International Journal of Computer and Electrical Engineering, Vol. 2, No. 4, August, 2010
1793-8163

Digital Combinational Circuits Design with the Help of Symmetric Functions Considering Heat Dissipation by Each QCA Gate

Pijush Kanti Bhattacharjee, *Member*

***Abstract*—Different logic gates like MV, NOT, AOI, NNI etc under QCA nanotechnology are introduced. NNI gate is highly effective regarding space and speed consideration. In earlier work, realizing symmetric functions in binary reversible logic is studied. In that, 2-inputs and 2-outputs AND-NAND and OR-NOR cells are designed. Ultimately a general equation for the minimum number of gates required to an arbitrary number of input variables causing synthesis of symmetric function is achieved. In current work, I compare the heat dissipation due to the garbage outputs for designing different combinational digital circuits with the help of symmetric functions realized circuits. It provides a significant reduction in hardware cost and switching delay with respect to the other existing techniques.**

***Index Terms*—Majority Voter (MV) gate, And-Or-Inverter (AOI) gate, Nand-Nor-Inverter (NNI) gate, 2-Inputs and 2-Outputs AND-NAND (A-NA) gate, 2-Inputs and 2-Outputs OR-NOR (O-NO) gate, Symmetric Function, Adder circuit.**

I. Introduction

Reversible Quantum Computers (QCs) has drawn the attention of researchers recently as such computers result improvement in speed, reducing cost and space etc [1]-[6], [11]-[18]. Reversible gates and circuits are the main components of Quantum Computer. Reversible circuits or gates have the same number of inputs and outputs, also have one-to-one mappings between vectors of inputs and outputs; thus the vector of the input states can be uniquely reconstructed from the vector of the output states. A symmetric function [11]-[18] means a Boolean function invariant to the permutation of any of its input variables. Also every Boolean function can be made symmetric by repeating its input variables. The application of symmetric function is in enormous field of the processor (IC) design and in higher mathematics problem solving. Any sort of Boolean functions can be simplified by symmetric functions.

In practical problems like design of complex circuitry, complex control systems with different parameters or variables, solving more variables consisting of functions etc may be done by proper application of symmetric function.

Fredkin Gate (FG) is a fundamental concept in reversible and quantum computing, the base of "realization-related" papers. It has been introduced by Ed Fredkin and Tomasso Toffoli in 1982 [12], [13], [16]-[18].

(i) Every Boolean function can be build from (binary) Fredkin Gates (FGs), such that it has three inputs A, B and C and three outputs P, Q and R like

P = A

Q = if A then C else B

R = if A then B else C

(ii) Feyman Gates is "Controlled NOT" or "quantum XOR", such gates have two inputs A and B and two outputs P and Q, they are called linear likewise

P = A

Q = A EXOR B

In earlier work [16], [18], I studied the synthesis of symmetric Boolean functions using a simple recursive arrangement of 2-inputs, 2-outputs AND-NAND (A-NA) and OR-NOR (O-NO) logic gates realized with QCA (Quantum Dot Cellular Automata) technology. It eliminates the use of NOT or inverter gate which is more costly and spacious as well as introduced delay in realizing the symmetric function. By this approach, the symmetric functions are synthesized for an arbitrary number of input variables. A general equation for estimating the number of gates (AND-NAND, OR-NOR etc) required for symmetric function realization [16] is obtained. Further, the proposed technique is applicable for any symmetric functions either unate or non unate functions results a logic design with less hardware cost which drastically reduces the "garbage" compared to the other existing techniques. Here I propose comparison of the heat dissipation due to the garbage outputs which rise for designing combinational circuits e.g. adder circuit etc with the help of symmetric functions realization circuit. Moreover it is observed that adder combinational circuits are obtained from symmetric functions realized circuits, further any combinational digital circuits can be synthesized from these adder circuits easily.

In unate symmetric functions, all variables or literals are either non complemented form, called unate positive or complemented form, called unate negative function.

e. g. $X_1X_2 + X_2X_3 + X_3X_1 \longrightarrow$ Unate positive function

$X_1' + X_2' + X_3' \longrightarrow$ Unate negative function

In non unate symmetric function, the variables are complemented or non complemented form i.e. imposing no restriction of the variable form,

e. g. $X_1'X_2'X_3 + X_1X_2'X_3' + X_1'X_2X_3'$

II. Quantum Dot Cellular Automata

QCA (Quantum-dot Cellular Automata) [1]-[18] devices encode and process binary information as charged arrays of charge coupled quantum dots. A quantum cell can be viewed as a set of four charge containers or dots positioned at the corners of a square, as shown in Fig. 1. It contains two extra mobile electrons. The electrons can quantum mechanically

tunnel between dots but can not come out from the cell and are forced to settle at the corner positions due to coulomb interaction. Thus, there exists two equivalent energetically minimal arrangements for the electrons in a QCA cell (Fig. 1), i.e. the polarization P = +1 (representing logic 1) and P = –1 (representing logic 0).

In Fig. 1, a QCA cell and its binary logic are shown, the energetically position of the diagonal electrons identifies the binary logic 0 or 1. This phenomenon is useful in nano technology which affects high resolution fast electronic circuits. In this power consumption for changing the charge of electron is very much less compare to that of general charge carriers (hole-electron) electronic components.

