IMPLEMENTATION OF VLSI ARCHITECTURE OF MAC BINARY ADDER IN QCA

Ms. Thahseen Thahir
Assistant Professor, ECE
JCT College of Engineering & Technology
Pichanur, Coimbatore, Tamil Nadu
thahseenthahir@gmail.com

Ms.B.Vani
Assistant Professor,ECE
JCT College of Engineering & Technology
Pichanur, Coimbatore, Tamil Nadu
vanibalu16@gmail.com

Abstract— As transistors decrease in size a lot of and a lot of of them may be accommodated in an exceedingly single die, so increasing chip machine capabilities. However, transistors cannot get a lot of smaller than their current size. The quantum-dot cellular automata (QCA) approach represents one in every of the attainable solutions in overcoming this physical limit, despite the fact that the look of logic modules in QCA isn't forever easy. during this transient, we have a tendency to propose a replacement adder that outperforms all state-of-the- art competitors and achieves the most effective area-delay trade off. The on top of benefits ar obtained by victimization AN overall space just like the cheaper styles acknowledged in literature. The 64-bit version of the novel adder spans over eighteen.72 µm2 of active space and shows a delay of solely 9 clock cycles, that's simply thirty six clock phases.

Index Terms—Adders, nano computing, quantum-dot cellular automata (QCA).

1. INTRODUCTION

Dimensional scaling of field-effect transistors (FETs) used in logic circuits is approaching its fundamental limit. Nowadays, CMOS logic circuits suffer from short channel, variability and quantum effects that cause severe degradation of FETs operations. Furthermore, power density is significantly increasing. To continue the miniaturization trend and to sustain cost reduction, researchers are investigating several alternative information processing devices and micro-architectures. Current transistor-based semiconductor devices are becoming resistant to scaling. Due to the decreasing supply voltage, the power consumption from leakage current is a big challenge for transistor circuits. Nanotechnology is a possible alternative to these problems. Quantum-dot cellular automata (QCA) are 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, with the main interest focused on the binary addition that is the basic operation of any digital system.

Of course, 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. The carry-flow adder (CFA) was mainly an improved RCA in which detrimental wires effects were mitigated. Parallel-prefix architectures, including Brent–Kung (BKA), Kogge–Stone, Ladner–Fischer, and Han–Carlson adders, were analyzed and implemented in QCA. Recently, Hybrid adder was also introduced which is the existing method.

In this brief, an innovative technique is presented to implement high-speed low-area adders in QCA. Theoretical formulations demonstrated for CLA and parallel-prefix 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 consumes number of majority gates lesser than all state-of-the-art competitors and achieves the lowest area-power product. This proposed work is implemented into a Multiplier Accumulator (MAC) unit. In the design MAC (multiply accumulate) binary adder, the circuit is operational to perform a MAC (multiply accumulate) operation and to perform a multiply operation without interfacing with the accumulate value of MAC operation using Quantum-dot cellular automata (QCA) and compares the area and power with existing adder.

2. BACKGROUND

The previous work was a Hybrid adder using QCA technology. Hybrid adder circuitry is provided for IC such as programmable integrated circuits. Generally hybrid adder is the combination of the capabilities of multiple adder architecture. In previous work, a hybrid adder is the combination of Ladner–Fischer prefix adder and Ripple Carry Adder (RCA). Adder functionality may be implemented using the resources of logic regions on the programmable IC. Each logic regions may include combinational logic such as look-up table and register circuitry. The hybrid adder circuitry may receive input words to be added from the combinational circuitry and may produce corresponding arithmetic sum output signal to the register circuitry.

3. PROPOSED SYSTEM

The proposed work is a QCA adder which achieves speed performances higher than the existing hybrid adder. Also the area of the adder will be reduced when compared. Hence power consumption will be reduced. This proposed work is implemented to a Multiplier Accumulator (MAC) unit as an application level.

Quantum-dot Cellular Automata (QCA) is an attractive emerging nanotechnology suitable for the development of ultra dense low power high performance digital circuits. QCA has a single cell as the basic element. The cell is used as a building block to construct gates and wires. It is introduced by Lent.et.al and he combined the discrete nature of both cellular automata

and quantum mechanics to create nano-scale devices capable of performing computation at very high switching speeds and consuming extremely small amounts of electrical power.

