

VIVADO仿真

此教程将仿真一个GPR（通用寄存器组）作为仿真教程。

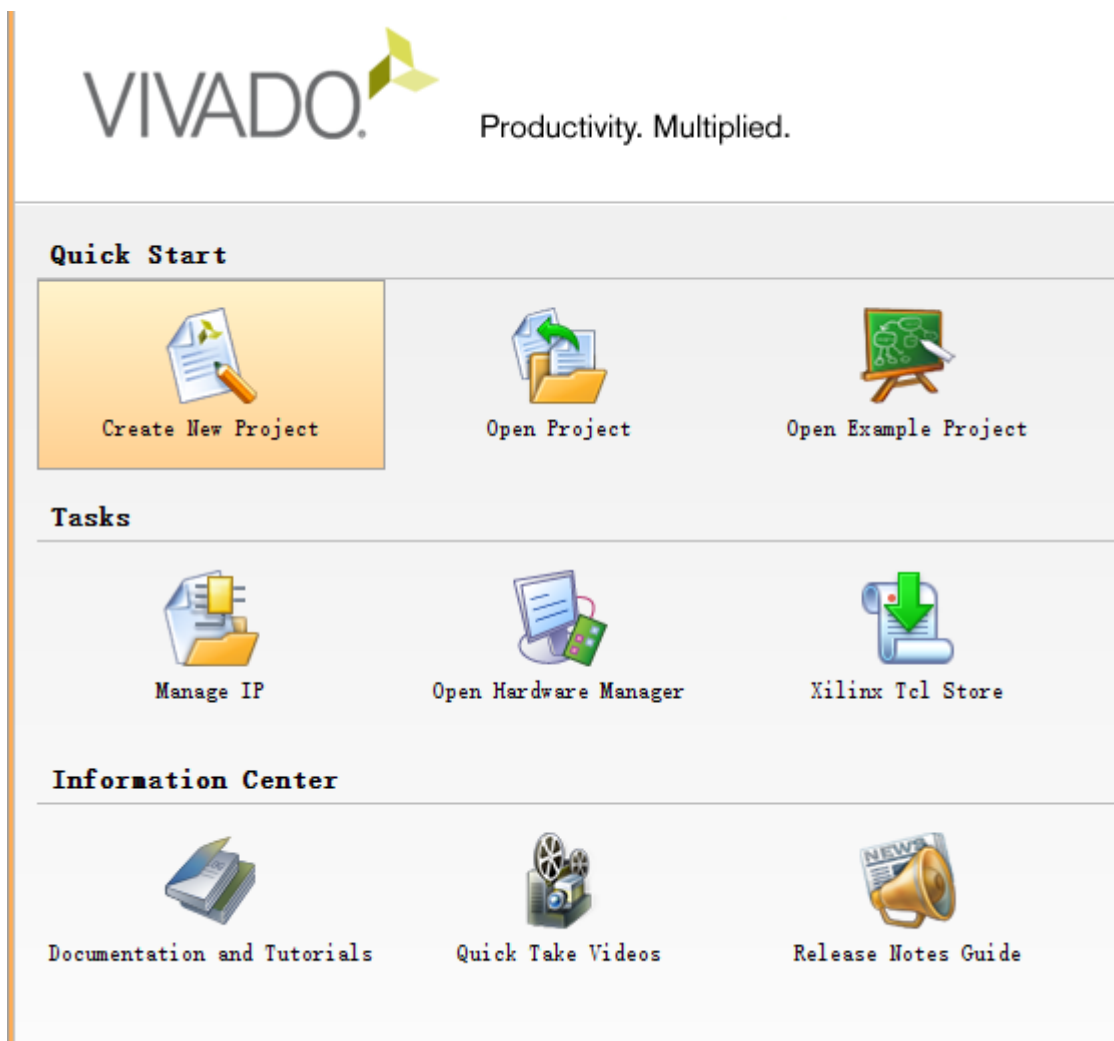
在Vivado中共有五种仿真类型：

1. run behavioral simulation 行为级仿真，也是通常说的功能仿真
2. post-synthesis function simulation综合后的功能仿真
3. post-synthesis timing simulation综合后带时序信息的仿真，和真实运行的时序就相差不远了
4. post-implementation function simulation布线后的功能仿真
5. post-implementation timing simulation（布局布线后的仿真） 执行后的时序仿真 最接近真实的时序波形

