International Journal of Emerging Engineering Research and Technology
Volume 4, Issue 9, September 2016, PP 18-25
ISSN 2349-4395 (Print) & ISSN 2349-4409 (Online)

Low Area and High Efficient Comparators in Quantaum – Dot Cellular Automata

P. Rushikesawar Reddy, P. Ashok Babu

M.Tech, ECE Department , Malla Reddy Engineering college, Secunderabad, Telangana, India

ABSTRACT

Quantum dot Cellular Automata is a transistor less device which gives answers for density and interconnection. As now a day's CMOS size gets reduced we are encountering many problems and Moore's law of scaling is not obeyed. By using quantum dots we can reduce the size of overall architectures. This paper proposes a new design for implementation of binary comparators in QCA.The work uses novel implementation strategies, formulas of basic logic equations to work comparator efficiently. For design of two different comparators new methods had been exploited. Comparators presented exhibits high speed with low parasitic, with low area. The existing and proposed comparators are synthesized using Xilinx and performance is evaluated in terms of number of gates and delay.

Keywords: Binary comparators, majority gates, Quantum-dot cellular automata (QCA).

INTRODUCTION

CMOS circuit design process consists of defining circuit output and input, circuit calculations, layouts of the circuits, simulations including parasitic, reevaluation of the circuit input and output, fabrication and testing. As the CMOS technology shrinks year to year problems encountering in silicon is more. As the size decreases metal routings and widths of metals are unable to decrease, so parasitic due to metals is more. To overcome this we are going for Q.C.A's. By using Q.C.A's special efforts are made to built arithmetic circuits, adders, multipliers and comparators. QCA implementation existing in literature is mainly provided for comparing two signals. The structures provided in the paper provide higher computational capabilities and circuits able to recognize all three possible conditions in which a=b, a>b, a<b are described. The 32 bit comparator proposed in [1] is improved to 64 bit.

This paper concentrates on different architectures of 64 bit full comparator. The main contribution of this paper is the introduction of novel design methodology that allows low computational time and very low space occupying layouts. The rest of the paper is as follows, an introduction of QCA design and QCA implementation of binary comparator.

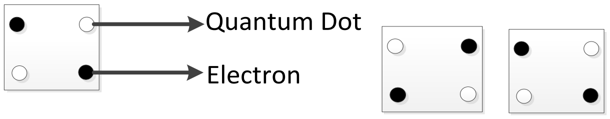
BACKGROUND AND RELATED WORKS

Origin

First cellular automata are implemented as a software programs. However in 1993 a physical implementation of an automation using quantum dot cells is proposed. Automation gained popularity very fast and it was fabricated in 1997. Both automata and quantum mechanics are combined and create a very high switching nano-scale device and consuming less electric power.

Modern Cells

Q.C.A is a transistor level device which gives answers for device density and interconnection. For so many years micro electronics industry has enjoyed dramatic improvements in the speed and size of electronic devices. This has obeyed Moore's law of scaling. The basic Q.C.A cell consists of four dots which act as vessels to trap the electrons. These electrons are of dimension between few nanometers to few micrometers. The distance between quantum dots is about 20nm and distance between cells is about 60nm [1]. Quantum cellular automata are based on simple interaction rules between cells placed in grid. The electrons in the vessels (quantum dots) are movable between those dots. If electrons are there in the vessel it is taken as one position either 1 or 0.



QCA Cell Logic '1' Logic '0'

By lithographic techniques we can construct by using aluminum. Electrons are used to store and transmit data in the cells. These electrons in Q.C.A cell [1] can tunnel and enter into dots. According to the electron placement in the cell we can assign binary states 1 and 0. The nearby cells interact through electrostatic forces and tend to align their polarization or sates. Data cannot flow intrinsically in Q.C.A cells. (fig of Q.C.A cells).To control the data flow in Q.C.A cells power signal is not required, all it need is proper clock zones. Four clock signals are necessary for propagation of information in Q.C.A circuits. Each clock is phase shifted [3] by 90.Each clock zone has four phases named switch, hold, release and relax.

Switch: In this stage the electrons get affected by columbic charges of neighbors because of this force tunneling hindrance is very high.

Hold: In this state the electrons achieved state will not get changed and tunneling hindrance is very high.

