CPU Architecture

LAB1 preparation report

VHDL part1 – Concurrent code

Hanan Ribo

23/03/2020

Table of contents

1. Aim of the Laboratory

- Obtaining skills in VHDL part1 code, which contains Code Structure, Data Types, Operators and Attributes, Concurrent Code, Design Hierarchy, Packages and Components.
- Obtaining basic skills in ModelSim (multi-language HDL simulation environment).
- General knowledge rehearsal in digital systems.
- Proper analysis and understanding of architecture design.

2. System Design definition

In this laboratory you will design a module which contains the next two sub-modules:

- Generic Adder/Subtractor between two vectors x, y of size n-bit (the default value is n=8).
- Barrel Shifter of 8-bit (its inner components are generic).

Sel	Operation	Note
0	Res=X+Y	
1	Res=X+Y+cin	
2	Res=X-Y	
3	Res=RLA X,Y(2 to 0)	Rotate left X of Y(2..0) times

Table 1: Selected operations

You are required to design the whole system and make a test bench for testing.

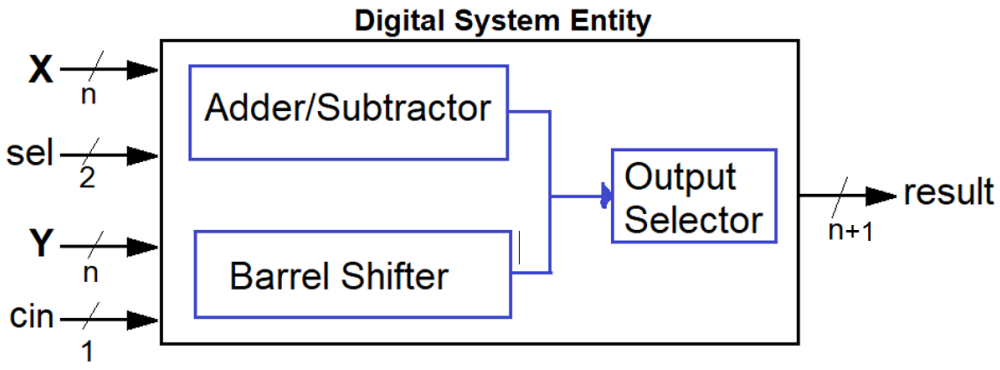

Figure 1 : System structure

- The Top Level design must be Structural and contains the three entities, as depicted in figure 1.
- you are given the next two files that you must use in your project: **top.vhd**, **aux_package.vhd (**you can only add code to these files, you are not allowed to erase nothing**).**

3. Generic Adder/Subtractor module:

- You are required to design a Generic Adder/Subtractor between two vectors x, y (represented in 2's complement) of size n-bit (the default value is n=8), using the next diagram. The design must be Structural of a least two levels.
- In order to do so, you are asked to use the Generic Adder code, that was given in Moodle.

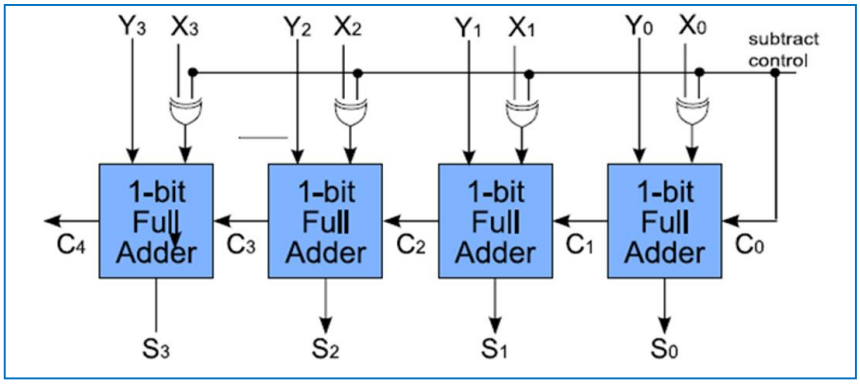

Figure 2: Generic Adder/Subtractor

4. Barrel Shifter of 8-bit Module:

You are required to design a Barrel Shifter of 8-bit (its inner components are generic). The encompassed parts in red has to be generic and their interconnections must be structural. (using sll, srl are not allowed).

