Reed Electronics Group

MICROPROCESSOR REPORT

www.MPRonline.com

THE INSIDER'S GUIDE TO MICROPROCESSOR HARDWARE

The New Power Architecture

Freescale and IBM Work Together and Begin Revamping PowerPC

By Tom R. Halfhill {8/21/06-01)

After years of following different paths, the two key founders of the PowerPC architecture have renewed their historic collaboration. Working closely together again—now within the Power.org industry consortium—Freescale (the former semiconductor division of Motorola) and IBM are uniting their visions for the 15-year-old microprocessor architecture.

Power.org has announced a new architectural definition that brings together features from both Freescale and IBM and lays the groundwork for future convergence. For the first time, all the documentation will be consolidated in a common format. And hereafter, the common architecture will be called the Power Architecture. "PowerPC" is relegated to existing products and historical references.

Marketing whizzes at Freescale, IBM, and other Power.org companies have adopted a snazzy new logo for the Power Architecture: a Möbius strip twisted in the shape of a "P" (see Figure 1). But the most important result of the Freescale and IBM rapprochement is Power ISA 2.03, a newly defined instruction-set architecture that merges key parts of existing PowerPC ISAs and adds new features. Freescale and IBM want to present a unified, scalable architecture that preserves software compatibility and provides for future expansion. Power ISA 2.03 is only the beginning of their restored collaboration.

Despite some competitive differences between IBM and Motorola/Freescale, the PowerPC architecture has remained cohesive throughout its history. It has always maintained software compatibility at the user ISA level. Its diverging evolution reflected the different market priorities of the two companies, more than any other factor. Both companies have steadily invested in the architecture and naturally want to preserve their investments. At the same time, technology convergence in the marketplace is making a versatile microprocessor architecture like the PowerPC even more valuable. In short, Freescale and IBM have important business and technical reasons for renewing their alliance.

Serious negotiations over Power ISA 2.03 began after Freescale ended its holdout earlier this year and joined Power.org, an open-membership consortium that IBM formed in 2004 to promote the Power Architecture. (See

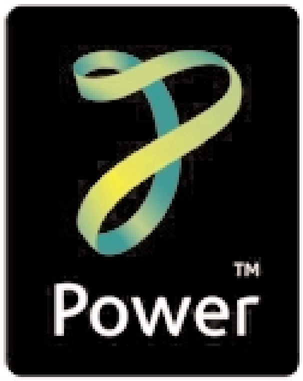

Figure 1. This new logo symbolizes the newly unified Power Architecture. Supposedly, the Möbius strip represents motion, energy, and infinite possibilities. Although a Möbius strip does indeed go on forever, as Freescale and IBM hope their new collaboration will, it can also have many twists and turns.

MPR 3/6/06-01, “Freescale Strengthens Power.Org,” and *MPR 12/27/04-02*, “Bringing Power to the People.”) Both companies belong to an exclusive group-within-a-group called the Power Architecture Advisory Council (PAAC). In fact, Freescale and IBM are the only members of PAAC, which makes the big decisions regarding architectural definitions. The 40-odd other members of Power.org can offer input but lack binding votes.

Merging the two branches of the PowerPC architecture is a huge step. Now that the architecture’s key founders are working together again, the vast community of Power Architecture developers and customers can breathe easier and look forward to future developments. Potential customers should be reassured, too. And the harmony is particularly important for architectural licensees, which are allowed to design their own Power Architecture microprocessor cores. In addition to Freescale and IBM, the publicly known architectural licensees are AMCC, P.A. Semi, and Rapport.

Preserving Scalability and Compatibility

The first result of PAAC’s work is Power ISA 2.03. Although some details (including new features) will remain secret until September, Power ISA 2.03 is the first step toward merging the various parts of PowerPC into a single, coherent ISA. The base ISA has always been consistent and compatible across all PowerPC processors, but, over the years, Freescale and IBM have introduced minor modifications and major extensions.

