

Design of Complex Adders and Parity Generators Using Reversible Gates

Soham Bhattacharya¹, Sourav Goswami², Anindya Sen³

^{1,2,3}*Electronics and Communication Engineering Department, Heritage Institute of Technology, Kolkata, India*

Abstract- This paper shows efficient design of an odd and even parity generator, a 4-bit ripple carry adder, and a 2-bit carry look ahead adder using reversible gates. Number of reversible gates used, garbage output, and percentage usage of outputs in implementing each combinational circuit is derived. The CLA used 10 reversible gates with 14 garbage outputs, with 50% percentage performance usage.

Keywords - Reversible Logic Gates, Parity Generator, Carry Look Ahead adder, Ripple Carry Adder, Garbage output.

I. INTRODUCTION

Million irreversible gates are required to design almost all conventional computers. In this age of nanometer semiconductor nodes, heat dissipation as well as power consumption plays a major role in maintaining the life time of the device. Usually less power consumption means more life of the working device. As a result, it reduces the quantum cost.

According to Landauer[1,2], the amount of energy dissipated for every irreversible bit operation is at least $KT \ln 2$ joules, where $K=1.3806505 \times 10^{-23} \text{m}^2 \text{kg}^{-2} \text{K}^{-1}$ (joule/Kelvin-1) is the Boltzmann's constant and T is the absolute temperature at which operation is performed. A circuit is reversible if the input is recoverable from the output. Reversible computing supports both forward and backward movement process as one generates inputs from the outputs. In this paper, we have designed ripple carry adder using reversible gates where each carry bit gets rippled to the next stage.

Properties of Reversible Computing are as Follows:

1. Reversible logic elements are needed for recovering the state of inputs from the outputs.
2. Reversible gates provides bijective mapping between input and output, without any feedback.
3. For each input pattern, there should be unique output pattern. Finally, the circuit obtained will be acyclic, i.e, feedback is there but Fan-out is not more than one.

Parameters to determine the complexity and performance of circuits[3,4] are the number of reversible gates, Garbage output, Quantum cost and Delay. Garbage output is defined as the number of unused outputs used in the implementation of the reversible gates. Quantum cost is defined as the total number of quantum gates in a given reversible circuit, where

the reversible gates are substituted by a cascade of basic quantum gates.

II. THE CONCEPT

Reversibility in computing implies that the information about computational states can never be lost and be used when needed [5]. In 1973, Bennett [6] showed that energy dissipation problem is avoided with circuits built using reversible logic gates. Reversible logic gates are represented in order of $n \times n$, i.e. if 3 inputs are assigned, 3 outputs will be implemented. Reversible gates provides of one-to-one mapping between input and output, without any feedback.

III. LITERATURE REVIEW

By Moore's law, after every eighteen months, the number of transistors got doubled. Next is the Dennard Scaling which states that as transistors got smaller their power density stays constant, so that their power use stays in proportion with area. It also states that with increasing current and voltage, parasitic also increases, which limits the increase of clock frequency of a circuit, which in turn causes the failure of this property. With increase in number of transistors there is increase of power generated which in turn causes excessive heat generation and as a result, operation time of the circuit gets reduced, mobility is decreased, feasibility cost of the circuit gets increased and at last, reliability gets reduced. Toffoli[7] characterized reversible logic in his 1980 work, stating "Using invertible logic gates, it is ideally possible to build a sequential computer with zero internal power dissipation." Draper proposed a logarithmic depth quantum carry look ahead adder [8], which demonstrates two versions of addition namely in place and out-of-place method of addition for n -bit numbers, but the design is not optimized in terms of gates, garbage outputs, quantum cost and delay. Thapliyal proposed a novel methodology for reversible carry look-ahead adder [9] where presentation of improved designs of both in-place and out of place designs of reversible carry look-ahead adder had been made.

IV. METHODS

Using reversible gates, the following devices are implemented:

- A. Parity Generators.
- B. 4-bit Ripple Carry Adder.
- C. 2-bit Carry Look Ahead Adder.

A. Parity Generators:

A 4X4 reversible HNG gate is used to realize even and odd parity generator circuit.

