Power-Conscious Design of the Cell Processor's Synergistic Processor Element

THE AUTHORS DESCRIBE THE LOW-POWER DESIGN OF THE SYNERGISTIC PROCESSOR ELEMENT (SPE) OF THE CELL PROCESSOR DEVELOPED BY SONY, TOSHIBA AND IBM. CMOS STATIC GATES IMPLEMENT MOST OF THE LOGIC, AND DYNAMIC CIRCUITS ARE USED IN CRITICAL AREAS. TIGHT COUPLING OF THE INSTRUCTION SET ARCHITECTURE, MICROARCHITECTURE, AND PHYSICAL IMPLEMENTATION ACHIEVES A COMPACT, POWER-EFFICIENT DESIGN.

Osamu Takahashi
IBM Systems and Technology Group

Scott Cottier
Sang H. Dhong
Brian Flachs
Joel Silberman
IBM T.J. Watson Research Center

•••••• The Cell processor has been developed through the partnership of three companies: SONY Computer Entertainment, Toshiba, and IBM. The high-level design phase started from the late fall of the year 2000. The design team spent most of the year 2001 designing a test chip as a design concept verification vehicle. After completing the test chip design, the product development started. The objective of the project is to develop a processor design highly tuned for media processing and the expected demands for more complex and larger data handling. The possible applications of such a processor may include image processing for high definition TV, image processing for medical usages, high performance computing, and gaming. These are just a few of the possibilities. The lists of possible applications may be close to endless.

The Cell processor is tailored for media processing yet is flexible enough to be a general-purpose microprocessor that supports high-level-language programming.[1] Figure 1 shows a die photo of the Cell processor. It contains a 64-bit power core and eight Synergistic Processor Elements (SPEs). The SPE implements a brand new architecture.[2-3] This article describes the low-power design of the dual-issue, 32-bit, 4-way SIMD SPE.

We designed the SPE with a target 11 fan-out of 4 (FO4) cycle time, allowing high-frequency operation. As Figure 2 shows, the SPE has two clock domains: one with an 11 FO4 cycle time and the other with a 22 FO4 cycle time. We implemented the SPE mainly using custom design, especially in the high-frequency domain. The SPE contains 256 Kbytes of dedicated local store memory (four 64-Kbyte SRAM arrays[4]). The 128-bit, 128-entry general-purpose register file,

Published by the IEEE Computer Society

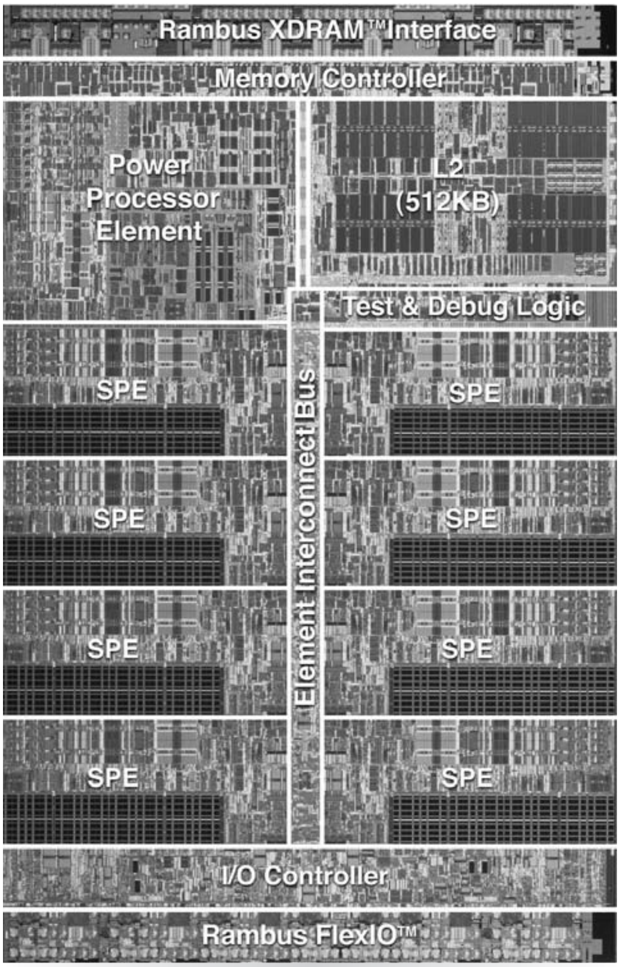


Figure 1. Die photo of the Cell processor.

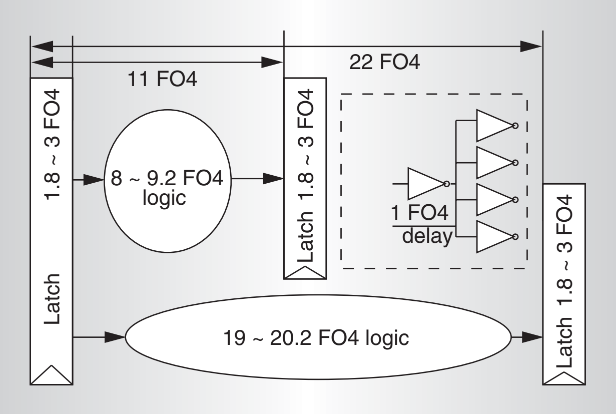


Figure 2. Timing diagram of the 11 FO4/22 FO4 SPE design.

