www.ijecs.in
International Journal Of Engineering And Computer Science
Volume 12 Issue 11 November 2023, Page No. 25894-25899
ISSN: 2319-7242 DOI: 10.18535/ijecs/v12i11.4769

Mobile CPU Architecture Evolution: From Early Designs to Modern Power-Efficient Processors

Priyanshu Ghosh[1], Saikat Das[2], Anirban Bhar[3], Suchismita Maiti[4]

[1,2] B. Tech student, Department of Information Technology, Narula Institute of Technology, Kolkata, India.
[3] Assistant Professor, Department of Information Technology, Narula Institute of Technology, Kolkata, India.
[4] Associate Professor, Department of Information Technology, Narula Institute of Technology, Kolkata, India.

Abstract

The mobile phone is become an essential part of our everyday lives. The processor architecture of mobile phones has undergone substantial modifications due to technological breakthroughs, resulting in the transition from regular mobile phones of the 1990s to contemporary smart phones. Communication, performance, and low-power operation have a significant impact on the design and adoption of mobile processors over time. Mobile handsets now provide a plethora of data services, made possible by the shift from analogue to digital telephony. Processor architecture has grown far more intricate to accommodate these services. Generation after generation, mobile processors continue to expand at a dizzying rate. Examining different mobile phone processing architectures is the objective of this article.

We look at mobile CPU design from the beginning to the present and into the future in this study. We take a look at the ways that changes in mobile CPU designs have affected consumers, designers of hardware, and mobile devices as a whole. We take a look at how ten state-of-the-art mobile CPU designs that came out in the last seven years have changed over time. We detail the evolution of mobile CPU performance, power, energy consumption, and user happiness. To keep up with the power, performance, and energy gains required by mobile computing, specialized accelerators seem to be a viable alternative to desktop-like scaling, which is becoming increasingly difficult for mobile CPUs due to technological limitations. Based on our findings, there are various design considerations that should be made about the CPU's role in future systems on chips (SoCs) due to this paradigm change.

Keywords: Processor architecture, DSP, VLIW, SoC, ARM Processors.

1. Introduction

Aspirational user needs are the engine that propels mobile device design. User expectations are rising for ever-thinner form factors, quicker computing, and longer battery life with each new generation. Mobile system-on-chip (SoC) architectures are doomed to fail if they don't embrace and adapt to the ever-evolving environment of new application use cases, wireless technologies, and sensor capabilities.

The primary components of a smartphone's hardware are application processors, random access memory (RAM), digital signal processing (DSP), central processing unit (CPU), and system-on-a-chip (SoC). Farinaz et al. [1] states that design metrics for mobile phone processors prioritize low power consumption, speed-to-market, and cost. Mobile application processors have certain unique properties, like limited programmability, due to real-time calculation demands, limited power and cost resources, and other factors.

A more intricate user interface, adaptable operating systems, and the ability to offer more services are all requirements for the processor architecture of mobile devices that provide data services. More complex designs may make use of hardware coprocessors or several digital signal processors to meet these supplementary needs.

2. Mobile Processor Architecture: Traditional DSP vs Modern DSP

The introduction of cellular handsets has been characterized by two separate strategies. Two methods exist, one of which makes use of application-specific integrated circuit (ASIC) approaches and the other of which places an emphasis on programmable digital signal processors [2].

Mobile phones make use of DSPs, which are specialized microprocessors. Single, standalone integrated circuits (ICs) formed the backbone of traditional digital signal processors. Now, very large-scale integration designs commonly include embedded digital signal processors. The digital cellular telephony wireless handset market is dominated by programmable digital signal processors.

Analog transmission was the backbone of the first generation of mobile communications (1G systems), which had a number of drawbacks such as high transmission power requirements and a cap on users [3].

Following the initial generation of mobile communication for analog cellular networks, the Global System for Mobile Communications (GSM) standard underwent evolution. When it comes to 2G systems, digital signal processors are among the most crucial types of mobile embedded processors. Since they have a shorter product life cycle and are widely employed in GSM mobiles, DSP architectures are preferred over ASIC. A programmable digital signal processor (DSP) offers a versatile and inexpensive design for mobile phones. In 1979, AT&T introduced the first digital signal processor (DSP), and later on, Texas Instruments developed other DSPs.

