University of North Alabama

Computer Architecture and Orginization

Cortex A15 Processor

Author:

Jeffrey Allen

Professor:

Dr. Patricia Roden

October 14, 2014

Contents

1 History/Background of ARM Cortex-A15 **4**

2 Memory Specifications **4**

3 Instruction Sets **5**

3.1 RISC Vs. CISC 5

3.2 Main ISA: ARMv7-A 6

3.3 Extentions to ARMv7-A Instructions 6

3.3.1 Thumb & ThumbEE 7

3.3.2 Advanced SIMD 7

3.3.3 VFP 7

4 Instruction Format **7**

5 Registers **8**

6 Data Types **8**

7 Addressing Modes **9**

8 I/O **10**

9 Unusual Features **10**

9.1 Limited Registers 10

9.2 big.LITTLE with the A15 & A9 10

10 Uses/Applications **11**

10.1 Mobile Computing 11

11 Contributions to the computer architecture landscape **12**

11.1 Low Energy Computing 12

1 History/Background of ARM Cortex-A15

In 1985, the Acorn Computer Group birthed the ARM architecture in the United Kingdom. This was only the beginning to the quick pace ARM began developing and implementing updates in its architecture. Two short years afterwards, Acorn released its first RISC processor that was optimized for low-cost personal computers. Later, in 1990, the ARM name which originally stood for Acorn RISC Machine, developed its Advanced RISC Machine, which defined its 32-bit RISC architecture.

Fast foward to the year 2014. The A15 is fourth generation in the family, following the A5, A7, and A9 processors.

2 Memory Specifications

In reviewing the ARM Architecture Reference Manual ARMv7-A and ARMv7-R edition, the list of following data types are supported in memory:

Byte 8́ bits

Halfword 1́6 bits

Word 3́2 bits

Doubleword 6́4 bits

Load and store operations are able to transfer bytes, halfwords, or words to and from memory. The loading of bytes or halfwords zero-extend (pad with zeroes) or sign-extend (increasing number of bits of a binary number while preserving the number's sign and value) the data as it is loaded, or specified by the load instruction.

Instructions are word-algined.

3 Instruction Sets

I'll begin this section with explaining that an interface is an intermediary between two entities. Humans have utilized this concept in order to build rockets which have reached mars, create heart rate monitors, and even predict weather. A processor which is only designed to fetch, decode, and execute various instructions must first be able to understand what its supposed to do. Since humans and computers are worlds apart when it comes to communicating with each other, the Instruction Set Architecture interface was created. This interface saves humans from learning the exact strings of 0's and 1's the processor understands in order to carry out an instruction. This is the agreed-upon interface designed for a machine essentially allows software to able to communicate with the hardware that executes it.

3.1 RISC Vs. CISC

Two types of processor design philosophies include Reduced Instruction Set Computers (RISC) and Complex Instruction Set Computers (CISC). CISC systems can be described through an example. Subtracting two integers is considered "simple". Then there is a type of instruction which can be found in a bubble sort. In array A, copying one element

α located at A_x and "bubbling" it up in order to swap it with another element located at A_y found to be the α's correct position. CISC systems are based off of these complex instructions that performed multiple operations in a single instruction. These instructions which were used primarily in the 1950's through the 1970's and early 1980's because of the limitations of memory and cost (16KB of memory $\approx$ \$500) which architecture designers were able to optimize all instructions for a specific task.

RISC systems were designed with simplicity in mind, keeping all instructions small in order to execute the more efficiently. A notable difference between the two philosophies is that RISC assumes that required operands are not in memory, but directly in the processor's internal registers.

3.2 Main ISA: ARMv7-A

At the end of the day, it is all about communication. Moore's Law, stating that the density of transistors on an integrated circuit is doubling ever 18 months, is one of the main reasons why communication between humans and computers has changed, and continues to change so rapidly. This concept applied to the range of years between 1960 to 2014 describes a fair amount exponential increase in transitors, or methods of communication. As Linda Null and Julia Lobor point out though, RISC in today's society something of a misnomer. It can be said that almost all recently created instruction sets are a mixture of RISC's and CISC's. The main instruction set supported by the Cortex-A15 processor is the 32-bit ARMv7-A.

