BÀI 7

MÁY TRẠNG THÁI HỮU HẠN

Mục tiêu:

- Thực hành lập trình ứng dụng trên máy tính, biên dịch chương trình, nạp chương trình vào FPGA và sử dụng KIT DE2 để kiểm chứng.
- Điều khiển thiết bị ngoại vi bằng các port của FPGA
- Mô tả máy trạng thái hữu hạn dùng ngôn ngữ veilog HDL

Yêu cầu:

- Nắm vững cách viết chương trình dùng ngôn ngữ Verilog HDL
- Nắm vững cách kết nối máy tính với KIT DE2
- Sử dụng thành thục phần mền Quatus để viết chương trình, biên dịch chương trình, nạp chương trình vào FPGA
- Nắm vững cấu trúc phần cứng của công tắc, nút nhấn, led đơn, led bảy đoạn trên KIT DE2
- Sinh viên làm theo các bước hướng dẫn trong các bài thực hành

Nội dung:

I. Thiết bị và vật dụng sử dụng:

- Máy vi tính đã cài đặt phần mềm Quatus và driver USB blaster

- KIT DE2, sử dụng chip CycloneII EP2C35F672C6

- Cáp USB blaster kết nối KIT DE2 và máy tính

- DC adapter 12V/2A cấp nguồn cho KIT DE2

II. Lý thuyết

1 Giới thiệu chung về máy trạng thái hữu hạn (FSM):

- FSM là một công cụ dùng để mô tả các hành vi của hệ thống số. Nó chỉ ra tất cả các trạng thái có thể có của hệ thống, và những sự kiện nào sẽ gây ra sự thay đổi trạng thái.

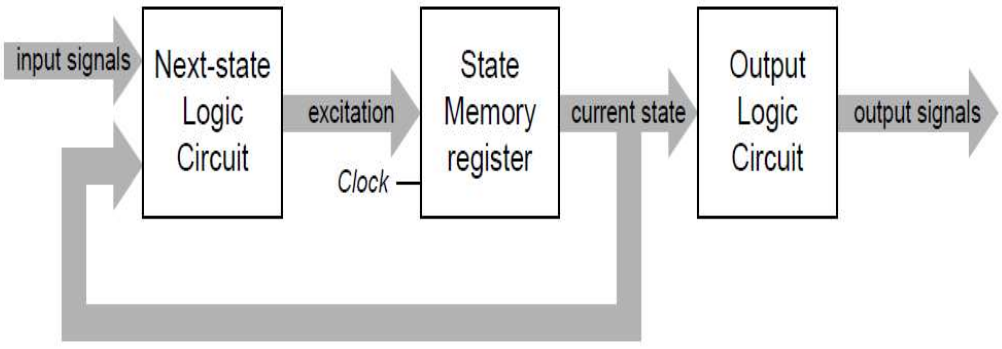
- Thành phần cơ bản của FSM là: xung nhịp ck, khối thanh ghi, khối tác động trạng thái kế tiếp, khối tác động ngõ ra

- Số ngõ vào/ra của FSM tùy thuộc vào yêu cầu mạch thiết kế

- Hoạt động của FSM thường được mô tả bằng lưu đồ trạng thái Moore hoặc Mealy

2 Thiết kế FSM theo lưu đồ MOORE

a. Sơ đồ khối



Hình 7. 1: Sơ đồ khối tổng quát của FSM dạng MOORE

Next state logic circuit: khối tác động trạng thái kế tiếp

State memory register: khối thanh ghi (lưu trạng thái trong của FSM)

