

Tutorial 1: Simulation with Modelsim-Altera

1 Introduction

1.1 Goals

- Practice on Combinational and Sequential circuits.
- Design and implement simple circuits using Quartus/ModelSim software.
- Simulate the implemented circuits on ModelSim by writing testbench.

1.2 Preparation

- Each group (**Maximum 4 members**) brings **at least one laptop** with Quartus/ModelSim installed.
- Revise Chapter 2 and 3.
- Read the provided materials for using modelsim and examples of simple testbench.

1.3 Contents

- Design and implement combinational/sequential circuits on Quartus/ModelSim.
- Simulate the operation of circuits (all cases).

1.4 Report Requirements *

- Compress all source (.v) files and for each exercise, capture 2 pictures of Wave results.
- Submit on Bkel system.
- Deadline: 2 Week.

For Modelsim instruction, please read the attached materials (Using_Modelsim.pdf)

2 Exercise

Write testbench for each exercise, the test bench should present all cases.

Exercise 1: (5 points) Design the **majority** circuit using:

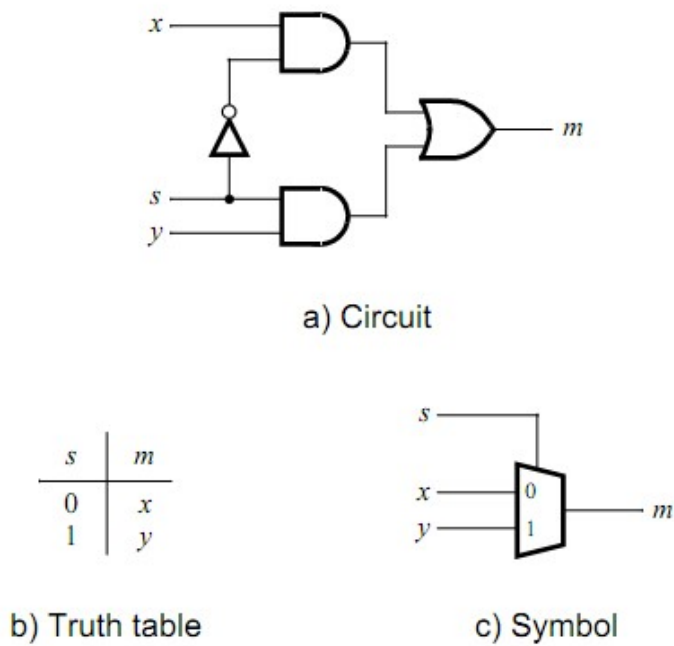
- Structural model.
- RTL model.
- Behavioral model.

Write one testbench file to simulate/evaluate all 3 models (Using the same input signals). The testbench should use "\$display" or "\$monitor" command and presents all possible cases.

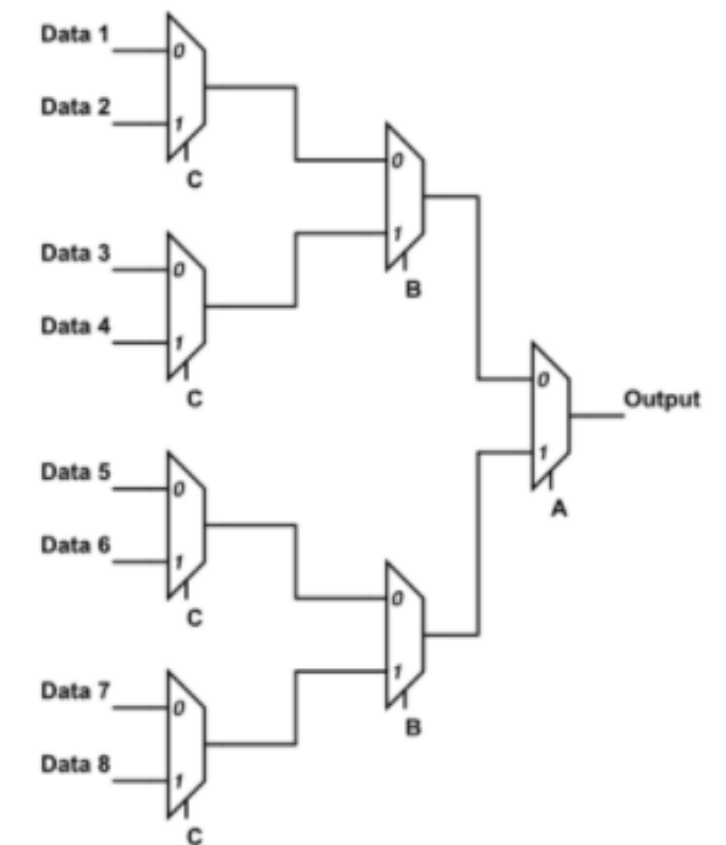
Exercise 2: (5 points) Design a Multiplexer as following:

- Input: 12 signals:
 - 3 Select signals S_{0-2} .
 - 1 enable signal E (active low).
 - 8 Data signals I_{0-7}
- Output: 2 signals Z và \bar{Z} . When enable is active ($E=0$), the circuit run as a multiplexer. Otherwise, if enable signal equal to 1 ($E=1$), Z equal to 0.

Use RTL and structural model to implement 8-to-1 MUX. Firstly, you should make 2-to-1 MUX (Figure 1) using RTL model. Then connect 2-to-1 MUXs together into 8-to-1 MUX (Figure 2).



Hình 1: 2-to-1 MUX



Hình 2: 8-to-1 MUX from 2-to-1 MUXs

Write testbench to simulate your MUX.

Exercise 3: (optional) Implement and write testbench for "string compare" exercise.

By Implement and simulation sub-module:

- Adder
- Comparator
- Multiplexer
- Counter

(Review file 02_ Exercises.pptx)