

An Design of Radix-4 Modified Booth Encoded Multiplier and Optimised Carry Select Adder Design for Efficient Area and Delay

1. K. Nivetha,
PG Scholar, Dept of ECE,
Nandha Engineering College,
Erode.

2. Ms. S. Nandhini,
Assistant Professor, Dept of ECE,
Nandha Engineering College,
Erode.

3. E. Sharmila,
PG Scholar, Dept of ECE,
Nandha Engineering College,
Erode.

Abstract— To design an efficient Carry Select Adder, the logic operations involved in Conventional CSLA and Binary to Excess-1 Converter based CSLA are analyzed to study the data dependence and to identify redundant logic operations. The main objective of this project is to eliminate all the redundant logic operations present in the conventional CSLA and to propose a new logic formulation for CSLA and to develop an low power optimised booth encoded multiplier. The structure of CSLA is such that there is further scope of reducing the area, delay and power consumption. Finally to implement the proposed CSLA in multiplier design in order to prove the proposed design is efficient. In the existing designs, logic is optimized without giving any consideration to the data dependence. To overcome this problem, the proposed logic formulations for the CSLA is based on the optimized carry generator and carry selection design and to remove all redundant logic operations and sequence logic operations based on their data dependence. The proposed SQR-4 CSLA design involves significantly less Area and consumes less energy than the existing CSLA design on average bit-widths. At last the implementation of proposed CSLA block in booth encoded multiplier design reduces delay and consumes less power. The simulation of this project is carried out by using Tanner EDA v13.0

Keywords – Carry select adder, Arithmetic unit, Modified booth encoded multiplier, Low power design.

I. INTRODUCTION

Recent trends in electronics and communication systems make extensive use of Digital Signal Processing (DSP) providing custom accelerators for the real-time application like audio and video signal processing with large capacity data processing are increasingly being demanded. The adder and multiplier are the essential elements of the digital signal processing such as filtering, convolution, and inner products. A conventional carry select adder (CSLA) is an RCA–RCA configuration that generates a pair of output bits corresponding to the predicted input-carry and selects one out of each pair for final sum and carry [3]. A conventional CSLA produce less CPD than an RCA, but the design is not attractive due the dual structure of RCA. Kim and Kim used one RCA and one add-one circuit is implemented using a multiplexer (MUX) instead of two RCA [4]. He et al. proposed a square-root SQR-4 CSLA [7] to implement large bit-width adders with less delay. The main

objective of SQR-4 CSLA design is to provide a parallel path for carry propagation that helps to reduce the overall adder delay and avoid critical path used in single CSLA structure. Ram kumar and Kittur optional a method binary to BEC-based CSLA which involves less logic resources than the conventional CSLA, but it has slightly higher delay than existing methods [6]. A CSLA based on common Boolean logic (CBL) is also proposed in [7] and [8]. The CBL-based CSLA of [7] involves significantly less logic resource than the conventional CSLA but it has longer carry propagation delay. The CBL-based SQR-4 CSLA design of [8] requires more logic resource and delay. We observe that logic optimization largely depends on accessibility of redundant operations in the formulation, whereas adder delay mainly depends on data dependence. In the existing method, logic operation is formulated without giving any deliberation on data dependence. The analysis has been made on logic operations involved in existing CSLAs to analyse the data dependence and to categorize disused logic operations. Based on this analysis the new logic formulation for the CSLA is proposed. In general, a multiplier uses Booth's algorithm [9] and linear structure of full adders (FAs), or Wallace tree [10] instead of parallel adder, i.e., this multiplier mainly consists of the three parts: Booth encoder like Wallace tree to compress the partial product and final adder [11]. In the architecture proposed in [12], the critical path was reduced by eliminating the adder for accumulation and decreasing the number of input bits in the final adder. Whose performance is better than the previous MAC architectures and critical path is reduced, and output rate is also improved. The main intention of this project is to design an efficient Modified Booth encoded Radix-4 multiplier design based on proposed CSLA block to improve the performance of complex DSP applications.

