UNIVERSITY OF JORDAN

COMPUTER ENGINEERING DEPARTMENT

REPORT RESEARCH

TITLE

A Comparative Study on Heterogeneous and Homogeneous Multiprocessors

---

Abeer Hyari

Submission date: 2-DEC-2009

ABSTRACT

Nowadays there is an increase interests in the design of multicore processor in order to improve system throughput, and reducing processor power. These multicore maybe of the same characteristics they called "Homogeneous", or have different characteristics they called "Heterogeneous". This report presents a comparison between homogeneous multicore processor and heterogeneous one. This report examine in details the performance, throughputs and power consumption by the two different architecture. Also, it presents some actual examples of Differences in architectures for Intel's Core 2 Duo, Advanced Micro Devices' Athlon 64 X2, Sony-Toshiba- IBM's CELL Processor, and finally Tilera's TILE64. This report demonstrates that Heterogeneous multicore architecture can provide significantly higher performance than a Homogeneous chip multiprocessor. It does so by matching the various jobs of a diverse workload to the various cores.

INTRODUCTION

The increase in need for machines with higher performance, computational power and increase in complexity in the design of uniprocessor has been the driving force for increase in interest in design of multicore architecture [1]. A multicore processor has multiple cores integrated on a single chip. A multicore architecture where every core is just an image of the other is called homogeneous multicore. Heterogeneous is a set of cores which may differ in area, performance, power dissipated etc. The various design issues in multicore architecture include resource sharing power consumption, performance, area of the cache, cache coherence etc. In order to harness the resources provided by a multicore architecture application must show a certain level of parallelism [3] [7].

The trend of increasing a processor's speed to get a boost in performance is a way of the past. Multicore processors are the new direction manufacturers are focusing on. Using multiple cores on a single chip is advantageous in raw processing power, but nothing comes for free.

With additional cores, power consumption and heat dissipation become a concern and must be simulated before layout to determine the best floorplan which distributes heat across the chip, while being careful not to form any hot spots. Distributed and shared caches on the chip must adhere to coherence protocols to make sure that when a core reads from memory it is reading the current piece of data and not a value that has been updated by a different core. With multicore processors come issues that were previously unforeseen. Like:

How will multiple cores communicate? Should all cores be homogenous, or are highly specialized cores more efficient? And most importantly, will programmers be able to write multithreaded code that can run across multiple cores?

PERFORMANCE COMPARISON

In a heterogeneous multicore architecture as it mentioned previously a core may differ from the other cores in many ways. First, a concentration on the situation where the cores differ in their performance. It will examine how the performance heterogeneity in the multicore will enhance the performance of the overall system. Building a performance heterogeneous core is desired because many simple cores together provide high parallel performance while complex cores help in providing high serial performance[7][5]. From Amdahl's law it can be concluded that the serial execution plays an important part in overall performance of the system [10].

$$\text{Speedup}_{\text{overall}} = \frac{\text{ExTime}_{\text{old}}}{\text{ExTime}_{\text{new}}} = \frac{1}{(1 - \text{Fraction}_{\text{enhanced}}) + \frac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}}}$$

Heterogeneous multicore processor reduces the effect of serial execution on system performance.

To efficiently utilize the performance potential of the multicore processor the application must be broken into parallel set of tasks. It is always desirable to have more number of parallel tasks than the number of cores.

