Lab2 Notes

zjx@ustc.edu.cn

2019.4.4

REG型变量

- **申明为reg型的变量，综合后不一定生成寄存器！**

```
wire [7:0]  a, b;
reg [7:0]  r1, r2, r3;
```

```
always @(*)    // (en, a, b)
    if (en)  r1 = a;
    else  r1= b;
```

```
always @(en, a)    // @(*)
    if (en)  r2 = a;
```

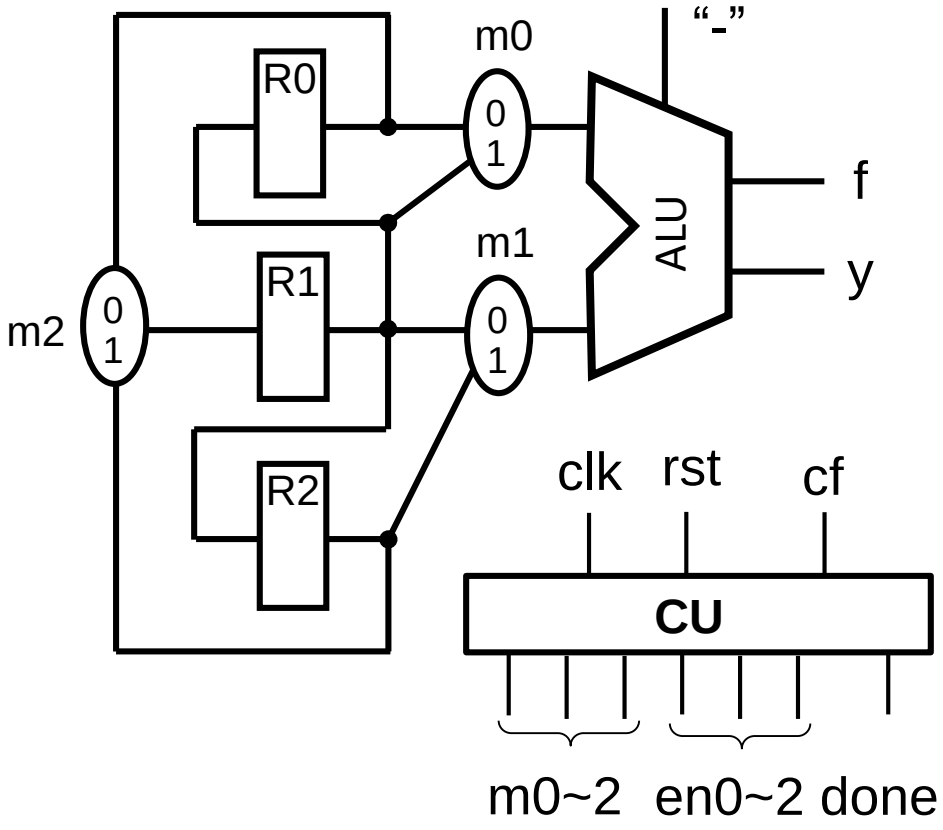
```
always @(posedge  clk)
    if (en)  r3 = a;
```