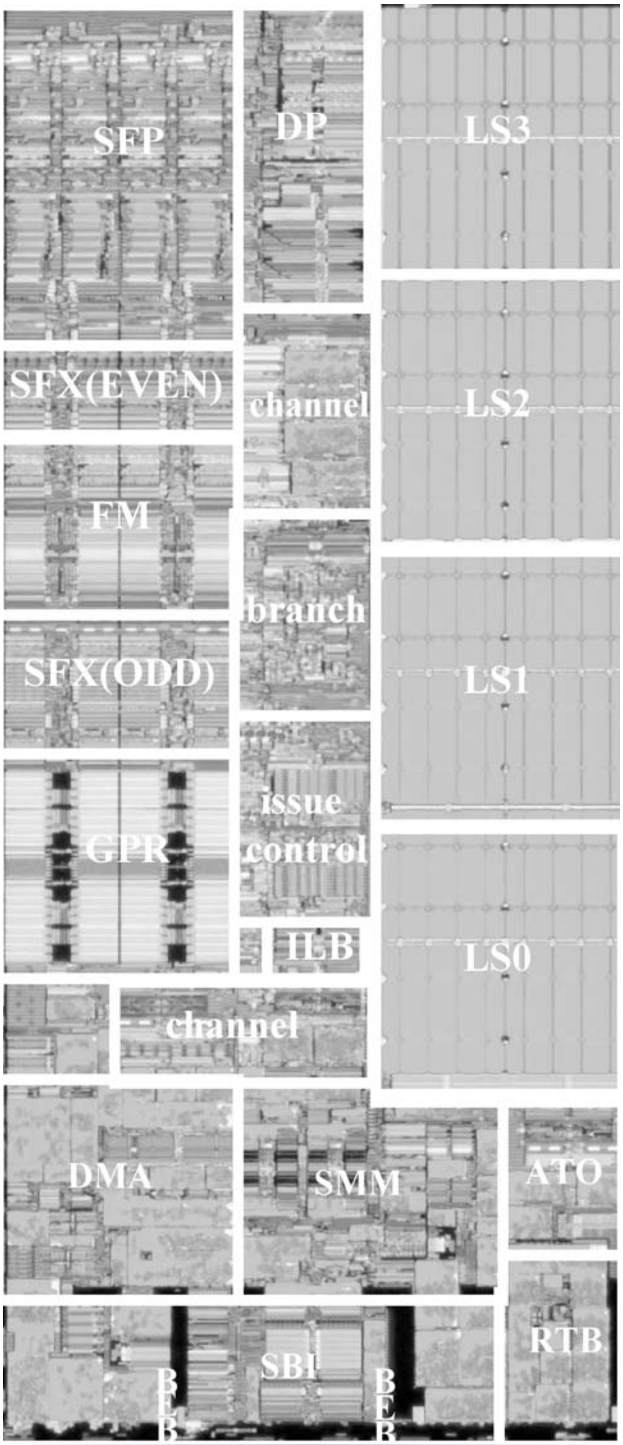


Figure 3. Die photo of the synergistic processing element (SPE).

with six read ports and two write ports, is implemented as two 64-bit, 128-entry macros, four 32-bit fixed-point units,[5] four single-precision floating-point units,[6] and one double-precision floating-point unit.

Figure 3 illustrates the SPE and memory flow control (SMF) floorplans. The SMF operates at half the SPE's frequency. The SPE

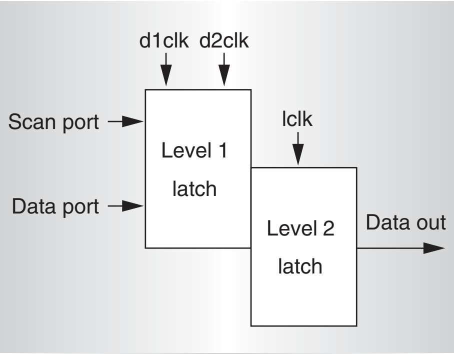


Figure 4. Transmission gate latch.

