CPU Architecture

Second Laboratory

Arithmetic and Floating point Synthesis

Hanan Ribo, Boris Braginsky and Prof. Hugo Guterman

15.04.2018

Table of contents

List of Figures

List of Tables

1. Aim of the Laboratory

- Understanding arithmetic synthesis and FPGA arithmetic limitations.
- ALU design
- Floating point design

2. Assignment definition

In this laboratory you will have to synthesize an ALU from the first assignment for the Cyclone II FPGA with impact on performance and logic usage. You will have to do the two following test cases for the ALU. Additionally, to ALU floating point hardware will be designed. The following commands must be implemented with floating point hardware:

1. MULF C = A * B
2. ADDF C = A+C

where A,B and C 32 bit numbers in floating point IEEE standard.

The functions of floating point can't be taken from existing libraries.

Performance Test Case

In This test case you have to test the performance and functionality of the 8bit ALU(with floating point). The ALU will be placed between two synchronous registers (DFF based) operating from the same clock as described in the diagram below to estimate the performance.

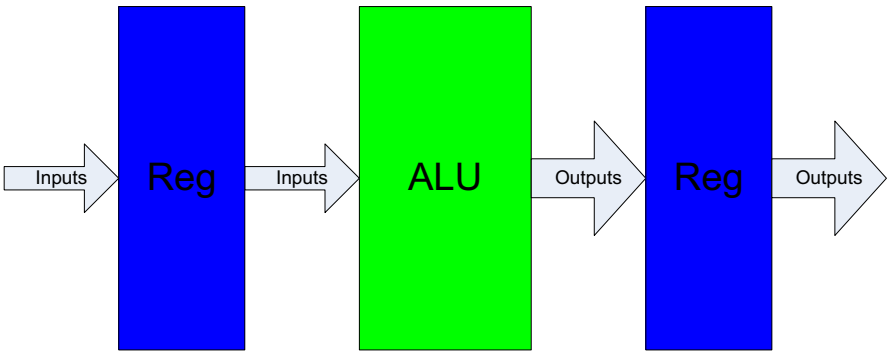

Figure 1 : First Test Case

You have to do the following tasks:

- Modelsim Simulation with maximal coverage

- Quartus Compilation without pin assignments and design loading to check the performance
- Find the maximal operating clock, set the clock constrain to large value
- Analyze the critical path
- Analyze the logic usage
- Find the frequency limiting operation and explain why it is happening
- Propose solution for CPU frequency improvements in two cases (current ALU)
    - The problematic operation is commonly used in software
    - The problematic operation is almost unused

The following must be presented in the report.

- RTL Viewer results for each logic block
- Logic usage for each block (Combinational and Flip-Flops)
- Critical path for each logic block and overall system critical path
- Optimizations that you have done on the code for the FPGA
- Logic usage report from Quartus II

Hardware Test Case

In the hardware test case you will have to test an 8bit ALU and floating point at the FPGA board. A board switches (SW) and push buttons (KEY) will be used to select the input data and output data. The ALU input data is generated using 3 registers connected in parallel to the board switched with 3 push buttons serving as register enables. All the system must work from a single 50MHz (non gated clock). All the system must be connected to the Altera board interfaces according to the following diagram. For floating point commands (described above) the eight-bit number will be transfer to floating point format while first two bits will be describing the fraction (LSB) and entire six bits the integer part, MSB describe sign of the number (see the following example).

010101 11 = $21\frac{3}{4}$

The result of floating point commands will be present on seven segment display in floating point format. While each segment (total 4) can be used for present 4 bit in HEX format. Using SW8 ones will be present 16 LSB bits and ones 16 MSB bits, for example:

SW8 = 1 => 16 MSB bits presented.

SW8 = 0 => 16 LSB bits presented.

