ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
Vol. 2, Issue 4, April 2013

Efficient Design of MAC Hybrid Adder in Quantum-Dot Cellular Automata

Bhuvaneswaran M[1], Shankar N.K.[2]

ME Student [VLSI Design], Dept.of ECE, Muthayammal Engineering College, Rasipuram, Tamilnadu, India[1]

Asst.Professor, Dept of ECE, Muthayammal Engineering College, Rasipuram, Tamilnadu, India[1]

ABSTRACT: Minimizing area and power is the more challenging task in modern VLSI design. Adders are the most widely used components in many circuits. The design of area and power efficient high-speed data path logic systems forms the largest areas of research in VLSI system design. Several types of adders are available in practice, each type is used for particular purpose based on their performance and features. In this project, design an efficient hybrid adder that combines the Ladner–Fischer adder with a ripple carry adder. And show that the hybrid adder has better performance (in terms of latency) in Quantum-dot cellular automata (QCA) than a Ladner–Fischer or a ripple carry adder. In proposed, design MAC (multiply accumulate) hybrid adder , the circuit is operational to perform a MAC (multiply accumulate) operation and to perform a multiply operation without interfacing with the accumulate value of MAC operation using Quantum-dot cellular automata (QCA). And Compare the area and power with existing adders.

Keywords: Majority Gates, Quantum-dot Cellular Automata,. Multiply Accumulate, Hybrid Adder.

I.INTRODUCTION

A.Quantum-dot Cellular Automata(QCA)

CMOS technology has experienced serious problems such as short channel effects, doping fluctuations, increasingly difficult and expensive lithography at nano scale, high leakage current and speed limitation in GHz range. Many alternative technologies have been introduced in the Industry Technology Roadmap for Semiconductors (ITRS). A Quantum-dot cellular automaton (QCA) is one of these promising nanotechnologies that could be utilized instead of conventional transistor technology in the future. Quantum-dot Cellular Automata provides new possibilities to achieve outstanding properties such as extremely high density and fast operation speed at Tera Hertz frequencies together with low power dissipation .

QCA circuits are implemented with two principle gates: majority and inverter gates. As the basic element in QCA is majority gate, the structure of this gate is a significant factor in designing circuits in QCA. The basic element of QCA is a cell. A QCA cell, shown in Fig 1, due to the Columbic interactions consists of four quantum dots at the corners of a square with two extra mobile electrons in two different configurations, thus we have two polarizations (p= +1 ,p= -1) used for encoding binary information. Quantum Cellular Automata(QCA) is a structure made up of identical cells realized though a variety of technologies such as electrodynamic, ferromagnetic and molecular. Molecular QCA is particularly attractive because of its projected density of $1x10^{12}$ devices/cm^2 and switching speeds in the THz range.The Columbic interactions between two neighbour cells cause these cells to have the same polarizations. Therefore a series of QCA cells performs as wire in QCA.

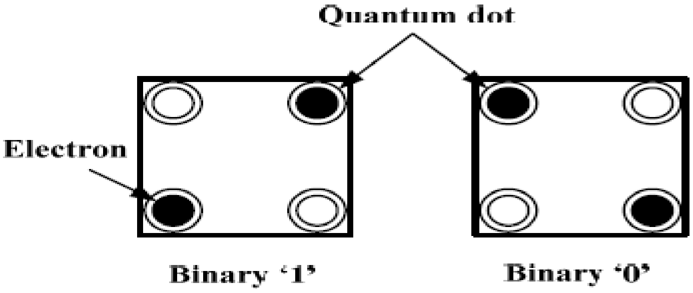


Fig. 1 Basic QCA Cell and two possible polarizations

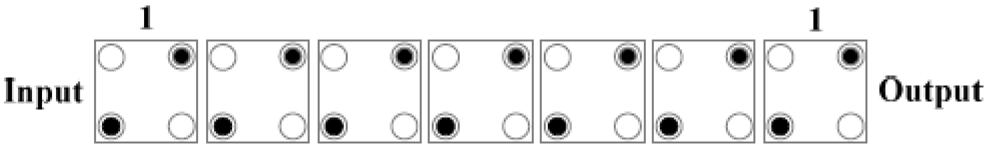


Fig. 2 QCA wire

B. Three Input Majority Gate

The principle QCA logical circuits are majority and inverter gates. The Basic QCA Cell and two possible polarizations and corresponding QCA wire is shown in figure 1 and figure 2 respectively. Fig 3 demonstrates two different inverter gates.

