Arm® DSTREAM-HT

Version 1.0

System and Interface Design Reference Guide

arm

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
101761_0100_01_en

Arm® DSTREAM-HT

System and Interface Design Reference Guide

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.

Release Information

Document History

Issue	Date	Confidentiality	Change
0100-00	15 July 2019	Non-Confidential	New document for v1.0
0100-01	13 December 2019	Non-Confidential	Documentation update for version 1.0 release

Non-Confidential Proprietary Notice

This document is protected by copyright and other related rights and the practice or implementation of the information contained in this document may be protected by one or more patents or pending patent applications. No part of this document may be reproduced in any form by any means without the express prior written permission of Arm. **No license, express or implied, by estoppel or otherwise to any intellectual property rights is granted by this document unless specifically stated.**

Your access to the information in this document is conditional upon your acceptance that you will not use or permit others to use the information for the purposes of determining whether implementations infringe any third party patents.

THIS DOCUMENT IS PROVIDED "AS IS". ARM PROVIDES NO REPRESENTATIONS AND NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE DOCUMENT. For the avoidance of doubt, Arm makes no representation with respect to, and has undertaken no analysis to identify or understand the scope and content of, third party patents, copyrights, trade secrets, or other rights.

This document may include technical inaccuracies or typographical errors.

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL ARM BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF ARM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

This document consists solely of commercial items. You shall be responsible for ensuring that any use, duplication or disclosure of this document complies fully with any relevant export laws and regulations to assure that this document or any portion thereof is not exported, directly or indirectly, in violation of such export laws. Use of the word "partner" in reference to Arm's customers is not intended to create or refer to any partnership relationship with any other company. Arm may make changes to this document at any time and without notice.

If any of the provisions contained in these terms conflict with any of the provisions of any click through or signed written agreement covering this document with Arm, then the click through or signed written agreement prevails over and supersedes the conflicting provisions of these terms. This document may be translated into other languages for convenience, and you agree that if there is any conflict between the English version of this document and any translation, the terms of the English version of the Agreement shall prevail.

The Arm corporate logo and words marked with ® or ™ are registered trademarks or trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. All rights reserved. Other brands and names mentioned in this document may be the trademarks of their respective owners. Please follow Arm's trademark usage guidelines at *http://www.arm.com/company/policies/trademarks*.

Copyright © 2019 Arm Limited (or its affiliates). All rights reserved.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

LES-PRE-20349

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.

Web Address

www.arm.com

Conformance Notices

This section contains conformance notices.

Federal Communications Commission Notice

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

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 Conformity

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 distributer where it was purchased. The distributer 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.

The CE Declaration of Conformity for this product is available on request.

The system should be powered down when not in use.

It is recommended that ESD precautions be taken when handling this product.

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

- Ensure attached cables do not lie across any sensitive equipment.
- Reorient the receiving antenna.
- Increase the distance between the equipment and the receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

Note

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

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

Contents

Arm® DSTREAM-HT System and Interface Design Reference Guide

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

List of Figures
Arm® DSTREAM-HT System and Interface Design Reference Guide

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

List of Tables
Arm® DSTREAM-HT System and Interface Design Reference Guide

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

Preface

This preface introduces the *Arm® DSTREAM-HT System and Interface Design Reference Guide*.

It contains the following:

- *About this book* on page 10.

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

About this book

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

Using this book

This book is organized into the following chapters:

Chapter 1 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.

Chapter 2 Target interface connectors

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

Chapter 3 Target board design

When you design a target board to connect to the DSTREAM-HT system, you must consider the rules that are discussed in this chapter.

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.

Typographic conventions

italic
: Introduces special terminology, denotes cross-references, and citations.

bold
: Highlights interface elements, such as menu names. Denotes signal names. Also used for terms in descriptive lists, where appropriate.

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

`monospace`
: Denotes a permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.

`monospace italic`
: Denotes arguments to monospace text where the argument is to be replaced by a specific value.

`monospace bold`
: Denotes language keywords when used outside example code.