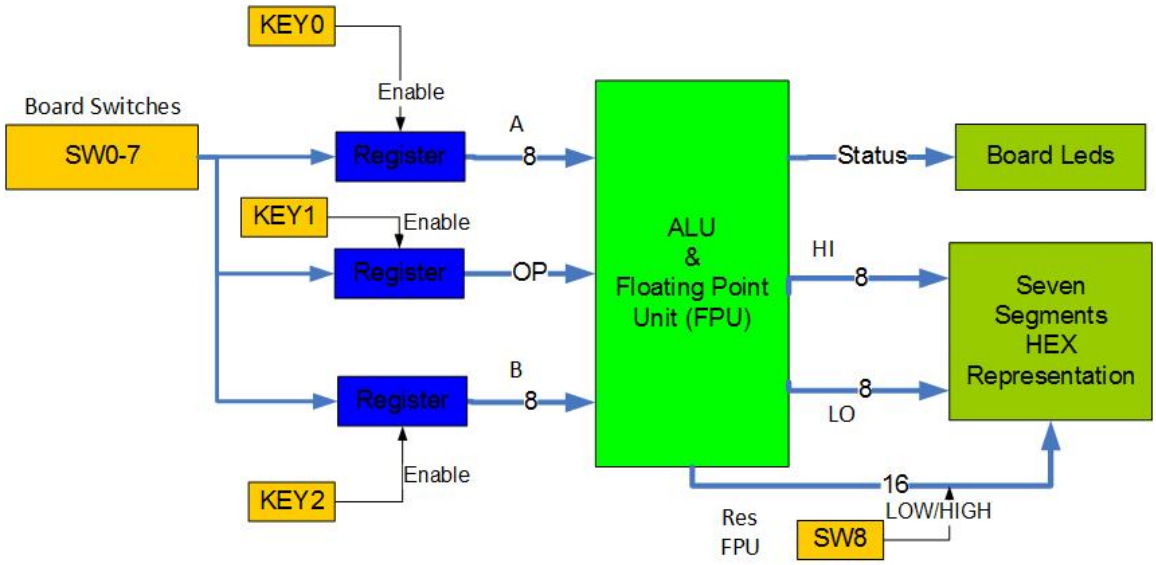

Figure 3: Hardware Test Case

3. Requirements

1. Contents with page numbers
2. Images and tables will be numbered. The caption of an images and tables below the images or tables

   The top level design must be structural, all other modules can be structural/ behavioral or mixed.
3. The behavioral parts of the design (except the test bench) must be synthesizable, pay attention on the logic that you are describing.
4. Block diagrams for the behavioral parts of the design that describes the logic behind the behavioral code , can be taken from Quartus RTL Viewer
5. The design must be well commented.
6. The system must work from only one clock.
7. All the resets in the system must be synchronous.
8. Do not use latches, the system must be fully synchronous.
9. **Important:** For each module :
   - Graphical description (a square with ports going in and out).
   - Port Table (direction, size, functionality).
   - Short descriptions.
   - Block Diagram of the internals
   - Critical Path
10. Elaborated analysis and wave forms:
    - Remove irrelevant signals.
    - Zoom on regions of interest.
    - Draw clouds on the waveform with explanations of what is happening
    - Change the waveform colors in Modelsim for clear documentation (Tools->Edit Preferences->Wave Windows).
    - Resource Usage from Quartus
    - Maximal Frequency and critical paths from Timing Analyzer
    - Proof of work using Signal Tap and **Saleae** scope plots.

11. Conclusions
12. A zip file in the form of id1_id2.zip where id1 and id2 are the identification number of the submitters, and id1 < id2. The file will contain :
    - The full *.doc or *.pdf report file, with operation manual of the watch.
    - The Modelsim simulations
    - **All the Quartus project including the VHDL files and the project itself**
    - The SignalTap files used in project debug
    - Quartus Project SOF file
    - The ZIP must not contain temporary files (Files that are produced by compiler and not required for the compilation)
    - The ZIP file directory structure must be arranged according to Table 1.
    - After organizing the ZIP file, verify compilation and that no files are missing.

Directory	**Contains**	**Comments**
VHDL	Project VHDL files	Only VHDL files , excluding test bench
TB	VHDL files that are used for test bench	
SIM	Modelsim project and DO files	Do not place files that are not relevant for compilation or is a result of compilation
DOC	Project documentation	Readme.txt and PDF report file
Quartus	Quartus Project files	Do not place files that are not relevant for compilation or is a result of compilation

Table 1 : Directory Structure

4. Grading Policy

Weight	Task	Description
30%	Requirements	The entire list of requirements in the assignment paper
20%	Documentation	The "clear" way in which you presented the requirements and the analysis
20%	Design	The overall system design, code styling, comments, logic usage and code/files organization.
20%	Analysis and Test	The correct analysis of the system and test bench coverage
10%	Conclusions	Asserted conclusions on the work you've done

Table 1 : Grading

Under the above policies you'll be also evaluated using common sense:

- Your files will be compiled and checked, the system must work.
- Your design and architecture must be intelligent, effective and well organized.
- The design must be well commented

Submission date 6/05/2018 using email: borisbr@post.bgu.ac.il with subject LAB2 CPU.

For a late submission the penalty is 2^days;

For early submission the reward is 2*days (up to 3 days);

5. References

[1] Altera Cyclone II data book on http://www.altera.com./literature/lit-cyc2.jsp
[2] Quartus II manuals : http://www.altera.com/support/software/sof-quartus.html
[3] DE1 User Manual on http://hl2.bgu.ac.il – Course Library=>FPGA
[4] Cyclone II Technical information http://www.altera.com./literature/lit-cyc2.jsp