Power and Delay Analysis of Logic Circuits Using Reversible Gates

Soham Bhattacharya, Anindya Sen
Electronics and Communication Engineering Department, Heritage Institute of Technology, Kolkata, India

***Abstract* - This paper determines the propagation delay and on chip power consumed by each basic and universal gates and basic arithmetic functions designed using existing reversible gates through VHDL. Hence a designer can choose the best reversible gates to use for any logic circuit design. The paper does a look up table analysis of truth tables of the reversible gates to find the occurrence of the AND OR, NAND, NOR and basic arithmetic functions, useful to build complex combinational digital logic circuits.**

***Keywords* — Reversible logic, Logic circuits, Power dissipation, Quantum cost, Timing analysis, On-chip power**

I. INTRODUCTION

Dark silicon[1] problem has imposed several challenges in VLSI technology with power dissipation and design implementation. Consequently application of reversible logic has received attention in the recent years to reduce the power dissipation with low power VLSI design, and it has a vast impact in low power CMOS, quantum computation and nanotechnology.

According to Landauer[2,3], the amount of energy dissipated for every irreversible bit operation is at least KTln2 joules, where K=1.3806505*10-23m2kg-2K-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 1973, Bennett[4] showed that energy dissipation problem is avoided with circuits built using reversible logic gates.

The 8086 micro-processor had less than 30,000 transistors, in comparison with current CPU and GPU's which have billions of transistors leading to complex management of cost, and power. One can have billions of transistors on a chip, but, all of those transistors cannot be used simultaneously. This fact has made a big difference in how CPUs are designed, and the problem will only increase in the future. A processor that can use only 5% of its transistors at any given time will vary with the processor which can use 50% of its transistors in several characteristics, mainly timing and power[1], hence researchers are considering the use of reversible gates. In software parallel programming research themuch progress is achieved but physical limitation is speed shall

II. THE CONCEPT

Reversibility in computing implies that the information about computational states can never be lost and be used when needed. Logical reversibility is the process in which any early stage can be recovered in computing backwards or un-computing the results. Physical reversibility refers to no energy dissipation of heat. After Physical, logical reversibility is achieved[5]. The generation of heat is gained when the voltage levels changes from low/high to high/low corresponding to zero/one to one/zero bit change in digital logic of computers. Most of the energy is required in making the changes. Hence reversible computing shall affect digital logic design. Reversible logic elements are needed for recovering the state of inputs from the outputs. Reversible logic gates are represented in order of n X 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. MOTIVATION/ PURPOSE

In this era of nanometer semiconductor nodes, the transistor scaling and voltage scaling are no longer in line with each other, resulting in the failure of the Dennard scaling[6], thus posing a severe design challenge.

High-performance chips releasing considerable amounts of heat have self imposed practical limitation on its performance. Reverse computing[7] offers a plausible solution to the heat problem faced by the chip manufacturers, and it has been extensively studied in the area of computer architecture. Power usage is closely related to heat dissipation and this work aims to lower heat dissipation so that heat related design problem can be handled more effectively by the chip manufacturers.

IV. BACKGROUND/ LITERATURE REVIEW

Toffoli[8] 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." By Moore's Law, for every eighteen months, the number of transistors will be doubled. For every eighteen months, it is possible to build higher performance general purpose processors. The catch is powering its transistors without melting the chip. The reasons of heat dissipation are as follows:

1. Increase in the number of transistors on chip.[9]
2. High Power dissipation leads to:

- Reduced time of operation.
- Reduced mobility.
- High efforts for cooling.
- Increasing operational costs.
- Reduced reliability.

V. REVERSE COMPUTING

'Reversible computing' is a model of computing where the computational process at some point, is reversible. That means, along with time, it can store the data as long as it requires to be used when necessary.

Reversible logic gate is an n- input and n- output device with one-to-one mapping. It determines outputs from the given inputs, and the inputs can also be recovered from the outputs[10,11]. Parameters to determine the complexity and performance of circuits[12,13] are:

A. The number of reversible gates (N): Number of reversible gates used in the circuit.
B. Garbage output**:** Number of unused outputs used in the reversible gates. One cannot avoid the garbage outputs as these are very essential to achieve reversibility.
C. Quantum cost : Cost of the circuit with respect to the cost of a primitive gate.

VI. MATERIALS AND METHODS

Materials:

The reversible gates were simulated in Xilinx Vivado under Windows 10, 64 bit OS using Intel® Core™ i3-5005U CPU @ 2.00GHz. This was used for the analysis of elapsed time and dissipated power.

Methods:

Using reversible gates Toffoli, Feynmann, Double Feynmann, Peres, Fredkin, HNG and TSG, the most economical implementation of basic gates and universal gates are shown in terms of propagation delay and power dissipation. An example circuit using above properties and the mapping from input to output is shown. Finally an reversible half adder, reversible half subtractor and a reversible full adder are implemented using the above mentioned reversible gates.

