Application Report
SLAA777A–October 2017–Revised January 2019

Using SimpleLink™ MSP432E4 microcontrollers over the JTAG interface

Amit Ashara

ABSTRACT

The IEEE Standard 1149.1-1990, IEEE Standard Test Access Port and Boundary-Scan Architecture (JTAG) is a method for verifying designs and testing printed circuit boards after assembly. It is used as the primary means for transferring data to a nonvolatile memory of an embedded system and debugging embedded software.

This application report describes the physical connections for JTAG and design considerations to be taken into account for a custom board. It also shows how to use the JTAG interface on the SimpleLink™ MSP432E4 LaunchPad™ development kit for debugging the onboard microcontroller using an external debugger, or by using the onboard debugger for debugging an off-board microcontroller.

NOTE: This document applies to MSP432E4 microcontrollers. All screen captures reflect the MSP432E4 devices.

Contents

List of Figures

List of Tables

Trademarks

SimpleLink, LaunchPad, Code Composer Studio are trademarks of Texas Instruments.
Arm, Keil, µVision are registered trademarks of Arm Limited.
IAR Embedded Workbench is a registered trademark of IAR Systems.
All other trademarks are the property of their respective owners.

1 Introduction

JTAG defines a Test Access Port (TAP) and Boundary Scan Architecture for digital integrated circuits and provides a standardized serial interface for controlling the associated test logic. It can be used to test the interconnections of assembled printed circuit boards and obtain manufacturing information on the components. It also provides a means of accessing and controlling the design for test features such as I/O pin observation and control, scan testing, and debugging.

The JTAG port on MSP432E4 uses of four pins:

- Test Clock (TCK)
- Test Mode Select (TMS)
- Test Data In (TDI)
- Test Data Out (TDO)

Even though the MSP432E4 devices support Serial Wire Debug (SWD) mode, this application report describes the use of JTAG, which is a more widely adopted interface.

2 Overview of JTAG Protocol

Before the interfacing of MSP432E4 devices over JTAG can be discussed, it is important to understand the basic concepts of the JTAG protocol and terminologies. This aids the debugging of system issues when JTAG does not work as expected.

2.1 JTAG State Machine

JTAG accesses the Test Access Port (TAP) of a device by changing TMS and TDI in conjunction with TCK and reading results through TDO.

- TDI and TMS are sampled on the rising edge of TCK by the TAP.
- TDO is changed on the falling edge of TCK by the TAP.

The JTAG works by accessing the Instruction Register (IR) and the Data Register (DR). The IR is a 4-bit serial scan chain connected between the TDI and TDO pins. When the TAP controller is in the correct state, bits can be shifted into the IR. Once the IR is loaded, they are decoded to get access to the DR. The DR format is specific to the register being accessed. As an example:

- When the IR is loaded with the BYPASS instruction, the DR length is a 1-bit shift register.
- When the IR is loaded with IDCODE instruction, the DR length is a 32-bit shift register, which on shift out, gives the JTAG ID of the device that is specific to the manufacturer name, part number and version of the Arm® core.

At power on, the TAP state machine (see Figure 1) is initialized to be in Test Logic Reset state. It moves from one state to another based on the TMS value (shown as logic 0 or 1 on the transition arrow) with every TCK. Once the state machine enters the Shift state, the TDI pin is used to serially shift in the IR or DR. At the same time, the value captured in the shift register during the Capture state is shifted out on to the TDO. When the Update state is executed, the value shifted in during the Shift state is updated to the TAP for the next JTAG cycle.

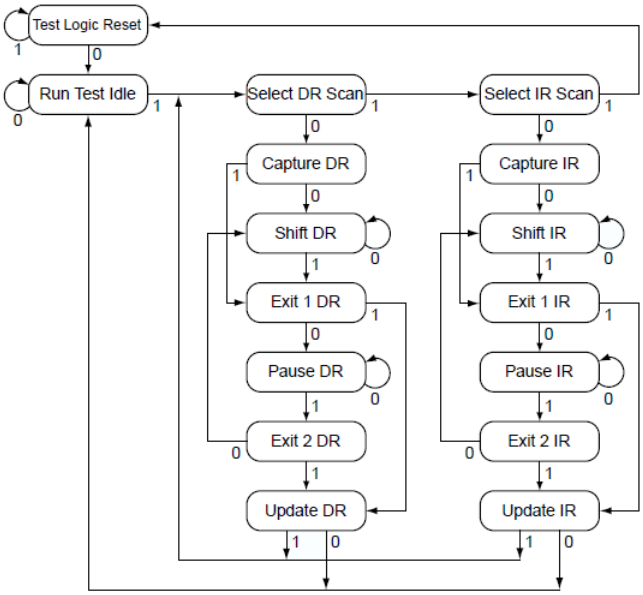

Figure 1. Test Access Port State Machine

2.2 JTAG System Implementations

A system with JTAG can be implemented in two different methods: point-to-point connection and daisy chain connection.

2.2.1 Point-to-Point Connection

Using a point-to-point method, a single set of JTAG pins are connected to a single device (see Figure 2). If there are multiple devices in a system, then multiple JTAG headers are required. The advantage of the point-to-point connection is that the speed of the device access is higher when compared to the daisy chain connection (discussed in Section 2.2.2). However, this increases the BOM cost and is more difficult to manage when performing cross trigger debug.