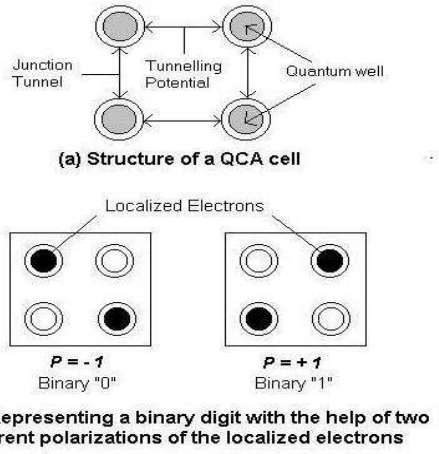

Fig. 1. A QCA cell and its binary logic

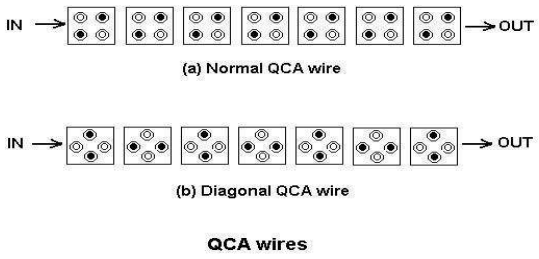

Fig. 2. Information propagating through QCA wires.

A QCA Cell with its binary logic creates a new direction in nano technology [1]-[9]. It requires minimum current or energy to change any state i.e. previous state. Thus, a minimum recurring cost is effective in this QCA gates which is highly applicable in super fast processors. Also complex circuitry are afforded with the minimum number of QCA gates, thus space-cost-speed-accuracy efficiency are achieved. Special type of QCA gates called AND-NAND (A-NA), OR-NOR (O-NO) etc are designed, by which the symmetric functions are realized. Later on these symmetric function realization circuits yield different type of combinational digital circuits including basic building block circuit i.e. adder circuit for an arbitrary number of input variables.

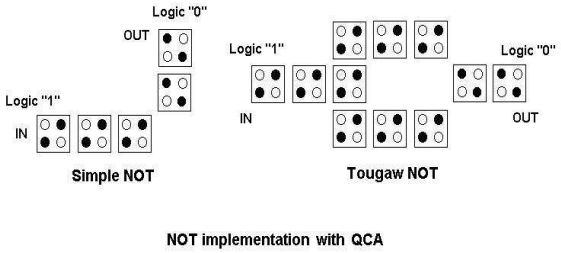

Fig. 3. Implementation of NOT with QCA Gate.

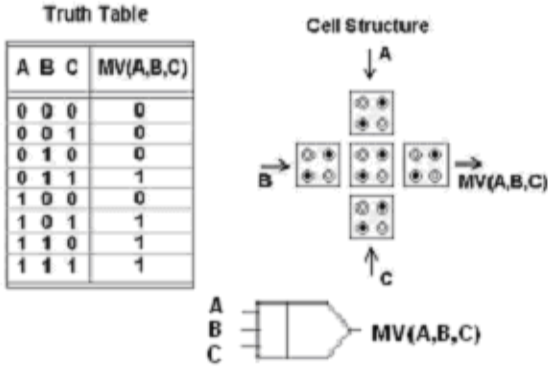

Fig. 4. QCA Majority Voter (MV) Gate

The basic QCA logic elements [1]-[8] include a QCA wires are shown in Fig. 2, QCA inverter or NOT gate in Fig. 3. This is a special type of QCA NOT gate in Fig. 3, which is designed on the basis of nano technology principle. Majority Voter (MV) or Majority Gate (Maj) is shown in Fig. 4. In diagonal (also called 45^0) wire, as shown in Fig. 2, binary signal alternates polarization in successive cells. The QCA Majority Voter (MV) realizes MV(A, B, C) = Maj(A, B, C) = AB+BC+CA, as in Fig. 4, outputs '1' if there are two or more 1s in an input pattern. The classical AND and OR gates can be realized with the majority gate by fixing an input as 0 and 1 respectively. The majority gate is not a universal gate. It can not realize the logical NOT operation. The functionally complete set is {MV, NOT}. Therefore, the designers have to use separate QCA cell arrangement for realization of the logical NOT. The 5-input (A, B, C, D and E) And-Or-Inverter (AOI) gate [7] proposed the universal gate function. AOI suffers from the limitation of proper separation of input and output binary wires i.e. distances d1, d2 and d3, shown in Fig. 5.

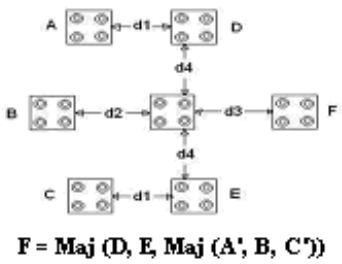

Fig. 5. QCA And-Or-Inverter (AOI) Gate.

And-Or-Inverter is a very typical gate which is rarely used in practical purposes. Another drawback of A-O-I gate is that it requires more space compare to other QCA gates, also the A-O-I gates are more complex nature comparing to that of MV or NOT gates.

