HEWLETT
PACKARD

Hewlett-Packard's 7100: A High-speed Superscaler PA-RISC Processor

Abstract: *A new processor implementing Hewlett-Packard's PA-RISC 1.1 (Precision Architecture) has been designed. A single chip implemented in a 0.8 micron 3-level metal CMOS technology includes the integer processor and a floating point coprocessor. The design operates at 100MHz and is the first superscalar PA-RISC design. The processor cache is a large configurable memory implemented with industry-standard SRAMs. High performance is achieved by high frequency operation and a variety of techniques used to reduce the average number of cycles per instruction.*

Introduction

The newest member of Hewlett Packard's PA-RISC (Precision Architecture) [1][2] processor family further extends the exponential performance growth which has been a characteristic of implementations of this architecture since the first systems were introduced in 1986. With this addition, the PA-RISC architecture now spans seven [3]-[9] different processor implementations, three different IC technologies, three computer system product lines and two operating systems, demonstrating the inherent scalability which was a primary objective of the original architecture definition.

The new processor design is highly leveraged from a previous PA-RISC design [3][9] which is currently employed in HP's high-performance Series 700 workstations and Series 800 and 900 multiuser computer systems. The primary goal of this design was to provide both a significant performance increase while decreasing the number of VLSI components. Like previous generations, this design uses high-performance circuit design techniques, packaging, and external cache memories. Major improvements in this generation include combining all CPU and floating point coprocessor functions onto a single chip, implementing a superscalar instruction execution pipeline, scaling the FET channel lengths to 0.8 microns allowing clock speeds of 100 MHz or greater, and several additional feature set improvements to reduce the effective memory latency and minimize pipeline stalls. The integration of the integer processor and the floating point coprocessor on a single die reduces the system component count, facilitates the implementation of a superscalar design, and also increases the processor frequency by reducing the load on the cache SRAM parts. The floating point coprocessor portion of the chip utilizes circuit design techniques developed by HP to achieve exceptional performance and density.

Processor Overview

The processor chip connects directly to an external 64-bit-wide instruction cache and an external 64-bit-wide data cache. These caches are direct mapped and are composed of industry- standard high-speed SRAMs which are cycled at the processor frequency. At 100MHz, each cache has a read bandwidth of 800MB/sec and a write bandwidth of 400MB/sec. The processor chip supports a D-Cache size from 4K up to 2Mbytes and an I-Cache size from 4K up to 1Mbyte. The cache addresses are hashed in order to reduce the-cache miss rate. Both caches are protected by parity, and I-Cache errors are corrected by hardware. A processor frequency of 100MHz is attainable due to improvements in both packaging and SRAM speeds.

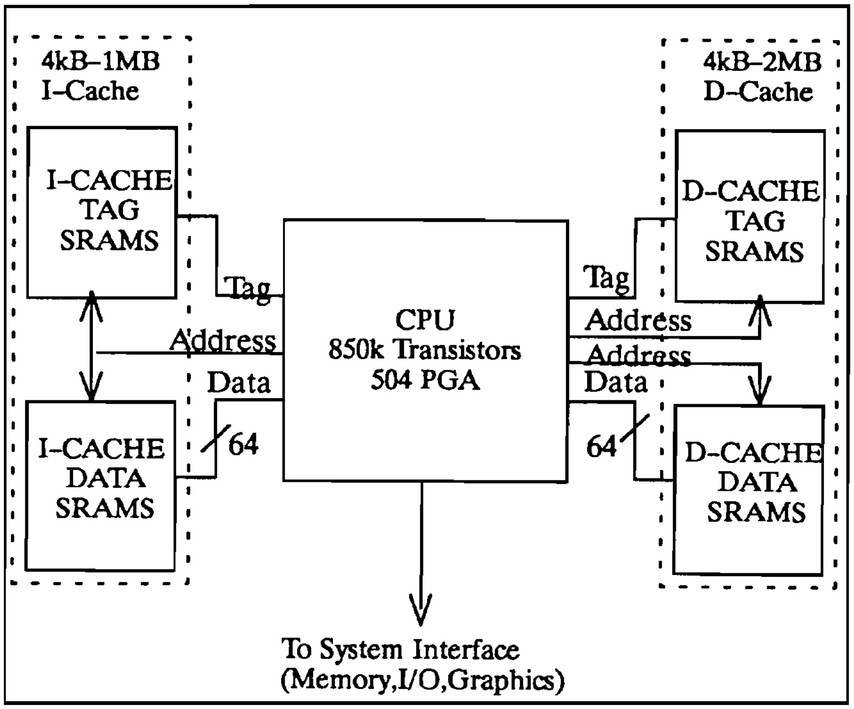
Use of off-chip caches instead of on-chip caches has many advantages. First, off-chip caches can be very large, which helps to achieve balanced performance across a wide range of applications. In addition, most processors with on-chip caches also require the additional cost of a secondary cache to achieve high-performance levels. Lastly, processors which do not cycle their secondary caches at the processor frequency suffer large first level cache miss penalties unless the on-chip cache is large.

