RC25146 (W1104-060) April 15, 2011
Computer Science

IBM Research Report

IBM Power Architecture

Tejas S. Karkhanis, José E. Moreira
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Cambridge - Haifa - India - T. J. Watson - Tokyo - Zurich

IBM Power Architecture

Tejas S. Karkhanis and José E. Moreira
IBM T.J. Watson Research Center
Yorktown Heights, NY 10598

SYNONYMS

IBM Power, IBM PowerPC

DEFINITION

The IBM Power architecture is an instruction set architecture (ISA) implemented by a variety of processors from IBM and other vendors, including Power7, IBM's latest server processor. The IBM Power architecture is designed to exploit parallelism at the instruction, data and thread level.

DISCUSSION

1 Introduction

IBM's Power ISA™ is an instruction set architecture designed to expose and exploit parallelism in a wide range of applications, from embedded computing to high-end scientific computing to traditional transaction processing. Processors implementing the Power ISA have been used to create several notable parallel computing systems, including the IBM RS/6000 SP, the Blue Gene family of computers, the Deep Blue chess playing machine, the PERCS system, the Sony Playstation 3 game console, and the Watson system that competed in the popular television show Jeopardy!

Power ISA covers both 32-bit and 64-bit variants and, as of its latest version (2.06 Revision B [8]), is organized in a set of four "books", as shown in Figure 1. Books I and II are common to all implementations. Book I, Power ISA User Instruction Set Architecture, covers the base instruction set and related facilities available to the application programmer. Book II, Power ISA

Virtual Environment Architecture, defines the storage (memory) model and related instructions and facilities available to the application programmer. In addition to the specifications of Books I and II, implementations of the Power ISA need to follow either Book III-S or Book III-E. Book III-S, Power ISA Operating Environment Architecture – Server Environment, defines the supervisor instructions and related facilities used for general purpose implementations. Book III-E, Power ISA Operating Environment Architecture – Embedded Environment, defines the supervisor instructions and related facilities used for embedded implementations. Finally, Book VLE, Power ISA Operating Environment Architecture – Variable Length Encoding Environment, defines alternative instruction encodings and definitions intended to increase instruction density for very low end implementations.

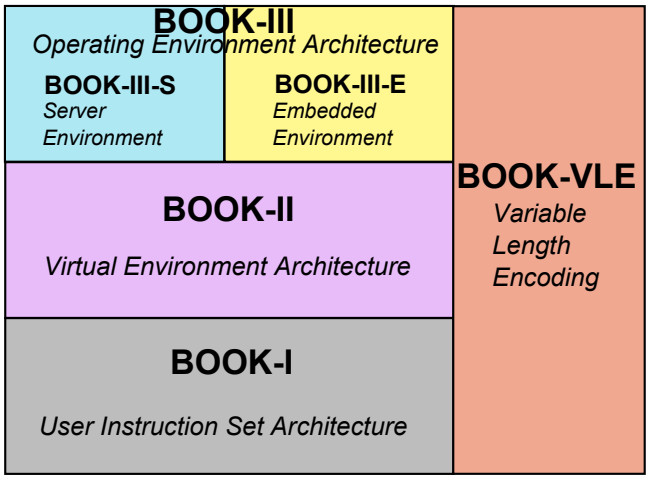

Figure 1: Books of Power ISA version 2.06.

Figure 2 shows the evolution of the main line of Power architecture server processors from IBM, just one of the many families of products based on Power ISA. The figure shows, for each generation of processors, its introduction date, the silicon technology used, and the main architectural innovations delivered in that generation. Power7 [9] is IBM's latest-generations server processor and implements the Power ISA according to Books I, II and III-S of [8]. Power7 is used in both the PERCS and Watson systems and in a variety of servers offered by IBM. The latest machine in the Blue Gene family, Blue Gene/Q, follows a Book III-E implementation.

