Switching Circuit and Logic Design 2023

Lab2-Sequential Circuit

Deadline: 2023.12.08 23:59:59

Introduction

In this lab, you are asked to design three sequential circuits, *Sequence Detector*, *Accumulator* and *Weighted Sequence Counter*. You will learn how to use software tool Quartus-II to simulate your designed circuits, verify their functionality through testing patterns, and visualize the digital waveform.

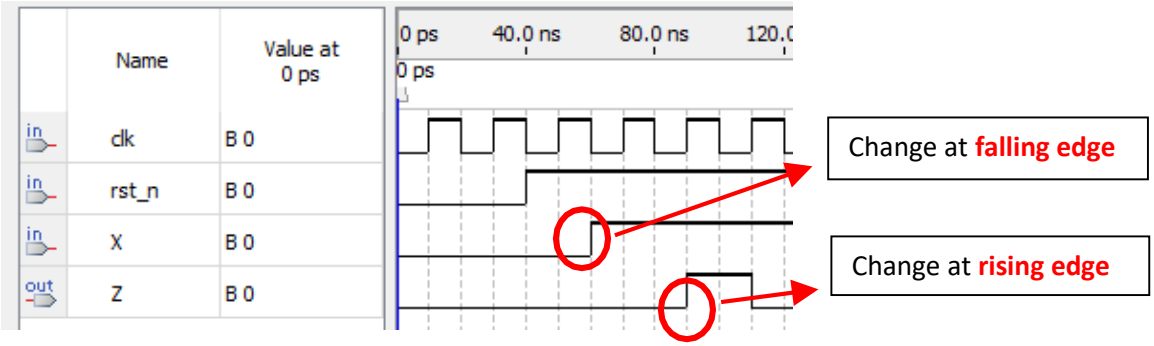
Tasks

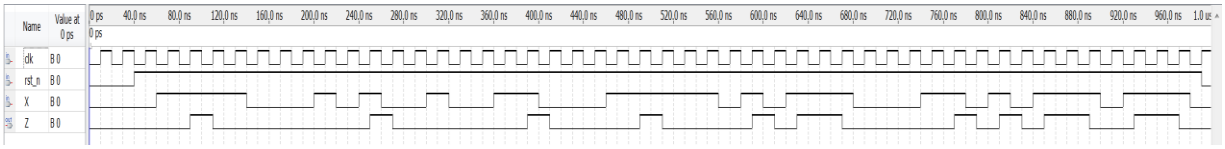
◆ Sequence Detector

Implement a *Sequence Detector* (SD) with binary input ***X***, clock input ***clk***, reset input ***rst_n*** and a binary output ***Z***. This detector should detect **input sequence 011 or 101** from left to right and raise output *Z* to 1 when detected. Otherwise, *Z* remains 0.

X =	1	0	1	1	0	0	1	1	0	1	0
Z =	0	0	**1**	**1**	0	0	0	**1**	0	**1**	0
(time =	0	1	2	3	4	5	6	7	8	9	10)

After reset rises, *X* may or may not change at every **falling edge** of *clk*. While your output *Z* should change at the **rising edges** of *clk*. The tested waveform is shown as below.





A similar design to this Sequence Detector has been shown in the tutorial. However, you should make some modifications since the detected pattern is different.

◆ Accumulator

Implement a 4-bit accumulator with **signed** input ***x_i*[3:0]**, ***load***, ***clk***, ***rst_n***, and output ***done, sum_o*[3:0]**. The inputs and outputs are denoted as:

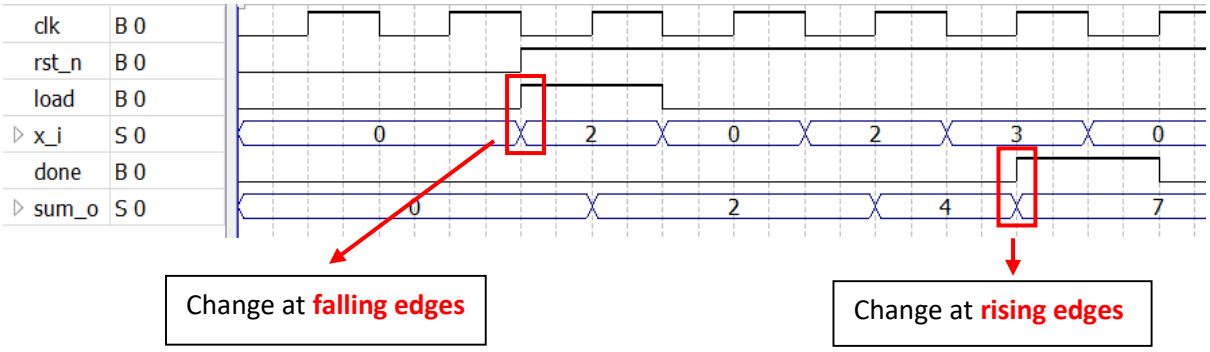
Inputs:

- ☐ *clk*: Clock signal for sequential circuits.
- ☐ *rst_n*: **Active-low** reset signal. (i.e. reset when *rst_n* is 0)
- ☐ *load*: Start to accumulate when *load* is **high**.
- ☐ *x_i*: 4-bit **signed** **(2's complement)** signal to be accumulated.

