Performance Estimation of n-bit Classified Adders

Oday Abdul Lateef Abdul Ridha
Electronics and Communications Eng. Dept.
University of Baghdad
Baghdad - Iraq

ABSTRACT

This work presents a performance estimation of classified n-bit binary adders. Since gate count and gate level depth directly related to speed, area, and power consumption of the adder circuit, therefore they are adopted as performance criteria. 2-inputs gate model is adopted as basic unit for the performance evaluation. The main focus of the work on carry ripple, carry lookahead, and multilevel carry select adders. The study showed that ripple carry adder is the best among other adders from area point of view. Whereas carry lookahead and multilevel carry select based on carry lookahead adders are the best from delay point of view. For the area delay product, 3 levels carry select adder based on carry ripple showed that it is the best.

General Terms

Binary adder, Performance estimation.

Keywords

Performance estimation, gate level depth, gate count, delay, area, power consumption, ripple adder, carry ahead adder, multilevel carry select adder.

1. INTRODUCTION

In the majority of digital control and signal processing systems or applications the critical operations are the addition, multiplication and accumulation. Addition is an indispensable operation for any digital system. More than 8.72% of typical scientific program or application use addition. Moreover, addition represents the basic operation of other arithmetic operations such as subtraction, multiplication, and division [1]. Therefore a fast and accurate operation of a digital system is greatly influenced by the performance of the adders that contain. Hence, the need of higher performance adder is increasing as the need of high performance systems or applications are increasing. The improving performance of the digital adder would extensively advance the execution of binary operations inside a circuit compromised of such blocks [2]. Performance in this context means fast, small area, and less power consumption. The area and power consumption are directly related constraints. They are mainly depending on the number of basic building blocks of the circuit, i.e., they are dependent upon logic gate count of the circuit. Although the power consumption is also depends on the circuit switching rate. Delay and consequently speed of any logic circuit depends on the largest gate level depth in the logic circuit. In most cases, including adder circuit, Area and speed are two conflicting constraints. Improving speed results always in larger areas. Optimizing the speed and area of the adder is a major design issue. Over the last decades several researchers had worked on improving adder architecture in terms of area, speed, or both [3-7]. Other researchers studied and analysis exiting architecture in the aim of finding optimized architecture that merges conflicting constraints [8-10].

Prior works either use HDL simulation software, which take a lot of time, technology dependent, and do not give the clear answer to a question about their behavior in n-bit devices. Or they aren't focusing on all performance criteria. In this work, selected class adders are analyzed. Closed form expressions for counting gates and level depth for these classes are derived. These expressions can be easily used for estimating area, speed, and power consumption of selected adders and for n-bit. Moreover combination of these expressions can be used for constructing more complex criteria.

This paper organized in to two parts. In the first part, the description and analysis of Ripple Carry (RCA), Carry LookAhead (CLA), Carry Select (CSeA), multi-level carry select based on ripple carry (MCSeR), and multi-level carry select based on carry lookahead (MCSeL) adders are developed. In the second part of the paper, the comparisons are presented.

2. CLASSIFIED BINARY ADDER ARCHITECTURES

In this section, different classes of adder architectures are briefly described and analyzed. The analysis includes estimation of circuit delay in term of gate level depth and area (complexity) in term of gate count. The selected classes of adders are: Ripple Carry (RCA), Carry LookAhead (CLA), Carry Select (CSeA), multi-level carry select based on ripple carry (MCSeR), and multi-level carry select based on carry lookahead (MCSeL) adders. For the analysis and estimation purposes, 2-inputs basic gate model is used. This means that in spite of used blocks and gates in logic circuit, the circuit is seen as if it is constructed from 2-input OR, AND, NOR, and/or NAND gates only. NOT gates of the circuit are neglected in the analysis. For gates of other types or have more inputs, they can be expressed by using their equivalent 2-inputs gates. For example, to estimate the area and delay of the circuit shown in fig.1a, equivalent circuit of fig.1b is used.
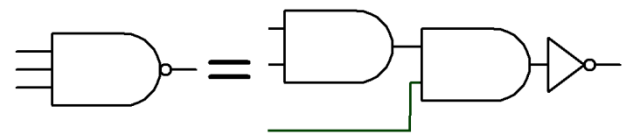

Fig.1 a) 3-input NAND gate b) its equivalent circuit

Thus the 3-input NAND gate has two gate count and two level gate depth. Estimated area and delay of the circuit can be easily expressed as in eq.(1) and (2), respectively.

$$Area = 2g_a \qquad (1)$$

$$Delay = 2.g_d, \qquad (2)$$

where g_a is the area of 2-input gate, g_d is the propagation delay of 2-input gate.

