Multi-ICE System Design Considerations

Application Note 72

ARM

ARM DAI 0072A
Open Access

Application Note 72
Multi-ICE System Design Considerations

Release information

Change history

Description	Issue	Change
October 1998	A	First edition

Proprietary notice

ARM is a registered trademark of ARM Limited.

Multi-ICE, EmbeddedICE, and TrackingICE are the trademarks of ARM Limited.

All other products or services mentioned herein may be trademarks of their respective owners.

Neither the whole nor any part of the information contained in, or the product described in, this document may be adapted or reproduced in any material form except with the prior written permission of the copyright holder.

The product described in this document is subject to continuous developments and improvements. All particulars of the product and its use contained in this document are given by ARM Limited in good faith. However, all warranties implied or expressed, including but not limited to implied warranties or merchantability, or fitness for purpose, are excluded.

This document is intended only to assist the reader in the use of the product. ARM Limited shall not be liable for any loss or damage arising from the use of any information in this document, or any error or omission in such information, or any incorrect use of the product.

Confidentiality status

This document is Open Access (no restriction on distribution).

Product status

The information in this document is Final (information on a developed product).

Web address

`http://www.arm.com/`

Preface

About this document

This document is intended to help engineers who are designing ARM-based ASICs and PCBs to design systems that can be debugged using Multi-ICE.

Intended audience

This document has been written for experienced hardware engineers.

Typographical conventions

The following typographical conventions are used in this document:

bold highlights signal names within text.

italic highlights ARM-specific terminology, cross references and references to other publications.

`typewriter` identifies file and program names, source code, and text (such as commands) that may be entered at the keyboard.

`typewriter italic` identifies arguments to commands or functions which should be replaced by a specific value.

Related publications

The *Multi-ICE User Guide* (ARM DUI0048) describes how to use Multi-ICE and the Multi-ICE Debugger for Windows (MDW), and documents the TAPOp interface for third-party debuggers.

Application Note 31, *Using EmbeddedICE* (ARM DAI 0031), discusses some of the debugger issues from a software standpoint. This can be downloaded from the Documentation section of the ARM website.

Application Note 41, *TrackingICE* (ARM DAI 0041), describes TrackingICE. This can be downloaded from the Documentation section of the ARM website.

Further reading

IEEE Standard Test Access Port and Boundary Scan Architecture (IEEE Std 1149.1) describes the JTAG ports with which Multi-ICE communicates.

Feedback on this manual

If you have any comments or suggestions about this document, please send an email to `errata@arm.com` giving:

- the document title
- the document number
- the page number(s) to which your comments refer
- a concise explanation of your comments.

Feedback on this product

If you have any comments or suggestions about this product, please contact your supplier giving:

- the product name
- a concise explanation of your comments.

Contents
Multi-ICE System Design Considerations Application Note 72

Preface

Chapter 1 **Introduction**

Chapter 2 **System Design**

Chapter 3 **ASIC Guidelines**

Chapter 4 **PCB Guidelines**

Chapter 1
Introduction

This chapter contains:

1.1 Overview

This document is intended to help engineers, who are designing ARM-based ASICs and PCBs, to design systems that can be debugged using Multi-ICE.

This document discusses the following:

- The method of interconnecting multiple *Test Access Port* (TAP) Controllers in systems comprising more than one element that can be debugged via JTAG. For example, there might be several ARM processor cores, or an ARM core plus a *Digital Signal Processor* (DSP).
- Using the Multi-ICE adaptive clocking feature to control the JTAG clock rate.
- Reset signals, giving an example of devices to be used on a PCB.
- The connector pinout for the cable to the Multi-ICE interface unit.
- The interface electrical characteristics.

1.2 Multi-ICE

Multi-ICE is a system for debugging embedded processor cores via a JTAG interface. It is made up of four elements:

- Debug hardware within the ASIC, which in the case of ARM processors, is the EmbeddedICE macrocell.
- The Multi-ICE interface unit, which connects to the JTAG port on the target system, and to the parallel port on a host PC.
- Multi-ICE Server software running on the host PC, which manages the connection between the debug software tools and the Multi-ICE interface unit.
- Debug software tools, which communicate with the Server via the Multi-ICE *Dynamic Linked Library* (DLL).

Refer to Figure 1-1.

