TRANSISTOR LEVEL IMPLEMENTATION OF DIGITAL REVERSIBLE CIRCUITS

K.Prudhvi Raj[1] and Y.Syamala[2]

[1]PG student, Gudlavalleru Engineering College, Krishna district, Andhra Pradesh, India
[2]Departement of ECE, Gudlavalleru engineering college,
Krishna district, Andhra Pradesh, India

ABSTRACT

Now a days each and every electronic gadget is designing smartly and provides number of applications, so these designs dissipate high amount of power. Reversible logic is becoming one of the best emerging design technologies having its applications in low power CMOS, Quantum computing and Nanotechnology. Reversible logic plays an important role in the design of energy efficient circuits. Adders and subtractors are the essential blocks of the computing systems. In this paper, reversible gates and circuits are designed and implemented in CMOS and pass transistor logic using Mentor graphics backend tools. A four-bit ripple carry adder/subtractor and an eight-bit reversible Carry Skip Adder are implemented and compared with the conventional circuits.

KEYWORDS

Low power, Reversible logic gates, Adder, Subtractor, Mentor graphics tools.

1. INTRODUCTION

In modern VLSI systems, power dissipation is the critical limiting factor for more complex circuits. According to the Landauer's principle [1], every conventional combinational circuit dissipates KTln2 Joules of energy for one-bit loss of information, where k is Boltzmann's constant (1.3807 x 10^{-23} joules per Kelvin) and T is absolute temperature. Reversible computation [2], is a research area having characteristics that are both forward and backward computations. In ideal cases, these circuits have zero information loss. Therefore, reversible computing is an appealing solution in the design of energy efficient circuits, which have low power dissipation.

1.1 Reversible logic

Reversible logic is a very forthcoming approach of logic synthesis for power reduction in future computing technologies. Most of the gates used in digital design are not reversible for example AND, OR, EXOR gates do not perform reversible operation. A reversible gate/circuit can generate unique output vector for corresponding input vector i.e. one to one mapping is between input and output vectors. Therefore, out of all commonly used gates NOT gate is the only reversible gate with one input and one output (1×1).

A basic structure of reversible gate is shown in figure 1. A gate/circuit is said to be reversible if it follows the below characteristics.

DOI : 10.5121/vlsic.2014.5606

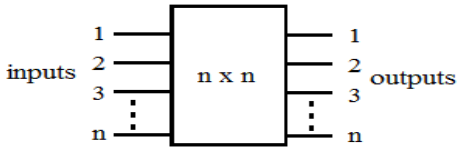

Figure 1. Basic structure of reversible gate

- A reversible logic gate must have equal number of input and output vectors i.e. 2×2, 3×3, 4×4 ... n×n.
- For each input pattern, there must be a unique output pattern.
- Each output must be used only once.
- Loops and feedbacks are not permitted in reversible designing.

1.2 Reversible Gates

Reversible gate is an n input and n output logic function, which has one to one correspondence between the inputs and outputs. There exists many number of reversible gates at present [3], the simplest reversible gate is NOT (1×1) gate. Feynman gate, which is also known as controlled NOT gate, is an example for 2×2 gates. Fredkin, Toffoli, TR and Peres gates are the 3×3 reversible gates. Any reversible gate is realised by using 1×1 and 2×2 reversible gates by which the quantum cost of the gate is calculated. A reversible gate also satisfies the following performance parameters.

Constant input: This refers to the input, which is maintained as constant at either 0 or 1 in order to attain appropriate logic function.

Garbage output: The output of a gate, which is not given as the input of another gate, is referred as garbage output. For better performance, number of garbage outputs must be less.

Quantum cost: Quantum cost refers to the cost of the circuits in terms of the cost of primitive gates, i.e. the number of primitive gates such as 1×1 and 2×2 required for the realization of a reversible gate/circuit.

VHDL and verilog are the coding techniques used to implement reversible gates/circuits in HDL designing and Xilinx ISE simulator. Many number of reversible adders, subtractors, multipliers and ALUs are implemented by using these coding techniques. Mentor graphics tools is one of the backend techniques to implement and simulate the circuits in transistor level. Here the reversible gates and circuits are implemented in CMOS and pass transistor logic [4] and compared with each other. Any logic expression can be implemented by using CMOS logic family, which contains PMOS and NMOS transistors as pull up and pull down networks respectively. To obtain a logic function it has to design the inversion of that function, because CMOS is inversion logic. In pass transistor logic, the source input is passed to drain output if gate input of PMOS is zero, or if the gate input of NMOS is one.

2. TRANSISTOR REALISATION OF REVERSIBLE GATES

2.1 Feynman gate

Figure 2 represents the Feynman gate, which realises XOR gate with a garbage output ‘A’. If B=‘0’ it duplicates the input ‘A’ and if B=‘1’, then it inverts the input ‘A’ to the output Q.

Figure 2. Feynman gate

Figure 3 shows the CMOS realisation of Feynman gate, the first output P is a buffer from the input A, and to make the input 'A' pass through the output 'P' simply the gate of PMOS pass transistor is grounded. The second output is a XOR function and 12 transistors required to implement the XOR function**.**

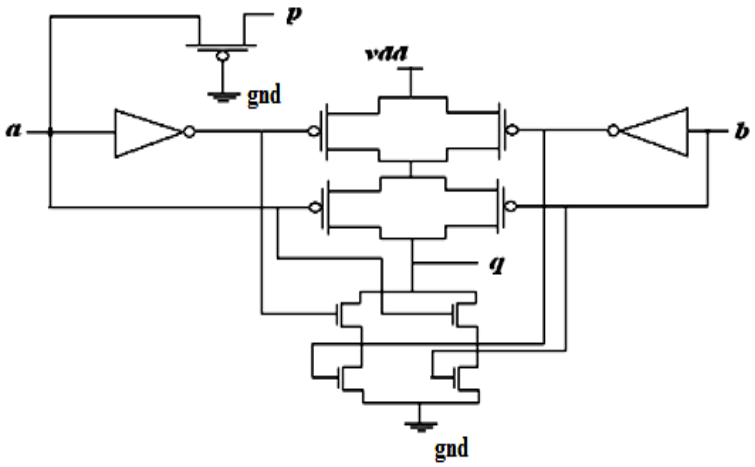

Figure 3. CMOS realisation of Feynman gate

The pass transistor realisation of Feynman gate is shown in figure 4. Consider inputs as a='1' and b='1', then the transistors Q_2, Q_4 and Q_6 are ON and the remaining transistors are OFF. So the Vdd value '1' is directly passed to the output 'p' and the ground value '0' is passed to the output q. So, p='1' and q='0'.

