IMPLEMENTATION OF LOW POWER AREA EFFICIENT ALU WITH LOW POWER FULL ADDER USING MICROWIND DSCH3

Ritafaria D[1], Thallapalli Saibaba[2]
Assistant Professor, CJITS, Janagoan, T.S, India

Abstract

In this paper implemented a low Power Area efficient ALU using XNOR logic. The 4bit ALU design. ALU is an Arithmetic and Logic Unit, which performs arithmetic operation like ADD, SUB, PASS THROUGH, TWO'S COMPLEMENT, etc. and logic operation like AND, OR, EXCLUSIVE OR, EXCLUSIVE NOR, etc. Full adder is the basic component for an ALU. By reducing the power of full adder, the ALU power also be reduced. Compared with Gate Diffusion Input Full Adder, 50% power reduced in the XNOR based Full Adder Technique. The simulation is carried out using Microwind DSCH3.

INTRODUCTION

In the era of growing technology and scaling of devices up to nanometer regime, the arithmetic logic circuits are to be designed with compact size, less power and propagation delay. Arithmetic operations are indispensable and basic functions for any high speed low power application digital signal processing, microprocessors, image processing etc. Addition is most important part of the arithmetic unit rather approximately all other arithmetic operation includes addition. Thus, the primary issue in the design of any arithmetic logic unit is to have low power high performance adder cell.

Power has become one of the major important issues in the VLSI circuits. As the complexity increases and number of transistors in a chip increase, one has to consider the power dissipation. Higher power dissipation, increases the temperature and hence increases the chip temperature. As power dissipation goes on increasing, the battery life, reliability, cooling system and cost are affected. Hence as a designer it is important to design circuits to consume less power by avoiding unwanted switching actions. The majority of the power dissipation is due to dynamic power dissipation. That is due to switching action of the node and internal node capacitance of the circuit. Other sources of power dissipation are static current and short circuit current. Static power dissipation is basically due to the reverse current and sub threshold current when the transistor is off. Compared to dynamic power dissipation, static power is less around (20% - 50%) but increases as complexity increases. Short circuit current is due to non-zero rise time and fall time and is around (5% - 10%). Many techniques are used to reduce the leakage current and hence to reduce the static power dissipation such as power gating, multiple Vt, etc... There are many techniques involved to reduce the dynamic power dissipation by reducing the switching action such as Block enabling, Pre computation technique, etc... Clock gating technique is used to reduce the clock power. The dynamic power dissipation is around (40%-70%) and is relatively reducing with today's technology.

An Arithmetic and Logic Unit (ALU) is a digital circuit that performs arithmetic and logic operations. The ALU is a fundamental building block of the central processing unit of a computer. The power consumed by the ALU has a direct impact in the power dissipated from the processor. Hence, a design is required to implement the ALU in a fashion where the performance of the processor is improved and also the power consumed is less. To be precise Power consumption of whole data path can be reduced by reducing power consumption of ALU. Adder is the basic building block for an ALU. To reduce the power consumption from ALU first we need to reduce through full adder.

ALU (Arithmetic Logic Unit)

The heart of every computer is an Arithmetic Logic Unit (ALU). This is the part of

the computer which performs arithmetic operations on numbers, e.g. addition, subtraction, etc. In this lab you will use the Verilog language to implement an ALU having 10 functions. Use of the case structure will make this job easy.

The ALU Design and Operation A 4 - bit ALU has been designed for 3.0 V operation in which, the full adder design has been implemented using MIFG CMOS inverters. The ALU has four stages, each stage consisting of three parts: a) input multiplexers b) full adder and c) output multiplexers. The ALU performs the following four arithmetic operations, ADD, and SUBTRACT INCREMENT and DECREMENT. The four logical operations performed are EXOR, EXNOR, AND and OR. The input and output sections consist of 4 to 1 and 2 to 1 multiplexers. The multiplexers were designed using the pass transistor logic. A set of three select signals has been incorporated in the design to determine the operation being performed the inputs and outputs being selected.