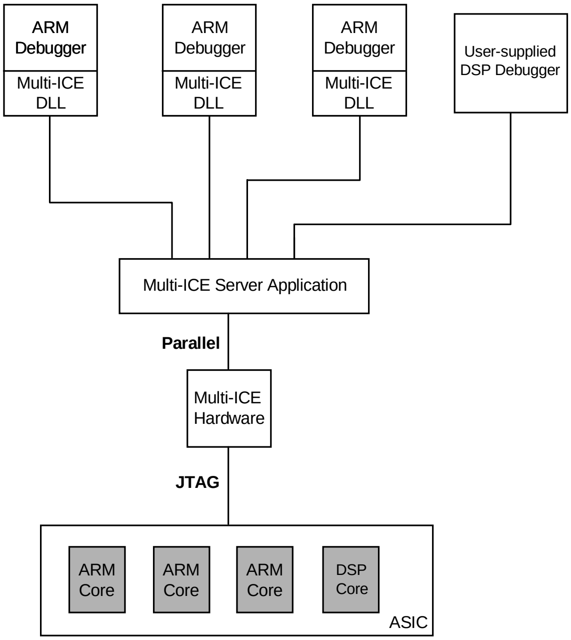


Figure 1-1 Elements of the Multi-ICE system

Chapter 2
System Design

This chapter contains:

2.1 Mixing ARM cores with other devices

Multi-ICE can be used to debug systems that mix ARM processor cores with other devices. The TAP Controllers should be daisy chained as described in *ASICs containing multiple devices* on page 3-2. When accessing a particular device, Multi-ICE places all other TAP Controllers in bypass mode, which only requires that Multi-ICE is informed of the length of the TAP Controller Instruction Register. For more information see *Configuring non-ARM devices* in the *Multi-ICE User Guide*.

Multi-ICE provides a software interface layer that facilitates the writing of user-supplied drivers for debugging non-ARM devices. For more information, refer to *Chapter 5, TAPOp Procedure Calls* in the *Multi-ICE User Guide*.

2.2 Using adaptive clocking to synchronize the JTAG port

Normal ARM-based ASICs using only hard macrocells, such as ARM7TDMI and ARM920T, can use the standard five wire JTAG interface (**TCK**, **TMS**, **TDI**, **TDO** and **nTRST**). Some target systems, however, require the JTAG actions to be synchronized to a clock within the system, in which case an extra signal is added to the JTAG port. For example, such synchronization is required in:

- an ASIC with single rising-edge D-type design rules, such as one based on an ARM7TDMI-S processor core
- a system where scan chains external to the ARM macrocell must meet single rising-edge D-type design rules
- a system using TrackingICE technology, where both processors must see inputs on the same clock edge (for more information on TrackingICE refer to Application Note 41, *TrackingICE*).

The adaptive clocking feature of Multi-ICE addresses this requirement. When adaptive clocking is enabled, Multi-ICE issues a **TCK** signal and waits for the **RTCK** (**R**eturned **TCK**) signal to come back. Multi-ICE does not progress to the next **TCK** until **RTCK** is received.

The same principle can also be used as an interface to targets with slow or widely varying clock frequency, such as battery-powered equipment that varies its clock speed according to processing demand. In such a system, the **TCK** could easily be hundreds of times faster than the system clock, and the debugger might lose synchronization with the target system. Adaptive clocking ensures that the JTAG port speed will, when necessary, automatically adapt to the slow system speed without compromising the programmed **TCK** rate that is to be used when the target is operating at full speed.

Figure 2-1 on page 2-4 illustrates a circuit for basic applications. The delay can be reduced by clocking the flip-flops from opposite edges of the system clock, because the second flip-flop is merely to provide better immunity to metastability problems. Note that even a single flip-flop synchronizer would never completely miss **TCK** events, because **RTCK** is part of a feedback loop controlling **TCK**.

