ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology
Vol. 1, Issue 4, December 2014

Optimized area-delay and power efficient carry select adder

Mr. MoosaIrshad KP[1], Mrs. M. Meenakumari[2], Ms. S. Sharmila[3]

PG Scholar, Department of ECE, SNS College of Engineering, Coimbatore, India [1,3]

Associate Professor, Department of ECE, SNS College of Engineering, Coimbatore, India[2]

Abstract: In the design of Digital Integrated Circuits area occupancy, delay and power play an important role because of the increasing necessity of portable and fast systems. Adders are one of the most widely used digital components in various Digital Circuits. Carry Select Adder (CSLA) is one of the fastest adders used in many processors, multipliers, and different applications. Carry select method is having a good compromise between cost and performance in carry propagation method. In this paper a modified area and low power carry select adder along with lower delay is proposed. The proposed system has low area and power consumption with lower delay suitable for FPGA design than the existing carry select adders. Due to the optimized area, power and delay, the proposed CSLA design is a good substitution for all the existing CSLA. The proposed architecture is designed using VHDL and is then synthesized using XILINX 13.2i and is simulated in ISim for Spartan 3E FPGA.

Keywords: carry select adder, area efficient, delay efficient, low power adder

I. INTRODUCTION

VLSI systems are increasingly used in many applications. The Low power, Area efficient and less delay VLSI systems are being used in vast range of applications such as portable and mobile devices, multistandard wireless receivers and biomedical instrumentation. Adder is a basic component of arithmetic unit. An adder is not only used for adding but is also used for subtraction, so the adder will be used in the critical paths of many applications. For measuring the quality of an adder the most important things to be considered are propagation delay, area and power utilized by the adder. All of these three things must be in a balanced manner for getting a best adder.

Ripple carry adder is one of the conventional methods of adder. The ripple carry adder consists of a chain of full adders with length n. It can add multi bits using less area but the problem occurs due to its bigger carry propagation delay. So we go for carry look ahead adder, carry skip adder, carry select adder etc. Carry select adder is the fastest adder among these.

The most common scheme for accelerating carry propagation is carry look-ahead scheme proposed by Weinberger and Smith in 1958 [1]. Here they proposed look-ahead technique rather than carry-rippling technique to speed-upthe carry propagation. In Carry Look Ahead adder, the delay for adding two numbers depends on the logarithm of the size of the operands and hence it is theoretically one of the fastest schemes used for the addition. The carries generated and propagated depends only on the digits of the original numbers.

Carry-skip scheme was proposed by Kilburn et al. [2] it describes the technique to accelerate the carry propagation. A carry-skip adder reduces the time needed to propagate the carry by skipping over groups of consecutive adder stages. In VLSI technology the carry-skip adder is compatible in speed to
the carry look-ahead technique but it need only low power and area. Carry Skip.

