RESEARCH ARTICLE OPEN ACCESS

Implementation and Comparison of Efficient 16-Bit SQRT CSLA Using Parity Preserving Reversible Gate

Rupali Patel*, M.A Khan**
**(PG Student, VLSI and Embedded, LKCT Indore)*
*** (Assistant Professor, Department of Electronics and Communication, LKCT Indore)*

ABSTRACT
In Very Large Scale Integration (VLSI) outlines, Carry Select Adder (CSLA) is one of the quickest adder utilized as a part of numerous data processing processors to perform quick number crunching capacities. In this paper we proposed the design of SQRT CSLA using parity preserving reversible gate (P2RG). Reversible logic is emerging field in today VLSI design. In conventional circuits, the logic gates such as AND gate, OR gate is irreversible in nature and computing with irreversible logic results in energy dissipation. This problem can be circumvented by using reversible logic. In ideal condition, the reversible logic gate produces zero power dissipation. The proposed design is efficient in terms of delay as compare to irreversible SQRT CSLA. The simulation is done using Xilinx.
***Keywords*:** Adder/Subtactor, SQRT CSLA, parity preserving gate, P2RG, reversible logic

I. INTRODUCTION

The binary addition is the fundamental math operation in digital circuits and it got to be crucial in a large portion of the computerized frameworks including Arithmetic and Logic Unit (ALU), microprocessors and Digital Signal Processing (DSP). Outline of range and control proficient rapid information way rationale frameworks are a standout amongst the most significant territories of exploration in VLSI framework plane. In digital adder speed of addition is restricted when required to propagate a carry through the adder. The total addition of each and every bit position in an elementary adder is generated sequentially only after the previous bit position has been summed and carry propagate into the next position. The CSLA is utilized as a part of numerous computational frameworks to reduce the issue of carry propagation delay by freely producing numerous carries and then select a carry to generate the sum [1][2]. In fastly developing electronic industry, faster units are of concern toward outline as well as littler region and less power get to be real attentiveness towards configuration of VLSI circuits. So a VLSI optimize area, delay and power constraints for expanding convey ability and battery life of compact gadget.

In CMOS technology the pace of expansion is constrained by increasing heat dissipation in silicon chips. In VLSI circuits almost all conventional circuits comprise million numbers of gates (i.e. AND gate, OR gate) that are irreversible in nature. During computation some information is lost due to this heat dissipation and energy loss is occurs. According to R Landauer [3], if the manufacturing material and factor of technology is not considered then the energy loss is only occurred by the irreversible logical operations. The circuits with irreversible components, during computation each bit of information loss generates kTln2 joules of energy, where k is Boltzmann's constant and T is absolute temperature. As the heat dissipation of circuit is increases, the performance and life of the circuit is decreases. This problem can be overcome by using the components which offers low power consumption and less dissipation. According to C H Bennet [4], we can achieve the zero energy dissipation in the circuits if we use reversible logic gate in place of irreversible logic components in the circuits.

II. PARITY PRESERVING REVERSIBLE GATE

Parity preserving is the property that will enables the system to continue its operation properly when failure occurs in any of the component. The error detection and correction will be easier if the system will be made by fault tolerance components. Parity checking is most commonly used method for error detection in digital logic circuits. It will most commonly used in arithmetic and other processing systems because those systems do not preserve the parity of the data, there have been attempts at performing arithmetic operations on specially encoded operands in a way to check the parity. These types of methods will require more development and they are rarely used. B Parhami [5] shown some methods of error detection in reversible circuits, those standard methods of error detection will present some problems because in reversible logic circuits fan out are not allowed and we have to take care of garbage bits. For parity preserving

output data, the data can be checked in manner i.e. in a computational critical path. For parity preserving gate the ex-or of input will matches with the ex-or of output. For a 4*4 reversible gate it will satisfies the condition of A ⊕B ⊕ C ⊕ D = P ⊕ Q ⊕ R ⊕ S. Where A, B, C and D are gate inputs and P, Q, R and S are gate outputs.

Some parity preserving reversible gates are as follows:

Feynman Double Gate (F2G):

Fig. 1 shows 3*3 Feynman Double Gate (F2G) [6]. It has A, B and C input vector and output vector as P = A, Q = A⊕B, and R = A⊕C.

