Energy Efficient And High-Speed using A Rounding-Based Approximate Multiplier

U HimavarshaM.E, Assistant Professor[1] Saleema HumeraB.E Scholar[2]
M. Yashna ChowdaryB.E Scholar[3] K. MounikaB.E Scholar[4]
Department of ECE, Stanley College of Engineering and Technology for women, Chapel Rd, Abids, Telangana

Abstract : A multiplier is one of the important hardware blocks in most digital and high-performance systems such as microprocessors. With improving in technology, many researchers have tried and are trying to design multipliers which offer high speed, low power consumption and less area. However area and speed are two most important constraints. In this paper, we propose an approximate multiplier that is high speed yet energy efficient. The approach is to round the operands to the nearest exponent of two. This way the computational intensive part of the multiplication is omitted improving speed and energy consumption at the price of a small error. The proposed approach is applicable to both signed and unsigned multiplications. We propose three hardware implementations of the approximate multiplier that includes one for the unsigned and two for the signed operations. The efficiency of the proposed multiplier is evaluated by comparing its performance with those of some approximate and accurate multipliers using different design parameters. In addition, the efficacy of the proposed approximate multiplier is studied in two image processing applications, i.e., image sharpening and smoothing.

Index Terms— Accuracy, approximate computing, energy efficient, error analysis, high speed, multiplier.

1. Introduction

Energy minimization is major requirements in almost any electronic systems, especially the portable ones such as smart phones, tablets, and different gadgets. It is extremely desired to attain this minimization with minimal performance (speed) penalty [1]. Digital signal processing (DSP) blocks are most wanted in transportable components for realizing various multimedia applications. The computational core of these blocks is the ALU where the multiplications and additions are the major part [6]. The multiplications plays foremost operation in the processing elements which can leads to high consumption of energy and power. Many of the DSP cores implement image and video processing algorithms where final outputs are either images or videos prepared for human consumptions. It facilitates to go for approximations for improving the speed and energy in the arithmetic circuits. This originates from the limited perceptual abilities in observing an image or a video for human beings.

In addition to the image and video processing applications, there are other areas where the exactness of the arithmetic operations is not critical to the functionality of the system (see [2],[3]). Approximate computing provides an accuracy, speed and power/energy consumption. The advantage of approximate multiplier reduces the error rate and gain high speed. For correcting the division error compare operation and a memory look up is required for the each operand is required which increases the time delay for entire multiplication process [4].

At various level of abstraction including circuit, logic and architecture levels the approximation is processed [5]. In the category for approximation methods in function, a number of approximating arithmetic building blocks, such as adders and multipliers, at different design levels have been suggested in various structures [6],[7]. Broken array multiplier was designed for efficient VLSI implementation[8]. The error of mean and variance of the imprecise model increase by only 0.63% and 0.86% with reverence to the

precise WPA and the maximum error increases by 4%. Low-Power DSP uses approximate adders which are employed in different algorithms and design for signal processing. In contrast with standard multiplier, the dissipated power for the ETM dropped from 75% to 90%. While maintaining the lower average error from the conventional method, the proposed ETM achieves an impressive savings of more than 50% for a 12 x 12 fixed-width multiplication.

2. Literature Survey

Approximate Multipliers: Approximation can be performed using different techniques such as allowing some timing violations (e.g., voltage over scaling or over-clocking) and function approximation techniques (e.g., modifying the Boolean function of a circuit) or a mixture. Approximate multiplier designs mainly use three approximation approaches:

- Approximation in generating the partial products.
- Applying truncation in the partial product tree.
- Using approximate adders &compressors to accumulate the partial products.

P. Kulkarni et.al. Proposed multiplier architecture for 2x2 inaccurate multiplier done by Karnaugh map simplification for generating approximate partial products[10].It is an approximate 2x2 multiplier block by altering one entry in the K-Map of a 2x2 multiplier. Based on the 2x2 block, larger under designed multipliers (UDMs) can be built. This multiplier design introduces error in generating partial products while the adder tree remains accurate.

H. R. Mahdiani proposed multiplier called Broken-Array Multiplier (BAM)[12].It is very similar to the structure of an array multiplier. The BAM operates by omitting some carrysave adders in an array multiplier in both horizontal and vertical directions.

F. Farshchi, M. S. Abrishami Based on BAM, a BrokenBooth Multiplier (BBM) was presented [5]. Compared to BAM, BBM uses modified Booth algorithm to generate partial products and only omits carry-save adders to the right of a vertical line. BBM has a smaller power-delay-product (PDP) than BAM.

