TEXAS INSTRUMENTS

JTAG/MPSD Emulation

Technical Reference

1994

Printed in U.S.A., December 1994
2617709–9741 revision A

SPDU079A

JTAG/MPSD Emulation Technical Reference

December 1994
SPDU079

Printed on Recycled Paper

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

TRADEMARKS

ABEL is a trademark of DATA I/O.

PAL® is a registered trademark of Advanced Micro Devices, Inc.

Contents

Figures

Chapter 1

Designing for JTAG Emulation

This chapter assists you in meeting the design requirements of the XDS510 emulator with respect to JTAG designs and discusses the XDS510 cable (manufacturing part number 2617698-0001). This cable is identified by a label on the cable pod marked **JTAG 3/5V** and supports both standard 3-volt and 5-volt target system power inputs.

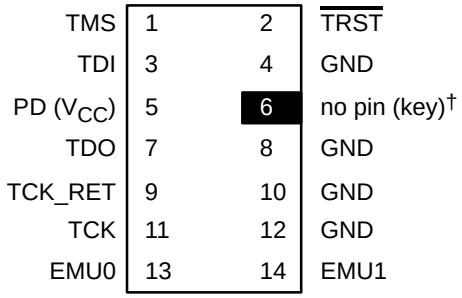
The term *JTAG,* as used in this book, refers to TI scan-based emulation, which is based on the IEEE 1149.1 standard.

Topic **Page**

1.1 Designing Your Target System's Emulator Connector (14-Pin Header)

JTAG target devices support emulation through a dedicated emulation port. This port is a superset of the IEEE 1149.1 standard and is accessed by the emulator. To communicate with the emulator, **your target system must have a 14-pin header** (two rows of seven pins) with the connections that are shown in Figure 1–1. Table 1–1 describes the emulation signals.

Figure 1–1. 14-Pin Header Signals and Header Dimensions



Header Dimensions:
Pin-to-pin spacing, 0.100 in. (X,Y)
Pin width, 0.025-in. square post
Pin length, 0.235-in. nominal

† While the corresponding female position on the cable connector is plugged to prevent improper connection, the cable lead for pin 6 is present in the cable and is grounded, as shown in the schematics and wiring diagrams in this document.

Table 1–1. 14-Pin Header Signal Descriptions

Signal	Description	Emulator† State	Target† State
TMS	Test mode select	O	I
TDI	Test data input	O	I
TDO	Test data output	I	O
TCK	Test clock. TCK is a 10.368-MHz clock source from the emulation cable pod. This signal can be used to drive the system test clock	O	I
$\overline{\text{TRST}}$‡	Test reset	O	I
EMU0	Emulation pin 0	I	I/O
EMU1	Emulation pin 1	I	I/O
PD(V_{CC})	Presence detect. Indicates that the emulation cable is connected and that the target is powered up. PD should be tied to V_{CC} in the target system.	I	O
TCK_RET	Test clock return. Test clock input to the emulator. May be a buffered or unbuffered version of TCK.	I	O
GND	Ground		

† I = input; O = output

‡ Do not use pullup resistors on $\overline{\text{TRST}}$: it has an internal pulldown device. In a low-noise environment, $\overline{\text{TRST}}$ can be left floating. In a high-noise environment, an additional pulldown resistor may be needed. (The size of this resistor should be based on electrical current considerations.)

Although you can use other headers, recommended parts include:

straight header, unshrouded	DuPont Connector Systems part numbers: 65610–114
	65611–114
	67996–114
	67997–114

1.2 Bus Protocol

The IEEE 1149.1 specification covers the requirements for the test access port (TAP) bus slave devices and provides certain rules, summarized as follows:

- The TMS/TDI inputs are sampled on the rising edge of the TCK signal of the device.
- The TDO output is clocked from the falling edge of the TCK signal of the device.

When these devices are daisy-chained together, the TDO of one device has approximately a half TCK cycle setup to the next device's TDI signal. This type of timing scheme minimizes race conditions that would occur if both TDO and TDI were timed from the same TCK edge. The penalty for this timing scheme is a reduced TCK frequency.

The IEEE 1149.1 specification does not provide rules for bus master (emulator) devices. Instead, it states that it expects a bus master to provide bus slave compatible timings. The XDS510 provides timings that meet the bus slave rules.

1.3 IEEE 1149.1 Standard

For more information concerning the IEEE 1149.1 standard, contact IEEE Customer Service:

Address: IEEE Customer Service
445 Hoes Lane, PO Box 1331
Piscataway, NJ 08855-1331

Phone: (800) 678–IEEE in the US and Canada
(908) 981–1393 outside the US and Canada

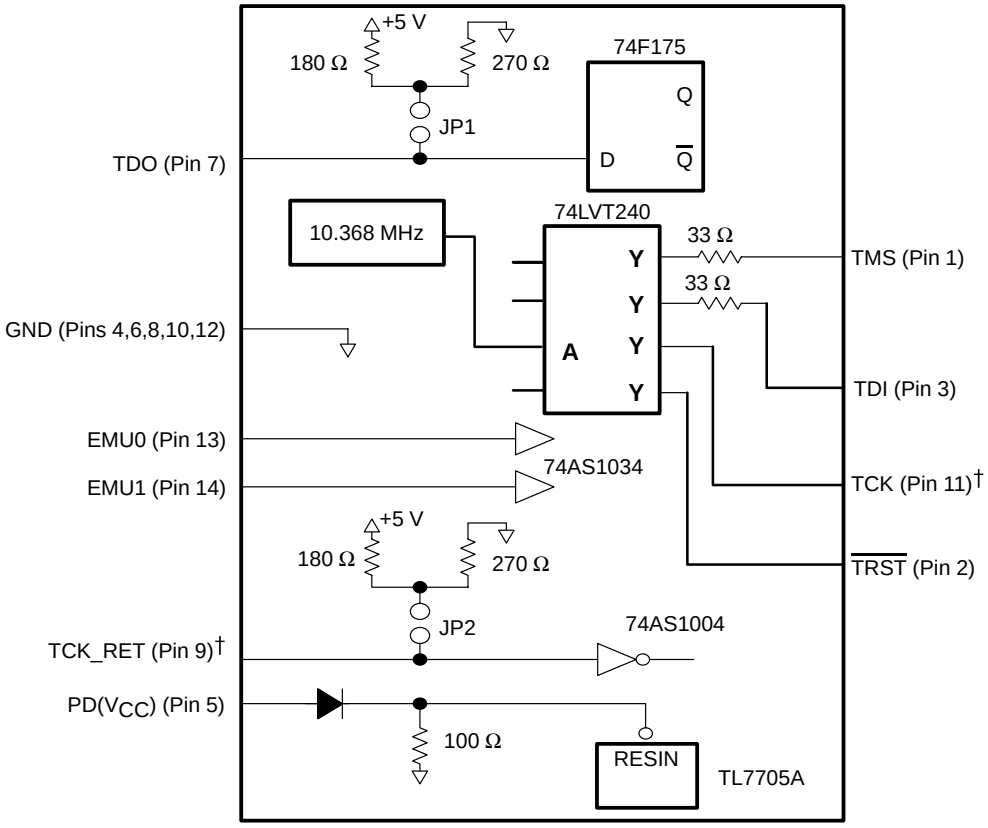
FAX: (908) 981–9667 Telex: 833233

1.4 JTAG Emulator Cable Pod Logic

Figure 1–2 shows a portion of the emulator cable pod. These are the functional features of the pod:

- Signals TDO and TCK_RET can be parallel-terminated inside the pod if required by the application. By default, these signals are not terminated.
- Signal TCK is driven with a 74LVT240 device. Because of the high-current drive (32 mA I_{OL}/I_{OH}), this signal can be parallel-terminated. If TCK is tied to TCK_RET, then you can use the parallel terminator in the pod.
- Signals TMS and TDI can be generated from the falling edge of TCK_RET, according to the IEEE 1149.1 bus slave device timing rules.
- Signals TMS and TDI are series-terminated to reduce signal reflections.
- A 10.368-MHz test clock source is provided. You may also provide your own test clock for greater flexibility.

Figure 1–2. JTAG Emulator Cable Pod Interface



† The emulator pod uses TCK_RET as its clock source for internal synchronization. TCK is provided as an optional target system test clock source.

1.5 JTAG Emulator Cable Pod Signal Timing

Figure 1–3 shows the signal timings for the emulator cable pod. Table 1–2 defines the timing parameters. These timing parameters are calculated from values specified in the standard data sheets for the emulator and cable pod and are for reference only. Texas Instruments does not test or guarantee these timings.

The emulator pod uses TCK_RET as its clock source for internal synchronization. TCK is provided as an optional target system test clock source.

Figure 1–3. JTAG Emulator Cable Pod Timings

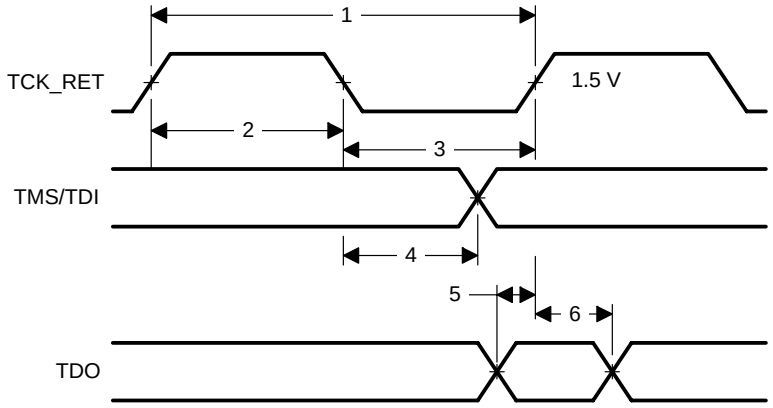


Table 1–2. Emulator Cable Pod Timing Parameters

No.	Reference	Description	Min	Max	Units
1	$t_{c(TCK)}$	TCK_RET period	35	200	ns
2	$t_{w(TCKH)}$	TCK_RET high-pulse duration	15		ns
3	$t_{w(TCKL)}$	TCK_RET low-pulse duration	15		ns
4	$t_{d(TMS)}$	Delay time, TMS/TDI valid from TCK_RET low	6	20	ns
5	$t_{su(TDO)}$	TDO setup time to TCK_RET high	3		ns
6	$t_{h(TDO)}$	TDO hold time from TCK_RET high	12		ns

1.6 Emulation Timing Calculations

The following examples help you calculate emulation timings in your system. For actual target timing parameters, see the appropriate device data sheets.

