ARM CoreSight ™ DAP Lite (TM940) **Errata Notice**

This document contains all errata known at the date of issue for revision r1p2 of CoreSight DAP Lite

Proprietary notice

Document confidentiality status

This document is Non Confidential.

Web address

http://www.arm.com/

Feedback on the product

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

- The product name
- A concise explanation of your comments.

Feedback on this document

If you have any comments on about this document, please send email to mailto:errata@arm.com giving:

- The document title
- The documents number
- The page number(s) to which your comments refer
- A concise explanation of your comments

General suggestion for additions and improvements are also welcome.

Contents

Introduction

Scope

This document describes errata categorised by level of severity. Each description includes:

- the current status of the defect
- where the implementation deviates from the specification and the conditions under which erroneous behavior occurs
- the implications of the erratum with respect to typical applications
- the application and limitations of a 'work-around' where possible

Component Revisions

This document reports the errata for all revisions of CoreSight DAP Lite. Each release of DAP Lite contains several CoreSight sub-components, the revisions of each sub-component for each specific DAP Lite revision are documented in Table1 below.

Component	CoreSight DAP Lite			
	r0p0	r1p0	r1p1	r1p2
SWJTAG Dual Mode Debug Port	n/a	r0p1	r0p1	r0p2
JTAG Debug Port	r0p1	n/a	n/a	n/a
APB Access Port	r0p0	r0p1	r0p1	r0p1
APBMUX for DAP	r0p0	r0p0	r0p0	r0p1

Table 1 CoreSight DAP Lite component revisions.

Categorisation of Errata

Errata recorded in this document are split into three levels of severity:

Category 1 Behavior that is impossible to work around and that severely restricts the use of the product in all, or the majority of applications, rendering the device unusable.

Category 2 Behavior that contravenes the specified behavior and that might limit or severely impair the intended use of specified features, but does not render the product unusable in all or the majority of applications.

Category 3 Behavior that was not the originally intended behavior but should not cause any problems in applications.

Change Control

15 May 2008: Changes in Document v5

Page	Status	ID	Cat	Summary
11	New	507413	Cat 3	[SW JTAG Dual-mode Debug Port] Unsynchronized clock domain crossing in SWJ-DP
11	New	529172	Cat 2	[APBMUX for DAP] Unsafe clock domain crossing in DAP APB Multiplexor

02 Nov 2007: Changes in Document v4

No new errata in this revision (revision updates only).

18 Dec 2006: Changes in Document v3

Page	Status	ID	Cat	Summary
10	Updated	393293	Cat 2	[JTAG Debug Port] DAPCLK side synchronizers do not use DAPCLKEN
9	Updated	374331	Cat 2	[APB Access Port] Two signals differentiated only by capitalisation

11 Oct 2006: Changes in Document v2

Page	Status	ID	Cat	Summary
10	New	393293	Cat 2	[JTAG Debug Port] DAPCLK side synchronizers do not use DAPCLKEN
9	New	374331	Cat 2	[APB Access Port] Two signals differentiated only by capitalisation

06 Jan 2006: Changes in Document v1

No errata in this revision.

Errata Summary Table

The errata associated with this product affect product versions as below.

A cell shown thus **X** indicates that the defect affects the revision shown at the top of that column.

ID	Cat	Summary of Erratum	r0p0	r1p0	r1p1	r1p2
374331	Cat 2	[APB Access Port] Two signals differentiated only by capitalisation	X			
393293	Cat 2	[JTAG Debug Port] DAPCLK side synchronizers do not use DAPCLKEN	X	n/a	n/a	n/a
507413	Cat 3	[SW JTAG Dual-mode Debug Port] Unsynchronized clock domain crossing in SWJ-DP		X	X	
529172	Cat 2	[APBMUX for DAP] Unsafe clock domain crossing in DAP APB Multiplexor	X	X	X	

Note : The JTAG Debug Port component (TM904-MN-22100) is not part of the r1p0 or later releases, it has been replaced with the SW JTAG Dual-mode Debug Port component (TM945-MN-22100).

Errata - Category 1

There are no Errata in this Category

Errata - Category 2

374331: [APB Access Port] Two signals differentiated only by capitalisation

Status

Affects: product APB Access Port, CoreSight DAP Lite.

