

**Nguyễn Vũ Lục Lam**

**21200305**

## **BÁO CÁO ĐỒ ÁN THỰC HÀNH THIẾT KẾ LOGIC KHẢ TRÌNH**

**Câu 1: Tìm hiểu và thiết kế mạch tổ hợp như hình ở dưới, sử dụng ngôn ngữ Verilog HDL và lập bảng chân trị (truth table) phù hợp cho mạch này. Cho biết biết  $S0 \rightarrow S2$  và  $A0 \rightarrow A7$  là các ngõ vào, và Y ngõ ra. Hãy cho biết đây là mạch gì và chức năng của nó?**

**Thực hiện mạch trên board FPGA DE10 với S0, S1, S2 được gán với KEY0, KEY1, và KEY2,  $A0 \rightarrow A7$  được gán với SW0  $\rightarrow$  SW7, và Y được gán với LED0.**

### **BÀI LÀM**

KEY2 KEY1 KEY0 tương ứng với 3 bit gồm 7 trạng thái là từ 0 đến 7

Còn A0 đến A7 là tương ứng từ số 0 đến số 7

Nếu A0 thì có nghĩa là k bấm KEY nào cả là 000

A1 thì bấm KEY: 001 số 1

A2 thì bấm KEY: 010 số 2

A3 thì bấm KEY: 011 số 2

Tương tự cho các cái sau

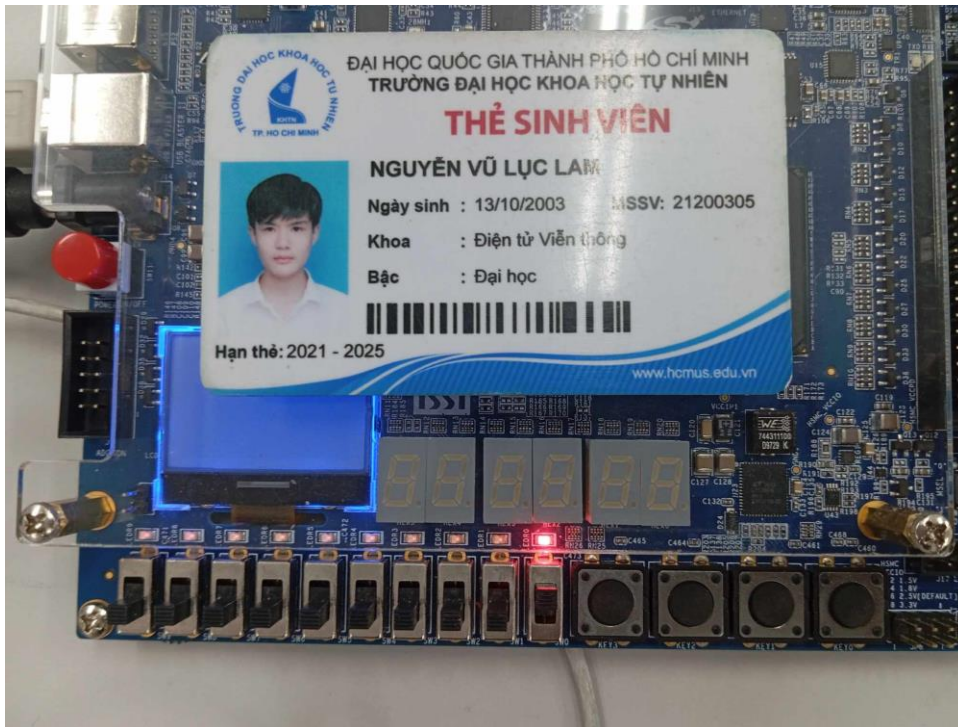
Kết luận mạch này là mạch đa hợp có công dụng là dùng 1 logic bất kỳ thay vì sử dụng cổng logic

