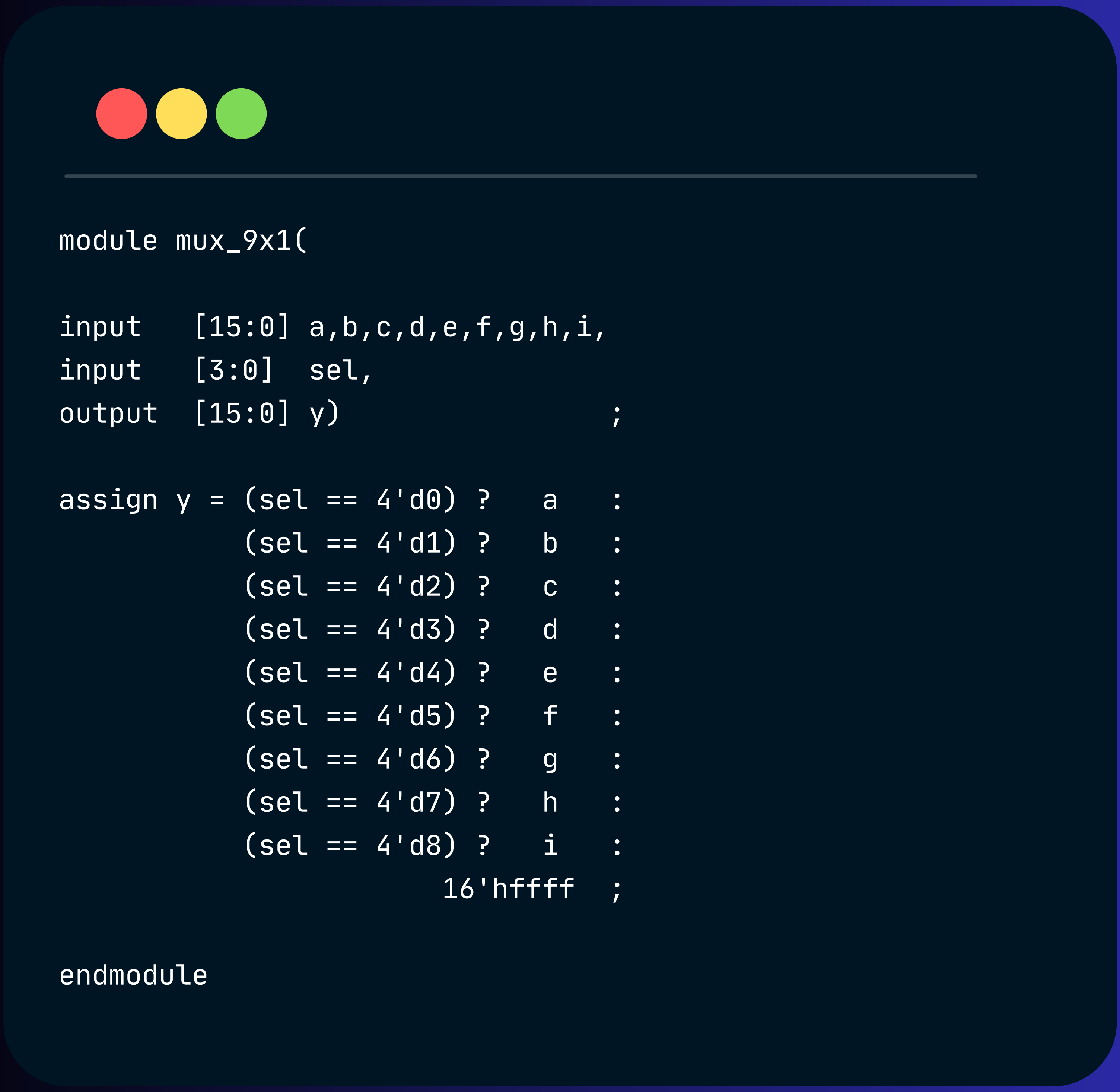
#100rtldays Challenge

#Day_10

Design a 16 bit wide 9 to 1 multiplexer.

