DESIGN OF EFFICIENT BINARY COMPARATORS IN QUANTUM-DOT CELLULAR AUTOMATA

Tadimalla Rajeswari[1]

trajeswari64@gmail.com[1]

Venna Prathap Reddy[2]

vennaprathap@gmail.com[2]

[1]PG Scholar, Dept of ECE, Nalanda Institute of Engineering & Technology, Sattenapalli, Guntur,A.P.

[2] Assisant Professor, Dept of ECE, Nalanda Institute of Engineering & Technology, Sattenapalli, Guntur,A.P.

Abstract:

Nowadays exponential advancement in reversible computation has lead to better fabrication and integration process. It has become very popular over the last few years since reversible logic circuits dramatically reduce energy loss. It consumes less power by recovering bit loss from its unique input-output mapping. This paper presents two new gates called RC-I and RC-II to design an n-bit signed binary comparator where simulation results show that the proposed circuit works correctly and gives significantly better performance than the existing counterparts. An algorithm has been presented in this paper for constructing an optimized reversible n-bit signed comparator circuit. Moreover some lower bounds have been proposed on the quantum cost, the numbers of gates used and the number of garbage outputs generated for designing a low cost reversible signed comparator. The comparative study shows that the proposed design exhibits superior performance considering all the efficiency parameters of reversible logic design which includes number of gates used, quantum cost, garbage output and constant inputs. This proposed design has certainly outperformed all the other existing approaches.

Keywords-Reversible Comparator, Quantum Computing, Signed Arithmetic, Low Power, Binary comparators, majority gates.

I. Introduction

In recent time every sort of computation is getting complex and researchers are facing enormous challenges over billions of arithmetic operations per second. According to Gordon Moore, the transistor count and performance of logic circuits is doubled in every two years, this process will continue until semiconductor circuits reach to its physical limit. Thus achieving ultra-speed computation leads us to various kinds of computing technology such as Quantum Information Processing [2], DNA Computing [3], etc. But we live in a world of classical or irreversible architecture where unused energy is dissipated due to power loss. This is because, Landauer [4] proved erasure of each bit of information dissipates at least $KT \times \ln2$ joules of energy where K is the Boltzamann's constant and T is the absolute temperature at which the operation is being performed. In 1973, Bennet [2, 5] had shown that energy dissipation problem of VLSI circuits can be overcome by using reversible logic. This is so because reversible computation does not erase any bit of information and consequently it does not dissipate any energy for computation. Generally, reversible logic performs Boolean operations having equal number of inputs and outputs where input states are uniquely mapped to individual output states or vice versa [6]. Also reversible logic should never permit feedbacks. Reversible logic does multiple operations per cycle without losing any input bits. As a result zero power dissipation would be achieved if a logic circuit consists of reversible gates. Reversible logic can easily be manipulated for Fault Testing

[5], Embedded Devices [7], Digital Signal Processing [8], Quantum Dot Cellular Automata [9-10], etc.

Comparison of two binary numbers finds its wide application in general purpose microprocessors, communication systems, encryption devices, sorting networks, etc [11]. This paper presents an n bit signed comparator which has less number of gates, produces less garbage outputs and quantum cost. In the process of designing this architecture, two very efficient reversible gates RC-I and RC-II have also been proposed. Quantum realization of these two gates shows that they significantly improve the overall cost of the proposed n-bit signed binary comparator. With the help of theorems and lemmas the efficiency of reversible logic synthesis of n-bit signed comparator has also been proved in this paper. This design is based on the reversible comparator gate RC-I which produces 2 outputs q= xLy(xless than y)= x'y and r= xGy(x greater than y)= xy' and RC-II gate which produces 3 outputs r= xEy(xequals y)= $(x \oplus y)'$, q= xLy(xless than y)= x'y and p= xGy(xgreater than y)= xy'. RC-I has been used to compare two single bits whereas RC-II compares two signed bits therefore they are named as Reversible Comparator gate RC-I and RCII.

Cellular automata

The information storage and transport on quantum-dot cellular automata is not based on the flow of electrical particle current, but on the local position of the charged particles inside a small section of the circuit, called a cellular automaton. This QCA cell has a limited number of quantum-dots, which the particles can occupy, and these dots are arranged such that the cell can have only two polarizations (two degenerate quantum mechanical ground states), representing binary value zero or one. A cell can switch between the two states by letting the charged particles tunnel between the dots quantum mechanically.

