I.LÝ THUYẾT

Máy trạng thái hữu hạn (Finite State Machine – FSM):

- Máy trạng thái hữu hạn (FSM - Finite State Machine) là một mô hình tính toán được sử dụng để mô tả hành vi của các hệ thống đơn giản. Nó bao gồm một tập trạng thái, một tập các ký hiệu đầu vào, một tập các ký hiệu đầu ra và một bảng chuyển trạng thái, thể hiện cách mà hệ thống chuyển đổi từ một trạng thái này sang một trạng thái khác dựa trên các ký hiệu đầu vào.

- Mỗi trạng thái trong FSM biểu thị một tình huống cụ thể trong hệ thống, và các trạng thái được kết nối với nhau bằng các chuyển đổi dựa trên các ký hiệu đầu vào. Khi hệ thống nhận được một ký hiệu đầu vào, nó sẽ chuyển đổi sang một trạng thái mới, tùy thuộc vào trạng thái hiện tại và ký hiệu đầu vào.

- FSM là một công cụ hữu ích trong việc thiết kế các hệ thống điều khiển, điều khiển tự động, và các hệ thống logic khác. Nó cũng được sử dụng rộng rãi trong lĩnh vực lập trình máy tính để kiểm soát luồng điều khiển của chương trình.

- FSM được chia làm 2 loại là FSM Moore và FSM Mealy.

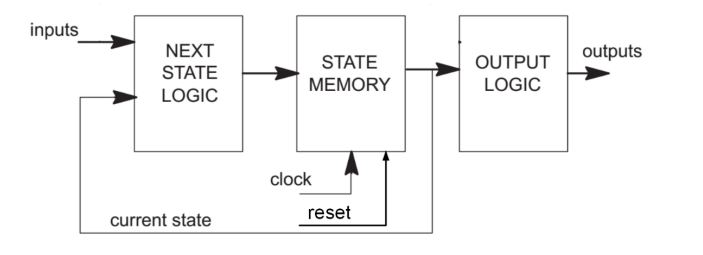
**FSM Moore**:

- FSM Moore là một loại máy trạng thái hữu hạn (FSM) trong đó trạng thái hiện tại của hệ thống xác định đầu ra của nó. Điều này có nghĩa là đầu ra được gắn với mỗi trạng thái, thay vì chỉ gắn với từng chuyển đổi giữa các trạng thái. FSM Moore được đặt tên theo Elwyn B. Moore, người đã phát minh ra loại FSM này vào năm 1956.

- Trong FSM Moore, khi hệ thống chuyển đổi từ trạng thái này sang trạng thái khác, đầu ra được xác định bởi trạng thái hiện tại. Điều này có nghĩa là đầu ra được xác định ngay khi hệ thống chuyển đổi sang trạng thái mới, mà không cần chờ đến khi một ký hiệu đầu vào được nhận.

- FSM Moore được sử dụng rộng rãi trong các ứng dụng điều khiển, ví dụ như điều khiển các thiết bị điện tử và các hệ thống tự động hóa khác. Nó cũng được sử dụng trong lĩnh vực thiết kế mạch điện tử, nơi nó có thể được sử dụng để mô hình hành vi của các mạch số và kiểm tra tính đúng đắn của chúng.

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



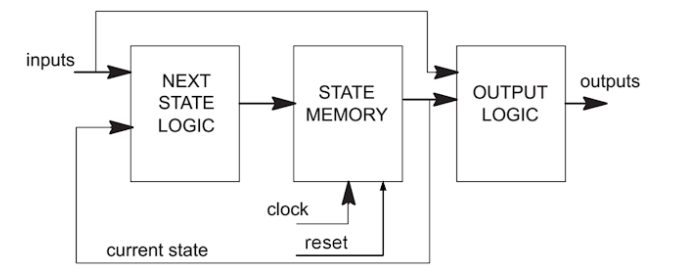
**FSM Mealy**:

- FSM Mealy là một loại máy trạng thái hữu hạn (FSM) trong đó đầu ra của hệ thống được xác định bởi trạng thái hiện tại cùng với các ký hiệu đầu vào. Điều này có nghĩa là đầu ra được gắn với từng chuyển đổi giữa các trạng thái, thay vì chỉ gắn với mỗi trạng thái. FSM Mealy được đặt tên theo George H. Mealy, người đã phát minh ra loại FSM này vào năm 1955.

