# Contrasting Processors: ARM vs. Intel

Technical Article for COMP-2660, Computer Architecture II: Micro Program, University of Windsor

Dennis Dao, Student ID:110010116

Mar 1, 2021

"I confirm that I will keep the content of this assignment confidential. I confirm that I have not received any unauthorized assistance in preparing for or writing this assignment. I acknowledge that a mark of 0 may be assigned for copied work."

# Contrasting Processors: ARM vs. Intel

Dennis Dao
*School of Computer Science*
*University of Windsor*
Windsor, ON, Canada
dao114@uwindsor.ca

***Abstract*—The central processing unit (CPU) is the brains of the computer, working when given the instruction set to move data between registers and memory. An ARM processor or an Intel processor can be used to complete the operation, both with a different kind of instruction set. ARM processors run on a RISC architecture and the Intel processors run on a CISC architecture. With these differences, the overall performance, power use, and energy use of the devices using them will be affected. These differences are explored, making one processor preferable over the other depending on the intended use, whether it be performing numerical operations or a low-energy use computing. A discussion is written on the thoughts about the differences of these processors and with what has been done in the past and what could be done in the future.**

***Index Terms*—ARM processor, x86, central processing unit (CPU), reduced instruction set computing (RISC), complex instruction set computing (CISC), big.LITTLE, thermal design power (TDP)**

## I. Introduction

Without the central processing unit (CPU), the computer would not know how to execute instructions and commands given by the user. The CPU is the brain of the computer and works when the instruction set is given to move data between the registers and the memory to perform numerical operations. With code from applications on devices such as smartphones, it is compiled for specific instruction sets to run correctly. It is then further decoded into microcode ops within the central processing unit, where it needs silicon space and power. These operations can affect the overall performance of the device using the processor and energy use. Like human brains, not all processor families will run similarly. Intel processors and ARM processors are examples of two different processor families that have different performance levels and features. Intel processors are powerhouses and can be found in high-end desktop PCs and laptops while ARM processors are energy efficient and can be found in devices where a constant power source is not possible, such as smartphones and mobile devices. Differences can be found between the ARM processor family and the Intel processor family, affecting the performance and use of the devices using them.

## II. Importance

Why should the differences between the ARM processors and the Intel processors be discussed and why is it important? With devices evolving over time, processors change over time for new features and performance improvements. These improvements allow faster computing and better experience using the processors. Depending on the instructions performed and the device used, the instruction sets of a processor should be considered. If low power use is required, then the instruction set should be kept simple. If higher performance is needed, a complex instruction set and hardware are needed, coming at the cost of power. In order to compute data for any purpose effectively, the speed at the processor executes instructions matters. When consumers buy a CPU, they may have the performance of the CPU in mind [1]. The difference between the performance of processors and the software, hardware, or application they intend to use can influence the processor chosen to perform the task.

### A. *Past*

"In 2006, Apple switched its Mac computers and software from PowerPC chips to Intel's x86 platform," [2] with the first Intel Mac released in January 2006. Apple has used Intel processors for the past 15 years and Microsoft has used Intel processors for the past 30 years. "In 1990 about 20 million PCs were shipped, and shipments increased by about 15% every year until PC shipments reached their zenith in 2011" [3]. Intel and Microsoft improved the performance and price-performance of Intel chips every 18 months, and new releases of Microsoft Windows 3-4 years. Together, they "increased the functions and speed of servers against the reduced instruction-set computing or RISC vendors from the bottom up" [3].

### B. *Present and Future*

Several companies today have swapped from Intel processors to ARM processors, promising users better features in the devices using the new processors. For example, Apple began to switch to ARM processors in June 2020. Previously, Apple had been using the Intel Inside processors [4]. By switching, this allowed the company greater control over the hardware and software. This switch promised users powerful devices that are efficient, have a longer battery life, and the same instant-on features as a phone [4]. Apple says that switching to its own chips "not only opened up performance but also new, innovative technologies such as the AI neural engine, high-performance graphics and best-in-class security" [4]. Emulation technology known as Rosetta 2 will "make sure that existing apps that haven't been updated will run on the new Apple ARM-chips from the beginning" [4]. Another example is Microsoft, where they used ARM chips in their Surface Windows PCs, recently with the Surface Pro X, having a "a custom-designed chip made by Microsoft with mobile chip-maker Qualcomm" [4]. However, Microsoft struggles to bring third-party software to ARM-based PCs [4]. Within both companies, their applications are being ported for the switch from Intel processors to ARM processors. These differences bring transitions between the two processors, because of lower costs, lower power requirements, a common platform, faster software adoption and hardware innovations for higher productivity for end-users [3].

