Journal of Applied Sciences

ISSN 1812-5654

science alert

ANSInet
an open access publisher
http://ansinet.com

Journal of Applied Sciences 8 (10): 1841-1849, 2008
ISSN 1812-5654

A Prospect for Future Generation of Quantum Dots in Computers

[1]Mirmansour Ziabari, [1]Ahmad Mohades Kassai, [1]Shahin Enayati Maklavani and [2]Amirkoushyar Ziabari
[1]Electronic Research Center, Iran University of Science and Technology, Tehran, Iran
[2]Sharif University, Tehran, Iran

Abstract: In this study, first we described the different proposed design of QCAs and their functions. Then we described cell to cell interaction in the presence of external clocking voltage and finally we designed a 4-bit computer using a 4-bit processor which can be used as an 8-bit computer. This design study is based on this specification of 4-bit accumulator so that 8-bit data can be parallel processed in even and odd clock cycles. Our aim is to provide some evidences that quantum dot computers have the potential of increasing their word length and data capacity.

Key words: Quantum dot cellular automata, majority gate, cell to cell interaction, diagonal anti-voting, allyl group dots, alkyl bridge, hyper threading

INTRODUCTION

In recent years the mankind is observing a drastic improvement in technology and specification of computers. This improvement is estimated to be exponential in speed, memory value and size reduction, but it is believed that semiconductor devices and MOS technology is reaching to a limit and the further improvement needs a qualitative change. This change undoubtedly means shifting from electronic semiconductor field to quantum mechanical area which in turn is based on some kind of new cells, of coupled quantum dots. These are called Quantum dot Cellular Automata or in abbreviated QCA.

Further the future generation of computers which presumably will be fabricated using QCA will have many attractive features. Most important of them are higher speed, smaller size and ultra low power consumption. In these computers, different transistor switches will be substituted by QCA. It is obvious that utilization of these computers need some kind of clocking systems to control the coulomb interaction between neighboring cells.

This device can be assumed as a structured charge container in which the charge configuration can be encoded with digital data. A proposed configuration for this cell is shown in Fig. 1a. It can be seen that the cell is composed of four QDs which are fixed in the corners of a square. If this cell is charged with two extra electrons, it is obvious that the electrons will be located diagonally. This kind of location is due to Columbic repulsion forces which don't allow them to locate in other arrangements. An important area of theoretical and experimental research

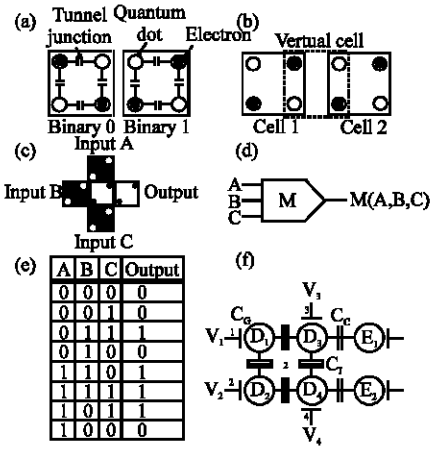

Fig. 1: (a) Basic QCA cell and two possible polarizations, (b) Interaction between two cells, (c) a QCA majority gate, (d) logical symbol of the majority gate, (e) truth table for the majority gate and (f) schematic of our QCA system

can be the cell-cell interaction. This interaction is shown in Fig. 1b, i.e., when the electron in one cell is located in a special diagonal (logic 1), this polarization induces the electrons in the neighbor QCA to be located with the same polarization. The actual switching gate is not just two simple QCAs. Actually it consists of 5 QCAs (Fig. 1c). And it can be shown by a three input logic gate (Fig. 1d). This kind of logic gate is named Majority Gate. It can be simply observed from Fig. 1c that with A = 0 the output will be B∧C and with A = 1 the output will be B∨C. The input A can be called programming input. These

Corresponding Author: Mirmansour Ziabari, Electronic Research Center, Iran University of Science and Technology (IUST), Tehran, Iran Tel: +981315556474 Fax: +981315548989

results in truth table form are tabulated in Fig. 1e. In recent years many research groups are trying to realize the actual QCA.

One of the latest reports showing realization of QCA logic network (Orlov *et al*., 2003). This report can be regarded as an extension of another report which is explained by Snider *et al*. (2001). Both these manuscripts and many other reports show that the only successful attempt to fabricate QCA logic network is employing aluminum tunnel junction technology which combines electron beam lithography with a suspended mask technique and *in situ* oxidation. Using this method, thin film aluminum dots separated by tunnel junctions are produced. So far this technique is preferred because of its good uniformity, but the charging energy (E_c) for aluminum dots is something around 1 meV which limits the operating temperature to 1 K. According to Orlov *et al*. (2003) over this temperature range the problem of superconductivity of aluminum appears and to resolve this problem the experiment should be performed in a magnetic field of less than of 1 Tesla in magnitude. For the aforementioned limitations as is reported by Kyosun *et al*. (2006), so far no general guidelines have been proposed and the previous design works have not been found to work always properly. It seems that fabricating a complete QCA design work is an extremely time-consuming process.