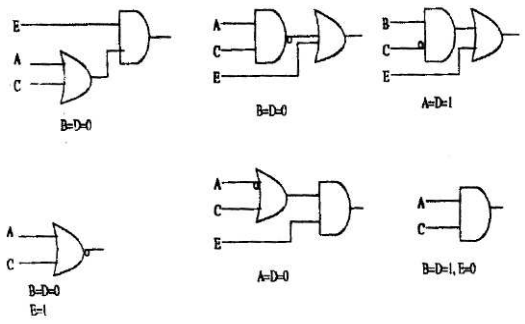

Fig. 6. Various logic functions realized with AOI Gate

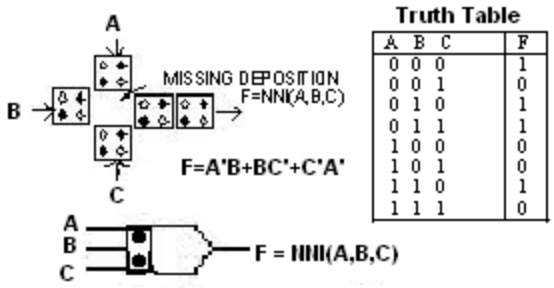

Fig. 7. QCA Nand-Nor-Inverter (NNI) Gate

Thus to implement MV with NOT functions, a new gate called Nand-Nor-Inverter (NNI) [1]-[10] is constructed, where NNI(A, B, C) = MV(A', B, C') = A'B+BC'+C'A'. It is shown in Fig. 7. The NNI gate is a universal gate and can be employed for realizing versatile Boolean logic functions [16]-[18]. It proves to be as effective as the AOI (And-Or-Inverter) gate and requires lesser overhead, for setting the variables, than that of an AOI, while realizing the basic logic gates. Recent developments in manufacturing QCA gates involve molecular assembly of QCA devices to supersede metal based implementations.

III. Synthesis Of Symmetric Function

The symmetric functions are paying attention to the researchers in the field of VLSI and nanotechnology design, especially for logic synthesis. A number of synthesis technique for Boolean symmetric functions are reported [12]-[18]. I represent a simple cost effective synthesis with QCA gates for future generation digital design.

A. Symmetric Function

A switching function $f(x_1,x_2,\ldots\ldots\ldots\ldots,x_n)$ is called totally symmetric with respect to the variables $x_1, x_2, x_3,\ldots\ldots\ldots,x_n$, [11]-[18], if it is invariant under any permutation of the variables. Total symmetry can be specified by a set of integers (called a numbers) $A = (a_i,\ldots..,a_j,\ldots\ldots\ldots,a_k)$ where $A \subset (0,1,2,\ldots,n)$; all the vertices with weight $w \in A$ will appear as true minterms in the function.

A n-variable symmetric function is denoted as $S^n(a_i,\ldots..,a_j,\ldots\ldots\ldots,a_k)$. A symmetric function is called consecutive, if the set A consists of only consecutive integers $(a_l, a_{l+1},\ldots\ldots\ldots,a_r)$. It is expressed as $S^n(a_l - a_r)$ where $l<r$.

For n variables, there can be $(2^{n+1} - 2)$ different symmetric functions (excluding constant functions 0 and 1). Each totally symmetric functions, $S^n(A) = S^n(A_1) + S^n(A_2) +\ldots\ldots\ldots\ldots+ S^n(A_m)$, where m is minimum, $\forall$ i, j, $l \le i, j \le m$, $A_i \cap A_j = \varnothing$ (null), and $i \neq j$. Thus, a symmetric function may be broken in lower order consecutive symmetric functions keeping all condition same.

IV. Methodology And Implementation

I discuss the proposed method of symmetric function synthesis and its implementation [16]-[18] in this section. Two type of gates e.g. AND-NAND (A-NA) and OR-NOR (O-NO) gates are used. The output of an AND-NAND gate is AND and NAND functions of the inputs. Similarly, the output of an OR-NOR gate is OR and NOR functions of the inputs. The gate is having two inputs and two outputs. These two gates are the simplest gates of QCA cells under nano technology. Thus the use of NOT or inverter gate is eliminated which takes more area, costlier, time and delay consuming.

In the synthesis of symmetric functions [16], [18] I get all the combinations of symmetric functions for a particular set of input variables including unate positive and unate negative symmetric functions. This becomes a generalized solution for the symmetric functions realization for any number of input variables. Further adder circuits are derived from these symmetric functions synthesis, ultimately it uses for construction of any combinational digital circuits.

A. Method of Synthesis Symmetric Function

First of all, I take 2-inputs and 2-outputs AND-NAND, OR-NOR gates as shown in Fig. 8.

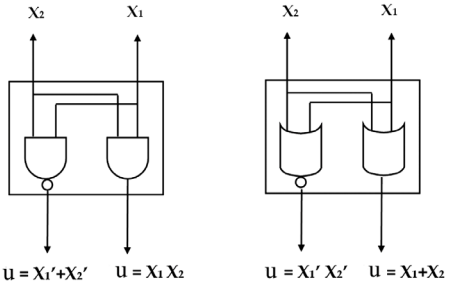

Fig. 8. 2-inputs and 2-outputs AND-NAND Gate, 2-inputs and 2-outputs OR-NOR Gate Circuits.

Here, AND-NAND gate and OR-NOR gate are two type of QCA gates which are designed on nano technology process. It requires less overhead area and at the same time ensures best efficiency in speed.