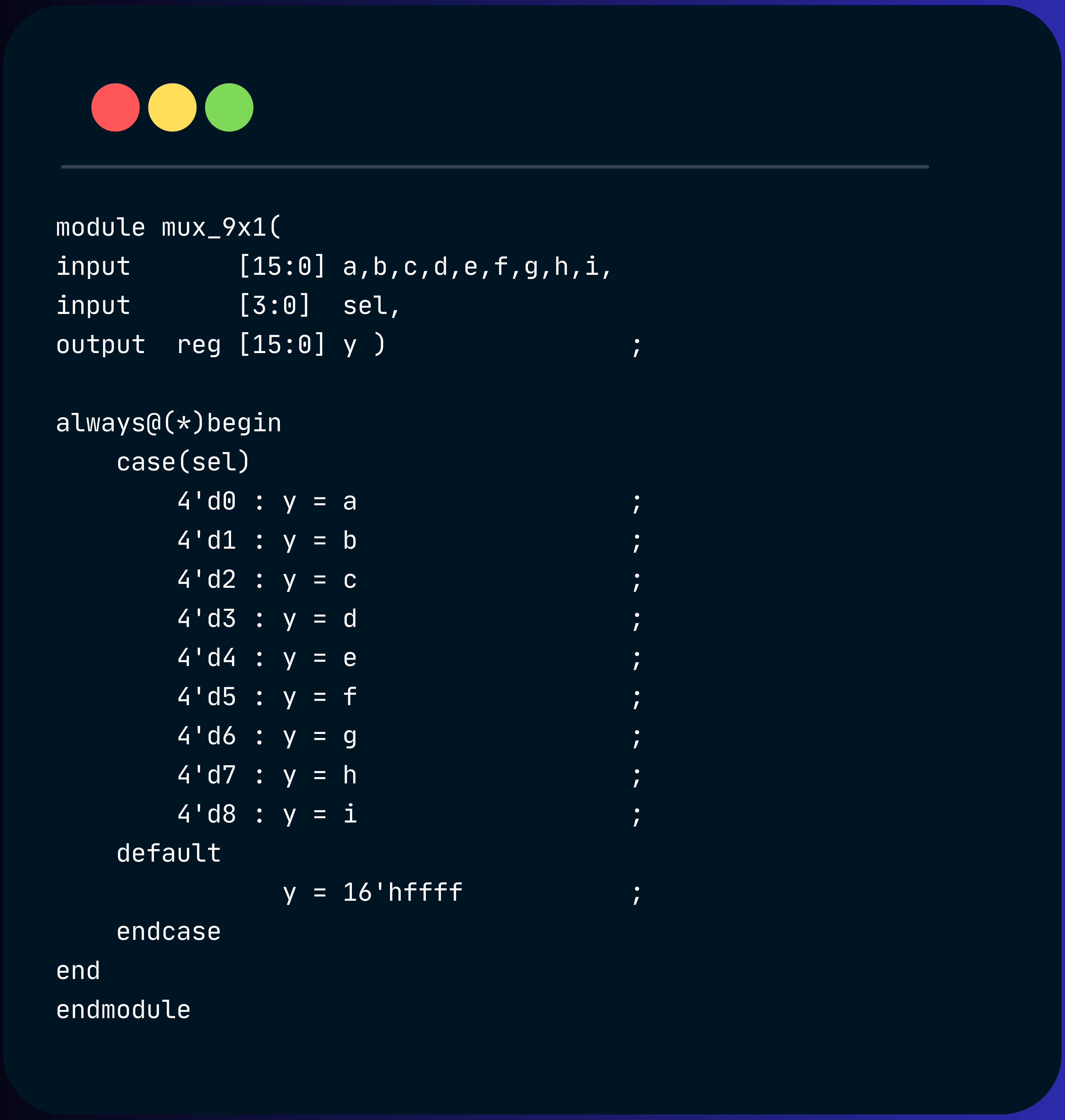
if sel=0,1,..8 then,
    y=a,b,..i
if sel=9,...15 then,
    y=ffff

DESIGN (method:1)



