

Aaron Kobayashi
The IBM POWER4
10/8/2003

Assumptions:

One assumption made in this paper is that compilers and programmers can be made smart enough to take advantage of the high degree of parallelism offered by the POWER4 implementation. It also assumes that higher frequencies will perform better than lower frequencies that perform more work per instruction.

Summary:

IBM is a company that has been on the leading edge of technology for years. With their wide market presence, extensive resources, and experience they have a great deal of knowledge when it comes to building high performance infrastructures. These attributes shine through when examining the POWER4 microprocessor.

The POWER4 was an extension of previous research processors created at IBM and was designed to address both commercial and scientific computing. It achieves this goal by implementing many features such as SMP optimization, very high frequency, balanced technical and commercial performance, and binary compatibility with both 32 and 64-bit PowerPC applications. In addition to these goals, the POWER4 maintains a very high reliability by utilizing redundancy and error correction as well as transforming hard machine stops into interrupts to allow a processor to keep working if avoiding a problem is possible.

The POWER4 features two processors on a single chip which share a unified L2 cache (Eight-way, 128-byte line, 1.5MB) through a core interface unit (CIU) but maintain their own L1 cache (Two-way, 128-byte line, 64KB per processor). The CIU is a crossbar switch that feed 32 bytes of data per cycle to each CPU. L2 cache also implements an enhanced MESI coherency protocol in order to assist the processor when working in SMP systems. The processor also has special hardware responsible for handling instruction-serialization to maintain sanity when performing load and store operations. The L3 cache is implemented in a split fashion. The directory for the third-level cache and its controller are located directly on the POWER4 chip while the actual L3 data array is on two 16MB eDRAM chips mounted on a separate module. To support extendibility, a separate memory controller can be attached to the L3 module which allows the L3 cache to grow to a maximum of 128 MB.

While each POWER4 chip contains two processors, each individual processor is identical and has a superscalar out-of-order execution design enabling up to eight instructions to issue per cycle. These processors utilize traditional super-scalar techniques such as register-renaming, deep pipelining to enable high clock frequencies, and multiple execution units to support out-of-order execution. With any pipelined implementation, branch prediction schemes are used to minimize the cost associated with squashing improperly fetched instructions. The POWER4 uses three branch history tables (BHT) when predicting the taken direction. The first is a local predictor which indexes data based on the branch instruction address. The second is a global predictor that predicts based on the direction of the last 11 branches. The third predictor is called a

selector table. It keeps track of which of the previous schemes was more effective and selects the result of the more accurate predictor.

As previously mentioned, instructions are executed out of order. For a processor to be valid, it needs to remember the program order of all instructions currently in flight. To minimize the logic necessary, instructions are bundled into groups that are then executed together. Upon exception, the oldest group that has not been retired is restored. These groups are formed at the decode stage and contain up to five instructions. The oldest instruction is placed in slot 0 with newer instructions occupying the other slots. Slot four is reserved for branch instructions and is filled with a NOP if a branch instruction is not available. A group can complete (and modify the state of the processor) only when all older groups have completed and all instructions in the group have completed execution. On the front end of the processor, when a group is dispatched it first gets an entry in the group completion table (GCT). The GCT contains the address of the first instruction in the group and enables the processor to keep track of its current state. Once issued an entry into the GCT, a group is put in the issue queue slot where it waits to be passed on to the execution hardware.

The POWER4 architecture was designed with large multiprocessor systems in mind. Consequently the chip includes hardware for handling multiprocessor configurations. Four POWER4 chips can be packaged onto a single multichip module (MCM) to form an eight-way SMP. Four modules can then be interconnected to form a 32-way SMP. Inside a single MCM (four chips, eight-way SMP), each chip writes to its own bus, arbitrating between the L2, I/O controller, and L3 controller. While this is happening, each of the chips snoops all the buses for transactions to stay updated on events in the other processors. When interconnecting multiple MCMs, the intermodule buses act as repeaters that move requests and responses from one module to another in a ring like topology.

Great care was taken to insure that the POWER4 maintained scalability and balance while still offering great performance. As additional chips and MCMs are added to an SMP system, additional resources are added. Each addition expands the system by adding extra memory (adds extra memory cards), memory bandwidth, and L3 cache. Because the buses are pegged to half the processor speed, as the technology allows for higher clock speeds, the design can scale to utilize the added speed.

In summary the POWER4 is a new architecture that was designed to be a piece of a total system solution as opposed to just a processor. It has been designed to inherently support SMP while maintaining a scalable and balanced system architecture. Furthermore it provides a high frequency, highly reliable 64 bit environment that maintains binary compatibility with both 32-bit and 64-bit PowerPC applications.

Test Question:

In the POWER4 architecture, why are instructions grouped together in the decode stage? Also describe an instruction group's structure.

Solution (Pg10, last paragraph):

In the POWER4 architecture, instruction groups are implemented to minimize the amount of logic necessary to have an out of order execution engine. A group contains up to five instructions with the oldest instruction in slot 0, the next oldest

in slot 1 and so on. Slot 4 is reserved for branch instructions and can never have any other instruction (excluding nop) occupying its space.