

A NOVEL 4-BIT ARITHMETIC LOGIC UNIT IMPLEMENTATION IN QUANTUM-DOT CELLULAR AUTOMATA

NILESH PATIDAR¹, NAMIT GUPTA¹, AMITA KHABIA¹, SUMANT KATIYAL², & K.K. CHOUDHARY³

¹Department of Electronics, SVITS, Indore, Affiliated to RGPV, Bhopal, Madhya Pradesh, India

²School of Electronics, DAVV, Takshshila Campus, Indore, India

³Department of Physics, SVITS, Indore, Affiliated to RGPV, Bhopal, Madhya Pradesh, India

ABSTRACT

Quantum cellular automata (QCA) is an advanced nanotechnology that attempts to create general computational at the nano scale by controlling the position of single electrons. QCA technology has large potential in terms of high space density and power dissipation with the development of the faster computer with smaller size & low power consumption. The logic design of ALU, an important constituent part of CPU, is described in this paper. A design constructing 4-bit Arithmetic Logic Unit (ALU) based on the QCA (Quantum-Dot Cellular Automata) is presented. The proposed 4-bit Arithmetic Logic Unit is simulated using the QCADesigner tool and experiment result shows that the arithmetic & logical function of the designed circuit is correct. Our aim is to provide evidence that QCA has potential applications in future computers provided that the underlying technology is made feasible.

KEYWORDS: ALU, Nano-Technology, QCA

INTRODUCTION

The advancement of complementary metal-oxide semiconductor (CMOS) digital design lies in reducing the process technology. This scaling is projected to end in the next 20 years due to CMOS fundamental physical limits. Thus, the future points to nano-electronics as the way to continue the improvements that have been gained using CMOS technology. QCA architecture is among the new emerging nanotechnologies presently being investigated as potential CMOS candidates. Several studies have reported that QCA can attain high device density, ultra lower power consumption, and very high switching speed.

The QCA (*Quantum-dot Cellular Automata*) is considered to be a promising technology for future generation ICs. Since it was first introduced in 1993, experimental devices for semiconductor, molecular, and magnetic approaches have been developed. Recent papers show that QCA can achieve high density, fast switching speed, and room temperature operation.

The QCA cell consists of a system of four quantum dots charged with only two free electrons. Electrostatic repulsion between these electrons force them to occupy only the diagonal sites creating a so called “polarization” used to encode binary information, as seen in Figure 1. Interactions between neighboring cells allow for the layout of functional circuits, where the objective is to layout the cells in such a way that the ground state polarization of the output cells represents the correct output of a function to a given set of input vectors.

In computing, an arithmetic logic unit (ALU) is a digital circuit that performs arithmetic and logical operations. The ALU is a fundamental building block of the central processing unit of a computer, and even the simplest microprocessors contain one for purposes such as maintaining timers. Arithmetic logic unit is a critical component of a

microprocessor and is the core component of central processing unit. An ALU performs arithmetic and logic operations. Arithmetic operations performed are: addition, subtraction, increment and decrement. In addition to these operations it also takes AND of two numbers, OR of two numbers, Compliment of a number and Transfer a number are performed by logical unit.

The structure of the paper is as follows: Section II explains the QCA Background and QCA logic gates; Section III shows the structure of the 4-bit arithmetic and logic unit. In Section IV, proposed layout of the arithmetic and logic unit in QCA. Section V provides the simulation results and discussion. Section VI provides the conclusion.

QCA REVIEW

Background

The basic building block of QCA devices named cell is presented in Figure 1(a). QCA cell consists of four quantum dots in a square array coupled by tunnel barriers, two electrons are injected into the cell. Due to Coulombic repulsion, the two electrons reside in opposite corners representing two polarizations. Some basic elements for QCA logic implementation are the wire, inverter, and majority voters as described below.

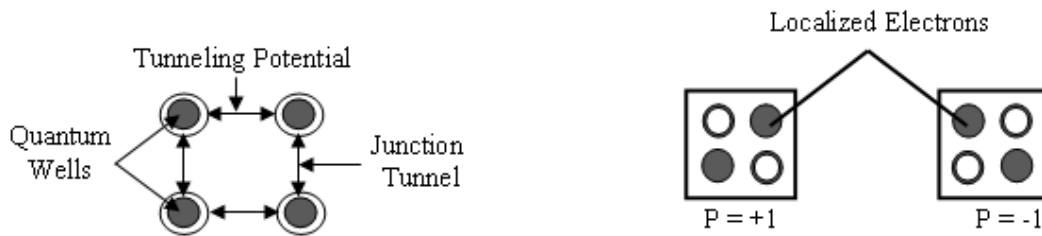


Figure 1 (a): QCA Cell and its Polarization

QCA wire is formed by an array of QCA cells shown in Figure 1(b), which provides a medium for data propagation based on Coulomb interactions.



Figure 1 (b): QCA Wire

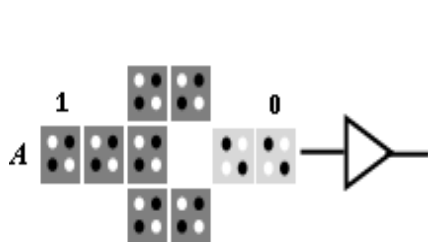


Figure 1 (c): QCA Inverter

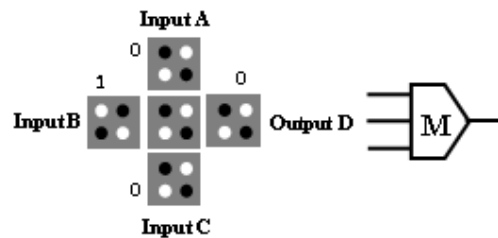


Figure 1 (d): QCA Majority Voter

A QCA inverter is implemented by placing QCA cell at the corner as shown in fig. 1(c), in which the polarization of the output QCA cell is the opposite of the polarization of input QCA cell. QCA majority voter (MV) and its logic symbol are shown in Figure 1(d), MV, is equivalent to a logic function $F(A,B,C)=AB+AC+BC$ and can be implemented by five QCA cells arranged in a cross. Cells A, B, and C are input cells, and cell D is the output cell that is polarized according to the polarization of the majority of the input cells. For example, since two (out of three) input QCA cells are

polarized to -1. The output cell is also polarized to be -1 (see Figure 1(d)). Logical AND and OR functions can be implemented from Majority Voter by presetting one input immutably to binary values 0 and 1, respectively.

QCA Clocking

A QCA cell has four clock phases; they are Switch, Hold, Release and Relax. Figure 2 shows four phases of the QCA clock operation process. During the switch phase, QCA cells begin un-polarized and their inter dots potential barriers are low. The barriers are then raised during this phase, and the QCA cells become polarized according to the state of their driver (i.e. their input cell). It is in this clock phase that the actual computation (or switching) occurs. By the end of this clock phase, barriers are high enough to suppress any electron tunneling and cell states are fixed. During the hold phase, barriers are held high so the outputs of the sub array can be used as inputs to the next stage. In the release phase, barriers are lowered and cells are allowed to relax to an un-polarized state. Finally, during the fourth clock phase, the relax phase, cell barriers remain lowered and cells remain in an un-polarized state. In the meantime, the large scale QCA circuit is partitioned into four clock zones; Fig. 2 shows each clock zone signal and demonstrates the pipeline mechanism. All cells in a certain zone are controlled by the same QCA clock signal. Cells in each zone perform a specific calculation; the state of a zone is then fixed so that it can serve as input signals to the next zone. Information transfers in a pipelined fashion.

