ENGINEERING IN ADVANCED RESEARCH SCIENCE AND TECHNOLOGY

ISSN 2278-2566
Vol.01, Issue.03
April -2019
Pages: -107-114

POWER OPTIMISED EFFICIENT BCD ADDER USING PTL LOGIC

1.BOPPANA NAGALAKSHMI, 2. NAGULAPATI SOWJANYA

1. PG Scholar, Dept. Of ECE, Nova College of Engineering and Technology, Ibrahimpatnam, A.P
2. Assistant Professor, Dept. Of ECE, Nova College of Engineering and Technology, Ibrahimpatnam, A.P

ABSTRACT:

A rapid growth is been observed in the area of Integrated Circuits (IC) technology. All the ICs are to be designed in an optimized way so that they meet all the requirements of being faster, occupying less area and reduced power consumption .Among the emerging technologies recently proposed as alternatives to the classic CMOS, Quantum-dot cellular automata (QCA) is one of the most promising solutions to design ultra lowpower and very high speed digital circuits. Efficient QCA-based implementations have been demonstrated for several binary and decimal arithmetic circuits, but significant improvements are still possible if the logic gates inherently available within the QCA technology are smartly exploited. This brief proposes a new approach to design QCA-based BCD adders. Exploiting innovative logic formulations and purpose designed QCA modules, computational speed significantly higher than existing counterparts are achieved without sacrificing either the occupied area or the cells count. Further, this concept is enhanced by using PTL logic for further reduction of power consumption.

KEYWORDS: Quantum-dot cellular automata (QCA), Pass Transistor Through Logic (PTL), Binary Code decimal.

INTRODUCTION:

Quantum-dot cellular automata (QCA) technology provides a promising opportunity to overcome the limits of conventional CMOS technology [1]–[2]. To designing more complicated circuits in future the QCA is getting recognition by designing engineers. In direction of QCA some arithmetic circuits and logical circuit already had been designed such as adder [3]-[4], multipliers[5]–[7], and comparators [8]–[10]. Autonomously of the performed logic function, nonelementary digital modules are designed intelligently combining inverters and majority gates (MGs), which are the basic fundamental logic gates inherently available within the QCA technology. Numerical decimal arithmetic has currently received a great deal of attention since several financial, economical, commercial, and Internet-based applications increasingly require higher precision. In these processes, the errors coming from the conversion between decimal and binary number representations could not be acceptable and several recent microprocessors [11]–[12] include hardware decimal arithmetic units in their core based on the IEEE 754–2008 standard [13]. The design of such digital circuits requires proper planning at both logic and layout levels to improve performance and Area.

Quantum Cellular Automata (QCA) is a nanotechnology that has recently been identified as one of the most emerging technology with potential application in future generation processing units [1, 2] as the CMOS technologies approach [3, 4] its fundamental physical limits in the recent years to development of nanotechnology for future generation ICs [5, 6].In the recent years QCA gets popularity to create computing devices and implementing any logic function. The basic design block of QCA circuit is majority gate; hence, constructing QCA circuits using majority gates has attracted a lot of attentions [7, 8]. Lot of studies has reported that QCA can be used to implement general purpose computational and memory circuits. QCA achieved high device density, fast switching speed, room temperature operation and very low power consumption, compare to any recent emerging technology [2]. The objective of this paper is to design a novel XOR gate by which we design a half adder circuit, here we also implement detailed design, layout and simulation of combinational circuits [9]. We proposed an optimal design for XOR based half adder circuit. The aim is to maximize the circuit density and focus on the layouts that are simple and minimal in their use of cells. The proposed QCA

circuits have been designed and simulated using the QCA designer tool [10]. The rest of the paper is organized as follows. Microprocessor manufacturing processes was governed by Moore's law, and consequently microprocessor performance till now. Today many integrated circuits are manufactured at 0.25-0.33 micron processes. But recent studies indicate that as early as 2010, the physical limits of transistor sizing may be reached [2]. However the performance of various circuits in current CMOS-based architectures is close to reaching the limit. If the feature size of transistors is further reduced to a nanometer, it will produce quantum effects such as tunneling. Further, during device scaling process due to the effects of wire resistance and capacitance, the interconnections never scale automatically. Addition is an essential operation many Digital, Analog, or Control system [9]-[12]. Fast and accurate operation of all digital system depends on the performance of adders .