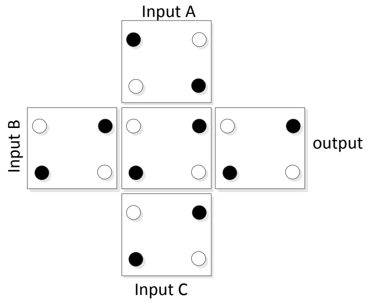
Release: In this state cells are tied to their hold state and tunneling hindrance is started to slow down.

Relax: In this state tunneling barrier is null and cells are free to acquire new values for propagation of information in Q.C.A circuits.

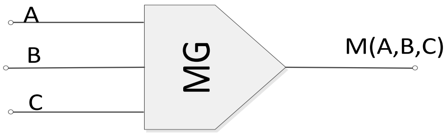
Logic Gates

The fundamental logic gates inherently in the Q.C.A technology are the inverters and [4] majority gates (MG).

$$M\ (a, b, c) = a.\ b + b.\ c + c.\ a$$



Majority Gate



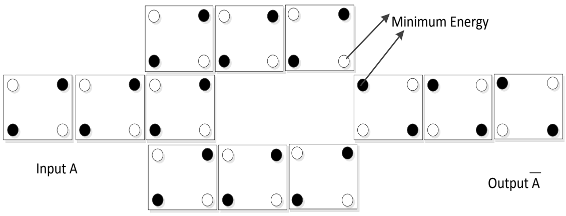
Symbol of Majority gate

Majority Gate

An MG gate consists of three inputs and one output. In this structure the electrical field plays a major role. The direction in which field is more output will be in that direction , whatever the inputs are in majority becomes the out [5] of the cell. Example, if inputs A and B exist in binary 0 state and inputs C exists in binary 1 state, the output will be in binary 0 state, because the electric field of A and B is greater than C. Other types of gates namely AND and OR can be constructed by using MG.

Inverter

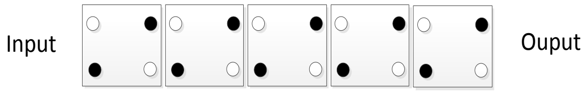
We can construct AND and OR gates by changing the value of any of the variable in the [2] MG gate. Example if we fix C=0 we get AND gate and if we fix C=1 we get OR gate.



Inverter

Q.C.A Wires

The cell which are adjacent to each other try to settle in ground state. The state of adjacent cell depends on the neighboring cells state. So if we apply input to a cell that input is taken forward by another cell by changing polarization of cell to that of adjacent cell. This is how data get propagated from one cell to another.fig of wire.



QCA wire

QCA Implementation of N-Bit Comparators

The theorems and corollaries proposed in this section increases the speed of QCA full comparator and reduces basic gates. The novel formulations can be exploited in the design

Of n-bit full comparators splitting the operands A(n-1:0) = an-1 ..a0 and B(n-1:0) = bn-1…b0 into proper number of 2-bit and 3-bit sub words that can be compared applying theorems 1 and 2.The results obtained in middle can be processed in theorem 3 and 4 together with Corollaries 1 and 2.

Theorem 1: If $A_{(k-1:k-2)} = a_{k-1}a_{k-2}$ and $B_{(k-1:k-2)} = b_{k-1}b_{k-2}$, with k = 2, 4, …., n-2, n, are two 2-bit sub words of the n-bit numbers $A_{(n-1:0)}$ and $B_{(n-1:0)}$, respectively [3] then $A_{big}B_{(k-1:k-2)}$ as defined in (2) is equal to 1 if and only if $A_{(k-1:k-2)} > B_{(k-1:k-2)}$; $\overline{B_{big}A_{(k-1:k-2)}}$as defined in (3) is equal to 0 if and only if $A_{(k-1:k-2)} < B_{(k-1:k-2)}$

$$A_{big}B_{(k-1:k-2)} = M(a_{k-1}, \overline{b_{k-1}}, a_{k-2}).M(a_{k-1}, \overline{b_{k-1}}, b_{k-2}) \quad (2)$$

$$B_{big}A_{(k-1:k-2)} = M(a_{k-1}, \overline{b_{k-1}}, a_{k-2}) + M(a_{k-1}, \overline{b_{k-1}}, \overline{b_{k-2}}) \quad (3)$$

