

# A Comparative Study on the Performance of 64-bit ARM Processors

<https://doi.org/10.3991/ijim.v17i13.39395>

Ala'a Al-Shaikh<sup>1</sup>(✉), Ameen Shaheen<sup>2</sup>, Mohammed Rasmi Al-Mousa<sup>1</sup>,  
Khaled Alqawasmi<sup>1</sup>, Ala'a Saeb Al Sherideh<sup>1</sup>, Hebatullah Khattab<sup>1</sup>

<sup>1</sup>Faculty of Information Technology, Zarqa University, Zarqa, Jordan

<sup>2</sup>Khawarizmi University Technical College, Amma, Jordan

ashaikh@zu.edu.jo

**Abstract**—Mobile devices are playing an important role in our daily lives. Nowadays, mobile devices are not only phones to call and text, but they are also smart devices that enable users to do almost any task that could be done on a regular PC. At the heart of the design of smartphones, there lies the processor to which almost all the development in the smartphone arena is attributed. Recently, ARM processors are among the most prominent processors used in mobile devices, smartphones, and embedded systems. This paper conducts an experimental comparative study of ARM 64-bit processors in terms of performance and their effect on power consumption, CPU temperature, and battery temperature. We use a number of well-known benchmarks to evaluate those characteristics of three smartphones, namely, Snapdragon 778G+, Exynos 1280 and HiSilicon Kirin 980. Those smartphones are all equipped with ARM 64-bit processors. Our results reveal that none of the three-selected smartphones was the best in all characteristics; each has superiority amongst others in certain characteristics and is dominated by others in other characteristics.

**Keywords**—ARM 64-bit, mobile processors, smartphones, RISC, SoC.

## 1 Introduction

Mobile devices and IoT are ubiquitously playing an important role in our daily lives [1, 2, 3]. In the recent years, mobile devices are increasingly spreading between children, adults, and elders [4, 5, 6]. The widespread of smartphones and their applications [7] paved the way to new technologies to rise in order to support the new trends in computation, infrastructure, and storage, such as distributed systems [8], fog computing [9], and cloud computing which became ubiquitous and inseparable from smartphones [10, 11].

The Advanced RISC Machine (ARM) is a microprocessor technology, as well as a family of processors based on the Reduced Instruction Set Computers (RISC) architecture. This kind of processors is used heavily in smart systems and electronic devices due to the prevailing features it comprises, such as, (1) high performance, (2) low cost, and (3) low power consumption [12, 13].

ARM was originally designed to be embedded in a System-on-Chip (SoC) device [14]. An SoC comprises on one chip the following components: (1) several processor cores, (2) specialized hardware accelerators, and (3) I/O interfaces [15, 16].

The Instruction Set Architecture (ISA) of ARM is a RISC-style architecture [12, 17]. Compared to CISC, RISC has a smaller number of instructions and addressing modes [17].

ARM inherently implements a load/store architecture [18], where all operations are executed on the registers of the Arithmetic and Logic Unit (ALU) rather than on memory; this implies that all operands must be loaded from memory to ALU registers, on demand, in order for the execution to take place and the results could be stored back, on demand, to memory locations.

Nevertheless, RISC is disadvantageous in terms of instruction length when compared to CISC. While the instruction length is fixed in RISC architectures, CISC processors have variable-length instruction length [19]. In fact, code density is an important parameter in embedded systems as it has a remarkable influence on power consumption and performance [20]. The reason behind that is that codes with higher densities occupy less memory, which results in less power consumption [19]. One of the solutions to this problem was the introduction of Thumb 16-bit Instruction Set by ARM, which compresses the 32-bit ARM instructions into 16-bit instructions. Those instructions are decompressed before they enter the pipeline [21].

As power consumption is the factor of greatest concern in designing embedded systems [22]. This motivates researchers to study and investigate power consumption in those systems. Such systems must use optimized systems in order to achieve high throughput accompanied with low power consumption [23].

The remainder of this paper is organized as follows: we will provide a background about computer architectures and some related work in Sect. 2. Then, in Sect. 3, we introduce some details about the ARM architecture. A comparison between the three processors that were used in this paper is provided in Sect. 4. Section 5 shows our experimental results. Finally, we make our conclusions and suggest the future work in Sect. 6.

## **2 Related work**

The Von Neumann architecture is the basic and general computer architecture [24]. Basically, the Von Neumann computer consists of three components, namely: (1) Central Processing Unit (CPU), (2) memory, and (3) I/O interfaces [16]. The components of the Von Neumann architecture are shown in Figure 1.

One of the drawbacks of the Von Neumann architecture is the Memory Wall problem which is characterized by the existing of a gap between the CPU speed and memory latency which causes a system bottleneck [25]. Adding levels of cache to the system is one solution to the memory wall problem [26].

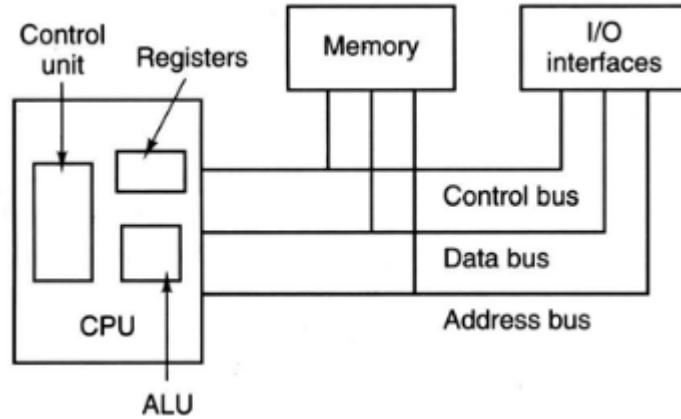


Fig. 1. Components of the Von Neumann Architecture [27]

Harvard architecture is another type of computer architectures that exhibits an enhancement to the Von Neumann architecture. Harvard architecture is common in embedded systems which require low power consumption and higher levels of reliability [28]. In Harvard architecture, there are two distinct memories, one for the program and the other is for the data [29]. Figure 2 illustrates the main difference between both architectures.

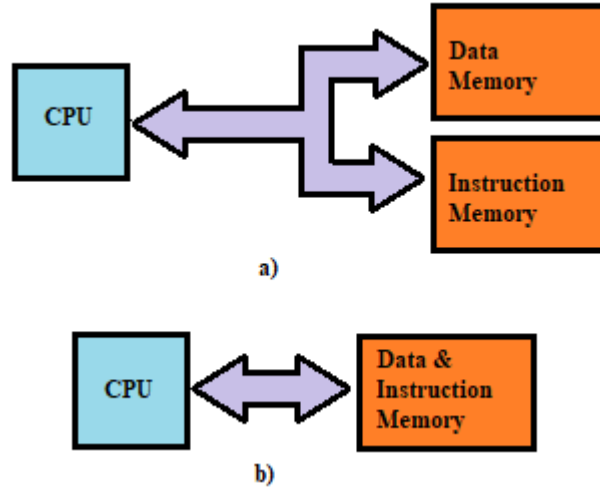


Fig. 2. a) Harvard Architecture, b) Von Neumann Architecture

As shown in Figure 2, Harvard architecture implements separate paths (buses) for each memory, one bus for the data memory and another bus for the instruction memory. This solves the bottleneck problem in Von Neumann architecture in which the same path is used for both data and instructions. Consequently, the performance is enhanced.

Pipelining enables concurrent execution of multiple instructions by overlapping instruction execution [30]. Instruction-Level Parallelism (ILP) refers to the technique in which instruction execution is overlapped [31]. Consequently, instructions are executed in parallel, which speeds up execution [32].