- Trong FSM Mealy, khi hệ thống chuyển đổi từ trạng thái này sang trạng thái khác, đầu ra được xác định bởi trạng thái hiện tại cùng với ký hiệu đầu vào. Điều này có nghĩa là đầu ra được xác định khi hệ thống nhận được một ký hiệu đầu vào và chuyển đổi sang trạng thái mới.

- FSM Mealy được sử dụng rộng rãi trong các ứng dụng điều khiển và xử lý tín hiệu, ví dụ như trong các hệ thống điện tử, điều khiển nhiệt độ và các hệ thống tự động hóa khác. Nó cũng được sử dụng trong lĩnh vực thiết kế mạch điện tử để mô hình hành vi của các mạch số và kiểm tra tính đúng đắn của chúng.

- 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:

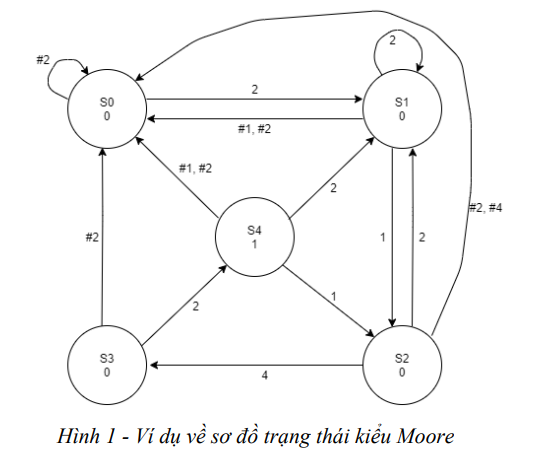
a. Mạch tạo trạng thái kế tiếp (Next state logic) là mạch tổ hợp phụ thuộc vào ngõ vào FSM và giá trị trạng thái hiện tại lấy từ bộ nhớ trạng thái (state memory)

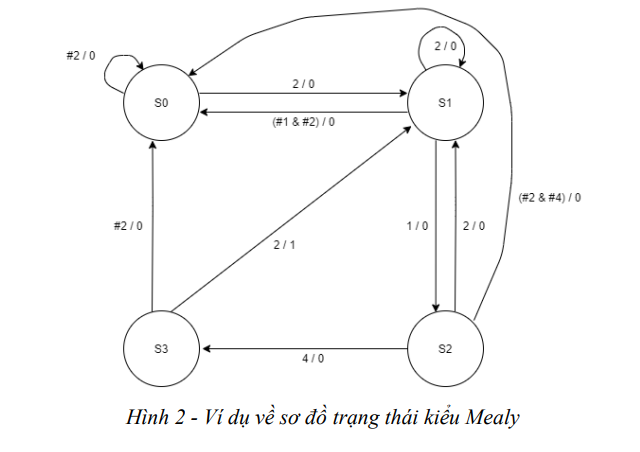
b. Bộ nhớ trạng thái (state memory) là phần tử lưu trạng thái hiện tại của FSM, nó có thể là Flip-Flop, Latch, ... và lấy ngõ vào từ mạch tạo trạng thái kế tiếp.

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.

Các bước thiết kế một máy trạng thái hữu hạn:

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





+ Mã hóa và rút gọn trạng thái

+ Lập bảng chuyển trạng thái

+ Thiết kế mạch và mô phỏng

II. 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:

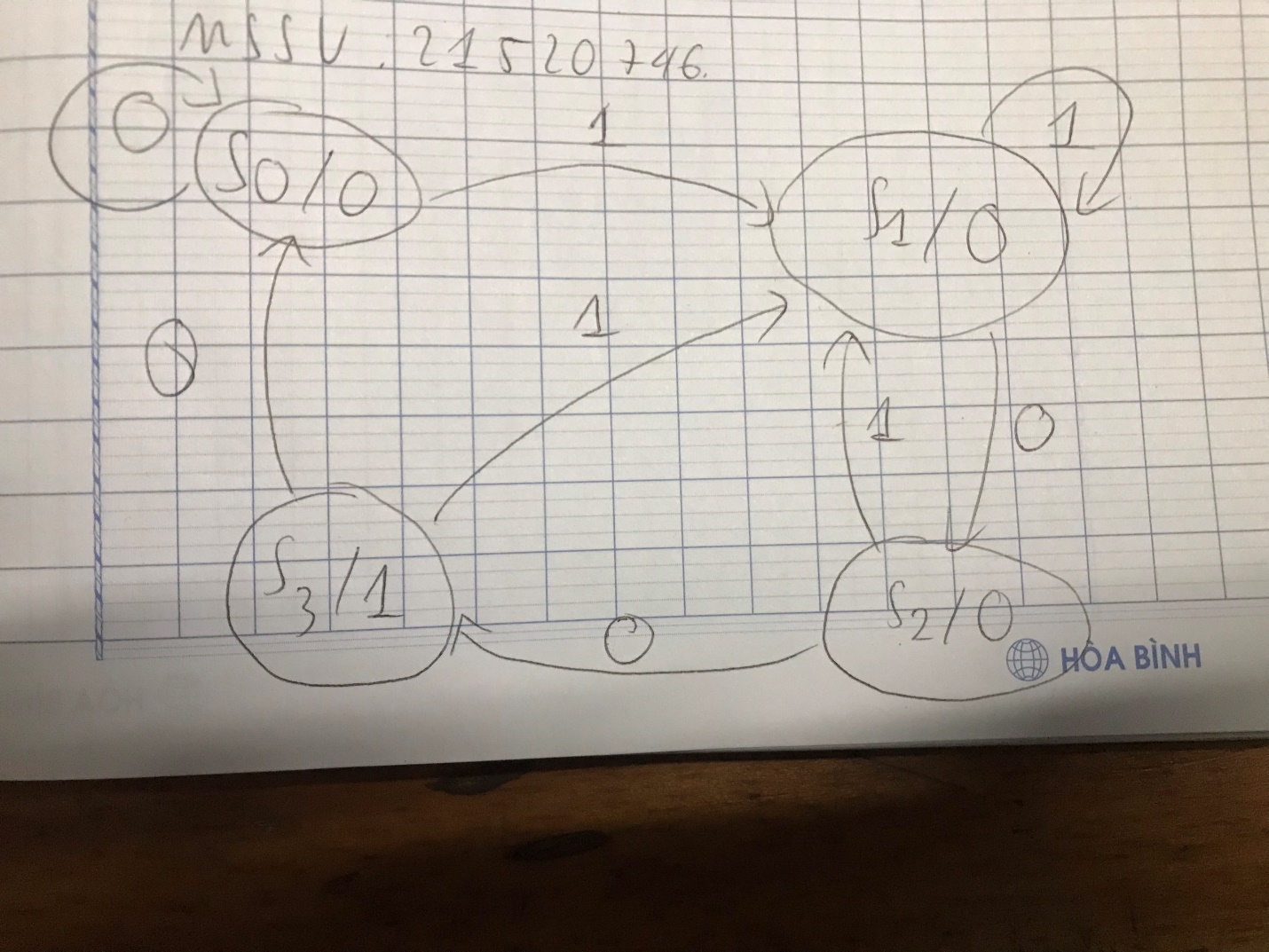
- Nếu số thứ 7 là số lẻ thì chuyển số cuối ấy thành 4-bit, nếu chưa đủ sinh viên thêm các bit 0 vào đầu. (VD: số cuối MSSV là 2 thì dãy cần phát hiện là 0010, số cuối MSSV là 9 thì dãy cần phát hiện là 1001).

