Application Report

SPRAAK6–May 2007

Common Trace Transmission Problems and Solutions

Ronald Lerner

ABSTRACT

This document provides guidelines for identifying and solving common problems associated with the collection of high-speed data. On a trace-capable device, the trace interface is one of the highest performance interfaces. Although only used during design, development, and debug, the trace interface must be implemented correctly to achieve full functionality and optimal performance.

Contents

List of Figures

List of Tables

Trademarks

XDS560, Code Composer Studio are trademarks of Texas Instruments.

Windows is a registered trademark of Microsoft Corporation in the United States and/or other countries.

1 Introduction

This section includes background information and usage notes.

1.1 Background

Trace is a technology that gives you a detailed, historical account of application code execution on a trace-enabled device. On-chip debug technology enables this using capabilities that filter, collect, compress and export trace information for analysis. Unless explicitly configured by the consumer, all of this is done in real-time and without impacting the execution of the system. Figure 1 shows the parts involved in trace.

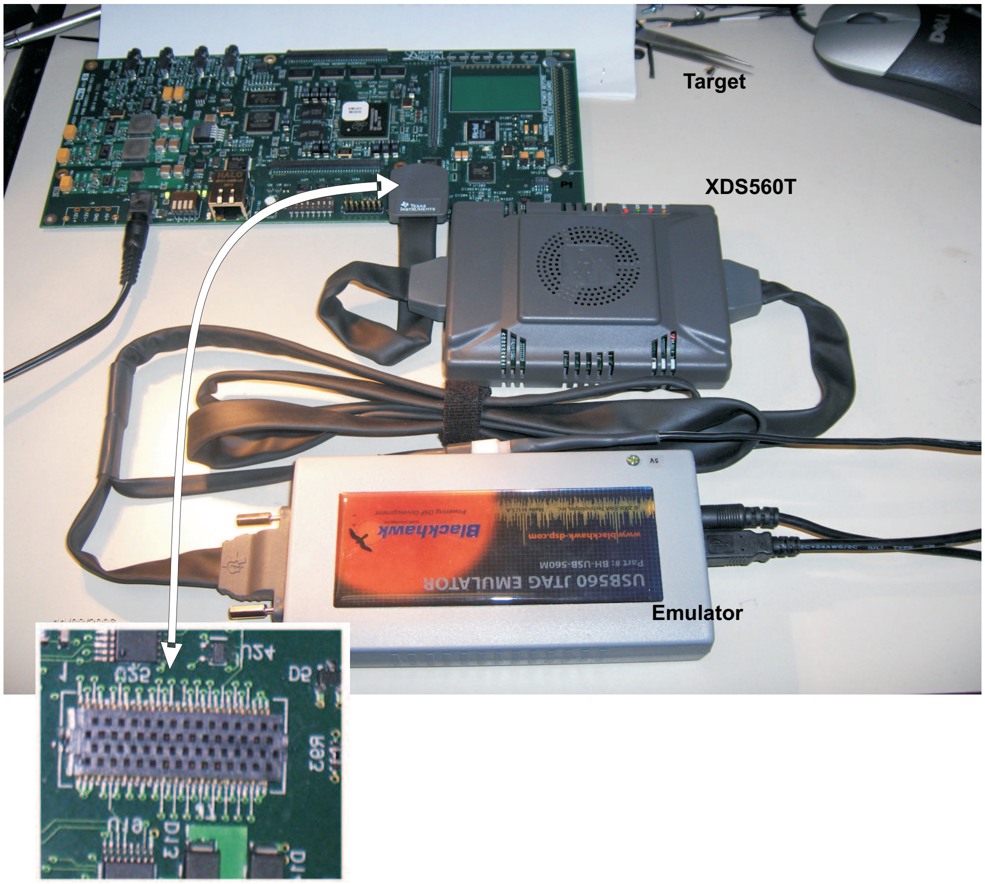

Figure 1. Missing Title

Despite the use of compression techniques, an enormous quantity of data must be sent off-chip and collected by Texas Instruments' XDS560™ Trace pod software. Data is exported through the device I/Os and ultimately to the trace receiver in the XDS560T trace pod. The system (DSP, target board with 60-pin header, and trace pod) must be designed correctly and must be capable of handling the exported double data-transfer rates of 167 MHz (333 Mb/pS/per pin) inclusive of the 8-11 inches of cabling in the trace pod. With the XDS560 trace pod, significant attention has been paid to provide the highest quality trace signal between the trace receiver and the 60-pin connector. The same amount of attention must be given to the design of the target board to ensure that trace signal integrity is not compromised between the trace-enabled device and the 60-pin header. Section 2 covers in more detail the concerns to be taken into account when designing a target board that supports trace export.

1.2 Usage Notes

There are a few items of which you should be aware before bringing up Code Composer Studio™ -- even before plugging the XDS560 trace pod into the 60-pin header. All of this information and troubleshooting tips appear in Section 5.

2 Board Design

Improper board design can impact the signal quality of a high-speed bus like trace and even a lower-speed bus like JTAG by introducing phenomena such as ringing, reflections, cross-talk, and voltage drop. Techniques that can minimize these phenomena are discussed below.

Note: On-chip trace can generate a predefined pattern for export out of the device. These patterns are used by the trace software to calibrate the XDS560 trace pod software upon starting a trace session. If this calibration fails, it is likely to be an issue with the trace signal integrity on the target board; however, even if the calibration passes, it is possible that a signal integrity issue still exists. It is imperative that the steps described in Section 2.1 be followed to characterize the EMU and TCK signal quality.