Fault status: Cat 2, Present in: r0p0, Fixed in r1p0.

Description

In the file DAPApbApDapIf.v two signals have the same name with different capitalisation. The code syntax is legal in Verilog yet it may lead to problems with synthesis and linting tools.

Input signal:

```
input          DEVICEEN;       // Device enable
```

Internal Register:

```
reg            DeviceEn;
```

Implications

Synthesis results may result in incorrectly functioning code as a result of the two signals being different only in capitalisation. This will not occur if the tool being used is compliant with the Verilog Standard.

Equivalence checking scripts supplied with the deliverables will fail where functional differences occur.

Workaround

None

393293: [JTAG Debug Port] DAPCLK side synchronizers do not use DAPCLKEN

Status

Affects: product JTAG Debug Port, CoreSight DAP Lite.

Fault status: Cat 2, Present in: r0p0,r1p0,r1p1,r1p2, Open [Fixed in r0p3 of JTAG Debug Port].

Description

None of the JTAG Debug Port's synchronizers on the **DAPCLK** side use **DAPCLKEN** to validate the clock signal. The problem that can be caused by this omission is only likely to be observed in a timing annotated simulation where it can cause timing violations if **DAPCLK** is faster than the synthesis constraint.

Conditions

The **DAPCLK** frequency is faster than its synthesis constraint.

Implications

As **DAPCLKEN** is not factored into the **DAPCLK** synchronisers, this will result in sampling the asynchronous signal at the **DAPCLK** rate and not at the **DAPCLKEN** rate, which the next register in the path will be operating at. This can result in setup and hold time violations for the registers working at **DAPCLKEN** which sample signals from the synchronisers and timing goals not being met with that configuration.

If synthesis has been constrained to the actual target speed and the **DAPCLK** frequency is not higher than this speed, there will not be any problems.

If the synthesis has been under constrained or **DAPCLK** is faster than the target speed and then gated with **DAPCLKEN** to generate the constraint clock, then timing violations will occur. The reason for this is that the synchronizers will toggle at the **DAPCLK** full speed, whereas the rest of the logic will follow the **DAPCLK && DAPCLKEN** signal; in this case, a change in the synchronizer input could be propagated to its output too close to an enabled **DAPCLK && DAPCLKEN** edge, thus violating timing.

Workaround

System implementers can either:

- When implementing JTAG-DP, make use of an external clock enable signal if the **DAPCLK** frequency is higher than the maximum operating frequency at synthesis for the block. Tie the **DAPCLKEN** signal high and connect the externally gated clock to the **DAPCLK** input of JTAG-DP

or

- Synthesize the block for the target **DAPCLK** frequency and not **DAPCLK && DAPCLKEN** frequency (i.e. ignore the gating effect of **DAPCLKEN** in the final design).

529172: [APBMUX for DAP] Unsafe clock domain crossing in DAP APB Multiplexor

Status

Affects: product APBMUX for DAP, CoreSight DAP Lite.

Fault status: Cat 2, Present in: r0p0,r1p0,r1p1, Fixed in r1p2.

Description

The Debug Access Port (DAP) APB Multiplexor drives the debug APB bus, multiplexing between debugger-generated accesses from the APB-AP and system-generated accesses from the APB slave port. The APB slave port is typically driven by one or more processors in the system. The APB slave port clock (PCLKSYS) can be asynchronous to the Debug APB clock (PCLKDBG).

If PCLKSYS is asynchronous to PCLKDBG, protocol violations might occur on the debug APB. PENABLEDBG and PSELDBG might remain high after PREADYDBG has gone high for a cycle to signify the end of a transaction, as if PREADYDBG had been ignored. This might result in Unpredictable behavior, including lock-up of the debug APB bus.

Conditions

- PCLKSYS is asynchronous to PCLKDBG.
- A system-generated accesses on the APB slave port occurs.

Implications

On-chip APB bus masters, such as processors, might cause unpredictable APB behavior if they attempt to access debug components in systems where the debug APB is asynchronous to other system buses.

Workaround

This is a workaround for system implementors.

