Tektronix®
COMMITTED TO EXCELLENCE

INTFAC. RUN
1340 TO PDP-11 MULTI-INTERRUPT INTERFACE VERIFICATION PROGRAM
062-4816-00

INSTRUCTION MANUAL

Tektronix, Inc.
P.O. Box 500
Beaverton, Oregon 97077

070-3647-00
Product Group 70

PLEASE CHECK FOR CHANGE INFORMATION AT THE REAR OF THIS MANUAL.

First Printing JUN 1980
Revised DEC 1982

SOFTWARE SUPPORT POLICY

Unless otherwise provided, this software is furnished on an "as is" basis. Service, if available for this software, will be provided at the rates in effect at the time service is requested.

SOFTWARE LICENSE

Software supplied by Tektronix, Inc., as a component of a system or as a separate item is furnished under a license for use on a single system and can be copied (with the inclusion of copyright notice) only for use on that single system.

Products of Tektronix, Inc. and its subsidiaries are covered by U.S. and foreign patents and/or pending patents.

TEKTRONIX, TEK, SCOPE-MOBILE, and  are registered trademarks of Tektronix, Inc. TELEQUIPMENT is a registered trademark of Tektronix U.K. Limited.

Specification and price change privileges are reserved.

CONTENTS

PREFACE

It is assumed the reader is familiar with the 1340 Hardware and Software, the operation of the PDP-11 Controller, and TEKTEST III Programming Language.

The information in this manual applies not only to the 1340, but to the R1340 also.

GENERAL INFORMATION

INTRODUCTION

The 1340 (or R1340) Data Coupler interfaces the system controller and the rest of the test system. Except for the 2943/2944 Programmable Clock Generators, all other system components receive and transmit data to the controller via the 1340 Data Coupler. The 1340 also has capability to expedite non-processor transfer data from information.

This manual contains instructions for using the INTFAC program to test and verify the operation of the 1340 to PDP-11 Interface circuitry. Also included are troubleshooting guides to locate interface problems that may occur during execution of INTFAC.

It is not necessary that the interface be installed in a completed test system. Testing is limited to the PDP-11 to 1340 Interface circuitry and does not include the various interface and programming cards used in the 1340 Data Coupler.

Documentation Conventions

The terms "computer," "CPU," and "processor," as used in this document, are interchangeable. All terms refer to the system controller (PDP-11).

Where the term "instruction" is mentioned, it refers to a PDP-11 assembly language instruction. See the Digital Equipment Corp. *PDP-11 Processor Handbook* for more detail.

The system controller is assumed to be a PDP-11/34. Where use of another PDP-11 model would require different operator action, instructions are provided.

"CPU console" refers to the PDP-11 Programmer's Console located on the front panel of the computer.

Definitions of Abbreviations

Abbreviations used in this document are defined in the glossary located in Section 5.

Test Limitations

This program can only be used to test the 021-0105-00 or 021-0237-00 1340 to PDP-11 Multi-Interrupt Interface. It will not test the 021-0097-00 PDP-11 Interface.

Because the program does not know what cards are installed in 1340 slots 4–12, it does not check the select lines to the cards or the ready lines from the cards. It also does not check some of the more esoteric functions of the NM (which TEKTEST III does not use).

OPERATION

Loading and Running the Program

Install the program under the user's ident, **:MNT** using the TEKTEST III LOAD program.

NOTE

If the system does not boot, refer to Section 4 of this manual.

Verify that there are no other users on the system; other users will be aborted when the program begins execution.

Type **RUN INTFAC [.RUN][:MNT]** followed by a carriage return.

NOTE

The program can only be started from the console terminal.

The program will print a reminder that other users will be aborted and asks to continue. Answer Y or 1 to continue; answer O, N, or <RETURN> to exit.

The program will then prompt for the address of the interface to be tested. Respond with:

166000 for 1340 #1

166200 for 1340 #2

or the address of the interface under test as listed in your system documentation. Type a carriage return after entering the address to start testing the interface. As a test is finished, the test number will be printed on the terminal screen. There are 17 tests. The program will run until completion or until an error is found.

Upon completion, pressing any terminal key (except <CTRL>, <PAGE>, <RESET>, or <SHIFT>) will boot the system.

IMPORTANT

If the system drive at the time INTFAC is run is not an RK05, the program will not boot the system on completion. After completion, the user must boot the processor using the standard boot procedure. The only peripheral INFAC supports for bootstrap is the RK05 disk drive.

Looping on the Program

To loop on the program (loop on test) set switch register bit 11 on the programmer's console. This is accomplished in the following manner:

1. Press CLR (not applicable to PDP-11/35).
2. Enter 4000 via the keypad on the programmer's console (set switch register bit 11 up on PDP-11/35).
3. Press LSR (not applicable to PDP-11/35).

After Tests 1 through 17 are run, the terminal screen will be erased. The program will then loop on this until switch register bit 11 is cleared or an error occurs.

When an error does occur, INTFAC initiates a loop on the failed test routine. This allows troubleshooting the circuitry with the aid of an oscilloscope (thus the term "scope loop").

Restarting the Program

To restart the program after running a scope loop, without booting the system again, the following procedure may be used:

1. Halt the computer by simultaneously pressing CNTRL and HALT on the CPU programmer's console (set HALT/ENABLE switch down on PDP-11/35).

2. Press CLR. Then, press LAD to load restart address O into the computer. (Set switches 0-17 down, then momentarily depress LOAD ADRS on PDP-11/35).

3. Press CNTRL and START (lift HALT/ENABLE switch and press START on PDP-11/35).

The program will now start Test 1 and execute normally from this point.

NOTE

This method can only be used when the same interface is being tested again. The initial messages, which set the interface BASE ADDRESS, are not printed. Therefore, the address entered when the program was first started will be the BASE ADDRESS in use. If the 1340 loses power while the program is running, the system must be booted, and the test restarted. This is due, in part, to the 1340 power-down sequence.

Putting a Card on an Extender Without Losing the Scope Loop

1. Simultaneously press CNTRL and HALT on the CPU programmer's console (press HALT/ENABLE on PDP-11/35).

2. Turn off the 1340.

3. Put the card on an extender.

4. Restore power to the 1340.

5. Simultaneously press CNTRL and CONT on the CPU programmer's console (lift HALT/ENABLE switch and press CONT on PDP-11/35).

Test Sequence

The program checks the interface in the following sequence:

1. Check the ability of the PM to return slave sync ($\overline{SSYN}$) when it should. Also, check the ability of the DR to accept and transmit data.

2. Check the ability of the rest of the internal registers to accept and transmit data.

3. Check the internal ready lines: TCR=0, CELL FILLED, and UNIBUS TIMEOUT, using the addresses: BASE+122, BASE+132, and BASE+136.

4. Check the ability of the interface to initialize itself.

5. Check the ISC with interrupts disabled; then enable interrupts and test the IBAR and IAA.

6. Check the IM. Verify that the correct interrupt occurs when programmed.

7. Check the ability of the NM to stop on TCR=0.

8. Check the ability of the NM to stop on CELL FILLED.

9. Check the ability of the NM to stop on ABORT.

10. Check the ability of the NM to stop on INITIALIZE.

11. Check the ability of the NM to stop on UNIBUS TIMEOUT.

IMPORTANT

When INTFAC.RUN is used with the PDP-11/24 processor, tests 13 through 17 will not reliably run to completion. The processor should be halted, and the system booted, after test 12 has run.

12. Check the ability of the DR, TCR, and LTAR to increment or decrement correctly using NPTs.

13. Check the ability of the UTAR to increment correctly.

14. Check the ability of the NM to transfer data using Memory Management (not tested on systems without MM capability). Check the ability of the NM to transfer data from memory.

15. Check the ability of the NM to transfer data from memory while interrupts and programmed transfers are occurring. The memory locations read are incremented, then rewritten.

