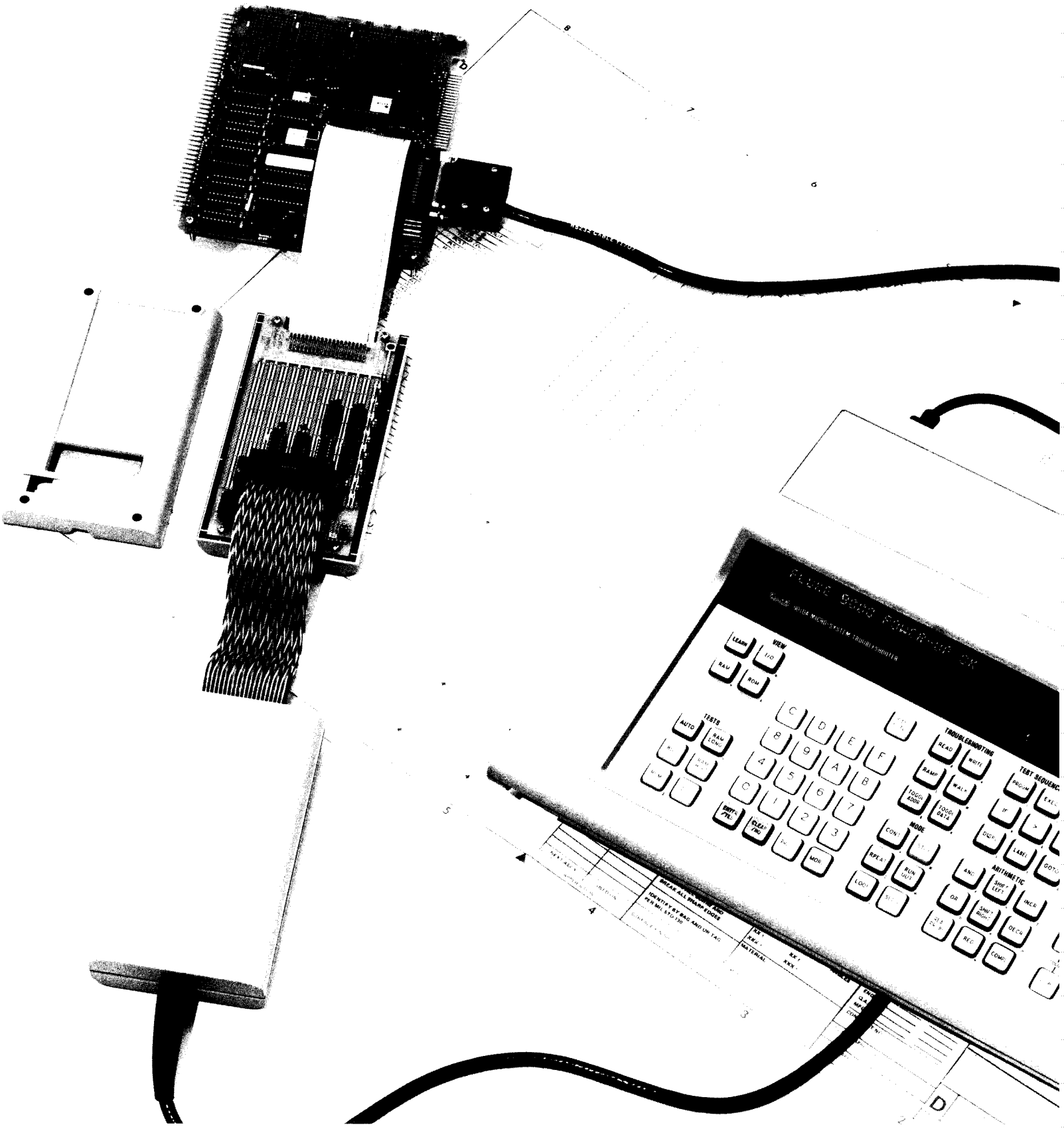
FLUKE

Technical Data

Application Information B0156

9000-Series User Designed Interface Pod Adapters



Introduction

Many microprocessor systems not directly supported by 9000-Series interface pods can still be tested with 9000-Series products. If you have such a microprocessor system, this technical note will help you understand how 9000-Series products can be adapted for use with your system.

The 9000-Series interface pods are usually converted for use with "podless" micro-systems by an "interface pod adapter." A pod adapter usually consists of a simple piece of digital circuitry that connects between the Pod UUT* cable and the UUT microprocessor socket. A block diagram of a 9000-Series system that employs a pod adapter is shown in Figure 1.

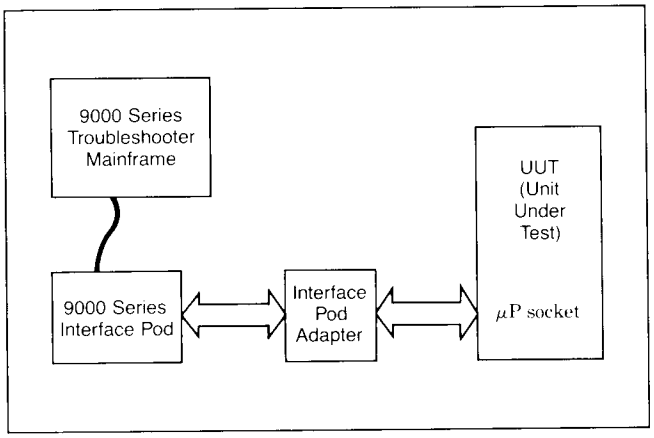


Figure 1. A 9000-Series system that employs an "interface pod adapter."

