International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 2 (2015) pp. 4865-4873

http://www.ripublication.com

Area And Power Efficient Analysis In Various Types Of Adder

R.Dhayabarani[1]

[1]Associate Professor, Department of Electronics and Communication Engineering, V.S.B.Engineering College, Karur [1]dhayamurthy@yahoo.com

R.S.D.Wahida Banu [2]

[2]Professor, Department of Electronics and Communication Engineering, Govt. college of Engineering, Salem
[2]drwahidabanu@gmail.com

Abstract

Addition operation is most frequently used in general purpose computation. Multiplication, subtraction and division can be performed using addition. Addition is a very crucial operation because it involves propagating a carry signal from each bit to its higher bit position. This gives substantial circuit delay. The critical delay path determines the system overall speed. Reducing the power consumption of the designed adder is essential because of increasing status of computing and communication systems[1]. Hence adder is extremely important to study the impact of power speed and area when designing the digital subsystems. Improving performance of the digital adder would greatly move forward the execution of binary operations inside a circuit that compromise such blocks. The performance of a digital block is estimated by analyzing its power dissipation, layout area and its operating speed.

Keyword: VHDL, Ripple Carry Adder, Manchester Carry-Chain Adder (MCA), Carry Select Adder, Carry Save Adder, Kogge_stone_adder and Brent_Kung adder.

1. Introduction

The implementation of several types of adders and their characteristics and performance are compared and found which type of adder shows less power and area. These adders have different Structures, Size and Power Dissipation. Arithmetic Unit

is important block in digital system like Digital Signal Processor, Microprocessor, Microcontroller and other Data Processing Units. Adders are more critical hardware unit for the implementation of arithmetic unit and multiplier block. Addition operations are used in complement operation, encoding, decoding, and multiplication etc. In general, addition is a process which involves two numbers is added and carry will be generated. All complex adder architectures are constructed from its basic building like half adder and full adder. In this paper, the various types of adders are synthesized using 45nm from Taiwan Semiconductor Manufacturing Corporation (TSMC) using RTL compiler from Cadence and then the performance of various adders are calculated and compared.

2. Basic Construction Of Adder

The design of various adders like Ripple Carry Adder (RCA), Carry Look-Ahead Adder (CLA), Manchester Chain Adder (MCA), Carry Select Adder (CSL), Carry Save Adder (CSA, Kogge_stone_adder (KSA), Brent_Kung adder (BKA) are discussed below.

The combinational circuit that adds only two bits is called "Half-adder". To add more than one bit, we have to give a way for carry signal between bit positions. This operation is called a "Full-adder".

Full Adder is a logic circuit that adds a pair of bits of two numbers expressed in binary form and carry from previous stage producing a sum and new carry. It is also called a three input adder.

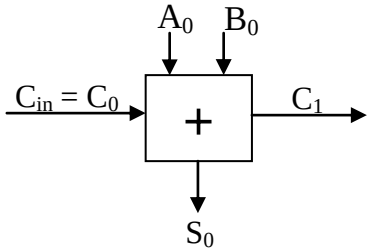
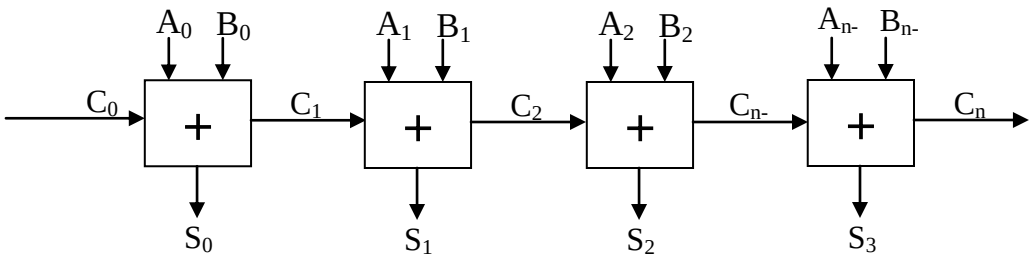


Fig 1: Full Adder Symbol

2.1 Ripple Carry Adder (RCA)

The ripple carry adder is constructed by cascading full adders (FA) in series. One full adder is responsible for the addition of two binary digits at any stage of the ripple carry. The carryout of one stage is fed directly to the carry-in of the next stage.



