

MICROPROCESSOR REPORT

THE INSIDERS' GUIDE TO MICROPROCESSOR HARDWARE

VOLUME 7 NUMBER 13

OCTOBER 4, 1993

IBM Regains Performance Lead with Power2

Six-Way Superscalar CPU In MCM Achieves 126 SPECint92

by Linley Gwennap

In the ongoing game of performance leapfrog, IBM has sprung into the lead with its Power2 chip set. The new POWER processor is rated at 126 SPECint92 and 260 SPECfp92 at its top speed of 71.5 MHz. The integer rating edges Digital's 200-MHz 21064, the previous performance leader, and the FP rating gives Power2 a clear advantage. It is interesting for its exploitation of advanced design techniques, many of which will eventually appear in merchant-market, single-chip processors.

The new design is an extension of Power1 (see MPR 8/21/91, p. 10), which is also known as RIOS. IBM has retained the same chip partitioning, the same pipeline, the same buses, and the same basic memory structure from Power1. The major changes involve quadrupling the size of most memory elements, widening the buses, and doubling the number of functional units. The wider buses and new dual-ported data cache attempt to maintain enough memory bandwidth to feed the larger array of functional units.

Power2 supports the highest execution rate of any RISC processor: up to six instructions per cycle. This peak rate, however, requires a careful mix of instructions: two integer, two floating-point, and two branch or condition-code instructions. (POWER evaluates conditional branches based on preset condition codes; see MPR 9/4/91, p. 8.) All loads and stores, including those to the FP registers, count as integer instructions, but the chip set does support two loads or stores per cycle using a fully dual-ported cache subsystem.

The downside of all this complexity is its impact on clock frequency: at 71.5 MHz, Power2's clock is much slower than single-chip processors from Digital, MIPS, and HP. In fact, the new CPU's clock is not much faster than Power1, which tops out at 62.5 MHz. Even that comparison is not really fair because Power2 requires a multichip module (MCM) package to reach 71.5 MHz; it would be slower than Power1 if packaged discretely.

As a result of the multichip design and the costly

packaging, Power2 will be expensive to manufacture, even though the individual dice are relatively small. Its impressive performance will improve the competitiveness of IBM's RS/6000 workstation line (see sidebar below), which had been lagging HP and Digital in performance. The particularly good floating-point numbers also make Power2 well-suited for IBM's PowerParallel line, which is used for high-end scientific applications.

Architectural Extensions

For Power2, IBM has added a few extensions to the original POWER architecture. The most important are quad-word load and store instructions, which reference a 128-bit value in memory. Since the FP registers are 64 bits wide, the quad-word instructions affect a pair of adjacent registers. These instructions, and Power2's ability to execute them in a single cycle, nearly double the performance of programs such as Linpack that are limited by the cache bandwidth.

Power2 also adds a hardware square root instruction and two new instructions that provide a more efficient conversion of floating-point values to integers. The processor uses a new page-table format that improves the speed of the hardware table-walking algorithm used to resolve a TLB miss. The chip set also includes a set of registers for performance monitoring.

Instruction Dispatch

As shown in Figure 1, Power2 uses three processor chips: the instruction cache unit (ICU), the fixed-point unit (FXU), and the floating-point unit (FPU). The data cache consists of four data-cache unit (DCU) chips. Power1 allows for a two-DCU configuration for lower-cost systems; it is likely that this feature is available in Power2 as well, although IBM would not confirm it. The system-control unit (SCU) controls the main memory and peripherals. All chip-to-chip buses are protected by parity or ECC.

The ICU contains a 32K instruction cache and a 128-entry instruction TLB partitioned as two sets. Both

are similar in structure to the corresponding units in the original Power1; the size of each is quadrupled.

The instruction cache is organized as eight individually addressable arrays, each 40 bits in width. This design can deliver eight consecutive instructions per cycle even across cache line boundaries. Each array is wide enough for a single 32-bit instruction, one parity bit, and seven predecode bits that are calculated when instructions are loaded into the cache. These bits indicate the instruction type (FXU, FPU, branch, or condition code) and assist the dispatcher.

