

HP Reveals Superscalar PA-RISC Implementation

By Brian Case

At a time when many other vendors are using a "start from scratch" approach to create very highly integrated, extremely complex processor implementations, HP continues to use a straightforward, conservative approach for evolving its PA-RISC architecture. This latest PA-RISC implementation, the 7100, takes the "simple" steps of integrating the integer unit and floating-point unit together on a single chip, incorporating a modest superscalar capability, and improving the raw clock rate by 50%. Instead of integrating small first-level caches on chip, this design still relies on fast I/O drivers and fast SRAMs to allow the external caches to be cycled at the processor frequency. In this respect, HP's design style is at odds with the rest of the industry.

HP has not announced any systems using the 7100 processor, and there are no immediate plans to make it available on the merchant market. HP says that systems using the chips will be shipped late this year. No benchmark results have been released, but HP claims that it will perform at more than 120 SPECmarks. The chip's floating-point performance will be much higher than its integer performance, however, and the SPECint rating is likely to be closer to 75—still outstanding by today's standards, but probably not much better than a 75/150-MHz R4000 or a 50-MHz SuperSPARC, which should ship in the same timeframe.

Chip and System Overview

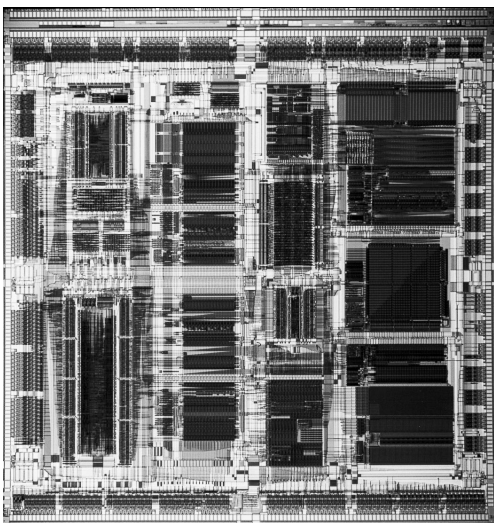
The CPU chip incorporates the basic integer unit and cache control circuitry from the previous "Snakes" processor design. To increase the clock frequency from

66 to 100 MHz, the design was shrunk from 1.0 micron to 0.8 micron and some slow timing paths were sped up. While the Snakes design used a separate floating-point chip designed jointly with Texas Instruments, the 7100's on-chip FPU was designed by HP. The unified TLB has 120 standard translation entries (compared with 96 in Snakes), which is large by the standards of other processors. It also has 16 large-page entries that can each map an area from 512 KB to 64 MB in size.

The chip has 850,000 transistors, 300,000 in the FP unit and 550,000 in the integer unit, instruction control, and bus interface logic. The chip size is 14×14 mm, which is large for a sub-one-million transistor chip. HP's fabrication technology is certainly fast, but it is not as dense as some other 0.8 micron processes. Metal pitches range from 2.0 to 5.0 microns, which is wide by today's standards, but such wide metal may contribute to the speed advantage of the process. The chip is also mostly logic, so it is not as dense as most other high-end processors that use half of their transistors (or more) for cache.

The chip has 480 pads (the package is a 504-pin PGA) to accommodate the large number of buses and power and ground pins needed to operate the buses at high speed. Figure 1 shows a block diagram of the chip and cache subsystem. The cache data buses are each 64 bits wide and can deliver eight bytes on each clock cycle. The caches are direct mapped to eliminate the performance impact of multiplexers that would be required with set-associativity. To reduce the cache miss rates, the addresses are hashed before they drive the cache SRAMs.

Continued on page 25



Die photo of HP's 7100 PA-RISC processor.

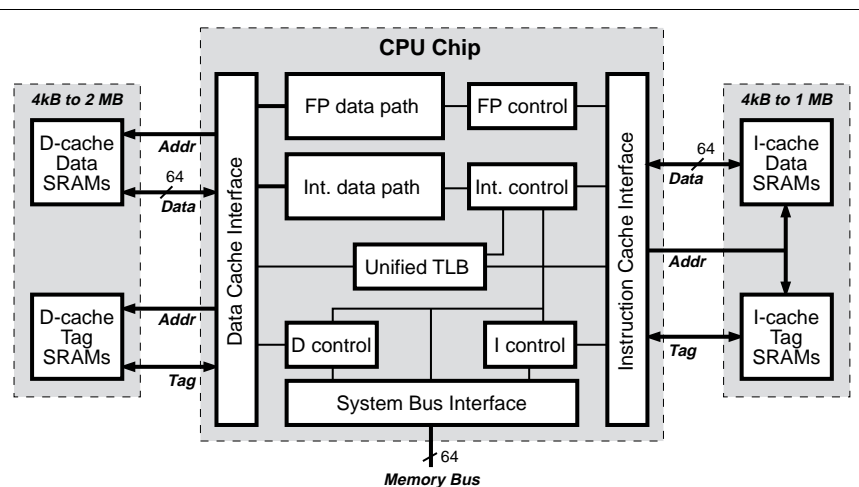


Figure 1. Block diagram of a PA-RISC 7100 CPU subsystem.

result, sometimes an architectural feature that was innocuous in a simple sequential implementation requires gates and gate delays out of all proportion when present in a compatible, but more aggressive, implementation. What seems to irritate some CISC designers the most is needing significant implementation complexity just to watch out for some truly rare event. (In bar conversations with such designers, this irritation often surfaces in comments of the form "You RISC wimps! You don't know how lucky you are not to have to worry about *this*, or *that*, or *those* nonobvious implications of architecture XYZ.")

