DESIGN OF APPROXIMATE MULTIPLICATION CIRCUITS FOR REDUNDANT BINARY MULTIPLIERS

KARANAM SWATHI[1], D. INDRASENA REDDY[2]
[1]PG SCHOLAR.DEPT OF ECE, SKD ENGINEERING COLLEGE, GOOTY, AP, INDIA
[2] ASSISTANT PROFESSOR, DEPT OF ECE, SKD ENGINEERING COLLEGE, GOOTY, AP, INDIA

Abstract: As technology scaling is reaching its limits, new approaches have been proposed for computational efficiency. Approximate computing is a promising technique for high performance and low power circuits as used in error-tolerant applications. Among approximate circuits, approximate arithmetic designs have attracted significant research interest. In this paper, the design of approximate redundant binary (RB) multipliers is studied. The approximate design of a radix-4 Booth multiplier as one of the most popular schemes for unsigned multiplication. a radix-4 Booth multiplier, a radix-4 modified Booth encoding (MBE) is used to generate the partial products a radix-4 MBE can reduce the number of partial products by a factor of two. Two approximate Booth encoders and two RB 4:2 compressors based on RB (full and half) adders are proposed for the RB multipliers. The approximate design of the RB-Normal Binary (NB) converter in the RB multiplier is also studied by considering the error characteristics of both the approximate Booth encoders and the RB compressors. Both approximate and exact regular partial product arrays are used in the approximate RB multipliers to meet different accuracy requirements. Error analysis and hardware simulation results are provided. The proposed approximate RB multipliers are compared with previous approximate Booth multipliers; the results show that the approximate RB multipliers are better than approximate NB Booth multipliers especially when the word size is large. Case studies of error-resilient applications are also presented to show the validity of the proposed designs.

1.INTRODUCTION

ENERGY minimization is one of the main design requirements in almost any electronic systems, especially the portable ones such as smart phones, tablets, and different gadgets. It is highly desired to achieve this minimization with minimal performance (speed) penalty. Digital signal processing (DSP) blocks are key components of these portable devices for realizing various multimedia applications. The computational core of these blocks is the arithmetic logic unit where multiplications have the greatest share among all arithmetic operations performed in these DSP systems. Therefore, improving the speed and power/energy-efficiency characteristics of multipliers plays a key role in improving the efficiency of processors.

We are at the threshold of an explosion in new data, produced not only by large, powerful scientific and commercial computers, but also by the billions of low-power devices of various kinds. While traditional workloads including transactional and database processing continue to grow modestly, there is an explosion in the computational footprint of a range of applications that aim to extract deep insight from vast quantities of structured and unstructured data. There is an exactness implied by traditional computing that is not needed in the processing of most types of these data. Yet today, these cognitive applications continue to be executed on general purpose (and accelerator) platforms that are highly precise and designed with reliability from the ground up. Approximate computing aims to relax these constraints with the goal of obtaining significant gains in computational throughput - while still maintaining an acceptable quality of results.

A primary goal of research in approximate computing is to determine what degrees of approximations in the several layers of the system stack (from algorithms down to circuits and semi-conductor devices) are feasible so that the produced results are acceptable, albeit possibly different from those obtained using precise computation. Approximate computing techniques studied by various researchers have focused primarily on optimizing one layer of the system stack and have shown benefits in power or execution time. In this work we set out to investigate if combining multiple approximation techniques spanning more than one layer of the system stack compounded the benefits, and if these compounded benefits are widely applicable across different application domains.

In order to provide a concrete demonstration, we focused on three approximation categories: skipping computations, approximation of arithmetic computations themselves, and approximation of communication between computational elements. As representatives of each category we evaluated loop perforation, reduced arithmetic precision, and relaxation of synchronization. We selected applications that are computationally expensive but have the potential to significantly impact our lives if they became cheap and pervasive. Our applications spanned the domains of digital signal processing, robotics, and machine learning. Across the set of applications studied, our results show that we were able to perforate hot loops in the studied applications by an average of 50%, with proportional reduction in overall execution time, while still producing acceptable quality of results. In addition, we were able to reduce the width of the data used in the computation to 10-16 bits from the currently common 32 or even 64 bits, with potential for significant performance and energy benefits. In the parallel applications we studied, we were able to reduce execution time by 50% through partial elimination of synchronization overheads.