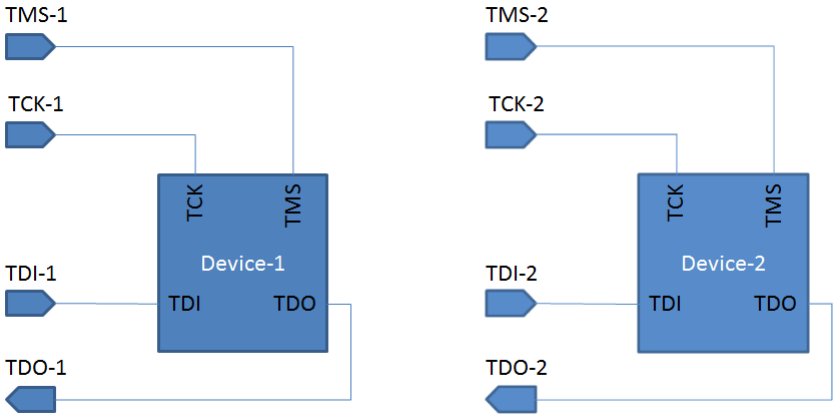

Figure 2. Point-to-Point Connection

2.2.2 Daisy Chain Connection

Using the daisy chain connection method, a single set of JTAG pins are connected to multiple devices (see Figure 3). The TDI from the JTAG header is connected to the TDI of the first device. The TDO of the first device is then connected to the TDI of the next device and so on until the last device. The TDO of the last device is connected to the TDO on the JTAG header. TCK and TMS are shared between all the devices. The advantage of a daisy chain connection is that the BOM cost is lower as individual headers per device is not required, PCB is simpler to layout and cross trigger debugging is convenient. Since multiple device TAPs are in the path, JTAG access is slower as additional shifts are required to bring each device TAP to the correct state.

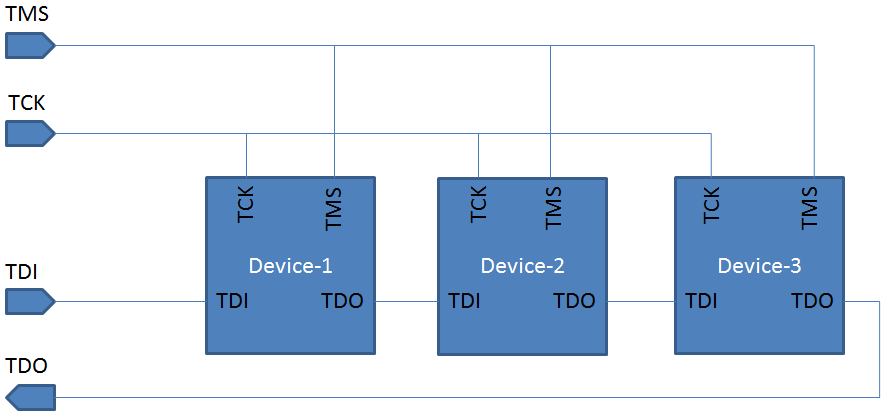

Figure 3. Daisy Chain Connection

2.3 Additional JTAG Information

In addition to the four JTAG pins, two more pins are available on some devices: TRST and RTCK.

- TRST: Test Reset. TRST is an optional pin and can be used to reset the JTAG TAP state machine. It is an active low signal.
- RTCK: Return TCK. RTCK is a clock that is sourced by the device. When available, the TDO is sampled with this clock allowing for higher JTAG operation frequency. This is also referred to as adaptive clocking.

NOTE: TRST and RTCK are unavailable on all MSP432E4 devices.

3 JTAG Debug Probes

To access the JTAG interface on an embedded system requires JTAG debug probes. These are often referred to as emulators (though this is a misnomer). The JTAG debug probes have control logic that generates the JTAG signaling and comes with drivers that a PC-based application like IDE's and programmers can utilize to download a firmware or debug an application.

3.1 JTAG Header Pinout

The JTAG debug probe connects to headers on the PCB. There are different types of JTAG connectors and even though they serve the same function, it can be confusing when making the selection of the debug probe. To be able to make the correct connection, it is necessary to first understand the pin mapping of the JTAG header pinout. Table 1 shows the pinout for the different type of headers.

Table 1. JTAG Header Pin Out

Pin Number	TI 14-Pin	Compact TI 20-Pin cTI	Arm 10-Pin	Arm 20-Pin
	100-mil Pitch	50-mil Pitch	50-mil Pitch	100-mil Pitch
1	TMS	TMS	VTRef	VTRef
2	nTRST	nTRST	TMS	VSupply
3	TDI	TDI	GND	nTRST
4	TDIS	TDIS	TCK	GND
5	VTRef	VTRef	GND	TDI
6	KEY	KEY	TDO	GND
7	TDO	TDO	KEY	TMS
8	GND	GND	TDI	GND
9	RTCK	RTCK	GNDDetect	TCK
10	GND	GND	nRESET	GND
11	TCK	TCK		RTCK
12	GND	GND		GND
13	EMU0	EMU0		TDO
14	EMU1	EMU1		GND
15		nRESET		nRESET
16		GND		GND
17		EMU2		NC
18		EMU3		GND
19		EMU4		NC
20		GND		GND

3.2 Connecting Debug Probes to MSP432E4

Section 3.1 lists the different pinout based on the connector selected. However, based on the function of each pin, it may still be unclear what needs to be connected to the MSP432E4 for debug and what to do with unused pins. Table 2 describes the function of each of the pins listed earlier in Table 1 and how they need to be connected to MSP432E4 devices.

Table 2. Pin Connectivity

Pin Name	MSP432E4 Pin to Connect	Comments
TCK	PC0 pin with a 10-kΩ pullup	Test clock
TMS	PC1 pin with a 10-kΩ pullup	Test mode select
TDI	PC2 pin	Test data in
TDO	PC3 pin with a 10-kΩ pulldown	Test data out
VTRef	VDD supply of MSP432E4 through a 100-Ω series resistor	Reference voltage used by level shifter on the debug probe to level shift signals if the IO voltage on the embedded system and debug probe are different
VSupply	NC	Not used
TDIS	GND	Target disconnect detect
RTCK	NC	Return test clock for adaptive clocking
EMU0	NC	Function depends on the target device
EMU1	NC	Function depends on the target device
EMU2	NC	Reserved for future use
EMU3	NC	Reserved for future use
EMU4	NC	Reserved for future use
KEY	NC	Pin removed from the header, so that the debug probe can be connected on unshrouded header
nTRST	NC	Test reset pin
nRESET	RST_N pin with a 10-kΩ pullup resistor	Target reset pin
GNDDetect	GND	Same as TDIS
GND	GND	Common ground

NOTE: The pullup values mentioned in Table 2 are only for guidance. If there is a strong noise source close to the JTAG traces or reset pin, then the pullup resistor value needs to be decreased to make it less susceptible to noise coupled from traces.

