The Cell architecture

Innovation Matters Source: http://domino.research.ibm.com/comm/research.nsf/pages/r.arch.innovation.html

The Cell architecture grew from a challenge posed by Sony and Toshiba to provide power-efficient and cost-effective high-performance processing for a wide range of applications, including the most demanding consumer appliance: game consoles. Cell - also known as the Cell Broadband Engine Architecture (CBEA) - is an innovative solution whose design was based on the analysis of a broad range of workloads in areas such as cryptography, graphics transform and lighting, physics, fast-Fourier transforms (FFT), matrix operations and scientific workloads. As an example of innovation that ensures the clients' success, a team from IBM Research joined colleagues from IBM Systems Technology Group, Sony and Toshiba, to lead the development of a novel architecture that represents a breakthrough in performance for consumer applications. IBM Research participated throughout the development, implementation and software enablement of the architecture, ensuring the timely and efficient application of novel ideas and technology into a product that solves real challenges.

Cell is a heterogeneous chip multiprocessor that consists of an IBM 64-bit Power Architecture™ core, augmented with eight specialized co-processors based on a novel single-instruction multiple-data (SIMD) architecture called Synergistic Processor Unit (SPU), which is for data-intensive processing, like that found in cryptography, media and scientific applications. The system is integrated by a coherent on-chip bus.

Based on the analysis of available die area, cost and power budgets, and achievable performance, the best approach to achieving the performance target was to exploit parallelism through a high number of nodes on a chip multiprocessor. To further reduce power, the team opted for a heterogeneous configuration with a novel SIMD-centered architecture. This configuration combines the flexibility of an IBM 64-bit Power Architecture™ core with the functionality and performance-optimized SPU SIMD cores.

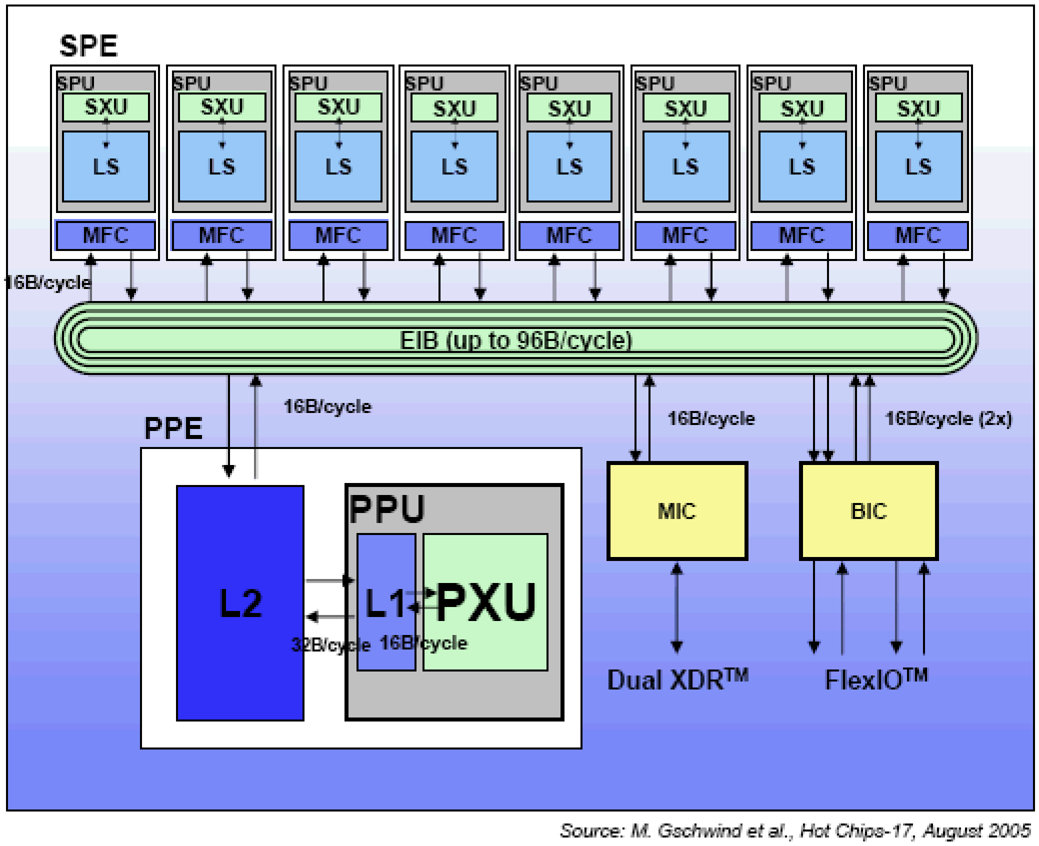
In this organization, the SPU accelerators operate from a local storage that contains instruction and data for a single SPU. This local storage is the only memory directly addressable by the SPU.
The SPU architecture was built to:

- provide a large register file,
- simplify code generation,
- reduce the size and power consumption by unifying resources, and
- simplify decode and dispatch.

These goals were achieved by building a novel SIMD-based architecture with 32-bit wide instructions encoding a three-operand instruction format. Designing a new instruction set architecture (ISA) allowed IBM Research to streamline the instruction side, and provide seven-bit register operand specifiers to directly address 128 registers from all instructions, using a single pervasive SIMD computation approach for both scalar and vector data. In this approach, a unified 128 entry 128-bit SIMD register file provides scalar, condition and address operands, such as for conditional operations, branches and memory accesses.

While the SPU ISA is a novel architecture, the operations selected for the SPU are closely aligned with the functionality of the Power™ VMX unit. This facilitates and simplifies code portability between the IBM 64-bit Power Architecture™ processor and the SPU SIMD-based processors. However, the range of data types supported in the SPU has been reduced for most computation formats. While VMX supports a number of densely packed saturating integer data types, these data types lead to a loss of dynamic range which typically degrades computation results. The preferred computation approach is to widen integer data types for intermediate operations and perform them without saturation. Unpack and saturating pack operations allow memory bandwidth and memory footprint to be reduced while maintaining high data integrity.

Floating point data types automatically support a wide dynamic data range and saturation, so no additional data conditioning is required. To reduce area and power requirements, floating point arithmetic is restricted to the most common and useful modes. As a result, denormalized numbers are automatically flushed to 0 when presented as input, and when a denormalized result is generated. Also, a single rounding mode is supported.



Cell Block Diagram

Single precision floating point computation is geared for throughput of media and three-dimensional graphics objects. In this vein, the decision to support only a subset of IEEE floating point arithmetic and sacrifice full IEEE compliance was driven by the target applications. Thus, multiple rounding modes and IEEE-compliant exceptions are typically unimportant for these workloads, and are not supported. This design decision is based the real time nature of game workloads and other media applications: most often, saturation is mathematically the right solution. Also, occasional small display glitches caused by saturation in a display frame is tolerable. On the other hand, incomplete rendering of a display frame, missing objects or tearing video due to long exception handling is objectionable.

Memory access is performed via a DMA-based interface using copy-in/copy-out semantics, and data transfers can be initiated by either the IBM Power™ processor or an SPU. The DMA-based interface uses the Power Architecture™ page protection model, giving a consistent interface to the system storage map for all processor structures despite its heterogeneous instruction set architecture structure. A high-performance on-chip bus connects the SPU and Power Architecture™ computing elements.