II. PROPOSED CARRY SELECT ADDER DESIGN

The proposed adder block has two units: (i) the sum and carry generator unit (SCG) and (ii) the sum and carry selection unit. The SCG unit consumes most of the logic resources of CSLA and extensively contribute to the delay in propagation path. The main theme is to remove all redundant logic operations and sequence logic operations based on their data dependence to improve the performance.

A. The new logic operation involved in Proposed CSLA design

The logic operation involves a considerable amount of logic resources for calculating sum and carry corresponding to its input carry. Instead, one can select the required carry word from the anticipated carry words either cin as 1 or 0 to calculate the last sum. The particular carry word is added with the half-sum to generate the final-sum output. Using this method, one can have three design advantages: 1) Calculation of S at 0 is avoided in the SCG unit; 2) the n-bit select unit is required instead of the (n + 1) bit; and 3) small output-carry delay. All these features result in an optimised design of area and delay for the CSLA. Finally all the redundant logic operations are removed. The proposed logic formulation for the CSLA is given as

$$s0(i) = A(i) \oplus B(i) \quad c0(i) = A(i) \cdot B(i) \quad (4a)$$

$$C01(i) = c01(i-1) \cdot s0(i) + c0(i) \text{ for } c01(0) = 0 \quad (4b)$$

$$C11(i) = c11(i-1) \cdot s0(i) + c0(i) \text{ for } c11(0) = 1 \quad (4c)$$

$$c(i) = c01(i) \text{ if } (cin = 0) \quad (4d)$$

$$c(i) = c11(i) \text{ if } (cin = 1) \quad (4e)$$

$$cout = c(n-1) \quad (4f)$$

$$s(0) = s0(0) \oplus cin \quad s(i) = s0(i) \oplus c(i-1) \quad (4g)$$

B. The optimised structure of proposed CSLA design

The proposed CSLA structure is developed based on the logic formulation given in (4a)–(4g), and it is shown in Fig.1. It consists of HSG unit, FSG unit, CG unit, and CS unit. The CG unit is composed of two parts CG₀ and CG₁ corresponding to the input carry ‘0’ and ‘1’. The HSG receives two n-bit operands A and B to generate half-sum word s₀ and half-carry word c₀ of width n bits each. Both CG₀ and CG₁ receive s₀ and c₀ from the HSG unit and generate two n-bit full-carry words c⁰₁ and c¹₁ corresponding to the input carry ‘0’ and ‘1’, respectively. The CS unit selects any one of final carry from the two carry words available at its input line using the control signal input carry like mux operation. It selects c⁰₁ when input carry = 0; otherwise, it selects c¹₁. The CS unit operates like an 2-to-1 MUX. However, we find from the truth table of the CS unit that carry words c⁰₁ and c¹₁ follow a specific bit pattern. If c⁰₁(i) = ‘1’, then c¹₁(i) = 1, respective to both s₀(i) and c₀(i), for 0 ≤ i ≤ n - 1.

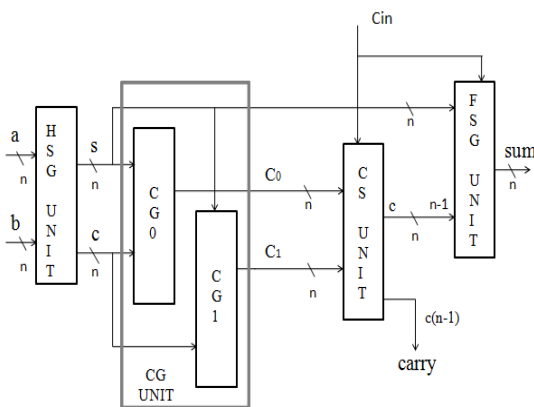


Fig.1. Proposed carry select adder design, where n represents the input operand bit-width.

This characteristic is used for logic optimization of the CS unit. The final output carry is obtained from the CS unit. The MSB of c is input to output c_{out}, and its (n - 1) LSB is XOR with (n - 1) MSB of half-sum (s₀) in the FSG to obtain (n - 1) MSB of final-sum (s). The LSB of s₀ is XORED with c_{in} to obtain the LSB of s.

