ARMED for the Mobile Processing War

By Douglas A. Lyon
Fairfield University, 1074 N. Benson Rd. Fairfield CT 06824
dlyon@fairfield.edu

Abstract

This article offers an explanation for why the ARM computer architecture won the hearts and minds of the mobile device manufacturers when incumbents like Intel and AMD had such a huge head start and such a large economic barrier to entry.

Our conclusion is that ARM had a better business model, and not a better architecture. To make up for the perception of speed issues, ARM is joining the multi-core arms race. This may soon lead to desktop domination leaving only a legacy code base and sequential programming to enable incumbents to hold on.

Keywords CPU Architecture, Mobile Processor, ARM, INTEL, CISC, RISC

1. Introduction

According to a report by Forrester Research, there were over one billion PCs in use, worldwide, in 2008 and over 2 billion in 2015. The number of smartphone users grew from 2.1 billion in 2016 to 2.5 billion in 2019, with smartphone penetration rates increasing as well. Just over 36 percent of the world's population had a smartphone in 2018, up from 10 percent in 2011 as discussed in Statista [1]. Thus, in 2015, 68% of Americans had smartphones and 45% had tablets. More-over the number of other digital devices (desktops or laptops) are showing negative growth as in Pew [2].

The majority of mobile devices use the ARM processor. Why is this? We assert that the reason for this is that ARM does not make chips. Even so, its partners have shipped more than 125 billion chips with ARM inside. Google, Apple, Huawei, Samsung, etc., all have ARM inside their smartphones [3].

Intel has not been sitting still, as it too would like to be used in the mobile market. Intel has been trying for the last 10 years. However, its' latest effort, (Broxton) was cancelled in 2016. This is essentially capitulation from a $224 B USD company. AMD also has been trying to break into the market with the Ryzen mobile processor. But look as we might, it is hard to find anyone that has a cell phone with either Intel or AMD inside. The incumbents lost the mobile war!

2. How did Arm Architecture capture the Mobile Market?

Some have suggested that Arm processors are used in mobile devices because they are RISC machines. I do not think that is correct. After all, we have our choice of RISC processors, such as PowerPC, MIPS, Atmel AVR, M88000, etc. Why weren't these used? My theory is, the business model for ARM is fundamentally different from these prior entrants and this is a key factor to winning the mobile war.

The ARM business approach is to license its IP and encourage partners to incorporate this IP (Intellectual Property) into ASICS (Application-Specific Integrated Circuits). ASICS enable microelectronic packaging that adds value by lowering cost, reducing size and reducing power consumption. For example, Broadcom created SoCs for wireless embedded solutions that enabled inexpensive consumer electronics to be sold to a consumer market. In such SoCs ARM plays a central role, but overall system design and microelectronic packaging of a correct feature set are critical to the success of any product line.

After the design is completed, partners need only to select a fabrication house with the right capability/cost mix. The outsourcing of manufacturing enables the minimization of capital outlay normally required for in-house fabrication. For example, Intel has 1 MM square feet of clean room space and keeps $250MM USD in cash as a tactical resource that enables updating of the space. No wonder, considering an ASML machine like the NXE:3350B cost $90 MM USD in 2016.

Such machines are the price we pay to continue to make transistors smaller. But the outcome is a larger transistor budget. For example, an Intel Core i5 (with 4 cores in a MacBook Pro 13) has a 14 nm design rule for its transistors. In fact, the i5 for generation 5 through generation 9 used 14 nm design rule. This gives the Core i5 between 1.3 and 1.9 Billion transistors on a die that ranges from 82-133 mm square and has TDP (Thermal Design Power) of between 15 and 45 watts.

Now consider Apples' embrace of the ARM processor mobile applications. Apple A12 (iPhone XS Max) is an ARM SoC that gives their phone a hexacore option with an 83 mm square die and 6.9 billion transistors using the 7 nm FinFet process fabricated at TSMC (Taiwan Semiconductor). The Apple A12 consumes 3.64 watts [4] and clocks in at 2.5 Ghz. ARM superiority is obtained by enabling designers to pack many cores into SoCs that can be fabricated with state-of-the-art facilities that welcome the outsourcing and avoid the cash requirements of in-house fabrication.

