SYSC5603 Paper Analysis:

Multiprocessors for DSP

Author: Trevor Burton

Submitted to: Professor Miodrag Bolic

Carleton Student ID: 100278854

Date: Monday, February 6, 2006

1. Introduction

The performance of single-core microprocessors, or uniprocessors, has been driven over the years by the decreasing size and increasing speed of silicon transistors. This has resulted in faster and more powerful processors in accordance with Moore's Law, which states that the number of transistors that can be fabricated on a give chip size doubles approximately every two years. However, as the number of transistors on a chip has increased, along with speed at which they switch, so has the amount of power they consume and heat they generate. These power consumption and heat dissipation concerns, along with other factors such as dwindling performance gain from exploiting instruction-level parallelism (ILP) [1] in these superscalar processors, have resulted in a slowdown in processor performance increases [2].

As a result, manufactures have adopted a multi-core architecture where instead of focusing on designing bigger, more complex, and power hungry uniprocessors, several smaller energy efficient processors are integrated onto a single chip. These new multi-core processors operate at a lower temperature, are more energy efficient, and achieve greater throughput than their uniprocessor counterparts. The greater amount of throughput that can be attained by these new multi-core processors comes as a result of exploiting parallelism within the system. Many major manufactures of microprocessors for the desktop, laptop, and server market now offer multi-core processors [3-6]. Moving forward it is now believed that no high performance commercial microprocessor will be constructed using a single core alone [7].

The trend towards multi-core computing that has come into acceptance in the home computing environment has been around for many years in embedded systems in the form of multiprocessor systems-on-chips (MPSoCs). MPSoCs are generally employed in specific applications that require real-time concurrent computations, where area and energy efficiency are also required [8]. Examples of embedded applications where MPSoCs are used include cellular phones, telecommunication equipment, digital televisions, and video game systems to name a few [8]. General purpose processors (GPPs) tend not to be used in embedded applications because of their high cost, poor energy efficiency, and inability to meet real-time performance requirements. Manufactures of programmable digital signal processors (PDSPs) including Texas Instruments, Analog Devices, and Freescale Semiconductor all have commercially available multi-core PDSPs [9-11]. Also, field programmable gate array manufactures (FPGA) such as Xilinx and Altera now offer FPGAs with built in microprocessor cores that can be used to create MPSoCs [12-13].

With the recent development of multi-core processors by Intel, AMD, IBM, and Sun Microsystems, along with the already established MPSoCs in the embedded computing community, the multi-core approach appears to be future for microprocessor design. The remainder of this paper is organized as follows. Section two discusses the chip multiprocessor (CMP) or multi-core architecture, its advantages and disadvantages, current CMP solutions, software issues, along with giving a detailed example of a current multiprocessor design. Section three outlines and compares some of the development

tools that exist for designing and testing CMPs. Finally, conclusions are presented in section four.

2. Chip Multiprocessors

A CMP integrates more than one processing core or processing element (PE) onto a single chip. These PEs can either be the same (homogeneous CMP) or different (heterogeneous CMP) cores. MPSoCs are an example of heterogeneous CMPs as the PEs that comprise them perform different specialized tasks. Homogeneous CMPs contain the same PEs and are the type of CMP architecture that is employed by vendors in the dual-core desktop microprocessors that are available today. Since CMPs contain multiple PEs then can perform more work in parallel than their single processor counterparts which results in increased system throughput. However, having multiple PEs on a single chip makes software development more difficult compared to standard uniprocessor programming. CMP architectures, their advantages and disadvantages, software implications, and current hardware solutions including a detailed example are discussed in the sections below.

2.1. The CMP Architecture

As mentioned above a CMP incorporates smaller energy efficient PEs onto a single chip instead of one large complex power hungry core. An example of a homogeneous CMP is the Stanford Hydra CMP which consists of four MIPS based reduced instruction set computer (RISC) processors on a single chip [1]. A block diagram of the Hydra CMP is shown in Figure 1 below.

