

The IBM RISC System/6000 processor: Hardware overview

by H. B. Bakoglu
G. F. Grohoski
R. K. Montoye

A highly concurrent superscalar second-generation family of RISC workstations and servers is described. The RISC System/6000* family is based on the new IBM POWER (Performance Optimization With Enhanced RISC) architecture; the hardware implementation takes advantage of this powerful RISC architecture and employs sophisticated design techniques to achieve a short cycle time and a low cycles-per-instruction (CPI) ratio. The RS/6000 CPU features multiple-instruction dispatch, multiple functional units that operate concurrently, separate instruction and data caches, and zero-cycle branches. In this superscalar implementation, at a given cycle the equivalent of five operations can be executed simultaneously (a branch, a condition-register operation, and a floating-point multiply-add). The RS/6000 family supports the IBM Micro Channel architecture as well as high-speed serial optical links to provide a high-bandwidth I/O subsystem.

©Copyright 1990 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to *republish* any other portion of this paper must be obtained from the Editor.

Introduction

The IBM RISC System/6000* family consists of POWERstations and POWERservers, including desktop, deskside, and rack-mounted models. The operating system is IBM AIX Version 3, which is derived from UNIX System V and is POSIX IEEE Standard 1003.1-1988 conformant and Berkeley Software Distribution 4.3 (4.3 BSD) compatible. The RISC System/6000 (RS/6000) CPU is a highly concurrent superscalar second-generation RISC engine based on the new IBM POWER architecture. It combines this powerful RISC architecture with sophisticated hardware-design techniques to achieve a short cycle time and a low cycles-per-instruction (CPI) ratio. Like earlier RISC processors, the RS/6000 system employs a simple register-oriented instruction set, the CPU is hard-wired rather than microcoded, and it features a pipelined implementation. Unlike earlier RISC processors, however, the RS/6000 CPU employs several advanced architectural and design features, including a superscalar implementation (which involves multiple-instruction dispatch and simultaneous execution of fixed- and floating-point instructions), separate instruction and data caches, and a unique branch architecture that facilitates zero-cycle branches. The RS/6000 system also has a floating-point unit that is not a coprocessor but is highly integrated with the CPU. This significantly enhances floating-point performance.

A superscalar CPU that can execute multiple instructions simultaneously requires high instruction and data bandwidths to keep it busy. The RS/6000 has a

* RISC System/6000 is a trademark of International Business Machines Corporation.

four-word[†] memory bus, a four-word instruction-fetch bus from the I-cache arrays, a one-word data bus between the fixed-point unit and D-cache, and a two-word data bus between the floating-point unit and D-cache (Figure 1). These wide buses provide the high instruction and data bandwidths required for a high-performance superscalar implementation. In a single cycle, *four* instructions can be executed simultaneously (a branch, a condition-register instruction, a fixed-point instruction, and a floating-point instruction). In addition, the floating-point unit has a multiply-add ($A \times B + C$) instruction that executes with the same delay as a multiply or add. Counted as two operations, this increases the peak execution rate to *five* operations per cycle. The result is very high fixed- and floating-point performance.

Two popular performance tests are the Dhrystone and LINPACK benchmarks. Dhrystone is a benchmark written in C which measures fixed-point performance. It simulates the instruction mix of typical systems programming applications, and is commonly used to quote the fixed-point performance of RISC workstations and servers in units of VAX 11/780 MIPS (million instructions per second). (The VAX 11/780 is a member of a well-known family of computing systems manufactured by Digital Equipment Corporation.) LINPACK is a FORTRAN benchmark with a high percentage of floating-point operations that solves dense systems of linear equations. The result of LINPACK is reported in MFLOPS (millions of floating-point operations). At a 30-MHz CPU clock frequency, the RS/6000 CPU can execute 72 200 Dhrystones per second, which translates to 41.1 MIPS, and 13.0 double-precision LINPACK MFLOPS (for a 100×100 system of linear equations and all FORTRAN code—no coded BLAS).

Key design decisions

The RISC System/6000 processor is optimized to perform well in numerically intensive engineering and scientific applications as well as in multi-user commercial environments. A number of design choices were made specifically to ensure a high level of sustained performance in various environments.

To extend the performance beyond the capabilities of first-generation RISCs, the RS/6000 processor employs a superscalar implementation, which means that multiple instructions are issued and executed simultaneously. This requires independent functional units that can execute concurrently and a high instruction bandwidth to feed them. The RS/6000 CPU achieves this by implementing separate branch, fixed-point, and floating-point units, and by establishing a four-word interface to the I-cache arrays in order to be able to dispatch a maximum of four

[†] One word = 32 bits of data.

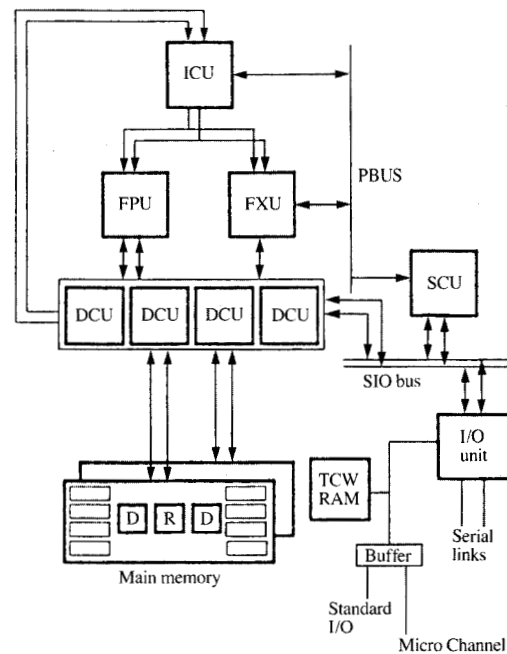


Figure 1

RISC System/6000 central electronics complex. Each line represents one word of data. One word contains 32 data bits, plus parity or ECC as appropriate.

instructions per cycle. This satisfies the peak instruction demand: a branch, a condition-register instruction, a fixed-point instruction, and a floating-point instruction. It is important to note that the POWER architecture was defined to enable such superscalar implementations with high parallelism at the instruction level.