`<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>
```


Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

SMALL CAPITALS
: Used in body text for a few terms that have specific technical meanings, that are defined in the *Arm® Glossary*. For example, IMPLEMENTATION DEFINED, IMPLEMENTATION SPECIFIC, UNKNOWN, and UNPREDICTABLE.

Feedback

Feedback on this product

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

- The product name.
- The product revision or version.
- An explanation with as much information as you can provide. Include symptoms and diagnostic procedures if appropriate.

Feedback on content

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

- The title *Arm DSTREAM-HT System and Interface Design Reference Guide*.
- The number 101761_0100_01_en.
- If applicable, the page number(s) to which your comments refer.
- A concise explanation of your comments.

Arm also welcomes general suggestions for additions and improvements.

Note

Arm tests the PDF only in Adobe Acrobat and Acrobat Reader, and cannot guarantee the quality of the represented document when used with any other PDF reader.

Other information

- *Arm® Developer*.
- *Arm® Information Center*.
- *Arm® Technical Support Knowledge Articles*.
- *Technical Support*.
- *Arm® Glossary*.

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

Chapter 1
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.

Note

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`.

It contains the following sections:

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

1.1 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.

Test Data In (TDI)

The **TDI** 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.

Test Mode Select (TMS)

The **TMS** 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.

Test Clock (TCK)

The **TCK** 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`.

Test Data Out (TDO)

The **TDO** 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`.

Basic 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:

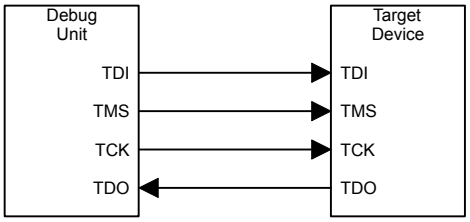

Figure 1-1 Simple JTAG connection

Note

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

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

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

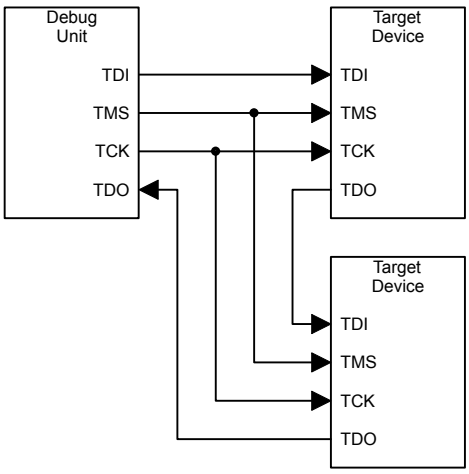

Figure 1-2 Chained JTAG connection

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

——— **Warning** ———

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.

For more information, see *JTAG port buffering* on page 3-54.

———————————

JTAG timing characteristics

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

TDI and **TMS** are set up by the DSTREAM-HT 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-HT system on the next rising edge of **TCK**.

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

Basic JTAG timing:

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

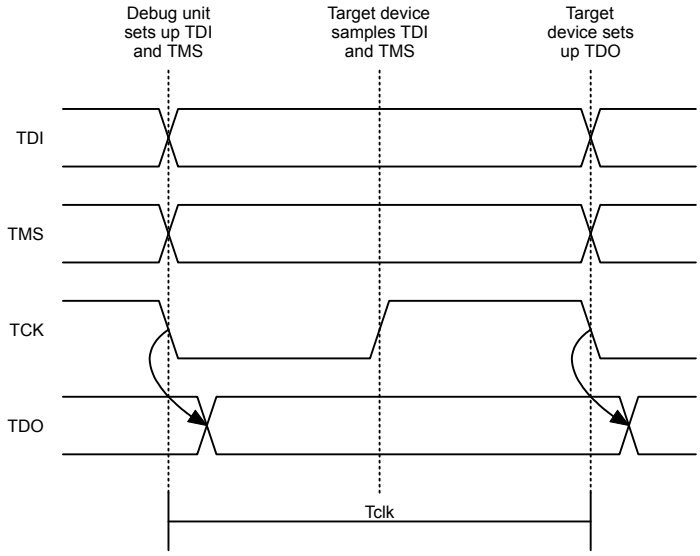

Figure 1-3 JTAG timing diagram

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-HT 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-HT 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 1-1 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.com*.

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.

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

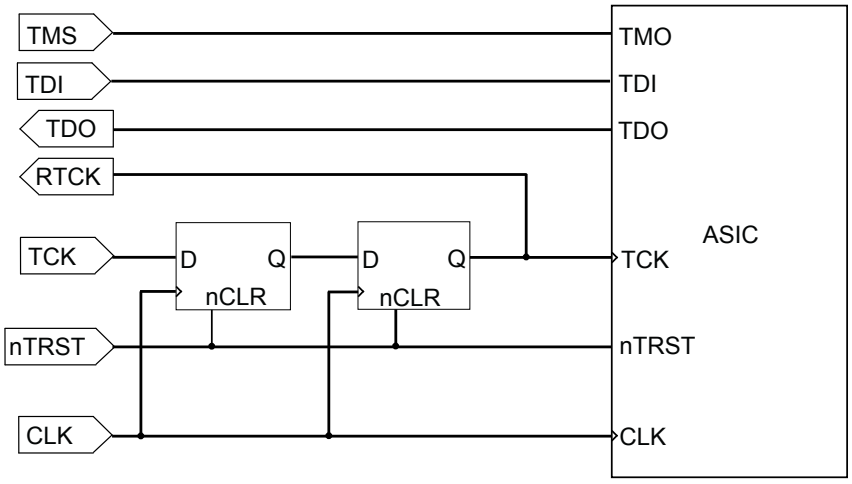

Figure 1-4 Basic JTAG port synchronizer

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.

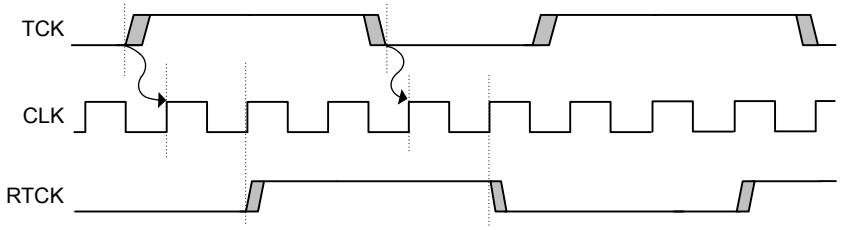

Figure 1-5 Timing diagram for the Basic JTAG synchronizer

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:

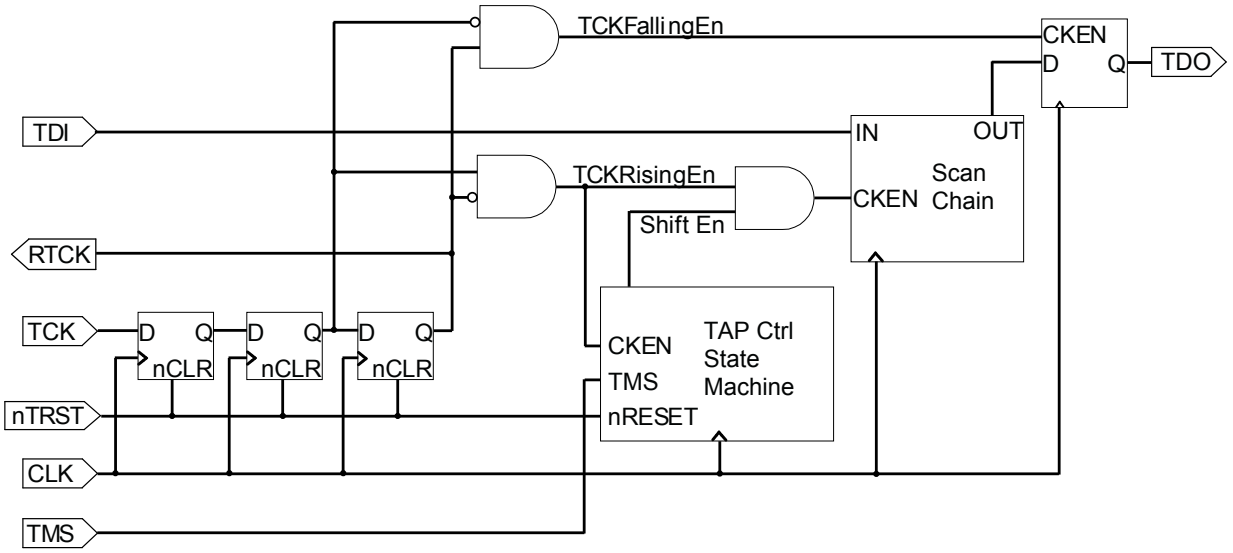

Figure 1-6 JTAG port synchronizer for single rising-edge D-type ASIC design rules

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

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**.

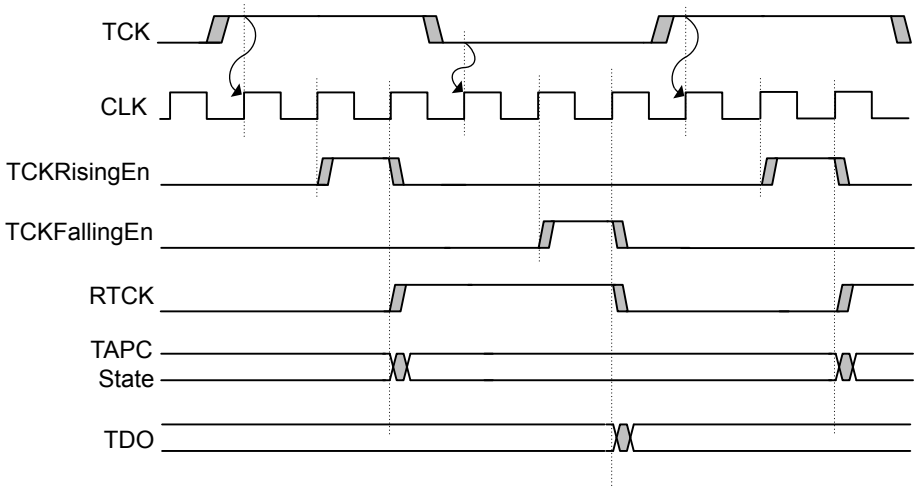

Figure 1-7 Timing diagram for the D-type JTAG synchronizer

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

1.2 Return Clock (RTCK) 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.

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

If used, the **RTCK** signal must be pulled `LOW` on the target.

——— **Warning** ———

RTCK must never be directly linked 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.

———————————

Adaptive clocking

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.

——— **Note** ———

- 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.
- Adaptive clocking can be enabled using the configuration settings in Arm Development Studio. For more information, see *Debug Hardware configuration* in the Arm Development Studio User Guide.
