A 600-MHz Superscalar Floating-Point Processor

J. Clouser, M. Matson, R. Badeau, R. Dupcak, S. Samudrala, R. Allmon, and N. Fairbanks

***Abstract*—The floating-point unit of a 600-MHz, out-of-order, superscalar RISC Alpha microprocessor is described. The unit achieves 59 SpecFP95 and can transfer register data at up to 9.6 GB/s. It has two independent pipelines for multiply and add/subtract operations, with iterative divide and square-root circuits, and is fabricated in a 2.2-V, 0.35-μm CMOS process.**

I. Introduction

THE floating-point unit of the Alpha 21 264 [1], a 600-MHz, out-of-order, superscalar RISC microprocessor, attains 59 SPECfp95 in early systems. In comparison to its predecessor [2], the unit has higher performance function unit architectures, fewer gate delays per cycle, a higher load/store bandwidth, and more registers, and implements a new square-root instruction. It contains 580 000 transistors and dissipates approximately 9.6 W, measuring 2.1×9.6 mm^2 in a 0.35-μm, 2.2-V, planarized CMOS process with six metal layers (four signal layers and two reference planes). A photomicrograph appears in Fig. 1.

The design was optimized for circuit speed while restricting power consumption. The 1.7-ns cycle time allows a dozen gate delays per clock period, mandating techniques such as latchless pipelining, fully decoded dynamic logic, and low swing signals. Reduced supply voltage and conditional clocking decreased power consumption; low-power complementary CMOS circuits were used where speed was not critical. The multiple conditional clocks led to race issues that were carefully controlled by minimizing clock skew and exhaustive delay and edge rate extraction with an in-house analysis tool [3].

In addition, the design posed new challenges due to process scaling and a requirement for low-frequency operation. At the reduced feature sizes, transistor performance improved more than did the interconnect delays; moreover, the effective interconnect distances and lateral capacitive coupling increased. Furthermore, although the scaled transistors offered better delays, their gate-to-source/drain overlap capacitances were proportionally larger, exacerbating undesirable coupling and charge-share effects. Subthreshold conduction also increased, potentially disrupting dynamic circuits, particularly at low clock frequencies.

II. Major Functional Units

The block diagram (Fig. 2) shows the unit's multiple clock generators, register file, load/store unit, add/subtract/convert pipeline, multiplier pipeline, iterative divider, and iterative square root. These sections can simultaneously receive and execute a multiply, an add/subtract/convert, and up to two load/store instructions from the processor's floating-point instruction queue. Data can be transferred directly between integer and floating-point register files. Both divide and square-root instructions can execute concurrently with the foregoing instructions. The multiplier has its own final adder that includes rounding circuitry, while the add/subtract/convert, divider, and square-root sections share a common final adder.

Fig. 1. Photomicrograph.

A. Clock Generation and Distribution

The Alpha 21 264 has a single-node global clock, called GCLK [4]. The floating-point unit buffers GCLK to form local, conditional clocks that control pipeline stages. Power is saved by not clocking idle stages, clock skew is diminished, and stage timing may be adjusted. However, these distinct clocks introduce signal races between adjacent stages. In the floating-point unit, the ratio of the signal and clock delays is improved by sharing a common GCLK_L wire among clock generators, as shown in Fig. 3. The clocks are distributed in coarse metal and have dispersion delays under 60 ps.

Manuscript received November 2, 1998; revised January 20, 1999.

The authors are with Compaq Computer Corp., Shrewsbury, MA 01545 USA.

Publisher Item Identifier S 0018-9200(99)04732-0.