Power ISA was designed to support high program execution performance and efficient utilization of hardware resources. To that end, Power ISA pro-

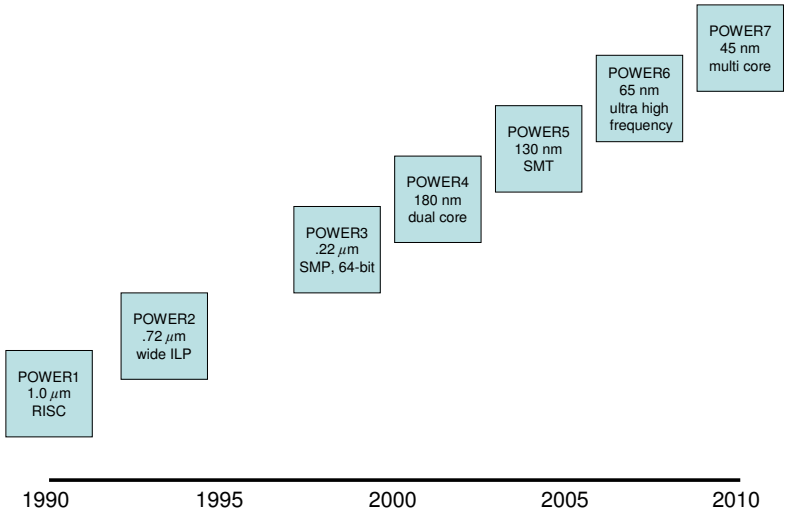

Figure 2: Evolution of main line Power architecture server processors.

vides facilities for expressing instruction level parallelism, data level parallelism, and thread level parallelism. Providing facilities for a variety of parallelism types gives the programmer the flexibility in extracting the particular combination of parallelism that is optimal for his or her program.

2 Instruction level parallelism

Instruction level parallelism (ILP) is the simultaneous processing of several instructions by a processor. ILP is important for performance because it allows instructions to overlap, thus effectively hiding the execution latency of long latency computational and memory access instructions. Achieving ILP has been so important in the processor industry that processor core designs have gone from simple multicycle designs to complex designs that implement superscalar pipelines and out-of-order execution [15]. Key aspects of Power ISA that facilitate ILP are: independent instruction facilities, reduced set of instructions, fixed length instructions, and large register set.

2.1 Independent instruction facilities

Conceptually, Power ISA views the underlying processor as composed of several engines or units, as illustrated in the floor plan for the Power7 processor core shown in Figure 3. Book I of the Power ISA groups the instructions

into “facilities”, including (1) the branch facility, with instructions implemented by the instruction fetch unit (IFU); (2) the fixed-point facility, with instructions implemented by the fixed-point unit (FXU) and load-store unit (LSU); (3) the floating-point facility, with instructions implemented by the vector and scalar unit (VSU); (4) the decimal floating-point facility, with instructions implemented by the decimal floating-point unit (DFU); and (5) the vector facility, with instructions implemented by the vector and scalar unit (VSU). Also shown in Figure 3 is the instruction-sequencing unit (ISU), which controls the execution of instructions, and a level-2 cache.

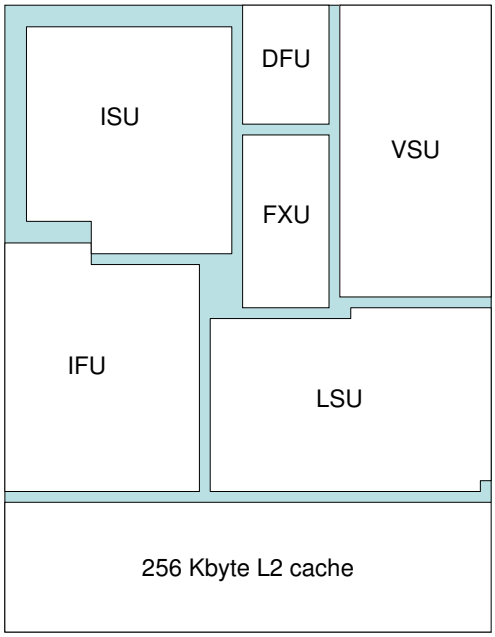

Figure 3: Power7 processor core floorplan, showing the main units.