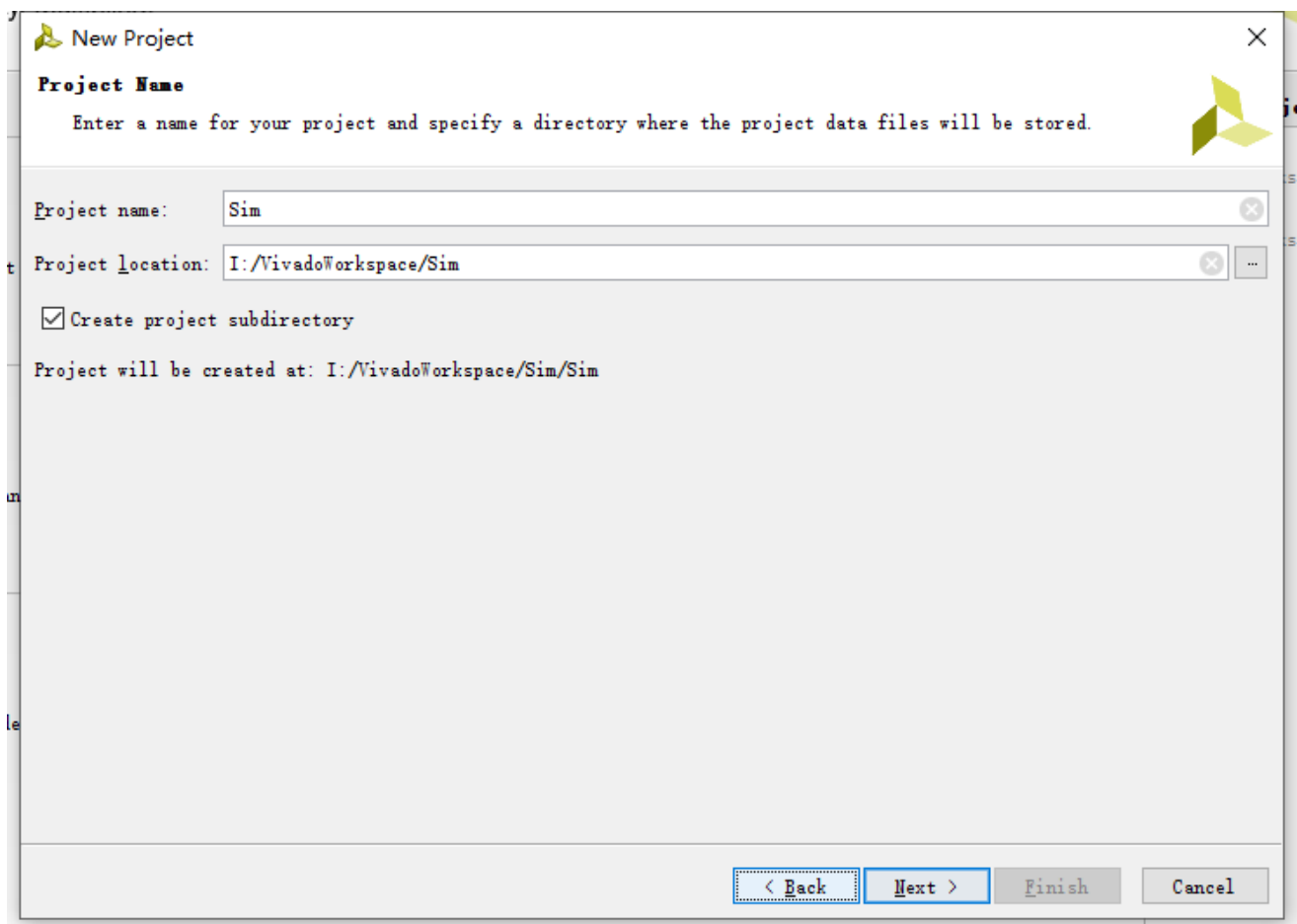
本教程主要介绍最基础和常用的行为级功能仿真。

导入源文件

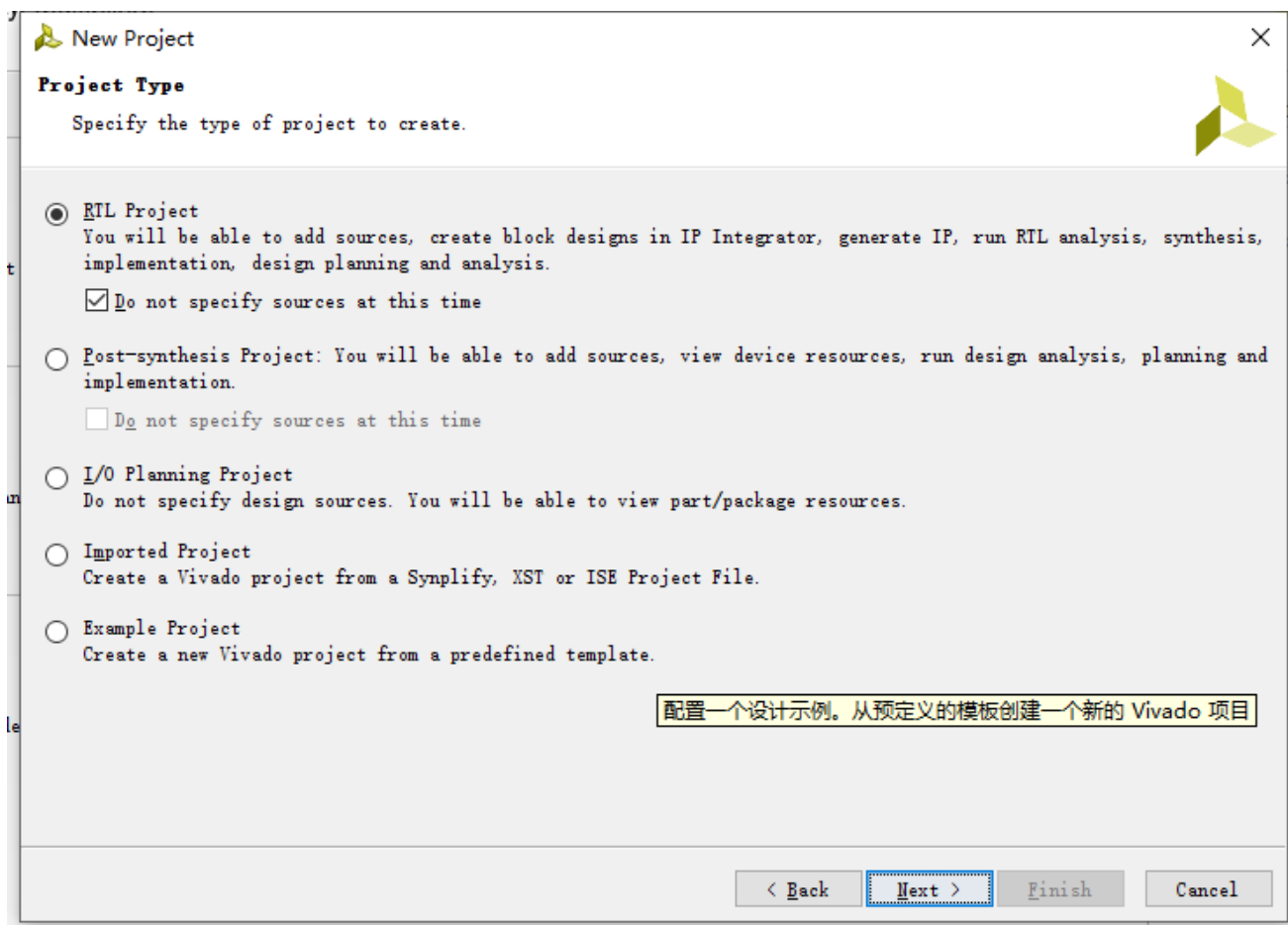