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

The processor is connected to memory and I/O via a variable frequency synchronous bus. This bus is the same bus used in two existing PA-RISC processor designs [3][7][8][9]. Hence, this processor can be used with three existing-memory controllers as well as future memory controller designs. This provides for a wide variety of system configurations at various price/performance points. The processor can operate at three different frequency ratios with respect to the external bus - 1:1, 3:2, and 2:1. Operating the processor at a higher frequency than the external bus allows for compatibilitywith existing memory systems. A processor block diagram is shown in Figure 1.

Figure 1: Processor Block Diagram



CPU Overview

There are six major blocks on the processor chip (CPU). These are the 1) Integer Datapath, 2) Cache Control Datapath, 3) Floating Point Datapath, 4) Interface Circuits, 5) Unified TLB, and 6) Control Unit. The Integer Datapath includes the ALU, SMU (Shift-Merge Unit), branch adder, condition evaluation circuits, result bypass circuits, general-purpose register file, architected control registers, and address pipeline registers. The Cache Control Datapath contains registers for servicing cache misses/flushes and cache coherency checks. This datapath also includes the architected space (address segment) registers, a TLB look-aside buffer, and hashing hardware for the hardware TLB handler. The Floating Point

Figure 2: CPU Block Diagram

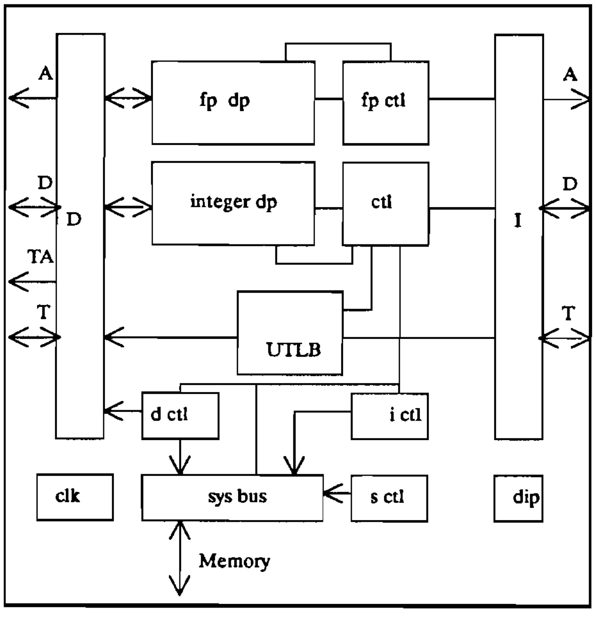
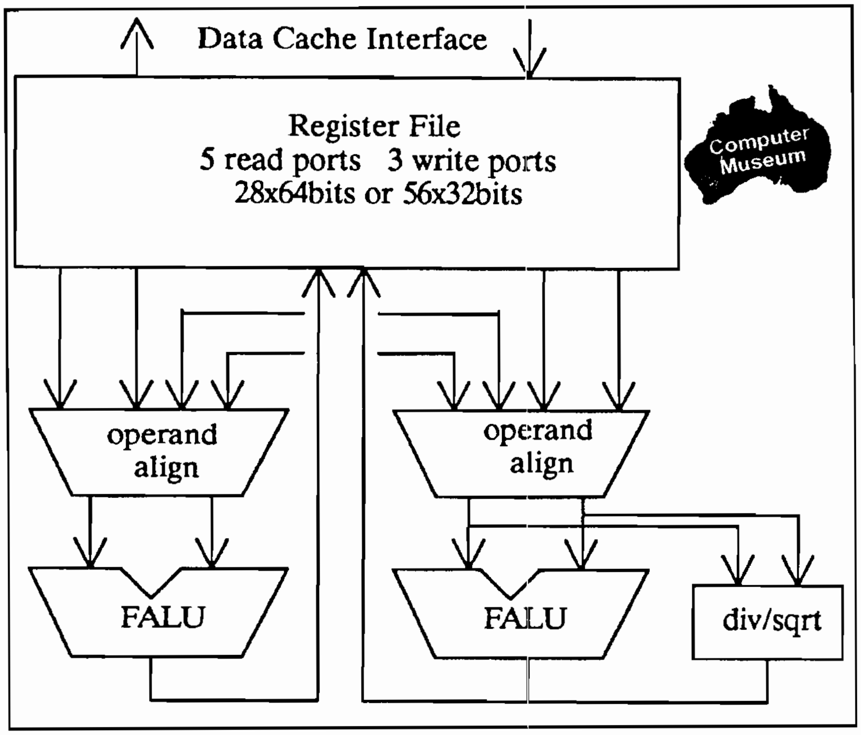


