McGill University

ECSE 325

Digital Systems

Lab 1 Report

Authors (Group 07):	*Student Number:*
John Wu	260612056
Ege Ozer	260688342

March 2, 2018

Contents

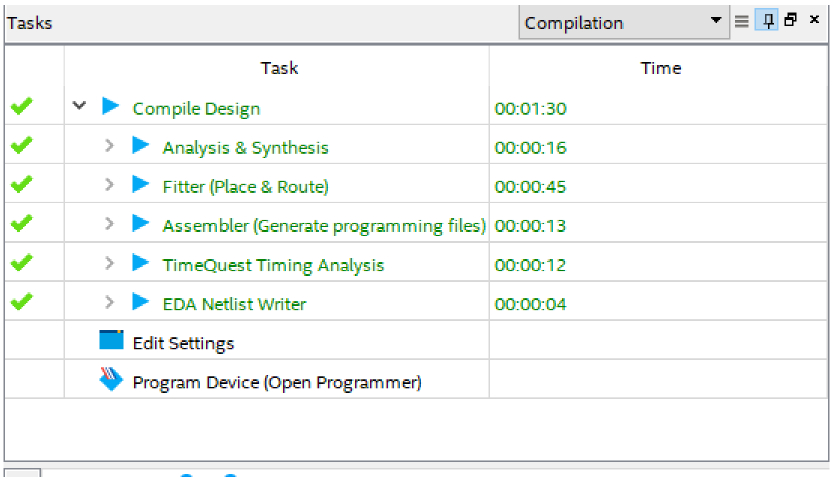
1 Introduction

The goal of the following lab is to learn the basics of the Quartus-II FPGA software as well as how to compile the synchronous circuit VHDL declarations to a target FPGA. We will also be looking at how the fitter of the Quartus compiler maps the design to the FPGA hardware. In order to do so, we will be going through the process of making an 8-bit counter which has functions for counting up, counting down, and reseting.

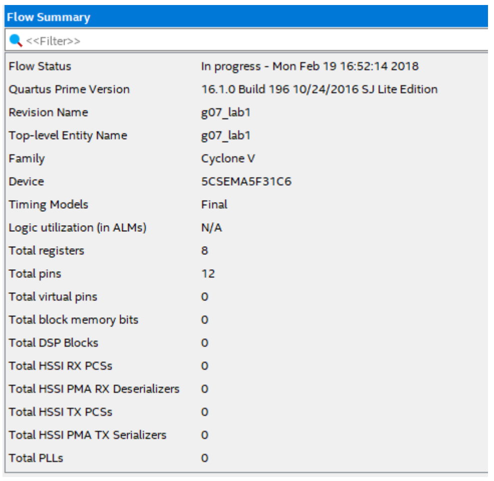
2 VHDL Code

See attached g07_lab1.file for the VHDL code that implements the 8-bit counter.

3 Compilation Report



	Task	Time
✓	Compile Design	00:01:30
✓	Analysis & Synthesis	00:00:16
✓	Fitter (Place & Route)	00:00:45
✓	Assembler (Generate programming files)	00:00:13
✓	TimeQuest Timing Analysis	00:00:12
✓	EDA Netlist Writer	00:00:04
	Edit Settings	
	Program Device (Open Programmer)	



Flow Status	In progress - Mon Feb 19 16:52:14 2018
Quartus Prime Version	16.1.0 Build 196 10/24/2016 SJ Lite Edition
Revision Name	g07_lab1
Top-level Entity Name	g07_lab1
Family	Cyclone V
Device	5CSEMA5F31C6
Timing Models	Final
Logic utilization (in ALMs)	N/A
Total registers	8
Total pins	12
Total virtual pins	0
Total block memory bits	0
Total DSP Blocks	0
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	0

Shown above on the left is our successful compilation report. On the right, we have a flow summary of the compilation. We can see from the summary that the design was successfully uploaded to the 5CSEMA5F31C6 device that is part of the Cyclone V family. In addition, we can notice that a summary of the resource utilization in the report which will be described in the next section.

4 Resource Utilization

From the RTL-viewer we observe that we have three 2X1 Multiplexers(MUX), two adders and one D-flipflop. More precisely with the flowchart above, it states that our design uses a total of 12 registers and 8 pins. we can think of few possibilities that might cause more recources to be used than it should when we back-track the implementation of our design.