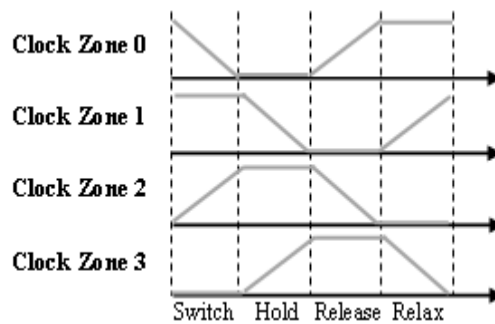
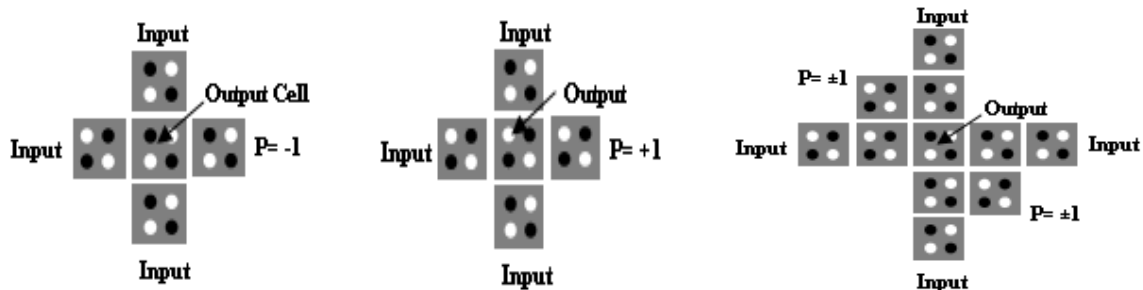


Figure 2: Clocking Zones

Three-Bit & Four-Bit and & or Using Majority Voter Gate

Logical 3-bit, 4-bit gates (AND, OR) functions can be implemented from Majority Voter by presetting cells according to figure 3(a), 3(b) and 3(c) shown below immutably to binary values 0 and 1.



(a) 3-Bit and

(b) 3-Bit or

(c) 4-Bit and/or

Figure 3

ARCHITECTURE OF 4-BIT ALU

An Arithmetic Logic Unit (ALU) is a multi-operation, combinational-logic digital function. It can perform a set of basic four arithmetic operations and a set of four logic operation. The ALU has three selection lines to select a particular

operation in the unit. M is master select line to select the operation either arithmetic or logical and S0 and S1 are other select lines to select various operations. The Arithmetic Logic Unit consist three basic units; arithmetic unit, logical unit and Ripple carry adder circuit. The block diagram of 4-bit ALU is depicted in figure 4.

