Cell Broadband Engine processor: Design and implementation

M. W. Riley
J. D. Warnock
D. F. Wendel

The Cell Broadband Engine™ (Cell/B.E.) processor was developed by Sony, Toshiba, and IBM engineers to deliver a high-speed, high-performance, multicore processor that brings supercomputer performance via a custom system-on-a-chip (SoC) implementation. To achieve its goals, the Cell/B.E. processor uses an innovative architecture, new circuit design styles, and hierarchical integration and verification techniques. The Cell/B.E. processor design point was also targeted at high-volume manufacturing. To meet high-volume manufacturing requirements, the chip was designed so that it could be completely tested in less than 26 seconds. In addition to the above items, the Cell/B.E. processor was designed with the "triple design constraints" of maximizing performance while minimizing area and power consumed. The initial application was targeted at real-time systems that require high-speed data movement for both on-chip and off-chip transfers. This application also required very high speed compute and real-time response processes.

Introduction: Overview of the Cell Broadband Engine processor

The Cell Broadband Engine[†] (Cell/B.E.) processor is a high-performance system on a chip (SoC). It contains processing elements, a high-speed interconnect, a high-speed memory controller, a high-speed input/output (I/O) controller, and global control and debug facilities. **Figure 1** shows a high-level block diagram of the Cell/B.E. processor chip. The Cell/B.E. processor uses two processing elements: the PowerPC* processor element (PPE) and the synergistic processor element (SPE).

The PPE complies with the 64-bit implementation of the PowerPC Architecture* [1]. It is a dual-threaded processing core that includes an integer unit, a floating-point (FP) unit, a vector multimedia extensions (VMX) unit, and a memory management unit (MMU). The PPE instruction cache is 32 KB and the data cache is 32 KB. In addition, a 512-KB Level 2 (L2) cache is included in the PPE complex, which also has a memory flow controller (MFC) that enables it to perform direct memory access (DMA) transfers to and from main memory, SPEs, or I/O. The MFC also provides memory-mapped I/O (MMIO) transfers to on-chip and off-chip devices. Though based on the PowerPC Architecture, the PPE is a new "bottoms-up" processor design; it is pipelined extensively to allow it to operate at frequencies greater than 3 GHz. Program execution is performed in order. Designed into the PPE, allocation management allows portions of a resource time to be allocated to a specific resource allocation group. This capability assists in developing real-time applications. All caches are covered by an allocation management scheme. Communication to and from SPEs can be performed via DMA and/or mailboxes. The PowerPC Architecture hypervisor extension is also included in the design to allow multiple operating systems to run simultaneously via thread management.

The SPE [2] is a new processor architecture that is designed to accelerate media and streaming operations. The Cell/B.E. uses eight copies of the SPE per chip. Since the design point required the use of multiple SPEs, the SPE was developed with reduction of power consumption as a top priority. To minimize power consumption and maximize performance, the design made hardware and software trade-offs. The SPE is a single-instruction multiple-data (SIMD) processor with a 128-bit data flow. Many of the instructions are 128-bit operands that are divided into 32-bit words. To improve performance, the

0018-8646/07/$5.00 © 2007 IBM

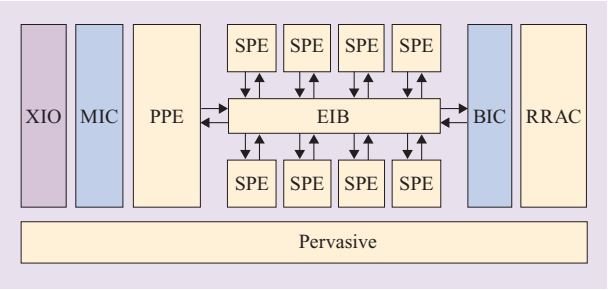

Figure 1

Cell/B.E. processor high-level diagram. (XIO: Rambus XDR controller I/O; MIC: memory interface controller; EIB: element interconnect bus; BIC: bus interface controller; RRAC: Redwood Rambus ASIC Cell.)

128-bit operands are stored in a 128-entry unified register file. The register file can contain integer, FP, and conditional operation operands. In addition to the register file, the SPE has a 256-KB local store (LS) static random access memory (SRAM) [3]. Loads, stores, and instruction fetch operations can occur from the LS. Access to the LS can be 6 bytes or 128 bytes per access. Each SPE has an MFC that complies with the PowerPC Architecture. The MFC provides address translation to the effective address and then asynchronously transfers (relative to the SPE compute element) data between the LS and main storage.

The Cell/B.E. processor main memory connection is an on-chip memory interface controller (MIC) unit, which interfaces to the Rambus XDR** controller I/O (XIO) unit. The XIO unit communicates directly to XDR dynamic RAM (DRAM) modules and provides the appropriate timing interface. The MIC provides the high-level memory control functions and the interface to the processor cores. The MIC supports two Rambus XDR memory channels, each of which is 36 bits in width with an independent control bus. The two channels are interleaved in the address space of the processor, but for increased system flexibility, the MIC can be configured to support just a single channel of Rambus XDR memory. For added reliability, the MIC supports error-correction code (ECC) and a periodic ECC scrub of memory. The Rambus XDR interface operates asynchronously to the processor and I/O interfaces (IOIFs); hence, the MIC contains speed matching SRAM buffers and logic, and it requires two clock domains. One clock domain operates at half the global processor clock rate, while the other domain operates at half the rate of the Rambus XDR interface. When the processor frequency is lower than that of the Rambus XDR interface, the MIC is configured to pace memory requests in order to avoid overrunning the SRAM buffers. The asynchronous interface provides greater flexibility, and the separation of clocks allows the processor to be stopped and its system state scanned out without affecting the training of the transceiver. Memory I/O training is required because of the high speed of the Rambus XDR interface. The memory I/O training sequence is performed by either the PPE or one of the eight SPEs.

The bus interface controller (BIC) provides the mechanism for Cell/B.E. processors to communicate with system components. The BIC consists of two flexible interfaces with programmable protocols and bandwidth capabilities. This functionality provides the capability for the IOIF to be configured for different systems to maximize the cost performance for the specific system. The IOIFs can be configured as two non-coherent IOIFs (IOIF 0 and IOIF 1) or as one IOIF and a coherent symmetric multiprocessor (SMP) bus interface (IOIF and BIF). The BIC communicates with the Rambus FlexIO** interface via the Redwood Rambus ASIC Cell (RRAC). When configured in BIF mode, there are seven transmitter and five receiver bytes of Rambus RRAC I/Os providing substantial bandwidth support. For increased flexibility, the RRAC transmitters and receivers operate asynchronously to the processors and memory, and the available bandwidth is configurable between the two interfaces. The interface of the RRAC can also be run at half rate, a design feature that provides the support for multiple configurations without the need to redesign I/O control logic or repackage the chip to accommodate different I/O configurations. Communication between processor elements and I/O occurs on the element interconnect bus (EIB). The BIC manages the asynchronous interface that exists between the EIB and the two IOIFs. The BIC is involved in synchronizing data transfers between three different clock domains. The BIC interfaces with the core clock domain at one-half the core clock frequency and with the I/O side of the RRAC at one-third the bit rate frequency of the RRAC I/O blocks. A distribution network inside of the BIC operates at one-half the operating frequency of the RRAC I/O blocks. To synchronize the different clock domains, the BIC contains speed-matching SRAM buffers and synchronization logic to manage the three clock domains. The high-speed transmitters and receivers require training at the bit level and byte level for proper operation. To assist with these training exercises, an elastic buffer is used to eliminate the skew between the bytes comprising the interface. Training of the RRAC interface requires an external service controller.

