International Journal for Modern Trends in Science and Technology
Volume: 02, Issue No: 12, December 2016
ISSN: 2455-3778
http://www.ijmtst.com

Design of High Speed Area & Power Efficient Parallel Prefix Adders with QCA Majority Logic

K.J.S. Nanda[1] | N. Praveen Kumar[2] | J.E.N. Abhilash[3]

[1]PG Scholar, Department of ECE, Nova Engineering College
[2]Head of Department, Department of ECE, Nova Engineering College.
[3]Associate Professor, SCET, Narsapur.

To Cite this Article
K.J.S. Nanda, N. Praveen Kumar and J.E.N. Abhilash, "Design of High Speed Area & Power Efficient Parallel Prefix Adders with QCA Majority Logic", *International Journal for Modern Trends in Science and Technology*, Vol. 02, Issue 12, 2016, pp. 43-48.

ABSTRACT

As transistors reduce in size more and more of them can be accommodated in a one die, thus growing chip computational capabilities. However, transistors cannot find much lesser than their current size. The quantum-dot cellular automata (QCA) approach represents one of the possible solutions in overcoming this physical edge, level while the propose of logic modules in QCA is not always straight forward. In this brief, we propose some adder that out performs all state-of-the art competitors and achieves the best area-delay tradeoff.

KEYWORDS: *Adders, nano computing, quantum-dot cellular automata (QCA).*

I. INTRODUCTION

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 mechanical phenomena. 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.

- 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.

Another feature that is often considered important for a model of quantum cellular automata is that it should be universal for quantum computation (i.e. that it can efficiently simulate quantum Turing machines,[1][2] some arbitrary quantum

circuit[3] or simply all other quantum cellular automata[4][5]).

Models which have been proposed recently impose further conditions, e.g. that quantum cellular automata should be reversible and/or locally unitary, and have an easily determined global transition function from the rule for updating individual cells.[2] Recent results show that these properties can be derived axiomatically, from the symmetries of the global evolution.[6][7][8]

In 1982, Richard Feynman suggested an initial approach to quantizing a model of cellular automata.[9] In 1985, David Deutsch presented a formal development of the subject.[10]Later, Gerhard Grössing and Anton Zeilinger introduced the term "quantum cellular automata" to refer to a model they defined in 1988,[11] although their model had very little in common with the concepts developed by Deutsch and so has not been developed significantly as a model of computation.

Models of universal quantum computation

The first formal model of quantum cellular automata to be researched in depth was that introduced by John Watrous.[1] This model was developed further by Wim van Dam,[12] as well as Christoph Dürr, Huong LêThanh, and Miklos Santha,[13][14] Jozef Gruska.[15] and Arrighi.[16] However it was later realised that this definition was too loose, in the sense that some instances of it allow superluminal signalling.[6][7] A second wave of models includes those of Susanne Richter and Reinhard Werner,[17] of Benjamin Schumacher and Reinhard Werner,[6] of Carlos Pérez-Delgado and Donny Cheung,[2] and of Pablo Arrighi, Vincent Nesme and Reinhard Werner.[7][8] These are all closely related, and do not suffer any such locality issue. In the end one can say that they all agree to picture quantum cellular automata as just some large quantum circuit, infinitely repeating across time and space.

Models of physical systems:

Models of quantum cellular automata have been proposed by David Meyer,[18][19] Bruce Boghosian and Washington Taylor,[20] and Peter Love and Bruce Boghosian[21] as a means of simulating quantum lattice gases, motivated by the use of "classical" cellular automata to model classical physical phenomena such as gas dispersion.[22] Criteria determining when a quantum cellular automaton (QCA) can be described as quantum lattice gas automaton (QLGA) were given by Asif Shakeel and Peter Love.[23]

Quantum dot cellular automata:

A proposal for implementing classical cellular automata by systems designed with quantum dots has been proposed under the name "quantum cellular automata" by Doug Tougaw and Craig Lent,[24] as a replacement for classical computation using CMOS technology. In order to better differentiate between this proposal and models of cellular automata which perform quantum computation, many authors working on this subject now refer to this as a quantum dot cellular automaton.