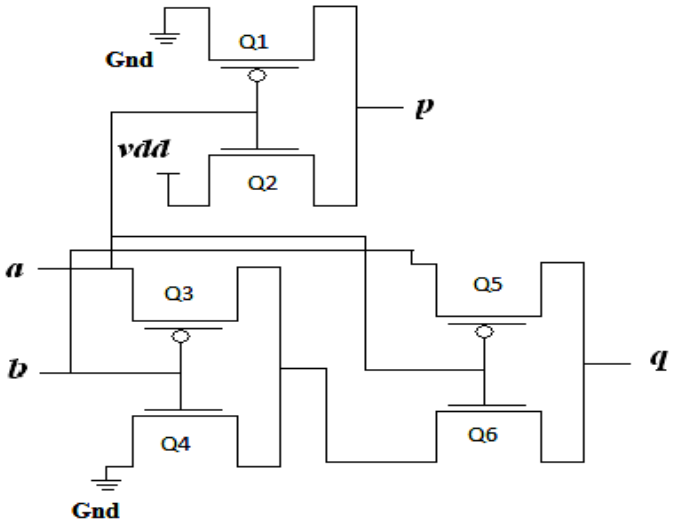

Figure 4. Pass transistor realisation of Feynman gate

2.2 Peres gate

The figure 5 represents the Peres gate, with A, B, C as inputs and P, Q, R as outputs, where P=A, Q=A⊕B and R=AB⊕C.

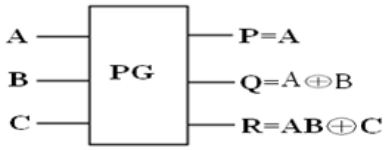

Figure 5. Peres gate

Peres gate is a modified Toffoli gate. It is a combination of Toffoli gate and Feynman gate. Figure 6 represents the CMOS realisation of Peres gate. The first output of the Peres gate is buffer of the first input. So, a PMOS transistor with grounded gate is used. The second output is a XOR function. So an XOR gate is realised using 12 transistors and the third output is an XOR function of the third input with an AND function of the first two inputs.

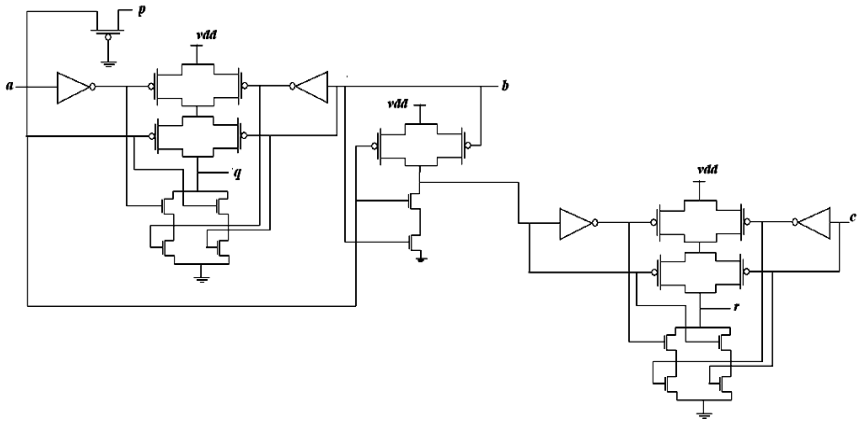

Figure 6. CMOS realisation of Peres gate

The pass transistor realisation of Peres gate is shown in figure 7. Consider the inputs are a='1', b='0', c='0'. Since a='1' the transistor Q_1 is OFF and Q_2 is ON. So the output p='1'. Since a='1' and b='0', Q_3, Q_8 are ON and Q_4, Q_7 are OFF. So the input 'a' is passed and the output q='1'. Since a='1', b='0' and c='0', then Q_6, Q_9, Q_{11} are ON and Q_5, Q_{10}, Q_{12} are OFF. So the input 'c' is passed and the output r='0'.

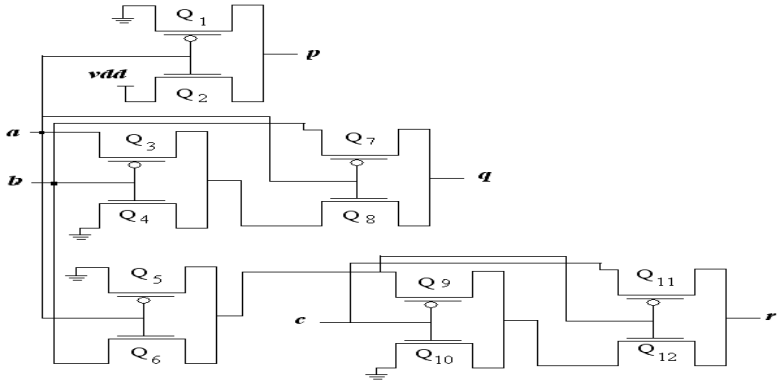

Figure 7. Pass transistor realisation of Peres gate

2.3 Fredkin gate

Figure 8 represents the Fredkin gate with inputs A, B, C and outputs P, Q, and R. Where P=A, Q=A'B+AC and R=AB+A'C.

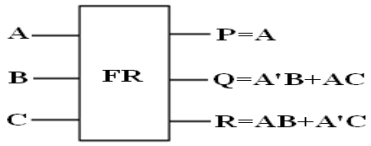

Figure 8. Fredkin gate

Fredkin gate acts as a Multiplexer, if the input A is either '0' or '1' then the outputs Q and R swaps between the inputs B and C.

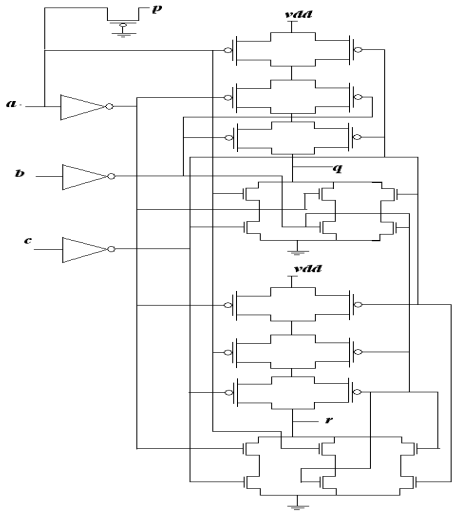

Figure 9. CMOS realisation of Fredkin gate

The CMOS realization of Fredkin gate is shown in figure 9. The first output of the Fredkin gate is buffer of the first input. So, a PMOS transistor with grounded gate is used. The second output Q=A'B+AC and the third output R=AB+A'C are realised by the pull up and pull down networks with PMOS and NMOS transistors respectively.

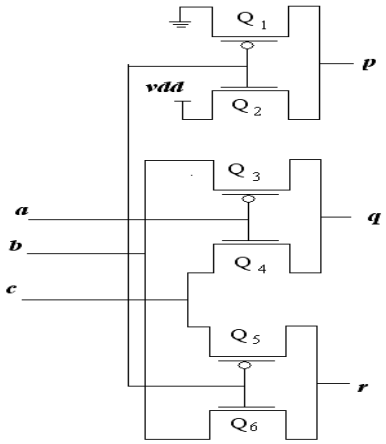

Figure 10. Pass transistor realisation of Fredkin gate

The pass transistor realisation of Fredkin gate is shown in figure 10. Consider the inputs are a='1', b='0', c='1'. Since a='1', transistors Q_2, Q_4, Q_6 are ON, then Vdd is passed through Q_2 and the output p='1'. The input 'c' is passed through the transistor Q_4. So q='1' and the input 'b' is passed through the transistor Q_6. So r='0'.

2.4 TR gate

Figure 11 represents the TR gate with A, B, C as inputs and P, Q, R as outputs, where P=A, Q=A⊕B and R=AB'⊕C. This gate is proposed by Tapliyal and Ranganathan. The figure 12 shows the CMOS realisation of TR gate.

