ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Low Power Efficient MAC Unit using Proposed Carry Select Adder

Y Rama Krishna
PG Scholar in VLSI Design,
Department of ECE,
Dhanekula Institute of Engineering & Technology,
Ganguru, Krishna Dist., Andhra Pradesh, India.

K Suresh Babu
Assistant Professor,
Department of ECE,
Dhanekula Institute of Engineering & Technology,
Ganguru, Krishna Dist., Andhra Pradesh, India.

Abstract:
A design of high performance 8 bit Multiplier-and-Accumulator (MAC) is implemented in this paper. MAC unit performs important operation in many of the digital signal processing (DSP) applications. The multiplier is basic array multiplier and the adder is proposed carry select adder. 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 good for square-root (SQRT) CSLA. The total design is coded with Verilog-HDL and the synthesis is done using Cadence RTL complier using typical libraries of TSMC 180nm technology.The total power dissipation is 605uW.

Keywords: ***Adder, arithmetic unit,low power design, Array multiplier, Carry select adder, multiplier and accumulator (MAC).***

INTRODUCTION

Low power,area-efficient, and high-performance VLSI systems are increasingly used in portable and mobiledevices, multistandard wireless receivers, and biomedical instrumentation [2], [3]. An adder is the main component of anarithmetic unit. A complex digital signal processing (DSP) system involves several adders. An efficient adder design essentiallyimproves the performance of a complex DSP system. A ripple carry adder (RCA) uses a simple design, but carrypropagation delay (CPD) is the main concern in this adder. Carry look-ahead and carry select (CS) methods have beensuggested to reduce the CPD of adders.

A conventional carry selects adder (CSLA) is an RCA–RCA configuration that generates a pair of sum words and outputcarry bits corresponding the anticipated input-carry (Cin = 0 and 1) and selects one out of each pair for final-sum and final-output-carry [4]. A conventional CSLA has less CPD than an RCA, but the design is not attractive since it uses a dual RCA. Few attempts have been made to avoid dual use of RCA in CSLA design. Kim and Kim [5] used one RCA andone add-one circuit instead of two RCAs, where the add-onecircuit is implemented using a multiplexer (MUX). In [6] they proposed a square-root (SQRT)-CSLA to implement largebit-width adders with less delay. In a SQRT CSLA, CSLAswith increasing size are connected in a cascading structure. The main objective of SQRT-CSLA design is to provide a parallelpath for carry propagation that helps to reduce the overall adder delay. As suggested[7] a binary toBEC-based CSLA. The BEC-based CSLA involves less logic resources than the conventional CSLA, but it has marginally higher delay. A CSLA based on common Boolean logic (CBL) is also proposed in [8] and [9]. The CBL-based CSLA of [8] involves significantly less logic resource than the conventional CSLA but it has longer CPD, which is almost equal to that of the RCA. To overcome this problem, a SQRT-CSLA based on CBL was proposed in [9]. However, the CBL-based SQRTCSLA design of [9] requires more logic resource and delay than the BEC-based SQRT-CSLA of [7]. We observe that logic optimization largely depends on availability of redundant operations in the formulation, whereas adder delay mainly depends on data dependence. In the existing designs, logic is optimized without giving any consideration to the data dependence. In this brief, we made an analysis on logic operations involved in

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

conventional and BEC-based CSLAs to study the data dependence and to identify redundant logic operations. Based on this analysis, we have proposed a logic formulation for the CSLA. The main contributions in this brief are logic formulation based on data dependence and optimized carry generator (CG) and CS design. Most of digital signal processing methods use nonlinear functions such as discrete cosine transform (DCT) or discrete wavelet transforms (DWT). Because they are basically accomplished by repetitive application of multiplication and addition, speed of the multiplication and addition arithmetic determines the execution speed and performance of the entire calculation. Multiplication –and -accumulate operations are typical calculation. Multiplication –and – accumulateoperations is typical for digital filters. Therefore, the functionality of the MAC unit enables high-speed filtering and other processing typical for DSP applications. Since the MAC unit operates completely independent of the CPU, it can process data separately and there by reduce CPU load. The application like optical communication systems which is based on DSP, require extremely fast processing of huge amount of digital data. The Fast Fourier Transform(FFT)alsorequiresadditionandmultiplication. AMACunitconsistsofamultiplierand an accumulator containing the sum of the previous successive products. The MAC inputs are obtained from the memory location and given to the multiplier block. The design consists of 8 bitarraymultiplier,16 bit carry selectadderandaregister.Thispaperisdividedintosixsecti ons.Inthefirstsectionthe introduction about MAC unit is discussed. In the secondsectiondiscussaboutthedetailedoperationofMAC unit.Thethirdand fourth section deals with the operation of modified array multiplier and carry select adder respectively. In the fifth section, the obtainedresultforthe8bitMACunitisdiscussedandfinally the conclusion is made in the sixth section.

