CPU Architecture

First Task

ALU Design with VHDL

Hanan Ribo

5/05/2020

Table of contents

List of Figures

List of Tables

1. Aim of Task1:

- System design using concurrent and sequential logic principles using advanced Simulation methods (based on material given in LAB1 and LAB2 pre tasks).
- Preparation for Task2 – FPGA based design synthesis of this given ALU.
- Proper analysis and understanding of architecture design.

2. Assignment definition:

In this task you will design a basic registered ALU. The ALU will have following features (all the next sizes will be used with generic statement as given in tb.vhd):

- Two Input bus named A and B, width of 8-bit each as a default value
- Two Output buses named RES(HI,LO), width of 8-bit each as a default value
- One status bus output of 2-bit and an op-code input of 5-bit

You are required to design the whole system and make a test bench that tests all the system.

3. ALU high-level design:

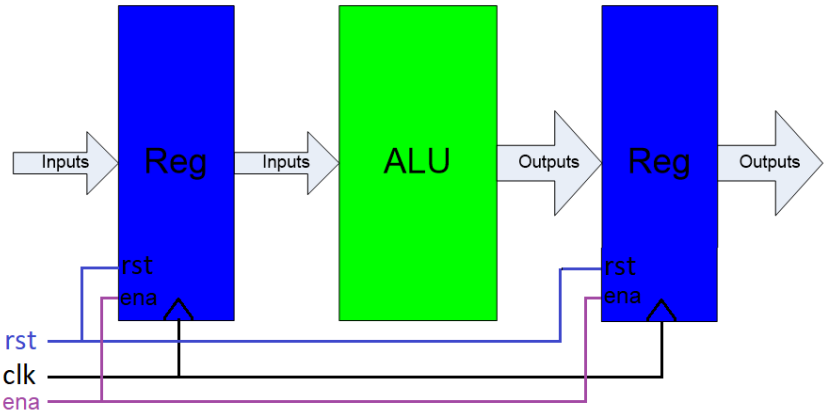


Figure 1: Overall DUT structure

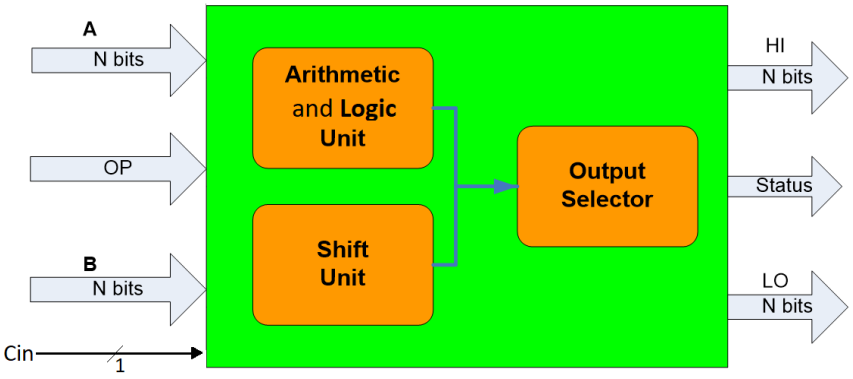


Figure 2: ALU structure

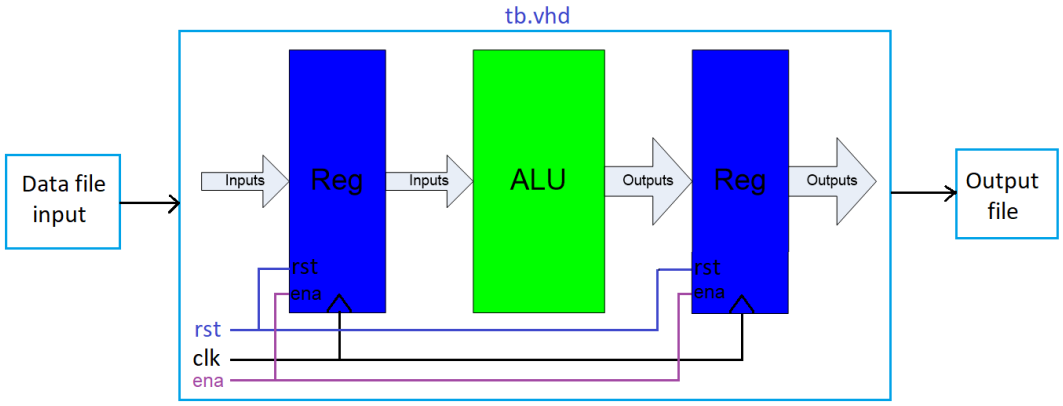


Figure 3: File based simulation of DUT

The following table describes all the available ALU Op codes and operation

Instruction	Opcode	Operation	Note
ADD	00001	RES(HI,LO) = A + B	As in lab1 requirement (see FAQ file)
SUB	00010	RES(HI,LO) = A - B	As in lab1 requirement (see FAQ file)
ADDC	00011	RES(HI,LO) = A + B + Cin	As in lab1 requirement (see FAQ file)
MULT	00100	RES(HI,LO) = A*B	Multiply two integer signed numbers (Result is N*2 bits)
MAC	00101	ACC= ACC+A*B RES(HI,LO) = ACC	Multiply Accumulate (MAC is internal N*2 bits register) integer unsigned numbers
MAC_RST	00110	MAC=0	Reset MAC
MAX	00111	RES(0,LO) = Max (A,B)	maximum between A and B
MIN	01000	RES(0,LO) = Min (A,B)	minimum between A and B
AND	01001	RES(0,LO) = AND (A,B)	Bitwise operation
OR	01010	RES(0,LO) = OR (A,B)	Bitwise operation
XOR	01011	RES(0,LO) = XOR (A,B)	Bitwise operation
RLA	01100	RES(0,LO) = rotation result	Rotate left A arithmetically B(2..0) times