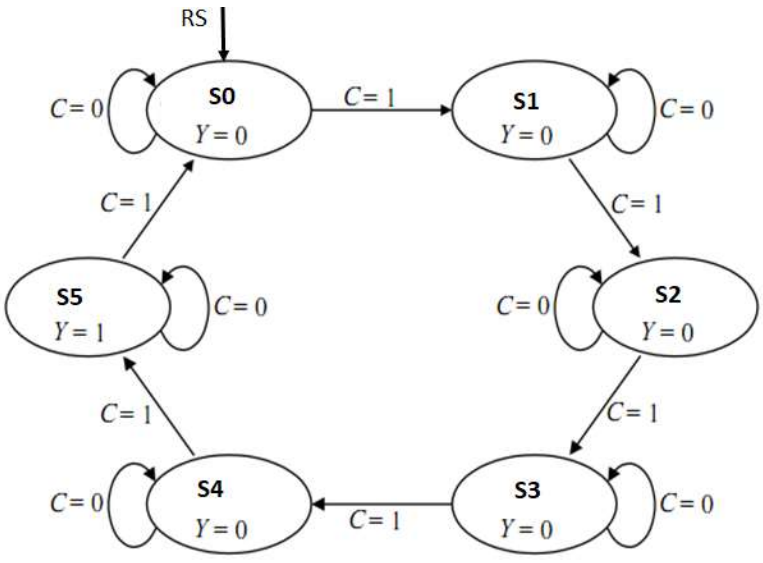
Output logic circuit: khối tác động ngõ ra

b. Các bước thiết kế FSM dạng moore

- Dựa vào yêu cầu thiết kế hoặc lưu đồ trạng thái xác định số ngõ vào, số ngõ ra, và số trạng thái trong của FSM
- Chọn loại và số lượng flip flop để lưu trạng thái trong của FSM
- Lập bảng chuyển đổi trạng thái
- Rút gọn và viết biểu thức của các ngõ vào của FF và ngõ ra của FSM *theo trạng thái hiện tại.*
- Vẽ mạch thiết kế (nếu cần)
- Viết code theo mô hình hành vi

c. Ví dụ:

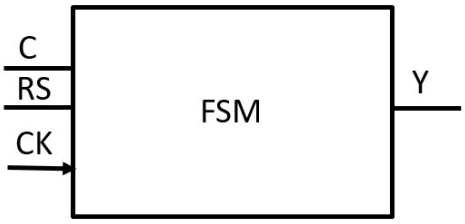
Mô tả FSM theo giản đồ trạng thái dạng Moore



Hình 7. 2: giản đồ moore mô tả FSM

Bài giải:

- Sơ đồ chân của FSM



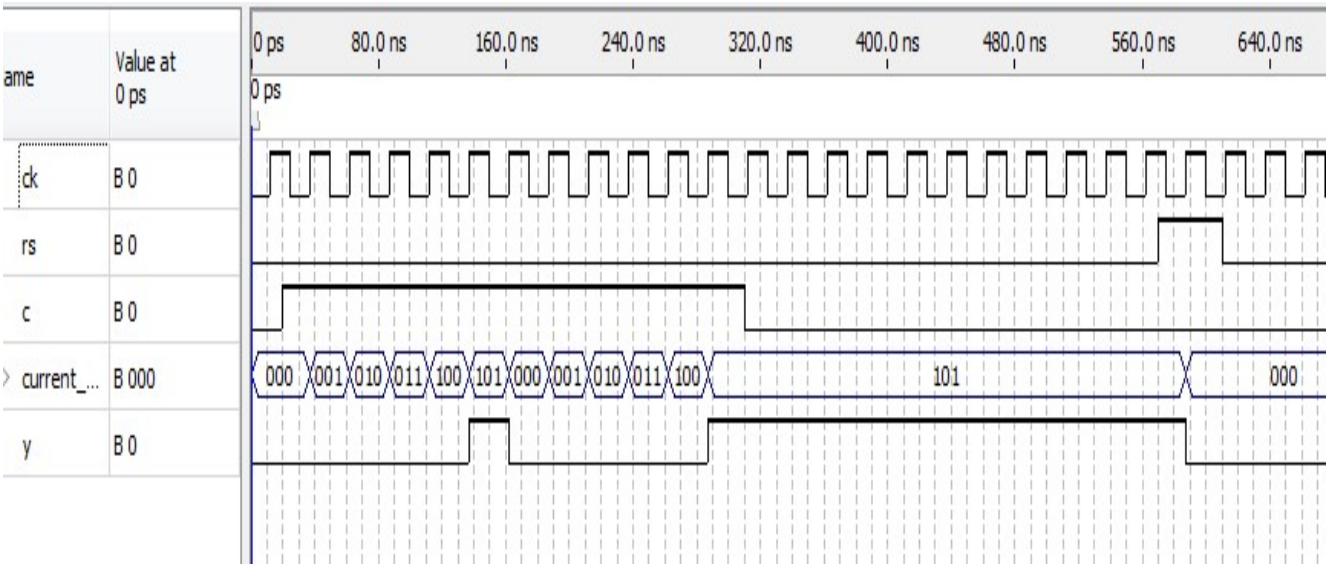
Hình 7. 3: Sơ đồ chân input/output của FSM