Quantum-dot cellular automata (QCA) is an smart rising technology suitable for the development of ultra-dense low-power high-performance digital circuits . For this cause, in the previous few years, the design of efficient logic circuits in QCA has expected a great deal of notice. 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 in[2]. Of course, the architectures normally working in traditional CMOS designs are considered a first location for the new design environment. Ripple-carry (RCA), carry look-ahead (CLA), and conditional sum adders were existing. The carry-flow adder (CFA) shown and was mainly an better RCA in which damaging wires effects were mitigated.Parallel prefix architectures, including Brent–Kung (BKA), Kogge–Stone, and Han–Carlson adders,were analyzed and implemented in QCA. newly, more efficient designs were proposed for the carry look ahead adder and the brent kung adder, and for the carry look ahead adder and the Carry flow adder. In this brief, an new technique is presented to implement high-speed low-area adders in QCA. Theoretical formulations verified for CLA and parallel-prefix adders are here broken for the realization of a novel 2-bit addition slice in [11]. The latter allows the carry to be propagated through two following bit-positions with the delay of just one majority gate (MG). In addition, the bright top level architecture leads to very compact layouts, thus avoiding needless clock phases due to long interconnections. An adder designed as proposed runs in the RCA approach, but it exhibits a computational delay lower than all state-of the-art competitors and achieves the lowest area-delay product (ADP).

II. KOGGE STONE ADDER

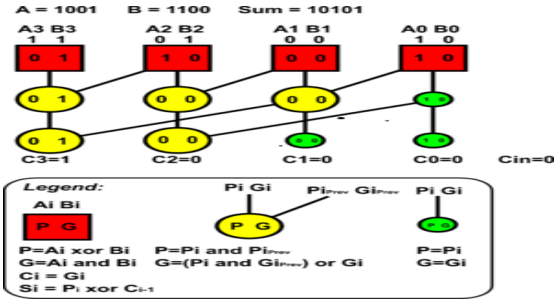

Fig.1. 4 bit Kogge stone adder

The Kogge–Stone adder is a parallel prefix form carry look-ahead adder. Other parallel prefix adders include the Brent-Kung adder, the Han Carlson adder, and the fastest known variation, the Lynch-Swartzlander Spanning Tree adder.[1]

The Kogge–Stone adder takes more area to implement than the Brent–Kung adder, but has a lower fan-out at each stage, which increases performance for typical CMOS process nodes. However, wiring congestion is often a problem for Kogge–Stone adders. The Lynch-Swartzlander design is smaller, has lower fan-out, and does not suffer from wiring congestion; however to be used the process node must support Manchester Carry Chain implementations. The general problem of optimizing parallel prefix adders is identical to the variable block size, multi level, carry-skip dder optimization problem, a solution is shown in this model

An example of a 4-bit Kogge–Stone adder is shown to the right. Each vertical stage produces a "propagate" and a "generate" bit, as shown. The culminating generate bits (the carries) are produced in the last stage (vertically), and these bits are XOR'd with the initial propagate after the input (the red boxes) to produce the sum bits. E.g., the first (least-significant) sum bit is calculated by XORing the propagate in the farthest-right red box (a "1") with the carry-in (a "0"), producing a "1". The second bit is calculated by XORing the propagate in second box from the right (a "0") with C0 (a "0"), producing a "0".

The Kogge–Stone adder concept was developed by Peter M. Kogge and Harold S. Stone, which they published in 1973 in a seminal paper titled A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence Equations.[3]

1. Pre processing: This step involves computation of generate and propagate signals consequent too

each pair of bits in A and B. These signals are specified by the logic equations below:

$p_i = A_i$ xor B_i

$g_i = A_i$ and B_i

2. Carry look ahead network: This block differentiates KSA from other adders and is the main force behind its high performance. This step involves computation of carries related to each bit. It uses group propagate and create as intermediate signals.

$p = p_i$ and $p_{i\ prev}$

$s = (p_i$ and $g_{i\ prev})$or g_i

3. Post processing :This is the final step and is com -mon to all adders of this family (carry look ahead). It involves calculation of sum bits. Sum bits are comp- uted by the reason given below

$s_i = p_i$ xor c_{i-1}

$c_i = g_i$

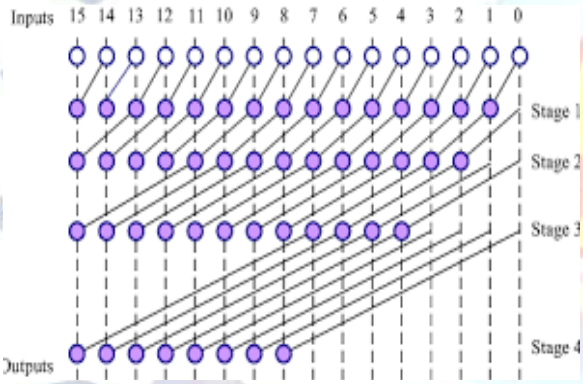

Fig.2. 8 bit Kogge stone adder

III. BRENT KUNG ADDER

The parallel prefix adders [are more flexible and are used to speed up the binary add-ons. Parallel prefix adders are obtained from Carry Look Ahead (CLA) structure. We use tree construction form to raise the speed of arithmetic operation. Parallel prefix adders are best adders and these are used for high performance arithmetic circuits in industries in[12]. The construction of parallel prefix adder involves three stages