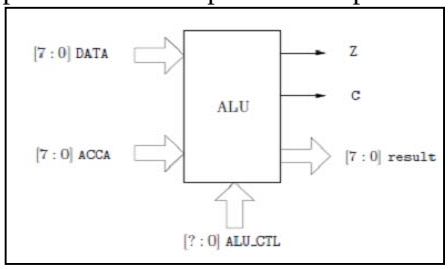


Fig 1: ALU Block Diagram

DESIGN OF ALU USING LOW POWER FULL ADDER

An arithmetic and logic unit is a fundamental block for many processors. It performs many processors. It performs many operations like addition, subtraction, XOR, XNOR, buffer, NAND, OR etc.The 4 bit ALU operation can be implemented using eight 4x1 multiplexer, four full adder, four 2x1 multiplexer. Depends upon the three selection line s2, s1, s0, the arithmetic and logic operation can be performed.

The block of 4x1 multiplexer consists of four inputs, which is logic 0, logic 1, B and B'. Depends upon the s0 and s1 selection line; the desired output can be generated. These outputs.

The next stage of 4x1 multiplexer has the input of full adder sum, which is EXOR, EXNOR, AND, and OR. Depends on the s0 and s1 selection lines, the output can be generated. It acts as an input for the 2x1 multiplexer. Another input of 2x1 multiplexer is the full adder sum output. Finally, the output stage of 2x1 multiplexer can be generated by using s2 selection line.

Table: Operations of ALU

S2	S1	S0	OPERATIONS
0	0	0	Buffer
0	0	1	EXOR
0	1	0	EXNOR
0	1	1	OR
1	0	0	ADDITION
1	0	1	SUBTRACTION
1	1	0	BITWISE NAND
1	1	1	Inverter

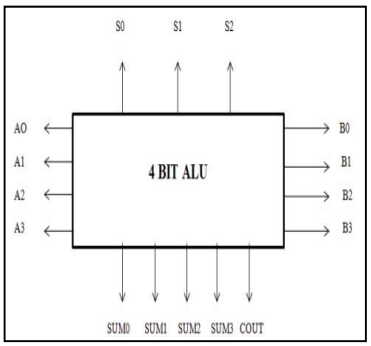


Fig2: Logo representation of ALU

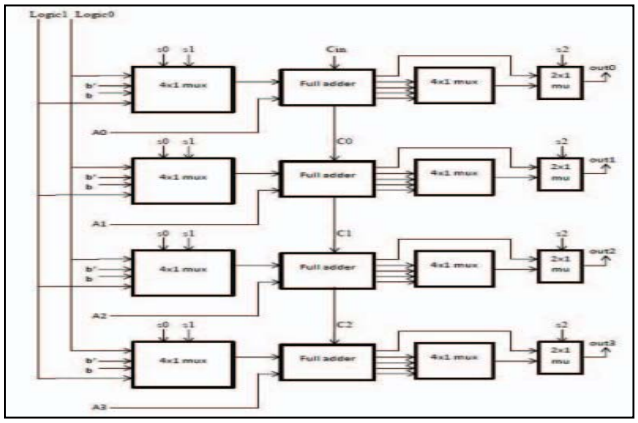


Fig 3: Process of 4bit arithmetic and logic unit

Multiplexer logic at the input stage consists of 4x1 multiplexer and 2x1 multiplexer. Depends upon the selection line of s0, s1 and s2, the output of full adder has been computed. Multiplexer logic at the output stage consists of 4x1 multiplexer and 2x1 multiplexer. Depends upon the selection line of s0, s1, and s2, the output of ALU has been computed. Input stage multiplexer consists of VDD, VSS, B and B'. The output stage multiplexer consists of EXOR,

EXNOR, AND and OR input. Fig.8 shows input stage of multiplexer architecture and fig9 shows output stage of multiplexer architecture.

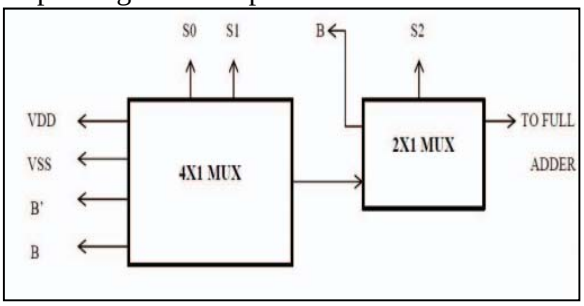


Fig 4: Architecture of multiplexer logic at the input stage

This paper presents a new technique of XNOR logic, to design the full adder is resolved by gate diffusion input technique which proven to have high power consumption and compared with XNOR logic. These new approach of XNOR logic gives excellent result then previous design in charge of power consumption, area as well as propagation delay.

