Homework 3 - COMP2020

Student name: Nguyen Tuan Anh

Student ID: V202100512

Question 1

We will choose the Binary Encoding scheme as:

- `Init`: `00`
- `X1`: `01`
- `X1-X2`: `10`
- `X1-X2-X2`: `11`

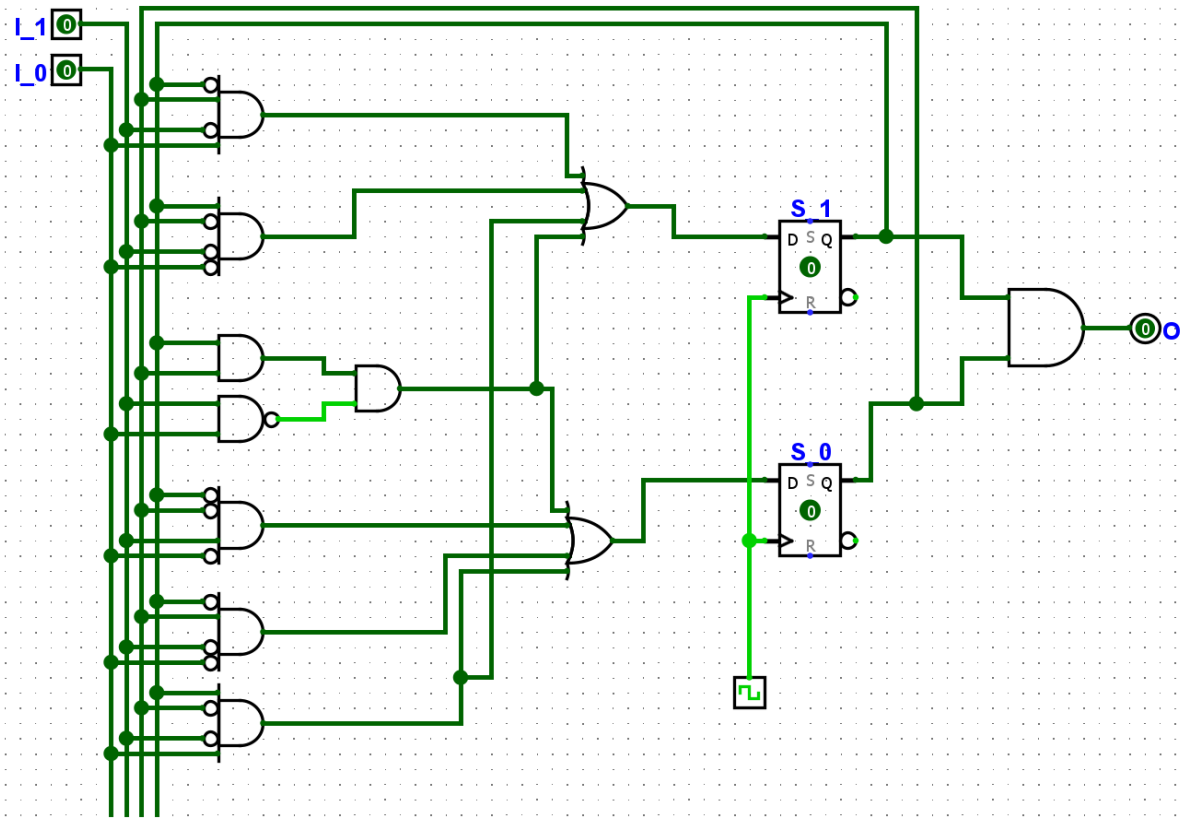
From the FSM, we can have the following truth table:

Current State		Input		Next State		Output
s1	s0	i1	i0	s1n	s0n	
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	0	1	0
0	0	1	1	0	0	0
0	1	0	0	0	1	0
0	1	0	1	1	0	0
0	1	1	0	0	0	0
0	1	1	1	0	0	0
1	0	0	0	1	0	0
1	0	0	1	1	1	1
1	0	1	0	0	0	0
1	0	1	1	0	0	0
1	1	0	0	1	1	1
1	1	0	1	1	1	1
1	1	1	0	1	1	1
1	1	1	1	0	0	0

We can write the binary equation for s_{0n} and s_{1n} as:

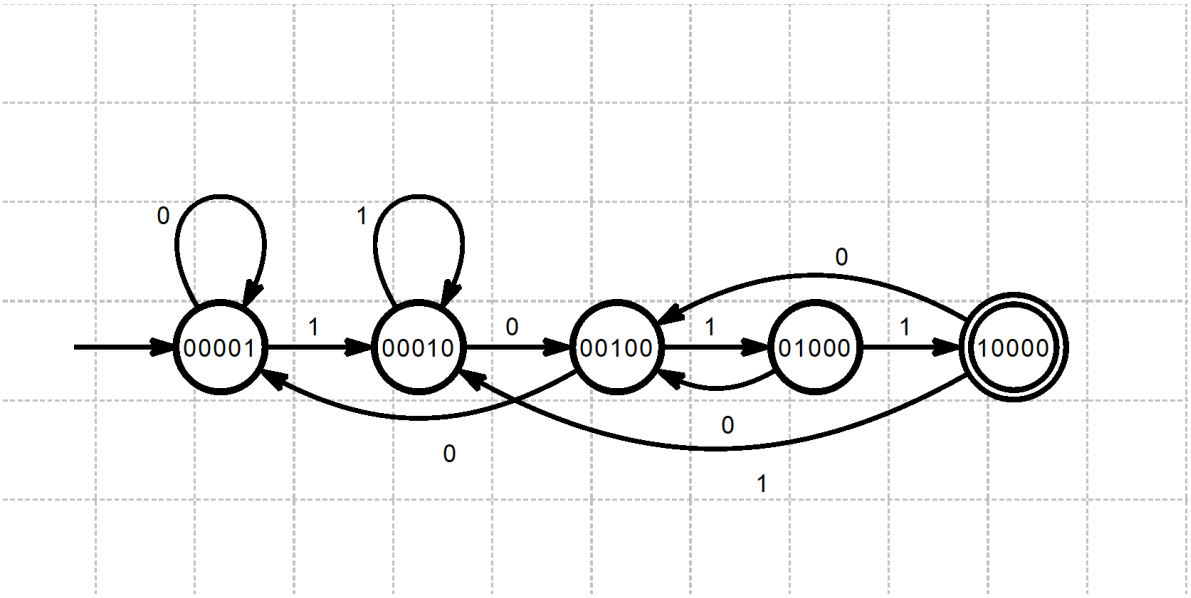
$$s_{0n} = \bar{s_1}\bar{s_0}i_1\bar{i_0} + \bar{s_1}s_0\bar{i_1}\bar{i_0} + s_1\bar{s_0}\bar{i_1}i_0 + s_1s_0(\bar{i_1}\bar{i_0} + \bar{i_1}i_0 + i_1\bar{i_0})$$
$$s_{1n} = \bar{s_1}s_0\bar{i_1}i_0 + s_1\bar{s_0}\bar{i_1}\bar{i_0} + s_1\bar{s_0}\bar{i_1}i_0 + s_1s_0(\bar{i_1}\bar{i_0} + \bar{i_1}i_0 + i_1\bar{i_0})$$

We can implement the circuit like this:



Question 2

For the sequence detector, we have this FSM:



Also, we have this truth table:

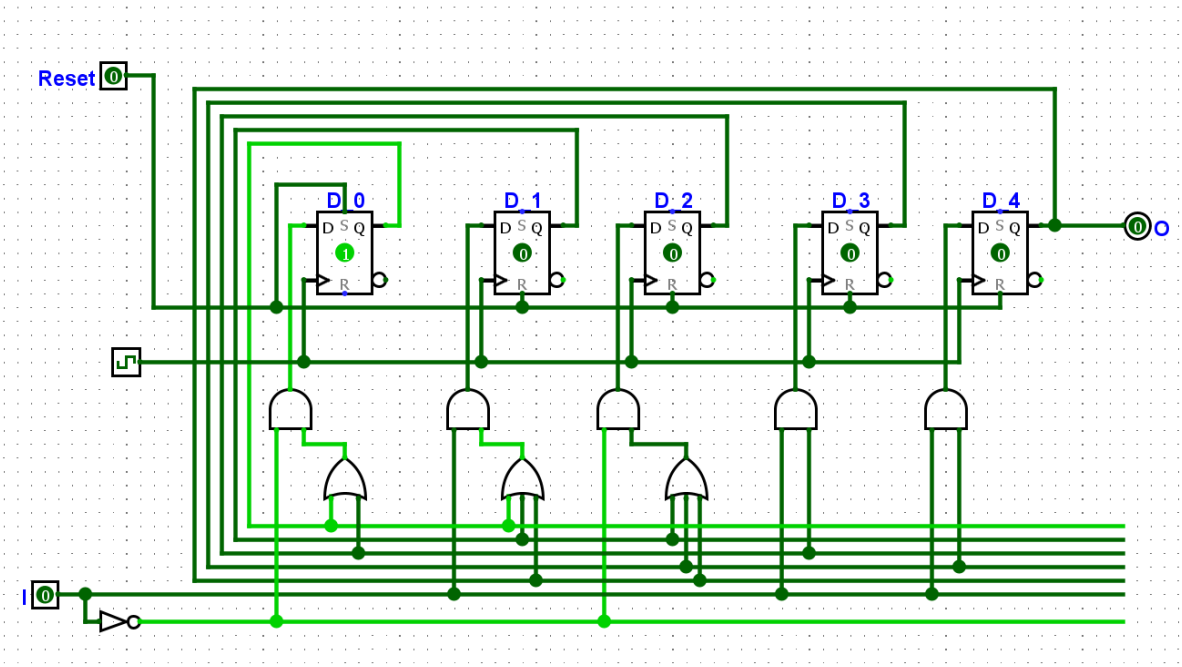
State	Input		Output
S	1	0	O
00001	00010	00001	0
00010	00010	00100	0
00100	01000	00001	0
01000	10000	00100	0
10000	00010	00100	1