.

Fig 2: 4bit Ripple Carry Adder (RCA)

This is very simple adder and can be used to add unrestricted bit length numbers. But it is not very efficient when large bit numbers are used. Main drawback of this adder is delay increases linearly with bit length.

2.2 Carry Look-Ahead Adder (CLA)

Carry Look-ahead Adder calculating the carry signals in advance based on the input signals. It reduces carry propagation time.

The working principle of carry look-ahead adder can understand by manipulating the Boolean expression dealing with the full adder. The Propagate (P) and Generate (G) in a full adder are given as:

$P_i = A_i \oplus B_i$ Carry Propagate

$G_i = A_i B_i$ Carry Generate

Both propagate and generate signals depend only on the input bits and will be valid after one gate delay. The output sum and the carryout are given by

$$S_i = P_i \oplus C_{i-1} \quad (1)$$

$$C_{i+1} = G_i + P_i C_i \quad (2)$$

The above equations show that a carry signal will be generated in two ways

If both bits A_i and B_i are 1.

If either A_i or B_i is 1 and the carry-in C_i is 1.

Let's apply these equations for 4 bit adder

$$C_1 = G_0 + P_0 C_0 \quad (3)$$

$$C_2 = G_1 + P_1 C_1 = G_1 + P_1 (G_0 + P_0 C_0)$$

$$= G_1 + P_1 G_0 + P_0 P_1 C_0 \quad (4)$$

$$C_3 = G_2 + P_2 C_2 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0 \quad (5)$$

$$C_4 = G_3 + P_3 C_3 \quad (6)$$

$$= G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_0$$

The general expression is

$$C_{i+1} = G_i + P_i G_{i-1} + P_i P_{i-1} G_{i-2} + \dots\dots\dots P_i P_{i-1} \dots\dots\dots P_2 P_1 G_0 + P_i P_{i-1} \dots\dots\dots P_1 P_0 C_0.$$

From the above expressions, we can say that $C_{2,} C_3$ *and* C_4 does not have any correlation with previous carry-in. Hence, C_4 doesnot need to wait for $C_3$to propagate. C_4 can reach steady state, after computing C_0.

2.3 Manchester Carry-Chain Adder (MCA)

The Manchester Carry-Chain Adder (MCA) is a chain of pass-transistors that are used to implement the carry chain. During Precharge, all intermediate nodes are charged to V_{DD}. During the evaluation phase, C_{out} is discharged if there is an incoming carry C_{in} and the previous propagate signals ($P_0 \ldots P_{k-1}$) are high.

Four diffusion capacitances are present at each node, but distributed RC chain results in a delay that is quadratic with the number of bits.

