

Design and Implementation of 8-Bit ALU based on Sub-threshold Adiabatic Logic (SAL)

Ranjith K. G.
 M.Tech, VLSI & Embedded System
 R.V. College of Engineering
 Bengaluru-059, India

Arunkumar P. Chavan
 Asst.Prof. ECE Department
 R.V. College of Engineering
 Bengaluru-059, India

H. V. Ravish Aradhya, PhD
 Prof. ECE Department
 R.V. College of Engineering
 Bengaluru-059, India

ABSTRACT

Sub-threshold Adiabatic Logic (SAL) is a power saving technique used in VLSI circuit design for applications where power saving is of primary importance. An 8-Bit ALU is designed and implemented based on Sub-Threshold Adiabatic Logic (SAL) using Cadence 45nm technology node. Initially basic blocks decoder, adder and multiplexer are designed and tested for their functionality. Designed basic blocks are integrated to design 8-Bit ALU. Designed ALU is simulated for different inputs to check its functionality. Propagation delay, average power delivered by source, and Power Delay Product (PDP) are computed to compare with CMOS implementation of ALU. A 40db improvement in relative power consumed is observed in SAL compared to CMOS logic. Physical design of the ALU is built to compute chip area of the designed ALU and the total chip area observed is 32,190 μm^2 with area utilization of 89%..

Keywords

Sub threshold Adiabatic Logic (SAL), Low power, ALU.

1. INTRODUCTION

The increase in demand for low power electronics day by day has increased the demand for research in different low power techniques for designing low power VLSI circuits. These days the main demand of handheld portable devices is long battery life which can be achieved by designing VLSI circuits using various low power techniques. Sub-threshold Adiabatic Logic (SAL) is one such technique under research which can be used to design ultra-low power circuits for devices where low power consumption is of primary importance^[1]. Implanted portable Bio-medical applications are applications which demand for increased battery life which can be achieved by designing these devices using circuits designed using low power techniques. In case of Adiabatic logic energy consumed by the circuit is restored back to the source to reduce the overall drawn energy. In case of Sub-threshold logic, the supply voltage V_{DD} given to the circuits is lower than V_T (Threshold voltage). Here sub-threshold leakage current acts as operating current. This leakage current acts as switching current and since its value is very low it takes a lot more time for the load capacitances to charge. This degrades the performance of the circuits. Sub-threshold Adiabatic Logic (SAL) is obtained by operating the adiabatic circuits in sub-threshold region to obtain the benefits of both the logics^[8]. Here in this paper an 8-Bit ALU is designed and implemented using Sub-threshold adiabatic logic to compare and study its performance with its CMOS counterpart.

1.1 Literature Survey

Literatures on adiabatic logic and sub-threshold logic are studied and the concepts behind these logics are

understood^{[3][4]}. Comparison between these two logics and the advantages of combining adiabatic logic with sub-threshold logic to obtain Sub-threshold Adiabatic Logic is studied. Implementation of basic logic circuits is also observed from the literatures^[2].

1.2 Adiabatic Logic

In thermodynamics, any process is adiabatic if there is no exchange of energy between the surroundings.^[2] In logic circuits, as a result of switching of states energy is dissipated as heat. Adiabatic logic circuits aim at reducing the amount of power dissipation as heat in the circuits and also restore the energy supplied by the circuits. In case of CMOS logic energy stored in the load capacitors is discharged to the ground where as in case of adiabatic logic circuits energy stored in load capacitors is restored back to the source.

