Application Note 132

Connecting Multiple JTAG Devices

Document number: ARM DAI 132A
Issued: 1st June, 2008
Copyright ARM Limited 2008

ARM®

Copyright © 2008 ARM Limited. All rights reserved.
Open Access

Application Note 132
Connecting Multiple JTAG Devices

Copyright © 2008 ARM Limited. All rights reserved.

Release information

Change history

Date	Issue	Change
June 2008	A	First release

Proprietary notice

Words and logos marked with © and ™ are registered trademarks owned by ARM Limited, except as otherwise stated below in this proprietary notice. Other brands and names mentioned herein may be the trademarks of their respective owners.

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

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

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

Confidentiality status

This document is Open Access. This document has no restriction on distribution.

Feedback on this Application Note

If you have any comments on this Application Note, please send email to errata@arm.com giving:

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

General suggestions for additions and improvements are also welcome.

ARM web address

`http://www.arm.com`

Table of Contents

Copyright © 2008 ARM Limited. All rights reserved.
Open Access

1. Introduction

Hard ARM macrocells such as the ARM7TDMI are generally fully JTAG compliant and do not require clock synchronization. These cores can accept TCK at any speed, even if TCK is faster than the core clock. With ARM cores that have JTAG inputs (TCK, TDI, TDO, TMS) you may connect the same TCK and TMS to each TAP controller.

Synthesizable ARM cores and some foundry parts do not have a JTAG compliant interface, and require the use of a synchronization circuit as described in the Technical Reference Manuals and Multi-ICE / RealView-ICE user guides. This circuit synchronizes TCK to the ARM cores internal clock domain, generates the DBGTCKEN input required by the ARM core and provides a feedback clock (RTCK) which enables the debugger to use adaptive clocking speeds.

1.1 Document Scope

This document will address the issue of connecting multiple JTAG devices (specifically synthesizable ARM cores) in a daisy-chain configuration on one ASIC. Daisy chaining TAP controllers is relatively simple to implement, however, complications can arise when connecting together ARM cores running at different clock frequencies. Three main situations will be discussed:

- Connecting together synthesizable ARM cores running at the same clock speed
- Connecting together synthesizable ARM cores running at different clock speeds
- Connecting together ARM cores with other JTAG devices (including other ARM cores)

Within this document devices are labeled as MultiICE would see them, from 0 (the first device after TDI) through to n-1, where n is the number of devices in the chain.

For clarity, pad pull up/pull down information has been omitted from this document.

Copyright © 2008 ARM Limited. All rights reserved.
Open Access

2. ARM cores with a common clock

ARM cores running at the same clock speed can easily be connected in a daisy-chain configuration. For cores that require adaptive clocking, TCK synchronization and RTCK generation logic is needed. Only one synchronizer is required in this case, and the synchronizer outputs can be parallel loaded onto each of the TAP controllers in the daisy chain as shown by devices 0-2 in the following diagram.