Origins of this conceptual decomposition are in the era of building processors with multiple integrated circuits (chips). With a single processor spread across multiple chips, the communication between two integrated circuits took significantly more time relative to communication internal to a chip. Consequently, either the clock frequency would have to be reduced or the number of stages in the processor pipeline would have to be increased. Both approaches, reducing clock frequency and increasing the number of pipeline stages, can degrade performance. The decomposition into multiple units allowed a clear separation of work and each unit could be implemented on a single chip for maximum performance.

Today, the conceptual decomposition provides two primary benefits. First, because of the conceptual decomposition, the interfaces between the engines are clearly defined. Clearly defined interfaces lead to hardware design that is simpler to implement and to verify. Second, the conceptual decomposition addresses the inability of scaling frequency of long on-chip wires that can be a performance limiter, just as it addressed wiring issues between two or more integrated circuits when the conceptual decomposition was introduced.

2.2 Reduced set of non-destructive fixed length instructions

Power ISA consists of a reduced set of fixed length 32-bit instructions. A large fraction of the set of instructions are non-destructive. That is, the result register is explicitly identified, as opposed to implicitly being one of the source registers. A reduced set of instructions simplifies the design of the processor core and also verification of corner cases in the hardware.

Ignoring the Book-VLE case, which is targeted to very low end systems, Power ISA instructions are all 32-bits in length, thus the beginning and end of every instruction is known before decode. The bits in an instruction word are numbered from 0 (most significant) to 31 (least significant), following the big-endian convention of the Power architecture. All Power ISA instructions have a major opcode that is located at instruction bits 0 to 5. Some instructions also have a minor opcode, to differentiate among instructions with the same major opcode. The location and length of the minor opcode depends on the major opcode. Additionally, every instruction is word-aligned. Fixed length, word aligned, and fixed opcode location make the instruction predecode, fetch, branch prediction, and decode logic simpler, when compared to the decode logic of variable length ISA. Srinivasan *et. al.* [16] present a comprehensive study of optimality of pipeline length of Power processors from a power and performance perspective.

Instruction set architectures that employ destructive operations (*i.e.*, one of the source registers is also the target) must temporarily save one of the source registers, if the contents of that register are required later in the program. Temporarily saving and later restoring registers often lead to store to and load from, respectively, a memory location. Memory operations can take longer to complete than a computational operation. Non-destructive operations in Power ISA eliminate the need for extra instructions for saving and restoring one of the source registers, facilitating higher instruction level parallelism.

2.3 Large register set

Power ISA originally specified 32 general purpose (fixed-point, either 32- or 64-bit) and 32 floating-point (64-bit) registers. An additional set of 32 vector (128-bit) registers were added with the first set of vector instructions. The latest specification, Power ISA 2.06, expands the number of vector registers to 64. A large number of registers means that more data, including function and subroutine parameters, can be kept in fast registers. This in turn avoids load/store operations to save and retrieve data to and from memory and supports concurrent execution of more instructions.

2.4 Load/store architecture

Power ISA specifies a load-store architecture consisting two distinct types of instructions: (1) memory access instructions, and (2) compute instructions. Memory access instructions load data from memory into computational registers and store the data from the computational registers to the memory. Compute instructions perform computations on the data residing in the computational registers. This arrangement decouples the responsibilities of the memory instructions and computational instructions, providing a powerful lever to hide the memory access latency by overlapping long latency of memory access instructions with compute instructions.

2.5 ILP in Power7

Power7 is an out-of-order superscalar processor that can operate at frequencies exceeding 4 GHz. In a given clock cycle, a Power7 processor core can fetch up to eight instructions, decode and dispatch up to six instructions, issue and execute up to eight instructions, and commit up to six instructions. To ensure a high instruction throughput, Power7 can simultaneously maintain about 250 instructions in various stages of processing. To further extract independent instruction for parallelism, Power7 implements register renaming – each of the architected registers are mapped to a much larger set of physical registers. Execution of the instructions is carried by a total of 12 execution units. Power7 implements the Power ISA in a way that extracts high levels of instruction level parallelism while operating at a high clock frequency.

