IJCSE International Journal of Computer Science and Engineering Open Access
Review Paper Volume-2, Issue-3 E-ISSN: 2347-2693

A Study on Architectures for Embedded Devices

Arun Radhakrishnan[1*] and T. Muralikrishna[2]

[1*]Department of ECE, Jimma University, Ethiopia, cudarun@gmail.com
[2] Department of IT, WolaitaSodo University, Ethiopia, murali2007tel@gmail.com

www.ijcseonline.org

Received: 2 March 2014 Revised: 12 March 2014 Accepted: 22 March 2014 Published: 30 March 2014

Abstract— Embedded designers can't afford "the blue screen of death." Desktop operating systems can get rebooted every so often, but embedded devices often have to run for years without a single reboot. Devices that are used in medical fields are one of the most appropriate examples in which reliability is more important. Also other embedded devices such as industrial-automation systems, security systems and automotive systems have to be more reliable. And the heart of the embedded system is the processor used in it. In recent days there is a big debate among the embedded device manufacturers in selecting the type of architectures which can be used in their embedded devices. The two most commonly used architectures in embedded devices are X86 and ARM. The decision to chose X86 or ARM needs to be investigated thoroughly based on their advantages and limitations. The selection is not only based on its performance of processors and costs but also on power consumption, the type of hardware and software components that it can support and the type of input and output devices supported. In this paper both the architectures are analyzed.

Keywords — Embedded System, Architectures, ARM, x86

I. INTRODUCTION

A combination of software and hardware components along with some mechanical and sensor parts which are designed to perform a particular function continuously is called as an embedded device. Mostly these devices are parts in a larger system for example antilock braking system in a car.

Embedded systems have very large and varying requirements and limitations. Size of these systems should be small and cost also should be low. Some devices have very short lifetime where as some have long. Also these systems have to withstand many environmental conditions such as radiations, vibrations and also climatic conditions.

All embedded devices will interact with their environment which makes these devices as real-time constraint ones. Apart from these issues all embedded devices will do multitasking which increases the complexity of real-time constraints. Figure 1, shows a general organization of embedded system. Apart from processor and memory, a large number of elements will be connected to the system.

Embedded Devices will be interfaced with many sub systems which will measure and interacts with external environment and some system for human interaction starting from a simple LED to a complex graphics LCD. These types of devices will also have some specific purpose ports for diagnosing the errors.

Generally the processor which is the heart of the system will be either of X86 or ARM. In this paper both the architectures will be discussed based on their usage in embedded devices.

Corresponding Author: *cudarun@gmail.com*

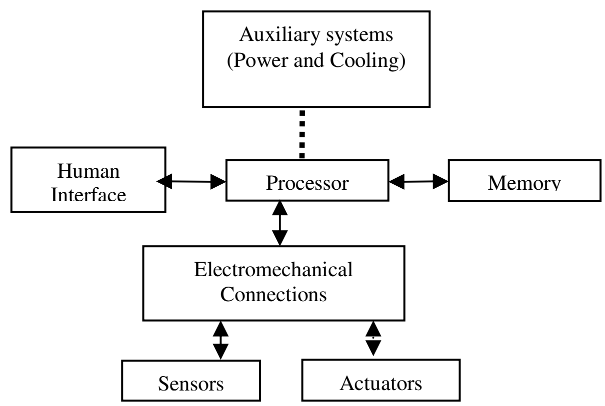

Figure 1 Example for organisation of embedded system

From the date of its introduction in market the CISC type X86 architecture dominates the market but not in embedded systems, even though many advances have happened dramatically in the technology and organization over the years. But the instruction set architecture (ISA) is still backward compatible with earlier versions. Because of all these advancements the embedded device manufacturers have started to use X86 architecture based processors in their products.

A family of RISC-based microprocessors produced by ARM Ltd is called as ARM, which has introduced many numbers of families with high performance and increasing functionality. ARM processors are targets mostly on embedded real time systems and in applications where security is the most important factor.

II. Comparison on X86 and ARM

In this paper two processors from both architectures are compared based on some parameters.

Cortex A9: ARM processors have a very wide range of processors starting from ARM7 processors which runs at 15MHz to the Cortex-A9 running at more than 1 GHz. [1] The Cortex A9 is the most power efficient and high performance ARM processor available in the market currently. Cortex A9 is a 32-bit RISC architecture with Harvard architecture. The operating frequency of Cortex A9 varies from 600MHz to over 1 GHz. The Cortex-A9 uses the ARM7 architecture which includes two pipelines. First one is a 13-stage integer pipeline and second one is a 10-stage NEON pipeline which is used for signal processing and accelerating multimedia applications. The Cortex A9 also has Jazelle RCT technology which is used for fast compiling of Java byte codes.[1] It also has a method which reduces the code size with maintaining the same performance is called as Thumb technology.

