Homayoun

Reference 15

Introduction to the Cell multiprocessor

J. A. Kahle
M. N. Day
H. P. Hofstee
C. R. Johns
T. R. Maeurer
D. Shippy

This paper provides an introductory overview of the Cell multiprocessor. Cell represents a revolutionary extension of conventional microprocessor architecture and organization. The paper discusses the history of the project, the program objectives and challenges, the design concept, the architecture and programming models, and the implementation.

Introduction: History of the project

Initial discussion on the collaborative effort to develop Cell began with support from CEOs from the Sony and IBM companies: Sony as a content provider and IBM as a leading-edge technology and server company. Collaboration was initiated among SCEI (Sony Computer Entertainment Incorporated), IBM, for microprocessor development, and Toshiba, as a development and high-volume manufacturing technology partner. This led to high-level architectural discussions among the three companies during the summer of 2000. During a critical meeting in Tokyo, it was determined that traditional architectural organizations would not deliver the computational power that SCEI sought for their future interactive needs. SCEI brought to the discussions a vision to achieve 1,000 times the performance of PlayStation2** [1, 2]. The Cell objectives were to achieve 100 times the PlayStation2 performance and lead the way for the future. At this stage of the interaction, the IBM Research Division became involved for the purpose of exploring new organizational approaches to the design. IBM process technology was also involved, contributing state-of-the-art 90-nm process with silicon-on-insulator (SOI), low-*k* dielectrics, and copper interconnects [3]. The new organization would make possible a digital entertainment center that would bring together aspects from broadband interconnect, entertainment systems, and supercomputer structures. During this interaction, a wide variety of multi-core proposals were discussed, ranging from conventional chip multiprocessors (CMPs) to dataflow-oriented multiprocessors.

By the end of 2000 an architectural concept had been agreed on that combined the 64-bit Power Architecture* [4] with memory flow control and "synergistic" processors in order to provide the required computational density and power efficiency. After several months of architectural discussion and contract negotiations, the STI (SCEI–Toshiba–IBM) Design Center was formally opened in Austin, Texas, on March 9, 2001. The STI Design Center represented a joint investment in design of about $400,000,000. Separate joint collaborations were also set in place for process technology development.

A number of key elements were employed to drive the success of the Cell multiprocessor design. First, a holistic design approach was used, encompassing processor architecture, hardware implementation, system structures, and software programming models. Second, the design center staffed key leadership positions from various IBM sites. Third, the design incorporated many flexible elements ranging from reprogrammable synergistic processors to reconfigurable I/O interfaces in order to support many systems configurations with one high-volume chip.

Although the STI design center for this ambitious, large-scale project was based in Austin (with IBM, the Sony Group, and Toshiba as partners), the following IBM sites were also critical to the project: Rochester, Minnesota; Yorktown Heights, New York; Boeblingen (Germany); Raleigh, North Carolina; Haifa (Israel); Almaden, California; Bangalore (India); Yasu (Japan); Burlington, Vermont; Endicott, New York; and a joint technology team located in East Fishkill, New York.

Program objectives and challenges

The objectives for the new processor were the following:

- Outstanding performance, especially on game/multimedia applications.

0018-8646/05/$5.00 © 2005 IBM

- Real-time responsiveness to the user and the network.
- Applicability to a wide range of platforms.
- Support for introduction in 2005.

Outstanding performance, especially on game/multimedia applications

The first of these objectives, outstanding performance, especially on game/multimedia applications, was expected to be challenged by limits on performance imposed by memory latency and bandwidth, power (even more than chip size), and diminishing returns from increased processor frequencies achieved by reducing the amount of work per cycle while increasing pipeline depth.