B. 2-Inputs Symmetric Function Synthesis

For 2-input variables, I take one AND(A)-NAND(NA) gate and two OR(O)-NOR(NO) gates to get all the symmetric functions composed by two variables. The number of symmetric functions consists of two variables as per $(2^{n+1} - 2)$ formula are 6 followings:

i) AB = u1 = $S^2(2)$ = Carry
ii) A'+B' = u2 = $S^2(0,1)$
iii) A+B = u3 = $S^2(1,2)$
iv) A'B' = u4 = $S^2(0)$
v) AB+A'B' = u5 = $S^2(0,2)$
vi) AB'+A'B = u6 = $S^2(1)$ = Sum

The realization of symmetric functions by

AND(A)-NAND(NA) and OR(O)-NOR(NO) gates is shown in Fig. 9. These QCA based gates are taking minimum space area as well as minimum power consumed to formulate any type of symmetric outputs. Therefore, these symmetric functions realization integrated circuits may be manufactured at a minimum cost surprising all other type of invented ICs, especially when it is manufactured at large scale. This type of symmetric functions synthesis may be implemented in all other complex circuit designing also.

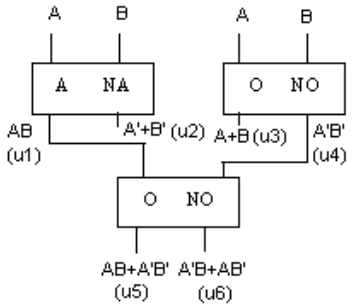

Fig. 9. Symmetric functions circuit diagram for 2-Input variables.

For two input variables, total three gates are required, one AND-NAND gate and two OR-NOR gates and no output from any gate is garbage output i.e. not belonging to the symmetric functions. Therefore, this construction of 2-input symmetric functions is the simplest form and ensures fastest operation. Since this circuit is not having any extra output i.e. garbage output, it ultimately curtails the cost of this 2-input symmetric function IC design. For 2-binary input variables adder circuit designed as above by symmetric functions synthesis, to get sum at u6 i.e. $S^2(1)$ and carry at u1 i.e. $S^2(2)$.

C. 3-Inputs Symmetric Function Synthesis

For 3-input variables, additional 8 gates are required to get all the symmetric functions (14 numbers). So, total 11 number of gates are necessary if starting from 2-input variables. The numbers of symmetric functions consisting of three input variables are followings.

i) ABC = u7 = $S^3(3)$
ii) A'+B'+C' = u8 = $S^3(0,1,2)$
iii) A+B+C = u9 = $S^3(1,2,3)$
iv) A'B'C' = u10 = $S^3(0)$
v) ABC+A'B'C' = u11 = $S^3(0,3)$
vi) A'B+AB'+A'C+AC'+B'C+BC' = u12 = $S^3(1,2)$
vii) AB+BC+CA = u13 = $S^3(2,3)$ = Carry
viii) A'B'+B'C'+C'A' = u14 = $S^3(0,1)$
ix) A'B'+B'C'+C'A'+ABC = u15 = $S^3(0,1,3)$
x) A'BC+AB'C+ABC' = u16 = $S^3(2)$
xi) AB+BC+CA+A'B'C' = u17 = $S^3(0,2,3)$
xii) AB'C'+A'BC'+A'B'C = u18 = $S^3(1)$
xiii) AB'C'+A'BC'+A'B'C+ABC = u19 = $S^3(1,3)$ = Sum
xiv) A'BC+ABC'+AB'C+A'B'C' = u20 = $S^3(0,2)$

In Fig. 10, total 2 numbers of AND-NAND gates and 6 numbers of OR-NOR gates are required for construction of all the symmetric functions (14 numbers) by 3-input variables and hence total 8 gates are required. Two outputs from the AND-NAND gate are garbage outputs i.e. not a symmetric function. Out of 2-garbage outputs, one output is not associated to act as further input. So, one output is completely garbage output. The other garbage output is feeding as input to next stage OR-NOR gate. The efficiency of the circuit regarding power consumption and speed is the maximum.

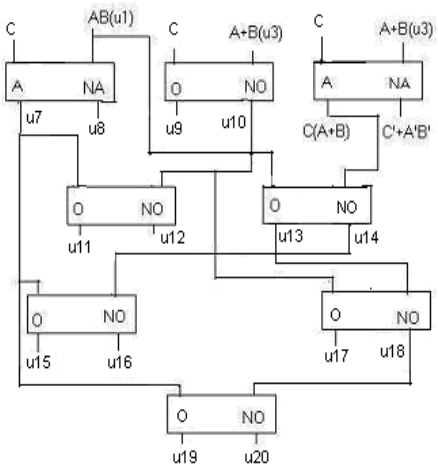

Fig. 10. Symmetric functions circuit diagram for 3-Input variables.

Full adder circuit by employing two half adder circuits is constructed. In the proposed technique, I use AND-NAND & OR-NOR QCA gates only [12], [16]-[18]. Here Full adder Boolean functions for sum and carry are at u19 i.e. $S^3(1,3)$ and at u13 i.e. $S^3(2,3)$ respectively. Therefore the adder circuit outputs are derived from this 3-input symmetric function synthesis circuit.

It requires total eight QCA gates, out of which two number AND-NAND gates and six number OR-NOR gates are used as shown in Fig. 10. In this case, the garbage outputs are one in number.