MATERIALS AND METHODS

A well known model for an idealized cell shown in Fig. 1f is introduced (Lent and Tougaw, 1997) and is as follows:

$$H^{cell} = \sum_{i,\sigma}(E_0 + V_i)\hat{n}_{i,\sigma} + \sum_{i>j,\sigma} t_{i,j}\left(\hat{a}^{\dagger}_{i,\sigma}\hat{a}_{j,\sigma} + \hat{a}^{\dagger}_{j,\sigma}\hat{a}_{i,\sigma}\right) + \sum_{i} E_Q\hat{n}_{i,\uparrow}\hat{n}_{i,\downarrow} + \sum_{i>j,\sigma,\sigma'} V_Q\frac{\hat{n}_{i,\sigma}\hat{n}_{j,\sigma'}}{|R_i - R_j|} \quad (1)$$

In this equation $\hat{a}_{i,\sigma}$ and $\hat{a}^{\dagger}_{i,\sigma}$ show annihilation and creation of electron in site i, respectively. The first term represents the onsite energy of each dot and V_i is the potential energy of an electron in dot i due to other charges. The second term accounts for electron tunneling between sites. The third term means the charging cost which is needed to put two electrons with opposite spins on the same dot and the last term is Columbic interaction between electrons on different sites within a cell.

To find the stationary states of the cell, we should solve the following Schrödinger equation:

$$\hat{H}^{cell}|\psi_i\rangle = E_i|\psi_i\rangle \quad (2)$$

In the above equation $|\Psi_i\rangle$ is the i'th eigenstate and E_i is the corresponding eigenvalue. Using the method of many-particle site-ket basis obviously the 16 different acceptable functions will be as follows:

$$|\phi_1\rangle = \begin{vmatrix}0&0&0&1\\0&0&0&1\end{vmatrix}\rangle, |\phi_2\rangle = \begin{vmatrix}0&0&0&1\\0&0&1&0\end{vmatrix}\rangle,$$
$$|\phi_3\rangle = \begin{vmatrix}0&0&0&1\\0&1&0&0\end{vmatrix}\rangle, |\phi_4\rangle = \begin{vmatrix}0&0&0&1\\1&0&0&0\end{vmatrix}\rangle, \quad (3)$$
$$\ldots \quad , |\phi_{16}\rangle = \begin{vmatrix}1&0&0&0\\1&0&0&0\end{vmatrix}\rangle.$$

In this notation each column represents the dot (D_1 to D_4 in Fig. 1f) and row represents the spin of electron in which upper row means upper spin and lower row means lower spin. It should be emphasized that Φ_is (for i = to 16) are 2 by 4 matrixes with just one 1 in each row. Then H_{ij} can be computed using the following equation:

$$H_{ij} = \langle\phi_i|\hat{H}|\phi_j\rangle \quad (4)$$

The set of H_{ij}s make the Hamiltonian matrix to be a (16×16) matrix and accordingly the ground state Ψ_0 is as follows:

$$|\psi_0\rangle = \sum_j \psi_j^0|\phi_j\rangle \quad (5)$$

In the condition of high tunnel barrier (weak tunneling) the electron in each site is in a more or less quantized condition and when the tunneling energy is much less than Columbic energy, the electrons remain largely localized and the cell will be like Fig. 1a.

RESULTS AND DISCUSSION

As mentioned in the previous section, the extra electrons in the condition of an isolated cell can have two energetically equivalent arrangement or two cell polarization P = +1 and P = -1 which are used to encode the binary information 1 and 0.

From the theoretical point of view, these polarizations are related to ground state eigenfunctions $|\Psi_0\rangle$ with the following equation;

$$P = \frac{(\rho_1 + \rho_3) - (\rho_2 + \rho_4)}{\rho_1 + \rho_2 + \rho_3 + \rho_4} \quad (6)$$

in which

$$\rho_i = \langle\psi_0|\hat{n}_i|\psi_0\rangle \quad (7)$$

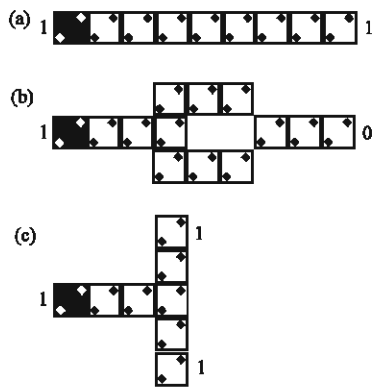

Fig. 2: Fundamental of QCA devices: (a) the binary wire, allows transmission of information from one point to another within the array, (b) the inverter uses diagonal anti-voting behavior to invert the signal and (c) fanout allows the result of a calculation to be propagated trough two or more other arrays

