DIGITAL DESIGN

LAB8 COMBINATORIAL CIRCUIT VERILOG-SUMMARY (1)

2024 FALL TERM @ CSE . SUSETCH

WANGWG@SUSTECH.EDU.CN

LAB8

- Sequential vs Parallel in verilog
  - begin - end vd fork - join
  - The parallel in the design
- Verilog-memory
- Verilog summary(1)
  - constant, variable, input, output, wire, reg, signed, unsigned(by defalut)
  - operator
  - continuous assignment(assign, wire), procedual assignment(always@*, reg)
  - design style: data-flow, structrual(gates, modules), behavior modeling
  - module: circuit design, testbench

MEMORY IN VERILOG

- **Memory can be seen as a set of registers with the same bit width. Modeling memory by building arrays of reg variables, and addressing each unit of the array by array index.**
- Definition:

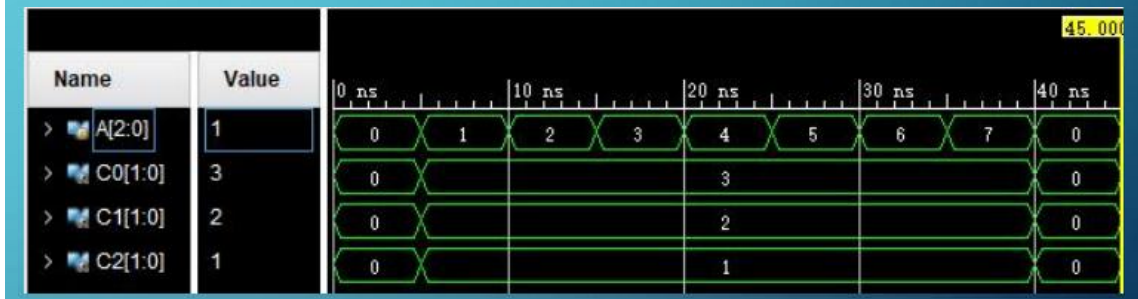
reg [n-1:0] memory name [m-1:0]; // there are **m** unit in memory, the size of each unit in the memory is **n**.

- Notes:
- **A n-bit register can be assigned in an assignment statement, but a full memory CAN NOT.**
- **If you need to read and write a storage unit in memory, you must specify the address of the unit in memory.**
- **reg [2:0] Mema [4:0];** // define a memory named Mema which has 5 memory units, each with a bit width of 3bits.
- **Mema [1]= 3'b101;** // assign 3’ b101 to Mema [1] unit in Mema

MEMORY-TEST

```
module test(
    A, C0, C1, C2
    );
        input [2:0] A;
        output [1:0] C0, C1, C2;
        reg[1:0] B [2:0];
        assign {C0, C1, C2} = {B[0], B[1], B[2]};
        always @(A)
        if(A)
        begin
            B[0] = 2'b11;
            B[1] = 2'b10;
            B[2] = 2'B01;
        end
        else
        begin
            B[0] = 2'b00;
            B[1] = 2'b00;
            B[2] = 2'B00;
        end
endmodule
```

```
module test(
    A, C0, C1, C2
    );
        input [2:0] A;
        output [1:0] C0, C1, C2;
        reg[1:0] B [2:0];
        assign {C0, C1, C2} = {B[0], B[1], B[2]};
        always @(A)
        if(A)
        begin
            {B[0], B[1], B[2]} = 6'b011011;
            /*B[0] = 2'b11;
            B[1] = 2'b10;
            B[2] = 2'B01;*/
        end
        else
        begin
            {B[0], B[1], B[2]} = 6'b0;
            /*B[0] = 2'b00;
            B[1] = 2'b00;
            B[2] = 2'B00;*/
        end
endmodule
```



WHAT'S THE SIMULATION RUNNING TIME

```
// part A
module decoder_mux_sim( );
reg sdne;
reg [1:0] sdx;
wire [3:0] sdy;
reg [15:0] smx;
reg [3:0] smsel;
wire smy;
d74139 u1(sdne,sdx,sdy);
mux16to1 u2(smx,smsel,smy);
```

```
initial begin     // part B
    {sdne,sdx} = 3'b0;
    repeat(7) #10 {sdne,sdx} = {sdne,sdx} + 1;
       #10 $finish();
end
```

```
initial begin     // part C
    smsel=4'b0;
    smx = 16'h0001;
    repeat(15) begin
        #10 smsel = smsel + 1;     smx = smx<<1;
    end
    #10 $finish();
end
```

```
endmodule  // part D
```

Q1. What's the simulation time while running the testbench on the left hand?

