International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878, Impact Factor (2014): 3.05

Area-Delay Efficient Modified Majority Gate Binary Adders in Quantum-Dot Cellular Automata

S. Preetha

PG Scholar, ECE, PGP College of Engineering and Technology, Namakkal, TamilNadu, India

Abstract: ***As transistors decrease in size more and more of them can be accommodated in a single die, thus increasing chip computational capabilities. However, transistors cannot get much smaller than their current size. The QCA approach represents one of the possible solutions in overcoming this physical limit.Quantum-dot cellular automata (QCA) is considered as an advanced technology compared to complimentary metal-oxide-semiconductor (CMOS) due to QCA's merits. Many logical circuits are designed using QCA which consume low power. Adders are the basic building block of digital circuits. Initially transistors are used to implement the circuits in digital systems.Nanotechnology is the better alternative to these problems. So in this project I use Quantum-dot Cellular Automata which is an emerging nanotechnology. This technology has the potential for faster speed, smaller size and lower power consumption. In Existing System, an adder is designed which runs in RCA Fasion. In this paper, I propose a new adder design by reducing majority gate and implementing this in the CLA adder. Thus, the reduction of majority gates in the proposed system causes low complexity compared to that of existing system which also causes area and power reduction of about 20% than the existing adder.***

Keywords: Adders, majority gate (MG), Verilog HDL, quantum-dot cellular automata (QCA).

1. Introduction

A Quantum-dot cellular automata (QCA) is an attractive emerging technology suitable for the development of ultra-dense low-power high-performance digital circuits. The design of efficient logic circuits in QCA has received a great deal of attention. Special efforts are directed to arithmetic circuits, with the main interest focused on the binary addition that is the basic operation of any digital system. The architectures commonly employed in traditional CMOS designs are considered a first reference for the new design environment. Ripple-carry (RCA), carry look-ahead (CLA), and conditional sum adders were presented.

An innovative technique is presented to implement high-speed low-area adders in QCA. Theoretical formulations demonstrated for CLA adders are here exploited for the realization of a novel 2-bit addition slice. The latter allows the carry to be propagated through two subsequent bit-positions with the delay of just one majority gate (MG). In addition, the clever top level architecture leads to very compact layouts, thus avoiding unnecessary clock phases due to long interconnections. An adder designed as proposed runs in the RCA fashion, but it exhibits a computational delay lower than all state-of the- art competitors and achieves the lowest area-delay product (ADP).

QCA which employs array of coupled quantum dots to implement Boolean logic function. The advantage of QCA lies in the extremely high packing densities possible due to the small size of the dots, the simplified interconnection, and the extremely low power delay product. A basic QCA cell consists of four quantum dots in a square array coupled by tunnel barriers. Electrons are able to tunnel between the dots, but cannot leave the cell. If two excess electrons are placed in the cell, Coulomb repulsion will force the electrons to dots on opposite corners. There are thus two energetically equivalent ground state polarizations can be labeled logic "0"and "1".The basic building blocks of the QCA architecture are AND,OR and NOT. By using the Majority gate we can reduce the amount of delay by calculating the propagation and generational carries.

2. QCA

Quantum Dot Cellular Automata (sometimes referred to simply as quantum cellular automata, or QCA) are proposed models of quantum computation, which have been devised in analogy to conventional models of cellular automata introduced by von Neumann. Standard solid state QCA cell design considers the distance between quantum dots to be about 20 nm, and a distance between cells of about 60 nm. Just like any CA, Quantum (-dot) Cellular Automata are based on the simple interaction rules between cells placed on a grid. A QCA cell is constructed from four quantum dots arranged in a square pattern. These quantum dots are sites electrons can occupy by tunneling to them.