In a RISC processor, the pipeline delay penalty caused by branches can be a significant portion of the overall CPI. This problem is addressed by performing a sufficiently far lookahead to eliminate the branches from the instruction stream. The branch unit achieves this by taking advantage of the four-word interface to the I-cache arrays and by implementing a large number of instruction buffers.

Floating-point performance is advanced beyond the limits of floating-point coprocessors by designing a floating-point unit that can execute concurrently with the fixed-point unit and by dispatching two instructions per cycle to the fixed- and floating-point units. Floating-point performance is further enhanced by implementing an accumulate (multiply-add) instruction that can be executed with the same delay as a multiply or add, and by establishing a two-word interface between the floating-point unit and the D-cache. The RS/6000 CPU has a full

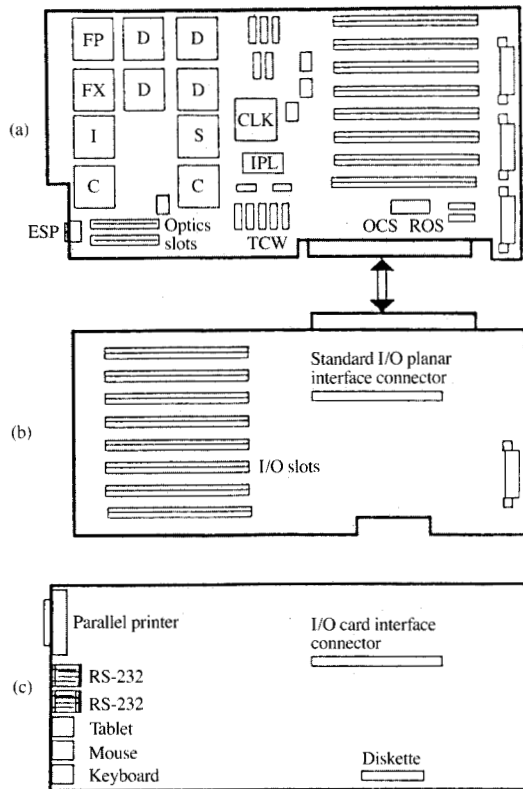


Figure 2

Planar boards: (a) CPU, (b) I/O, (c) standard I/O.

64-bit floating-point engine, and thirty-two 64-bit floating-point registers in addition to thirty-two 32-bit fixed-point registers.

In RISC computers the performance is extremely sensitive to the structure of the cache and memory subsystem [1]. Accordingly, the RS/6000 system has a highly optimized cache structure. Separate I- and D-caches provide conflict-free access to instructions and data, and allow independent optimization of both caches. For example, a four-word interface to I-cache arrays integrated with the branch processor is essential for the implementation of zero-cycle branches and multiple-instruction dispatch. In addition, both caches are structured as set-associative to minimize the possibility of cache thrashing. Typically, a set-associative cache achieves hit rates comparable to the hit rates of a direct-mapped cache that is twice the size of the set-associative cache.

In a system with fine-tuned caches, translation delay also becomes critical. The RS/6000 processor has separate instruction and data translation lookaside buffers (TLBs). The TLBs are large and set-associative. The fixed-point unit supports hardware TLB-reload and page-table-update functions. This provides a significant performance advantage over managing the TLB reloads and page tables in software, which is common practice in many RISC processors.

The memory interface is also carefully optimized. The memory bus is four words wide to ensure minimal interference between I-cache, D-cache, and direct-memory accesses (DMA) to memory. A two-word system I/O (SIO) bus provides an interface for the I/O unit. The separate SIO bus allows independent optimization of memory and I/O buses and provides electrical isolation between the two.

Another feature of the RS/6000 processor is that it supports *precise interrupts*; when an instruction causes an interrupt, the pipeline is stopped before the subsequent instructions can affect the machine state. Consequently, return from an interrupt can be resumed at the interrupting instruction. To implement a superscalar machine and still support precise interrupts was a challenge. It required very careful synchronization of the execution in branch, fixed-, and floating-point units. For example, as the branch unit looks ahead to fetch and decode the upcoming branch instructions, it keeps track of the interruptable dispatched instructions and keeps back copies of its state, so that it can back up to the state corresponding to the instruction that caused the interrupt.

The POWER architecture instruction set contains several powerful instructions, including string operations and an update form of load/store operations (similar to autoincrement). Typically these are not available in other RISC architectures. These instructions yield shorter path lengths in operating system and application code and improve performance. For ease of programmability and extendibility, the RS/6000 system provides four petabytes (2^{52}) of virtual address space and four gigabytes (2^{32}) of real address space.

The RS/6000 system features Micro Channel Architecture and high-speed optical serial links that provide a high I/O bandwidth.

Because of these careful design choices, the RS/6000 system not only attains excellent performance in industry standard benchmarks that usually deal with limited data volumes, but it also has plenty of reserve power for complex workloads with large applications or many users.

In the design of the RS/6000 processor, reliability, availability, and serviceability (RAS) were major considerations. All of the chip-to-chip data buses have

parity, and the memory bus has error detection and correction (ECC) and bit steering. Most of the chip data paths, including registers, register files, cache, TLB, and directory arrays, also have parity. For main memory, double-bit-detect, single-bit-correct ECC, bit steering, and memory scrubbing are implemented. The RS/6000 ECC word is 40 bits (32 bits of data, 7 bits of ECC, and 1 spare bit). ECC detects and corrects bit errors, bit steering substitutes a good spare bit for a failing bit, and memory scrubbing periodically passes the memory contents through ECC and bit-steering logic to correct the single-bit soft errors before they accumulate into double-bit errors. In addition, bit scattering is implemented to prevent more than one bit in an ECC word coming from the same by-four DRAM chip. This makes it possible to correct the errors due to a single bad DRAM chip on a memory card with bit steering and still have additional error protection due to ECC. When power is turned on, all of the CPU chips go through a built-in self-test (BIST) sequence where all the logic and memory are extensively tested, and any errors are recorded [2].