The Cortex A9 is specially designed for mobile and embedded devices, so it has to consume very less power but with high performance. To achieve this some modifications has been done to the older variants of ARM. Dynamic branch prediction is used to reduce branch penalties. [5]The Cortex A9 also has an average value of 0.9 for Instructions per Cycle.

ATOM 450- Two variants of atom have been released by Intel that are specially designed for mobile processor market. Those two versions are Atom N and Atom Z series. Atom Z series is designed for low cost desktops and notebook markets while Atom N series is for mobile internet devices. Both of these processors run between 0.8 to 2.0 GHz [3]. In this paper Atom N450 is analyzed which has single core with two threads.

The Atom N450 is an x86-based architecture with dual in order instructions which supports both 32 bit and 64-bit architecture. In Atom 450 all instructions will be translated into micro operations which contain a load and store instruction for all ALU operations [3]. Atom N450 has a 16 stage pipeline, which includes three stages, Decode, Instruction dispatch and data cache access. It also has two Arithmetic and Logic Unit and two Floating Point Units. Shift operations will be managed by first ALU and Control jumps will be managed by second one. FPU is especially for arithmetic and integer operations. All addition operations will be done by first FPU, multiply and divide will be done by second FPU Atom N450 also has Hyper Threading technology [3]. It is an Intel's technology which implements multithreading. By using this technique operating system will create two virtual processors for a single processor core and those two virtual cores will share the workload between them. Intel has enhanced some power saving techniques for Atom N450, which consists of six low-power modes for the bus and the cache memory [3].

III. Instruction Set Architecture

The X86 and ARM uses two different instruction set architectures, Complex Instruction Set Computing (CISC) and Reduced Instruction Set Computing (RISC). Basically, the difference is that CISC focuses on completing a function in just one instruction, while RISC would use a couple of simple uniformly formatted instructions to do the same. At a glance, it might appear that RISC is less efficient. However, we have to keep in mind that the simplistic nature of RISC instruction allows it to be processed in usually just one clock cycle [4]. For an operation if RISC takes 2 instructions to accomplish the operation in 2 or 3 clock cycles whereas CISC will take only one instruction to accomplish the task but it may take 3 or 4 clock cycles. So more number of clock cycles means then it will consume more amount of energy, which is not preferable in embedded devices [7].

IV. Power Consumption

All embedded devices are powered through batteries only and very few embedded devices are powered with wall mount supplies.

In general there is a view that ARM consumes less power when compared to X86. This is because from the starting time onwards ARM focuses on low power consumption. But the new processors from X86 based architectures also concentrates more on low power consumption.

If a processor consumes more amount of energy then it will dissipate large amount of heat, which is not preferable in embedded devices. So to keep the system in normal condition it's necessary to keep some cooling parts, because of these cooling devices the final size of the embedded devices will be increased [2].

Both processors have many power saving mechanisms and the following table 1 shows power consumption of both processors.

Cortex A9	Atom N450
1.9 W	5.5W

Table 1Maximum TDP

V. Power saving techniques

The Arm Cortex A9 has some inbuilt mechanisms to control dynamic and static power dissipation. It has a very accurate branch prediction technique which will reduce the power consumption due to incorrect instruction fetch and decode operations [1]. To reduce the number of cache flushes, the Cortex A9 uses physically addressed caches, which will save some power.

Generally some amount of power will be wasted during the translation and protection look up cycles, to reduce this Cortex A9 uses micro TLBs and access to cache memory in

Cortex A9 will be in sequential manner to reduce the number of access to Random Access Memory [1].

The Table 2 shows which parts of the processor will be in which state during different modes of Cortex-A9.

Processor modes	RAM Arrays	Processor logic	Data Engine
Full run mode	ON	ON	ON
Standby	ON	ON	ON
Dormant	Retention State	OFF	OFF
Shutdown	OFF	OFF	OFF

Table 2 Processor modes and its power consumption

In full run and in standby mode, it is possible to either disable Media Processing Engine (MPE) or to switch it by considering this MPE as a Coprocessor.

The Atom N450 uses Intel's Enhanced Intel SpeedStep Technology. This technology optimizes the processor's operating frequency and the core voltage used by core based on workload. When the processor is not executing any codes, it will be in an idle state [3]. This frequency selection technology can be controlled by software also by writing in Model Specific Registers (MSR).

VI. Area Occupied by the Processor

In embedded devices space occupied by components will be a huge constraint for the designers. All users of embedded devices would like to use the devices which are very compact. In table 3 a small comparison has been done between Cortex A9 of ARM and Atom N450 of X86 architectures.

