IJCDS 1570830453

AN EFFICIENT CARRY SAVE MULTIPLIER FOR SIGNAL PROCESSING APPLICATIONS

P. Divya Parameswari [1] and A. V. Ananthalakshmi[2]

[1]Department of Electronics and Communication Engineering,Puducherry technological university, Puducherry, India

[2]Department of Electronics and Communication Engineering,Puducherry technological university, Puducherry, India

Abstract:High performance computing architecture heavily depends on the performance of a multiplier. The key contribution of thiswork is to design an efficient carry save multiplier architecture which can be widely employed in signal processing. Two novelcontributions proposed in this work. Firstly, a modified vector merging adder is proposed that has less latency, power and hardware complexity than the existing adder, and the second contribution is a multiplexer- based full adder which has better area efficiency. This work focuses on achieving efficient carry save multiplication through the use of the proposed modified vector merging adder. The modified vector merging adder and multiplexer-based full adder provides an efficient design of carry save multiplier when compared to the existing design and also the proposed design is efficient in terms of speed, power consumption and transistor count. Xilinx ISE Design 14.7 version is used for the verification.

Keywords: Modified vector merging adder, Carry Save Multiplier, Multiplexer based full adder, VLSI design.

1. INTRODUCTION

In order to meet high speed input data processing, real-time Digital Signal Processing (DSP) architecture ne- cessitates a less complex, delay-free and energy- efficient multiplier. In any computing unit, multiplication is one of the most fundamental arithmetic operations, but it neces-sitates more hardware. Because digital signal processing requires a large number of filtrations, convolutions and otheroperations, the multiplier is an essential component. As a result, the multiplier's performance limits the performance of digital signal processing systems. Extensive research hasbeen done to improve the multiplier's speed and reduce its power consumption. The technique used to add the partial products has a big impact on the multiplier's performance. The main area of research in VLSI system design is area and power reduction in data path logic systems. High performance processors and systems have always required high – speed addition and multiplication. The basic block inmultiplier is adder. High speed multipliers depends heavily on the efficient design of the adder. The structure of atraditional carry save multiplier (CSM) is straightforward and consistent. The carry bits in a CSM are saved to pass diagonally downwards rather than being immediately added. While the carry save operation improves speed, the final vector-merging operation has an impact on delay performance. Various addition schemes like Carry Select Adder have been proposed over the years [1]. Comparison of delay between different adders were carried out in [2]. Multipliers are made efficient by improving the final addition [3]. The square root carry select adder is a vector merging adder used in the carry save multiplier to produce end products with extremely high transistor counts, resulting in increased latency and power consumption [4]. Low power VLSI design is highly essential so as to increase the speed of the system. The need for low power design is not onlyto increase the speed but also to include more featureson the single chip.To bring down the transistor count, power consumption and the delay of the work reported in [4] the proposed work focuses on modifying the existing vector merging adder architecture. This work has two novel contributions. A modified vector merging adder is proposedand secondly a multiplexer based full adder is proposed. Using the proposed modified vector merging adder and a full adder based on a multiplexer, an efficient carry save multiplier is designed which is efficient in terms of delay, power consumption and transistor count.

.

Section 1 givesan introduction. Exhaustive literature survey is carried out in section 2. Existing carry save multiplier architecture is discussed in section 3. Section 4 discusses in detail about the proposed work. Simulation results are shown in section 5. Comparison of the proposed work with the existing work is shown in section 6 followed by conclusion.

2. LITERATURE SURVEY

Aiswarya Simson and Deepak [1] proposed a mongrel CSLA which utilizes the advantages of both Kogge Stone adder and look ahead adder(CLA) to obtain higher speed. Kogge Stone adder is a particular type of Adder and hence it's extensively regarded as one of the high speed addition styles due to briskly carry generation. Look ahead adders are employed in the original stages of the modified adderto boost its speed as its calculation performance is better if the number of bits are less.

Yuan-Ho Chen[2]proposed a data scaling technology (DST) for use in a low-error fixed-width Booth multiplier (FWBM) to reduce truncation errors. The suggested DST produces more effective bits in low-error FWBMs by re-ducing the amount of redundant bits in the multiplicand. The FWBMs' truncationerrors are decreased by inserting both an error-compensation circuit and the suggested DST into them. We discovered that the proposed DSTFWBM(1 bit) had a signal-to-noise ratio that was more than 1.05 dB greater than an FWBM without a DST circuit. It was demonstrated that the DST method significantly increased the FWBMs' accuracy, making them appropriate for use in digital signal processing procedures.

Moslem Heidarpur and Mitra Mirhassani[3] pro- posed a unique hardware architecture for efficient field-programmable gate array (FPGA) implementation of Finite field multipliers for ECC. Performance parameters were established by implementing the proposed hardware on var-ious FPGA devices for varied operand sizes. The proposed method produced a lower combinational delay and area-delay product when compared to state-of-the-art works, demonstrating the effectiveness of the concept. According to implementation results, the suggested technique is, on average, 20% quicker than the Overlap-free Karatsuba al-gorithm (OKA) and 30% faster than Karatsuba

P. Kavipriya, S. Lakshmi, T. Vino M.R. Ebenezar Je- barani, G. Jegan[4] proposed a work which move towards toreduce the delay, Power, area of CSLA(Carry Select Adder)circuits. The typical carry select adder has drawbacks, such as high power consumption and larger chip size. In comparison to all other adder designs, the improvedSQRT CSLA (Square Root Carry Select Adder) employing common Boolean logic has less delay, low power, and lowered space. It also moves a little faster. Additionally, theproposed SQRT-CSLA has fewer transistors and requires less space and power, making it simple and efficient to im- plement on VLSI hardware. It has to be further enlarged to accommodate more bits. With the aid of new architectures, space and power can be reduced.