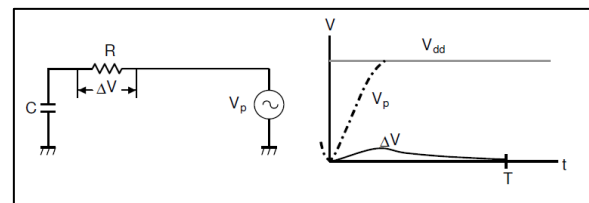


Figure 1: RC Tree Model with Adiabatic charging plot

In the plot of Voltage (V) vs. Time (t) in Figure 1, the value of ΔV is small throughout the clock period. This shows that the energy dissipated as heat is less in case of adiabatic circuits. The mathematical expression for power dissipation in adiabatic circuits is given by Equation (1)^[5]

$$E_{\text{adia}} = (RC/T)CV^2 \dots \dots \dots (1)$$

And the dissipated power in CMOS logic circuits is given by Equation (2)^[5]

$$E_{\text{CMOS}} = (1/2)CV^2 \dots \dots \dots (2)$$

From the above two equations it is inferred that by increasing the period of supply more than $2 \times R \times C$ the dissipated energy in case of adiabatic circuits can be made less than their CMOS counterparts.

1.3 Sub-threshold Logic Operation

In the analysis of the MOS transistor model, the drain current is assumed to be Zero when the voltage between gate and source (V_{GS}) is lower than V_T (Threshold voltage), But there will be still a very small current in a transistor. This current is called sub-threshold leakage current. This leakage current acts as the operating current in case of Sub-threshold logic circuits. The mechanism behind this current generation is explained below.

At this low voltage between gate and source, the electron rich source releases electrons to the region under the gate. Drain will also release electrons if the voltage between source and drain is zero^[6].

After the release of electrons from both source and drain terminals the concentration of electrons under the gate will be uniform as the three terminals are in equilibrium. Uniform electron concentration will be maintained below the Gate (G) terminal of MOS Transistor^[6].

If a small positive voltage is applied at the drain terminal, the density of electrons spilled into the region below gate at the drain end of the MOS transistor will be less compared to the electrons spilled at the source end. To maintain the equilibrium concentration the electrons will move from source end to drain end below the gate terminal. Since drain terminal is maintained at a small positive voltage few electrons move into the drain terminal which results in small drain current. The magnitude of this current is very small as the movement of electrons from source end to drain end below gate terminal is very slow. This small current is called sub-threshold leakage current^[6].

Making the adiabatic circuits to operate in sub-threshold domain, that is by providing supply voltage lower than V_T (Threshold voltage) to obtain Sub-threshold Adiabatic Logic (SAL) whose operating current is sub-threshold leakage current.

2. 8-BIT SAL-ALU ARCHITECTURE

Proposed 8-Bit SAL Architecture for studying the Sub-threshold Adiabatic Logic (SAL) performance is shown in Figure 2. It is divided into sub-blocks like supply decoder, Arithmetic unit, Logical Unit and Output Mux. ALU takes

two 8-bit inputs carry-in input and gives an 8-bit output along with Carry out output. 3-Bit control signal selects the operation to be performed by the ALU.

Supply gating technique is implemented using the supply decoder block. Decoder helps in supplying power to only those blocks which are required to perform the selected operation by the control signals. Other unnecessary blocks will be turned OFF. Power consumed by the ALU is reduced by implementing this technique.

Arithmetic Unit consists of an 8-Bit Look Ahead carry adder for performing arithmetic operations like addition, subtraction, increment and decrement operations on two 8-Bit input signals. XOR block helps in performing 2's complement operation for subtraction and decrement operation. OR block is used to provide unit input of increment and decrement operation.

Logical Unit has four different logical blocks to perform logical operations like NAND, NOR, XOR and NOT on the input signals.

Output Mux block selects the output from the outputs of different blocks performing different operations based on the operation selection control signal and makes it available at the same 8-Bit output terminals of ALU for all the operations selected.

All the sub-blocks inside ALU architecture are initially designed and implemented using Sub-threshold adiabatic logic and are finally integrated to obtain 8-Bit SAL-ALU.

Implementation of the designed ALU in Cadence Virtuoso tool is explained in the next section.

