DESIGN OF NOVEL 4- BIT ALU USING VERILOG HDL

[1]R.Mahesh Kumar,[2]Dr.G.Ramprabhu

[1]Assistant Professor, [2]Professor,

[1]ECE Department

Annamacharya Institute of Technology and sciences, Rajampet, India.

Abstract : The paper primarily aims at reducing the power of the adder and multiplier modules that are critical functional units of ALU in order to reduce overall energy consumption without sacrificing the processor speed. The ALU circuit only ensures that arithmetic or logic operations are performed at once, ensuring low power consumption by a single set of circuits. The whole ALU circuit is rendered by Verilog. The key logic circuits we will study are the adders. Adders form a mandatory part of all modern integrated circuits. The prerequisite of an adder is that its energy consumption and the chip area are mainly fast and secondarily efficient. The topology used in this work is, carry look-ahead adder, ripple carry adder, carry skip adder, carry select adder, for incrementation, saving an adder and carrying an bypass adder. Verilog compares the module functionality and performance issues, such as field, power dissipation and propagation delay. Each processor's output depends on its power and time. To get an efficient processor, the power and delay should be less. The most widely used architecture in processors is a multiplier. If multiplier power and delay are that, the effective processor can be produced.

Key words—, adder, ALU Verilog HDL, delay, power processors, chip area.

I. INTRODUCTION

A central processing unit (CPU), memory card and input and output unit are the basic components of a device. The computer's Processor is practically the same as a human's brain. It includes all registers, the control unit and the ALU. ALU is known as the most powerful digital computer subsystem. An arithmetical logic unit (ALU) is a digital circuit that operates arithmetically, logically and change in two digital words per n-bit. The ALU can be functionally divided into three circuits: the numerical circuit, the logical circuit and the move circuit.

The nucleus of a microprocessor is an arithmetic logic unit (or ALU). As its name suggests, a number of arithmetic and logical operations can be performed. Many operations (e.g. addition) operate on two binary inputs A and B to generate a binary output. Some (e.g. increase or decrease) function on one input. Therefore, an ALU typically has two binary inputs and one binary output for data processing. It also has a binary input for selecting the appropriate operation. You can create a simple ALU from your 4bitadd part by adding a single control line to make it add and subtract.

ALUs have two main functional components-the Logic block and the Arithmetic block. Arithmetic block is used for performing the addition, subtraction, and contrast of arithmetic operations. An adder is the center of the arithmetic unit. Logic block is used for simple logic operations such as AND, OR and XOR.

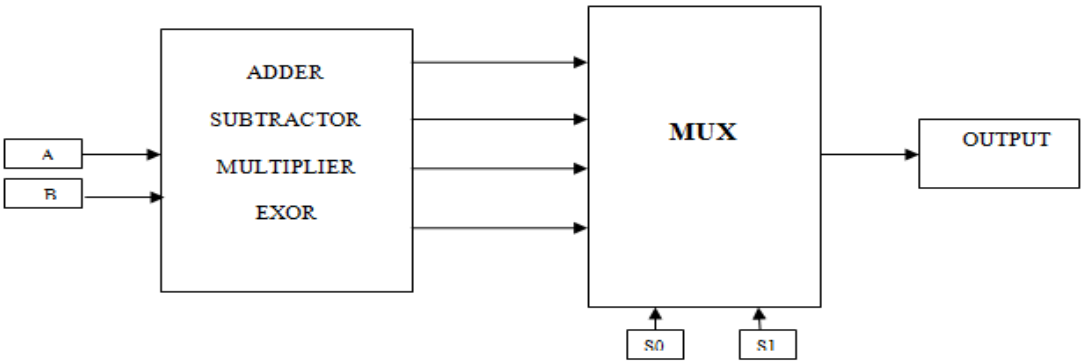

Fig.1 4-bit ALU block

A schematic of the 4-bit ALU block is shown above. The ALU consists of four stages, with two components in each stage: a) multiplexers b) complete adder. The ALU performs the next addition, subtraction, multiplication of arithmetic operations. The logical operation is EXOR.

There are multiplexers in the output parts. The multiplexers were equipped with the logic of the transfer transistor. A sequence of three select signals was included in the design to assess the operation and to select inputs and outputs. The figure shows the 4-bit ALU that cascades the CARRY bit from stage one to stage four. In the figure the ALU architecture consists of addresses, subtractors, propagators, EXOR gates. The 4-bit ALU has been built in a transistor level diagram. All multiplexers were implemented using transistors and the full adder alone was equipped with transistors.

