【实验目的】

熟悉 Vivado 软件的下载、安装及使用

学习使用 Verilog 编写仿真文件

学习使用 Verilog 进行仿真，查看并分析波形文件

【实验环境】

PC 一台

Windows 或 Linux 操作系统

Vivado 工具

vlab.ustc.edu.cn(包含 Vivado 下载安装及使用教程)

【实验过程】

Step1.

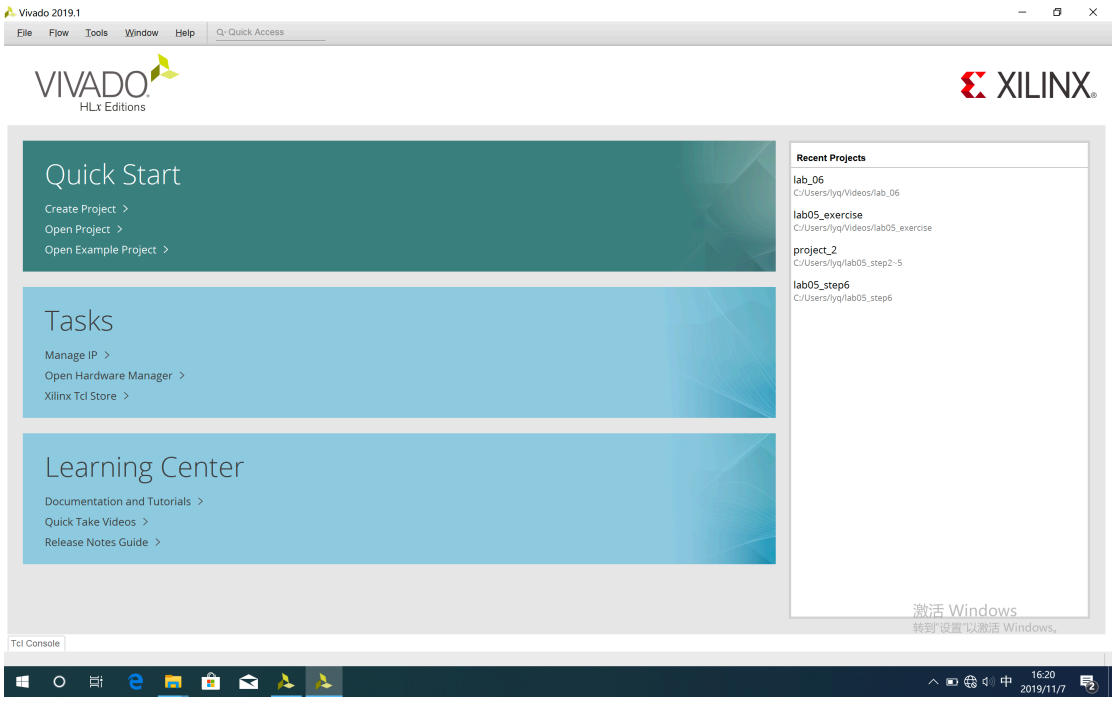
——下载并安装 Vivado 环境

根据讲义中的教程及助教的指导，顺利完成了安装

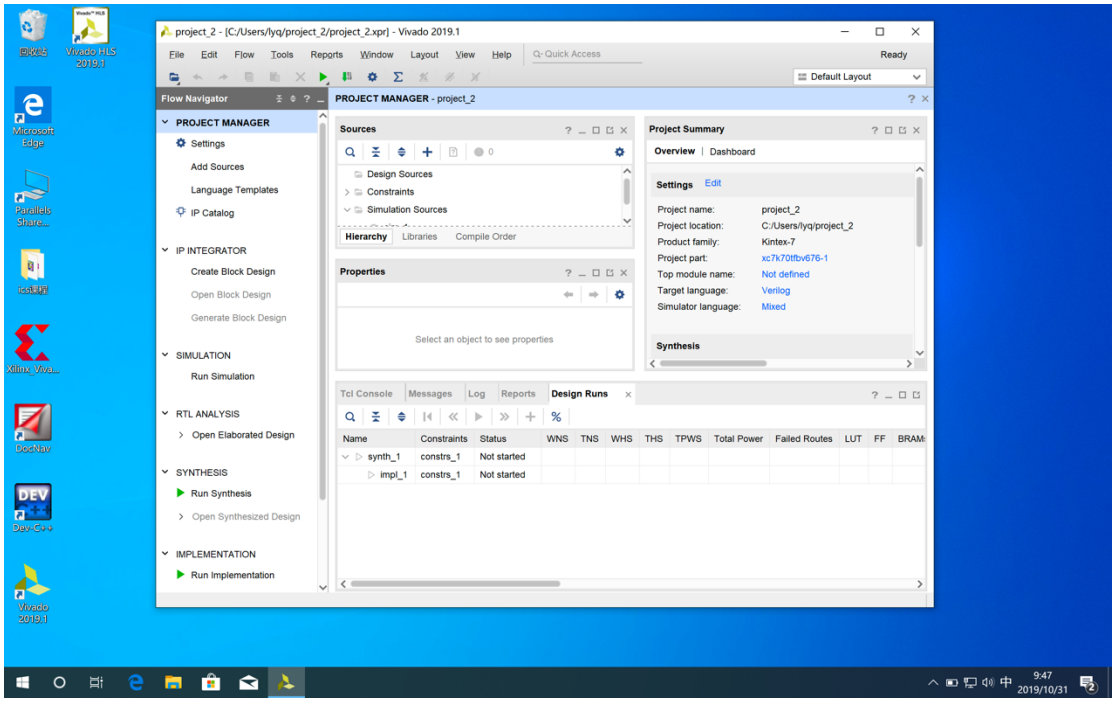
Step2&3.