The EIB connects the processing elements, memory, and I/O devices and is fundamentally the central expressway for transfer of data for the Cell/B.E. processor. The EIB is a coherent bus, and it is organized

as four 16-byte-wide rings, each of which supports up to three simultaneous data transfers. The transfers occur at half the core clock frequency, which results in an effective transfer rate of 96 bytes per processor cycle. A separate address-and-command network manages bus requests and the coherence protocol. There are 12 on-ramps and 12 off-ramps that are used by the processors, memory, and I/O to connect to the EIB, which is located horizontally along the center of the chip. Each ramp is 16 bytes wide and operates at half the core clock rate.

The pervasive unit manages overall chip operation for the Cell/B.E. processor and contains all of the global logic needed to enable basic functional operation of the chip. The pervasive unit is also involved in the management of laboratory debugging facilities and manufacturing test facilities. Debugging mechanisms have been incorporated into the design to assist with the bring-up and characterization of the logic and circuits during functional operation.

The pervasive unit provides several mechanisms that are needed for functional operation of the chip. A serial peripheral interface (SPI) bus is provided to enable an external controller to communicate with the chip during power up and initialization of the chip. During normal functional operation, an external controller can use the SPI to receive status information from the pervasive unit. During training of the BIC and RRAC interfaces, an external controller uses the SPI to communicate with the BIC and RRAC and the chips that are to be trained to communicate with the Cell/B.E. processor.

Control for clock generation and distribution is also provided by the pervasive unit. Clock control logic determines which clock is used by the logic during functional, debug, and test operations. The Cell/B.E. processor can operate from the internal phase-locked loop (PLL), the reference clock for the PLL (bypass the PLL), or an external clock. The clock control logic is also responsible for putting the correct clock on each of the three clock grids (core, BIC, and MIC). For example, during the power-on reset (POR) sequence, the core clock grid initially uses the reference clock and then switches to the PLL output once the PLL frequency is locked. Some of the logic in the core clock domain runs at half frequency. The clock control logic in the pervasive unit creates the half clock frequency and ensures that the phase of the half clock control signal is correct throughout the Cell/B.E. chip.

The POR sequence for the Cell/B.E. processor involves several steps and is dependent on the operating mode of the chip that is being used in the system. The operating mode is determined by selecting four pins on the chip. Tying these pins high or low at the board level determines the operating mode. Based on the mode of operation, the POR sequence can be altered. To manage the level of complexity required by the different possible POR sequences, a 32-instruction programmable state machine was created. The POR engine has four hard-coded initialization sequences that can be selected by the configuration pins on the chip. In debug mode, the POR engine can be programmed to provide custom sequences. The programming capability of the POR engine allows instructions to be single stepped, skipped, or performed out of order. This flexibility was very useful in debugging early hardware.

The pervasive unit contains chip-level error checking and reporting mechanisms in the form of global fault isolation registers. These registers can be used during normal operation of the chip and during laboratory (lab) debugging. For lab debugging, recoverable errors can be reported as non-recoverable errors in order to cause a *checkstop*[1] condition. Global fault isolation registers are also provided to allow the operating system to quickly determine which unit generated an error condition. Once an error is determined at the global level, the diagnostic software can then interrogate the units that reported the problem for further problem isolation.

Performance monitoring (PFM) capability is provided in the Cell/B.E. processor to assist with software and hardware analysis. This data can be used for application and system optimization. The Cell/B.E. processor PFM facility allows acquisition of performance information from each of the processor cores and logic islands on the chip. The PFM facility consists of a centralized unit connected to all functional units on the chip via a trace/debug bus. The PFM facility can create performance histograms by storing information in the on-chip trace array. The trace array is a local SRAM that allows 1,024 64-bit words to be stored. Two copies of the trace array are used. The PFM facility shares some of the controls and the trace array with an on-board trace logic analyzer (TLA). The TLA has the capability to capture and store internal signals while the chip is running at full speed. The TLA is programmable and allows complex trace/capture sequences to be created. Capabilities provided by the TLA are similar to features provided by lab workbench logic analyzers.

For lab debug, an on-chip control processor (COP) is provided. Communication with the COP logic is performed via the industry standard IEEE 1149.1 interface. External debug tools such as RISCWatch* [4] can be used to interrogate the internals of the Cell/B.E. chip. Custom instructions have been created to access features of the pervasive unit; these instructions allow the Cell/B.E. processor to be scanned, registers to be read, and processor control functions to be performed (e.g., single instruction step).

[1]When hardware detects a condition that it cannot resolve and that prevents normal operation, it stops executing instructions in order to enable further debugging.

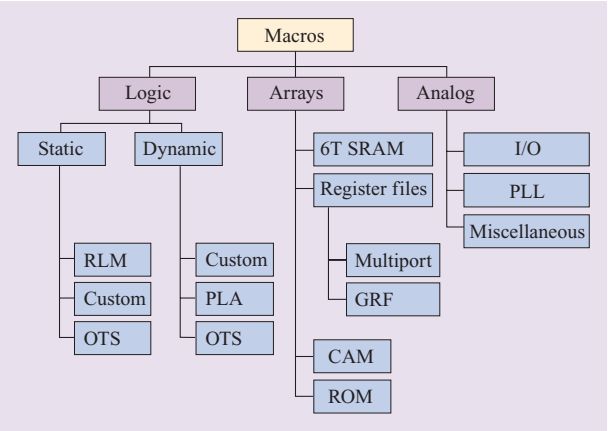

Figure 2

Macro types. (RLM: random logic macro; PLL: phase-locked loop; PLA: programmable logic array; GRF: growable register file; OTS: off the shelf; CAM: content addressable memory; 6T SRAM: six-transistor static random access memory.)

Reducing manufacturing test time was a priority for the Cell/B.E. processor design. The pervasive unit provides 11 test modes that determine the interface that the automatic test equipment connects to and reconfigures the Cell/B.E. processor I/O test pins and internal scan chains. The different modes, which include scanning and test debug modes, allow the Cell/B.E. processor to be placed in standard test scenarios. Array built-in self test (ABIST) and logic BIST (LBIST) are controlled by the pervasive unit. There are 44 ABIST engines in the Cell/B.E. processor that have a fixed pattern set and are programmable to provide debugging capability and the ability to add customized test patterns during manufacturing tests. All ABIST engines are operated in parallel to reduce test time. The LBIST mechanism consists of a centralized LBIST controller that is housed in the pervasive unit and 15 LBIST satellites that reside in the processor cores and logic blocks of the chip. Both ABIST and LBIST are capable of scanning and capturing data at full functional speed to improve ac test speed and coverage. All internal scan chains are timed at the full functional speed to ensure support for full-speed operation of ABIST and LBIST.

The pervasive unit contains the electronic fuses (eFUSEs) and their control. eFUSEs are used on the Cell/B.E. processor to perform array repair actions on arrays that have redundant elements. eFUSEs are also used to program the default parameters for the PLL. This allows the Cell/B.E. processor to be customized for different systems and customers with a simple program change for the eFUSEs during the manufacturing test operation. eFUSEs are also used to store information that customizes the Cell/B.E. processor, including IDs and information related to thermal calibration and operation. A unique 48-bit customer ID can be defined by the customer.

Power dissipation and thermal management are key issues that were considered early in the development of the Cell/B.E. processor. A power management unit (PMU) provides a mechanism to allow software controls to reduce the chip power when the full processing capabilities are not needed. The PMU allows the operating system to throttle, pause, or stop single or multiple units or the entire chip, in order to manage chip power. For thermal monitoring and control, the processor employs two types of thermal sensors and a hardware-controlled thermal management unit (TMU). A linear diode connected to two module pins allows an external device to monitor the temperature of the processor. This sensor is located in a relatively constant temperature location, giving a reading of the global temperature of the processor. This sensor is designed to be used for controlling external cooling mechanisms (e.g., fans). Ten digital thermal sensors are distributed on the chip to monitor temperatures in critical local regions. The TMU can be programmed to interrupt the processor when specified temperatures are observed by the sensors. The TMU also has the ability to shut down the chip clocks automatically if the maximum chip temperature is exceeded. This is a self-protection mechanism for the chip.