The multiplexer stage selects the correct inputs based on the state of the selected signals and gives it to the entire adder. The total adder measures the results. The multiplexer selects the correct output at the output stage and sends it out. The table shows the truth table for the

ALU's operations depending on the identified signals status. The top module has a four-bit ALU. CMOS ALU has four feature instructions: addition, subtraction, multiplication, XOR. Every operation is performed bitwise on two four-bit inputs. Bit by bit, the 4X1 multiplexer is made. That block of the single bit is cascaded into a four bit ALU.

1. Topologies

Each section describes and explains the different topologies used in the ALU work.

We use the various types of adders and subtract the different adders that we used:

- Ripple Carry Adder
- Carry Save Adder
- Carry Look-Ahead Adder
- Carry Increment adder
- Carry Skip Adder
- Carry Bypass Adder
- Carry Select Adder

From these adders select the best adder by using the VERILOG code to test the delay.

- Array Multiplier
- Wallace tree Multiplier
- Booth Multiplier
- Modified Booth Multiplier

The results of these topologies are tested for region robustness, delay and power dissipation. You are selected for this work because they are widely used in many applications

3.Arithmetic Logic Unit Design

The central processing unit (CPU) is based on a core component of the Arithmetic Logic Unit (ALU). The adders, subtractors, multipliers and a door must be used in the ALU design. In this case we used the best adders and multipliers to build the appropriate circuit. The addition and multiplier selection is rendered here based on the contrast of the others with the delay. ALU is a digital circuit performing between two numbers an arithmetic (adding, subtraction, etc.) and logic operations (exclusive-OR, AND, etc). Demand for low-power consumption efficiency in today's processor has put severe limitations on ALU design. Adders are essential components of any arithmetic circuit and often on the critical path. There are several hierarchical layers that can boost additional operations. Next, architecture can be changed. The first and most fundamental adder is a Ripple carry adder. Using a faster propagation process, for example prefix tree adder schemes. The adder Kogge-Stone and Han-Carlson can minimize delay.

The Adder of Kogge-Stone[10] is a simultaneous suffix for bearing an adder. The time taken to produce the transmission signals is equal to the level number and O (log2N). It is widely considered to be the fastest possible adder style. It takes more field to introduce, but at every point it has a lower fan out, which improves efficiency. The Han-Carlson adder incorporates the combining tree of Brent-Kung and KoggeStone Carry for a change in pace and region respectively. The time to measure is just one more step than the Kogge-Stone adder.

ALU acts as an essential part of the central processing unit (CPU) data processing system. It is also the main actor in any electronic system. ALU is a multifunctional circuit, providing one of the few functions that can be implemented with two operands A and B, depending on the controls.

Data shift from address decoding to machine arithmetic is needed for many main computer operations. For communication applications such as encryption and error control coding, a cyclic shifter is necessary for rotating operations. As regards design style, Stack, Barrel and Logarithmic shifters provide three styles of shifters for circuit designers.

A shifter array decodes the shift value in individual shift bit lines that match all data values. But it took an additional decoder. It's also getting nuanced and too sluggish for higher shift values. A logarithmic shifter is based on an approach to the point. The cumulative change value is broken down into shifts with two forces. No. For the maximum shift width M, the total stages are log2 M stages where the ith stage varies over 2i or passes the data unchanged. It occupies a small area and needs no decoder, however the pass transistor serial connection slows down the shifter for larger shift values.

