ARM Cortex-A8 vs. Intel Atom: Architectural and Benchmark Comparisons

University of Texas at Dallas
EE6304 Computer Architecture Course Project – Fall 2009
Katie Roberts-Hoffman, Pawankumar Hegde

***Abstract*—Mobile Internet Devices (MIDs) are increasingly being found in the consumer electronic markets that require low power and high performance processors. ARM and Intel have become major competitors in this domain. In order to meet the architectural requirements of MIDs, both companies have developed the following processor architectures to address market needs: the Cortex-A series from ARM and the Atom-N series by Intel. In this paper, the Cortex-A8 and Atom N330 are compared with respect to their architectures and their real-world performance in specific implementations. For performance comparison, a Texas Instruments OMAP3530-based Cortex-A8 was selected along with an Intel Atom N330-based Desktop PC. Four benchmarks are used to compare the performance with respect to clock rates, power, price, and die size. Results show that the Atom has better raw performance while the Cortex-A8 has significantly greater power efficiency.**

I. Introduction

Over the past two years, two key devices have drastically changed the consumer electronics marketplace. The first is Apple's iPhone, which has single-handedly redefined the concept of "app stores," increased mobile web browsing expectations, and created a demand for all-in-one devices containing GPS, Bluetooth, media players, e-mail, instant messenger clients, and more. This list of applications is actually the same as those found on the second market-changing device: the netbook. Netbooks are synonymous with low cost, low power, and small size. Thus it seems that the iPhone (and similar type smartphones, such as the Motorola Droid) and netbooks share a number of traits and purposes although they are based on two very different microprocessor architectures: ARM Cortex-A8 series and the Intel Atom.

ARM processors now dominate the mobile phone space due to their lower power requirements—with 98% of mobile phones having an ARM processor. Last year, ARM licensed over 1.5 billion processors to companies such as Texas Instruments, Qualcomm, and Freescale [1]. ARM Cortex-A8 processors can now also be found outside of the handset market in set-top-boxes, gaming systems, e-books, point-of-sale systems, and even occasionally netbooks. As ARM's processor performance increases to Desktop PC levels, they are starting to compete for traditional x86 sockets alongside the Intel Atom, which was originally designed for netbooks. As Intel continues to decrease the Atom's power requirements, it may become quickly adopted as the defacto embedded processor in handheld devices as well. Both of these companies have modified their respective processor architectural designs to meet the requirements for Mobile Internet Devices (MIDs). In this paper, a comparison of the ARM Cortex-A8 and the Intel Atom N330 processor architectures is presented with a focus on performance and power. In addition, specific implementations of both architectures are used for a real-world performance comparison including an OMAP3530-based Cortex-A8 and an Intel Atom N330-based Desktop PC. Performance results using four key processor benchmarks are presented which can be used as an aid in showing how the architectures ultimately compare in regards to clock rates, power, price, and die size.

II. Architecture Overview

A. Cortex-A8

ARM processors range in performance, power and price from ARM7 processors at 15MHz to the Cortex-A8 running at more than 1 GHz, and the recently announced symmetric multi-core Cortex-A9. For practical purposes, even the fastest ARM9 and ARM11 processors at clock speeds of 500MHz cannot compete with modern x86 architectures used in typical Desktop PCs. Thus, this paper only focuses on the Cortex-A8 architecture. Limited comments on the Cortex-A9 are also provided for perspective, but as the device is not widely available, benchmarking results are

not provided.

The Cortex-A8 is the highest performance and most power efficient ARM processor currently in production today. Like other ARM processors, it is a 32-bit RISC architecture with 16 registers and a Harvard memory architecture. It can run at speeds between 600MHz to over 1 GHz while averaging as little as 300mW of power. The Cortex-A8 uses the ARMv7 architecture, which is ARM's first superscalar architecture. The ARMv7 architecture includes two pipelines: a 13-stage integer pipeline and a 10-stage NEON pipeline (useful for accelerating multimedia and signal processing applications). In addition, the ARMv7 architecture includes the Jazelle-RCT technology that allows for just-in-time compilation of bytecode languages (such as Java), and Thumb-2 technology that reduces code size while maintaining equivalent performance by allowing 16-bit instructions to coexist with 32-bit instructions [2].