Instructions are first delivered into a prefetch buffer, which holds up to 16 instructions. The dispatcher then fully decodes the first six available instructions. Any combination of up to four FXU/FPU instructions can be dispatched onto the instruction bus each cycle. The ICU can also issue up to two instructions to its internal units: either two branches or one branch and one instruction that modifies the condition codes.

The dispatcher does not issue all six instructions if these limits are exceeded. If either the FXU or FPU instruction queues are full, these instruction types cannot be issued. The dispatcher does not look for data dependencies; these are handled within the FXU and FPU. Instructions are always dispatched in order. The dispatch logic uses a large portion of the ICU due to the number of decode units and wide multiplexers required to route all these instructions to the correct function units with no alignment restrictions.

Branch Processing

The branch units in the ICU prevent the arithmetic chips from ever seeing branch instructions; IBM calls this feature, first implemented in Power1, "zero-cycle branching." While this is generally true for unconditional branches, conditional branches can have up to a three-cycle delay in Power1 if the compare instruction is placed immediately before the branch.

Power2 takes advantage of its excess instruction fetch bandwidth by fetching the first portion of the target stream before the branch condition is resolved. Figure 2 shows how this works. In the cycle immediately following the dispatch of the branch, the target instructions are fetched while the ICU continues to dispatch buffered instructions from the sequential stream. The ICU contains an additional four decode units devoted to the target instructions, allowing them to be dispatched during cycle 3 in the example. Having separate decoders simplifies the control logic and reduces the cycle time.

By cycle 3, the compare instruction has reached the FXU and is resolved. If the branch is not taken, the target instructions that were dispatched are removed from the instruction queues before they are executed. If the branch is taken, some of the sequential instructions must be cancelled, causing a one-cycle penalty. This penalty can be avoided entirely by placing the compare instruction at least one cycle ahead of the branch. Even one cycle, however, represents a potential loss of six issue