System hardware

The RS/6000 electronics are distributed among the CPU, I/O, and standard I/O planars as shown in Figure 2. The CPU planar contains the CPU chip set and the memory-card slots. RS/6000 memory cards are 8, 16, or 32 Mbytes, and a system can have 2 to 8 memory cards providing 16 Mbytes to 256 Mbytes of total memory. With the 4-Mbit DRAM technology, the memory card capability can be increased to 64 Mbytes, and the total memory can be increased to 0.5 Gbytes. The Micro Channel interface from the CPU is attached to the I/O planar, where it is buffered and feeds eight I/O slots. These I/O slots can be occupied by Micro Channel cards such as file adapters, tape-drive adapters, LAN adapters (Ethernet or token ring), display and graphics adapters, coprocessors, terminal emulators, and printer adapters. The I/O planar also contains the on-card sequencer (OCS), which initializes the CPU during initial program load (IPL) and controls the BIST sequence [2]. Other functions that are on the I/O planar are the nonvolatile RAM for configuration and error logging, operator-panel interface for error display, time-of-day clock, computer-reset register, and system status and configuration registers. The standard I/O planar contains the interfaces and connectors to keyboard, mouse, tablet, parallel-printer port, diskette, and two RS-232 serial ports. The I/O planar board is placed next to the CPU planar board and is attached to it by a connector. The standard I/O planar board fits underneath the I/O planar board and is attached to it through a connector (Figure 2 and Figure 3).

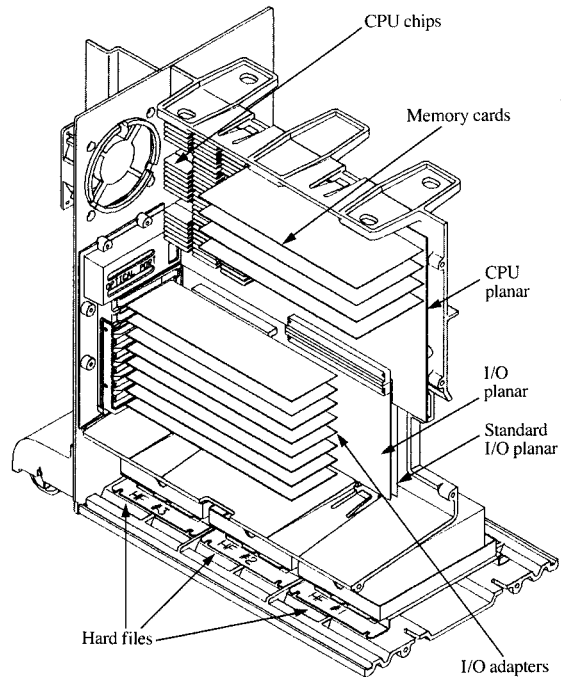


Figure 3

Floor-standing workstation.

Central electronics complex

The RS/6000 central electronics complex (Figure 1) contains several semicustom chips: an instruction-cache unit (ICU), a fixed-point unit (FXU), a floating-point unit (FPU), four data-cache units (DCU), a storage-control unit (SCU), an input/output interface unit (C), and a clock chip (CLK). Every memory card contains two data-multiplexing chips (D) and one control chip (R) for interleaving.

• I-cache and branch-processing unit

The ICU contains a two-way set-associative 8-Kbyte I-cache with a line size of 64 bytes [3]. It also has the I-cache directories and a 32-entry two-way set-associative I-TLB. The ICU processes branches, condition-register instructions, and supervisor calls, and dispatches the rest of the instruction to the fixed- and floating-point units. The ICU contains the machine-state registers and has the central control for interrupts. The instruction bandwidth requirements are satisfied by integrating the cache arrays

on the same chip with the branch processing and condition-register logic units. Four instructions per cycle can be fetched from the I-cache arrays to the instruction buffers and dispatch unit, which can dispatch up to four instructions per cycle. Two of these are internal dispatches to the ICU (branches and condition-register instructions) and two are external dispatches to the FXU and FPU. There is no restriction on the combination of instructions that are dispatched to the FXU and FPU. They can be a fixed- and a floating-point instruction, two fixed-point instructions, or two floating-point instructions. Because the fixed- and floating-point instructions are not mated together, instruction dispatch bandwidth or code space is not wasted. The instruction buffers have 12 entries. Fixed- and floating-point units never see any branches, and in most cases they receive an uninterrupted instruction stream and do not see the effect of the branches. This is referred to as *zero-cycle branches*.

Usually, unconditional branches cause no delay in the pipeline. Conditional branches that are not taken (fall-through) also have no penalty because the ICU dispatches the branch-not-taken path to the FXU and FPU before the outcome of the branch is determined. The branch-not-taken path instructions are canceled if the conditional branch is taken. The branch-taken path is fetched from the I-cache arrays and placed in the I-buffers, but dispatch is held off until the outcome of the branch is known. Conditional branches that are taken may delay the pipeline by zero to three cycles, depending on how much earlier the condition register was set. The compiler tries to move condition-code-setting instructions far ahead of the conditional branches to minimize the conditional-branch penalty. The condition register (CR) of the POWER architecture is unique; it has multiple independent condition-code fields [3]. This allows several CR-altering instructions to be outstanding and increases the parallelism between the functional units, because each functional unit can send its condition code to different CR fields. The branch processor knows which outstanding instructions can affect the dependent portion of the CR, and resolves the branch as soon as these instructions are completed. With multiple condition codes, multiple compare instructions can be dispatched, followed by the branch instructions that depend on those compares. In this way the distance between compares and the related branch instructions can be increased, which minimizes the compare-and-branch penalty. The ICU also implements a branch-and-count instruction which is useful in loops with a known count (e.g., $DO\ i = 1, 100$). This instruction is fully implemented in the ICU and does not use the register file or ALU of the FXU. The count register is in the ICU and decrementing is done there. The branch-and-count also yields zero-cycle branches.