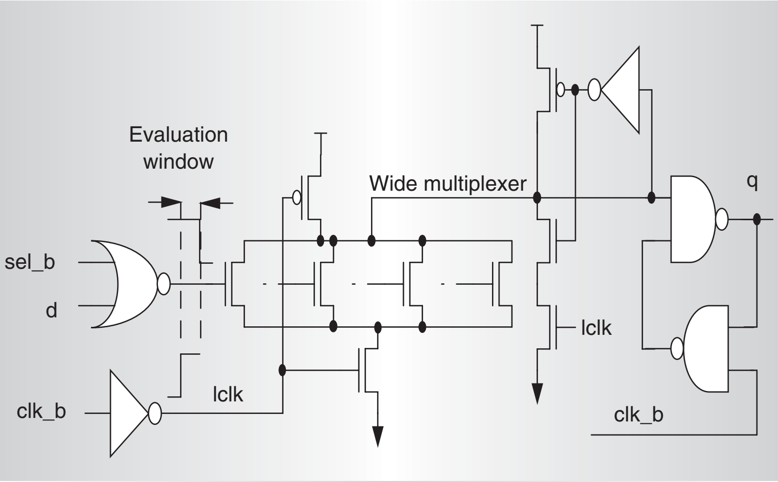


Figure 5. Dynamic multiplexer latch.

design contains roughly 20.9 million transistors. The SPE chip area including the SMF is 14.8 square mm (2.54 mm × 5.81 mm), fabricated in 90-nm silicon-on-insulator (SOI) technology in a 300-mm line. The technology offers 46-nm Lpoly, 1.05-nm Tox for digital circuits, 2.2-nm Tox for analog and I/O circuits, dual-threshold voltages, 1.0-V nominal supply voltage, eight levels of copper layers plus one local interconnect, and low-k dielectric. The SPE implements extensive fine-grained clock gating to ensure that only necessary logic is activated. In paths with enough timing slack, we changed transistors from regular to high-threshold-voltage devices. We used full CMOS static gates and transmission gates to implement most of the logic. For timing-critical areas, occupying 19 percent of the SPE's non-SRAM area, we used dynamic circuits selectively.

We have tested the SPE at various temperatures, supply voltages, and operating frequencies and have observed correct operation of up to 5.6 GHz at a 1.4 V supply and 56° C. The SPE's measured power consumption is in the range of 1 W to 11 W, depending on operating clock frequency, temperature, and workload.

Triple design constraints

Because the Cell contains eight copies of the SPE, optimization of the SPE's power and area is critical to the overall chip design. We made a conscious effort to reduce SPE area and power while meeting the 11 FO4 cycle time performance objectives. We optimized the design to balance the three constraints of power, area, and performance, considering many tradeoffs to achieve the overall best results. We particularly considered five items for reducing power consumption: appropriate latch selection, a fine-grained clock-gating scheme, a multiclock-domain design, use of dual-threshold voltage, and selective use of dynamic circuits.

Latch selection

As Figure 2 shows, the available time for implementing logic in the 11 FO4 design is 8 to 9 FO4. The remaining cycle time is consumed by latch insertion delays. We used several latch types with various insertion delays.[1] We selected latch types by trading off among the triple constraints in a particular area.

Figure 4 shows the SPE's main workhorse latches, which are transmission gates combining level 1 and level 2 latches. They come in two varieties: scannable and nonscannable. Each has several power levels. We use these latches throughout the SPE. The second latch type is the pulsed clock latch, which is basically nonscannable. It has a small insertion delay and a small area with relatively low power consumption, and we use it in the most timing- and area-critical locations.

The third type is the dynamic multiplexer latch, shown in Figure 5. This latch is scannable and has multiplexing widths from 4 to 10, including a scan port. When "sel_b" in Figure 5 is asserted, the corresponding data is selected and latched in the cross-coupled NAND. In addition to multiplexing capability, it has a small insertion delay. We use it in locations where timing is critical and the microarchitecture requires multiplexing. A typical use of this type of latch is the SPE's dataflow operand latches. The dynamic multiplexer latch includes a static OR or and-or-invert (AOI) gate as a data input gate. In dataflow operand latches, data signals travel long distances over very active dataflow regions. The inclusion of a static gate helps to increase noise immunity.

The fourth type of latch is a dynamic programmable-logic-array latch.[3] This is a scannable latch used mainly to generate control signals, especially clock-gating signals. The last two types of latches consume slightly higher power, but they can complete complex tasks with a given insertion delay. Without using the last two latch types, we couldn't have realized the SPE's pipeline stages with the 90nm SOI technology. This is an example of a tradeoff among the triple constraints. We made every effort to use the transmission gate latches or low-power pulsed-clock latches and used the other tw o types only when absolutely necessary. This is the main contributor to the SPE's power-conscious design.

Fine-grained clock gating

Clock gating is an effective method of reducing active power.[7,8] We used it extensively in the SPE design. Figure 6 shows the design of the local clock buffer, which supplies a local clock signal to a bank of latches, resulting in a fine-grained clock-gating scheme. The LCB fires a local clock when the test hold signal (testhold_b) is not asserted and either scan enable (scan) or clock activation (clockgate_b) is asserted. If that condition is met, the global mesh clock is propagated through the circuitry and buffered to drive a bank of latches. The LCB's feedback behaves as a latch to keep the LCB in the same state until the condition changes.