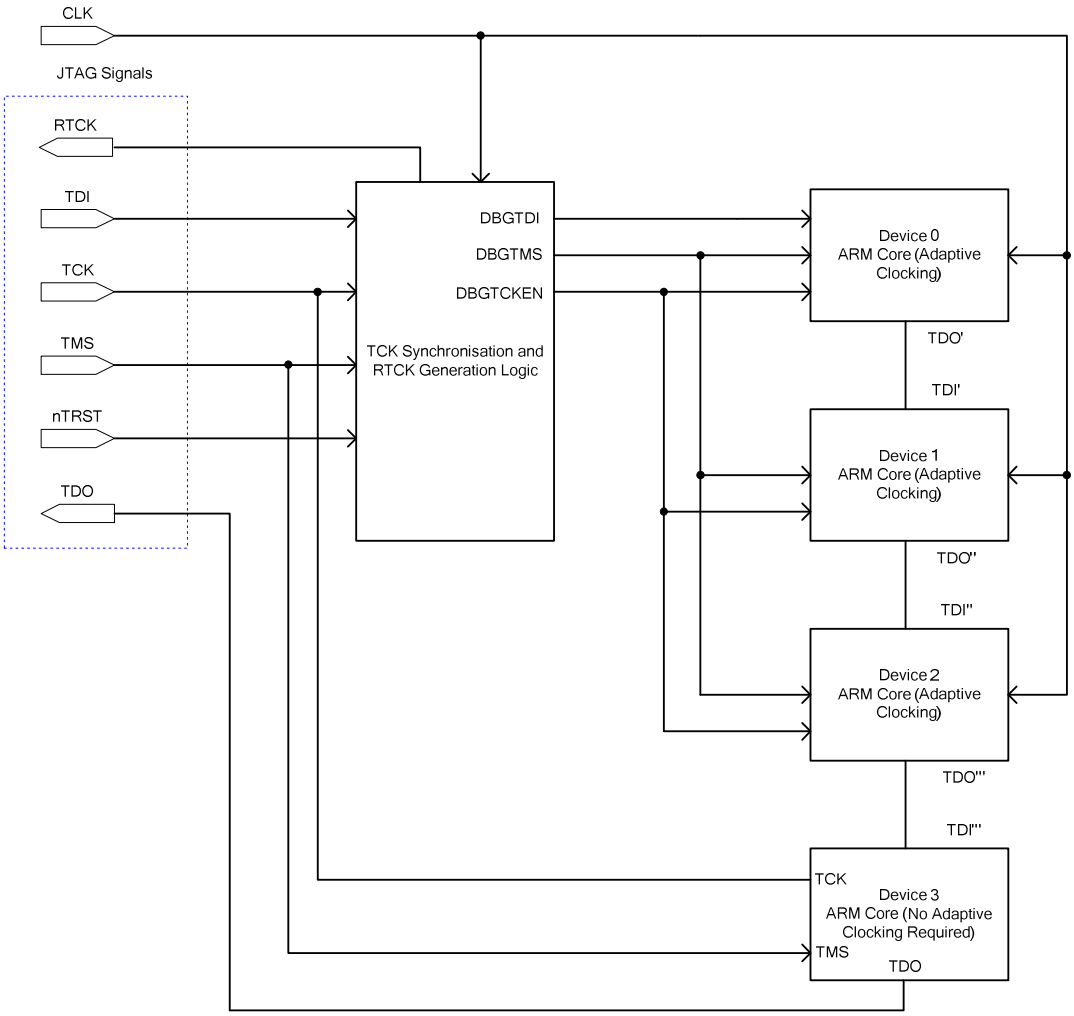

Figure 1: Daisy chain with 4 TAP controllers running from a common clock

Copyright © 2008 ARM Limited. All rights reserved.
Open Access

3. ARM cores with different clocks

If a daisy chain is to be formed for ARM cores that are clocked at different speeds, each require synchronization and RTCK generation logic irrespective of the relationship between the individual clocks. Each synchronizer takes TCK from the JTAG port and generates its own DBGTCKEN and RTCK output, and then some additional logic is required to create a single RTCK output signal (that is, the signal connected to the debugger, $RTCK_{tot}$). $RTCK_{tot}$ must only change when all the individual RTCK signals have changed. The function of this extra logic is summarized in the following state diagram.

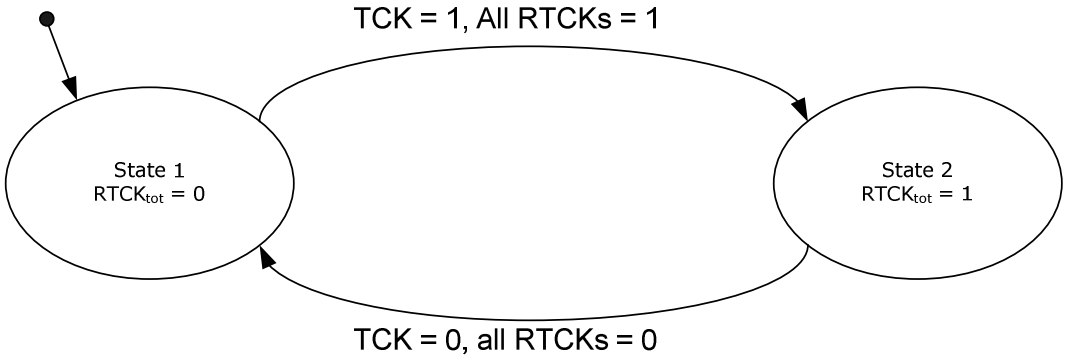

Figure 2: State diagram for $RTCK_{tot}$ generation logic

This logic is implemented as follows for 4 RTCK signals. If *n* RTCK signals are present, simply extend the logic gates AND_1 and OR_1 into *n*-input logic gates.