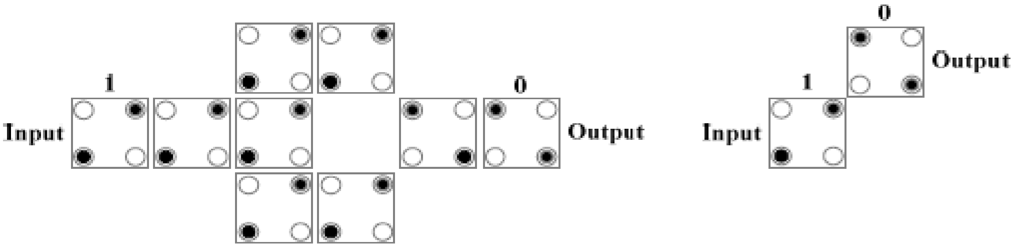


Fig. 3 QCA inverter gates

Up to now, most circuits have been implemented by three majority gate that act based on Eq. (1) and its corresponding QCA as well as symbol is shown below in figure 4.

$$M(A,B,C)=AB+AC+BC \longrightarrow \quad \mathbf{(1)}$$

QCA promises to provide the highest device density with low power consumption and high switching speeds. In addition,QCA uses the same technology to build both the logic gates and the wires carrying logic signals.

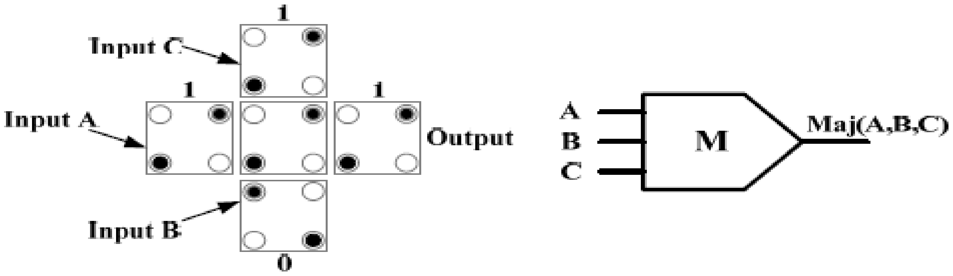


Fig. 4 QCA three-input Majority voter

II.EXISTING METHOD

A.QCA Design of Ladner–Fischer Adder

The prefix graph of a 16-bit ladner–fischer adder (assuming non-zero initial carry and denoted by $C_{0)}$. It is worth noting that the 8-bit prefix graph is a subset of Fig.5 and corresponds to the portion to the right of $C_{8.}$ Fig. 5 assumes availability of g_i's and p_i 's, where g_i and p_i are defined as x_iy_i and $x_i+y_{i,}$ respectively. The small shaded circle in Fig. 3.1 represents the associative operation “0” and is defined as shown in

$$(g_2,p_2)0(g_1,p_1)=(g_2+p_2g_1,p_2p_1) \longrightarrow \quad (2)$$

With reference to Fig. 3.1, the carry c_1 is calculated as .$c_1=g_0+p_0c_o$ This can be written in terms of the operation as