2.1 Checklist for Board Design

To characterize the EMU and TCK signal quality, use the following check list:

Table 1. Checklist for Board Design

√	No.	Items to Check
	1	If the target board being used or if the design of your new target board is based on the TMS320C6455 DSP Starter Kit (DSK), see Section 3 for known issues and recommendations.
	2	If the target board being used or if the design of your new target board is based on the TMS320TCI6482 DSK, see Section 3 for known issues and recommendations.
	3	Validate signal quality of TCK as described in Section 2.4.1. Check for stubs and proper ac termination.
		a TCK will be running after connecting to the target via Code Composer Studio.
	4	Resolve any signal quality issues with TCK.
	5	Validate signal quality of the EMU signals as described in Section 2.4.2. Check for stubs and proper series termination.
		a EMU signal activity occurs during trace calibration. This can be used as a means of generating EMU signal activity for analysis. Using the *Tools* → *Trace* → *Control* window in the Code Composer Studio, calibration occurs if you change the number of pins or *Export clock rate* and click *Apply*.
	6	Resolve any signal quality issues with the EMU signals.

2.2 Transmission Lines

A transmission line has five fundamental parts:

- Driver
- Driver Output Impedance ($Z_{O(driver)}$)
- Series Termination, Impedance Matching Resistor
- Load Impedance (Z_L)
- Receiver(s)

Two transmission line examples are discussed: point-to-point and multiple-destinations. In both examples it is important to note that trace stubs should be minimized if absolutely required, or eliminated if not.

Figure 2 is an example of a point-to-point transmission line. The defining characteristic of a point-to-point transmission line is the single driver and a single receiver. All EMU signals require a point-to-point transmission line. See Section 2.4.2 for details about the series termination.

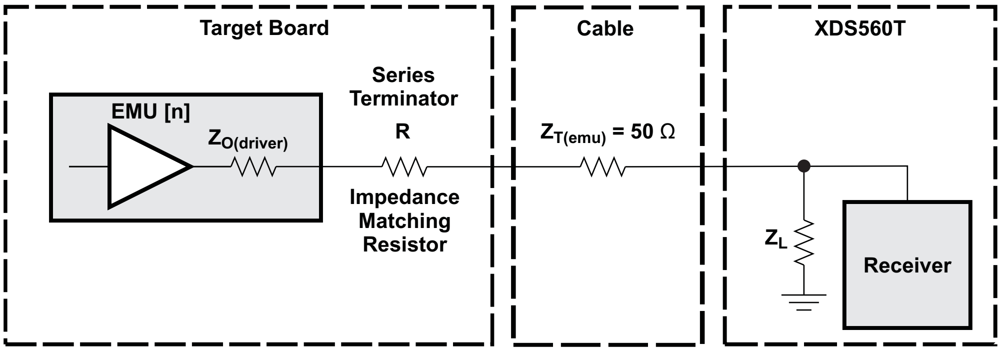

Figure 2. Point-To-Point Transmission Line

In some situations a single driver is required to drive multiple destinations. As illustrated in Figure 3, an example of such a case is the JTAG TCK driver in a system with multiple devices in the JTAG scan chain. There are two important items to note when doing this:

- First, stub lengths should be minimized. A stub encounters an impedance mismatch at the high-impedance input of the receiver, causing reflections (see Section 2.3).
- Second, *the termination should be placed closest to the last receiver on the transmission line*.

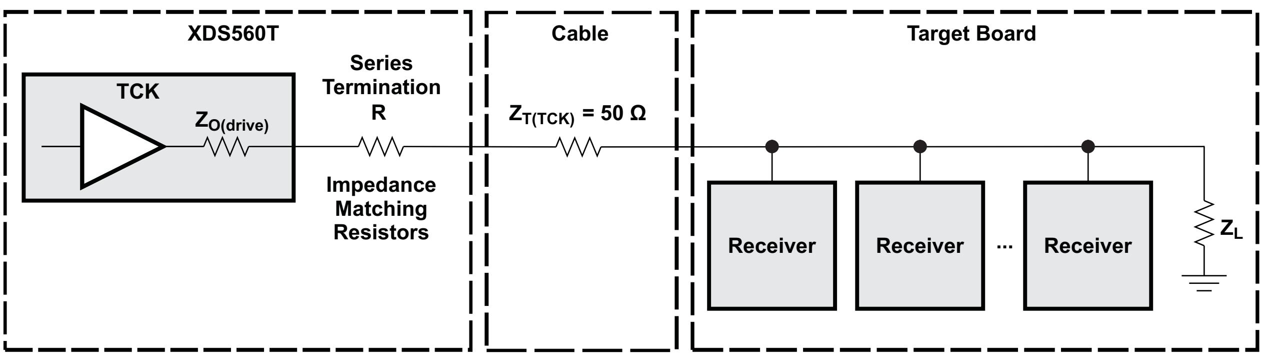

Figure 3. Transmission Line With Multiple Receivers

2.3 Impedance Matching

When an electrical signal passes from one medium (such as a PCB board) to another (such as the cabling on the XDS560 trace pod) any change in impedance results in a reflection back to the source. The loss of energy resulting from a reflection causes attenuation of the signal as it proceeds to its destination. The reflection itself can result in ringing and jitter as it collides with other signals being generated by the source.

When connecting to an XDS560 trace pod, the impedance that must be matched by the target board is 50 Ω. The same 50-Ω impedance applies to XDS560 pods. For the target board, three components should be considered when looking at impedance matching:

- Output impedance of the I/O buffer
- Impedance of the signal termination
- Impedance of the PCB trace

2.3.1 TI C6000 Output Impedances

The output impedance on the I/Os on all TI devices is given in the IBIS model made available through TI's website (http://www.ti.com) [6]. For all existing C6x trace-enabled devices, the output impedance is 25 Ω. This means that an additional 25 Ω is needed on the EMU signals before they can be safely transmitted off the PCB. The recommended means of accomplishing this is discussed in Section 2.4.