In general, it can be shown that any *n*-inputs gate can be expressed as combination of $(n-1)$ 2-inputs gates with a propagation delay time or level depth equals to $\lceil \log_2 n \rceil$. In the literature, there are many types of adders. Part of them, which are belong to different classes, are chosen for analysis. The other can be analyzed in the same way.

2.1 Ripple carry adder (RCA)

The well-known architecture of an *n*-bit ripple adder is composed of *n* cascaded full adders (FAs), as shown in the fig.2 [11].
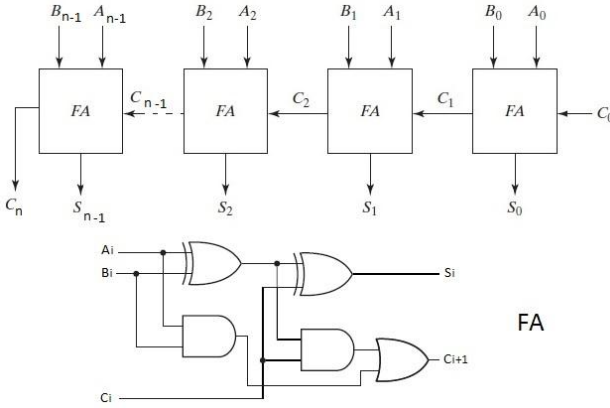

Fig.2 Architecture of ripple carry adder

From the figure, it can be easily determine number of logic elements that needed to construct the adder. To determine the overall delay and 2-inputs gates count, it must be noted that XOR is consists of 3 2-inputs gates and has two level depth. It can see that each bit of the adder requires nine gates. It can also see that carry path has the largest gate level depth, except for single bit adder. The path elongated by 2 for each additional bit. Thus, gate count (area) and gate level depth (delay) of *n*-bit carry ripple adder can be expressed using Eq.(3) and (4), respectively.

$$GateCount_{RCA}(n) = 9n \qquad (3)$$

$$LevelDepth_{RCA}(n) = 2n \text{ for n} \geq 2 \qquad (4)$$

2.2 Carry lookahead adder (CLA)

The most widely used technique for reducing the delay of carry propagation is carry lookahead (CLA) algorithm [12]. In this technique, by adding some more logic circuit, all incoming carries are generated in parallel. The CLA exploits the fact that the carry generated by a bit-position *i* depends on the three inputs to that position. If A_i and B_i are two inputs then if $A_i = B_i = 1$, a carry is generated independently of the carry from the previous bit position and if $A_i = B_i = 0$, no carry is generated. Similarly if $A_i \neq B_i$, a carry is generated if and only if the previous bit-position generates a carry. 'C_i' is initial carry, "S_i" and "C_{i+1}" are output sum and carry respectively, then Boolean expression for calculating next carry and addition is:

$P_i = A_i \oplus B_i$ -- Carry Propagation
$G_i = A_i \cdot B_i$ -- Carry Generation
$Ci+1 = G_i + (P_i \cdot Ci)$ -- Next Carry
$S_i = A_i \oplus B_i \oplus C_i$ -- Sum Generation

Thus, for *n*-bit adder, carry can be extended, as shown below:

$C_1 = G_0 + P_0 \cdot C_0$
$C_2 = G_1 + P_1 \cdot C_1 = G_1 + P_1 \cdot G_0 + P_1 \cdot P_0 \cdot C_0$
$C_3 = G_2 + P_2 \cdot G_1 + P_2 \cdot P_1 \cdot G_0 + P_2 \cdot P_1 \cdot P_0 \cdot C_0$
$C_4 = G_3 + P_3 \cdot G_2 + P_3 \cdot P_2 \cdot G_1 + P_3 \cdot P_2 \cdot P_1 \cdot G_0 + P_3 \cdot P_2 \cdot P_1 \cdot P_0 \cdot C_0$

.
.
.