——创建新工程并添加 Verilog 设计文件

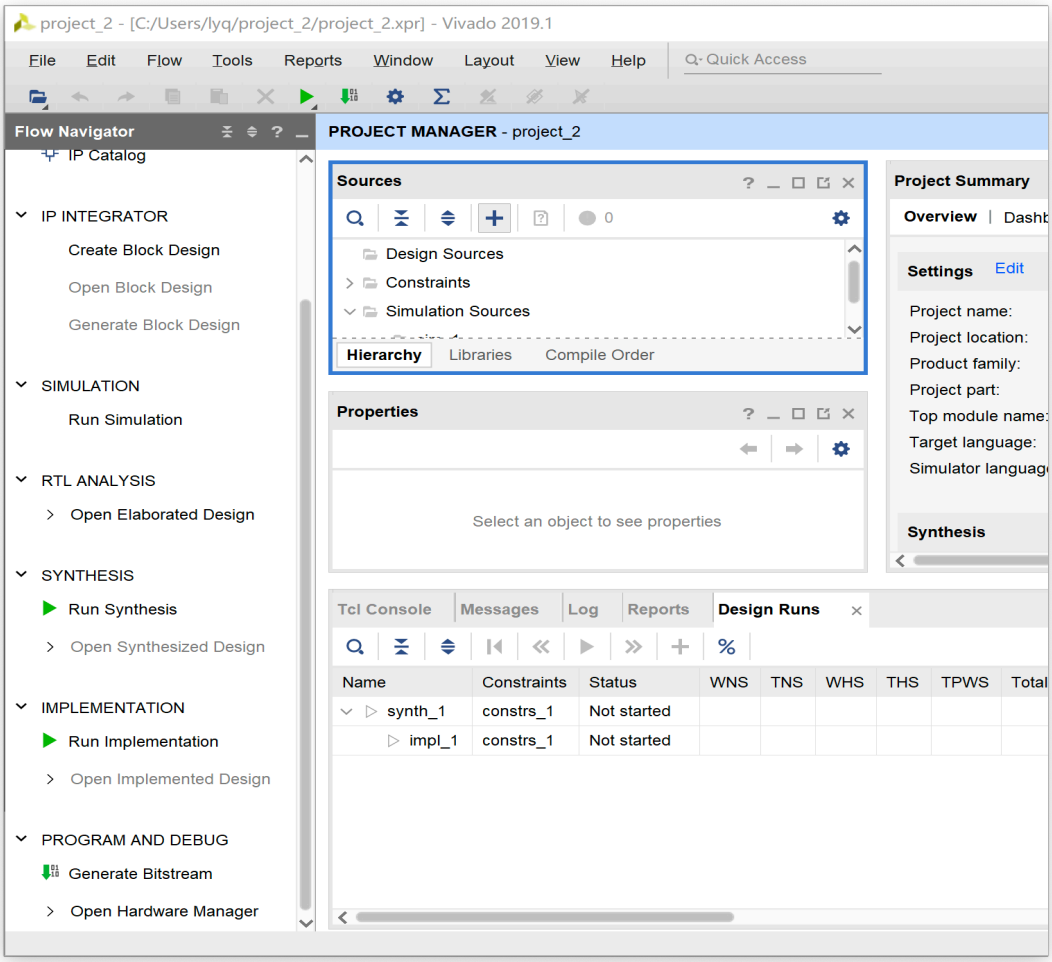
进入程序后界面如下：



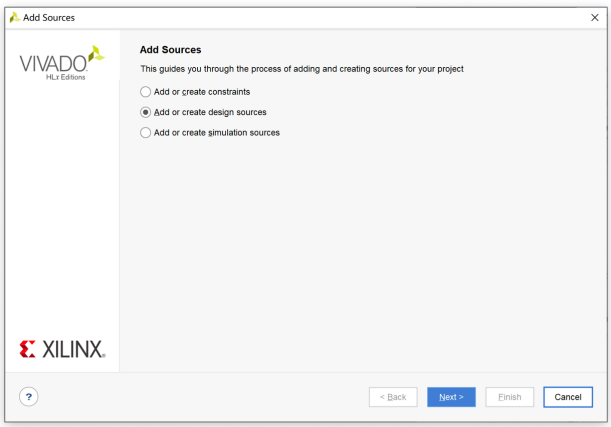
打开工程文件后，后界面如下，包含“Project Manager”、“Sources”、“Project Summary”、“Design Runs”四大区域：