3.3 JTAG Debug Probes and Software

There are multiple vendors that supply standalone JTAG debug probes like Spectrum Digital and Black Hawk, to name two. Besides the standalone debug probes, TI embedded LaunchPad development kit (MSP-EXP432E401Y) comes with an onboard debug probe. Also, TI provides IDE and standalone programming software Uniflash to download firmware. This section briefly introduces some of the debug probes that can be interfaced with MSP432E4 and software support.

3.3.1 XDS100 Series

XDS100 is the entry level debug probe and is provided by multiple vendors. MSP432E4 is supported by both v2 and v3 versions of XDS100 series.

3.3.2 XDS110 Series

XDS110 is the next level of debug probe to XDS110. MSP432E4 is supported by XDS110 stand-alone debug probe.

3.3.3 XDS200

XDS200 is an intermediate debug probe that has better performance than XDS100 and lower cost than XDS560v2.

3.3.4 XDS560v2

XDS560v2 is a high performance debug probe and is provided by multiple vendors. When setting XDS560v2 as the debug probe, ensure that the Connection Properties has the option *JTAG TCLK Frequency (MHz)* set to *TCLK looped-back with a user-specified limit* (see Figure 4).

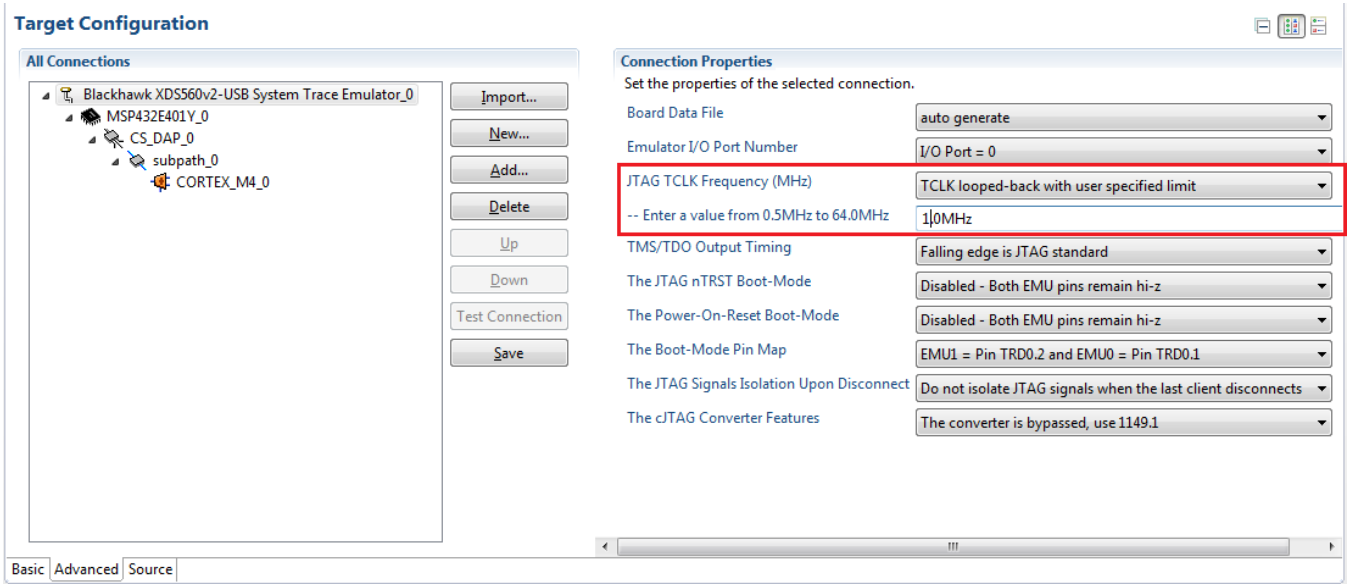

Figure 4. XDS560v2 Debug Probe Special Consideration

3.3.5 I-jet

The I-jet debug probe is provided by IAR Systems for IAR Embedded Workbench® IDE.

3.3.6 ULINK2

The ULINK2 debug probe is provided by Arm for Keil® µVision® IDE.

3.3.7 Software Support for Debug Probe

It is also important to know which software to use when using a JTAG debug probe and what features are supported for MSP432E4. Table 3 is a useful look up when selecting a debug probe and programming software.

Table 3. Software Support for Debug Probe

Debug Probe	IDE Supported	UniFlash Support	Unlock Feature
XDS100	Code Composer Studio	Yes	Yes, with UniFlash
XDS200	Code Composer Studio	Yes	Yes, with UniFlash
XDS560	Code Composer Studio	Yes	No
XDS110	Code Composer Studio, IAR Embedded Workbench, Keil µVision	Yes	Yes, with UniFlash
IAR I-jet	IAR Embedded Workbench	No	No
uLink2	Keil µVision	No	No

4 Using JTAG Debug Probes With MSP432E4

After having a basic understanding of the JTAG protocol, debug probes and pinout, the next step is to interface the debug probe to a MSP432E4 microcontroller. The following sections reference schematics for the different header pinouts.

4.1 Sample Schematic for TI 14-Pin Header

Figure 5 shows how the JTAG header on the board needs to be connected to the MSP432E4 device for a TI 14-pin header.

