ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

A New Approach Of Designing High Speed Area & Power Efficient Adders With QCA Majority Logic Gates

S.Rajababu
Swarnandhra College of Engineering and Technology, Seetaramapuram , Narsapur.

R.Murali Krishna
Swarnandhra College of Engineering and Technology, Seetaramapuram , Narsapur.

S.Srilali
Swarnandhra College of Engineering and Technology, Seetaramapuram , Narsapur.

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.

Index Terms:

Adders, nanocomputing, quantum-dot cellular automata (QCA).

1.INTRODUCTION:

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.CARRY LOOKAHEAD ADDER:

A QCA is a nanostructure having as its basic cell a square four quantum dots structure charged with two free electrons able to tunnel through the dots within the cell . Because of Coulombic revulsion, the electrons will always away in opposite corners. The locations of the electrons in the polarization determine two possible stable states that can be associated to the binary states one and zero. Although side by side cells contact through electrostatic forces and tend to align their polarizations, quantum dot cellular automata cells do not have intrinsic data flow directionality. To achieve controllable data orders, the cells quantum dot cellular automata within a design are partitioned into the so-called clock zones that are progressively associated to four clock signals, each phase shifted by 90°. This clock method, named the zone clocking scheme, makes the quantum dot cellular automata methods intrinsically pipelined, as every clock zone behaves similar to a D-latch.

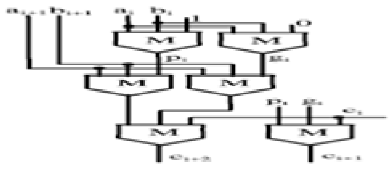


Fig. 1. Novel 2-bit basic module.

QCA cells are used for both logic structures and interconnections that can develop either the coplanar cross or the bridge technique. The primary logic gates naturally available with in[5] the QCA technology are the not and the majority gate. Given three inputs a, b, and c, the majority gate performs the logic function reported in (1) provided that all input cells are connected to the same clock signal clk x (with x ranging from 0 to 3), whereas the remaining cells of the MG are linked to the clock signal clkx+1

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

a number of designs of adders in QCA exist in text. The RCA and the CFA process n-bit operands by cascad-ing n full-adders (FAs). Even while these addition circuits use different topologies of the generic full adder, they have a carry input to carryout put path consisting of one majority gate, and a carry-in to sum bit path containing two MGs plus one inverter in[17]. As a importance, the worst case computational paths of the n-bit RCA and the n-bit CFA consist of (n+2) MGs and one inverter. A CLA architecture created by 4-bit slices was also presented. In particular, the auxiliary generated to and generate signals, namely pi = ai + bi and gi = ai • bi, are computed for each bit of the operands and then they are grouped four by four. Such a designed n-bit CLA has a computational path collected of 7+4×(log4 n) cascaded MGs and single not gate.

This can be easily verified by observe that, given the spread and generate signals (for which only one MG is required), to compute grouped propagate and grouped generate signals; four cascaded MGs are introduced in the computational path. In addition, to compute the carry signals, one level of the CLA logic is necessary for each factor of four in the operands length. This means that, to process n number of bit addends, log4 n levels of CLA logic are required in[14], each contributing to the computational path with four cascaded MGs. Finally, the computation of sum bits introduces two further cascaded MGs and one inverter. The parallel-prefix BKA demonstrated, exploits more efficient basic CLA logic structures.

As its main advantage over the previously described adders in[9], the BKA can achieve lower computational delay. When n-bit operands are processed, its most terrible case computational path consists of 4 × log2 n – 3 cascaded MGs and one inverter in[4]. Apart from the level required to compute propagate and generate signals, the prefix tree consists of 2×log2n–2 stages. From the logic equations provided , it can be easily verified that the first stage of the tree introduces in the computational path just one MG; the last stage of the tree contributes with only one MG; whereas, the middle stages introduce in the critical path two cascaded MGs each in [10]. Finally, for the computation of the addition bits, added two cascaded MGs and one inverter are added.

