The ARM Instruction Set Architecture

Dean Vidafar (Z5205847)

UNSW SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Author Note

For Any Enquiries Contact

<Z5205847@student.unsw.edu.au>

Abstract

The ARM processor is a 32-bit Reduced Instruction Set Computer (RISC), a microprocessor that recognises a limited number of instructions. An advantage of RISC microprocessors is that they can execute instructions relatively quick since the instructions are elementary. RISC chips also require fewer transistors and other components, making both RND and manufacturing more cost-efficient. This architecture is significantly different from other CISC (Complex instruction set computer) microprocessors and is primarily designed for embedded systems. In this report, the ARM Instruction Set Architecture, Memory Models, Registers, Instruction Set, and Data Types will be discussed. (Shee, 2004)

# ARM7TDMI Core Diagram

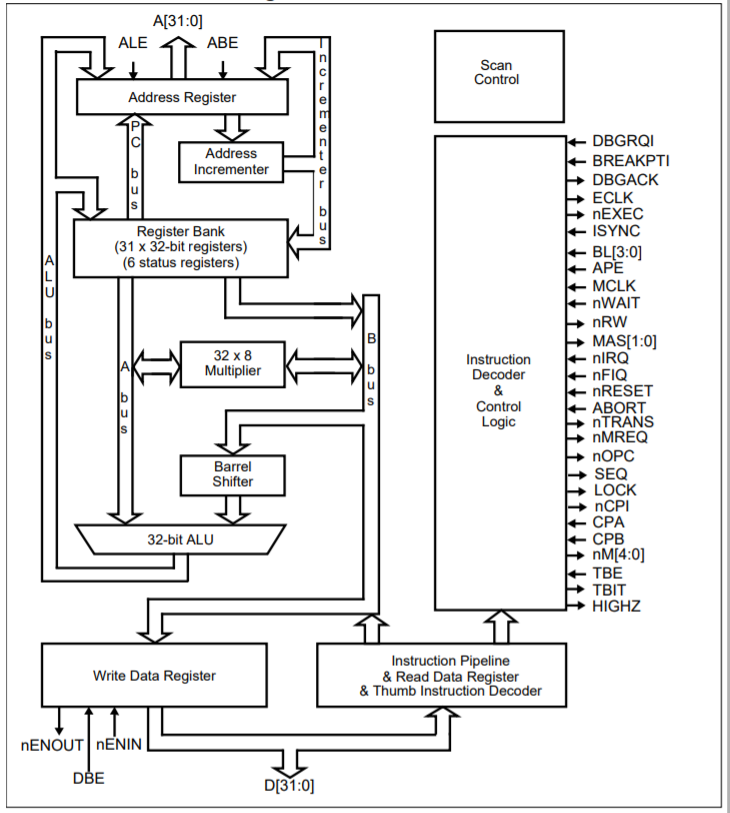


Figure 1: The ARM7DMI Core Diagram (Infocentre, ARM, 2004)

The ARM Instruction Set Architecture

The ARM microprocessor is based on the load/store architecture. ARM7TDMI offers no support for unaligned memory accesses. This microprocessor features two instruction sets, with both having mostly single clock-cycle execution. (McDermott, 2008). These instruction sets are:

* A 32-bit ARM instruction set
* A 16-bit Thumb instruction set

# The Thumb Instruction Set

The thumb instruction set is a subset of the most commonly used 32-bit ARM instructions. Each thumb instruction is 16 bits long, with a corresponding 32-bit ARM instruction that preforms the same operation. There is excellent interoperability between ARM and Thumb states due to the thumb instructions operating with the standard ARM register configuration. 16-bit thumb instructions are transparently decompressed to full 32-bit instructions on execution without performance loss. (Infocentre, ARM, 2004)

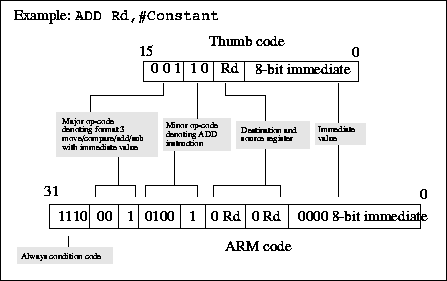


Figure 2: Example of how the ADD instruction is converted

from Thumb to ARM. (Bong-Ho, 1998)

## Why do we use thumb?

Thumb has all the advantages of using a 32-bit core such as:

* 32-bit registers
* 32-bit shifter
* 32-bit Arithmetic Logic Unit (ALU)
* 32-bit address space
* 32-bit memory transfer

Thumb offers a long branch range, powerful arithmetic operations and a large address space. Thumb Code is typically 65% of the size of ARM code and provides 160% of the performance of ARM code when running from a 16-bit memory system. Therefore, the ARM7TDMI core is ideally suited to embedded applications where there is restricted memory bandwidth and code density as well as footprint is integral.

# ARM Instruction Set

The ARM Instruction set offers a fixed instruction width of 32 bis to ease decoding and pipelining, at the cost of decreased code density.