## III. Technological Details

Both the Intel processors and the ARM processors have distinct differences that make one processor preferable to use over the other. Depending on the device used, processors are designed with the performance, hardware, and power usage of the device in mind.

### A. ARM Processors

The ARM processor families are made to target energy efficiency with a lower power, instruction set, and hardware. ARM processors are part of the reduced instruction set computing (RISC) architecture, where one cycle is used to execute a command, reducing the number of functions. Because of the single-cycle computing set, less battery life is used, and the processor operates at a lower temperature [5]. The chips require fewer transistors to function, also contributing to power and heat reduction [6]. For performance features, ARM relies heavily on software [5]. With a simplified construction, it is the best choice for low-power devices [2]. Although ARM processors are slower than Intel processors, these processors are designed to work with low power consumption. Most of the time, ARM processors are more mobile-friendly than Intel processors. The architecture features heterogeneous compute, where the architecture is made with different CPU parts for improved efficiency [7]. The idea was first attempted in 2011 with the big.LITTLE using the big Cortex-A15 core and the little Cortex-A7 core. This idea was further built on in 2017 using the DynamIQ (Figure 1) and the ARMAv8.2 architecture, "allowing different CPUs to sit in the same cluster, sharing memory resources for far more efficient processing" [7]. Doing so allowed a 2+6 CPU design, commonly found in mid-range chips.

Two execution states, AArch32 and AArch64 allow a clean 64-bit implementation, the states running 32 and 64 bits, respectively. The processor can swap from one state to the other during normal execution. The decoder, then, will not need to maintain compatibility with the 32-bit era but the whole processor remains compatible. Intel-based devices can run the full range of Android apps, even ones meant for ARM architecture. However, ARM-specific code must be translated before running on Intel devices. This requires time and energy, which can affect battery life and overall performance. Because of this, most ARM processors can be found in mobile devices such as laptops and smartphones, devices that do not always have a constant power source. In selling processors, ARM processors sell the licenses, allowing companies to customize the processors to fit their device and performance requirements.

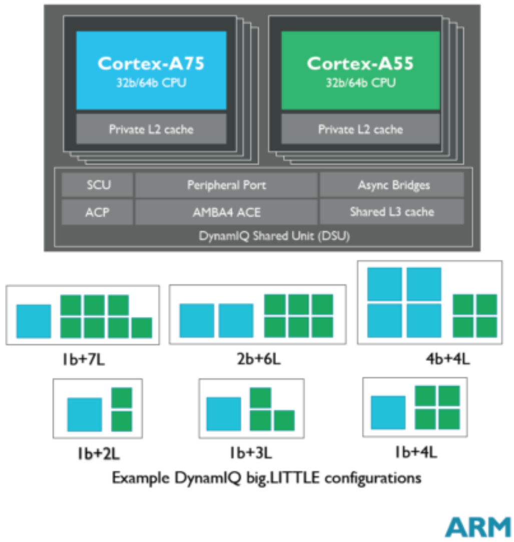


Fig. 1. DYNAMIQ.big.LITTLE configurations; Source: Adapted from [7]

### B. Intel Processors

The Intel processor families are made to target peak performance and can be found in devices where heat dissipation, space, and power usage are not as concerning when compared to a mobile device. To target peak performance, their instruction set is a complex instruction set computing (CISC) architecture, where multiple operations can be done for better performance but at the cost of more power use to decode the instructions. These complex instructions require more time cycles to complete. The command codes are simple but use several cycles to perform these command codes. In the CISC chips, transistors are needed to store these instructions and the logic, requiring more power usage and heat production [6]. For better performance, these processors rely on hardware. Together, this allows faster and more powerful computing than the ARM processors. However, in devices such as laptops, Intel processors trail behind ARM processors in battery life, although the gap in battery life is not large.