Table 2. Nominal Output Impedances for Debug Signals on TI Devices

Device	Nominal Output Impedance of EMU Pins ($Z_{o[emu]}$)
TMS320C6414/15/16	25 Ω
TMS320C6416T	25 Ω
TMS320DM642	25 Ω
TMS320C6455	25 Ω
TMS320TCI6482	25 Ω

2.3.2 PCB Trace Impedance

For maximum signal integrity, the EMU signal traces on the PCB must be designed for 50-Ω impedance. All connectors and interface devices are also 50-Ω impedance.

2.4 Termination Techniques

A proper termination is critical to the performance of digital signals. Termination techniques available for designers include: Parallel, Thevinin, Series, and ac.[3] TI recommends the use of a Series termination on EMU signals and an ac termination on TCK. How these terminations are determined and analyzed with an oscilloscope is described in the subsections that follow.

2.4.1 TCK AC Termination

An ac termination includes a resistor (R) and a capacitor (C). The ac terminations are frequently used to terminate a clock input and should always be placed as close as possible to the receiver. In situations where there are multiple receivers on TCK (i.e., multiple devices in the same JTAG scan-chain), the ac termination should be placed as close to the last receiver as possible (see Figure 4).

The resistor value (R) should match the impedance of the transmission line ($Z_{T(tck)}$). Choosing a capacitor value is more complicated and an incorrect value can result in ringing. The capacitor value should be carefully calculated based on the board design and requirements. Design equations are provided below to help aid determining the value of the capacitor to use [3].

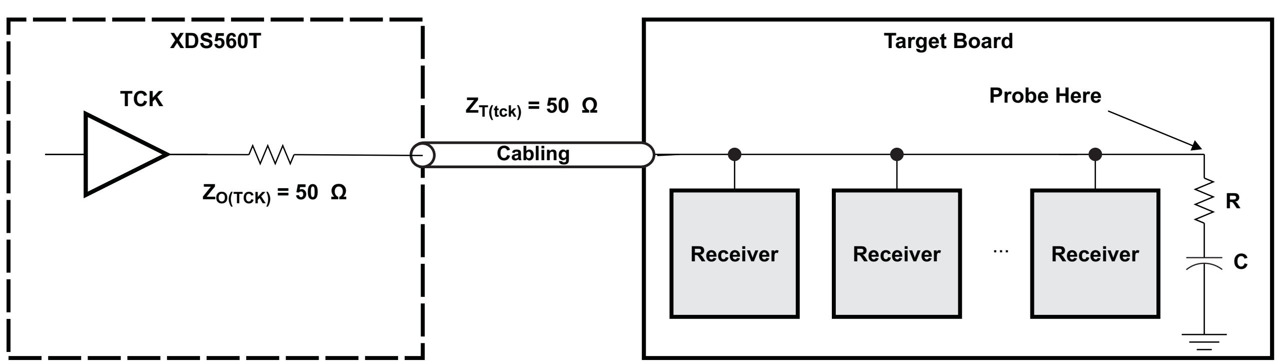

Figure 4. TCK AC Termination

Equation 1. Calculating R Value in TCK AC Termination

$$R = Z_{T(tck)} = 50\ \Omega$$

Equation 2. Calculating C Value in TCK AC Termination

$$C = \tau / (2.2 * Z_o)$$

C = Capacitor value

τ = Rise time of TCK

$Z_{T(tck)}$ = Transmission line impedance

2.4.1.1 Analyzing the TCK Signal With an Oscilloscope

When analyzing the TCK signal, a high-impedance, low capacitance oscilloscope probe should be used with a minimum bandwidth of 1 GHz. The oscilloscope should be rated for a minimum of 1 GHz to provide adequate bandwidth for your analysis. Waveforms should be analyzed at the ac termination, as highlighted in Figure 5. Figure 5 and Figure 6 represent properly and improperly ac terminated signals, respectively.

Note: Waveforms collected within this document were captured using a Tektronix TDS7245B 2.5 GHz oscilloscope and P7225 probes.

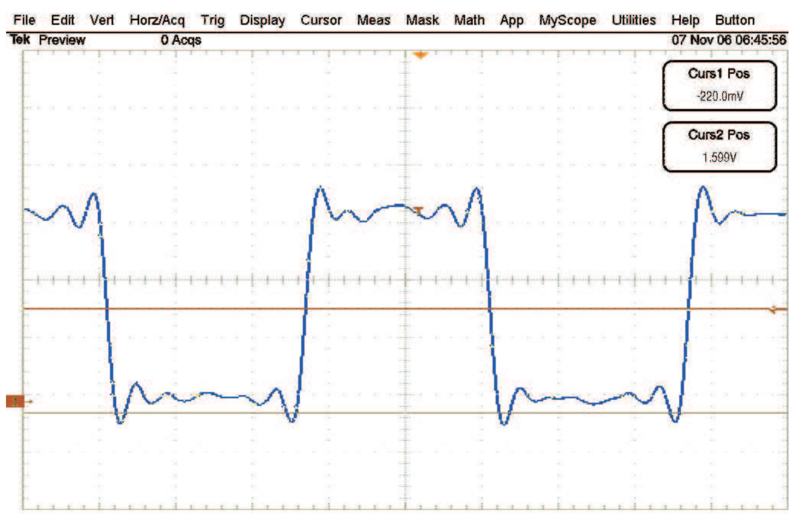

Figure 5. Waveform of a Properly AC Terminated Signal

Figure 6. Waveform of an Improperly AC Terminated Signal

To obtain the proper termination values you must select the proper resistor and capacitor values. Depending on the target design, traditional resistor-capacitor values are 51.1 Ω and a 3.3 – 15 pF capacitor. If the target impedance varies from 50 Ω (not recommended), the component values may need adjusting to provide the best results.

