ISSN: 0975-8585

Research Journal of Pharmaceutical, Biological and Chemical Sciences

Survey on Advanced RISC Machine CPU.

B Ravali* and N Mathan.

Dept. of ECE, Sathyabama University, Chennai, Tamil Nadu, India.

ABSTRACT

The processing speed in any devices depends on processors. As today, the smart phones, tablets, and even some laptops are becoming more popular because of their high performance, energy efficiency and console gaming. Processors are integral parts of these devices and are responsible for the efficiency of these parameters. Processors play crucial part in SOC and the reach of this high performance is obtained by ARM processors.

Keywords: ARM processor, instruction set, pipelining, memory, Cortex, clock cycle.

**Corresponding author*

ISSN: 0975-8585

INTRODUCTION

ARMs previously Acorn RISC Machine, also known as Advanced RISC Machine is a family of reduced instruction set computing (RISC), architectures for computer processor, configured for various environments, developed by British company named ARM Holdings.ARM processors belongs to a family of 32-bit microprocessors developed by Advanced RISC Machines, Ltd. in the 1980s. Today ARM processor has 64-bit influence a wide variety of electronic devices including mobile phones, tablets, multimedia players and more.

It's not just simple to work out the ARM family [1]. ARM cores (ARM1, ARM2) don't have any matching with architecture numbers (ARMv1,ARMv2).

ARM's architecture is compatible with all following major platform operating systems Symbian OS, Palm OS, Windows CE, and Linux.

ARM SERIES

ARM 1:

Architecture: ARMv1

The first Acorn RISC Machine CPU known as ARM1.It was designed by R&D team at Acorn computers Ltd between 1983 and 1985[2]. The ARM is a unique architecture, but carries many standard RISC traits. Most importantly, it has a fixed instruction length of 32 bits, and it is load/store architecture: the only instructions which operate on memory are for loading values to registers, and stores the register values in memory. All registers are general-purpose, and the instruction set is orthogonal.
ARM1 was used, on a co-processor board in small numbers, for the BBC micro family[3].

ARM 2:

Architecture: ARMv2

The ARM2, later in 1985 became the first commercially available RISC processor. The ARM1 versions were made in limited run, but early enough to claim the prize of the first commercially available RISC processor. The ARM2 was designed to support chips like IOC, MEMC, VIDC [4].

One of the key problem was a lack of hardware multiply support. This directs the software through routine multiplication, using shifts and add, which was "horribly slow". This was fixed in the ARM2, adding two instructions: MUL (Multiply) and MLA (Multiply with accumulate). These allowed the ARM2 to be realistically used for mathematical calculation, and very simple digital signal processing, in particular generating synthesised sound.

This increased performance of the very common FIQ by lowering memory access to stack registers. The ARM2 was manufactured at a slightly smaller fabrication size than the ARM1, 2.5µm as opposed to 3.5µm, had over 25000 transistors and clocked at 8MHz.

ARM 3:

Architecture: ARMv2a

Due to the lack of floating point hardware in ARM processors, Acorn has decided to add hardware co-processor support to ARM2 and later date proposed to ship an optional floating point accelerator. The first hardware FP came as an option with the ARM3 powered A5000.

The ARM3 was the first ARM processor with integrated memory cache. ARM elected to use simple 64 way set-associative cache with random write through replacement method, and 128-bit cache lines. Alterations were made to the co-processor interface to support this, and the cache system was designated co-processor zero.

The SWP (swap) instruction was introduced, allowing the user to atomically read and write to memory in single instruction.

ARM 6:

Architecture: ARMv3

Technical Changes:

The most important change in the version3, ARM6 is first processor to support 32-bit address[5]. This required a large redesign of the internal operation of the CPU, previously, the processor status and program counter shared the 32-bit R15 register, clearly not sustainable when the program counter need to be 32-bits wide, itself. The transfer to new addressing system needed four 32-bit versions of the standard ARM processor modes, USR32, IRQ32, FIQ32, and SVC32.

ARM6 configured for 32-bit program and data space, but this family also supports 26-bit modes operation for the compatibility with the ARM processor which have 26-bit address space [5].

ARM 7:

Architecture: ARMv3

ARM7 is follow-up processor to the ARM6, was in its basic form functionally equivalent to its forebear, but due to fabrication variations had improved the factors, It used less power, and was able to operate with higher clocked speed.

Core Modifications:

The ARM7 processor is of Von Neumann–style architecture, both data and instructions use the same bus. By this the occurrence of reduction on performance of ARM takes place, but it is overcome by pipeline process [6].

Core has four optional letters after the series of the processor denote one or more of Thumb instruction support, debug support, enhanced multiplication, and in-circuit emulation.

Thumb encodes the 32-bit ARM instructions into a 16-bit space [7]. Thumb has higher performance than ARM on a processor with a 16-bit data bus, but has less performance than ARM on a 32-bit data bus. Thumb implementation code reduces 30% of overall memory usage than ARM.

