of time saved per circuit board using composite signature**

TROUBLESHOOTING STEP	NUMBER OF COMPARISONS			AVERAGE TIME SAVED PER CIRCUIT BOARD @ 5 SEC / COMPARISON
	Min.	Avg.	Max.	
Verification Nodes	5	15	25	1¼ Min.
First Bad IC (Pins x IC's)	80 (16 x 5)	100 (20 x 5)	200 (40 x 5)	8⅓ Min.
Backtracing (Inputs x IC's)	80 (8 x 10)	100 (10 x 10)	200 (20 x 10)	8⅓ Min.
Total	165	215	425	∽ 18 Min. Average 5 – 35 Min. Range

USING A COMPOSITE SIGNATURE ANALYZER

The HP 5006A automatically generates a unique "composite signature" after each triggered signature measurement. The composite signature is a computed signature, representative of the sum of all the triggered signatures accumulated since the last CLEAR or power-up. Only triggered signatures (those preceded by pressing the Data Probe Switch) can affect the composite signature. In operation, the composite signature is automatically recalculated and updated with each new measurement, based on two factors; the previous composite signature and the new signature. In this manner, the composite signature remains representative of all signatures taken, even after signatures have dropped off the bottom of the memory stack.

A composite signature can be used to represent as many signatures as you like, from a set of two pins to an entire circuit board. You should be aware that the larger the number of pins involved, the greater the chance of a duplication, misprobe or forgery. Approximately twenty measurements per composite signature is generally a safe and convenient number.

The composite signature can be reviewed at any time by pressing the front panel RECALL key. In the RECALL function, the composite signature is identified in two ways; the front panel COMPOSITE SIGNATURE LED annunciator lights and all four decimal points in the display light (the latter is advantageous during remote operation). To return to normal operation press RECALL a second time to turn the function off.

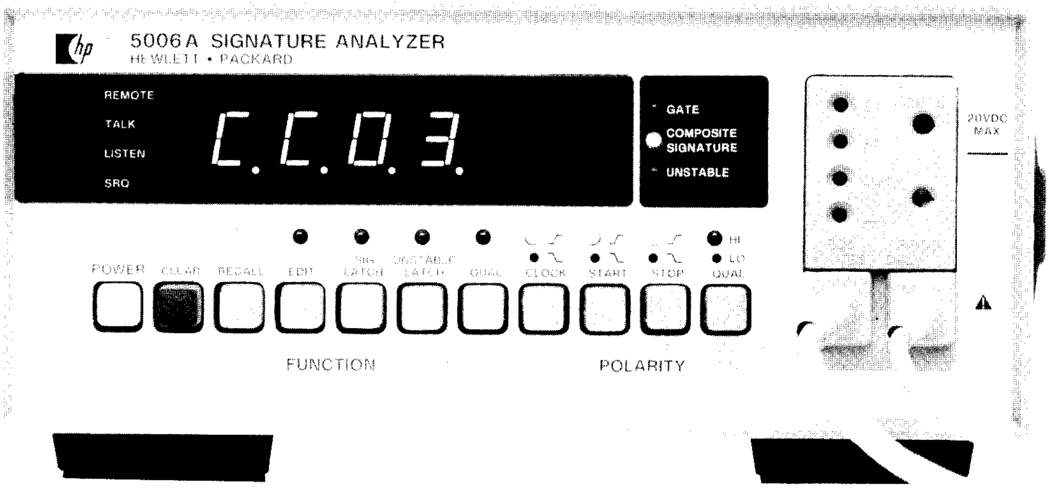


Figure 8. Troubleshooting is easier and faster with composite signature for those who do not have a computer-aided system. Composite signature is indicated in the RECALL mode with decimal points in the display and a lit LED between the gate and unstable LEDs just to the right of the display.

Editing the Signature Memory Stack

If a composite signature comes up bad for an IC or bus, it may be because the user has probed the same pin twice, for example. The user does **not** have to reprobe every pin in the group. Single signatures can be changed in the stack in the EDIT mode. Composite signature will be automatically corrected when the individual signature is remeasured. The remeasured signature will be written over the old entry in the stack. The new composite and individual signature can be reviewed by going back to the RECALL mode.

CALCULATING COMPOSITE SIGNATURE

One may wish to calculate composite signatures for existing products rather than measuring them with the HP 5006A. Calculating composite signature is straightforward. It is simply the sum of the individual signatures. However, since signatures are represented in a unique character set (see Table 2), they must first be converted to hexidecimal and then reconverted after they are summed. Carries beyond 16 bits are simply discarded.