The cells exchange information by classical Coulombic interaction. An input cell forced to a polarization drives the next cell into the same polarization, since this combination of states has minimum energy in the electric field between the charged particles in neighboring cells. Information is copied and propagated in a wire consisting of the cell automata. Figure 1 shows the available two cell types, which are orthogonal and have minimal interaction with each other, enabling the coplanar wire crossing, where the wires consist of different cell types and can operate independently on the same fabrication layer. A traditional multi-layer crossing can be constructed with either cell type, but the technology has not been demonstrated yet.

II.Literature Survey

Before going into the detail of reversible n-bit signed comparator, some preliminaries on reversible computations and related topics have been discussed in this section.

Reversible Gate

Reversible Gate is an $n \times n$ data stripe block which uniquely maps between inputvector Iv= (I0, I1 ,. . . In) and output vector Ov= (O0, O1, . . . On) denoted as Iv↔Ov
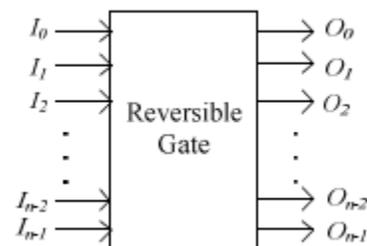

Fig 1. A k× kReversible Gate

The basic element of a nanostructure based on QCA is a square cell with four quantum dots and two free electrons. The latter can tunnel through the dots within the cell, but, owing to Coulombic repulsion, they will always reside in opposite corners, thus leading to only two possible stable states, also named polarizations. Locations of the electrons in the cell are associated with the binary states 1 and 0.

Adjacent cells interact through electrostatic forces and tend to align their polarizations. However, QCA cells do not have intrinsic data flow directionality. Therefore, to achieve controllable data directions, the cells within a QCA design are partitioned into the so-called clock zones that are progressively associated with four clock signals, each phase shifted by 90∘ . This clock scheme, named the zone clocking scheme, makes the QCA designs intrinsically pipelined, since each clock zone behaves like a D-latch.

QCA cells are used for both logic structures and interconnections that can exploit either the coplanar cross or the bridge technique. The fundamental logic gates inherently available within the QCA technology are the inverter and the majority gate (MG). Given three inputsa, bandc, the MG performs the logic function reported in (1) provided that all input cells are associated with the same clock signal clk$_x$ (with x ranging from 0 to 3), whereas the remaining cells of the MG are associated with the clock signal clk$_{x+1}$