Conclusions

Fast computers often use similar implementation techniques, and few of those techniques are truly new. However, architecture strongly influences the ability to use various implementation techniques, the cost of doing so, and the resulting performance gained.

Electrical engineering fundamentals say that complexity still costs. Transistors may get to be almost free, but wires and gate delays will not.

Contrary to the belief of some, there are some clear architectural distinctions between the CPUs commonly labeled RISC and CISC, and there is very little sign of architectural convergence.

Of course, none of this proves that RISC is automatically better than CISC, and in fact, a good CISC implementation should beat a poor RISC implementation. Perhaps we should seek more specific A-vs.-B comparisons, but it would certainly be better if people would stop trying to obfuscate well-known computer history. ♦

References

[BEL71] Bell and Newell, *Computer Structures: Readings and Examples*, McGraw-Hill, 1971. Many currently-popular aggressive implementation methods date from the late 1960s. For example, study CDC 6600 (1964) and IBM 360/91 (1967).

[BHA91] Bhandarkar and Clark, "Performance from Architecture: Comparing a RISC and a CISC with Similar Hardware Organization," *Proceedings of ASPLOS III, ACM/IEEE*, April 1991. Serious analysis of where the performance goes.

[HEN90] Hennessy and Patterson, *Computer Architecture: A Quantitative Approach*, Morgan Kaufman, San Mateo, CA, 1990. The classic book.

[MAS86] Mashey, "RISC, MIPS, and the Motion of Complexity," *Proceedings of UniForum*, 1986, Anaheim, CA., pp. 116–124. For the record, what the author was saying about RISC 5 years ago, for comparison.

[MAS91] Mashey, "CISCs are not RISCs, 1991 Edition," *MIPS Computer Systems*, September 1991. The presentation of which this article is a small subset. You can request a copy by sending e-mail with your paper mailing address to bdeprima@mips.com, or call 408/524-7012 and ask for a copy of the "CISCAnard" presentation.

[PRA89] N.S. Prasad, *IBM Mainframes: Architecture and Design*, McGraw-Hill, New York, 1989.

Superscalar PA-RISC

Continued from page 17

Special I/O drivers enable the external caches to operate at 100 MHz using 9-ns SRAMs. In addition, attention to SRAM characteristics and board layout are required. One benefit of integrating the FP unit and the integer unit was a reduction of the load on the SRAM data bus, which eases the timing problems slightly. HP expects to use multi-chip module packaging to reach even higher clock rates.

Where the Snakes FP unit could issue an instruction every two cycles and had a uniform latency of three cycles, the FP unit on the 7100 can issue every cycle and operation latency is reduced to two cycles.

The 7100 also incorporates a few small improvements that will increase general performance. While a miss is being processed, the data cache does not block further loads and stores that hit in the cache. Only when the data for a load miss is actually needed does the processor stall. On Snakes, a store followed by another cache access incurs a two-cycle penalty, but the penalty is only one cycle on the 7100. To speed some important operating system functions, the block-copy hint (don't allocate and zero the block being written) for the store instruction is implemented in the 7100.

Superscalar Capabilities

Like DEC's Alpha chip but unlike the SuperSPARC and 88110, the 7100 cannot issue two integer ALU instructions at the same time. The superscalar capabilities only allow an integer and an FP instruction to be issued together. Unlike Alpha, however, the 7100 does not have a separate branch or load/store unit, which is probably a consequence of using an existing integer unit design. For this processor, FP loads and stores are considered integer instructions. To reduce the time needed to decode multiple-issue opportunities, the I-cache stores a "pre-decode" bit with each instruction that indicates whether the instruction is destined for the integer or FP data paths.

Since PA-RISC has a composite multiply-add instruction, the peak execution rate at 100 MHz is 200 MFLOPS. The combination of the short, 2-cycle latency of add, subtract, and multiply (only one stall cycle is inserted between dependent operations) with the ability to issue an FP operation and an FP load or store together should make this implementation a floating-point screamer. Many FP applications are limited by operand bandwidth, and small, on-chip first-level caches just make the problem worse. A 100-MHz system using this chip with a 2-MB external data cache is likely to out-perform a 150-MHz Alpha implementation for FP applications even if the Alpha system has a large secondary cache. ♦