TABLE 2 — Hex to Signature Conversion Chart

HEX	SIGNATURE
A	A
B	C
C	F
D	H
E	P
F	U

Figure 9 shows an example for the composite address signature of the 8085 microprocessor when free-running. A microprocessor can be made to freerun by forcing a NOP on its data bus with hardware. The processor will cycle through its address space continuously searching for an instruction only to find another NOP. Freerun is useful for checking out the microprocessor, ROM's, address bus and data bus.

ADDRESS	SIGNATURE		HEX
0	UUUU		FFFF
1	5555		5555
2	CCCC		BBBB
3	7F7F		7C7C
4	5H21		5D21
5	0AFA		0ACA
6	UPFH		FECD
7	52F8		52C8
8	HC89		DB89
9	2J70		2D70
10	HPPO		DEE0
11	1293		1293
12	HAP7		DAE7
13	3C96		3B96
14	3827	+	3827
15	755P		755E

Total 0579

Composite Address Signature = 0579 (8085 FREERUN)

Figure 9. Example — Free Run Address Composite Signature for the 8085

CONCLUSION

Composite signature is the sum of individual signatures. The individual signatures can be from IC's, buses or any other logical grouping of signals. Composite signature saves troubleshooters time when making comparisons of the display to printed documentation. Preliminary estimates show from 5 to 35 minutes can be saved per circuit board tested.

With the HP 5006A, signature gathering and comparing to documentation can be done separately and in groups. The signature memory has allowed probing to be done without having to look at the display. The benefit is that the operator can concentrate on probing, making miscounts and misprobes less likely. Individual signatures do not have to be RECALLED if the measured composite matches the documentation, saving time.

APPENDICES — THEORY OF OPERATION

The two appendices show why composite signature works. The linearity property of signature analysis allows us to add signatures to form a composite signature. The section on "probability of error detection" shows that the chance a composite signature will **not** detect an altered waveform (forgery) is only slightly less than that for individual signatures (for composites made up of moderate amounts of individual signatures).

APPENDIX A

LINEARITY

Signature Analysis is linear. Because of this property, composite Signature Analysis is possible. Linearity means signatures can be summed. Composite signature should possess the normal properties of addition such as commutativity and associativity

The idea was generated from a ROM testing technique. A ROM can be tested by reading each address in a loop and then taking signatures on the data bus. Another technique serializes the individual data bus lines and generates a single signature for all the data. Thus, one signature represents whether the ROM was working or not. There is then no need to test the individual signatures of each bus line.

Figure 10 shows the probability of not detecting a waveform bit error as a function of waveform bit sequence length. This phenomenon has been called a forgery. The signature appears good although the waveform is different. The probability of forgery approaches 2^{-16} asymptotically as bit sequences get long. It is effectively 2^{-16} for bit sequences longer than 25.

With microprocessor clocks typically 1 MHz or greater, it is easy to get waveforms with thousands of bits. Intuitively the probability of forgery (not detecting a waveform error) is less for individual signatures than composite signatures. It only becomes significant when the number of individual signatures making up the composite becomes large. Composite signatures can be used to structure digital testing and provide a "top-down" approach to troubleshooters.