3.1 QCA BASICS

QCA cells with electrons indicating possible polarizations are given in below Fig.1.

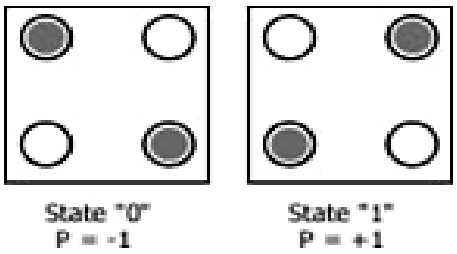

Fig.1. QCA cells with electrons indicating possible polarizations.

Primitives in the QCA model consist of a wire, inverter and majority gate and are depicted in Fig.2. A majority gate in QCA takes three inputs and implements the majority function of three Boolean variables a, b, and c. The majority function, denoted by M(a, b, c), is defined as

$$M(a, b, c) = a.b + b.c + a.c \quad (1)$$

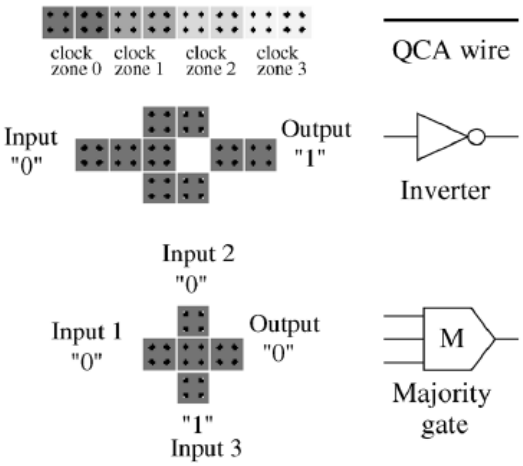

Fig.2. QCA wire, inverter, and majority gate

3.2 EXISTING WORK

A 16-bit Hybrid adder that combines Ladner–Fischer adder and the Ripple Carry Adder is performed as the existing work. While the Ladner–Fischer adder supports parallelism, the requirement of majority gates (which contributes to the overall area) is quite high. The large number of majority gates has an indirect effect on the wire (delay and amount). It is therefore of interest to explore ways of reducing the area.

From the literature, it is known that ripple carry adders are simple and have low area requirement. This fact is taken advantage of in our design of a hybrid adder. This hybrid adder has advantages especially in the QCA domain over the Ladner–Fischer adder as well as the ripple carry adder in terms of delay. Further, the hybrid adder requires a substantially lower number of majority gates for different adder sizes in comparison to a Ladner–Fischer adder. Fig.3. illustrates the generation of carries of a 16-bit hybrid adder.

The calculation of the remaining carries, namely C_i, i= 1, 2, 3, 4, 5, 6, 9, 10, 13, 14 can be done in ripple-carry style (each C_i can be expressed in terms of M (x_{i-1},y_{i-1},c_{i-1}). For example, carry C_5 can be calculated as M(x_4,y_4,c_4). So ten majority gates are required for these ten carries. The requirements for sum are the same as for the Ladner–Fischer adder. In particular, 32 majority gates, and 16 inverters are required for a 16-bit hybrid adder. The total requirements are summarized as A 16-bit hybrid adder requires 86 majority gates and 16 inverters.

In Fig.4, some of the carries computed in ripple carry style are shown enclosed in ellipses (and labeled with $). For example, C_5 and C_6 in Stage 3 are computed in ripple carry style with one majority gate (delay) difference between C_5 and C_6. However, C_7 and C_8 are computed in prefix style and in parallel with C_6. Similarly, in Stage 4, C_9 and C_{10} are computed in ripple carry style but C_{10} is computed in parallel with C_{11} and C_{12} (the latter are obtained in prefix style).