- Viết code mô tả phần cứng:

```
module fsm_VD1 (ck,rs,c,y);
parameter s0 = 3'b000;
parameter s1 = 3'b001;
parameter s2 = 3'b010;
parameter s3 = 3’b011;
parameter s4 = 3'b100;
parameter s5 = 3'b101;
input ck,rs,c;
output y;
reg [2:0]current_state, next_state;
reg y;
always @ (*)
begin
case (current_state)
  s0 : if (c) next_state = s1;
     else next_state = current_state;
  s1 :  if (c) next_state = s2;
      else next_state = current_state;
  s2 :  if (c) next_state = s3;
```

```
        else next_state = current_state;
    s3 :  if (c) next_state = s4;
        else next_state = current_state;
    s4 :  if (c) next_state = s5;
        else next_state = current_state;
    s5 :  if (c) next_state = s0;
       else next_state = current_state;
   default next_state = s0;
  endcase
  end
  always @ (posedge ck)
  begin
    if (rs == 1) current_state <= s0;
    else current_state <= next_state;
  End
  always @ (*)
  begin
     if (current_state == s5) y = 1'b1;
     else y = 1'b0;
  end
  Endmodule
```

- Kết quả mô phỏng:



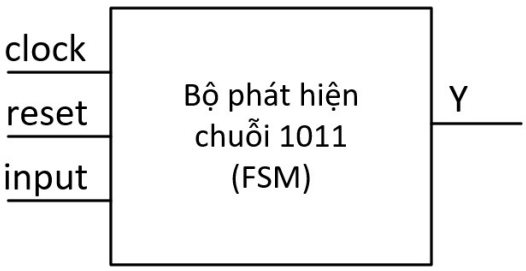
Hình 7. 4: kết quả mô phỏng FSM

III. Phần thực hành

Bài tập 1: Mô tả máy trạng thái hữu hạn (FSM) có chức năng nhận biết chuỗi 1011 từ ngõ vào input. Chân Ck tác động theo cạnh lên, chân RS không đồng bộ và tích cực mức cao. Khi phát hiện đúng chuỗi 1011 thì ngõ ra Y sẽ bằng 1.

Hướng dẫn thực hiện:

B1: Xác định số ngõ vào, ngõ ra



Hình 7. 5: Sơ đồ chân input/output bài tập 1

Gán chân:

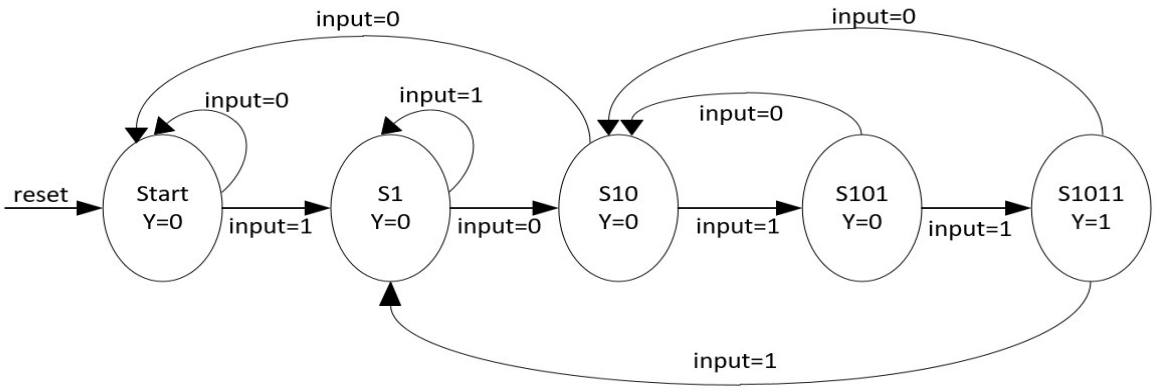
clock : SW[0]

reset : SW[1]

input : SW[2]

y : LEDR[0]