To accommodate those differences, PAAC has created new subdivisions called “categories” for specific application classes. Some categories will have extensions not found in other categories. These subdivisions are unavoidable, because the Power Architecture spans an unusually wide range of application domains. Some features simply aren’t appropriate for some applications.

Consider that at one end of the spectrum, some existing PowerPC cores in deeply embedded systems measure only 1.0mm^2. At the other end, Freescale has its speedy MPC744x processors, and IBM has its multicore Power chips packaged in multichip modules for enterprise-class servers and world-class supercomputers. (See *MPR 7/5/05-01*, “PowerPC Ain’t Dead Yet,” and *MPR 2/27/06-02*, “ISSCC ‘06: The Large and the Fast.”)

The few other CPU architectures that span a similar range, though less successfully, are MIPS, SPARC, and the x86. MIPS Technologies no longer designs server or workstation processors, having long ago abandoned big iron for the embedded market. SPARC is still popular in servers and workstations, but its popularity in embedded systems is waning. The x86 is a strong across-the-board competitor, despite a dearth of low-power embedded cores and licensable cores. Other CPU architectures are more often found at one end of the computing spectrum or the other. For instance, ARM is the low-power king, and Intel’s IA-64 (Itanium) is frozen as a server architecture.

Freescale and IBM want to preserve Power’s vast scalability without compromising software compatibility. Of course, there will necessarily be differences between Power cores designed for, say, deeply embedded microcontrollers in automotive systems and the monster processors designed for supercomputers installed at national laboratories. Freescale focuses on the embedded market, especially communications, whereas IBM is more interested in heavy-duty computing and electronic gaming. IBM Power-based chips are found in the world’s fastest supercomputers and in all three new-generation home videogame consoles: the Microsoft Xbox, Nintendo Wii, and Sony PlayStation 3. (See *MPR 10/11/04-01*, “IBM Makes Designer Genes”; *MPR 3/13/06-01*, “The Cell, At One”; and *MPR 7/18/05-02*, “Powering Next-Gen Game Consoles.”)

Unfortunately, in the middle of the computing spectrum is a huge gap where the Power Architecture is virtually absent: desktop and notebook PCs. Apple’s recent defection to the x86 has pretty much ended the early ambitions to establish PowerPC in the mainstream PC market. (See *MPR 2/6/06-02*, “Apple + Intel = New Products.”) Perhaps that’s why “PowerPC” has been truncated to “Power.” Borrowing some nomenclature from Coca-Cola, Freescale and IBM now refer to early versions of the architecture as “PowerPC Classic.”

Fifteen Years of Evolution

To understand where Power ISA 2.03 is now, it’s necessary to review the somewhat confusing history of the PowerPC architecture. Although its official birthday was 1991, it is actually an outgrowth of an earlier IBM architecture that began shipping in 1990 with the RISC System/6000. Later, this 32-bit RS/6000 architecture was named POWER (Performance Optimization With Enhanced RISC)—not to be confused with today’s Power Architecture. It was a 32-bit architecture, and the first implementations occupied nine or ten chips.

In 1991, Apple, IBM, and Motorola formed the AIM Alliance to develop the PowerPC architecture. (See *MPR* 10/16/91, p. 1, “Apple, IBM, and Motorola Sign Contracts for Far-Reaching Collaboration.”) They jointly opened and operated the Somerset Design Center in Austin, Texas. IBM contributed most of the ISA from the RS/6000 architecture. Motorola designed most of the I/O buses for the first PowerPC implementations by adapting its 88110 RISC architecture. Apple was a minor design partner but a major early customer, committing to switch the Macintosh from Motorola 68000-based processors to PowerPC.

AIM defined PowerPC as a 64-bit architecture from the start—13 years before the x86 made the transition from 32 bits. However, few customers in the early 1990s needed 64-bit processing, so the first PowerPC chips were 32-bit implementations. Exceptions were the PowerPC 620 and 630; these were either stillborn or found little traction in the marketplace.

