Journal of Applied Sciences

ISSN 1812-5654

science alert

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

Journal of Applied Sciences 8 (14): 2569-2576, 2008
ISSN 1812-5654
© 2008 Asian Network for Scientific Information

Designing a Hamming Coder/Decoder Using QCAs

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

Abstract: In this research, the arithmetic structures of Hamming coder/decoder are discussed and a general formula for the case of a single bit error in composite dataword parity bits is derived and using this formula a design work for logic circuitry which functions as Hamming coder/decoder and exclusively is composed of XOR-gates is presented. The advantages of unique capabilities of Quantum dot Cellular Automata (QCA) are stated. Then many design consideration are described and showed that how NAND/NOR operations routing mechanism can be implemented using QCAs. Substituting these blocks with XORs leads to a design work which is completely composed of NAND gates and in turn of QCAs. For the reason of complexity of the final design work, the advantage of an advanced software tool i.e., the QCADesigner for final implementation of the QCA network is taken. To minimize total delay and chip area and in turn total manufacturing cost the design work is so modified that the number of clocking zone between input and output to be minimum and this can be the reason of superiority of the proposed design work.

Key words: Clocking zone, QCA routing element, QCA chip, single bit error

INTRODUCTION

In digital communications and specifically in interplanetary message transmission, there is always the problem of contamination of a single pulse by noise and how to detect the pulse in the noisy conditions i.e., the reconstruction of pulse from noisy samples. The most common method to prevent the problem of false pulse detection is to increase signal to noise ratio in destination, but it can be obviously achieved by increasing crucially the amount of power in the transmitter. In some cases like sending messages by astronauts, however it is impossible as the systems S/N ratio can not be increased and consequently error rate is unacceptably high. Then some other means of improving reliability such as error control coding must be sought. In the aforementioned case in which request of error correction or retransmission is impossible, the error control must take the form of Forward Error Correction (FEC) or sending error correction code.

This kind of correction code normally consists of an n bit binary word and k parity bits which are added to the n-bit word to form a new n+k bits word. Of course, the word can have any length but the eight bit version should have k = 4 bits. At the receive end, the received dataword-parity bits should be checked. The process of checking is similar to parity bit generation i.e., the same combination of bits including the parity bit should be checked.

In this manuscript it has been shown that all the parity and check bits can be generated using XOR-gates and also there is a four NAND-gates network which is equivalent to an XOR-gate. One important capability of the QCA technology is the implementation of NAND/NOR gates. As the final design was composed only of XOR-gates, then a four NAND gate block which functions as an XOR-gate is introduced. Substituting these blocks with XOR gates leads to a design work which is completely composed of NAND gates and hence of QCAs. An important challenge with QCA design is latching problem. It can be easily proved that total delay is directly proportional to the maximum of clocking zones between input and output and the number of gates. So $3\times2 = 6$ zone routing elements are used to minimize the total delay and chip area and hence total manufacturing cost.

MATERIALS AND METHODS

Quantum dot cellular automata, which was proposed by Lent *et al.* (1993a, b) in can be a convenient substitution for silicon/CMOS switches in near future. This promising pattern, which was actually fabricated in 1997 (Lent and Tougaw, 1997), basically consists of four charge containers or dots which should be located in the corners of a square as shown schematically in Fig. 1. As can be seen, QCA can be supposed as a novel nano