B2: Graph trạng thái dạng moore



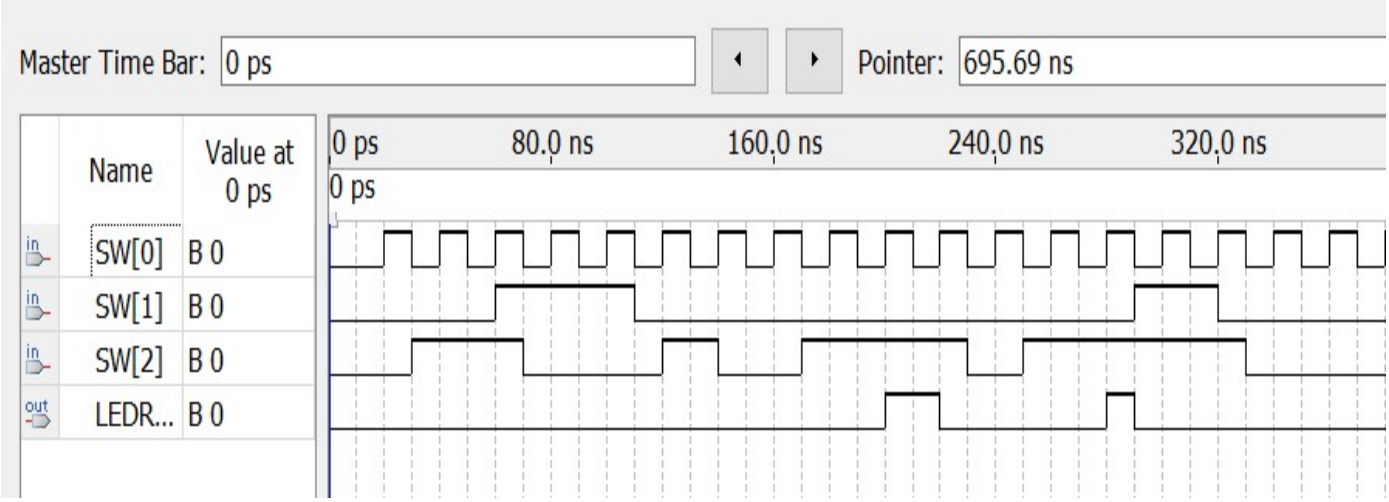
Hình 7. 6: Giản đồ trạng thái bài tập 1

B3: Viết chương trình dùng ngôn ngữ Verilog

```
module bai1 (SW,LEDR);
input [2:0]SW;
output reg [0:0]LEDR;
parameter start      = 3'b000;
parameter s1         = 3'b001;
parameter s10        = 3'b010;
parameter s101       = 3'b011;
parameter s1011      = 3'b100;
reg [2:0]current_state, next_state;
//mach to hop tac dong ngo vao cho FF cua fsm
always @ (*)
begin
        case (current_state)
        start :  if (SW[2]) next_state = s1;
                 else next_state = current_state;
        s1    :  if (~SW[2]) next_state = s10;
                 else next_state = current_state;
        s10   :  if (SW[2]) next_state = s101;
                 else next_state = start;
        s101  :  if (SW[2]) next_state = s1011;
                 else next_state = s10;
        s1011 :  if (SW[2]) next_state = s1;
                 else next_state = s10;
        default next_state = start;
        endcase
end
//mach tuan tu tao trang thai trong cho  fsm
always @ (posedge SW[0] or posedge SW[1])
begin
        if (SW[1]) current_state <= start;
        else current_state <= next_state;
end
//mach to hop tac dong trang thai ngo ra fsm
always @ (*)
begin
if (current_state == s1011) LEDR[0] = 1'b1;
else LEDR[0] = 1'b0;
end
endmodule
```

B4 : Biên dịch để phân tích, tổng hợp và tạo ra file .sof

B5 : Mô phỏng và kiểm tra kết quả



Hình 7. 7: Kết quả mô phỏng bài tập 1

B6 : Gán chân cho FPGA

B7 : Nạp file.sof vào FPGA, kiểm tra hoạt động của mạch.

