International Journal of Scientific Engineering and Technology Research

ISSN 2319-8885
Vol.05,Issue.01,
January-2016,
Pages:0048-0051

www.ijsetr.com

VLSI Realization of Brent Kung Adder using QCA Technology

DOOLAM GOPI[1], NANDIPATI GN PRASAD[2]

[1]PG Scholar, KIET, Korangi, A.P, India, Email: d.gopi.415@gmail.com.
[2]Assistant Professor, KIET, Korangi, A.P, India, Email: prasad471@gmail.com.

Abstract: Quantum-dot cellular automata (QCA) are a new technology suitable for the implementation of ultra-dense low-power high-performance digital circuits. As transistors decrease in size more and more transistors can be accommodated in a single chip, thus increasing chip computational capabilities. However, transistors cannot get much smaller than their current size. A quantum-dot cellular automaton has a simple cell as the basic element. The cell is used as a building block to construct gates and wires. The quantum-dot cellular automata (QCA) approach represents one of the possible solutions in overcoming this physical limit. In this paper designed a new Brent Kung adder (BKA) that performs and achieves the best area-delay tradeoff. The 32-bit version of the Brent-Kung adder (BKA) requires 60 LUTs and shows a delay of 14.876ns which is lower value compared to existing Ripple carry adder (RCA).

Keywords: QCA (Quantum Dot Cellular Automata), Majority Gates (MG).

I. INTRODUCTION

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. The Notre Dame group has developed a new paradigm for ultra-dense and ultra-fast information processing in nano electronic systems. These "Quantum Cellular Automata" (QCA's) are the first concrete proposal for a technology based on arrays of coupled quantum dots. The physical interaction between neighboring cells has been exploited to implement logic functions. New functionality may be achieved in this fashion. A major finding is that all logic functions may be integrated in a hierarchical fashion which allows the design of complicated QCA structures. QCA is a novel emerging technology in which logic states are not stored as voltage levels, but rather the position of individual electrons. Conceptually, QCA represents binary information by utilizing a bi-stable charge configuration rather than a current switch. A major finding is that all logic functions may be integrated in a hierarchical fashion which allows the design of complicated QCA structures. Extensive modeling work of semiconductor quantum dot structures has helped identify optimum design parameters for QCA experimental implementations. A QCA cell can be viewed as a set of four “dots” that are positioned at the corners of a square. A quantum dot is a site in a cell in which a charge can be localized. The cell contains two extra mobile electrons that can quantum mechanically tunnel between dots, but not cells. In the ground state and in the absence of external electrostatic perturbation, the electrons are forced to the corner positions to maximize their separation due to Coulomb repulsion. As shown in below Fig1,

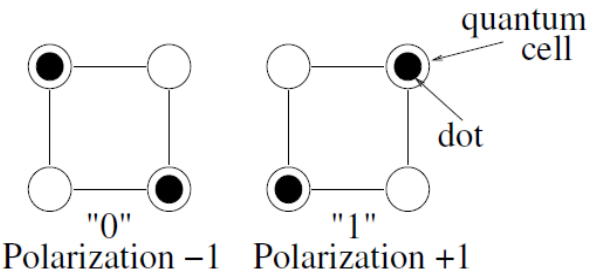

Fig1. QCA Cell.

The two possible charge configurations are used to represent binary“0” and “1”. Note that in the case of an isolated cell, the two polarization states are energetically degenerate.

II. EXISTING METHOD

The existing QCA adder runs in RCA (ripple carry based adder) fashion. The block diagram of N-bit RCA-QCA adder given below.

A. QCA Adder

To introduce the architecture for implementing ripple adders in QCA, let consider two n-bit addends A = an−1,. . . , a0 and B = bn−1, . . . , b0 and suppose that for the i th bit position (with i = n − 1, . . . , 0) the auxiliary propagate and generate signals, namely pi = ai + bi and gi = ai · bi, are computed. ci being the carry produced at the generic (i−1) th bit position, the carry signal ci+2,furnished at the (i+1) th bit

position.In this way, the RCA action, needed to propagate the carry ci through the two subsequent bit positions, requires only one MG. Conversely, conventional circuits operating in the RCA fashion, namely the RCA and the CFA, 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 conventional RCA and CFA. To prove the basic operation of QCA, we must perform the operation

$$M(abc) = a \cdot b + a \cdot c + b \cdot c \quad (1)$$

Equation is exploited in the design of the 2-bitmodule.That also shows the computation of the carry ci+1 = M(pi gi ci). The proposed n-bit adder is then implemented by cascading n/2 2-bit modules as shown in Fig1(a).