Since the Cortex-A8 was designed for the mobile/embedded space, the previous ARM architectures needed to be modified to provide the best performance at the lowest power. To achieve this, the ARMv7 architecture uses a super-scalar architecture with in-order instruction issue and support for forwarding paths. An in-order instruction is less complex than an out-of-order one and thus yields lower power consumption. With multiple ALU pipelines, the Cortex-A8 has an average Instructions Per Cycle (IPC) of 0.9. Branch penalties are kept at a minimum by using dynamic branch predictors which were designed with a 95% level of prediction accuracy and by resolving most branches in a single stage [3].

One of the most interesting performance-increasing features of ARMv7 architecture is its NEON technology that can be used as a SIMD accelerator. This allows the Cortex-A8 to perform four multiply-accumulates instructions per cycle via dual-issue instructions to two pipelines [4]. NEON is a hybrid 64/128-bit architecture that is capable of both integer and floating-point operations. Geared towards accelerating multimedia and graphics, it provides native support for complex numbers, and coordinates to avoid "data shuffling" overhead. The NEON pipeline has its own register file, but shares its caches with the main pipeline. NEON support has been added to compilers so that programmers need not hand code NEON assembly or use intrinsic to take advantage of the extra pipeline, although specific compiler flags are needed to invoke this feature. ARM has stated that 4.5x performance increase is expected for MPEG4 decoding using NEON versus older ARMv5 technology [3].

B. Intel's Atom

Intel has released two variants of the Atom for the mobile processor market: the Atom-N and Atom-Z series. While the Atom-Z was designed to cater to the low-cost desktop and notebook market, the Atom-N is targeted at MIDs. Compared to the Intel Celeron architecture, the performance has been reduced by nearly half. All of the processors in the Atom series are clocked between 0.8-2.0GHz and built with 45nm technology. The Atom N330, used for benchmarking in this paper, is a dual core processor whereas other Atom processors in the family that is single-core. The Atom N330 runs at 1.6 GHz with a FSB running at 533MHz and an average consumption of 8W [5].

The Atom family is an x86-based architecture with dual in-order instructions. The Atom N330 is the only 64-bit architecture in the family apart from N230, with the remainder using 32-bit. All instructions in the CPU are translated into micro-operations (μ-ops) that contain a memory load and a store operation for each ALU operation. This kind of design is similar to that of traditional x86 architectures but makes use of even more powerful μ-ops than those used in previous designs. Such architecture enables significant performance per cycle for each core without any instruction re-ordering, speculative execution or register re-naming. However, this μ-op architecture increases the branch miss-prediction penalty. Intel's Atom has a 16-stage pipeline, including three instruction phase stages: decode (3 stages), instruction dispatch (2 stages), and data cache access (3 stages). Each Atom core is also equipped with two ALUs and two FPUs. The first ALU manages shift operations, and the second controls jumps. All arithmetic operations, including integer operations, are automatically sent to the FPUs. The first FPU is simple and

limited to addition, while the second manages SIMD and multiply/divide operations. While basic instructions are executed in one cycle, it can take up to 31 cycles (for floating point division) for complex instructions [6].

To increase performance, Atom also supports Hyper-Threading. Hyper-Threading is Intel's proprietary technology that implements simultaneous multi-threading. For each processor core that is physically present, the operating system can create two virtual processors and shares the workload between them. Hence a dual-core Atom, such as the N330, appears to have four cores to the operating system [7]. Finally, in order to support the multimedia operations, the Atom is commonly paired with a 945G chipset that further increases both its performance and power consumption.

Figure 1. Intel Atom motherboard includes a fan for heat dissipation [8]

Intel has put significant efforts in reducing the power consumption for the Atom processor family. It consists of six low-power modes for the bus and the cache. The Atom N330's Thermal Density Power (TDP) rate is measured at 2.5W. The Atom N330 is expected to consume an average 8W of power when paired with the 945GSE chipset, and has a specified maximum TDP of 11.8W[5].