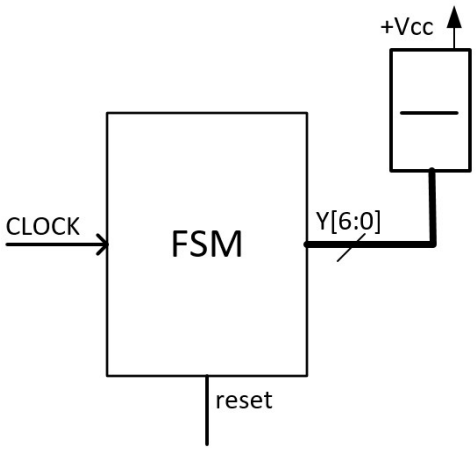
Bài tập 2: Mô tả máy trạng thái hữu hạn (FSM) có chức năng nhận biết chuỗi 0110 từ ngõ vào input. Chân Ck tác động theo cạnh xuống, chân RS không đồng bộ và tích cực mức thấp. Khi phát hiện đúng chuỗi 1011 thì ngõ ra Y sẽ bằng 1.

Bài tập 3: Mô tả máy trạng thái hữu hạn (FSM) có chức năng nhận biết chuỗi 001 và 110 từ ngõ vào input. Chân Ck tác động theo cạnh xuống, chân RS đồng bộ và tích cực mức cao. Khi phát hiện đúng chuỗi 001 thì ngõ ra Y1 sẽ bằng 1, khi phát hiện đúng chuỗi 110 thì ngõ ra Y2 sẽ bằng 1.

Bài tập 4: Mô tả máy trạng thái hữu hạn (FSM) thực hiện mạch đếm xuống 5 về 0, ngõ ra hiển thị trên 1 led bảy đoạn loại anode chung. Chân Ck tác động theo cạnh lên, chân RS không đồng bộ và tích cực mức cao, Thời gian hiển thị một chữ số là 1s, khi RS tích cực thì các led hiển thị số 5

Hướng dẫn thực hiện:

B1: Xác định số ngõ vào, ngõ ra



Gán chân:

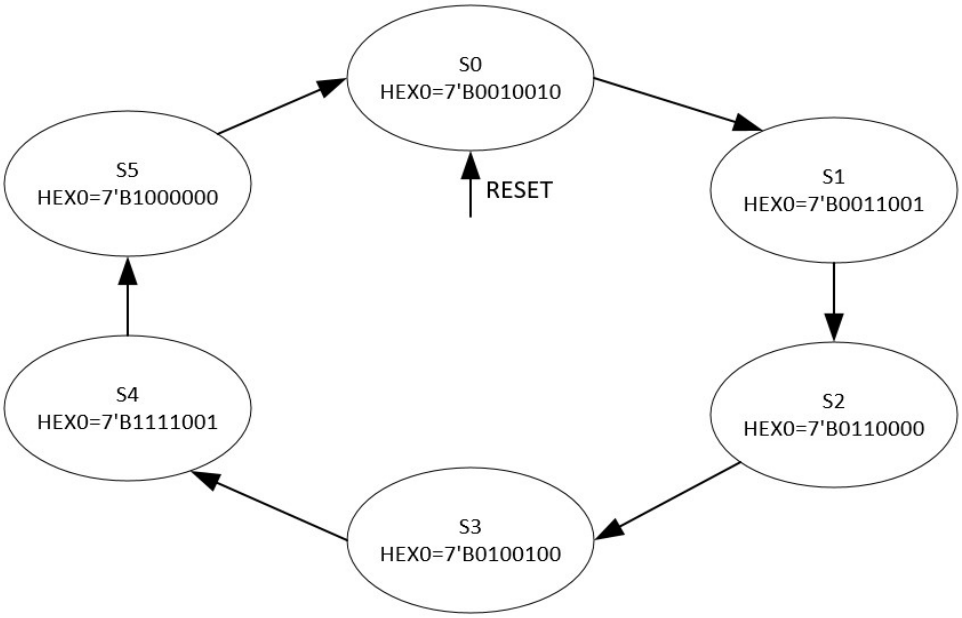
clock : CLOCK_50

reset : SW[0]

