Editor: Michael A. Gray, gray@american.edu

THE PLAYSTATION 3 FOR HIGH-PERFORMANCE SCIENTIFIC COMPUTING

By Jakub Kurzak, Alfredo Buttari, Piotr Luszczek, and Jack Dongarra

Is real-world gaming technology the next big thing in the more academically based high-performance computing arena? The authors put PlayStation 3 to the test.

The heart of the Sony PlayStation 3—the Cell processor—wasn't originally intended for scientific number crunching, just as the PlayStation 3 itself wasn't meant primarily to serve such purposes. Yet, both these items could impact the high-performance computing world in significant ways. This introductory article takes a closer look at their potential to do so; an extended version of it is published as a University of Tennessee technical report (www.cs.utk.edu/~library/2008).

The Cell in a Nutshell

The Cell processor's main control unit is the Power Processing Element (PPE), which is a 64-bit, two-way simultaneous multithreading (SMT) processor that's binary-compliant with the PowerPC 970 architecture. The PPE consists of the Power Processing Unit (PPU), 32 Kbytes of L1 cache, and 512 Kbytes of L2 cache.

Although the PPU uses the PowerPC 970 instruction set, it has a relatively simple architecture with in-order execution, which results in considerably less circuitry than its out-of-order execution counterparts as well as lower energy consumption. The high clock rate, high memory bandwidth, and dual threading capabilities make up for the potential performance deficiencies stemming from the PPU's in-order execution architecture. The SMT feature, which comes at a small 5 percent increase in the hardware's cost, can deliver up to a 30 percent increase in performance. The PPU also includes a short-vector single instruction, multiple data (SIMD) engine called VMX, which is an incarnation of the PowerPC's AltiVec.

However, the Cell processor's real power lies in the eight Synergistic Processing Elements (SPEs) that accompany the PPE. Each SPE consists of a Synergistic Processing Unit (SPU), 256 Kbytes of private memory (referred to as the local store), and a memory-flow controller that delivers powerful direct memory-access capabilities to the SPU. The SPEs are the Cell's short-vector SIMD workhorses and possess a large 128-entry, 128-bit vector register file as well as a range of SIMD instructions that can operate simultaneously on two double-precision values, four single-precision values, eight 16-bit integers, or 16 8-bit characters. Most instructions are pipelined and can complete one vector operation in each clock cycle, including fused multiplication–addition in single precision, which means that the SPU can accomplish two floating-point operations on four values in each clock cycle. This translates to a peak of $2 \times 4 \times 3.2$ GHz = 25.6 Gflop/s for each SPE and adds up to a staggering peak of 8×25.6 Gflop/s = 204.8 Gflop/s for the entire chip. Figure 1 shows a schematic of the Cell processor's design.

All the Cell processor's components, including the PPE, the SPEs, the main memory, and the I/O system, are connected via the element interconnection bus, which has four unidirectional rings (two in each direction) and a token-based arbitration mechanism that plays the role of traffic light. Each participant is hooked up to the bus with a bandwidth of 25.6 Gbytes/s; the bus has an internal bandwidth of 204.8 Gbytes/s, which means that for all practical purposes, you shouldn't be able to saturate it.

The Cell chip draws its power from the fact that it's a parallel machine with eight small, fast, specialized number-crunching and processing elements. The SPEs, in turn, rely on a simple design with short pipelines, a huge register file, and a powerful SIMD instruction set.

The Cell is essentially a distributed-memory system on a chip, on which each SPE possesses its private memory stripped of any indirection mechanisms to make it faster. This puts explicit control over data motion in the hands of the programmer, who must use techniques closely resembling message passing, a model that some might think is challenging but is the only one known to be scalable today.

