A new Simulation of a 16-bit Ripple Carry Adder and a 16-bit Skip Carry Adder

Akbar Bemana

Abstract: Simulation of a Full Adder (FA) and 16-bit adder are represented in this paper. Ripple Carry Adder (RCA) and Skip Carry Adder (SCA) are used to simulated 16-bit adder. SCA is simulated for different structures such as 2, 4 and 8-blocks. Simulation results show that SCA is faster than RCA. Further more, 8-block SCA is faster than 4-block SCA and 4-block SCA is faster than 2-block SCA too.

1 Introduction:

Computation speeds have increased dramatically during the past three decades resulting from the development of various technologies. The execution speed of an arithmetic operation is a function of two factors. The first is the circuit technology and the second is the used algorithm. Algorithm goals include the adder structures and the Data format (i.e. such as redundant, 2-compliment format and etc). It can be rather confusing to discuss both factors simultaneously; for instance, a RCA implemented in GaAs technology may be faster than a SCA implemented in CMOS. Further more, in any technology, logic path delay depends upon many different factors: The number of gates through which a signal has to pass before a decision is made, the logic capability of each gate, cumulative distance among all such serial gates, the electrical signal propagation time of the medium per unit distance, etc. Because the logic path delay is attributable to the delay internal and external to logic gates, a comprehensive model of performance would have to include technology, distance, placement, layout, electrical and logical capabilities of the gates. It is not feasible to make a general model of arithmetic performance and include all these variables. Since addition is the basic operation of most computations, we can increase speed of the adders to increasing the computation speed. The purpose of this paper is representing a comparison between the RCA and the SCA which are used in design of the arithmetic operators. This paper is organized as follow:

- *Akbar Bemana* Islamic Azad University Dashtestan branch, Borazjan, Iran

In section 2, simulating of a Manchester-type FA with non-complimentary carry is represented. A 16-bit RCA and a 16-bit SCA are simulated in section 3 and 4 respectively. In later section, SCA is simulated for 2, 4 and 8-block structures. Then 16-bit adders are compared with together. In section 5, numerical results are presented and At the end a conclusion is presented.

2 1-bit Full Adder:

It is the generic cell used not only to perform addition but also arithmetic multiplication division and filtering operation. Figure 1 shows the Manchester-type FA with non-complimentary carry structure. The adder cell receives two operands Xi and Yi, and an incoming carry Ci. It computes the sum (Si) and the outgoing carry Co.

$$Co = Xi \cdot Yi + Xi \cdot Ci + Ci \cdot$$

$$Yi = Pi \cdot Ci + Gi$$

Where

Pi = Yi XOR Xi is the PROPAGATION signal, Gi = (Xi And Yi) is the GENERATION signal, Si = Xi XOR Yi XOR Ci.

In this section we simulated the Manchester-type FA with non-complimentary carry using “HSPICE” software. The transistor surfaces are 2/2 and 4/2 for NMOS/ PMOS respectively. Figure 2 shows Simulation results. This figure shows the input carry, output carry and sum (Si).