2.4.2 EMU Series Termination

The series termination, also known as a *source termination*, is a series resistor applied at the source end of a signal. A series termination should only be used on a point-to-point transmission lines, all EMU signals should be routed point-to-point. The series termination resistor should be chosen so the value of the resistor (R) and the output impedance of the EMU pin I/O buffers ($Z_{O[emu]}$) equal the impedance of the transmission line ($Z_{T[emu]}$). Figure 7 illustrates the probe point when analyzing a series termination line.

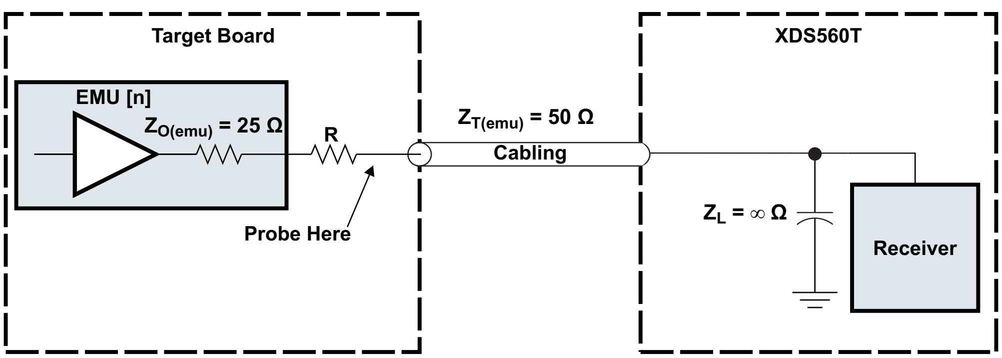

Figure 7. EMU Series Termination

As explained in Section 2.2, the values of $Z_{O[emu]}$ and $Z_{T[emu]}$ are known.

Equation 3. Calculating R Value in EMU Series Termination

$$R = Z_{T(emu)} - Z_{O(EMU)} = 50\ \Omega - 25\ \Omega = 25\ \Omega$$

It is recommended that all EMU signals be series-terminated with a 25-Ω resistor adjacent to the driver pin.

2.4.2.1 Analyzing EMU Signals With an Oscilloscope

When analyzing the EMU signals a high-impedance, low capacitance oscilloscope probe should be used. Waveforms should be analyzed at the receiver side of the series termination, as highlighted in Figure 7. Figure 8 and Figure 9 have been generated using this as an observation point and assuming a 3.3-V I/O.

Figure 8 illustrates the rising edge of an EMU signal with no impedance mismatch. This is the desired waveform. The ledge at the 50% point is caused by an expected reflection from the high-impedance termination at the end of the XDS560 trace pod cable. A t1 of 4 ns represents the amount of time the EMU signal takes to leave the probe position, arrive at the high-impedance termination, and return as a reflection.

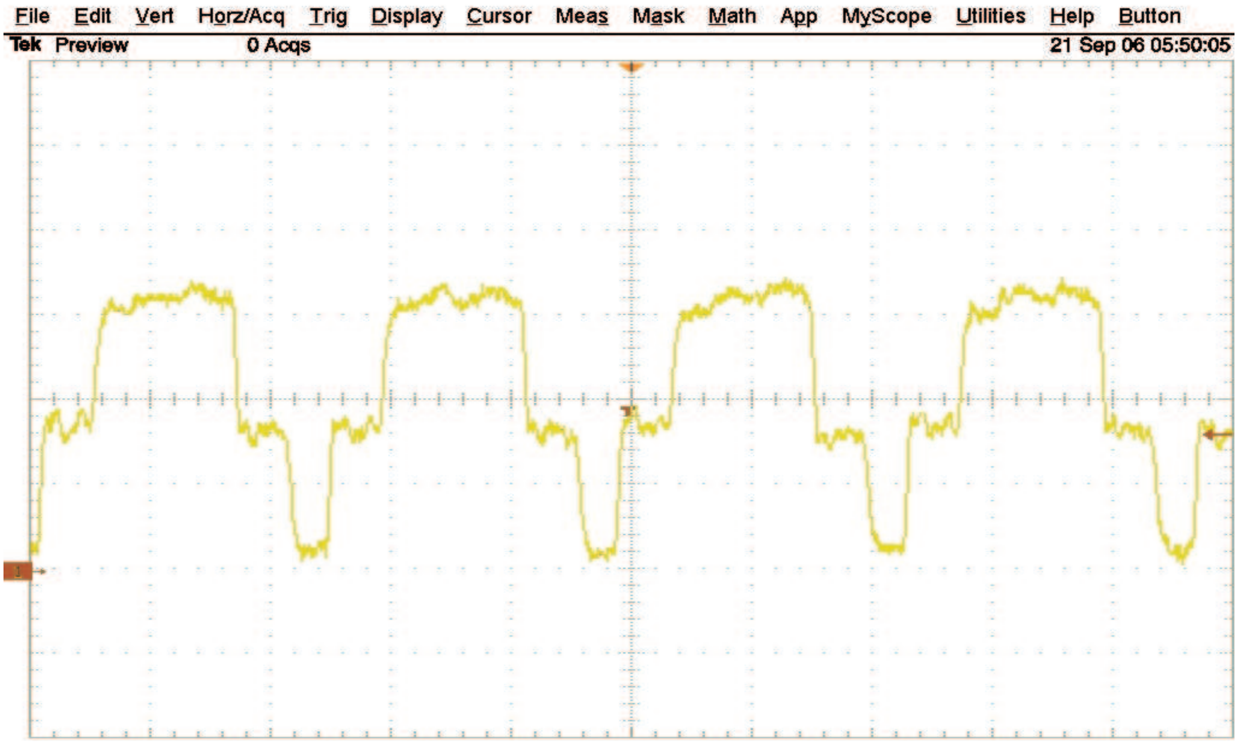

Figure 8. EMU Signal at Probe Point With Matched Impedance

Figure 9 is an illustration of the rising edge of an EMU signal with an impedance mismatch. The output-impedance of the driver plus the series termination is greater than the impedance of the transmission line. In other words, $Z_{O[emu]} + R > Z_{T[emu]}$. In this case, a smaller series termination resistor should be chosen.