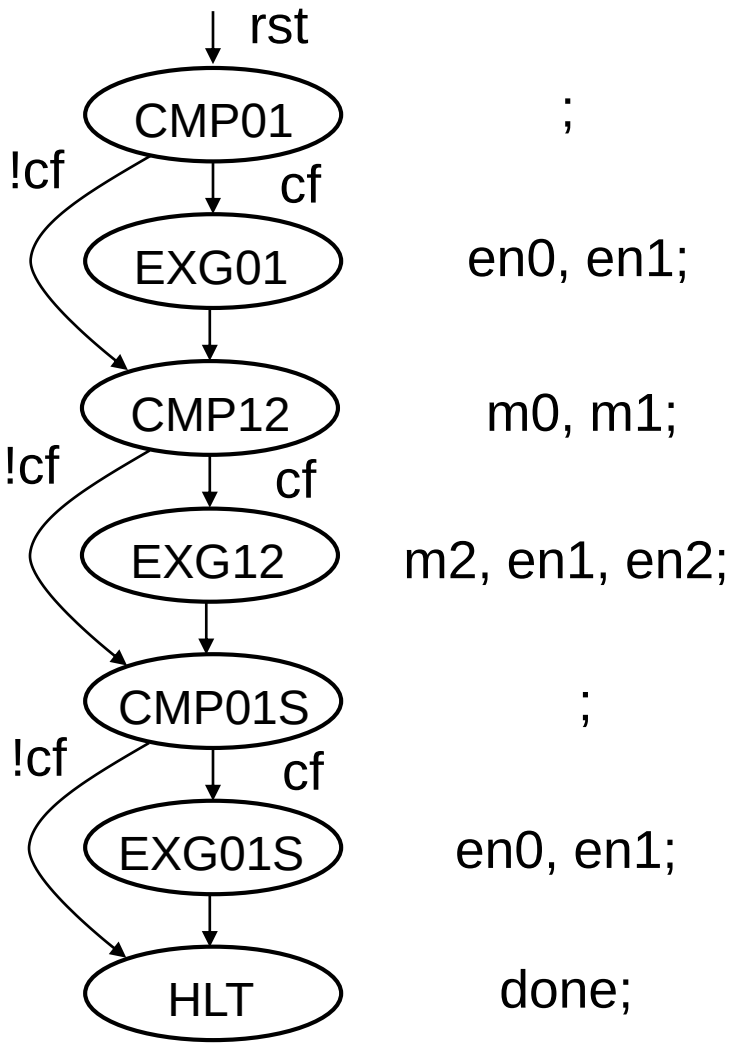
组合电路：
敏感变量不要遗漏
所有分支均有赋值
不能含反馈，如r1=r1+1

锁存器：尽量避免使用

寄存器：必有触发时钟

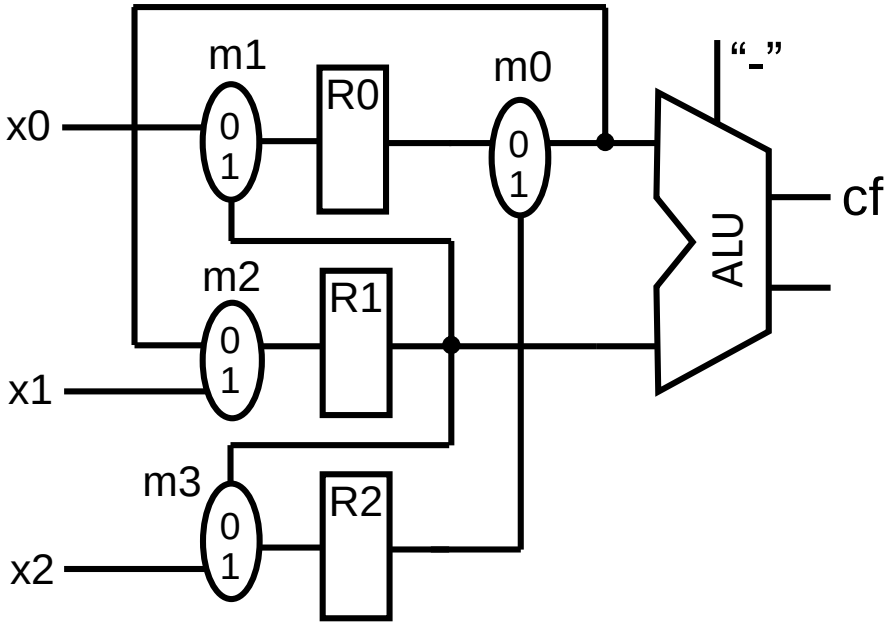
示例：三个数排序

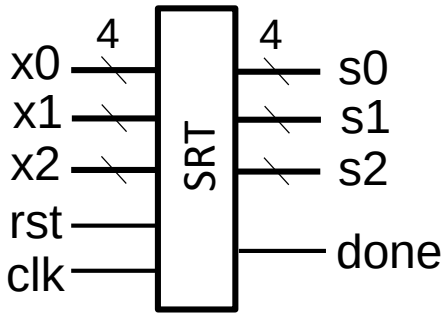