Conventionally, there are 5 pipeline stages, namely, (1) Fetch, (2) Decode, (3) Execute, (4) Memory, and (5) Write Back [33]. The number of stages of a pipeline are referred to as the depth of the pipeline [34].

In deeper pipelines, i.e., pipelines with greater number of steps, a given step can be implemented with simpler circuitry, which results in a faster processor clock [35].

A compiler is a kind of system software that is responsible for translating programs written in different programming languages to machine code. Programs are executed either:

- In-order: instructions are issued and executed in the order they appear in the program. One disadvantage of this execution model is the dependency between instructions, i.e., if one instruction stalls, then all instructions that are dependent on the stalled instruction will also stall, which will result in pending program execution until the stalled instruction is released [34].
- Out-of-order: aka. dynamic execution [36] or dynamic scheduling [37], a technique, or execution model, used to improve performance [38]. Instructions are fetched in order. They also complete in order. But their execution may not be in order, and hence the name out-of-order execution [39].

It is noteworthy that dynamic scheduling adds extra work to static scheduling, i.e., compile-time scheduling, which in turn affects power consumption remarkably [37]. Nevertheless, out-of-order execution is significantly faster and requires more core logic, smarter CPU architecture and more registers to deal with.

Tiwari et al. [40] analyzed the performance of 64-bit ARM processors that are installed on a cluster of computers in terms of performance and energy efficiency. The cluster is an ARM-based cluster that is used for High-Performance Computing (HPC). The performance was also compared with the performance of an x86 Intel Ivy Bridge system. The Energy-Delay Product (EDP) was computed as the product of energy by performance. The results show that the higher the number of cores is, the more efficiency is obtained.

Paunski and Angelov [41] studied the performance and power consumption of single board computers with respect to their engagement in the educational service robots. They conducted their work on a number of processors including the Raspberry Pi 3B+ which an ARM-based processor. The results revealed that although the Raspberry Pi processor showed the highest levels of power consumption, still it is the number 1 choice in robotics due to the large area of community, support, and packages availability.

Taffoni et al. [42] studied how energy consumption is affected by computation on CPUs and Graphics Processing Units (GPUs) on ARM MPSoC platforms. They presented a comparison between CPUs and GPUs on SoCs in terms of power efficiency. Similarly, Qasaimeh et al. [43] compared the performance of three hardware accelerators for embedded vision applications. Also, the performance of ARM processors on

Deep Learning was investigated by Dolz et al. [44]. ARM Cortex-A57 and Cortex-A78AE CPUs were studied among other processors.

The emphasis in the work that was conducted by Marantos et al. [45] was on designing a tool that allows performance and energy consumption estimation of applications that run on embedded devices. They proposed a framework that is based on static analysis.

Eum et al. [46] studied the ARIA block cipher algorithm on ARMv8 processors. They compared their work to other work in the literature and found that their results reveal 8.73x performance improvements using ARM processors.

The work that was conducted by Chheda et al [47] targeted the performance of machine learning training and inference on 64-bit ARM processors. The work concluded that the A64-bit ARM processor can be used for workload scaling in machine learning.

### 3 ARM architecture

ARM microprocessors have many generations:

- The 32-bit version of processors which comprises six main processor families: ARM7TDMI, ARM9TDMI, ARM7EJ, ARM9E, ARM10E and ARM Cortex.
- The 64-bit version such as the ARMv8-A processor.

For applications where the code density is a priority and high-volume applications with memory restrictions [48], ARM introduced the Thumb architecture. Thumb is a 16-bit architecture; it is a 16-bit instruction set that is a subset of the 32-bit instruction set. Statistically, the size of code written using the Thumb instruction set is 65% smaller than the code written using regular 32-bit instruction set. Additionally, the performance of such code is 160% higher when running from a 16-bit memory [49]. This makes the Thumb instruction set ideal for the situations that imply restrictions on the memory. As well, the higher code density it provides makes it ideal for systems that require low power consumption.

Some ARM processors are based on the Von Neumann architecture while others are based on the Harvard architecture. However, there are some differences between processors of the same family. For instance, in ARM7TDMI the depth of the pipeline was 3 stages, and the architecture was a Von Neumann computer architecture [49]. As an upgrade, the ARM9TDMI used a deeper pipeline with 5 stages and moved to the Harvard architecture which incurs improvement over the traditional Von Neumann architecture [50]. However, both processors are 32-bit processors [49, 50].

ARMv7 defines three architecture profiles [38]:

- Application Profile (A-Profile): supports Virtual Memory System Architecture (VMSA) based on a Memory Management Unit (MMU). This profile can also be referred to as VMSAv7 [38]. It is designed for high-performance systems that can run operating systems [51].
- Real-time Profile (R-profile): supports Protected Memory System Architecture (PMSA) based on a Memory Protection Unit (MPU). It is also known as PMSAv7

[38]. This profile is designed for systems that require deterministic timing and low interrupt latency [51].

- Microcontroller Profile (M-profile): is a variant of the PMSAv7, designed for low latency interrupt processing [38].

For real-time systems, there is no need for the processor to be equipped with a full-fledged MMU; a PMU would be sufficient, as it only performs the required protection operations and does not perform anything related to memory management and address translation [52].

The migration from 32-bit to 64-bit architecture is fundamental to many recent computing problems. Reasons to migrate to higher word width may include the capability to process data volumes larger than the capacity of a given CPU, i.e., the word width of the data to be processed could be larger than that of the CPU which prevents the CPU from processing that data and makes migration a necessity. Furthermore, it could be a form of Single-Instruction Multiple-Data (SIMD) processing, which is one kind of parallel processing. Multiple instructions, or event data items can be fetched to the CPU simultaneously if the word width of the data to be transferred is less than the bus width [53].

Thus, migration to 64-bit architectures doubles the speed and enables the processor to access more physical memory beyond 4GB or higher of physical memory [54], which in turn increases the computational capabilities and achieves higher performance.

ARMv8 is a 64-bit version of ARM with support for backward compatibility for 32-bit ARMv7 programs and applications. The migration also includes deeper pipelines with out-of-order, speculative, and superscalar execution [54].

ARM microprocessor designs benefit from the big.LITTLE architecture, aka. Heterogenous Computing architecture, in which a microprocessor comprises a number of high-performance *big* cores and a number of low-power *LITTLE* cores [55]. The objective is to create a multi-core processor that can achieve: (1) high performance, and (2) low power consumption; actually, these are the two constraints of mobile systems design [56]. Switching between the *big* and *LITTLE* cores is done by means of the inter-cluster core switching, aka. big.LITTLE Switching [55].

Figure 3 shows a schematic diagram for the big.LITTLE architecture by ARM.

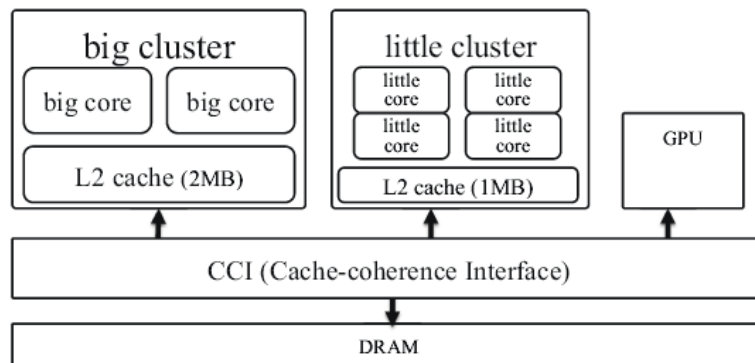


Fig. 3. ARM big.LITTLE (Heterogenous Computing) Architecture [55]

## 4 Processor architecture comparison

In the following we introduce three types of processors from different vendors. All of them contain several ARM cores. The processors are: (1) Snapdragon 778G+ 5G Mobile Platform, (2) Exynos 1280, and (3) Hisilicon Kirin 980.