- 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

MSSV: 21520746 -> chọn số thứ 8 là 6

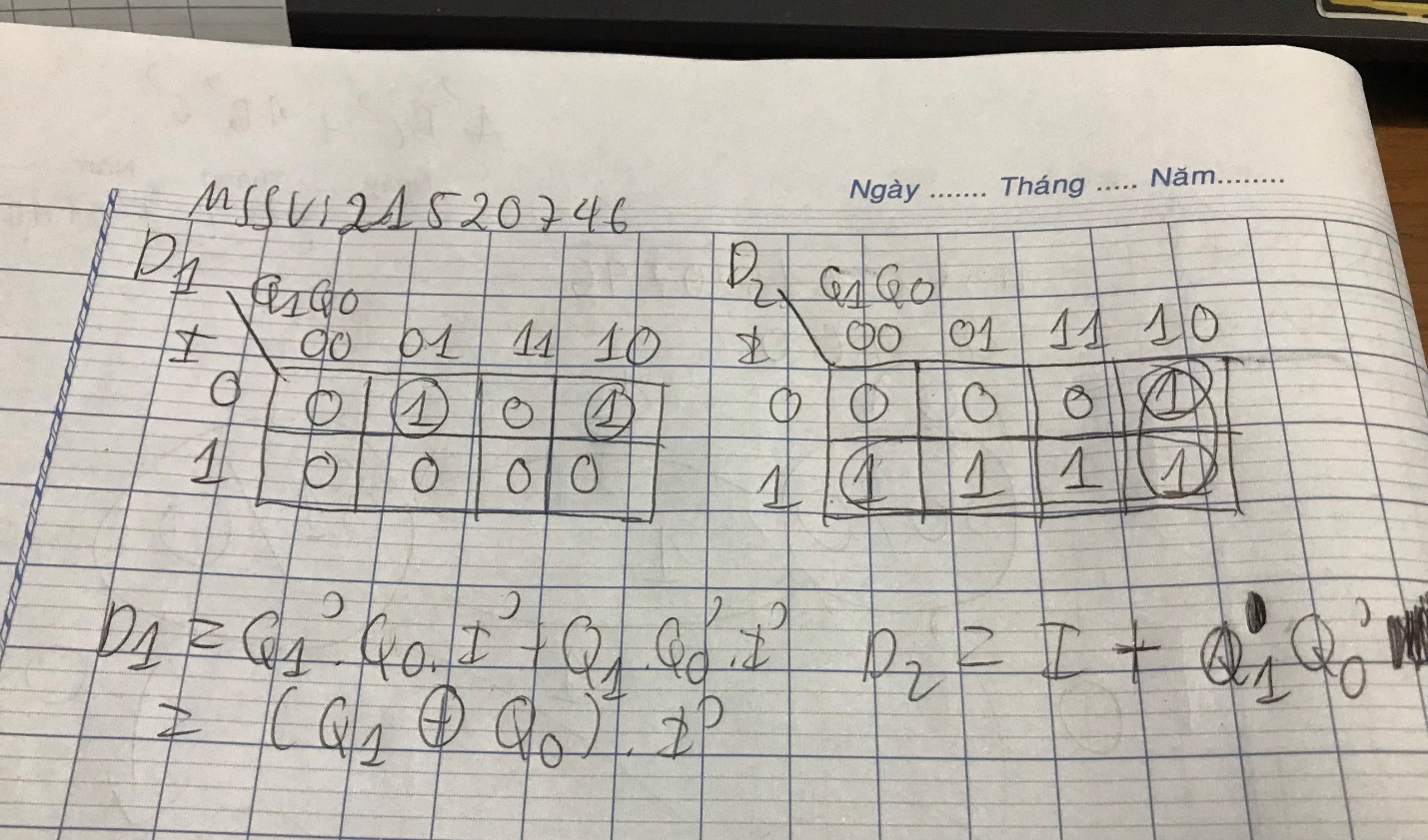
FSM MOORE: 6 - 2 =4 -> 100



Bảng chuyển trạng thái:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Q1 | Q0 | I | D1 | D0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 |

Karnaugh map:



D1 = ( Q1 XOR Q2 ) . I’

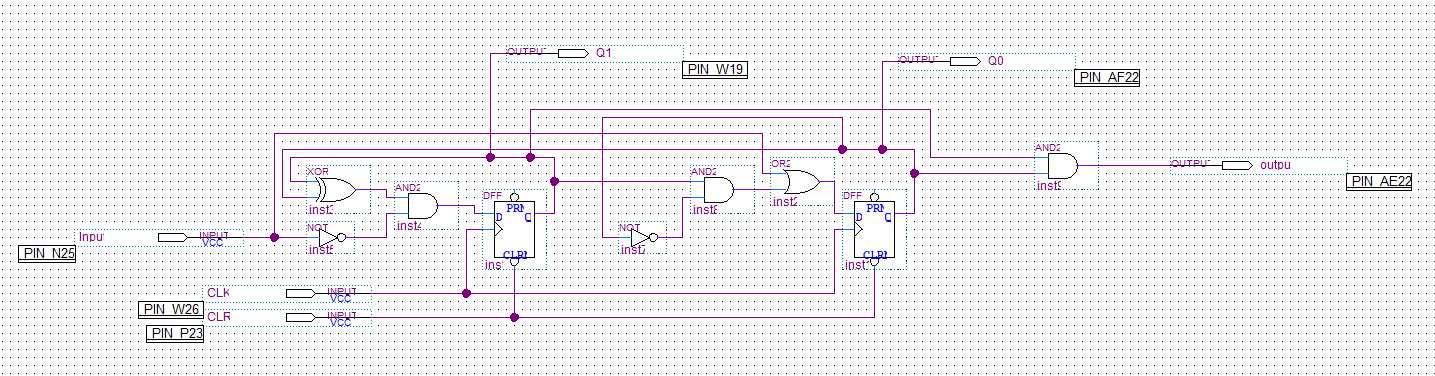
D2 = I + Q1Q0’