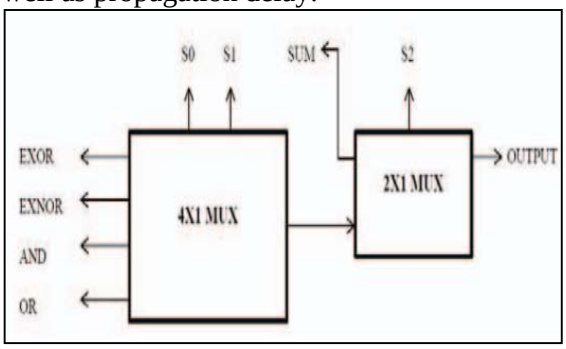
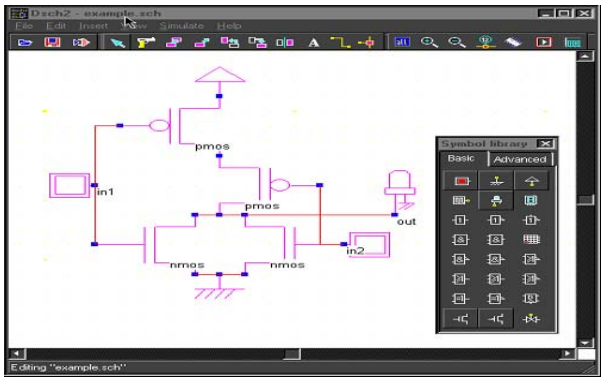


Fig5: Architecture of multiplexer logic at the output stage

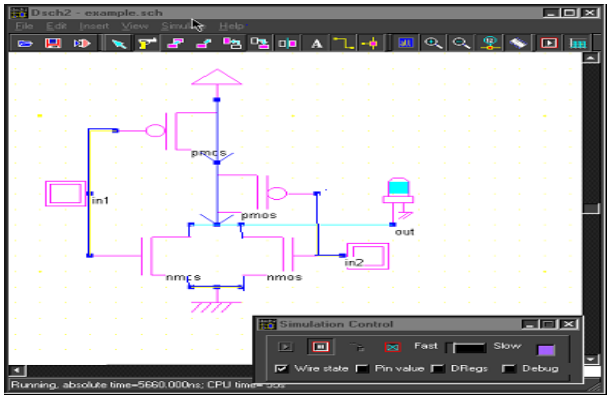
Micro wind / DSCH NOR Example: NOR Gate Logic

- Open the Schematic Editor in Micro wind (DSCH3). Click on the transistor symbol in the symbol Library on the right.
- Instantiate NMOS or PMOS transistors from the symbol library and place them in the editor window.
- Instantiate 2 NMOS and 2 PMOS transistors.
- Connect the drains and sources of transistors.
- Connect Vdd and GND to the schematic.
- Connect input button and output LED.

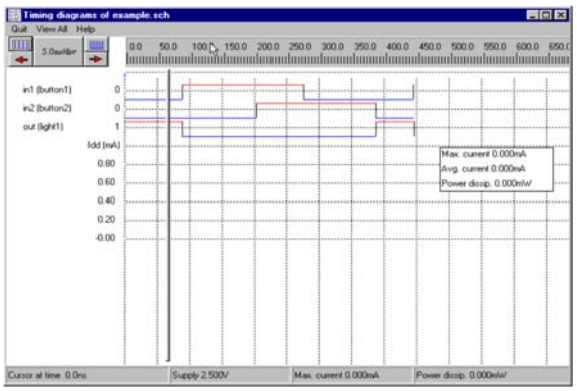


Now we have NOR schematic ready.

- Use your logic simulator to verify the functionality of your schematic.
- The next step is to simulate the circuit and check for functionality.
- Click on, Simulate -> Start simulation.
- This brings up a Simulation Control Window.
- Click on the input buttons to set them to 1 or 0. Red color in a switch indicates a '1'. As shown,

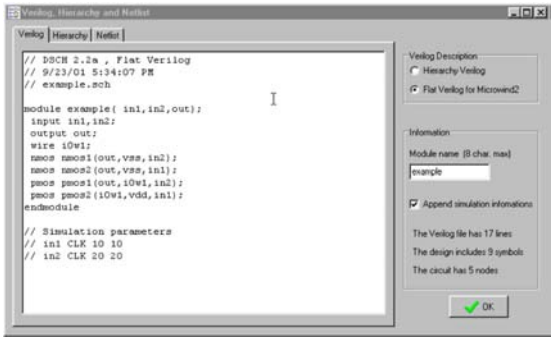


The simulation output can be observed as a waveform after the application of the inputs as above. Click on the timing diagram icon in the icon menu to see the timing diagram of the input and output waveforms.