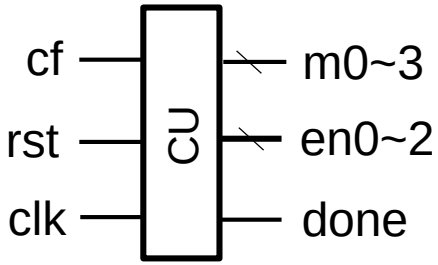


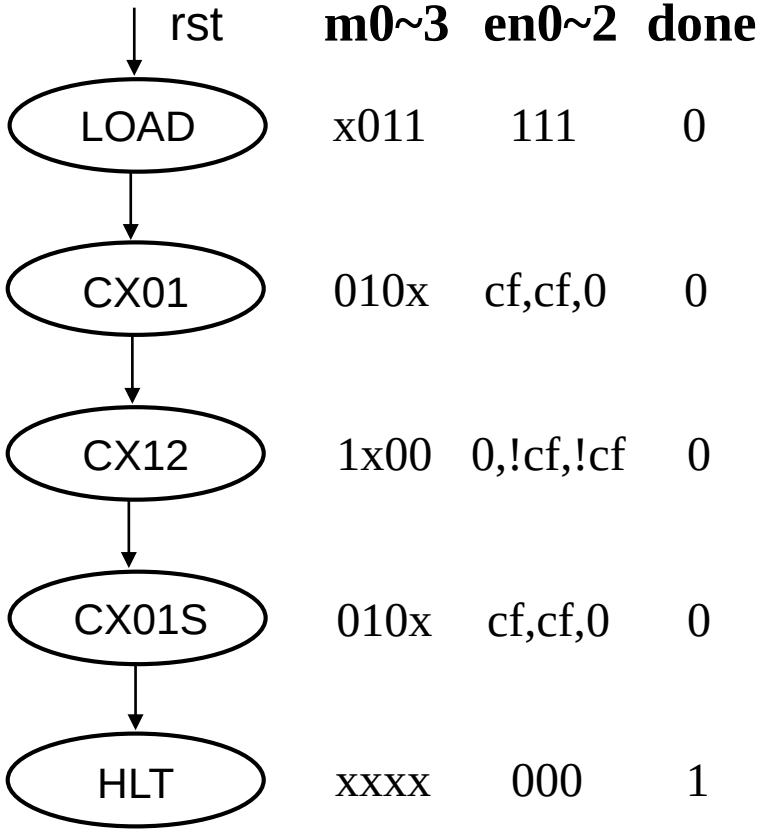
思考题：如何优化？

示例：三个数排序


m1
m0
"-"
x0
R0
0
1
ALU
cf
m2
R1
x1
m3
R2
x2


4
4
x0
x1
x2
rst
clk
SRT
s0
s1
s2