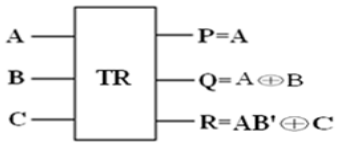

Figure 11. TR Gate

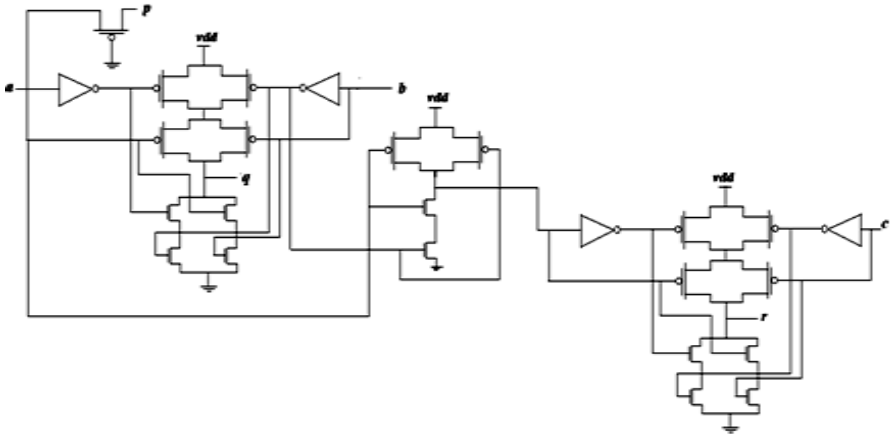

Figure 12. CMOS realisation of TR gate

The first output of the TR gate is buffer of the first input. So, a PMOS transistor with grounded gate is used. The second output is a XOR function so an XOR gate is realised using 12 transistors and for the third output an AND function of first two inputs having XOR with the third input is realised.

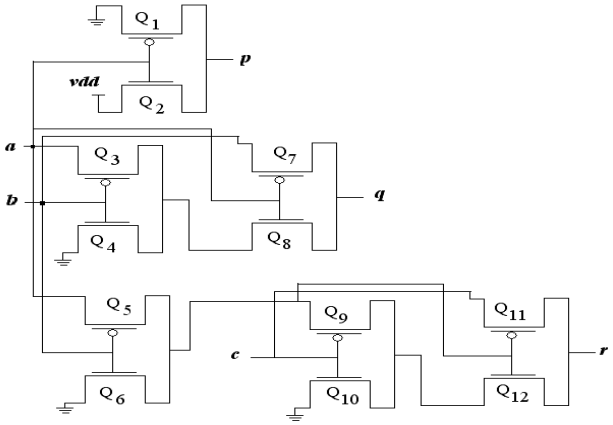

Figure 13. Pass transistor realisation of TR gate

The pass transistor realisation of TR gate is shown in figure 13. Consider the inputs are a='1', b='1', c='0'. Since a='1' the transistor Q_1 is OFF and Q_2 is ON. So the output p='1'. Since a='1' and b='1', Q_4, Q_8 are ON and Q_3, Q_7 are OFF. So the ground value '0' is passed through Q_4, Q_8 then the output q='0'. Since a='1', b='1' and c='0', then Q_6, Q_9, Q_{12} are ON and Q_5, Q_{10}, Q_{11} are OFF. So the ground value '0' is passed through Q_{10}, Q_{12} then the output r='0'.

3. TRANSISTOR REALISATION OF REVERSIBLE CIRCUITS

Reversible circuits are implemented using the reversible gates only. There are many designs of one bit full adder/subtractor circuits [5, 6]. Here, two designs of full adder/subtractor use 8 and 4 gates respectively. In this work these designs are implemented in CMOS and pass transistor logics using Mentor graphics tools.

3.1 One-Bit Reversible full Adder/Subtractor

Figure 14 shows a one-bit reversible adder/subtractor [7] using three Feynman gates, two Peres gates, two TR gates and one Fredkin gate. A control input is given to switch between adder and subtractor. If control input is '1' addition is performed else if it is '0' subtraction is performed.

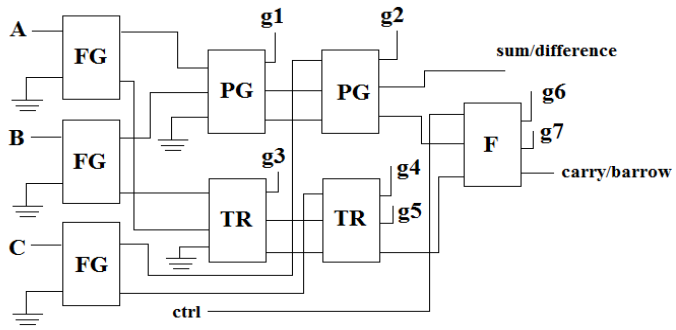

Figure 14. Design 1of one bit full adder/subtractor

Figure 15 shows a one-bit reversible adder/subtractor [8] using two Feynman gates and two Peres gates. A control input is given to switch between adder and subtractor. If control input is '0' addition is performed else if it is '1' subtraction is performed.

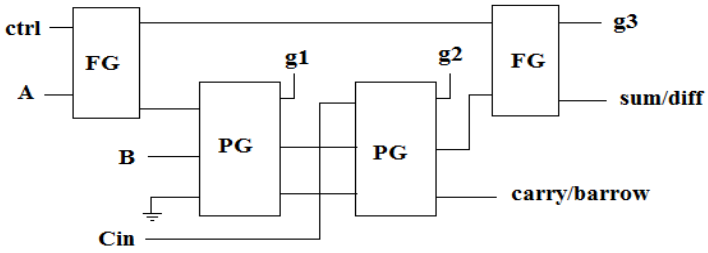

Figure 15. Design 2 of one bit full adder/subtractor

3.2 Four-bit ripple carry adder/subtractor

By using the one-bit adder/subtractor, a four-bit ripple carry adder/subtractor [9] is implemented as shown in figure 16. A control signal is given to all the full adder/subtractor circuits by duplicating the signal using Feynman gates.

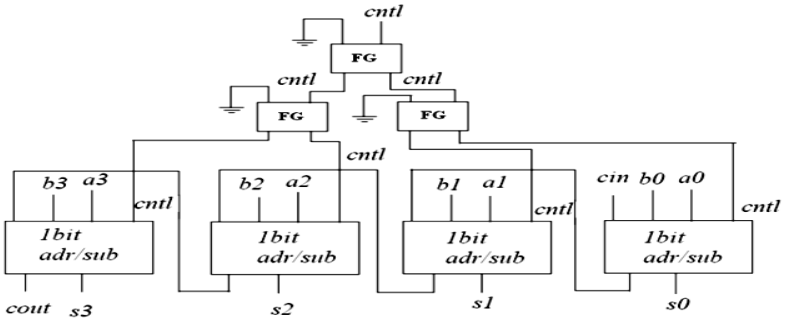

Figure 16. Four-bit ripple carry adder/subtractor

Two four bit operands A(a3-a0), B(b3-b0) are given to the circuit and verified the addition and subtraction operations.