$$M(a, b, c) = a \cdot b + a \cdot c + b \cdot c$$
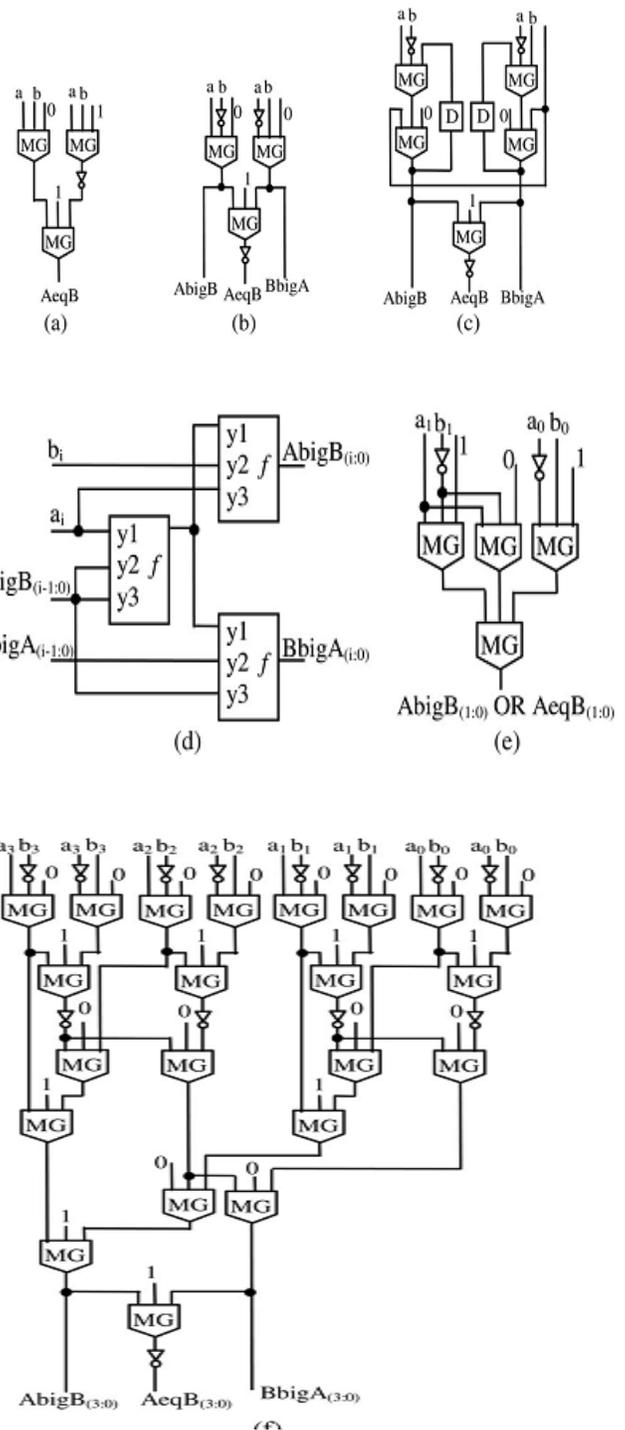
There are several QCA designs of comparators in the literature. A 1-bit binary comparator receives two bits a and b as inputs and establishes whether they are equal, less than or greater than each other. These possible states are

represented through three output signals, here named A eq B, A big B,B big A, that are asserted, respectively, when a=b, a>b, and a<b. Full comparators are those that can separately identify all the above cases, whereas non-full comparators recognize just one or two of them. As an example, the comparator designed in and depicted in Fig. 1(a) can verify only whether a=b. Conversely, the circuits shown in Fig. 1(b) and (c), are full comparators. The latter also exploits two 1-bit registers D to process n-bit operands serially from the least significant bit to the most significant one.

With the main objective of reducing the number of wire crossings, which is still a big challenge of QCA designs.the universal logic gate (ULG)f(y1,y2,y3)=

$M(M(y1,y2,0),M(y1,y3,1),1)$was proposed and then used to implement the comparator illustrated in Fig. 1(d). It is worth noting that, two n-bit numbers $A_{(n-1:0)}=a_{n-1}...a_0$ and $B_{(n-1:0)}=b_{n-1}...b0$can be processed by cascading n instances of the 1-bit comparator. Each instance receives as inputs the ith bits ai and bi (with i=n−1,...,0) of the operands and the signals$A_{big}B_{(i-1:0)}$ and $B_{big}A_{(i-1:0)}$. The former is asserted when the sub word $A_{(i-1:0)}=a_{i-1}...a_0$ represents a binary number greater than $B_{(i-1:0)}=b_{i-1..}.b_0$. In a similar way, B_{big} $A_{(i-1:0)}$ is set to 1 when $A_{(i-1:0)}$ $<B_{(i-1:0)}$.The outputs A_{big} $B_{(i:0)}$ and B_{big} $A_{(i:0)}$ directly feed the next stage. It can be seen that this circuit does not identify the case in which A=B, therefore it cannot be classified as a full-comparator.

The design described exploits a tree-based (TB) architecture and exhibits a delay that in theory logarithmically increases with n. The 2-bit version of such designed comparator is illustrated in Fig. 1(e).

Also the full comparator exploits a TB architecture to achieve high speed. As shown in Fig. 1(f), where 4-bit operands are assumed, one instance of the 1-bit comparator presented for

each bit position. The intermediate results obtained in this way are then further processed through a proper number of cascaded 2-input OR and AND gates implemented by means of MGs having one input permanently set to 1 and 0, respectively.

Analyzing existing QCA implementations of binary comparators it can be observed that they were designed directly mapping the basic Boolean functions consolidated for the CMOS logic designs to MGs and inverters, or ULGs. Unfortunately, in this way the computational capability offered by each MG could be underutilized. As a consequence, both the complexity and the overall delay of the resulting QCA designs could be increased in vain.