Quantum networks are required to be built from reversible logic components[14,15]. Truth tables of reversible gates are used to find the basic and universal gate function.

The block diagram of 1X1 NOT Gate is represented by Fig. 1(A).The quantum cost for 1X1 NOT Gate from Fig. 1(B) is 0. Total logical calculation is T=1δ. Here, δ represents a two input XOR gate calculation.

Fig. 1(A)

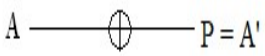

Fig. 1(B)

INPUT	OUTPUT
0	1
1	0

Fig. 1(C)

Fig.1. In this figure 1(A) shows block diagram of 1X1 NOT gate, 1(B) shows its quantum representation and 1(C) shows its truth table

The block diagram of 3X3 Toffoli Gate is represented by Fig. 2(A). The quantum cost of 3X3 Toffoli gate is 5 from Fig. 2(B). Total logical calculation is T = 1δ + 1β.[16]

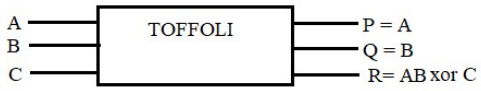

Fig. 2(A)

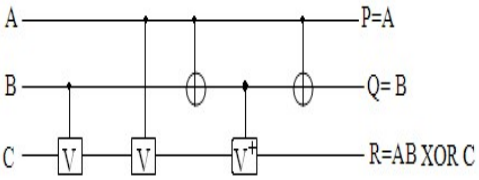

Fig. 2(B)

INPUT			OUTPUT		
A	B	C	P =A	Q = B	R = AB XOR C
0	0	0	0	0	0
0	1	0	0	1	0
1	0	0	1	0	0
1	1	0	1	1	1
0	0	1	0	0	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

Fig. 2(C)

Fig.2. In this figure 2(A) shows block diagram of 3X3 Toffoli gate, 2(B) shows its quantum representation and 2(C) shows its truth table.

The block diagram of 2X2 Feynmann gate is represented by Fig. 3(A). The quantum cost of 2X2 Feynmann gate is 1 from Fig. 3(B). Total logical calculation is T = 1α[17]. Here, α represents a NOT calculation.

Fig. 3(A)

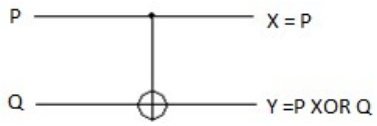

Fig. 3(B)

INPUT		OUTPUT	
P	Q	X = P	Y = P XOR Q
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

Fig. 3(C)

Fig.3. In this figure 3(A) shows block diagram of 2X2 Feynmann gate, 3(B) shows its quantum representation and 3(C) shows its truth table.

The block diagram of 3X3 Double Feynmann gate is represented by Fig. 4(A). The quantum cost is 2 from Fig. 4(B). Total Logical Calculation is T = 2α.

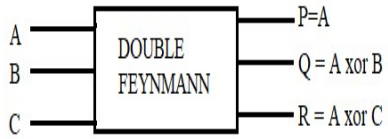

Fig. 4(A)

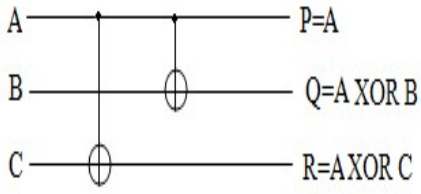

Fig. 4(B)

INPUT			OUTPUT		
A	B	C	P = A	Q = A XOR B	R = A XOR C
0	0	0	0	0	0
0	1	0	0	0	1
1	0	0	1	1	0
1	1	0	1	1	1
0	0	1	0	1	1
0	1	1	0	1	0
1	0	1	1	0	1
1	1	1	1	0	0

Fig. 4(C)

Fig.4. In this figure 4(A) shows block diagram of 3X3 Double Feynmann gate, 4(B) shows its quantum representation and 4(C) shows its truth table.

The block diagram of 3X3 Peres gate is represented by Fig. 5(A). The quantum cost is 4 from Fig. 5(B). Total logical calculation is T = 2α + 1β[18]. Here, α represents a NOT calculation and β represents a two input AND gate calculation.

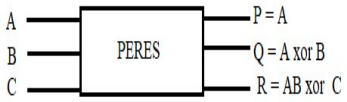

Fig. 5(A)

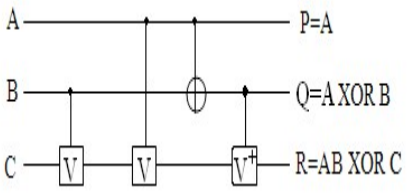

Fig. 5(B)

INPUT			OUTPUT		
A	B	C	P = A	Q = A XOR B	R = AB XOR C
0	0	0	0	0	0
0	1	0	0	0	0
1	0	0	1	1	0
1	1	0	1	1	1
0	0	1	0	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	0	0

Fig. 5(C)

Fig.5. In this figure 5(A) shows block diagram of 3X3 Peres gate, 5(B) shows its quantum representation and 5(C) shows its truth table.

