Multicore Processor Challenges – Design Aspects

[1]Rahul V Mehta, [2]Dr.Kiritkumar R. Bhatt, [3]Dr.Ved Vyas Dwivedi

[1]Research Scholar, [2]Principal Sardar Patel College of Engineering, Anand, [3]Vice Chancellor, Gokul Global University
[1]C U Shah University, Electronics and Communication Department, Surendranagar
[1]India

Abstract : Day by day processor's computational performance is increasing with complexities of tasks it needs to execute. Processor should be capable enough to have higher performance without compromising power consumption, functionality and thermal effects. Designers have come up with alternative architectures like parallel processing, scheduling of processes, multi core architecture with complex switching interfaces etc. Multi-core architecture has become choice of designers as it has excellent instruction execution capabilities with better performance, reduced power consumption and excellent handlining of complex process like multi-tasking. In this paper the challenges faced by designer are explored.

***Keywords* – Cache, Multicore, Coherence.**

I. INTRODUCTION

From the evaluation of the microprocessor, industry continues to have great importance in technological advancements of its architecture [3]. The growing market and the demand for faster performance drove the industry to manufacture faster and smarter chips. One of the most classic and proven techniques to improve performance is to clock the chip at higher frequency which enables the processor to execute the programs in a much quicker time [4, 5] and the industry has been following this trend from 1983 – 2002 [6]. Additional techniques have also been devised to improve performance including parallel processing, data level parallelism and instruction level parallelism which have all proven to be very effective [1]. One such technique which improves significant performance boost is multi-core processors. Multi-core processors have been in existence since the past decade, but however have gained more importance off late due to technology limitations single-core processors are facing today [7] such as high throughput and long lasting battery life with high energy efficiency [6].

II. MULTI-CORE PROCESSORS

A multi-core processor is a single computing component with two or more independent actual central processing units (called "cores"), which are the units that read and execute program instructions. The instructions are ordinary CPU instructions such as add, move data, and branch, but the multiple cores can run multiple instructions at the same time, increasing overall speed for programs amenable to parallel computing. Manufacturers typically integrate the cores onto a single integrated circuit die (known as a chip multiprocessor or CMP), or onto multiple dies in a single chip package.

The Need For Multicore CPU The high performance speed achieved by multi-processors (multiple CPUs on different chips attached to the same motherboard), produce undesirably high power consumption, and as a result, alternative research trends encouraged the production of multicore CPUs in order to reduce power consumption, while simultaneously increasing the processing speed. The architecture of multicore CPUs provided the hungry applications and devices, speed and performance with lower power consumption.

There are many advantages of multicore processors like Multicore processors can finish more work than single-center processors, They can finish synchronous work as low recurrence. They can deal with more information than single-center processors. Complex works like filtering can be achieved easily with complex instructions. Needs less space in case of utilizing multi-core processors.

Multicore Processors can be Homogenous or Heterogeneous. Homogenous supports all identical processor cores can support same instruction set architecture (ISA). MPC8641, Intel Core Duo are Homogenous processors. Heterogeneous Multi-Core Processor have all non-identical processor cores which can support different instruction set architecture (ISA). For example – Intel CE 2110 Media because it is comprised of Intel Xscale processor core and an Intel Micro Signal Architecture (MSA) DSP core.