Y[6:0] : HEX0[6:0]

Hình 7. 8: : Sơ đồ chân input/output bài tập 4

B2: Graph trạng thái dạng moore



Hình 7. 9: Giản đồ trạng thái bài tập 4

B3: Viết chương trình dùng ngôn ngữ Verilog

```
/RESET: SW0,   CK: CLOCK_50,   Y: HEX0
module bai1 (SW,CLOCK_50,HEX0);
input [0:0]SW;
input CLOCK_50;
output reg [6:0]HEX0;
parameter s0  = 3'b000;
                parameter s1  = 3'b001;
                parameter s2  = 3'b010;
                parameter s3  = 3'b011;
                parameter s4  = 3'b100;
                parameter s5  = 3'b101;
                reg [2:0]current_state, next_state;
                integer q;
                reg clock_1s = 1'b0;
                always @ (posedge CLOCK_50)
                        begin
                                q=q+1;
                                if (q == 49999999)
                                begin
                                clock_1s = ~clock_1s;
                                q = 0;
                                end
                        end
                //mach to hop tac dong ngo vao cho FF cua fsm
                always @ (*)
                begin
                        case (current_state)
                        s0 :    next_state = s1;
                        s1 :    next_state = s2;
                        s2 :    next_state = s3;
                        s3 :    next_state = s4;
                        s4 :    next_state = s5;
                        s5 :    next_state = s0;
                        default next_state = s0;
                        endcase
                                end
                                //mach tuan tu tao trang thai trong cho  fsm
                                always @ (posedge clock_1s or posedge SW[0])
                                begin
                                        if (SW[0]) current_state <= s0;
                                        else current_state <= next_state;
```

```
end
//mach to hop tac dong trang thai ngo ra fsm
always @ (*)
begin
        case (current_state)
        s0: HEX0 = 7'b0010010;
        s1: HEX0 = 7'b0011001;
        s2: HEX0 = 7'b0110000;
        s3: HEX0 = 7'b0100100;
        s4: HEX0 = 7'b1111001;
        s5: HEX0 = 7'b1000000;
        endcase
end
endmodule
```

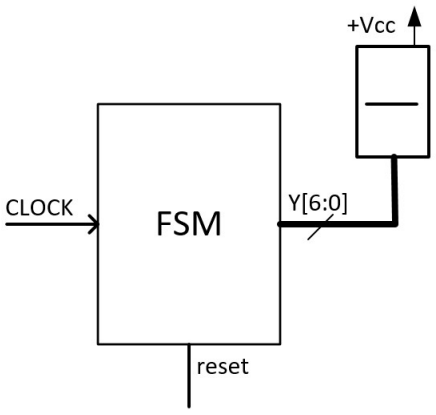
B4 : Biên dịch để phân tích, tổng hợp và tạo ra file .sof

B5 : Gán chân cho FPGA

B6 : Nạp file.sof vào FPGA, kiểm tra hoạt động của mạch.

Bài tập 5: Lập trình điều khiển một led bảy đoạn anode chung, hiển thị lần lượt các ký tự trong chuỗi "HELLO". Chân Ck tác động theo cạnh lên, chân RS không đồng bộ và tích cực mức cao, thời gian hiển thị một ký tự là 2s, khi RS tích cực thì các led tắt.