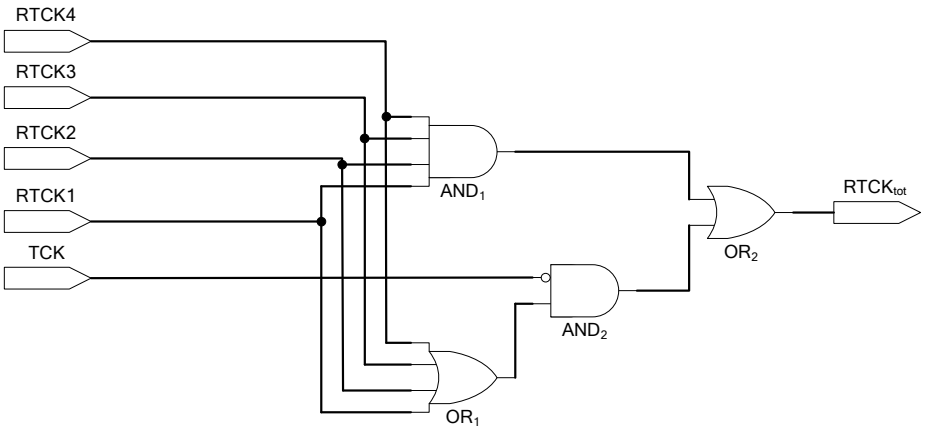

Figure 3: $RTCK_{tot}$ generation logic

An example implementation of daisy chaining 3 processors all running at different clock frequencies is shown in figure 4.

Copyright © 2008 ARM Limited. All rights reserved.
Open Access

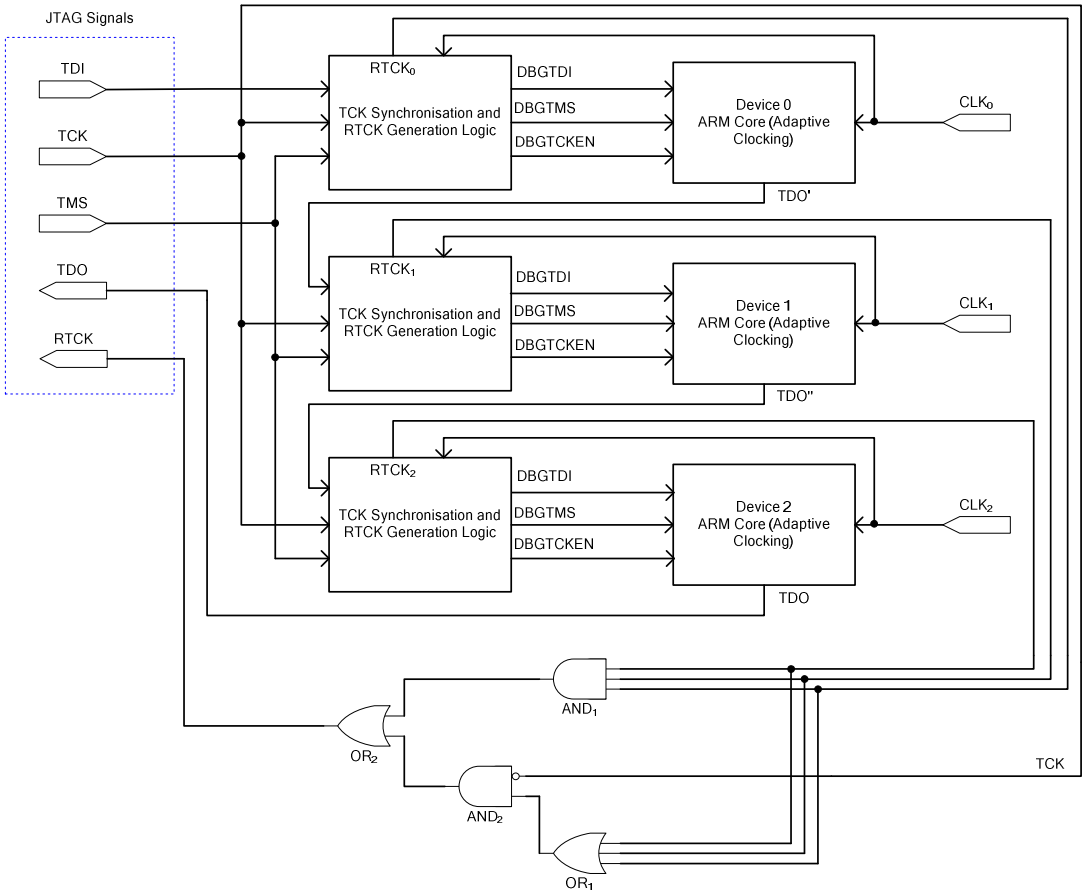

Figure 4: Example implementation of a daisy chain with 3 processors running at different clock frequencies

To avoid the use of $RTCK_{tot}$ generation logic, the TCK synchronization logic can be daisy chained. Simply connect the RTCK of the previous logic module to the TCK of the next logic module in the chain, with the first module connected to the JTAG TCK and RTCK from the last module connected to the debugger. This implementation is not desirable for most applications, as it is extremely slow as you are adding registers on the path of TCK to RTCK.