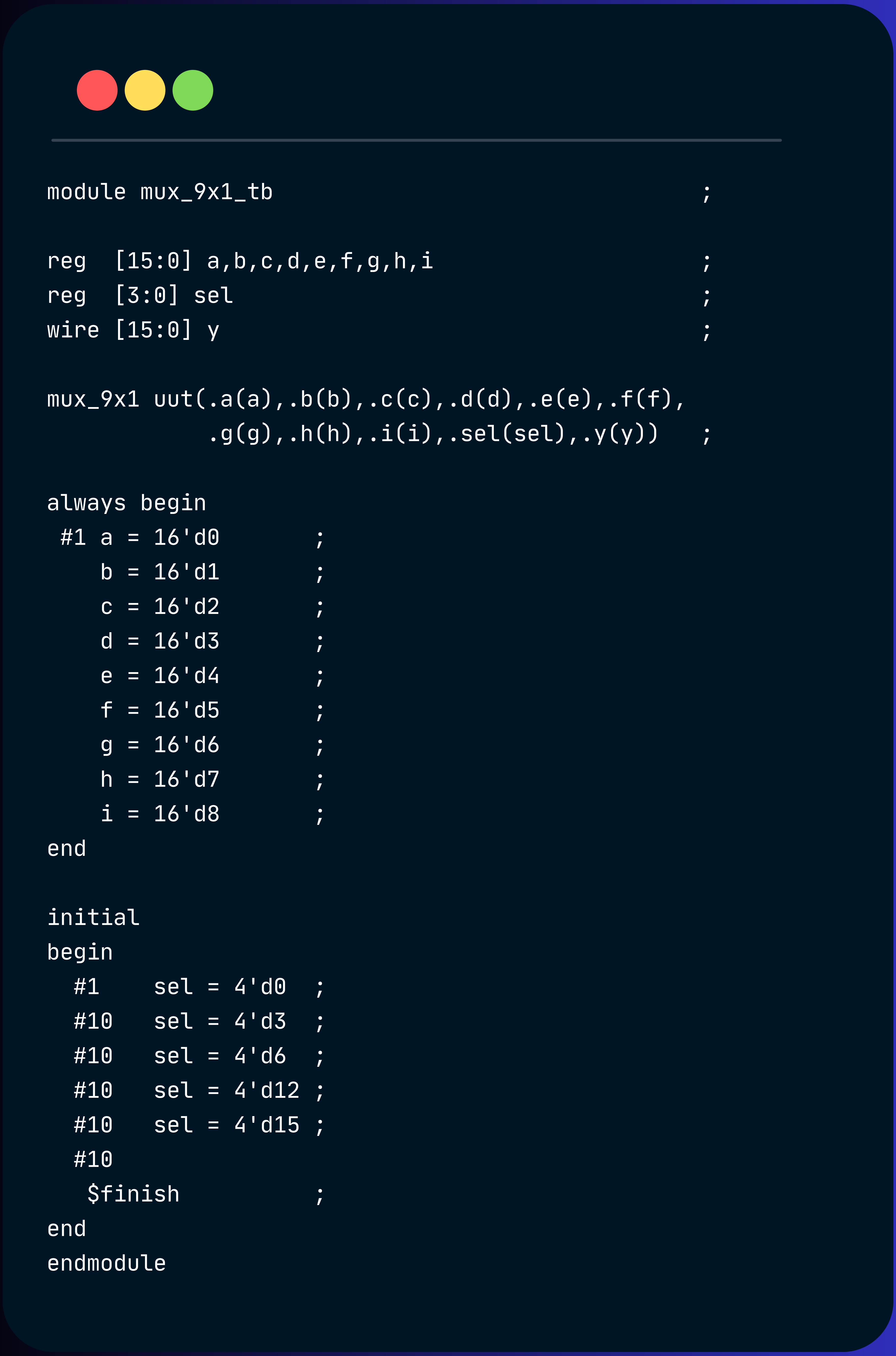
```
module mux_9x1(

input   [15:0] a,b,c,d,e,f,g,h,i,
input   [3:0]  sel,
output  [15:0] y)                ;

assign y = (sel == 4'd0) ?   a   :
           (sel == 4'd1) ?   b   :
           (sel == 4'd2) ?   c   :
           (sel == 4'd3) ?   d   :
           (sel == 4'd4) ?   e   :
           (sel == 4'd5) ?   f   :
           (sel == 4'd6) ?   g   :
           (sel == 4'd7) ?   h   :
           (sel == 4'd8) ?   i   :
                       16'hffff  ;

endmodule
```

DESIGN (method:2)