16. Check the ability of the NM to transfer data from memory without releasing the UNIBUS between NPT cycles.

17. Check the ability of the NM to transfer data to memory while the IM is running. The UNIBUS is released between NPT cycles.

On all error conditions that send an error message, except BUS ERROR and BAD INTERRUPT, the program sets up a scope loop.

Locating Schematic Areas

Cryptic notes, such as P3-6C5, are schematic-locating information. P3-6C5 means that the circuit is found on the P3 card, schematic sheet 6, on square C5. P3 <9> means P3 card, schematic sheet 9.

P1 is on sheets <1>, <2>, and <3>

P2 is on sheets <4> and <5>

P3 is on sheets <6>, <7>, <8>, and <9>

Two-digit U, Q and Y numbers are on the main boards (U31, Q20, etc.). Three-digit U, Q and Y numbers are located on the piggyback boards (U301, etc.). Three-digit C, CR and R numbers are located on the main boards, with the four-digit components located on the piggybacks.

Using the Switch Register to Help Troubleshoot

Reading Data from a Register

In places, this writeup will ask you to read a register. This is done as follows:

1. Press CNTRL and HALT simultaneously on the CPU console (press HALT/ENABLE on PDP-11/35).

2. Press CLR, then enter the address of the register using the console keypad.

NOTE

The address must be loaded into the PDP-11/35 switch register in binary. Set the address switches up for a one and down for a zero.

Example: Octal Address 7 6 6 1 2 4
Binary Address 111 110 110 001 010 100

3. Press LAD (LOAD ADRS on PDP-11/35).

4. Press EXAM. Information that is stored in the register will be displayed (in the data lights on PDP-11/35).

5. Repeat steps 2 through 4 to read other registers. Resume the scope loop by pressing CNTRL and CONT simultaneously (raise the HALT/ENABLE switch and press CONT on the PDP-11/35).

Loading Data into a Register

1. Follow steps 1 through 3, as in Reading Data from a Register above.

2. Press CLR, then enter the data to be loaded into the register using the console keypad (use the switch register as in step 2 above on the PDP-11/35).

3. Press DEP. The data is now loaded into the register.

4. Repeat the above for other registers.

Some Useful Addresses

Register	Software Name	Address
DR	DATA	BASE + 124
LIER	INTRA	BASE + 130
UIER	INTRB	BASE + 134
IBAR	ITVBAS	BASE + 140
ISC	INTSEL	BASE + 144
NSR	DMASTA	BASE + 150
NCR	DMACON	BASE + 154
LTAR	DMASLV	BASE + 160
UTAR	DMASLX	BASE + 164
TCR	DMAWC	BASE + 170
NMR	DMAMAS	BASE + 174
R0	—	777700
R1	—	777701
R2	—	777702
•		
•		
•		
R7	—	777707

for 1340 #1, BASE = 766000
and 1340 #2, BASE = 766200

Therefore, the address of the IBAR is:

766140 in 1340 #1
and 766340 in 1340 #2

ERROR MESSAGES

Error Messages: What They Mean and Some Troubleshooting Hints

The following error messages are listed in the order in which they are encountered in the program. Therefore, most of the hardware listed in previous error messages should have already been adequatedly checked out. That is, if the program gives you an error message, the portions of the interface implied by the preceding messages have been tested.

Refer to the *1340 to PDP-11 Multi-Interrupt Interface* manual (070-3192-00) when troubleshooting failures. The *PDP-11 Processor Handbook* and the *PDP-11 Peripherals Handbook*, published by Digital Equipment Corp., are also useful references.

ERROR MESSAGES FOR TESTS 1-4

Failed to Return $\overline{\text{SSYN}}$ on DATI-0

The computer tested the DR (TST @#DR) and the interface did not return $\overline{\text{SSYN}}$. The scope loop is doing a TST @#DR. Trigger on $\overline{\text{MSYN}}$ (P3- edge pin A21).

Possible problems:

1340 power off
P1 address not strapped correctly
UNIBUS cables not plugged in or defective
Typed in wrong address

Circuit areas that are suspect:

P1 card – UNIBUS Transceivers U41, U51, U52, U61, U62. <1>

Address Decoder U42, U54, U65, U74, U87. <1>

P3 card – PM (see P3 <8> for IC numbers)
Output gating P3-6A4: U13C, U21
Oscillator (CLOCK) P3-8D6

Check for BASE ADDRESS. If there is no pulse at BASE ADDRESS, the problem is on P1; otherwise, check P3.

Make sure the machine is getting all of the inputs it needs (BASE ADDRESS, $\overline{\text{MSYN}}$, $\overline{\text{DO STROBE}}$, $\overline{\text{NM ILLEGAL MASTER}}$) P3-8B1.

See what state the PM is in.

Check the oscillator and the output gating.

Failed to Return $\overline{\text{SSYN}}$ During Ready Check

The interface did not return $\overline{\text{SSYN}}$ when checking the ready status of the DR (TST @#DR+2).

Failed to Return $\overline{\text{SSYN}}$ During DATI+1

The interface did not return $\overline{\text{SSYN}}$ when the computer read the upper byte of the DR (TST @#DR+1).

Failed to Return $\overline{\text{SSYN}}$ During SSYN DATO+0

The interface did not return $\overline{\text{SSYN}}$ when the computer attempted to write the DR (MOV R5, @#DR).

Failed to Return $\overline{\text{SSYN}}$ During DATIP DATO+0

The interface did not return $\overline{\text{SSYN}}$ when the computer performed a read-modify-write to the DR (BIC R5 @#DR).

Incorrectly Returned $\overline{\text{SSYN}}$ During DATOB+1

The interface returned $\overline{\text{SSYN}}$ when the computer attempted to write to the upper byte of the DR (MOVB R5, @#DR+1).

Incorrectly Returned $\overline{\text{SSYN}}$ During DATOB+3

The interface returned $\overline{\text{SSYN}}$ when the computer attempted to write to the upper byte of the DR status word (MOVB R5, @#DR+3).

The following strategy applies to the preceding six error messages:

Put the P1 card on an extender. Trigger on $\overline{\text{1340 INITIALIZE}}$ P1- 1A1 (very long cycle time). The program is looping on the PDP-11 instruction that failed. Suspect areas are the Address Decoder (P1- 1B2: U54, U65, U74, U87) and the UNIBUS Transceivers (P3-6A4: U11).

Check for BASE ADDRESS. There should be no BASE ADDRESS in the cases when $\overline{\text{SSYN}}$ was incorrectly returned.

An Internal Register Has Dropped Some Data

<asource> is the address that was read from

<read> is the data that was read from that address

<expect> is the data that was expected

<write> is the data that was sent to that address

For all of the internal registers, except the ISC and IBAR, the program writes and reads all possible combinations of bits to each register, and checks that the correct data was read back. When an error is found, the program loops on writing and reading the register that failed. <asource> is the address of the register that failed, <read> the data read, and <expect> is the data expected. In some cases, writing to a register will return different bits set when read. <write> is the data written to the register.

If <asource> = DR

If <expect> is all ones and <read> is all zeros, or vice versa, then the control lines are a probable suspect.

If groups of two or four bits are wrong, then the P2 card is probably bad.

If an odd collection of bits are wrong, then the P1 card may not be generating the correct $\overline{\text{SELECT}}$ signal.

If the P2 card is suspected, put P2 on an extender, choose one of the bits that was wrong, and follow it from the UNIBUS through the UNIBUS data transceivers, the DBUS MUX, to the DR. The data should be good on the UNIBUS and through the transceivers and DBUS MUX while $\overline{\text{CONNECT UNIBUS D TO DBUS}}$ (P2-4E1) is low, and into the DR while $\overline{\text{LOAD DR}}$ (P2-5E1) and $\overline{\text{CONNECT UNIBUS D TO DBUS}}$ are low. While $\overline{\text{CONNECT DR TO UNIBUS D}}$ is low, follow the data from the DR through the transceivers and onto the UNIBUS.