Q2. If move the partC ahead of partB in the testbench, will the simlation time change?

WHAT'S THE SIMULATION RUNNING TIME

```
// part A
module decoder_mux_sim( );
reg sdne;
reg [1:0] sdx;
wire [3:0] sdy;
reg [15:0] smx;
reg [3:0] smsel;
wire smy;
d74139 u1(sdne,sdx,sdy);
mux16to1 u2(smx,smsel,smy);
```

```
initial begin     // part B
    {sdne,sdx} = 3'b0;
    repeat(7) #10 {sdne,sdx} = {sdne,sdx} + 1;
       #10 $finish();
end
```

```
initial begin     // part C
    smsel=4'b0;
    smx = 16'h0001;
    repeat(15) begin
        #10 smsel = smsel + 1;     smx = smx<<1;
    end
    #10 $finish();
end
```

```
endmodule  // part D
```

Q1. What's the simulation time while running the testbench on the left hand?
A1. 80

Q2. If move the partC ahead of partB in the testbench, will the simlation time change?
A2. NO, won't change

SEQUENTIAL VS PARALLEL

```
module block2();
    reg [1:0]x,y;
    initial
    begin
        #10 x=2'd0;
        #10 x=2'd1;
        #10 x=2'd2;
        #10 x=2'd3;
    end

    initial
    fork
        #10 y=2'd0;
        #20 y=2'd1;
        #30 y=2'd2;
        #40 y=2'd3;
    join

    initial
        #50 $finish(1);
endmodule
```

Answer the following question according to the code on the left hand

- Is “block2” a design module ?
- There are three “inital” blocks in module “block2”, does the inital on the top run firstly, the initial on the buttom run lastly?
- While running the module “block2”, what’s its simulation time?
- Guess the difference between “begin-end” and “fork-join”

SEQUENTIAL BLOCK VS PARALLEL BLOCK

```
module block2();
    reg [1:0]x,y;
    initial
    begin
        #10 x=2'd0;
        #10 x=2'd1;
        #10 x=2'd2;
        #10 x=2'd3;
    end

    initial
    fork
        #10 y=2'd0;
        #20 y=2'd1;
        #30 y=2'd2;
        #40 y=2'd3;
    join

    initial
        #50 $finish(1);
endmodule
```

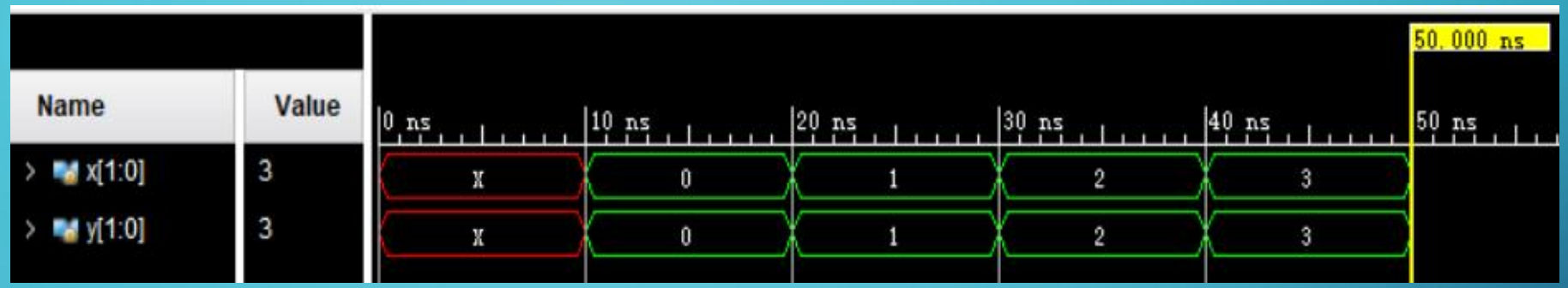
- In one module **all the block(including initial block and always block) executes at the same time(time 0)**
- **Sequential block(begin … end):**
  - **synthesizable(could be used in the circuit design)**
  - all the statements in one sequential block executes with the order of writing.
- **Parallel block(fork … join):**
  - **Not synthesizable(CAN NOT be used in the circuit design)**
  - all the statements in one parallel block executes at same time