The LCB version shown in Figure 6 drives scannable transmission gate latches (Figure 4). It generates d1clk for the data port (second half-cycle clock), d2clk for the scan port (second half-cycle clock), and lclk for the level 2 latch (first half-cycle clock). The base block design is common for LCBs that drive all four latch types. The LCB's driver sections vary according to the requirements of downstream latches. We minimized the setup time for an LCB activation signal (clockgate_b) so that the activation signal is generated within one cycle before the gated cycle. This leads to using the fewest registers necessary for generation of LCB activation signals and the least AC power for clock-gating control logic. Figure 7 shows the clock-gating control generation timing diagram.

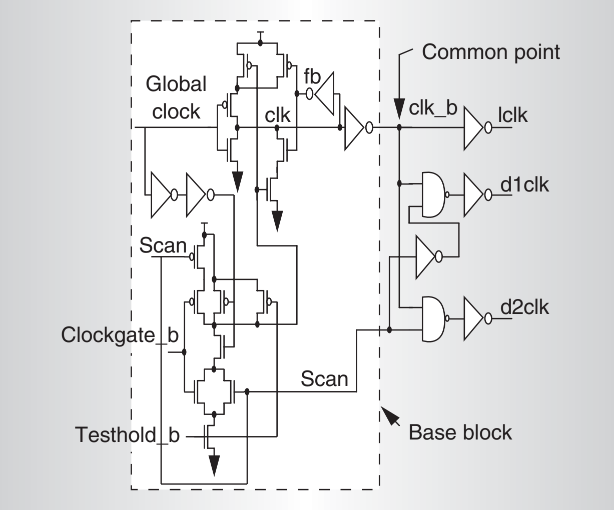


Figure 6. Local clock buffer (LCB).

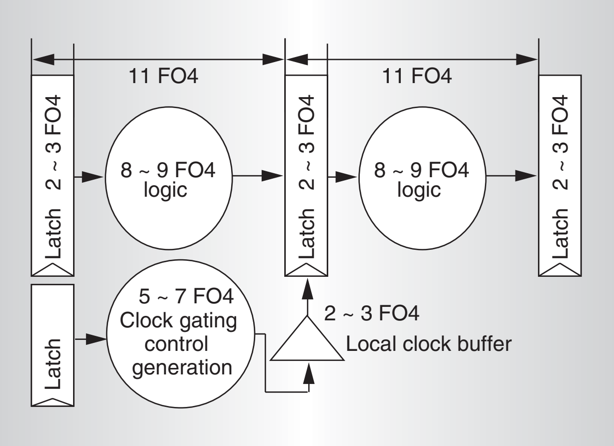


Figure 7. Clock-gating control generation timing diagram.

In our SPE microarchitecture implementation, the registers are normally turned off and are turned on only when necessary through the fine-grained clock-gating scheme. In other words, the SPE activates only necessary pipeline stages. Figure 8 is the SPE pipeline diagram, showing how flush and fetch interact with other instruction processing.[2] The activated stages ripple to the next stages as the pipeline proceeds. The SPE design is capable of dual issue. Its load/store,