If P1 or P3 is suspected, put P3 on an extender. Check for:

1. LOAD SAR (6B7: U76A pin 3); 200 ns positive pulse.
2. ASSERT SELECT (9D7: U180 pin 5); 600 ns positive pulse.
3. $\overline{\text{SEL DR}}$ (6C4: U45 pin 9); 600 ns negative pulse.

If LOAD SAR and ASSERT SELECT are present, but $\overline{\text{SEL DR}}$ is missing, then the P1 card is bad. If $\overline{\text{SEL DR}}$ is okay, then check:

4. $\overline{\text{CONNECT DR TO UNIBUS D}}$ (6B7: U13 pin 6); negative pulse.
5. $\overline{\text{LOAD DR}}$ (6D5: U28 pin 8); negative pulse.

If any of the above five signals, except $\overline{\text{SEL DR}}$, are missing, then check the logic that generates the missing signal. The PM may also be bad.

The state diagram for the PM shows when the crucial control signals should happen (see Figure 3-1).

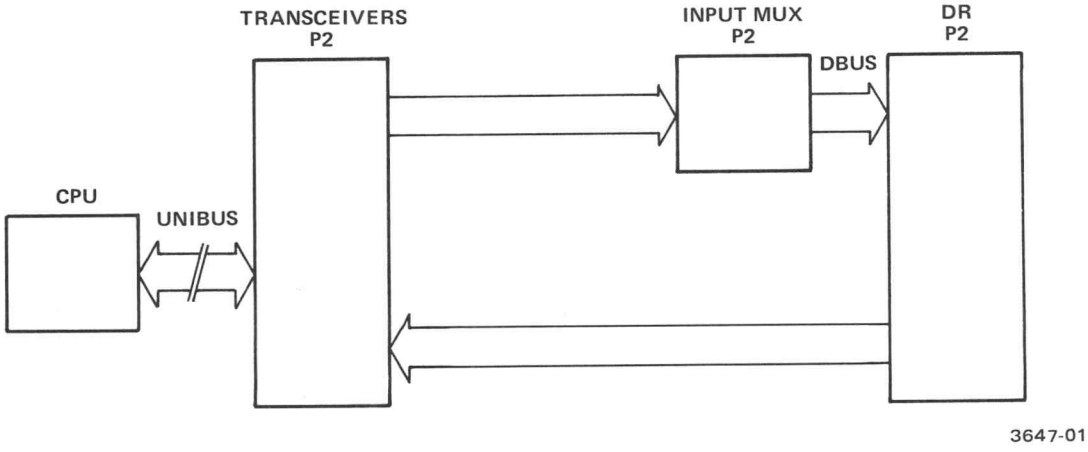


Figure 3-1. Data Path for BASE+124 (DR)

If <asource> = LIER, UIER, LTAR, or UTAR

Trigger on BASE ADDRESS

Suspect Circuits:

P1 – Select circuitry (SAR, SELECT MUX, SELECT DECODER)
DBUS MUX
Harmonica connectors
The register addressed (LIER, UIER, LTAR, or UTAR)

Strategy:

1. Check for the correct $\overline{\text{SELECT}}$.
2. If <write> = 177777, check control lines.
3. Follow a wrong bit through the path.

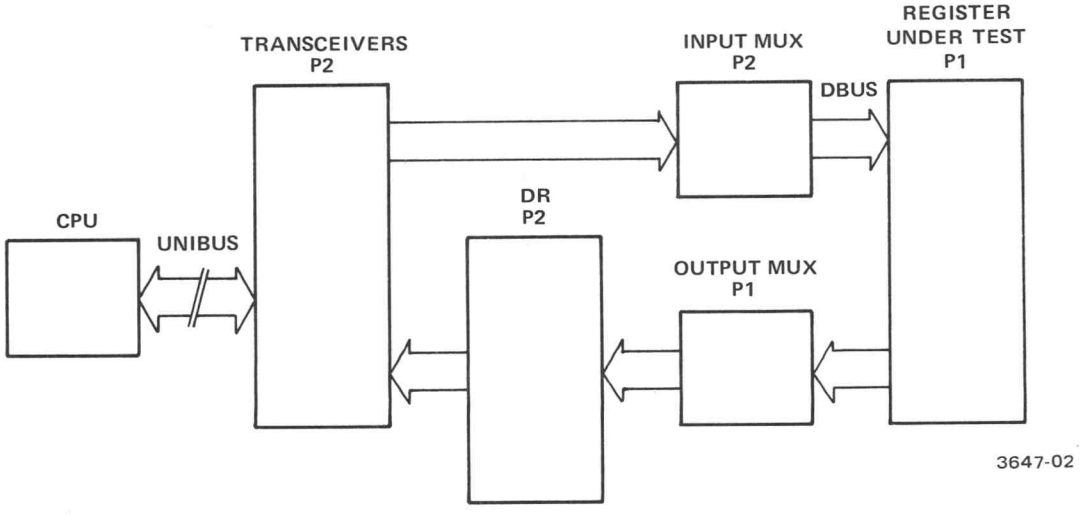


Figure 3-2. Data Path for BASE+130 (LIER), BASE+134 (UIER), BASE+160 (LTAR) and BASE+164 (UTAR)

If <asource> = NSR or NCR

Trigger on BASE ADDRESS.

Suspect Circuits:

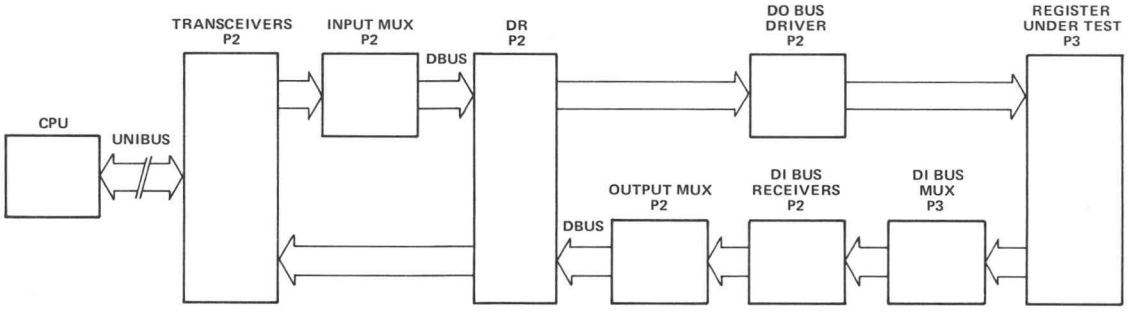
P1 – Select Circuitry

P2 – DO BUS Drivers
DI BUS Receivers

P3 – DI BUS Multiplexers
The register addressed

Strategy:

1. Check for correct $\overline{\text{SELECT}}$.
2. If <write> = 177777, check control lines.
3. Follow erroneous bit.



3647-03

Figure 3-3. Data Path for BASE+150 (NSR), BASE+154 (NCR)

If <asource> = TCR

Suspect DBUS Input MUX (P2 card). Use same strategy as above (check select, control, data path).