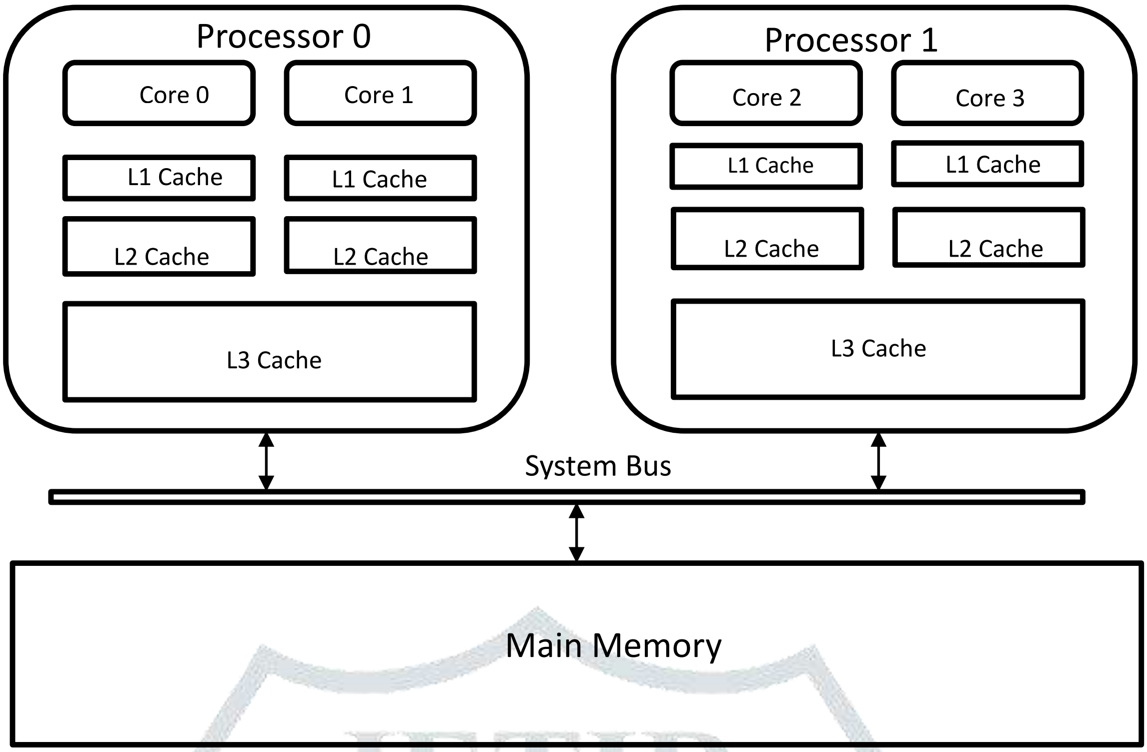

Fig: Multi-core processor architecture with 3 level Cache Memory

III. Multi-core Challenges Conclusion

In spite of the many advantages that multi-core processors come with, there are a few major challenges the technology is facing.

A. Memory Hierarchy:

Memory System for multicore processor is one of the major challenges that designers are facing. The program execution is often limited by the memory bottleneck which happens not due to the limitations of processors or its low speed but because a heavy portion of applications always lies in main memory until it is executed by the processor. Also, care should be taken to conserve memory bandwidth and avoid memory contention. It is required to note that the memory hierarchies in parallel machines are more difficult than in the single-processor's system, especially in multi-core processors where L2 and L3 cache are shared by the multiple cores within a chip. This leads to more complicated memory hierarchy design in CMPs.

B. Cache Levels:

Three level cache schemes are trending in current generation of multi core processor. Dual, quad and eight core processors are having L1, L2 and L3 cache. L1 cache is the fastest memory in the computer and closest to the processor it is also known as the "primary cache. The L2 cache feeds the L1 cache, which feeds the processor. L2 memory is slower than L1 memory. the L3 cache is generally shared among all the processing cores. Each subsequent cache is slower and larger than L1, and instructions and data are staged from main memory to L3 to L2 to L1 to the processor. The L3 cache feeds the L2 cache, and its memory is typically slower than the L2 memory, but faster than main memory. The L3 cache feeds the L2 cache, which feeds the L1 cache, which feeds the processor. In this cache model, the L2 cache may be private, shared or split. But as number of cores increases, this may cause the bottleneck and result in data traffic congestions and performance degradation.

C. Cache Coherence:

When data are updated in a cache but not yet transferred to the target memory or disk, the chance of corruption is greater. It very challenging to manage a cache so that data are not lost or overwritten. Let’s take an example of a dual-core processor where each core transfers a block of data from memory into its private cache. First core writes a value to a specific memory location; when the second core tries to read that value from its cache it will not have the updated copy unless a cache miss occurs. This cache miss forces the update operation on the second cores cache entry. If this policy would not have been in place then invalid results would have been produced. Since the shared and private caches exist at different levels in multi-core processors, this cache coherence problem becomes worse. This issue of Cache Coherency can be accomplished by well-designed algorithms that keep track of every read and write event; cache coherency is even more critical in symmetric multiprocessing (SMP) where memory is shared by multiple processors [16].