System implementors should make PCLKSYS and PCLKDBG operate synchronously.

Errata - Category 3

507413: [SW JTAG Dual-mode Debug Port] Unsynchronized clock domain crossing in SWJ-DP

Status

Affects: product (NTBSS) SW JTAG Dual-mode Debug Port, CoreSight DAP Lite.

Fault status: Cat 3, Present in: r1p0,r1p1, Fixed in r1p2.

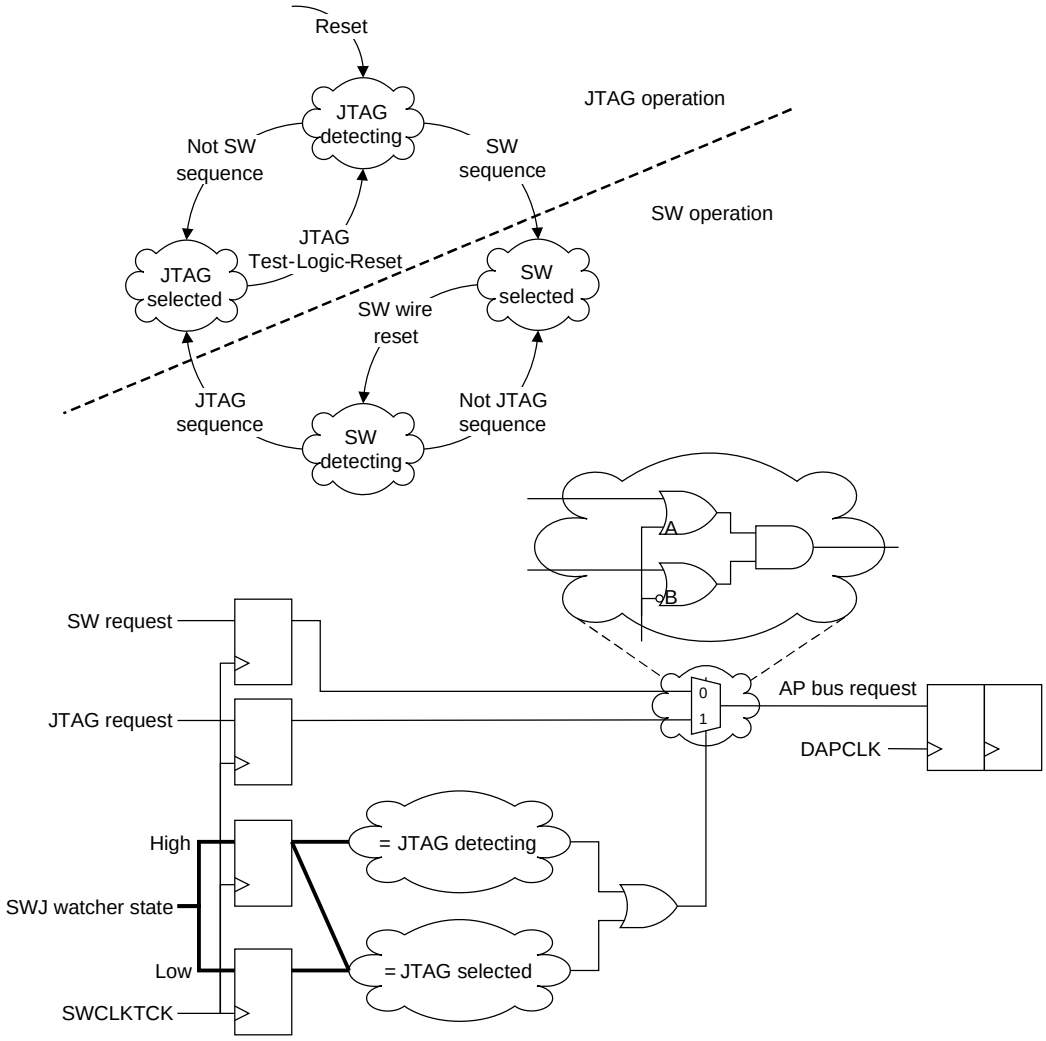
Description

A combinatorial path exists in control signals crossing the asynchronous clock boundary between SWCLKTCK and DAPCLK. This logic might be implemented in a manner which is unsafe when crossing between asynchronous clock domains.