This technical note is intended to help you decide if a pod adapter will be a viable solution to your testing and troubleshooting needs. The note consists of two parts. The first part takes the form of a question and answer session on pod adapters. The second part presents two pod adapter examples.

Understanding Interface Pod Adapters

Why are pod adapters feasible?

The unique nature of the pod's relationship with the UUT makes it possible to adapt the functional and timing characteristics of the pod to those of the UUT microprocessor. Here's how it works: The pod contains a microprocessor of the type that the pod emulates. When the pod communicates with the UUT, the pod executes individual read or write cycles (not full instruction cycles) on the UUT bus. These read or write cycles have the same timing characteristics as those of the pod microprocessor. This means you can often use digital logic to adapt the read and write cycle characteristics of the pod to those of the UUT.

NOTE

The UUT read or write cycle (performed by the pod) is performed upon command from the troubleshooter† —not from the UUT system software. Therefore, UUT system software does not play a role in determining the characteristics of the pod's accesses to the UUT. Even the troubleshooter and pod software characteristics do not change the basic nature of the UUT read or write cycle that is performed by the pod.

The other factor that makes pod adapters feasible is the fact that different microprocessor types often have very similar functional and timing characteristics. Also, a given microprocessor type may be a member of a "family" of microprocessors. These family microprocessors often have identical timing characteristics. Some even have compatible instruction sets. (Refer to the 6500 Microprocessor Family example.)

What does a pod adapter consist of?

Pod adapters can consist of something as simple as a DIP plug and a DIP socket connected together in a way that adapts the pin configuration of a particular pod microprocessor to that of the UUT microprocessor. (Refer to the 6500 Microprocessor Family example.) More complicated adapters will typically consist of digital logic such as gates, flip-flops, or buffers.

**Unit Under Test*

† In this note, as in other 9000 Series literature, the word troubleshooter refers to the 9000 Series mainframe (as shown in Figure 1).

Copyright © 1983 John Fluke Mfg. Co., Inc. All rights reserved.

FLUKE®

What are the limitations of pod adapters?

Although a pod adapter can be a very effective device for interfacing 9000-Series products to a particular micro-system, it may not be able to do *everything*. Here are some possible limitations:

9000-Series Functions. Differences in functional and timing characteristics of the pod and the UUT microprocessor can present some limitations in bringing the full capabilities of the 9000-Series products to your microprocessor system. Some functions of the 9000-Series cannot always be used with a particular adapter, and some functions or capabilities of the UUT microprocessor cannot always be implemented.

Timing Aspects. An adapter may not be able to perfectly adapt the pod's timing characteristics to those of the UUT. However, exact timing characteristics are not always necessary for testing purposes. If the adapter provides signals that occur within the timing constraints of the pod and the UUT circuitry, testing with 9000-Series products will generally be possible.

Drivability Checking. The pod checks the drivability of every pod output line during each UUT bus cycle. Adapter circuitry can adversely affect this drivability check. For example, if a certain output line is adapted to a UUT through an inverter, a drivability error on the UUT side of the inverter would not be detected by the pod.

Universality. It is typically a lot easier to design and construct a pod adapter to perform with only one type of UUT. Designing an adapter that will work on more than one type of UUT (for a given type of μP) is definitely possible, but it can complicate the development of the adapter by making the design requirements more stringent. The amount of "universality" you will want to design into a pod adapter is also sometimes limited by the differences in the functional and timing characteristics of the pod and the UUT microprocessor, as mentioned above.

Which 9000-Series functions can a pod adapter utilize?

The functions that can be used with a pod adapter depend upon which signal lines of the pod are adaptable to the UUT. It can also depend upon how well the timing characteristics of the pod and the UUT microprocessor are matched. The following paragraphs should help you determine the features that a pod adapter should have if a particular function is desired.

Reads and Writes. The most important capability that a pod adapter can have is the ability to execute read and write cycles on the UUT. If it is possible to execute individual read and write cycles on the UUT, you should be able to use any of the following functions of the 9000-Series:

LEARN
AUTO TEST
BUS TEST
ROM TEST
RAM LONG TEST*
RAM SHORT TEST*
I/O TEST
READ @ (address)
WRITE @ (address) = (data)
RAMP
WALK
TOGGLE DATA
TOGGLE ADDRESS

All of these functions are nothing more than organized sets of read and write cycles—as seen from the UUT. The key point: You can often utilize the 9000-Series functions listed above simply by adapting the pod and UUT signals necessary for performing read and write cycles. These signals are often nothing more than:

—clock signal(s)

—address and data bus signals

—read/write control signal(s)

These are often the easiest signals to adapt from the pod to the UUT.