Table 1: Chip Specifications

Packaged Chip Specifications	
Chip Size	*1.4x1.4cm*
Transistors (small)	*850k*
Transistors (Floating Point)	*300k*
Transistors (Integer)	*550k*
Pads	*480*
Package	*504 PGA*

Figure 3: Floating Point Datapath



Computer Museum

Datapath contains the floating point multiplier, ALU, divide/ square root circuits, register file, and result bypass circuits. Interface Circuits include all circuits needed to interface to the I-Cache, D-Cache, and the memory bus. These circuits include register buffering and pipelining in addition to custom designed high-speed pad drivers and receivers. The Unified TLB contains 120 fixed-sized fully associative TLB entries as well as 16 variable-sized TLB entries. The Control Unit is actually many physically distinct circuitblocks implemented in PLAs, standard cell logic, and custom logic. The chip contains 850,000 transistors. Table 1 summarizes the packaged chip specifications. A block diagram of the CPU is shown in Figure 2.

The floating point datapath implements IEEE 754 compliant single- and double-precision math [10]. The multiplier uses a radix-4 Booth encoded array and also implements integer multiplication.

The divide/square root circuit uses an SRT radix-4 algorithm, but runs at 2X the processor clock frequency to achieve radix 16 performance. The FALU performs addition, subtraction, and format conversions. The register file has 28 64-bit registers, each of which can be used as two 32-bit registers for single precision operations. The register file has five read ports and three write ports in order to allow concurrent execution of a multiply, an add, and a load or store.The floating point datapath is implemented in 265k transistors and 32 square millimeters. A block diagram of the floating point datapath is shown in Figure 3.

High Speed Operation

The processor derives much of its performance improvement directly by increasing its clock frequency to 100MHz from the previous 66MHz design. Several developments combined to make this possible.

First, the CMOS device lengths were scaled algorithmically from 1.0u to 0.8u. This alone enabled the majority of existing circuits to be leveraged directly into the new design. Careful characterization of the 1u circuits,well in excess of the 66MHz design goal, identified circuit paths which would require modification to meet the 100MHz target in the scaled process. This information was incorporated into the new CPU design to ensure adequate performance.

Table 2: HP-CMOS26B Process Parameters

CPU Process Parameters	
Gate Oxide Thickness	160A
N-channel FET Leff (Typ)	0.61μm
N-Channel FET Ldrawn	0.75μm
P-channel FET Leff (Typ)	0.66μm
P-channel FET Ldrawn	0.75μm

Interconnect Pitch	Contacted	Uncontacted
Silicided Poly	2.6μm	2.0μm
Metal 1	2.6μm	2.0μm
Metal 2	2.6μm	2.0μm
Metal 3	5.0μm	5.0μm

Table 2 summarizes key parametrers of the process technology.