Weihang Tan, Benjamin M. Case, Antian Wang,Shuhong Gao, and Yingjie Lao[5]presented a novel high- speed modular multiplier architecture for polynomial mul- tiplication. The suggested architecture uses a divide-and- conquer tactic and takes advantage of a unique modulusto boost parallelism, accelerate computation, and enable broader applications across different cryptosystems. Ac- cording to the testing findings, our design reduces space consumption and delay, respectively, by about 27%and 39%when compared to earlier efforts.

Mohammad Saeed Ansari, Bruce F. Cockburn, Jie Han[6] proposed an improved logarithmic multiplier (ILM) that, unlike existing designs, rounds both inputs to their nearest powers of two by using a proposed nearest-one detector (NOD) circuit. Some entries in the truth table of a conventional adder cannot exist because the NOD's output utilises a one-hot representation. As a result, a little adderis created for the smaller truth table. Comparing the 88ILM to a recently published LM in the literature, power consumption can be reduced by up to 17.48.

Mary Christina Joy, Ansa Jimmy, Tony C.Thomas, and Manju I.Kollannur [7] compare the delays of different 16-bitadders. The latency of logic gates has been mathematically modelled using logical effort in180nm. This delay is then utilised to calculate the delay of various adders. Functional verification has been carried out using Modelsim.

R. S. Waters and E. E. Swartzlander [8] proposed an area efficient Wallace multiplier which reduces the area of amultiplier by using a smaller final adder. Synopsys Design Compiler used to create the designs using 90nm process technology. The synthesis results show that, as expected, thesuggested Wallace multiplier has the smallest area among the Wallace-based multipliers.

Pramod Patali and Shahana Thottathikkulam Kassim [9presented high throughput FIR filter designs with reduced effective latency using retiming and redesigned CSLA- based adders. FIR filter architectures are proposed in two different forms. The retiming of delay elements in the RSF module, as well as the utilization of RCAs resulted in a high throughput, low power and less complicated FIR filter structure. The retiming of delay elements in the RSF module, as well as the usage of proposed delay efficient adders and multipliers along the delay channels resulted in a high throughput energy efficient retimed FIR filter structure.

P.Pramod and T.K Shahana [10] proposed Modular hybrid adders with reduced critical path latency for various bit-widths. Modularity, reusability and a low design time for broad bit-width adder applications are the major features of the suggested architecture. Concatenation and incrementa- tion techniques are used to form a CLA-RCA-CSKA hybridstructure, which yields the adder structures.

H. Baba, T. Yang, M. Inoue, K. Tajima, T. Ukezono and T. A. Sato [11]developed an accuracy scalable multiplier that uses less power than a Wallace Tree Multiplier. The reduction ratio is determined by the level of precision required. To construct the accuracy compensation vector, the simple circuit of the basic OR gate is chosen at the expense of accuracy. This precision- configurable approximate multiplier will be useful in applications where power and area are more important than accuracy.

Raghava Katreepalli, Drona Merugubonia, Themistoklis [12] presented a high speed, power efficient carry select adder design based on manchester carry chain. When compared to previous designs, the suggested adder offered a significant improvement in terms of power delay product and hardware overhead.

Dr, Minal Saxena, Deepak Kumar patel, Raksha Chouksey and Deepak Kumar Patel[13] proposed and implemented a strategy that efficiently minimises the delay and greatly increases the speed of FIR filter operation. The proposed modified Adder is a straightforward programme approach for reducing the delay in Carry select Adder by introducing URDHWA multiplier compressor technique. By utilising this, an FIR filter is implemented.

K.Golda Hepzibha and C.P. Subha[14] proposed a CSLA design based on the Brent Kung(BK) parallel prefix adder, which outperforms RCA and Binary to Excess-1Converter (BEC) based CSLAs in terms of speed. The BEC-based BK CSLA adder outperforms other adders in termsof speed while sacrificing a tiny amount of transistor count.

R. Bala Sai Kesava, B. Lingeswara Rao, B. E. Studen, Bhimavaram, K. Bala Sindhuri, and N. Udaya Kumar[15] proposed employing CSLA to reduce the area of the Wallace tree multiplier. When compared to Wallace tree multipliers using CSLA and Wallace tree multipliers using BEC, the Wallace tree multiplier utilising CSLA with BEC takes up less space, memory and power.

A. Anline Reeta and N. Kaleeswari [16] proposed an adder tree scheduled in bit- level using CSA (carry select adder) and it is designed and implemented in VHDL using the Xilinx 13.2 ISE tool. The CSA is a high-speed, low- power constant addition block. FPGA is also used to implement it.

AshwiniA. LokhandeV.G. Raut[17] proposed carry select adder design which uses D- latch rather of using RCA waterfall structure for cin = 1or cin = 0. This CSLA is enforced in the adder of FIR sludge .the proposed design achieves the two folded advantages in terms of power and detection.

Bhavani Prasad Y, Ganesh Chokkuala, Srikanth Reddy.P and Samhitha.N.R[18] suggested Vedic multiplier consumes less power than existing multiplier approaches because of the addition of a new module of modified carry select adder. Instead of two RCAs in the carry select adder, a binary to Excess one convertor was utilised, which resulted in considerable reduction in delay.