3.3 Carry skip adder

Carry skip adder [10, 11] provides a compromise between a ripple carry adder and a CLA (Carry Look Ahead adder). In carry skip adder the delay is reduced due to carry computation. In a full adder if one of the operand is '1' and the other one is '0' then the carry input is equals to carry output of that full adder. Therefore in such cases of n bit adder the carry in of the first stage directly propagate to the last stage, so delay is reduced, so it is also known as carry bypass adder. The figure 17 represents the conventional model of a four bit carry skip adder. If the propagate $P_i=X_i \oplus Y_i$ is 1 then it provides an alternative path for the incoming carry signal to block the carry out. Therefore, delay is reduced.

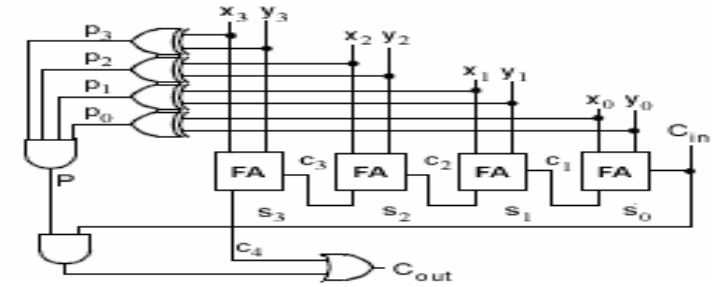

Figure 17. A four-bit conventional carry skip adder

Here an eight-bit CSA is implementing using Double Peres gate, which can individually acts as full adder and verified in transistor level.

3.3.1 Double Peres Gate

Double Peres gate, which is shown in figure 18, can work singly as a reversible full adder with two garbage outputs. It is a 4×4 reversible gate with A, B, C and D as inputs and P, Q, R and S as outputs.

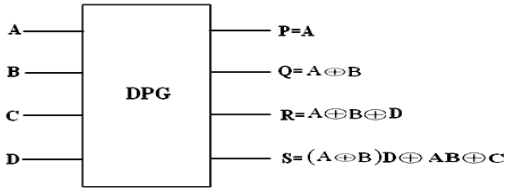

Figure 18. Double Peres Gate.

The outputs are P=A, Q=A⊕B, R=A⊕B⊕D and S= (A⊕B)D⊕AB⊕C. To act as a full adder the third input of the DPG gate must be zero. Table 1 gives the truth table for DPG.

Table 1. Truth table of DPG gate

INPUTS				OUTPUTS			
A	**B**	**C**	**D**	**P**	**Q**	**R**	**S**
0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	0
0	0	1	0	0	0	0	1
0	0	1	1	0	0	1	1
0	1	0	0	0	1	1	0
0	1	0	1	0	1	0	1
0	1	1	0	0	1	1	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	1	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	0	0
1	1	0	0	1	0	0	1
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	0
1	1	1	1	1	0	1	0

The pass transistor realization of DPG gate is shown in below figure 19. Consider the inputs are a='1', b='0', c='0' and d='0'. Since a='1' then Q_2 is ON. So Vdd passed to the output 'p', then p='1'. Since a='1' and b='0', Q_3, Q_6 are ON and Q_4, Q_5 are OFF. Then input 'a' passed to the output 'q'. So q='1'. Since a='1', b='0' and d='0', then Q_3, Q_6, Q_7, Q_{10} are ON. So the input 'a' passed to the output 'r'. So r='1'. Since a='1', b='0', c='0' and d='0', then the transistors Q_3, Q_6, Q_{11}, Q_{13}, Q_{15}, Q_{18}, Q_{19} and Q_{21} are ON. So, the output s='0'. By concluding this if the input vector is "1000", and then the output vector will be "1110".

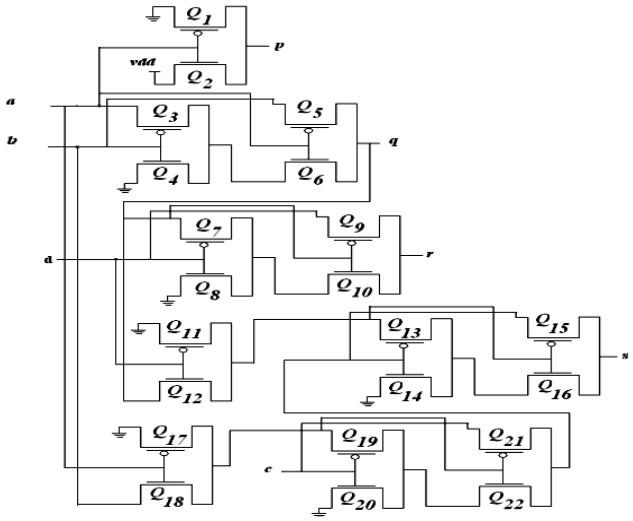

Figure 19. Transistor realization of DPG gate

3.3.2 Eight-bit carry skip adder

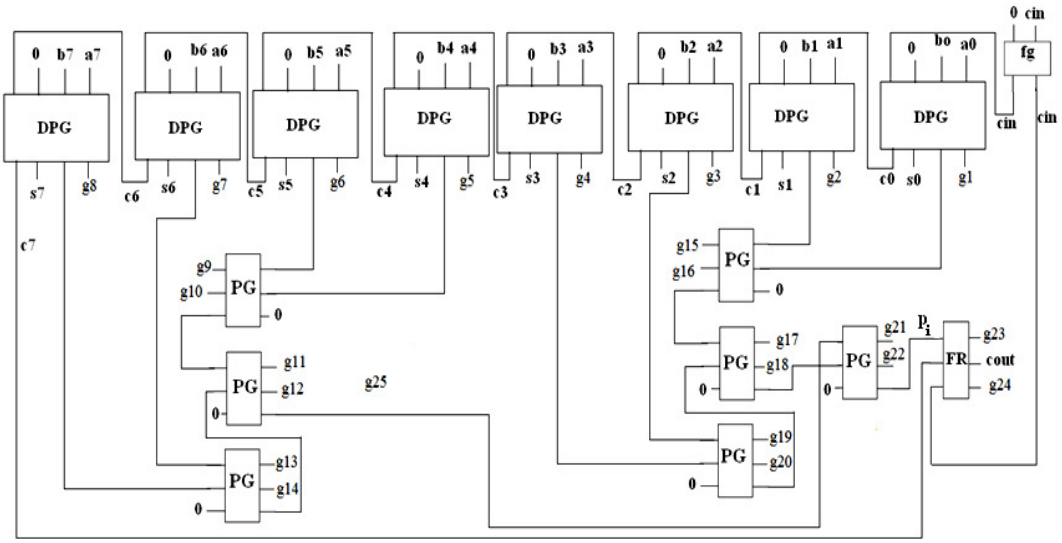

Figure 20. Eight-bit carry skip adder

An eight bit reversible carry skip adder [12] is designed by cascading the DPG gates as shown in figure 20, if the propagate signal p_i is '1' then the carry input takes an alternative path to the last stage of the carry skip adder. So, the delay is reduced and if p_i is '0' then the carry propagates through all the stages. The transistor realizations of reversible eight bit Carry Skip Adder is verified by Mentor graphics tools using ELDO simulator.

