An Area Efficient Carry Select Adder for Signal Processing Applications

M.Ramalingam[1], T.Rajesh[2]

[1]II M.E (Embedded System Technologies), PSN College of Engineering and Technology, Tirunelveli, Tamilnadu, India

[2]Asst. Professor, Department of ECE, PSN College of Engineering and Technology, Tirunelveli, Tamilnadu, India

Abstract—*In this concise, the logic operations indeed with Carry Select Adder (CSLA) and binary to exces-1 converter (BEC) to avoid the existing logic operations and to achieve less Area delay product (ADP) by using carry select (CS) operation in the proposed system and identify the data dependence. It also involves less area and delay than the BEC based CSLA. The Ripple Carry adder is mainly used with Add one circuit called Multiplexer to find out and select the correct output to the system. Here the square root CSLA method is also used using different bit widths. The application-specified integrated circuit (ASIC) synthesis is used here to show the results of the system. We Use optimized logic units for the design of an efficient CSLA.*

Keywords— ASIC, Carry Select Adder, Area efficient

I. INTRODUCTION

An adder is the main component of an arithmetic unit. A complex digital signal processing (DSP) system involves several adders. The CS method is used mainly to reduce the Carry Propagation Delay (CPD). The Ripple carry adder (RCA) plays a major role here. In a SQRT CSLA, CSLAs with increasing size are connected in a cascading structure.

A CSLA based on common Boolean logic (CBL) is also proposed and it involves less logic source and it must be having longer CPD. So, the SORT-CSLA is used and the proposed CSLA design involves nearly 32% less ADP and consumes 33% less energy than that of the corresponding SQRT-CSLA.

These are mainly used in portable mobile devices, wireless receivers and bio medical instrumentations. We observe that logic optimization largely depends on availability of redundant operations in the formulation. However CSLA is not area efficient it uses multiple pairs of RCA to generate partial sum and carry by considering Cin=0 & Cin=1, then final sum and carry are selected by using multiplexer process.

The basic concept for consuming area, delay and power is by reducing number of gates used in the system.

II. FORMULATION OF LOGIC OPERATIONS

In this logic formulation operations, there are two units are obtained. Carry Generation (CG) unit and Carry Selection (CS) unit. A and B are the two operands with n bit-widths and generates the Half sum generation (HSG) unit and obtains the sum word and carry word.

The Input must be send through the carry selection unit and we obtained the output from the same area unit with the help of Full sum generation (FSG) unit.

The carry generator unit is composed of two CGS (CG_0 and CG_1) with sum word S_0 and carry word C_0. This must be showed detailed in the below figure (a).

Two *n*-bit operands are added in the conventional CSLA, then RCA-1 and RCA-2 generate *n*-bit *sum* (*s*0 and *s*1) and output-carry (*c*0 out and *c*1 out) corresponding to input-carry (*c*in = 0 and *c*in = 1), respectively. Logic expressions of RCA-1 and RCA-2 of the SCG unit of the *n*-bit CSLA are given.

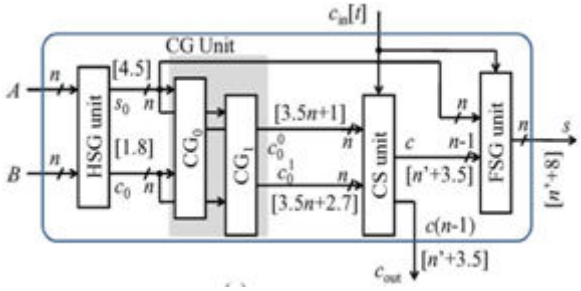

Fig: Carry generation and Selection Unit

III. CARRY SELECT ADDER OPERATION

The carry-select adder generally consists of two Ripple Carry adders and Multiplexer with Adding two n-bit

numbers with a carry-select adder is done with two adders (therefore two ripple carry adders) in order to perform the calculation twice, one time with the assumption of the carry being zero and the other assuming one. After the two results are calculated, the correct sum, as well as the correct carry, is then selected with the multiplexer once the correct carry is known.

