HISTORY AND FUTURE TRENDS OF MULTICORE COMPUTER ARCHITECTURE

Abdulrahman Alseqyani and Abdullah Almutairi

Department of Computer and Information Technology, Zulfi College of Technology, TVTC, Saudi Arabia

ABSTRACT

The multicore technology concept is centered on the parallel computing possibility that can boost computer efficiency and speed by integrating two or more CPUs (Central Processing Units) in a single chip. A multicore architecture places multiple processor cores and groups them as one physical processor. The primary goal is to develop a system that can handle and complete more than one task at the same time, thereby getting a better system performance in general. This paper will describe the history and future trends of multicore computer architecture.

KEYWORDS

Multicore technology, multicore architecture, and performance.

1. INTRODUCTION

Multicore refers to an architecture whereby a single physical processor integrates the core logic of two or more processors. In this case, one integrated circuit is adapted to hold or package these processors. A die involves these single integrated circuits. According to [1], a multicore architecture places multiple processor cores, and groups them as one physical processor. The primary goal is to develop a system, which can handle and complete more than one task at the same time, thereby getting a better system performance in general. [2] add that multicore architecture technology is commonly applied in multicore processors where two or more processor cores or chips run at the same time as one system. Notably, multicore-based processors are used in desktops, mobile devices, servers, and workstations.

2. OVERVIEW

The multicore technology concept is centered on the parallel computing possibility that can boost computer efficiency and speed by integrating two or more CPUs (Central Processing Units) in a single chip. In their study, [3] point out that the integration of two or more CPUs in a single chip reduces the power and heat consumption in the system, which means a better performance is realized, but with the same or less amount of energy. Therefore, [2] document that the multicore processor architecture enables and facilitates communication between every available core in ensuring that the processing tasks are assigned and divided accurately. Once the task is completed, the processed data from every core is delivered back to the motherboard through one shared gateway – a technique that significantly enhances performance compared to a single-core processor using the same speed. [3] conclude that multicore technology is highly effective when handling challenging applications and tasks such as 3-D gaming, encoding, and video editing. There are two main types of processors, and these are single-core processors and multicore processors.

DOI: 10.5121/ijcga.2023.13201

2.1. Single-Core Processor

A single-core processor refers to a microprocessor with one core on a chip, and that it runs a single thread at one given time. According to [4], the term became common following the appearance and development of multi-core processors as it ensured single-core designs were distinguished from others. A single-core microprocessor comprises an integrated circuit, which facilitates or implements exactly a single independent core (physical execution unit) in a package with a single chip. For example, Intel developed and released a Core 2 Duo and a Core 2 Solo, with the latter becoming a variant with a 'single-core' [3]. Research shows that most microprocessors were single-core before the multi-core era.

According to [4], before the multi-core era of processors, it was difficult to achieve performance gains with single-core units, especially from the increased transistor count and clock speed granted by Moore's law. In this case, there were decreasing returns to increasing the pipeline depth, add execution units, and increase CPU cache sizes. As observed by [2], the first smartphones designed and fitted with dual-core processors were available in the market in 2010, but before that, these gadgets had single-core processors with a maximum speed of 1.4GHz. Additionally, the basis of using the single-core processors before the current multi-core ones was based on a single factor, which is facilitating speed in addition to power efficiency.

2.2. Multicore Processors

The terms 'dual-core' and 'multi-core' are commonly referred to as some sort of CPU (Central Processing Unit); sometimes the terms are applicable to the System of a Chip (SoC) and Digital Single Processors (DSP). In their study, [4] emphasize that the terms are generally used to refer to multi-core microprocessors developed on a similar integrated circuit. The microprocessors used currently in almost all gadgets, especially personal computers, and desktops are multicore, which ensures multiprocessing is implemented in a single physical package. As cautioned by [6], it should be noted that cores may share cores or not, and the shared memory or message passing methods of inter-core communication may be implemented. Multicore processors are used across several application domains, which include embedded, general-purpose, network, graphics (GPU), and Digital Signal Processing (DSP).

The improvement of speed and performance acquired through the integration of a multicore processor depends a lot on the software algorithms, including their implementation. Particularly, there are limited possible gains by the software fraction, which can run simultaneously and parallel on multicores [1].