Exercising μP Input and Output Lines (i.e. lines other than address and data lines.) In 9000-Series literature, μP input lines (such as interrupts) to the pod from the UUT are considered to be status lines. Output lines (such as strobe lines) from the pod to the UUT are considered to be control lines. All 9000-Series pods allow input lines to be individually checked with the READ STS function of the troubleshooter. Most output lines can be individually activated with the WRITE CTL function.

**During RAM tests, dynamic RAM will require the proper refresh control signals. Refer to the appropriate Pod Manual for more information.*

Pod adapters can usually be designed to utilize the pod's status (input) and control (output) lines. The READ STS and the WRITE CTL functions of the troubleshooter can then be used. Adaptation of the appropriate pod status (input) lines will allow the troubleshooter to automatically report active interrupts and other active input lines.

NOTE

During UUT accesses, some pod output (control) lines such as read or write control strobes are automatically activated. These lines are activated with the proper timing characteristics, which allows the UUT read or write cycle to occur in a normal fashion. Generally, these lines cannot be activated by the WRITE CTL function, but they should be considered when a pod adapter is being designed.

The Probe Functions. The troubleshooter probe can be synchronized to gather information during valid address or data periods of the pod. The pod generates (and sends to the troubleshooter) a sync pulse that latches probe data at the proper address or data valid time.

The probe functions can usually be implemented on adapted UUT's. The address and data synchronization modes of the probe can be used if the pod's address and data sync pulses occur during the valid address and data periods on the UUT. These sync pulses are illustrated with the UUT access timing diagrams in each pod manual.

The RUN UUT Mode. In the RUN UUT mode, the pod executes machine code that resides in the UUT itself. Generally, the RUN UUT mode can be implemented with adapted pods only if the instruction set of the UUT microprocessor is upward compatible with the instruction set of the pod microprocessor.

Complementary Functions. All of the complementary functions of the 9000-Series that do not involve interaction with the UUT (such as the VIEW, AUX I/F, TAPE, TEST SEQUENCING, and ARITHMETIC functions) can be used with any pod or any adapter.

What do I need to know to design a pod adapter?

To develop an adapter, you'll only need to deal with the read and write cycle timing (and functional) aspects of the pod and the UUT microprocessor. You don't need to know the instruction sets of either the pod microprocessor or the UUT microprocessor.

How do I decide which Interface Pod to use?

The available 9000-Series interface pods support a wide variety of microprocessor types. You will want to select a pod that most closely matches the bus structure, timing characteristics, and signal types of the UUT microprocessor—in order to utilize the functions of the 9000-Series that are most important to you.

You can use the pod selection procedure outlined below to determine if a particular pod is well suited for adaptation to your UUT. The Pod Characteristics Chart on Page 5 can help you through the first three steps of the procedure. The chart should help you narrow the field to two or three pods. Then you can use the appropriate pod manuals and the appropriate μP data manuals to help you through steps 4 and 5. The back page of this note provides ordering information for all available pod manuals.

Pod Selection Procedure

1. *Investigate the family.* Find out if the UUT microprocessor is part of a family of microprocessors. If there is a pod that represents a member of that family, proceed through the remaining steps with that pod in the front of your mind.

2. *Investigate clock compatibility.* Determine which pods are compatible with the clock requirements of the UUT. For example, if the UUT provides a single-phase clock signal, pods that require a two-phase clock signal will be difficult to implement. Pay careful attention to maximum and minimum clock rates for both the pod and the UUT. Remember that the pod operates from UUT-generated clock signals, as does the UUT microprocessor.

3. *Compare bus structures.* Determine which pods have a similar number of address and data lines as the UUT microprocessor. If the address and data lines of the UUT microprocessor are multiplexed, determine which pods share this characteristic. The Pod Characteristics Chart can help you with this step.

4. *Compare bus cycle timing characteristics.* Determine the difficulty of adapting the bus cycle timing characteristics of a pod to those of the UUT microprocessor. You can do this by "lining up" the bus cycle timing diagrams for the pod and the UUT's microprocessor. (Timing information for each pod is presented in the corresponding pod manual.) You may want

to consider combining lines (either combinationally or sequentially) in order to match the timing characteristics of a particular line. Make certain that the functional and timing characteristics of any read/write control lines can be adapted.

5. *Compare input and output lines.* Determine which pods have the "resources" to provide the necessary input and output lines for proper testing and troubleshooting of the UUT. In other words, try to determine if the functional and timing characteristics of the UUT microprocessor's input and output lines can somehow be matched to those of a pod. You may have to make tradeoffs in this area in order to find a pod that can perform read and write operations on your UUT.

You may have to perform the above steps for more than one pod before you find one that meets your testing and troubleshooting requirements. You may even want to go through a few design steps before you make the final determination.