The cache interface was designed and verified directly by measurement of the earlier CPUs fabricated in the scaled process. Using this device it was possible to characterize a variety of combinations of packages, SRAMs, and processor configurations to establish the best options for cost and performance. One result of this investigation was the development of a new multi-layer ceramic pin grid array. In addition, a processor layout was obtained which was assured to achieve the100MHz goal. Further advanced packaging techniques can be used to package the CPUwith the cache SRAMs in order to achieve even greater processor frequencies.

Many new circuits were designed for the CPU. These were designed-to meet the 100MHz operating point in the scaled process. The floating point data path is the largest new circuit. It is composed of four functional blocks: a double precision multiplier and alu, a divide and square root unit which employs an SRT algorithm and an eight ported register file.

Circuit density was a prime concern for the floating point components. Early in the design it was realized that the highly parallelized algorithms commonly used in standalone coprocessor chip designs could not be compressed onto the CPU die. It was further recognized that fully combinatorial algorithms would be required for the FALU and multiply circuits to achieve the required level of performance. The solution was to use dynamic logic to exploit the well known speed and density-characteristics of that circuit type. Typical dynamic circuits are not capable of performing inverting logic without introducing race hazards. To overcome this problem a system of self-timed logic was devised. By this method, a multiplier and FALU were designed to compute full double precision results in 20ns.

Instruction Execution

The integer pipeline consists of six stages: I-Cache Read (IR), Operand Read (OR), Execute / D-Cache Read (DR), D-Cache Read Complete (DRC), Register Write (RW), and D-Cache Write (DW). The IR stage represents the instruction fetch portion of the pipeline. The implementation of dual instruction issue requires a small prefetch buffer which causes instructions to be fetched up to two cycles earlier than the IR stage when superscalar execution is not in effect. In the OR stage, all execution units decode their operands and start to calculate their results. In the DR stage the integer execution units complete their calculations. In addition, the D-Cache is read but data does not return until the DRC stage. Floating-point ADD and MULTIPLY results are also valid by the end of the DRC stage. General registers and floating point registers are written during the RW stage. Writes of the data cache for STORE instructions requires two processor cycles. The earliest two cycle STORE window occurs in the RW and DW stages. However, this is a sliding window because stores to the data cache only occur when another store instruction occurs. Floating point DIVIDE and SQUARE ROOT operations finish many cycles after the DW stage.

The pipeline was designed to maximize the amount of time allowed-to read the off-chip cache SRAMs. This maximizes the processor frequency for a given SRAM speed. All LOAD instructions execute in a single cycle and require only one cycle of D-Cache bandwidth. Since the instruction and data caches are accessed on separate busses, there is never any pipeline penalty due to conflicts between D-Cache accesses and instruction fetches.

The processor can execute one integer instruction and one floating point operation instruction in the same cycle. The I-Cache has enough bandwidth to support the continuous execution of two instructions per cycle. There are no alignment or order constraints on the pair of instructions which are executed together. In addition, there are no penalty cycles associated with switching between the execution of pairs of instructions and the execution of single instructions. Care was taken not to let the implementation of dual instruction issue limit the processor frequency. To achieve this, a dedicated predecoded bit in the I-Cache was implemented to steer instructions to the integer and floating point units. This predecoded bit minimizes the amount of decoding required to steer instructions properly.

The penalties for data and control dependencies for this pipeline are minimal. LOAD instruction execute in a single cycle except when the immediately following instruction uses the LOAD destination register. The compiler can usually avoid this one cycle penalty. A branch prediction algorithm is used in this processor to reduce the branch penalty. Forward branches are predicted untaken and backward branches are predicted taken to optimize the performance for loops. Correctly predicted branches execute in a single cycle.