ARM 8:

Architecture: ARMv4

ARM 8 was the 1996 follow on the engross success of ARM7. The goal was to develop next core with double the performance of the ARM7, by retaining the low power usage and simplicity of implementation. ARM8 also follow the Von Neumann- style architecture.

ARM8 was later designed with extension of ARM7 by introducing halfword and signed halfword/bit support system mode. Thumb instruction (v4t) was also added.

The only way to significantly improve CPI is to modify the memory architecture such that multiple 32-bit access are possible in single cycle. Two way to proceed by a separate instruction and data memory, or double the bandwidth of memory.

To accomplish an increased clock rate, the ARM's 3-stage pipeline had to be modified accordingly by use of more "standard" RISC 5-stage pipeline, very similar to that of the MIPS processor. The five stages are:

instruction pre-fetch, decode/register read, ALU operation/shift, data memory access/ALU write, and register write-back [9].
One drawback of this pipeline process is that instruction scheduling becomes more vital. Loading a value to register from memory (LDR) and immediately using that register as a source operand causes a single clock cycle pipeline bubble. Re-ordering of instruction as to done to avoid data dependency.

ARM 9:

Architecture: ARMv5

ARM9 is improvement over the ARM8 was supportive to TDMI extensions, the feature of this is same 5-stage pipeline design, with a strong ARM-like Harvard memory architecture.

ARM9, allows simultaneous access to instruction and data memories. In the 5-stage ARM pipeline, this is critically useful. The stages which access memory, instruction **fetch** and **memory** read/write, are four stages apart.

The ARM9 cores are used for high performance applications that earlier could not be implemented at the same cost. ARM9 family cores were developed with a goal of double the performance of the ARM7 without change to the architecture.

Pipeline alteration:

The pipeline used in the ARM8 is almost identical to that used in the ARM9. Instantaneous instruction and data memory access are possible, it is no longer necessary to have a pre-fetch unit buffering sequences of instructions. Moreover, the branch prediction capability of the prefetch unit was not used in the ARM9.

ARM 10:

Architecture: ARMv5

ARM10's motto was again to increase the performance of its predecessor on the same fabrication, by allowing for further enhancement with smaller processes.

As the travel from ARM7 to ARM8, effective performance was enhanced using two methods. By varying the pipeline process to enable a huge clock speed, and enhancing the number of cycles taken to execute each instruction. The ARM9's pipeline was already optimal, and it's CPI is very low, so relatively complicated techniques were required to meet the target. A fine balance between core complexity and power consumption had to be hit for ARM's reputation as low power architecture to be upheld.

Pipeline optimization:

By the slowest pipeline stage, maximum clock speed can always be determined. To increase clock speed proceed with one common approach, favoured most eminently by Intel, is superscalar pipelining.

The ARM10 cores first approach was to simply optimize each stage as much as possible to generate better clock scaling than the ARM 9.

6 stage pipeline:

ARM10's core pipeline has six stages. Fetch Issue, Decode, Execute, Memory, and Write. The initial stage contains the static branch prediction unit, and the instruction fetcher. The preceding stage converts Thumb instructions into appropriate ARM counterpart. Next, parallel process has done with rest of the instruction decoding.

The fourth stage is the most complex, the address calculation unit determines the address of a branch or memory access, whereas the hardware multiplier generates partial products and arithmetic

operations are done by the ALU and barrel shifter. In the fifth stage the addition of partial products completes multiplication operation, or data memory is read. Lastly, the ALU or data memory stages write-back to the register file as appropriate.

ARM10 support floating point co-processor which is on-chip vector, i.e. VFP10.

ARM 11:

Architecture: ARMv6

The new version6 was introduced with core ARM11. It comprise of SIMD media instructions, multi-core support and new cache (memory) architecture. The implementation included a significantly improved instruction processing pipeline, compared to previousARM9 orARM10 core families, and used in smart-phones from Apple, and others.

The operation, power, speed, area, cost parameters and performance must be balanced to meet the requirements of application. ARMv6 offers better techniques to optimize the above constraints [10].

In developing ARM10 key focus and effort has been placed in five areas: Memory management, multiprocessing, multimedia support, data handling, exceptions and interrupts [10].

CORTEX-A SERIES:

Architecture: ARMv7-a

Cortex-A processor are introduced specially to execute complex functions and application required for customer specified devices like smart phones, and tablets.

Cortex-A processor has low power architecture which can with stand all day browsing connectivity, support quality gaming, technologies like NEON and sustenance for the widest mobile app system.

Cortex-A processor empower highly scalable keys to match performance requirements for power efficient package transfer, edge routers and server etc.

All Cortex-A processors has common supported architecture and feature set, with each processor based on either the ARMv7-A or ARMv8-A architecture and feature set. The ARMv8-A has a 64 bit execution state and it also support existing 32-bit. This enhances the compatibility and strengthens the 64-bit system [11].

Cortex-A5:

This achieves better performance, better power and energy efficiency than ARM11 and ARM 9. This is smallest and lowest power ARMv7-A core. Cortex-A5 is the most mature, configurable and delivers high range of features [12].