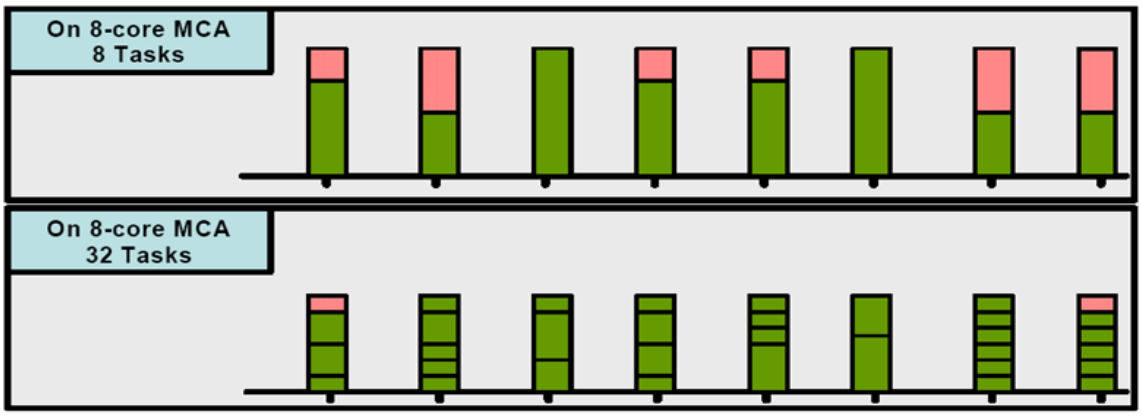

Figure 1: shows impact of parallelism in multicore processor [1][2].

Figure 1 above indicates how the resources are utilized when the program is broken into parallel tasks. However it is not always possible to break an application into parallel tasks. In such a situation there is wastage of the resources and overall performance is affected. As a result, comes the main advantage of Heterogeneous multicore processor over homogenous. In such a situation where several executions are required Heterogeneous multicore processor has the upper hand. The control mechanism in Heterogeneous multicore processor schedules the several tasks on the core with high performance and parallel tasks are executed on the simpler cores. By doing so the execution time of the several tasks reduces which help in overall increase of the performance [1][2].

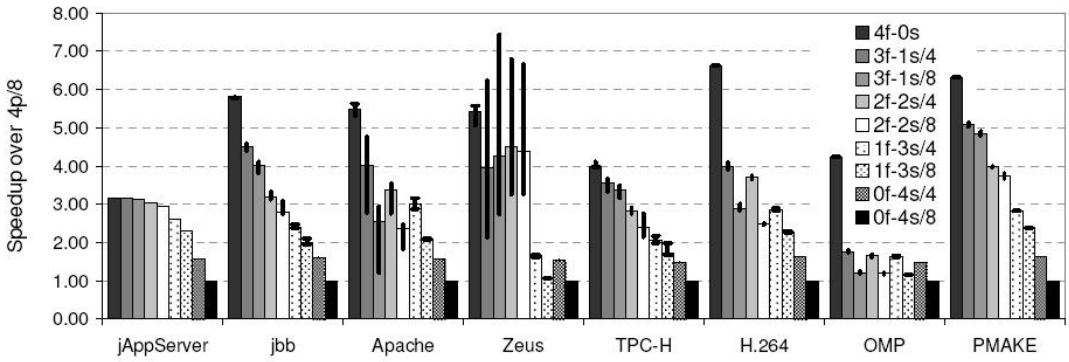

Figure2: shows superior performance of heterogeneous and homogeneous multicore processor [7].

The above figure indicates the superior performance of heterogeneous over homogeneous multicore architecture. Where 'f' indicates the number of faster cores and's' indicates the slower cores and s/scale indicates the number of times the core is slower than the faster core [7].

Throughput advantages

A heterogeneous multicore architecture can match each application to the core best suited to meet its performance demands, and it can provide improved area-efficient coverage of the entire spectrum of workload demands seen in a real machine, from low thread-level parallelism that provides low latency for few applications on powerful cores to high thread-level parallelism in which simple cores can host a large number of applications simultaneously. For those two reasons mentioned previously, given a fixed circuit area, using heterogeneous multicore architectures instead of homogeneous multicore can provide significant performance advantages for a multiprogrammed workload [3].

So, for example, in a homogeneous architecture with four large cores, it would be possible to replace one large core with five smaller cores, for a total of eight cores. In the best case, intelligently scheduling jobs on the smaller cores that

would have seen no significant benefit from the larger core would yield the performance of eight large cores in the space of four.

Overall, a representative heterogeneous processor using two core types achieves as much as a 63 percent performance improvement over an equivalent-area homogeneous processor. Over a range of moderate load levels—five to eight threads, for example—the heterogeneous multicore processor has an average gain of 29 percent [5].