SEQUENTIAL VS PARALLEL

```
module block2();
    reg [1:0]x,y;
    initial
    begin
        #10 x=2'd0;
        #10 x=2'd1;
        #10 x=2'd2;
        #10 x=2'd3;
    end

    initial
    fork
        #10 y=2'd0;
        #20 y=2'd1;
        #30 y=2'd2;
        #40 y=2'd3;
    join

    initial
        #50 $finish(1);
endmodule
```



Answer the following question according to the code on the left hand

- Is “block2” a design module ?
  - **NO, it’s NOT a desing module**
- There are three “inital” blocks in module “block2”, does the inital on the top run firstly, the initial on the buttom run secondly?
  - **NO, they run at the same time**
- While running the module “block2”, what’s its simulation time? **50**

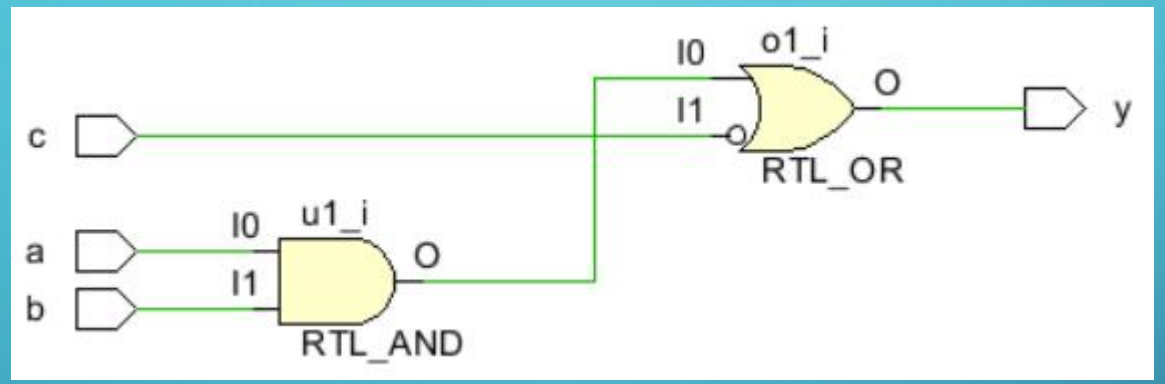
VERILOG-SUMMARY(1)

- Q1. In verilog, the constant “0” equal to “1’b0”, is it true or false?
- Q2. Could we using “123port” as the name a module, a variable or a port? Why?
- Q3. While define a variable to bind with the output port, what’s the data type of the variable?
- Q4. To define two input ports: **a** is **2**bits, **b** is **1**bit, which option(s) is(are) correct?
  - A. input a,b;
  - B. input reg a,b;
  - C. input [2:0] a,b;
  - D. input [1:0] a,b;
  - E. input [1:0]a; input b;
  - F. input a,[2:0] b;
- Q5. To define one output ports: F which is 2bits and to be assigned in an always block, which option(s) is(are) correct?
  - A. output reg [1:0] F;
  - B: output [1:0] F; reg F;
  - C: output[1:0] F; reg[1:0] F;
  - D: output [1:0] reg F;

WANGW6@SUSTECH.EDU.CN

VERILOG-SUMMARY(2)

- Q6. There are two modules at the bottom of the page, which one(s) is(are) same with the circuit bellow?




```
module demo1(input a,b,c,output y);
wire w1,nc;

not n1(nc,c);
and u1(w1,a,b);
or o1(y,w1,nc);
endmodule
```

```
module demo2(input a,b,c,output y);
wire w1,nc;

and u1(w1,a,b);
not n1(nc,c);
or o1(y,w1,nc);
endmodule
```