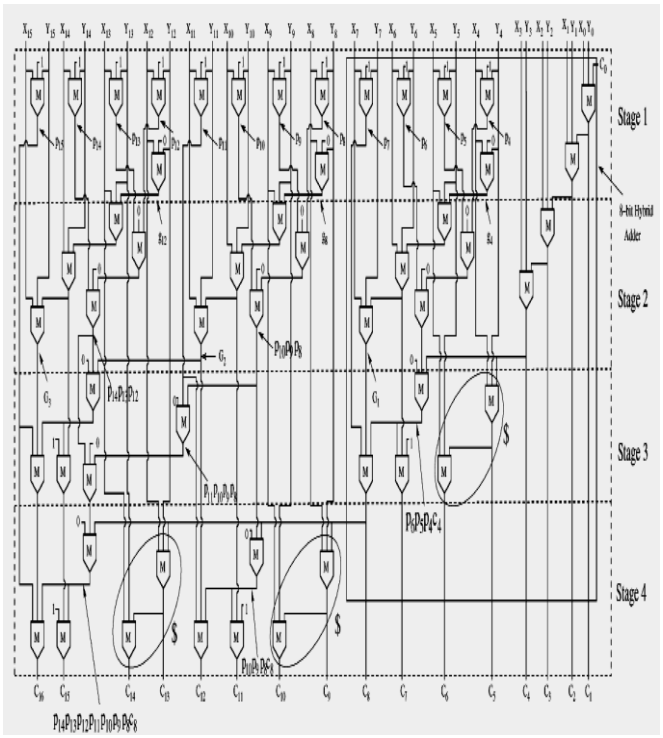

Fig.3. Carry generation for a 16-bit hybrid adder.

3.3 DRAWBACKS OF EXISTING WORK

Hybrid adder combines Ladner–Fischer adder and the ripple carry adder which provides reduction of majority gates when compared to Ladner-Fischer. Number of Majority gates used (area consumed) by Hybrid adder is more. Hence complexity is more. And also the power consumed by this design is also more. To overcome this problem, a new adder is proposed which consumes less power and uses less number of gates (area).

4. METHODOLOGY AND PROCEDURE

4.1 NOVEL QCA ADDER

To introduce the novel architecture proposed for implementing ripple adders in QCA, let consider two n-bit addends A = $a_{n-1}, \ldots, a_0$ and B = $b_{n-1}, \ldots, b_0$ and suppose that for the i^{th} bit position (with i = n − 1, . . . , 0) the auxiliary propagate and generate signals, namely $p_i = a_i + b_i$ and $g_i = a_i \cdot b_i$, are computed. c_i being the carry produced at the generic $(i-1)^{th}$ bit position, the carry signal c_{i+2}, furnished at the $(i+1)^{th}$ bit position, can be computed using the conventional CLA logic reported in (2). The latter can be rewritten as given in (3), by exploiting Theorems 1 and 2. In this way, the RCA action, needed to propagate the carry c_i through the two subsequent bit positions, requires only one MG. Conversely, conventional circuits operating in the RCA fashion, require two cascaded MGs to perform the same operation. In other words, an RCA adder designed as proposed has a worst case path almost halved with respect to the existing method.

QCA cells are used for both logic structures and interconnections that can exploit either the coplanar cross or the bridge technique. The fundamental logic gates inherently available within the QCA technology are the inverter and the MG. Given three inputs a, b, and c, the MG performs the logic function reported in (1) provided that all input cells are associated to the same clock signal clk_x (with *x* ranging from 0 to 3), whereas the remaining cells of the MG are associated to the clock signal clk_{x+1}.

Equation (1) is exploited in the design of the novel 2-bit module shown in Fig.4. that also shows the computation of the carry $c_{i+1} = M(p_i\ g_i\ c_i)$.

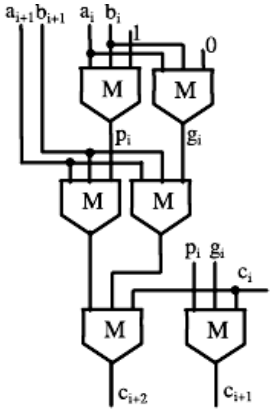

Fig.4. Novel 2-bit basic module.

