A 110 GOPS/W 16-bit Multiplier and Reconfigurable PLA Loop in 90-nm CMOS

Steven K. Hsu, *Member, IEEE*, Sanu K. Mathew, *Member, IEEE*, Mark A. Anders, *Member, IEEE*, Bart R. Zeydel, *Member, IEEE*, Vojin G. Oklobdzija, *Fellow, IEEE*, Ram K. Krishnamurthy, *Senior Member, IEEE*, and Shekhar Y. Borkar, *Member, IEEE*

***Abstract*—This paper describes a 16 × 16 bit single-cycle 2's complement multiplier with a reconfigurable PLA control block fabricated in 90-nm dual-V_t CMOS technology, operating at 1 GHz, 9 mW (measured at 1.3 V, 50 °C). Optimally tiled compressor tree architecture with radix-4 Booth encoding, arrival-profile aware completion adder and low clock power write-port flip-flop circuits enable a dense layout occupying 0.03 mm^2 while simultaneously achieving: 1) low compressor tree fan-outs and wiring complexity; 2) low active leakage power of 540 μW and high noise tolerance with all high-V_t usage; 3) ultra low standby-mode power of 75 μW and fast wake-up time of <1 cycle using PMOS sleep transistors; 4) scalable multiplier performance up to 1.5 GHz, 32 mW measured at 1.95 V, 50 °C, and (v) low-voltage mode multiplier performance of 50 MHz, 79 μW measured at 570 mV, 50 °C.**

***Index Terms*—Booth encoding, flip-flop, multiplier, programmable logic array (PLA), radix-4, reconfigurable, sleep transistor, 2's complement.**

I. Introduction

SHORT bit-width (<16 b) 2's complement multipliers with single-cycle throughput and latency are essential ingredients of high-performance embedded processor and DSP execution cores. Parallel clusters of multiplier/multiply–add/multiply–accumulate cores are required to perform complex SIMD and filter operations while consuming ultra low energy/operation [1]. Key components of many DSP algorithms, such as finite-impulse response (FIR) filters, infinite-impulse response (IIR) filters, discrete cosine transforms (DCTs), and fast Fourier transforms (FFTs), consist of repetitive multiplication operations that equate to over half of the total operations. These constraints require an energy-efficient multiplier with a compact layout footprint that enables low compressor tree fan-outs and minimizes the wiring complexity in the multiplier core.

Several traditional parallel multiplier schemes improve the speed proportional to the log of the operand length, such as Wallace and Dadda carry–save trees. A Wallace tree [2] requires $log_{3/2}(N/2)$ levels of (3:2) counters to reduce the N inputs down to two carry–save redundant form outputs, where the (3:2) counter converts three inputs into two-count encoded outputs. In the Dadda tree [3], the number of counters in a compression tree is minimized. A higher order (4:2) compressor by Weinberger [4] requires $log_2(N/2)$ levels of compressors. This type of partial product tree reduces the number of bits and simplifies the internal horizontal routing within the multiplier. Beyond (4:2) compressors, even higher order (9:2) compressor based partial produce trees show delay improvements [5]. Further improving the multiplier delay, Oklobdzija *et al.* [6] developed a three-dimensional optimization method (TDM) which appropriately connects fast/slow inputs and slow/fast outputs. This optimizes the tiles of a partial product tree as one Nth-order compressor instead of individual smaller order compressors, finding a global optimum rather then a local optimum.

In this paper, a single-cycle 16-bit multiplier and reconfigurable programmable logic array (PLA) control engine [7] fabricated in 90-nm dual-V_t CMOS technology [8] is described. A radix-4 Booth encoding, optimally tiled partial product tree and a hybrid completion adder are employed to improve power efficiency in the multiplier. A fully static CMOS multiplier design enables low clock power, low active leakage and dynamic power consumption, high DC noise robustness, and a dense layout. The 16-bit multiplier operates at 1 GHz measured at 1.3 V, 50 °C and consumes 9-mW total power. Multiplier performance is scalable up to 1.5 GHz measured at 1.95 V, 50 °C consuming 32 mW. During low-voltage mode, the multiplier is scalable down to 50 MHz measured at 570 mV, 50 °C consuming 79 μW. Write-port flip-flops are used throughout the chip to reduce active power even further. PMOS sleep transistors power gate the virtual supply with the nominal supply to reduce the standby leakage power enabling single cycle wake up from sleep mode.

The remainder of this paper is organized as follows. Section II describes the organization of the multiplier and reconfigurable PLA loop; Sections III and IV present the architecture and circuits of the dynamic reconfigurable PLA and energy-efficient multiplier; the write-port flip-flops are discussed in Section V; Section VI discusses the benefits of this design over a conventional Wallace tree implementation; Section VII presents the 90-nm dual-V_t CMOS implementation and silicon measurement results; the operation of the PMOS sleep transistor is described in Section VIII. Finally the paper is summarized in Section IX.

Manuscript received May 13, 2005; revised September 6, 2005.

S. K. Hsu, S. K. Mathew, M. A. Anders, R. K. Krishnamurthy, and S. Y. Borkar are with the Circuits Research Laboratories, Intel Corporation, Hillsboro, OR 97124 USA (e-mail: steven.k.hsu@intel.com; sanu.k.mathew@intel.com; mark.a.anders@intel.com; ram.krishnamurthy@intel.com; shekhar.borkar@intel.com).

B. R. Zeydel and V. G. Oklobdzija are with the Advanced Computer Systems Engineering Laboratory, Department of Electrical and Computer Engineering, University of California, Davis, CA 95616 USA (e-mail: brzeydel@acsel-lab.com; vojin@acsel-lab.com).

Digital Object Identifier 10.1109/JSSC.2005.859893

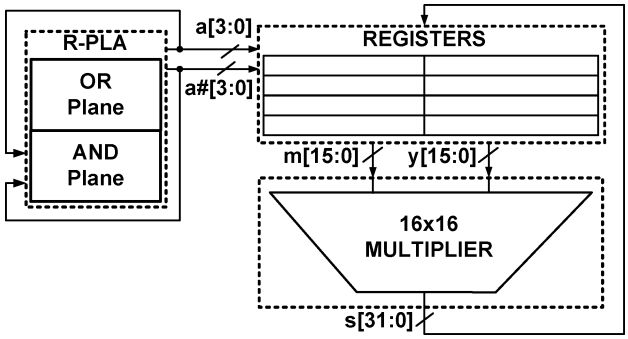


Fig. 1. 16 bit multiplier and reconfigurable PLA loop organization.

