design and implementation of approximate booth multiplier using different 4:2 compressors

G.V. Spandana, J. Manideepika, B. Usha Sri , M. Divya, B. Lakshmi,

student, student, student, student, Assistant professor

Abstract - Presently, the design of a multiplier is playing a vital role in the stream of VLSI signal processing, DSP, Modern wireless communication etc. In order to speed up the processing operation and optimize the performance of the system, some high-performance approximate multipliers with reduced area, power and delay are required. An area, power and delay efficient approximate booth multiplier is designed by using booth algorithm in this study. The proposed approximate booth multiplier is designed by using 4:2 compressor by ensuring better performance than the previous existing booth multiplier. Tabulated the comparison between different 4:2 compressors in the proposed design. Based on the area, power and delay results of different compressors, a new approximate booth multiplier design is implemented with best compressor on FPGA Kit.

Index Terms - Booth Algorithm, Booth Multiplier, Approximate Multiplier, 4:2 Compressor and FPGA kit.

I.INTRODUCTION

The multiplier design is mostly classified into two types which is signed and unsigned multiplier. In the signed multiplier it will perform both positive and negative multiplication. But in the unsigned multiplier is used to imply only positive number of multiplication. For example, Array multiplier, Wallace multiplier, parallel multiplier etc. are unsigned multiplier. From that booth multiplier is the one among the signed multiplication scheme .A Booth multiplier consists of three parts: partial product generation using a Booth encoder, partial product accumulation using compressors and final product generation using a fast adder. The Booth encoding has been proposed for improving the performance of multiplication of two's complement binary numbers and it has been further improved by the 4:2 compressor. Therefore, multiplier designs are mainly focused on high-speed, low area and low power. These parameters are achieved by approximate multipliers. Generally, approximate computing has a significant attention as a rising strategy to decrease power consumption of error tolerant applications like image processing. Booth Radix-4 algorithm can reduce the number of partial products that must be calculated. By implementing this algorithm in higher radix, Approximate Booth Algorithm can reduce the number of calculations in exchange to the design complexity. For digital circuit implementation, radix-4 is considered the best trade-off between speed and complexity because it is able to halve the number of partial products needed to be calculated while only adding a bit-shift function as an additional operation. The Booth Radix-4 algorithm reduces the number of partial products by half while keeping the circuit's complexity down to a minimum. This results in lower power operation in an FPGA.The Radix-4 Booth Recoding is simply a multiplexer that selects the correct shift-and-add operation based on the groupings of bits found in the product register. The product register holds the multiplier. The multiplicand and the two's complement of the multiplicand are added based on the recoding value.

(i) Radix-4 Booth Multiplier

Block	Partial product (operation)
000	0
001	+1*multiplicand
010	+1*multiplicand
011	+2*multiplicand
100	-2*multiplicand
101	-1*multiplicand
110	-1*multiplicand
111	0

Figure (1)

(ii) Booth Algorithm

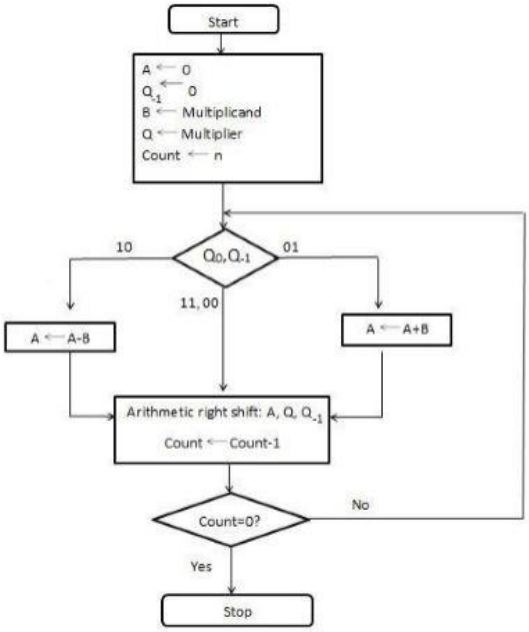

Figure (2)

Step 1: Load the initial values for registers

A = 0 (Accumulator), Qres = 0, M = Multiplicand, Q = Multiplier

Step 2: Check the value of {Q0,Qres}. If 00 or 11, go to step 5. If 01, go to step 3. If 10, go to step 4.

