International Journal For Technological Research In Engineering
Volume 4, Issue 1, September-2016 *ISSN (Online): 2347 - 4718*

INTRICACIES IN DEVELOPMENT OF INTER CONNECTION NETWORKS FOR MULTI-CORE PROCESSORS

Dr. Ajay Mathur[1], Ashish Sharma[2]
[1]HoD (C.S.E.) Govt. Polytechnic College Jodhpur, [2]HoD (C.S.E) Jodhpur National University

Abstract: This paper studies the intricacies in terms of space complexity, energy consumption, time complexity and on-chip interconnection based architectural issues in multiprocessors. A core simply implies to a processor unit that can read instructions & perform specific task. The available designs for the interconnects are found to have differential effects on its chip. Our research aims at finding difficulties during designing of interconnects that can be independently implemented practically. Many of the cases are reviewed for the need for careful co-design. Such as expanding interconnect band width requires space that then constrains the number of cores or cache sizes, and does not necessarily increase performance. It is crucial for increasing the speed of processor to get a boost for performance in a Multi-core architecture. Thus the design of processors are required to be dynamically analyzed every time manufacturers are fabricating them. Our main goal is to illustrate important challenges while developing Interconnection Networks for multi-core processors.

I. INTRODUCTION

A core simply implies to a processor unit that can read instructions & perform specific tasks. Instructions are chained together to run in real time& make up a computer we experience today. Examples from daily lifethat requires processing core are opening a folder, typing into a word document. Other things like drawing thedesktop environment, the windows, and game graphics are the job of your graphics card which contains hundreds of processing cores to quickly work on data in parallel.

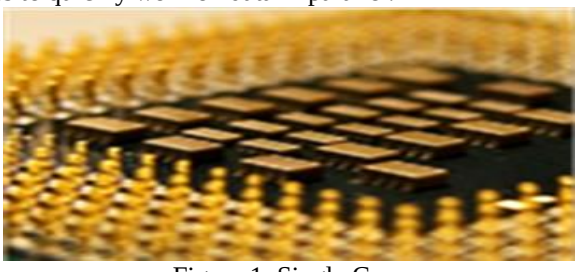

Figure 1: Single Core

To some extent they require computer's processing core as well. The designs of cores are extremely complex and vary widely between companies and even models.

II. MULTICORE PROCESSORS

Multicore processor architecture entails silicon design engineers placing two or more execution cores, or computational engines, within a single processor package. This multicore processor plugs directly into a single processor socket, but the operating system perceives each of its execution cores as a discrete logical processor with all the associated execution resources. Multicore chips who do more workper clock cycle, are able to run at a lower frequency, and may enhance user experience in several ways such as improving performance of compute and band width in tensive activities.

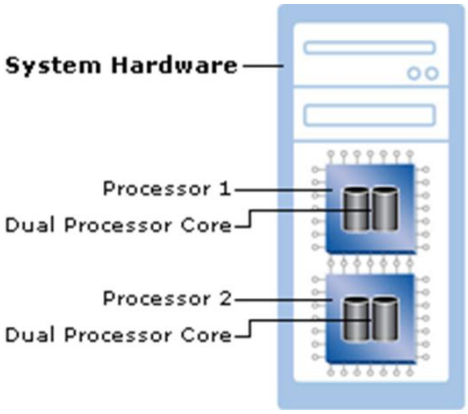

Figure 2: Multicore Processor

2.1WORKING OF MULTICORE PROCESSOR

The architecture of cores by manufacturers are constantly being improved to pack in the most amount of performance in the least amount of space and energy consumption. But despite all the architectural differences, processors go through four main steps whenever they process instructions: fetch, decode, execute, and write back.

•Fetch:In this step, the processor core retrieves instructions that are waiting for it, usually from some sort of memory

•Decode: Once it has fetched the immediate instruction, it goes on to decode it. Instructions often involve multiple areas of the processor core such as arithmetic and the processor core needs to figure this out.

•Execute: The execute step is where the processor knows what it needs to do and actually goes ahead and does it.

•WriteBack: The final step, called writeback, simple places the result of what's been worked on back into memory.

This entire process is called an instruction cycle. . These instruction cycles happen ridiculously fast, especially now that we have powerful processors with high frequencies.

The following changes are made when multicore are developed:

2.2 Moore's Law

1965: Intel's Gordon Moore predicted that the number of transistors on a chip would double every 12 months into the near future (he later refined this, in 1975, to every two years)

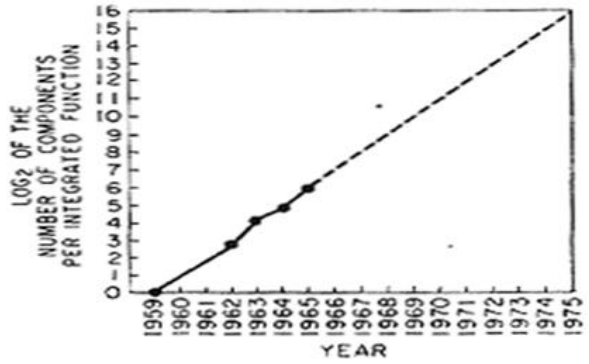

Figure 3:Evolution Of Transistor

2.3 Transistor Count

The constant decrease in feature size leads to an increase on transistor count on chip area which enables designing of more complex processors.