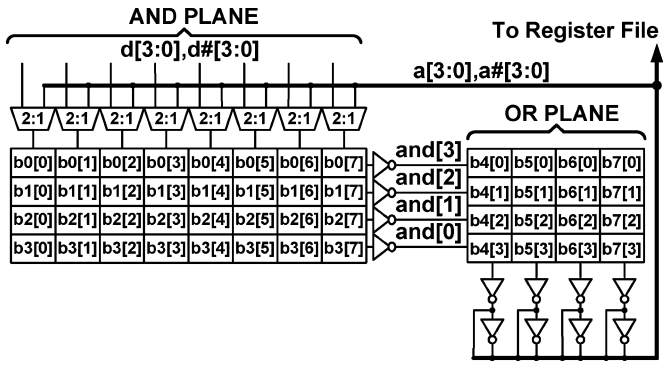


Fig. 2. 4-minterm 4-input/4-output reconfigurable PLA control engine.

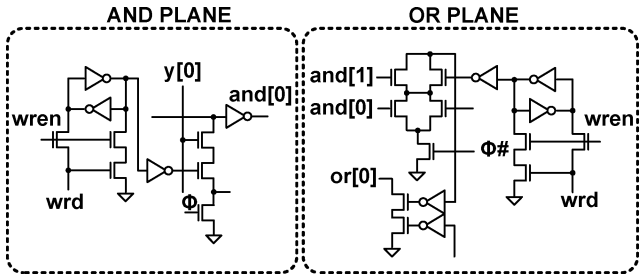


Fig. 3. Reconfigurable PLA AND/OR plane circuits.

II. Multiplier and Reconfigurable PLA Loop Organization

The multiplier and PLA loop is designed for efficient implementation of single-cycle back-to-back multiplications, typically found in DSP operations, such as, DCTs, FFTs, and FIR and IIR filters. These operations involve scaling (i.e. multiplying) the input data stream with an array of coefficients and accumulation of the product over several cycles. Since the value and periodicity of the coefficients are fixed for a given DSP algorithm, these coefficients can be stored in an on-chip memory, with a programmable memory read access pattern that delivers the appropriate coefficient to the multiplier at the corresponding cycle. During configuration time, the coefficient array is loaded into memory and a reconfigurable control block is programmed to implement a finite-state machine. The outputs of this finite state machine provide the memory read addresses in a predetermined cyclical pattern. Such an organization enables single-cycle throughput for multiply–accumulate operations required in matrix multiplications, digital filters, discrete cosine and fast Fourier transforms. Fig. 1 shows the organization of the proposed multiplier-PLA loop. The multiplier's 16 bit input data (m[15:0]) and coefficient operands (y[15:0]) are held in 4×32 bit registers. The decoded addresses a[3:0] and a#[3:0] of the register file are provided by a single-cycle reconfigurable PLA. The PLA implements a 4-minterm function that determines the read/write access pattern of the register file. The four dual-rail outputs of the PLA are fed back into its inputs for next-cycle address calculation, thus implementing a programmable finite-state machine. The AND- and OR-planes of the PLA are programmed through the scan-chain by writing into a distributed configuration memory during startup. The 16×16 bit multiplier produces a 32-bit result, which loops back over an output bus and is written to the operand registers for future computation. This organization of the multiplier and PLA loop enables programmable single-cycle 16-bit multiply operations essential in high-performance/low-power embedded processor and DSP applications.

III. Reconfigurable PLA Control Block

The reconfigurable PLA (Fig. 2) computes a 4-minterm 4-operand logic function in a single cycle operation. The four dual-rail PLA outputs (a[3:0] and a#[3:0]) are sent to the register file, while simultaneously looping back into the PLA inputs for next cycle computation. Inputs to the PLA are multiplexed, choosing from the 4 external dual-rail inputs (d and d#) or the PLA outputs that are looping back. The PLA inputs directly connect to the AND plane circuits producing the 4 single rail minterms (and), which in turn connect to the OR plane circuits producing the PLA outputs. The reconfigurable AND plane and OR plane circuits use conventional domino logic, requiring a 50% duty-cycle 2-phase domino timing plan (Fig. 3). The 2Φ domino timing plan enables seamless time-borrowing between the AND and OR planes. The AND plane of the PLA is implemented using footed dynamic 8-wide NORs that produce four minterms during the first phase (Φ) of the clock cycle. In the next phase $(\Phi\#)$, 2-input footed dynamic NANDs in the OR plane select a combination of these minterms at each output.

The configuration bits of the PLA are stored in a distributed 48 bit memory that is sequentially written during initialization. 32 configuration memory cells are stored in the AND plane, while 16 are stored in the OR plane. These configuration memory cells enable reconfiguration of the PLA AND plane and OR plane circuits. The configuration bits are stored in the cross-coupled inverter memory cells and can be written into using a single write-port structure with the select write enable and write data. Programming is performed during initialization and completes in eight cycles to configure all 48 memory cells. Vertical rows of configuration memory bits are written in parallel performing a ganged write [9], thus reducing the programming time. In the AND plane, the configuration bits select each input (y or y#) of the footed dynamic 8-wide NOR gate. In the OR plane, the configuration bits have the capability to bypass each minterm input (and) of the 2-input footed dynamic NAND producing a final PLA output (or).

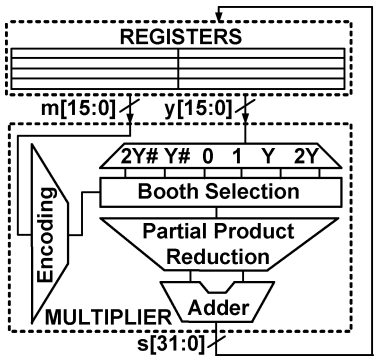


Fig. 4. 16 × 16 bit multiplier organization.

