ARM® DS-5™

Version 5

ARM RVI™ System and Interface Design Reference

ARM DUI 0517H (ID092812)

ARM DS-5
ARM RVI System and Interface Design Reference

Release Information

The following changes have been made to this book.

Change history

Date	Issue	Confidentiality	Change
May 2010	A	Non-confidential	First release
November 2010	B	Non-confidential	Second release
30 April 2011	C	Non-Confidential	DSTREAM and RVI v4.2.1 Release
29 July 2011	D	Non-Confidential	Update 1 for DSTREAM and RVI v4.2.1
30 September 2011	E	Non-Confidential	DSTREAM and RVI v4.4 Release
29 February 2012	F	Non-Confidential	Update 1 for DS-5
29 July 2012	G	Non-Confidential	Update 2 for DS-5
12 October 2012	H	Non-Confidential	Update 3 for DS-5

Product Status

The information in this document is final, that is for a developed product.

Web Address

`http://www.arm.com`

Conformance Notices

This section contains conformance notices.

Federal Communications Commission Notice

This device is test equipment and consequently is exempt from part 15 of the FCC Rules under section 15.103 (c).

CE Declaration of Conformity

The system should be powered down when not in use.

It is recommended that ESD precautions be taken when handling ARM RVI equipment.

The ARM RVI module generates, uses, and can radiate radio frequency energy and may cause harmful interference to radio communications. There is no guarantee that interference will not occur in a particular installation. If this equipment causes harmful interference to radio or television reception, which can be determined by turning the equipment off or on, you are encouraged to try to correct the interference by one or more of the following measures:

- ensure attached cables do not lie across the target board
- reorient the receiving antenna
- increase the distance between the equipment and the receiver
- connect the equipment into an outlet on a circuit different from that to which the receiver is connected
- consult the dealer or an experienced radio/TV technician for help

Note

It is recommended that wherever possible shielded interface cables be used.

Contents

ARM DS-5 ARM RVI System and Interface Design Reference

Chapter 1 Conventions and feedback

Chapter 2 RVI Debug Unit System Design Guidelines

2.1 Using adaptive clocking to synchronize the JTAG port 2-2
2.2 Reset signals 2-5
2.3 ARM reset signals 2-6
2.4 RVI reset signals 2-7
2.5 Example reset circuits 2-8
2.6 ASIC guidelines 2-10
2.7 ICs containing multiple devices 2-11
2.8 Boundary scan test vectors 2-12
2.9 PCB guidelines 2-13
2.10 PCB connections 2-14
2.11 Target interface logic levels 2-15
2.12 JTAG signal integrity and maximum cable lengths 2-17

Chapter 3 JTAG Interface Connections

3.1 JTAG interface pinouts 3-2
3.2 JTAG interface signals 3-3
3.3 JTAG port timing characteristics 3-6

Chapter 4 User I/O Connections

4.1 The RVI User I/O connector 4-2

Chapter 5 Serial Wire Debug

5.1 Target interface 5-2

5.2 SWD timing requirements 5-3

Chapter 1
Conventions and feedback

The following describes the typographical conventions and how to give feedback:

Typographical conventions

The following typographical conventions are used:

`monospace` Denotes text that can be entered at the keyboard, such as commands, file and program names, and source code.

`monospace` Denotes a permitted abbreviation for a command or option. The underlined text can be entered instead of the full command or option name.

`monospace italic`

Denotes arguments to commands and functions where the argument is to be replaced by a specific value.

`monospace bold`

Denotes language keywords when used outside example code.

italic Highlights important notes, introduces special terminology, denotes internal cross-references, and citations.

bold Highlights interface elements, such as menu names. Also used for emphasis in descriptive lists, where appropriate, and for ARM® processor signal names.

Feedback on this product

If you have any comments and suggestions about this product, contact your supplier and give:

- your name and company

- the serial number of the product
- details of the release you are using
- details of the platform you are using, such as the hardware platform, operating system type and version
- a small standalone sample of code that reproduces the problem
- a clear explanation of what you expected to happen, and what actually happened
- the commands you used, including any command-line options
- sample output illustrating the problem
- the version string of the tools, including the version number and build numbers.

Feedback on content

If you have comments on content then send an e-mail to `errata@arm.com`. Give:

- the title
- the number, ARM DUI 0517H
- if viewing online, the topic names to which your comments apply
- if viewing a PDF version of a document, the page numbers to which your comments apply
- a concise explanation of your comments.

ARM also welcomes general suggestions for additions and improvements.

ARM periodically provides updates and corrections to its documentation on the ARM Information Center, together with knowledge articles and *Frequently Asked Questions* (FAQs).

Other information

- ARM Information Center, `http://infocenter.arm.com/help/index.jsp`
- ARM Technical Support Knowledge Articles, `http://infocenter.arm.com/help/topic/com.arm.doc.faqs/index.html`
- ARM Support and Maintenance, `http://www.arm.com/support/services/support-maintenance.php`
- ARM Glossary, `http://infocenter.arm.com/help/topic/com.arm.doc.aeg0014-/index.html`.

Chapter 2
RVI Debug Unit System Design Guidelines

The following topics provide information on developing ARM® architecture-based devices and *Printed Circuit Boards* (PCBs) that can be debugged using ARM RVI™:

- *Using adaptive clocking to synchronize the JTAG port* on page 2-2
- *Reset signals* on page 2-5
- *ARM reset signals* on page 2-6
- *RVI reset signals* on page 2-7
- *Example reset circuits* on page 2-8
- *ASIC guidelines* on page 2-10
- *ICs containing multiple devices* on page 2-11
- *Boundary scan test vectors* on page 2-12
- *PCB guidelines* on page 2-13
- *PCB connections* on page 2-14
- *Target interface logic levels* on page 2-15
- *JTAG signal integrity and maximum cable lengths* on page 2-17.

2.1 Using adaptive clocking to synchronize the JTAG port

ARM architecture-based devices using only hard macrocells, for example ARM7TDMI® and ARM920T, use the standard five-wire JTAG interface (**TCK**, **TMS**, **TDI**, **TDO**, and **nTRST**). Some target systems, however, require that JTAG events are synchronized to a clock in the system. To handle this case, an extra signal (**RTCK**) is included on the JTAG port. For example, this synchronization is required in:

- an *Application-Specific Integrated Circuit* (ASIC) with single rising-edge D-type design rules, such as one based on an ARM7TDMI-S™ processor
- a system where scan chains external to the ARM macrocell must meet single rising-edge D-type design rules.

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

Note

- Adaptive clocking is automatically configured in ARM DS-5™ as required by the target.
- If you use the adaptive clocking feature, transmission delays, gate delays, and synchronization requirements result in a lower maximum clock frequency than with non-adaptive clocking. Do not use adaptive clocking unless it is required by the hardware design.
- If, when autoconfiguring a target, the RVI run control unit receives pulses on **RTCK** in response to **TCK** it assumes that adaptive clocking is required, and enables adaptive clocking in the target configuration. If the hardware does not require adaptive clocking, the target is driven slower than it could be. You can disable adaptive clocking using controls on the JTAG settings dialog box.
- If adaptive clocking is used, RVI cannot detect the clock speed, and therefore cannot scale its internal timeouts. If the target clock frequency is very slow, a JTAG timeout might occur. This leaves the JTAG in an unknown state, and RVI cannot operate correctly without reconnecting to the processor. JTAG timeouts are enabled by default. You can disable JTAG timeouts by deselecting the option JTAG Timeouts Enabled in the Debug Hardware Config utility.