The block diagram of 4X4 HNG gate is represented by Fig. 6(A). The quantum cost of 4X4 HNG gate is 6 from Fig. 6(B). Total logical calculation is T = 4α + 2β.[19]

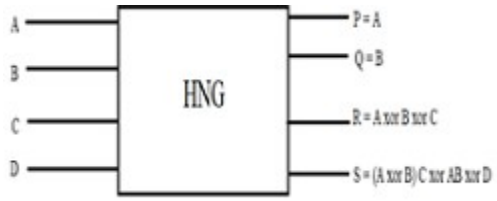

Fig. 6(A)

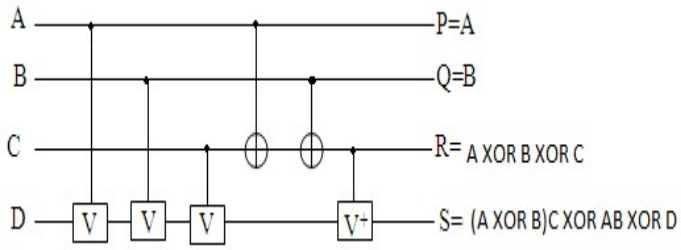

Fig. 6(B)

INPUT				OUTPUT			
A	B	C	D	P = A	Q = B	R = A XOR B XOR C	S = (A XOR B)C XOR AB XOR D
0	0	0	0	0	0	0	0
0	1	0	0	0	1	1	0
1	0	0	0	1	0	1	0
1	1	0	0	1	1	0	1
0	0	0	1	0	0	0	1
0	1	0	1	0	1	1	1
1	0	0	1	1	0	1	1
1	1	0	1	1	1	0	0
0	0	1	0	0	0	1	0
0	1	1	0	0	1	0	1
1	0	1	0	1	0	0	1
1	1	1	0	1	1	1	1
0	0	1	1	0	0	1	1
0	1	1	1	0	1	0	0
1	0	1	1	1	0	0	0
1	1	1	1	1	1	1	0

Fig. 6(C)

Fig.6. In this figure 6(A) shows block diagram of 4X4 HNG gate, 6(B) shows its quantum representation and 6(C) shows its truth table.

Every Boolean Function can be build using 3X3 Fredkin gate. The block diagram of 3X3 Fredkin gate is represented by Fig. 7(A). The quantum cost of Fredkin gate is 5 from Fig. 7(B). Total logical calculation is T = 2α+4β+1δ.[20,21]

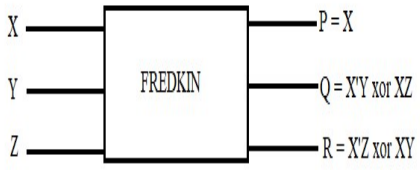

Fig. 7(A)

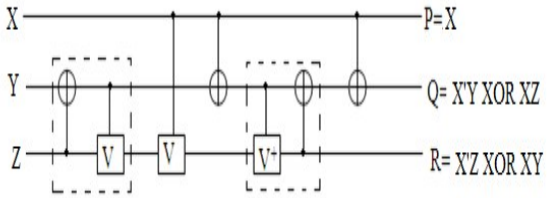

Fig. 7(B)

INPUT			OUTPUT		
X	Y	Z	P = X	Q = X'Y XOR XZ	R = X'Z XOR XY
0	0	0	0	0	0
0	1	0	0	1	0
1	0	0	1	0	0
1	1	0	1	0	1
0	0	1	0	0	1
0	1	1	0	1	1
1	0	1	1	1	0
1	1	1	1	1	1

Fig. 7(C)

Fig.7. In this figure 7(A) shows block diagram of 3X3 Fredkin gate, 7(B) shows its quantum representation and 7(C) shows its truth table.

The block diagram of 4X4 TSG gate is represented by Fig. 8(A). The quantum cost of 4X4 TSG gate is 17 from Fig.8(B)[6]. Total logical calculation is T= 4α+3β +3δ.[22]

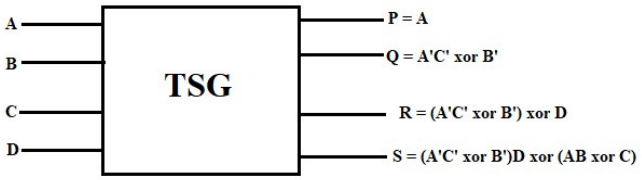

Fig. 8(A)