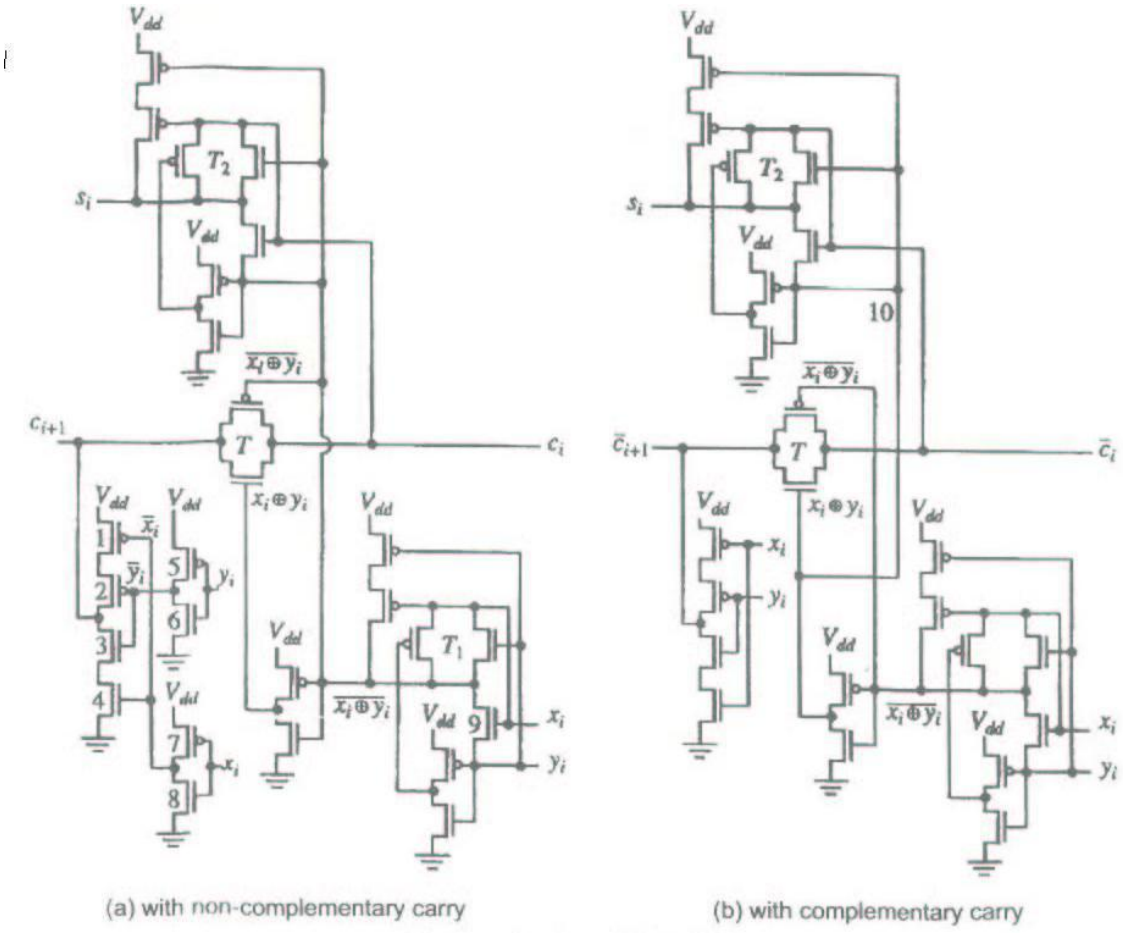

Figure 1- Manchester-type Full Adders .

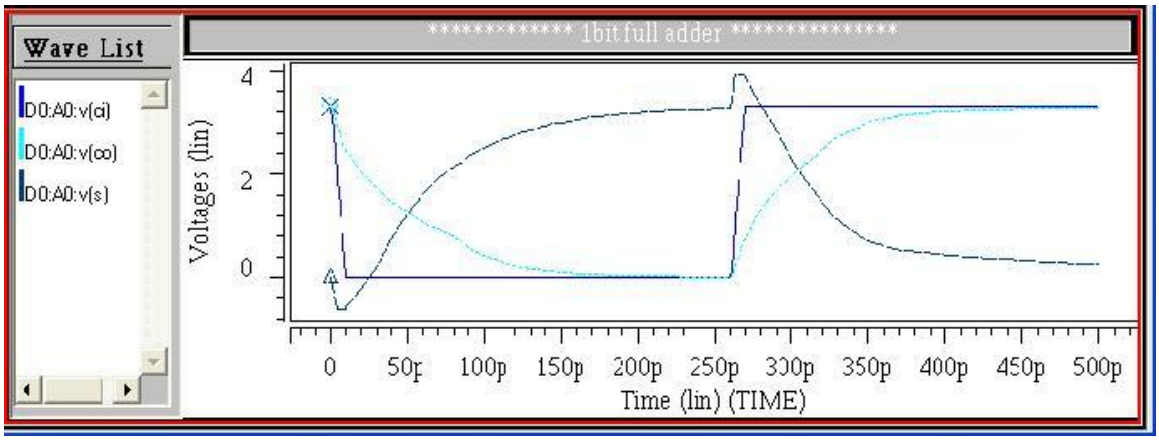

Figure 2- Simulation results of 1-bit FA.