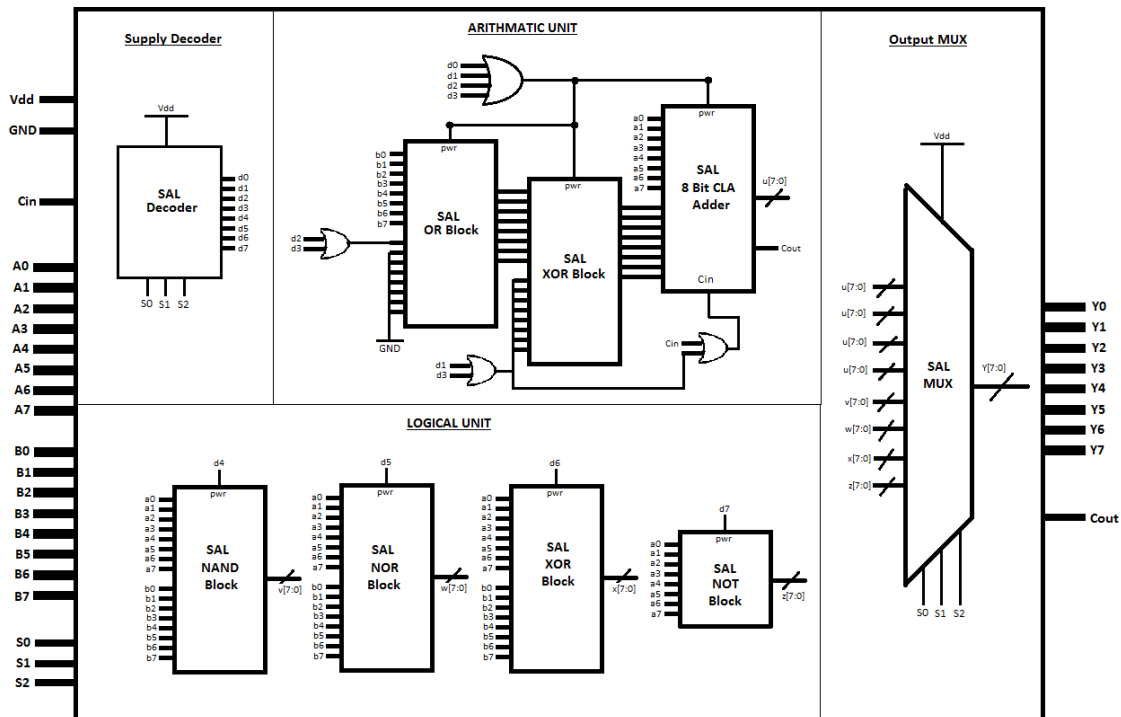


Figure 2: Proposed 8-Bit SAL-ALU Architecture

3. IMPLEMENTATION

Implementation of proposed 8-Bit ALU architecture of Figure 2 in cadence virtuoso schematic editor tool is shown in Figure

3. Table 1 lists the different arithmetic and logical operations performed in the designed ALU.

Table 1: Implemented Operations in proposed ALU

Control Signal			Operations
S0	S1	S2	
0	0	0	Addition
0	0	1	Subtraction
0	1	0	Increment
0	1	1	Decrement
1	0	0	NAND
1	0	1	NOR
1	1	0	XOR
1	1	1	NOT

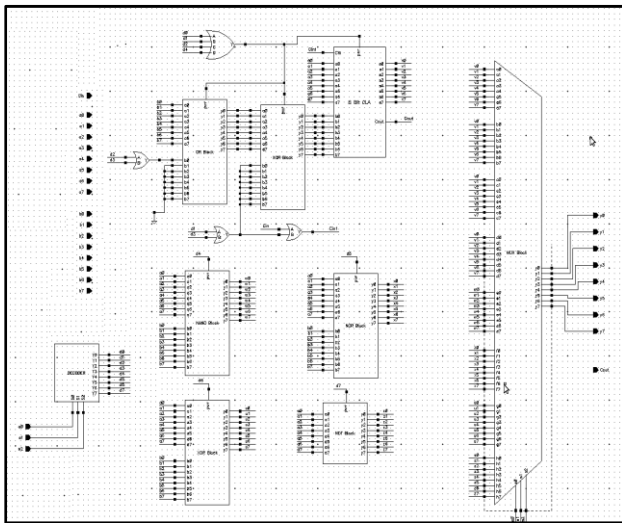


Figure 3: SAL-ALU Schematic in Cadence Virtuoso

Circuits inside the sub-blocks of proposed SAL-ALU designed using basic gates are also shown in this section.