It is obvious that ρ_i can have values of 1 or 0, i.e., when in an isolated cell an electron is localized in site D_1, $\rho_1 = 1$ and as a consequence $\rho_3 = 1$, $\rho_2 = \rho_4 = 0$. Substituting these values in Eq. 1 gives P = +1 and for the other case, the value of P will be -1. Now considering the interaction between two cells, we suppose the cells are located like Fig. 1b. Obviously if we fix the polarization of cell 1 at P = +1, the Columbic effect inside the virtual cell forces the electrons to be localized as shown, and again the polarization of cell 2 using Eq. 6 will be P = +1. This result can be generalized for a row of QCA like Fig. 2a, i.e., fixing the first QCA in such a row in polarization P = +1 induces the same polarization in all the following QCAs. This arrangement is called binary wire, but changing the arrangement as Fig. 2b and fixing the first QCA in polarization P = +1 induces the next QCAs in such a way that right half will have the opposite polarization. Figure 2b can be supposed as a logic NOT or as inverter gate. Also arrangement of QCAs as Fig. 2c is widely used and is equivalent to fanout, which means the data is spread in different directions by Sayeeda *et al.* (2006).

A very important feature of QCA logic networks is that the binary wires can intersect each other. It means that we can conduct the information in a wire like Fig. 3a and b with correct positioning of QCAs. The correct information can be extracted from the data outways. If we expand these figures like Fig. 3c, using symmetry of Columbic effect, we can easily find that the same polarization will be transferred from cell 3 to cell 4. It means that data can cross another binary wire without causing any disturbance.

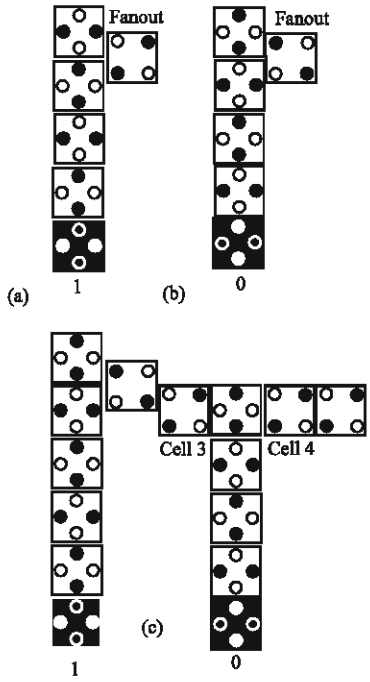

Fig. 3: (a) and (b) Fanout by correct positioning of cells and (c) transferring data through two crossing binary wires

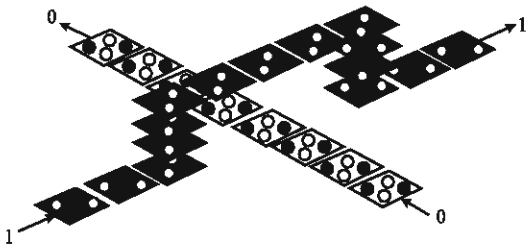

Fig. 4: Layout of multilayer wire crossing

Some recent researches propose that information wires transfer data in different layers (Fig. 4), i.e., the wires bridge over each other (Cho and Swartzlander, 2007a). However, no evidence was presented showing that such a design is practically possible.

In Fig. 1c and d, we have introduced a combination of 5 QCAs and their schematic representation, along with the explanation of different functions of such a QCA arrangement is presented in truth table in Fig. 1e. In Fig. 1d the inputs A, B and C are same as those which are written in truth table in Fig. 1e (Cho and Swartzlander, 2007b). This gate is actually fabricated and according to one of the latest reports (Cho and Swartzlander, 2007a), the cells consist of QDs each having 5 nm in diameter and the cell itself has 18 nm in both width and height and with a separation of 20 nm of centers. This kind of cells

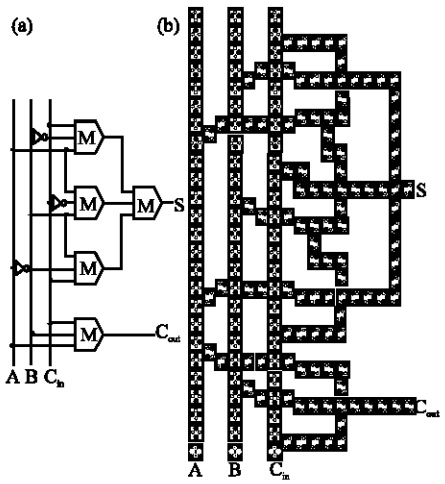

Fig. 5: (a) Schematic diagram of one-bit QCA full adder and (b) its layout

combination is called majority gate. Again Cho and Swartzlander (2007b), a more sophisticated arrangement of cells, called full adder has been proposed. The schematic diagram of this proposed logic network is shown in Fig. 5a and the actual layout of QCAs is shown in Fig. 5b.