4. RESULTS

4.1 Simulation results of Feynman and Peres gates

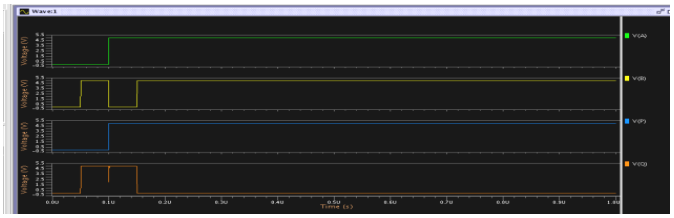

Figure 21. Simulation results of Feynman gate

The figure 21 gives the simulation results of Feynman gate. If the inputs are a='0' and b='1' then the outputs are p='0' and q='1'.

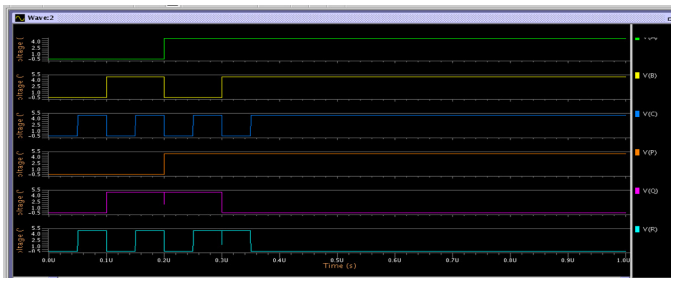

Figure 22. Simulation results of Peres gate

The simulation results of Peres gate are shown in figure 22, if the inputs are a='1', b='0' and c='1' then the outputs are p='1', q='1' and r='1'.

Table 2. Synthesis results of Feynman gate using CMOS and pass transistor logic

Reversible gate	Logic family	No of transistors required	L/W of transistor	Voltage applied	Power dissipation(watts)
Feynman gate	CMOS	13	0.25u/50u	5v	23.576 m
			0.5u/25u		10.576 n
			2u/10u		96.764 p
			2u/10u	3v	34.128 p
			2u/10u	1.5v	8.1648 p
	PASS transistor logic	6	0.25u/50u	5v	12.509 u
			0.5u/25u		53.9004 p
			2u/10u		22.835 p
			2u/10u	3v	7.7837 p
			2u/10u	1.5v	1.7092 p

Table 3. Synthesis results of Peres gate using CMOS and pass transistor logic

Reversible gate	Logic family	No of transistors required	L/W of transistor	Voltage applied	Power dissipation(watts)
Peres gate	CMOS	31	0.25u/50u	5v	71.0433 m
			0.5u/25u		23.4959 n
			2u/10u		231.328 p
			2u/10u	3v	81.924 p
			2u/10u	1.5v	19.785 p
	PASS transistor logic	12	0.25u/50u	5v	12.509 u
			0.5u/25u		53.9004 p
			2u/10u		22.8358 p
			2u/10u	3v	7.7837 p
			2u/10u	1.5v	1.7092 p

Table 2 and table 3 give the comparisons between the CMOS and pass transistor realisations of Feynman gate and Peres gate respectively.

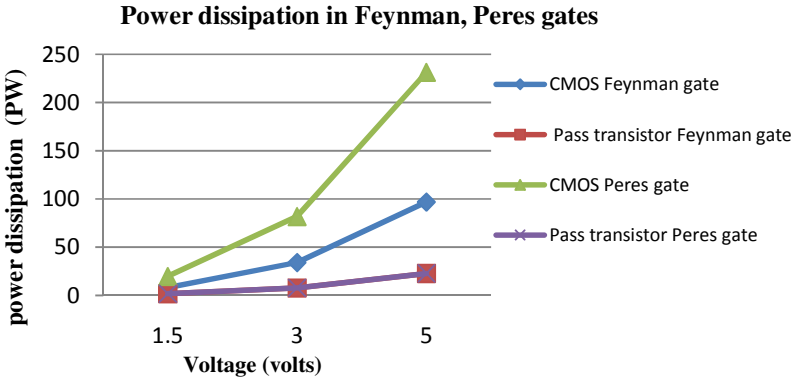

Figure 23. Power analysis of Feynman and Peres gates

The power dissipations of the gates for different voltages are graphically represented in the figure 23 and it is observed that power dissipation of the gates is optimised in pass transistor realisation compared to CMOS technique.

4.2 Simulation results of Fredkin gate and TR gate

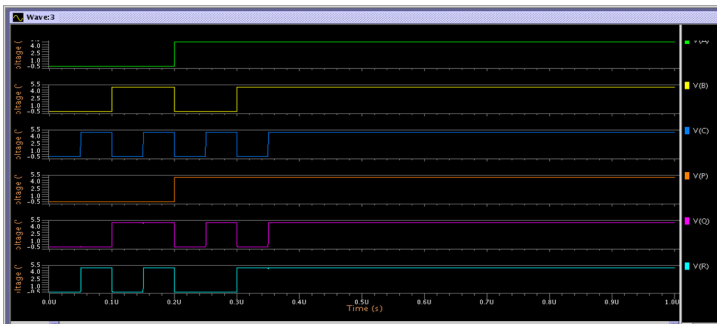

Figure 24. Simulation results of Fredkin gate

The simulation results of Fredkin gate are shown in figure 24. If the inputs are a='1', b='0' and c='1' then the outputs are p='1', q='1' and r='0'.

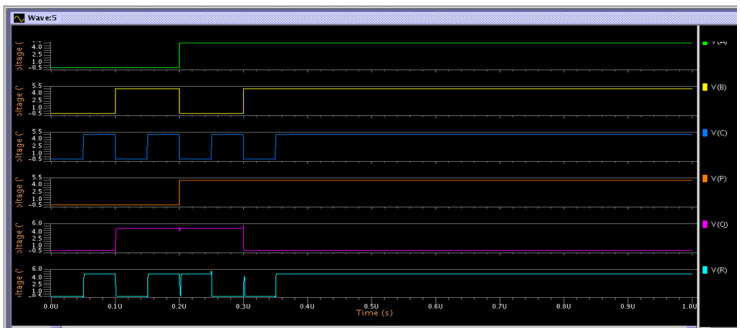

Figure 25. Simulation results of TR gate

The simulation results of TR gate are shown in figure 25. If the inputs are a='1', b='0' and c='0' then the outputs are p='1', q='1' and r='1'.

Table 4. Synthesis results for Fredkin gate using CMOS and pass transistor logic

Reversible gate	Logic family	No of transistors required	L/W of transistor	Voltage applied	Power dissipation(watts)
Fredkin gate	CMOS	13	0.25u/50u	5v	2.6201 m
			0.5u/25u		13.5041n
			2u/10u		126.838 p
			2u/10u	3v	45.960 p
			2u/10u	1.5v	11.680 p
	PASS transistor logic	6	0.25u/50u	5v	12.509 u
			0.5u/25u		53.8996 p
			2u/10u		22.835 p
			2u/10u	3v	7.7837 p
			2u/10u	1.5v	1.7092 p

Table 5. Synthesis results for TR gate using CMOS and pass transistor logic

Reversible gate	Logic family	No of transistors required	L/W of transistor	Voltage applied	Power dissipation (watts)
TR gate	CMOS	31	0.25u/50u	5v	72.2086 m
			0.5u/25u		27.0169 n
			2u/10u		244.125 p
			2u/10u	3v	86.566 p
			2u/10u	1.5v	20.9683 p
	PASS transistor logic	12	0.25u/50u	5v	12.509 u
			0.5u/25u		53.9005 p
			2u/10u		22.835 p
			2u/10u	3v	7.7837 p
			2u/10u	1.5v	1.7092 p

Table 4 and table 5 give the comparisons between the CMOS and pass transistor realisations of Fredkin gate and TR gate respectively. The power dissipations of the gates for different voltages are graphically represented. By observing the figure 26, it is known that power dissipation of the gates is optimised in pass transistor realisation compared to CMOS technique.