D. 4-Inputs Symmetric Function Synthesis

Similarly, I realize the symmetric functions for 4-input variables. Total numbers of symmetric functions for 4 input variables (30) are shown below:

i) A+B+C+D = u21 = $S^4(1,2,3,4)$
ii) A'B'C'D' = u22 = $S^4(0)$
iii) ABCD = u23 = $S^4(4)$
iv) A'+B'+C'+D' = u24 = $S^4(0,1,2,3)$
v) ABCD+A'B'C'D' = u25 = $S^4(0,4)$
vi) A'B+AB'+B'C+BC'+A'C+AC'+A'D+AD'+B'D+BD'+ C'D+ CD' = u26 = $S^4(1,2,3)$
vii) ABC+ABD+BCD+ACD = u27 = $S^4(3,4)$ = Carry
viii) A'B'+B'C'+C'A'+A'D'+B'D'+C'D' = u28 = $S^4(0,1,2)$
ix) A'B'C'+A'B'D'+A'C'D'+B'C'D'= u29 = $S^4(0,1)$
x) AB+BC+CA+AD+BD+CD= u30 = $S^4(2,3,4)$
xi) A'B'+B'C'+C'A'+A'D'+B'D'+C'D'+ABCD = u31 = $S^4(0,1,2,4)$
xii) A'BCD+AB'CD+ABC'D+ABCD' = u32 = $S^4(3)$
xiii) ABC'D'+A'BCD'+A'BC'D+AB'CD'+AB'C'D+ A'B'CD = u33 = $S^4(2)$
xiv) ABC+ABD+ACD+BCD+A'B'C'+B'C'D'+A'C'D'+ A'B'D' = u34 = $S^4(0,3,4,1)$
xv) ABC+ABD+BCD+ACD+A'B'C'D' = u35 = $S^4(0,3,4)$
xvi) AB'C'+AC'D'+A'B'D+A'C'D+A'CD'+BC'D'+AB'D'+ A'B'C+B'C'D+B'CD'+A'BD'+A'BC' = u36 = $S^4(1,2)$
xvii) AB+BC+CA+AD+BD+CD+A'B'C'D' = u37 = $S^4(0,2,3,4)$
xviii) AB'C'D'+A'BC'D'+A'B'CD'+A'B'C'D = u38 = $S^4(1)$
xix) A'B'C'+A'B'D'+A'C'D'+B'C'D'+ABCD = u39 = $S^4(0,1,4)$

xx) A'BC+A'CD+A'BD+AB'C+B'CD+AB'D+ABC'+BC'D + AC'D+ABD'+BCD'+ACD' = u40 = $S^4(2,3)$

xxi) A'BCD+AB'CD+ABC'D+ABCD'+A'B'C'D' = u41 = $S^4(0,3)$

xxii) AB'C'+AC'D'+A'B'D+A'C'D+A'CD'+BC'D'+AB'D'+ A'B'C+B'C'D+B'CD'+A'BD'+A'BC'+ABCD = u42 = $S^4(1,2,4)$

xxiii) ABC'D'+A'BCD'+A'BC'D+AB'CD'+AB'C'D+ A'B'CD + A'B'C'D' = u43 = $S^4(0,2)$

xxiv) ABC+ABD+BCD+ACD+AB'C'D'+A'BC'D'+A'B'C D'+ A'B'C'D = u44 = $S^4(1,3,4)$

xxv) A'BC+A'CD+A'BD+AB'C+B'CD+AB'D+ABC'+BC'D + AC'D+ABD'+BCD'+ACD'+A'B'C'D' = u45 = $S^4(0,2,3)$

xxvi) AB'C'D'+A'BC'D'+A'B'CD'+A'B'C'D+ABCD = u46 = $S^4(1,4)$

xxvii) ABC'D'+A'BCD'+A'BC'D+AB'CD'+AB'C'D+ A'B'CD + ABCD = u47 = $S^4(2,4)$

xxviii) A'B'C'+A'B'D'+A'C'D'+B'C'D'+A'BCD+AB'CD+ ABC'D+ABCD' = u48 = $S^4(0,1,3)$

xxix) ABC'D'+A'BCD'+A'BC'D+AB'CD'+AB'C'D+A'B'C D+ ABCD+A'B'C'D' = u49 = $S^4(0,2,4)$

xxx) A'BCD+AB'CD+ABC'D+ABCD'+A'B'C'D+A'B'CD'+ A'BC'D'+AB'C'D' = u50 = $S^4(1,3)$ = Sum

I require total 17 gates, out of which total AND-NAND gates are 3 numbers and OR-NOR gates are 14 numbers for 4 input variables. Fig. 11 shows the circuit diagram of the realized symmetric functions by taking 4-input variables and Table-I indicates comparison of all the parameters of symmetric functions realization in a nutshell. Total numbers of gates are required for two variables are 3, for three variables are 8, for four variables 17 etc. For adding 4-inputs binary variable, the sum is obtained as u50 or $S^4(1,3)$ and the carry is obtained as u27 or $S^4(3,4)$ from symmetric function synthesis. Garbage output is two numbers. Generally sum of the adder circuit is appearing at output of the last gate.