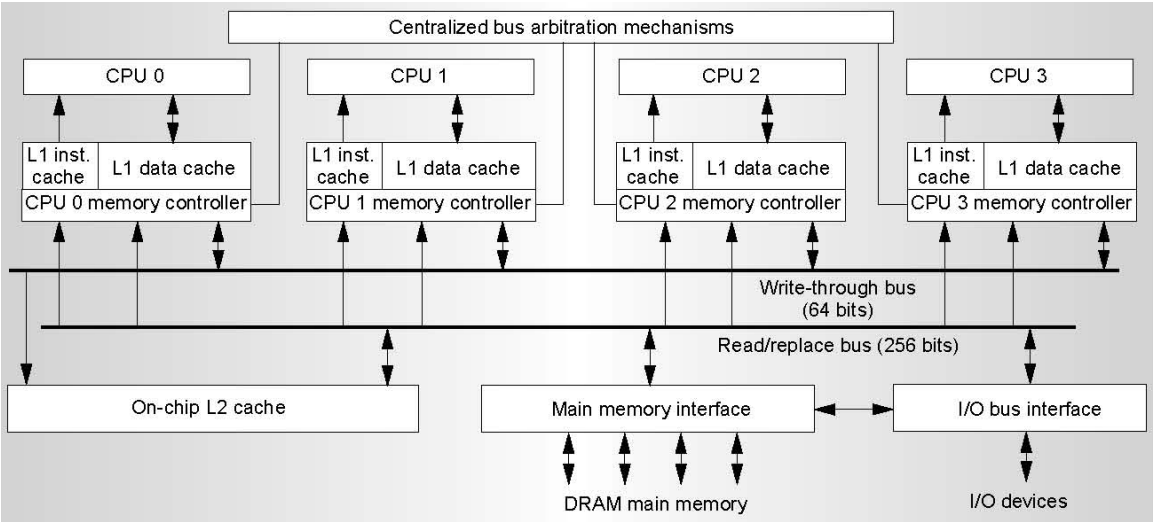

Figure 1 – Block diagram of the Hydra CMP [1]

Each MIPS core contains its own program and data cache memory and share an on chip level two cache memory. The processor cores and level two cache are connected together via separate read and write buses which also provide connection to off chip memory and I/O devices.

An example of a heterogeneous CMP is the Emotion Engine chip used in the Sony PlayStation 2 [8] and is shown in Figure 2 below.

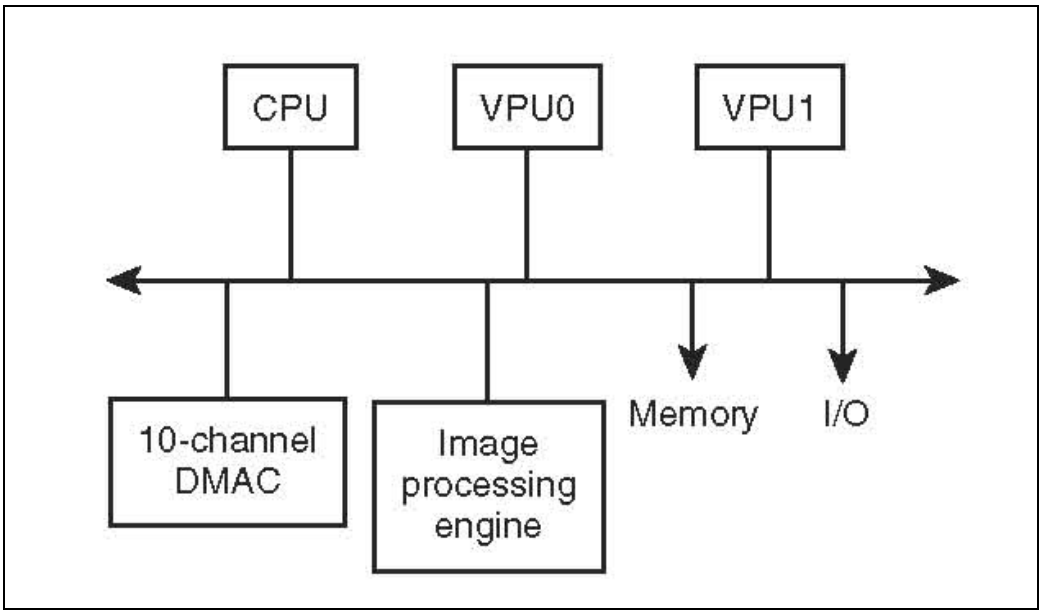

Figure 2 – Block Diagram of the Sony PlayStation 2 Emotion Engine [8]

The Emotion Engine chip contains a general purpose MIPS RISC processor along with two vector processing units that have single instruction multiple data – very long instruction word (SIMD-VLIW) architectures [14]. VPU0 generally works as a coprocessor to the MIPS CPU for performing complex calculations while VPU1 works independently performing simple geometry operations [14]. The Emotion Engine chip is tailored for video and image processing and can deliver over 5 billion floating point operations per second (FLOPS) [8].