Assumptions:

$t_{su(TTMS)}$	Target TMS/TDI setup to TCK high	10 ns
$t_{d(TTDO)}$	Target TDO delay from TCK low	15 ns
$t_{d(bufmax)}$	Target buffer delay, maximum	10 ns
$t_{d(bufmin)}$	Target buffer delay, minimum	1 ns
$t_{(bufskew)}$	Target buffer skew between two devices in the same package: $[t_{d(bufmax)} - t_{d(bufmin)}] \times 0.15$	1.35 ns
$t_{(TCKfactor)}$	Assume a 40/60 duty cycle clock	0.4 (40%)

Given in Table 1–2 (on page 1-5):

$t_{d(TMSmax)}$	Emulator TMS/TDI delay from TCK_RET low, maximum	20 ns
$t_{su(TDOmin)}$	TDO setup time to emulator TCK_RET high, minimum	3 ns

There are two key timing paths to consider in the emulation design:

- The TCK_RET-to-TMS/TDI path, called $t_{pd(TCK\_RET–TMS/TDI)}$, and
- The TCK_RET-to-TDO path, called $t_{pd(TCK\_RET–TDO)}$.

Of the following two cases, the worst-case path delay is calculated to determine the maximum system test clock frequency.

Case 1: Single processor, direct connection, TMS/TDI timed from TCK_RET low.

$$t_{pd\,(TCK\_RET–TMS/TDI)} = \frac{\left[t_{d\,(TMSmax)} + t_{su\,(TTMS)}\right]}{t_{(TCKfactor)}}$$

$$= \frac{[20ns + 10ns]}{0.4}$$

$$= 75ns\ (13.3\ MHz)$$

$$t_{pd\,(TCK\_RET–TDO)} = \frac{\left[t_{d\,(TTDO)} + t_{su\,(TDOmin)}\right]}{t_{(TCKfactor)}}$$

$$= \frac{[15ns + 3ns]}{0.4}$$

$$= 45ns\ (22.2\ MHz)$$

In this case, the TCK_RET-to-TMS/TDI path is the limiting factor.

Case 2: Single/multiprocessor, TMS/TDI/TCK buffered input, TDO buffered output, TMS/TDI timed from TCK_RET low.

$$t_{pd\,(TCK\_RET-TMS/TDI)} = \frac{\left[t_{d\,(TMSmax)} + t_{su\,(TTMS)} + t_{(bufskew)}\right]}{t_{(TCKfactor)}}$$

$$= \frac{\left[20ns + 10ns + 1.35ns\right]}{0.4}$$

$$= 78.4ns\ (12.7\ MHz)$$

$$t_{pd\,(TCK\_RET-TDO)} = \frac{\left[t_{d\,(TTDO)} + t_{su\,(TDOmin)} + t_{d\,(bufmax)}\right]}{t_{(TCKfactor)}}$$

$$= \frac{[15ns + 3ns + 10ns]}{0.4}$$

$$= 70ns\ (14.3\ MHz)$$

In this case, the TCK_RET-to-TMS/TDI path is the limiting factor.

In a multiprocessor application, it is necessary to ensure that the EMU0–1 lines can go from a logic low level to a logic high level in less than 10 μs. This can be calculated as follows:

$$t_r = 5(R_{pullup} \times N_{devices} \times C_{load\_per\_device})$$
$$= 5(4.7\ k\Omega \times 16 \times 15\ pF)$$
$$= 5.64\ \mu s$$

1.7 Connections Between the Emulator and the Target System

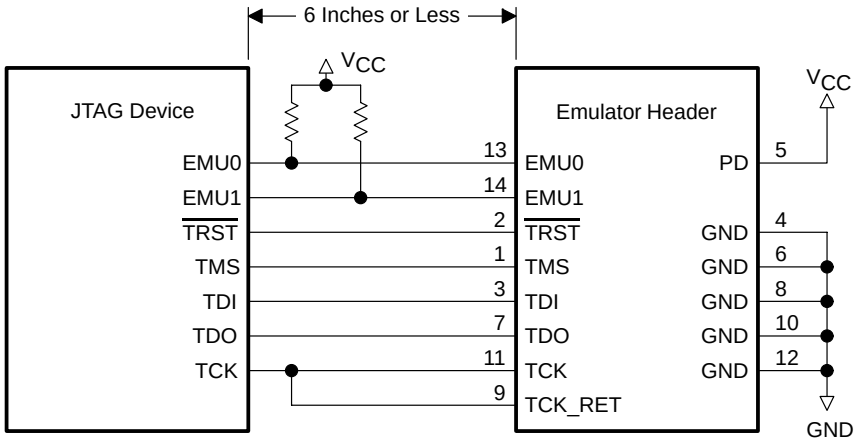
It is extremely important to provide high-quality signals between the emulator and the JTAG target system. Depending upon the situation, you must supply the correct signal buffering, test clock inputs, and multiple processor interconnections to ensure proper emulator and target system operation.

Signals applied to the EMU0 and EMU1 pins on the JTAG target device can be either input or output (I/O). In general, these two pins are used as both input and output in multiprocessor systems to handle global run/stop operations. EMU0 and EMU1 signals are applied only as inputs to the XDS510 emulator header.

1.7.1 Buffering Signals

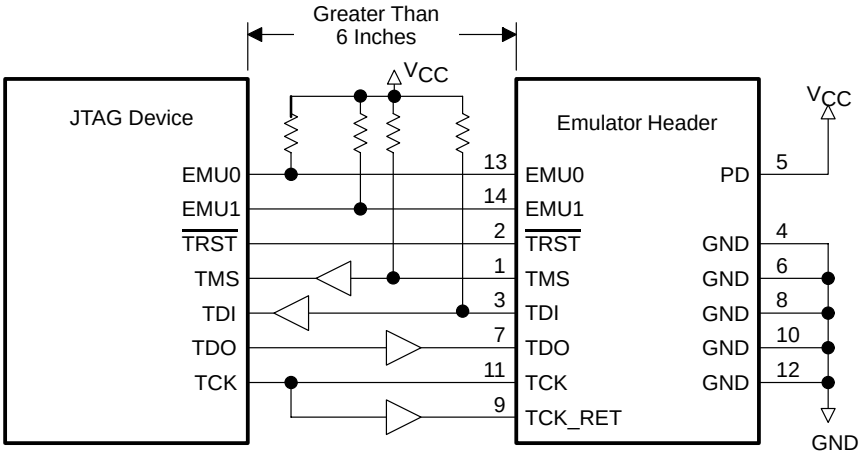
If the distance between the emulation header and the JTAG target device is greater than six inches, the emulation signals must be buffered. If the distance is less than six inches, no buffering is necessary. The following illustrations depict these two situations.

- **No signal buffering.** In this situation, the distance between the header and the JTAG target device should be no more than six inches.



The EMU0 and EMU1 signals must have pullup resistors connected to V_{CC} to provide a signal rise time of less than 10 μs. A 4.7-kΩ resistor is suggested for most applications.

❑ **Buffered transmission signals.** In this situation, the distance between the emulation header and the processor is greater than six inches. Emulation signals TMS, TDI, TDO, and TCK_RET are buffered through the same package.

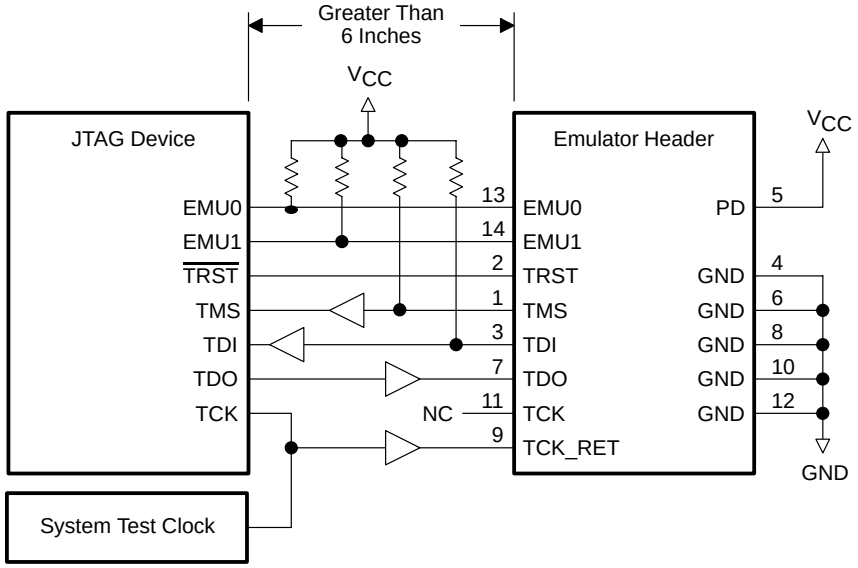


- The EMU0 and EMU1 signals must have pullup resistors connected to V_{CC} to provide a signal rise time of less than 10 µs. A 4.7-kΩ resistor is suggested for most applications.
- The input buffers for TMS and TDI should have pullup resistors connected to V_{CC} to hold these signals at a known value when the emulator is not connected. A resistor value of 4.7 kΩ or greater is suggested.
- To have high-quality signals (especially the processor TCK and the emulator TCK_RET signals), you may have to employ special care when routing the PWB trace. You also may have to use termination resistors to match the trace impedance. The emulator pod provides optional internal parallel terminators on the TCK_RET and TDO. TMS and TDI provide fixed series termination.
- Since $\overline{\text{TRST}}$ is an asynchronous signal, it should be buffered as needed to insure sufficient current to all target devices.