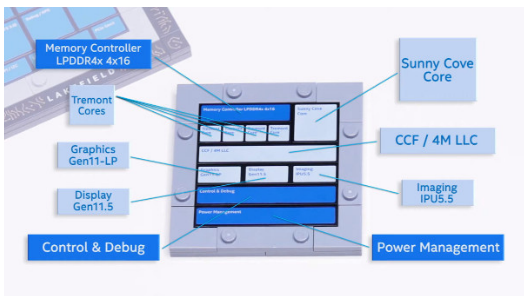


Fig. 2. Intel Lakefield design; Source: Adapted from [7]

The Intel Lakefield (Figure 2) competes with the heterogeneous compute from ARM, having similar features. Although the 10nm Lakefield card combines a high-performance Sunny Cove Core with four power-efficient Tremont cores, it has a 7W thermal design power (TDP), a power usage too high for smartphones. In selling processors, Intel maintains the manufacturing of their processors from start to end and directly sells their chips.

### C. Comparing the Processors

TABLE I
INTEL PURLEY VS. CENTRIQ

| | **Processors** | |
|---|---|---|
| **Feature** | **Intel Purley** | **Centriq (Qualcomm)** |
| Process | 14nm | 10nm |
| Issue | 8 µops/cycle | 8 instructions/cycle |
| Dispatch | 4 µops/cycle | 4 instructions/cycle |
| # Cores | 10 x 2S + HT (40 threads) | 46 |
| Clock Speed | 2.2 GHz (3.1 GHz turbo) | 2.5 GHz |
| TDP | 170W (85W x 2S) | 120W |

[a]Source: Adapted from [8]

In the comparison from Table 1, the Intel processor is much faster when in turbo and powerful but draws more power and has a much higher TDP. The Qualcomm can do almost the same amount of instructions per cycle as the Intel processor but with a lower TDP and a higher base clock speed [8].

Concerning these processors, in general, can be seen in Table 2.

TABLE II
ARM ARCHITECTURE VS. X86 ARCHITECTURE

| | ***ARM*** | ***x86 (Intel)*** |
|---|---|---|
| Instruction Size | Unified size, execution is done within single clock cycle | Size varies on complexity |
| Fetching and Decoding | Simple fetching, accessing opcodes and operands is simulataneously possible | Complex fetching: different formats, several addressing modes |
| Power Use | Simplified design of control unit with better power efficiency | Complex design causes high level of power consumption from control unit |
| Program Size | Specific functions requires simple instructions, causing large sizes | Similar function demands need less instructions, causing smaller sizes |
| # Cores | 10 x 2S + HT (40 threads) | 46 |

[a]Source: Adapted from [9]

In [9], an evaluation for average cycles per instruction, L2 cache miss rate, total energy, and throughput using the Gem5 simulator was held. It was concluded that ARM processors outperform x86 in most cases unless the CPU model is out-of-order. "ARM is always more energy efficient than x86 except for the basicmath benchmark when the CPU model is out-of-order" [9]. To determine this, a comparison of applications running on x86 and ARM architectures was conducted. ARM devices consumed 3-4 times less energy than the x86, but the x86 performs well in floating point computation.

From [10], a question that was asked was: "Are ARM and x86 code densities similar?" It was found a similar code density did not affect the instruction set architecture. The formatting of RISC and CISC code can be seen in Figure 3, matching with the contents of Table 3.

