Efficient Design of Low-Power 4-bit ALU using HVT Cell Concept

R.Durga Bhavani[1] V.Silpa Kesav[2]
[1] CVR College of Engineering, ECE Dept.
Email: bhavaniregalla@gmail.com
[2] CVR College of Engineering, ECE Dept.
Email: shilpakesav@gmail.com

***Abstract*—In this paper a 4-bit two input ALU is designed using hvt cells in 45nm technology. ALU is most important element in CPU. It consists of AE, LE, FA, and CE. FA is designed using Transmission gate based multiplexer concept. This ALU is designed to manipulate both arithmetic and logical operations and internally arithmetic block is designed to calculate both signed and unsigned numbers. The LE and FA are implemented using hvt cells. The ALU is designed and implemented using CADENCE Tools with GPDK 45nm Technology. The power is compared between normal threshold voltage level transistors and high threshold voltage level transistors. The simulation results show that design of LE and FA through high threshold voltage level cells is more power efficient than with low threshold voltage level cells.**

***Index Terms*—Arithmetic Extender, Logic Extender, Carry Extender, Fulladder**

I. Introduction

ALU is a digital circuit and a basic building block in cpu, microprocessors and other micro controller devices. It is used for performing both arithmetic and logical operations. Modern CPU's consist of very powerful and complex ALU's.

ALU also contributes to one of the highest power density locations on the processor, as it is clocked at the highest speed and is busy mostly all the time which results in thermal hotspots and sharp temperature gradients within the execution core. Therefore, this motivates us strongly for energy efficient ALU designs that satisfy the high performance requirements and average power dissipation [5].

Here in this paper we have tried to attempt the design cells using hvt concept. The hvt concept is also called as higher threshold voltage level transistor concept. Using these hvt cells, power can be reduced very much comparing with normal threshold voltage level transistors. It can be better explained by considering sub-threshold current of a transistor. When transistor is in sub-threshold region, the sub-threshold current of a transistor increases as technology goes on improving. In sub-threshold region the leakage current is related to threshold voltage (V_t) and is shown in the below equation.

$$Ioff(nA) = 100\frac{W}{L}e^{-qVt/\eta Kt}$$

From this current equation it can be observed that for less threshold voltage (V_t) the sub-threshold leakage current increases exponentially which in turn increases leakage power. Otherwise if threshold voltage (V_t) is large, then sub-threshold leakage current decreases which in turn decreases leakage power. From this it can be concluded that larger the threshold voltage (V_t) lesser the leakage power.

Here we have designed a four bit ALU with three select lines for performing eight operations. In these eight operations, four operations are performed for logical and four operations are performed for arithmetic operations. The design includes four basic blocks. They are Logic Extender (LE), Arithmetic Extender (AE), Carry Extender (CE) and Fulladder (FA). The function of LE is to manipulate logic operations, AE is to manipulate arithmetic operations partially, CE is for carry operations and FA is for actual arithmetic operations.

II. Existing ALU

ALU is designed using the concept of Adder/subtractor to add both signed and unsigned numbers. Adder/Subtractor is used as the building block along with Logic block and Arith blocks in front of the two input operands of each fulladder. By using this technique the primary input will be modified accordingly depending on the operations being performed before being passed to the fulladder [1]. It consists of four LE blocks, four AE blocks, four FA blocks and one CE block. Block diagram of existing ALU Block is shown in Figure 1. ***Abstract*—In this paper a 4-bit two input ALU is designed using hvt cells in 45nm technology. ALU is most important element in CPU. It consists of AE, LE, FA, and CE. FA is designed using Transmission gate based multiplexer concept. This ALU is designed to manipulate both arithmetic and logical operations and internally arithmetic block is designed to calculate both signed and unsigned numbers. The LE and FA are implemented using hvt cells. The ALU is designed and implemented using CADENCE Tools with GPDK 45nm Technology. The power is compared between normal threshold voltage level transistors and high threshold voltage level transistors. The simulation results show that design of LE and FA through high threshold voltage level cells is more power efficient than with low threshold voltage level cells.**

***Index Terms*—Arithmetic Extender, Logic Extender, Carry Extender, Fulladder**

DOI: 10.32377/cvrjst0912

I. Introduction

ALU is a digital circuit and a basic building block in cpu, microprocessors and other micro controller devices. It is used for performing both arithmetic and logical operations. Modern CPU's consist of very powerful and complex ALU's.

