Middle East Technical University - Department of Computer Engineering

CENG 232

Logic Design

Spring '2021-2022

Lab 3

Part 1 Due Date: 13 May 2022, Friday, 23:55
Part 2 Due Date: 20 May 2022, Friday, 23:55
No late submissions

1 Introduction

This assignment aims to make you familiar with Verilog language and the related software tools. There are two parts in this assignment. The first part is a Verilog simulation of an imaginary flip-flop design. The second part consists of the implementation of a Lab Entrance System.

2 Part 1: Warm-up (50 pts)

You are given a specification of a new type of flip-flop, and a new chip that uses this flip-flop. This is an individual part. Your task is to implement these in Verilog, and prove that they work according to their specifications by using testbenches.

2.1 AB Flip-Flop Module

Implement the following AB flip-flop in Verilog with respect to the provided truth table. An AB flip-flop has 4 operations when inputs A and B are: **00 (no change), 01 (set to 1), 10 (set to 0), 11 (complement)**.
Please note that the AB flip-flop changes its state **only at rising clock edges**. Initially, the flip flop should be set to one.

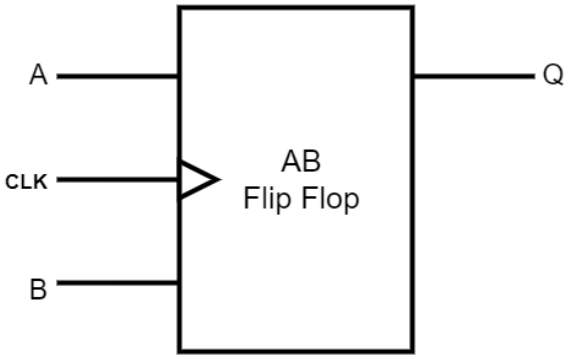

Figure 1: AB Flip-flop diagram

A	B	Q	Q_{next}
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Table 1: AB Flip-flop truth table

2.2 ic1337 Module

Implement ic1337 chip given in Figure 1 that contains two AB flip-flops and has I_0, I_1, I_2, and clk as inputs; Q_0, Q_1 and Z as outputs. Please note that I_1' and I_2' are the complements of I_1 and I_2, respectively.
Use the following module definitions for the modules:

```
module ab(input A, input B, input clk, output reg Q)
module ic1337(input I0, input I1, input I2, input clk, output Q0, output Q1, output Z)
```