- 1.创建工程文件，对于已有工程文件可跳过**导入源文件**过程。



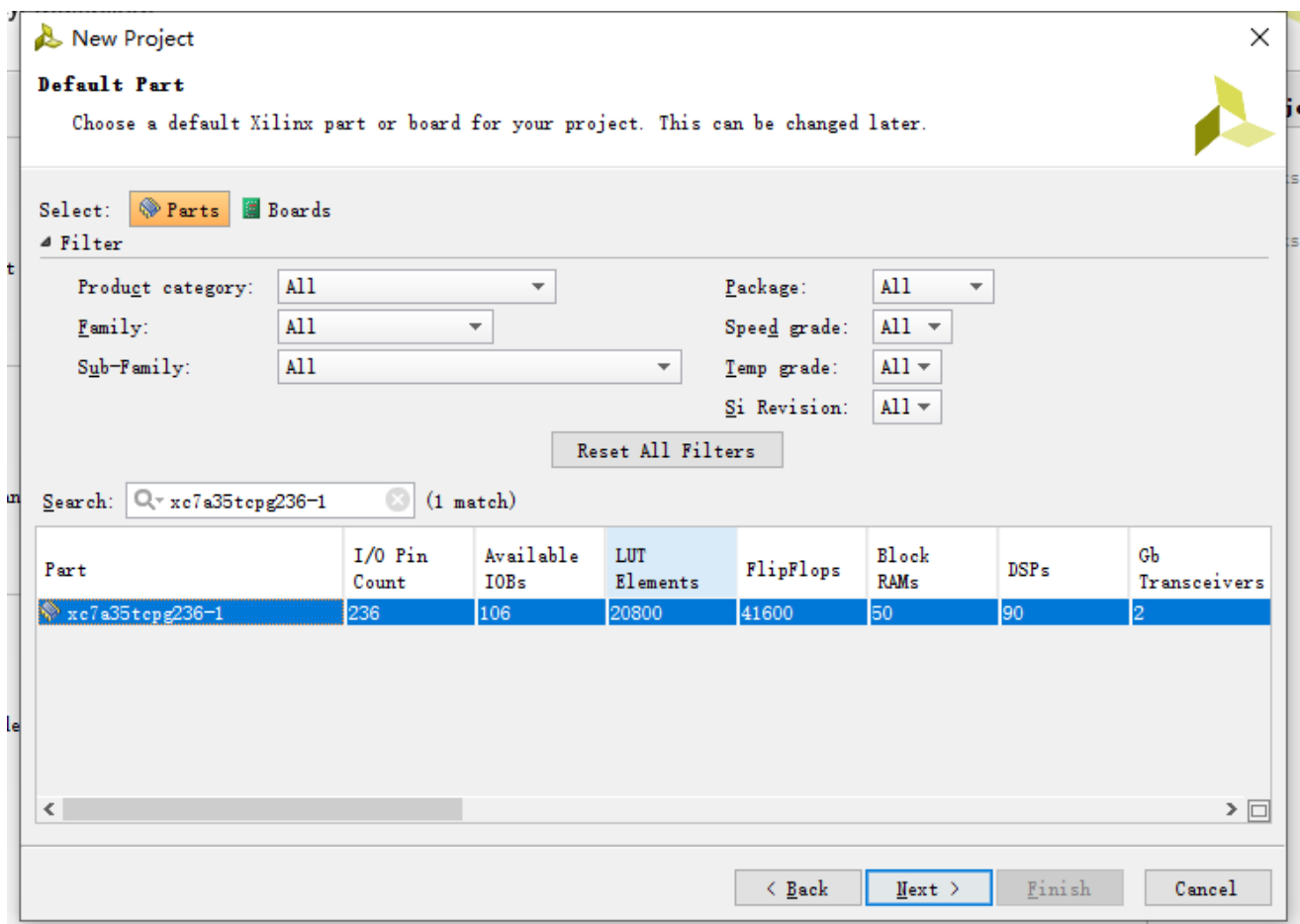
- 2.输入工程名和路径。