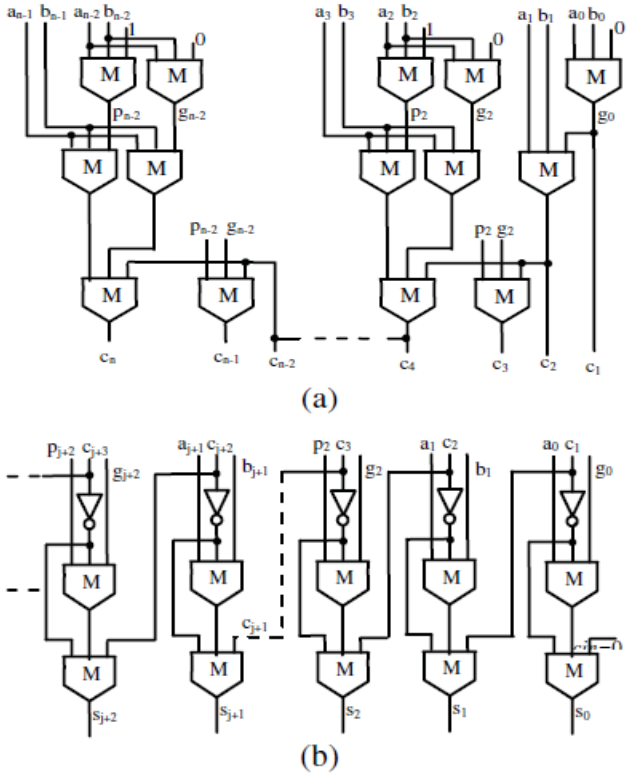

Fig2. Novel n-bit adder (a) carry chain and (b) sum block.

Having assumed that the carry-in of the adder is cin = 0, the signal p0 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. 2.1 (b). 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 sub sequent bit positions to the most significant one. In this case, the first2-bit module computes c2, contributing to the worst case computational path with two cascaded MGs. The subsequent 2-bitmodules 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.

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

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

III. PROPOSED SYSTEM

The proposed Brent-Kung adder is a parallel prefix adder. Parallel prefix adders are special class of adders that are based on the use of generate and propagate signals. The block diagram of 4-bit Brent-Kung adder is shown in Fig3. Simpler Brent-Kung adders was been proposed to solve the disadvantages of Kogge-Stone adders. The cost and wiring complexity is greatly reduced. Furthermore, the study been carried out by implementing Brent Kung Adder in Basic Logic Gate and Compound Gate, then simulate the design in various sizes of transistors in order to see the effects on propagation delay and the number of transistors used. At the end of this paper, evidently the improvement of transistors size contributes reducing the propagation delay and area. Moreover, the requirements of the adder are that it is primarily fast and secondarily efficient in terms of delay and chip area. Parallel Prefix Adder as terminology background is describing prefix as the outcome of the execution of the operation depends on the initial inputs. This is done by segmentation into smaller pieces that computed in parallel.

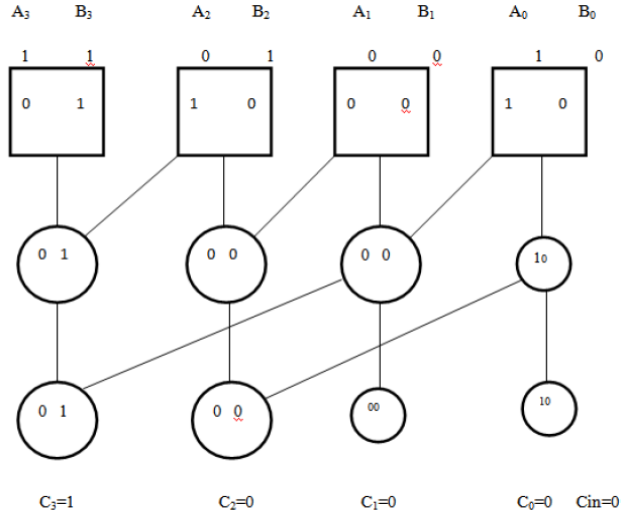

Fig3. Brent-Kung prefix adder.