2.2. CMP Advantages

CMPs have several advantages over single processor systems as well as over symmetric multiprocessor (SMP) systems comprised of multiple individual chip processors. These advantages include energy and silicon area efficiency, which are especially important in embedded computing applications. Also, CMPs can achieve greater throughput than a single processor when it comes to performing work in parallel.

By incorporating smaller less complex cores onto a single chip significant energy savings can be realized with CMPs. This can be achieved by dynamically switching between cores and powering down cores that are not currently being used as discussed by Kumar *et al.* in [15]. The authors in [15] also discussed that using heterogeneous cores can improve the energy consumed per instruction by four to six times compared to a larger single core system, where power savings techniques such as voltage scaling could only result in two to four times improvement in energy per instruction. Also, by consuming less energy CMPs dissipate less heat than a single large complex processor which can

help mitigate the costs of thermal packaging, fans, and electricity while reducing the chances of failure due to overheating.

The throughput advantage of CMPs occurs as a result of exploiting parallelism within the system. In embedded DSP applications task level parallelism is exploited by MPSoCs. Each PE in the MPSoC generally has a specific task that is executed concurrently with the others. For example, in [8] the authors describe an embedded MPEG-2 encoding system where motion estimation, discrete cosine transform, and Huffman coding are preformed by individual PEs concurrently. Trying to use a uniprocessor or even a multi-chip SMP system to perform MPEG-2 encoding is impractical for an embedded application as these systems would not be able to meet the real-time, energy, or I/O requirements as discussed in [8].

In more general purpose computing applications the finest level of parallelism is ILP. Single superscalar processors have been exploiting ILP, which involves executing multiple instructions concurrently, for a number of years but only for a single thread of execution at a time. The performance improvement that can be achieved by exploiting ILP alone is limited as instructions within any particular sequence tend to be interdependent [1]. The next level of parallelism is thread-level parallelism (TLP). Simultaneous multithreading (SMT) processors, such as Intel's Pentium 4 chip, allow multiple threads of execution to run instructions concurrently on available processor execution units within the same clock cycle, resulting in greater throughput. In addition to the ILP inherent to the VLIW architecture generally used in PDSPs, SMT can also be employed in DSP applications to further exploit parallelism, thereby increasing throughput. Various approaches for incorporating SMT into PDSPs are given in [16-20] and include target applications such as multimedia and image processing. The coarsest level of parallelism is process-level parallelism (PLP) which involves independent application processes running concurrently on system resources [21]. CMPs can take advantage of all three types of parallelism mentioned above in order to achieve superior throughput, or work performed per unit of time. Hammond *et al.* in [21] performed simulations to show that under high TLP and/or PLP conditions their CMP, consisting of eight small superscalar processors, achieved almost eight times performance improvement over their single superscalar processor and achieved considerable performance gains over their SMT processor. However, under single-thread conditions the authors in [21] showed that their CMP is slower than their superscalar and SMT processors as only one processor in the CMP can be used. But, the authors in [21] also point out that under conditions when code cannot be multithreaded the remaining processors can be used to increase throughput by running other independent processes.

Chaudhry *et al.* in [22] showed that incorporating several SMT cores onto a single chip results in significant improvement in performance. For example they reported a performance improvement of 3.7 times when four SMT cores, each capable of running four threads simultaneously, were integrated onto a single chip compared to a single SMT core. The authors in [22] also reported that good single-thread performance was achieved using their CMP.