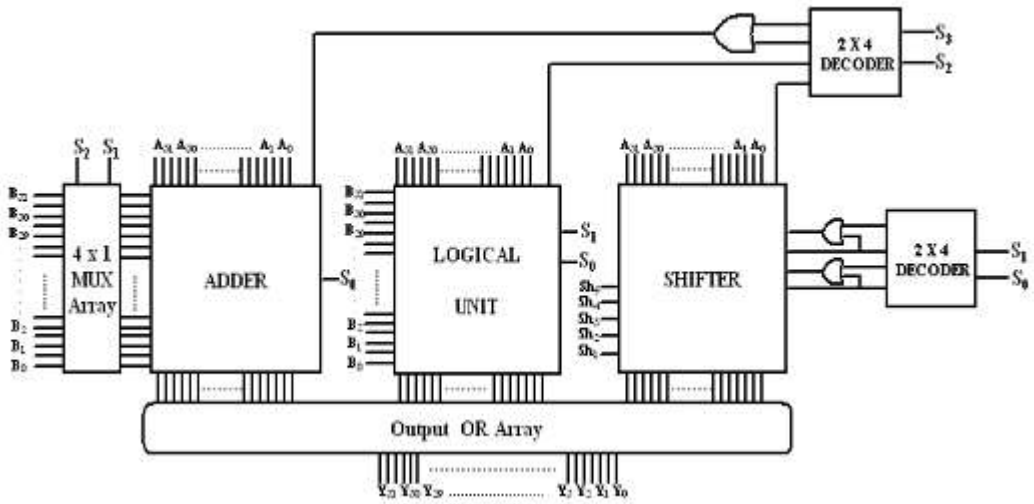

Figure 2. Implemented Design of ALU

The ALU is equipped as a control unit, an arithmetical unit, a logical unit and a shifter unit in four parts. An implemented ALU A 2x4 decoder architecture is used as a control unit for the selection of the different units with control signals S3 and S for desired operation. Table II displays a detailed table of the planned ALU. The arithmetic unit consists of a multiplexer unit and 4x1. The addition unit is used for two operand add-ons while the multiplexer unit selects the correct input operand for the adder unit in compliance with the control signal S2 and S. The 2x4 decoder with S1and S control signals selects various logical and shifting operations. Shifting values are set by Sh05, Sh4, Sh3, Sh2 and Sh for the shifter array. Such control signals are binary weighted for the shifter function. Hence the overall change value. All the outputs of the various ALU units are eventually combined with the OR output array, where three input OR gates are accompanied by an' Y' buffer.

Table 1. Functional Table of the designed ALU

Operation Select				Operation	Function
S_3	S_2	S_1	S_0		
0	0	0	0	Addition	$F = A + B$
0	0	0	1	Addition with carry	$F = A + B + 1$
0	0	1	0	Subtraction with barrow	$F = A + \bar{B}$
0	0	1	1	Subtraction	$F = A + \bar{B} + 1$
0	1	0	0	Decrement	$F = A - 1$
0	1	0	1	Transfer	$F = A$
0	1	1	0	Transfer	$F = A$
0	1	1	1	Increment	$F = A + 1$
1	0	0	0	A AND B	$F = A \cdot B$
1	0	0	1	A OR B	$F = A + B$
1	0	1	0	A XOR B	$F = A \oplus B$
1	0	1	1	Complement A	$F = \bar{A}$
1	1	0	0	Logical Left Shift	
1	1	0	1	Circular Left Shift	
1	1	1	0	Logical Right Shift	
1	1	1	1	Circular Right Shift	

3. Results & Discussion

The outcome of the simulation using XILINX is shown in Figure 3 and the output represents RTL schematic of the modeled ALU and can be checked by the simulation.

Table 2. Device Utilization Summary of Designed ALU

Logic utilization	**Used**	**Available**	**Utilization**
Number of 4 input LUTs	56	9,312	1%
Number of occupied slices	29	4,656	1%
Number of slices containing only related logic	29	6	100%
Number of slices containing unrelated logic	0	6	0%
Number of bonded IOBs	19	232	8%
Average Fanout of non-clock nets	3.82		

Figure 3. RTL schematic of proposed ALU

The performance of ALU simulated using MODELSIM for the proposed ALU is shown in Figure 4. The generated waveforms follow the Table 1.