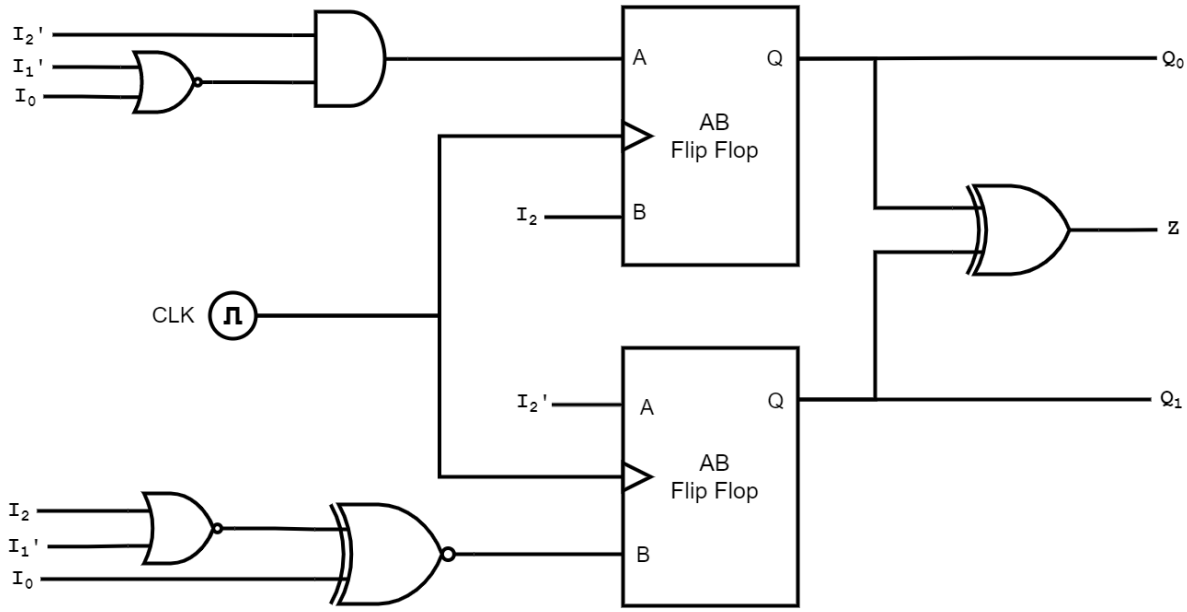

Figure 2: ic1337 module, inputs and outputs.

2.3 Simulation

A sample testbench for AB flip-flop module will be provided to you. You may want to extend the testbench, and also write a testbench for ic1337 module to test different scenarios.

2.4 Deliverables

- Implement both modules in a single Verilog file. Upload only **lab3_1.v** file to ODTUClass system. Do **NOT** submit your testbenches. You can share your testbenches on ODTUClass discussion page.
- Submit the file through the ODTUClass system before the deadline given at the top.
- This is an individual work, any kind of cheating is not allowed.

3 Part 2: COVID-19 Related Lab Entrance System (50 pts)

As the COVID-19 pandemic continues, our department assists to the students and undertakes the work related to safety and health to prevent an outbreak of the pandemic inside the department. To ensure the norms of physical distancing, the department considers reducing the lab size and distributing the students in multiple laboratories to maintain physical distancing during studying.

You will implement a system that can be summarized as follows:
There are 2 computer labs, namely Digital and Mera. Students enter/exit the labs using their smart cards. The system uses 5 bits of the smart cards to allow/disallow students to enter the labs. The same 5-bit code may exist in more than one smart card. Your aim is to design the system summarized above having the following specifications:

1. The labs have the capacity of 30 computers each.
2. Initially, both labs will be empty.
3. If a lab is full, the system will give warning (*isFullDigital*, *isFullMera*).
4. If a lab is empty, the system will give warning (*isEmptyDigital*, *isEmptyMera*).
5. Students can either enter or exit the labs with their smart cards.
6. There will be three modes in the system, which are all synchronous and triggered by **positive edge** of the clock:
   - Entry Mode (01): A student outside a lab wants to **enter** that lab by scanning his/her smart card.
   - Exit Mode (00): A student inside a lab wants to **exit** that lab by scanning his/her smart card.
   - Idle Mode (10,11): No entry/exit operation. Only clear warning messages or lock the door, when necessary.
7. The current number of students in any lab is recorded. In case a student enters/exits a lab, the number of students in that lab should be updated (*numOfStuInDigital*, *numOfStuInMera*).
8. If a student is allowed to **enter** or **exit** a lab, the door of that lab should be unlocked (*unlockDigital, unlockMera*). The unlocked door should be **locked again in the next clock cycle** unless another successful enter/exit operation is performed on that lab.
9. The entrance of students ($mode = 01$) to the labs are arranged is the following order:
   (a) If the lab that the student wants to enter is full, then student cannot enter that lab. The system should **not** unlock the door of that lab.
   (b) If the lab that the student wants to enter has less than 15 students in it, then the student can enter that lab whatever his/her 5-bit smart card code is. The system unlocks the door of that lab (*check item 8*).
   (c) If the lab that the student wants to enter has more than or equal to 15 students in it and it is not full, then check student's 5-bit smart card code:

i. If the smart card code has even number of 1's, then the student cannot enter Digital lab (*therefore do not unlock the door*). If the lab to be entered is Digital lab, the system will give a warning (*restrictionWarnDigital*). The warning **should be cleared in the next clock cycle** unless a student with even number of 1's wants to enter Digital lab again.

ii. If the smart card code has odd number of 1's, then the student cannot enter Mera lab (*therefore do not unlock the door*). If the lab to be entered is Mera lab, the system will give a warning (*restrictionWarnMera*). The warning **should be cleared in the next clock cycle** unless a student with odd number of 1's wants to enter Mera lab again.

iii. If items *i* and *ii* are passed, then the student can enter that lab. The system unlocks the door of that lab (*check item 8*).