Step 3: Perform A = A + M. Go to step 5.

Step 4: Perform A = A - M.

Step 5: Perform Arithmetic Shift Right of {A, Q, Qres} and decrement count.

Step 6: Check if counter value n is zero. If yes, go to next step. Else, go to step 2.

Step 7: Stop

(iii)Booth Algorithm Example

Steps	A		Q	Q res	Operation
	0000		0100	0	Initial
1	0000		0010	0	Right shift
2	0000		0001	0	Right shift
3	0101		0001	0	A=A-B
	0010		1000	1	Right shift
4	1101		1000	1	A=A+B
	1110		1100	0	Right shift

Figure (3)

II.DESIGN AND IMPLEMENTATION OF EXISTING BOOTH MULTIPLIER

The design of existing booth multiplier is done by using half adders and full adders.

1.full adder

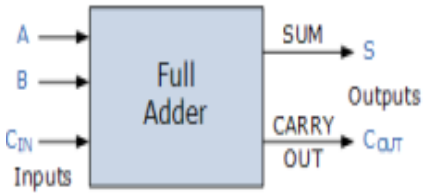

Figure (4)

Full Adder is the adder that adds three inputs and produces two outputs. The first two inputs are A and B and the third input is an input carry as C-IN. The output carry is designated as C-OUT and the normal output is designated as S which is SUM.

$$Sum = A \oplus B \oplus C_{IN}$$

$$C_{out} = A\,B + B\,C_{in} + A\,C_{in}$$

2.half adder

Figure (5)

$$S = \overline{A} \cdot B + A \cdot \overline{B}$$

$$C = A \cdot B$$

A half adder is a digital logic circuit that performs binary addition of two single-bit binary numbers. It has two inputs, A and B, and two outputs, SUM, and CARRY.

3.architecture of existing design

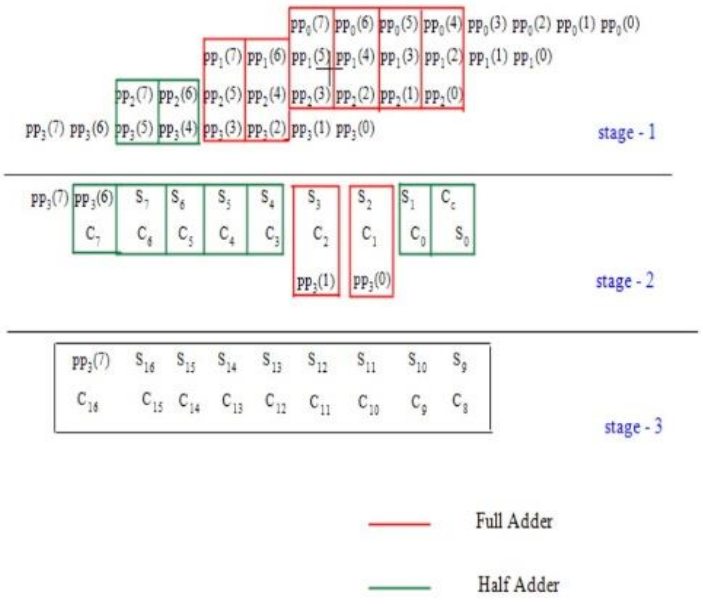

Figure (6)

Here in stage 1 taking 4 8-bit partial products as p1, p2,p3,p4 ,for every partial product it is shifting the register with 2 bits and assuming the partial product which consists of three is considered as full adder and the partial product which consists of two partial products considered as half adders. The remaining partial products are propagated into stage 2.

In stage 2 also they are segregated as full adders and half adders and remaining propagated into stage 3.

In stage 3 sum and carry are the results of the existing design.

III. Design and implementation of proposed booth Multipliers

(a)exact 4:2 compressor

A compressor is a device which is mostly used in multipliers to reduce the operands while adding terms of partial products. A typical M-N compressor takes M equally weighted input bits and produces N-bit binary number

The compressor is a useful element that is widely used in VLSI circuits and systems. It is generally used as a processing element. In this work, a 4-2 compressor has been designed with XOR-XNOR module and multiplexer module. The XOR-XNOR module is consists of six transistors and the multiplexer is consists of transmission gate.