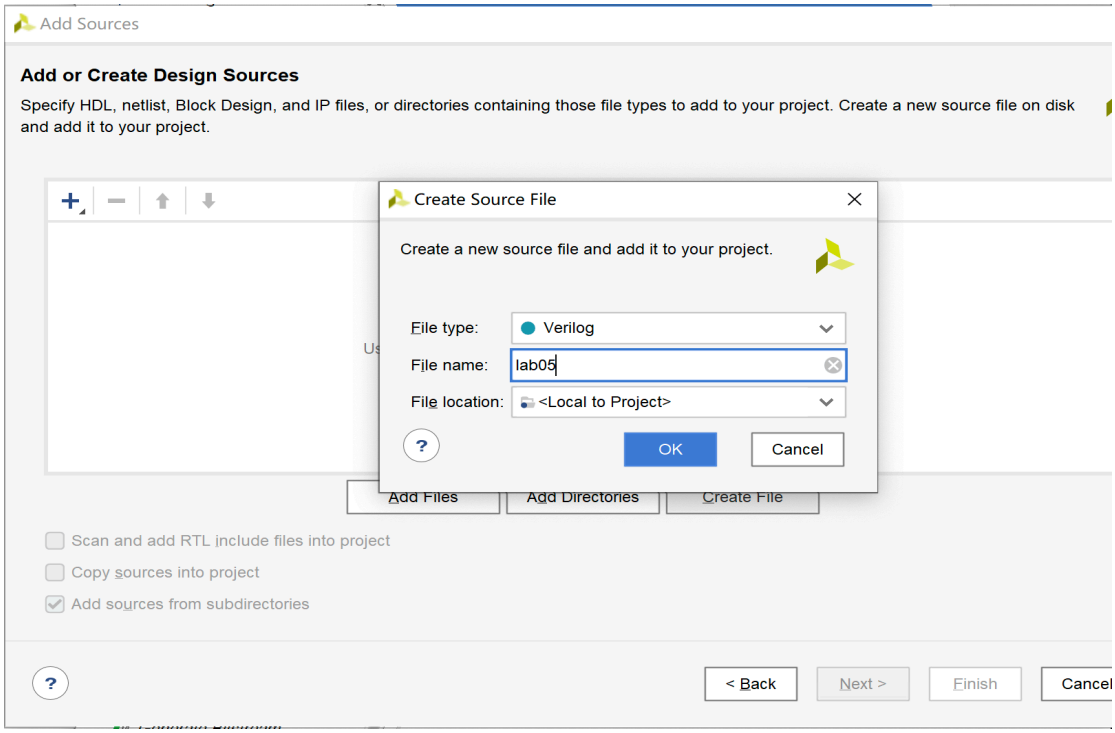
之后“Add Source”添加文件：



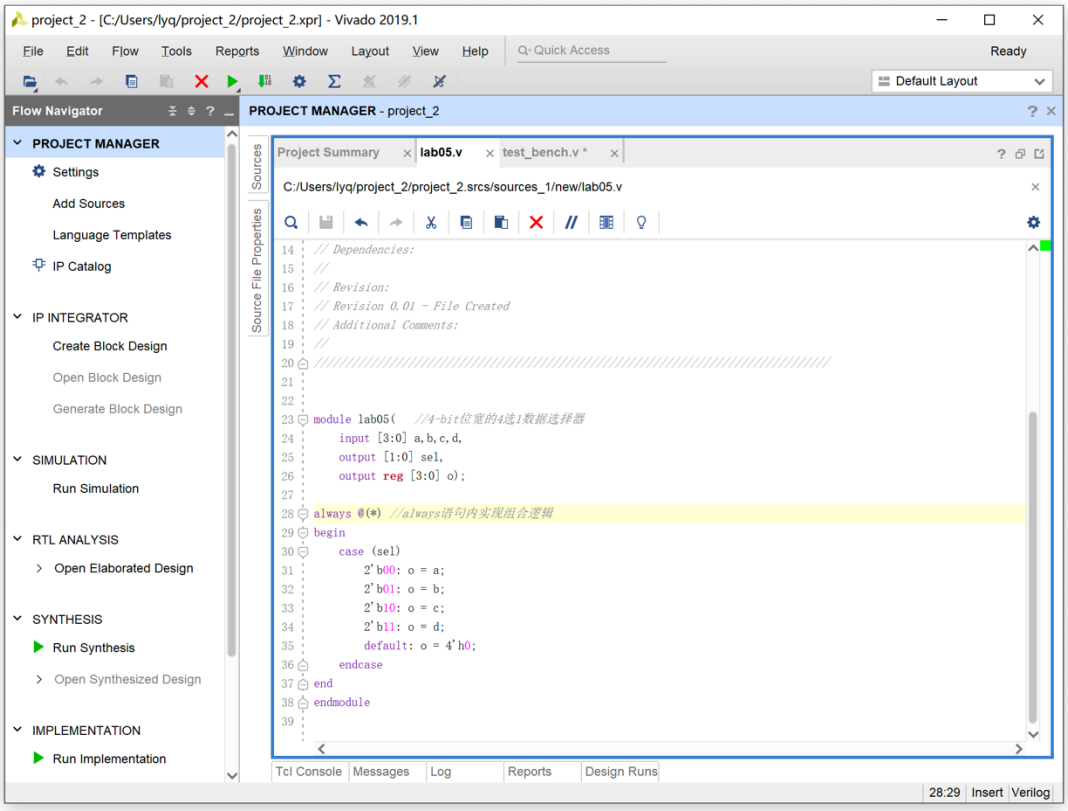
需要添加 Verilog 设计文件，因此选择“Add or create design source”