You can use adaptive clocking 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 this system, **TCK** might be hundreds of times faster than the system clock, and the debugger loses synchronization with the target system. Adaptive clocking ensures that the JTAG port speed automatically adapts to slow system speed.

The following figure shows a circuit for a basic JTAG port synchronizer.

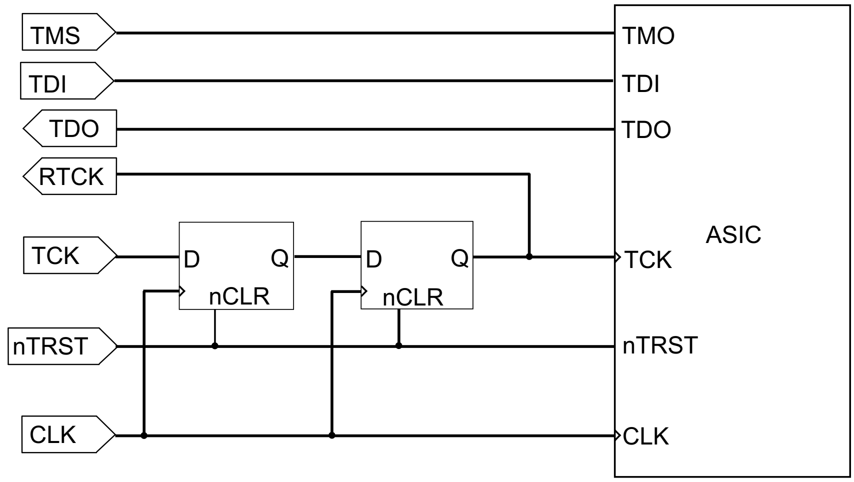

Figure 2-1 Basic JTAG port synchronizer

The following figure shows a partial timing diagram for the basic JTAG synchronizer. The delay can be reduced by clocking the flip-flops from opposite edges of the system clock, because the second flip-flop only provides better immunity to metastability problems. Even a single flip-flop synchronizer never completely misses **TCK** events, because **RTCK** is part of a feedback loop controlling **TCK**.

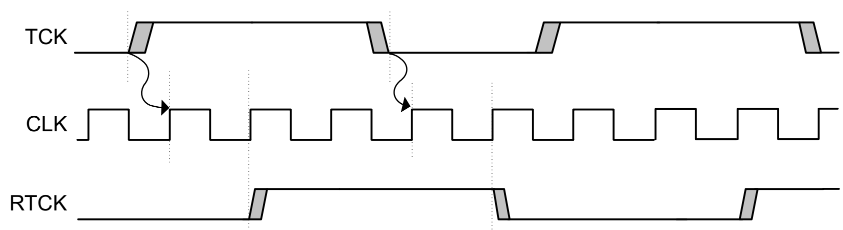

Figure 2-2 Timing diagram for the Basic JTAG synchronizer

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.

The following figure shows one possible implementation of this circuit.

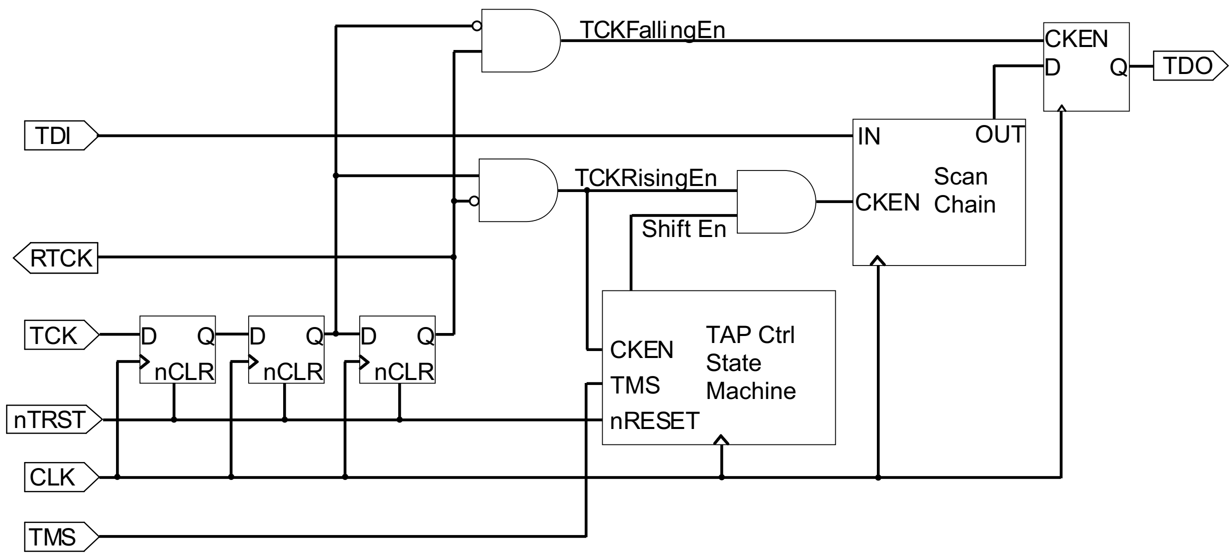

Figure 2-3 JTAG port synchronizer for single rising-edge D-type ASIC design rules

The following figure shows a corresponding partial timing diagram, and how **TCKFallingEn** and **TCKRisingEn** are each active for exactly one period of **CLK**. It also shows how these enable signals gate the **RTCK** and **TDO** signals so that they only change state at the edges of **TCK**.

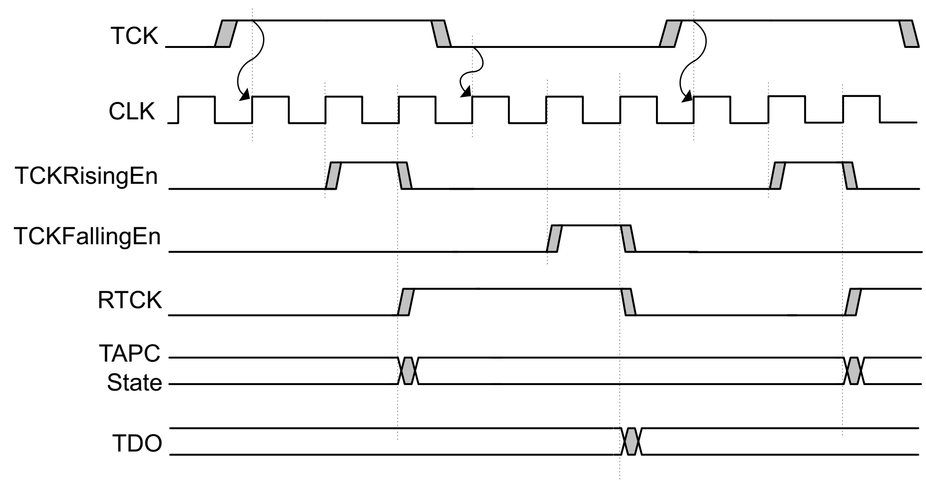

Figure 2-4 Timing diagram for the D-type JTAG synchronizer

2.1.1 See also

Concepts

- *Reset signals* on page 2-5
- *ASIC guidelines* on page 2-10
- *PCB guidelines* on page 2-13
- *JTAG signal integrity and maximum cable lengths* on page 2-17.