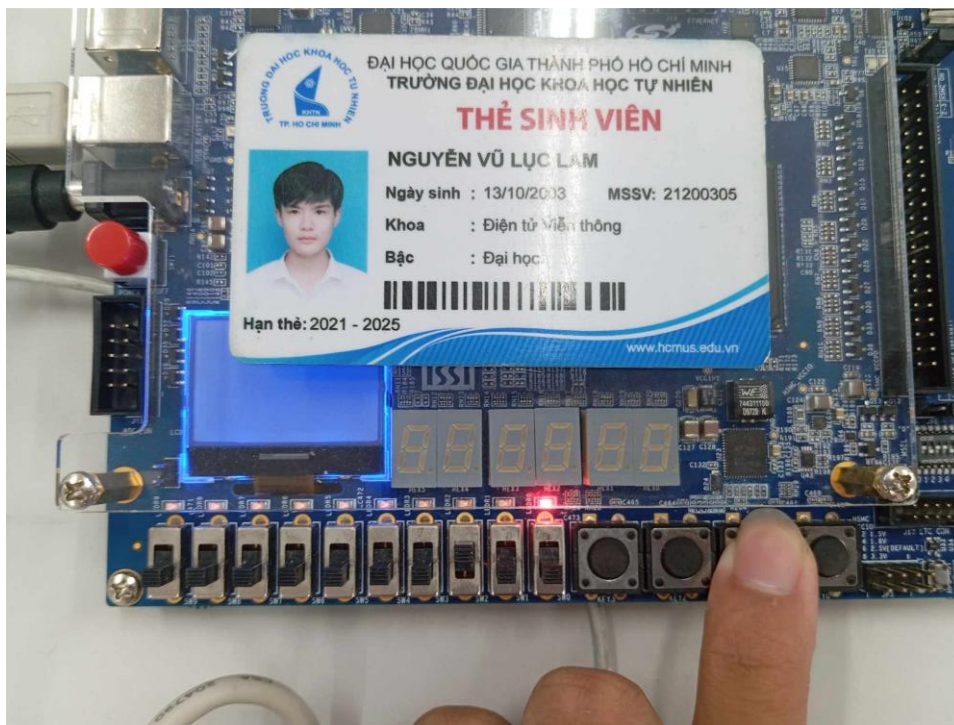
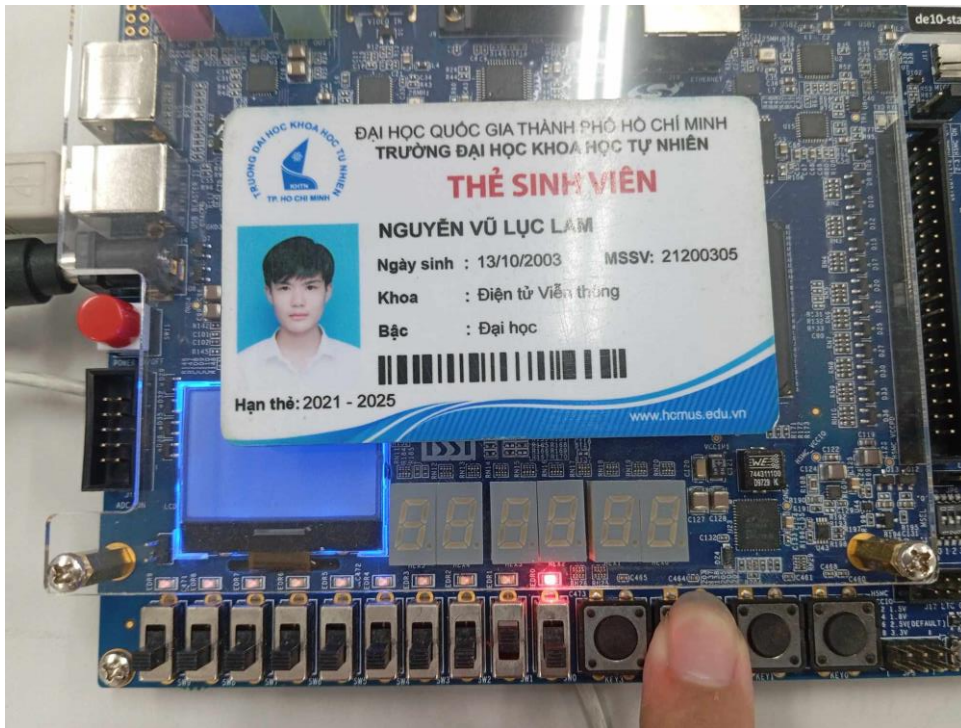
module Bai1(input [2:0] S, input [7:0] A, output Y); //Khai báo module bai1 input gồm S có 3 bit và A có 8 bit

```
assign Y = A[0]& (S[0])& (S[1])& (S[2])
          |A[1]& (S[0])& (S[1])& (~S[2])
          |A[2]& (S[0])& (~S[1])& (S[2])
          |A[3]& (S[0])& (~S[1])& (~S[2])
          |A[4]& (~S[0])& (S[1])& (S[2])
          |A[5]& (~S[0])& (S[1])& (~S[2])
          |A[6]& (~S[0])& (~S[1])& (S[2])
          |A[7]& (~S[0])& (~S[1])& (~S[2]) ;
```

