

DELAY COMPARISON OF RADIX-8 MULTIPLIER USING MAC UNIT

Kishan Ramesh; Nithin Bidare puttaraju

Dept. of Electronics & Communication

Global Academy of Technology (GAT)

Bengaluru, India

E-mail: kishanramesh06@gmail.com;nithinb23@gmail.com

ABSTRACT

This paper mainly focuses on designing of 64-bit signed multiplication using radix-8 based multiplier. A new multiplier design is proposed which reduces the number of partial products. The multiplier has been used with Carry Save Adder (CSA), Carry Select Adder (CSLA) and Kogge-Stone Adder (KSA) to implement Multiplier Accumulator Unit (MAC) and the parameter propagation delay is measured and compared

Keywords Radix-8, Kogge-Stone Adder, Carry Save Adder, Carry Select Adder, Partial products

I. INTRODUCTION

In modern computing, especially in many high-speed Digital Signal Processing, image processing, multimedia applications, the multiplier plays a very important role as it determines the performance in accuracy and speed of all these systems ultimately. Multiplier is the non-memory sub-block of the microcontroller with the largest size and delay that has a big impact on the cycle time. High throughput multiplier-and- accumulator is constantly an important factor to attain elevated performance digital signal of instantaneous applications of signal processing. Multiplier-and-accumulator is essential element of digital signal as well as image/audio processing system for instance filtering, convolution as well as inner products consequently high speed is critical to expand for actual processing applications.

In the conventional multiplier, the number of partial products to be added are determined by the number of bits the multiplier or multiplicand being used. Bigger the number of bits the multiplicand or the multiplier contain, the longer time it takes to produce the product. Delay of the multiplier is determined largely by the number of partial products to be added.

Standard advances in technology approved to devise multipliers which are speed and has constancy in design appropriate for VLSI performance. Motivating point in usage of radix-8 recording is less number of transistors ensuring in condensed power dissipation in addition to area size evaluated to radix-4. Radix-8 structural design has a slighter number of partial products and consequently lower power expenditure.

Radix-8 recording permit a time gain in partial products summing up but is not functional since we have to make one of multiples of the multiplicand by means of a high-speed adder. As the number of bits increase more number of partial products are generated, addition of these partial products using a normal adder increases the delay hence to reduce the delay a specially designed Carry Save Adder (CSA) is used to add the partial product.

II. AN OVERVIEW OF MULTIPLIER-AND-ACCUMULATOR

Rising demands of processing high speed data signal encouraged researchers to look for fastest processors. The multiplier as well as multiplier-and-accumulator are building blocks of processor and encompass a great impact on processor speed.

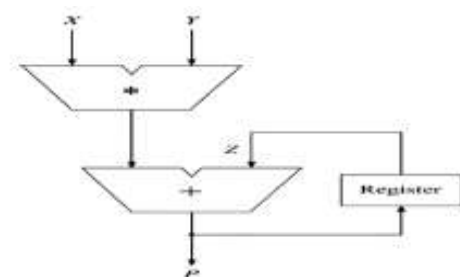


Fig 1 Overview of MAC unit

High throughput multiplier-and-accumulator is constantly an important factor to attain elevated performance digital signal

processing applications in support of instantaneous applications of signal processing [5]. Since multiplier necessitates highest delay among fundamental operation in digital system, significant path is limited by multiplier. Multiplier essentially consists of three functioning steps such as Booth Encoder, Partial product reduction network as well as final adder. Multiplier-and-accumulator is essential element of digital signal as well as image/audio processing system for instance filtering, convolution as well as inner products consequently high speed is critical to expand for actual processing applications. Numerous researchers have attempted in scheming MAC for elevated computational performance as well as low power consumption. For high speed multiplication, Modified Booth Algorithm is most normally used, in which partial product is produced from

Multiplicand(X) as well as Multiplier (Y). Booth multiplication permit for minor, quicker multiplication circuits all the way through encoding signed bits to 2's complement which is moreover criterion system in chip design and make available substantial development by reducing partial products.