The number of D-Cache cycles required for word and double-word STORE instructions has been reduced from three cycles to two cycles. In the previous processor design an extra cycle was necessary in order to read the cache tag to ensure a cache hit, and also to merge the old cache data with the data being stored. This processor uses a separate tag address bus to overlap the reading of a cache tag with the data write of the previous STORE instruction. In addition, separate write enable signals for each word of the data-cache eliminate the need to merge old cache data with new cache data for word and double-word STORE instructions. This algorithm requires SRAM writes for STOREs to occur only after it has been determined that the STORE was a cache hit and did not trap. This requires an extra stage of pipelining between the tag read and the data write. This pipelining does not cause any additional penalty cycles because bypass circuitry has been designed to forward pending STORE data to subsequent LOADs or sub-word STOREs. For this processor the pipeline penalty for word and double-word STORE instructions is zero cycles if the immediately following instruction is not a LOAD or STORE and one cycle otherwise. The penalty for a sub-word-store varies from zero to two cycles. Trace data has indicated that the great majority of STORE instructions are in fact word and double-word STOREs.

All floating-point operations except divide and square root are fully pipelined with a two cycle latency, in both single and double precision. The processor can issue an independent floating-point operation (flop) every cycle with no penalty cycles. Consecutive flops with a register dependency will incur a one-cycle penalty. Divides and square roots take 8 cycles in single precision and 15 cycles in double precision. Instruction execution does not stop for divide/sqrt until the resultregister is needed or another divide/sqrt is issued.

The processor can execute one integer instruction and one floating point instruction in parallel. Here "integer instruction" includes loads and stores of floating point registers and "floating-point instruction" includes the FMPYADD and FMPYSUB instructions. The FMPYADD and FMPYSUB instructions combine a multiply with an independent ALU operation which is executed in parallel. The peak execution rate is 200 MFlops, for a sequence of FMPYADD instructions in which adjacent FMPYADDs are register-independent.

The load-use penalty for floating-point operations is one cycle if the load and flop are adjacent but not issued together and two cycles if they are issued together. There is no penalty for stores using the result of a flop, even when they are executed in parallel.

Cache Miss Optimizations

D-Cache Optimizations

A number of features were implemented to reduce the penalty associated with D-Cache misses. The D-Cache miss penalty is minimized by four different techniques: 1) "Hit under Miss" for LOADs and STOREs, 2) Data-Cache "Streaming", 3) STORE cache hint to avoid copying a missing line into cache, and 4) semaphore operations in cache.

The "Hit under Miss" feature allows any type of instruction to execute while servicing a Data Cache miss. For a LOAD Miss, execution proceeds until the target register is needed as an operand for another instruction. The compiler can use this feature to prefetch into cache well before-the data is actually needed. For a STORE miss, execution proceeds until a load or sub-word store occurs to the missing line. The compiler can use this feature to execute instructions while storing results for previous calculations. During the latency of a LOAD or STORE cache miss, LOAD and STORE instructions with D-Cache hits can execute as well as normal integer and floating point operation instructions. During the latency period of a STORE miss, other STOREs to the same cache line can occur without a penalty. The processor maintains a scoreboard bit for each word in the cache line in order to prevent the copy fill from writing over the words written by the STORE instructions. The "Hit under Miss" feature applies to both integer and floating point LOADs and STOREs.

Instruction execution halts when the target register of a LOAD miss is needed as an operand of another instruction. The Data-Cache "Streaming" feature allows execution to continue once the critical word or double-word returns from memory. Thus, instruction execution can occur both during the latency period of the miss and during the copy fill period of the miss.

The PA-RISC 1.1 architecture defines a cache hint encoding ("block copy") for STORE instructions which indicates that hardware does not need to fetch the missing line from memory because software intends to write to the entire cache line. In this situation, the D-Cache is only accessed during the time it takes to copy out the old cache line at the same cache address (if it is dirty) and to write a new cache tag. This processor implements this cache hint feature for both privileged and non-privileged code. For non-privileged code the processor is designed to prevent a user from reading another process' data. However, the implementation of the cache hint allows the performance of non-privileged STORE hints to be equal to that of privileged STORE hints as long as the user does not try to read the line being stored. If the user code does not wish to STORE to the entire cache line the processor will read the original cache line from memory. As with STORE misses, the copy fill will not write over words written by the STORE instructions.This implementation of the "block copy" hint can greatly improve the performance of block copying and zeroing by the operating system. In addition, user- mode applications which perform many block-copy operations (e.g,. networking software) or code which performs operations on large arrays can achieve significant performance improvements.

