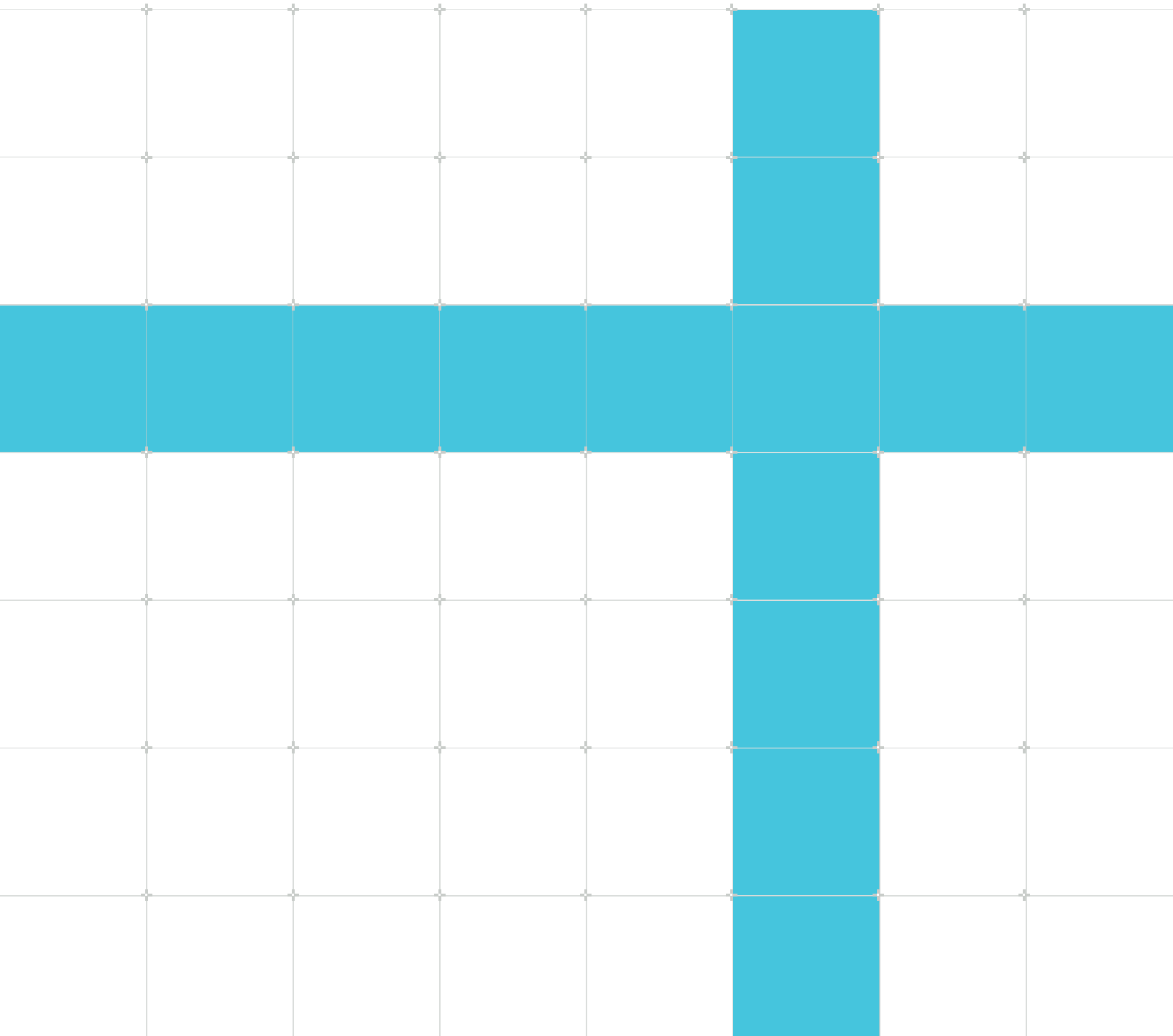
DSTREAM-XT

Version 1.0

System and Interface Design Reference Guide

Issue 01
102444_1.0_01_en



DSTREAM-XT

System and Interface Design Reference Guide

Release information

Document history

Issue	Date	Confidentiality	Change
0100-00	15 July 2021	Non-Confidential	New document for v1.0
0100-01	10 October 2022	Non-Confidential	Documentation update for version 1.0 release

Proprietary Notice

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(LES-PRE-20349|version 21.0)

Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by Arm and the party that Arm delivered this document to.

Unrestricted Access is an Arm internal classification.

Product Status

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

Feedback

Arm® welcomes feedback on this product and its documentation. To provide feedback on the product, create a ticket on https://support.developer.arm.com

To provide feedback on the document, fill the following survey: https://developer.arm.com/documentation-feedback-survey.

Inclusive language commitment

Arm values inclusive communities. Arm recognizes that we and our industry have used language that can be offensive. Arm strives to lead the industry and create change.

Previous issues of this document included language that can be offensive. We have replaced this language.

To report offensive language in this document, email terms@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).

Class A

Important: This is a Class A device. In residential areas, this device may cause radio interference. The user should take the necessary precautions, if appropriate.

CE/UKCA Conformity

These marks indicate that this product meets all essential health, safety and environmental requirements. The CE mark indicates conformity within EU member states and the UKCA mark indicates conformity within the UK.

The Declarations of Conformity are available on request.

The *Waste Electrical and Electronic Equipment* (WEEE) marking, that is, the crossed out wheelie-bin figure, indicates that this product must not be disposed of with general waste within the European Union. To prevent possible harm to the environment from uncontrolled waste disposal, the user is required to recycle the product responsibly to promote reuse of material resources. To comply with EU law, you must dispose of the product in one of the following ways:

- Return it to the distributor where it was purchased. The distributor is required to arrange free collection when requested.
- Recycle it using local WEEE recycling facilities. These facilities are now very common and might provide free collection.
- If purchased directly from Arm, Arm provides free collection. Please e-mail weee@arm.com for instructions.

- End-of-Life products can be disposed of safely using an *Approved Authorized Treatment Facility* (AATF). To support safe disposal, Arm has partnered with B2B Compliance. B2B can be contacted at the following weblink: https://b2bcompliance.org.uk

During the lifetime of the product, you are advised to:

- Inspect the product regularly to ensure that it is in good working order.
- Ensure that the product is free from dust and debris that might cause damage.
- Clean the product with an air duster when necessary.
- Power down the system when not in use.
- Observe ESD precautions when handling the product.

The product can radiate Radio Frequency Interference (RFI) or Electromagnetic Interference (EMI) and might cause harmful interference to radio communications. There is no guarantee that interference cannot occur in a particular installation. If you suspect that this equipment is causing interference to other equipment, you are encouraged to try to correct the interference by one or more of the following measures:

- Ensure attached cables do not lie across any sensitive equipment.
- Increase the distance between the product and the receiver.
- Connect the equipment to an outlet on a circuit different from that to which the product is connected.
- Consult Arm for help.

The product can be sensitive to Radio Frequency Interference (RFI) or Electromagnetic Interference (EMI) which might cause incorrect operation of the product:

- Avoid using the product near sources of EMI.
- Never use the product in *Safety-Critical-Systems* (SCS), or *Life-Critical-Systems* (LCS).

Arm recommends that, wherever possible, shielded interface cables be used.

Contents

List of Figures

List of Tables

1. Introduction

DSTREAM-XT System and Interface Design Reference Guide describes the interfaces of the DSTREAM-XT debug and trace units, with details about designing Arm architecture-based devices and PCBs. This document is written for those using DSTREAM-XT or those designing target boards to be used with DSTREAM-XT.

1.1 Conventions

The following subsections describe conventions used in Arm documents.

Glossary

The Arm Glossary is a list of terms used in Arm documentation, together with definitions for those terms. The Arm Glossary does not contain terms that are industry standard unless the Arm meaning differs from the generally accepted meaning.

See the Arm® Glossary for more information: developer.arm.com/glossary.

Typographic conventions

Arm documentation uses typographical conventions to convey specific meaning.

Convention	Use
italic	Citations.
bold	Interface elements, such as menu names. Terms in descriptive lists, where appropriate.
`monospace`	Text that you can enter at the keyboard, such as commands, file and program names, and source code.
`monospace underline`	A permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.
`<and>`	Encloses replaceable terms for assembler syntax where they appear in code or code fragments. For example: `MRC p15, 0, <Rd>, <CRn>, <CRm>, <Opcode_2>`
SMALL CAPITALS	Terms that have specific technical meanings as defined in the *Arm® Glossary*. For example, **IMPLEMENTATION DEFINED**, **IMPLEMENTATION SPECIFIC**, **UNKNOWN**, and **UNPREDICTABLE**.
Caution	Recommendations. Not following these recommendations might lead to system failure or damage.
Warning	Requirements for the system. Not following these requirements might result in system failure or damage.
Danger	Requirements for the system. Not following these requirements will result in system failure or damage.

Convention	Use
 Note	An important piece of information that needs your attention.
 Tip	A useful tip that might make it easier, better or faster to perform a task.
 Remember	A reminder of something important that relates to the information you are reading.

1.2 Useful resources

This document contains information that is specific to this product. See the following resources for other useful information.

Access to Arm documents depends on their confidentiality:

- Non-Confidential documents are available at developer.arm.com/documentation. Each document link in the following tables goes to the online version of the document.
- Confidential documents are available to licensees only through the product package.

Arm product resources	Document ID	Confidentiality
Arm Development Studio Getting Started Guide	101469	Non-Confidential
Arm Development Studio Heterogeneous system debug with Arm Development Studio	102021	Non-Confidential
Arm Development Studio User Guide	101470	Non-Confidential
Arm DSTREAM-ST Getting Started Guide	100892	Non-Confidential
Arm DSTREAM-ST System and Interface Design Reference Guide	100893	Non-Confidential
Arm DSTREAM-XT Getting Started Guide	102443	Non-Confidential

Non-Arm resources	Documentation	Organization
JTAG Specification	IEEE	Institue of Electrical and Electronics Engineers
PCIE Specification	PCI-SIG	PCI-SIG

1.3 Other information

See the Arm website for other relevant information.

- Arm® Developer.
- Arm® Documentation.

- Technical Support.
- Arm® Glossary.

2. Debug and trace interface

The Arm debug and trace interface enables powerful software debug and optimization on an Arm® processor-based target system. It is based on the IEEE 1149.1 (JTAG) interface coupled with various additional signals. This chapter introduces these signals and describes their use within the interface.

Unless otherwise specified:

- All pull-up/pull-down resistors that are discussed in this chapter must be between 1K and 100K (10K is recommended).
- Any signals that begin with a lowercase 'n' are, by default, active *low*.

2.1 JTAG signals

Most Arm®-based devices are physically equipped with several pins that are dedicated to debug and test purposes. Four of these pins make up the *IEEE 1149.1* interface, also known as the *JTAG* interface. This interface is often used for boundary-scan testing during the manufacture of printed circuit boards. The interface also provides a useful way to access one or more cores, and other components in a device, while running its application software.

TDI

The **TDI** (Test Data In) signal is an input to the target device which provides a stream of serial data from the debug unit.

The **TDI** signal must be pulled *high* on the target to keep the signal inactive when no debug unit is connected.

TMS

The **TMS** (Test Mode Select) signal is an input to the target device which controls its JTAG state-machine.

The **TMS** signal must be pulled *high* on the target to keep the signal inactive when no debug unit is connected.

TCK

The **TCK** (Test Clock) signal is an input to the target device which synchronizes its JTAG state-machine. On each rising edge of the **TCK** signal, the target samples the **TDI**, and **TMS** signals.

Consider **TCK** as a strobe signal, rather than a clock signal, because it is typically non-continuous and only becomes active during debug communications.

TCK can be pulled *high* on the target, however, to maintain full compatibility with other JTAG equipment, Arm recommends you pull **TCK** *low*.

TDO

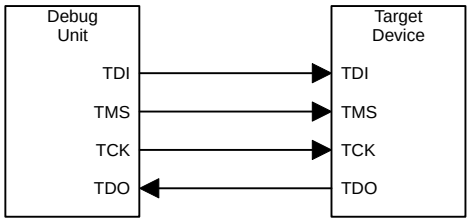
The **TDO** (Test Data Out) signal is an output from the target device which returns a stream of serial data to the debug unit.

TDO can be left floating on the target, however, to maintain full compatibility with other JTAG equipment, Arm recommends you pull **TDO** *high*.

JTAG connection

In the simplest form (omitting pull-up and pull-down resistors), a connection between the debug unit and the target device looks like this:

Figure 2-1: Simple JTAG connection



Note

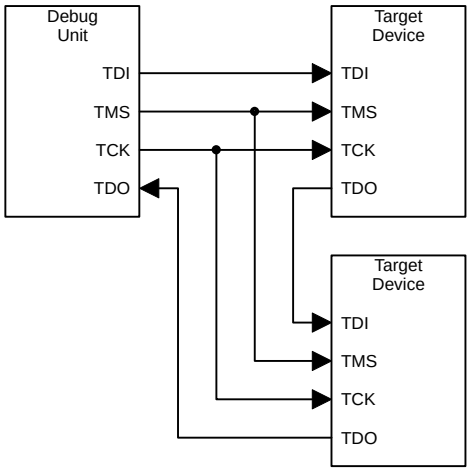
The naming convention of the **TDI** (Test Data In) and **TDO** (Test Data Out) signals is always with respect to the target device.

Caution

When multiple devices are used in a *scan-chain*, the **TCK** and **TMS** signals must be branched to each device. Good digital design practice must be used to ensure that these branches do not reduce the signal integrity of the signals, causing false edges to be received by the devices.

The flexible design of the JTAG interface enables you to connect multiple devices to a single debug unit:

Figure 2-2: Chained JTAG connection



A group of JTAG devices that are linked or *daisy-chained* together is often known as the JTAG chain or *scan-chain*.

For more information, see JTAG port buffering.

JTAG timing characteristics

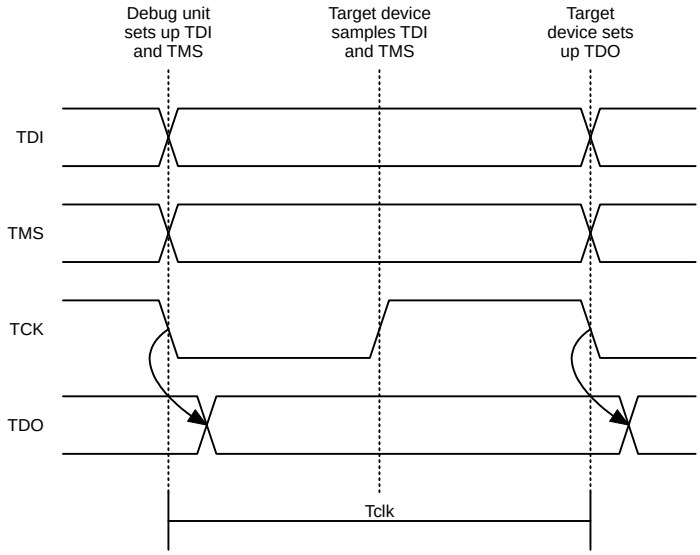
The JTAG timing characteristics of DSTREAM-XT systems conform to the requirements of the *IEEE 1149.1* (JTAG) specification.