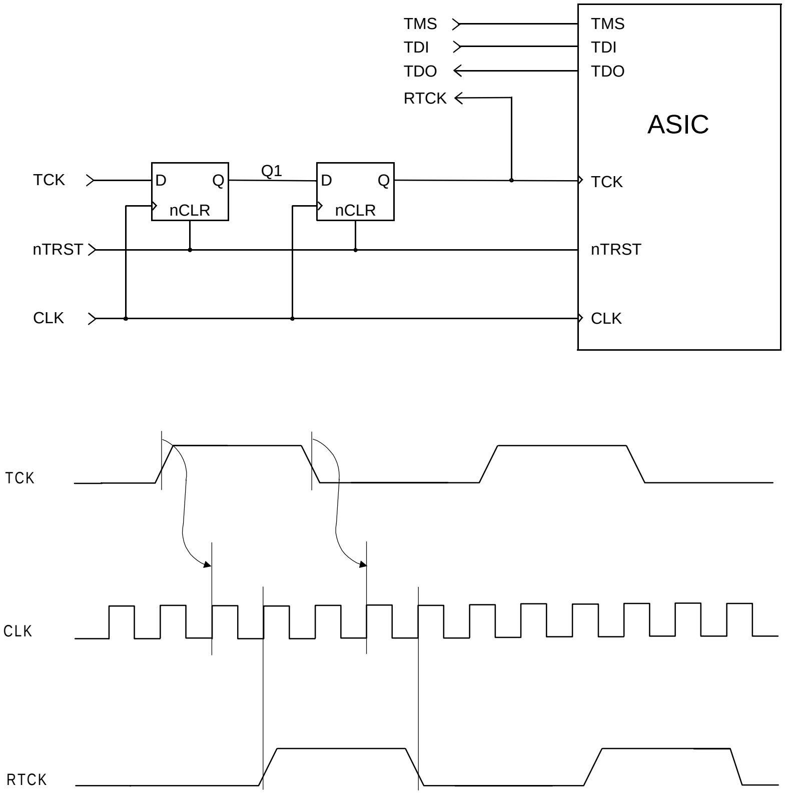


Figure 2-1 Basic JTAG port synchronizer

It is common for an ASIC design flow and its design rules to impose a restriction that all flip-flops in a design are clocked by one edge of a single clock. To interface this to a JTAG port that is completely asynchronous to the system, it is necessary to convert the JTAG **TCK** events into clock enables for this single clock, and to ensure that the JTAG port cannot overrun this synchronization delay.

Figure 2-2 on page 2-5 illustrates a circuit for such a system. **TCKFallingEn** and **TCKRisingEn** are each active for exactly one period of **CLK**, and enable the JTAG actions that should take place on the falling and rising edges, respectively, of **TCK**.