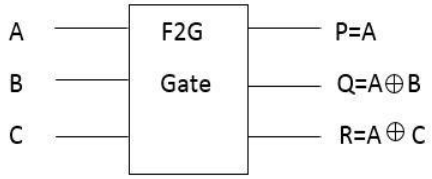

Fig. 1. Feynman Double Gate (F2G)

Fredkin Gate (FRG):

Fig. 2 shows 3*3 Fredkin gate (FRG) [7]. It has A, B and C input vector and output vector as P = A, Q= A'B⊕ AC and R = A'C⊕ AB.

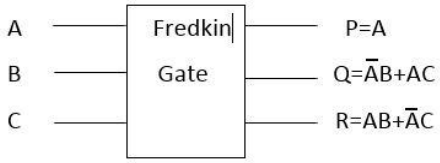

Fig.2. Fredkin Gate (FRG)

Modified IG Gate (MIG):

Fig. 3 shows 4*4 Modified IG [8] gate. It has A, B, C and D input vector and output vector as P = A, Q = A⊕ B, R = AB ⊕C and S = AB'⊕D.

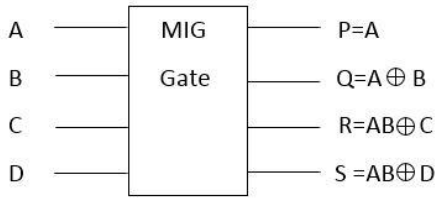

Fig. 3. Modified IG Gate (MIG)

Parity Preserving Reversible Gate (P2RG):

Fig.4 shows 5*5 parity preserving reversible gate [9], P2RG. It has A, B, C, D and E input vector and output vector as P=A, Q= (A'C'⊕B') ⊕D, R= (A'C'⊕B') D⊕AB⊕C, S=AB'⊕C⊕ (A'C'⊕B')'D and T= (D⊕E) ⊕AC.

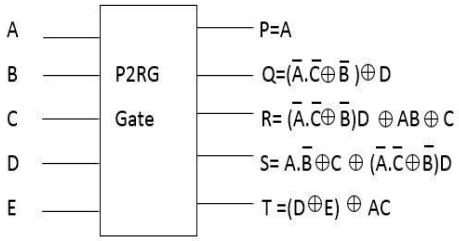

Fig.4. Parity Preserving Reversible Gate (P2RG)

III. IRREVERSIBLE SQRT CSLA

According to Basant Kumar [1] The square-root carry select adder is designed by making the delay equal via two carry chains and the multiplexer signal from previous stage. It is also called as non-linear carry select adder. The square-root (SQRT) CSLA to execute huge bit-width adders with minimum delay. In a SQRT CSLA, CSLAs with expanding size are associated in a falling structure. The primary goal of SQRT-CSLA outline is to provide a parallel way for carry propagation that decreases the in general adder. The CSLA is having two units:

1) The sum and carry generator unit (SCG)
2) The sum and carry selection unit.

The SCG unit consumes maximum logic resources of CSLA and mainly contributes to the critical path. Different logic designs were suggested for efficient implementation of the SCG unit. the BEC based CSLA [10], made of one RCA unit and one BEC unit as shown in fig5.

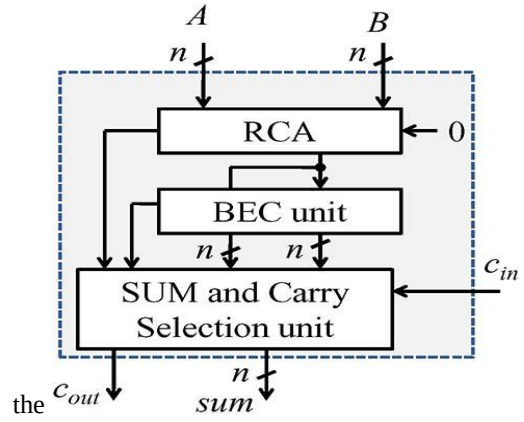

Fig. 2. Structure of the BEC-based CSLA; *n* is the input operand bit-width.

$$s_0 = A(i) \oplus B(i) \qquad c_0(i) = A(i).B(i) \tag{1a}$$
$$c_1^0 = c_1^0(i-1).s_0(i) + c_0(i) \quad for\ (c_1^0\ (0) = 0) \tag{1c}$$
$$c_1^1 = c_1^1(i-1).s_0(i) + c_0(i) \quad for\ (c_1^1\ (0) = 1) \tag{1c}$$
$$c(i) = c_1^0\ (i)\ \ if(c_{in} = 0) \tag{1d}$$
$$c(i) = c_1^1\ (i)\ \ if(c_{in} = 1) \tag{1e}$$
$$c_{out} = c(n-1) \tag{1f}$$
$$s(0) = s_0(0) \oplus c_{in} \qquad s(i) = s_0\ (i) \oplus c(i-1) \tag{1g}$$