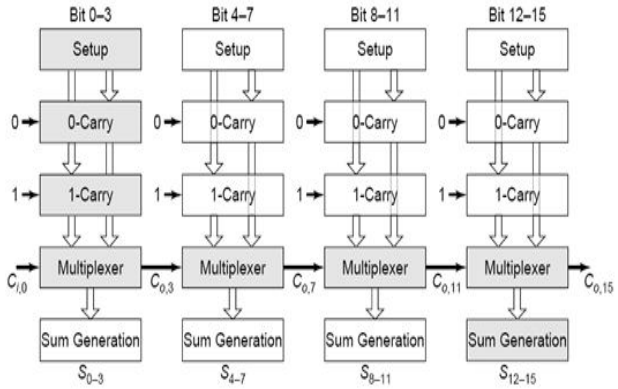

Fig 1. CSLA Process

The number of bits in each carry select block can be uniform, or variable. In the uniform case, the optimal delay occurs for a block size of root n. When variable, the block size should have a delay, from addition inputs A and B to the carry out, equal to that of the multiplexer chain leading into it, so that the carry out is calculated just in time. The delay is derived from uniform sizing, where the ideal number of full-adder elements per block is equal to the square root of the number of bits being added, since that will yield an equal number of MUX delays.

The carry-select adder design can be complemented with a Carry look Ahead adder structure to generate the MUX inputs, thus gaining even greater performance as a parallel prefix adder while potentially reducing area.

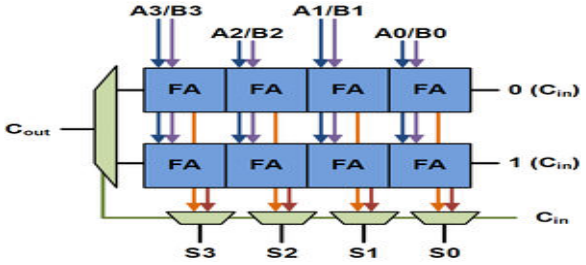

Fig 2. Basic Carry Select adder circuit

IV. AREA DELAY ESTIMATION METHOD

The delay and area evaluation methodology considers all gates to be made up of AND, OR, and Inverter, each having delay equal to 1 unit and area equal to 1 unit. The area evaluation is done by counting the total number of AOI gates required for each logic block. Based on this approach, the CSLA adder blocks of 2:1mux, half adder (HA) and Full adder (FA) are evaluated and listed in Table I.

TABLE I

DELAY AND AREA COUNT OF THE BASIC BLOCKS OF CSLA

Adder Blocks	Delay	Area
XOR	3	5
2:1 Mux	3	4
HA	3	6
FA	6	13

V. BEC-1 & SQRT BASED CSLA OPERATION

The Binary to Excess-1 converter increases the data dependence in CSLA and it reduces the area and power consumption instead of RCA with Cin=1. This process must be having higher delay.

The SQRT-CSLA implements large bit-width address with less delay. Increasing size are connected in cascade structure and it provides parallel path for carry propagation which helps to reduce over all adder delay.

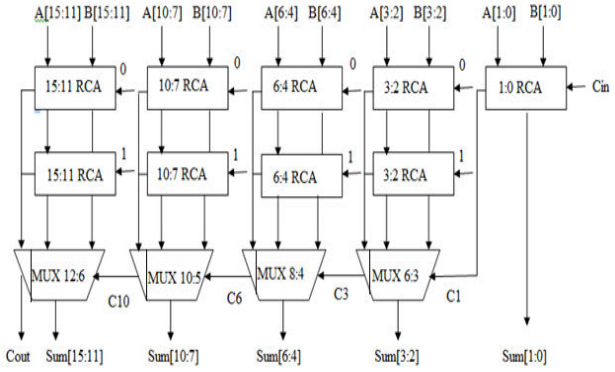

Fig 3. Regular 16-bit SQRT CSLA

It is also known as non-linear CSLA and it is used for equalizing the delay through two carry chains and block MUX signal from the previous stage.

