

A 2GHz 13.6mW 12x9b Multiplier for Energy Efficient FFT Accelerators

Steven Hsu, Vishak Venkatraman*, Sanu Mathew, Himanshu Kaul, Mark Anders, Saurabh Dighe, Wayne Burleson*, Ram Krishnamurthy
 Electrical and Computer Engineering, University of Massachusetts, Amherst, MA 01002, USA
 Circuit Research, Intel Labs, Intel Corporation, Hillsboro, OR 97124, USA
 vvenkatr@ecs.umass.edu, ram.krishnamurthy@intel.com

Abstract:

Two's complement multipliers are performance and power-critical components for wireless baseband signal processing applications. Parallel clusters of multiplier, multiply-add, multiply-accumulate cores are required to perform complex filter operations in Fast Fourier Transform (FFT) accelerators while consuming ultra low energy/operation [1]. A 12x9b single-cycle two's complement twiddle multiplier for FFT acceleration implemented in 90nm dual-V_t CMOS technology [2], operating at 2GHz and consuming 13.6mW at 1.3V, 110°C is presented. 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 this aggressive power-performance by achieving (i) low compressor tree fan-outs and wiring complexity, (ii) low active leakage power of 1.3mW and high noise tolerance with all high-V_t usage, (iii) scalable multiplier performance up to 2.5GHz, 33mW at 1.7V, 110°C, and (iv) low-voltage mode multiplier performance of 35MHz, 50μW at a supply of 300mV, 110°C.

1. Introduction:

With the rapid development of handheld electronics, the need for high performance and low power signal processing systems with very high data throughput has grown in recent years. Reduced bit-width (≤16b) two's complement multipliers are essential ingredients of signal processing systems in high performance embedded processor and digital signal processing (DSP) cores. To obtain the necessary throughput and a high power-performance ratio, efficient multiplier design is necessary for signal processing applications.

Finite impulse response (FIR) filtering, a key operation in DSP applications, uses transformation techniques like the Fast Fourier Transform (FFT) to efficiently compute the Discrete Fourier Transform (DFT). A typical FFT operation involves scaling (i.e. multiplying) the input data stream $x(n)$ with an array of coefficients and accumulation of the product over several cycles. For example, a common multiplication routine in DFT involves the computation of the sequence $X(k)$ of N complex-valued numbers using the equation given below, where $x(n)$ is a N samples time-domain sampled signal and $X(k)$ is its DFT in the frequency-domain.

Since the value and periodicity of the twiddle factors W_N are fixed for a given DSP algorithm, these coefficients can be pre-computed and 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.

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{kn}, \quad 0 \leq k \leq N-1$$

$$W_N = e^{-j2\pi / N}$$

Our proposed multiplier, which is an integral part of this FFT engine, is used to compute a 128-point FFT. The 128-point FFT is divided as 4x4x4x2 stages. This implies that there are 3 consequent butterfly radix-4 stages followed by a butterfly radix-2 stage. In each butterfly radix-4 blocks, there are four inputs and four outputs and the four inputs are complex numbers. This implies that each butterfly radix-4 requires 8 complex adders and 3 complex multipliers. For each of the multiplier, one of the inputs is twiddle factor, which is a constant number, saved in ROM, and the other input is the previous stage output.

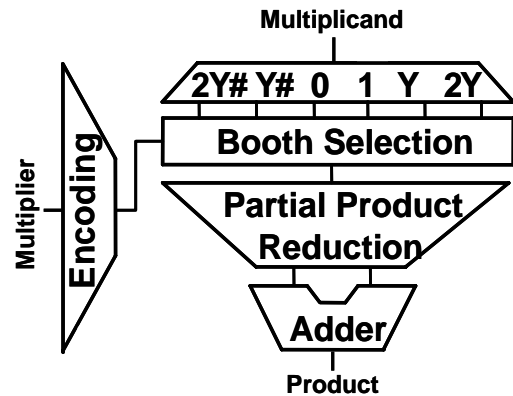


Figure 1: Multiplier Architecture

2. Multiplier architecture:

Fig. 1 shows the architecture of the proposed two's complement FFT multiplier, which is partitioned into three stages. The first stage generates the select bits used later for generating the partial products. The encoding is performed using modified-Booth algorithm with sign-

extension. In the second stage, a partial product reduction tree compresses the Booth-encoded partial products via 3:2 compressors to produce output in carry save format. The third stage adds the two partial product reduction tree outputs via a fast 21b carry propagate adder to produce the final product. Write-port input-output flip-flops latch the data to and from the multiplier respectively.