Expression for Even Parity Generator can be given as:

$$P = A \text{ xor } B \text{ xor } C \tag{1}$$

Expression for Odd Parity Generator can be given as:

$$Q = A' \text{ xor } B \text{ xor } C \tag{2}$$

When port 'D' is grounded, a reversible even parity generator can be implemented using a 4X4 HNG gate and the output can be derived from the output pin 'R'. The truth table of the reversible even parity generator using 4X4 HNG gate is shown in Fig. 1(B).

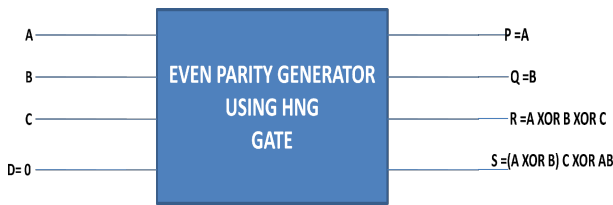


Fig. 1(A)

INPUT				OUTPUT			
A	B	C	D	P	Q	R	S
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	1	1	0
0	1	1	0	0	1	0	1
1	0	0	0	1	0	1	0
1	0	1	0	1	0	0	0
1	1	0	0	1	1	0	1
1	1	1	0	1	1	1	1

Fig. 1(B)

Fig. 1: (A) shows the block diagram of 3 bit reversible even parity generator using 4X4 HNG gate, where A,B,C,D are the input ports, R is the output even parity generator port of A,B,C. Lines P,Q,S are the garbage outputs. (B) shows truth table of 3 bit reversible even parity generator.

When port 'D' is grounded and port 'A' is connected to a NOT gate, a reversible odd parity generator can be implemented using 4X4 HNG gate and the output can be derived from the output pin 'R'. In both cases, lines P, Q, S are the garbage outputs. The truth table of the reversible odd parity generator using a 4X4 HNG gate is shown in Fig. 2(B).

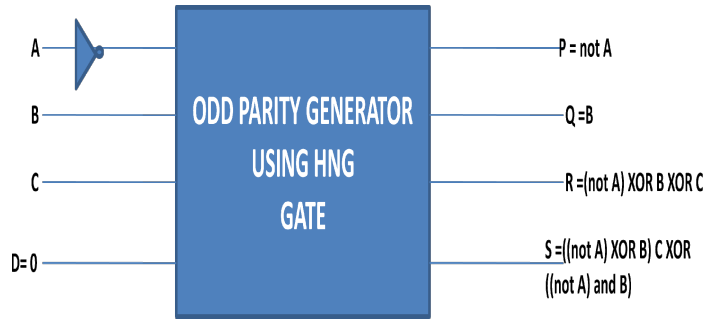


Fig. 2(A)

INPUT				OUTPUT			
A	B	C	D	P	Q	R	S
0	0	0	0	1	0	1	0
0	0	1	0	1	0	0	1
0	1	0	0	1	1	0	1
0	1	1	0	1	1	1	1
1	0	0	0	0	0	0	0
1	0	1	0	0	0	1	0
1	1	0	0	0	1	1	0
1	1	1	0	0	1	0	1

Fig. 2(B)

Fig. 2: (A) shows the block diagram of 3 bit reversible odd parity generator using 4X4 HNG gate, where A,B,C,D are the input ports, R is the output odd parity generator port of A,B,C. Lines P,Q,S are the garbage outputs. (B) shows truth table of 3 bit reversible odd parity generator.

B. Ripple Carry Adder:

The simplest form of a binary adder is a ripple carry adder. It can be of 2^p form, where $p = 0, 1, 2, 3, \dots$ up to n terms. Ripple carry adder is a combinational logic circuit used for the purpose of adding two n -bit binary numbers. 4-

Bit ripple carry adder is used for adding two 4-bit binary numbers. An n -bit ripple carry adder is used for adding two N -bit binary numbers [10]. In the design of a "n" bit ripple carry adder with reversible gates, each full adder can be replaced by a reversible HNG gate. A single bit full adder was designed using HNG gate [5].