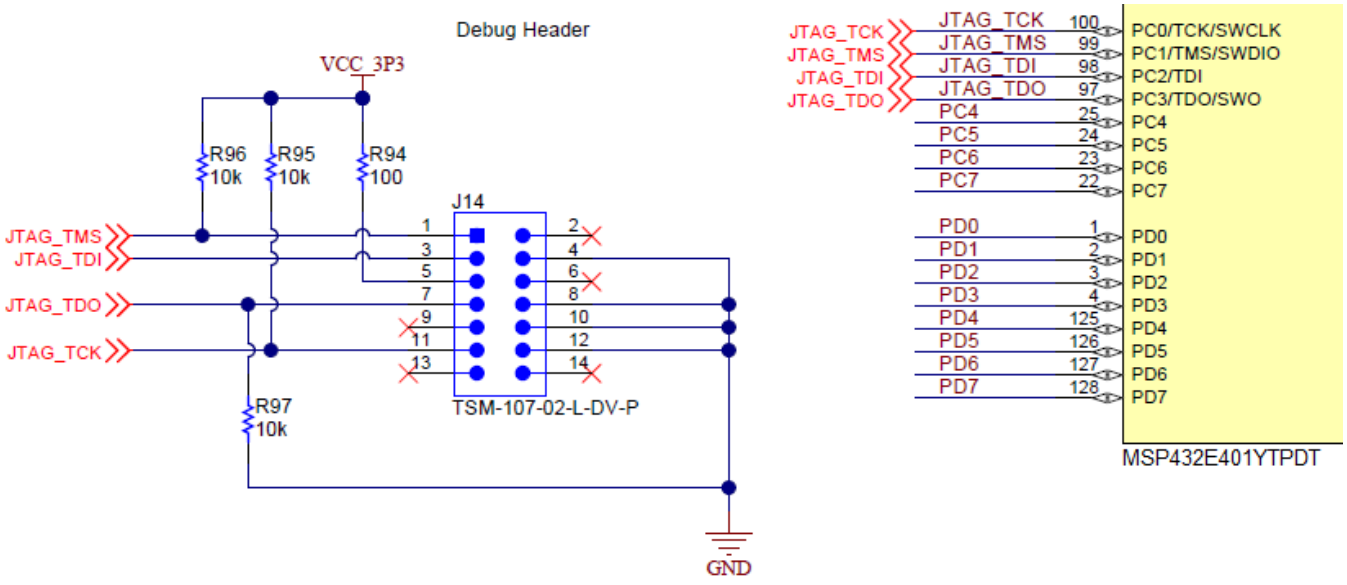

Figure 5. TI 14-Pin Header

4.2 Sample Schematic for Compact TI 20-Pin Header

Figure 6 shows how the JTAG header on the board needs to be connected to the MSP432E4 device for a compact TI 20-pin header.

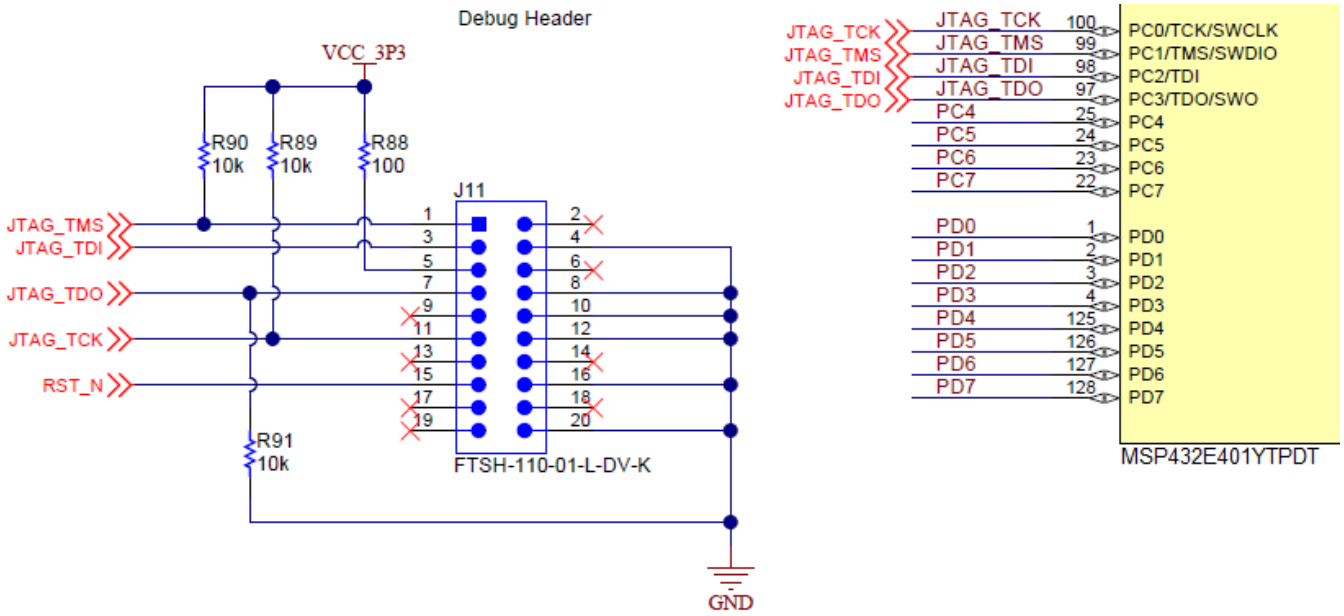

Figure 6. CTI 20-Pin Header

4.3 Sample Schematic for Arm 10-Pin (Cortex Debug) Header

Figure 7 shows how the JTAG header on the board needs to be connected to the MSP432E4 device for an Arm 10-pin header.