C. Comparison Overview

While there are a number of whitepapers and architecture guides for the Cortex-A8 and Atom, these can't provide a direct comparison of performance between the two architectures. For this, the actual implementations of both architectures are needed. Due to cost and availability, the Beagle Board, based on Texas Instrument's OMAP3530 processor, is compared to an Intel Atom N330 Desktop computer.

The OMAP3530 contains a Cortex-A8 as the GPP, a SGX510 GPU and C64x+ DSP for multimedia. The Beagle Board's Cortex-A8 is clocked at 600MHz with 256MB DRAM and 256 NAND. The OMAP3530 uses a number of power management techniques to reduce power, including dynamic voltage and frequency scaling, dynamic power switching, and exceptionally low stand-by power. The OMAP3530 supports multiple modes of operation with varying power requirements, including: device-off mode (consumes 0.59mW), standby mode (7mW), audio and video decode mode (with ARM at 500MHz and DSP at 360MHz consumes 540mW), and full-on (with ARM at 600MHz, DSP at 430MHz, and SGX510 at 110MHz consumes 1.5W)[9]. The 1k-unit price for the OMAP3530 is $32[10].

The Intel Atom N330 Desktop used in this experiment has two Atom cores that are each clocked at 1.6 GHz. The N330 dual-core processor actually looks as if Intel simply combined two Atom N230 single-core processors. The added core unfortunately doubles the processor's TDP from N230's 4W to 8W. The N330 is available as part of Intel's D945GCLF2 desktop motherboard, consisting of an Intel 945GC Express Chipset, an Intel GMA 950 for integrated graphics, and a 533 MHz system bus. The Intel D945GCLF2 motherboard also has an S-Video connector, gigabit Ethernet, 6-channel high definition audio, and a single DIMM socket that can support up to 2 GB of DDR2 667/533 memory. The 1k-unit price for the Atom 330 is $43[5].

For the following set of experiments, each processor is running Linux 2.6.28, using the Angstrom distribution for the Beagle Board and Ubuntu 9.04 for the Intel Desktop. Linux was chosen due to its widespread availability and since it is the predominant operating system for high performance consumer electronics, whether that is mobile phones, set-top boxes or GPS systems. In addition, there are a number of similar tools available in the open source community. The GNU C Compiler (GCC) is available for both the ARM Cortex-A8 platform and for x86-based processors.

Here it is assumed the reader is familiar with the standard process of compiling code and booting a Linux kernel for an x86 target, but a brief detour for the ARM processor is given. The Cortex-A8 itself can compile code (known as native compilation), however it is frequently faster to cross-compile code on an x86 target (assuming that a modern x86 processor can run at significantly greater speeds than an ARM processor). The Code Sourcery 2007q3 cross-compiler was chosen for the Beagle Board. The cross-compiler compiles the specified source on an x86 system. Following compilation, the compiler out file is copied to an SD card that contains the Beagle Board's file system. The same SD card also contains the Linux uImage and bootloaders needed for booting the device. This SD card is inserted into the Beagle Board and the operator can interact with the Beagle Board via RS232 serial port or through the DVI connector with USB mouse and keyboard. For further instructions on how to cross-compile and boot the Linux kernel on the Beagle Board, see http://www.beagleboard.org.

