IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 8, Issue 4, Ver. I (Jul. - Aug. 2018), PP 51-57
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

A Fast Multiplier Using Modified Radix-4 Booth Algorithm With Redundant Binary Adder For Low Energy Applications

R.Mallikarjuna Sharma, K.Raju
PG Scholar, G.Pulla Reddy Engineering College (Autonomous), Kurnool, AP, India,
Asst Prof, G.Pulla Reddy Engineering College (Autonomous), Kurnool, AP, India.
Corresponding Author: R.Mallikarjuna Sharma

***Abstract**: The fundamentalobjective of this paper is to execute a multiplier for rapid and low energy applications. Multipliers are the building blocks of elite frameworks like FIR filters, computerized flag processors, and so on in which speed is the commanding variable. There are numerous multiplier structures created to build the speed of polynomial math. Stall calculation is the best calculation utilized for quick exhibitions. This works by presenting a superior multiplier utilizing Modified Radix-4 corner calculation with Redundant Binary Adder to get rapid. A relative investigation of various corner calculations as far as power utilization, delay, zone, vitality and vitality defer item is additionally talked about in this work. Every one of the circuits are reproduced in the Cadence reproduction device utilizing 180nm innovation. The test results demonstrate that the proposed stall multiplier indicates rapid, low vitality and low vitality postpone item contrasted with the current corner multipliers*

Date of Submission: 18-08-2018 Date of acceptance: 03-09-2018

I. Introduction

A binary multiplier performs multiplication of two binary numbers. The execution of numerous computational issues is regularly commanded by the speed at which an increase task can be executed. The least complex approach to perform augmentation is to utilize a solitary two info snake. For N inputs that are M and N bits wide, the augmentation takes M cycles utilizing N bit snake. This move and include calculation for duplication includes M halfway items. The fractional items are created by duplicating multiplicand with a touch of the multiplier which is an AND task and by moving the outcome based on the multiplier bit position. The commanding parameter for increase is speed.The delay of the multiplication operation depends upon the number of partial products to be used. The number of partial products to be added is determined by the number of bits that the multiplier or multiplicand have used. In order to get fast performance, one of the methods is to arrange the circuit to generate all the partial products parallel and organizes in an array. The second and widely used algorithm which leads to faster performance is booth algorithm .

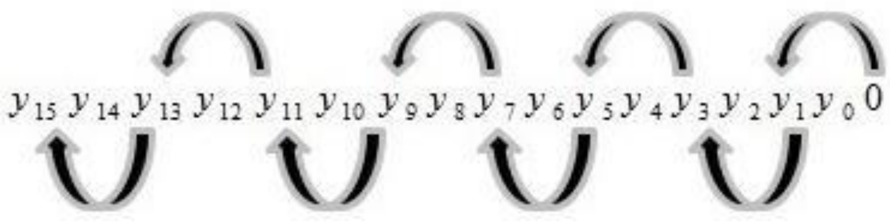
The stall duplication calculation is a calculation that increases two marked double numbers [3]. The calculation was developed by Andrew Donald Booth in 1950.The first algo-rithm is the Radix-2 algorithm[9]. In Radix-2 calculation first will annex zero to the multiplier and gathering the multiplier such that each gathering comprises of 2 bits. With the goal that first match comprises of annexed zero and lsb of multiplier bits. What's more, the following pair is the covering of the first match in which msb of the first combine will be the lsb of the second combine. With the goal that the aggregate n fractional items will get. It require an expansive number of full adders and there is no decrease of incomplete items. Next calculation is the modified Radix-2 calculation which additionally called Radix-4 calculation. In this calculation gathering of multiplier is done such that each gathering comprises of 3 bits. In the wake of creating all the incomplete items stretch out the join to the item size[4]. Since it requires huge number of full adders in view of the sign expansion which thusly causes high deferral. In Radix-8 calculation the gathering will be 4 bits. On account of Radix-8 calculation, it requires complex encoder circuit[7]. The proposed multiplier takes care of the above issues..