Several designs of adders in QCA exist in literature. Even though these addition circuits use different topologies of the generic FA, they have a carry-in to carry-out path consisting of one MG, and a carry-in to sum bit path containing two MGs plus one inverter. As a consequence, the worst case computational paths of the *n*-bit RCA consist of (*n*+2) MGs and one inverter. With the main objective of trading off area and delay, the hybrid adder (HYBA) combines a parallel-prefix adder with the RCA. In the presence of *n*-bit operands, this architecture has a worst computational path consisting of 2×log2 *n* +2 cascaded MGs and one inverter.

The proposed *n*-bit adder is then implemented by cascading *n*/2 2-bit modules as shown in Fig.5 (a). Having assumed that the carry-in of the adder is *cin* = 0, the signal *p*0 is not required and the 2-bit module used at the least significant bit position is simplified. The sum bits are finally computed as shown in Fig.5 (b).

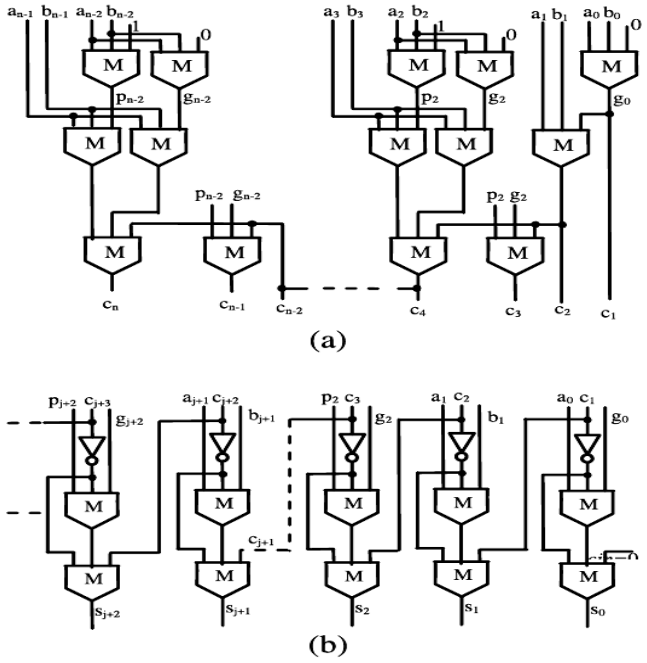

Fig.5. Novel *n*-bit adder (a) carry chain and (b) sum block.

It must be noted that the time critical addition is performed when a carry is generated at the least significant bit position (i.e., g0 = 1) and then it is propagated through the subsequent bit positions to the most significant one. In this case, the first 2-bit module computes c2, contributing to the worst case computational path with two cascaded MGs. The subsequent 2-bit modules contribute with only one MG each, thus introducing a total number of cascaded MGs equal to (n − 2)/2. Considering that further two MGs and one inverter are required to compute the sum bits, the worst case path of the novel adder consists of (n/2) + 3 MGs and one inverter.

$$c_{i+2} = g_{i+1} + p_{i+1} \cdot g_i + p_{i+1} \cdot p_i \cdot c_i \quad (2)$$

$$c_{i+2} = M(M(a_{i+1}, b_{i+1}, g_i)M(a_{i+1}, b_{i+1}, p_i)\ c_i) \quad (3)$$

4.2 MULTIPLIER ACCUMULATOR (MAC) UNIT

In proposed to describe about MAC, the circuit is operational to perform a operation and to perform a multiply operation without interfacing with the accumulate value of MAC operation using QCA. The circuit includes a first register, a second register, a multiplier circuit, and an accumulate circuit. The first register is addressable using either a primary first address or an alias second address. The circuit performs multiply operation to generate the product value based on data in the first and second register after a

write operation to first register or second register. The MAC circuit must check for overflow, which might happen when the number of MAC operations is large. Overflow in a signed adder occurs when two operands with the same sign produce a result with a different sign. Fig.6. shows the structure of the proposed MAC unit. The MAC unit consists of the multiplier and an accumulator unit. The multiplier array, that consist of multiplexers, half adders, full adders and the add cell (to add 0 or 1 to the LSB of the partial products) 1. The QCA adder in the adder array is used here.