POD CHARACTERISTICS

	CLOCK CHARACTERISTICS					BUS STRUCTURE			ADDRESS SPACE CHARACTERISTICS			
POD	Direct Crystal Connect?	# of input clock phases	# of output clock phases	Input clock freq.	Output clock freq.	Non-multiplexed data lines	Non-multiplexed address lines	Multiplexed address and data lines	Memory address space (8-bit bytes)	Memory address space (16-bit words)	I/O address space (8-bit bytes)	I/O address space (16-bit words)
Z80	NO	1	—	.5MHz-4MHz	—	D0-D7	A0-A15	—	64K	—	64K	—
1802[5]	YES	1	—	.03MHz-3.2MHz	—	BUS0-BUS7	MA0-MA7[4]	—	64K	—	7	—
6502	NO	1	2	.04MHz-2MHz	.04MHz-2MHz	DB0-DB7	AB0-AB15	—	64K	—	—	—
6800	NO	2	—	0.1MHz-2MHz	—	D0-D7	A0-A15	—	64K	—	—	—
6802[6]	YES	1	1	.4MHz-8MHz	.1MHz-2MHz	D0-D7	A0-A15	—	64K	—	—	—
6809/ 6809E	YES/ NO	1/2	2/—	.4MHz-8MHz .1MHz-2MHz	.1MHz-2MHz	D0-D7	A0-A15	—	64K	—	—	—
8048/[7] 8041	YES	1	1	600Hz-11MHz	200Hz-3.67MHz	—	P20-P23[9]	DB0-DB7	4K 320-512[2]	—	7[10]	—
8080	NO	2	—	3MHz (Max.)	—	D0-D7	A0-A15	—	64K	—	256	—
8085	YES	1	1	1MHz-10MHz	0.5MHz-5MHz	—	A8-A15	AD0-AD7	64K	—	256	—
8086	NO	1	—	2MHz-10MHz	—	—	—	AD0-AD15 A16-A19/S3-S4	4 Meg	2 Meg	64K	32K
8088	NO	1	—	2MHz-8MHz	—	—	A8-A15	AD0-AD7 A16-A19/S3-S4	4 Meg	—	64K	—
9900	NO	4	—	3MHz (Max.)	—	D0-D15	A1-A15	—	64K	32K	4K[10]	—
68000	NO	1	1	2MHz-10MHz	.2MHz-1MHz	D0-D15	A0-A23	—	16 Meg	8 Meg	—	—
Fill in your UUT μP characteristics in this row												

1. Program Memory (Read Only).
2. Data Memory.
3. Software Configurable.
4. The 8 upper order address lines are multiplexed with the 8 lower order address lines.
5. Also supports 1804/1805/1806.
6. Also supports 6802NS, 6808.
7. Also supports 8035/8039/8040/8041/8041A/8042/8049/8050/8741/8741A/8742/8748/8749.
8. Serial Bits.
9. Multiplexed with P2 I/O signals.
10. Three 8-bit ports; Four 4-bit ports.

FLUKE®

Examples of Pod Adapters

The 6500 Microprocessor Family

This example will illustrate how easy it can be to implement a pod adapter.

The 6502, 6503, 6504, 6505, 6506, and 6507 microprocessors have identical bus cycle timing characteristics; they also feature a common instruction set. These microprocessors simply have different "signal sets" and different pin-out configurations. This makes it very easy to adapt the 9000A-6502 pod to UUT's based on the 6503, 6504, 6505, 6506, or 6507.

The signal set of each of these microprocessors is a subset of the 6502 signal set. Therefore adaptation simply requires connecting a subset of the pins on the 6502 pod UUT cable plug to another DIP plug. The DIP plug can then be plugged into the UUT microprocessor socket, and testing can begin. Of course, the pinout configuration on the DIP plug must match the pinout configuration of the UUT microprocessor.

With this simple type of adapter, UUT's based on all of the above-mentioned microprocessors can be tested virtually as if the 6502 pod was designed for those systems. Since the signal sets of these microprocessors are simply subsets of the 6502 signal set, all of the signals of the UUT microprocessor can be emulated by the 6502 pod. All of the functions of the 9000-Series can be utilized, including RUN UUT.

The pinout configuration chart shown below illustrates the pin connections that must be made for adapting the 6502 pod to 6503, 6504, 6505, 6506, or 6507-based UUT's.

SIGNAL NAME	6502 POD PIN NO.	6503 (UUT) PIN NO.	6504 (UUT) PIN NO.	6505 (UUT) PIN NO.	6506 (UUT) PIN NO.	6507 (UUT) PIN NO.
Vss	1	2	2	2	2	2
RDY	2	5	NP	3	NP	3
CLK1 (out)	3	NP	NP	NP	3	NP
$\overline{IRQ}$	4	3	3	4	4	4
*	5	NP	NP	NP	NP	NP
$\overline{NMI}$	6	4	4	5	5	4
SYNC	7	NP	NP	NP	NP	NP
Vcc	8	5	4	5	5	4
AB0	9	6	5	6	6	5
AB1	10	7	6	7	7	6
AB2	11	8	7	8	8	7
AB3	12	9	8	9	9	8
AB4	13	10	9	10	10	9
AB5	14	11	10	11	11	10
AB6	15	12	11	12	12	11
AB7	16	13	12	13	13	12
AB8	17	14	13	14	14	13
AB9	18	15	14	15	15	14
AB10	19	16	15	16	16	15
AB11	20	17	16	17	17	16
Vss	21	2	2	2	2	2
AB12	22	NP	17	NP	NP	17
AB13	23	NP	NP	NP	NP	NP
AB14	24	NP	NP	NP	NP	NP
AB15	25	NP	NP	NP	NP	NP
DB7	26	18	18	18	18	18
DB6	27	19	19	19	19	19
DB5	28	20	20	20	20	20
DB4	29	21	21	21	21	21
DB3	30	22	22	22	22	22
DB2	31	23	23	23	23	23
DB1	32	24	24	24	24	24
DB0	33	25	25	25	25	25
R/$\overline{W}$	34	26	26	26	26	26
*	35	NP	NP	NP	NP	NP
*	36	NP	NP	NP	NP	NP
CLKO(in)	37	27	27	27	27	27
S. O.	38	NP	NP	NP	NP	NP
CLK2(out)	39	28	28	28	28	28
$\overline{RES}$	40	1	1	1	1	1