ARRAY MULTIPLIER

Consider the multiplication of two unsigned n-bit numbers, where A=an–1an–2 …a0 is the multiplicand and B = bn–1bn–2 …b0 is the multiplier. The product P = p2n– 1, p2n–2 …p0 can be written as follows:

$$P=\sum_{i=0}^{n-1}\sum_{j=0}^{n-1} a_i.b_j.2^{i+j}$$

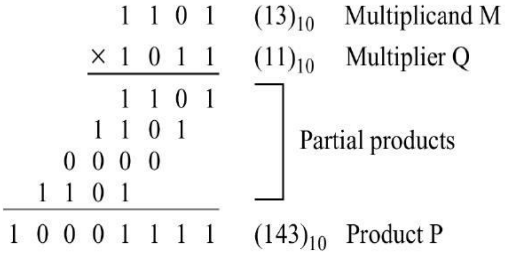

Fig.1 Multiplication Example

An array implementation is shown in Figure 2. In the 4x4 Array multiplier, the multiplier array consists of 3 rows of Carry-save adders (CSAs), in which each row contains 3 full adders (FAs). Each FA has three inputs and two outputs. The sum bit and the carry bit. 3 FAs in the first CSA row that have only two valid inputs can be replaced by 3 half adders (HAs) and 3 FAs in the last row can be constructed as a 3-bit ripple-carry adder.

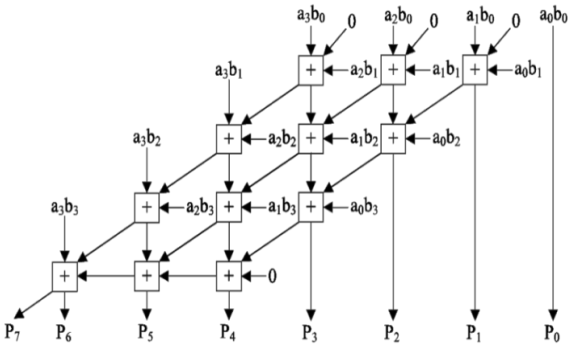

Fig.2.Array Multiplier

On the other hand, the Baugh-Wooley multiplier uses the same array structure to handle 2's complement multiplication, with some of the partial products replaced by their complements. The multiplier array consists of (n–1) rows of carry-save adders (CSA), in which each rows contains (n–1) full adders (FA).The last row is a ripple adder for carry propagation

PROPOSED ADDER DESIGN

The proposed CSLA is based on the logic formulation given in (3a)-(3g), and its structure is shown in Fig.3.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

It consists of one HSG unit, one FSG unit, one CG unit, and one CS unit. The CG unit is composed of two CGs (CG0 and CG1) corresponding to input-carry '0' and '1'. The HSG receives two *n*-bit operands (*A* and *B*) and generate *half-sum* word S_0 and *half-carry* word C_0 of width *n* bits each. Both CG0 and CG1 receive S_0 and C_0 from the HSG unit and generate two *n*-bit full-carry words c_1^0 and c_1^1 corresponding to input-carry '0' and '1', respectively. The logic diagram of the HSG unit is shown in 3(b). The logic circuits of CG0 and CG1 are optimized to take advantage of the fixed input-carry bits. The optimized designs of CG0 and CG1 are shown in 3(c) and (d), respectively.