WANGW6@SUSTECH.EDU.CN

VERILOG-SUMMARY(3)

- Q7. Does the following pieces of verilog code relate to the same circuit?

```
module demo1(input a,b,c,output [5:0] y);
      assign y = {1'b0,a,1'b0,b,1'b0,c};
endmodule
```

```
module demo2(input a,b,c,output [5:0] y);
      assign y = 6'b0;
      assign y[0] = c;
      assign y[2] = b;
      assign y[4] = a;
endmodule
```

```
module demo3(input a,b,c,output [5:0] y);
      assign y = 6'b0;
      assign y[4] = a;
      assign y[2] = b;
      assign y[0] = c;
endmodule
```

```
module demo4(input a,b,c,output [5:0] y);
      assign y = {0,a,0,b,0,c};
endmodule
```

VERILOG-SUMMARY(4)

- Q8. Design a circuit and build a testbench to test its function. Which of the following two method(s) is(are) correct?

```
//Option1
module demo1(input a,b,c,output [5:0] y);
    assign y = {1'b0,a,1'b0,b,1'b0,c};
endmodule

module tb();
    reg a,b,c;
    wire [5:0] y;
    demo1 u1(a,b,c,y);
    initial begin
        {a,b,c} = 3'b000;
        repeat(7) #10 {a,b,c} = {a,b,c} +1;
        #10 $finish;
    end
endmodule
```

```
//Option2
module tb();
    reg a,b,c;
    wire y;

    assign y = {1'b0,a,1'b0,b,1'b0,c};

    initial begin
        {a,b,c} = 3'b000;
        repeat(7) #10 {a,b,c} = {a,b,c} +1;
        #10 $finish;
    end
endmodule
```

VERILOG-SUMMARY(5)

- Q9. Design a circuit and build a testbench to test its function. Which of the following module(s) is(are) correct?

```
//Option1
module demo1(input a,b,c,
output [5:0] y);
assign y = {1'b0,a,1'b0,b,1'b0,c};
endmodule

module tb();
reg a,b,c;
wire  [5:0]y;

demo1 u1(a,b,c,y);

initial begin
{a,b,c} = 3'b000;
repeat(7) #10 {a,b,c} = {a,b,c} +1;
#10 $finish;
end
endmodule
```

```
//Option2
module demo1(input a,b,c,
output [5:0] y);
assign y = {1'b0,a,1'b0,b,1'b0,c};
endmodule

module tb();
reg a,b,c;
wire  [5:0]y;

demo1 u1(y,c,b,a);

initial begin
{a,b,c} = 3'b000;
repeat(7) #10 {a,b,c} = {a,b,c} +1;
#10 $finish;
end
endmodule
```

```
//Option3
module demo1(input a,b,c,
output [5:0] y);
assign y = {1'b0,a,1'b0,b,1'b0,c};
endmodule

module tb();
reg a,b,c;
wire  [5:0]y;

demo1 u1(.a(a), .b(b), .c(c), .y(y));

initial begin
{a,b,c} = 3'b000;
repeat(7) #10 {a,b,c} = {a,b,c} +1;
#10 $finish;
end
endmodule
```

```
//Option4
module demo1(input a,b,c,
output [5:0] y);
assign y = {1'b0,a,1'b0,b,1'b0,c};
endmodule

module tb();
reg a,b,c;
wire  [5:0]y;

demo1 u1(.y(y), .c(c), .b(b), .a(a));

initial begin
{a,b,c} = 3'b000;
repeat(7) #10 {a,b,c} = {a,b,c} +1;
#10 $finish;
end
endmodule
```

VERILOG-SUMMARY(6)