3 Data level parallelism

Data level parallelism (DLP) consists of simultaneously performing the same type of operations on different data values, using multiple functional units, with a single instruction. The most common approach of providing DLP in general purpose processors is the Single Instruction Multiple Data (SIMD) technique. SIMD (also called vector) instructions provide a concise and efficient way to express DLP. With SIMD instructions, fewer instructions are required to perform the same data computation resulting in lower fetch, decode and dispatch bandwidth, and consequently higher power efficiency.

Power ISA 2.06 contains two sets of SIMD instructions. The first one is the original set of instructions implemented by the vector facility since 1998 and also known as AltiVec [6] or Vector Media Extensions (VMX) instructions. The second is a new set of SIMD instructions called Vector-Scalar Extension (VSX).

3.1 VMX instructions

VMX instructions operate on 128-bit wide data, which can be vectors of byte (8-bit), half-word (16-bit) and word (32-bit) elements. The word elements can be either integer or single-precision floating point numbers. The VMX instructions follow the load/store model, with a 32-entry register set (each entry is 128-bit wide) that is separate from the original (scalar) fixed- and floating-point registers in Power ISA 2.06.

3.2 VSX instructions

VSX also operates on 128-bit wide data, which can be vectors of word (32-bit) and double word (64-bit) elements. Most operations are on floating-point numbers (single and double precision) but VSX also includes integer conversion and logical operations. VSX instructions also follow the load/store model, with a 64-entry register set (128 bits per entry) that overlaps the VMX and floating-point registers. VSX requires no operating-mode switches. Therefore, it is possible to interleave VSX instructions with floating point and integer instructions.

3.3 Power7 vector and scalar unit (VSU)

The vector and scalar unit of Power7 is responsible for execution of the VMX and VSX SIMD instructions. The unit contains one vector pipeline and four double-precision floating-point pipelines. A VSX floating-point instruction

uses two floating-point pipelines and two VSX instructions can be issued every cycle, to keep all floating-point pipelines busy. The four floating-point pipelines can each execute a double-precision fused multiply-add operation, leading to a performance of 8 flops/cycle for a Power7 core.

4 Thread level parallelism

Thread level parallelism (TLP) is the simultaneous execution of multiple threads of instructions. Unlike ILP and DLP, that rely on extracting parallelism from within the same program thread, TLP relies on explicit parallelism from multiple concurrently running threads. The multiple threads can come from the decomposition of a single program or from multiple independent programs.

4.1 Thread level parallelism within a processor core

In the first systems that exploited thread level parallelism, different threads executed on different processor cores and shared a memory system. Today, processor core designs have evolved such that multiple threads can run on single processor core. This increases resource utilization and, consequently, the computational throughput of the core. Effectively, TLP within a core enables hiding of the long latency events of stalled threads with forward progress of active threads. Examples of Power ISA processors that support multithreading within a core include Power5 [14], Power6 [10], and Power7 [9] processors.

4.2 Memory coherence models

For programs where the concurrent threads share memory while working on a common task, the memory consistency model of the architectures plays a key role in the performance of TLP as a function of the number of threads. The memory consistency model specifies how memory references from different threads can be interleaved. Power ISA specifies a *release consistency* memory model. A release consistency model relaxes the ordering of memory references as seen by different threads. When a particular ordering of memory references among threads is necessary for the program, explicit synchronization operations must be used.

4.3 TLP support in Power7 processor

The structure of a Power7 processor chip is shown in Figure 4. There are eight processor cores and three levels of cache in a single chip. Each processor core (which includes 32-Kbyte level 1 data and instruction caches) is paired with a 256-Kbyte level 2 (L2) cache that is private to the core. There is also a 32-Mbyte level 3 (L3) cache that is shared by all cores. The level 3 cache is organized as eight 4-Mbyte caches, each local to a core/L2 pair. Cast outs from an L2 cache can only go to its local L3 cache, but from there data can be cast out across the eight local L3s.

