Homework 2 (Due: Sep 27)

NetID: hb174 Name: Haochi Bai 2021/9/25

Instructions

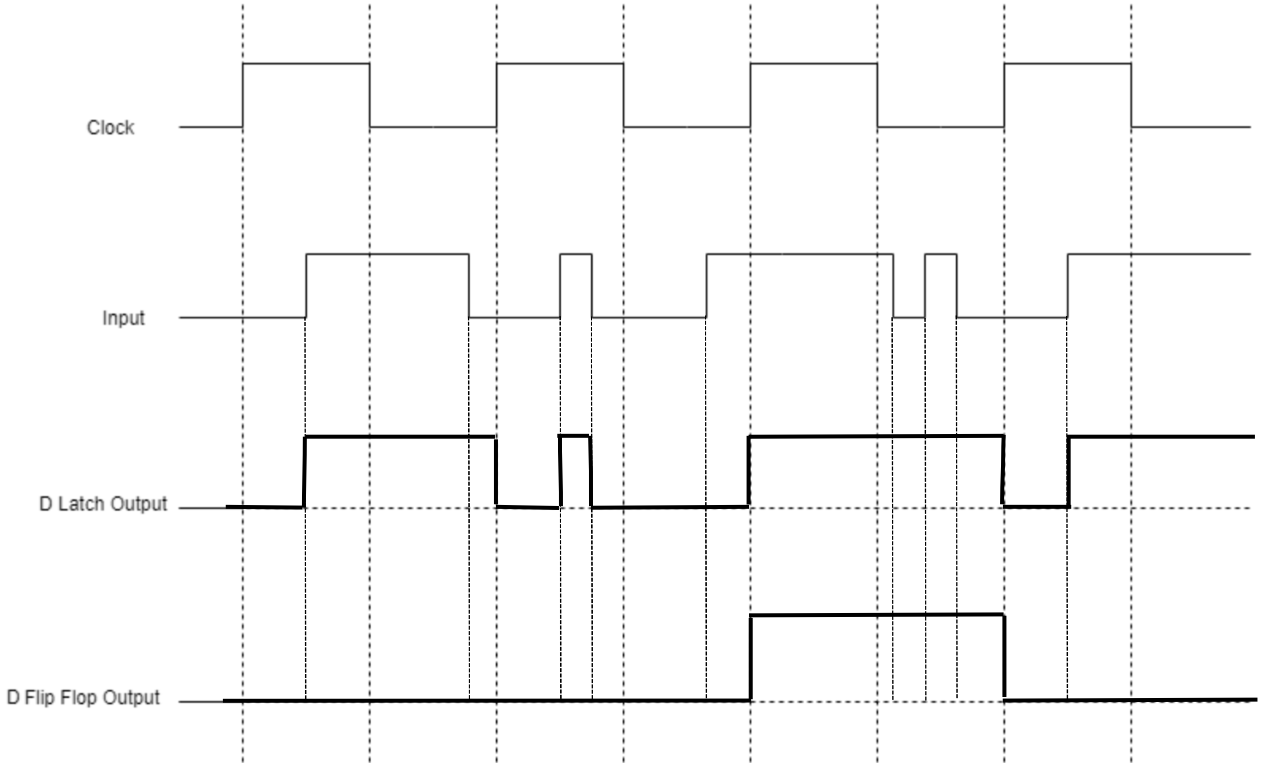
- Complete the following questions to the best of your ability.
- Answers should be clear, concise, and justified with work.
- Hand-drawn circuits will not be accepted due to the confusion they might cause.
    - Use software, such as Logisim, to draw them.
- Please write your name and NetID on the hardcopy of your solution, and email a PDF copy of your solution to Weiqiang Lv (weiqiang.lv@dukekunshan.edu.cn) before 1:15PM on the due day.