Bảng trạng thái Output:

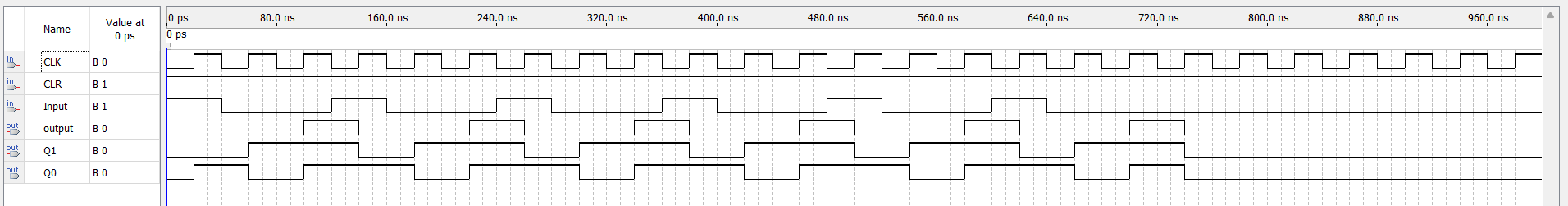
|  |  |  |
| --- | --- | --- |
| Q1 | Q0 | O |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

O = Q1.Q2

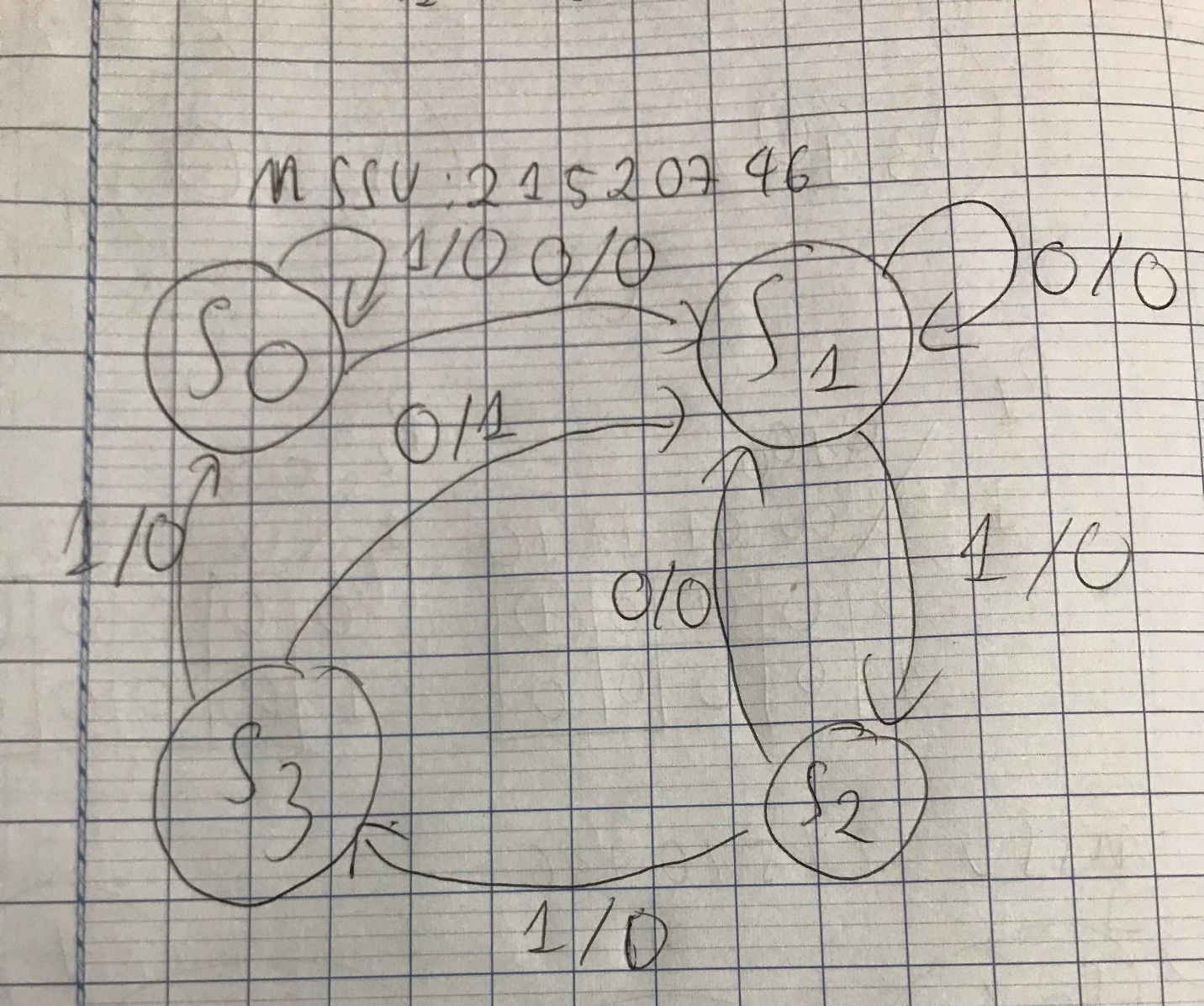
Mạch Quartus:



Chạy waveform:



FSM MEALY: 6 -> 0110

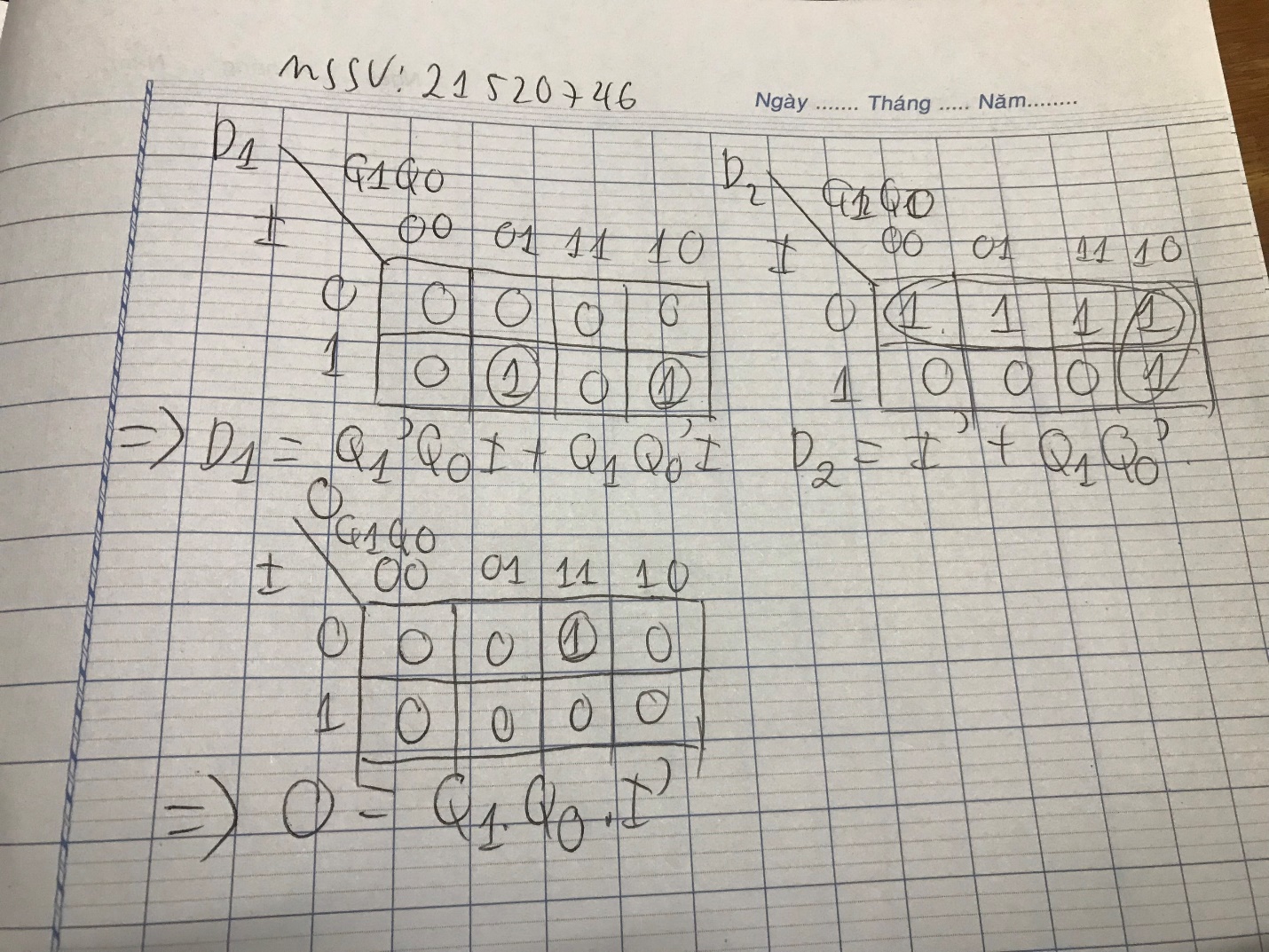


Chu trình: 0->1->1->0->1->1->0…

Bảng chuyển trạng thái:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Q1 | Q0 | I | D1 | D0 | O |