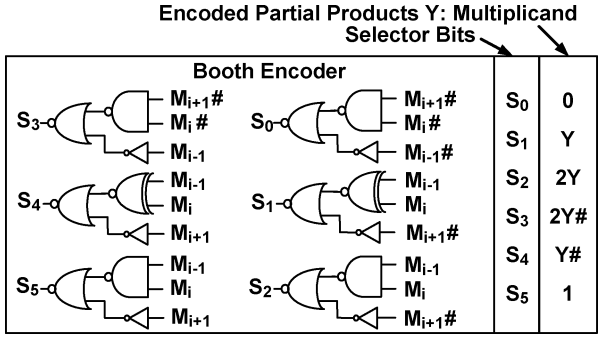


Fig. 5. Radix-4 Booth encoding.

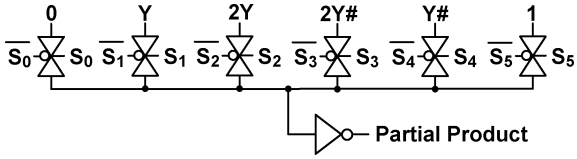


Fig. 6. One-hot 6-to-1 Booth encoding multiplexer.

IV. Energy-Efficient Multiplier Core

The 16 × 16 bit 2's complement multiplier (Fig. 4) is implemented using Booth encoding, reducing the total number of partial products. The operand registers provide two 16-bit inputs to the Booth multiplexer (y) and the Booth encoder (m). This Booth encoder allows the correct multiplicand terms to feed into the partial product reduction tree. Partial product tree outputs fed into the 32 bit final adder completing a full single cycle 16 × 16 bit multiplication operation. The 32-bit multiplier output (s) loops back and is written into the 32-bit operand register file as inputs for computation later. These operands are later read to become new 16-bit inputs for the multiplier's Booth encoding.

A. Booth Encoder

The first block of the multiplier performs radix-4 modified Booth encoding with sign-extension, generating eight 17-bit partial products. Modified Booth encoding [10] provides advantages over traditional Booth encoding [11] since it generates the hard 3Y multiple by using a negative partial product. An optimized Booth encoding scheme (see Fig. 5) needs to select the correct Booth encoded partial products. Sets of 3 adjacent multiplier inputs M are compared to produce 6 selector bit signals S0 to S5 which feed into a 6:1 multiplexer (see Fig. 6).

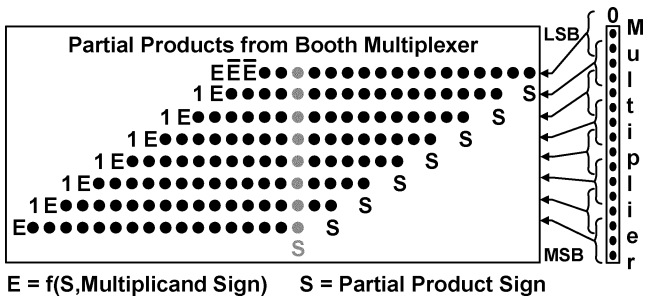


Fig. 7. Reduced sign extension.

TABLE I
One-Hot Booth Encoding Table

Partial Product Selection Table	
Multiplier Bits	Selection
000	+ 0
001	+ Multiplicand
010	+ Multiplicand
011	+ 2 x Multiplicand
100	- 2 x Multiplicand
101	- Multiplicand
110	- Multiplicand
111	- 0

- S = 0 if Partial Product is positive
- S = 1 if Partial Product is negative
- E = 1 if Multiplicand is positive and partial product is positive, or Multiplicand is negative and partial product is negative, or partial product is + 0
- E = 0 if Multiplicand is positive and partial product is negative or Multiplicand is negative and partial product is positive or partial product is - 0

This 6:1 transmission-gate multiplexer selects the correct Booth encoded partial product Y of the multiplicand. One hot Booth encoding reduces the contention in the Booth select multiplexer, lowering the delay and reducing the short circuit power. Conventional Booth encoding has an extra XOR gate in the critical path to create the complement of the Booth encoded partial products. This scheme removes the extra XOR, enabling a critical path delay of the multiplexer of only two gate stages. The outputs of this optimized Booth encoder connect to the inputs of the partial product tree.

B. Reduced Sign Extension

Compression of the sign-extension bits (see Fig. 7) is achieved by merging the signs of the partial products with the multiplicand and pre-computing their sum (Table I), thereby removing the sign-extension bits from the critical path of the compressor tree [12], [13]. A partial product tree is formed when the Booth encoding generates eight sign-extended 17-bit partial products. In a conventional design, the sign extension bits are fully extended to the most significant bit of each of the eight partial product rows. This sign extension results in 30% extra transistors across the boundary of the partial product tree and longer wire-loading on the Booth multiplexers. Compression of the sign-extension bits, represented by E, is achieved by first merging the signs of the partial products, S, with the multiplicand. Their sum is then pre-computed as shown in each partial product row. This removes the sign-extension bits from the critical path of the compressor tree. The reduced sign extension results in 23% reduction in partial product bits (from 208 to 160), and a subsequent 15% overall power reduction. The critical path through the partial product reduction tree involves the compression of 9 bits, implemented using seven (3:2) compressor circuits.

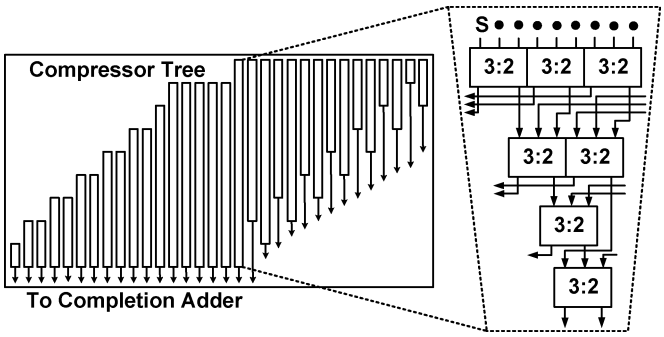


Fig. 8. Partial product compressor tree.