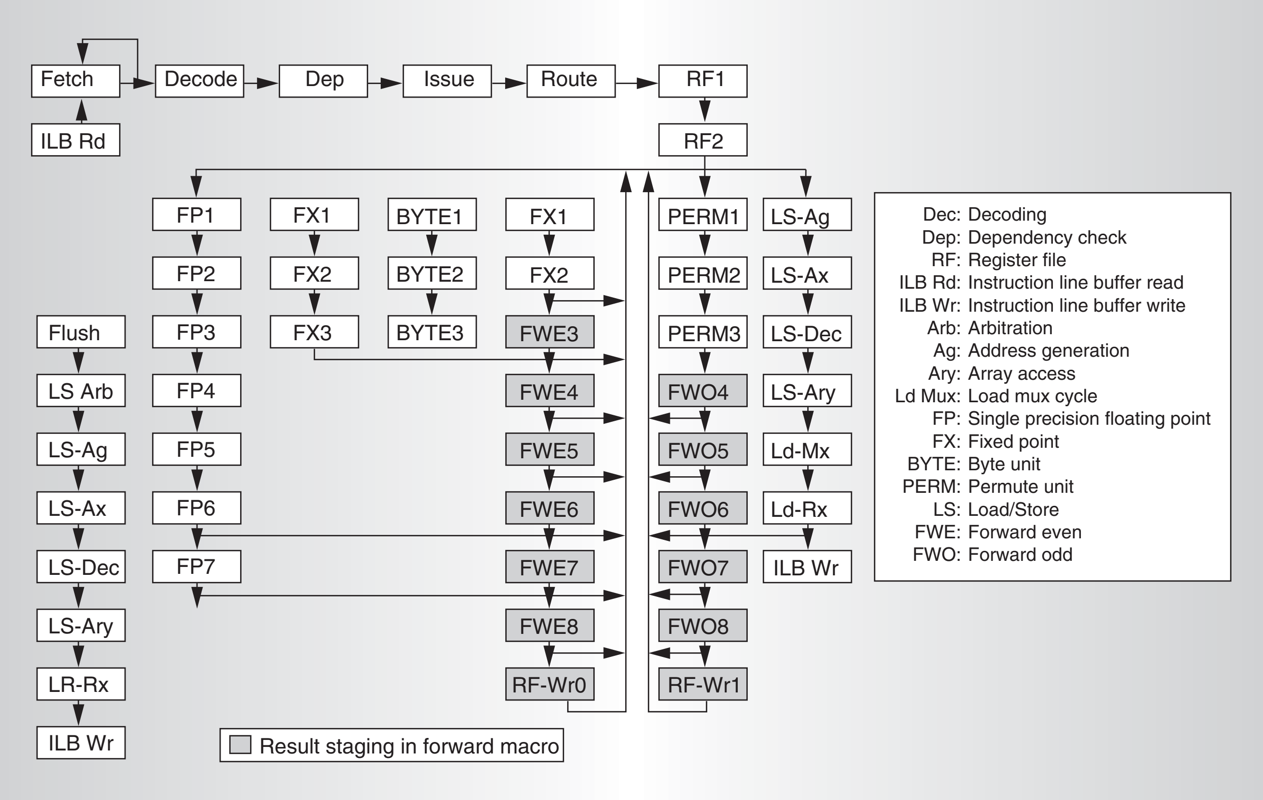


Figure 8. SPE pipeline diagram.

permute, 2-cycle latency fixed-point, 3-cycle latency fixed-point, byte, and single-precision floating-point operations are fully pipelined. Thus, two corresponding pipe stages in parallel execution units can be active in a given cycle.

By carefully planning in the early design phases, we consciously implemented effective clock gating in the SPE microarchitecture. In fact, clock gating is so tightly integrated in the microarchitecture that it is part of the SPE's functionality and therefore can be fully verified through simulation. We have developed a suite of in-house tools for simulating cycle-accurate active power consumption. The power simulation tools work in two steps. First, they simulate a functional block under active power for one clock cycle and tabulate the results for different percentages of clock and data pattern activation. Second, using the active power results and the SPE RTL model, they simulate the SPE's active power consumption for each cycle. The tools feed simulation results back to the SPE design team for further active power optimization. We have achieved a 50 percent active power reduction for a given workload through this design process.

Multiple clock frequency domains

A high-frequency design increases the performance of the SPE execution units, but it incurs some penalties, including higher clock power, higher percentage of latch insertion delay in a clock cycle (Figure 2), and a shorter distance that a signal can travel in a clock cycle. The SPE has several units that require high frequency designs for better performance, and it also has units whose performance does not depend solely on operating frequency. After careful study, we decided to operate the memory flow control block (SMF) at half the frequency of the SPE core units. The 11 FO4 clock domain includes the following functional blocks:

- register file (128 entries, 128 bits, 6 read and 2 write),
- fixed-point unit,
- floating-point unit,
- data forwarding,

- load/store,
- main memory (4 × 64-kbyte local store),
- branch,
- instruction issue control, and
- main control.

The 22 FO4 clock domain includes the direct memory access unit and the bus control.

Figure 9 shows the location of the two clock domains. Ideally, two clocks with different frequencies can be distributed to reduce high-frequency clock power. In our implementation, however, we distributed one high-frequency clock to both domains, and a secondary clock-like signal activates the SMF blocks every other cycle. This avoids physical implementation difficulties caused by the expected asynchrony at the boundary between the SPE core and the SMF. Operating the SMF at half frequency lets us escape the latch insertion delay and travel distance penalties so that a higher percentage of a clock cycle can be dedicated to logic and signal distribution. As a result, a higher percentage of SMF paths become noncritical, allowing the SMF circuits to use smaller transistors and, in some cases, high-threshold-voltage transistors. Thus, we optimized the SMF design for both area and power and optimized the SPE overall design without sacrificing SPE core performance.

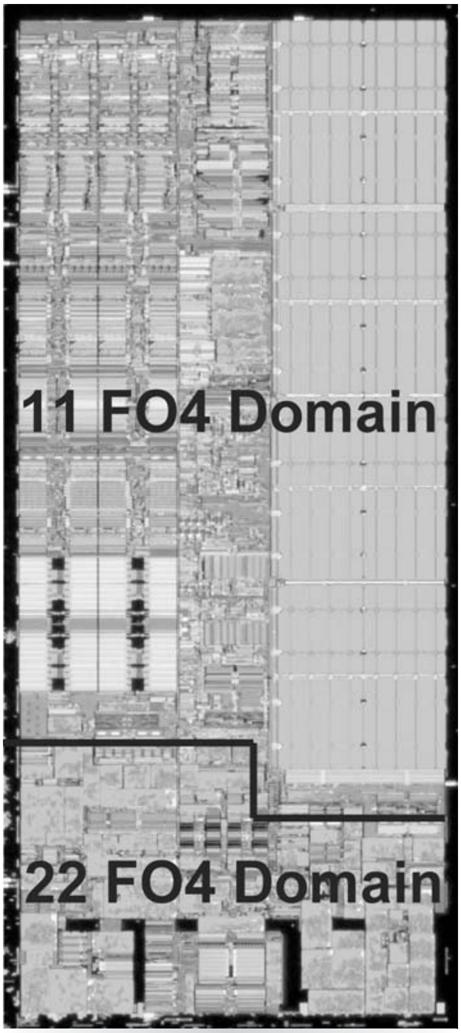


Figure 9. SPE's two clock domains.