$$(c_1,0)=(g_0,p0)o(c_0,0) \longrightarrow \quad (3)$$

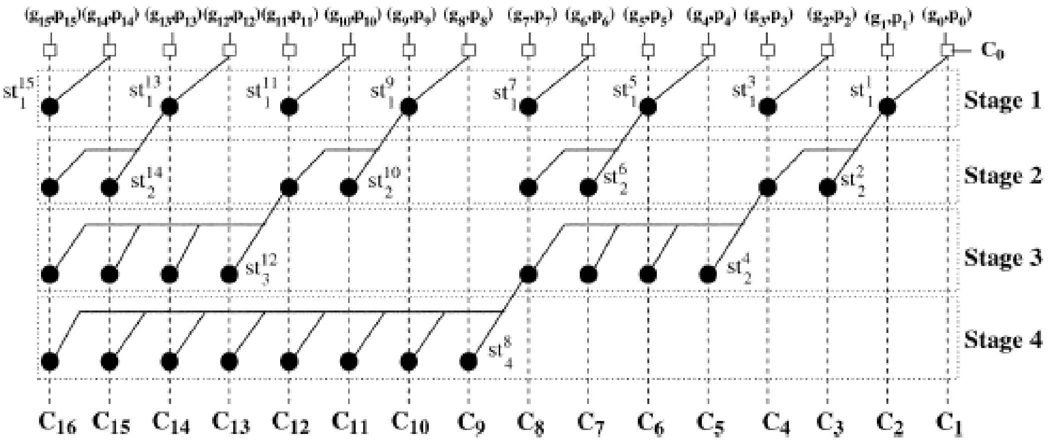

(g15,p15)(g14,p14) (g13,p13)(g12,p12)(g11,p11) (g10,p10) (g9,p9) (g8,p8) (g7,p7) (g6,p6) (g5,p5) (g4,p4) (g3,p3) (g2,p2) (g1,p1) (g0,p0)
C0
st15 1
st13 1
st11 1
st9 1
st7 1
st5 1
st3 1
st1 1
st14 2
st10 2
st6 2
st2 2
st12 3
st4 2
st8 4
Stage 1
Stage 2
Stage 3
Stage 4
C16 C15 C14 C13 C12 C11 C10 C9 C8 C7 C6 C5 C4 C3 C2 C1

Fig. 5 Ladner–Fischer 16-bit adder prefix graph

The calculation of carries of an 8-bit Ladner–Fischer adder therefore requires three stages (while four stages are required for a 16-bit Ladner–Fischer adder) excluding the stage that involves calculation of g_i and p_i The direct calculation of carry $c_1 = g_0 + p_0c_0$ requires two majority gates, namely one for AND operation and another for OR operation. We now present a new proposition that shows that c_1 requires only one majority gate. Satisfy $g_ip_i = g_i$ *and* $g_i = p_i = p_{i.}$ Similarly,$(g_{i+1},pi_{+1})_{o} (g_ip_i)$ is calculated as,

$$(g_{i+1},pi_{+1})_{o} (g_ip_i)=(M(,(g_{i+1},pi_{+1,}g_i) ,M(pi_{+1,} pi,0)) \longrightarrow (4)$$

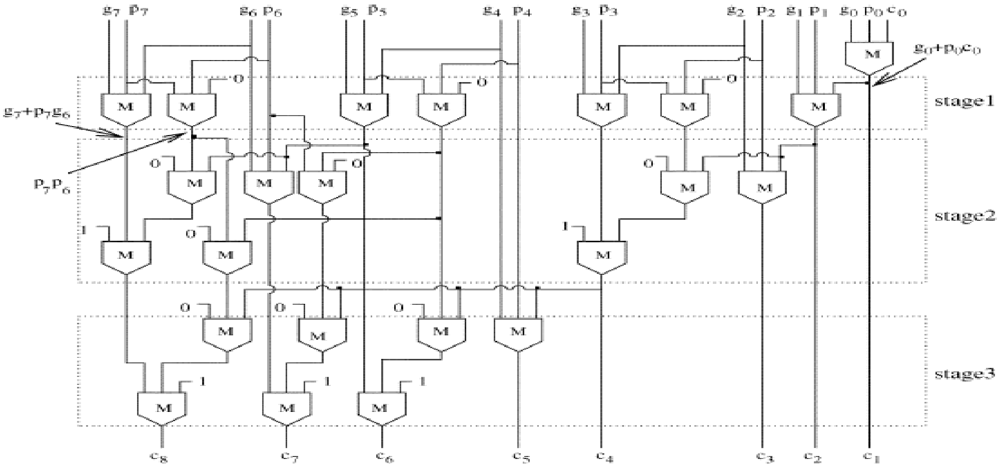


Fig. 6 Majority gate diagram for carry generation of 8-bit Ladner–Fischer adder

Using these results, we have the majority gate based representation for carry generation in an 8-bit Ladner–Fischer adder as shown in Fig. 6. Now we proceed to describe how the eight sums, denoted by s_i , I = 0,…,7 for an 8-bit Ladner–Fischer adder are generated. It is worth noting that the requirement for s_0 can be duplicated for the remaining sum bits $s_{1,\ldots,}$ $s_{7.}$ where the authors develop an expression for carry and sum in a one-bit full-adder as shown in (4). This requires three majority gates and two inverters.