ALU also contributes to one of the highest power density locations on the processor, as it is clocked at the highest speed and is busy mostly all the time which results in thermal hotspots and sharp temperature gradients within the execution core. Therefore, this motivates us strongly for energy efficient ALU designs that satisfy the high performance requirements and average power dissipation [5].

Here in this paper we have tried to attempt the design cells using hvt concept. The hvt concept is also called as higher threshold voltage level transistor concept. Using these hvt cells, power can be reduced very much comparing with normal threshold voltage level transistors. It can be better explained by considering sub-threshold current of a transistor. When transistor is in sub-threshold region, the sub-threshold current of a transistor increases as technology goes on improving. In sub-threshold region the leakage current is related to threshold voltage (V_t) and is shown in the below equation.

$$Ioff(nA) = 100\frac{W}{L}e^{-qVt/\eta kT}$$

From this current equation it can be observed that for less threshold voltage (V_t) the sub-threshold leakage current increases exponentially which in turn increases leakage power. Otherwise if threshold voltage (V_t) is large, then sub-threshold leakage current decreases which in turn decreases leakage power. From this it can be concluded that larger the threshold voltage (V_t) lesser the leakage power.

Here we have designed a four bit ALU with three select lines for performing eight operations. In these eight operations, four operations are performed for logical and four operations are performed for arithmetic operations. The design includes four basic blocks. They are Logic Extender (LE), Arithmetic Extender (AE), Carry Extender (CE) and Fulladder (FA). The function of LE is to manipulate logic operations, AE is to manipulate arithmetic operations partially, CE is for carry operations and FA is for actual arithmetic operations.

II. Existing ALU

ALU is designed using the concept of Adder/subtractor to add both signed and unsigned numbers. Adder/Subtractor is used as the building block along with Logic block and Arith blocks in front of the two input operands of each fulladder. By using this technique the primary input will be modified accordingly depending on the operations being performed before being passed to the fulladder [1]. It consists of four LE blocks, four AE blocks, four FA blocks and one CE block. Block diagram of existing ALU Block is shown in Figure 1.

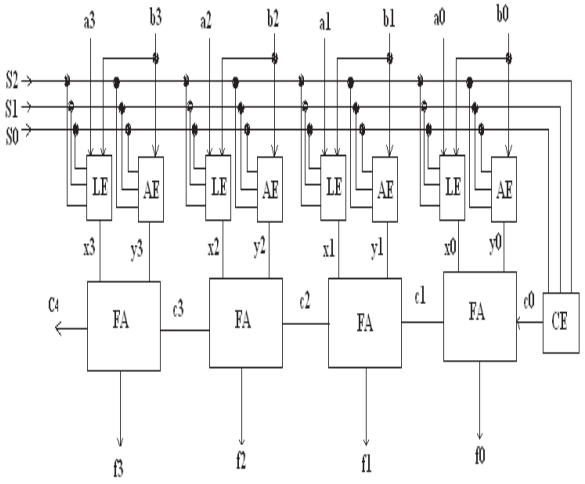

Figure 1. Existing ALU System

Here all the blocks of ALU are designed using pull-up and pull-down logic. The operations of ALU which we implemented in this ALU are shown in TableI.

TABLE I

ALU FUNCTION TABLE

S2	S1	S0	Operation.Name	Operation	X(LE)	Y(AE)	C0(CE)
0	0	0	Pass	Pass A to output	A	0	0
0	0	1	AND	A AND B	A AND B	0	0
0	1	0	OR	A OR B	A OR B	0	0
0	1	1	NOT	A`	A`	0	0
1	0	0	Addition	A+B	A	B	0
1	0	1	Subtraction	A-B	A	B`	1
1	1	0	Increment	A+1	A	0	1
1	1	1	Decrement	A-1	A	1	0

From this function table, we can observe that the S2 line selects between the arithmetic operations and the logical operations. When S2='0' logical operations are performed and when S2='1' arithmetic operations are performed. The two select lines S1 and S0 are used to select one among four possible arithmetic operations and four logical operations. Thus this ALU can implement eight different operations.