Dual-threshold-voltage devices

As technologies advance to deep-submicron regions, leakage has become a significant portion of processors' power consumption. Neither the clock-gating scheme nor the two clock domains can reduce DC power consumption. An obvious approach to reducing DC power is to use high-threshold-voltage transistors. The penalty, of course, is their slower switching time. Thus, we couldn't use high-threshold-voltage transistors everywhere in the SPE, at least not with the 90nm SOI technology. Instead, where there was enough timing slack after optimizing path delays, we changed the type of transistors in the paths from regular- to high-threshold-voltage devices. We did this extensively for synthesized logic. As explained previously, having two clock domains makes some SMF paths less critical. We replaced the transistors in these paths with high-threshold-voltage devices.

Table 1 lists the numbers of different transistor types in the SPE, excluding analog devices. High-threshold-voltage devices are used roughly 11 percent in the 22 FO4 domain and 0.2 percent in the 11 FO4 domain. Overall, 2.9 percent of non-SRAM

Table 1. Multiple-V_T devices used in the SPE (HVT: high-threshold-voltage).

Device type	Device count
NMOS	2,962,365
NMOS HVT	93,751
PMOS	2,810,167
PMOS HVT	76,464
NMOS SRAM	9,991,744
PMOS SRAM	4,992,896
Total	20,927,387

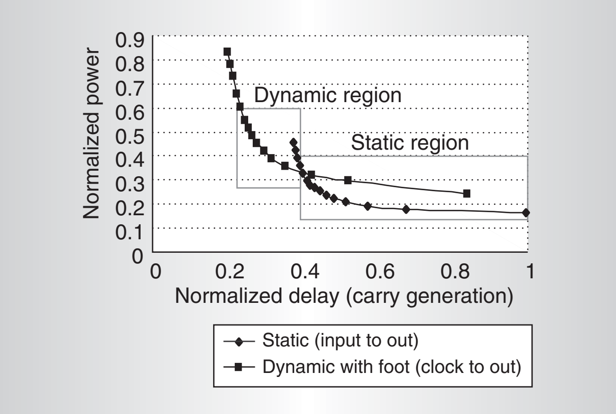


Figure 10. Comparison of power-speed curves.

transistors in the SPE are high-threshold-voltage devices.

Selective use of dynamic circuits

Static circuits are predominant in processor designs. Because they have clear advantages over their dynamic counterparts, static circuits are widely used in the industry. Their advantages include

- design ease,
- low switching factor,
- tool compatibility, and
- technology independence.

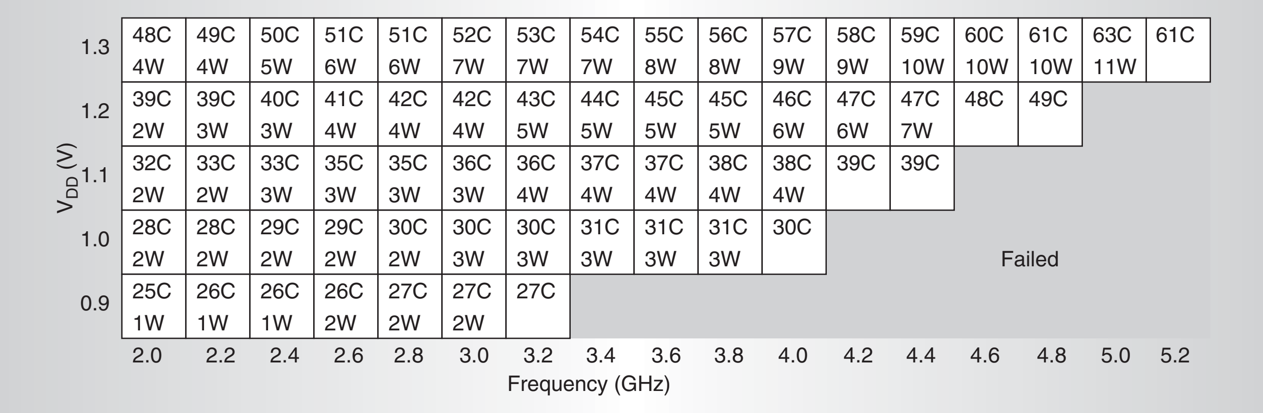
On the other hand, the advantages of dynamic circuits include

- faster speed due to lower node capacitance at dynamic nodes,[9]
- larger gain by inverters after dynamic nodes,
- microarchitectural efficiency through combining multiple stages into fewer stages, and
- smaller area.

