**Advanced Microelectronics Lab**

# (ELCT 1005)

# Lab Report 4

|  |
| --- |
| *Martel Megalaa 49-3843* |

## Introduction:

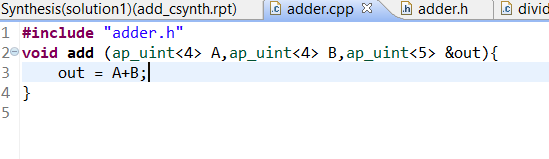
Over the years, digital designs are getting more complex. To eliminate this complexities in lareg designs, high level synthesis is used. High level synthesis is writing a c++ code, and this code is then converted to hardware description language. Hence, this approach reduces design complexity in large designs .

## Task 1:

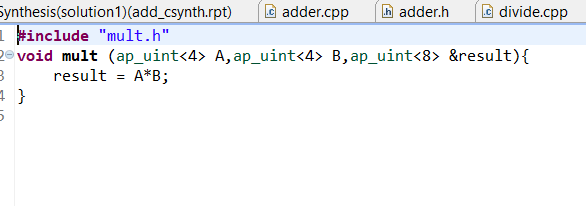
**A. Methodology and Implementation**

Task was about generating 4-bit-adder, 4-bit-multiplier and 4-bit-divider using C++ , and then utilizing vivado HLS tool to convert this C++ files to vhdl files. Here are all the C++ files and the header files are very basic containing function headers (“#include ap\_int.h” statement)

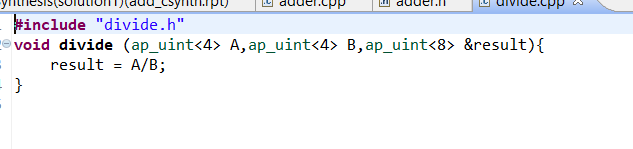
* + - **Adder:**



* + - **Multiplier**:

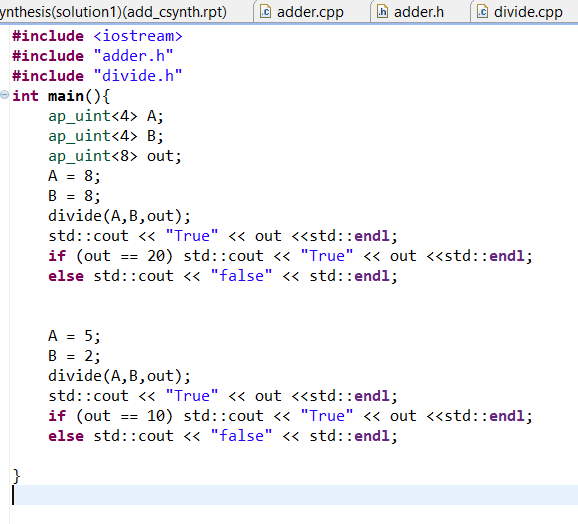


* + - **Divider:**



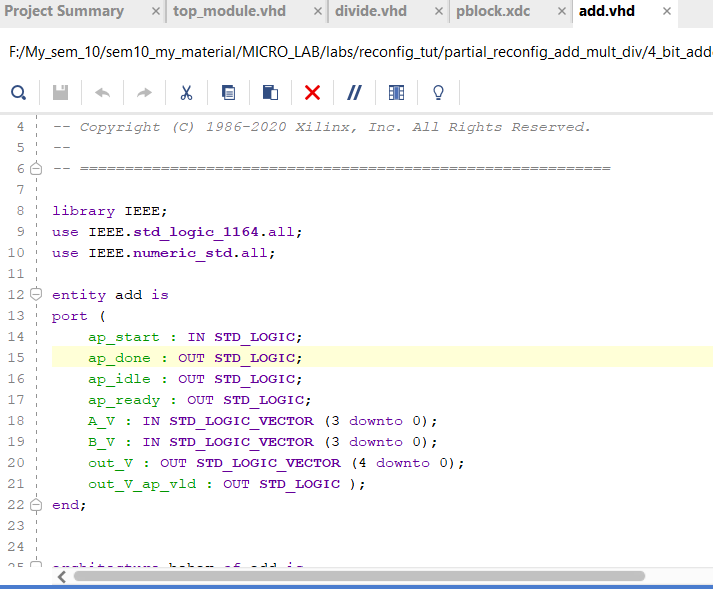
**Testbench:**

This TB was global for all files. With just changing the name of function between divide, add and multiply, I have tested all functions and it is working correctly.