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

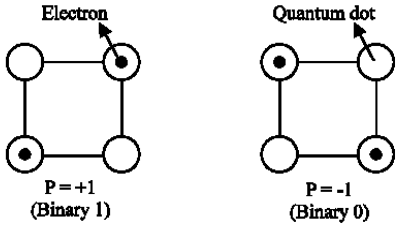


Fig. 1: Basic QCA cell and two possible polarizations states

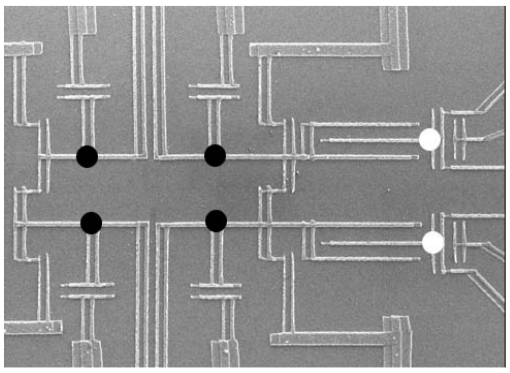


Fig. 2: Scanning electron micrograph of a four-dot cell

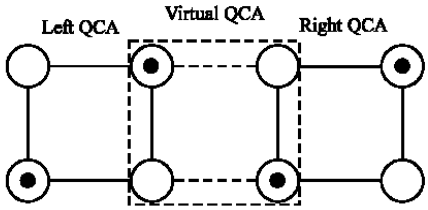


Fig. 3: Interaction between two QCAs

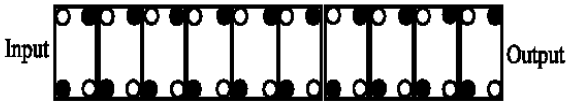


Fig. 4: QCA wire

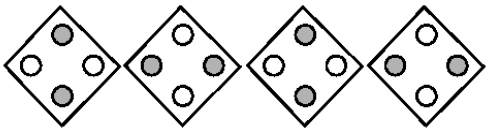


Fig. 5: Binary wire constructed from 45 degree angle oriented cells

device which stores binary information in the form of position of individual electrons (Lieberman *et al*., 2002).

After the fabrication report, many improvements have been reported in the field of QCA realization in the form of metal dot individual devices or QCA with chemical molecules (Lent, 2000). The metallic type which is called metallic QCA cell is actually four Al nano scale island with metallic tunnel junctions. A scanning electron micrograph of an actual prototype of this cell is shown in Fig. 2. In Fig. 2, the black dots indicate the position of four metal islands and white dots represent single electron transistor electrometers which are used to measure the output (Mo, 2006). So far, different kind of logic gates, latches and clocked devices have been designed and realized with the aid of QCA individual devices (Demadis *et al*., 2001).

Having a careful look in localization of electrons in the QCA will reveal that two neighboring QCAs like Fig. 3 will interact with each others. It means that setting the left QCA in position 1 forces the right QCA through Columbic repulsion and absorption forces to the same position. This criterion i.e., controlling the position of a single electron in a QCA by the position of another single electron in a neighboring QCA has been practically demonstrated in the University of Notre dame (Amlani *et al*., 2000).

Extending the above mentioned interacting property to one row of QCAs like Fig. 4 shows that Coulombic forces cause all the cells to adopt the same polarity. This configuration is called a QCA wire (Orlov *et al*., 1999).

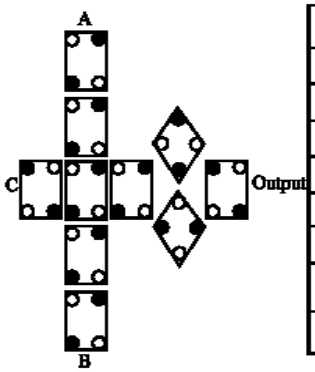
It is important to note that unlike CMOS technology no current flows during this process, in other words QCA relies on quantum configuration rather than voltage level to indicate states and Columbic interaction rather than current flow for information propagation.

Binary wire can also be constructed with dots in each cell oriented at a 45 degree angle from the standard cell (Fig. 5).

This allows binary wires to cross in the same plane or layer without interacting with each other. This configuration causes an inversion inside each cell. Care must be taken to use the correct number of cells to prevent accidental signal inversion (Mary, 2006).

Evidently, the next logic structure after QCA wire is QCA NAND-gate. The most practical design for NAND gates was proposed by Michael *et al*. (2003), which is a majority gate with three inputs A, B and C. This three-input gate with its truth table is shown in Fig. 6. C is control input and it can be easily shown that when C = 0 the output equals A^B. It is obvious that when C = 0, the logic element is equivalent to a NAND gate and when C = 1 it operates as a NOR gate.

The next step in design of a logic network is to find some kind of routing element which is mainly used to conduct the information flow and splitting. For example in any programmable device there are vertical and horizontal



C	A	B	Output
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Fig. 6: A logic element equivalent to NAND/NOR gate and its truth table

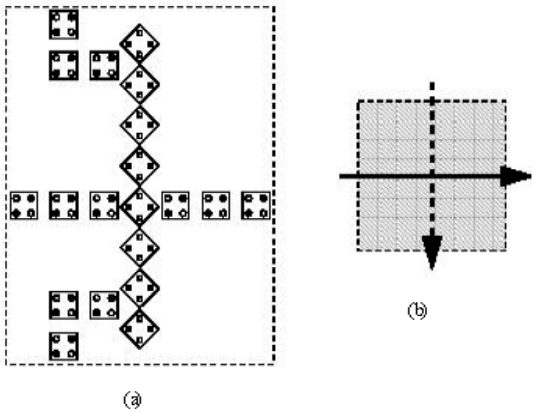


Fig. 7: (a) Crossing of two signals and (b) Ultimately ineffective as a routing element