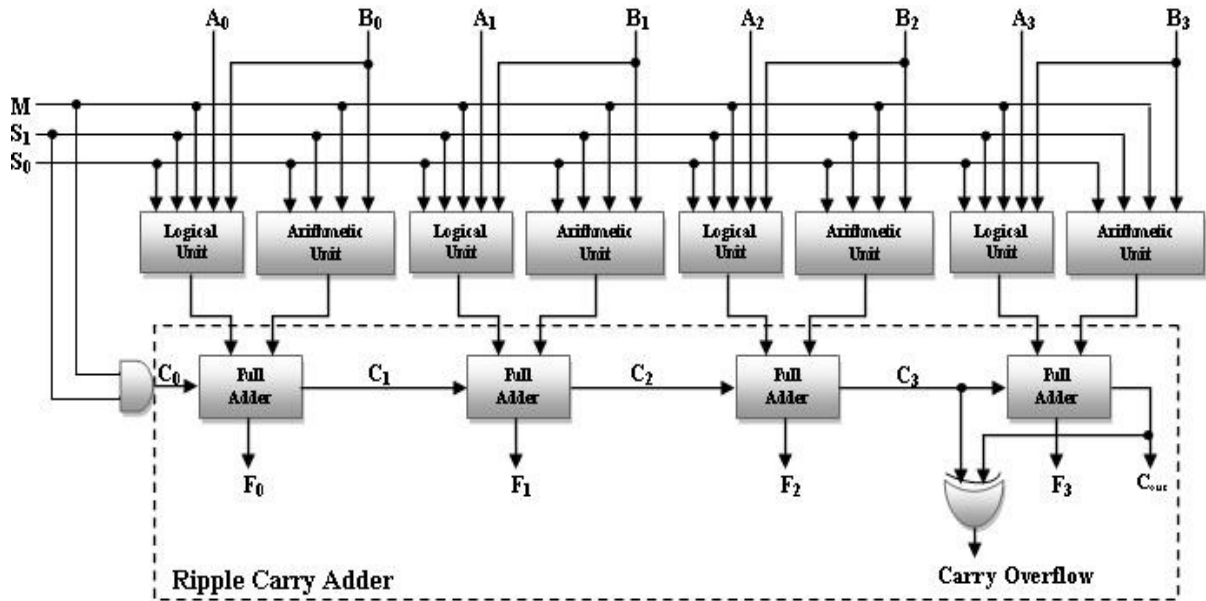


Figure 4: Block Diagram of 4-Bit Arithmetic Logic Unit

Arithmetic Unit

The Arithmetic Unit modifies the second operand and passes it to the Full adder to the arithmetic. The arithmetic operations performed are addition, subtraction, increment and decrement. The function table, truth table and logic diagram are described. Initial carry C_0 is given by equation below, where

$$C_0 = MS_1 \tag{1}$$

Table 1: Functional Table of Arithmetic Unit

M	S ₁	S ₀	Function Name	Function	X	Y	C ₀
1	0	0	Decrement	A-1	A	all 1's	0
1	0	1	Add	A+B	A	B	0
1	1	0	Subtract	A+B'+1	A	B'	1
1	1	1	Increment	A+1	A	all 0's	1

Table 2: Truth Table of Arithmetic Unit

M	S ₁	S ₀	B	Y
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	1	0	0
1	1	1	0	0
1	1	1	1	0

According to truth table we write Boolean equation of arithmetic unit shown in equation (2) and design logic circuit of arithmetic unit shown in figure 5.

$$Y = M\bar{S}_1B + M\bar{S}_0\bar{B} \quad \dots (2)$$

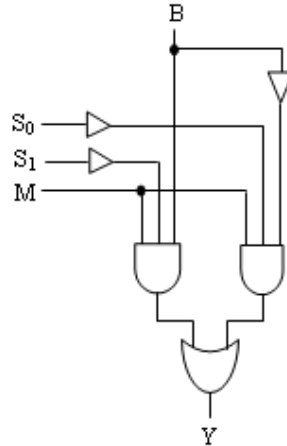


Figure 5: Logic Diagram of Arithmetic Unit

Logical Unit

The logic operations are performed in the logic Unit. These are Complement, AND, OR and Identity.

Table 3: Functional Table of Logical Unit

M	S ₁	S ₀	Function Name	Function	X	Y	C ₀
0	0	0	Complement	A'	A'	0	0
0	0	1	AND	A and B	A&B	0	0
0	1	0	Identity	A	A	0	0
0	1	1	OR	A or B	A B	0	0

Table 4: Truth Table of Logical Unit

M	S ₁	S ₀	X
0	0	0	A'
0	0	1	A & B
0	1	0	A
0	1	1	A B
1	X	X	A

According to truth table we write Boolean equation of logical unit shown in equation (3) and design logic circuit of logical unit shown in figure 6.

$$X = \bar{M}\bar{S}_1\bar{S}_0\bar{A} + \bar{M}S_1S_0B + S_0AB + S_1A + MA \quad \dots (3)$$