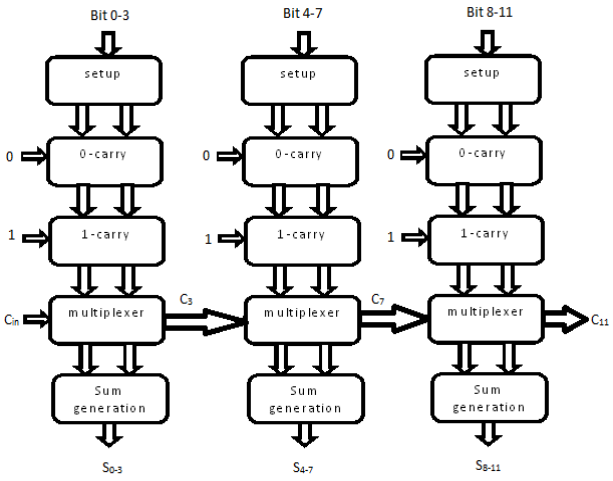

Fig.1: Conventional carry select adder

Adders take the advantages of both the generation and the propagation of the carry signal. The basic theme of carry select adder is to divide a long adder into fixed size adder sections and proceed with simultaneous section additions with appropriate carry input to select the true sum output. Like carry skip schemes, carry select scheme divides adder into blocks of ripple carry adder each with two replicas, one replica evaluates for carry in of 1, while the other one evaluates with carry-in of 0. This system was proposed by Bedrij in 1962 [3]. In this scheme, the carry-out from less significant block selects the sum and carryout of more blocks. In this way, the critical signal is generated at the least significant bit, and ripples through the least significant block and then conditionally selects the output of succeeding blocks. The Carry select adder generally consists of two Ripple Carry Adders and one multiplexer. Addition of two n-bit numbers with a Carry select adder is done with two adders in order to perform

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology
Vol. 1, Issue 4, December 2014

the calculation twice, one time with assumption of the carry being zero and the other being one. As the two results are calculated, the required sum as well as the apt carry is then selected with a carry selector (multiplexer) and is controlled by the carry from previous adders.

II. LITERATURE SURVEY

A conventional carry select adder is a configuration of dual RCA in which one RCA generates sum and carry output by assuming C_{in} = 0and the other RCA produce carry and sum and sum assuming C_{in} = 1 [3]. This conventional carry select adder has less carry propagation delay than conventional RCA adder but increases the complexity due to dual RCA structure. A carry select adder generating carry of block with carry in as 1 from the block with carry in as 0 was proposed by Tyagi.A[4] in 1990. Later in 1998 T.Y.Ceiang and M.J.Hsiao[5] proposed a carry select adder consisting of single ripple carry adder. This was a real breakthrough in the carry select adder history.

In 2001 a further modified carry select adder with increased delay but reduced area and power was presented by Kim and Kim [6]. Here the RCA section with c_{in}=1 was replaced using an add one circuit using multiplexer (MUX). Later in the year 2005 a further modified carry select adder which reduces the area and power consumption was proposed by Amelifard B, Fallah F and Pedram.M[7]. It reduces the gap between carry select adder and ripple carry adder.

Later a SQRT-CSLA was proposed by He et al[8] which helps in implementing large bit width adders with less delay. In this system the CSLA's with increasing bit widths are cascaded with each other. It helps in reducing the overall adder delay. A BEC based CSLA was further proposed by Ramkumar and Kittur[9] which had less resources than conventional CSLA but with more delay. A CBL (common Boolean logic) based CSLA[10] was also proposed which requires less logic resources but CPD(carry propagation delay) was similar to that of RCA. A CBL based SQRT CSLA [11] was also proposed but the design requires more logic resource and delay than BEC based SQRT CSLA.

Now a further modification of CSLA called Area-Delay-Power Efficient Carry Select Adder [12] was proposed. Here the carry generation is faster but the area consumption is not much reduced. The carry of the system is calculated before the sum generation.Also the carry generation unit was also replaced using an optimised logic.Thus the system have lesser carry output delay than all other system. Though the carry generation is faster, the area and power consumption are not much reduced. So a further modification with a reduction in area and power consumption, thus obtaining an optimized area-delay and power carry efficient carry select adder is proposed here.

III.OPTIMIZED AREA-DELAY AND POWEREFFICIENT CARRY SELECT ADDER

Optimized Area-Delay and Power Efficient Carry Select Adder.is having all the features of Area-Delay-Power Efficient Carry Select Adder.Here the redundant logic operations of the system are identified and eliminated and new logic formulations are proposed for the system. Also the AND, OR and XOR logic used in the system ischanged into a mux based gates with optimal usage of FPGA resources. This substitution helps to reduce the area and power consumption of the whole system. Also this substitution helps in optimized utilisation of the FPGA slices. Thus this carry select adder can be a good substitute for all the current adders and can be used in fast, power and area efficient devices.

A. *Architecture*

The optimized carry select adder architecture is shown in fig 2.The carry select adder mainly consist of four sections

1) Half sum generator
2) Carry generator
3) Carry selection
4) Full sum generator

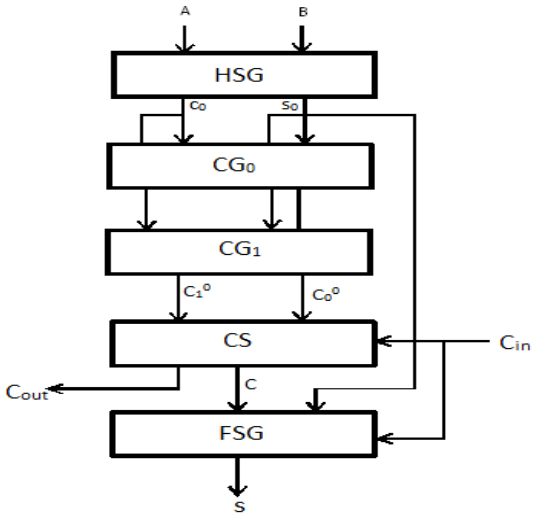

Fig 2: Architecture of optimized carry select adder

1) Half sum generation unit:

The sum generation unit makes use of full adder implementation using two half adders. Herethe first half adder receives the n-bit input and provides half adder sum and carry. The Half Sum Generator generates half-sum word s_0 and half-carry word $c_0$from the given inputs. It simply uses a half adder. The number of adders needed is same as the number of input bits.

$S_0(i) = A(i)$ B(i) ⊕

$C_0(i) = A(i)*B(i)$

Here the AND, XOR and OR gates used are modelled using multiplexer (MUX) in order to optimize the area-power and delay of the system by efficient utilization of slices in the FPGA.

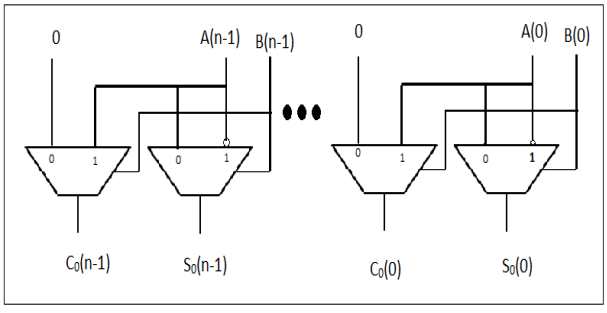

Fig 3: Half sum generation unit

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology
Vol. 1, Issue 4, December 2014

This half adder results are given as input for carry generator and full sum generator. The full sum generator obtains output after receiving the carry input.

2) Carry generation:

The output of the half adder is given as input to the carry generator circuit. Two carry generator circuits are used in the design, CG_0 and CG_1. CG_0 is used to generate carry by assuming carry input as 0 and CG_1 is used to generate carry by assuming input carry as 1. Both CG_0 and CG_1 receives half carry word and half sum word from the half adder and generate two n-bit full carry words C_1^0 and C_1^1 corresponding to input carry 0 and 1 respectively.

$$C_1^0(i) = C_1^0(i-1)*S_0(i)+C_0(i) \text{ for } (C_1^0(0)=0)$$
$$C_1^1(i) = C_1^1(i-1)*S_0(i)+C_0(i) \text{ for } (C_1^1(0)=1)$$

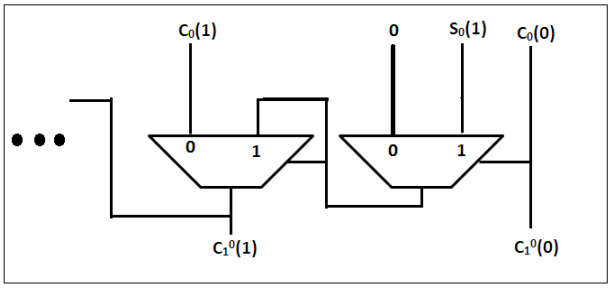

Fig 4: Carry generation unit for C_{in}=0

Fig 4 shows the mux implementation for the carry generator CG_0. The output of this section is given as input to the carry selection section.

