# 数字系统设计 实验课 2020.3.9

# LAB1

*Goal*

To learn how to use the Xilinx circuit design software to design and implement digital circuits.

*Procedure*

Design a 2-1 Mux with the Xilinx software. Connect the input a, b, s to three switches. Connect the y output to an LED. Implement the circuit onto the FPGA board to test and verify that it works correctly.

Design it with AND/OR/NOT gate level Verilog description.

Design it with Boolean function Verilog description.

Design it with Always statement and if-else statement.

Design it with always statement and case statement.

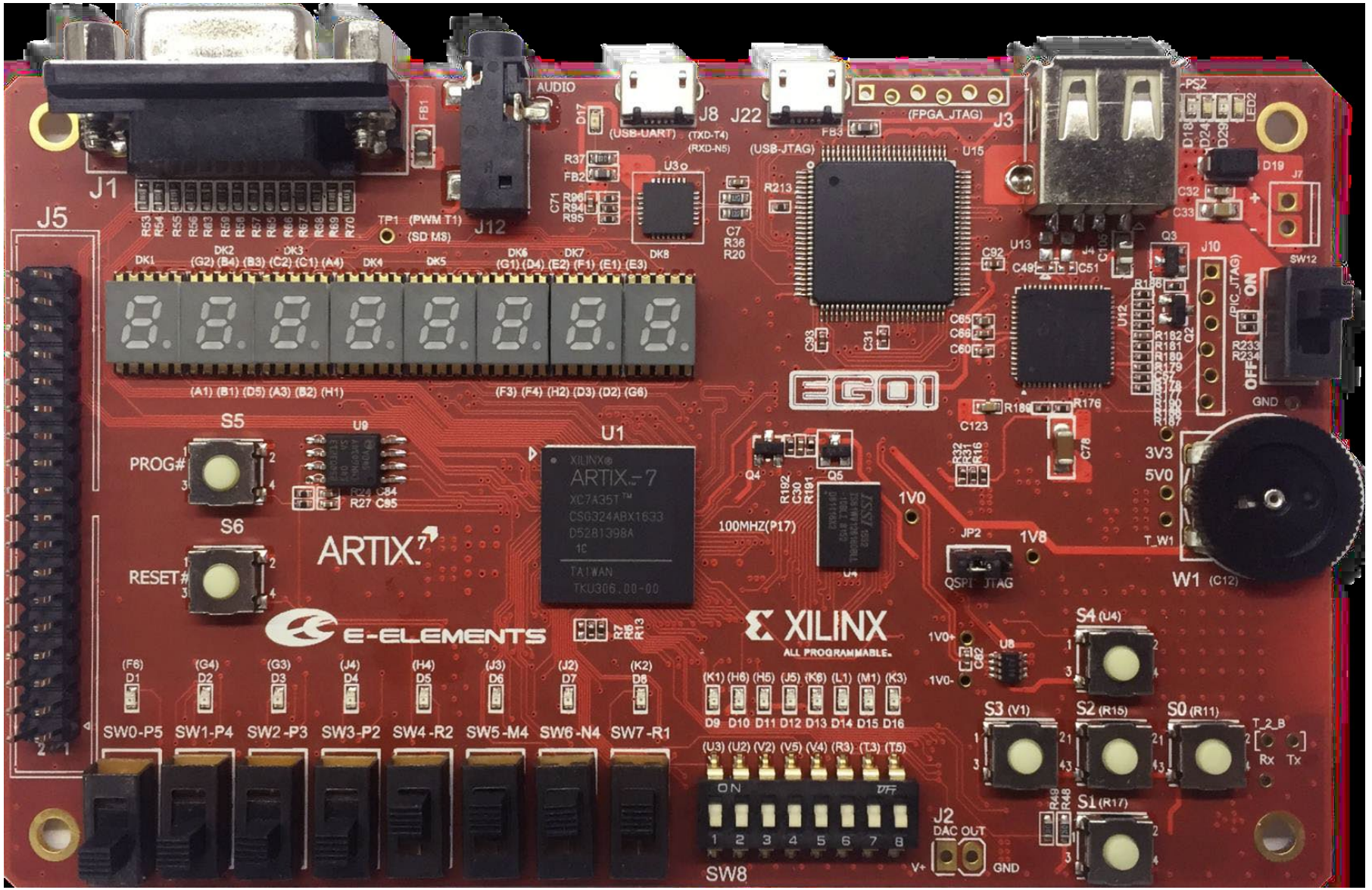
Design 4-1 Mux by using hierarchy structure of the module 2-1 Mux designed above.

# 目录

# 0. 开发板基本模块熟悉

**FPGA：** Xilinx Artix-7 XC7A35T
**存储器：**
SRAM：2Mbit
SPI Flash：N25Q032A
**通用IO：**
Switch：x8
LED：x16
Button：x5
DIP：x8
**通用扩展IO：** 32pin
**音视频/显示：**
7段数码管：x8
VGA视频输出接口
Audio音频接口
**通信接口：**
UART：USB转UART
Bluetooth：蓝牙模块
**模拟接口：**
DAC: 8-bit分辨率
XADC: 2路12bit 1Msps ADC

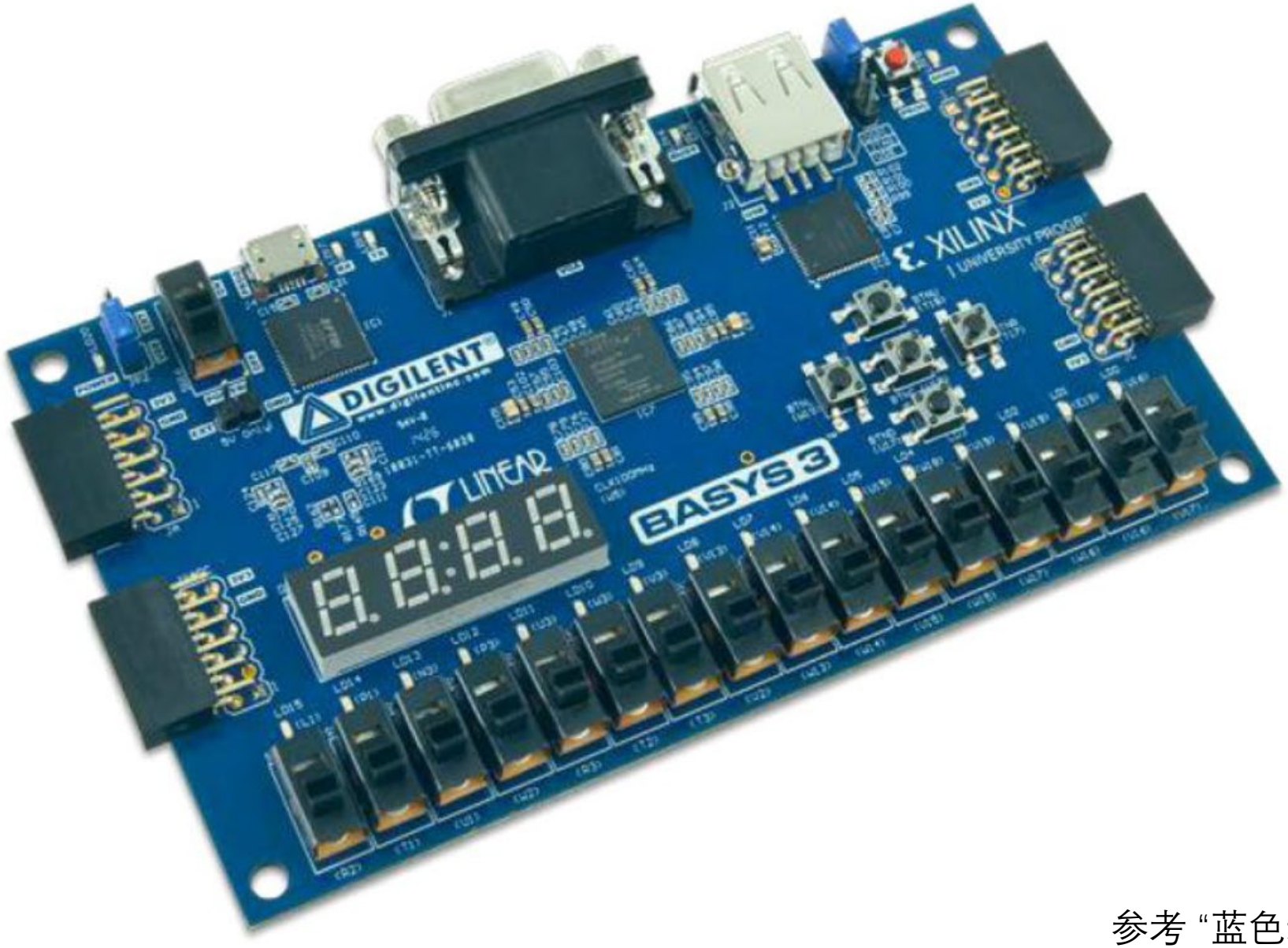
EGO1




参考 “红色新板子_EGo1资料”