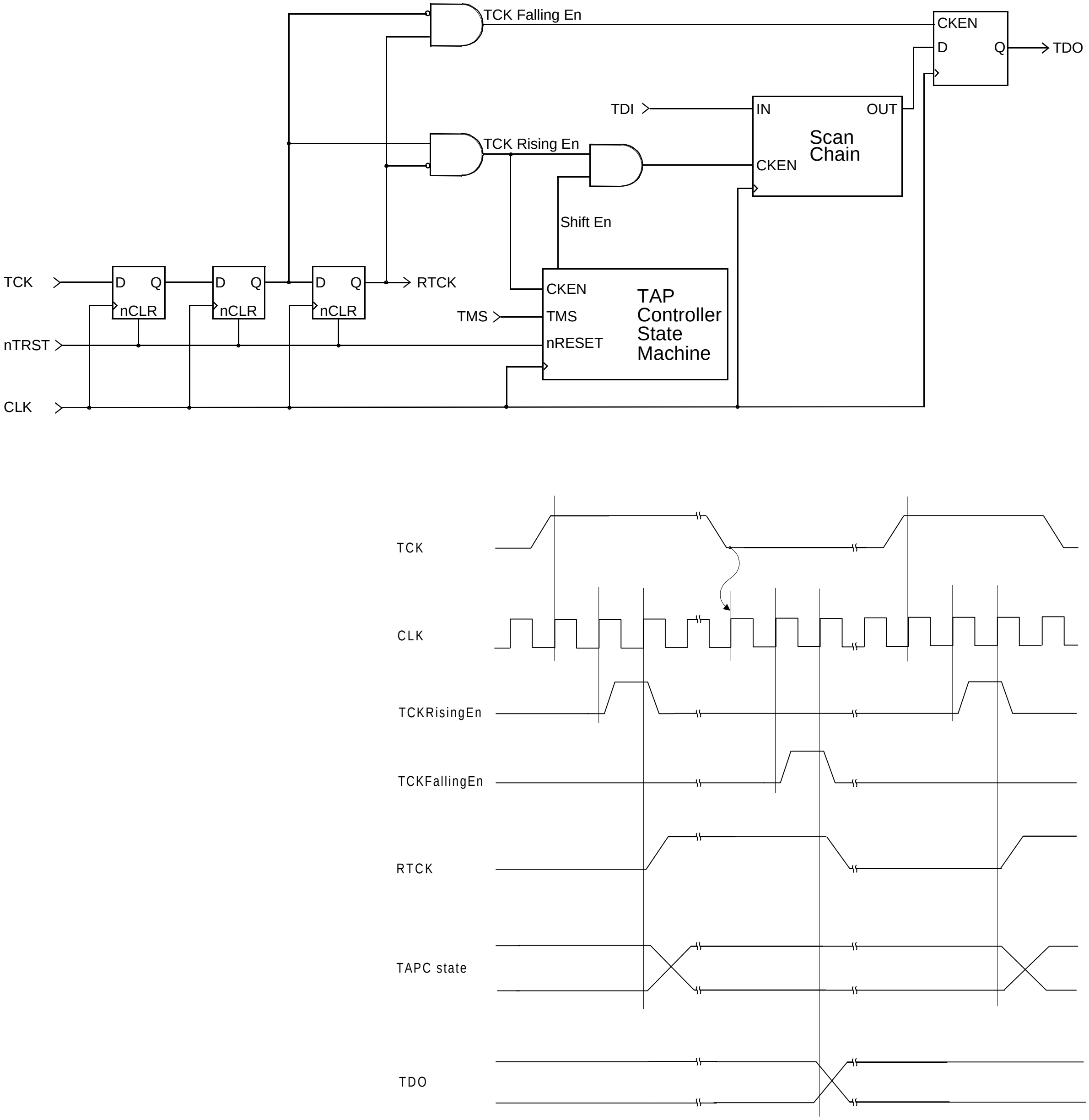


Figure 2-2 JTAG port synchronizer for single rising-edge D-type ASIC design rules

2.3 Reset signals

This section describes the following reset signals:

- ARM reset signals
- Multi-ICE reset signals.

2.3.1 ARM reset signals

All ARM cores have a main processor reset called **nRESET** (or **BnRES**, when there is an AMBA ASB bus interface included with the core). In a typical system, this should be activated by the following conditions:

- power on
- manual push button
- remote reset from the debugger (for example, via Multi-ICE)
- watchdog circuit (if appropriate to the application).

In addition, any ARM core that includes the EmbeddedICE macrocell will have a JTAG interface that has a second reset input called **nTRST**. This resets the EmbeddedICE macrocell, the TAP controller, and the boundary scan cells. In a typical system, this should be activated by the following conditions:

- power on
- remote JTAG reset (for example, from Multi-ICE).

2.3.2 Multi-ICE reset signals

The Multi-ICE interface unit has two reset signals connected to the debug target board:

1. **nTRST** is intended to drive the JTAG **nTRST** signal on the ARM. It is an open collector output that is activated whenever the Multi-ICE software needs to re-initialize the debug interface in the target system. The target board should include a pull-up resistor, and will typically OR this with an on-board power-on reset signal.

2. **nSRST** is a bidirectional signal that is intended both to activate, and to sense the system reset signal on the target board. The open collector output can be driven low by the debugger to re-initialize the target system. If the target system generates a reset due either to a power-on condition, or the user activating a manual push-button, Multi-ICE senses this and reports the condition back to the debugger. The target board should include a pull-up resistor.

2.3.3 Example reset circuits

The circuits shown in Figure 2-3 and Figure 2-4 illustrate how the desired behavior can be achieved.

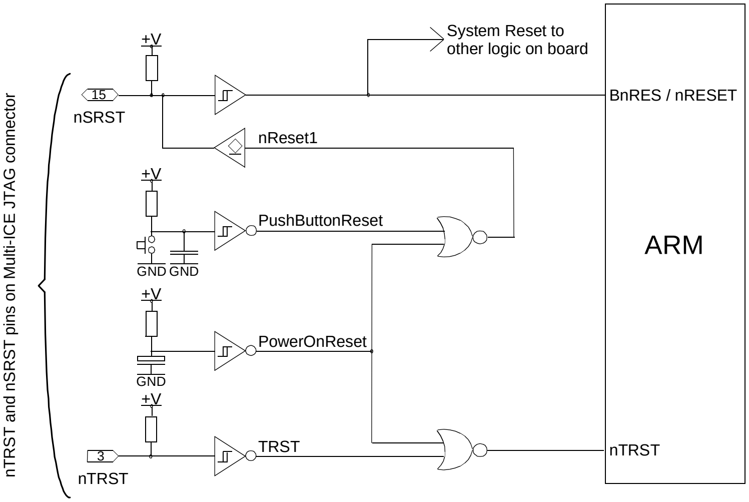


Figure 2-3 Example reset circuit logic

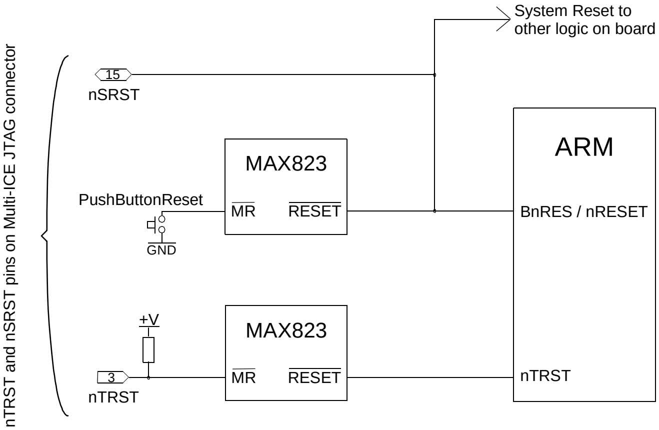


Figure 2-4 Example reset circuit using power supply monitor ICs

The MAX823 is a typical power supply supervisor IC. It has a current limited **/RESET** output that can be overdriven by the Multi-ICE unit.