Reference

- Chapter 3 *JTAG Interface Connections*
- Chapter 5 *Serial Wire Debug*

2.2 Reset signals

The reset signals that are available on ARM devices, and how RVI expects them to be wired, are described in the following topics:

- *ARM reset signals* on page 2-6
- *RVI reset signals* on page 2-7
- *Example reset circuits* on page 2-8.

2.2.1 See also

Concepts

- *Using adaptive clocking to synchronize the JTAG port* on page 2-2
- *ASIC guidelines* on page 2-10
- *PCB guidelines* on page 2-13
- *JTAG signal integrity and maximum cable lengths* on page 2-17.

Reference

- Chapter 3 *JTAG Interface Connections*
- Chapter 5 *Serial Wire Debug*

2.3 ARM reset signals

All ARM processors have a main processor reset that might be called **nRESET**, **BnRES**, or **HRESET**. This is asserted by one or more of these conditions:

- power on
- manual push button
- remote reset from the debugger (using RVI)
- watchdog circuit (if appropriate to the application).

Any ARM processor including the JTAG interface has a second reset input called **nTRST** (TAP Reset). This resets the EmbeddedICE® logic, the TAP controller, and the boundary scan cells. It is activated by a remote JTAG reset.

It is strongly recommended that both signals are separately available on the JTAG connector. If the **nRESET** and **nTRST** signals are linked together, resetting the system also resets the TAP controller. This means that:

- it is not possible to debug a system from reset, because any breakpoints previously set are lost
- you might have to start the debug session from the beginning, because RVI does not recover when the TAP controller state is changed.

2.3.1 See also

Concepts

- *RVI reset signals* on page 2-7
- *Example reset circuits* on page 2-8.

2.4 RVI reset signals

The RVI run control unit has two reset signals connected to the debug target hardware:

- **nTRST** drives the JTAG **nTRST** signal on the ARM processor. It is an output that is activated whenever the RVI debug software has to re-initialize the debug interface in the target system.
- **nSRST** is a bidirectional signal that both drives and senses the system reset signal on the target. The output is driven LOW by the debugger to re-initialize the target system.

The target hardware must include a pull-up resistor on both reset signals. In the RVI unit, the strong pull-up/pull-down resistance is approximately 100Ω, and the weak pull-up/pull-down resistance is approximately 4.7kΩ. Because you can select the drive strength for **nTRST** and **nSRST**, target assemblies with a variey of different reset configurations can be supported.

2.4.1 See also

Concepts

- *ARM reset signals* on page 2-6
- *Example reset circuits* on page 2-8.

Reference

Using the Debug Hardware Configuration Utilities:

- *Configuring the debug hardware Advanced settings* on page 5-46.

2.5 Example reset circuits

The circuits shown in the following figures illustrate how the behavior relating to ARM reset signals and RVI reset signals can be achieved.

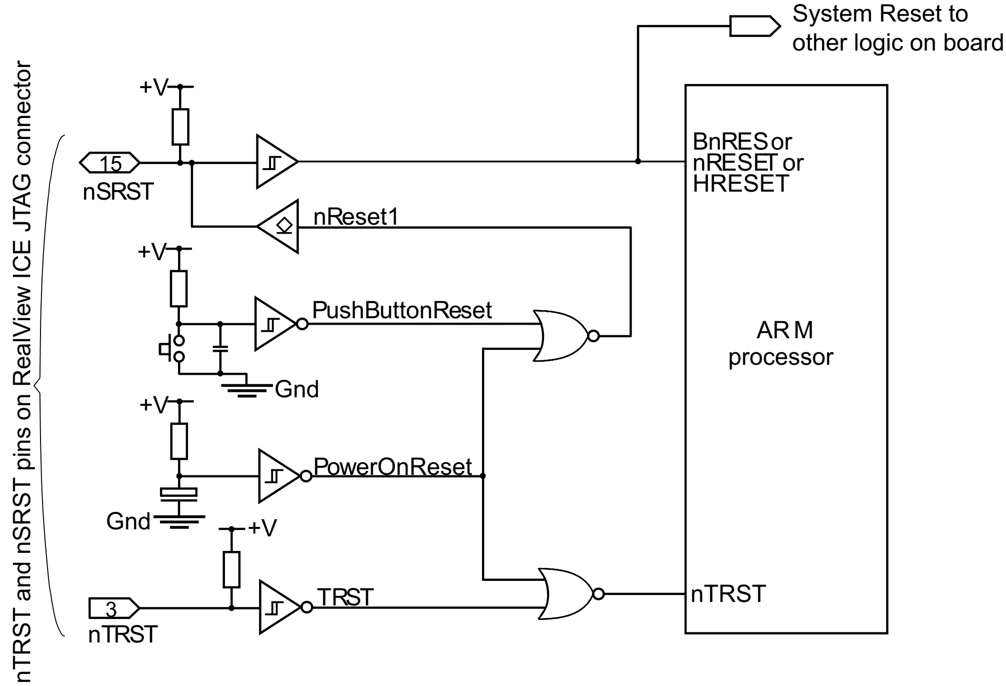

Figure 2-5 Example reset circuit logic

——— **Note** ———

The symbol for nReset1 signifies a sink driver, with an open collector output.

———————————

The MAX823 used in the following figure is a typical power supply supervisor. It has a current limited **nRESET** output that can be overdriven by the RVI unit.