III. New Formulations for QCA Implementations of N-Bit full comparators

In this section, four original theorems and two corollaries are enunciated that can significantly increase the speed performances of QCA-based designs of full comparators and can significantly reduce the number of used MGs and inverters with respect to existing comparators, thus reducing also the number of used cells and the overall active area. The Appendix at the end of the paper provides the proofs of the novel theorems and corollaries.

The novel formulations can be exploited in the design of n-bit full comparators splitting the operands $A_{(n-1:0)}=a_{n-1...}a_0$ and $B_{(n-1:0)}=b_{n-1...}b_0$ into a proper number of 2-bit and 3-bit sub words that can be compared applying Theorems 1 and 2. The intermediate results obtained in this way can be then further processed by applying Theorems 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 then-bit numbers $A_{(n-1:0)}$and $B_{(n-1:0)}$, respectively, then A big $B_{(k-1:k-2)}$as defined

in (2) is equal to 1 if and only if $A_{(k-1:k-2)} > B_{(k-1:k-2)}$; B big $A_{(k-1:k-2)}$ as defined in (3) is equal to 0 if and only if $A_{(k-1:k-2)} < B_{(k-1:k-2)}$

$$A_{\text{big}}B_{(k-1:k-2)} = M\left(a_{k-1}, \overline{b_{k-1}}, a_{k-2}\right)$$
$$\cdot M\left(a_{k-1}, \overline{b_{k-1}}, \overline{b_{k-2}}\right)$$
$$\overline{B_{\text{big}}A_{(k-1:k-2)}} = M\left(a_{k-1}, \overline{b_{k-1}}, a_{k-2}\right)$$
$$+ M\left(a_{k-1}, \overline{b_{k-1}}, \overline{b_{k-2}}\right)$$

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 $k=3,6,...,n-3,n$, are 3-bit sub words of then-bit numbers $A_{(n-1:0)}$ and $B_{(n-1:0)}$, respectively, then $A_{\text{big}} B_{(k-1:k-3)}$ as defined in (4) is equal to 1 if and only if $A_{(k-1:k-3)} > B_{(k-1:k-3)}$; B big $A_{(k-1:k-3)}$ as given in (5) is equal to 0 if and only if $A_{(k-1:k-3)} < B_{(k-1:k-3)}$

$$A_{\text{big}}B_{(k-1:k-3)} = M(M\left(a_{k-1}, \overline{b_{k-1}}, a_{k-2}\right),$$
$$M\left(a_{k-1}, \overline{b_{k-1}}, \overline{b_{k-2}}\right), a_{k-3} \cdot$$
$$\overline{B_{\text{big}}A_{k-1:k-3}} = M(M\left(a_{k-1}, \overline{b_{k-1}}, a_{k-2}\right),$$
$$M\left(a_{k-1}, \overline{b_{k-1}}, \overline{b_{k-2}}\right), a_{k-3} \cdot$$

IV. Designing Binary Comparators Exploiting the New Theorems

The circuits illustrated in Fig. 2 were designed to implement in QCA the novel equations demonstrated in the previous Section. The generic module Ti, with I ranging between 1 and 4, implements the equations enunciated in the ith theorem, whereas C1 and C2 compute the signals $A_{\text{big}}B_{(k-1:0)}, B_{\text{big}}A_{(k-1:0)}$, and $A_{\text{eq}}B_{(k-1:0)}$ as shown above in Corollaries 1 and 2, respectively. As examples of application, the above QCA modules have been used to design two different structures of full comparators here named cascade-based and TB architectures. However, many other structures can be designed by combining the basic modules in different manners.

A. Novel QCA Comparators

The first proposed comparator exploits a cascade-based (CB) architecture. To explain better how the overall computation is performed, the schematic diagram illustrated in Fig. 3 is provided. It shows a possible implementation of a 32-bit comparator based on the proposed theory. Following the criterion illustrated in Fig. 3, an n-bit CB full comparator designed as proposed here uses: n/3 instances of T1 and/or T2; n/3 cascaded instances of T4 through which the signals $A_{\text{big}}B_{(n-1:0)}$ and $B_{\text{big}}A_{(n-1:0)}$ are computed; and one instance of C2, needed to compute also $A_{\text{eq}}B_{(n-1:0)}$. Circles visible in Fig. 3 indicate the additional clock phases that have to be inserted on wires to guarantee the correct synchronization of the overall design.