3 Ripple Carry Adder:

In most computers, the augmented operand is replaced by the sum, whereas the addend is unchanged. High speed adders are not only for addition but also for subtraction, multiplication and division. The speed of a digital processor depends heavily on the speed of adders. The adders add vectors of bits and the principal problem is to speed- up the carry signal. A traditional and non optimized 16-bit adder can be made by the use of the generic one-bit adder cell

connected one to the other (figure 3). It is called RCA. In this case, the sum resulting at each stage need to wait for the incoming carry signal to perform the sum operation. The carry propagation can be speed-up in two ways. The first – and most obvious– way is to use a faster logic circuit technology. The second way is to generate carries by means of forecasting logic that does not rely on the carry signal being rippled from stage to stage of the adder. This adder is simulated in worst condition (i.e. Maximum propagation delay). Maximum propagation delay occurs when Xi=1 , Yi=0 and Cin=1 ; it takes the maximum time for transmitting the input carry to the output. Figure 4 shows Simulation results. This figure shows the input carry (Cin), output carry (C_{16}) and S_{16}.

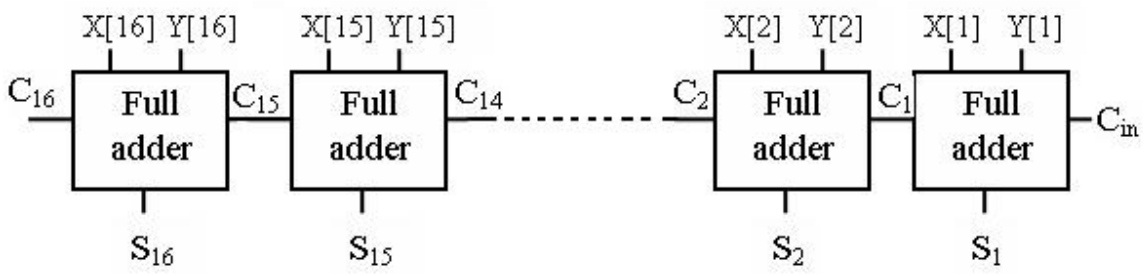

Figure 3- 16-bit RCA.

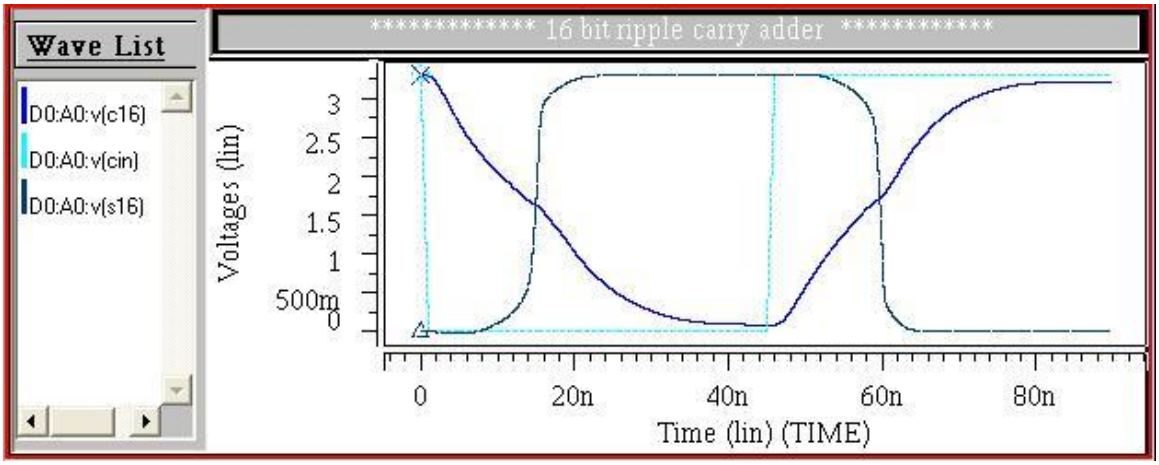

Figure 4- Simulation results of 16-bit RCA .