3. The RISC in CISC

The industry has always had problems deciding between CISC and RISC processors. This is nowhere more evident than in Apple's CISC/RISC switching history. The Apple II was a 6502 8-bit CPU, the Thin/Thick/SE Macs were all based in 68000, The Mac II based in the 68020, 68030 and 68040. The RISC era ushered in the PowerPC with the PPC601, PPC604, and then the 740/750. The G series Macs came next (i.e. G3, G4 and G5). Finally, Apple moved back into the CISC world with the Intel x86 line in 2006 (first generation intel-based Macs). What will the future hold? Will Apple transition its' desktops back to RISC?

And what about Intel? Oddly enough, Intel already made the switch, and few noticed! Internally, the Intel chips are RISC (since Pentium II) and emulate CISC x86 instructions in order to run legacy software. Perhaps it is time for Apple to make another switch. Speculation that this will be done by 2020 has already been published. With the Microsoft/Qualcomm partnership moving Windows into the ARM processor, it is likely Apple will not be the only one embracing the ARM business model on the desktop [5].

4. The CORE Wars

Rather than the promotion of clock speed, the modern method for marketing CPUS appears to be about the number of CORES. The smaller footprint RISC architecture is able to budget a larger number of cores for the same transistor budget. Even better, it is possible to shutdown cores to conserve power.

AMD launched an 8-core Ryzen CPU last year and made it price competitive with Intel's 4-core CPUs. This drove both companies into a multi-core CPU race. Now the 2nd generation Ryzen (Threadripper) has 32 cores with 64 threads; beating out the Intel Core i9 with 18 cores (and 36 thread).

5. CPU Power Consumption and TDP

TDP is also a factor, as Intel Core i7 CPU is an active fan-cooled device, while the Apple A12X Bionic CPU has a passive thermal subsystem. The Geek benchmark indicates these two processors are close in performance (at least for a single core) [6]. As the number of cores increases in the AMD/INTEL CPUs, the TDP skyrockets; 250 W for the Ryzen Threadripper and 165 W for the Core i9-7980XE.

Mobile devices are powered by batteries and battery life is a significant feature. Also important is the amount of heat a device generates. Many-core CPUs may the future of mobile computing. Qualcomm has started sampling it 48-core 10nm processor [7]. Turning off cores may be an efficient way to throttle down power consumption and heat dissipation rather than throttling down the clock speed.

6. Conclusion

Clearly, multi-core processors have become the standard for delivering greater performance, improved performance per watt, and new capabilities across desktop, mobile, and server platforms. The trouble is, parallelization of sequential programs is still not easy. Many who would like a single application to run faster on multi-core systems are sure to be disappointed.

On the other hand, embarrassingly parallel algorithms and fine-grained parallelism enable massively parallel hardware to dominate. For real-time raytracing, computational fluid dynamics, n-body problems, video processing, etc. the many-core systems can dominate. NVIDIA now has a Titan Xp card with 3,840 CUDA cores running at 1.6 GHz and rated at a peak 12 TFLOPS performance. The newer Titan V has 5,120 CUDA cores with 110 TeraFLOPS if peak performance.

Where will the core race end? Will we have millions of cores? And how will they all be programmed?

I am reminded of a story told by Daniel Hills who was having lunch with Richard Feynman when he disclosed his idea to start a computer company that made computers with a million processors. Feynmans' reaction: "That is positively the dopiest idea I ever heard".

Perhaps Feynman knew, parallel programming is not easy. And we have seen the many core super computers before (IBM Blue Gene/P, Cray T3D, Pixel Machine, Mark IV Hypercube, GPGPUs, ILLIAC IV, etc.). They all had one thing in common, programming is hard, and parallel programming is really hard, and getting peak performance from parallel programs is super hard. Automatic parallel programming is the future of computer science and perhaps, it always will be.

Conflicts of Interest

The author has no conflict of interest.

References

1. [Statista] https://www.statista.com/statistics/330695/number-of-smartphone-user\s-worldwide/
2. [Pew] http://www.pewinternet.org/2015/10/29/technology-device-ownership-2015/
3. [Samsung] https://www.samsung.com/semiconductor/minisite/exynos/technology/custom-cpu/
4. [Reddit]https://www.reddit.com/r/apple/comments/9midcx/apple_really_undersold_the_a12_cpu_its_almost
5. [Verge] https://www.theverge.com/2018/4/3/17191986/apple-intel-cpu-processor-design-competition
6. [Geek] https://browser.geekbench.com/v4/cpu/10567048 vs https://browser.geekbench.com/macs/430
7. [AnandTech] https://www.anandtech.com/show/13597/just-when-you-thought-it-was-dead-qualcomm-centriq-arm-server-systems-spotted