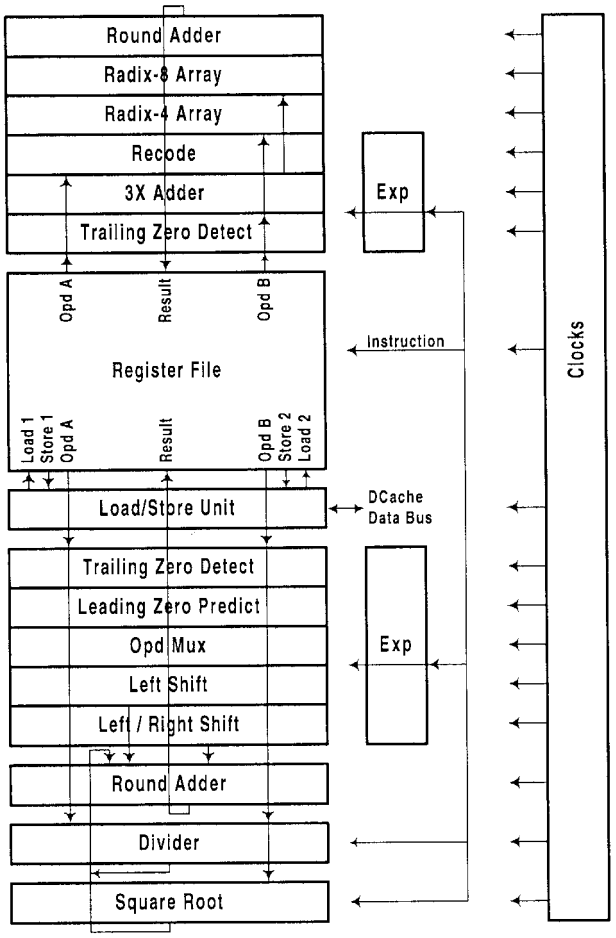

Fig. 2. Block diagram.

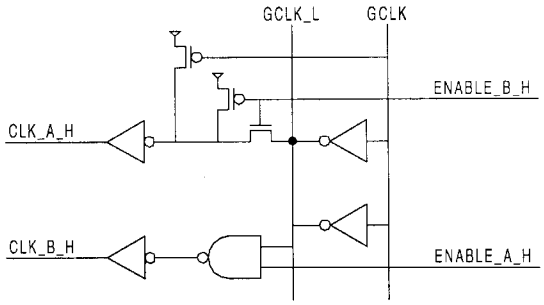

Fig. 3. Conditional clocks.

B. Register File

The register file contains 72 physical registers, each having a 64-bit data field, a zero bit, and a condition bit. There are six read ports and four write ports. To improve speed and power consumption, all read bit lines have low voltage swings. The read ports for the multiply and add/subtract/convert pipelines are differential; those for the store ports are single ended. The write ports are all single ended.

Single-ended ports were preferred because they reduced the datapath bit pitch, conserving area. Several techniques improved the speed of these single-ended accesses. For the store ports, larger devices discharge the bit lines sufficiently to permit single-ended sensing. The store-port sense amplifier

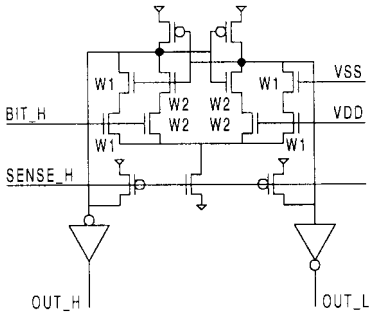

Fig. 4. Store-port sense amplifier.

(Fig. 4) is sized symmetrically for better immunity to process variations. The trigger voltage was chosen by disabling part of the reference side's discharge path, avoiding a nonrail reference voltage with its accompanying noise and power considerations. For the write ports, operand multiplexors forward result and load data directly to the function units, bypassing the register file. Consequently, the single-ended writes do not impose a performance penalty.

Managing noise on the read ports' bit lines was crucial to achieving a short cycle time. The single-ended bit lines were routed alongside the power rails. The differential lines were routed with a three-way bit-twisting scheme, which ensures that the majority of the capacitive coupling is common mode.

C. Add/Subtract/Convert Pipeline

The add/subtract/convert pipeline executes add, subtract, convert, and conditional move instructions. A new instruction can be processed every cycle, and the latency is four cycles. The pipeline also provides operands for the square-root unit and divider and processes their results.