$C_n = G_{n-1} + P_{n-1} \cdot G_{n-2} + P_{n-1} \dots P_{n-2} G_{n-3} + \dots + P_{n-1} \dots P_1 \cdot G_0 + P_{n-1} \dots P_0 \cdot C_0$

Fig.3 represents a 3-bit carry lookahead adder. The circuit of carry generator is shown in fig.4.
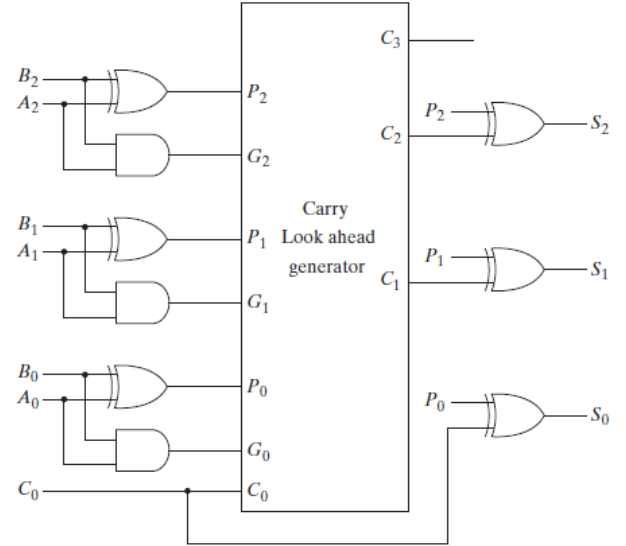

Fig.3 Carry lookahead adder

From fig 3, it can be divide the gate count into two parts. First part which requires seven 2-inputs gates (two XOR and one AND) for each bit of the adder. Second part which contains carry generator circuit (cg). It can see from fig.4 that for one-bit adders, carry generator circuit requires one 2-input OR gate and one 2-input AND gate. For two-bit adder, carry generator requires additional one 2-input AND, one 3-input AND, and one 3-input OR gate. For three-bit adder, additional one 2-input AND, one 3-input AND, one 4-input AND, and one 4-input OR gate are required. It has been shown that *n*-input gate can be built using (*n-1*) 2-input gates. Therefore, the 2-input OR count for *n*-bit carry generator can be expressed as in Eq.(5).

$$2ORcount(n) = \sum_{i=1}^{n} i \qquad (5)$$

For counting the number of 2-AND gates in the *n*-bit carry generator Eq.(6) can be used.

$$2ANDcount(n) = \sum_{j=1}^{n} (\sum_{i=1}^{j} i) \qquad (6)$$

Adding Eq.(5) with Eq.(6) get the total 2-input gates count. Using series theorems and properties, the total 2-input gates count of *n*-bit carry generator circuit can be expressed in closed form as in Eq.(7)

$$2AND\_OR\_count = \frac{n^3 + 6n^2 + 5n}{6} \qquad (7)$$

Thus, the overall gate count (area) of *n*-bit carry lookahead adder (two parts) can be expressed in Eq.(8)

$$\text{GateCount}_{\text{CLA}}(n) = \frac{n^3 + 6n^2 + 47n}{6} \qquad (8)$$

The delay of carry lookahead adder can be determined by the longest gate level depth. Form fig.4, it seems that path of the highest order carry is the longest path since the gate level depth of *n*-inputs gate equals to $\lceil \log_2 n \rceil$. But looking to the sum and carry generation paths together (fig.3), will reveal that the second most high order carry along with sum path always represents the deepest level. Therefore the overall gate level depth of *n*-bit carry lookahead adder can be expressed as in Eq.(9) .