- If adaptive clocking is used, the debug unit cannot detect the clock speed, and therefore cannot scale its internal timeouts. If the target 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.

———————————

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

1.3 Reset signals

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

System Reset (nSRST)

The 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

When no debug unit is connected, unintended resets can occur. To avoid unintended resets, the **nSRST** signal must be pulled to its inactive logic level on the target.

By default, the **nSRST** signal has a logic level of active `LOW`. To avoid unintended resets, pull the **nSRST** signal `HIGH`.

The polarity of the **nSRST** signal is configurable in Arm Development Studio.

TAP Reset (nTRST)

The TAP reset signal initializes the Test Access Port, debug logic, and boundary scan cells in the target device.

When no debug unit is connected, unintended resets can occur. To avoid unintended resets, the **nTRST** signal must be pulled to its inactive logic level on the target.

By default, the nTRST signal has a logic level of active `LOW`. To avoid unintended resets, pull the nTRST signal `HIGH`.

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

Note

Arm strongly recommends that the **nSRST** and **nTRST** signals are 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:

- Depending on your target, it might not be possible to debug a system from reset because any breakpoints previously set might be lost.
- You might need to restart the debug session because the JTAG interface might not recover when the TAP controller state is changed.

It is expected that the assertion of the **nSRST** line by the DSTREAM-HT 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-HT 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:

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

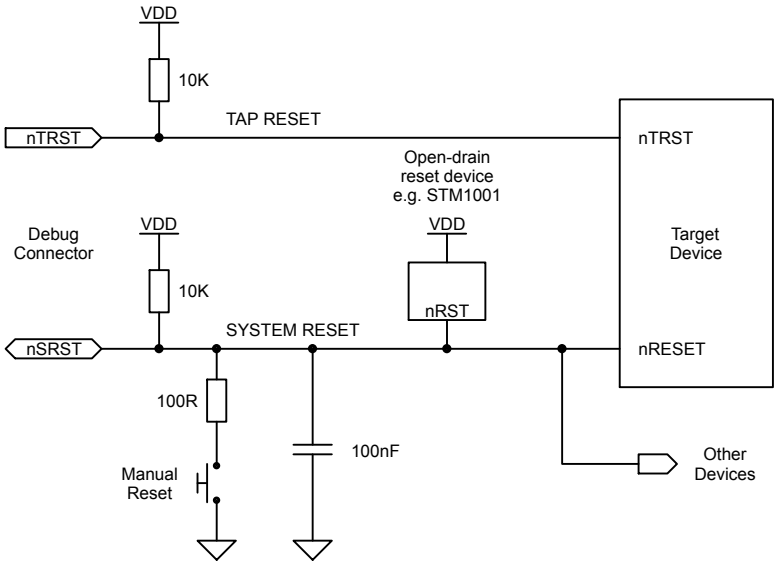

Figure 1-8 Example reset circuit

The push-button, 100R 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.

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

1.4 Run-Control signals

The run-control signals are now deprecated within Arm Development Studio, however, low-speed control might still be possible through ConfigItems and the command-line utilities.

Debug Request (DBGRQ)

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

Arm recommends that this signal is no longer used. You can leave the signal open on the target board.

——— **Warning** ———

If the signal is used, it must be pulled `LOW` on the target.

———————————

Debug Acknowledge (DBGACK)

The Debug Acknowledge (**DBGACK**) 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 this signal is no longer used. You can leave the signal open on the target board.

——— **Warning** ———

If the signal is used, it must be pulled `LOW` on the target.

———————————

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

1.5 Serial Wire Debug (SWD) 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.

Serial Wire Data I/O (SWDIO)

During debugging, the Serial Wire Data I/O (**SWDIO**) signal is bidirectional. It can send and receive serial data from the target.

The **SWDIO** signal must be pulled `HIGH` on the target to keep the signal inactive when no debug unit is connected.

Warning

SWDIO signal is bidirectional and the functionality is shared with a unidirectional JTAG **TMS** signal. Ensure that there are no buffers on the target which would prevent bidirectional communication.

Serial Wire Clock (SWCLK)

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

The **SWCLK** signal must be pulled `LOW` on the target to keep the signal inactive when no debug unit is connected.

Serial Wire Output (SWO)

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

The **SWO** signal must be pulled `HIGH` on the target to keep the signal inactive when no debug unit is connected.

SWD timing requirements

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

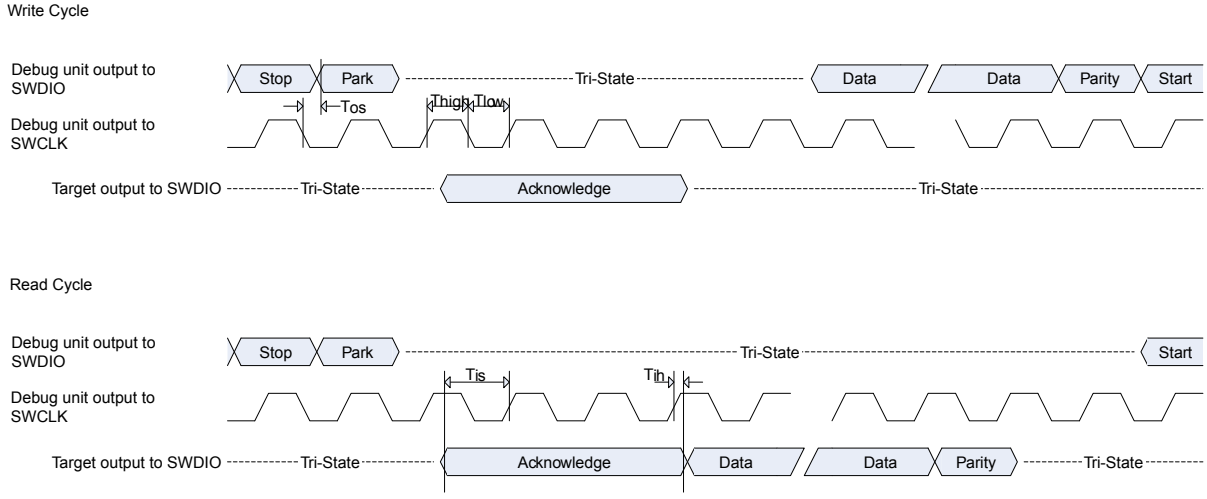

Figure 1-9 SWD timing diagrams

The debug unit:

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

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

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 1-2 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**.
T[ih]	1ns	-	Input hold time that is required between **SWDIO** and rising edge **SWCLK**.

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

1.6 Trace signals