D. Developing Multicore Software:

"Applications on multi-core systems don't get faster automatically as cores are increased" [7]. Program runs a bit slower on multi-core processors as compared to single core processors. Moreover, majority of applications used today are written to run on a single core processor, which failed to use the capability of multi-core processors. Hence, the dignified challenge the industry faces is how to port these software programs which were developed years ago to multi-core adaptable software programs. Although it sounds very possible to redesign programs, it's really not a business profitable decision in today's world where in companies have to keep in mind the key parameters like time to market, customer fulfilment and cost cut. It has however been correctly pointed out that "The throughput, energy efficiency and multitasking performance of multicore processors will all be fully realized when application code is multi-core ready" [8]. Program will not run any faster unless one employ concurrency. Making programs efficiently concurrent is no simple task. If done poorly, making your program concurrent can actually make isystem slower. For example, if you spend lots of time spawning threads, and do work on a very small chunk size, or if you frequently synchronize your data , or if you frequently write to data in the same cache line between multiple threads , then you can seriously harm the performance with concurrent programming.

E. Level of Parallelism:

The level of parallelism of the process is one of the gigantic factors that affect the performance of a multicore processor significantly. Performance will increase with the decrease of completion time of a process. Parallelism can be achieved by Instruction Level Parallelism and Thread Level Parallelism. TLP increases overall parallelism by breaking a program into many small threads and execute them concurrently. Hence to achieve a high level of parallelism, software developers must write such algorithms that can take full advantage of multicore design. Also, companies like Microsoft and Apple have designed their operating systems which can run efficiently on multicore configuration [10] [19].

F. False-Sharing:

In same memory address region if two different processors is operating on a independent data that resides on the same cache line and the mechanism may force the whole line to interconnect with every data write, results in the wastage of system bandwidth is considered as False-sharing. In multi-core processors, as the number of cores increases, the number and size of private and shared caches will also increase, and this might result in increase of false sharing [18].

G. Power and Temperature:

As and when we increase number of cores in a chip, it will consume more power as they may generate large amount of heat if the processor is not modified for heat which may result in malfunctioning of a chip. Cores with consumes more power and generates more heat is called hot-spot core. To mitigate this there are many designs and techniques are developed. Some multicore designs incorporate a power control unit that can force the unused cores to shut down that are not required at times known as Power getting. The chip is architected such that the amount of heat generated in the chip is well distributed across the chip. There are various ways to tackle the problem of power dissipation and temperature which includes thread migration, DVFS (Dynamic Voltage and Frequency Scaling) [21], Clock Getting, power getting etc. In the thread migration technique, a low power consuming process or thread is moved to an overheated core. In DVFS (Dynamic Voltage and Frequency Scaling) technique, voltage and frequency of the hot core is reduced since the power dissipation or heating is a function of both voltage and frequency which slightly affects the overall performance also [10] [19].

H. Communication Minimization or Interconnect Issues:

The program execution and its performance in multi-core processors is impacted when the interaction between on chip components viz. cores, cache, memories and if integrated—memory controllers and network controllers which are used for memory-memory and memory-processor communication, where bus contention and latency are the key areas of concern. As number of core increases respective caches increases which will increase inter memory and memory- processor communication also increases exponentially. So, the memory hierarchy of multi-core processors should be designed in such a way that these communications are contained. It is rightly pointed that: "The performance of the processor truly depends on how fast a CPU can fetch data rather than how fast it can operate on it to avoid data starvation scenario" [10]. Special crossbars or mesh technologies have been enforced on hardware to solve this issue. For instance, AMD CPUs employ a crossbar, and the Tilera TILE64 implements a fast non-blocking multi-link mesh [11].

IV. CONCLUSION

Before multi-core processors the performance increase from generation to generation was easy to see, an increase in frequency. This model broke when the high frequencies caused processors to run at speeds that caused increased power consumption and heat dissipation at detrimental levels. Adding multiple cores within a processor gave the solution of running at lower frequencies, but added interesting new problems. Multi-core processors are architected to adhere to reasonable power consumption, heat dissipation, and cache coherence protocols. However, many issues remain unsolved. In order to use a multi-core processor at full capacity the applications run on the system must be multithreaded. There are relatively few applications written with any level of parallelism. And finally the memory systems and interconnection networks also need improvement.