The first major barrier to performance is increased memory latency as measured in cycles, and latency-induced limits on memory bandwidth. Also known as the "memory wall" [5], the problem is that higher processor frequencies are not met by decreased dynamic random access memory (DRAM) latencies; hence, the effective DRAM latency increases with every generation. In a multi-GHz processor it is common for DRAM latencies to be measured in the hundreds of cycles; in symmetric multiprocessors with shared memory, main memory latency can tend toward a thousand processor cycles. A conventional microprocessor with conventional sequential programming semantics will sustain only a limited number of concurrent memory transactions. In a sequential model, every instruction is assumed to be completed before execution of the next instruction begins. If a data or instruction fetch misses in the caches, resulting in an access to main memory, instruction processing can only proceed in a speculative manner, assuming that the access to main memory will succeed. The processor must also record the non-speculative state in order to safely be able to continue processing. When a dependency on data from a previous access that missed in the caches arises, even deeper speculation is required in order to continue processing. Because of the amount of administration required every time computation is continued speculatively, and because the probability that useful work is being speculatively completed decreases rapidly with the number of times the processor must speculate in order to continue, it is very rare to see more than a few speculative memory accesses being performed concurrently on conventional microprocessors. Thus, if a microprocessor has, e.g., eight 128-byte cache-line fetches in flight (a very optimistic number) and memory latency is 1,024 processor cycles, the maximum sustainable memory bandwidth is still a paltry one byte per processor cycle. In such a system, memory bandwidth limitations are latency-induced, and increasing memory bandwidth at the expense of memory latency can be counterproductive. The challenge therefore is to find a processor organization that allows for more memory bandwidth to be used effectively by allowing more memory transactions to be in flight simultaneously.

Power and power density in CMOS processors have increased steadily to a point at which we find ourselves once again in need of the sophisticated cooling techniques we had left behind at the end of the bipolar era [6]. However, for consumer applications, the size of the box, the maximum airspeed, and the maximum allowable temperature for the air leaving the system impose fundamental first-order limits on the amount of power that can be tolerated, independent of engineering ingenuity to improve the thermal resistance. With respect to technology, the situation is worse this time for two reasons. First, the dimensions of the transistors are now so small that tunneling through the gate and sub-threshold leakage currents prevent following constant-field scaling laws and maintaining power density for scaled designs [7]. Second, an alternative lower-power technology is not available. The challenge is therefore to find means to improve power efficiency along with performance [8].

A third barrier to improving performance stems from the observation that we have reached a point of diminishing return for improving performance by further increasing processor frequencies and pipeline depth [9]. The problem here is that when pipeline depths are increased, instruction latencies increase owing to the overhead of an increased number of latches. Thus, the performance gained by the increased frequency, and hence the ability to issue more instructions in any given amount of time, must exceed the time lost due to the increased penalties associated with the increased instruction execution latencies. Such penalties include instruction issue slots[1] that cannot be utilized because of dependencies on results of previous instructions and penalties associated with mispredicted branch instructions. When the increase in frequency cannot be fully realized because of power limitations, increased pipeline depth and therefore execution latency can degrade rather than improve performance. It is worth noting that processors designed to issue one or two instructions per cycle can effectively and efficiently sustain higher frequencies than processors designed to issue larger numbers of instructions per cycle. The challenge is therefore to develop processor microarchitectures and implementations that minimize pipeline depth and that can efficiently use the issue slots available to them.

Real-time responsiveness to the user and the network

From the beginning, it was envisioned that the Cell processor should be designed to provide the best possible

[1] An instruction issue slot is an opportunity to issue an instruction.

experience to the human user and the best possible response to the network. This "outward" focus differs from the "inward" focus of processor organizations that stem from the era of batch processing, when the primary concern was to keep the central processor unit busy. As all game developers know, keeping the players satisfied means providing continuously updated (real-time) modeling of a virtual environment with consistent and continuous visual and sound and other sensory feedback. Therefore, the Cell processor should provide extensive real-time support. At the same time we anticipated that most devices in which the Cell processor would be used would be connected to the (broadband) Internet. At an early stage we envisioned blends of the content (real or virtual) as presented by the Internet and content from traditional game play and entertainment. This requires concurrent support for real-time operating systems and the non-real-time operating systems used to run applications to access the Internet. Being responsive to the Internet means not only that the processor should be optimized for handling communication-oriented workloads; it also implies that the processor should be responsive to the types of workloads presented by the Internet. Because the Internet supports a wide variety of standards, such as the various standards for streaming video, any acceleration function must be programmable and flexible. With the opportunities for sharing data and computation power come the concerns of security, digital rights management, and privacy.