Hierarchy

To meet the design challenges that were presented, the Cell/B.E. processor design team chose a hierarchical design style. Using a hierarchy allowed the design work to proceed in parallel and be at different levels. This divide-and-conquer approach helped greatly in meeting the aggressive design schedule.

The fundamental circuit building block is the transistor macro. A *transistor macro* is the minimum placeable object; simply stated, it is a circuit element that provides a particular function. A transistor macro consists of 0–1,000 transistors. A buffer is an example of a macro. The Cell/B.E. processor is designed with digital and analog transistor macros. **Figure 2** shows the different macro types that are in the design. The next macro in the hierarchy is the logic macro. A logic macro consists of several transistor macros. Next in the hierarchy are unit macros. Unit macros consist of several logic macros that create a function. Units in general can be simulated as a single entity. After the unit macro, there is the island, which consists of all the macros to make a full functional element of the Cell/B.E. processor. An example of an island is the L2 cache complex. The next step of the hierarchy is the partition. A partition is a fully functional

unit. The PPE and SPE are classified as partitions. Partitions are grouped to form the chip. The chip is the top stage of the partition. Similar to the design activities, verification of the design is also developed in a hierarchy. The lowest level of the hierarchy requires the least amount of computing resources while the highest level (chip) requires the most computing resources for verification. This aspect requires that detailed operations be confirmed at low levels in the hierarchy while global functions are verified at the chip level.

Clocking

The Cell/B.E. processor design team evaluated different clocking styles that could be used for the design. The general scan design (GSD) approach was selected. With the GSD clocking approach, only one clock is used for both functional and test operations. A single dedicated pin is used to determine whether the chip is in functional mode or in test mode.

The Cell/B.E. processor contains three clock distribution networks. The largest of the distributions covers 85% of the chip. This grid is normally referred to as the *core clock grid*. The memory subsystem and I/O subsystems each contain a unique clock grid. All clock grids have independent PLLs that run asynchronously to one another. The core clock grid overlaps the memory and I/O grids in the BIC and MIC units. The BIC and MIC contain circuits that operate on the core clock and the I/O or memory grids, respectively. During chip initialization and test activity, all of the circuits of BIC and MIC can be forced to run off the core clock grid. This capability simplifies the initialization of the chip and allows for manufacturing tests to scan all of the BIC and MIC latches without hazards. During functional operation, BIC and MIC can operate from the core and their respective native clock network.

The clock grid is a highly tuned network. The initial Cell/B.E. processor design contains 850 individually tuned buffers. Since clock loads are not uniformly distributed throughout the chip, tuning is required to balance clock loads to improve clock frequency and account for varying clock loads. The Cell/B.E. processor clock distribution network was modeled after an earlier PowerPC processor clock distribution network [5]. However, enhancements were made that reduce the power dissipation by 20%. The enhancements include reducing clock load, reducing capacitance on lower-level twig and mesh wires, and reducing capacitance on clock buffers. The clock grids were implemented on the final two layers of metal, which have the lowest level of impedance. Design of the clock grid includes the use of wire simulation models that include frequency-sensitive inductance and resistance elements. Extracted capacitance was also used in the simulations. Custom tools were created that allow the entire clock grid to be simulated in less than four hours.

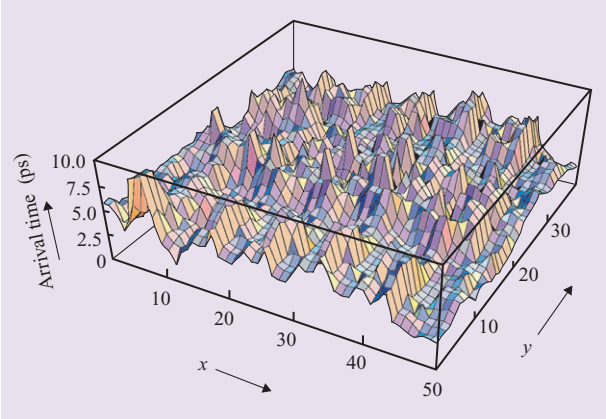

Figure 3

Clock arrival time (z) vs. location (x, y) on chip. Main clock distribution normalized arrival time is within 12 ps across the chip. 325 μm/unit in x and y dimension. (©2005 IEEE. Reprinted, with permission, from [1].)

The clock grids provide a low skew clock to all portions of the chip. Simulation of the grid shows that the worst-case skew is less than 12 ps. **Figure 3** shows the clock skew across the chip. The clock signal must also be distributed locally to clocked elements (e.g., latches). A local clock buffer (LCB) is used to distribute the clock locally. The Cell/B.E. processor LCB structure allows for cycle-accurate mode switching from functional mode to test mode. The LCB provides the gain for driving the clocked devices and the gating mechanisms needed for ac power savings and test operations.

The LCB provides a clock-splitter function. The latches used in the Cell/B.E. processor design are master–slave latches that provide edge-triggered operation. The LCB takes in the single-phase base clock from the clock grid and generates the local clock. **Figure 4** shows the block diagram for the basic LCB.

Inputs to the LCB include the grid clock (global clock), scan, clockgate_b, and testhold_b. Outputs are load clock (lclk) and two data capture clocks, d1clk and d2clk. The d1clk is used during functional operation of the chip, and the d2clk is used during scan test operations.

Referring to the block diagram, "global clock" is the grid clock. This signal is a hard connection to the clock grid. The clockgate_b signal is used by the functional logic to gate the outputs of the LCB. This effectively causes the latch(es) connected to the LCB to stop. This signal is used by the logic designer to gate logic functions (e.g., state machines) and to reduce ac power by gating latch clocks when latch update is not needed. The "scan"

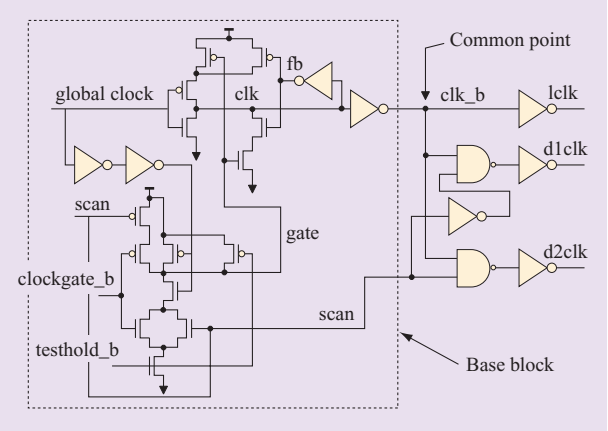

Figure 4

Basic local clock components. (©2005 IEEE. Reprinted, with permission, from [1].)

input is a test control signal that indicates that the operation is a test scan cycle. When the scan input is active, the clockgate_b signal is disabled. The testhold_b signal has the overall control of starting and stopping the LCB outputs. When this signal is a logic "1," the LCB is free to generate clocks. When the signal is "0," clocks are stopped. Note that the core clock grid is never stopped; only the outputs from the LCB are stopped with the clock gating signals. All signals to the LCB are timed at full operational speed. This allows for the latches of the chip to be scanned at full operation speed and the capability to go from functional to scan mode on a per-clock-cycle basis.