1.7.2 Using a Target-System Clock

Figure 1–4 shows an application with the system test clock generated in the target system. In this application, the TCK signal is left unconnected.

Figure 1–4. Target-System-Generated Test Clock



Note: When the TMS/TDI lines are buffered, pullup resistors should be used to hold the buffer inputs at a known level when the emulator cable is not connected.

There are two benefits to having the target system generate the test clock:

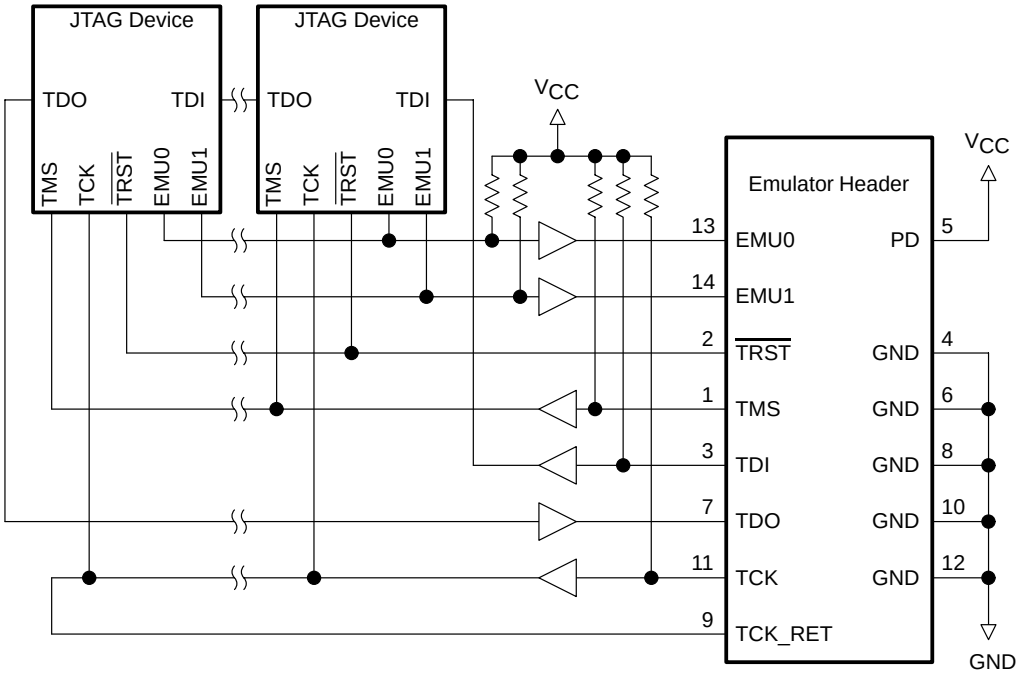
- The emulator provides only a single 10.368-MHz test clock. If you allow the target system to generate your test clock, you can set the frequency to match your system requirements.
- In some cases, you may have other devices in your system that require a test clock when the emulator is not connected. The system test clock also serves this purpose.

1.7.3 Configuring Multiple Processors

Figure 1–5 shows a typical daisy-chained multiprocessor configuration, which meets the minimum requirements of the IEEE 1149.1 specification. The emulation signals in this example are buffered to isolate the processors from the emulator and provide adequate signal drive for the target system. One of the benefits of this type of interface is that you can generally slow down the test clock to eliminate timing problems. You should follow these guidelines for multiprocessor support:

- The processor TMS, TDI, TDO, and TCK signals should be buffered through the same physical package for better control of timing skew.
- The input buffers for TMS, TDI, and TCK should have pullup resistors connected to V_{CC} to hold these signals at a known value when the emulator is not connected. A resistor value of 4.7 kΩ or greater is suggested.
- Buffering EMU0 and EMU1 is optional but highly recommended to provide isolation. These are not critical signals and do not have to be buffered through the same physical package as TMS, TCK, TDI, and TDO. Unbuffered and buffered signals are shown in this section (page 1-8 and page 1-9).

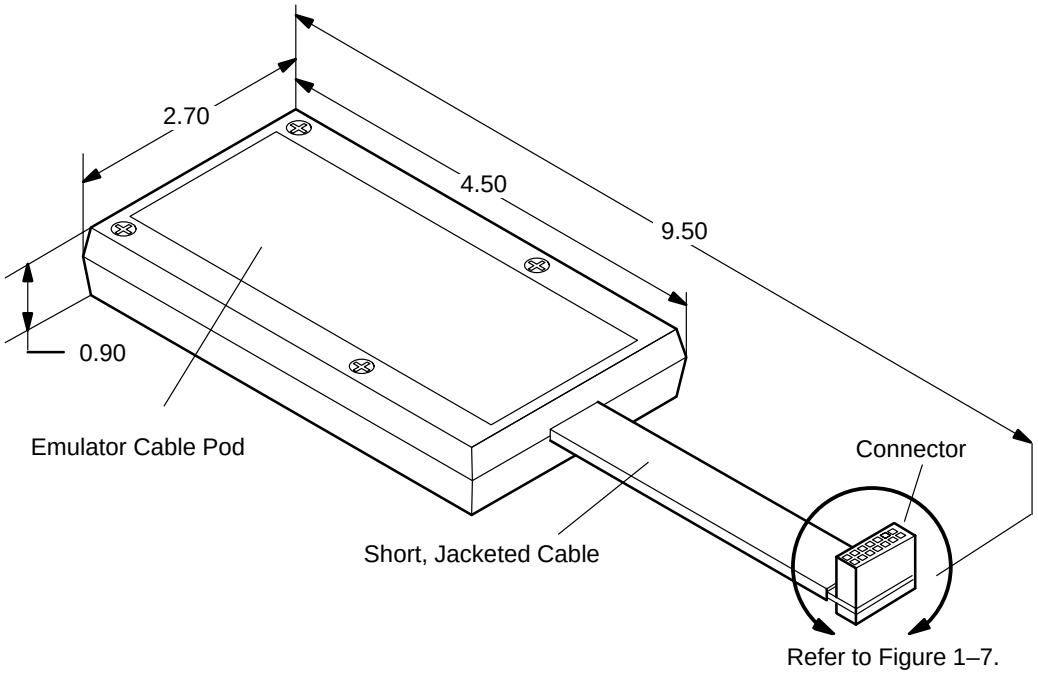
Figure 1–5. Multiprocessor Connections



1.8 Mechanical Dimensions for the 14-Pin Emulator Connector

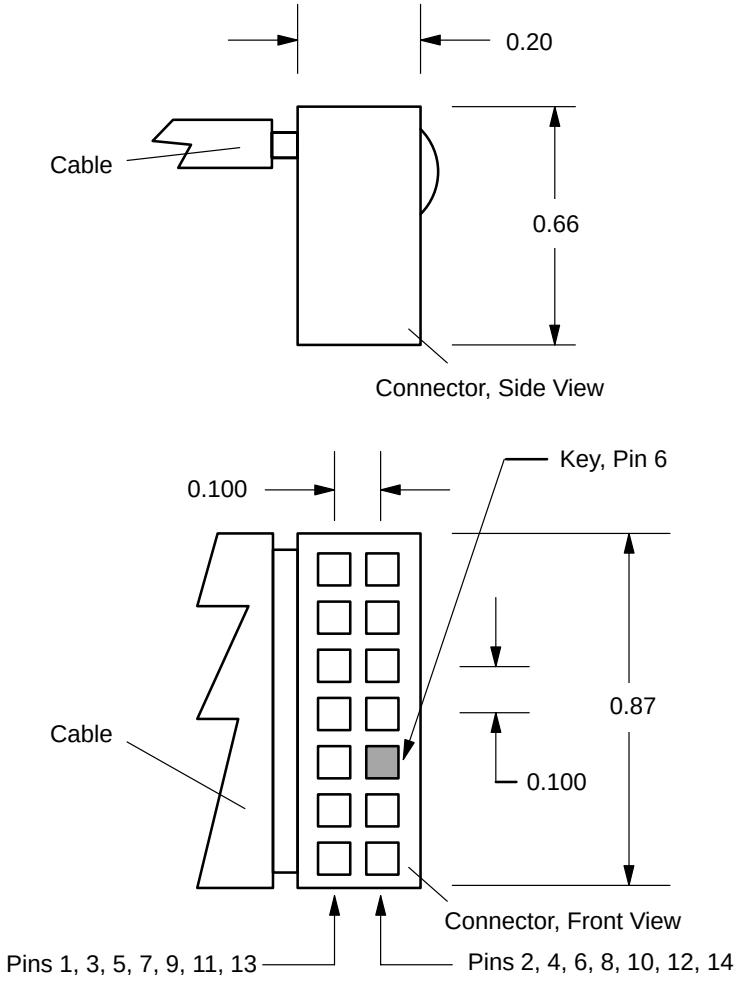
The JTAG emulator target cable consists of a 3-foot section of jacketed cable, an active cable pod, and a short section of jacketed cable that connects to the target system. The overall cable length is approximately 3 feet 10 inches. Figure 1–6 and Figure 1–7 (page 1-13) show the mechanical dimensions for the target cable pod and short cable. Note that the pin-to-pin spacing on the connector is 0.100 inches in both the X and Y planes. The cable pod box is nonconductive plastic with four recessed metal screws.

Figure 1–6. Pod/Connector Dimensions



Note: All dimensions are in inches and are nominal dimensions, unless otherwise specified.

Figure 1–7. 14-Pin Connector Dimensions



Note: All dimensions are in inches and are nominal dimensions, unless otherwise specified.

1.9 Emulation Design Considerations

This section describes the use and application of the scan path linker (SPL), which can simultaneously add all four secondary JTAG scan paths to the main scan path. It also describes the use of the emulation pins and the configuration of multiple processors.

1.9.1 Using Scan Path Linkers

You can use the TI ACT8997 scan path linker (SPL) to divide the JTAG emulation scan path into smaller, logically connected groups of 4 to 16 devices. As described in the *Advanced Logic and Bus Interface Logic Data Book* (literature number SCYD001), the SPL is compatible with the JTAG emulation scanning. The SPL is capable of adding any combination of its four secondary scan paths into the main scan path.

