DESIGN AND IMPLEMENTATION OF LOW POWER HIGH SPEED 64-BIT HCSKA

C.Sirisha, PG student, Dept. of ECE, GPREC, Kurnool. challasirishareddy@gmail.com
M.Madhusudan Reddy, Assistant Professor, Dept. of ECE, GPREC, Kurnool. msreddym11@gmail.com

Abstract—The main objective of this paper is to design a 64 bit carry skip adder (CSKA) which is having higher speed and low power consumption when compare to conventional carry skip adder (CSKA). In the conventional model 2X1 multiplexers is used for carry skip logic purpose. To achieve speed enhancement in 64- bit (CSKA) compound gates are used. By replacing the compound gates instead of multiplexers we can increase the speed and decrease the power for 64- bit CSKA compared to conventional procedure. The structure may be realized with fixed & hybrid carry skip adder. The results are obtained using CADENCE TOOL and power, area and delay was decreased.

Index Terms—Carry skip adder (CSKA), Compound gates, hybrid carry skip adder.

I. INTRODUCTION

The addition operation is one of the most and fundamental arithmetic operations used in digital signal processing [1]. Several implementations relate to adder are carry skip adder, look-ahead carry, carry skip and various parallel prefix adders are available to satisfy different area, delay and power requirements[2]. An efficient adder design essentially improves the performance of a complex digital signal processing system.

In existing method, it will use multiplexer logic for that it takes more number of gates, power consumption by the carry skip adder area is more and delay is high. The construction of ripple carry adder uses cascading of full adders blocks in series. Ripple carry adder structure is simple but critical path delay is more. In carry selection adder, Speed, power and usage of area are considerably larger than Ripple carry adder (RCA).

The main contribution of the paper is summarized as follows.

1) In the proposed method 64 bit CSKA structure by adding the compound gates Schemes to conventional one structure for improving the speed and decreasing the area of the adder. The modification provides the AOI/OAI logical gates instead of the multiplexer logic.

2) Constructing an improved 64 bit CSKA structure for better results.

3) The hybrid Carry Skip Adder is based on the extension of the proposed CSKA which was replaced in the middle stages by using parallel prefix network structure.

II. EXISTING SYSTEM

I. Conventional Carry Skip Adder (CSKA)

In CSKA conventional structure it contains cascading of full adders, ripple carry adder and 2X1 multiplexer. The ripple carry adder blocks are connected each other by using 2X1 multiplexer which are placed at more level structures [3]. Some methods have been suggested for finding the optimum number of Full adders [4-11].

Alioto and Palumbo proposed a simple conventional structure of CSKA based on VSS technique where we can determine near optimal numbers of the full adders based on the delay of the multiplexer and time required to carry to ripple through a full adder.

Fig 1: Conventional Carry Skip Adder

Fig 2: 8-Bit Conventional CSKA
Fixed Stage Carry Skip Adder

In fig.1 shows the N-bit conventional carry skip adder structure, which is based on ripple carry adder blocks. Ripple carry adder contains cascaded full adders in each stage, the propagation delay of two N-bit numbers A&B full adders belong to propagation mode.

\[ P_i = A_i \oplus B_i = 1 \quad \text{for} \quad i = 1, \ldots, N \]

Where

\( P_i \) is the propagation signal

For N full adders of the carry skip adders are grouped in Q stages. Each ripple carry adder stage contains Mj full adders and skip logic. The inputs of skip logic are the carry input of stage and the carry output of its RCA block.

\[ T_D = [M \times T_{\text{CARRY}}] + [(N/M - 1) \times T_{\text{MUX}}] + [(M - 1) \times T_{\text{CARRY}} + T_{\text{SUM}}] \]

\( T_D \) = Critical path Delay

\( T_{\text{CARRY}} \) = propagation delays of the carry output of an full adder

\( T_{\text{MUX}} \) = Output delay of multiplexer

\( T_{\text{SUM}} \) = propagation delay sum output of an full adder

III. PROPOSED SYSTEM

In proposed method concatenation and incrementation methods are used to reduce the power & area and to increase the speed. Instead of 2X1 multiplexer we use AOI & OAI compound gates are used because it consists of minimum number of transistors compare to 2X1 multiplexer. The gates which consists of fewer transistors have low delay, power when compare with 2X1 multiplexer. When compared to conventional one it has lower propagation delay with a slightly smaller area. Compound gates is defined as the sum of the individual gates to design AOI and OAI gates . These are very simple and more efficient compare to individual gates. In the proposed method I have designed 2 types of carry skip adders

1. Concatenation and Incrementation carry skip adder(CI-CSKA)
2. Hybrid carry skip adder( H-CSKA)

Concatenation and Incrementation carry skip adder (CI-CSKA)

The use of 2X1 multiplexers leads to increase in the area usage and delay when compared to the AOI & OAI compound gates. [12][13]. Concatenation means a series of the ripple carry adders where as incrementation block uses generated intermediate results by the ripple carry adder block and the carry output of the previous stage. The carry input for all blocks is zero except for the first ripple carry adder block and first adder in the ripple carry adder chain is a half adder.

