www.ijiarec.com

ISSN:2348-2079

Volume-7 Issue-1

International Journal of Intellectual Advancements and Research in Engineering Computations

Approximate hybrid high radix-4096 encoding for energy efficient inexact multipliers

V.Saravanan[1], P.Susan Peace[2]

Associate Professor[1]/ VLSI Design PG scholar[2]

Department of ECE, Knowledge Institute of Technology, Salem.

ABSTRACT

Approximate computing forms a design alternative that exploits the intrinsic error resilience of various applications and produces energy-efficient circuits with small accuracy loss. The approximate hybrid high radix encoding is generated in the partial products in signed multiplications that encode the most significant bits with the accurate radix-4 encoding and the least significant bits with an approximate higher radix encoding. The approximations are performed by rounding the high radix values to their nearest power of two. The approximate encoding is commonly used in signed multipliers although these techniques perform fast multiplications. The number of the partial products is reduced in the design. In radix-4 encoder the partial product tree is reduced and the number of radix increases. The MSBs of the multiplicand are encoded with the accurate radix-4 encoding, while its *k* LSBs are encoded with an approximate high radix-2^k encoding, with *k* being a configuration parameter that adjusts the tradeoff between accuracy and energy consumption.

INTRODUCTION

Multipliers are widely used in arithmetic units of microprocessors, multimedia and digital signal processors. Moreover, high performance and low power multipliers are in high demand for embedded systems. It is becoming extremely difficult to further improve performance and reduce the power consumption of multipliers under the requirement of full accuracy. High precision and exactness in the operations of digital logic circuits are related to the generally accepted requirement of correctness of information processing; numerous error-tolerant applications can be found in computing and by relaxing the requirement of strict accuracy, performance and power consumption can be substantially improved. This design principle is generally known as approximate or inexact computing.

Digital Signal Processing (DSP) is the numerical manipulation of signals, usually with the intention to measure, filter, produce or compress continuous analog signals. It is characterized by the use of digital signals to represent these signals as discrete time, discrete frequency, or other discrete domain signals in the form of a sequence of numbers or symbols to permit the digital processing of these signals.

Digital signal processing and analog signal processing are subfields of signal processing.DSP applications include audio and speech signal processing, sonar and radar signal processing, sensor array processing, spectral estimation, statistical signal processing, digital image processing, signal processing for communications, control of systems, biomedical signal processing, Seismic data processing, among others.

DSP algorithms have long been run on standard computers, as well as on specialized processors called digital signal processors, and on purpose-built hardware such as application-specific integrated circuit.

Author for correspondence:
Department of ECE, Knowledge Institute of Technology, Salem.

Currently, there are additional technologies used for digital signal processing including more powerful general purpose microprocessors, Field-Programmable Gate Arrays (FPGAs), digital signal controllers (mostly for industrial applications such as motor control), and stream processors, among other.

Digital signal processing can involve linear or nonlinear operations. Nonlinear signal processing is closely related to nonlinear system identification and can be implemented in the time, frequency, and spatio-temporal domains.

Achieving high energy efficiency has become a key design objective for embedded and mobile computing devices due to their limited battery capacity and power budget. To improve energy efficiency of such computing devices, significant efforts have already been devoted at various levels, from software to architecture, and all the way down to circuit and technology levels. Embedded and mobile computing devices are frequently required to execute some key DSP and classification applications.

Problem statement

The pre-encoded Non Redundant Radix-4 Signed Digit multipliers with carry save adder designs are need more area. And multipliers need extra hardware to generate the signals for the ModelSim. So, access time is increased and it reduces speed.

Objective of the work

The objective of good multiplier is to provide a physically compact high speed and low power consumption unit. Being a core part of arithmetic processing unit multipliers are in extremely high demand on its speed and low power consumption. To save significant power consumption of multiplier design it is a good direction to reduce number of operations there by reducing a dynamic power which is a major part of total power dissipation. This was achieved by using Radix-4096 encoder.