QUANTUM CELLULAR AUTOMATON (QCA):

A **quantum cellular automaton (QCA)** is an abstract model of quantum computation, devised in analogy to conventional models of cellular automata introduced by von Neumann. The same name may also refer to quantum dot cellular automata, which are a proposed physical implementation of "classical" cellular automata by exploiting quantum mechanicalphenomena. QCA have attracted a lot of attention as a result of its extremely small feature size (at the molecular or even atomic scale) and its ultra-low power consumption, making it one candidate for replacing CMOS technology.

In the context of models of computation or of physical systems, *quantum cellular automaton* refers to the merger of elements of both (1) the study of cellular automata in conventional computer science and (2) the study of quantum information processing. In particular, the following are features of models of quantum cellular automata:

- The computation is considered to come about by parallel operation of multiple computing devices, or **cells**. The cells are usually taken to be identical, finite-dimensional quantum systems (e.g. each cell is a qubit).
- Each cell has a neighborhood of other cells. Altogether these form a network of cells, which is usually taken to be regular (e.g. the cells are arranged as a lattice with or without periodic boundary conditions).
- The evolution of all of the cells has a number of physics-like symmetries. Locality is one: the next state of a cell depends only on its current state and that of its neighbours. Homogeneity is another: the evolution acts the same everywhere, and is independent of time.
- The state space of the cells, and the operations performed on them, should be motivated by principles of quantum mechanics.

THE QCA CELL:

In contrast to electronics based on transistors, QCA does not operate by the transport of electrons, but by the adjustment of electrons in a small limited area of only a few square nanometers. QCA is implemented by quadratic cells, the so-called QCA cells. In these squares, exactly four potential wells are located, one in each corner of the QCA cell (see figure 1). In the QCA cells, exactly two electrons are locked in. They can only reside in the potential wells. The potential wells are connected with electron tunnel junctions. They can be opened for the electrons to travel through them under a particular condition, by a clock signal. A later chapter will cover this in more detail. Without any interaction from outside, the two electrons will try to separate from each other as far as possible, due to the Coulomb force that interacts between them. As a result, they will reside in diagonally located potential wells, because the diagonal is the largest possible distance for them to reside (see figure 2).

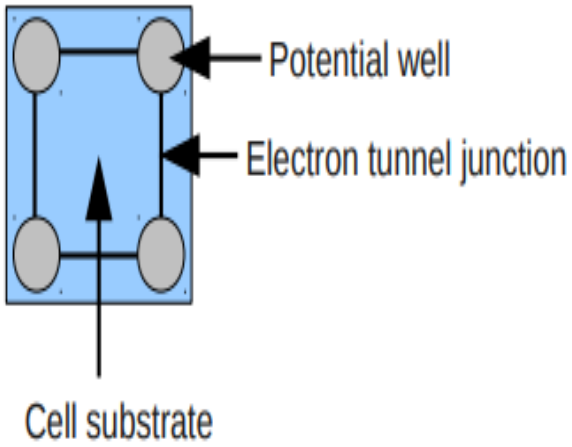


Figure : Anatomy of a QCA cell

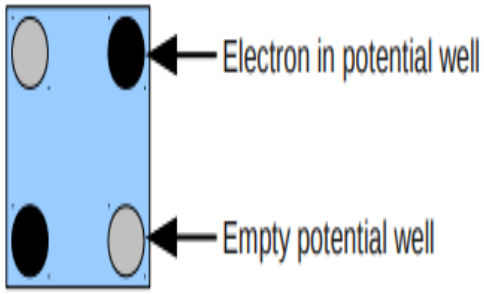


Figure : Electrons in potential wells

There are two diagonals in a square, which means the electrons can reside in exactly two possible adjustments in the QCA cell. Regarding these two arrangements, they are interpreted as a binary '0' and binary '1', i.e. each cell can be in two states. The state '0' and the state '1', as shown in figure 3. A binary system is something familiar, as boolean logic is used already in today's computers. There, a high voltage is often interpreted as binary '1' and a low voltage as binary '0'.