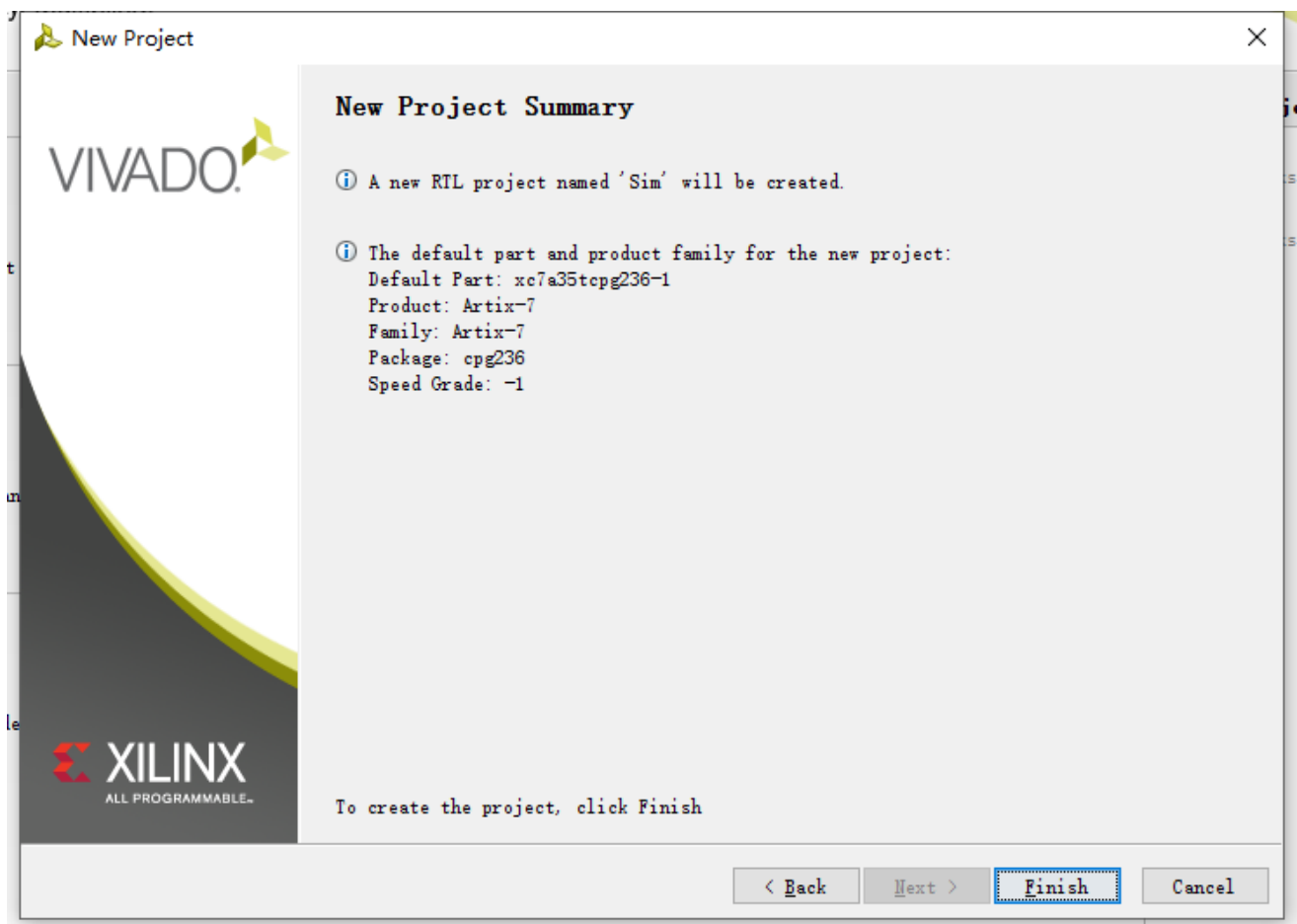
3. 点击Next ,出现选择工程类型的界面， 选择RTL Project， 并且选择 Do not specify sources at this time。