TDI and **TMS** are set up by the DSTREAM-XT system on the falling edge of **TCK**. These signals are then sampled by the target device on the rising edge of **TCK**. The target device must set up its **TDO** signal when it detects the falling edge of **TCK** which, in turn, will be sampled by the DSTREAM-XT system on the next rising edge of **TCK**.

These timings are considered correct at the debug connector of the target board.

Basic JTAG timing:

Figure 2-3: JTAG timing diagram



Since all signals are set up on the falling edge of **TCK** and sampled on the rising edge, the effective setup and hold times for the target device and DSTREAM-XT system are approximately *Tclk*/2.

Issues with signal timing can usually be resolved by decreasing the **TCK** frequency. Decreasing the **TCK** frequency increases the setup and hold times.

TDO is always slightly delayed, relative to the other signals, because it takes a finite amount of time for the target device to detect the falling edge of **TCK** and then set up **TDO**. This slight delay, and the round-trip delay of the debug cable, are compensated for by the DSTREAM-XT system.

Note

There are no separate timing requirements for the adaptive clocking mode. In adaptive clocking mode, the debug unit samples **TDO** on the rising edge of **RTCK** instead of **TCK**, so **TDO** timing is relative to **RTCK**.

Table 2-1: JTAG timing Characteristics

Parameter	Min	Max	Description
$F_{[clk]}$	10Hz	180MHz	**TCK** frequency
$T_{[clk]}$	5.556ns	100ms	**TCK** period
$T_{[ds]}$	49%	51%	**TCK** Duty Cycle

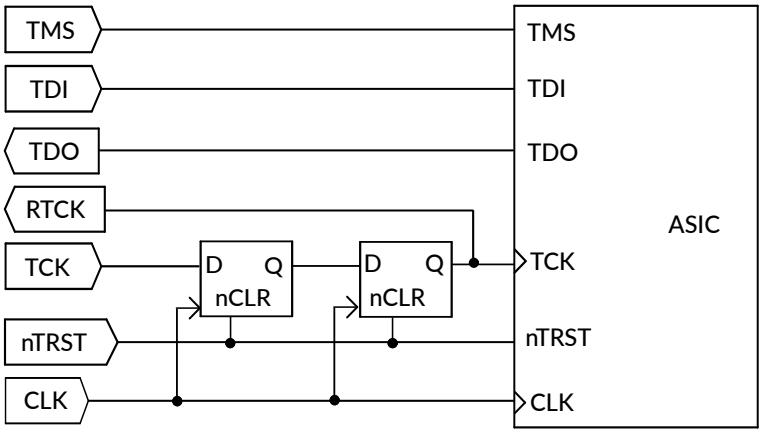
For further details on the JTAG interface, a full specification is available from: www.ieee.org.

Synchronization

As debug data is transferred to and from the target device, it must pass between two clock domains (**TCK** and the internal system clock of the target device). To achieve synchronized data transfer without suffering any meta-stability issues, a synchronizer circuit must be used within the target device.

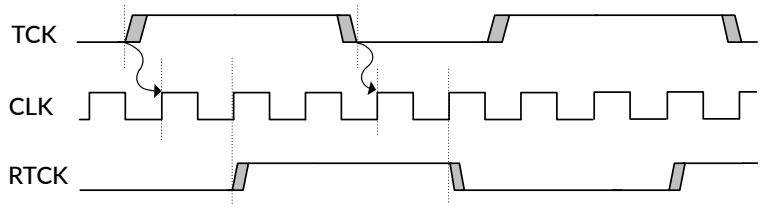
The following figure shows a circuit for a basic JTAG port synchronizer.

Figure 2-4: Basic JTAG port synchronizer



The following figure shows a partial timing diagram for the basic JTAG synchronizer. To reduce the delay, and because the second flip-flop only provides better immunity to metastability problems, clock the flip-flops from opposite edges of the system clock.

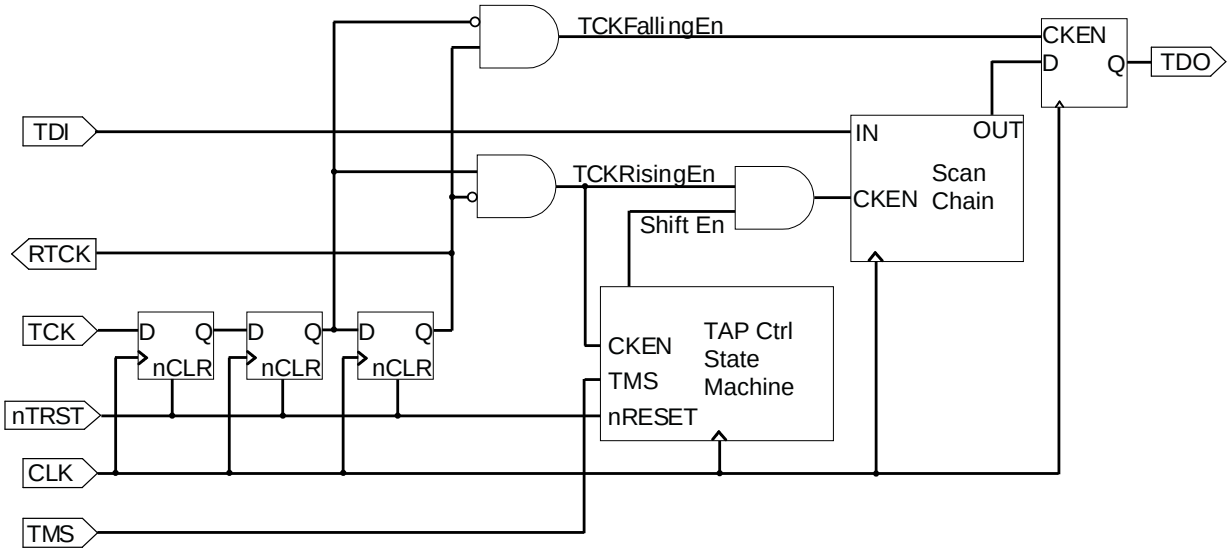
Figure 2-5: Timing diagram for the Basic JTAG synchronizer



ASIC design rules often impose a restriction that all flip-flops in a design must be clocked by one edge of a single clock. To interface the clocking restriction to a JTAG port that is asynchronous to the system, you must convert the JTAG **TCK** events into clock enables for this single clock. You must also ensure that the JTAG port cannot overrun this synchronization delay.

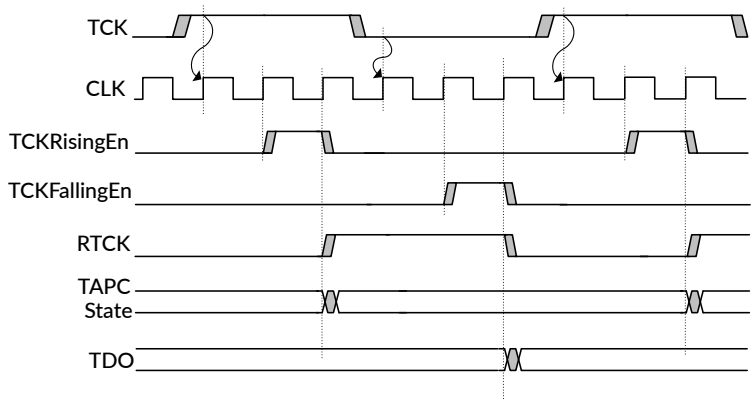
One possible implementation of this circuit, is:

Figure 2-6: 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**. The figure also shows how these enable signals gate the **RTCK** and **TDO** signals so that they only change state at the edges of **TCK**.

Figure 2-7: Timing diagram for the D-type JTAG synchronizer



2.2 Return Clock signal

Occasionally, a target device requires the JTAG interface to be externally synchronized to a clock within the device due to it being slow, non-continuous, or variable. The adaptive clocking feature uses the Return Clock signal (**RTCK**) to address this requirement.

You must never directly link **RTCK** to **TCK** on the target board. If the signal is directly linked, it is likely to cause false clock-edges to be received by the **TCK** input of the target device.

The **RTCK** signal is an output from the target system which is typically fed from the last flip-flop in the synchronization chain within the target device.

If used, the **RTCK** signal must be pulled *low* on the target. If not used, **RTCK** can be left floating or tied *low* on the target.

Adaptive clocking

- If you use the adaptive clocking feature, then the transmission delays, gate delays, and synchronization requirements might result in a lower clock frequency, compared to using fixed clocking. Adaptive clocking mode is not recommended unless the target design requires it.
- If you use adaptive clocking, the debug unit cannot detect the JTAG clock frequency, and therefore cannot scale its internal timeouts. If the clock frequency is too low, a JTAG timeout might occur, leaving the JTAG interface in an unknown state. To recover the connection, you must reset the debug unit. To disable JTAG timeouts, use the configuration settings in Arm® Development Studio. For more information, see Debug Hardware configuration in the Arm Development Studio User Guide.

When adaptive clocking is enabled, the debug unit issues a **TCK** signal and waits for the **RTCK** signal to return before sampling **TDO**. The debug unit does not progress to the next **TCK** transition until **RTCK** is received, allowing the target device to control the flow of the JTAG interface, as required.

You can enable Adaptive clocking using the configuration settings in Arm Development Studio. For more information, see Debug Hardware configuration in the Arm Development Studio User Guide.

2.3 Reset signals

Arm debug units have the ability to control two reset signals on the target: **nSRST** and **nTRST**.

nSRST

The **nSRST** (System Reset) signal, sometimes known as **nRESET** or **HRESET**, is an input to the target which performs a warm boot of the core (or cores) and other devices in the target system. The signal is often asserted by one or more of these conditions:

- Power On Reset (POR)
- Manual push-button reset
- Remote reset from a debug unit
- Watchdog reset

The **nSRST** signal must be pulled to its inactive logic level on the target.

By default, the **nSRST** signal is active *low*.

The polarity and drive strength of the **nSRST** signal is configurable in Arm® Development Studio.

nTRST

The **nTRST** (*TAP* Reset) signal initializes the *Test Access Port*, debug logic, and *boundary scan* cells in the target device.

The **nTRST** signal must be pulled to its inactive logic level on the target.

By default, the **nTRST** signal is active *low*.

The polarity and drive strength of the **nTRST** signal is configurable in Arm Development Studio.


Note

Arm strongly recommends that the **nSRST** and **nTRST** signals are made separately available on the JTAG connector. If the **nSRST** and **nTRST** signals are linked together, resetting the system also resets the *TAP* controller, which means:

- It might not be possible to debug a system from reset because any breakpoints have been lost.
- You might need to restart the debug session because the JTAG interface can't recover when the *TAP* controller is left in an unknown state.

It is expected that the assertion of the **nSRST** line by the DSTREAM-XT system causes a warm reset of the target system. If the **nSRST** line triggers a full, Power On Reset (*POR*), then the debug connection might be lost.

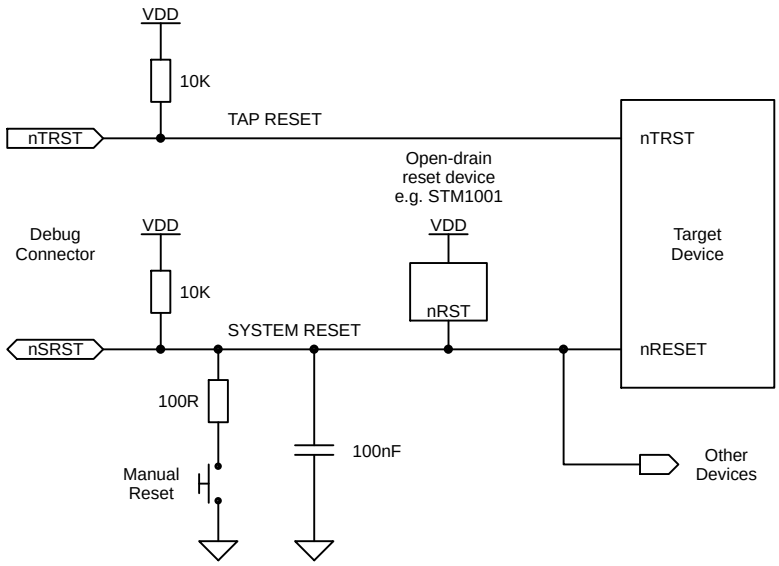
With regards to the reset signals output from the DSTREAM-XT system, the strong pull-up/pull-down resistance is approximately 33Ω, and the weak pull-up/pull-down resistance is approximately 4.7kΩ.

Because it is possible to switch the polarity and drive strength of **nTRST** and **nSRST**, target systems with various different reset configurations are supported.

Example reset circuit

A typical reset circuit which would be present on the target board, is:

Figure 2-8: Example reset circuit



The push-button, 100Ω resistor, and 100nF capacitor shown here are an example of how a manual reset button can be interfaced with the **nSRST** signal. This is optional and would typically be used on development boards.

The reset device that is shown here would keep the target device, and any other system devices, in their reset state until the power rail has reached a minimum valid voltage. If the target device has a separate Power On Reset (*POR*) input, any voltage monitoring devices would typically connect to that instead. If the target device is equipped with internal voltage monitoring circuitry, external monitoring devices can be omitted.

2.4 Run-Control signals

The run-control signals are now deprecated in Arm® Development Studio. However, low-speed control might be possible through *ConfigItems* and the command-line utilities.

DBGRQ

The **DBGRQ** (Debug Request) pin stops the target processor and puts it into its debug state.

Arm recommends that you do not use this signal. You can leave the signal open on the target board.


Note

If the signal is used by the target processor, it must be pulled *low* on the target board.

DBGACK

The **DBGACK** (Debug Acknowledge) pin notifies the debug unit that a debug request has been received, and that the target processor is now in its debug state.

Arm recommends that you do not use this signal. You can leave the signal open on the target board.


Note

If the signal is used by the target processor, it must be pulled *low* on the target board.

2.5 Serial Wire Debug signals

Serial Wire Debug (**SWD**) is commonly used on reduced pin-count target devices. SWD only requires two pins, instead of the four pins used by JTAG.

SWDIO


Caution

The **SWDIO** signal is bidirectional and its functionality is shared with the unidirectional JTAG **TMS** signal. Check that there are no buffers on the target that might prevent bidirectional communication.

The **SWDIO** (Serial Wire Data I/O) signal is a bidirectional signal which sends and receives serial data to and from the target.

When no debug unit is connected, the **SWDIO** signal must be pulled *high* on the target to keep the signal inactive.