10. The student smart card code will not be checked whenever a student wants to leave a lab. When a student in a lab uses his/her smart code to exit a lab, the system should unlock the door of that lab(*check item 8*).

11. It is not possible to leave an empty lab.

12. The 5-bit smart card code **will not be saved** when the student enters or exits the lab.

3.1 Sample Input/Output

				Current State												Next State										
Line No	**smartCode**	**mode**	**lab**	**restrictionWarnMera**	**isFullMera**	**isEmptyMera**	**unlockMera**	**restrictionWarnDigital**	**isFullDigital**	**isEmptyDigital**	**unlockDigital**	**numOfStuInMera**	**numOfStuInDigital**	**clk**	**restrictionWarnMera**	**isFullMera**	**isEmptyMera**	**unlockMera**	**restrictionWarnDigital**	**isFullDigital**	**isEmptyDigital**	**unlockDigital**	**numOfStuInMera**	**numOfStuInDigital**	**Explanation**
1															0	0	1	0	0	0	1	0	0	0	Initial state
2	00000	01	0	0	0	1	0	0	0	1	0	0	0	↑	0	0	1	0	0	0	0	1	0	1	A student (00000) wants to enter Digital lab. Unlock Digital. (item 9.b)
3	11001	01	1	0	0	1	0	0	0	0	1	0	1	↑	0	0	0	1	0	0	0	0	1	1	A student (11001) wants to enter Mera. Unlock Mera. (item 9.b) Digital is locked again because 1 clock passed and there is no other enter/leave operation on Digital.
4	11011	01	1	0	0	0	1	0	0	0	0	1	1	↑	0	0	0	1	0	0	0	0	2	1	Another student (11011) wants to enter Mera. (item 9.b)
5	X	00	1	0	0	0	1	0	0	0	0	2	1	↑	0	0	0	1	0	0	0	0	1	1	A student wants to leave Mera.
6	X	00	0	0	0	0	1	0	0	0	0	1	1	↑	0	0	0	0	0	0	1	1	1	0	A student wants to leave Digital. Unlock Digital. Mera is locked because 1 clock passed and there is no other enter/leave operation on Mera.
7	X	10	X	0	0	0	0	0	0	1	1	1	0	↑	0	0	0	0	0	0	1	0	1	0	System is in idle state. Digital is locked because 1 clock passed and there is no other enter/leave operation on Digital.
8	...																								
9															0	0	0	0	0	0	0	0	1	29	After some time, now there are 29 students in Digital Lab.
10	11000	01	0	0	0	0	0	0	0	0	0	1	29	↑	0	0	0	0	1	0	0	0	1	29	A student (11000) wants to enter Digital. Since the smart card code has even number of 1's, student is not allowed to enter. Show restriction warning. (item 9.c.i)
11	11110	01	0	0	0	0	0	1	0	0	0	1	29	↑	0	0	0	0	1	0	0	0	1	29	Another student (11110) wants to enter Digital. Since the smart card code has even number of 1's, student is not allowed to enter. Restriction warning is still active. (item 9.c.i)
12	11111	01	0	0	0	0	0	1	0	0	0	1	29	↑	0	0	0	0	0	1	0	1	1	30	A student (11111) wants to enter Digital. Since the smart card code has odd number of 1's, student can enter the lab. (item 9.c.iii) Remove restriction warning. Unlock Digital. Digital is full now.
13	11110	01	0	0	0	0	0	0	1	0	1	1	30	↑	0	0	0	0	0	1	0	0	1	30	A student (11110) wants to enter the full Digital lab. Since the lab is full, do not unlock the door. (item 9.a) There is no need to check items 9.b and 9.c Digital is locked again because 1 clock passed and there is no other enter/leave operation on Digital.