C. Area-Delay Estimation Method

The main advantage of these design is all gates are made up of 2-input AND, 2-input OR, and inverter (AOI). A 2-input XOR is composed of 2 AND, 1 OR, and 2 NOT gates. The area and delay of the 2-input AND, 2-input OR, and NOT gates as shown in Table I.

Table I

Design	AND Gate	OR Gate	NOT Gate
Area	7.37	7.37	6.45
Delay	180	170	100

The area and delay estimation of a design are planned using the following relations.

$$A = a \cdot N_a + r \cdot N_o + i \cdot N_i \quad (5a)$$

$$T = n_a \cdot T_a + n_o \cdot T_o + n_i \cdot T_i \quad (5b)$$

where (N_a, N_o, N_i) and (n_a, n_o, n_i), respectively, represent the (AND, OR, NOT) gate counts of the total design and its critical path. (a, r, i) and (T_a, T_o, T_i), respectively, represent the area and delay of one (AND, OR, NOT) gate Using (5a) and (5b), the area and delay of each design are calculated.

C. Proposed 16-bit Multistage CSLA (SQRT-CSLA) design

The multipath carry propagation feature of the CSLA is fully exploited in the SQRT-CSLA [8], which is composed of a chain of CSLAs to provide parallel propagation of input to all adder block. CSLAs of increasing size are used in the path of carry propagation. By implementing the SQRT-CSLA design, large-size adders are implement with radically less delay than a single-stage CSLA of similar size. On the other hand, carry propagation delay between the CSLA stages of SQRT-CSLA is critical on the whole of adder delay. Due to the production of output-carry with multipath carry propagation feature, the proposed CSLA design is more favourable than the existing CSLA designs for area-delay efficient implementation of SQRT-CSLA. A 16-bit SQRT-CSLA design using the proposed CSLA is shown in Fig. 2, where the 2-bit RCA, 2-bit CSLA, 3-bit CSLA, 4-bit CSLA, and 5-bit CSLA are used.

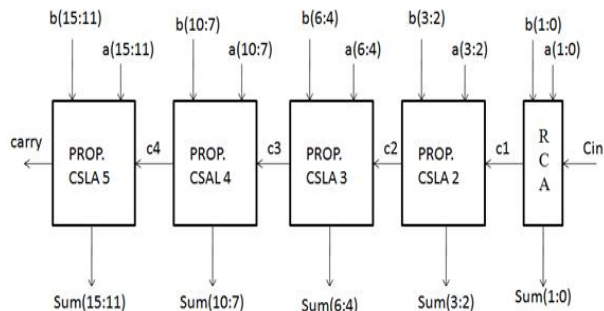


Fig.2. Proposed SQRT-CSLA for $n = 16$. All intermediate and output signals are labelled with delay (shown in square brackets).

III. BOOTH ENCODED MULTIPLIER DESIGN

A. Modified Booth encoding scheme

Modified Booth (MB) is a prevalent form used in multiplication [13] based on Radix-4. It is a redundant signed-digit radix-4 encoding technique. The modified Booth encoding (MBE) scheme is known as the most efficient Booth encoding and decoding scheme. The multiplication of X and Y input terms are done by using the modified Booth are shown in Fig.3 and Fig.4. The algorithm starts from grouping Y by three bits and encoding into one of $\{-2, -1, 0, 1, 2\}$. Table I shows the rules to generate the encoded signals by MBE scheme. Its main advantage is that it reduces by half the number of partial products in multiplication comparing to any other radix-2 representation. The most significant digit term is formed based on sign extension of the initial 2's complement number.

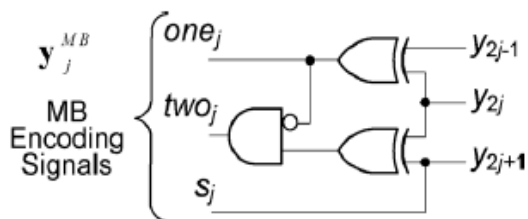


Fig.3. Schematic representation of booth encoding signal.

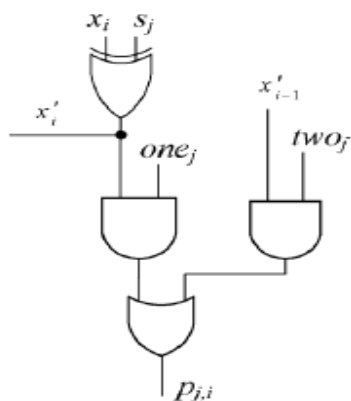


Fig.4. Generation of i th-bit partial product using booth decoding scheme.