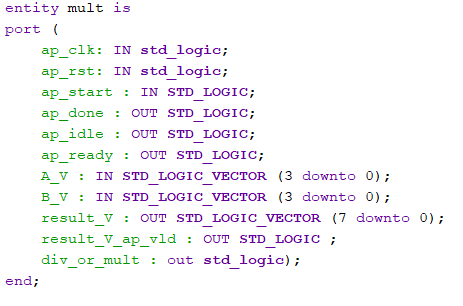
****

**Output after C synthesis of the above files:**

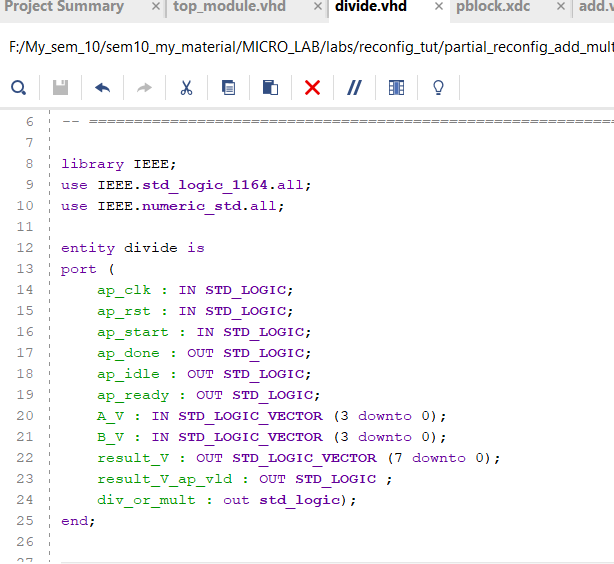
* + - * **Adder: (Adder I/O Ports)**



* + - * **Multiplier:**

****

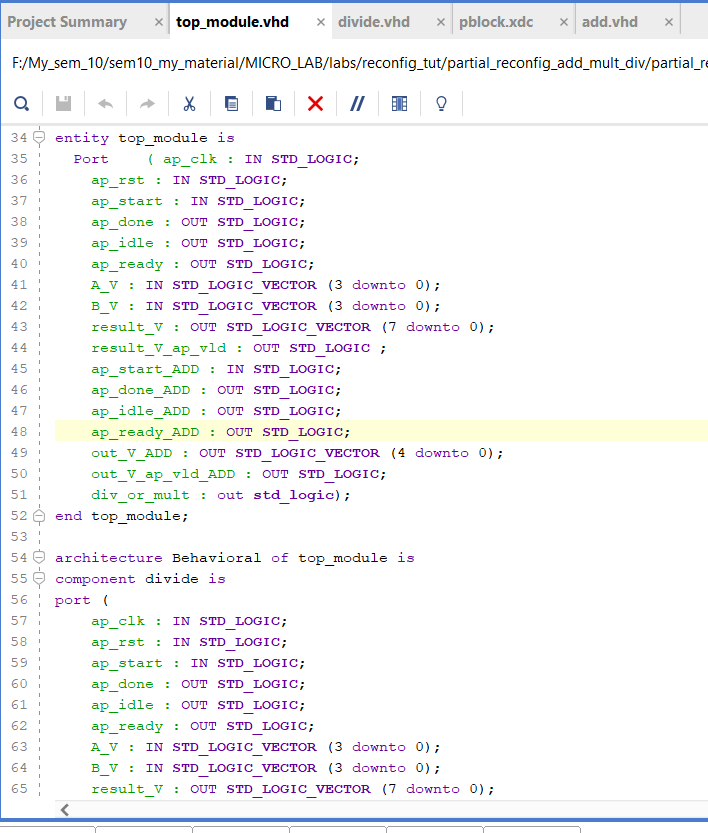
* + - * **Divider:**

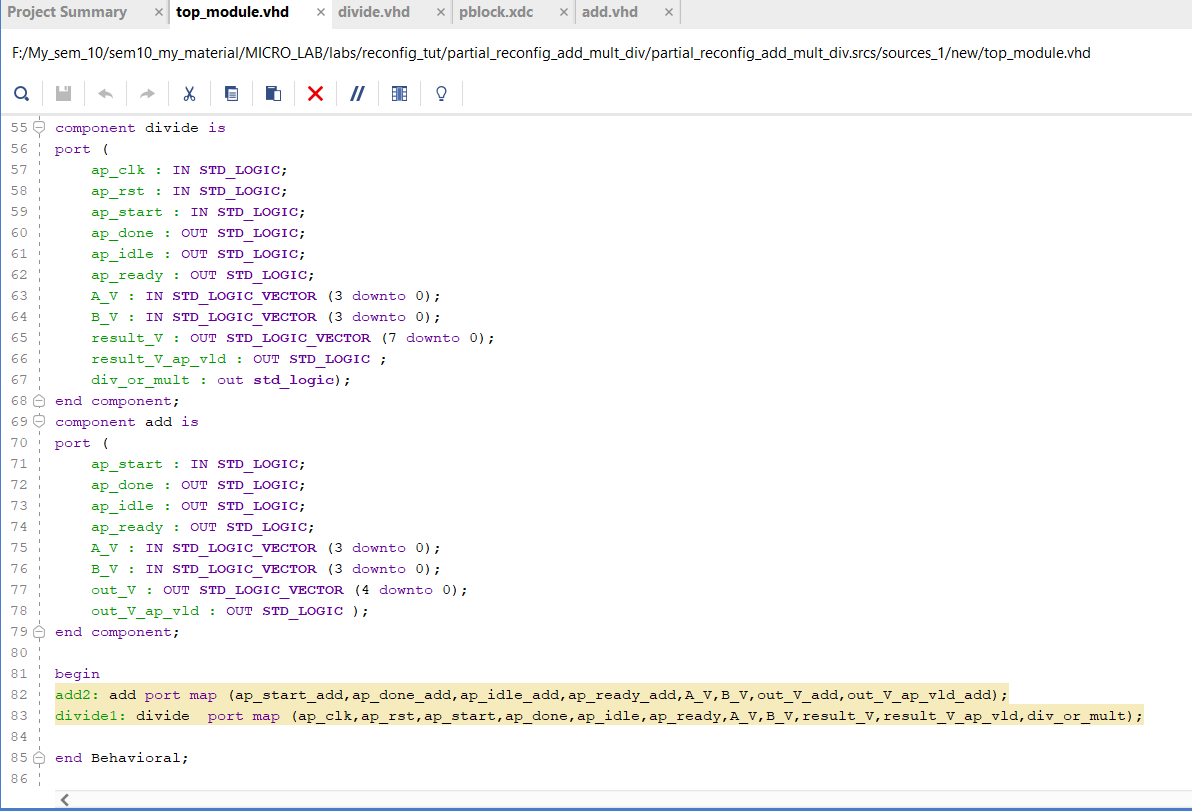


**Task 2:**

Task 2 was about using partial reconfiguration feature in Basys3-FPGA