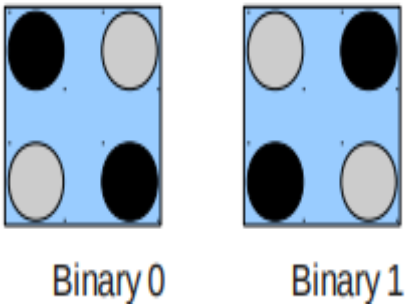


Figure: Binary interpretation of adjustments

If two QCA cells are placed next to each other, it is possible to exchange their states, i.e. the adjustments of the electrons in them. The QCA cell that should transfer its state to a neighboring cell must have its tunnel junctions closed, the tunnel junctions in the neighboring cell have to be open, to allow the electrons to travel through the tunnel junctions between the potential wells. As soon as they open, the electrons in the neighboring cell are pushed by the Coulomb force of the original cell as far away as possible. As they also are pushed away from each other, they will travel into the same potential wells as in the original cell. As soon as the tunnel junctions are closed again, the transfer of the state is completed. The state of a cell can also be transferred to multiple neighboring cells. It works the very same way as with a single neighbor cell, but the tunnel junctions of all the sequentially neighboring cells should be open at the same time, which makes the transfer much faster then transferring the state cell by cell. This allows us to build "wires", made of QCA cells, to transport information over larger distances.

NOVEL QCA ADDER:

The BCD adder here presented follows the traditional toplevel structure illustrated in Fig. 1, but it exploits the novel logic expressions demonstrated in the following by *Theorems* 1 and 2. As the main result, the proposed approach leads to the best tradeoff between the overall occupied area and the speed performances. To understand the newdesign strategy, let us examine first the 4-b binary adderADD1. It receives the digits $dA(3:0)$ and $dB(3:0)$ and the carry cin as inputs and computes the binary results bcout and $bS(3:0)$. [26, Lemma 4] demonstrates that, for QCA-based rippling adders, the optimal logic structure for propagating a carry Ci through a single bit position is represented by (2a) that introduces only one MG between Ci and $Ci+1$ $Ci+1 = M(dAi, dBi, Ci)$ (2a) $Ci+2 = M(Ci, M(dAi+1, dBi+1, gi), M(dAi+1, dBi+1, pi))$ (2b) Consequently, the propagation of Ci through two bit positions would require two cascaded MGs to obtain the carry $Ci+2$. Conversely, as discussed in [6] and given in (2b), by exploiting the auxiliary generate and propagate signals $gi = dAi \cdot dBi$ and $pi = dAi \cdot dBi$, the carry $Ci+2$ can be computed by propagating Ci through just one MG. *Theorem 1* demonstrates a novel way to propagate Ci through two consecutive bit positions that also introduces just one MG between Ci and $Ci+2$ but avoiding the computation of gi and pi.

Let us consider the consecutive bits $dAi+1$, dAi $dBi+1$, and dBi of the addends $dA(3:0)$ and $dB(3:0)$. IfCi is the carry signal inputted at the ith bit position, then the carry $Ci+2$ produced at the $(i + 1)$th bit position is given by $Ci+2 = M(Ci, M(dAi+1, dBi+1, dAi), M(dAi+1, dBi+1, dBi))$. *Proof:* By applying the conventional Carry-Look- Ahead (CLA) logic and considering that the propagate and generate signals $Pi+1$, Pi, $Gi+1$, and Gi are defined as

$Pi+1 = dAi + dBi+1$, $Pi = dAi + dBi$, $Gi+1 = dAi+1 \cdot dBi + 1$, and

$Gi = dAi \cdot dBi$, we have

$$Ci+2 = Gi+1 + Pi+1 \cdot Gi + Pi+1 \cdot Pi \cdot Ci$$
$$= Gi+1 + Pi+1 \cdot Gi + Pi+1 \cdot Pi \cdot Ci + Gi+1 \cdot (Ci + Pi+1 \cdot Pi)$$
$$= Gi+1 + Pi+1 \cdot dAi \cdot dBi + Pi+1 \cdot (dAi + dBi) \cdot Ci + Ci \cdot Gi+1 + Gi+1 \cdot Pi+1 \cdot (dAi + dBi)$$
$$= Ci \cdot (Gi+1 + Pi+1 \cdot dAi) + Ci \cdot (Gi+1 + Pi+1 \cdot dBi) + (Gi+1 + Pi+1 \cdot dAi) \cdot (Gi+1 + Pi+1 \cdot dBi)$$