Some target devices can output high-bandwidth 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, DSTREAM-ST 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.

The HSSTP probe, in the DSTREAM-HT system, extends this functionality by supporting up to six lanes of HSSTP trace.

The trace signals supported by the DSTREAM-HT system are:

TX[0-5][+/-]

The TX signals are differential outputs from the target, which are used to collect HSSTP trace data.

CLK[+/-]

The CLK signal is a differential output from the DSTREAM-HT system. You can use this signal as a very low-jitter clock source to drive the HSSTP output stage of the target device.

You can configure the frequency of this clock signal to provide a wide range of frequencies.

TRACEDATA[0-3]

The Trace Data signals are single-ended outputs from the target and can be used to collect 1-bit to 4-bit trace data.

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

TRACECLK

The 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.

The 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 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.

The following figure and table describe the timing for **TRACECLK**:

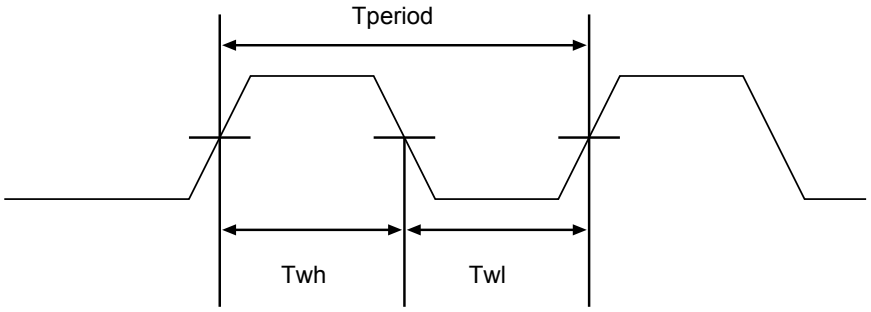

Figure 1-10 TRACECLK timing diagram

Table 1-3 TRACECLK characteristics

Parameter	Min	Max	Description
Tperiod (min)	1.667ns	125ms	Clock period
Twh (min)	833ps	62.5ms	High pulse width
Twl (min)	833ps	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 you connect an unpowered DSTREAM-ST unit to a powered target, on any of the debug or trace signals, there is a maximum leakage current into the DSTREAM-ST unit of ±10µA.

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

1.7 Target Voltage Reference (VTREF) signals

The Target Voltage Reference, or **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 one or more 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 (Vi(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, **VTREF** signals must be above 800mV. **VTREF** signals above 800mV illuminate the **VTREF** LED on the HSSTP probe.
- 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:

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

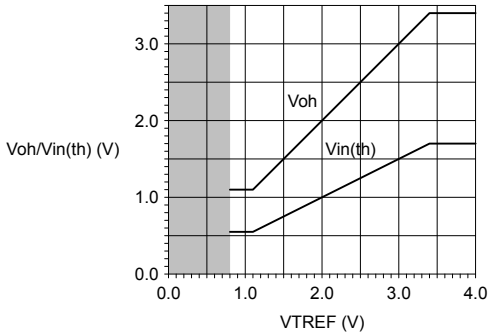

Figure 1-11 Target interface logic levels

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

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

1.8 I/O diagrams for DSTREAM-HT signals

The following diagrams and descriptions illustrate 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.

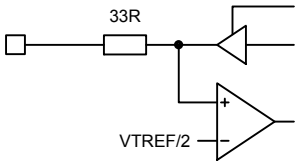

Figure 1-12 Input/Output signals

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.

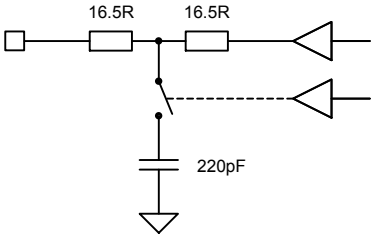

Figure 1-13 TCK signal

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.

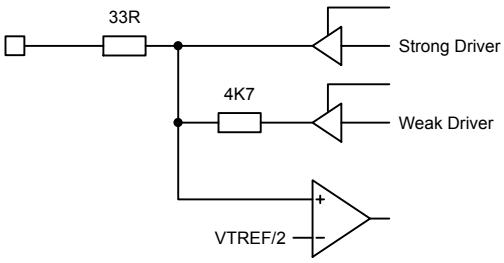

Figure 1-14 Reset signals

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

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.

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

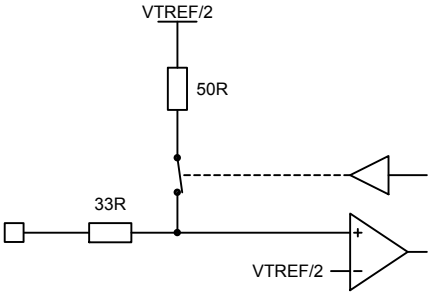

Figure 1-15 Parallel trace signals

HSSTP trace signals

The HSSTP trace signals (**Tx[0-5]**) are differential inputs which, by default, are terminated to a 900mV source through 50Ω resistors.

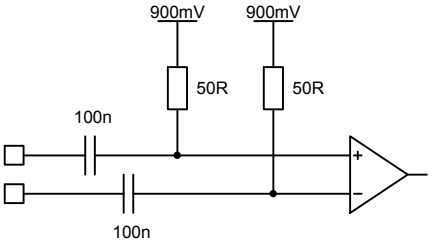

Figure 1-16 HSSTP trace signals

VTREF signals

The **VTREF** signals (**VTREF**, **DEBUG_VTREF** and **TRACE_VTREF**) are buffered to 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.

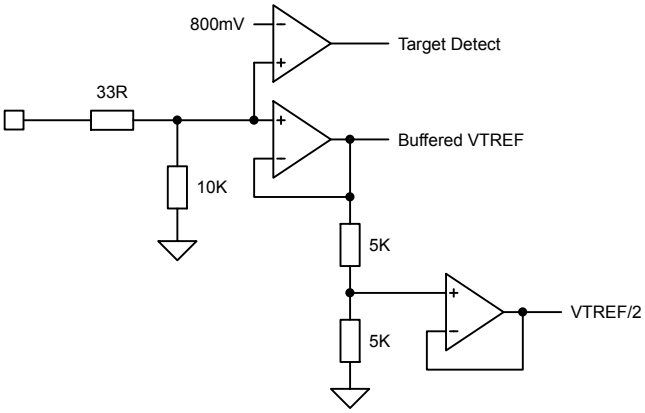

Figure 1-17 VTREF signals

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

1.9 Typical SWD circuit

A typical SWD circuit:

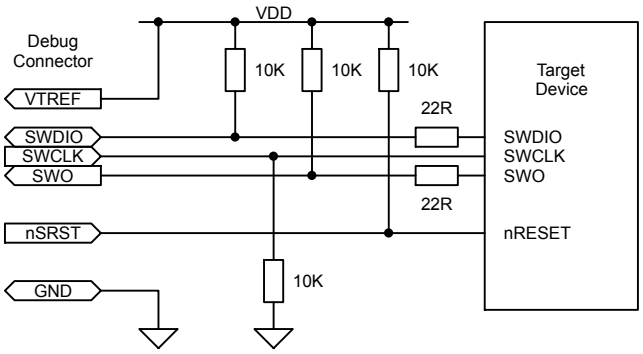

Figure 1-18 Typical SWD circuit

Note

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

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

1.10 Typical JTAG circuit

A typical JTAG circuit:

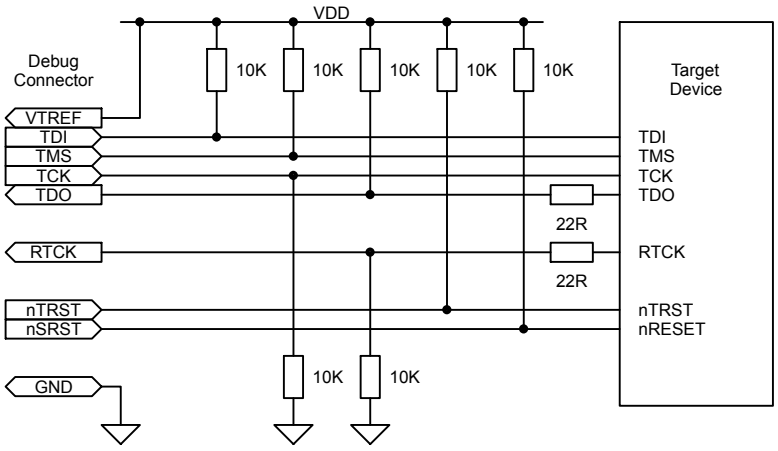

Figure 1-19 Typical JTAG circuit

Note

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

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

Chapter 2
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 HSSTP probe provides an additional HSSTP 40-way connector.

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-HT Getting Started Guide*.

If your target system uses a connector which is not currently supported, and you are considering a volume order of Arm debug units, *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.

It contains the following sections:

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

- *2.11 User I/O connector* on page 2-49.

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