In Figure 3 Shows the adder contains 4 stages each stage with 16 bits inputs, A and B. Each ripple carry adder block which is having the size of Mj. In the first stage carry input for C, one and for the remaining ripple carry adder blocks are zero are calculated in parallel which is known as concatenation.

\[ T_{\text{D}} = [M_j T_{\text{CARRY}}] + [(Q-2)T_{\text{SKIP}}] + [(M_j - 1)T_{\text{AND}} + T_{\text{XOR}}] \]

\( T_{\text{D}} \) = Critical path Delay

\( T_{\text{CARRY}} \) = propagation delays of the carry output of an full adder

\( T_{\text{AND}} \) = Delay of the AND gate

\( T_{\text{XOR}} \) = Delay of the XOR gate

[\((M_j - 1)\)T_{\text{AND}} + T_{\text{XOR}}] shows the critical path delay of the jth incrementation block.

\( T_{\text{SKIP}} = (T_{\text{AOI}} + T_{\text{OAI}}) / 2 \)
Hybrid carry skip adder

Hybrid carry skip adder is based on the concatenation and incrementation carry skip adder. In this structure one of the middle stage is replace with the parallel prefix adder. Because it is one of the fastest adder but area and power consumption is more compare to others. So only one of the middle stage is replace with PPA (Parallel prefix adder) in concatenation and incrementation carry skip adder. The Hybrid carry skip adder structure is shown in fig 5.

In PPA the addition operation in three different stages. They are:-
1. First stage is the Pre-processing stage where we obtain the Group Generate and Group Propagate signals.
2. Second stage is the Carry generation stage where we generate the carry using the Group Generate and Group Propagate signals.
3. Third and Final stage is where we obtain the Sum bit using the Carry bit and the Propagate signal.

Brent Kung adder is one of the type of PPA. The Brent Kung adder computes the prefixes for 2 bit groups. These prefixes are used to find the prefixes for the 4 bit groups, which in turn are used to compute the prefixes for 8-bit groups and so on. These prefixes are then used to compute the carry out of the particular bit stage. These carries will be used along with the Group Propagate of the next stage to compute the Sum bit of that stage [14].

IV. SIMULATION RESULTS

In this paper the design has been developed using cadence tool. Conventional, concatenation and incrementation and hybrid carry skip adders are designed for 64 bit. The Simulations were performed using Encounter(R) RTL Compiler RC14.25 - v14.20-s046_1 cadence tool. In the simulation results shows the low power and area, delay are decreased when compared to the conventional one.

Fig 7: Conventional 64-bit carry skip adder

Fig 8: Conventional 64-bit Carry Skip adder RTL Schematic Structure
Fig 9: Floorplanning and placement of Conventional CSKA

Fig 10: Routing of Conventional 64-bit Carry Skip Adder

Fig 11: 64-bit CI-CSKA simulation result

Fig 12: 64-bit CI-CSKA RTL Schematic Structure
Fig 17: Floorplaning and placement of Hybrid CSKA

Fig 18: Routing of 64-bit Hybrid Carry Skip Adder

Table 1: Comparison Table

<table>
<thead>
<tr>
<th>Adder Techniques</th>
<th>Delay</th>
<th>Area</th>
<th>Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conventional CSKA</td>
<td>0.24 ns</td>
<td>2510 µm</td>
<td>162459.361 Nw</td>
</tr>
<tr>
<td>CI-CSKA</td>
<td>0.21 ns</td>
<td>2464 µm</td>
<td>161452.762 Nw</td>
</tr>
<tr>
<td>Hybrid CSKA</td>
<td>0.19 ns</td>
<td>1938 µm</td>
<td>87882.944 Nw</td>
</tr>
</tbody>
</table>

V. CONCLUSION

In this paper the speed enhancement was achieved by AOI and OAI compound gates are used instead of 2X1 multiplexer. In addition compound gates are used for the skip logic purpose. A 64-bit CSKA structure called CI-CSKA was proposed which results in high speed and low power consumption compared to conventional carry skip adder. In the middle stage of CI-CSKA one stage will be replaced by Brent Kung adder which produces hybrid carry skip adder. The suggested hybrid carry skip adder will increase the speed.

REFERENCES

[14] Design and Implementation of a delay and area efficient 32x32bit Vedic Multiplier using Brent Kung Adder #1Ayushi Sharma, #2Er. Ajit Singh