


Họ tên sinh viên	MSSV	Lớp (thứ - tiết)	

Chú ý: Sinh viên thay bằng QR code của mã số sinh viên (ví dụ: 23119012), có thể tham khảo tại <https://barcode.tec-it.com>)

Quick question : chapter 04

Lưu ý: Trong mỗi thiết kế yêu cầu sinh viên thực hiện

- *Sơ đồ khối (nguyên lý, cấu trúc)*
 - *Bảng trạng thái*
 - *Mô tả bằng ngôn ngữ Verilog cho module cần thiết kế,*
 - *Mô tả Verilog cho module dùng để kiểm tra thiết kế*
 - *Kết quả mô phỏng quá trình kiểm tra, có phân tích*
 - *Module test được đặt tên theo cú pháp: tensv_testbench_tenmodule, ví dụ để test module encoder, sinh viên Nguyen Van An phải đặt tên module test như sau: An_testbench_encoder. Các kết quả mô phỏng phải được chụp màn hình bao gồm cả tên của module test trong đó có tên sinh viên thì mới hợp lệ*
1. Thiết kế và mô phỏng kiểm chứng mạch cộng 4 bit từ mạch cộng toàn phần 1 bit, sử dụng mô tả cấu trúc.

2. Thiết kế và mô phỏng kiểm chứng mạch đa hợp 8 sang 1

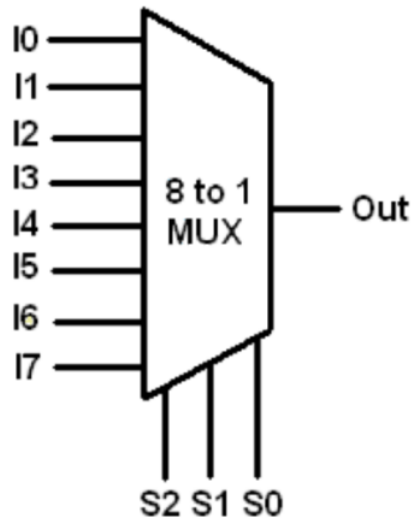


Figure 1: Sơ đồ khối mạch đa hợp 8 sang 1

Bảng trạng thái

S1	S2	S3	O
0	0	0	I1
0	0	1	I2
0	1	0	I3
0	1	1	I4
1	0	0	I5
1	0	1	I6
1	1	0	I7
1	1	1	I8

Mô tả bằng ngôn ngữ Verilog

```
module mux8to1(  
input wire [7:0] w,  
input wire [2:0] s,  
output reg y);
```

```

always @(w,s)
case(s)
0: y = w[0];
1: y = w[1];
2: y = w[2];
3: y = w[3];
4: y = w[4];
5: y = w[5];
6: y = w[6];
default: y = w[7];
endcase
endmodule

```

Mô tả Verilog cho module dùng để kiểm tra thiết kế

```

`timescale 1ns/1ns // define timescale
module dung_tb_mux8to1();

reg [7:0] w;
reg [2:0] s;
wire y;

// initialize w and s to 000 and 00, respectively
initial begin
w = 0 ;
s = 0;
end

// generate w0- w3
always forever #10 w[0] = ~w[0];
always forever #20 w[1] = ~w[1];
always forever #40 w[2] = ~w[2];
always forever #80 w[3] = ~w[3];
always forever #100 w[4] = ~w[4];
always forever #200 w[5] = ~w[5];
always forever #400 w[6] = ~w[6];
always forever #800 w[7] = ~w[7];

//generate 4 state of s by increasing s by 1
always forever #500 s = s + 1;
// conect w,s,y to moduke mux41
mux8to1 m0(w,s,y);
endmodule

