Optimal Performance Analysis of Carry Skip Adder on FPGA

K. Murali Chandra Babu¹, Dr. P.A.Harsha Vardhini²

¹Department of ECE, Vignan Institute of Technology and Science, Deshmukhi, Telangana, India
muralichandra454@gmail.com, pahv19@rediffmail.com

Received # Mon.20##, Revised # Mon. 20##, Accepted # Mon. 20##, Published # Mon. 20##

Abstract: Demand for Portability and performance in VLSI circuits is a preferred requisite in designing low power and high speed circuits. The accuracy and processor speed of VSLI circuits depends on adder module in various applications like digital signal processing, microprocessors, microcontrollers image processing, radar beamforming, etc., This can be achieved by integrating low power schemes at various phases of the design. RCA architecture is known for adding N-bits with fast calculation speed. But RCA architecture full adders are connected in series that introduce long delay due to long carry chain. Look-ahead carry adder improves the delay by using propagate and generate functions. In 32-bit carry skip adder with uniform module size, skip logic developed using group propagate and generate functions. The skip logic circuit of each module is designed using the multiplexer with the selection line AND operation of propagate signals of the module. A novel 32-bit CSK architecture design is proposed by replacing the multiplexer in skip logic with OAI and AOI complementary logic which improves the carry propagation delay. Novel technique is proposed to enhance the operation speed. This novel technique is designed in Verilog HDL and implemented in Xilinx ISE tool and prototyped on to FPGA board.

Keywords: Carry skip adder, Ripple carry adder, Verilog HDL, Xilinx ISE tool, FPGA

1. INTRODUCTION

VLSI circuits with high performance are great challenge for portable devices, mobile phones and most advancing biomedical application can be achieved by the speed of operation, low power schemes, less area occupancy. A Digital circuit contains Arithmetic Logic Unit (ALU) along with many operational blocks. Adder is the major component in ALU. Designing an adder efficiently will improve the performance of the digital circuits. Addition operation is used in most of the applications like encoding, decoding, beam-forming etc.[1,2]. In general addition is adding of two numbers and getting sum and carry as outputs. In digital circuits binary adder will decide the performance of the circuit and is used in many applications like multiplier, dividers, calculating the address in cache memories [3]. This binary adder is designed with advanced techniques and technologies for better functioning of the digital circuit. The operational speed of adder depends on time taken by the input data to travel towards the output that relies on carry chain. As input data is more propagating a carry towards the output will take a long duration which is limiting the performance of the circuit. Research has been continuing by the designers to improve and enhance the performance of the circuits [4,5]. Novel design architectures are innovated to compensate the tradeoff between area & power and to achieve high accuracy and performance.

2. CONVENTIONAL ARCHITECTURE

A. RCA

Half Adder is the basic digital circuit used for addition which is further improved the logic in Full Adder (FA). Carry output which is generated decides the performance of adder. More the bits at the input, greater number of full adder circuits are used where the output carry propagates towards the next full adder. Though computation is fast, delay will add while acquiring the output carry. In figure 1 a full adder circuit is shown the propagation delay can be calculated by its critical path A to C_out.

![Full Adder Diagram](image)

Figure 1. Full Adder

The speed of the full adder can be measured how fast the input bits travel towards the output. This speed is even less if word size is more. Ripple Carry Adder (RCA) is a digital circuit where the number of input bits added in...
parallel. RCA circuit is built by connecting full adder
circuits in series in which the carry-out of one full adder
tie up to carry-in of consecutive full adder. Figure 2
convey the RCA circuit with 4-bit.

\[ P_i = A_iB_i' + A_i'B \]  
\[ G_i = A_iB_i \]  

The execution of RCA depends on the carry propagation
towards the output. More the word-size RCA will take
more time to get the carry output.

B. Look-ahead Carry adder (CLA)

RCA has longer duration due to presence of number of
full adders for addition. Alternatively, a modified RCA
called look-ahead adder (CLA) was designed for
improvement in execution. In CLA two functions
propagate and generate used to reduce the delay.
Carry will propagate when P=1 and generate when G=1 by using these P and G values the carry propagate path in RCA is controlled thus reduces the delay in the circuit. In CLA generation of carry becomes simple with hardware complexity required to generate the output carry is complex because it occupies more area. Longer the input bits more the circuit complex in CLA.

C. Carry Skip Adder (CSK)

Please do not revise any of the current designations.

The CSK minimizes the circuit delay by passing the carry for a group of consecutive adder blocks. In RCA individual full adder will be idle until the carry-out of preceding full adder generated. This long dependency will be eliminated by adding skip logic. The carry propagation, carry-out of the \( i^{th} \) stage FA can be given as

\[
P_i = A_iB_i' + A_i'B_i
\]  
(3)

\[
C_{i+1} = A_iB_i + P_iC_i
\]  
(4)

When both \( A = B = 0 \) then \( P_i = 0, C_{i+1} = 0 \)

When \( A = B = 1 \) then \( P_i = 1, C_{i+1} = 1 \)

From the above two cases the carry depends only on the input bits not on the previous carry. When \( A_i = B_i \), then \( C_{i+1} = C_i \) in this case output carry is same as that of the input carry it will not wait for the generation of the sum output. In this case, it will propagate towards the next block by skipping the adder stages within the block. By this the delay of the circuit will reduce, but with higher hardware complexity.

D. CSK with uniform module size

On the subject of 32-bit CSK the input bits are divide into equal parts of 4 bits each. Each 4-bit adder module is connected as RCA as it best results with less delay. Figure 3 presents 32-bit CSK with uniform module size. The multiplexer at the end of each block is used to bypass the carry without propagating through the block when all carry propagating values are 1. The internal view of block is shown in figure 4. When all the carry propagation values are 1, the last block output sum needs to await till the carry-out of preceding module this gives worst path delay in this adder. This worst path delay will be reduced by replacing the multiplexer by AND and OR gates which is shown in the figure 5.

---

**Figure 4. Internal Structure of CSK block**

**Figure 5. Skip Logic with AND OR gates**
When A and B inputs are not equal then all propagation signals $P_{3:0}$ is 1 and carry of the last full adder $c_4$ is 0 then from the skip logic block input carry $c_0$ is propagated towards the output without passing through the full adder block. Thus, improve the CSK by replacing multiplexer with skip logic.

3. **NOVEL CSK ARCHITECTURE**

Even though CSK with uniform module size works with lesser delay than RCA architecture, later blocks in CSK will produce faster output and it will wait till the carry passes through all skip logic blocks. To overcome this, a new logic is proposed to optimize the adder circuit. The optimization can be done by varying the module size. For speed up circuit the size of the first block with 4-bits and further increase the block size as 6-bits and 18-bits and last block with 4-bits because the critical path includes the last block. Figure 6 depicts block diagram of the design.

![Figure 6. CSK with non uniform module size](image1)

![Figure 7. Internal Structure of first block](image2)
A. First Block (4-bits)

As the carry bits are involved in the critical path, LSB carry generation is preferred rather than the sum bits. This block is connected as a RCA structure to calculate carry very fast. The propagation and generation signals are combined with a small bit to make equal delays in the carry chain. The first bit is added with a standard full adder circuit and remaining 3 bits are added with the P and G signals to generate the carry quickly. The internal logic circuit is shown in figure 7.

The OR-AND-INVERTER (OAI) and AND-OR-INVERTER (AOI) logic is used with the skip the carry without propagation through the adder blocks. The final carry C4 is generated by balancing carry of each adder.

B. Second Block (6-bits)

This block contains 6 bits group to match the delay with preceding carry-out C4 of first block. 3-bits are grouped in this block with P and G signals are used to skip logic. The internal circuit diagram of the second block is depicted in figure 8.

The group propagation and generation are available after the two gate delays and fed to block to generate P and G signals P_{9:4} and G_{9:4} by using these and C4 input carry the output carry C10 is generated after four gates delays.

C. Third block (18-bits)

Carry from the previous block takes 4 gate delays and fed as input to this block, to balance carry chain, block this blocked is grouped with 18 bits which is depicted in figure 9.

As the number of bits increases calculating the sum become difficult. Hence these 18 bits are divided into 3-bit blocks, P and G signals are generated from each block and then forms group propagate and group generation. By this group propagation and generation, the final carry out C28 is generated with 5 complex gate delays.

D. Fourth Block (4-bits)

Block generation of sum becomes a tough task as sum bits are involved in the critical path the logical diagram of this block is represented in figure 10.

The decrease the delay in this block a carry select logic is developed. The propagation and generation signals are combined to generate the final Carry out C32.
Figure 9. Logic diagram of third block

Figure 10. Logic diagram of fourth block
4. **Design Implementation and Results**

The conventional and proposed designs are designed and synthesized in Xilinx ISE tool. The process of converting the programming code into a netlist is called synthesis. In Xilinx ISE Xilinx Synthesis Tool (XST tool) and Xilinx ISE simulator tool are integrated for synthesize and simulate the HDL design for FPGA prototyping. The results from Synthesis tool will be observed in two different ways.

- Technology Independent or Register Transfer Logic (RTL) schematic in which the design is synthesized in the logic gates and flipflops circuits.
- Technology dependent schematic in which the design is synthesized in FPGA blocks like Look-up Tables (LUT) and flipflops.

The 32-bit RCA has the inputs and outputs pins is shown in figure 11.

![Figure 11. Inputs and Outputs of RCA](image1)

The technology independent schematic with logic gates and flipflop is depicted in figure 12. In this schematic the full adder circuit is connected in series and The technology dependent diagram with LUTs and flipflop us shown in figure 13.

![Figure 12. RTL Schematic of RCA](image2)

A 32 bit CSK with uniform module size is designed in Verilog HDL and implemented in Xilinx ISE simulator. Figure 14 and figure 15 depicts RTL and technology schematic.

![Figure 13. Technology schematic of RCA](image3)

A novel 32-bit CSK with non-uniform module size is designed using Verilog HDL and synthesized in the Xilinx ISE tool. The RTL and Technology Schematic is shown in figure 15 (a) and (b).
Conventional and proposed design are implemented on Xilinx Spartan 3 XC3S400PQ208 FPGA board. The device utilization of FPGA for these adders is shown in Table I.

**TABLE I. HARDWARE UTILIZATION OF CONVENTIONAL AND PROPOSED ADDER**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>32-bit RCA</th>
<th>32-bit CSK with fixed size</th>
<th>Proposed design</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of 4 input LUTs</td>
<td>64</td>
<td>104</td>
<td>101</td>
</tr>
<tr>
<td>Number of occupied slices</td>
<td>48</td>
<td>56</td>
<td>57</td>
</tr>
<tr>
<td>Number of bounded IOBs</td>
<td>98</td>
<td>98</td>
<td>98</td>
</tr>
<tr>
<td>Total number of gate count</td>
<td>384</td>
<td>696</td>
<td>630</td>
</tr>
</tbody>
</table>

The total gate count in the proposed design requires less in number compared with the conventional adders. Even though gate count is large for the novel design, the path delay of proposed architecture is much lesser enhancing the operating speed. Figure 16 illustrates the optimized path delay of the proposed architecture in comparison with 32-bit RCA and 32-bit CSK with fixed size.

**Simulation Results**

HDL is used to create a hardware model by simulating a design. Applying proper input vectors and comparing the results with the predicted values at the output pins is referred as called simulation. The conventional and the proposed design is designed in Verilog HDL and implemented in Xilin FPGA XC3S400PQ208 board.
The 32 bit RCA consists of inputs A and B of 32-bit each, $C_{in}$ and 32-bit sum, $C_{out}$. RCA is designed in Verilog HDL. The Simulation waveform of this design is shown in figure 17. The inputs A, B and output S (sum bits) are represented in decimal format and the output carry $C_{out}$ represented in binary format. The some of the input combinations is shown in table II.

32-bit CSK with uniform module size is designed in Verilog HDL and implemented in XC3S400PQ208 FPGA board. The simulation waveform is shown in figure 18 and the inputs A, B and Sum are represented in decimal format. Some of the input combinations are shown in Table III.

<table>
<thead>
<tr>
<th>Table II. Addition of test vectors using RCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
</tr>
<tr>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>2354</td>
</tr>
<tr>
<td>234</td>
</tr>
<tr>
<td>1704</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table III. Addition of test vectors using CSK with uniform module size</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
</tr>
<tr>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>5702</td>
</tr>
<tr>
<td>1568</td>
</tr>
<tr>
<td>5708</td>
</tr>
</tbody>
</table>

The proposed design 32 bit is also designed in Verilog HDL simulation waveform is shown in figure 19. The inputs A, B and output sum is presented in Hexadecimal system and the input carry $C_{in}$ and output carry $C_{out}$ are
represented in binary number. Table IV explains some of the input combinations with its corresponding output.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C_in</th>
<th>sum</th>
<th>C_out</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1568</td>
<td>1470</td>
<td>1</td>
<td>3039</td>
<td>0</td>
</tr>
<tr>
<td>5708</td>
<td>5519</td>
<td>1</td>
<td>11238</td>
<td>0</td>
</tr>
<tr>
<td>5702</td>
<td>5710</td>
<td>0</td>
<td>11412</td>
<td>0</td>
</tr>
</tbody>
</table>

The proposed design is prototyped onto XC3S400PQ208 FPGA. The output is displayed through LEDs, by applying the inputs by the switches available on board. The prototyping output is depicted in figure 20 as below.

Figure 20. Prototype of proposed adder in FPGA.

5. CONCLUSION

CSK with non-uniform module size is designed to balance the delays in critical path by changing the block size. In the proposed design the carry chain delay will be reduced by skipping the carry over a block of bits instead of propagating through the blocks. The proposed adder uses a look ahead adder logic to improve carry chain delay. Proposed adder is designed using Verilog HDL and prototyped on to Xilinx FPGA board with 28.418ns delay.

REFERENCES


K. Murali Chandra Babu, Assistant Professor in Department of Electronics and Communication Engineering, Vignan Institute of Technology and Science. He published over 15 journals and conference papers including SCOPUS, IEEE and UGC. His research interests cover Low power design, Electronic circuits, VLSI, renewable energy resources and IoT based applications.

Dr. P. A. Harsha Vardhini, professor in the Department of Electronics and Communication, Vignan Institute of Technology and Science, Deshmukhi has nineteen years of teaching experience. She pursued her Ph.D in VLSI Design from the department of ECE, JNTUH. Her research interests include FPGA architectures, IoT Embedded Applications & Wireless Communication. She is an IEEE & WIE member and life member of IEI, ISOI, IAENG, IAOE. She has more than 100 publications in various journals and conferences at national and International level including SCI, SCOPUS, IEEE, Springer, Elsevier and UGC.