For an open system with random job arrivals, the heterogeneous architecture has a much lower average response time and remains stable for arrival rates 43 percent higher than for a homogeneous architecture. Dynamic thread-to-core assignment policies have also been demonstrated that realize most of the potential performance gain. One simple assignment policy outperformed naive core assignment by 31 percent. Heterogeneity also can be beneficial in systems with multithreaded cores. Despite the additional scheduling complexity that simultaneous multithreading cores pose due to an explosion in the possible assignment permutations, effective assignment policies can be formulated that do derive significant benefit from heterogeneity [5].

POWER COMPARISON

Power consumption and heat dissipation have become key challenges in high-performance processor designs. Third, a concentration on the situation where the cores differ in their dissipated power. The Alpha chip is a 64-bit RISC microprocessor designed for providing high-performance migration path for VAX customers. Even though this processor was phased out by HP, it still is a perfect example for the comparison for heterogeneous multiprocessors in terms of power and performance [3].

The power and relative performance of Alpha cores scaled to 0.10μm is shown in Table 1. Their respective performance is normalized with respect to the performance of the Alpha EV4 chip. It could be seen that the EV4 chip consumes the least power of them all, as well as the lowest IPC (instructions per clock cycle). Its CPI translates to 1 clock cycles per instruction.

Core	Peak Power (Watts)	Average Power (Watts)	Perfomance (IPC)
EV4	4.97	3.73	1.0
EV5	9.83	6.88	1.3
EV6	17.8	10.68	1.87
EV8	92.88	46.44	2.14

Table 1: Performance comparison of Alpha CoresPerformance is normalized to EV4 performance [3].

As the processor becomes more powerful and complex, its peak power and average power increases dramatically. Its IPC however is somewhat comparable to the simpler cores. It may achieve higher single-thread performance, it comes with a cost – loss of area and efficiency. Some applications require high ILP, this can be exploited with a core that can issue multiple instructions per cycle. The same core, however, might be very inefficient with applications with low ILP. A simpler core may be matched with the applications requirements.

Using single-ISA heterogeneous multicore processor could significantly reduce processor power dissipation. Recently, the industry had introduced techniques for power reduction. Clock gating and voltage/frequency scaling are used. These are very useful and effective when applied in single-core level. As for multiple cores, these techniques however suffer from limitations. Data must be passed onto different parts of the processor, and may be passed on unused portions that have been gated off, which consumes a substantial amount of power. Gating only reduces dynamic power. Large unused portions of the chip still consume leakage power. This is the same also with voltage/frequency scaling techniques [4].

The ability to dynamically switch between different cores, and power down unused cores is the key in asymmetric chip multiprocessing. It was shown that a representative heterogeneous processor using two core types achieves as much as a 63 percent performance improvement over an equivalent-area homogenous processor [4]. Heterogeneous multiprocessors achieve better coverage of a spectrum of load levels.

As for Amdahl's law, during the execution of the serial phases, the large core allows the serial phase to be executed as quickly as possible. For the parallel phases, the chip's power budget is used more efficiently by running the parallel phase on smaller, area and power efficient cores. This maximizes the ratio of

performance to power dissipation, as well as the performance per area metric [10].

RECENT DEVELOPMENTS

- **Multicore Implementations**

Like any technology, multicore architectures from different manufacturers have many variations. Such as differences in communication and memory configuration another variance comes in the form of how many cores the microprocessor has. And in some multicore architecture different cores have different functions, hence they are heterogeneous. Differences in architectures are discussed below for Intel's Core 2 Duo, Advanced Micro Devices' Athlon 64 X2, Sony-Toshiba- IBM's CELL Processor, and finally Tilera's TILE64.

a) Intel and AMD Dual-Core Processors

Intel and AMD are the main manufacturers of microprocessors. Intel produces many different types of multicore processors: the Pentium D is used in desktops, Core 2 Duo is used in both laptop and desktop environments, and the Xeon processor is used in servers. AMD has the Althon lineup for desktops, Turion for laptops, and Opteron for servers/workstations. Although the Core 2 Duo and Athlon 64 X2 run on the same platforms their architectures differ greatly.

