

## BÀI 10: TỔNG HỢP

Câu 1: Sử dụng Multiplexer (MUX) để tổng hợp chức năng logic và viết code verilog mô tả mạch chức năng logic.

a)

b)

c)

$w_1$	$w_2$	$w_3$	$f$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

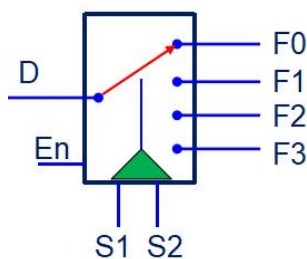
$w_1$	$w_2$	$w_3$	$f$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

$w_1$	$w_2$	$w_3$	$f$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Kiểm tra hoạt động của các mạch chức năng logic trên board DE1, sử dụng các tín hiệu trên board như sau:

- SW[0] =  $w_1$
- SW[1] =  $w_2$
- SW[2] =  $w_3$
- LEDR[0] =  $f$

Câu 2: Cho mạch tách kênh (demultiplexer) như sau:

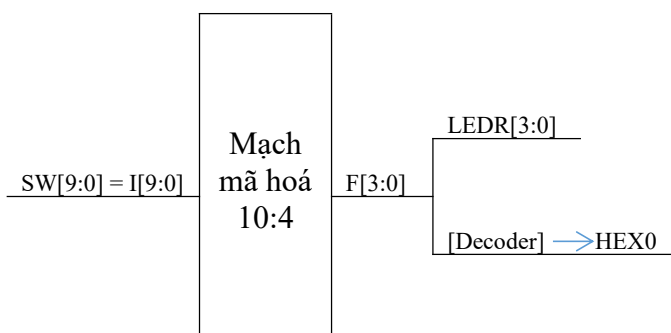


En	S1	S0	Ngõ ra
0	x	x	$F0=F1=F2=F3=0$
1	0	0	$F0 = D$
1	0	1	$F1 = D$
1	1	0	$F2 = D$
1	1	1	$F3 = D$

Mạch tách kênh từ 1 đường sang 4 đường nên có 2 ngõ chọn (S1, S2). Khi ngõ cho phép En ở mức 0 thì cấm không cho phép dữ liệu vào được truyền ra ở bất kì ngõ ra nào nên tất cả các ngõ ra đều bằng 0. Khi En ở mức 1, nếu ngõ vào chọn S1S0=00 thì dữ liệu D được đưa ra ngõ F0, các ngõ khác không đổi. Tương tự với các tổ hợp S1S0 khác thì lần lượt D sẽ ra ở ngõ F1, F2 và F3.

Viết code verilog mô tả mạch tách kênh 1 sang 4 trên và kiểm tra hoạt động của mạch trên board DE1. Sử dụng các tín hiệu trên board như sau: SW[0]=D; SW[1]=S1; SW[2]=S2; SW[3]=En; LEDR[0]=F[0]; LEDR[1]=F[1]; LEDR[2]=F[2]; LEDR[3]=F[3].

Câu 3: Viết code verilog mô tả mạch mã hoá 10 đường sang 4 đường như sau:

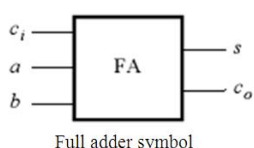


$I_0$	$I_1$	$I_2$	$I_3$	$I_4$	$I_5$	$I_6$	$I_7$	$I_8$	$I_9$	F3	F2	F1	F0
1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	0	0	0	1	0	0
0	0	0	0	0	1	0	0	0	0	0	1	0	1
0	0	0	0	0	0	1	0	0	0	0	1	1	0
0	0	0	0	0	0	0	1	0	0	0	1	1	1
0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	0	0	0	0	0	0	0	1	1	0	0	1

Câu 4: Viết code verilog mô tả mạch thực hiện chức năng chuyển đổi số hệ nhị phân 4 bit (B[3:0]) thành số hệ thập phân 2 chữ số (d1 d0). Kiểm tra hoạt động của mạch trên board DE1, sử dụng các tín hiệu: - SW[3:0] = B[3:0] : làm 4 bit ngõ vào  
- HEX1, HEX0 = d1, d0 : hiển thị số thập phân 2 chữ số

B[3:0]	d1	d0
0000	0	0
0001	0	1
0010	0	2
...	...	...
1001	0	9
1010	1	0
1011	1	1
1100	1	2
1101	1	3
1110	1	4
1111	1	5

Câu 5: Viết code verilog mô tả mạch Full Adder 4 bit được ghép từ 4 Full Adder 1 bit. Cho mạch Full Adder 1 bit như sau:



b	a	ci	co	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Full adder truth table

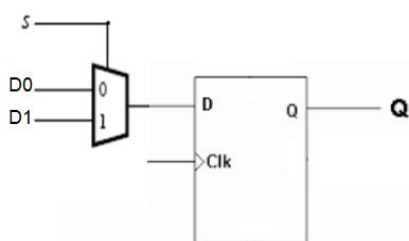
```

module fulladder1bit(cin,a,b,s,cout);
  input cin,a,b;
  output s,cout;
  assign s=a^b^cin;
  assign cout=(a&b)|(cin&a)|(cin&b);
endmodule

```

Câu 6: Viết code verilog mô tả mạch Full Adder 4 bit sử dụng **parameter n=4**.