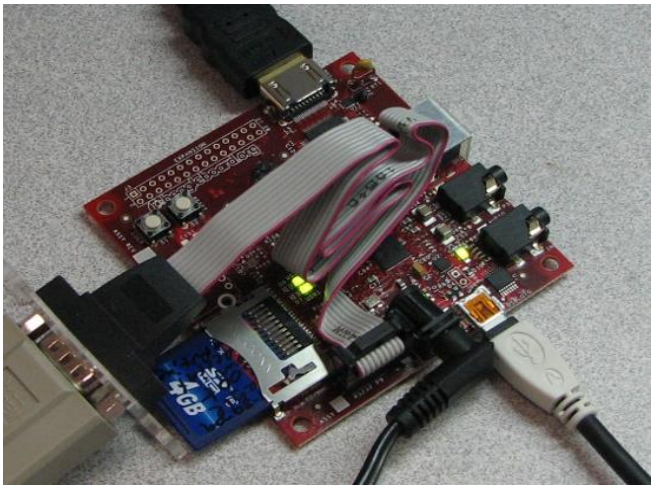


Figure 2. Beagle Board shown with SD card, Serial cable and USB cable [11]

Once both platforms were functional, a set of benchmarks had to be selected. For reproducibility, only open source benchmarks were selected. Four benchmarks were chosen—two integer-based and two floating-point-based. The former gives insight into typical text-based applications, such as word-processing, whereas the latter may be indicative of scientific computing and audio processing. The following benchmarks were selected with summaries provided for each below:

1. Fhourstone: An integer benchmark that effectively solves the "Connect-4" game. The benchmark calculates the number of positions searched per second.
2. Dhrystone: A traditional synthetic integer benchmark designed to test a spread of commonly used functions. It outputs the number of Dhrystone loop iterations run per second, also known as Dhrystone MIPS (DMIPS).
3. Whetstone: A common synthetic floating-point benchmark used to measure typical scientific computing performance. It outputs the number of Whetstone Instructions per Second (WIPS).
4. Linpack: A floating-point benchmark used to measure linear algebra performance by calculating the time spent solving *N* by *N* linear equations. It outputs the number of floating-point operations per second performed (FLOPS).

III. Benchmark Results

Table I displays the results obtained from the four benchmarks run. Clearly each benchmark can only be compared to itself, as the resulting values are meaningless outside of that benchmark's context. The same gcc compiler options were used on the Beagle Board and Intel Desktop for the same benchmark. By varying the compiler options, better performance may be possible than that reported below. ARM research indicates an expected 1200 DMIPS on the Beagle Board while only 883 were found in this experiment [2]. It is hypothesized that sub-optimal compiler flags led to this disparity. Below, the raw performance measurements are given, followed by performance per MHz, performance per Watt, and performance per price. The latter three should give insights into how strongly or weakly each device performs for a targeted market. For example, an e-book may be very sensitive to power but due to the novelty of the item, may be less sensitive to price. Note that the power comparison uses the datasheet power consumption values for both processors which are expected to be accurate as each CPU is fully loaded for each of these benchmarks (verified via

'top' during execution).

From these benchmarks, it appears that the Atom Desktop consistently outperforms the Beagle Board with respect to raw processing power. It's clear here that the Beagle Board does significantly better on integer benchmarks (Fhourstone and Dhrystone) versus the floating-point benchmarks (Whetstone and Linpack). The NEON pipeline is only used when specific compiler flags are used, specifically, "–mtune=cortex-a8 –mfpu=neon –free-vectorize –mfloat-abi=softfp". If these compiler flags are not used, the Whestone benchmark performance decreases by a factor of 2, and the Linpack performance by 4.5x. The Intel Atom does particularly well on floating-point operations since all arithmetic operations (both integer and floating-point) use the Atom's FPUs and it is a fair assumption that Intel has optimized these units. However, all is not lost for the Beagle Board as it appears significantly more power efficient and provides more performance per dollar than the Atom Desktop.