3. HISTORY OF MULTICORE ARCHITECTURE

The first commercial multicore processor architecture was Power 4 processor developed by IBM in 2001 for its RISC servers [5]. At the same time, the first dual-core processor was Pentium Processor Extreme Edition 840 by Intel, released in 2005. Approximately two weeks later, the Opteron 800 Series and the Athlon 64 X2 multicore processor architectures were brought to the market by AMD.

Several business objectives and motives drove the design and development of multicore architectures. For many decades, it was difficult to boost CPU performance by reducing the area of the IC (integrated circuit) as it led to the reduction of the cost per device on the IC [6]. At the same time, for the same area of the circuit, additional transistors could be applied in the design that increased functionality, more so for the CISC (Complex Instruction Set Computing)

architectures. According to [7], clock rates are boosted by magnitude orders throughout the 20th century ranging from many megahertz in the early 1980s to a lot of gigahertz at the beginning of 2000.

Throughout the early 1980s and 1990s, following the gradual slowing down of clock speed improvements, increased adoption of parallel computing was evident as more multicore processors were pursued, especially by organizations like Intel and IBM, to improve the overall processing performance [1]. Additionally, [3] documented that the early 2000s saw the use of multiple cores, especially on the same CPU chip. On the other hand, there were more sales of the CPU chips, especially those with more than two cores. For instance, Intel and IBM began producing a 48-core processor, with each having an x86 architecture, especially for cloud computing research.

In the last two decades, in the search for more processing power, especially for personal computers, the supercomputer improvement required changes, and these changes led to the integration of more than one processing core in personal computers. In this way, [5] argued that personal computers were expected to continue improving, especially in performance, without the need to continue increasing the clock speed of the processor. Therefore, in 2005, because of the highly competitive marketplace and some alternatives, some of the major CPU manufacturers like IBM opted for alternative processors.

Between 2005 and 2010, these CPU manufacturers worked on the development and release of central processor units, and it was later referred to as the multicore revolution. The Multicore revolution period marked the beginning of a trend that resulted in a massive shift in the consumer computing market evolution [7]. Today, it is difficult to buy a personal computer with a single-core CPU since even low-power and low-end central processors have been designed with two or more cores per die. [5] conclude that in 2008, Intel released the L5420 and L5410 quad-core green processors for work stations and servers that operate faster than the previous quad-core processors without consuming additional power.

4. PROCESSOR ARCHITECTURE

The term “architecture” typically means a construction and building design. “Architecture,” in the computing world, refers to the design of computer systems. [8] point out that computer architecture comprises everything from the relationship between multiple computer systems like the “client-server” model to specific components from within a computer. The processor architecture of a computer is the most important hardware design as it determines the kind of software the computer runs in, as well as what other supported hardware components [9]. For instance, Intel’s x86 as a standard type of processor architecture most PCs use. By using this hardware design, computer manufacturers can develop machines, including different components of hardware, but using similar software. For many years, [8] added that Apple shifted to x86 architecture from the PowerPC architecture to make the compatibility of the Macintosh platform with Windows PCs more certain and possible.

The motherboard architecture is also an integral plat of the computer system, especially in determining what software and hardware computer systems will support. The design of the motherboard is often referred to as the “chipset” and defines various processor models, including other components working with the motherboard. In this case, while two motherboards may support the x86 processors, one would function with a new processor model. At the same time, a newer chipset could also demand a faster RAM, as well as a different video card type compared to older models. Notably, a lot of modern computers have chipsets and 64-bit processors, while old computers operate on 32-bit processor architecture. At the same time, computers with 64-bit

chipsets support a lot of memory compared to those with 32-bit chipsets and can operate software designed with 64-bit processors.

Modern microprocessors are among the most complex systems ever developed by humans. For instance, a single silicon chip, the size of a fingernail, may contain a complete process with high-performance, "large cache memories, and the logic required to interface it to external devices" [9]. Regarding performance, the processors on a single chip may overcome large supercomputers costing over $10 million two decades ago. Again, even the embedded processors found in daily devices such as personal digital assistant and cell phones are powerful compared to the early developers of computer systems.

Part of the processor architecture is the Y86 instruction set architecture, which includes the various state elements, their encodings, the instructions set, and a set of programming conventions, which handles exceptional events. Part of this architecture is the Y86 program, where the processor state can be modified, and the memory locations are referenced using "virtual addresses" [10]. A combination of the operating system software and hardware translates these into physical or actual addresses, which indicates where the values are located or stored in the memory.

