ARM® DSTREAM-ST

Version 1.0

System and Interface Design Reference Guide

ARM

ARM 100893_0100_00_en

ARM® DSTREAM-ST

System and Interface Design Reference Guide

Release Information

Document History

Issue	Date	Confidentiality	Change
0100-00	31 March 2017	Non-Confidential	First release

Non-Confidential Proprietary Notice

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

http://www.arm.com

Conformance Notices

This section contains conformance notices.

Federal Communications Commission Notice

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

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 Declaration of Conformity

CE

The system should be powered down when not in use.

It is recommended that ESD precautions be taken when handling DSTREAM-ST equipment.

The DSTREAM-ST modules generate, use, 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 reception, which can be determined by turning the equipment off or on, you are encouraged to try to correct the interference by one or more of the following measures:

- Ensure attached cables do not lie across the target board.
- 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 ARM Support for help.

Note

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

Contents
ARM® DSTREAM-ST System and Interface Design Reference Guide

List of Figures
ARM® DSTREAM-ST System and Interface Design Reference Guide

List of Tables
ARM® DSTREAM-ST System and Interface Design Reference Guide

Preface

This preface introduces the *ARM® DSTREAM-ST System and Interface Design Reference Guide*.

It contains the following:

- *About this book* on page 10.

About this book

DSTREAM-ST System and Interface Design Reference Guide describes the interfaces of the DSTREAM-ST debug and trace unit, with details about designing ARM® architecture-based ASICs and PCBs. This document is written for those using DSTREAM-ST or those designing PCBs.

Using this book

This book is organized into the following chapters:

Chapter 1 ARM® DSTREAM-ST system design guidelines

The ARM® DSTREAM-ST debug and trace unit enables powerful software debug and optimization on an ARM processor-based hardware target. Use the information in this chapter to design your own ARM-architecture-based devices and Printed Circuit Boards (PCBs) that can be debugged using the DSTREAM-ST unit.

Chapter 2 ARM® DSTREAM-ST target interface connections

This chapter describes the target connector pinouts and their interface signals available on the ARM DSTREAM-ST unit.

Chapter 3 ARM® DSTREAM-ST USER Input/Output (IO) connections

This chapter describes the additional input and output connections provided in the ARM DSTREAM-ST unit.

Chapter 4 Target board design for tracing with ARM® DSTREAM-ST

This chapter describes some considerations for the design of a target board that can be connected to the DSTREAM-ST trace feature.

Chapter 5 Reference

Lists other information that might be useful when working with DSTREAM-ST.