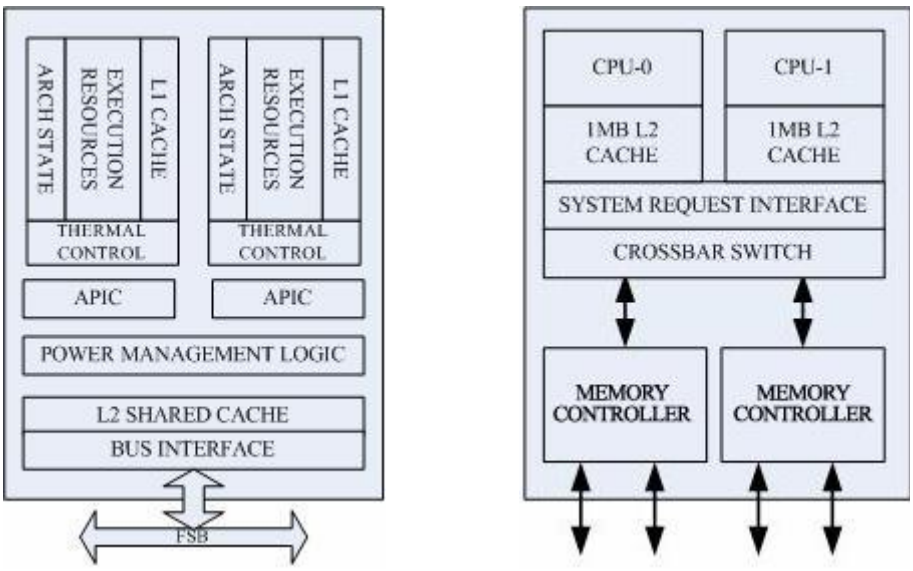

Figure 3 [11] (a) Intel Core 2 Duo, (b) AMD Athlon 64 X2

Figure 3 shows block diagrams for the Core 2 Duo and Athlon 64 X2, respectively. Both architectures are homogenous dual-core processors. The Core 2 Duo adheres to a shared memory model with private L1 caches and a shared

L2 cache. If a L1 cache miss occurs both the L2 cache and the second core's L1 cache are traversed in parallel before sending a request to main memory. In contrast, the Athlon follows a distributed memory model with discrete L2 caches. These L2 caches share a system request interface, eliminating the need for a bus. The system request interface also connects the cores with an on-chip memory controller and an interconnect called HyperTransport. HyperTransport effectively reduces the number of buses required in a system, reducing bottlenecks and increasing bandwidth. The Core 2 Duo instead uses a bus interface. The Core 2 Duo also has explicit thermal and power control units on-chip. There is no definitive performance advantage of a bus vs. an interconnect, and the Core 2 Duo and Athlon 64 X2 achieve similar performance measures, each using a different communication protocol [11].

b) CELL Processor

A Sony-Toshiba-IBM partnership (STI) built the CELL processor for use in Sony's PlayStation3; therefore, CELL is highly customized for gaming/graphics rendering which means superior processing power for gaming applications. The CELL is a heterogeneous multicore processor consisting of nine cores, one Power Processing Element (PPE) and eight Synergistic Processing Elements (SPEs), as can be seen in figure 4.