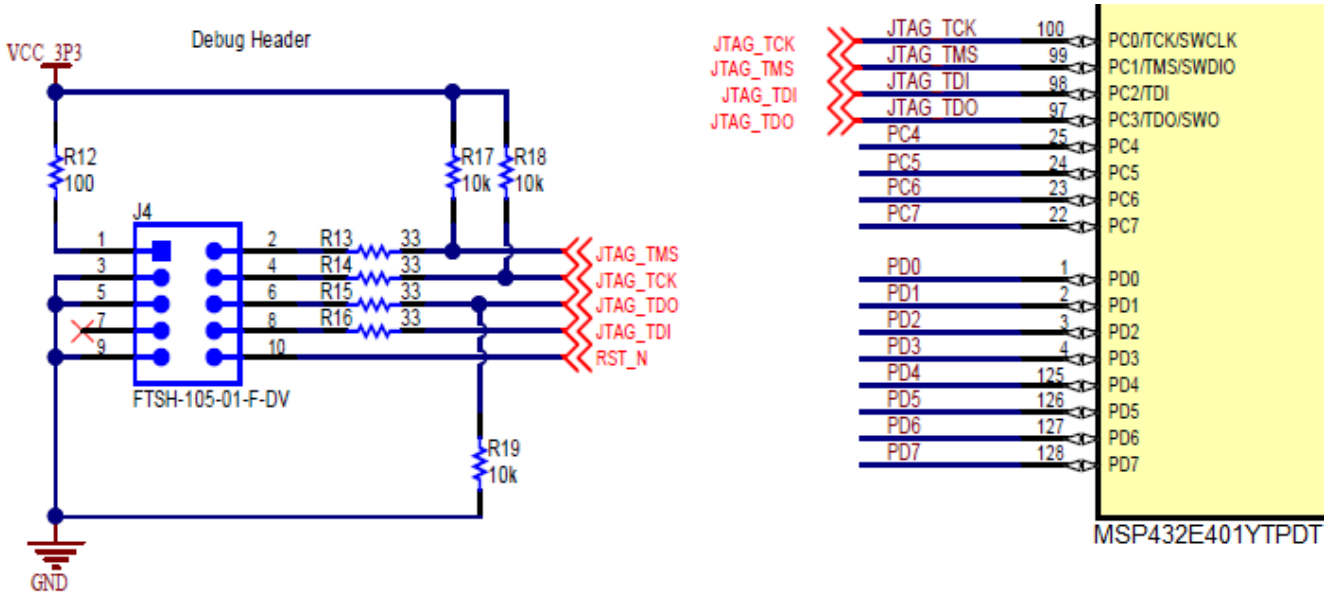

Figure 7. Arm 10-Pin Header

4.4 Sample Schematic for Arm 20-Pin Header

Figure 8 shows how the JTAG header on the board needs to be connected to the MSP432E4 device for an Arm 20-pin header.

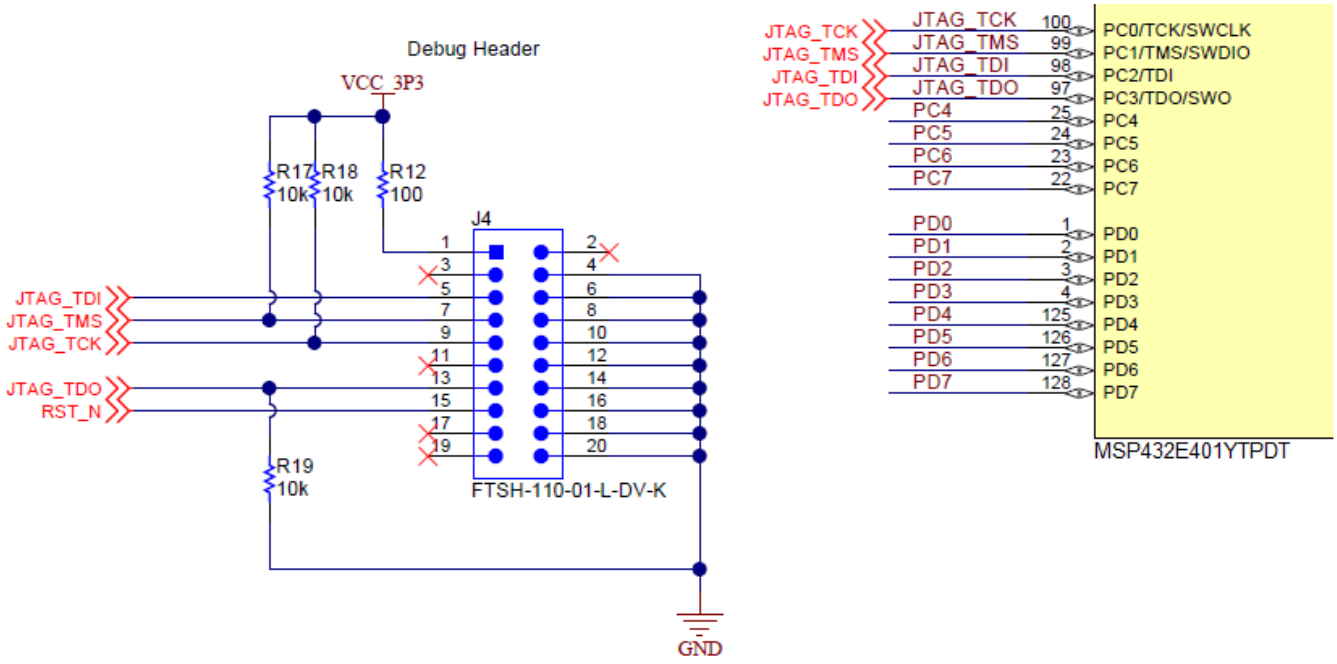

Figure 8. Arm 20-Pin Header

4.5 Using an External Stand-Alone Debug Probe With MSP-EXP432E401Y

The MSP-EXP432E401Y LaunchPad development kit comes with an onboard XDS110 debug probe. However, you may want to use an external debug probe. The JTAG pins are available as an Arm 10-pin header marked J11 (see Figure 9). Make sure that the header pins corresponding to RST, TMS, TCK, TDO, and TDI are removed from the header J101 on the MSP-EXP432P401Y. The debug USB cable can still be used to provide power to the board.

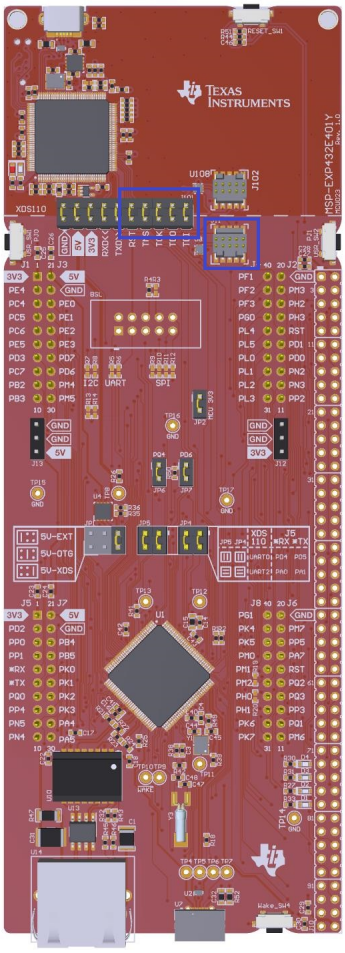

Figure 9. Connecting External Debug Probe

4.6 Using XDS110 on MSP-EXP432E401Y to Debug Off-Board MSP432E4

The MSP-EXP432E401Y onboard XDS110 can also be used to debug an off-board MSP432E4. To be able to connect the on board XDS110 as a debug probe, you must prepare the board (see Figure 10) as follows:

1. Remove the jumpers for RST, TMS, TCK, TDO and TDI on connector J101.
2. Connect a Arm 10-pin header and cable from J102 to the target device.