The first official PowerPC processor was the 32-bit 601, but it inherited so much genetic material from IBM's RS/6000 that it was really a stopgap measure until the Somerset engineers could finish a true PowerPC chip. That chip was the PowerPC 603, announced in 1993. (See *MPR* 10/25/93, p. 11, "Motorola and IBM Unveil PowerPC 603.") The PowerPC 601, 603, and 604 processors began shipping in 1994, the same year Apple introduced the first Power Macs.

Now called PowerPC Classic 1.0, the first architectural definition in 1991 consisted of three "books," or sections. That terminology and much of the books' contents survives in the new Power ISA 2.03. Book I defines the user-level ISA, including basic instructions for flow control, memory reference, integer arithmetic, and floating-point arithmetic. These are the features that application programmers use. Book II defines the virtual environment architecture, which is for programmers writing nonprivileged-level library routines. Book III defines the operating-environment architecture, which provides resources for privileged-level programming, such as operating systems and hypervisors.

Some Books Were Never Published

More books followed later in the 1990s, after IBM and Motorola began drifting apart. Motorola, already focusing more intently on the embedded market, developed Book VLE, which added variable-length instruction encoding to PowerPC. VLE supplements the standard 32-bit instructions with some 16-bit opcodes to improve code density, an important consideration for embedded-system programmers.

In concept, Book VLE is similar to ARM's Thumb. Indeed, almost every 32-bit RISC architecture with eyes on the embedded market has embraced some form of abbreviated instruction encoding. IBM and Motorola also tried to improve code density by introducing different methods of real-time code decompression, such as IBM's CodePack. (See ***MPR 10/26/98-05***, "PowerPC Adopts Code Compression.") However, these methods required a special on-chip decompression unit and caused problems for software developers.

Motorola never actually published Book VLE. It remained an unofficial extension to the PowerPC architecture, although Freescale introduced variable-length encoding in 2005. This is one example of the divergence that began creeping into the architecture as IBM and Motorola/Freescale followed different paths. Their joint venture at the Somerset Design Center ended in 1998.

Cooperation continued, however. In 1999, IBM and Motorola introduced Book E, which adds features specifically for embedded applications. Although variable-length encoding was an unrealized goal in Book E, the revision did simplify the PowerPC memory-management model, introduce two-level interrupts, provide resources for timers, and add modeless 32/64-bit execution. (See ***MPR 5/10/99-02***, "PowerPC Architecture Gets Makeover.") The modeless 32/64-bit execution is particularly notable, because it created a new problem that Power ISA 2.03 is resolving only now, five years later.

Freescale says Microsoft was an advocate for modeless 32/64-bit execution—this back in the days when a version of Windows NT ran on PowerPC. But as things turned out, nobody ever implemented the feature in a PowerPC processor. IBM's 64-bit PowerPC 970, introduced in 2002, didn't use the modeless execution of Book E. (See MPR *10/28/02-02*, "IBM Trims Power4, Adds AltiVec.")

