

TECHNIQUES FOR IMPROVING LOW-POWER, HIGH-PERFORMANCE BOOTH MULTIPLIERS' PERFORMANCE

^{#1}Mr. P. MADHUSHEKHAR, Assistant Professor,
 ^{#2}Dr. V. SRIDHAR, Associate Professor,
 ^{#3}Ms. M. MAMATHA, Assistant Professor,
 ^{#1,2,3}Department of Electronics and Communication Engineering,

TRINITY COLLEGE OF ENGINEERING AND TECHNOLOGY, PEDDAPALLY.

ABSTRACT: Multipliers are crucial in many digital technologies. Highly integrated SoC cores and GPU-based processors use them. The previous few decades have been spent improving these system components since they are so crucial. Radix-4 modified Booth encoding (MBE) has reduced delay and silicon area, hence most high-performance multipliers employ it. Booth encoding reduces the amount of partial products needed by twice as much as non-Booth variations. Integer addition and multiplication affect digital media and signal processing applications. Many approaches based on radix-4 Booth recoding have been proposed. This basic method reduces a multiplier's peak value by 50% and only requires simple multiplication operations. It is commonly used in multiplier design. VLSI systems are easily reconfigurable, therefore the technique uses that. The proposed structure utilizes little power, as measured by power efficiency, area consumption, and logic usage. A proposed system in this study uses a low-power Radix-4 booth multiplier. A changeable path selection approach allows the repeater save the carry path fast. The last sum component adds both lines' findings.

Keywords: Multipliers, VLSI, Radix, modified, diminish, widespread.

1.INTRODUCTION

Digital hardware needs multipliers, found in SoC processors and GPU accelerators. Since they often affect system performance, their performance has been prioritized for decades. Since batterypowered ubiquitous devices are so common, lowoperation design objective. power is а Performance remains vital. Due to their complicated combinatorial modules and imbalanced reconvergent paths, most proposed multipliers high-performance have higher capacitive loads and spurious activities, which could make them the main power dissipator.

Due of its small silicon size and low latency, highperformance multipliers use Radix-4 modified Booth encoding (MBE). The number of incomplete products needed to be added is around twofold lower with Booth encoding than without. MBE speeds partial product addition with Wallace, optimized Wallace-tree (OWT), Dadda, Braun's, and three-dimensional minimization (TDM) adder-tree-reduction methods. Example: Carry-save propagation and OWT scheme

known for reducing logarithmic delays in complete adder or 4-to-2 compressor adder trees [18–20]. A common adder tree implementation uses the latter.Despite its speedier performance, the MBE's energy efficiency is questioned because to its complex encoding-decoding circuitry and higher spurious activity. The input operands' reduced dynamic range and 2's complement notation make this more clear. Results include Baugh-Wooley, sign magnitude (SM), and gray coding (GC) multiplier systems.

260

The Baugh-Wooley approach, which uses a 2input AND array, was 25% more power efficient and somewhat higher delay than the Booth method for partial product generation (PPG). SM and GC use format conversion logic at both multiplier extremities but reduce signal changes with number representation. SM solutions reduce switching activity by 90% and 50% compared to MBE, while GC implementations save 45% power. These approaches fail for applications where input operands change fast across the word length. In some systems, critical path conversion circuits slow slower and consume more power under severe time constraints. The Booth multiplier has been optimized structurally and gate-level in the literature.

A more regular array of partial products was suggested to reduce carry summation adder rows. The solution improves performance by 25% over standard implementations. Kang and Gaudiot's fast 2's complement generating circuit omits carry-in terms to restructure the partial product array. The research suggests a hardware method that requires fewer resources to get the same result. These solutions reduce power consumption by 15–33% and boost performance by 5–9.1% for 8-bit versions. The improved circuits have more balanced data pathways and a more efficient partial-product array structure than higher-level implementations. Leap-frog (LFR) and left-toright structures were offered as alternatives to OWT to reduce sum-carry imbalance, however their area and delay overhead are not negligible.

However, area and speed are performance restrictions that conflict. Increasing innovation pace always increases area. The proposed architecture speeds up the well-known Wallace tree multiplier on an FPGA. The standard Wallace multiplier is structurally optimized to reduce circuit delay. If the partial products in the n-k least significant columns are all ones or zeros, a truncated multiplier with constant correction has the maximum error. The variable corrective truncated multiplier is proposed. It adjusts the correction term for column n-k-1. If all partial products in column n-k-1 are 1, the correction term rises. If all partial products in this column are zero, the correction term decreases. A simplified

JNAO Vol. 14, Issue. 2, : 2023

22 multiplier block may be used to generate larger arrays. To accelerate the partial product reduction tree and reduce power dissipation, compressors commonly utilized in fast multiplier are architecture. Kelly and Ma et al. also studied compression for approximation multiplication. An approximation signed multiplier for arithmetic data value speculation (AVDS) uses the Baugh Wooley algorithm for multiplication. The computation is inaccurate, hence no new compressor design is advised.

