

# Application Note 139

## ARM<sup>®</sup> RealView<sup>®</sup> ETB Support

Document Number: ARM DAI 0139A

Issued: January 2005

Copyright ARM Limited 2005

# ARM<sup>®</sup>

## Application Note 139

### ARM RealView ETB Support

Copyright © 2005 ARM Limited. All rights reserved.

#### Release information

The following changes have been made to this Application Note.

Change history		
Date	Issue	Change
January 2005	A	First Release

#### Proprietary notice

ARM, RealView and the ARM logo are registered trademarks of ARM Limited. ARM926EJ-S, ARM966E-S, ARM1136J-S Embedded Trace Buffer and ETB are trademarks of ARM Limited.

All other products, or services, mentioned herein may be trademarks of their respective owners.

#### 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](mailto: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>

## ***Intended Audience***

This note provides essential guidance to ASIC, ASSP and other designers of ARM-powered SoC devices who are including an on-chip Embedded Trace Buffer™ (ETB™) and should be read in conjunction with the ETB Technical Reference Manual (ARM DDI 0242) or the ETB11 Technical Reference Manual (ARM DDI 0275).

This note only gives guidance on the ordering of devices connected in to the JTAG scan chain.

This note does not address connectivity of the Embedded Trace Macrocell (ETM). The JTAG interface of the ETM is normally connected to scan chain 6 of the TAP controller in the associated ARM core, so not shown in the primary JTAG scan chains illustrated in this note.

This note assumes that each ETM is implemented in a manner such that there is a data path from the ETM to the associated ETB. How this is done is not addressed in the note.

Failure to adhere to the guidance below may result in the ASIC not being supported by the ARM RealView debug tool chain

Contact [support@arm.com](mailto:support@arm.com) should further clarification be required.

## ***Terms***

Embedded Trace Buffer (ETB) means either the ARM Embedded Trace Buffer, product code TM060, or the ARM11 Embedded Trace Buffer (ETB11), product code TM070.

## ***RealView ETB Support***

### ***One Core with One ETB***

Configuring ETB support for one core and one ETB is trivial; simply set the ETB option (using RVConfig) on the core to indicate that ETB trace should be used.

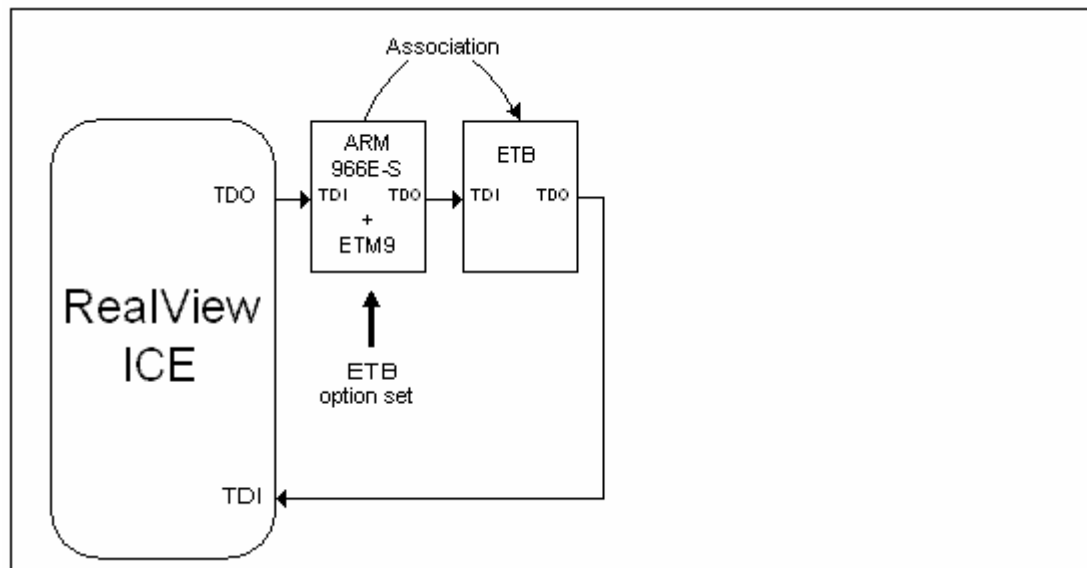


Figure 1: JTAG scan chain ordering for one core, one ETB

## ***Two (or more) Cores with One ETB.***

In the following scheme there is only one ETB in the scan chain. The cores both have an ETB option which can be set by the user at configuration time (using RVConfig).