REFERENCES

[1] Goodacre, J. & Sloss, A.N. 2005, "Parallelism and the ARM instruction set architecture", Computer, vol. 38, no. 7, pp. 42-50.
[2] D. T. Marr, F. Binns, D. L. Hill, G. Hinton, D. A. Koufaty, J. A. Miller and M. Upton, "Hyper-Threading Technology Architecture and Microarchitecture," Intel Technology Journal, vol. 6, pp. 1, 02/14, 2002.
[3] Aspray, W. 1997, "The Intel 4004 microprocessor: what constituted invention?", Annals of the History of Computing, IEEE, vol. 19, no. 3, pp. 4-15.
[4] Cass, S. (2010). Multicore Processors Create Software Headaches. Technology Review, 113(3), 74-75.
[5] Parkhurst, J., Darringer, J. & Grundmann, B. 2006, "From Single Core to Multi-Core: Preparing for a new exponential", Computer-Aided Design, 2006. ICCAD '06. IEEE/ACM International Conference on, pp. 67.
[6] R. Ramanathan. Intel multi-core processors: Making the move to quad-core and beyond. Technology@Intel Magazine, Dec 2006.
[7] Lizhe Wang, Jie Tao, Gregor von Laszewski, Holger Marten. 2010, “Multicores in Cloud Computing: Research Challenges for Applications”, Journal of Computers, Vol 5, No 6 (2010)
[8] Cass, S. “Multicore Processors Create Software Headaches,” Technology Review, vol. 113(3), pp. 74-75, 2010.
[9] R. Kumar, D. M. Tullsen, N. P. Jouppi and P. Ranganathan, "Heterogeneous chip multiprocessors," Computer, vol. 38, pp. 32-38, 2005.
[10] Anil Sethi, Himanshu Kushwah,”Multicore Processor Technology-Advantages and Challenges,” International Journal of Research in Engineering and Technology, vol. 04(09), pp. 87-89, 2015.
[11] Yaser Ahangari Nanehkaran, Sajjad Bagheri Baba Ahmadi,” The Challenges of Multi-Core Processor,” International Journal of Advancements in Research & Technology, vol. 2(6), pp. 36-39, June 2013.
[12] J. Yan and W. Zhang. “Hybrid multicore architecture for boosting single-threaded performance”, In ACM SIGARCH Computer Architecture News, Vol 35, No.1, pp. 141-148, March, 2007.
[13] Sodan, A., Machina, J., Deshmeh, A., Macnaughton, K. & Esbaugh, B. 2010, "Parallelism via Multithreaded and Multicore CPUs", Computer, vol. PP, no. 99, pp. 1-1.
[14] Brooks, D.M., Bose, P., Schuster, S.E., Jacobson, H., Kudva, P.N., Buyuktosunoglu, A., Wellman, J., Zyuban, V., Gupta, M. & Cook, P.W. 2000, "Power-aware microarchitecture: design and modeling challenges for next-generation microprocessors", Micro, IEEE, vol. 20, no. 6, pp. 26-44.
[15] Y. Hoskote, S. Vangal, A. Singh, N. Borkar and S. Borkar, "A 5-GHz Mesh Interconnect for a Teraflops Processor," Micro, IEEE, vol. 27, pp. 51-61, 2007.
[16] Roy, A., Jingye Xu & Chowdhury, M.H. 2008, "Multi-core processors: A new way forward and challenges", Microelectronics, 2008. ICM 2008. International Conference on, pp. 454.
[17] http://www.ibm.com/ibm/history/ibm100/us/en/icons/power4/; access date: April 23, 2021.
[18] Hironori Kasahara , Keiji Kimura ,“Multicore Cache Coherence Control by a Parallelizing Compiler”, IEEE 41st Annual Computer Software and Applications Conference (COMPSAC), July 2017
[19] V. Shanker, “Optimization of a Parallel Application for Multi-Core Environments,” [Online].
[20] Alexandra Fodor; Gabriel Chindris; Rajmond Jano; Dan Pitica “Thermal Modelling and Simulation Techniques for Multicore Processors”, 2019 42nd International Spring Seminar on Electronics Technology (ISSE) IEEE 2019
[21] Shervin Hajiamini; Behrooz Shirazi; Aaron Crandall; Hassan Ghasemzadeh “A Dynamic Programming Framework for DVFS-Based Energy-Efficiency in Multicore Systems” IEEE Transactions on Sustainable Computing March 2020