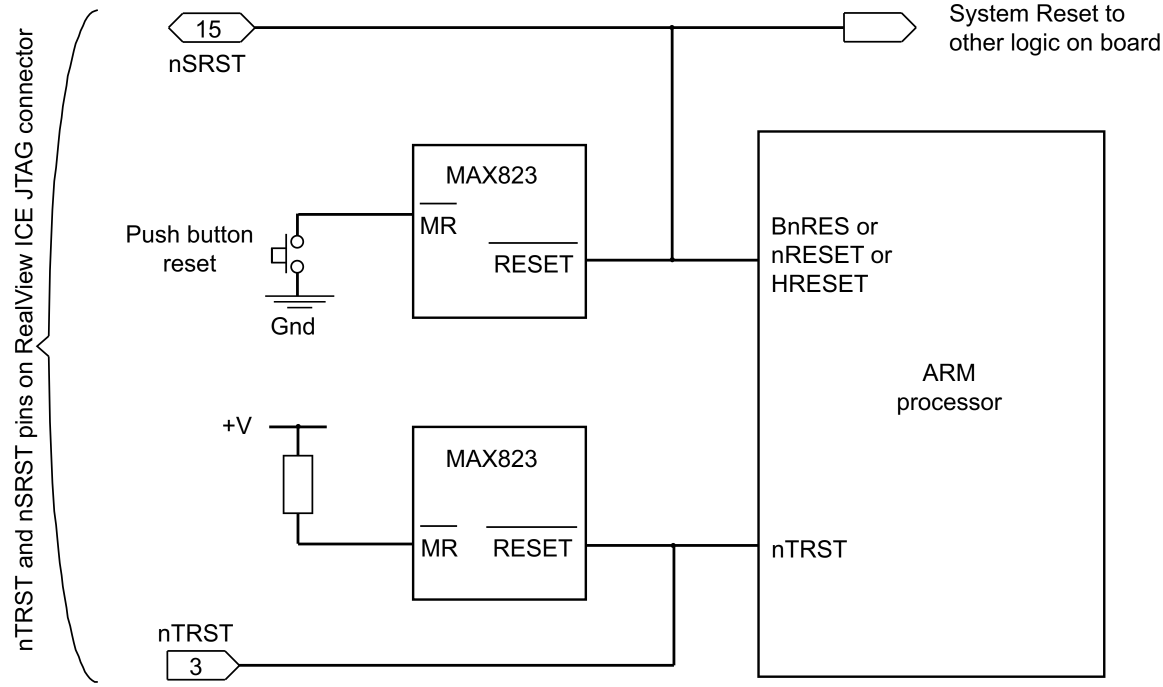

Figure 2-6 Example reset circuit using power supply monitor ICs

2.5.1 See also

Concepts

- *ARM reset signals* on page 2-6
- *RVI reset signals* on page 2-7.

2.6 ASIC guidelines

ASIC guidelines are described in the following topics:

- *ICs containing multiple devices* on page 2-11
- *Boundary scan test vectors* on page 2-12.

2.7 ICs containing multiple devices

If your ASIC contains multiple devices that have a JTAG TAP controller, you must serially chain them so that RVI can communicate with all of them simultaneously. The chaining can either be within the ASIC, or externally.

Note

There is no support in RVI for multiplexing **TCK**, **TMS**, **TDI**, **TDO**, and **RTCK** between a number of different processors.

2.7.1 TAP controllers serially chained within the ASIC

The JTAG standard originally described serially chaining multiple devices on a PCB. This concept can be extended to serially chaining multiple TAP controllers within an ASIC. This configuration does not increase the package pin count. It does increase JTAG propagation delays, but this impact can be small if you put unaddressed TAP controllers into bypass mode, as shown in the following figure:

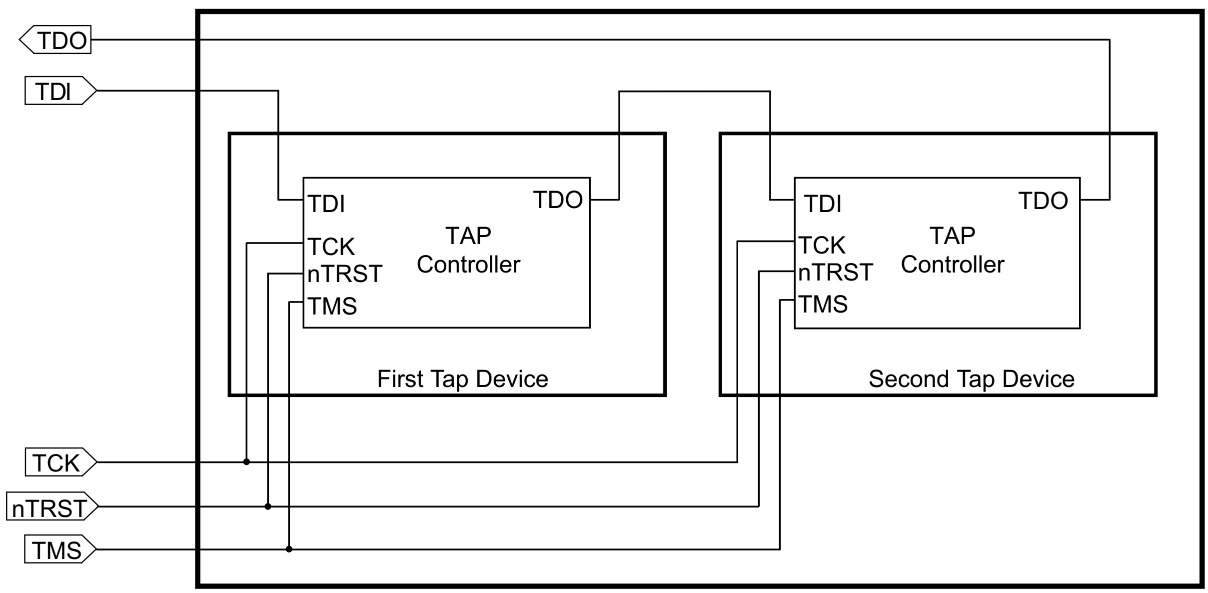

Figure 2-7 TAP Controllers serially chained within an ASIC

2.7.2 TAP controllers serially chained externally

You can use separate pins on the ASIC for each JTAG port, and serially chain them externally (for example on the PCB). This configuration can simplify device testing, and gives the greatest flexibility on the PCB. However, this is at the cost of many pins on the device package.

2.7.3 See also

Concepts

- *Boundary scan test vectors* on page 2-12.

Other information

- *CoreSight Technology System Design Guide*, `http://infocenter.arm.com/help/topic/com.arm.doc.dgi0012-/index.html`

2.8 Boundary scan test vectors

If you use the JTAG boundary scan test methodology to apply production test vectors, you might want to have independent external access to each TAP controller. This avoids the requirement to merge test vectors for more than one block in the device. One solution to this is to adopt a hybrid, using 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** and **TDI** signals, and to multiplex out independent JTAG ports on pins that are used for another purpose during normal operation.

2.8.1 See also

Concepts

- *ICs containing multiple devices* on page 2-11.

2.9 PCB guidelines

PCB guidelines on the physical and electrical connections present on the target board are described in the following topics:

- *PCB connections* on page 2-14
- *Target interface logic levels* on page 2-15.

You must consider the following:

- JTAG interface connector pinouts
- *Serial Wire Debug* (SWD) connections.

2.9.1 See also

Concepts

- *ASIC guidelines* on page 2-10
- *JTAG signal integrity and maximum cable lengths* on page 2-17.

Reference

- Chapter 3 *JTAG Interface Connections*
- Chapter 5 *Serial Wire Debug*.

2.10 PCB connections

It is recommended that you place the 20-way JTAG header as closely as possible to the target device, because this minimizes any possible signal degradation caused by long PCB tracks. The header must be a 0.1 inch pitch standard box header.