### 4.1 Snapdragon 778G+ 5G Mobile Platform

Snapdragon 778G+ 5G Mobile Platform comprises a Qualcomm Kryo 670 CPU which is an octa-core, i.e., eight-core, 64-bit processor with an up to 2.7 GHz clock speed [57]. The Kryo 670 CPU supports the ARMv8 Cortex technology [58]. The Kryo 670 CPU comprises the following set of cores [57, 58]:

1. Kryo Gold plus, which is a Cortex-A78 based core with speed up to 2.7 GHz.
2. Kryo Gold, which comprises three Cortex-A78 based cores with 2.4 GHz speed each.
3. Kryo Silver, which consists of four Cortex-A55 based cores with 1.9 GHz speed.

The Snapdragon 778G+ SoC leverages the 5-nm FinFET manufacturing process [57]. The Fin-FET technology aims at minimizing the number of transistors that are used in the chip, minimizing the risk of leakage by adding more gates to the transistors, achieving higher performance, and consuming less power [59].

The big core, Cortex-A78 is a high-performance low-power, it has a superscalar variable-length pipeline [60]. On the other hand, the LITTLE Cortex-A55 is a low-power processor with an 8-stage in-order pipeline with symmetric dual issue for most instructions [61].

### 4.2 Exynos 1280

Exynos 1280 is an octa-core 64-bit processor that leverages the 5-nm FinFET manufacturing process [62]. Two of its cores are big ARM CortexA-78 cores, each with a clock speed up to 2.4 GHz, used for rigorous tasks. The other six cores are LITTLE ARM Cortex-A55 cores, each with a clock speed up to 2.0 GHz used for small tasks [63].

Both Exynos 1280 and Snapdragon 778G+ have the same number of cores, as well as similar architecture, which implies that both process have the same pipeline architecture and depth.

### 4.3 Hisilicon Kirin 980

Huawei HiSilicon Kirin 980 is a 64-bit octa-core processor that comprises two Cortex-A76 cores clocked at 2.6 GHz, two Cortex-A76 cores clocked at 1.92 GHz, and four Cortex-A55 cores clocked at 1.8 GHz [64].

Table 1 summarizes the characteristics of the three-selected processors.

**Table 1.** Summary of the Specifications of the Three-Selected Processors

Factor / Brand	Snapdragon 778G+	Exynos 1280	Kirin 980
Cores	8		
Pipeline Depth	Variable-length (big cores) 8 (LITTLE cores)		
CPU	1x Cortex-A78 4x Cortex-A55	2x Cortex-A78 6x Cortex-A55	4x Cortex-A76 4x Cortex-A55
CPU Clock	1x 2.5GHz (A78) 3x 2.4GHz (A78) 4x 1.9GHz (A55)	2x 2.4GHz (A78) 6x 2.0GHz (A55)	2x 2.6 GHz (A78) 2x 1.92 GHz (A78) 4x 1.8 GHz (A55)
Architecture	64-bit ARM		

## 5 Experimental results

The experiments are conducted on three mobile devices, namely, (1) Motorola Edge 30 5G equipped with a Snapdragon 778G+ microprocessor, (2) Samsung Galaxy XCover Pro 2 with an Exynos 1280 processor, and (3) Huawei P40 Pro with a HiSilicon Kirin 980 processor.

There are no select ion criteria on the phones, except that we intended to experiment all the three types of processors. Those were the devices available when conducting the research.

The performance test are configured to run using the following benchmarks: (1) Geekbench, (2) Benchmark & Tuning, (3) CPU and Battery Temperature, and (4) Power Consumption.

### 5.1 Geekbench

It is a benchmark commonly used for performance testing and is implemented on single and multicore [65]. Figure 4 shows results of applying Geekbench on the three aforementioned CPUs using a single core.

Results of Figure 4 do not consider the number of cores in the SoC. According to the results, Snapdragon 778G+ comes in the first place with 775 points, followed by Exynos 1280 with 746 points, then Kirin 980 with 701 points. Actually, Snapdragon and Exynos came in the first two places due to the type of pipeline used in each of them, which comprises additional stages and out-of-order stages. Additionally, they both have Cortex-A78 with higher CPU Clock rate rather than Kirin processor.



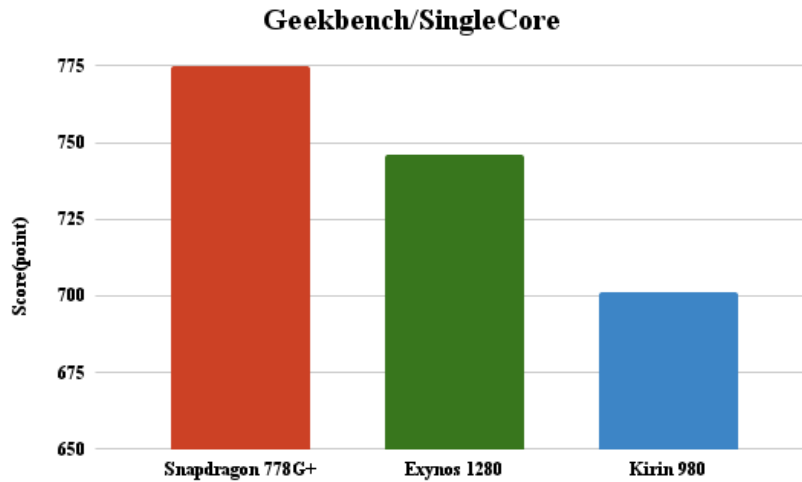


Fig. 4. Geekbench Scores on Single Core

Figure 5 shows the results of running the Geekbench on the same three processors, this time on a multicore basis.

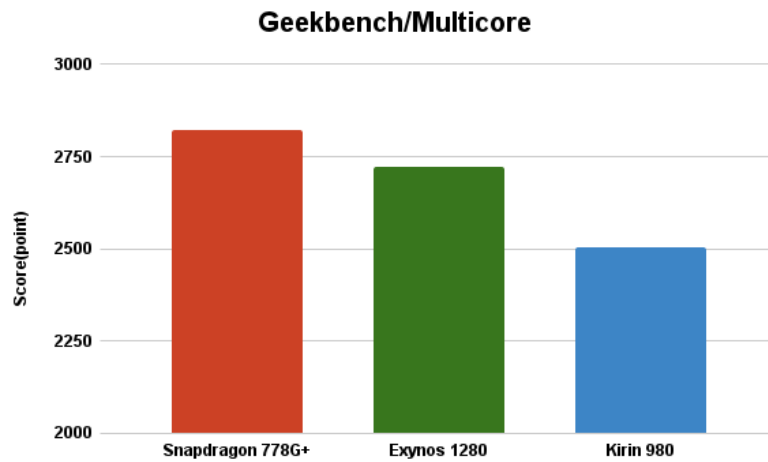


Fig. 5. Geekbench Scores on Multicore

Kirin 980 is in the last place again, this is because it has Cortex- A78 cores and Cortex- A55 cores which have lower specifications than those of the Cortex-A78 and Cortex-A55 cores that used in Snapdragon 778G+ and Exynos 1280. Snapdragon 778G+ is in the first rank again, followed by Exynos1280. Although they both have Cortex A78 and Cortex A55 cores, as well as the same pipeline stages and architecture,

but the 5nm Fin-FET technology used in Snapdragon 778G+ reveals the difference, as it comprises multi-gate transistors which improves the current flow, reduces the current leakage, and thus improves the overall performance.