Because dynamic circuits generally require a clocklike signal that toggles every cycle, a dynamic implementation tends to consume more AC power than an equivalent static implementation. It also tends to require both true and complementary signals, naturally resulting in higher AC power. However, a static implementation tends to hit a speed wall earlier than a dynamic implementation.

Figure 10 shows a comparison of the power-speed curves of two implementations of a carry-generation circuit. For the static implementation, the normalized active power consumption is plotted as a function of the normalized delay of data from the input to the output of the gate. For the dynamic implementation, the normalized active power consumption is plotted as a function of the normalized delay from clock arriving at the gate to the output appears. For both cases, the loading at the output is increased towards the left and therefore the device sizes of gates are increased to have more drive capability. When the target speed is not aggressive, the static implementation clearly uses less power than the dynamic implementation. However, when the target speed becomes more aggressive, the static implementation seems to hit a speed wall. The transistors in static circuits become too large and no longer effective. When the speed target is higher than the speed wall, the static implementation consumes more AC power than the equivalent dynamic approach. To realize a high-frequency design, identifying the speed wall for each macro is critical. At the same time, each circuitry implementation can have different requirements such as area, aspect ratio, porosity, power consumption, noise immunity, and delay. Selecting an appropriate implementation that best satisfies all requirements is essential to a successful processor design.

Our SPE design philosophy was to implement logic in static circuits including full CMOS and transmission gates as much as possible. We considered alternatives only when static implementations did not meet our initial objectives. For example, we first designed a general-purpose register file using static and tristate circuits.[3] After the static design failed to achieve speed and area objectives, we considered and finally chose a dynamic circuit approach. Another example is the forward macro in the SPE data flow.[3] It is basically a wide (16-way) multiplexer with memory elements, and its read path is replicated six times to have six ports. In other words, it is a complex, 16-entry, six-read-port register file with internal shift capability. Again, we first designed it using static and tristate circuits. However, the static implementation didn't meet speed, power, area, and porosity objectives, so we chose a dynamic implementation similar to that of the general-purpose register file.



V_{DD} (V) / Frequency (GHz)	2.0	2.2	2.4	2.6	2.8	3.0	3.2	3.4	3.6	3.8	4.0	4.2	4.4	4.6	4.8	5.0	5.2
1.3	48C 4W	49C 4W	50C 5W	51C 6W	51C 6W	52C 7W	53C 7W	54C 7W	55C 8W	56C 8W	57C 9W	58C 9W	59C 10W	60C 10W	61C 10W	63C 11W	61C
1.2	39C 2W	39C 3W	40C 3W	41C 4W	42C 4W	42C 4W	43C 5W	44C 5W	45C 5W	45C 5W	46C 6W	47C 6W	47C 7W	48C	49C		
1.1	32C 2W	33C 2W	33C 3W	35C 3W	35C 3W	36C 3W	36C 4W	37C 4W	37C 4W	38C 4W	38C 4W	39C	39C				
1.0	28C 2W	28C 2W	29C 2W	29C 2W	30C 2W	30C 3W	30C 3W	31C 3W	31C 3W	31C 3W	30C				Failed		
0.9	25C 1W	26C 1W	26C 1W	26C 2W	27C 2W	27C 2W	27C										

Figure 11. SPE shmoo plot (frequency versus AC power, excluding global clock).

A third example is an opposite case. The SPE dependency check macro calculates dependencies among instructions in the SPE pipeline.[3] It requires a combination of complex calculations such as bit-by-bit EXOR, a wide OR, and a wide AND. The early phase cross-sectional simulations showed a static implementation might not satisfy the speed goal, so we designed a dynamic implementation. However, this macro's location is in the SPE's main control, and its area and porosity requirements are not as demanding as those of dataflow macros such as the general-purpose register file and the forward. Therefore, we successfully converted the design to a static circuit meeting all the objectives.

We used dynamic circuits only where absolutely necessary. All the dynamic circuits have static interfaces, ensuring that no dynamic signal crosses a macro boundary and minimizing downstream logic's switching factors. Roughly, dynamic circuits occupy 19 percent of non-SRAM area in the SPE and include the following macros:

- dataflow forwarding,
- multiport register file,
- floating-point unit result multiplexer,
- dynamic programmable logic array (control),
- multiplexer latch, and
- instruction line buffer.

SPE hardware measurements

We have rigorously tested the SPE and observed it correctly performing operations for complicated workloads such as 3D picture rendering. The fastest operation ran at 5.6 GHz with a 1.4 V supply at 56° C under laboratory conditions. We expect products using the Cell processor to trade operating frequency for low-voltage operation and significantly reduced power.