2.1 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 2-1 Connector attributes

Connector	Arm JTAG 20	CoreSight 10	CoreSight 20	TI JTAG 14	MICTOR 38	MIPI 34	MIPI 60	HSSTP 40
JTAG supported	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
SWD supported	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
SWO trace supported	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Parallel trace supported	No	No	Yes	No	Yes	Yes	Yes	No
Max parallel trace width (1)	N/A	N/A	4	N/A	16	4	32	N/A
Separate debug/trace voltage domains	No	No	No	No	Yes	Yes	Yes	N/A
Requires adapter	No	No	No	Yes	Yes	Yes	Yes	No
Cable signal integrity	Medium	Good	Good	Poor	Excellent	Good	Excellent	Excellent
MIPI pinout compatible	No	Yes	Yes	No	No	Yes	Yes	No
Target connector cost	Low	Low	Low	Low	High	Low	Medium	Medium
Connector durability (2)	High	Low	Low	High	Medium	Low	Medium	Medium
Approximate footprint area (mm^2) (3)	297	65	95	250	221	140	170	170
Through-hole/SMD	Either	Either	Either	Either	SMD (4)	Either	SMD	SMD (5)
Ease of assembly (placement/soldering)	High	High	High	High	Low	High	Medium	Medium

Note

1. The trace width supported by the connector. DSTREAM-ST supports up to 4-bit wide parallel trace.
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. Mictor 38 connector is not recommended for future designs.
5. The HSSTP 40-way connector has locking pins which must be solder-pasted from the component side.

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

2.2 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.

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

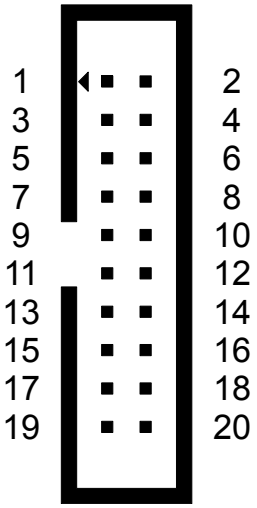

Figure 2-1 Arm JTAG 20 connector pinout

Arm JTAG 20 pinout table

Table 2-2 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**

——— Warning ———

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.

———————————

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

2.3 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.

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

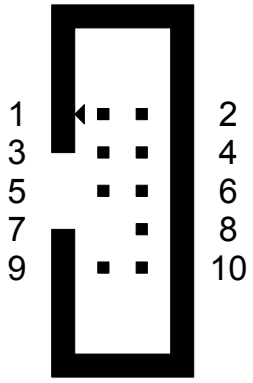

Figure 2-2 CoreSight 10 connector pinout

CoreSight™ 10 pinout table

Table 2-3 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**)	8	**TDI**
9	**GND**	10	**nSRST**

Note

Pin 7 must be removed for compatibility with DSTREAM-ST and MIPI specifications.

Warning

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.

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

2.4 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.

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

Warning

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

If you do not configure the CoreSight 20 connector correctly, your DSTREAM-HT system might not operate correctly. For example, if the pinout was configured for debug and trace, instead of debug only: the **nTRST** signal might terminate as if it was a **TRACEDATA** signal, which causes the **nTRST** signal to be asserted, and might cause a TAP reset when the DSTREAM-HT system is connected.

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 either:

- `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.

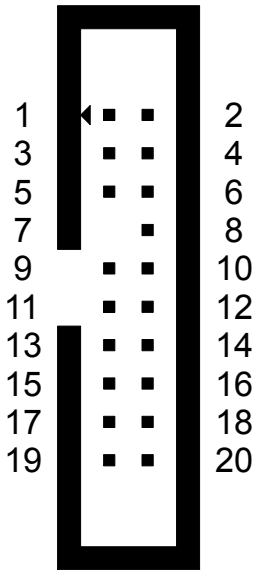

Figure 2-3 CoreSight 20 connector pinout

CoreSight™ 20 pinout tables

Pinout when `DSTREAMCS20` is set to `0`:

Table 2-4 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**

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

Table 2-4 Arm CoreSight 20 pinout table (DSTREAMCS20=0) (continued)

Pin	Signal name	Pin	Signal name
7	Key (**NC**)	8	**TDI**
9	**GND**	10	**nSRST**
11	**GND** (1)	12	**TRACECLK**
13	**GND** (1)	14	**TRACEDATA[0]**
15	**GND**	16	**TRACEDATA[1]**
17	**GND**	18	**TRACEDATA[2]**
19	**GND**	20	**TRACEDATA[3]**

Pinout when `DSTREAMCS20` is set to `1`:

Table 2-5 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**)	8	**TDI**
9	**GND**	10	**nSRST**
11	**GND** (1)	12	**RTCK**
13	**GND** (1)	14	**SWO**
15	**GND**	16	**nTRST**
17	**GND**	18	**DBGRQ**
19	**GND**	20	**DBGACK**

1. Although these pins 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.

Note

Pin 7 must be removed for compatibility with DSTREAM-ST and MIPI specifications.

Warning

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.

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

2.5 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.

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

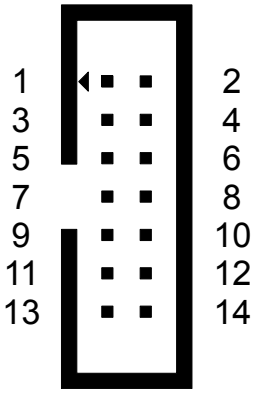

Figure 2-4 TI JTAG 14 connector pinout

TI JTAG 14 pinout table

Table 2-6 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**

Warning

- 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.