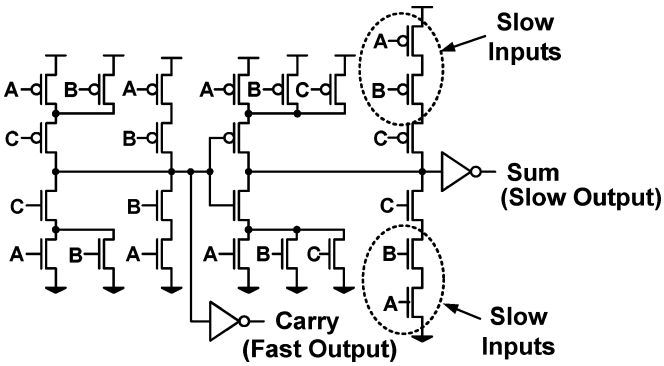


Fig. 9. Static mirror compressor.

C. Partial Product Reduction Tree

An optimally tiled partial-product reduction tree (see Fig. 8) compresses the Booth-encoded partial products using (3:2) compressors to produce 32-bit outputs in carry–save format. This optimal tiling is enabled by the inherent delay differences between the sum and carry outputs of the mirror adder circuit. The (3:2) static mirror compressor (see Fig. 9) used in the partial product reduction tree has a delay imbalance of 31% between *Sum* and *Carry* outputs. Compared to a high order compressor, a (3:2) compressor-based partial product tree provides the finest granularity to connect the fast outputs and slow inputs. Compressor layout is very dense since the PMOS and NMOS chains are completely symmetrical and require only 28 transistors. The critical path of seven compressors is optimally tiled in the compressor tree to exploit the delay difference between the fast and slow-arriving outputs. This optimization accounts for the vertical routing of the sum bits, as well as the horizontal routing of the carry bits, minimizing the total propagation delays. This organization also represents the layout of the compressors, which reduces wiring complexity and length at the expense of some layout area. Fast-arriving *Carry* signals are connected to slow upper-stack (A, B) inputs of the next compressor, resulting in 8% reduction in total compressor tree delay compared to the conventional Wallace-tree approach [6]. The absence of a full carry–propagation in the partial product tree produces a 32-bit output that remains in carry–save format. The compressor tree output arrival-profile shows a 4-compressor delay difference between the earliest and latest arriving completion adder inputs.

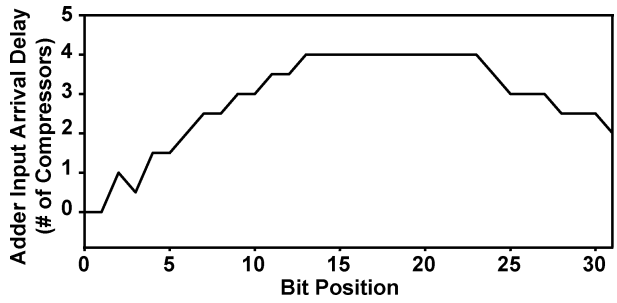


Fig. 10. Completion adder input profile.

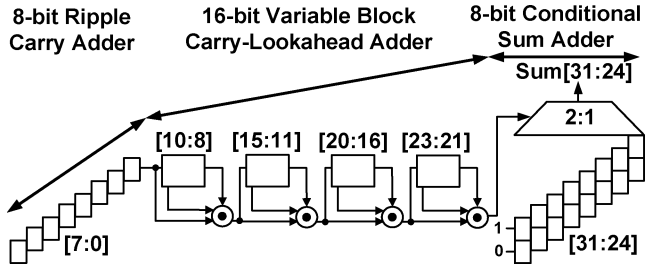


Fig. 11. 32-bit hybrid completion adder.

D. Hybrid Completion Adder

Fig. 10 shows the arrival delay variation for each input bit position of the 32-bit completion adder. Between the earliest and latest arriving completion adder inputs, the input arrival-profile shows a 4-compressor delay difference. The lower and upper order 8 bits arrive early, while the middle 16 bits are the most critical. An arrival-profile aware 32-bit completion adder converts the compressor tree outputs into a 2's complement final result. By taking advantage of the uneven arrival-time profile of the compressor tree outputs, the energy consumed is minimized by the completion adder.

To exploit this delay profile, a hybrid adder architecture (see Fig. 11) is used: ripple carry for bits $\langle 7 : 0 \rangle$, variable block carry-lookahead [14] for bits $\langle 23 : 8 \rangle$ and conditional sum ripple carry for bits $\langle 31 : 24 \rangle$. This results in a total critical path of 9 gate stages in the 16 bit variable block adder followed by one transmission gate multiplexer in the conditional sum adder. This hybrid architecture enables 20% power reduction with no performance penalty in the completion adder compared to a conventional high-performance carry-lookahead [15].

V. Write-Port Flip-Flops

Write-port flip-flops with NMOS-only clock transistors (Fig. 12) are used to reduce clock power throughout this chip, including the multiplier and reconfigurable PLA clock boundaries. This topology uses a conventional register file write-port for the master and slave stages, reducing the total clock load to only six transistors [16]. This topology results in 24% clock power reduction and 13% average flip-flop power reduction with no delay penalty compared to conventional pass-gate flip-flops. Strong cross-coupled keepers and dual-ended writes using a complementary 2-NMOS pull down stack ensure robust full-swing transitions on the storage nodes with good low

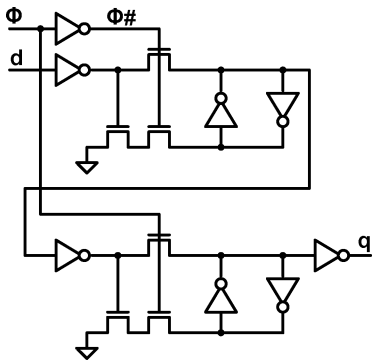


Fig. 12. Write-port master–slave flip-flop.

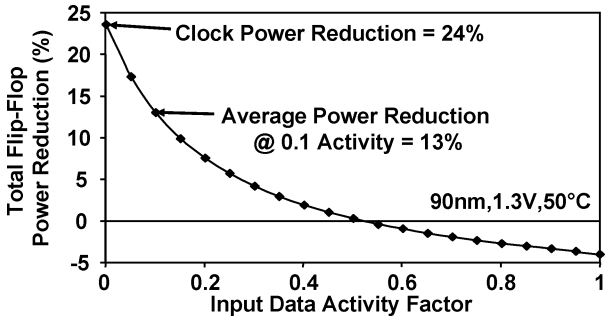


Fig. 13. Write-port power reduction.