2.LITERATURE SURVEY

A signed binary multiplication approach. Booth, David.

Computers can use application-specific, highspeed processors thanks to digital arithmetic. Recently developed digital circuits have excellent clock rate, input/output latency, small silicon area, and low power dissipation. This work implements numerous sinusoidal generation methods using cutting-edge digital arithmetic to optimize output and performance. Advanced digital oscillator structures with and without pipelining are by study. Pipelining recommended this outperforms other sinusoidal generating methods in maximum frequency and signal resolution. Thus, the proposed digital oscillator chip is developed this way.

A parallel approach for two's complement array multiplication. After B. and R. A. Wooley.

A fast m-by-n-bit, two's complement, parallel array multiplication algorithm. All partial product bits are positive ANDs of multiplier and multiplicand bits.Z designed a high-performance, low-power left-to-right array multiplier. Both Chen and M. Ercegovac, D.

We describe a high-performance, low-power linear array multiplier architecture using left-toright leapfrog (LRLF) signal flow, separating the reduction array into upper and lower sections, and optimizing signal flow in [3:2] adder array for partial product reduction. Using produced upper/lower LRLF (ULLRLF) multipliers to compare tree multipliers. In automatic layout tests, ULLRLF multipliers for n/spl les/32 have 261

equivalent power, latency, and area to tree multipliers. The more regular and shorter interconnects of the ULLRLF structure make it a viable option to tree topologies for designing fast, low-power multipliers for deep submicron VLSI technology.

3.EXISTING SYSTEM

The matrix geometry below is the basis for the WALLACE multiplier algorithm. In the first step, AND stages form the partial product matrix.





Fig.1- Steps in the 4x4 WALLACE Algorithm for WALLACE TREE Multipliers:

Multiplying (or ANDing) each argument bit by the other yields N outcomes. Conductor weights depend on multiplied bit locations.

Building two layers of complete adders from incomplete output. Sort the wires into two numbers using a standard adder.



Fig 2 Several AND gates generate product names.

Ripple carry adders concatenate numerous additions with carry in sand carry outs. Thus, the ripple carry adder uses many adders. A logical circuit with several adders may add multi-bit numbers. Each full adder inputs its predecessor's Cout, Cin. This is a ripple carry adder because each carry bit "ripples" to the next complete adder. Figures 9-11 show the Wallace Multiplier Algorithm design using RCA. A full adder accepts three identical-weighted values. A wire with the weight input will same as the result.

JNAO Vol. 14, Issue. 2, : 2023

Multiplication yields an initial product fraction. In each stage, the carry is multiplied by the following two data from three conductors added with adders. The similar strategy reduces partial products to two full adder layers. The ripple carry adder method is used to acquire product terms p1–p8 in the final stage.

4.PROPOSED SYSTEM

The multiplier is calculated using Radix-4 Booth. The product p has a 2n two's complement value when x and y are the n-bit multiplicand and multiplier, respectively. N-segment multiplication analyzes several Y digits.

$$N = \left\lfloor \frac{n+2}{2} \right\rfloor.$$

Equation describes computation.

$$p = (Y_1 + Y_0)x + \sum_{i=1}^{N} 2^{2i-1} (Y_{2i+1} + Y_{2i} - 2Y_{2i-1})x.$$
2

1

Y denotes the multiplier y's length-N-digit vector. Three multiplier Y digits are used to calculate the radix-4 Booth encoding e.

			3
Y_{i+2}	Y_{i+1}	Y_i	e,
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	2
1	0	0	$\overline{2}$
1	0	1	ī
1	1	0	ī
1	1	1	0

Table I Booth Corner

I is the Ith digit. Table I compares Yi+2Yi+1Yi = 0 and 1.

111 causes a 0, and the multiplicand is scaled by 1, 2, 2, or 1 depending on encoding.

Determine the fragmented product i of an incomplete item to use ei.

Partial Product_i =
$$e_i x = (Y_{2i+1} + Y_{2i} - 2Y_{2i-1})x$$
.

This partial product is adjusted by the left shift (22i1), and the sum determines the final product (p). No digit exists for Y1, hence PartialProduct0 = (Y1, the 0th intermediate item).