点击“Create File”，输入文件名 ：

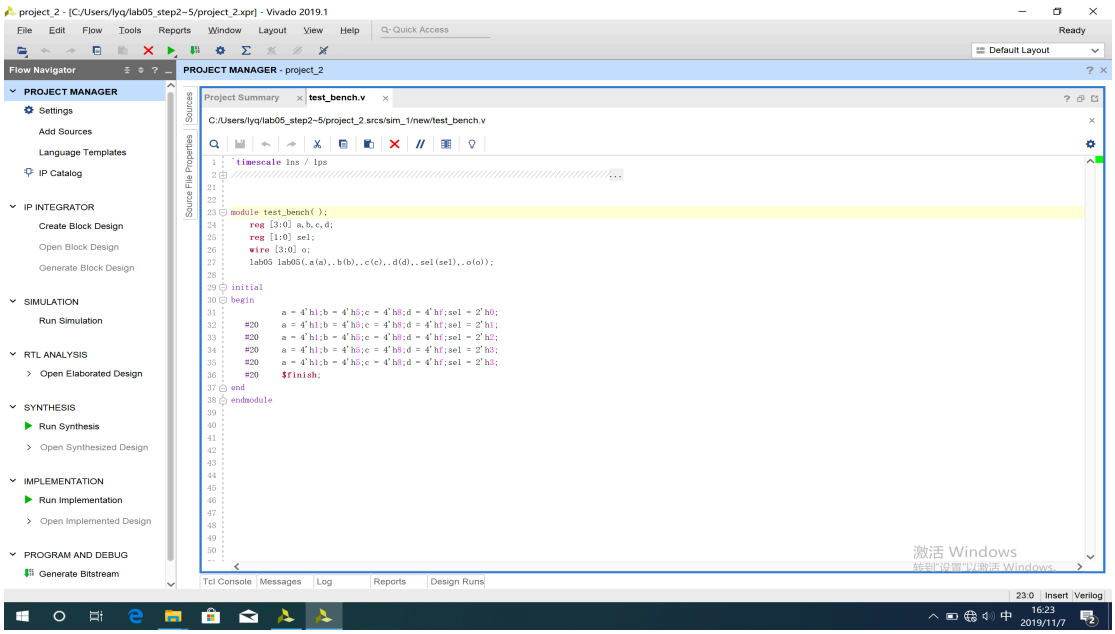


在生成的 verilog 文件中输入如下代码：



Step4.

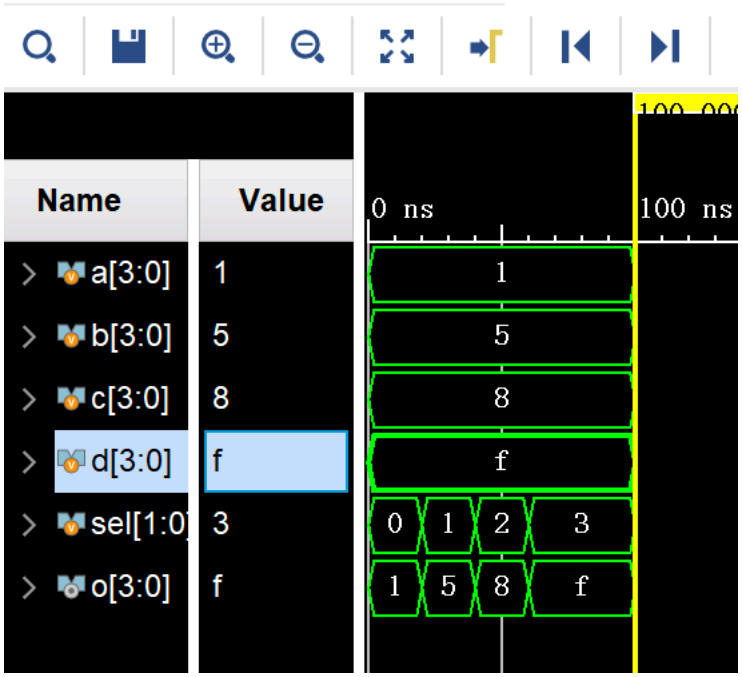
现在，我们要在 Vivado 中添加 Verilog 仿真测试文件，如下图所示。与 Step3 不同的是，在创建文件时，需要选择“Add or create simulation sources”选项。 仿真文件内容如下：