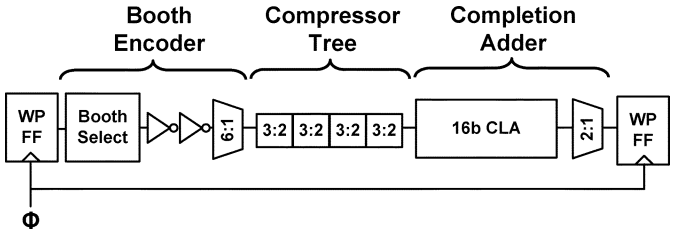


Fig. 14. Multiplier critical path.

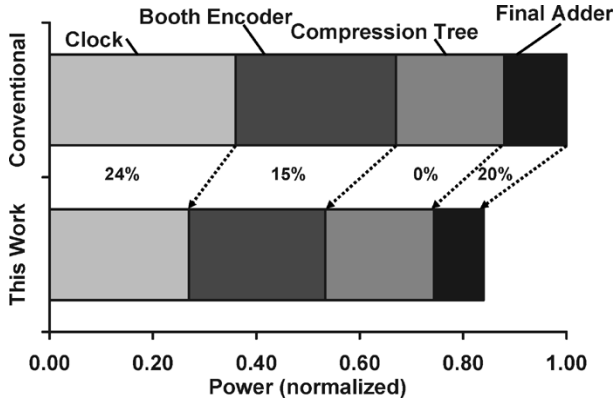


Fig. 15. Improvements over conventional.

voltage performance. Internal clock routing is also simplified since complementary wire routes for full transmission gates are not required, thereby reducing interconnect capacitance.

Write-port flip-flops trade off increased data power for reduced clock power. The increased data power is due to the increased contention within the uninterrupted cross-coupled storage node inverters. Therefore, as the input data switching activity increases, causing the storage nodes to transition, the data power component increases. This, in turn, diminishes the overall flip-flop power savings. Fig. 13 quantifies this trade-off for the write-port flip-flop compared to a conventional fully interrupted master–slave pass-gate flip-flop, both optimized for the same performance. The break-even input data activity factor is 0.55, beyond which the write-port flip-flop demonstrates higher power consumption (up to 4% higher) than the conventional pass-gate design. At all activity factors below 0.55, the write-port flip-flop's clock power dominates the data power, resulting in substantial total power reduction. During quiescent operation with no input data activity, the clock power reduction is 24% and at a representative activity factor of 0.1, the average flip-flop power reduction is 13%.

VI. Benefits Over Conventional Wallace Tree

Fig. 14 shows the total single-cycle critical path of 27 static gate stages through the multiplier, bounded by write-port flip-flops at the clock boundaries. The critical path distribution between Booth logic, compression tree and completion adder is 5, 12, and 10 gate stages, respectively. To reduce total active leakage and switching power, the multiplier and PLA loop uses only high-V_t transistors. Most transistors use minimum sizes with optimal device sizing performed on selective critical path gates. By optimizing the entire multiplier critical path, the overall power was reduced compared to a conventional design.

Achieving an overall energy-efficient multiplier design resulted in 3 key design choices: the one-hot 6:1 Booth encoding scheme, the hybrid adder scheme, and the write-port sequential flip-flops (see Fig. 15). The Booth encoding scheme enabled low contention current in the multiplexer and a short 2-stage multiplexer critical path delay resulting in a 15% power reduction. The hybrid adder scheme enabled to further reduce the power by 20% with no delay penalty by taking advantage of the input arrival profile. The write-port sequential flip-flops enabled further reduction in power by reducing the clock power. As a result of these design choices, a cumulative power reduction of 15% was achieved. The top two contributors were the clock power and Booth encoder power, contributing to 36% and 31% of the overall power, respectively. The partial product tree's power contribution was 21% while the final adder's power contribution was the lowest component at 12%.

VII. Measurement Results

Fig. 16 shows the microphotograph of the die implemented in 90-nm dual-V_t CMOS technology, with the reconfigurable PLA and 16-bit multiplier in the middle. The total die area is 0.474 mm^2, while the multiplier and PLA occupy an area of 0.03 mm^2 (see Table II). Fig. 17 shows the multiplier layout, which fits within a dense 215 μm $\times$ 130 μm template. The interconnect stack is comprised of 1 poly and 7 layer copper metal with low-k dielectric. The total pad count is 50, while the total number of transistors within the multiplier, PLA, register file, and test circuitry is 54 000.

Frequency and power measurements of the multiplier were obtained by sweeping the supply voltage from 0.57 to 1.95 V in a temperature-stabilized environment of 50 °C. Nominal

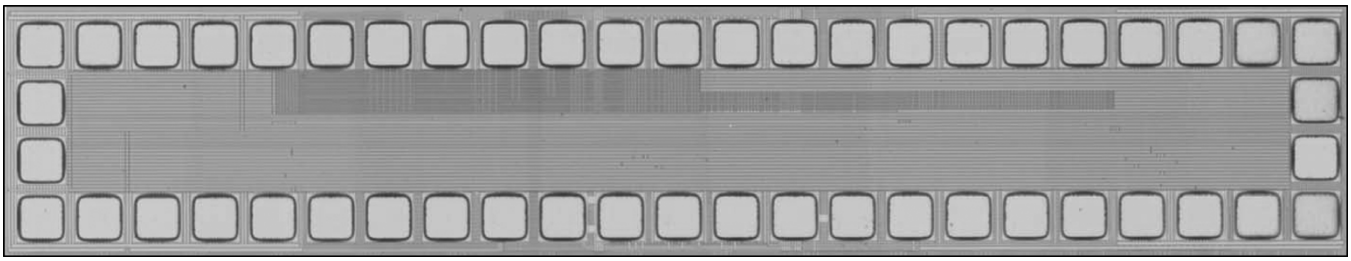


Fig. 16. Die microphotograph.