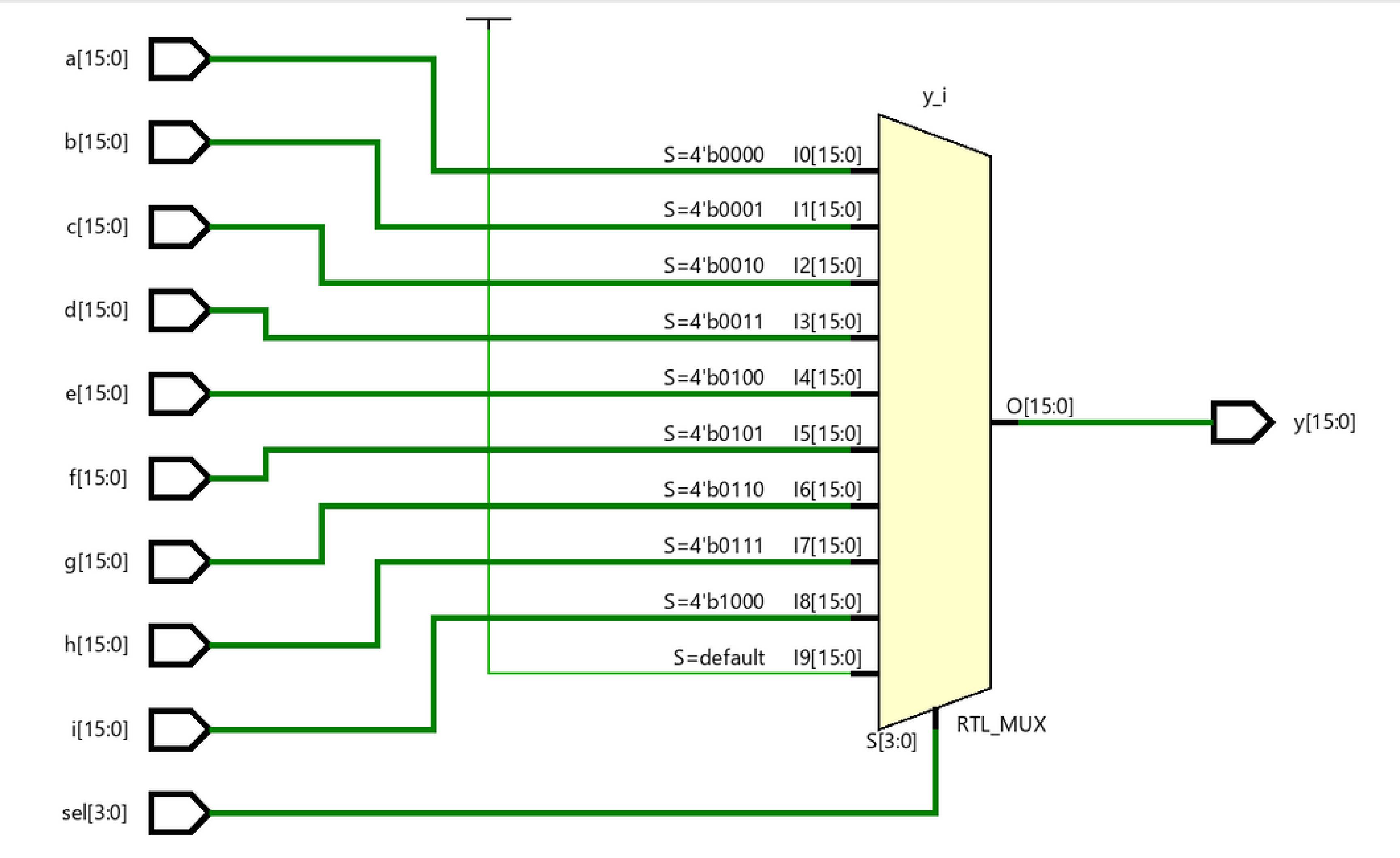
```
module mux_9x1(
input       [15:0] a,b,c,d,e,f,g,h,i,
input       [3:0]  sel,
output  reg [15:0] y )                ;

always@(*)begin
    case(sel)
        4'd0 : y = a                  ;
        4'd1 : y = b                  ;
        4'd2 : y = c                  ;
        4'd3 : y = d                  ;
        4'd4 : y = e                  ;
        4'd5 : y = f                  ;
        4'd6 : y = g                  ;
        4'd7 : y = h                  ;
        4'd8 : y = i                  ;
    default
               y = 16'hffff           ;
    endcase
end
endmodule
```