Scope of the work

Radix-4 algorithm reduces the number of partial products from N to N/3, where N is the number of multiplier bits. Thus it allows a time gain in the partial products summation. By this way partial products and number of transistors are reduced. As a result, significant area savings are observed and the critical path delay of the recoding process is reduced which improves the speed. Also the power consumption was reduced.

LITERATURE SURVEY

Akhilesh Tyagi, et al (1993), described in carry select and parallel-prefix adders to derive a more area-efficient implementation for both the cases. The carry-select scheme is assessed relative to carry-ripple, classical carry-select, and carry-skip adders. The analytic evaluation is done with respect to the gate-count model for area and gate-delay units for time. The gate-count of the carry-select scheme is 25%higher than that of a carry-skip adder for a 30% gain in speed. The select-prefix adder when built with parallel-prefix blocks of equal size gives rise to a family of select-prefix adders that cover the area-time performance gap.

C. Liu, et al (2014), considered for error-tolerant applications that can tolerate some loss of accuracy with improved performance and energy efficiency. Multipliers are
key arithmetic circuits in many such applications such as digital signal processing (DSP). A novel approximate multiplier with lower power consumption and a shorter critical path than traditional multipliers is for high-performance DSP applications. This multiplier leverages a newly-designed approximate adder that limits its carry propagation to the nearest neighbors for fast partial product accumulation.

Georgios Zervakis, et al (2015), described the significant attention as a promising strategy to decrease power consumption of inherently error tolerant applications. The hardware-level approximation is introduced in the partial product perforation technique for designing approximate multiplication circuits. The mathematically rigorous manner is performed in partial product; the imposed errors are bounded and predictable, depending only on the input distribution. Through extensive experimental evaluation, the partial product perforation method on different multiplier architectures and expose the optimal architecture

perforation configuration pairs for different error constraints.

H. Jiang, et al (2016), described the high performance signed multiplication by encoding and thereby reducing the number of partial products. A multiplier using the radix-4 (or modified Booth) algorithm is very efficient due to the ease of partial product generation, whereas the radix-8 Booth multiplier is slow due to the complexity of generating the odd multiples of the multiplicand. An approximate 2-bit adder is deliberately designed for calculating the sum of 1_ and 2_ of a binary number. This adder requires a small area, a low power and a short critical path delay.

METHODOLOGY

Radix multipliers

High radix encoding offer partial products reduction, and as a result, the accumulation requires smaller trees, leading to energy, area and delay savings. However, high radix encodings require complex encoding and partial product generation circuits, negating thus the benefits of the partial products reduction. In this section, the hybrid high radix encoding and the performed approximations for simplifying its circuit complexity are presented. In the technique, the multiplicand *B* is encoded using the approximate high radix encoding, generating B and the approximate multiplication A B is performed. Finally, its adaptation on inexact 16-bit hardware multipliers is described, and a qualitative analysis is conducted, targeting to estimate the potential area gains.

Hybrid high radix encoding

In the hybrid high radix encoding, *B* is divided in two groups: the MSB part of $n-k$ bits and the LSB part of k bits. The configuration parameter, $k \geq 4$, is an even number, namely, $k = 2m$: $m \in Z$, with $m \geq 2$. The MSB part is encoded using the radix-4 (modified Booth) encoding, while the LSB part is encoded with the high radix-2^k encoding.

The hybrid high radix encoding is characterized by increasing logic complexity, due to the high radix values, that are not power to two. However, in order to retain high accuracy, the radix-4 encoding of the MSB is performed accurately.