K. Khaing Yin, G. Wang Ling, and Y. KiatSeng proposed Error-Tolerant Multiplier. (ETM) is divided into a multiplication part for the MSBs and a non-multiplication part for the LSBs[11]. A NOR gate based control block is used to deal with two cases: a) If the product of the MSBs is zero, then the multiplication section is activated to multiply the LSBs without any approximation b) If the product of the MSBs is nonzero, the non-multiplication section is used as an approximate multiplier to process the LSBs, while the multiplication section is activated to multiply the MSBs.

K. Bhardwaj, P. S. Mane proposed approximate Wallace tree multiplier [4]. It is power and area efficient multiplier. AWTM is based on a bit-width aware approximate multiplication algorithm and Carry-in Prediction technique is employed which significantly reduces the critical path delay. BAM, BBM, ETM, AWTM multipliers uses approximation in the partial product tree. This compressor is used in a bottom-up tree topology to implement a high-speed, areaefficient and power-aware approximate multiplier. An inaccurate 4-bit Wallace multiplier based on a newly designed (4:2) approximate compressor. The inaccurate 4-bit multiplier is then used to build larger multipliers with error detection and correction circuits. This multiplier is referred to as Inaccurate Compressor based Multiplier (ICM).

3. Proposed Method

ROBA Multiplier

The main idea behind the proposed approximate multiplier is to make use of the ease of operation when the numbers are two to the power n (2n). To elaborate on the operation of the approximate multiplier, first, let us denote the rounded numbers of the input of A and B by Ar and Br, respectively. The multiplication of A by B may be rewritten as

$$A\ x\ B\ =\ (Ar\ -\ A)\ x\ (Br\ -\ B)\ +\ Ar\ x\ B\ +\ Br\ x\ A\ -\ Ar\ x\ Br\ ----1$$

The key observation is that the multiplications of $Ar \times Br, A\,r \times B, and\ Br \times A$ may be implemented just by the shift operation. The hardware implementation of $(Ar - A) \times (Br - B)$, however, is rather complex. The weight of this term in the final result, which depends on differences of the exact numbers from their rounded ones, is typically small. Hence, we propose to omit this part from (2), helping simplify the multiplication operation. Hence, to perform the multiplication process, the following expression is used:

$$A * B = (Ar * B) + (Br * A) - (Ar * Br)\ (2)$$

Thus, one can perform the multiplication operation using three shift and two addition/subtraction operations. In this approach, the nearest values for A and B in the form of 2n should be determined. When the value of A (or B) is equal to the $3 \times 2^{(P-2)}$(where p is an arbitrary positive integer larger than one), it has two nearest values in the form of 2n with equal absolute differences that are 2p and 2p−1. While both values lead to the same effect on the accuracy of the proposed multiplier, selecting the larger one (except for the case of p = 2) leads to a smaller hardware implementation for determining the nearest rounded value and hence, it is considered in this paper.

It originates from the fact that the numbers in the form of $3 \times 2^{(P-2)}$ are considered as do not care in both rounding up and down simplifying the process, and smaller logic expressions may be achieved if they are used in the rounding up. It should be noted that contrary to the previous work where the approximate result is smaller than the exact result, the final result calculated by the ROBA multiplier may be either larger or smaller than the exact result depending on the magnitudes of Ar and Br compared with those of A and B, respectively. Note that if one of the operands (say A) is smaller than its corresponding rounded value while the other operand (say B) is larger than its corresponding rounded value, then the approximate result will be larger than the exact result. This is due to the fact that, in this case, the multiplication result of $(Ar\ -\ A)\ \times (Br\ -\ B)$ will be negative.

Since the difference between (2) and (3) is precisely this product, the approximate result becomes larger than the exact one. Finally, it should be noted the advantage of the proposed ROBA multiplier exists only for positive inputs because in the two's complement representation, the rounded values of negative inputs are not in the form of 2n..Hence, we suggest that, before the multiplication operation starts, the absolute values of both inputs and the output sign of the multiplication result based on the inputs signs be determined and then the operation be performed for unsigned numbers and, at the last stage, the proper sign be applied to the unsigned result.