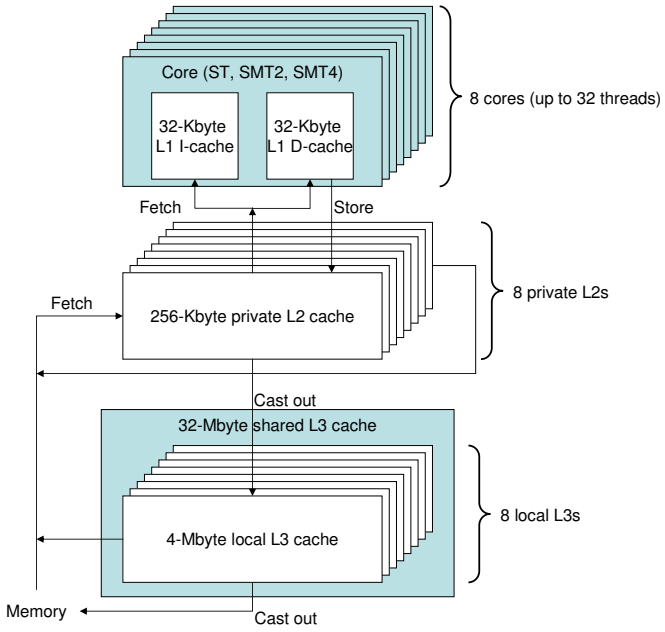

Figure 4: Structure of a Power7 processor chip.

Each core is capable of operating in three different threading modes: single-threaded (ST), dual-threaded (SMT2), or quad-threaded (SMT4). The cores can switch modes while executing, thus adapting to the needs of different applications. The ST mode delivers higher single-thread performance, since the resources of a core are dedicated to a the execution of that single thread. The SMT4 mode partitions the core resources among four threads, resulting in higher total throughput at the cost of reduced performance for each thread. The SMT2 mode is an intermediate point.

A single Power7 processor chip supports up to 32 simultaneous threads of execution (8 cores, 4 threads per core). Power7 scales to systems of 32

processor chips or up to 1024 threads of execution sharing a single memory image.

5 Summary

Since its inception in the Power1 processor in 1991, the Power architecture has evolved to address the technology and applications issues of the time. The ability of Power architecture to provide instruction, data and thread level parallelism has enabled a variety of parallel systems, including some notable supercomputers.

Power ISA allows exposing and extraction of ILP primarily because of the RISC principles embodied in the ISA. The reduced set of fixed length instructions enables simple hardware implementation that can be efficiently pipelined, thus increasing concurrency. The larger register set provides several optimization opportunities for the compiler as well as the hardware.

Power ISA provides facilities for data level parallelism via SIMD instructions. VMX and VSX instructions increase the computational efficiency of the processor by performing the same operation on multiple data values. For some programs DLP can be extracted automatically by the compiler. For others, explicit SIMD programming is more appropriate.

Power ISA supports thread level parallelism through a release consistency memory model. Because of the release consistency, Power ISA based systems permit aggressive software and hardware optimizations that would otherwise be restricted under a sequential consistency model.

The Power7 processors implements the latest version of Power ISA and exploits all forms of parallelism supported by the instruction set architecture: instruction level parallelism, data level parallelism and thread level parallelism.

RELATED ENTRIES

Blue Gene, PERCS, IBM SP, Cell Processor

BIBLIOGRAPHIC NOTES

Official information on the Power instruction set architecture is available in the POWER.ORG website[1]. In particular, the latest version of the Power ISA

[1] `www.power.org`

(2.06 revision B), implemented by the Power7 processor, can be found in [8].

An early history of the Power architecture is provided by Diefendorff [3]. Details of Power architecture including the instruction specification and programming environment are given in several reference manuals [7, 11, 12, 13].