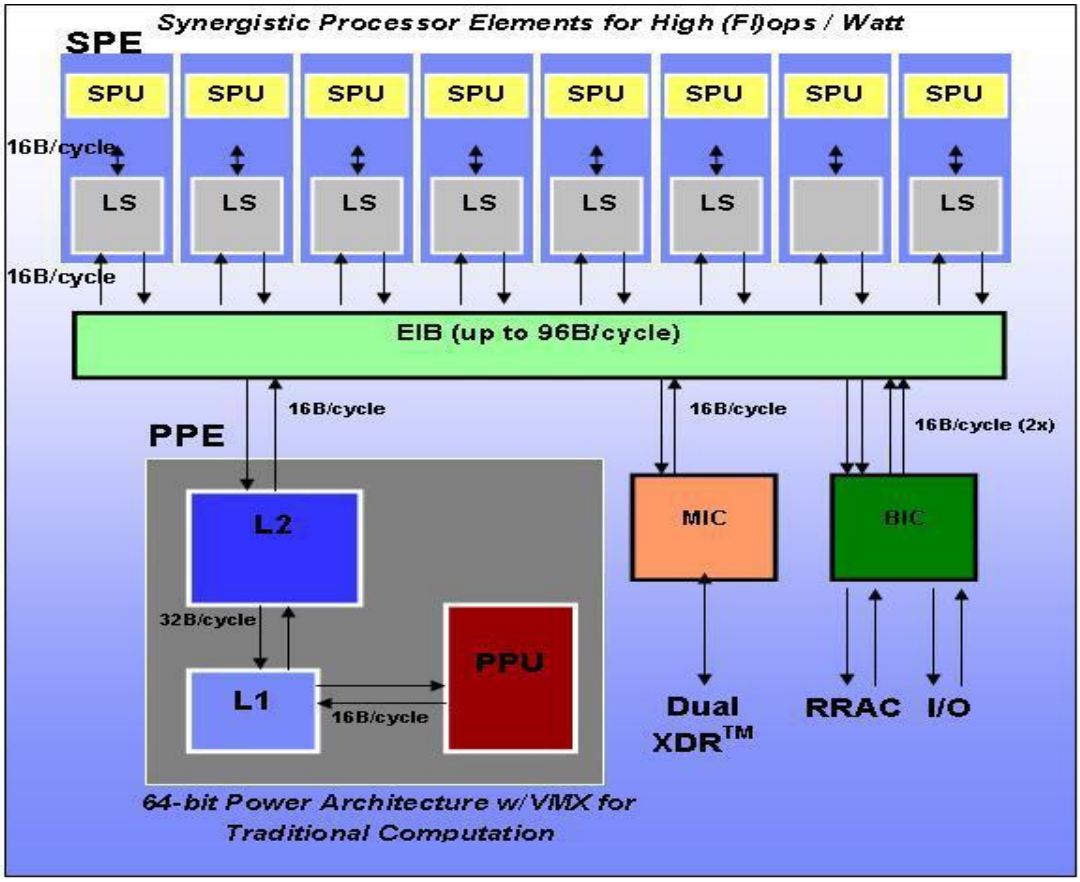

Figure 4: Cell processor [12]

With CELL's real-time broadband architecture, 128 concurrent transactions to memory per processor are possible. The PPE is an extension of the 64-bit PowerPC architecture and manages the operating system and control functions. Each SPE has simplified instruction sets which use 128-bit SIMD instructions and have 256KB of local storage. Direct Memory Access is used to transfer data between local storage and main memory which allows for the high number of concurrent memory transactions. The PPE and SPEs are connected via the Element Interconnect Bus providing internal communication. Other interesting features of the CELL are the Power Management Unit and Thermal Management Unit. Power and temperature are fundamental concerns in microprocessor design. The PMU allows for power reduction in the form of slowing, pausing, or completely stopping a unit. The TMU consists of one linear sensor and ten digital thermal sensors used to monitor temperature throughout the chip and provide an early warning if temperatures are rising in a certain area of the chip. The ability to measure and account for power and temperature changes has a great advantage in that the processor should never overheat or draw too much power[12].

c) Tilera TILE64

Tilera has developed a multicore chip with 64 homogeneous cores set up in a grid, shown in Figure 5.

