ARM v7-M Architecture Application Level Reference Manual

RevB errata list

Architecture Group

Document number:	**PRD03-GENC-009270 1.0**
Date of Issue:	30 September 2008
Author:	
Authorised by:	

Abstract

This is the errata list for the RevB version of the ARM Architecture Application level Reference Manual. This applies to document ARM DDI 0405B.

Keywords

Distribution list

Name	Function	Name	Function

Contents

1 ABOUT THIS DOCUMENT

1.1 Change control

1.1.1 Change history

Change history is managed in Domino.doc

1.2 References

This document refers to the following documents.

Ref	Doc No	Author(s)	Title
	ARM DDI 0405B		ARMv7-M Architecture Application Level Reference Manual

1.3 Terms and abbreviations

This document uses the following terms and abbreviations.

Term	Meaning

2 SCOPE

This document holds a list of issues and/or minor errata relating to the ARMv7-M Architecture Application Level Reference Manual, RevB.

3 ARM DDI 0405C ADDITIONS

3.1 Section B1.2.3

Additional information on power management and SEV, WFE and WFI instructions added to the end of this section.

4 ARM DDI 0405B ERRATA DETAILS

4.1 Section A6.7.14, BFI

Update the assembler syntax <lsb> and <width> definitions with information on the valid range of values.

4.2 Sections A6.7.{47,56,60} LDR{B,H,SB} (register)

To the descriptions under the Assembler syntax heading, add:

<Rt> Specifies the destination register.

4.3 Sections A6.7.{55,59,63} LDR{H,SB,SH} (literal)

To the descriptions under the Assembler syntax heading, change: <Rd> to <Rt>

4.4 Section A6.7.50, LDRD (literal)

Operational pseudocode amended to the following:

```
if ConditionPassed() then
   EncodingSpecificOperations();
   if PC<1:0> != '00' then UNPREDICTABLE;
   address = if add then (PC + imm32) else (PC - imm32);
   R[t] = MemA[address,4];
   R[t2] = MemA[address+4,4];
```

The architected exceptions are MemFault and BusFault (UsageFault removed from the list).

NOTE: for the ARMv7 A and R profiles, the PC is word-aligned by the instruction and does not have the alignment constraint with respect to the address of the instruction itself.

4.5 Section A6.7.82 MSR (register) instruction

In the instruction encoding diagram:

hw1[4] is shown as '0' when the bit should be '(0)'.

hw2[11:8] are shown as '1000' when they should be '(1)(0) (0) (0)'.

4.6 Section A6.7.92 PLD (immediate, literal)

See clarification in section 5.5.

4.7 Section A6.7.96 (now A6.7.97), POP instruction

The operation pseudocode does not include the SP writeback. The following is added to the end of the operation:

```
   SP = SP + 4*BitCount(registers);
```

4.8 Section A6.7.116 (now A6.7.117), STM instruction

The revA operation pseudocode did not cover the case of writeback and Rn in the register list where Rn is not the first register in the list. For this case, the resultant write to the memory location is UNKNOWN. The pseudocode change to revB erroneously removed the register list bit check that control updates.

The operation pseudocode should read:

```
if ConditionPassed() then
    EncodingSpecificOperations();
    address = R[n];

    for i = 0 to 14
       if registers<i> == '1' then
            if i == n && wback && i != LowestSetBit(registers) then
                MemA[address,4] = bits(32) UNKNOWN;    // encoding T1 only
            else
                MemA[address,4] = R[i];
            address = address + 4;

    if wback then R[n] = R[n] + 4*BitCount(registers);
```

4.9 Section B3.1.3 MSR (register) instruction

In the instruction encoding diagram:

hw1[4] is shown as '0' when the bit should be '(0)'.

hw2[11:8] are shown as '1000' when they should be '(1)(0) (0) (0)'.

5 ARM DDI 0405B CLARIFICATIONS

5.1 Table A5-17 and Section A5.3.6 title

With reference to LDRD and STRD instructions, replace:

Double => Dual

5.2 Table A5-22, op == 0010 entry

For the Rn == 1111 case, add a reference to a new subsection "Move register and immediate shifts" immediately following this table. The subsection inserts a new table A5-23.

5.3 Table A5-25 (now A5-26) title

The title is incorrect as the M profile does not support sum of absolute differences instructions.

The title is amended to "Multiply, and multiply accumulate operations".

5.4 Table A5-26 (now A5-27) title

The title is a copy of the title for Table A5-25 which is incorrect.

The title is amended to "Long multiply, long multiply accumulate and divide operations".

5.5 Section A6.7.33, DSB

Typographic error in the mnemonic associated with the T1 encoding diagram

DMB => DSB

5.6 Section A6.7.92, PLD (immediate, literal)

This instruction is split into two instructions in line with a similar revision made in the ARMv7-AR manual. The option of implementing a PLDW variant, introduced in ARMv7-A with the MP Extensions, is allowed in ARMv7-M.

- PLD, PLDW (immediate)
- PLD, PLDW (literal).

5.7 Section C1.1 - debug

To remove ambiguity on whether a core can hold its' main AMBA bus in reset while the core is reset, the statement indicating that system memory is always accessible is removed.

Change:

Debug is normally accessed via the Debug Access Port (DAP, see ***Error! Reference source not found.***). The DAP allows access to debug and system memory whether the processor is running, halted, or held in reset. When a core is halted, the core is in Debug state.

To:

Debug is normally accessed via the Debug Access Port (DAP, see ***Error! Reference source not found.***). The DAP allows access to debug resources when the processor is running, halted, or held in reset. When a core is halted, the core is in Debug state.

5.8 Appendix B, Table B-1

Amend the LSL <Rd>, <Rm>, #<imm> entry to indicate MOVS <Rd>, <Rm> if <imm> == 0, otherwise it is LSLS <Rd>, <Rm>, #<imm>