- Q10.Which of the following module(s) is(are) correct?

```
//Option1
module demo1(input a,b,c,
output [5:0] y);
assign y = {1'b0,a,1'b0,b,1'b0,c};
endmodule

module top(input [3:0] in1,
output [5:0] y);

demo1 u1( .a(in1[2]), .b(in1[1]),
 .c(in1[0]),  .y(y) );

wire [2:0] x;
assign  x = 3'b000;

demo1
u1(.a(x[2]), .b(x[1]), .c(x[0]),
.y(y) );

endmodule
```

```
//Option2
module demo1(input a,b,c,
output [5:0] y);
assign y = {1'b0,a,1'b0,b,1'b0,c};
endmodule

module top(input [3:0] in1,
output reg [5:0] y);

wire [5:0]y1;
demo1 u1( .a(in1[2]),  .b(in1[1]),
.c(in1[0]),  .y(y1) );
always@*
if( in1[3] ==1'b1)
y=y1;
else
y = 6'b00_0000;

endmodule
```

```
//Option3
module demo1(input a,b,c,
output [5:0] y);
assign y = {1'b0,a,1'b0,b,1'b0,c};
endmodule

module top(input [3:0] in1,
output [5:0] y);

always@*
if( in1[3] ==1'b1)
demo1 u1(.a(in1[2]),  .b(in1[1]),
.c(in1[0]),   .y(y) );
else
y = 6'b00_0000;

endmodule
```

```
//Option4
module demo1(input a,b,c,
output [5:0] y);
assign y = {1'b0,a,1'b0,b,1'b0,c};
endmodule

module top(input [3:0] in1,
output [5:0] y);

always@*
if( in1[3] ==1'b1)
demo1 u1( .a(in1[2]),  .b(in1[1]),
.c(in1[0]),  .y(y[5:0]) );
else  begin
assign  x = 3'b000;
demo1  u1(.a(x[2]), .b(x[1]),
.c(x[0]),  .y(y) );
end

endmodule
```


WANGW6@SUSTECH.EDU.CN

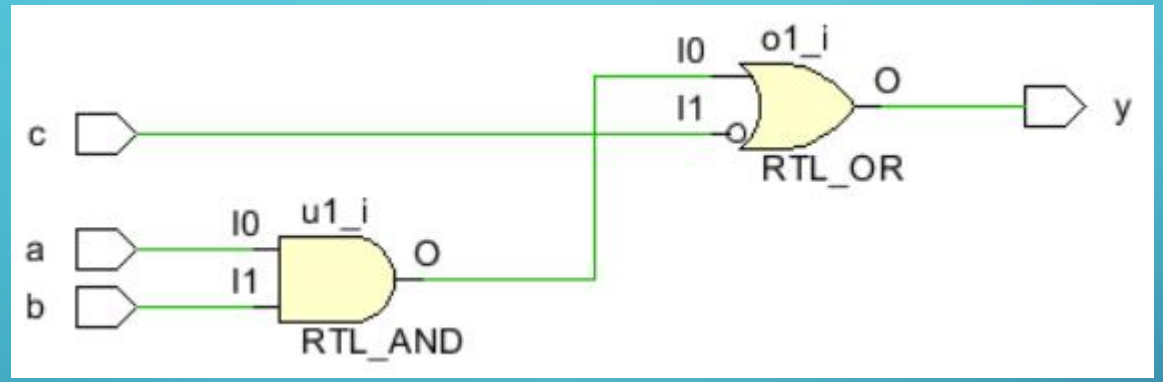
VERILOG-SUMMARY(1)

- Q1. In verilog, the constant “0” equal to “1’b0”, is it true or false?
  - **False**
- Q2. Could we using “123port” as the name a module, a variable or a port? Why?
  - No, **the name of module, variable and port CAN NOT start with number**.
- Q3. While define a variable to bind with the output port, what’s the data type of the variable?
  - **wire**
- Q4. To define two input ports: **a** is **2**bits, **b** is **1**bit, which option(s) is(are) correct?
  - A. input a,b;
  - B. input reg a,b;
  - C. input [2:0] a,b;
  - D. input [1:0] a,b;
  - E. **input [1:0]a; input b;**
  - F. input a,[2:0] b;
- Q5. To define one output ports: F which is 2bits and to be assigned in an always block, which option(s) is(are) correct?