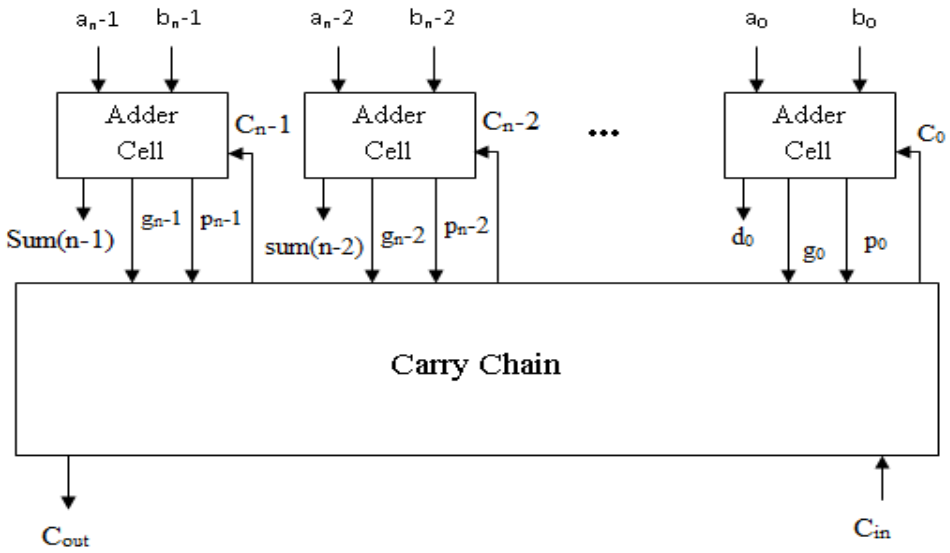


Fig 3: N bit Manchester Carry-Chain Adder (MCA)

2.4 Carry Select Adder (CSL)

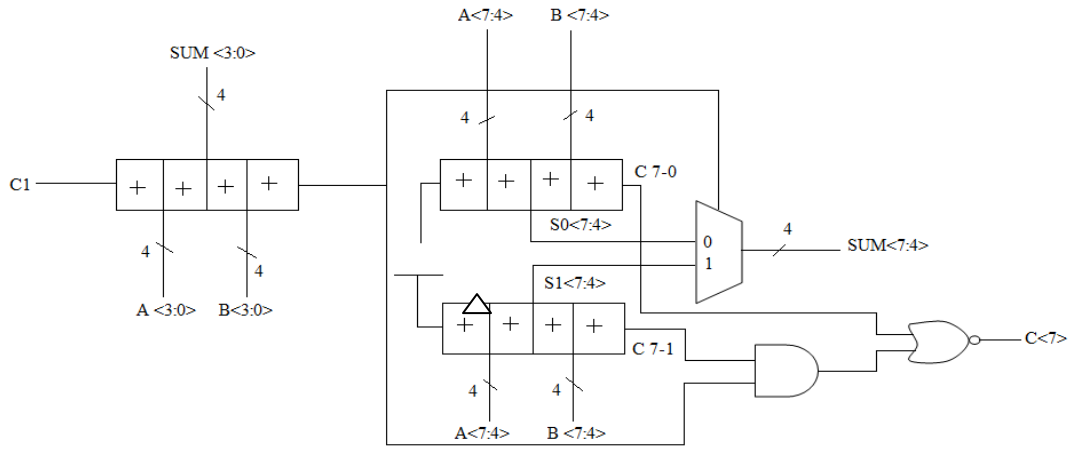


Fig. 4: 8 bit Carry Select Adder with 4 bit sections

Carry Select Adder is divided into sectors, each of which except for the least significant performs two additions in parallel, one assuming a carry-in of zero and other carry-in of one.

The 16bit carry select adder is divided into sectors of length 1, 2, 3, 4 and 6 proceeding from least significant to most significant bit. Within the sector, there are two 4bit ripple carry adders receiving the same data inputs but different carry-in [2].

Adder has a carry-in of zero and a lower adder has carry-in of one. The actual carry-in from the preceding sector selects one of the two adders. If the carry-in is zero then the sum and carry-out of the upper adder is selected. If the carry-in is one then the sum and carry-out of the lower adder is selected.

Carry Select adder is used to calculate alternate results in parallel and subsequently selecting the correct result with single or multiple stage hierarchical techniques [5]. Carry select adder increases its area requirements. In carry select adders, sum and carry bits are calculated for two input "0" and "1". Once the carry-in is delivered, the correct computation is chosen by using MUX to produce the desired output. Instead of waiting for the carry-in to calculate the sum, the sum is directly giving output after getting carry-in as input. Therefore the time taken to calculate the sum is reduced [4].

Carry select adder can be divided into equal or unequal sections. Figure 4 shows the implementation of an 8 bits carry-select adder with 4 bit sections. For each section, the calculation of two sums is accomplished using two 4bit ripple carry adders. One of these adders is fed with a 0 as carry-in and other is fed with 1. Based on real carryout of the previous section, the correct sum is chosen using multiplexer. This is expanded to any length.