Truth table of 4:2 Compressor

n^*	C_{in}	C_{out}	Carry	Sum
0	0	0	0	0
1	0	0	0	1
2	0	1	0	0
3	0	1	0	1
4	0	1	1	0
0	1	0	0	1
1	1	0	1	0
2	1	1	0	1
3	1	1	1	0
4	1	1	1	1

Figure (7)

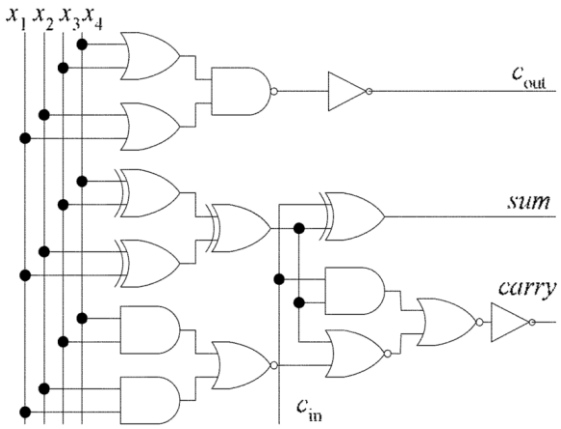

Figure (8)

here by designing exact 4:2 compressor got report for the area, power and delay. area is 60 LUTs, power is 0.32W and delay is 13.43ns.

(b)approximate 4:2compressor-1

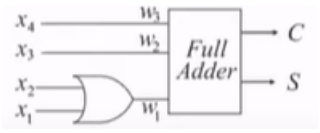

Figure (9)

Here by designing approximate 4:2 compressor got results for the compressor (1) are area is 60 LUTs, power is 0.321W and delay is 10.2ns.

(c)approximate 4:2compressor-2

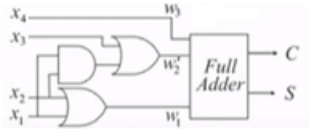

Figure (10)

Here by designing approximate 4:2 compressor got results for the compressor (2) are area is 60LUTs, power is 0.321W and delay is 10.20ns.

(d)approximate 4:2compressor-3

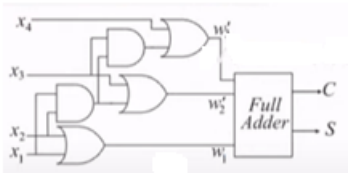

Figure (11)

Here by designing approximate 4:2 compressor got results for the compressor (3) are area is 60LUTs, power is 0.321W and delay is 10.203ns.

(e)approximate 4:2compressor-4

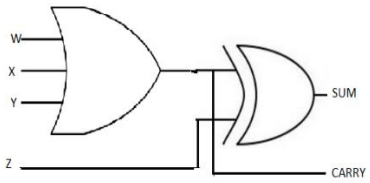

Figure (12)

Here by designing approximate 4:2 compressor got results for the compressor (4) are area is 44LUTs, power is 0.014W and delay is 13ns

Truth table of approximate 4:2 compressor

INPUTS			EXACT OUTPUTS		APPROXIMATE OUTPUTS		ABSOLUTE DIFFERENCE
X1	X2	X3	CARRY	SUM	CARRY	SUM	
0	0	0	0	0	0✓	0✓	0
0	0	1	0	1	0✓	1✓	0
0	1	0	0	1	0✓	1✓	0
0	1	1	1	0	1✓	0✓	0
1	0	0	0	1	0✓	1✓	0
1	0	1	1	0	1✓	0✓	0
1	1	0	1	0	0×	1×	1
1	1	1	1	1	1✓	0×	1

Figure (13)

