CS 207 Digital Logic - Spring 2020
Lab 6

James Yu

Mar. 25, 2020

Objective

1. Implement basic combinational logic circuits.

Lab Exercise Submission

1. You should name all source code as instructed. Mis-named files will not be recognized.
2. You should submit all source code files with an extension "`.v`".
3. You should submit all source code directly into the sakai system below. **Do not compress them into one folder.**

   `https://sakai.sustech.edu.cn/portal/site/ebf68254-68c5-4cfe-9d42-b26758f854ee`

4. Lab exercises should be submitted before the deadline, typically one week after the lab session. No late submission policy applies to lab exercises.

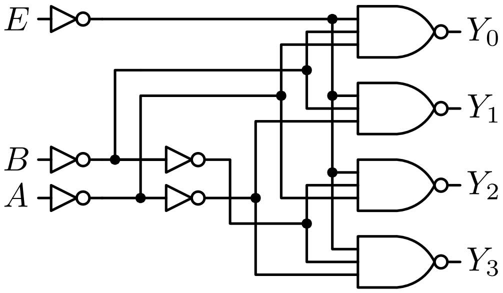
1 2-4 Decoder

Implement a 2-4 decoder as follows:

Compared with gate-leveling modeling, behavioral modeling can be more expressive and easier to understand by human.

dec2to4.v

```
module dec2to4(Y, E, A, B);
input E;
input A, B;
output [3:0] Y;
```



```
reg [3:0] Y;

always @(E or A or B)
    if (E)
        Y <= 4'b1111;
    else begin
        case({B, A})
        0: Y <= 4'b1110;
        1: Y <= 4'b1101;
        2: Y <= 4'b1011;
        3: Y <= 4'b0111;
        endcase
    end
endmodule
```

Or, we can have an even straight-forward Boolean logic function-based modeling:

dec2to4.v

```
module dec2to4(Y, E, A, B);
input E;
input A, B;
output [3:0] Y;

assign Y[0] = ~(~A & ~B & ~E);
assign Y[1] = ~(A & ~B & ~E);
assign Y[2] = ~(~A & B & ~E);
assign Y[3] = ~(A & B & ~E);
endmodule
```

To test the code, we have the following testbench:

dec2to4.v

```
module dec2to4_tb;
wire [3:0] IN;
reg A,B;
reg E;
dec2to4 dec(E, A, B, IN);

initial E=0;

initial begin
  $monitor("IN0 = %b, IN1 = %b, IN2 = %b, IN3 = %b",IN[0],IN[1],IN[2],IN[3]);
  $display("%3t: B = %b, A = %b, E = %b",$time,B,A,E);
  #5 B=0; A=0;
  #5 B=0; A=1;
  #5 B=1; A=0;
  #5 B=1; A=1;
  #10 $finish;
end
endmodule
```

Try them out and see if the results are same.

2 Exercise

2.1 Exercise 1

Implement a 3-8 decoder `74LS138`, whose truth table is as follows:

Input					Output							
E_1	$E_2' + E_3'$	A_2	A_1	A_0	Y_0'	Y_1'	Y_2'	Y_3'	Y_4'	Y_5'	Y_6'	Y_7'
0	-	-	-	-	1	1	1	1	1	1	1	1
-	1	-	-	-	1	1	1	1	1	1	1	1
1	0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	0	1	1	0	1	1	1	1	1	1
1	0	0	1	0	1	1	0	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1	1	1
1	0	1	0	0	1	1	1	1	0	1	1	1
1	0	1	0	1	1	1	1	1	1	0	1	1
1	0	1	1	0	1	1	1	1	1	1	0	1
1	0	1	1	1	1	1	1	1	1	1	1	0

The module should follow the design below:

`74LS138.v`

```
module dec3to8(nY, E1, nE2, nE3, A);
```

```
// Module Code
endmodule
```

and save in file `74LS138.v`.

Assignment

Save the source code in **`74LS138.v`**. Upload this file to Sakai under **Assignments → Lab Exercise 6**.

2.2 Exercise 2

Implement a priority encoder `74LS148`, whose truth table is as follows:

Input									Output				
E_I	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	Y_2	Y_1	Y_0	E_S	E_O
1	-	-	-	-	-	-	-	-	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	-	-	-	-	-	-	-	0	0	0	0	1
0	1	0	-	-	-	-	-	-	0	0	1	0	1
0	1	1	0	-	-	-	-	-	0	1	0	0	1
0	1	1	1	0	-	-	-	-	0	1	1	0	1
0	1	1	1	1	0	-	-	-	1	0	0	0	1
0	1	1	1	1	1	0	-	-	1	0	1	0	1
0	1	1	1	1	1	1	0	-	1	1	0	0	1
0	1	1	1	1	1	1	1	0	1	1	1	0	1

The module should follow the design below:

`74LS148.v`

```
module enc8to3(Y, ES, EO, EI, I);
// Module Code
endmodule
```

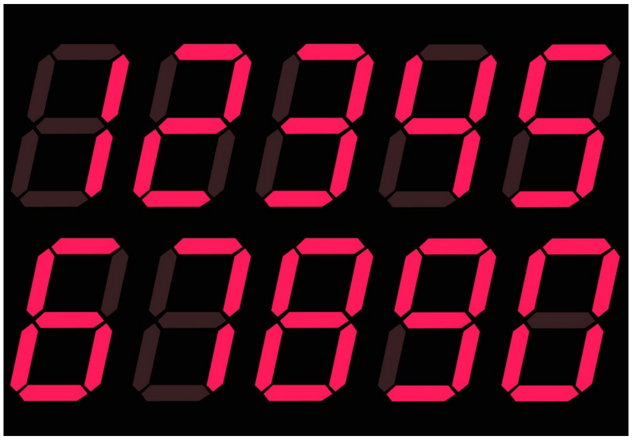
and save in file `74LS148.v`.

Assignment

Save the source code in **`74LS148.v`**. Upload this file to Sakai under **Assignments → Lab Exercise 6**.

2.3 Exercise 3

Implement the display logic of a numeric nixie tube, which shows numbers:



Starting from the top most bar, the tubes are labelled `a` to `f` clockwisely (or F_0 to F_5). The tube in the middle is `g` or F_6. When applied with low voltage (logic `0`), the tube will light up. Therefore, the truth table of the logic is:

	0	1	2	3	4	5	6	7	8	9	Others
D_3	0	0	0	0	0	0	0	0	1	1	-
D_2	0	0	0	0	1	1	1	1	0	0	-
D_1	0	0	1	1	0	0	1	1	0	0	-
D_0	0	1	0	1	0	1	0	1	0	1	-
F_0	0	1	0	0	1	0	0	0	0	0	1
F_1	0	0	0	0	0	1	1	0	0	0	1
F_2	0	0	1	0	0	0	0	0	0	0	1
F_3	0	1	0	0	1	0	0	1	0	0	1
F_4	0	1	0	1	1	1	0	1	0	1	1
F_5	0	1	1	1	0	0	0	1	0	0	1
F_6	1	1	0	0	0	0	0	1	0	0	1

The module should follow the design below:

led.v

```
module led(F, D);
// Module Code
endmodule
```

and save in file `led.v`.

⚠ **Assignment**

Save the source code in `led.v`. Upload this file to Sakai under **Assignments → Lab Exercise 6**.

⚠ **Assignment**

When you finished Lab Exercise 6, there should be three `.v` files in the Sakai system: `74LS138.v`, `74LS148.v`, and `led.v`.