Applicability to a wide range of platforms

The Cell project was driven by the need to develop a processor for next-generation entertainment systems. However, a next-generation architecture with strength in the game/media arena that is designed to interface optimally with a user and broadband network in real time could, if architected and designed properly, be effective in a wide range of applications in the digital home and beyond. The Broadband Processor Architecture [10] is intended to have a life well beyond its first incarnation in the first-generation Cell processor. In order to extend the reach of this architecture, and to foster a software development community in which applications are optimized to this architecture, an open (Linux**-based) software development environment was developed along with the first-generation processor.

Support for introduction in 2005

The objective of the partnership was to develop this new processor with increased performance, responsiveness, and security, and to be able to introduce it in 2005. Thus, only four years were available to meet the challenges outlined above. A concept was needed that would allow us to deliver impressive processor performance, responsiveness to the user and network, and the flexibility to ensure a broad reach, and to do this without making a complete break with the past. Indications were that a completely new architecture can easily require ten years to develop, especially if one includes the time required for software development. Hence, the Power Architecture* was used as the basis for Cell.

Design concept and architecture

The Broadband Processor Architecture extends the 64-bit Power Architecture with cooperative offload processors ("synergistic processors"), with the direct memory access (DMA) and synchronization mechanisms to communicate with them ("memory flow control"), and with enhancements for real-time management. The first-generation Cell processor (**Figure 1**) combines a dual-threaded, dual-issue, 64-bit Power-Architecture-compliant Power processor element (PPE) with eight newly architected synergistic processor elements (SPEs) [11], an on-chip memory controller, and a controller for a configurable I/O interface. These units are interconnected with a coherent on-chip element interconnect bus (EIB). Extensive support for pervasive functions such as power-on, test, on-chip hardware debug, and performance-monitoring functions is also included.

The key attributes of this concept are the following:

- A high design frequency (small number of gates per cycle), allowing the processor to operate at a low voltage and low power while maintaining high frequency and high performance.
- Power Architecture compatibility to provide a conventional entry point for programmers, for virtualization, multi-operating-system support, and the ability to utilize IBM experience in designing and verifying symmetric multiprocessors.
- Single-instruction, multiple-data (SIMD) architecture, supported by both the vector media extensions on the PPE and the instruction set of the SPEs, as one of the means to improve game/media and scientific performance at improved power efficiency.
- A power- and area-efficient PPE that supports the high design frequency.
- SPEs for coherent offload. SPEs have local memory, asynchronous coherent DMA, and a large unified register file to improve memory bandwidth and to provide a new level of combined power efficiency and performance. The SPEs are dynamically configurable to provide support for content protection and privacy.