The final enhancement to the D-Cache management is that this processor can perform the "Load and Clear" semaphore operation in cache rather than in memory. When the semaphore operation is performed in cache it executes with the same pipeline penalties as any other STORE instruction. This not only reduces the pipeline penalty but also reduces the traffic on the memory bus. This penalty is always much smaller than the penalty of performing the semaphore operation in main memory. The PA-RISC 1.1architecture also provides another cache hint encoding which eliminates the constraint that a semaphore operation must be synchronized with I/O devices. Using this cache hint in a multi-processor system will increase the opportunities for processors to perform the semaphore operation in cache.

I-Cache Optimizations

The I-Cache control on the processor allows instruction execution to resume after an I-Cache miss as soon as the missing instruction returns from memory. Instructions are executed as they are being copied into the I-Cache as long as a taken branch does not occur.

The 64-bit path to I-Cache allows the write bandwidth for copy fill to match the maximum bandwidth of the external memory bus which is 400Mbytes/sec at 100MHz.

TLB Optimizations

The number of penalty cycles for TLB misses has been minimized by a number of design features. First, a unified 120-entry fully associative first-level TLB is implemented on the processor. In addition, 16 additional entries can each be programmed to map a variable-sized region of memory. Each variable-sized entry can map 512k-64M of memory. Mapping large areas of the OS and graphics frame buffers reduces the overall TLB miss rate. In addition, a second level variable-sized table is maintained by software but read by the processor for every first-level TLB miss. A first-level miss with a second-level hit can execute in just 10 cycles. The hardware TLB handler also saves the memory address of the next softwareentry in order to reduce the miss penalty when the second level TLB misses. Two other features were also designed to reduce the penalty for TLB misses that end up being serviced by software. First, a set of "shadow" general registers is provided to avoid a penalty for saving and restoring general registers in the TLB miss trap handler. Second, a new implementation specific TLB insert instruction was implemented to eliminate more cycles from the-software TLB miss penalty.

Since this processor does not contain a separate ITLB, a one-entry-look-aside buffer is used to avoid conflicts between data accesses and instruction accesses. Special care was made to minimize the penalties associated with updating the look-aside buffer from the UTLB. Trace data has shown that most look-aside buffer updates are due to just three types of branch instructions. For these branch instructions the processor was designed to overlap the already existing one cycle branch penalty with the look-aside buffer update. For all other look-aside buffer updates, there is only a one-cycle penalty. Hence, the overall penalty for look-aside buffer misses is almost negligible.

Multiprocessing

The processor is designed to operate in two types of multiprocessor-configurations. In the first configuration, each processor connects to an interface chip which snoops on all transactions on the main memory bus. In such a system the interface chip maintains cache coherency by issuing transactions to the processor. The "private," "dirty," and "valid" states of each cache line of the write-back D-Cache are modified according to the transactions issued and received by the processor.

The second type of multiprocessor configuration supported by this processor consists of two processors connected to the same local memory bus as the memory /I/O controller. Such a configuration requires no interface chips and is compatible with an existing memory system. Cache coherency is maintained by snooping on the local memory bus. Cache-to-cache line transfers are performed with no assistance or interference from the memory / I/O controller. This configuration provides for a very low-cost high-performance multiprocessor system.

Graphic Support

All the features which were specifically targeted for improving graphics performance on the current Series 700 processor are also implemented on this processor. Graphics enhancements for PA-RISC workstations are discussed in a separate paper [11]. The processor support for enhanced graphics performance includes block move, z-buffering, color-interpolation, graphics clip test instructions, and floating point STORE and LOAD instructions to/from I/O space. Superscalar execution and improved floating point latencies combined with these features will provide extremely high graphics performance.