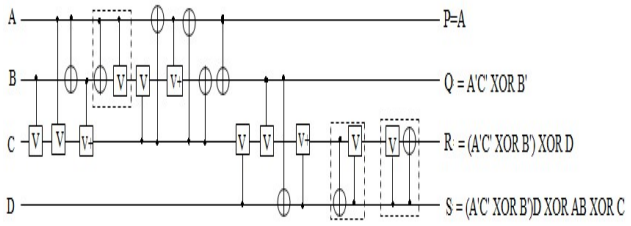

Fig. 8(B)

INPUT				OUTPUT			
A	B	C	D	P = A	Q = A'C' XOR B'	R = (A'C' XOR B') XOR D	S = (A'C' XOR B')D XOR (AB XOR C)
0	0	0	0	0	0	0	0
0	1	0	0	0	1	1	0
1	0	0	0	1	1	1	0
1	1	0	0	1	0	0	1
0	0	0	1	0	0	1	0
0	1	0	1	0	1	0	1
1	0	0	1	1	1	0	1
1	1	0	1	1	0	1	1
0	0	1	0	0	1	1	1
0	1	1	0	0	0	0	1
1	0	1	0	1	1	1	1
1	1	1	0	1	0	0	0
0	0	1	1	0	1	0	0
0	1	1	1	0	0	1	1
1	0	1	1	1	1	0	0
1	1	1	1	1	0	1	0

Fig. 8(C)

Fig.8. In this figure 8(A) shows block diagram of 4X4 TSG gate, 8(B) shows its Quantum representation and 8(C) shows its truth table.

i. Universal gates using Basic Reversible gates:

Reversible computation arises from the point of view to reduce heat dissipation, thereby allowing [23]:

– Higher densities.

– Higher speed .

Reversible Logic:

Reversible logic are circuits (gates) that have one-to-one mapping between vectors of inputs and outputs; thus the vector of input states can be always reconstructed from the vector of output states.

From the truth table in Fig. 9, the characteristics of two input AND gate operation is done, if C =0 and we will get the output from Port 'R', using Toffoli gate.

INPUT			OUTPUT		
A	B	C	P = A	Q = B	R = AB XOR C
0	0	0	0	0	0
0	1	0	0	1	0
1	0	0	1	0	0
1	1	0	1	1	1

Fig.9. Implement a two input AND gate with inputs A & B and output R, using 3X3 Toffoli gate when C is grounded.

From the truth table in Fig. 10, characteristics of two input AND gate operation is done, if C = 0 and we will get the output from Port 'R', using Peres gate.

INPUT			OUTPUT		
A	B	C	P = A	Q = A XOR B	R = AB XOR C
0	0	0	0	0	0
0	1	0	0	0	0
1	0	0	1	1	0
1	1	0	1	1	1

Fig.10. Implement a two input AND gate with inputs A & B and output R, using 3X3 Peres gate when C is grounded.

From the truth table in Fig.11, we will get the characteristics of two input AND gate operation when both C and D are grounded, and then, we will get the output through port 'S', using HNG gate.

INPUT				OUTPUT			
A	B	C	D	P = A	Q = B	R = A XOR B XOR C	S = (A XOR B)C XOR AB XOR D
0	0	0	0	0	0	0	0
0	1	0	0	0	1	1	0
1	0	0	0	1	0	1	0
1	1	0	0	1	1	0	1

Fig.11. Implement a two input AND gate with inputs A & B and output R, using 4X4 HNG gate when C and D are grounded.

From the truth table in Fig.12, we will get the characteristics of two input AND gate output through Port 'R' when 'Z' is grounded, using Fredkin gate.

INPUT			OUTPUT		
X	Y	Z	P = X	Q = X'Y XOR XZ	R = X'Z XOR XY
0	0	0	0	0	0
0	1	0	0	1	0
1	0	0	1	0	0
1	1	0	1	0	1

Fig.12. Implement a two input AND Gate with inputs X & Y and output R, using 3X3 Fredkin gate when Z is grounded.

From the truth table in Fig.13, we will get the characteristics of two input AND gate output when C and D are taken as '0' and we will get the output through Port 'S', using TSG gate.

INPUT				OUTPUT			
A	B	C	D	P = A	Q = A'C' XOR B'	R = (A'C' XOR B') XOR D	S = (A'C' XOR B')D XOR (AB XOR C)
0	0	0	0	0	0	0	0
0	1	0	0	0	1	1	0
1	0	0	0	1	1	1	0
1	1	0	0	1	0	0	1

Fig.13. Implement a two input AND gate with inputs A & B and output S, using 4X4 TSG gate when C and D are grounded.

From the truth table in Fig.14, characteristics of two input OR gate operation is done through Port 'Q' and when Z = 1, using Fredkin gate.

INPUT			OUTPUT	
X	Y	Z	P = X	Q = X'Y XOR XZ
0	0	1	0	0
0	1	1	0	1
1	0	1	1	1
1	1	1	1	1

Fig.14. Implement a two input OR gate with inputs X & Y and output Q, using 3X3 Fredkin gate when Z is 1.