● *Fixed-point unit*

The FXU decodes and executes all fixed-point instructions and floating-point load and store instructions [3]. Both fixed- and floating-point instructions go to the I-buffers of the FXU and FPU and are executed concurrently in the FXU and FPU. The FXU contains the general-purpose registers and the arithmetic logic unit. Register tagging is implemented to allow data-cache accesses (loads and stores) to overlap with the execution of subsequent independent register-register instructions. The FXU features a fixed-point multiply/divide unit; the multiply instruction takes 3 to 5 cycles, and the divide takes 19 to 20 cycles.

The FXU contains the segment registers and a 128-entry two-way set-associative D-TLB for address translation, page protection, and data locking. Page-table lookups for I-TLB and D-TLB reloads and page-table updates are performed by the FXU hardware. In addition, hardware data locking and hardware lock granting are supported, for improved performance in database and transaction-processing applications [4, 5].

D-cache directories and controls are also located in the FXU. Consequently, address generation and D-cache controls for both fixed- and floating-point load/store instructions, as well as cache operations, are performed by the FXU. Data and address of one fixed-point store instruction can be held in the *store buffers* in the FXU waiting for a convenient time to be written into the D-cache. Consequently, fixed- and floating-point loads can get ahead of the fixed-point stores, and the fixed- and floating-point execution units obtain the data they need sooner. It should be noted that the instructions are not executed out of order from the perspective of the execution unit and checking for data storage interrupts; only the access to the D-cache is out of order.

● *Floating-point unit*

Unlike typical floating-point coprocessor chips, the RS/6000 FPU is tightly coupled to the rest of the CPU [6]. The FPU and FXU are independent functional units of equal priority. They receive instructions from the ICU simultaneously and execute them concurrently. The RS/6000 FPU conforms to the IEEE 754 binary floating-point standard [7]; it has a double-precision-wide data path and executes floating-point arithmetic operations (multiply, add, divide, subtract). The floating-point unit generates one double-precision result per cycle, and the execution pipeline latency is only two cycles. The short execution pipeline latency is crucial in achieving high performance because in a longer pipeline the possibility of creating hold-off conditions due to dependent operations that stall the pipeline increases. For example, with a single-cycle execution pipeline and full bypassing, two dependent operations can follow one another

without stalling the pipeline. In a two-cycle execution pipeline, the compiler needs to insert only one independent operation between two dependent operations in order to avoid any bubbles in the pipeline.

A key feature of the POWER architecture floating-point instruction set and the RS/6000 FPU is the *multiply-add* instruction, $(A \times B) + C$. The multiply-add operation is executed with a single round and with the *same* delay as a multiply or add instruction. This reduces instruction-path lengths by combining two instructions into one, and enhances the floating-point performance significantly. This is true especially in numerically intensive scientific or graphics applications where matrix operations can take advantage of the multiply-add instruction. The RS/6000 multiply-add instruction also has additional accuracy compared to more traditional implementations, because the result of the multiply is not rounded before the add, and consequently no accuracy is lost between the multiply and add steps. Due to the full 64-bit data flow, the FPU can execute a double-precision multiply, add, or multiply-add every cycle in a pipelined fashion. A two-word interface to the DCU provides the required data bandwidth, and load and store operations are fully overlapped with the execution of arithmetic instructions for maximum performance.

The FPU uses *register renaming* to increase the overlap of the execution of floating- and fixed-point functional units. This allows floating-point loads to be executed independently of the floating-point arithmetic operations and makes it possible to carry on loads to a target register of a floating-point arithmetic instruction while the arithmetic instruction is still going on. This is done by remapping the target register to one of the remap registers. As a result, the FXU can perform floating-point loads without having to wait for previous floating-point arithmetic operations to be completed.

Similarly, a five-entry *pending-store queue* and a four-entry *store data queue* in the FPU enable the FXU to execute floating-point stores before the FPU produces the data. This allows the FXU to generate the address, initiate TLB-reload or cache-reload sequences, and check for data protection for a floating-point store instruction, and then continue executing the subsequent instructions without being held back by the FPU. It also allows the FXU and FPU to read data from the DCU before writing the data of a previous floating-point store instruction in the DCU. This allows fixed- and floating-point loads to get ahead of the floating-point stores, and reduces the possibility of the fixed- and floating-point execution units starving for data.

The RS/6000 FPU has thirty-two 64-bit floating-point registers, six rename registers, and two divide registers. It features a leading-zero anticipator [8] to avoid the full delay of a leading-zero detector and to ease the overlap of

multiplication and addition. This makes the two-cycle-latency floating-point execution pipeline possible.

• *Data-cache unit*

The RS/6000 processor has a four-way set-associative 64-Kbyte D-cache divided into four identical DCU chips of 16 Kbytes each [9]. The cache-line size is 128 bytes, and the cache is implemented as a store-back cache to minimize the memory bus traffic. (When data is stored in the D-cache, it is not sent to memory. The data is written into main memory only when a dirty line in the cache is replaced.) The four DCUs form a four-word interface to memory, a two-word interface to the FPU, a single-word interface to the FXU, a two-word interface to the ICU, and a two-word interface to the I/O unit. DCUs support bit steering and ECC for load/store, I-cache reload, DMA, and memory-scrub operations. DCUs also contain temporary storage buffers for I-cache reloads and DMA to account for the width difference between the memory bus and I-cache reload and SIO buses. D-cache directories, status arrays, and D-TLBs are in the FXU.