2.LITERATURE SURVEY

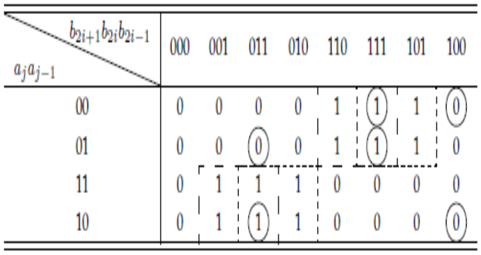
Wireless sensor networks (WSNs) are today widely employed in real world applications. However, their lifetime is still challenging and the most critical limitation for the success of this technology. In fact, wireless sensors nodes, which are the backbone of the network, are typically powered by limited energy storage devices (i.e. small batteries or supercaps) and their short lifetime is a critical issue. Wake-up radio receivers are very effective in minimizing idle listening. This fact has resulted in a significant number of wake-up radio receiver architectures proposed in last decade. In this work we present an advanced design and implementation of an advanced wake-up radio that is capable of both processing the received data (i.e. for addressing) and retransmitting data or wake up messages to the neighbours when necessary. With these features it can be possible to further enhance the energy efficiency of the communication and allowing ultra-low power multi-hop communication. Experimental results demonstrate the functionality as well as the power and range of the proposed design which is ready for future energy efficient and pure-asynchronous MAC protocols.

Low power is an imperative requirement for portable multimedia devices employing various signal processing algorithms and architectures. In most multimedia applications, human beings can gather useful information from slightly erroneous outputs. Therefore, we do not need to produce exactly correct numerical outputs. Previous research in this context exploits error resiliency primarily through voltage overscaling, utilizing algorithmic and architectural techniques to mitigate the resulting errors. In this paper, we propose logic complexity reduction at the transistor level as an alternative approach to take advantage of the relaxation of numerical accuracy. We design architectures for video and image compression algorithms using the proposed approximate arithmetic units and evaluate them to demonstrate the efficacy of our approach. We also derive simple mathematical models for error and power consumption of these approximate adders. Furthermore, we demonstrate the utility of these approximate adders in two digital signal processing architectures (discrete cosine transform and finite impulse response filter) with specific quality constraints. Simulation results indicate up to 69% power savings using the proposed approximate adders, when compared to existingimplementations using accurate adders.

3.PROPOSED SYSTEM

Two approximate Booth encoders are designed based on the conventional modified Booth encoding method and the new modified Booth encoding method, respectively.

3.1Radix-4 Approximate MBE



$b_{2i+1}b_{2i}b_{2i-1}$ a_ja_{j-1}	000	001	011	010	110	111	101	100
00	0	0	0	0	1	1	1	0
01	0	0	0	0	1	1	1	0
11	0	1	1	1	0	0	0	0
10	0	1	1	1	0	0	0	0

TABLE 4 :K-Map of R4AMBE6

The K-map of the radix-4 approximate modified Booth encoder (R4AMBE6), i.e., appij6□1, with 6 errors in the Kmap is shown in Table 1, where 0 denotes an entry in which a ’1’ is replaced by a ’0’ and 1 denotes a ’0’ entry that has been replaced by a ’1’. Only 6 entries are modified to simplify the Booth encoding. This approximate design relies on the property that the truth table is as symmetrical as possible for a design with the least complexity. Therefore, three modifications change a ’1’ to a ’0’ and three modifications change a ’0’ to a ’1’ in the K-map. The output of R4AMBE6 is given as follows:

$$app_{ij6-1} = (b_{2i} + b_{2i-1})(b_{2i+1} \oplus a_i)$$

$$E_i = (b_{2i+1}\overline{b_{2i}}) + (b_{2i+1}\overline{b_{2i-1}})$$

Compared with the exact MBE, R4AMBE6 can significantly reduce both the complexity and the critical path delay of Booth encoding. The error rate, denoted by Pbe, is given by:

$$P_{be} = 6/32 = 18.75\%$$

The gate level structure of R4AMBE6 is shown in Fig.1. The conventional design of MBE consists of four XNOR-2 gates, one XOR-2 gate, one OR-3 gate, one OR- 2 gate and one NAND-2 gate. The R4AMBE6 design only requires one XOR-2 gate, one AND-2 gate and one OR-2 gate.

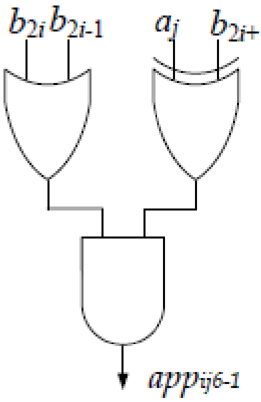


Fig. 1. The gate-level circuit of the proposed R4AMBE6.

3.2 Radix-4 Approximate NMBE

The approximate Radix-4 with the new modified Booth Encoding (R4ANMBE6), i.e., appij6--1, with 6 errors in the K-map is shown in Table 1. In this approximate design, there are more entries changed from ’0’ to ’1’ than those changed from ’1’ to ’0’. Therefore, the approximate results produced by R4ANMB6 will be usually larger than its exact counterpart. From Table 5, the approximate ppij is derived as follows:

$$app'_{ij6-1} = b_{2i+1} \oplus a_j + b_{2i}b_{2i}$$

This design further reduces the complexity of the correction term (i.e., Ei). Its error rate is the same as R4AMBE6:

$$P'_{be} = 6/32 = 18.75\%$$

The gate level circuit of R4ANMBE6 is shown in Fig. 2. The R4AMBE6 design only requires one XOR-2 gate, one AND-2 gate and one OR-2 gate, which has the same complexity as R4AMBE6.