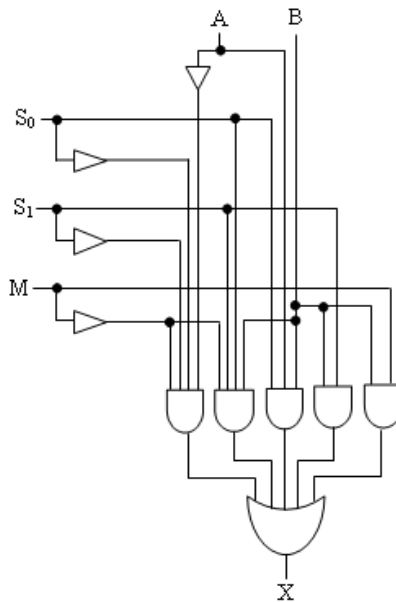


Figure 6: Logic Diagram of Logical Unit

Ripple Carry Adder

Adders are usually implemented by combining multiple copies of simple components. The natural components for addition are half adders or full adders. The principal problem in constructing an adder for n-bit numbers out of small pieces is propagating carries from one piece to the next.

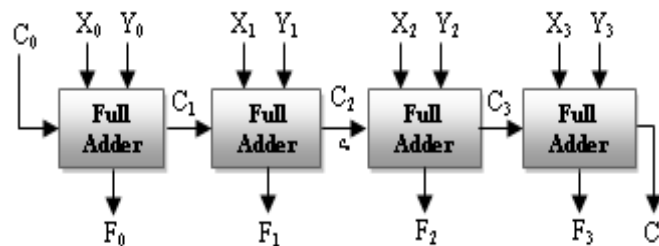


Figure 7: Four bit Ripple Carry Adder

The most obvious way to solve this with a ripple carry adder, consisting of n full adders, illustrated in Figure 7, where each block consists of full adder. The carry out of one full adder is connected to the carry in of the adder for the next most significant bit.

The carries ripple from the least significant bit (on the right) to the most significant bit. In general, the time a circuit takes to produce an output is proportional to the maximum number of logic levels through which a signal travels. It takes two levels for first carry to compute from A₀ and B₀. Then it takes 2n levels for n bits. The ripple carry adders are slowest but also the cheapest. It can be built with only n simple cells, connected in a simple, regular way.

IMPLEMENTATION IN QCA

Implementation of 4-bit Arithmetic Logic Unit in QCA is verified using QCA Designer tool. Arithmetic unit, Logical unit and Ripple carry adder is highlighted in blocks. The inputs are at the top side and the outputs are at the bottom side in QCA layout of ALU. The proposed design is shown in Figure 8.