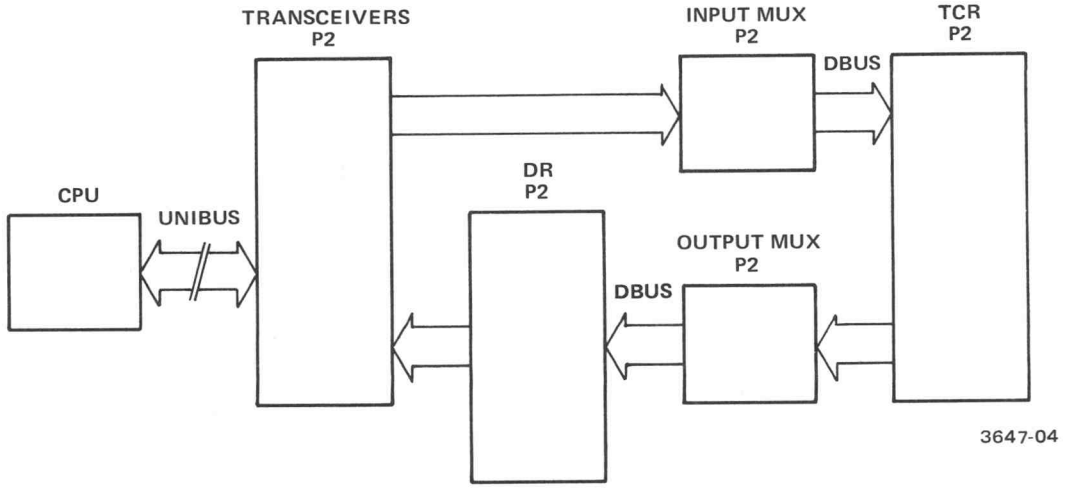


Figure 3-4. Data Path for BASE+170 (TCR)

If <asource> = NMR

Trigger on BASE ADDRESS. Suspect card select (P1), DBUS MUX (P1), NMR (P1). Check for correct $\overline{\text{SELECT}}$ signal and follow the erroneous bit.

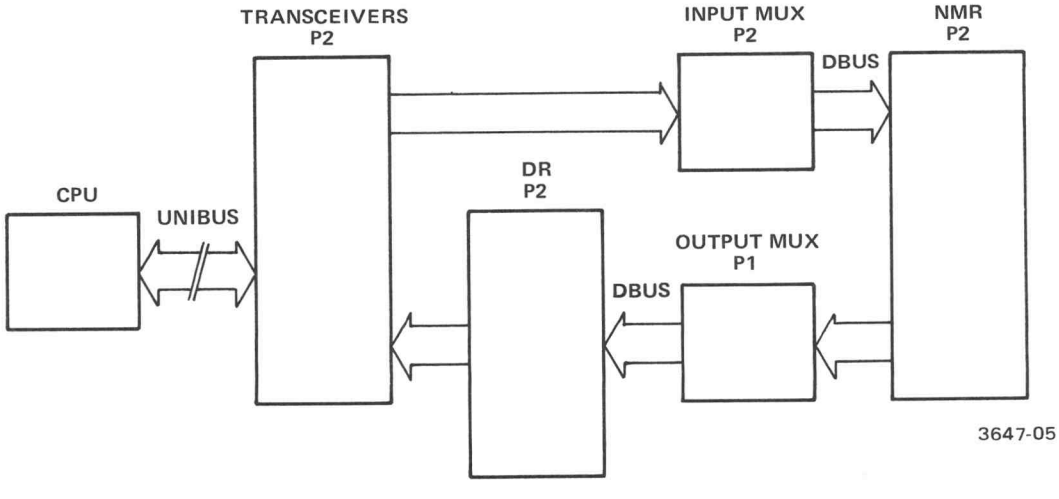


Figure 3-5. Data Path for BASE+174 (NMR)

<ready> Is the Address of an Internal Ready Line that Failed

Three status bits in the NSR can be read as if they were $\overline{\text{READY}}$ bits from 1340 cards:

NSR Bit	READY Address
1	BASE + 122
2	BASE + 132
3	BASE + 136

If the NSR bit is true, then the address shown in the above table should be READY (bit 15 true). <ready> is the address of the line that failed. The scope loop will be continously reading that address.

Read the NSR to determine whether the address should be ready. If it is ready, follow the $\overline{\text{READY}}$ information from the input of the PT Ready selector to the DI BUS. Suspect the PT Ready selector (IC6), P3: U48D, U75A, and the cabling between P1 and P3.

The Clear Instruction (CLR BASE+4) Caused the Interface to Bus Error

The PM was used to clear BASE+4. The interface failed to return $\overline{\text{SSYN}}$. The scope loop is looping on the PDP-11 instruction that failed (CLR @#BASE+4)

Trigger on CLEAR (P2-5A6: edge pin A42). Check if the PM gets to State 5 (State counter = 1000). If not, the $\overline{\text{INITIALIZE}}$ line may be shorted to $\overline{\text{CLEAR}}$. To verify this, check that $\overline{\text{INITIALIZE}}$ (P2-5A6: edge pin B42) is high. If not, follow the signal back from this point. Suspect U86, U83 (5A5, 5A6).

The Interface Did Not Initialize Correctly
<asource> is the address that was read from
<read> is the data that was read from that address
<expect> is the data that was expected

The program loads all possible internal registers with all ones (177777). It then initializes the interface (CLR @#BASE), and reads the registers to ensure they contain the correct data. After initialization, all registers should contain zero (0), except the IBAR, ISC and NSR. The program loads 40 into the UIER to stop the ISC at 124, so the ISC should contain 324 and the IBAR should read 124. The scope loop initializes the interface, and then tests the register that failed (TST @#reg) without checking results.

<asource> is the address of the register that did not initialize correctly. <read> is the data the register contained after initialization, and <expect> is the data expected.

Check which card the failing register is on and extend that card. Put the P3 card on an extender. Trigger on BASE ADDRESS, and check for $\overline{\text{INITIALIZE}}$ at the register indicated by <asource>.

Put the P3 card on an extender. Check that $\overline{\text{SEL INITIALIZE}}$ is present at the input of P3 U64C (7A2). If not, extend the P1 card and trace the signal back to P1 U71 pin 15 (1D5). Verify that $\overline{\text{PRGM INITIALIZE}}$ (P3-7A5) is being generated after $\overline{\text{SEL INITIALIZE}}$.

Circuitry:

P1 – Select Decoder (1D4)

P2 – Gating for $\overline{\text{INITIALIZE}}$ (5A5)

P3 – Gating for $\overline{\text{PRGM INITIALIZE}}$ (7A2-7A5)

Bus Error
The Computer is halted. Reboot is necessary.

The interface, or memory, did not return $\overline{\text{SSYN}}$. The UNIBUS, therefore, timed out. This was tested at the beginning of the program (Tests 1-4). A power line glitch, or turning off the 1340 during the test, can cause this message. If it is repeatable with no power problems, it may be the $\overline{\text{INITIALIZE}}$ circuitry. This can be checked by keying in and running the following program:

	Address	Data
	4	6
	6	2
Start here →	100	5037
	102	166000 (or 166200) ← 1340 Base Address
	104	137
	106	100

If $\overline{\text{MSYN}}$ (P3 edge pin A21) is low for 10 to 20 μs followed by a few 1 μs pulses and then repeats, proceed as for an $\overline{\text{INITIALIZE}}$ problem. Try to find out why the interface does not send back $\overline{\text{SSYN}}$.

ERROR MESSAGES FOR TESTS 5-6

Bit 7 of the ISC Is Incorrectly Set

When no interrupts are enabled, bit 7 of the ISC should not be set. The program initializes the interface, then checks to see if bit 7 of the ISC is zero. The scope loop is reading the ISC.

Trigger on BOX ADDRESS. Put P1 on an extender.

Circuitry:

P1 –	2C5: U311
P2 – Interrupt enabled scanner	3C5: U14, U24, U34, U75C

ISC Is Not Correct

<asource> is the address that was read from

<read> is the data that was read from that address

<expect> is the data that was expected

The program has enabled an interrupt, and set the corresponding interrupt condition true, so that the ISC should be stopped at a particular value. The processor priority is set to level 7 so the interrupt does not occur. <expect> contains the number the program expected to find in the ISC. <read> contains the number found in the ISC when the test failed. <asource> should contain the address of the ISC. The scope loop is reading the ISC.