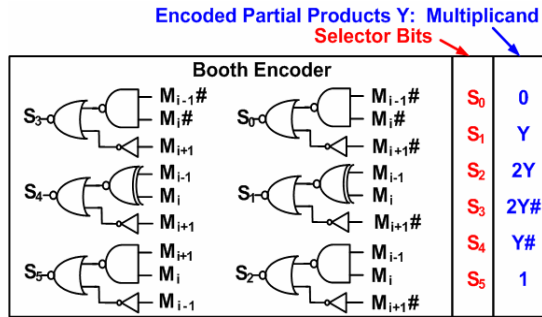


Figure 2: Booth Encoder

3. Booth encoding and partial product selection:

The first block of the multiplier, shown in Fig. 1, performs radix-4 Booth encoding with sign-extension, generating 5 14b partial products. An optimized one-hot Booth encoding scheme shown in Fig. 2 is used to encode the multiplier. The select bits ($S_0 - S_5$) select the sign-extended partial products, enabling a critical path delay of only 2 gate stages. Compression of the sign-extension bits is achieved by merging the signs of the partial products with the multiplicand and pre-computing their sum, thereby removing the sign-extension bits from the critical path of the compressor tree [3]. This results in 23% reduction in partial product bits and subsequent 15% reduction in overall power.

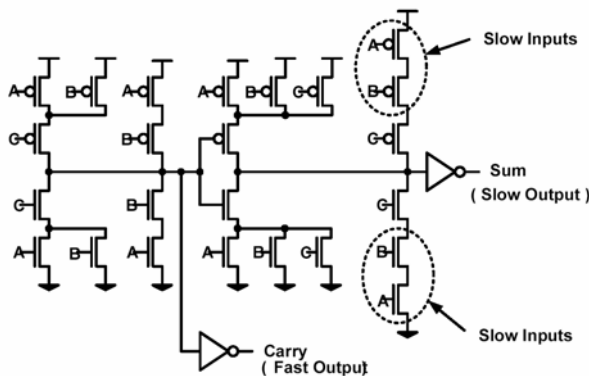


Figure 3: 3:2 Static Mirror Compressor

4. Partial Product Compression:

An optimally tiled partial product reduction tree compresses the Booth-encoded partial products using 3:2

compressors to produce 21b outputs in carry-save format. A 3:2 static mirror compressor shown in Fig. 3, is used in the partial product reduction tree. The 3:2 compressor has a delay imbalance of 31% between Sum and Carry outputs. This delay difference is exploited to optimally tile the compressors, minimizing total horizontal and vertical tree propagation delays. Fast-arriving Carry signals are connected to slow upper-stack inputs of the next compressor, resulting in 8% reduction in total compressor tree delay compared to the conventional Wallace-tree approach [4].