# Arithmetic and the ARM7TDMI

The ARM7TDMI Microprocessor includes integer arithmetic operations for add, subtract and multiply, however, no support is provided for division. The microprocessor supports 32-bit by 32-bit multiplication with either a 32-bit result or 64-bit result. Arithmetic instructions can either set or preserve the conditions bits at the will of the programmer. The ARM7TDMI offers a range of addressing modes such as:

* Direct Addressing: ADD r3, r0, r1
* Immediate Addressing ADD r3, r0, #7
* Direct Addressing with A Shift or Rotate ADD r3, r0, r1, LSL#2

The Microprocessor supports some basic logical operators: AND, OR, XOR and BIC (Diba, 2014)

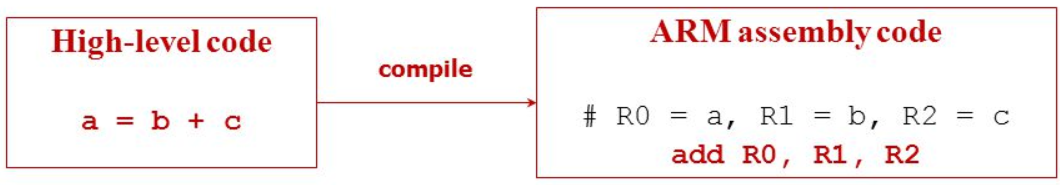


Figure 3: The conversion of High-Level C, to Low Level ARM. (Suh, 2015)

# Registers in the ARM7TDMI

The ARM7TDMI Microprocessor has a total of thirty-seven registers – thirty-one general-purpose 32-bit registers and six status registers - but these cannot all be seen at once. The processor state and operating mode dictate which registers are available to the programmer. In ARM state, 16 general registers and one or two status registers are visible at any one time. The ARM state register set contains 16 directly accessible registers: R0 to R15. All of these except R15 are general-purpose and may be used to hold either data or address values. In addition to these, there is a seventeenth register used to store status information.

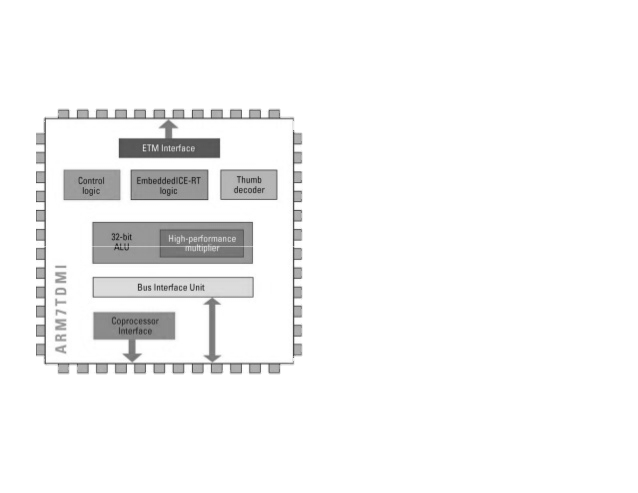


Figure 4:ARM7TDMI Architecture (ARM AAE, 2012)

References

ARM AAE, 2012. *Architecture.* Sydney, Techcon.

Bong-Ho, K., 1998. Introduction to Thumb. *Characteristics of ARM Processor,* pp. 2-10.

Diba, M., 2014. ARM Arithmetic and Bitweise Instructions. *Computer Organization and Systems Programming,* pp. 5-20.

Infocentre, ARM, 2004. *ARM7TDMI Technical Reference Manual.* [Online]   
Available at: http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0210c/CACBCAAE.html  
[Accessed 20 August 2018].

McDermott, M., 2008. The ARM Instruction Set Architecture. *The University of Texas at Austin,* pp. 2-2.

Shee, S. L., 2004. *The ARM Architecture,* Sydney: UNSW.

Footnotes

1[Add footnotes, if any, on their own page following references. For APA formatting requirements, it’s easy to just type your own footnote references and notes. To format a footnote reference, select the number and then, on the Home tab, in the Styles gallery, click Footnote Reference. The body of a footnote, such as this example, uses the Normal text style. (Note: If you delete this sample footnote, don’t forget to delete its in-text reference as well. That’s at the end of the sample Heading 2 paragraph on the first page of body content in this template.)]

Tables

Table 1

[Table Title]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Head | Column Head | Column Head | Column Head | Column Head |
| Row Head | 123 | 123 | 123 | 123 |
| Row Head | 456 | 456 | 456 | 456 |
| Row Head | 789 | 789 | 789 | 789 |
| Row Head | 123 | 123 | 123 | 123 |
| Row Head | 456 | 456 | 456 | 456 |
| Row Head | 789 | 789 | 789 | 789 |

Note: [Place all tables for your paper in a tables section, following references (and, if applicable, footnotes). Start a new page for each table, include a table number and table title for each, as shown on this page. All explanatory text appears in a table note that follows the table, such as this one. Use the Table/Figure style, available on the Home tab, in the Styles gallery, to get the spacing between table and note. Tables in APA format can use single or 1.5 line spacing. Include a heading for every row and column, even if the content seems obvious. A default table style has been setup for this template that fits APA guidelines. To insert a table, on the Insert tab, click Table.]

Figures title:

Figure 1. [Include all figures in their own section, following references (and footnotes and tables, if applicable). Include a numbered caption for each figure. Use the Table/Figure style for easy spacing between figure and caption.]

For more information about all elements of APA formatting, please consult the APA Style Manual, 6th Edition.