The performance advantages of a custom-designed D-cache come mainly from the *cache-reload buffers* (CRB) and *store-back buffers* (SBB) [10]. A 128-byte CRB divided across four DCUs receives data from memory, FXU, or FPU. Unlike simpler cache implementations that do not have a CRB, the RS/6000 CPU need not wait for the entire cache line to be brought from main memory before it can access the cache arrays. Instead, the line is brought from the memory and stored in the CRB such that the first packet contains the data that caused the cache miss. A fast load-through path that bypasses the cache arrays and CRB is provided from the memory bus to the FXU/FPU to minimize the cache-miss penalty for loads. This bypass is done in parallel with error detection, and any errors are signaled to the FXU/FPU at the beginning of the next cycle.

Consequently, ECC comes at minimal performance penalty. The cache line is loaded into the CRB in eight cycles, but the cache arrays are not tied up during the reload sequence. Any subsequent load/store can read/write data from the cache arrays or CRB before the data in the CRB is loaded into the arrays. If the data were being written into the arrays as it arrives from memory, subsequent loads and stores would not be able to access the cache arrays and would have to wait for the completion of the reload. In the RS/6000 cache, the entire line is loaded into the cache arrays simultaneously later when the cache is not being used.

Store-back buffers are also 128 bytes wide. They can accept data from D-cache arrays or directly from the CRB and pass it to main memory. The SBB improves performance, because with an SBB the dirty line need not be written back to the memory before the new line is

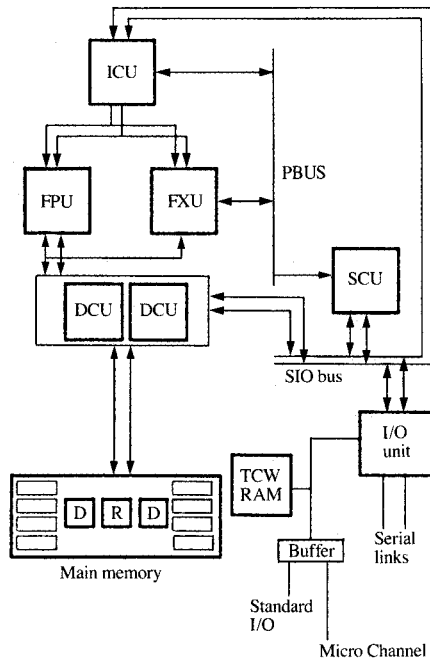


Figure 4

Cost-reduced CPU.

brought into the cache. In addition, the data-cache arrays are not kept busy during the whole store-back sequence. The entire line is parallel-loaded into the SBB in two cycles, and the store-back data is sent to memory after the new line is brought back from memory. The DCUs can service the CPU during cache-reload and store-back sequences because the cache arrays are freed up by the CRB and SBB. In addition, the store-back data can be left in the SBB and stored back later if a higher-priority memory access is pending. Without the CRB and SBB, D-cache line size would be limited to a shorter length and would require larger directories. Longer cache lines minimize the cache-directory area and make it possible to fit them in the FXU. Longer cache lines also provide a prefetch-like effect and improve the hit ratios. In addition, a longer cache line makes it possible to overlap the memory access latency (leading edge) with the previous data transfer (trailing edge) and enable a peak memory-bus utilization of 100%.

For increased reliability and availability, the DCU supports memory scrubbing, double-bit-detect, single-bit-correct ECC, and bit steering. Memory scrubbing is performed periodically by reading the memory locations, passing them through the ECC and bit-steering logic, and writing them back to memory when a correctable error is

detected. In this way soft errors can be corrected before too many of them can accumulate and become uncorrectable. When a correctable error is detected during scrubbing, the sequence is repeated to determine whether the error was soft or hard. Bit-steering logic in the DCU can be programmed to bypass a hard failure in memory by substituting a good spare bit for a failing bit. Bit steering is reprogrammed during every system power-up. The combination of memory scrubbing, ECC, bit steering, and bit scattering minimizes the possibility of encountering an uncorrectable memory error.

• Storage control unit

The SCU is the central system controller [9]. All of the communication between CPU (ICU, FXU, DCU), main memory, and I/O is arbitrated by the SCU. The CPU sends I-cache reload, D-cache reload, and D-cache store-back requests to the SCU over the PBUS, and the SCU generates the appropriate memory-control signals. The SCU is the bus master for the memory and SIO buses. It controls the interface between D-cache and system memory, and oversees DMA operations between main memory and the I/O unit. The SCU provides a data path for I/O loads and stores between the CPU and I/O unit via the PBUS and SIO bus. The SCU also forms an interface to the IPL ROS. Memory scrubbing is controlled by the SCU, and memory errors detected by the DCU are recorded by the SCU. The SCU contains the bank configuration registers, which indicate the size and starting point of each bank of storage in system memory.

• Memory cards

RISC System/6000 memory cards implement a four-way-interleaved design in order to provide two words of data every machine cycle from each memory card. A minimum of two memory cards is required to support the four-word memory bus to the CPU (Figure 1). The interleaving is performed by two data-multiplexing chips (D) and one control chip (R) on the memory cards. The memory cards accept generic read/write instructions and generate the required read/write, refresh, and page-mode read/write signals for a variety of DRAMs. The memory cards can buffer up to four instructions and 16 words of write data. Using standard modular SIMM packages for the DRAMs provides flexibility in memory capacity and cost. RS/6000 memory cards can support both 1-megabit and 4-megabit DRAMs. The SIMMs are 1, 2, or 4 Mbytes. There are eight SIMM sockets per memory card, yielding 8-, 16-, or 32-Mbyte memory cards. A system can have 2 to 8 memory cards providing 16 Mbytes to 256 Mbytes of total memory. With the 4-Mbit DRAM technology, the memory card capability can be increased to 64 Mbytes, and the total memory capability can be increased to 0.5 Gbytes. Bit scattering is implemented to

guarantee that no more than one bit of a 40-bit ECC word is stored in an individual DRAM. Accordingly, the bit-steering circuitry in the DCU can detect and correct errors caused by a single bad by-four DRAM chip and still have additional error protection due to ECC.