II. Conventional Radix-4 Algorithm

1) Assign a zero to the left of the multiplier.
2) The multiplier is divided into groups of 3 bits, MSB and LSB of each group is overlapped which is shown in figure 1.
3) Each group is encoded to select a single partial product according to the selection table.
4) In the next step, partial product is shifted by two bit positions with respect to its neighbors. In this way, the total number of partial products for 16 bit multiplication has been 8 .

5) Extend the sign of each partial product upto the product size.
6) Finally will be the addition of all partial products

The grouping of the multiplier bit is shown in figure
1 Each gathering comprises of 3 bits. These bits will bolstered to the encoder which gives encoded flag 0,1,- 1,2, - 2. Along these lines, the aggregate number of halfway items for 16 bit increase has been decreased from 16 to 8. The lessened halfway item will be n/2 incomplete items, where n is the aggregate operand length. The diverse required products can be acquired by a basic move of the multiplicand. Negative products, in 2's supplement frame, can be assessed utilizing an a little bit at a time supplement of the individual positive numerous, with a 1 included at the LSB position of the fractional product[2]. The table 1 shows the encoded signal and corresponding partial products.


$y_{15}\ y_{14}\ y_{13}\ y_{12}\ y_{11}\ y_{10}\ y_9\ y_8\ y_7\ y_6\ y_5\ y_4\ y_3\ y_2\ y_1\ y_0\ 0$

In the wake of creating all the incomplete items final step will be the expansion. For a 16 bit duplication halfway item generator produces 16 bit incomplete item at that point will stretch out the join to 32 bit position. The speck documentation of the expansion of every single halfway item is shown in figure 2

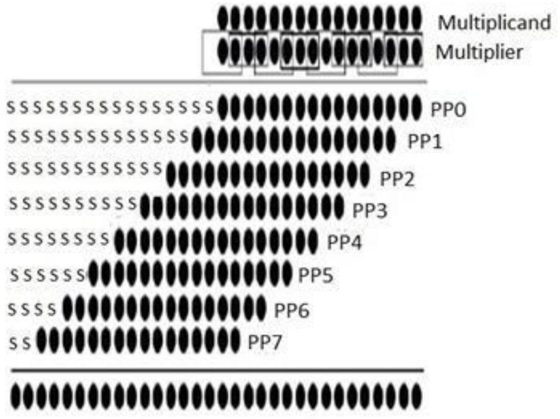


Figure 2. Radix-4 dot notation

Proposed Multiplier

The proposed multiplier utilizes Modified Radix-4 calculation and after the age of halfway items the expansion is performed utilizing Redundant Binary Adder.

A. Modified Radix-4 algorithm
1) Append zero to one side and 2 zeros to one side, which is appeared in figure 3.
2) Group the multiplier bits such that each gathering comprises of 3 bits.
3) The first gather contains the added zero and the slightest 2 significant bits.
4) Apply this gathering to the stall encoder and encoder will deliver the incomplete item as indicated by the table 2.
5) Group the rest of the bits that are appeared in figure 3.
6) After creating the fractional item expand the sign piece of every incomplete item up to the twentieth position.
7) Add every single halfway item.

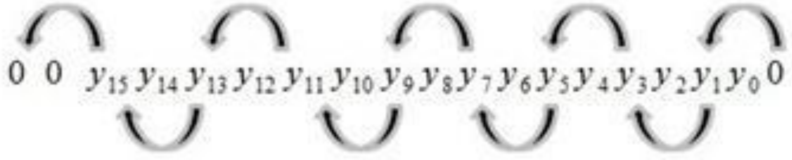

$0\ 0\ y_{15}\ y_{14}\ y_{13}\ y_{12}\ y_{11}\ y_{10}\ y_9\ y_8\ y_7\ y_6\ y_5\ y_4\ y_3\ y_2\ y_1\ y_0\ 0$

Figure 3. Grouping of the multiplier bits

Figure 3 demonstrates the gathering of multiplier bits.For a 16 bit operand, Modified Radix-4 calculation produces 9 halfway items which is half. These gatherings will apply to the stall encoder and it produces flag. That encoded flag will bolstered to the fractional item generator and creates incomplete items. The gathering and the comparing halfway item is appeared in table 2. In the wake of creating all the fractional items final step will be the addition.The speck documentation of the expansion of every single incomplete item is shown in figure 4.