(f) Architecture for proposed system

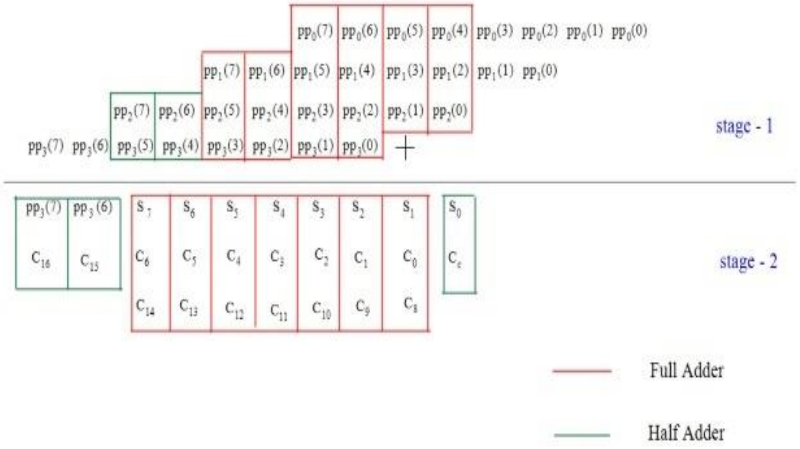

Figure (14)

As of existing design here in proposed design also taking 8 bit 4 partial products and shifting them by two steps for each partial product and segregating them as full adder, half adder and compressor, then remaining partial product is propagated into the next stage in which sum, carry and partial product are segregated as full adders and half adders.

IV. Simulation results

(o)simulation result of basic booth algorithm

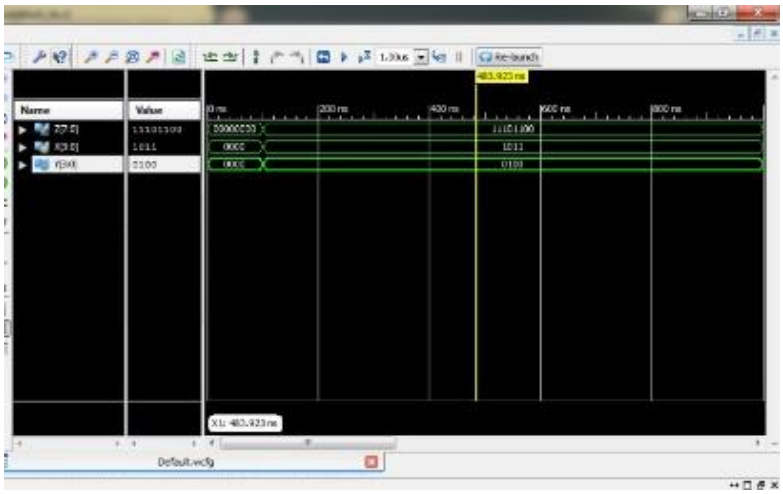

Figure (15)

(p) simulation results of existing design

Here by designing the existing design got the results for area 49LUTs, power is 0.036W and delay is 14ns

waveforms:

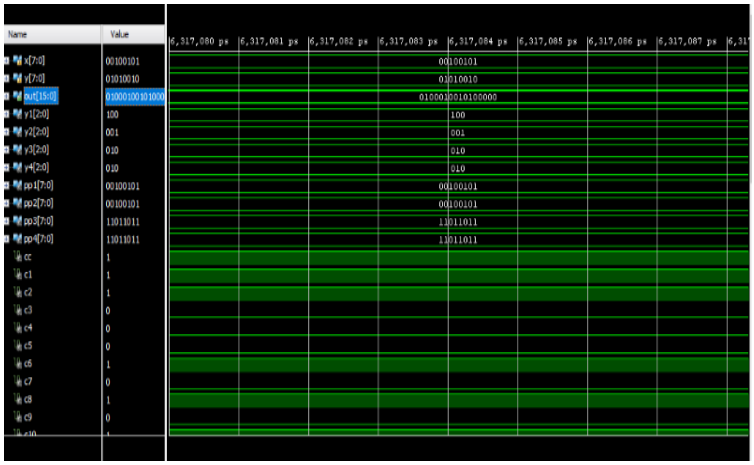

Figure (16)

(q)simulation result of proposed design

Here by designing approximate 4:2 compressor got results for the compressor (4) are area is 44LUTs, power is 0.014W and delay is 9ns.

