Name: Jacob Branaugh **ECE 472 Final**

ID#: 931-724-634

In today's age, computers are ubiquitous. Most homes in the U.S. have at least one PC or laptop, single board computers are available for low prices, and microcontrollers can be found almost anywhere. But not all electronic devices are equal. Many different processor architectures exist and vary in different ways. In this course, we have been looking at and discussing the ARM, or Advanced RISC Machine, architecture. ARM is one of the most widely used architectures on the planet, comfortably dominating the mobile market [1]. In an attempt to gain a greater understanding of the ARM architecture, it will be compared and contrasted against the PowerPC architecture.

ARM, originally known as the Acorn RISC Machine, was first developed by Acorn Computers in 1985 [2]. While other companies such as Intel and AMD were developing larger scale chips, ARM was developing small scale simply due to start up budget issues. This early division from competing companies landed arm in its dominant position in the smaller scale market that it still holds today. ARM hit its stride in the early 90's, when it was approached by both TI and Samsung. As of 2013, ARM has reported production of 37 billion processors. ARM currently features the Cortex 32-bit architecture family, as well as the 64-bit ARMv8-A architecture as of 2011 [3]. PowerPC, or Performance Optimization With Enhanced RISC-Performance Computing, was initially created in 1991 by Apple, IBM, and Motorola jointly. PowerPC is a RISC, superscalar architecture that was intended to be both low cost and high performance [4]. The goal of PowerPC in the consumer market was an attempt to counter the popularity of Intel's CISC 80386, 80486, and upcoming Pentium processors. PowerPC had success in Apple machines, as well as in video game consoles of the 00's. Today, PowerPC exists as the Power ISA.

Both ARM and PowerPC are RISC architectures. RISC stands for Reduced Instruction Set Computing, and was designed with the idea in mind that simpler instruction are capable of providing higher performance through faster execution of instructions. ARM has many different ways it can address operations. ARM can address direct to memory, direct to a register, to an immediate, indirect to a register with optional offset and incrementing, and relative to the program counter [5]. ARM address length in most flavors of the architecture are 32 bits, while ARMv8-A is 64 bit. PowerPC, on the other hand, can only address relative to the program counter, relative to a base register, and relative to a base register with an additional register as an index [6]. PowerPC supports both 32 and 64 bit addresses. Both ARM and PowerPC can address using pages as small as 4KB.

The ARM architecture has 16 general purpose registers. However, not all are usable as general purpose registers. R13 contains the stack pointer, R14 contains the link register, and R15 contains the program counter. Outside of these three, registers R0 through R12 are available for use in a program. The PowerPC architecture has a total of 64 available registers, 32 of which being general purpose registers and 32 being floating point registers. ARM features three different eight stage pipelines. All pipelines have fetch 1, fetch 2, decode, and register read stages. Following these four, each has a different set of execution cycles. The first pipeline handles register to register operation, the second handles multiplication and accumulation, and the third handles loading and storing. Most ARM instruction only require one cycle to execute. PowerPC has a 6 stage pipeline, but some instruction may take up to 3 execute cycles [7]. When a pipeline is full, an instruction is being completed every cycle, so the number of stages is irrelevant. However, since ARM has fewer stages than PowerPC, it will not take as long to refill the pipeline after flush. ARM does not use microcode, which allows it to run much more efficiently. PowerPC, on the other hand, does use microcode. This adds an additional level of power to the architecture, at the price of microcode

being costly to execute.

As previously discussed, both ARM and PowerPC have internal registers. ARM processors have 16 general purpose registers, of which 13 are usable due to the stack pointer, link pointer, and program counter. PowerPC has a total of 64 registers, half being general purpose and half being floating point dedicated. While the first ARM processors did not have caches, modern ARM processors can have up to 64K L1 cache and up to 4M L2 cache depending on the processor core. The first PowerPC processor had a 32K L1 cache and support for an external L2 cache. Modern POWER processors used in servers and supercomputers can have up to 64K split L1 cache, 256K per core L2 cache, and 80M L3 cache [8]. ARM uses a two level page table for 4KB pages in 1MB sections, while PowerPC uses parallel lookup through 4 or 8 sets. This allows for much faster lookups, as a lookup hit in any of the sets will cause the search to end.

