RESEARCH ARTICLE OPEN ACCESS

FPGA Based Implementation for Ripple Carry Adder with Reduced Area and Low Power Consumption

Amit Jain, Nitin Meena
IES College of Technology Bhopal, M.P.

Abstract
A multiplier is one of the key hardware blocks in most digital and high performance systems such as FIR filters, digital signal processors and microprocessors etc. With advances in technology, many researchers have tried and are trying to design multipliers which offer either of the following- high speed, low power consumption, regularity of layout and hence less area or even combination of them in multiplier. However area and speed are two conflicting constraints. So improving speed results always in larger areas. So here we try to find out the best trade off solution among the both of them.

While comparing the adders we found out that Ripple Carry Adder had a smaller area while having lesser speed, in contrast to which Carry Select Adders are high speed but possess a larger area. And a Carry Look Ahead Adder is in between the spectrum having a proper tradeoff between time and area complexities.

Keywords: -Ripple Carry Adder, Carry Select Adder, Carry Skip Adder, Carry Look Head Adder FIR Filter.

I. INTRODUCTION

Addition usually impacts widely the overall performance of digital systems and a crucial arithmetic function. In electronic applications adders are most widely used. Applications where these are used are multipliers, DSP to execute various algorithms like FFT, FIR and IIR. Wherever concept of multiplication comes adders come in to the picture. As we know millions of instructions per second are performed in microprocessors. So, speed of operation is the most important constraint to be considered while designing multipliers.

Improved adders generate carries simultaneously [1]. These adders employ the principle that the carry from each bit position may be generated independently as an explicit function of all the less significant addend and augend bits. However, because of the inherent limitations in available components, the constructors of simultaneous carry- generation adders are not always practical.

Multiplication is a fundamental operation in most signal processing algorithms. Multipliers have large area, long latency and consume considerable power. Therefore low-power multiplier design has been an important part in low- power VLSI system design. There has been extensive work on low-power multipliers at technology, physical, circuit and logic levels. A system's performance is generally determined by the performance of the multiplier because the multiplier is generally the slowest element in the system. Furthermore, it is generally the most area consuming. Hence, optimizing the speed and area of the multiplier is a major design issue. However, area and speed are usually conflicting constraints so that improving speed results mostly in larger areas. As a result, a whole spectrum of multipliers with different area- speed constraints has been designed with fully parallel.

The rest of the paper is organized as follows. In section 2, a brief about ripple carry adder, carry select adder and carry skip is given. In section 3 carry look head adder is introduced along with partitioning methodology. Also a new architecture with clock sharing is introduced. Section 4 provides the simulation results obtained. Section 5 provides the conclusion and future scope obtained.

II. ADDITION ALGORITHM

- **Ripple Carry Adder**

This procedure is an asynchronous addition used in the first electronic computers. Addition of binary numbers of several digits is accomplished in the same manner as that of decimal numbers.

Concatenating the N full adders forms N bit Ripple carry adder. In this carry out of previous full adder becomes the input carry for the next full adder. It calculates sum and carry according to the following equations. As carry ripples from one full adder to the other, it traverses longest critical path and exhibits worst case delay.

$$S_i = A_i \oplus B_i \oplus C_i \qquad (1)$$

$$C_{i+1} = A_i \bullet B_i + B_i \bullet C_i + C_i \bullet A_i \qquad (2)$$

Where i = 0, 1, 2, 3 …n-1

RCA is the slowest in all adders but it is very compact in size. If the ripple carry adder is implemented by concatenating N full adders, the delay of such an adder is 2N gate delays from C_{in} to C_{out}. The delay of adder increases linearly with increase in number of bits. Block diagram of RCA is shown in Figure 1.