## 5.2 Antutu benchmark

This benchmark is used to measure the performance of the CPU, Memory, and I/O without wasting the battery's energy. The benchmark supports multicore, with the capability to run single-threaded and multithreaded tests [66]. Figure 6 shows results of running this benchmark on the three-selected processors.

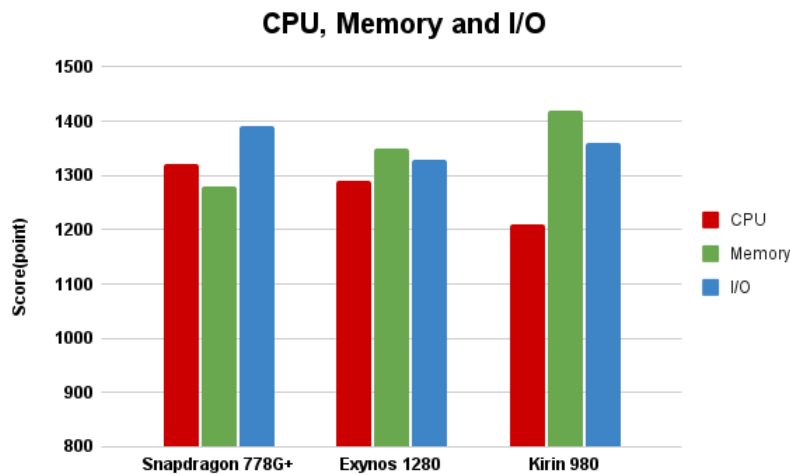


Fig. 6. Benchmark & Tuning Results

The benchmark measures CPU capability by executing arithmetic operations, sorting algorithm, and cryptography operations. To measure memory capability, the benchmark uses some memory operations such as resources allocation. Finally, I/O capability is measured by executing sequential-file access, random-file access, and database reading/writing operations.

Again, Snapdragon 778G+ achieved the highest scores in CPU tests. Exynos 1280 follows in the second place with a slight difference. Finally, Kirin 980 comes in the last place.

Typically, the results that were obtained so far by this benchmark conform to the previous results that we obtained by the Geekbench test, which can be considered as a double check on the CPU performance.

### 5.3 CPU and battery temperature

This test measures CPU and battery temperature when the CPU load increases. Figure 7 shows CPU temperature and Figure 8 shows battery temperature. In both figures, the lower the values, the better the results are.

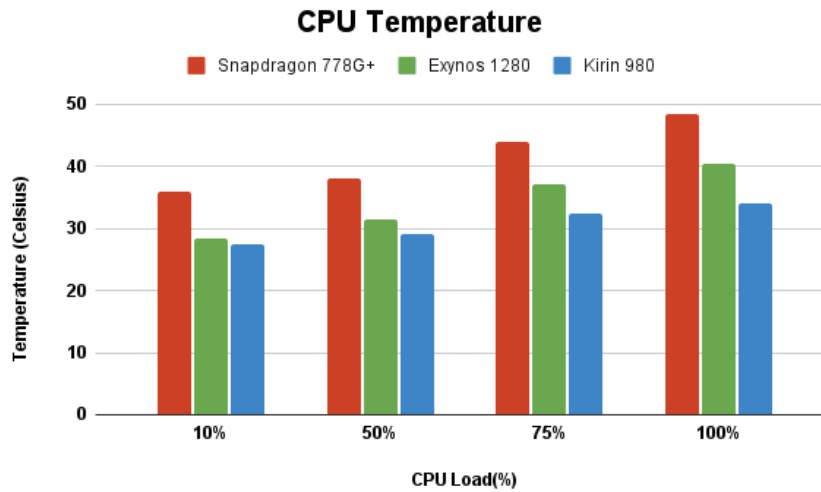


Fig. 7. CPU Temperature

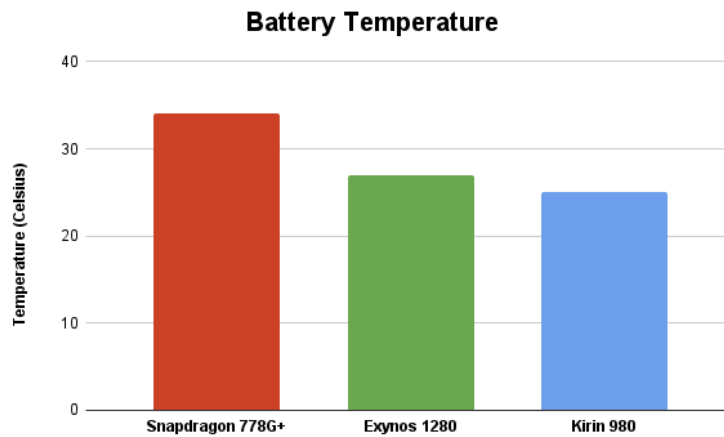


Fig. 8. Battery Temperature

Although Snapdragon 778G+ and Exynos 1280 have approximately the same CPU clock rate, a slight difference in battery temperature is realized when the CPU load was 100%, this is because Snapdragon processor throttles down several times when big

cores switch to the maximum speed (2.5 GHz). This throttle causes the CPU temperature to boost. On the other hand, Exynos 1280 big cores take several minutes on the 2.4 GHz speed before throttling down. Additionally, the 5nm FinFET SoC has a thermal headroom available to cool the SOC down when the CPU core load goes high. However, Hisilicon Kirin 980 have more LITTLE cores, it is suitable for the power consumption situation. This, in turn, has a great influence on reducing the CPU and battery temperature.

#### 5.4 Power consumption

This test has been conducted to compare power consumption capability when the CPU load increases. Figure 9 shows results on four CPU load values with Hisilicon Kirin 980 showing the best results.

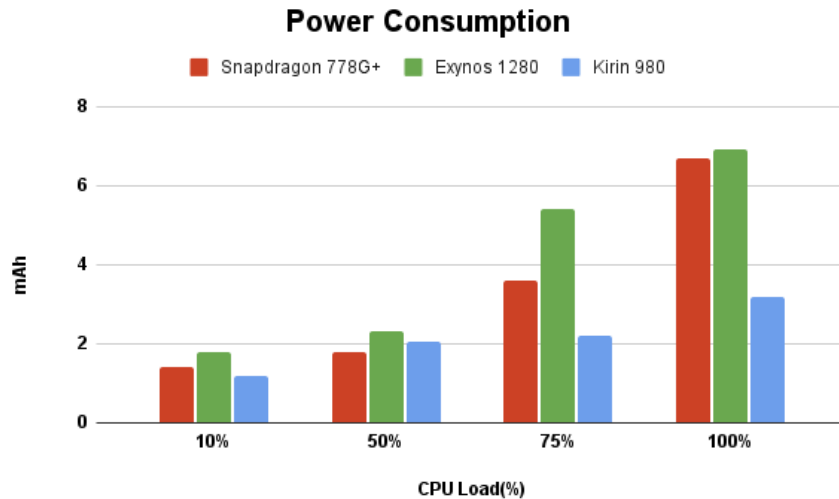


Fig. 9. Power Consumption

The values are obtained by measuring total energy consumption during each test, which takes two minutes, and repeating the test five times to ensure accuracy of the obtained results, then we take the average of the readings.