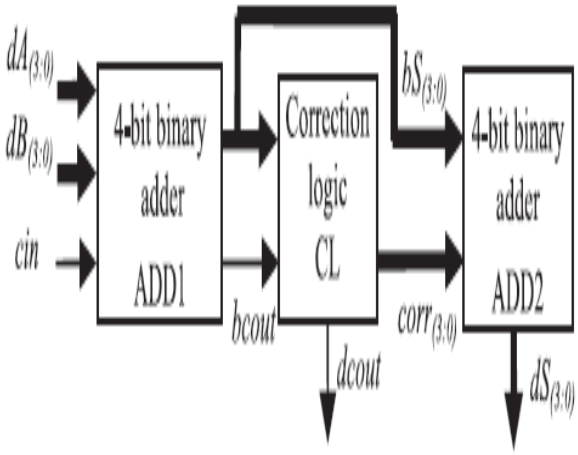


Fig. 1. Structure of the BCD adders

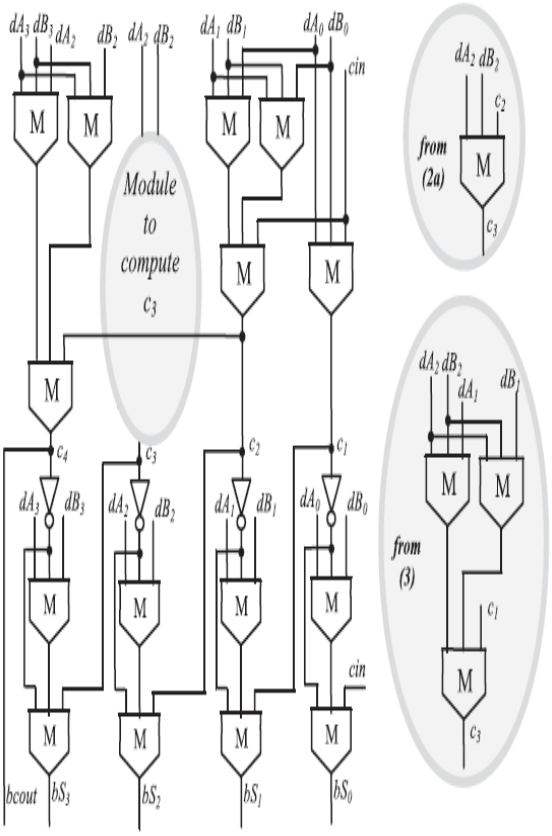


Fig. 2. New ADD1 module.

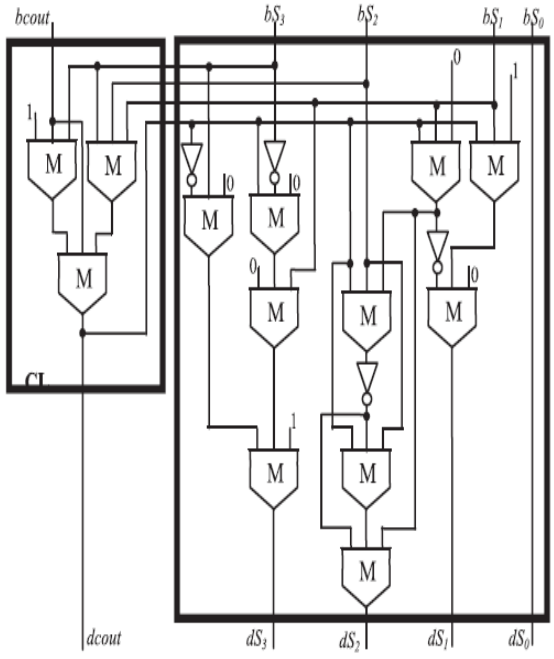
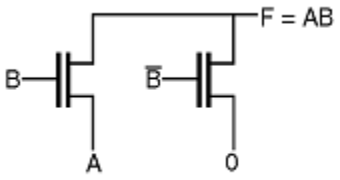


Fig. 3. Module CL and ADD2 of the new BCD adder.