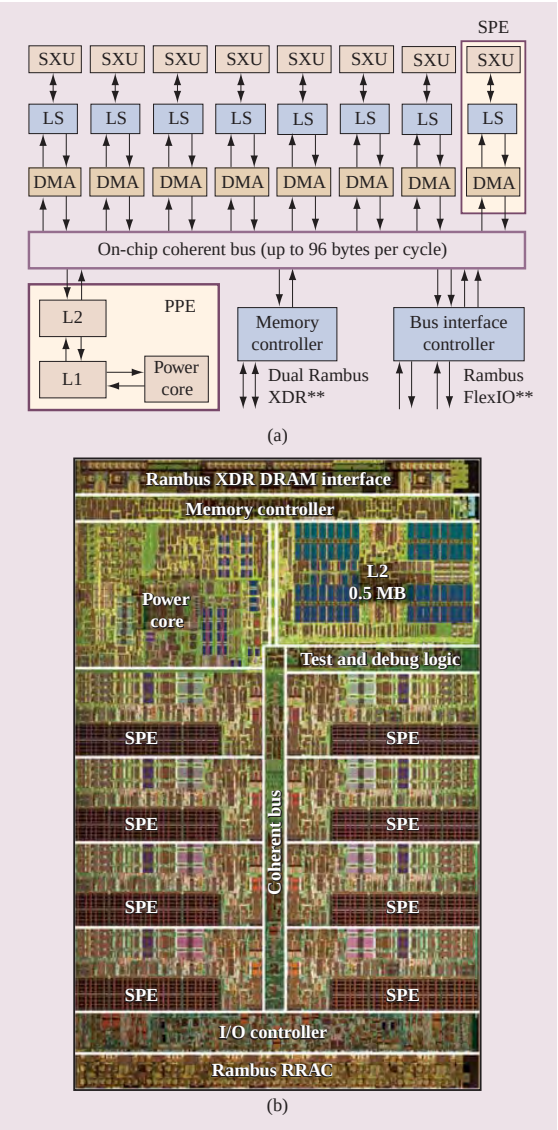

Figure 1

(a) Cell processor block diagram and (b) die photo. The first generation Cell processor contains a power processor element (PPE) with a Power core, first- and second-level caches (L1 and L2), eight synergistic processor elements (SPEs) each containing a direct memory access (DMA) unit, a local store memory (LS) and execution units (SXUs), and memory and bus interface controllers, all interconnected by a coherent on-chip bus. (Cell die photo courtesy of Thomas Way, IBM Burlington.)

- A high-bandwidth on-chip coherent bus and high bandwidth memory to deliver performance on memory-bandwidth-intensive applications and to allow for high-bandwidth on-chip interactions between the processor elements. The bus is coherent to allow a single address space to be shared by the PPEs and SPEs for efficient communication and ease of programming.
- High-bandwidth flexible I/O configurable to support a number of system organizations, including a single-chip configuration with dual I/O interfaces and a "glueless" coherent dual-processor configuration that does not require additional switch chips to connect the two processors.
- Full-custom modular implementation to maximize performance per watt and performance per square millimeter of silicon and to facilitate the design of derivative products.
- Extensive support for chip power and thermal management, manufacturing test, hardware and software debugging, and performance analysis.
- High-performance, low-cost packaging technology.
- High-performance, low-power 90-nm SOI technology.

High design frequency and low supply voltage

To deliver the greatest possible performance, given a silicon and power budget, one challenge is to co-optimize the chip area, design frequency, and product operating voltage. Since efficiency improves dramatically (faster than quadratic) when the supply voltage is lowered, performance at a power budget can be improved by using more transistors (larger chip) while lowering the supply voltage. In practice the operating voltage has a minimum, often determined by on-chip static RAM, at which the chip ceases to function correctly. This minimum operating voltage, the size of the chip, the switching factors that measure the percentage of transistors that will dissipate switching power in a given cycle, and technology parameters such as capacitance and leakage currents determine the power the processor will dissipate as a function of processor frequency. Conversely, a power budget, a given technology, a minimum operating voltage, and a switching factor allow one to estimate a maximum operating frequency for a given chip size. As long as this frequency can be achieved without making the design so inefficient that one would be better off with a smaller chip operating at a higher supply voltage, this is the design frequency the project should aim to achieve. In other words, an optimally balanced design will operate at the minimum voltage supported by the circuits and at the maximum frequency at that minimum voltage. The chip should not exceed the maximum power tolerated by the application. In the case of the Cell processor, having eliminated most of the barriers that cause inefficiency in high-frequency designs, the initial design objective was a cycle time no more than that of ten fan-out-of-four

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.