The logical operation of n-bit RCA is performed in four stages: 1) HSG unit, 2) FSG unit,

3) CG unit, and 4) CS unit. The logic formulas are given in equation (1a)–(1g), and structure is shown in Fig. 6(a). The CG unit is made of two CGs (CG_0 and CG_1) 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 CG_0 and CG_1 receive s_0 and c_0 from the HSG unit and generate two n-bit full-carry words c_0^1 and $c^1{}_1$ corresponding to input-carry '0' and '1', respectively. The logic diagram of the HSG unit is shown in Fig. 6(b). The optimized designs of CG_0 and CG_1 are shown in Fig. 6(c) and (d), respectively. 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_0^1 when c_{in} =0; otherwise, it selects $c^1{}_1$. The CS unit can be implemented using an n-bit 2-to-l MUX. The optimized design of the CS unit is shown in Fig. 6(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 Fig. 6(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.

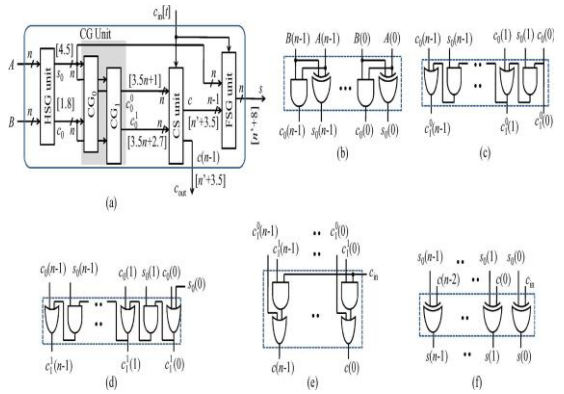

Fig. 6. (a) Proposed CS adder design, where *n* is the input operand bit-width, and [∗] represents delay (in the unit of inverter delay), $n = \max(t, 3.5n + 2.7)$. (b) Gate-level design of the HSG. (c) Gate-level optimized design of (CG0) for input-carry = 0. (d) Gate-level optimized design of (CG1) for input-carry = 1. (e) Gate-level design of the CS unit. (f) Gate-level design of the final-sum generation (FSG) unit.

IV. PROPOSED WORK

The proposed CSLA is design by using reversible logics. The design of reversible SQRT CSLA is same as irreversible SQRT CSLA the difference is only that the logic gates are reversible in nature. We used the 5*5 parity preserving reversible gates in place of irreversible gates (AND gate, OR gate and XOR gate). The logic formulations are same as in equation (1a-1g). The logic diagram for HSG unit consist n-bit parallel adder. So we are design the parity preserving half adder using one P2RG gate and one Fredkin gate as shown in fig 7(a). This design has two inputs A and B and a control line Ctrl which will control mode of operation, i.e. when Ctrl is at logic 0, the circuit will act as adder and when ctrl is at logic 1, the circuit will act as subtractor. It will give three constant inputs and four garbage bits g1 to g4. Boolean expressions to realize the functionality of half adder and half subtractor are given below:

Sum/Difference = A ⊕ B
Carry = A.B
Borrow = A'. B

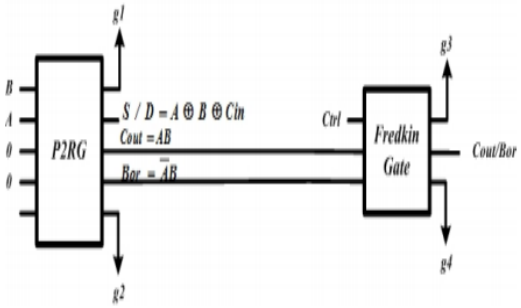

Fig. 7(a) Parity Preserving Full Adder/Subtractor using P2RG gate.

The logic diagarm of AND gate using P2RG is shown in fig 7(b). when input B= D=E=0 then output T performs AND operation i.e. (A.B). The logic diagram of reversible OR gate is realized using one P2RG gate and one Fredkin gate as shown in fig 7(c). when input B=1 and D=E=0 then output Q performs XOR operation at P2RG gate. This XOR out is applied to the NOT gate and output of the NOT gate performes OR operation i.e. (A+C). Not gate is also a reversible gate because it has equal number of input and output.