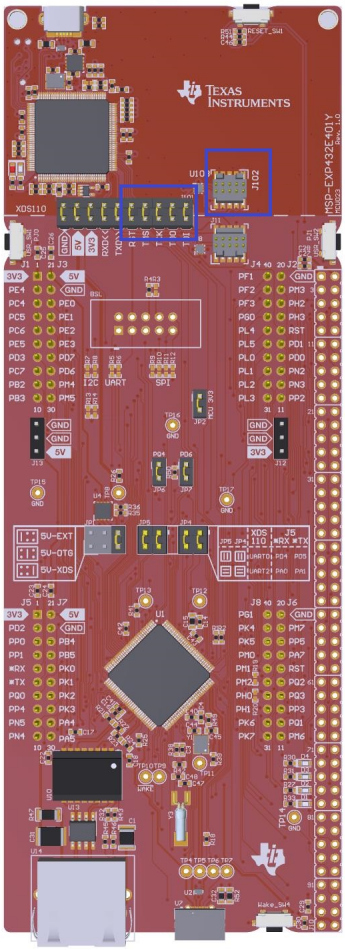

Figure 10. Preparing MSP-EXP432E401Y as a Debug Probe

Now the onboard XDS110 can be used to debug an off-board MSP432E4 device.

4.7 JTAG Adapter

JTAG adapters are available from many JTAG debug probe vendors to convert one JTAG pinout to another. This is very useful because the correct JTAG probe may not have been procured or one is not readily available. Having the adapter allows you to use another JTAG debug probe with a different pinout to be interfaced. These adapters mostly come with a single conversion option. As part of this application report, the schematic (see Figure 11) shows a universal adapter that can be used for any of the 4 JTAG pinouts.

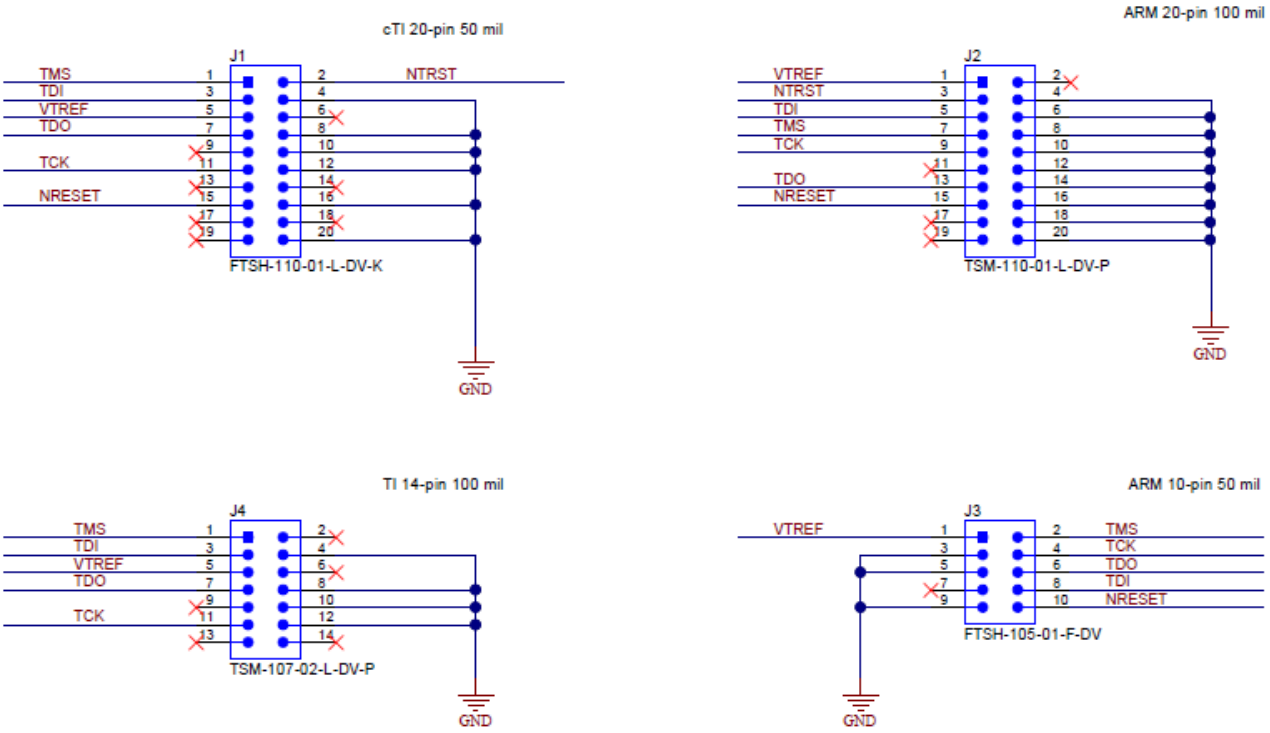

Figure 11. JTAG Adapter Schematic

5 MSP432E4 JTAG Interface Specific Behaviors

This section describes the specific behavior of JTAG pins on the MSP432E4 devices.

5.1 Default JTAG Pin Behavior on MSP432E4 Devices

The pins PC0-3 after power on reset are configured as JTAG pins and not GPIO pins. Also, these pins are locked for JTAG operation by means of a commit control register. This provides a layer of protection against accidental programming of the pins to a non-JTAG function. If the customer application requires JTAG pins to be used as GPIO, then the application must perform the following steps:

1. Enable the Clock to GPIO port C, by setting bit-2 in system control RCGCGPIO register. The application must ensure that the system control register PRGPIO bit-2 reads 1 before accessing the GPIO port C address range.
2. Unlock the GPIO port C to access the commit control register by writing 0x4C4F434B to GPIOLOCK register of GPIO port C. A read of the register must return the value 0x0.
3. Set the commit control bits by writing bits 0-3 of the GPIOCR register of GPIO port C with the value 0xF.
4. Lock the GPIO Port C by writing any value other than 0x4C4F434B to the GPIOLOCK register of GPIO port C.