2.5 Carry Save Adder (CSA)

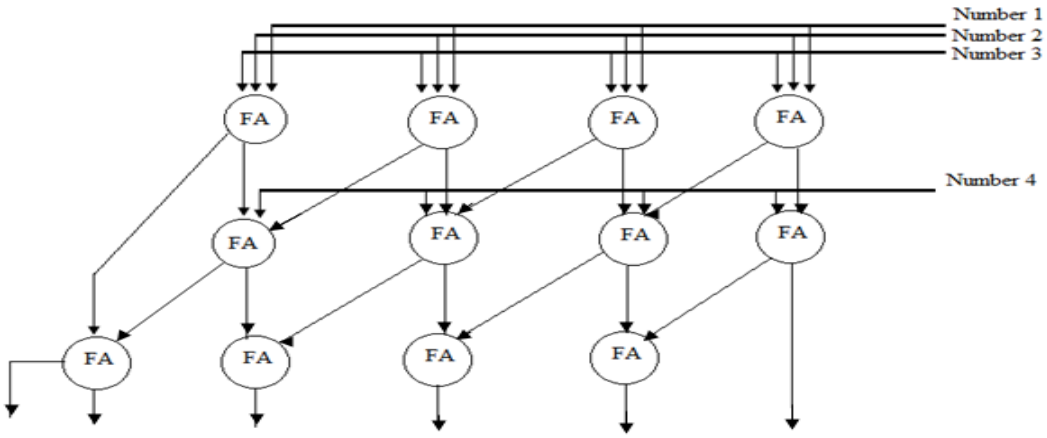


Fig. 5: Four bit Carry Save Adder

Carry Save Adder consists of a ladder of standalone full adders and carryout a number of partial additions. In this adder, the carry has a higher power of 2 and routed to the next column. While doing addition operation using carry save adder technique, the time and logic are saved [3].

In this method, the first 3 numbers a row of full adders are used. A row of full adders are added for each additional number. The final results SUM and CARRY are summed with a carry propagate adder or any other type of adder [6].

2.6 Brent Kung Adder

The Brent-Kung adder is a parallel prefix adder. It is simple and design of parallel adder that addresses the problems of connecting gates in a way to minimize chip area. The operation of this adder depends on group carry propagate and generate signals. A group of bits generate a carry when its carry is independent of the carry-in and propagate a carry when its carryout is true [13].

2.7 Kogge-Stone Adder (KSA)

Kogge-stone adder (KSA) is a parallel prefix form of carry look-ahead adder. It can be represented as a parallel prefix graph consisting of carry operator nodes. It is the fastest adder with focus on design time. The better performance of Kogge-stone adder is minimum logic depth and bounded fan-out. It occupies large silicon area.