- For the pin-out of the TI JTAG 14 connector, do not use 14-way IDC cables to connect the target board and debug unit. To avoid cross-talk issues between **nTRST**, **TMS**, and **TDI**, the DSTREAM-ST TI JTAG 14 adapter must connect directly to the target board.

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

2.6 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.

——— Warning ———

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 a 2-767004-2 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.

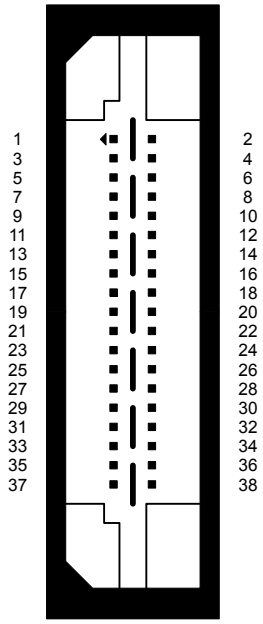

Figure 2-5 Mictor 38 connector pinout

Mictor 38 pinout table

Table 2-7 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** (1)
11	**TDO**	12	**TRACE_VTREF** (2)
13	**RTCK**	14	**DEBUG_VTREF** (2)
15	**TCK**	16	**TRACEDATA[7]**
17	**TMS**	18	**TRACEDATA[6]**

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

Table 2-7 Mictor 38 pinout table (continued)

Pin	Signal name	Pin	Signal name
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]**
29	**TRACEDATA[12]**	30	**Logic 0** (3)
31	**TRACEDATA[11]**	32	**Logic 0** (3)
33	**TRACEDATA[10]**	34	**Logic 1** (3)
35	**TRACEDATA[9]**	36	**TRACECTL** (4)
37	**TRACEDATA[8]**	38	**TRACEDATA[0]**

Note

1. The **EXTTRIG** signal is deprecated and not supported by Arm Development Studio.
2. 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.
3. These signals are not used by DSTREAM-ST. To maintain compatibility with other debug units, connect the signals to the appropriate power rails.
4. The **TRACECTL** signal is not currently supported by DSTREAM-ST because only continuous-mode TPIU trace is supported.

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

2.7 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.

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.

Note

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

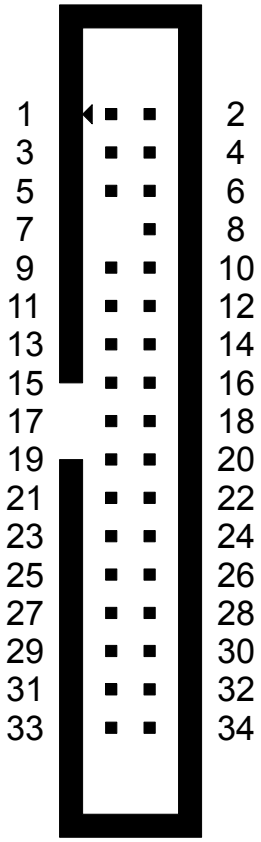

Figure 2-6 MIPI 34 connector pinout

MIPI 34 pinout table

Table 2-8 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**)	8	**TDI**
9	**GND**	10	**nSRST**
11	**GND** (1)	12	**RTCK**

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

Table 2-8 MIPI 34 pinout table (continued)

Pin	Signal name	Pin	Signal name
13	**GND** (1)	14	**TRST_PD** (2)
15	**GND**	16	**nTRST**
17	**GND**	18	**DBGRQ**
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** (3)
33	**GND**	34	**TRACE_VTREF**

Note

1. Although these pins 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.
2. 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.
3. The **TRACEEXT** signal is not supported by DSTREAM-ST.

Note

Pin 7 must be removed for compatibility with DSTREAM-ST and MIPI specifications.

Warning

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.

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

2.8 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 QTH-030-01-L-D-A from Samtec.

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.

Note

- The MIPI 60 adapter is not supplied with the DSTREAM-HT system, but is available on request.
- 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.

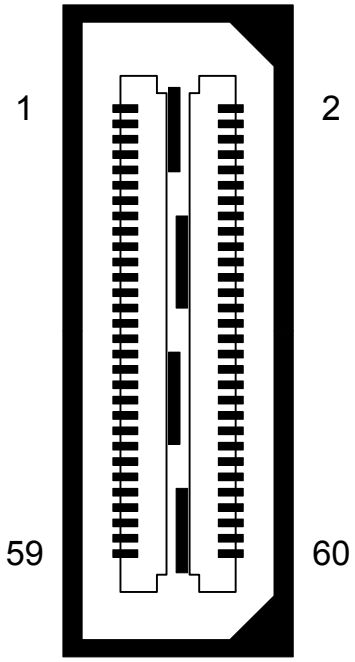

Figure 2-7 MIPI 60 connector pinout

MIPI 60 pinout table

Table 2-9 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** (1)
9	**nTRST**	10	**DBGRQ**
11	**DBGACK**	12	**TRACE_VTREF**
13	**TRACECLK[0]**	14	RESERVED
15	**GND**	16	**GND**

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

Table 2-9 MIPI 60 pinout table (continued)

Pin	Signal name	Pin	Signal name
17	**TRACECTL**	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]**
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
45	**TRACEDATA[13]**	46	RESERVED
47	**TRACEDATA[14]**	48	RESERVED
49	**TRACEDATA[15]**	50	RESERVED
51	**TRACEDATA[16]**	52	RESERVED
53	**TRACEDATA[17]**	54	RESERVED
55	**TRACEDATA[18]**	56	RESERVED
57	**GND**	58	**GND**
59	RESERVED	60	RESERVED

Note

1. 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.

Note

- DSTREAM-ST only supports one channel of parallel trace.
- Pins marked as RESERVED might be internally connected in DSTREAM-ST, but are not currently supported.

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

2.9 HSSTP 40 connector

The HSSTP 40 connector is a 40-way 0.8mm pitch socket which supports JTAG debug, Serial Wire Debug, SWO trace, and up to six lanes of HSSTP trace.

——— **Warning** ———

The locking ground pins of the HSSTP socket must be solder-pasted on the component side of the PCB. If these pins are solder-pasted or hand-soldered from the opposite side, it might cause mechanical stability issues.

———————————