The CB full comparator was designed for operands word lengths ranging from 2 to 32 and using, for n>2, the split criterion summarized in Table I. Obviously, alternative splits could be used.

(a)   (b)

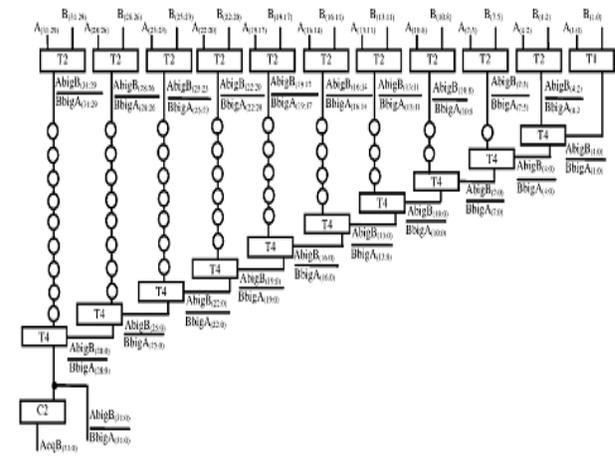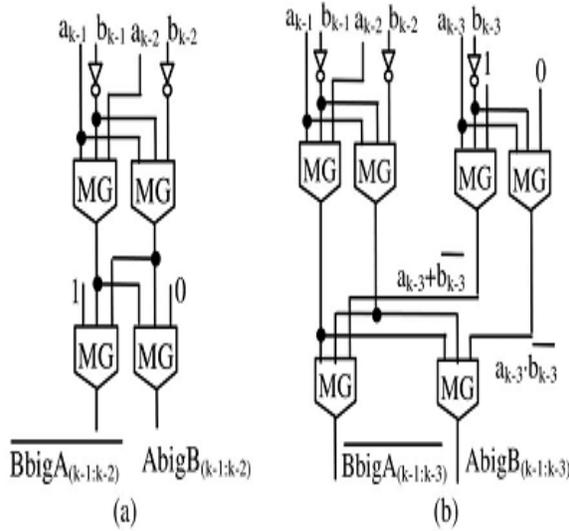Fig. 2. QCA modules: (a) T1; (b) T2; (c) T3; (d) T4; (e) C1; and (f) C2

Fig. 3. Novel 32-bit CB full comparator.

TABLE I

SPLITTINGCRITERIONADOPTED IN THECB COMPARATORS

| n	Splitting of the operands	
4	$A_{(3:2)}A_{(1:0)}$	$B_{(3:2)}B_{(1:0)}$
8	$A_{(7:5)}A_{(4:2)}A_{(1:0)}$	$B_{(7:5)}B_{(4:2)}B_{(1:0)}$
16	$A_{(15:14)}A_{(13:11)}A_{(10:8)}A_{(7:5)}A_{(4:2)}A_{(1:0)}$	
	$B_{(15:14)}B_{(13:11)}B_{(10:8)}B_{(7:5)}B_{(4:2)}B_{(1:0)}$	
32	$A_{(31:29)}A_{(28:26)}A_{(25:23)}A_{(22:20)}A_{(19:17)}A_{(16:14)}A_{(13:11)}A_{(10:8)}A_{(7:5)}A_{(4:2)}A_{(1:0)}$	
	$B_{(31:29)}B_{(28:26)}B_{(25:23)}B_{(22:20)}B_{(19:17)}B_{(16:14)}B_{(13:11)}B_{(10:8)}B_{(7:5)}B_{(4:2)}B_{(1:0)}$	

(c)   (d)

(e)   (f)