Cortex-A7:

Cortex-A7 was designed to enhance the performance from entry-level to mid-level smart phones, for low power embedded and customer applications [12].

Cortex-A9:

Cortex-A9 was designed with out-of-order, pipeline of8 to 11 stages, high efficiency and with dual issue superscalar.

All cortex-A17, A15, A9, A7, A5 comes under ARMv7 architecture. By the combination of cores cortex-A15, Cortex-A17 provides high performance. Cortex-A9 gives high efficiency, and cortex-A7 compact with cortex-A5 provides ultra-high efficiency [12].

Cortex-A53:

Architecture: ARMv8

The ARM Cortex-A53 processor is the best power efficient ARM-v8 processor proficient of supporting both 32-bit and 64-bit. The Cortex-A53 delivers 64-bit capability and significantly increases performance over Cortex-A7, it is cost sensitive application [13]. Cortex-A53 is smallest and lower power compared to Cortex-A9 core and conveys more performance. This means it can support devices with the compute power of today's high end smart phone in the lowest power and less area.

Cortex-A57:

The ARM Cortex-A57 processor is also support both 32 and 64-bit and provides high performance. To prolong the capabilities of mobile and computing applications including 64-bit applications such as high end computer, smart phones, tablet and server products.

By the architectural alignment of cortex-A72, cortex-A57 empowers high performance, cortex-A53 alone provides high efficiency, and cortex-A35 gives ultra-high efficiency [12].

The ARMv8 processor is designed to work on 64-bit code. The main purpose to design version8 is to compatible with OS which support only 64-bit processing.

At higher level, ARMv8-A describes both a 32-bit and 64-bit architecture respectively AArch32 and AArch64 [14].

AArch64 belongs to ARMv8-A 64-bit execution state, that uses 64-bit general purpose registers, stack pointer (SP), exception link registers (ELR), and a 64-bit program counter. It provides a single instruction set, A64 [15].

Related works on Existing

In paper [16] briefs about Terga k1 SOC contains a entirely compatible ARMv8 dual core Denver CPU which supports both AArch32 and AArch64.DenverCPU will extend the reach of clock speed up to 2.5GHz. To provide platform for mobile with high efficiency, high performance and gaming [16].The main role in this architecture is seven ways super-scalar unit which executes 7 instructions per single clock cycle.

In paper [17] speaks about the Potenza, the first generation ARMv8 processor. Potenza is an integrated design unit. It was designed to be scalable for different server configuration. This processor uses Mesh-on-chip with four wide superscalar micro-architecture.

In paper [18] describes about the sensor application using ARM processor. In this they developed and implemented temperature sensor which runs on Arm processor.

CONCLUSION

ARM processors are industry standard microprocessor, with low power and high performance. ARM cores provides full solutions supporting broad range of applications. ARM architecture enables high performance computing, visual computing.

REFERENCES

[1] www.slideshare.net/ARM_Based_Group/beginning-in-arm-architectures
[2] everything2.com/title/ARM1

[3] www.bbcbasic.co.uk/bbcbasic/birthday/
[4] www.heyrick.co.uk/armwiki/The_ARM_family#The _ARM1_.28ARMv1.29
[5] web.archive.org/web/20070809230809/http://www.arm.com/pdfs/Apps11vC.html
[6] www.slideshare.net/yayavaram/unitii-arm-architecture?from_action=save
[7] homepages.thm.de/~hg10013/Lehre/MMS/WS0304_SS04/Ioannis/PDF/arm.pdf
[8] www.slideshare.net/ARM_Based_Group/beginning-in-arm-architectures
[9] www.poppyfields.net/acorn/news/armpress/arm8view.shtml
[10] students.mimuw.edu.pl/~zbyszek/asm/arm/ARMv6_Architecture.pdf
[11] www.arm.com/products/processors/cortex-a/index.php
[12] community.arm.com/groups/processors/blog/2015/11/05/introducing-cortex-a35-arms-most-efficient-application-processor
[13] www.arm.com/products/processors/cortex-a/cortex-a53-processor.php
[14] www.arm.com/files/downloads/ARMv8_Architecture.pdf
[15] file:///F:/major%20pg/ARM%20Processors_%20ARMv8%20Architecture,%20The%20whys%20&a...%20_%20ARM%20Connected%20Community.html
[16] Denver: NVidia first 64-bit ARM processor Boggs, D.; Brown, G.; Tuck N. ; venkatraman, K. S. Micro, IEEE Year:2015, volume:35, Issue:2 pages: 46 55, IEEE Journals & Magazines
[17] A 3GHz 64b ARM v8 Processor in 40nm Bulk CMOS Technology, Alfred Yeung, Hamid Partovi, Qawi Harvard, Luca Ravezzi, John Ngai, Russ Homer, Matthew Ashcraft, Greg Favor(2014).
[18] A Bare Machine Sensor Application for an ARM Processor Alexander Peter, Ramesh K.Karne and Alexander L.Wijesinha Department of Computer& Information Sciences Towson, MD 21252 apeter9@students.towson.edu,(rkarne,awijesinha)@towson.ed (2013)