A. Design of LE

The combinational circuit named LE (Logic Extender) performs the logical operations. The LE performs the actual logic operations on the two primary operands 'ai' and 'bi' before passing the result to the first operand 'xi' which is one of the inputs to the full adder. The 'X' column from the function table shows the values to which it must generate different logic operations. LE truth table is derived from the function Table I and it is shown below.

DOI: 10.32377/cvrjst0912

TABLE II
TRUTH TABLE FOR LE

S2	S1	S0	Xi
0	0	0	ai (Pass A)
0	0	1	ai bi (A AND B)
0	1	0	ai + bi (A OR B)
0	1	1	ai' (NOT of A)
1	X	X	Ai

From the truth table we can observe that logical operations are performed only when S2='0' as discussed previously. The operand 'ai' value is just passed through without any modifications when S1=S0='0'. AND operation is performed between the 'ai' and 'bi' operands when S1='0' and S2='1'. OR operation is performed between the 'ai' and 'bi' operands when S1='1' and S0='0'. NOT operation is performed when S1=S0='1'. During these logical operations, the outputs of AE and CE blocks are set to zero because we do not want the fulladder to change the results. Here the output variable xi is dependent on S2, S1, S0, ai and bi variables. From the truth table the circuit is derived by using K-map equation. The equation for LE is shown below.

$$X_i=S_2a_i+S_0'a_i+S_1a_ib_i+S_2'S_1a_i(S_0+b_i).$$

Schematic of LE which is designed using gates is shown in Figure 2. It consists of four Inverters, one 2-input OR gate, two 2-input AND gates, one 4-input AND gate and one 5-input OR gate.

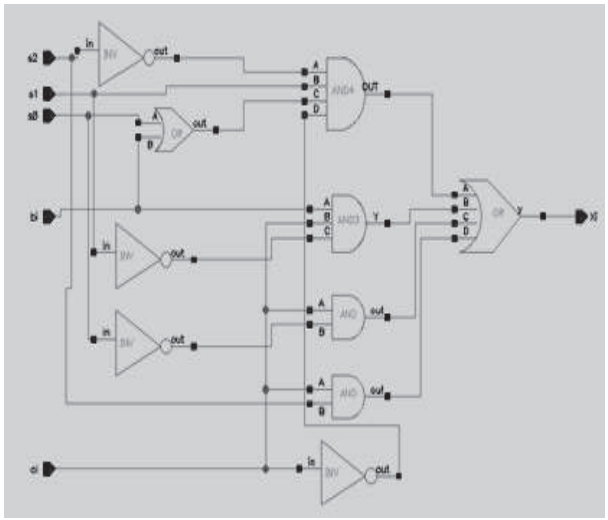

Figure 2. Schematic of LE

B. Design of AE

The combinational circuit named AE (Arithmetic Extender) performs the arithmetic operations. AE modifies the second operand bi and passes it to the second operand 'yi' which is one of the inputs to the full adder where the actual arithmetic operation is performed. The 'yi' column from the function table shows the values to which it must generate different operations. AE truth

table is derived from the function Table I and it is shown below.

TABLE III
TRUTH TABLE FOR AE

S2	S1	S0	Bi	Yi
0	X	X	X	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

From the truth table we can observe that logical operations are performed only when S2='1'as discussed previously. Here along with two select lines (S1, S0) we have taken one more operand 'bi' as the input because we need to modify 'yi', which is the second operand to the full adder so that all operations can be done with additions. Thus, the AE only takes the second operand of the primary input 'bi' as its input and modifies the value
depending on the operation being performed.

When select lines S1=S0='0', then 'bi' value will be passed out itself to perform addition. When S1='0' and S0='1' the complement of 'bi' value is passed out to perform subtraction. When S1='1' and S0='0', then output will be set to '0' to perform increment. When S1=S0='1' then output will be set to '1' to perform decrement. Here the output variable 'yi' is dependent on S2, S1, S0 and bi variables. From the truth table the circuit is derived by using K-map equation. The equation for AE is shown below.

$$Y_i=S_2S_0\ (S_1+b_i')+S_2S_1'S_0'b_i$$

Schematic of AE which is designed using gates is shown in Figure 3. It consists of three inverters, two 2-input OR gates, one 3-input AND gate and one 4-input AND gate.