From the truth table in Fig. 15, characteristics of two input OR gate operation is done through Port S, when C = 1 and D is grounded, using HNG gate.

INPUT				OUTPUT			
A	B	C	D	P = A	Q = B	R = A XOR B XOR C	S = (A XOR B)C XOR AB XOR D
0	0	1	0	0	0	1	0
0	1	1	0	0	1	0	1
1	0	1	0	1	0	0	1
1	1	1	0	1	1	1	1

Fig.15. Implement a two input OR gate with inputs A & B and output S, using 4X4 HNG gate when C is 1 and D is grounded.

From the truth table in Fig. 16, characteristics of two input OR gate operation is done through Port S, when C is grounded and D = 1, using TSG gate.

INPUT				OUTPUT			
A	B	C	D	P = A	Q = A'C' XOR B'	R = (A'C' XOR B') XOR D	S = (A'C' XOR B')D XOR(AB XOR C)
0	0	0	1	0	0	1	0
0	1	0	1	0	1	0	1
1	0	0	1	1	1	0	1
1	1	0	1	1	0	1	1

Fig.16. Implement a two input OR gate with inputs A & B and output S, using 4X4 TSG gate when C is grounded and D is 1.

From the truth table in Fig. 17, characteristics of two input NAND gate operation is done through Port R, when C = 1, using Toffoli gate.

INPUT			OUTPUT		
A	B	C	P = A	Q = B	R = AB XOR C
0	0	1	0	0	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

Fig.17.Implement a two input NAND gate with inputs A & B and output R, using 3X3 Toffoli gate when C is 1.

From the truth table in Fig. 18, two input NAND gate operation is done through Port R, when C = 1, using Peres gate.

INPUT			OUTPUT		
A	B	C	P = A	Q = A XOR B	R = AB XOR C
0	0	1	0	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	0	0

Fig.18. Implement a two input NAND gate with inputs A & B and output R, using3X3 Peres gate when C is 1.

From the truth table in Fig. 19, characteristics of two input NAND gate operation is done through Port S, when C is Grounded and D = 1, using HNG gate.

INPUT				OUTPUT			
A	B	C	D	P = A	Q = B	R = A XOR B XOR C	S = (A XOR B)C XOR AB XOR D
0	0	0	1	0	0	0	1
0	1	0	1	0	1	1	1
1	0	0	1	1	0	1	1
1	1	0	1	1	1	0	0

Fig.19. Implement a two input NAND gate with inputs A & B and output S, using 4X4 HNG gate when C is grounded and D is 1.

From the truth table in Fig. 20, characteristics of two input NAND gate operation is done through Port S, when C = 1 and D is Grounded, using TSG gate.

INPUT				OUTPUT			
A	B	C	D	P = A	Q = A'C' XOR B'	R = (A'C' XOR B') XOR D	S = (A'C' XOR B')D XOR (AB XOR C)
0	0	1	0	0	1	1	1
0	1	1	0	0	0	0	1
1	0	1	0	1	1	1	1
1	1	1	0	1	0	0	0

Fig.20. Implement a two input NAND gate with inputs A & B and output S, using 4X4 TSG gate when C is 1 and D is grounded.

From the truth table in Fig. 21, characteristics of two input NOR gate operation is done through Port S, when C and D, both are 1, using HNG gate.

INPUT				OUTPUT			
A	B	C	D	P=A	Q=B	R=A XOR B XOR C	S = (A XOR B)C XOR AB XOR D
0	0	1	1	0	0	1	1
0	1	1	1	0	1	0	0
1	0	1	1	1	0	0	0
1	1	1	1	1	1	1	0

Fig.21. Implement a two input NOR gate with inputs A & B and output S, using a 4X4 HNG gate when C and D are 1.

ii. Small Combinational circuits using Basic Reversible gates:

a. Reversible Half Adder:

When C and D are grounded, 4X4 Reversible Half Adder can be designed using HNG gate in Fig. 22(A).

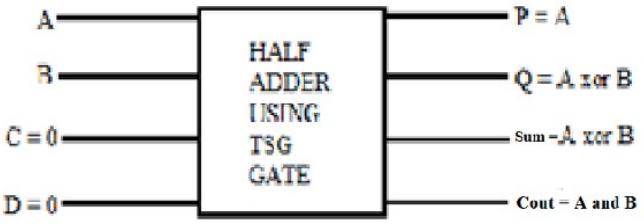

Fig. 22(A)

The truth table of 4X4 Reversible HALF ADDER using HNG gate is shown in Fig. 22(B).

INPUT				OUTPUT			
A	B	C	D	P	Q	Sum	Cout
0	0	0	0	0	0	0	0
0	1	0	0	0	1	1	0
1	0	0	0	1	1	1	0
1	1	0	0	1	0	0	1

Fig. 22(B)

Fig.22. In this figure 22(A) shows a 4X4 Reversible Half Adder using 4x4 TSG gate, where Q and Sum represent the sum and Cout represents the carry out, when C and D are grounded and 22(B) shows its truth table.

b. Reversible Half Subtractor:

The block diagram of 4X4 Reversible Half Subtractor Using TSG GATE is shown in Fig. 23(A).