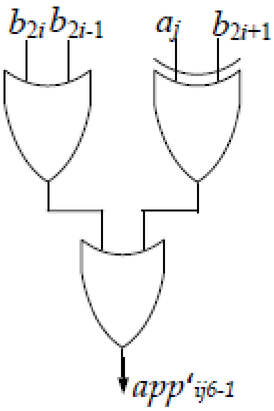


Fig. 2. The gate-level circuit of the proposed R4AMBE6.

3.3 The Proposed Approximate RB 4:2 Compressors

The gate level structure of the proposed design shows that the critical path of this compressor has still a delay of 3 , so it is the same as for the exact compressor of Figure 5. However, the propagation delay through the gates of this design is lower than the one for the exact compressor. For example, the propagation delay in the XOR*gate that generates both the XOR and XNOR signals in [8], is higher than the delay through a

XNOR gate of the proposed design. Therefore, the critical path delay in the proposed design is lower than in the exact design and moreover, the total number of gates in the proposed design is significantly less than that in the optimized.

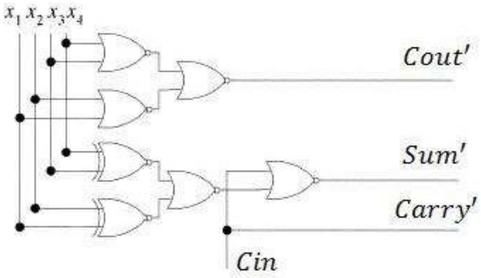


Figure 6. Gate level implementation of Design

3..4 Design of Approximate RB Multipliers

In this section, the approximate RB multipliers are designed as follows. The proposed approximate Booth encoders, i.e., R4AMBE6 and R4ANMBE6, are used to generate approximate PPs. Approximate RB compressors, i.e., ARBC-1 and ARBC-2, are used for RB PP reduction, which can reduce the delay for compression and significantly improve speed performance when the operand size is a power of 2. The approximate RB-NB converter (made of NOR gates) is used to convert the RB digit to the NB digit.

An approximation factor p (p=1, 2, ..., 2N) that has been proposed in [21] is used. This is defined as the number of least significant PP columns that are generated by the approximate Booth encoders. As p column PPs are already approximate, the approximate PPs can be accumulated with an approximate RB 4:2 compressor to further improve speed and reduce power consumption. For the same reason, the p least significant RB digits are also converted by the approximate RB-NB converter to calculate the final product.

Four approximate RB multipliers are proposed. They use the exact regular PP array when p _ (N - 4) (as detailed in [36]), and the approximate regular PP array when p >(N - 4) where the bit pairs (E2, 0) and (E3, 1) of Fig. 4 can be ignored in the approximate design of the RB Booth multipliers; however they all use the proposed approximate RB-NB converter. For the 2N-p most significant PP columns, the exact design is used for the final results.

b_p	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A								a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0
B									b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
PP_0^-						$\overline{p_{09}^-}$	p_{09}^-	p_{08}^-	p_{07}^-	p_{06}^-	p_{05}^-	p_{04}^-	p_{03}^-	p_{02}^-	p_{01}^-	p_{00}^-
PP_0^+						$\overline{p_{08}^+}$	p_{07}^+	p_{06}^+	p_{05}^+	p_{04}^+	p_{03}^+	p_{02}^+	p_{01}^+	p_{00}^+	0	0
PP_1^-		$\overline{p_{19}^-}$	p_{19}^-	p_{18}^-	p_{17}^-	p_{16}^-	p_{15}^-	p_{14}^-	p_{13}^-	p_{12}^-	p_{11}^-	p_{10}^-	1	E_1	1	E_0
PP_1^+		$\overline{p_{18}^+}$	p_{17}^+	p_{16}^+	p_{15}^+	p_{14}^+	p_{13}^+	p_{12}^+	p_{11}^+	p_{10}^+	0	0	0	1	0	0
										E_3	1	E_2				
										1	0	0				

Fig. 3. RB PP generation of an 8-bit RB multiplier using a Booth encoder.

The four RB multipliers are different in the p PP columns as follows:

1) The first approximate RB multiplier (R4ARBM1) uses R4AMBE6 to generate the p least significant PP columns and ARBC-1 to perform the approximate PP accumulation.

2) The second approximate RB multiplier (R4ARBM2) uses R4AMBE6 to generate the p least significant PP columns and ARBC-2 for the corresponding approximate PP accumulation.