TABLE II
PROTOTYPE TABLE SUMMARY

Process	90nm CMOS
Nominal Supply	1.3V
Interconnect	1 poly, 7 metal
Die Area	0.474mm^2
Number of Transistors	54K
Pad Count	50

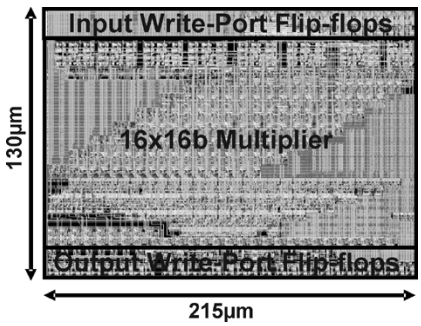


Fig. 17. 16 × 16 bit multiplier layout.

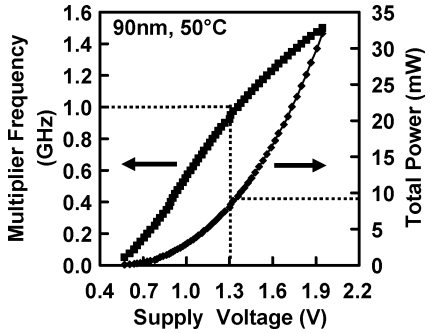


Fig. 18. Delay and power measurements.

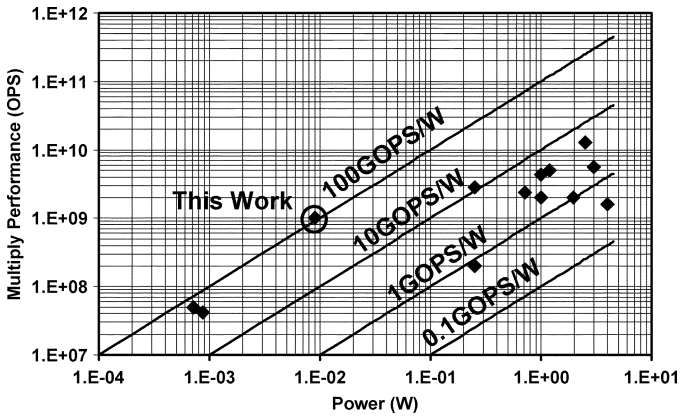


Fig. 19. Comparisons versus previously published designs.

supply voltage for this process is 1.3 V. The multiplier operates at a maximum frequency (F_{max}) of 1 GHz (measured at nominal 1.3 V, 50 °C), and consumes 9 mW total power, delivering 110 GOPS/W, where 1 operation is a complete single-cycle 16 × 16 bit multiply, including the 32 bit completion adder operation. The active leakage power component (540 μW) is 6% of total power. The reconfigurable PLA operates at a F_{max} of 2.3 GHz (measured at 1.3 V, 50 °C), and consumes 4.2 mW total power with an active leakage component of 100 μW. At 1 GHz, 1.3 V, 50 °C nominal loop operation, the PLA consumes 2 mW total power. Fig. 18 shows the multiplier F_{max} and total power measurements versus supply voltage. Multiplier performance is scalable up to 1.5 GHz consuming 32 mW (measured at 1.95 V, 50 °C). In low-voltage mode (measured at 570 mV, 50 °C), the multiplier operates at 50 MHz consuming 79 μW.

Fig. 19 shows comparisons of the 16 × 16 bit multiply performance and power consumption over 12 previously reported implementations. This proposed implementation achieves the highest reported measured power-performance operation at 110GOPS/W.

VIII. PMOS SLEEP TRANSISTOR OPERATION

The sleep transistors [17] are implemented using a PMOS switch (Fig. 20) which power gates the main supply with the virtual supply (Vs). The layout is surrounded by the sleep transistors which equate to a total transistor width of 200 μm. The sleep transistor is approximately 2.5% of the multiplier's active transistor width. Block activation and deactivation must be performed quickly to minimize the performance impact. Simulated transient power down time at 1.3 V, 50 °C, is 40 ns for the virtual supply to fully collapse to its natural state of less then 50 mV (Fig. 21). At a frequency of 1 GHz, this power down time equates to 40 cycles. Once in standby mode, the sleep transistor needs to wake-up quickly for any new operations that are triggered. Activation cycle time of the sleep transistor is also important because it determines the maximum leakage savings during standby mode. Simulated wake up time at 1.3 V, 50 °C, is 800 ps for the virtual supply to fully charge from the natural

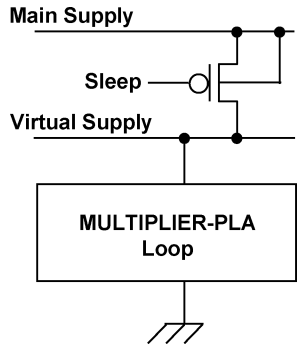


Fig. 20. Sleep transistor for standby.

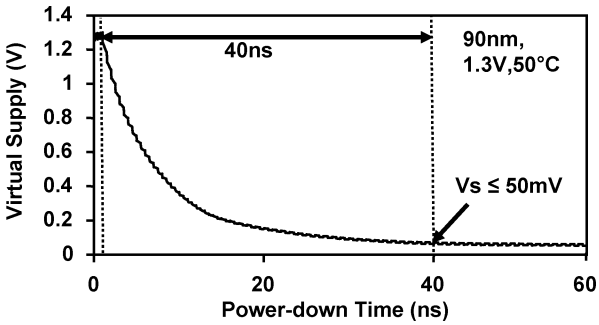


Fig. 21. Sleep transistor power-down time.

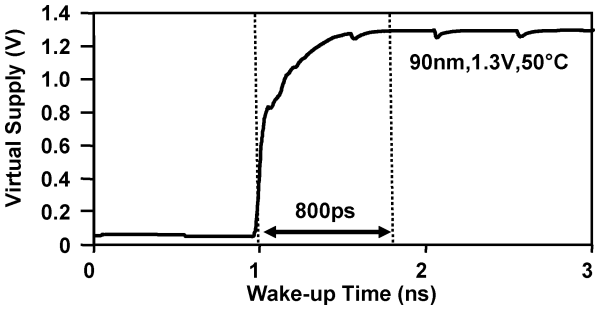


Fig. 22. Sleep transistor wake-up time.