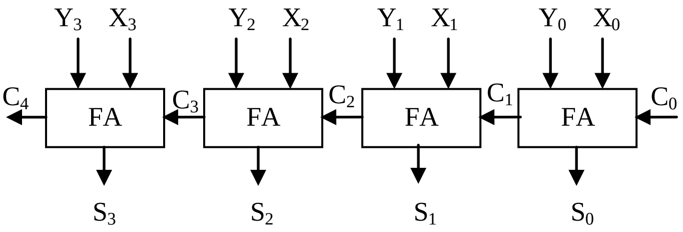

Figure 1: Block Diagram of RCA

- Carry Select Adder

In Carry select adder scheme, blocks of bits are added in two ways: one assuming a carry-in of 0 and the other with a carry-in of 1.This results in two precomputed sum and carry-out signal pairs $(S^0_{i-1:k}, C^0_i ; S^1_{i-1:k}, C^1_i)$, later as the block's true carry-in (C_k) becomes known , the correct signal pairs are selected. Generally multiplexers are used to propagate carries.

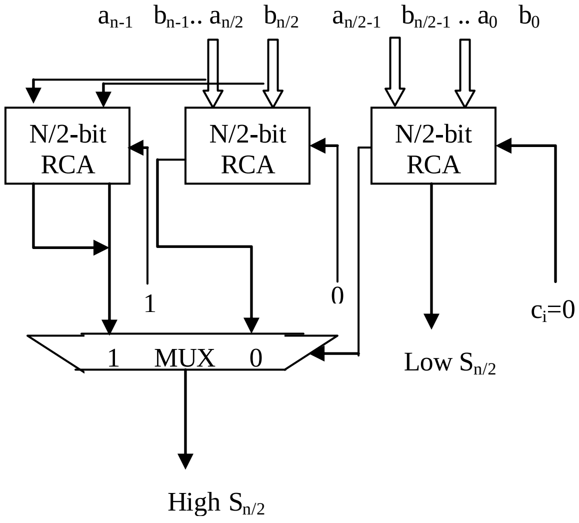
a_{n-1} b_{n-1}.. $a_{n/2}$ $b_{n/2}$ $a_{n/2-1}$ $b_{n/2-1}$ .. a_0 b_0
N/2-bit RCA
N/2-bit RCA
N/2-bit RCA
1
0
c_i=0
1 MUX 0
Low $S_{n/2}$
High $S_{n/2}$

Figure 2: A Carry Select Adder with 1 level using n/2-bit RCA

Because of multiplexers larger area is required.
Have a lesser delay than Ripple Carry Adders (half delay of RCA).
Hence we always go for Carry Select Adder while working with smaller no of bits.

- Carry Skip Adder

A carry skip divides the words to be added in to groups of equal size of k-bits. Carry Propagate *pi* signals may be used within a group of bits to accelerate the carry propagation. If all the pi signals within the group are pi=1, carry bypasses the entire group as shown in figure 2.

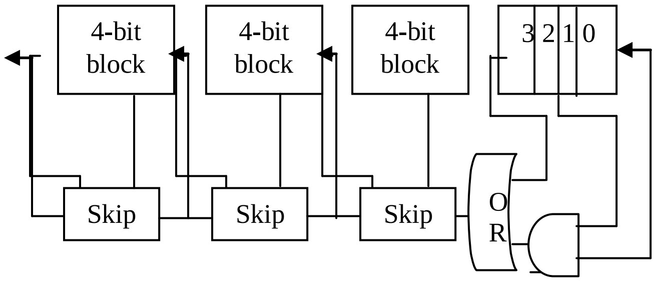

Figure 3: Carry Skip Adder

In this way delay is reduced as compared to ripple carry adder. The worst-case carry propagation delay in a N-bit carry skip adder with fixed block width *b*, assuming that one stage of ripple has the same delay as one skip, can be derived:

Block width tremendously affects the latency of adder. Latency is directly proportional to block width. More number of blocks means block width is less, hence more delay. The idea behind Variable Block Adder (VBA) is to minimize the critical path delay in the carry chain of a carry skip adder, while allowing the groups to take different sizes. In case of carry skip adder, such condition will result in more number of skips between stages.
Such adder design is called variable block design, which is tremendously used to fasten the speed of adder. In the variable block carry skip adder design we divided a 32-bit adder in to 4 blocks or groups. The bit widths of groups are taken as; First block is of 4 bits, second is of 6 bits, third is 18 bit wide and the last group consist of most significant 4 bits.

- Ripple Carry Adder

In ripple carry adders, the carry propagation time is the major speed limiting factor as seen in the previous lesson.
Most other arithmetic operations, e.g. multiplication and division are implemented using several add/subtract steps. Thus, improving the speed of addition will improve the speed of all other arithmetic operations.

Accordingly, reducing the carry propagation delay of adders is of great importance. Different logic design approaches have been employed to overcome the carry propagation problem.
One widely used approach employs the principle of carry look-ahead solves this problem by calculating the carry signals in advance, based on the input signals.
This type of adder circuit is called as carry look-ahead adder (CLA adder). It is based on the fact that a carry signal will be enerated in two cases:
(1) when both bits A_i and B_i are 1, or
(2) when one of the two bits is 1 and the carry-in (carry of the previous stage) is 1.

To understand the carry propagation problem, let's consider the case of adding two n-bit numbers A and B.

In this circuit, the 2 internal signals P_i and G_i are given by:

$$P_i = A_i \oplus B_i \quad (3)$$

$$G_i = A_i \oplus B_i \quad (4)$$

The output sum and carry can be defined as:

$$S_i = P_i \oplus C_i \quad (5)$$

$$C_{i+1} = G_i + P_i C_i \quad (6)$$

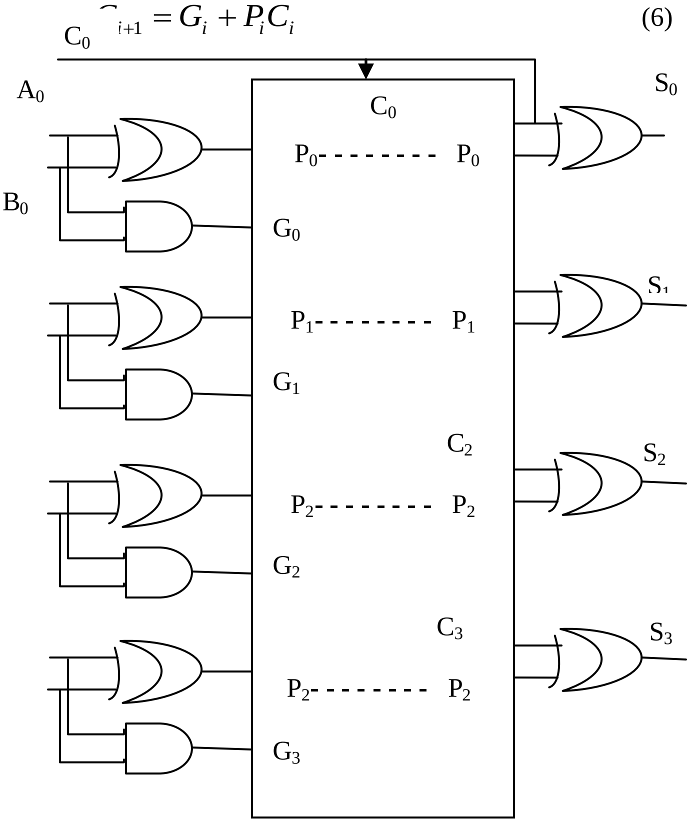

Figure 4: Carry Look head Block

G_i is known as the carry Generate signal since a carry (C_{i+1}) is generated whenever G_i =1, regardless of the input carry (C_i).

P_i is known as the carry propagate signal since whenever P_i =1, the input carry is propagated to the output carry, i.e., C_{i+1}. = C_i (note that whenever P_i =1, G_i =0).

Thus, these signals settle to their steady-state value after the propagation through their respective gates.

Computed values of all the P_i's are valid one XOR-gate delay after the operands A and B are made valid.

Computing the values of P_i and G_i only depend on the input operand bits (A_i & B_i) as clear from the Figure and equations. Computed values of all the G_i's are valid one AND-gate delay after the operands A and B are made valid.

III. RESULT AND SIMULATION

We have implemented Ripple Carry Adder (RCA), Carry Select Adder (CSA), and Carry Look-head Adder (CLA architecture. A comparison between RCA, CSA and CLA design based architecture for 4-bit, 16-bit is given in Table I and Table II respectively.

TABLE I
Comparison between RCA, CSA and CLA design based architecture for 4-bit

	RCA	CSA	CLA
Number of Slices	4 out of 6144	7 out of 6144	6 out of 6144
Number of 4 input LUTs	8 out of 12288	13 out of 12288	10 out of 12288
Maximum combinational path delay	7.087ns	6.431ns	7.435ns

TABLE II
Comparison between RCA, CSA and CLA design based architecture for 16-bit

	RCA	CSA	CLA
Number of Slices	18 out of 6144	28 out of 6144	20 out of 6144
Number of 4 input LUTs	32 out of 12288	52 out of 12288	35 out of 12288
Maximum combinational path delay	14.431ns	10.338ns	14.652ns

Implementing the proposed CLA based architecture has been captured by VHDL and the functionality is verified by RTL and gate level simulation. To estimate the number of slices, flip flop, required time and minimum period information for ASIC design, we have used Xilinx Design Compiler to synthesize the design into gate level. Comparison of practical result up to third decomposition level architecture for 2-D DWT is given in Table I and Table II respectively.

IV. CONCLUSION

We studied about different adders among compared them by different criteria like number of slice and Time so that we can judge to know which adder was best suited for situation. After comparing all we came to a conclusion that Carry Look-head Adders are best suited for situations where Speed is the only criteria. Similarly Ripple Carry Adders are best suited for Low Power Applications. But Among all the Carry Look Ahead Adder had the least Area-Delay product that tells us that, it is suitable for situations where both low power and fastness are a criteria such that we need a proper balance between both as is the case with our Paper.

REFERENCES

[1] K. Rawwat, T. Darwish, and M. Bayoumi, “.A low power carry select adder with reduces area”, Proc. Of Midwest Symposium on Circuits and Systems, pp. 218- 221, 2001.

[2] W. Jeong and K. Roy, “Robust high-performance low power adder”, Proc. of the Asia and South Pacific Design Automation Conference, pp. 503-506, 2003.

[3] Y. Kim and L-S Kim, “64-bit carry-select adder with reduced area”, Electronics Letters, vol. 37, pp. 614-615, May 2001.

[4] O. Kwon, E. Swartzlander, and K. Nowka, “A fast hybrid carry-look ahead/carry-select adder design”, Proc. of the 11th Great Lakes symposium on VLSI, pp.149-152, March 2001.

[5] B. Parhami, Computer Arithmetic, Algorithm and Hardware Design, Oxford University Press, New York, pp.91-119, 2000.

[6] Wang, Y. Pai, C.Song, X., “The design of hybrid carry look ahead/ carry-select adders”, Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on Volume 49, pp.16-24, 2002.

[7] Z. Abid, H. El-Razouk and D.A. El-Dib, “Low power multipliers based on new hybrid full adders”, Microelectronics Journal, Volume 39, Issue 12, Pages 1509-1515, 2008.

[8] Hasan Krad and Aws Yousif Al-Taie, “Performance Analysis of a 32-Bit Multiplier with a Carry-Look-Ahead Adder and a 32-bit Multiplier with a Ripple Adder using VHDL”, Journal of Computer Science 4 (4): 305-308, 2008.

[9] Behnam Amelifard, Farzan Fallah, and Massoud Pedram, "Closing the gap between carry select adder and ripple carry adder: a new class of low-power high-performance adders", in Proc. of IEEE International Symposium on Quality Electronic Design (ISQED), 2005.