1-point Homework 3

Use the last 3 digits of your personal number (PN) to select which question you have to answer. Modulo divide your 3 digits with the number of questions, in this case, 2, plus 1. The resulting number is the question that you should answer.

!!! (PN mod 2) + 1 !!!

For example, if your PN ends in 730, then (730 mod 2) + 1 = 1, meaning you must answer question 1.

Make sure you comment on the top of each file you submit, your PN, name and the question you are answering.

!!! IMPORTANT !!!

If you answer the wrong question, your submission will be invalid.

We can call you to explain your solution. If you cannot explain your answer, the homework will be invalid!

KTH has a zero-tolerance policy against cheating.

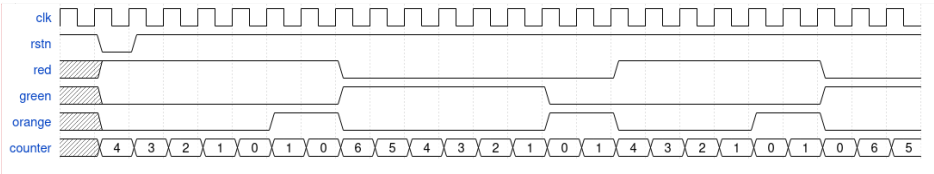
https://www.kth.se/en/student/stod/studier/fusk-1.997287

If you have questions or need clarifications, you can ask in the discussion forum in Canvas.

QUESTIONS

1.1 QUESTION 1

Implement a state machine to control a traffic light. The FSM should implement the above states and transitions with the corresponding delays, which will be specified as a parameter. Below you can find a sample of a timing diagram of its operation.



Module parameters

Name	Width	Description
RED_DELAY	--	Duration of Red state

Commented [JG1]: delete??

Commented [JG2]: {signal: [
{name: 'clk', wave: 'p.....'},
{name: 'rstn', wave: '101.....'},
{name: 'valid_key', wave: '0..10..1010101010..'},
{name: 'key', wave: '2..2...2.2.2.2.2...',
data: [0,'B',1,2,3,4,'C']},
{name: 'state', wave: 'x0..1.....0..'}

]}

RED_ORANGE_DELAY	--	Duration of RedOrange state
GREEN_DELAY	--	Duration of Green state
ORANGE_DELAY	--	Duration of Orange state

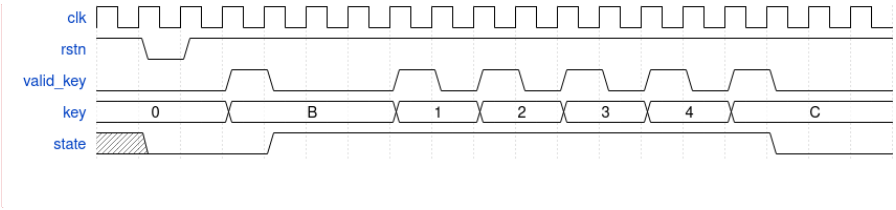
Module pinout

Name	**Direction**	**Width**	**Description**
clk	in	1	Clock signal
rstn	in	1	Active low reset
red	out	1	Signals when red light should be on
orange	out	1	Signals when orange light should be on
green	out	1	Signals when green light should be on

1.2 QUESTION 2

Implement a state machine to control a 4 digit digital lock. The lock has a keyboard with digits 0-9 and letters A-F. (It can be interpreted as hexadecimal input).

Above you can see an example of locking and unlocking the code using 1234 as pin.



Commented [JG3]: {signal: [
{name: 'clk', wave: 'p.....'},
{name: 'rstn', wave: '101.....'},
{name: 'valid_key', wave: '0..10..1010101010..'},
{name: 'key', wave: '2..2...2.2.2.2.2...',
data: [0,'B',1,2,3,4,'C']},
{name: 'state', wave: 'x0..1.....0..'}

]}

After reset, the lock starts in the unlocked state. After that the user can lock it by pressing key B. Once in locked state the lock can be unlocked by pressing the 4 correct digits followed by key C. The user can change the password while in the unlocked state by pressing the key B followed by the 4 digits of the new password.

Note that they signal valid_key should be present in all transitions but it has been omitted to simplify the diagram.

Module pinout

Name	Direction	Width	Description
clk	in	1	Clock signal
rstn	in	1	Active low reset
key	in	4	The “value” of the pressed key
valid_key	in	1	1 when the user presses a key
state	out	1	1: locked, 0: unlocked