Circuits

The power, frequency, and area requirements of the Cell/B.E. processor introduced many challenges to the circuit design team. Since the Cell/B.E. processor was targeted at high-volume production, the circuit team also had to consider techniques that would maximize manufacturing yield. Several circuit design styles were used. The circuit library that was created contains a mixture of custom-designed and synthesized logic. Custom designs include data-flow elements, SRAM arrays, register files, specialty analog circuits, I/Os, and thermal sensors. Random logic macros (RLMs) are synthesized from VHSIC[2] hardware description language (VHDL) using the design toolset.

Most of the circuits are static. Many design productivity tools were developed to aid in the optimization of the static complementary metal-oxide semiconductor (CMOS) designs. RLMs are predefined to always be static. Custom macros can all be static or they can have a combination of static and dynamic logic. Dynamic logic was used in special cases in which performance is critical. There is a small number of dynamic critical macros. Some SRAMs also use dynamic circuits to meet critical timing requirements.

Latch libraries were developed to assist the logic designers in meeting the defined power, frequency, and area requirements. In addition to the basic transmission gate flip-flops, several special-purpose designs were created. For example, a high-performance latch (HPL) was developed that is a hybrid static and dynamic latch. The HPL accepts static inputs and drives static outputs; it also has a multiplexer (mux) and a latch function. The mux is a dynamic NOR that can be up to five inputs wide. Another power-saving specialty latch is the pulsed latch. This latch is operated with pulsed clocks to save power. The specialty latches can also be controlled with the clockgate_b and testhold_b signals and can be scanned during test operations. Another performance- and area-saving measure that was taken with the latch library is the introduction of non-scan latches. Although these latches are not scanned during the test process, they are controllable. Since these devices do not scan, the scan control signal on the LCB is replaced with a force_update signal. This force_update signal allows the manufacturing test patterns to control when the latch data is updated during the test process. **Figure 5(a)** shows the circuit diagram for the master–slave latch. **Figure 5(b)** shows the circuit diagram for the HPL latch, and **Figure 5(c)** shows the clocking mechanism for the pulsed latch.

SRAM arrays

The Cell/B.E. processor contains more than 270 arrays that are used throughout the design. Performance, power, and area constraints drove different designs for the memory arrays. The PPE and SPE contain the largest SRAM devices.

The PPE has two 32-KB L1 SRAM macros that are organized as two-way associative memory. Interleaving is used with the L1 SRAM macros to improve parity-checking performance. The read latency of the L1 SRAM is three cycles: The first cycle is used by the sum address decoder (SAD) for address generation. The second cycle is used to access the array, and the third cycle is used for parity checking, data formatting, and way select. An L1 SRAM macro is internally constructed as 512 wordlines that access 32 bytes (with parity per byte) per way. Writing of the L1 is performed one cache line at a time and can be 64 bytes or 32 bytes. The writing of one 64-byte-wide cache line addresses two consecutive wordlines. Data that is 16 bytes wide can be read with parity checking on each access. The L1 caches are clocked at full clock rate of the core clock grid.

[2] *VHSIC* stands for very high speed integrated circuit.

The L2 cache is a 512-KB eight-way associative cache. The cache is constructed of four $1{,}024 \times 8 \times 140$ SRAM macros. Unlike the L1 caches, the L2 cache is clocked at one-half of the clock rate of the core clock grid. Power conservation is achieved by decoding the way selected prior to accessing the array and then activating one-eighth of the L2 macro. The L2 array can be accessed as a 140-bit or a 280-bit read. Writes to the array are pipelined and completed in two cycles. Read operations complete in three cycles for the first 140 bits of data. For 280 bits of data, the second 140 bits of data can be completed in the fourth cycle. Writes and reads can be interleaved to provide continuous data transfers to and from the L2 array.

The LS array is a 256-KB SRAM array that consumes one-third of the total SPE area. The LS macro consists of a sum address decoder, four 64-KB memory arrays, write accumulation buffers, and read accumulation buffers. The LS completes writes in four cycles and reads in six cycles. The first cycle is used by the SAD to add operands. The second cycle is used to distribute the pre-decoded indices to the four 64-KB subarrays. During the third cycle, decoding of the address is completed and the wordline is selected. The addressed subarray is addressed in the fourth cycle. For reads, the sense amplifier senses the bitline differential signal and holds the value until it is captured in the read latch. In the sixth cycle, the data is forwarded to the executions units. The LS runs at full core clock frequency.

SPE general-purpose register file

The SPE register file has six read ports and two write ports. There are 128 entries that are each 128 bits wide. Access requires three clock cycles. The first cycle is used for address pre-decoding and decoded signal distribution. The second cycle is for final decoding of the address and array access. The final cycle is for data-flow distribution. Eight operations can be performed every clock cycle (six reads and two writes). Collision avoidance is handled by the SPE control logic. The design uses a dynamic two-stage domino read scheme and a static write scheme.

Dynamic programmable logic array

Dynamic programmable logic arrays (DPLAs) are generated and used in the high-speed control logic for the SPE to meet the frequency, power, and area requirements. The DPLA AND terms are implemented with a dynamic footed NOR gate followed by a strobe circuit. When the input pattern to an AND term results in a true output, the precharged AND node stays high. When the local clock signal arrives at the strobe circuit, the strobe node will discharge to the low state, causing the input to the OR gate plane to be pulled high. For the case in which the input pattern to an AND term results in a

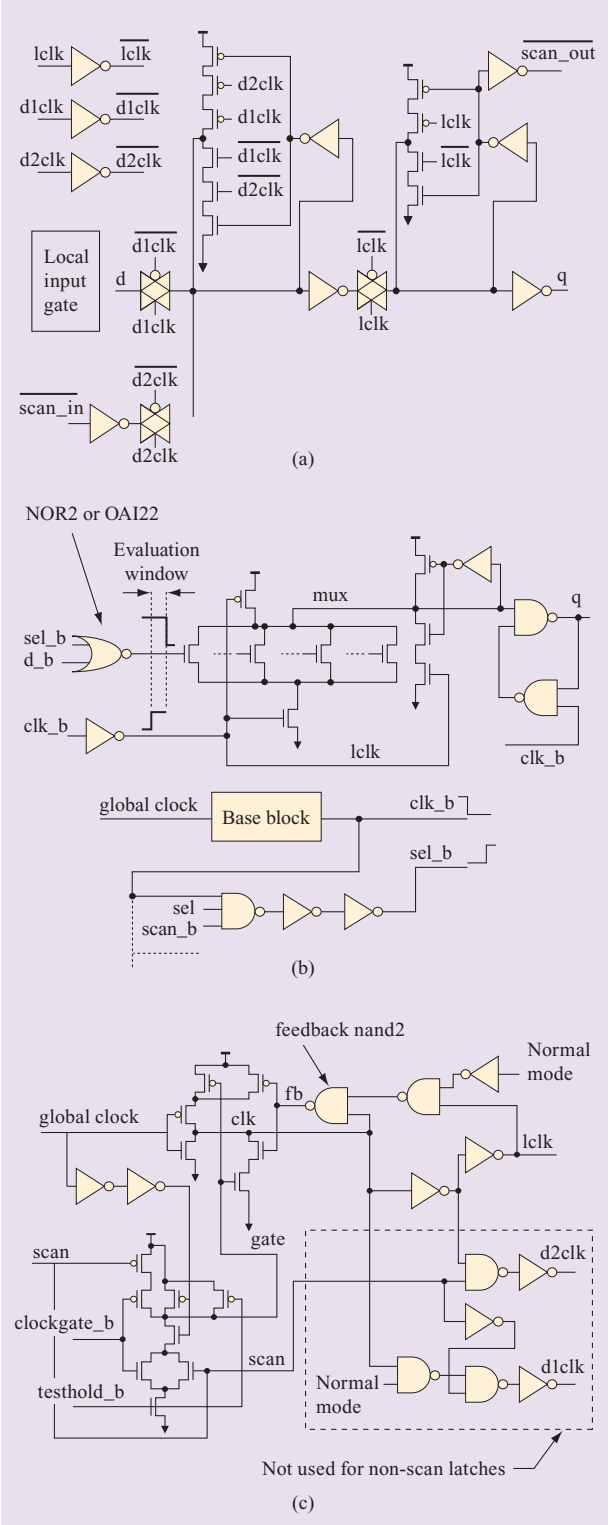

Figure 5

(a) Standard master–slave flip-flop. (b) High-performance latch and clock details. (©2005 IEEE. Reprinted, with permission, from [1].) (c) Clocking for pulsed-mode operation of flip-flop.