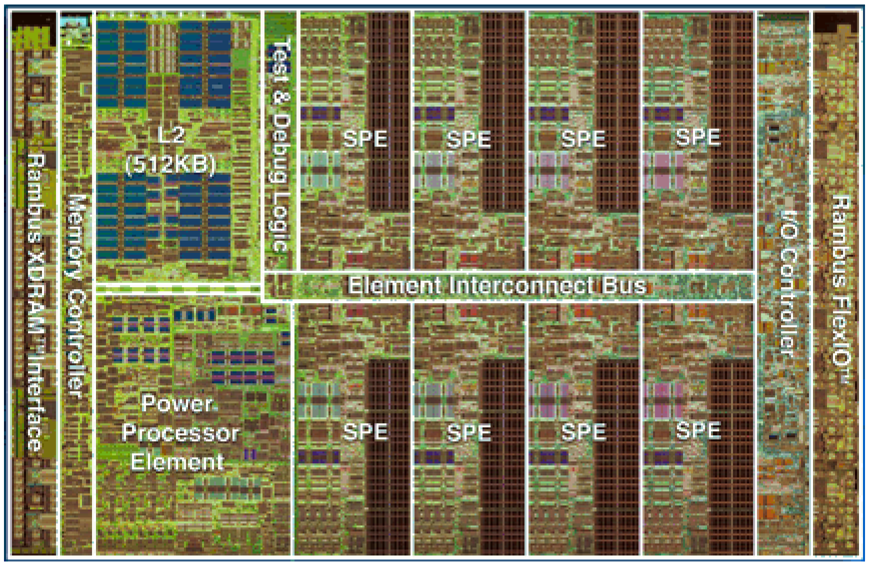
The SPU is an in-order dual-issue statically scheduled architecture. Two SIMD instructions can be issued per cycle: one compute instruction and one memory operation. The SPU branch architecture does not include dynamic branch prediction, but instead relies on compiler-generated branch prediction using "prepare-to-branch" instructions to redirect instruction prefetch to branch targets.

The SPU was designed with a compiled code focus from the beginning, and early availability of SIMD-optimized compilers allowed development of high-performance graphics and media libraries for the broadband architecture entirely in the C programming language.

Based on these decisions to share compute semantics, data types, and virtual memory model, the SPUs synergistically exploit and amplify the advantages when combined with the IBM Power Architecture™ to form the Cell Broadband Engine Architecture.

IBM Research grew its partnership in the development of the broadband processor architecture beyond its initial definition. During the course of this partnership with the STI Design Center, members of the original Cell team developed the first SPU compiler, which was a guiding force for the definition of the SPU architecture and the SPU programming environment, and sample code to exploit the strengths of the broadband processor architecture. The extended partnership led to further contributions by IBM Research, including the development of an advanced parallelizing compiler with auto-SIMDization features based on IBM XL compiler technology, the design of the high-frequency Power Architecture™ core at the center of the Cell architecture, and a full-system simulation infrastructure.

Cell is not limited to game systems. IBM has announced a Cell-based blade, which leverages the investment in the high-performance Cell architecture. Other future uses may include HDTV sets, home servers, game servers and supercomputers. Also, Cell is not limited to a single chip, but is a scalable system. The number of attached SPUs can be varied, to achieve different power/performance and price/performance points. And, the Cell architecture was conceived as a modular, extendible system where multiple Cell subsystems each with a Power Architecture™ core and attached SPUs, can form a symmetric multiprocessor system.



Cell Die

Some Cell statistics:

- Observed clock speed: > 4 GHz
- Peak performance (single precision): > 256 GFlops
- Peak performance (double precision): >26 GFlops
- Local storage size per SPU: 256KB
- Area: 221 mm²
- Technology: 90nm SOI
- Total number of transistors: 234M

Related Publications

An Open Source Environment for Cell Broadband Engine System Software (IEEE Computer, June 2007, pdf) (M. Gschwind, D. Erb, S. Manning, M. Nutter)

The Cell Broadband Engine: Exploiting multiple levels of parallelism in a chip multiprocessor(International Journal of Parallel Programming, June 2007, pdf) (M. Gschwind)

Cell Broadband Engine: Enabling density computing for data-rich environmnents (ISCA 2006 tutorial, Boston, MA, pdf)

Synergistic processing in Cell's multicore architecture (IEEE Micro, March 2006, pdf) (M. Gschwind, P. Hofstee, B. Flachs, M. Hopkins, Y. Watanabe, T. Yamazaki)

Chip multiprocessing and the Cell Broadband Engine (IBM, February 28, 2006, pdf) (M. Gschwind)

Using advanced compiler technology to exploit the performance of the Cell Broadband Engine™ architecture (IBM Systems Journal, Volume 45, Number 1, 2006. HTML and pdf formats)

A novel SIMD architecture for the Cell heterogeneous chip multiprocessor (IBM, 2005, pdf)

Optimizing compiler for the Cell processor (IEEE, September 2005, pdf)

Power efficient architecture and the Cell processor (HPCA-11, February 2005.) (P. Hofstee)