It is noteworthy that each time we deduct the power of the screen in order to give a pure representation of the load power of the SoC only.

As long Kirin 980 uses four Cortex-A55 1.8 GHz cores, then, the power consumption by those cores are the least. This is because the CPU overhead is reduced in those cores. Exynos 1280 comes in the second place, followed by Snapdragon 778G+, which comes in the last place.

In average, Figure 10. Shows that Snapdragon 778G+ achieved the best results. This gives an intuition that Exynos 1280 is not affected by CPU throttling or CPU and battery overheating. Exynos 1280 comes in the second place, in average. Finally, Hisilicon Kirin 980 comes last except in power consumption tests.

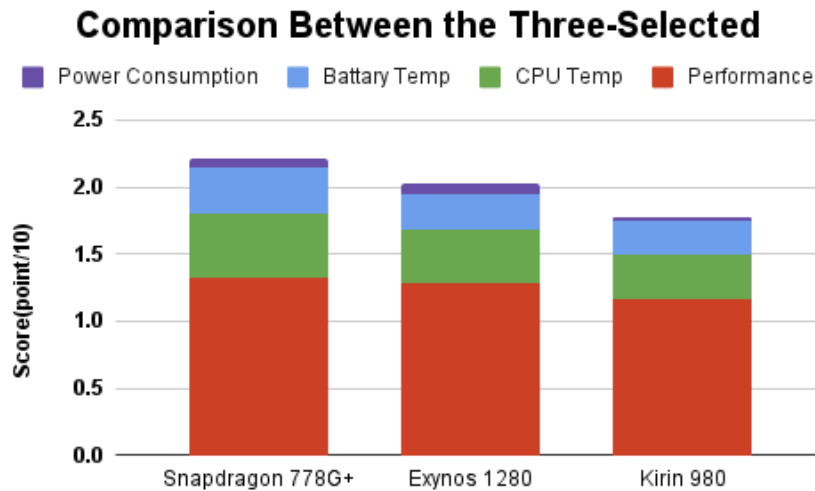


Fig. 10. Exynos, Snapdragon, and Kirin Comparison

According to the results we obtained, we can divide the SoC's into two categories. The first is the high-performance category, which contains Snapdragon 778G+ and Exynos 1280. Actually, these two processors recorded slight differences between them due to leveraging the big.LITTLE architecture. The second category is the low-power-consumption category, which is the Hisilicon Kirin 980.

In conclusion, each SoC has its advantages and disadvantages. For instance, Figure 10 shows Snapdragon 778G+ has an increased CPU performance compared to Kirin 980 by 1000 unit (almost 25%). On the other hand, Kirin 980 achieves lower device power consumption. Regarding Battery temperature, there is a 10-degree Celsius for Kirin 980 lower than Snapdragon 778G+, which is considered a high difference, and thus it appears to be a disadvantage of Snapdragon 778G+. Additionally, there is a 9-degree Celsius difference for Kirin 980 over Exynos 1280, which constitutes a heating problem with Exynos 1280.

Similarly, Snapdragon 778G+ performance increases by 25%, power consumption increases by 20%, and CPU temperature increase by 19%. Those readings for Snapdragon 778G+ are considered valuable. On other hand Hisilicon Kirin 980 increase the performance by only 18% and decrease the power consumption by 25% but increase the CPU temperature by 17%, which consider worthless. Regarding Exynos 1280, it increases the performance by 20%, with relatively high-power consumption by 23%, and CPU temperature by 21%. Briefly, Snapdragon 778G+ is the best SoC, followed by Exynos 1280 and finally Kirin 980.

## 6 Conclusion

ARM processors have combined the benefits of RISC architecture's while implementing some not so trifling components. The right arrangement of architecture and the simplicity of ARM cores made them the mostly used processor cores in the mobile world.

In this paper, we evaluated mobile ARM CPU designs and how design development impacted the mobile's performance and power consumption. We studied advantages of the 64-bit architecture. We considered how pipeline design and depth affects the performance, we also analyzed the effect of the processor cores on both CPU and battery temperature. The results indicate that mobile ARM processors design generations have exhibited significant performance improvements through aggressive core scaling techniques.

we believe that there are no significant variations between various types of phones with the same processors. As a result, there are no significant differences in the results if the experiments were conducted on different devices.

As a future work, machine learning and deep learning [67] techniques can be used to predict power and energy consumption and efficiency based on the architecture of the processor [68, 6], such as artificial neural networks (ANN) [69], and convolutional neural networks (CNN) [70]. Metaheuristic algorithms can be also used to suggest optimized solutions to the performance of mobile devices, such as harmony search (HS) [71], genetic algorithm (GA) [72, 73], and grey wolf optimizer (GWO) [74]. Also, processor performance can be investigated on IoT [75]. Fuzzy logic can also be incorporated in the investigation and design of energy-efficient mobile solutions [76, 77].

## 7 References

- [1] A. Guezzaz, M. Azrou, S. Benkirane, M. Mohy-Eddine, H. Attou and M. Douiba, "A Lightweight Hybrid Intrusion Detection Framework using Machine Learning for Edge-Based IIoT Security," *International Arab Journal of Information Technology*, vol. 19, no. 5, pp. 822-830, 2022. <https://doi.org/10.34028/iajit/19/5/14>
- [2] A. Al-Shaikh, R. Al-Sayyed and A. Sleit, "A Case Study for Evaluating Facebook Pages with Respect to Arab Mainstream News Media," *Jordanian Journal of Computers and Information Technology*, vol. 3, no. 3, pp. 142-156, 2017. <https://doi.org/10.5455/jjcit.71-14965-37248>
- [3] G. Samara, H. A. Munshar and R. Quaddoura, "Using Security Centre in Indoor Internet of Things," *International Journal of Quality Volume*, vol. 5, no. 2, pp. 130-136, 2022. <https://doi.org/10.34028/ijqa/5/2/223>
- [4] A. Al-Shaikh and A. Sleit, "Evaluating IndexedDB performance on web browsers," in 2017 8th International Conference on Information Technology (ICIT), Amman, Jordan, 2017. <https://doi.org/10.1109/ICITECH.2017.8080047>
- [5] M. R. Al-Mousa, Q. Al-Zaqebah, A. S. Al-Sherideh, M. Al-Ghanim, G. Samara, S. Al-Matarneh and M. Asassfeh, "Examining Digital Forensic Evidence for Android Applications," in 2022 International Arab Conference on Information Technology (ACIT), Abu Dhabi, United Arab Emirates, 2022. <https://doi.org/10.1109/ACIT57182.2022.9994221>