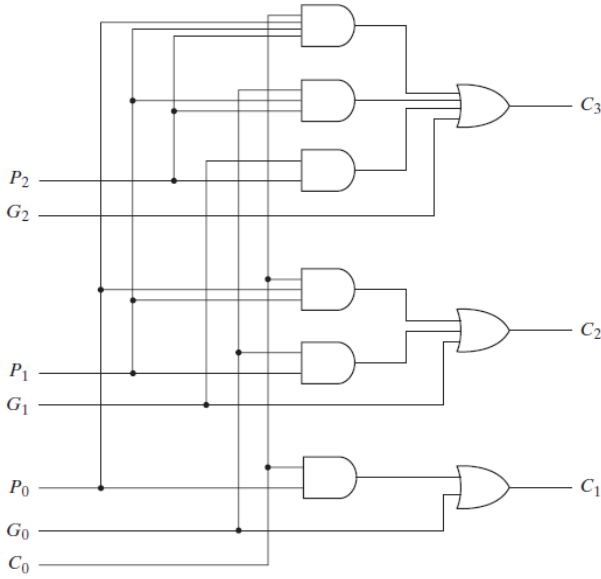$$\text{LevelDepth}_{\text{CLA}}(n) = [4 + 2. \lceil \log_2(n-1) \rceil] \qquad (9)$$

Fig.4 Carry generator circuit

2.3 Carry Select Adder (CSeA)

Instead of waiting for the carry propagates *n* positions from low order position towards high order position, as in ripple carry adder, carry select adder is used. In this scheme, as shown in fig.5, the *n* bits of the operands are split into two groups. First group contains the *k* low order bits of the operands. The other group contains the reminder *n-k* bits of the operands[13,14]. Each group are summed separately. First group requires single *k*-bit adder. The *n-k* bits in the second group are added in two ways: one assuming a carry-in of the group is 0 and the other with a carry-in of 1. This process results two precomputed sum and carry-out pairs. Later as the carry-out of the first group becomes known, the correct pair is selected. In this way, selecting the correct set of the outputs (out of the two sets) are only needed without waiting for the carry to further propagate through the *n-k* positions. Adding bits in this group requires two *n-k* adders and a multiplexer for selecting correct set. It can be seen that carry select architecture can utilized any adder type including ripple or lookahead or even mixed of them.

Thus, regardless of utilized adders, the gate count and gate level depth and consequently area and delay of the carry select adder can be determined using Eq.(10) and Eq.(11), respectively.

$$\text{Gatecount}_{\text{CSeA}}(n) = 2 \times \text{GateCount}_{(n-k)-\text{bit atop adder}}$$
$$+ \text{GateCount}_{k-\text{bit atop adder}} + \text{GateCount}_{(k-n) \text{ atop mux}} \qquad (10)$$

$$\text{LevelDepth}_{\text{CSeA}}(n) = \text{LevelDepth}_{\text{mux}}$$
$$+ \max(\text{LevelDepth}_{k-\text{bit atop adder}}, \text{LevelDepth}_{(n-k)-\text{bit atop adder}}) \qquad (11)$$

It can be shown that for better utilization of carry select architecture, *k* must be $\frac{n}{2}$, if *n* is even or $\frac{n+1}{2}$, otherwise.

It well-known that the construction of *n*-bit 2to1 multiplexer requires gate count equals $3 \times n$ 2-input gates and has a gate level depth of 2 [10]. Therefore, if *n* is even and previous condition is satisfied, then Eq.(10) and Eq.(11) can be rewritten as in Eq.(12) and Eq.(13), respectively.

$$\text{GateCount}_{\text{CSeA}}(n) = 3 \times \text{Gatecount}_{\left(\frac{n}{2}\right)-\text{bit atop adder}} + 3\left(\frac{n}{2}\right) \qquad (12)$$

$$\text{LevelDepth}_{\text{CSeA}}(n) = \text{LevelDepth}_{\left(\frac{n}{2}\right)-\text{bit atop adder}} + 2 \qquad (13)$$

2.3.1 Carry Select Architecture based on ripple carry adder (CSeR)

First variant of carry select adder is the utilization of ripple carry adder in its architecture. The Eq.(12) and Eq.(13) represent general expressions for calculating gate count and gate level depth of carry select adder. These expressions can be rewritten for carry select architecture based ripple carry adder by substituting gate count and gate level depth of ripple carry adder, Eq.(3) and Eq.(4). Eq.(14) and Eq.(15) for calculating gate count and gate level depth of carry select adder are obtained, respectively.