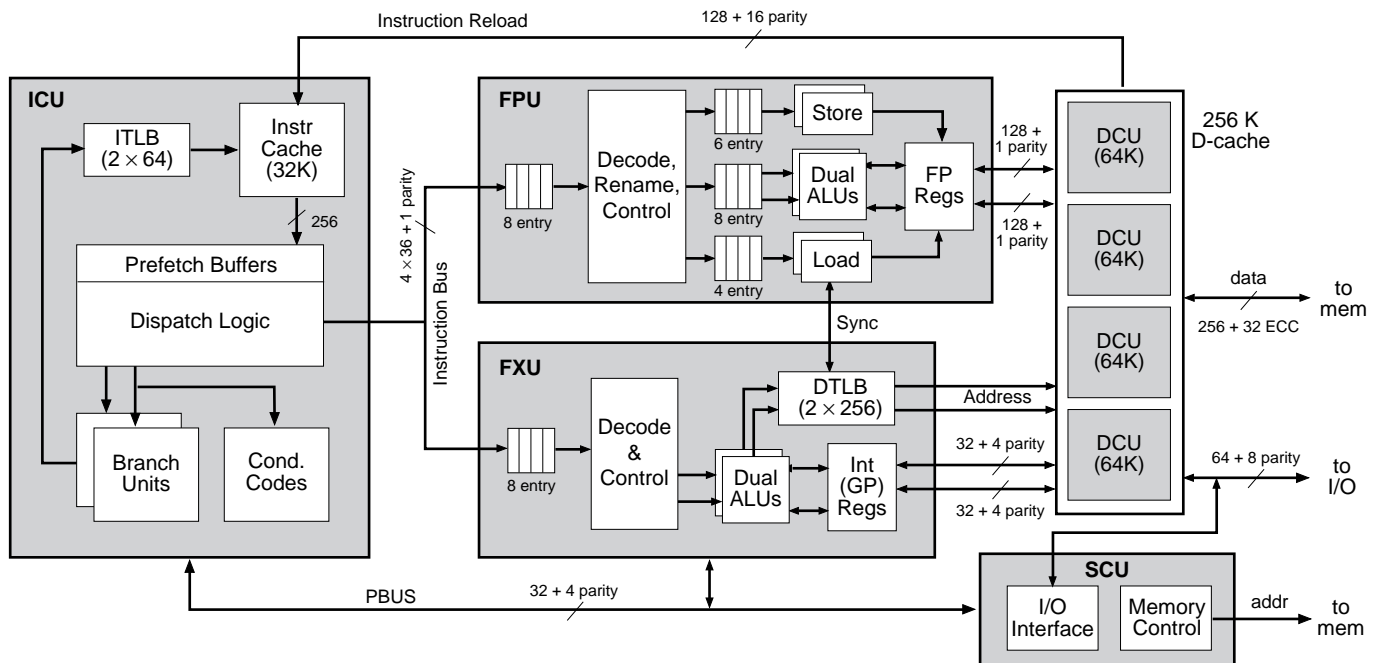


Figure 1. The Power2 design includes eight primary chips. The instruction-cache unit (ICU) handles instruction dispatch, while the fixed-point unit (FXU) and floating-point unit (FPU) handle integer and FP math, respectively. The data-cache units (DCU) provide data storage, while the system control unit (SCU) connects the processor to memory and I/O.

slots, but the Power2 designers chose not to add branch prediction to reduce the number of mispredictions.

With dual branch units, Power2 can execute up to two branches per cycle. Only one set of instructions can be dispatched conditionally, however, so two branches can be handled only if the first branch is known to be not taken when it is encountered (because its condition code is already determined). The ICU also has two partial decode units that look for branches in the seventh and eighth instruction slots (the first six instructions having been fully decoded) and can dispatch them into the branch units. These additional decode units help keep the prefetch buffers full.

Decoupled Execution Units

Power2, like its predecessor, uses instruction queues to decouple the integer and FP execution units from the dispatch stream, as shown in Figure 1. For the new processor, these FIFO queues have been extended to eight entries each for the FXU and FPU. The queues allow the ICU to dispatch up to four integer or four FP instructions at a time, even though the execution units can handle only two of each per cycle. This flexibility maximizes the throughput of the instruction dispatch bus.

All integer instructions are received and decoded by the FXU, which contains two general-purpose ALUs. One has a standard two-input adder and a shifter; the other includes a three-input adder, a shifter, and a 36×36 multiply/divide unit. The FXU checks for data dependencies between the two instructions and executes them sequentially if needed. Integer instructions are always issued in order, maintaining a precise exception model.

There is one special case in which dependent execution is permitted: the three-input adder allows a sequence of two instructions such as:

$$R1 + R2 \rightarrow R3; R3 + R4 \rightarrow R5$$

to be executed in a single cycle. In this case, the two-input adder updates R3, while the three-input adder calculates the sum of R1, R2, and R4. Similarly, two load-with-update instructions that modify the same address register can also be executed in the same cycle. IBM says that its performance simulations showed that these sequences occur frequently enough to justify the extra die area of the three-input adder.

The dedicated multiplier takes two cycles for any type of integer multiplication. The FXU takes advantage of this multiplier by using a modified Goldschmidt division algorithm that requires a table lookup, five multiplies, and two subtractions for a total of 13–14 cycles in most cases. During this latency, other integer instructions cannot be executed due to the in-order restriction.

The ALUs are also used for address calculations on loads and stores to either the integer or FP registers. Thus, these instructions count as integer instructions for all instruction dispatch purposes. The POWER architec-

CYCLE →	1	2	3	4	5	6
FETCH	Sequential	Target	Seq.	Seq.	Seq.	Seq.
DISPATCH	CMP BR C1 C2 C3	C4 C5 C6 C7	T1 T2 T3 T4	C8 C9 C10 C11		
DECODE		CMP C1	C2 C3	C4 C5	C6 C7	
EXECUTE			CMP C1	C2 C3	C4 C5	C6 C7

Branch Not Taken Timing

FXU resolves branch condition

CYCLE →	1	2	3	4	5	6
FETCH	Sequential	Target	Seq.	Seq.	Seq.	Seq.
DISPATCH	CMP BR C1 C2 C3	C4 C5 C6 C7	T1 T2 T3 T4			
DECODE		CMP C1	C2 C3	T1 T2	T3 T4	
EXECUTE			CMP	X	T1 T2	T3 T4

Branch Taken Timing

FXU resolves branch condition

Figure 2. On an unresolved conditional branch (BR), instructions are dispatched first from the sequential stream (C1...C8), and then from the target stream (T1...T4). Once the branch condition (CMP) is resolved, execution can continue with no delay for untaken branches and only a one-cycle delay for taken branches.

ture includes CISC-style, multi-cycle string load-and-compare instructions; these operations use both ALUs in parallel to improve performance.