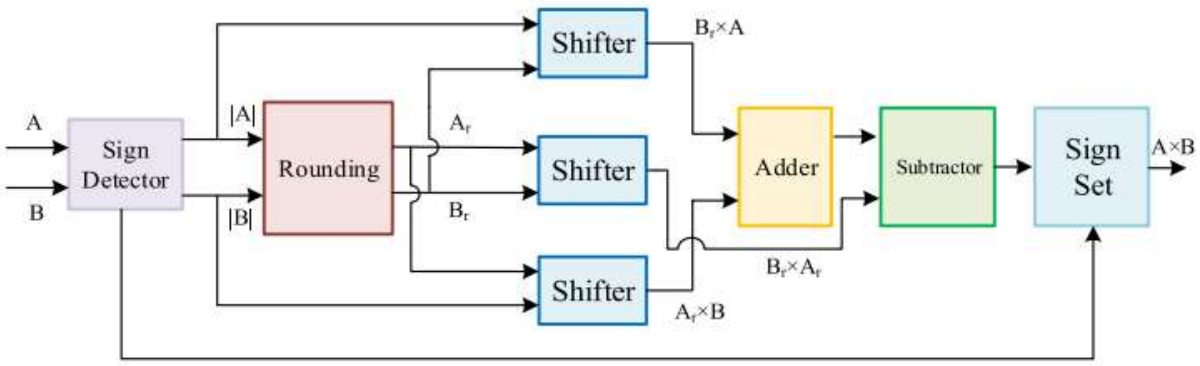

Fig. 1. Block diagram for the hardware implementation of the proposed multiplier.

B. Hardware Implementation of RoBA Multiplier

Based on (2), we provide the block diagram for the hardware implementation of the proposed multiplier in Fig. 1 where the inputs are represented in two's complement format. First, the signs of the inputs are determined, and for each negative value, the absolute value is generated. Next, the rounding block extracts the nearest value for each absolute value in the form of 2n. It should be noted that the bit width of the output of this block is n (the most significant bit of the absolute value of an n-bit number in the two's complement format is zero).

4. Simulation Results

Fig. 2. Output waveform using design 1

Fig. 3. Output waveform using design 2

Fig. 4. Output waveform using design 3

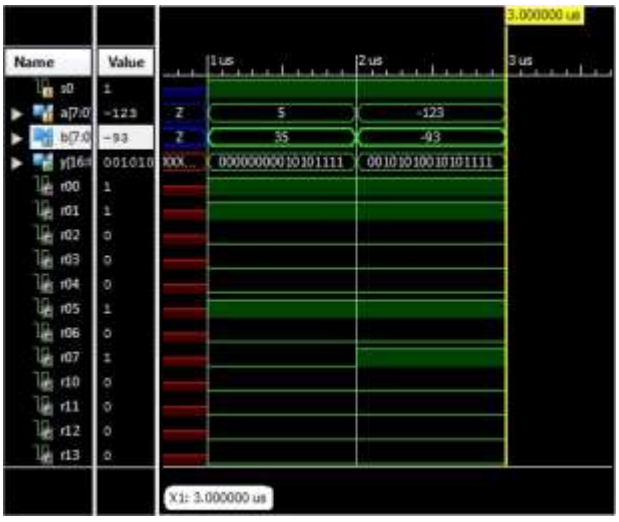

Fig. 5. Output waveform using design 4

Fig. 6 Output waveform using design 2 in binary format

Fig. 7 Total delay of the multiplier

Device Utilization Summary				[-]
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of 4 input LUTs	199	1,920	10%	
Number of occupied Slices	112	960	11%	
Number of Slices containing only related logic	112	112	100%	
Number of Slices containing unrelated logic	0	112	0%	
Total Number of 4 input LUTs	199	1,920	10%	
Number of bonded IOBs	33	66	50%	
Average Fanout of Non-Clock Nets	3.13			

Fig. 8 Device utilization summary

Fig. 9. Matlab Input Image

Fig.10.Image Sharpening with mask sharpening (pixel extraction) technique

Fig.11. Image Sharpening with mask sharpening technique

Fig.12. Image Sharpening with mask smoothing technique

Device	
Family	Spartan3e
Part	xc3s100e
Package	vq100
Grade	Commercial
Process	Typical
Speed Grade	-5

Environment	
Ambient Temp (C)	25.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	0

Characterization	
PRODUCTION	v1.2,06-23-09

On-Chip	Power (W)	Used	Available	Utilization (%)
Logic	0.000	251	1920	13
Signals	0.000	254	---	---
IOs	0.000	34	66	52
Leakage	0.034			
Total	0.034			

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	49.0	83.4	26.6

Supply Summary		Total	Dy
Source	Voltage	Current (A)	Cur
Vccint	1.200	0.008	
Vccaux	2.500	0.008	
Vcco25	2.500	0.002	

	Total	Dy
Supply Power (W)	0.034	

Fig. 13. Power consumption analysis

4. Conclusion

High-speed and energy efficient approximate multiplier were proposed. The RoBA multiplier had a high accuracy depend upon the 2n input form. The high exhaustive computation part is neglected to provide high performance. So hardware structural design is designed for S-RoBA, RoBA and AS-RoBA multiplier. The efficiencies of the RoBA multiplier were compared with some existing accurate and approximate multipliers with different parameters.

