

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KỸ THUẬT MÁY TÍNH**

**BÀI BÁO CÁO CE118.P11.1
THIẾT LUẬN LÝ SỐ TUẦN 2**

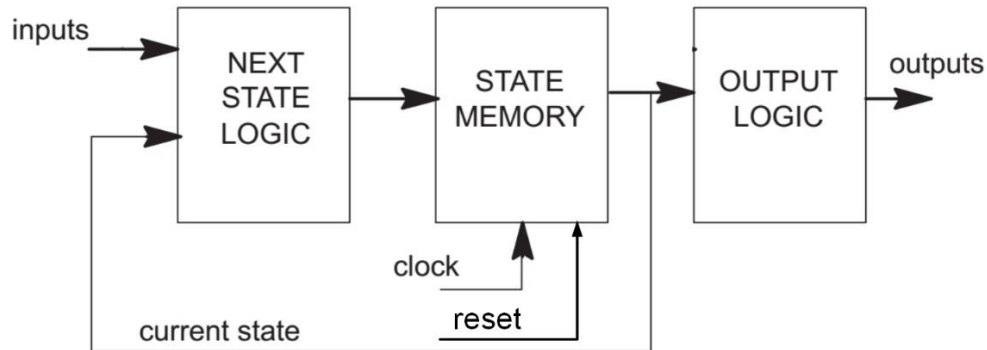
**THỰC HÀNH LAB02: THIẾT KẾ MÁY TRẠNG
THÁI HỮU HẠN (FINITE STATE MACHINE)**

**Sinh viên: Trương Thiên Quý
MSSV: 23521321
Lớp: CE118.P11.1
Giảng viên hướng dẫn: Tạ Trí Đức**

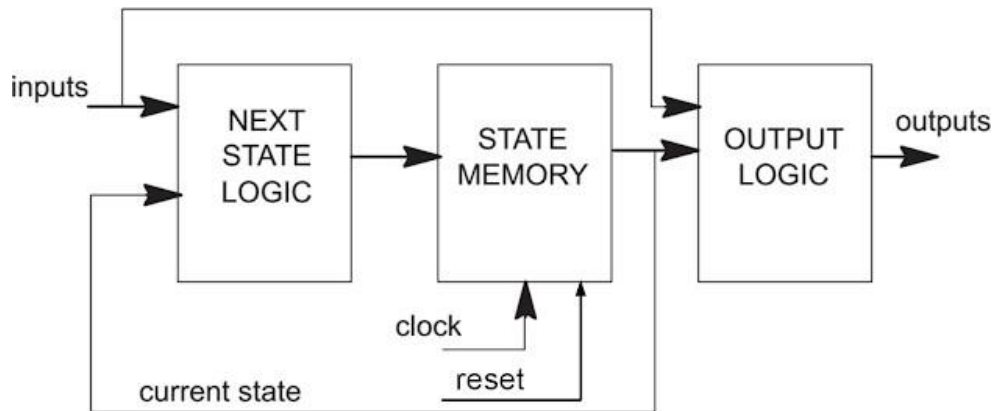
1. Lý Thuyết.

Giảng viên giới thiệu về **Máy trạng thái hữu hạn (Finite State Machine – FSM)** FSM được chia làm 2 loại:

FSM Moore là loại mạch có ngõ ra (output) không phụ thuộc trực tiếp vào ngõ vào (input).



FSM Mealy là loại mạch có ngõ ra (output) phụ thuộc trực tiếp vào ngõ vào (input).



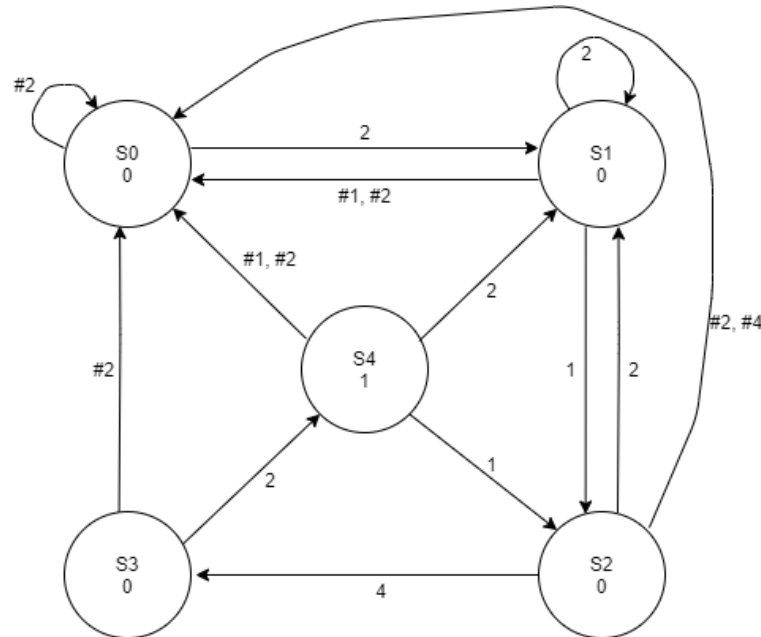
Một FSM gồm có 3 thành phần cơ bản như sau:

- Mạch tạo trạng thái kế tiếp (Next state logic) là mạch tổ hợp phụ thuộc vào ngõ vào FSM và giá trị trạng thái hiện tại lấy từ bộ nhớ trạng thái (state memory)
- Bộ nhớ trạng thái (state memory) là phần tử lưu trạng thái hiện tại của FSM, nó có thể là Flip-Flop, Latch, ... và lấy ngõ vào từ mạch tạo trạng thái kế tiếp.