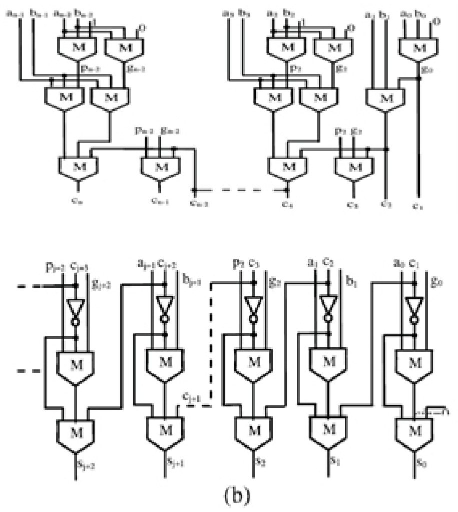


Fig. 2. Novel n-bit adder (a) carry chain and (b) sum block.

With the main point of trading off area and delay, the hybrid adder (HYBA) described and combines a parallel-prefix adder with the Ripple carry adder. In the attendance of n-bit operands, this structural design has a most terrible computational path consisting of 2×log2 n+2 cascaded MGs and one inverter. When the methodology recently proposed and was exploited, the worst case path of the CLA is reduced to 4 × _log4 n_ + 2 ×_log4 n_ – 1.By applying the decomposition method demonstrated the computational paths of the CLA and the CFA are reduced to 7 + 2×log2(n/8) MGs and one not gate and to (n/2)+3 MGs and one inverter, respectively.

III.KOGGE STONE ADDER:

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

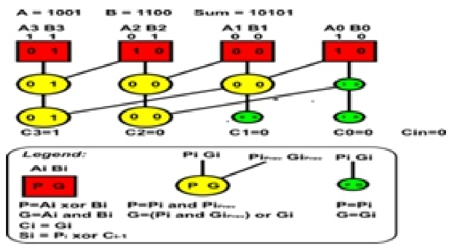


Fig.3. 4 bit Kogge stone adder

KSA is a parallel prefix form carry look ahead adder. It generates carry in O (log n) time and is generally considered as the fastest adder and is widely used in the industry for high performance arithmetic circuits in [15]. In KSA, carries are computed fast by computing them in parallel at the cost of increased area. The complete performance of KSA can be easily comprehended by analyzing it in terms of three distinct parts 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:

pi= Ai xor Bi

gi= Ai and Bi

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=pi and pi prev

s=(pi and gi prev)or gi

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

si=pi xor ci-1

ci=gi

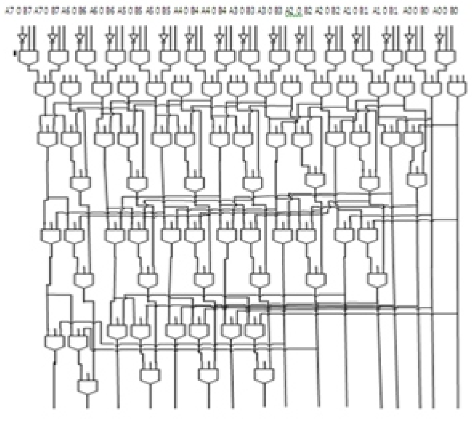


Fig.4. 8 bit Kogge stone adder

IV.RIPPLE CARRY ADDER:

A quantum-dot cellular automaton (QCA) is an attractive emerging technology suitable for the development of ultra dense low-power higher performance digital circuits. For this reason in the final few years, the design of capable logic circuits in[7] .QCA has received a great deal of attention. particular efforts are going to arithmetic circuits, with the major interest focused on the binary addition that is the basic operation of any digital system. Of course, the designs commonly working in traditional CMOS designs are considered a first reference for the new design environment in [8]. Ripple-carry, carry look-ahead (CLA), and conditional sum adders were implemented .