1. Pre- processing stage
2. Carry generation network
3. Post processing

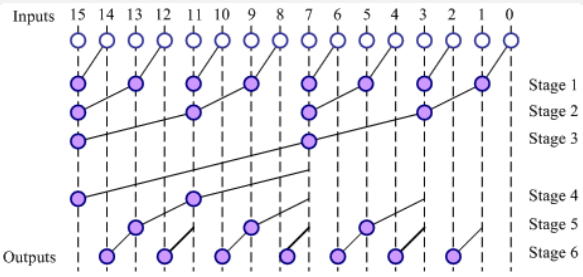

Fig.3. 16 bit Brent kung adder

IV. LADNER-FISCHER

Ladner- Fischer parallel prefix adder was developed by R. Ladner and M. Fischer in 1980. Ladner-Fischer prefix tree is a structure that sits between Brent-Kung and Sklansky prefix tree. The LF adder [5] has minimum logic depth but it has large fan-out. Ladner- Fischer adder has carry operator nodes. The delay for the type of Ladner-Fischer prefix tree is log2n+1. Fig.5 shows the 16-bit LF adder.

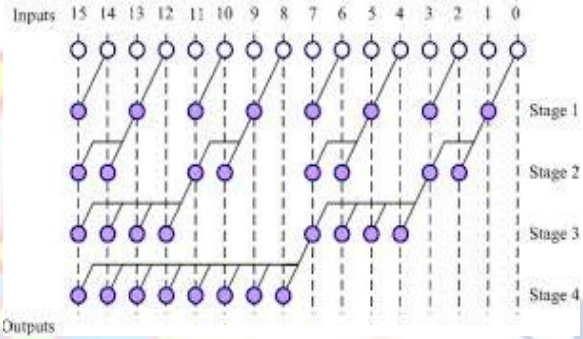

Fig.4. shows 16 bit Ladne fishcer adder

V. RESULTS

Result waveforms:

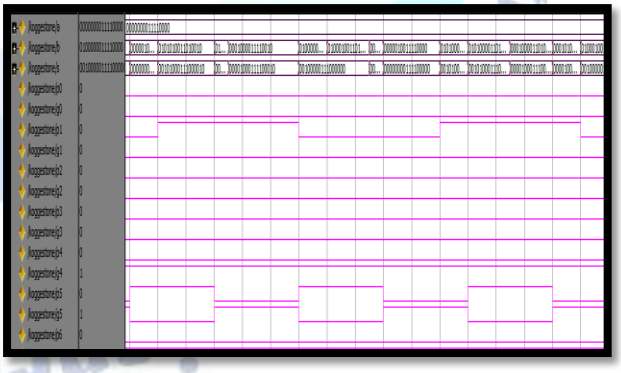

Fig 5. shows kogge stone adder output waveforms

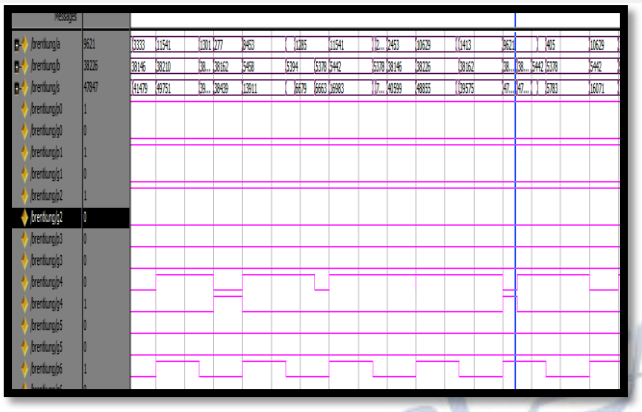

Fig.6. shows Brent kung adder output waveforms

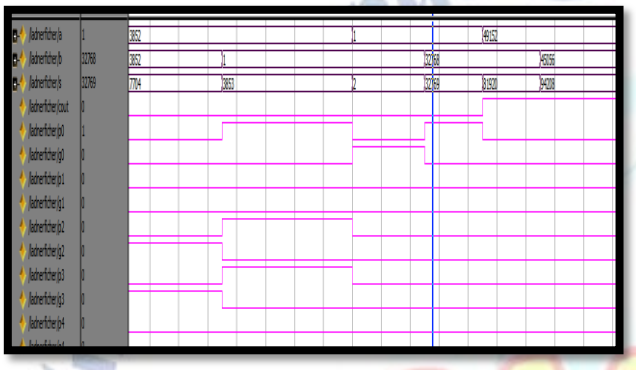

Fig.6. shows Ladne fiscer adder output waveforms

TABLE I
Comparison Of Different Adders

ADDER TYPE	NO OF SLICES	NO OF LUT'S	DELAY(ns)
16 BIT KOGGE-STONE ADDER	34	60	19.651
16 BIT LADNER-FISCER ADDER	21	39	19.218
16 BIT BENT-KUNG ADDER	19	37	18.479