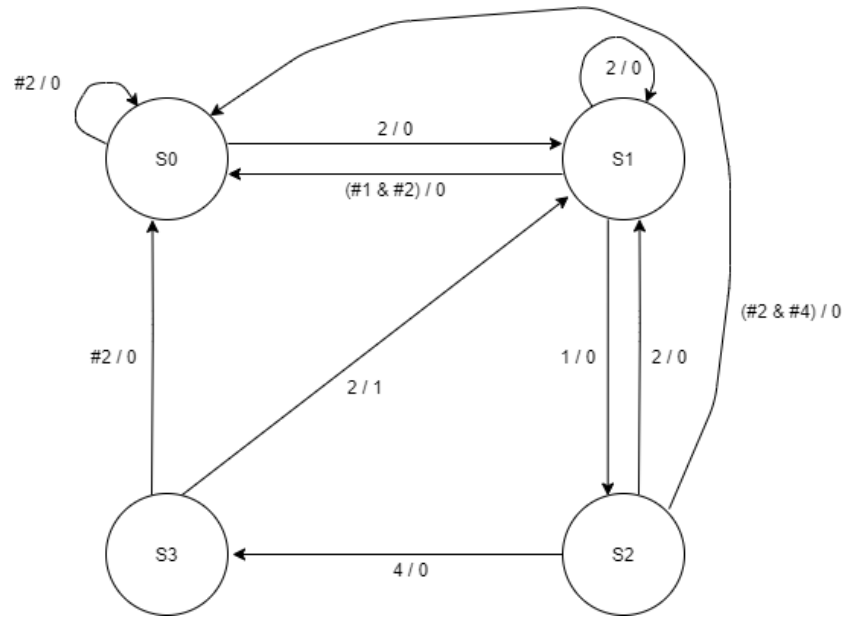
c. Mạch tạo ngõ ra (output logic) là mạch tổ hợp tạo giá trị ngõ ra tương ứng với trạng thái hiện tại của FSM.

Giảng viên hướng dẫn cách thiết kế một máy trạng thái hữu hạn bao gồm các bước cơ bản như sau:

+ Xác định số lượng trạng thái và lập sơ đồ trạng thái



Hình 1 - Ví dụ về sơ đồ trạng thái kiểu Moore



Hình 2 - Ví dụ về sơ đồ trạng thái kiểu Mealy

- + Mã hóa và rút gọn trạng thái
- + Lập bảng chuyển trạng thái
- + Thiết kế mạch và mô phỏng

1. Thực hành

Thiết kế bộ phát hiện số cuối của mã số sinh viên (số thứ 8) theo quy ước sử dụng Moore và Mealy:

- Nếu số thứ 7 là số lẻ thì chuyển số cuối ấy thành 4-bit, nếu chưa đủ sinh viên thêm các bit 0 vào đầu. (VD: số cuối MSSV là 2 thì dãy cần phát hiện là 0010, số cuối MSSV là 9 thì dãy cần phát hiện là 1001).
- Nếu số thứ 7 là số chẵn thì lấy số cuối trừ đi 2 và chuyển số cuối ấy thành 3-bit, riêng các bạn sinh viên có số cuối là 0,1,2 thì số sau khi trừ lần lượt là 5, 6, 7. Nếu chưa đủ sinh viên thêm các bit 0 vào đầu. (VD: số cuối MSSV là 0 thì dãy cần phát hiện là 101, số cuối MSSV là 3 thì dãy cần phát hiện là 001).