# 0. 开发板基本模块熟悉

BASYS 3



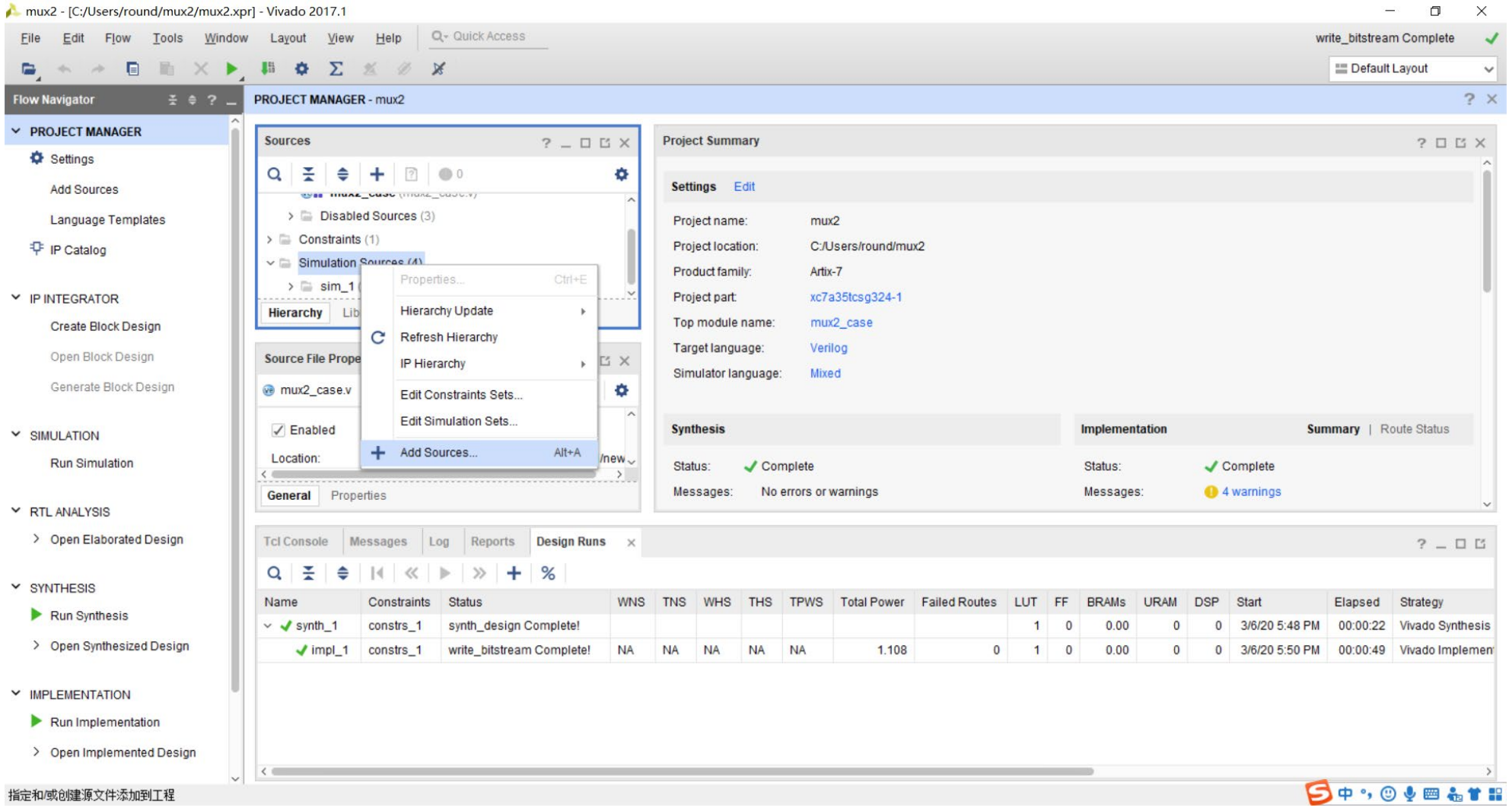

参考 “蓝色老板子_Basys3”

参考 “vivado2017.4使用手册”