There is also the status code which is the final part of the program state architecture, and it indicates the overall state of executing a program. The status code will either indicate normal operation or that there is an exception that has taken place like in scenarios where an invalid memory address has been detected. The Y86 instructions also make up the processor architecture, and it is used as a target for processor implementations. The Y86 instructions set is a subset of what [10] understood to be an IA32 instruction set. The set comprises 4-byte integer operations, and it comprises some addressing modes, including smaller operations set. The use of 4-byte data, for instance, can be referred to as "words" with no doubt. The assembly-code format appears like the ATT format, especially for IA32.

4.1. Core Organization

The multiple cores concept may appear simple but when it comes to scalability issues, there are various tradeoffs under consideration. For instance, it is important to consider whether the processor should be homogenous or heterogeneous. According to [10], some current general-purpose multicore processors show homogeneity both in performance and instruction-set architecture. It means that the organization of these cores can execute similar binaries, and from a functional viewpoint from which the running of a core program is done, it does not matter [9]. The recent organization of multicore architecture allows for system software to influence the frequency of the clock for every core targeted at either save power or to improve single-thread performance temporarily.

On the other hand, there are two or more different core types in a heterogeneous architecture and may be different in both the ISA (instruction set architecture), performance and functionality. The Cell BE architecture is the most widespread and popular heterogeneous multicore architecture developed following a combined effort of Sony, IBM, and Toshiba; it is used in such areas as computers and gaming devices targeting high-performance computing. Instead of that, a homogeneous architecture mounted with a shared global memory can easily be programmed for parallelism where the entire core is used to program as opposed to a heterogeneous architecture where there are no similar instruction sets.

Additionally, cores organizations can show major differences internally. According to [8], every modern core design can be pipelined today where there are decoding and execution of

instructions in stages to boost the overall throughput although there may be similar or improved instruction latency. [12] added that most high-performance architecture designs are presented with core speculative instruction scheduling integrated into hardware. Its presence ensures the average number of instructions is increased per clock cycle. Still, due to the ILP (instruction-level parallelism) in legacy applications, they become less important when it comes to multicore architectures.

4.2. Interconnection Networks

Having a chip with multiple cores requires mechanisms for inter-core communication. According to [5], the historical way where there is communication between the individual processors and multicore processors in a shared memory multiprocessor has been through a common bus every processor shares. This was also the case with the early general-purpose vendors like Intel or AMD especially in the earlier multicore processors. [11] point out that the shared bus also ensures the cache coherency implementation is facilitated like a broadcast medium of communication.

The current designs are based on the understanding and realization that such medium of share communication as buses have problems both in bandwidth and latency [12]. A shared busy has numerous electrical wires, and the presence of potential slave units results in the capacity load becoming slower. At the same time, the fact that there are numerous units share means the bus ends up limiting the bandwidth from every core. Lastly, [1] emphasizes that some general-purpose multicore processors currently apply a cross-bar interconnect between such processor modules as one or two-level cache memories, as well as the last-level memory and cache interfaces. The increase in core numbers of on-chip networks of communication will increase results in power constraints and face scalability.

4.3. Memory Architectures

The memory interface makes up an important component of high-performance processors, and most importantly, a multicore processor since it is a shared resource among all cores on the chip [12]. The Intel and AMD's recent high-end chips, the memory architecture was moved onto the chip, and is different from the I/O interfaces towards increasing the memory bandwidth, while enabling access to both the I/O memory and devices [1]. Particularly, there is the DRAM controller.

4.4. Support for Parallel Programming

The needed support for parallel programming is facilitated by store atomicity, where the consistency problem of the memory needs to be tackled in any multicore design. Therefore, where there are multiple copies of a similar memory location, there should be a propagation of the memory location where zero time is realized in every core before parallel programming is realized and facilitated. [11] observe that while programmers may be taken for granted, memory consistency is one of the most complex and critical issues for consideration when multicore systems are designed. Understanding the fundamentals of the workability of memory access is makes it essential when handling tricky bugs of concurrency or the implementation of support to synchronize primitives.

