Individual design and research project
Stacks in ISAs

1 Objectives

1. To implement a stack data structure in SystemVerilog.
2. To investigate uses of stack structures in modern Instruction Set Architectures (ISAs).

2 Final grade weight

This delivery constitutes 10% of your final grade.

3 Deadline

23:59 hours on Thursday June 4th 2020

4 Teamwork policy

This is an individual assignment.

5 Pre-requisites

It is assumed that you are familiar with working with ModelSim and Quartus. If you require assistance, you can refer to the first assignment tutorial.

6 Stack design specifications

This is an individual design and research assignment. As a result of this, only the minimum details will be provided in this document.

Table 1.1 specifies the top-level ports required for this assignment.

Table 1.1: Top-level ports.

Port name	Direction	Width	Description
`clk`	`input`	1	Clock signal
`asyn_n_rst`	`input`	1	Asynchronous active-low reset
`push`	`input`	1	Active-high write request
`pop`	`input`	1	Active-high read request
`data_in`	`input`	`DATA_WIDTH`	Input data
`data_out`	`output`	`DATA_WIDTH`	Output data
`full`	`output`	1	Indicates stack is full
`empty`	`output`	1	Indicates stack is empty

When `push` in Table 1.1 is asserted, `data_in` should be written into the stack only if the stack is not full. Similarly, when `pop` in Table 1.1 is asserted, `data_out` should be updated with the last input data only if the stack is not empty. Push operations have priority over pop operations.

The depth of your stack must be controlled by the parameter `STACK_DEPTH`.

6.1 SystemVerilog design and testbench files

The required SystemVerilog design files and testbenches are listed below.

- `lifo.sv`. Top-level stack module.
- `tb_lifo.sv`. Top-level testbench.
- `lifo.do`. ModelSim compile and simulation script.

7 Report

You are required to submit a report, which must include the following sections.

1. **Introduction**. An introduction to the report and to the project.
2. **Stack use in Instruction Set Architectures (ISAs)**. This section must answer the following questions.
   (a) What is a stack?
   (b) How does a stack-based ISA work?
      i. What differences can you spot between a stack-based ISA and the ISA used in your Microprocessor without Interlocked Pipeline Stage (MIPS) project?
   (c) Give an example of a modern stack-based ISA.
      i. Name of ISA.
      ii. General ISA characteristics such as, type(s) of addressing modes, type(s) of instruction encoding, which instructions use the stack? etc.

   (d) For which purpose could you use a stack in the ISA used in your MIPS project?

3. **Stack design, simulation and synthesis.** You must include
   (a) A brief explanation of your SystemVerilog design.
   (b) A screenshot of your ModelSim simulation.
   (c) A screenshot of the generated Register-Transfer Level (RTL) in Quartus.
4. **Conclusions.** Concluding remarks on what you've learned in this project.
5. **References.** List of citations in IEEE format used for the report.

NOTE. There is no restriction on the number of pages. However, quality is prioritized over quantity. A lengthy but meaningless report will result in a lower score than a shorter but more meaningful report.

8 Grading criteria

The grade of this project is divided as presented in Table 1.2.

Table 1.2: Grading criteria.

Rubric	Percentage
Stack in SystemVerilog & testbench	50%
Report	50%
TOTAL	100%

In addition to the grading criteria specified in Table 1.2, I will consider the following aspects.

1. **The correct functionality of your designs.** I will use my own testbenches in order to automatically stress your designs and verify that they perform the tasks according to the specifications. For example, I will try different values for the parameters in your designs and I expect them to still perform according to the specifications. This is why it is paramount that you follow the name convention specified for file names and port names. Moreover, it is important that your designs and testbenches compile in ModelSim without warnings and without errors. **Each RTL warning will deduct 5% of your project grade. Your maximum grade for this assignment will automatically drop to 50/100 should ModelSim trigger a compilation or simulation error.**

2. **The quality of your testbenches.** Even though I will use my own testbenches, I expect you to consider a thorough and concious set of test scenarios. In this way, you should be able to spot any mismatches between the expected results and the actual results provided by your designs.

3. **Your designs must be synthesized in Quartus without latches, without RTL warnings and without errors.** **Each RTL warning will deduct 5% of your project grade. Your maximum grade for this assignment will automatically drop to 50/100 should Quartus trigger a synthesis error or generate unwanted latches.**

   The only warnings that will be tolerated are those that are not related to the RTL itself. Examples of such warnings are:

- `Warning (18236): Number of processors has not been specified which may cause overloading on shared machines. Set the global assignment NUM_PARALLEL_PROCESSORS in your QSF to an appropriate value for best performance.`
- `Warning (13046): Tri-state node(s) do not directly drive top-level pin(s).`
- `Warning (292013): Feature LogicLock is only available with a valid subscription license. You can purchase a software subscription to gain full access to this feature.`
- `Warning (15714): Some pins have incomplete I/O assignments. Refer to the I/O Assignment Warnings report for details.`

Remember that a design is not useful if it can't be synthesized.

9 Deliverables and Submission instructions

Prepare a single `zip` file containing all your design, testbench, script and report files as indicated in Section 6.1 and Section 7.

Submit your assignment through Blackboard **no later** than 23:59 hours on Thursday June 4th 2020. Only one submission per team is necessary.
Please send any questions to isaac.perez.andrade@tec.mx.