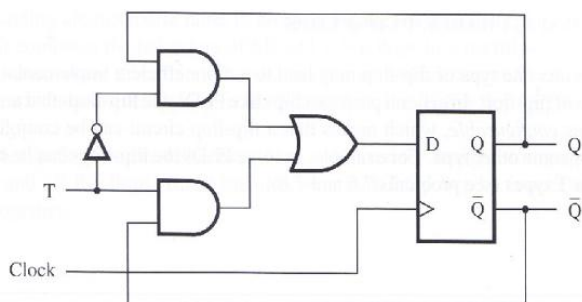
Câu 7. Viết code verilog mô tả mạch Flip-Flop D với Mux 2:1 ở ngõ vào D (4 bits) như hình sau:



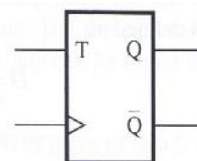
Kiểm tra mạch trên Board DE1 bằng cách sử dụng các tín hiệu sau:

- SW[3:0] = D0
- SW[7:4] = D1
- SW[9] = s
- LEDR[3:0] = Q

Câu 8. Viết code verilog mô tả mạch Flip-Flop T như sau:



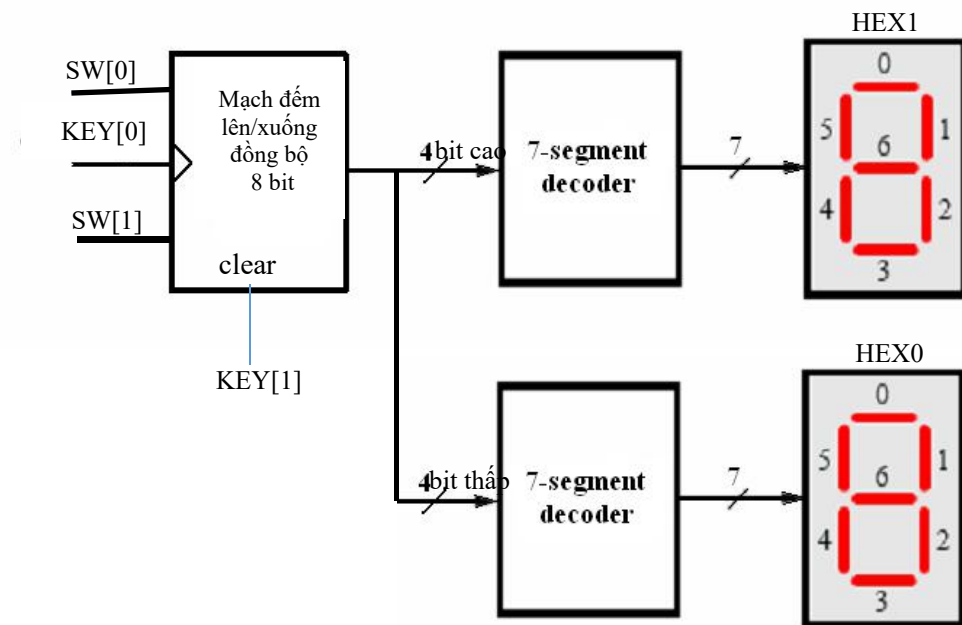
T	Q(t+1)
0	Q(t)
1	$\bar{Q}(t)$



Kiểm tra mạch trên Board DE1 bằng cách sử dụng các tín hiệu:

- SW[9] = T
- KEY[0] = Clk
- LEDR[0] = Q

Câu 9: Viết code verilog mô tả mạch đếm lên/xuống đồng bộ 8 bit sử dụng biểu thức toán học.



Yêu cầu sử dụng các tín hiệu trên board như sau:

- SW[0] làm tín hiệu Enable.
- SW[1] làm tín hiệu Select: nếu SW[1]=0 thực hiện mạch đếm lên, SW[1]=1 thực hiện mạch đếm xuống.
- KEY[0] làm tín hiệu xung clock.
- KEY[1] làm tín hiệu clear bộ đếm về 0.
- HEX1, HEX0 hiển thị kết quả của bộ đếm lên từ 00 <--> FF

Câu 10: Tương tự câu 9, nhưng thay tín hiệu xung clock bằng việc sử dụng bộ dao động CLOCK\_50MHZ để tạo trì hoãn 1/2 giây.