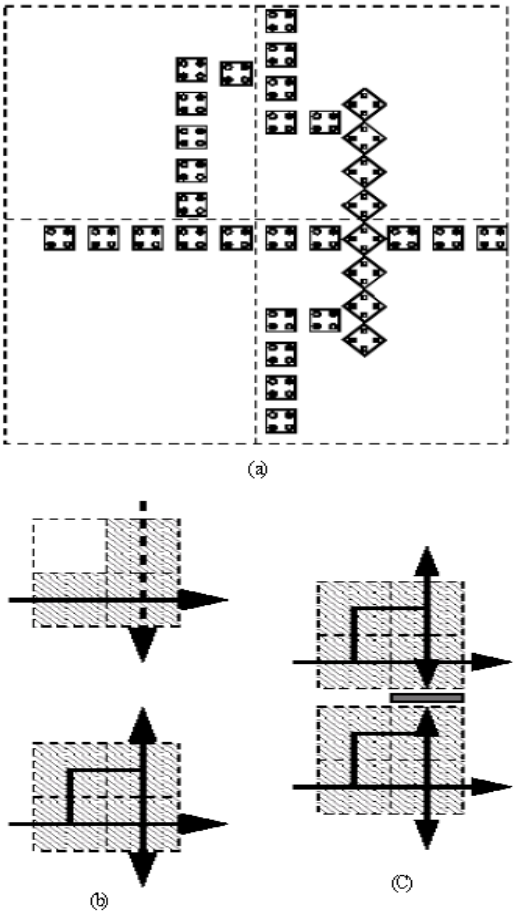


Fig. 8: (a) A 2×2 zone routing element (upper b) Signal flow when extra wire is unblocked, (lower b) Signal propagate in both directions and (c) Signals can interfere with adjacent routing element

lines which should be interconnected according to a specified programming pattern. In a normal ROM, these interconnections are shown by cross signs which show the existence of intact fuses between lines. Overlapping lines without cross signs means no interconnection which shows existence of blown fuses between lines. These interconnections provide paths for information flow and using them is inevitable in all programmable devices like ROM, PROM, EPROM, EEPROM, PLA and PAL. This kind of routing elements also was proposed in reference (Michael *et al.*, 2003) which is shown in Fig. 7. The simplest form of a routing element is a 2-by-1 multiplexer which can be built from three majority gates. By generalizing, it will result to designing a large scale any by any multiplexer. But the number of needed QCAs increases exponentially e.g., a 4×4 MUX can be built from circa 1100 QCAs (Michael *et al.*, 2003). For this reason a large amount of research work has been done to find a more efficient routing mechanism. Figure 7 consists of two QCAs wires. One composed of cells oriented at 90 degrees and the other from cells oriented at 45 degrees. It is obvious that the two wires can cross each other without any interference.

As can be seen in the left side block of Fig. 7 a group of clocked QCAs can be held in four different phases i.e., relax, release, hold and switch. When a QCA is set in the switch phase it means that it is influenced by the adjacent QCA which is previously held in the hold phase or a QCA with initial polarization of release phase. But if a QCA is set in relax phase it means that it acts as a buffer i.e., this phase removes a group of QCA cells and interconnect the data link of two adjacent QCAs (Michael *et al.*, 2003; Mary, 2006; Walus *et al.*, 2003). In Fig. 7 there is no mechanism by which the signals from 45 degree wire can jump to the 90 degree wire i.e., the signal in this configuration is not able to turn corner. To solve this problem another configuration is proposed in which

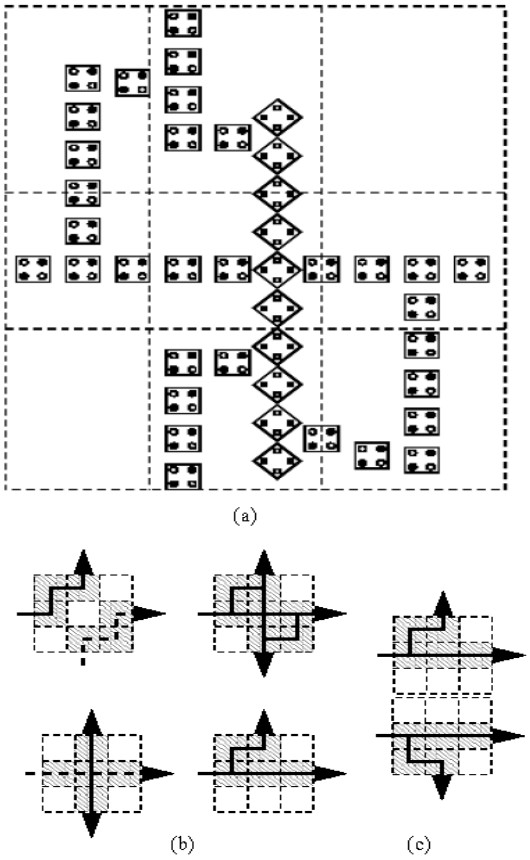


Fig. 9: (a) QCA routing element with 3×3 zones, (b-top left) data propagate in two insulated routes, (b-top right) routing element acts as fan out, (b-bottom left) signal propagate in two perpendicular QCA lines, (b-bottom right) routing element acts as buffer and (c) provide protections against interference