5. MULTICORE ISSUES

During an upgrade of the hardware platform to a newer and more powerful Central Processing Unit (CPU) with faster multicores, the expectation is the application to run faster as additional cores should reduce the CPU load on average, thus reducing delays [11]. However, in most cases, the application may slow down, while the CPU load remains similar for older CPUs. In this case, with high-end or modern CPUs, there are examples were interferences are observed, thus breaking determinism [12]. The solution is the development of scalability unless there is an architect of an application where a multicore environment is taken advantage of. As a result, 4-core and a single-core IPC end up performing almost similarly contrary to the belief that scale linearly of the RTOS application should be present to facilitate 4-times faster 4-core IPCs compared to IPCs with a single core [10]. At the same time, even if the application seeks to apply multiple cores, other architectural optimizations, which involve IO, memory access, data synchronization, and caching strategies, among others, must be considered for the system to realize the best optimal scalability.Therefore, since a system may lack that which delivers linear scalability, the theoretical limit can be attained following every application.

5.1. Scalability

Multicore scalability and efficiency are measured and determined by adopting timing routines in comparison with theoretical values. However, computational capacity should be independent of temperature, power consumption, and computational load, although it does not hold when there is enabling of power-saving features. [11] observe that using multiple cores consumes a lot of power as well as generate more heat, as well as create more room to overclock where some cores are active. Therefore, frequency scaling is necessary.

The frequency of CPU is established following a base clock and multiplier frequency. Therefore, by lowering the multiplier, the CPU frequency will be reduced, and the rate of power consumption will go down. According to [13], there is also dependency in terms of voltage and frequency, which means the lowering of the CPU frequency results in a consequent lowering of voltage, thus reducing power consumption. Scalability is also realized when unused sections of the CPU still release energy is minimized when Power Gating is used to turn off these sections. The basis of this scalability is evidenced by AMD’s and Intel’s capabilities for automatic overclocking. The frequency boost can also be controlled using BIOS, and the user can also change the boost multipliers depending on the CPU.

5.2. Programmability

Caches play a critical role in the overall performance of single-core processors/systems because of the existing gap between the primary memory latency and processor speed. According to [14], first-level caches are restricted strongly by their access time; however, current processors can abide by the majority of their latency through out-of-order execution, in addition to missing overlapping approaches or techniques. In multicore processes/systems, memory hierarchies are important given the increasing number of cores sharing the bandwidth provided by the memory. The reduction in requests in the memory hierarchy means a lot of cycles are necessary to increase them.

Therefore, attempting to make caches more efficient means a lot of memory hierarchies under chip multiprocessors ensure there are last-level caches allocated and integrated across different threads. However, continuing caching techniques, while tackling different caching techniques, will benefit from multicore workloads and platforms. In this case, programmability is necessary

as it guarantees cache coherence approaches and techniques by introducing it to facilitate fast access while protecting the data coherence, although some of the coherence protocols are necessary and important, especially in real-time processors [11]. Programmability also ensures frequent inter-cache communications are facilitated to ensure unpredictable interferences between the multicores are analyzed.

6. CHALLENGES AND FUTURE TRENDS

One of the most significant trends of raising the processor speed is getting an increase in multicore performance. While multi-core processors have numerous advantages, more so for those seeking to increase their power for the multitasking computer system as they provide some processors with fewer executions. [13] observe that each of the cores has its cache, thus the only OS with enough resources, and offers a recognizable improvement to multitask can handle intensive tasks at the same time. However, there are some challenges whenever multicore system/processor architectures are added more cores. One of the challenges is power and temperature [14]. For instance, if two cores were placed on single chips without modifications, the chip would consume double the amount of power and generate a lot of heat. There are also extreme cases where a multicore processor overheats the computer to the point of combustion. This is often attributed to designs running multiple cores at lower frequencies, thus increasing the amount of power consumed.

There is also the issue of starvation. For instance, failure to develop a program correctly for use in multiple processors may result in the starving of more cores for data. The challenge is often evident when there is running of an application with a single thread. As a result, the thread ends up running in a single core while others sat idle. Accordingly, memory service denial is another challenge considering cores sharing the DRAM memory system, but on the same chip share multiple programs [14]. This means that executing on various cores can impact or interfere with the memory access requests of multiple programs, thus end up negatively affecting the performance of another.