3) The third approximate RB multiplier (R4ARBM3) uses R4ANMBE6 to generate the p least significant PP columns and ARBC-1 to perform the approximate PP accumulation.

4) The fourth approximate RB multiplier (R4ARBM4) uses R4ANMBE6 to generate the p least significant PP columns and ARBC-2 to perform the approximate PP accumulation.

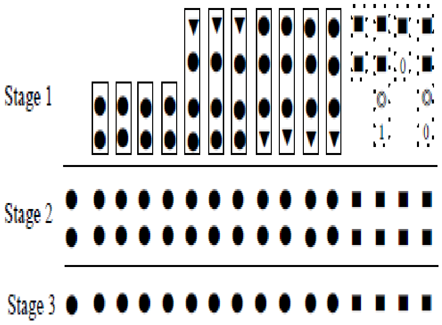
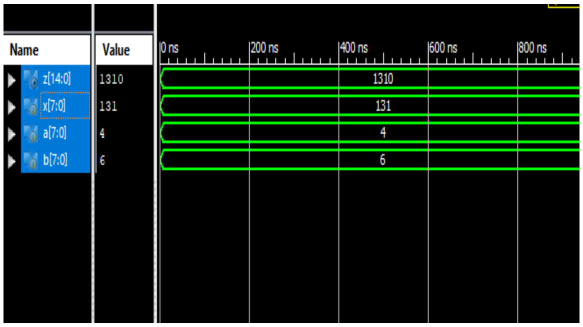


Fig. 4. The dot diagram of the proposed 8-bit approximate RB multiplier (R4ARBM) with p=4.

As the error can be controlled by the approximation factor p, a reasonable accuracy can be achieved for different applications. Fig. 9 shows an approximate 8-bit RB multiplier with p=4 using an approximate Booth encoder, an approximate RB compressor, an approximate RB-NB converter, and an exact regular PP. A box with a solid line denotes the use of an exact RB compressor, and a box with a dotted line denotes an approximate RB 4:2 compressor. The exact PP is represented by l , the modified PP after logic simplification is represented by t , while the approximate PP term is represented by _. _ represents Ei.

4. RESULTS



5. COMPARISON TABLE

SYSTEM	POWER(w)	Delay(ns)
Existing Method	0.00583	8.78ns
Proposed Method	0.00190	6.07ns

6.CONCLUSION

The proposed approximate Booth encoders on conventional exact Booth encoders (R4AMBE6) and new exact Booth encoders (R4ANMBE6) have moderate error and energy consumption; also they achieve a very good tradeoff between error and performance. Approximate RB multipliers have been designed based on approximate Booth encoders, approximate RB 4:2 compressors, (exact and approximate) regular partial product arrays, and approximate RB-NB converters. Simulation show that the approximate RB multipliers are very good designs when considering the NEP as metric.

5. FUTURE SCOPE

This multipliers plays a very important role in our day to day life. In future the multipliers are going to play a major role. The speed of the multipliers is increased by using carry save adders, carry look ahead adder, and so on. Rounding patterns will be optimized based on required accuracy and different compression techniques. The area and delay can be reduced in future by using advanced technology.

BIBLIOGRAPHY

[1] 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.

[2] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-power digital signal processing using approximate adders," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 32, no. 1, pp. 124–137, Jan. 2013.

[3] 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.

[4] 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.

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

[6] P. Kulkarni, P. Gupta, and M. Ercegovac, "Trading accuracy for power with an underdesigned multiplier

architecture," in Proc. 24th Int. Conf. VLSI Design, Jan. 2011, pp. 346–351.

[7] D. R. Kelly, B. J. Phillips, and S. Al-Sarawi, "Approximate signed binary integer multipliers for arithmetic data value speculation," in Proc. Conf. Design Archit. Signal Image Process., 2009, pp. 97–104.

[8] K. Y. Kyaw, W. L. Goh, and K. S. Yeo, "Low-power high-speed multiplier for error-tolerant application," in Proc. IEEE Int. Conf. Electron Devices Solid-State Circuits (EDSSC), Dec. 2010, pp. 1–4.

[9] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "Design and analysis of approximate compressors for multiplication," IEEE Trans. Comput., vol. 64, no. 4, pp. 984–994, Apr. 2015.

[10] K. Bhardwaj and P. S. Mane, "ACMA: Accuracy-configurable multiplier architecture for error-resilient system-on-chip," in Proc. 8th Int. Workshop Reconfigurable Commun.-Centric Syst.-Chip, 2013, pp. 1–6.

[11] K. Bhardwaj, P. S. Mane, and J. Henkel, "Power- and area-efficient approximate wallace tree multiplier for error-resilient systems," in Proc. 15th Int. Symp. Quality Electron. Design (ISQED), 2014, pp. 263–269.