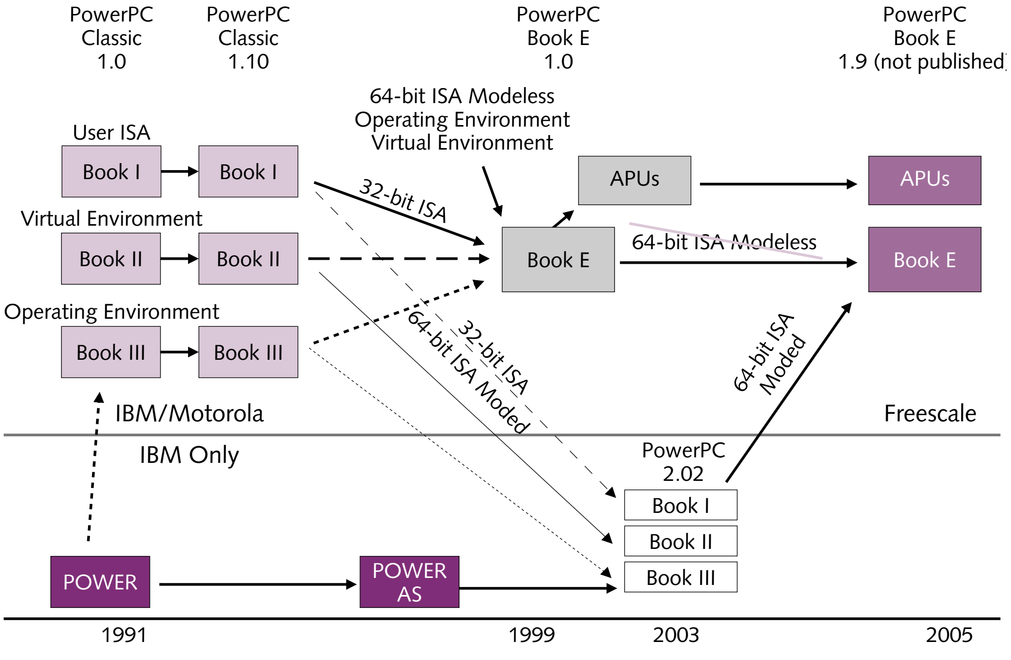

Figure 2. In 1991, IBM contributed DNA from its RISC System/6000 architecture to help Motorola and Apple define the original PowerPC architecture, now called PowerPC Classic 1.0. The AIM (Apple-IBM-Motorola) Alliance published various aspects of the PowerPC architectural definition in a series of books identified by Roman numerals. A major enhancement came in 1999 with Book E, which added new features specifically for embedded PowerPC processors. Freescale Semiconductor—a Motorola spinoff—wrote, but never published, Book E 1.9 in 2005. The modeless 32/64-bit execution introduced in Book E was dropped in Book E 1.9 and in the latest Power ISA 2.03.

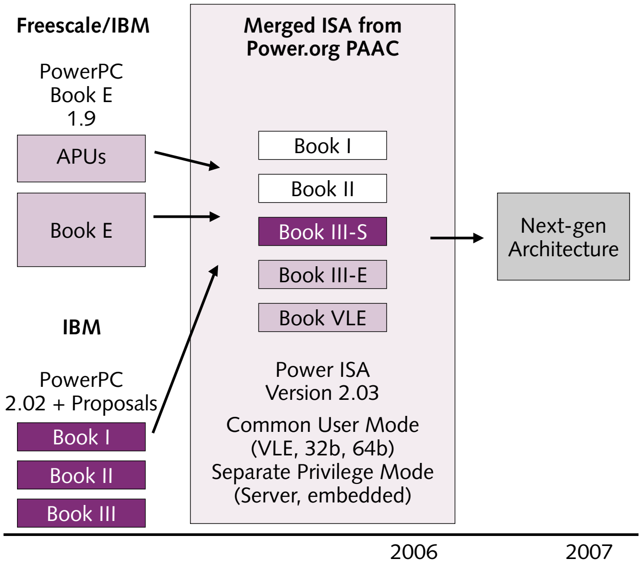

Figure 3. As the sole members of Power.org's Power Architecture Advisory Council (PAAC), Freescale and IBM have created Power ISA 2.03. The consolidated ISA merges features from the Freescale and IBM modifications to the PowerPC Classic architecture. Power ISA 2.03 includes books of features brought forward from previous definitions of the architecture.

In 2005, Freescale—after spinning off from Motorola the year before—created Book E 1.9, which dropped modeless 32/64-bit processing from the architectural definition. However, as with Book VLE, Freescale never officially released Book E 1.9. Not until Freescale and IBM joined forces in PAAC did they agree to officially drop the modeless 32/64-bit features in Power ISA 2.03. Figure 2 illustrates the evolution of the PowerPC architecture from 1991 to 2005.

Power ISA 2.03 Restores Order

Another significant addition to PowerPC in the late 1990s was Motorola's introduction of auxiliary processing units (APU). These are extension packages that enhance the architecture for various application domains, usually in the embedded realm. The extensions may consist of new instructions, registers, memory-management structures, or other features. One of PAAC's challenges was deciding how to integrate the APUs into the merged Power Architecture.