Simulate your system with your hand calculated transistor sizes.

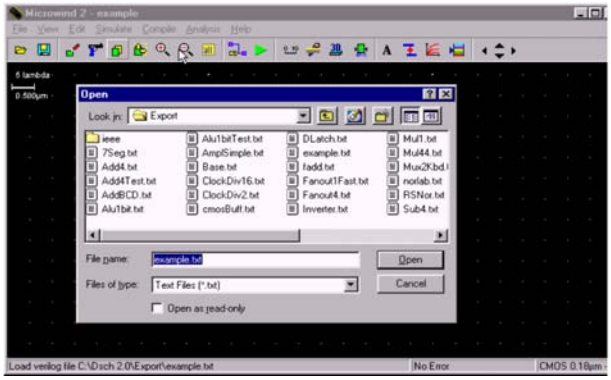
- Click File -> Make Verilog File. The Verilog, Hierarchy and Netlist window appears. This window neither shows the Verilog representation of NOR gate.
- Click OK to save the Verilog as a .txt file.



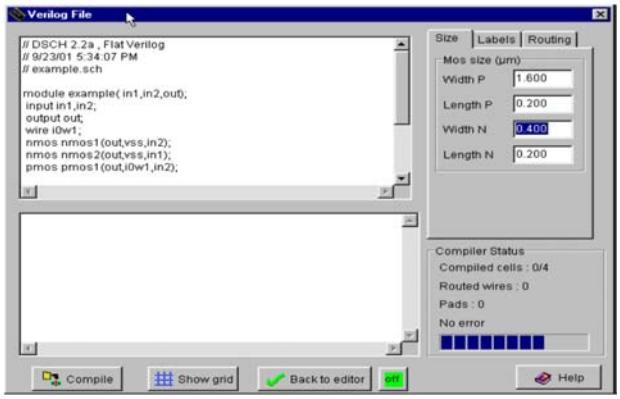
3.9.2 Microwind / DSCH NOR Example: Circuit Design

- Open the layout editor window in Microwind. Click *File -> Select Foundry* and select *X.rul*.This sets your layout designs in X technology.

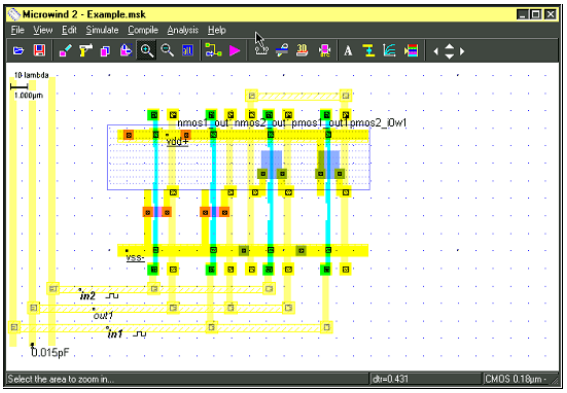
- Click on *Compile -> Compile Verilog File.* An *Open* Window appears Select the .txt verilog file saved before and opens it.



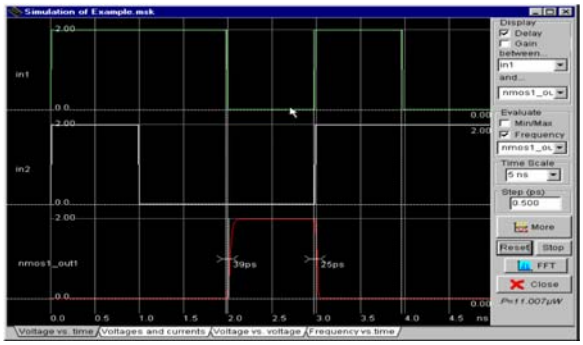
After selecting the .txt file, a new window appears called Verilog file. Click on Size on the right top menus. This shows up the NMOS and PMOS sizes. Set the sizes according to choice.



- Click Compile and then Back to editor in the Verilog File Window. This creates a layout in layout editor window using automatic layout generation procedure.
- Add a capacitance to the output of the design. The value of the capacitance depends on your choice.
- Click on OK. The capacitance is shown on the left bottom corner with a value of 0.015fF.