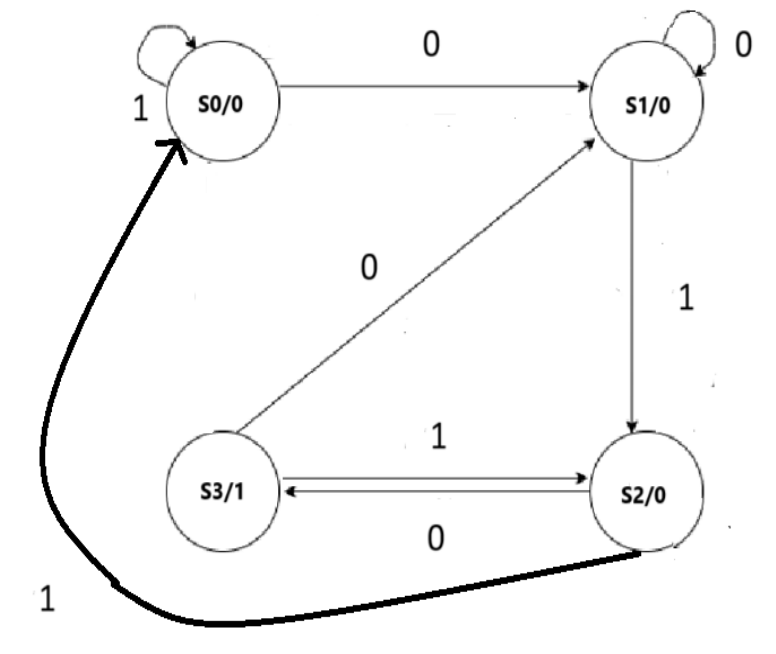
Khi phát hiện đúng số cuối theo quy ước trên thì output = 1, ngược lại bằng 0

****MOORE**

MSSV: 23521321

- Vì số thứ 7 là số 2 chuyển thành 3-bit sẽ là 010.

***Xác định số lượng trạng thái và lập sơ đồ trạng thái**



***Mã hóa trạng thái:**

Trạng thái	Mã hóa(Q1Q0)
S0	00
S1	01
S2	10
S3	11

-Vì có 4 trạng thái nên sẽ có tối thiểu 2 flip-flop được dùng

***Chọn loại Flip-Flop và lập bảng chuyển trạng thái:**

-Chọn D flip-flop để thiết kế mạch, khi đó bảng chuyển trạng thái sẽ được trình bày như trong bảng sau:

TTHT	TTKT		Ngõ vào flipflop				Ngõ ra
Q1Q0	Q1+Q0+		D1		D0		Z
	X=0	X=1	X=0	X=1	X=0	X=1	
00	01	00	0	0	1	0	0
01	01	10	0	1	1	0	0
10	11	00	1	0	1	0	0
11	01	10	0	1	1	0	1

***Rút gọn luận lí:**

X\Q1Q0	00	01	11	10
0	0	0	0	1
1	0	1	1	0

$$D1 = XQ0 + X'Q1Q0'$$

X\Q1Q0	00	01	11	10
0	1	1	1	1
1	0	0	0	0

$$D0 = X'$$

Q1\Q0	0	1
0	0	0
1	0	1

$$Z = Q1Q0$$

/



- **Input = 0:** Mạch nhận 0 và chuyển từ S0 sang S1 để bắt đầu tìm chuỗi 010.
- **Trạng thái hiện tại: S1**

- **Input = 1:** Mạch nhận **1** và chuyển từ **S1** sang **S2**, chờ nhận **0** để hoàn thành chuỗi **010**.
- **Trạng thái hiện tại:** S2

- **Input = 1:** Chuỗi bị gián đoạn bởi **1** thay vì **0**. Mạch không hoàn tất chuỗi và quay lại **S0** để tìm kiếm lại từ đầu.
- **Trạng thái hiện tại: S0**

- **Input = 0:** Mạch nhận **0** và chuyển từ **S0** sang **S1**, bắt đầu tìm kiếm chuỗi **010** mới.
- **Trạng thái hiện tại:** S1

5. Trạng thái S1:

- **Input = 1:** Mạch nhận **1** và chuyển từ **S1** sang **S2**, chờ nhận **0** để hoàn thành chuỗi **010**.
- **Trạng thái hiện tại:** S2

6. Trạng thái S2:

- **Input = 0:** Mạch nhận **0** và hoàn thành chuỗi **010**.
- **Trạng thái hiện tại:** S3

7. Trạng thái S3:

- **Chuỗi 010 đã hoàn tất, phát ra output = 1**
- **Input = 0:** Mạch nhận **0** và chuyển từ **S3** sang **S1** để bắt đầu tìm chuỗi **010** mới.
- **Trạng thái hiện tại:** S1

8. Trạng thái S1:

- **Input = 1:** Mạch nhận **1** và chuyển từ **S1** sang **S2**, chờ nhận **0** để hoàn thành chuỗi **010**.
- **Trạng thái hiện tại:** S2

9. Trạng thái S2:

- **Input = 0:** Mạch nhận **0** và hoàn thành chuỗi **010**.
- **Trạng thái hiện tại:** S3

10. Trạng thái S3:

- **Chuỗi 010 đã hoàn tất, phát ra output = 1**
- **Input = 1:** Mạch nhận **1** và chuyển từ **S3** sang **S2** để bắt đầu tìm chuỗi **010** mới.
- **Trạng thái hiện tại:** S2

11. Trạng thái S2:

- **Input = 1:** Chuỗi bị gián đoạn bởi **1** thay vì **0**. Mạch không hoàn tất chuỗi và quay lại **S0** để tìm kiếm lại từ đầu.
- **Trạng thái hiện tại:** S0