Figure 11 shows the voltage frequency shmoo plot and power characteristics of the SPE. The measured temperature and power are listed for each point of the plot. An on-chip thermal sensor measured temperature. We measured power at a supply by subtracting the power of one SPE activated from that of two SPEs activated at the same time. Thus, the power number includes only active power and excludes power consumed by global clock distribution and leakage. The global clock mesh's measured power is 1.3 W per SPE at a 1.2-V supply and 2.0-GHz clock frequency. The SPE measured DC power at a 1.2-V supply and 39°C is 1.7 W. The power measurements reflect a single SPE running a single-precision, floating-point, computation-intensive lighting and transformation workload using a 16-kbyte portion of local store memory.

The Cell architecture is compatible with the 64b Power architecture so that applications can be built on the Power investments. It can be considered as a non-homogenous coherent chip multi-processor. High design frequency has been achieved through highly optimized implementation. Its streaming DMA architecture helps to enhance memory effectiveness of a processor. We believe that we have contributed to the definition of the Cell architecture, especially at the early design phase by feeding back the feasibility of high speed processor design, high speed memory design, and effective modular design method. The SPE design has been achieved only through a close communication between the logic and the physical design teams. We have kept pushing the envelope of all the design aspects to achieve power efficient, compact, and high speed Synergistic Processor Element design. MICRO

Acknowledgments

We thank the following contributors to the SPE project: Koji Hirairi (SONY Computer Entertainment of America), Atsushi Kawasumi, Takashi Machida, Hideaki Murakami, Hiroaki Murakami, Takaaki Nakazato, Hiromi Noro, and Naoka Yano (Toshiba America Electronic Components), Toru Asano, Russ Cook, Omar Delgado, Jens Leenstra, Brad Michael, Hiroo Nishikawa, Hwa-Joon Oh, Shohji Onishi, Juergen Pille, Jieming Qi, Dieter Wendel, Michael White, and Suksoon Yong (IBM).

References

1. D. Pham et al., "The Design and Implementation of a First-Generation Cell Processor," *Proc. Int'l Solid-State Circuits Conf.* (ISSCC 05), IEEE Press, 2005, pp. 184-185.
2. B. Flachs et al., "A Streaming Processor Unit of a Cell Processor," *Proc. Int'l Solid-State Circuits Conf.* (ISSCC 05), IEEE Press, 2005, pp. 134-135.
3. O. Takahashi et al., "The Circuits and Physical Design of the Synergistic Processor Element of a Cell Processor," *Proc. Symp. VLSI Circuits*, IEEE Press, 2005, pp. 20-23.
4. T. Asano et al., "A 4.8GHz Fully Pipelined Embedded SRAM in the Streaming Processor of a Cell Processor," *Proc. Int'l Solid-State Circuits Conf.* (ISSCC 05), IEEE Press, 2005, pp. 486-487.
5. J. Leenstra et al., "The Vector Fixed Point Unit of the Synergistic Processor Element of a Cell Processor," to be published in *Proc. 31st European Solid-State Circuit Conf.* (ESSCIRC 05), IEEE Press, 2005.
6. H. Oh et al., "A Fully-Pipelined Single-Precision Floating Point Unit in the Synergistic Processor Element of a Cell Processor," *Proc. Symp. VLSI Circuits*, IEEE Press, 2005, pp. 24-27.
7. H. Kapadia et al., "Reducing Switching Activity on Datapath Buses with Control-Signal Gating," *IEEE J. Solid-State Circuits*, vol. 34, no. 3, Mar. 1999, pp. 405-414.
8. G.K. Yeap, *Practical Low Power Digital VLSI Design*, Kluwer Academic, 1998, chap. 6.
9. N.H.E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design: A System Perspective*, Addison-Wesley, 1992, chap. 5.

Osamu Takahashi is an IBM senior technical staff member and the manager of the circuit design team for the synergistic processor element (SPE) developed at the Sony-Toshiba-IBM Design Center. Takahashi has a BS in engineering physics and an MS in electrical engineering, both from the University of California, Berkeley. He has a PhD in computer and mathematical sciences from Tohoku University, Sendai, Japan.

Scott Cottier is responsible for memory arrays in the SPE of the Cell processor at the Sony-Toshiba-IBM Design Center. Cottier has a BS in electrical engineering from Texas A&M University.

Sang H. Dhong is the chief hardware engineer at the Sony-Toshiba-IBM Design Center. His research interests include microprocessor design and CMOS technology. Dhong has a PhD in electrical engineering from the University of California, Berkeley. He is an IEEE Fellow.

Brian Flachs served as architect, microarchitect, and unit logic lead for the SPE team at IBM Systems and Technology Group. His research interests include low-latency, high-frequency processors. Flachs has a BSEE from New Mexico State University and an MS and a PhD from Stanford University.

Joel Silberman is a research staff member at the IBM T.J. Watson Research Center in Yorktown Heights, New York. His research interests include tradeoffs in microarchitecture and circuit performance of advanced cache memory and instruction issue logic. Silberman has a PhD in electrical engineering from Stanford University.

Direct questions and comments about this article to Osamu Takahashi, IBM, 11501 Burnet Road, MS 906-2003C, Austin, TX 78758; osamu@us.ibm.com.