Typically, the socket used is a ERF8-020-05.0-S-DV-L from Samtec. In high Electrostatic Discharge (ESD) environments, Arm recommends using the Samtec ASP-130367-01 socket, which is identical except that the HSSTP pins are slightly shorter. This provides further ESD protection to sensitive HSSTP transceivers.

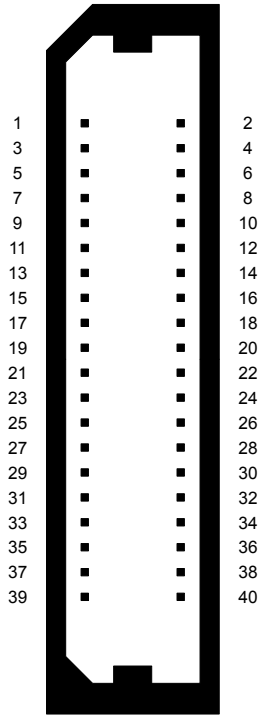

Figure 2-8 HSSTP 40 connector pinout

HSSTP 40 pinout table

Table 2-10 HSSTP 40 pinout table

Pin	Signal name	Pin	Signal name
1	**TX4+**	2	**VTREF**
3	**TX4-**	4	**TCK**
5	**GND**	6	**GND**
7	**TX2+**	8	**TMS**
9	**TX2-**	10	**nTRST**

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

Table 2-10 HSSTP 40 pinout table (continued)

Pin	Signal name	Pin	Signal name
11	**GND**	12	**GND**
13	**TX0+**	14	**TDI**
15	**TX0-**	16	**TDO**
17	**GND**	18	**GND**
19	**CLK+**	20	**nSRST**
21	**CLK-**	22	**DBGRQ** (1)
23	**GND**	24	**GND**
25	**TX1+**	26	**DBGACK** (1)
27	**TX1-**	28	**RTCK**
29	**GND**	30	**GND**
31	**TX3+**	32	**TRIGIN** (1)
33	**TX3-**	34	**TRIGOUT** (1)
35	**GND**	36	RESERVED
37	**TX5+**	38	RESERVED
39	**TX5-**	40	RESERVED

Note

1. The **DBGRQ**, **DBGACK**, **TRIGIN** and **TRIGOUT** signals are deprecated and are not supported by Arm Development Studio.

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

2.10 Auxiliary (AUX) connector

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

——— **Warning** ———

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.

———————————

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

2.11 User I/O connector

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

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

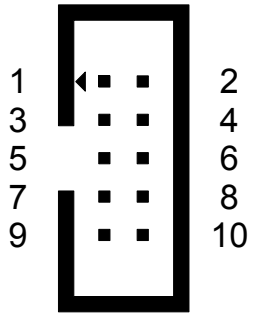

Figure 2-9 User I/O connector pinout

Note

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

User I/O pinout table

Table 2-11 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**

Note

- User outputs use the 3.3V LVCMOS standard and have a 100R 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. The inputs can be safely driven up to a maximum of 5V.
- The 3.3V power output can be used 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.

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

Chapter 3
Target board design

When you design a target board to connect to the DSTREAM-HT system, you must consider the rules that are discussed in this chapter.

It contains the following sections:

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

3.1 Overview of high-speed design

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

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.

——— **Note** ———

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

———————————

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

——— **Note** ———

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.

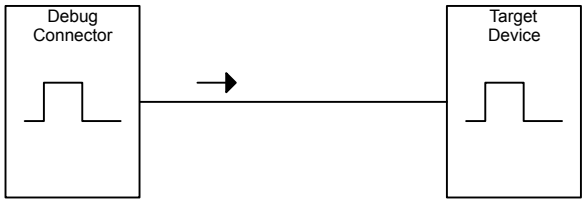

Figure 3-1 Point-to-point signal

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.

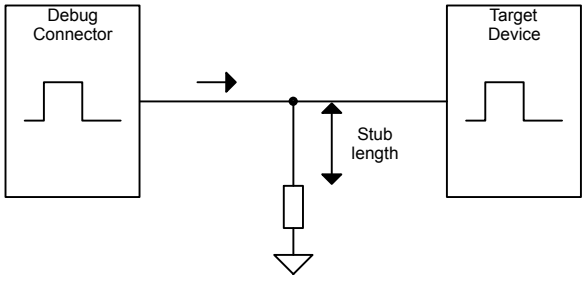

Figure 3-2 Stub length

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.

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

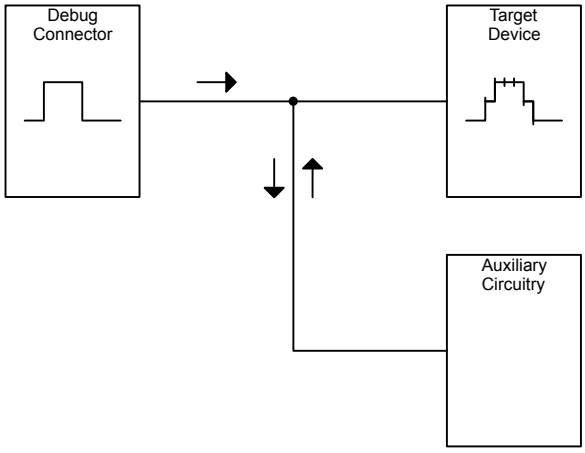

Figure 3-3 Long stub causing false edges

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.

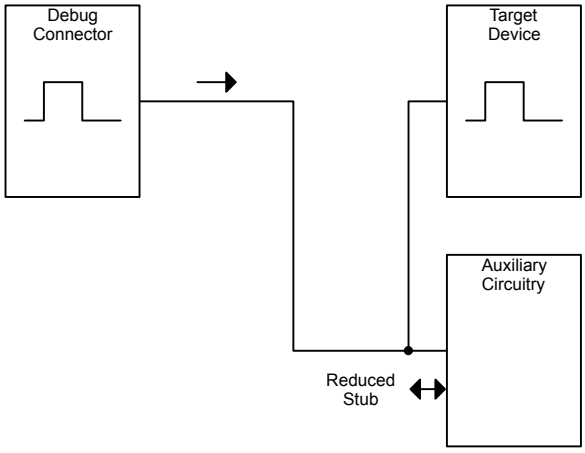

Figure 3-4 Improved route with shorter stub

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 edges are short, the return signal usually 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.

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

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.

  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.

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

3.2 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:

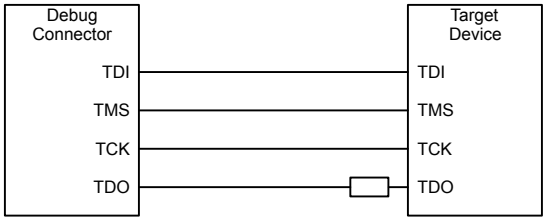

Figure 3-5 JTAG connection without buffers

Note

Pull-up and pull-down resistors are omitted for clarity.

To act as a series terminator, you must place a resistor close to the **TDO** pin of target device. Placing a resistor close to the **TDO** pin is the simplest option, and achieves good signal integrity because each signal is point-to-point.

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:

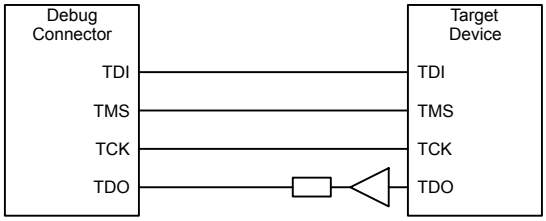

Figure 3-6 JTAG connection with TDO buffer

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

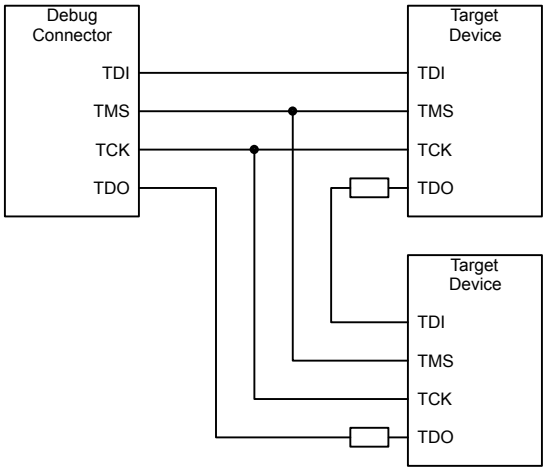

Figure 3-7 Daisy-chained JTAG connection without buffers

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