Extensive use of parallelism and early normalization prediction made the four-cycle latency possible. Consider the two basic add/subtract flows. In the typical case, the operands' exponent difference is calculated and the smaller operand is shifted downwards. Rounding bits are injected, and the aligned operands are added. However when subtracting operands of comparable magnitude, the subtraction may clear many of the upper bits. Traditionally the result is corrected by a postalignment shift and then rounded; the 21 264's short cycle time precluded this approach. Instead, a dual operand shifter configuration is used, where leading zero predictors anticipate how many of the upper bits will be cleared, and both operands are shifted upward prior to the actual subtraction. This approach has speed, power, and area advantages over the traditional scheme.

D. Multiplier Pipeline

The multiplier pipeline also has a four-cycle latency. Several techniques help make optimal use of the available time. A hybrid recoder computes radix-4 recodes for the first partial products and radix-8 recodes for later ones; the radix-4 recodes allow summations to proceed while a dedicated adder computes $3\times$ the multiplicand, required by the radix-8 logic. The adder exploits properties of the $3\times$ addition to achieve

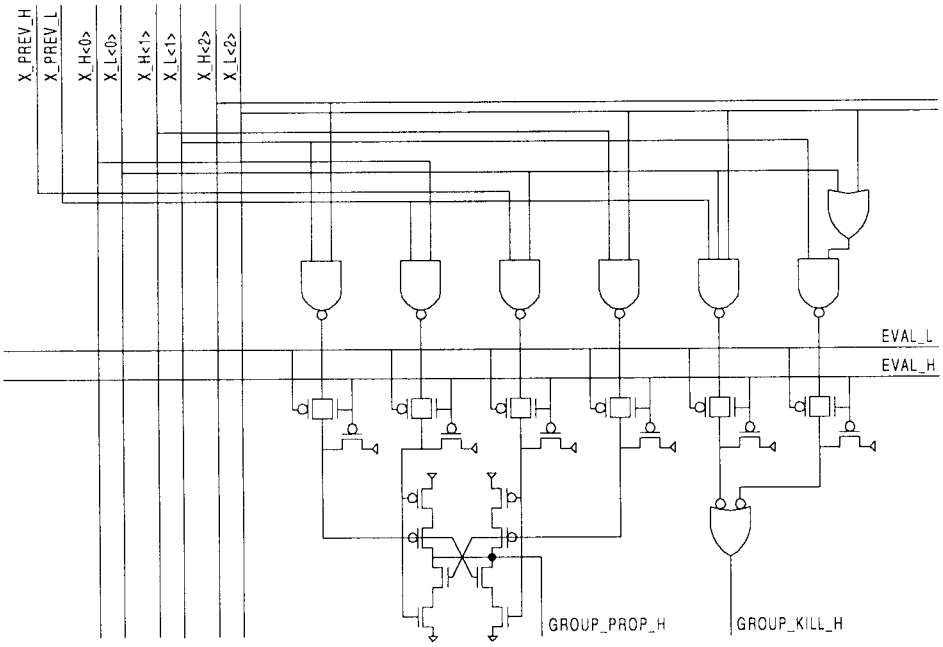

Fig. 5. A 3× carry lookahead.

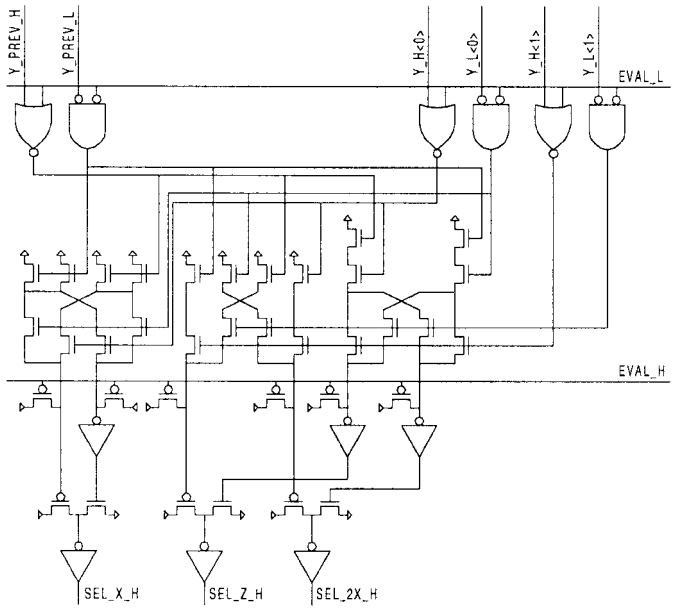

Fig. 6. Radix-2 Booth recoder.