Collaboration

- Professor/TAs
    - You **may** discuss any content with any professor or TA.
- Students
    - You **may** discuss high-level concepts, techniques, jargon, keywords, related problems, or course content that is relevant to the material.
    - You **may not** discuss particular solutions to any of the questions.
    - If you have substantial conversations with any students, please note their name and, if you feel it necessary, the extent of your collaboration.
- Outside Sources (Internet, books)
    - You **may** use external references for any course content.
    - You **may** use an external tool to verify your solutions where appropriate, but not to generate solutions to any questions.
    - List any outside sources that you use. Formal citations are not necessary; links are fine.

Question 1 (24 points)

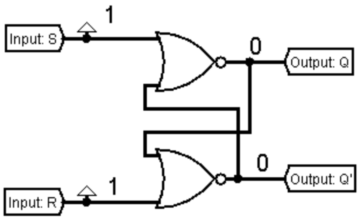
You have a D latch and a positive-edge-triggered D flip flop wired to the same data and clock inputs shown below. Assume that both are initially set to zero. Draw their respective outputs on the diagram. Note that you should fill out a total of 8 time slots for each output.



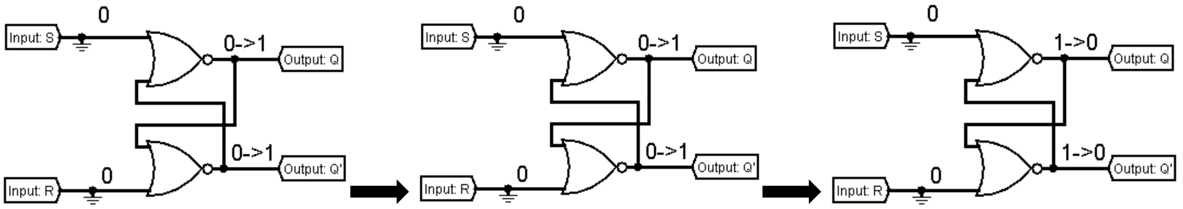
Question 2 (26 points)

In this question you will try to explain why it is bad to set both S and R in an SR-latch to 1. To do so, discuss what happens when SR = 11 then SR = 00 afterwards. How will the SR-latch behave in this case and why is this bad?

The initial status of SR-latch when SR=11,QQ'=00 is shown as below.

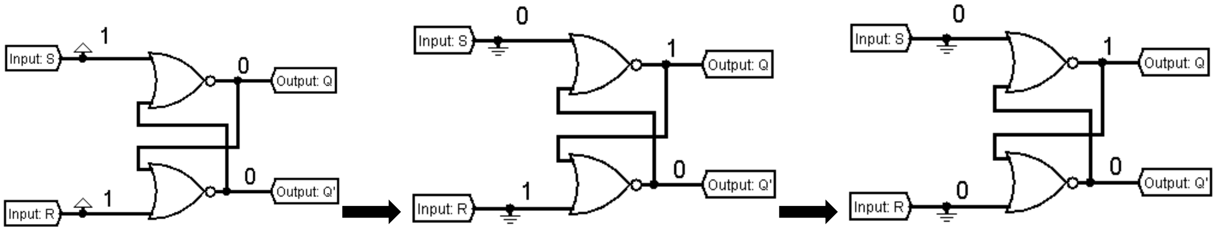


In ideal case, when S,R are set from 11 to 00, the output QQ' will change from 00->11->00….. constantly, it's definitely bad but will never happen in the real world.

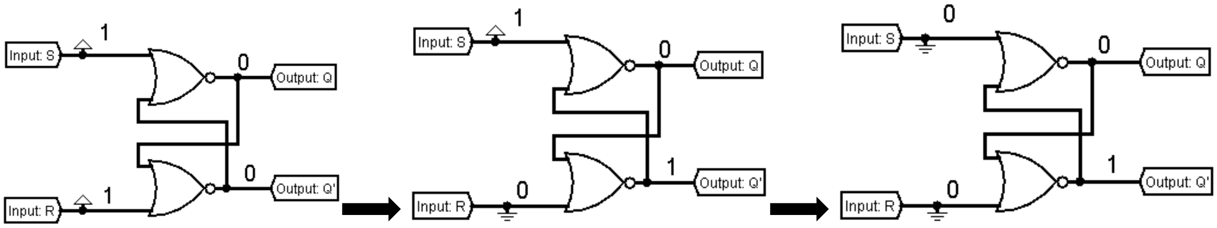


We consider what happens in the real world. Latency exists in real circuit, when S,R are set from 11 to 00, even though you set S and R to 0 simultaneously, one of the switches will still be set earlier than other one.

