数字系统设计作业报告

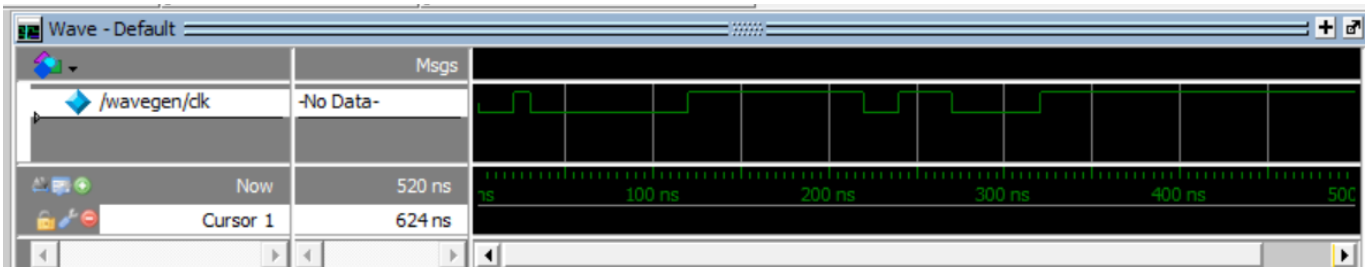
- 学号：518021910628
- 姓名：吴小茜

EXP2-1

设计/测试模块

```
// File: wavegen.v

`timescale 10 ns / 1 ns

module wavegen;

    reg clk;

    initial begin
        clk = 1'b0;
        #2 clk = 1'b1;
        #1 clk = 1'b0;
        #9 clk = 1'b1;
        #10 clk = 1'b0;
        #2 clk = 1'b1;
        #3 clk = 1'b0;
        #5 clk = 1'b1;
    #20 $stop;
    end

endmodule
```

仿真结果



EXP2-2

设计模块

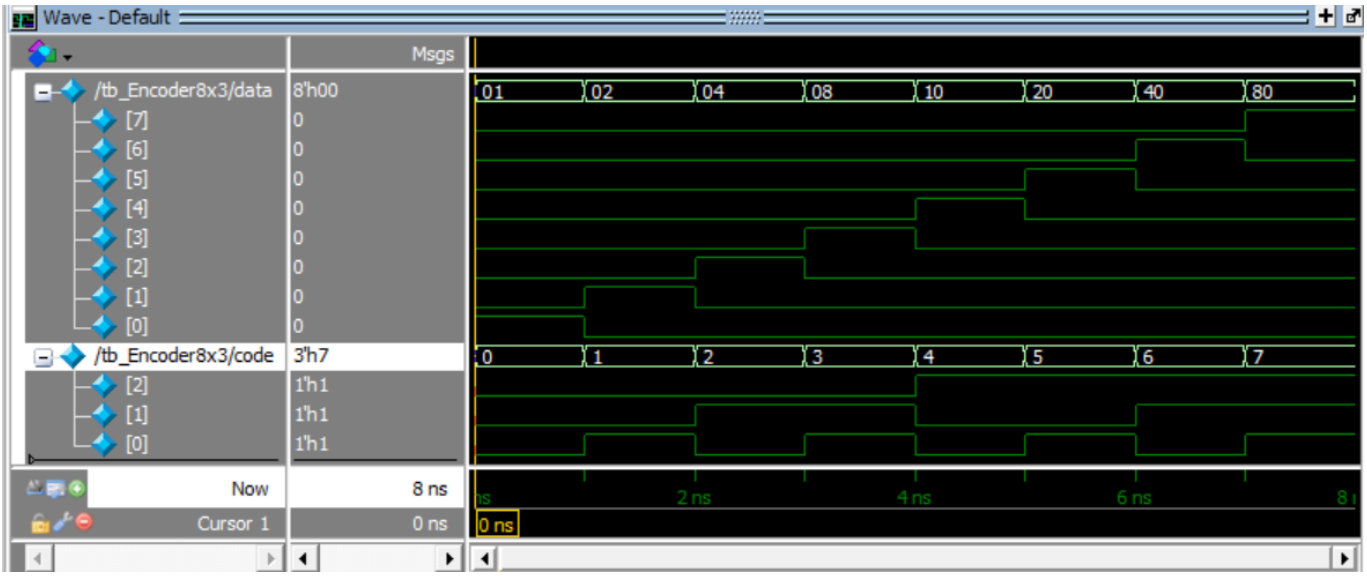
```
// File: encoder8x3.v
module Encoder8x3( output reg[2:0] code, input [7:0] data );
```

```
 always @(*)
 begin
 case ( data )
 8'b0000_0001: code = 3'd0;
 8'b0000_0010: code = 3'd1;
 8'b0000_0100: code = 3'd2;
 8'b0000_1000: code = 3'd3;
 8'b0001_0000: code = 3'd4;
 8'b0010_0000: code = 3'd5;
 8'b0100_0000: code = 3'd6;
 8'b1000_0000: code = 3'd7;
 default: code = 3'bx;
 endcase
 end
endmodule
```