# 1. 2-1选择器设计　　1.2仿真

## 创建仿真源文件

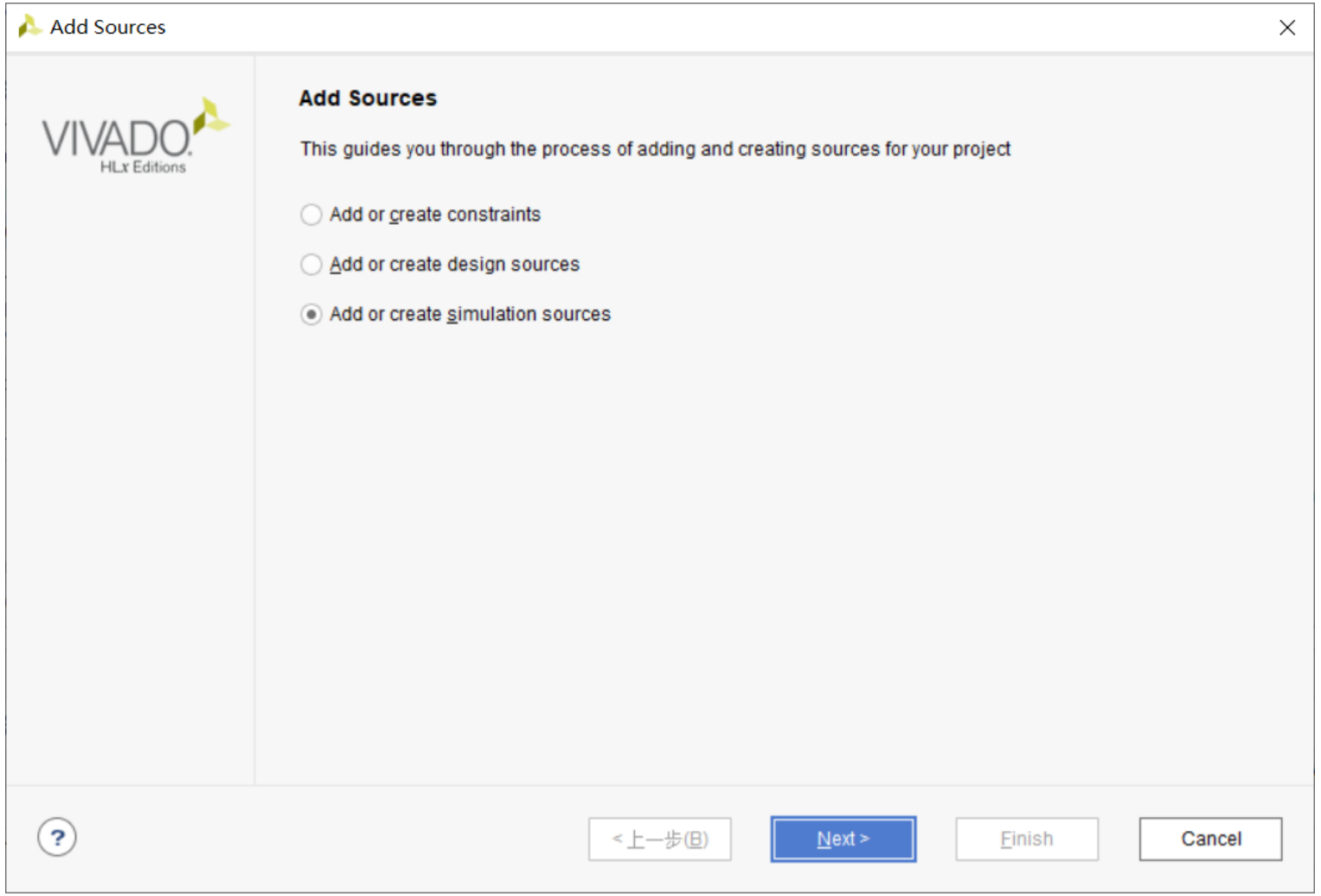
1、右键simulation Source，点击Add sources



## 1. 2-1选择器设计　　1.2仿真

## 创建仿真源文件

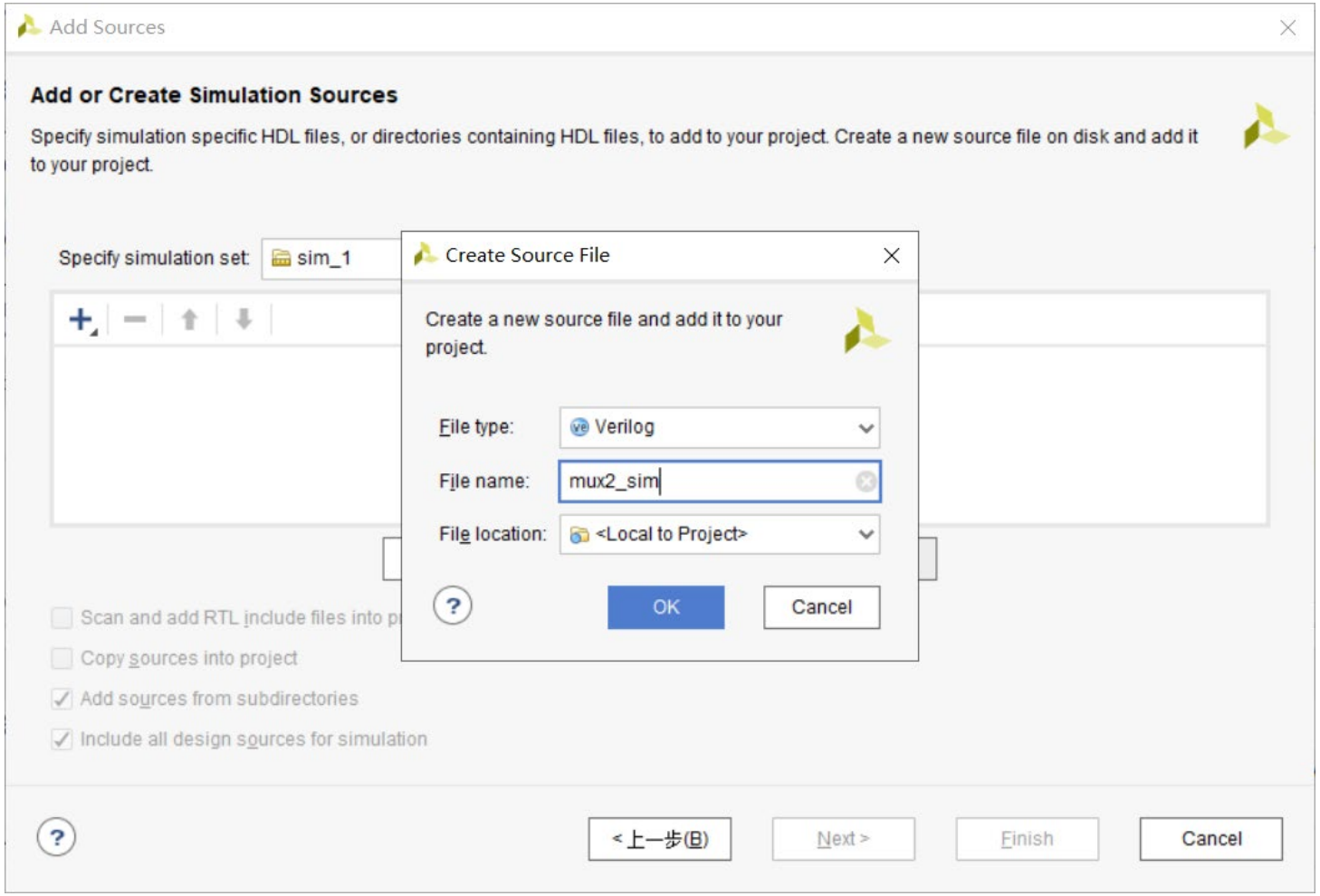
2、选择 add or create simulation sources



## 1. 2-1选择器设计　　1.2仿真

### 创建仿真源文件

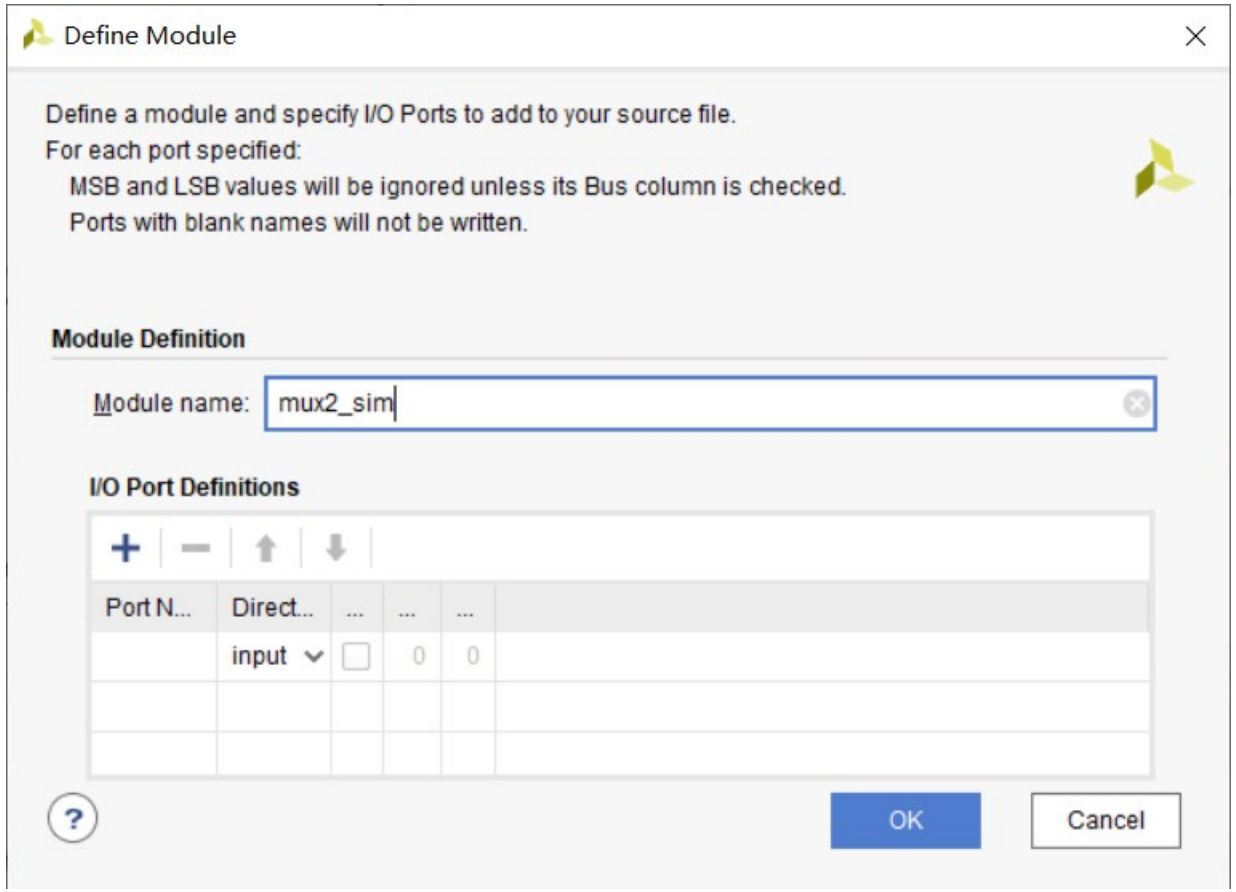
3、点击createfile（如果已经有写好的代码，可点击AddFiles添加）
此处及以下创建文件步骤和添加design sources相同



**1. 2-1选择器设计**　　1.2仿真

**创建仿真源文件**

3、定义模块。这里仿真文件是没有输入和输出的，所以直接跳过。



## 1. 2-1选择器设计　　1.2仿真

**编写源代码**

1、编写仿真源代码（以下代码表示ina为周期200ns的方波，inb为周期800ns的方波，选择端sel每5us切换一次）

参数定义→

调用模块→

初始化→

```
module mux_sim();
reg ina,inb;
reg sel;
wire outy;

mux2 mux2sim_1(ina,inb,sel,outy);

initial
begin
ina = 0;
inb = 0;
sel = 0;
end
```