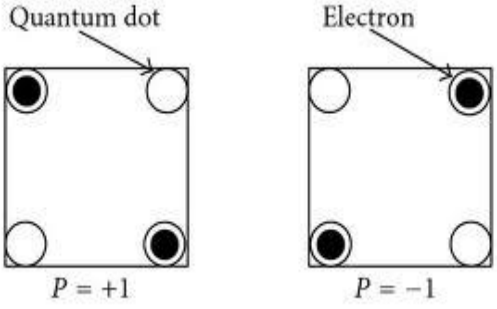

Figure 1: Cell Design

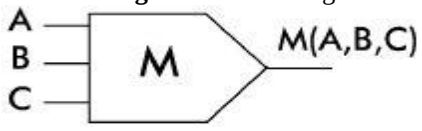

Figure 2: Structure of Majority gate

The QCA majority gate performs a three-input logic function. Assuming the inputs are A ,B and C, the logic function of the majority gate is

M =AB+BC+CA

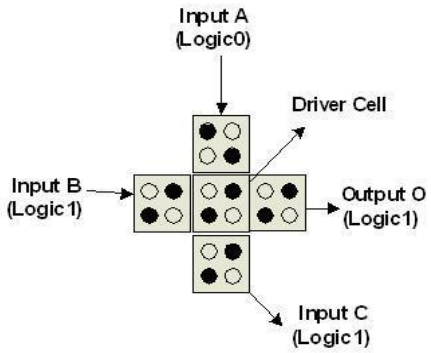

Figure 3: QCA Majority Gate

3. Ripple-Carry Adder

The ripple carry adder is constructed by cascading full adders (FA) blocks in series. One full adder is responsible for the addition of two binary digits at any stage of the ripple carry. The carryout of one stage is fed directly to the carry-in of the next stage.

A number of full adders may be added to the ripple carry adder or ripple carry adders of different sizes may be cascaded in order to accommodate binary vector strings of larger sizes. For an n-bit parallel adder, it requires n computational elements (FA). It is composed of four full adders. The augend's bits of x are added to the addend bits of y respectfully of their binary position. Each bit 6 addition creates a sum and a carry out.

The carry out is then transmitted to the carry in of the next higher-order bit. The final result creates a sum of four bits plus a carry out (c4).

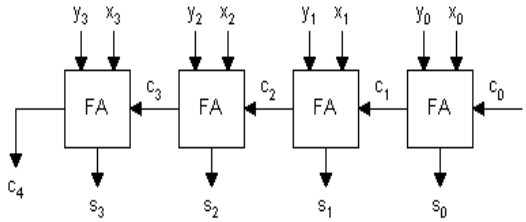

Figure 4: Ripple Carry Adder

Even though this is a simple adder and can be used to add unrestricted bit length numbers, it is however not very efficient when large bit numbers are used. One of the most serious drawbacks of this adder is that the delay increases linearly with the bit length. As mentioned before, each full adder has to wait for the carry out of the previous stage to output steady-state result. Therefore even if the adder has a value at its output terminal, it has to wait for the propagation of the carry before the output reaches a correct value. Excluding a serial adder, which consists of just a full adder and 1-bit storage, the simplest possible adder is the carry-ripple. The design of an n-bit ripple adder that takes two operands,

A= An-1An-2An-3...A0, B= Bn-1Bn-2Bn-3...B0 and produces a sum of S= Sn-1Sn-2Sn 3...S0. Ci-1 is the carry into the adder and is usually 0 for unsigned arithmetic; Cn-1 is the carry out of the adder. The logical equations are

Si= (Ai^Bi)^Ci-1

Ci=AiBi + (Ai^Bi) ^ Ci-1

If performance is measured in terms of logical date-delays, then the serial adder appears to be rather slow, because the full adders cannot always operate in parallel. In general, the full adder at stage ihas to wait for a possible carry from stage i-1, which in turn has to wait for a possible carry from stage i-2, and so forth. The operational time is therefore O(n), in contrast with the O(log n) of the fastest adders.

4. Regular QCA Architecture

The existing system introduces the Quantum-dot Cellular Automata (QCA) approach which presents a new adder that outperforms all state-of-the-art competitors and achieves the best area-delay trade off. A quantum-dot cellular automaton (QCA) is an attractive emerging technology suitable for the development of ultra-dense low-power high-performance digital circuits. For this reason, in the last few years, the design of efficient logic circuits in QCA has received a great deal of attention. Special efforts are directed to arithmetic circuits An adder designed in the existing system runs in the RCA fashion, but it exhibits a computational delay lower than all state-of-the-art competitors and achieves the lowest area-delay product (ADP).