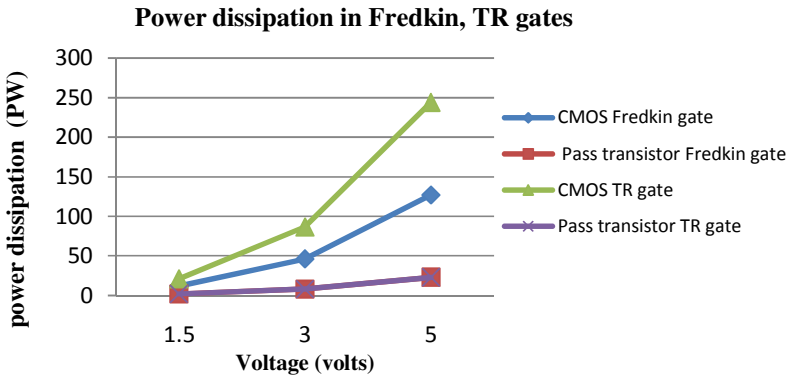

Figure 26. Power analysis of Fredkin, TR gates

4.3 Physical realisation of one-bit full adder/subtractor

The figure 27 gives the simulation design of one-bit adder/subtractor. If the control input is '0', then it performs addition operation and if it is '1' subtraction is performed.

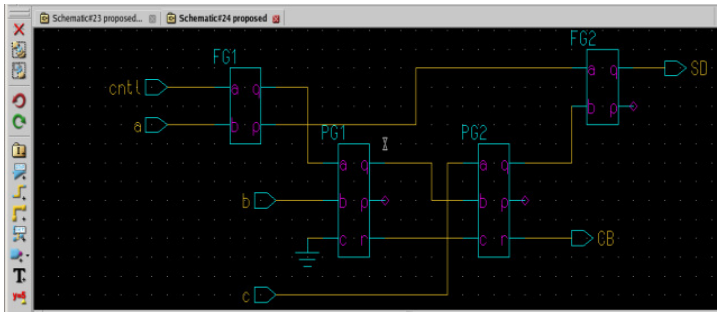

Figure 27. Design 1 simulation results of one-bit full adder/subtractor

The simulation design of another one-bit adder/subtractor is shown in figure 28. If the control input is '1' the circuit performs addition operation and if it is '0' subtraction is performed.

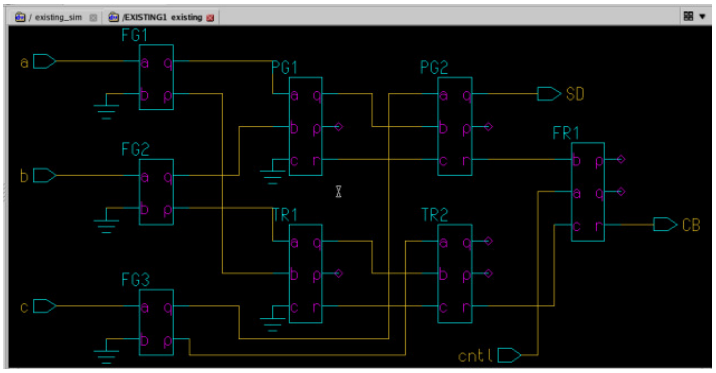

Figure 28. Design 2 simulation results of one-bit full adder/subtractor

The simulation results of adder are shown in below figure 29. If A=0, B=1 and Cin=1 then the outputs are sum=0 and carry=1.

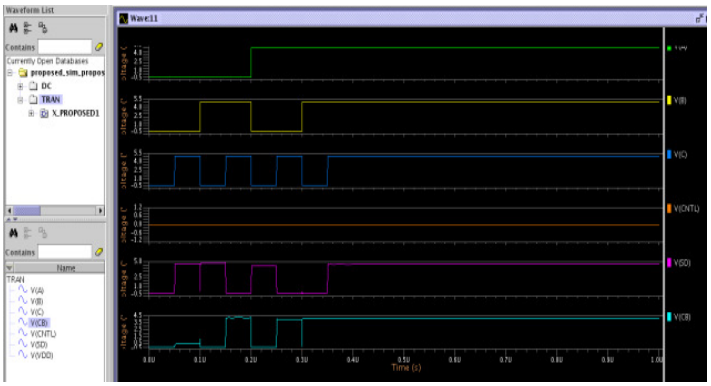

Figure 29. Simulation results of Adder

The simulation results of subtractor are shown in the figure 30. If A='1', B='1' and Cin='1' then the outputs are difference='1' and barrow='1'.

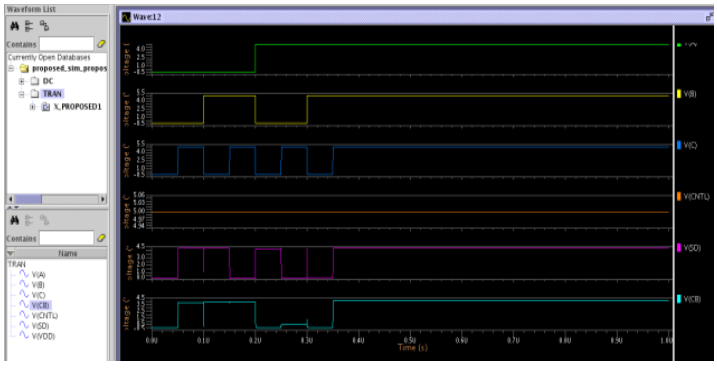

Figure 30. Simulation results of subtractor

Table 6. Synthesis results of one-bit adder/subtractor circuits for supply voltage 5V

Reversible circuit	Logic family	No of transistors required	Length/width of transistor	Power dissipation (watts) adder	Power dissipation (watts) Subtractor
adder/subtractor design 1	CMOS	194	0.5u/25u	143.1620 n	146.258 n
			2u/10u	1.367 n	1.3680 n
	Pass transistor	72	0.5u/25u	1.4283 n	613.7312 p
			2u/10u	185.6219 p	184.3425 p
adder/subtractor design 2	CMOS	88	0.5u/25u	68.144 n	82.680 n
			2u/10u	696.199 p	656.185 p
	Pass transistor	36	0.5u/25u	276.6720 p	9.9716 n
			2u/10u	90.5589 p	86.8545 p

Table 6 gives the comparisons between the CMOS and pass transistor realisations of one-bit full adder/subtractor designs. The power dissipations of the adders and subtractors for different transistor sizes are represented in figures 31 and figure 32 respectively.