done


cf
rst
clk
CU
m0~3
en0~2
done

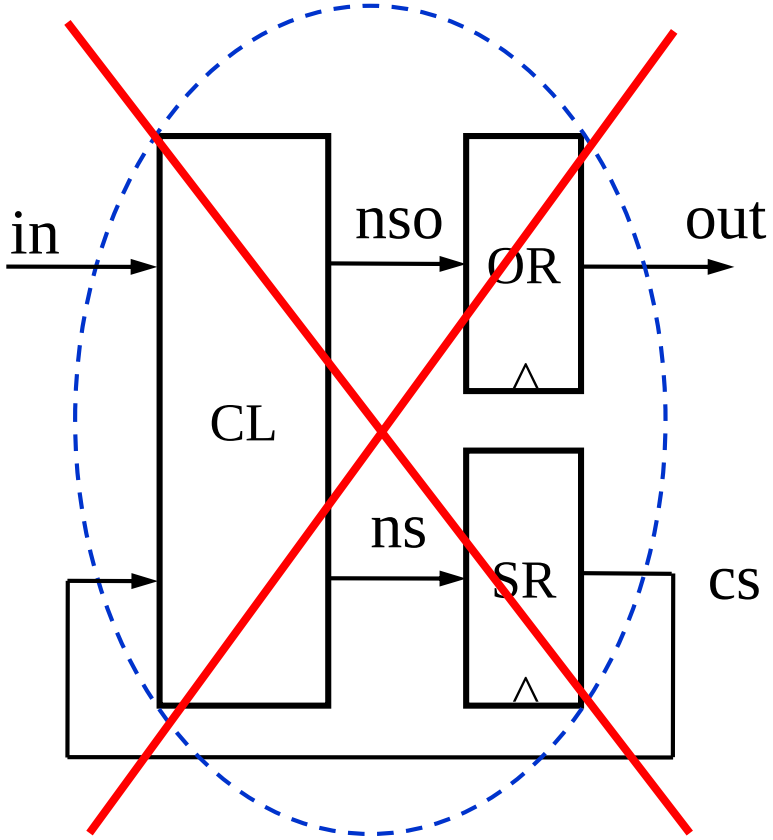

rst
m0~3
en0~2
done
LOAD
x011
111
0
CX01
010x
cf,cf,0
0
CX12
1x00
0,!cf,!cf
0
CX01S
010x
cf,cf,0
0
HLT
xxxx
000
1