4. 选择硬件平台，选择xc7a35tcpg236-1。

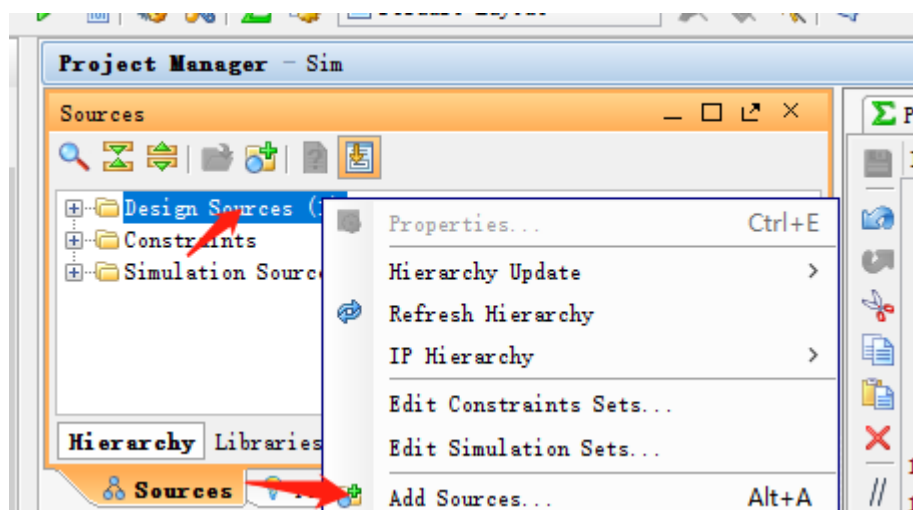


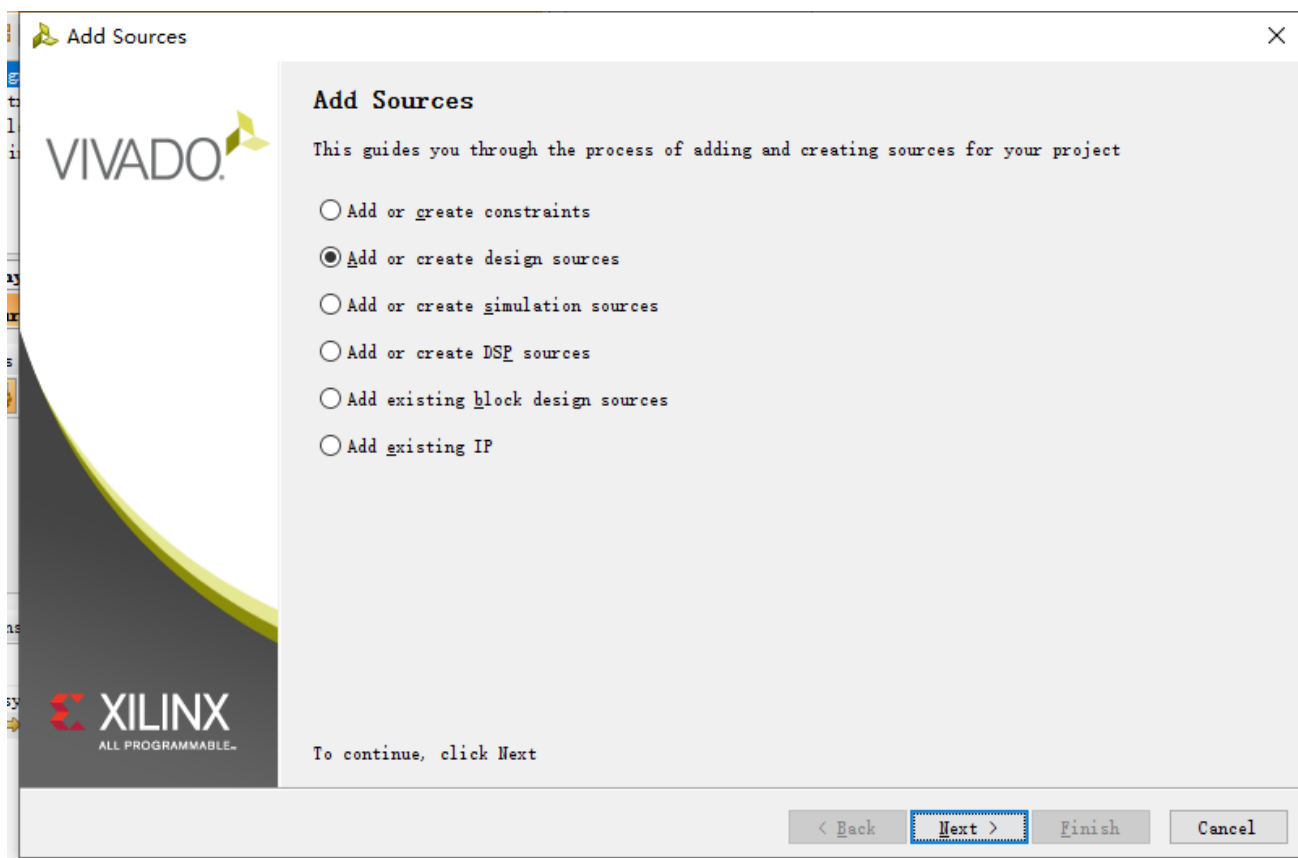
5. 点击next，出现Summary 窗口，点击Finish，工程就建立完成了。