Because it allows for high memory bandwidth and many operand operations, this architecture reduces the number of cycles needed to execute a certain function. Commonly linked to digital signal processor (DSP) designs are multiply-accumulate (MAC) commands.

Some new digital signal processor (DSP) designs have emerged specifically for mobile devices, in addition to more conventional ones. A state-of-the-art digital signal processor (DSP) with a Harvard design that uses one code read bus and three data read buses is the TMS320C55 [3]. Enhancing CPU utilization at high speeds, the TMS320C55 DSP architecture incorporates features such as configurable idle modes and automated power conservation.

VLIW (Very long instruction word) digital signal processors are examples of which the TMS320C62XX is one. A compiler-based environment that is friendly to programmers is provided by the VLIW architecture in DSP. These very large instruction set (VLIW) CPUs are designed using the EPIC architecture.

An example of one of the many cutting-edge aspects of the TigerSHARK DSP architecture is its usage of "short vectors" for processing data via the SIMD architecture [2].

DSPs' real-time functioning and low power consumption have made them popular in mobile devices. The alignment of future mobile devices must take computing power requirements into account as they integrate more functionality. Improvements in digital signal processing allow mobile phones to run at higher clock frequencies while using less electricity per MIPS [4].

Parallelism in superscalar architectures is one method of customisation that will be widely used in future mobile CPUs. Due to their ability to decrease CPU chip frequency and voltage without sacrificing performance, VLIW and SIMD architectures are gaining popularity in today's cellular handsets. Having the ability to handle parallel processing allows modern DSPs to be more effective.

Improvements in chip fabrication have led to a dramatic increase in the computing power available in

modern DSP systems. Table 1 shows that in 1980, the identical DSP chip could only handle 5 MIPS (Milli-Instructions-Per-Second), but by 2010, that number had increased to 50 GIPS. Even when other considerations are taken into account, the progress made in DSP integration is noteworthy.

To make the most of VLSI resources, several FPGA (Field-programmable gate array) designs have been developed for DSP. The use of VLIW compilers to provide instruction level parallelism is widespread in DSP designs. Creating DSP systems that are both dependable and resistant to faults is a current topic in DSP architecture design. Performance, power, reliability, cost, and development time are just a few of the design criteria that are considered while reconfiguring DSP systems. These days, reconfigurable DSP architectures constitute the foundation of modern DSP systems.

Duplicating the processor cores allows current DSPs to expand their architecture. The SIMD operations are used by enhanced DSPs, whereas VLIW or superscalar architectures can be implemented by multiple-issue DSPs.

3. SoC Based Architecture

Thanks to SOC designs, the architecture of mobile device processors became very simple. With the help of a better DSP hybrid chip, mobile devices can control their responsiveness in real time. Low power functioning in mobile devices is made possible by lowering the chip's voltage. It was suggested by Matthias et al. [5] to use a novel scalable DSP architecture in system-on-chip (SoC) applications. Integrating microcontrollers, dedicated ASICs, or DSPs on a single chip allows system job management in SoC based architectures.

Super integrated circuits (SoCs) that use multicore technology to achieve low power consumption and excellent performance have recently appeared. Architectural options are frequently constrained by low power operation. A quick memory system, such as cache memories, is necessary for mobile devices to implement VLIW designs with high throughput.

Instruction set adjustments have been made by numerous organizations to speed up the use of mobile devices.In order to facilitate additional read-write operations, ARM Ltd. has performed substantial instruction set modification by encoding the most often used instructions in 16-bit.

Figure 3 shows how changing the instruction set affects performance and memory usage. By modifying the instruction set to use 8-bit data types instead of 32-bit ones, we can boost performance and memory usage.

A reconfigurable CPU architecture for mobile phones was presented by Martin at el [6]. Modular System on Chip (CSoC) designs have been fine-tuned for use in mobile networks. Core logic, RAM, application-specific integrated circuit (ASIC) cores, and reconfigurable hardware units (RHUs) are the building blocks of a custom-designed system-on-chip (CSoC).