+Y0)x. Enhancement is sequential by calculating

262 each incomplete item in N cycles

$$p[0] = 2^{n-2}(Y_1 + Y_0)x$$

$$p[j+1] = 2^{-2}(p[j] + 2^n(Y_{2j+1} + Y_{2j} - 2Y_{2j-1})x),$$

$$j = 1, ..., N - 1 - 5$$

Two changes boost equipment use. First, the multiplier y is assigned to the item p (p = y), using the n least significant bits of the p register instead of storing y in a separate register. As the item is shifted right, its three LSBs form the following encoding ei (p = sra(p, 2)). The fractional item's left realignment shift (2n) is eliminated by combining the partial product with its n normally huge components (P[2_ B1: B]+ = Partial Product).



Fig. 3. A little TSM. An additional control circuit allows bypassing, and it uses two delay pathways. Previous research has shown that Booth encoding and decoding can be optimized with the right intermediary signals. Fig. 2(a) - 2(d)show literature-described MBE circuit implementations. This analysis only addressed full-swing circuit topologies. Fig. 2(a) (BED13) shows a hybrid encoder-decoder circuit with 36 and 10 transistors [46]. Decoder block has the fewest transistors among non-CMOS implementations. There are some difficulties with this implementation. The unbuffered selector circuit (SEL) of four pass transistors generates resistive circuits cascading from the decoder inputs to the outputs, as shown in Fig. 3(a). Due to an imbalance in driving loads supplied to SEL blocks for different input

JNAO Vol. 14, Issue. 2, : 2023

configurations, arrival times vary. Second, decoding block routing congestion in Fig. Due to 2(a)'s substantial growth, PPG parasitic interconnects increase.



Fig. 4. Multiple Booth encoder/decoder implementations. (a) BED13. (b) BED20. (c) BED22.

Poor Booth circuitry. e) This work's 6T-XOR/XNR circuits (WM1M8 = 0.15). BED18 encoder-decoder circuits recommended in (f). (g) Decoder AO22 (J3) (WM1M4 = 0.16, WM5M8 = 0.15).

Like the circuits in Fig. The utilization of transmission gate pairs for encoders in 2(b) (BED20) enables PPG to operate faster. However, the unbuffered encoder outputs become susceptible to the hazards introduced by the circuit itself. Additional wiring and an increase in capacitive capacitance at the decoder also contribute to a higher power consumption in PPG. The arrangement shown in Fig. 2(c) (BED22) is the optimal variant in terms of transistor count and signal synchronization. The decoders share the XOR operations that result in ny j - ny j, and the AOI22 cell provides encoder signals with balanced loads. Since it was chosen, the reduced multiplication of [41] was also favored. Due to functional defects when all encoder inputs (b2i 1b2i) are present, the evaluation does not take into account the novel Booth circuits provided.

They are at the logic "1" (+1) level.

Fig. depicts the proposed MBE circuits in this study. 2(e)-(g). Fig. demonstrates the essential

263

leaf cell of the proposed circuitry. 2(e). This version of the XOR/XNR architecture has lower capacitances than previous full-swing gate implementations. Despite this benefit, it is hindered by signal route delay asymmetry. If, for example, the circuit depicted in Fig. Due to the inertial and propagation delays of the inverter, when both inputs transition from 0 to 1 in 2(e), M1 of the XOR temporarily controls the output, causing a malfunction at the XOR output. By virtue of the inversely proportional relationship between inertial and propagation delays, the freedom of device size is constrained. Therefore, these XOR/XNR outputs were directly if interfaced to high fan-out nets, the erroneous activities in PPG could only become worse.



Fig. 5. Many low-power, full-swing adders. (a) RFL22. (b) TFA22. (c) BFA22. (d) HFA26. (e) the CMOS28. Proposed (PBFA26) can be found in (f).

Full adders are the fundamental construction blocks of the multiplier adder tree. Fig. demonstrates the most common static rail-to-rail adder solutions. 3(a)- (e). The buffered variants of the original implementation are considered for a fair comparison. The blue arrow line indicates the critical path of each complete adder. Fig. 3(a)–(c) requires a minimum of 22 transistors (plus the inverters for the undrawn input signals). The numbers in Fig. 3(d)–(f) occur successively. Fig. A dashed line indicates the simultaneous sixtransistor XOR-XNR circuit used in 3(a). Despite this circuit's small size, its regenerative feedback