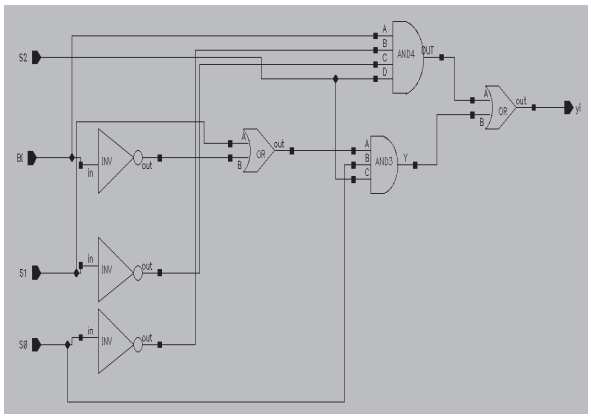

Figure 3. Schematic of AE

C. Design of CE

The combinational circuit named CE (carry Extender) acts as primary carry-in signal c0 to full adder. The C0 column from the function table shows the values that the

CE must generate for different operations. CE truth table is derived from the function Table 1 and it is shown below.

TABLE IV

TRUTH TABLE FOR CE

S2	S1	S0	C0
0	X	X	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

From the truth table we can observe that the carry output is getting activated only when S2='1' because during logical operations carry signal is not required and output of carry is set to zero. When S1=S0='0' then output of C0 is set to '0' because initially during addition carry-in signal is not required. When S1='0', S0='1' and S1='1', S0='0' then output of C0 is set to '1' because it is selected only during addition operations. When S1=S0='1' then output of C0 is set to '0' because during decrement carry-in signal is not required. Here the output variable 'C0' is dependent only on three select lines S2, S1, S0. From the truth table the circuit is derived by using K-map equation. The equation for CE is shown below.

$$C_0=S_2(S_1'S_0+S_0'S_1)$$

Schematic of CE which is designed using gates is shown in the Figure 4. It consists of one 2-input XOR gate and one 2-input AND gate.

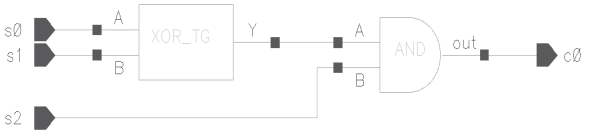

Figure 4. Schematic of CE

D. Design of FA

In general the operation of full adder is to perform addition. The inputs of FA are xi, yi and co which are respectively the outputs of LE, AE and CE blocks. Here it cannot change the logical operations because during that period the output of AE will be zero and it passes directly through it without any modifications.

From the function Table I we can observe that when S2='1', S1='0' and S0='0' it performs addition between the operands xi and yi. When S2='1', S1='0' and S0='1' it should perform subtraction but instead of subtracting B directly, we will Add '-B'. This can be possible only by changing B to two's complement format. It can be achieved by flipping the bits of 'B' and then adding one. Thus, 'yi' output of AE gets inverse of 'B' and one is added through the carry-in C0 and performs subtraction. When s2='1',S1='1' and S0='0' the increment operation is performed from taking 'ai' as one input and C0 as the other input while setting 'yi' to all zeros. When S2=S1=S0='1' the decrement operation is performed. The sum and carry equations for FA are shown below.

$$Sum=AB'Cin'+ A'BCin'+ A'B'Cin+ ABCin$$
$$Carry=AB+BCin+ACin$$

Schematic of FA which is designed using gates is shown in the Figure 5. It consists of two 2-input XOR gates, two 2-input AND gates and one 2-input OR gate.

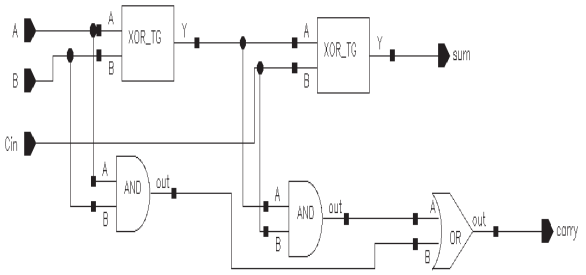

Figure 5. Schematic of FA using Gates

III. PROPOSED ALU

In proposed ALU, FA is designed using transmission gate based multiplexer concept and also a XOR gate is added to collect both unsigned and signed overflow bit value which is shown in Figure 6. Here the XOR gate is designed using transmission gates and in turn transmission gates are designed using hvt cells. The FA which is designed here functions faster based on the critical timing conditions than the full adder which is designed using gates and also power is reduced very much due to hvt cells as explained earlier. The highlighted blocks in the figure show that the cells are designed using hvt cell concept and remaining blocks are designed using normal threshold voltage level transistors.