signals are carried in two perpendicular 45 and 90 degree wires, but an additional wire connects them. This configuration is shown in Fig. 8.

If this extra wire is left in the relax phase, signals will only propagate in a straight forward manner. But if this new region is clocked, a signal can propagate from one of the wires to others i.e., the signal will fan out. The drawback of this configuration is that the signal from one wire will propagate along inputs of all adjacent blocks.

One way to modify the system so that the routing is controllable is to expand the aforementioned 2×2 element to 3×3 clock region element as in Fig. 8 and 9. It can be seen that this configuration needs 9/4 = 2.25 times more area which is the main drawback of this design.

Having considered the too much area occupation, guided the scientists to select a compromise design i.e., a 3×2 routing element like Fig. 10.

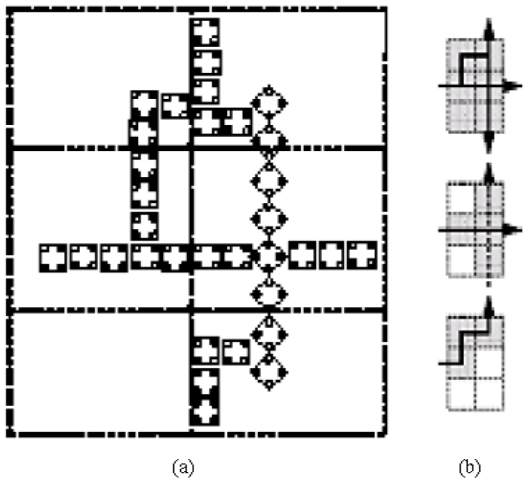


Fig. 10: (a) QCA routing element with 3×2 zones, (b-top) routing element acts as fan out, (b-middle) signal propagate in two perpendicular QCA lines, (b-bottom) signal turns corner upward and is buffered from the down side

As mentioned earlier, our final aim is to design a programmable device which operates as a Hamming coder/decoder.

For the case of n = 8 and k = 4 the total word length is 12. The arrangement of these bits is as Table 1. Four parity-bits are designated by p_1, p_2, p_4 and p_8 which are positioned in location numbers 1, 2, 4 and 8, respectively. The remaining bits are shown by b_0 to b_7. It can be seen that these 8 bits which are the bits of our primary word are positioned in the location numbers 3, 5, 6, 7, 9, 10, 11 and 12, respectively. The parity bits can be calculated as follows:

$$\begin{aligned} p_1 &= \text{XOR of bits}(3,5,7,9,11) \\ p_2 &= \text{XOR of bits}(3,6,7,10,11) \\ p_4 &= \text{XOR of bits}(5,6,7,12) \\ p_8 &= \text{XOR of bits}(9,10,11,12) \end{aligned} \tag{1}$$

It is obvious that p_1, p_2, p_4 and p_8 can be realized with the logic networks shown in Fig. 11a-d.

The four parity bits p_1 to p_8 which are calculated according to aforementioned operations should be located in the positions 1, 2, 4 and 8 as shown in Table 1 and the 12-bit composition word can be sent through communication operations. An interesting feature of these codes is their even parity. It means that every 12-bit composite word always contain an even number of ones. When the 12 bits are received by the receiver, they are checked again for possible errors. The four check bit are evaluated as follows:

Table 1: The positions of parity bits and data bits in a composition 12-bit word

Bit position	1	2	3	4	5	6	7	8	9	10	11	12
Bits	p_1	p_2	b_0	p_4	b_1	b_2	b_3	p_8	b_4	b_5	b_6	b_7