I design full adder circuit for an arbitrary number of binary (digital) inputs from symmetric functions realized circuit; accordingly the sum and the carry are obtained as output. Therefore, a general equation for addition of any number of binary (digital) inputs i.e. full adder circuit design regarding the number of gates required is formulated and the garbage outputs are calculated. Total comparison regarding all parameters and attributes for adder circuit design from symmetric functions realization circuit for an arbitrary number of inputs are explained in Table-I.

Likewise multiplexer, demultiplexer, encoder (Decimal to Octal, Binary to Hexadecimal etc), decoder, code converter (BCD to Gray, Gray to ASCII etc) and different logic combination functions can be well realized by this adder circuit only.

V. General Equation For Synthesis Of Symmetric Functions By And-Nand, Or-Nor Qca Gates

A general equation for total number of gates requiring [16], [18] is developed. For n number of input variables, we have $(2^{n+1} - 2)$ numbers of symmetric functions, that requires total $(2^n + n - 3)$ number of gates. Similarly, for 5 input variables, to realize all the 62 numbers symmetric functions, at least 34 numbers of total gates are required, accordingly AND-NAND, OR-NOR gates are distributed.

Therefore, maximum optimization is done to have all the symmetric functions for an arbitrary number of input variables (say n variables) by AND-NAND and OR-NOR gates on QCA based design principle. I achieve a general equation for requiring the total number of gates (like AND-NAND, OR-NOR gates) to n-number of input variables as $(2^n + n - 3)$ number of gates. This equation is explained and modified as below:

If X is the total number of gates requiring for n number of input variables comprising with $(2^{n+1}-2)$ number of symmetric functions,

$$X = (2^n + n - 3)$$

or,

$$X = \frac{\text{Total No of n-variable Symmetric Fuction}}{2} + (n-2)$$

or,

$$X = \frac{\text{Total No of n-variable Symmetric Fuction}}{2} + (\text{Number of garbage output})$$

Here, $(n - 2)$ is the garbage output. The garbage output is neither a symmetric function nor fed as further input to the gates. Thus, the garbage output can not be used further. Symmetric functions realized for 2 input variables, garbage output is 0; for 3 inputs, garbage output 1; for 4 inputs garbage outputs 2; for 5 inputs garbage outputs 3 etc, so garbage outputs are in AP series, which is clearly indicated in Table-I.

Again, $X = 2^n + n - 3 = (n - 1) + (2^n - 2)$

or, X = Number of AND-NAND gates + Number of OR-NOR gates

I require at least total $(n - 1)$ number of AND-NAND gates and total $(2^n - 2)$ number of OR-NOR gates to have all the symmetric functions realized by n-input variables. The detailed comparison of the parameters for symmetric function synthesis is shown in Table-I. These symmetric functions yield adder circuit functions in addition to all other different kind of Boolean functions i.e. digital combinational circuits.

VI. Symmetric Functions Synthesis Compared By And-Or-Inverter (Aoi) With And-Nand, Or-Nor Qca Gates Specifying Heat Dissipation Due To Garbage Outputs

One AND-NAND gate or one OR-NOR gate is constructed by two AND-Or-Inverter (A-O-I) gates. Therefore, the garbage outputs obtained from the implementation of symmetric functions by using A-O-I gates are double number of the garbage outputs derived from AND-NAND or OR-NOR gates. Thus the heat dissipation due to the garbage outputs using AOI gate is also double that of using AND-NAND and OR-NOR gates based on QCA technology. Nanotechnology supersedes the limitations of CMOS at the end of the roadmap. One of the hurdles in the development of super fast computing system is energy

dissipation. In reversible computing, the relation between energy dissipation and computing at logic level [17] is calculated. Reversible computation is established at logic level by a one to one onto mapping between the input and the output states of a circuit. It is proved by Landauer that the lower bound of heat dissipation for every bit of information lost in computing or for the each garbage output which is not fed as further input is KTln2 Joules, where K is Boltzmann's constant and T is the temperature in ^{0}Kelvin. Moreover, heat dissipation can be avoided if computation is carried out with no loss of information or without any garbage output. Due to the bijective property, reversible logic gates are information lossless i.e. the information at the output of a reversible circuit is maximized.

Thus, the amount of heat dissipation by the garbage output for realization of the symmetric functions for an arbitrary number of input variables are computed according to the equation KTln2 Joules dissipating for one garbage output. It is clearly described in Table-I. Therefore in all other combinational digital circuits, designed from adder circuit, heat dissipation can also be computed. .

VII. Conclusions

Although symmetric functions are difficult to realize practically with the minimum number of gates, my proposed design with adoption of QCA technology enlightens a new direction in the field of symmetric function synthesis and other digital combinational circuits e.g. adder circuits etc [16], [18] as well as heat dissipation comparison with A-O-I gates. Ultimately a general solution for minimum number of QCA gates (AND-NAND and OR-NOR) requiring [16], [18] is developed, identifying the "garbage" output. In this paper, I compute the heat dissipation due to the garbage outputs for various numbers of the inputs using either AOI gate or our designed AND-NAND, OR-NOR gates based on QCA nanotechnology. Therefore QCA based design of symmetric functions are simulated by AND-NAND, OR-NOR gates and compared with the results of conventional CMOS based technique and A-O-I gate. These proposed QCA gates AND-NAND, OR-NOR gates are able to design adder circuits with an arbitrary number of binary inputs. Ultimately a general solution for minimum number of QCA gates (AND-NAND and OR-NOR) requiring is invented, identifying the "garbage" output. It is also observed that the implementation of symmetric functions with QCA design is more advantages than that of CMOS design. In all aspects, this QCA based design symmetric functions, yielding other combinational digital circuits i.e. Boolean functions, are highly suitable for higher growth of IC processors in the forthcoming generation.