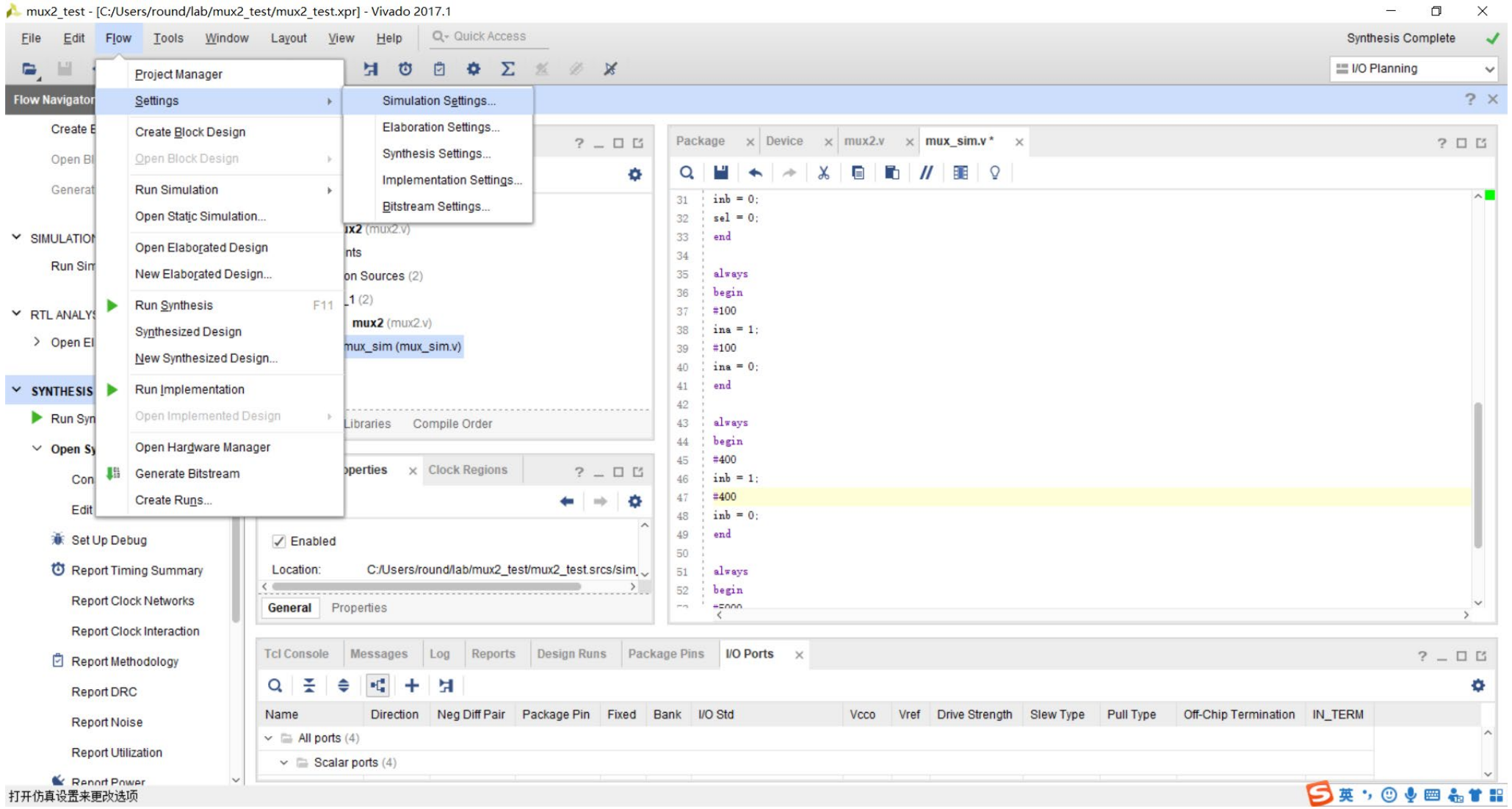
设置input参数行为↓

```
always
begin
#100
ina = 1;
#100
ina = 0;
end

always
begin
#400
inb = 1;
#400
inb = 0;
end

always
begin
#5000
sel = 1;
#5000
sel = 0;
end
endmodule
```

# 1. 2-1选择器设计　　1.2仿真

## 仿真

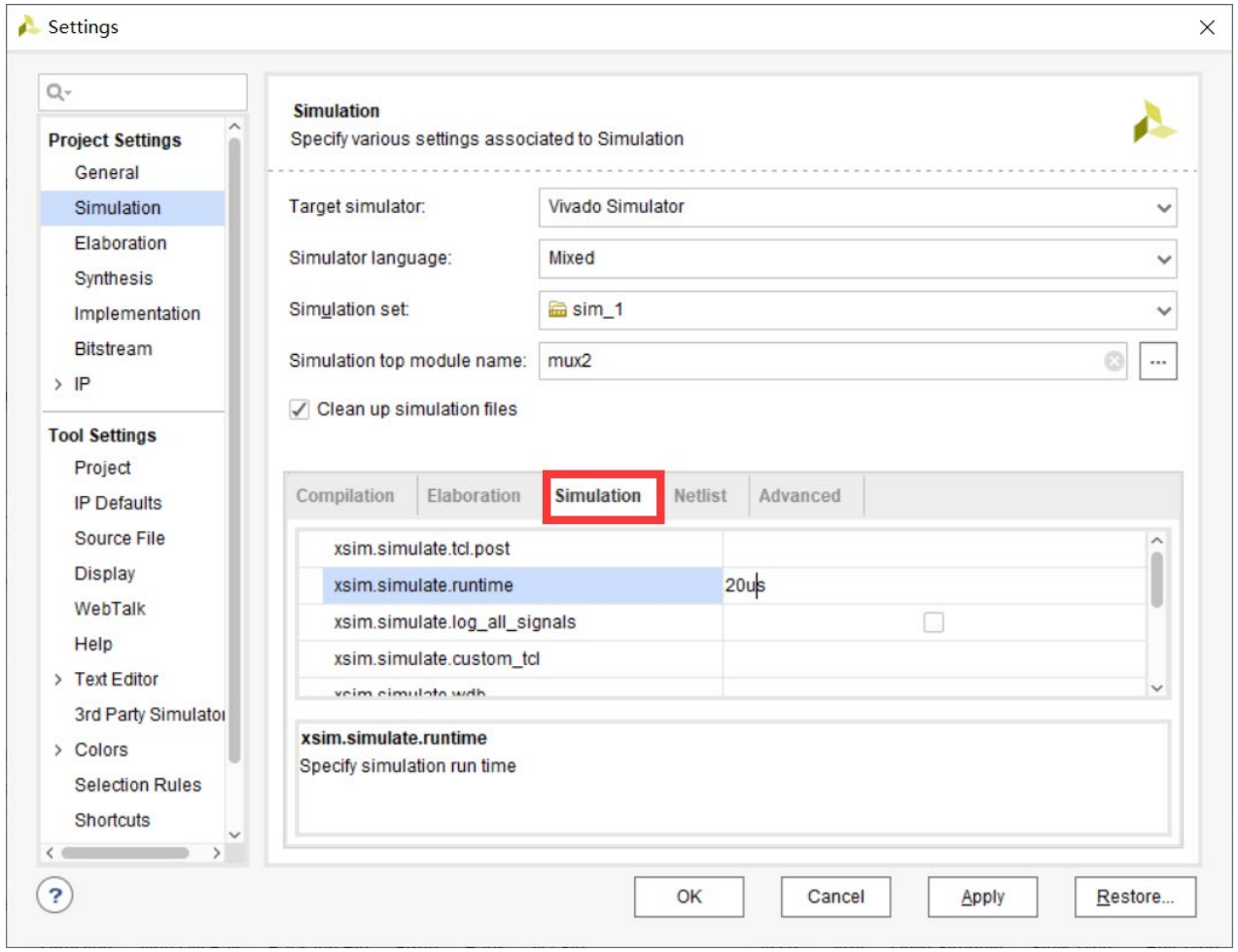
1、仿真时长设置 点击flow→setting→simulation settings