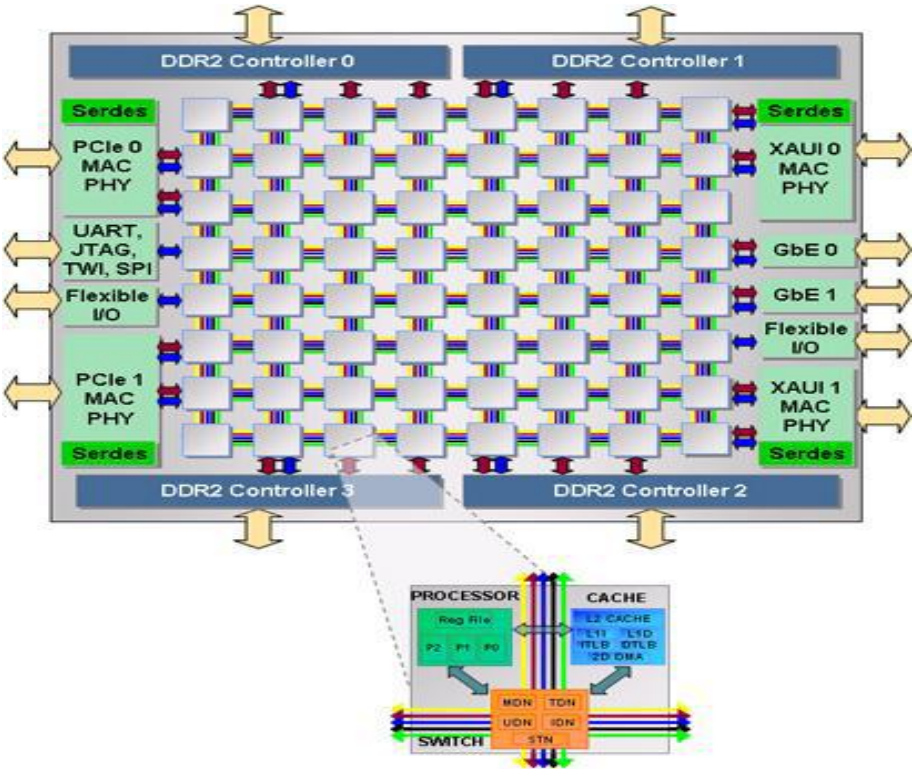

Figure 5: Tilera TILE64 [13]

An application that is written to take advantage of these additional cores will run far faster than if it were run on a single core. Each processor has its own L1 and L2 cache for a total of 5MB on-chip and a switch that connects the core into the mesh network rather than a bus or interconnect. The TILE64 also includes on-chip memory and I/O controllers. Like the CELL processor, unused tiles (cores) can be put into a sleep mode to further decrease power consumption. The TILE64 uses a 3-way VLIW (very long instruction word) pipeline to deliver 12 times the instructions as a single-issue, single-core processor. When VLIW is combined with the MIMD (multiple instructions, multiple data) processors, multiple operating systems can be run simultaneously and advanced multimedia applications such as video conferencing and video-on-demand can be run efficiently. [13]

- **Implementation summary**

Many architects have studied whether the cores in a multicore environment should be homogeneous or heterogeneous, and there is no final answer…yet. Homogenous cores are all exactly the same: equivalent frequencies, cache sizes, functions, etc. However, each core in a heterogeneous system may have a different function, frequency, memory model, etc. There is an apparent trade-off between processor complexity and customization. All of the designs discussed above have used homogeneous cores except for the CELL processor, which has one Power Processing Element and eight Synergistic Processing Elements. Homogeneous cores are easier to produce since the same instruction set is used across all cores and each core contains the same hardware. But are they the most efficient use of multicore technology? Each core in a heterogeneous environment could have a specific function and run its own specialized instruction set. Depending on the CELL example, explained previously, a heterogeneous model could have a large centralized core built for generic processing and running an OS, a core for graphics, a communications core, an enhanced mathematics core, an audio core, and so on.

FUTURE DIRECTIONS

There is a bright future in multiprocessor performance. But optimizations are still needed in order to get most out of the current advances in technology. In order to increase the performance to its maximum, it was shown that the serial and parallel phase of a software programs must be equal.