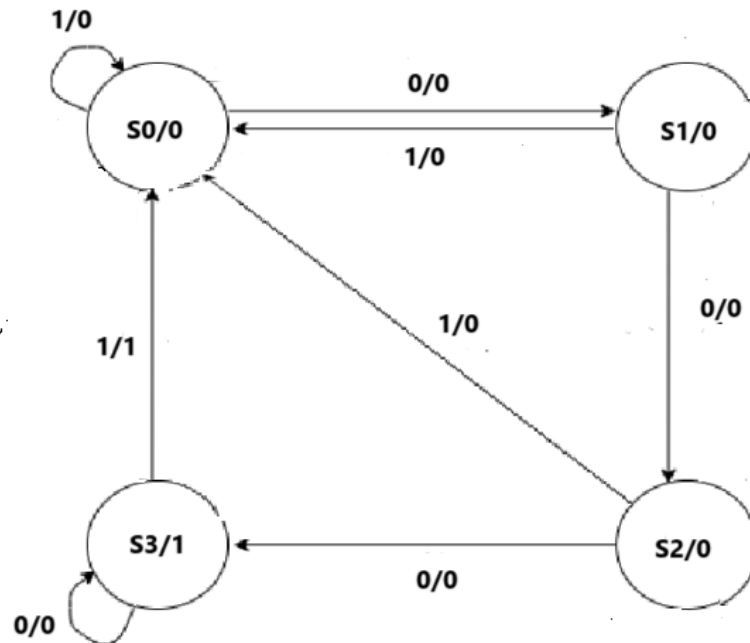
***Nạp KIT: [Lab2 Moore.mp4 - Google Drive](#)**

***MEALY:

MSSV: 23521321

-Vì số thứ 8 là số 1 nên 4-bit sẽ là 0001.

*Xác định số lượng trạng thái và lập sơ đồ trạng thái



*Mã hóa trạng thái:

Trạng thái	Mã hóa(Q1Q0)
S0	00
S1	01
S2	10
S3	11

-Vì có 3 trạng thái nên sẽ có tối thiểu 2 flip-flop được dùng

***Chọn loại Flip-Flop và lập bảng chuyển trạng thái:**

-Chọn D flip-flop để thiết kế mạch, khi đó bảng chuyển trạng thái sẽ được trình bày như trong bảng sau:

TTHT	TTKT		Ngõ vào flipflop				Ngõ ra	
Q1Q0	Q1+Q0+		D1		D0		Z	
	X=0	X=1	X=0	X=1	X=0	X=1	X=0	X=1
00	01	00	0	0	1	0	0	0
01	10	00	1	0	0	0	0	0
10	11	00	1	0	1	0	0	0
11	11	00	1	0	1	0	0	1