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>
```

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-ST System and Interface Design Reference Guide*.
- The number ARM 100893_0100_00_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*.
- *Support and Maintenance*.
- *ARM Glossary*.

Chapter 1
ARM® DSTREAM-ST system design guidelines

The ARM® DSTREAM-ST debug and trace unit enables powerful software debug and optimization on an ARM processor-based hardware target. Use the information in this chapter to design your own ARM-architecture-based devices and Printed Circuit Boards (PCBs) that can be debugged using the DSTREAM-ST unit.

It contains the following sections:

1.1 Reset signals

All ARM processors have a main processor reset that might be called **nRESET**, **BnRES**, or **HRESET**.

This is asserted by one or more of these conditions:

- Power on reset.
- Manual push-button reset.
- Remote reset from the debugger (using DSTREAM-ST).
- Watchdog circuit reset (if appropriate to the application).

Any ARM processor including the JTAG interface has a second reset input called **nTRST** (TAP Reset). This resets the debug logic, the *Test Access Port* (TAP) controller, and the boundary scan cells. It is activated by remote JTAG reset (from DSTREAM-ST).

Note

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

- It is not possible to debug a system from reset, because any breakpoints previously set are lost.
- You might have to start the debug session from the beginning, because DSTREAM-ST might not recover when the TAP controller state is changed.

DSTREAM-ST reset signals

The DSTREAM-ST unit has two reset signals connected to the debug target hardware, **nTRST** and **nSRST**.

What the signals do:

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

  Note

  It is expected that the assertion of the **nSRST** line by the DSTREAM-ST unit will cause a warm reset of the target system. If the **nSRST** line triggers a power-on reset (POR), then the debug connection might be lost.

The target hardware must pull the reset lines to their inactive state to assure normal operation when the JTAG interface is disconnected. In the DSTREAM-ST unit, the strong pull-up/pull-down resistance is approximately 33Ω, and the weak pull-up/pull-down resistance is approximately 4.7kΩ.

As it is possible to alter the drive strength for **nTRST** and **nSRST**, target assemblies with various different reset configurations can be supported.

Example reset circuits

The diagram shows a typical reset circuit logic for the ARM reset signals and the DSTREAM-ST reset signals.

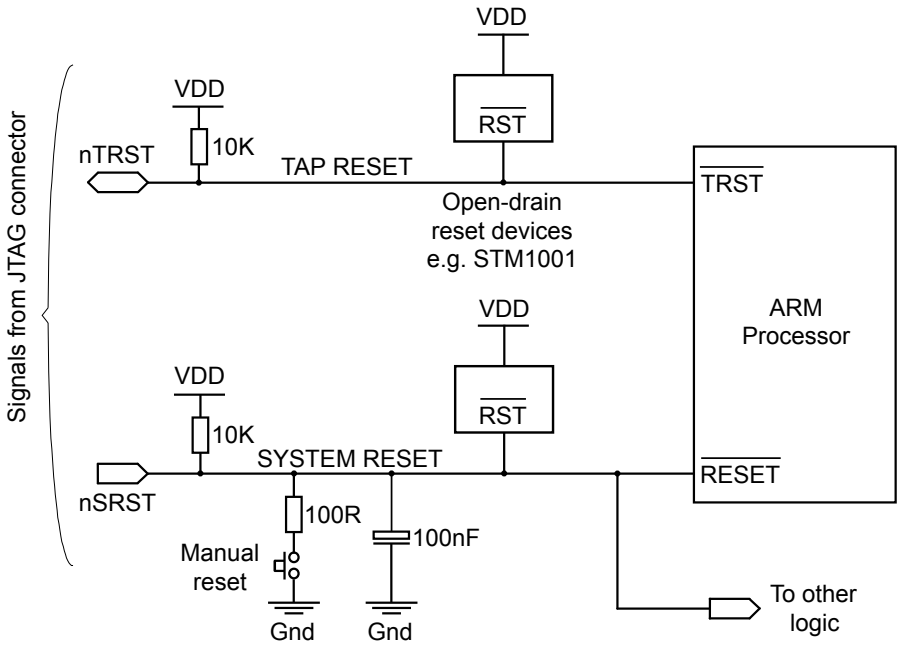

Figure 1-1 Example reset circuit logic

1.2 Working with Application-Specific Integrated Circuits (ASIC) or System-on-Chips (SoC)

Use the information in this section to work with Application-Specific Integrated Circuits (ASIC) or System-on-Chips (SoCs).

This section contains the following subsections:

1.2.1 ASICs containing multiple devices

If your system contains multiple devices that each have a JTAG *Test Access Port* (TAP) controller, you must serially chain them so that DSTREAM-ST can communicate with all of them simultaneously. The chaining can either be within the ASIC, or externally.

Note

There is no support in DSTREAM-ST for multiplexing **TCK**, **TMS**, **TDI**, **TDO**, and **RTCK** between several different processors.

TAP controllers serially chained within the ASIC

The JTAG standard originally described serially chaining multiple devices on a PCB. This concept can be extended to serially chaining multiple TAP controllers within an ASIC, as shown in the following figure:

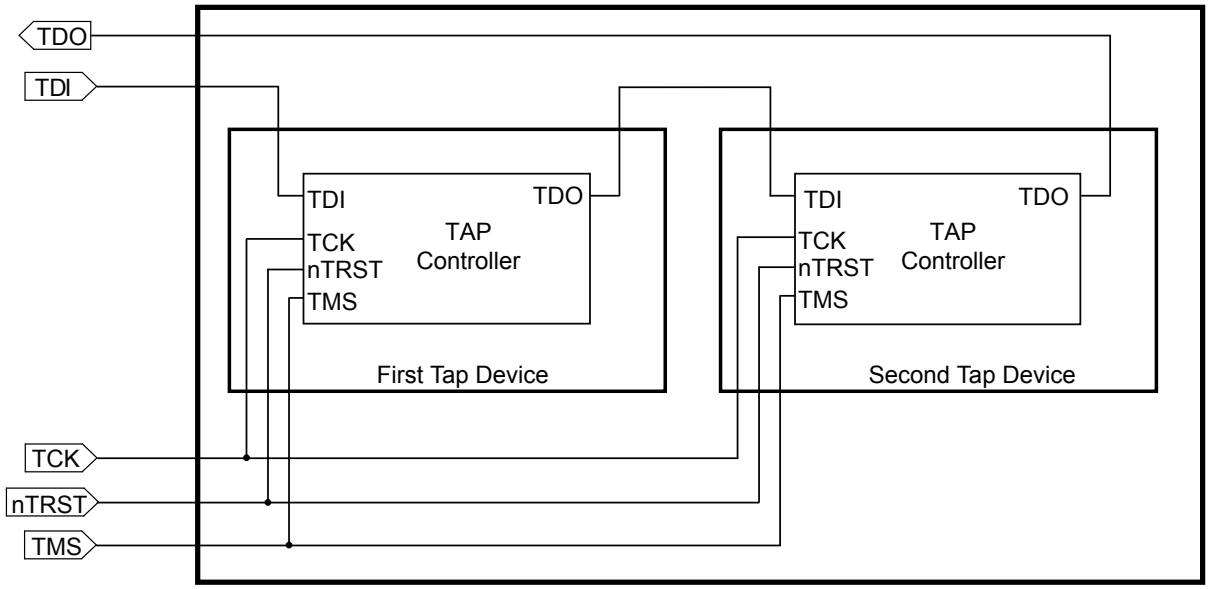

Figure 1-2 TAP Controllers serially chained within an ASIC

This configuration does not increase the package pin count. It does increase JTAG propagation delays, but this impact can be small if unaddressed TAP controllers are placed into bypass mode.

TAP controllers serially chained externally

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

1.2.2 Boundary scan test vectors

If you use the JTAG boundary scan test methodology to apply production test vectors, you might want to have independent external access to each *Test Access Port* (TAP) controller. This avoids the requirement to merge test vectors for more than one block in the device.

One solution to this is to adopt a hybrid approach, using a pin on the package that switches elements of the device into a test mode. You can use this to break the internal daisy chaining of **TDO** and **TDI** signals, and to multiplex out independent JTAG ports on pins that are used for another purpose during normal operation.

1.3 Physical and electrical connection guidelines

Use the guidelines below to define physical and electrical connections on your target board.

JTAG connection scheme

The diagram shows a typical JTAG connection scheme.

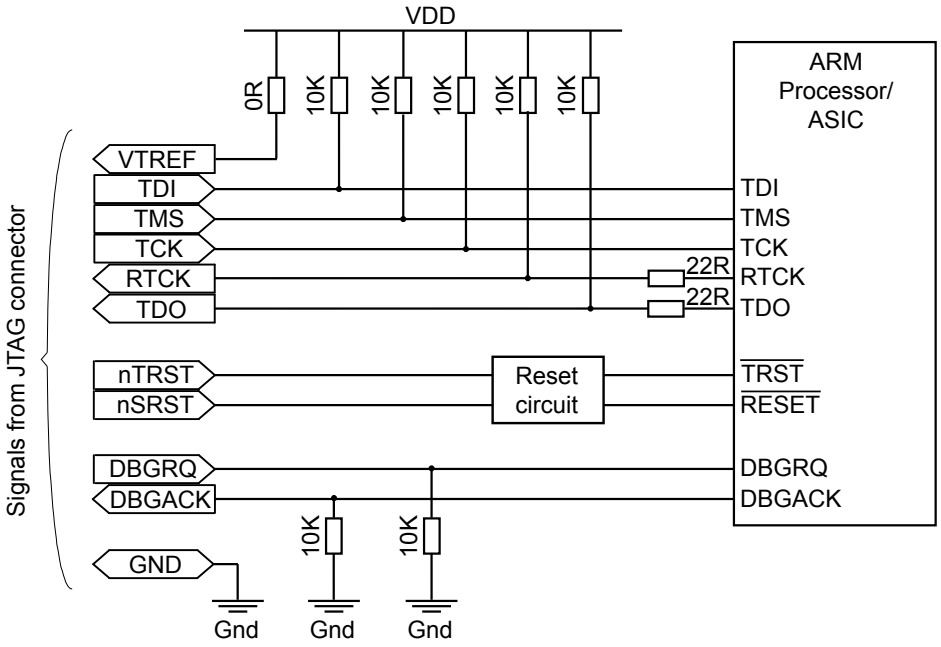

Figure 1-3 Typical JTAG connection scheme

Note

- The signals **TDI**, **TMS**, **TCK**, **RTCK** and **TDO** are typically pulled up on the target board to keep them stable when the debug equipment is not connected.
- **DBGRQ** and **DBGACK** if present, are typically pulled down on the target.