Pass-Transistor-Logic

The Pass transistor logic is required to reduce the transistors for implementing logic by using the primary inputs to drive gate terminals, source and drain terminals. In complementary CMOS logic primary inputs are allowed to drive only gate terminals.

Figure below shows implementation of AND function using only NMOS pass transistors. In this gate if the B input is high the left NMOS is turned ON and copies the input A to the output F. When B is low the right NMOS pass transistor is turned ON and passes a '0' to the output F. This satisfies the truth table of AND gate reproduced in Table below for verification.



AND gate using pass transistor logic

In electronics, **pass transistor logic** (PTL) describes several logic families used in the design of integrated circuits. It reduces the count of transistors used to make different logic gates, by eliminating redundant transistors. Transistors are used as switches to pass logic levels between nodes of a circuit, instead of as switches connected directly to supply voltages.[1] This reduces the number of active devices, but has the disadvantage that the difference of the voltage between high and low logic levels decreases at each stage. Each transistor in series is less saturated at its output than at its input.[2] If several devices are chained in series in a logic path, a conventionally constructed gate may be required to restore the signal voltage to the full value. By contrast, conventional CMOS logic switches transistors so the output connects to one of the power supply rails, so logic voltage levels in a sequential chain do not decrease. Simulation of circuits may be required to ensure adequate performance..

The major advantage of pass transistor logic is that fewer transistors are required to implement a given function. To illustrate this consider the implementation of AND gate using complementary CMOS logic. If we compare this with the same AND gate implementation using pass transistor logic the number of transistors required are four including the two transistor required to invert the input B. The another advantage of pass transistor logic is the lower capacitance because of reduced number of transistors. As discussed NMOS evices are effective in passing strong '0' but it is poor at pulling a node to VDD. Hence when the pass transistor pulls a node to high

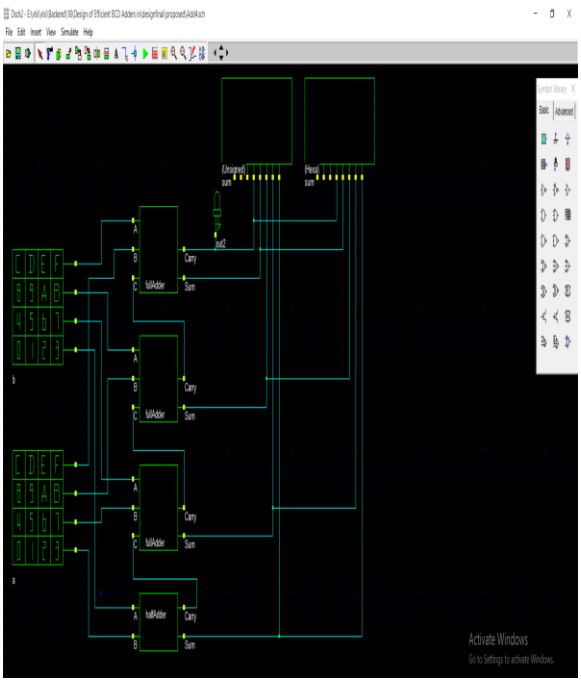
logic the output only changes upto VDD–VTh. This is the major disadvantage of pass transistors.

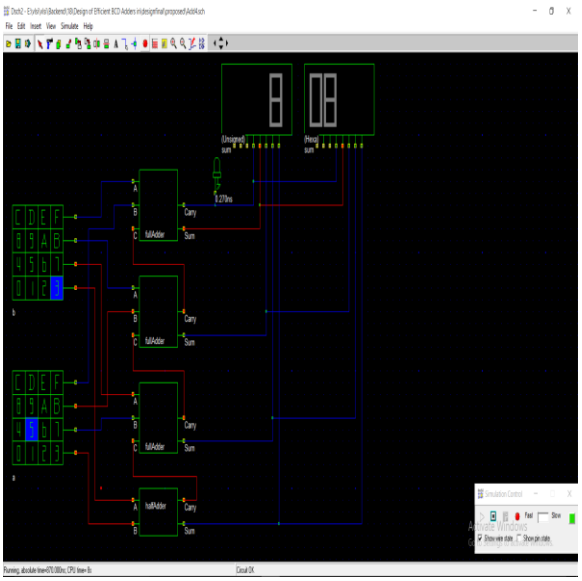
A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