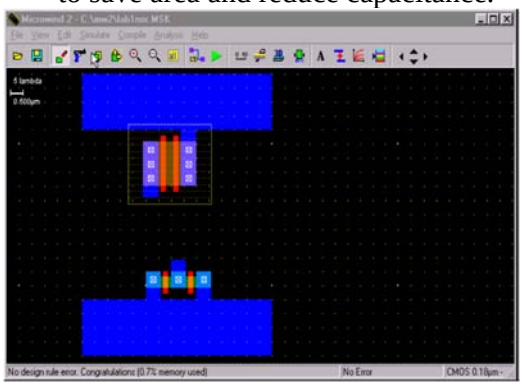
- Click Simulate -> Run simulation. A simulation window appears with inputs and output, shows the tphl, tplh and tp of the circuit. The power consumption is also shown on the right bottom portion of the window.
- If you are unable to meet the specifications of the circuit change the transistor sizes. Generate the layout again and run the simulations till you achieve your target delays.



3.9.3 Microwind / DSCH NOR Example:

Layout Design

- Design the layout manually. Open the layout editor window in Microwind. Click File -> Select Foundry and select X.rul, Vdd and GND rails are of Metal1. The top rail is used as Vdd and the bottom one as GND. Click on Metal 1 in the palette and then creates the required rectangle in the layout window.
- The next step is to build the NMOS transistors. Click on the transistor symbol in the palette. Set the W, L of the transistor
- Then click on Generate device. The source of the transistor is connected to the GND rail. Create another NMOS and place it in parallel to the first NMOS device. We share the two devices' drain diffusions. A DRC check can be run by clicking on Analysis -> Design Rule Checker.
- The next step is to place two PMOS transistors in series. Place the PMOs transistor on layout close to the Vdd rail on the top. To construct two PMOS transistors in series, diffusions are shifted to a side and another poly line is added as second transistor. The diffusion is shared to save area and reduce capacitance.



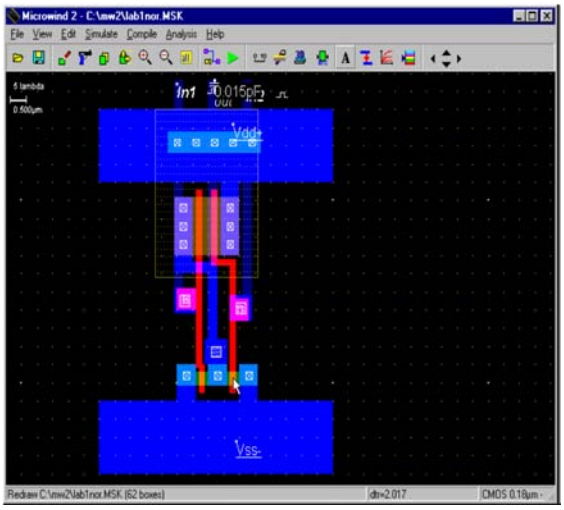
- The next step is to connect the inputs and the output of the two transistors. Poly inputs is connected. Metal output is connected.
- The next step is to connect the poly to metal1 and then to metal2. The first symbol in the first row of the palette is the poly to metal1 contact.
- Then we connect the metal1 to metal2 contact to the previous contact. This is the 4th contact on the first row.
- The next step is to connect the output Metal1 to Metal2. Once again use the 4th contact in the first row.
- Now we connect metal2 to the two inputs and one output and bring them to the top to go out of the cell. Observe the two inputs (left & right) and an output (middle) above the Vdd rail in dark blue color.

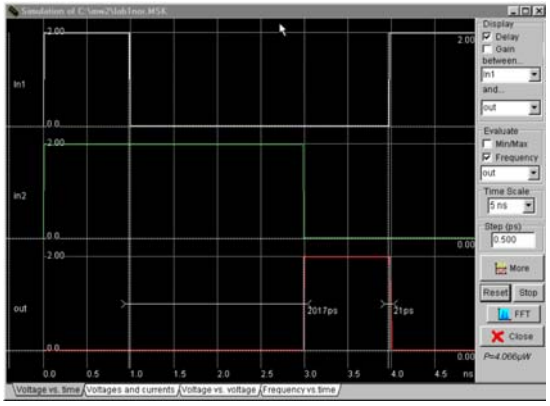
Now we label the inputs and output as In1, In2 and out. Click on Add a Pulse Symbol in the palette (5th from the right in the 3rd row). Then click on the metal2 of one of the inputs. A window appears. Change the name of the input signal. Insert a 01 sequences and click on Insert. The click on Assign. Similarly assign the 2nd input a pulse.