The following figure shows the layout of possible PCB connections:

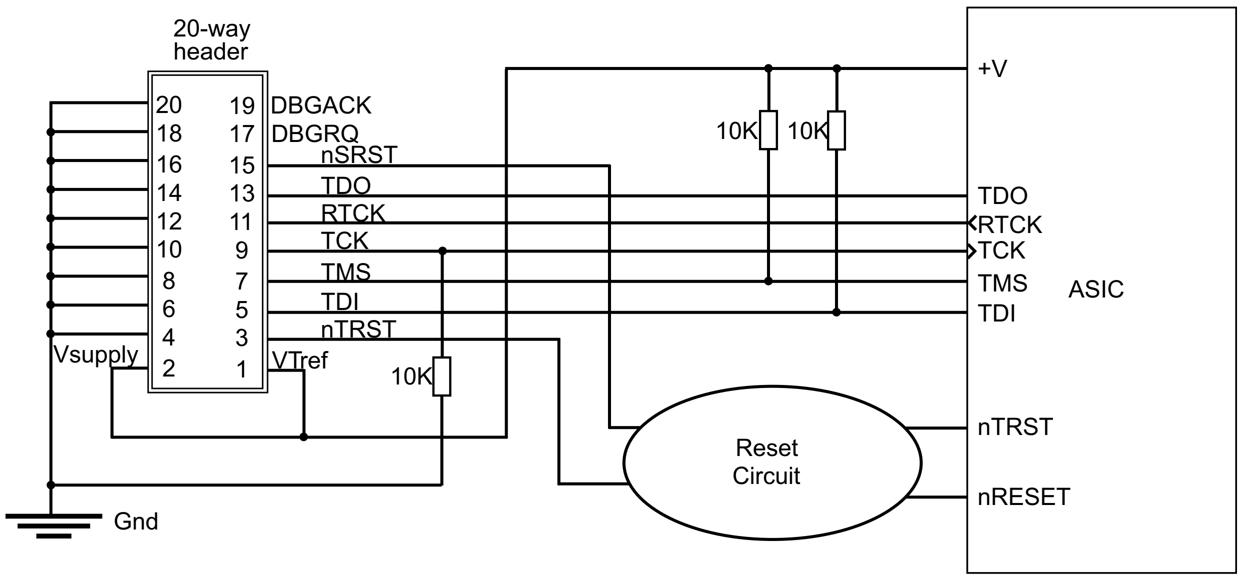

Figure 2-8 Typical PCB connections

Note

- The signals **TMS** and **TDI** must be pulled up on the PCB, as shown.
- **TCK** must be pulled down on the PCB, as shown.
- **DBGRQ** and **DBGACK** are not used by the ARM tools, but are used by some third party tools. If your tools do not use **DBGRQ** and **DBGACK**, and your device has a **DBGRQ** input, this must be pulled down on the PCB.
- **RTCK** is used by -S processor variants, such as the ARM966E-S processor. If your device does not use **RTCK**, then **RTCK** must be pulled down on the PCB.

2.10.1 See also

Concepts

- *Target interface logic levels* on page 2-15.

2.11 Target interface logic levels

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

VTref (pin 1 on the JTAG header connector) feeds the reference voltage to the RVI run control unit. This voltage, clipped at approximately 3.2V, is used as the output high voltage (**Voh**) 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 (**Vi(th)**) for the **TDO**, **RTCK**, and **nSRST** inputs is 50% of the **Voh** level, and so is clipped to approximately 1.6V. The relationships of **Voh** and **Vi(th)** to **VTref** are shown in the following figure:

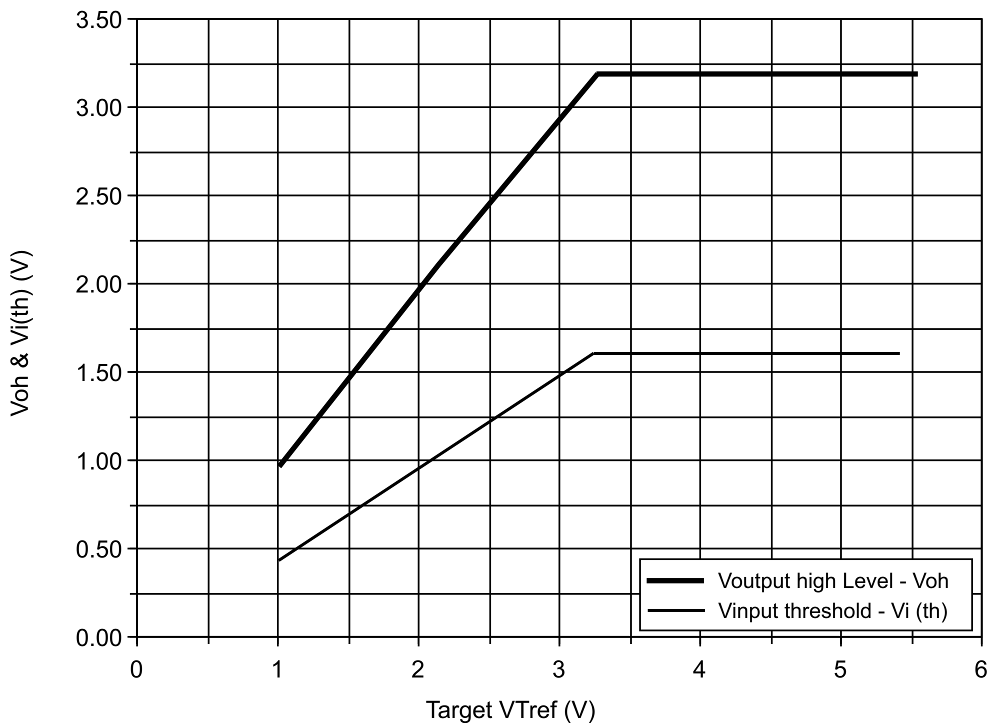

Figure 2-9 Target interface logic levels

RVI can adapt interface levels down to **VTref** less than 1V. If, however, **VTref** becomes less than approximately 0.55V, RVI interprets this condition as Target Not Present, and the software reports this as an error condition.

The **nTRST** output from RVI is effectively driven as an active low signal, so it is actively pulled to 0V but relies on a 4.7kΩ pull-up resistor 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 RVI is a similarly-driven active low signal, and must be pulled-up with a resistor in the target system. Because this signal is also an input to the RVI run control unit, there is a 4.7kΩ internal pull-up resistor. This is to avoid spurious lows on the input when **nSRST** is not connected to the target system.

The input and output characteristics of the RVI run control 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 on the *Low Voltage Differential Signaling* (LVDS) probe of the **TCK**, **TMS**, and **TDI** signals is approximately 47Ω

- the output impedance of all other signals is approximately 100Ω.

2.11.1 See also

Concepts

- *RVI reset signals* on page 2-7
- *PCB connections* on page 2-14.

2.12 JTAG signal integrity and maximum cable lengths

For JTAG-based debugging, you must have a very reliable connection between RVI and the target, because there is no way to detect or correct errors. For this reason it is important to guarantee good signal integrity.

One factor that can limit the maximum cable length is propagation delays. Normally the RVI run control unit samples data returning from the target using the same clock as for sending data, **TCK**. If the propagation delay gets too long then the RVI run control unit samples the signal at the wrong time. This can be resolved by using *adaptive clocking*. In this mode the target returns a clock, **RTCK**, and RVI does not sample data on **TDO**, or send more data on **TDI**, until clocked by this signal.