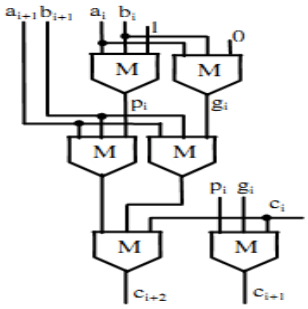

Figure 5: Novel 2-Bit Basic Module

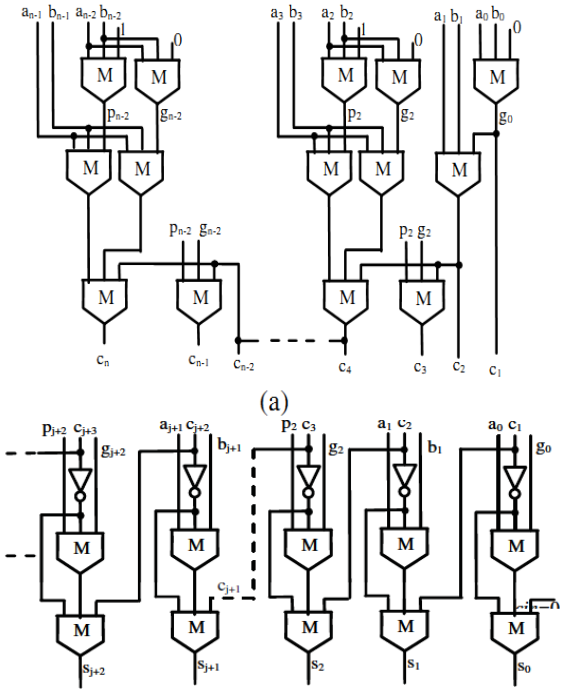

Figure 6: Novel *n*-bit adder (a) carry chain and (b) sum block.

5. Carry Look Ahead Adder

Carry-look aheadis the most important technique in the design of fast adders, especially large ones. In straightforward addition, e.g. in a ripple adder, the operational time is limited by the (worst-case) time allowed for the propagation of carries and is proportional to the number of bits added. So faster adders can be obtained by devising a way to determine carries before they are required to form the sum bits. Carry-look ahead does just this, and, in certain cases the resulting adders have an operational time that is independent of the operands word-length. A carry, Ci, is produced at bit-stage i if either one is generated at that stage or if one is propagated from the preceding stage. So a carry is generated if both operand bits are 1, and an incoming carry is propagated if one of the operand bits is 1 and the other is 0. Let Pi and Gidenote the generation and propagation, respectively, of a carry at stage i, Ai and Bi denote the two operands bits at that stage, and Ci-1 denote the carry into the stage. Then we have

Gi=AiBi

Pi = Ai^Bi

Ci = Gi + PiCi-1

and the sum can be written as Si== Pi^Ci-1 which allows the use of shared logic to produce Si and Pi.

C0 = G0 + P0Ci-1

C1 = G1 + P1P0C-1+ P1G0

.

.

.

Ci = Gi + Pi-1Gi-1+PiPi-1Gi-2+…+PiPi-1Pi-2…P0C-1

where Ci-1 is the carry into the adder. The equation for Ci states that there is a carry from stage i if there is a carry generated at stage i, or if there is a carry that is generated at stage i-1 and propagated through stage i or if , or if the initial carry-in, Ci-1, is propagated through stages 0,1,… i. The complete set, of equations show that, in theory at least, all the carries can be determined independently, in parallel, and in a time (three gate delays) that is independent of the number of bits to be added. The same is also therefore true for all the sum bits, which require only one additional gate delay.