***Rút gọn luận lí:**

X\Q1Q0	00	01	11	10
0	0	1	1	1
1	0	0	0	0

$$\rightarrow D1 = X'(Q1 + Q0)$$

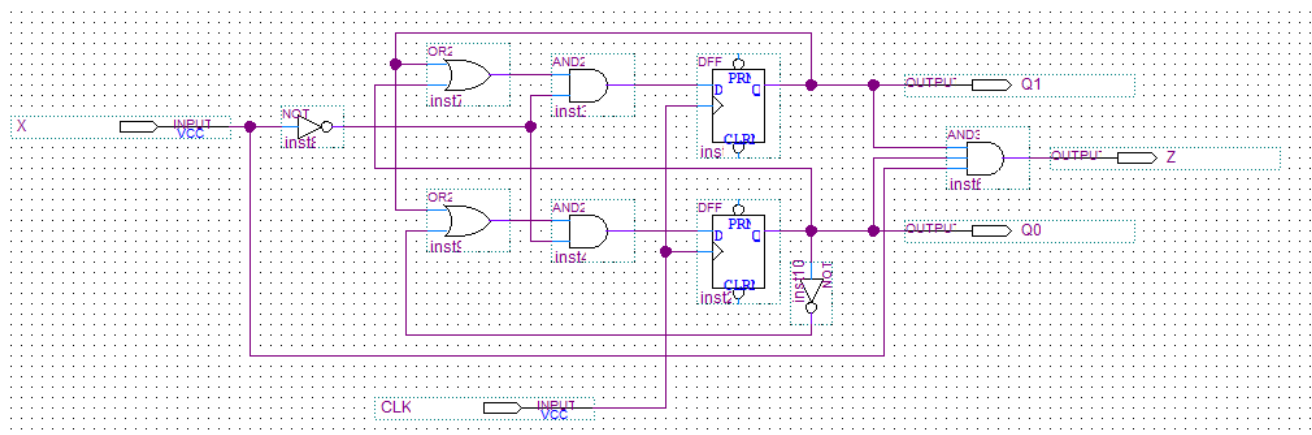
X\Q1Q0	00	01	11	10
0	1	0	1	1
1	0	0	0	0

$$\rightarrow D2 = X'Q0' + X'Q1$$

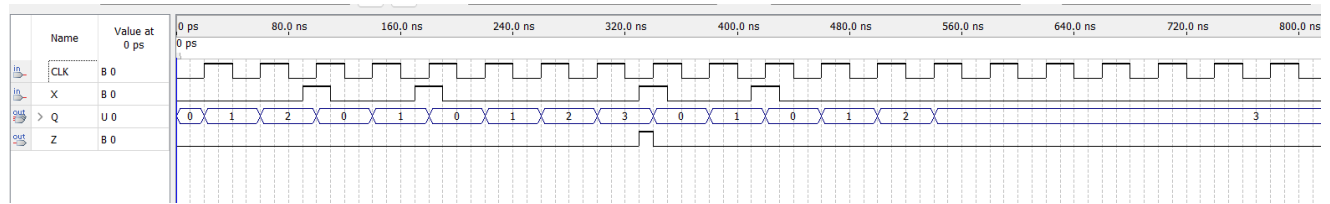
X\Q1Q0	00'	01'	11'	10'
0	0	0	0	0
1	0	0	1	0

$$Z = XQ1Q0$$

***Vẽ mạch luận lý:**



***Xác minh chức năng của mạch luận lý (Mô phỏng)**



→ **Giả sử nhập: 00101000101 với xung clock kích cạnh lên:**

1. **Bắt đầu ở trạng thái S0** (mạch chưa nhận được đầu vào nào):

- **Input = 0:** Khi mạch nhận được **0** đầu tiên, nó chuyển từ **S0** sang **S1**, chờ nhận thêm **0** để tiếp tục chuỗi.

- **Trạng thái hiện tại: S1**

2. **Trạng thái S1:**

- **Input = 0:** Mạch nhận được **0** thứ hai, chuyển từ **S1** sang **S2**, chờ thêm một **0** để đạt được ba **0** liên tục.

- **Trạng thái hiện tại: S2**

3. **Trạng thái S2:**

- **Input = 1:** Chuỗi bị gián đoạn do mạch nhận được **1** thay vì **0**. Do đó, mạch quay lại **S0** để bắt đầu lại từ đầu.

- **Trạng thái hiện tại: S0**

4. **Trạng thái S0:**

- **Input = 0:** Khi mạch nhận được **0** đầu tiên, nó chuyển từ **S0** sang **S1**, chờ nhận thêm **0** để tiếp tục chuỗi.

- **Trạng thái hiện tại: S1**

5. **Trạng thái S1:**

- **Input = 1:** Chuỗi bị gián đoạn do mạch nhận được **1** thay vì **0**. Do đó, mạch quay lại **S0** để bắt đầu lại từ đầu.

- **Trạng thái hiện tại: S0**

6. **Trạng thái S0:**

- **Input = 0:** Khi mạch nhận được **0** đầu tiên, nó chuyển từ **S0** sang **S1**, chờ nhận thêm **0** để tiếp tục chuỗi.

- **Trạng thái hiện tại: S1**

7. Trạng thái S1:

- **Input = 0:** Mạch nhận được 0 thứ hai, chuyển từ **S1** sang **S2**, chờ thêm một 0 để đạt được ba 0 liên tục.
- **Trạng thái hiện tại:** S2

8. Trạng thái S2:

- **Input = 0:** Mạch nhận được 0 thứ ba liên tiếp, chuyển từ **S2** sang **S3**, chỉ còn thiếu 1 để hoàn tất chuỗi **0001**.
- **Trạng thái hiện tại:** S3

9. Trạng thái S3:

- **Input = 1:** Mạch nhận được 1 sau ba 0, hoàn thành chuỗi **0001**. Mạch phát hiện chuỗi và phát ra **output = 1**, sau đó chuyển trở lại **S0** để bắt đầu tìm kiếm chuỗi mới.
- **Trạng thái hiện tại:** S0

10. Trạng thái S0:

- **Input = 0:** Mạch nhận 0 và chuyển từ **S0** sang **S1**, chuẩn bị tìm kiếm chuỗi **0001** mới.
- **Trạng thái hiện tại:** S1

11. Trạng thái S1:

- **Input = 1:** Kết thúc chuỗi đầu vào mà không có thêm tín hiệu 0 để khởi động chuỗi mới. Mạch ở lại S1.