It is important to note that if the JTAG pins are configured as any other function, then the JTAG function will not be accessible on execution of the application. You must provide a mechanism in the application to either revert the GPIO to JTAG function, or hold the device in a while loop by reading the state of a GPIO pin before the GPIO is configured to be a non-JTAG function. If this is not done, the only method to access JTAG function is to execute the "Unlock Sequence" as provided in the device-specific data sheet.

5.2 Effect of BOOTCFG Register on JTAG Function

All MSP432E4 devices have the BOOTCFG register. This register influences the behavior of the device boot after a power on reset. One of the functions is the ability to disable the JTAG function without configuring the GPIO port C as GPIO's. This is achieved by clearing the bits 0 and 1. For more information, see device-specific data sheet. When the bits are cleared, the JTAG function is disabled.

If the register is not committed, a subsequent power on reset would configure it to its default value allowing for the JTAG function to work. However, if the register is committed, then a subsequent power on reset would make the change permanent. The only method to access the JTAG function is to execute the "Unlock Sequence" as provided in the device-specific data sheet.

5.3 Executing Unlock Sequence

The unlock sequence is a method to reset the MSP432E4 microcontrollers to the factory state. When an unlock sequence is run, it has the following effects on the device:

- Flash and EEPROM are erased.
- All user committed registers are reset to the default value as described in the device-specific data sheet and the MSP432E4 SimpleLink™ Microcontrollers Technical Reference Manual.

The actual protocol sequence is specified in the device data sheets in the *JTAG Interface* section. To execute the unlock sequence, the user must have a debug probe and software (see Table 3) that can perform this action.

5.3.1 Unlock Sequence Using Uniflash and XDS Debug Probes

The Uniflash utility supports XDS100v2, XDS110, and XDS200 debug probes if the utility package is 7.0.98.0 or later. To execute the unlock sequence, follow these steps:

1. Start a Windows command prompt and change the working directory to the path to dbgjtag.exe under Uniflash. For example, in Figure 12, the path is "C:\ti\ccsv7\ccs_base\common\uscif" for dbgjtag.exe.
2. Run dbgjtag.exe to make sure the version of the utility package is at least 7.0.98.0 (see Figure 12).
3. Type "dbgjtag.exe –f @xds110 –Y unlock,mode=msp432e4" if you are using XDS110 (see Figure 12).

NOTE: Use "dbgjtag.exe -f @xds100v2 -Y unlock,mode=msp432e4" when using XDS100v2 debug probe.

Use "dbgjtag.exe -f @xds200 -Y unlock,mode=msp432e4" when using XDS200 debug probe.

To run these commands in Windows PowerShell, escape the @ character (type `@ instead of @).

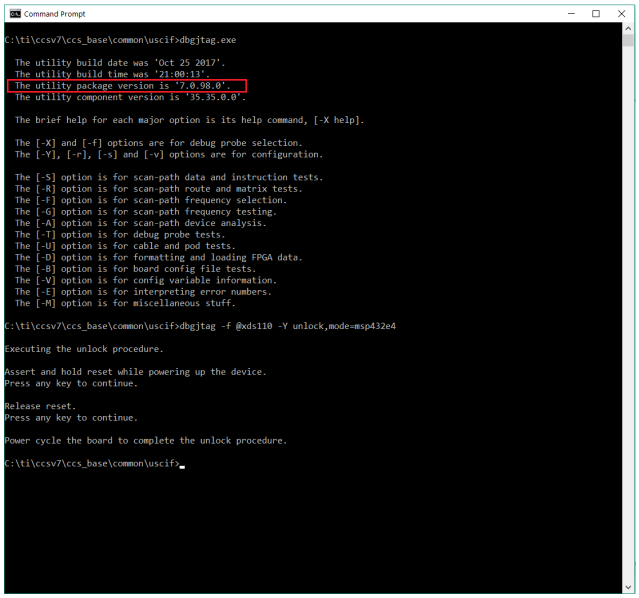

Figure 12. Unlock Sequence With Uniflash and XDS110

4. Power down the board containing the locked microcontroller.
5. Power up the board, while holding down the reset. The reset must remain pressed unless instructed by the dbgjtag.exe to be released.
6. Press Enter to execute the dbgjtag.exe with the options provided earlier, and then follow the instructions given by dbgjtag.exe.
7. When dbgjtag.exe instructs to release the reset, then release the reset and power cycle the board.

At the end of this step, the MSP432E4 microcontroller should return to its factory state.

6 Debugging JTAG Connection Failure

The JTAG interface is very useful to debug embedded software. However, when the JTAG interface does not work as expected, it is extremely important to understand how to debug when JTAG is not functional.

6.1 JTAG Not Working on the First Bring Up

When a new custom board is made, bringing up the JTAG interface is the most critical part, as only then can the firmware be programmed to the microcontroller. At the same time, this is the most difficult debug for JTAG. Make sure that you have an XDS class debug probe, as it provides a utility called *Test Connection* (see Figure 13) that is useful in debugging the root cause.