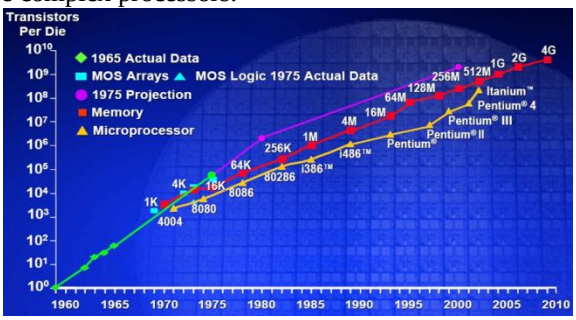

Figure 4:Increasing Transistor Counts

2.4 Power

This amount of transistors on the chip area increases power consumption and produces more heat.

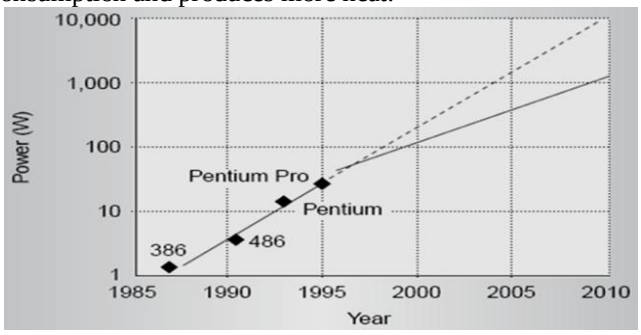

Figure 5: Increasing Power Consumption

2.5 Power v/s Frequency

Power consumption and heat limited using frequency as a way of improving processor performance is showing gradual increase.

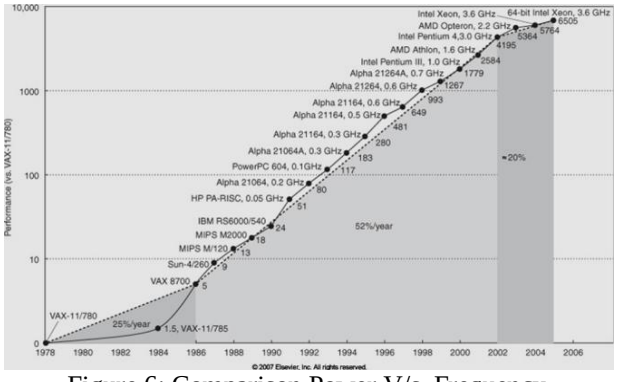

Figure 6: Comparison Power V/s Frequency

III. INTERCONNECT FUNDAMENTALS

Interconnection framework concerns with how multicore components interacts with the rest of the architecture in a system .For a given number of cores, the "best" interconnection architecture in a given chip multi processing environment depends on factors such as

- Performance objectives,
- Power/area budget
- Bandwidth requirements,
- Technology, and even the system software

Interconnection networks can be categorized according to criteria such as topology, routing strategies, and switching techniqueswhich are fixed and reconfigurable.

Routing strategies are procedures used to set switches and plays a crucial role in the performance of multistage interconnection networks.

Switching techniques are ways that data packets are handled on their way from a source to a destination processor.

Topology is the pattern in which the individual processors are connected to other elements .Below figure shows three commonly used on-chip topologies.

Figure 7 show , a ring topology (Fig.:a),mesh topology (Fig:b) and torus topology (Fig:c).

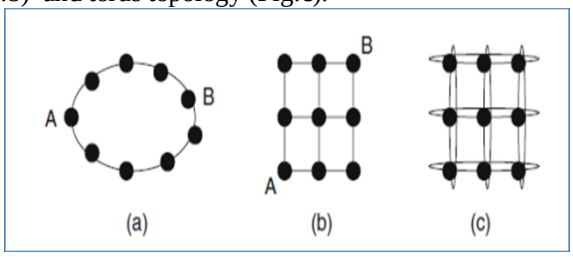

Figure 7: (a).Ring (b).Mesh (c)Torus

IV. CHALLENGES UNDER INTERCONNECTION NETWORK IN MULTICORE PROCESSOR

4.1 Power management

If two cores were placed on a single chip without any modification, the chip would, in theory, consume twice as much power and generate a large amount of heat. Run the multiple cores at a lower frequency to reduce power consumption. Incorporate a power management unit that has the authority to shut down unused cores or limit the amount of power.