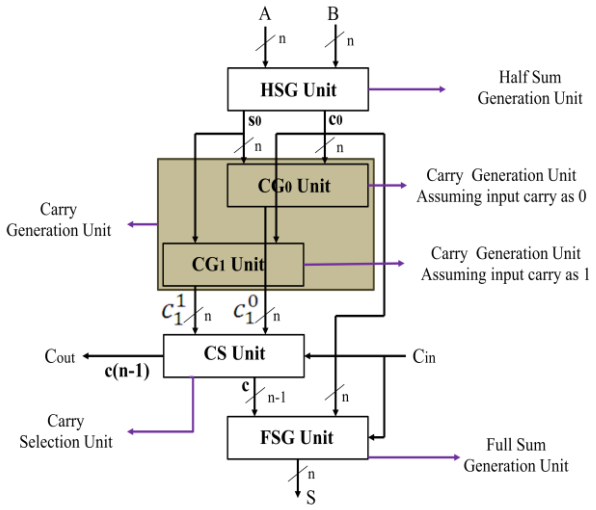

Fig.3 Proposed architecture

The CS unit selects one final carry word from the two carry words available at its input line using the control signal C_{in}. It selects c_1^0 when C_{in} = 0; otherwise, it selects c_1^1. The CS unit can be implemented using an *n*-bit 2-to-l MUX. However, we find from the truth table of the CS unit that carry words c_1^0 and c_1^1 follow a specific bit pattern. If $c_1^0(i)$ = '1', then $c_1^1(i)$ = 1, irrespective of $S_0(i)$ and $C_0(i)$, for $0 \leq i \leq n-1$. This feature is used for logic optimization of the CS unit. The optimized design of the CS unit is shown in 3(e), which is composed of *n* AND–OR gates. The final carry word *c* is obtained from the CS unit. The MSB of *c* is sent to output as C_{out}, and $(n-1)$ LSBs are XORed with $(n-1)$ MSBs of *half-sum* (S_0) in the FSG [shown in 3(f)] to obtain $(n-1)$ MSBs of *final-sum* (s). The LSB of S_0 is XORed with C_{in} to obtain the LSB of *S*.

$$s_0(i) = A(i) \oplus B(i)$$

$$c_0(i) = A(i).B(i) \qquad 3(a)$$

$$c_1^0(i) = c_0(i) + s_0(i).c_1^0(i-1)$$

For $(c_1^0(0) = 0)$ 3(b)

$$c_1^1(i) = c_0(i) + s_0(i).c_1^1(i-1)$$

For $(c_1^1(0) = 1)$ 3(c)

C(i)=$c_1^0(i)$ if c_{in}=0 3(d)

C(i)=$c_1^1(i)$ if c_{in}=1 3(e)

$C_{out} = C(n-1)$ 3(f)

$S(0) = S_0(0) \oplus C_{in}$

$S(i) = S_0(i) \oplus C(i-1)$ 3(g)

MAC IMPLEMENTATION

TheMultiplier-Accumulator(MAC)operation is thekeyoperation not only in DSP applications but also in multimedia information processing and various other applications. MAC unit consist of multiplier, proposed adder and an accumulator. In this paper, we used 8-bitarray multiplier.The MAC inputs are obtained from the memory location andgiven tothemultiplierblock.Theinputwhichisbeingfedfromthe memorylocation is 8 bit.Whentheinput isgiventothemultiplieritstartscomputing value for the given 8-bit input and hence the output will be16 bits.

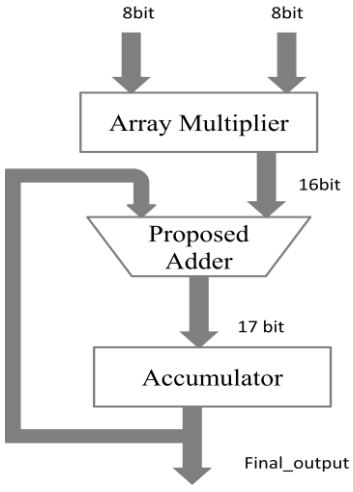

Fig4. MAC Unit

Themultiplieroutputisgivenastheinput to proposed carry select adder whichperformsaddition.Theoutputof

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

proposed carryselect adderis17 biti.e.onebit is for the carry (16 bits+ 1 bit). Then, the output is given tothe accumulator.Theoutputoftheaccumulatoristaken out or fed back as one of the input to the carry select adder.The figure 1 shows the basic architecture of MACunit.

