CPU Architecture

LAB3 assignment

Digital System Design with VHDL

Hanan Ribo

12/04/2021

Table of contents

1. Aim of the Task:

- System design using concurrent and sequential logic principles using advanced Simulation methods (based on material given in LAB1 and LAB2 tasks).
- Controller design based on methodology of Control and Datapath separation.
- Preparation for LAB4 task – FPGA based design synthesis of the given design.
- Proper analysis and understanding of architecture design.

2. Assignment definition:

In this task, you are required to design a controller-based system that operates on serial stream of data (operation on series of numbers). The System reads and operates on $\boldsymbol{n}$ numbers in a row:

Input:

$$\mathbf{0}, \dots, \mathbf{0}, \boldsymbol{opc}, \boldsymbol{n}, \ \boldsymbol{x_1}, \ \dots, \boldsymbol{x_n}, \boldsymbol{opc}, \boldsymbol{m}, \ \boldsymbol{y_1}, \ \dots, \boldsymbol{y_m}, \boldsymbol{opc}, \ \boldsymbol{k}, \ \boldsymbol{z_1}, \ \dots, \boldsymbol{z_k}, \mathbf{0}, \dots, \mathbf{0}$$

$Where\ opc > 0\ ;\ n > 0\ ;\ x_i, y_i, z_i\ are\ in\ 2's\ complement\ (can\ be\ zero).$

Note: the given example contains 3 data sets, without limiting generality.

Output:

$$\coprod_{i=1}^{n} x_i \ , \ \coprod_{i=1}^{m} y_i \ , \coprod_{i=1}^{k} z_i \quad where \ \coprod \ is\ an\ ALU\ operation\ (except\ opcode\ zero).$$

Example (given in Decimal for convenience):

Data input: $\mathbf{0}, \dots, \mathbf{0}, \mathbf{1}, \mathbf{3}, \ \mathbf{10}, \ \mathbf{20}, \mathbf{25}, \mathbf{2}, \mathbf{4}, \ \mathbf{20}, \ \mathbf{30}, \mathbf{55}, \mathbf{25}, , \mathbf{0}, \dots, \mathbf{0}$ → t

