IBM's Power3 to Replace P2SC

Power3 to Complete IBM's Transition to PowerPC Architecture

by Peter Song

To complete its transition from POWER to PowerPC, IBM has announced the Power3 processor, a 64-bit PowerPC chip that will replace the P2SC (see MPR 8/26/96, p. 14) in its RS/6000 workstations and servers in 2H98. At last month's Microprocessor Forum, Mark Papermaster of IBM Microelectronics described IBM's latest design, which has its heritage in both the ill-fated PowerPC 620 and P2SC. As the P2SC runs out of gas, reaching only 160 MHz in a 0.25-micron process, IBM plans to replenish its RS/6000 product lines with Power3 chips, which will eventually reach 500 MHz in more advanced processes.

Executing two multiply-add operations in each cycle, Power3 maintains the emphasis on floating-point performance that has become the trademark of high-end RS/6000 systems. Executing two integer and two load-store operations—twice the rate of the P2SC—in each cycle, it also improves that chip's poor SPECint95 performance by 35% at equivalent clock speeds. Although its 64K data cache is only half the size of the P2SC's, its advanced core, a dedicated L2 cache, and aggressive prefetching mechanisms more than make up for the smaller cache.

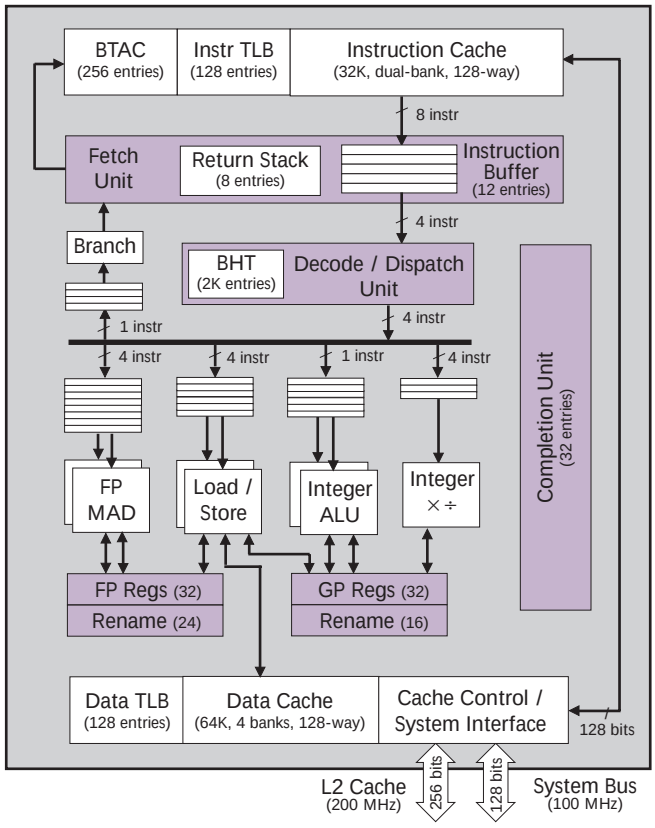

Figure 1. Power3 can execute two load and two integer or floating-point instructions in each cycle.

Power3 is the processor that will complete IBM's transition from the POWER to the PowerPC architecture. Among its many advantages, the PowerPC architecture offers better multiprocessor scalability, while enabling simpler designs for high-end systems than possible with the POWER architecture. It also brings a 64-bit instruction set to a growing segment of multiprocessor servers that demand huge amounts of memory. Its 64-bit format will become even more useful when multimedia instruction-set extensions, which the PowerPC alliance has been quietly developing, are added to PowerPC processors.

Modified PowerPC 620 Core

The Power3 design team started with the largely finished 620 design (see MPR 10/24/94, p. 12) for the 64-bit PowerPC core, backside cache interface, and PowerPC 6xx bus. The team added the second floating-point and load-store units, as Figure 1 shows, making Power3 better suited for applications that combine floating-point math with large data sets. The team also added more queues and rename registers, increasing to 32 the number of instructions that can be in process. The two additional execution units give Power3 a peak execution rate of eight instructions per cycle. The team reduced the sustainable execution rate from the P2SC's six instructions per cycle to four per cycle, limited by the decode/dispatch unit, but the reduction is unlikely to affect overall performance.