The ordinary Radix-4 calculation produces 8 fractional items for 16 bit increase and Modified Radix-4 calculation produces add up to 9 halfway items. What's more, the Modified Radix-4 calculation lessens the aggregate number of full adders required in light of the fact that it stretches out the sign just up to twentieth piece position. In any case, on account of ordinary calculation the sign expansion is up to 32 bit position which causes countless adders which thusly causes high postponement and power utilization. With the goal that the Modified Radix-4 multiplier lessens the quantity of full snake required and the equipment cost.After creating all the halfway products,the expansion is finished utilizing Redundant Binary Adder

A. Redundant Binary Adder

Redundant Binary Adder is a fast design which can be utilized for rapid multipliers[2]. The design comprises of one to change over from ordinary double to excess paired frame and the circuit for the expansion of two repetitive numbers. An excess paired snake is utilized for the expansion of 9 fractional items. The square outline of multipliers is demonstrated as follows.

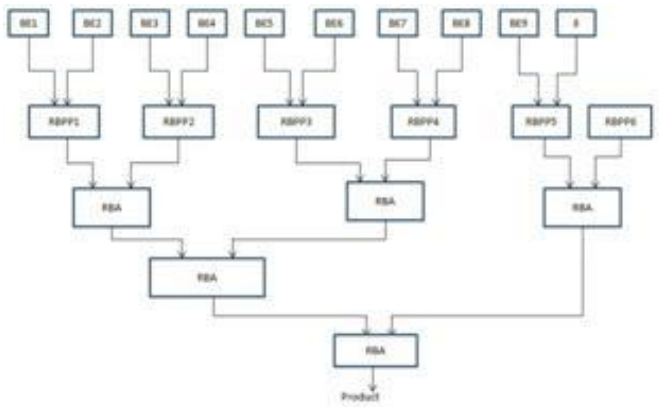


Figure 5. RBA logic block diagram

1st stage is the transformation of ordinary twofold fractional item to excess double halfway product.ConsiderAn and B as the typical paired incomplete items.

A+B=A-(-B)

-B can be written as $\bar{B}$ +1 So the equation 1 becomes A+B=A-$\bar{B}$-1

It is rewritten asA+B= (A, $\bar{B}$) + (0,1) This is shown in Table 3.Thus the redundant binary partial product(RBPP) is equal to the sum of two NB partial products, is generated by inverting one of the 2 partial products and adding (0,1) to the lowest digit.

a_i	b_i	RB digit	Value
0	0	0	0
0	1	$\bar{1}$	-1
1	0	1	1
1	1	1	0

For each bit addition adder produces intermediate sum and carry. That sum will add to the preceding carry and produce the sum. Intermediate sum and carry for each bit is shown in Table 4

Augend (ai)	Addend (bi)	Digits at the next lower order position	Intermediate carry	Intermediate sum
1	1	Any value	1	0
1	0	Both are non-negative	$\bar{1}$	$\bar{1}$
0	1	Otherwise	0	1
0	0	Any value	0	0
1	$\bar{1}$	Any value	0	0
$\bar{1}$	1	Any value	0	0
0	$\bar{1}$	Both are non-negative	0	$\bar{1}$
$\bar{1}$	0	Otherwise	$\bar{1}$	1
$\bar{1}$	$\bar{1}$	Any value	$\bar{1}$	0

Table IV COMPUTATION RULE

since it as of now changes over (1,1) to (0,0). The moderate whole and going before convey don't get 1 or - 1 at the same time as per the table The information set takes one of the qualities (0,0), (0,1), (1,0) and no (1,1) esteem due to the thought of indication of past sets. Consequently RBA maintains a strategic distance from nonstop convey spread. With the goal that the deferral is incredibly diminished.