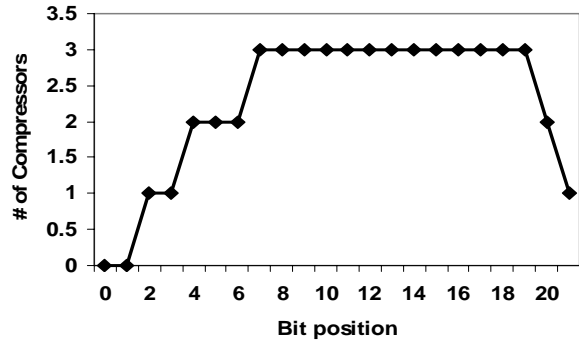


Figure 4: Arrival profile of adder inputs

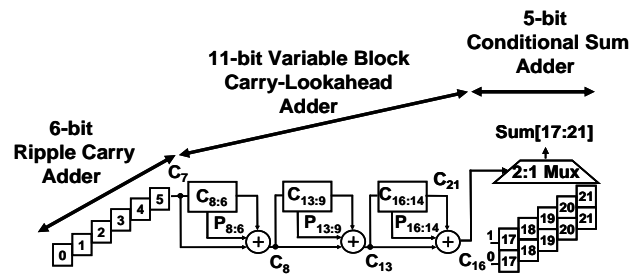


Figure 5: Arrival profile aware final adder

5. Vector Merging:

An arrival-profile aware 21b completion adder converts the compressor tree outputs into two's complement result. This adder takes advantage of the uneven arrival-time profile of the compressor tree outputs to minimize energy consumed by the final addition. The compressor tree output arrival-profile shown in Fig. 4 shows a 3-compressor delay difference between earliest and latest arriving completion adder inputs. To exploit this delay profile, hybrid adder architecture, shown in Fig. 5, is used. The hybrid adder consists of a ripple carry adder for bits $\langle 5:0 \rangle$, a variable block carry-lookahead for bits $\langle 16:6 \rangle$ and conditional sum ripple carry for bits $\langle 21:17 \rangle$. This results in a total critical path of 6 gate stages in the 11b 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 implementation [5].

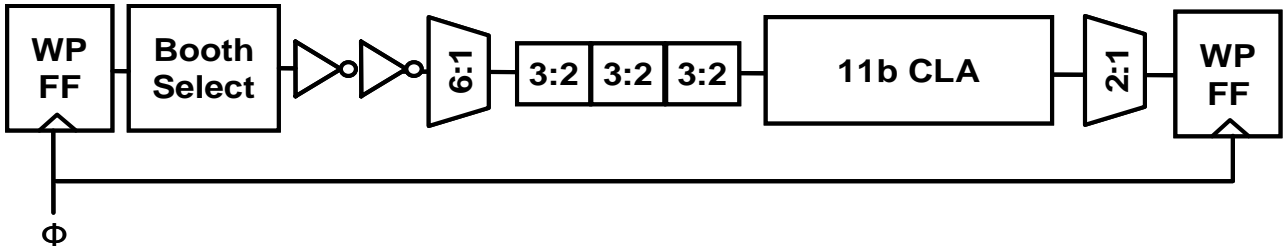


Figure 6: Critical path of the multiplier

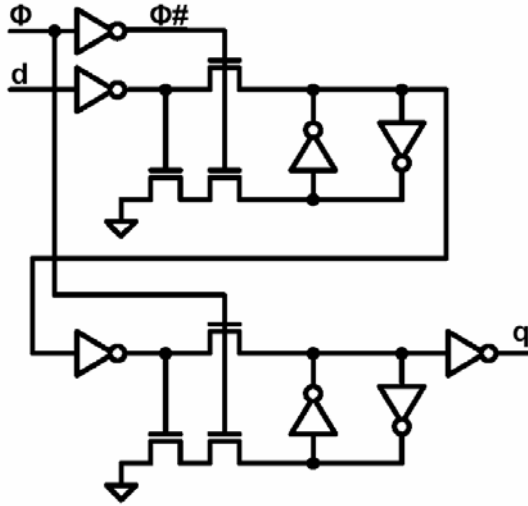


Figure 7: Write Port Master-Slave Flip-Flop

6. Input-output Flip-flops:

Write-port flip-flops with NMOS-only clock transistors are used to reduce clock power within the multiplier. This topology, shown in Fig. 7, uses a conventional register file write-port for the master and slave stages, reducing the total clock load to only 6 transistors. This results in 24% clock power reduction and 13% average flip-flop power reduction with no performance 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-voltage performance and tolerance to P/N skew variations [6].

7. Critical Path Analysis:

Fig. 6 shows the critical path of the proposed multiplier. The critical path consists of three main stages. The first stage is the Booth select mux followed by two inverter stages and a 6:1 mux, which comes to a total of 5 gate stages. The second stage comprises of the three 3:2 compressors totaling to 9 gate stages. The third stage comprises of a 11b Carry Lookahead adder, which comprises of 6 gate stages. The ripple carry adder does not contribute to the gate stages due to the arrival profile of the inputs to the final adder. The mux at the conditional sum stage contributes to one more gate stage, which comes to a total to 21 gate stages in this critical

path. This path can be excited with the following input patterns. When the multiplicand is at 0xFFFF and the multiplier goes to 0x001 from 0x000, the output of the multiplier transitions to 0xFFFF and the carry ripples through all the 21 gate stages.