Theorem 2: If $A_{(k-1:k-3)} = a_{k-1}a_{k-2}a_{k-3}$ and $B_{(k-1:k-3)} = b_{k-1}b_{k-2}b_{k-3}$, with k = 2, 4, …., n-3, n, are two 3-bit sub words of the n-bit numbers $A_{(n-1:0)}$ and $B_{(n-1:0),}$ respectively then $A_{big}B_{(k-1:k-3)}$ as defined in (4) is equal to 1 if and only if $A_{(k-1:k-3)} > B_{(k-1:k-2)}$; $\overline{B_{big}A_{(k-1:k-3)}}$ as defined in (5) is equal to 0 if and only if $A_{(k-1:k-3)} < B_{(k-1:k-3)}$.

$$A_{big}B_{(k-1:k-3)} = M(M(a_{k-1}, b_{k-1}, \overline{a_{k-2}}), M(a_{k-1}, \overline{b_{k-1}}, \overline{b_{k-2}}), a_{k-3} . \overline{b_{k-3}} \quad (4)$$

$$\overline{B_{big}A_{(k-1:k-3)}} = M (M(a_{k-1}, \overline{b_{k-1}}, a_{k-2}), M(a_{k-1}, \overline{b_{k-1}}, \overline{b_{k-2}}), a_{k-3} + b_{k-3}) \quad (5)$$

Theorem 3: Given two n-bit numbers $A_{(n-1:0)}$ and $B_{(n-1:0)}$, (6) gives $A_{big}B_{(n-1:0)} = 1$ if and only if $A_{(n-1:0)} > B_{(n-1:0)}$, Whereas (7) gives $\overline{B_{big}A_{(n-1:0)}} = 0$ if and only if $A_{(n-1:0)} < B_{(n-1:0)}$.

$$A_{big}B_{(n-1:0)}) = M(M(a_{n-1}, \overline{b_{n-1}}, a_{n-2}), M(a_{n-1}, \overline{b_{n-1}}, \overline{b_{n-2}}), A_{big}B_{(n-3:0)}) \quad (6)$$

$$\overline{B_{big}A_{(n-1:0)}} = M(M(a_{n-1}, \overline{b_{n-1}}, a_{n-2}), M(a_{n-1}, \overline{b_{n-1}}, \overline{b_{n-2}}), B_{big}A_{(n-3:0)} \quad (7)$$

Theorem 4: Given two n-bit numbers $A_{(n-1:0)}$ and $B_{(n-1:0)}$, $A_{big}B_{(n-1:n-3)}$ and $B_{big}A_{(n-1:n-3)}$ is computed with (4) and (5) , then $A_{big}B_{(n-1:0)}$ as defined in (8) is equal to 1 if $A_{(n-1:0)} > B_{(n-1:0)}$, whereas $\overline{B_{big}A_{(n-1:0)}}$ defined in (9) is equal to 0 if $A_{(n-1:0)} < B_{(n-1:0)}$.

$$A_{big}B_{(n-1:0)} = M(A_{big}B_{(n-1:n-3)}), \overline{B_{big}A_{(n-1:n-3)}}, A_{big}B_{(n-4:0)}) \quad (8)$$

$$\overline{B_{big}A_{(n-1:0)}} = M(A_{big}B_{(n-1:n-3)}), \overline{B_{big}A_{(n-1:n-3)}}, \overline{B_{big}A_{(n-4:0)}}) \quad (9)$$

Corollary

Let's look over two n-bit numbers $A_{(n-1:0)}$ and $B_{(n-1:0)}$,and they are divided into the subword $A_{(n-1:h)}$,$B_{(n-1:h)}$ and $B_{(h-1:0)}$. If $A_{big}B_{(n-1:h)}$, $A_{big}B_{(h-1:0)}$, $\overline{B_{big}A_{(n-1:h)}}$ and $\overline{B_{big}A_{(h-1:0)}}$ are calculated by applying theorems 3 and 4, then $A_{big}B_{(n-1:0)}$ as defined in (10) is equal to 1 if and only if $A_{(n-1:0)} > B_{(n-1:0)}$, whereas $\overline{B_{big}A_{(n-1:0)}}$ as defined in (11) is = 0 if $A_{(n-1:0)} < B_{(n-1:0)}$.