• *I/O unit*

The I/O unit contains an I/O channel controller (IOCC) and two serial link adapters (SLAs). The IOCC generates a Micro Channel interface, and the SLAs provide an interface to optics cards that drive fiber-optic links. Support for this high-speed optical link is planned for a future release. The data interface between CPU/system memory and the I/O unit is via the two-word SIO bus. Micro Channel has a one-word address bus and a one-word data bus. Data transmission between an SLA and the optics cards are via one-byte interfaces.

The IOCC supports an I/O architecture geared for performance, robustness, and error recoverability [11]. The main function of the IOCC is to transfer data between system memory and adapters on the Micro Channel. The CPU can transfer data to/from adapters using I/O load and store operations, and adapters can transfer data to/from system memory using DMA. The IOCC supports both DMA bus masters and DMA slaves. All data transfers support address-protection mechanisms to provide data security. Up to 15 DMA channels and 16 levels of interrupts are supported by the IOCC. The Micro Channel Architecture features include *streaming data*, address and data parity, and synchronous exception reporting (I/O loads/stores cause precise interrupts like regular loads/stores) [12]. With the new streaming-data mode, multiple data packets can be transferred within one bus envelope. This is accomplished by sending a starting address and then transferring a block of data consisting of multiple consecutive data packets. This is in contrast to the regular mode, where an address is required for every data transfer. This amortizes device-selection overhead across the entire packet and nearly doubles the performance for large data bursts [12]. Precise I/O load/store interrupts improve error recoverability.

The SLA implements the serial I/O architecture, which supports point-to-point connections where two RS/6000 machines communicate with each other via optical fibers [13]. The SLA receives the data from the CPU, puts it into 10-bit packets, and transmits the packets to the optics card. The optics card serializes the data and feeds it to a fiber-optic link. The SLA also supports I/O loads/stores to/from the CPU. Each SLA contains two 256-byte data buffers to transfer data between the SIO bus and the optics card, and one 16-word tag table to provide pointers to a system-memory buffer for DMA operations.

Table 1 Physical characteristics of the IBM RISC System/6000 chip set.

Chip	Transistor count		Die size (mm × mm)	Signal pin count
	Logic	Memory		
ICU	200,000	550,000	12.7 × 12.7	252
FXU	250,000	250,000	12.7 × 12.7	256
FPU	360,000	60,000	12.7 × 12.7	224
DCU × 4	700,000	3,800,000	11.3 × 11.3	184
SCU	230,000	—	11.3 × 11.3	255
I/O unit	300,000	200,000	12.7 × 12.7	293
Total	2,040,000	4,860,000	1284 mm ²	

• *Cost-reduced CPU*

One of the goals of this design was to use a common chip set to produce a family of processors with varying cost and performance. This was accomplished by designing the FXU, DCU, and SCU in such a way that they can operate with two DCUs as well as with four. This system configuration is illustrated in **Figure 4**. Because the chips are common to the two versions, they can be stored so that faster chips are used in the high end and slower ones in the entry-level configurations. In this way, sorting for high speed is achieved without sacrificing the overall yield.

This configuration has a lower cost for two reasons. First, it has only two data-cache chips rather than four. Second, it requires a minimum of one memory card rather than two. (This is because two DCUs in the cost-reduced CPU have a two-word memory interface compared to four DCUs in the full CPU, which have a four-word interface. Consequently, some of the bit-scattering features described for the full-size CPU do not apply to the cost-reduced CPU.) To accommodate the smaller cache size and narrower memory bus width, the D-cache line size is reduced to 64 bytes. In the cost-reduced CPU, fixed- and floating-point data buses are dotted together. In addition, the DCU sends the data to reload the I-cache over the SIO bus rather than having a dedicated I-cache reload bus to the ICU.

Chip and packaging technologies

Physical attributes of the RS/6000 chip set are summarized in **Table 1**. The chips are implemented in a CMOS technology, which has 1- μ m minimum feature sizes, 0.9- μ m effective channel lengths, one level of polysilicon, and three levels of metal wiring. Maximum die size is 12.7 mm. Chips are placed on individual ceramic pin-grid-array modules, which can support a maximum of 300 signal pins and 4 W of power dissipation. The pins on the regular grid are 100 mil spaced. Additional pins are placed at the interstitial points to support the required pin count. A large number of power/ground pins are provided to minimize the

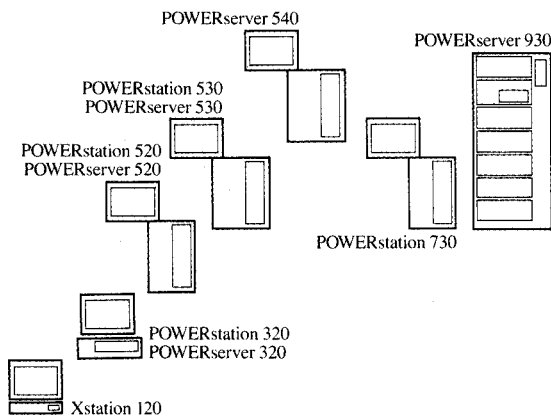


Figure 5

RISC System/6000 workstations and servers.

simultaneous switching noise. The chip carriers are mounted on an advanced mixed-grid card by through-hole technology. The card supports holes drilled on demand on a 50-mil-spaced grid. There are four signal and four power/ground planes (four power/ground planes in the middle, and two signal planes on both sides).