Figure 3 illustrates the way Freescale and IBM created Power ISA 2.03, using all the existing books and APUs from both companies. Most features from Book E and the APUs are merged into a new version of Book III for embedded processors, creating Book III-E. Most features from IBM's Book I, Book II, and Book III—which are a superset of the PowerPC Classic 1.0 books—are merged into a new version of Book III for server-class processors, called Book III-S. Freescale's Book VLE is another volume on the shelf. The collective result is Power ISA 2.03.

Freescale and IBM refer to Power ISA 2.03 as the "merged architecture," because it unites the base architectural features of PowerPC Classic 1.10, the embedded features of Book E, Freescale's APUs, and the Advanced Server (AS) features from IBM's branch of the family tree. The two companies already are working on a near-future version, Power ISA 2.04, which they call the "converged architecture." This revision will go further toward cleaning up the

Power/PowerPC Book	Contents	Some Instructions and Features	Programming Environment
Book I (1991)	User-level instruction-set architecture (UISA)	Memory reference, flow control, integer math, FP math, numeric acceleration, applilcation-level programming model	Application-level programming, usually with compiled code
Book II (1991)	Virtual environment architecture (VEA)	Time, synchronization, cache management, storage features, byte ordering	Library-level nonprivileged programming, usually with assembly language
Book III (1991)	Operating environment architecture (OEA)	Exceptions, interrupts, memory management, debugging, special control instructions and resources	Supervisors and hypervisors
Book VLE (Unofficial: 2005)	Variable-length encoding (VLE)	Compact instruction encoding using 16- and 32-bit opcodes	User- and system-level embedded programming
Book E 1.0 (1999)	Enhanced ISA for embedded	Modeless 32/64-bit operations, two levels of interrupts	User- and system-level embedded programming
Book E 1.9 (2005)	Never-published update to Book E	Drops the modeless 32/64-bit execution defined in Book E	User- and system-level embedded programming
Book III-E (2006)	Updates Book E for Power ISA 2.03	Drops the modeless 32/64-bit execution defined in Book E	System-level embedded programming
Book III-S (2006)	Updates Book III for servers	Advanced memory management and other high-end features	Supervisors and hypervisors

Table 1. The PowerPC architecture is defined in a collection of books dating back to the original definition in 1991. Each book describes a different aspect of the architecture and has been revised over the past 15 years. Some revisions, such as Book VLE and Book E 1.9, were never publicly released. Power ISA 2.03—the latest definition of the common Power Architecture—inherits something from all the books in this table.

architectural definition, perhaps by resolving the differences between Book III-E and Book III-S, among other details. However, the amount of work remaining to be done might push true convergence beyond Power ISA 2.04; more will be known later this year. Table 1 summarizes the books playing a role in Power ISA 2.03.

To merge Freescale's APUs into Power ISA 2.03, the two companies created the previously mentioned "categories," which are optional packages of extensions or subsets of the architecture. An alternative solution would have been to integrate all the existing extensions and APUs into a single monolithic architecture, but that would saddle Power processors with too much excess baggage. Some features required for server processors (such as extensive memory-management structures implemented in hardware) are too heavyweight for embedded processors, while some features for embedded processors (such as variable-length instruction encoding) are superfluous in server processors.

Base Category Defines Common Ground

One Power ISA 2.03 category, called *Base*, defines the basic architectural features that all Power processors must support. This category preserves a great degree of software compatibility, all the way back to PowerPC Classic 1.10, and ensures that future Power processors will share much in common. *Base* includes most of Book I and Book II, defining almost all the instructions and registers.

Notably absent from the *Base* category is support for floating-point math, which is a separate category. Actually, there are multiple floating-point categories, because one of Freescale's APUs defines a lightweight FPU that's more appropriate for embedded processors than the heavyweight FPU in workstation and server processors. Table 2 shows how PAAC converted Freescale's APUs in Book I, Book II, Book III-E, and Book VLE into new Power ISA 2.03 categories.