SWCLK

The **SWCLK** (Serial Wire Clock) signal is an input to the target. It clocks data into, and out of, the target device.

When no debug unit is connected, the **SWCLK** signal must be pulled *low* on the target to keep the signal inactive.

SWO

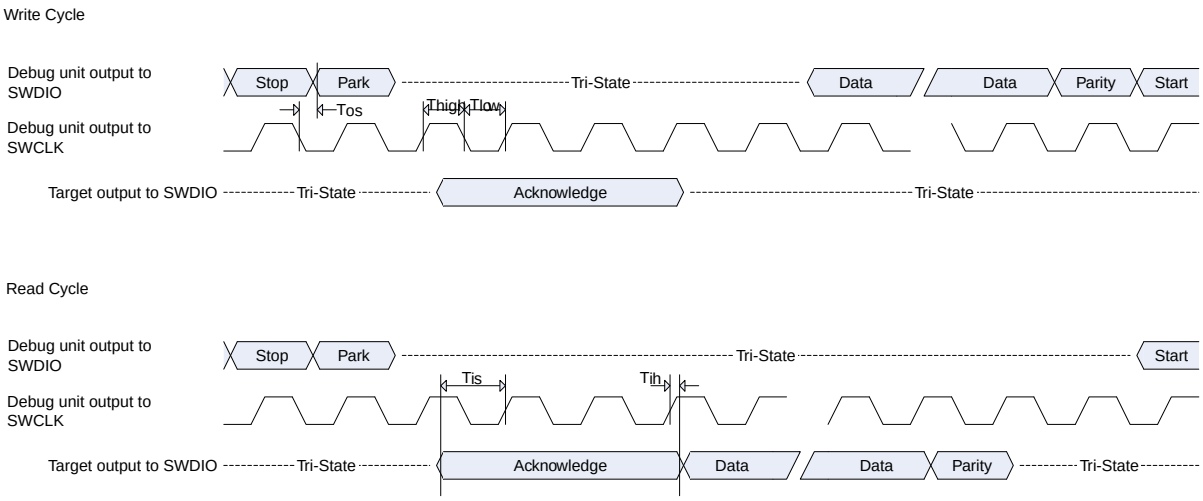
The **SWO** (Serial Wire Output) signal is an output from the target which is often used alongside the **SWD** signals to provide low-bandwidth trace.

When no debug unit is connected, the **SWO** signal must be pulled *high* on the target to keep the signal inactive.

SWD timing requirements

The diagrams in the following figure separate the **SWDIO** line to show when it is driven by either the debug unit or target device:

Figure 2-9: SWD timing diagrams



The debug unit:

- Writes data to **SWDIO** on the falling edge of **SWCLK**.
- Reads data from **SWDIO** on the rising edge of **SWCLK**.

The target:

- Writes data to **SWDIO** on the rising edge of **SWCLK**.
- Reads data from **SWDIO** on the rising edge of **SWCLK**.

The following table shows the timing requirements for **SWD**:

Table 2-2: SWD timing requirements

Parameter	Min	Max	Description
$T_{[high]}$	4ns	50ms	**SWCLK***high* period.
$T_{[low]}$	4ns	50ms	**SWCLK***low* period.
$T_{[os]}$	-1ns	1ns	**SWDIO** output skew to falling edge **SWCLK**.
$T_{[is]}$	4ns	-	Input setup time that is required between **SWDIO** and rising edge **SWCLK**.

Parameter	Min	Max	Description
$T_{[ih]}$	1ns	-	Input hold time that is required between **SWDIO** and rising edge **SWCLK**.

2.6 Parallel Trace signals

Some target devices can output high-bandwidth parallel trace data while the target application is running. Capturing this data and decoding it in Arm® Development Studio allows you to examine the sequence of instructions, and changes in data, around a given point or *trigger*.

For CoreSight™-compliant systems, the main DSTREAM-ST unit supports parallel trace capture of up to 4-bit wide continuous-mode Trace Port Interface Unit (TPIU) formatted trace, at up to 600Mbps per trace signal.

TRACEDATA[0-3]

The **TRACEDATA[0-3]** (Trace Data) signals are single-ended outputs from the target and can be used to collect 1-bit to 4-bit trace data.

TRACECLK

The **TRACECLK** (Trace Clock) signal is a single-ended output from the target which is used to clock the parallel trace data into the debug unit.

The trace clock signal does not need to be phase-shifted from the data signals. By default, the debug unit incorporates the appropriate delay to provide the necessary setup and hold timings for aligned **TRACEDATA** and **TRACECLK** signals.

DSTREAM-ST only supports DDR clocking mode. Parallel trace data is captured on both the rising and falling edges of the trace clock signal.


Note

- Although the debug unit can compensate for large amounts of skew between the parallel trace signals, to avoid the extra calibration step during configuration, Arm recommends matching the lengths of the trace signals within a 10mm window.
- No pull-up or pull-down resistors are required for the trace signals.
- To improve signal integrity, it is good practice to provide impedance matching resistors on the **TRACEDATA** and **TRACECLK** outputs close to the target device. The value of these resistors, added to the impedance of the driver, must be approximately equal to 50Ω.
- To achieve the maximum data rate, Arm recommends using the short 20-way 0.05" pitch ribbon cable.
- For the collection of parallel trace streams wider than 4 bits, consider using a DSTREAM-PT system instead. See the DSTREAM-PT web page.

The following figure and table describe the timing for the **TRACECLK** signal.

Figure 2-10: TRACECLK timing diagram

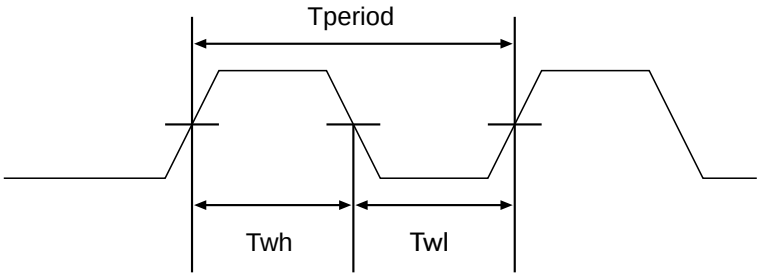


Table 2-3: TRACECLK characteristics

Parameter	Min	Max	Description
Tperiod	3.333ns	125ms	Clock period
Twh	1.667ns	62.5ms	High pulse width
Twl	1.667ns	62.5ms	Low pulse width

Switching thresholds

The debug unit detects the target reference voltage and automatically adjusts its switching thresholds to 50% of this voltage. For example, on a 3.3V target system, the switching thresholds are set to 1.65V.

Leakage current

If an unpowered DSTREAM-ST unit is connected to a powered target, the maximum leakage current is ±10µA per signal.

2.7 Target Voltage Reference signals

The Target Voltage Reference (**VTREF**) signals are used by DSTREAM-ST to determine the correct logic levels of all inputs and outputs of the debug and trace interface.

To work with debug and trace interfaces on differing voltage rails, the DSTREAM-ST unit supports separate debug and trace voltage domains.

VTREF

When using either the CoreSight™ 20 or Arm JTAG 20 connector, only one voltage domain is supported. The voltage domain is determined using the **VTREF** signal.

DEBUG_VTREF

When using the Mictor adapter, or optional MIPI-34 or MIPI-60 adapters, the voltage domain of the debug signals is determined using the **DEBUG_VTREF** signal.

TRACE_VTREF

When using the MIPI-60 cable, Mictor adapter, or optional MIPI-34 or MIPI-60 adapters, the voltage domain of the trace signals is determined using the **TRACE_VTREF** signal.

Note

If only the **TRACE_VTREF** signal is connected on a Mictor, MIPI-34, or MIPI-60 connector of a target, the DSTREAM-ST unit uses that signal to determine the logic levels of both the debug and trace signals.

Arm recommends connecting **VTREF** signals directly to the appropriate power rails on the target board. If a series resistor is used for short-circuit protection, the value used must be less than 100Ω.

VTREF signals that are received by the DSTREAM-ST are loaded with a resistance of approximately 10K to ground. The signals are filtered, limited, and buffered to provide the required VDD (Voh) and reference voltages (Vin(th)) for the I/O stages of the debug unit.

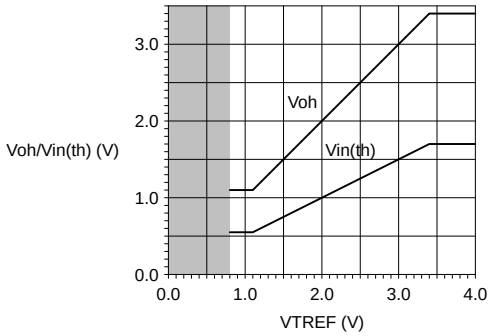
DSTREAM-ST supports debug and trace logic levels between 1.2V and 3.3V.

Note

- To be recognized by DSTREAM-ST as a valid target reference voltage, the **VTREF** signal must be above 800mV. Detection of a valid **VTREF** signal is indicated by the **Target** LED on the DSTREAM-ST unit.
- Logic levels outside the 1.2V to 3.3V window might work, but are not guaranteed to work because the DSTREAM-ST unit internally limits the **VTREF** signal to a minimum of approximately 1.1V, and a maximum of approximately 3.4V.

The relationships of Voh and Vi(th) to **VTREF** are:

Figure 2-11: Target interface logic levels



The input and output characteristics of the DSTREAM-ST system are compatible with logic levels from TTL-compatible, or CMOS logic in target systems.

2.8 PCIe signals

The DSTREAM-XT probe extends the debug and trace functionality of DSTREAM-ST by adding a *PCIe* (Peripheral Component Interconnect Express) interface. It supports up to eight lanes at 8Gbps

(Gen 3), or four lanes at 16Gbps (Gen 4). DSTREAM-XT is also backward compatible with PCIe Gen 1 and 2 (2.5 and 5Gbps respectively). It can act as either a *root port* or an *endpoint*.

When designing a PCIe target board it is strongly advised that the 'PCI Express Base Specification Revision 4.0' and 'PCI Express Card Electromechanical Specification Revision 4.0' are used to define the connector and its signals. To avoid duplication, this document will only briefly describe the supported signals and any specific limitiations imposed by the DSTREAM-XT system.

For full details on the PCIe interface, specifications are available from: pcisig.com.

PET[p/n][0-7]

The **PET** (PCI Express Transmit) signals are differential outputs. The *positive* or *true* signal is denoted by a 'p', while the *negative* or *complementary* signal is denoted by an 'n'. The DSTREAM-XT system supports lanes 0-7 in Gen 3 (8Gbps) mode, or lanes 0-3 in Gen 4 (16Gbps) mode.

PER[p/n][0-7]

The **PER** (PCI Express Receive) signals are differential inputs. The *positive* or *true* signal is denoted by a 'p', while the *negative* or *complementary* signal is denoted by an 'n'. The DSTREAM-XT system supports lanes 0-7 in Gen 3 (8Gbps) mode, or lanes 0-3 in Gen 4 (16Gbps) mode.

The directions of the **PET** and **PER** signals might be misleading. For the standard PCIe expansion slot system, the *PCI Express Card Electromechanical Specification* defines the **PET** (Transmit) and **PER** (Recieve) signals always with respect to the *root port*. In other standards, such as *Mini PCIe* or *OCuLink*, the **PET** and **PER** signals are instead defined with respect to the local device, and then crossed-over at the connector or within the cable.

When designing a target system, refer to the appropriate standard to check that your signal directions are correct.

REFCLK[+/-]

The **REFCLK** (Reference Clock) signals typically form a 100MHz differential signal, generated at the *root port*, to keep the *endpoint* properly sychronised. While the PCIe specification allows for independant **REFCLK** signals to be used by the *root port* and *endpoint*, DSTREAM-XT is only guaranteed to function using a common **REFCLK**. When acting as a *root port*, DSTREAM-XT supplies a 100MHz **REFCLK** signal to the target. When acting as an *endpoint*, it requires a **REFCLK** signal to be supplied by the target.

PERST#

The **PERST#** (PCI Express Reset) signal is an open drain, active *low* output from the *root port*. It is released when all power rails and the **REFCLK** signal have stabilized. The *endpoint* device typically uses this signal as a global reset. When acting as a *root port*, DSTREAM-XT pulls this signal *low* until it is ready to begin link negotiation. When acting as an *endpoint*, DSTREAM-XT waits for this signal to go inactive before attempting link negotiation.

SMBCLK/SMBDAT

The **SMBCLK/SMBDAT** (System Management Bus Clock/Data) signals form an optional two-wire interface which allows a *root port* to communicate with low-speed devices at the *endpoint*, independantly of the PCIe link.

DSTREAM-XT does not currently support these signals.

WAKE#

The **WAKE#** signal is an optional open drain, active *low* signal. It is used by some systems to support sleep/wake functionality.

DSTREAM-XT does not currently support this signal.

CLKREQ#

The **CLKREQ#** (Clock Request) signal is an open drain, active *low* signal. It is used by some systems to request that the **REFCLK** be made active.

DSTREAM-XT does not currently support this signal.

2.9 I/O diagrams for DSTREAM-XT signals

The following diagrams and descriptions show a simplified view of how each signal type is connected within the debug unit.

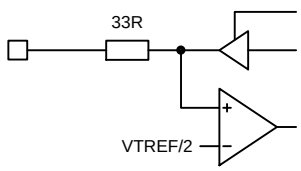
Note

All comparator inputs have an indeterminate band of 100mV above and below **VTREF**/2. Signals that are output from the target system, when passing through this voltage region, must be *monotonic*.

Input/Output signals

Standard input/output signals (**TDI**, **TMS**, **TDO**, **RTCK**, **SWDIO**, **DBGRQ**, **DBGACK**) use *LVCMOS* output buffers and comparator inputs with a series 33Ω resistor.

Figure 2-12: Input/Output signals

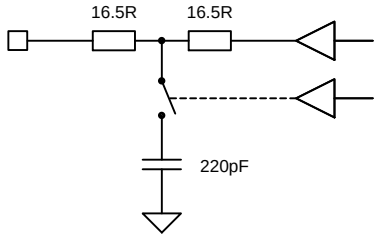


TCK signal

The **TCK** output signal is similar to a standard output signal, but also has a switchable capacitor, forming a T-filter, which can reduce the **TCK** slew-rate.

Enabling this filter is not currently supported in Arm® Development Studio.

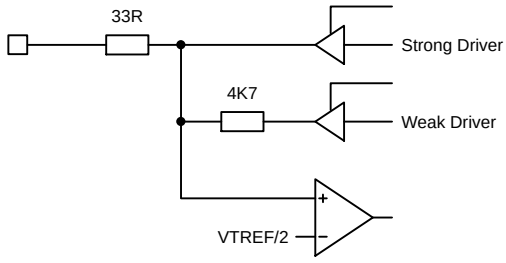
Figure 2-13: TCK signal



Reset signals

The reset signals (**nSRST** and **nTRST**) are similar to the standard input/output signals. However, they have an extra *LVCMOS* driver, which is connected using a 4K7 resistor, that provides the weak pull-up and pull-down functionality.