Design Methodology

The design of the CPU is best described as structured custom. Chip control is implemented with a combination of programmable logic arrays and automated place and route tools. Some use was made of logic synthesis tools. Global signal interconnect was produced by automated routing tools. The design cycle was thirteen months from initial staff through tape release.

Conclusions

This newest addition to HP's high-performance RISC processor family provides further evidence contrary to the often stated predictions that the performance of RISC processors is at or nearing fundamental physical limits. In fact, significant performance increases are achieved even though combined with both cost reduction goals and a very short development cycle time through the use of detailed analysis and rational engineering tradeoffs. Relative to other RISC designs, the most notable contributions of this processor implementation include:

- very high clock frequencies in low-cost production CMOS technology made possible by efficient high-speed circuit and package designs

- large, high-bandwidth external cache memories to eliminate sustained performance bottlenecks associated with small on-chip caches or hierarchical caches

- a highly efficient instruction-pipeline implementation capable of executing up to two instructions per clock cycle with infrequent interlocks, and several features to minimize cache and TLB miss penalties

- exceptionally high floating-point performance due to high-frequency, low-latency, pipelined floating-point units and superscalar execution

- full support for two different types of multiprocessor implementations

- a straightforward growth path to higher performance future implementations by taking advantage of VLSI and packaging technology advances as well as design evolution to further reduce the cycles per instruction.

Authors

Eric DeLano, Will Walker, Jeff Yetter, and Mark Forsyth

Acknowledgements

The authors wish to acknowledge the contributions of the many individuals who participated in this project. This includes design and verification work done by HP's Engineering Systems Laboratory in Fort Collins, CO and verification done by Workstations Systems Division in Cupertino, CA. Finally, the authors wish to thank William Jaffe, Robert Miller, Darius Tanksalvala, John Wheeler, and Alan Wiemann for providing the atmosphere and leadership which made the project possible.

References

[1] M. J. Mahon, R. B. Lee, T. C. Miller, and W. R. Bryg, "Hewlett Packard Precision Architecture: The Processor," Hewlett Packard Journal, Vol. 37, pp4-21; August, 1986.

[2] R. B. Lee, "Precision Architecture," IEEE COMPUTER, Vol. 22 pp78-91; January, 1989.

[3] M. Forsyth, S. Mangelsdorf, E. DeLano, C. Gleason, J. Yetter, and D. Steiss, "CMOS PA-RISC Processor for a new family of Workstations," Compcon Spring 91:Digest of Technical Papers (Feb 91).

[4] J. Yetter, M. Forsyth, W. Jaffe, D. Tanksalvala, and J. Wheeler, "A 15 MIPS 32b Microprocessor," ISSCC DIGEST OF TECHNICAL PAPERS, pp26-27; February, 1987.

[5] M. Forsyth, W. S. Jaffe, D. Tanksalvala, J. Wheeler, and J. Yetter, "A 32-bit VLSI CPU with 15 MIPS Peak Performance," IEEE JOURNAL OF SOLID-STATE CIRCUITS, pp768-775; October, 1987.

[6] B. D. Boschma, D. M. Burns, R. Chin, N. S. Fiduccia, C. Hu, M. J. Reed, T. I. Rueth, F. X. Schumacker, and V. Shen, "A 30 MIPS VLSI CPU,"P ISSCC DIGEST OF TECHNICAL PAPERS, pp82-83; February, 1989.

[7] D. Tanksalvala, et al, "A 90 MHz CMOS RISC CPU Designed for Sustained Performance," ISSCC DIGEST OF TECHNICAL PAPERS, pp52-53; February, 1990.

[8] J. Lotz, B. Miller, E. DeLano, J. Lamb, M. Forsyth, and T. Hotchkiss, "A CMOS RISC CPU Designed for Sustained High Performance on Large Applications," IEEE JOURNAL OF SOLID-STATE CIRCUITS, pp1190-1198; October, 1990.