Data output: $\mathbf{55}, -\mathbf{20}$ → t

3. Controller based system:

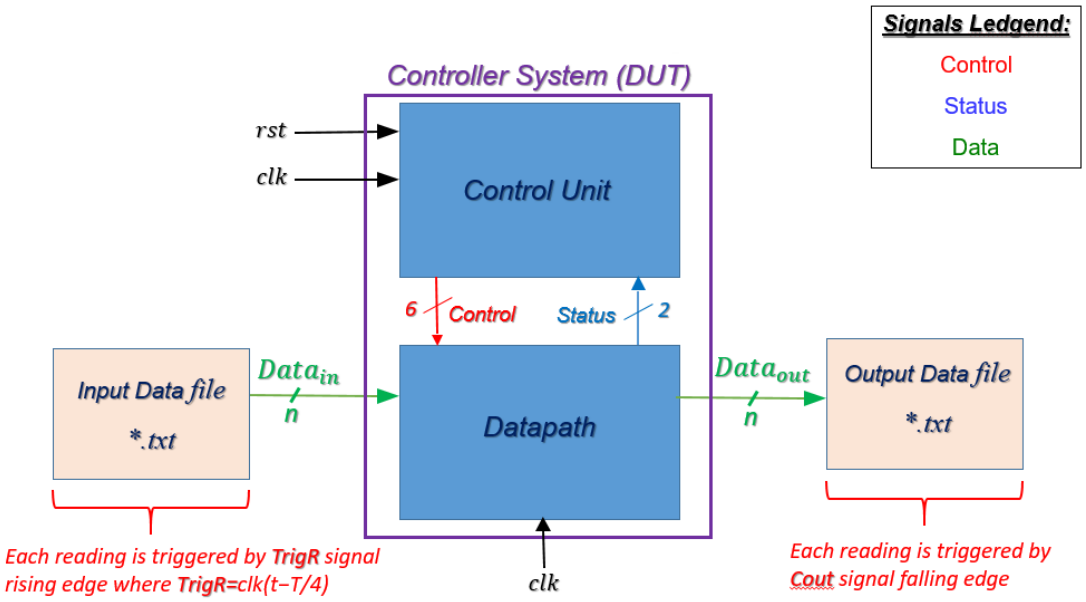


Figure 1: Overall DUT structure

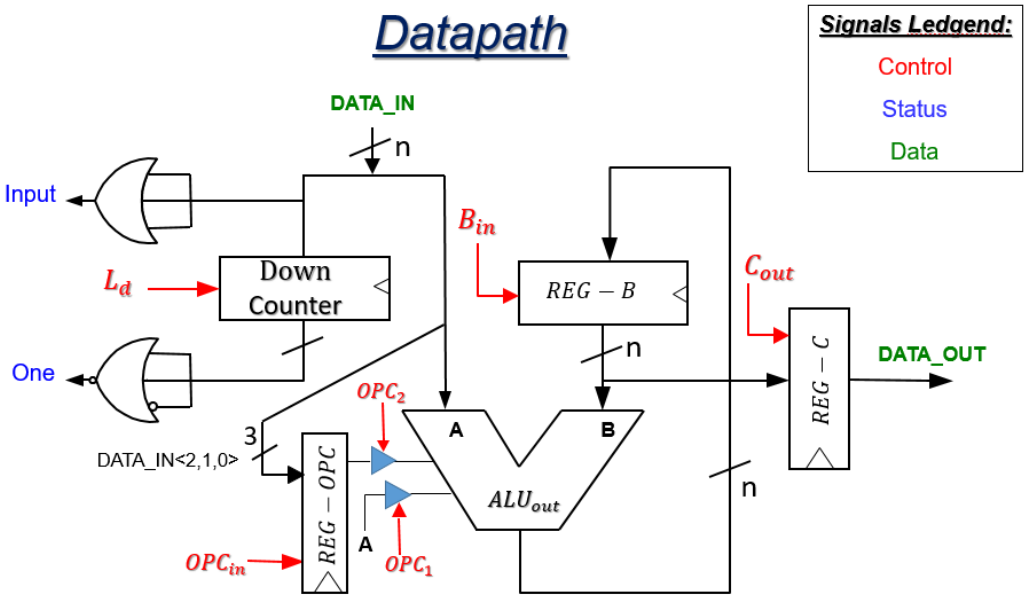


Figure 2: Datapath structure

Decimal value	ALUFN	Operation
0	000	ALUout=A
1	001	ALUout=A+B
2	010	ALUout=A-B
3	011	ALUout=A OR B
4	100	ALUout=A AND B
5	101	ALUout=A XOR B

Table 1 : ALU Opcodes

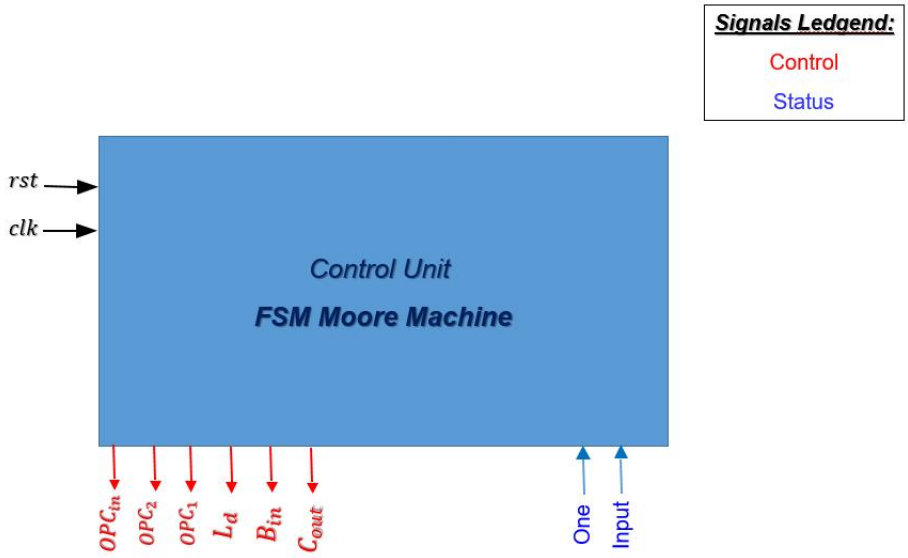
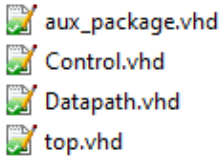