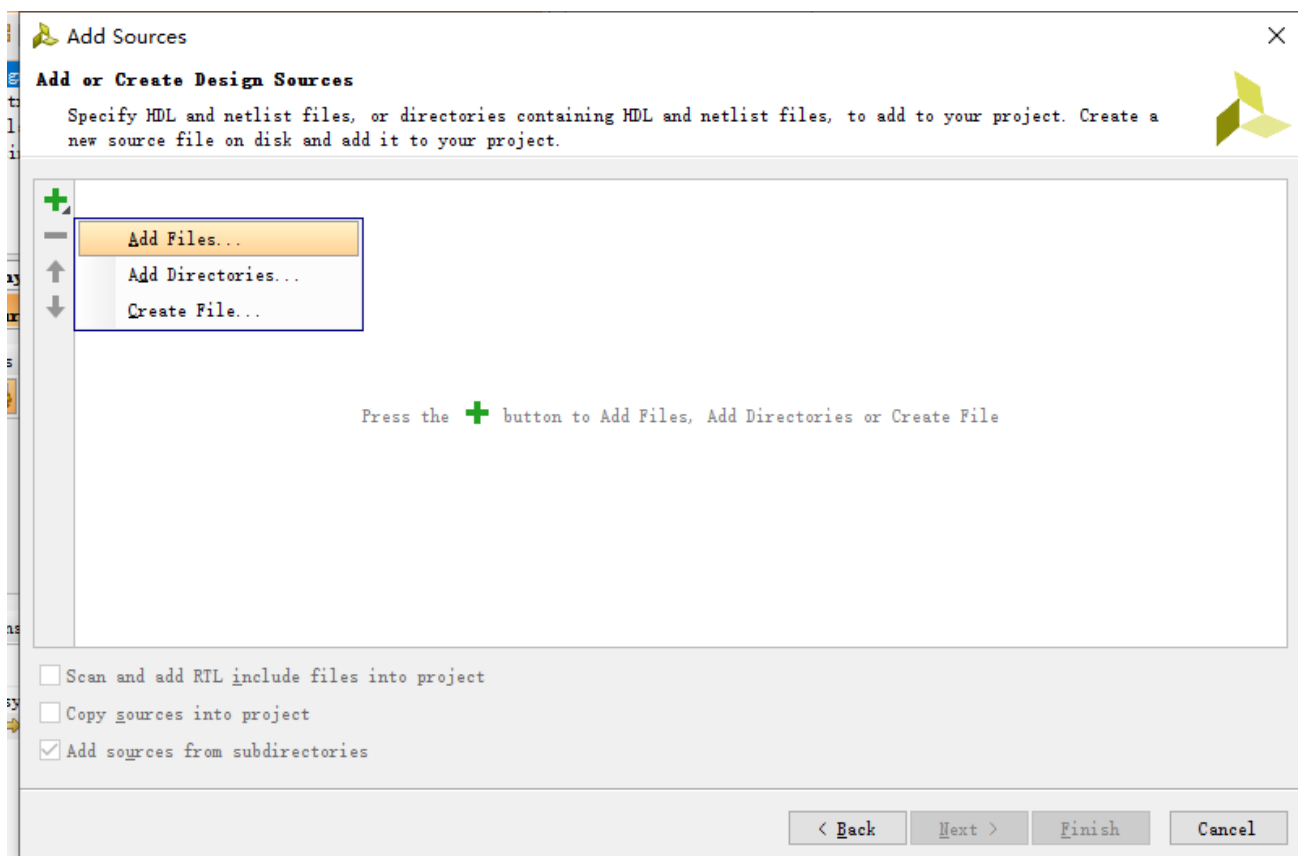
6.代码的导入或新建，本例子有2个源文件：待测试的Verilog 文件，testbench测试文件。

