

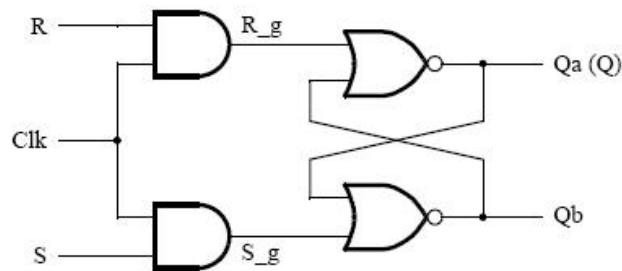
BÀI 3: THIẾT KẾ MẠCH CHỐT, FLIP FLOP VÀ THANH GHI

1. Mục tiêu bài thực hành

Trong bài thực hành này sinh viên sẽ đi thiết kế các vi mạch tuần tự như: mạch chốt (Latch), Flipflop, thanh ghi (register). Sau đó trong từng câu ta sẽ tạo ra các module verilog để thực hiện các mạch này và kiểm tra trực tiếp trên board DE1.

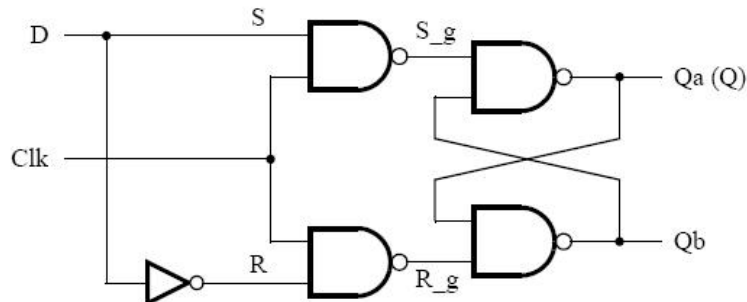
2. Yêu cầu của bài thực hành

Câu 1. Thực hiện module tạo ra mạch chốt RS, trong đó sử dụng SW[0] và SW[1] cho 2 tín hiệu ngõ vào R và S, SW[2] cho tín hiệu ngõ vào Clk. Kết nối tín hiệu ngõ ra Qa với LEDR0.



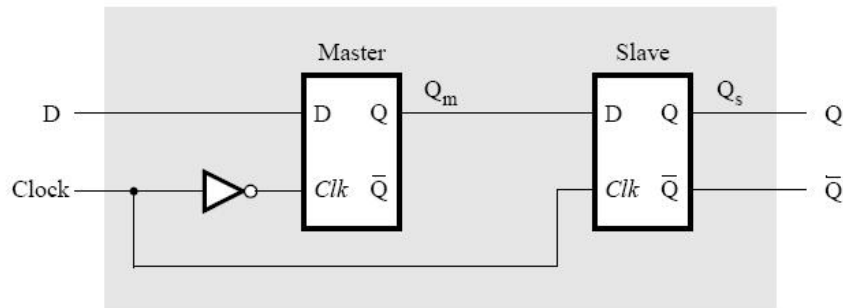
Hình 1.46: Mạch chốt RS [22]

Câu 2. Thực hiện module tạo ra mạch chốt D (D latch)



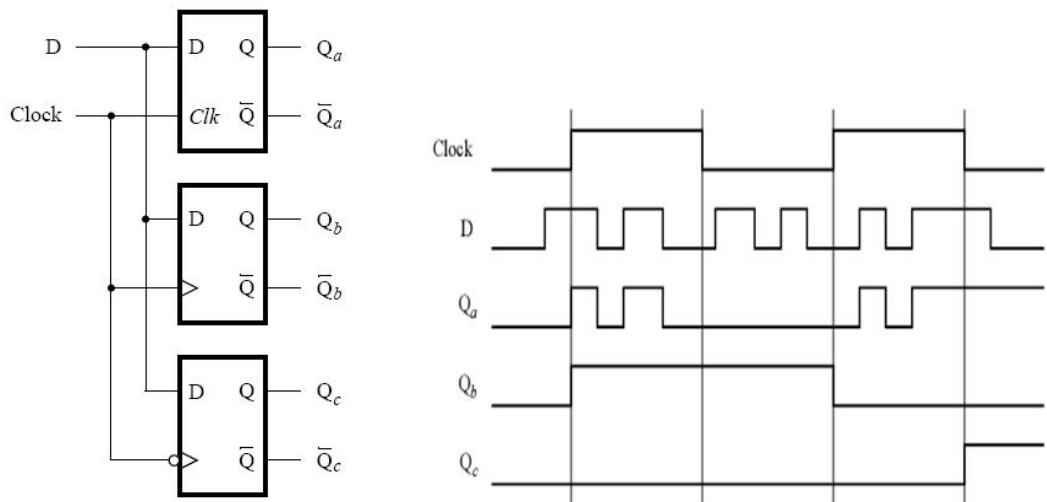
Hình 1.47: Mạch chốt D [22]