The team made significantly more changes to the 620 chip's memory subsystem to sustain two loads and two floating-point operations per cycle. The team doubled the data-cache size to 64K, improving the cache-hit rate. The cache is

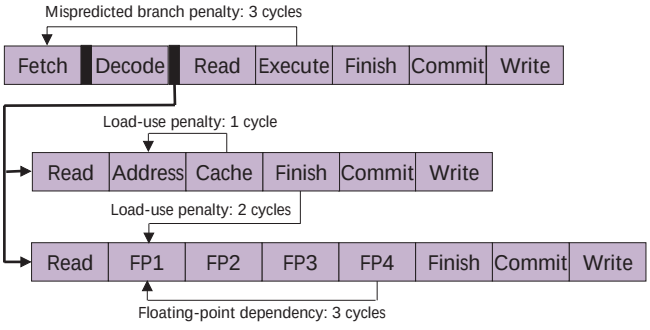

Figure 2. Power3's short pipeline keeps its branch-misprediction penalty to three cycles. Instructions incur possible delays, shown with black bars, in the instruction buffer and the issue queues.

interleaved into eight banks on double-word boundaries, reducing conflicts among the two loads and a store that can access the cache in each cycle. Taking advantage of the greater localities found in large engineering and scientific data sets, the team doubled the line size to 128 bytes, increasing the L2 interface to 32 bytes wide to boost bandwidth and accommodate the larger line size.

Power3 Uses a Short Pipeline

Unlike competitive chips such as Digital's 21264 (see MPR 10/28/96, p. 11) or HP's PA-8500 (see MPR 11/17/97, p. 20), which use five pipeline stages before instructions enter the first execution stage, Power3 keeps this front end of the pipeline short, using only three stages. Taking advantage of its lower clock speed, Power3 needs only one cycle to access the instruction cache, one cycle to decode and dispatch the instructions to different execution units, and one more cycle to access the operands, as Figure 2 shows. Power3's relatively short pipeline keeps its mispredicted branch penalty to only three cycles, 2–4 cycles shorter than its competitors'.

Up to eight instructions are fetched from any position within a 128-byte cache line and placed into the 16-entry instruction buffer. The large line size and the word-aligned cache access increase the frequency of fetching eight useful instructions in each cycle. The instruction cache consists of two banks of fully-associative arrays, yielding 128 sets in each array. The dual-bank design supports both instruction fetch and cache reload to different banks in the same cycle. All 128 bytes of a cache line are updated in a single cycle, further reducing the number of cycles the cache is unavailable for instruction fetch.

MICHAEL MUSTACCHI

Mark Papermaster describes Power3's short pipeline that keeps branch penalty low.

In each cycle, up to four instructions in the instruction buffer are decoded and dispatched in program order to the execution units. Except for the branch and multicycle-integer units, which can accept at most one instruction per cycle, each execution unit can accept up to four instructions per cycle, provided that the unit has enough available slots in its issue queue. The instructions within the buffer are shifted by zero, two, or four slots, ensuring that at least three instructions are available for dispatch in each cycle. Supporting all possible shift options would have doubled the number of inputs to each entry.

Up to eight instructions—two floating-point, two load/store, two single-cycle integer, a multicycle integer, and a branch—can begin execution in each cycle. Ready instructions are issued out of order from the issue queues, allowing instructions of different types, as well as of the same type, to execute out of order. The load/store and branch instructions are issued in program order, simplifying the design. Unlike the 620's reservation stations, Power3's issue queues do not store the operands, requiring an additional cycle to access the operands in most cases. When the queues and the execution pipelines are empty, such as when the machine is recovering from a mispredicted branch or processing an exception, instructions read the operands during the dispatch cycle, bypassing the issue stage.

When an instruction finishes execution, its result is written to a rename register during the finish stage. The instruction's execution status—that it finished execution and did or did not detect an exception—is written to the completion buffer. Up to four instructions can be retired in the commit stage. The results from retired instructions are copied from the rename registers to the architectural registers in the write stage. Because execution results are made available in the rename buffers as soon as they are produced, the last three pipeline stages have little effect on performance as long as enough rename registers are available.

Only a Few Branches Are Predicted

The PowerPC branch model allows Power3 to execute branch instructions as early as the decode stage, reducing the number of branch instructions that are predicted. In contrast to the *compare-and-branch* model used in the PA-RISC and MIPS architectures, the POWER and PowerPC architectures support the *condition-code-and-branch* model. This model separates the instructions that generate the branch condition from the instructions that change the program flow, allowing the branch condition to be generated in advance. In addition, the POWER and PowerPC architectures provide eight sets of condition codes, allowing multiple branch conditions to be precomputed.