The lower part of the layout is the QCAs arrangement which produces C_{out}. It is mentioned before that a majority gate has four minterms m_3, m_5, m_6, m_7. Using the Karnaugh map, we can easily find its Boolean function as follows:

$$M(A,B,C) = AB + AC + BC \tag{8}$$

Using the truth table of a full adder like Table 1a and with the help of Karnaugh map for C_{out} which is shown in Table 1b, we find that:

$$C_{out} = AB + AC_{in} + BC_{in} \tag{9}$$

It obviously shows that a single majority gate like the lower part of a full adder can produce C_{out}. With the same reasoning and using Karnaugh map in Table 1c, we find that:

$$S = ABC_{in} + A\bar{B}\bar{C}_{in} + \bar{A}B\bar{C}_{in} + \bar{A}.\bar{B}C_{in} \tag{10}$$

As reported by Wei *et al.* (2003) the upper part in Fig. 5a will produce the output S.

In the basic QCA, the concept of directionality is ambiguous, i.e., in a line of QCAs the information flow can happen in both directions. In the cases like shift registers in which we need the information flow happen only in one direction we have to use some kind of clocking system.

Table 1: (a) Truth table of the full adder, (b) Karnaugh map for C_{out}, (c) Karnaugh map for S

(a)

A	B	C_{in}	S	Count
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

(b)

A \ BC_{in}	00	10	11	01
0	0	0	1	0
1	0	1	1	1

(c)

A \ BC_{in}	00	10	11	01
0	0	1	0	1
1	1	0	1	0

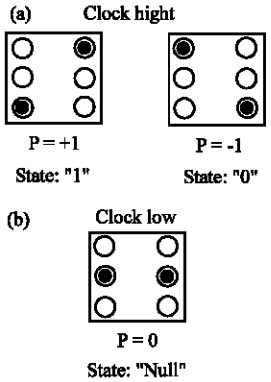

Fig. 6: Schematic of a clocked six-dot QCA cell

This kind of QCAs has recently been intensively studied and a schematic representation of such QCAs is shown in Fig. 6. As can be shown in Fig. 6, these kinds of QCAs consist of six dots and as in four dots QCA, two extra electrons. Pushing these electrons into the middle row of dots means setting the QCA in Null state and in this state QCA does not contain any bit information. By inserting an external repulsive potential, the charges move to the corners and the cell switches to a 0 or 1.

For this three-state QCA, a Hamiltonian can be constructed whose basis vector corresponds to the charge configuration of the three logic states. The basis vector consists of a null state (P = 0) and two completely polarized states (P = ±1). Considering the interaction between different QCAs or simply an array of QCAs, the Hamiltonian of the j'th cell in the array will be as follows (Timler and Lent, 2003):

$$\hat{H}_J = \begin{bmatrix} -\frac{E_k}{2}\sum_{m\neq J} f_{J,m}P_m & 0 & -\gamma \\ 0 & +\frac{E_k}{2}\sum_{m\neq J} f_{J,m}P_m & -\gamma \\ -\gamma & -\gamma & E_c \end{bmatrix} \tag{11}$$

Here γ represents the tunneling energy between one of the symmetric polarization states (Fig. 6a) and the null state (Fig. 6b). These tunneling energies can only happen in the upper right or lower left elements and the diagonal element does not contain these kinds of energy which means that the only tunneling path connecting them is through the null state. As reported by Lu *et al.* (2006) the numerical value of γ is estimated to be 0.15 and E_c is determined by the interaction of the cell's central dots with an external clocking field. The main property of this energy is variation with time and therefore it can be shown by $E_c(t)$.

The effects of this energy on a single cell are: null, active, locked and erased. When the clock is low (E_c is zero or negative), the electrons occupy the middle row (null state) and as mentioned before the cell contains no binary information. As the energy E_c increases from a certain limit E_k, the electrons are forced out of the central dots. But the polarization depends on the neighboring QCA. When the clocking energy $E_c(t)$ decreases, the cell will depolarize, i.e., any information stored, will be erased.

In the condition of geometrically symmetrical cell, the previous Hamiltonian will be simplified into the following matrix:

$$\hat{H}_j = \begin{bmatrix} E_0 - 0.18\varepsilon & 0 & -0.15 \\ 0 & E_0 + 0.18\varepsilon & -0.15 \\ -0.15 & -0.15 & E_0 + E_c \end{bmatrix} \quad (12)$$

As it is seen in the aforementioned matrix, the diagonal elements have an extra term E_0 which can be incorporated as energy of active dots or on-site energy.