**1. 2-1选择器设计**　　1.2仿真

**仿真**

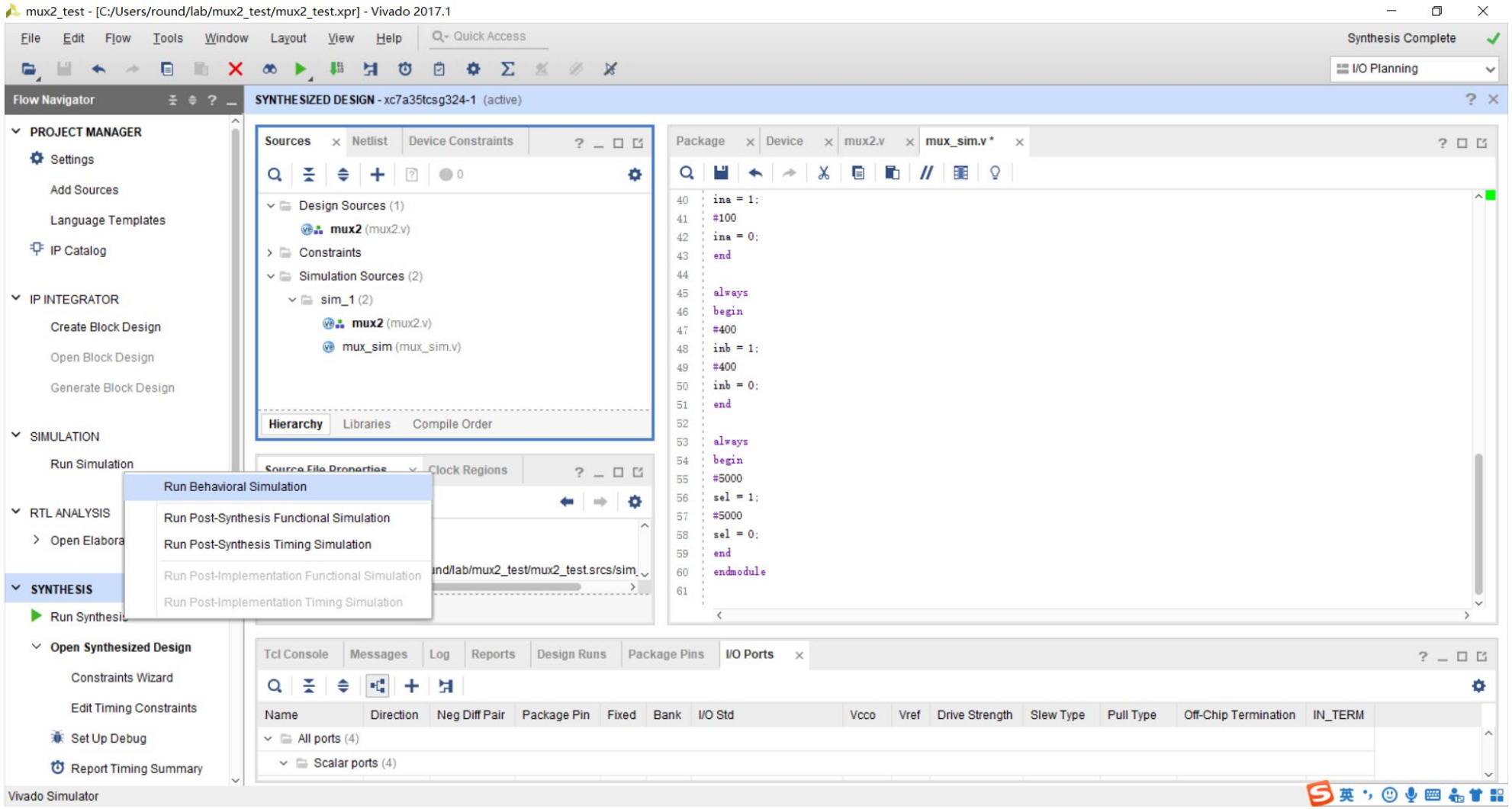
2、点击simulation，改变xsim.simulate.runtime为合适时长。这里我设置为四次切换sel的时长20us



# 1. 2-1选择器设计　　1.2仿真

## 仿真

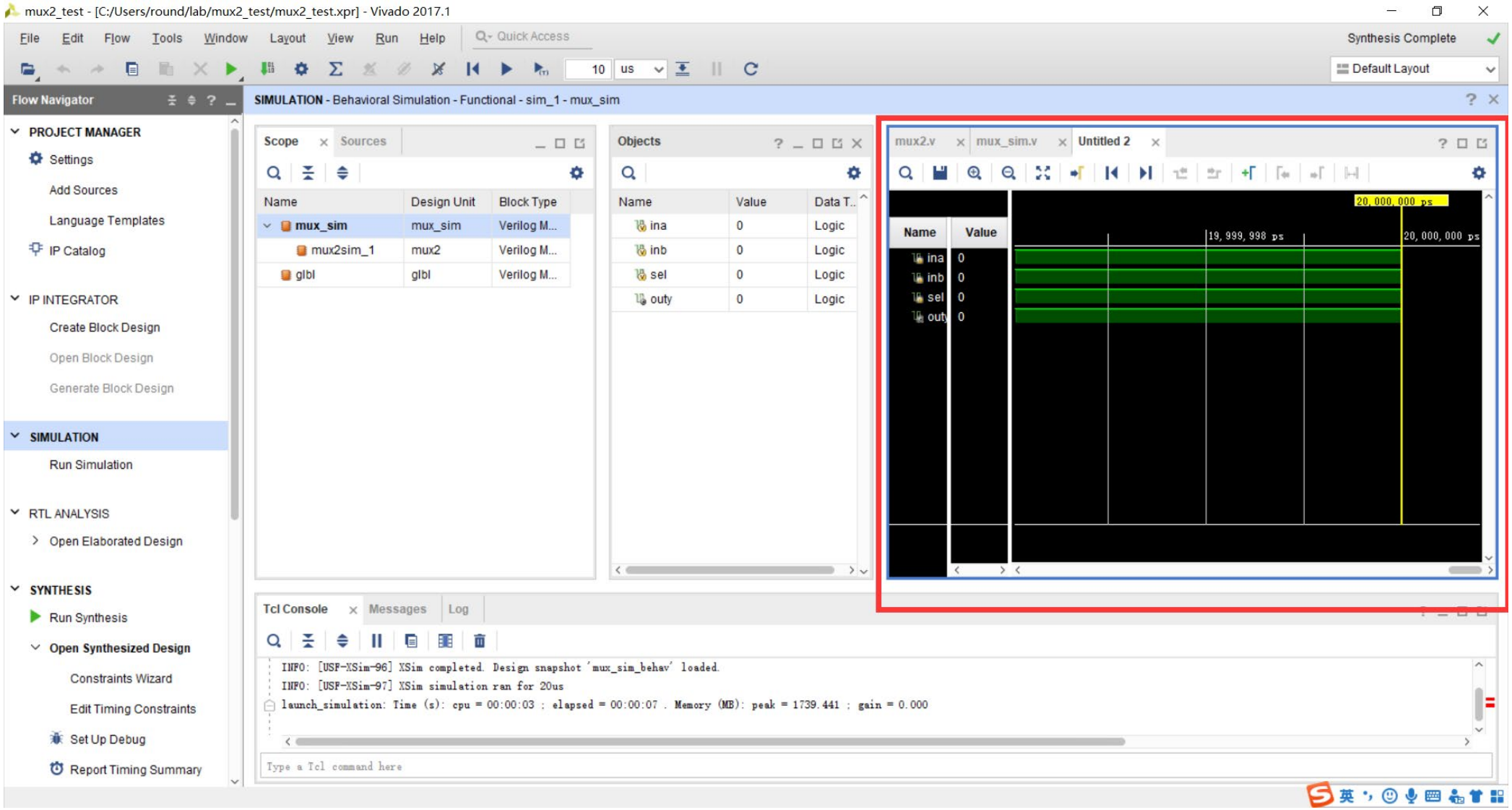
3、点击run simulation→ run behavioral simulation进行行为级仿真