The existing modified SQRT-CSLA uses BEC-1 instead of RCA with Cin=1 to achieve lower delay with slightly increase in area.

In the N-bit carry ripple adder, the delay time can be expressed as:

$$TCRA = (N-1)\,T\,carry + T\,sum \qquad (1)$$

In the N-bit carry select adder, the delay time is:

TCSA = T setup + (N/M) T carry + M T mux + T sum (2)

In our proposed N-bit area-efficient carry select adder, the delay time is:

T new = T setup + (N-1) T mux + T sum (3)

In the case of the BEC-based CSLA, c^1_1depends on s^0_1, which otherwise has no dependence on s^0_1 in the case of the conventional CSLA. The BEC method therefore increases data dependence in the CSLA. We have considered logic expressions of the conventional CSLA and made a further study on the data dependence to find an optimized logic operation for the CSLA.

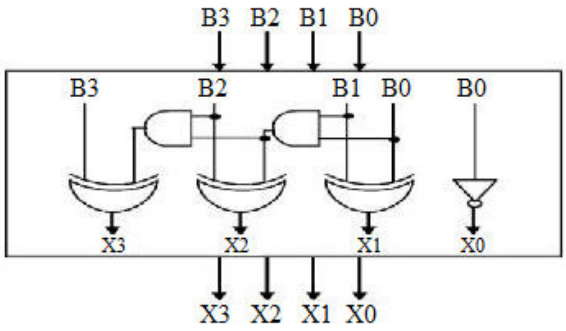

Fig 4. 4-bit Binary to Excess-1 Converter

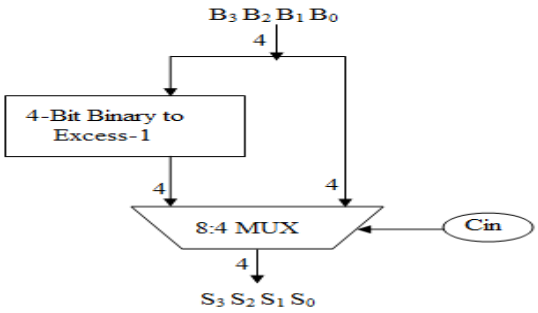

Fig 5. 4-bit Binary to Excess-1 logic with 8:4 multiplexer

VI. COMMON BOOLEAN LOGIC

The main aim of this process is used to remove the duplicated adder cells in CSLA and it saves many transistor counts and thus achieves the low power.

Once Cin signal is ready, it select Cout signal according to logic state of Cin signal. This method replaces BEC add one circuit by CBL.

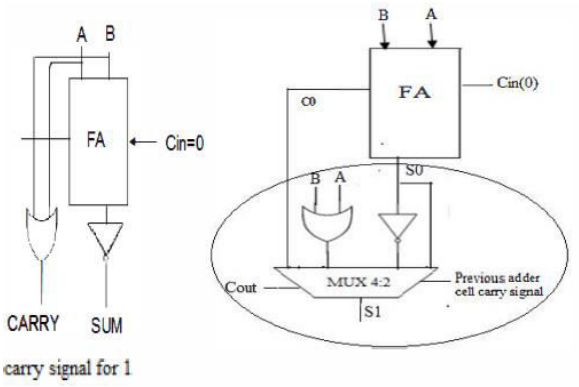

Fig 6. Single bit FA with Common Boolean Logic

The CBL based SQRT-CSLA requires more logic resource and delay than BEC based SQRT-CSLA. Through analyzing the truth table of a single-bit full adder, To find out that the output of summation signal as carry-in signal is logic "0" is the inverse signal of itself as carry-in signal is logic "1 ".

To share the common Boolean logic term, it only needs to implement one OR gate with one INV gate to generate the Carry signal and summation signal pair. Once the carry-in signal is ready, then select the correct carry-out output according to the logic state of carry-in signal.