Simulation Results And Discussion

A. Modified Radix-4 multiplier

The proposed multiplier yield waveform is appeared in figure 6.The two data sources 771 and 29 are encouraged to the circuit and the circuit gives the yield 22359.In the second case inputs are 771 and 45085 got yield 34760535.

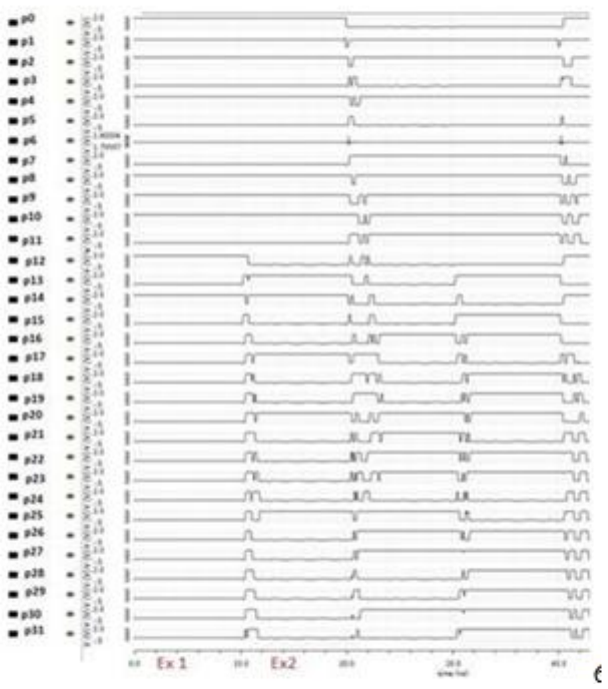


Figure 6. Multiplier output

The circuit and the waveform of corner encoder is appeared in figure 7 and 8. The contribution of stall encoder is the bits of the gathering. Relies on the bits, it produces yield M or 2M and s is the indication of three bits.If the bits is 000 then its yield M and 2M will be zero and the sign will be 0. On the off chance that the bits are 001 then M progresses toward becoming 1 and 2M moves toward becoming 0. In like manner it delivers the yield. The yield flag of an encoder will be the contributions of the halfway item generator

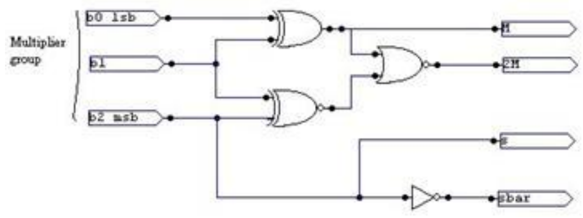


Figure 7. Booth Encoder

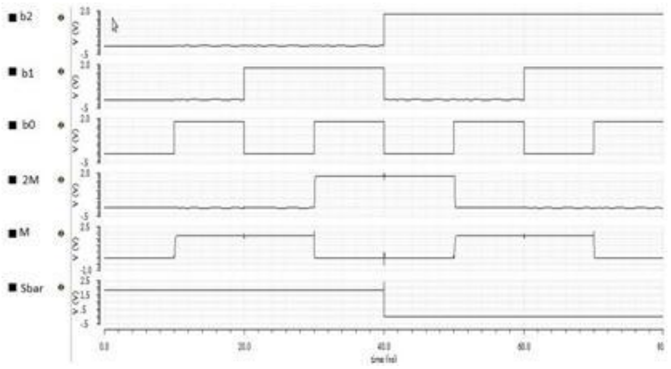


Figure 8. Booth Encoder waveform

The circuit of a fractional item generator is appeared in figure 9.Figure 9 demonstrates one example.Consider the mul-tiplicand whose esteem is 6664910 and its parallel number is 1000000100010110012.According to the encoded flag incomplete item generator produces 9 halfway items.