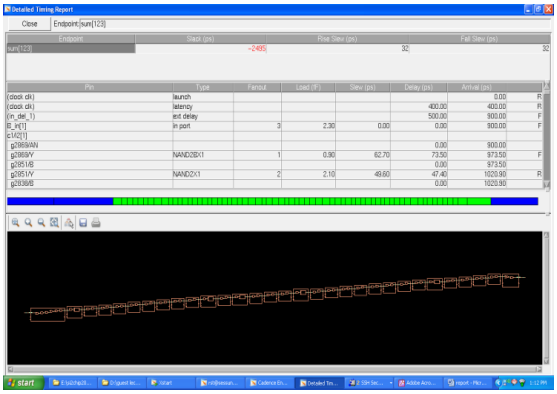
3. Results And Conclusion

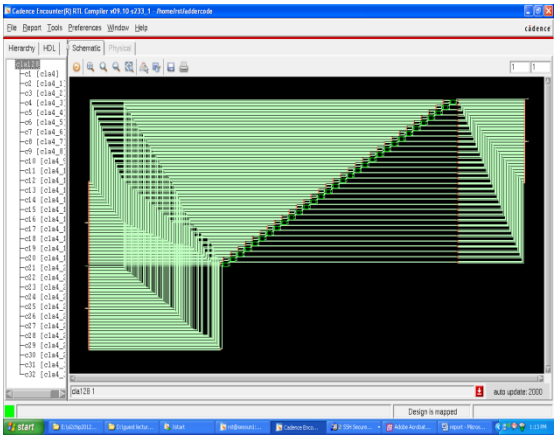
The various adders are compared for different parameters like, Silicon Area, Power Dissipation and Timing [12]. An overall performance evaluation and comparison is conducted based on 45nm from Taiwan Semiconductor Manufacturing Corporation (TSMC) using RTL compiler from Cadence. The Table 1 below shows the area and power required for various adder structures. The Carry Save Adder (CSA) occupies the smallest area as compared to other adders and less power dissipation. The CSA is particularly suitable for applications where the area saving are critical.

Table 1: Performance of Various types of adderTypes	**Bits**	**Area (um2)**	**Power (nW)**	**Timing (ps)**
Ripple Carry Adder	16	215	16975	2666
	32	386	31731	4122
	64	791	65042	6908
	128	1536	126731	10898
Carry Select Adder	16	149	10269	2096
	32	354	25663	3038
	64	612	49227	4244
	128	1165	86991	7200
Carry Save Adder	16	32	1542	750
	32	41	2044	900
	64	57	4116	1732
	128	79	4040	2023
Carry Look Ahead Adder	16	179	11453	2099
	32	388	26888	2348
	64	423	53241	3068
	128	1473	104393	4595
Manchester Adder	16	197	15514	2742

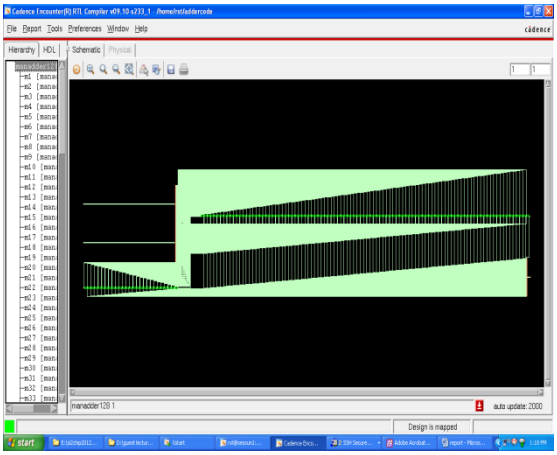
Table 1: Performance of Various types of adderTypes	**Bits**	**Area (um2)**	**Power (nW)**	**Timing (ps)**
	32	276	17933	4746
	64	821	68533	6880
	128	1591	134505	12493
Kogge_Stone Adder	16	281	13555	2099
	32	452	16235	2487
	64	982	58324	3246
	128	1832	123758	5876
Brent_Kung Adder	16	217	12001	2099
	32	402	11678	1989
	64	867	28956	2267
	128	1689	118966	5679

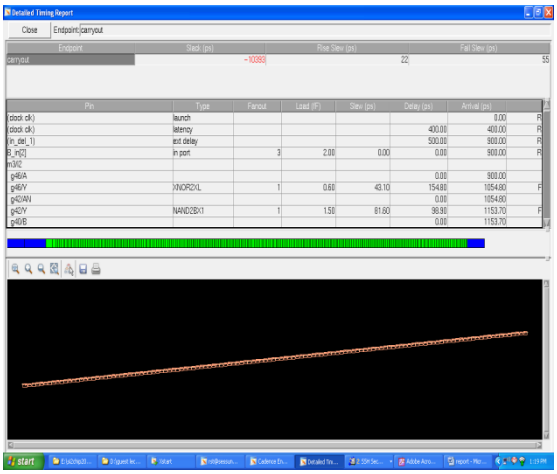
CLA128
Critical Path:



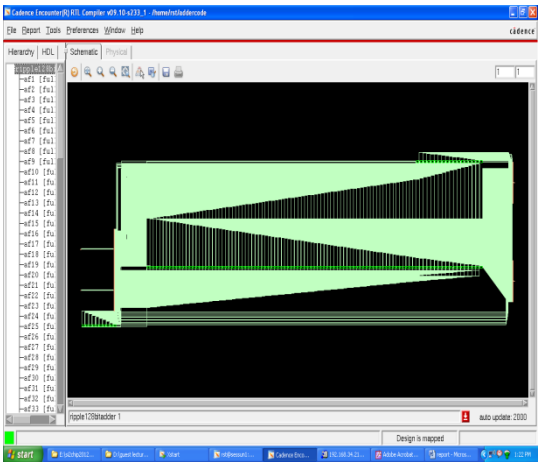


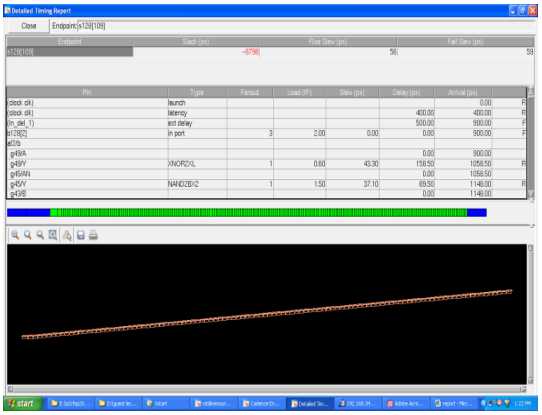
MANCHESTER 128
Critical path:





RCA 128:
Critical path:





4. References

1] Wayne Wolf, Modern VLSI Design, 2nd edition, Prentice Hall PTR, Upper Saddle River, NJ, 1998.

2] O. J. Bedrij "Carry-select adder", *IRE Trans. Electron. Comput.*, pp.340 -344 1962

3] B. Ramkumar , H. M. Kittur and P. M. Kannan "ASIC implementation of modified faster carry save adder", *Eur. J. Sci. Res.*, vol. 42, no. 1, pp.53 -58 2010

4] T. Y. Ceiang and M. J. Hsiao "Carry-select adder using single ripple carry adder", *Electron. Lett.*, vol. 34, no. 22, pp.2101 -2103 1998

5] Y. Kim and L.-S. Kim "64-bit carry-select adder with reduced area", *Electron. Lett.*, vol. 37, no. 10, pp.614 -615 2001

6] J. M. Rabaey *Digtal Integrated Circuits—A Design Perspective*, 2001 :Prentice-Hall

7] Y. He , C. H. Chang and J. Gu "An area efficient 64-bit square root carry-select adder for low power applications", *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 4, pp.4082 -4085 2005

8] N. Weste and D. Harris *CMOS VLSI Design*, 2004 :Addison Wesley

9] Z. Chen and I. Koren "Techniques for yield enhancement of VLSI adders", *Proc. Int. Conf. Appl. Specific Array Process.*, pp.222 -229 1995

10] T. Y. Chang and M. J. Hsiao "Carry-select adder using single ripple-carry adder", *Electron. Lett.*, vol. 34, no. 22, pp.2101 -2103 1998

11] K. Youngjoon and L.-S. Kim "A low power carry-select adder with reduced area", *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 4, pp.218 -221 2001

12] J. C. Lo "Novel area-time efficient static CMOS totally self-checking comparator", *IEEE J. Solid-State Circuits*, vol. 28, no. 2, pp.165 -168 1993

13] B. Parhami *Computer Arithmetic: Algorithms and Hardware Design*, 2000 :Oxford University Press.

14] F. C. Cheng , S. H. Unger , M. Theobald and W.-C. Cho "Delay- insensitive carry-lookahead adders", *Proc. Int. Conf. VLSI Design*, pp.322 -328 1997