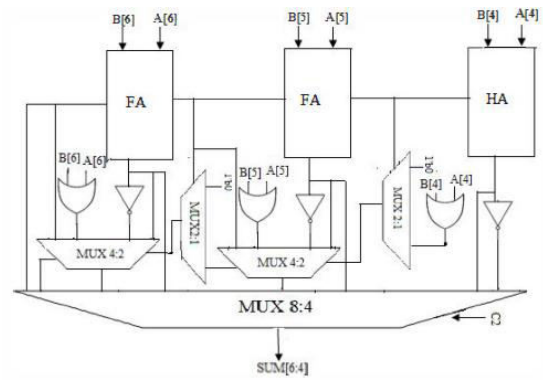

Fig 7. Group 3 using CBL.

The Summation and carry signal for FA which has Cin=l, Generate by INV and OR gate. Through the multiplexer, we can select the correct output result according to the logic state of carry-in signal.

VII. ASIC SYNTHESIS RESULTS

All the designs are synthesized in the Synopsys Design Compiler (DC).The net list file obtained from the DC are processed in the IC Compiler (ICC). The proposed SQRT-CSLA involves significantly less area and less delay and consumes less power than the existing designs. The proposed SQRT-CSLA design offers a saving of 39% ADP and 37% energy than the RCA-based conventional

SQRT-CSLA; 32% ADP and 33% energy than the BEC-based SQRT-CSLA.

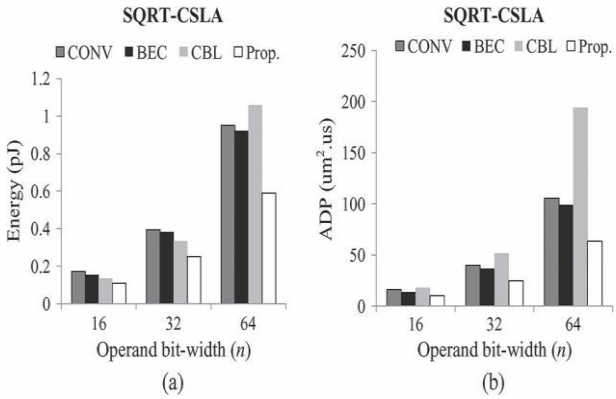

Fig 8. (a) Comparison of energy consumption. (b) Comparison of ADP

TABLE V Comparison of Adders for Delay, Area and Power

Word size	Adder	Area(No. of Gate count)	Delay (ns)	Power (mw)
8-bit SQRT CSLA	**Regular** (dual RCA)	144	11.92	193
	Modified (with BEC)	132	13.69	180
	Proposed (with CBL)	111	11.15	119
16-bit SQRT CSLA	**Regular** (dual RCA)	348	16.15	315
	Modified (with BEC)	291	18.77	268
	Proposed (with CBL)	276	15.48	177
32-bit SQRT CSLA	**Regular** (dual RCA)	698	28.97	553
	Modified (with BEC)	762	34.44	448
	Proposed (with CBL)	552	26.23	321
64-bit SQRT CSLA	**Regular** (dual RCA)	1592	52.82	860
	Modified (with BEC)	1498	64.61	745
	Proposed (with CBL)	1104	47.74	555

VIII. CONCLUSION

The data dependence and redundant logic operations are analyzed in the BEC based and conventional CSLA and eliminate all the redundant logic operations by using the formulation of CSLA operation process. Thus the CS operation is done before the calculation of final sum and the carry words must be generated by CSLA follows specific bit pattern. Due to the small carry output delay, the proposed CSLA design is a good candidate for the SQRT adder. The ASIC synthesis result shows that the existing BEC-based SQRT-CSLA design involves 48% more ADP and consumes 50% more energy than the proposed SQRTCSLA, on average, for different bit-widths.

REFERENCES

[1] K. K. Parhi, VLSI Digital Signal Processing. New York, NY, USA: Wiley, 1998. P. Chandrakasan, N. Verma, and D. C. Daly, "Ultralow-power electronics for biomedical applications," Annu. Rev.Biomed. Eng., vol. 10, pp. 247–274, Aug. 2008.

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

[3] 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.

[4] 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.

[5] B. 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, pp. 371–375, Feb. 2012.

[6] 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.