Figure 4 shows the circuit which performs single bit addition. Eight such similar blocks are implemented inside 8-Bit SAL Carry Look Ahead (CLA) sub-block of ALU architecture to perform 8-Bit addition and subtraction operations

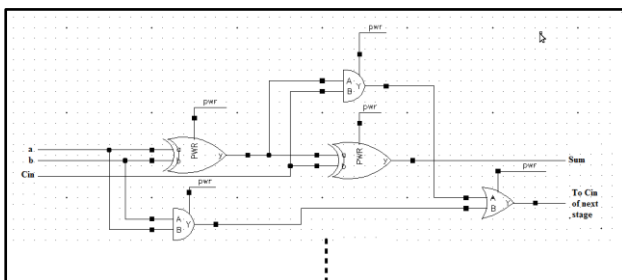


Figure 4: Single Bit Adder in Cadence Virtuoso

Circuit inside the supply decoder block implemented in cadence virtuoso schematic editor is shown in Figure 5. Here s2,s1,s0 are the control signal inputs and d0-d7 are the decoder outputs which provides power supply to different blocks of ALU based on the operation selected by control signals.

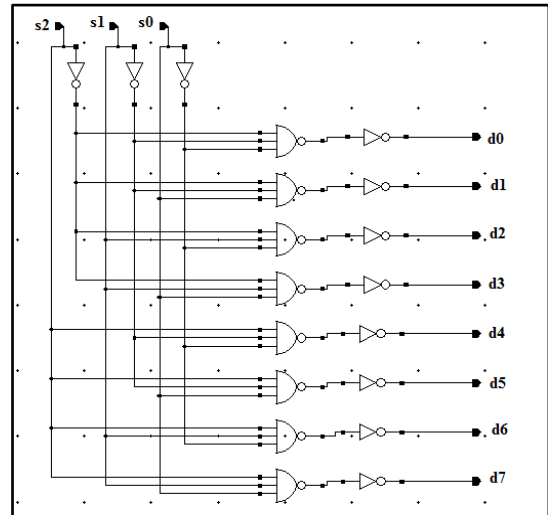


Figure 5: Supply decoder circuit in Cadence Virtuoso

Implementation of 2:1 Mux using basic gates is shown in Figure 6. Using the 2:1Mux, 4:1 Mux is designed and using both 2:1 and 4:1 Mux an 8:1 Mux is designed and finally using eight such blocks of 8:1 Mux, the Output Mux Block of proposed SAL-ALU architecture is designed.

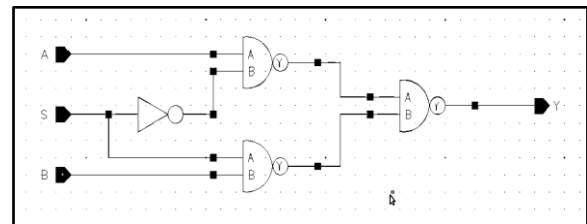


Figure 6: 2:1 MUX using basic gates in Cadence Virtuoso

3.1 Simulation results of operations in ALU

Test bench for simulation of the SAL-ALU schematic shown in Figure 3 is designed in Cadence virtuoso schematic editor as shown in Figure 7. Output is observed for different inputs and for different operations selected by control signals