Product family

The RISC System/6000 family offers a broad range of products to satisfy the needs of a wide customer base (Figure 5 and Table 2).

The Xstation 120 is a high-function LAN-attached X-server station that is capable of performing X-windows functions. It processes the X-windows protocols but is dependent on a host machine to run the application. Xstations are attached to the RISC System/6000 hosts via a LAN (token-ring or Ethernet). The X-Windows System protocol allows the separation of the computing function at the POWERserver host from the user interaction at the Xstation, thus utilizing both the processing power of the network server and the graphics capabilities of the Xstation 120. It is targeted to be attached to one of the RS/6000 servers to provide a low-cost multi-user solution. In many applications, an Xstation does not load the LAN as much as a diskless workstation because the Xstation does not have to download the application code or the data from the server. Only the data that needs to be displayed is sent to the Xstation in an efficient format.

The POWERstation/POWERserver 320 is the entry-level desktop unit. It contains the cost-reduced CPU running at 20 MHz and has four Micro Channel slots and two memory card slots. It supports 8 Mbytes to 32 Mbytes of main memory. The desktop unit can

accommodate up to two internal 3½-in. SCSI disk files providing up to 640 Mbytes of storage.

There are multiple deskside models. A deskside model can have the cost-reduced or full CPU and comes in a standard enclosure or in a wide enclosure. The POWERstation/POWERserver 520 has the cost-reduced CPU running at 20 MHz. The POWERstation/POWERserver 530 has the full CPU running at 25 MHz. The POWERserver 540 has the full CPU running at 30 MHz. The POWERstation 730 has the full CPU running at 25 MHz; it is the high-end graphics workstation and uses the wider enclosure. The wider enclosure has a 950-W power supply in place of the standard 650-W power supply and includes a high-function graphics subassembly. All of the deskside models accommodate up to eight memory cards, up to eight Micro Channel cards, and up to two optics cards (providing up to four optical serial links). The maximum memory configuration is 128 Mbytes if 1-Mbit DRAM chips are used. With the 4-Mbit DRAM technology, the memory card capability can be increased to 64 Mbytes, and the total memory capability can be increased to 0.5 Gbytes. The deskside models support up to three internal 5¼-in. full-height SCSI disk files providing a maximum of 2.5 Gbytes of storage. As an alternative configuration, the system can be configured with one to six internal 3½-in. SCSI disk files. External disk files are available to expand the disk storage of the desktop and deskside models. A portable disk with removable disk files is provided for high-security environments.

The POWERserver 930 is the rack-mounted model and serves as a minicomputer. It has the same CPU, I/O, and standard I/O planars as the deskside models with the full CPU, and can have up to two optics cards. The rack-mounted model can include up to four disk file drawers in addition to the CPU drawer. Every drawer can have up to four 5¼-in. SCSI disk files totaling a maximum of 12 Gbytes of disk storage capacity. The rack-mounted model supports battery backup capability, so that system operation can continue in the event of power failure.

Removable storage media offerings for backup and other purposes include 3½-in. and 5¼-in. diskette, nine-track ½-in. tape, ¼-in. streaming tape, 8-mm tape, and CD ROM. A rich set of printers, plotters, tables, digitizers, keyboards, mice, dials, and lighted PF keys are offered as I/O devices. The display choices include various ASCII displays and 16-, 19-, and 23-in. monochrome and color graphics displays. The supported connectivity and communications hardware includes token-ring and Ethernet for LANs, multiport asynchronous adapters for asynchronous terminals, synchronous communication (SDLC, BSC, and X.25), and various host attachments for mainframes [3270 emulation, 5080 support, 5086 attachment, and DFT (TCA/DCA)].

Table 2 RISC System/6000 family.

Model	320	520	530	730	930	540
Configuration	Desktop	Desktop	Desktop	Desktop	Rack	Desktop
Memory						
Standard (Mbytes)	8	8	16	16	16	64
Maximum (Mbytes)	32	128	128	128	128	256
Internal hard disk						
Standard (Mbytes)	120	355	355	355	670	640
Maximum (Mbytes)	640	2,571	2,571	2,571	11,998	2,571
Micro Channel slots	4	8	8	8	8	8
Parallel port	1	1	1	1	0	1
Serial ports	2	2	2	2	2	2
Performance						
Clock (MHz)	20	20	25	25	25	30
MIPS	27.5	27.5	34.5	34.5	34.5	41.1
Dhrystones (KDhry/s)	48.3	48.3	60.7	60.7	60.7	72.2
LINPACK DP (MFLOPS)	7.4	7.4	10.9	10.9	10.9	13.0

There are three basic graphics adapters that address various needs. They all support 1280-by-1024 pixels of resolution and a 16-million-color color palette. The entry adapter is available in monochrome or color and is targeted for 2D graphics applications. It also performs 3D graphics with software assistance. The monochrome version can draw 76 000 2D lines per second, and the color version can draw 131 000 2D lines per second. The midrange graphics adapter is targeted for 3D applications. It can draw 90 000 3D vectors per second, and process 10 000 Gouraud-shaded triangles per second. It also offers additional options such as Z-buffers and double buffers. The high-end graphics display is used with the POWERstation 730 and is suitable for high-end 3D graphics applications and animation. It can draw 990 000 3D vectors per second and 120 000 Gouraud-shaded triangles per second.

Conclusion

One of the primary goals of this second-generation RISC project was to design a high-performance and truly balanced RISC computer that avoided bottlenecks in the CPU, caches, memory interface, and I/O subsystems. This is achieved by

- The POWER architecture, which makes possible a high degree of parallelism at the instruction level.
- A superscalar implementation with multiple functional units that can execute concurrently.
- Highly fine-tuned caches and memory subsystem.
- A high-bandwidth I/O subsystem with Micro Channel and high-speed optical serial links.