3.CONVENTIONAL CARRY SAVE MULTIPLIER

A 4-bit unsigned Carry Save Multiplier is shown inFig.1 . It is made up of three rows of half and full addersfor adding partial products, as well as a vector merging adder for generating final multiplication results. Eight-bit unsigned Carry Save Multiplier working is as follows. By multiplying the first bit of the multiplier series with all of the bits in the multiplicand series, the first row is obtained. By multiplying the second bit of the multiplier series with all of the bits in the multiplicand series, the second row is obtained. The third and fourth rows are made by adding the first two rows using half adder. By multiplying thethird bit of the multiplier series with all of the bits in the multiplicand series, the fifth row is obtained. The sixth and seventh rows are obtained by adding the third, fourth, and fifth rows using full adder, and the process is repeated. After obtaining the last two rows same steps can't be followed in order to obtain the output, the carry has to propagate to next bit. Since the bits are merged, it is called as a vector merging adder.

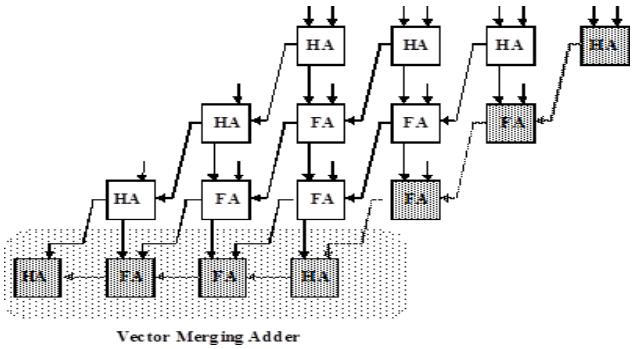


Fig.1 4-bit unsigned Carry Save Multiplier

4. MODIFIED CARRY SAVE MULTIPLIER

The modified carry save multiplier is created by combining the methods listed below:-

1)The improved full adder (IFA) structure[9] is replacedby the proposed multiplexer based full adder.

2)The square root carry select vector merging adderis replaced by modified vector merging adder which isefficient in terms of delay, area and power.

A)proposed multiplexer based full adder

A conventional full adder is made up of 66 transistors.In order to reduce the number of transistors in the existing work, an improved full adder shown in Fig.2 requires only 44 transistors[9]. The proposed full adder based on multiplexer has further reduced the transistor count to 38. When Cin is zero, S [0] represents the sum and C[0] represents the carry. When Cin is one, S[0] represents the sum and C[1] represents the carry.

This can be expressed as :-

$$\text{Sum} = s[0]\overline{cin} + s[1]cin$$

$$\text{Cout} = c[0]\overline{cin} + c[1]cin$$

Where

$$s[0] = A \oplus B$$

$$s[1] = \overline{A \oplus B}$$

Thus

$$\text{Sum} = (A \oplus B)\overline{cin} + \left(\overline{A \oplus B}\right)cin$$

$$\text{Sum} = A \oplus B \oplus cin$$

While

$$c[0] = AB$$

$$c[1] = A + B$$

Thus

$$\text{Cout} = (AB)\overline{cin} + (A + B)cin$$

Sum and Cout expressions represents the standard expressions of a full adder. Thus a full adder can be implemented using a multiplexer.

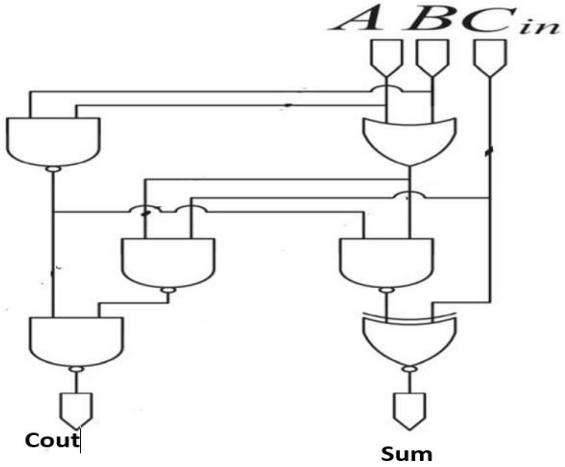


Fig. 2. Existing Improved Full Adder

Four NAND gate,one XNOR gate and one OR gate is required in the existing improved full adder thereby resulting in a total of 44 transistors as per the transistor count represented in [9]. The proposed multiplexer-based full adder as shown in Fig.3., on the other hand, only requires two NAND, two NOT, one OR gate and one 2X1 multiplexer, resulting in a total of 38 transistors. As a result,the proposed full adder based on a multiplexer is efficient interms of logic utilization. Comparison of device utilization of existing improved full adder and proposed multiplexer-based full adder is shown in table1.