higher speeds and more efficient area utilization as compared to a standard, general-purpose adder. This can be seen in a subsection of the adder's lookahead circuit (Fig. 5) where the propagate and kill logic has been simplified and the delay shortened relative to standard implementations. The path delay of the recoders (Fig. 6) is correspondingly improved by merging the logic, latches, and drivers.

The partial products are added in a latchless pipelined, carry-save-adder array using two parallel summation paths. Differential domino circuits form the array. To meet the cycle time constraint, the array does not use latches to pipe the intermediate sums and carries forward. Instead, the precharges for the domino stages are interlocked so that the receiving, opposite phase domino stages have evaluated before their inputs are precharged away; on the following phase, the receiving stage precharges before next-phase input data arrives. Rounding bits are injected into the final stage of the array, and the resulting carry and sum vectors are then added.

E. Divider

The divider uses a reciprocal and quotient approximation method. In each iteration, a new partial remainder is formed by subtracting the product of the new quotient digit and the divisor from the previous remainder. The next quotient digit is the product of the upper bits of the remainder and the approximate reciprocal. The new remainder is used for the multiplication in the next iteration. The divider performs one iteration per cycle, generating eight quotient bits per iteration; it uses the add/subtract/convert pipeline for final rounding and has latencies of 12 and 15 cycles for single and double precision results, respectively.

There are several keys to meeting the short cycle time. First, the new remainder is created in one phase by a carry-save-adder array constructed as a modified Wallace tree multiplier. Second, when this new remainder's upper bits become the multiplier in the next iteration, the carry and sum vectors are directly recoded rather than first passing through a time-consuming carry propagate addition. This recoder converts the upper bits of the carry and sum vectors to a radix-4 representation in the second phase of the iteration. Third, all latches were removed from the iteration loop using differential domino logic similar to that in the multiplier.

The divider's circuitry illustrates the effect of the technology's increased subthreshold conduction and gate overlap capacitance. The carry-save adder (Fig. 7) shows the enhancements needed to remedy the ensuing leakage and charge-redistribution problems. Cross-coupled pMOS devices

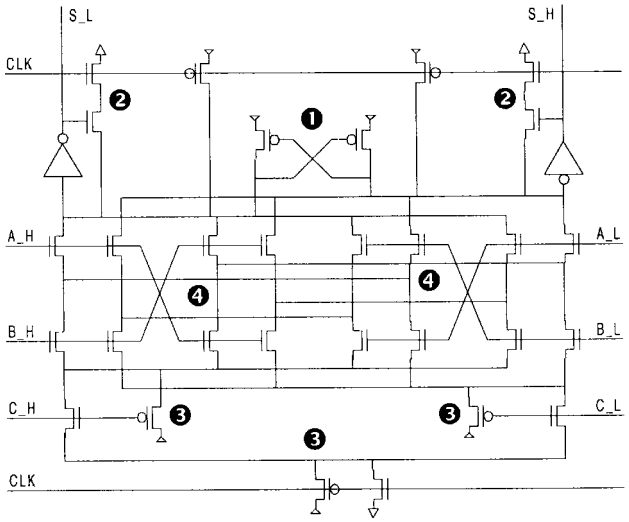

Fig. 7. Corrected carry-save-adder cell.