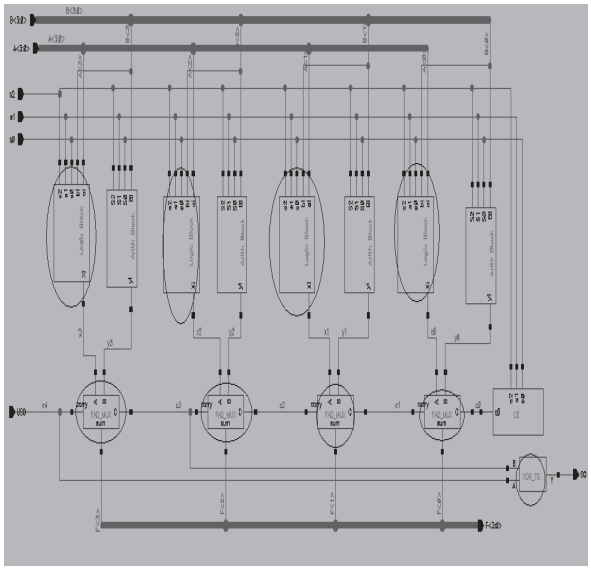

Figure 6. Proposed ALU System

A. Design of FA using Multiplexer

The FA designed here consists of two 4X1 Multiplexers and one inverter. The 4X1 Multiplexer is designed using transmission gates. The transmission gates and inverter are designed using hvt cells. The first 4x1

DOI: 10.32377/cvrjst0912

multiplexer is designed to get Sum function and second one is designed to get Carry function. Schematic of proposed FA is shown in Figure 7.

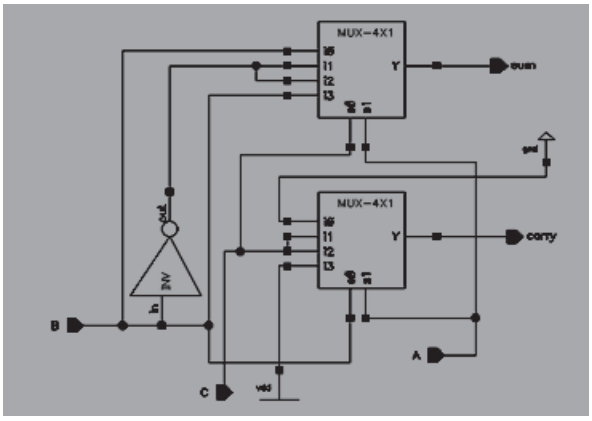

Figure 7. Schematic of FA using Multiplexers

B. Design of AE using hvt cells

The proposed AE which is designed using hvt cells is shown in the Fig 8. It consists of three inverters, two 2-input OR gates, one 3-input AND gate and one 4-input AND gate.

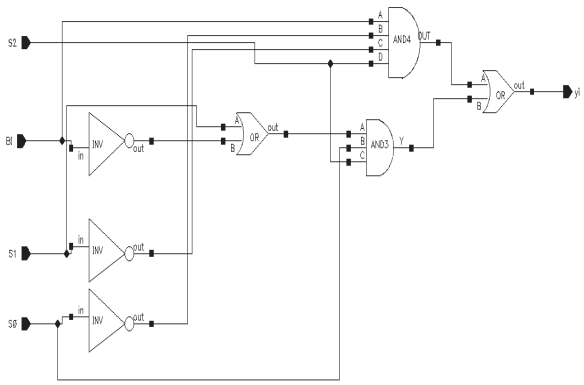

Figure 8. Schematic of AE using hvt cells

IV. Results

The entire ALU blocks are designed using Cadence Virtuoso Schematic XL Editor in GPDK 45nm technology. The designs are simulated by spectre SPICE simulator using Cadence Analog Design Environment L-Editor. The simulation result of ALU for the values A="1110" and B="0001" is shown in Figure 9. Power is calculated using Cadence Virtuoso Analyzer calculator.