TABLE III
RISC AND CISC TRENDS

| | ***ARM / RISC*** | ***x86 / CISC*** |
|---|---|---|
| Format | Single cycle<br>Simple operations<br>Single function operations | Simple encoding<br>Fixed length instructions |
| Operations | Complex<br>Multi-cycle instructions<br>Encryption<br>String manipulation<br>Transcendentals | Variable length instructions<br>Short common instances<br>Longer special instances<br>1B to 16B long |
| Operands | Registers<br>Immediates<br>Few addressing modes<br>16 general purpose registers | Memory<br>Registers<br>Immediates<br>Many addressing modes<br>Eight 32B and six 16B registers |
| Program Size | Specific functions requires simple instructions, causing large sizes | Similar function demands need less instructions, causing smaller sizes |
| # Cores | 10 x 2S + HT (40 threads) | 46 |
| Clock Speed | 2.2 GHz (3.1 GHz turbo) | 2.5 GHz |
| TDP | 170W (85W x 2S) | 120W |

[a]Source: Adapted from [10]

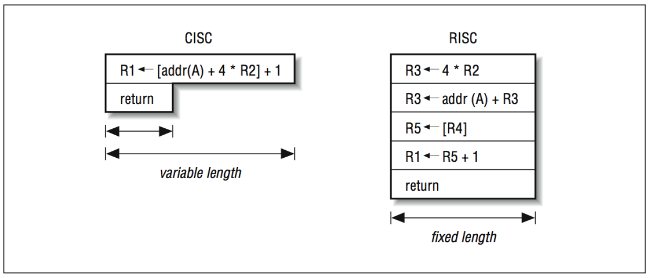


Fig. 3. CISC and RISC code comparisons; Source: Adapted from [11]

The steps for RISC assembly code are broken down to the basic steps while CISC assembly code can perform multiple commands at once [6]. For example, the command codes to multiply two numbers in CISC can be written as MULT 2,3. On the other hand, code to multiply two numbers in RISC is a sequence of instructions. The instruction set uses the LOAD command to load the numbers from the registers into the processor. Then, the numbers are multiplied with the PROD (product) command. Lastly, the result of the multiplication must be stored back into the correct register with the STOR command. With the CISC assembly code, it combines all the RISC assembly code into one command code that can be called by the processor.

Lastly, there are cases where software needs to be able to run on both the ARM devices and the Intel devices. Compiling the native software for both can be done with code emulation. Although less efficient and degrades the performance of the software compared to native apps, it is possible to have good emulation ensuring they work [7].

## IV. Discussion

In discussing the differences of ARM and Intel processors, it is true that not all processors will run the same – each processor has its own perks and drawbacks when compared to one another. With each new processor, features are changed, and the performance is improved. Each processor is developed differently, with the device that it could end up with in mind. In analyzing the difference between the two types of processors, the instruction set is important in discussing the differences. Like most programming problems, there is more than one way to write and format the solution to a problem. With these processors, the same can be said. Functions can be made to modularize the operations, requiring more logic, seen in the CISC computing sets. True to its name, the complex instruction set computing performs multiple operations at once, using functions. These functions end up being used in large number computing and in applications such as graphics. Another way is that the operations can be done at their most basic step, requiring no functions, seen in the RISC computing sets. With less functions, the complexity of the logic and instructions are reduced. There is less complexity, allowing more energy efficiency, key for devices such as laptops and smartphones. The different ways in solving these problems however does affect how much time and space it uses to execute the operation. However, like any other computer program in any high-level language, the correct logic, or syntax must be used when using functions. The more logic and instructions there are, the more transistors are needed for the processor to store and complete the operation correctly. With more transistors, more power is needed, and more heat is created during the operation. Depending on the device in mind, the increased heat and power draw can be a drawback.

I believe that ARM processors and Intel processors will always be different because one type of processors cannot possibly cater to all our data computing needs and energy uses. A feature of the processor is sacrificed in order to improve the other feature, seen with the ARM and Intel processors. In order to reach peak performance, more energy must be taken in which creates more heat. In order to make an energy efficient machine, performance and speed must be sacrificed in order to make an operating system running at a low temperature. This results in one processor preferable for one situation and another processor preferable for another. In giving these processors to the public, I feel ARM has the right approach in selling the licenses and allowing companies to customize how they need it to fit their devices. This allows more energy-efficient and potentially powerful devices, compared to the restrictions on the Intel processors that are manufactured from start to finish.