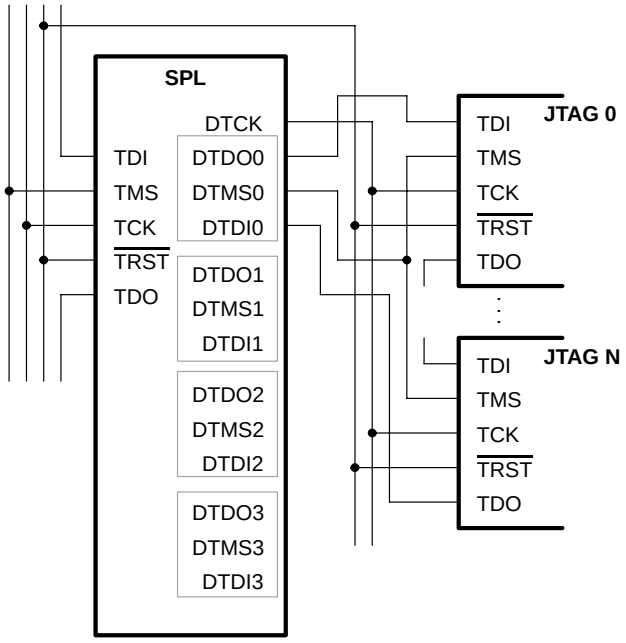
A system of multiple, secondary JTAG scan paths has better fault tolerance and isolation than a single scan path. Since an SPL has the capability of adding all secondary scan paths to the main scan path simultaneously, it can support global emulation operations, such as starting or stopping a selected group of processors.

TI emulators do not support the nesting of SPLs (for example, an SPL connected to the secondary scan path of another SPL). However, you can have multiple SPLs on the main scan path.

Although the ACT8999 scan path selector is similar to the SPL, it can add only one of its secondary scan paths at a time to the main JTAG scan path. Thus, global emulation operations are not assured with the scan path selector. For this reason, scan path selectors are not supported.

You can insert an SPL on a backplane so that you can add up to four device boards to the system without the jumper wiring required with nonbackplane devices. You connect an SPL to the main JTAG scan path in the same way you connect any other device. Figure 1–8 shows you how to connect a secondary scan path to an SPL.

Figure 1–8. Connecting a Secondary JTAG Scan Path to an SPL



The $\overline{\text{TRST}}$ signal from the main scan path drives all devices, even those on the secondary scan paths of the SPL. The TCK signal on each target device on the secondary scan path of an SPL is driven by the SPL's DTCK signal. The TMS signal on each device on the secondary scan path is driven by the respective DTMS signals on the SPL.

DTDO on the SPL is connected to the TDI signal of the first device on the secondary scan path. DTDI on the SPL is connected to the TDO signal of the last device in the secondary scan path. Within each secondary scan path, the TDI signal of a device is connected to the TDO signal of the device before it. If the SPL is on a backplane, its secondary JTAG scan paths are on add-on boards; if signal degradation is a problem, you may need to buffer both the $\overline{\text{TRST}}$ and DTCK signals. Although less likely, you may also need to buffer the DTMS*n* signals for the same reasons.

1.9.2 Emulation Timing Calculations for SPL

The following examples help you to calculate the emulation timings in the SPL secondary scan path of your system. For actual target timing parameters, see the appropriate device data sheets.

Assumptions:

$t_{su(TTMS)}$	Target TMS/TDI setup to TCK high	10 ns
$t_{d(TTDO)}$	Target TDO delay from TCK low	15 ns
$t_{d(bufmax)}$	Target buffer delay, maximum	10 ns
$t_{d(bufmin)}$	Target buffer delay, minimum	1 ns
$t_{(bufskew)}$	Target buffer skew between two devices in the same package: $[t_{d(bufmax)} - t_{d(bufmin)}] \times 0.15$	1.35 ns
$t_{(TCKfactor)}$	Assume a 40/60 duty cycle clock	0.4 (40%)

Given in the SPL data sheet:

$t_{d(DTMSmax)}$	SPL DTMS/DTDO delay from TCK low, maximum	31 ns
$t_{su(DTDLmin)}$	DTDI setup time to SPL TCK high, minimum	7 ns
$t_{d(DTCKHmin)}$	SPL DTCK delay from TCK high, minimum	2 ns
$t_{d(DTCKLmax)}$	SPL DTCK delay from TCK low, maximum	16 ns

There are two key timing paths to consider in the emulation design:

- The TCK-to-DTMS/DTDO path, called $t_{pd(TCK–DTMS)}$, and
- The TCK-to-DTDI path, called $t_{pd(TCK–DTDI)}$.

Of the following two cases, the worst-case path delay is calculated to determine the maximum system test clock frequency.

Case 1: Single processor, direct connection, DTMS/DTDO timed from TCK low.

$$t_{pd\,(TCK–DTMS)} = \frac{\left[t_{d\,(DTMSmax)} + t_{d\,(DTCKHmin)} + t_{su\,(TTMS)}\right]}{t_{(TCKfactor)}}$$

$$= \frac{[31ns + 2ns + 10ns]}{0.4}$$

$$= 107.5ns\ (9.3\ MHz)$$

$$t_{pd\,(TCK–DTDI)} = \frac{\left[t_{d\,(TTDO)} + t_{d\,(DTCKLmax)} + t_{su\,(DTDLmin)}\right]}{t_{(TCKfactor)}}$$

$$= \frac{[15ns + 16ns + 7ns]}{0.4}$$

$$= 9.5ns\ (10.5\ MHz)$$

In this case, the TCK-to-DTMS/DTDL path is the limiting factor.

Case 2: Single/multiprocessor, DTMS/DTDO/TCK buffered input, DTDI buffered output, DTMS/DTDO timed from TCK low.

$$t_{pd\,(TCK–TDMS)} = \frac{\left[t_{d\,(DTMSmax)} + t_{(DTCKHmin)} + t_{su\,(TTMS)} + t_{(bufskew)}\right]}{t_{(TCKfactor)}}$$

$$= \frac{[31ns + 2ns + 10ns + 1.35ns]}{0.4}$$

$$= 110.9ns\ (9.0\ MHz)$$

$$t_{pd\,(TCK–DTDI)} = \frac{\left[t_{d\,(TTDO)} + t_{d\,(DTCKLmax)} + t_{su\,(DTDLmin)} + t_{d\,(bufskew)}\right]}{t_{(TCKfactor)}}$$

$$= \frac{[15ns + 15ns + 7ns + 10ns]}{0.4}$$

$$= 120ns\ (8.3\ MHz)$$

In this case, the TCK-to-DTDI path is the limiting factor.

1.9.3 Using Emulation Pins

The EMU0/1 pins of TI devices are bidirectional, three-state output pins. When in an inactive state, these pins are at high impedance. When the pins are active, they function in one of the two following output modes:

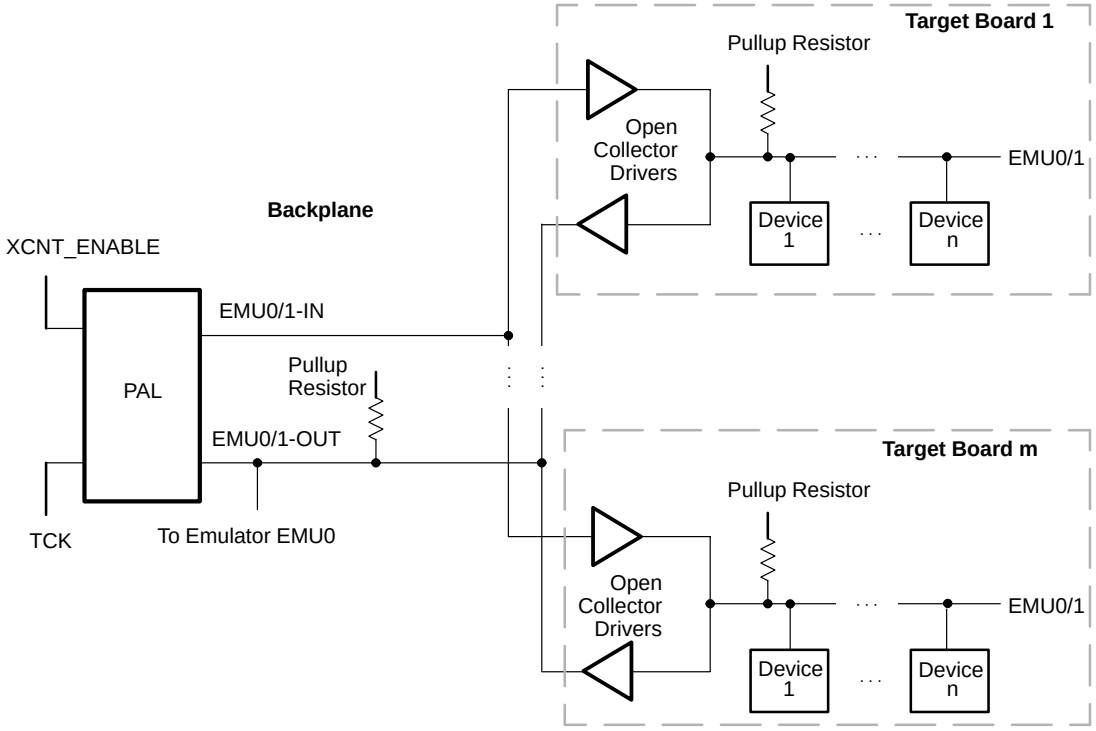
- **Signal Event**
  The EMU0/1 pins can be configured via software to signal internal events. In this mode, driving one of these pins low can cause devices to signal such events. To enable this operation, the EMU0/1 pins function as open-collector sources. External devices such as logic analyzers can also be connected to the EMU0/1 signals in this manner. If such an external source is used, it must also be connected via an open-collector source.