Circuitry:

P1 – ISC	2D3: U312A, U410 2C3: U110B, U210C, U211A, U212A, U611A 2C2: U210A, U210B
P1 – DBUS MUX	2C5, 2D5: U311, U511 2C7, 2D7: U78A, U78B, U78D, U88D
P1 – Interrupt Request Scanner	3C2, 3D2, 3E2: U13, U23, U33
P1 – Interrupt Enabled Scanner	3C4, 3D4, 3E4: U14, U24, U34 3E5: U75C
P3 –	9B7: U630 8B8: U710B

Put P1 on an extender. Check to see if the ISC is counting, and if the interrupt signal is true. There are four possible cases:

CASE 1 – Counter counting and WANT INT true.
Check harmonica connector, U210C, U611A.

CASE 2 – Counter counting and WANT INT false.
Check for missing interrupt enable bit.
Check for corresponding ready bit.
Trigger on ISC6.
Check for counter incrementing correctly.
Check U13, U23, U33, U14, U24, U34, U75C, U78A, U78B, U88D.

CASE 3 – Counter not counting and WANT INT true.
Check U311, U511, and the ISC mentioned in CASE 2.
Check U13, U23, U33, U14, U24, U34, U75C, U78A, U78B, U88D.

CASE 4 – Counter not counting and WANT INT false.
Check U611A and U210C.
Check P3B: U630, U710B.

The IBAR or the IAA Is Not Correct

<asource> is the address that was read from

<read> was the data that was read from that address

<expect> is the data that was expected

<intbas> is the data being sent to the IBAR

<offset> + the data being sent should equal the data read

This part of the program checks the IBAR and the interrupt adder (IAA). The interface is supposed to add the contents of the ISC to the data sent to the IBAR, and return that sum when the IBAR is read. If the sum does not match that calculated by the program, the scope loop is entered. The scope loop is reading the IBAR.

Put P2 on an extender. Trigger on $\overline{\text{LOAD REG}}$, and verify that the data loaded into the IBAR (P2-4A3, 4B3, 4C3: U16, U26, U36, U46) matches <intbas> returned by the error message. Read the ISC. Verify that ISC bits 2-6 at the input of the IAA (P2-4A4, 4B4, 4C4) match the value returned by <offset> in the error message. Verify that the data on the DBUS while $\overline{\text{CON REG TO DBUS}}$ and $\overline{\text{SEL IBAR}}$ are low is the same as <offset> above. Verify that <read> matches the data on the output of the IAA. The IBAR output should equal <expect>, but when read contained <read>. Check the associated latches if bits are in error.

The Interface Interrupted Through A Wrong Vector

<badvec> is the vector that was used

<vector> is the vector that should have been used

The interface was programmed to generate an interrupt. The vector that the interface put on the UNIBUS was incorrect. The scope loop runs the IM without checking results.

Run the program again to see if it will stop elsewhere. If it stops here again, read the ISC and verify that bit 7 is true. Then read the IBAR to check if it is correct. If it contains the vector used, <badvec>, put P2 on an extender. Trigger on CON IAA TO DBUS (P2-4E1). Compare the vector used with the vector that should have been used, <vector>, and choose a bit that is different between the two. During the time CON IAA TO DBUS (P2-4E1) is low, trace the vector from the interrupt address register (P2-4A4, 4B4, 4C4: U15, U25, U35, U45) through the DBUS MUX (P2-4A7, 4B7, 4C7, 4E7) to the DR (P2-5C2, 5D2, tE2). While CON IAA TO DBUS and CON DR TO UNIBUS D are both low, trace the vector from the DR through the UNIBUS data transceivers (P2-5C6, 5D6, 5E7: U11, U21, U31, U41). Somewhere in the probram the bit should be changing state incorrectly.

NEVER READY Interrupted

The ready line corresponding to LIER bit 0, and internal address 0 should never be ready. LIER<0> was enabled and an interrupt occurred.

Read the LIER to check if bit 0 is still set and, if not, deposit a one. Read the ISC, ignoring bit 7, to determine the source of the interrupt. It should contain 0.

Read the UIER and the LIER. If any bits other than LIER<0> are set, they are not being set by the CPU. Find out what is setting them. If no other bits in the LIER or UIER are set, the problem is most likely in the interrupt request scanner (P1-3C2, 3D2, 3E2: U13, U23, U33) or in the interrupt enabled scanner (P1-3C4, 3D4, 3E4: U14, U24, U34).

An Interrupt Failed to Occur

The interface was programmed to generate an interrupt. The interrupt failed to occur and the scope loop was entered.

The scope loop enables the interrupt that failed and then waits for it to occur. If it does not occur within a fixed time, the IBAR, NSR and UIER are reloaded and the above procedure is repeated.

Halt the CPU and check if the UIER contains 20. If not, the UIER is not being loaded. Put P2 on an extender, trigger on the positive transition of LOAD REG • SEL UIER (P1-3B3: U75 pin 4) and verify the data being loaded into the UIER.

Possible causes are defective latches in the UIER (P1-3A3, 3B3: U28, U56).

If the UIER contained the correct data, other possibilities are:

1. Missing grant continuity cards in the CPU.
2. If 1340 #n is failing, the GM in 1340 #(n-1) is not passing the BUS GRANT (BG) to the next device.
3. The GM in the 1340 that is failing is bad. The GM is located on P3 (6A3, 6A4, 6B2, 6B3).

Trace the BG7 signal from the CPU toward the IM being tested and find out where it gets lost. BG7 is a high-true pulse. If there is a 1340 in front of the one being tested, pulses should be seen at BG7 IN and BG7 OUT of that 1340 (P3-6A2: edge pins A14 and A15).

A Bad Interrupt Occurred

The interface generated an interrupt when none was enabled. There is no scope loop for this problem.

Read the ISC to find the internal address that generated the interrupt. Check if the corresponding interrupt was enabled. If not, check the interrupt enabled scanner (P1-3C4, 3D4, 3E4).

ERROR MESSAGES FOR TESTS 7-13

Terminate on TCR=0 Does Not Work

<asource> is the address that was read from

<read> is the data that was read from that address

<expect> is the data that was expected

The program loaded 1 into the TCR and started a 1 word NPT (NCR=4001: stop on TCR=0), and then checked to see that the enable bit of the NCR (bit 0) had been cleared by the machine. The scope loop runs the NPT, but does not check the NCR.

This problem could be caused by:

1. The NM didn't run, in which case <asource> = NCR and <expect> = 4000.
2. The TCR didn't decrement, in which case <asource> = TCR and <expect> = 0.
3. The TERMINATE ON TCR=0 circuitry doesn't work, which may have the same error format as (2) above.

Put the P3 card on an extender. Check for the $\overline{\text{NPR}}$ signal (P3-6A5: edge connector pin A9). $\overline{\text{NPR}}$ is a low-true pulse about 2 μs wide. If $\overline{\text{NPR}}$ is missing, the NM didn't start. Check for positive level or pulses on NPT REQUEST (P3-7A8: square pin CW). NPT REQUEST must be high for the machine to run. If NPT REQUEST is always low, check around U84A, U84D (P3-7A2, 7A3). If $\overline{\text{NPR}}$ is present, check for DEC TCR (P3-6C7: U67 pin 3). If DEC TCR gets to the interconnecting cabling, check for TCR EMPTY at U66 pin 2 (P3-7B2). TCR EMPTY is high-true and lasts at least 3 μs. If it is there, find out why it doesn't go through U66A, U65D, and U53C to clear out U82A, U82B, U42A, U65D, and U47B to generate RELEASE UNIBUS. If TCR EMPTY can't be found, the problem is on the P2 card.

Put P2 on the extender. Check for DEC TCR at U54 pin 14 (P2-4C6). If the TCR is counting, find out why U85A (P3-4A6) doesn't generate TCR EMPTY when the TCR contains zero.