$$c_{i+1} = M(x_i, y_i, c_i) \longrightarrow \quad (5)$$

$$s_i = x_i \otimes y_i \otimes c_i$$
$$= M(\overline{c_{i+1}}, M(x_i, y_i, \overline{c_i}), c_i). \longrightarrow \quad (6)$$

To reduce the inverter requirements for sum s_i in particular, two relations as shown in (5) and (6) can be used. We omit the proof of the results in (5) and (6) since they directly follow from the definition of majority function for three Boolean variables.Considering the left half corresponding to generation of C_{16} to C_9 to in Fig.6 we note that Stages 1, 2, and 3 involve four associative operations each. Each associative operation requires three majority gates. Hence, a total of 36 majority gates is required for these three stages. Stage 4 involves eight associative operations (that is, *n/2,* where *n* is 16).

B. 16-BIT HYBRID ADDER

Ladner–Fischer adder supports parallelism, the requirement of majority gates (which contributes to the overall area) is quite high. The large number of majority gates has an indirect effect on the wire (delay and amount). The proposed hybrid adder is based on the idea that a number of carries not explicitly labelled in Fig. 4.2 (in particular, c_1 , c_2 , c_3 , c_4 , c_5 , c_6 , c_9 , c_{10} , c_{13} , c_{14}*.)* highly efficient adder in terms of majority gates and delay. The majority gate requirement for carries c_{16} , c_{12} , c_8 , and c_4. With reference to fig. 6, we note that c_{16} , c_{12} , and c_8 depend on , G_3 , G_2 , G_1 and , respectively. Let , G_j , j = 1, 2, 3 represent the generate of ($g_i +3$,$p_i +3$) o ($g_i +2$,$p_i +2$) o ($g_i +1$,$p_i +1$) o (g_i ,p_i)where i= 4, 8, and 12, respectively. G_j Can be expanded as

$$G_j = g_i +3 + ,p_i +3 (g_i +2 + ,p_i +2 (g_i +1 + ,p_i +1 g_i)). \longrightarrow (7)$$

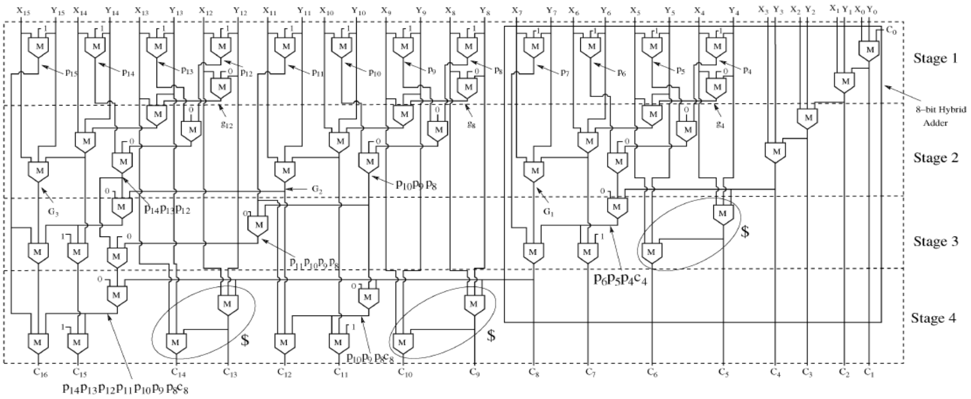

X15 Y15 X14 Y14 X13 Y13 X12 Y12 X11 Y11 X10 Y10 X9 Y9 X8 Y8 X7 Y7 X6 Y6 X5 Y5 X4 Y4 X3 Y3 X2 Y2 X1 Y1 X0 Y0
C0
M
p15
p14
p13
p12
p11
p10
p9
p8
p7
p6
p5
p4
g12
g8
g4
G3
G2
G1
p14p13p12
p10p9 p8
p11p10p9 p8
p6p5p4c4
p10p9 p8c8
p14p13p12p11p10p9 p8c8
$
Stage 1
8–bit Hybrid Adder
Stage 2
Stage 3
Stage 4
C16 C15 C14 C13 C12 C11 C10 C9 C8 C7 C6 C5 C4 C3 C2 C1

Fig. 7 Carry generation for a 16-bit hybrid adder

Using , G_3 , G_2 , and G_1, compute , c_8 , c_{12} , and c_{16}, as shown below equation respectively. It is to be emphasized that obtaining c_8 (as also c_{12} and c_{16}) this way using g_i's and p_i's provides for higher parallelism (when compared to a ripple carry adder). c_8 , c_{12} , and c_{16}, three other carries are computed in prefix style.

$$c_8 = G_1 + p_7\, p_6\, p_5\, p_4\, c_4 \longrightarrow (8)$$

$$c_{12} = G_2 + p_{11}\, p_{10}\, p_9\, p_8\, c_8 \longrightarrow (9)$$

$$c_{16} = G_3 + p_{15}\, p_{14}\, p_{13}\, p_{12}\, G_2 + p_{15}\, p_{14\ldots}\, p_8\, c_{8.} \longrightarrow (10)$$

These are c_7 , c_{11} , and c_{15}, the equations shown above. For c_7 , ten majority gates are required (including the requirements for p_i's and g_4, majority gate for AND of p_6 and p_5, AND of p_4 and c_4 , AND of and $p_6\, p_5$ and $p_4\, c_4$ the OR operation).

III.PROPOSED METHODOLOGY

In proposed to describe about MAC, the circuit is operational to perform a operation and to perform a multiply operation without interfacing with the accumulate value of MAC operation using QCA. The circuit includes a first register, a second register, a multiplier circuit, and an accumulate circuit. The first register is addressable using either a primary first address or an alias second address. The circuit performs multiply operation to generate the product value based on data in the first and second register after a write operation to first register or second register. The MAC circuit must check for overflow, which might happen when the number of MAC operations is large. Overflow in a signed adder occurs when two operands with the same sign produce a result with a different sign.Fig. 8 shows the structure of the proposed MAC unit. The MAC unit consists of the multiplier and an accumulator unit. We first started implementing the multiplier, as it was the major portion of what we were planning to achieve. The multiplier array, that consist of multiplexers, half adders, full adders and the add cell (to add 0 or 1 to the LSB of the partial products)1. The carry save adder in the adder array is used here as it enables a very fast operation of the adding operation.

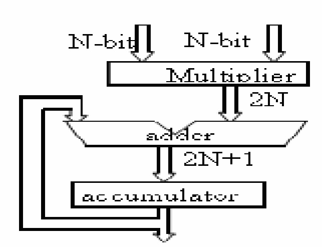
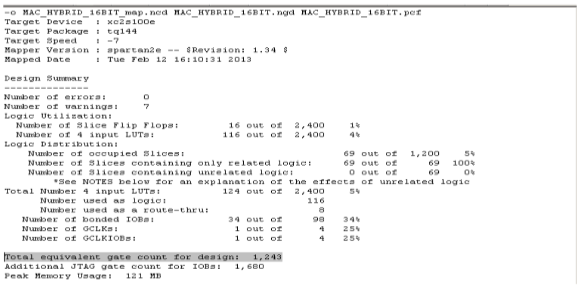


Fig 8 Basic structure of MAC

The accumulator performance can greatly increase the performance of the MAC unit. But the bottleneck for the accumulator is the multiplier unit. The accumulator must wait till it has correct logic values at its input for accumulation. The accumulator unit must comply with our initial objective of low power and high performance.

IV .RESULTS AND DISCUSSIONS

In this paper, we have presented efficient QCA designs for the Ladner–Fischer prefix adder and a hybrid of Ladner–Fischer and the ripple carry adder. The designs are based on new results concerning majority logic. The hybrid adder is shown to be particularly well suited to the QCA model, it has better performance (in terms of latency) in QCA than a Ladner–Fischer or a ripple carry adder. And also show that the hybrid adder has a smaller area-delay product than existing adder designs in QCA. Area, power are Compared with prior work are presented and the detailed simulation results are also given.

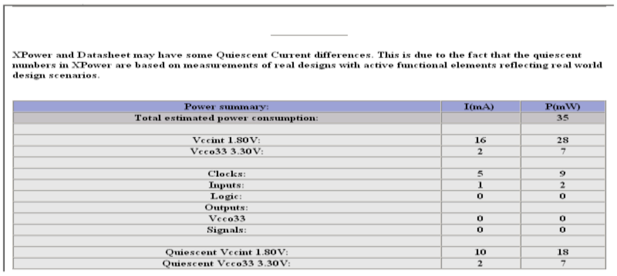


```
-o MAC_HYBRID_16BIT_map.ncd MAC_HYBRID_16BIT.ngd MAC_HYBRID_16BIT.pcf
Target Device  : xc2s100e
Target Package : tq144
Target Speed   : -7
Mapper Version : spartan2e -- $Revision: 1.34 $
Mapped Date    : Tue Feb 12 16:10:31 2013

Design Summary
--------------
Number of errors:      0
Number of warnings:    7
Logic Utilization:
  Number of Slice Flip Flops:        16 out of  2,400    1%
  Number of 4 input LUTs:           116 out of  2,400    4%
Logic Distribution:
    Number of occupied Slices:                        69 out of  1,200    5%
    Number of Slices containing only related logic:   69 out of     69  100%
    Number of Slices containing unrelated logic:       0 out of     69    0%
        *See NOTES below for an explanation of the effects of unrelated logic
Total Number 4 input LUTs:          124 out of  2,400    5%
      Number used as logic:                     116
      Number used as a route-thru:                8
   Number of bonded IOBs:            34 out of     98   34%
   Number of GCLKs:                   1 out of      4   25%
   Number of GCLKIOBs:                1 out of      4   25%

Total equivalent gate count for design:  1,243
Additional JTAG gate count for IOBs:  1,680
Peak Memory Usage:  121 MB
```

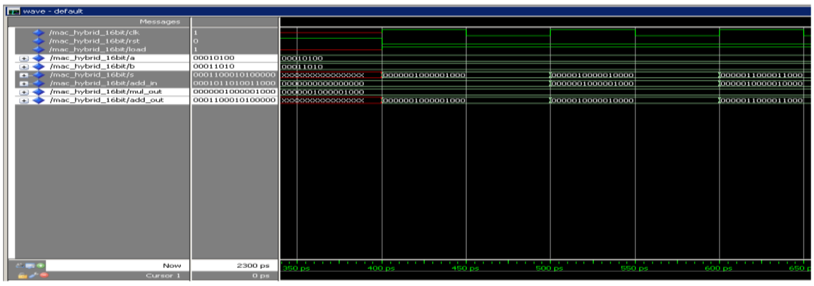
Fig 9. No of gate count for design of Hybrid Adder in MAC



XPower and Datasheet may have some Quiescent Current differences. This is due to the fact that the quiescent numbers in XPower are based on measurements of real designs with active functional elements reflecting real world design scenarios.

Power summary:	I(mA)	P(mW)
Total estimated power consumption:		35
Vccint 1.80V:	16	28
Vcco33 3.30V:	2	7
Clocks:	5	9
Inputs:	1	2
Logic:	0	0
Outputs:		
Vcco33	0	0
Signals:	0	0
Quiescent Vccint 1.80V:	10	18
Quiescent Vcco33 3.30V:	2	7

Fig 10. Total estimated power consumption for hybrid adder in MAC



```
wave - default
Messages
/mac_hybrid_16bit/clk        1
/mac_hybrid_16bit/rst        0
/mac_hybrid_16bit/load       1
/mac_hybrid_16bit/a          00010100
/mac_hybrid_16bit/b          00011010
/mac_hybrid_16bit/s          0001100010100000
/mac_hybrid_16bit/add_in     0001011010011000
/mac_hybrid_16bit/mul_out    0000001000001000
/mac_hybrid_16bit/add_out    0001100010100000
Now                          2300 ps
Cursor 1                     0 ps
```

Fig 11.Simutation Result for Hybrid Adder in MAC

The multiplier array, that consist of multiplexers, half adders, full adders and the add cell (to add 0 or 1 to the LSB of the partial products)1. The carry save adder in the adder array is used here as it enables a very fast operation of the adding operation. The accumulator performance can greatly increase the performance of the MAC unit. Shows that the hybrid adder has better performance (in terms of latency) in Quantum-dot cellular automata **(**QCA) than a Ladner–Fischer or a ripple carry adder.

Table 1 Comparison of Area and power with various adders

S.No	**Adders(16-Bit in MAC)**	**Area**	**Memory Usage**	**Power(mW)**
1	Lander Fisher Adder	2,175gates	122 MB	42
2	**Hybrid Adder**	**1,243 gates**	**121MB**	**35**

Table 2 Output analysis of LUT's & IOB's

Logic utilization	**Used**	**Available**	**Utilization**
Number of 4 input LUT's	533	3,840	13%
Number of occupied slices	279	1,920	14%
Number of bonded IOB's	64	173	36%

ACKNOWLEDGMENTS

The authors would like to thank ,Dr.Mrs.S.Nirmala, Head Of The Department, Department of Electronics and Communication Engineering, Muthayammal Engineering College, for her stimulating comments, which helped me in bringing things in my way.We also acknowledge the Professors who collaborated with us for the design.

V.REFERENCES

[1] Vikramkumar Pudi and K. Sridharan, Senior Member, IEEE, Efficient Design of a Hybrid Adder in Quantum-Dot, IEEE Transactions on Very Large Scale Integration (VLSI) systems, vol. 19, no. 9, september 2011.

[2] S. Srivastava, S. Sarkar, and S. Bhanja, "Estimation of upper bound of power dissipation in QCA circuits," IEEE Trans. Nanotechnol., vol. 8, no. 1, pp. 116–127, Jan. 2009.

[3] T. Dysart and P. M. Kogge, "Analyzing the inherent reliability of moderately sized magnetic and electrostatic QCA circuits via probabilistic transfer matrices," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 17, no. 4, pp. 507–516, Apr. 2009.

[4] H. Cho and E. E. Swartzlander, "Adder and multiplier designs in quantum-dot cellular automata," IEEE Trans. Comput., vol. 58, no. 6, pp. 721–727, Jun. 2009.

[5] K. Kim, K. Wu, and R. Karri, "The robust QCA adder designs using composable QCA building blocks," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 26, no. 1, pp. 176–183, Jan. 2007.

[6] S. Srivastava and S. Bhanja, "Hierarchical probabilistic macromodeling for QCA circuits," IEEE Trans. Comput., vol. 56, no. 2, pp. 174–190, Feb. 2007.

[7] H. Cho and E. E. Swartzlander, "Adder designs and analyses for quantum-dot cellular automata," IEEE Trans. Nanotechnol., vol. 6, no. 3, pp. 374–383, May 2007.

[8] Hanninen and J. Takala, "Robust adders based on quantum-dot cellular automata," in Proc. IEEE Int. Conf. Appl.-Specific Syst., Arch.Processors, 2007, pp. 391–396.

[9] J. Huang, M. Momenzadeh, and F. Lombardi, "An overview of nanoscale devices and circuits," IEEE Des. Test Comput., vol. 24, no. 4, pp. 304–311, Jul.–Aug. 2007.

[10]K. Walus, T. Dysart, G. Jullien, and R. Budiman, "QCADesigner: A rapid design and simulation tool for quantum-dot cellular automata,"*IEEE Trans. Nanotechnol.*, vol. 3, no. 1, pp. 26–29, Jan. 2004.

BIOGRAPHY

Bhuvaneswaran.M, received his Bachelor of Engineering degree in Electronics and Communication Engineering from Hindusthan College of Engineering & Technology, Coimbatore, India in 2008. He is currently pursuing Final year M.E VLSI Design degree in Muthayammal Enggineering College, Rasipuram, Namakkal Dist, Tamilnadu, India. His research interests in Area efficient and Low power VLSI .

Shankar.N.K, received his Master of engineering degree in VLSI Design from K.S.R College of Technology, Tiruchengode,Tamilnadu, India in 2011. He is currently working as Assistant Professor in Electronics and communication Engineering Department at Muthayammal Engineering College, Rasipuram, Namakkal Dist,Tamilnadu,India.He is having 2 years of teaching Experience. His research interest in Low power VLSI.