RLC	01101	RES(0,LO) = rotation result	Rotate left A through carry B(2..0) times
RRA	01110	RES(0,LO) = rotation result	Rotate right A arithmetically B(2..0) times
RRC	01111	RES(0,LO) = rotation result	Rotate right A through carry B(2..0) times

Table 1 : ALU Op Codes

- Do **not use** additional arithmetic hardware for MAC operation; you must use the existing ADD/SUB and MULT modules.

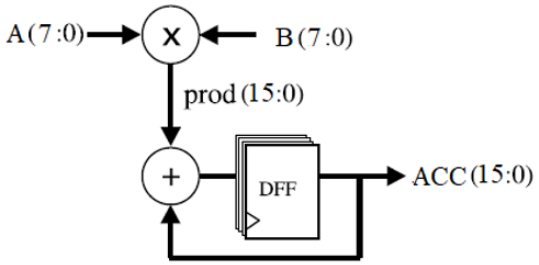


Figure 4: MAC circuit

- The Status Bus of the ALU outputs is depicted in the next table:

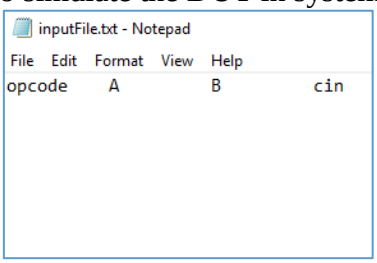
Status Flag Name	**Condition**	**Indicator** $(\text{status}_1, \text{status}_0)_2$
C	Last OP produces carry	01
Z	Last OP result is zero	10
	else	00

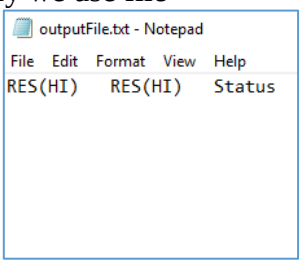
Table 2 : Status Bus

- 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**).**
- The ALU Top Level (without two wrapper registers) design must be Structural and contains the following entities:
  - ✓ Arithmetic and Logic Entity (For MULT, MAC, ADD/SUB, etc.)
  - ✓ Shift Entity (For shifts operations)
  - ✓ Output selector that formulates the output busses and status
- The Add/Sub module must be designed structurally (taken from LAB1).
- Other entities can be designed behaviorally, structurally, etc.

4. File based simulation of DUT:

- As depicted in figure 3, in order to simulate the DUT in systemly we use file based simulation.
- Input and Output files structure:





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 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	Only one **tb.vhd** for the overall DUT
SIM	DO files of wave and list forms	Only for **tb.vhd** of the overall DUT
DOC	Project documentation	**Readme.txt** and **pre2.pdf** report file

Table 2: Directory Structure

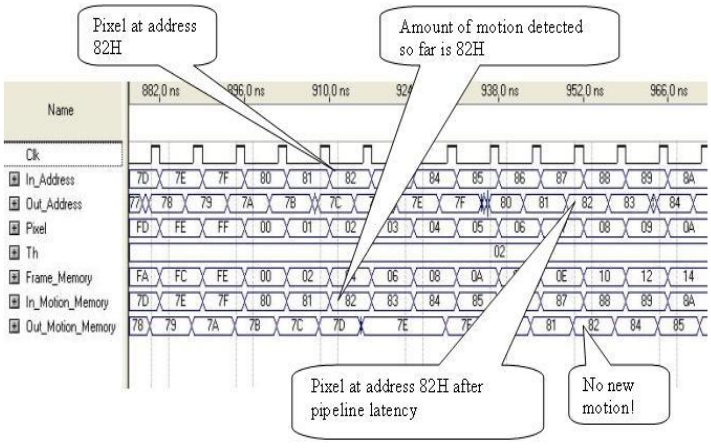


Figure 3: Clouds over the waveform example

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 policy, you will 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.

4. **For a late submission the penalty is 2^{day}**