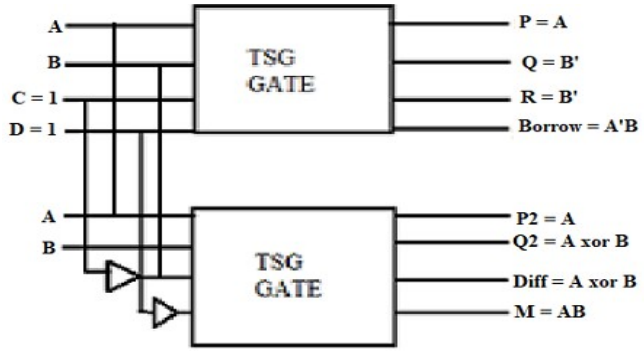

Fig. 23(A)

The truth table of 4X4 Reversible Half Subtractor using TSG is shown in Fig. 23(B) (i) and (ii).

INPUT				OUTPUT			
A	B	C	D	P	Q2	Diff	M
0	0	0	0	0	0	0	0
0	1	0	0	0	1	1	0
1	0	0	0	1	1	1	0
1	1	0	0	1	0	0	1

Fig. 23(B)(i)

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

Fig. 23(B)(ii)

Fig.23. In this figure 23(A) shows 4X4 reversible half subtractor using TSG gate, where one can get the borrow from Borrow output port when C and D are 1 and the difference from Q2 and Diff output port when C and D are grounded, 23(B)(i) shows truth table of difference output from Q2 and Diff ports when C and D are grounded and 23(B)(ii) shows truth table of borrow output from Borrow port when C and D are 1.

c. Reversible Full Adder:

When D is grounded, 4X4 Reversible Full Adder can be designed using HNG gate in Fig. 24(A). The truth table of 4X4 Reversible full adder using HNG gate is shown in Fig. 24(B).

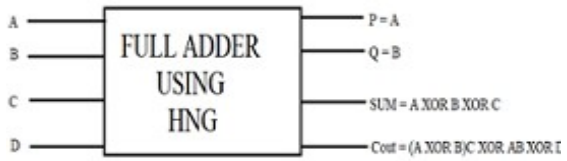

Fig. 24(A)

INPUT				OUTPUT			
A	B	C	D	P	Q	Sum	Cout
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	1
1	1	0	0	1	1	0	1
1	1	1	0	1	1	1	1

Fig. 24(B)

Fig.24. In this figure 24(A) shows 4X4 reversible full adder using HNG gate, where A & B are the inputs and one can get the sum from Sum output port and carry from Cout output port and 24(B) shows its truth table.

When C is grounded and D is taken as 'Cin', 4X4 Reversible Full Adder can be designed using TSG gate in Fig. 25(A).

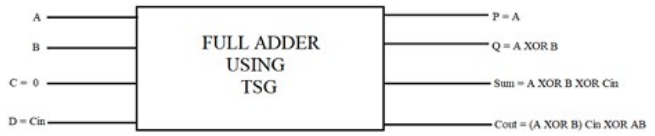

Fig. 25(A)

INPUT				OUTPUT			
A	B	C	D	P	Q	Sum	Cout
0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	0	1
1	0	0	0	1	1	1	0
1	0	0	1	1	1	0	1
1	1	0	0	1	0	0	1
1	1	0	1	1	0	1	1

Fig. 25(B)

Fig.25. In this figure 25(A) shows 4X4 reversible full adder using TSG gate, where A and B are the inputs and one can get the sum from Sum output port and carry from Cout output port and 25(B) shows its truth table.

4X4 Reversible full adder can also be done using Two Peres gates. The following block diagram along with the truth table is shown in Fig. 26(A) and Fig. 26(B) respectively.

The block diagram of 4X4 Reversible full adder using Peres gate is shown in Fig. 26(A).

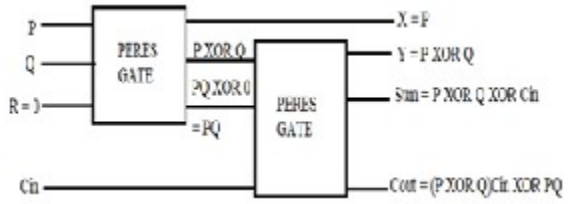

Fig. 26(A)

INPUT				OUTPUT			
P	Q	R	S	X	Y	Sum	Cout
0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	0	1
1	0	0	0	1	1	1	0
1	0	0	1	1	1	0	1
1	1	0	0	1	0	0	1
1	1	0	1	1	0	1	1

Fig. 26(B)

Fig.26. In this figure 26(A) shows 4X4 reversible full adder using Peres gate, where P and Q are the inputs and one can get the sum from Sum output port and carry from Cout output port and 26(B) shows its truth table.