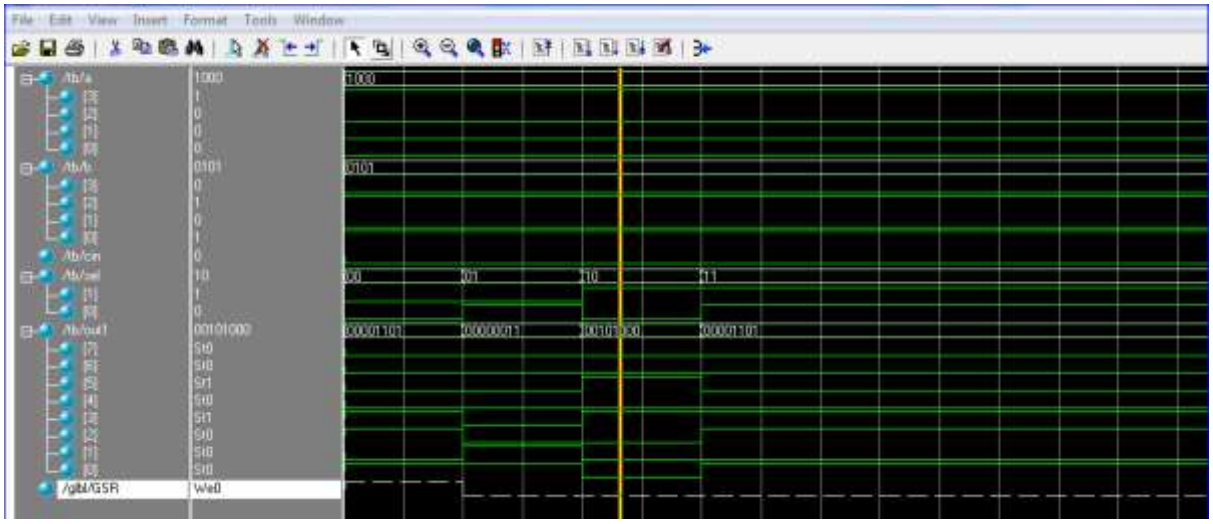

Figure 4. Behavioral Simulation Result for the designed ALU

```
Delay:               12.737ns (Levels of Logic = 10)
  Source:            a<1> (PAD)
  Destination:       out1<4> (PAD)

  Data Path: a<1> to out1<4>
                                Gate     Net
    Cell:in->out      fanout    Delay    Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             18    1.218    1.243  a_1_IBUF (a_1_IBUF)
     LUT3:I0->O             5    0.704    0.668  Madd_old_w5_18_addsub0000_cy<2>11
(Madd_old_w5_18_addsub0000_cy<2>)
     LUT4:I2->O             1    0.704    0.000  Mmux__old_w1_1971 (Mmux__old_w1_197)
     MUXF5:I1->O            1    0.321    0.595  Mmux__old_w1_197_f5 (_old_w1_19<3>)
     LUT2:I0->O             1    0.704    0.000  Madd__add0001_lut<3> (Madd__add0001_lut<3>)
     MUXCY:S->O             1    0.464    0.000  Madd__add0001_cy<3> (Madd__add0001_cy<3>)
     XORCY:CI->O            1    0.804    0.595  Madd__add0001_xor<4> (_add0001<4>)
     LUT2:I0->O             1    0.704    0.000  Mmux_out1<4>21051 (Mmux_out1<4>2105)
     MUXF5:I1->O            1    0.321    0.420  Mmux_out1<4>2105_f5 (out1_4_OBUF)
     OBUF:I->O                   3.272           out1_4_OBUF (out1<4>)
    ----------------------------------------
    Total                      12.737ns (9.216ns logic, 3.521ns route)
                                        (72.4% logic, 27.6% route)
```

Figure 5. Delay of Proposed ALU

Modes of Operation

1. **Addition Operation**

The designed ALU is simulated in one of its mode of operation and its results are shown below.

Inputs:

A = 1000
B = 0101
Cin = 0
S_0 S_1 = 00

Sum = 00001101
Cout = 0

Implementation:

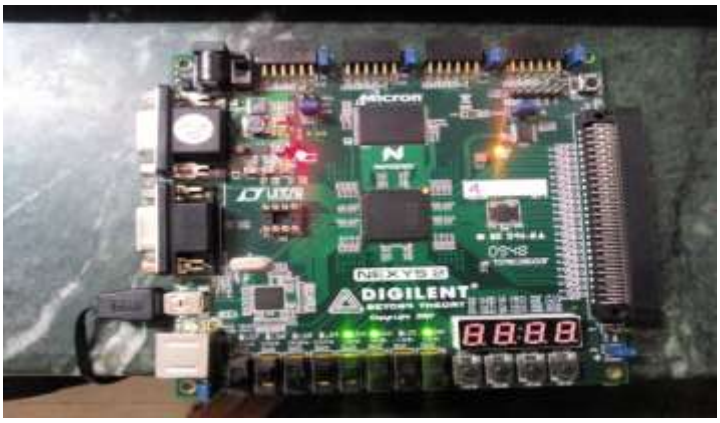

Figure 6. FPGA implementation of ALU operating in addition operation

2. Subtraction Operation

The designed ALU is simulated in one of its mode of operation and its results are shown below.