```

Kết quả mô phỏng

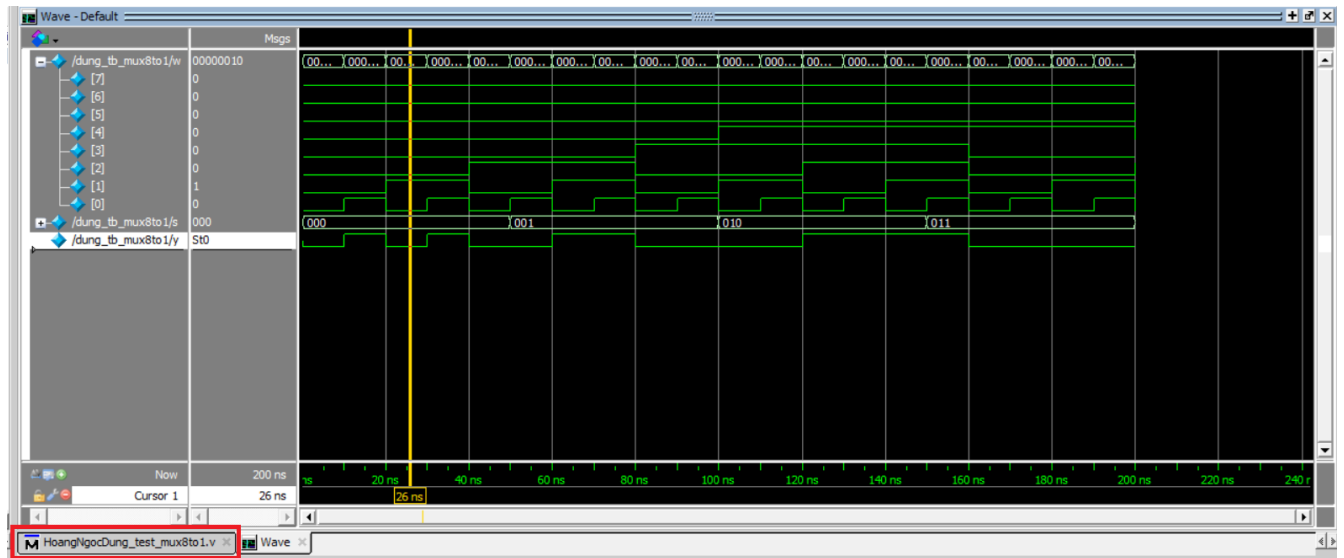


Figure 2: TUTU viết mô tả

3. Thiết kế và mô phỏng kiểm chứng mạch giải đa hợp 1 sang 8

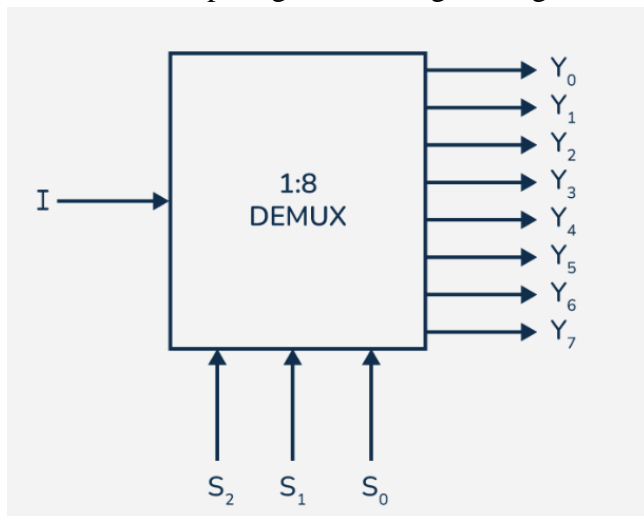


Figure 3: Sơ đồ mạch giải đa hợp 1 sang 8

Bảng trạng thái

Selection Inputs			Outputs							
S2	S1	S0	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
0	0	0	0	0	0	0	0	0	0	I
0	0	1	0	0	0	0	0	0	I	0
0	1	0	0	0	0	0	0	I	0	0
0	1	1	0	0	0	0	I	0	0	0
1	0	0	0	0	0	I	0	0	0	0
1	0	1	0	0	I	0	0	0	0	0
1	1	0	0	I	0	0	0	0	0	0
1	1	1	I	0	0	0	0	0	0	0

Mô tả bằng ngôn ngữ Verilog

```
module mux8to1(  
input wire [7:0] w,  
input wire [2:0] s,  
output reg y);  
  
always @(w,s)  
case(s)  
0: y = w[0];  
1: y = w[1];  
2: y = w[2];  
3: y = w[3];  
4: y = w[4];  
5: y = w[5];  
6: y = w[6];  
default: y = w[7];  
endcase  
endmodule
```

Mô tả Verilog cho module dùng để kiểm tra thiết kế

```
`timescale 1ns/1ns // define timescale  
module dung_tb_demux1to8();  
  
reg in;  
reg [2:0] sel;  
wire [7:0] out;  
  
initial begin  
in = 1;  
sel = 0;  
end  
// generate sel0- sel2  
always forever #10 sel[0] = ~sel[0];  
always forever #20 sel[1] = ~sel[1];  
always forever #40 sel[2] = ~sel[2];  
always forever #80 in = in + 1;  
// conect in,sel,out to module demux8to1  
demux1to8 m1(in,sel,out);  
  
endmodule
```