The carry-flow adder shown in was mainly an improved RCA in which damaging wires effects were mitigated. Parallel-prefix architectures, as well as Brent–Kung , Kogge–Stone, Ladner–Fischer, and Han–Carlson adders, were analyzed and implemented in [11] QCA. Recently, further efficient designs were proposed for the CLA and the BKA, and for the Carry look ahead Adder and the Carry Flow Adder. In this brief, an original technique is presented to implement high speed low-area adders into QCA. Theoretical formulations established for CLA and parallel-prefix adders are here exploited for the realization of a novel 8-bit addition portion. The former 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 in [6], thus avoiding unnecessary clock phases due to lengthy interconnections.

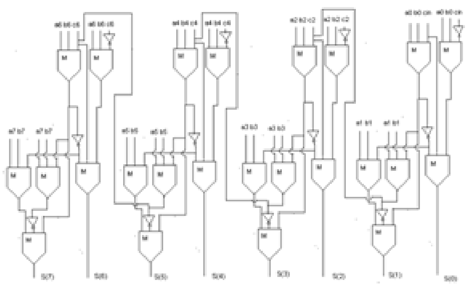


Fig.5. 8 bit Ripple carry adder

V. 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

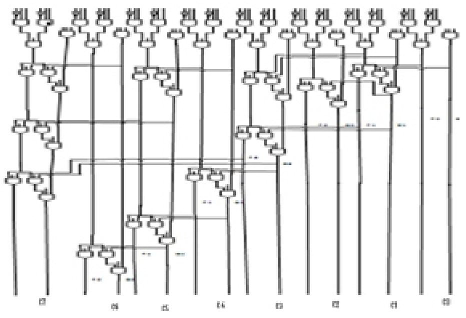


Fig.6. 8 bit Brent kung adder

The Brent-Kung adder is a parallel prefix adder. Parallel prefix adders are unique class of adders that are based on the use of generate and propagate signals. Simpler Brent-Kung adders was been projected to solve the disadvantages of Kogge-Stone adders. The rate and wiring density is really condensed. But the logic strength of Brent-Kung adders increases to 2log (2n-1), so the speed is lower. The block diagram of 8-bit Brent-Kung adder is shown in above Fig 6.

VI. LING ADDER:

Ling introduced the pseudo carry notion and showed that the delay in the carry path could be reduced by forming the pseudo-carry term (Hn:0 = gn + Gn−1:0). Ling's pseudocarry is less complex than the Gn:0 as the fan-in of each AND gate to compute Ling pseudo carry gets reduced by one in[13]. Thus, Hn:0 is easier to implement than Gn:0.Ling also showed that recursion can be applied to produce group pseudo carry by hierarchically combining the pseudo carries over sub-groups in a similar way as in a prefix adder. This means that the pseudo carry Hj:i of a group from position j to position i can be constructed from the pseudo carries Hj:k and Hk−1:i of two subgroups formed one from position j to k and other from position k − 1 to i, respectively.

Hj:i =Hj:k + Kj−1:k−1Hk−1:i Ling adder using valency-2 at its first level of carry computation will have the equation for group generate as Hn:n−1 = gn + gn−1 = (an • bn) + (an−1 • bn−1). This equation can be implemented using a single gate in fast computation of sum with delay of the adder reduced by approximately 1 inverter (FO4) gate delay. But in terms of conventional prefix adder computation, this would be two levels of logic gates. The first level is two input NAND/NOR gate to compute g and k terms and the second level is a three input gate. Many adders are implemented as shown in using the Ling theory of factorization to the first level of the carry computation stage. These work faster than the prefix adders but it would it be better to apply the Ling factorization in [16].each level of carry tree to reduce the complexity of the group generate function and make the adder work even faster. This is achieved by the recent contribution of Jackson and Talwar . They noted that Eq. can be further factorized as