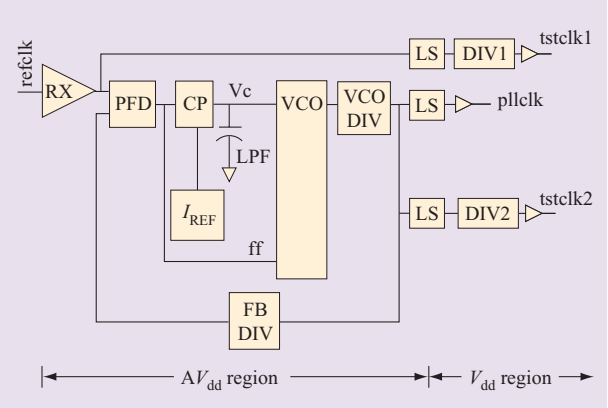

Figure 6

Phase-locked loop (PLL) block diagram.

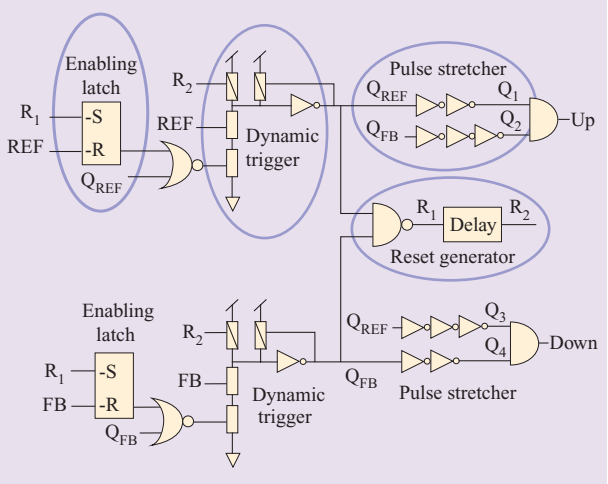

Figure 7

Phase–frequency detector.

false output, the AND node is discharged low and the strobe does not trigger the OR plane input. The OR function is provided by a dynamic NOR circuit followed by a set–reset latch to capture the output of the OR function on a clock boundary. The latches that are designed into the DPLA are also scannable.

Core PLL

Cell/B.E. clock generation logic uses the output of a PLL to generate the core clock. **Figure 6** shows a block diagram of the PLL. The PLL contains a differential receiver, a phase–frequency detector (PFD), a charge pump (CP), a current reference (I_{REF}), a filter capacitor/low-pass filter (LPF), a voltage-controlled oscillator (VCO), as well as frequency dividers in the forward (VCODIV) and feedback (FBDIV) paths for frequency scaling capabilities. The PLL power is supplied by a separate analog supply (AV_{dd}), while the logic circuits and PLL-to-logic level shifters are powered by the regular chip supply (V_{dd}).

Previous self-resetting CMOS PFD architectures [6] have effectively eliminated dead zones by using a combination of simple static and dynamic circuits. The Cell/B.E. processor PFD design uses similar concepts but has enhanced resetting techniques for extending the performance to higher frequencies. **Figure 7** shows the PFD topology consisting of symmetric paths containing enabling latches, dynamic triggers, and pulse stretchers for the reference (REF) clock and feedback (FB) clock paths, and a common reset generator path. When the REF signal arrives before FB, the dynamic trigger for REF responds to the rising edge of REF, causing Q_{REF} to rise and issuing a fast localized disable of the trigger. The Up output is then asserted after a delay δ_1 created by the two inverters. A subsequent rising transition of FB similarly causes the FB trigger to respond and issue its own local disable. After delay δ_2 of the three inverters, the Up output is de-asserted, and this output is stretched by an amount $\delta_2 - \delta_1$. The reset generator signal R_1 now forces a reset to both enabling latches in preparation for the reset of the triggers. After some delay, the reset generator signal R_2 precharges both dynamic triggers, which in turn force R_1 and R_2 high, completing the reset process. Finally, when REF and FB falling transitions occur, the triggers are enabled in preparation for the next cycle.

To reduce the large transient turn-on currents that can occur in the charge pump, a conventional charge pump circuit topology has complementary current paths added to it, which prevents internal switching nodes from drifting toward the supply rails during the high-impedance Off state [7]. Matched replica devices are also added to the charge pump and are connected to opposite polarities of the control signals from the PFD to counteract the error introduced by parasitic gate-source and gate-drain capacitances during switching. Direct feed-forwarding of the PFD signals into the VCO was used to avoid the problems typically associated with integrated PLL resistors, such as high capacitance and lack of adjustability. The VCO [8] consists of a ring of five delay elements with interleaved control/feed-forward interconnections, in which each delay element consists of a main ring inverter, a control section, and a feed-forward section. The VCO mostly uses two levels of device stacking, contributing to large signal swings, high operating frequency, and low-voltage operation. Simulated VCO gain for the interleaved VCO design is 9.4 GHz/V.

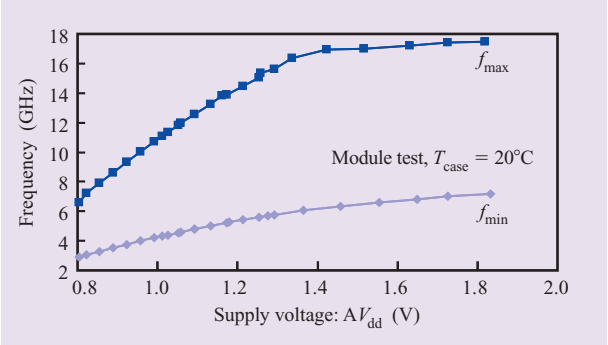

Figure 8

PLL lock range vs. analog supply voltage.

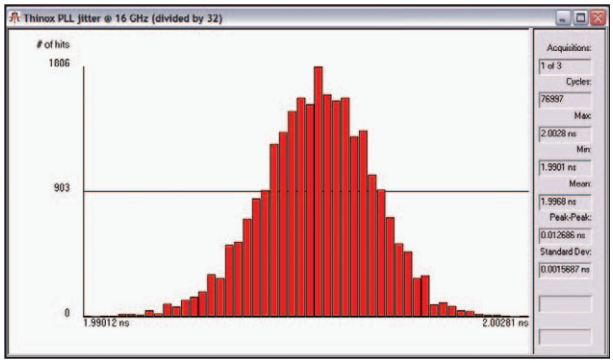

Figure 9

PLL jitter histogram at 16-GHz cycle–cycle; jitter = 12.7 ps peak to peak and 1.57 ps rms.

Body-contacted devices were used in areas of the PLL where the history effect [9] caused by SOI may be a concern, although this usually resulted in a larger area and increased parasitic capacitance. Body contacts were also used in the analog circuits to improve threshold voltage (V_t) matching and reduce current variation with drain-source voltage (V_{ds}).