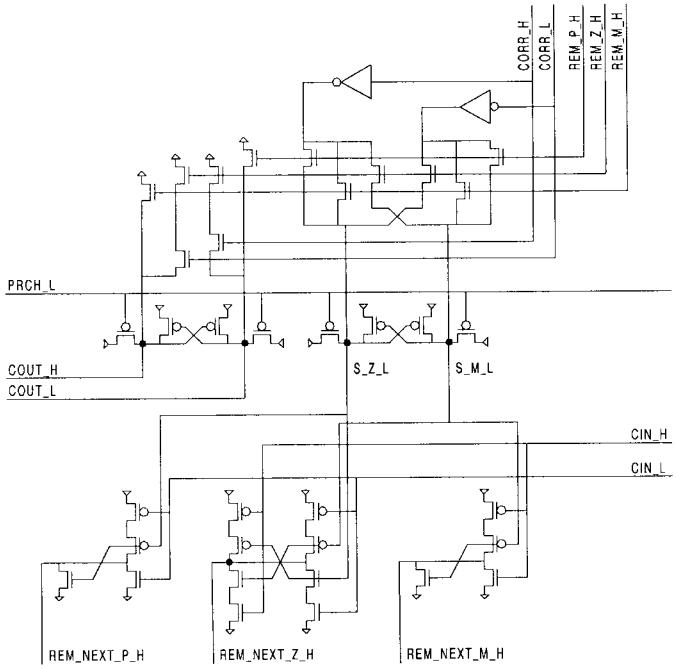

Fig. 8. Signed digit plus binary adder.

1) sustain precharged levels, while nMOS devices 2) sustain low levels after the inputs from opposite phase domino logic have returned to zero. Other pMOS devices 3) diminish charge redistribution. Finally, series connections have been split into two parallel paths 4) with their input ordering exchanged, reducing charge redistribution for this particular logic function.

F. Square Root

The square-root unit uses a "completing the least squares" algorithm, an iterative procedure that repetitively diminishes the partial remainder. Each iteration is accomplished in three stages of domino logic, fast enough for a single clock phase. Two iterators are cascaded to generate two root bits per cycle; this is more efficient than a single, more complex, radix-4 iterator. A compact layout was essential to satisfy area constraints. The latencies are 18 and 33 for single and double precision results, respectively.

In each iteration, the upper bits of the partial remainder are examined to determine the next root digit and then create a corresponding correction term. The correction—either positive or negative—is added to the current partial remainder to form the next partial remainder. A signed digit plus binary adder (Fig. 8) produces a fully decoded remainder, facilitating the correction choice. For greater speed, latches were eliminated from the critical path. Instead, the precharges for the domino stages are interlocked in the fashion described previously.

III. Conclusion

Innovative design techniques enabled the Alpha 21 264 to attain industry-leading performance levels. This achievement was made possible by simultaneously optimizing the design across the microarchitectural, logic, circuit, and layout domains. The unit was fully functional on first-pass silicon.

Acknowledgment

The authors gratefully acknowledge the contributions of B. Grundmann, M. Lamere, D. Brown, M. Gaetz, B. Gieseke, J. Heath, J. Krause, K. Lu, R. Matthew, D. Sarrazin, S. Sayadi, and many other members of the Alpha 21 264 team who participated in this project.

References

[1] B. Gieseke, R. Allmon, D. Bailey, B. Benschneider, S. Britton, J. Clouser, H. Fair III, J. Farrell, M. Gowan, C. Houghton, J. Keller, T. Lee, D. Leibholz, S. Lowell, M. Matson, R. Matthew, V. Peng, M. Quinn, D. Priore, M. Smith, and K. Wilcox, "A 600 MHz superscalar RISC microprocessor with out-of-order execution," in *ISSCC Dig. Tech. Papers*, Feb. 1997, pp. 176–177.

[2] J. Kowaleski, Jr., G. Wolrich, T. Fischer, R. Dupcak, P. Kroesen, T. Pham, and A. Olesin, "A dual-execution pipelined floating-point CMOS processor," in *ISSCC Dig. Tech. Papers*, Feb. 1996, pp. 358–359.

[3] E. Shriver, D. Hall, N. Nassif, N. Rahman, N. Rethman, G. Watt, and J. Farrell, "Timing verification of the 21 264: A 600 MHz full-custom microprocessor," in *Proc. 1998 IEEE Int. Conf. Computer Design*, Oct. 1998, pp. 96–103.

[4] D. Bailey and B. Benschneider, "Clocking design and analysis for a 600-MHz Alpha microprocessor," *IEEE J. Solid-State Circuits*, vol. 33, pp. 1627–1633, Nov. 1998.