In an ASIC or ASSP (for example, in ARM processor-based microcontrollers) the **TDO** and **RTCK** signals are not typically implemented with a stronger driver than other signals on the device. The strength of these drivers varies from device to device. An example specification is to sink or source 4mA. Many designs connect these pins on the device directly to the corresponding pins on the RVI connector.

Over very short lengths of cable, such as the one supplied with RVI, this type of weak driver is adequate. However, if longer cables are used then the cable becomes harder to drive as the capacitive load increases. When using longer cables it becomes essential to consider the cable as a transmission line and to provide appropriate impedance matching, otherwise reflections occur.

RVI has much stronger drivers and they are connected through 100Ω series resistors to impedance match with the JTAG cable. This is very much better than the typical circuit used at the target end.

With the typical situation at the target end (weak drivers, no impedance matching resistors) you can only expect reliable operation over short cables (approximately 30cm). If operation over longer cables is required:

- For very long cables, a solution is to buffer the JTAG signals through differential drivers, such as the *Low Voltage Differential Signaling* (LVDS) cable and probe supplied with RVI. Reliable operation is possible over tens of meters using this technique.
- For intermediate lengths of cables, you can instead improve the circuitry used at the target end. The recommended solution is to add an external buffer with good current drive and a 100Ω series resistor for the **TDO** (and **RTCK** if used) signals on your target hardware. Using this technique you can debug over cable lengths up to several meters. Depending on cable length and propagation delays through your buffers and cables, it might still be necessary to use adaptive clocking.

  If you are not already using adaptive clocking in your design, you can generate **RTCK** at the target end by using the **TCK** signal fed through the same buffer and impedance matching circuit as used for **TDO**.

Reducing the clock speed used by RVI avoids some, but not all, of the problems associated with long cables. If reducing the speed of downloading code and reading memory in the debugger is not a significant problem, try experimenting with lowering this clock speed.

2.12.1 See also

Concepts

- *ASIC guidelines* on page 2-10
- *PCB guidelines* on page 2-13.

Reference

- Chapter 3 *JTAG Interface Connections*
- Chapter 5 *Serial Wire Debug*.

Chapter 3
JTAG Interface Connections

The following topics provide descriptions of the interface connections on the ARM® RVI™ unit:

- *JTAG interface pinouts* on page 3-2
- *JTAG interface signals* on page 3-3
- *JTAG port timing characteristics* on page 3-6.

3.1 JTAG interface pinouts

The RVI run control unit is supplied with a short ribbon cable, and a longer ribbon cable and *Low Voltage Differential Signaling* (LVDS) probe. These both terminate in a 20-way 2.54mm pitch *Insulation Displacement Connector* (IDC) connector. You can use either cable to mate with a keyed box header on the target. The pinout is shown in the following figure:

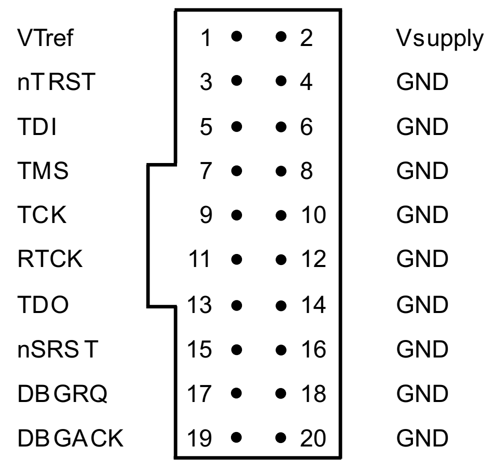

Figure 3-1 JTAG interface pinout

Note

All **GND** pins must be connected to **0V** on the target hardware.

3.1.1 See also

Concepts

- *JTAG interface signals* on page 3-3
- *JTAG port timing characteristics* on page 3-6.

Reference

- Chapter 4 *User I/O Connections*.

3.2 JTAG interface signals

The following table describes the signals on the JTAG interfaces:

Table 3-1 JTAG signals

Signal	I/O	Description
DBGACK	-	This pin is connected in the RVI run control unit, but is not supported in the current release of the software. It is reserved for compatibility with other equipment to be used as a debug acknowledge signal from the target system. It is recommended that this signal is pulled LOW on the target.
DBGRQ	-	This pin is connected in the RVI run control unit, but is not supported in the current release of the software. It is reserved for compatibility with other equipment to be used as a debug request signal to the target system. The RVI software maintains this signal as LOW. When applicable,RVI uses the scan chain 2 of the processor to put the processor in debug state. It is recommended that this signal is pulled LOW on the target.
GND	-	Ground.
nSRST	Input/output	Active Low output from RVI to the target system reset, with a 4.7kΩ pull-up resistor for de-asserted state. This is also an input to RVI so that a reset initiated on the target can be reported to the debugger. This pin must be pulled HIGH on the target to avoid unintentional resets when there is no connection.
nTRST	Output	Active Low output from RVI to the Reset signal on the target JTAG port, driven to the VTref voltage for de-asserted state. This pin must be pulled HIGH on the target to avoid unintentional resets when there is no connection.
RTCK	Input	Return Test Clock signal from the target JTAG port to RVI. Some targets must synchronize the JTAG inputs to internal clocks. To assist in meeting this requirement, you can use a returned, and retimed, **TCK** to dynamically control the **TCK** rate. RVI provides Adaptive Clock Timing, that waits for **TCK** changes to be echoed correctly before making more changes. Targets that do not have to process **TCK** can ground this pin. **RTCK** is not supported in *Serial Wire Debug* (SWD) mode.
TCK	Output	Test Clock signal from RVI to the target JTAG port. It is recommended that this pin is pulled LOW on the target.
TDI	Output	Test Data In signal from RVI to the target JTAG port. It is recommended that this pin is pulled HIGH on the target.
TDO	Input	Test Data Out from the target JTAG port to RVI. It is recommended that this pin is pulled HIGH on the target.

Table 3-1 JTAG signals (continued)

Signal	I/O	Description
TMS	Output	Test Mode signal from RVI to the target JTAG port. This pin must be pulled HIGH on the target so that the effect of any spurious **TCK**s when there is no connection is benign.
Vsupply	Input	This pin is not connected in the RVI unit. It is reserved for compatibility with other equipment to be used as a power feed from the target system.
VTref	Input	This is the target reference voltage. It indicates that the target has power, and It must be at least 0.628V. **VTref** is normally fed from $\mathbf{V_{dd}}$ on the target hardware and might have a series resistor (though this is not recommended). There is a 10kΩ pull-down resistor on **VTref** in RVI.

3.2.1 JTAG interface signal details

VTref is used to create the logic-level reference for the input comparators on **TDO**, **RTCK** and **nSRST**. RVI clips the logic-level reference to 3.3V. RVI inputs (**TDO**, **RTCK** and **nSRST**) are taken to high-impedance inputs of comparators. Each input is read as a logic 1 when it exceeds half the voltage reference.