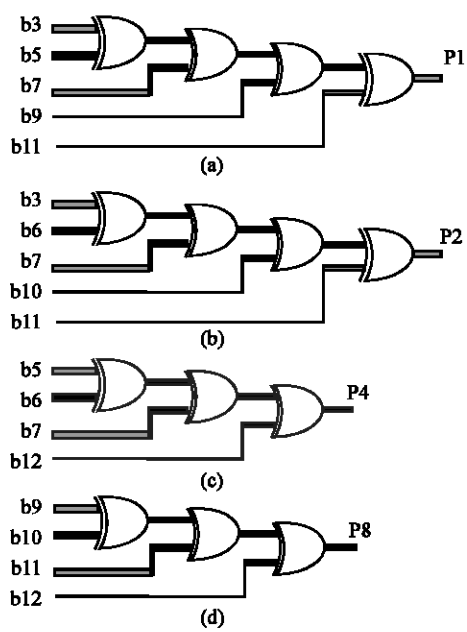


Fig. 11: The exclusive OR circuit generating parity bits p_1, p_2, p_4 and p_8

$$\begin{aligned} C_1 &= \text{XOR of bits } (1, 3, 5, 7, 9, 11) \\ C_2 &= \text{XOR of bits } (2, 3, 6, 7, 10, 11) \\ C_4 &= \text{XOR of bits } (4, 5, 6, 7, 12) \\ C_8 &= \text{XOR of bits } (8, 9, 10, 11, 12) \end{aligned} \tag{2}$$

The result of above mentioned parity check bits is combined to form a four-bit word in the form of C_8 C_4 C_2 C_1. Slepian showed that the binary content of this four-bit word is equal the position number of possible occurrences of a single bit error. By generalizing the above mentioned equations and assuming our primary binary n-bit word to be $b_0, b_1, \ldots, b_{n-1}$, k parity bits $p_1, p_2, \ldots, p_{2^{k-1}}$ can be defined and composing them together will result in a composition n+k bit word. In the case of k an integer and $n = 2^k$, k is equal to $(1+\log_2^n)$. But in the other cases the formula for k should be modified as follows:

$$k = 2 + [\log_2^n] \tag{3}$$

With these assumptions it would be convenient if we localized the n+k bit word in a position table like Table 2 with position numbers 1,2,3…,n+k. The check bits C_{2k-1} can be derived by the Eq. 4:

$$C_{2k-1} = \text{XOR}\left(\text{Set}(\text{Set}(BP_{(2m+1)2^k+s}) \mid {}^{2^{k-1}}_{s=0}{}^{-1}\right) \mid {}^{M}_{m=0} \tag{4}$$

In Eq. 4 the terms $BP_1, BP_2, \ldots BP_{n+k}$ are the content of columns 1,2,…,n+k of Table 2, respectively. The set(BP) function is defined as follows:

$$\text{Set}(BP_i)\Big|_{i=1}^{M} \equiv (BP_1, BP_2, \ldots, BP_M) \tag{5}$$

As a numerical example, let n = 12, therefore $k = 2 + [\log_2^{12}] = 5$. It means the number of parity bits is 5 which can be shown by $p_1, p_2, p_4, p_8, p_{16}$ and they are arranged together with 12-bit word in Table 3.

Now formulas (3), (4), (5) and Table 3 can be used to estimate the logic relations for C_1, C_2, C_3 C_4, C_8, C_{16}, putting k = 1 in formula 4. C_1 can be estimated as follows:

$$C_{2^{k-1}} = C_1 = \text{XOR}\left(\text{Set}\left(\text{Set}(BP_{(2m+1)+S})\Big|_{S=0}^{S=0}\right)\Big|_{S=0}^{S=0}\right) \tag{6}$$

The inner parenthesis consists of only one sentence BP_{2m+1} and by increasing m from zero to M produces different sentences of outer parenthesis which equals:

$$C_1 = \text{XOR}(BP_1, BP_3, BP_5, \ldots, BP_{M=2m+1}) \tag{7}$$

In the above mentioned equation, M is the largest allowed number less than or equal n+k which in this special case is 17. With the same procedure C_2, C_4, C_8, C_{16} are derived through following relations:

$$C_2 = \text{XOR}\begin{pmatrix} BP_2, BP_3, BP_6, BP_7, \\ BP_{10}, BP_{11}, \ldots, BP_{4m+2}, BP_{M=4m+3} \end{pmatrix} \tag{8}$$

$$C_4 = \text{XOR}\begin{pmatrix} BP_4, BP_5, BP_6, BP_7, \\ BP_{12}, BP_{13}, BP_{14}, BP_{15} \end{pmatrix} \tag{9}$$

$$C_8 = \text{XOR}\begin{pmatrix} BP_8, BP_9, BP_{10}, BP_{11}, \\ BP_{12}, BP_{13}, BP_{14}, BP_{15} \end{pmatrix} \tag{10}$$

$$C_{16} = \text{XOR}(BP_{16}, BP_{17}) \tag{11}$$

The next step which we normally should be taken before the above mentioned computation is to estimate p_1, p_2, p_4, p_8, p_{16}. There is no need to derive independent formulas. However the first sentence of the functions C_1, C_2, C_4, C_8, C_{16}, can be taken away i.e., p_1, p_2, p_4, p_8, p_{16} will be as follows:

Table 2: The positions of parity bits and data bits in a composition n + k bit word

Bit position	1	2	3	4	5	6	7	8	…	n + k
Bits	p_1	p_2	b_0	p_4	b_1	b_2	b_3	p_8	…	Bp_{n+k}

Table 3: The positions of parity bits and data bits in a composition n + k = 17 bit word

Bit position	1	2	3	4	5	6	7	8	…	16	17
Bits	p_1	p_2	b_0	p_4	b_1	b_2	b_3	p_8	…	p_{16}	$BP_{17} = b_{11}$

Table 4: The positions of parity bits and data bits in a composition n + k = 17 bit word

Bit position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Bits	1	0	1	0	0	1	1	0	0	1	1	1	0	0	1	1	1

Table 5: The positions of parity bits and data bits in a composition n + k = 17 bit word

Bit position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Bits	1	0	1	0	0	1	1	0	0	1	1	1	1	0	1	1	1

$$p_1 = XOR(BP_3, BP_5, \ldots, BP_{15}, BP_{17}) \quad (12)$$

$$p_2 = XOR(BP_3, BP_6, BP_7, BP_{10}, BP_{11}, BP_{14}, BP_{15}) \quad (13)$$

$$p_4 = XOR(BP_5, BP_6, BP_7, BP_{12}, BP_{13}, BP_{14}, BP_{15}) \quad (14)$$

$$p_8 = XOR(BP_9, BP_{10}, BP_{11}, BP_{12}, BP_{13} BP_{14}, BP_{15}) \quad (15)$$

and

$$p_{16} = XOR(BP_{17}) = BP_{17} \quad (16)$$

As a final numerical example let the 12-bit binary word to be 101101110011. Inserting the bits in formulas (13) through (17), will result in the following parity bits:

$$p_1 = 1, \quad p_2 = 0, \quad p_4 = 0, \quad p_8 = 0, \quad p_{16} = 1$$

Which are together primary 12-bit word tabulated in Table 4.

As mentioned earlier, in the case of occurrence of a single bit error e.g. in bit position No. 13, the composite 17-bit word will be as Table 5 and C_1 through C_{16} will be computed as follows:

$$C_1 = 1, \quad C_2 = 0, \quad C_4 = 1, \quad C_8 = 1, \quad C_{16} = 0$$

The result is $C = C_{16}C_8C_4C_2C_1 = 01101$.

It is obvious that the binary content of five bit binary number formed by check bits is equal 13 and gives the position of the erroneous bit.

RESULTS AND DISCUSSION

So far, the words with arbitrary length and the needed parity bits are discussed. It was proved that both the parity bits generator and check bit decoder can be synthesized using XOR gates.

As in interplanetary communication, the actual word length does not exceed 16, the design work is proposed to be implemented for this word size. However, this implementation can be used as well for word length less than 16, i.e., it is enough to put zero for the extra bits.

For the case in which n = 16 and k = 5, the parity bits generation logic functions can be written as follow:

$$p_1 = XOR(BP_3, BP_5, \ldots, BP_{19}, BP_{21}) \quad (17)$$

$$p_2 = XOR(BP_3, BP_6, BP_7, \ldots, BP_{18}, BP_{19}) \quad (18)$$

$$p_4 = XOR\begin{pmatrix} BP_5, BP_6, BP_7, \ldots, BP_{12}, BP_{13}, \\ BP_{14}, BP_{15}, \ldots, BP_{20}, BP_{21} \end{pmatrix} \quad (19)$$

$$p_8 = XOR\begin{pmatrix} BP_9, BP_{10}, BP_{11}, BP_{12}, \\ BP_{13} BP_{14}, BP_{15} \end{pmatrix} \quad (20)$$

$$p_{16} = XOR(BP_{17}, BP_{18}, BP_{19}, BP_{20}, BP_{21}) \quad (21)$$

This logic functions can be implemented also for shorter word lengths e.g. data word with 12 bits. To do so, it is just enough to put zero for $BP_{18}, BP_{19}, BP_{20}, BP_{21}$.