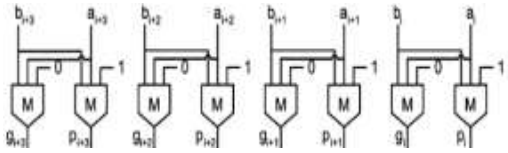

Figure 7: Generation Of Propagate and Generate Bits

Gi = AiBi

Pi = Ai^Bi

Compared with a ripple adder, as well as some of the other adders, a pure carry-look ahead adder has high logic costs. Furthermore, high fan-in and fan-out requirements can be problematic: the fan-out required of the Gi and Pi signals grows rapidly with n, as does the fan-in required to form Ci. For sufficiently large values of n, the high fan- in and fan-out requirements will result in low performance, high cost, or designs that simply cannot be realized.

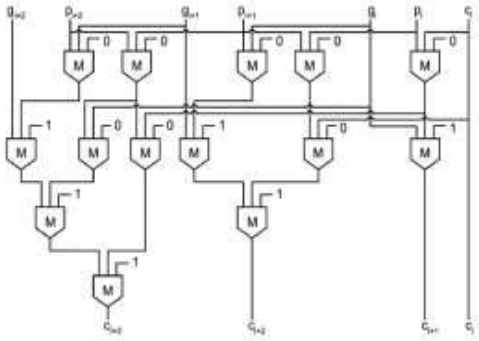

Figure 8: Carry Block

This carry block is cascaded with the propagate and generate block. So, that carry is obtained with thefollowing equation.

Ci = Gi + PiCi-1

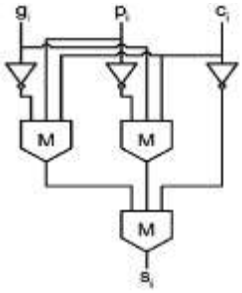

Figure 9: Sum Block

This sum block is cascaded with the above carry block to obtain the sum. The following equation gives the sum bit

Si = Pi ^ Ci-1.

6. Modified QCA Architecture

In Proposed System, majority logic is reduced for the reduction of complexity.Given three binary inputs, a, b, and c, the majority voting logic function can be expressed in terms of fundamental Boolean operators M(a, b, c)= ab + bc + ac.

Rule 1: If a, b and c are three binary inputs, then M(a, b, M(a, b, c)) = M(a, b, c).

Rule 2: Let a, b and c be three binary inputs. Then

$$M(a, b, \overline{M(a, b, c)}) = M(a, b, \overline{c}).$$

Rule3: Letf1,f2, and f3 be three Boolean functions such that f1 and f2 satisfy f1 f2 = f1 and f1 + f2 = f2. Then M(f1 ,f2 ,f3)= f1 + f2 f3.

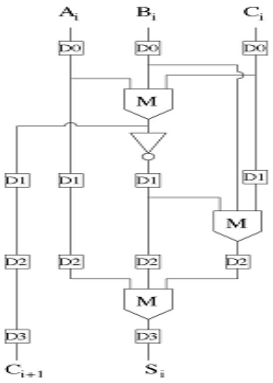

Figure 10: Full Adder Realization Using Three Majority Gates and One Inverter; Numbered D-Latches Enable Delay Determination