When Multi-ICE senses a reset condition, it latches the state to inform the debugger that a reset has occurred at some time. This is particularly useful if the target system were utilizing the MAX823's watchdog circuit (which can reset the system after a defined period of inactivity) because this might be the only evidence that such a reset had occurred.

Chapter 3
ASIC Guidelines

This chapter contains:

3.1 ASICs containing multiple devices

The JTAG standard originally described *daisy-chaining* multiple devices on a PCB. This concept is now extended to multiple cores within a single package. If more than one JTAG *Test Access Port* (TAP) Controller is present within your ASIC, they must all be serially chained so that Multi-ICE can communicate with all of them simultaneously.

There are a few possible configurations of multiple TAP Controllers, and these are listed below:

1. TAP controllers serially chained within the ASIC. This is the natural extension of the JTAG board level interconnection, and is the one recommended for use with Multi-ICE. There is no increase in package pin count, and only a very small impact on speed (because unaddressed TAP Controllers can be put into bypass mode). Refer to Figure 3-1.

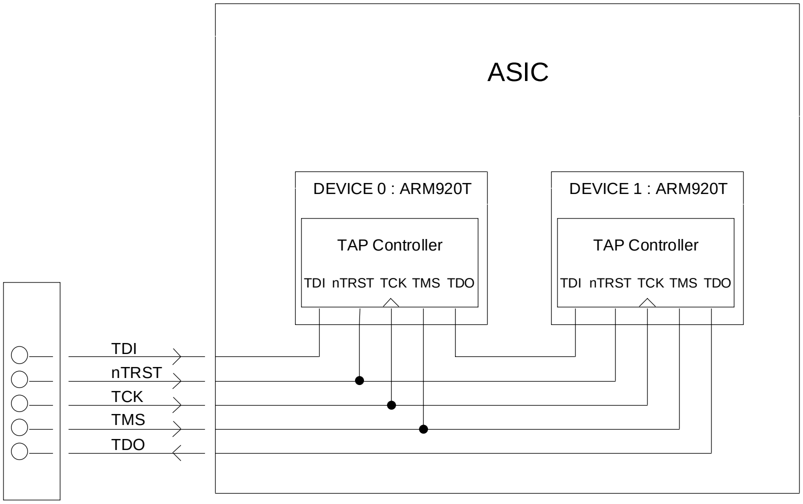


Figure 3-1 TAP Controllers serially chained in an ASIC

2. Each set of JTAG connections is pinned out separately. This gives the greatest flexibility on the PCB, but at the cost of many pins on the device package. If this method is chosen to simplify device testing, the JTAG ports should be serially chained on the PCB when Multi-ICE is to be used. The separate JTAG ports can be tracked to separate headers on the PCB, but this then needs one Multi-ICE Interface Unit per header, and is unnecessary. Refer to Figure 3-2 on page 3-3.

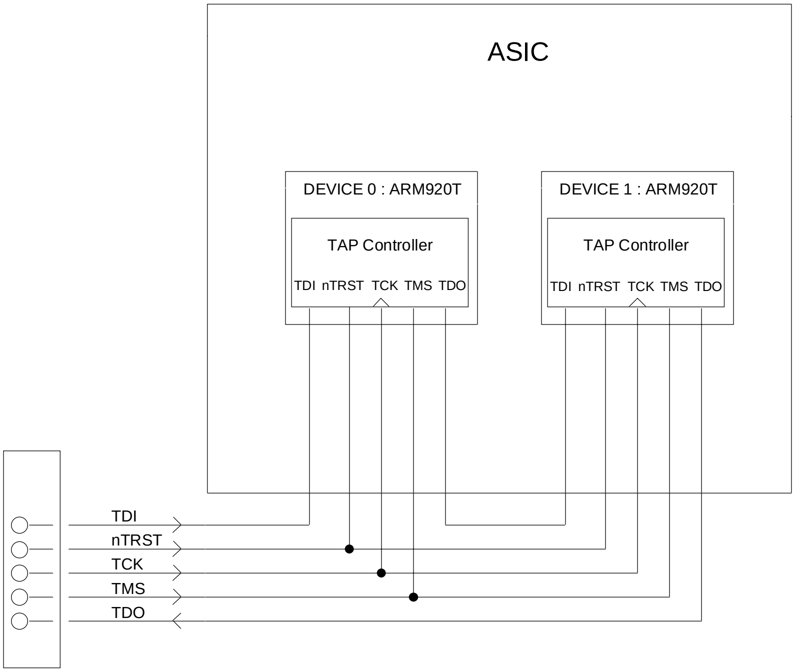


Figure 3-2 TAP Controllers serially chained outside an ASIC

3. Multiplexing of data signals. It is possible to provide some means of gating **TCK** or **TMS** signals so that only one TAP Controller is active at once, and then to multiplex their data outputs so that the JTAG debug interface sees a single device.