- [6] M. Al-Khateeb, M. R. Al-Mousa, A. S. Al-Sherideh, D. Almajali, M. Asassfeh and H. Khafajeh, "Awareness model for minimizing the effects of social engineering attacks in web applications," *International Journal of Data and Network Science*, vol. 7, no. 2, pp. 791-800, 2023. <https://doi.org/10.5267/j.ijdns.2023.1.010>
- [7] R. L. Ayala, S. R. Cosi and L. Andrade-Arenas, "Design of a Mobile Application to Improve the Lifestyle of Patients with Diabetes," *International Journal of Interactive Mobile Technologies*, vol. 17, no. 5, p. 100–116, 2023. <https://doi.org/10.3991/ijim.v17i05.37441>
- [8] A. al-Qerem, A. Hammarsheh, A. M. Ali, Y. Alslman and M. Alauthman, "Using Consensus Algorithm for Blockchain Application of Roaming Services for Mobile Network," *International Journal of Advances in Soft Computing and its Applications*, vol. 15, no. 1, pp. 99-112, 2023.
- [9] M. Asassfeh, W. Almobaideen and N. Obeid, "Spatial Cloaking for Location Privacy Protection of Smart Health Care Systems in Fog Computing," *Journal of Theoretical and Applied Information Technology*, vol. 100, no. 4, pp. 5142-5156, 2022.
- [10] A. Al-Shaikh, H. Khattab, A. Sharieh and A. Sleit, "Resource Utilization in Cloud Computing as an Optimization Problem," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 6, pp. 336-342, 2016. <https://doi.org/10.14569/IJACSA.2016.070643>
- [11] I. M. Rabbani, M. Aslam and A. M. Martinez-Enriquez, "SAFRank: Multi-Agent based Approach for Internet Services Selection," *International Arab Journal of Information Technology*, vol. 19, no. 3, pp. 298-306, 2022. <https://doi.org/10.34028/iajit/19/3/2>
- [12] F. G. H. Leite, R. B. B. Santos, N. H. Medina, V. A. P. Aguiar, R. C. Giacomini, N. Added, F. Aguirre, E. L. Macchione, F. Vargas and M. A. G. da Silveira, "Ionizing radiation effects on a COTS low-cost RISC microcontroller," in *2017 18th IEEE Latin American Test Symposium (LATS)*, Bogota, Colombia, 2017. <https://doi.org/10.1109/LATW.2017.7906762>
- [13] A. Al-Shaikh, H. Khattab and S. Al-Sharaeh, "Performance Comparison of LEACH and LEACH-C Protocols in Wireless Sensor Networks," *Journal of ICT Research and Applications*, vol. 12, no. 3, pp. 219-236, 2018. <https://doi.org/10.5614/itbj.ict.res.appl.2018.12.3.2>
- [14] J. Goodacre and A. Sloss, "Parallelism and the ARM instruction set architecture," *Computer*, vol. 38, no. 7, pp. 42-50, 2005. <https://doi.org/10.1109/MC.2005.239>
- [15] J. Holt, J. Dastidar, D. Lindberg, J. Pape and P. Yang, "A Full Lifecycle Performance Verification Methodology for Multicore Systems-on-Chip," *ACM Transactions on Design Automation of Electronic Systems*, vol. 17, no. 3, pp. 1-18, 2012. <https://doi.org/10.1145/2209291.2209294>
- [16] M. Keley, A. Khademzadeh and M. Hosseinzadeh, "Efficient Mapping Algorithm on Mesh-based NoCs in Terms of Cellular Learning Automata," *International Arab Journal of Information Technology*, vol. 16, no. 2, pp. 312-322, 2019.
- [17] N. Pinckney, T. Barr, M. Dayringer, M. McKnett, N. Jiang, C. Nygaard, D. M. Harris, J. Stanley and B. Phillips, "A MIPS R2000 implementation," in *DAC '08: Proceedings of the 45th annual Design Automation Conference*, Anaheim, California, 2008. <https://doi.org/10.1145/1391469.1391497>
- [18] ARM, *ARM Developer Suite: Assembler Guide*, 1.2 ed., ARM Limited, 2001.
- [19] H. Lozano and M. Ito, "Increasing the Code Density of Embedded RISC Applications," in *2016 IEEE 19th International Symposium on Real-Time Distributed Computing (ISORC)*, York, UK, 2016. <https://doi.org/10.1109/ISORC.2016.33>
- [20] T. Bonny and J. Henkel, "Efficient Code Density Through Look-up Table Compression," in *2007 Design, Automation & Test in Europe Conference & Exhibition*, Nice, France, 2007. <https://doi.org/10.1109/DATE.2007.364390>

- [21] S. Furber, "Microprocessors: the engines of the digital age," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 473, no. 20160893, p. 20160893, 2017. <https://doi.org/10.1098/rspa.2016.0893>
- [22] A. Leone, G. Rescio, A. Caroppo, P. Siciliano and A. Manni, "Human Postures Recognition by Accelerometer Sensor and ML Architecture Integrated in Embedded Platforms: Benchmarking and Performance Evaluation," *Sensors*, vol. 23, no. 2, p. 1039, 2023. <https://doi.org/10.3390/s23021039>
- [23] J. Yu and X. Lu, "Improvement and Application of Task Scheduling Algorithm for Embedded Real-Time Operating System," in *Proceedings of the World Conference on Intelligent and 3-D Technologies (WCI3DT 2022)*. Smart Innovation, Systems and Technologies, vol. 323, R. Kountchev, K. Nakamatsu, W. Wang and R. Kountcheva, Eds., Springer, Singapore, 2023, p. 621–628. [https://doi.org/10.1007/978-981-19-7184-6\\_51](https://doi.org/10.1007/978-981-19-7184-6_51)
- [24] D. Shin and H.-J. Yoo, "The Heterogeneous Deep Neural Network Processor With a Non-von Neumann Architecture," *Proceedings of the IEEE*, vol. 108, no. 8, pp. 1245-1260, 2020. <https://doi.org/10.1109/JPROC.2019.2897076>
- [25] S. Wang, X. Jiang, F. Bai, W. Xiao, X. Long, Q. Ren and Y. Kang, "A True Process-Heterogeneous Stacked Embedded DRAM Structure Based on Wafer-Level Hybrid Bonding," *Electronics*, vol. 12, no. 5, 2023. <https://doi.org/10.3390/electronics12051077>
- [26] P. Yang, Q. Wang, H. Ye and Z. Zhang, "Partially shared cache and adaptive replacement algorithm for NoC-based many-core systems," *Journal of Systems Architecture*, vol. 98, pp. 424-433, 2019. <https://doi.org/10.1016/j.sysarc.2019.05.002>
- [27] M. Zargham, *Computer Architecture*, Prentice Hall, 1996.
- [28] K. Watts and P. Oman, "Stack-Based Buffer Overflows in Harvard Class Embedded Systems," in *Critical Infrastructure Protection III*. ICCIP 2009. IFIP Advances in Information and Communication Technology, vol. 311, C. Palmer and S. Sheno, Eds., Springer, Berlin, Heidelberg, 2009, p. 185–197. [https://doi.org/10.1007/978-3-642-04798-5\\_13](https://doi.org/10.1007/978-3-642-04798-5_13)
- [29] R. Pawson, "The Myth of the Harvard Architecture," *IEEE Annals of the History of Computing*, vol. 44, no. 3, pp. 59-69, 2022. <https://doi.org/10.1109/MAHC.2022.3175612>
- [30] S. Misra, A. A. Alfa, S. O. Adewale, M. A. Akogbe and M. O. Olaniyi, "A Two-Way Loop Algorithm for Exploiting Instruction-Level Parallelism in Memory System," in *Computational Science and Its Applications – ICCSA 2014*. ICCSA 2014. Lecture Notes in Computer Science, vol. 8583, B. Murgante, S. Misra, A. M. A. C. Rocha, C. Torre, J. G. Rocha, M. I. Falcão, D. Taniar, B. O. Apduhan and O. Gervasi, Eds., Springer, Cham, 2014, p. 255–264. [https://doi.org/10.1007/978-3-319-09156-3\\_19](https://doi.org/10.1007/978-3-319-09156-3_19)
- [31] Z. Wang, X. Dong, Y. Kang and H. Chen, "Parallel SHA-256 on SW26010 many-core processor for hashing of multiple messages," *The Journal of Supercomputing*, vol. 79, p. 2332–2355, 2023. <https://doi.org/10.1007/s11227-022-04750-7>
- [32] S. Shivaraju and G. Pudur, "Splay Thread Cooperation on Ray Tracing as a Load Balancing Technique in Speculative Parallelism and GPGPU," *International Arab Journal of Information Technology*, vol. 15, no. 1, pp. 167-176, 2018.
- [33] V. Saravanan, K. D. Pralhaddas, D. P. Kothari and I. Woungang, "An optimizing pipeline stall reduction algorithm for power and performance on multi-core CPUs," *Human-centric Computing and Information Sciences*, vol. 5, 2015. <https://doi.org/10.1186/s13673-014-0016-8>
- [34] J. Hennessy and D. Patterson, *Computer Architecture: A Quantitative Approach*, 6th ed., Morgan Kaufmann, 2017.
- [35] J. P. Shen and M. H. Lipasti, *Modern Processor Design: Fundamentals of Superscalar Processors*, 1st ed., Waveland Press, Inc., 2013.