测试模块

```
// File: tb_encoder8x3.v

`include "encoder8x3.v"

module tb_Encoder8x3;

    parameter STEP = 9;

    wire [2:0] code;
    reg [7:0] data;
    integer k;

    Encoder8x3 a(code, data);

    initial begin
        data = 8'b0000_0001;
        for ( k=1; k<STEP; k=k+1 )
            #1 data = data << 1;
    end

    initial begin
        $monitor("At time %4t, data=%b, code=%d",
                    $time, data, code);
    end

endmodule
```

仿真结果




```
# At time    0, data=00000001, code=0
# At time    1, data=00000010, code=1
# At time    2, data=00000100, code=2
# At time    3, data=00001000, code=3
# At time    4, data=00010000, code=4
# At time    5, data=00100000, code=5
# At time    6, data=01000000, code=6
# At time    7, data=10000000, code=7
# At time    8, data=00000000, code=x
```

EXP2-3

a) 2选1多路选择器

设计模块

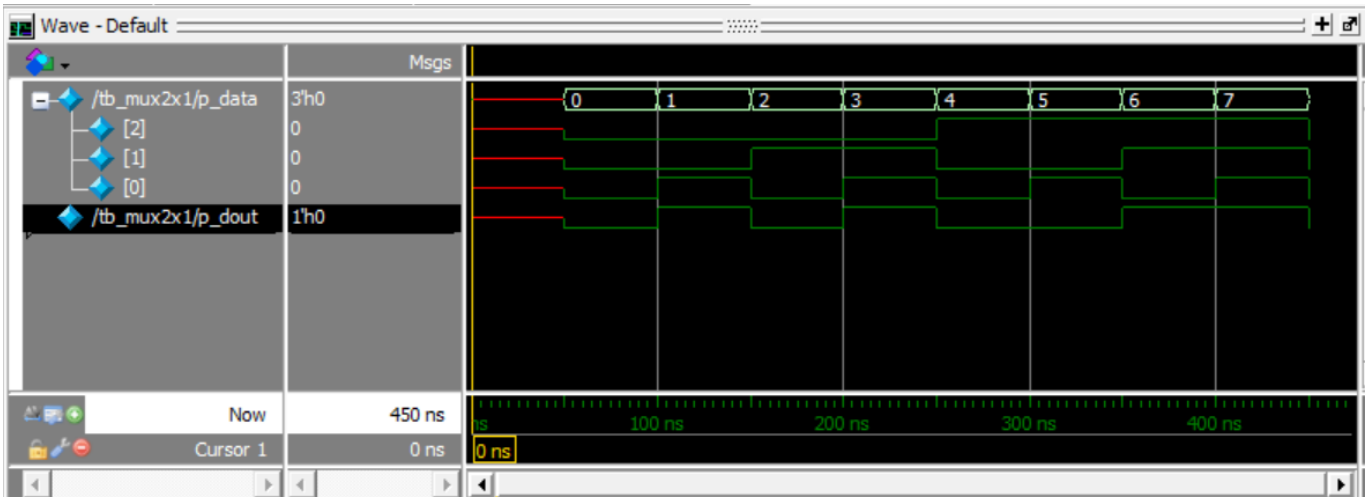
```
// File: mux2x1.v
module mux2x1( dout, sel, din );
 output dout;
 input sel;
 input [1:0] din;
 bufif1 b2( dout, din[1], sel );
 bufif0 b1( dout, din[0], sel );
endmodule
```