FPU Increases Parallel Execution

The floating-point chip has been significantly redesigned to increase the amount of parallel instruction execution. The dual FP ALUs can each handle any FP arithmetic instruction. The FPU also includes two load and two stores units, although the requirement of using the integer ALUs for address calculation limits the processor to two load/store instructions per cycle (in any combination). The store units normalize data before placing it in the data cache; in Power1, this was done by the ALU, reducing throughput.

Each of the three pairs of function units is buffered by its own instruction queue, as shown in Figure 1. This allows loads, stores, and arithmetic instructions to execute out of order if there are no data dependencies. The FPU contains complex logic to detect and resolve data dependencies.

In addition to these three queues, each FP ALU has a single-entry buffer that allows arithmetic instructions to execute out of order. For example, in the sequence:

$$R1 \div R2 \rightarrow R3; R3 + R4 \rightarrow R5; R1 \times R4 \rightarrow R6$$

the add instruction is dependent on the long-latency divide operation and thus must be held in the ALU's buffer. The multiply, however, and other non-dependent instructions can continue to be issued to the second ALU during the divide latency. A second dependent arithmetic instruction, however, would lock up the FPU until

New Power2 Systems

IBM has announced three RS/6000 systems using its new Power2 processor. The Model 58H runs the processor at 55 MHz, while the Model 590 hits 66 MHz. These two deskside systems support up to 2G of memory and 12G of disk. Both systems include a CD-ROM drive and seven Micro Channel slots. The base price of the 58H is \$62,500, and the Model 590 starts at \$72,500. These prices include 64M of memory and 2G of disk.

The Model 990, the most powerful of the new systems, pushes the clock rate up to 71.5 MHz. This rack-mounted system increases disk capacity to 74G, or as much as 840G using RAID disk arrays. It also includes a CD-ROM plus two Micro Channel buses with a total of 15 available slots. The base price is \$124,500.

The performance of these systems scales well with the clock frequency. The Model 58H is rated at 98 SPECint92 and 204 SPECfp92; the Model 590 is rated at 117 and 242; the Model 990 achieves 126 and 260.

All of these systems use the Power2 multichip module. IBM plans to put Power2 into its Series 300 desktop line by mid-1994, but these products will probably use Power2 chips packaged in ball-grid arrays. The BGA package reduces cost but may be limited to 50–60 MHz. At the other end of the scale, Power2 will also be deployed in IBM's PowerParallel system, which currently supports up to 64 processors. This product will probably use the MCM version to achieve maximum performance.

the divide completes.

Because of the instruction queues, FP instruction execution can also be out of order relative to integer instructions. Although Power2 guarantees precise exceptions on integer instructions, exceptions are not precise relative to FP instructions. The integer and FP instruction streams are synchronized only on loads and stores. If an FP math operation, for example, causes a trap, the FPU completes all in-process calculations before signalling the trap, preventing exact identification of the trapping instruction. For debugging purposes, the FPU can be set to execute instructions serially, but this greatly reduces throughput.

Like Power1, the new processor uses register renaming (see MPR 8/21/91, p. 10) to reduce register conflicts in the FPU. This requires the chip set to provide more physical registers than the 32 defined in the architecture; these extra registers are allocated when data is loaded. The number of physical registers is increased from 40 to 54 in Power2.

Floating-point latencies have not improved from Power1, although the higher frequency reduces the execution times slightly. The new square-root instruction, however, is significantly faster than software emulation, as shown in Table 1.

Operation	Power2 (71.5 MHz)		Power1 (62.5 MHz)	
	Cycles	Time	Cycles	Time
FP Add*	2	28 ns	2	32 ns
FP Multiply*	2	28 ns	2	32 ns
FP Multiply-Add*	2	28 ns	2	32 ns
FP Divide	17	238 ns	20	320 ns
FP Square Root	27	378 ns	**53	848 ns

Table 1. Floating-point latencies for Power2 are generally the same as Power1. *These instructions are pipelined and can be issued on each cycle. **Square root requires software emulation in Power1.