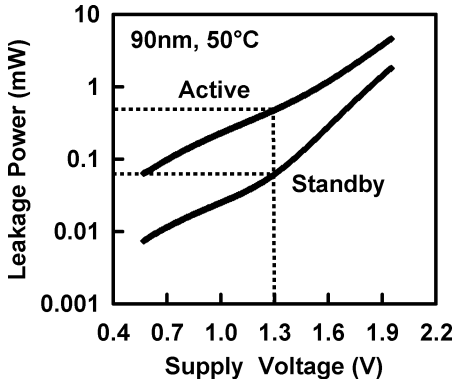


Fig. 23. Leakage measurements.

state to the nominal supply voltage of 1.3 V (Fig. 22). Note that this wake-up time achieved is less than 1 cycle (1 ns) recovery time to enable single cycle wake-up from standby.

Active and standby power measurements of the multiplier were obtained by sweeping the supply voltage from 0.57 V to 1.95 V in a temperature-stabilized environment of 50 °C (Fig. 23). At 1.3 V, the multiplier active leakage power is 540 μW, which is 6% of the overall power. When the PMOS sleep transistor is turned off, the multiplier standby leakage power is 75 μW, equating to a 7$\times$ leakage reduction compared to active mode.

IX. Summary and Conclusions

The design of an energy-efficient 16 $\times$ 16 bit 2's-complement multiplier and reconfigurable PLA control engine loop operating at 1 GHz in a 1.3 V, 90-nm CMOS technology, consuming 9 mW total power has been described. The use of an optimally tiled Booth-encoded compression tree, low clock power write-port flip-flop and the arrival-profile aware completion adder resulted in the most power-efficient multiplier (110GOPS/W) reported to date. The leakage component of total power was limited to 540 μW (6%) by maximizing the usage of high-V_t and minimum sized transistors, with selective upsizing on critical paths. Multiplier performance is scalable to 1.5 GHz, 32 mW, at 1.95 V. In the low-voltage mode of operation at 570 mV, the multiplier operates at 50 MHz and consumes 79 μW. Ultra-low standby power of 75 μW and $<$1 cycle wake-up time was achieved using PMOS sleep transistors, resulting in 7$\times$ reduction in measured leakage compared to active mode. This prototype addresses the challenges involved in designing energy-efficient hardware for power-constrained applications in high-performance process technologies, while consuming ultra-low energy.

Acknowledgment

The authors thank the Pyramid Probe Division of Cascade Microtech, Inc. for high bandwidth wafer level membrane probing solution; C. Webb, G. Gerosa, K. Soumyanath, F. Carroll, E. Tsui, L. Snyder for discussions; D. Trammo, C. Le for layout help; and M. Haycock, J. Schutz, J. Rattner, and S. Pawlowski for their encouragement and support.

References

[1] L. Clark *et al.*, "A scalable performance 32 bit microprocessor," in *IEEE ISSCC Dig. Tech. Papers*, Feb. 2001, pp. 230–231.
[2] C. Wallace, "A suggestion for a fast multiplier," *IEEE Trans. Electron. Comput.*, vol. EC-34, pp. 14–17, Feb. 1964.
[3] L. Dadda, "Some schemes for parallel multipliers," *Alta Freq.*, vol. 34, pp. 349–356, Mar. 1965.
[4] A. Weinberger, "A 4:2 carry-save adder module," *IBM Tech. Disclosure Bull.*, vol. 23, Jan. 1981.
[5] P. Song *et al.*, "Circuit and architecture trade-offs for high speed multiplication," *IEEE J. Solid-State Circuits*, vol. 26, no. 9, pp. 1184–1198, Sep. 1991.
[6] V. Oklobdzija *et al.*, "A method for speed optimized partial product reduction and generation of fast parallel multipliers using an algorithmic approach," *IEEE Trans. Comput.*, vol. 45, no. 3, pp. 294–306, Mar. 1996.
[7] S. K. Hsu *et al.*, "A 110GOPS/W 16 bit multiplier and reconfigurable PLA loop in 90 nm CMOS," in *IEEE ISSCC Dig. Tech. Papers*, Feb. 2005, pp. 376–377.

[8] K. Kuhn *et al.*, "A 90 nm communication technology featuring SiGe HBT transistors, RF CMOS, precision R-L-C RF elements and 1 μm^2 6-T SRAM cell," in *IEDM Tech. Dig.*, Dec. 2002, pp. 73–76.
[9] K. Mai *et al.*, "Architecture and circuits for a reconfigurable memory block," in *IEEE ISSCC Dig. Tech. Papers*, Feb. 2004, pp. 500–501.
[10] O. L. MacSorley, "High speed arithmetic in binary computers," *Proc. IRE*, vol. 49, no. 1, pp. 67–91, Jan. 1961.
[11] A. D. Booth, "A signed binary multiplication technique," *Quart. J. Mech. Appl. Math.*, pt. 2, vol. 4, pp. 236–240, 1951.
[12] M. Flynn *et al.*, "The SNAP project: toward sub-nanosecond arithmetic," in *Proc. IEEE Int. Symp. Computer Arithmetic*, Jul. 1995, pp. 75–82.
[13] G. Bewick, "Fast multiplication: Algorithms and implementation," Ph.D. dissertation, Stanford Univ., Stanford, CA, 1994.
[14] V. Oklobdzija *et al.*, "Some optimal schemes for ALU implantation in VLSI technology," in *Proc. 7th Symp. Computer Arithmetic*, Jun. 1985.
[15] B. Zeydel *et al.*, "A 90 nm 1 GHz 22 mW 16 × 16-bit 2's complement multiplier for wireless baseband," in *Proc. Symp. VLSI Circuits*, Jun. 2003, pp. 235–236.
[16] R. Krishnamurthy *et al.*, "Dual supply voltage clocking for 5 GHz 130 nm integer execution core," in *Symp. VLSI Circuits*, Jun. 2002, pp. 128–129.
[17] S. Mutoh *et al.*, "1-V power supply high-speed digital circuit technology with multi-threshold voltage CMOS," *IEEE J. Solid-State Circuits*, vol. 30, no. 8, pp. 847–854, Aug. 1995.