Kết quả mô phỏng

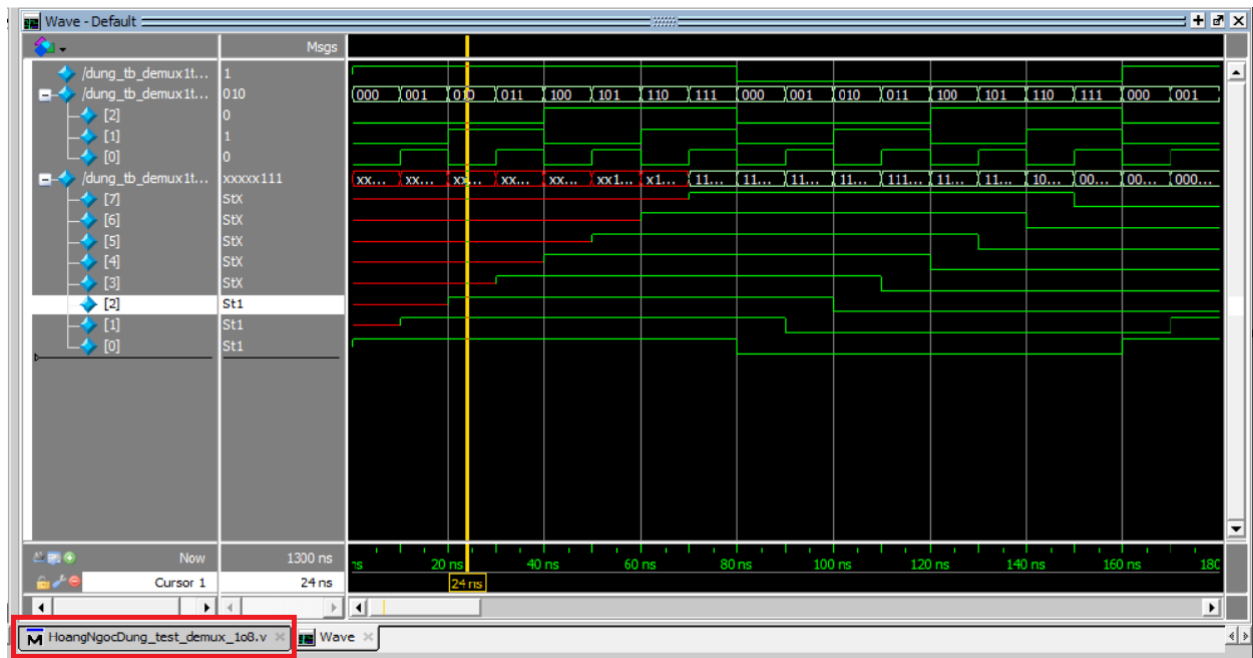


Figure 4: Mô tả sau..

4. Thiết kế và mô phỏng mạch giải mã 3 sang 8 có tín hiệu cho phép (enable –EN) ngõ ra tích cực mức thấp

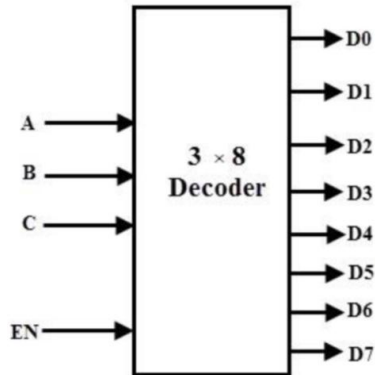


Figure 5: tín hiệu cho phép mức cao ($EN = 1$) của mạch giải mã

Bảng trạng thái

Input				Output							
En	A	B	C	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
1	x	x	x	1	1	1	1	1	1	1	1
0	0	0	0	1	1	1	1	1	1	1	0
0	0	0	1	1	1	1	1	1	1	0	1
0	0	1	0	1	1	1	1	1	0	1	1
0	0	1	1	1	1	1	1	0	1	1	1
0	1	0	0	1	1	1	0	1	1	1	1
0	1	0	1	1	1	0	1	1	1	1	1
0	1	1	0	1	0	1	1	1	1	1	1
0	1	1	1	0	1	1	1	1	1	1	1

Mô tả bằng ngôn ngữ Verilog

```

module decoder3to8_en0(
input wire [2:0] in,
input wire en,
output reg [7:0] out);

always @(in or en)
    if (en == 0)
        out = ~(8'b00000001 << in);
    else
        out = 8'b11111111;
endmodule