- **External Count**
  The EMU0/1 pins can be configured via software as totem-pole outputs for driving an external counter. If the output of more than one device is configured for totem-pole operation, then these devices can be damaged. The emulation software detects and prevents this condition. However, the emulation software has no control over external sources on the EMU0/1 signal. Therefore, all external sources must be inactive when any device is in the external count mode.

TI devices can be configured by software to halt processing if their EMU0/1 pins are driven low. This feature, in combination with the use of the signal event output mode, allows one TI device to halt all other TI devices on a given event for system-level debugging.

If you route the EMU0/1 signals between boards, they require special handling because these signals are more complex than normal emulation signals. Figure 1–9 shows an example configuration that allows any processor in the system to stop any other processor in the system. Do not tie the EMU0/1 pins of more than 16 processors together in a single group without using buffers. Buffers provide the crisp signals that are required during a RUNB (run benchmark) debugger command or when the external analysis counter feature is used.

Figure 1–9. EMU0/1 Configuration



Notes: 1) The low time on EMUx-IN should be at least one TCK cycle and less than 10 μs. Software will set the EMUx-OUT pin to a high state.

2) To enable the open-collector driver and pullup resistor on EMU1 to provide rising/falling edges of less than 25 ns, the modification shown in this figure is suggested. Rising edges slower than 25 ns can cause the emulator to detect false edges during the RUNB command or when the external counter selected from the debugger analysis menu is used.

These seven important points apply to the circuitry shown in Figure 1–9 and the timing shown in Figure 1–10:

- Open-collector drivers isolate each board. The EMU0/1 pins are tied together on each board.
- At the board edge, the EMU0/1 signals are split to provide IN/OUT. This is required to prevent the open-collector drivers from acting as a latch that can be set only once.
- The EMU0/1 signals are bused down the backplane. Pullup resistors are installed as required.
- The bused EMU0/1 signals go into a PAL® device (see Appendix A), whose function is to generate a low pulse on the EMU0/1-IN signal when

a low level is detected on the EMU0/1-OUT signal. This pulse must be longer than one TCK period to affect the devices, but less than 10 μs to avoid possible conflicts or retriggering, once the emulation software clears the device's pins.

- During a RUNB debugger command or other external analysis count, the EMU0/1 pins on the target device become totem-pole outputs. The EMU1 pin is a ripple carry-out of the internal counter. EMU0 becomes a *processor-halted* signal. During a RUNB or other external analysis count, the EMU0/1-IN signal to all boards must remain in the high (disabled) state. You must provide some type of external input (XCNT_ENABLE) to the PAL to disable the PAL from driving EMU0/1-IN to a low state.
- If sources other than TI processors (such as logic analyzers) are used to drive EMU0/1, their signal lines must be isolated by open-collector drivers and be inactive during RUNB and other external analysis counts.
- You must connect the EMU0/1-OUT signals to the emulation header or directly to a test bus controller.

Figure 1–10. Suggested Timings for the EMU0 and EMU1 Signals

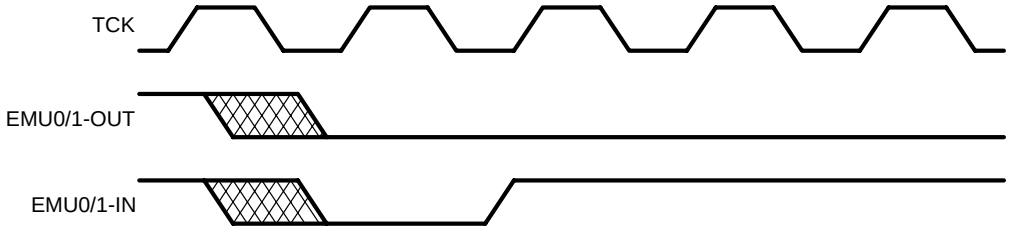
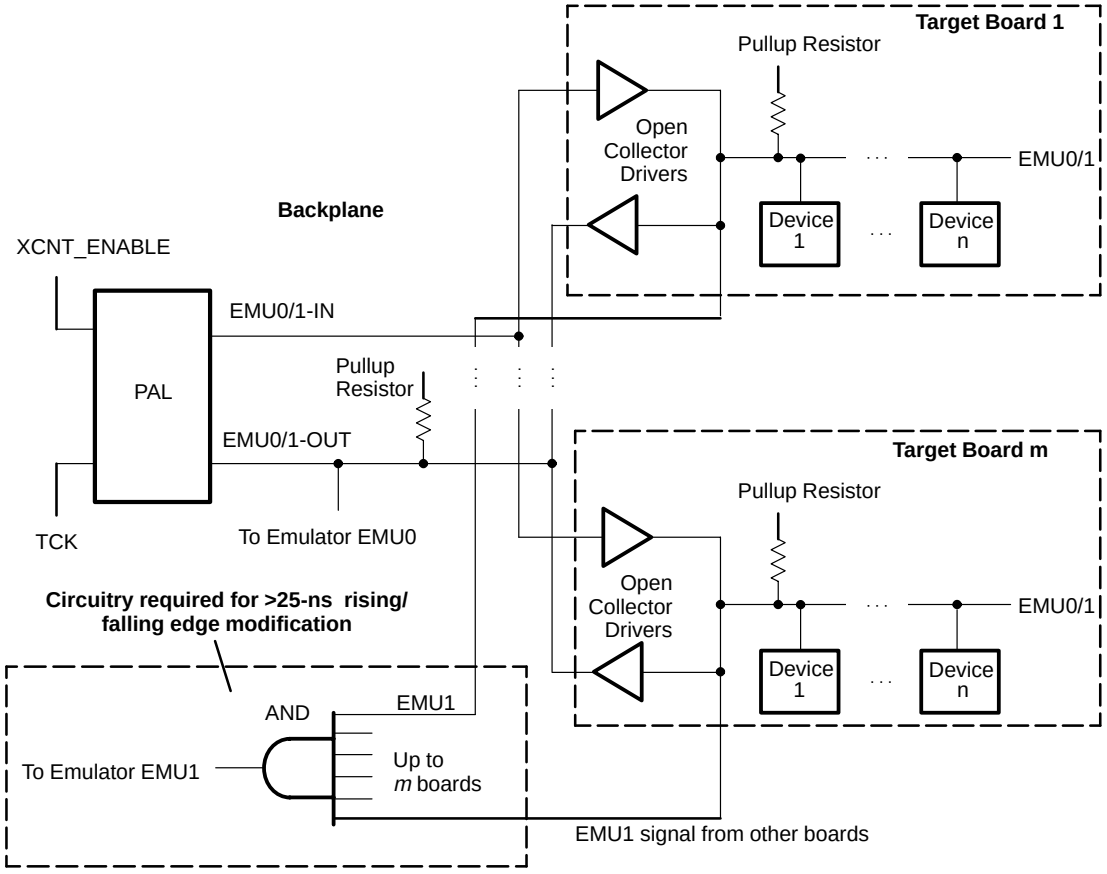


Figure 1–11. EMU0/1 Configuration With Additional AND Gate to Meet Timing Requirements

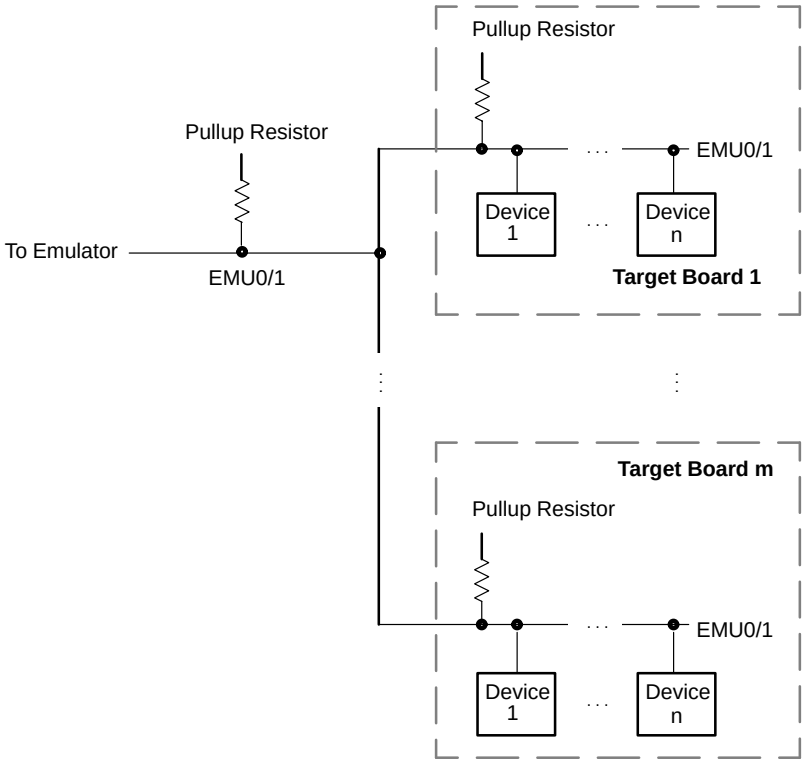


Notes: 1) The low time on EMUx–IN should be at least one TCK cycle and less than 10 μs. Software will set the EMUx–OUT pin to a high state.

2) To enable the open-collector driver and pullup resistor on EMU1 to provide rising/falling edges of less than 25 ns, the modification shown in this figure is suggested. Rising edges slower than 25 ns can cause the emulator to detect false edges during the RUNB command or when the external counter selected from the debugger analysis menu is used.

If having devices on one target board stopped by devices on another target board via the EMU0/1 signals is not important, then the circuit in Figure 1–12 can be used. In this configuration, the global-stop capability is lost. It is important not to overload EMU0/1 with more than 16 devices.