This quantum mechanical system can be shown as Fig. 7a. It is obvious that the test cell Hamiltonian vector depends on external clocking field and the polarizations of left side cell (driver) and right cell (demon cell). We expect that aggregation of the polarizations of the two cells will result in the values of P. When we put a bit in the driver stage, it will be copied in the test cell and the demon cell relaxes to null state. This bit information will be erased later, but at the time of erasure the bit information will be copied in demon cell. After that the bit will be retained in the demon cell. Figure 7b shows the direction of energy flow in the cell, i.e., energy will enter the cell from the previous stage (driver) and E_{clock} and will flow out to the next stage (demon cell) or dissipates.

These kinds of QCAs are actually made using metal dot technique and some valuable experimental measurements have been made. These measurements showed that at ultimate single molecule devices which by estimation, in which the density will approach $10^{14}/cm^2$

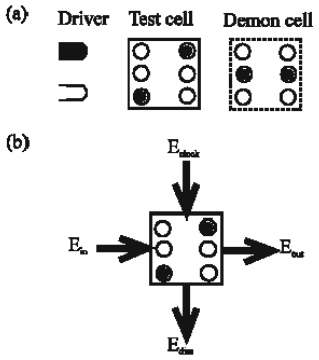

Fig. 7: (a) QCA array for examining logically reversible and irreversible erasure and (b) sign conventions for energy flow in a QCA cell

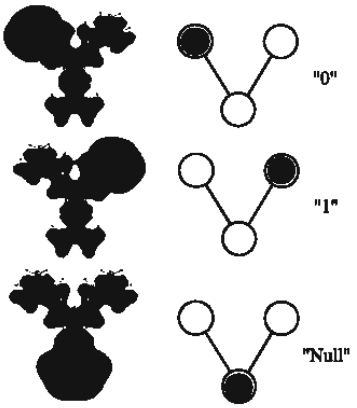

Fig. 8: Molecular structure of three charged configuration of allyl group dots. The calculated isopotential surfaces of the three states are shown on the left and schematic representations of the states are shown on the right

and with a clock frequency of 1000 GHz, the rate of energy absorption and dissipation will be so high that the chip will be melted.

A very clever alternative design is like Fig. 8 which is not like the previous metallic dot type. In opposite, they are allyl type, i.e., they are composed of alkyl bridges in a V shape. As can be seen in Fig. 8, the upper electron can belong to two upper holes of allyl molecules which alternatively represents logic 0 or 1. But when this electron drops to the lower hole allyl molecule, it represents a null state. It is obvious that for an isolated molecule 0 and 1 configuration have the same energy level and only the lower dot may have a different energy level. However the energy consumption will be less than that of the previous model, i.e., the two information bearing states model (Lent and Isaksen, 2003).

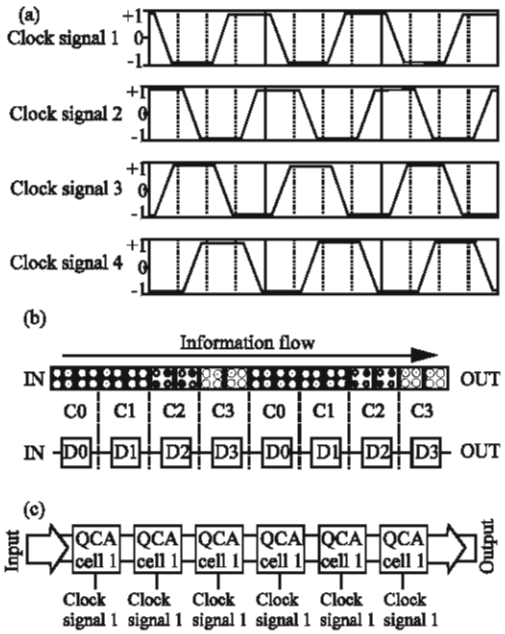

Fig. 9: (a) The four clock signals which order the propagation of data through the shift register, (b) QCA wire shown with cells and schematic representation. C0, C1, C2, C3 are the four phases of the clock. Each of the clocking zones maps to a latch in the circuit representation. Notice that only one clocking zone is latched, (c) Schematic of a QCA shift register with a four phase clocking schemes

As the clocking signal we expect it should be composed of four signals, each shifted 90° in phase relative to previous one (Fig. 9a). Of course we have a long way ahead to develop supporting system which can generate these signals and implement them to a row of QCAs. A QCA wire which can produce these different clocking signals is proposed like Fig. 9b. In this Fig. 4 QCAs are isolated and we can suppose them as a D latch. The clock signal implied to the clock input will propagate with a one quarter of cycle delay per QCA through the array. This is explained by Timler and Lent (2003), but in the same reference it is mentioned that so far a huge amount of work is done to fabricate the model systems and the simulation work like this study can be used by experimentalist groups as a feedback to correct their lines.

