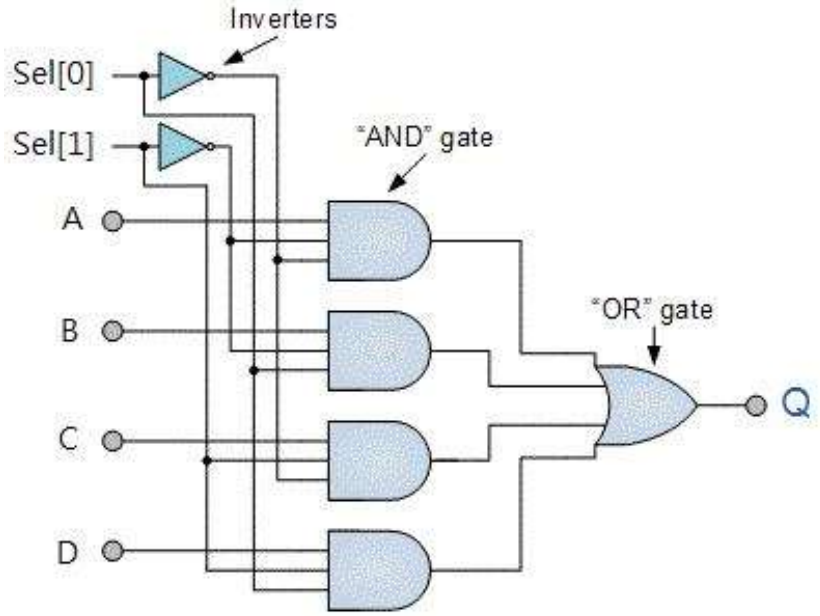
2018 Digital IC Design

Homework 1: Multiplexer

1. Introduction

Multiplexer is a device that can connect multiple output devices to one input device at the same time, that is, hardware that allows more than two signals (data or voice) to be transmitted on the same line (such as telephone line, microwave communication, TV broadcast). If it has two to the power of n input lines, there will be n select lines, and the bit combination of the select lines determines which one is selected. For this question, please make a 4:1 Multiplexer with gate-level design, as shown below.



2. Design Specifications

2.1 Block Overview

2.2 I/O Interface

Signal Name	I/O	width	Description
Sel	**Input**	**2**	**Selection line**
A	**Input**	**1**	
B	**Input**	**1**	
C	**Input**	**1**	
D	**Input**	**1**	
Q	**Output**	**1**	

2.3 File Description

File Name	Description
mux.v	RTL code for using Verilog
mux_tb.v	Test bench for verifying design
cycloneii_atoms.v	Simulation library for gate-level simulation

3. Scoring

3.1 Functional Simulation (pre-sim) **[70 %]**

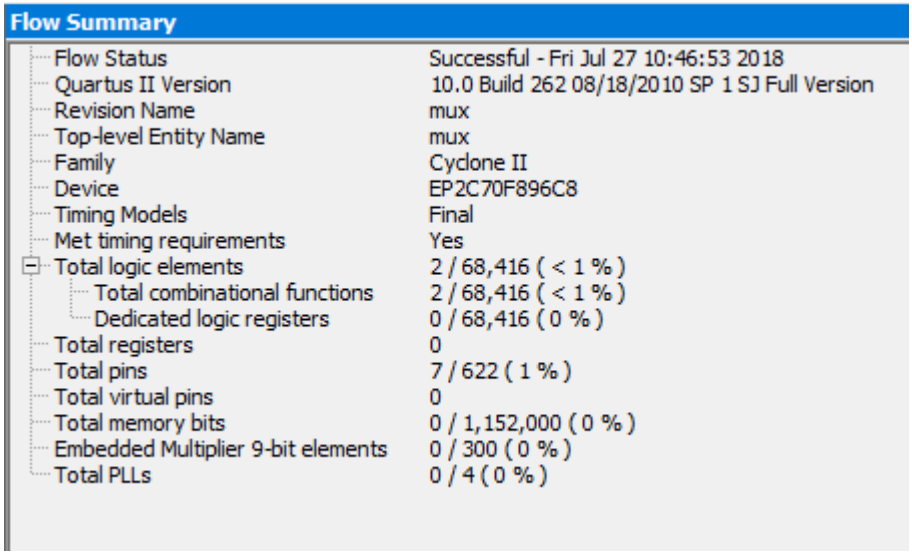
All of the result should be generated correctly, and you will get the following message in ModleSim simulation.

```
#          56 data is correct
#          57 data is correct
#          58 data is correct
#          59 data is correct
#          60 data is correct
#          61 data is correct
#          62 data is correct
#          63 data is correct
#          64 data is correct
# --------------------PASS--------------------
# All data have been generated successfully!
# ** Note: $stop    : C:/Users/DIC/Desktop/2018_DIC/
#    Time: 3300 ns  Iteration: 0  Instance: /mux_tb
```

3.2 Gate-Level Simulation (post-sim) **[30 %]**

3.2.1 Synthesis

Your code should be synthesizable. After synthesizing in Quartus, a file named *mux.vo* will be obtained.



Device : Cyclone II EP2C70F896C8

3.2.2 Simulation

All of the result should be generated correctly using *mux.vo*, and you will get the following message in ModleSim simulation.

```
#          59 data is correct
#          60 data is correct
#          61 data is correct
#          62 data is correct
#          63 data is correct
#          64 data is correct
# --------------------PASS--------------------
# All data have been generated successfully!
# ** Note: $stop    : C:/Users/DIC/Desktop/2018_DIC/201
#    Time: 1380 ns  Iteration: 0  Instance: /mux_tb
```

3.3 Performance **[0 %]**

The performance is scored by the logic elements you used and the simulation time in post-sim. The scoring equation is *(Total logic elements + total memory bit+ 9*embedded multiplier 9-bit element) × (longest gate-level simulation time in ns)*. (The smaller the better).

3.4 Note

You can modify your file mux_tb's clock cycle in post-sim

```
`timescale 10ns / 1ps
`define CYCLE 5
module mux_tb;

reg A, B, C, D;
reg [1:0] Sel;
```

4. Submission

4.1 Submitted files

You should classified your files into three directories and compressed to .zip format. The naming rule is **HW1_*studentID_name_version*.zip**. The vision is v1 for the first submission, and v2, v3… for the revisions.

	RTL category
*.v	All of your verilog RTL code
	Gate-Level category
*.vo	Gate-Level netlist generated by Quartus
	Documentary category
*.pdf	The report file of your design (in pdf).

4.2 Report file

Please follow the spec of report. You are asked to describe how the circuit is designed as detailed as possible, and the flow summary result and minimum CYCLE in post-sim are necessary.

4.3 Please submit your .zip file to folder HW1 in the ftp site.

Deadline: 2018-11-02 23:55

上傳至FTP class_IC/HW1中
IP:140.116.245.92
User: ic_design
Password: ic_design

5. If you have any problem, please contact by the TA by email :

weiting84610@gmail.com 陳威廷
davidshih0908@gmail.com 施信宇