In comparing homogenous CMPs to heterogeneous CMPs Kumar *et al.* in [15] showed that a heterogeneous CMP comprised of two different core types can reach as much as a 63 percent performance improvement compared to an equal-area homogenous CMP. The authors also demonstrate that only heterogeneous CMPs with multithreaded cores provide high performance under all levels of thread parallelism. The authors in [15] attribute the increased performance of heterogeneous CMPs over homogenous CMPs to heterogeneous CMPs being able to better match the computing requirements of an application to a specific core and to better balancing of the varying TLP within the workload.

2.3. Software Issues

In the world of embedded computing, where most real-time DSP applications are addressed, programming a MPSoC is considerably more difficult than conventional parallel programming as pointed out by the authors in [8]. This is due to the generally large number of drastically different PEs in a MPSoC. Thus, to overcome this it is necessary to obtain efficient models and methods of MPSoC programming [8]. Also, since MPSoCs tend to be used in applications requiring real-time high performance along with low power consumption, their software needs to be optimized in terms of code size and execution time in order to meet these constraints [8].

For more general purpose computing, applications need to be developed that are multithreaded in order to take full advantage of commercially available CMPs. In [15] the authors state that the computing needs of an application need to be predicted beforehand and scheduled to a core that closely matches those needs in order to take advantage of heterogeneous CMPs. Also, since heterogeneous CMPs employ cores with different processing performance programming and/or compiling applications will need to be aware of this asymmetry and try to take this discrepancy into account. The authors in [21] also state that in order to maximize CMP performance programmers must take TLP into consideration. They also point out that even under heavy communication between threads in a CMP, TLP can still be exploited due to the low on-chip communication latencies. The communication latencies between threads in SMP systems have been a bottleneck that can prevent parallel programs from attaining speedups on these systems [21].

2.4. Current CMP Solutions

Many different types of CMP solutions exist today and choosing one depends on the intended application. For example in the desktop computing industry all major manufactures of processors in this area now offer multi-core solutions. Intel offers its Pentium D dual-core processors for desktop use, AMD offers its Athlon 64 X2 dual-core processor, and IBM offers a dual-core processor based on their PowerPC architecture. Sun Microsystems also offers multi-core processors based on their UltraSPARC architecture. These multi-core desktop processors tend to be homogeneous CMPs.

Manufactures of PDSPs also offer multi-core versions of their products. Texas Instruments offers multi-core versions of its TMS320C54x line of processors. Analog Devices offers multi-core versions of its SHARC line of processors. Also, Freescale Semiconductor offers multi-core versions of its StarCore line of PDSPs. These processors tend to be homogenous CMPs; however heterogeneous CMPs in the form of MPSoCs are available from many major semiconductor manufactures. For example Texas Instruments offers many varieties of its open multimedia applications platform (OMAP) processors that combine many different PEs including ARM RISC type processors, DSP processors, graphic accelerators, along with video and audio PEs to name a few. These MPSoCs from Texas Instruments are used in embedded applications ranging from cellular phones and personal digital assistants (PDAs) to modems.

FPGA manufactures including Xilinx and Altera offer programmable devices with immersed microprocessor cores that can be used along with embedded and configurable logic to create powerful heterogeneous CMPs. Xilinx offers its Virtex-II FPGAs with built in PowerPC cores along with soft intellectual property (IP) cores. Similarly, Altera offers its Stratix II FPGAs with its NIOS II processor embedded.

A good example of a new commercial heterogeneous CMP is the Cell Broadband Engine (Cell BE) processor developed jointly by Sony, Toshiba, and IBM. The Cell BE processor was designed with performance per area and performance per Watt in mind and achieves this by incorporating many powerful yet simple PEs onto the chip [23]. The Cell BE processor consist of a 64-bit general purpose POWER processing element (PPE) based on IBM's POWER processing architecture, eight synergistic processing elements (SPEs), an on chip high speed memory controller, and an integrated I/O controller [24]. A block diagram of the Cell BE processor is shown in Figure 3 below.