8. Comparison:

Fig. 8 shows energy breakup comparisons of the proposed multiplier vs. a conventional custom Wallace-tree based multiplier implementation in 90nm CMOS technology [2]. Both multipliers are optimized for the same target performance of 2GHz single-cycle throughput. The conventional multiplier consumes 36%, 31%, 21% and 12% within the clock network (including flip-flops), Booth encoder, compression tree and completion adder. The proposed multiplier demonstrates a 24% reduction in clock power due to write-port flip-flops and 15% power reduction within the Booth encoder. The hybrid completion adder enabled an additional 20% power benefit, resulting in a cumulative power improvement of 15% while retaining the same performance. Further, the proposed multiplier is compared to a conventional RTL-synthesized 12x9b multiplier optimized for the same 2GHz single-cycle throughput target. The conventional multiplier was synthesized using commercial Synopsys® design compiler blocks [7] in 90nm CMOS technology [2]. The synthesized multiplier employs a non-Booth Wallace-tree architecture. Table 1 summarizes the comparison results at nominal operating supply voltage of 1.3V, 110°C. Compared to the synthesized implementation, the proposed multiplier transistor count was 19.9% lower. Worst-case power dissipation of the proposed multiplier was 52.5% lower than that of the synthesized multiplier.

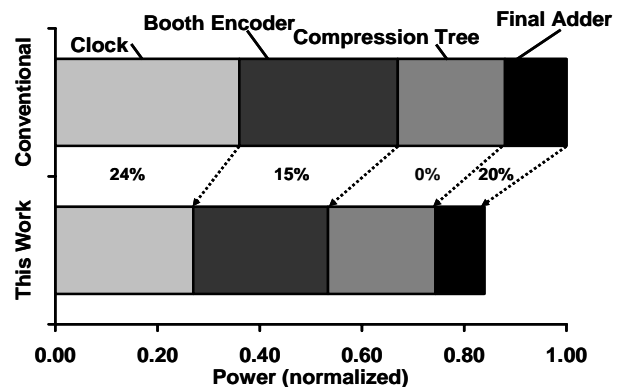


Figure 8: Energy breakup comparison

Supply (V)	Proposed		Synthesized	
	Power (mW)	Transistors (#)	Power (mW)	Transistors (#)
1.3	13.59	4472	28.61	5583

Table 1: 90nm power and transistor count comparisons of proposed vs. conventional synthesized implementation