NOTE

You could stop the program and load zero into the TCR to check this out.

Terminate on Cell Filled Does Not Work

<asource> is the address that was read from

<read> is the data that was read from that address

<expect> is the data that was expected

The interface was programmed for two NPTs (DATIP, increment slave, terminate on cell filled (NCR=6222)). Either the wrong number of NPTs occurred, or the wrong value was found in the NSR after the NPTs were finished. The scope loop sets up NPTs, but doesn't check registers. Some clues to the cause of the failure may be found by examining the data returned by the error message. <asource> will be the address of some internal interface register, <read> will be the data read from that register, and <expect> will be the data that should have been found.

Put the P3 card on the extender. Trigger on NPR (P3-6A5: edge connector pin A9). Check that INC DR (P3-6D8), which is a 100 ns wide positive pulse, occurs about 2 ns after NPR. If INC DR is missing, check U36B, U46A (P3-6D7). Make sure the machine is running by checking for pulses on square pin CS (P3-9A8). Check for DR FULL by halting the computer, loading 177777 into the DR, and while the CPU is still stopped, check for a high level on the DR FULL line (P3-7D3 U53 pin 4). Press CNTRL/CONTINUE on the CPU console, and check for a positive pulse on the DR FULL line. If DR FULL is missing in either case, the problem is on P2 in the DR or DR FULL gates (P2-5C2, 5E2, 5D4: U12, U22, U32, U42, U85B). If the DR FULL pulse is present, trace the signal through U53B (P3-7D4), U62B (7D4), U46B (7B2), U65D (7B2), U53C (7B3), U47B (7B3) to generate the signals TERMINATE NPT and RELEASE UNIBUS.

Terminate on Abort Does Not Work

<asource> is the address that was read from

<read> is the data that was read from that address

<expect> is the data that was expected

The interface was programmed for a large number of NPTs (TCR=4000 and NCR=14001) and to stop on the ABORT signal. The ABORT line is pulsed, and the NSR is then checked to verify that the NM stopped because of the ABORT. <asource> should contain the address of NSR, <read> will contain the erroneous data found in the NSR, and <expect> should contain 100001 (the correct NSR data). The scope loop runs the NPTs without checking registers.

Put the P3 card on the extender. If NSR<0> is not set, trigger on the falling edge of DO 4-8 (P3-7D1: edge connector pin B38). Check for an ABORT pulse about 200 ns wide at edge pin B41 (P3-7E1). If no ABORT pulse is found, check U64A (7D6), U22C (7D6), R712 and (7D6). If an ABORT pulse is found, trace the signal through U32E (7E3), U53A (7E3), and U52B (7D4). If NSR<0> is set, trigger on ABORT and trace NSR<0> (7D4: U52 pin 9) through U46B (7B2). The rest of the path should already have been checked.

Terminate on Initialize Does Not Work

<asource> is the address that was read from

<read> is the data that was read from that address

<expect> is the data that was expected

The program starts NPTs (NCR=1), then initializes the interface (CLR @#BOX), and checks for NSR=10000 and NCR=0. <asource> will contain the address of the register which failed (NSR or NCR). <read> will be the data found in the register when the test failed, and <expect> will be the data that should have been there.

Put the P3 card on an extender. Trigger on 1340 INITIALIZE (P3-7E1: edge pin B42). Compare <read> with <expect> and note which bit is wrong. Find out why that bit is not correct.

Terminate on UNIBUS Timeout Did Not Work Correctly

<asource> is the address that was read from

<read> is the data that was read from that address

<expect> is the data that was expected

The NM was programmed to perform a DATO data transfer to 776770 (reserved by DEC for the ADO1 – unused by TEKTEST) which should have terminated with a UNIBUS timeout. The NM did not timeout. <asource> should be the address of the NSR, <read> is the incorrect data found when the NSR was read, and <expect> is the data which the NSR should have contained. The scope loop is looping on the NPT (DATO to 776770). The NCR is being loaded with 4002 to initiate the transfer.

This message can be generated if another UNIBUS device returns SSYN when 776770 is addressed. Turn off the R2943 and restart the test. If the test passes, look for missing pullup resistors on the UNIBUS address lines in the 2943. The 2943 is returning SSYN when it should not.

Put the P3 card on the extender. Trigger on N1 (P3-7C2: U74A pin 2), and check for TIMEOUT pulses on square pin BY (7A8). If pulses are not present, trace the signal from U74A to BY. If no pulses are present on N1, check the NM STATE COUNTER (P3-9B7: U630).

The TAR Did Not Increment Correctly

<asource> is the address that was read from

<read> is the data that was read from that address

<expect> is the data that was expected

The program set up the NM to do one NPT (NCR = 4401), and then checked to see that the LTAR was incremented by two. <asource> should be the address of the LTAR, <read> is the data that the LTAR contained after being incremented, and <expect> is the data the LTAR should contain. The scope loop runs the NPTs without checking the results.

If <expect>=<read>+2, INC TAR was not getting through. Put the P3 card on the extender. Trigger on NPR (edge pin A9), and check for INC TAR at U83C pin 8 (6C7). INC TAR is a 100 ns positive pulse occurring about 2 ns after the rising edge of NPR. If INC TAR is not found here, U83C and U67A (6C7) are suspect. Check that it gets off P3 to the interconnecting cables.

Put P1 on an extender. Check for pulses on edge pins A16 and A17. If any are observed, U41 (P1-1A2) is suspect. Check for INC TAR at U16 pin 14 (3E6).

If <read>+2 did not equal <expect>, note where the LTAR bits get fouled up as you go from the least significant to most significant bit. That is where the increment signal was stopped. The LTAR is U16, U26, U36, U46, U65C (P1-3B6, 3C6, 3D6, 3E6).

The TCR Did Not Decrement Correctly

<asource> is the address that was read from

<read> is the data that was read from that address

<expect> is the data that was expected

The NM was programmed to stop on DR FULL (NCR = 6221). So that the NM will terminate after one NPT cycle, -2 is placed in the slave address. The program runs the NPTs and then checks to see if the TCR decremented correctly. <asource> should be the TCR, <read> is the data the TCR contained when the test failed, and <expect> is the data the program expected to find in the TCR. The scope loop runs the NPTs without checking the results.

Put the P2 card on the extender. Trigger on DEC TCR. Follow the decrement signal through the DR until you find where it gets fouled up. The TCR is on P2-4A6, 4B6, 4C6: U54, U64, U74, U84.

The DR Did Not Increment Correctly

<asource> is the address that was read from

<read> is the data that was read from that address

<expect> is the data that was expected

The NM was programmed to run a single NPT (NCR = 4221, TCR = 1) in the increment slave mode. The DR was then checked to see that the DR was incremented. <asource> is the address the NM transferred data to, <read> is the data that was transferred, and <expect> is the data that was expected. The scope loop runs the NPTs without checking the results.

Put the P2 card on the extender. Trigger on INC DR (P2-5E1). Follow the increment signal through the DR until you find where the register is not incrementing correctly. The DR is on P2-5C2, 5D2.

The UTAR Did Not Increment Correctly

<asource> is the address that was read from

<read> is the data that was read from that address

<expect> is the data that was expected

The program set all bits in the LTAR, and then incremented the UTAR by clearing the LTAR. The UTAR did not increment correctly. <asource> should be the address of the UTAR, <read> is the value read back from the UTAR, and <expect> is the expected value of the UTAR. The scope loop is incrementing the UTAR.

Put the P1 card on the extender. Trigger on the carry from the LTAR (P1-3B6: U48 pin 14). Check that U48 is incrementing correctly. If the output of U48 is right, suspect the DBUS MUX (P1-3E7: U18).

