THIẾT KẾ HỆ THỐNG SỐ VỚI HDL
BÀI THỰC HÀNH 3

GVHD: Tạ Trí Đức

Sinh viên thực hiện: Phạm Quốc Tiến – 22521472

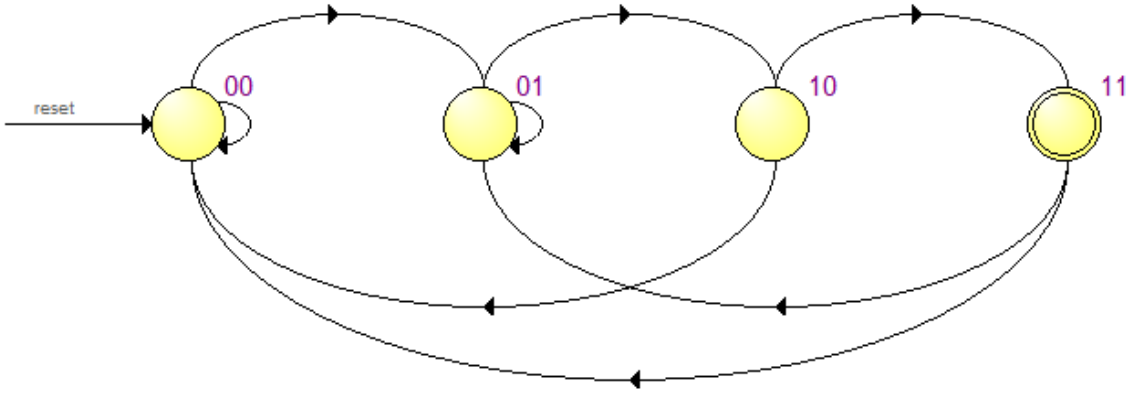
1) Thiết kế mạch tuần tự

a. Kiểu Moore

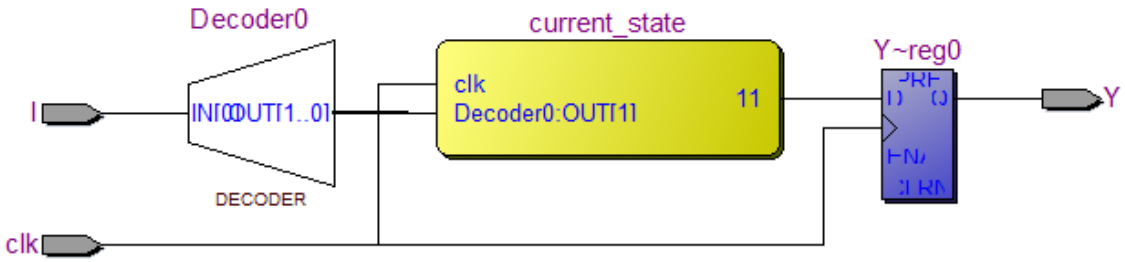
Code:

```
1  module moore(input I, clk, output reg Y);
2  reg [1:0] current_state;
3  always @(posedge clk)
4  case(current_state)
5     0 : begin
6              case(I)
7                 0: current_state <= 1;
8                 1: current_state <= 0;
9              endcase
10             Y <= 0;
11          end
12    1 : begin
13             case(I)
14                0: current_state <= 1;
15                1: current_state <= 2;
16             endcase
17             Y <= 0;
18          end
19    2 : begin
20             case(I)
21                0: current_state <= 3;
22                1: current_state <= 0;
23             endcase
24             Y <= 0;
25          end
26    3 : begin
27             case(I)
28                0: current_state <= 1;
29                1: current_state <= 0;
30             endcase
31             Y <= 1;
32          end
33 endcase
34 endmodule
```