Dual-Ported Cache Built Virtually

To increase memory bandwidth to match the higher execution bandwidth, Power2 emulates a dual-ported data cache. The DCU chips are able to perform two independent load/stores (in any combination) per cycle. Unlike Pentium and TFP, which emulate dual-porting using interleaved banks, the DCUs have no alignment restrictions regarding the two memory operations.

Instead of using a dual-ported SRAM cell, which would have greatly increased the area of the array, Power2 uses a technique called virtual multiplexing (VMP). Because the 0.6-micron process allows an SRAM cell that is much faster than the 14-ns processor cycle time, the DCU can "double-pump" its memory to deliver two results per cycle. One downside of this method is that the second read finishes later than the first, since they are done in series; this allows less time for the data to propagate back to the FXU/FPU. With its fast CMOS process and slightly overlapped accesses, the DCU delivers the first result in 5.6 ns and both results in 9.2 ns.

Main memory data is transferred into the cache through a 288-bit bus (256 data bits plus ECC) with an astounding peak data bandwidth of 2288 Mbytes/s. The DCUs can correct single-bit errors using the ECC, but data is stored internally with parity protection only. A cache refill typically takes 20 cycles (13-1-1-1-1-1-1-1 access pattern), depending on the type of memory used. The memory bus allows requests to be overlapped, helping to hide the initial memory latency and providing a sustainable bandwidth near the peak value.

To support the dual-port cache, the DTLB is also dual-ported. In this case, a dual-ported SRAM cell is used to reduce the access time. Table 2 lists the details of the cache and TLB configurations, which are quite similar to those in Power1.

MCM Improves Performance

A critical issue in multichip designs is the time it takes for signals to cross from one chip to another. Although the Power2 design is honed so that nearly all critical paths have only one chip-to-chip crossing, this transit time is still a large portion of the cycle time.

To improve chip-to-chip communication, IBM chose to package Power2 in a ceramic multichip module (see

071304.PDF). The Power2 module contains the three processor chips, four DCU chips, and the SCU. This design is well-suited to an MCM because, despite the thousands of inter-chip connections, only 512 signals need to go off of the MCM. Most of these signals are used for the 288-bit memory bus, memory address and control, and a 64-bit bus that connects to the I/O system. The Power2 module uses a 736-pin PGA design and measures 64 × 64 mm.

IBM estimates that the MCM design improves the clock speed by about 20% but increases the processor cost somewhat due to the complex ceramic substrate and increased testing and rework costs. For lower-speed systems, the company will package the chip set in ball-grid array (BGA) packages similar to those currently used for Power1 (see **071203.PDF**). The BGA packaging is required for a hypothetical two-DCU configuration.

The dice are fabricated in the same 0.6-micron, four-metal-layer CMOS process used for the PowerPC 601. As shown in Table 3, the ICU uses 2.8 million transistors (most for the instruction cache). The three processor chips, plus the control logic on the DCU chips, add up to 6.7 million transistors. Without even including the memory on the DCUs, which most processors would implement using discrete SRAMs, Power2 requires more than twice as many transistors as any single-chip processor currently available.

Brainiacs Lead—But at What Cost?

Power2 is a prime example of a “Brainiac” processor design favoring instruction parallelism over clock speed (see **0703ED.PDF**). With its six-way dispatch and multitude of functional units, it achieves over 1.7 SPECint92 per MHz; SuperSPARC, the previous leader, delivers about 1.3. This complex design outperforms all of the Speed Demons, including Digital’s 21064, which is rated at just 0.6 SPECint92 per MHz but can reach 200 MHz.

The IBM design indicates that Brainiacs can beat the Speed Demons, but they still have to pay the price of complexity. The MPR Cost Model (see **071004.PDF**) estimates that the Power2 manufacturing cost, in BGA packages, is \$540 for just the ICU, FXU, and FPU. The cost of the Power2 module is unknown due to the MCM itself, but in BGA packages the eight chips total to \$1050; IBM admits that the MCM version costs even more than discrete packaging but would not quantify the difference. Even at \$540 for the core processor, Power2 is still much more expensive than the 21064 or the R4400, which are each estimated to cost less than \$220 to manufacture.