Figure 2-14: Reset signals



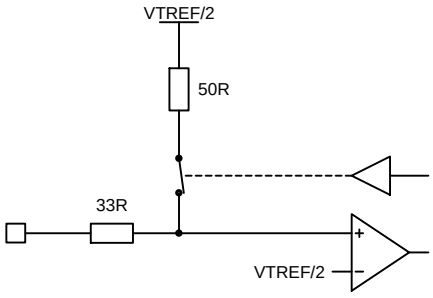
Parallel trace signals

The parallel trace signals (**TRACEDATA[0-3]** and **TRACECLK**) are similar to standard inputs, but are also terminated to **VTREF**/2 through 50Ω resistors. These resistors prevent signals from being reflected back to the target system, increasing signal integrity and the maximum data rate.

Note

Disabling the input terminations is not currently supported in Arm Development Studio.

Figure 2-15: Parallel trace signals



PCIe signals

Refer to the 'PCI Express Base Specification' and 'PCI Express Card Electromechanical Specification' for information on PCIe signalling.

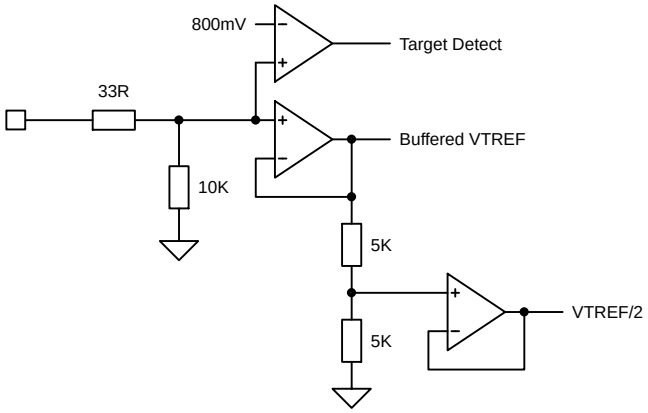
VTREF signals

The **VTREF** signals (**VTREF**, **DEBUG_VTREF** and **TRACE_VTREF**) are buffered. They provide:

- A *VDD* rail for the *LVCMOS* output buffers.
- The **VTREF**/2 reference/termination rail.

For the debug unit to detect that a target is present, the **VTREF** signal must be higher than 800mV.

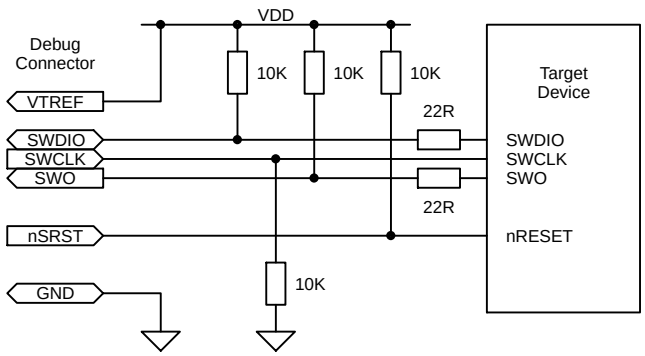
Figure 2-16: VTREF signals



2.10 Typical SWD circuit

A typical SWD circuit:

Figure 2-17: Typical SWD circuit



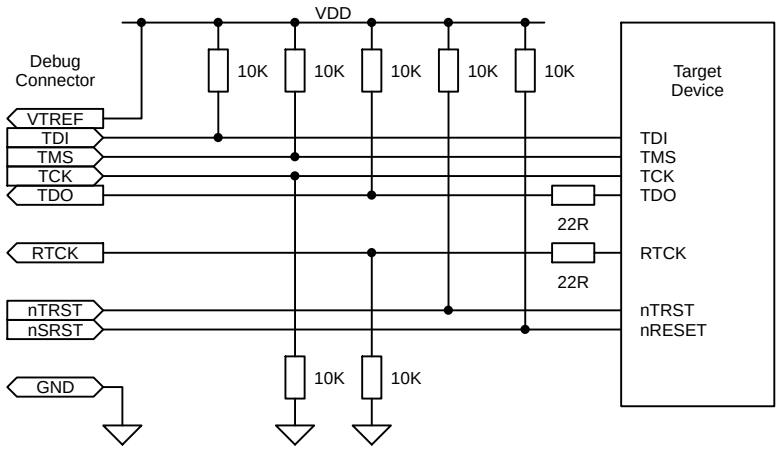
Note

To improve signal integrity, it is good practice to provide an impedance-matching resistor on the **SWDIO** and **SWO** outputs of the target device. The value of these resistors, added to the impedance of the driver, must be approximately equal to 50Ω.

2.11 Typical JTAG circuit

A typical JTAG circuit:

Figure 2-18: Typical JTAG circuit




Note

To improve signal integrity, it is good practice to provide an impedance matching resistor on the **TDO** and **RTCK** outputs of the target device. The value of these resistors, added to the impedance of the driver, must be approximately equal to 50Ω.

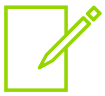
3. Target interface connectors

DSTREAM-ST has an Arm JTAG 20 connector, a CoreSight™ 20 connector, an auxiliary connector, and a user I/O connector.

The XT probe provides an additional connection path using standard PCIe connectors.

To adapt debug connectors for other target connectors, you can use cables and adapter boards. Some of these cables and adapter boards are supplied with the debug unit. Others can be requested from Arm. For a list of provided adapters, see the Arm DSTREAM-XT Getting Started Guide.

If your target system uses a connector which is not currently supported, and you are considering a purchase of Arm DSTREAM-XT, contact Arm support with your requirements. Arm might be able to supply a compatible adapter on a fast-turn, prototype basis.


Note

All connector pinouts in this chapter are shown as they would appear on the target board.

3.1 Target connector selection guide

When choosing a debug or trace connector to design into a target board, there are many connector attributes to consider.

The connector attributes are:

Table 3-1: Connector attributes

Connector	Arm JTAG 20	CoreSight™ 10	CoreSight 20	TI JTAG 14	MICTOR 38	MIPI 34	MIPI 60
JTAG supported	Yes	Yes	Yes	Yes	Yes	Yes	Yes
SWD supported	Yes	Yes	Yes	Yes	Yes	Yes	Yes
SWO trace supported	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Parallel trace supported	No	No	Yes	No	Yes	Yes	Yes
Max parallel trace width [1]	N/A	N/A	4	N/A	16	4	32

[1] The trace width supported by the connector. DSTREAM-ST supports up to 4-bit wide parallel trace.

Connector	Arm JTAG 20	CoreSight™ 10	CoreSight 20	TI JTAG 14	MICTOR 38	MIPI 34	MIPI 60
Separate debug/trace voltage domains	No	No	No	No	Yes	Yes	Yes
Requires adapter	No	No	No	Yes	Yes	Yes	Yes
Cable signal integrity	Medium	Good	Good	Poor	Excellent	Good	Excellent
MIPI pinout compatible	No	Yes	Yes	No	No	Yes	Yes
Target connector cost	Low	Low	Low	Low	High	Low	Medium
Connector durability [2]	High	Low	Low	High	Medium	Low	Medium
Approximate footprint area (mm^2) [3]	297	65	95	250	221	140	170
Through-hole/ SMD	Either	Either	Either	Either	SMD [4]	Either	SMD
Ease of assembly (placement/ soldering)	High	High	High	High	Low	High	Medium

3.2 Arm JTAG 20 connector

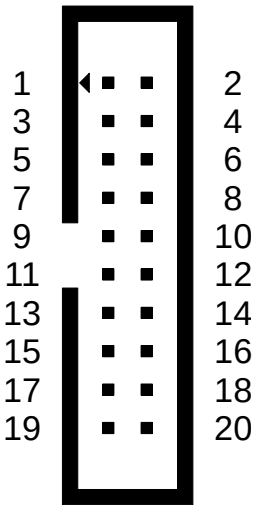
The Arm JTAG 20 connector is a 20-way 2.54mm pitch box header which supports JTAG debug, Serial Wire Debug, and SWO trace.

Using a non-shrouded header on the target board can lead to short-circuits or signal contention. To ensure the correct polarity and position, Arm recommends that you use a fully shrouded box header.

To use this connector with DSTREAM-ST, use the supplied Arm JTAG 20 debug cable.

[2] Through-hole variants of connectors are more durable than SMD variants.
[3] Assumes an SMD part is used. Through-hole parts use additional space on the opposite side of the board.
[4] The Mictor 38 connector is a hybrid part that has SMD signal pins and through-hole ground pins. The ground pins must be solder-pasted from the component side. The Mictor 38 connector is not recommended for future designs.

Figure 3-1: Arm JTAG 20 connector pinout



Arm JTAG 20 pinout table

Table 3-2: Arm JTAG 20 pinout table

Pin	Signal name		Pin	Signal name
1	VTREF		2	NC
3	nTRST		4	GND
5	TDI		6	GND
7	TMS/SWDIO		8	GND
9	TCK/SWCLK		10	GND
11	RTCK		12	GND
13	TDO/SWO		14	GND
15	nSRST		16	GND
17	DBGRQ		18	GND
19	DBGACK		20	GND

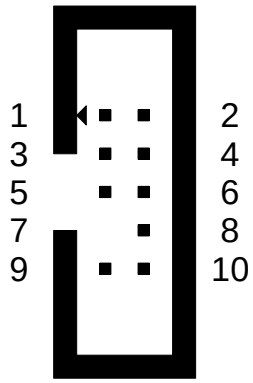
3.3 CoreSight 10 connector

The CoreSight™ 10 connector is a 10-way 1.27mm pitch box header which supports JTAG debug, Serial Wire Debug, and SWO trace.

Using a non-shrouded header on the target board can lead to short-circuits or signal contention. To ensure the correct polarity and position, Arm recommends that you use a fully shrouded box header.

To use this connector with DSTREAM-ST, use the supplied CoreSight 10/20 debug cable.

Figure 3-2: CoreSight 10 connector pinout



CoreSight 10 pinout table

Table 3-3: Arm CoreSight 10 pinout table

Pin	Signal name		Pin	Signal name
1	**VTREF**		2	**TMS/SWDIO**
3	**GND**		4	**TCK/SWCLK**
5	**GND**		6	**TDO/SWO**
7	Key (**NC**) [5]		8	**TDI**
9	**GND**		10	**nSRST**

3.4 CoreSight 20 connector

The CoreSight™ 20 connector is a 20-way 1.27mm pitch box header which supports JTAG debug, Serial Wire Debug, SWO trace, and up to 4-bit wide continuous-mode TPIU trace.

Caution

Using a non-shrouded header on the target board can lead to short-circuits or signal contention. To ensure the correct polarity and position, Arm recommends that you use a fully shrouded box header.

To use CoreSight 20 connector with the DSTREAM-ST unit, use the supplied CoreSight 20 debug cable.

You must configure the pinout mode of the CoreSight 20 connector before using it.

Caution

If you do not configure the CoreSight 20 connector correctly, your DSTREAM-ST unit might not operate correctly. For example, if the target connector has been pinned-out for debug-only mode and DSTREAM-ST is set to debug and trace mode,

[5] Pin 7 must be removed for compatibility with CoreSight and MIPI specifications.

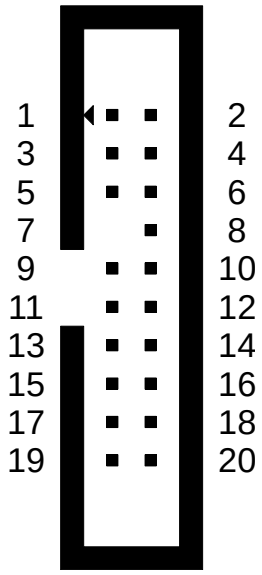
the target's test access port could inadvertantly be held in reset by the **nTRST** signal.

To configure the pinout mode, use the Platform Configuration Editor (PCE) in Arm® Development Studio. In the PCE, select **Debug Adapter**, then select the **Probe Configuration** tab. In the configuration items table, set the **DSTREAMCS20** configuration item to:

- `0`: to use the connector in JTAG debug and trace mode.
- `1`: to use the connector in JTAG debug-only mode.

For more information, see Configure your debug hardware unit for platform autodetection in the Arm Development Studio User Guide.

Figure 3-3: CoreSight 20 connector pinout



CoreSight 20 pinout tables

Table 3-4: Arm CoreSight 20 pinout table (`DSTREAMCS20=0`)

Pin	Signal name		Pin	Signal name
1	**VTREF**		2	**TMS/SWDIO**
3	**GND**		4	**TCK/SWCLK**
5	**GND**		6	**TDO/SWO**
7	Key (NC) [6]		8	**TDI**
9	**GND**		10	**nSRST**
11	**GND** [7]		12	**TRACECLK**

[6] Pin 7 must be removed for compatibility with CoreSight and MIPI specifications.
[7] Although pins 11 and 13 are typically grounded on the target board, the MIPI specification also allows them to carry power. If they are connected to a power rail (or rails) on the target board, these pins must also be AC-coupled to **GND**. To couple the pins to **GND**, use 100nF capacitors that are close to the connector.

Pin	Signal name		Pin	Signal name
13	**GND** [7]		14	**TRACEDATA[0]**
15	**GND**		16	**TRACEDATA[1]**
17	**GND**		18	**TRACEDATA[2]**
19	**GND**		20	**TRACEDATA[3]**

Table 3-5: Arm CoreSight 20 pinout table (DSTREAMCS20=1)

Pin	Signal name		Pin	Signal name
1	**VTREF**		2	**TMS/SWDIO**
3	**GND**		4	**TCK/SWCLK**
5	**GND**		6	**TDO/SWO**
7	Key (NC) [6]		8	**TDI**
9	**GND**		10	**nSRST**
11	**GND** [7]		12	**RTCK**
13	**GND** [7]		14	**SWO**
15	**GND**		16	**nTRST**
17	**GND**		18	**DBGRQ**
19	**GND**		20	**DBGACK**

3.5 TI JTAG 14 connector

The Texas Instruments (TI) JTAG 14 connector is a 14-way 2.54mm pitch box header which supports JTAG debug, Serial Wire Debug, and SWO trace.

- Using a non-shrouded header on the target board can lead to short-circuits or signal contention. To ensure the correct polarity and position, Arm recommends that you use a fully shrouded box header.
- To avoid cross-talk issues between **nTRST**, **TMS**, and **TDI**, you must connect the DSTREAM-ST TI JTAG 14 adapter directly to the target board.

To use this connector with DSTREAM-ST, the supplied TI JTAG 14 adapter must be used with the Arm JTAG 20 debug cable.