ERROR MESSAGES FOR TESTS 14-17

NM Did Not Transfer a Data Word Correctly
An Address Transceiver is Probably Bad

<asource> is the address that was read from

<read> is the data that was read from that address

<expect> is the data that was expected

<coreloc> is the core location that was read

The interface was programmed to do DATI NPTs, increment the TAR, start on DR ready, and terminate on TCR=0. The test failed in one of the following modes:

CASE 1 – <asource> = NSR
<read> = NSR data
<expect> = expected NSR contents
<coreloc> = two words: contents of the UTAR, and LTAR.

An NPT, which should have caused the NM to timout, returned incorrect status to the NSR. <coreloc> is the 18-bit address of the memory location that was read.

CASE 2 – <asource> = DR
<read> = DR contents
<expect> = expected DR contents
<coreloc> = (same as in CASE 1)

The NM transferred a data word incorrectly. If any bits do not match in <read> and <expect>, suspect the UNIBUS data bus transceiver(s) corresponding to the same bits.

NM Did Not Transfer a Data Word Correctly
An Address Transceiver is Probably Bad
Memory Management Is Turned On

<asource> is the address that was read from

<read> is the data that was read from that address

<expect> is the data that was expected

<coreloc> is the core location that was read

Memory management was turned on. The interface was programmed to do DATI NPTs, increment the TAR, start on DR ready, and terminate on TCR=0. The test failed in one of the following modes:

CASE 1 – <asource> = NSR
<read> = NSR data
<expect> = expected NSR contents
<coreloc> = two words: contents of the UTAR and LTAR

An NPT, which should have caused the NM to timeout, returned incorrect status to the NSR. <coreloc> is the 18-bit address of the memory location that was read.

CASE 2 – <asource> = DR
<read> = DR contents
<expect> = expected DR contents
<coreloc> = (same as in CASE 1)

The NM transferred a data word incorrectly. If any bits do not match in <read> and <expect>, suspect the UNIBUS data bus transceiver(s) corresponding to the same bits.

An Unexpected Interrupt Happened While All 3 Machines Were Running Press CONTINUE to Re-boot

The interface was programmed to run single NPTs (increment slave, start on DR ready, terminate on TCR=0) while the PM and IM (interrupt on TCR=0) were running. A power line glitch or invalid 1340 address (BASE+0 through BASE+114, BASE+130, or BASE+134) interrupted. There is no scope loop for this test.

Restart the program. If the program stops at the same point, disconnect any other TEK peripherals from the UNIBUS. Install a UNIBUS terminator after the 1340 under test (see Figures 3-6 and 3-7).

Restart the program again. If the problem still exists, a DEC peripheral is probably defective. Call DEC service.

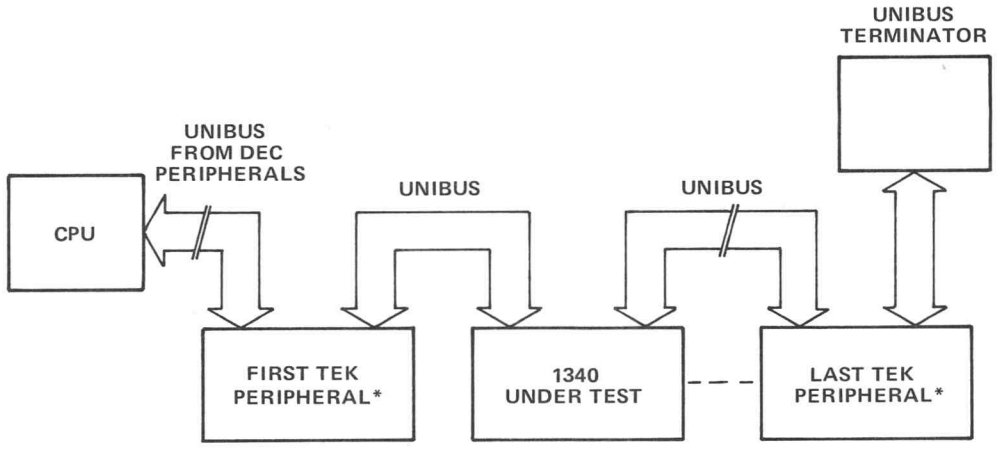


Figure 3-6. Normal UNIBUS Connections

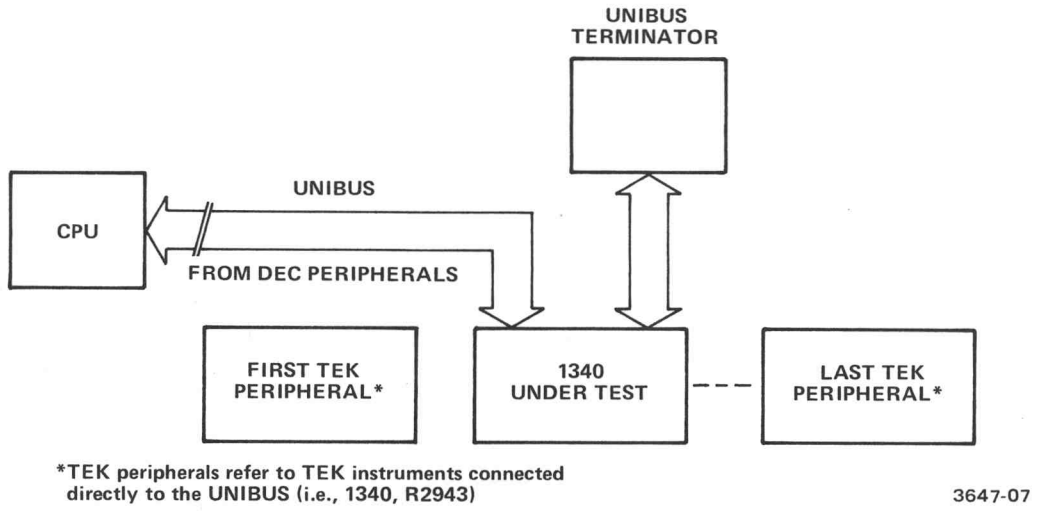


Figure 3-7. UNIBUS Connections to Isolate the 1340 Under Test

The NM Did Not Do DATIP Correctly While the IM and the PM Were Running

<asource> is the address that was read from

<read> is the data that was read from that address

<expect> is the data that was expected

The Computer is Halted. Press CONTINUE for Scope Loop

The interface was programmed to run single NPTs (increment slave, start on DR ready, terminate on TCR=0) while the PM and IM (interrupt on TCR=0) were running. The results of an NPT did not match the expected value.

The scope loop sets up the NM and PM machines as above, then loads 125252 into the DR. When the DR comes ready, the NM runs single NPTs until the TCR contains 0. At this point, the NM is finished, and the IM handles the interrupt generated by TCR=0, NSR<1>.

The most likely cause of a failure at this point is the TCR not being decremented all the time. If available, obtain a logic analyzer. Connect the CLOCK input to DEC TCR, trigger on TCR=0, and observe the TCR output. You should see a discontinuity in the count sequence when the NM fails.

Connect the logic analyzer CLOCK input to CLK (8D7, square pin CE) and observe the NM output (9A7, 9B7). Trigger the analyzer on NM State 27 (N<0:6> = 160). Refer to the NM state diagrams in the interface manual, and check that the state machine follows the correct sequence.

The NM Did Not Do DATI Data Transfers Correctly

<asource> is the address that was read

<dsource> is the data that was read from that address

<read> is the data that was expected

The Computer is Halted. Press CONTINUE for Scope Loop.

The NM was programmed to transfer <expect> words from memory to <asource> (the DR). The data read back from the DR, <read>, was incorrect. Either the number of words the NM transferred was wrong, the DR is dropping bits, or the IM ran when it shouldn't have. The scope loop will run NPTs (DATI, terminate on TCR=0, hog the bus), and interrupt on ALWAYS READY (UIER<5>) after the transfer is complete. The TCR should be loaded with <expect> and the NCR will contain 4402 when the NPT is initiated.