Câu 3. Thực hiện module tạo ra mạch master-slave D flipflop.



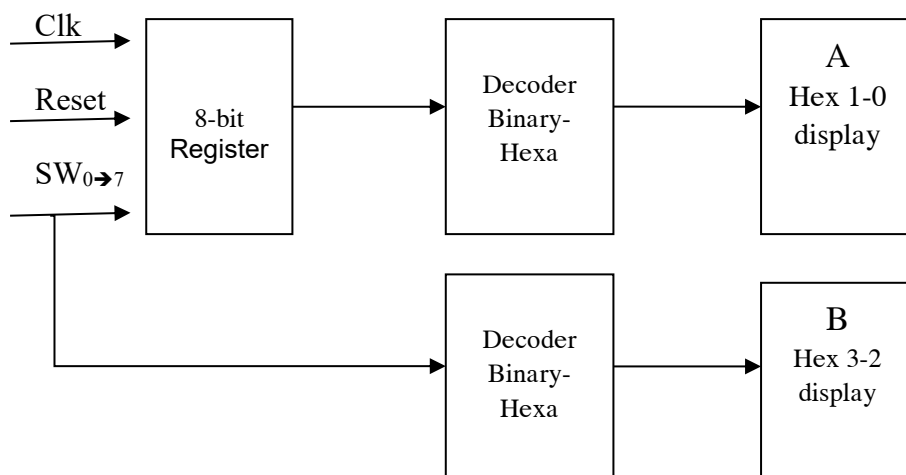
Hình 1.48: Mạch master-slave D flipflop [22]

Câu 4. Thực hiện module tạo ra các mạch flipflop D khác nhau theo hình vẽ sau:



Hình 1.49: Mạch flipflop D [22]

Câu 5. Thiết kế các module sau: một bộ giải mã số hexa 0 → F, thiết kế thanh ghi 8 bit. Ta dùng hết 4 đèn led 7 đoạn trên board DE1, thanh ghi 8 bit ngõ ra kết nối với bộ giải mã số hexa cho 2 đèn led đầu, và 8 bit ngõ vào điều khiển 2 đèn led 7 đoạn còn lại.



Hình 1.50: Mạch hiển thị số HEX sử dụng thanh ghi 8 bit

3. Hướng dẫn thực hành

Câu 1. Thực hiện module tạo ra mạch chốt RS bằng cách dùng các hàm dựng sẵn: **and(out, in1, in2), nor(out, in1, in2).**

Bước 1: Tạo một project

Bước 2: Ta thực hiện module verilog như sau:

```
module RS_FF (R,S,clk,Qa,Qb);
    /*****/
    input      R, S, clk;
    wire      R_g, S_g, Qa, Qb;
    output    Q; // ngõ ra chính.
```

```

/*****/
// Bắt đầu đoạn code

and (R_g, R, Clk);
and (S_g, S, Clk);
nor (Qa, R_g, Qb);
nor (Qb, S_g, Qa);

assign Q = Qa;

// kết thúc đoạn code

/*****/
endmodule

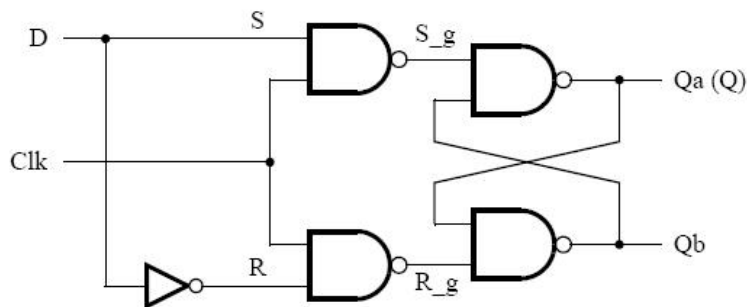
```

Bước 3: Biên dịch và chạy mô phỏng/nạp vào board FPGA.