Figure 8 shows the lock range as a function of AV_{dd} for the PLL, where f_{max} and f_{min} represent the maximum and minimum lock frequencies, respectively. The measured f_{max} increases roughly linearly with frequency from 6.6 GHz at 0.804 V to 16.4 GHz at 1.337 V and maintains an average f_{max}/f_{min} ratio of 2.6 over this range. Above 1.3 V, the slope of f_{max} decreases significantly, and a maximum lock frequency of 17.5 GHz was measured at 1.819 V for this part. An operational frequency of 10.0 GHz was achieved at a supply voltage of less than 0.960 V. Maximum lock frequency measured on early hardware was 18.560 GHz.

Cycle-to-cycle (C–C) jitter measurements were taken using a proprietary time-interval analysis tool. **Figure 9** shows the C–C jitter histogram taken at 1.2 V for the PLL output divided by 32 with jitter of 12.7 ps peak to peak (P–P), 1.57 ps root mean square (rms).

Circuit checks and design analysis

To ensure design integrity, a rigorous system of design checks was implemented, along with a detailed design analysis methodology. **Figure 10** shows a block diagram of the circuit design check methodology. Digital macros (RLMs and custom macros) had to pass a unified set of electrical and topological checks that applied a consistent set of margins and specifications to the design. These checks focused on such items as local clock integrity (clock component loading specifications, clock RC, clock line hostile coupling capacitance, local clock buffer connections and circuit topology), latch and flip-flop

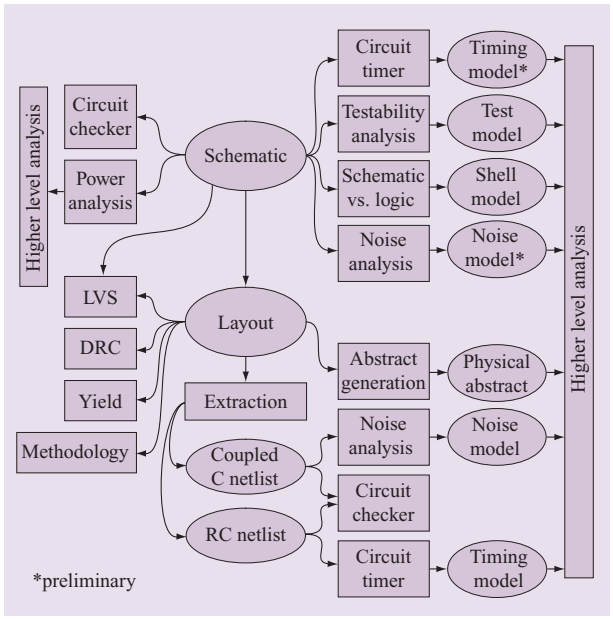

Figure 10

Circuit design verification and checking. (LVS: layout vs. schematic; DRC: design rule checking.)

usage (input gate type, pull-up and pull-down strengths, input gate proximity to flop, proper clock connections), dynamic circuit usage (topology checks, keeper ratios, output stage beta ratio), and rules for static CMOS and transmission gates (maximum stack heights, beta ratio limits, topology checks, transmission gate width, depth, and driving gate rules). All designs also had to pass electromigration (EM) and current-resistance (I_R) drop checks, ensuring a robust power supply to all components.

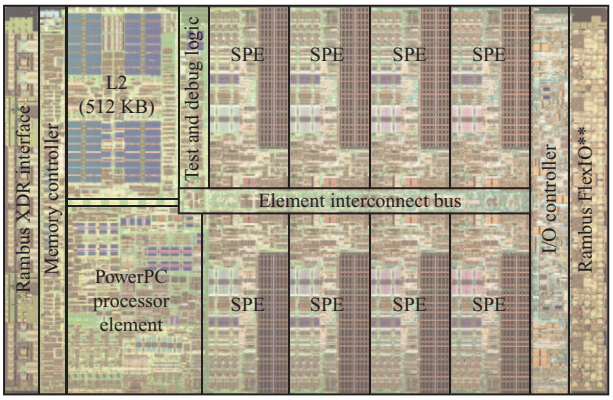

Figure 11

Die photograph of a 90-nm Cell/B.E. processor. (©2005 IEEE. Reprinted, with permission, from [1].)

In general, because more aggressive design styles were used, the number and difficulty of the checks increased. For example, clock component loading specifications for pulsed latches were tighter than those for the flip-flops, and several post-layout iterations were usually required to get all components powered correctly. Dynamic circuits have to meet strict precharge/evaluate overlap-avoidance timing rules. This also requires post-layout design iteration and simulation with several specified process, voltage, and temperature (PVT) conditions in order to guarantee robust functionality. These requirements affected the productivity of the designers using these circuit styles but were necessary for robust functionality across a wide PVT window.

In addition to electrical design checks, various physical design, design rule, and layout-versus-schematic (LVS) checks were carried out. Certain "best practices" rules for the layout were checked, with the goal of enhancing overall yield of the chip. For these rules, 100% compliance was not required, but designs were graded against each rule to ensure that the physical implementation followed the defined best practices whenever possible, without affecting the area of the design. Physical methodology rules were also enforced to ensure easy integration of the macro into the next level of the chip hierarchy.

Transistor-level timing, noise, and power analyses were carried out on all macros, using identical flows for custom macros and RLMs. Timing and noise analyses were complicated by the hazards associated with the floating body potential in silicon-on-insulator (SOI) technology. Analysis tools were set to use either the lowest (slowest) body voltage settings or the highest (fastest) settings depending on the analysis being performed. Given the prevalence of pulsed latches, high-performance mux latches, and delayed-clock flip-flops, the timing analysis for race conditions was particularly important. Each check was required to have a margin proportional to the total delay of the clock from the point of divergence to the checkpoint, plus a fixed nonscaling amount. This methodology ensured that margins always scaled in a manner that was proportional to the amount of clock delay inserted. Furthermore, since the same basic timing engine was used for macro timing (transistor level analysis) as that used for chip timing (block-level analysis), all margins and timing checks were carried forward in a consistent manner from the macro up to the chip-level timing runs.

Technology

The Cell/B.E. processor has been implemented in the IBM 90-nm CMOS SOI process and 65-nm CMOS SOI process. Although the function observed by the programming environment is the same, the physical design is different. Moving from 90 nm to 65 nm was not a simple remap exercise. The 65-nm process presented new challenges that had to be solved. Some of the problems encountered include scaling mismatches between analog, I/O, and array devices. The back-end thicknesses of line metal stacks are different. For 90 nm, the metal stack uses one local interconnect layer, five 1X layers, and two 6X layers.[3] By contrast, the 65-nm metal stack eliminates the local interconnect layer and provides four 1X layers, four 2X layers, and two 8X layers. For 65 nm, manufacturing requires the design to align polysilicon unidirectionally for yield improvements. To compensate for the memory cell size reduction in the 65 nm, an additional power supply (V_{dd}CS) was used for SRAM arrays.

Physical integration

90-nm design

Figure 11 shows a die photograph of the 90-nm Cell/B.E. chip. The chip has approximately 241M transistors. There are 17 major physical partitions and 8,912 discrete chip-level floor-planned blocks (e.g., PLL, I/O, thermal sensors, engineered bus transport). The total chip consists of 177K floor-planned blocks, 580K repeaters, and 1.55M nets. These statistics cover all unit, partition, and global signal components. As shown in Figure 11, at the center of the chip is the EIB. This is an engineered bus that is composed of four 128-bit data rings plus a 64-bit tag operated at half of the core clock frequency. The wires are arranged in groups of four, interleaved with ground (GND) and supply voltage V_{dd}. The shields are twisted at the center to reduce coupling

[3] 1X is the minimum layer thickness; 6X is six times the minimum layer thickness.

noise on the two unshielded wires. To ensure signal integrity, many of the nets were custom tailored, and more than 50% of the 57K chip-level nets were engineered with 32K repeaters. The Cell/B.E. chip uses 3,349 controlled collapse chip connections (C4s) in a 90-column by 61-row varied pitch matrix with five regions of different row–column pitches attached to the low-cost organic package described earlier. This structure supports 20 separate power domains on-chip, many of which overlap physically on the die. From processor elements to power and clock distributions, global routing, and the chip assembly techniques, the entire chip hierarchy is engineered to support a modular design in a building-block–like construction.