仿真模块

```
// File: tb_mux2x1.v
`timescale 10ns / 1ns
`include "mux2x1.v"

module tb_mux2x1();
```

```
  reg [2:0] p_data;
  wire p_dout;
  integer i;
  initial begin
  p_data = 3'bx;
  #5 p_data = 3'b0;
  for ( i = 1; i < 9; i = i + 1 )
   #5 p_data = p_data + 3'b1;
  end

  mux2x1 m0( .dout( p_dout ), .sel(p_data[2]), .din(p_data[1:0]) );
  initial
  $monitor( "At time %t, dout=%1b, sel=%1b, din=%2b", $time, p_dout, p_data[2],
p_data[1:0]);

endmodule
```

仿真结果




```
VSIM 58> run -all
# At time                    0, dout=x, sel=x, din=xx
# At time                   50, dout=0, sel=0, din=00
# At time                  100, dout=1, sel=0, din=01
# At time                  150, dout=0, sel=0, din=10
# At time                  200, dout=1, sel=0, din=11
# At time                  250, dout=0, sel=1, din=00
# At time                  300, dout=0, sel=1, din=01
# At time                  350, dout=1, sel=1, din=10
# At time                  400, dout=1, sel=1, din=11
# At time                  450, dout=0, sel=0, din=00
```

b) 4选1多路选择器

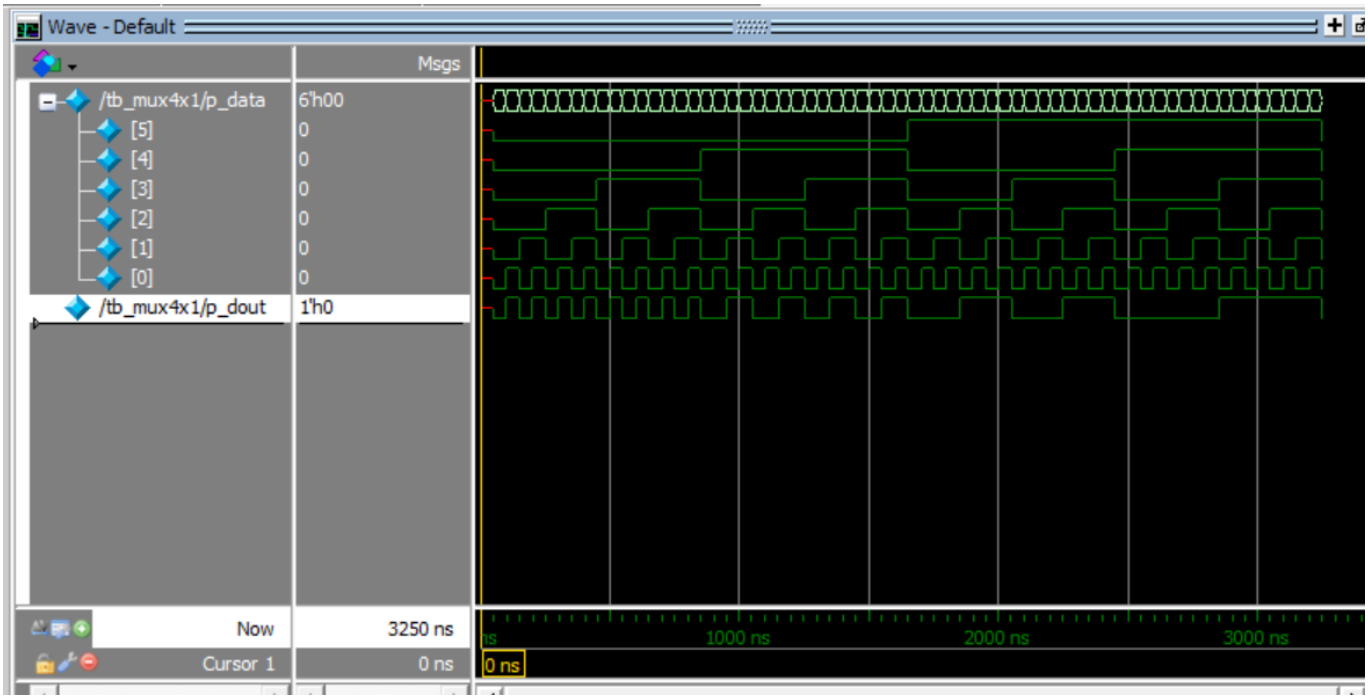
设计模块

```
// File: mux4x1.v

`include "mux2x1.v"

module mux4x1(dout, sel, din);

    output dout;
    input [1:0] sel;
    input [3:0] din;

    wire ta, tb;

    mux2x1 a(ta, sel[0], din[1:0]),
           b(tb, sel[0], din[3:2]),
           c(dout, sel[1], {tb, ta});

endmodule
```