Achieving good 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 important because until a rising edge of **TCK** signal is detected, it is ignored by the target device. The signal integrity of the **TCK** signal is important because any false edges cause the target device to sample **TDI** and **TMS** signals too many times. Sampling the **TDI** and **TMS** signals too many times corrupts the serial data stream that is seen by the target devices.

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

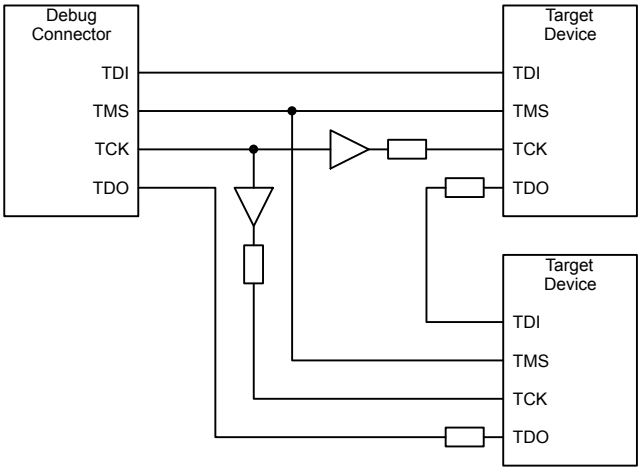

Figure 3-8 Daisy-chained JTAG connection with TCK buffers

The solution in the above figure prevents the two **TCK** branches from interacting and ensures good signal integrity with minimal overshoot. You must place buffers and series termination resistors as close as possible to the T-junction of the **TCK** signal.

This causes some skew between the **TDI**, **TMS**, and **TCK** signals. To correct this skew, use the same type of buffers on the **TDI**, **TMS**, and **TCK** signals. For example:

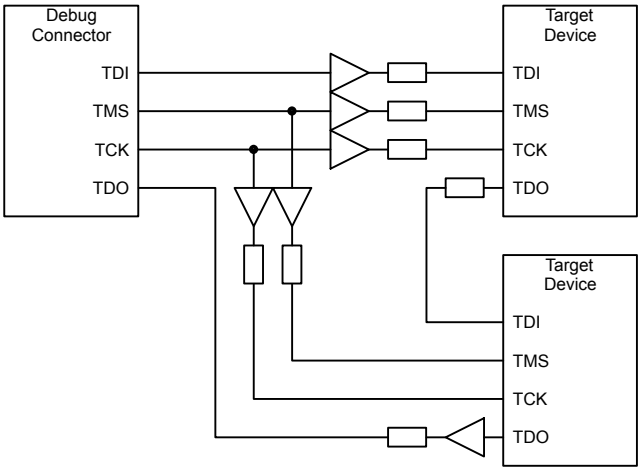

Figure 3-9 Fully buffered JTAG connection

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

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

Note

- For added noise rejection, Schmitt buffers can be used instead of standard buffers.
- Arm recommends you use buffers with a drive strength of 24mA or above.
- For any buffered signal, place the signal pull-up or pull-down resistor at the input-side of the buffer.

For guidance on selecting series termination resistors, see *Series termination* on page 3-57.

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

3.3 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, use a series resistor 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 observes a perfect 100% logic transition, without any overshoot or ringing.

To ensure that a reliable signal is delivered to the DSTREAM-ST unit, Arm recommends that all outputs from the target system are simulated, and, if necessary, series terminated. Some overshoot or undershoot is acceptable, but Arm recommends 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 DSTREAM-ST is 50Ω.

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

Table 3-1 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. Using weaker drive transistors limits the data throughput of the driver.

To achieve the highest data rates with the best signal integrity, Arm recommends using:
- A fast and strong driver.
- An appropriate series terminating resistor.

If you determine that series terminating resistors are not required, as a backup option, Arm recommends that 0Ω links are placed close to the driver.

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 without much risk of placement or tombstoning issues during production.

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

3.4 Parallel trace modeling

For trace bit rates of 0-600Mbps, 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 DSTREAM-ST 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 DSTREAM-ST 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 the Spartan 3 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 the Spartan 3 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.

Note

To achieve good signal integrity, Arm recommends using series termination resistors on all target outputs.

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

3.5 Target design checklist

To ensure your target design is compatible with the DSTREAM-ST unit or DSTREAM-HT system, your answer to each applicable question in this checklist must be ‘Yes’.

——— **Note** ———

Not all questions are applicable to every target design.

———————————

Table 3-2 Target design checklist

Check item	Status
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 signals)?	
Are all debug and single-ended trace signals impedance-matched to 50Ω?	
Are all HSSTP differential trace signals impedance-matched to 100Ω?	
To obtain 50Ω output impedance, have you routed any outputs from the target device through series termination resistors?	
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 same side of the board?	
If using an HSSTP 40-way socket, are the locking GND pins solder-pasted on the same 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, have you considered the removal of pin-7?	
To ensure the continuity of return paths, do any signal vias have return vias placed close to them?	

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential

Table 3-2 Target design checklist (continued)

Check item	Status
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 debug and trace signals been minimized?	

Copyright © 2019 Arm Limited or its affiliates. All rights reserved.
Non-Confidential