TABLE I
Benchmark Performance Results for BeagleBoard and an Atom 330 Desktop

	Fhourstone (Kpos/s)	Dhrystone* (DMIPS)	Whetstone (MIPS)	Linpack (KFLOPS)
Raw Benchmark Performance				
Beagle Board	731	883	100	23376
Atom 330 Desktop	2054	1822	1667	933638
Benchmark Performance per MHz (Beagle at 600MHz, Atom at 1.6GHz)				
Beagle Board	1.22	1.47	0.17	38.96
Atom 330 Desktop	1.28	1.14	1.04	583.52
Benchmark Performance per W (Beagle at 0.5W and Atom at 2W)				
Beagle Board	1462	1766	200	46752
Atom 330 Desktop	256.75	227.75	208.38	116704.75
Benchmark Performance per Dollar(Beagle at $32, Atom at $43)				
Beagle Board	22.84	27.59	3.13	730.5
Atom 330 Desktop	47.77	42.37	38.77	21712.51
Benchmark Performance per mm2 of the final die size (OMAP at 60mm2, Atom at 52mm2)				
Beagle Board	12.18	14.72	1.67	389.6
Atom 330 Desktop	39.5	35.03	32.06	17954.58

There is a variety of other benchmark data available, usually targeting specific end equipments. For example, a gaming unit needs graphics performance whereas a personal media player needs strong video decoder performance. Below are other benchmarks that were not reproduced, but provide insight into the two devices. The reader is cautioned in advance that the following multimedia benchmark contains references from three different sources thus the conditions were not the same for each simulation. However, it does demonstrate that a system-on-chip device like OMAP3530 can achieve multimedia playback at significantly less power (approximately 1W for the Beagle Board with DSP decoding versus approximately 8W for the Atom 330 Desktop solution). Not only does the OMAP3530 exhibit power savings, but also allows the ARM to service other tasks during the decoding.

Table II
Media Benchmark Performance Results

Media Benchmarks - D1 MPEG4 Decode	
	Load (Million cycles per sec)
Beagle Board (ARM)	500
Beagle Board (DSP)	90
Atom 330 Desktop	880

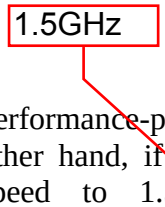
[4][12]

Similarly, the OMAP3530 shines in graphics performance at low power. The SGX510 accelerator is capable of delivering 10 million polygons per second without adding a significant power penalty. This type of performance is comparable to graphics found on PlayStation2. The Intel Atom currently requires an external graphics chip to achieve this performance that increases the power consumption by 6 or more Watts as previously discussed.

IV. Conclusions

The basic benchmarking results show nothing surprising: the Cortex-A8 provides significant power savings while the Intel Atom's pure processing power is unmatched. The main culprit to the Atom's power efficiency is the external 945GC Express Chipset as the Atom processor core alone is relatively low power at 2W. Unfortunately this chipset is usually used in multimedia applications indicating that the Cortex-A8 has considerable advantages in multimedia areas if it is integrated on a system-on-chip device like OMAP3530 with a DSP.

Intel's roadmap includes two power saving features: moving to 28nm technology and integrating the 945GC chipset onto the main processor chip. If Intel is able to reduce the processor (with 945GC chipset integrated) to less than 1.5W, it will achieve approximately the same



performance-per-power as the OMAP3530. On the other hand, if the Cortex-A8 increases its clock speed to 1.5MHz, it can achieve similar performance to the Intel Atom as both chips provide approximately the same performance-per-MHz In an Intel like fashion, ARM's roadmap includes symmetric multi-core Cortex-A9 processors. The Cortex-A9 is the next generation Cortex-A8 and is expected to achieve clock speeds in excess of 2GHz. Assuming that both of these companies achieve their roadmap goals, then their processors will remain competitive in terms of performance, power, and price.

Thus the feature that may lead to marketplace dominance may not be limited to the hardware but in software. While there are many more ARM processors in production than x86-based processors, more consumers are familiar with x86-based software. For example, Linux netbooks have been relatively unsuccessful in the market despite their lower cost, as the majority of consumers prefer to run Windows. In this sense, ARM must ensure adequate consumer software is available. With the introduction of the Apple iTunes app store and Google's Android marketplace, software is become more readily available for ARM processors to the end consumer. Supposing ARM software continues to become more abundant, the ARM and Intel rivalry can be expected to continue.