ARM has a few interesting features that are uncommon in other RISC architectures. First is the ability to perform shifts/rotates and arithmetic/logic/moving instructions as a single instruction. For example, performing the action of $a = a - 8 \times b$ would normally require multiplying b by 8, and then subtracting from a. In ARM, a single instruction can perform this, in that the normal subtraction instruction has a logical shift left by 3 instruction embedded within. Additionally, ARM's ability to use PC relative, pre-incrementing, and post-incrementing address modes is not seen elsewhere as often. The advantage of PC relative addressing is that the program doesn't have to be anchored anywhere specifically in memory. Without PC relative addressing, a jump must be performed by finding the address that the desired instruction corresponds to. With PC relative addressing, the address relative to the current location is all that is needed. So instead of jumping to 0x0034FA24 (for example), the jump is to current + 16 (for example). While PowerPC does not support these features, it does support a few unique features of its own. PowerPC was one of the earlier 64 bit RISC architectures available outside of the supercomputing world [9]. This allowed for more general purpose registers than competitors, as well as the possibility of more effective memory mapping. Another advantage of the PowerPC architecture is the fact that it is a superscalar architecture. What this means is that the processor is capable of executing two instructions simultaneously. In a pipelined scenario (assuming the pipeline is full), this means that instead of a single instruction being executed every cycle, there are two instructions being executed every cycle. This allows for, theoretically, a doubling of performance.

Part of the reason as to why ARM achieves the level of performance it does is the lack of microcode. Though microcode can be a powerful tool, it is costly (in terms of resources) to implement. Due to its absence, ARM can achieve much lower power draws, while keeping performance up by focusing on caching for performance boosts. Additionally, ARM gets a performance boost due to the fact that is has different pipelines for different types of instructions. PowerPC gets a performance boost largely in part from the fact that it is a superscalar architecture. The fact that instructions can be executed in direct parallel means an increase in performance, as more tasks are being performed per given time. However, this also means that even more time is wasted on a pipeline flush, as it is dumping and then refilling twice the amount of data. It also leaves the processor much more open to pipeline hazards. An additional performance boost is gained from the way in which memory is mapped. Since PowerPC implements parallel lookup, the average time taken to perform a lookup is lower due to the fact that a hit in any set causes successful lookup. This is especially important in high performance computing, where lookups are constantly being performed.

Though ARM is a powerful architecture, its power lies in the fact that it performs well considering how low power it is. In terms of raw computing power, the PowerPC architecture is the clear winner. Neither architecture is better than the other, and neither architecture is really better or

worse than any competing architecture. The effectiveness of an architecture ultimately boils down to the application. In a low power application, ARM may not even be as effective as some other, lower power architecture. Likewise, PowerPC may not be the most effective architecture for a high performance computing application. Understanding the pros and cons of various architectures is the key to selecting the one that fits the problem best and to creating the best product.

References

[1] Bent, Kristin. *ARM Snags 95 Percent Of Smartphone Market, Eyes New Areas For Growth.* CRN. N.p., 165 July 2012. Web. 09 Dec. 2013. http://www.crn.com/news/components-peripherals/240003811/arm-snags-95-percent-of-smartphone-market-eyes-new-areas-for-growth.htm.

[2] Levy, Markus. *The History of The ARM Architecture: From Inception to IPO.* Convergence Promotions, n.d. Web. 9 Dec. 2013. http://www.reds.ch/share/cours/ReCo/documents/TheHistoryOfTheArmArchitecture.pdf.

[3] Grisenthwaite, Richard (2011). *ARMv8-A Technology Preview.* 9 Dec. 2013.

[4] *Tech Files Columns, 1987-1900.* 9 Dec. 2013.

[5] Regina, University of. *ARM Addressing Modes.* Addressing Lab. Department of Computer Science, n.d. Web. 09 Dec. 2013. http://www.cs.uregina.ca/Links/class-info/301/ARM-addressing/lecture.html.

[6] Physinfo. *Addressing Modes.* Addressing Modes. N.p., n.d. Web. 09 Dec. 2013. http://physinfo.ulb.ac.be/divers_html/powerpc_programming_info/intro_to_ppc/ppc3_software2.html.

[7] Abramson, David. *Power PC Pipeline Stages.* Power PC Pipeline Stages. University of Montana, 2000. Web. 09 Dec. 2013. http://www.csse.monash.edu.au/ davida/teaching/cse3304/Web/Chapter5/sld036.htm.

[8] *IBM POWER Microprocessors.* Wikipedia. Wikimedia Foundation, 12 June 2013. Web. 09 Dec. 2013. http://en.wikipedia.org/wiki/IBM_POWER_microprocessors.

[9] J. W. Bishop, et al.: *PowerPC AS A10 64-bit RISC microprocessor*, IBM Journal of Research and Development, Volume 40, Number 4, July 1996, IBM Corporation.