Designing of 128-Bit ALU (Arithmetic Logic Unit) using VHDL

Kapil¹, Ramnish Kumar²
¹Student, M.Tech (ECE), GJUS&T, Hisar
²Asst. Professor, ECE, GJUS&T, Hisar

Abstract: The VHDL (Very high speed integrated circuit Hardware Description Language) is a very popular tool for designing the digital system. In this paper, a VHDL design of 128-bit Arithmetic Logic Unit (ALU) is simulated. Mainly ALU is the basic and fundamental unit of a microprocessor which implements all the elementary operations established on the control input selection (Select Line). ALU looks like a multiplexer which performs all the operations on the basis of select lines. The ALU performs the addition, subtraction, comparison and all the logic operations using Xilinx ISE 8.1i tool. Here a mixed modeling of VHDL is used for implementation and synthesis ALU which includes both logical and arithmetic operations.

Keywords: - VHDL, ALU, Xilinx ISE 8.1i, mixed modelling

I. INTRODUCTION

The VHDL is a hardware description language used for analysis and synthesis of digital circuit. Now it is a standard by IEEE and appropriated by various FPGA and ASIC vendors. VHDL does not stand for any simulation control or monitoring skill within the language. These skills are tool dependent. There are many design automation tools which support VHDL have been developed by CAD (Computer Aided Design) engineering companies. Generally, VHDL language is operated by the two main tools i.e. simulation and synthesis.

II. ALU

An arithmetic logic unit (ALU) as shown in fig. 1 is a multi-operated device which performs combinational logic digital function with a set of basic arithmetic operations as well as logic operations. These two operations are depend upon the number of selection lines to select a particular operation in the unit. Mainly ALU contains two inputs which are controlled by the select line for the multi-operation. ALU defines for the two most important units and they are arithmetic and logic units. In this paper, the arithmetic unit defines for the 128-bit adder, 128-bit subtractor and 128-bit comparator where in the logic units we define all the gates of 128-bit.

A. Arithmetic Unit

The arithmetic unit responsible for the mathematic calculation. There are various types of arithmetic operations e.g. Addition, subtractions, comparison and they are details as:-

1) Adder: The simplest 1-bit adder calculates the output in term of the carry out and sum so, the relationship between input and output is defined below:

\[ \text{Sum} = A \text{ XOR } B \text{ XOR } \text{Cin} \]  
\[ \text{Cout} = (A) \text{ OR } (B) \text{ OR } (A \text{ AND Cin}) \]  

Where A and B are inputs and Cin is carry input.

The adder have many types on the basis of their performance. In this paper, we use the Carry Look ahead Adder (CLA) which is faster one. The ripple-carry adder has a limiting factor of the time that it take to propagate the carry. The Carry Look-ahead Adder (CLA) removed this limiting factor by calculating the carry signals in advance, based on the input signals. This result reduces the carry propagation time in term of carry propagate (P_i) and carry generate (G_i) and where P_i and G_i are:

\[ P_i = (A_i) \text{ XOR } (B_i) \]  
\[ G_i = (A_i \text{ AND B_i}) \]  

2) Subtractor: Subtractor is the digital circuit which is used for subtract two binary numbers (digit) and provides difference and borrow out as an output.

\[ D = (A \text{ XOR B}) \text{ XOR Bin} \]  
\[ B_{out} = ((\text{NOT A}) \text{ OR B}) \text{ OR ((NOT A) \text{ OR Bin})} \]  

Where A and B are inputs and Bin is borrow input.

Same here we use the Borrow Look ahead Subtractor (BLS) which is also faster one.
B. Logical Unit

In this unit, the logic operations are performed. Here two different inputs are fed at input and at the output we will obtain the logical output. In digital systems, there are seven gates which perform the various logics and they are OR, AND, NOT, XOR, XNOR, NAND, and NOR gate. The logic unit defines the logic level of the signals. It defines the signal on the two levels i.e. high level and low level. Mainly high level for ‘1’ and low level for the ‘0’.

C. Xilinx Simulation

Simulation of 128-Bit ALU for the mixed model has been performed for 1000 nano-seconds (ns). Each Clock cycle has 10 ns rise time and 10 ns fall time. The simulation of 128-Bit ALU (if rising edge (CLK) = 0). This RTL is generated by the Xilinx simulation. The fig. 2 shows the RTL view. Also, the waveform is generated by the software with the help of the testbench. The waveform is shown in fig.3. In this simulation the spartan3 family is consider and the device is xc3s200 as well the package is FT256 and the speed is taken of -4.
The fig. 3 shows the waveform where a and b are input bits of 128 bit. The select line selects the particular logic and it is indicated by sl. The input carry and borrow are shown by $c_{in}$ and $b_{in}$ respectively and the outputs of adder are sum and $c_{out}$ (carry out) and output for subtractor are diff. and $b_{out}$ (borrow out). Where the comparator output shown by less, gr (greater) and eq (equal) and the last seven outputs are logic output. The undefined line takes place because there is no action consider by the program.