We can write the binary equation for every state s_{xn}: (the output)

$$s_{0n} = \bar{i} \cdot (s_0 + s_2)$$
$$s_{1n} = i \cdot (s_0 + s_1 + s_4)$$
$$s_{2n} = \bar{i} \cdot (s_1 + s_3 + s_4)$$
$$s_{3n} = i \cdot s_2$$
$$s_{4n} = i \cdot s_3$$

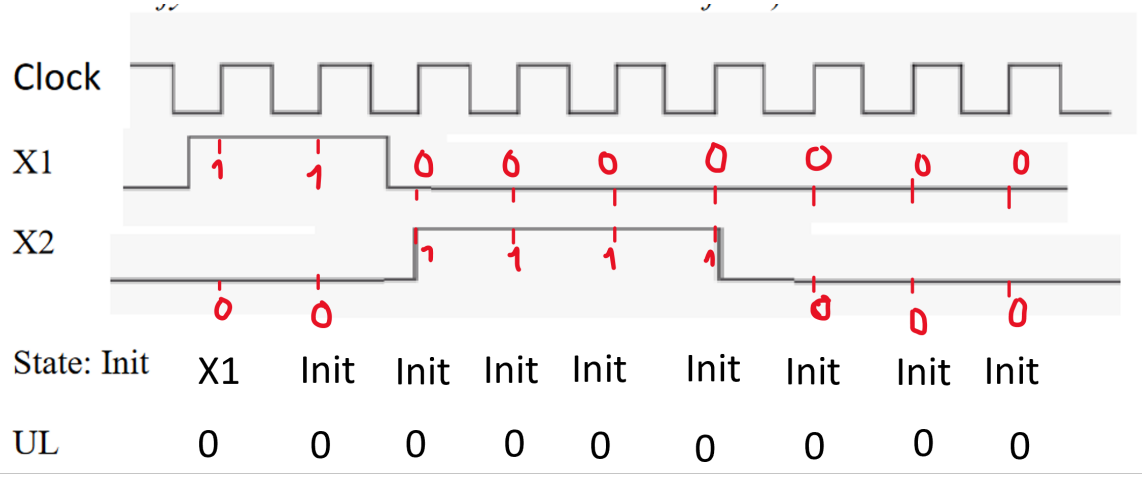
Where s_0, s_1, s_2, s_3, s_4 is the current state, and i is the input.

We can implement the circuit as:



Question 3

Assume the signal will be received on rising edge of the clock. We have the following result:



Question 4

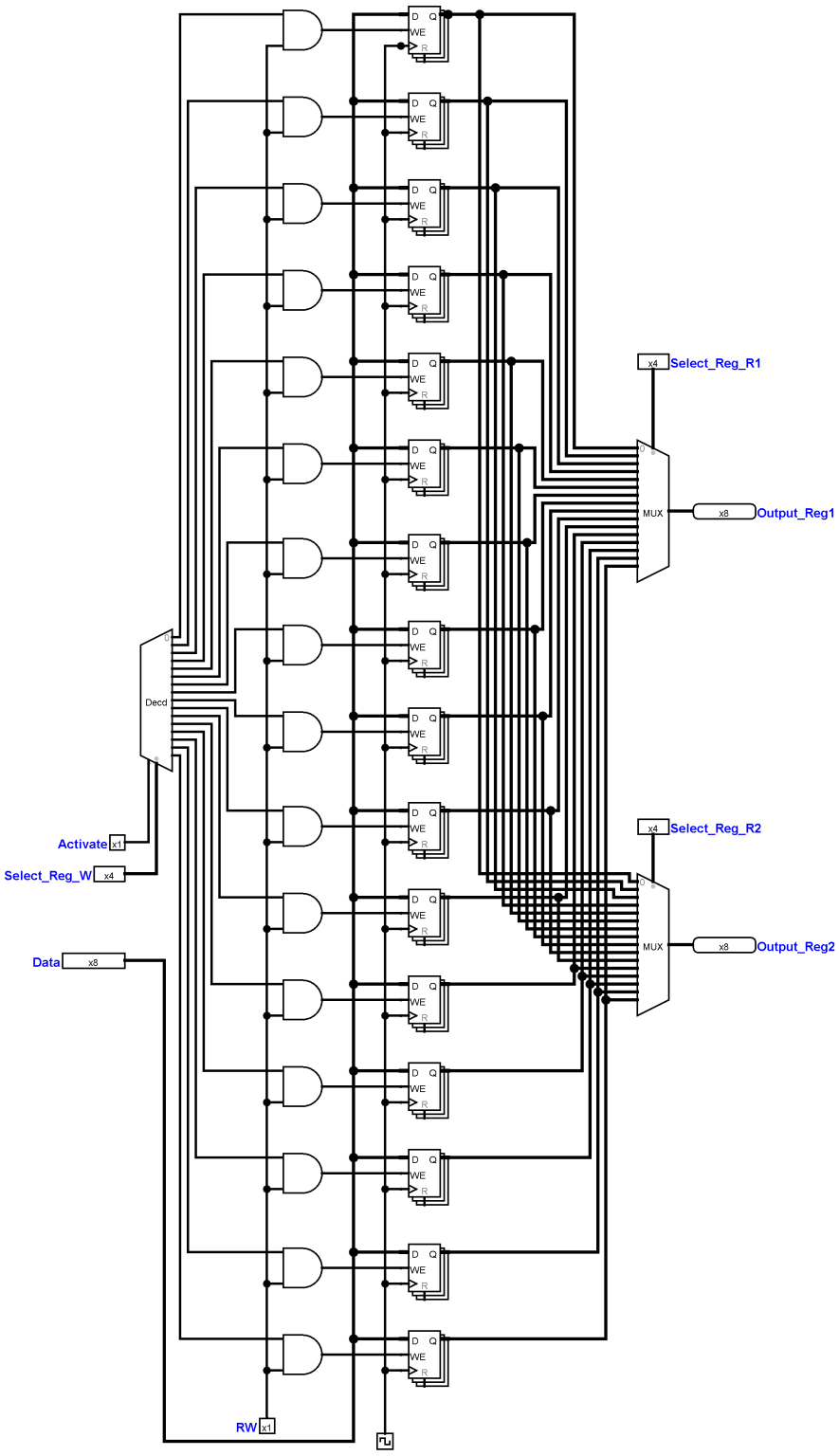
This register file supports 16 8-bit registers, with one write port and two write ports.

About I/O Ports:

- `RW`: 1 for allowing write to the register, 0 for read-only.
- `Activate`: Activate the Decoder for selecting the registers to write.
- `Select_Reg_W`: 4-bit value to select 1/16 registers (`0000` to `1111`) to write.
- `Data`: The 8-bit value for parsing to a register, based on the choice of `Select_Reg_W`.
- `Select_Reg_R1`, `Select_Reg_R2`: Two 4-bit values for selecting the data from 1/16 registers.
- `Output_Reg1`, `Output_Reg2`: Two 8-bit values, come from the register that selected from `Select_Reg_R1` and `Select_Reg_R2`.

The mechanism of the register file:

- The register file stores data of registers in 16 D Flip-flop, with number of data bits is 8.
- **Write**: With the `RW` and `Activate` enabled, value from `Select_Reg_W` will be parsed by decoder to the selected register. The AND Gate is used for enabling write action to a register when both write mode and the signal to that register is on.
- **Read**: 16 values from D Flip-flops will be sent to a multiplexer. The output of multiplexer will be chosen based on our selection of `Select_Reg_Rx`. Duplicate this design twice and we have two read ports.


Select_Reg_R1
x4
MUX
x8
Output_Reg1
Decd
Activate
x1
Select_Reg_W
x4
Select_Reg_R2
x4
MUX
x8
Output_Reg2
Data
x8
RW
x1
D
Q
WE
R