**No 6502 signal exists for the corresponding 6502 pin.*

NP means that the UUT μP socket has no pin for connection to the corresponding 6502 signal; the corresponding 6502 pin is therefore not connected.

NOTE

Members of the 6500 family other than the 6502 have fewer address lines (and thus a smaller address space) than the 6502. For example, the 6502 has 16 address lines (AB0-AB15) and an address range of 0-FFFF; the 6503 has only 12 address lines (AB0-AB11) and an address range of 0-FFF. Since the 6502 pod will allow addressing all 65K of its address space (0-FFFF), performing a troubleshooter operation (such as a Read or a Write) at an address beyond the valid range of the 6503 microprocessor will *not* result in the display of an "ILLEGAL ADDRESS" error message.

The NSC800 Microprocessor

Introduction

Adapting to an NSC800-based UUT is slightly more involved than adapting to a 6500 family system. This adapter example should illustrate some additional concepts regarding pod adapters.

Choosing a Pod for the Adapter

You can use the Pod Characteristics Table (discussed earlier) to help you select the pod for this adapter. When you compare the characteristics of the NSC800 with those of each pod in the table, you'll quickly discover that the 8085 pod appears to be very similar to the NSC800.

If you consult the NSC800 Handbook, you'll find that the NSC800 indeed features a signal set and a bus structure that are almost identical to those of the 8085 microprocessor. The read/write cycle timing characteristics of the NSC800 are also very similar to those of the 8085. These similarities in bus structure and cycle timing make the 9000A-8085 pod the logical choice for adapting to an NSC800-based UUT.

The Adapter Circuitry

The circuitry required for an adapter will depend on the requirements of the UUT involved. For example, if your NSC800 UUT does not use a particular μP line, any circuitry required to adapt that line to the 8085 pod will not be necessary. As you read through the example, you should refer to the timing diagram of Figure 2. Refer also to the schematic diagrams of Figures 3, 4, and 5.

NOTE

The NSC800 is a CMOS microprocessor, which allows it to be operated at power supply voltages other than +5 volts. The 8085 pod (and the troubleshooter probe) operate at TTL signal levels. If the NSC800 UUT operates at a power supply voltage other than +5 volts, level shifting circuitry will have to be included for every line (except ground) connected between the UUT and the 8085 pod. This circuitry is *not* shown in the schematic diagrams of this example. If buffer circuitry is necessary, the ability of the pod to check for drivability errors can be affected.

The Clock Circuitry. The clock requirements for both the NSC800 and the 8085 pod are basically the same; both processors can be driven directly by a crystal or by a UUT-generated clock signal. The minimum and maximum operating frequencies of the NSC800 fall within those of the 8085 pod, so no clock division or multiplication circuitry is required.

Both processors generate a single-phase clock output signal. A quick comparison of the clock signals (shown in Figure 2) for the pod and the NSC800 shows that the 8085 pod CLK output signal must be inverted for use on the UUT. This will allow the 8085 pod bus signals to be synchronized with those on the NSC800 UUT. If the CLK signal is not used on the UUT, no inverter is necessary.

NOTE

If a crystal is used on your UUT for generating clock signals, the length of the 8085 pod cable lines (and the adapter cable lines) may make it necessary for you to include a clock oscillator in the adapter circuitry. For more information regarding the generation of 8085 Pod clock signals, refer to the Fluke technical note titled "Guide to 8085 Microprocessor-Based System Testing with a 9000-Series Micro-System Troubleshooter." Contact your local Fluke office for a copy of the note. Ask for technical note B0151.

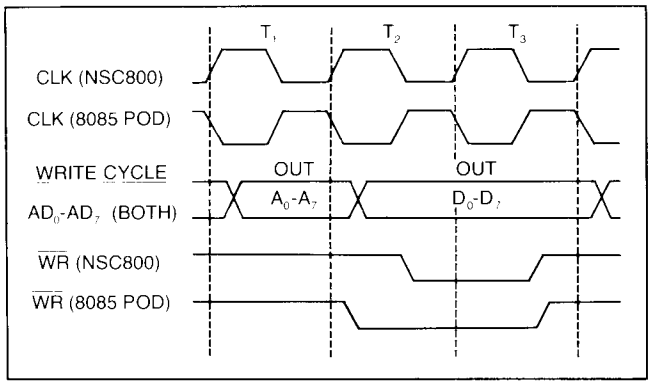


Figure 2. Timing differences between NSC800 μP and 8085 pod signals.