Steven K. Hsu (M'99) received the B.S. and M.S. degrees in electrical engineering in 1999 and 2001, respectively, from Oregon State University, Corvallis, and is currently working toward the Ph.D. degree at the same university.

He has been with Intel Corporation for over six years, and is currently a Senior Circuits Research Engineer in the High-Performance Circuits research group at Intel Corporation's Circuits Research Laboratories, Microprocessor Technology Laboratories, Hillsboro, OR. He has given tutorials on high performance CMOS circuits at the HPCA 2005, GLSVLSI 2004, ASIC/SoC 2004 circuit conferences. He serves as a SRC and Intel mentor on various university research projects. He has published 13 conference/journal papers and holds more than 10 U.S. patents.

Sanu K. Mathew (M'00) received the B.Tech. degree in electronics and communications engineering from the College of Engineering, Trivandrum, India, and the M.S. and Ph.D. degrees in electrical engineering from State University of New York at Buffalo in 1996 and 1999 respectively. His Ph.D. research focused on asynchronous circuit design.

He is currently a Senior Staff Research Engineer in the High-Performance Circuits research group at Intel Corporation's Circuits Research Laboratories, Microprocessor Technology Laboratories, Hillsboro, OR.

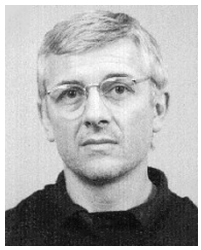
Dr. Mathew serves on the technical program committee of the IEEE International ASIC/SoC Conference and as SRC and Intel mentor on various university research projects.

Mark A. Anders (M'99) received the B.S. and M.S. degrees in electrical engineering from the University of Illinois at Urbana-Champaign, in 1998 and 1999, respectively.

Since graduation, he has been with Intel Corporation's Circuits Research Laboratory, Microprocessor Technology Laboratories, Hillsboro, OR, where he is currently an engineer in the High-Performance CIrcuits research group. His research interests are in high-speed and low-power data-path, DSP, and on-chip interconnects.

Bart R. Zeydel (S'00–M'01) was born in Orange, CA, on August 6, 1978. He received the B.S. degree in computer engineering from the University of California, Davis, in 2001. He is currently working toward the Ph.D. degree in electrical and computer engineering at the University of California, Davis.

In 2000, he worked at Mentor Graphics on the VRTX real-time operating system. In 2001, he worked at Fujitsu Microelectronics where he designed datapath elements for a VLIW processor and at Telairity Semiconductor, where he developed portable hard-IP datapath blocks. In 2003, he was an intern at Intel Corporation's Circuits Research Laboratories, Hillsboro, OR, where he designed datapath elements for DSPs. His research interests include high-performance and low-power datapath circuits, design methodologies for energy-efficient high-performance and low-power digital circuits, and the development of CAD tools for design in the energy-delay space.



Vojin G. Oklobdzija (S'78–M'82–SM'88–F'96) received the Dipl. Ing. degree from the Electrical Engineering Department of the University of Belgrade, Yugoslavia, in 1971, and the Ph.D. degree from the University of California at Los Angeles in 1982.

From 1982 to 1991 he was at the IBM Thomas J. Watson Research Center, where he made contributions to the development of RISC processors and supercomputer design. From 1988 to 1990 he was an IBM visiting faculty member at the University of California at Berkeley. Since 1991, he has been a professor at the University of California Davis where he directs the ACSEL laboratory, which is involved in digital circuits optimization for low-power and ultra low-power, high-performance system design and sensor nodes. He has served as a consultant to many companies, including Sun Microsystems, Bell Laboratories, Hitachi, Fujitsu, SONY, Intel, Samsung and Siemens Corporation, where he was a principal architect for the Infineon TriCore processor. He holds 14 U.S. and 7 international patents and has 5 other patents pending. He has published more than 140 papers, 3 books, and several book chapters in the areas of circuits and technology, computer arithmetic and computer architecture. He has given over 150 invited talks and short courses in the U.S., Europe, Latin America, Australia, China, and Japan.

Dr. Oklobdzija is an IEEE Fellow and Distinguished Lecturer of the IEEE Solid-State Circuits Society. He serves as associate editor for the IEEE TRANSACTIONS ON COMPUTERS, *IEEE Micro*, and *Journal of VLSI Signal Processing*. He served as Associate Editor of IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS from 1995 to 2003, the ISSCC digital program committee from 1996 to 2003, and numerous other conference committees. He was a General Chair of the 13th Symposium on Computer Arithmetic.

Ram K. Krishnamurthy (S'92–M'98–SM'04) received the B.E. degree in electrical engineering from Regional Engineering College, Trichy, India, in 1993, and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, in 1998. His Ph.D. research focused on low-power DSP circuit design.

Since graduation, he has been with Intel Corporation's Circuits Research Labs, Microprocessor Technology Laboratories, Hillsboro, OR, where he is currently a Principal Research Engineer and heads the High-Performance and Low-Voltage Circuits research group. He is an adjunct faculty of Department of Electrical and Computer Engineering, Oregon State University, where he teaches VLSI System Design. He holds 48 patents issued, 50 patents pending, and has published over 75 papers in refereed journals and conferences.

Dr. Krishnamurthy serves on the SRC ICSS Design Sciences Task Force and the program committees of the ISSCC, CICC, and SoC conferences. He is the Technical Program Chair/General Chair for the 2005/2006 IEEE International SoC Conference.

Shekhar Y. Borkar (M'97) was born in Mumbai, India. He received the B.S. and M.S. degrees in physics from the University of Bombay, Mumbai, India, in 1979, and the M.S. degree in electrical engineering from the University of Notre Dame, Notre Dame, IN, in 1981.

He joined Intel Corporation in 1981, and is currently an Intel Fellow and Director of Microprocessor Research at Intel Corporation, Hillsboro, OR. He worked on the design of the 8051 family of microcontrollers, iWarp multi-computer, and high-speed signaling technology for Intel supercomputers. He is an adjunct member of the faculty of the Oregon Graduate Institute. He has published more than 10 articles and holds 11 patents.