Hj:i = (Lj:k + Hj:k)(Hj:k + Hk−1:i) (11)

where Lj:k = Kj−1:k−1. the first term is denoted as Dj:k and the second term is denoted as H or Rj−k+1j:i . In the latter, the subscript indicates the range of group over which this carry is computed and the superscript indicates the number of missing not kill bits in the reduced generate or group pseudo carry equation. These missing not kill bits are included in the D term which needs to be combined with reduced group generate equation to get the actual group generate function. Gj:i = (Dj:k)(Hj:i) Thus, Ling factorization is applied to compute group pseudo carry in terms of H or R for the carry tree stage. For a higher valency like 4 and 5 the same factorization (of any number of not kill terms at any level of carry tree) can bedone but still the carry equation requires more than one logic level to implement using gates with maximum 4 inputs. The sum computation can be achieved by implementing a multiplexer with Hn−1:0 as a control signal for sum out at bit position n without any extra delay. In this case the control signal is Rj−k+1j:i . The equation to compute sum for bit position n is given as,

sn = Rn−1:0(pn) + Rn−1:0(pn Dn−1:0)

VII.RESULTS :

Result waveforms:

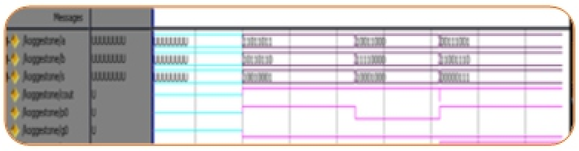


Fig.7. shows kogge stone adder output waveforms

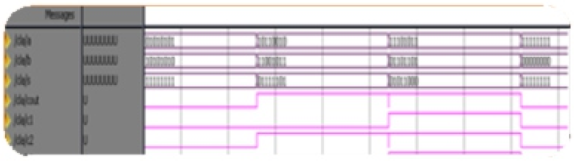


Fig.8. shows carry look ahead adder output waveforms

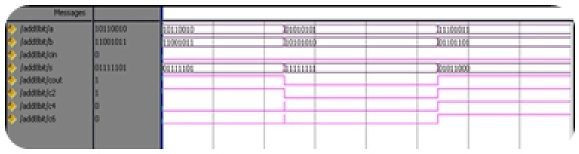


Fig.9. shows ripple carry adder output waveforms

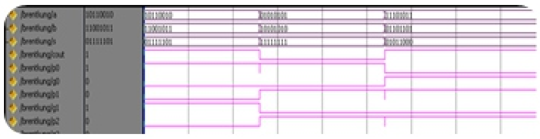


Fig.10. shows Brent kung adder output waveforms

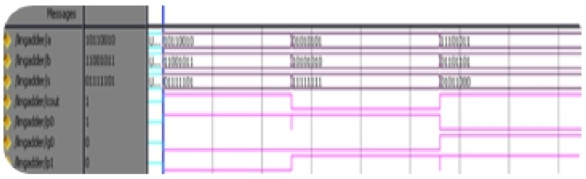
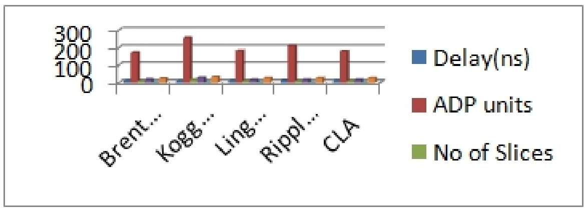


Fig.11. shows Ling adder output waveforms

VIII.CONCLUSION:

TABLE I: Comparison Of Different Adders

Adder	n	Delay	ADP units	No .of Slices	No. of LUT's	Power(W)	PDP units
Brent Kung	8	10.045ns	170.765	9	17	0.1307	22.3189
Kogge Stone	8	9.154ns	256.312	16	28	0.1390	35.6273
Ling Adder	8	12.142ns	182.13	9	15	0.1305	23.7679
Ripple Carry	8	13.169ns	210.704	9	16	0.1306	27.5179
CLA	8	11.947ns	179.205	9	15	0.1305	23.3862



In this paper we have realized different high speed adders like Brent Kung, Kogge stone, Ling, Ripple carry adder and Carry look ahead adder of 8 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 Ling Adder is 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 16 bit, 32 bit and 64 bit also 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.

[18] C. S. Lent, P. D. Tougaw, W. Porod, and G. H. Bernestein, “Quantum cellular automata,” Nano- technology, vol. 4, no. 1, pp. 49–57, 1993.