Select the Visible Node symbol from the palette (7th in the third row). Select it and click on the output. The 'Add a Visible Property' window appears. Change the label name to out. Select Visible in Simulation. Click on Assign. Now the output is also labeled.

Select Vdd Supply and GND from the palette (third row). Also click on the capacitor (3rd in 2nd row) symbol and add it to the output. Also, extend the pwell into the Vdd Rail. The click on Edit -> Generate -> Contacts. Select PATH and then in Metal choose Metal1 and N+ polarization.

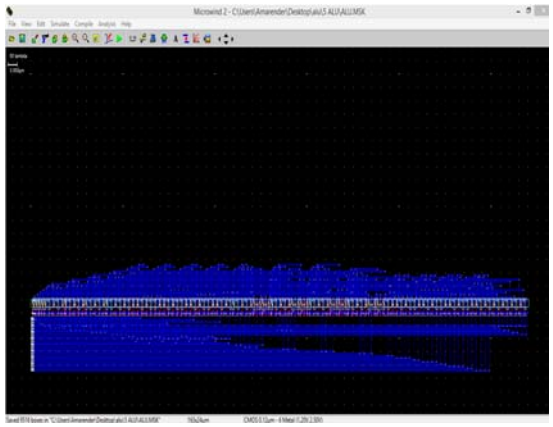
To run the Simulation of your circuit, click on Simulate -> Start Simulation. Depending on the input sequences assigned at the input the output is observed in the simulation. The power value is also given.



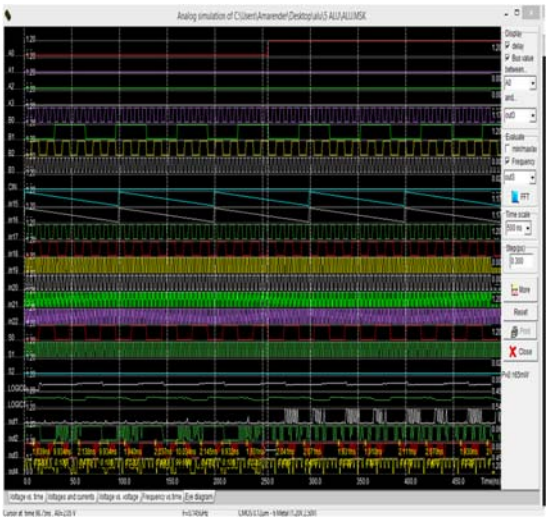


LAYOUT DESIGN

LAYOUT DESIGN FOR ALU



SIMULATION RESULTS



REFERENCE

[1]. R. Shalem, E. John, and L.K.John, "A novel lowpower energyrecovery full adder cell," in Proc. Great Lakes Symp.VLSI,Feb. 1999, pp.380–383

[2]. A.Sharma, R Singh and R. Mehra, " Low Power TG Full Adder Design Using CMOS Nano Technology,"

[3]. L.Bisdounis, D.Gouvetas and O.Koufopavlou, "A comparativestudy of CMOS circuit design styles for lowpower high-speedVLSI circuits" Int. J. of Electronics, Vol.84, No.6, pp 599-613,1998. Anu Gupta, Design Explorations of VLSI ArithmeticCircuits, Ph.D. Thesis, BITS,Pilani, India, 2003.

[4]. T. Esther Rani, M. Asha Rani, Dr.Rameshwarrao, "AREAOPTIMIZED LOW POWER ARITHMETIC AND LOGICUNIT"

[5]. R.Zimmermannn and W.Fichtner, "Low-power logic styles:CMOS versus pass-transistor logic," IEEE J. Solid-StateCircuits, vol. 32, pp. 1079–1090, July 1997. Y.Jiang, Y.Wang,and J.Wu, "Comprehensive Power Evaluation of Full Adders,"Florida Atlantic Univ., Boca Raton, Tech. Rep., 2000.

[6]. R.Uma and P. Dhavachelvan," Modified Gate Diffusion InputTechnique: A New Technique for Enhancing Performance inFull Adder Circuits" 2nd International Conference on Communication, Computing & Security [ICCCS-2012].