Some categories have subcategories, which are indicated using notation similar to that in object-oriented programming. For example, the *Embedded* category has subcategories named *Embedded.Cache Locking* and *Embedded.Performance Monitor*, among others. *Embedded.Cache Locking* provides support for cache line-locking, which stops the processor from flushing critical instructions or data from portions of the cache. *Embedded.Performance Monitor* provides support for recording run-time data, which engineers can use to analyze the processor's performance.

The renowned AltiVec multimedia extensions, which Motorola introduced in 1998, comprise the new *Vector* category. (See ***MPR 11/16/98-04***, "G4 is First PowerPC With AltiVec.") Motorola trademarked the name AltiVec; IBM referred to the extensions as VMX, and Apple used the term Velocity Engine. *Vector* is a relatively bland but descriptive name for the extensions. (See ***MPR 5/11/98-01***, "AltiVec Vectorizes PowerPC.")

Another interesting category is *Alternate Time Base*, carried over directly from a Freescale APU of the same name. This extension allows developers to implement a 64-bit timer that counts in increments synchronized to the processor's core clock frequency. The PowerPC architecture already has a 64-bit time-base register, but it usually counts in increments synchronized to the I/O bus clock. Synchronizing a timer to the core frequency allows finer-grain measurements.

Existing Freescale APU	New Power 2.03 Category	Description
Book I		
AltiVec	Vector	128-bit-wide multimedia SIMD
Isel (instruction select)	Base	Conditional register move to reduce conditional branches
Signal-processing engine (SPE)	Signal-Processing (SP) Engine	64-bit-wide SIMD for DSP
Embedded FP (Single, double, vector)	SP.FS, SP.FD, SP.FV	Low-cost, saturating floating-point operations
Book II		
Alternate time base	Alternate Time Base	Fine-grain timer function, can increment at core frequency
Book III-E		
Cache line locking	Embedded.Cache Locking	Lock cache lines to prevent cache thrashing
Enhanced debug	Embedded.Enhanced Debug	Additional interrupt level for debugging
Performance monitor	Embedded.Performance Monitor	Records performance-related run-time data
Machine check	Embedded	Additional interrupt level for hard-error interrupts
Book VLE		
Variable-length encoding	Variable-Length Encoding	16/32-bit instruction encoding

Table 2. In the 1990s, Motorola introduced extension packages that adapt PowerPC processors for embedded applications. Called auxiliary processing units (APU), these extensions amounted to unofficial modifications of the architecture. Power ISA 2.03 makes the APUs official by renaming them "categories" and merging them into the new definition of the Power Architecture.

For More Information

Power.org press release about Power ISA 2.03:

- *www.power.org/news/pr/view?item_key=dadf2018a9f3e4a258a62dbf2c16a12e7b4b5c6c*

Additional information is available at these sites:

- *www.power.org*
- *www.freescale.com/powerarchitecture*
- *www.ibm.com/chips/power/*

PowerPC Classic 1.10 Memory Management	Power ISA 2.03 Book III-E Memory Management
Segmented virtual address space **16 segment registers**	**Unsegmented virtual address space** No segment registers
Hardware-managed TLB **Hashed reverse page tables**	**Software-managed TLB** Hardware assistance for TLB replacement; no required page-table format
Fixed-size (4KB) pages **Variable-size translation using block address translation (BAT)**	**Fixed and variable-size pages**
Separate side translations for instructions and data	**Unified instruction and data TLB**
Real mode and virtual mode **Real mode: translation off** **Virtual mode: translation on**	**Virtual mode only** MMU can emulate real mode

Table 3. Power ISA 2.03 adapts PowerPC's Book E to create Book III-E, which defines memory-management features for Power Architecture embedded processors. Book III-E allows embedded processors to manage memory using fewer hardware structures than PowerPC's Book III does. Simpler memory management conserves silicon, a vital consideration for embedded processors.