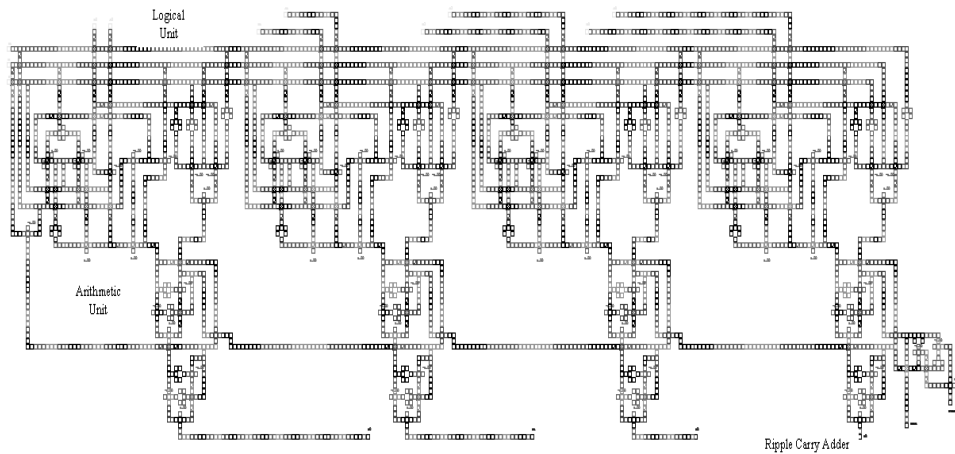


Figure 8: Implementation of 4-Bit ALU in QCA

SIMULATION RESULTS AND DISCUSSIONS

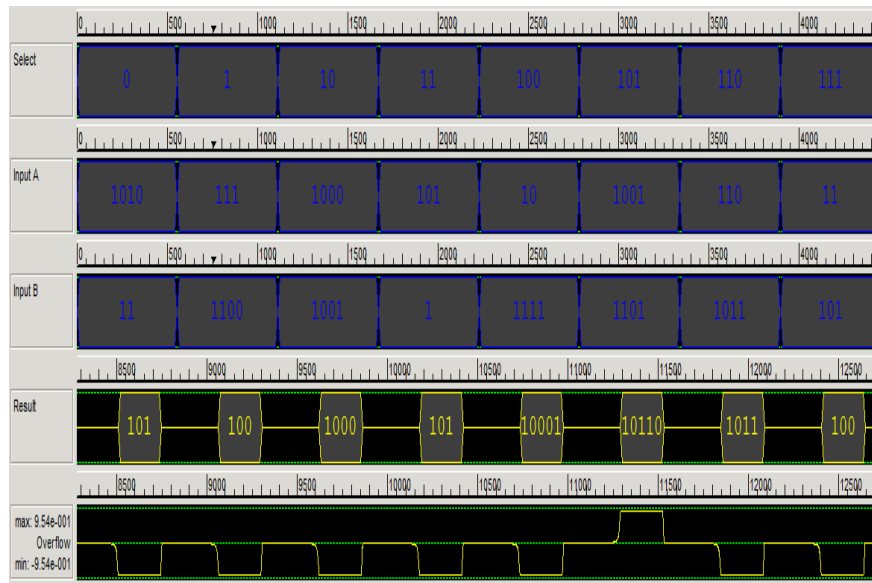


Figure 9: Simulation Results of 4-Bit Arithmetic Logic Unit

All the designs were verified using QCADesigner tool ver. 2.0.3. In the bi-stable approximation, we used the following parameters: cell size=18 nm, number of samples=12800, convergence tolerance=0.001000, radius of effect=65.00 nm, relative permittivity=12.900000, clock high=9.800000e-022, clock low=3.800000e-023, clock amplitude factor=2.000000, layer separation=11.500000, maximum iteration per sample=100. All of these parameters which used are default parameters in QCADesigner tool. In our QCA layouts, we have the goal of workable designs with compact layout. The simulation Results of 4-bit ALU is shown in figure 9.

According to simulation results, as we seen the highlighted area is the output of ALU and the QCA circuit of 4-bit ALU has delay of 15 clocks. This result shows the outputs are same as function table of ALU.

CONCLUSIONS

This paper presents the design of a novel 4-bit Arithmetic Logic Unit based on Quantum-dot Cellular Automata,

which uses multilayer crossovers. Presented design construct using arithmetic, logical and ripple carry adder which have minimum complexity (4738 cells), smaller size of $7.64\mu\text{m}^2$ and lower computational delay of 15 clocks (high speed) and cheap in cost because of its size. The proposed design of 4-bit ALU is much more beneficial.

ACKNOWLEDGEMENTS

I would like thanks to Mr. Namit Gupta sir for guiding and helping to complete this work.

REFERENCES

1. Orlov et al (1999), A. DeHon et al (2004), C. S. Lent et al (1997), C. S. Lent et al (1993), G. Bourianoff et al (2003), H. Qi et al (2003), H. Cho et al (2009), I. Amlani et al (2000), ITRS, "International Technology Roadmap for Semiconductors," [Online] Available: www.itrs.net. J. M. Seminario et al (2004), K. Walus (2002), K. Navi et al (2010), P. D. Tougaw et al (1994), R. P. Cowburn et al (2000), R. K. Kummamuru et al (2003), V. V. Zhirnov et al (2005), W. Wang et al (2003), Y. Wang et al (2004).