Block Diagram of a 4-bit ripple carry adder is shown in Fig. 3(A). The truth table of the ripple carry adder is shown in Fig. 3(B). Using Reversible Gates, the 4-bit carry Look ahead adder has been implemented and shown in Fig. 3(C), using four 4X4 HNG gates. Here,

$$S_i = A_i \text{ xor } B_i \text{ xor } C_i \tag{3}$$

$$C_{i+1} = (A_i \text{ xor } B_i) \cdot C_i + (A_i \cdot B_i) \tag{4}$$

Assuming that, the full adder is implemented with the help of two half adders.

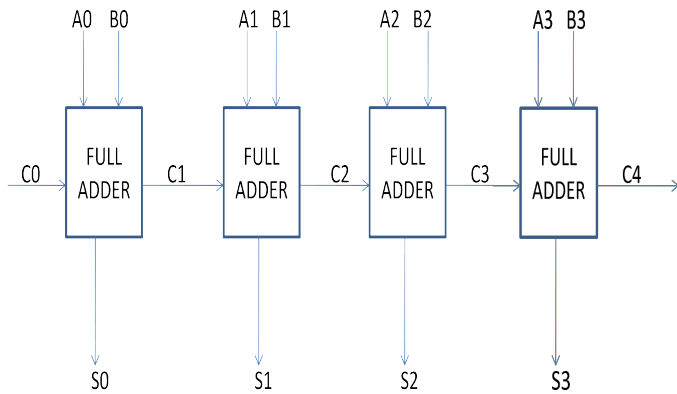


Fig. 3(A)

A	B	C	C _{i+1}	Sum
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Fig. 3(B)

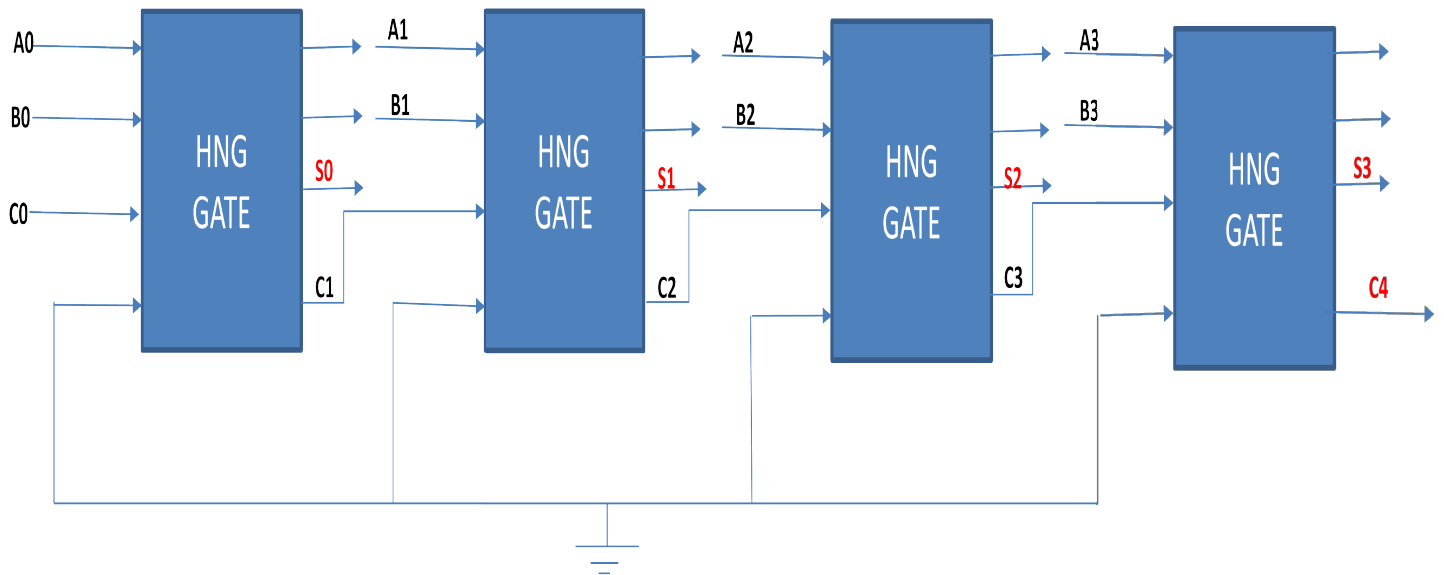


Fig. 3(C)

Fig. 3: (A) shows the block diagram of 4-bit ripple carry adder, with inputs A0 to A3, B0 to B3 and C0 as the carry input. S0 to S3 are the sum outputs and C4 is the carry output. C1 to C3 are the carry propagated from previous stage to next stage. (B) shows truth table of a single bit adder. (C) implements a 4-bit ripple carry adder using four HNG gates.