The actual network to generate these parity bits is shown in Fig. 13. The composite word consisting of above parity bits and data word normally is sent through communication links and sometimes it may be received with a single bit error. In this case, the error can be corrected using parity check bit logic functions $C_1, C_2, C_4, C_8, C_{16}$. As mentioned before these functions are similar to $p_1, p_2, p_4, p_8, p_{16}$ except for in the first term. Then $C_1, C_2, C_4, C_8, C_{16}$ can be written as follows:

$$C_1 = XOR(BP_1, BP_3, BP_5, \ldots, BP_{19}, BP_{21}) \quad (22)$$

$$C_2 = XOR(BP_2, BP_3, BP_6, BP_7, \ldots, BP_{18}, BP_{19}) \quad (23)$$

$$C_4 = XOR\begin{pmatrix} BP_4, BP_5, BP_6, BP_7, \ldots, BP_{12}, BP_{13}, \\ BP_{14}, BP_{15}, \ldots, BP_{20}, BP_{21} \end{pmatrix} \quad (24)$$

$$C_8 = XOR\begin{pmatrix} BP_8, BP_9, BP_{10}, BP_{11}, \\ BP_{12}, BP_{13} BP_{14}, BP_{15} \end{pmatrix} \quad (25)$$

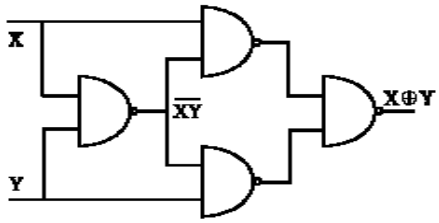


Fig. 12: Exclusive-OR constructed from NAND gates

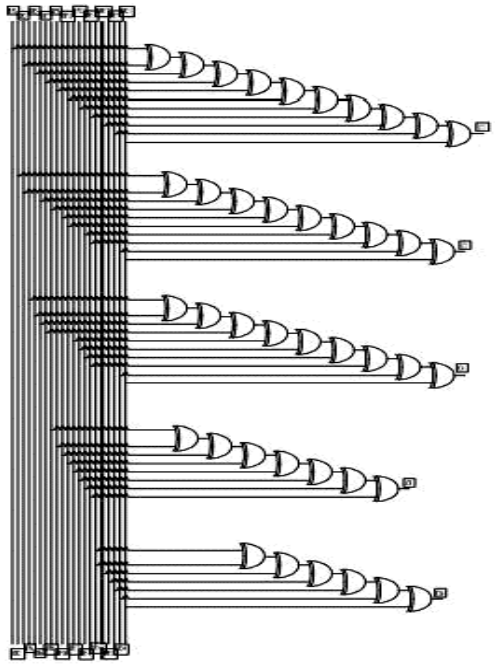


Fig. 13: The proposed design work for Hamming parity bits generator for n = 12 and k = 5

$$C_{16} = XOR(BP_{16}, BP_{17}, BP_{18}, BP_{19}, BP_{20}, BP_{21}) \quad (26)$$

In the above discussion, it was obviously shown that both parity generating bits and parity check bits were generated using exclusive-OR gates. But it can be proved using Boolean algebra that any digital network can be implemented with NAND gates and with some simple manipulation it can be shown that the following logic diagram is equivalent to an exclusive-OR gate (Fig. 12).

At this point, each XOR-gate in Fig. 13. should be replaced by its equivalent from Fig. 12. To construct a total QCA network each NAND-gate once more should be replaced by its equivalent from Fig. 6. Obviously, the final design is very complicated and it will be convenient to do the job with the aid of a computer software called the QCADesigner. The program and the final drawing are prepared and are available as a project results in electronic research center of IUST. Now, we are in a position to give a modular pattern composed of logic and routing elements with the help of this design work. These modules can be called QCA chips. Of course, the actual fabrication of these chips needs many sophisticated technologies, the most important of them is two layer grid routing techniques (Michael *et al*., 2003).