由上述仿真代码可以看出，Verilog 仿真文件与 Verilog 设计文 件有些不同。第一，仿真文件不需要输入输出信号，所有的信号都是 模块的内部信号。第二，在仿真文件内对被测试模块进行实例化，并 对被测试模块构造输入信号。第三，仿真文件只用于仿真，最终不会 被综合成电路，会经常用到“initial”等 Verilog 设计文件中不会 用到的关键字或语法，这些语法很多是不可综合的 。

Step5.

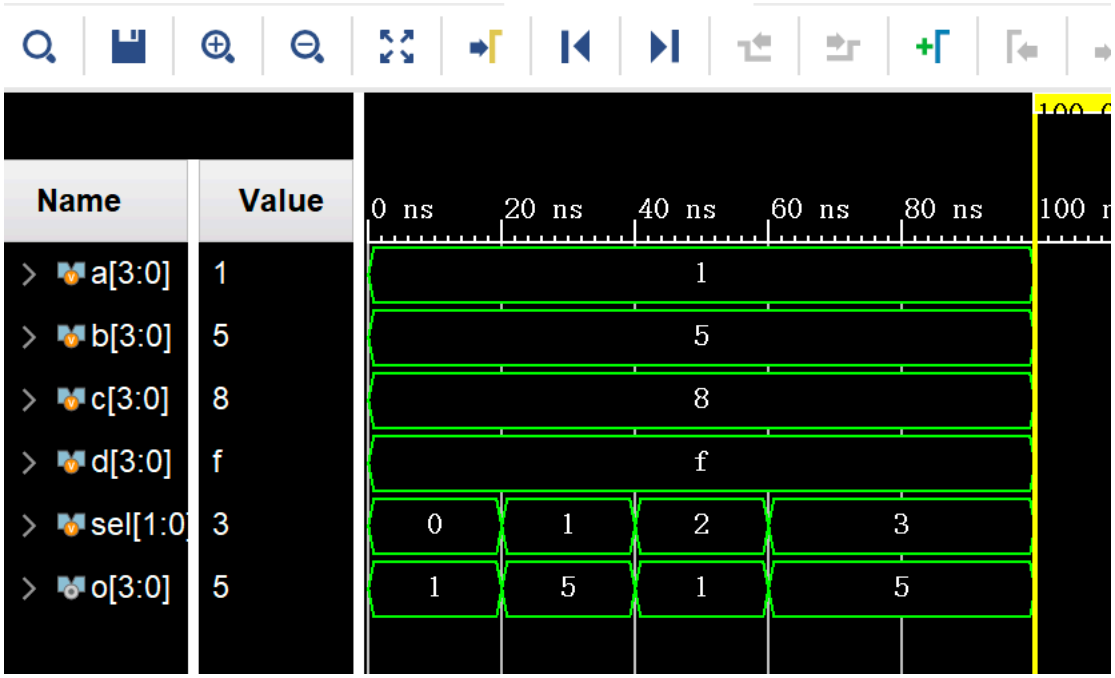
点击“Run Simulation”运行仿真工具，会出现如下图所示的界面：



通过观察波形我们可以发现，该电路的仿真波形符合四选一选择的行为特性，Verilog 代码设计正确。

之后，关闭波形仿真窗口，打开前面的 Verilog 设计文件，

将其中的“input [1:0] sel,”改成“input sel,”重新进行仿真， 观察波形结果：



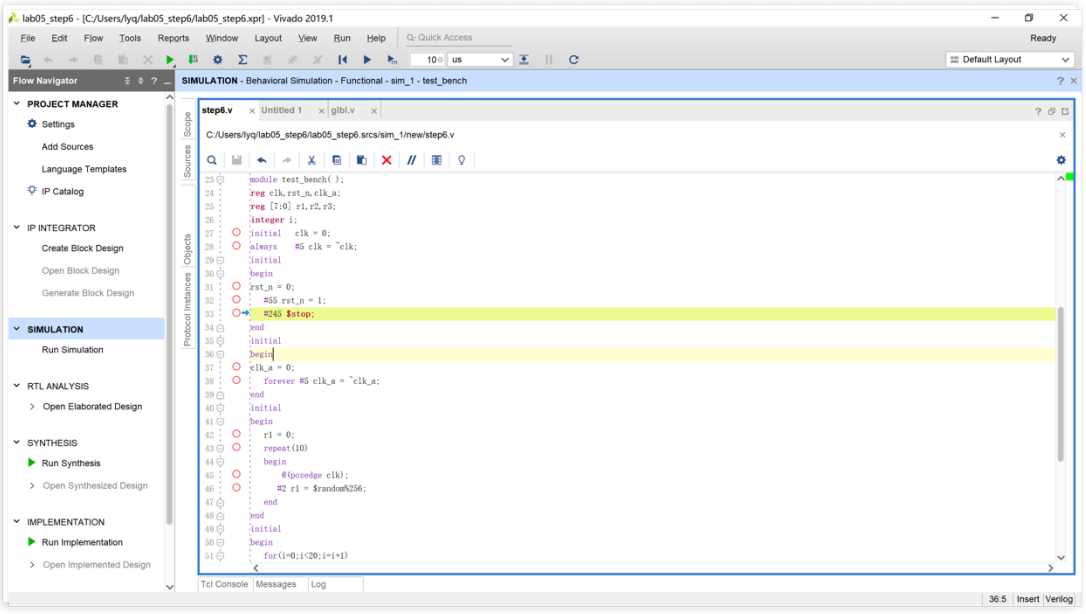
此图中，sel 为 2 或 3 时输出错误，不符合 4 to 1 MUX 的逻辑功能。实际上，信号位宽不匹配是 Verilog 代码编写过程中经常出现的一个错误， 这类错误不会导致语法错误，阅读代码是也不容易发现，但通过仿真工具可以比较容易的定位到。

Step6.