Outputs:

- ☐ *done*: Set to **high** when the accumulation is finished.
- ☐ *sum_o*: 4-bit **signed** **(2's complement)** accumulated result.

This circuit should add **4 successive input *x_i***. When *load* is 1, first *x_i* will be sent to the circuit and next *x_i* will come at next falling edge of *clk*, and so on. After receiving 4 successive inputs, set *done* to 1 and output ***sum_o*** which takes value of **x_i1+x_i2+x_i3+x_i4**. **We will only check your *sum_o* when *done* is 1**. If *done* is 0, *sum_o* can be any value you want. For simplicity, we have limited our input *x_i* so that **No Overflow is occurred during accumulation**. (i.e. -8 <= *sum_o* <= 7). The tested waveform is shown below.



An unsigned accumulator design is shown in the tutorial. For this task, you should consider the characteristics of 2's complement.

◆ Weighted Sequence Counter

Implement a 4-bit weighted sequence counter (WSC) with **unsigned** input sequence ***X***, ***load***, ***clk***, ***rst_n.***
signed output ***score*[3:0], *done***.
When the weighted sequence counter detected sequence **011**, score **decreased** by **1**;
When it detected sequence **101**, score **increased** by **3**. For simplicity, we have also limited our input sequence pattern so that **No Overflow will be occurred**. (i.e. -8 <= *score* <= 7).
Done signal can be pull to high one-clock-period right after the load signal is assigned 0. We will check done signal by the negative edge of clock. Once the done signal is pull to high, please reset *score* before next load signal pull to high.
The inputs and outputs are denoted as:

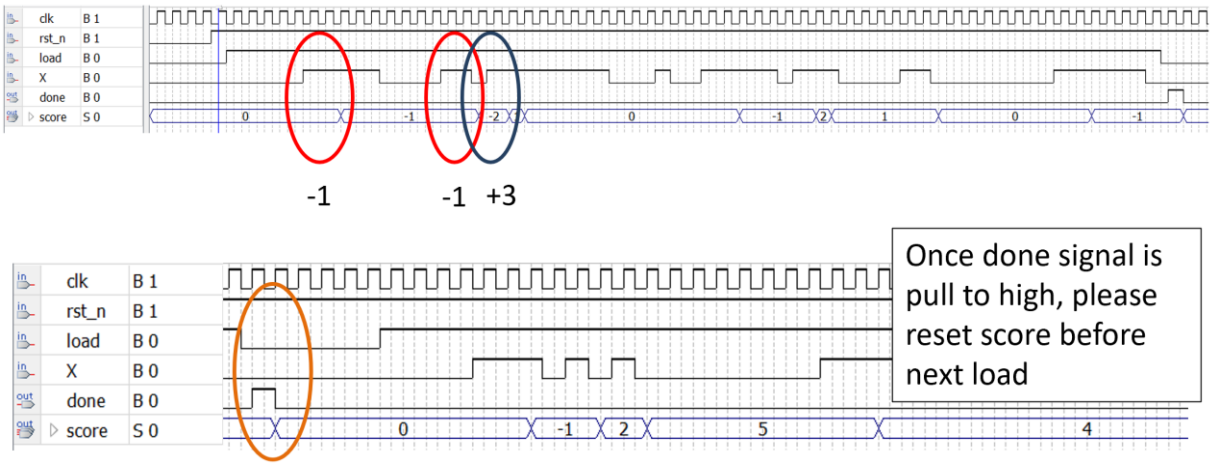
Inputs:

- ☐ *clk*: Clock signal for sequential circuits.
- ☐ *rst_n*: **Active-low** reset signal. (i.e. reset when *rst_n* is 0)
- ☐ *load*: Weighted Sequence Counter is performed when *load* is **high**.
- ☐ *X*: Sequence to be detected

Outputs:

- ☐ *done*: Set to **high** when the accumulation is finished (which load = 0).
- ☐ *score*: 4-bit **signed (2's complement)** integer in counting weighted sequence

When *load* is 1, *X* will be sent to the circuit. As your computation is done, set *done* to 1 and output *score*. **We will only check your *score* when *done* is 1, and please make sure they are aligned together**. The tested waveform is shown below.



Report

◆ Name your report as Your_Student_ID_report.pdf (e.g. b12901xxx_report.pdf)

◆ In your report, there must include:

☐ All the figures of your designs, including sub-circuits

e.g. Sequence Detector → SD.jpg
4-bit Accumulator → AC4.jpg, FSM.jpg
4-bit Weight Sequence Counter → WSC.jpg, FSM.jpg
Note: If you have used more sub-circuits, please paste them all.

☐ Functionality results for each task. i.e., the waveforms

◆ You can export your designs including all sub-circuits to .jpg file or take a screenshot.

Requirements

◆ Pass the test patterns provided by TAs for correct functionality.