Table II

Binary			y_j^{MB}	MB Encoding			Input Carry
y_{2j+1}	y_{2j}	y_{2j-1}		$sign=s_j$	$\times 1=one_j$	$\times 2=two_j$	$c_{in,j}$
0	0	0	0	0	0	0	0
0	0	1	+1	0	1	0	0
0	1	0	+1	0	1	0	0
0	1	1	+2	0	0	1	0
1	0	0	-2	1	0	1	1
1	0	1	-1	1	1	0	1
1	1	0	-1	1	1	0	1
1	1	1	0	1	0	0	0

B. Implementation of CSLA adder block in an multiplier design

The modified Booth algorithm reduces the number of partial products by half in the first step. The design of two different array multipliers are presented, one by using Carry Select Adder using conventional logic for addition of partial product terms and another by introducing Carry Select Adder using proposed logic in partial product lines. The comparison is done on the basis of three performance parameters i.e. total Area, delay, and power consumption. The modified booth multiplier is based on Radix-4 booth algorithm followed by proposed 2-bit proposed carry select adder design.

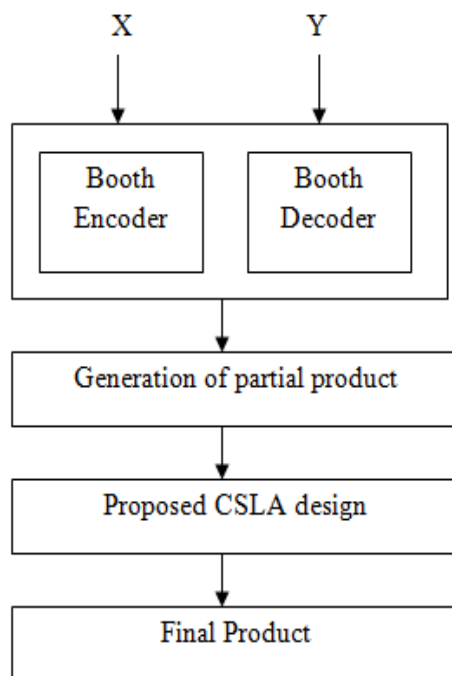


Fig.5. Implementation of CSLA block in Booth encoded multiplier

In case of multiplier using adder as CSLA block, all partial product accompaniments as well as final addition is carried out by using carry select logic. Fig.5 shows the schematic of a 2 bit CSLA based array multiplier. In Array multiplier, almost identical cells array is used for generation of the bit-products. All bit-products are generated in array of sequential basis and collected through an array of adders. Booth multiplier has habitual that simplifies the wiring and the layout structure.

IV. RESULTS AND ANALYSIS

The Sqrt-CSLA is coded in VHDL using these proposed CSLA design and existing CSLA design of 16 bit width is simulated in ModelSim platform and analysis is done by using Xilinx ISE compiler.

A. Simulation result of proposed Sqrt-CSLA design

The simulation output of 16-bit Proposed Sqrt-CSLA shown in Fig.6 below is generated by assigning the input values and produces corresponding output values.

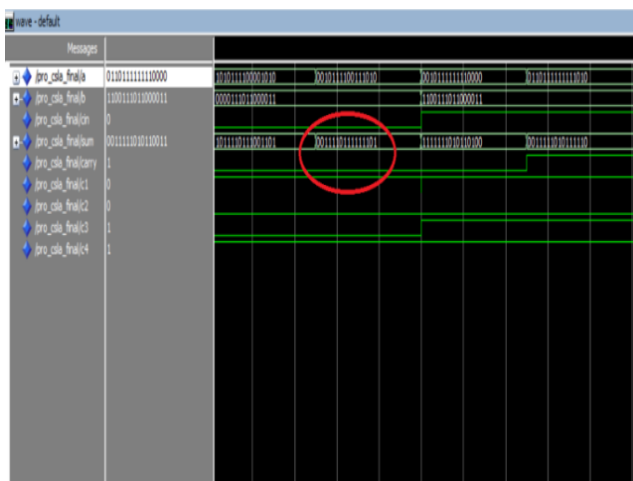


Fig.6. Output waveform of 16-bit Proposed Sqrt-CSLA

The synthesis result in terms of Area, Delay and Power of Existing and proposed 16-bit Sqrt-CSLA design was analyzed using Xilinx simulator and are shown in Table III