$R2^k$ Digit		Output				
$y_0^{R2^k}$	$\hat{y}_0^{R2^k}$	*sign*	$\times 2^{k-1}$	$\times 2^{k-2}$	$\times 2^{k-3}$	$\times 2^{k-4}$
$[0, 2^{k-5})$	0	0	0	0	0	0
$[2^{k-5}, 2^{k-4}+2^{k-5})$	2^{k-4}	0	0	0	0	**1**
$[2^{k-1}+2^{k-5}, 2^{k-3}+2^{k-4})$	2^{k-3}	0	0	0	**1**	0
$[2^{k-3}+2^{k-4}, 2^{k-2}+2^{k-3})$	2^{k-2}	0	0	**1**	0	0
$[2^{k-2}+2^{k-3}, 2^{k-1})$	2^{k-1}	0	**1**	0	0	0
$[-2^{k-1}, -2^{k-2}-2^{k-3})$	-2^{k-1}	**1**	**1**	0	0	0
$[-2^{k-2}-2^{k-3}, -2^{k-3}-2^{k-4})$	-2^{k-2}	**1**	0	**1**	0	0
$[-2^{k-3}-2^{k-4}, -2^{k-4}-2^{k-5})$	-2^{k-3}	**1**	0	0	**1**	0
$[-2^{k-1}-2^{k-5}, -2^{k-5})$	-2^{k-1}	**1**	0	0	0	**1**
$[-2^{k-5}, 0)$	0	**1**	0	0	0	0

Table 3.2. Approximate radix-2^k encoding

Input			R4 Digit	Output		
b_{2j+1}	b_{2j}	b_{2j-1}	y_j^{R4}	$sign_j$	$\times 2_j$	$\times 1_j$
0	0	0	0	0	0	0
0	0	1	1	0	0	1
0	1	0	1	0	0	1
0	1	1	2	0	1	0
1	0	0	-2	1	1	0
1	0	1	-1	1	0	1
1	1	0	-1	1	0	1
1	1	1	0	1	0	0

Table 3.1. Accurate radix-4 encoding

In particular, in the approximate encoding, all the values that are not power of two and the k-4 smallest powers of two as well, are rounded of the 4 largest powers of two of 0, so that the sum of all the values of the approximate digit is 0.

So, that to keep only the four largest powers of two and the radix-2 encoding circuit requires only about the double area in comparison with the accurate radix-4 encoder.

2radix-4 modified booth encoder

Booth algorithm is a powerful algorithm for signed number multiplication, Since a k-bit binary number can be interpreted as k/2-digit Radix-4 number, a k/3-digit Radix-8 number and so on, it can deal with more than one bit of the multiplier in each cycle by using high radix multiplication. The major disadvantage of the Radix-2 algorithm was that the process required n shifts and an average of n/2 additions for an n bit multiplier.

This variable number of shift and add operations is inconvenient for designing parallel multipliers. The Radix-4 modified Booth algorithm overcomes all these limitations of Radix-2 algorithm. For operands equal to or greater than 16 bits, the modified Radix-4 Booth algorithm has been widely used.

Table. 3.4. Recoding Table for Radix-4 Booth Multiplier

Block	Recoded Digit	Operation on Multiplicand
000	0	0*Multiplicand
001	+1	+1*Multiplicand
010	+1	+1*Multiplicand
011	+2	+2*Multiplicand
100	-2	-2*Multiplicand
101	-1	-1*Multiplicand
110	-1	-1*Multiplicand
111	0	0*Multiplicand

It is based on encoding the two's complement multiplier in order to reduce the number of partial products to be added to n/2.

Booth algorithm which scans strings of three bits is given below:

- Extend the sign bit 1 position if necessary to ensure that n is even.
- Append a 0 to the right of the LSB of the multiplier.
- According to the value of each vector, each Partial Product will be 0,+M,-M,+2M or -2M.

Hybrid encoding for radix

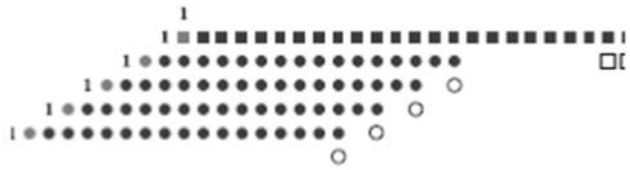

Fig 3.3. Partial product tree based on the hybrid encoding of accurate radix-4 and approximate radix-256