3.3 Extentions to ARMv7-A Instructions

The A15 is designed with ability to decode different instruction sets. Depending on what "mode" of operation of execution the instruction . This mode is controlled by a simple switch of the designated T bit J bit in the Current Program Status Register (CSPR). With the A15 having the ability to support multiple ISA's, this allows for is the Large Physical Address Extension (LPAE) architecture to enable virtual environments.

Why have support for virtual environments though? As stated before, the Cortex-A series is a family of processors. Being that utilizes multiple instruction sets, or extensions of them. These includes the ARMv7-A, Thumb, ThumbEE, NEON, and VFP instruction sets. ARM's A15 reference manual indicates on various occasions when to refer to a repective to describe it's own features.

3.3.1 Thumb & ThumbEE

3.3.2 Advanced SIMD

Advanced SIMD extension is a media and signal processing architecture that adds instructions targeted primarily at audio, video, 3-D graphics, image, and speech processing.

3.3.3 VFP

VFP extension performs single-precision and double-precision floating-point operations.

4 Instruction Format

Since the ARMv7-A architecture is based off of the RISC design, all instructions are the same size. However, since the Cortex A15 extends this base ARMv7-A architecture, 64-bit instructions also exist. This is only in virtual states of processing though. In the many layers of abstraction that exist in a computer, the endianess of an architecture must be carefully observed. The endianess of of an architecture refer to the "byte order" of the computer. This can also be explained as the method of which a computer stores multiple-byte data elements. For example, the 4-bit string made up of 2 2-bit data elements 0001, can be read by a computer in little-endian as 4_{10}. Alternatively, read as big-endian it would be 1_{10} The ARMv7-A architecture exclusivley exhibits little-endian practices. The A15 however, has an interesting but dangerous characteristic which gives it the ability to configure itself between little-endian or big-endian through the use of ARM and Thumb instruction $SETEND\ BE$ which translates to 0, or $SETEN\ LE$ which translates to 1.

5 Registers

As explained before, there are multiple

The A15 processor has a total of 15 registers, which are "grouped" in different categories. As explained in

6 Data Types

The following data types use these 32 bits to represent:

- 32-bit pointers
- Unsigned or signed 32-bit integers
- Unsigned 16-bit or 8-bit integers, held in zero-extended form
- Signed 16-bit or 8-bit integers, held in sign-extended form
- Two 16-bit integers packed into a register

- Four 8-bit integers packed into a register
- Unsigned or signed 64-bit integers held in two registers

Unsigned according to this processor's specification manual is described as a data value representing a non-negative integer in the range 0 to 2^{N-1}.

7 Addressing Modes

As described before, in only instructions that access memory are the load and store instructions.

8 I/O

9 Unusual Features

9.1 Limited Registers

One of the characteristics that make the ARM architecture unique is that only supports 16 general purpose registers. Compared to other architectures, this amount of registers can be called limited. All of the other major architectures, including the MIPS, SPARC, and the PowerPC architectures all support 32 registers.

To compensate for this limitation of registers though, virtualized environment instructions have been designed and implemented for this processor.

9.2 big.LITTLE with the A15 & A9

10 Uses/Applications

10.1 Mobile Computing

Ph.D. Steve Furber, one of the principal designers of the BBC Microcomputer System and ARM microprocessors when the "A" in ARM stood for "Acorn", had this to say when interviewed by Jason Fitzpatrick. "The ARM architecture is the most widely used 32-bit RISC architecture and the ARM processor's power efficiencyperforming the same amount of work as other 32-bit processors while consuming one-tenth the amount of electricityhas resulted in the widespread dominant use of the ARM processor in mobile devices and embedded systems." This whole research paper is based off of fascination related to ubiquitous computing. So when I decided to conduct research on what was behind the popular Samsung's Galaxy S4 successful mobile phone design, it was not a coincidence to find out it was ARM behind the processor design.

11 Contributions to the computer architecture landscape

11.1 Low Energy Computing