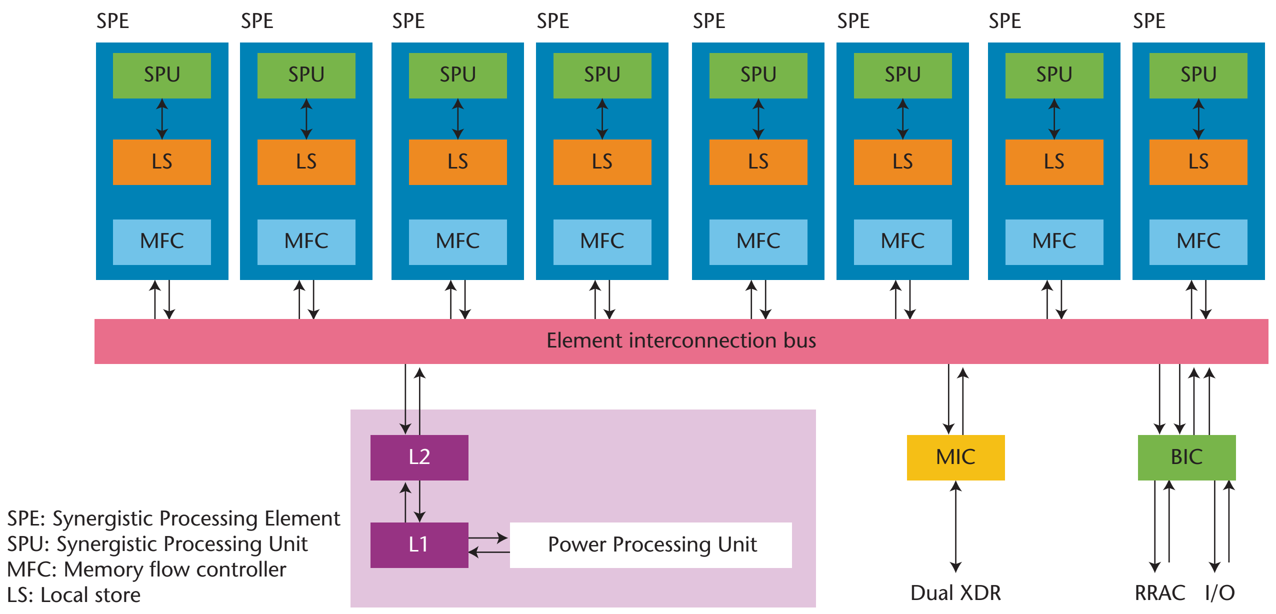
SPE
SPU
LS
MFC
Element interconnection bus
L2
L1
Power Processing Unit
MIC
BIC
Dual XDR
RRAC
I/O
SPE: Synergistic Processing Element
SPU: Synergistic Processing Unit
MFC: Memory flow controller
LS: Local store

Figure 1. Schematic of the Cell processor's design. The main components are the Power Processing Unit, eight Synergistic Processing Elements (SPEs), and the element interconnection bus.

The PlayStation 3

The PlayStation 3 is probably the cheapest Cell-based system on the market: it contains a Cell processor (with the number of SPEs reduced to six), 256 Mbytes of main memory, an NVIDIA graphics card with 256 Mbytes of its own memory, and a gigabit Ethernet (GigE) network card.

Sony made several convenient provisions for installing Linux on the PlayStation 3 in a dual-boot setup. Installation instructions are plentiful on the Web, but the basic gist is that a virtualization layer—also called the hypervisor—separates the Linux kernel from the hardware. Devices and other system resources are virtualized, but Linux device drivers can work with them. The Cell processor in the PlayStation 3 is identical to the one you would find in high-end IBM or Mercury blade severs, with the exception that two SPEs aren't available (one is disabled for chip yield reasons). Nevertheless, a Cell with one defective SPE still passes as a good chip in the PlayStation 3. If all the SPEs are nondefective, a good one is disabled during manufacturing. Another SPE is hidden from the application by the operating system's hypervisor.

The GigE card is accessible to the Linux kernel through the hypervisor, which both makes it possible to turn the PlayStation 3 into a networked workstation and facilitates building PlayStation 3 clusters via network switches. You can program such installations by using the message-passing interface (MPI). The network card has a direct memory-access unit, which you can set up via dedicated hypervisor calls that enable data transfers without requiring the main processor's intervention.

Programming

All Linux distributions for the PlayStation 3 come with the standard GNU compiler suite, including C (GCC), C++ (G++), and Fortran 95 (GFORTRAN), which now also provides support for OpenMP through the GNU GOMP library. The programmer can use OpenMP to exploit the PPE's SMT capabilities. IBM's software development kit for Cell delivers a similar set of GNU tools, along with an IBM compiler suite that includes C/C++ and, more recently, Fortran (with support for Fortran 95 and partial support for Fortran 2003). The kit is available for installation on Cell- or x86-based systems, with code compiled and built in cross-compilation mode, a method often preferred by experts. These tools practically guarantee compilation of any existing C, C++, or Fortran code on the Cell processor, which makes the initial port of any existing software basically effortless.

As Table 1 shows, several programming models and environments have emerged for the Cell processor; it seems to have ignited similar enthusiasm in the scientific high-performance computing, embedded systems, and graphics communities as well. Naturally, the programming techniques proposed for the Cell are as diverse as the communities involved: they include shared-memory, distributed-memory, and stream-processing models and represent both data- and task-parallel approaches.