4 Skip Carry Adder :

With a RCA, if the input bits Xi and Yi are different for all position i, then the carry signal is propagated at all positions and the addition is completed when the carry signal has propagated through the whole adder. In this case, the RCA is as slow as it is large. Actually, RCA are fast only for some configurations of the input words, where carry signals are generated at some positions. Depending on the position at which a carry signal has been generated, the propagation time can be variable. Carry Skip Adders take advantage both of the generation or the propagation of the carry signal. They are divided into blocks, where a special circuit detects quickly if all the bits to be added are different (Pi = 1 in the entire block). The signal produced by this circuit will be called block propagation signal. If the carry is propagated at all positions in the block, then the carry signal entering into the block can directly bypass it and so be transmitted through a multiplexer to the next block. As soon as the carry signal is transmitted to a block, it starts to propagate through the block, as if it had been generated at the beginning of the block. Figure 5 shows the structure of a 16-bits skip carry adder, divided into 2, 4 and 8-blocks.

OPTIMISATION TECHNIQUE WITH BLOCKS OF EQUAL SIZE:

It becomes now obvious that there exist a trade-off between the speed and the size of the blocks. In this part we analyze the division of the adder into blocks of equal size. We denote k_1 the time needed by the carry signal to propagate through an adder cell (one FA propagates), and k_2 the time it needs to skip over one block. Suppose the N-bit Carry Skip Adder is divided into M blocks, and each block contains P adder cells. The actual addition time of a RCA depends on the configuration of the input words. The completion time may be small but it also may reach the worst case, when all adder cells propagate the carry signal. In the same way, we must evaluate the worst carry propagation time for the Carry Skip Adder. The worst case of carry propagation is for Xi=1 and Yi=0 and Cin=1 inputs.

In SCA, Consequently, in the first block, the last adder cells must wait for the carry signal, which comes from the first cell of the first block. When going out of the first block, the carry signal is distributed to the 2nd, 3rd and last block,

$$T_w = T + T' = k_1 \times P + k_2 (M - 1)$$

$$\Rightarrow T_w = k_1 \frac{N}{M} + k_2 (M - 1)$$

So that the function to be minimized is:

$$f(x) = k_1 \frac{N}{x} + k_2 (x - 1)$$

where it propagates. In these blocks, the carry signals propagate almost simultaneously (we must account for the multiplexer delays). We can formalize that the total adder is made of N adder cells. It contains M blocks of P adder cells. The total of adder cells is then

$N=M.P$

The time T needed by the carry signal to propagate through P adder cells is

$T=k_1.P$

The time T' needed by the carry signal to skip through M-1 adder blocks is

$T'=k_2.(M-1)$

The problem to solve is to minimize the worst case delay (T_w) which is:

The minimum is obtained for: $x_{\min} = \sqrt{k_1 N / K_2}$

To summarize, if in a block all Ai's?Bi's, then the carry signal skips over the block. If they are equal, a carry signal is generated inside the block and needs to complete the computation inside before to give the carry information to the next block .

Figure 6, 7 and 8 show Simulation results of 2, 4 and 8-block SCA respectively. This figures show the input carry (Cin), output carry (C_{16}) and S_{16}.