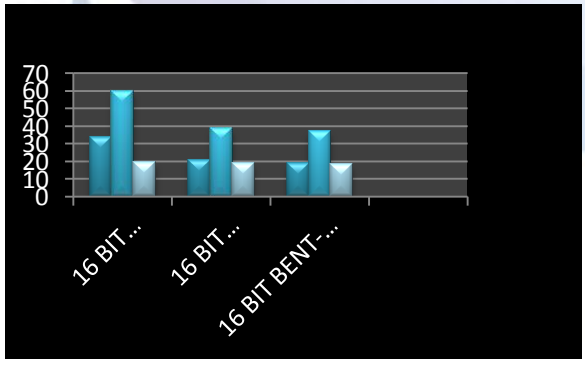

Fig 7.Performance Comparisons of adders in this paper

VI. Conclusion

In this paper we have realized different high speed adders like Brent Kung, Kogge stone,Ladner Fischer of 16 bit. It is implemented with only QCA Majority gate logic. which is a latest approach, we have implemented, simulated, synthesized all the VHDL codes on spartan 3E FPGA board of number XC3S500e-5cp132.We have observed that Kogge stone is having Less Delay more area. It is only suitable where area is not constraint, If area is less and Delay is not constraint Ladner fiscer suitable. Out of all adders Brent kung adder is having good Area Delay Product (ADP), and Power Delay Product (PDP), so we are concluding that Brent kung adder implementation using QCA Majority gate is more Area, Power, and Delay efficient. Our work can be extended for 32 bit, 64 bit and also 128 for further analysis.

References

[1] Stefania Perri,Pasquale Corsonello,and Giuseppe Cocorullo. "Area-Delay Efficient Binary Adders in QCA"*IEEE Trans. Very Large Scalentegr (VLSI) Syst.*,vol. 22, no. 5, , pp. 1174–1179, May 2014.

[2] L. Lu, W. Liu, M. O'Neill, and E. E. Swartz lander, Jr., QCA systolic array design," *IEEE Trans. Comput.*, vol. 62, no. 3, pp. 548–560,Mar. 2013.

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

[4] V. Pudi and K. Sridharan, "New Decom Posi -tion theorems on majority logic 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] V. Pudi and K. Sridharan, "Low complexity design of ripple carry and Brent–Kung adder in QCA," *IEEE Trans. Nanotechnol.*, vol. 11, no. 1, pp. 105–119, Jan. 2012.

[6] M. Janez, P. Pecar, and M. Mraz, "Layout design of manufacturable quantum-dot cellular automata," *Microelectron. J.*, vol. 43, no. ,pp. 501–513, 2012.

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

[8] J.D.Wood and D.Tougaw, "Matrix multi-plication using quantum dot cellular automata to implement conventional microelectronics," *IEEE Trans. Nanotechnol.*, vol. 10, no. 5, pp. 1036–1042, Sep. 2011.

[9] W. Liu, L. Lu, M. O'Neill, and E. E. Swartz lander, Jr., "Design rules for quantum-dot cellular automata," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2011, pp. 2361–2364.

[10] K. Kong, Y. Shang, and R. Lu, "An optimized majority logic synthesis methology for quantum-dot cellular automata," *IEEE Trans. Nanotechnol.*, vol. 9, no. 2, pp. 170–183, Mar. 2010.

[11] K. Walus, G. A. Jullien, and V. S. Dimitrov, "Computer arithmetic structures for quantum cellular automata," in *Proc. Asilomar Conf. Sygnals, Syst. Comput.*, Nov. 2003, pp. 1435–1439.

[12] S. Bhanja, M. Ottavi, S. Pontarelli, and F. Lom -bardi, "QCA circuits for robust coplanar crossing," *J. Electron. Testing, Theory Appl.*, vol. 23, no. 2, pp. 193–210, Jun. 2007.

[13] 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.

[14] J. Huang and F. Lombardi, *Design and Test of Digital Circuits by Quantum -Dot Cellular Automata*. Norwood, MA, USA: Artech House,2007.

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

[16] K.Kim, K. Wu, and R.Karri, Toward designing robust QCA architectures in the presence of sneak noise paths," in *Proc. IEEE Design, Autom. Test Eur Conf. Exhibit.*, Mar. 2005, pp. 1214–1219.

[17] A. Gin, P. D. Tougaw, and S. Williams, "An alternative geometry for quantum dot cellular auto-mata," *J. Appl. Phys.*, vol. 85, no. 12,pp. 8281–8286, 1999.