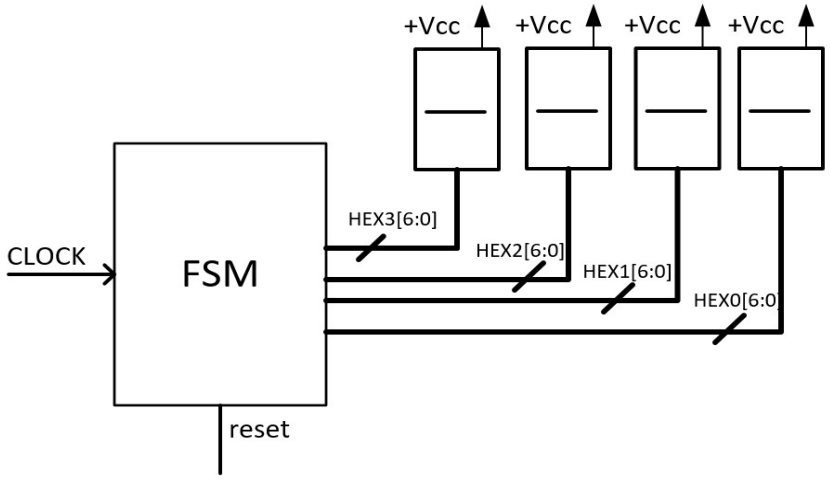
Sơ đồ phần cứng như hình sau:



Hình 7. 10: Sơ đồ chân input/output bài tập 5

Bài tập 6: Lập trình FSM điều khiển 4 led bảy đoạn anode chung, hiển thị chữ "BABY" chớp tắt với tần số 1 Hz. Chân Ck tác động theo cạnh lên, chân RS không đồng bộ và tích cực mức cao, khi RS tích cực thì các led tắt hết.

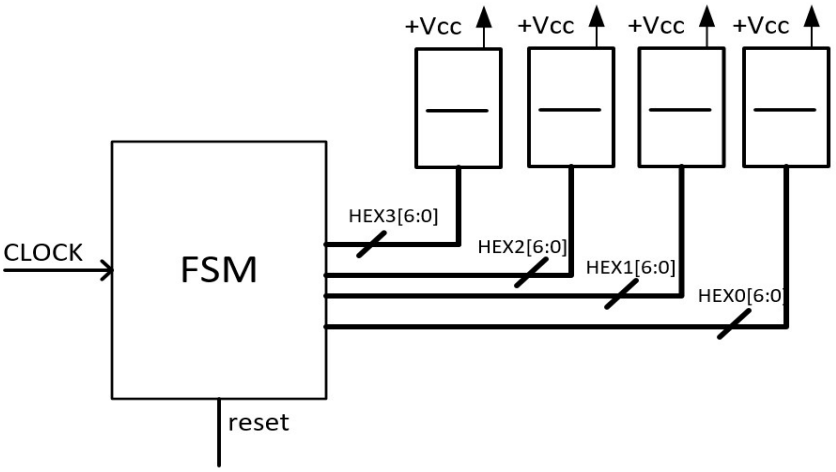
Sơ đồ phần cứng như hình sau:



Hình 7. 11: Sơ đồ chân input/output bài tập 6

Bài tập 7: Lập trình FSM điều khiển 4 led bảy đoạn anode chung, hiển thị chuỗi "FPGA" dịch từ trái qua phải. Chân Ck tác động theo cạnh lên, chân RS không đồng bộ và tích cực mức cao, khi RS tích cực thì các led tắt hết.

Sơ đồ phần cứng như hình sau:



Hình 7. 12: Sơ đồ chân input/output bài tập 7

Bài tập 8: Lập trình FSM điều khiển 8 led bảy đoạn anode chung, hiển thị chớp tắt "ngày-tháng – năm" của ngày đang lập trình. Chân Ck tác động theo cạnh lên, chân RS không đồng bộ và tích cực mức cao, khi RS tích cực thì các led tắt hết.

Bài tập 9: Lập trình FSM điều khiển 8 led bảy đoạn anode chung, hiển thị chuỗi "DE2 - FPGA" dịch từ phải qua trái. Chân Ck tác động theo cạnh lên, chân RS không đồng bộ và tích cực mức cao.

Bài tập 10: Lập trình FSM điều khiển led theo yêu cầu sau:

Tám led bảy đoạn anode chung (HEX7 → HEX0), hiển thị chuỗi "DE2 - FPGA" chớp tắt với tần số 1Hz , đồng thời 18 led đỏ (LEDR[17:0]) sáng dần từ hai biên vào với thời gian tồn tại 1 trạng thái là 200ms .

Chân Ck tác động theo cạnh lên, chân RS không đồng bộ và tích cực mức cao, khi RS tích cực thì các led tắt hết.