Note

There is no standard definition of how to build or control such circuitry, and this method is not supported by Multi-ICE.

3.1.1 Boundary scan test vectors

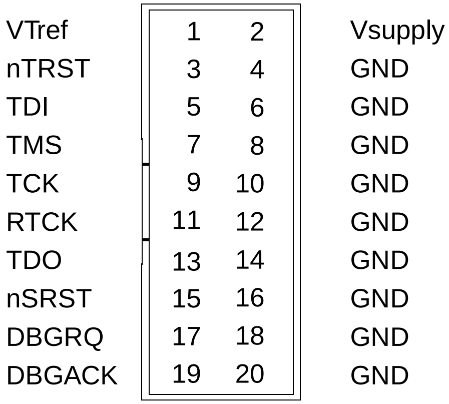
If the JTAG Boundary Scan test methodology is used to apply production test vectors, it may be desirable to have independent external access to each TAP controller. This avoids the need to merge test vectors for more than one block in the device. One solution to this is to adopt a hybrid of interconnection methods 1 and 2 (see *ASICs containing multiple devices* on page 3-2). Typically, there will be a pin on the package that switches elements of the device into a test mode. This can be used to break the internal daisy chaining of **TDO**/**TDI** signals, and to multiplex out independent JTAG ports on pins that will be used for another purpose during normal operation.

Chapter 4
PCB Guidelines

This chapter contains:

4.1 Multi-ICE JTAG header connector

This section displays the JTAG pin connections, and describes each signal.



VTref	1	2	Vsupply
nTRST	3	4	GND
TDI	5	6	GND
TMS	7	8	GND
TCK	9	10	GND
RTCK	11	12	GND
TDO	13	14	GND
nSRST	15	16	GND
DBGRQ	17	18	GND
DBGACK	19	20	GND

Figure 4-1 JTAG pin connections

The connector is a 20-way, 2.54mm pitch pin header that mates with an IDC socket mounted on a ribbon cable.

Note

All GND pins should be connected to 0V on the target board.

4.1.1 Signal descriptions

Table 4-1 lists each pin number and its corresponding signal name and description.

Table 4-1 Pin descriptions

Pin number	Target direction	Name	Description
Pin 1	Output	**VTref**	This is the target reference voltage. It indicates that the target has power and it is also used to create the logic-level reference for the input comparators on **TDO** and **RTCK**. It also controls the output logic levels to the target. It is normally fed from **Vdd** on the target board.
Pin 2	Output	**Vsupply**	This is the supply voltage to Multi-ICE. It draws its supply current from this pin via a step-up voltage converter. This is normally fed by the target **Vdd** which *must not* have a series resistor in the feed to this pin. If the target supply voltage or its current capability is too low, this path is broken by an external power jack on an inline header.
Pin 3	Input	**nTRST**	Open collector output from Multi-ICE to the Reset signal on the target JTAG port. This pin should be pulled high on the target to avoid unintentional resets when there is no connection.
Pin 4		**GND**	
Pin 5	Input	**TDI**	Test Data In signal from Multi-ICE to the target JTAG port. It is recommended that this pin is pulled to a defined state.
Pin 6		**GND**	
Pin 7	Input	**TMS**	Test Mode Select signal from Multi-ICE to the target JTAG port. This pin should be pulled up on the target so that the effect of any spurious **TCK**s when there is no connection is benign.
Pin 8		**GND**	
Pin 9	Input	**TCK**	Test Clock signal from Multi-ICE to the target JTAG port. It is recommended that this pin is pulled to a defined state.
Pin 10		**GND**	