If S changes from 1 to 0 first, the output QQ' will be 10.



If R changes from 1 to 0 first, the output QQ' will be 01.

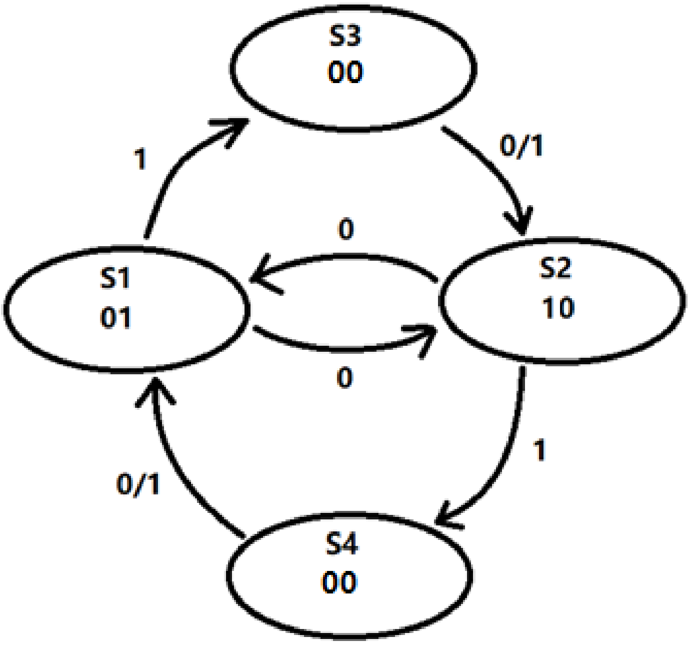


Since we cannot predict the output when we set SR=11 then SR=00, it is bad to set both S and R to 1.

Question 3 (50 points)

Design a simplified traffic-light controller that switches traffic lights on a crossroads (north-south direction intersects with east-west direction). When a traffic light signal is '1', the green light is on; otherwise, when the signal is '0', the red light is on. The lights alternate every one minute. Assume east-west direction starts at green first. There is also a "Walk" signal that pedestrians can activate. If this signal is activated, the next time the traffic lights alternate they would both stay red for one additional minute, and then they continue their normal operation. Assume that pressing the "Walk" button has no effect if both lights are red.

A. Draw the Moore state diagram of the described FSM.



B. Derive the state table.

State	State Next		Output	
	Input=0	Input=1	NS_Signal	WE_Signal
S1	S2	S3	0	1
S2	S1	S4	1	0
S3	S2	S2	0	0
S4	S1	S1	0	0

C. Derive the excitation table.

Current	Next	Walk (Input)
S1	S3	1
S1	S2	0
S2	S1	0
S2	S4	1
S3	S2	0/1
S4	S1	0/1

D. Write the equations for the next-state and output bits and minimize them.

Output bits:

	NS_Signal	WE_Signal
S1	0	1
S2	1	0
S3	0	0
S4	0	0

State transfer logic:

I define S1=(01), S2= (00), S3=(11), S4=(10),

State_Next <= S3 when State_Current= S1 and walk = 1 else

State_Next <= S2 when State_Current= S1 and walk = 0 else

State_Next <= S1 when State_Current= S2 and walk = 0 else

State_Next <= S4 when State_Current= S2 and walk = 1 else

State_Next <= S2 when State_Current= S3 and walk = 0 or 1 else

State_Next <= S1 (when State_Current=S4 and walk = 0 or 1)