## Conclusion

Thus, ARM processors and Intel processors have key differences, affecting the devices that use them. The key difference with both processors is the instruction sets that they use. The instruction sets end up affecting the rest of the architecture of the processor. ARM processors use the reduced instruction set computing architecture, executing instructions within a single cycle. The single-cycle computing set requires less logic to be held so fewer transistors are needed. With fewer transistors, less power is drawn, allowing energy efficiency and lower operating temperature. In contrast, the Intel processors use the complex instruction set computing architecture, executing complex instructions within multiple cycles to complete. With modularized functions, more transistors are required to hold the instructions and logic, so more power is drawn, and more heat is produced. This all causes the operands, operations, and format of the codes to differ between processors. However, these differences are not bad because one processor type cannot cater to all our needs. These differences make one processor always preferable over another processor, so it all depends on the situation that the processor will be used for.

## References


[1] "CPU Comparison: X86 vs ARM - Will Intel i9 9900K Stay Atop?," Fossbytes, 22-Apr-2019. [Online]. Available: https://fossbytes.com/cpu-comparison-x86-arm-cpu-benchmark/. [Accessed: 15-Feb-2021].

[2] B. Clark, "Intel vs. ARM Macs: Whats the difference and which should you buy?," LaptopMag, 07-Nov-2020. [Online]. Available: https://www.laptopmag.com/news/intel-vs-arm-macs. [Accessed: 15-Feb-2021].

[3] D. Floyer, "Exiting x86: Why Apple and Microsoft are embracing the Arm-based PC," SiliconANGLE, 29-Jun-2020. [Online]. Available: https://siliconangle.com/2020/06/26/exiting-x86-apple-microsoft-embracing-arm-based-pc/. [Accessed: 16-Feb-2021].

[4] "Apple ditches Intel for ARM processors in Mac computers with Big Sur," The Guardian, 23-Jun-2020. [Online]. Available: https://www.theguardian.com/technology/2020/jun/22/apple-ditches-intel-for-arm-processors-in-big-sur-computers. [Accessed: 15-Feb-2021].

[5] D. Graham-Smith, "ARM vs Intel Processors: What's the Difference?," Alphr, 17-Dec-2020. [Online]. Available: https://www.alphr.com/features/390064/arm-vs-intel-processors-what-s-the-difference/#: :text=Intel processors (commonly referred to,complex instruction with several cycles. [Accessed: 15-Feb-2021].

[6] A. Fox, "The Differences Between ARM and Intel," Make Tech Easier, 12-Oct-2020. [Online]. Available: https://www.maketecheasier.com/differences-between-arm-and-intel/. [Accessed: 15-Feb-2021].

[7] Bilal, "Arm vs x86: Instruction sets, architecture, and all key differences explained," Business, 29-Jun-2020. [Online]. Available: https://ipsnews.net/business/2020/06/29/arm-vs-x86-instruction-sets-architecture-and-all-key-differences-explained/. [Accessed: 15-Feb-2021].

[8] V. Krasnov, "ARM Takes Wing: Qualcomm vs. Intel CPU comparison," The Cloudflare Blog, 08-Nov-2017. [Online]. Available: https://blog.cloudflare.com/arm-takes-wing/. [Accessed: 16-Feb-2021].

[9] A. Abudaqa, T. Al-Kharoubi, M. Mudawar, A Kobilica, "Simulation of ARM and x86 Microprocessors Using Inorder and Out-of-order CPU Models with Gem5 Simulator", 5th International Conference on Electrical and Electronics Engineering, 978-1-5386-6392-9/18, 2018.

[10] E. Blem, J. Menon, and K. Sankaralingam, "Power struggles: Revisiting the RISC vs. CISC debate on contemporary ARM and x86 architectures," in Proc. IEEE Int. Symp. High Perform. Comput. Archit. (HPCA), Shenzhen, China, Feb. 2013, pp. 1–12.

[11] E. Engheim, "What Does RISC and CISC Mean in 2020," Medium, 24-Jan-2021. [Online]. Available: https://medium.com/swlh/what-does-risc-and-cisc-mean-in-2020-7b4d42c9a9de. [Accessed: 15-Feb-2021].