The Address and Data Buses. The address and data bus structures of the 8085 pod and the NSC800 are basically identical. Both multiplex the lower order address bits with the data bits. The timing of the multiplexing (with relation to the clock signal) is the same. The ALE (address latch enable) signals are also identical. Therefore, no adapter circuitry is required for the address/data busses or the ALE signal.

Read (RD) and Write ($\overline{WR}$) Control Signals. The $\overline{RD}$ control signal of the NSC800 is nearly identical to that of the 8085 pod. However, Figure 2 shows that the $\overline{WR}$ control signals of the two processors are different. The $\overline{WR}$ pulse of the NSC800 begins one-half of a clock period (T-state) later than the $\overline{WR}$ Pulse of the 8085 pod. The trailing edge of each write pulse occurs at the same point in the cycle. The leading edge of the NSC800 $\overline{WR}$ pulse occurs when the data on the bus is guaranteed to be valid; this is not true of the 8085 pod $\overline{WR}$ Pulse.

If your NSC800 UUT latches data only on the trailing edge of the $\overline{WR}$ pulse, the above mentioned timing difference is not important; you can simply connect the $\overline{WR}$ pin of the 8085 pod to the $\overline{WR}$ line of the NSC800 UUT. If, however, the UUT performs some special action on the leading edge of the $\overline{WR}$ pulse, the 8085 pod $\overline{WR}$ pulse may have to be modified such that the leading edge occurs when valid data is on the bus.

The $\overline{WR}$ pulse of the 8085 pod can be adjusted to begin at almost the same point as the $\overline{WR}$ pulse of the NSC800. Note from the NSC800 timing diagram that the NSC800 $\overline{WR}$ pulse has a duration of about one T-state. Thus a rising-edge-triggered D flip-flop can be used to delay the start of the 8085 pod $\overline{WR}$ pulse by one-half T-state. The CLK output of the 8085 pod is used to clock the flip-flop. The circuit necessary for this function is shown in Figure 3 below.

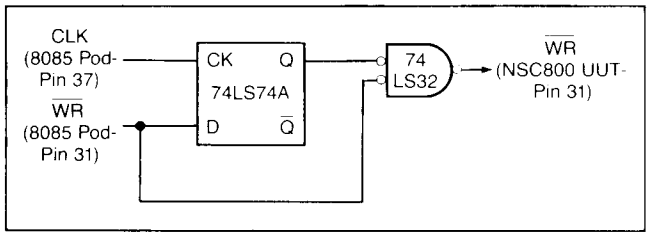


Figure 3. A Circuit for Adapting the 8085 pod $\overline{WR}$ signal to an NSC800 UUT.

Note that connecting the flip-flop between the 8085 pod and the NSC800 UUT cancels the pod's ability to check for drivability errors on the $\overline{WR}$ control line. For this reason, you should only use the circuitry of Figure 3 if the UUT requires it.

I/O Reads and Writes. The 8085 pod and the NSC800 feature identical I/O bus structures. However, there is a difference between the NSC800 and the 8085 in the timing of I/O operations. The NSC800 automatically inserts a wait state into an I/O read or write cycle. The 8085 does not have this characteristic; wait states must be generated by external circuitry. If your UUT requires this extra wait state, the adapter will need to include the circuitry necessary for generation of an 8085 wait state. Figure 4 below shows an example of a wait state generation circuit for the 8085 pod. The wait state will only be generated during an I/O operation. Note that the circuit of Figure 5 also allows wait states to be generated by the UUT through the use of the NSC800 $\overline{\text{WAIT}}$ line.

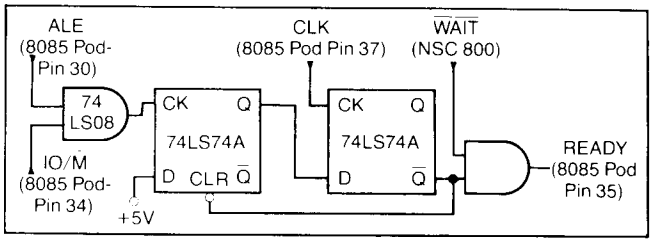


Figure 4. A wait state generation circuit for an 8085 pod/NSC800 adapter.

If your UUT does not require the extended I/O cycle of the NSC800, you can simply connect the WAIT line of the NSC800 directly to the READY pin of the 8085 pod. (Refer to Figure 5.)

Dynamic Ram Refresh Signals. The NSC800 microprocessor features on-chip dynamic RAM refresh control circuitry. However, few NSC800 UUT's use this capability. If yours does, you might have to design refresh control circuitry into the adapter. (The 8085 pod does not feature refresh control circuitry.) In this case the 8202 Dynamic RAM Controller should be considered. It is designed to be compatible with 8085 microprocessor characteristics.