III. BACKGROUND STUDY

Various approaches have been proposed in literature for designing a multiplier aiming to reduce the hardware and to improve to the performance. There were studies done in

[1] On relative performance comparison of Radix-based Booth Encoding multiplier. Multipliers included in the comparison are Radix-2 Booth Encoding multiplier, Radix-4 Booth Encoding multiplier, Radix-8 Booth Encoding multiplier, Radix-16 Booth Encoding multiplier and Radix-32 Booth Encoding multiplier.

[2] A new multiplier design is proposed which reduces the number of partial products by 25%. This multiplier has been used with different adders available in literature to implement multiplier accumulator (MAC) unit and parameters such as propagation delay, power consumed and area occupied have been compared.

[3] Optimum Booth integer multiplier for low power applications. Booth radix-4, radix- 8 and radix-16 were compared for area, speed and power using standard cell ASIC design flow and 28nm CMOS technology.

[4] Design of a low error fixed-width radix-8 Booth multiplier which produces an n-bit product with two n-bit inputs. The truncation of the 2n product bits to n bits is achieved by removing about half the adder cells that are required to add the partial products. However, in order to keep the truncation error to a minimum, error compensation biases are obtained and applied to the inputs of the retained adder cells.

[5] A select probability estimation bias (SPEB) for fixed-width Booth multipliers. Considering the special condition in Booth encoder, the selection mode is included to compensate truncation error in fixed-width Booth multiplier design. The probability estimation method is adopted in the general condition. Compared to the existing works, the proposed SPEB 16×16 fixed-width Booth multiplier achieves higher accuracy with only a little area overhead.

[6] Operations involved in conventional carry select adder (CSLA) and binary to excess-1 converter (BEC)-based CSLA are analyzed to study the data dependence and to identify redundant logic operations. In the proposed scheme, the carry select (CS) operation is scheduled before the calculation of final-sum, which is different from the conventional approach. An efficient CSLA design is obtained using optimized logic units. The proposed CSLA design involves significantly less area and delay than the recently proposed BEC-based CSLA. Due to the small carry-output delay, the proposed CSLA design is a good candidate for square-root (SQRT) CSLA.

IV. RADIX-8 BOOTH ENCODING

Multiplication can be completed in 3 steps:

1. Generation of partial products.
2. Accumulation of partial products.
3. Addition of these partial products.

First step can be completed with the help of Modified Booth's Algorithm (Radix-8). Radix-8 booth's algorithm generates N/3 partial product rows.

After applying Radix-8 booth's algorithm on 64*64 bit multiplier we have N/3 partial products (N= number of bits). Second and third

step can be completed by accumulating and adding the generated partial product.

Radix-8 Booth Multiplier recoding applies the same algorithm as any other radix based Multiplier, but here we take quartets. Consequently, a multiplier based on this radix-8 scheme generates fewer partial products than radix-4 and radix-2 multipliers, but the computation of each partial product is more complex. In particular, a partial product corresponding to an encoding $x=\pm 1, 2, 3$ requires the computation, and therefore a full addition.

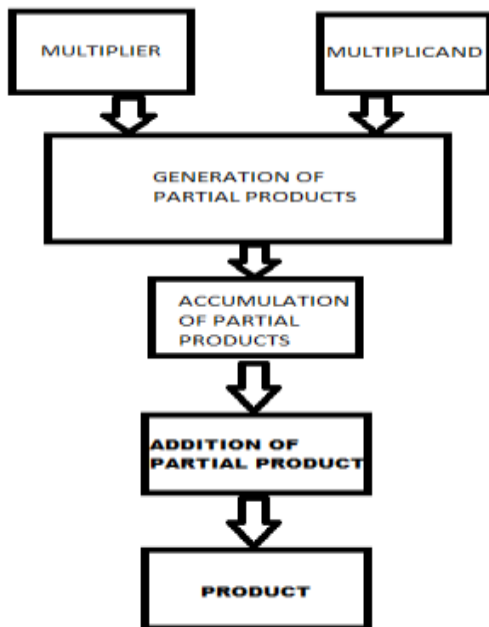


Fig 2 Flowchart of radix-8 multiplier

Table 1 Encoding Table

PP bits of b	Partial Products
0 0 0 0	0a
0 0 0 1	+1a
0 0 1 0	+1a
0 0 1 1	+2a
0 1 0 0	+2a
0 1 0 1	+3a
0 1 1 0	+3a
0 1 1 1	+4a
1 0 0 0	-4a

1 0 0 1	-3a
1 0 1 0	-3a
1 0 1 1	-2a
1 1 0 0	-2a
1 1 0 1	-1a
1 1 1 0	-1a
1 1 1 1	0a

In order to obtain +2a shift once, +3a shift and add multiplicand once, for negative bits do the same process and perform 2's compliment.