VII. ILLUSTRATION

fn(a, b, c, d) be a four valued Boolean function as shown in (1). Implementation of this function using reversible logic gates, with truth table is shown in Fig. 27.

$$fn(a,b,c,d) = a(b+c) + b.d \text{ ---------- (1)}$$

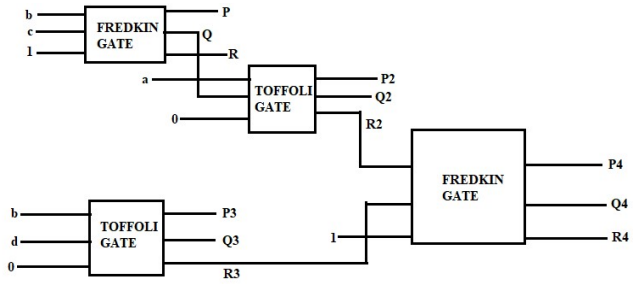

Fig. 27(A)

[Here, P =b; Q = b'c xor b = b'c + b; R = b' xor bc; P2 = a; Q2 = Q; R2 = a(b'c +b); P3 = b; Q3 = d; R3 = bd xor 0 = bd; P4 = a(b'c +b); Q4 = ab'c + ab + a'bd; R4 = (a(b'c+b))' xor abd.]

INPUT				OUTPUT
a	b	c	d	Q4
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Fig: 27(B)

Fig.27. In this figure 27(A) shows the implementation of a Boolean expression using basic reversible gates and 27(B) shows its truth table.

VIII. EXPERIMENTAL RESULTS:

On-Chip Power and Timing Analysis:

Using Xilinx VHDL simulator, the power and time delay of the basic and universal gates and small combinational circuits which are realized from the Reversible gates, are determined and shown in Fig. 28, 29.

Based on the result shown in Fig. 9 -13, the power and delay output of the AND gate is calculated. For example, in Fig. 9, the output line R of the Toffoli gate provides the AND gate

output. Hence we compute the delay as the maximum time taken from inputs A and B to reach output line R. The minimum delay is taken as the smallest time from inputs A and B to output R. The value of power is taken as provided by Xilinx VHDL (Family - Artix-7, Package - csg324). Similarly the line choice for power and delay calculation is done for the other derived AND gates from Fig. 10 - 13. The power and delay values for the OR gates are based on results obtained from Fig. 14 - 16. The power and delay values for the NAND gates are based on results obtained from Fig. 17 - 20. The power and delay values for the NOR gates are based on results obtained from Fig. 21.

Universal Gates	Reversible Gates	On-Chip Power(W)	Output Port	Max. Delay (ns)	Min. Delay (ns)
AND					
	Toffoli	**1.005**	**R**	**6.084**	**2.209**
	Peres	**1.191**	**R**	**6.092**	**2.173**
	HNG	**1.648**	**S**	**6.170**	**2.266**
	Fredkin	**1.021**	**R**	**5.773**	**2.102**
	TSG	**1.731**	**S**	**6.134**	**2.350**
OR					
	HNG	**1.648**	**S**	**6.170**	**2.266**
	TSG	**1.731**	**S**	**6.134**	**2.350**
	Fredkin	**1.021**	**Q**	**6.080**	**2.210**
NAND					
	Toffoli	**0.812**	**R**	**5.784**	**2.181**
	Peres	**1.327**	**R**	**5.780**	**2.164**
	HNG	**1.648**	**S**	**6.170**	**2.266**
	TSG	**1.731**	**S**	**6.134**	**2.350**
NOR					
	HNG	**1.648**	**S**	**6.170**	**2.266**

Fig.28. Experimental results of on-chip power and propagation delay of AND, OR, NAND NOR gates using basic reversible gates.

The power and delay values for the half adder, half subtractor and full adder are based on results obtained from Fig. 22 - 26.

Combinational Circuits	*Universal Gates*	*On-chip power(W)*	*Propagation Delay Measurement*	*Max. delay(ns)*	*Min. delay(ns)*
Half Adder	*TSG*	*1.731*	*Sum*	*6.307*	*2.399*
		1.731	*Cout*	*6.134*	*2.350*
Half Subtractor	*TSG*	*2.371*	*Diff*	*5.626*	*2.163*
		2.371	*Borrow*	*5.810*	*2.244*
Full Adder	*HNG*	*1.648*	*Sum*	*6.268*	*2.198*
		1.648	*Cout*	*6.505*	*2.283*
	PERES	*1.664*	*Sum*	*6.136*	*2.144*
		1.664	*Cout*	*6.307*	*2.216*
	TSG	*1.669*	*Sum*	*6.185*	*2.195*
		1.669	*Cout*	*6.428*	*2.280*

Fig.29. Experimental results of on-chip power and propagation delay of half adder, half subtractor and full adder using basic reversible gates.