A separate problem is related to programming for a cluster of PlayStation 3s—such a cluster is essentially a distributed-memory machine, and there's almost no programming

Table 1. Programming environments for the Cell processor.*

	Origin	Available	Free
Cell SuperScalar	Barcelona Supercomputer Center	X	X
Sequoia	Stanford University	X	X
Accelerated Library Framework	IBM	X	X
CorePy	Indiana University	X	X
Multicore Framework	Mercury Computer Systems	X	
Gedae	Gedae	X	
RapidMind	RapidMind	X	
Octopiler†	IBM	X	X
MPI Microtask‡	IBM		

** Available means you can get if for free or buy it as a product*
† Official name is the Single Source Compiler
‡ MPI Microtask is a research project inside IBM; there's no outside access to this software

alternative to using MPI. Several freely available implementations exist, with the most popular being MPICH2 from Argonne National Laboratory and OpenMPI, an open source project in active development by a team of 19 organizations, including universities, national laboratories, companies, and private individuals. Due to the PPE's compliance with the PowerPC architecture, you could compile any of these libraries for execution on the Cell almost out of the box. The good news is that using MPI on a PlayStation 3 cluster isn't any different than on any other distributed-memory system.

Scientific Computing

In spite of its power, the PlayStation 3 has severe limitations for scientific computing. First, it can only achieve its astounding peak of 153.6 Gflop/s for compute-intensive tasks in single-precision arithmetic, which, besides delivering less precision, isn't compliant with the IEEE floating-point standard (its double-precision peak is less than 11 Gflop/s). Second, it only implements truncation rounding, so denormalized numbers are flushed to zero, and NaNs ("not a number") are treated as normal numbers. Finally, memory-bound problems are limited by the main memory's bandwidth of 25.6 Gbytes/s. This is a very respectable value compared to cutting-edge heavy-iron processors, but it sets the upper limit of memory-intensive single-precision calculations to 12.8 Gflop/s and double-precision calculations to 6.4 Gflop/s, assuming two operations are performed on one data element.

However, the largest disproportion in the PlayStation 3's performance is between the Cell processor's speed and that of the GigE interconnection. GigE isn't crafted for performance and, in practice, only about 65 percent of its peak bandwidth can be achieved in MPI communication. Also, because of the extra layer of indirection between the operating system and the hardware (specifically, the hypervisor), the incurred latency can be as big as 200 μs, which is at least an order of magnitude below today's standards for high-performance interconnections. Even if you could lower the latency and gain a larger fraction of the peak bandwidth, the communication capacity of 1 Gbyte/s is way too small to keep the Cell processor busy. A common remedy for slow interconnections is to run larger problems, but in this case, the main memory's small size (256 Mbytes) turns out to be the limiting factor.

Taken all together, even simple examples of compute-intensive workloads such as matrix multiplication can't benefit from running in parallel on more than two PlayStation 3s. Rather, only the extremely compute-intensive, embarrassingly parallel problems have a fair chance of success in scaling to PlayStation 3 clusters. Such distributed computing problems, often referred to as screen-saver computing, have gained popularity in recent years: the trend initiated by the SETI@Home project had many followers, including the very successful Folding@Home project.

The idea of many-core processors reaching hundreds, if not thousands, of processing elements per chip is emerging, with some researchers aiming for distributed-memory systems on a chip, an inherently more scalable solution than shared-memory setups. Owing to this, the technology delivered by the PlayStation 3 through its Cell processor provides a unique opportunity to gain experience, which is likely to be priceless in the near future.

But a major shortcoming of the current Cell processor for numerical applications is the relatively slow speed of double-precision arithmetic. The next incarnation promises to include a fully pipelined double-precision unit that will deliver the speed of 12.8 Gflop/s from a single SPE clocked at 3.2 GHz and 102.4 Gflop/s from an eight-SPE system, which is going to make the chip a very serious competitor in the world of scientific and engineering computing.

Although in agony, Moore's law is still alive, and we're entering the era of billion-transistor processors. Given this, the current Cell processor uses a rather modest number of transistors (234 million). It isn't hard to envision a Cell processor with more than one PPE and many more SPEs, perhaps exceeding the performance of one teraflop/s for a single chip.