The generated partial product after encoding the multiplier is then accumulated and added using KSA/CSA/CSLA.

V. PROPOSED MODEL

In the proposed model the multiplication is carried out using the radix-8 encoding method, to increase the speed of multiplication process the generated partial product during this operation is given to different adders, here in this paper the generated partial product is given to Kogge-Stone Adder (KSA) Carry Select Adder(CSLA) and Carry Save Adder (CSA) and the parameters propagation delay is measured and compared.

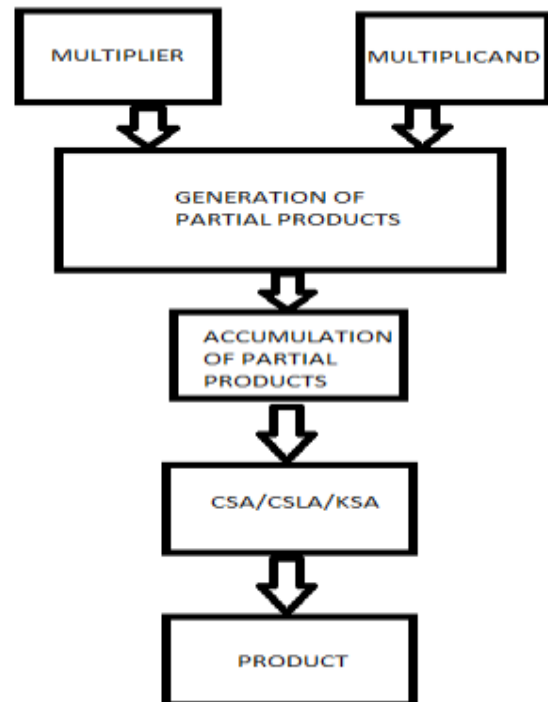


Fig 3 Flowchart of proposed radix-8 multiplier

1. CARRY SAVE ADDER

A carry save adder is a type of digital adder, used in computer microarchitecture to compute the sum of three or more n -bit numbers in binary.

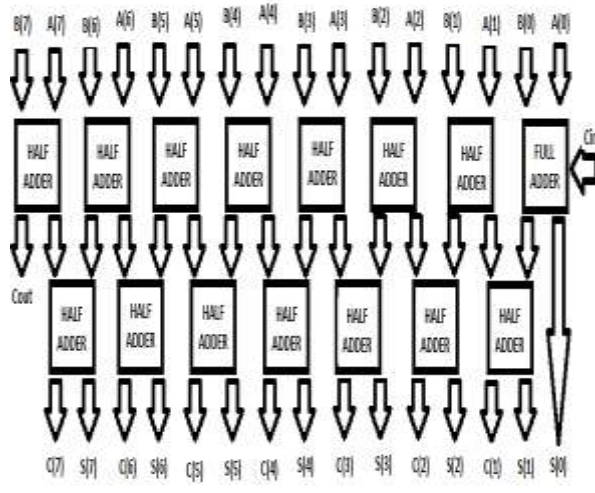


Fig 4 Block diagram of Carry Save Adder

It differs from other digital adders in that it outputs two numbers of the same dimensions as the inputs, one which is a sequence of partial sum bits and another which is a sequence of carry bits.

A carry look-ahead adder can reduce the delay. In principle the delay can be reduced so that it is proportional to $\log n$, but for large numbers this is no longer the case, because even when carry look-ahead is implemented, the distances that signals have to travel on the chip increase in proportion to n , and propagation delays increase at the same rate.

The CSA tree however propagates the carries obtained in all the first stages immediately as inputs to the next consecutive adders in the second stage without waiting for entire column to be computed at a time. As a result of it next columns need not wait for carries until all stages in previous columns are computed. Similarly carries obtained in all second stages are propagated as inputs to next successive adder's stage.