The future lies in the careful and intelligent design of Heterogeneous Chip Multiprocessors. The advantages of Heterogeneous Chip Multiprocessors clearly outweigh that of Homogeneous Chip Multiprocessors. It will just take a matter of time before the microprocessor architecturer move to a new direction in

microprocessor design, just like the jump from uniprocessor to multiprocessors. Maybe soon, the term “transistor count” previously used to measure the microprocessors will become “processor count”.

CONCLUSIONS

Multicore processors are designed as a result to reasonable power consumption, heat dissipation, and cache coherence protocols. However, many issues remain unsolved. In order to use a multicore processor at full capacity the applications run on the system must be parallel. The memory systems and interconnection net-works also need improvement. The advantages of Heterogeneous Chip Multiprocessors clearly outweigh that of Homogeneous Chip Multiprocessors. The greatest difficulty remains in teaching parallel programming techniques (since most programmers are so versed in sequential programming) and in redesigning current applications to run optimally on a multicore system. Multicore processors are an important innovation in the microprocessor timeline. With skilled programmers capable of writing parallelized applications multicore efficiency could be increased dramatically.

REFERNECES

[1] Sanjeev kumar , Christopher J Hughes, Anthony Nguyen."CARBON: Architecture support for fine-grained parallelism on chip multiprocessors". In the Proceedings of IEEE/ACM International Symposium on Computer Architecture (ISCA), San Diego, California, June 2007.

[2] Sanjeev kumar , Christopher J Hughes, Anthony Nguyen. "Architectural support for fine grained parallelism on Multicore architecture". Intel cooperation Intel technology journal, Volume II ,Issue 3-2007.

[3] R. Kumar, D. Tullsen, N. Jouppi, R. Ranganathan. “Heterogeneous Chip Multiprocessors” Computer. Volume 38, Issue 11, pages 32-38, Nov. 2005.

[4] R. Kumar et al., “Single-ISA Heterogeneous Multicore Architectures: The Potential for Processor Power Reduction,” Proc. Intel Symp.Microarchitecture, IEEE CS Press, 2003, pp. 81-92.

[5] R. Kumar et al., “Single-ISA Heterogeneous Multicore Architectures for Multithreaded Workload Performance,” Proc. Intel Symp.Computer Architecture, IEEE CS Press, 2004, pp. 64-75.

[6] R. Kumar, V. Zyuban, and D. M. Tullsen. "Interconnections in multi-core architectures: Understanding mechanisms, overheads and scaling,". In Proceedings of International Symposium on Computer Architecture, 2005.

[7] S. Balakrishnan et al., "The Impact of Performance Asymmetry in Emerging Multicore Architectures," Proc. Int'l Symp. Computer Architecture, IEEE CS Press, 2005, pp. 506-517.

[8] S. Ghiasi, T. Keller, and F. Rawson. "Scheduling for heterogeneous processors in server systems". In Proceedings of Computing Frontiers, 2005.

[9] T. Y. Morad, U. C. Weiser, A. Kolodny, M. Valero, and E. Ayguade. "Performance, power efficiency and scalability of asymmetric cluster chip multiprocessors". In Computer Architecture Letters, Vol 4, July 2005.

[10] John L. Hennessy , David A. Patterson. "Computer Architecture: A Quantitative Approach". 4th Edition.

[11] L. Peng et al, "Memory Performance and Scalability of Intel's and AMD's Dual-Core Processors: A Case Study", IEEE, 2007

[12] D. Pham, S. Asano, M. Bolliger, M. N. Day, H. P. Hofstee, C. Johns, J. Kahle, A.Kameyama, J. Keaty, Y. Masubuchi, M. Riley, D. Shippy, D. Stasiak, M. Suzuoki, M. Wang, J. Warnock, S. Weitzel, D. Wendel, T. Yamazaki, K. Yazawa. " The Design and Implementation of a First-Generation CELL Processor". In 2005 IEEE International Solid-State Circuits Conference.

[13] Tilera, "Tile 64 Product Brief", Tilera, 2008, http://www.tilera.com/pdf/ProductBrief_Tile64_Web_v3.pdf