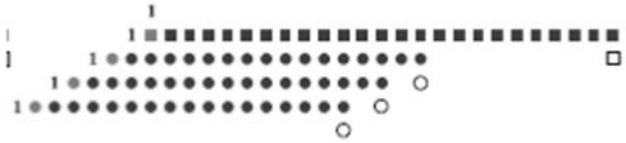

Fig 3. 4. Partial product tree based on the hybrid encoding of accurate radix-4 and approximate radix-1024 encoding

The approximate hybrid high radix encoding is designed in the energy-error efficient inexact multipliers. The difference with the related work is that it concerns approximations on the generation of the partial products, and can be combined with any accumulation technique, approximate or not.

Another significant aspect of the error imposed depends only on the configuration parameter *k*, and as a result, it can be calculated without the need for exhaustive simulations.

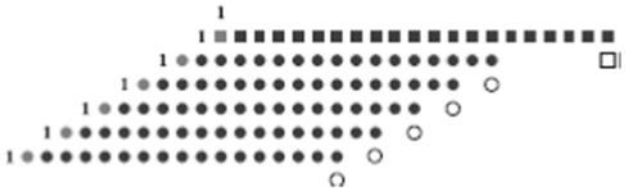

Fig 3.2. Partial product tree based on the hybrid encoding of accurate radix-4 and approximate radix-64

In radix-256 encoder, to increase the number of radix by reducing the partial product tree as shown in above fig 3.2.

In radix-1024 encoder, to increase the number of radix by reducing the partial product tree as shown in above fig 3.4.

Consequently, a precise estimation of the output quality can be extracted for the application's inputs, giving the flexibility to target the maximum energy reduction for a specific error bound.

The RAD256 multiplier and the distribution of relative error distance (RED) according to the encoded operand *B* for the multipliers RAD2^k, *k*= 6, 8, 10 are presented. The pdf of RED is similar for the other two multipliers and shows that the probability of having small RED is high. The error distribution follows a Gaussian distribution with bounds for maximum error.

Approximate hybrid encoding for radix-4096

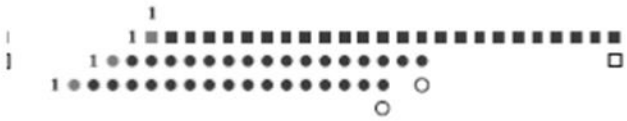

Fig 3. 5. Partial product tree based on the hybrid encoding of accurate radix-4 and approximate radix-4096 encoding

The approximate hybrid high radix multiplier is performed in the simple logic, resulting in fast operation and low power performance. Although overhead is added because of the encoding circuits, it is insignificant because of the approximations made. Also, this offset is compensated with the partial product generators that deliver low area, and the reduction of the number of the partial products.

The accurate radix-4 n-bit multiplier uses $n/2$ radix-4 encoders and $n^2/2$ radix-4 partial product generators to produce the $n/2$ n-bit partial products. Similarly, the approximate radix-2^k , with $k \geq 4$, requires $(n-k)/2$ radix-4 encoders, 1 radix-2^k encoder, $n(n-k)/2$ radix-4 partial product generators, and $n+k-2$ radix-2^k partial product generators.

The total number of area unit gates required in the accurate radix-4 16-bit multiplier. The approximate multipliers (RAD64, RAD256, RAD1024 and RAD4096) are described. The results concern the radix encoding, the partial product generation, the partial product accumulation, and the final addition. The area reduction is achieved in the design.