Figure 1–12. EMU0/1 Configuration Without Global Stop

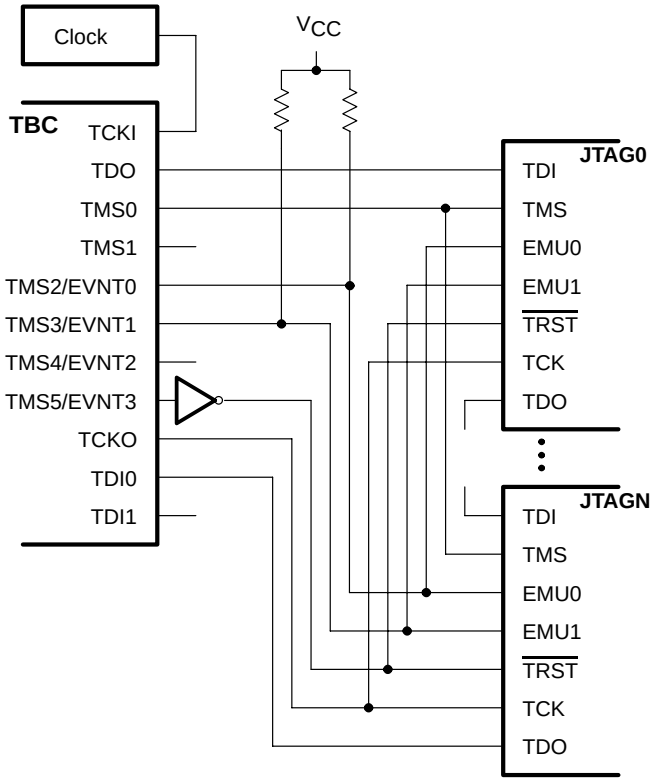


Note: The open-collector driver and pullup resistor on EMU1 must be able to provide rising/falling edges of less than 25 ns. Rising edges slower than 25 ns can cause the emulator to detect false edges during the RUNB command or when the external counter selected from the debugger analysis menu is used. If this condition cannot be met, then the EMU0/1 signals from the individual boards should be ANDed together (as shown in Figure 1-11) to produce an EMU0/1 signal for the emulator.

1.9.4 Performing Diagnostic Applications

For systems that require built-in diagnostics, it is possible to connect the emulation scan path directly to a TI ACT8990 test bus controller (TBC) instead of the emulation header. The TBC is described in the Texas Instruments *Advanced Logic and Bus Interface Logic Data Book* (literature number SCYD001). Figure 1–13 shows the scan path connections of *n* devices to the TBC.

Figure 1–13. TBC Emulation Connections for n JTAG Scan Paths



In the system design shown in Figure 1–13, the TBC emulation signals TCKI, TDO, TMS0, TMS2/EVNT0, TMS3/EVNT1, TMS5/EVNT3, TCKO, and TDI0 are used, and TMS1, TMS4/EVNT2, and TDI1 are not connected. The target devices' EMU0 and EMU1 signals are connected to V_{CC} through pullup resistors and tied to the TBC's TMS2/EVNT0 and TMS3/EVNT1 pins, respectively. The TBC's TCKI pin is connected to a clock generator. The TCK signal for the main JTAG scan path is driven by the TBC's TCKO pin.

On the TBC, the TMS0 pin drives the TMS pins on each device on the main JTAG scan path. TDO on the TBC connects to TDI on the first device on the main JTAG scan path. TDI0 on the TBC is connected to the TDO signal of the last device on the main JTAG scan path. Within the main JTAG scan path, the TDI signal of a device is connected to the TDO signal of the device before it. $\overline{\text{TRST}}$ for the devices can be generated either by inverting the TBC's TMS5/EVNT3 signal for software control or by logic on the board itself.

Chapter 2

Designing for MPSD Emulation

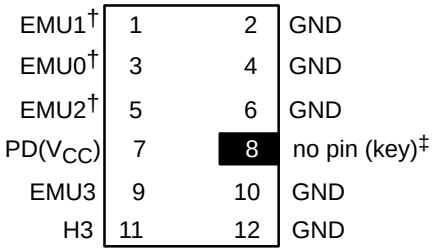
This chapter assists you in meeting the design requirements of the XDS510 emulator with respect to TMS320C3x modular port scan device (MPSD) designs and describes the XDS510 cable (manufacturing part number 2617699-0001). This cable is identified by a label on the cable pod marked **MPSD 3/5V** and supports both standard 3-volt and 5-volt target system power inputs.

Topic **Page**

2.1 Designing Your MPSD Emulator Connector (12-Pin Header)

The ’C3x uses a modular port scan device (MPSD) technology to allow complete emulation via a serial scan path of the ’C3x. To communicate with the emulator, **your target system must have a 12-pin header** (two rows of six pins) with the connections that are shown in Figure 2–1. Pin 8 is cut out to provide keying. Table 2-1 describes the emulation signals and lists the ’C3x pin connections.

Figure 2–1. 12-Pin Header Signals and Header Dimensions



Header Dimensions:
Pin-to-pin spacing, 0.100 in. (X,Y)
Pin width, 0.025-in. square post
Pin length, 0.235-in. nominal

† These signals should always be pulled up to V_{CC} with separate 20-kΩ resistors.
‡ While the corresponding female position on the cable connector is plugged to prevent improper connection, the cable lead for pin 8 is present in the cable and is grounded, as shown in the schematics and wiring diagrams in this document.

Table 2–1. 12-Pin Header Signal Descriptions and ’C3x Pin Connections

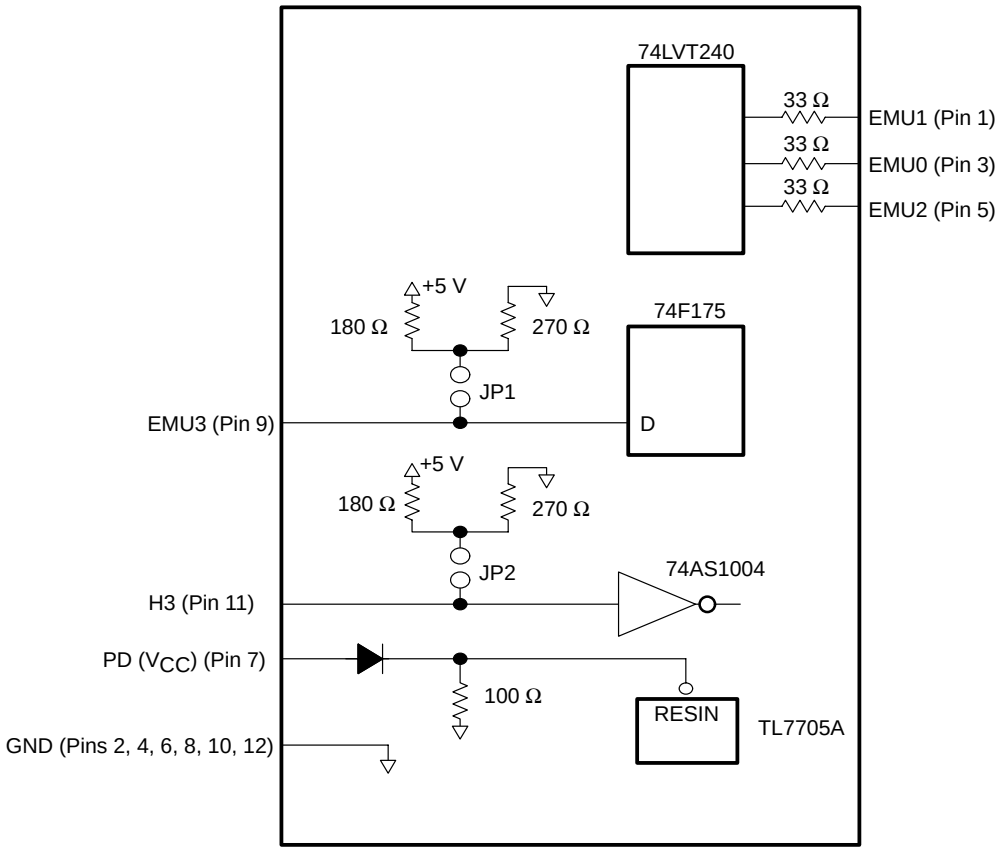
XDS510 Signal	Description	’C30 Pin Number	’C31 Pin Number
EMU0	Emulation pin 0	F14	124
EMU1	Emulation pin 1	E15	125
EMU2	Emulation pin 2	F13	126
EMU3	Emulation pin 3	E14	123
H3	’C3x H3	A1	82
PD(V_{CC})	Presence detect. Indicates that the emulation cable is connected and that the target is powered up. PD should be tied to V_{CC} in the target system.		

Although you can use other headers, recommended parts include:

straight header, unshrouded — DuPont Connector Systems
part numbers: 65610–112
65611–112
37996–112
67997–112

Figure 2–2 shows a portion of logic in the emulator cable pod. Note that 33-Ω resistors have been added to the EMU0, EMU1, and EMU2 lines; this minimizes cable reflections.

Figure 2–2. MPSD Emulator Cable Pod Interface



2.2 MPSD Emulator Cable Pod Signal Timing

Figure 2–3 shows the signal timings for the emulator pod. Table 2–2 defines the timing parameters. The timing parameters are calculated from values specified in the standard data sheets for the emulator and cable pod and are for reference only. Texas Instruments does not test or guarantee these timings.