After comparing with different compressors this gives the best result.

waveforms:

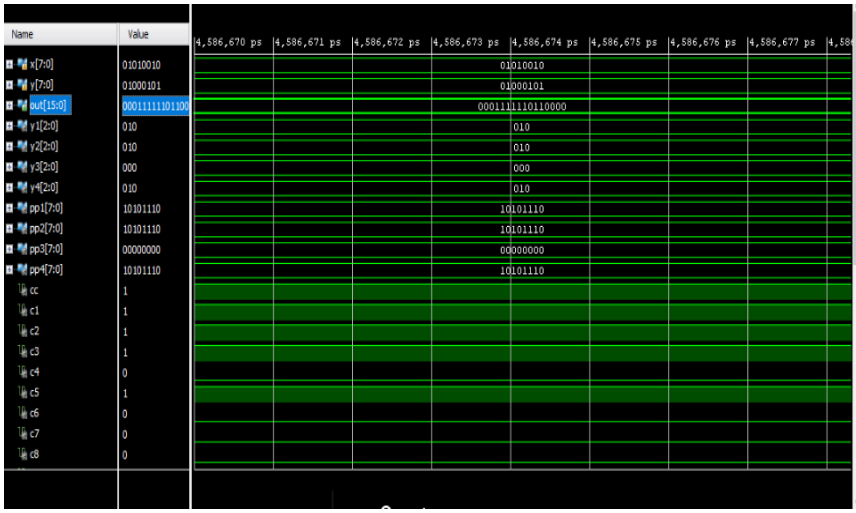

Figure (17)

V. COMPARISONS

(r) comparison results of different 4:2 compressors

COMPRESSORS	AREA	POWER	DELAY
Exact compressor	60LUT	0.321	13.43
Compreesor-1	60LUT	0.321W	10.203ns
Compressor-2	60LUT	0.321W	10.203ns
Compressor-3	60LUT	0.321W	10.214ns
Compressor-4	44LUT	0.014W	9.8ns

(s) comparison between existing and proposed design

SYSTEM	AREA	POWER	DELAY
EXISTING	49LUT	0.036W	14ns
PROPOSED	44LUT	0.014W	9.8ns

VI. CONCLUSION:

By using booth algorithm implemented the basic booth multiplier in order to reduce the partial product by half that means the reduction of area to check the working of booth algorithm. Focused on the previous design implementation in order to show the comparison results of existing model and proposed model. Implemented the previous design and got the area report, power report, delay report and got the simulation results. Designed the new system with different approximate 4:2 compressors and recorded the area, power report, delay report and simulation results. Implemented the new system with best 4:2 compressor. Have shown the comparison between existing model and proposed model.

VI. REFERENCES:

[1] Zainab Aizaz and Kavita Khare "Area and Power Efficient Truncated Booth Multipliers using Approximate Carry-Based Error Compensation"-IEEE Truncations on Circuits and Systems (Published on 2 February 2022)

[2] Haroon Waris, Chenghua Wang, Weiqiang Liu : "Hybrid Partial Product-Based High-Performance Approximate Recursive Multipliers"-IEEE Truncations on Emerging Topics in Computing (Published on 4 March 2022)

[3] Mohammad Ahmadaniejad,Mohammad Hossein Moaiyeri : "Energy and Quality Efficient Approximate Multipliers for Neural Network and Image Processing Applications"-IEEE Truncations on Emerging Topics in Computing (published on 7 June 2022)

[4] Sithara Raveendran , Nithin Kumar , Vasantha: "Inexact Signed Wallance Tree Multiplier Design Using Reversible Logic"-IEEE Access (published on 9 August 2021)

[5] Haoran Pei,Hang Zhou,Yajuan He: "Design Of Ultra Low Power Consumption Approximate 4:2 Compressors Based On Compensation Characteristics"-IEEE Truncations on Circuits and Systems (published on 1 january 2021)

[6] Fang Yi Gu,Ing Chao Lin: "A Low Power And High Accuracy Approximate Multiplier With Reconfigurable Truncation"-IEEE Access (published on 13 June 2022)

[7] Bhavya Lahari Gundapaneni, JRK Kumar Dabbakuti. "Booth Algorithm for the Design of Multiplier. " International Journal of Innovative Technology and Exploring Engineering(IJITEE)ISSN:2278-3075, Volume-8 Issue (Published on 7, May, 2020)

[8] Bhavya Lahari Gundapaneni, JRK Kumar Dabbakuti. "Booth Algorithm for the Design of Multiplier. " International Journal of Innovative Technology and Exploring Engineering (IJITEE)ISSN: 2278-3075, Volume-8 Issue (Published on 7, May, 2020)