Today, the majority of smartphones use system on chips (SoCs) with one or two cores. The majority of mobile applications can be outperformed by dual-core CPUs, which are speedier, as opposed to quad-core system on chips. As time goes on, mobile system on chips will get smarter, which will boost performance everywhere.

4. ARM Mobile Processors

Nowadays, most smart phones employ ARM-based processors. The ARM architecture is a RISC-based 32-bit instruction set [7]. The low power consumption and excellent performance of ARM processors make them ideal for usage in smart phones.

Apple, Qualcomm, and other third-party vendors build their products based on the architecture given by ARM holdings, which includes chip design and instruction set customisation licenses.

Smartphones with lower-end hardware often employ the ARMv5 architecture, whereas more recent high-performance devices have made use of the ARMv6 and ARMv7 architectures. A hardware floating-point

unit (FPU) is a part of ARMv7 that makes it faster. One of the most common types of mobile device architectures is the 32-bit ARM, which includes ARMv7-A.

Android, iOS, Windows Phone, Windows RT, Bada, Blackberry OS/Blackberry10, MeeGo, Firefox, Tizen, Ubuntu Touch, Sailfish, and Igelle are just a few of the mobile operating systems that rely on the ARM architecture.

5. Mobile CPU in focus of Specialization

Mobile CPU design is falling behind mobile technologies. Due to the dark silicon problem (transistor density exceeding what a single chip can fully turn on), specialization is a possible method for sustaining the power, performance, and energy increases needed for future computer systems. Mobile SoCs already have numerous specialized processing modules, and that number will grow. The six latest Apple SoCs included 3.5X more fixed-function accelerators [7], and the ITRS predicts thousands of on-chip accelerators by 2022 [8].

Future mobile systems will use CPUs differently due to specialization. This section examines three primary CPU responsibilities in future mobile systems and their design consequences. The growth of specialized processing units will minimize CPU compute duty while increasing system complexity for many applications. The CPU will remain a target for code portability and backwards compatibility for other programs. Irregularity Engine Parallelism-rich computing domains are best for acceleration, leaving CPU-intensive regions. The heterogenous system architecture (HSA) framework, which aims to program future accelerator-rich SoCs, uses a single-instruction multiple thread (SIMT) programming style that implicitly implies parallel calculations. The majority of programs cannot be fully performed on accelerators and require CPU computing.

The CPU will be a “irregular code accelerator.” As mobile CPUs handle more irregular code, many standard CPU performance increase strategies will likely be used less than they would have been without accelerators. A recent study on CPU-GPU calculations [9] found that instruction window size and stride-based memory prefetchers were less effective in CPU-GPU workloads than in typical CPU workloads. ILP and locality, which both methods require, were offloaded to the GPU. As recommended in [9], stronger branch predictors, prefetching methods, cache management strategies, and memory structure optimizations might help future CPUs extract ILP and locality.

Organizer System CPUs alone run operating system, runtime framework, and device driver code. This paradigm increases system complexity for the CPU to manage at runtime by increasing the number of SoC processing modules. As more CPU operations are offloaded to accelerators, more CPU time will be spent on system orchestration tasks like compute scheduling, resource provisioning, data transportation, and system monitoring.

System administration is inconsistent and unpredictable but essential to the program. These traits collide with future mobile CPU design. Modern mobile CPUs optimize power efficiency during long inactive times. Most optimizations reduce responsiveness. Disabling CPU subsystems like LLCs saves power in idle CPUs, but later CPU computations suffer the performance penalty to restore state. Making CPUs aware of accelerator execution characteristics to adapt is promising. This notion has been shown with CPU-GPU computations [10] for DVFS, but it can be expanded to other accelerators and used in prediction techniques for other CPU subsystems, such as memory prefetching. Legacy Code Goal Despite trends toward increasing hardware specialization in future mobile SoCs, many key apps will continue to run on the CPU. CPUs are the most accessible mobile systems programming interface among processing technologies. Most developers use CPU-targeted programming languages, and all SoCs have a CPU subsystem. Therefore, CPUs are crucial to mobile platform backwards compatibility and code portability.