- If there is no **RTCK** signal provided on the processor, it can either be pulled to a fixed logic level or connected to the **TCK** signal to provide a direct loop-back.
- All pull-up and pull-down resistors must be in the range 1K-100KΩ.
- The **VTRef** signal is typically connected directly to the VDD rail. If you use a series resistor to protect against short-circuits, it must have a value no greater than 100Ω.
- 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.

Target interface logic levels

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

VTRef feeds the reference voltage to the DSTREAM-ST unit. This voltage is clipped internally at approximately 3.4V, and is used as the output high voltage (**Voh**) for logic 1s (ones) on **TCK**, **TDI**, and **TMS**.

For logic 0s (zeroes), 0V is used as the output low voltage. The input logic threshold voltage (**Vi(th)**) for the **TDO**, **RTCK**, and **nSRST** input is 50% of the **Voh** level, and so is clipped to approximately 1.7V.

The relationships of **Voh** and **Vi(th)** to **VTRef** are shown in the following figure:

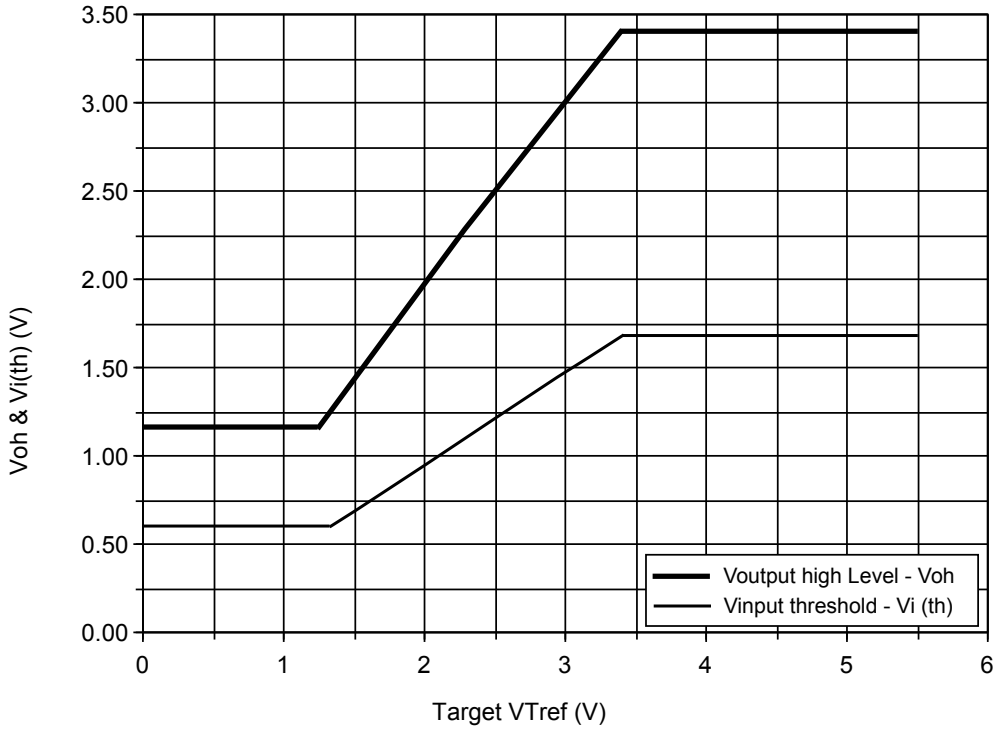

Figure 1-4 Target interface logic levels

DSTREAM-ST can adapt interface levels down to **VTRef** of 1.2V.

By default, the **nTRST** and **nSRST** signals are pulled-up by 4.7K resistors within DSTREAM-ST and driven (strong) low during resets. This allows the reset signals to be driven by other open-drain devices or switches on the target board. The polarity and high/low drive strengths can be configured within the software.

The input and output characteristics of the DSTREAM-ST unit are compatible with logic levels from TTL-compatible, or CMOS logic in target systems. When assessing compatibility with other logic systems, the output impedance of all signals is approximately 50Ω.

Chapter 2
ARM® DSTREAM-ST target interface connections

This chapter describes the target connector pinouts and their interface signals available on the ARM DSTREAM-ST unit.

It contains the following sections:

2.1 About the ARM® JTAG 20 connector pinouts and interface signals

The ARM JTAG 20 connector is a 20-way 2.54mm pitch connector. You can use it in either standard JTAG (IEEE 1149.1) mode or *Serial Wire Debug* (SWD) mode.

The following figure shows the ARM JTAG 20 connector pinout:

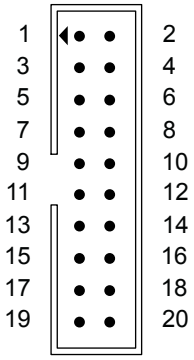

Figure 2-1 ARM JTAG 20 connector pinout

ARM® JTAG 20 pinouts

The table shows the ARM JTAG 20 pinouts as used on the target board.

Table 2-1 ARM JTAG 20 interface pinout table

Pin	Signal name	I/O diagram		Pin	Signal name	I/O diagram
1	**VTRef**	F		2	**NC**	NA
3	**nTRST**	D		4	**GND**	H
5	**TDI**	B		6	**GND**	H
7	**TMS/SWDIO**	B/C		8	**GND**	H
9	**TCK/SWCLK**	B		10	**GND**	H
11	**RTCK**	A		12	**GND**	H
13	**TDO/SWO**	A		14	**GND**	H
15	**nSRST**	E		16	**GND**	H
17	**DBGRQ**	B		18	**GND**	H
19	**DBGACK**	A		20	**GND**	H

ARM® JTAG 20 interface signals

The table describes the signals on the ARM JTAG 20 interface.

Table 2-2 ARM JTAG 20 signals

Signal	I/O	Description
TDI	Output	The Test Data In pin provides serial data to the target during debugging. **TDI** can be pulled HIGH on the target.
TDO	Input	The Test Data Out pin receives serial data from the target during debugging. You are advised to series terminate **TDO** close to the target processor. **TDO** is typically pulled HIGH on the target.

Table 2-2 ARM JTAG 20 signals (continued)

Signal	I/O	Description
TMS	Output	The Test Mode Select pin sets the state of the *Test Access Port* (TAP) controller on the target. **TMS** can be pulled HIGH on the target to keep the TAP controller inactive when not in use.
TCK	Output	The Test Clock pin clocks data into the **TDI** and **TMS** inputs of the target. **TCK** is typically pulled HIGH on the target.
RTCK	Input	The Return Test Clock pin echoes the test clock signal back to DSTREAM-ST for use with adaptive mode clocking. If **RTCK** is generated by the target processor, you are advised to series terminate it. **RTCK** can be pulled HIGH or LOW on the target when not in use.
nTRST	Output	The Test Reset pin resets the *TAP* controller of the processor to allow debugging to take place. **nTRST** is typically pulled HIGH on the target and pulled strong-LOW by DSTREAM-ST to initiate a reset. The polarity and strength of **nTRST** is configurable.
nSRST	Input/ Output	The System Reset pin fully resets the target. This signal can be initiated by DSTREAM-ST or by the target board (which is then detected by DSTREAM-ST). **nSRST** is typically pulled HIGH on the target and pulled strong-LOW to initiate a reset. The polarity and strength of **nSRST** is configurable.
DBGRQ	Output	The Debug Request pin stops the target processor and puts it into debug state. **DBGRQ** is rarely used by current systems and is usually pulled LOW on the target.
DBGACK	Input	The Debug Acknowledge pin notifies DSTREAM-ST that a debug request has been received and the target processor is now in debug state. **DBGACK** is rarely used by current systems and is usually pulled LOW on the target.