Truth table of AND gate

Pass transistor logic often uses fewer transistors, runs faster, and requires less power than the same function implemented with the same transistors in fully complementary CMOS logic.[3] XOR has the worst-case Karnaugh map -- if implemented from simple gates, it requires more transistors than any other function. The designers of the Z80 and many other chips saved a few transistors by implementing the XOR using pass-transistor logic rather than simple gates The pass transistor is driven by a periodic clock signal and acts as an access switch to either charge up or charge down the parasitic capacitance C_x, depending on the input signal V_{in}. Thus, two possible operations when the clock signal is active (CK = 1) are the logic "1" transfer (charging up the capacitance C_x to a logic-high level) and the logic "0" transfer (charging down the capacitance C_x to a logic-low level). In either case, the output of the depletion load nMOS inverter obviously assumes a logic-low or a logic-high level, depending upon the voltage V_x. Some authors use the term "complementary pass transistor logic" to indicate a style of implementing logic gates that uses transmission gates composed of both NMOS and PMOS pass transistors.[5]

RESULT:





CONCUSION:

The design is very useful for the future computing techniques like low power digital circuits and advanced computers. A new design approach has been presented and demonstrated to achieve efficient QCA-based implementations of decimal adders. Unconventional logic formulations and purposedesigned logic modules here proposed allow outperforming decimal adders. The design method is definitely useful for the construction of future computer and other computational structures. Finally, by exploiting the Pass Transistor Through Logic, a more feasible implementation of the new adder has been obtained without compromising the advantages achieved over its direct competitors.

REFERENCES:

[1] Yvan Van Rentergem and Alexix De Vos, (2005) "Optimal design of a reversible full adder" Int. Journ. Of Unconventional Computing, Vol. 1, pp. 339-355. [2] Landauer R., (1961) "Irreversibility and heat generation in the computing process". IBM J. Research and Development, 5(3): 183-191. [3] Parhami B., (2006) "Fault tolerant reversible circuits". Proc. 40th Asilomar Conf. Signals, Systems, and Computers, October, Pacific Grove, CA. [4] Bennett C. H., (1973) "Logical reversibility of computation". IBM J. Research and Development, 17: 525-532. [5] H.R.Bhagyalakshmi, M.K.Venkatesha, (2010) "Optimized reversible BCD adder using new

reversible logic gates", JOURNAL OF COMPUTING, VOLUME 2, ISSUE 2, ISSN 2151-9617
HTTPS://SITES.GOOGLE.COM/SITE/JOURNALOFCOMPUTING/ [6] X. Susan Christina, M.Sangeetha Justine, K.Rekha, U.Subha and R.Sumathi, (2010) "Realization of BCD adder using Reversible Logic", International Journal of Computer Theory and Engineering, Vol. 2, No. 3, 1793-8201 [7] Hayes B., (2006) "Reverse engineering. American Scientist", 94: 107-111.
[8] Fredkin E. and T. Toffoli, (1982) "Conservative logic". Int'l J. Theoretical Physics, 21: 219-253. [9] R. Feynman, (1985) "Quantum Mechanical Computers", Optics News, 11, pp. 11-20. [10] DAMU RADHAKRISHNAN, STERLING R. WHITAKER, and GARY K. MAKI, (1985) "Formal Design Procedures for Pass Transistor Switching Circuits", IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. SC-20, NO.2. [11] D. Markovic, B. Nikolic and V.G. Oklobdzija, (2000) "A general method in synthesis of passtransistor circuits", Microelectronics Journal 31 991-998. [12] Okihiko Ishuzuka, (1986) "Synthesis of a Pass Transistor Network Applied to Multi-Valued Logic", IEEE Trans. [13] Hafiz Md. Hasan Babu, Moinul Islam Zaber, Md. Mazder Rahman and Md. Rafiquil Islam, "Implementation of Multiple-Valued Flip-Flips Using Pass Transistor Logic", Proceedings of the EUROMICRO Systems on Digital System Design (DSD'04).