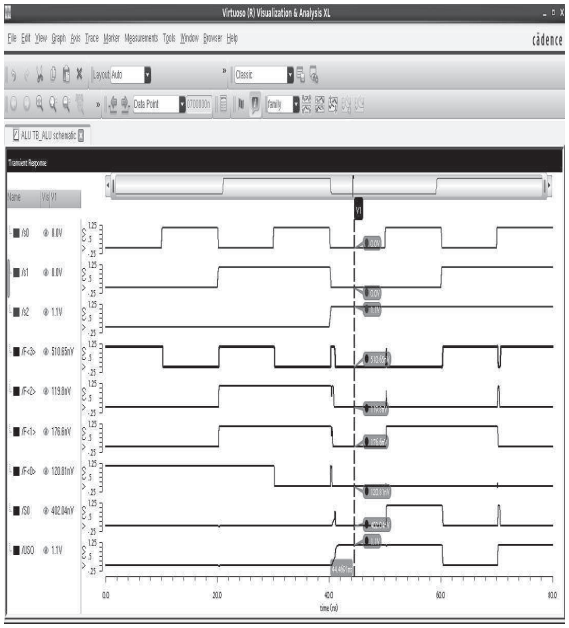

Figure 9. Simulation Result of ALU

Power comparison between the existing cells and proposed cells is shown in the Table V.

TABLE V

POWER ANALYSIS OF DIFFERENT BLOCKS

Cell	Normal V_t	High V_t
XOR with TG	51.3nw	46.93nw
LE	508.3nw	454.8nw
FA with Gates	176nw	150nw
FA with TG	126.4nw	105.5nw

The total ALU consumes 3.043uw of power by using normal threshold voltage transistors and after replacing by proposed blocks with high threshold voltage transistors, the consumption power has been reduced to 2.71uw. The designed ALU consists of 598 transistors. This includes 14 transistors for CE, 288 transistors for four LE blocks, 144 transistors for each four AE & FA blcoks and 8 transistors for XOR gate.

V. Conclusion

In this paper the proposed ALU is designed for some specified functions. When the inputs switch from pass 'ai' to pass 'bi' in LE then OR gate needs to be replaced by XNOR gate. Similarly by the replacement of different gates, the functionality of different operations can be performed in the total ALU. The design can also be extend for sixteen operations by changing the k-map equations. The design gives significant reduction in power consumption. Initially, the ALU circuit is designed using FA with gates and later it is replaced with multiplexer based FA through which power reduction is achieved. The entire blocks are designed and simulated using Cadence Tools.

REFERENCES

[1] Microprocessors Design Principles and Practices with VHDL by Encoh O.Hwang .

[2] Ziyan Qi, Xingguo Xiong, Linfeng Zhang and Md Tauseef Riasat "Design of Efficient Pass transistor Binary ALU" in ASEE 2014 Zone I Conference, April 3-5, 2014, University of Bridgeport, USA.

[3] P.Satyamoorthy, S.Vijya Lakshmi and A.Daneil Raj "Efficient Design of Low power ALU using PTL-GDI Logic Fulladder",in IJCA Dec 2013.

[4] K.Lakshmi Swetha and K.Kalpana "An Efficient Design and Implementation of ALU using Gated Diffusion Index", in IJECS Volume 4 Issue 5 May 2015.

[5] Akshey Dhenge, Abhilash Kapse and Sandip Kakde "VLSI Implementation of Area Optimized ALU using GDI Technique" Proceedings of International Conference on Research in Electrical, Electronic and Mechanical Engineering, Dehradun, 26th April-2014.

[6] Arun Prakash Singh, Rohit Kumar, "Implementation of 1-bit Full Adder using Gate Diffusion Input (GDI) cell ", International Journal of Electronics and Computer Science Engineering, ISSN-2277-1956/V1N2-333-342.

[7] Dan Wang, Maofeng Yang, Wu Cheng, Xuguang Guan, Zhangming Zhu, Yintang Yang, "Novel Low Power Full Adder Cells in 180nm CMOS Technology", ICIEA 2009, 978-1-4244-2800-7/09/$25.00 ©2009 IEEE

[8] Arun Prakash Singh, Rohit Kumar, "Implementation of 1-bit Full Adder using Gate Diffusion Input (GDI) cell ", International Journal of Electronics and Computer Science Engineering, ISSN-2277-1956/V1N2-333-342.

DOI: 10.32377/cvrjst0912