仿真模块

```
// File: tb_mux4x1.v
`timescale 10ns / 1ns
`include "mux4x1.v"

module tb_mux4x1();
 reg [5:0] p_data;
 wire p_dout;
 integer i;
 initial begin
 p_data = 6'bx;
 #5 p_data = 6'b0;
 for ( i = 1; i < 65; i = i + 1 )
  #5 p_data = p_data + 6'b1;
 end

 mux4x1 m0( .dout( p_dout ), .sel(p_data[5:4]), .din(p_data[3:0]) );
 initial
 $monitor( "At time %t, dout=%1b, sel=%2b, din=%4b", $time, p_dout, p_data[5:4],
p_data[3:0]);

endmodule
```

仿真结果




```
VSIM 90> run -all
# At time                    0, dout=x, sel=xx, din=xxxx
# At time                   50, dout=0, sel=00, din=0000
# At time                  100, dout=1, sel=00, din=0001
# At time                  150, dout=0, sel=00, din=0010
# At time                  200, dout=1, sel=00, din=0011
# At time                  250, dout=0, sel=00, din=0100
# At time                  300, dout=1, sel=00, din=0101
# At time                  350, dout=0, sel=00, din=0110
# At time                  400, dout=1, sel=00, din=0111
# At time                  450, dout=0, sel=00, din=1000
# At time                  500, dout=1, sel=00, din=1001
# At time                  550, dout=0, sel=00, din=1010
# At time                  600, dout=1, sel=00, din=1011
# At time                  650, dout=0, sel=00, din=1100
# At time                  700, dout=1, sel=00, din=1101
# At time                  750, dout=0, sel=00, din=1110
# At time                  800, dout=1, sel=00, din=1111
# At time                  850, dout=0, sel=01, din=0000
# At time                  900, dout=0, sel=01, din=0001
# At time                  950, dout=1, sel=01, din=0010
# At time                 1000, dout=1, sel=01, din=0011
# At time                 1050, dout=0, sel=01, din=0100
# At time                 1100, dout=0, sel=01, din=0101
# At time                 1150, dout=1, sel=01, din=0110
# At time                 1200, dout=1, sel=01, din=0111
# At time                 1250, dout=0, sel=01, din=1000
# At time                 1300, dout=0, sel=01, din=1001
# At time                 1350, dout=1, sel=01, din=1010
```

```
# At time                    1400, dout=1, sel=01, din=1011
# At time                    1450, dout=0, sel=01, din=1100
# At time                    1500, dout=0, sel=01, din=1101
# At time                    1550, dout=1, sel=01, din=1110
# At time                    1600, dout=1, sel=01, din=1111
# At time                    1650, dout=0, sel=10, din=0000
# At time                    1700, dout=0, sel=10, din=0001
# At time                    1750, dout=0, sel=10, din=0010
# At time                    1800, dout=0, sel=10, din=0011
# At time                    1850, dout=1, sel=10, din=0100
# At time                    1900, dout=1, sel=10, din=0101
# At time                    1950, dout=1, sel=10, din=0110
# At time                    2000, dout=1, sel=10, din=0111
# At time                    2050, dout=0, sel=10, din=1000
# At time                    2100, dout=0, sel=10, din=1001
# At time                    2150, dout=0, sel=10, din=1010
# At time                    2200, dout=0, sel=10, din=1011
# At time                    2250, dout=1, sel=10, din=1100
# At time                    2300, dout=1, sel=10, din=1101
# At time                    2350, dout=1, sel=10, din=1110
# At time                    2400, dout=1, sel=10, din=1111
# At time                    2450, dout=0, sel=11, din=0000
# At time                    2500, dout=0, sel=11, din=0001
# At time                    2550, dout=0, sel=11, din=0010
# At time                    2600, dout=0, sel=11, din=0011
# At time                    2650, dout=0, sel=11, din=0100
# At time                    2700, dout=0, sel=11, din=0101
# At time                    2750, dout=0, sel=11, din=0110
# At time                    2800, dout=0, sel=11, din=0111
# At time                    2850, dout=1, sel=11, din=1000
# At time                    2900, dout=1, sel=11, din=1001
# At time                    2950, dout=1, sel=11, din=1010
# At time                    3000, dout=1, sel=11, din=1011
# At time                    3050, dout=1, sel=11, din=1100
# At time                    3100, dout=1, sel=11, din=1101
# At time                    3150, dout=1, sel=11, din=1110
# At time                    3200, dout=1, sel=11, din=1111
# At time                    3250, dout=0, sel=00, din=0000
```