In Fig. 9c, the schematic diagram of a QCA shift register is shown. The actual representation of the shift register using previously introduced six dots QCAs demonstrated in Fig. 10. At the beginning it is supposed that the first row has arbitrary four bit information (1N00) and the driver stage is set in the polarization P = +1.

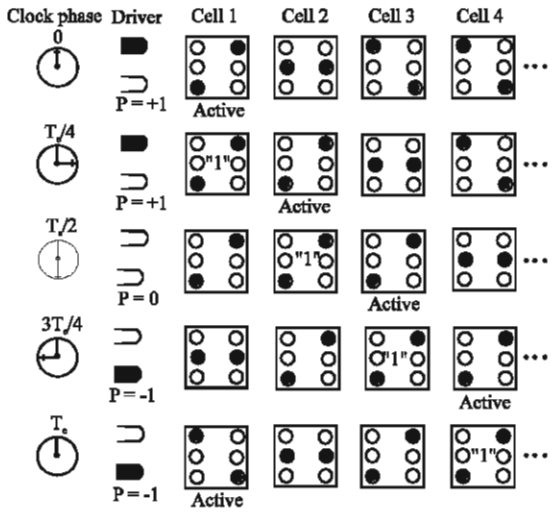

Fig. 10: Data propagation in a QCA shift register over one clock cycle

In the first quarter of clock cycle, the voltage $E_c(t)$ drops to -1 volt and as a consequence the first QCA will be set in locked state and will also push the second one in the P = +1 state. The reason can be outlined in one sentence as follows: The main specification of a six dot QCA, in every quarter period delay, is a locked cell polarizes its active downstream neighbor without any backstream from the following cell downstream.

Another important part of computer is memory, by which the microprocessor will be able to read and store information. This in turn will make the computer to transfer the stored data to Arithmetic Logic Unit (ALU) for further computations. In Fig. 10, we showed the layout of a shift register. Repeating eight registers like this figure and setting all the inputs in parallel and arranging the output in wire ored form of all the column outputs, we can have an eight word memory as Fig. 11. The main part of this memory is the left side column which plays the role of a 3-bit input decoder. Every two confronting wires act as a NAND gate and for this reason we proposed to invert the output of these NAND gates and repeating the same operation with next bit to make 3-bit input NAND gates. Obviously these NANDs together operate as a 3-bit input decoder, i.e., only one output will be in 1 state and all the others will be in 0 state. In other words, all the registers except one will be disabled. In this figure the output of all binary cells in one column is wire ored. It means that the small extension of vertical QCA wires are kept in 1 state and according to Fig. 1c the intersection of two QCA wires with C = 1 acts as an OR gate. But the eight outputs of decoder are so designed that their intersection with vertical QCA wires does not change the information content of these wires. According to Walus *et al*. (2005),

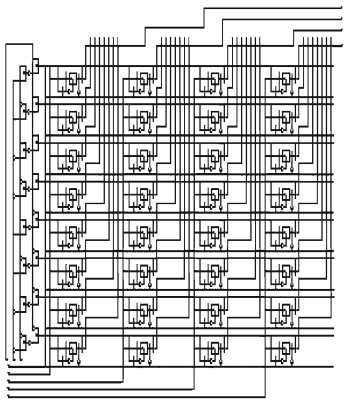

Fig. 11: The 8×4-bit memory block, as laid out with QCA

the area occupied by the memory is around 45 μm^2 and it takes 4 clock cycles for data to be stored and 5 clock cycles for data to appear in the output. Each write immediately followed by read operation takes 6 clock cycles together.

Accumulator unit should be designed such that its output data after passing through a feedback loop (normally one channel of ALU) can appear at its input. By drawing a pipeline diagram we can easily find that the total propagation time for one instruction to be done equals two clock cycles. This can be supposed as an advantage of this kind of accumulator because one can process two series of data at once. It means that accumulator in even clock cycle will hold data from the first series and in odd clock cycle will hold data from the second one, i.e., accumulator processes the two series in parallel. Recently this kind of time saving or speed increasing is nominated by hyper threading. It will be a very clever idea to use the 4-bit accumulator to support 8-bit data, i.e., the low nibble in even clock cycles and high nibble in odd clock cycles. Of course, some additional logic should be added to the system to do the above mentioned task and also care should be taken so that the memory areas of the two processes do not mix with each other.