and implementing a design having adder as static solution and the dynamic solutions are the multiplier and divider. My methodology to prove that fpga is partially programmed is by an LED which is turned on if multiplier is used and off otherwise. Here is the used VHDL TOP module along with vhdl codes of adder, multiplier and divider:

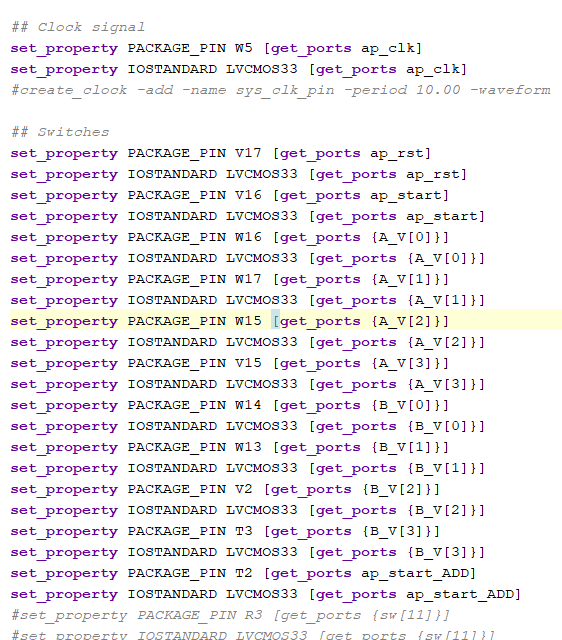
Top module:

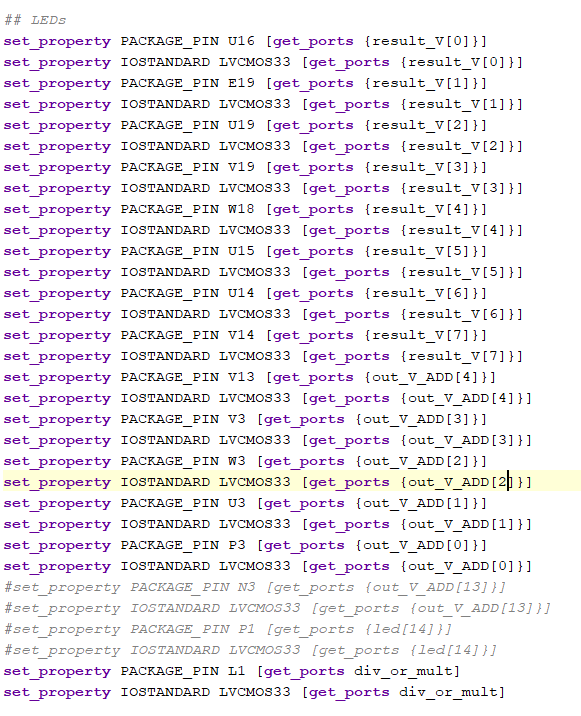




**Constraint Files:**

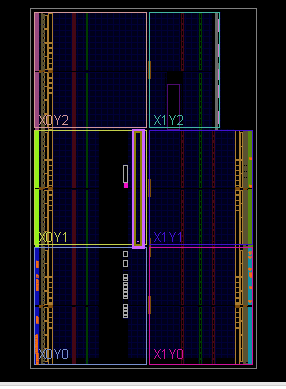
* + - * **Basys Constraint file:**

****

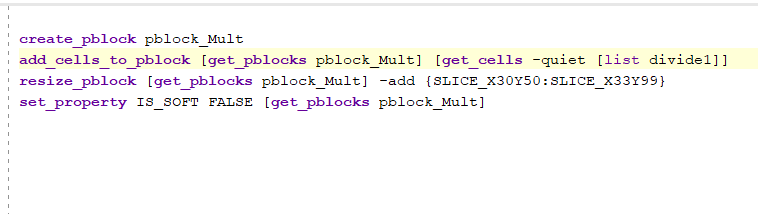
****

**PBlock Constraint:**

* + - * **Used area on the fpga:**

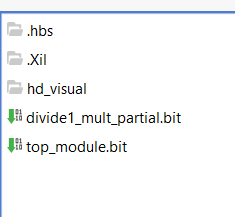
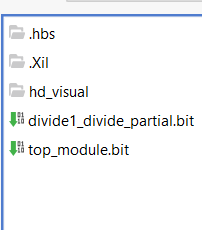


* + - * **Constraint file:**

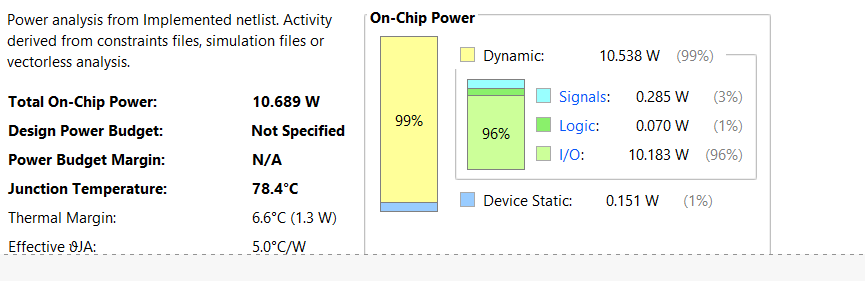


**Output Results:**

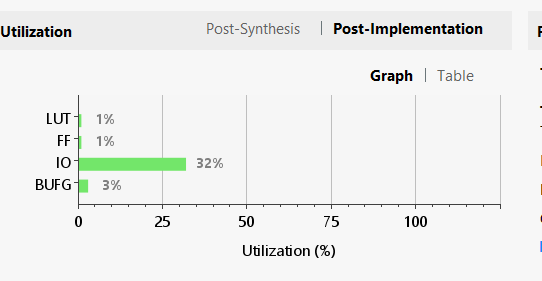
* + - * **BIT FILES:**



1. **Power Utilization**



1. **Area Utilization**



1. **Conclusion**

After testing both bitstreams (mult and divide) with the top module bitstream, I have found that the LED associated to specifying the module used (multiplier or divider ) is toggling when I am switching between the two bitstreams. Hence, Partial reconfiguration on the FPGA is approved.