$$A_{big}B_{(n-1:0)} = M (A_{big}B_{(n-1:h)}, \overline{B_{big}A_{(n-1:h)}}, A_{big}B_{(h-1:0)}) \quad (10)$$

$$\overline{B_{big}A_{(k-1:0)}} = M(A_{big}B_{(n-1:h)}, \overline{B_{big}A_{(n-1:h)}}, \overline{B_{big}A_{(h-1:0)}}) \quad (11)$$

Corollary 2:

Let us consider two n-bit numbers $A_{(n-1:0)}$ and $B_{(n-1:0)}$, if $A_{big}B_{(n-1:0)}$ and $\overline{B_{big}A_{(n-1:0)}}$ are calculated by applying theorems 1, 2, 3 and 4 and/or corollary 1, then $A_{eq}B_{(n-1:0)}$ defined in (12) is equal to 1 if $A_{(n-1:0)} = B_{(n-1:0)}$.

$$A_{eq}B_{(n-1:0)} = M(\overline{A_{big}B_{(n-1:0)}}, \overline{B_{big}A_{(n-1:0)}}, 0) .$$

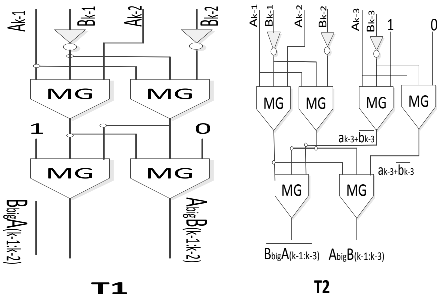
By using above proposed theorems and corollaries we can construct different architectures.

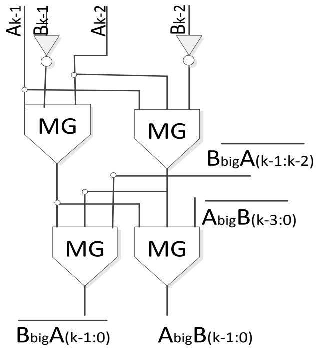
Novel Design of Comparators

The generic module Ti, with i ranging between 1 to 4,implements the equation enunciated in the i^th theorem, where as C1 and C2 compute the signals $A_{big}B_{(k-1:0)}$, $B_{big}A_{(k-1:0)}$, and $A_{eq}B_{(k-1:0)}$. The QCA modules which are represented are used to design two different structures of full comparators here named as cascade-based (CB) and tree based (TB).

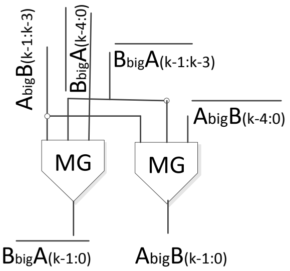
Novel Q.C.A Comparators

For a better understanding of operation, architecture of 64 bit is provided. It is cascaded based 64 bit. It is made by combining two 32 bit architectures.

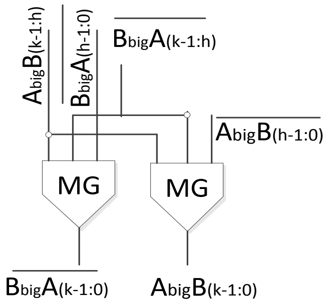




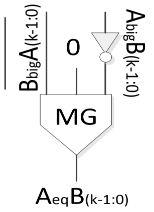
T3



T4



C1



C2

The overall T1 used is two and T2 is n/3 and T4 is n/3 through which $A_{big}B$ and $B_{big}A$ are computed. And one instance of C2 is used. As number of MG's gets increased in the path delay gets increased. One MG module introduces one clock phase in the overall delay.

T1 and T2 contributes two MG's and one inverter so over all three clock cycle delay they contribute and T4 introduces one more MG whereas C2 is responsible for one MG and one inverter. The critical computational path of CB full comparator consists of n/3+3 MG's and two inverters. As number of bits gets increased more MG's get added in the path and delay gets increased.

To overcome this delay we can go for tree based architecture. In this the operands A and B are divided [3] into $A=A_{MSB}A_{LSB}$ and $B=B_{MSB}B_{LSB}$.

TableI.

