

1. Introduction to the ARM Architecture.

NXSOL-OJT-2024

Exported on 02/27/2024

Table of Contents

1	Introduction to the ARM Architecture.....	3
1.1	A1.1 About the ARM architecture	3
1.1.1	Key Characteristics	3
1.1.2	Enhancements	3
1.1.3	Coprocessor	3
1.1.4	A1.1.1 ARM Registers.....	3
1.1.5	A1.1.2 Exceptions	4
1.2	A1.2 ARM Instruction Set	4
1.2.1	Classification	4
1.2.2	Features.....	4
1.2.3	A1.2.1 Branch Instructions.....	4
1.2.4	A1.2.2 Data-Processing Instructions.....	5
1.2.5	A1.2.3 Status Register Transfer Instructions.....	5
1.2.6	A1.2.4 Load and Store Instructions	5
1.2.7	A1.2.5 Coprocessor Instructions	5
1.2.8	A1.2.6 Exception-Generating Instructions.....	5
1.3	A1.3 Thumb Instruction Set	5

1 Introduction to the ARM Architecture

1.1 A1.1 About the ARM architecture

1.1.1 Key Characteristics

ARM is a Reduced Instruction Set Computer (**RISC**) architecture, which includes:

- Large uniform register file.(These registers are uniform, meaning they can all be used interchangeably for a wide range of instructions. This uniformity simplifies the instruction set and improves the speed of computation.)
- Load/store architecture.(Data-processing operations (like addition, subtraction, etc.) are performed only on data in the registers, not directly on data in memory.)
- Simple addressing modes.(Addressing modes refer to the ways in which instructions specify the location of data. The smaller number of simple addressing modes makes it easier to design the hardware to decode and execute instructions, which can lead to faster processing speeds.)
- Uniform and fixed-length instruction fields, to simplify instruction decode.(to facilitate pipeline)

1.1.2 Enhancements

- Control over the Arithmetic Logic Unit (ALU) and shifter.
- Auto-increment and auto-decrement addressing modes.
- Load and Store Multiple instructions.
- Conditional execution of almost all instructions.

These enhancements to a basic RISC architecture allow ARM processors to achieve a good balance of high performance, small code size, low power consumption, and small silicon area.

1.1.3 Coprocessor

Coprocessors can be attached to the ARM processor. A coprocessor extends the processing features of a core by extending the instruction set or by providing configuration registers.

1.1.4 A1.1.1 ARM Registers

- Total of 37 registers
 - 31 general-purpose 32-bit registers, including a program counter. R0 ~ R15 visible at any time.
 - 6 status registers

1.1.5 A1.1.2 Exceptions

- **Types of Exceptions:**
 - reset
 - attempted execution of an Undefined instruction
 - software interrupt (SWI) instructions, can be used to make a call to an operating system
 - Prefetch Abort, an instruction fetch memory abort
 - Data Abort, a data access memory abort
 - IRQ, Interrupt request
 - FIQ, Fast interrupt request

1.2 A1.2 ARM Instruction Set

1.2.1 Classification

1. **Branch Instructions**
2. **Data-Processing Instructions**
3. **Status Register Transfer Instructions**
4. **Load and Store Instructions**
5. **Coprocessor Instructions**
6. **Exception-Generating Instructions**

1.2.2 Features

- Conditional execution based on condition code flags.
- Most data-processing instructions can update the four condition code flags in the CPSR (Negative, Zero, Carry and oVerflow) according to their result.

1.2.3 A1.2.1 Branch Instructions

Used for control flow alteration by writing the PC(Program Counter)

- **Standard Branch Instruction:** Uses a 24-bit signed word offset for forward and backward branches.
- **Branch and Link (BL):** Preserves return address in the Link Register.
- **Instruction Set Switching:** Allows switching between ARM and Thumb instruction sets.

1.2.4 A1.2.2 Data-Processing Instructions

- **Arithmetic/Logic Instructions:** Perform operations on two source operands and write the result.
- **Comparison Instructions:** Perform operations without writing the result, just updating condition flags.
- **SIMD Instructions:** Single Instruction, Multiple Data) Treat operands as parallel 16-bit or 8-bit numbers (ARMv6).
- **Multiply Instructions:** Various classes for different operations.

1.2.5 A1.2.3 Status Register Transfer Instructions

Transfer contents between CPSR/SPSR and general-purpose registers.

1.2.6 A1.2.4 Load and Store Instructions

Move data between memory and registers.

- **Types:** Regular, Multiple registers, Exclusive.
- **Addressing Modes:** Offset, Pre-indexed, Post-indexed.
- **Support for Unaligned Accesses:** Introduced in ARMv6.

1.2.7 A1.2.5 Coprocessor Instructions

- **Types:** Data-processing, Data transfer, Register transfer.

1.2.8 A1.2.6 Exception-Generating Instructions

- **Software Interrupt (SWI):** Used for OS-defined services (software interrupt).
- **Software Breakpoint (BKPT):** Causes an abort exception for debugging.

1.3 A1.3 Thumb Instruction Set

A subset of ARM instruction set with 16-bit encoding for instructions. It offers code density improvements. (Code density refers to the amount of instructions that can be packed into a given amount of memory space.)