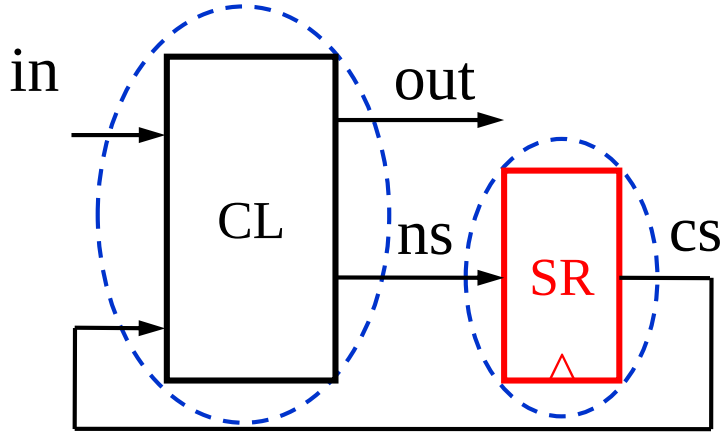
FSM描述方式

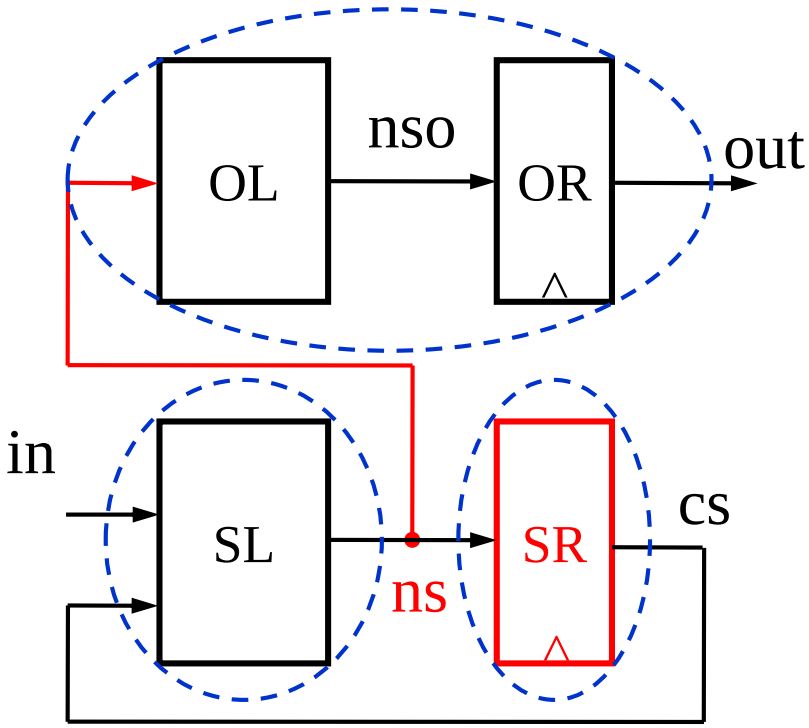
一段式

两段式

三段式

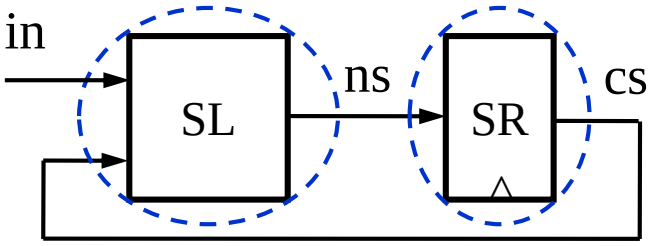






示例：三个数排序 (续1)

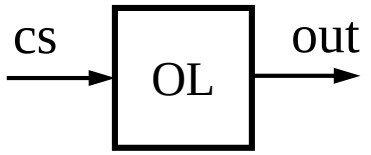
```
wire [4:0]  i0, i1, i2, r0, r1, r2, a;

// Data Path
register  R0 (i0, en0, rst, clk, r0),
          R1 (i1, en1, rst, clk, r1),
          R2 (i2, en2, rst, clk, r2);
alu    ALU (a, r1, SUB, cf);
mux  M0 (m0, r0, r2, a),
       M1 (m1, x0, r1, i0),
       M2 (m2, a, x1, i1),
       M3 (m3, r1, x2, i2);

// Control Unit
always @(posedge clk, posedge rst)
      if (rst) current_state <= LOAD;
      else
        current_state <= next_state;
```

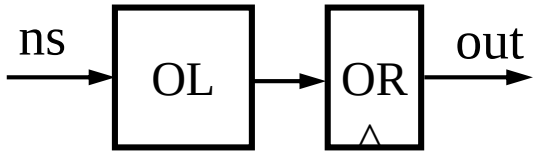



```
always @(*) begin
  case (current_state)
    LOAD: next_state = CX01;
    CX01: next_state = CX12;
    CX12: next_state = CX01S;
    CX01S: next_state = HLT;
    HLT: next_state = HLT;
    default: next_state = HLT;
  endcase
end
```

示例：三个数排序 (续2)

```
// 两段式：控制信号--组合输出
always @(*) begin
    {m0, m1, m2, m3, en0, en1, en2, done} = 8'h0;
    case (current_state)
        LOAD: {m2, m3, en0, en1, en2} = 5b'11111;
        CX01, CX01S: begin m1 = 1; en0 = cf; en1 = cf; end
        CX12: begin m0 = 1; en1 = ~cf; en2 = ~cf; end
        HLT: done = 1;
    endcase
end
```