It is well known that ALU programs have different arithmetic and logic operations. Normally, the main constituent of an ALU is a full adder. It is obvious that the data before entering the inputs of this full adder first should pass through some kind of modifying logic blocks. So that the two data inputs should be processed according to different arithmetic and logic instructions and then appear on the output of ALU. The most widely used logic blocks for this purpose is 4 multiplexers and the total logic networks will be as Fig. 12.

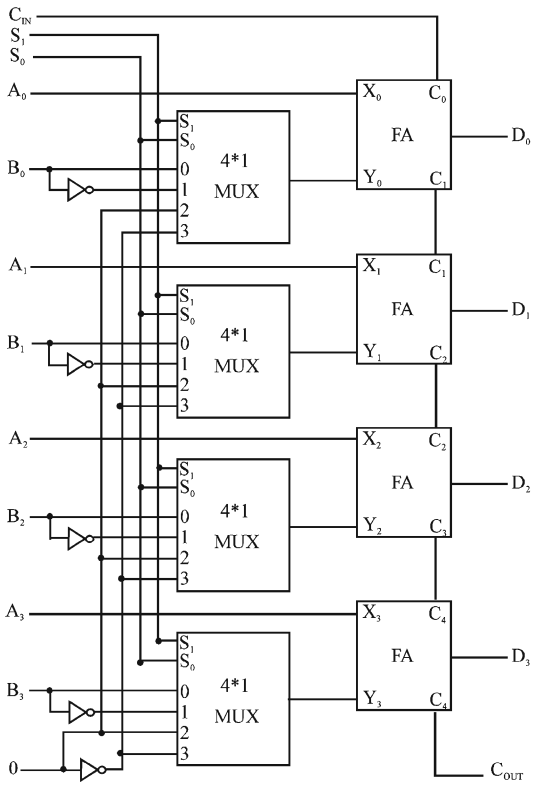

Fig. 12: A 4-bit logic network

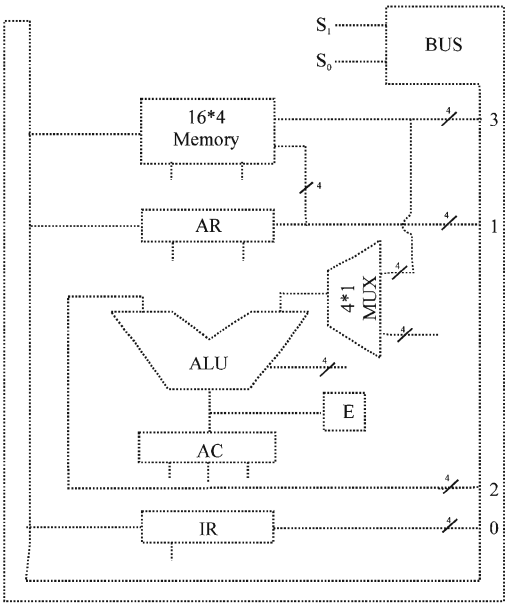

Fig. 13: A schematic diagram of architecture of the proposed computer system

Of course, this unit should have some Select inputs so that ALU can distinguish which one of the instructions

Table 2: The arithmetic operations which can be performed by the proposed ALU

S_2	S_1	S_0	C_{IN}	Operation	Instruction
0	0	0	0	ADD	F = A+B
0	0	0	1	ADD and DEC	F = A+B+1
0	0	1	0	SUB and DEC	F = A+B
0	0	1	1	SUB	F = A-B
0	1	0	0	TRANSFER	F = A
0	1	0	1	INC	F = A+1
0	1	1	0	DEC	F = A-1
0	1	1	1	TRANSFER	F = A+$\bar{0}$+1

Table 3: The logic operations which can be performed by the proposed ALU

S_2	S_1	S_0	C_{IN}	Operation	Instruction
1	0	0	×	AND	F = A∧B
1	0	1	×	OR	F = A∨B
1	1	0	×	XOR	F = A⊕B
1	1	1	×	COMP	F = $\bar{A}$

Table 4: Minimum microinstructions which, with the aid of them a complete program can be written and the proposed binary addresses and logic description of them

Instruction	Address	Description
AND	0× 8×	AC ← AC MEM(ADRS)
ADD	1× 9×	AC ← AC MEM(ADRS)
LDA	2× A×	AC ← MEM(ADRS)
STA	3× B×	MEM(ADRS) ← AC
BUN	4× C×	BUN ← MEM(ADRS)
BSA	5× D×	Branch and save address
ISZ	6× E×	Increment and skip if zero
CLA	7D	(Accumulator)ACC ← 0
CLE	7C	E ← 0
CMA	7B	ACC ← $\overline{ACC}$
CME	7A	E ← $\bar{E}$
CIR	79	Rotate right 1 bit
CIL	78	Rotate left 1 bit
INC	77	Increment
SPA	76	Skip if ACC positive
SNA	75	Skip if ACC negative
SZA	74	Skip if ACC zero
SZE	73	Skip if E zero
HLT	72	Half
INP	F8	ACC ← INP
OUT	F7	OUT ← ACC
SKI	F6	Skip if In flag ON
SKO	F5	Skip if Out flag ON
ION	F4	Interrupt On
IOF	F3	Interrupt Off

should be performed. We call these inputs C_{IN}, S_0, S_1 and S_2. In the case of S_2 = 0 only arithmetic operations are performed and when S_2 = 1, the logic operations will be performed. In Table 2, the different arithmetic operations are classified. Table 3 shows different logic operations which can be performed by the proposed ALU.