◆You are restricted to use the same I/O pin name as shown above. Any violation of this rule will lead to some penalty to your score. (your total score *0.8).

e.g. Sequence Detector → Inputs: *clk*, *rst_n*, *X*
Output: *Z*
4-bit Accumulator → Inputs: *clk*, *rst_n*, *load*, *x_i*[3:0]
Output: *done*, *sum_o*[3:0]
4-bit Weighted Sequence Counter → Inputs: *clk*, *rst_n*, *load*, *X*
Output: *done*, *score*[3:0]
Note: Differences between uppercase and lowercase does matter.

◆ You can **only** use the logic gates provided by TAs to design your circuits.
(i.e. Only the logic gates in NTUEE_LogicDesign_Lib/elements folder.)

◆ **For each task, export your design including all sub modules to Verilog (.v) file.**
e.g. if you have AC4.bdf and FSM.bdf in your folder, you should put AC4.v and FSM.v in this folder too.

Submission

- Submit your .zip file to NTU COOL by the deadline (2023.12.08).

◆ The project names of 3 tasks, please name as following:

☐ Sequence Detector → project name: SD
Contain: SD.qpf, SD.bdf, SD.v and all the other sub-circuit files. (such as FSM.bdf, FSM.bsf, FSM.v)

☐ 4-bit Accumulator → project name: AC4
Contain: AC4.qpf, AC4.bdf, AC4.v and all the other sub-circuit files. (such as FSM.bdf, FSM.bsf, FSM.v, FA4.v, FA4.bdf, FA4.bsf)

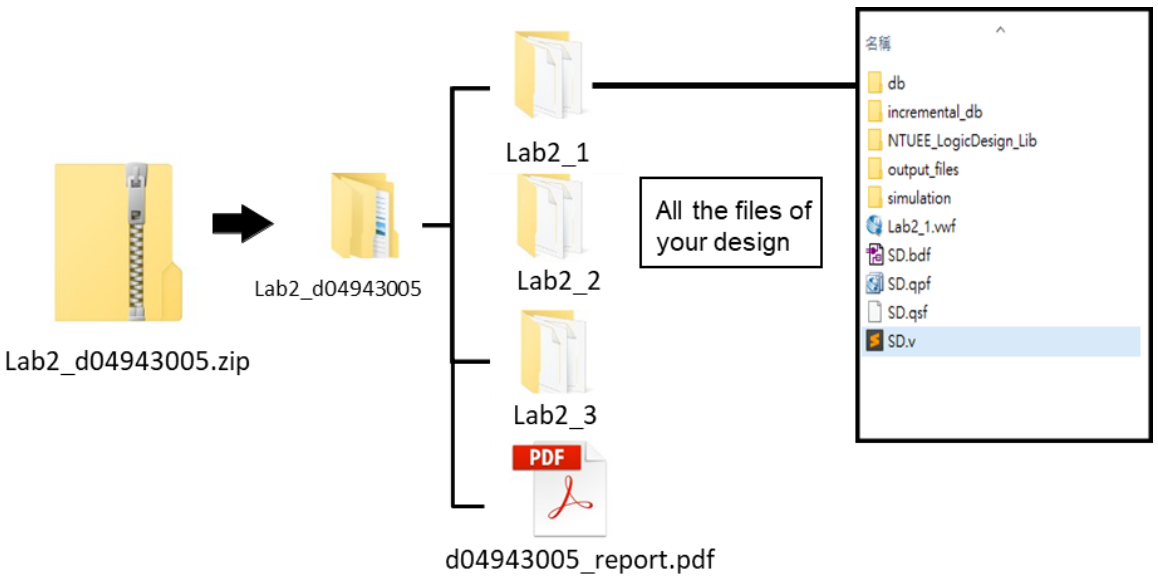
☐ Weighted Sequence Counter → project name: WSC
Contain: WSC.qpf, WSC.bdf, WSC.v and all the other sub module file (like FA4.bdf, FA4.bsf, FA4.v, FSM.v, FSM.bdf, FSM.bsf)

Note: For top module, we restricted you to use the same name as we did. Any violation of this naming rule will lead to some penalty to your score. (your total score *0.8). But you can use any name you want for the sub module (like FSM.bdf, Controller.bdf), just make sure you use the same name for all .bdf, .bsf and .v files.

◆ Please archive all the files and name as Lab2_StudentID.zip (e.g. Lab2_ b12901xxx.zip)

◆ Your directory should be like following and put all the related files in the same folder for each task.

◆Please follow the naming rule of folders and hierarchy indicated below, any violation of this rule will lead to some penalty to your score. (your total score *0.8)



Grading

- Sequence Detector (40%), 4-bit Accumulator (40%), and 4-bit Weighted Sequence Counter (20%).
- Late submission rule:
  - 20% off per day late.
  - e.g. submit on 12/09, total score*0.9, on 12/10 total score *0.8, on 12/11 total score *0.6 and so on.

Reminder

- Be sure you have uploaded all the required files.
- Be sure to match all the pin names, folder names and (.qpf, .bdf, .bsf, .v) file names.
- **DO NOT COPY OTHERS WORK!!! WE WILL CHECK YOUR DESIGN CAREFULLY. THOSE WHO HAVE IDENTICAL DESIGN WILL GET 0 POINT AND BE REPORTED TO PREFESSORS.**