Splitting of the operands

32-Bit

A(31:29)A(28:26)A(25:23)A(22:20)A(19:17)A(16:14)A(13:11)A(10:8)A(7:5)A(4:2)A(1:0)

B(31:29)B(28:26)B(25:23)B(22:20)B(19:17)B(16:14)B(13:11)B(10:8)B(7:5)B(4:2)B(1:0)

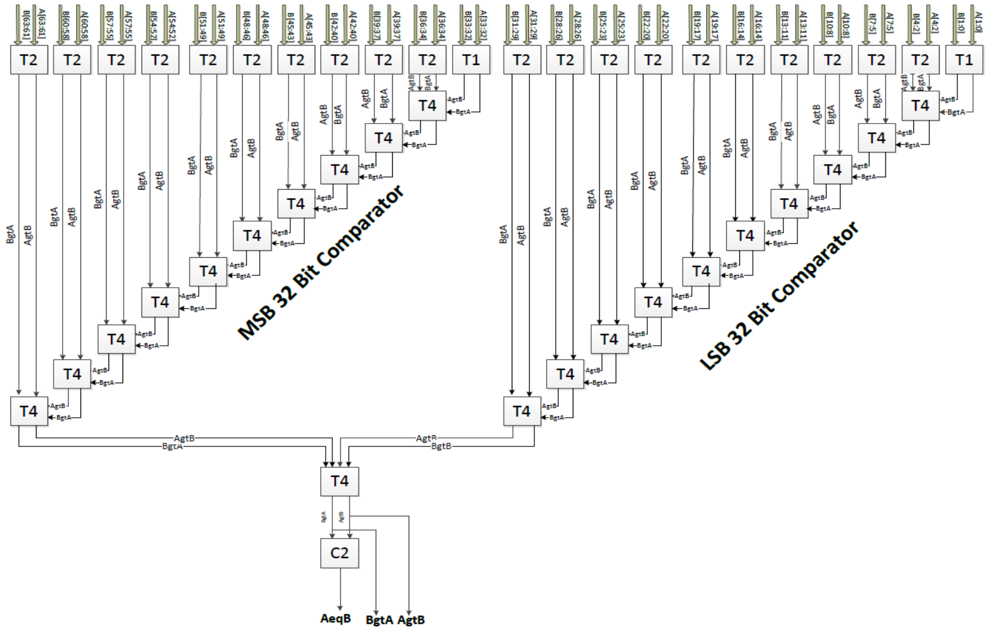
64 –Bit

A(31:29)A(28:26)A(25:23)A(22:20)A(19:17)A(16:14)A(13:11)A(10:8)A(7:5)A(4:2)A(1:0)

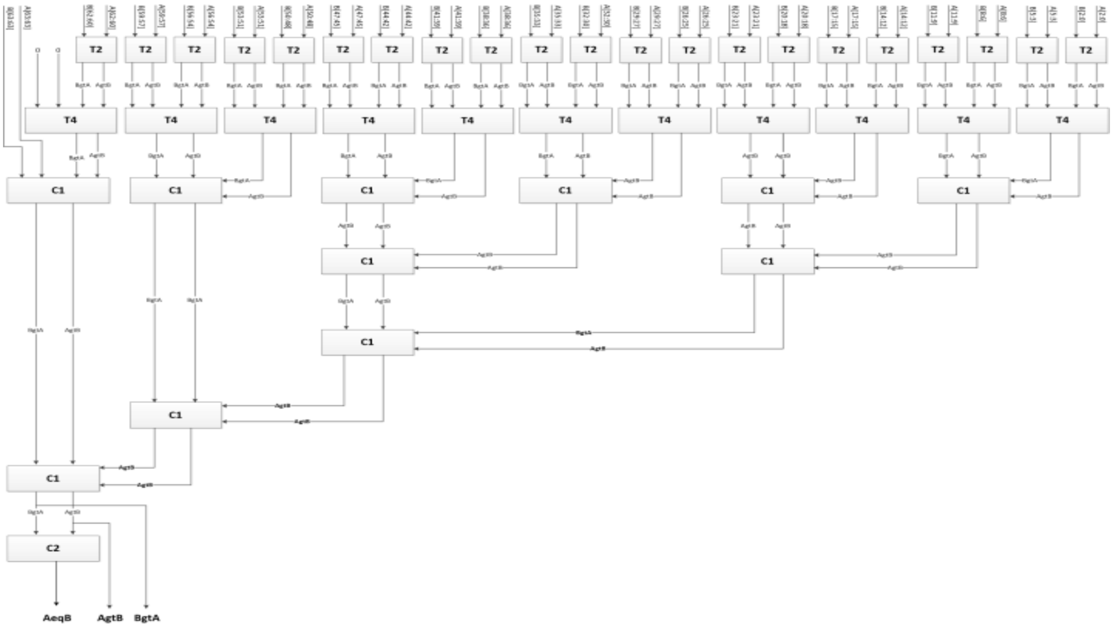
A(63:61)A(60:58)A(57:55)A(54:52)A(51:49)A(48:46)A(45:43)A(42:40)A(39:37)A(36:34)A(33:3)