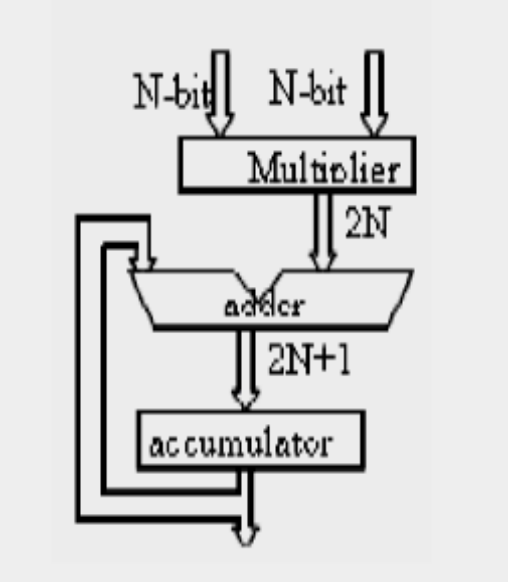

Fig.6. Basic structure of MAC

The accumulator performance can greatly increase the performance of the MAC unit. But the bottleneck for the accumulator is the multiplier unit. The accumulator must wait till it has correct logic values at its input for accumulation. The accumulator unit must comply with our initial objective of low power and high performance.

5. RESULTS AND DISCUSSION

5.1 SIMULATION ENVIRONMENT

The simulation parameters have been analyzed with the help of Xilinx ISE and Modelsim for verification.

5.1.1 Xilinx ISE

Xilinx ISE (Integrated Software Environment) is a software tool produced by Xilinx for synthesis and analysis of HDL designs, enabling the developer to synthesize ("compile") their designs, perform timing analysis, examine RTL diagrams, simulate a design's reaction to different stimuli, and configure the target device with the programmer. It provides synthesis and programming for a limited number of Xilinx devices. In particular, devices with a large number of I/O pins and large gate matrices are disabled. The low-cost Spartan family of FPGAs is fully supported by this edition, as well as the family of CPLDs, meaning small developers and educational institutions have no overheads from the cost of development software.

5.1.2 Modelsim

ModelSim is a widely-used logic simulation tool for verification and debugging of digital circuits. It is mainly used for programming VLSI ASICs, FPGAs, CPLDs, and SoCs. Altera provides a version of ModelSim software, which includes libraries for Altera's FPG Users of the Quartus II web edition software, should also download the ModelSim package. ModelSim advanced code coverage capabilities provide valuable metrics for systematic verification. ModelSim combines simulation performance and capacity with the code coverage and debugging capabilities required to simulate multiple blocks and systems and attain ASIC gate-level sign-off. ModelSim eases the process of finding design defects with an intelligently engineered debug environment. Its debug environment efficiently displays design data for analysis and debug of all languages.