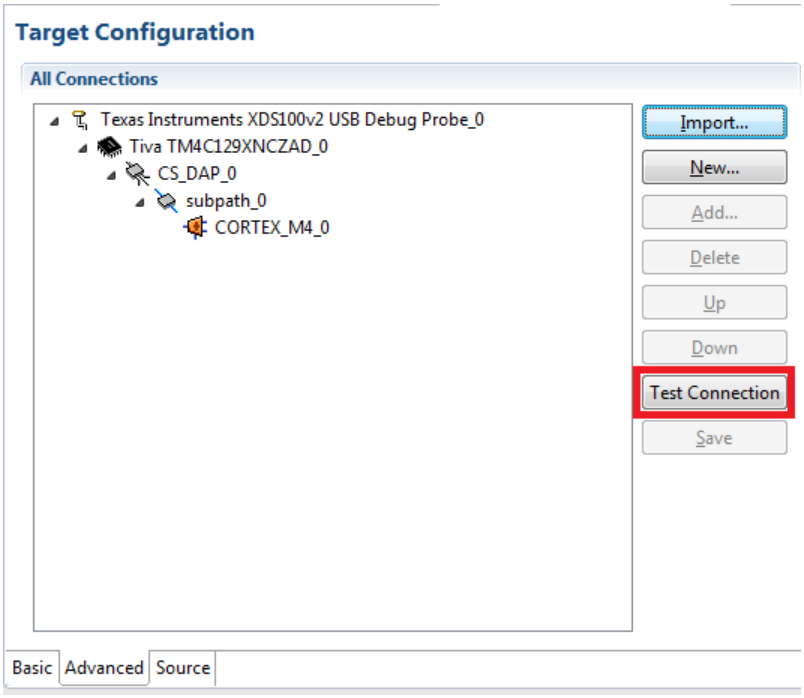

Figure 13. Test Connection Utility in CCS

When the button *Test Connection* is pressed, Code Composer Studio™ IDE executes the JTAG instruction of BYPASS that checks whether the JTAG physical connection integrity can be established. The result is shown in a window (see Figure 14).

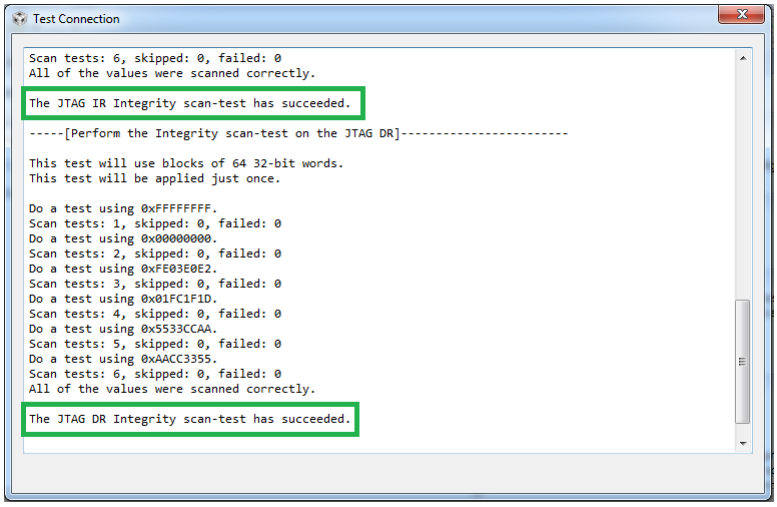

Figure 14. Test Connection Result

If the JTAG IR and DR integrity scan-test succeeds, the device core is out of reset and may not have initialized itself. If, however, the integrity scan-test fails, the issue is in the power up process.

Perform the following steps to make sure every known cause is eliminated and to find the source of the issue:

1. Check with a digital multimeter that the VDD and VDDA supply rails are 3.3 V.
   a. If not, then check the power supply aspect of the design.
2. Check with a digital multimeter that the VDDC rail is 1.2 V.
   a. If not and providing a power from an external power source, make sure that the current limit is set around 150 mA.
3. If the VDDC rail is at 1.2 V, make sure that the capacitance on the rail is as per the device-specific data sheet and the layout of the capacitors are as per the system design guidelines and application reports (see Section 8).
4. Check whether the JTAG header is correctly mounted and the TDIS pin (if available) is connected to GND.
5. Check on the JTAG header the VTREF pin is 3.3 V.
6. Check whether the reset pin of the microcontroller is at 3.3 V.
   a. If not, connect an external pullup.
7. If using an external crystal oscillator, connect an oscilloscope on pin OSC0.
   a. If the crystal is not oscillating, check the solder on the crystal and capacitors.
   b. Always use the recommended crystals listed in the device-specific data sheet.
8. If not using an external crystal oscillator, make sure that the pin OSC0 is connected to GND.
9. If using a MSP432E4 device, make sure that the RBIAS resistor is populated according to the specification in the device-specific data sheet.

6.2 JTAG Not Working After Software Was Loaded

If software was downloaded to the MSP432E4 and then it stopped working, the possible causes are limited and are likely be one of the items in Table 4.

Table 4. Debugging JTAG Failure on Software Download

Possible Causes	Remedial Measures
BOOTCFG was modified incorrectly	Execute the unlock sequence.
GPIO Port C pins were modified to be a function other than JTAG	Execute the unlock sequence.
System clock must be 10x the JTAG clock	If debug probe allows reducing JTAG clock then reduce JTAG clock frequency and try the JTAG connection again. If not then execute unlock sequence.
Device is in low-power mode	Wake up the device from low-power mode and then try the JTAG connection again. If it does not work, then execute the unlock sequence.
Power failure during software execution	Restore the power, and determine if the total power budget of the system is supported by the power supply.

7 Conclusion

This application report provides details on the JTAG interface of MSP432E4 microcontrollers, the debug probes and the software tools that can be used to erase, program, debug and unlock the microcontroller. Also this document provides sample schematics for connecting different JTAG header pinouts with the MSP432E4 microcontroller and an in-depth description on using MSP432E4 LaunchPad development kits as a debug probe. Finally, this document provides a detailed explanation on how to debug a JTAG connection failure, which is critical when bringing up an application-specific embedded system design.

8 References

The following related documents and software are available.

1. SimpleLink™ MSP432E411Y LaunchPad™ Development Kit User's Guide
2. System Design Guidelines for SimpleLink™ MSP432E4 Microcontrollers
3. Processors wiki for JTAG
4. XDS100 wiki page
5. XDS110 wiki page
6. XDS200 wiki page
7. XDS560 wiki page

Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from October 30, 2017 to January 11, 2019	Page
• Added the comment about escaping the @ character for Windows PowerShell in the note in Section 5.3.1, *Unlock Sequence Using Uniflash and XDS Debug Probes*	16

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

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