C. Carry Look Ahead Adder:

A carry look-ahead adder (CLA) reduces the propagation delay of the carry in ripple carry adder. Here, output C4 is a function of its inputs A3:A0, B3:B0, C0. For each stage, C1:C3 is also a function of the inputs. There is no propagation of carry [11].

a 4-bit CLA circuit.

Block diagram of a 4-bit CLA is shown in Fig. 4. Using reversible gate of type Peres, it is possible to implement

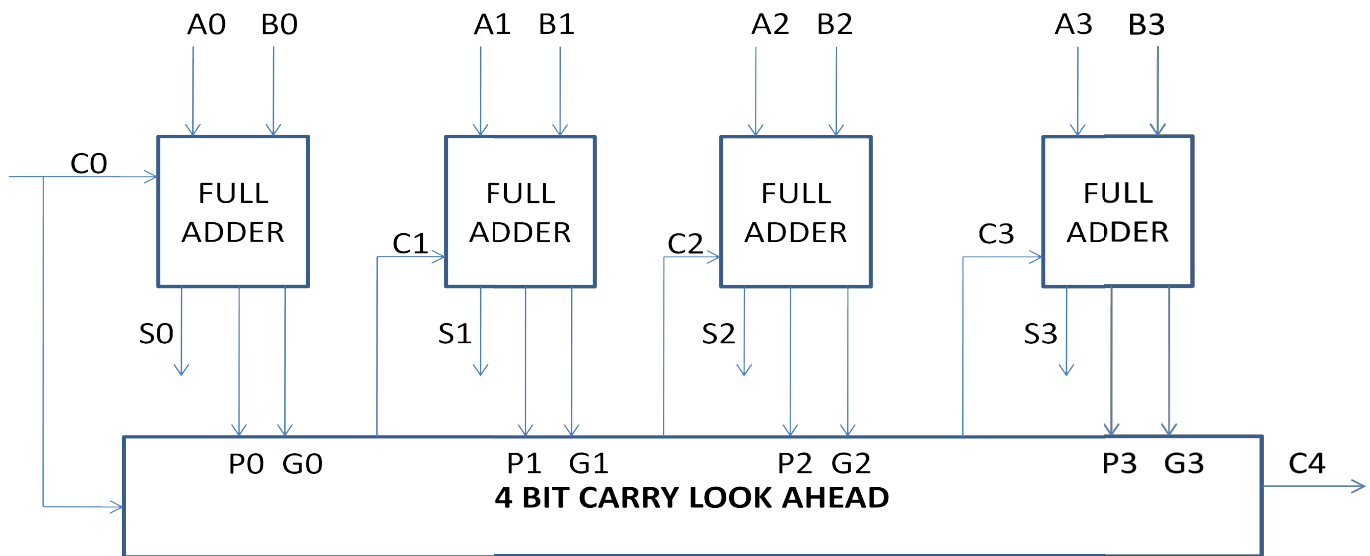


Fig. 4 shows the block diagram of 4-bit carry look ahead adder, where A0 to A3, B0 to B3, and carry in C0 are the inputs, S0 to S3 represent sum as outputs and C4 represents carry out as output.

Expression for carry generation and carry propagation in ripple carry adder is given as:

$$G_i = A_i B_i \quad (5)$$

$$P_i = A_i \text{ xor } B_i \quad (6)$$

Where, G_i is the carry generation term and P_i is the carry propagation term.