In terms of interconnect, the one important feature regarding the multicore chips' performance is communication among various on-chip components such as caches, cores, and memory and network controllers [13]. With private caches on-chip, the exchange of data between running threads means there is a need for off-chip interconnects. In this case, the challenge is often evident when on-chip caches are not shared; therefore, data exchange is not naturally supported in multicores.
Future trends, therefore, considering the above challenges, the integration of more multicores onto dies has provided short-term solutions to a lot of computing issues considering there is a limitation of their utility. Therefore, future trends may consider multicore architectures as stopgaps, which means increasing their usability and functionality means single cores must be integrated into external memory systems [12]. At the same time, there is the possibility different cores would grow based on the bandwidths. There will also be a reduction in single processor performances where computing capabilities will benefit from future technological advancements to use more cores.

7. CONCLUSION

The literature review focused on the architecture of a multicore processor by exploring the concept of multicore technology before documenting the details of the single-core processor; the main area was the multicore processors to determine the basis of its application before determining the history of multicore architecture. Multicore was defined as an architecture

whereby a single physical processor integrates the core logic of two or more processors. The multicore technology concept centers on the parallel computing possibility that can boost computer efficiency and speed by integrating two or more CPUs in a single chip. Lastly, it was established that the improvement of speed and performance acquired through the integration of a multicore processor depends a lot on the software algorithms, including their implementation. Additionally, the focus was also on the processor architecture, where it was characterized by interconnects, core organization, support for parallel programming, including multicore issues such as programmability and scalability.

REFERENCES

[1] Krishna, A. S. V. B. “Evolution of multicore processors.” International Journal of Latest Trends in Engineering and Technology (IJLTET). 2013

[2] Janssen, C. L., Adalsteinsson, H. L., Cranford, S. L., Kenny, J. P., Pinar, A. P., Evensky, D. A., and Mayo, J. A. “A simulator for large-scale parallel computer architectures.” International Journal of Distributed Systems and Technologies. 2010

[3] Cho, K. M., Tsai, C. W., Chiu, Y. S., and Yang, C. S. "A high performance load balance strategy for real-time multicore systems." The Scientific World Journal., 2014.

[4] Schmeisser, M., Heisen, B. C., Luettich, M., Busche, B., Hauer, F., Koske, T., Knauber, K.-H., ... Stark, H. “Parallel, distributed and GPU computing technologies in single-particle electron microscopy. ActaCrystallographica Section. 2009.

[5] Gourdain, N., Montagnac, M., Wlassow, F., &Gazaix, M. “High-performance computing to simulate large-scale industrial flows in multistage compressors.” The International Journal of High-Performance Computing Applications. 2010

[6] Hanford, N., Ahuja, V., Farrens, M., Ghosal, D., Balman, M., Pouyoul, E., and Tierney, B. “Improving network performance on multicore systems: Impact of core affinities on high throughput flows.” Future Generation Computer Systems. 2016

[7] Eicker, N., Lippert, T., Moschny, T., and Suarez, E.: The DEEP Project an alternative approach to heterogeneous cluster-computing in the many-core era.” Concurrency and Computation: Practice and Experience. 2016.

[8] Pyka, A., Rohde, M. and Uhrig, S. “A real-time capable coherent data cache for multicores. Concurrency and Computation: Practice and Experience, vol. 26(6). 2014.

[9] Nanehkaran, Y. A. and Ahmadi, S. B. B. “The challenges of multicore processor.” International Journal of Advancements in Research & Technology, vol. 2(6). 2013

[10] Tomkins, J. L., Brightwell, R. L., Camp, W. J., Dosanjh, S. J., Kelly, S. M., Lin, P. T., Vaughan, C. T., ... Tipparaju, V. T. “The Red Storm Architecture and early experiences with multicore processors.” International Journal of Distributed Systems and Technologies. 2010.

[11] Garcia, V., Rico, A., Villavieja, C., Carpenter, P., Navarro, N., and Ramirez, A. “Adaptive Runtime-Assisted Block Prefetching on Chip-Multiprocessors. International Journal of Parallel Programming.” 2017

[12] Doallo, R. and Plata, O. “Multicore cache hierarchies: Design and programmability issues.” Concurrency and Computation. 2016

[13] Suh, G. E., Rudolph, L., and Devadas, S. “Dynamic partitioning of shared cache memory.” The Journal of Supercomputing. 2004

[14] Jin, S. “A performance study of multiprocessor task scheduling algorithms.” Journal of Supercomputing. 2008.