右键点击Design Source，选择Add Source。

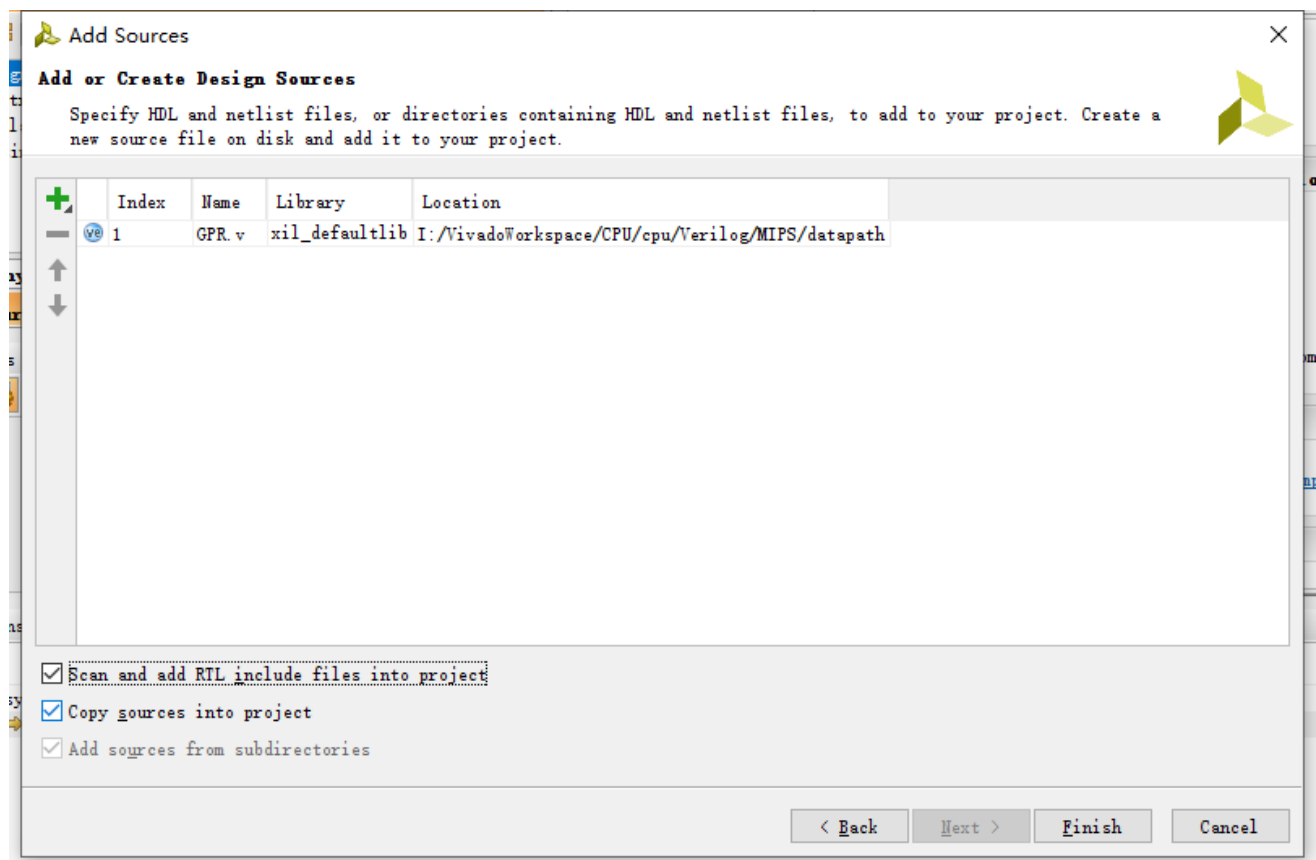




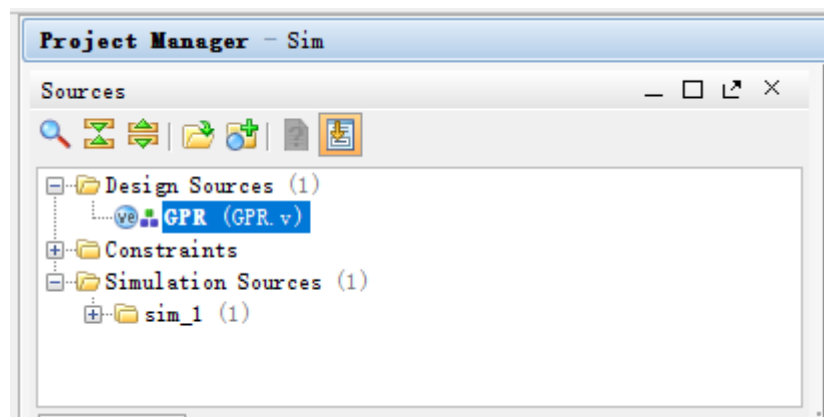
7. 选择是添加还是新建，这里添加已写好的文件。



以下两个勾选框，第一个用于自动搜索并添加文件的关联引用，第二个是否从源文件地址拷贝文件还是直接引用源文件。可以根据自己需求选择。完成后finish。



8.文件被添加到工程中。



其code如下：

```
module GPR(I_clk, I_reset, I_RegWrite, I_RW, I_RA, I_RB, I_busW, O_busA, O_busB);
    input      I_clk, I_reset, I_RegWrite;
    input  [4:0] I_RW, I_RA, I_RB;
    input  [31:0] I_busW;