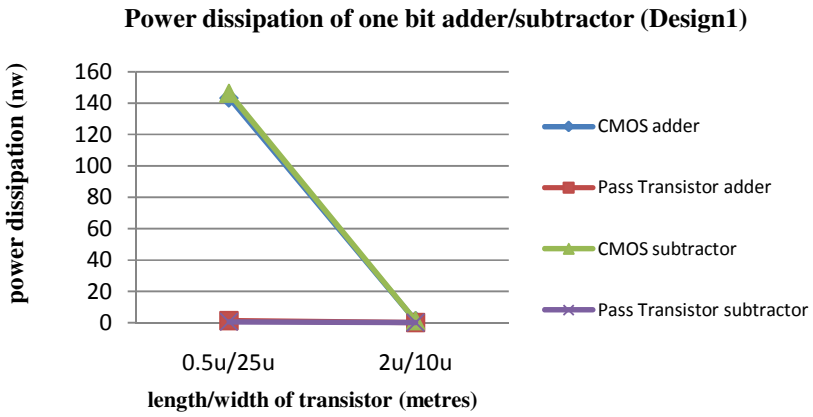

Figure 31. Transistor sizes Vs power dissipation of design1one bit full adder/subtractor

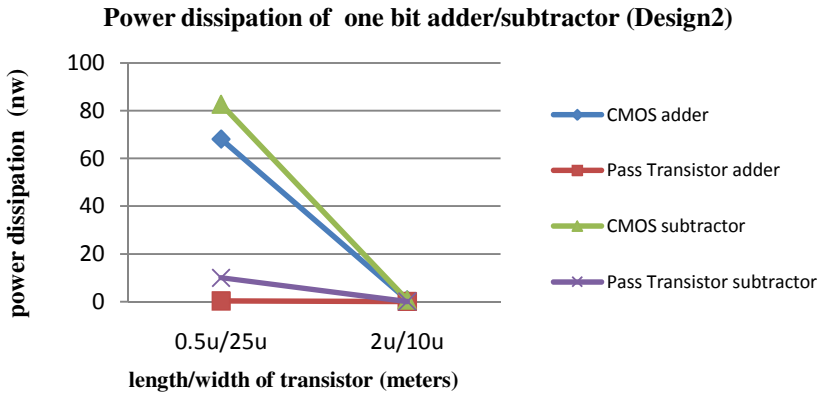

Figure 32. Transistor sizes Vs power dissipation of design2 one bit full adder/subtractor

From the figures, it is known that power dissipations of the circuits are optimised in pass transistor realisation compared to CMOS technique.

4.4 Simulation results of four bit ripple carry adder/subtractor

Figure 33 gives the simulation design of four-bit ripple carry adder/subtractor. If the control input is '0' four bit addition is performed and if the control is '1' four bit subtraction is performed.

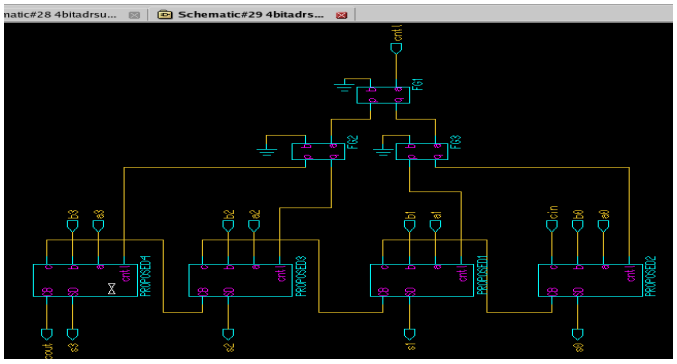

Figure 33. Simulation design of four-bit ripple carry adder/subtractor

The simulation results are shown in figure 34. If A= "1100", B= "1001", Cin='0' and ctrl='0', then addition is performed and the result is "10101". The MSB is the carry out. If A="1100", B= "1001", Cin='0' and ctrl='1', then subtraction is performed and the result is "00011". The MSB is the barrow out.

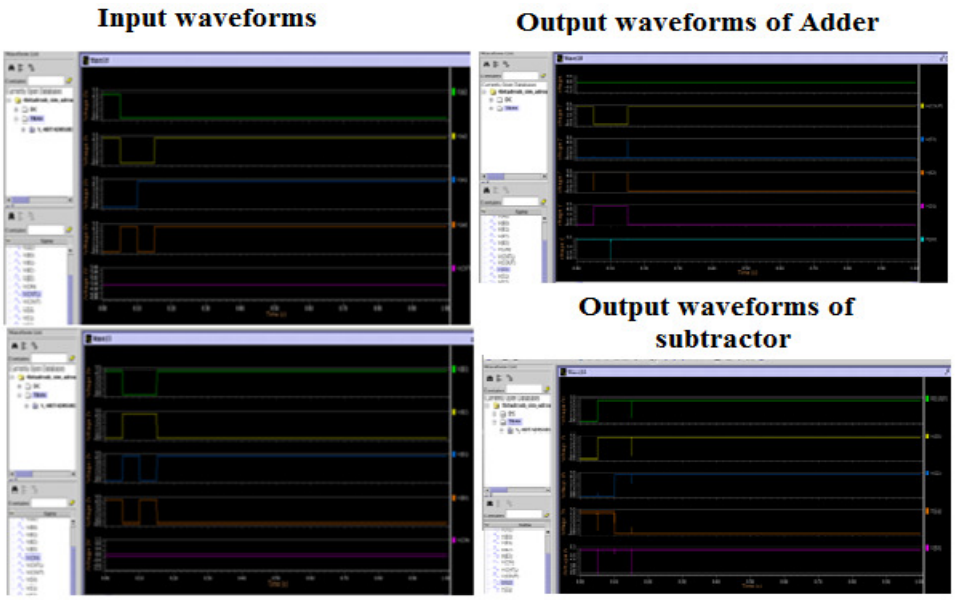

Figure 34. Simulation results of four-bit ripple carry adder/subtractor

Table 7 gives the comparisons between the CMOS and pass transistor realisations of four-bit ripple carry adder. The power dissipation of the adder is optimised in pass transistor realisation compared to CMOS technique.

Table 7. Synthesis results of four-bit adder/subtractor for different supply voltages

Reversible circuit	Logic family	No of transistor s required	Length/width of transistor	Voltage applied	Power dissipation (watts) adder	Power dissipation (watts) subtractor
Four bit adder/subtractor	CMOS	391	0.25u/50u	5v	1.9677	1.9263
			0.5u/25u		360.156 n	356.597 n
			2u/10u		3.046 n	3.116 n
			2u/10u	3v	1.120 n	1.025 n
			2u/10u	1.5v	279.3899 p	268.525p
	PASS transistor logic	162	0.25u/50u	5v	118.564 m	260.271 m

4.4 Simulation results of Eight-bit carry skip adder

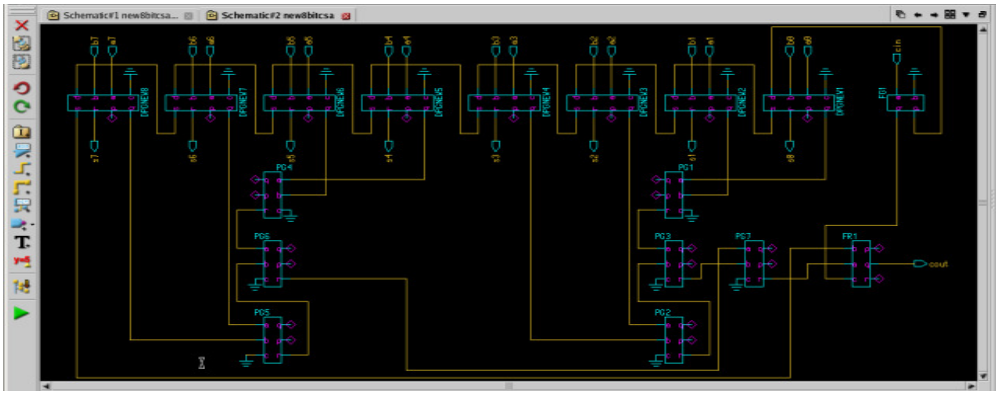

Figure 35. Simulation design of eight bit CSA

Figure 35 gives the simulation design of eight-bit carry skip adder using Double Peres Gate. The simulation results are shown in the figure 36.