As it is well known, the number of cascaded MGs within the worst computational path of a QCA design directly affects the delay achieved. In fact, each MG introduces one clock phase in the overall delay. From Fig. 2, it can be seen that the modules T1 and T2 contribute to the computational path with one inverter and two MGs. Each instance of T4 introduces one more MG, whereas C2 is responsible for one MG and one inverter. As a consequence, the critical computational path of the novel n-bit CB full comparator consists of $n/3+3$ MGs and 2
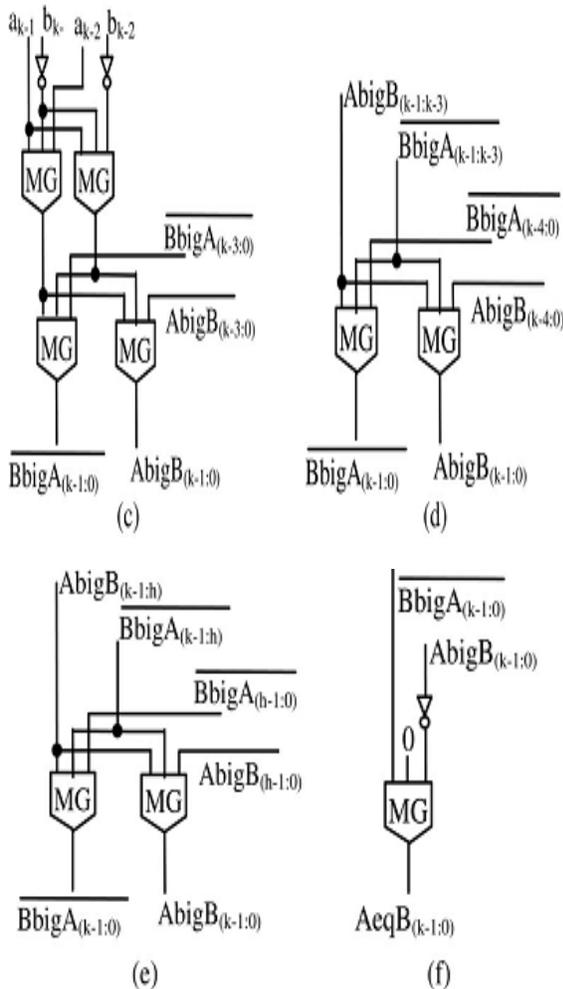
inverters. As an example, the 32-bit implementation depicted in Fig. 3 has the worst-case path made up of 13 MGs and 2 inverters. As always happens in CB computational architectures, the number of MGs within the computational path of the above described comparator linearly increases with n. An alternative solution presented here adopts TB architecture to achieve shorter computational paths. When this approach is exploited, several implementations of an n-bit full comparator can be designed differently combining the novel theorems and corollaries, as well as their QCA implementations depicted in Fig. 2. The TB comparators implement the comparison function recursively. The operands A and B are preliminarily partitioned as $A=A_{MSB}A_{LSB}$ and $B=B_{MSB}B_{LSB}$. The portions A_{MSB} and B_{MSB} are compared independently of the portions A_{LSB} and B_{LSB}. The depth of the recursion directly impacts the whole architecture. Examples of TB structures designed for 16- and 32-bit comparators are illustrated in Fig. 4. In Fig. 4(b) and (d), the recursion with its minimum depth is adopted. The portions A_{MSB} and B_{MSB}, as well as the portions A_{LSB} and B_{LSB}, are separately compared trough two independent CB architectures. The overall result is finally built with the modules C1 and C2. Fig. 4(a) and (c) shows comparators designed adopting deeper recursions.

In the following of the paper, the 16- and 32-bit TB implementations illustrated in Fig. 4(b) and (d) are deeply analyzed. Referring to the QCA modules depicted in Fig. 2, it can be easily verified that the former uses 35 MGs and 17 inverters and its critical computational path consists of 7MGs and 2 inverters, whereas the latter utilizes 83 MGs and 33 inverters and it has a worst-case path composed by 9 MGs and 2 inverters.

(a)