Copyright © 2008 ARM Limited. All rights reserved.
Open Access

4. Daisy chaining synchronized ARM cores with other JTAG devices

ARM cores can be daisy chained with any other JTAG compliant device (including ARM processor cores not requiring synchronization). All other devices on the chain can have their TCK and TMS inputs driven directly from the JTAG port, but care must be taken in the connection of TDI and TDO. The device that precedes the synchronized ARM cores must have its TDO output driven in to the TDI input of the synchronizer block. The following device must have its TDI input connected to the TDO output of the last ARM core.

All devices requiring synchronization should be grouped together preferably before any other device on the chain as this makes timing constraints easier, and RTCK generated using one of the mechanisms discussed.

Although the TCK input is taken from the JTAG TCK port or synchronizer RTCK output, TMS is always taken from the JTAG port.

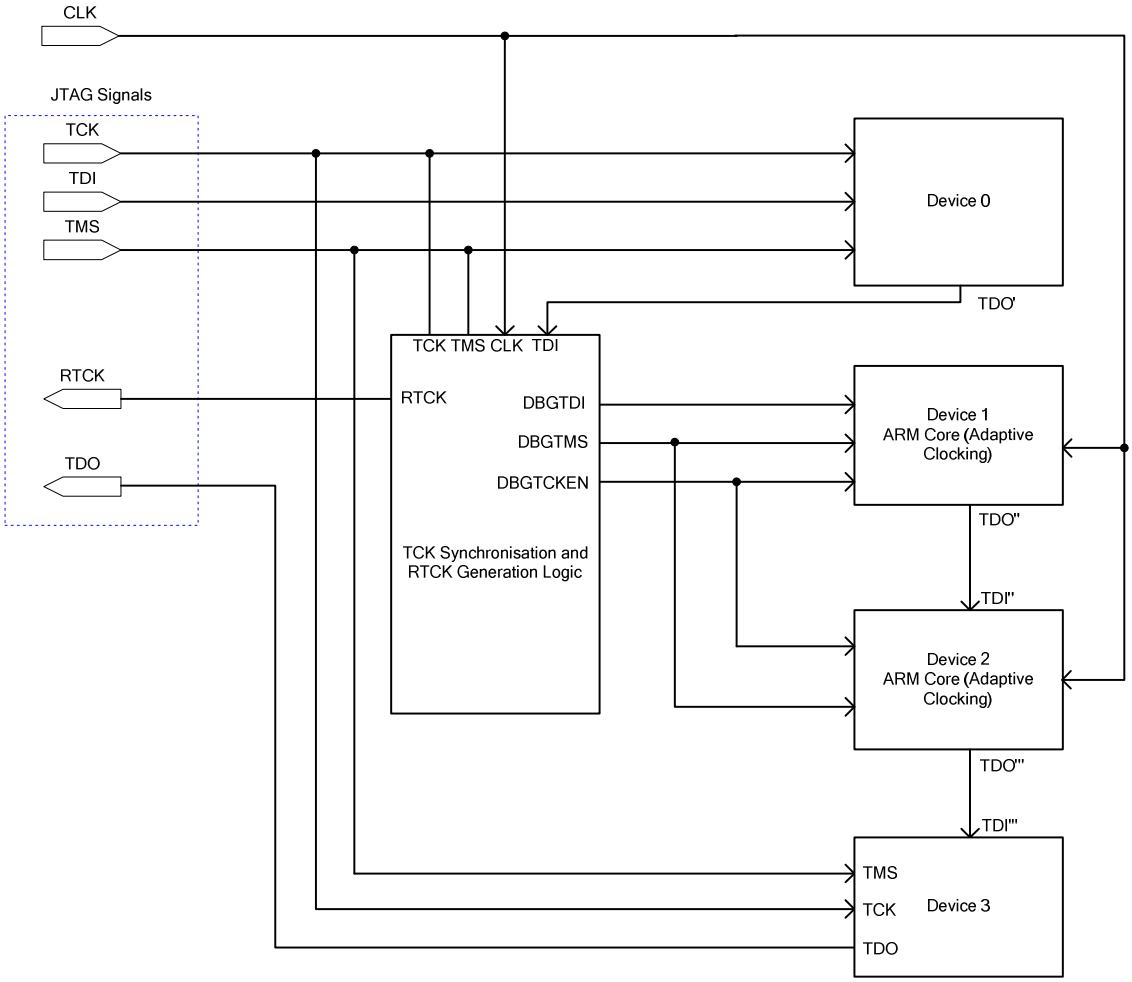

Figure 5: Example implementation of different devices (two ARM cores running at the same frequency)

Care must also be taken when connecting other ARM devices, such as the Embedded Trace Macrocell which sometimes require the TAP controller to be wired in parallel with the ARM core TAP controller. You should follow the documentation of these devices for correct hook up.

Copyright © 2008 ARM Limited. All rights reserved.
Open Access