Abstract—The ARM architectures are reduced instruction set computer (RISC) instruction set architectures (ISA). The x86 architecture is CISC design with importance on backward compatibility. ARM architecture is more widely used than the x86 architecture. ARM is a simple architecture, which leads to small silicon area and have lot of power save features while x86 becoming a power beast in terms of both power consumption and production. [1]

Index Terms-ARM, x86, RISC,CISC

I. Introduction

ARM-based products’ market share in 2010 was over 95% in smart phone market .[2] ARM is more widely used than x86 architecture in mobile phones. The reason being the difference in the design dogma that the two architectures follow. The x86 is a CISC architecture i.e. Complex Instruction Set Computer. The ARM is a RISC-based designs i.e. Reduced Instruction Set Computer. CISC architectures can have up to thousands of individual commands supported by the processor that can be used in machine code. The assembly commands can be of single operation to several hundred or more in length. Whereas, RISC-based CPUs understand only a few different instructions which are necessary to get the job done.

![Figure 1 Image Source](image-source)

<table>
<thead>
<tr>
<th>Format</th>
<th>Operations</th>
<th>Operands</th>
</tr>
</thead>
<tbody>
<tr>
<td>RISC/ARM</td>
<td>• Fixed length instructions</td>
<td>• Simple,single function operations</td>
</tr>
<tr>
<td></td>
<td>• Relatively simple encoding</td>
<td>• Single cycle</td>
</tr>
<tr>
<td>CISC/x86</td>
<td>• Variable length instructions</td>
<td>• Complex multi-cycle instructions</td>
</tr>
<tr>
<td></td>
<td>• Common instructions longer/complex x</td>
<td>• Encryption</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• String manipulation</td>
</tr>
</tbody>
</table>

II. ARM Architecture

The ARM architecture is main driving force behind the ARM and Thumb Instruction Sets and debug, memory and execution models used by ARM processors. Memory models include virtual memory, caches, Tightly Coupled Memory (TCM), and memory protection. It also includes additional features such as floating-point support, Single Instruction Multiple Data (SIMD) instructions, Security extensions, Java Byte code acceleration, and Multiprocessing support. [3]

The ARM processor architecture supports 32 bit RAM and 16 bit Thumb ISA’s.

**RISC has following features:-**

- Large uniform register file
- Load/store architecture
- Simple addressing modes
- Addressing modes which increment and decrement automatically
Load and store multiple instructions

**Features of Thumb architecture:**
- It is an extension to the 32-bit ARM architecture.
- It features a subset of the most commonly used 32-bit ARM instructions which are compressed into 16-bit opcodes.
- It has got excellent code-density so that we get minimal memory size, reduced cost and power efficiency. [4]

**Figure 2** Image source - http://www.embeddedinsights.com/epd/arm/arm-arm1156t2-s-arm1156t2f-s.php

**x86 Architecture**
The processors having x86 architecture use complex instructions set computer (CISC) architecture, i.e. there is a finite number of special-purpose registers that are used. The complicated special-purpose instructions predominate over the general-purpose instructions. The x86 processor traces its existence at least as far back as the 8-bit Intel 8080 processor.

**CISC has following features:**
- More emphasis on hardware.
- It includes multi-clock complex instructions.
- Memory to Memory: "LOAD" and "STORE" incorporated in instructions.
- Code size is small therefore high cycles per second.
- Transistors are used for storing complex instructions.

**Figure 3** Image source - http://www.cosc.brocku.ca/~bockusd/3p92/Local_Pages/8086_architecture.htm
Performance Comparison
The comparison is not that easy because we are dealing about two different instruction sets—complex instruction set computing (CISC) and reduced instruction set computing (RISC). The basic difference is that CISC focuses on completing a function in just one instruction, whereas doing the same job RISC processor uses a couple of simple uniformly formatted instructions. It might appear, that RISC is less efficient than CISC. However, because of the simplistic nature of RISC, an RISC instruction can be processed in just one clock cycle whereas, an CISC instruction takes couple of cycles for it to be processed.