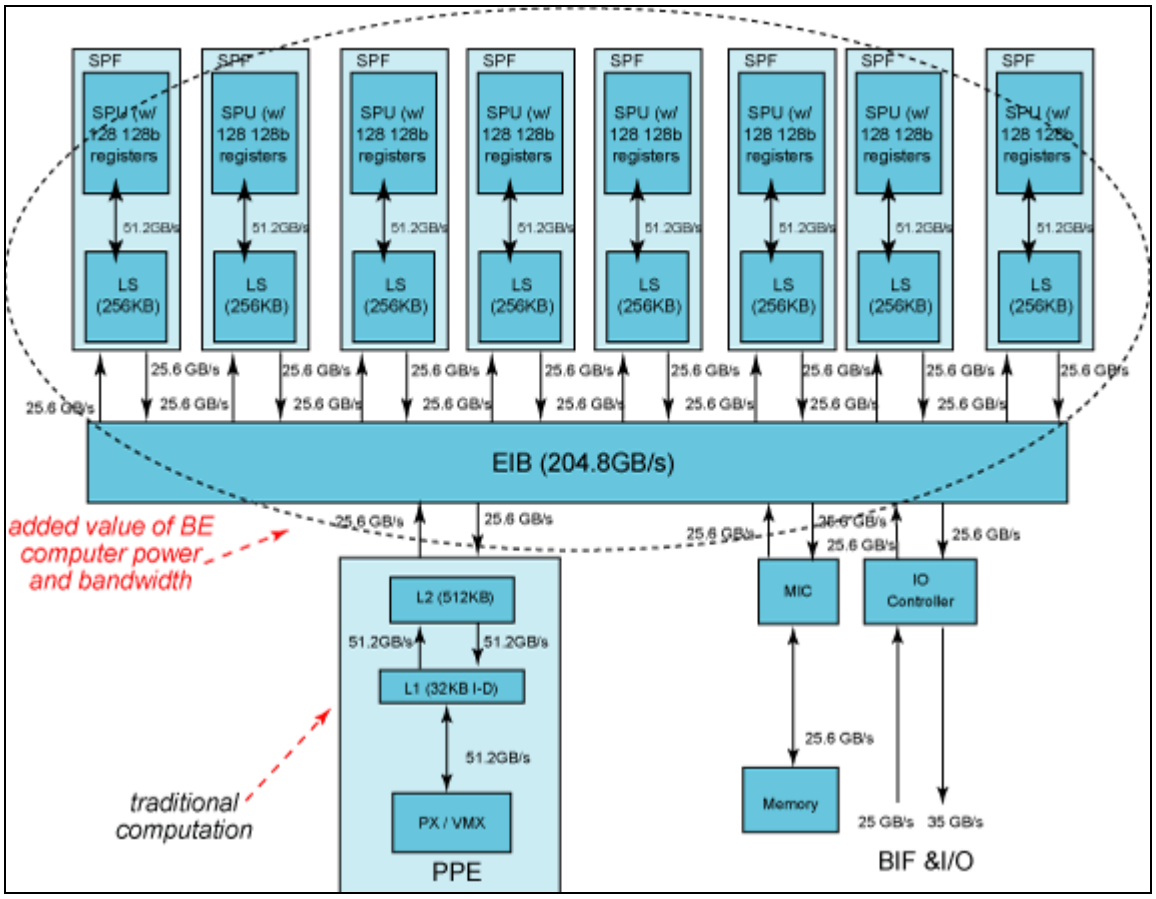

Figure 3 – Block diagram of the Cell BE processor [23].

The PPE is a SMT processor responsible for running the operating system and coordinating the SPEs [23]. The SPEs are each comprised of a synergistic processing unit (SPU) and a memory flow controller. The SPU is a SIMD computational engine that can perform either floating point or integer operations by operating directly on instructions fetched from its 256KB local store [23]. The memory flow controller can operate independently of the SPU allowing for DMA transfers while the SPU is busy and is also responsible for synchronizing operations with other SPUs and the PPE. The element interconnect bus (EIB) is the main system bus and allows for communication between the PPE, SPEs, off chip memory, and external I/O [23]. The EIB can support a peak bandwidth of 204.8GB/s for intra-chip communication. The integrated memory interface controller supports high memory bandwidth via external RAMBUS XDR memory banks.

The Cell BE processor has been shown to outperform leading GPPs by approximately an order of magnitude under a variety of workloads [23], while achieving power and area efficiency. The Cell BE processor can perform 192 GFLOPS and is intended for multimedia rich applications such as video game consoles, televisions, and other image and video processing systems.