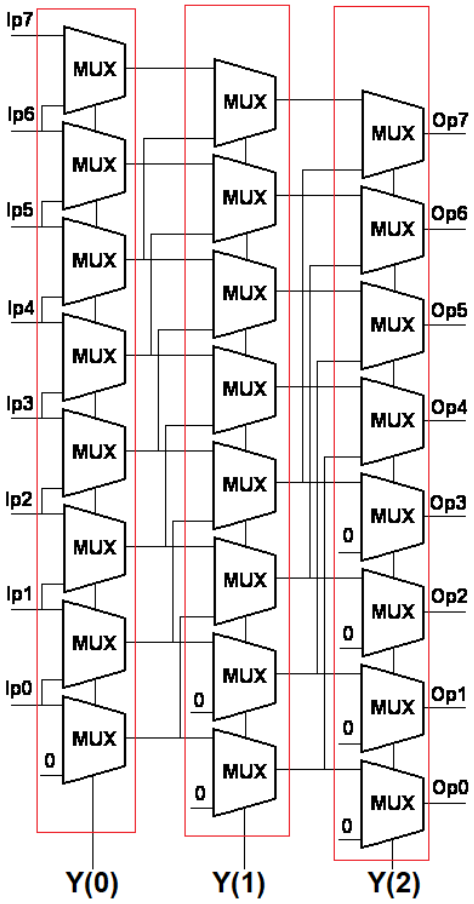

Figure 3: Barrel Shifter of 8-bit

5. Test and Timing:

- Design a test bench which tests all the system.
- Analyze the results by zooming on the important transactions in the waveforms. explain these (input/output/internal signals of the system).
- You are welcome to use the Internet also as reference.
- **Good tip for beginners**: Build a test bench for each module you are designing for easy debugging, otherwise you will waste a lot of time for whole system debug.
- The timing of the system will be ideal (means a functional simulation).

6. Requirements

1. Due to the situation, the lab assignment is for individuals (not in pairs).
2. The design must be well commented.
3. **Important:** For each of two submodules:
   - Graphical description (a square with ports going in and out) and short descriptions.
4. Elaborated analysis and wave forms:
   - Remove irrelevant signals.
   - Zoom on regions of interest.
   - Draw clouds on the waveform with explanations of what is happening (Figure 4).
   - Change the waveform colors in ModelSim for clear documentation
     (Tools->Edit Preferences->Wave Windows).
5. A ZIP file in the form of **id1_id2.zip** (where id1 and id2 are the identification number of the submitters, and id1 < id2) must be upload to Moodle only by id1 student. The ZIP file will contain:

Directory	Contains	Comments
VHDL	Project VHDL files	Only VHDL files, excluding test bench **Note: your project files must be well compiled without errors as a basic condition before submission**
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

Table 2: Directory Structure

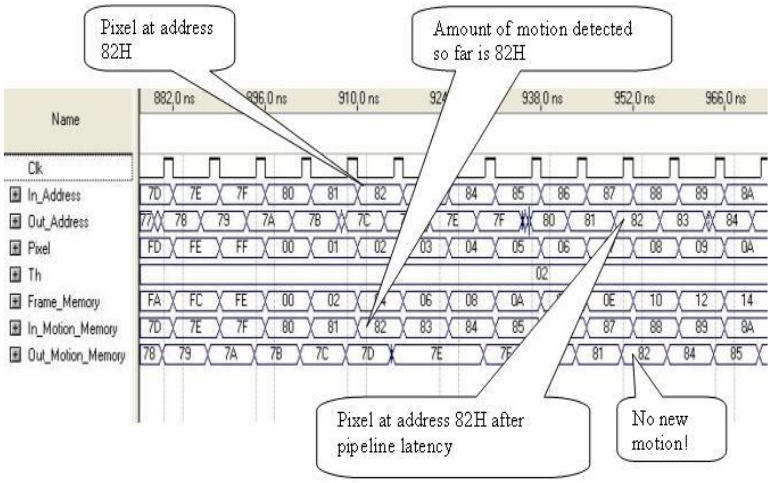

Figure 4: Clouds over the waveform

7. Grading Policy

Weight	Task	Description
10%	Documentation	The "clear" way in which you presented the requirements and the analysis and conclusions on the work you've done
90%	Analysis and Test	The correct analysis of the system (under the requirements)

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, minimal, effective and well organized.

For a late submission the penalty is 2^{days}