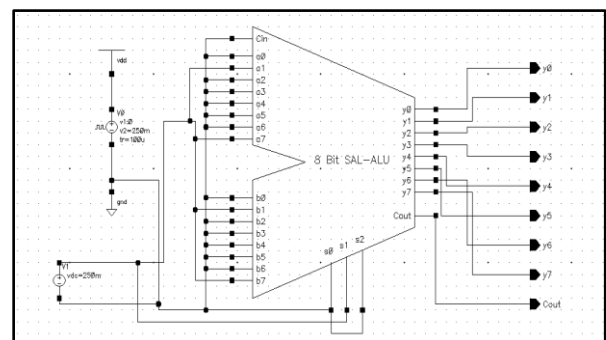


Figure 7: SAL-ALU Test Bench

For the ALU implemented pwr_clk is the power supply used as shown in Figure 8 with following voltage specifications:

- i. $V_{DDmax} = 250mV$
- ii. $f_{supply} = 4KHz$

When the output is driven high it follows the pwr_clk as shown in the simulated waveforms of the circuit below.

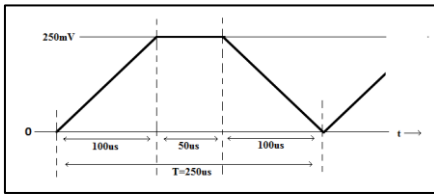


Figure 8: pwr_clk Signal

3.1.1 Addition

Control signals s2,s1,s0 are made 0,0,0 for selecting addition operation in the test bench. The simulation results obtained for the inputs given below are shown in Figure 9.

Inputs

A= 0000 0001

B= 0001 1001

Cin =1

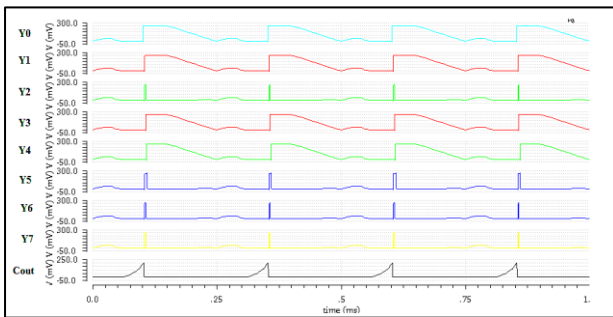


Figure 9: Simulation output for Addition

3.1.2 Subtraction

Control signals s2,s1,s0 are made 0,0,1 for selecting subtraction operation in the test bench. The simulation results obtained for the inputs given below are shown in Figure 10.

Inputs

A= 0000 1000

B= 0000 0001

Cin =0

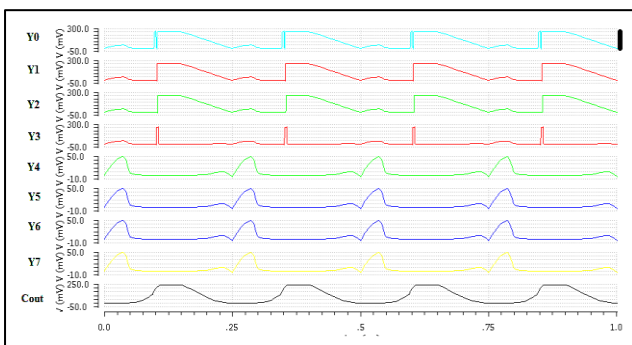


Figure 10: Simulation Output for Subtraction

3.1.3 NOR Operation

Control signals s2,s1,s0 are made 1,0,1 for selecting logical operation NOR in the test bench. The simulation results obtained for the inputs given below are shown in Figure 11.