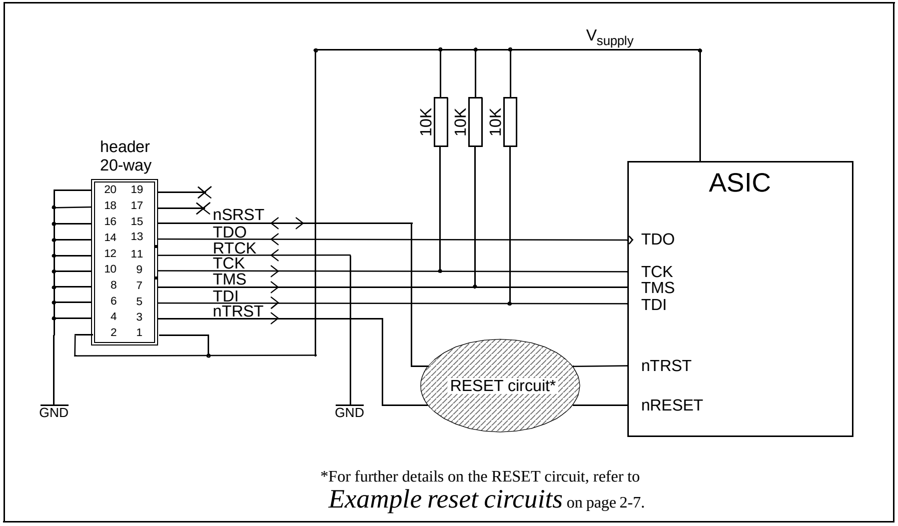
Table 4-1 Pin descriptions (continued)

Pin 11	Output	**RTCK**	Return Test Clock signal from the target JTAG port to Multi-ICE. Some targets, such as ARM7TDMI-S or ASICs using TrackingICE technology, need to synchronize the JTAG port to internal clocks. To assist in meeting this requirement, a returned (and re-timed) **TCK** can be used to dynamically control the **TCK** rate. Multi-ICE provides Adaptive Clock Timing, which waits for **TCK** changes to be echoed correctly before making further changes.
Pin 12		**GND**	
Pin 13	Output	**TDO**	Test Data Out from the target JTAG port to Multi-ICE.
Pin 14		**GND**	
Pin 15	Input/Output	**nSRST**	Open collector output from Multi-ICE to the target system reset. This is also an input to Multi-ICE so that a reset initiated on the target may be reported to the debugger. This pin should be pulled up on the target to avoid unintentional resets when there is no connection.
Pin 16		**GND**	
Pin 17	Input	DBGRQ	This pin is not connected in the Multi-ICE unit. It is reserved for compatibility with other equipment where it is used as a debug request signal to the target system.
Pin 18		**GND**	
Pin 19	Output	DBGACK	This pin is not connected in the Multi-ICE unit. It is reserved for compatibility with other equipment where it is used as a debug acknowledge signal from the target system.
Pin 20		**GND**	

4.2 PCB connections

It is recommended that the JTAG header is placed as closely as possible to the Target Device, as this will minimize any possible signal degradation due to long PCB tracks.

Figure 4-2 shows the layout of possible PCB connections.



*For further details on the RESET circuit, refer to *Example reset circuits* on page 2-7.

Figure 4-2 Typical PCB connections

4.3 Target interface logic levels

Multi-ICE is designed to interface with a wide range of target system logic levels. It does so by adapting its output drive and input threshold to a reference voltage supplied by the target system.

VTref (pin1 on the JTAG header connector) feeds the reference voltage to the Multi-ICE Interface Unit. This voltage, clipped at approximately 3.3V, is used as the output high voltage (V_{oh}) for logic 1s (ones) on **TCK**, **TDI** and **TMS**. 0V is used as the output low voltage for logic 0s (zeroes). The input logic threshold voltage ($V_{i(th)}$) for the **TDO**, **RTCK** and **nSRST** inputs is 50% of the V_{oh} level (so is clipped to ≈1.65V). The relationships of V_{oh} and $V_{i(th)}$ to **VTref** are illustrated in Figure 4-3.