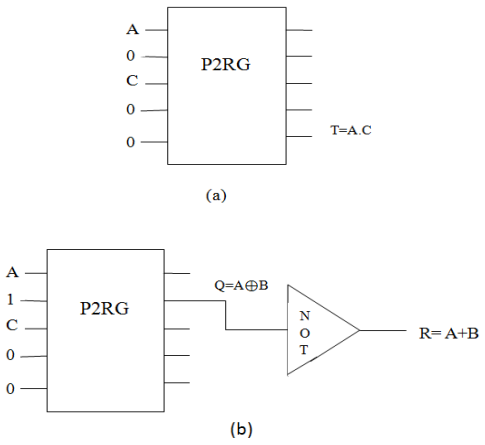

Fig.8: Reversible (a) AND Gate (b) OR Gate

V. RESULT

The 16-bit reversible SQRT CSLA is designed using VHDL (Very High Speed Integration Hardware Description Language). The coding is

done on Xilinx ISE 14.7 on Spartan 3 using target device: XC3S50-PQ208 at speed grade of -5. Simulation can be done using ModelSim SE 10.4a simulator. The simulation results is shown in fig 9.

Fig. 9(a). Simulation result of reversible SQRT CSLA at cin=0

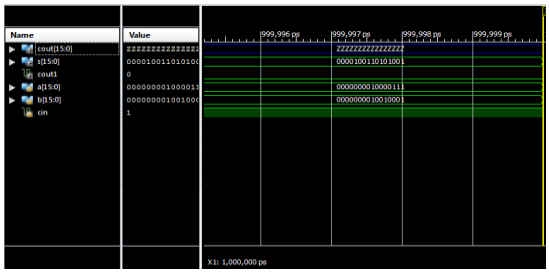

Fig. 9(b). Simulation result of reversible SQRT CSLA at cin=1

Table 1 shows the comparison of 16-bit reversible SQRT CSLA and 16-bit irreversible SQRT CSLA. Table 2 shows the comparative results of parity preserving 16-bit parallel adder and 16-bit SQRT CSLA.

Table I Comparative results of 16-bit reversible SQRT CSLA and 16-bit irreversible SQRT CSLA

16-bit SQRT CSLA		
	Reversible	**Irreversible**
LUT's	32	27
Slices	16	14
Bonded IOB's	50	50
Gate Count	188	269
Delay (ns)	14.794	17.510

Table III Comparative results of 16-bit parallel adder and 16- bit SQRT CSLA using P2RG.

16-bit ADDER		
	SQRT CSLA	**Parallel Adder**
LUT's	32	32
Slices	16	16
Bonded IOB's	50	50
Gate Count	188	32
Delay (ns)	14.794	26.642

VI. CONCLUTION

This paper presents efficient way of designing the reversible SQRT CSLA. The proposed design demonstrates less hardware complexity and less gate count. Gate count reduction is a sign of area reduction. The proposed 16-bit SQRT CSLA offers less gate count and delay as compare to irreversible SQRT CSLA. When it is compare with the 16-bit parallel adder using P2RG it offers less delay but gate count is increases. For future to explore in this work is to design more optimized fault tolerant adder and other fault tolerant circuits in a way to reduce the area power and delay.

REFERENCES

[1]. Area–Delay–Power Efficient Carry-Select Adder Basant Kumar Mohanty, Senior Member, IEEE, and Sujit Kumar Patel IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS, VOL. 61, NO. 6, JUNE 2014

[2]. 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.

[3]. R. Landauer, "Irreversibility and heat generation in the computing process," IBM journal of research and development, vol. 5, no. 3, pp. 183–191, 1961.

[4]. C. H. Bennett, "Logical reversibility of computation," IBM journal of Research and Development, vol. 17, no. 6, pp. 525–532, 1973.

[5]. B. Parhami, "Fault-tolerant reversible circuits," in Signals, Systems and Computers, 2006. ACSSC'06. Fortieth Asilomar Conference on. IEEE, 2006, pp. 1726–1729.

[6]. P. Kaur and B. S. Dhaliwal, "Design of fault tolerant full adder/subtarctor using reversible gates," in Computer Communication and Informatics (ICCCI), 2012 International Conference on. IEEE, 2012, pp. 1–5.

[7]. E. Fredkin and T. Toffoli, "Conservative logic", Intl. Journal of Theoretical Physics, pp. 219-253, 1982.

[8]. P. NG and M. Anandaraju, "Design and synthesis of fault tolerant full adder/subtractor using reversible logic gates."

[9]. Ragani Khandelwal AND Sandeep Saini, "Parity Preserving Adder/Subtractor using a Novel Reversible Gate", 2015 Fifth International Conference on Communication Systems and Network Technologies IEEE 2015