Figure 2–3. MPSD Emulator Cable Pod Timings

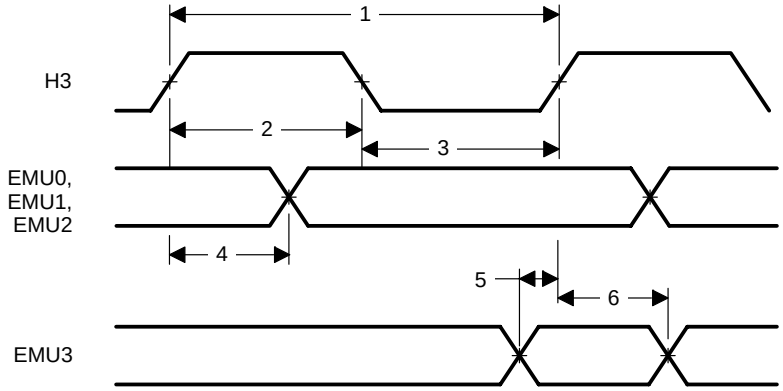


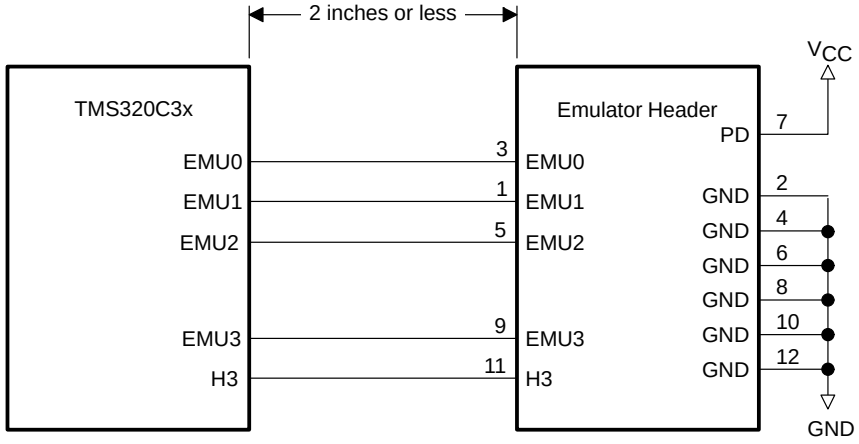
Table 2–2. Emulator Cable Pod Timing Parameters

No.	Reference	Description	Min	Max	Unit
1	$t_{c(H3)}$	H3 period	35	200	ns
2	$t_{c(H3H)}$	H3 high-pulse duration	15		ns
3	$t_{c(H3L)}$	H3 low-pulse duration	15		ns
4	$t_{d(EMU0, 1, 2)}$	Delay time, EMU0, 1, 2 valid from H3 high	7	23	ns
5	$t_{su(EMU3)}$	EMU3 setup time to H3 high	3		ns
6	$t_{h(EMU3)}$	EMU3 hold time from H3 high	11		ns

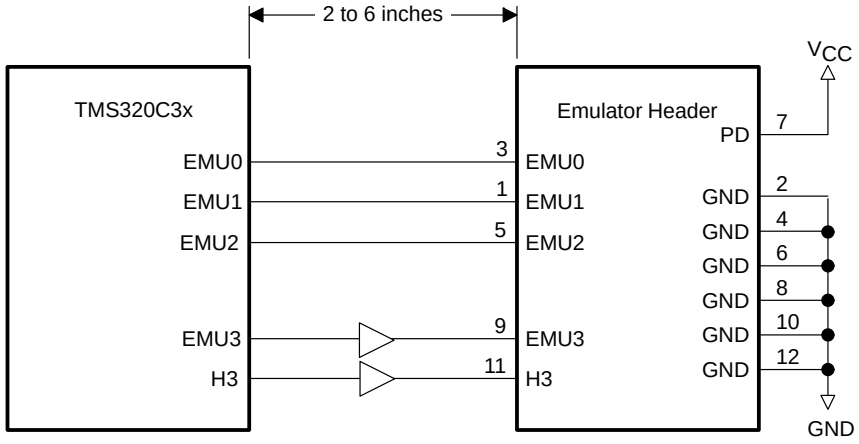
2.3 Connections Between the MPSD Emulator and the Target System

It is extremely important to provide high-quality signals between the MPSD emulator and the ’C3x on the target system. To do so, the signal must be buffered in many cases. The distance of the emulation header from the ’C3x determines the need for signal buffering requirements with respect to three categories:

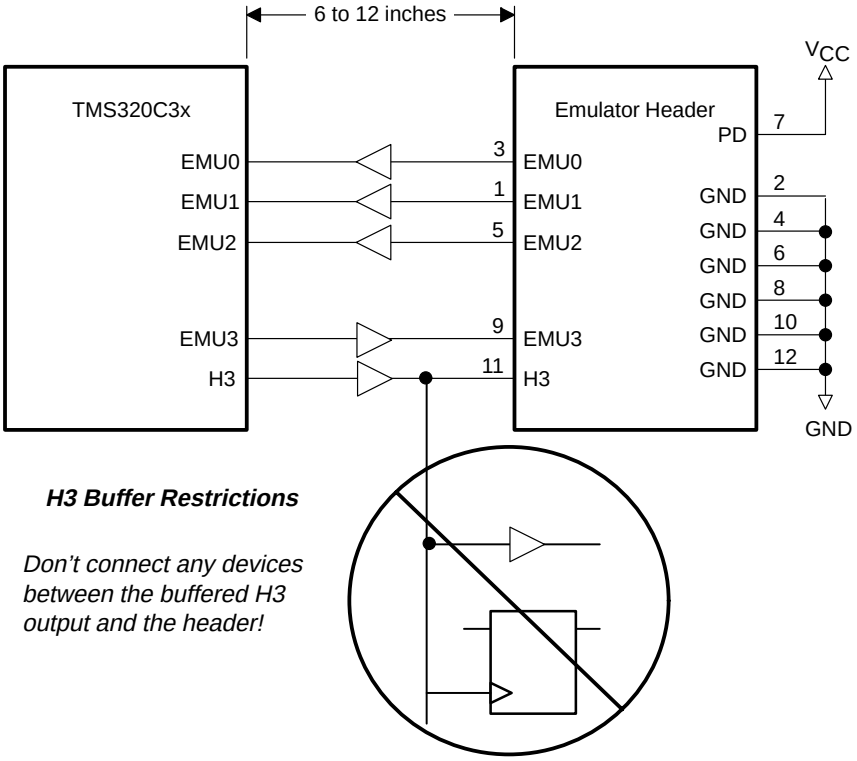
- **No signal buffering.** In this situation, the distance between the header and the ’C3x should be no more than two inches.



- **Buffered transmission signals.** In this situation, the distance between the emulation header and the ’C3x is greater than two inches but less than six inches. The transmission signals—H3 and EMU3—are buffered through the same package.



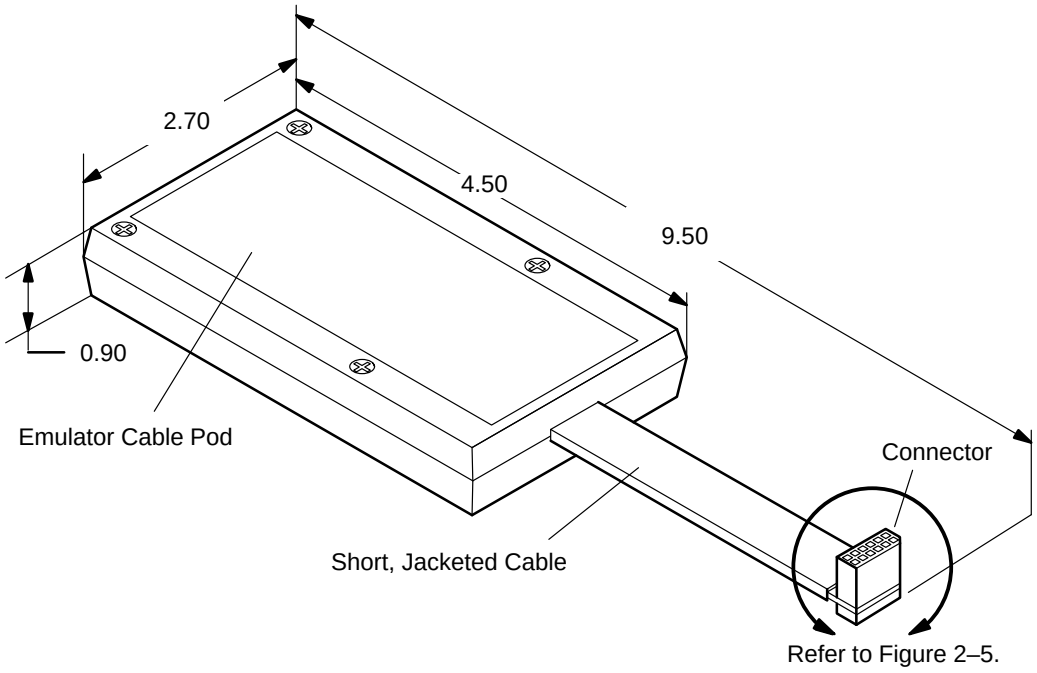
- **All signals buffered.** The distance between the emulation header and the ’C3x is greater than 6 inches but less than 12 inches. All ’C3x emulation signals—EMU0, EMU1, EMU2, EMU3, and H3—are buffered through the same package.



2.4 Mechanical Dimensions for the 12-Pin MPSD Emulator Connector

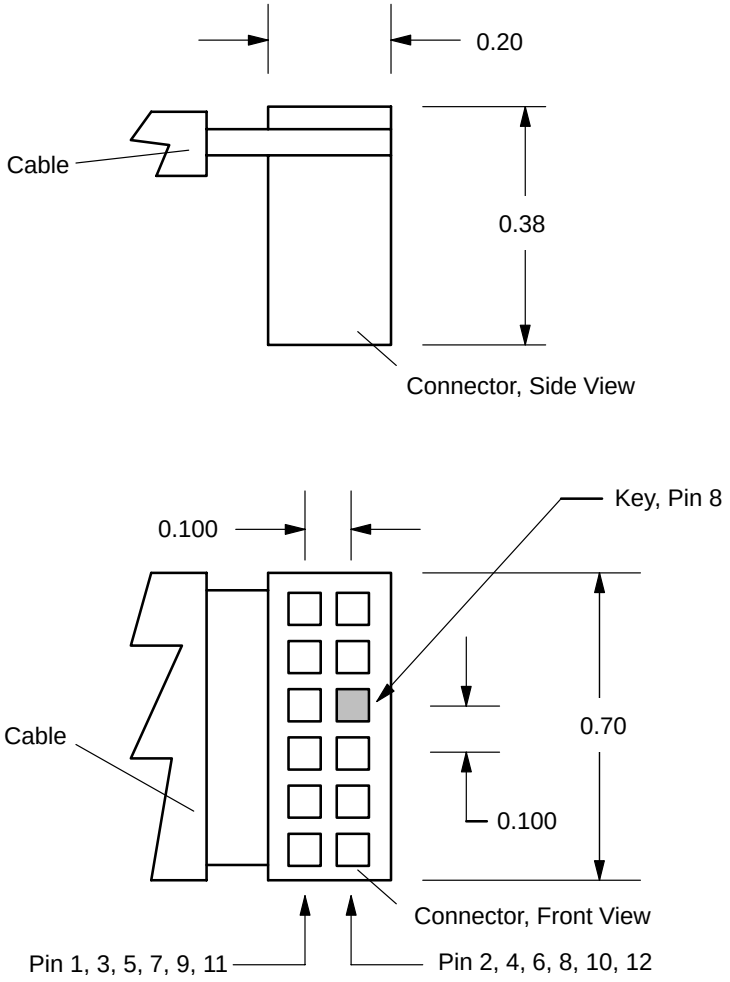
The ’C3x MPSD emulator target cable consists of a 3-foot section of jacketed cable, an active cable pod, and a short section of jacketed cable that connects to the target system. The overall cable length is approximately 3 feet 10 inches. Figure 2–4 and Figure 2–5 show the mechanical dimensions for the target cable pod and short cable. Note that the pin-to-pin spacing on the connector is 0.100 inches in both the X and Y planes. The cable pod box is nonconductive plastic with four recessed metal screws.