Figure 3-4: TI JTAG 14 connector pinout

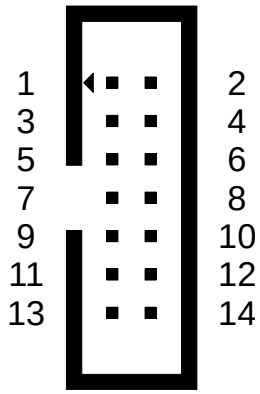


Table 3-6: TI JTAG 14 pinout table

Pin	Signal name		Pin	Signal name
1	**TMS/SWDIO**		2	**nTRST**
3	**TDI**		4	**GND**
5	**VTREF**		6	NC
7	**TDO/SWO**		8	**GND**
9	**RTCK**		10	**GND**
11	**TCK/SWCLK**		12	**GND**
13	**DBGRQ**		14	**DBGACK**

3.6 Mictor 38 connector

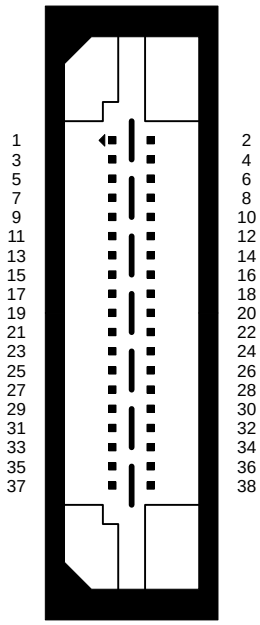
The Mictor 38 connector is a 38-way 0.635mm pitch socket which supports JTAG debug, Serial Wire Debug, SWO trace, and up to 16-bit wide, continuous-mode Trace Port Interface Unit (TPIU) trace.

The center ground pins of the Mictor socket must be solder-pasted on the same side of the PCB as the connector. If you do not solder-paste on the same side of the PCB as the connector, it might cause mechanical or signal integrity issues.

Typically, the socket used is part number 2-5767004-2 from TE Connectivity.

To use this connector with DSTREAM-ST, the supplied 4-bit Mictor adapter must be used in conjunction with both the Arm JTAG 20 debug cable and the CoreSight™ 20 debug cable.

Figure 3-5: Mictor 38 connector pinout



Mictor 38 pinout table

Table 3-7: Mictor 38 pinout table

Pin	Signal name		Pin	Signal name
1	NC		2	NC
3	NC		4	NC
5	**GND**		6	**TRACECLK**
7	**DBGRQ**		8	**DBGACK**
9	**nSRST**		10	**EXTTRIG** [8]
11	**TDO**		12	**TRACE_VTREF** [9]
13	**RTCK**		14	**DEBUG_VTREF** [9]
15	**TCK**		16	**TRACEDATA[7]**
17	**TMS**		18	**TRACEDATA[6]**
19	**TDI**		20	**TRACEDATA[5]**
21	**nTRST**		22	**TRACEDATA[4]**
23	**TRACEDATA[15]**		24	**TRACEDATA[3]**
25	**TRACEDATA[14]**		26	**TRACEDATA[2]**
27	**TRACEDATA[13]**		28	**TRACEDATA[1]**

[8] The **EXTTRIG** signal is deprecated. It is not supported by Arm Development Studio.
[9] Although the Arm CoreSight specification only supports a single **VTREF** (on pin 12), DSTREAM-ST can support separate debug and trace **VTREF**s. If only **TRACE_VTREF** is powered, the DSTREAM-ST assumes that both debug and trace are to operate at that voltage.

Pin	Signal name		Pin	Signal name
29	**TRACEDATA[12]**		30	Logic 0 [10]
31	**TRACEDATA[11]**		32	Logic 0 [10]
33	**TRACEDATA[10]**		34	Logic 1 [10]
35	**TRACEDATA[9]**		36	**TRACECTL** [11]
37	**TRACEDATA[8]**		38	**TRACEDATA[0]**

3.7 MIPI 34 connector

The MIPI 34 connector is a 34-way 1.27mm pitch box header which supports JTAG debug, Serial Wire Debug, SWO trace, and up to 4-bit wide continuous-mode TPIU trace.


Note

This connector is rarely used on target boards. The MIPI 34 adapter and debug cable are not supplied with the DSTREAM-XT system, but are available on request.

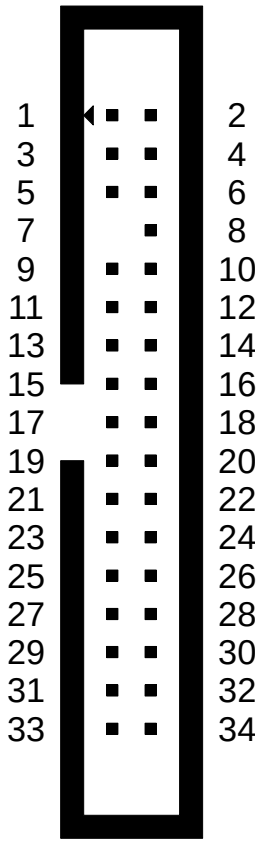

Note

class: caution

- Using a non-shrouded header on the target board can lead to short-circuits or signal contention. To ensure the correct polarity and position, Arm recommends that you use a fully-shrouded box header.
- The MIPI 34 connector supports separate voltage domains for the debug and trace signals. You must supply the appropriate voltages to both of the **VTREF** pins.

[10] **Logic 0** and **Logic 1** signals are not used by DSTREAM-ST. To maintain compatibility with other debug units, connect the signals to the appropriate power rails.
[11] The **TRACECTL** signal is not supported by DSTREAM-ST since only continuous-mode TPIU trace is supported.

Figure 3-6: MIPI 34 connector pinout



MIPI 34 pinout table

Table 3-8: MIPI 34 pinout table

Pin	Signal name		Pin	Signal name
1	**DEBUG_VTREF**		2	**TMS/SWDIO**
3	**GND**		4	**TCK/SWCLK**
5	**GND**		6	**TDO/SWO**
7	Key (NC) [12]		8	**TDI**
9	**GND**		10	**nSRST**
11	**GND** [13]		12	**RTCK**
13	**GND** [13]		14	**TRST_PD** [14]
15	**GND**		16	**nTRST**
17	**GND**		18	**DBGRQ**

[12] Pin 7 must be removed for compatibility with DSTREAM-ST and MIPI specifications.
[13] Although pins 11 and 13 are typically grounded on the target board, the MIPI specification also allows them to carry power. If they are connected to power rail (or rails) on the target board, these pins must also be AC coupled to **GND** using 100nF capacitors that are close to the connector.
[14] The **TRST_PD** signal allows the target board to have a second TAP reset signal which is normally pulled-down. For more information, see the MIPI debug connector specification.

Pin	Signal name		Pin	Signal name
19	**GND**		20	**DBGACK**
21	**GND**		22	**TRACECLK**
23	**GND**		24	**TRACEDATA[0]**
25	**GND**		26	**TRACEDATA[1]**
27	**GND**		28	**TRACEDATA[2]**
29	**GND**		30	**TRACEDATA[3]**
31	**GND**		32	**TRACEEXT** [15]
33	**GND**		34	**TRACE_VTREF**

3.8 MIPI 60 connector

The MIPI 60 connector is a 60-way 0.5mm pitch socket which supports JTAG debug, Serial Wire Debug, SWO trace, and up to 32-bit wide continuous-mode TPIU trace.

Typically, the socket is a QSH-030-01-L-D-A from Samtec.

Note

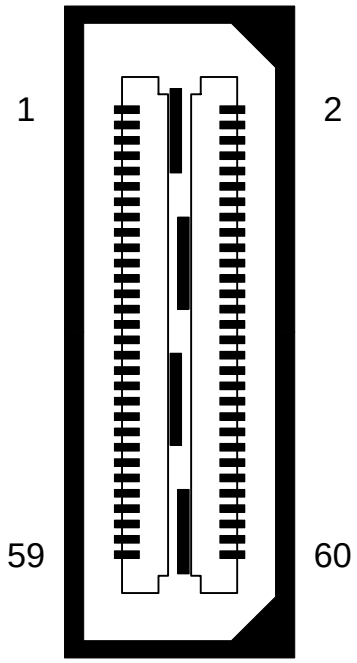
- DSTREAM-ST only supports one channel of parallel trace.
- DSTREAM-ST is only capable of capturing up to 4-bit wide trace (**TRACEDATA[0-3]**).
- The MIPI 60 connector is not supplied with DSTREAM-XT, but is available on request. To request one, contact Arm support.
- To use this connector with DSTREAM-ST, a MIPI 60 adapter must be used in conjunction with the 4-bit Mictor adapter, Arm JTAG 20 debug cable and the CoreSight™ 20 debug cable.

Caution

- The MIPI 60 connector supports separate voltage domains for the debug and trace signals. It is necessary to supply the appropriate voltages to both of the **VTREF** pins.
- Due to variations in naming conventions, care must be taken when mapping the trace pins from the target device to the MIPI-60 connector. The table given here is correct for ETMv3 or later protocols.

[15] The **TRACEEXT** signal is not supported by DSTREAM-ST.

Figure 3-7: MIPI 60 connector pinout



MIPI 60 pinout table

Table 3-9: MIPI 60 pinout table

Pin	Signal name		Pin	Signal name
1	DEBUG_VTREF		2	TMS/SWDIO
3	TCK		4	TDO
5	TDI		6	nSRST
7	RTCK		8	TRST_PD [16]
9	nTRST		10	DBGRQ
11	DBGACK		12	TRACE_VTREF
13	TRACECLK		14	RESERVED [17]
15	GND		16	GND
17	TRACECTL [18]		18	TRACEDATA[19]
19	TRACEDATA[0]		20	TRACEDATA[20]
21	TRACEDATA[1]		22	TRACEDATA[21]
23	TRACEDATA[2]		24	TRACEDATA[22]
25	TRACEDATA[3]		26	TRACEDATA[23]
27	TRACEDATA[4]		28	TRACEDATA[24]
29	TRACEDATA[5]		30	TRACEDATA[25]

[16] The **TRST_PD** signal allows the target board to have a second TAP reset signal which is normally pulled-down. For more information, see the MIPI debug connector specification.
[17] Pins marked as RESERVED might be internally connected in DSTREAM-ST, but are not currently supported.
[18] DSTREAM-ST ignores the TRACECTL pin since only continuous-mode (TPIU) trace is supported.

Pin	Signal name		Pin	Signal name
31	**TRACEDATA[6]**		32	**TRACEDATA[26]**
33	**TRACEDATA[7]**		34	**TRACEDATA[27]**
35	**TRACEDATA[8]**		36	**TRACEDATA[28]**
37	**TRACEDATA[9]**		38	**TRACEDATA[29]**
39	**TRACEDATA[10]**		40	**TRACEDATA[30]**
41	**TRACEDATA[11]**		42	**TRACEDATA[31]**
43	**TRACEDATA[12]**		44	RESERVED [17]
45	**TRACEDATA[13]**		46	RESERVED [17]
47	**TRACEDATA[14]**		48	RESERVED [17]
49	**TRACEDATA[15]**		50	RESERVED [17]
51	**TRACEDATA[16]**		52	RESERVED [17]
53	**TRACEDATA[17]**		54	RESERVED [17]
55	**TRACEDATA[18]**		56	RESERVED [17]
57	**GND**		58	**GND**
59	RESERVED [17]		60	RESERVED [17]

3.9 PCIe Card connectors

The PCIe card slot connector is often used on system motherboards to enable PCIe expansion cards to be added, such as graphics or network adapter cards.

DSTREAM-XT supports connections to any standard length PCIe slot, and allows PCIe connections of x1, x2, x4 or x8 lanes. DSTREAM-XT also supports target boards which are in the PCIe card format. It can act as a fully-functional root port in place of a motherboard. For more details about the specification of these connectors, refer to the 'PCI Express Card Electromechanical Specification'.

DSTREAM-XT supports up to eight PCIe lanes in Gen 3 mode (8Gbps) or up to four lanes in Gen 4 mode (16Gbps).

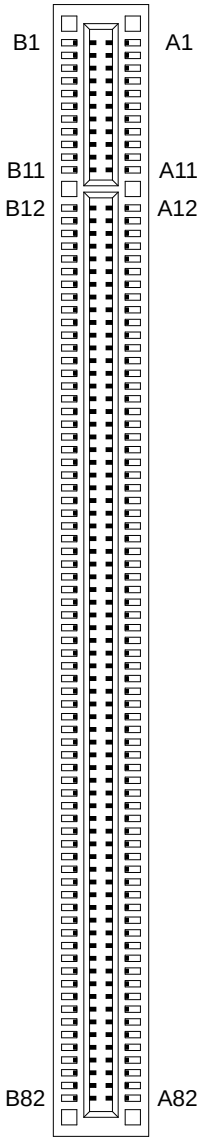
The PCIe card adapters are bare PCBs which are powered by the XT probe. When not connected to a target system, keep the adapter away from conductive surfaces or other shorting hazards. If a short does occur, DSTREAM-XT disables the adapter power until the unit has been reset or power-cycled.

To use a PCIe slot connector with DSTREAM-XT, the appropriate card edge adapter (x1/x4/x8) must be used with the 98-way ribbon cable.

To use a PCIe edge connector with DSTREAM-XT, the x16 card slot adapter must be used with the 98-way ribbon cable.

The full x16 PCIe connector is shown here with the limitations of each connector size shown in the pin-out table.