With the help of comparison table, RoBA multiplier provides the better area, power, and energy efficient when compared with some already proposed accurate and approximate multiplier. The negation of output bit in the signed multiplier causes the maximum error rate. Therefore in future, the maximum error rates can be compact by minimizing the error rate equation which is identical to the error rate equation of unsigned RoBA multiplier. In future, the proposed multiplier will be applied in the various images processing applications.

References

[1]Reza Zendegani; Mehdi Kamal; MiladBahadori; Ali Afzali-Kusha; MassoudPedram,"RoBA Multiplier: A Rounding-Based Approximate Multiplier for High Speed yet Energy-Efficient Digital Signal Processing" IEEE Trans. Very Large Scale Integr. (VLSI) Syst.,pp.1-9, July 2016.

[2] S.Narayanamoorthy, H. A. Moghaddam, Z. Liu, T. Park, and N. S. Kim, "Energy-efficient approximate multiplication for DSP and classification applications," IEEE Trans.Very Large Scale Integr. (VLSI) Syst., vol. 23, no.6, pp.1180– 1184, Jun. 2015.

[3]S.Hashemi, R.I. Bahar, and S. Reda, "DRUM: A dynamic range unbiased multiplier for approximate applications," in Proc.IEEE/ACM Int.Conf.Comput.-Aided Design (ICCAD), Austin, TX, USA, 2015, pp. 418–425.

[4] K. Bhardwaj, P. S. Mane, and J. Henkel, "Power&areaefficient approximate wallace tree multiplier for errorresilient systems," in Proc. 15th Int. Symp. Quality Electron. Design, 2014, pp. 263–269.

[5]F.Farshchi, M.S.Abrishami, and S.M. Fakhraie,"New approximate multiplier for low power DSP," in Proc.17th Int. Symp. Comput.Archit.Digit. Syst.Oct.2013, pp.25–30.

[6] C.-H. Lin and I.-C. Lin, "High accuracy approximate multiplier with error correction," in Proc. 31st Int. Conf. Comput. Design (ICCD), 2013, pp. 33–38.

[7] M. Alioto, "Ultra-low power VLSI circuit design demystified and explained: A tutorial," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 59, no. 1, pp. 3–29, Jan. 2012.

[8]R. Venkatesan, A. Agarwal, K. Roy, and A. Raghunathan, "MACACO: Modeling and analysis of circuits for approximate computing," in Proc. Int. Conf. Comput.-Aided Design, Nov. 2011, pp. 667–673.

[9] P. Kulkarni, P. Gupta, and M. Ercegovac, "Trading accuracy for power with an under designed multiplier architecture," in Proc. 24th Int. Conf.VLSI Design, Jan. 2011, pp. 346–351.

[10] K. Khaing Yin, G. Wang Ling, and Y. Kiat Seng, "Low-power high speed multiplier for error-tolerant application," in Proc. EDSSC, 2010, pp. 1–4.

[11] H. R. Mahdiani, A. Ahmadi, S. M. Fakhraie, and C. Lucas, "Bio-inspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 4, pp. 850–862, Apr. 2010.

[12] Chip-Hong Chang , Ravi Kumar Satzoda, "A Low Error and High Performance Multiplexer-Based Truncated Multiplier", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Volume: 18, Issue: 12,pp: 1767 – 1771, Dec. 2010.

[13]Vinay K. Chippa, DebabrataMohapatra, Anand Raghunathan, Kaushik Roy, Srimat T. Chakradhar, "Scalable effort hardware design: Exploiting algorithmic resilience for energy efficiency", Design Automation Conference (DAC), 47th ACM/IEEE, pp:555-560,Dec.2010.

[14]R.Hegde N.R.Shanbhag,"Soft digital signal processing", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Volume: 9, Issue: 6, pp:813 – 823, Dec. 2001.