JNAO Vol. 14, Issue. 2, : 2023

pathways result in delayed transitions. Sum-carry generation (SCG), which is exacerbated by cascaded transmission gates, makes outputs more susceptible to errors. In Figure 1. When input C ="1," the Sum output (S) is generated more quickly in 3(b) (TFA22) in comparison to other input combinations. In addition, output S mav experience glitches due to the delayed arrival of the XOR-XNR signals at the SCG. Fig. In contrast, the control signals for the transmission gates in system 2 are distinct. 3(c) (BFA22) is reasonably synchronized, with the exception of its input signals, which include the early onset of input C when XOR01 is a likely scenario for glitch production at output S. Figure 2 shows that HFA26 is comparable to RFL22 and TFA22. 3(d) while operating at a higher speed, route delays vary. Fig. 3(e) (CMOS28) symbolizes the classic CMOS full adder, which is fairly glitch-resistant. Fig. depicts the proposed full adder (PBFA26). 3(f). Two things distinguish this layout from others. First, the internal signals are capacitively terminated at the SCG stage, and then, similar to Booth circuits, possible errors are absorbed by the transmission gate pairs in SCG. Second, a lowoverhead intra cell delay element, illustrated by M1 to M4 in Fig., is applied to synchronize all signals to the SCG. 3(f). Through their smaller than C g drain-source parasitic Cd /Cs, M1 and M4 supply the appropriate delay to the input C. C g of both M1 and M4 is not switched in comparison to an inverter-based delay element, resulting in a significant reduction in its parasitic contribution to the total adder's dynamic power. Consequently, C's presence can be autonomously handled without incurring large costs.

5.SIMULATION RESULT OF MULTIPLIER:



Fig 6 Simulation of Multiplication Outcomes Here, A=63 and B=62 are the inputs, and the output is 3906.

SYSTEM	POWER(*)	Deb(s(in)
Existing(Wallace tree multiplier)	0.200	32.681
Proposal mathod	0.215	24.305

Table 2 Comparison of time and force

6. CONCLUSION

To reduce parasitic and unforeseen power consumption in high-performance Booth multipliers, this research suggests and examines glitch-optimized circuit blocks. This goal is achieved by combining a PASR with circuit-level techniques. The proposed approach is ideal for energy-constrained, high-performance multiplication with a minor delay increase. Two multiplier structures (Prop-W, Prop-LFR) made of these circuit blocks were compared to the highly optimized array and tree multipliers made of the latest building blocks published in the literature. Postlayout calculations show that the proposed variants are 10% to 30% more power efficient than baselines.

REFERENCE

- M. D. Ercegovac and T. Lang, Digital Arithmetic (Morgan Kaufmann Series in Computer Architecture and Design). San Mateo, CA, USA: Morgan Kaufmann, 2004.
- 2. B. Dinesh, V. Venkateshwaran, P.Kavinmalar, and M. Kathirvelu, "Comparison of regular and tree based multiplier architectures with modified booth encoding for 4 bits on layout level using 45 nm technology," in Proc. Int.

JNAO Vol. 14, Issue. 2, : 2023

Conf. Green Comput. Commun. Elect. Eng., Mar. 2014, pp. 1–6.

- L. P. Rubinfield, "A proof of the modified Booth's algorithm for multiplication," IEEE Trans. Comput., vol. C-24, no. 10, pp. 1014– 1015, Oct. 1975, doi: 10.1109/T-C.1975.224114.
- P. Judd, J. Albericio, T. Hetherington, T. M. Aamodt, and A. Moshovos, "Stripes: Bit-serial deep neural network computing," in Proc. 49th Annu. IEEE/ACM Int. Symp. Microarchitecture, Oct. 2016, pp. 1–12.
- G. C. T. Chow, W. Luk, and P. H. W. Leong, "A mixed precision methodology for mathematicaloptimisation," in Proc. IEEE 20th Int. Symp. Field-Program. Custom Comput. Mach., Apr./May 2012, pp. 33–36.
- G. C. T. Chow, A. H. T. Tse, Q. Jin, W. Luk, P. H. Leong, and D. B. Thomas, "A mixed precision Monte Carlo methodology for reconfigurable accelerator systems," in Proc. ACM/SIGDA Int. Symp. Field Program. Gate Arrays, 2012, pp. 57–66.
- S. J. Schmidt and D. Boland, "Dynamic bitwidth assignment for efficient dot products," in Proc. Int. Conf. Field Program. Log. Appl., Sep. 2017, pp. 1–8.
- V. Sze, Y.-H. Chen, J. Emer, A.Suleiman, and Z. Zhang, "Hardware for machine learning: Challenges and opportunities," CoRR, vol. abs/1612.07625, Dec. 2016. [Online]. Available: https://arxiv.org/abs/1612.07625
- B. Rashidi, S. M. Sayedi, and R. R. Farashahi, "Design of a low-power and low-cost Boothshift/add multiplexer- based multiplier," in Proc. Iranian Conf. Elect. Eng. (ICEE), May 2014, pp. 14–19.
- 10. P. Devi, G. P. Singh, and B. Singh, "Low power optimized array multiplier with reduced area," in High Performance Architecture and Grid Computing, A. Mantri, S. Nandi, G. Kumar, and S. Kumar, Eds. Berlin, Germany: Springer, 2011, pp. 224–232.