Figure 3-8: PCIe x16 slot connector pinout



PCIe x1, x4, x8, x16 pinout table

Table 3-10: PCIe x1, x4, x8, x16 pin-out table

Pin	Signal name		Pin	Signal name
B1	**+12V**		A1	**PRSNT1#**
B2	**+12V**		A2	**+12V**
B3	**+12V**		A3	**+12V**

Pin	Signal name		Pin	Signal name
B4	**GND**		A4	**GND**
B5	**SMBCLK**		A5	**TCK**
B6	**SMBDAT**		A6	**TDI**
B7	**GND**		A7	**TDO**
B8	**+3.3V**		A8	**TMS**
B9	**nTRST**		A9	**+3.3V**
B10	**+3.3Vaux**		A10	**+3.3V**
B11	**WAKE#**		A11	**PERST#**
Mechanical Key	-		-	-
B12	**CLKREQ#**		A12	**GND**
B13	**GND**		A13	**REFCLK+**
B14	**PETp0**		A14	**REFCLK-**
B15	**PETn0**		A15	**GND**
B16	**GND**		A16	**PERp0**
B17	**PRSNT2#**		A17	**PERn0**
B18	**GND**		A18	**GND**
End of the x1 Connector	-		-	-
B19	**PETp1**		A19	RSVD
B20	**PETn1**		A20	**GND**
B21	**GND**		A21	**PERp1**
B22	**GND**		A22	**PERn1**
B23	**PETp2**		A23	**GND**
B24	**PETn2**		A24	**GND**
B25	**GND**		A25	**PERp2**
B26	**GND**		A26	**PERn2**
B27	**PETp3**		A27	**GND**
B28	**PETn3**		A28	**GND**
B29	**GND**		A29	**PERp3**
B30	**PWRBRK#**		A30	**PERn3**
B31	**PRSNT2#**		A31	**GND**
B32	**GND**		A32	RSVD
End of the x4 connector	-		-	-
B33	**PETp4**		A33	RSVD
B34	**PETn4**		A34	**GND**
B35	**GND**		A35	**PERp4**
B36	**GND**		A36	**PERn4**
B37	**PETp5**		A37	**GND**
B38	**PETn5**		A38	**GND**
B39	**GND**		A39	**PERp5**
B40	**GND**		A40	**PERn5**

Pin	Signal name		Pin	Signal name
B41	**PETp6**		A41	**GND**
B42	**PETn6**		A42	**GND**
B43	**GND**		A43	**PERp6**
B44	**GND**		A44	**PERn6**
B45	**PETp7**		A45	**GND**
B46	**PETn7**		A46	**GND**
B47	**GND**		A47	**PERp7**
B48	**PRSNT2#**		A48	**PERn7**
B49	**GND**		A49	**GND**
End of the x8 Connector	-		-	-
B50	**PETp8**		A50	RSVD
B51	**PETn8**		A51	**GND**
B52	**GND**		A52	**PERp8**
B53	**GND**		A53	**PERn8**
B54	**PETp9**		A54	**GND**
B55	**PETn9**		A55	**GND**
B56	**GND**		A56	**PERp9**
B57	**GND**		A57	**PERn9**
B58	**PETp10**		A58	**GND**
B59	**PETn10**		A59	**GND**
B60	**GND**		A60	**PERp10**
B61	**GND**		A61	**PERn10**
B62	**PETp11**		A62	**GND**
B63	**PETn11**		A63	**GND**
B64	**GND**		A64	**PERp11**
B65	**GND**		A65	**PERn11**
B66	**PETp12**		A66	**GND**
B67	**PETn12**		A67	**GND**
B68	**GND**		A68	**PERp12**
B69	**GND**		A69	**PERn12**
B70	**PETp13**		A70	**GND**
B71	**PETn13**		A71	**GND**
B72	**GND**		A72	**PERp13**
B73	**GND**		A73	**PERn13**
B74	**PETp14**		A74	**GND**
B75	**PETn14**		A75	**GND**
B76	**GND**		A76	**PERp14**
B77	**GND**		A77	**PERn14**
B78	**PETp15**		A78	**GND**
B79	**PETn15**		A79	**GND**

Pin	Signal name		Pin	Signal name
B80	**GND**		A80	**PERp15**
B81	**PRSNT2#**		A81	**PERn15**
B82	RSVD		A82	**GND**

Connector-specific signals

PRSNT1#/PRSNT2#

The **PRSNT1#/PRSNT2#** (presence detection) signals are used by an add-in PCIe card, to indicate to a motherboard (root port) that the card is present and the number of lanes that its interface can support. These signals are supported by all DSTREAM-XT adapters. See the PCI Express Card Electromechanical Specification Revision 4.0 for further details on their usage.

PWRBRK#

The **PWRBRK#** (Power Break) signal is an open drain, active *low* signal, used by some systems to request that the endpoint enters a low power state to conserve power. DSTREAM-XT does not currently support this signal.

+12V

The **+12V** pins of the PCIe connector are normally used by a motherboard to power the attached PCIe card. While DSTREAM-XT does not draw any power from these pins, you can use them for target power detection in place of a **VTREF** signal. If the target board is in the PCIe card format, you can attach a separate ATX power supply to the supplied x16 card slot adapter, if needed.

+3.3V

The **+3.3V** pins of the PCIe connector are normally used by a motherboard to power the attached PCIe card. While DSTREAM-XT does not draw any power from these pins, you can use them for target power detection in place of a **VTREF** signal. If the target board is in the PCIe card format, you can attach a separate ATX power supply to the supplied x16 card slot adapter, if needed.

+3.3Vaux

The **+3.3Vaux** pins of the PCIe connector are normally used by a motherboard to power the attached PCIe card when in standby mode. While DSTREAM-XT does not draw any power from these pins, you can use them for target power detection in place of a **VTREF** signal. If the target board is in the PCIe card format, you can attach a separate ATX power supply to the supplied x16 card slot adapter, if needed.

RSVD

The **RSVD** pins are reserved by the PCIe specification and are not connected on DSTREAM-XT adapters.

3.10 Mini-PCIe Connector

The PCI Express Mini (or *Mini-PCIe*) card slot connector is often used on development boards and laptop motherboards to enable Mini-PCIe expansion cards to be added, such as wireless LAN/ WAN adapter cards.

The Mini-PCIe adapter is a bare PCB which is powered by the XT probe. When not connected to a target system, keep the adapter away from conductive surfaces or other shorting hazards. If a short does occur, DSTREAM-XT disables the adapter power until the unit has been reset or power-cycled.

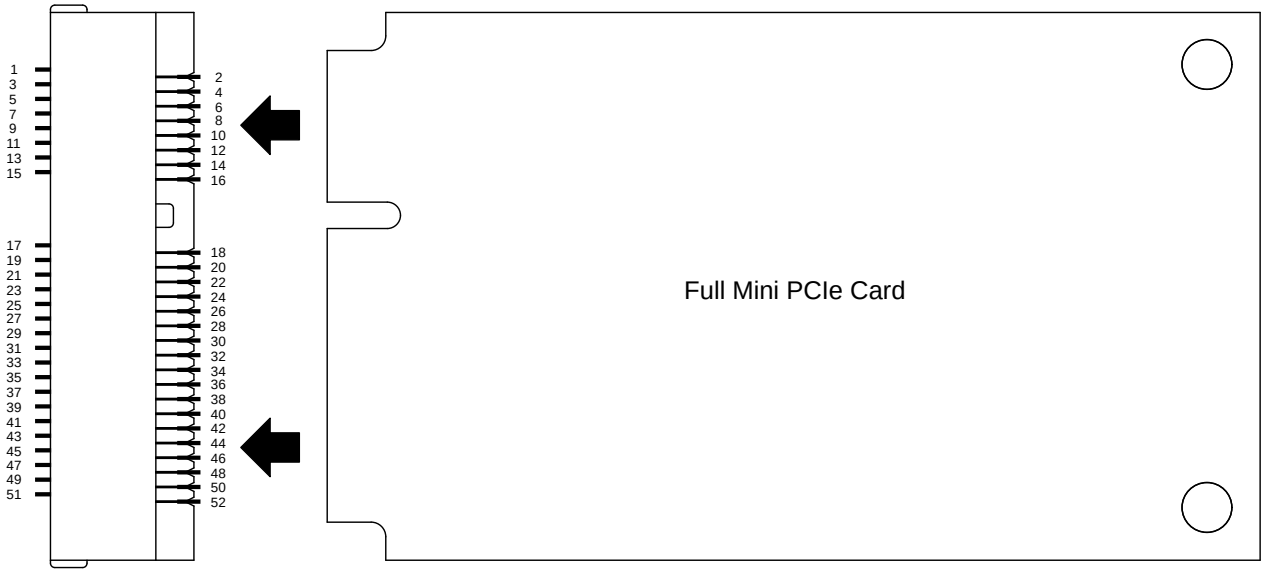
DSTREAM-XT supports an optional Mini-PCIe adapter which can be used in place of a full sized (30.00x50.95mm) Mini-PCIe card and allows an x1 PCIe connection. For further details about the specification of this connector, please refer to the 'PCI Express Mini Card Electromechanical Specification'.

- The Mini-PCIe adapter is not supplied with DSTREAM-XT but is available on request, with any purchase of DSTREAM-XT.
- While no significant current is drawn from the power pins of this connector, DSTREAM-XT can use them to detect the power state of the target system in place of a dedicated **VTREF** signal.

To use a Mini-PCIe slot connector with DSTREAM-XT, the Mini-PCIe adapter must be used in conjunction with the 98-way ribbon cable.

The 98-way ribbon cable extends directly upward from the component side of the Mini-PCIe adapter and might not be compatible with target system where this height is restricted.

Figure 3-9: Mini-PCIe connector pinout



Mini-PCIe pinout table

Table 3-11: Mini-PCIe pinout table

Pin	Signal name		Pin	Signal name
1	**WAKE#**		2	**3.3Vaux**
3	**COEX1**		4	**GND**
5	**COEX2**		6	**1.5V/COEX3**
7	**CLKREQ#**		8	**UIM_PWR**
9	**GND**		10	**UIM_DATA**
11	**REFCLK-**		12	**UIM_CLK**
13	**REFCLK+**		14	**UIM_RESET**
15	**GND**		16	**UIM_SPU**
Mechanical Key	-		-	-
17	**UIM_IC_DM**		18	**GND**
19	**UIM_IC_DP**		20	**W_DISABLE1#**
21	**GND**		22	**PERST#**
23	**PERn0/SSRX-**		24	**+3.3Vaux**
25	**PERp0/SSRX+**		26	**GND**
27	**GND**		28	**+1.5V/ ANTCTRL0**
29	**GND**		30	**SMB_CLK**
31	**PETn0/SSTX-**		32	**SMB_DATA**
33	**PETp0/SSTX+**		34	**GND**
35	**GND**		36	**USB_D-**
37	**GND**		38	**USB_D+**
39	**+3.3Vaux**		40	**GND**
41	**+3.3Vaux**		42	**LED_WWAN#**

Pin	Signal name		Pin	Signal name
43	**GND**		44	**LED_WLAN#**
45	**ANTCTRL2**		46	**LED_WPAN#**
47	**ANTCTRL3**		48	**+1.5V/ ANTCTRL1**
49	RESERVED		50	**GND**
51	**W_DISABLE2#**		52	**+3.3Vaux**

Connector-specific signals

These signals are not supported by DSTREAM-XT and are disconneted on the Mini-PCIe adapter:

- **COEX[1/2/3]**
- **UIM_[*]**
- **W_DISABLE[1/2]#**
- **ANTCTRL[0/1/2/3]**
- **USB_D[+/-]**
- **LED_W[W/L/P]AN#**
- **Reserved**

SS[R/T]X[+/-]

The **SS[R/T]X[+/-]** (USB Super Speed) signals are not supported by DSTREAM-XT. You can only use these pins in PCIe mode.

3.11 OCuLink Connector

The OCuLink connector is often used when PCIe is required to operate via a cable, typically between a motherboard and high-speed storage device.

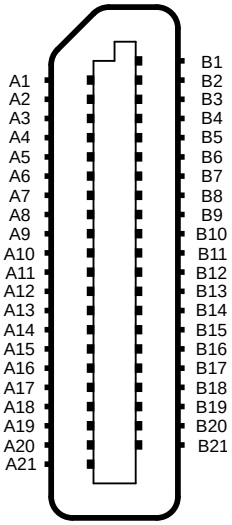
The OCuLink adapter is a bare PCB which is powered by the XT probe. When not connected to a target system, keep the adapter away from conductive surfaces or other shorting hazards. If a short does occur, DSTREAM-XT disables the adapter power until the unit has been reset or power-cycled.

DSTREAM-XT supports an optional OCuLink adapter which you can use in either *root port* or *endpoint* modes, with PCIe lane widths of x1, x2 or x4 at up to 16Gbps (Gen 4). For more details about the specification of this connector, refer to the 'PCI Express OCuLink Specification'.

The OCuLink adapter is not supplied with DSTREAM-XT but is available on request, with any purchase of DSTREAM-XT.

To use an OCuLink connector with DSTREAM-XT, the OCuLink adapter must be used with an appropriate OCuLink cable. The OCuLink adapter does not require the 98-way ribbon cable, but instead plugs directly into the DSTREAM-XT probe unit and provides the required OCuLink connector.

Figure 3-10: OCuLink connector pinout



OCuLink pinout table (root port)

This pinout is used at the *root port* end of the connection, for example on a motherboard.

Table 3-12: OCuLink pinout table (root port)

Pin	Signal name		Pin	Signal name
Row offset - No pin	-		B1	**POWER 5V**
A1	**POWER 3.3Vact**		B2	**GROUND**
A2	**GROUND**		B3	**PETp0**
A3	**PERp0**		B4	**PETn0**
A4	**PERn0**		B5	**GROUND**
A5	**GROUND**		B6	**PETp1**
A6	**PERp1**		B7	**PETn1**
A7	**PERn1**		B8	**GROUND**
A8	**GROUND**		B9	**2-WIRE CLOCK**
A9	**BP TYPE/VSP**		B10	**2-WIRE DATA**
A10	**CWAKE#**		B11	**GROUND**
A11	**GROUND**		B12	**PERST#**
A12	**VSP/REFCLK+**		B13	**CPRSNT#**
A13	**VSP/REFCLK-**		B14	**GROUND**
A14	**GROUND**		B15	**PETp2**
A15	**PERp2**		B16	**PETn2**
A16	**PERn2**		B17	**GROUND**

Pin	Signal name		Pin	Signal name
A17	**GROUND**		B18	**PETp3**
A18	**PERp3**		B19	**PETn3**
A19	**PERn3**		B20	**GROUND**
A20	**GROUND**		B21	**POWER 3.3Vact**
A21	**POWER 5V**		Row offset - No pin	-

This pinout is used at the *endpoint* of the connection, for example on a disk-drive.

Table 3-13: OCuLink pinout table (endpoint)

Pin	Signal name		Pin	Signal name
Row offset - No pin	-		B1	**POWER 5V**
A1	**POWER 3.3Vact**		B2	**GROUND**
A2	**GROUND**		B3	**PETp0**
A3	**PERp0**		B4	**PETn0**
A4	**PERn0**		B5	**GROUND**
A5	**GROUND**		B6	**PETp1**
A6	**PERp1**		B7	**PETn1**
A7	**PERn1**		B8	**GROUND**
A8	**GROUND**		B9	**BP TYPE/VSP**
A9	**2-WIRE_CLOCK**		B10	**CWAKE#**
A10	**2-WIRE_DATA**		B11	**GROUND**
A11	**GROUND**		B12	**VSP/REFCLK+**
A12	**PERST#**		B13	**VSP/REFCLK-**
A13	**CPRSNT#**		B14	**GROUND**
A14	**GROUND**		B15	**PETp2**
A15	**PERp2**		B16	**PETn2**
A16	**PERn2**		B17	**GROUND**
A17	**GROUND**		B18	**PETp3**
A18	**PERp3**		B19	**PETn3**
A19	**PERn3**		B20	**GROUND**
A20	**GROUND**		B21	**POWER 3.3Vact**
A21	**POWER 5V**		Row offset - No pin	-


Note

- While the **REFCLK** signals are an optional part of the OCuLink specification, they are required by DSTREAM-XT to maintain proper synchronisation.
- In the OCuLink specification the **PET** and **PER** signals are with respect to the local device. These signals are then crossed-over by the cable. For example, **PET** signals of the *root port* connect to **PER** signals of the *endpoint*.