  A. **output reg [1:0] F;** B: output [1:0] F; reg F; C: **output[1:0] F; reg[1:0] F;** D: output [1:0] reg F;

VERILOG-SUMMARY(2)

- Q6. There are two modules at the bottom of the page, which one(s) is(are) same with the circuit bellow? demo1 and demo2 are same




```
module demo1(input a,b,c,output y);
wire w1,nc;

not n1(nc,c);
and u1(w1,a,b);
or o1(y,w1,nc);
endmodule
```

```
module demo2(input a,b,c,output y);
wire w1,nc;

and u1(w1,a,b);
not n1(nc,c);
or o1(y,w1,nc);
endmodule
```


WANGW6@SUSTECH.EDU.CN

VERILOG-SUMMARY(3)

- Q7. Does the following pieces of verilog code relate to the same circuit?
  - No, only demo2 and demo3 are same, but they are illegal.

```
module demo1(input a,b,c,output [5:0] y);
      assign y = {1'b0,a,1'b0,b,1'b0,c};
endmodule
```

```
module demo2(input a,b,c,output [5:0] y);
      assign y = 6'b0;
      assign y[0] = c;
      assign y[2] = b;
      assign y[4] = a;
endmodule
```

```
module demo3(input a,b,c,output [5:0] y);
      assign y = 6'b0;
      assign y[4] = a;
      assign y[2] = b;
      assign y[0] = c;
endmodule
```

```
module demo4(input a,b,c,output [5:0] y);
      assign y = {0,a,0,b,0,c};
endmodule
```

VERILOG-SUMMARY(4)

- Q8. Design a circuit and build a testbench to test its function. Which of the following two method(s) is(are) correct?

```
//Option1
module demo1(input a,b,c,output [5:0] y);
    assign y = {1'b0,a,1'b0,b,1'b0,c};
endmodule

module tb();
    reg a,b,c;
    wire [5:0] y;
    demo1 u1(a,b,c,y);
    initial begin
        {a,b,c} = 3'b000;
        repeat(7) #10 {a,b,c} = {a,b,c} +1;
        #10 $finish;
    end
endmodule
```

```
//Option2
module tb();
    reg a,b,c;
    wire y;

    assign y = {1'b0,a,1'b0,b,1'b0,c};

    initial begin
        {a,b,c} = 3'b000;
        repeat(7) #10 {a,b,c} = {a,b,c} +1;
        #10 $finish;
    end
endmodule
```

VERILOG-SUMMARY(5)

- Q9. Design a circuit and build a testbench to test its function. Which of the following module(s) is(are) correct?

//Option1

```
module demo1(input a,b,c,
output [5:0] y);
assign y = {1'b0,a,1'b0,b,1'b0,c};
endmodule

module tb();
reg a,b,c;
wire [5:0]y;

demo1 u1(a,b,c,y);

initial begin
{a,b,c} = 3'b000;
repeat(7) #10 {a,b,c} = {a,b,c} +1;
#10 $finish;
end
endmodule
```

//Option2

```
module demo1(input a,b,c,
output [5:0] y);
assign y = {1'b0,a,1'b0,b,1'b0,c};
endmodule

module tb();
reg a,b,c;
wire  [5:0]y;

demo1 u1(y,c,b,a);

initial begin
{a,b,c} = 3'b000;
repeat(7) #10 {a,b,c} = {a,b,c} +1;
#10 $finish;
end
endmodule
```

//Option3

```
module demo1(input a,b,c,
output [5:0] y);
assign y = {1'b0,a,1'b0,b,1'b0,c};
endmodule

module tb();
reg a,b,c;
wire  [5:0]y;

demo1 u1(.a(a), .b(b), .c(c), .y(y));

initial begin
{a,b,c} = 3'b000;
repeat(7) #10 {a,b,c} = {a,b,c} +1;
#10 $finish;
end
endmodule
```

//Option4

```
module demo1(input a,b,c,
output [5:0] y);
assign y = {1'b0,a,1'b0,b,1'b0,c};
endmodule

module tb();
reg a,b,c;
wire  [5:0]y;

demo1 u1(.y(y), .c(c), .b(b), .a(a));

initial begin
{a,b,c} = 3'b000;
repeat(7) #10 {a,b,c} = {a,b,c} +1;
#10 $finish;
end
endmodule
```

VERILOG-SUMMARY(6-1)

