Comparison of Efficient and High Speed Adders for Vedic Multipliers: A Review
Shiksha Pandey¹ Deepak kumar²

M.Tech. Scholar, EC Department, Vidyapeeth Institute of Science & Technology Bhopal, M.P., India
Assistant Professor, EC Department, Vidhyapeeth Institute of Science & Technology, Bhopal, M.P., India

Abstract: In the VLSI system design, the main areas of research are the reduced size & high speed path logic systems. A fundamental requirement of high speed addition and multiplication is always needed for the high performance processors. In the digital system, the speed of addition depend on the propagation of carry which is generated sequentially after the previous bit has been summed & carry is propagated into the next position. There are many types of adders available such as Ripple Carry Adder, Carry Look Ahead Adder, Carry Save Adder, Carry Skip Adder and Carry Select Adder, which have their own advantages and disadvantages. With the advances in technology, design of Carry select adder (CSA) which offers either of the high speed, low power consumption, regularity of layout less area and compact VLSI design implementation. Researchers justify that Ripple Carry Adder had a smaller area but having lower in speed, in comparing with Carry Select Adders are fastest speed but posses a larger area. And a Carry Look Ahead Adder is in between the spectrum having proper tradeoffs between time and area complexities.

Keywords: RCA, CLA, CSA.

I. INTRODUCTION
Addition is the basic fundamental operation of any arithmetic function, which is required in any digital systems, digital processing systems or control system where multiplication, subtraction or division programming is done; the adder circuit plays a very important role in the calculations.

The most basic arithmetic operation is the addition of two binary digits 0 or 1 and they are known as bits. A combinational digital circuit which adds two bits i.e. 0 or 1 is known as half adder. A full adder is one that adds three bits, that consist of two bit and one carry bit which is in fact produced from the addition of previous two bits. Full adder circuit is easily implemented by combining two half adder circuits.

This paper is organized into three sections. Firstly a brief study of different types of fast digital adders with their architecture and working will be discussed in Section II. 

In Section III, we will be discussing about the hardware requirements in terms of LUT memories and cells, speed of switching, advantages and disadvantages by comparing the results obtained by exhaustive simulations of Ripple Carry Adder (RCA), Carry Look Ahead Adder (CLA) and Carry Select Adder (CSA) [1,2].

Conclusions drawn after comparing results and future work proposed will be discussed in the section IV.

II. ARCHITECTURE OF ADDERS

A. Full Adders

Addition is a fundamental operation for any digital system, digital signal processing or control system. A fast and accurate operation of a digital system is greatly influenced by the performance of the resident adders. Adders are also very important component in digital systems because of their extensive use in other basic digital operations such as subtraction, multiplication and division. Hence, improving performance of the digital adder would greatly advance the execution of binary operations inside a circuit compromised of such blocks. Basic full adder architecture is depicted in fig.1a and input-output relation is shown in fig.1b. Different techniques have been developed for addition optimizing the performance in terms of speed, power and area [3]. In this section we will review the architecture of RCA, CLA and CSA at gate level [4].

Fig.1a Gate level Architecture of 1 Bit Full Adder
TABLE 1-
Truth Table for Full Adder Circuit

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

B. Ripple Carry Adder

The ripple carry adder is constructed by cascading full adders (FA) blocks in series. One full adder (FA) adds two binary digits at any stage of ripple carry adder and the carry output of one stage is directly fed to the next stage of the adder circuit.

As number of full adders may be increased or decreased according to the requirement of the design. In case of the increasing the stages of Ripple Carry Adder (RCA), number of full adders (FA) cascaded in series depending on the bit requirement. They are different sizes also. Like, for designing of n-bit Ripple Carry Adder (RCA) it requires N number of Full Adders (FA). Fig. 1c shows an example of a parallel adder: a 4-bit ripple-carry adder includes four full adder circuits. The augends bits of x are added to the addend bits of y respectfully of their binary position. Each bit addition creates a sum and a carry out. The carry out is then transmitted to the carry in of the next higher-order bit. The final result creates a sum of four bits plus a carry out (c4).