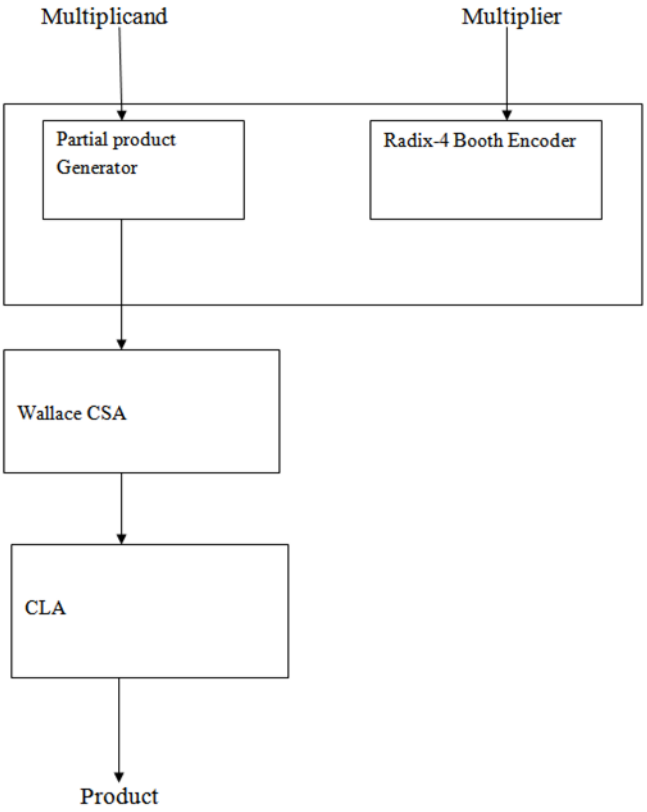

Fig 3.6. Block diagram of Booth Encoder Multiplier for signed multiplier

RESULTS AND DISCUSSION

Simulation results

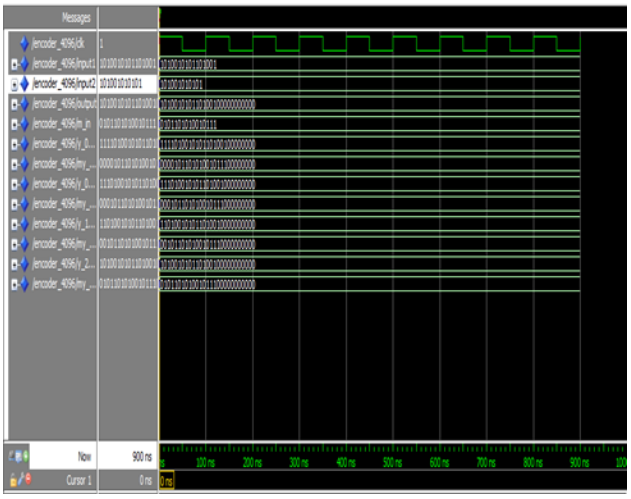

Figure 4.1Simulation result for radix-1024encoding

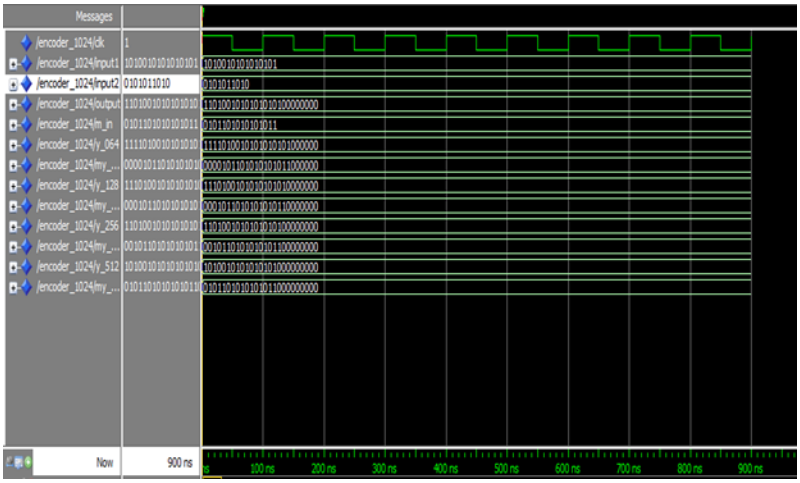

Figure 4.2 Simulation result for radix-4096 encoding

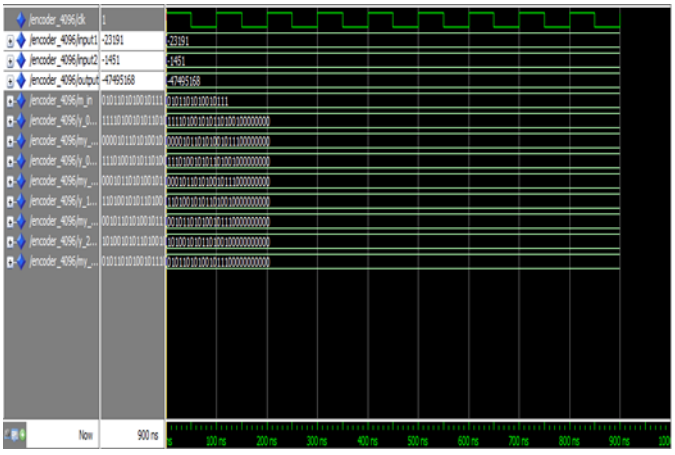

Figure 4.3.Simulation result for radix-4096 encoder using decimal value

In the radix encoder, the approximate multipliers (RAD1024 and RAD4096) are simulated. The results concern the radix encoding, the partial product generation and decimal value are shown in fig 4.1-4.3.

CONCLUSION

The approximate hybrid high radix encoding is generated in the partial products of a signed multiplier. The MSBs of the multiplicand are encoded with the accurate radix-4 encoding, while its k LSBs are encoded with an approximate high radix-2^k encoding, with k being a configuration parameter that adjusts the tradeoff between accuracy and energy consumption. The error of the technique follows a Gaussian distribution with near zero average. Compared with state-of-the-art inexact multipliers, the radix encoder constitute better approximate design alternative, outperforming them in both energy consumption and accuracy. Furthermore, the efficiency of the multipliers is applied in real-life applications. Finally, the technique is scalable, delivering higher energy savings for the same error, as the multiplier's size increases.

REFERENCES

[1]. W. Liu, L. Qian, C. Wang, H. Jiang, J. Han, and F. Lombardi, "Design of approximate radix-4 booth multipliers for error-tolerant computing," IEEE Trans. Comput., 66(8), 2017, 1435–1441.