B(31:29)B(28:26)B(25:23)B(22:20)B(19:17)B(16:14)B(13:11)B(10:8)B(7:5)B(4:2)B(1:0)

B(63:61)B(60:58)B(57:55)B(54:52)B(51:49)B(48:46)B(45:43)B(42:40)B(39:37)B(36:34)B(33:32)



Novel 64-Bit CB Full Comparator



Novel 64-Bit Full TB Comparator

RESULTS

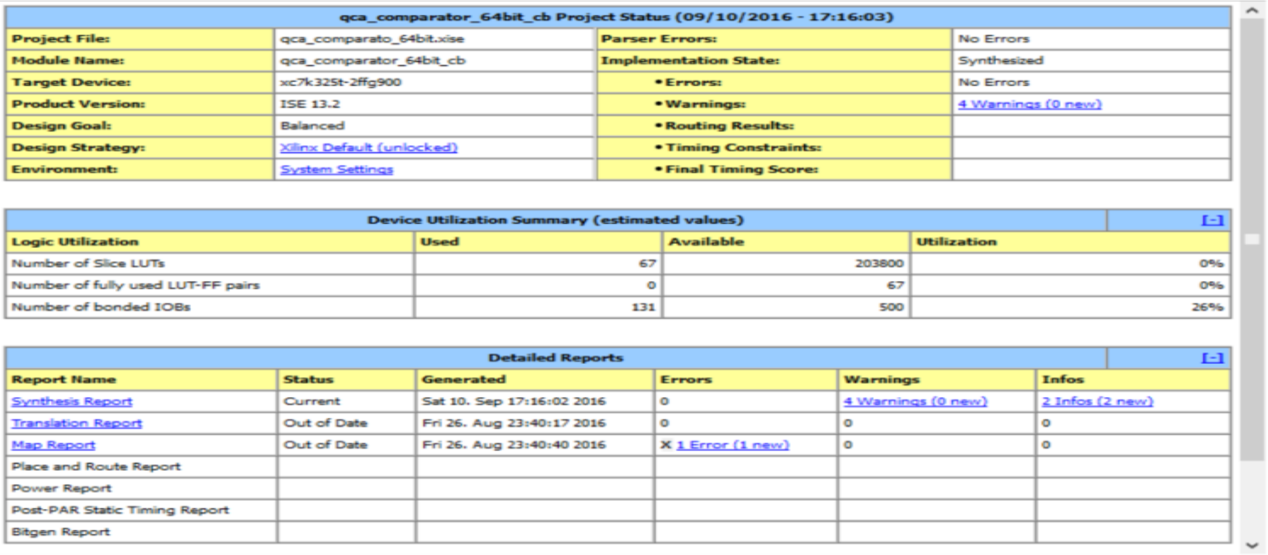
Results obtained for the novel comparators at several operands of 32and 64 bit are reported in the below table. The complexity in design is represented in terms of MG's and inverters within worst computational paths. The computational capability of each architecture is also specified in the table.

TableII. *Reimplementation Results*

Design	Computational capability	Size(n)	MG's	Inverters	MG's in worst computational path
CB-32	full	32	85	33	13
CB-64	full	64	171	107	14
TB-32	full	32	83	33	9
TB-64	full	64	171	108	9