In addition, throughout the system, all of the components are designed to meet high reliability, availability, and serviceability criteria.

Acknowledgments

Most of the key processor concepts for the IBM RISC System/6000 computer were conceived at the IBM

Thomas J. Watson Research Center under the leadership of John Cocke; important early contributions were made by Sharon Chuang to the fixed-point pipeline design and by Richard Matick to the conception of the functional cache. The vision of Andrew Heller played a major role in turning those concepts into reality in a second-generation RISC workstation. Finally, the RISC System/6000 computer as a product would not have been possible without the dedication of the entire Austin team, including the authors of the cited references and companion papers and the logic design, physical design, design tools, simulation, verification, and test groups at Austin. The hardware and manufacturing technology have been provided by several IBM locations, including Burlington, Endicott, and Austin.

References

1. R. E. Matick and D. T. Ling, "Architecture Implications in the Design of Microprocessors," *IBM Syst. J.* **23**, 264-280 (1984).
2. I. M. Ratiu and H. B. Bakoglu, "Pseudorandom Built-In Self-Test Methodology and Implementation for the IBM RISC System/6000 Processor," *IBM J. Res. Develop.* **34**, 78-84 (1990, this issue).
3. G. Grohoski, J. Kahle, L. Thatcher, and C. Moore, "Branch and Fixed-Point Instruction Execution Units," *IBM RISC System/6000 Technology*, Order No. SA23-2619, pp. 24-33, 1990; available through IBM branch offices.
4. R. R. Oehler and R. D. Groves, "IBM RISC System/6000 Processor Architecture," *IBM J. Res. Develop.* **34**, 23-36 (1990, this issue).
5. A. Chang and M. F. Mergen, "801 Storage: Architecture and Programming," *ACM Trans. Computer Syst.* **6**, 28-50 (February 1988).
6. B. Olsson, R. Montoye, P. Markstein, and M. Nguyenphu, "RISC System/6000 Floating-Point Unit," *IBM RISC System/6000 Technology*, Order No. SA23-2619, pp. 34-43, 1990; available through IBM branch offices.
7. "IEEE Standard for Binary Floating-Point Arithmetic," *ANSI/IEEE Standard No. 754*, American National Standards Institute, Washington, DC, 1988.
8. E. Hokenek and R. K. Montoye, "Leading-Zero Anticipator (LZA) in the IBM RISC System/6000 Floating-Point Execution Unit," *IBM J. Res. Develop.* **34**, 71-77 (1990, this issue).
9. W. Hardell, D. Hicks, L. Howell, W. Maule, R. Montoye, and D. Tuttle, "Data Cache and Storage Control Units," *IBM RISC System/6000 Technology*, Order No. SA23-2619, pp. 44-51,

- 1990; available through IBM branch offices.
10. Richard E. Matick, "Functional Cache Chip for Improved System Performance," *IBM J. Res. Develop.* **33**, 15-32 (1989).
 11. J. Nicholson, D. Neal, S. Dhawan, R. Arimilli, and D. Siegel, "RISC System/6000 I/O Structure," *IBM RISC System/6000 Technology*, Order No. SA23-2619, pp. 56-61, 1990; available through IBM branch offices.
 12. J. O. Nicholson, "Micro Channel Features," *IBM RISC System/6000 Technology*, Order No. SA23-2619, pp. 52-55, 1990; available through IBM branch offices.
 13. J. W. Irwin and J. R. Mathis, "Serial I/O Architecture and Implementation," *IBM RISC System/6000 Technology*, Order No. SA23-2619, pp. 62-67, 1990; available through IBM branch offices.

Received June 29, 1989; accepted for publication January 18, 1990

H. B. Bakoglu *IBM Advanced Workstations Division, 11400 Burnet Road, Austin, Texas 78758 and IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598.* Dr. Bakoglu received the B.S.E. degree in electrical engineering and computer science in addition to engineering physics from Princeton University in 1982, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University in 1984 and 1986, respectively. He joined the IBM Thomas J. Watson Research Center in 1986, and is currently on assignment with the IBM Advanced Workstations Division in Austin, Texas, where he is a Systems Architect in the Systems Engineering and Architecture area. Dr. Bakoglu has received many awards, including the IBM graduate fellowship, the Jeffrey O. Kephard Prize in engineering physics, the Treen Scholarship, and the Princeton University Jadwin and Hall-Mercer scholarships. He has served on the program committee of the IEEE International Solid-State Circuits conference, has published numerous papers on VLSI design, and holds patents on semiconductor devices. Dr. Bakoglu is a member of Phi Beta Kappa, Sigma Xi, and Tau Beta Pi, and is the author of *Circuits, Interconnections, and Packaging for VLSI* (Addison-Wesley, 1989).

Gregory F. Grohoski *IBM Advanced Workstations Division, 11400 Burnet Road, Austin, Texas 78758.* Mr. Grohoski received a B.S. with distinction in electrical engineering from Cornell University in 1980 and an M.S. in electrical engineering from the University of Illinois at Urbana-Champaign in 1981. That same year he joined the IBM Research Division at the Thomas J. Watson Research Center in Yorktown Heights, New York, where he worked on high-performance RISC machine designs. In 1986 he transferred to IBM Austin to work on the RISC System/6000 project. Mr. Grohoski holds two IBM Invention Achievement Awards and an IBM Outstanding Technical Achievement Award; he has applied for five patents. He is currently an Advisory Engineer in the hardware architecture group.

Robert K. Montoye *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598.* Dr. Montoye received his B.S. in physics in 1977 and his M.S. in 1981 and Ph.D. in 1983 in computer science from the University of Illinois. He joined IBM in 1983 and began research into high-performance CMOS design, including the MAF floating-point unit. Dr. Montoye is the author of numerous articles and holds patents in parallel processing, VLSI architectures, and design automation.