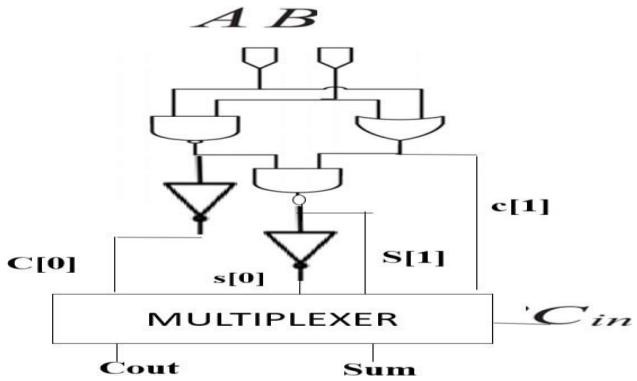


Fig. 3. Proposed Multiplexer Based Full Adder

TABLE 1:- Comparison of Device Utilization of Existing And Proposed Full Adder

	EXISTING IMPROVED FULL ADDER			PROPOSED MULTIPLEXER BASED FULL ADDER		
GATES	NO. OF GATES	NO. OF TRANSISTORS	TOTTAL TRANSISTORS	NO. OF GATES	NO. OF TRANSISTORS	TOTAL TRANSISTORS
NAND	4	4	16	2	4	8
XNOR	1	22	22	-	-	-
OR	1	6	6	1	6	6
NOT	-	-	-	2	2	4
MULTIPLEXER	-	-	-	1	20	20
	TOTAL		44	TOTAL		38

1)CMOS realization of a NAND gate requires 4 transistors

1.To realize existing improved full adder, we require 4 NAND gates so (4*4=16) totally we require 16 transistors.

2.Proposed multiplexer based full adder requires 2 NAND gate so (2*4=8) totally we require 8 transistors.

3.Therefore, when compared to proposed multiplexer based full adder the existing improved full adder requires 8 more transistor for NAND gate.

2)CMOS realization of a OR gate requires 6 transistors

1.To realize both the existing improved full adder and Proposed multiplexer based full adder, we require 1 OR gate so (1*6=6) totally we require 6 transistors in terms of OR gate.

2.Therefore, when compared to proposed multiplexer based full adder, the existing improved full adder has no improvement in transistor count for OR gate.

3.Therefore, when compared to the existing improved full adder, the proposed multiplexer based full adder requires 20 more transistor for multiplexer.

3)CMOS realization of a XNOR gate requires 22 transistors

1.To realize existing improved full adder, we require 1 XNOR gate so (1*22=22) totally we require 22 transistors.

2.Proposed multiplexer based full adder requires noXNOR gate.

3.Therefore, when compared to proposed multiplexer based full adder the existing improved full adder requires22 more for XNOR gate.

4)CMOS realization of a 2 x 1 MULTIPLEXER requires 20 transistors

1.The multiplexer consists of 1 NOT gate requires 2 transistors.2 AND gate each requiring 6 transistors and 1 OR gate which requires 6 transistors. So, totally multiplexer requires 20 transistors.

2.To realize existing improved full adder, multiplexer isnot required.

3.Proposed multiplexer based full adder requires 1 multiplexer so (1*20=20) totally we require 20 transistors

B)Concept behind the Proposed Vector Merging Adder

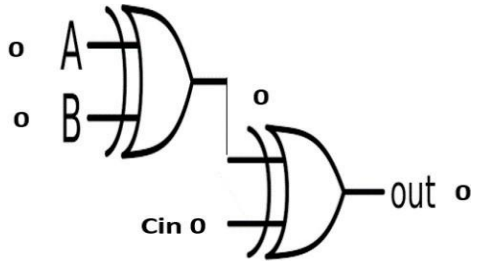


Fig. 4. Two Consecutive XOR Gate

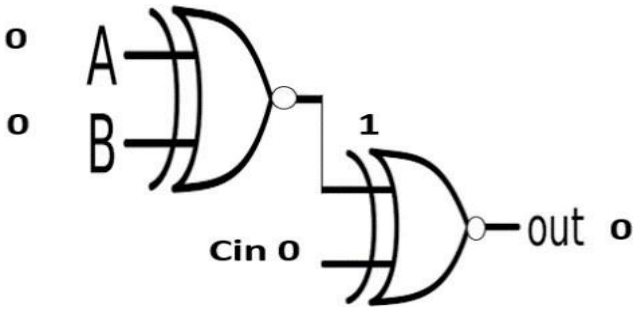


Fig. 5. Two Consecutive XNOR Gate

Full adder output is obtained by using two consecutive XOR gates or XNOR gates. In Fig.4, the sum output iszero when all the three inputs are zero. In the same way,in Fig.5 the sum output is zero when all the three inputsare zero. This is true for all inputs. Thus in modified vector merging adder the sum output can be obtained by using two consecutive XNOR gates instead of employing XOR gates. Truth table of full adder is shown in Table 2.

TABLE:-2 Truth Table of Full Adder

Inputs			Outputs	
A	B	C_{in}	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

c)Efficient Proposed Modified Vector Merging Adder

The efficient modified vector merging adder consists of three blocks as depicted in Fig.6.

1)Sum Generation (SG) block
2)Carry Generation (CG) block
3)Final Sum Generation (FSG)Blo