```


Mô tả Verilog cho module dùng để kiểm tra thiết kế

```
`timescale 1ns/1ns // define timescale
module dung_tb_decoder3to8_en0();

reg [2:0] in;
reg en;
wire [7:0] out;

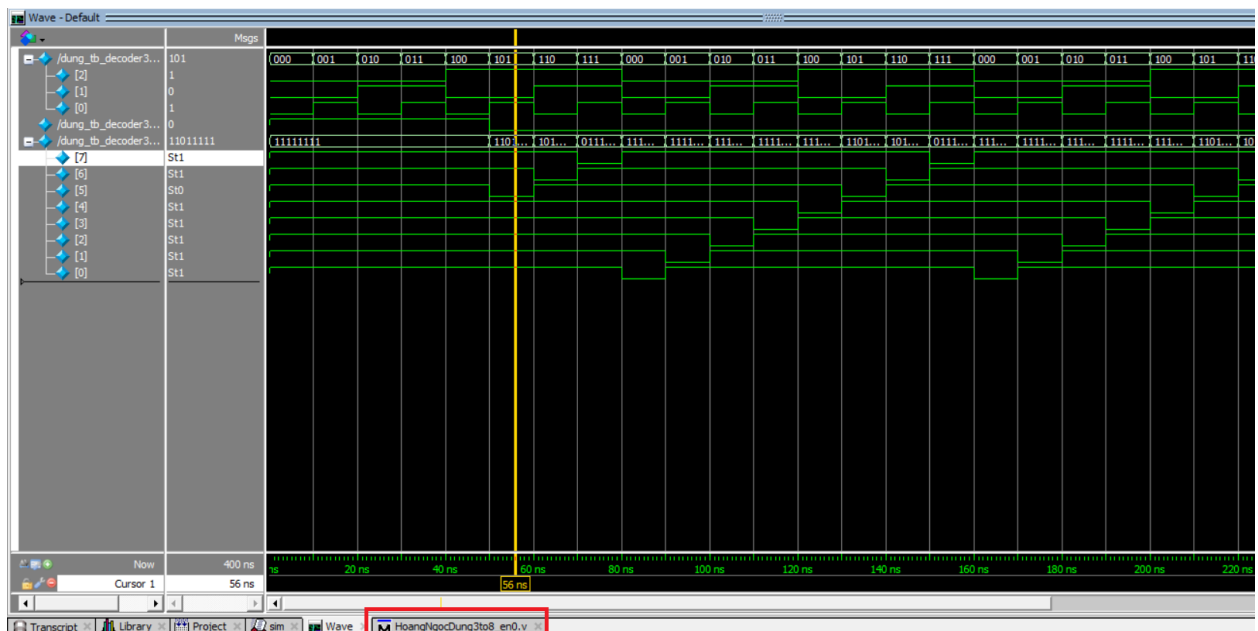
// initialize w and s to 000 and 00, respectively
initial begin
en = 1;
in = 0;
end

always forever #10 in[0] = ~in[0];
always forever #20 in[1] = ~in[1];
always forever #40 in[2] = ~in[2];
always forever #50 en = 0;

decoder3to8_en0 m3(in,en,out);

endmodule
```

Kết quả mô phỏng



5. Thiết kế và mô phỏng mạch giải mã 3 sang 8 có tín hiệu cho phép (enable –EN), ngõ ra tích cực mức cao

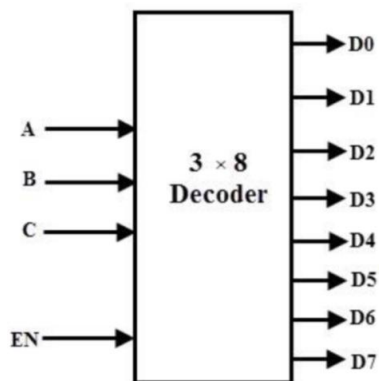


Figure 6: tín hiệu cho phép mức cao (EN = 1) của mạch giải mã

Bảng trạng thái

Input				Output							
En	A	B	C	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
0	x	x	x	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	0	1	0	0
1	0	1	1	0	0	0	0	1	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0

Mô tả bằng ngôn ngữ Verilog

```

module decoder3to8_en1(
input wire [2:0] in,
input wire en,
output reg [7:0] out);

always @(in or en)
    if (en == 1)
        out = (8'b00000001 << in);
    else
        out = 8'b00000000;
endmodule

```

Mô tả Verilog cho module dùng để kiểm tra thiết kế

```
`timescale 1ns/1ns // define timescale
module dung_tb_decoder3to8_en1();

reg [2:0] in;
reg en;
wire [7:0] out;

// initialize w and s to 000 and 00, respectively
initial begin
en = 0;
in = 0;
end

always forever #10 in[0] = ~in[0];
always forever #20 in[1] = ~in[1];
always forever #40 in[2] = ~in[2];
always forever #40 en = ~en;

decoder3to8_en1 m2(in,en,out);

endmodule
```

Kết quả mô phỏng