$$\text{GateCount}_{\text{CSeR}}(n) = 15.n \qquad (14)$$

$$\text{LevelDepth}_{\text{CSeR}}(n) = n + 2 \qquad (15)$$

2.3.2 Carry Select Architecture based on carry lookahead adder (CSeL)

First variant of carry select adder is the utilization of carry lookahead adder in its architecture. Again, by substituting gate count and gate level depth of carry lookahead adder, Eq.(8) and Eq.(9) in respective Eq.(12) and Eq.(13), expressions for calculating gate count and gate level depth of carry select adder are obtained, as in Eq.(16) and Eq.(17).

$$\text{GateCount}_{\text{CSeL}}(n) = \frac{n^3 + 12n^2 + 212n}{16} \qquad (16)$$

$$\text{LevelDepth}_{\text{CSeL}}(n) = 6 + 2. \left\lceil \log_2\left(\frac{n}{2} - 1\right) \right\rceil \qquad (17)$$

2.3.3 Multi-level Carry Select Architecture (MCSe)

It can be note that carry select technique can be applied recursively, i.e., the utilized adders in carry select are in turn has a carry select architecture. In this work; each time carry select architecture is applied, the "level" of adder increases. For example, 3-level carry select adder means three times recursively carry select architecture is used. It should note that regardless of adder levels, at final stage either ripple or

lookahead carry adders must be utilized. However, If $n=2^i$, for any integer i, closed form expressions for calculating the gate count (area) and the gate level depth (delay) can be derived. To do so

1) Rewrite Eq.(11) and Eq.(12) in a general form as in Eq.(18).

$$F(n) = A.F\left(\tfrac{n}{2}\right) + B\left(\tfrac{n}{2}\right), \qquad (18)$$

where A is a constant, F and B arbitrary functions of n.

2) By applying Eq.(18) two and three times recursively, Eq.(19) and Eq.(20) are obtained.

$$F_2(n) = A.\left(A.F\left(\tfrac{n}{4}\right) + B\left(\tfrac{n}{4}\right)\right) + B\left(\tfrac{n}{2}\right) \qquad (19)$$

$$F_3(n) = A\left(A.\left(A.F\left(\tfrac{n}{8}\right) + B\left(\tfrac{n}{8}\right)\right) + B\left(\tfrac{n}{4}\right)\right) + B\left(\tfrac{n}{2}\right) \qquad (20)$$

3) From Eq.(18) through (20), a general form can be obtained, as shown in Eq.(21).

$$F(n,l) = A^l.F\left(\tfrac{n}{2^l}\right) + \sum_{i=1}^{l} A^{i-1} B\left(\tfrac{n}{2^l}\right), \qquad (21)$$

where l represents the number of applying carry select architecture recursively,
n represents the adder's operand length in bits.

Now, from Eq.(3) and Eq.(12), $A=3$, $B(n)=3n$, and $F(n)=9n$. By applying them in Eq.(21), the general form for calculating the gate count of the multi-level carry select architecture based on carry ripple adder (MCSeR) is obtained.

$$\text{GateCount}_{MCSeR}(n,l) = 3.n.\left[4\left(\tfrac{3}{2}\right)^l - 1\right] \quad (22)$$

From Eq.(4) and Eq.(13), $A=1$, $B(n)=2$, and $F(n)=2n$. In the same way, applying them in Eq.(21), the general form for calculating the gate level depth (delay) of the multi-level carry select architecture based on carry ripple adder (MCSeR) is obtained.

$$\text{LevelDepth}_{MCSeR}(n,l) = 2\left[\tfrac{n}{2^l} + l\right] \qquad (23)$$

For the multi-level carry select architecture based on carry lookahead adder (MCSeL) same thing can be used. But using Eq.(8) and Eq.(9). Thus, the gate count of MCSeL can be expressed as follows:

$$\text{GateCount}_{MCSeL}(n,l) = [3^l\left(\frac{n^3}{6.2^{3l}} + \frac{n^2}{2^{2l}} + \frac{47n}{6.2^l}\right) \\ + 3n\left(\left(\tfrac{3}{2}\right)^l - 1\right)] \qquad (24)$$

The general expression for calculating the gate level depth of MCSeL is

$$\text{LevelDepth}_{MCSeL}(n,l) = 2\left[2 + l + \left\lceil log_2\left(\tfrac{n}{2^l} - 1\right)\right\rceil\right] \qquad (25)$$