(b)
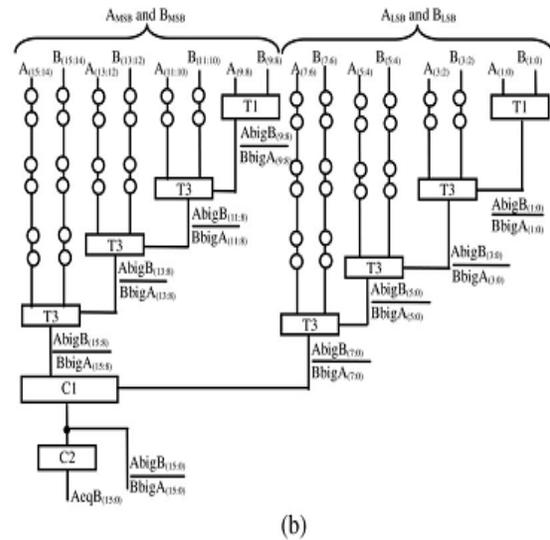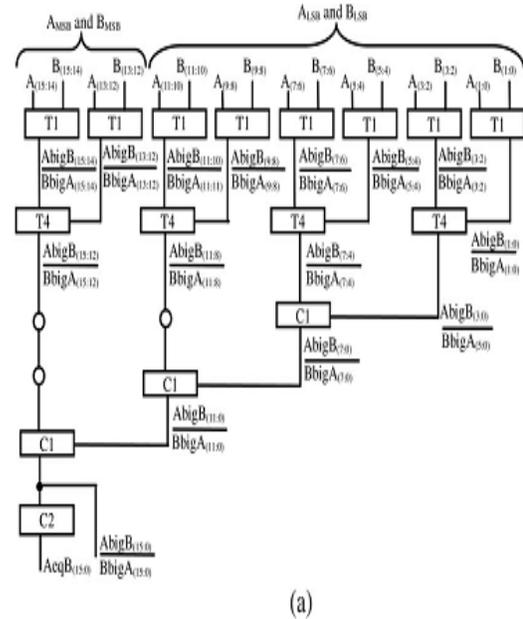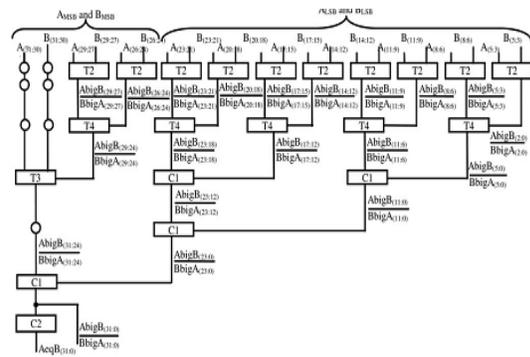
(c)
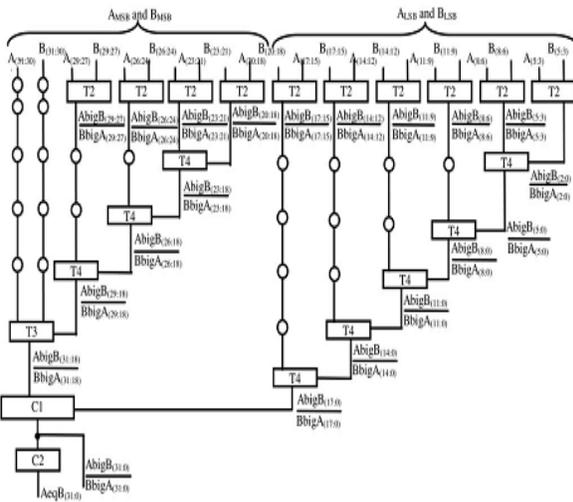
(d)

Fig. 4. Examples of novel TB comparators with: (a) and (b) 16-bit operands; (c) and (d) 32-bit inputs
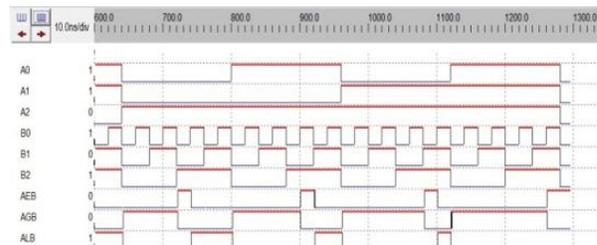
Simulation and Comparison