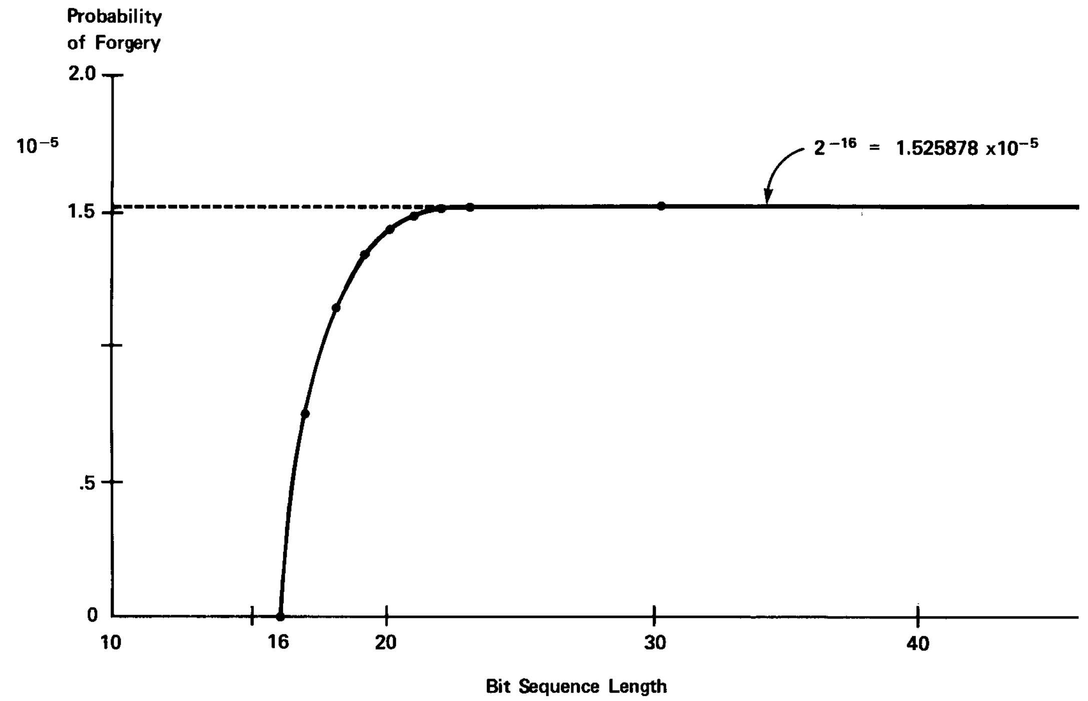


Figure 10. Probability of forgery as a function of bit sequence length.

APPENDIX B

PROBABILITY OF ERROR DETECTION

There are two ways a composite signature will not catch a bit error. A forgery of one or more of its individual signatures is one way. A signature is forged when a bit error occurs, but the signature remains unchanged. A sum of forgeries is a forgery. The second way is when two or more bad signatures cancel in the sum. The probabilities are addressed by the following questions:

1. What is the chance all individual signatures are good and the composite is bad? ANS: 0. i.e., A bad composite always indicates a bad individual.
2. What is the chance one individual is bad but composite is good? ANS: 0. Because of linearity, the one altered individual will produce an altered sum.
3. What is the chance two or more individual signatures are bad and cancel out when summed in the composite? ANS: 2^{-16}.
4. What is the chance one or more individual signatures are forged, and thus, the composite is forged?

   ANS: $1\text{-}(1\text{-}p)^n$

   where n = number of individual signatures in the sum.

   p = probability of individual forgery ($\sim 2^{-16}$ for 25 bits or more).

Proof of 3:

Suppose individual bit errors occur to alter 2 individual signatures.

$A \text{—>} A' = A + a$ a not = 0
$B \text{—>} B' = B + b$ b not = 0

If the composite is to remain unaltered then:

$A + B = A' + B'$

Substituting gives

$A + B = A + a + B + b$
$0 = a + b$ for unaltered composite

Thus, for every "a" there is only one "b" out of 2^{16} possibilities that give a sum equal to zero.

For three or more signatures, the argument would be similar.

Please see Figure 11 for a graphical explanation.

Stated simply: A sum of signatures will be in the set of all signatures. It will be equally likely that a sum will be zero as any other signature. Thus, the chance that two or more signatures are bad and cancel in the composite is $1/2^{16}$.

$$\begin{array}{ccccc} \underline{\text{SIGNATURE}} & & \underline{\text{HEX}} & & \underline{\text{CHANGE}} \\ S_1 & \longrightarrow & H_1 & + & \Delta_1 \\ S_2 & \longrightarrow & H_2 & + & \Delta_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ S_n & \longrightarrow & H_n & + & \Delta_n \\ \hline S_c & \longleftarrow & \sum\limits_{i=1}^{n} H_i & + & \sum\limits_{i=1}^{n} \Delta_i \end{array}$$

$$P\left(\sum_{i=1}^{n} \Delta_i = 0\right) = \frac{1}{2^{16}}$$

Figure 11. Probability of two or more individual signatures cancelling in the sum (composite signature).

Derivation of 4:

Because it is a sum, the probability that a composite signature has not been forged is the probability that no individual signature has been forged.

Let n = number of individual signatures.
f_c = composite forgery
f_i = individual forgery

$$\text{not } f_c = \mathop{\Omega}_{i=1}^{n} \text{not } f_i$$

Assuming forgeries occur independently:

$$p(\text{not } f_c) = \prod_{i=1}^{n} \left\{1 - p(f_i)\right\}$$

$$\approx (1 - 2^{-16})^n \text{ for bits sequences longer than 25 bits.}$$

Figure 12 graphs the probability of composite forgery as a function of individual nodes (n).