Figure 2–4. MPSD Pod/Connector Dimensions



Note: All dimensions are in inches and are nominal dimensions, unless otherwise specified.

Figure 2–5. 12-Pin Connector Dimensions

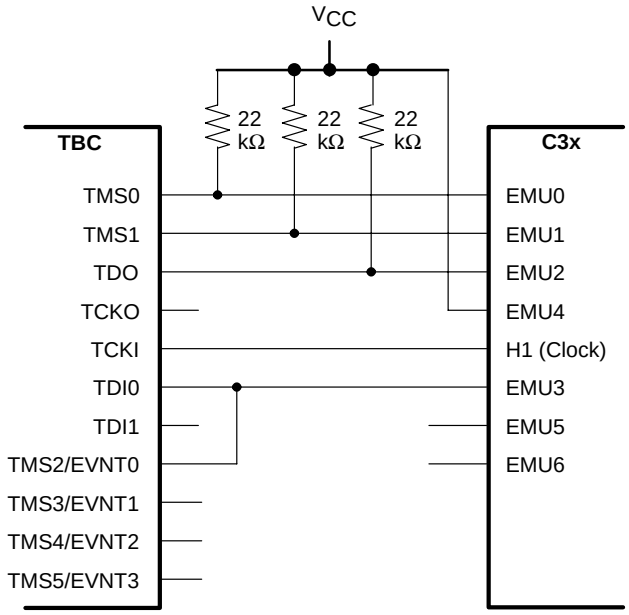


Note: All dimensions are in inches and are nominal dimensions, unless otherwise specified.

2.5 Diagnostic Applications

For system diagnostics applications or embedding emulation compatibility on your target system, you can connect a ’C3x device directly to a TI ACT8990 test bus controller (TBC), as shown in Figure 2–6. The TBC is described in the Texas Instruments *Advanced Logic and Bus Interface Logic Data Book* (literature number SCYD001). A TBC can connect to only one ’C3x device.

Figure 2–6. TBC Emulation Connections for ’C3x Scan Paths



Notes: 1) In a ’C3x design, the TBC can connect to only one ’C3x device.

2) The ’C3x device’s H1 clock drives TCKI on the TBC. This is different from the emulation header connections in which H3 is used.

Appendix A

PAL Programming

This appendix contains the programmable logic source for the PAL from Figure 1–9 on page 1–19 and Figure 1–11 on page 1–21. This PAL equation was reduced with DATA I/O ABEL™ version 3.2 at a reduction level of 3.

```
module EMU_backplane
title'
DATE       09/01/94'

  U0                device              'pl6r4'
  "inputs
  oe_               Pin 11;                  "outut enable, tied to GROUND
  clk               Pin  1:                  "IEEE 1149.1 TCLK
  EMU0_OUT          Pin  2:                  "active low EMU0 from daughters
  EMU1_OUT          Pin  3:                  "active low EMU1 from daughters
  XCNT_ENBL         Pin  4;                  "active high External Count enable
  "EXT_EMU          Pin  5;                  "active low external trigger for EMU

  "power
  gnd               Pin 10;                  "GROUND
  vcc               Pin 20;                  "POWER

  "outputs
  EMU0_IN           Pin 17;                  "active low EM0 to daughters
  EMU1_IN           Pin 16;                  "active low EMU1 to daughters
  S0                Pin 15;                  "state for EMU0_IN
  S1                Pin 14;                  "state for EMU1_IN

"aliases for ease of use
  c                 = .C.;                   "input : clock edge falling
  X                 = .X.;                   "unknown

"states
  idle              = ^b11                   "waiting for EMU active
  count1            = ^b10                   "sending EMU active for first TCLK
  count2            = ^b00                   "sending EMU active for second TCLK
  wait              = ^b01                   "waiting for EMU inactive again

state_diagram       [S0, EMU0_IN]
         state        idle:
                       IF ( !EMU0_OUT & !XCNT_ENBL ) THEN     count1
                       IF ( !EXT_EMU & !XCNT_ENBL )  THEN     count1
                                  ELSE                        idle;

         state        count1:
                           GOTO                               count2;

         state        count2:
                           GOTO                               wait;

         state        wait:
                           IF(!EMU0_OUT & !XCNT_ENBL) THEN    wait
                           IF(!EXT_EMU & !XCNT_ENBL)  THEN    wait

state_diagram     [S0, EMU0_IN]
                       ELSE                                   idle;

         state        idle:
                       IF (!EMU1_OUT & !XCNT_ENBL )   THEN    count1
                       IF ( !EXT_EMU & !XCNT_ENBL )   THEN    count1
                       ELSE                                   idle;
```

(Continued on page A-3)

```
        state       count1:
                        GOTO                                count2;

        state       count2:
                        GOTO                                wait;

        state       wait:
                        IF(!EMU0_OUT & !XCNT_ENBL) THEN     wait
                        IF(!EXT_EMU & !XCNT_ENBL)  THEN     wait
                        ELSE                                idle;
test_vectors"1-14
"test EMU0_IN and SO
(  [oe_ ,clk ,EMU0_OUT,XCNT_ENBL ]->[SO ,EMU0_IN ])
   [ 0  , c  ,   1    ,   0      ]->[ X ,  X     ];"reset to known state:idle
   [ 0  , c  ,   1    ,   0      ]->[ X ,  X     ];"reset to known state:idle
   [ 0  , c  ,   1    ,   0      ]->[ 1 ,  1     ];"reset to known state:idle
   [ 0  , c  ,   1    ,   0      ]->[ 1 ,  1     ];"EMU input inactive, stay in idle
   [ 0  , c  ,   0    ,   1      ]->[ 1 ,  1     ];"EMU input and XCNT active, idle
   [ 0  , c  ,   0    ,   0      ]->[ 1 ,  0     ];"EMU input active, goto count1
   [ 0  , c  ,   0    ,   1      ]->[ 0 ,  0     ];"any EMU input, goto count2
   [ 0  , c  ,   0    ,   1      ]->[ 0 ,  1     ];"any EMU input, goto wait
   [ 0  , c  ,   0    ,   0      ]->[ 0 ,  1     ];"EMU input active, stay in wait
   [ 0  , c  ,   1    ,   0      ]->[ 1 ,  1     ];"EMU input inactive, goto idle
   [ 0  , c  ,   0    ,   0      ]->[ 1 ,  0     ];"EMU input active, goto count1
   [ 0  , c  ,   0    ,   1      ]->[ 0 ,  0     ];"any EMU input, goto count2
   [ 0  , c  ,   0    ,   1      ]->[ 0 ,  1     ];"any EMU input, goto wait
   [ 0  , c  ,   0    ,   1      ]->[ 1 ,  1     ];"EMU input and XCNT inactive, idle
"possible additional tests for EXT_EMU

test_vectors"15-28
"test EMU1_IN and S1
(  [oe_ ,clk ,EMU1_OUT,XCNT_ENBL ]->[SO ,EMU1_IN ])
   [ 0  , c  ,   0    ,   1      ]->[ X ,  X     ];"reset to known state:idle
   [ 0  , c  ,   0    ,   1      ]->[ X ,  X     ];"reset to known state:idle
   [ 0  , c  ,   0    ,   1      ]->[ 1 ,  1     ];"reset to known state:idle
   [ 0  , c  ,   1    ,   1      ]->[ 1 ,  1     ];"EMU input inactive, stay in idle
   [ 0  , c  ,   0    ,   1      ]->[ 1 ,  1     ];"EMU input and XCNT active, idle
   [ 0  , c  ,   0    ,   0      ]->[ 1 ,  0     ];"EMU input active, goto count1
   [ 0  , c  ,   1    ,   0      ]->[ 0 ,  0     ];"any EMU input, goto count2
   [ 0  , c  ,   1    ,   0      ]->[ 0 ,  1     ];"any EMU input, goto wait
   [ 0  , c  ,   0    ,   0      ]->[ 0 ,  1     ];"EMU input active, stay in wait
   [ 0  , c  ,   1    ,   1      ]->[ 1 ,  1     ];"EMU input inactive, goto idle
   [ 0  , c  ,   0    ,   0      ]->[ 1 ,  0     ];"EMU input active, goto count1
   [ 0  , c  ,   1    ,   1      ]->[ 0 ,  0     ];"any EMU input, goto count2
   [ 0  , c  ,   1    ,   1      ]->[ 0 ,  1     ];"any EMU input, goto wait
   [ 0  , c  ,   0    ,   1      ]->[ 1 ,  1     ];"EMU input and XCNT inactive, idle
"possible additional tests for EXT_EMU
end EMU_backplane
```

Index

I

J

M

P

R

S

T

X

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.