//  $Y = 1$  nếu  $A[0] = 1$  và  $(S[0] = 1$  và  $(S[1] = 1$  và  $(S[2] = 1$  hoặc  $A[0] \& (S[0] \& (S[1] \& (S[2]$  hoặc  $A[2] \& (S[0] \& (\sim S[1] \& (S[2]$  hoặc  $A[3] \& (S[0] \& (\sim S[1] \& (\sim S[2])$  hoặc  $A[4] \& (\sim S[0] \& (S[1] \& (S[2]$  hoặc  $A[5] \& (\sim S[0] \& (S[1] \& (\sim S[2])$  hoặc  $A[6] \& (\sim S[0] \& (\sim S[1] \& (S[2]$  hoặc  $A[7] \& (\sim S[0] \& (\sim S[1] \& (\sim S[2])$ ).

Endmodule //Kết thúc module bài1

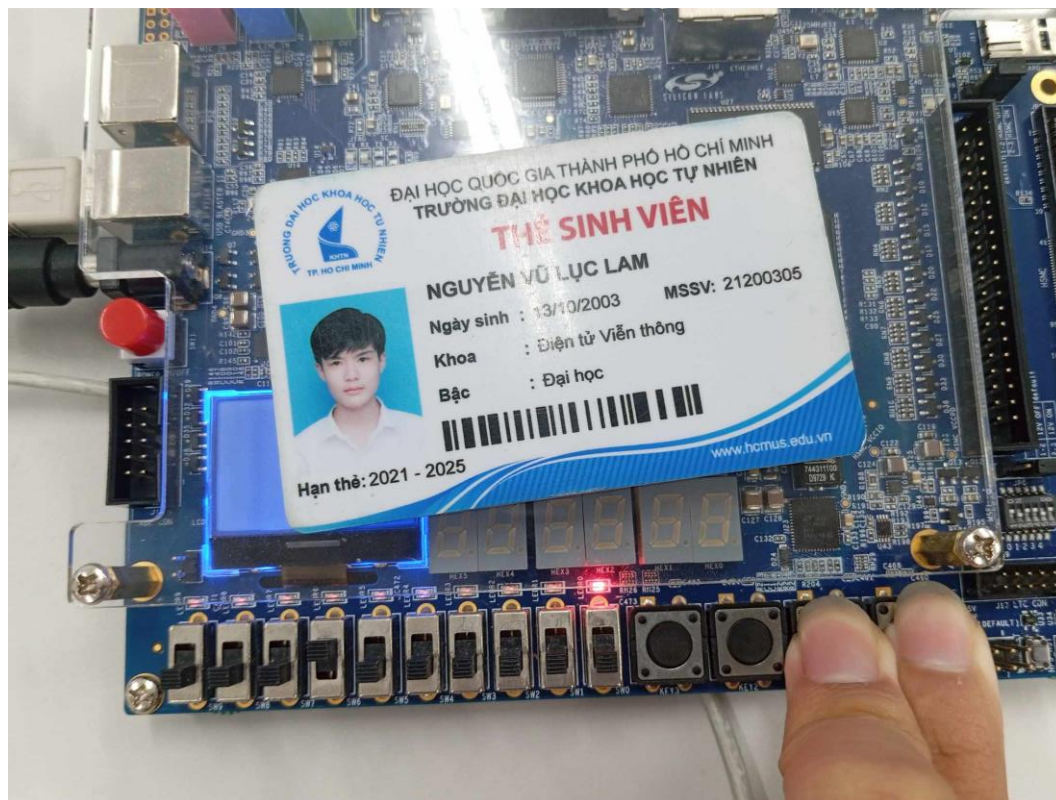














**Câu 2A:** Sử dụng ngôn ngữ Verilog HDL thiết kế mạch dịch led sử dụng 10 LED đỏ, và được điều khiển bởi các SW và KEY. Trong đó: • LED chạy theo hướng được điều khiển bởi SW0 (tự quy định), khi LED dịch đến cạnh thì quay trở lại từ LED đầu tiên • LED được dịch với tần số chính xác 1 Hz hoặc 8 Hz theo SW1 (tự quy định) được chia từ clock 50 MHz của board FPGA DE10 • Khi nhấn KEY0, led sẽ đứng yên tại chỗ cho đến khi KEY0 được nhấn lần nữa.