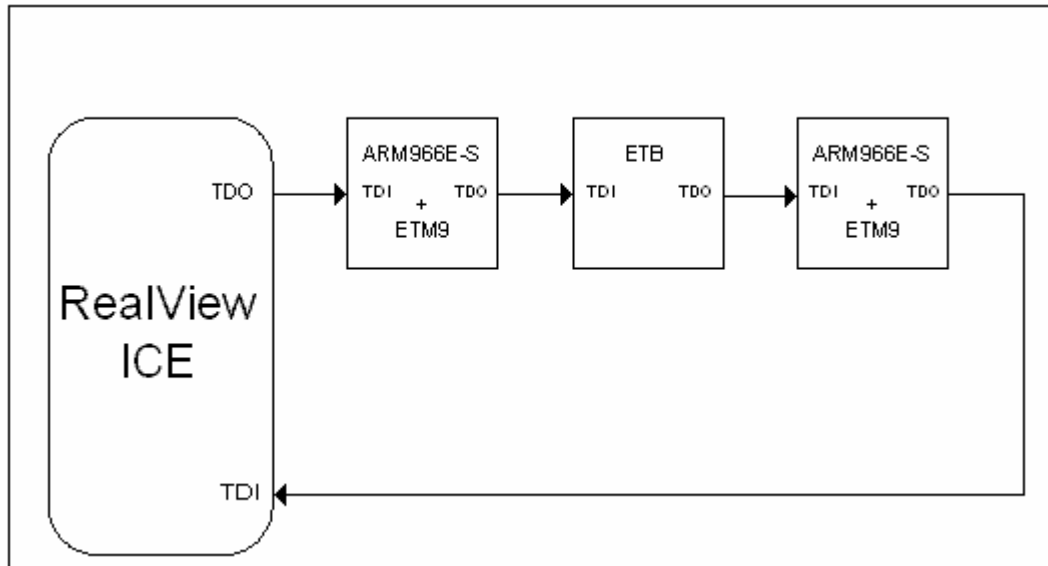


Figure 2: JTAG scan chain ordering for two cores, one ETB

If the ETB option is set on the first ARM966E-S™ then the tools will decode the trace capture based on the code image in that core's memory.

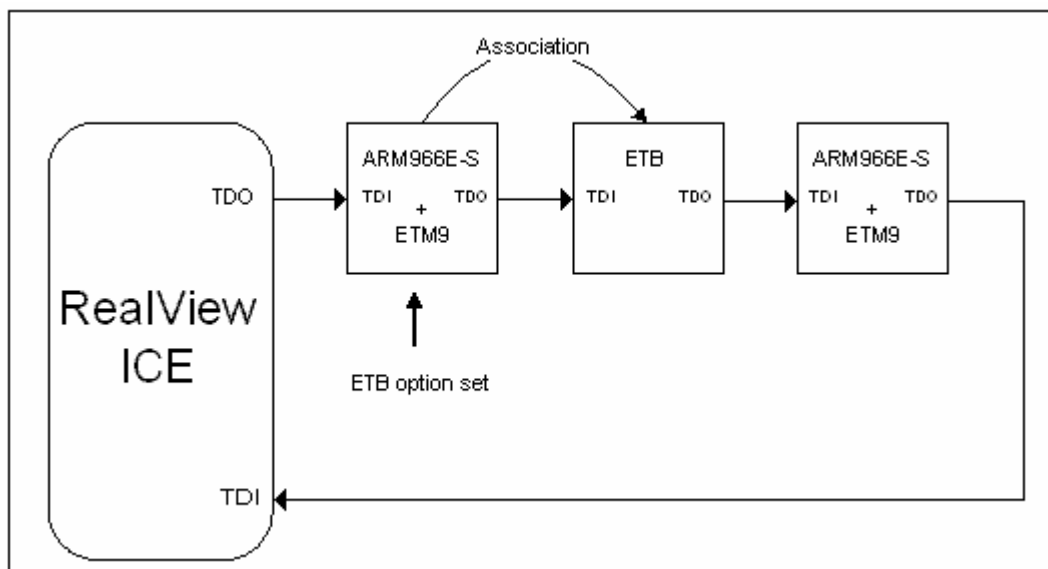


Figure 3: ETB option set for first of two cores, one ETB

If the ETB option is set on the second ARM966E-S™ then the tools will decode the trace capture based on that cores memory contents.

