Experiment : 4 Bit Ripple Carry Adder

Badal Varshney, Roll Number 19D070015

EE-214, WEL, IIT Bombay

January 28, 2021

## Overview of the experiment:

|  |
| --- |
| The purpose of the experiment was to design a system capable of adding two four bit numbers. Since the design of a Full Adder was already implemented, the next natural step was to design a four bit adder. The experiment demonstrated how the ripple carry in the Full Adder may be used to design a four bit adder. The design was implemented for two 4 bit numbers but can easily be extended to more number of bits too.  The design of the project was done in Quartus Prime and a generic testbench was used which reads data from a trace file containing the test cases to check whether the expected answers match the outputs. |

## Approach to the experiment:

|  |
| --- |
| The 4 bit adder was implemented using 4 Full Adders. The carry for the first adder was initialized to 0 and the two inputs to it were the first bits of the 2 numbers. The sum would be the value of the output’s corresponding bit and carry was propagated to the next adder. The last carry generated from the Full Adder was ignored. The design used where A and B are the two inputs . Testbench was compiled which read data from the trace file whose first column was the 8 bit input, the second column was the expected output and the last was the mask bit to determine whether to check the corresponding bit or not( checked if set to 1). The trace file was generated using python code. |

## Design document and VHDL code if relevant:

|  |
| --- |
| Code of Ripple Carry adder-    Code of DUT file- |

## RTL View:

|  |
| --- |
| Attach screen-shot of the RTL view generated by Quartus. |

## DUT Input/Output Format:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Mention the format (LSB/MSB of input and output) and few test cases from trace-file.   |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1st Input | | | | 2nd Input | | | | Output | | | | | MSB |  |  | LSB | MSB |  |  | LSB | MSB |  |  | LSB | | B3 | B2 | B1 | B0 | A3 | A2 | A1 | A0 | O4 | O3 | O2 | O1 | | 1  1 | 1  1 | 1  1 | 1  1 | 1  1 | 1  1 | 1  1 | 0  1 | 1  1 | 1  1 | 0  1 | 1  0 | |

## RTL Simulation:

|  |
| --- |
| Attach the clearly visible screen-shot of RTL simulation waveforms. |

## Gate-level Simulation:

|  |
| --- |
| Attach the clearly visible screen-shot of Gate-level Simulation. |

## Krypton board\*:

|  |
| --- |
| Map the logic circuit to the Krypton board and attach the images of the pin assignment and output observed on the board (switches/LEDs). |

## Observations\*:

|  |
| --- |
| You must summarize your observations, either in words, using figures and/or tables. |

## References:

|  |
| --- |
| You may include the references if any. |

\* To be submitted after the tutorial on ”Using Krypton.