SWDIO (SWD mode)	Input/ Output	The Serial Wire Data I/O pin sends and receives serial data to and from the target during debugging. You are advised to series terminate **SWDIO** close to the target processor. —— **Note** —— **SWDIO** signal is bidirectional and the functionality is shared with a unidirectional JTAG TMS line. Ensure that there are no buffers preventing bidirectional communication functionality. ——
SWCLK (SWD mode)	Output	The Serial Wire Clock pin clocks data into and out of the target during debugging.
SWO (SWD mode)	Input	The Serial Wire Output pin provides trace data to DSTREAM-ST. You are advised to series terminate **SWO** close to the target processor.
VTRef	Input	The Voltage Target Reference pin supplies DSTREAM-ST with the debug rail voltage of the target to match its I/O logic levels. **VTRef** can be tied HIGH on the target. —— **Note** —— If **VTRef** is pulled HIGH by a resistor, its value must be no greater than 100Ω. ——
GND	-	Ground.

2.2 About the CoreSight™ 20 connector pinouts and interface signals

You can use the CoreSight™ 20 connector in either standard JTAG (IEEE 1149.1) mode or *Serial Wire Debug* (SWD) mode. It can also optionally capture up to 4 bits of parallel trace in *Trace Port Interface Unit* (TPIU) continuous mode.

When this connector is configured to be a parallel trace source, pins 12 to 20 switch to their alternate trace functions.

The following figure shows the CoreSight 20 connector pinout:

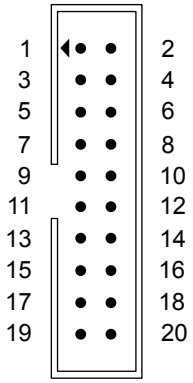

Figure 2-2 CoreSight 20 connector pinout

CoreSight™ 20 pinouts

Table 2-3 CoreSight 20 interface pinout table

Pin	Signal name	I/O diagram		Pin	Signal name	I/O diagram
1	**VTRef**	G		2	**TMS/SWDIO**	B/C
3	**GND**	H		4	**TCK/SWCLK**	B
5	**GND**	H		6	**TDO/SWO**	A
7	**KEY (NC)**	NA		8	**TDI**	B
9	**GND**	H		10	**nSRST**	E
11	**NC**	I		12	**RTCK/TRACECLK**	A
13	**NC**	I		14	**SWO/TraceD0**	E
15	**GND**	H		16	**nTRST/TraceD1**	E
17	**GND**	H		18	**DBGRQ/TraceD2**	A
19	**GND**	H		20	**DBGACK /TraceD3**	A

CoreSight™ 20 interface signals

Table 2-4 CoreSight 20 signals

Signal	I/O	Description
TDI	Output	The Test Data In pin provides serial data to the target during debugging. **TDI** can be pulled HIGH on the target.
TDO	Input	The Test Data Out pin receives serial data from the target during debugging. You are advised to series terminate **TDO** close to the target processor. **TDO** is typically pulled HIGH on the target.
TMS	Output	The Test Mode Select pin sets the state of the *Test Access Port* (TAP) controller on the target. **TMS** can be pulled HIGH on the target to keep the *TAP* controller inactive when not in use.
TCK	Output	The Test Clock pin clocks data into the **TDI** and **TMS** inputs of the target. **TCK** is typically pulled HIGH on the target.
RTCK	Input	The Return Test Clock pin echoes the test clock signal back to DSTREAM-ST for use with adaptive mode clocking. If **RTCK** is generated by the target processor, you are advised to series terminate it. **RTCK** can be pulled HIGH or LOW on the target when not in use.
nTRST	Output	The Test Reset pin resets the *TAP* controller of the processor to allow debugging to take place. **nTRST** is typically pulled HIGH on the target and pulled strong-LOW by DSTREAM-ST to initiate a reset. The polarity and strength of **nTRST** is configurable.
nSRST	Input/Output	The System Reset pin fully resets the target. This signal can be initiated by DSTREAM-ST or by the target board (which might be detected by DSTREAM-ST). **nSRST** is typically pulled HIGH on the target and pulled strong-LOW to initiate a reset. The polarity and strength of **nSRST** is configurable.
DBGRQ	Output	The Debug Request pin stops the target processor and puts it into debug state. **DBGRQ** is rarely used by current systems and is usually pulled LOW on the target.
DBGACK	Input	The Debug Acknowledge pin notifies DSTREAM-ST that a debug request has been received and the target processor is now in debug state. **DBGACK** is rarely used by current systems and is usually pulled LOW on the target.
SWDIO (SWD mode)	Input/Output	The Serial Wire Data I/O pin sends and receives serial data to and from the target during debugging. You are advised to series terminate **SWDIO** close to the target processor. ——— **Note** ——— **SWDIO** signal is bidirectional and the functionality is shared with a unidirectional JTAG TMS line. Ensure that there are no buffers preventing bidirectional communication functionality. ———
SWCLK (SWD mode)	Output	The Serial Wire Clock pin clocks data into and out of the target during debugging.
SWO (SWD mode)	Input	The Serial Wire Output pin provides trace data to DSTREAM-ST. You are advised to series terminate **SWO** close to the target processor. **SWO** is configurable to be captured on pin 6 or 14.
TraceD[0-3]	Input	The Trace Data [0-3] pins provide DSTREAM-ST with *TPIU* continuous mode trace data from the target. You are advised to series terminate these signals close to the target processor.
TRACECLK (Trace mode)	Input	The Trace Clock pin provides DSTREAM-ST with the clock signal necessary to sample the trace data signals. You are advised to series terminate **TRACECLK** close to the target processor.

Table 2-4 CoreSight 20 signals (continued)

Signal	I/O	Description