Today's CPU-level heterogeneity trends should persist in future CPUs since they will execute both extremely irregular and regular applications. CPU-level heterogeneity allows CPU designs suitable for programs with these two fundamentally distinct execution characteristics. Current heterogeneous CPUs, like ARM's huge. LITTLE technology [11] offers out-of-order and in-order CPU core designs for power-performance trade-offs. This model can be extended to optimize for common CPU demands like the Android software stack [12] and Web browser [13].

6. Conclusion

Various suppliers are focusing on the future of mobile computing to provide architectures for mobile processors that use less power. Various cellular companies give ARM-based mobile processors fancy names, but they're all essentially the same. Smartphones with improved graphics processing unit (GPU) cores, memory interfaces, and other cutting-edge capabilities will be available in the near future thanks to updated mobile CPUs. In order to boost device performance, future mobile system on chips will investigate next-generation CPU architecture. The makers of mobile processing units are putting a lot of effort into creating more powerful mobile phones. New methods in processor architecture design are required to accommodate data-centric mobile devices of the next generation. Still, improvements in areas like battery efficiency, UI speed, time to market, etc., are propelling mobile CPU development.

By actively embracing desktop-like design principles, mobile CPU designs have improved over the past decade to deliver end-users satisfactory experiences. However, present methods for designing mobile CPUs do not provide the anticipated performance needed within the strict energy and thermal limitations that will be imposed by increasingly complicated and computationally intensive mobile software in the future. Throughout the specialization age, our article argues, mobile CPU design methodologies must be rethought by future mobile hardware designers in order to prepare for their position in system-on-chips (SoCs).

References

1. Farinaz Kaushanfar, Vandana Prabhu, Miodrag Potkonjak, Jan M. Rabaey: “Processors for Mobile Applications”.
2. Alan Gatherer, Trudy Stetzler, Mike McMahan, and Edgar Auslander, Texas Instruments: “DSP-Based Architectures for Mobile Communications: Past, Present, and Future”.
3. Dr. Margarita Esponda: “Trends in Hardware Architecture for Mobile Devices”.
4. Shiv Chaturvedi: “The role of digital signal processors (DSP) for 3G mobile communication systems”, International Journal on Emerging Technologies 1(1): 23-26(2010).
5. Matthias H. Weiss, Frank Engel, and Gerhard P. Fettweis: “A new Scalable DSP Architecture for System on Chip (SOC) Domains”, IEEE International Conference on Acoustics, Speech, and Signal Processing.
6. Martin Vorbach, Gurgen Becker: “Reconfigurable processor architectures for mobile phones”, Parallel and Distributed Processing Symposium, 2003. Proceedings. International.
7. Leonid Ryzhyk: “The ARM Architecture”.
8. System Drivers, ITRS, 2013. [Online]. Available: http://www.itrs.net/ITRS%201999-2014%20Mtgs,%20Presentations%20&%20Links/2013ITRS/2013Chapters/2013SysDrivers_Summary.pdf
9. M. Arora, S. Nath, S. Mazumdar, S. B. Baden, and D. M. Tullsen, “Redefining the role of the cpu in the era of cpu-gpu integration,” Micro, IEEE, 2012.
10. K. Ma, X. Li, W. Chen, C. Zhang, and X. Wang, “Greengpu: A holistic approach to energy efficiency in gpu-cpu heterogeneous architectures,” in Proc. of ICPP. IEEE, 2014.

11. Big.LITTLE Technology. [Online]. Available: https://www.arm.com/products/processors/technologies/biglittleprocessing.php
12. G. Venkatesh, J. Sampson, N. Goulding, S. Garcia, V. Bryksin, and M. B. Taylor, “Conservation cores: Reducing the energy of mature computations,” in ACM SIGARCH Computer Architecture News. ACM,2010.
13. Y. Zhu and V. J. Reddi, “Webcore: architectural support for mobile web browsing,” in Proc. of the ISCA, 2014.