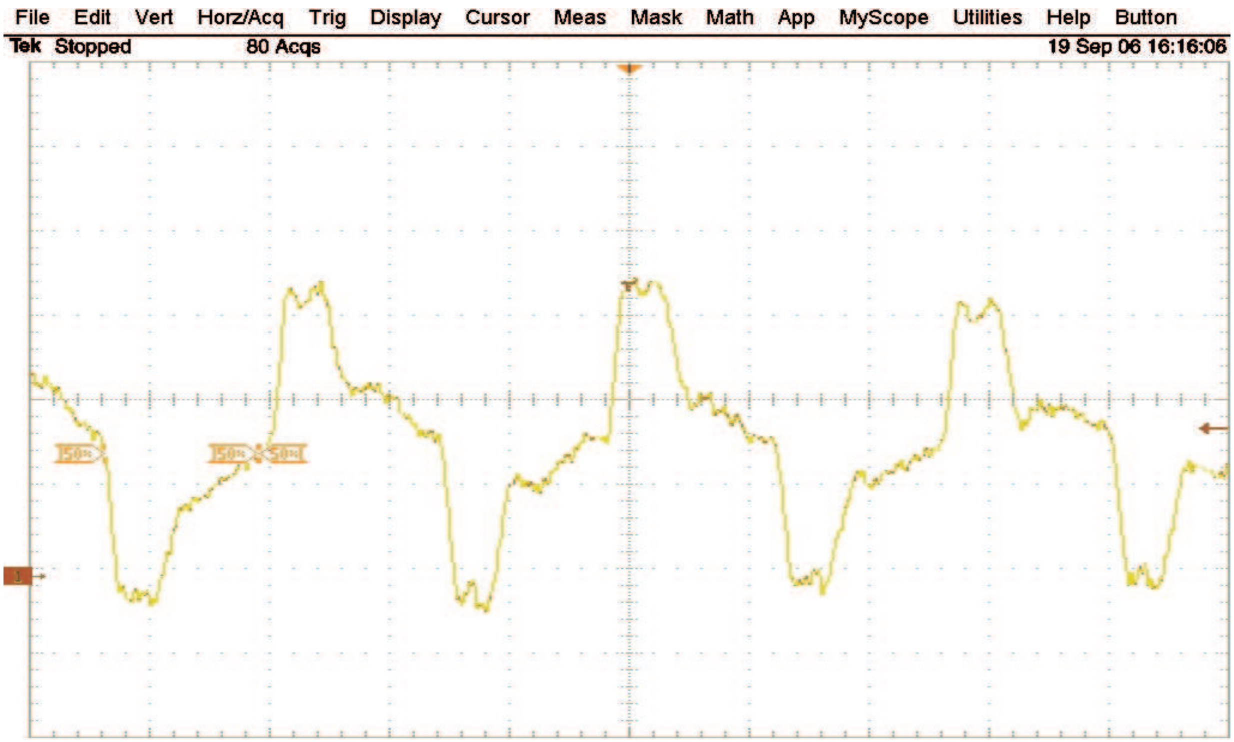

Figure 9. EMU Signal at Probe Point With Impedance Mismatch

Figure 10 is another illustration of the rising and falling edge of an EMU signal with an impedance mismatch.