TEST BENCH

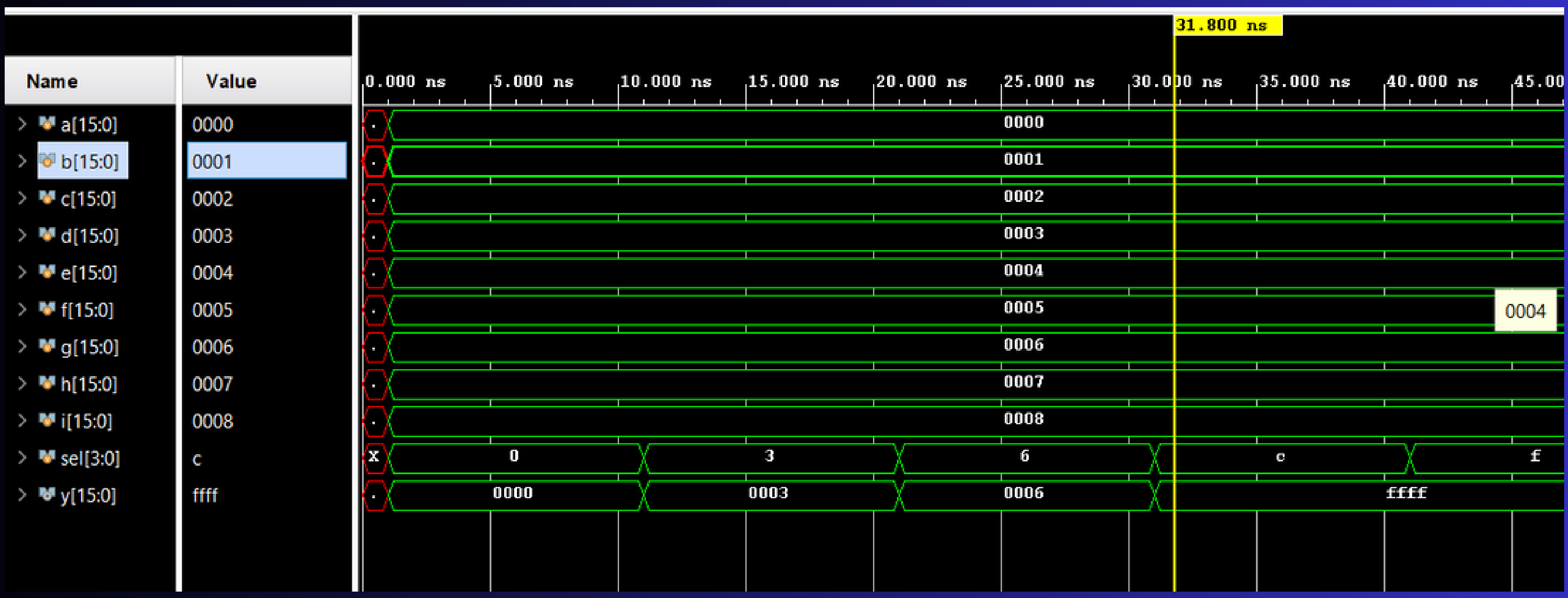


```
module mux_9x1_tb                                ;

reg  [15:0] a,b,c,d,e,f,g,h,i                    ;
reg  [3:0] sel                                   ;
wire [15:0] y                                    ;

mux_9x1 uut(.a(a),.b(b),.c(c),.d(d),.e(e),.f(f),
            .g(g),.h(h),.i(i),.sel(sel),.y(y))   ;

always begin
 #1 a = 16'd0       ;
    b = 16'd1       ;
    c = 16'd2       ;
    d = 16'd3       ;
    e = 16'd4       ;
    f = 16'd5       ;
    g = 16'd6       ;
    h = 16'd7       ;
    i = 16'd8       ;
end

initial
begin
  #1    sel = 4'd0  ;
  #10   sel = 4'd3  ;
  #10   sel = 4'd6  ;
  #10   sel = 4'd12 ;
  #10   sel = 4'd15 ;
  #10
   $finish          ;
end
endmodule
```

SCHEMATICS



SIMULATION



Name	Value
a[15:0]	0000
b[15:0]	0001
c[15:0]	0002
d[15:0]	0003
e[15:0]	0004
f[15:0]	0005
g[15:0]	0006
h[15:0]	0007
i[15:0]	0008
sel[3:0]	c
y[15:0]	ffff