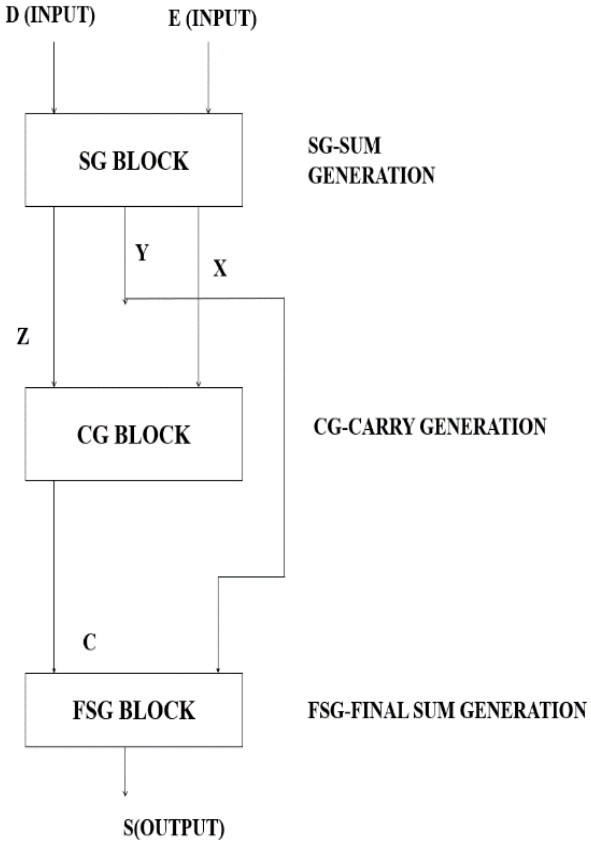


Fig. 6. Block Diagram of Proposed Modified Vector Merging Adde.

1.SUM GENERATION BLOCK

The Sum Generation (SG) block uses modified XNOR gate as shown in figure 7a consisting of two NAND and an OR gate for the generation of bitwise sum, except for the first sum bit.However, one bit in the first sum bit is zero. As a result,in a half adder, when cin is zero, the sum is obtained using the modified XOR gate as shown in 7b, while the carry is obtained using the AND gate.

Consider an example as shown in Fig.8, the inputs are from D[0] to D[6] and from E[0] to E[6]. The outputs are from X[0] to X[6], from Y[0] to Y[6], and from Z[0] to Z[6]. To obtain the outputs, the inputs enter NAND OR gate while the outputs obtained enter as inputs to the AND gate. The NAND gate enters the NOT gate to obtain the output for the first sum bit alone.

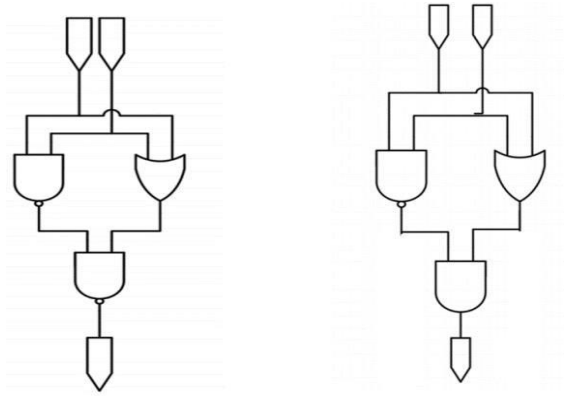


Fig. 7. (a) Modified XNOR Gate[9] (b) Modified XOR Gate[9]

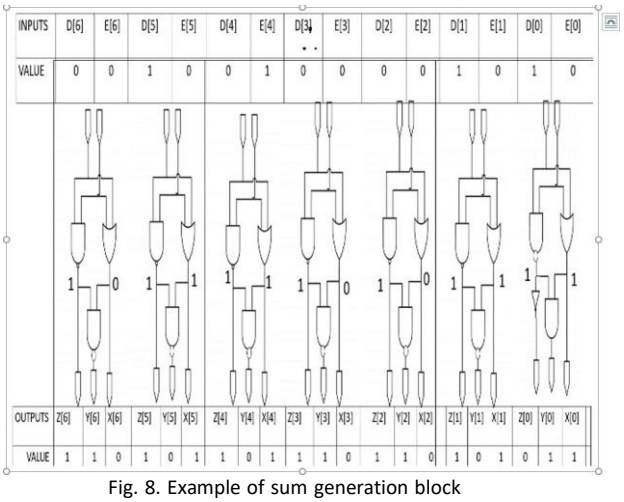


Fig. 8. Example of sum generation block

2.CARRY GENERATION BLOCK

To generate the carry output, the Carry Generate (CG) block shown in Fig.9 employs a series of NAND gates. Each NAND gate series output is propagated to the next NAND gate series. Despite the fact that seven bits are added, only six carry bits are generated because the first sum bit's carry is already generated using the half adder concept. The inputs of CG block is obtained from previous SG block.

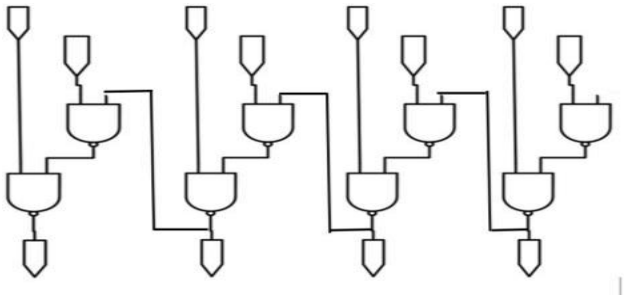
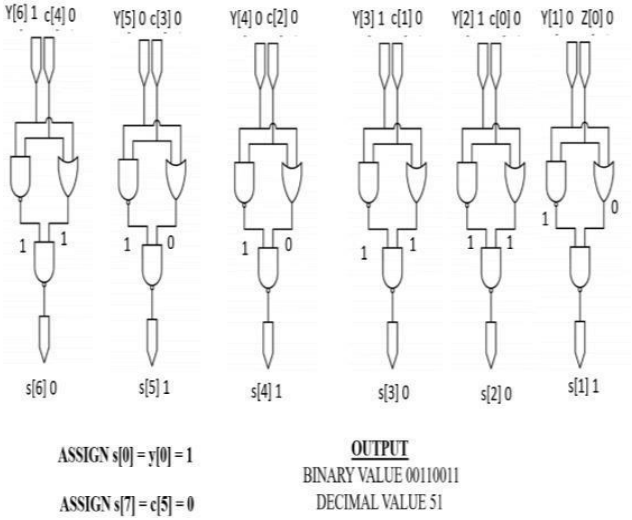