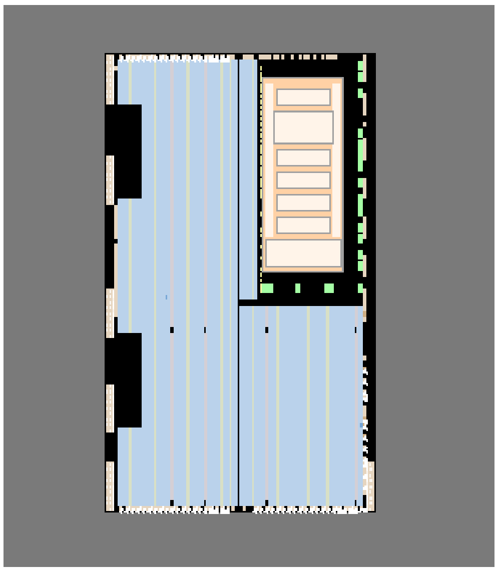
Our design's 8-bit counter variable is using 2^n integers where n represents the amount of bits used as a way of storing the values. Although it sounds logical, this decision might be problematic as it may take more resources. For instance, an 8 bit counter requires a signal with a range $[0, 255]$ to store $2^8 = 256$ numbers. Now to put into perspective, suppose we have a 32-bit counter, this would imply that we would require $2^{32} = 4294967296$ values for own counter signal.

Additionally, since the output is declared using `STD_Logic_Vector`, we need to convert it to unsigned type in order to prevent compilation errors. However, this might have caused additional ressources and overhead for our 8-bit counter. We presume that the overhead will increase exponentially with the increase of the bit size of the counter.

If we could re-design our 8-bit counter, we would use only `STD_Logic_Vector` for the counter variable. This would make future designs and higher bit counters more practical and more efficient to implement. n-bit counters would only require a range of $[0, n-1]$ instead of $[0, 2^n-1]$. As a result, this fix would also get rid of type casting conversion and the use of `IEEE.NUMERIC_STD.ALL` that migh create an unnecessary overhead.

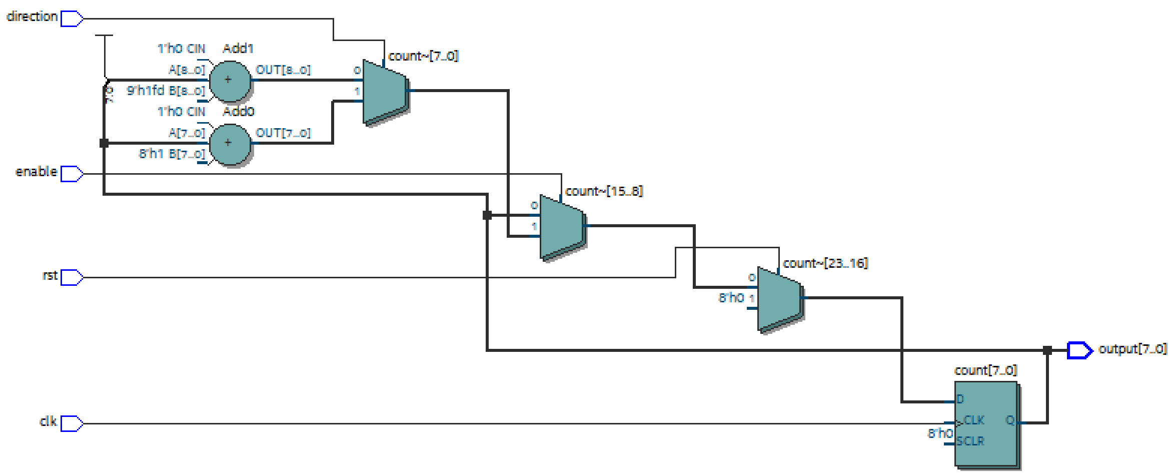
To summarise, the resource utilization could be improved by using only `STD_Logic` type variables which would also get rid of overhead.

4.1 Chip Planner



Chip planner screenshot, with used resources highlighted.

4.2 RTL View



RTL view of the circuit, with description of resource

5 Conclusion

By completing this lab, we have covered the basics of using an FPGA software and learning how to debug code that is written in a hardware description language. In this particular experiment, very few resources were needed to implement our 8-bit counter in the FPGA. However, the techniques learned in this experiment for interpreting the VHDL design will become crucial skills in future labs when the designs become more complex and resource management is crucial.