VTref also controls the output logic levels to the target. RVI uses analog switches to drive the output signals. The output is connected to ground for a logic 0 and to the JTAG interface voltage for a logic 1.

TDI, **TMS** and **TCK** have 47Ω series resistors on the *Low Voltage Differential Signaling* (LVDS) probe. All other outputs from the LVDS probe and the RVI 20-way connector have 100Ω series resistors.

nSRST and **nTRST** are both active low signals. When asserted, both these signals are connected to ground for a logic 0. When de-asserted, nSRST uses a 4.7kΩ pull-up for a logic 1, whereas nTRST is driven to the VTref voltage for de-asserted state.

You must ensure that your board has appropriate pull-up and pull-down resistors on the JTAG signals:

- **TMS**, **TDI**, **TDO**, **nSRST** and **nTRST** must have pull-ups.
- **TCK** must have a pull-down to enable hot swap and post-mortem debugging
- **RTCK** must have a pull-down to fix a stable value on that signal when debugging a non-synthesizable processor.
- **DBGRQ** must have a pull-down. This ensures that the processor does not enter debug state in an uncontrolled way.
- **DBGACK** must have a pull-down, so the default value that the debugger sees is processor not in debug state.

The recommended value for pull-ups and pull-downs is 10kΩ, although the optimum value depends on the signal load. For example, pull-downs must be about 1kΩ when working with TTL logic.

3.2.2 See also

Concepts

- *JTAG interface pinouts* on page 3-2

- *JTAG port timing characteristics* on page 3-6.

Reference

- Chapter 4 *User I/O Connections*.

3.3 JTAG port timing characteristics

You must consider the timing characteristics of a RVI unit if you design a target device or board and want to be able to connect RVI at a particular **TCK** frequency. The characteristics relate to the RVI hardware. You must consider them in parallel with the characteristics of your target.

The following figure shows the JTAG port timing and parameters:

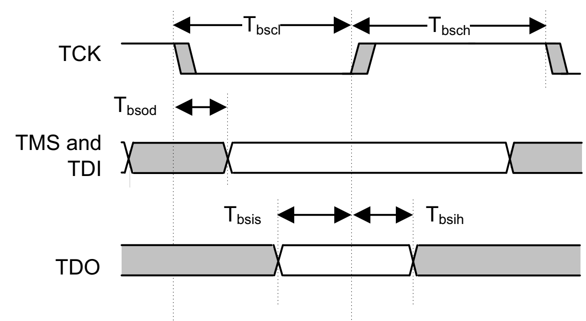

Figure 3-2 JTAG port timing diagram

In a JTAG device that fully complies to IEEE1149.1-2001, **TDI** and **TMS** are sampled on the rising edge of **TCK**, and **TDO** changes on the falling edge of **TCK**. To take advantage of these properties, RVI samples **TDO** on the rising edge of **TCK** and changes its **TDI** and **TMS** signals on the falling edge of **TCK**. This means that with a fully compliant target, issues with minimum setup and hold times can always be resolved by decreasing the **TCK** frequency, because this increases the separation between signals changing and being sampled.

——— **Note** ———

There are no separate timing requirements for adaptive clocking mode, because the minimum T_{bsch} and T_{bscl} times are identical and are the same as for non-adaptive clocking. T_{bsis} and T_{bsih} are relative to RTCK rising, and not TCK rising, as RTCK is used to sample TDO in adaptive clocking mode.

———————

The only real timing difference is that in adaptive mode, RVI samples **TDO** on the rising edge of **RTCK** and not **TCK**, so **TDO** timing is relative to **RTCK**.

The following table shows the timing requirements for the JTAG A port, measured open circuit (no target connection, except for 3.3V reference on **VTref**) with the supplied JTAG cable connected:

Table 3-2 RVI JTAG A timing requirements

Parameter	Min	Max	Description
T_{bscl}	50ns	500µs	**TCK** LOW period
T_{bsch}	50ns	500µs	**TCK** HIGH period
T_{bsod}	-	6.0ns	**TDI** and **TMS** valid from **TCK** (falling)
T_{bsis}	15.0ns	-	**TDO** setup to **TCK** (rising)
T_{bsih}	6.0ns	-	**TDO** hold from **TCK** (rising)

The following table shows the timing requirements for the JTAG B port, measured open circuit (no target connection, except for 3.3V reference on **VTref**) with no cable connected:

Table 3-3 RVI JTAG B timing requirements

Parameter	Min	Max	Description
T_{bscl}	10ns	500µs	**TCK** LOW period
T_{bsch}	10ns	500µs	**TCK** HIGH period
T_{bsod}	-	3.2ns	**TDI** and **TMS** valid from **TCK** (falling)
T_{bsis}	6.2ns	-	**TDO** setup to **TCK** (rising)
T_{bsih}	4.5ns	-	**TDO** hold from **TCK** (rising)

Note

- The RVI software enables you to change the **TCK** frequency. The **TCK** LOW:HIGH mark-space ratio is always 50:50. The other parameters must be considered with the specific values of T_{bscl} and T_{bsch} that you have chosen. The default values for an autoconfigured single-TAP system are, nominally, T_{bscl}=50ns and T_{bsch}=50ns.
- T_{bsod} is the maximum delay between the falling edge of **TCK** and valid levels on the **TDI** and **TMS** RVI output signals. The target samples these signals on the following rising edge of **TCK** and so the minimum setup time for the target, relative to the rising edge of **TCK**, is $T_{bscl}-T_{bsod}$.
- T_{bsis} is the minimum setup time for the **TDO** input signal, relative to the rising edge of **TCK** when RVI samples this signal. The target changes its **TDO** value on the previous falling edge of **TCK** and so the maximum time for the target **TDO** level to become valid, relative to the falling edge of **TCK**, is $T_{bscl}-T_{bsis}$.

3.3.1 See also

Concepts

- *JTAG interface pinouts* on page 3-2
- *JTAG interface signals* on page 3-3.

Reference

- Chapter 4 *User I/O Connections*.

Chapter 4
User I/O Connections

The following topic describes the additional input and output connections provided in ARM® RVI™, and consists of:

- *The RVI User I/O connector* on page 4-2.

4.1 The RVI User I/O connector

The User *Input/Output* (I/O) connector is situated on an end panel of the RVI unit. The connector is a 10-way 2.54mm pitch *Insulation Displacement Connector* (IDC) header that mates with IDC sockets mounted on a ribbon cable, as shown in the following figure.

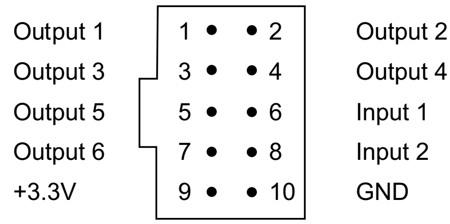

Figure 4-1 User I/O pin connections

——— **Warning** ————

You must establish a common ground between the RVI unit and the target hardware before you connect any of the User I/O signals.

The following table shows the User I/O pin connections.