Inputs:

A = 1000
B = 0101
Cin = 0
S_0 S_1 = 00
Sum = 00001101
Cout = 0

Implementation:

Figure 7. FPGA implementation of ALU operating in subtraction operation

3. Multiplication Operation

The designed ALU is simulated in one of its mode of operation and its results are shown below.

Inputs:

A = 1000
B = 0101
Cin = 0
S_0 S_1 = 00
Sum = 00001101
Cout = 0

Implementation:

Figure 8. FPGA implementation of ALU operating in multiplication operation

4. **EXOR Operation**

The designed ALU is simulated in one of its mode of operation and its results are shown below.

Inputs:

A = 1000
B = 0101
Cin = 0
S_0 S_1 = 00
Sum = 00001101
Cout = 0

Implementation:

Figure 9. FPGA implementation of ALU operating in EXOR operation

Displayed in Figure 3. is the outcome when the CADENCE simulation is used and the output shows the RTL schematic of the ALU design and the adder circuits feature can be tested by simulation.

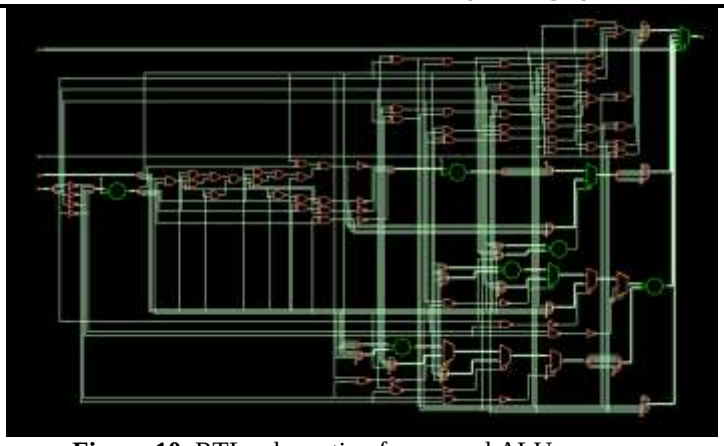

Figure 10. RTL schematic of proposed ALU

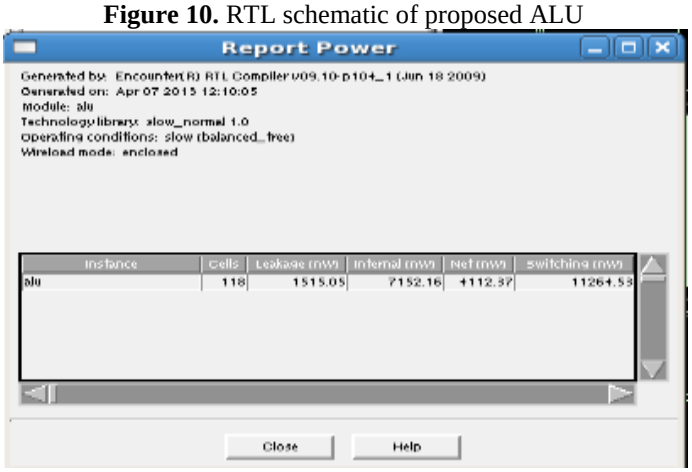

Figure 11. Power Report of Proposed ALU

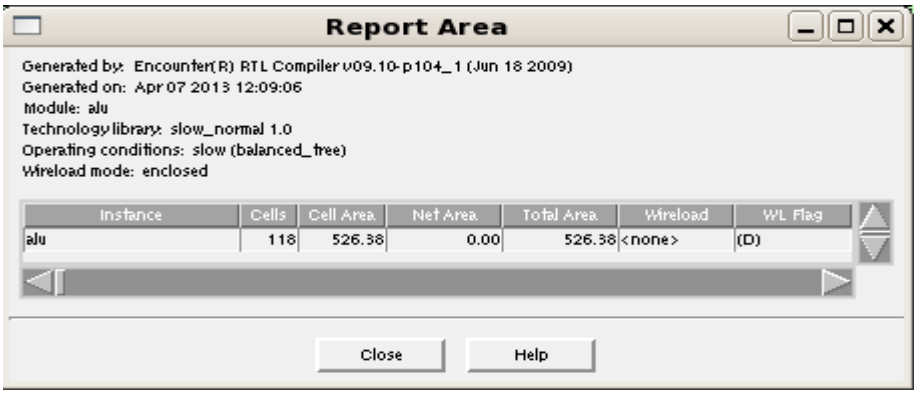

Figure 12. Area report of proposed ALU

5. Conclusion

Until now, we have compared and selected different adder and multipliers in terms of the field, power and delay and the one with a less area, delay, and lower power consumption by using this adder and multiplier. The tests are compared by Xilinx and cadence software for two ALUs.