VTRef	Input	The Voltage Target Reference pin supplies DSTREAM-ST with the debug rail voltage of the target to match its I/O logic levels. **VTRef** can be tied HIGH on the target. **Note** If **VTRef** is pulled HIGH by a resistor, its value must be no greater than 100Ω.
GND	-	Ground.
KEY	-	This pin must not be present on the target connector.

2.3 About Serial Wire Debug (SWD)

The following describes the *Serial Wire Debug* (SWD) connection to a *Debug Access Port* (DAP).

SWD connections

The diagram shows a typical SWD connection scheme.

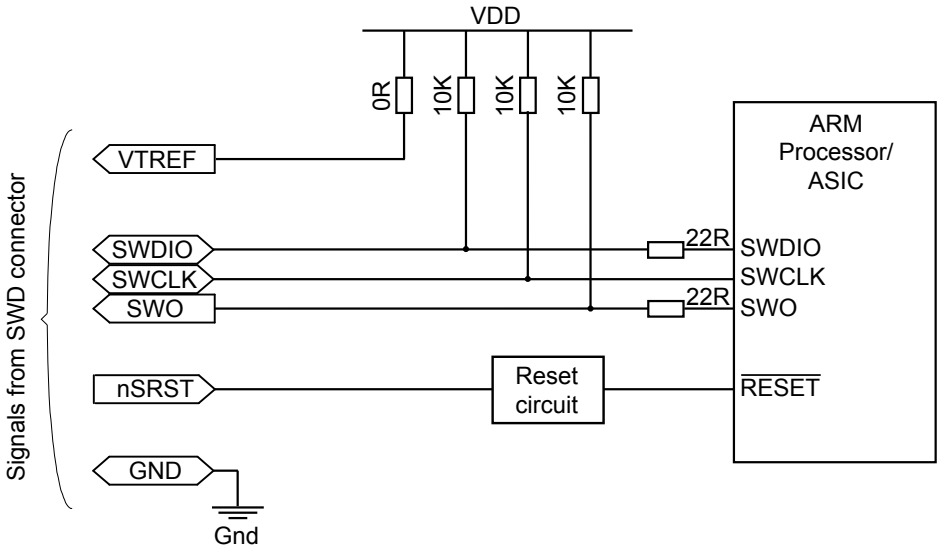

Figure 2-3 Typical SWD connections

Note

- **SWDIO** signal is bidirectional and the functionality is shared with a unidirectional JTAG TMS line. Ensure that there are no buffers preventing bidirectional communication functionality.
- The **SWDIO**, **SWCLK**, and **SWO** signals are typically pulled up on the target to keep them stable when the debug equipment is not connected.
- All pull-up resistors must be in the range 1K-100KΩ.
- The **VTRef** signal is typically connected directly to the VDD rail. If you use a series resistor to protect against short-circuits, it must have a value no greater than 100Ω.
- 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.

SWD timing requirements

The SWD interface uses only two lines, SWDIO and SWDCLK.

For clarity, the diagrams shown in the following figure separate the SWDIO line to show when it is driven by either the DSTREAM-ST unit or target:

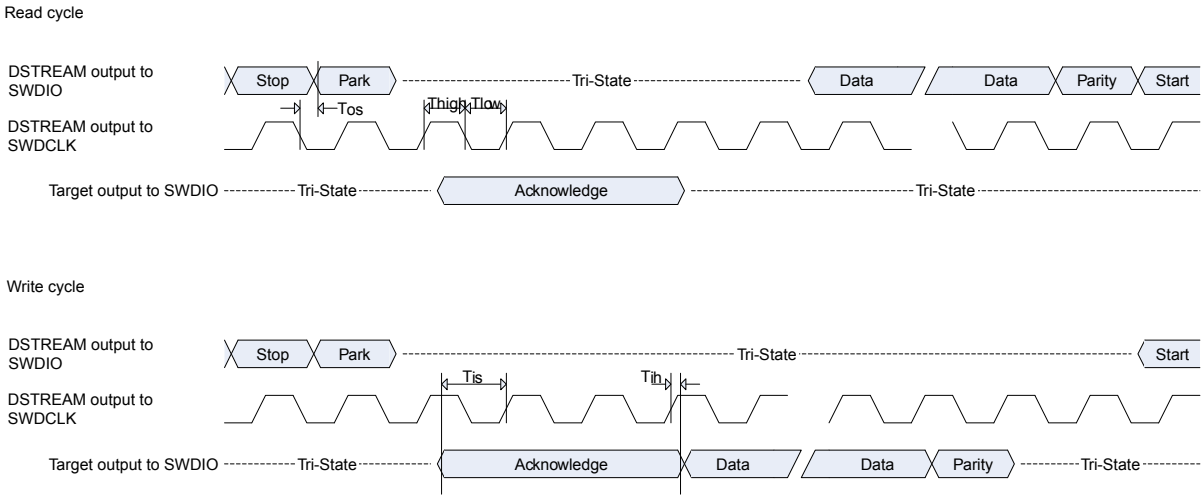

Figure 2-4 SWD timing diagrams

The DSTREAM-ST unit writes data to SWDIO on the falling edge of SWDCLK. The DSTREAM-ST unit reads data from SWDIO on the rising edge of SWDCLK. The target writes data to SWDIO on the rising edge of SWDCLK. The target reads data from SWDIO on the rising edge of SWDCLK.

The following table shows the timing requirements for the SWD:

Table 2-5 SWD timing requirements

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

2.4 About trace signals

Data transfer is synchronized by the **TRACECLK** signal.

Clock frequency

For capturing trace port signals synchronous to **TRACECLK**, the DSTREAM-ST trace feature supports up to 600Mbps per trace signal using DDR clocking mode.

——— **Note** ———

To achieve the maximum trace data rate, use the short (15cm) cable provided with DSTREAM-ST.

———————————

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

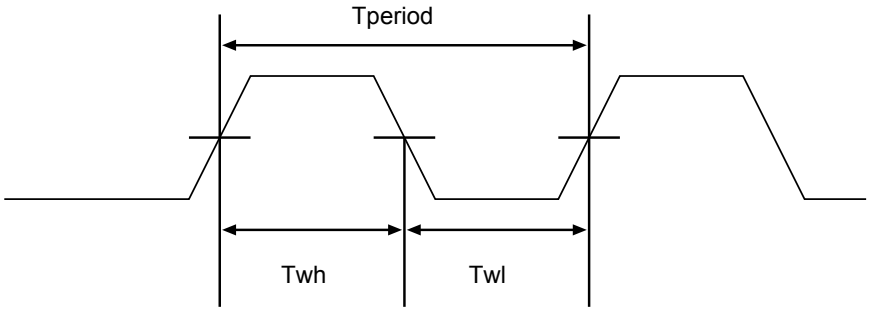

Figure 2-5 Clock waveforms

Table 2-6 TRACECLK frequencies

Parameter	DSTREAM-ST	Description
Tperiod (min)	2.08ns	Clock period
Twh (min)	1.0ns	High pulse width
Twl (min)	1.0ns	Low pulse width

Switching thresholds

The trace probe detects the target signaling reference voltage (**VTRef**) and automatically adjusts its switching thresholds to **VTRef**/2. For example, on a 3.3V target system, the switching thresholds are set to 1.65V.

Hot-plugging

If you power-up the DSTREAM-ST unit when it is plugged into an unpowered target, or if you plug an unpowered DSTREAM-ST unit into a powered target, trace functionality is not damaged.

If you connect an unpowered DSTREAM-ST unit to a powered target, there is a maximum leakage current into the DSTREAM-ST unit of ±10µA on any of the debug or trace signals.

2.5 About JTAG port timing characteristics

The JTAG port timing characteristics of the DSTREAM-ST unit are in-line with the requirements of the IEEE 1149.1 specification.

TMS and **TDI** are setup by the DSTREAM-ST unit on the falling edge of **TCK**. This is then sampled by the target on the rising edge of **TCK**. Ideally, the target device must output **TDO** signal on the falling edge of **TCK** for DSTREAM-ST to sample it on the next rising edge of **TCK**.

Note

Any delays in **TDO** are not critical, but might reduce the maximum frequency of the JTAG interface.

These timings are considered correct at the target JTAG connector. Delays that are introduced by the JTAG cable are compensated for within the DSTREAM-ST unit.

Since all signals are setup on the falling edge of **TCK** and sampled on the rising edge, the effective setup and hold times for the target device and the DSTREAM-ST unit is **Tclk**/2.

The following figure shows the JTAG port timing:

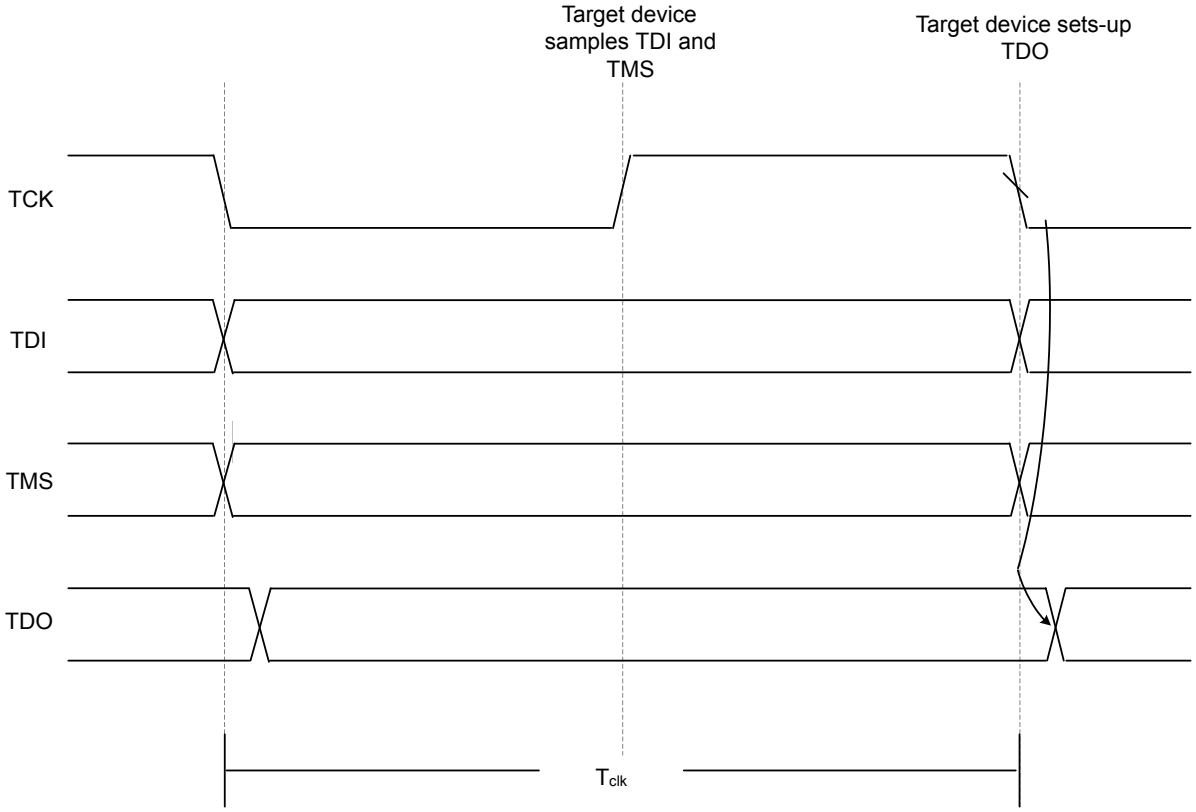

Figure 2-6 JTAG port timing diagram

Issues with minimum setup and hold times can always be resolved by decreasing the **TCK** frequency, because this increases the separation between signals changing and being sampled.

Note

There are no separate timing requirements for the adaptive clocking mode. In adaptive mode, the DSTREAM-ST unit samples **TDO** on the rising edge of **RTCK** and not **TCK**, so **TDO** timing is relative to **RTCK**.

The following table shows the timing requirements for the JTAG signals on the DSTREAM-ST probe:

Table 2-7 DSTREAM-ST JTAG Characteristics

Parameter	Min	Max	Description
T_{clk}	16.67ns	100ms	**TCK** period
T_{ds}	49%	51%	**TCK** Duty Cycle

2.6 About JTAG port buffering

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

In most cases, the JTAG connector of a target system connects to a single device, for example:

Note

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

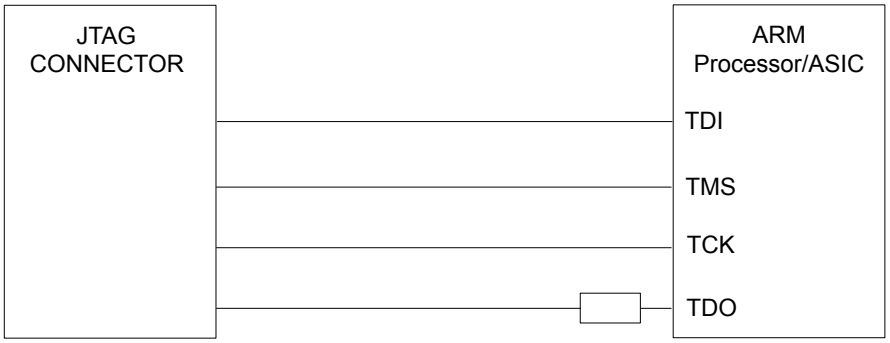

Figure 2-7 JTAG connection without buffers