Then all bits of the sum will begin the process concurrently. There are a lot of parallel prefix adders been developed example in 1960: J. Sklansky–conditional adder, 1973: Kogge-Stone adder, 1980: Ladner-Fisher adder, 1987: Han Carlson adder and 1999: S. Knowles. Other parallel adder architectures also include H. Ling adder in 1981 and 2001: Beaumont-Smith. Practically, the Brent Kung Parallel Prefix Adder has a low fan-out from each prefix cell but has a long critical path and is not capable of extremely high speed addition. In spite of that, this adder proposed as an optimized and regular design of a parallel adder that addresses the problems of connecting gates in a way to minimize chip area. Accordingly, it considered as one of the better tree adders for minimizing wiring tracks, fan out and gate count and used as a basis for many other networks. High performance

microprocessor units require high performance adders and other arithmetic units. Modern microprocessors are however 32 bits as that is the minimum required for floating point arithmetic as per the IEEE 754 Standard.

A. Brent Kung Architecture

In order to approach the structure known as the Brent Kung Structure, which uses the logarithmic concept, the entire architecture is easily understood by dividing the system into three separate stages:

1. Generate/Propagate Generation
2. The Dot (·) Operation
3. Sum generation

B Generate/Propagate Generation

If the inputs to the adder are given by the signals A and B, then the generate and propagate signals are obtained according to the following equations.

$$G = A.B$$
$$P = A \text{ xor } B$$

C. The Dot (·) Operation

The most important building block in the Brent Kung Structure is the dot (·) operator. The basic inputs to this structure are the generate and propagate signals generated in the previous stage. The · operator is a function that takes in two sets of inputs-- (g, p) and (g', p') and generates a set of outputs-- (g+ pg', pp'). These building blocks are used for the generation of the carry signals in the structure. For the generation of the carry signals, the carry for the kth bit from the carry look-ahead concept is given by

$$Co,k=Gk+Pk(Gk-1+Pk-1+Pk-1(\ldots+P1(G0+P0Ci,0))) \quad (4)$$

Using the dot operator explained above the Equation 4 can be written for the different carry signals as

$$Co,0 = G0 + P\,Ci,0 = a\,(G0,P0)$$
$$Co,1 = G1 + G0\,P1 = a\,((G1, P1)\cdot(G0, P0))$$
$$C0,k = a\,((Gk,Pk)\cdot(Gk-1,Pk-1)\cdot\ldots\cdot(G0,P0))$$

where a is a function defined in order to access all the tuples. The 8-bit Brent Kung Structure is shown in Figure 3.2. This figure shows all the carry signals generated at different stages in the structure. In the structure, two binary tree structure are represented the forward and the reverse trees. The forward binary tree alone is not sufficient for the generation of all the carry signals. It can only generate the signals shown as Co,0,Co,1, Co,3 and Co,7. The remaining carry signals are generated by the reverse binary tree.

D. Sum Generation

The final stage in this architecture is the sum generation stage. The sum is given by

$$S = A \text{ xor } B \text{ xor } C$$

where A and B are the input signals, and C is the carry signal. The carry is obtained from the dot operator stage discussed earlier, and the exclusive of A and B is actually the propagate signal itself. Hence the sum 'S' can finally be represented and realized as

$$S = P \text{ xor } C$$

Using the above four stages, the complete architecture is built. The complete block level design of a 32-bit Brent Kung Structure is made. The simulation results for this structure are also available and will be discussed towards the end of Chapter 4 in comparison to the simulation results obtained from the proposed conditional sum adder.

IV.RESULT

The test bench is developed in order to test the modeled design. The developed test bench will automatically force the inputs and will make the operation of the adder. The design implemented and verified using XILINX ISE tool.

A. Existing Results

For n-bit RCA adder the simulation Results given as

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	37	4656	0%
Number of 4 input LUTs	64	9312	0%
Number of bonded IOBs	128	232	55%

Fig4. Device Utilization.