***Nap KIT: [Lab2 Mealy.mp4 - Google Drive](#)**

2. Bài tập làm thêm

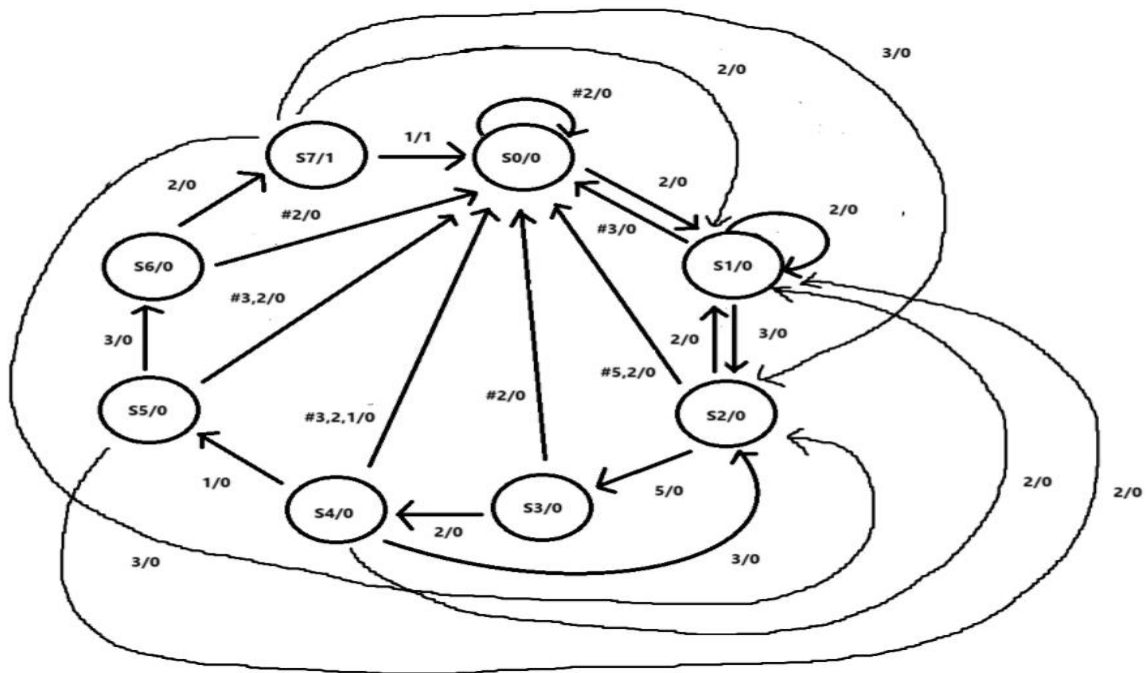
Thiết kế bộ phát hiện MSSV (8 ký số) theo FSM Moore và Mealy.

Giả sử MSSV là 23456789. Khi input lần lượt là 2, 3, 4, 5, 6, 7, 8, 9, lúc input = 9 thì output = 1, có nghĩa là đã phát hiện 8 số của MSSV.

Ngược lại output = 0 khi input không thỏa mãn điều trên, chẳng hạn: input lần lượt là 2, 6, 7, 3, 9... hay 3, 6, 8, 9 ...

***Xác định số lượng trạng thái và lập sơ đồ trạng thái**

✓ MSSV: 23521321



Từ sơ đồ trên, ta có:

	0	1	2	3	4	5
S0	S0	S0	S1	S0	S0	S0
S1	S0	S0	S1	S2	S0	S0
S2	S0	S0	S1	S0	S0	S3
S3	S0	S0	S4	S0	S0	S0
S4	S0	S5	S1	S2	S0	S0
S5	S0	S0	S1	S6	S0	S0
S6	S0	S0	S7	S0	S0	S0
S7	S0	S0/1	S1	S2	S0	S0

- Vì có **8** trạng thái nên sẽ có tối thiểu **3 flip-flop** được dùng

***Mã hóa trạng thái:**

Trạng thái	Mã hóa(Q2Q1Q0)
S0	000
S1	001
S2	010
S3	011
S4	100
S5	101
S6	110
S7	111

***Chọn loại Flip-Flop và lập bảng chuyển trạng thái:**

-Chọn D flip-flop để thiết kế mạch, khi đó bảng chuyển trạng thái sẽ được trình bày như trong bảng sau:

Input			TTHT			TTKT			Ngõ vào DFF			Output
X2	X1	X0	Q2	Q1	Q0	Q2+	Q1+	Q0+	D2	D1	D0	O
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0	0	0
0	0	1	0	1	1	0	0	0	0	0	0	0

0	0	1	1	0	0	1	0	1	1	0	1	0
0	0	1	1	0	1	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0	0	0	0	0	0
0	0	1	1	1	1	0	0	0	0	0	0	1
0	1	0	0	0	0	0	0	1	0	0	1	0
0	1	0	0	0	1	0	0	1	0	0	1	0
0	1	0	0	1	0	0	0	1	0	0	1	0
0	1	0	0	1	1	1	0	0	1	0	0	0
0	1	0	1	0	0	0	0	1	0	0	1	0
0	1	0	1	0	1	0	0	1	0	0	1	0
0	1	0	1	1	0	1	1	1	1	1	1	0
0	1	0	1	1	1	0	0	1	0	0	1	0
0	1	1	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	1	0	1	0	0	1	0	0
0	1	1	0	1	0	0	0	0	0	0	0	0
0	1	1	0	1	1	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	0	0
0	1	1	1	0	1	1	1	0	1	1	0	0
0	1	1	1	1	0	0	0	0	0	0	0	0
0	1	1	1	1	1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0	0	0
1	0	0	0	1	1	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0	0	0	0	0	0
1	0	0	1	1	1	0	0	0	0	0	0	0

1	0	1	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0	0	0
1	0	1	0	1	0	0	1	1	0	1	1	0
1	0	1	0	1	1	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	0
1	0	1	1	0	1	0	0	0	0	0	0	0
1	0	1	1	1	0	0	0	0	0	0	0	0
1	0	1	1	1	1	0	0	0	0	0	0	0
X	X	X	X	X	X	0	0	0	0	0	0	0

***Rút gọn luận lí:**

X2'X1'				
X0Q2\Q1Q0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	0	0	0
10	0	0	0	0

X2'X1				
X0Q2\Q1Q0	00	01	11	10
00	0	0	1	0
01	0	0	0	1
11	0	1	0	0
10	0	0	0	0

X2X1				
X0Q2\Q1Q0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	0	0
10	0	0	0	0

X2X1'				
X0Q2\Q1Q0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	0	0
10	0	0	0	0

$$\rightarrow D2 = X2'X1'X0Q2Q1'Q0' + X2'X1X0'Q2'Q1Q0 + X2'X1X0'Q2Q1Q0' + X2'X1X0Q2Q1'Q0$$

X2'X1'				
X0Q2\Q1Q0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	0	0
10	0	0	0	0

X2'X1				
X0Q2\Q1Q0	00	01	11	10
00	0	0	0	0
01	0	0	0	1
11	0	1	0	0
10	0	1	0	0

X2X1				
X0Q2\Q1Q0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	0	0
10	0	0	0	0

X2X1'				
X0Q2\Q1Q0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	0	0
10	0	0	0	1

$$\rightarrow D1 = X2'X1X0'Q2Q1Q0' + X2X1'X0Q2'Q1Q0' + X2'X1X0Q1'Q0$$

X2'X1'				
X0Q2\Q1Q0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	0	0	0
10	0	0	0	0

X2'X1				
X0Q2\Q1Q0	00	01	11	10
00	1	1	0	1
01	1	1	1	1
11	0	0	0	0
10	0	0	0	0

X2X1				
X0Q2\Q1Q0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	0	0
10	0	0	0	0

X2X1'				
X0Q2\Q1Q0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	0	0
10	0	0	0	1

$$\rightarrow D0 = X2'X1'X0Q2Q1'Q0' + X2X1'X0Q2'Q1Q0' + X2'X1X0'Q1' + X2'X1X0'Q0' + X2'X1X0'Q2$$

X2'X1'				
X0Q2\Q1Q0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	1	0
10	0	0	0	0

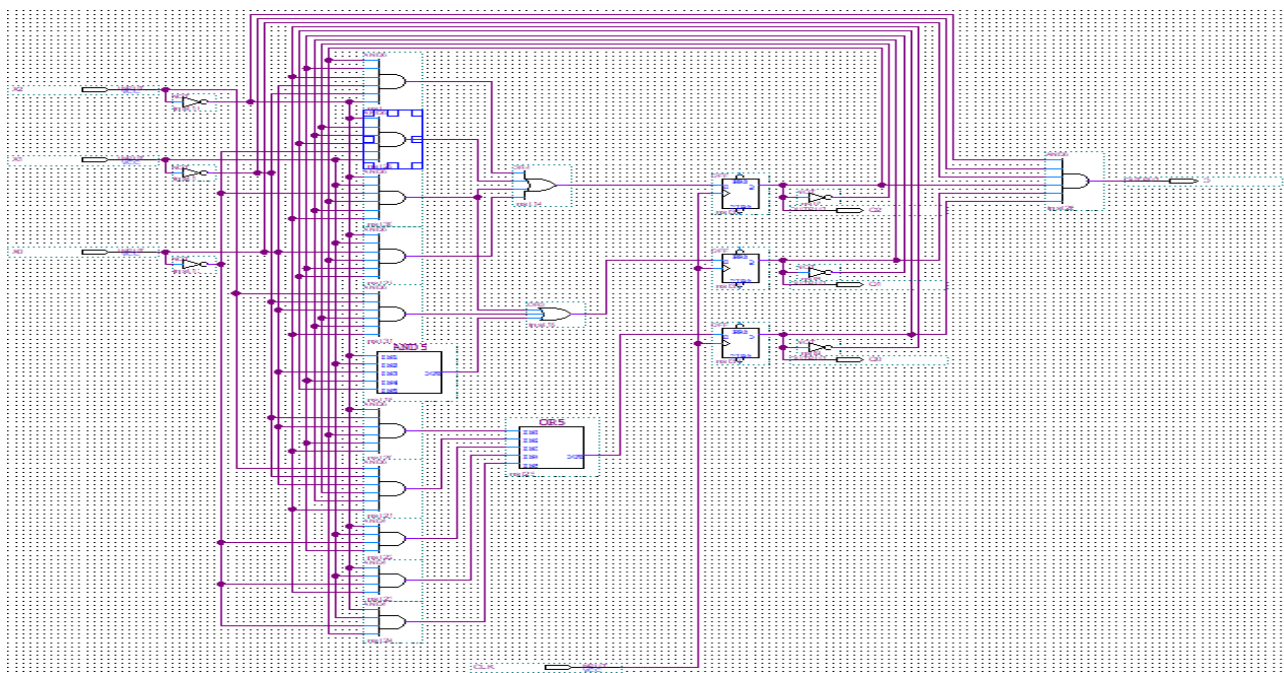
X2'X1				
X0Q2\Q1Q0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	0	0
10	0	0	0	0