3. Tools for Designing Chip Multiprocessors

Of the multi-core approaches that exist, including those from general purpose processor manufactures such as Intel, AMD, and IBM, along with solutions from PDSP and FPGA manufactures, none offer support for a multi-core design from scratch. Instead customers decide on a multi-core approach from a specific vendor and then use the tools available from that vendor to program, test, and verify their multi-core solution.

One solution for the design and simulation of multi-processor systems from the ground up is offered by Cmpware Inc. They offer a configurable multi-processor development kit (CMP-DK) which provides fast simulation models for microprocessors, integrated into a multi-processor simulation engine [7]. Their development environment lets processors and their interconnection network be quickly and simply defined, where existing processor cores along with their associated tools, libraries, and software can be used [25]. Since new processors cores and tool do not have to be designed a multi-core model can be quickly and efficiently designed using the CMP-DK. Also, users are able to specify their own custom processor architectures and interconnect networks or modify an existing architecture as needed.

Another development environment that can be used to design and simulate multi-processors systems is the ConvergenSC family of tools based on SystemC by CoWare. Users can design their systems using a top-down design flow where the designer creates a functional specification in SystemC then builds or selects a processing platform containing one or more processor models as needed [26]. Alternatively, designers can follow a platform-based design flow where there systems can be created and validated at the transaction level using SystemC. Also, hardware descriptive language (HDL) blocks can be imported into the system at the transaction level allowing for a very flexible design.

4. Conclusion

As the size of transistors has shrunk the number that can be integrated onto a chip has increased according to Moore's Law. However, with the ever increasing number of transistors that semiconductor manufactures put on chips the power they consume and heat they dissipate has resulted in diminishing increases in performance for the typical microprocessor design. As a result, manufactures have begun to steer away from the single large complex microprocessor design philosophy and have turned to more energy and area efficient designs involving multiple smaller less complex microprocessor cores on a single chip. This is a trend that has been going on in the PDSP and application specific integrated circuit (ASIC) environments for many years as vendors in these areas offer multi-core products in the form of MPSoCs. Also, FPGA manufactures now offer products with multiple microprocessor cores already embedded to allow for powerful MPSoCs to be easily created. Also, all major manufactures of microprocessors for the desktop, laptop, and server markets offer dual-core solutions.

These CMP approaches have shown to be much more powerful in terms of throughput than their single core counterparts, especially under multithreaded and/or multi-process

workloads. In [21] the authors simulations showed an almost eight times performance increase for their CMP over their single SMT processor under multithreaded conditions. In [22] the authors showed an almost four times performance improvement with their four core SMT processor compared to a single SMT processor. Also, heterogeneous CMPs have been shown to perform better than equivalent homogenous CMPs by up to 63 percent in [15]. However, in order to take advantage of the power of CMPs software must be multithreaded and scheduled efficiently so as to use the parallel resources of the CMP maximally.

To Implementing a CMP system a designer would generally use existing hardware from a specific manufacture along with their specific tools for programming and verification. However, designers can now create their own multiprocessor solutions using development environments from several companies including Cmpware and CoWare. These companies provide tools that allow multiprocessor systems to be designed and simulated quickly using existing microprocessor cores along with other IP cores and custom components.

References

[1] L. Hammond, B. Hubbert, M. Siu, M. Prabhu, M. Chen and K. Olukotun, "The Stanford Hydra CMP", *IEEE Micro*, vol. 20, no. 2, pp. 71-84, Mar./Apr. 2000.

[2] D. Geer, **"**Chip makers turn to multicore processors," *IEEE Computer***,** vol. 38, no. 5, pp. 11-13, 2005.