[9] C. Gleason, M. Forsyth, C. Kohlhardt, S. Mangelsdorf, B. Arnold, and R. Luebs, "CMOS Processor Circuit Design in Hewlett-Packard's Series 700 Workstations," Proceedings of the IEEE 1991 International Conference on Computer Design; October, 1991.

[10] ANSI/IEEE Standard 754-1985 for Binary Floating-Point Arithmetic; IEEE Computer Society Press, Los Alamitos, CA, 1985.

[11] C. Dowdell and L. Thayer, "Scalable Graphics Enhancements for PA-RISC Workstations,"Compcon Spring 92:Digest of Technical Papers (Feb 92).

Computer Museum

HEWLETT
PACKARD

For more information, call 1-800-637-7740. In Canada, call 1-800-387-3867. Or contact your local HP sales office or authorized HP reseller.

United States:
Hewlett-Packard Company
4 Choke Cherry Road
Rockville, MD 20850
(301) 670 4300

Hewlett-Packard Company
5201 Tollview Drive
Rolling Meadows, IL 60008
(708) 255 9800

Hewlett-Packard Company
5161 Lankershim Blvd.
North Hollywood, CA 91601
(818) 505 5600

Hewlett-Packard Company
2015 South Park Place
Atlanta, GA 30339
(404) 955 1500

Hewlett-Packard Company
300 Apollo Drive
Chelmsford, MA 01824
(508) 256 6600
FAX: (508) 256 2384

Canada:
Hewlett-Packard (Canada) Ltd.
6877 Goreway Drive
Mississauga, Ontario L4V 1M8
Canada
(416) 678 9430

Latin America:
Hewlett-Packard
Latin America Region Headquarters
Monte Pelvoux No. 111-2nd Floor
Lomas de Chapultepec
11000 Mexico, D.F.
(525) 202 0155

Japan:
Yokogawa-Hewlett-Packard Ltd.
Kohrakuen Shinjuku Bld.
15-7, Nishi Shinjuku 4 Chome
Shinjuku-ku, Tokyo 160
Japan
+81 3/5371 1351

Asia Pacific:
Hewlett-Packard Asia Pacific Ltd.
22/F Bond Centre
West Tower
89 Queensway, Central
Hong Kong
+852/ 848 7777

Australia/New Zealand:
Hewlett-Packard Australia Ltd.
31-41 Joseph Street
Blackburn, Victoria 3130
Melbourne, Australia
+61 3/272 2895

European Headquarters:
Hewlett-Packard S.A.
Route du Nant-d'Avril 150
CH-1217 Meyrin 2/Geneva
Switzerland
+41/22/780 8111

Austria:
Tel:+43/222/2500 0

Belgium:
Tel:+32/2/761 3400

Czechoslovakia:
Tel:+42/2/471 73 21

Denmark:
Tel:+45/45/99 10 00

Finland:
Tel:+358/0/88 721

France:
Tel:+33/1/698 260 60

Germany:
Tel:+49/6172/16 0

Greece:
Tel:+30/1/682 88 11

Hungary:
Tel:+38/61/5584 72

Iceland:
Tel:+354/1/67 10 00

Ireland:
Tel:+253/12/88 33 99

Italy:
Tel:+39/ 2/92 19 91

Netherlands:
Tel:+31/ 20/54 769 11

Norway:
Tel:+47/2/87 97 00

Poland:
Tel:+48/22/36 83 00

Portugal:
Tel:+35/1/130 173 30

Russia:
Tel:+007/95/923 50 01

Slovenia:
Tel:+38/61/55 84 72

Spain:
Tel:+34/1/626 16 00

Sweden:
Tel:+48/ 8/750 20 00

Switzerland:
Tel:+41/ 57/31 31 11

Customer Information Center:
Tel:+41/46/05 15 05

Turkey:
Tel:+901/1/175 29 70

United Kingdom:
Tel:+44/ 344/360 000

For Countries not listed:

European Multicountry Sales Region:
Tel:+41/22/780 81 11

Middle East and Africa Operations:
Tel:+41/ 22/780 71 11

Copyright © Hewlett-Packard Co., 1992

Printed in U.S.A. 8/92
5091-5386E