[2]. R. Zendegani, M. Kamal, M. Bahadori, A. Afzali-Kusha, "RoBa multiplier: A rounding-based approximate multiplier for high-speed yet energy-efficient digital signal processing," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., 25(2), 2017, 393–401.

[3]. H. Jiang, J. Han, F. Qiao, and F. Lombardi, "Approximate radix-8 booth multipliers for low-power and high-performance operation," IEEE Trans. Comput., 65(8), 2016, 2638–2644.

[4]. Zervakis, K. Tsoumanis, S. Xydis, D. Soudris, "Design-efficient approximate multiplication circuits through partial product perforation," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., 24(10), 2016, 3105–3117.

[5]. S. Hashemi, R. I. Bahar, and S. Reda, "DRUM: A dynamic range unbiased multiplier for approximate applications," IEEE/ACM Int. Conf. Comput.-Aided Design, 2015, 418–425.

[6]. S. Narayanamoorthy, H. A. Moghaddam, Z. Liu, "Energy-efficient approximate multiplication for digital signal process-ing and classification applications," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., 23(5), 2015, 1180–1184.

[7]. G. Zervakis, S. Xydis, K. Tsoumanis, D. Soudris, "Hybrid approximate multiplier architectures for improved power-accuracy trade-offs," in Proc. IEEE/ACM Int. Symp. Low Power Elec-tron. Design, 2015, 79–84.

[8]. C. Liu, J. Han, and F. Lombardi, "A low-power, high-performance approximate multiplier with configurable partial error recovery," in Proc. Design, Autom. Test Eur., 2014, 1–4.

[9]. V. K. Chippa, S. T. Chakradhar, K. Roy, "Analysis and characterization of inherent application resilience for approximate computing," IEEE Design, 2013, 1–9.

[10]. J. Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate and probabilistic adders," IEEE Trans. Comput., 62(9), 2013, 1760–1771.