In **clk** column of table above, "↑" represents the rising edge of the clock.

3.2 Input/Output Specifications

Name	Type	Size
smartCode	Input	5 bits
Clock (CLK)	Input	1 bit
lab	Input	1 bit
mode	Input	2 bits
numOfStuInDigital	Output	6 bits
numOfStuInMera	Output	6 bits
restrictionWarnMera	Output	1 bit
isFullMera	Output	1 bit
isEmptyMera	Output	1 bit
unlockMera	Output	1 bit
restrictionWarnDigital	Output	1 bit
isFullDigital	Output	1 bit
isEmptyDigital	Output	1 bit
unlockDigital	Output	1 bit

- **smartCode** represents 5 bits of the smart card.
- **CLK** is the clock input for the module.
- **lab** is used for the selection of the lab.
  - $lab = 0 \Rightarrow$ Digital
  - $lab = 1 \Rightarrow$ Mera
- **mode** is used to indicate whether the system is idle or the student enters or leaves the lab.
  - $mode = 00 \Rightarrow$ exit
  - $mode = 01 \Rightarrow$ enter
  - $mode =$ otherwise $\Rightarrow$ idle.
- **numOfStuInDigital**, **numOfStuInMera** shows the number of students in Digital and Mera labs, respectively.
- **restrictionWarnDigital** When Digital lab is not full and there are more than or equal to 15 participants in this lab, this warning shows whether the 5-bit smart code of a student has even number of 1's when he/she wants to enter Digital lab.
  - $restrictionWarnDigital = 1 \Rightarrow$ Show the warning, the smart code has even number of 1's.
  - $restrictionWarnDigital = 0 \Rightarrow$ Do not show the warning, the smart code has odd number of 1's.
- **restrictionWarnMera** When Mera lab is not full and there are more than or equal to 15 participants in this lab, this warning shows whether the 5-bit smart code of a student has odd number of 1's when he/she wants to enter Mera lab.
  - $restrictionWarnMera = 1 \Rightarrow$ Show the warning, the smart code has odd number of 1's.
  - $restrictionWarnMera = 0 \Rightarrow$ Do not show the warning, the smart code has even number of 1's.
- **isFullDigital**, **isFullMera** shows whether Digital/Mera is full.
  - $isFullDigital/isFullMera = 1 \Rightarrow$ Digital/Mera is full.
  - $isFullDigital/isFullMera = 0 \Rightarrow$ Digital/Mera is not full.
- **isEmptyDigital**, **isEmptyMera** shows whether Digital/Mera is empty.
  - isEmptyDigital/isEmptyMera $= 1 \Rightarrow$ Digital/Mera is empty.

- isEmptyDigital/isEmptyMera = 0 ⇒ Digital/Mera is not empty.

- **unlockMera**, **unlockDigital** shows whether the door of Digital/Mera is unlocked.
  - unlockMera/unlockDigital = 1 ⇒ The door is open (unlocked), so that a student can enter/leave the lab.
  - unlockMera/unlockDigital = 0 ⇒ The door is closed (locked), so that the student cannot enter/leave the lab.

3.3 FPGA Implementation

You will be provided with a Board232.v file (and a ready-to-use Xilinx project), which will bind inputs and outputs of the FPGA board with your Verilog module. You are required to test your Verilog module on the FPGA boards.