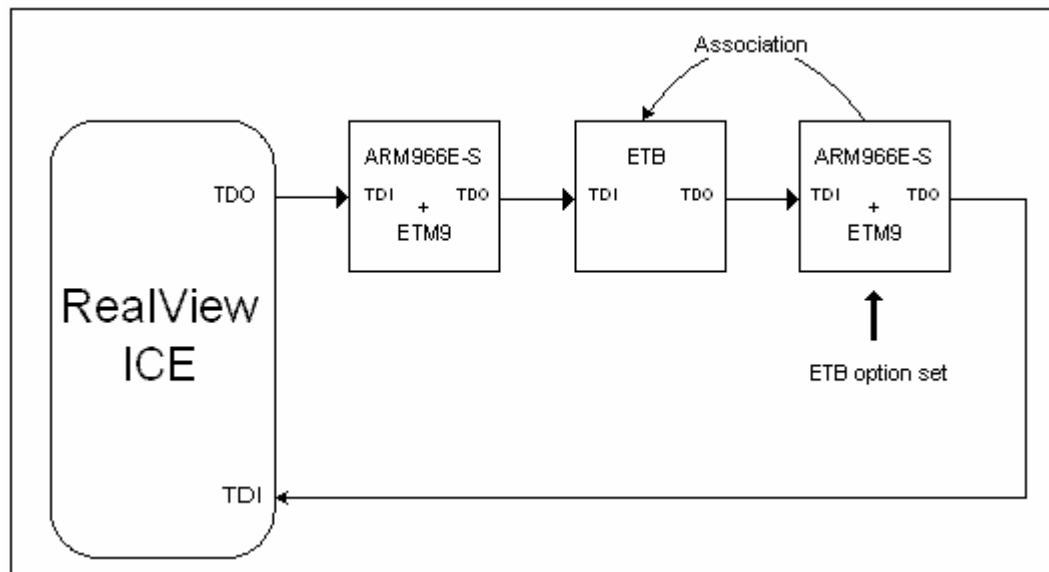


Figure 4: ETB option set for second of two cores, one ETB

NOTE: It is illegal to set the ETB option on both cores. Two cores cannot be associated with the same ETB at the same time.

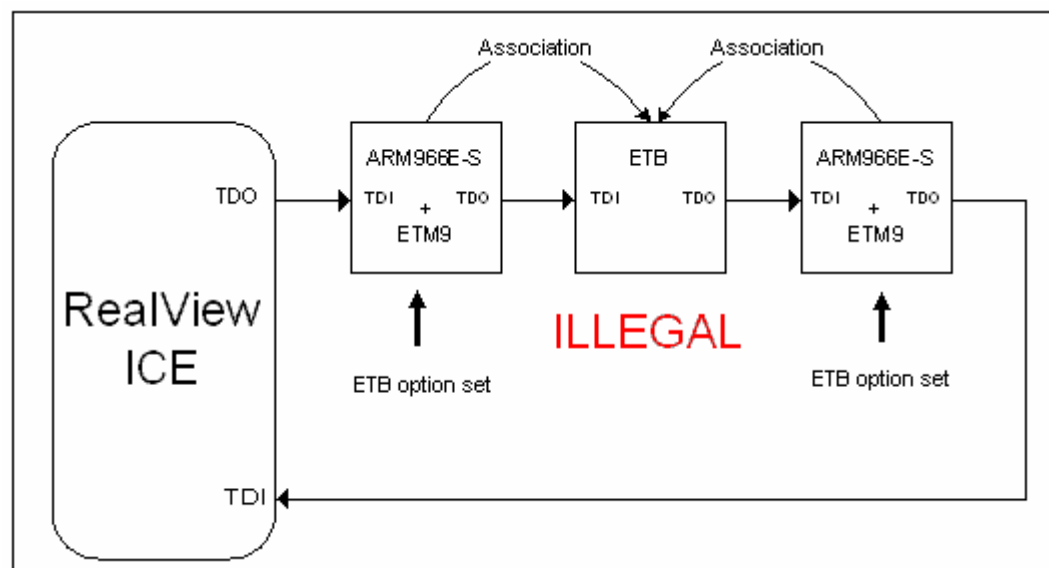


Figure 5: Illegal setting of both ETB options for two cores, one ETB

## Multiple Cores with Multiple ETBs

Consider the following scheme:

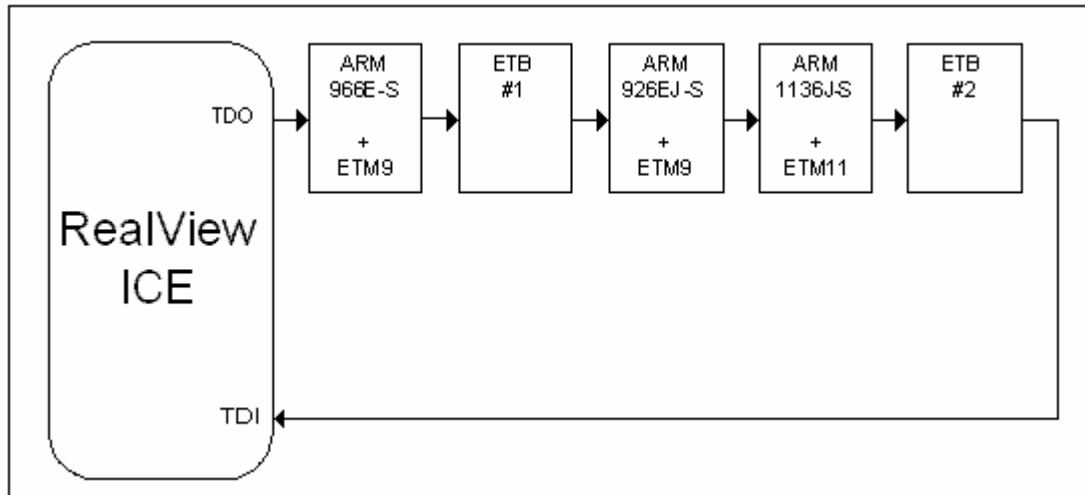


Figure 6: JTAG scan chain ordering for three cores, two ETBs

In the above scheme there are now two ETBs and three cores in the scan chain. The cores all have an ETB option and two could be set legally. However, the tools make the association on a first come first serve basis; that is, the first core in the chain that has the ETB option set is associated with the first ETB found in the scan chain (ETB #1) and the second core in the scan chain that has the ETB option set is associated with the second ETB in the scan chain (ETB #2).

Let's say we wanted to associate ETB #1 with the ARM966E-S and ETB #2 with the ARM1136J-S, simply set the ETB option on those 2 cores.

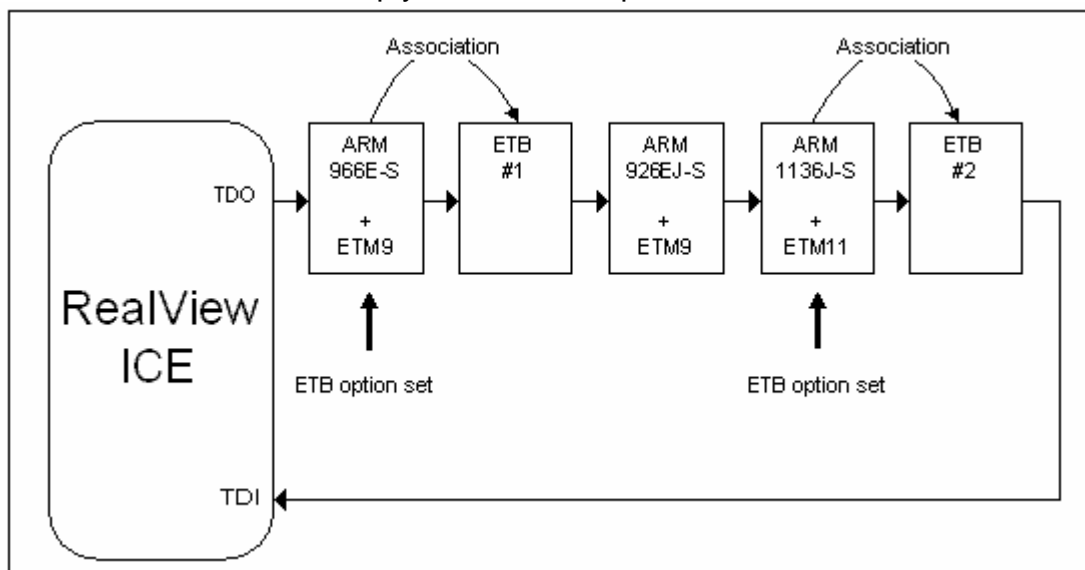


Figure 7: ETB option set for two out of three cores, two ETBs, example 1

Similarly, if we wanted to associate ETB #1 with the ARM926EJ-S and ETB #2 with the ARM1136J-S then we could do so by setting the ETB option on each of the cores. The association would then look like this:

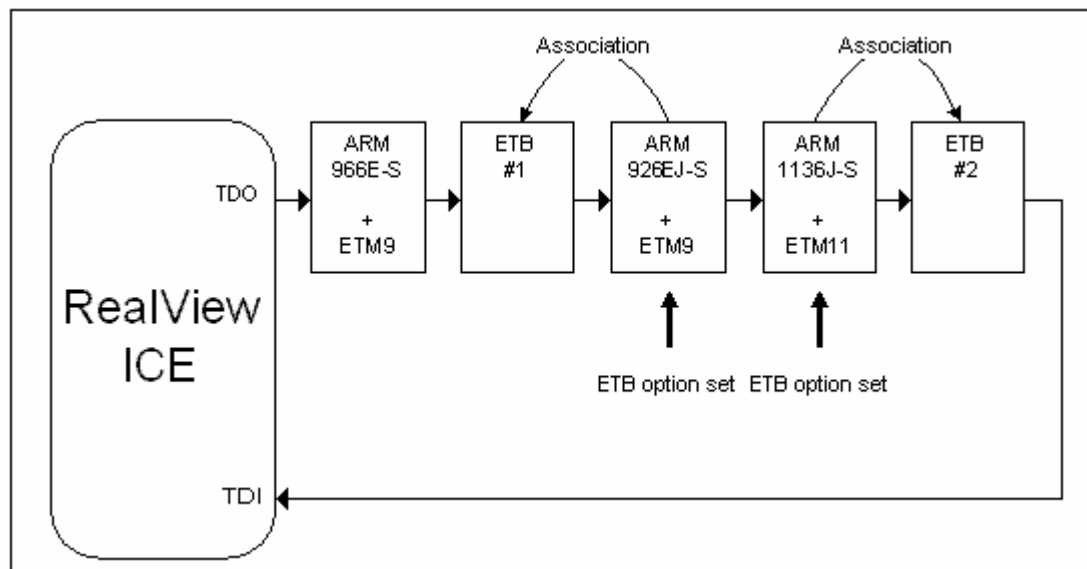


Figure 8: ETB option set for two out of three cores, two ETBs, example 2

However, it is not possible to associate ETB #2 with the ARM966E-S and ETB #1 with the ARM1136J-S. This is not possible because the associations are out of order; ETB #2 comes after ETB #1. If we did set the options on those cores then the association would be the same as in figure 7, i.e. not what we intended.

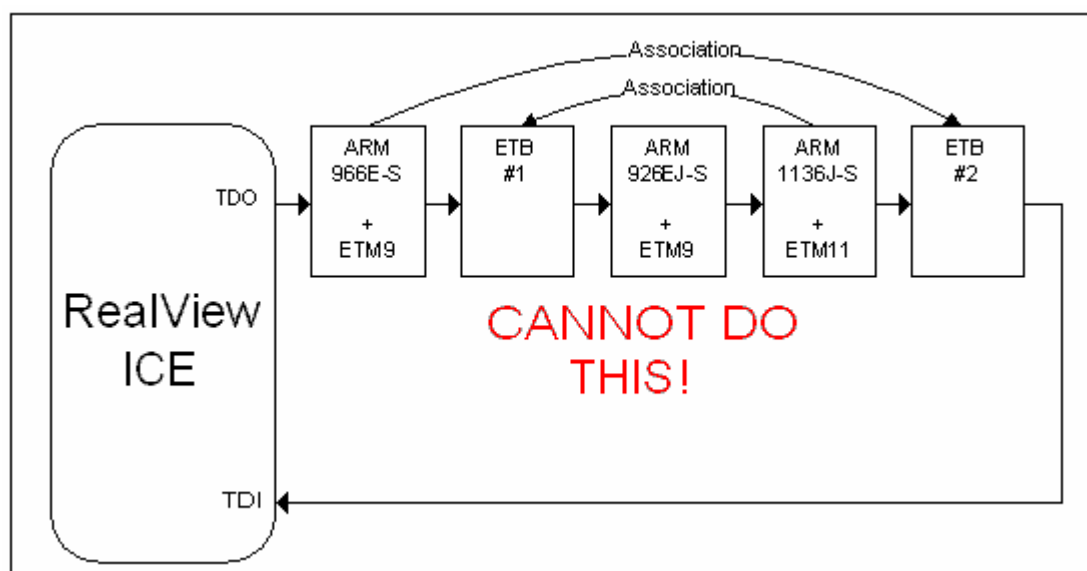


Figure 9: Impossible association of two ETB to two out of three cores