The proposed designs of reversible binary comparators have been functionally verified through simulations in DSCH. The simulation results show that the comparators give perfect output for all possible combinations of inputs. The simulation results for 1-bit, 2-bit, 3-bit and 4-bit reversible binary comparator are shown in Fig
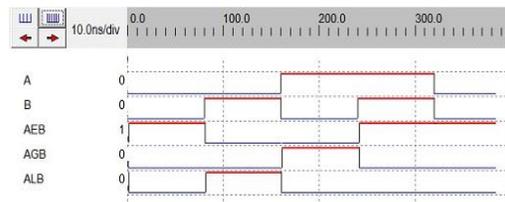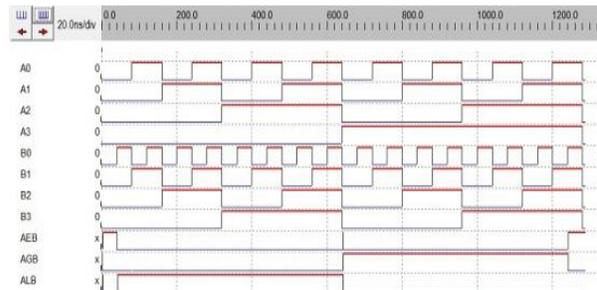

Simulation Result of Reversible 1-bit Comparator
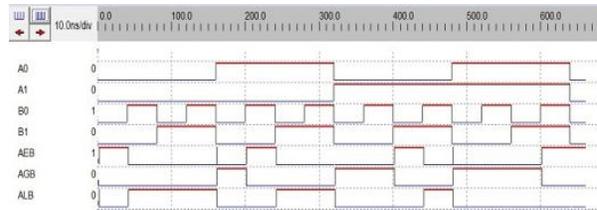
Simulation Result of Reversible 2-bit Comparator

Simulation Result of Reversible 3-bit Comparator

Simulation Result of Reversible 4-bit Comparator

Conclusion

This paper presents a systematic approach to design an n-bit signed comparator in reversible mode. With the help of comparisons and theorems it has been shown that the paper exhibits its efficiency over all the existing designs in terms of all the performance parameters. Quantum cost minimization is the strength of the

proposed architecture. Since comparison of two numbers can be useful in many operations inside the microprocessor, communication systems, encryption devices, sorting networks [11] and many more, it can be expected that the optimized low cost design will surely bring more efficiency and scalability in the world of reversible computing. Currently, studies are being performed to extend this design in a form of tree based architecture to bring more features of reversibility.

References

[1] Moore, Gordon E, (1975) "Progress in digital integrated electronics", Electron Devices Meeting, Vol. 21(1975), IEEE 11-13.

[2] Steane, Andrew, Feb(1998) "Quantum computing", Reports on Progress in Physics, Vol. 61. No. 2, pp117-173.DOI: http://dx.doi.org/10.1088/0034-4885/61/2/002

[3] Paun, Gheorghe, Grzegorz Rozenberg, and Arto Salomaa, (1998) "DNA computing: new computing paradigms", Springer-Verlag New York Inc. Secaucus,NJ. USA(1998), ISBN:3540641963.

[4] Rolf Landauer, (1961) "Irreversibility and heat generation in the computing process", IBM journal of research and development, Vol. 44. No. 1(1961), pp261-269.DOI: http://dx.doi.org/10.1147/rd.441.0261

[5] Patel, Ketan N., John P. Hayes, and Igor L. Markov, (2004)"Fault testing for reversible circuits",Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions, Vol. 23. No. 8 (2004),pp1220-1230.

[6] Tommaso Toffoli, (1980) " Reversible computing", MIT Lab for Computer Science. Seventh Colloquium Noordwijkerhout, The Netherlands. Vol. 85.Jul(1980), pp632–644. DOI: http://dx.doi.org/10.1007/3-540-10003-2_104

[7] Morita, Kenichi, (2001) "A simple universal logic element and cellular automata for reversible computing", Machines, Computations, and Universality, Springer Berlin Heidelberg, 2001, pp102-113.

[8] Voyiatzis, Ioannis, Dimitris Gizopoulos, and Antonis Paschalis, (2005)"Accumulator-based test generation for robust sequential fault testing in DSP cores in near-optimal time",Very Large Scale Integration (VLSI) Systems, IEEE Transactions, Vol 13.No. 9 (2005),pp1079-1086.

[9] Snider, G. L., et al,(1999) "Quantum-dot cellular automata: Review and recent experiments",Journal of Applied Physics, Vol. 85.No. 8 (1999),pp4283-4285.

[10] WALUS, Konrad, et al, (2004)"QCA Designer: A rapid design and simulation tool for quantum-dot cellular automata", IEEE transactions on nanotechnology, Vol. 3.No. 1 (2004), pp26-31.