Fig shows number of LUT's utilized, and it also shows number of inputs

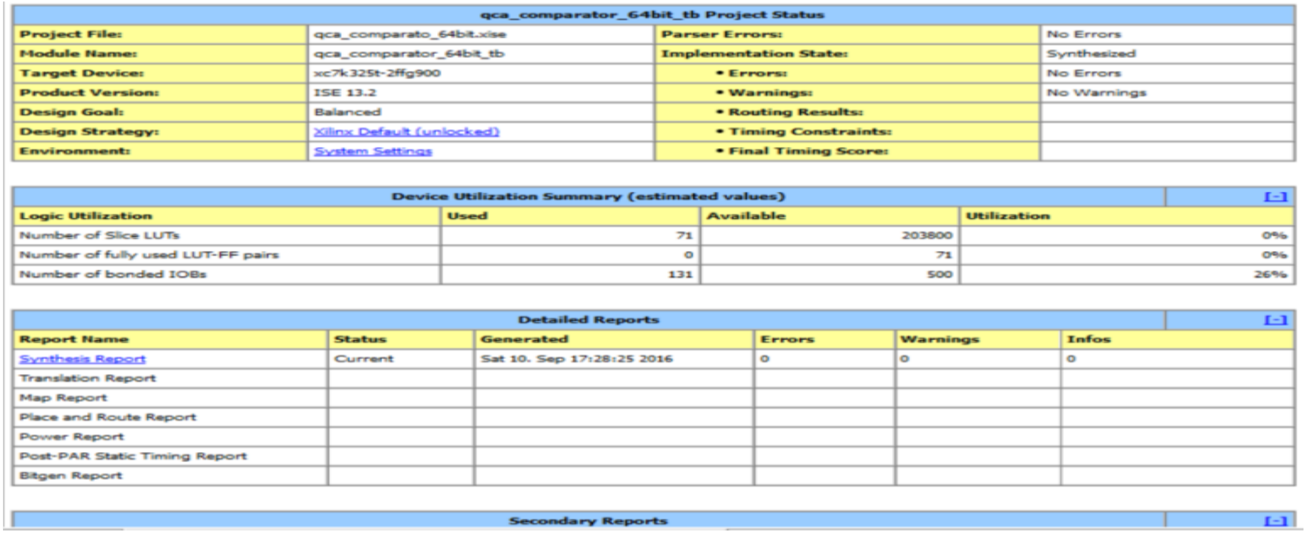
Equivalent Cascaded based Comparator RTL Design



qca_comparator_64bit_cb Project Status (09/10/2016 - 17:16:03)			
Project File:	qca_comparato_64bit.xise	**Parser Errors:**	No Errors
Module Name:	qca_comparator_64bit_cb	**Implementation State:**	Synthesized
Target Device:	xc7k325t-2ffg900	**• Errors:**	No Errors
Product Version:	ISE 13.2	**• Warnings:**	4 Warnings (0 new)
Design Goal:	Balanced	**• Routing Results:**	
Design Strategy:	Xilinx Default (unlocked)	**• Timing Constraints:**	
Environment:	System Settings	**• Final Timing Score:**	

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	67	203800	0%
Number of fully used LUT-FF pairs	0	67	0%
Number of bonded IOBs	131	500	26%

Detailed Reports					[-]
Report Name	**Status**	**Generated**	**Errors**	**Warnings**	**Infos**
Synthesis Report	Current	Sat 10. Sep 17:16:02 2016	0	4 Warnings (0 new)	2 Infos (2 new)
Translation Report	Out of Date	Fri 26. Aug 23:40:17 2016	0	0	0
Map Report	Out of Date	Fri 26. Aug 23:40:40 2016	✗ 1 Error (1 new)	0	0
Place and Route Report					
Power Report					
Post-PAR Static Timing Report					
Bitgen Report					



qca_comparator_64bit_tb Project Status			
Project File:	qca_comparato_64bit.xise	**Parser Errors:**	No Errors
Module Name:	qca_comparator_64bit_tb	**Implementation State:**	Synthesized
Target Device:	xc7k325t-2ffg900	**• Errors:**	No Errors
Product Version:	ISE 13.2	**• Warnings:**	No Warnings
Design Goal:	Balanced	**• Routing Results:**	
Design Strategy:	Xilinx Default (unlocked)	**• Timing Constraints:**	
Environment:	System Settings	**• Final Timing Score:**	