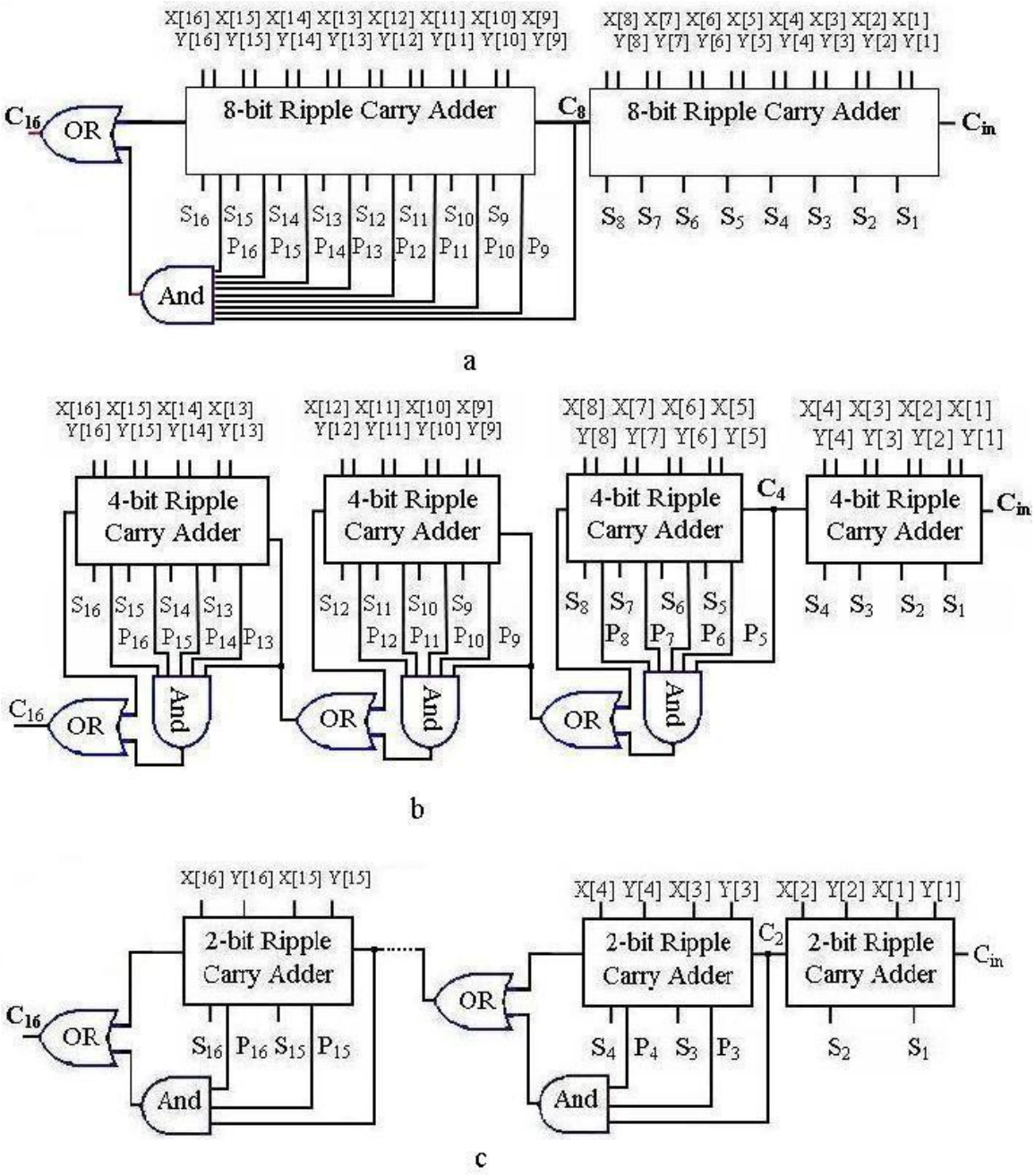

Figure 5- a: 2-block SCA, b: 4-block SCA and c: 8-block SCA .