SIMULATIONS AND LAYOUT

We have designed the SQRT-CSLA in Verilog using the proposed CSLA design and the existing CSLA designs of [7] and [8] for bit-widths 16, 32,64, and 128. All the designs are synthesized in the Cadence RTL Compiler (RTL) using the SAED 180-nm CMOS library. The netlist file Extracted from RTL Compiler. As shown in Table I, the proposed SQRT-CSLA involves significantly less area and less delay and consumes less power than the existing designs. We can find from Fig. 5 that the proposed SQRT-CSLA design offers a saving of 21.9% area and 44.3% power than the RCA-based conventional SQRT-CSLA, 7.16% area and 33.3% power than BEC-based SQRT-CSLA and 27% area and 52.2% power than CBL-based SQRTCSLA on average, for different bit-widths. The power comparison in µW is shown in fig 6. Power required for the proposed design is very low compared to the other designs. The area overhead is shown in fig 7 and Data Arrival Time comparison is shown in fig 8 as a graph.The 8-bit MAC is designed using proposed SQRT CSLA and it is compared with previous designed addersConventional,BEC,CBL (comparison shown in table-II).The simulation result and layout is shown below. The proposed MAC design offers a saving of 6.1% area and 14% power than the MAC using CONV-CSLA; 1.57% area and 9.59% power than the MAC using BEC based CONV-CSLA; 7.9% area and 17.2% power than the MAC using CBL based CONV-CSLA. From fig 9 it can be seen that the area required for MAC is very less compared to the conventional MAC designs. Fig 10 shows the power comparison graph and fig 11 shows the Data Arrival Time comparison graph. Fig 12 shows the Layout of Mac using the proposed Adder.

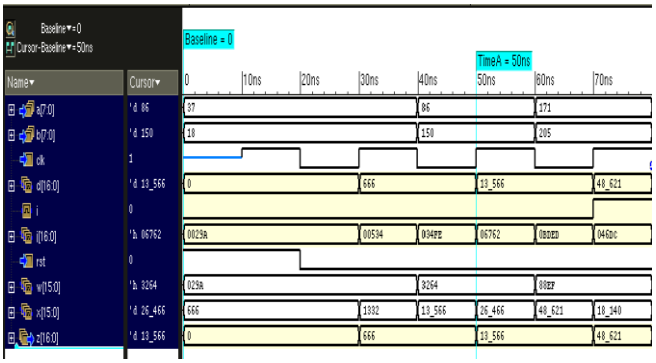

Fig.5 Simulation output of 8-Bit MAC unit

TABLE –I: Post layout ASIC synthesis results Comparison

Design	Width(n)	DAT(ns)	Area(um²)	Power(uw)
SQRT-CSLA (CONV)	16	2.903	2259	171.797
	32	4.523	4787	390.738
	64	7.231	9883	828.898
	128	11.753	20115	1773.023
SQRT-CSLA (BEC)	16	2.860	1899	143.522
	32	4.705	3948	314.673
	64	7.177	8060	634.420
	128	11.631	16296	1339.036
SQRT-CSLA (CBL)	16	7.517	2418	200.189
	32	14.287	5026	436.288
	64	27.569	10242	892.739
	128	53.580	20676	1879.710
Proposed SQRT-CSLA	16	2.284	1763	95.613
	32	4.119	3526	195.676
	64	7.789	7052	387.720
	128	15.130	14104	808.866