Figure 3: Control unit structure

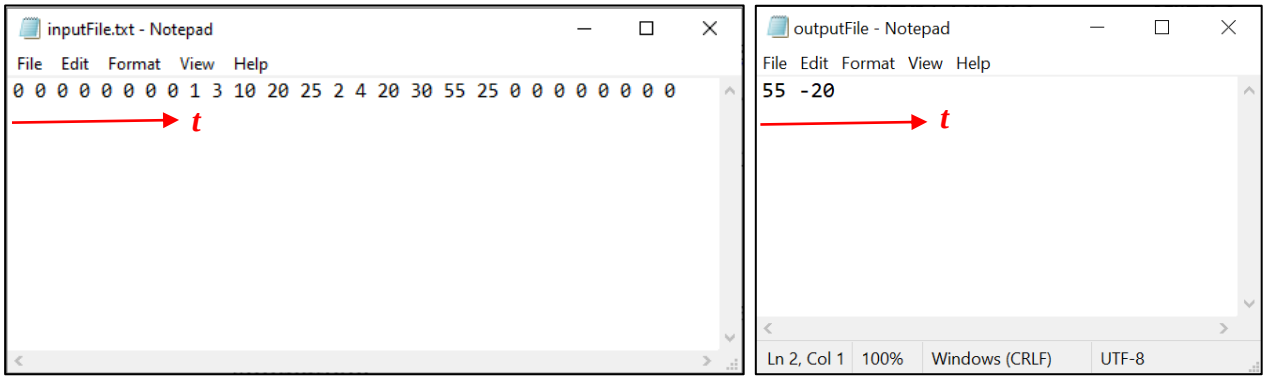
- The given files <u>that you must use in your project</u>:



  - ✓ You can only add code to these files, you are not allowed to erase anything**.**
  - ✓ You can add additional VHDL source code files.
  - ✓ Other entities can be designed and modeled behaviorally, structurally, etc.

4. File based simulation of DUT:

- As depicted in figure 1, in order to simulate the DUT systemly we use file based simulation.
- <u>Input and Output files structure (according the above given example at clause 2)</u>:



5. Requirements:

1. The design must be well commented.
2. 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).**
3. 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 student with id1* (any of these rules violation disqualifies the task submission).

4. The **ZIP** file will contain (***only the exact next sub folders***):

Directory	Contains	Comments
DUT	Project VHDL files	**Only VHDL files of DUT**, 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	One **tb.vhd** for the Datapath One **tb.vhd** for the Control unit One **tb.vhd** for the overall DUT
SIM	ModelSim DO files (wave, list)	One **tb.vhd** for the Datapath One **tb.vhd** for the Control unit One **tb.vhd** for the overall DUT (with Datatpath and Control unit signals included)
DOC	Project documentation	• ***readme.txt*** (list of the DUT **.vhd* files with their brief functional description) • ***pre3.pdf*** (report file that includes brief explanation of the top module with its wave diagrams as shown in figure 4)

Table 2: Directory Structure

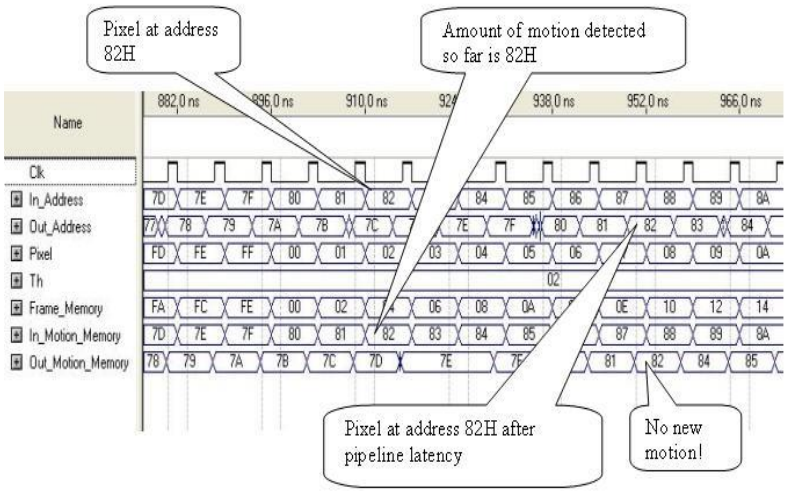


Figure 4: Clouds over the waveform example

6. 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 3 : 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}