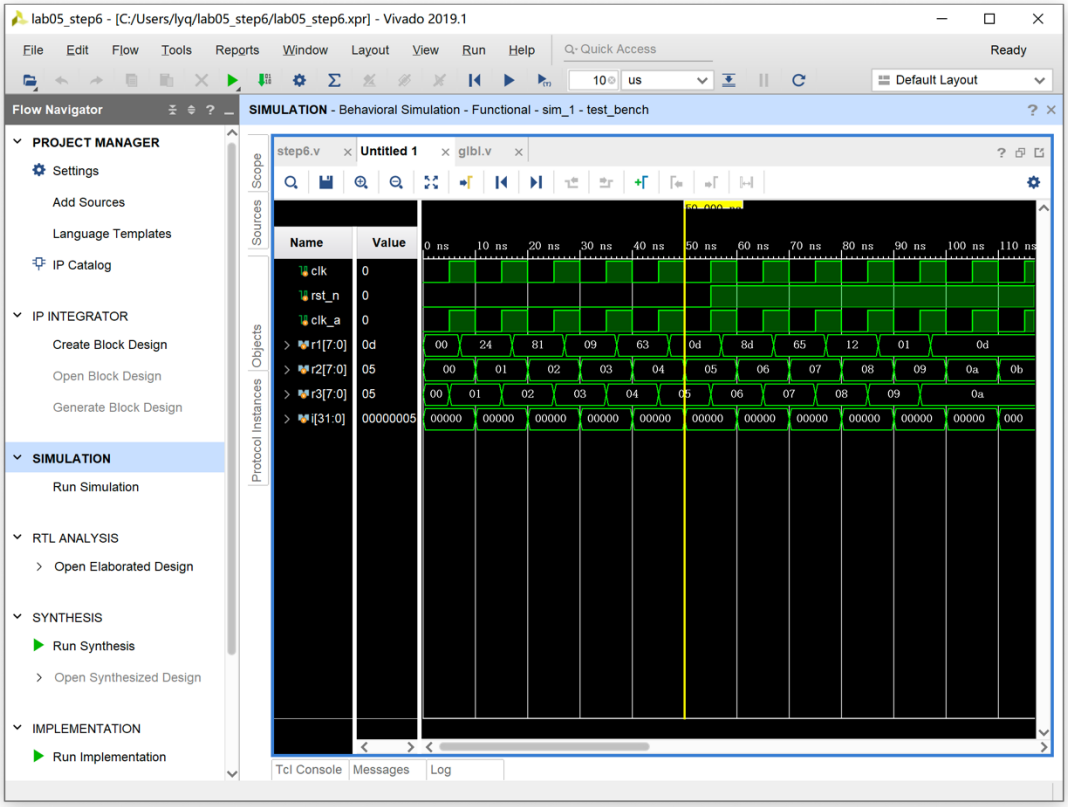
——波形仿真

在 Vivado 中新建一个工程，加入下面的仿真文件，进行仿真，观察各信号的波形

具体代码如下：



得到波形如下：



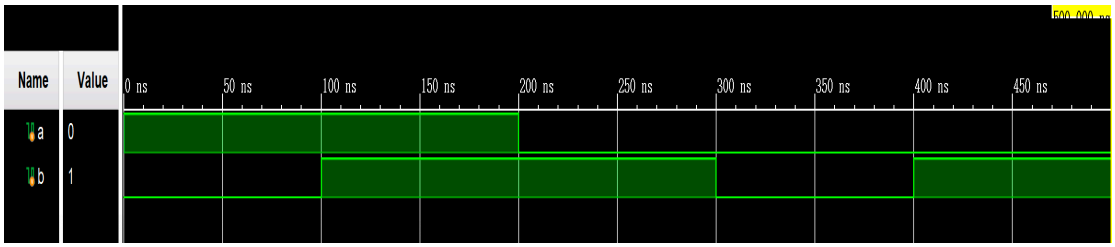
【实验练习】

题目 1：

对照讲义中“实验步骤”的内容即可写出代码，如下图所示：

```
1  `timescale 1ns / 1ps
2  //////////////////////////////////////////////////////////////////////////////////
21
22
23 module exercise_1( );
24 reg a,b;
25 initial
26 begin
27     a = 1'b1;b = 1'b0;
28 #100 b = 1'b1;
29 #100 a = 1'b0;
30 #100 b = 1'b0;
31 #100 b = 1'b1;
32 #100 $finish;
33 end
34 endmodule
35
```

进行仿真后，画出波形图如下：

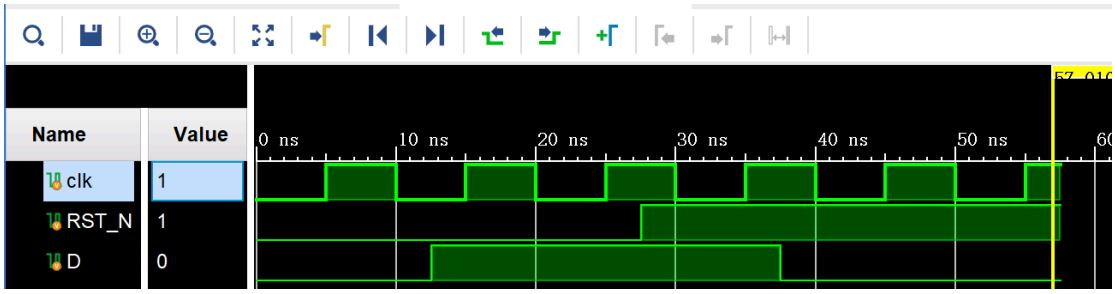