Connector-specific signals

2-WIRE CLOCK/DATA

The **2-WIRE CLOCK/DATA** signals are equivalent to the **SMBCLK/DAT** signals used in the standard PCIe slot interface.

These signals are not currently supported by DSTREAM-XT.

CWAKE#

The **CWAKE#** signal is equivalent to the **WAKE#** signal used in the standard PCIe slot interface.

This signal is not connected on the OCuLink adapter.

BP TYPE

The **BP TYPE** signal is used by the *root port* to indicate to the *endpoint* whether or not the 2-wire interface uses I2C signalling.

This signal is not connected on the OCuLink adapter.

CPRSNT#

The **CPRSNT#** (Cable Present) signal is an active *low* signal that is used by the *endpoint*, to indicate to the *root port* that a cable is present and that the *endpoint* is powered.

This signal is used by DSTREAM-XT to indicate to the target system that it is connected and ready for PCIe ennumeration.

VSP

The **VSP** (Vendor Specific) signals are not supported by DSTREAM-XT.

3.12 M.2 Connector

The M.2 connector is primarily used to support solid-state drives which you can mount directly on a motherboard.

The M.2 adapter is a bare PCB which is powered by the XT probe. When not connected to a target system, keep the adapter away from conductive surfaces or other shorting hazards. If a short does occur, DSTREAM-XT disables the adapter power until the unit has been reset or power-cycled.

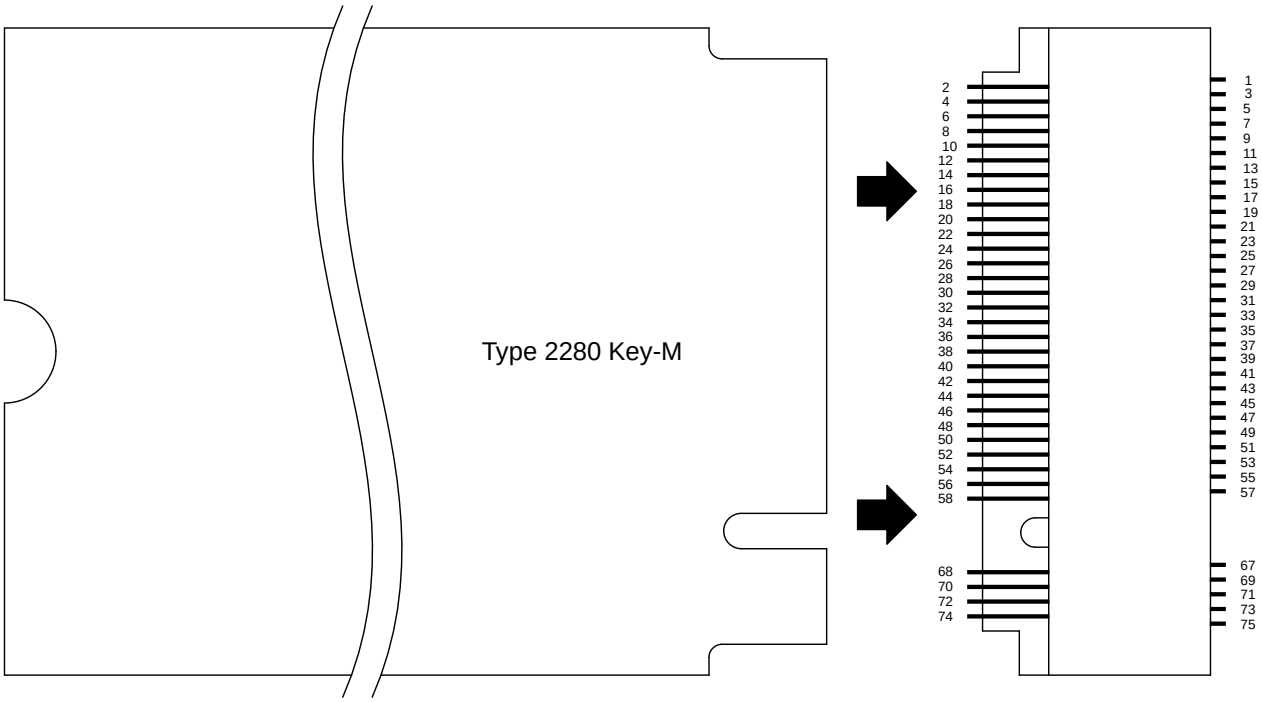
DSTREAM-XT supports an optional M.2 adapter which can be used in place of an 80mm (type 2280-M) *SSD* card using PCIe lane widths of x1, x2 or x4 at up to 16Gbps (Gen 4). For further details about the specification of this connector, please refer to the *PCI Express M.2 Specification*.

Note

The M.2 adapter is not supplied with DSTREAM-XT but is available on request, with any purchase of DSTREAM-XT.

To use an M.2 connector with DSTREAM-XT, the M.2 adapter must be used in conjunction with the 98-way ribbon cable.

Figure 3-11: M.2 connector pinout



M.2 pinout table

Table 3-14: M.2 pinout table

Pin	Signal name		Pin	Signal name
2	**3.3 V**		1	**GND**
4	**3.3 V**		3	**GND**
6	NC		5	**PERn3**
8	NC		7	**PERp3**
10	**DAS/DSS**		9	**GND**
12	**3.3 V**		11	**PETn3**
14	**3.3 V**		13	**PETp3**
16	**3.3 V**		15	**GND**
18	**3.3 V**		17	**PERn2**
20	NC		19	**PERp2**
22	NC		21	**GND**

Pin	Signal name		Pin	Signal name
24	NC		23	**PETn2**
26	NC		25	**PETp2**
28	NC		27	**GND**
30	NC		29	**PERn1**
32	NC		31	**PERp1**
34	NC		33	**GND**
36	NC		35	**PETn1**
38	**DEVSLP**		37	**PETp1**
40	**SMB_CLK**		39	**GND**
42	**SMB_DATA**		41	**PERn0**
44	**ALERT#**		43	**PERp0**
46	NC		45	**GND**
48	NC		47	**PETn0**
50	**PERST#**		49	**PETp0**
52	**CLKREQ#**		51	**GND**
54	**PEWAKE#**		53	**REFCLK-**
56	NC		55	**REFCLK+**
58	NC		57	**GND**
60	KEY		59	KEY
62	KEY		61	KEY
64	KEY		63	KEY
66	KEY		65	KEY
68	**SUSCLK**		67	NC
70	**3.3 V**		69	**PEDET**
72	**3.3 V**		71	**GND**
74	**3.3 V**		73	**GND**
-	-		75	**GND**


Note

In the M.2 specification the **PET** and **PER** signals are with respect to the local device. These signals are then crossed-over at the connector, e.g. **PET** signals on the motherboard connect to **PER** signals on the card.

Connector-specific signals

These signals are not supported by DSTREAM-XT and are disconnected on the M.2 adapter:

- **DAS/DSS**
- **DEVSLP**
- **ALERT#**
- **SMB_CLK**
- **SMB_DATA**

- **SUSCLK**

SATA-[A/B][+/-]

The **SATA-[A/B][+/-]** (Serial Advanced Technology Attachment) signals are not supported by DSTREAM-XT. You can only use these pins in PCIe mode.

3.13 Auxiliary connector

The Auxiliary (AUX) connector on the front of DSTREAM-ST is used to support external probes for high-speed trace capture.

- This connector is not intended for user I/O. Do not attempt to connect anything other than Arm® DSTREAM-ST compatible probes.
- This connector is not compatible with older *RealView Trace* (*RVT*) probes.

3.14 PCIe connector

The PCIe connector on the XT probe is used to connect to additional DSTREAM-XT adapters which provide support for many of the standard PCIe connectors.

This connector is not intended for user I/O. Do not attempt to connect anything other than Arm® DSTREAM-XT compatible adapters.

3.15 User I/O connector

To set up custom input or output signals to your target, use the user Input/Output (I/O) connector.

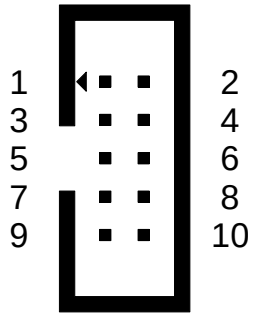
When connecting and disconnecting the user I/O port, Arm recommends that all equipment is powered-down.

- User outputs use the 3.3V LVCMOS standard and have a 100Ω series resistor for short-circuit protection.
- User inputs use the 3.3V LVCMOS standard and have a 10K series resistor and 100K pull-up resistor. You can safely drive the inputs up to a maximum of 5V.

- You can use the 3.3V power output to supply external circuitry up to a maximum current of 150mA. If an over-current condition occurs, this power output shuts down until the debug unit is reset.

The user I/O connector is a 10-way 2.54mm pitch box header on the rear of DSTREAM-ST.

Figure 3-12: User I/O connector pinout



User I/O pinout table

Table 3-15: User I/O pinout table

Pin	Signal name		Pin	Signal name
1	**Output 1**		2	**Output 2**
3	**Output 3**		4	**Output 4**
5	**Output 5**		6	**Input 1**
7	**Output 6**		8	**Input 2**
9	**3.3V (output)**		10	**GND**

4. Target board design

When you design a target board to connect to any Arm debug system, you must consider the rules that are discussed in this chapter.

4.1 Overview of high-speed design

When designing a target board that will be connected to an Arm debug system, it is important to use good digital design practice to achieve high *Signal Integrity* (SI).

A target system might work perfectly when it is connected to an older or slower debug unit, but it could fail to work with newer debug system because of the shorter rise/fall-times.

While many target boards already take SI into consideration for trace signals, it is also important to use the same design methodology for the debug signals. To achieve the high-data throughput that is required to debug modern target systems, DSTREAM-ST units are designed to drive their JTAG interfaces at up to 180MHz. To drive at this frequency, DSTREAM-ST units use fast output drivers with short rise-times.

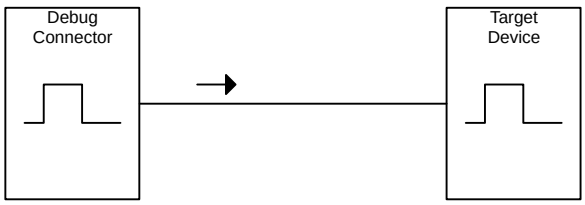
There are many design rules that you can implement to ensure high SI in the debug and trace interfaces.

While these rules apply to all digital signals, Arm recommends giving special attention to the clock signals, **TCK**, **RTCK**, and **TRACECLK**.

Avoid stubs

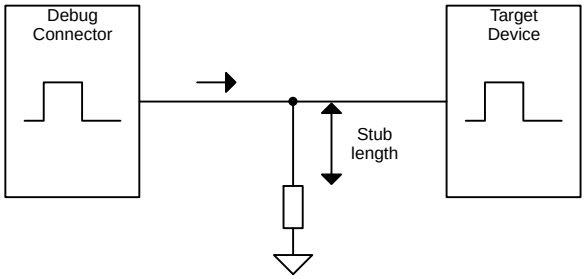
Where possible, debug and trace signals should be point-to-point between the driver and receiver of the signal with no T-junctions or branches leading to other circuitry on the target board.

Figure 4-1: Point-to-point signal



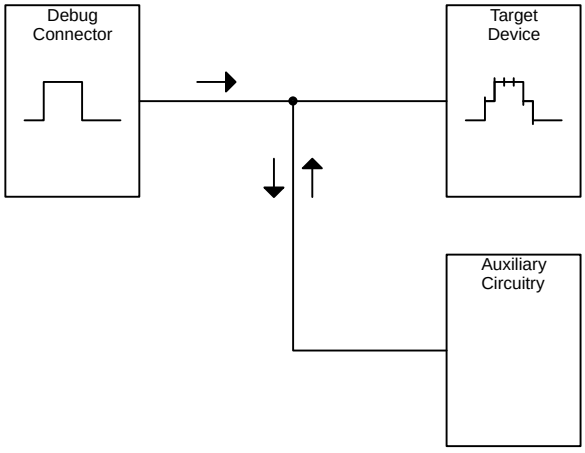
For debug signals, pull-up or pull-down resistors are often required. Pull-up or pull-down resistors might create a branch or stub in the signal path. It is important to keep the stub length in the signal path as short as possible.

Figure 4-2: Stub length



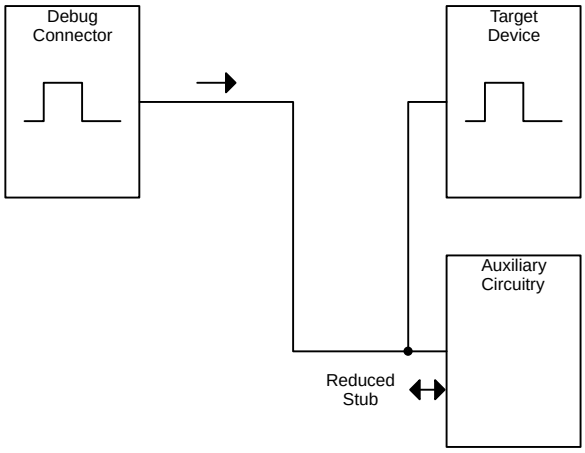
If a signal is routed with a long stub, the signal from the driver is split two ways when it reaches the T-junction. The signal that reaches the target device initially has a lower amplitude until the other half of the signal has reflected back from the end of the stub. The reflection has the effect of creating a stepped signal at the target device. A stepped signal at the target device can cause extra false signal edges to be received.

Figure 4-3: Long stub causing false edges



The simplest method to avoid a long stub causing false edges is to shorten the stub length by rerouting the signal. While rerouting the signal might add length to the signal route, the reduction in stub length is much more favorable.

Figure 4-4: Improved route with shorter stub



Alternative methods include:

- To prevent signal reflections affecting the incoming signal, use a buffer at T-junctions. This method is used to replicate clock signals.
- To route a signal without stubs, use an analog switch. This method is used when a device pin has multiple functions, for example, JTAG and general I/O.
- To deflect a larger portion of the signal away from the stub, use a resistor at the junction of the stub. This method is used when the stub leads to lower-bandwidth circuitry.

Ensure the continuity of return signals

As a digital signal propagates along its route, an inverse signal travels through the adjacent plane because of the electric-field coupling between the signal and the plane. When the signal rise/fall times are short, the return signal follows the path of least inductance, rather than resistance. This means that the return signal flows through a path in the adjacent plane, that is as close as possible to the signal route. When the return path is interrupted, it causes distortion and some loss in the signal.