Bước 4: Kiểm tra chức năng hoạt động của mạch bằng việc thay đổi các giá trị ngõ vào và giá trị ở ngõ ra.

Câu 2. Thực hiện module tạo ra mạch chốt D (D latch).

Ta tạo mạch chốt D dựa vào mạch chốt RS.



Hình 1.52: Mạch chốt D [22]

Dùng lệnh assign đối với hình trên, hoặc dùng if-else.

Lưu ý: Trong câu này chúng ta sẽ thực hiện các bước tương tự như câu 1, sau đó kiểm tra trực tiếp trên board DE1 sử dụng các thành phần sau: SW0 làm cổng vào D, SW1 làm cổng vào clock, LEDR0 làm ngõ ra Q.

Câu 3. Thực hiện module tạo ra mạch master-slave D flipflop.

Ta thực hiện tạo module verilog FF_D.

Bước 1: Tạo một project

Bước 2: Ta thực hiện module verilog như sau:

```

module M_S_FFD (
    SW,
    LEDR
);
/*****/

```

```

        input  [1:0]SW;
        output [0:0]LEDR;
        wire   Qm;
    /***/
    // Bắt đầu đoạn code
    /***/
    // giả sử các bạn đã tạo ra module FF_D ( d_out, d, clk )
    FF_D ( Qm,          SW[0],          ~ SW[1] );
    FF_D ( LEDR[0],     Qm,              SW[1]);

    // kết thúc đoạn mã
    /***/
endmodule

```

Bước 3: Biên dịch và nạp vào board FPGA.

Bước 4: Kiểm tra chức năng hoạt động của mạch bằng việc thay đổi các giá trị ngõ vào và giá trị ở ngõ ra. Sau khi nạp vào board ta quan sát ngõ ra LEDR[0].

Câu 4. Thực hiện module tạo ra các mạch flipflop D khác nhau.

Đối với flipflop D kích cạnh xuống hoặc lên của xung clock thì ta sử dụng code sau, thực hiện trong khối lệnh *always*: *always @ (negedge clk)* hoặc *always @ (posedge clk)* đối với cạnh xuống hoặc cạnh lên. Tạo ra ba loại flipflop D: Kích cạnh lên, kích cạnh xuống, kích ở mức logic 1.

Bước 1: Tạo một project

Bước 2: Ta thực hiện module verilog như sau:

```

module      FF_3KIND_OF_D ( d,clk,q1,q2,q3);
    /***/
    input          d, clk;
    output reg      q1, q2, q3; /*việc khai báo từ khóa reg đối với ngõ ra là bắt buộc khi
    sử dụng always construct .*/
    /***/
    // bắt đầu đoạn code
    /***/
    always @ (clk) // Flipflop D kích ở mức logic 1
        if (clk)
            q1 = d;
    always @ ( negedge clk) // Flipflop D kích cạnh xuống.
        ....
    /***/
    // kết thúc đoạn code
    /***/
endmodule

```

Bước 3: Biên dịch và nạp vào board FPGA.

Bước 4: Kiểm tra chức năng hoạt động của mạch bằng việc thay đổi các giá trị ngõ vào và giá trị ở ngõ ra.

Câu 5. Thiết kế các module sau: một bộ giải mã số hexa 0 -> F, thiết kế thanh ghi 8 bit. Ta dùng hết 4 đèn led 7 đoạn trên board DE1, thanh ghi 8 bit ngõ ra kết nối với bộ giải mã số hexa cho 2 đèn led đầu, và 8 bit ngõ vào điều khiển 2 đèn led 7 đoạn còn lại.

Bước 1: Tạo một project

Bước 2: Ta thực hiện :

- Tạo module flipflip D có reset .
- Tạo bộ giải mã binary to Hexa.
- Trong Top-module ta có sử dụng thêm push button có tên định danh là KEY. Có 2 nút nhấn KEY[0] và KEY[1]. Ta khai báo như là cổng vào.

Bước 3: Biên dịch và nạp vào board FPGA.

Bước 4: Kiểm tra chức năng hoạt động của mạch bằng việc thay đổi các giá trị ngõ vào và giá trị ở ngõ ra.