Though it is a simple adder circuit and can be used to add unhindered bit length numbers, but not very efficient when large numbers of bits are used. One of the most serious disadvantage of this adder circuit is that the increase in delays which depend on the bit length. In the working of Ripple Carry Adder (RCA) every stage of Full Adder depends on the carry of the previous stage. Taking again the example in figure 4, in the 4th stage of the 16-bit Ripple Carry Adder (RCA), addition of x4 and y4 will not reach their final condition until the carry out c4 is generated. For this final carry out c4 to come depend on carry c3, which carry out of the previous stage i.e. from the 3rd stage of the Ripple Carry Adder (RCA). Similarly, it goes on from higher to lower stage i.e. carry of current stage to come depend on the carry of previous stage such as c4⇒c3⇒c2⇒c1. Therefore, suppose if one full adder takes T seconds to complete its adding process, so for 4 full adders it requires 4T seconds to accomplish this operation. If the area of RCA is denoted by A. Then there will be very small change in the area or size of the circuit. If it is priory known that initially carry bit is zero, at that time we can simply replace full adder (FA) with a half adder circuit and that will known as first stage of the circuit. Let the delay count is given by Tgate counted each single gate area, by Agate then by evaluating the basic circuit, we found that delay becomes 3n Tgate and area 5n Tgate, n is the number of bit size [5,6].

The worst-case delay of the RCA is when a carry signal transition ripples through all stages ofadder chain from the least significant bit to the most significant bit, which is approximated by:

\[ t = (n - 1) t_c + t_e \]

where, 

- \( t_c \) is the delay through the carry stage of a full adder,
- \( t_e \) is the delay to compute the sum of the last stage.

The delay of ripple carry adder is linearly proportional to n, the number of bits; therefore the performance of the RCA is limited when n grows bigger.

C. Carry Look Ahead Adder

As seen in the ripple-carry adder, its limiting factor is the time it takes to propagate the carry. The carry look-ahead adder solves this problem by calculating the carry signals in advance, based on the input signals. The result is a reduced carry propagation time. To be able to understand how the carry look-ahead adder works, we have to manipulate the Boolean expression dealing with the full adder. The Propagate P and generate G in a full-adder, is given as:

\[ P_i = A_i \oplus B_i \text{ Carry Propagate} \]
\[ G_i = A_iB_i \text{ Carry Generate} \]

Note that both propagate and generate signals depend only on the input bits and thus will be valid after one gate delay.

The new expressions for the output sum and the carryout are given by:

\[ S_i = P_i \oplus C_{i-1} \]
\[ C_i = G_i + P_iG_i \]

These equations show that a carry signal will be generated in two cases:

1) if both bits \( A_i \) and \( B_i \) are 1
2) if either \( A_i \) or \( B_i \) is 1 and the carry-in \( C_i \) is 1.

Let's apply these equations for a 4-bit adder:

1. \( C_1 = G_0 \oplus P_0C_0 \)
2. \( C_2 = G_1 + P_1C_1 = G_1 + P_1 (G_0 + P_0C_0) = G_1 + P_1G_0 + P_1P_0C_0 \)
3. \[ C_3 = G_2 + P_2C_2 = G_2 + P_2G_1 + P_2P_1G_0 + P_2P_1P_0C_0 \]

4. \[ C_4 = G_3 + P_3C_3 = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0 + P_3P_2P_1P_0C_0 \]

These expressions show that \( C_2, C_3 \) and \( C_4 \) do not depend on its previous carry-in. Therefore \( C_4 \) does not need to wait for \( C_3 \) to propagate. As soon as \( C_0 \) is computed, \( C_4 \) can reach steady state. The same is also true for \( C_2 \) and \( C_3 \).