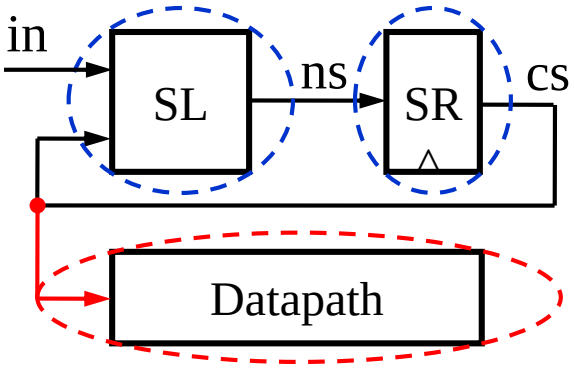


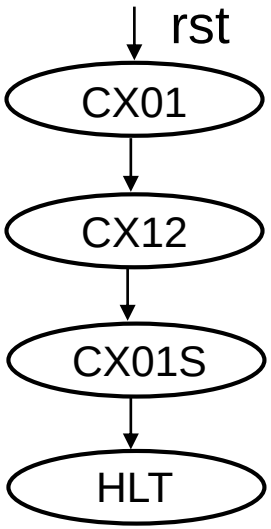

```
// 三段式：控制信号--寄存器输出
always @(posedge clk, posedge rst)  begin
    if (rst) {m1, m2, m3, en0, en1, en2, done} <= 7b'011_1110;  // LOAD
    else   case (next_state)
        CX01, CX01S: {m0, m1, m2, en0, en1, en2} <= {3b'010, cf, cf, 1'b0};
        ……
```

有何问题？如何解决？

示例：三个数排序 (续3)

```
// 将数据通路和控制信号合二为一
reg [4:0]  r0, r1, r2;
always @(posedge clk, posedge rst)  begin
if (rst)  begin
      r0 <= x0; r1 <= x1; r2 <= x2;
end
else
      case (state)
      CX01, CX01S:
            If (r1 > r0)  begin r0 <= r1; r1 <= r0; end
      CX12:
            If (r2 > r1)  begin r1 <= r2; r2 <= r1; end
      HLT: done <= 1;
      endcase
end
```