5.1.3 Waveform analysis

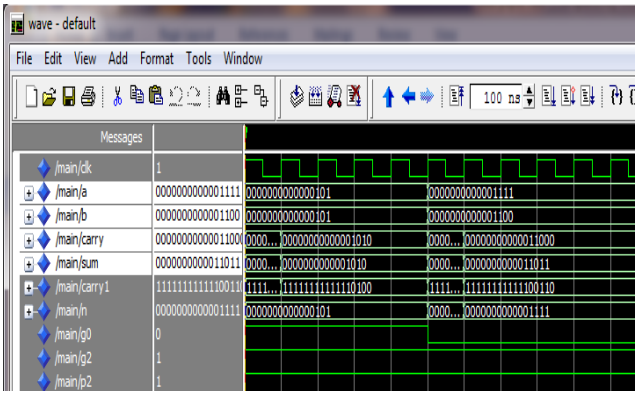

Fig.7. Waveform for both Hybrid adder and binary QCA adder

5.1.4 Waveform analysis

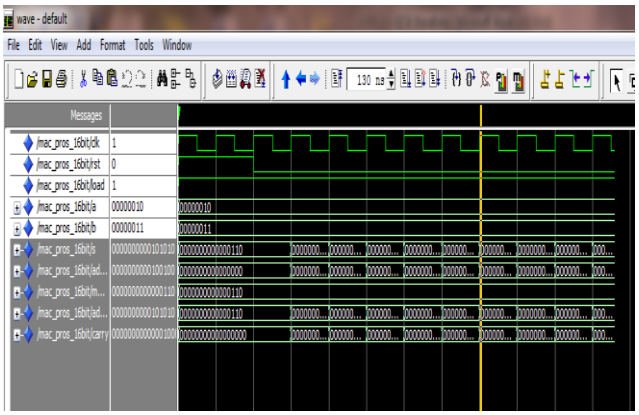

Fig.8. Waveform for binary QCA adder in MAC

The simulation waveform of both Hybrid adder and Binary QCA adder is shown in Fig.7. and Fig.8. shows the simulation waveform of binary QCA adder in MAC.

5.2 ANALYSIS

An efficient binary adder using QCA design is presented and compared with the Hybrid adder of Ladner-Fischer and the Ripple Carry adder. These designs are mainly based on new results concerning majority logic. Simulations have been done using Modelsim SE 6.3f and Xilinx ISE 8.1i tools and values have been recorded. The binary adder shows better performance than the hybrid adder and also have lesser area and consumes less power. Comparison of area and power consumed by the Hybrid adder and the Binary QCA adder is shown in Table.1.

Table 1. Comparison of Area and Power with existing adder

S.NO	Adders (16-bit)	Area (No.of gate counts)	Power (mW)
1	Hybrid adder	896	37
2	Binary QCA adder	217	32

5.3 HARDWARE

Hardware details of FPGA kit is given in below table

Table.2. Hardware details of FPGA kit

FPGA kit used is XC3S100E TQ144	
Device	XC3S100E
System Gates	100K
Equivalent Logic Cells	2,160
CLB Array (One CLB = Four Slices)	
Rows	22
Columns	16
Total CLBs	240
Total Slices	960
Distributed RAM bits(1)	15K
Block RAM bits(1)	72K
Dedicated Multipliers	4
DCMs	2
Maximum User I/O	108
Maximum Differential I/O Pairs	40
Package	TQ144

6. CONCLUSION

A new 16-bit adder designed in QCA was presented. This novel adder is operated in RCA fashion, but it could propagate a carry signal through a number of cascaded MGs significantly lower than existing adders. This efficient binary adder using QCA design is compared the area and the power consumed with the Hybrid adder of Ladner-Fischer and the Ripple Carry adder and observed that the area and the power consumed is reduced. Finally proposed QCA adder is implemented into the MAC unit as an application level for arithmetic operations.

REFERENCES

[1] Stefania Perri, Pasquale Corsonello, and Giuseppe Cocorullo, "Area-Delay Efficient Binary Adders in QCA", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, VOL. 22, NO. 5, MAY 2014.

[2] Bhuvaneswaran M, Shankar N.K., "Efficient Design of MAC Hybrid Adder in Quantum-Dot Cellular Automata", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering Vol. 2, Issue 4, April 2013.

[3] C. S. Lent, P. D. Tougaw, W. Porod, and G. H. Bernestein, "Quantum cellular automata," Nanotechnology, vol. 4, no. 1, pp. 49–57, 1993.

[4] H. Cho and E. E. Swartzlander, "Adder design and analyses for quantum-dot cellular automata," IEEE Trans. Nanotechnol., vol. 6, no. 3, pp. 374–383, May 2007.

[5] V. Pudi and K. Sridharan, "Efficient design of a hybrid adder in quantum-dot cellular automata," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 19, no. 9, pp. 1535–1548, Sep. 2011.

[6] S. Perri and P. Corsonello, "New methodology for the design of efficient binary addition in QCA," IEEE Trans. Nanotechnol., vol. 11, no. 6, pp. 1192–1200, Nov. 2012.

[7] R. E. Ladner and M. J. Fischer, "Parallel prefix computation," J. Association for Comput. Machinery, vol. 27, no. 4, pp. 831–838, Oct. 1980.