The $\overline{\text{RFSH}}$ (Refresh) control signal of the NSC800 is used to alert circuits on the UUT that a refresh operation is taking place. If your NSC800 system uses the $\overline{\text{RFSH}}$ line, you may wish to use the 8085 pod to activate this signal for testing purposes. To do this, you can connect the SOD (serial output data) pin on the 8085 pod to the $\overline{\text{RFSH}}$ line on the NSC800 UUT. During normal troubleshooter operation, the refresh line should be tied to Vcc if it is to be held inactive, or tied to ground if it is to be held active. Connecting the $\overline{\text{RFSH}}$ line to the SOD line is useful only for testing the drivability of the $\overline{\text{RFSH}}$ line with the WRITE CTL and BUS TEST functions of the troubleshooter.

The Power Save Line (NSC800). The $\overline{\text{PS}}$ (Power Save) line of the NSC800 has no counterpart on the 8085 pod. However, many NSC800-based UUT's activate this pin to force the NSC800 into a "power save" mode. If you wish to use the 8085 pod to detect an active level on this line, you can connect $\overline{\text{PS}}$ to the SID (serial input data) pin on the 8085 pod. A READ STS command from the troubleshooter keyboard can then be used to detect the level on this line.

Adapting Interrupt Lines. All of the interrupt lines on the NSC800 UUT can easily be adapted to the 8085 pod. Figure 5 shows that only inverters are used to adapt these lines. If your UUT does not use an interrupt signal of the NSC800, you should tie the corresponding pin on the 8085 pod to its inactive (low) level.

Adapting DMA lines. There is a slight timing difference between the DMA signals of the 8085 and the NSC800. This timing difference is not critical in adapting the NSC800 DMA signals ($\overline{\text{BACK}}$ and $\overline{\text{BREQ}}$) to the corresponding 8085 pod signal (HLDA and HOLD). You can adapt these lines through the use of inverters (see Figure 5). If your UUT does not use the $\overline{\text{BREQ}}$ and $\overline{\text{BACK}}$ lines, you should tie the HOLD line of the 8085 pod to its inactive (low) level.

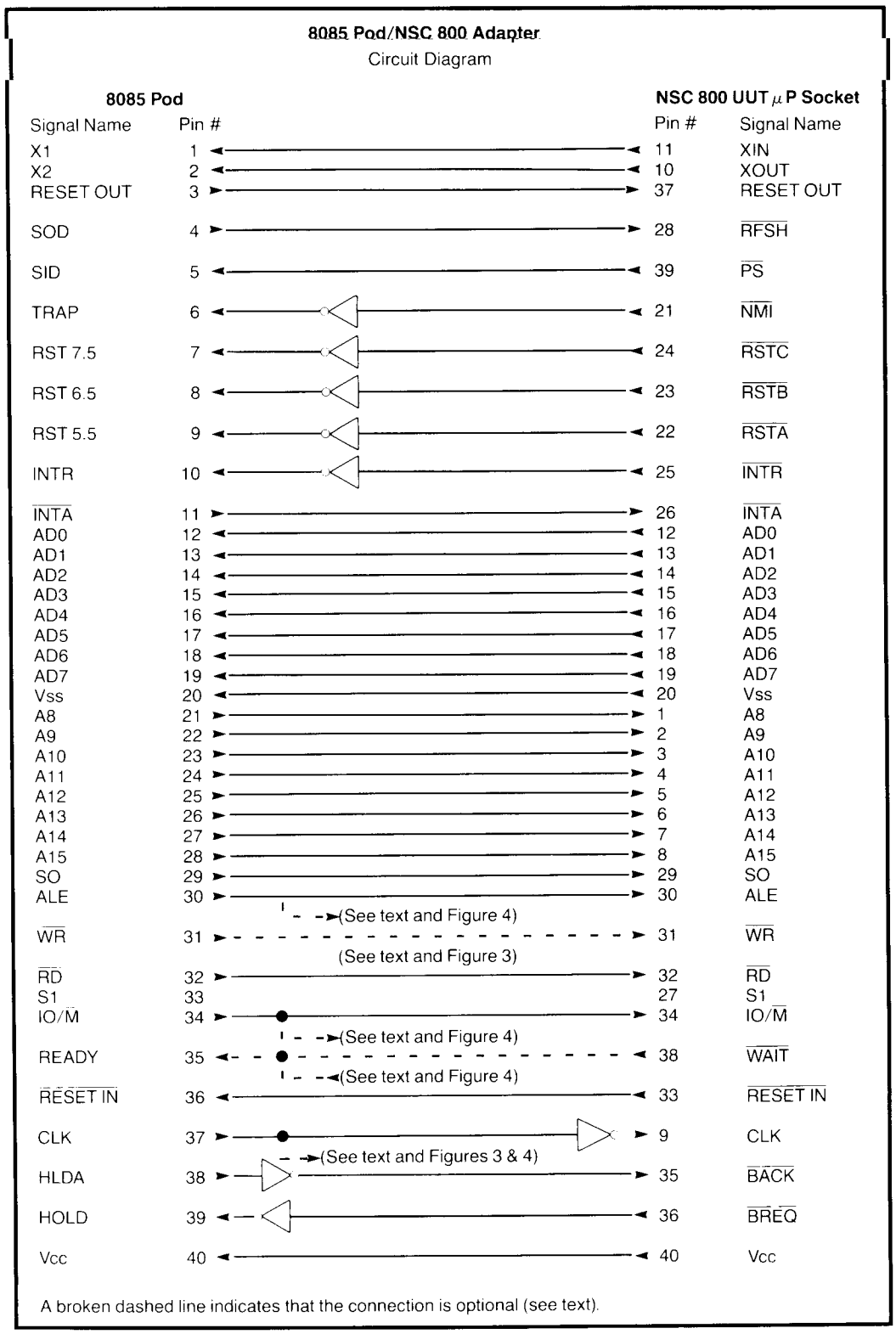


Figure 5. A schematic diagram of an 8085 Pod/ NSC800 adapter. Note that level shifting (buffer) circuitry is not included in the diagram.