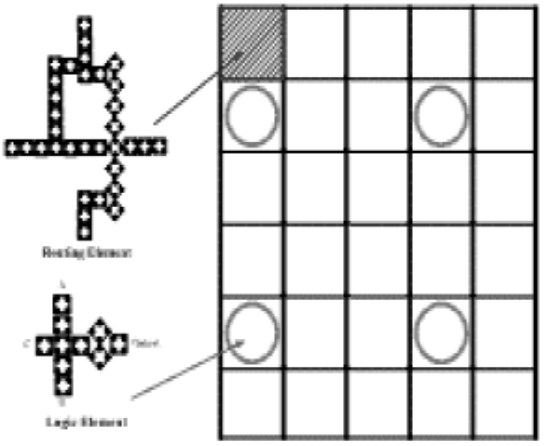


Fig. 14: Proposed pattern for a block of modular chip

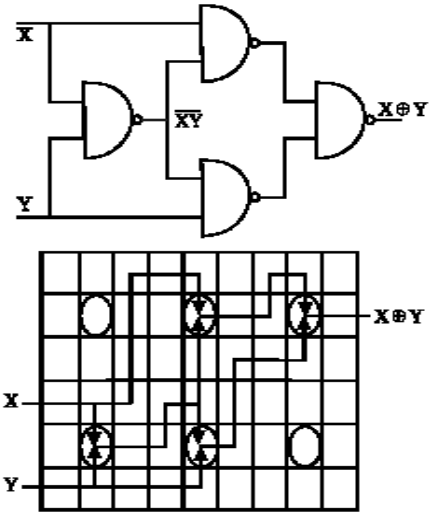


Fig. 15: How the NAND gates can be mapped in NAND-based QCA chip to simulate the logic of XOR-gate

The quantum cell must be small on the order of 20 nm to be efficient. At the moment the technology which can reliably manufacture quantum cells of this size does not exist. The final pattern is severely dependent on ultimate precision by which the optimal channel depth can be fabricated (Michael *et al*., 2003; Mary, 2006).

A block of this pattern is drawn as Fig. 14. In Fig. 14, circles represent logic blocks and squares represent the

possible path from one gate to another one. These paths are programmable and controllable by clocking zones. To accomplish the desired routing, a mechanism same as already used in traditional CMOS, is usable.

As mentioned earlier, the logic of XOR-gate can be converted to NAND logic (Fig. 12). The next step is placing this logic in QCA NAND-based chip (Fig. 15). Then the final design will be simply a repetition of Fig. 15.

CONCLUSION

In this research, some kind of QCA logic which is equivalent to a NAND/NOR gate, was presented.

Then after introducing Hamming coder/decoder the final design was implemented only by XOR-gates. Finally it was shown that how a routing element can be constructed by QCAs and how can QCA chip be made by periodic repetition of QCA NAND gates and QCA routing elements. With a small size chip of this kind one XOR-gate was constructed. It is also shown that with the aid of programming a large size chip of this kind, a complex design work such as Hamming coder/decoder can be implemented. As a great amount of research work is going to produce robust QCAs which can operate efficiently in room temperature, this kind of ultra small size and ultra low consumption Hamming coder/decoder will be used in future interplanetary communication.

REFERENCES

Amlani, I., A.O. Orlov, R.K. Kummamuru, G.H. Bernstein, C.S. Lent and G.L. Snider, 2000. Experimental demonstration of a leadless quantum-dot cellular automata cell. Applied Phys. Lett., 77: 738-740.

Demadis, K., C. Hartshorn and T. Meyer, 2001. The localized-to delocalized transition in mixed-valance chemistry. Chem. Rev., 101: 2655-2685.

Lent, C., P. Tougaw and W. Porod, 1993a. Bistable saturation in coupled quantum dots for quantum cellular automata. Applied Phys. Lett., 62: 714-716.

Lent, C., P. Tougaw, W. Porod and G. Bernstein, 1993b. Quantum cellular automata. Nanotechnology, 4: 49-57.

Lent, C.S. and P.D. Tougaw, 1997. A device architecture for computing with quantum dots. Proceedings of the IEEE, April 1997, pp: 541-547.

Lent, C.S., 2000. Molecular electronics: Bypassing the transistor, paradigm. Science, 288: 1597-1599.

Lieberman, M., S. Chellamma, B. Varughese, Y. Wang, C. Lent, G. Bernstein, G. Snider and F. Peiris, 2002. Quantum-dot cellular automata at a molecular scale. Ann. New York Acad. Sci., 960: 225-239.

Mary, J.B., 2006. Design and simulation of fault tolerant quantum-dot Cellular Automata (QCA) Not gates. M.Sc. Thesis ,Wichita State University, Kansas.

Michael, T.N., A.F. Rodrigues and P.M. Kogge, 2003. A Potentially Implementable FPGA for Quantum Dot Cellular Automata. Published in Workshop on Non-Silicon Computation (NSC), in Conjunction with Int. Symp. On High Performance Computer Architecture (HPCA, 03).

Mo, L., 2006. Robustness and power dissipation in quantumdot cellular automata. M.Sc. Thesis, Graduate School of the University of Notre Dame, India.

Orlov, A., I. Amlani, C. Lent, G. Bernstein and G. Snider, 1999. Experimental demonstration of a binary wire for quantumdot cellular automata. Applied Phys. Lett., 74: 2875-2877.

Walus, K., G.A. Jullien and V.S. Dimitrov, 2003. Computer Arithmetic Structures for Quantum Cellular Automata. IEEE, 2: 1435-1439.