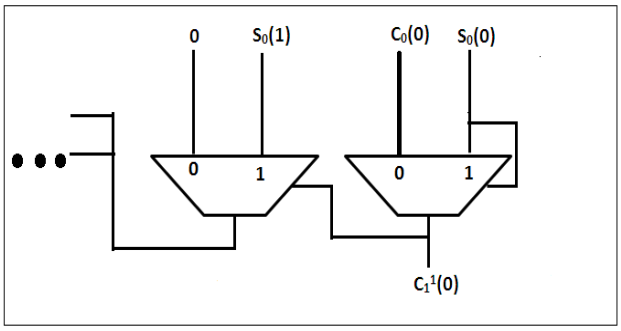

Fig 5: Carry generation unit for C_{in}=1

Fig 5 is the MUX implementation of carry generator CG_1. Here the output carry for carry input 1 is generated. This is also given as input to the carry selection section.

3) Carry selection unit:

The carry selection unit selects one the final carry from the two carry words available at its input line using the input carry c_{in}. Here the carry select unit selects the output of CG_0 if the input carry (c_{in}) is 0 and selects the output of CG1 If the input carry (c_{in}) is 1. The carry select unit here is implemented using an optimized design.

$C(i) = C_1^0(i)$ if (C_{in}=0)
$C(i) = C_1^0(i)$ if (C_{in}=1)
C_{out}= C(n-1)

Fig 6 shows the implementation of carry selection unit using MUX. The selection of carry output according to input carry is done here. The full sum of the system is generated by adding the selected carry with the half sum obtained in the half adder.

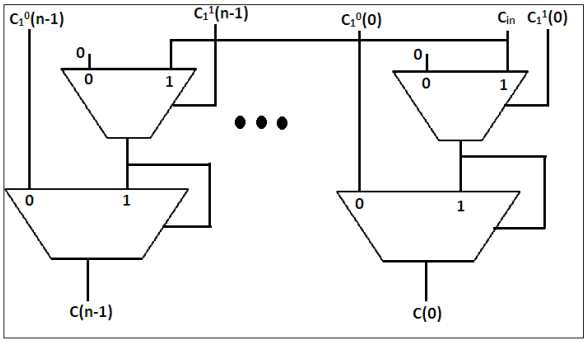

Fig 6: Carry selection unit

4) Full sum generation unit:

The full sum of the system is generated by XORing the selected carry from carry selection unit with the half sum obtained in the half adder. Thus the half sum and the obtained carry are both XORed to get the final sum of the system.

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

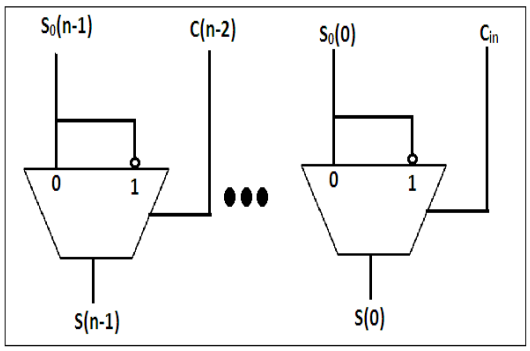

Fig 7: Full sum generation unit

Fig 7 shows the mux implementation of full sum generation. Here the circuit is used to Xor the selected carry from carry selection unit with the half sum obtained in the half sum generator. The MSB of c is sent to output as c_{out}, and (n−1). LSBs are XORed with (n−1) MSBs of half-sum(s0)in the FSG to obtain the final sum