PAAC deemed one Freescale APU valuable enough to include in the *Base* category in Power ISA 2.03. That APU is called Isel (instruction select). It's a single instruction that can move operands in or out of registers, depending on the state of a bit in the existing condition register. Isel can eliminate separate branch-and-compare instructions, reducing the chance that the processor will mispredict a conditional branch. Mispredicted branches are particularly costly in automotive MCUs dependent on flash memory; Freescale currently implements the Isel APU in its PowerPC e200 and e500 processor cores.

Coming Soon: More Convergence

Memory management was another challenge for PAAC. Big processors for workstations and servers need robust memory-management features to run their multithreaded, multitasking, multiprocessor operating systems. The latest trend is to add a hypervisor sublayer that allows a machine to run multiple operating systems simultaneously. In contrast, small processors for embedded systems have stingy transistor budgets and often make do with simpler operating systems—or, in some cases, with no operating system at all.

Book E addressed the issue of memory management in 1999. Book III-E in Power ISA 2.03 carries forward all those features. Table 3 compares memory management in PowerPC Classic 1.10 with memory management in Book III-E.

As time passes, however, embedded systems tend to adopt features formerly found only in PCs, workstations, and servers. Multithreading, multitasking, and even multiprocessing are becoming commonplace in some types of embedded processors, especially those used in communications and networking. Even a hypervisor isn't out of the question. Some embedded developers want to run multiple operating systems simultaneously in order to optimize different tasks or to preserve software compatibility across different hardware systems. Another possible reason for an embedded hypervisor is to isolate different tasks for security purposes—for example, to prevent user-application software downloaded to a cellphone from interfering with the basic telephony functions.

Keeping those trends in mind, Freescale and IBM hint that future revisions of the Power ISA will adapt higher-end features to embedded processors. Figure 4 shows how Power ISA 2.03 defines privilege levels today. User and supervisor levels are standard, and the optional hypervisor level is part of the Server category. Don't be surprised if a hypervisor privilege level appears in a future version of Book III-E, or if Book III-E and Book III-S merge in some fashion—perhaps with the additional privilege level defined as an optional category.

Register files are another vital part of a CPU architecture. Power ISA 2.03 doesn't tinker with the existing registers, other than to associate them with the appropriate categories. Of course, as the Power Architecture gains new features, it will add new registers. For now, the *Base* category includes the canonical RISC set of 32 general-purpose registers for integer operands, plus a few programmer-visible control registers, such as the aforementioned condition register and time-base register. These registers may be 32 or 64 bits wide, depending on the implementation. The 32 floating-point registers are part of the optional floating-point categories. AltiVec registers are part of the *Vector* category, and

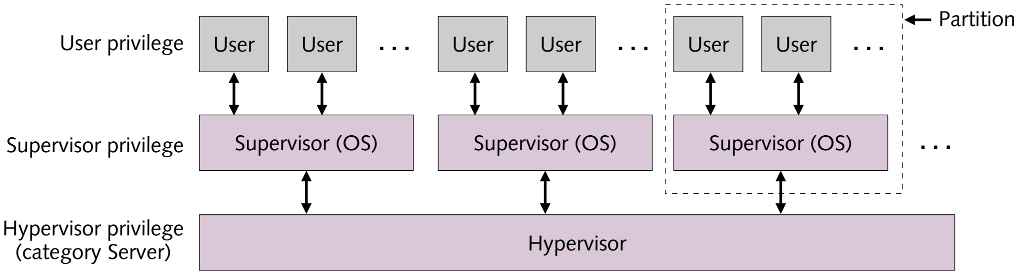

Figure 4. Power ISA 2.03 defines three privilege levels for software execution, but one level is optional. User level is for application software; supervisor level is for operating systems; and the optional hypervisor level is for a sublayer that allows multiple operating systems to run on a machine simultaneously, each in its own virtual partition. Currently, the hypervisor level is part of the *Server* category in Book III-S.

additional registers are associated with signal-processing extensions and other optional categories. Figure 5 illustrates the register sets in the Power ISA 2.03 programmer's model.