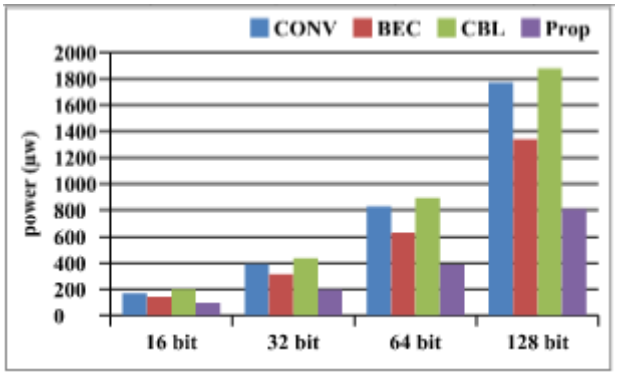

Fig.6. Power Comparison

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

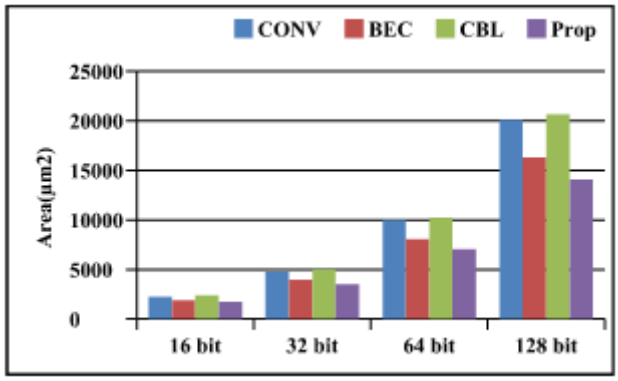

Fig.7Area Comparison

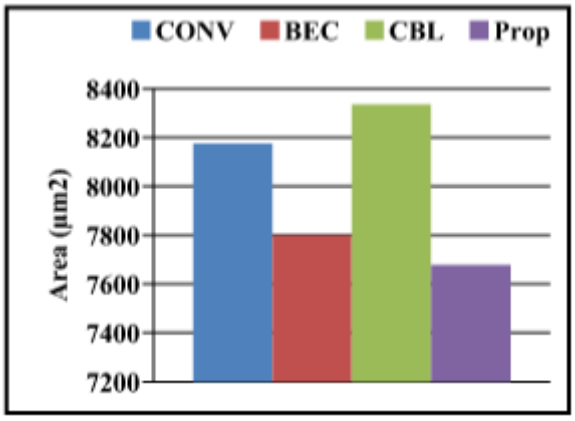

Fig.8DAT Comparison

TABLE - II: Implemented Post Layout Synthesis Results Comparison for MAC

Design	DAT(ns)	Area(um²)	Power(uw)
MAC using SQRT-CSLA (CONV)	6.948	8176	705.160
MAC using SQRT-CSLA (BEC)	7.137	7800	670.039
MAC using SQRT-CSLA (CBL)	8.555	8336	732.207
MAC using Proposed SQRT-CSLA	6.795	7677	605.765