Evolution of IBM's RISC philosophy is explained in [2]. More detailed information on the microarchitecture of specific Power processors can be found for Power4 [17], Power5 [14], Power6 [10], and Power7 [9] processors.

The AltiVec Programming Environment Manual [12] and AltiVec Programming Interface Manual [13] are two thorough references for effectively employing AltiVec. Gwennap [6] and Diefendorff [4] have a good survey of Power AltiVec.

Methods for extracting instruction level parallelism for the Power architecture are described in [7]. One of the key impediments to data level parallelism is unaligned memory accesses. To overcome these unaligned accesses Eichenberger, Wu and O'Brien [5] present some data level parallelism optimization techniques. Finally, Adve and Gharacharloo's tutorial on shared memory consistency model [1] is a great reference for further reading on thread level parallelism.

References

[1] Sarita V. Adve and Kourosh Gharachorloo. Shared memory consistency models: A tutorial. *IEEE Computer*, 29:66–76, 1996.

[2] John Cocke and Victoria Markstein. The evolution of RISC technology at IBM. *IBM Journal of Research and Development*, 40:48–55, 1990.

[3] Keith Diefendorff. History of the PowerPC architecture. *Commun. ACM*, 37(6):28–33, 1994.

[4] Keith Diefendorff, Pradeep Dubey, Ron Hochsprung, and Hunter Scales. Altivec extension to PowerPC accelerates media processing. *IEEE Micro*, pages 85–96, 2000.

[5] Alexandre E. Eichenberger, Peng Wu, and Kevin O'Brien. Vectorization for SIMD architectures with alignment constraints. *SIGPLAN Not.*, 39(6):82–93, 2004.

[6] Linley Gwennap. AltiVec vectorizes PowerPC. *Microprocessor Report*, 12(6):1–5, May 1998.

[7] Steve Hoxey, Faraydon Karim, Bill Hay, and Hank Warren, editors. *The PowerPC Compiler Writer's Guide*. Warthman Associates, 1996.

[8] IBM. *Power ISA Version 2.06 Revision B*. `http://www.power.org/resources/downloads/PowerISA_V2.06B_V2_PUBLIC.pdf`.

[9] Ron Kalla, Balaram Sinharoy, William J. Starke, and Michael Floyd. Power7: IBM's next-generation server processor. *IEEE Micro*, 30(2):7–15, March/April 2010.

[10] H. Q. Le, W. J. Starke, J. S. Fields, F. P. O'Connell, D. Q. Nguyen, B. J. Ronchetti, W. M. Sauer, E. M. Schwarz, and M. T. Vaden. IBM Power6 microarchitecture. *IBM J. Res. Dev.*, 51(6):639–662, 2007.

[11] Cathy May, Ed Silha, Rick Simpson, and Hank Warren, editors. *The PowerPC architecture: a specification for a new family of RISC processors*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1994.

[12] Freescale Semiconductor. *AltiVec Technology Programming Environments Manual*. 1999.

[13] Freescale Semiconductor. *AltiVec Technology Programming Interface Manual*. 1999.

[14] B. Sinharoy, R. N. Kalla, J. M. Tendler, R. J. Eickemeyer, and J. B. Joyner. Power5 system microarchitecture. *IBM J. Res. Dev.*, 49(4/5):505–521, 2005.

[15] James E. Smith and Gurinder S. Sohi. The microarchitecture of superscalar processors. In *In Proceedings of the IEEE*, volume 83, pages 1609–1624, 1995.

[16] Viji Srinivasan, David Brooks, Michael Gschwind, Pradip Bose, Victor Zyuban, Philip N. Strenski, and Philip G. Emma. Optimizing pipelines for power and performance. In *Proceedings of the 35th annual ACM/IEEE international symposium on Microarchitecture*, MICRO 35, pages 333–344, Los Alamitos, CA, USA, 2002. IEEE Computer Society Press.

[17] J. M. Tendler, J. S. Dodson, J. S. Fields, H. Le, and B. Sinharoy. Power4 system microarchitecture. *IBM J. Res. Dev.*, 46(1):5–25, 2002.