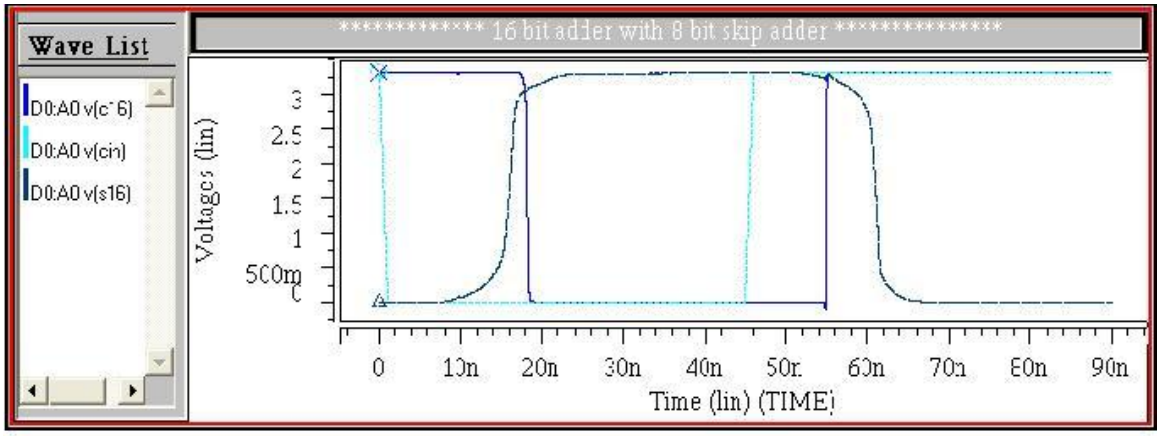

Figure 6- Simulation results of 2-block SCA .

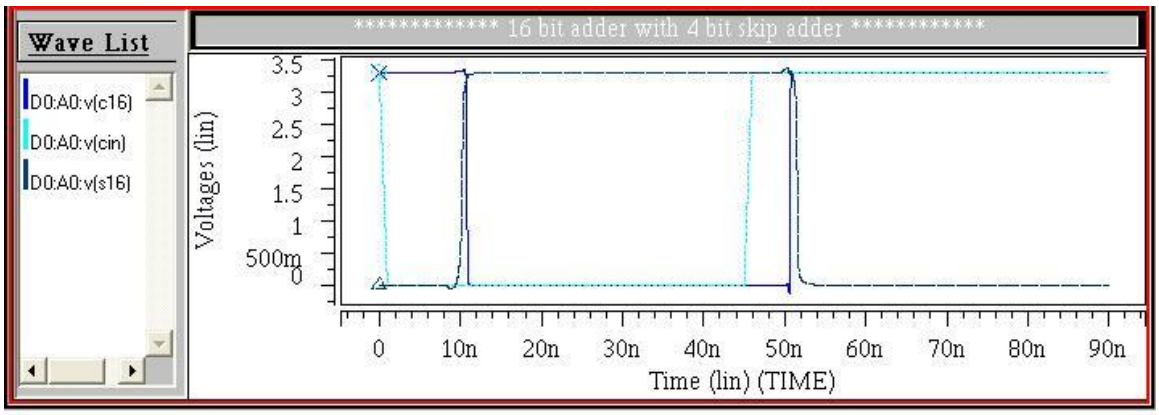

Figure 7- Simulation results of 4-block SCA .