- [36] S. V. Adve and K. Gharachorloo, "Shared memory consistency models: a tutorial," *Computer*, vol. 29, no. 12, pp. 66-76, 1996. <https://doi.org/10.1109/2.546611>
- [37] R. D. Barnes, E. M. Nystrom and J. (. Navarro, "Beating in-order stalls with "flea-flicker" two-pass pipelining," *IEEE Transactions on Computers*, vol. 55, no. 1, pp. 18-33, 2006. <https://doi.org/10.1109/TC.2006.4>
- [38] ARM, "ARM® Architecture Reference Manual: ARMv7-A and ARMv7-R edition," ARM, 2012.
- [39] J. Dean, J. E. Hicks, C. A. Waldspurger, W. E. Weihl and G. Chrysos, "ProfileMe: hardware support for instruction-level profiling on out-of-order processors," in *Proceedings of the 30th Annual ACM/IEEE International Symposium on Microarchitecture (MICRO 30)*, 1997.
- [40] A. Tiwari, K. Keipert, A. Jundt, J. Peraza, S. S. Leang, M. Laurenzano, M. S. Gordon and L. Carrington, "Performance and energy efficiency analysis of 64-bit ARM using GAMESS," in *Co-HPC '15: Proceedings of the 2nd International Workshop on Hardware-Software Co-Design for High Performance Computing*, Austin, Texas, USA, 2015. <https://doi.org/10.1145/2834899.2834905>
- [41] Y. K. Paunski and G. T. Angelov, "Performance and power consumption analysis of low-cost single board computers in educational robotics," *IFAC-Papers OnLine*, vol. 52, no. 25, pp. 424-428, 2019. <https://doi.org/10.1016/j.ifacol.2019.12.575>
- [42] G. Taffoni, S. Bertocco, I. Coretti, D. Goz, A. Ragagnin and L. Tornatore, "Low Power High Performance Computing on Arm System-on-Chip in Astrophysics," in *Proceedings of the Future Technologies Conference (FTC) 2019. FTC 2019. Advances in Intelligent Systems and Computing*, vol. 1069, K. Arai, R. Bhatia and S. Kapoor, Eds., Springer, Cham, 2019, p. 427-446. [https://doi.org/10.1007/978-3-030-32520-6\\_33](https://doi.org/10.1007/978-3-030-32520-6_33)
- [43] M. Qasaimeh, K. Denolf, J. Lo, K. Vissers, J. Zambreno and P. H. Jones, "Comparing Energy Efficiency of CPU, GPU and FPGA Implementations for Vision Kernels," in *2019 IEEE International Conference on Embedded Software and Systems (ICISS)*, Las Vegas, NV, USA, 2019. <https://doi.org/10.1109/ICISS.2019.8782524>
- [44] M. F. Dolz, S. Barrachina, H. Martínez, A. Castelló, A. Maciá, G. Fabregat and A. E. Tomás, "Performance–energy trade-offs of deep learning convolution algorithms on ARM processors," *The Journal of Supercomputing*, 2023. <https://doi.org/10.1007/s11227-023-05050-4>
- [45] C. Marantos, K. Salapas, L. Papadopoulos and D. Soudris, "A Flexible Tool for Estimating Applications Performance and Energy Consumption Through Static Analysis," *SN Computer Science*, vol. 2, 2021. <https://doi.org/10.1007/s42979-020-00405-7>
- [46] S. Eum, H. Kim, H. Kwon, M. Sim, G. Song and H. Seo, "Parallel Implementations of ARIA on ARM Processors and Graphics Processing Unit," *Applied Sciences*, vol. 12, no. 23, 2022. <https://doi.org/10.3390/app122312246>
- [47] S. Chheda, A. Curtis, E. Siegmann and B. Chapman, "Performance Study on CPU-Based Machine Learning with PyTorch," in *HPC Asia '23 Workshops: Proceedings of the HPC Asia 2023 Workshops*, Raffles Blvd, Singapore, 2023. <https://doi.org/10.1145/3581576.3581615>
- [48] Atmel ES2, "ARM7TDMI (Thumb) Data Sheet," Atmel ES2, 1999.
- [49] ARM, "ARM7TDMI Technical Reference Manual," ARM, 2001.
- [50] ARM, "ARM9TDMI Technical Reference Manual," ARM, 2000.
- [51] ARM, "Architectures, Processors, and Devices: Development Article," ARM, 2009.
- [52] S. Furber, *ARM System-on-Chip Architecture*, 2nd ed., Pearson Education Limited, 2000.
- [53] D. Brooks and M. Martonosi, "Dynamically Exploiting Narrow Width Operands to Improve Processor Power and Performance," in *Proceedings Fifth International Symposium on High-Performance Computer Architecture*, 1999. <https://doi.org/10.1109/HPCA.1999.744314>