Fig.9 series of NAND gate

Fig.11 Example of final sum generation block



ASSIGN s[0] = y[0] = 1

ASSIGN s[7] = c[5] = 0

OUTPUT

BINARY VALUE 00110011

DECIMAL VALUE 51

As shown in the Fig.10, Z[0] to Z[6] and X[1] to X[6] are the inputs obtained from the previous SG block. Z[0]is the first bit which acts as one of the input. The X series input, as well as the previous output enters NAND gate. Theoutputs enter another NAND gate with Z series as another input to obtain the outputs. The outputs are labelled from C[0] to C[6].

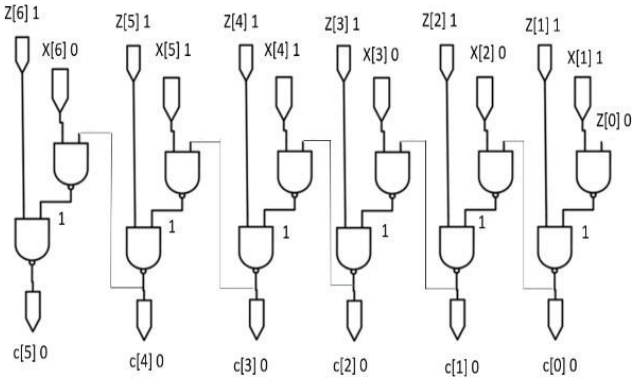


Fig. 10. EXAMPLE OF CARRY GENERATE BLOCK

3.FINAL SUM GENERATION BLOCK

The Final Sum Generation (FSG) block uses modified XNOR gate. This is the last block. Despite the fact that seven bits are added, only six carry bits are generated because the first sum bit's carry is already generated using the half adder concept To obtain the final value, the inputs enter the NAND OR gate combination, whose outputs also enter a NAND gate.As shown in Fig.11, Y[0] to Y[6], C[0] to C[4] and Z[0] are the inputs. Z[0] is the first bit in the first sum bit. To obtain the final value, the inputs enter the NAND OR gate combination, whose outputs also enter a NAND gate. S[1] to S[6] are the output finally, S[0] = Y[0] = 1 and S[7] = C[5] = 0 are assigned.

D. SIMULATION RESULTS

The simulation results of the modified efficient Vector Merging Adder and proposed carry save multiplier is shownin Fig.12 and Fig.13 respectively. The design is simulated and realized with the help of Xilinx Isim 14.7 Simulator.

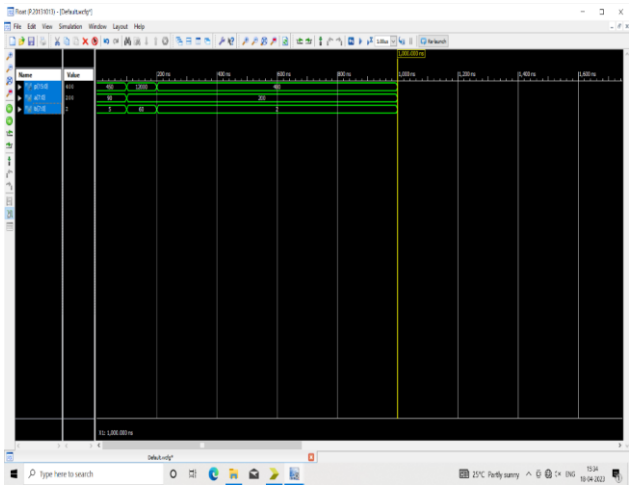


Fig. 12. Efficient Modified Vector Merging Adder Output

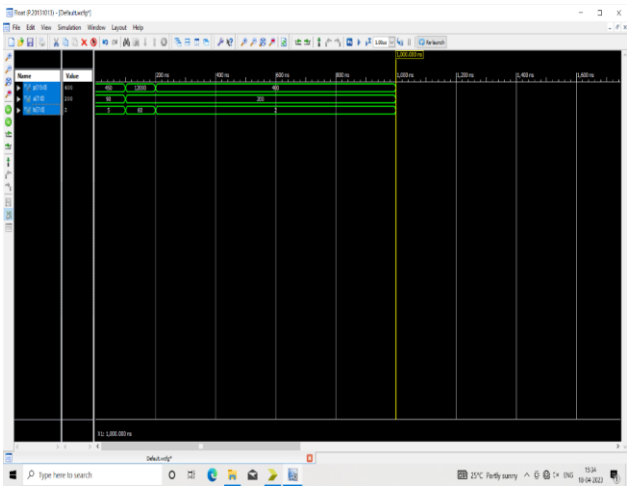


Fig. 13. Proposed Carry Save Multiplier Ouput

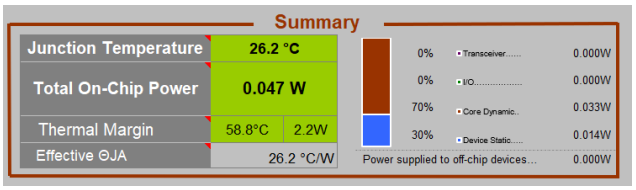


Fig. 14. Power Consumption of Existing Carry save multiplier

E.OUTPUT COMPARISON

The delay of the existing square root carry select adder- based Carry Save Multiplier (CSM) is depicted in table 3 with a delay of 15ns.The proposed efficient modified vector merging adder-based carry save multiplier has a delay of 14ns, as shown in table 1. The proposed carrysave multiplier is 7% faster than the existing carry save multiplier.Power consumption of the proposed carry save mul-tiplier is analyzed using XPower estimator. The power consumption of the existing square root carry select adder- based Carry Save Multiplier (CSM) is 0.047W as shown in Fig.14. The proposed carry save multiplier has a power con-sumption of 0.040W as shown in figure 15.

