CPU Architecture

First Laboratory

VHDL and Modelsim

Boris Braginsky, Hanan Ribo and Prof. Hugo Guterman

25/03/2018

Table of contents

List of Figures

List of Tables

1. Aim of the Laboratory

- Obtaining basic skills in VHDL [1] and ModelSim [2].
- General knowledge rehearsal in digital systems [3].
- Proper analysis and understanding of architecture design.

2. Assignment definition

In this laboratory you will design a basic ALU. The ALU will have following features:

- Configurable input bus width between 8 and 32bit (using generic variable N)
- Two input Busses
- Two Output buses
- One status bus output and an op-code input

You will have to design the whole system and make a test bench that tests all the system. Feel free to improve the proposed architecture and the modules.

3. System Design

Figure 1 describes the overall system structure.

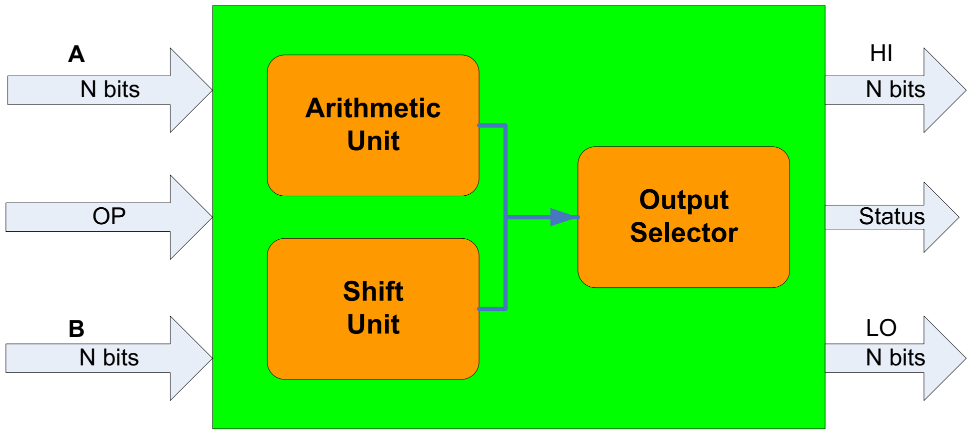

Figure 1 : Overall system operation

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

OP	Operation	Description
MUL	(HI,LO) = A*B	Multiply two signed numbers (Result is N*2 bits)
MAC	MAC= MAC+A*B (HI,LO)=MAC	Multiply Accumulate (MAC is internal N*2 bits register) signed numbers
MAX	LO = Max (A,B)	Return maximum between A and B
MIN	LO = Min (A,B)	Return minimum between A and B
RST	MAC=0	Reset MAC
SHR	LO = A >> B	Shift right
SHL	LO = A << B	Shift left
ADD	LO = A + B	Arithmetic Add
SUB	LO = A - B	Arithmetic Sub

Table 1 : ALU Op Codes

Do **not use** additional arithmetic hardware for MAC operation; utilize the ADD/SUB and MULT hardware instead.

The Status bus of the ALU outputs the comparison status of the ALU in case of SUB operation, do not use comparators for each condition, but utilize the ADD/SUB hardware instead (use carry output of the adder and comparison of output to zero).

The following table describes the status bus signals.

Status Flag Name	Condition
eq	A = B
ne	A != B
ge	A >= B
gt	A > B
le	A <= B
lt	A < B

Table 2 : Status Bus

The Top Level ALU design must be structural and contain the following entities:

- Arithmetic Entity (For MUL, MAC, ADD and SUB operations)
- Shift Entity (For SHL and SHR operations)

- Output selector that formulates the output busses and status

The adder for Add/Sub function must be designed both behaviorally and structurally (using basic gates AND, OR, NOT, XOR, NAND, NOR).

Other entities can be designed behaviorally, structurally or mixed. The synchronous parts of the system will be constructed using Flip-Flops (DFF).

Other entities can be designed behaviorally, structurally or mixed.

4. 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 starters**: 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

5. 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
3. The top level design must be structural, all other modules can be structural/ behavioral or mixed (except add/sub).
4. The behavioral parts of the design (except the test bench) must be synthesizable, pay attention on the logic that you are describing.
5. <u>Block diagrams for the behavioral parts of the design</u> that describes the logic behind the behavioral code.
6. The design must be well commented.
7. If some logic appears more than once in the design it must be used as a separate entity with structural instances
8. Do not use latches.

9. **Important:** For each module :
    - Graphical description (a square with ports going in and out).
    - Port Table (direction, size, functionality).
    - Short descriptions.
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 (Figure 6).
    - Change the waveform colors in Modelsim for clear documentation (Tools->Edit Preferences->Wave Windows).
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 must be submitted to email borisbr@post.bgu.ac.il . The file will contain :
    - The full *.pdf report file.
    - Project files and VHDL files
    - **All the relevant *.do files (wave files)**
    - A *readme.txt* that defines the compilation order.
    - The ZIP must not contain temporary files (Files that are produced by compiler and not required for the compilation)
13. The ZIP file directory structure must be arranged according to table 3.
14. 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

Table 3 : Directory Structure

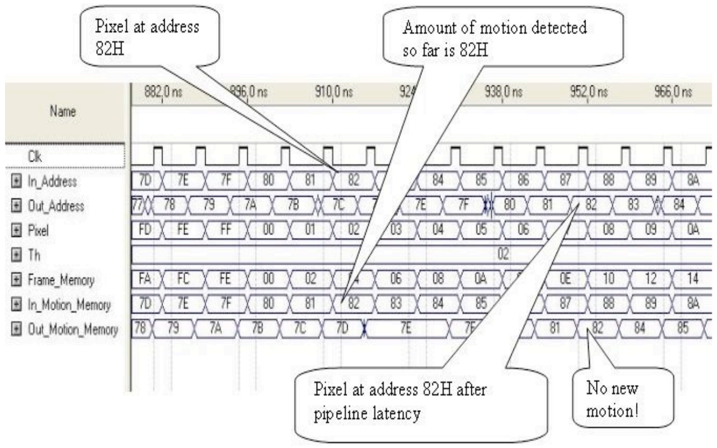

Figure 2 : Clouds on the waveform

6. 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 4 : 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 15/04/2017 using only email: borisbr@post.bgu.ac.il.

Use subject: CPU Work #1.

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

7. References

[1]. P. Ashden, "The designer's guide to VHDL", Morgan-Kaufmann publishing, 1996 or 2002.
[2]. www.model.com – ModelSim Website
[3]. G. Langholz , A. Kandel , C. Mott , "Digital Logic Design" , Brown Pub 1988