When switching between JTAG and Serial Wire modes of operation, or immediately following a JTAG reset operation, a single unexpected transaction might be performed by the SWJ-DP.

This erratum is based upon a theoretically possible implementation of the design, it is very unlikely that the conditions necessary for this erratum will be present in an implementation. If the implementation does meet the conditions necessary for this erratum to occur, it is unlikely to cause unexpected system behaviour.

The diagram below illustrates the conditions necessary for this erratum.



The SWJ-DP unit implements a watcher module which drives a state machine that watches for the sequence to switch from JTAG to Serial Wire (SW) operation and back. The state machine starts in JTAG operation and searches for a SW selection sequence. If the sequence is met then it switches to SW operation, otherwise it switches to JTAG operation as soon as the sequence is broken. From JTAG operation it watches for the SW selection sequence again if a JTAG reset command is issued by transitioning into the JTAG Test-Logic-Reset state. Similarly, once in SW operation, it starts watching for the JTAG selection sequence if a SW wire reset is issued, and either stays in SW operation or switches to JTAG operation depending upon whether the sequence is met.

If the watcher switches from JTAG operation to SW operation or back again, it can cause a glitch on the AP bus request signal, causing an unexpected access to a DAP Access Port (AP), such as the APB-AP. When the mode of operation switches, both the SW request and JTAG request signals will be Low. However, it is possible for the multiplexer which switches between these two signals to cause a glitch on its output if it is implemented as an AND of ORs structure, with a significant wire delay between points A and B of the select signal shown in the diagram above. The presence of such a delay might cause both OR gates to be asserted simultaneously, causing a glitch on the output of the AND gate.

For this erratum to occur, the glitch on the AP bus request signal must be significantly long for a logic '1' to be sampled in the following register. It is highly unlikely that the skew between points A and B will be sufficient to generate a large enough glitch for this to occur.

Furthermore, it theoretically possible for such a glitch to be generated when switching between the JTAG detecting state and the JTAG selected state because of the method used to compare against the SWJ watcher state. A glitch can be caused on the following multiplexer, which can in turn cause a glitch on the APB bus request signal as described above. This case is only possible if the implementation includes a significant skew between the data arrival times of the two inputs to the OR gate driving the multiplexer.

Conditions

1. The logic has been implemented in a manner which is capable of causing this erratum, as described above, and,
2. SWCLKTCK is asynchronous to DAPCLK (this is the usual mode of operation), and, Either:
    1. A switching sequence is used to change between JTAG and Serial Wire modes of operation.
    2. The JTAG state machine enters the Test-Logic-Reset state and the implementation meets additional conditions as described above.

Implications

This erratum is unlikely to have any implications because:

- The implementation is unlikely to meet the conditions necessary to make this erratum possible.
- The switching sequence and JTAG Test Logic Reset are typically only issued when a debugger initially connects to the SWJ-DP, when the DAP registers are in their reset state. If an unexpected transaction is performed when the DAP registers are in their reset state then it will perform a safe read and have no effect in all current DAP implementations. If the DAP registers are not in their reset state then very few operations will have an effect outside of the DAP; any AP registers that are affected are likely to be subsequently overwritten by the debugger.

If all the necessary conditions are met unpredictable behaviour might occur in any AP. This might result in unexpected transactions over the APB or AHB master interfaces of the DAP or unexpected operations performed on the JTAG interfaces of the JTAG-AP. In addition, any connected JTAG device might be placed into or removed from the reset state because the nCSTRST output from the JTAG-AP might be asserted or de-asserted.

Workaround

No workaround is necessary in the majority of cases, but the following workarounds can be used if necessary.

This is a workaround for tools vendors.

- When tools connect to the SWJ-DP that the state of any control registers in any APs is assumed to be unknown and must be explicitly programmed with known values before using the AP.

This is a workaround for system implementors.

- When implementing the SWJ-DP, some of the logic between SWCLKTCK and DAPCLK should be implemented using glitch-free logic to ensure that the implementation does not meet the conditions described above.

Errata - Documentation

There are no Errata in this Category