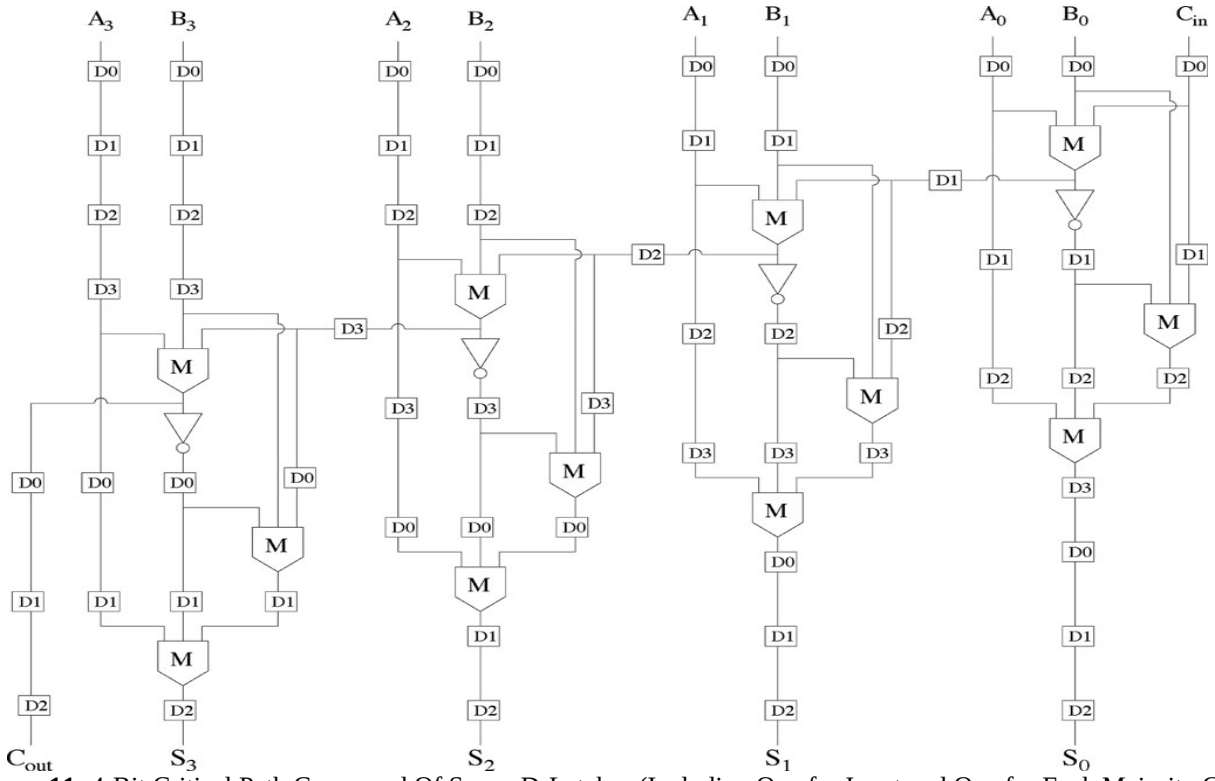

Figure 11: 4-Bit Critical Path Composed Of Seven D-Latches (Including One for Input and One for Each Majority Gate

7. Conclusion

A new adder designed in QCA was presented. It achieved speed performances higher than all the existing QCA adders, with an area requirement comparable with the cheap RCA and CFA demonstrated in.The novel adder operated in the RCA fashion, but it could propagate a carry signal through a number of cascaded MGs significantly lower than conventional RCA adders. In addition, because of the adopted basic logic and layout strategy, the number of clock cycles required for completing the elaboration was limited. A 128-bit binary adder designed as described in this brief exhibited a delay of only seventeen clock cycles, occupied an active area of 32.25 μm2, andachieved an ADP of only 548.25.

8. Acknowledgement

My wishes to acknowledge Mrs.P.VINITHA mam, Lecturer who guided us for the conference paper preparation. Thank you mam.

References

[1] StefaniaPerri, Pasquale Corsonello, andGiuseppe Cocorullo, “Area-Delay EfficientBinary Adders in QCA”, 2013 IeeeTransactions On Very Large ScaleIntegration (Vlsi) Systems
[2] C. S. Lent, P. D. Tougaw, W. Porod, and G.H. Bernestein, “Quantum cellularautomata,” Nanotechnology, vol. 4, no. 1,pp. 49–57, 1993
[3] H. Cho and E. E. Swartzlander, “Adderdesign and analyses for quantum-dotcellular automata,” IEEE Trans.Nanotechnol., vol. 6, no. 3, pp. 374–383,May 2007
[4] H. Cho and E. E. Swartzlander, “Adder andmultiplier design in quantum-dot cellularautomata,” IEEE Trans. Comput., vol. 58,no. 6, pp. 721–727, Jun. 2009
[5] W. Liu, L. Lu, M. O’Neill, and E. ESwartzlander, Jr., “Design rules for quantum-dot cellular automata,” in Proc. IEEE Int. Symp. Circuits Syst., May 2011, pp. 2361–2364