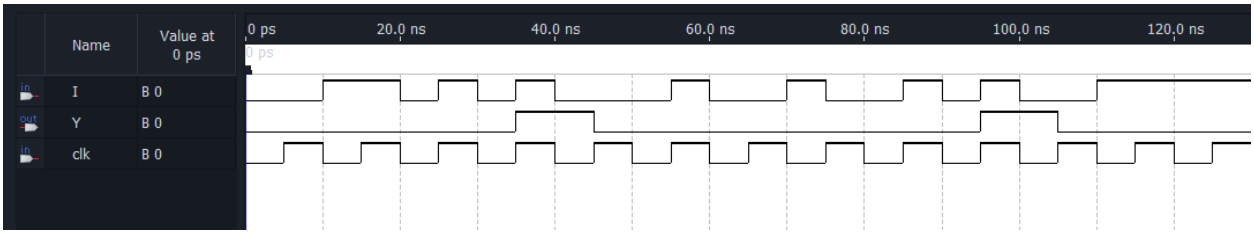
Sate machine:



RTL:



Waveform test function:

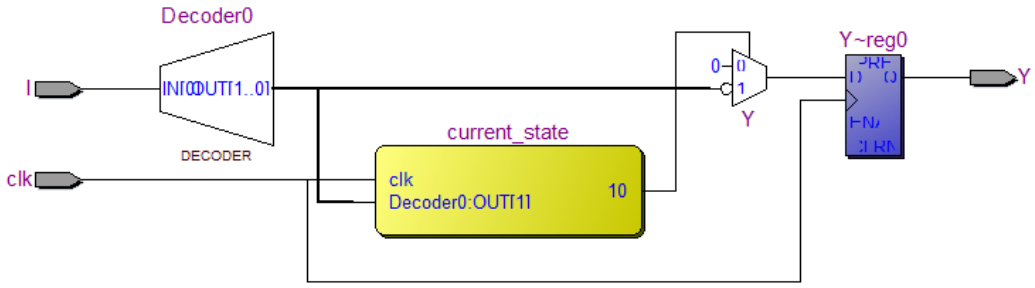


b. Kiểu Mealy

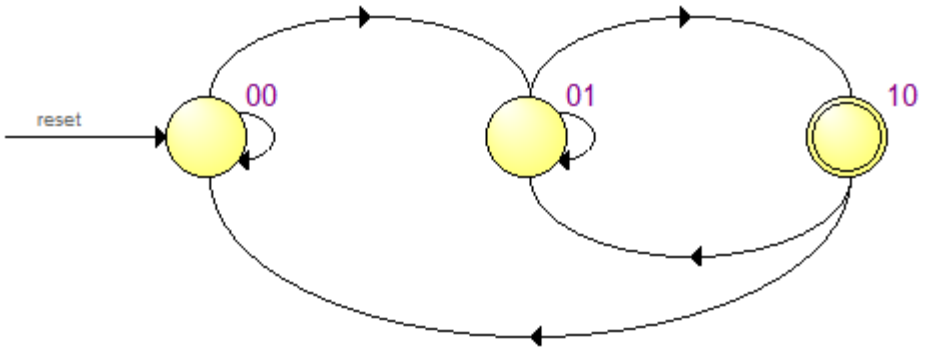
Code:

```
1  module mealy(input I, clk, output reg Y);
2  reg [1:0] current_state;
3  always @(posedge clk)
4  case(current_state)
5     0 : begin
6              case(I)
7                 0: begin current_state <= 1; Y <= 0; end
8                 1: begin Y <= 0; end
9              endcase
10    end
11    1 : begin
12             case(I)
13                0: begin current_state <= 1; Y <= 0; end
14                1: begin current_state <= 2; Y <= 0; end
15             endcase
16    end
17    2 : begin
18             case(I)
19                0: begin current_state <= 1; Y <= 1; end
20                1: begin current_state <= 0; Y <= 0; end
21             endcase
22    end
23 endcase
24 endmodule
25
```

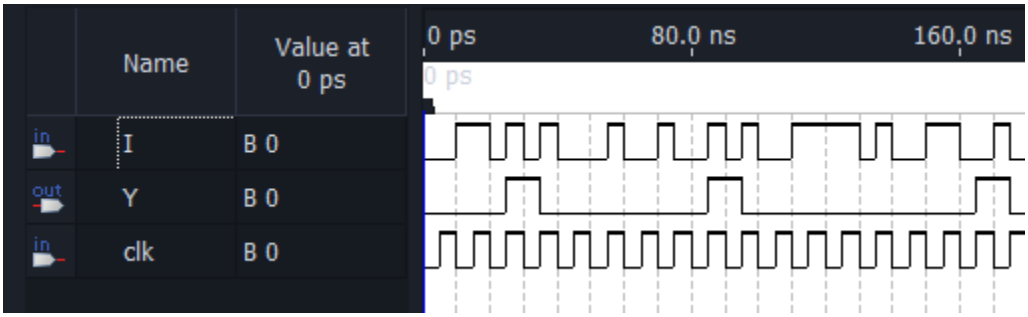
RTL:



State machine:



Waveform test function:



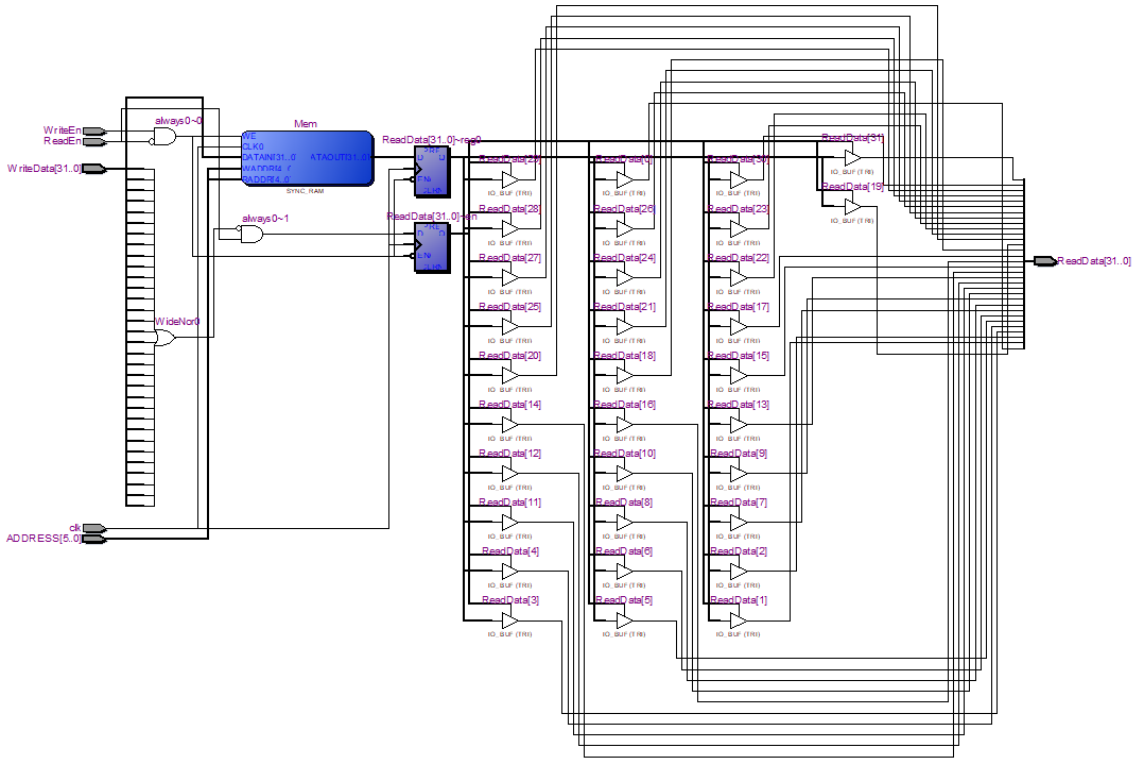
2) Thiết kế bộ nhớ SRAM

Code:

```
module SRAM(ADDRESS, WriteData, ReadData, WriteEn, ReadEn, clk);
input [5:0] ADDRESS;
input [31:0] WriteData;
reg [31:0] Mem [0:31];
output reg [31:0] ReadData;
input WriteEn, ReadEn, clk;

always @(posedge clk)
   begin
      if(WriteEn && !ReadEn)
         Mem[ADDRESS] <= WriteData;
      else if(ReadEn && !WriteData)
         ReadData <= Mem[ADDRESS];
      else
         ReadData <= 32'bzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz;
   end
endmodule
```

RTL:



Waveform test function: