ĐẠI HỌC QUỐC GIA

ĐẠI HỌC BÁCH KHOA TP HỒ CHÍ MINH

BÀI TẬP LỚN

MÔN: KIẾN TRÚC MÁY TÍNH

LỚP: L01 - EE3203

GIẢNG VIÊN HƯỚNG DẪN: TRẦN HOÀNG LINH

Họ và tên	MSSV	Phân công nhiệm vụ
Nguyễn Thanh Phong	**2312626**	- **Phân chia công việc** - **Bảng trạng thái** - **Viết code**
Vũ Tiến Tuấn	**2313774**	- **Bảng trạng thái** - **Viết code** - **Viết báo cáo**
Nguyễn Hoàng Tuấn	**2313746**	- **Phát triển ý tưởng** - **Bảng trạng thái** - **Viết báo cáo**

Thành phố Hồ Chí Minh – 2025

MỤC LỤC

I. MỤC TIÊU

- Hiểu nguyên tắc lập trình
- Ôn tập thiết kế logic cơ bản và FSM
- Thiết kế máy bán hàng sử dụng SystemVerilog

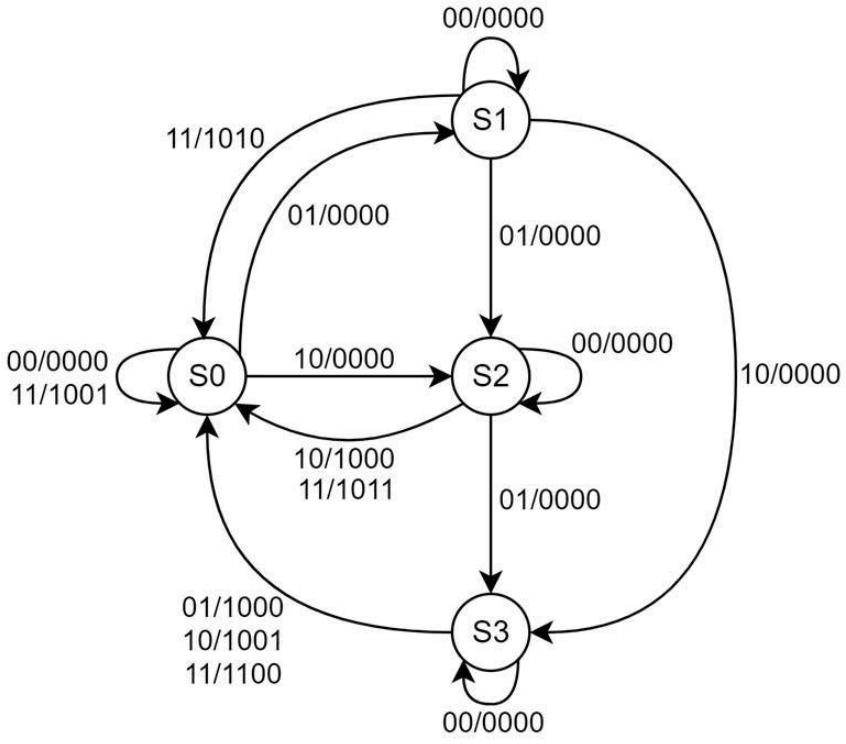
II. TỔNG QUAN

- Ngõ vào: Coin(2 bit) C_1C_0 với 00 01 10 11 -> ¢0 ¢5 ¢10 ¢25
- Sơ đồ trạng thái: 4 trạng thái ứng với 0 5 10 15
- Ngõ ra: o_soda (1 bit), o_change (3 bit)
- Ngõ ra delay 1 chu kì xung nhịp 20ns (f=50MHz): Thực hiện qua 4 Flip-Flop D

III. TRIỂN KHAI

1. Định nghĩa trạng thái và sơ đồ chuyển trạng thái

- S_0: Chờ nhận tiền (Ngõ ra nếu được xuất sẽ bị delay 1 clock sau đó mặc định quay về S_0)
- S_1: Đang có ¢5
- S_2: Đang có ¢10
- S_3: Đang có ¢15



- Mô tả chuyển trạng thái: Bắt đầu tại trạng thái S0, khi nhận các Ngõ vào C1C0 sẽ chuyển sang các trạng thái S1, S2 hoặc S3; nếu số tiền hiện có lớn hơn hoặc bằng ¢20, mặc định chuyển về S0 và xuất o_soda = 1, o_change = phần tiền dư.

- Mã hóa trạng thái:

Trạng thái	Mã hóa
S_0	00
S_1	01
S_2	10
S_3	11

2. Viết bảng trạng thái

Ngõ vào		TTHT		TTKT		Ngõ ra				D-FF	
C_1	C_0	Q_1	Q_0	Q_1^+	Q_0^+	Z_3	Z_2	Z_1	Z_0	D_1	D_0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	1
0	0	1	0	1	0	0	0	0	0	1	0
0	0	1	1	1	1	0	0	0	0	1	1
0	1	0	0	0	1	0	0	0	0	0	1
0	1	0	1	1	0	0	0	0	0	1	0
0	1	1	0	1	1	0	0	0	0	1	1
0	1	1	1	0	0	1	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	1	0
1	0	0	1	1	1	0	0	0	0	1	1
1	0	1	0	0	0	1	0	0	0	0	0
1	0	1	1	0	0	1	0	0	1	0	0
1	1	0	0	0	0	1	0	0	1	0	0
1	1	0	1	0	0	1	0	1	0	0	0
1	1	1	0	0	0	1	0	1	1	0	0
1	1	1	1	0	0	1	1	0	0	0	0

3. Xác định ngõ vào của Flip-Flop và ngõ ra

a. Ngõ vào FF

$D_0 = Q_0{}^+ = \overline{C_1}\,\overline{C_0}Q_0 + \overline{C_1}C_0\overline{Q_0} + \overline{C_0}\,\overline{Q1}Q_0$

$D_1 = Q_1{}^+ = C_1\overline{C_0}\,\overline{Q_1} + \overline{C_1}\,\overline{C_0}Q_1 + \overline{C_1}Q_1\overline{Q_0} + \overline{C_1}C_0\,\overline{Q_1}Q_0$

b. Ngõ ra Z (4 bit biểu diễn)

$$Z = Z_3Z_2Z_1Z_0$$

- *Bit Z_3:o_soda*

$Z_3 = C_1C_0 + C_1Q_1 + C_0Q_1Q_0$

- *3 Bit $Z_2Z_1Z_0$: o_change*

$Z_2 = C_1C_0Q_1Q_0$

$Z_1 = C_1C_0\overline{Q_1}Q_0 + C_1C_0Q_1\overline{Q_0}$

$Z_0 = C_1C_0\overline{Q_0} + C_1\overline{C_0}Q_1Q_0$

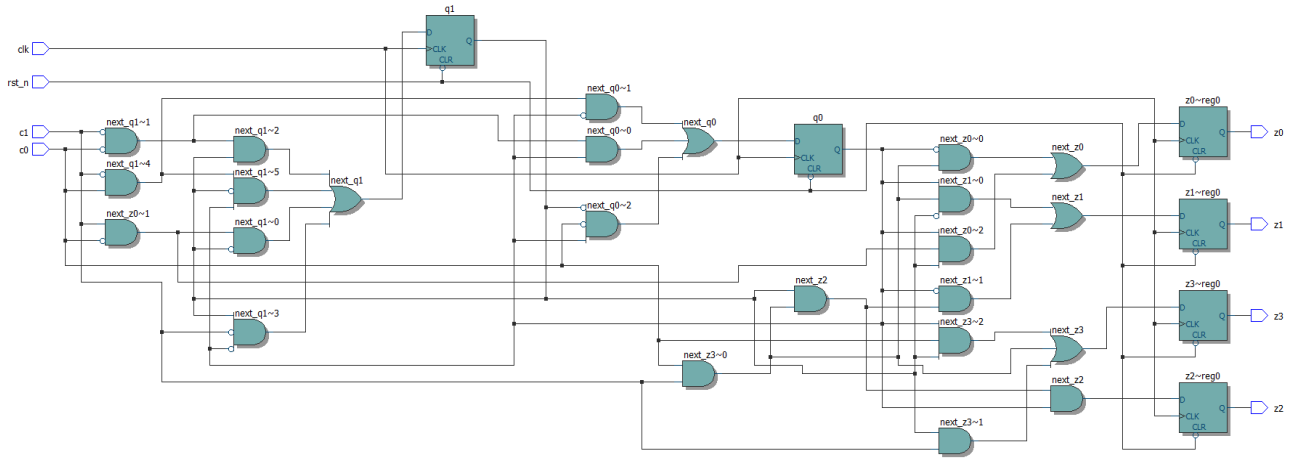
4. Code SystemVerilog

```
module milestone1(
   input  logic clk,
   input  logic rst_n,
   input  logic c1, c0,
   output logic z3, z2, z1, z0
);

   // Các thanh ghi trạng thái
   logic q1, q0;
   logic next_q1, next_q0;

   // Thanh ghi dữ liệu ra (lưu giá trị kế tiếp)
   logic next_z3, next_z2, next_z1, next_z0;

   // Trạng thái kế tiếp
   always_comb begin
      next_q1 = (c1 & ~c0 & ~q1) |
                (~c1 & ~c0 & q1) |
                (~c1 & q1 & ~q0) |
                (~c1 & c0 & ~q1 & q0);

      next_q0 = (~c1 & ~c0 & q0) |
                (~c1 & c0 & ~q0) |
```

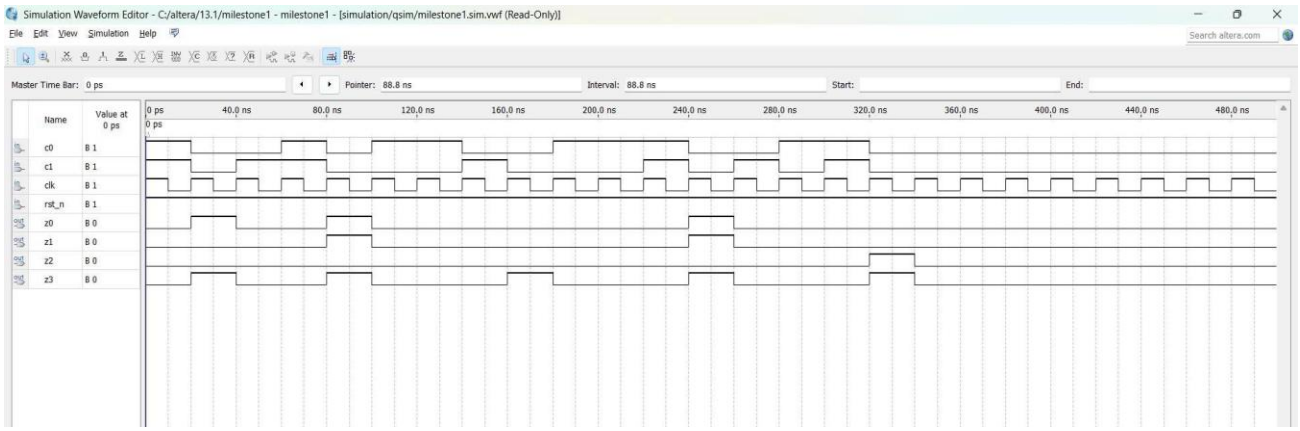
```
            (~c0 & ~q1 & q0);
    end

    // Update ngõ ra z sau 1 chu kỳ
    always_comb begin
        // (Giữ nguyên công thức gốc)
        next_z3 = (c1 & c0) | (c1 & q1) | (c0 & q1 & q0);
        next_z2 = c1 & c0 & q1 & q0;
        next_z1 = (c1 & c0 & ~q1 & q0) | (c1 & c0 & q1 & ~q0);
        next_z0 = (c1 & c0 & ~q0) | (c1 & ~c0 & q1 & q0);
    end

    // Sequential block: update state + registered outputs
    always_ff @(posedge clk or negedge rst_n) begin
        if (!rst_n) begin
            q1 <= 1'b0;
            q0 <= 1'b0;
            z3 <= 1'b0;
            z2 <= 1'b0;
            z1 <= 1'b0;
            z0 <= 1'b0;
        end else begin
            q1 <= next_q1;
            q0 <= next_q0;
            z3 <= next_z3;
            z2 <= next_z2;
            z1 <= next_z1;
            z0 <= next_z0;
        end
    end

endmodule
```

5. Kết quả mô phỏng

- Chọn chu kì xung (T_{clk}) = 20 (ns). Tương đương với tần số (f_{clk}) = 50 (MHz)
- Kết quả RTL Viewer:



- Kết quả mô phỏng dạng sóng (Waveform Simulation):



- Nhận xét:

+ Ở kết quả mô phỏng dạng sóng chúng ta thấy do chân rst_n(reset) tích cực thấp nên chúng ta phải cấp mức logic 1 cho chân rst_n trong quá trình khảo sát thì các ngõ vào và ngõ ra mới hoạt động.

+ Do chúng ta sử dụng thêm 4 D_FF ở ngõ ra để delay 1 chu kì xung nhịp, điều này đóng vai trò như một trạng thái đệm. Khi số lượng tiền (coin) ở ngõ vào đã cung cấp đủ 20 cent thì phải đến chu kì xung nhịp tiếp theo trạng thái ở ngõ ra Z mới bắt đầu xuất tín hiệu ra.

IV. TỔNG KẾT

Ở milestone 1, nhóm đã đáp ứng đủ các mục tiêu. Nhóm chúng em đã cùng nhau xây dựng các bước để thiết kế FSM của máy bán hàng tự động. Qua đó, nhóm hiểu đã hiểu được nguyên tắc lập trình FSM, ôn tập lại cách thiết kế sơ đồ trạng thái cũng như bảng trạng thái, từ đó xây dựng và chuyển sang code SystemVerilog.