Inputs

A= 0001 1000

B= 0000 0110

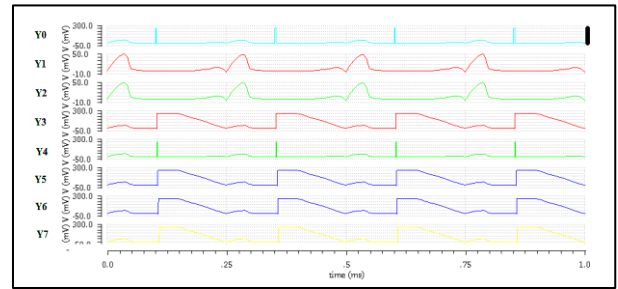


Figure 11: Simulation output for NOR operation

3.1.4 NOT Operation

Control signals s2,s1,s0 are made 1,1,1 for selecting logical operation NOT in the test bench. The simulation results obtained for the inputs given below are shown in Figure 12.

Inputs

A= 0000 0001

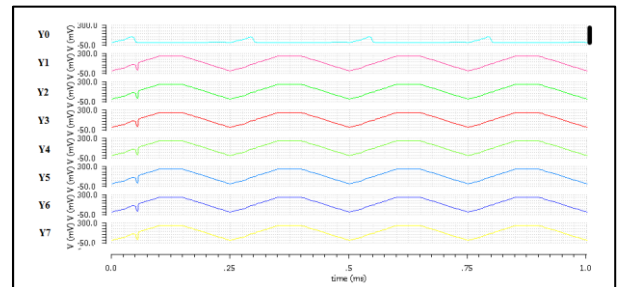


Figure 12: Simulation output for NOT operation

3.2 Physical Implementation of SAL-ALU

Physical Layout of the proposed 8-Bit SAL-ALU is built using Cadence Virtuoso Layout Suite. Initially the layout of Single Bit ALU is built and eight similar blocks are used to build the final layout of 8-Bit SAL-ALU shown in Figure 14. From the final Layout of the SAL-ALU built the total chip area is obtained from placement report shown in Figure 13.

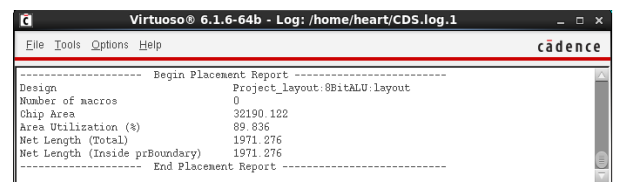


Figure 13: Placement report of 8-Bit SAL-ALU

From the placement report it is observed that the total chip area required to implement 8-Bit SAL-ALU is 32,190 μm^2 with an area utilization of 89%.

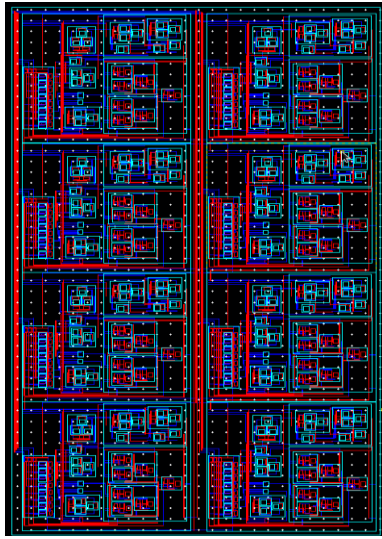


Figure 34: Physical Layout of 8-Bit SAL-ALU

4. RESULTS

Architecture of 8-Bit SAL-ALU is designed and implemented in Cadence Virtuoso Tools and is simulated to check for functionality. Average power consumed by the ALU and the delay in propagation of signal from input to output is computed from the simulation results and are tabulated below in Table 2. Power consumed is tabulated in dB and is plotted in Figure 15 for better comparison.

Table 2: $P_{average}$ and Propagation Delay Tabulation

Circuits	Propagation Delay(s)	$P_{average}$ (W)	Relative $P_{average}$ (dB) (Reference Power=1W)
8-Bit SAL-ALU	278×10^{-7}	361.7×10^{-12}	-99.5
8-Bit CMOS ALU	728×10^{-12}	960×10^{-9}	-58.17

From the placement report of 8-Bit SAL-ALU it is inferred that the total chip area required to physically implement the designed ALU is $32,190 \text{ um}^2$ with area utilization of 89%. The remaining 11% area is used for manually routing the design to create all the connections between the blocks in the design.