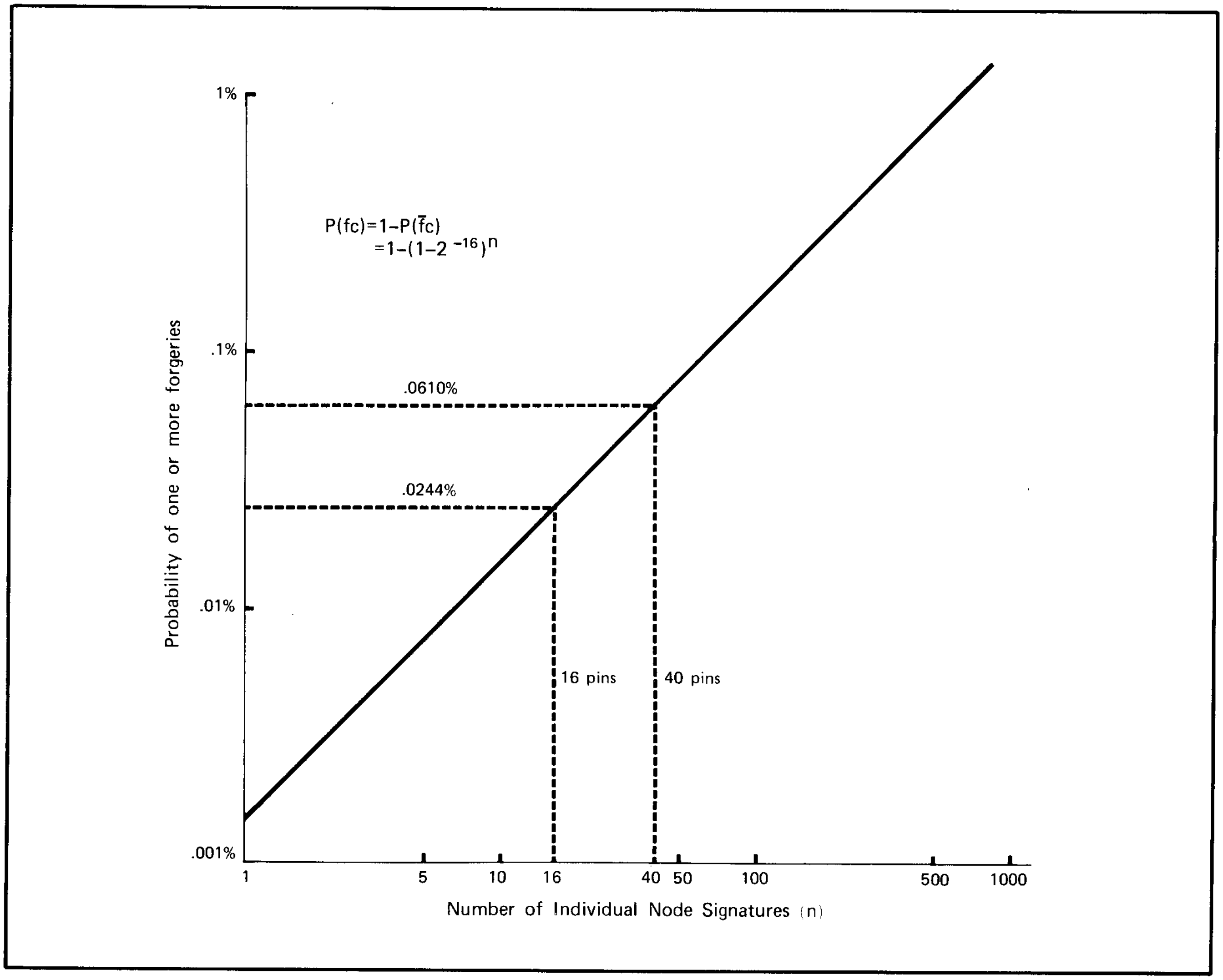


Figure 12. Probability of composite forgery as a result of one or more individual forgeries.

For more information, call your local HP Sales Office or nearest Regional Office: **Eastern** (201) 265-5000; **Midwestern** (312) 255-9800; **Southern** (404) 955-1500; **Western** (213) 970-7500; **Canadian** (416) 678-9430. Ask the operator for instrument sales. Or write Hewlett-Packard, 1501 Page Mill Road, Palo Alto, CA 94304. **In Europe**: Hewlett-Packard S.A., 7, rue du Bois-du-Lan, P.O. Box, CH 1217 Meyrin 2, Geneva, Switzerland. **In Japan**: Yokogawa-Hewlett-Packard Ltd., 29-21, Takaido-Higashi 3-chome, Suginami-ku, Tokyo 168.

02-5952-7684 **APRIL 1983** **PRINTED IN U.S.A.**