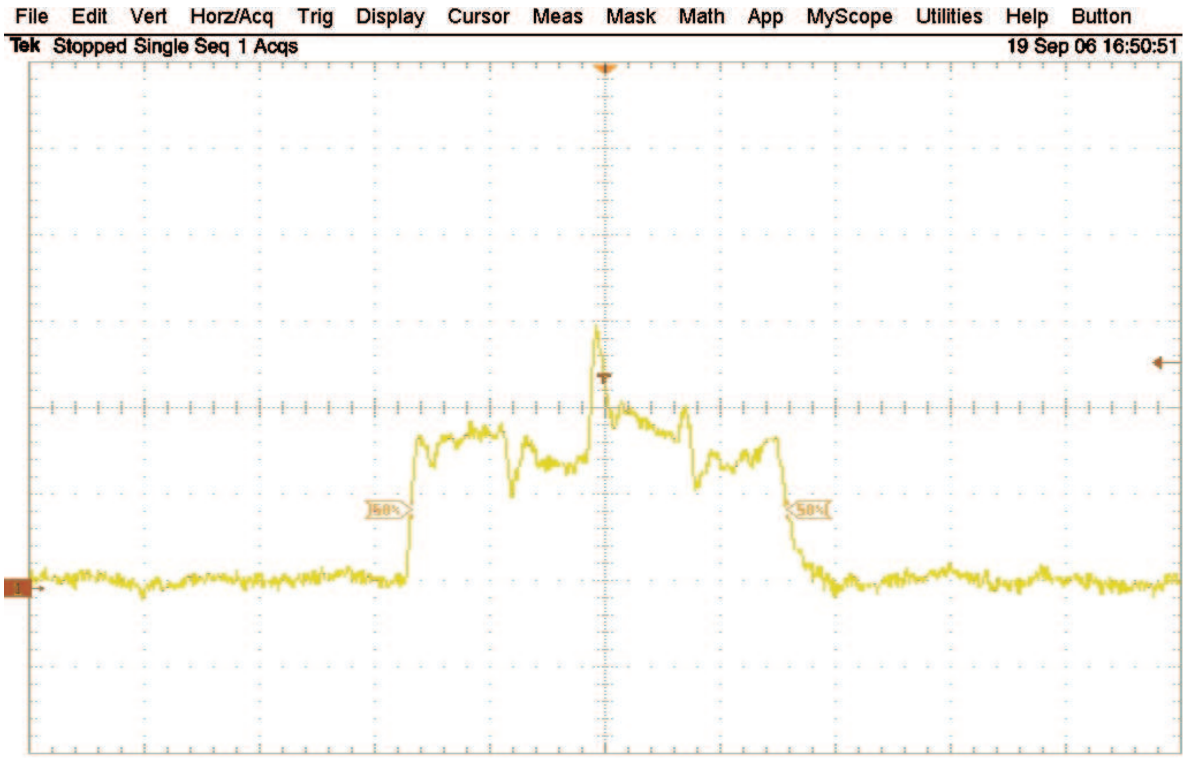

Figure 10. EMU Signal at Probe Point With Impedance Mismatch

3 Common Targets

3.1 TMS320C6455 DSK

3.1.1 Revision A, B

Revision A and B of the TMS320C6455 DSKs use TMS320C6455 revision 1.1 silicon, which has a known TCK sensitivity issue [1]. Using devices with this issue is not ideal; however, the steps described below improve the quality of the signal so this issue can be avoided. There are also known issues with the TMS320C6455 DSK that impact the integrity of the TCK and EMU signals. These issues, along with workarounds, are described in the subsection below. If you are using one of these DSKs it is important that you follow the recommendations described in Table 3 before attempting to use trace.

Table 3. TMS320C6455 DSK Revision A/B Recommendations

Problem	Recommendation
Missing ac termination on TCK (R95 and C126 are not populated)	Populate R95 with a 100 Ω resistor and C126 with an 8.2- pF capacitor.
Long trace stub on TCK to RTCK	Cut the trace to RTCK as close to the junction point as possible.(1)
Series termination resistor wrong value	Replace R96 and R93 with a 22-Ω resistor each
Long trace stub on EMU0 and EMU1	Remove resistors R510 and R511.
Resistor on TVD too large (R91)	Replace R91 with a 100-Ω resistor.

(1) The loopback capability of the XDS560 trace pod (described in Section 4.2.1) must be enabled after this change.

Note: All succeeding board revisions should be verified for design compliance.

3.1.1.1 Revision C

The revision C DSK is similar (slightly different board trace layout) to the revision A and B DSKs except that some revision C boards are delivered with TMX320C6455 revision 2.0 silicon that improve the TCK sensitivity issue described in Section 2.4.1.1. However, the board itself has the same issues present on revision A/B as described in Table 4. Resistor R105 can be removed to eliminate the long trace stub on TCK to RTCK instead of cutting a trace on the board itself. In revision C boards R95, C126 are installed. Should the modifications not fully solve the identified problems it is recommended that the trace output be reconfigured for 10-bit trace (8 data and 2 clocks) instead of the default 12-bit trace (10 data and 2 clocks). See Section 4.2.2 for the configuration of 10-bit trace.

In addition to the above modifications it is recommended that the following files be installed and used:

1. Code Composer Studio Configuration File: TCI6482_C6455_BHUSB560_Trace_TP.ccs. This file is intended to be used with a Blackhawk emulator. If a different emulator is used then the appropriate configuration file should be generated for the emulator used. The configuration file is located on the TI extranet and should be placed in the <CCSTUDIO_INSTALL>\drivers\import directory. During Code Composer Studio setup – it is necessary to import this particular file.
2. Code Composer Studio GEL File: DSK6455_TP.gel. The gel file allows you to generate trace patterns to test signal integrity. This file is located on the TI extranet for use. Always perform a *“Trace Off* sequence when you are through running a trace pattern export test. The Gel file should be placed in the <CCSTUDIO_INSTALL>\cc\gel directory. If the Code Composer Studio Setup is configured properly, this gel file will load automatically.

Table 4. TMS320C6455 DSK Revision C Recommendations

Problem	Recommendation
Missing ac termination on TCK (R95 and C126 are not populated)	Populate R95 with a 100 Ω and C126 is 8.2 pF
Long trace stub on TCK to RTCK	Cut the trace to RTCK as close to the junction point as possible.(1)
Series termination resistor wrong value	Replace R96 and R93 with a 22 Ω resistor each
Long trace stub on EMU0 and EMU1	Remove resistors R510 and R511.
Resistor on TVD too large (R91)	Replace R91 with a 100 Ω resistor.
Reflections from RTCK net	Remove R105

(1) Note that the loopback capability of the XDS560 trace pod (described in Section 4.2.1) must be enabled after this change.

Note: All succeeding board revisions should be verified for design compliance.

3.2 TMS320TCI6482

The TMS320TCI6482 target is typically supplied with a mezzanine card. See the *TMS320TCI6482 DSP EMAC/MDIO Module Reference Guide* (SPRUE12) for proper configuration. Current versions of the TMS320TCI6482 silicon have a known TCK sensitivity issue [1]. Using devices with this issue is not ideal; however, the steps described below improve the quality of the signal so this issue can be avoided. There are also known issues with the TMS320TCI6482 that impact the integrity of the TCK and EMU signals. These issues, along with workarounds, are described in the subsection below. If you are using one of these DSKs it is important that you follow the recommendations described in Table 5 before attempting to use trace.

Table 5. TMS320TCI6482 Revision Recommendations

Problem	Recommendation
Missing ac termination on TCK (R95 and C126 are not populated)	Populate R95 with a 100 Ω and C126 is 8.2 pF
Long trace stub on TCK to RTCK	Cut the trace to RTCK as close to the junction point as possible.(1)
Series termination resistor wrong value	Replace R96 and R93 with a 22-Ω resistor each
Long trace stub on EMU0 and EMU1	Remove resistors R510 and R511
Resistor on TVD too large (R91)	Replace R91 with a 100-Ω resistor
Reflections from RTCK net	Remove R105

(1) Note that the loopback capability of the XDS560 trace pod (described in Section 4.2.1) must be enabled after this change.