To minimize return path issues:

- Ensure that the return path that is adjacent to the signal is continuous with no slots or accidental voids that are caused by *anti-pads*.
- When routing a signal from one layer to another, link the planes close to the signal via using a return via. If the planes are at different voltages, use a low-value capacitor to AC-couple the return path.
- When routing signals to and from a cable connector, ensure that all of the return signals of the cable are used. Directly link the return signals or AC-couple them to ground, as necessary.

Minimize crosstalk

Every signal route on a target board has some effect on nearby signal tracks because of the coupling of electric and magnetic fields between the tracks. The electric and magnetic field coupling causes small variations in the surrounding signals which, over long enough distances, can cause data corruption.

There are several ways to minimize electric and magnetic field coupling:

- Space the signal tracks further apart. Arm recommends to keep adjacent signals at least three times further apart than they are from the nearest plane (the 3W rule).
- Bring the plane closer to the signals. To reduce the 3W distance that is needed between adjacent signals, use thinner laminates between the signal and plane layers.
- Keep the signal tracks as short as possible. To cut down on routing while also reducing crosstalk, place a debug or trace connector closer to the target device.

Use impedance matching

Every signal route on a target board has an effective impedance that is measured in Ohms. Effective impedance is the equivalent resistance to ground a signal experiences when it initially enters a signal route, before any reflection from the far end has occurred. It is important to note:

- If the different portions of a signal route have different impedances, it can cause reflections in the route. Reflections reduce the integrity of the signal.
- DSTREAM-ST is designed to work with target boards that use 50Ω signal tracks.

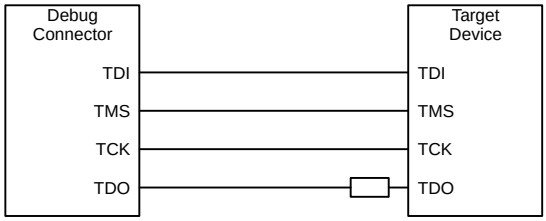
Most modern PCB design tools include functionality for calculating track impedance. There are also various free resources online for calculating the impedance of the various types of PCB track.

4.2 JTAG port buffering

JTAG buffering is sometimes required on the target board to improve signal integrity and increase the usable bandwidth of the interface. You can implement JTAG buffering using common, off-the-shelf parts at little cost.

Usually, the JTAG connector of a target system connects to a single device, for example:

Figure 4-5: JTAG connection without buffers



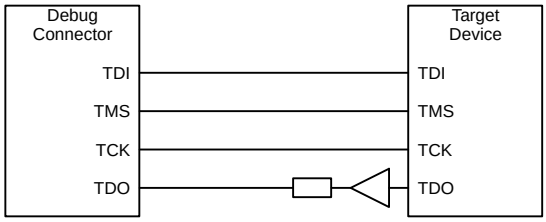
It is highly recommended to place a series termination resistor on any signals being output from the target device (in this case **TDO**) to ensure high signal integrity.

Note

- Pull-up and pull-down resistors are omitted for clarity.
- For more information about using and selecting series termination resistors, see Series termination.

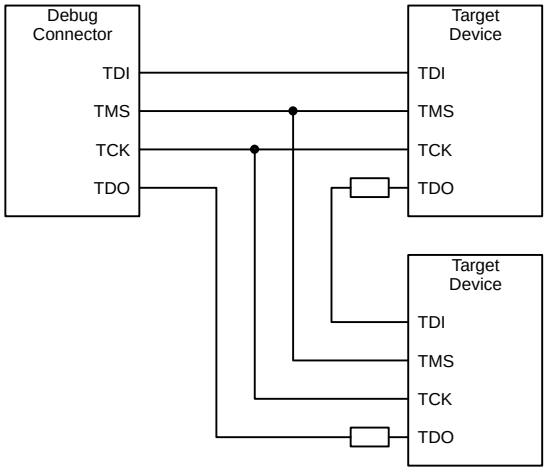
However, if the **TDO** output of the target device has a weak drive-strength (<4mA), the **TDO** output could significantly limit the maximum frequency of the JTAG interface. To prevent this, place a buffer close to the **TDO** pin of the target device with the appropriate series termination resistor:

Figure 4-6: JTAG connection with TDO buffer



Sometimes, two or more devices are chained together in the target system:

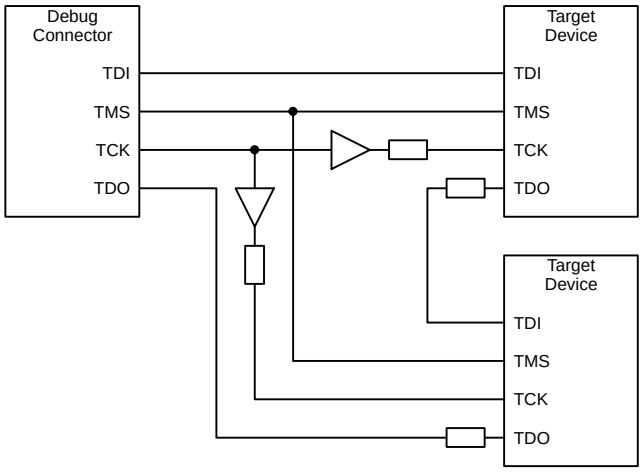
Figure 4-7: Daisy-chained JTAG connection without buffers



Achieving high signal integrity becomes difficult in this scenario because the **TMS** and **TCK** signals are branched at T-junctions. The signal integrity of the **TMS** signal is not very critical since it is only sampled by the target devices when the rising edge of **TCK** signal is detected, by which point **TMS** should be stable. The signal integrity of the **TCK** signal is critical because any unintended or *false* edges cause the target device to sample **TDI** and **TMS** signals too many times which corrupts the serial data stream.

To avoid this issue, always use buffering where the **TCK** signal is split:

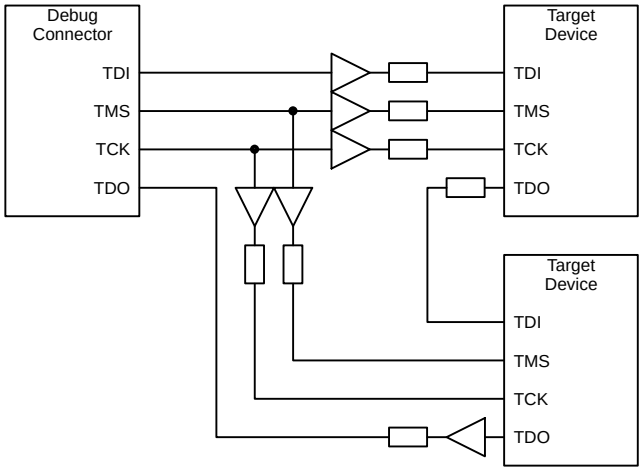
Figure 4-8: Daisy-chained JTAG connection with TCK buffers



The solution in the above figure prevents the two **TCK** branches from interacting and ensures high signal integrity with minimal overshoot. If using this method, you must place the buffers and series termination resistors as close as possible to the T-junction of the **TCK** signal.

While the above solution prevents the majority of signal integrity issues, it might reduce the maximum usable JTAG frequency by introducing skew between the **TCK** and **TDI**/**TMS** signals. If needed, you can correct this skew by using the same type of buffers on the **TDI**, **TMS**, and **TCK** signals. For example:

Figure 4-9: Fully buffered JTAG connection



This solution matches the skew between the **TCK** and **TDI**/**TMS** signals to achieve high JTAG frequencies. Again, place the buffers and series termination resistors as close as possible to the T-junctions of the **TMS** and **TCK** signals.


Note

- For added noise rejection, you can use Schmitt buffers instead of standard buffers.
- To enable the highest JTAG frequencies, Arm recommends you use buffers with a drive strength of 24mA or above.
- For any signals which require pull-up or pull-down resistors, place these resistors on the input-side of the buffers. This ensures the correct logic levels when the debug system is not connected.

4.3 Series termination

Series termination, or *source termination*, is a technique that is used in point-to-point signaling, to ensure that no excessive overshoot or ringing occurs.

To achieve series termination, a series resistor is used to reduce the source voltage by approximately 50% as it is transmitted by the driver. When the signal reaches the end of the transmission line, the high impedance of the receiver causes a reflection that reverts the signal to its original amplitude. When the reflection returns to the series terminating resistor, the potential across the resistor drops to zero, preventing any further current from entering the transmission line. The receiver effectively observes a perfect 100% logic transition, without any overshoot or ringing.

To ensure that reliable signals are delivered to the debug unit, Arm recommends that all outputs from the target system are simulated, and, if necessary, series terminated. Some overshoot or undershoot is acceptable, but we recommend ensuring this is kept less than ~0.5V. Above ~0.5V, the clamping diodes at the receivers start to cause high transient currents, which then cause increased crosstalk, radio emissions, and target power usage.

The target signal impedance for Arm debug systems is 50Ω.

When the outputs cannot be simulated, typical series terminating resistor values are:

Table 4-1: Typical series terminating resistor values

Driver strength	Typical series terminator	Notes
32mA	39Ω	Best signal integrity, highest speed
24mA	33Ω	-
16mA	27Ω	-
12mA	22Ω	-
8mA	15Ω	-
6mA	10Ω	Worst signal integrity, lowest speed

Some types of IC use *impedance matched* outputs to improve their signal integrity. Impedance matched outputs are commonly achieved by using weaker drive transistors to slow down the edge transitions. This can have the unintended side-effect of limiting the maximum data throughput of the driver.


Note

To achieve the highest data rates with the best signal integrity, Arm recommends using:

- A fast and strong driver.
- An appropriate series terminating resistor.

When series terminating multiple signals, it is common to use small quad resistor packages. Small quad resistor packages save board space, and reduce the parasitic effects with reduced risk of placement or *tombstoning* issues during production.


Note

If you determine that series terminating resistors are unlikely to be required, Arm strongly recommends that 0Ω links are placed close to the driver in case you need to add resistors later.

4.4 Parallel trace modeling

For trace bit rates of up to 600Mbps, basic signal integrity can be established using simplified modeling. Most of the transmission line model consists of the cable that is used to connect the debug unit to the target.

- The 30cm CoreSight™ cable is made using 0.635mm pitch ribbon, and can be modeled as a 66Ω transmission line, with a 1.5ns propagation delay, and 0.4Ω DC resistance. The connectors at either end of the cable can be modeled as a 0.5pF capacitance to ground.
- The 15cm CoreSight cable is made using 0.635mm pitch ribbon, and can be modeled as a 66Ω transmission line, with a 0.75ns propagation delay, and 0.2Ω DC resistance. The connectors at either end of the cable can be modeled as a 0.5pF capacitance to ground.
- The JTAG 20 cable is made using 1.27mm pitch ribbon, and can be modeled as a 100Ω transmission line, with a 1.5ns propagation delay, and 0.1Ω DC resistance. The connectors at either end can be modeled as a 1.0pF capacitance to ground.
- The MIPI-60 cable is made using 0.5mm pitch micro-coaxial ribbon, and can be modeled as a 50Ω transmission line, with a 1.5ns propagation delay, and 0.1Ω DC resistance. The connectors at either end can be modeled as a 0.25pF capacitance to ground.

The circuit at the debug unit end of the transmission line can be modeled using the following primitives:

- All resistors can be modeled as their ideal resistance values with minimum or zero parasitics.
- All capacitors can be modeled as their ideal capacitance values with minimum or zero parasitics.
- Input comparators can be modeled using a Xilinx Spartan 6 SSTLx_I model. The switching threshold can be assumed to be half of the VTREF voltage, as supplied by the target. The data is valid when it is 100mV above or below this threshold.
- Output drivers can be modeled using a Xilinx Spartan 6 LVCMOS Fast 16mA model. You must choose the model voltage to match the target system voltage.

All other parasitics and traces within the DSTREAM-ST are negligible for most purposes.

- To achieve good signal integrity, Arm recommends using series termination resistors on all target outputs.
- While we do not currently provide simulation models for Arm debug units, it is possible to provide cut-down layout and BOM files which expose the signal routing and component parasitics of individual signals. If you require these files for simulation purposes, please supply the following information to your support representative:
  - The model of Arm debug unit being used.
  - The signal names of interest.
  - The preferred format of the layout files (Gerber or ODB++).

4.5 Target design checklist

To ensure your target design is compatible with Arm debug systems, your answer to each applicable question in this checklist must be 'Yes'.

Not all questions are applicable to every target system.

Check that your design meets the following requirements, where applicable:

- Are any **TDI**, **TMS**, **TDO**, or **SWDIO** signals pulled *HIGH*?
- Are any **TCK**, **RTCK**, or **SWCLK** signals pulled *LOW*?
- Are any **nTRST** or **nSRST** signals pulled to their inactive state (usually *HIGH*)?
- To pass data between the **TCK** domain and the internal clock domain, does the target device contain the necessary synchronization logic?
- If used, does **RTCK** have its own driver (separate from **TCK**)?
- If **TCK** is routed to multiple devices, have you used buffers to fan-out the signal (to prevent signal reflections)?
- Can the debug unit drive **nTRST** and **nSRST** separately?
- To allow debug from reset, can you reset the target device without initializing its debug logic?
- If using Serial Wire Debug, is the **TMS**/**SWDIO** signal bidirectional (no uni-directional buffers)?
- To reduce the need to calibrate during setup, are any **TRACEDATA** and **TRACECLK** signals length-matched within a 10mm window?
- Where possible, have you eliminated stubs and other parasitic effects from debug and trace signals (especially HSSTP or PCIe signals)?

- Are all debug and single-ended trace signals impedance-matched to 50Ω?
- Are all HSSTP differential trace signals or REFCLK signals impedance-matched to 100Ω?
- Are all PCIe differential data signals impedance-matched to 85Ω?
- Are all single-ended outputs from the target device series terminated and impedance-matched to 50Ω?
- Have the appropriate **VTREF** signal (or signals) been connected to the debug or trace connector (or connectors)?
- Either directly or through a resistor of 100Ω or less, are **VTREF** pins connected to the debug/trace logic rail (or rails)?
- Are the debug/trace logic rails in the range of 1.2V to 3.3V?
- Are all **GND** pins of the debug/trace connector (or connectors) either directly connected, or AC-coupled to GND, close to the connector?
- If using a Mictor socket, are the central **GND** pins solder-pasted on the component side of the board?
- If using an HSSTP 40-way socket, are the locking **GND** pins solder-pasted on the component side of the board?
- If using dual Mictor sockets, are the connectors positioned with the correct spacing, orientation, and alignment?
- If using a standard 2.54mm or 1.27mm header, is the connector fully shrouded to avoid mis-connection (space permitting)?
- If using a CoreSight™ 10/20 or MIPI 34 connector, has pin-7 been removed to maintain proper compatibility?
- To ensure the continuity of return paths, do any signal vias have **GND** return vias placed close to them?
- Have you checked the board layout to ensure that no signals cross slots or voids in the adjacent plane (or planes)?
- Where possible, has crosstalk between individual debug or trace signals been minimized?