For a taken branch, the branch target address is also needed to redirect the fetch stream. In the decode stage, the program-counter-relative branch instructions readily provide the target address, but the register-direct branch instructions do not. Instead of a general-purpose register, these branch instructions in the POWER and PowerPC architectures use one of two special registers (LINK and COUNT registers). Because these special registers are much smaller than the register file and the instructions that update them are easily detected, they are faster and easier to access than a general-purpose register, especially in superscalar designs.

Many POWER and PowerPC designs take advantage of these features to execute branch instructions as soon as the branch conditions are known. In Power3, each group of 4-bit condition codes is mapped to a rename register, allowing speculative and out-of-order generation of multiple branch conditions. Most instructions that modify the LINK or COUNT registers, such as BRANCH-AND-LINK, DECREMENT-AND-BRANCH, or MOVE-TO-LINK, can also execute speculatively. These special

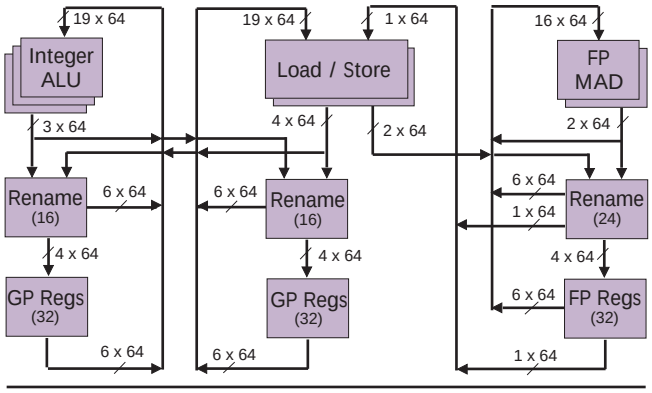

Figure 3. Power3's core requires many 64-bit buses to read operands and return execution results. The number of operand-read ports is reduced from 12 to 6 by using two sets of registers.

registers, along with their rename registers, are made accessible for executing a branch instruction in the decode stage.

BTAC Predicts Branch Condition, Target Address

For branch instructions whose conditions are not known in the decode stage, Power3 uses a 2,048-entry BHT (branch history table) to predict their branch direction. Because a branch is often resolved in the decode stage or soon thereafter, the benefit of the BHT when used to predict the current encounter of the branch is less in Power3 than in designs with deeper pipelines. To better utilize the BHT, however, Power3 uses the BHT to predict both the current and the next encounter of each conditional branch, using a BTAC (branch target address cache).

A BTAC is generally used to predict the target address of a predicted-taken or, in some designs, a known-taken branch instruction. It avoids a register-file access and an address calculation in many designs. Its prediction is highly accurate because branch instructions tend to jump repeatedly to the same target instruction, even in object-oriented programs that use dynamic binding.

In Power3, the 256-entry BTAC is used both to predict the target address of a branch and to predict the branch direction (taken or not-taken). By keeping only predicted-taken branch instructions in the BTAC, a branch that misses the BTAC is naturally predicted not taken. As each conditional branch is resolved in the execute stage, the instruction is added to the BTAC if its two-bit branch history changes from not taken to taken, effectively predicting the branch direction for its next encounter. Conversely, the instruction is deleted from the BTAC if its history changes from taken to not taken.

The BTAC is accessed in the fetch stage, resulting in a zero-cycle delay for taken-branch instructions when correctly predicted. It is accessed with the fetch address, not with the specific address of a branch, since the address of the branch instruction is unknown. In effect, the BTAC is accessed in case one of the eight instructions being fetched is a predicted-taken branch. When the access hits the BTAC—indicating that one of the instructions being fetched is a predicted-taken branch—the returned target address is used for accessing the instruction cache and the BTAC in the next cycle.

	Power3 (200 MHz)	PA-8500 (360 MHz)	21264 (600 MHz)
64-bit Integer Mul	9 cycles	n/a	7 cycles
64-bit Integer Divide	37 cycles	n/a	>64 cycles
FP Mul-Add (DP)	4 cycle	3 cycles	4 + 4 cycles*
FP Add/Mul (DP)	4 cycle	3 cycles	4 cycles
FP Divide (SP)	14 cycles	18 cycles	12 cycles
FP Square Root (SP)	14 cycles	18 cycles	16 cycles
FP Divide (DP)	18 cycles	32 cycles	16 cycles
FP Square Root (DP)	22 cycles	32 cycles	33 cycles

Table 1. Power3's execution latencies are comparable to those of its competitors but at a lower clock speed. n/a = not applicable. *using multiply and add instructions. (Source: vendors)

Other processors provide a zero-cycle delay for taken-branch instructions when correctly predicted. Instead of using a BTAC, the 21264 keeps a 12-bit field for each 32-byte line, providing the next-line and next-set prediction for each instruction-cache access. This design uses fewer bits per branch instruction than does Power3, since it does not need a separate tag array and uses only a 12-bit index instead of the virtual addresses for the branch and the target instructions.

Power3 Sustains Four Instructions Per Cycle

Power3 uses rename registers for the GPRs (general-purpose registers), FPRs (floating-point registers), and the CCR (condition-code register) to allow out-of-order and speculative execution of most instructions. The few exceptions are stores and certain move-to-special-register instructions that are difficult to undo. As instructions are dispatched to the issue queues in program order, their destination registers are mapped to the rename registers, allowing out-of-order access to the registers. Although instructions can be issued out of order from the issue queues and thus their operands can be read out of order from the register files, the rename registers eliminate anti- and output-dependency hazards by enabling the register files to be updated in program order.

As Figure 3 shows, Power3 uses two copies of the GPRs and the general-purpose rename registers, reducing the number of read ports needed to support the five execution units from 12 to 6. One copy provides two operands to each of the three integer units, and the second copy provides three operands—two for the address operands and one for the store data—to each of the two load-store units. The 21264 also uses two copies of integer registers but assigns a copy to a pair of load-store and integer execution units. Due to its short cycle time, the 21264 requires an extra cycle to transmit a result from one pair to the other. Operating at a lower speed, Power3 does not, simplifying the dispatch logic.

The rename registers have four more read ports for updating the GPRs with the results of retired instructions, matching the sustainable execution rate of four instructions per cycle. The rename registers also have seven write ports:

one from each of the integer units and two from each of the load-store units. The second write port for the load-store units allows them to execute a LOAD-WITH-UPDATE instruction, which returns both load data and an updated address value in each cycle.

The 21264 can execute two floating-point instructions per cycle, but only when one is a multiply and the other is an add. In contrast, Power3 has two identical FPUs, delivering twice as many floating-point results per cycle as the 21264. Power3's FPUs execute multiply-add instructions, as Table 1 shows, taking only half the number of cycles as the 21264 to calculate the common $A \times B + C$ operation. At only a third of the Alpha chip's 600-MHz clock speed, however, Power3 is unlikely to deliver better overall floating-point performance. HP's PA-8x00 can also execute two multiply-accumulate instructions in each cycle but takes only three cycles to produce the results.

Hardware Prefetch Reduces Cache Misses

Power3 uses instruction- and data-prefetch mechanisms to reduce pipeline stalls due to cache misses. The instruction cache is two-way interleaved on cache-line boundaries, allowing one bank to be accessed for instruction fetches while the other bank is accessed for the next cache line. When the former access hits in the cache but the latter access does not, a prefetch request for this next cache line is issued to the L2 cache. Because the prefetch is still speculative, the request is not propagated to the main memory if it misses in the L2 cache, allowing the request to be canceled upon detecting a mispredicted branch instruction. An instruction prefetch takes six cycles from the 200-MHz L2 cache.

For the data cache, a prefetch mechanism keeps track of up to four prefetch streams and, for each stream, fetches up to two cache lines ahead from the L2 cache or main memory. It identifies a prefetch stream only if the accesses in the stream advance from one 128-byte block of memory to the adjacent—sequentially succeeding or preceding—block of memory. The prefetch mechanism is therefore restricted to a stride of one cache line.

To establish a prefetch stream, the prefetch mechanism monitors every access that misses in the data cache, searching for cache-miss references to two adjacent 128-byte blocks of memory. Upon finding such a pair, it initiates a prefetch request for the next 128-byte block. The address of the prefetch request, along with the ascending or descending prefetch direction, is kept in a four-entry prefetch queue. Once a prefetch stream is identified, the address of every data-cache access is checked with the addresses in the prefetch queue. When a match is found, a prefetch request for the next 128-byte block is made, and the address in the matching entry is updated with the address of the new prefetch request.

Using this prefetch mechanism, Power3 executes the inner loop of DAXPY benchmark at the rate of 27 cycles per iteration, according to IBM. This loop contains 32 load and 16 store operations, which require 26 cycles for Power3 to execute, assuming no data-cache misses. With the prefetch mechanism disabled, Power3 executes the same loop in 36 cycles.

The Power3's prefetch mechanism is likely to be less effective, however, in programs that are not as well behaved as DAXPY. Requiring two cache-miss references to identify a prefetch stream and limiting the prefetch stride to 128 bytes, the mechanism's effectiveness depends largely on carefully organizing the data sets. UltraSparc-3's prefetch cache (see MPR 10/27/97, p. 29), in contrast, places no restriction on the prefetch stride, allowing complete freedom in organizing the data sets. In addition, the prefetch cache relies on software to identify the load instructions whose data should be prefetched, simplifying the prefetch hardware while incurring fewer cache-miss references than Power3.

Chip's Interfaces Cap CPU Speeds at 200 MHz

Power3 is built initially in IBM's CMOS-6S2 process, a hybrid of 0.25-micron transistors with 0.35-micron metal layers (see MPR 9/16/96, p. 11), and occupies a 270-mm^2 die. It uses the same 1,088-pin ceramic package as the P2SC but a different pinout. The MDR Cost Model estimates the manufacturing cost of the Power3 chips to be $160, only 40–60% of the competitors'. For these nonmerchant chips, however, the cost advantage provides no real benefit.

According to IBM's Papermaster, current Power3 chips are functional, running at an excess of 200 MHz, and will be available in systems in 2H98. Operating the cache and system interfaces at the full speed and half the speed of the processor, respectively, the 200-MHz Power3 chips deliver 6.4 Gbytes/s of cache bandwidth and 1.6 Gbytes/s of memory bandwidth. At 200 MHz, IBM estimates Power3 will deliver 28 SPECfp95 and 12 SPECint95 (base). By the time Power3 ships, it will compete against the 21264 and PA-8500, which are slated to deliver 30–40 SPECint95 and 50–60 SPECfp95 (base).

Although current Power3 chips may operate at speeds faster than 200 MHz, Power3 systems are unlikely to be offered at the faster speeds, because the current design does not support fractional processor-to-cache and processor-to-system clock ratios (i.e., 3:2 mode). Faster Power3 chips require either using correspondingly faster SRAMs in 1:1 mode or operating the cache interface in 2:1 mode, half the speed of the processor. The former solution is expensive and feasible at speeds only up to 250 MHz, while the latter delivers lower performance at speeds below 400 MHz. At speeds between 200 and 250 MHz, the system interface presents a dilemma that conflicts with the cache interface; it can no longer operate in 2:1 mode but degrades performance in 3:1 mode.

IBM is already working to remedy the situation, first by migrating the Power3 design to its CMOS-7S process (see MPR 8/4/97, p. 14). In this 0.2-micron process, which also uses copper interconnects, IBM expects first silicon of 350-MHz Power3 processors in 2Q98. The die size will shrink to 160 mm^2, with a few additional functions. Although the process will be in production by 2H98, the belated design will

Prices & Availability

Power3 will not be sold as a standalone product. Systems using Power3 will ship in 2H98. For more information, contact your local IBM sales office or the IBM RS/6000 Web site *www.austin.ibm.com/hardware*.

keep faster Power3 chips unavailable in systems until 1H99. In 2H99, IBM plans a second derivative of Power3 in a true 0.18-micron process, hoping to reach speeds up to 500 MHz.

The faster Power3 chips will support fractional bus modes—5:2 and 7:2 for processor-to-bus and 3:2 for processor-to-cache interfaces—which will allow the core to run at its full speed. Using a set-prediction mechanism, the new chips will also support a four-way set-associative L2 cache without using additional pins for the data bus, similar to the two-way set prediction used in SGI's R10000 (see MPR 10/24/94, p. 18). To select the correct set in a single cycle when the prediction is wrong, however, the chips will have 100 additional pins to read all four tags simultaneously.