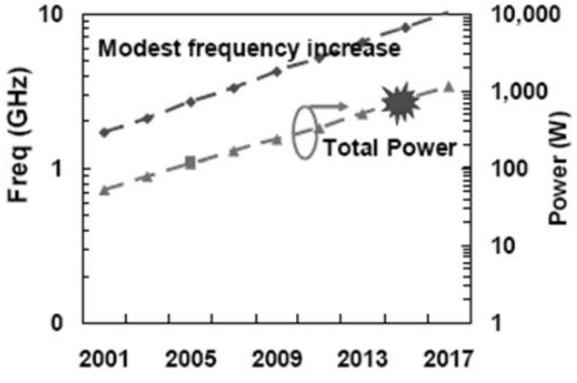

Figure 8: Power Management

4.2 Temperature Management

The chip is architected so that the number of hot spots doesn't grow too large and the heat is spread out across the chip .The CELL processor follows a common trend to build temperature monitoring into the system, with its Temperature management unit

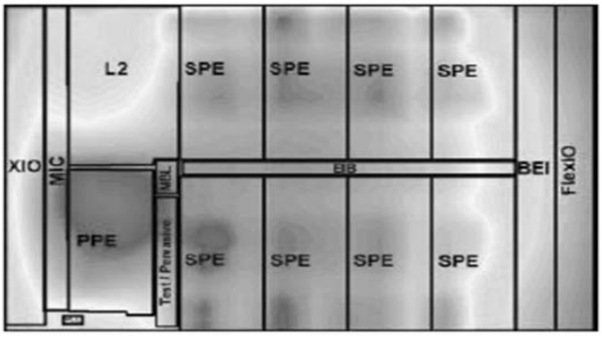

Figure 9: Temperature Management

4.3 Cache Coherence

Cache coherence is a concern in a multicore environment because of distributed L1 and L2 cache. Since each core has its own cache, the copy of the data in that cache may not always be the most up-to-date version. If a coherence policy wasn't in place garbage data would be read and invalid results would be produced, possibly crashing the program or the entire computer.

4.4 Multithreading

The most important, issue is using multithreading or other parallel processing techniques to get the most performance out of the multicore processor. Rebuilding applications to be multithreaded means a complete rework by programmers in most cases to write applications with subroutines able to be run in different cores. Applications should be balanced. If one core is being used much more than another, the programmer is not taking full advantage of the multicore system

4.5 Homogeneous vs. Heterogeneous Cores

Homogeneous cores are easier to produce since the same instruction set is used across all cores and each core contains the same hardware. Each core in a heterogeneous environment could have a specific function and run its own specialized instruction set. This model is more complex, but may have efficiency, power, and thermal benefits that outweigh its complexity.

4.6 Parallel Programming

In May 2007, Intel fellow Shekhar Borkar stated that "The software has to also start following Moore's Law, software has to double the amount of parallelism that it can support every two years." Since the number of cores in a processor is set to double every 24 months.programmers need to learn how to write parallel programs that can be split up and run concurrently on multiple cores instead of trying to exploit single-core hardware to increase parallelism of sequential programs.

4.7 Improved Memory System

With numerous cores on a single chip there is an enormous need for increased memory. 32-bit processors, such as the Pentium 4, can address up to 4GB of main memory. With cores now using 64-bit addresses the amount of addressable memory is almost infinite. An improved memory system is a necessity; more main memory and larger caches are needed for multithreaded multiprocessors.

4.8 Starvation

If a program isn't developed correctly for use in a multicore processor one or more of the cores may starve for data. This would be seen if a single-threaded application is run in a multicore system. The thread would simply run in one of the cores while the other cores sat idle. This is an extreme case, but illustrates the problem.

V. CONCLUSION

The best topology for an on-chip network withtechnology constraints in multicore processor followed by:

(i). The effect of a topology on overall network cost–performance.

(ii).Highlight the various tradeoffs between performance and power.

(iii).Highlight the various tradeoffs between performance and space.

(iv). Study of overhead of topologies, number of cores and on chipmemory size.

REFERENCES

[1] D. Pham, S. Asano, M. Bolliger, M. Day, H. Hofstee, C. Johns,J. Kahle, A.Kameyama,J. Keaty, Y. Masubuchiet al., "The designand implementation of a first-generation cell processor," in Solid-StateCircuits Conference, 2005. Digest of Technical Papers. ISSCC. 2005 IEEE International. IEEE, 2005, pp. 184–592.

[2] B. B. Brey, The Intel Microprocessors. Prentice Hall Press, 2008.

[3] D. Geer, "Chip makers turn to multicore processors," Computer, vol. 38, no. 5, pp. 11–13, 2005.

[4] R. Kumar, D. Tullsen, P. Ranganathan, N. Jouppi, and K. Farkas, "Singleisa heterogeneous multi-core architectures for multithreaded workload performance," in ACM SIGARCH Computer Architecture News, vol. 32 ,no. 2. IEEE Computer Society, 2004, p. 64.

[5] J. Bui, C. Xu, and S. Gurumurthi, "Understanding performance issues on both single core and multi-core architecture," Technical report, University of Virginia, Department of Computer Science, Charlottesville, Tech. Rep.,2007.

[6] R. Ubal, J. Sahuquillo, S. Petit, P. L´opez, Z. Chen, and D. Kaeli, "The multi2sim simulation framework.

[7] K. Hwang, Advanced computer architecture. Tata McGraw-Hill Education, 2003.

[8] R. Ubal, J. Sahuquillo, S. Petit, and P. L´opez, "Multi2sim: A simulation framework to evaluate multicore-multithread processors," in IEEE 19th International Symposium on Computer Architecture and High Performance computing, page (s), 2007, pp. 62–68.