The general expression is

\[ C_{i+1} = G_i + P_iG_{i-1} + P_iP_{i-1}G_{i-2} + \ldots + P_iP_{i-1} \ldots P_2P_1G_0 + P_iP_{i-1} \ldots P_1P_0C_0 \]

This is a two level Circuit. In CMOS however the delay of the function is nonlinearly dependent on its fan-in. Therefore large fan-in gates are not practical.

Carry look-ahead adder’s structure can be divided into three parts: the propagate/generate generator shown in Fig.2a, the sum generator Fig.2b and the carry generator Fig.2c.

![Fig. 2a Propagate / Generate Generator](image)

![Fig. 2b Sum Generator](image)

![Fig. 2c Look Ahead Carry Generator](image)

The size and fan-in of the gates needed to implement the Carry-Look-ahead adder is usually limited to four, so 4-bit Carry-Look ahead adder is designed as a block.

**D. Carry Select Adder**

Carry-select adders can be divided into equal or unequal sections. Fig.3a shows the implementation of an 8 bits carry-select adder with 4-bit sections. For each section, shown in Fig.3a, the calculation of two sums is accomplished using two 4-bit ripple-carry adders. One of these adders is fed with a 0 as carry-in whereas the other is fed a 1. Then using a multiplexer, depending on the real carryout of the previous section, the correct sum is chosen. Similarly, the carryout of the section is computed twice and chosen depending of the carryout of the previous section. The concept can be expanded to any length for example a 16-bits carry-select adder can be composed of four sections each section is shown in Fig. 3b. Each of these sections is composed of two 4-bits ripple-carry adders. This is referred as linear expansion. The delay of \( n \)-bit carry select adder based on an \( m \)-bit CLA blocks can be given by the following equation when using constant carry number blocks

\[ T = t_{\text{setup}} + mt_{\text{carry}} \frac{n}{m} t_{\text{mux}} + t_{\text{sum}} \]

And by the following equation when using successively incremented carry number blocks respectively.

\[ T = t_{\text{setup}} + mt_{\text{carry}} + (2n)^{1/2} t_{\text{mux}} + t_{\text{sum}} \]
III. SIMULATIONS AND RESULTS

The implementation of the different is done in the Verilog module using Xilinx 10.2i software platform and verify the simulation results.

Simulation Results

Simulation results of 16 bit Ripple Carry Adder is shown in Fig. 4a, 16 bit Carry Look Ahead Adder is shown in Fig.4b and 16 bit Carry select adder in Fig. 4c.

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>1</td>
</tr>
<tr>
<td>b</td>
<td>2</td>
</tr>
<tr>
<td>sum</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 1

<table>
<thead>
<tr>
<th>Logic Utilization</th>
<th>RCA</th>
<th>CLA</th>
<th>CSA</th>
</tr>
</thead>
<tbody>
<tr>
<td>No.of Slices</td>
<td>96</td>
<td>72</td>
<td>108</td>
</tr>
<tr>
<td>Number of 4 input LUTs</td>
<td>128</td>
<td>128</td>
<td>192</td>
</tr>
<tr>
<td>Number of Bonded IOBs</td>
<td>200</td>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td>Delay</td>
<td>96.686ns</td>
<td>96.686ns</td>
<td>88.092ns</td>
</tr>
</tbody>
</table>

IV. CONCLUSION AND FUTURE SCOPE

1. Conclusion

In this survey paper we have simulated parallel adders using Xilinx 10.2i version and 3s100evq100-4 is the targeted device. In terms of area Carry Look Ahead Adder is a better choice than Carry Select and Ripple Carry Adders. But in terms of gate delay count Carry Select Adder offers higher speed than the other two. For optimization between area and speed CSA has an edge of 5% over CLA and RCA.

2. Future Scope

In future one can obtained results for 32 bits or higher number of bits so that the convergence of results can accurately predicted. The CSA can be replaced in Vedic Multipliers and efficiency in terms of area and speed can be measured.

REFERENCES


[8] www.altera.com