A drawback of RISC is that it uses more RAM than CISC-based processors, which is why CISC-based processors were predominant until now. This drawback has been overcome with the inexpensive and fast availability of RAM. Most mobile device vendors are using ARM-based solutions to power their flagship devices.

System Level-Optimization
RISC is way simpler than CISC and it uses fewer transistors so that it remains cool under stressful load, thus providing a consistent performance. RISC uses a poorly implemented algorithm that has a complexity of O(nlogn) which will outperform the best implemented algorithm with a complexity of O(n2) on large problems.

III. BIOS Or Bootloader
BIOS (Basic Input/Output System) is used exclusively in X86 processors. BIOS is the program a personal computer's microprocessor uses to get the computer system started after it is turned on. It also manages the data flow between the computer's operating system and attached devices such as the hard disk, video adapter, keyboard, mouse, and other hardware devices. It also offers additional functionality such as hardware testing and diagnostics, flexible configuration and platform management. A disadvantage of BIOS is the prolonged start-up time. Whereas, ARM architecture uses the much faster boot loader approach for hardware configuration and operating system start-up. Some of the functionalities in the X86 BIOS are also feasible in ARM based applications but the development efforts are costlier.

IV. Conclusion
The basic difference in the technologies of ARM and Intel is that Intel offers processors, while ARM offers processor cores. This difference in their technologies enables ARM to offer a product that can be integrated into a full SOC optimized for use in a specific application. This is an important competitive advantage that Intel has failed to overcome through expensive process-level improvements.

ARM, currently holds a high ground, with most mobile devices selling with ARM on the board. For high-end video and audio capabilities, ARM is the undoubtedly the best choice.

<table>
<thead>
<tr>
<th>Comparison of ARM , Atom</th>
<th>Cortex A9(40nm)</th>
<th>Atom N270(45nm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Cores</td>
<td>2(4 maximum)</td>
<td>1 Core,2 HT threads</td>
</tr>
<tr>
<td>Frequency</td>
<td>800 Mhz(Po)</td>
<td>1.6 Ghz</td>
</tr>
<tr>
<td></td>
<td>2Ghz(Per)</td>
<td></td>
</tr>
<tr>
<td>L1 Cache Size</td>
<td>32KB I/D</td>
<td>32KB I/D</td>
</tr>
</tbody>
</table>

Figure 4 Image source- http://lizards.opensuse.org/tag/opensusearm

IV. Conclusion
The basic difference in the technologies of ARM and Intel is that Intel offers processors, while ARM offers processor cores. This difference in their technologies enables ARM to offer a product that can be integrated into a full SOC optimized for use in a specific application. This is an important competitive advantage that Intel has failed to overcome through expensive process-level improvements.

ARM, currently holds a high ground, with most mobile devices selling with ARM on the board. For high-end video and audio capabilities, ARM is the undoubtedly the best choice.

<table>
<thead>
<tr>
<th>Comparison of ARM , Atom</th>
<th>Cortex A9(40nm)</th>
<th>Atom N270(45nm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Cores</td>
<td>2(4 maximum)</td>
<td>1 Core,2 HT threads</td>
</tr>
<tr>
<td>Frequency</td>
<td>800 Mhz(Po)</td>
<td>1.6 Ghz</td>
</tr>
<tr>
<td></td>
<td>2Ghz(Per)</td>
<td></td>
</tr>
<tr>
<td>L1 Cache Size</td>
<td>32KB I/D</td>
<td>32KB I/D</td>
</tr>
</tbody>
</table>
Supply Voltage          1.05V(Per)          0.9-1.1625 V
Transistor Count        26,000,000?        47,000,000
Die Size                4.6 mm2(Po)         6.7 mm2 (Per)        26 mm2
Power Consumption       0.5 W (Po)          1.9 W(Per)           2.5W(TDP)

Figure 5 Image source-http://www.cs.virginia.edu/~skadron/cs8535_s11/ARM_Cortex.pdf

Figure 6 Image source – nvidia-arm-x86-shipments

References
3. Infocenter.arm.com – home>key features of ARM Architecture Versions
5. www.electronicsforu.com