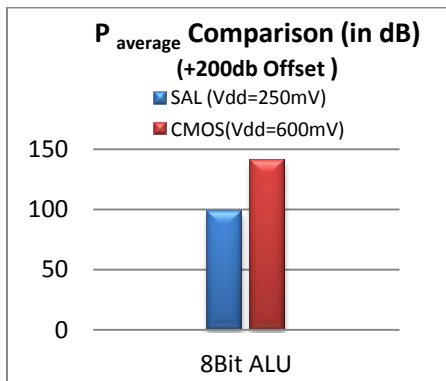


Figure 15: $P_{average}$ Comparison plots of 8-Bit ALU

5. CONCLUSION

From the results it is inferred that, SAL provides a very good reduction in power consumption at the cost of performance. Low frequency applications can be designed using SAL. There is almost 40dB reduction in power consumption in case of SAL-ALU compared to the CMOS counterpart as observed from the plot in Figure 15. For the portable applications which are battery operated and where performance is not the primary concern SAL is a good alternative.

6. FUTURE SCOPE

There is reduction in Noise margin as we design complex SAL circuits. Research can be undertaken in finding out different measures to increase noise margin of SAL-circuits. Using the ALU designed in this paper a complete processor or a small digital device can be built using this logic and practical performance can be verified.

7. REFERENCES

- [1] M. Chanda, S. Jain, S. De and C. K. Sarkar, "Implementation of Subthreshold Adiabatic Logic for Ultralow-Power Application," in IEEE Transactions on VLSI Systems, vol. 23, no. 12, pp. 2782-2790, Dec. 2015. doi: 10.1109/TVLSI.2014.2385817
- [2] Dr. H V Ravish Aradhya, et al, "Sub-threshold Adiabatic Logic(SAL) based building blocks for combinational system design" in RTEICT-2017, Sri Venkateshwara College of Engineering, Bengaluru – 562 157, 19-20 May 2017.
- [3] M. Chanda, R. Dutta, A. Rahaman and C. K. Sarkar, "Analysis of NAND/NOR gates using subthreshold adiabatic logic (SAL) for ultra low power applications," 2016 International Conference on Microelectronics, Durgapur, 2016, pp. 1-5
- [4] Dr. H V Ravish Aradhya, et al, "Design, and Performance comparison of Adiabatic 8-Bit Multipliers," in DISCOVER-2016, NITK, Surathkal, Karnataka, 13-14 August 2016
- [5] C. S. A. Gong et al., "Comparison of subthreshold logic with adiabatic circuit techniques," 2014 International SoC Design Conference (ISOC), Jeju, 2014, pp. 146-147. doi: 10.1109/ISOC.2014.7087669
- [6] Dr. H V Ravish Aradhya, et al, "Design, Analysis and Performance comparison of GNRfet based Adiabatic 8-Bit ALU," in RTEICT-2016, Sri Venkateshwara College of Engineering, Bengaluru – 562 157, 20-21 May 2016.
- [7] Y. Takahashi, T. Sekine, N. A. Nayan and M. Yokoyama, "Power-saving analysis of adiabatic logic in subthreshold region," 2012 International Symposium on Intelligent Signal Processing and Communications Systems, New Taipei, 2012, pp. 590-594. doi: 10.1109/ISPACS.2012.6473558
- [8] A. Vallandas, Arunkumar P. Chavan and S. Jagannathan, "Design of low power and energy efficient 5×5 multipliers," in ICRAIE-2014, Jaipur, 2014, pp. 1-5. doi: 10.1109/ICRAIE.2014.6909239
- [9] P. Bhati and N. Z. Rizvi, "Adiabatic logic: An alternative approach to low power application circuits," ICEEOT, Chennai, 2016, pp. 4255-4260. doi: 10.1109/ICEEOT.2016.775552