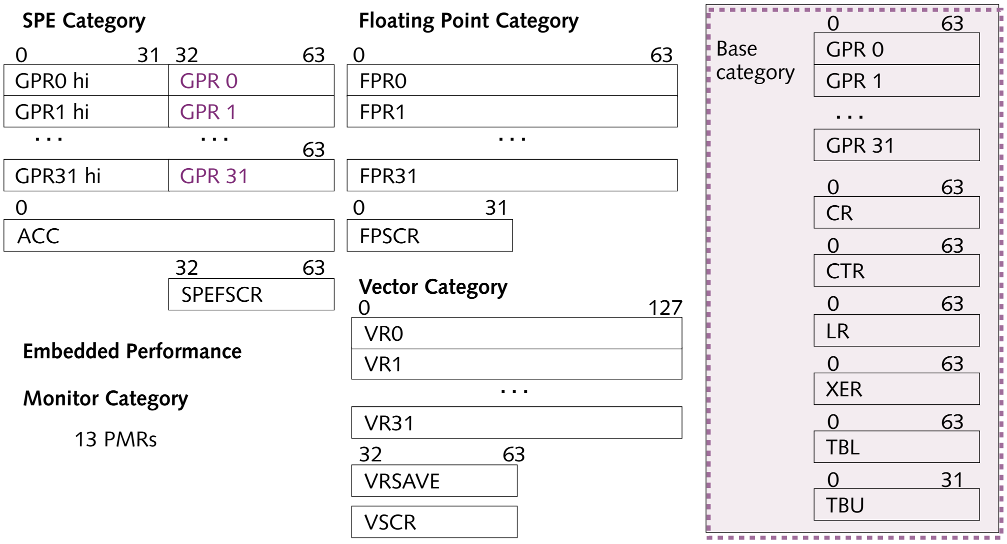

Figure 5. Power ISA 2.03 register files. All existing PowerPC and future Power Architecture processors support the integer and programmer-visible control registers in the *Base* category. Other register sets are optional and are part of Power ISA categories for specific application domains.

Cooperation Strengthens Power Architecture

Now that Freescale and IBM are fully cooperating again, the future of the Power Architecture looks bright. True, it's not quite as bright as in those heady days of the early 1990s, when the AIM Alliance dreamed of wresting the PC market from Intel. Since then, Apple has defected from the alliance and switched the Macintosh to the x86, dooming PowerPC on the desktop.

Fortunately, the Power Architecture is solidly entrenched in the embedded market. The architecture isn't quite that strong in the server market, but it might be a survivor if IBM can hold on. The Alpha, MIPS, and PA-RISC architectures have virtually vanished from servers. SPARC is threatened by downsizing at Sun, and Intel's Itanium isn't the world-beater it was meant to be.

Although the modern Power Architecture is 15 years old—a long time in the computer industry—it's actually one of the youngest CPU architectures still enjoying wide popularity. MIPS and SPARC date to the 1980s, and the x86 was born in the 1970s. ARM is about the same age as Power, depending on when you mark its birthday. That a 15-year-old CPU architecture could still seem new is a disquieting commentary on the state of progress in an industry that's typically viewed as fast-moving.

Sure, plenty of new architectures have made their debut in the past 15 years, and new ones keep coming. *MPR* has covered most of them. But almost all new CPU architectures are designed for fairly narrow applications, and none has won broad acceptance. They struggle to find a niche by dodging around the entrenched architectures.

Given the almost insurmountable difficulty of establishing a CPU architecture in the marketplace, any architecture that already has a beachhead is an extremely valuable property. It's not something to be squandered, for any reason. If Freescale and IBM can't make their new alliance work, they risk dooming the Power Architecture in the embedded and server markets, just as they have already lost the PC market. Then, like Apple, they might have nowhere to go but Intel's lair in Santa Clara. Our message to Freescale and IBM: There's no fate but what you make. ◇