In addition to the above modifications it is recommended that the following files be installed and used:

1. Code Composer Studio Configuration File: TCI6482_C6455_BHUSB560_Trace_TP.ccs. This file is intended to be used with a Blackhawk emulator. If a different emulator is used then the appropriate configuration file should be generated for the emulator used. The configuration file is located on the TI extranet and should be placed in the <CCSTUDIO_INSTALL>\drivers\import directory. During Code Composer Studio setup it is necessary to import this particular file.
2. Code Composer Studio GEL File: DSK6455_TP.gel (see Note below). The gel file allows you to generate trace patterns to test signal integrity. This file is located on the TI extranet for use. Always perform a *Trace Off* sequence when you are through running a trace pattern export test. The Gel file should be placed in the <CCSTUDIO_INSTALL>\cc\gel directory. If Code Composer Studio Setup is configured properly this gel file will load automatically.

Note: The GEL file has been combined with the TMS320C6455 GEL file.

3.3 TMS320C6416 DSK

3.3.1 Revision D

Note: Trace is not supported on order TMS320C6416 DSK's incorporating Rev. 1.0 silicon.

Revision D of the TMS320C6416 DSKs uses TMS320C6416 revision 1.1 silicon. This is the only known good version of the C6416 silicon available to date. Excessive trace lengths, primarily on EMU0 and EMU1 create reflections that are unacceptable when operating at the trace export frequency. Modifications on the C6416 DSK are extensive and require a significant amount of physical work to the board. The recommendations are identified in the following subsection and in the C6416 DSK ECN-001. It is strongly recommended that the ECN not be implemented and instead the trace output be reconfigured for 10-bit trace (8 data and 2 clocks) instead of the default 12-bit trace (10 data and 2 clocks). See Section 3.3.2 for the configuration of 10-bit trace.

In addition to the above modifications it is recommended that the following files be installed and used:

1. Code Composer Studio Configuration File: BH-USB560m_c64xx_11_TP.ccs. This file is intended to be used with a Blackhawk emulator. If a different emulator is used then the appropriate configuration file should be generated for the emulator used. The configuration file is located on the TI extranet and should be placed in the <CCSTUDIO_INSTALL>\drivers\import directory. During Code Composer Studio setup it is necessary to import this particular file.
2. Code Composer Studio GEL File: dsp641x_TP.gel. The gel file allows you to generate trace patterns to test signal integrity. This file is located on the TI extranet for use. Always perform a *Trace Off* sequence when you are through running a trace pattern export test. The Gel file should be placed in the <CCSTUDIO_INSTALL>\cc\gel directory. If Code Composer Studio Setup is configured properly this gel file will automatically load.

Table 6. TMS320C6416 DSK Trace Modifications

Problem	Recommendation
Excessive reflections on EMU0 and EMU1	Cut EMU0 and EMU1 nets on PCB per ECN -001

3.3.2 Enabling 10-Bit Trace

To disable the 12-bit trace within Code Composer Studio and enable 10-bit trace, the following procedures should be used:

Step 1. Start Code Composer Studio, and load your application.

Step 2. From the *Tools→Trace→Control*.

Step 3. From the Control window select *10-bit* trace.

Step 4. Depress the *Apply* button.

Step 5. Close all completed windows not needed.

Step 6. Operate trace as originally planned.

4 Usage Notes

4.1 Special Note on Connecting the 60-Pin Emulator to Target

Care should be exercised when connecting to the high-density 60-pin emulation header. The connector selected is a high density connector of the highest quality available. Please use caution when inserting and removing the emulator connector to the target board connector and never force the emulator connector onto the target connector. The connector, when inserted correctly, will seat fully. A minimum amount of force is required to make a proper connection. If excessive force is required or the mated pair do not fully seat it is possible that a pin has been bent. Use extreme care when attempting to straighten a pin. You can permanently damage the connector and emulator if not handled properly. Always examine the target connector and emulator connector prior to mating the two.

4.2 TMS320C6455

The TMS320C6455 device has a sensitive TCK input buffer that makes the device prone to failure by capturing false edges from the TCK signal. The issue is described in detail in the *TMS320C6455/54 Digital Signal Processor Silicon Errata (Silicon Revs 2.0, 1.1)* (literature number SPRZ234) [1].

As mentioned in the errata, you can work around the problem by ensuring proper board design (described in Section 2), and by using a loopback capability built into the XDS560 trace pod. TCK loopback uses a dedicated buffer in the XDS560 trace pod connector to drive the RTCK back to the emulator directly from the TCK source, effectively ignoring the RTCK signal on the 60-pin header.

The TMS320C6455 modifications may not solve all problems (depending on target design and tolerances). Should the modifications not fully solve the identified problems it is recommended that the trace output be reconfigured for 10-bit trace (8 data and 2 clocks) instead of the default 12-bit trace (10 data and 2 clocks). See Section 4.2.2 for the configuration of 10-bit trace.

To enable the loopback feature, follow the steps provided in the section below.

4.2.1 Enabling XDS560 Trace Pod TCK Loopback Feature

Code Composer Studio Setup can be used to enable the XDS560 trace pod loopback feature for the C6455 device. To do this:

Step 1. Select the *Auto-generate board data file with extra configuration file* option in your board’s *Connection Properties* window

Step 2. Specify the *xds560tracepod.cfg* file located in your <CCSTUDIO_INSTALL>\cc\bin\BrdDat" directory as the *Configuration File*.

4.2.2 Enabling 10 Bit Trace

To disable the 12-bit trace within Code Composer Studio and enable 10-bit trace the following procedures should be followed.