65-nm design

The 65-nm Cell/B.E. processor design required floorplan adjustments to optimize area usage. **Figure 12** shows a die photo of the 65-nm Cell/B.E. chip. The core PLL was moved from the left side of the chip to the right side of the chip. This move required redesign of the core clock distribution network. eFUSE macros were also relocated as a result of floorplan shifts.

The 65-nm design contained roughly 239M devices. The physical entities did not change from the 90-nm Cell/B.E. processor and remained at 17. The EIB wires continued to be arranged in groups of four, but no shielding or interleaving was required. The C4 count increased to 4,502. There are three regions of different row–column pitches for the C4s.

Noise analysis

Given the aggressive frequency targets and the circuit design techniques described in the previous sections, it was apparent that a detailed chip noise analysis strategy would be required to ensure stable operation of the chip. With the chip device counts described above, it is impractical to analyze the whole chip using a flat transistor-level simulator. Thus, in keeping with the hierarchical/modular design philosophy, a hierarchical noise analysis approach was applied, which relied on a macro-level static noise analysis [10], followed by unit-/chip-level noise analysis.

Macro-level static noise analysis was carried out on all designs, including static/dynamic circuits, arrays, and synthesized RLMs, creating a noise abstract at the same time. The analysis was performed by a transistor-level simulator, running on a net list with parasitic capacitance extracted from the layout. This noise tool carried out a static noise analysis on each channel-connected component, looking for noise failures throughout the design. A macro-level noise abstract was generated during the analysis, which included input noise tolerance level, input capacitance, and output resistance. Unit- or chip-level noise analysis was then performed using the macro noise abstracts and information from timing analyses. Macros were represented by their abstracts, which specified the input capacitance and output resistance. Extracted parasitic parameters are obtained from the global layout. Unit- or chip-level noise was then analyzed with an equivalent circuit composed of resistors, capacitors, and voltage sources that are driven according to the timing information. A noise failure was reported if the noise level on a net exceeded the input tolerance level of the receiver pin, which was determined during the macro-level noise analysis.

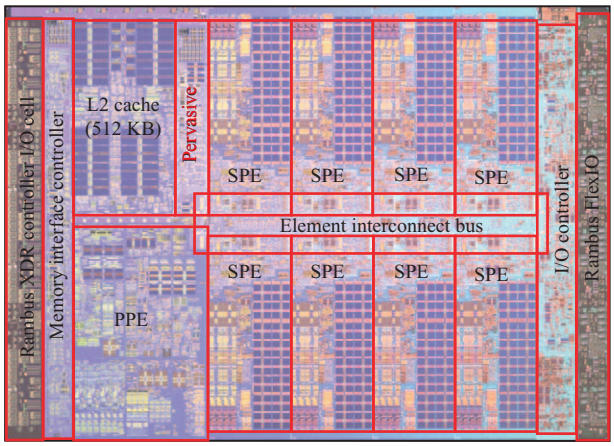

Figure 12

Die photograph of a 65-nm Cell/B.E. processor. (©2005 IEEE. Reprinted, with permission, from [1].)

Power analysis

To calculate the total power, input switching factors (SFs) and clock activity (CA) for each macro are monitored in a register transfer level (RTL) simulation for a given workload. For every macro instance, the input SF is calculated by observing the percentage of inputs that have changed state from the previous cycle. The CA is measured by observing the number of local clock buffers that are turned on in a given cycle. The ability to compute cycle-by-cycle power for each macro for a realistic workload makes this methodology very useful for package analysis. Since power is dependent on both SF and CA, these factors needed to be calculated for every cycle of the simulation and cannot be simply averaged over the entire simulation. In a given cycle, the total power is calculated as

$$\text{Total Power}(C) = \sum \text{Block Power}(SF, CA) + 1/2\, C_{\text{net}} V^2 f, \quad (1)$$

where C_{net} is the amount of global net capacitance switched.

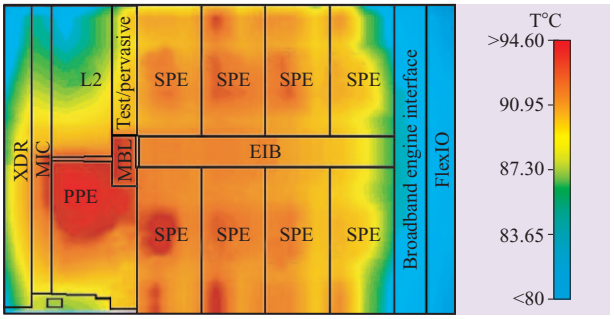

Figure 13

Die thermal map of the 90-nm Cell/B.E. processor. SPE: synergistic processor element; EIB: element interconnect bus; MIC: memory interface controller; PPE: PowerPC processor element. (©2005 IEEE. Reprinted, with permission, from [1].)

To estimate the power consumption of the Cell/B.E. processor and for refining and verifying the power management logic of the chip, each core or functional unit on the chip was required to run at least three types of workloads: idle, typical, and high power. The idle workload was expected to be the lowest power state, since it shuts off as many local clock buffers as possible. The high-power test case was used as a stimulus for power grid integrity analysis, thermal analysis, and as an input for the IR drop analysis.

Thermal Analysis

Due to local heating caused by independent operation of individual processing units, sophisticated local thermal sensing strategies and thermal control mechanisms were required to allow aggressively low-cost and quiet thermal solutions. The Cell/B.E. processor presented new challenges in chip thermal design. Under high heat flux conditions, the silicon substrate behaves like a relatively low-thermal conductance body. Given the definition of Bi (effective Biot number) as defined by Equation (2), it can be determined under which conditions small hot spots would create local heat spikes on top of the die global temperature distribution:

$$Bi = \frac{A_p}{A_s}\frac{b\,h}{\lambda}, \tag{2}$$

where A_p is substrate area, A_s is the source area, h is the effective heat transfer coefficient at the opposite side of silicon substrate, b is the substrate thickness, and λ is the thermal conductivity of silicon.

Then, peak temperature excesses can be estimated by analytic modeling techniques based on classical steady-state thermal diffusion theory [11], as well as on transient analysis [12]. These models approximate a rectangular source, with the general formula described in Equation (3):

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial\theta}{\partial r}\right)+\frac{\partial^2\theta}{\partial z^2}+g = \frac{1}{\alpha}\frac{\partial\theta}{\partial t}, \tag{3}$$

where g is the energy generation rate, r the radius, t the time, z the height of the cylinder, α the thermal diffusion of silicon, and θ the temperature.

A 100-micron-range length scale was found to be a high enough resolution for hot-spot analysis; thus, it was not necessary to consider any quantum effects [13] in this analysis. The hot-spot information was generated by the power maps constructed from the power analysis based on various workloads described earlier. In addition, external thermal spreading in the lateral direction, through the package, affects the die temperature global profile, causing a steeper slope from the center out toward the edges/corners.

Because of the high degree of complexity, extensive numerical thermal analysis was required. This analysis was carried out early in the design cycle to ensure that the maximum junction temperature and the mean die temperature, including transient behavior, would be within the design specifications. These data were analyzed to improve the design and floorplan of the chip and provided feedback for improved thermal sensor design and placement, following the thermal map shown in **Figure 13**. These sensors were then used by the PMU and TMU as described earlier, embodying an effective, low-cost power and thermal management control system.