B. *Implementation of Large Size Adders*

The SQRT-CSLA have fully exploited the multipath carry propagation feature of the carry select adder. SQRT-CSLA uses a chain of CSLAs instead of using a single CSLA. CSLAs with increasing sizes are connected in ascending order to extract the maximum concurrence in the path of carry propagation. SQRT-CSLAs design can be used for implementing large size adders with less delay than by using single CSLA. But the carry generation delays between stages of SQRT-CSLAs are of critical

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology
Vol. 1, Issue 4, December 2014

consideration. As the carry generation is early in our system, the proposed CSLA design is more efficient than all other CSLA designs for the implementation of SQRT-CSLA. A SQRT-CSLA design of 16 bit CSLA is shown in the fig. here we considered a cascaded combination of a 2-bit RCA, 2- bit CSLA, 3-bit CSLA 4-bit CSLA, and 5-bit CSLA. Also we have considered a 32 bit SQRT-CSLA using a cascaded configuration of 2-bit RCA, 2-bit, 3-bit, 4-bit, 6-bit, 7-bit and 8-bit CSLAs

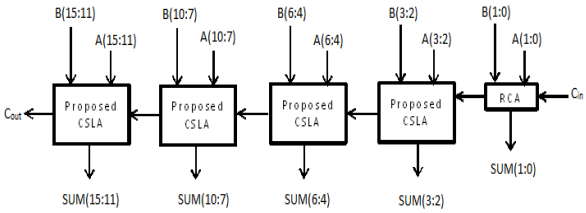

Fig 9: 16 bit square root carry select adder

IV. SIMULATION RESULTS

The proposed design is developed using VHDL and synthesized using XILINX 13.2i and is simulated in ISim for Spartan-3e FPGA series.

A. *Adder Output*

The output of optimized 32 bit carry select adder is shown in fig 10.

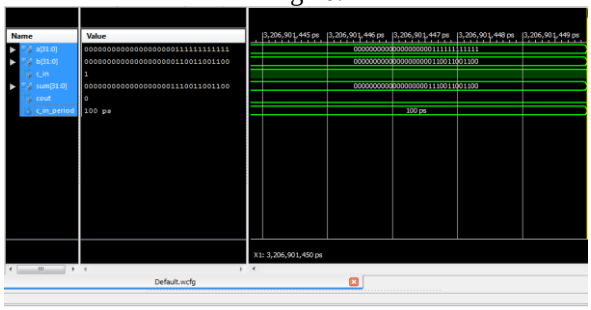

Fig 10: 32-bit CSLA Output

B. Area Utilization

By implementing the total system, the area consumed by the system is checked in the design summary. The area consumption is given in the terms of number of slices occupied. Fig 11 shows the area utilization of the proposed system in an FPGA.

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Number of 4 input LUTs	76	4,896	1%	
Number of occupied Slices	48	2,448	1%	
Number of Slices containing only related logic	48	48	100%	
Number of Slices containing unrelated logic	0	48	0%	
Total Number of 4 input LUTs	76	4,896	1%	
Number of bonded IOBs	98	158	62%	
Average Fanout of Non-Clock Nets	2.17			

Fig 11: Area Utilization of 32-bit optimized carry select adder

C. Power Consumption

The power consumption is the total power consumed by the system. The power consumption is also reduced. The power consumption is taken as a sum of leakage power and dynamic power. Thus the total consumed power can be obtained. Fig 12 shows the power consumption of the proposed system.

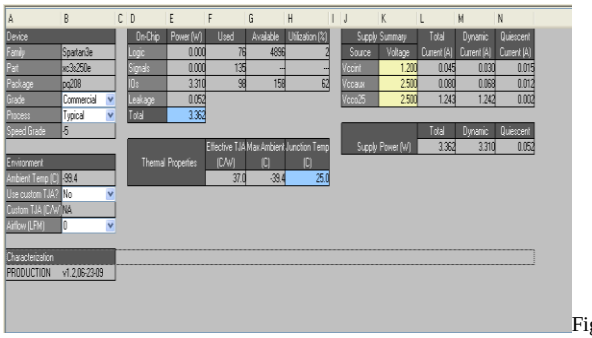

Fig 12: Power Consumption of 32-bit carry select adder

V. PERFORMANCE COMPARISON

The results shows that the power consumption, area and delay of the system have been reduced comparing to previous systems. Table1 shows the comparison of the area, power and delay between existing and proposed carry select adders. Also the simulation of the system using mux based gates instead of logic gates shows a reduction in area and power consumption than the system using logic gates in FPGA. Thus the carry select adder proposed is having reduced area, power and delay consumption than existing adders.