IBM will have to move aggressively to stay ahead of the Speed Demons. Digital plans a 250-MHz version of the 21064 for 1H94 using a 0.5-micron process. MIPS Technologies (MTI) expects the R4400 to hit 200 MHz using similar IC technology. As it has done with Power1, IBM may eventually be able to deliver faster versions of

		Power2	Power1 (original)	
		4 DCUs	4 DCUs	2 DCUs
Instruction	Cache Size	32K	8K	8K
	Number of Sets	2 sets	2 sets	2 sets
	Line Size	128 bytes	64 bytes	64 bytes
	Data Protection	parity	parity	parity
	TLB Entries	128	32	32
	TLB Sets	2 sets	2 sets	2 sets
Data	Cache Size	256K	64K	32K
	Number of Sets	4 sets	4 sets	4 sets
	Line Size	256 bytes	128 bytes	64 bytes
	Refill Rate / cycle	32 bytes	16 bytes	8 bytes
	Write Protocol	write-back	write-back	write-back
	Data Protection	ECC	ECC	ECC
	Read Ports	2 ports	1 port	1 port
	TLB Entries	512	128	128
	TLB Sets	2 sets	2 sets	2 sets
	TLB Read Ports	2 ports	1 port	1 port

Table 2. Power2 has quadrupled most of the cache sizes from Power1 but otherwise retains a similar memory structure.

Power2 as it, too, moves to smaller transistors.

An important competitor for Power2 is MTI’s TFP processor (see **071102.PDF**), which also uses a multichip design to achieve high floating-point performance. Both processors can execute two FP multiply-adds and two FP load/stores per cycle, although Power2 can simultaneously execute a conditional branch. The IBM processor, with out-of-order execution, should also sustain a higher instructions-per-cycle rate through its FPU. TFP, however, is slightly faster at 75 MHz, even without an expensive MCM package.

TFP also offers an interesting contrast to Power2’s memory design. Like most current processors, TFP uses a large off-chip cache, typically 4M. TFP can load two words per cycle from this large cache, yielding a peak bandwidth of 1.2 Gbytes/s. Power2 uses a relatively small 256K data cache but can access main memory at a 2.3 Gbytes/s, although the memory latency is longer than the five-cycle latency of TFP’s cache. MTI designed TFP to handle large scientific applications that overflow the small, single-cycle caches on most microprocessors, but Power2 should do equally well on these large applications—and could do even better on programs that over-

	Transistor Count			Die Area	Signal I/O
	Logic	Memory	Total		
ICU	547K	2,277K	2,824K	161 mm ²	473
FXU	583K	848K	1,431K	161 mm ²	504
FPU	1,001K	315K	1,316K	161 mm ²	464
DCU	280K	4,000K	4,280K	161 mm ²	366
SCU	349K	—	349K	88 mm ²	276
Total (4 DCUs)	3,597K	19,440K	23,037K	1,215 mm ²	512 (MCM)

Table 3. The Power2 module contains eight chips totalling over 23 million transistors and 1215 mm² in die area.