Expression for sum and carry out in ripple carry adder is given as:

$$S_i = P_i \text{ xor } C_i \quad (7)$$

$$C_{i+1} = G_i + P_i C_i \quad (8)$$

Where, S_i represents sum and C_{i+1} represents the carry out for the i 'th stage.

$$C_1 = G_0 + P_0 C_0 \quad (9)$$

Here, C_1 is the function of C_0 .

$$C_2 = G_1 + P_1 C_1 \quad (10)$$

Here, C_2 is the function of C_1 , but in eq. (11) and (12); C_2 is independent of C_1 and is the function of C_0 .

$$C_2 = G_1 + P_1(G_0 + P_0 C_0) \quad (11)$$

$$C_2 = G_1 + P_1 G_0 + P_1 P_0 C_0 \quad (12)$$

Similarly, for eq. (13) and (16), C_3 is the function of C_2 and C_4 is the function of C_3 respectively but in eq. (14) and (15), C_3 is independent of C_2 and is the function of C_0 and in eq.

(17) and (18), C_4 is independent of C_3 and is the function of C_0 .

$$C_3 = G_2 + P_2 C_2 \quad (13)$$

$$C_3 = G_2 + P_2(G_1 + P_1 G_0 + P_1 P_0 C_0) \quad (14)$$

$$C_3 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0 \quad (15)$$

$$C_4 = G_3 + P_3 C_3 \quad (16)$$

$$C_4 = G_3 + P_3(G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0) \quad (17)$$

$$C_4 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_0 \quad (18)$$

Let, us take an example,

$$C_{i+1} = G_i + P_i C_i \quad (19)$$

In Fig. 5, the design for eq.(19) can be implemented. But then C_{i+1} , a function of C_i and the final carry is not just dependent of the input variables, but also on the previous stage. Hence, this configuration cannot be used for CLA circuit.

For designing the 2-bit carry look ahead adder circuit, in eq. (9), for implementing carry out C_1 , 2 PERES gates (AND+OR) are required and for sum S_0 , one TKS gate (XOR) is required. In eq. (12), for implementing carry out C_2 , 5 PERES Gate (AND + OR) and 1 TKS gate (XOR) is required and for sum S_1 , 1 TKS gate (XOR) is required. So, total number of gates required to implement Carry Look Ahead adder is 10.

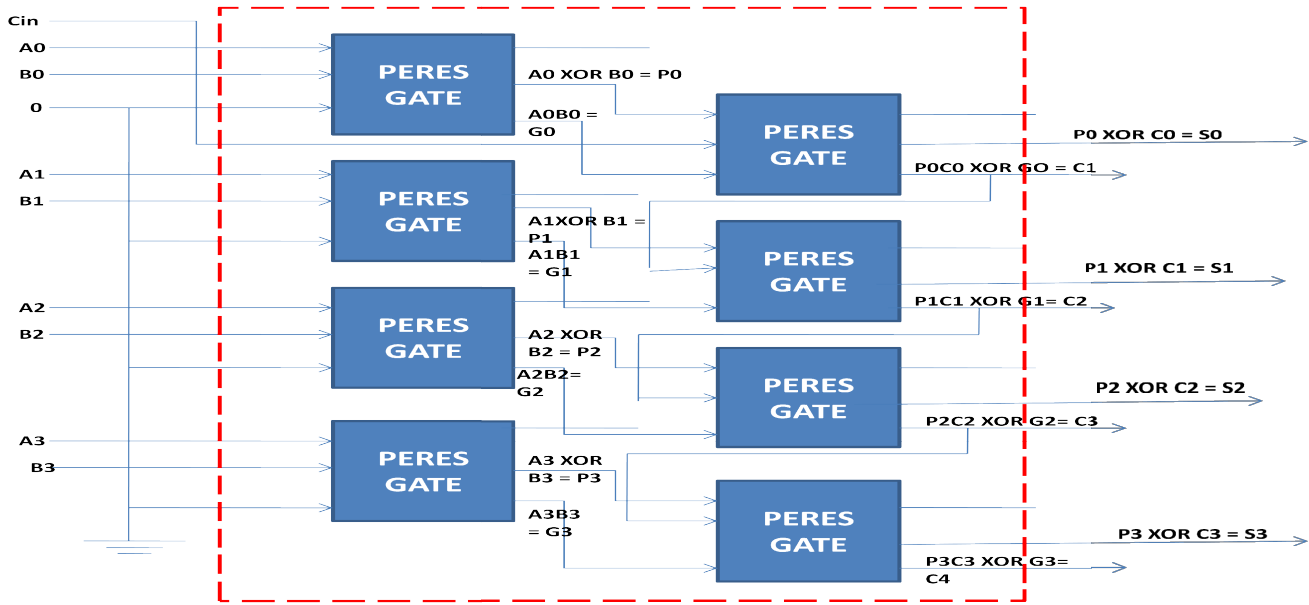


Fig. 5 shows the implementation of eq. (11) using reversible gates.

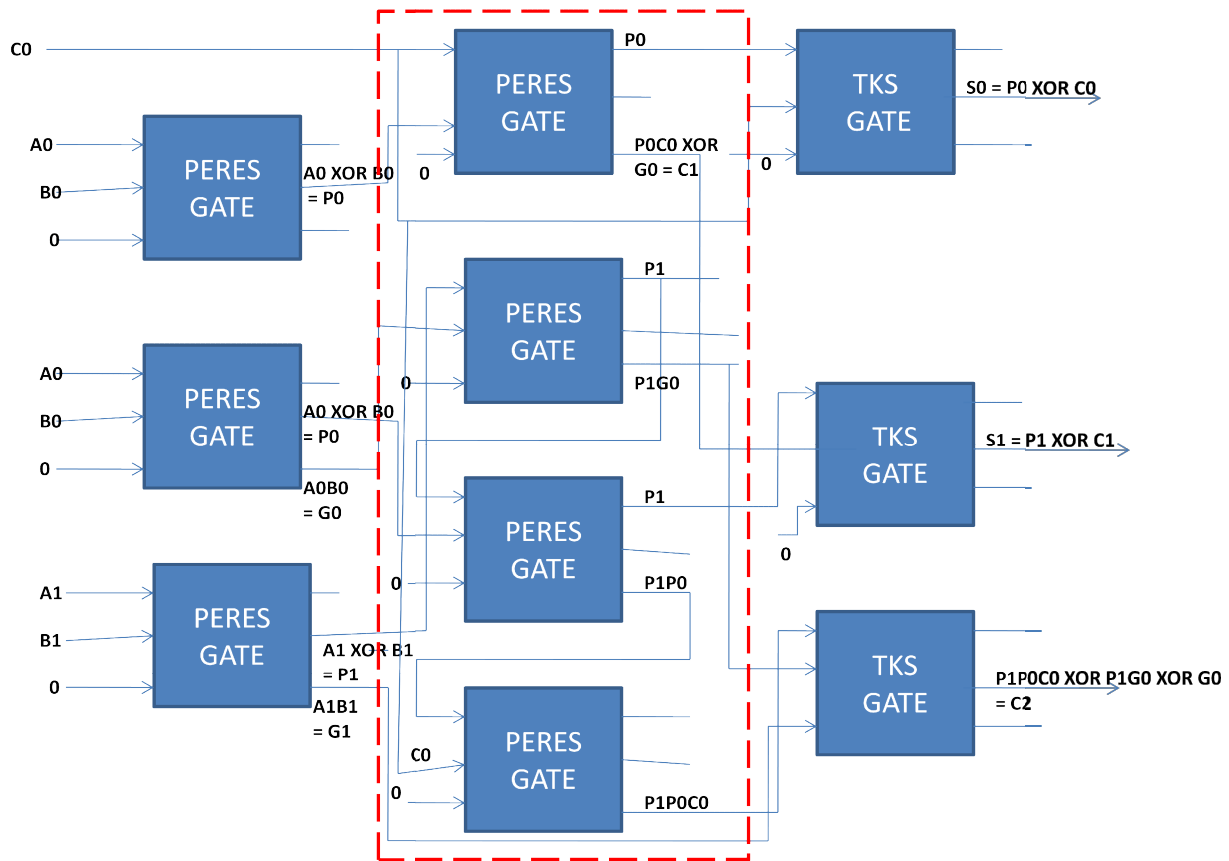


Fig.6 shows implementation of 2 bit carry look ahead adder, with inputs A_0 to A_1 , B_0 to B_1 and carry in C_0 and sum output S_0 to S_1 and carry output C_2 . CLA circuit generates C_1 and C_2 . Both C_1 and C_2 are only functions of the inputs.

V. PERFORMANCE ANALYSIS

Methods	Garbage outputs	Number of reversible gates	Percentage Usage of outputs
Even Parity Generator	3	1	25%
Odd Parity Generator	3	1	25%
4-bit ripple carry adder	8	4	50%
2-bit carry look ahead adder	15	10	50%

Fig. 7 shows the performance details of number of gates, garbage output and percentage usage of the outputs of the above implementations.

The table of Fig. 7 shows the performance of the implemented device using reversible gates in terms of number of gates, the garbage output and percentage usage of the outputs.

V. CONCLUSION

A processor that can only use 5% of its transistors at any given time will differ in properties from that other transistors which use 50%. Our aim will be to increase the percentage rate of usage of transistors with lowering of the heat dissipation. This paper presents an efficient approach for the designing of the parity generators, ripple carry adder and carry look ahead adder. From the table in Fig.7, for the design of 2-bit CLA, numbers of gates used are 10 and garbage outputs are 15. So, using less reversible gates, the carry look ahead adder can be implemented and so that the power consumption can be diminished and this design shows 50% improvement in performance of CLA. For the design of 4-bit ripple carry adder, even parity generator and odd parity generator, numbers of reversible gates used is 4, 1 and 1 respectively. Accordingly, garbage outputs are 8, 3 and 3 in respective order. So, one can use the best approach to design those combinational circuits from the table shown.

ACKNOWLEDGEMENT

We would like to express our gratitude to our Head of the Department of Electronics and Communication Engineering Department, Prof. (Dr.) Prabir Banerjee and VLSI Department Coordinator, Prof. Krishanu Dutta for their expert advice and encouragement, which helped us in preparing this manuscript. We would also like to express our sincere thanks to our families for their enormous support in our work.

REFERENCES

- [1] R.Landauer in "Irreversibility and Heat Generation in the Computing Process". IBM J. Research and Development,5(3): pp. 183-191, 1961.
- [2] R. Keyes, R. Landauer in "Minimal energy dissipation in logic", IBM J. Res. Dev. 14(1970) 152-157.
- [3] M. Mohammadi and M. Eshghi, "On figures of merit in reversible and quantum logic designs," Quantum Information Processing, vol. 8,no. 4, pp. 297-318, Aug. 2009.
- [4] D. Maslov and G. W. Dueck, "Improved quantum cost for n-bit toffoli gates" IEE Electronics Letters, vol. 39, no. 25, pp. 1790-1791, Dec.2003.
- [5] Soham Bhattacharya, Anindya Sen "Power and Delay Analysis of Logic Circuits Using Reversible Gates" International Journal of Latest Technology in Engineering, Management & Applied Science-IJLTEMAS vol.8 issue 12, December 2019, pp.54-63.
- [6] Charles H. Bennett , in "Logical Reversibility of computation", IBM Journal of Research and Development, vol. 17, no. 6, pp. 525-532, 1973.
- [7] T.Toffoli, Automata, Languages and Programming. Springer Verlag, 1980, ch. title: *Reversible Computing*, pp. 632-644.
- [8] Draper, T. G., Kutin, S. A., Rains, E. M., and Svore, K.M.: A logarithmic-depth quantum carry lookaheadadder. Quantum Information and computation.vol.6 No. 4&5, pp. 351-369 (2006).
- [9] H. Thapliyal, H. V. Jayashree, A. N. Nagamani, H.R. Arabnia, "Progress in Reversible Processor Design: A Novel Methodology for Reversible Carry Look-Ahead Adder", Springer Transactions on Computational Science XVII Lecture Notes in Computer Science Volume 7420, 2013, pp. 73-97.
- [10] Wikipedia on "Ripple Carry Adder".
- [11] NeelaShirisha, P.Kalyani, Dr.D.Nageshwar Rao in "Design of a Reversible Carry Look-Ahead Adder Using Reversible Gates".