HIGH PERFORMANCE PARALLEL PREFIX ADDERS WITH FAST CARRY CHAIN LOGIC

Anitha R\textsuperscript{1}, V Bagyaveereswaran\textsuperscript{2}

\textsuperscript{1}eranitharavi@gmail.com, \textsuperscript{2}bagyaveereswaran@gmail.com

VIT University, Vellore – 632014, TamilNadu, India.

ABSTRACT

Binary adders are the basic and vital element in the circuit designs. Prefix adders are the most efficient binary adders for ASIC implementation. But these advantages are not suitable for FPGA implementation because of CLBs and routing constraints on FPGA. This paper presents different types of parallel prefix adders and compares them with the Simple Adder. The adders are designed using Verilog HDL code and simulated and synthesized using Xilinx ISE13.2 software tool and Cadence RTL compiler. Among all the adders, Kogge-Stone adder provides better performance in ASIC implementation but it is not suitable for FPGA implementation. In order to make it suitable for FPGA implementation, Kogge-Stone adder is modified using fast carry logic technique. The modified adder provides better performance over the Simple adder for the higher order bit widths.

Keywords- FPGA, Binary addition, Carry tree adders, Prefix computation, Prefix addition.

1. INTRODUCTION

Adders are the most important element in all the digital circuit design. Among the various types of adders, Carry Tree Adder which is also known as Parallel Prefix Adders provide high performance in terms of speed in ASIC implementation. The Carry Tree adder performs three stage operations such as Pre-computation stage, Prefix-computation stage and Post-computation stage. The Pre-computation stage generates and propagates the carry signal, the Prefix computation stage generates the carry signal using prefix cells and the Post-computation stage generates the sum. This is shown in Fig.1 and the equation for these three stage addition operation is given in equation 1, 2 &3.

Field Programmable Gate Array [10] offers low cost and less development time over ASIC implementation. In this paper, the most efficient adder i.e., the carry tree adder like Kogge-Stone, Brent-Kung, Ladner-Fischer and Han-Carlson adders are designed and implemented on FPGA.
The problems involved in FPGA implementation are investigated and the possible FPGA architecture which can make the Carry Tree Adder to provide high performance over the Simple adder is explored. The possible trade-offs like area, power, delay, interconnect count and fan-out involved in the adders are examined.

![Fig.1. Block Diagram of Prefix addition](image)

The three stages addition consists of the following computations:

- **Pre-computation:** \( G_{m,n} = A_n \) and \( B_n \), \( G_0 = c_{in} \); \( P_{m,n} = A_n \oplus B_n \), \( P_0 = 0 \);
  \[ \text{......(1)} \]

- **Prefix-computation:** \((G_m, P_m) \circ (G_n, P_n) = (G_{n:k} + P_{n:k} \cdot G_{k-1:n}, P_{n:k} \cdot P_{k-1:j})\)
  \[ \text{(or) } G_{m:n} = G_{n:k} + P_{n:k} \cdot G_{k-1:n} \]
  \[ P_{m:n} = P_{n:k} \cdot P_{k-1:j} \]  \[ \text{......(2)} \]

- **Post-computation:** \( S_n = P_n \oplus G_{n-1:0} \)
  \[ \text{......(3)} \]

### II. CARRY TREE ADDERS

The various types of carry tree adders are shown in Fig.2. Each carry tree adder consists of three parts. They are: Upper part, Middle part, Lower part. Using these parts the carry tree adders computes \( N \) outputs from \( N \) inputs as shown in Fig.1. The Upper part generates and propagates the carry signal from the input to the prefix stage using the formula given in equation (1). The propagated and generated carry signals are combined using the associate operator “\( \circ \)”. This operation is performed in the middle part using the formula given in equation (2). The Middle part consists of prefix cells such as black cells, grey cells and white buffers [1]. The arrangement of these prefix cells in different order results in various types of Carry Tree adders, where the carry signals need not to be propagated. Such operations are performed by grey cells. The grey cells generate the carry signal only. Black cell generates and propagates the carry signal. There are some places the white buffers are used to reduce the loading effect for the further stages. The Lower part generates the overall sum using the formula given in equation (3).