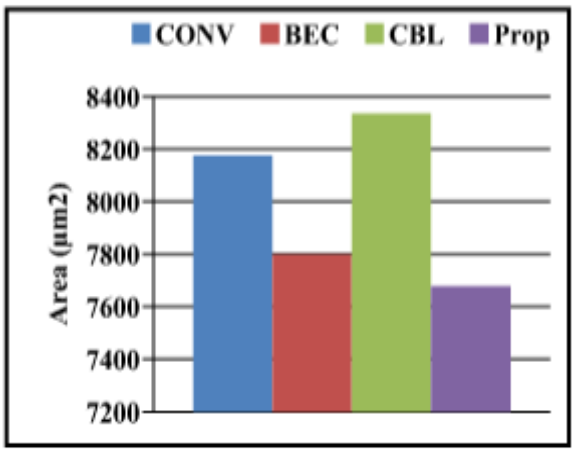

Fig.9 Area Comparison

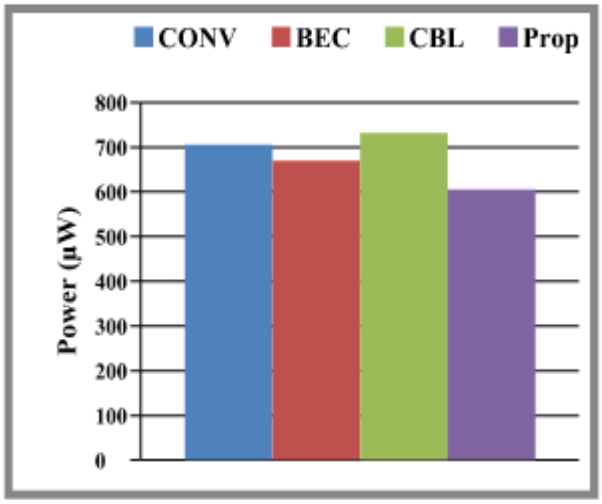

Fig.10 Power Comparison

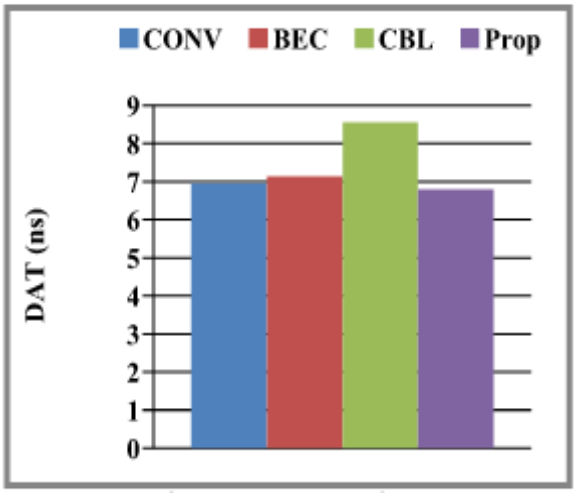

Fig.11 DAT Comparison

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Layout of MAC by using Proposed SQRT CSLA:

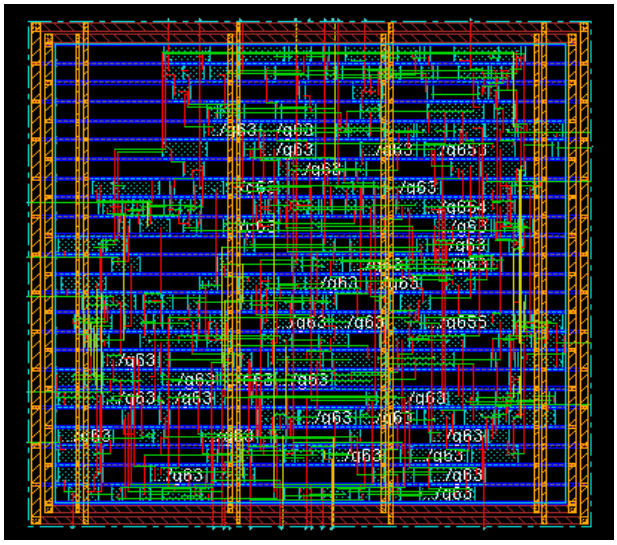

Fig.12.MAC Layout

CONCLUSION

We have eliminated all the redundant logic operations of the conventional CSLA and proposed a new logic formulation for the CSLA. The proposed CSLA design involves significantly less area and delay than the recently proposed BEC-based CSLA. Due to the small carryoutput delay, the proposed CSLA is a good Design for the SQRT adder. Due to this performance results CONV-CSLA, BEC, CBL, Proposed adders is placed in MAC and Achieved better results. In future array multiplier can replaced with any low power multiplier and extended for different bit widths.

REFERENCES

[1] Basant Kumar Mohanty, Sujitkumar Patel, "Area-Delay-Power efficient carry select adder", IEEE Transactions on Circuit and Systems, Vol.61, No.6, June-2014, pp.418-422.

[2] K. K. Parhi, VLSI Digital Signal Processing. New York, NY, USA:Wiley, 1998.

[3] A. P. Chandrakasan, N. Verma, and D. C. Daly, "Ultralow-power electronicsfor biomedical applications," Annu. Rev. Biomed. Eng., vol. 10, pp. 247–274, aug. 2008.

[4] O. J. Bedrij, "Carry-select adder," IRE Trans. Electron. Comput.,vol. EC-11, no. 3, pp. 340–344, Jun. 1962.

[5] 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.

[6] Y. He, C. H. Chang, and J. Gu, "An area-efficient 64-bit square root carryselectadder for low power application," in Proc. IEEE Int. Symp. CircuitSyst., 2005, vol. 4, pp. 4082–4085.

[7] B. Ramkumar and H.M. Kittur, "Low-power and area-efficient carry-selectadder," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 20, no. 2,pp. 371–375, Feb. 2012.

[8] I.-C. Wey,C.-C. Ho, Y.-S. Lin, and C.C. Peng, "An area-efficient carryselect adder design by sharing the common Boolean logic term," in ProcIMECS, 2012, pp.1-4.

[9] S.Manju and V. Sornagopal, "An efficient SQRT architecture of carry selectadder design by common Boolean logic," in Proc. VLSI ICEVENT, 2013,pp. 1–5.