题目 2:

与题目 1 思路基本一致，代码如下：

```
1
2  module exercise_2( );
3  reg clk,RST_N,D;
4  wire Q;
5  initial   clk = 0;
6  always    #5 clk = ~clk;
7  initial
8  begin
9              RST_N = 1'b0;D = 1'b0;
10       #12.5    D = 1'b1;
11       #15    RST_N = 1'b1;
12       #10    D = 1'B0;
13       #20    $finish;
14 end
15 endmodule
16
```

仿真如下：



题目 3:

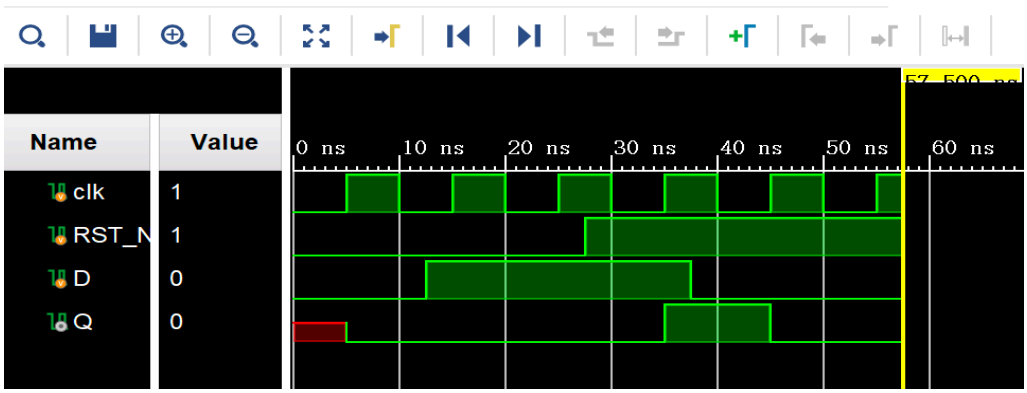
将题干中给出的代码写入设计文件“d_ff_r.v” 中：

```
module d_ff_r(
input clk,RST_N,D,
output reg Q);
always@(posedge clk)
begin
if(RST_N==0) Q <= 1'b0;
else
Q <= D;
end
endmodule
```