### III. RESULT

<table>
<thead>
<tr>
<th>Logic Utilization</th>
<th>Used</th>
<th>Available</th>
<th>Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of Slices</td>
<td>857</td>
<td>960</td>
<td>89%</td>
</tr>
<tr>
<td>No. of Slice Flip-Flop</td>
<td>65</td>
<td>1920</td>
<td>3%</td>
</tr>
<tr>
<td>No. of 4 input LUTs</td>
<td>1521</td>
<td>1920</td>
<td>79%</td>
</tr>
<tr>
<td>No. of bounded IOBs</td>
<td>279</td>
<td>108</td>
<td>258%</td>
</tr>
<tr>
<td>No. of GCLKs</td>
<td>1</td>
<td>24</td>
<td>4%</td>
</tr>
</tbody>
</table>

The 64-bit ALU shows the details in table 1 and in fig. 4 which is conventional result. The bar graph shows the highest peak of No. of bounded IOBs is 258 % whereas lowest peak is of No. of slices flip-flop is 3 %.
Table 2. 128-Bit ALU

<table>
<thead>
<tr>
<th>Logic Utilization</th>
<th>Used</th>
<th>Available</th>
<th>Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of Slices</td>
<td>946</td>
<td>1920</td>
<td>49%</td>
</tr>
<tr>
<td>No. of Slice Flip-Flop</td>
<td>255</td>
<td>3840</td>
<td>6%</td>
</tr>
<tr>
<td>No. of 4 input LUTs</td>
<td>1678</td>
<td>3840</td>
<td>43%</td>
</tr>
<tr>
<td>No. of bounded IOBs</td>
<td>1419</td>
<td>173</td>
<td>820%</td>
</tr>
<tr>
<td>No. of GCLKs</td>
<td>8</td>
<td>8</td>
<td>100%</td>
</tr>
</tbody>
</table>

The 128-bit ALU shows the details in table 2 and in fig. 5 which is proposed result. The bar graph shows the highest peak of No. of bounded IOBs is 820% whereas lowest peak is of No. of slices flip-flop is 6%. This huge difference made because of the no. of bit is doubled. The no. of bounded IOBs just detailed about the no. of input and output which are considered while it is programmed.

![Utilization Graph](image)

Figure 5. Utilization of 128-bit ALU

Table 3. Delay report comparison of ALU

<table>
<thead>
<tr>
<th>Parameter</th>
<th>64-Bit ALU</th>
<th>128-Bit ALU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min. Period</td>
<td>2.054 ns</td>
<td>2.332 ns</td>
</tr>
<tr>
<td>Max. Frequency</td>
<td>486.855 Mhz</td>
<td>428.816 Mhz</td>
</tr>
<tr>
<td>Min. input arrival time before clock</td>
<td>94.949 ns</td>
<td>13.748 ns</td>
</tr>
<tr>
<td>Max. output required time after clock</td>
<td>10.535 ns</td>
<td>8.574 ns</td>
</tr>
<tr>
<td>Max. combinational path delay</td>
<td>No path found</td>
<td>9.32 ns</td>
</tr>
</tbody>
</table>
IV. DISCUSSION

The delay report comparison between 64-bit and 128-bit of ALU is shown in fig.6 and in table 3. We can conclude that the delay of 128-bit ALU is much better than 64-bit. Also the no. of 4 input LUTs and no. of slices also better then that of 64bit ALU. It is obvious that as we increase the no. bit then the no. of bounded IOBs will increase.

V. CONCLUSION

The 64 bit ALU is designed and synthesized using Xilinx ISE v8.1i and targeted to Spartan 3 device. The ALU is a very important part of the CPU (Central Processing Unit). Its arithmetic unit performs the Addition, Subtraction, Comparator and all basic logical operations (AND, OR, NOT, NOR, XOR, XNOR, NAND). We analysed the results from Xilinx ISE Design Suit v8.1i with the theoretical results for all the operations that were performed and found that they ideal with the theoretical result.

REFERENCE


AUTHORS

Kapil, He received his B.Tech in Electronics and Communication Engineering from O.I.T.M., Hisar, Haryana and pursuing M.Tech in Electronics and Communication Engineering from Guru Jambheshwar University of Science and Technology, Hisar, Haryana, India. His area of interest is in Digital Electronics and VHDL simulation.

Ramnish Kumar, He received his B.E. (Electronics and Communication) from C.R.S.C.E. (now DCRUST), Murthal, Sonipat, Haryana; M.Tech. from R.E.C., Kurukshetra; Ph.D. from G.J.U.S.&T., Hisar, Haryana. Presently, he is working as an Assistant Professor in Department of E.C.E., G.J.U.S. &T., Hisar, Haryana. His present research interests include device modeling, control system and Digital Signal Processing.