A resistor must be placed close to the **TDO** pin of target device to act as a series terminator. This is the simplest scenario, and it is easy to achieve good signal integrity since each signal is point-to-point.

But, if the **TDO** output of the target device has a very weak drive-strength (<4mA), this could significantly limit the maximum frequency of the JTAG interface. You can resolve this by placing a buffer close to the **TDO** pin of the target device with the appropriate series termination resistor:

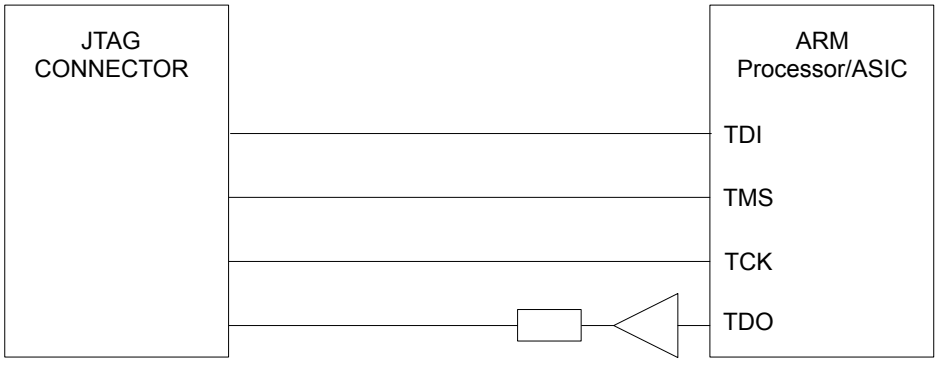

Figure 2-8 JTAG connection with TDO buffer

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

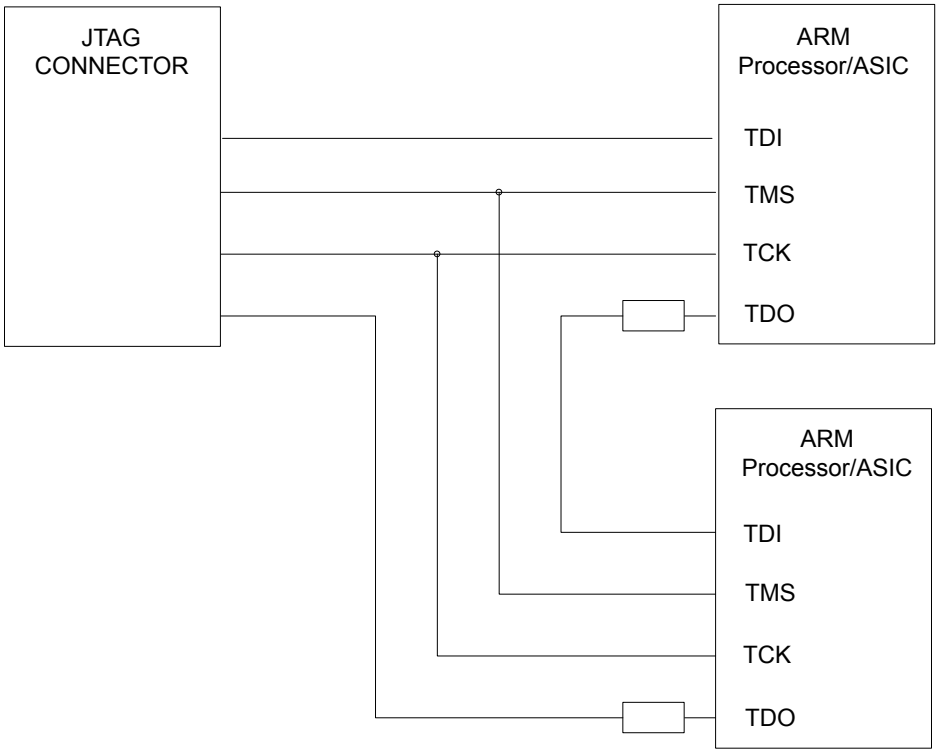

Figure 2-9 Daisy-chained JTAG connection without buffers

Achieving good signal integrity becomes difficult in this scenario due to the **TMS** and **TCK** signals being split several ways at T-junctions. The signal integrity of the **TMS** signal is not critical since it is ignored by the target device until a rising edge of **TCK** signal is detected. The signal integrity of the **TCK** signal is very critical, since any false edges cause the target device to sample **TDI** and **TMS** signals too many times, thereby corrupting the serial data stream as seen by the target devices.

To avoid this issue, buffering should always be used where the **TCK** signal is split:

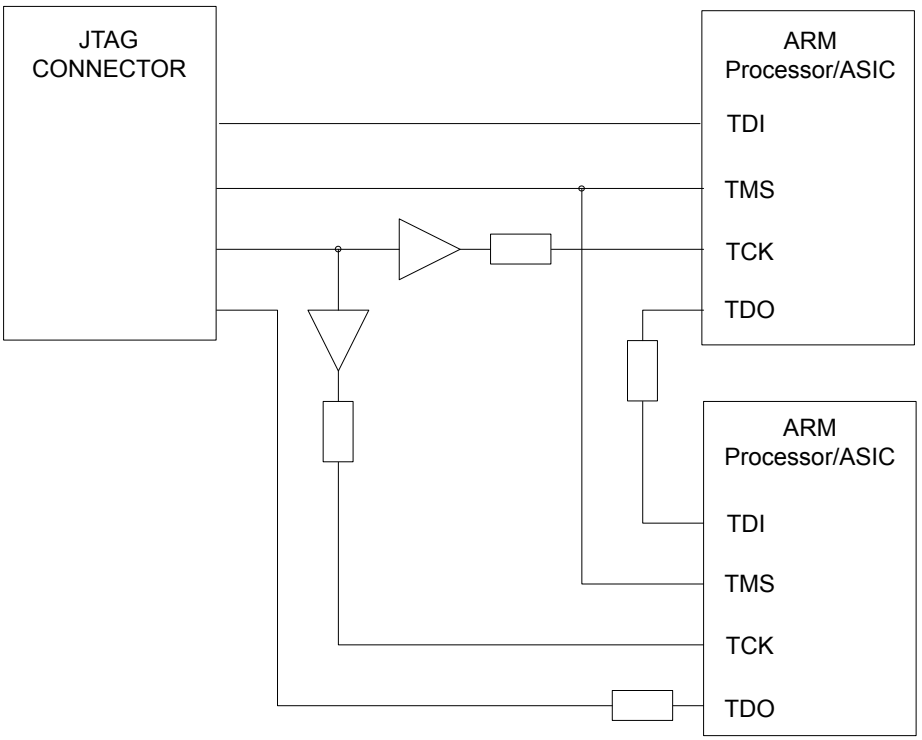

Figure 2-10 Daisy-chained JTAG connection with TCK buffers

This solution prevents the two **TCK** branches from interacting and ensures good signal integrity with minimal overshoot. The buffers and series termination resistors should be placed as close as possible to the T-junction of the **TCK** signal.

The only disadvantage to this solution is that it causes some skew between the **TDI**/**TMS** and **TCK** signals. It might be beneficial to use the same type of buffers on the **TDI** and **TMS** signals as on the **TCK** signals, for example:

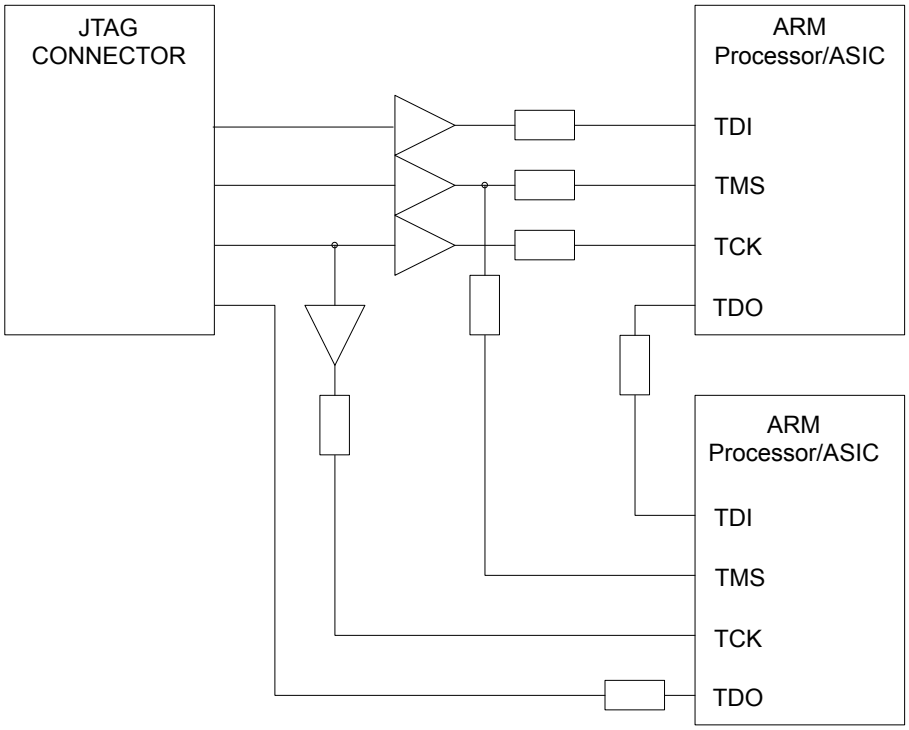

Figure 2-11 JTAG connection with de-skewed buffers

This solution matches the skew between **TDI**/**TMS** and **TCK** signals to achieve very high JTAG frequencies. By feeding **TMS** through separate resistors for each target device, termination of each signal is achieved without incurring the cost of extra buffers. All termination resistors should be placed as close as possible to the buffers.

Note

- For added noise rejection, Schmitt buffers may be used instead of standard buffers.
- Buffers with a drive strength of 24mA or above are recommended.

For guidance on selection of series termination resistors, see *2.9 Series termination* on page 2-37.

2.7 I/O diagrams for the DSTREAM-ST connectors

The diagrams show the pin I/O circuits for the debug and trace connectors on the DSTREAM-ST unit.

Diagram A - Input

The input circuit diagram is shown in the following figure:

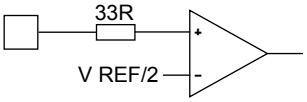

Figure 2-12 Input

Diagram B - Output

The output circuit diagram is shown in the following figure:

Figure 2-13 Output

Diagram C - Input/Output

The input/output circuit diagram is shown in the following figure:

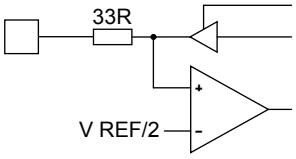

Figure 2-14 Input/Output

Diagram D - Reset output

The reset output circuit diagram is shown in the following figure:

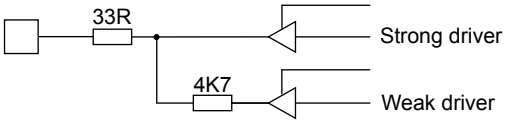

Figure 2-15 Reset output

Diagram E - Reset output with feedback

The reset output with feedback circuit diagram is shown in the following figure:

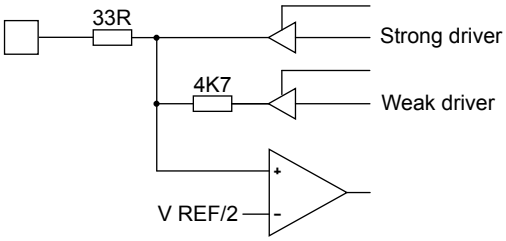

Figure 2-16 Reset output with feedback

Diagram F - VTRef input

The VTRef input circuit diagram is shown in the following figure:

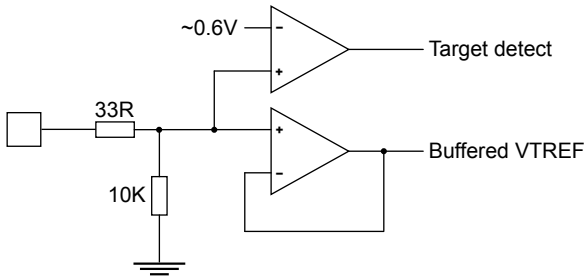

Figure 2-17 VTRef input

Diagram G - VTRef input (decoupled)

The VTRef input (decoupled) circuit diagram is shown in the following figure:

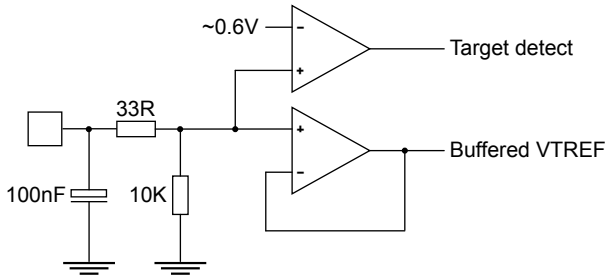

Figure 2-18 VTRef input (decoupled)

Diagram H - AC Ground

The AC Ground circuit diagram is shown in the following figure:

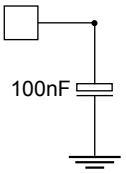

Figure 2-19 AC Ground

2.8 Voltage domains of the DSTREAM-ST unit

The DSTREAM-ST unit supports two separate voltage domains to work with debug and trace interfaces on differing voltage rails.

When using either the CoreSight 20 or ARM JTAG 20 connector, only one voltage domain is supported and is preset through the **VTRef** signal from the target.

When using the MICTOR adapter with both debug cables, two voltage domains are supported:

- Debug **VTRef**.
- Trace **VTRef**.

If only one of these pins is connected to a valid voltage on the target, it will be used for both debug and trace signals.

2.9 Series termination

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

This is achieved by reducing the source voltage by approximately 50% close to the driver. When the signal reaches the end of the transmission line, the high impedance of the receiver causes a reflection which approximately doubles the signal back to its original amplitude. When the reflection returns to the series terminating resistor, the potential across the resistor drops to zero which prevents any more current from entering the transmission line. From the perspective of the receiver, this gives a perfect 100% logic transition without any overshoot or ringing.