在第二题的代码中调用这个模块即可：

```
1
2   module exercise_2( );
3   reg clk,RST_N,D;
4   wire Q;
5   initial   clk = 0;
6   always    #5 clk = ~clk;
7   initial
8   begin
9               RST_N = 1'b0;D = 1'b0;
10      #12.5     D = 1'b1;
11      #15     RST_N = 1'b1;
12      #10     D = 1'B0;
13      #20     $finish;
14  end
15  d_ff_r d_ff_r(.clk(clk),.RST_N(RST_N),.D(D),.Q(Q));
16  endmodule
17
```

输出波形图：



题目 4:

（仿真文件）思路：设置了一个时钟信号，在上升沿到来时将 A 的值加 1，译码器中 A 值对应的线输出 1，其余输出 0；将所有情况遍历后，调用系统函数“$finish”结束仿真

代码如下：

```
1   `timescale 1ns / 1ps
2   //////////////////////////////////////////////////////////////////////////////////
21
22
23
24  module exercise_4();
25  reg clk;
26  reg [2:0]A;
27  wire [7:0]Y;
28  initial
29  begin
30      clk = 0;
31      #75 $finish;
32  end
33  always    #5 clk = ~clk;
34  initial   A = 3'b0;
35  always @(posedge clk)
36      A <= A + 3'b1;
37  decoder decoder(.A(A),.Y(Y));
38
39
40  endmodule
41
```

其中模块 decoder 定义如下：

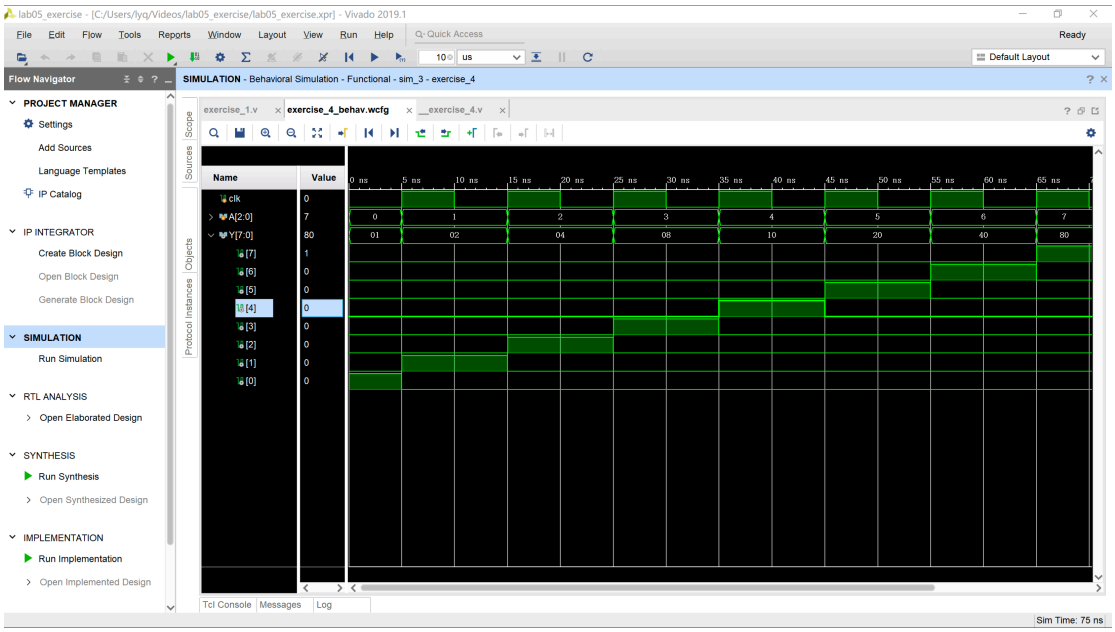
（设计文件）

使用 case 语句，通过 A 的取值对 8-bit 宽的 Y 信号赋值即可，具体

代码见下图：

```
23      module decoder(
24          input     [2:0] A,
25          output reg  [7:0] Y);
26
27      always @(*)
28      begin
29          case (A)
30              3'b000 : Y = 8'b00000001;
31              3'b001 : Y = 8'b00000010;
32              3'b010 : Y = 8'b00000100;
33              3'b011 : Y = 8'b00001000;
34              3'b100 : Y = 8'b00010000;
35              3'b101 : Y = 8'b00100000;
36              3'b110 : Y = 8'b01000000;
37              3'b111 : Y = 8'b10000000;
38              default: Y = 8'b0;
39          endcase
40      end
41      endmodule
```

画出波形图如下：



【总结与思考】

1.（1）本次实验是第一次接触 vivado，初步了解了软件的功能

（2）学习了许多之前很少接触的 verilog 关键字，如 initial，循环控制关键字，系统函数关键字等

（3）认识到设计文件与仿真文件的区别

（4）实验讲义中再次强调了“设置信号位宽”这一易错点，对我起到了警示的作用

2. 中等

3. 偏多

4. 题目 1 与题目 2 重复度太高，可以删去