3. PERFORMANCE COMPARISONS

In this section, the results obtained from comparing the adder architectures are presented. Comparisons include the gate count and gate level depth in terms of the unit gate. In order to overcome the conflict criteria, the product of gate count and gate level depth is also present. All adders are compared for different lengths. The comparisons are performed under a simple unit-gate model. In this model, each gate with two inputs has a gate count and a gate level depth of one, except for the XOR/XNOR gates having gate count and gate level depth of two. For the gates with more inputs, the gate count and gate level depth can be computed in terms of the ones given for the gates with two inputs. Also, inverters and buffers are ignored. Using this model gives a good approximation for measuring the area and speed of select adders. Fig.6, fig.7, and fig.8 show respectively the gate count (area), gate level depth (delay) and the products for the studied adder architectures as a function of operand length in bits. To give an overall view of a gate count and gate level depth of selected adders, logarithmic scale for x-axes and y-axes are used. This type of scale eases the process of comparison between adders. The selected adders for comparison are ripple carry (RCA), carry lookahead(CLA), carry select based ripple carry (CSeR), 3 levels multilevel carry select based on ripple carry (MCSeR3), carry select based on carry lookahead (CSeL), and 3 levels multilevel carry select based on carry lookahead (MCSeL3).
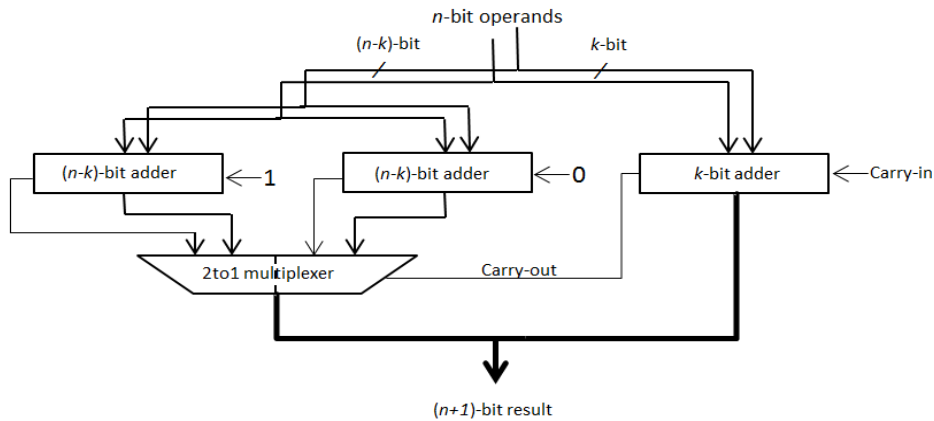

Fig.5 Architecture of carry select adder

4. RESULTS AND CONCLUSIONS

Obtained graphs showed that ripple carry adder is best from gate count (area) point of view. From gate level depth (delay) of view adders: CLA, CSeA, MCSeA were the best. All these adders have the same gate level depth. MCSeL3 showed that it is the best among the other adders form gate count gate level product point of view. The gate count graph (fig.6)

showed that the gate count increases with the increasing the level of multilevel carry select adder based on carry ripple. Whereas the situation in the carry select based carry lookahead is reversed. This because the gate counts of the carry lookahead adder has an order of 3. Divide the operand by two leads to divide the utilized adders by eight.
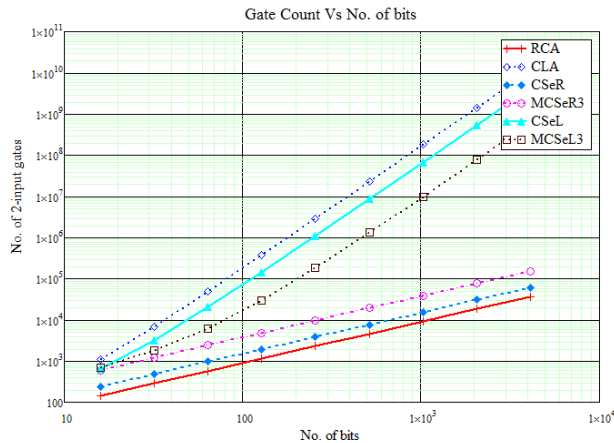
Fig.6 Gate count vs. no. of bits