References

[1] C. S. Lent, P. D. Taugaw, W. Porod and G. H. Berstein, "Quantum Cellular Automata", Nanotechnology, vol. 4, no. 1, pp 49-57, January 1993.

[2] Z. Kohavi, "Switching and Finite Automata Theory", Tata McGraw Hill Pub Company Ltd, New Delhi, 2nd Edition, 2007.

[3] I. Amlani, A. O. Orlov, G. Toth, C. S. Lent, G. H. Bernstein and G. L. Sinder, "Digital Logic Gate using Quantum Dot Cellular Automata", Science, vol. 284, no. 5412, pp. 289-291, April 1999.

[4] M. A. Nielsen and I. L. Chuang, "Quantum Computation and Quantum Information", 1st Edition, Cambridge University Press, New Delhi, 2006.

[5] A. O. Orlov, I. Amlani, G. H. Bernstein, C. S. Lent and G. L. Sinder,"Realization of a Functional Cell for Quantum Dot Cellular Automata", Science, vol. 277, no. 5328,.pp 928-930, August 1997.

[6] C. S. Lent, P. D. Taugaw, "A Device Architecture for Computing with Quantum Dots", Proceedings IEEE, vol. 85, no. 4, pp. 541-557, April 1997.

[7] M. Momenzadeh, M. B. Tahoori, J. Huang and F. Lombardi, "Characterization, Test and Logic Synthesis of AND-OR-INVERTER (AOI) Gate Design for QCA Implementation", IEEE Trans on Computer Aided Design of Integrated Circuits and Systems, vol. 24, no. 12, pp. 1881-1893, December, 2005.

[8] R. Zhang, K. Walus, W. Wang and G. A. Jullien, "A Method of Majority Logic Reduction for Quantum Cellular Automata", IEEE Trans on Nanotechnology, vol. 3, no. 4, pp. 443-450, Dec 2004.

[9] K. Walus, G. Schulhof, G. A. Jullien, R. Zhang, W. Wang, "Circuit Design Based on Majority Gates for Application with Quantum Dot Cellular Automata", IEEE Trans Signals, Systems and Computers, vol. 2, pp. 1354-1357, Nov 2004.

[10] M. Lieberman, S. Chellamma, B. Varughese, Y. Wang, C. S. Lent, G. H. Bernstein, G. L. Snider and F. Peiris, "Quantum Dot Cellular Automata at a Molecular Scale", Annals of the New York Academy of Sciences, vol. 960, pp. 225-239, 2002.

[11] M. Perkowski, P. Kerntopf, A. Buller, M. Chrzanowskajeske, A. Mishchenko, X. Song, A. Al-Rabadi, L. Jozwiak, A. Coppola, B. Massey, " Regular Realization of Symmetric Functions Using Reversible Logic", Euromicro Symposium on Digital Systems Design, pp. 245-252, 2001.

[12] D. T. Lee, S. J. Hong, "An Algortithm for Transformation of an Arbitrary Switching Function to a Completely Symmetric Function", IEEE Trans on Computer, vol. 25, no. 11, pp 1117-1123, November 1976.

[13] E. Fredkin, T. Toffoli, "Conservative Logic", International Journal of Theoretical Physics, vol. 21, no. 3-4, pp 219-253, April 1982.

[14] D. L. Dietmeyer, "Generating Minimal Covers of Symmetric Function", IEEE TCAD, vol. 12, no. 5, pp. 710-713, May 1993.

[15] S. Chakraborty, S. Das, D. K. Das and B. B. Bhattacharya, "Synthesis of Symmetric Functions for Path Delay Fault Testability", IEEE Trans. Comp. Aided Design of ICs and Sys., vol. 19, no. 9, pp. 1076-1081, Sept 2000.

[16] P. K. Bhattacharjee, "Efficient Synthesis of Symmetric Functions" (WORLDCOMP 2008) World Congress in Computer Science, Computer Engineering and Applied Computing in International Conference on Computer Design (CDES'08), pp. 23-29, las Vegas, USA,14-17th July, 2008.

[17] X. Ma, J. Huang, C. Metra, F. Lombardi, "Reversible Gates and Testability of One Dimensional Arrays of Molecular QCA", Journal of Electronic Testing, vol. 24, pp. 297-311, January, 2008.

[18] P. K. Bhattacharjee, "Use of Symmetric Functions Designed by QCA Gates for Next Generation IC", International Journal of Computer Theory and Engineering, Singapore, vol. 2, no. 2, pp 212-219, April, 2010.