Step 1. Start Code Composer Studio, and loading your application

Step 2. From the *Tools→Trace→Control*

Step 3. From the Control window, select *10-bit* trace

Step 4. Depress the *Apply* button

Step 5. Close all completed windows not needed

Step 6. Operate trace as originally planned

4.3 TMS320TCI6482

The TMS320TCI6482 device has a sensitive TCK input buffer that makes the device prone to failure by capturing false edges from the TCK signal. The issue is described in detail in the Silicon Errata of the TMS320TCI6482 [2].

As mentioned in the errata, the work around for this problem is by ensuring proper board design and by using a loopback capability built into the XDS560 trace pod. TCK loopback uses a dedicated buffer in the XDS560 trace pod connector to drive the RTCK back to the emulator directly from the TCK source, effectively ignoring the RTCK signal on the 60-pin header.

The TMS320TCI6482 modifications may not solve all problems (depending on target design and tolerances). Should the modifications not fully solve the identified problems, it is recommended that the trace output be reconfigured for 10-bit trace (8 data and 2 clocks) instead of the default 12-bit trace (10 data and 2 clocks). See Section 4.3.2 for the configuration of 10-bit trace.

To enable the loopback feature, follow the steps provided in Section 4.3.1.

4.3.1 Enabling XDS560 Trace Pod TCK Loopback Feature

Code Composer Studio setup can be used to enable the XDS560 trace pod loopback feature for the TCI6482 device. To do this:

Step 1. Select the *Auto-generate board data file with extra configuration file* option in your board's *Connection Properties* window

Step 2. Specify the *xds560tracepod.cfg* file located in your <CCSTUDIO_INSTALL>\cc\bin\BrdDat" directory as the *Configuration File*

4.3.2 Enabling 10 Bit Trace

To disable the 12-bit trace within Code Composer Studio and enable the 10-bit trace, the following procedures should be followed.

Step 1. Start Code Composer Studio and load your application

Step 2. From the *Tools→Trace→Control*

Step 3. From the Control window select *10-bit* trace

Step 4. Depress the *Apply* button

Step 5. Close all completed windows not needed

Step 6. Operate trace as originally planned

5 Troubleshooting

This section aids you in troubleshooting common issues that may be related to electrical issues. Online help, accessible through the *Trace Display* and *Trace Control* menus in the Code Composer Studio, should be utilized to troubleshoot issues not discussed in this section.

5.1 Code Composer Studio Disconnects From the Device When Running With Trace Enabled

If you are using a TMS320C6455 or TMS320TCI6482 and you encounter a Code Composer Studio disconnect when running with trace turned on, it is likely you are encountering the issues described in Section 4.2 or Section 4.3. Verify that the loopback mode has been turned on and the checks described in Section 2.1 have been performed.

5.2 Receiving Bad Data

A *Bad Data* warning may be generated in the *Trace Display* window for a number of reasons, including:

- Application code includes self-modifying code
- Erroneous trace data recorded because of transmission line problems (e.g., reflections)

To detect a transmission line problem, the first step to take is to be sure the checks described in

Section 2.1 have been performed. In the past there have been specific issues with the EMU0 and EMU1 signals on certain boards, so they should be looked at specifically. If there is an issue with the EMU0 and EMU1 signals but not the others, then you can remove the dependency on these pins by configuring trace to be exported over 10 pins instead of the default 12. This can be performed using the Trace Control window.

If signal looks good, the second thing to try is to lower the Trace Export Clock Rate through the *Trace Control* window.

The online help for trace should be referenced for additional information related to receiving a *Bad Data* warning message.

5.3 Pod Error/Pod Setup Error

Errors encountered during the trace calibration process may indicate an electrical or mechanical problem. Some of the more common calibration errors are described:

Pod Setup Error - Low frequency DAC adjust failed	
Description	The trace calibration process tests across all signal threshold DAC values at the lowest frequency possible. If no good values are found, this error terminates the trace calibration process.
Cause	This error may indicate the trace pod target cable is not seated properly or an electrical problem with the target's EMU signals.

Pod Setup Error - Skew adjust failed	
Description	The calibration process failed during skew adjust.
Cause	This error probably indicates an electrical problem with the target's EMU signals.

Pod Error - Less than half a nanosecond of sampling margin detected	
Description	The calibration process failed during skew adjust.
Cause	This error probably indicates an electrical problem with the target's EMU signals.

5.4 Known Software Issues

A list of known software issues is provided in the product release notes [5]. If a software issue is encountered that is not listed, it is recommended that a defect report be filed per the instructions also described in the product release notes.

5.4.1 Code Composer Studio Crashes

There are no known issues that should cause the Code Composer Studio application to unexpectedly exit. However, if this were to occur then it is possible you may have trace software running in the background erroneously. Using the Windows® Task Manager, search for and terminate any *traceserver.exe*, *tracecntrl.exe*, *tracedisplay.exe* and *tracecompmgr.exe* processes that may be running.

6 Glossary

AC Termination — The ac termination comprises a resistor, R, and a capacitor, C, that connect to the load end of the transmission line.

Reflections — Reflections are the result of an impedance mismatch in electrical signals. When voltage hits a discontinuity, some energy is reflected. This occurs in any change in a material's final stop (connection to another material). [3], [4]

Series Termination — Series termination is a source-end termination unlike other types. A series termination comprises a resistor between the driver's output and the line.

7 References

1. *TMS320C6455/54 Digital Signal Processor Silicon Errata (Silicon Revs 2.0, 1.1)* (SPRZ234)
2. *TMS320TCI6482 Digital Signal Processor Silicon Errata (Silicon Revs 2.0, 1.1)* (SPRZ235)
3. Ethirajan K, Nemec J, *Termination Techniques for High-Speed Buses*, EDN Design Feature
4. *60-Pin Emulation Header Technical Reference* (SPRU655)
5. *XDS560 Trace v0.1.0 Beta Release Notes* (SPRA377)
6. www.ti.com -- Texas Instruments website

IMPORTANT NOTICE

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
Low Power Wireless	www.ti.com/lpw	Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265