In Fig. 13 a schematic diagram of architecture of the proposed computer system is shown. The instruction word for this processor has 8 bits which in turn is loaded in the Instruction Register (IR). The first 4 bits are the op-code and the other 4 bits are the address. Like any other computer every mathematic operation should be performed during a set of instructions and as we mentioned before, accumulator can perform two sets of operations in even and odd clock cycles. Then we can use this architecture for 8-bit computer using 4-bit accumulator, i.e., when we want to add an 8-bit word with another 8-bit word and we know that the first one is written in memory numbers 0 and 1 and the second is written memory numbers 2 and 3. We can write the instructions so that the first low nibbles of the two quantities are added and memorized for example in memory No. 4 and then the high nibbles are added and memorized in memory No. 5.

Our design can be an elementary approach of systems which can support conditional instructions. For the sake of simplicity, in this design study, we have ignored the program counter and an external counter can do the same job. Anyhow each 4 binary bit combinations or op-code should be assigned to an equivalent instruction. Some of these instructions are classified in Table 4.

CONCLUSION

In this study, we have presented illustratively some kind of post transistor switches which are supposed as skeleton of future nanoscale computing systems. These switches which are called as QCAs are simple nano devices which can retain binary information. It was shown that with some modifications, they can be used as clocked controlled switches. In the next part of this study, we presented some design work for a 4-bit computer with a 4-bit processor and different logics such as register, memory and ALU. The major advantage of these design works is their capability of increasing word length by breaking every byte in two nibbles and processing them in parallel, in even and odd clock cycles. This study assumes that these design works are accurate reflection of final technology realization for the problem of word length and data increasing capacity of future computers, but we are sure that some changes may be required before final realization.

REFERENCES

Cho, H. and E.E. Swartzlander, 2007a. Serial parallel multiplier design in quantum-dot cellular automata. 18th IEEE Symposium on Computer Arithmetic (ARITH'07), 0-7695-2854-6/07.

Cho, H. and E.E. Swartzlander, 2007b. Adder designs and analyses for quantum-dot cellular automata. IEEE Trans. Nanotechnol., 6 (3): 374-383.

Kyosun, K., K. Wu and R. Karri, 2006. Quantum-dot cellular automata design guideline. IEICE. Trans. Fundam., E89-A (6): 1607-1614.

Lent, C.S. and P.D. Tougaw, 1997. A device architecture for computing with quantum dots. Proceedings of the IEEE, 85: 541-557.

Lent, C.S. and B. Isaksen, 2003. Clocked molecular quantum-dot cellular automata. IEEE. Trans. Elect. Devices, 50 (9): 1890-1896.

Lu, Y., M. Liu and C.S. Lent, 2006. Molecular electronics-from structure to circuit dynamics. IEEE, 1-4244-0078-3/06.

Orlov, A.O., R. Kummamuru, R. Ramasubramaniam, C.S. Lent, G.H. Bernstein and G.L. Snider, 2003. Surface Science. Elsevier, pp: 1193-1198.

Sayeeda, S., S. Al Imam and K. Radecka, 2006. Testing QCA Modular Logic, IEEE. 1-4244-0395-2/06, pp: 700-703.

Snider, G.L., A.O. Orlev, R.K. Kummamuru, R. Ramasubramaniam, I. Amlani, G.H. Bemstein, C.S. Lent, J.L. Merz and W. Porod, 2001. Quantum-dot Cellular Automata: Introduction and Experimental Overview. IEEE-NANO, pp: 465-470.

Timler, J. and C.S. Lent, 2003. Maxwell's demon and quantum-dot cellular automata. J. Applied Phys., 4 (2): 1050-1060.

Walus, K., M. Mazur, G. Schulhof and G.A. Jullien, 2005. Simple 4-bit processor based on quantum-dot cellular automata (QCA). Proceedings of the 16th International Conference on Application-Specific Systems, Architecture and Processors (ASAP'05), IEEE, 1063-6862/05, pp: 1-6.

Wei, W., K. Walus and G.A. Jullien, 2003. Quantum-dot cellular automata adders. 3rd IEEE Conference on Nanotechnology, 0-7803-7976-4/03, pp: 461-464.