| 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 |

Karnaugh map:

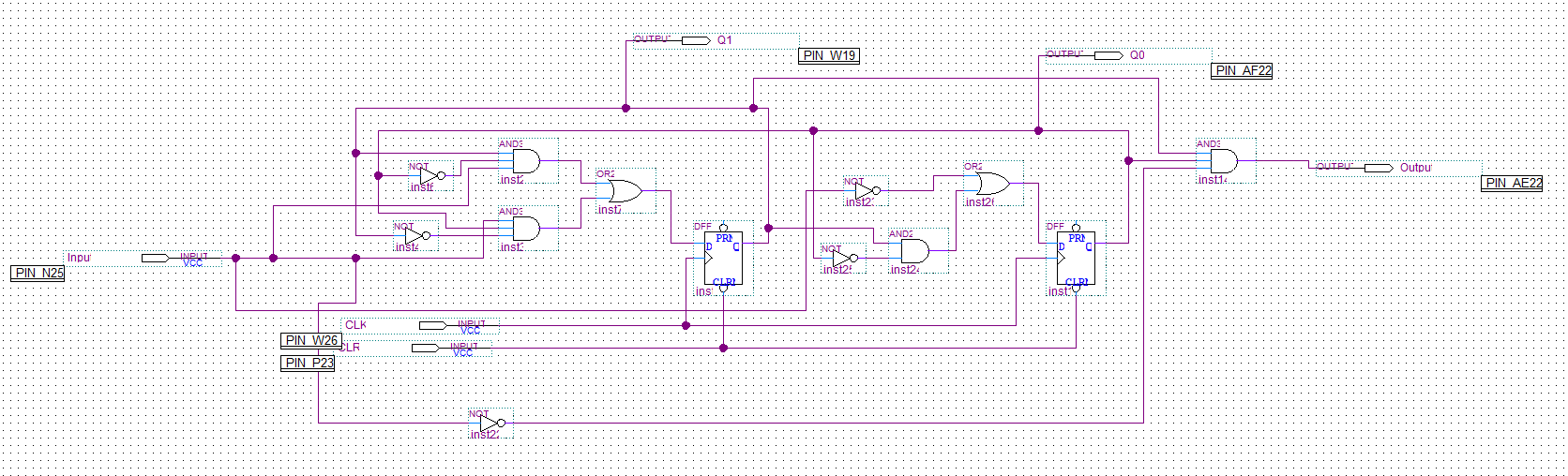


D1 = Q1’.Q0.I + Q1.Q0’.I

D2 = I’ + Q1Q0’

O = Q1.Q0.I’

Mạch Quartus:



Chạy waveform:

