International Journal of Advanced Trends in Engineering, Science and Technology(IJATEST)Volume.4,Special Issue.1Dec.2016

A High Speed Multiplier Using CSA In VLSI Circuit Design

Shaik Hameeda Noor[1] K.Sireesha[2] Syed Saleem[3]M.Venkata Subbaiah[4]
U.G.Student, ECE, AITS, Kadapa, India[1] U.G.Student, ECE, AITS, Kadapa, India[2]
Assistant Professor, ECE, AITS, Kadapa, India[3] Assistant Professor, ECE,AITS,kadapa,India[4]

Abstract: *A typical processor central processing unit devotes a considerable amount of processing time in performing arithmetic operations, particularly multiplication operations. Multiplication is one of the basic arithmetic operations and it requires substantially more hardware resources and processing time than addition and subtraction. In fact, 8.72% of all the instruction in typical processing units is multiplication. In this paper, comparative study of different multipliers is done for low power requirement and high speed. The paper gives information of array multiplier which is utilized for multiplication to improve the speed parameter of multipliers. Carry save multiplier suggests one more formula for multiplication which can increase the speed of multiplier by reducing the number of iterations. Xilinx 13.2 is the software used.*

Key Terms: *RCA, CSA, AM and CSM*

1. INTRODUCTION

Multiplication is an important fundamental function in arithmetic logic operation. Computational performance of a DSP system is limited by its multiplication performance and since, multiplication dominates the execution time of most DSP algorithms therefore high-speed multiplier is much desired. Currently multiplication time is still the dominant factor in determining the instruction cycle time of a DSP chip. With an ever-increasing quest for greater computing power on battery-operated mobile devices, design emphasis has shifted from optimizing conventional delay time area size to minimizing power dissipation while still maintaining the high performance . Traditionally shift and add algorithm has been implemented to design however this is not suitable for VLSI implementation and also from delay point of view. Some of the important algorithm proposed in literature for VLSI implementable fast multiplication is array multiplier and Carry save multiplier . This paper presents the fundamental technical aspects behind these approaches. The high speed VLSI can be implemented with different logic style. There are many proposed logics (or) high speed and each logic style has its own advantages in terms of speed.

2. RELATED WORK

2.1 Half Adder

A Half Adder (HA) is a combinational circuit used for adding two bits, a_i and b_i.

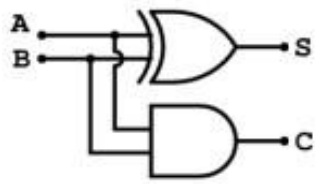

Fig: Half adder

The circuit has two outputs namely sum si and carry output co. Boolean expression refers sum and carry output are given:

Si=ai xor bi
Co=ai and bi

The critical path delay is one gate delay,

and it corresponds to the length of any one of the two paths.

Table 1: Truth Table for Half Adder

INPUT		OUTPUT	
A	**B**	**S**	**C**
0	**0**	**0**	**0**
0	**1**	**1**	**0**
1	**0**	**1**	**0**
1	**1**	**0**	**1**

2.2 Full Adder

A Full Adder (FA) is a combinational circuit used for adding three bits, a_i, b_i and c_i.

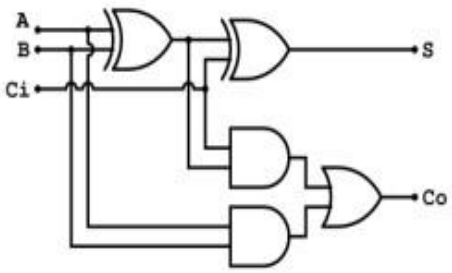

Fig : Full Adder

This circuit has two outputs namely sum s_i and carry output c_o. A 1-bit full adder adds three 1-bit numbers, often written as A, B and C_i here A,B are the operands and C_i is a bit carried in

Table 2: Truth Table for Full Adder

INPUT			OUTPUT	
A	**B**	**C**	**S**	**Cout**
0	**0**	**0**	**0**	**0**
0	**0**	**1**	**1**	**0**
0	**1**	**0**	**1**	**0**
0	**1**	**1**	**0**	**1**
1	**0**	**0**	**1**	**0**
1	**0**	**1**	**0**	**1**
1	**1**	**0**	**0**	**1**
1	**1**	**1**	**1**	**1**

A FA can also be constructed by cascading two HA. A and B inputs are connected to the input of first HA and the sum of first HA is connected as one input to second HA and second input to second HA is given through C_i. The final output sum of second HA is the final sum of FA and carries out of first and second HA is logically to produce final carry.

2.3 Ripple Carry Adder

A ripple adder that adds two N-bit operands requires N full adders. The speed varies linearly with the word length. The RCA implements the conventional way of adding two numbers. In this architecture the operands are added bitwise from the least significant bits (LSBs) to the most significant (MSBs), adding at each stage the carry from the previous stage.

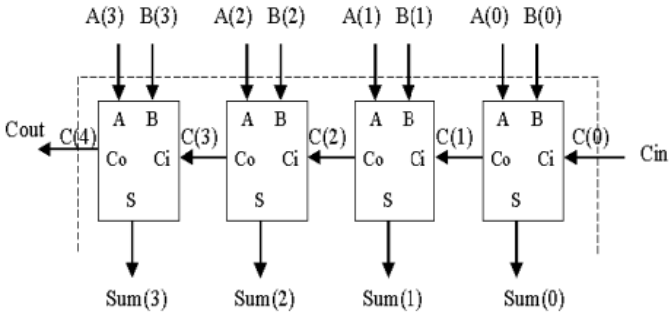

Fig : 4-bit Ripple Carry Adder

Thus the carry out from the FA at stage i goes into the FA at stage (i +1), and in this manner carry ripples from LSB to MSB (hence the name of ripple carry adder). The layout of a RCA is simple, which allows fast design time. However, RCA is relatively slow, since each full adder must wait for the carry bit which is coming from the previous full adder

2.4 Carry save Adder

All the adders discussed above are used for adding two operands, and then propagate carries from one bit position to the next in computing the final sum and are collectively known as carry propagate adders (CPA's). But when three or more operands are to be added in a single cycle using two-operand adders, the time consuming carry-propagation must be repeated several times. If the number of operands is k, then carries have to propagate (k-1) times .

So, a better option is to use that first reduces the three numbers to two and then any CPA adds the two numbers to compute the final sum.

From the timing and area perspective, the CSA is one of the most efficiently and widely used techniques for speeding up digital designs of signal processing systems dealing with multiple operands for addition and multiplication[2].

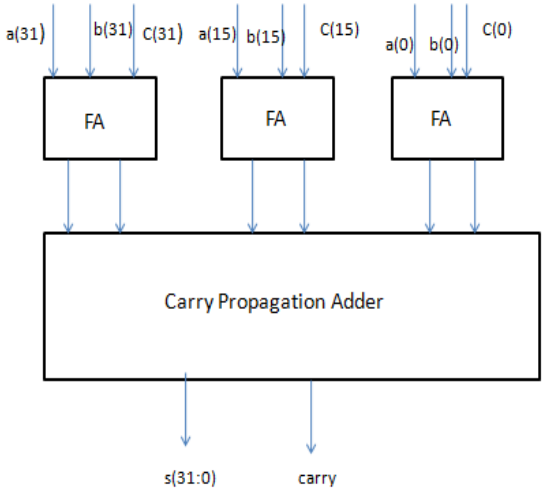

Fig: 32-bit Carry Save Adder

In carry save addition, carry is propagated in last step, while in all the other steps a partial sum and a sequence of carries is computed separately. Thus, the basic CSA accepts three n-bit operands and generates two n-bit results, an n-bit partial sum and an n-bit carry. A second CSA accepts these two bit sequences and another input operand, and generates a new partial sum and carry. A CSA is therefore, capable of reducing the number of operands to be added from 3:2, so it is also called 3:2 compressors.

3. EXISTING SYSTEM

Array Multiplier

Array multiplier is an efficient layout of a combinational multiplier. Multiplication of two binary number can be obtained with one micro-operation by using a combinational circuit that forms the product bit all at once thus making it a fast way of multiplying two numbers since only delay is the time for the signals to propagate through the gates that forms the multiplication array.

In array multiplier, consider two binary numbers A and B, of m and n bits. There are mn summands that are produced in parallel by a set of mn AND gates. n x n multiplier requires n (n-2) full adders, n half-adders and n^2 AND gates. Also, in array multiplier worst case delay would be (2n+1) td.

Array Multiplier gives more power consumption as well as optimum number of components required, but delay for this multiplier is larger. It also requires larger number of gates because of which area is also increased; due to this array multiplier is less economical .Thus, it is a fast multiplier but hardware complexity is high.

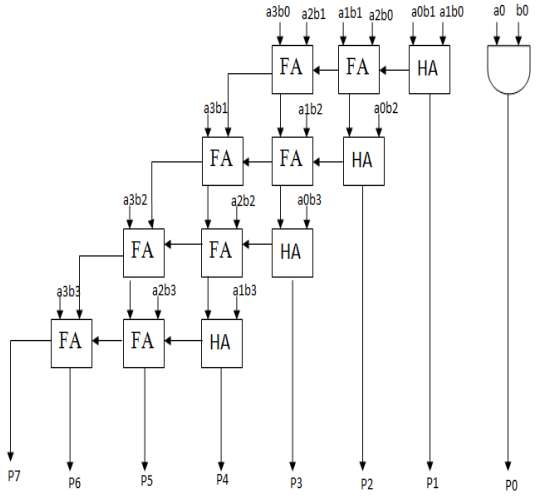

Fig: Conventional Array Multiplier

4. PROPOSED SYSTEM

Carry Save Multiplier

In the Carry Save Addition method ,the first row will be either Half Adders or Full Adders .If the first row of the partial products is implemented with Full Adders, Cin will be considered 0's. Then the carries of each full adders can be diagonally forwarded to the next row of the adder. The resulting multiplier is said to be carry save multiplier, because the carry bits or not immediately added, but rather are saved for the next stage. In the design if the full adder have two input data the third input is considered as 0. In the final stage ,carries and sums are merged in a fast carry propagates. This is the Conventional multipliers with CSA as shown in fig.

In the proposed method ,we implement all the partial product rows of the multiplier as

same as that of the conventional array. The final adder which is used to add carries and sums of the multiplier is removed in this method. Then the carries of the multiplier at the final stage is carefully added to the inputs of the multiplier as shown in fig. The carry of the fourth column of the multiplier is given to the input of the fifth column instead of zero. Then the carry of the fifth column is forwarded to the input of the sixth column so on. In this multiplier the carry of the the seventh column of the adder is not neglected, it is considered as MSB of the multiplier. Due to the elimination of four Full Adders in the final stage power and area can be trade off in the proposed design than that of the conventional array multiplier.

.

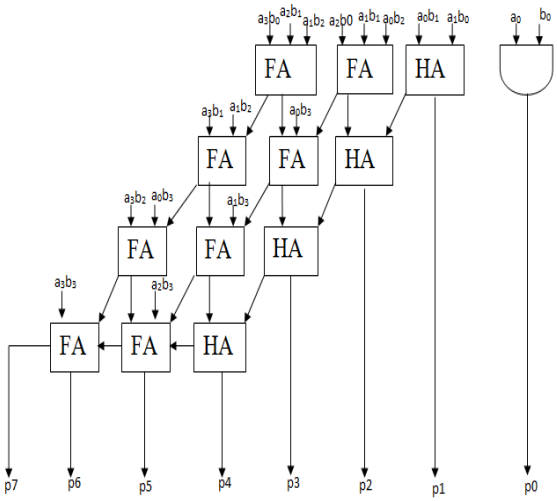

Fig: Proposed Array Multiplier with CSA

SIMULATION RESULT