EXP2-4

设计模块——利用门原语设计

```
// File: comb_str.v
module comb_str(output Y, input A,B,C,D);
 wire M1,M2,M3,M4;
 not u1(M1,D);
 and u4(M4,B,C,M1);
 or u3(M3,A,D);
 not u2(M2,M3);
 and u5(Y,M2,M4);
endmodule
```

设计模块——利用连续赋值语句设计

```
//File comb_dataflow.v
module comb_dataflow(output Y, input A,B,C,D);
 assign Y=(~D&B&C)&~(A|D);
endmodule
```

设计模块——利用过程语句设计

```
//File: comb_behavior.v
module comb_behavior(output reg Y, input A,B,C,D);
 always@(*) begin
  Y = ~(A | D) & (B & C & ~D);
 end
endmodule;
```

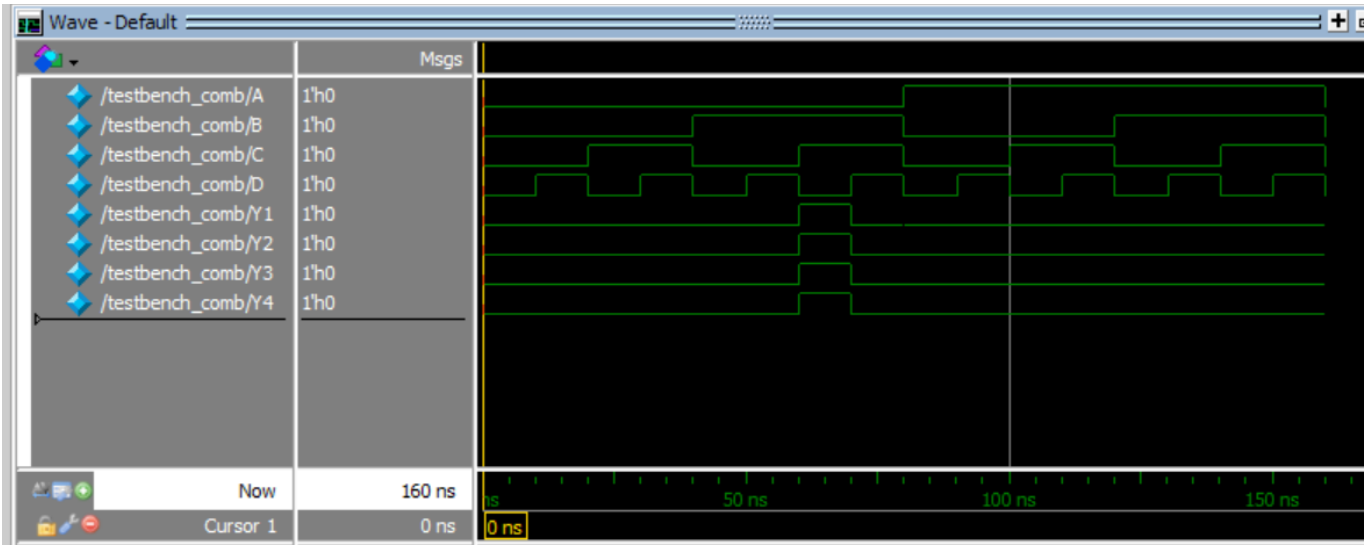
设计模块——利用UDP设计

```
// File: comb_prim.v
primitive comb_prim(output Y, input A,B,C,D);
 table
 //a b c d : y
   0 0 ? ? : 0;
   0 1 0 ? : 0;
   0 1 1 0 : 1;
   0 1 1 1 : 0;
   1 ? ? ? : 0;
 endtable
endprimitive
```