- [54] ARM, "ARM Cortex-A Series Programmer's Guide for ARMv8-A," ARM, 2015.
- [55] S. Yoo, Y. Shim, S. Lee, S.-A. Lee and J. Kim, "A Case for Bad Big.LITTLE Switching: How to Scale Power-performance in SI-HMP," in Proceedings of the Workshop on Power-Aware Computing and Systems (HotPower '15), 2015. <https://doi.org/10.1145/2818613.2818745>
- [56] ARM, "big.LITTLE Technology: The Future of Mobile," ARM, 2013.
- [57] Qualcomm, "Snapdragon 778G+ 5G Mobile Platform," Qualcomm Technologies, Inc., [Online]. Available: <https://www.qualcomm.com/products/application/smartphones/snapdragon-7-series-mobile-platforms/snapdragon-778g-plus-5g-mobile-platform>. [Accessed 04 March 2023].
- [58] Qualcomm, "Qualcomm® QCM6490/Qualcomm® QCS6490 SoCs," [Online]. Available: <https://developer.qualcomm.com/sites/default/files/docs/adreno-gpu/snapdragon-game-toolkit/gdg/cpu/cpu.html>. [Accessed 04 March 2023].
- [59] J. Anderson, Y. He, B. Bahr and D. Weinstein, "Integrated acoustic resonators in commercial fin field-effect transistor technology," Nature Electronics, vol. 5, p. 611–619, 2022. <https://doi.org/10.1038/s41928-022-00827-6>
- [60] ARM, "ARM® Cortex®-A57 MPCore Processor Technical Reference Manual," ARM, 2014. <https://doi.org/10.1016/B978-0-12-408082-9.00003-8>
- [61] ARM, "ARM Cortex-A53 MPCore Processor Technical Reference Manual," ARM, 2014.
- [62] Samsung, "Product finder | Support | Samsung Semiconductor Global," [Online]. Available: <https://semiconductor.samsung.com/support/tools-resources/product-finder/?tabname=processor>. [Accessed 04 March 2023].
- [63] Samsung, "Exynos 1280 | Exynos 1280 | Create infinite possibilities," [Online]. Available: <https://semiconductor.samsung.com/resources/brochure/Exynos1280.pdf>. [Accessed 04 March 2023].
- [64] HiSilicon, "Kirin 980 Chipset | HiSilicon Official Site," [Online]. Available: <https://www.hisilicon.com/en/products/Kirin/Kirin-flagship-chips/Kirin-980>. [Accessed 04 March 2023].
- [65] P. Dukan, A. Kovari and J. Katona, "Low consumption and high performance Intel, AMD and ARM based Mini PCs," in 2014 IEEE 15th International Symposium on Computational Intelligence and Informatics (CINTI), Budapest, Hungary, 2014. <https://doi.org/10.1109/CINTI.2014.7028662>
- [66] Antutu, "Antutu Benchmark - Know Your Android Better," [Online]. Available: <https://www.antutu.com/en/index.htm>. [Accessed 04 March 2023].
- [67] P. Khomand, M. Sabeti, R. Boostani, E. Moradi, M. Odeh and M. R. AL-Mousa, "Deep Learning for Automatic Determination of Bone Age in Children," in 2022 International Engineering Conference on Electrical, Energy, and Artificial Intelligence (EICEEAI), Zarqa, Jordan, 2022. <https://doi.org/10.1109/EICEEAI56378.2022.10050468>
- [68] M. Injadat, A. Moubayed, A. B. Nassif and A. Sham, "Multi-Stage Optimized Machine Learning Framework for Network Intrusion Detection," IEEE Transactions on Network and Service Management, vol. 18, no. 2, pp. 1803-1816, 2022. <https://doi.org/10.1109/TNSM.2020.3014929>
- [69] K. E. Alqawasmi and A. M. Alsmadi, "Estimation of ARMA Model Order Using Artificial Neural Networks," Circuits, Systems, and Signal Processing, 2023. <https://doi.org/10.1007/s00034-023-02305-6>
- [70] N. Al-Milli and B. H. Hammo, "A Convolutional Neural Network Model to Detect Illegitimate URLs," in 2020 11th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 2020. <https://doi.org/10.1109/ICICS49469.2020.239536>

- [71] A. Al-Shaikh, B. A. Mahafzah and M. Alshraideh, "Hybrid harmony search algorithm for social network contact tracing of COVID-19," *Soft Computing*, 2021. <https://doi.org/10.1007/s00500-021-05948-2>
- [72] N. Al-Milli, A. Hudaib and N. Obeid, "Population Diversity Control of Genetic Algorithm Using a Novel Injection Method for Bankruptcy Prediction Problem," *Mathematics*, vol. 9, no. 8, 2021. <https://doi.org/10.3390/math9080823>
- [73] N. Al-Milli, "Hybrid Genetic Algorithm with Great Deluge to Solve Constrained Optimization Problems," *Journal of Theoretical and Applied Information Technology*, vol. 59, no. 2, pp. 385-389, 2017.
- [74] A. Al-Shaikh, B. A. Mahafzah and M. Alshraideh, "Metaheuristic Approach using Grey Wolf Optimizer for Finding Strongly Connected Components in Digraphs," *Journal of Theoretical and Applied Information Technology*, vol. 97, no. 16, pp. 4439-4452, 2019.
- [75] G. S. a. M. A. F. Mohammad A. Hassan, "IoT Forensic Frameworks (DFIF, IoTDOTS, FSAIoT): A Comprehensive Study," *International Journal of Advances in Soft Computing and its Applications*, vol. 14, no. 1, pp. 72-86, 2022. <https://doi.org/10.15849/IJASCA.220328.06>
- [76] W. Polnigongit, W. Chanwimalueng and S. Fitzgerald, "The Development of a Fuzzy Model and Usability Test of a Recommended Interface Design for Mobile Phones for Elderly Users," *International Journal of Interactive Mobile Technologies*, vol. 17, no. 2, p. 118–136, 2023. <https://doi.org/10.3991/ijim.v17i02.33877>
- [77] F. Salvetti, T. L. Capshaw, L. Zanin, K. C. O'Connor, Q. Zeng and B. Bertagni, "The GW Mobile Learning Center: Mixed-Reality within an Immersive and Interactive Learning Setting," *International Journal of Advanced Corporate Learning*, vol. 16, no. 2, p. 93–108, 2023. <https://doi.org/10.3991/ijac.v16i2.35737>

## 8 Authors

**Ala'a Al-Shaikh** is an Assistant Professor of Computer Science at the Department of Cybersecurity, the Faculty of Information Technology, Zarqa University, Jordan. He received his PhD in Computer Science from the University of Jordan in 2020. Al-Shaikh served in different academic institutions in Jordan and Saudi Arabia. His Greatest-Of-All-Time achievement that he is always proud of is the computerization of the General Associate Degree Examination in Jordan, which is the second national examination in Jordan that grants students who pass it the associate degree as per the Law of Higher Education of Jordan. His research interests are in the fields of computational intelligence, machine learning, artificial intelligence, parallel computing, distributed systems, cloud computing, algorithms, e-learning, and cybersecurity (email: [ashaikh@zu.edu.jo](mailto:ashaikh@zu.edu.jo)).

**Ameen Shaheen** received his B.Sc. in Computer Information System in 2010, and MS in Computer Information System in 2012, and PhD in Computer Science in 2016. Shaheen is currently an assistant professor of Computer Science at Khawarizmi University Technical College. He was a Team leader at Ainon Company for computer systems. He was a faculty member at Philadelphia University and Applied Science University. He taught courses like programming, databases, algorithm and systems analysis.

**Mohammad Rasmi Al-Mousa** is currently serving as an associate professor in the Faculty of IT, and Head of Cybersecurity Department, Zarqa University. He received his PhD in Network Security from Universiti Sains Malaysia in 2013. His research interests include digital forensics, cybersecurity, E-government strategy, cloud computing and software engineering (e-mail: mmousa@zu.edu.jo).

**Khaled Alqawasmi** is working as an associate professor in the department of cybersecurity at Zarqa University, Jordan. He obtained his PhD degree from Amman Arab University in 2010. His research interests include digital signal processing (DSP), software quality assurance, multimedia, web engineering, web security, image processing, and artificial neural networks.

**Ala'a Saeb Al-Sherideh** is an Assistant Professor of Computer Science / Cybersecurity at the Department of Cybersecurity, the Faculty of Information Technology, Zarqa University, Jordan. He received his PhD from University Sains Islam Malaysia in 2021. His research interests include cybersecurity, Mobile government (MG), Security, and Privacy (email: asherideh@zu.edu.jo).

**Hebatullah Khattab** is an Assistant Professor of Computer Science at the Department of Computer Science, the Faculty of Information Technology, Zarqa University, Jordan. She received her PhD in Computer Science from the University of Jordan in 2020. Currently, she is the head of the Computer Science Department at the Faculty of Information Technology, Zarqa University, Jordan. Her research interests are in the fields of artificial intelligence, parallel computing, algorithms, and computer networks (email: h.awwad@zu.edu.jo).

Article submitted 2023-03-06. Resubmitted 2023-05-12. Final acceptance 2023-05-13. Final version published as submitted by the authors.