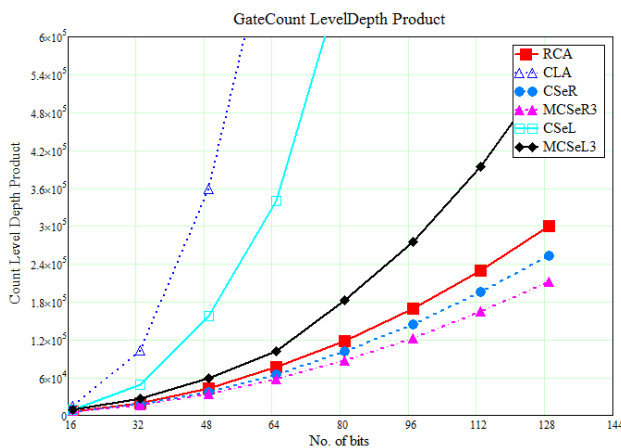**Fig.7 Gate level depth vs. no. of bits**

Fig.8 product of gate count and gate level depth vs. no. of bits

5. REFERENCES

[1] Olivieri, M., Design of synchronous and asynchronous variable-latency pipelined multipliers. IEEE transaction on very large scale integration (VLSI) systems, Vol.9: 256-262, 2004.

[2] R.UMA,Vidya Vijayan, M. Mohanapriya, Sharon Paul, Area, Delay and Power Comparison of Adder Topologies, International Journal of VLSI design & Communication Systems (VLSICS) Vol.3, No.1, February 2012.

[3] Shailesh Siddha, Area Efficient 4-Input Decimal Adder Using CSA and CLA, Journal of Science and Technology, Vol. 3 No.2, 2013

[4] R.Uma, 4-Bit Fast Adder Design: Topology and Layout with Self-Resetting Logic for Low Power VLSI Circuits, International journal of advanced engineering sciences and technologies, Vol. 7, No.2, 2011.

[5] Deepa Sinha, Tripti Sharma, K. G. Sharma, B. P. Singh, Ultra Low Power 1-Bit Full Adder, International Symposium on Devices MEMS, Intelligent Systems & Communication (ISDMISC) 2011 Proceedings published by International Journal of Computer Applications® (IJCA)

[6] Pudi, V. and Sridharan, K. "Low Complexity Design of Ripple Carry and Brent–Kung Adders in QCA ", IEEE Transactions on Nanotechnology, Volume:11 , Issue: 1, 2012.

[7] Jian-Fei Jiang; Zhi-Gang Mao; Wei-Feng He; Qin Wang, "A New Full Adder Design for Tree Structured Arithmetic Circuits", 2nd International Conference on Computer Engineering and Technology(ICCET), Vol.4, pp.246-249,2010.

[8] SubodhWairya, Rajendra Kumar Nagaria, and Sudarshan Tiwari, Performance Analysis of High Speed Hybrid CMOS Full Adder Circuits for Low Voltage VLSI Design, VLSI Design, Hindawi Publishing Corporation, Volume 2012, Article ID 173079

[9] Shubin, V.V., Analysis and comparison of ripple carry full adders by speed , International Conference and Seminar on Micro/NanoTechnologies and Electron Devices(EDM), pp.132-135, 2010.

[10] Ahmet Sertbaş and R.Selami özbey, A performance analysis of classified binary adder architectures and the vhdl simulations, Istanbul university-Journal of electrical & electronics engineering, vol.4, no.1, 2004

[11] Morris Mano, Michael D. Ciletti, Digital Design With an Introduction to the Verilog HDL, Pearson Education, Inc, fifth edition, 2013.

[12] Stephen Brown and Zvonko Vranesic, Fundamentals of digital logic with VHDL design, McGraw Hill, third edition, 2009.

[13] M.Rajesh, R.Manikandan, Efficient Implementation of Carry Free Select Adder, Journal of Science and Technology ,Vol. 3, No. 2, 2013.

[14] Padma Devi, Ashima Girdher, Balwinder Singh, Improved Carry Select Adder with Reduced Area and LowPower Consumption, International Journal of Computer Applications, Vol. 3, No.4, 2010.