In CSA at the output we obtain the sum and carry separately. So in order to obtain the final product we give the output of CSA to carry look-ahead adder.

2. CARRY SELECT ADDER

In electronics, a carry-select adder is a particular way to implement an adder, which is a

logic element that computes the $(n+1)$ -bit sum of two n -bit numbers. The carry-select adder is simple but rather fast, having a gate level depth of $O(n-\sqrt{n})$. The carry-select adder generally consists of two ripple carry adders and a multiplexer. Adding two n -bit numbers with a carry-select adder is done with two adders (therefore two ripple carry adders) in order to perform the calculation twice, one time with the assumption of the carry being zero and the other assuming one. After the two results are calculated, the correct sum, as well as the correct carry, is then selected with the multiplexer once the correct carry is known.

The number of bits in each carry select block can be uniform, or variable. In the uniform case, the optimal delay occurs for a block size of $\lfloor n-\sqrt{n} \rfloor$. When variable, the block size should have a delay, from addition.

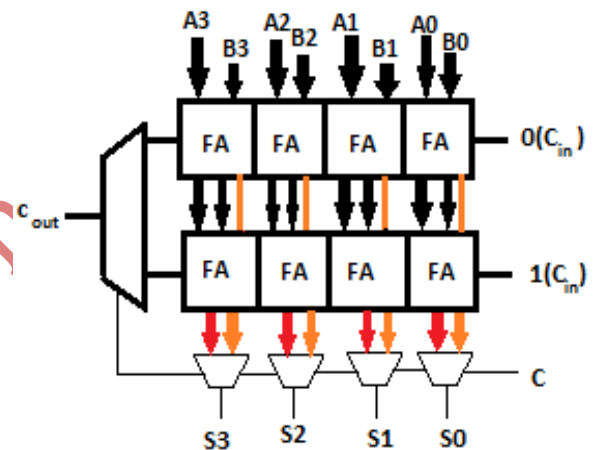


Fig 5 Block diagram of Carry Select Adder

3. KOGGE-STONE ADDER

KSA is a parallel prefix form carry look ahead adder. It generates carry in $O(\log n)$ time and is widely considered as the fastest adder and is widely used in the industry for high performance arithmetic circuits. In KSA, carries are computed fast by computing them in parallel at the cost of increased area.

The complete functioning of KSA can be easily comprehended by analyzing it in terms of three distinct parts:

1. Pre-processing.
2. Carry look ahead network.
3. Post processing.

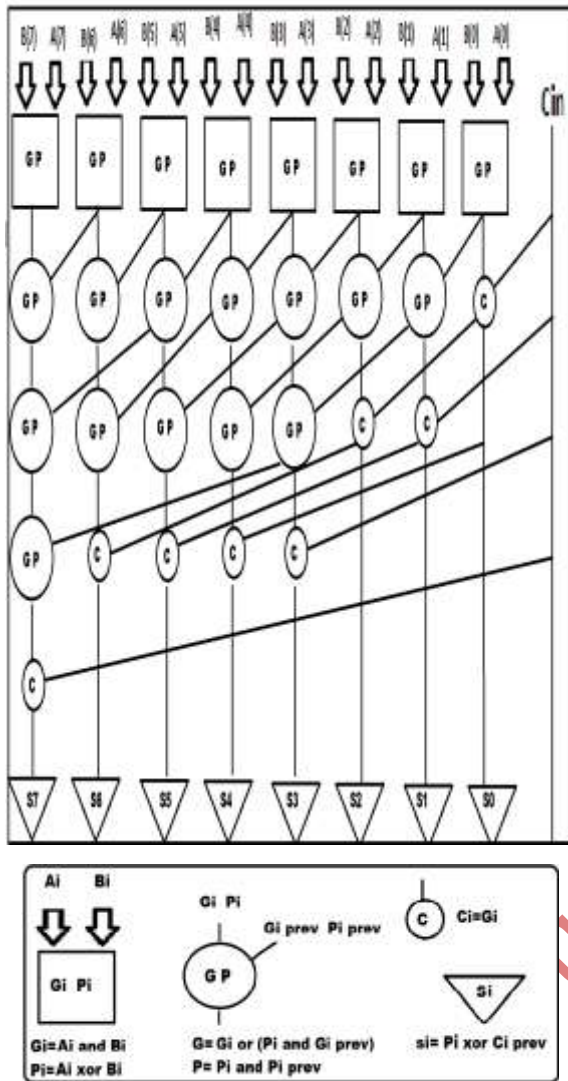


Fig 6 8-bit kogge-stone adder block diagram

Pre-processing involves computation of generate and propagate signals corresponding to each pair of bits in A and B. These signals are given by the logic equations

$$P_i = A_i \text{ xor } B_i$$

$$G_i = A_i \text{ and } B_i$$

Carry look ahead network differentiates KSA from other adders and is the main force behind its high performance. This step involves computation of carries corresponding to each bit.

It uses group propagate and generate as intermediate signals which are given by the logic equations

$$P = P_i \text{ and } P_i \text{ prev}$$

$$G = G_i \text{ or } (P_i \text{ and } G_i \text{ prev})$$

Post processing is the final step and is common to all adders of this family (carry look ahead). It involves computation of sum bits. Sum bits are computed by the logic equation

$$S_i = P_i \text{ xor } C_{\text{prev}}$$

VI. DELAY COMPARISON USING KOGGE-STONE ADDER AND CARRY SAVE ADDER

Table 2 Delay comparison

DESIGN	PROPAGATION DELAY
Radix-8 Booth Multiplier Using Kogge-Stone Adder	73.27 ns
Radix-8 Booth Multiplier Using Carry Save Adder	77.56 ns
Radix-8 Booth Multiplier Using Carry Select Adder	79.28ns

The delay of Radix-8 Booth multiplier using Kogge-Stone Adder, Carry Select Adder and Carry Save Adder. Here Radix-8 Booth Multiplier using Kogge-stone Adder has less delay compared to Radix-8 Booth Multiplier using Carry Save Adder and Radix-8 Booth Multiplier using Carry Select Adder.

VII. CONCLUSION AND FUTURE SCOPE

The 64 bit radix-8 multiplier is designed by using Carry Save Adder, Carry select adder and also by using Kogge-Stone adder to add the partial products and delay performance is compared for multiplier with Carry Save Adder, carry select adder and multiplier with Kogge-Stone Adder. Among these three designs multiplier using Kogge-Stone Adder is having less propagation delay compared to multiplier using Carry Save Adder and Carry Select Adder.

The propagation delay can be further reduced by using different combination of adders and also work can be done to reduce the area occupied and power dissipation.

REFERENCES

- [1] Kelly Liew Suet Swee, and Lo Hai Hiung, Department of Electrical and Electronics Engineering Universiti Teknologi PETRONAS, "Performance Comparison Review of Radix-Based Multiplier Designs" 2011.
- [2] Deepak S, Binsu J Kailath, Indian Institute of Information Technology Design and Manufacturing (IIITD&M) kancheepuram, Chennai, India, "Optimized MAC Unit Design " 2012.
- [3] HANI h. Saleh, Baker S. Mohammad, ECE department Khalifa University of Science, Technology and research Abu Dhabi, UAE and Earl E. Swartzlander, Jr ECE department, university of Texas at Austin, "The Optimum Booth Radix for Low Power Integer Multipliers", 2013.
- [4] Saroja S Bhusare, Department of ECE, JSS Academy of Technical Education Bengaluru, India and V.S Kanchana Bhaaskaran ,School of Electronics Engineering VIT University Chennai, India, "Design of Low-Error Fixed-width Radix-8 booth Multiplier",2014.
- [5] Wen-Quan He, Chieh-Yaang Liu, Wei-Yi Kiu and Yuan-Ho Chen, Department of information & computer Engineering, Chung Yuan Christian University, "A High Accuracy Fixed-Width booth multiplier using Probability Estimation Bias",2014
- [6] Basant Kumar Mohanty and Sujit Kumar Patel, "Area-Delay-Power Carry-Select Adder", IEEE transactions on circuits and systems-ii: express brief, vol.60,june 2014