↓

Truth table

Input:walk	Reg: State_Current		Out: State_Next		Output: Signal	
	Bit1	Bit0	Bit1	Bit0	NS_Signal	WE_Signal
1	0	1	1	1	0	1
0	0	1	0	0	0	1
0	0	0	0	1	1	0
1	0	0	1	0	1	0
X	1	1	0	0	0	0
X	1	0	0	1	0	0

Minimized Truth table

w	SC1	SC0	SN1	SN0	NS_Signal	WE_Signal
1	0	1	1	1	0	1
0	0	1	0	0	0	1
0	0	0	0	1	1	0
1	0	0	1	0	1	0
X	1	1	0	0	0	0
X	1	0	0	1	0	0

Equation:

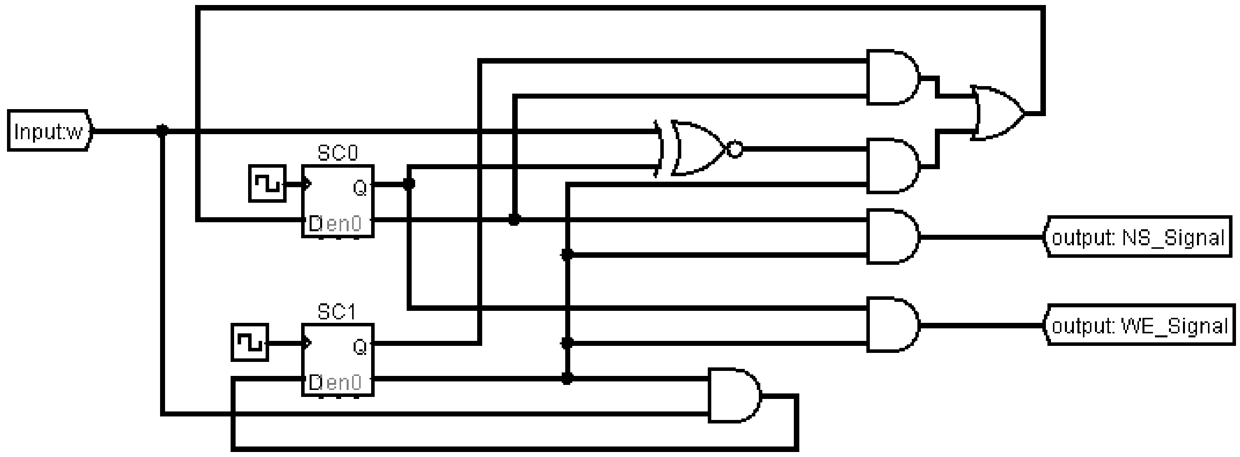
$$SN1 = w * \overline{SC1} * SC0 + w * \overline{SC1} * \overline{SC0} = w * \overline{SC1} * (SC0 + \overline{SC0}) = \boldsymbol{w * \overline{SC1}}$$

$$\begin{aligned} SN0 = {} & w * \overline{SC1} * SC0 + \overline{w} * \overline{SC1} * \overline{SC0} + SC1 * \overline{SC0} \\ & = \overline{SC1} * (w * SC0 + \overline{w} * \overline{SC0}) + SC1 * \overline{SC0} \\ & = \boldsymbol{\overline{SC1}} * (\boldsymbol{SC0} \odot \boldsymbol{w}) + \boldsymbol{SC1} * \boldsymbol{\overline{SC0}} \end{aligned}$$

Output Equation:

$$NS_{Signal} == \boldsymbol{\overline{SC1}} * \boldsymbol{\overline{SC0}}$$
$$WE_{Signal} == \boldsymbol{\overline{SC1}} * \boldsymbol{SC0}$$

E. Draw a schematic diagram for the FSM circuit using DFFs.


Input:w
SC0
Q
Den0
SC1
Q
Den0
output: NS_Signal
output: WE_Signal