Table III
Comparison table of existing and proposed 16-bit Sqrt-CSLA

S.No	Design	Conventional Sqrt-CSLA	BEC based Sqrt-CSLA	Proposed Sqrt-CSLA
1	Area (Gate count)	348	465	258
2	Delay (ns)	28.281	32.259	25.209
3	Power (mW)	97	95	72
4	ADP (ns)	9841.7	15000.4	6503.9
5	PDP (pWs)	2743.3	3064.6	1815.1

B. Simulation result of proposed Modified Booth multiplier design

The simulation output of a multiplier using proposed CSLA shown in Fig.7 below is generated by assigning the input values and produces corresponding output values is done by using tanner tool.

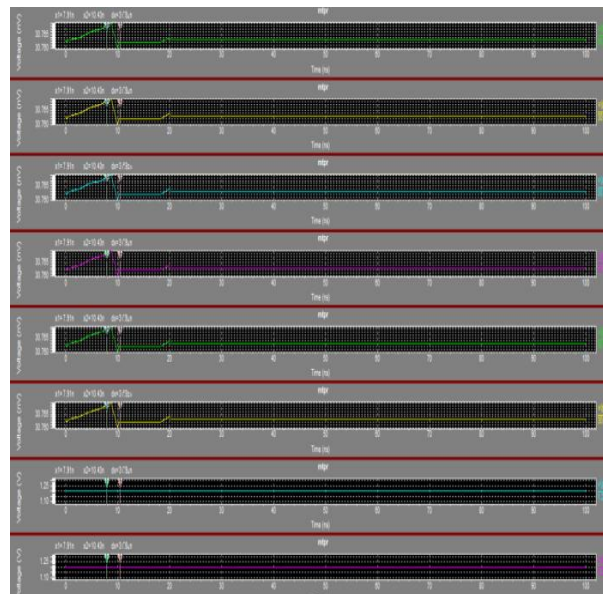


Fig.7. Output waveform of proposed Booth encoded multiplier using proposed CSLA.

The comparison result in terms of Area, Delay and Power of Existing and proposed CSLA based Booth encoded multiplier design was analyzed using Tanner simulator and are shown in Table IV

TABLE IV

Comparison table of existing and proposed Booth encoded multiplier design

S.No	Design	Existing Booth multiplier	Proposed Booth multiplier
1	Area (MOSFET count)	989	1416
2	Delay (ns)	4.03	2.69
3	Power (W)	2.822e^-004	6.335e^-007

C. Comparison chart

ADP and PDP comparison for existing and proposed 16-bit Sqrt-CSLA in terms of Area-Delay-Product (ADP) and Power-Delay-Product (PDP).

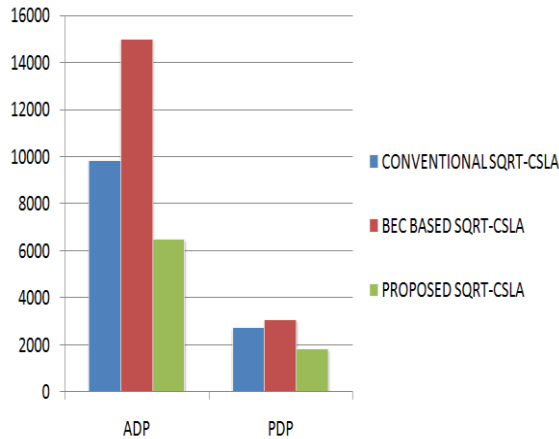


Fig.8. Comparison chart of existing and proposed Sqrt-CSLA

Comparison chart of existing and proposed multiplier design in terms of area, delay and power.