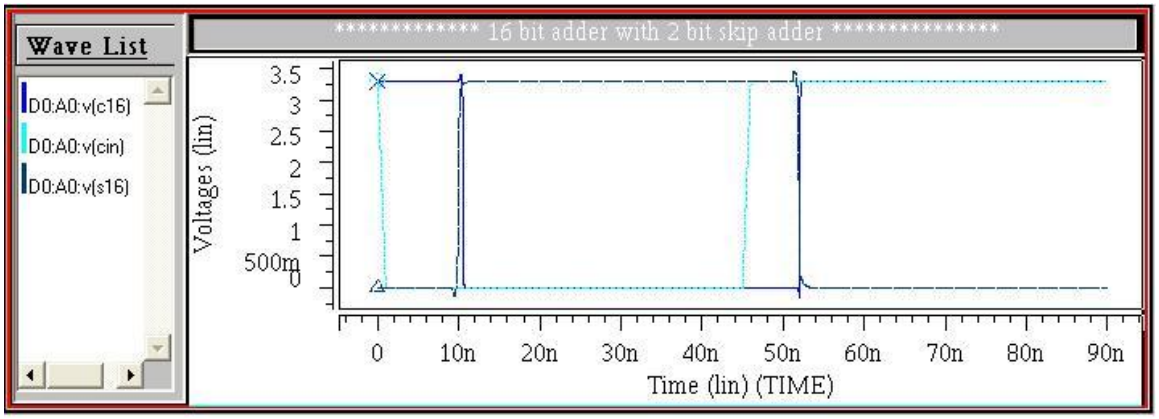

Figure 8- Simulation results of 8-block SCA .

5 Numerical Results :

Table 1 shows the comparing results of different type of 16-bit adders. With respect to the simulation results is known that the maximum operation frequency (MOF) for RCA obtains of C_{16} rise time's (T_r) , fall time's (T_f) and delay time (T_D) but for SCA maximum operation frequency is defined of propagation delays (T_{PHL} of C_{16} & T_{PLH} of S_{16}) . the maximum operation frequency represent in table.

Adder	MOF (MEG Hz)
16 bit RCA	20.2429
2-block 16 bit SCA	30.2114
4-block 16 bit SCA	60.9756
8-block 16 bit SCA	61.7283

6 Conclusion :

The mentioned adders compared at several point of view, consumption power, surface on wafer and speed. The SCA take more surface and have more power consumption than RCA. The surface increased when the number of SCA blocks increase. At point of speed view, SCA was faster than RCA. Further more, 8-block SCA was faster than 4-block SCA and 4-block SCA was faster than 2-block SCA too.

References :

[1] R. Govindan, J. M. Hellerstein, W. Hong, S. Madden, M. Franklin, and Scott Shenker. "**16-bit Ripple Carry Adder**". Tech. Report 02-771, USC/CS, 2008.

[2] Z.Tu,W.Liang." 16-bit Different Structures Skip Carry Adder". Australian National University,2006

[3] R.Rosemark ,W.C.Lee ,B.Urgaonkar, "Optimizing Simulation of Ripple carry adder".

[4] D.J.Abadi,S.Madden,W.Lindner," REED: Robust, Efficient Filtering and Event Detection in Sensor Networks". Proceedings of the 31st VLDB Conference, Trondheim, Norway, 2005

[5] Muhammad Umer," Selectivity-aware Query Processing in Sensor Networks", Sensys'07 Doctoral Colloquium, November 6, 2007, Sydney, Australia.

[6] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. "TinyDB: An acquisitional query processing system for sensor networks". *ACM TODS*, 30(1), November 2005.

[7] A. Demers, J. Gehrke, R. Rajaraman, N. Trigoni, and Y. Yao. "The cougar project: A work-in-progress report". *SIGMOD Record*, 32(4), 2003.

[8] N. Trigoni, Y. Yao, J. Gehrke, R. Rajaraman, and A. Demers." Multi-query optimization for sensor networks". In *Proc. of DCOSS*, 2005.

[9] S.Xiang ,H.B.Lim ,K.L.Tan," Impact of Multi query Optimization in Sensor Networks", Proceedings of the 3rd International Workshop on Data Management for Sensor Networks (DMSN'06), Seoul, South Korea, 2006.

[10] O.Gnawaliz, R.Govindanz, J.Heidemannx , "Implementing a Sensor Database System using a Generic Data Dissemination Mechanism",IEEE,2005.

[11] Yang Sun ," Query Processing for Sensor Network ", University ofWaterloo,2006.

[12] M.Demirbas, H.Ferhatosmanoglu," Peer-to-Peer Spatial QueriesinSensorNetworks".

1. [13] I.F.Akyildiz et al., "Wireless Sensor Networks: a survey", Computer Networks, Vol. 38, pp. 393-422, March 2002.

[14] . Yao and J. Gehrke." Query processing for sensor networks".In *Proc. ofCIDR*,2003