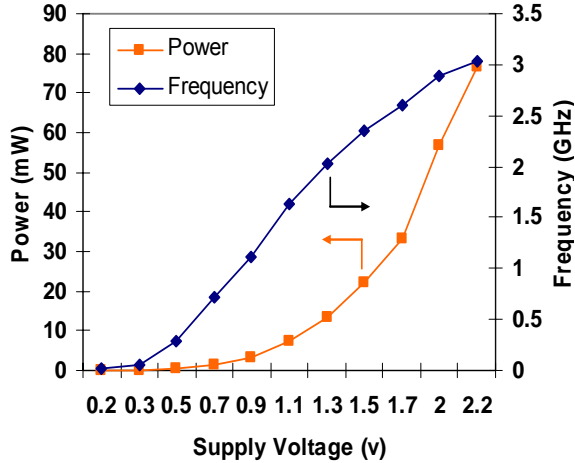


Figure 9: Power-performance vs. supply voltage in 90nm

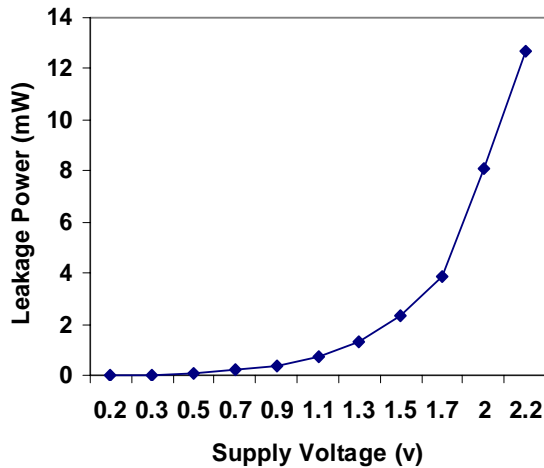


Figure 10: Leakage scaling with supply voltage in 90nm

9. Results:

Fig. 9 shows worst-case power-performance trade-off of the proposed multiplier. This reflects the total single-cycle critical path of 21 static gate stages through the multiplier, bounded by write-port flip-flops at the clock boundaries. To reduce total active leakage and switching power, the multiplier uses only high- V_t transistors. Most transistors use minimum sizes with optimal device sizing performed on selective critical path gates. The multiplier operates at a maximum frequency of 2GHz (at nominal 1.3V, 110°C), and consumes 13.6mW total power or 200GOPS/W, where one operation is a complete single-cycle 12x9b multiply. The active leakage power component of 1.3mW is 11% of total power, is shown in Fig. 10. The leakage power increases (decreases)

exponentially as supply voltage is increased (lowered). Multiplier performance is scalable up to 2.5GHz consuming 33mW (1.7V, 110°C). In low-voltage mode (300mV, 110°C), the multiplier operates at 35MHz consuming 50 μ W.

10. Conclusions:

We have presented a 12x9b two's complement multiplier for accelerating FFT operations in wireless baseband applications. This multiplier is an integral part of the 128-point FFT engine and was custom-designed in 90nm CMOS technology. It operates at a maximum frequency of 2GHz at 1.3V consuming 13.6mW of power and is scalable to 2.5GHz at 1.7V, all at 110°C. Further, the multiplier can be scaled to operate at 35MHz consuming 50 μ W of power at 300mV supply and 110°C. The proposed multiplier was compared with a speed-optimized synthesized 12x9b multiplier for area and power. The proposed multiplier achieved a power savings of 52.5% with 19.9% reduction in area when compared with the synthesized multiplier and 15% power reduction when compared with a custom multiplier design, all optimized for the same performance target.

11. Acknowledgments:

The authors thank V. Oklobdzija, B. Zeydel, M. Kazemi-Nia, 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, E. Hoffman, M. Schaecher, M. Biyani, D. Roberts, Yuyun Liao, A scalable performance 32b microprocessor, IEEE Intl. Solid-State Circuits Conference, Digest of Technical Papers, pp. 230-231, Feb. 2001.
- [2] K. Kuhn et al., A 90nm communication technology featuring SiGe HBT transistors, RF CMOS, precision R-L-C RF elements and 1 μ m² 6-T SRAM cell, IEDM Technical Digest, pp. 73-76, Dec. 2002.
- [3] S.F. Oberman, H. Al-Twaijry, M.J. Flynn, The SNAP project: towards sub-nanosecond arithmetic, Computer Arithmetic Symposium Proceedings, pp. 75-82, Jul. 1995.
- [4] V.G. Oklobdzija, D. Villeger, S.S. Liu, A method for speed optimized partial product reduction and generation of fast parallel multipliers using an algorithmic approach, IEEE Transactions on Computers, pp. 294-306, Mar. 1996.
- [5] B.R. Zeydel, V. Oklobdzija, S. Mathew, R. Krishnamurthy, S. Borkar, S. A 90nm 1GHz 22mW 16x16b 2's complement multiplier for wireless baseband, VLSI Circuits Symposium Digest of Technical Papers, pp. 235-236, Jun. 2003.
- [6] S.K. Hsu, S. Mathew, M.A. Anders, B. Bloechel, R. Krishnamurthy, S. Borkar, A 110GOPS/W 16b multiplier and reconfigurable PLA loop in 90nm CMOS, IEEE International Solid-State Circuits Conference, Digest of Technical Papers, pp. 376-377, Feb. 2005.
- [7] Synopsys® Design Compiler. Synthesis and Power Estimation Toolset, www.synopsys.com