Name	FPGA Board	Description	
smartCode	SW[7..3]	(A)	5 left-most switches
Clock(CLK)	BNT0	(B)	Right-most button
lab	SW[0]	(C)	Right-most switch
mode	SW[2:1]	(D)	The 2 switches to the left of (C)
numOfStuInDigital	AN[1,0]	(E)	2 right-most 7-seg disp
numOfStuInMera	AN[3,2]	(F)	2 left-most 7-seg disp
restrictionWarnMera	LD[7]	(G)	Leds
isFullMera	LD[6]	(H)	
isEmptyMera	LD[5]	(I)	
unlockMera	LD[4]	(J)	
restrictionWarnDigital	LD[3]	(K)	
isFullDigital	LD[2]	(L)	
isEmptyDigital	LD[1]	(M)	
unlockDigital	LD[0]	(N)	

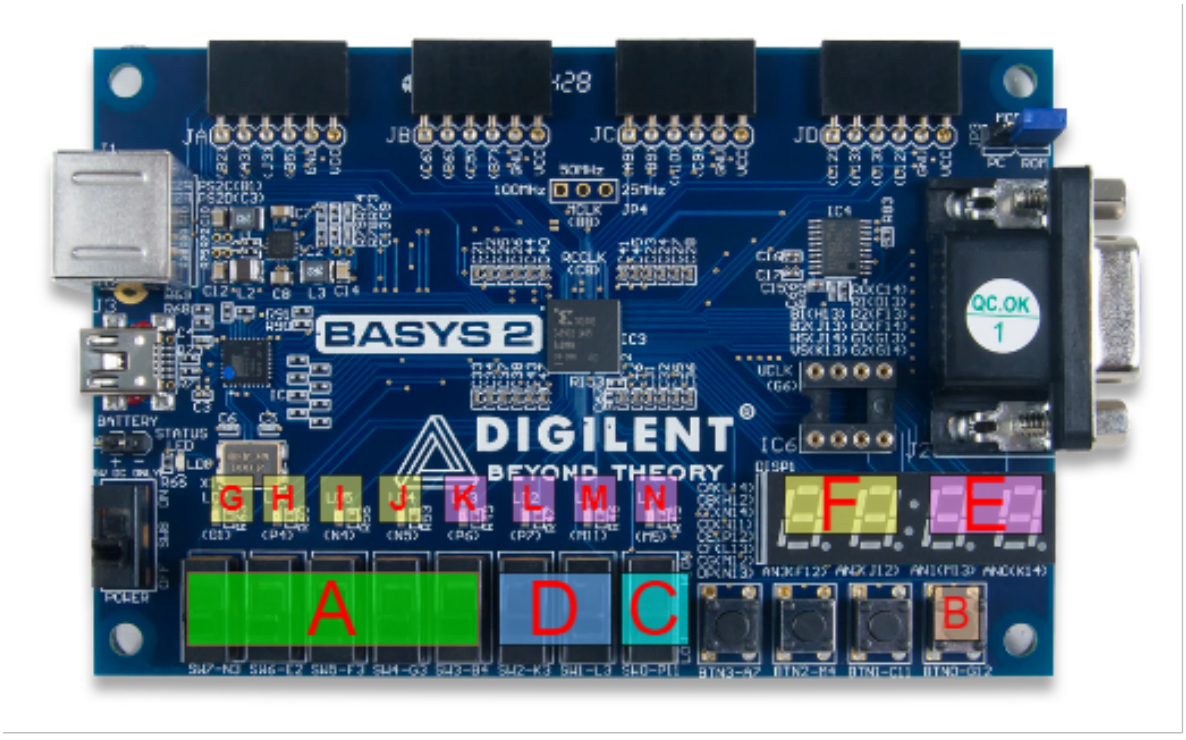

Figure 3: Board figure with labels

3.4 Deliverables

- Implement your module in a single Verilog file. Upload only **lab3_2.v** file to ODTUClass system.
  Do **NOT** submit your testbenches, bit files or other project files. You can share your testbenches on ODTUClass discussion page.
- Submit the file through the ODTUClass system before the deadline given at the top.
- Use the ODTUClass discussion for any questions regarding the homework.
- This is an individual work, any kind of cheating is not allowed.