Cortex A9	Atom N450
$70mm^2$	$66mm^2$

Table 3 Die Size of processors

VII. Amount of Memory Support

The amount of memory that the processor used in an embedded device is also a critical factor. It is not possible to connect large amount of external memory in embedded devices. It is also advisable for embedded programmers to use only the memory which is available in the processor.

Cortex A9	Atom N450
1GB	2GB

Table 4 Amount of memory support done by Processors

VIII. Special Features of ARM Cortex A9

The Performance Monitoring Unit (PMU) of Cortex A9 will gather statistics on the processor operation and memory by using six counters. These counters will count if any error has occurred and these counters and their associated registers can be accessed through debug interface. These registers will have the data about the particular error [1]. For example, these registers can be able to count how many Java byte codes are executed, how many Jazelle backward branches are executed , how many Instruction cache dependent stalls occurred, how many Load/Store Instructions are executed.

The Memory Management Unit (MMU) controls both the L1 and L2 memory systems and it is also responsible to translate virtual addresses to physical addresses. If external memory is connected to the embedded device then this unit controls those accesses also [1]. MMU also has permissions check capability

The Cortex-A9 processor uses a Program Trace Macrocell (PTM) interface, which works with the Program Flow Trace (PFT). PFT is an instruction only trace protocol that uses waypoints to trace the code. Waypoints are changes in the program flow or events such as branches.

Interfaces:

The processor has some external interfaces such as AMBA AXI interfaces, APB CoreSight interface and DFT interface.

IX. Special Features of Atom N450

The Intel Atom N450 has Thermal Monitor which helps the processor to control its temperature by activating the TCC (Thermal Control Circuit) when the processor silicon reaches its maximum operating temperature [3]. The temperature at which the Intel Thermal Monitor activates the TCC is not user configurable.

The Thermal Monitor uses two modes to activate the TCC, automatic mode and on demand mode. If both modes are in active state, then automatic mode takes precedence. There are two automatic modes called Intel Thermal Monitor-1 and Intel Thermal Monitor-2. By writing appropriate values in MSR these modes can be selected.

The Atom N 450 also contains a Digital Thermal Sensor (DTS) in which the values can be read out through Model Specific Registers (MSRs) [3]. This DTS will not work as an input output interface. The DTS is only valid while the processor is in the normal operating state.

X. Conclusion

Intel offers processors itself, whereas ARM offers processor cores only. This basic difference in their technologies enables ARM to offer a product that can be integrated into a full System on Chip which can be used in specific applications. This approach is a simple and important advantage which Intel has failed to overcome even with many advancements also.

Nowadays high end devices are also featured by x86, while the rest would rely on ARM. If the designer wants their devices to perform telemetry and data transmission over the Internet for critical applications, x86 would still be equally qualified (or rather better). On the other hand, if the device is meant for only video and audio applications, ARM is the straight choice. In near future smart phones will be used as mainstream computing devices, if it happens, then ARM could gain advantage over x86.

If the designer mainly concentrates on power consumption only then they can use ARM but if they have constraints like space, amount of memory support then they can use X86.

REFERENCES

[1] ARM Cortex-A9 Technical Reference Manual

[2] Lamber A. et. al., “Keys to Silicon Realization of Gigahertz Performance and Low Power ARM Cortex-A15”, Unpublished.

[3] Intel Atom Processor N400 & N500 Series Datasheet - Volume 1

[4] Emily Blem, Jaikrishnan Menon, and Karthikeyan Sankaralingam “Power Struggles: Revisiting the RISC vs. CISC Debate on Contemporary ARM and x86 Architectures” IEEE International Symposium on High Performance Computer Architecture, ISBN: **978-1-4673-5585-8**, Page No (**1-12**) Feb **23-27, 2013**.

[5] Katie Roberts Hoffman and Pawankumar Hedge “ARM Cortex A8 vs. Intel Atom: Architectural and /Benchmark comparisons” Unpublished and available at www.embedded.com/electrical-engineer-community/general/4400497/Cortex-A8-vs--Intel-Atom--Architecture-Benchmark-comparisons

[6] The ARM vs. x86 wars have begun: In-depth power analysis of Atom, Krait & Cortex A15 available at “:http://www.anandtech.com/show/6536/arm-vs-x86-the-real-showdown/.”

[7] D. Bhandarkar and D. W. Clark. “Performance from architecture: comparing a RISC and a CISC with similar hardware organization”, International Conference on Architectural support for programming languages and operating systems ISBN: **0-89791-380-9,** Page No (**310-319**) April **8-11, 1991**.