Depends on the arrangement of prefix cells, the carry tree adders involves in trade-offs like area, power, delay, interconnect count, fan-out and logic depth [3 & 4]. Fig.2 (a) shows the Brent-Kung The dark black line in the figure indicates the critical path of the adder. The critical path for Han-Carlson and Kogge-Stone are less. So these two
adders are expected to be the fastest adder. The power utilized by all the Carry Tree Adder is more than the Simple Adder.

![Brent Kung Diagram](image1)

(a) Brent Kung

![Kogge Stone Diagram](image2)

(b) Kogge Stone

minimum area and maximum logic depth. Due to the maximum logic depth, the delay of this adder is expected to be high. Fig.2 (b) shows the Kogge-Stone adder. It is designed in such a way that it provides maximum interconnect count and area but minimum logic depth and fan-out. Ladner-Fischer adder as shown in Fig.2 (c) provides minimum logic depth with improved area. Han-Carlson adder as shown in Fig. 2(d) provides minimum logic depth and minimum interconnect count.
Simple adder is designed using Verilog HDL ‘+’ operator. The carry chain structure on FPGA makes Simple Adder to provide high performance. But this is not an efficient adder for VLSI implementation. In this paper, Carry Tree Adder is compared with Simple Adder for both ASIC and FPGA implementation.

**III. RELATED RESEARCH AND PROPOSED WORK**

The different types of carry tree adders are discussed in [4]. In [5], the authors implemented different types of adders like Simple Adder, Carry Look Ahead Adder, Carry Skip Adder, and Carry Select Adder on the Virtex2 FPGAs and found that the Simple Adder provides better performance. In [3], the authors discussed various parallel prefix networks design and implementation on a Xilinx Virtex5 FPGA. It is observed that the Simple Adder provides better performance over the prefix networks for the bit widths up to 256 bits. This is due to the advantage of the carry chain structure on the FPGA. All these works by different authors shows that the simple adder provides better performance on FPGA. The area, delay results for these works depend upon synthesis reports. In [2], the authors described several Carry Tree Adders implemented on a Xilinx Spartan3E FPGA. It is found that the Kogge Stone Carry Tree Adder provide better delay performance for the higher order bits. The results obtained for this paper is similar to those presented in [2].
Carry Tree Adders are designed, coded, simulated and synthesized and then it is compared with the Simple Adder. The obtained area, power, delay results of various Carry Tree Adders are compared with each other and also with the Simple Adder. Among all the Carry Tree Adders, Kogge-Stone Adder and Han-Carlson Adder is expected to be the fastest adder in ASIC implementation but not in FPGA implementation.

In this paper, Kogge-Stone Adder is taken, since it is having minimum fan-out and logic depth than Han-Carlson Adder, and modified using Fast Carry Logic technique in order to make it suitable for FPGA implementation [6, 7, 8 & 9]. The addition operation performed by Simple Adder, which is generated by synthesis tool, is shown in Fig.3 (a). From Fig.3 (a), it is clear that the Prefix-computation stage of the Simple Adder uses multiplexers. Similarly, the Prefix-computation stage of Carry Tree Adder is replaced with the Fast Carry logic technique which uses muxes as shown in Fig.3 (b). The Fast Carry Logic architecture for 4-bit addition is shown in Fig.3(c). Instead of using Black cells, Grey cells and White buffers to propagate and generate the carry signals, simple muxes are used. The blocks present in Fast Carry Logic technique also uses muxes. The input to the Fast Carry Logic is the propagated and generated carry signal of the Pre-computation stage. The Pre-computation and Post-computation of the modified adder is similar to that of the normal carry tree adders.

IV. RESULTS

The delay, power and cell area results obtained by synthesizing the designed adders for 128bits using Cadence RTL compiler (90nm technology) is shown in Table 1, 2 & 3. The abbreviations used in the table are: KS for the Kogge Stone Adder, BK for the Brent Kung Adder, LF for the Ladner Fischer Adder and HC for Han Carlson Adder.
The delay is measured in terms of nanoseconds, power in terms of nanowatt. From the results it is found that the Carry Tree Adders provide best delay performance than the Simple adder. Among the Carry Tree Adders, Kogge-Stone Adder and Han-Carlson Adder provide best delay as it is expected but the area and power utilized by those adders are more. Comparatively, Brent-Kung Adder and Ladner-Fischer Adder utilizes less area and power.
Table 1: Delay Results of Carry Tree Adders compared with Simple Adder

<table>
<thead>
<tr>
<th>N</th>
<th>SIMPLE ADDER</th>
<th>CARRY TREE ADDERS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>KS</td>
</tr>
<tr>
<td>8</td>
<td>768</td>
<td>307</td>
</tr>
<tr>
<td>16</td>
<td>1605</td>
<td>599</td>
</tr>
<tr>
<td>32</td>
<td>3420</td>
<td>817</td>
</tr>
<tr>
<td>64</td>
<td>6834</td>
<td>886</td>
</tr>
<tr>
<td>128</td>
<td>8900</td>
<td>915</td>
</tr>
</tbody>
</table>

Table 2: Power Results of Carry Tree Adders compared with Simple Adder

<table>
<thead>
<tr>
<th>N</th>
<th>SIMPLE ADDER</th>
<th>CARRY TREE ADDERS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>KS</td>
</tr>
<tr>
<td>8</td>
<td>0.6464743</td>
<td>2.065801</td>
</tr>
<tr>
<td>16</td>
<td>2.065857</td>
<td>4636.68</td>
</tr>
<tr>
<td>32</td>
<td>4.062593</td>
<td>11383.85</td>
</tr>
<tr>
<td>64</td>
<td>8.056257</td>
<td>24264.12</td>
</tr>
<tr>
<td>128</td>
<td>16.1108015</td>
<td>686902.3</td>
</tr>
</tbody>
</table>

Table 3: Cell Area Results of Carry Tree Adders compared with Simple Adder

<table>
<thead>
<tr>
<th>N</th>
<th>SIMPLE ADDER</th>
<th>CARRY TREE ADDERS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>KS</td>
</tr>
<tr>
<td>8</td>
<td>119</td>
<td>291</td>
</tr>
<tr>
<td>16</td>
<td>254</td>
<td>675</td>
</tr>
<tr>
<td>32</td>
<td>509</td>
<td>1575</td>
</tr>
<tr>
<td>64</td>
<td>1029</td>
<td>3672</td>
</tr>
<tr>
<td>128</td>
<td>2088</td>
<td>8385</td>
</tr>
</tbody>
</table>

Fig. 4 shows the simulated delay results of the adders for the bit widths up to 128 bits using Xilinx ISE13.2 software tool. From the Fig. 4, it is found that the Simple Adder provides the best delay performance over the Carry Tree Adder. The obtained delay result is entirely different from the result shown in Table 1. This is because of the presence of Fast Carry chain structure on Xilinx FPGA. Among the Carry Tree adders, Kogge-Stone Adder provides the best delay as it is expected.
Fig. 5(a-d) shows the delay results of Kogge-Stone Adder, Kogge-Stone Modified Adder and Simple Adder for the FPGA families like Spartan-3E, Virtex-4, Virtex-5 and Virtex-6. Lower power. Some of the 64-bit adder structure cannot be fitted into all the devices under this family.

![Delay Results - Spartan3E](image)

Fig. 4 Simulated Delay Results of Carry Tree Adders compared with Simple Adder

Depending on the adder structure, the device and package has been selected. From the Fig it is found that, for Spartan-3E FPGA, Kogge-Stone adder provides best performance after it reaches 256 bits whereas Modified adder provides best performance after it reaches 128 bits, for Virtex-4 FPGA, Kogge-Stone adder provides best performance after it reaches 128 bits whereas Modified adder provides best performance from 128 bits, for Virtex-5 FPGA, Kogge-Stone adder provides best performance after it reaches 256 bits whereas Modified adder provides best performance from 128 bits, for Virtex-6 FPGA, it is able to reduce the delay of Carry Tree Adder but Simple Adder provides better delay performance.

![Delay Results - Spartan3E](image)

(fig. 5.a)
v. CONCLUSION

This paper presents different types of Carry Tree Adders. Kogge Stone adder is the fastest carry tree adder in VLSI implementation but it provides different result for the FPGA implementation.

(Fig. 5.b)

In order to make it suitable for FPGA implementation, the prefix computation stage is modified using “Fast Carry Logic”.

(Fig. 5.c)
The obtained delay results are compared with the Simple Adder for the various FPGA devices like Spartan3E, Virtex4, Virtex5, Lower power Virtex6. The Lower power Virtex6 FPGA provides best delay compared to that of all the FPGA devices. By using carry logic technique the Carry Tree Adders are able to provide better delay performance on FPGA over the Simple Adder for the higher order bit widths.

REFERENCES

10. www.xilinx.com