Table 4-1 User I/O pin connections

Pin	Signal	I/O	Description
Pin 1	**Output 1**	Output	This is a user output bit. It operates at a 3.3V swing, with a 100Ω series resistance.
Pin 2	**Output 2**	Output	This is a user output bit. It operates at a 3.3V swing, with a 100Ω series resistance.
Pin 3	**Output 3**	Output	This is a user output bit. It operates at a 3.3V swing, with a 100Ω series resistance.
Pin 4	**Output 4**	Output	This is a user output bit. It operates at a 3.3V swing, with a 100Ω series resistance.
Pin 5	**Output 5**	Output	This is a user output bit. It operates at a 3.3V swing, with a 100Ω series resistance.
Pin 6	**Input 1**	Input	This is a user input bit. It has a 10kΩ weak pull-up to the unit internal +3.3V supply, and requires a $\mathbf{V_{ih(min)}}$ of 2.0V and a $\mathbf{V_{il(max)}}$ of 0.8V. It can safely be driven by 5V logic levels, and has *Electro Static Discharge* (ESD) protection greater than the 2kV human body model. This pin is not currently supported.
Pin 7	**Output 6**	Output	This is a copy of the trigger output on the end panel of the RVI unit. It operates at a 3.3V swing, with a 100Ω series resistance.

Table 4-1 User I/O pin connections (continued)

Pin	Signal	I/O	Description
Pin 8	**Input 2**	Input	This is a copy of the trigger input on the end panel of the RVI unit. It has a 10kΩ weak pull-up to the unit internal +3.3V supply, and requires a $\mathbf{V_{ih(min)}}$ of 2.0V and a $\mathbf{V_{il(max)}}$ of 0.8V. It can safely be driven by 5V logic levels, and has ESD protection greater than the 2kV human body model. This pin is not currently supported.
Pin 9	**+3.3V**	Output	This is intended for powering external signal conditioning circuitry, to a maximum current of 100mA. Incorrect use of this output might cause the RVI unit to enter current limit.
Pin 10	**GND**	-	-

Note

Input is not currently supported on the User I/O pin connections.

4.1.1 See also

Reference

ARM RVI™ Setting up the Hardware:

- *The RVI debug unit* on page 2-9.

Chapter 5
Serial Wire Debug

The following topics describe *Serial Wire Debug* (SWD) connection to the *Debug Access Port* (DAP), and the functionality available for use with the *Low Voltage Differential Signaling* (LVDS) probe:

- *Target interface* on page 5-2
- *SWD timing requirements* on page 5-3.

5.1 Target interface

The functionality available for use with the *Low Voltage Differential Signaling* (LVDS) probe supports a *Serial Wire Debug* (SWD) connection to the *Debug Access Port* (DAP). SWD is an alternative protocol to JTAG for connecting to CoreSight processors, and has the advantage of requiring fewer pins than previous probes. It also supports higher data rates.

The following table shows the SWD pinout for the connector alongside the JTAG pinout.

Table 5-1 SWD interface pinout

Pin	Signal		Pin	Signal
	JTAG	Serial Wire		
1	VTref	VTref	2	NC
3	nTRST	NC	4	GND
5	TDI	NC	6	GND
7	TMS	SWDIO	8	GND
9	TCK	SWDCLK	10	GND
11	RTCK	NC	12	GND
13	TDO	SWO	14	GND
15	nSRST	nSRST	16	GND
17	DBGRQ	DBGRQ	18	GND
19	DBGACK	DBGACK	20	GND

5.1.1 See also

Concepts

- *SWD timing requirements* on page 5-3.

5.2 SWD timing requirements

The functionality supports a *Serial Wire Debug* (SWD) connection to the *Debug Access Port* (DAP). SWD is an alternative protocol to JTAG for connecting to CoreSight processors, and has the advantage of requiring fewer pins than previous probes. It also supports higher data rates.

ARM® RVI™ connects to the serial wire-enabled target using the *Low Voltage Differential Signaling* (LVDS) probe. The interface uses only two lines, but for clarity the diagrams shown in the following figure separate the SWDIO line to show when it is driven by either the RVI probe or target.

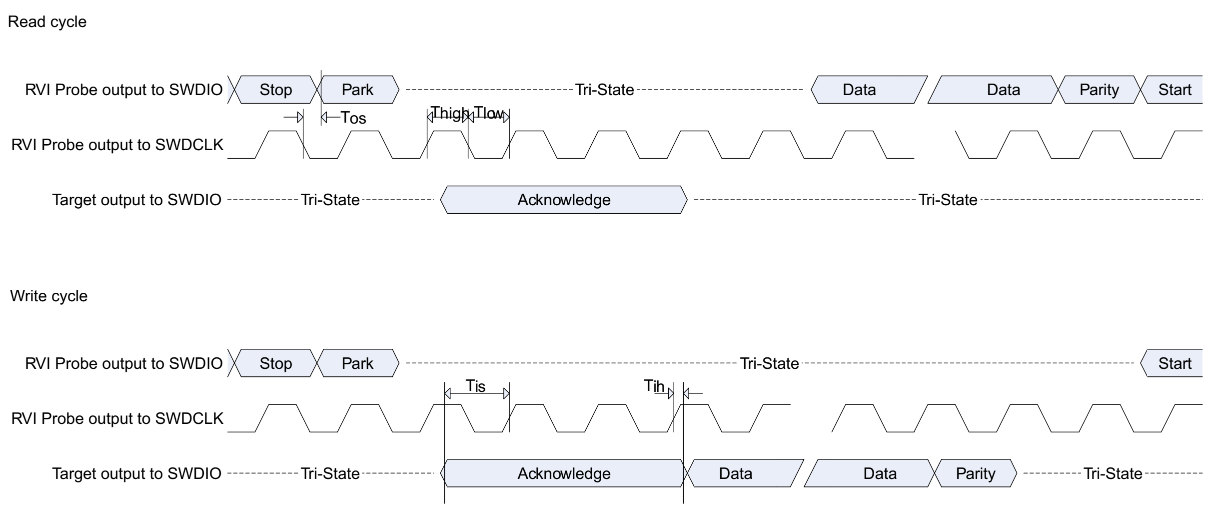

Figure 5-1 SWD timing diagrams

The probe outputs data to SWDIO on the falling edge of SWDCLK. The probe captures data from SWDIO on the rising edge of SWDCLK. The target outputs data to SWDIO on the rising edge of SWDCLK. The target captures data from SWDIO on the rising edge of SWDCLK.

The following table shows the timing requirements for the SWD.

Table 5-2 SWD timing requirements

Parameter	Min	Max	Description
T_{high}	10ns	500µs	**SWDCLK** HIGH period
T_{low}	10ns	500µs	**SWDCLK** LOW period
T_{os}	-5ns	5ns	**SWDIO** Output skew to falling edge **SWDCLK**
T_{is}	4ns	-	Input Setup time required between **SWDIO** and rising edge **SWDCLK**
T_{ih}	1ns	-	Input Hold time required between **SWDIO** and rising edge **SWDCLK**

5.2.1 See also

Concepts

- *Target interface* on page 5-2.