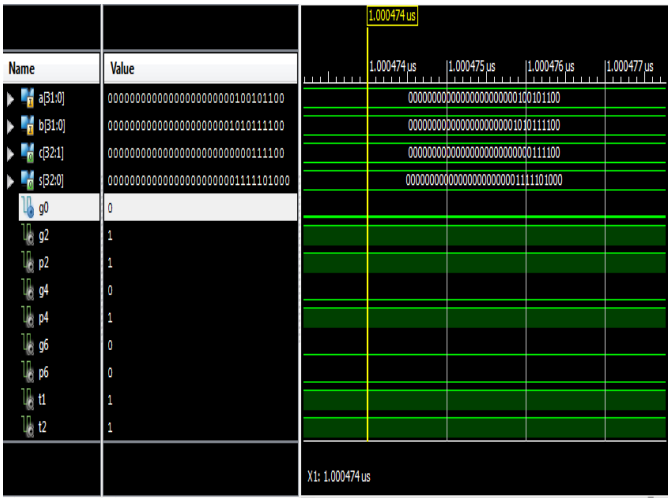

Fig5. Simulation Result.

B. Proposed Results

In proposed architecture 32-bit BRENT-KUNG adder results can be given as below using Xilinx ISE tool.

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	32	4656	0%
Number of 4 input LUTs	60	9312	0%
Number of bonded IOBs	85	232	36%

Fig6. Device Utilization.

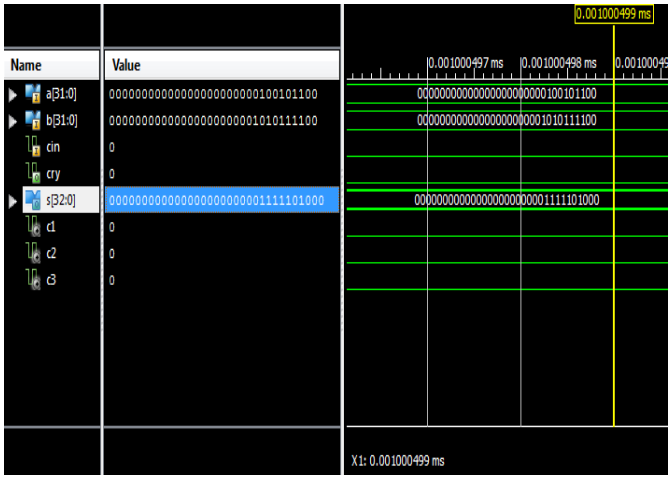

Fig7. Simulation Result

C. Proposed & Existing Comparisons

Table1. Comparison between existing and proposed technique

Parameter	Existing Design	Proposed Design
Delay	38.718nS	14.794nS
No: of LUT	64	60
No: of slices	37	32
Memory	191820kbytes	186764kbytes

From above table it states that the proposed architecture can produce lower delay in order of 14.794ns .and the number of 4-input LUTs required by the proposed method are 60 LUTs which are less compared to existing method 64LUTS.The number of slices required by the proposed method is 32 where as in existing these are 37.and the memory required are less in proposed technique than the existing technique.

V. CONCLUSION

A new adder designed in QCA was presented. Each individual module was tested for its correct functionality and then all the modules were integrated to form a entire Brent-kung adder. It operated in CLA fashion. The number of 4 input LUT's used in proposed Brent-kung adder(BKA) developed using QCA technology are 60 where as in existing 64 and delay also reduced from 38.714ns to 14.794ns.

VI. REFERENCES

[1] V. Pudi and K. Sridharan, "Low complexity design of ripple carry andBrent–Kung adders in QCA," IEEE Trans. Nanotechnol., vol. 11, no. 1,pp. 105–119, Jan. 2012.

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

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

[4] V. Pudi and K. Sridharan, "New decomposition theorems on majoritylogic for low-delay adder designs in quantum dot cellular automata,"IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 59, no. 10, pp. 678–682,Oct. 2012.

[5] K. Walus and G. A. Jullien, "Design tools for an emerging SoCtechnology: Quantum-dot cellular automata," Proc. IEEE, vol. 94, no. 6,pp. 1225–1244, Jun. 2006.

[6] S. Bhanja, M. Ottavi, S. Pontarelli, and F. Lombardi, "QCA circuits forrobust coplanar crossing," J. Electron. Testing, Theory Appl., vol. 23,no. 2, pp. 193–210, Jun. 2007.

[7] Stefania Perri,Pasquale Corsonello,and Giuseppe Cocorullo," IEEE Transactions On Very Large Scale Integration (VlSI) Systems, Vol. 22, No. 5, May 2014".