ARM recommends that all outputs from the target system be simulated, and, if necessary, series terminated to ensure that a reliable signal is delivered to the DSTREAM-ST unit. Some overshoot/ undershoot is acceptable but it is recommended to keep this below ~0.5V. Beyond this point, the clamping diodes at the receivers will start to cause high transient currents which in turn cause increased crosstalk, radio emissions and target power usage.

The target signal impedance for use with DSTREAM-ST is 50Ω.

The following table lists some typical series terminating resistor values for instances when the outputs cannot be simulated.

Table 2-8 Typical series terminating resistor values

Driver strength	Typical series terminator	
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. This is usually achieved by using weaker drive transistors to slow down the edge transitions. This has the side effect of limiting the data throughput of the driver.

To achieve the highest data rates with the best signal integrity, it is recommended to use a strong and fast driver and appropriate series terminating resistor.

If it is determined that series terminating resistors are not required, it is recommended that 0Ω links be placed close to the driver as a fall-back option.

When series terminating multiple signals, it is common to use small quad resistor packages. This saves board space and reduces parasitic effects without much risk of placement or tombstoning issues during production.

Chapter 3
ARM® DSTREAM-ST USER Input/Output (IO) connections

This chapter describes the additional input and output connections provided in the ARM DSTREAM-ST unit.

It contains the following section:

3.1 About the USER I/O connector pinouts

The USER *Input/Output* (IO) port is used to set up custom input or output connections to your target.

The USER I/O connector is a 10-way 2.54mm pitch *Insulation Displacement Connector* (IDC) header that mates with IDC sockets mounted on a ribbon cable. The USER IO port is at the rear of the DSTREAM-ST unit.

The following figure shows the pin connector details.

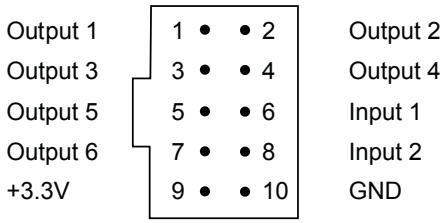

Figure 3-1 USER I/O connector pinouts

——— Warning ———

You must establish a common ground between the DSTREAM-ST unit and the target hardware before you connect any of the USER I/O signals.

———————————

USER I/O pinouts

The table shows the USER I/O pinouts available on DSTREAM-ST.

Table 3-1 USER I/O pin connections

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

Table 3-1 USER I/O pin connections (continued)

Pin	Signal	I/O	Description
Pin 8	**Input 2**	Input	This is a user input bit. It has a 100K weak pull-up to the unit internal +3.3V supply, and requires a $V_{ih(min)}$ of 2.0V and a $V_{il(max)}$ of 0.8V. It can safely be driven by 5V logic levels, and has *Electro Static Discharge* (ESD) protection greater than the 2kV human body model. This pin is not currently supported by the DSTREAM-ST firmware.
Pin 9	**+3.3V**	Output	This is intended as a voltage reference for external circuitry and is current limited to approximately 50mA.
Pin 10	**GND**	-	-

Note

USER inputs are not currently supported by the DSTREAM-ST firmware and are reserved for future use.

Chapter 4
Target board design for tracing with ARM® DSTREAM-ST

This chapter describes some considerations for the design of a target board that can be connected to the DSTREAM-ST trace feature.

It contains the following sections:

4.1 Overview of high-speed design

Failure to observe high-speed design rules when designing a target system containing an ARM *Embedded Trace Macrocell* (ETM) trace port can result in incorrect trace data being captured. You must give serious consideration to high-speed signals when designing the target system.

The signals coming from an ETM trace port or from a Trace Port Interface Unit (TPIU) can have very fast rise and fall times, even at relatively low frequencies. For example, a signal with a rise time of 1ns has an effective knee frequency of 500MHz and a signal with a rise time of 500ps has an effective knee frequency of 1GHz ($f_{knee} = 0.5/Tr$).

Note

These principles apply to all of the trace port signals, but special care must be taken with **TRACECLK**.

You must make the following considerations for high-speed design:

Avoid stubs

Stubs are short pieces of track that tee off from the main track carrying the signal to, for example, a test point or a connection to an intermediate device. Stubs cause impedance discontinuities that affect signal quality and must be avoided.

Special care must therefore be taken when trace signals are multiplexed with other pin functions and where the PCB is designed to support both functions with differing tracking requirements.

Minimize crosstalk

Normal high-speed design rules must be observed. For example, do not run dynamic signals parallel to each other for any significant distance, keep them spaced well apart, and use a ground plane and so forth. Particular attention must be paid to the **TRACECLK** signal. If in any doubt, place grounds or static signals between the **TRACECLK** and any other dynamic signals.

Use impedance matching and termination