# 1. 2-1选择器设计　　1.2仿真

## 仿真

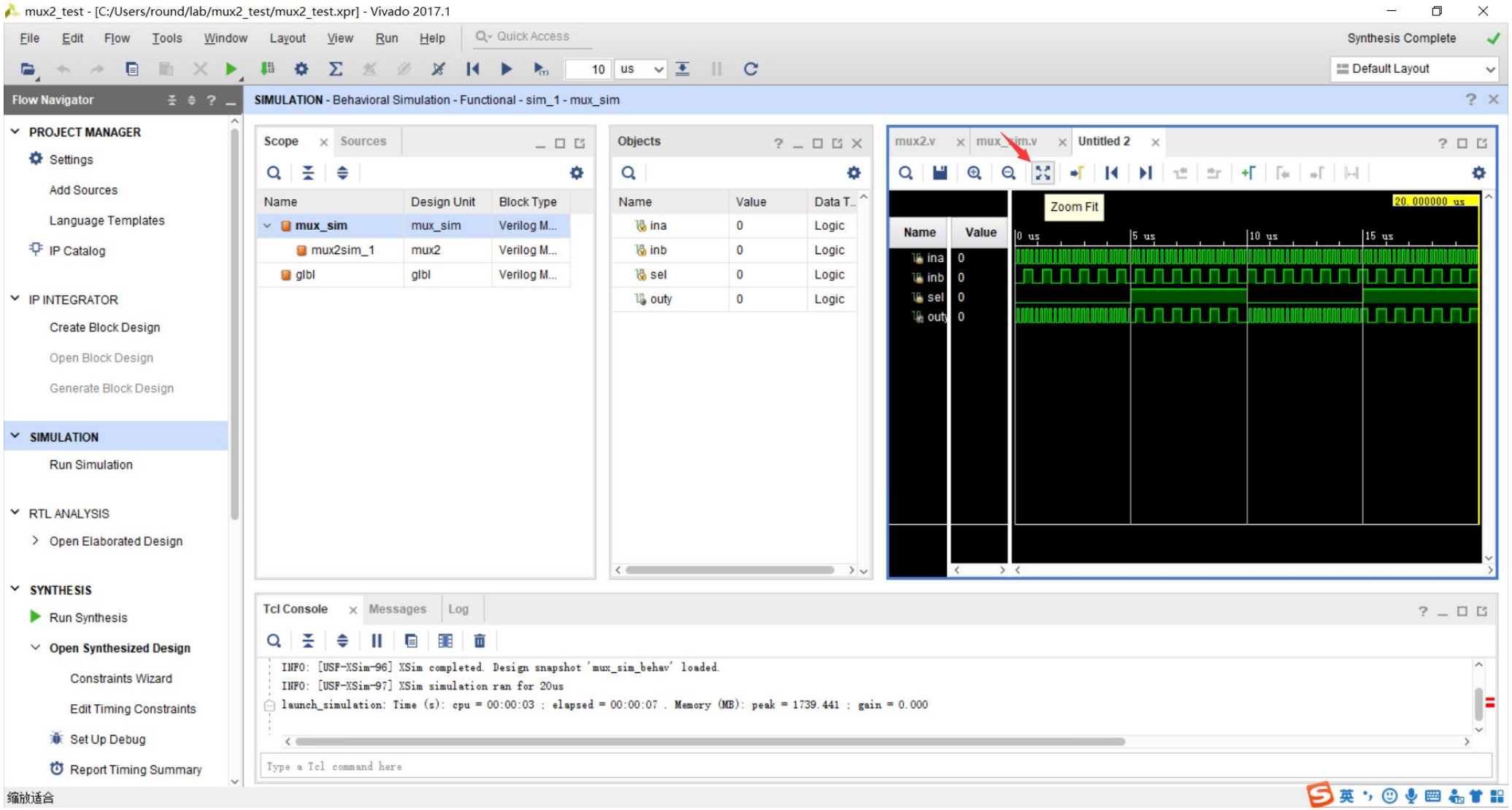
4、查看仿真结果



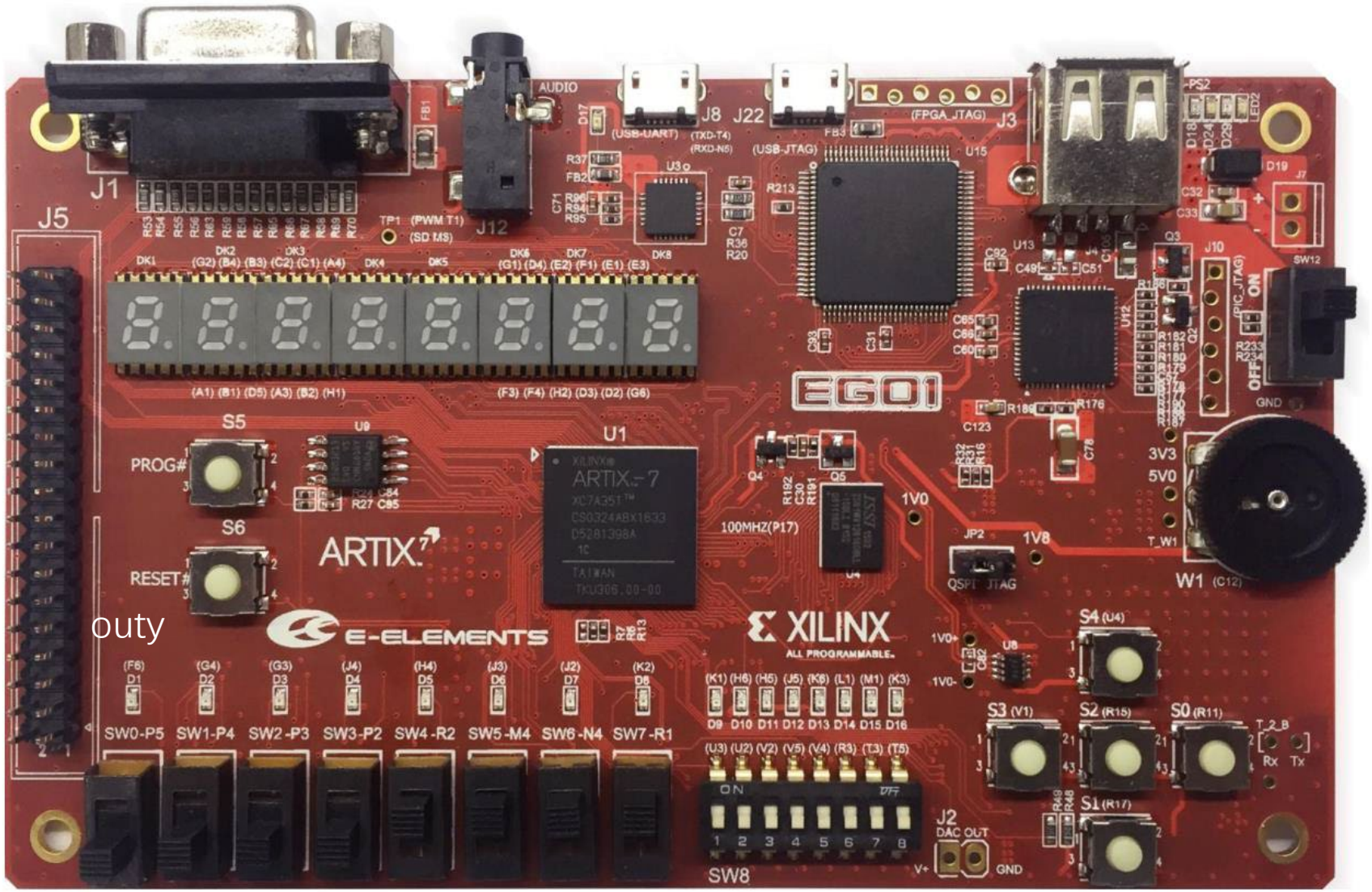
# 1. 2-1选择器设计　　1.2仿真

## 仿真

5、查看仿真结果（可以点击zoom fit来查看整个仿真时间段的结果）



# 1. 2-1选择器设计　　引脚约束




outy

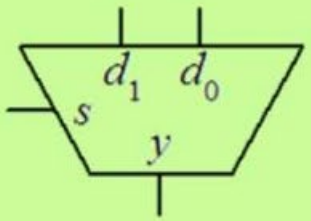
lna　inb　　sel

Design it with AND/OR/NOT gate level Verilog description.
Design it with Boolean function Verilog description.
Design it with Always statement and if-else statement.
Design it with always statement and case statement.

## 1. 2-1选择器设计　　1.3其他描述方式

Design it with AND/OR/NOT gate level Verilog description.