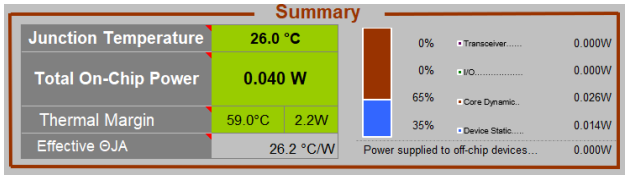


Fig. 15. Power Consumption of proposed Carry save multiplier

The proposed efficient modified vector merging adder-based carry save multiplier consumes 16% less power than the existing carrysave multiplier. The number of transistors used determines the size of an integrated circuit. Table 1 compares the squareroot carry select adder based Carry Save Multiplier (CSM) with the proposed efficient modified vector merging adder based carry save multiplier, and it is inferred that the latter uses fewer transistors than the former.

Table 4 compares the performance of the existing and proposed CSMs, revealing that the proposed CSM is 7,15 and 60 percent more efficient than the existing CSM in terms of delay, power and transistor count respectively.Table 2 compares the performance of the existing square rootcarry select adder[9],square root carry select adder using CBL(Common Boolean Logic)[4] and proposed square root carry select adder in terms of delay and Gates utilized. We are able to notice that the delay and gate utilization has got reduced in proposed work than the previous works[9],[4].

1)CMOS realization of a NAND gate requires 4 transistors
1.To realize existing Carry Save Multiplier, we require 215 NAND gates so (215*4=860) totally we require 860 transistors.

2.Proposed Carry Save Multiplier, requires 128 NAND gate so (128*4=512) totally we require 512 transistors.

3.Therefore, when compared to proposed Carry Save Multiplier, the existing Carry Save Multiplier, requires 348more transistors for NAND gate implementation.

2)CMOS realization of a AND gate requires 6 transistors
1.To realize existing Carry Save Multiplier, we require 215 AND gates so (60*6=360) totally we require 360 transistors.

2.Proposed Carry Save Multiplier, requires 43 AND gateso (43*6=258) totally we require 258 transistors.

3.Therefore, when compared to proposed Carry Save Multiplier, the existing Carry Save Multiplier, requires 102 more transistors for AND gate.

3)CMOS realization of a OR gate requires 6 transistors
1.To realize existing Carry Save Multiplier, we require 58 OR gates so (58*6=348) totally we require 348 transis- tors.

2.Proposed Carry Save Multiplier, requires 55 OR gate so (55*6=330) totally we require 330 transistors.

3.Therefore, when compared to proposed Carry Save Multiplier, the existing Carry Save Multiplier, requires 18 more transistors for OR gate.

4)CMOS realization of a NOR gate requires 4 transistors

1.To realize existing Carry Save Multiplier, we require 14 NOR gates so (14*4=56) totally we require 56 transis- tors.

2.Proposed Carry Save Multiplier, requires 7 NOR gate so (7*4=28) totally we require 28 transistors.

3.Therefore, when compared to proposed Carry Save Multiplier, the existing Carry Save Multiplier, requires 28 more transistors for NOR gate.

5)CMOS realization of a NOT gate requires 2 transistors

1.To realize existing Carry Save Multiplier, we require6 NOT gates so (6*2=12) totally we require 12 transistors.

2.Proposed Carry Save Multiplier, requires 84 NOT gateso (84x2=168) totally we require 168 transistors.

3.Therefore, when compared to existing Carry Save Multiplier, the proposed Carry Save Multiplier, requires 156 more transistors for NOT gate.

6)CMOS realization of a modified XNOR gate requires 14transistors

1.The XNOR gate consist of 2 NAND gates each requiring 4 transistors and 1 OR gate which requires 6transistors. So, totally XNOR gate requires 14 transistors.

2.To realize existing Carry Save multiplier and proposedCarry Save Multiplier we require 42 modified XNOR gates so (42*14=588) totally we require 588 transistors.

3.Proposed Carry Save Multiplier does not require modified XNOR gate.

4.Therefore, when compared to proposed Carry Save Multiplier, the existing Carry Save Multiplier requires 588 more transistors for modified XNOR gate.

7)CMOS realization of a modified XOR gate requires 16 transistors

1.The modified XOR gate consist of 1 NAND which requires 4 transistors,1 AND gate which requires 6 transis- tors and 1 OR gate which requires 6 transistors.

2.So, totally XNOR gate requires 16 transistors.To realize existing Carry Save Multiplier and proposedCarry Save Multiplier, we require 7 modified XOR gates so(7*16=112) totally we require 112 transistors.

3.Therefore, when compared to proposed Carry Save Multiplier, the existing Carry Save Multiplier has no im- provement in transistor count for modified XOR gate.