System	RS/6000 Model 990	RS/6000 Model 580	RS/6000 Model 250	DEC 7000 Model 610	SGI prototype	HP 9000 Model 735	Sun prototype	Intel prototype
Processor	IBM Power2	IBM Power1	PowerPC 601	DECchip 21064	MIPS R4400	HP PA7100	TI SuperSPARC	Intel Pentium
Clock Rate	71.5 MHz	62.5 MHz	66.7 MHz	200 MHz	75/150 MHz	99 MHz	60 MHz	66 MHz
Cache (on/off-chip)	32K/256K	32K/64K	32K/none	16K/4M	32K/4M	none/512K	36K/1M	16K/256K
espresso	93.8	61.0	58.4	114.1	82.5	92.3	69.5	60.4
li	130.7	74.5	74.2	109.1	105.1	86.4	77.3	88.0
equott	164.2	90.2	76.9	164.2	134.1	90.9	127.0	54.5
compress	112.6	57.0	41.2	64.7	69.6	66.0	42.1	41.4
sc	172.2	103.4	85.2	226.5	101.6	71.7	115.9	96.0
gcc	102.4	64.4	51.6	83.4	84.9	76.7	62.2	62.8
SPECint92	126.0	73.3	62.6	116.5	94.2	80.0	76.9	64.5
spice	138.4	73.7	43.4	99.5	80.0	91.9	66.4	49.0
doduc	148.8	88.6	56.4	136.8	83.8	142.0	96.8	49.2
mdljdp2	195.3	124.2	85.7	153.8	133.8	192.1	103.1	62.5
wave5	159.5	69.2	48.6	112.5	81.1	112.1	68.4	38.7
tomcatv	473.2	210.3	94.3	301.1	160.6	138.0	86.9	68.3
ora	195.3	103.1	61.1	157.5	111.2	276.9	191.1	64.6
alvinn	801.0	206.2	160.9	367.9	116.0	176.8	209.3	111.9
ear	516.2	174.2	143.8	588.9	210.9	258.4	113.3	129.7
mdljsp2	88.6	57.3	47.6	74.8	66.5	92.3	50.3	30.1
swm256	242.4	95.8	63.9	201.3	68.5	79.3	49.5	40.9
su2cor	481.3	208.1	72.6	278.6	115.7	177.2	136.0	48.2
hydro2d	235.8	126.7	53.6	204.2	117.3	166.1	93.7	55.6
nasa7	370.9	203.9	72.1	265.0	124.9	123.3	110.6	48.2
fpppp	297.7	172.6	89.7	189.3	82.4	237.1	121.3	62.9
SPECfp92	260.4	124.8	72.2	193.6	105.2	150.6	98.1	56.9

Table 4. Power2 delivers a 50% increase in performance over the fastest Power1 system or PowerPC 601 system and maintains a lead over the fastest microprocessors from other vendors as well.

flow TFP's 4M cache.

Power2 will beat TFP to market and it has a lower estimated manufacturing cost. Without TFP's performance figures, it is hard to accurately compare the two processors, but Power2 appears to have a significant edge in performance per estimated cost. Of the two, only TFP will be available as a stand-alone processor chip set.

Powering Up the High End

IBM plans to continue the POWER line into the foreseeable future as a high-end alternative to PowerPC, but has no plans to sell Power2 on the merchant market. The company keeps its POWER processors for in-house use while selling the lower-performance PowerPC line externally. The PowerPC chips will also be used to bolster the low end of IBM's RS/6000 line.

Power2 demonstrates a number of advanced processor design techniques. Features such as register renaming and decoupled execution units are retained from Power1. The new design includes out-of-order execution and a dual-ported cache design. Virtual multiporting allows the dual-ported cache to be implemented with no die area penalty. None of these techniques has been implemented on non-IBM microprocessors but probably will be used in the future.

IBM is in the vanguard of processor design because it is willing to accept a multichip design with a larger transistor budget than most other programs. Even the

complex TFP requires fewer than 3.5 million transistors. The large number of logic transistors shown in Table 3 indicates the cost of such a complex design.

Power2 should do well for applications that take advantage of its astounding memory bandwidth and powerful floating-point units. While the processor is also a leader in integer performance, it does not appear to be cost-competitive with other current microprocessors. Since IBM does not sell Power2 except in systems, it can hide the chip-set cost in the cost of the system; profit margins on high-end systems can be quite large.

Power2 will initially be deployed at the high end of IBM's workstation line. As with Power1, future advances in IC manufacturing will both increase the performance and decrease the manufacturing cost of Power2, allowing it to move down toward the low end. Other microprocessors will ride the same trends, however, and single-chip designs will probably retain their current edge in price/performance and may possibly gain an absolute performance edge.

The ultimate match-up in the battle between the Brainiacs and Speed Demons may occur when the PowerPC 620 is unveiled around the end of next year. The 620 uses a similar instruction set as Power2 and will presumably use the same IC process. IBM has indicated that the 620 will exceed 200 SPECint92; Power2 will have to increase its performance by more than 60% in the intervening time to match that figure. As a single-

chip processor, the PowerPC chip will undoubtedly be much less expensive to manufacture.

Power2 demonstrates that the complex route is a good way to achieve high performance—if cost is not an issue and your transistor budget is large. At this point,

however, it appears that the Speed Demon approach still delivers better price/performance. Power2 will do well for scientific and other high-end applications but may ultimately be replaced by the PowerPC 620 for mainstream applications. ♦