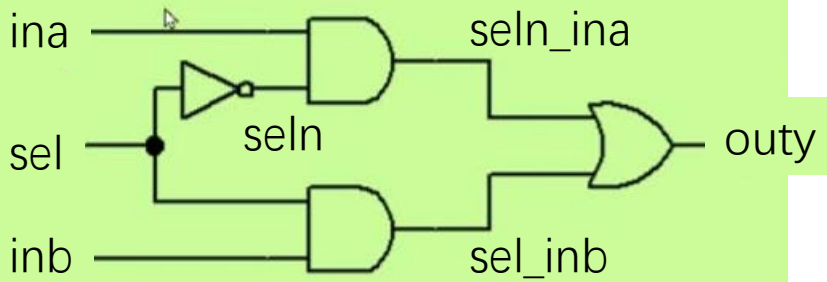
**Symbol**




**Truth table**

| s | $d_1$ | $d_0$ | y |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

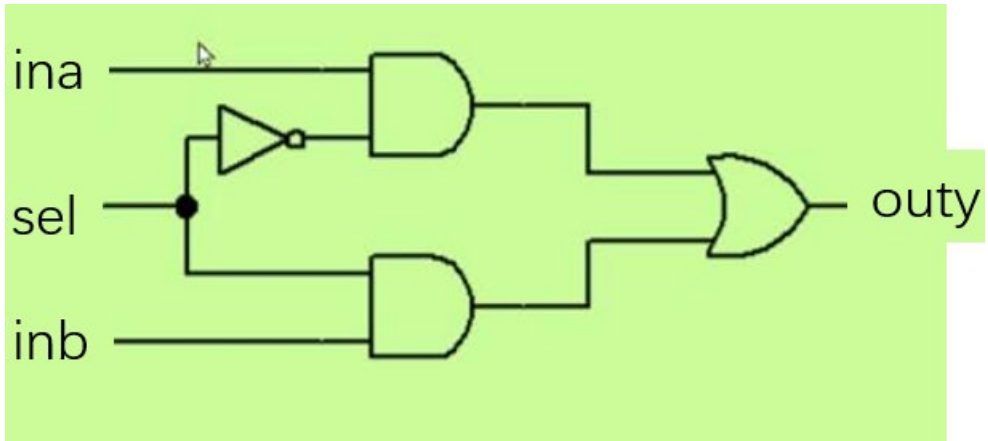
**Circuit**




```
module mux_gate(
    input ina,
    input inb,
    input sel,
    output outy
    );
    wire seln, seln_ina, sel_inb;
    not U1(seln, sel);
    and U2(seln_ina, ina, seln);
    and U3(sel_inb, inb, sel);
    or  U4(outy, seln_ina, sel_inb);

endmodule
```

## 1. 2-1选择器设计　　1.3其他描述方式

Design it with Boolean function Verilog description.

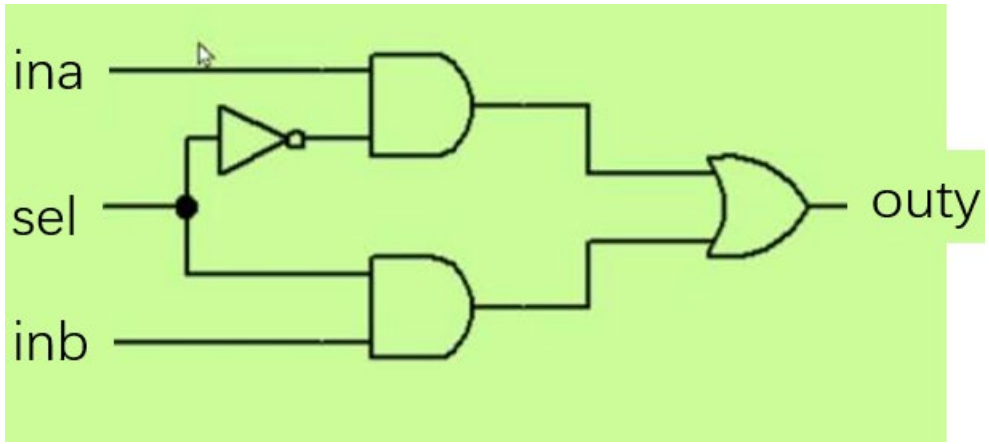



outy=ina·sel'+inb·sel

```
module mux2_boolean(
    input ina,
    input inb,
    input sel,
    output outy
    );
    assign outy = (ina & ~sel)|(inb & sel);

endmodule
```

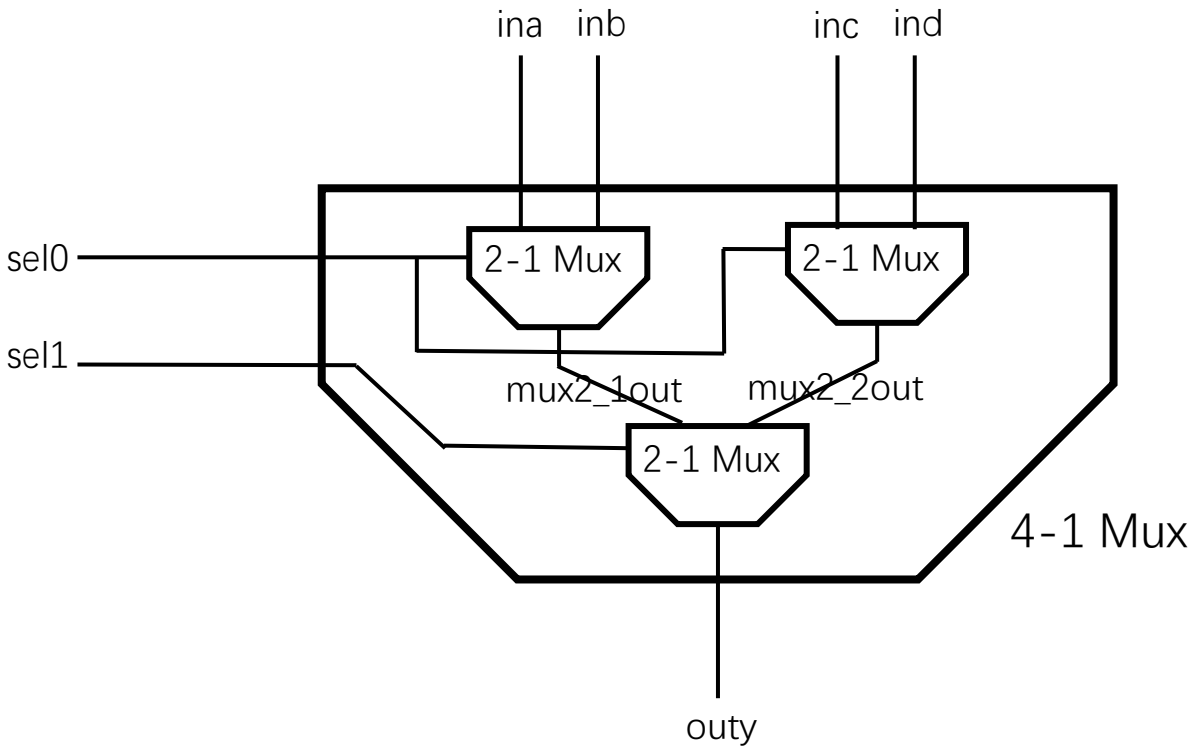
## 1. 2-1选择器设计　　1.3其他描述方式

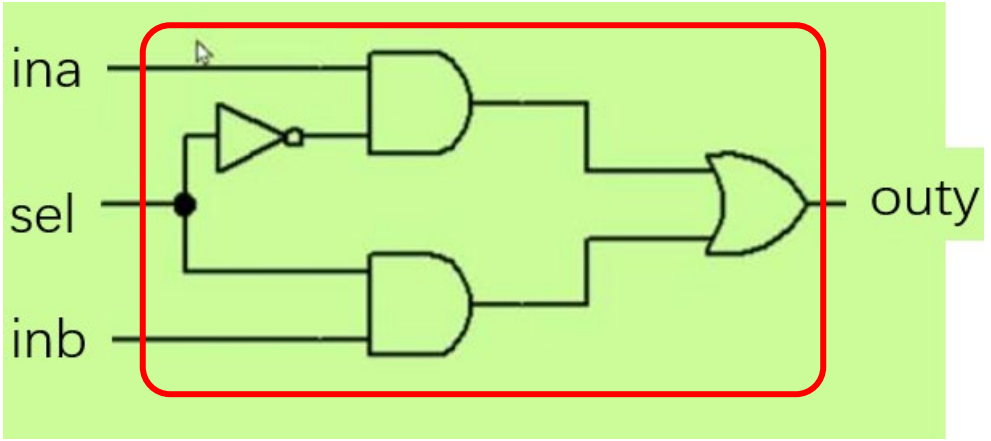
Design it with always statement and case statement.




```
module mux_case(
    input ina,
    input inb,
    input sel,
    output reg outy
    );
    always@(*)
    begin
    case(sel)
    1'b0: outy = ina;
    1'b1: outy = inb;
    default: outy = 1'bx;
    endcase
    end
endmodule
```