IX. CONCLUSION

From the analysis in Fig 28, we can conclude that, for a two input AND gate implementation and operation using reversible gates, the Toffoli gate is the best choice in terms of on-chip power, whereas HNG gate is the best for maximum delay and Fredkin gate is the best for minimum delay. For a two input OR gate implementation and operation, Fredkin gate is the best choice in terms of on chip power and minimum delay, while HNG gate is the best for maximum delay. For a two input NAND implementation, Toffoli gate is the best choice in terms of on chip power and maximum delay, while Peres gate is the best for minimum delay. For a two input NOR implementation, HNG is the best reversible gate both in terms of on-chip power and propagation delay. In Fig. 29, TSG gate is the best for implementing 4X4 Half Adder and 4X4 Half Subtractor. HNG gate is the best for implementing 4X4 Full Adder in terms of on-chip power whereas for maximum delay of Sum and Cout, HNG gate is the best and for minimum delay, Peres gate is the best reversible gate. One can implement any of the Boolean function with the best gate with lowest on-chip power and propagation delay.

As we know, a processor that can only use 5% of its transistors at any given time will vary in characteristics from that which use 50%, so our aim will be to increase the percentage rate of usage of transistors with lowering of the heat dissipation. So, we can conclude that with reverse computing, the problem of the whole world regarding heat dissipation of transistors can be partially resolved.

REFERENCES

[1] WIKIPEDIA on "*Dark Silicon*".
[2] R. Landauer in "*Irreversibility and Heat Generation in the Computing Process*". IBM J. Research and Development,5(3): pp. 183-191, 1961.
[3] R. Keyes, R. Landauer in *"Minimal energy dissipation in logic"*, IBM J. Res. Dev. 14(1970) 152–157.
[4] Charles H. Bennett , in *"Logical Reversibility of computation"*, IBM Journal of Research and Development, vol. 17, no. 6, pp. 525-532, 1973.
[5] W.D. Pan, M. Nalasani "Reversible logic", IEEE Potentials (Volume: 24 , Issue: 1 , Feb.-March 2005).
[6] B. Davari ; R.H. Dennard ; G.G. Shahidi in *"CMOS scaling for high performance and low power-the next ten years."*
[7] P. Gupta; A. Agrawal; N.K. Jha in *"An Algorithm for Synthesis of Reversible Logic Circuits"*.
[8] T.Toffoli, Automata, Languages and Programming. Springer Verlag, 1980, ch. title: *Reversible Computing*, pp. 632–644.
[9] B.Hema Latha in *"Necessities of Low Power VLSI Design Strategies and its involvement with new Technologies"*.
[10] Mayank Kumar Singh and Rangaswamy Nakkeeran in *"Design of novel reversible logic gate with enhanced traits"*.
[11] D.P. Vasudevan ; P.K. Lala ; Jia Di ; J.P. Parkerson in *"Reversible-logic design with online testability"*.
[12] 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.
[13] 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.
[14] M. A. Nielsen and I. L. Chuang, in *"Quantum Computation and Quantum Information"*, New York: Cambridge Univ. Press (2000)
[15] V.Vendral, A. Barenco, and A.Ekert,in "*Quantum networks for elementary arithmetic operations"*, Phys. Rev. A, vol. 54, no. 1, pp. 147-153, Jul1996.
[16] E. Fredkin and T. Toffoli in *"Conservative Logic"*. International Theoretical Physics Vo121, pp.219-253, 1982.

[17] R. Feynman in *"Quantum Mechanical Computers",* Optic News, Vol 11, pp 11-20 1985.
[18] A. Peres in *"Reversible logic and quantum computers",* Phys.Rev. A 32 (1985) 3266-3276.
[19] Shams, M., M. Haghparast and K. Navi in *"Novel reversible multiplier circuit in nanotechnology"*, World Appl.Sci. J.,3(5): 806-810.
[20] Dmitri Maslov, Gerhard W. Dueck, and D. Michael Miller in *"Synthesis of Fredkin–Toffoli Reversible Networks"* in IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, VOL. 13, NO. 6, JUNE 2005.
[21] G. W. Dueck, D. Maslov, and D. M. Miller, *"Transformation-based synthesis of networks of Toffoli/Fredkin gates,"* in Proc. IEEE Canadian Conf. Electrical and Computer Engineering, May 2003, pp. 211–214.
[22] Himanshu Thapliyal and M.B. Srinivas in *"A Novel Reversible TSG gate and Its Application for Designing Reversible Carry Look-Ahead and Other Adder Architectures"* Center for VLSI and Embedded System Technologies, IIIT,Hyderabad, India, Asia-Pacific Conference on Advances in Computer Systems Architecture.
[23] WIKIPEDIA *on "Reversible Computation".*