### BÀI LÀM

```
module Bai2 (CLOCK_50, SW, KEY, LEDR); //Khai báo tên module là Bai2
```

```
input CLOCK_50; //input là CLOCK_50
```

```
input [1:0] SW; //input SW có 2 bit
```

```
input KEY; //input là KEY
```

```
output reg [9:0] LEDR = 10'b0000000001; //khai báo LEDR gồm 10bit và xét trạng thái ban đầu là LED ngoài cùng bên phải sáng
```

```
reg timer1 = 1'b0; //Khai Báo timer1 và xét trạng thái đầu là bằng 0
```

```
reg timer2 = 1'b0; //Khai Báo timer2 và xét trạng thái đầu là bằng 0
```

```
reg enable = 1'b1; //enable có công dụng là cho phép mạch chạy hay dừng 1 chạy 0 là dừng xét trạng thái ban đầu cho nó là chạy luôn
```

```
reg [31:0] counter; // khai báo counter gồm 32 bit
```

```
wire speed, LR; //Khai báo 2 dây speed và LR, speed công dụng điều khiển tốc độ còn LR có công dụng là cho nó dịch trái hay phải
```

```
assign LR = (SW[0] == 1'b0) ? 1'b0 : 1'b1; //LR được nối với 0 nếu SW[0] = 0 không thì LR = 1, 0 thì dịch trái, 1 thì dịch phải
```

```
assign speed = (SW[1] == 1'b0) ? timer1 : timer2; //speed = timer1 nếu SW[1] = 0 còn không thì bằng timer2, timer1 là 1Hz, còn timer2 là 8Hz
```

```
always @(posedge CLOCK_50) //tích cực cạnh lên theo CLOCK_50
```

```
begin //xét để thực hiện lệnh tuần tự
```

```
counter <= counter + 1'b1; //gán song song để tạo tần số sau bằng ½ tần số trước
```

```
if (counter % 32'd49999999 == 1'b0) timer1 = 1'b1; //Nếu counter chia hết cho 49999999 thì timer1 đảo từ 0 sang 1
```

```
else timer1 = 1'b0; //không bằng thì đảo từ 1 sang 0 cứ như vậy lặp đi lặp lại đảo từ 0 sang 1, 1 sang 0 như xung nhịp vào tạo ra xung CLOCK 1Hz
```

```
if (counter % 32'd6249999 == 1'b0) timer2 = 1'b1; //Tương tự như trên
```

```
else timer2 = 1'b0;
```

```
end //Kết thúc begin
```

```
always @(posedge KEY) //tích cực cạnh lên theo KEY
```

```
if (KEY) //Nếu KEY = 1 thì anable lúc này là 1 chuyển sang 0 mà 0 là dừng không cho hoạt động nên LED sẽ dừng
```

```
enable <= ~enable;
```

```
always @(posedge speed) //Tích cực cạnh lên theo speed
```

```
if (enable) //Nếu anable là 1 cho hoạt động chạy LED
```

```
begin //thực hiện tuần tự
```

```
case (LR)
```

```
1'b0 : LEDR <= {LEDR[0], LEDR[9:1]}; //LEDR = 0 dịch  
trái
```

```
1'b1 : LEDR <= {LEDR[8:0], LEDR[9]}; //LEDR = 1 dịch  
phải
```

```
endcase
```

```
end // kết thúc begin
```

endmodule //kết thúc module

video bài 2:

[https://drive.google.com/file/d/17FUyuWx5mZskqt5C\\_4Vv0axp4mvGvntS/view?usp=sharing](https://drive.google.com/file/d/17FUyuWx5mZskqt5C_4Vv0axp4mvGvntS/view?usp=sharing)