EXP. 8 **Clock** Date: 10/25/2021

# Team members:

1 – Yaman Shullar – 201954350

2 - Ahmad Alzhrani – 201917030

|  |
| --- |
| 1. Objectives:   * Extending our knowledge in Verilog, including how can we implement a multiplexer using assign and always in addition to design a n-bit counter * Learning how to produce a slower signal from a faster one * Understanding how to utilize oscilloscopes with FPGA board * Introducing clock signal and clock frequency |
| 2. Procedure:  Task 4.1:   1. I opened a new project in the software with Verilog 2. I wrote the statements needed for implementing a main clock 3. I made a UCF file 4. I saved and cleaned up files   Task 4.2:   1. I created a new source 2. I wrote the code needed for implementing a n-bit counter 3. I specified the number of bits at first to be 27 bits and then 8 bits 4. I made a top module, making an instance of counter in it 5. I made a UCF file for the one with 27 bits 6. I made a simulation for the one with 8 bits 7. I implemented my work on a FPGA 8. I saved and cleaned up files     Task 4.3:   1. I opened new project 2. I crated new source which is Verilog module 3. I named it as the task 3 4. I wrote the code for the counter 5. I wrote the code for the top module 6. I simulated the module 7. I changed the module 8. I created the ucf file 9. I connected the inputs and output to the fpga board 10. I tried the experiment 11. I cleaned the project and take snapshots |
| 3. Problems Faced:   * Yaman: No problems faced * Ahmed: the zooming in the simulation was different each time.   4. Work Distribution:   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Names | Task 4.1 | Task 4.2 | Task 4.3 | Answers and snapshots | Lab report | | Yaman | 100% | 100% | 0% | 70% | 45% | | Ahmed | 0% | 0% | 100% | 30% | 55% | |
| 5. Snapshots:    Snapshot 1. Main Clock Signal implementation in Verilog    Snapshot 2. Digital Frequency Divider implementation in Verilog (27 bits)    Snapshot 3. Digital Frequency Divider implementation in Verilog (8 bits)  Graphical user interface, text, email  Description automatically generated  Snapshot 4. Verilog code unmodified task3  Graphical user interface, text, email  Description automatically generated  Snapshot 5. Verilog code modified task3  Graphical user interface  Description automatically generatedGraphical user interface  Description automatically generated  Chart  Description automatically generatedChart  Description automatically generated  Graphical user interface  Description automatically generated with medium confidenceGraphical user interface  Description automatically generated with medium confidence  A picture containing graphical user interface  Description automatically generatedA picture containing graphical user interface  Description automatically generated  Snapshot 6. The simulation of the Digital Frequency Divider (8 bits) task3        Snapshot 7. The simulation of the Digital Frequency Divider (8 bits)  Answers for questions:  Q1/ the frequency is very fast to be observed on a counter, hence, dividing it will slower it and solve the problem  Q2/ advantage: it is very easy to use by writing a code  Disadvantage: counters will not give us an exact frequency (but approximated one)  Q3/   |  |  |  |  | | --- | --- | --- | --- | | Select lines (S1 S0 ) | Value of n | Approx. Frequency (By divide by 2) | Actual Frequency (By calculations) | | 00 | 23 | 8 hz | 5.96 hz | | 01 | 24 | 4 hz | 2.98 hz | | 10 | 25 | 2 hz | 1.49 hz | | 11 | 26 | 1 hz | 0.745 hz |   6. Conclusion:  We learned in this lab new methods to use Verilog and new applications, and we learned how to make a counter in Verilog which can be used with different (n) and can be dynamic, and we learned to use the top module and know the difference in frequency by simulation. |