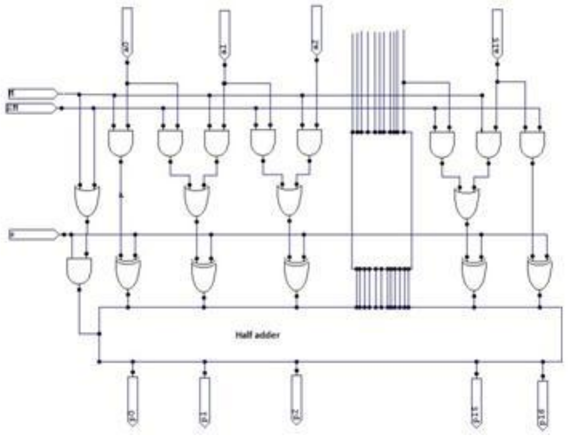


Figure 9. Partial product generator

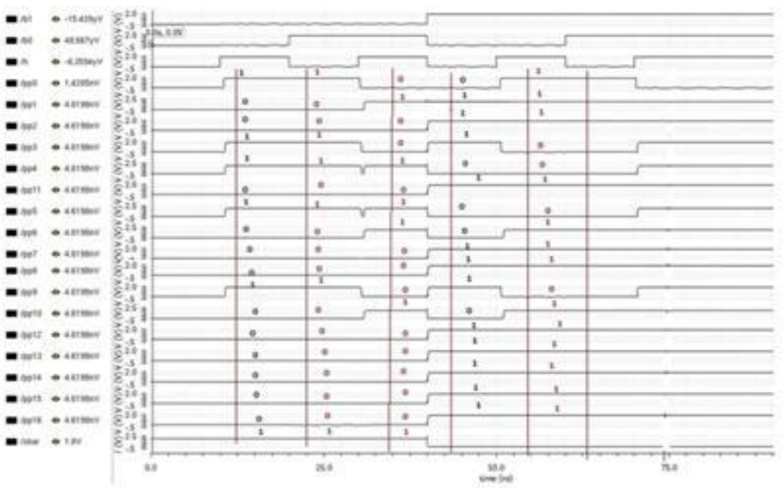


Figure 10. Partial product generator waveform

B. Redundant Binary Adder

The circuit and waveform of RBA is shown in figure 11 and 12.