[3] Intel Dual-Core Processors. [Online]. Available: http://www.intel.com/technology/computing/dual-core/index.htm

[4] AMD Multi-Core Products. [Online]. Available: http://multicore.amd.com/en/Products/

[5] N. Rohrer. (2005, Dec.). Introducing the IBM PowerPC 970MP. Presented at the MPR Fall Processor Forum. [Online]. Available: http://www-128.ibm.com/developerworks/library/pa-fpf970mp/#23N10101

[6] UltraSPARC Processors. [Online]. Available: http://www.sun.com/processors/index.html

[7] S. A. Guccione. (2005, June). Microprocessors: The New LUT. Presented at the International Conference on Engineering of Reconfigurable Systems and Algorithms. [Online]. Available: http://www.cmpware.com/Papers/ERSA2005/Cmpware_ERSA_2005.pdf

[8] A. A. Jerraya, and W. Wolf," The What, Why, and How of MPSoC's," in *Multiprocessor Systems-on-Chips*, Morgan Kaufman, 2004, pp. 1-18.

[9] TMS320 DSP Platforms. [Online]. Available: http://focus.ti.com/paramsearch/docs/parametricsearch.tsp?family=dsp&sectionId=2&tabId=25&familyId=44

[10] SHARC Processor. [Online]. Available: http://www.analog.com/processors/processors/sharc/

[11] Digital Signal Processors and Controllers. [Online]. Available: http://www.freescale.com/webapp/sps/site/homepage.jsp?nodeId=012795

[12] FPGAs for DSP Applications. [Online]. Available: http://www.xilinx.com/products/design_resources/dsp_central/info/fpga_overview.htm#23v2pro

[13] Stratix II Device Family Features. [Online]. Available: http://www.altera.com/products/devices/stratix2/features/st2-features.html

[14] M. Oka, and M. Suzuoki, "Designing and programming the emotion engine," *IEEE Micro*, vol. 20, no. 6, pp. 20-28, Nov./Dec. 1999.

[15] R. Kumar, D. M. Tullsen, N.P. Jouppi, and P. Ranganathan, **"**Heterogeneous Chip Multiprocessors," *IEEE Computer***,** vol. 38, no. 11, pp. 32-38, 2005.

[16] M. Berekovic, S. Moch, and P. Pirsch, "A scalable, clustered SMT processor for digital signal processing," *ACM SIGARCH Computer Architecture News*, vol. 32. no. 3, pp. 62-69, June 2004.

[17] E. Ozer, and T.M. Conte, "High-performance and low-cost dual-thread VLIW processor using Weld architecture paradigm," *IEEE Trans. Parallel Distrib. Syst.*, vol. 16, no. 12, pp 1132-1142, Dec. 2005.

[18] S. Kaxiras, G. Narlikar, A.D. Berenbaum, and Z. Hu, "Comparing power consumption of an SMT and a CMP DSP for mobile phone workloads," in *Proc.*

Int. Conf. on Compiler, Architecture and Synthesis for Embedded Systems, pp. 211-220, 2001.

[19] B. Iyer, S. Srinivasan, and B. Jacob, "Extended Split-Issue: Enabling Flexibility in the Hardware Implementation of NUAL VLIW DSPs," in *Proc. 31st Int. Symp. on Comp. Arch.*, pp. 364-375, 2004.

[20] J.P. Wittenburg, P. Pirsch, and G. Meyer, "A multithreaded architecture approach to parallel DSPs for high performance image processing applications," in *Proc. IEEE SiPS*, pp. 241-250, Oct. 1999.

[21] L. Hammond, B. Nayfeh, and K. Olukotun, "A single-chip multiprocessor," *IEEE Computer*, vol. 30, pp. 79-85, Sept. 1997.

[22] S. Chaudhry, P. Caprioli, S. Yip, and M. Tremblay, "High-Performance Throughput Computing", *IEEE Micro*, vol. 25, no. 3, pp. 32-45, May/June 2005.

[23] Cell Broadband Engine Architecture and its first implementation. [Online]. Available: http://www-128.ibm.com/developerworks/power/library/pa-cellperf/#23ref1

[24] H. P. Hofstee, "Power efficient processor architecture and the cell processor," in *Proc. 11th Int. Symp. on High-Performance Comp. Arch.*, pp. 258-262, Feb. 2005.

[25] S. A. Guccione. (2005, April). Programming Configurable Multiprocessors. Presented at the 12th Reconfigurable Architectures Workshop. [Online]. Available: http://www.cmpware.com/Papers/RAW2005/Cmpware_RAW_2005.pdf

[26] ConvergenSC technical overview. [Online]. Available: http://www.coware.com/products/convergensc_techoverview.php