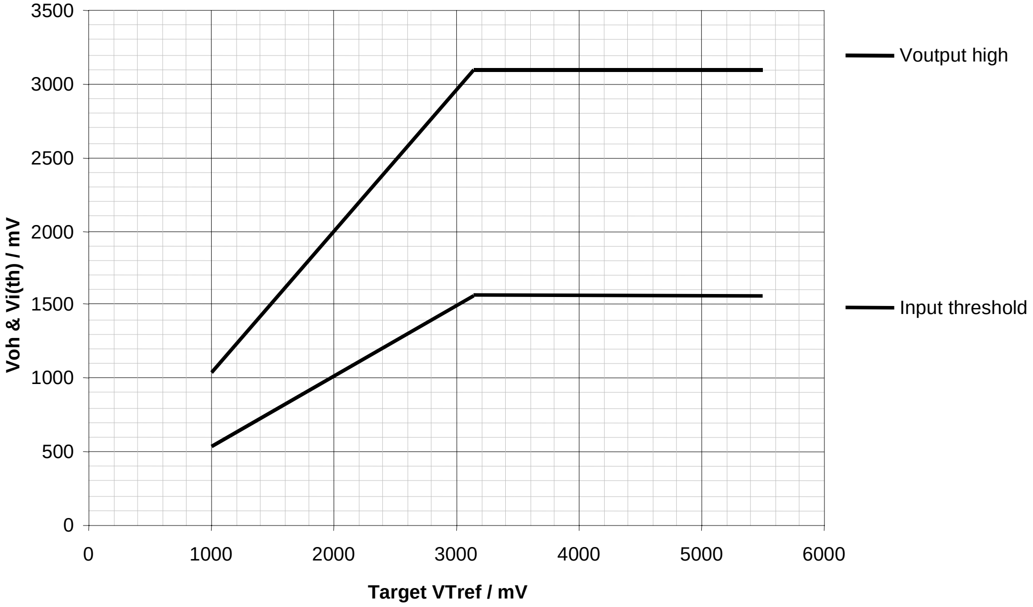


Figure 4-3 Target interface voltage levels

The adaptive interface levels work down to **VTref** below 1V. If, however **VTref** goes below approximately 0.85V, Multi-ICE interprets this condition as *Target Not Present*, and the software will report this error condition.

The **nTRST** output from Multi-ICE is effectively driven *open collector*. In other words, it is actively pulled to 0V but relies on a pull-up resistor within the target system to end the reset state. This is because it is common to wire-OR this signal with another source of **nTRST**, such as power-on reset in the target system.

The **nSRST** output from Multi-ICE is similarly driven open collector, and should be pulled-up with a resistor in the target system. As this signal is also an input to the Multi-ICE Interface Unit, there is a large-value internal pull-up resistor (51kΩ to Voh), which is to avoid spurious lows on the input when **nSRST** is not connected to the target system.

The input and output characteristics of the Multi-ICE Interface Unit are compatible with logic levels from TTL-compatible, or CMOS logic in target systems. For information when assessing compatibility with other logic systems, the output impedance of all signals is approximately 100Ω.

Chapter 5
Compatibility with EmbeddedICE Interface Target Connectors

The EmbeddedICE Interface Unit uses a 14-way connector for the interface to the target system. ARM provide adaptor boards for each direction so that Multi-ICE Interface Units can be connected to older target boards with 14-way connectors, and EmbeddedICE Interface Units can be connected to more recent boards with 20-way connectors.

This chapter contains:

- *Adaptor to connect a Multi-ICE interface unit to 14-way connectors* on page 5-2
- *Adaptor to connect an EmbeddedICE interface unit to 20-way connectors* on page 5-3.

5.1 Adaptor to connect a Multi-ICE interface unit to 14-way connectors

Each Multi-ICE Interface Unit is supplied with an HBI-0027B adaptor board. The 14-way socket on the adaptor board plugs into a target board with the older header, and the Multi-ICE ribbon cable is connected to the 20-way pin header on the adaptor board.

The three-pin header J3 has the following connections:

- Pin 1 **0V**
- Pin 2 Multi-ICE connector pin 2, **Vsupply**
- Pin 3 Target connector pin 1, **SPU**.

The jumper link supplied on the adaptor board connects pins 2 and 3 so that the Multi-ICE Interface Unit draws its power from the target system in the normal manner. If the target system cannot source a suitable voltage or current, an external 2V—5V DC supply can be connected to pins 1 and 2.

The three-pin header J4 has the following connections:

- Pin 1 **0V**
- Pin 2 Multi-ICE connector pin 11, **RTCK**
- Pin 3 Resistor fed by Multi-ICE connector pin 9, **TCK**.

The jumper link supplied on the adaptor board connects **0V** back to the Multi-ICE **RTCK** input. If the target system is to use Adaptive Clocking, **TCK** can be tapped off here, and the synchronized version used to clock the target can be fed back in as **RTCK**.

5.2 Adaptor to connect an EmbeddedICE interface unit to 20-way connectors

Each new EmbeddedICE Interface Unit is supplied with an HBI-0028A adaptor board. This allows EmbeddedICE Interface Units to be used with target system boards that use a Multi-ICE-compatible 20-way header.

The basic JTAG signals are connected straight through to the appropriate pins. The $\mathbf{V_{supply}}$ signal from the target (pin 2 on the 20-way target header) is not connected because the EmbeddedICE Interface Unit uses a separate DC power supply. **RTCK** from the target is not routed through because the EmbeddedICE Interface Unit does not support adaptive clocking. The EmbeddedICE Interface Unit does not support sensing of System Reset events originated on the target system.

Note

While this adaptor allows connection of an EmbeddedICE Interface Unit to any board, the ARM debugger software for use with the EmbeddedICE Interface Unit only supports the debugging of ARM7TDMI and ARM70DI based devices. Other ARM target devices are supported by Multi-ICE.