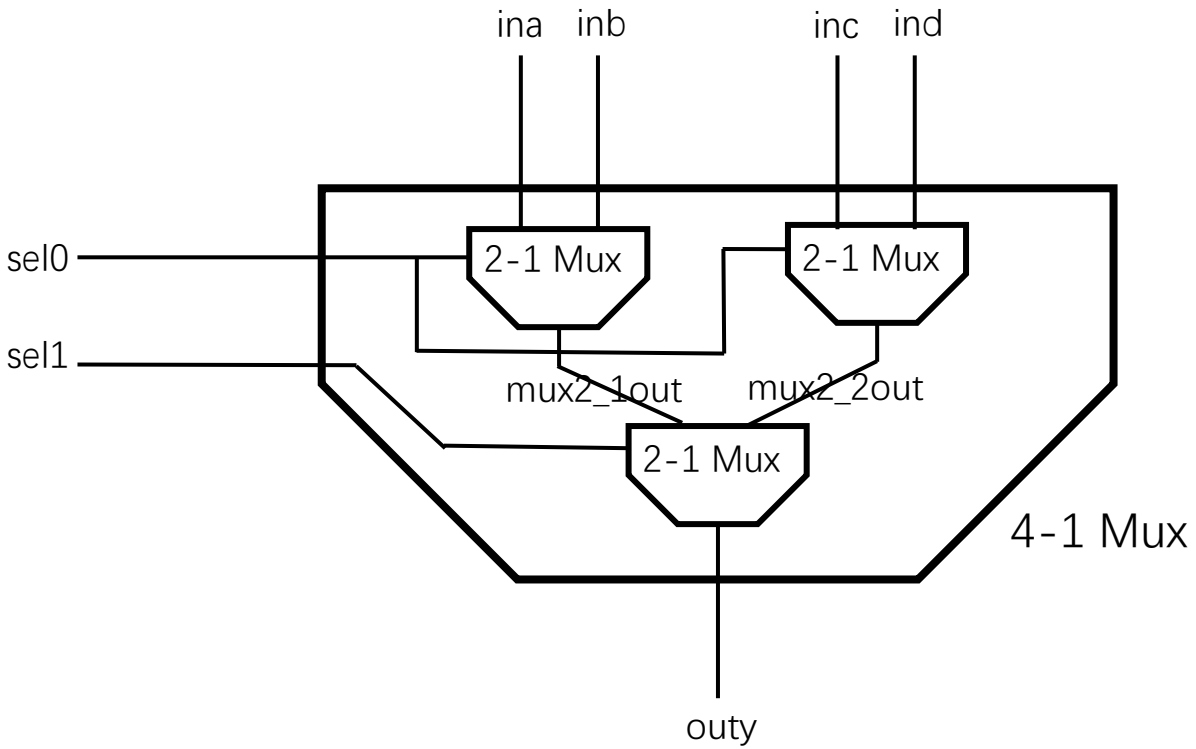
## 2. 4-1选择器设计（使用hierarchy和2-1mux模块）




| sel1 | sel0 | outy |
|---|---|---|
| 0 | 0 | ina |
| 0 | 1 | lnb |
| 1 | 0 | inc |
| 1 | 1 | ind |

## 2. 4-1选择器设计（使用hierarchy和2-1mux模块）




```
module mux4(
    input ina,
    input inb,
    input inc,
    input ind,
    input sel0,
    input sel1,
    output outy
    );
    wire mux2_1out,mux2_2out;
    mux2 U1(ina,inb,sel0,mux2_1out);
    mux2 U2(inc,ind,sel0,mux2_2out);
    mux2 U3(mux2_1out,mux2_2out,sel1,outy);

endmodule
```

## 2. 4-1选择器设计（使用hierarchy和2-1mux模块）

仿真
（下图仿真代码为ina、inb、inc分别为周期200ns，800ns，2000ns的方波，ind为恒为0的常数，sel端以周期为20us依次切换选择四个输入端的行为级仿真代码）

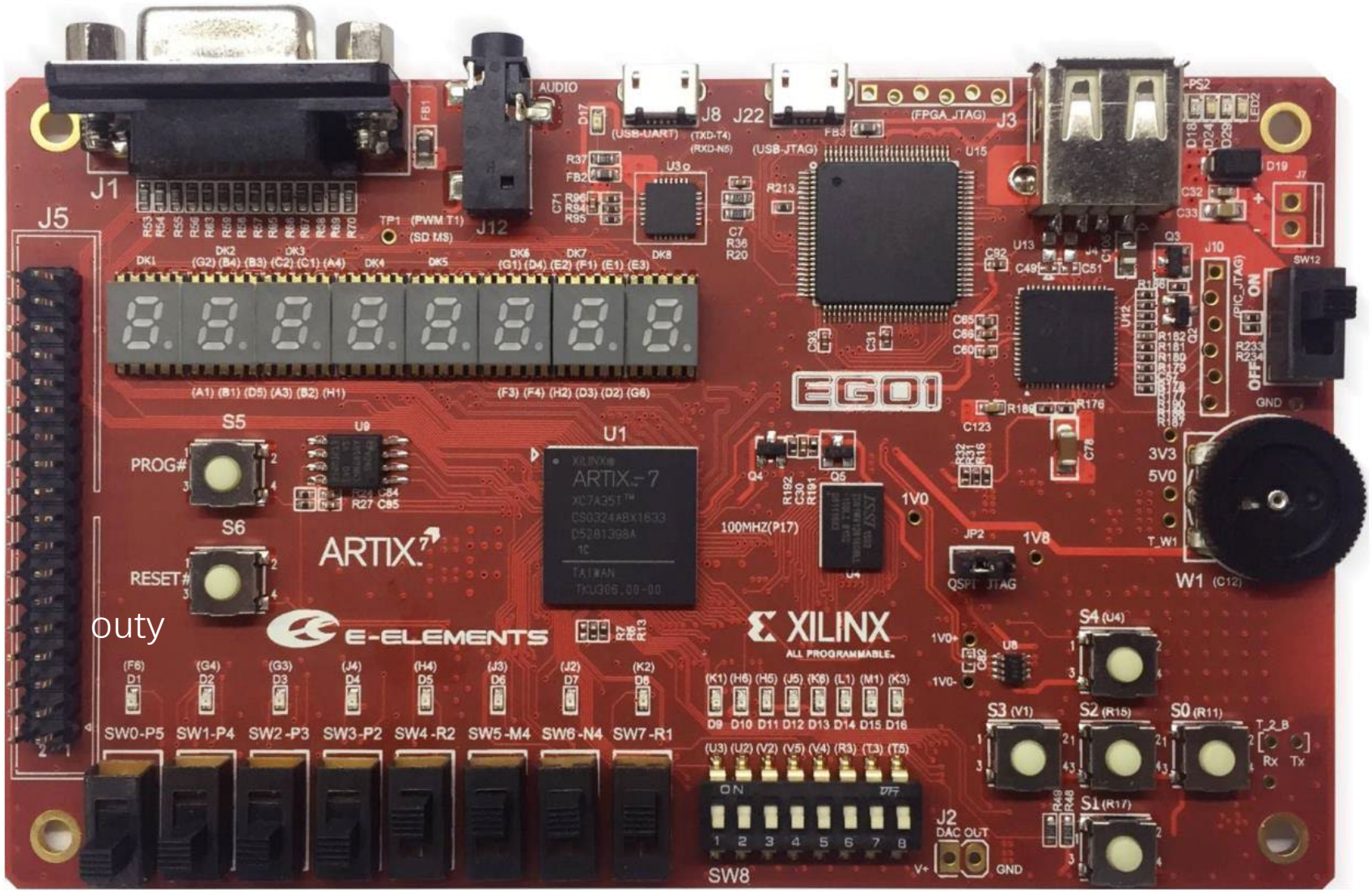
```
module mux4_sim();
reg  ina,inb,inc,ind,sel0,sel1;
wire outy;
mux4 mux4_simulate(ina,inb,inc,ind,sel0,sel1,outy);
initial
begin
ina = 0;
inb = 0;
inc = 0;
ind = 0;
sel0 = 0;
sel1 = 0;
end
```

```
always
begin
#100
ina=~ina;
end

always
begin
#400
inb=~inb;
end

always
begin
#1000
inc=~inc;
end
```

```
always
begin
#5000
sel0 = 1;
sel1 = 0;
#5000
sel0 = 0;
sel1 = 1;
#5000
sel0 = 1;
sel1 = 1;
#5000
sel0 = 0;
sel1 = 0;
end
endmodule
```

## 2. 4-1选择器设计（使用hierarchy和2-1mux模块）

引脚约束




lna inb inc ind sel1 sel0