- Q10.Which of the following module(s) is(are) correct? //option2

```
//Option1
module demo1(input a,b,c,
output [5:0] y);
assign y = {1'b0,a,1'b0,b,1'b0,c};
endmodule

module top(input [3:0] in1,
output [5:0] y);

demo1 u1( .a(in1[2]), .b(in1[1]),
 .c(in1[0]),  .y(y) );

wire [2:0] x;
assign  x = 3'b000;

demo1
u1(.a(x[2]), .b(x[1]), .c(x[0]),
.y(y) );

endmodule
```

```
//Option2
module demo1(input a,b,c,
output [5:0] y);
assign y = {1'b0,a,1'b0,b,1'b0,c};
endmodule

module top(input [3:0] in1,
output reg [5:0] y);

wire [5:0]y1;
demo1 u1( .a(in1[2]),  .b(in1[1]),
.c(in1[0]),  .y(y1) );
always@*
if( in1[3] ==1'b1)
y=y1;
else
y = 6'b00_0000;

endmodule
```

```
//Option3
module demo1(input a,b,c,
output [5:0] y);
assign y = {1'b0,a,1'b0,b,1'b0,c};
endmodule

module top(input [3:0] in1,
output [5:0] y);

always@*
if( in1[3] ==1'b1)
demo1 u1(.a(in1[2]),  .b(in1[1]),
.c(in1[0]),   .y(y) );
else
y = 6'b00_0000;

endmodule
```

```
//Option4
module demo1(input a,b,c,
output [5:0] y);
assign y = {1'b0,a,1'b0,b,1'b0,c};
endmodule

module top(input [3:0] in1,
output [5:0] y);

always@*
if( in1[3] ==1'b1)
demo1 u1( .a(in1[2]),  .b(in1[1]),
.c(in1[0]),  .y(y[5:0]) );
else  begin
assign  x = 3'b000;
demo1  u1(.a(x[2]), .b(x[1]),
.c(x[0]),  .y(y) );
end

endmodule
```


WANGW6@SUSTECH.EDU.CN

VERILOG-SUMMARY(6-2)

- Q10.Which of the following module(s) is(are) correct?

```
//Option1
module demo1(input a,b,c,
output [5:0] y);
assign y = {1'b0,a,1'b0,b,1'b0,c};
endmodule

module top(input [3:0] in1,
output [5:0] y);

demo1 u1( .a(in1[2]), .b(in1[1]),
 .c(in1[0]),  .y(y) );

wire [2:0] x;
assign  x = 3'b000;

demo1 u1(.a(x[2]), .b(x[1]),
.c(x[0]),  .y(y) );

endmodule
```

Error !

the output port y of top has multiple driver:

- from the output port y of u1
- from the output port y of u2

Error !

because a module could be instanced for several times, but can't share the same instrance name!

WANGW6@SUSTECH.EDU.CN

VERILOG-SUMMARY(6-3)

- Q10.Which of the following module(s) is(are) correct?

```
//Option3
module demo1(input a,b,c,
output [5:0] y);
assign y = {1'b0,a,1'b0,b,1'b0,c};
endmodule

module top(input [3:0] in1,
output [5:0] y);

always@*
if( in1[3] ==1'b1)
demo1 u1(.a(in1[2]),  .b(in1[1]),
.c(in1[0]),   .y(y) );
else
y = 6'b00_0000;

endmodule
```

Error !

Structured modeling statements cannot be nested in behavior modeling statements

WANGW6@SUSTECH.EDU.CN

VERILOG-SUMMARY(6-4)

- Q10.Which of the following module(s) is(are) correct?

```
//Option4
module demo1(input a,b,c,
output [5:0] y);
assign y = {1'b0,a,1'b0,b,1'b0,c};
endmodule

module top(input [3:0] in1,
output [5:0] y);

always@*
if( in1[3] ==1'b1)
demo1 u1( .a(in1[2]),  .b(in1[1]),
.c(in1[0]),  .y(y[5:0]) );
else  begin
assign  x = 3'b000;
demo1  u1(.a(x[2]), .b(x[1]),
.c(x[0]),  .y(y) );
end

endmodule
```

Error !

Both Structured modeling statements and data-flow modeling statements cannot be nested in behavior modeling statements

WANGW6@SUSTECH.EDU.CN