FLUKE®

Using the Adapter with the 8085 Pod and the Troubleshooter

You should be able to use virtually all of the functions of the 9000-Series troubleshooters with the adapter. Refer to the troubleshooter operator manual for instructions regarding the troubleshooter. Refer to the 8085 pod instruction manual for instructions about using the pod. Some points of caution are described in the paragraphs below.

The RUN UUT Mode. The RUN UUT mode of the troubleshooter cannot be implemented in the usual way. This is because the instruction set of the NSC800 is a superset of the instruction set of the 8085 pod microprocessor (an 8085). Therefore the 8085 pod cannot execute NSC800 code.

In many cases, however, the RUN UUT mode of the troubleshooter can be implemented. You can load 8085 instructions into UUT RAM using the 9000-Series WRITE function. The RUN UUT mode can then be used to execute the 8085 code, starting with the initial RAM address of the code.

Status and Control Lines. The $\overline{\text{RFSH}}$, $\overline{\text{INTA}}$, RESET OUT, and $\overline{\text{BACK}}$ lines of the NSC800 UUT can all be "written to" using the WRITE CTL function of the troubleshooter. The $\overline{\text{RSTC}}$, $\overline{\text{RSTB}}$, $\overline{\text{RSTA}}$, $\overline{\text{PS}}$, $\overline{\text{RESET IN}}$, $\overline{\text{INTR}}$, $\overline{\text{NMI}}$, $\overline{\text{BREQ}}$, and $\overline{\text{WAIT}}$ lines of the UUT can all be "read" using the READ STS function. Remember that NSC800 μP input or output lines that have inverters between the UUT and the pod will have opposite Status or Control bit values.

The Probe. The troubleshooter probe can be used normally on the NSC800 UUT—if the UUT signal levels are compatible. (CMOS signal levels are probe-compatible if the CMOS circuitry is operated at Vcc=+5 volts.)

The Pod Adapter Packaging Kit

If you decide to construct a pod adapter, you will want to consider using the Pod Adapter Packaging Kit (available from Fluke). The kit will help you overcome the mechanical difficulties of constructing a pod adapter. The kit provides all of the parts necessary for housing the adapter circuitry, connecting the pod to the adapter, and connecting the adapter to the UUT. The kit contains:

- —A blank printed circuit board for mounting adapter components
- —A 40-pin ZIF socket
- —A "small pod" case (with the hardware necessary for mounting the circuit board)
- —A ribbon cable for connecting the adapter to the UUT
- —A blank decal for labeling the adapter
- —A copy of this technical note

The circuit board provides the necessary mounting holes and power supply/ground traces for 10-30 DIP sockets. The 40-pin ZIF (Zero Insertion Force) socket can be attached (mechanically and electrically) to one side of the circuit board; this allows the connection of the pod to the adapter. The circuit board also provides for connection of the ribbon cable that connects the adapter to the UUT. Pre-mounted wire-wrap stakes are provided for connection of user-mounted adapter components to the ZIF socket and the ribbon cable connection.

Contact your nearest Fluke Sales Office for pricing and delivery information. The model number of the Adapter Packaging Kit is 9000A-200.

Interface Pod Manuals

You'll want to order Interface Pod Instruction Manuals for pods that you are considering using for your adapter. You can order these manuals from your nearest Fluke Sales Office. When ordering, use the part bers listed below.

Manual	Part No.
9000A-Z80 Interface Pod Manual	613794
9000A-1802 Interface Pod Manual	649384
9000A-6502 Interface Pod Manual	613760
9000A-6800 Interface Pod Manual	613752
9000A-6802 Interface Pod Manual	649392
9000A-6809 Interface Pod Manual	649400
9000A-8048 Interface Pod Manual	649418
9000A-8080 Interface Pod Manual	613786
9000A-8085 Interface Pod Manual	613778
9000A-8086 Interface Pod Manual	649426
9000A-8088 Interface Pod Manual	649434
9000A-9900 Interface Pod Manual	613745
9000A-68000 Interface Pod Manual	652594

John Fluke Mfg. Co., Inc.
P.O. Box C9090, Everett, WA 98206
(800) 426-0361 (toll free) in most of U.S.A.
(206) 356-5400 from AK, HI, and WA
(206) 356-5500 from other countries

Fluke (Holland) B.V.
P.O. Box 5053, 5004 EB, Tilburg, The Netherlands
(013) 673-973. Tlx: 52237

Phone or write for the name of your local Fluke representative.

Printed in U.S.A. B0156A-01U8304/SE EN