测试模块

```
// File: testbench_comb.v
`timescale 10ns / 1ns
`include "comb_str.v"
`include "comb_dataflow.v"
`include "comb_behavior.v"
`include "comb_prim.v"

module testbench_comb();
 wire Y1,Y2,Y3,Y4;
 reg A,B,C,D;
 integer k;

 comb_str a(Y1, A, B, C, D);
 comb_dataflow b(Y2, A, B, C, D);
 comb_behavior c(Y3, A, B, C, D);
 comb_prim d(Y4, A, B, C, D);

 initial begin
  {A,B,C,D}=4'b0;
  for (k=1;k<16;k=k+1)
   #1 {A, B, C, D} = {A, B, C, D} + 1'b1;
  #1 {A,B,C,D}=4'b0;
 end

 initial begin
  $monitor("At time %4t, A=%b, B=%b, C=%b, D=%b, Y1=%b, Y2=%b, Y3=%b, Y4=%b",
  $time, A, B, C, D, Y1, Y2, Y3, Y4);
 end

endmodule
```

仿真结果




```
VSIM 144> run -all
# At time     0, A=0, B=0, C=0, D=0, Y1=0, Y2=0, Y3=0, Y4=0
# At time    10, A=0, B=0, C=0, D=1, Y1=0, Y2=0, Y3=0, Y4=0
# At time    20, A=0, B=0, C=1, D=0, Y1=0, Y2=0, Y3=0, Y4=0
# At time    30, A=0, B=0, C=1, D=1, Y1=0, Y2=0, Y3=0, Y4=0
# At time    40, A=0, B=1, C=0, D=0, Y1=0, Y2=0, Y3=0, Y4=0
# At time    50, A=0, B=1, C=0, D=1, Y1=0, Y2=0, Y3=0, Y4=0
# At time    60, A=0, B=1, C=1, D=0, Y1=1, Y2=1, Y3=1, Y4=1
# At time    70, A=0, B=1, C=1, D=1, Y1=0, Y2=0, Y3=0, Y4=0
# At time    80, A=1, B=0, C=0, D=0, Y1=0, Y2=0, Y3=0, Y4=0
# At time    90, A=1, B=0, C=0, D=1, Y1=0, Y2=0, Y3=0, Y4=0
# At time   100, A=1, B=0, C=1, D=0, Y1=0, Y2=0, Y3=0, Y4=0
# At time   110, A=1, B=0, C=1, D=1, Y1=0, Y2=0, Y3=0, Y4=0
# At time   120, A=1, B=1, C=0, D=0, Y1=0, Y2=0, Y3=0, Y4=0
# At time   130, A=1, B=1, C=0, D=1, Y1=0, Y2=0, Y3=0, Y4=0
# At time   140, A=1, B=1, C=1, D=0, Y1=0, Y2=0, Y3=0, Y4=0
# At time   150, A=1, B=1, C=1, D=1, Y1=0, Y2=0, Y3=0, Y4=0
# At time   160, A=0, B=0, C=0, D=0, Y1=0, Y2=0, Y3=0, Y4=0
```