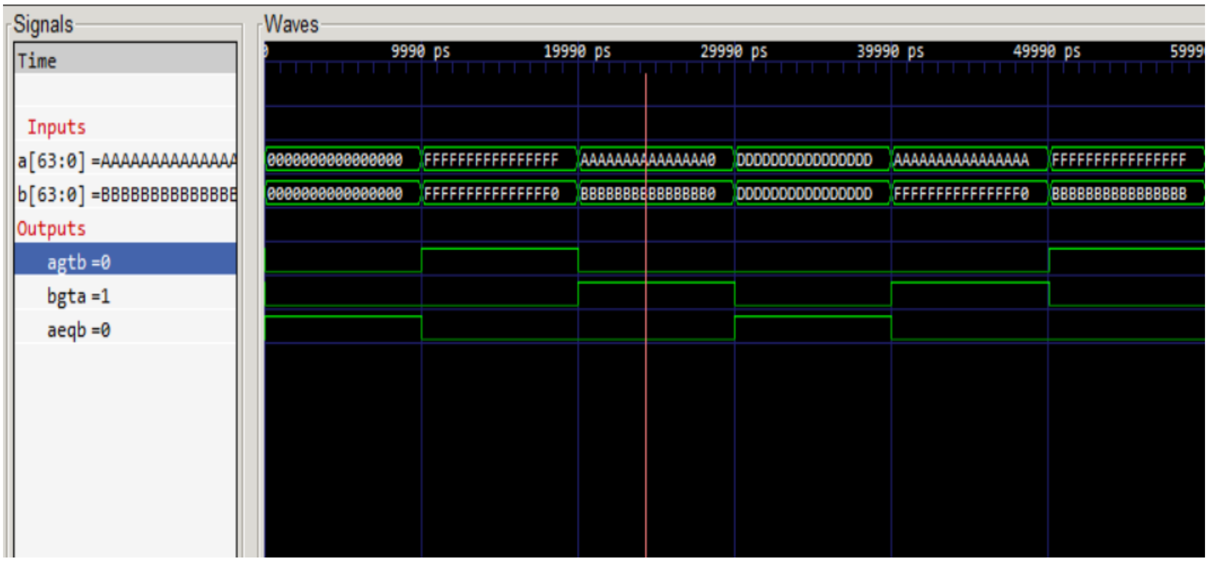
Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	71	203800	0%
Number of fully used LUT-FF pairs	0	71	0%
Number of bonded IOBs	131	500	26%

Detailed Reports					[-]
Report Name	**Status**	**Generated**	**Errors**	**Warnings**	**Infos**
Synthesis Report	Current	Sat 10. Sep 17:28:25 2016	0	0	0
Translation Report					
Map Report					
Place and Route Report					
Power Report					
Post-PAR Static Timing Report					
Bitgen Report					

Secondary Reports [-]

Equivalent Cascaded Based Comparator RTL Design

Fig shows number of LUT's utilized, and it also shows number of inputs.



CONCLUSION

The method to design binary QCA comparators is based on formulations which increase speed performance and reduce overall size. The novel comparator splits the given inputs into 2 and 3 bit sub words by applying thermos.

REFERENCES

[1] C. S. Lent, P. D. Tougaw, W. Porod, and G. H. Bernestein, “Quantum cellular automata,” Nanotechnology,vol.4,no.1,pp.49–57,1993.
J.Huang and F. Lombardi, Design and Test of Digital Circuits by Quantum-Dot Cellular Automata. Norwood, MA, USA: Artech House, 2007

[2] Stefania Perri, Pasquale Corsonello, Giuseppe Cocorullo “Design of Efficient Binary Comparators in Quantum-Dot Cellular Automata”

[3] W. Liu, L. Lu, M. O’Neill, and E. E. Swartzlander Jr., “Design rules forquantum-dot cellular automata,” in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), Rio DeJaneiro, Brazil, May2011, pp. 2361–2364.

[4] K. Navi, M. H. Moaiyeri, R. F. Mirzaee, O. Hashemipour, and B. M. Nezhad, “Two new low-power full adders based on majority-not gates,” Microelectron. J., vol. 40, pp. 126–130, 2009.

AUTHORS’ BIOGRAPHY

P. Rushikesawar Reddy, Born in Anantapur, Andhra Pradesh in 1992.Completed schooling in Kurnool. Received B.tech degree from JNTU in 2013.

Mr. P. Ashok Babu, obtained B.E Degree in 2001 from Andhra University, M.E (Communication Engineering) in 2005 Osmania University. He pursuing the Ph.D. from JNT University, Hyderabad in Digital Image Processing. He published Seven papers in International journals and three paper in International Conference. He is member of IEEE, IACSIT and IAENG. Presently he is working as Assoc. Professor, Department of ECE, Malla Reddy Engineering College, Dhullapally, Hyderabad, Andhra Pradesh (state) India.