APPENDIX

Steps for benchmark reproduction

Fhourstones benchmark:
Available from
http://homepages.cwi.nl/~tromp/c4/fhour.html
On the Linux host, to compile and run on an x86:
$ wget
http://homepages.cwi.nl/~tromp/c4/Fhourstones.tar.gz
$ tar -xvf Fhourstones.tar.gz
$ make
$ ./SearchGame < inputs
To run on Beagle Board, simple change the Makefile to use arm-none-linux-gnueabi-gcc (or other ARM cross compiler) instead of the host gcc.

Dhrystone benchmark:
Available from
http://www.xanthos.se/~joachim/dhrystone-src.tar.gz
On the Linux host, to compile and run on an x86:
$ wget
http://www.xanthos.se/~joachim/dhrystone-src.tar.gz
$ tar –xvf Dhrystone-src.tar.gz
$ gcc -O2 -o dhrystone dhry21a.c dhry21b.c timers.c
Similarly, to run on Beagle Board, simply cross-compile using the Code Sourcery cross compiler.

Whetstone benchmark:
Available from
http://netlib.org/benchmark/whetstone.c
$ wget http://netlib.org/benchmark/whetstone.c
$ gcc -O -o whetstone whetstone.c -lm
$ ./whetstone -c 100000
Similarly, to run on Beagle Board, simple cross-compile using the Code Sourcery cross compiler with CCFLAGS=–mtune=cortex-a8 –mfpu=neon –free-vectorize –mfloat-abi=softfp.

Linpack benchmark:
Available from
http://www.netlib.org/benchmark/linpackc.new
On the Linux host, to compile and run on an x86:
$ wget
http://www.netlib.org/benchmark/linpackc.new
$ gcc -O -o linpack linpack.c –lm
$ ./linpack
Similarly, to run on Beagle Board, simply cross-compile using the Code Sourcery cross compiler with CCFLAGS=–mtune=cortex-a8 –mfpu=neon –free-vectorize –mfloat-abi=softfp.

REFERENCES

[1] Krazit,Tom, Apri 3, 2006. *ARMed for the Living Room*. Available: http://news.cnet.com/ARMed-for-the-living-room/2100-1006_3-6056729.html

[2] ARM, March 2008. *ARM Cortex-A8 Microprocessor Overview*. Available: http://www.arm.com/pdfs/Cortex-A8_data_brief_FINAL_2_.pdf

[3] Grisenthwaite, Richard, Dec 5, 2005. *IEEE Cambridge Branch Seminar: Cortex A8 Processor*. Available: www.iet-

cambridge.org.uk/arc/seminar05/slides/RichardGrisenthwaite.pdf
[4] BDTI, Feb. 26, 2008. *ARM Cortex-A8 Benchmark Results*. Available: http://www.bdti.com/bdtimark/cortex_a8.htm
[5] Intel, 2008. *Intel Atom Processor 330*. Available: http://ark.intel.com/Product.aspx?id=35641
[6] Dandumont, Pierre, June 5, 2008. *Intel Atom CPU Review*. Available: http://www.tomshardware.com/reviews/intel-atom-cpu,1947-4.html
[7] Intel, 2008. *Intel Hyper-Threading Technology*. Available: http://www.intel.com/technology/platform-technology/hyper-threading/index.htm
[8] Kucharik, Eliot, Oct. 6, 2008. *Dual Core Atom 330 image*. Available: http://www.fudzilla.com/images/stories/2008/October/Reviews/Atom_330/atom_330_d945gclf2_front.jpg
[9] Musah, Arthur, Dykstra. Andy, 2008. *Power Management Techniques for OMAP35x Application processors*. Available: http://www.ti.com/litv/pdf/sprt495
[10] Texas Instruments, 2009. *OMAP3530 Application Processor*. Available: http://focus.ti.com/docs/prod/folders/print/omap3530.html
[11] Benson. Duane, Aug. 28, 2009. *Open Source Hardware*. Available: http://blog.screamingcircuits.com/2009/08/index.html
[12] Felipe, C, Mar. 26, 2009. *OMAP3 Public DSP binaries now work*. Available: http://felipec.wordpress.com/2009/03/26/omap3-public-dsp-binaries-now-work/