We believe bringing the L2 tag array on chip would offer performance advantages over adding more pins to support a better L2 cache. In addition to better supporting a set-associative L2 cache, the on-chip tag array would also reduce the L2-miss and snoop-response latency cycles. Because the array could be clocked at the CPU speed, not the L2-cache speed, it also offers more bandwidth for the CPU and the coherency traffic, making the design more suitable in large-scale MP configurations. The tag array for a 16M L2 cache would be comparable in size to the current Power3's 64K data cache, which occupies about 27 mm^2 (10% of the die size) in the current design. Since SRAMs are 2.3× denser in CMOS-7S than in CMOS-6S2, the array would occupy only about 11 mm^2 in the faster Power3, a reasonable area overhead for gaining the extra performance.

Power3 Completes IBM's Transition to PowerPC

Due to its "brainiac" design (see MPR 3/8/93, p. 3), Power3 operates at much lower speeds than its competitors, such as the PA-8500 and 21264 processors. These competitors are likely to operate at 1.5–3× the speeds of Power3, using a comparable process. Even with 30–50% fewer logic transistors, they can also process more instructions out of order than does Power3.

As Figure 4 shows, the Power3's caches occupy only 15% of the die area. In comparison, the 1.5M caches in the PA-8500 occupy nearly 60% of the die area. Within the Power3's heritage, the percentage of total transistors used for logic has steadily increased—from 20% in Power2 to 38% in the P2SC to now 47% in Power3. With wire delays taking an increasingly large portion of cycle times as process dimensions shrink, using more of the available transistors in logic is likely to make the wire-delay problem worse. Considering

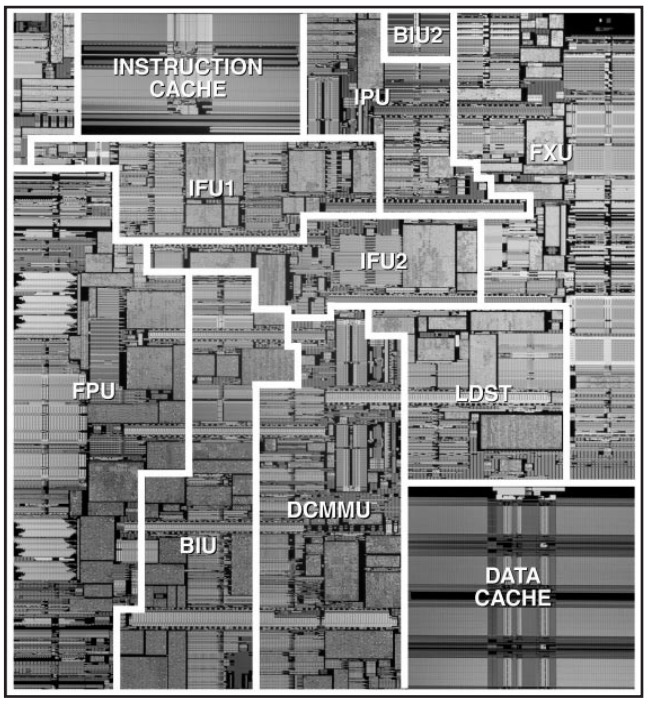

Figure 4. The Power3's two caches occupy only 15% of the 270-mm^2 die, which is built in a hybrid 0.25-micron process. In comparison, the 1.5M caches in the PA-8500 occupy 60% of its die area.

that migrating the P2SC from the 0.29-micron CMOS-6S to CMOS-6S2 increases its speed by only 20%, IBM is likely to face tough challenges in delivering Power3 derivatives that meet their clock-speed goals.

Higher clock speed is not the same as higher performance, however, especially for many server applications that have low hit rates for the on-chip caches. For these cache-busting applications, hiding memory latency and providing sufficient memory bandwidth is as important as high clock speed, a point the Power3 designers did not miss. For two processors that deliver comparable performance at different clock speeds, the processor with a lower clock speed should tolerate cache misses better, due to fewer memory-latency cycles. The ability to reorder a large number of instructions helps, but that ability cannot completely hide memory latencies that range up to more than a hundred cycles.

When IBM replaces the P2SC processor with Power3 in the RS/6000 product lines, it will complete the long-awaited transition from the POWER to the PowerPC instruction set, establishing a single architecture for IBM's entire computer product lines (except legacy mainframes and PCs). The transition is already completed in the AS/400 minicomputers, which use the 64-bit PowerPC A10 and A30 processors (see MPR 7/31/95, p. 15). Although many RS/6000 workstations and servers are already powered by PowerPC processors, the high-end servers still use the P2SC for its superior floating-point performance. With Power3 in high-end server systems, IBM will fulfill its vision of providing "palmtops to teraflops" using a single architecture. M