The NM Did Not Transfer Data Correctly During DATO, Single NPTs, INC TAR, and Terminate on TCR=0

<asource> is the address that was read from

<read> is the data that was read from that address

<expect> is the data that was expected

The NM was programmed to transfer 1-200 words to memory from the DR. The DR should contain the data which initiated the NPT (in this case, the contents of the NCR: 4441). One of two exits to the error routine was taken:

CASE 1 – If <expect> equals 4441, the data read back from the NPT destination was incorrect. <read> is the data transferred, and <asource> is the address of the destination which failed.

CASE 2 – If <expect> equals 0, the NM ran more than the programmed number of NPTs. <read> should equal 4441, and <asource> is the address the NM transferred data to incorrectly. The NM was expected to stop at memory location <asource>-2. The LTAR contained <asource>-2 after the NPTs were run.

Drive Was Not Ready. Press Any Key to Re-boot.

The interface test has finished, but the RK05 could not transfer data to memory to re-boot. Possible causes:

1. Drive was not loaded. Check for ready, LOAD/RUN switch to RUN.
2. Defective hardware. Attempt to boot using the standard procedure. If this doesn't work, see Section 4.

NOTE

When a terminal key is pressed, the program will attempt to boot to the system drive in use when the test was started.

CATASTROPHIC FAILURES

If You Can't Boot Up the System

1. Turn off the 1340 (if more than one, do one 1340 at a time) and try to boot up again.
2. Unplug interface cards P1, P2 and P3 and try to boot up.
3. Disconnect UNIBUS cable at the computer. Install a UNIBUS terminator in its place and try to boot up.

If the system boots after step 1, turn the 1340 back on and attempt to run INTFAC (the PDP-11/1340 Interface Test). If the program halts, check all UNIBUS lines with the 1340 power on and the CPU halted. The following lines should be high (>3 V):

P1 edge connector – A16 through A32; B32 <1>

P2 edge connector – A23 through A38; A22, B22 (+5 V) <4>

P3 edge connector – A6 through A10; A13; A18 through A21 <6>

The following lines should be low:

P3 edge connector – A11, A12; A14 through A17 <6>

If all else fails, check for shorts between UNIBUS lines with the system power off.

If the system boots after step 2, but not step 1, plug the cards back in and check the UNIBUS with the 1340 power off and the CPU halted.

If the system boots after step 3, but not step 2, you have a cable problem. Check the UNIBUS lines with the 1340 cards removed, the 1340 power off, and the UNIBUS connected to the computer.

If the system does not boot after step 3, call DEC service.

Program Does Not Finish

If the test number printed is from 7-17, the IM may not be receiving the BG7 signal. This is checked in test 6, but may fail after being checked. Halt the CPU and examine the LIER and the UIER. If an interrupt is enabled, troubleshoot the GM as for "An Interrupt Failed to Occur" error.

If the BG7 signal is not being sent by the computer, the NM may be waiting for NPG. Again, the GM is the likely suspect.

If the above conditions do not exist, record the following information as it may be useful in tracking down the problem.

– contents of the following CPU internal registers:

Register	Address	Description
R0	777700	General-purpose register
.	.	.
.	.	.
.	.	.
R5	777705	General-purpose register
R6	777706	Stack pointer (SP)
R7	777707	Program counter (PC)
R11	777711	Copy of last source operand
R12	777712	Copy of last destination operand
R13	777713	Copy of last instruction
R14	777714	Copy of last interrupt vector used

– contents of the interface internal registers

GLOSSARY

ABBREVIATIONS, ACRONYMS AND MNEMONICS

$\overline{\text{BBSY}}$	Bus Busy
$\overline{\text{BG}}$	Bus Grant
$\overline{\text{BR}}$	Bus Request
D bus	16-bit bus within the PDP-11 Interface P1 and P2 cards only.
DI bus	16-bit bus in 1340, on which data moves from instrument interface cards to the PDP-11 Interface.
DO bus	16-bit bus in 1340, on which data moves from the PDP-11 Interface to instrument interface cards.
DATI	Master requests 16-bit word from slave; Data In.
DATIP	Master requests word from slave, then slave is to wait for DATO or DATOB; Data In Pause.
DATO	Master sends 16-bit word to slave; Data Out.
DATOB	Master sends 8-bit byte to slave; Data Out Byte.
DMA	Direct Memory Access, synonymous with NPT.
DR	Data Register
GM	Grant Machine
IAA	Interrupt Address Adder
IBAR	Interrupt Base Address Register
IM	Interrupt Machine
$\overline{\text{INTR}}$	Interrupt
ISC	Interrupt Scan Counter
LIER	Lower Interrupt Enable Register
LTAR	Lower Transfer Address Register
$\overline{\text{MSYN}}$	Master Sync
NCR	NPT Control Register
NM	NPT Machine
NMR	NPT Master Register
$\overline{\text{NPG}}$	Non-Processor Grant
$\overline{\text{NPR}}$	Non-Processor Request
NPT	Non-Processor Transfer
NSR	NPT Status Register

PI	Programmed Interrupt
PM	Programmed Transfer Machine
PT	Programmed Transfer
$\overline{\text{SACK}}$	Selection Acknowledge
SAR	Select Address Register
SD	Select Decoder
SM	Select Multiplexer
$\overline{\text{SSYN}}$	Slave Sync
TCR	Transfer Count Register
TAR	Transfer Address Register; used synonymously with LTAR.
UIER	Upper Interrupt Enable Register
UNIBUS	Single, common set of signal wires that connect the processor, memory, and peripherals (see Digital Equipment Corp. *PDP-11 Peripherals Handbook* section on UNIBUS theory and operation).
UTAR	Upper Transfer Address Register

PROGRAM EXECUTION SAMPLES

```
INTFAC V03.00:  PDP-11 TO 1340 INTERFACE TEST

THIS PROGRAM TERMINATES ALL PROCESSING AND REBOOTS ON COMPLETION.
CONTINUE? Y   (OR 1)
ENTER LOWEST ADDRESS OF INTERFACE TO BE TESTED  166000
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
PRESS ANY KEY TO RE-BOOT
```

The above execution of INTFAC passed all tests. As each test is finished, the test number is printed. As can be seen above, a message is given indicating all processing terminates. All background users should be warned of this, giving them time to complete their work.

```
INTFAC V03.00:  PDP-11 TO 1340 INTERFACE TEST

THIS PROGRAM TERMINATES ALL PROCESSING AND REBOOTS ON COMPLETION.
CONTINUE? Y
ENTER LOWEST ADDRESS OF INTERFACE TO BE TESTED  166000
1
2
3
4
5
AN INTERRUPT FAILED TO OCCUR
```

In the above sample, an interrupt did not occur during execution of test 6. The test has gone into a loop to allow troubleshooting of the circuitry (see Section 3, page 3-11, "An Interrupt Failed to Occur").

```
INTFAC V03.00: PDP-11 TO 1340 INTERFACE TEST

THIS PROGRAM TERMINATES ALL PROCESSING AND REBOOTS ON COMPLETION.
CONTINUE? 1
ENTER LOWEST ADDRESS OF INTERFACE TO BE TESTED  166000
1
AN INTERNAL REGISTER HAS DROPPED SOME DATA
166150 IS THE ADDRESS THAT WAS READ FROM
100010 IS THE DATA THAT WAS READ FROM THAT ADDRESS
010000 IS THE DATA THAT WAS EXPECTED
000000 IS THE DATA THAT WAS SENT TO THAT ADDRESS
```

Above, on test 2, an internal register has dropped some data and the program is in a scope loop to allow troubleshooting. Listed is the information concerning the address and data (see Section 3, page 3-3, "An Internal Register Has Dropped Some Data").