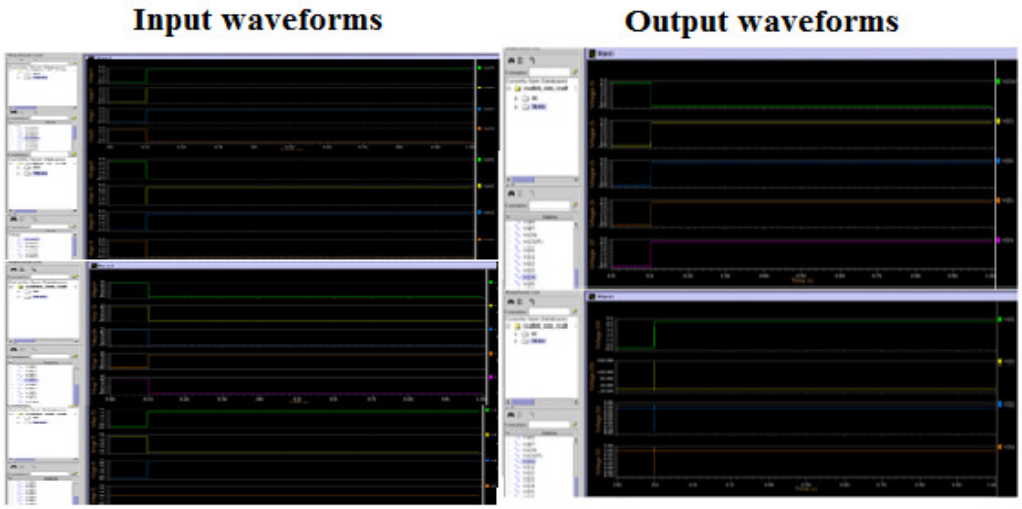

Figure 36. Simulation results of eight-bit Carry Skip Adder

If the operand A is given as "00011001" and operand B is given as "11101010" and Cin as '0', then the output of CSA is "100000011" as shown in the figure. The MSB is the carry out bit of CSA. Table 8 gives the comparison between the conventional and reversible logic realisations of eight-bit carry skip adder.

Table 8. Comparison table for eight-bit carry skip adder

Design	Logic	No of transistors required	Length/width of transistor	Voltage applied	Power dissipation(watts)
Eight bit carry skip adder	Conventional	486	0.25u/50u	5v	2228.67 m
	Reversible logic	268	0.25u/50u	5v	232.67 m

The number of transistors required for the design is less in reversible logic, so the designing area of the circuit is reduced and as per Landauer's principle, the power dissipations of reversible circuits are low compared to conventional designs. By observing the table it is observed that power dissipation of the circuit is optimised in reversible logic compared to conventional Carry Skip Adder.

5. CONCLUSION

The reversible logic gates and circuits are implemented in transistor level with CMOS and pass transistor logics and compared their performance by varying the transistor length and widths for different supply voltages. A four-bit reversible adder/subtractor is implemented and an eight-bit carry skip adder is verified with the conventional circuit using Mentor graphics backend tools and compared with each other. For different values of length/widths of transistor the outputs of CMOS are good but it uses large number of transistors. Number of transistors and the dissipating power are optimized in pass transistor logic compared with the CMOS logic. But, only for the high values of length/widths of the transistors the pass transistor circuits gives efficient outputs. One of the techniques for implementing the transistor level circuits to overcome these problems is Transmission gate logic.

REFERENCES

[1] R.landauer, "Irreversibility and heat generation in the computational Process", IBM Research and Development, pp. 183-191, 1961.

[2] C.H Bennett "Logical Reversibility of computations" IBM J. Research and development, pp 525-532, November-1973.

[3] Raghava Garipelly, P.MadhuKiran, A.Santhosh Kumar "A Review on Reversible Logic Gates and their Implementation", International Journal of Emerging Technology and Advanced Engineering, Volume 3, Issue 3, March 2013, pp.417-423.

[4] Rashmi S.B, Tilak B G, Praveen B "Transistor Implementation of Reversible PRT Gates" International Journal of Engineering Science and Technology (IJEST) Vol. 3 No. 3 March 2011, pp.2289-2297.

[5] Aakash Gupta, Pradeep Singla, Jitendra Gupta and Nitin Maheshwari, "An Improved Structure of Reversible Adder And Subtractor", International Journal of Electronics and Computer Science Engineering Volume 2, Number 2, pp.712-718.

[6] A.Kamaraj , I.Vivek Anand , P.Marichamy, "Design of Low Power Combinational Circuits using Reversible Logic and Realization in Quantum Cellular Automata" International Journal of Innovative Research in Science, Engineering and Technology,Volume 3, Special Issue 3, March 2014, pp.1449-1456.

[7] V.Kamalakannan, Shilpakala.V, Ravi.H.N, "Design of Adder / Subtractor Circuits Based on Reversible Gates" Ijareeie, Vol. 2, Issue 8, August 2013, pp.3796-3804.

[8] Rangaraju H G, Venugopal U, Muralidhara K N and Raja K B, "Low Power Reversible Parallel Binary Adder/ Subtractor".

[9] M.SinghSankhwar "Design of High Speed Low Power Reversible Adder Using HNG Gate", International Journal of Engineering Research and Applications, Vol. 4, Issue 1(Version 2), January 2014, pp.152-159.

[10] D. P. Bala Subramanian, K.Kalaikaviya and S.Tamilselvan, "Low-Geometry High Speed Feyman Toffoli 8 Bit Carry Skip Adder", Journal of Global Research in Electronics and Communication Volume 1, No. 1, November-December 2012, pp. 6-9.

[11] P. K. Lala, J.P. Parkerson, P. Chakraborty, "Adder Designs using Reversible Logic Gates ", WSEAS Transactions on Circuits And Systems, Volume 9, Issue 6, June 2010, pp. 369-378.

[12] HimanshuThapliyal, A.P Vinod "Transistor Realization of Reversible TSG Gate and Reversible Adder Architectures", IEEE Asia Pacific Conference on Circuits and Systems, APCCAS 2006, pp.418-421.

AUTHORS

Prudhvi Raj.K pursuing M.Tech in the branch of Digital Electronics and Communication Systems at Gudlavalleru Engineering College and received B.Tech degree in Electronics and Communication Engineering from Prakasam Engineering College in the year of 2011.

Syamala.Y received her B.E., M.E., from Bharathiyar University, Anna University in 2001, and 2005 respectively. She obtained Ph.D from JNTUH, Hyderabad in 2014. She has been a member of IEEE, FIETE and MISTE. She has published several papers in the area of VLSI. Her research interest includes Low power VLSI, Digital design and Testing.