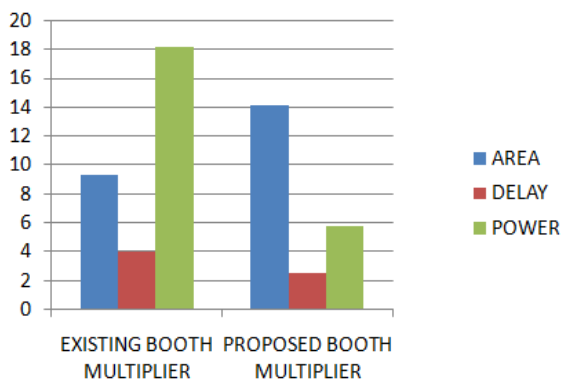


Fig.9. Comparison chart of existing and proposed Multiplier design.

V. CONCLUSION

The proposed new logic formulation for CSLA creates an optimized design for CS and CG units. The new logic formulation eliminated all the redundant logic operations of the conventional CSLA and proposed a new logic formulation for the CSLA. In the proposed scheme, the CS operation is scheduled before the calculation of final-sum, which is altered from the existing methods. Using these proposed logic optimization, an efficient design is results the proposed Sqrt-CSLA. The synthesis result shows that proposed Sqrt-CSLA design consumes less power and 57% less ADP and 41% less PDP than the existing BEC based Sqrt-CSLA which is best among the existing Sqrt-CSLA designs, on average bit-widths. Finally the implementation of Proposed CSLA block in Booth encoded multiplier design based on Radix-4 algorithm follows an reduction partial products more than the existing multiplier design. The analysis result shows that the proposed multiplier design involves saves more than 50% power and more than 30% of delay when comparing with the existing booth multiplier design. The future work is to further enhance the multiplication operation. However, there are limitations in these work and several future research directions are possible.

REFERENCES

- [1] Basant Kumar Mohanty and Sujit Kumar Patel, "Area-Delay-Power Efficient Carry-Select Adder," IEEE trans on circuits and systems—II, vol. 61, no. 6, June 2014.
- [2] Kostas Tsoumanis Sotiris Xydis, Constantinos Efstathiou, Nikos Moschopoulos, and Kiamal Pekmestzian "An Optimized Modified Booth Recoder for Efficient Design of the Add-Multiply Operator" IEEE trans on circuits and systems—I, Vol. 61, No. 4, April 2014.
- [3] O. J. Bedrij, "Carry-select adder," IRE Trans. Electron. Comput., vol. EC-11, no. 3, pp. 340–344, Jun. 1962.
- [4] Y. Kim and L.-S. Kim, "64-bit carry-select adder with reduced area," Electron. Lett., vol. 37, no. 10, pp. 614–615, May 2001.
- [5] Y. He, C. H. Chang, and J. Gu, "An area-efficient 64-bit square root carry-select adder for low power application," in Proc. IEEE Int. Symp. Circuits Syst., 2005, vol. 4, pp. 4082–4085.
- [6] Ramkumar and H. M. Kittur, "Low-power and area-efficient carry-select adder," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 20, no. 2, 371–375, Feb. 2012.
- [7] I.-C. Wey, C.-C. Ho, Y.-S. Lin, and C. C. Peng, "An area-efficient carry select adder design by sharing the common Boolean logic term," in Proc. IMECS, 2012, pp. 1–4.
- [8] S. Manju and V. Sornagopal, "An efficient Sqrt architecture of carry select adder design by common Boolean logic," in Proc. VLSI ICEVENT, 2013,1–5.
- [9] Booth, "A signed binary multiplication technique," Quart. J. Math., vol. IV, pp. 236–240, 1952.
- [10] S. Wallace, "A suggestion for a fast multiplier," IEEE Trans. Electron Comput., vol. EC-13, no. 1, pp. 14–17, Feb. 1964.
- [11] R. Cooper, "Parallel architecture modified Booth multiplier," Proc. Inst. Electr. Eng. G, vol. 135, pp. 125–128, 1988.
- [12] F. Elguibaly, "A fast parallel multiplier-accumulator using the modified Booth algorithm," IEEE Trans. Circuits Syst., vol. 27, no. 9, pp. 902–908, Sep. 2000.