References

1. R. H. Krambeck, C. M. Lee and H. F .S. Law," High speed compact circuits with CMOS," IEEE J. Solid State Circuits, vol. 17, no.3, June1982, pp.614-619.
2. Lawrence G. Heller and William R. Griffin, "Cascode voltage switch logic: A differential CMOS logic family," Solid-State Circuits Conference. Digest of Technical Papers. 1984 IEEE International, vol.27, pp. 16-17, Feb 1984.
3. K.M. Chu and D. L. Pulfrey, "A comparison of CMOS circuit techniques: differential cascade voltage switch logic versus conventional logic,"IEEE J. Solid State Circuits, vol. 22, no.4, Aug.1987, pp.528-532.
4. Leo C.M.G. Pfennings, Wim G.J. Mol, Joseph J.J. Bastiaens and Jan M.F. van Dijk, "Differential split-level CMOS logic for sub-nanoseconds speeds," Solid-State Circuits Conference. Digest of Technical Papers. 1985 IEEE International, vol. 28, no., pp. 212-213, Feb 1985.
5. Yadlapati, A., Kakarla, H.K. An Advanced AXI Protocol Verification using Verilog HDL (2015) Wulfenia, 22 (4), pp. 307-314.
6. Bindu Bhargavi, K., Hari Kishore, K. Low Power Bist on Memory Interface Logic (2015) International Journal of Applied Engineering Research, 10 (8), pp. 21079-21090.
7. Charan, N.S., Kishore, K.H. Recognization of delay faults in cluster based FPGA using BIST (2016) Indian Journal of Science and Technology, 9 (28).
8. Hari Kishore, K., Aswin Kumar, C.V.R.N., Vijay Srinivas, T., Govardhan, G.V., Pavan Kumar, C.N., Venkatesh, R.V. Design and analysis of high efficient UART on spartan-6 and virtex-7 devices (2015) International Journal of Applied Engineering Research, 10 (9), pp. 23043-23052.
9. Kante, S., Kakarla, H.K., Yadlapati, A. Design and verification of AMBA AHB-lite protocol using Verilog HDL (2016) International Journal of Engineering and Technology, 8 (2), pp. 734-741.
10. Bandlamoodi, S., Hari Kishore, K. An FPGA implementation of phase-locked loop (PLL) with self-healing VCO (2015) International Journal of Applied Engineering Research, 10 (14), pp. 34137-34139.
11. Murali, A., Hari Kishore, K., Rama Krishna, C.P., Kumar, S., Trinadha Rao, A. Integrating the reconfigurable devices using slow-changing key technique to achieve high performance (2017) Proceedings - 7th IEEE International Advanced Computing Conference, IACC 2017, art. no. 7976849, pp. 530-534.
12. A. Surendar, K. H. Kishore, M. Kavitha, A. Z. Ibatova, V. Samavatian “Effects of Thermo-Mechanical Fatigue and Low Cycle Fatigue Interaction on Performance of Solder Joints” IEEE Transactions on Device and Materials Reliability, P-ISSN: 1530-4388, E-ISSN: 1558-2574, Vol No: 18, Issue No: 4, Page No: 606-612, December-2018.
13. N Bala Dastagiri K Hari Kishore “A 14-bit 10kS/s Power Efficient 65nm SAR ADC for Cardiac Implantable Medical Devices” International Journal of Engineering and Technology (UAE), ISSN No: 2227-524X, Vol No: 7, Issue No: 2.8, Page No: 34-39, March 2018.
14. N Bala Dastagiri, Kakarla Hari Kishore "Reduction of Kickback Noise in Latched Comparators for Cardiac IMDs” Indian Journal of Science and Technology, ISSN No: 0974-6846, Vol No.9, Issue No.43, Page: 1-6, November 2016.
15. N Bala Dastagiri, K Hari Kishore "Analysis of Low Power Low Kickback Noise in Dynamic Comparators in Pacemakers” Indian Journal of Science and Technology, ISSN No: 0974-6846, Vol No.9, Issue No.44, page: 1-4, November 2016