Jakub Kurzak is a senior research associate in the Department of Computer Science at the University of Tennessee. His research interests include high-performance computing, parallel programming, and numerical algorithms. Kurzak has a PhD in computer science from the University of Houston. Contact him at kurzak@eecs.utk.edu.

Alfredo Buttari is a scientist at the French National Institute for Research in Computer Science and Control (INRIA). His research interests include high-performance computing, parallel programming, and numerical algorithms. Buttari has a PhD in computer science from the University of Rome. Contact him at buttari@eecs.utk.edu

Piotr Luszczek is a scientist at MathWorks. His research interests include high-performance computing, parallel programming, and numerical algorithms. Luszczek has a PhD in computer science from the University of Tennessee. Contact him at luszczek@eecs.utk.edu

Jack Dongarra is University Distinguished Professor in the Department of Electrical Engineering and Computer Science at the University of Tennessee, Distinguished Research Staff in the Computer Science and Mathematics Division at the Oak Ridge National Laboratory, and a professor in the School of Mathematics and School of Computer Science at the University of Manchester, UK. His research interests include high-performance computing, parallel programming, and numerical algorithms. Dongarra has a PhD in applied mathematics from the University of New Mexico. Contact him dongarra@eecs.utk.edu.

DEPARTMENT EDITORS

Books: Mario Belloni, Davidson College, mabelloni@davidson.edu
Computing Prescriptions: Isabel Beichl, Nat'l Inst. of Standards and Tech., isabel.beichl@nist.gov
Computer Simulations: Muhammad Sahimi, University of Southern California, moe@iran.usc.edu, and Dietrich Stauffer, Univ. of Köhn, stauffer@thp.uni-koeln.de
Education: Michael Dennin, Univ. of Calif., Irvine, mdennin@uci.edu, and Steven F. Barrett, Univ. of Wyoming, steveb@uwyo.edu
News: Rubin Landau, Oregon State Univ., rubin@physics.oregonstate.edu
Scientific Programming: Konstantin Läufer, Loyola Univ., Chicago, klaufer@cs.luc.edu, and George K. Thiruvathukal, Loyola Univ., Chicago, gkt@cs.luc.edu
Technologies: Michael A. Gray, American Univ., gray@american.edu, and James D. Myers, Collaborative Systems, NCSA, jimmyers@ncsa.uiuc.edu
Visualization Corner: Cláudio T. Silva, Univ. of Utah, csilva@cs.utah.edu, and Joel E. Tohline, Louisiana State Univ., tohline@rouge.phys.lsu.edu

STAFF

Senior Editor: Jenny Stout, jstout@computer.org
Senior Editorial Services Manager: Crystal R. Shif
Magazine Editorial Manager: Steve Woods
Staff Editors: Kathy Clark-Fisher, Rebecca L. Deuel, Shani Murray, and Brandi Ortega
Production Editor: Monette Velasco
Publications Coordinator: Hazel Kosky, cise@computer.org
Technical Illustrator: Alex Torres

Senior Advertising Coordinator: Marian Anderson
Marketing Manager: Georgann Carter
Senior Business Development Manager: Sandra Brown

AIP Staff:
Circulation Director: Jeff Bebee, jbebee@aip.org
Editorial Liaison: Charles Day, cday@aip.org

IEEE Antennas and Propagation Society liaison: Don Wilton, Univ. of Houston, wilton@uh.edu
IEEE Signal processing Society liaison: Elias S. Manolakos, Northeastevrn Univ., elias@neu.edu

CS PUBLICATIONS BOARD

Sorel Reisman (chair), Angela Burgess, Chita R. Das, Richard H. Eckhouse, Van Eden, Frank E. Ferrante, David A. Grier, Pamela Jones, Phillip A. Laplante, Simon Liu, Paolo Montuschi, Jon Rokne, Linda I. Shafer, Steven L. Tanimoto

CS MAGAZINE OPERATIONS COMMITTEE

Robert E. Filman (chair), David Albonesi, Arnold (Jay) Bragg, Carl Chang, Kwang-Ting (Tim) Cheng, Norman Chonacky, Fred Douglis, Hakan Erdogmus, James Hendler, Carl E. Landwehr, Dejan Milojicic, Sethuraman (Panch) Panchanathan, Maureen Stone, Roy Want, Jeff Yost

EDITORIAL OFFICE

COMPUTING IN SCIENCE & ENGINEERING
10662 Los Vaqueros Circle, Los Alamitos, CA 90720 USA
Phone +1 714 821 8380; www.computer.org/cise/

IEEE Antennas & Propagation Society

IEEE Signal Processing Society