X2X1				
X0Q2\Q1Q0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	0	0
10	0	0	0	0

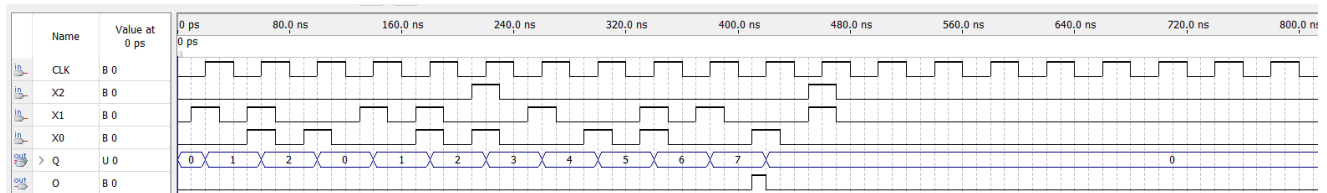
X2X1'				
X0Q2\Q1Q0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	0	0
10	0	0	0	0

$$\rightarrow O = X2'X1'X0Q2Q1Q0$$

***Vẽ mạch luận lý:**



*Xác minh chức năng và mô phỏng:



➔ Giả sử nhập: 23123521321 với xung clock kích cạnh lên:

- Khi input lần lượt là **2,3,1** thì **input không thỏa mãn** điều trên, do đó:
 1. Mạch sẽ ngay lập tức quay về **trạng thái 0**.
 2. **Output = 0** để biểu thị rằng chuỗi không thỏa mãn.
- Khi input lần lượt là **2, 3, 5, 2, 1, 3, 2, 1**, lúc **input = 1** thì **output = 1**, có nghĩa là đã **phát hiện 8 số của MSSV**:
 1. **Trạng thái ban đầu là 0**. Khi **input là 2**, mạch nhận tín hiệu đầu vào và thực hiện chuyển đổi trạng thái. Lúc này, mạch sẽ **chuyển từ trạng thái 0 sang trạng thái 1**.
 2. **Trạng thái hiện tại là 1**. Tiếp theo, khi **input là 3**, mạch một lần nữa nhận tín hiệu đầu vào và thực hiện chuyển đổi trạng thái tiếp theo. Do đó, **mạch chuyển từ trạng thái 1 sang trạng thái 2**.
 3. **Trạng thái hiện tại là 2**. Khi **input là 5**, tín hiệu đầu vào này khiến mạch tiếp tục thay đổi trạng thái. Mạch sẽ **chuyển từ trạng thái 2 sang trạng thái 3**.
 4. **Trạng thái hiện tại là 3**. Với **input là 2**, mạch sẽ nhận tín hiệu và thực hiện bước nhảy tiếp theo. Điều này giúp mạch **chuyển từ trạng thái 3 sang trạng thái 4**.
 5. **Trạng thái hiện tại là 4**. Khi **input là 1**, mạch lại tiếp tục nhận tín hiệu đầu vào và chuyển đổi trạng thái, lần này **từ trạng thái 4 sang trạng thái 5**.
 6. **Trạng thái hiện tại là 5**. Tiếp theo, khi **input là 3**, mạch thực hiện một bước nhảy tiếp, chuyển đổi **từ trạng thái 5 sang trạng thái 6**.
 7. **Trạng thái hiện tại là 6**. Khi **input là 2**, tín hiệu đầu vào này khiến mạch chuyển đổi trạng thái, và **mạch di chuyển từ trạng thái 6 sang trạng thái 7**.
 8. **Trạng thái hiện tại là 7**. Cuối cùng, khi **input là 1**, mạch nhận tín hiệu và **chuyển từ trạng thái 7 quay trở về trạng thái 0**. Tại thời điểm này, mạch phát ra tín hiệu **output = 1** để biểu thị rằng chu kỳ trạng thái đã hoàn tất và quay trở lại trạng thái ban đầu.

Sau khi quay lại **trạng thái 0**, nếu chu kỳ tiếp tục với các giá trị input tương tự, hệ thống sẽ lặp lại quy trình chuyển đổi như trên. Điều này có thể tạo ra một dạng vòng lặp hoặc đếm theo một chu kỳ cố định.