Table 1:Area and Power Consumption Comparison Of Existing and Proposed Carry Select Adders

Design	Bits	Area (No. Of Slices)	Power(nw)		
			Total	Dynamic	Quiescent
Area-delay-efficient carry select adder(12)	32	50	3.378	3.327	0.052
Optimized area-power and delay for efficient carry select adder	32	48	3.362	3.310	0.052

V. CONCLUSION

In this paper a new architecture for carry select adder using MUX based logic gates is proposed. Here the carry selection is done prior to the sum generation and also area and power consumption is reduced, which makes it most suitable for SQRT-CSLA for large size adders. The comparisons and results show that the proposed system has an optimum area and power performance along with lesser delay than all other existing carry select adders. Thus a carry select adder capable of substituting all the present adders in terms of optimized area, power and delay is proposed.

REFERENCES

[1] Weinberger, J.L. Smith, "A Logic For High-Speed Addition", National Bureau Of Standards, Circulation 591, P. 3-12, 1958
[2] T. Kilburn, D.B.G. Edwards and D. Aspinall, Parallel Addition In Digital Computers: A New Fast "Carry" Circuit, Proc. Iee, Vol.106, Pt.B. P.464, September 1959
[3] O. J. Bedrij, "Carry-Select Adder," Ire Trans. Electron. Comput., Vol. Ec-11, No. 3, Pp. 340–344, Jun. 1962.
[4] Tyagi,A."A Reduced Area Scheme For Carry-Select Adders" Computer Design: Vlsi In Computers And Processors, 1990. Iccd '90. Proceedings, 1990 Ieee International Conference, September 1990
[5] T. Y. Ceiang And M. J. Hsiao, "Carry -Select Adder Using Single Ripple Carry Adder," Electronics Letters, Volume:34 , Issue: 22 ,Page No:2101-2103, Oct 1998
[6] 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
[7] Amelifard B, Fallah F And PedramM "Closing The Gap Between Carry Select Adder And Ripple Carry Adder: A New Class Of Low-Power High-Performance Adders", Quality Of Electronic Design, 2005. Isqed 2005. Sixth International Symposium,Page No:148-152, March 2005
[8] Y. He, C. H. Chang, And J. Gu, "An Area-Efficient 64-Bit Square Root Carryselect Adder For Low Power Application," Inproc. Ieee Int. Symp. Circuits Syst., 2005, Vol. 4, Pp. 4082–4085
[9] B. RamkumarAnd H. M. Kittur, "Low-Power And Area-Efficient Carry-Select Adder,"Ieee Trans Very Large Scale Integr. (Vlsi) Syst., Vol. 20, No. 2, Pp. 371–375, Feb. 2012.
[10] I.-C. Wey C.-C.Ho, Y.-S. Lin, And C.C. Pang, "An Area-Efficient Carry Select AdderDesign By Sharing The Common Boolean Logic Term," Inproc.Imecs, 2012, Pp. 1–4.
[11] S. ManjuAnd V. Sornagopal, "An Efficient Sqrt Architecture Of Carry Select Adder Design By Common Boolean Logic," In *Proc. VlsiIcevent*, 2013, Pp. 1–5.
[12] Basant Kumar Mohanty, AndSujit Kumar Patel "Area–Delay–Power Efficient Carry-Select Adder" IEEE Transactions On Circuits And Systems—Ii: Express Briefs, Vol. 61, No. 6, June 2014