Dr. Pijush Kanti Bhattacharjee is a pioneer in Engineering, Management, Law, Indo-Allopathy, Herbal, Homeopathic and Yogic medicines. He is having qualifications M.E, MBA, MDCTech, A.M.I.E, B.Sc, B.A, LLB, BIASM, CMS, PET, EDT, FWT, DATHRY, B.Mus, KOVID, DH, ACE, FDCI etc. He worked as an Engineer in Department of Telecommunications (DoT), Govt. of India from June 1981 to Jan 2007 (26 years), lastly holding Assistant Director post at RTEC [ER], DoT, Kolkata, India. Thereafter, he worked at IMPS College of Engineering and Technology, Malda, WB, India as an Assistant Professor in Electronics and Communication Engineering Department from Jan,2007 to Feb,2008 and Feb, 2008 to Dec, 2008 at Haldia Institute of Technology, Haldia, WB, India. In Dec, 2008 he joined at Bengal Institute of Technology and Management, Santiniketan, WB, India in the same post and department. He has written two books "Telecommunications India" & "Computer". He is a Member of IE, ISTE, IAPQR, IIM, ARP, India; CSTA, USA; IACSIT, Singapore and IAENG, Hongkong. .His research interests are in Telecommunications including Mobile Communications, Image Processing, Network Security, VLSI, Nanotechnology etc.

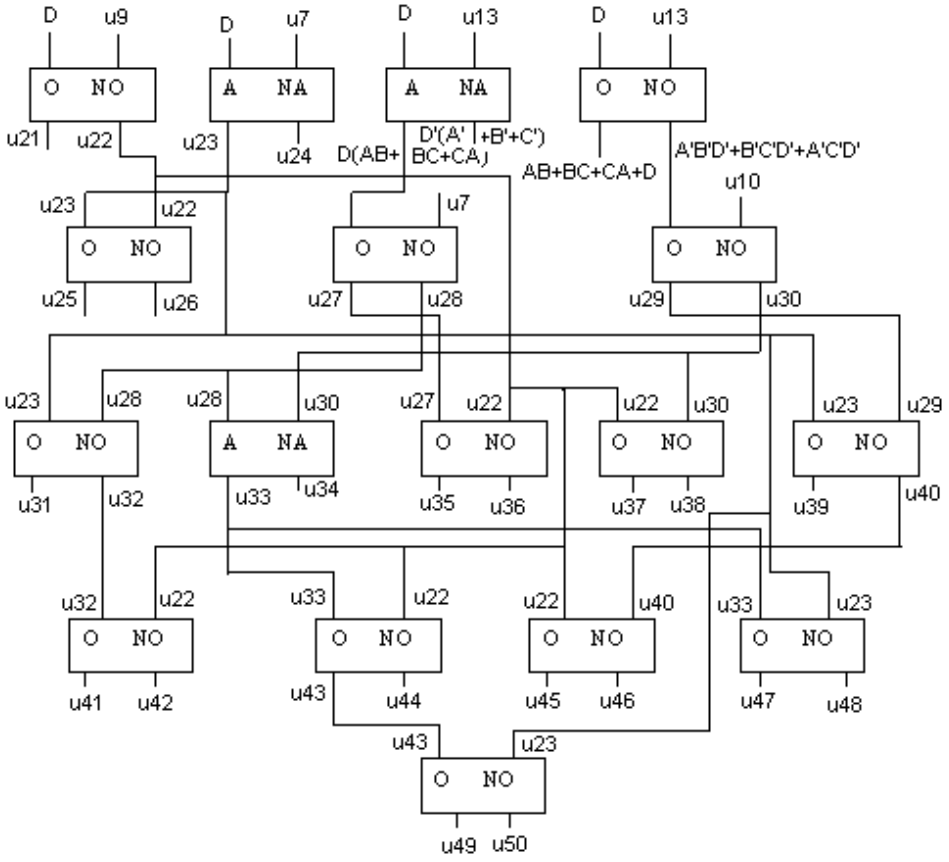

Fig. 11. Symmetric functions circuit diagram for 4-Input variables.

TABLE - ICOMPARISON OF SYMMETRIC FUNCTIONS & COMBINATIONAL DIGITAL CIRCUITS REALIZED BY ARBITRARY NUMBER OF INPUT VARIABLES

Srl No.	Number of Input Variables	Total Number of Symmetric Functions	Total Number of AND-NAND Gates	Total Number of OR-NOR Gates	Total No of gates i.e. AND-NAND, OR-NOR Gates	Total Number of Equivalent A-O-I Gates	Total No of Garbage outputs using AND-NAND, OR-NOR Gates	Total No of Garbage output not act as further input for A-NA & O-NO Gates	Heat Dissipation for total garbage outputs which do not fed as further inputs using AND-NAND, OR-NOR Gates	Total No of Garbage output not act as further input for Equivalent A-O-I Gates	Heat Dissipation for total garbage outputs which do not fed as further inputs using Equivalent A-O-I Gates
(1)	2	6	1	2	3	6	NIL	NIL	NIL	NIL	NIL
(2)	3	14	2	6	8	16	2	1	0.287 x 10^{-20} J	2	0.574 x 10^{-20} J
(3)	4	30	3	14	17	34	4	2	0.574 x 10^{-20} J	4	1.148 x 10^{-20} J
(4)	5	62	4	30	34	68	6	3	0.861 x 10^{-20} J	6	1.722 x 10^{-20} J
(5)	n	$2^{n+1} - 2$	n - 1	$2^n - 2$	$2^n + n - 3$	$2^{n+1} + 2n - 6$	2n - 4	n – 2	(n – 2) x 0.287 x 10^{-20} J	2(n – 2)	(n – 2) x 0.574 x 10^{-20} J