All PCB tracks carrying trace port signals must be impedance matched to approximately 50Ω. Series termination is highly recommended on all high-speed signals.

4.2 PCB track impedance

You can calculate the PCB track impedance using the microstrip impedance formula.

$$\text{Impedance} = \frac{87}{\sqrt{(E_r + 1.41)}} \ln\left[\frac{5.98h}{(0.81w + t)}\right] \Omega$$

where:

h
: Height above ground plane (inches)

w
: Trace width (inches), and `0.1 < w/h < 2`

t
: Trace thickness (inches)

E_r
: Relative permittivity of processor/prepreg, and $1 < E_r < 15$

This PCB track impedance formula applies only to microstrips (track on outer layer over a ground plane).

The dimensions `h`, `w`, and `t` are shown in the following figure.

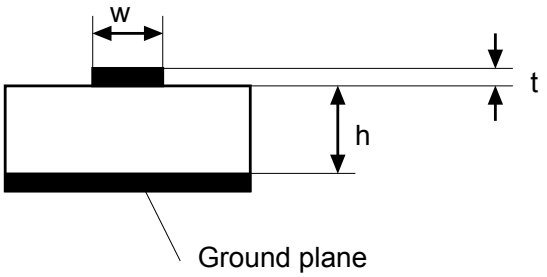

Figure 4-1 Track impedance

As an example, the following track (in microstrip form) has an impedance of 51.96Ω:

h
: 0.005 inch height above ground

w
: 0.007 inch width track

t
: 0.0014 inch thickness (1 oz. finished weight)

E_r
: 4.5 (FR4 laminate)

Note

As the track width increases, the impedance decreases.

4.3 Signal requirements

Use the information below to understand the data setup and hold requirements, and switching thresholds for the ARM DSTREAM-ST unit.

Data setup and hold

The following figure and table show the setup and hold timing of the trace signals with respect to **TRACECLK**.

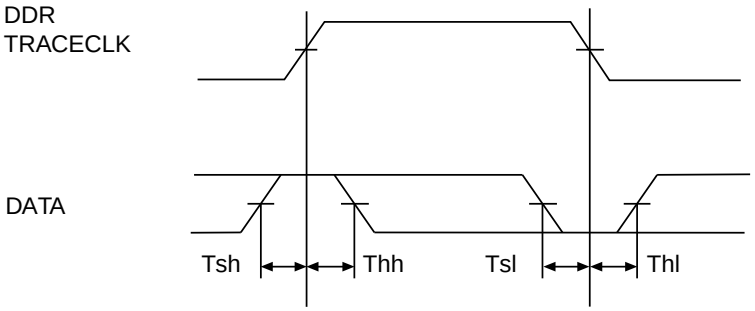

Figure 4-2 Data waveforms

Table 4-1 Data setup and hold

Parameter	DSTREAM-ST	Description
Tsh (min)	0.75ns	Data setup high
Thh (min)	0.75ns	Data hold high
Tsl (min)	0.75ns	Data setup low
Thl (min)	0.75ns	Data hold low

Note

DSTREAM-ST supports DDR clocking mode. Data is output on each edge of the **TRACECLK** signal and **TRACECLK (max)** <= 300MHz.

Switching thresholds

The DSTREAM-ST senses the target signaling reference voltage (**VTRef**) and automatically adjusts its switching thresholds to **VTRef**/2. For example, on a 3.3V target system, the switching thresholds are set to 1.65V.

4.4 Modeling

For trace bit rates of 0-600Mbps, basic signal integrity can be established using simplified modeling. The bulk of the transmission line model consists of the cable used between the DSTREAM-ST unit and 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 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 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 circuit at the DSTREAM-ST 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/zero parasitics.
- All capacitors can be modeled as their ideal capacitance values with minimum/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 and data can be assumed to be valid when it is 100mV above or below this threshold.
- Output drivers can be modeled using the Spartan 3 LVCMOS Fast 16mA model. The model voltage must be chosen to match the target system voltage.

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

You are strongly advised to use series termination on all target outputs to achieve good signal integrity.

Chapter 5
Reference

Lists other information that might be useful when working with DSTREAM-ST.

It contains the following section:

5.1 About adaptive clocking to synchronize the JTAG port

ARM architecture-based devices that use only hard macrocells, such as *ARM7TDMI* and *ARM920T*, use the standard five-wire JTAG interface. However, some target systems require that JTAG events are synchronized to a clock in the system. The adaptive clocking feature of DSTREAM-ST addresses this requirement.

The standard five-wire JTAG interface comprises the **TCK**, **TMS**, **TDI**, **TDO**, and **nTRST** signals. To ensure a valid JTAG CLK setting, systems that require the JTAG events to be synchronized to a clock in the system often supports an extra signal (**RTCK**) at the JTAG port:

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

When adaptive clocking is enabled, DSTREAM-ST issues a **TCK** signal and waits for the **RTCK** signal to come back. DSTREAM-ST does not progress to the next **TCK** until **RTCK** is received.

Note

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

You can use adaptive clocking as an interface to targets with slow or widely varying clock frequency, such as battery-powered equipment that varies its clock speed according to processing demand. In this system, **TCK** might be hundreds of times faster than the system clock, and the debugger loses synchronization with the target system. Adaptive clocking ensures that the JTAG port speed automatically adapts to slow system speed.

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

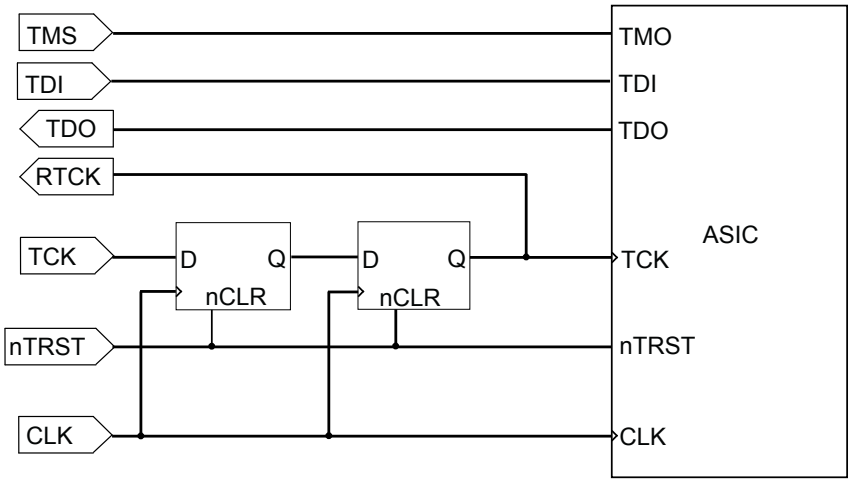

Figure 5-1 Basic JTAG port synchronizer

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

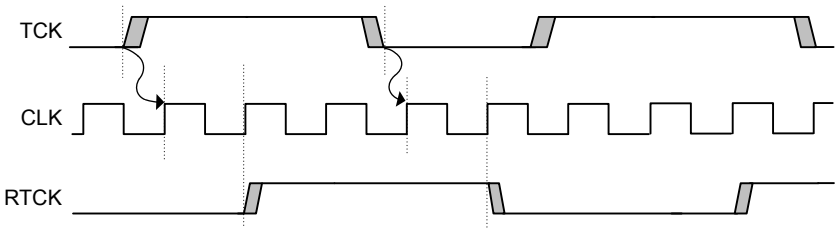

Figure 5-2 Timing diagram for the Basic JTAG synchronizer

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

The following figure shows one possible implementation of this circuit.

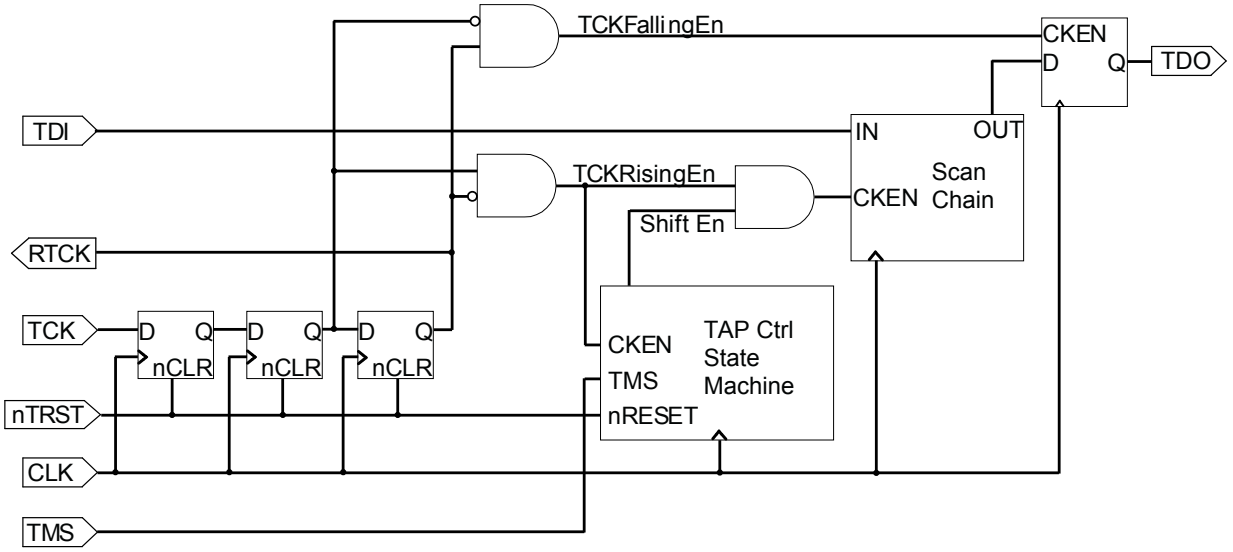

Figure 5-3 JTAG port synchronizer for single rising-edge D-type ASIC design rules

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

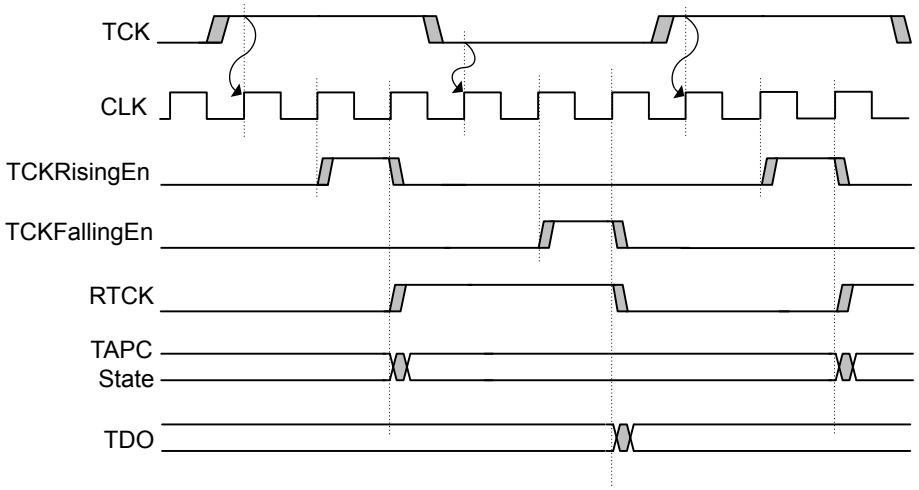

Figure 5-4 Timing diagram for the D-type JTAG synchronizer