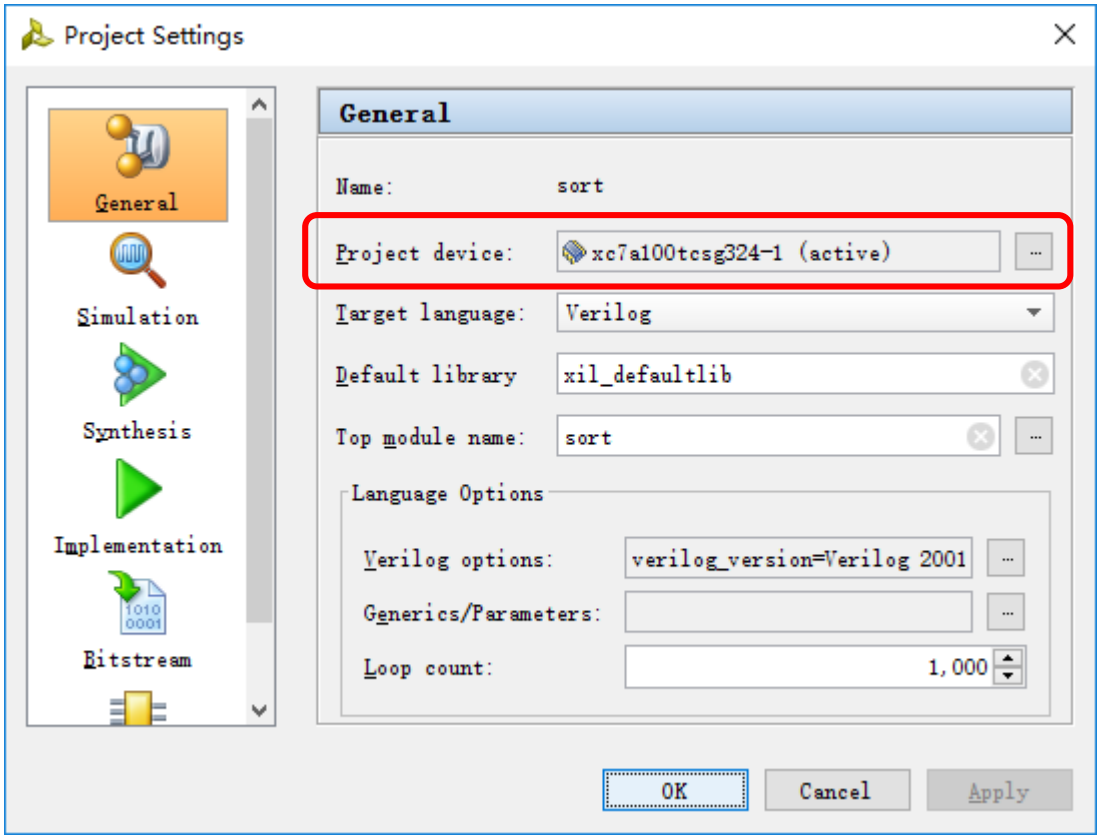




下载时端口映射问题

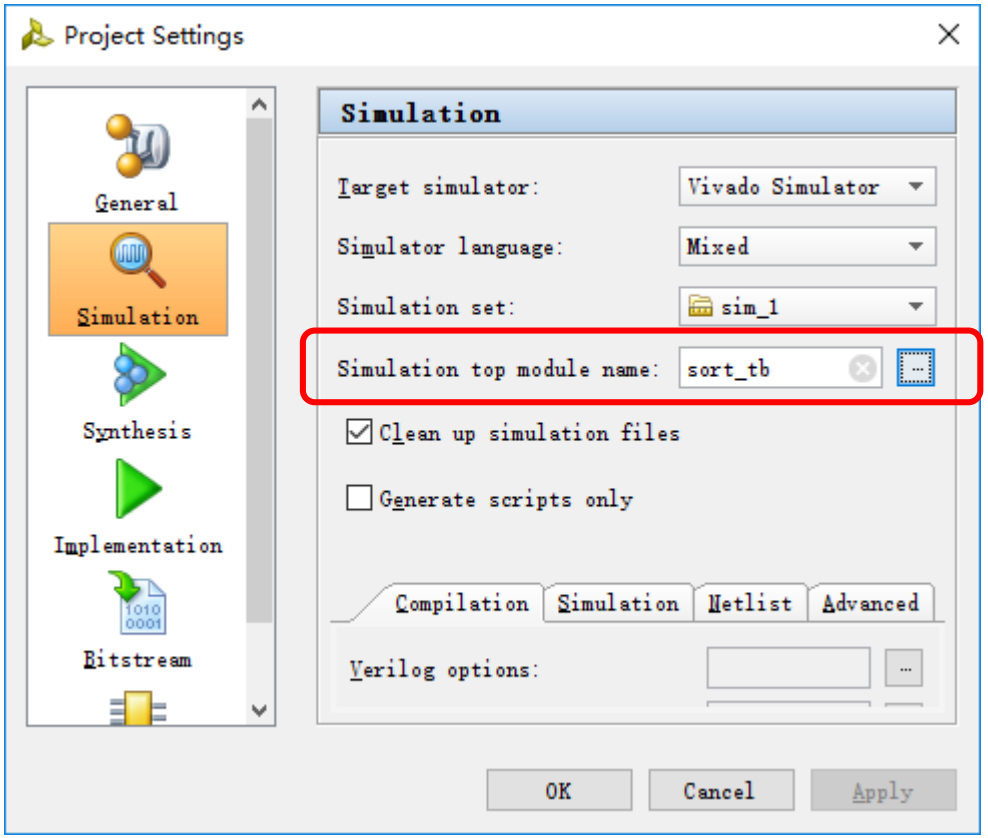
- **Flow Navigator窗口 >> Project Manager >> Project Settings**
  - General >>
  - Project device

xc7a100tcsg324-1



仿真时模块选择问题

- **Flow Navigator窗口 >> Project Manager >> Project Settings**
  - Simulation >>
  - Simulation top module name



The End