3)CONCLUSIONS AND FUTURE WORK

A delay, power and area efficient carry save multiplier ispresented in this work. Remarkably improved performance in terms of the performance metrics such as delay, power and transistor count is achieved through the use of theproposed multiplexer based full adder and modified vector merging adder. The structural simplicity and regularity of full adder and half adder array of the proposed Carry Save Multiplier (CSM) is improved when compared to the existing square root carry select adder. The transistor count of the multiplexer based full adder is less than the existing improved full adder because of the use of multiplexer wherethe sum and carry are selected based on Cin. The final vector merging addition using the proposed modified vector. merging adder further improves the speed. Future work is todesign a FIR filter using the proposed adder and multiplier.

TABLE I. Comparison of Performance of the Proposed CSM and Existing CSM

Multipliers	Delay(ns)	Power(W)	Transistor count
Existing CSM	15.634	0.047	2336
Proposed CSM	14.521	0.040	1408
Performance improvement	7%	16%	60%

TABLE II. comparison of the performance of the existing square root carry select adder[9],square root carry select adder using CBL(Common Boolean Logic)[4] and proposed square root carry select adder in terms of Gates utilized.

ADDER	GATE COUNT
Existing square root carry select adder[9]	80
Existing square root carry select adder using CBL[4]	1092
Proposed square root carry select adder (modified vector merging adder)	45
Performance improvement of proposed adder WRT CBL based square root CSLA[9]	95 % decrease in gate usage
Performance improvement of proposed adder WRT existing square root carry select adder[4]	43.7% decrease in gateusage

References

[1] A. Simson and D. S, "Design and implementation of high speed hybrid carry select adder," International Conference on Advances in Electrical, Computing, Communication and Sustainable Technolo- gies, pp. 978–1–7281–5791, 2021.

[2] Y. H. Chen, "Improvement of accuracy of fixed-width booth multi- pliers using data scaling technology," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 68, pp. 1018–1022, 2021.

[3] M. Heidarpur and M. Mirhassani, "An efficient and high-speed overlap-free karatsuba-based finite-field multiplier for fgpa imple- mentation," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 29, pp. 667–676, 2021.

[4] T. V. M. R. E. J. P. Kavipriya, S. Lakshmi and G. Jegan, "Booth multiplier design using modified square root carry-select-adder," In ternational Conference on Artificial Intelligence and Smart Systems(ICAIS), vol. 68, pp. 1647–1653, 2021.

[5] A. W. S. G. W. Tan, B. M. Case and Y. Lao, "High-speed modular multiplier for lattice-based cryptosystems," in IEEE Transactionson Circuits and Systems II: Express Briefs, vol. 68, pp. 2927–2931, 2021.

[6] B. F. C. M. S. Ansari and J. Han, "An improved logarithmic multi- plier for energy-efficient neural computing," in IEEE Transactions on Computers, vol. 70, pp. 614–625, 2021.

[7] T. C. M. I. Mary Christina Joy, Ansa Jimmy, "Modified 16 bit carry select and carry bypass adder architectures for high speed opera- tions," IEEE International Conference for Innovation in Technology,vol. 70, 2021.

[8] R. S. Waters and E. E. Swartzlander, "A reduced complexity wallace multiplier reduction," in IEEE Transactions on Computers, vol. 59, pp. 1134–11, 2020.

[9] P. Patali and S. T. Kassim, "An efficient architecture for signed carry save multiplication," IEEE Letters of the Computer Society, vol. 3, 2020.

[10] P. Pramod and T. K. Shahan, "High throughput fir filter architec- tures using retiming and modified csla based adders," IET Circuits Devices Syst.,, vol. 13, p. 1007–1017, 2019.

[11] M. I. K. T. T. U. H. Baba, T. Yang and T. A. Sato, "A low power and small-area multiplier for accuracy-scalable approximate computing," Proc. IEEE Comput. Soc. Annu. Symp. VLSI, p. 569–574, 2018.

[12] D. M. R. Katreepalli and T. Haniotakis, "A power-delay efficient carry select adder," 2nd International Conference for Convergence in Technology, pp. 1234–1238, 2017.

[13] R. C. D.K. Patel and M.Saxena, "Design of fast fir filter using compressor and carry select adder," 3rd International Conference on Signal Processing and Integrated Networks, pp. 460– 465, 2016.

[14] K. G. Hepzibha and C. P. Subha, "A novel implementation of high speed modified brent kung carry select adder," 10th International Conference on Intelligent Systems and Control, pp. 1–5, 2016.

[15] B. K. B. S. R. Bala Sai Kesava, B. Lingeswara B. E. Studen and N. U. Kumar, "Low power and area efficient wallace tree multiplier using carry select adder with binary to excess-1 converter," Confer- ence on Advances in Signal Processing (CASP) Cummins College of Engineering for Women, pp. 9–11, 2016.

[16] N. K. Aniline Reeta, "High performance fir filter using carry select adder to reduce area, delay," International Journal of Electrical and Electronics Engineers, ISSN- 2321-2055, vol. 07, 2015.

[17] V. G. R. Ashwini A. Lokhande, "Design and implementation of fir filter using carry select adder," International Journal of Science and Research, vol. 4, 2015.

[18] P. Y.B.Prasad, G. Chokkakula and N.R.Samhitha, "Design of low power and high speed modified carry select adder for 16 bit vedic multiplier," International Conference on Information Communica- tion and Embedded Systems, 2014.