Summary

The Cell/B.E. processor was designed to bring a new level of computing capability to real-time applications. The design challenge for the Cell/B.E. processor was to manage the triple constraint of power, performance, and area on a very aggressive schedule. The design builds on the PowerPC Architecture and introduces the SPE. The Cell/B.E. contains nine processor cores that can have ten parallel threads active at any one time. The design and methodology rely on hierarchy to divide and conquer the challenging task. Both 90-nm and 65-nm versions of the design have been completed.

Acknowledgments

The design and implementation of the Cell/B.E. processor was a monumental effort. The authors thank all of the members of the Sony–Toshiba–IBM (STI) Design Center and the extended teams for their dedication and contributions to the design and implementation of the Cell/B.E. processor.

**Trademark, service mark, or registered trademark of Rambus, Inc., in the United States, other countries, or both.

†Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc., in the United States, other countries, or both.

References

1. D. C. Pham, T. Aipperspach, D. Boerstler, M. Bolliger, R. T. Chaudhry, D. Cox, P. Harvey, et al., "The Design and Implementation of a First-Generation Cell Processor," *ISSCC Digest of Technical Papers*, February 2005, pp. 184–185.
2. B. Flachs, S. Asano, S. H. Dhong, H. P. Hofstee, G. Gervais, R. Kim, T. Le, et al., "The Microarchitecture of the Streaming Processor for a Cell Processor," *ISSCC Digest of Technical Papers*, February 2005, pp. 134–135.
3. T. Asano, S. H. Dhong, O. Takahashi, M. White, T. Nakazato, J. Silberman, A. Kawasumi, and H. Yoshihara, "A 4.8GHz Fully Pipelined Embedded SRAM in the Streaming Processor of a Cell Processor," *ISSCC Digest of Technical Papers*, February 2005, pp. 486–487.
4. "RISCWatch Debugger User's Manual," Seventeenth Edition, IBM Corporation, January 2007, *http://www.ibm.com/chips/techlib/techlib.nsf/techdocs/8A70ED3C8215AC5E872569D90050295E*.
5. P. J. Restle, C. A. Carter, J. P. Eckhardt, B. L. Krauter, B. D. McCredie, K. A. Jenkins, A. J. Weger, and A. V. Mule, "The Clock Distribution of the Power4 Microprocessor," *IEEE International Solid-State Circuits Conference Digest of Technical Papers*, 2002, pp. 144–145.
6. D. W. Boerstler and K. A. Jenkins, "A Phase-Locked Loop Clock Generator for a 1 GHz Microprocessor," Symposium on VLSI Circuits (VLSI 1998), *Digest of Technical Papers*, June 11–13, 1998, pp. 212–213.
7. D. Boerstler, K. Miki, E. Hailu, H. Kihara, E. Lukes, J. Peter, S. Pettengill, J. Qi, J. Strom, and M. Yoshida, "A 10+ GHz Low Jitter Wide Band PLL in 90 nm PD SOI CMOS Technology," Symposium on VLSI Circuits, *Digest of Technical Papers*, June 17–19, 2004, pp. 228–231.
8. D. W. Boerstler, "Interleaved VCO with Balanced Feedforward," U.S. Patent No. 6,744,326, June 1, 2004.
9. S. K. H. Fung, N. Zamdmer, P. J. Oldiges, J. Sleight, A. Mocuta, M. Sherony, and S. H. Lo, et al., "Controlling Floating-Body Effects for 0.13 μm and 0.10 μm SOI CMOS," *Technical Digest of International Electron Devices Meeting* (IEDM 2000), December 10–13, 2000, pp. 231–234.
10. K. L. Shepard, V. Narayannan, and R. Rose, "Harmony: Static Noise Analysis of Deep Submicron Digital Integrated Circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **18**, No. 8, August 1999, pp. 1132–1150.
11. S. Lee, S. S. Van Au, and K. P. Moran, "Constriction/Spreading Resistance Model for Electronics Packaging," *Proceedings of the ASME/JSME Thermal Engineering Conference*, **4**, 1995, pp. 199–206.
12. K. Yazawa and M. Ishizuka, "Thermal Modeling with Transfer Function for the Transient Chip-On-Substrate Problem," *Thermal Sci. Eng.* **13**, No. 1, pp. 37–40 (2005).
13. K. E. Goodson, Y. S. Ju, and M. Asheghi, "Thermal Phenomenon in Semiconductor Devices and Interconnects," Chapter 7, *Miroscale Energy Transport*, C. L. Tien, A. Majumdar, and F. M. Gerner, Editors, Taylor & Francis, New York, 1998, pp. 229–294.

Received February 2, 2007; accepted for publication May 2, 2007; Internet publication August 15, 2007

Mack W. Riley *IBM Systems and Technology Group, 11501 Burnet Road, Austin, Texas 78758 (mwriley@us.ibm.com).*

Mr. Riley is an IBM Distinguished Engineer. He received an M.S. degree in electrical engineering from Stanford University in California, and a B.S. degree in electrical engineering from Tuskegee University in Alabama. In 1981 Mr. Riley joined IBM as a member of the design and development team for the 5520 Administrative System working on the design of graphics and hard file interfaces to the central processor. Throughout his career, he has been involved with the design and development of systems such as the PC RT and RISC System/6000* workstations that were developed in Austin. Mr. Riley was Chief Engineer for the Austin graphics development team and was involved in the development of low and mid-range 3D graphics subsystems. He has also been a member of the WorldWide Design center as a project manager. Currently Mr. Riley is Chief Engineer for the Sony, Toshiba, and IBM Design Center in Austin, Texas.

James D. Warnock *IBM Systems and Technology Group, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598 (jwarnock@us.ibm.com).*

Dr. Warnock received a B.Sc. degree in Physics from Ottawa University in Ottawa, Ontario, Canada, in 1980, and a Ph.D. degree in physics from the Massachusetts Institute of Technology, Cambridge, Massachusetts, in 1985. Since then, he has been at IBM in Yorktown Heights, New York, initially studying advanced bipolar, complementary bipolar/BiCMOS, and CMOS silicon technologies. Later he was involved with work in the area of circuit design for high-performance digital microprocessors, including the S/390* G4 processor and the POWER4* chip, where he was the circuit design team leader. More recently, he was part of the circuit leadership team for the Cell Broadband Engine. Dr. Warnock is currently working on several microprocessor development programs within IBM, where he is involved with aspects of circuit design tools and methodology, circuit and physical design implementation, local clocking/latch design, and design–technology interactions. He has experience in process technology and device design, as well as in SOI digital circuit design, and has authored or co-authored more than 170 conference or journal papers. He is an IBM Distinguished Engineer and a member of the IBM Academy of Technology.

Dieter F. Wendel *IBM Systems and Technology Group, Boeblingen Development Laboratory, Schoenaicher Strasse 220, 71032 Boeblingen, Germany (WENDEL@de.ibm.com).*

Mr. Wendel is an IBM Distinguished Engineer who joined IBM in 1981 after receiving a B.S. degree in electrical engineering from the University of Wuerzburg, Germany. That same year he worked on the very large scale integration fellowship team in Boeblingen. After a two-year assignment from 1984 to 1986 at the IBM Research Laboratory in Yorktown Heights, New York, he joined the S/390 microprocessor development organization at the IBM Boeblingen Laboratory to work in several areas including test, array design, and custom logic design. Mr. Wendel joined the STI Design Center in Austin, Texas, when it was founded in 2001 as circuit lead. His current interests focus on concepts in high-frequency circuit design and the exploitation of new technologies.