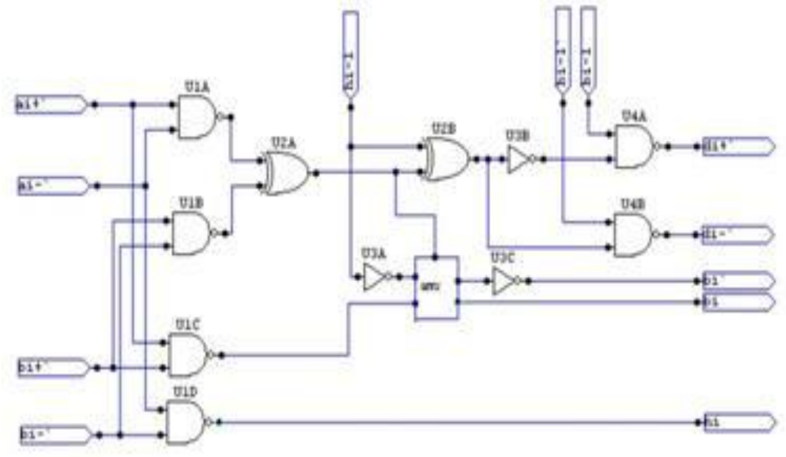


Figure 11. Redundant Binary Adder

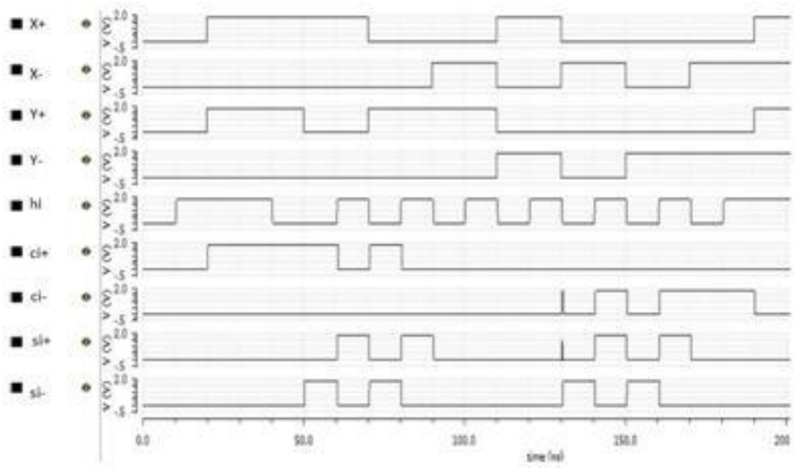


Figure 12. Redundant binary adder waveform

The waveform demonstrates the transitional whole and convey for each piece. Furthermore, z flag gives the indication of the former combine of two info bits.Multiplier circuits utilizing Radix-2,Radix-4,Modified Radix-4 are actualized. Examinations depend on the execution parameters, for example, transistor check, control utilization, delay,energy and vitality defer item. Reenactments were done utilizing Cadence Virtuoso instrument in 180nm innovation

Conclusion

A fast multiplier utilizing Modified Radix-4 Booth algorithm is Implemented.The proposed design utilizes Redundant Binary Adder which results rapid exhibitions and low vitality. The proposed engineering has been recreated in Cadence Virtuoso apparatus. Likewise executed Radix-2 ,Radix-4 calculation and contrasted them and modi-fied Radix-4 calculation. As indicated by the execution eval-uation results, it has been demonstrated that the Modified Radix-4 calculation has the best execution as far as speed,energy,power and vitality postpone item. The proposed engineering utilized repetitive double viper for the expansion which additionally enhanced the speed. As indicated by the investigation the proposed circuit is prepared to do quick execution and furthermore decreases Energy while contrasted with different structures

References

[1]. Kumre, Laxmi, Ajay Somkuwar, and Ganga Agnihotri. ”Imple- mentation of radix 4 booth multiplier using MGDI technique.” Emerging Research Areas and 2013 International Conference on Microelectronics, Communications and Renewable Energy (AICERA/ICMiCR), 2013 Annual International Conference on. IEEE, 2013.

[2]. Jose, Bijoy, and DamuRadhakrishnan. ”Fast redundant binary partial product generators for Booth multiplication.” Circuits and Systems, 2007. MWSCAS 2007. 50th Midwest Sympo- sium on. IEEE, 2007.

[3]. Rao, Pachara V., C. Prasanna Raj P, and S. Ravi. ”Vlsi design and analysis of multipliers for low power.” Intelligent Information Hiding and Multimedia Signal Processing, 2009. IIH-MSP’09. Fifth International Conference on. IEEE, 2009.

[4]. Cho, Ki-seon, et al. "54x54-bit radix-4 multiplier based on modified booth algorithm." Proceedings of the 13th ACM Great Lakes symposium on VLSI. ACM, 2003.
[5]. Venkatraman, S., et al. "Low Power Area Ecient Multiplier Us- ing Shannon Based Multiplexing Logic." International Journal of Embedded Systems Applications 2.2 (2012).
[6]. C.Senthilpari, Ajay Kumar Singh and K.Diwakar "Design of a low power, high performance, 8x8 bit multiplier using a Shannonbased adder cell." Microelectronics Journal 39 (2008) 812821.
[7]. Rajput, Ravindra P., and MN ShanmukhaSwamy. "High speed Modified Booth Encoder multiplier for signed and unsigned numbers." Computer Modelling and Simulation (UKSim), 2012 UKSim 14th International Conference on. IEEE, 2012.
[8]. Kaur, Lakhvinder, and Parminder Singh Jassal. "Synthesis And Simulation Of 8x8-Bit Modified Booth's Multiplier." signal processing 1: 2.
[9]. N. H E Weste,Principle of CMOS VLSI design, Adision- Wesley 1998.
[10]. J.P.Uyemura,Fundamental of MOS Digital Integrated Circuits, Reading Addison -Wesley, 1996pp.

R.Mallikarjuna Sharma "A Fast Multiplier Using Modified Radix-4 Booth Algorithm With Redundant Binary Adder For Low Energy Applications "IOSR Journal of VLSI and Signal Processing (IOSR-JVSP) , vol. 8, no. 4, 2018, pp. 51-57