    output [31:0] O_busA, O_busB;

    reg  [31:0] register [31:0];
    integer i;
```

```

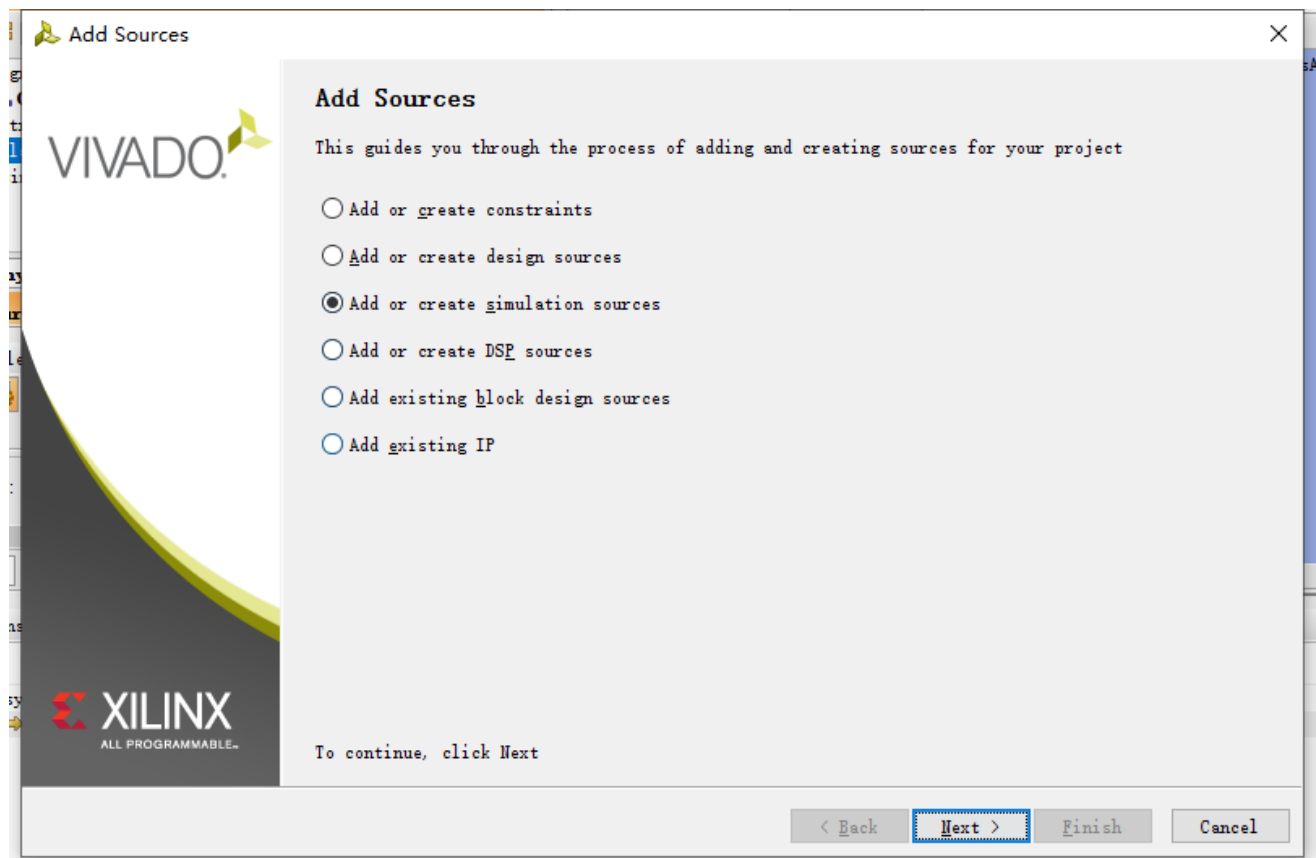
assign O_busA = register[I_RA];
assign O_busB = register[I_RB];

always @ ( posedge I_clk or posedge I_reset)
    if( I_reset )
        for( i = 0; i < 32; i = i + 1 )
            register[i] <= 32'b0;
    else
        if( I_RegWrite && I_RW != 0 )
            register[I_RW] <= I_busW;

endmodule

```

9.添加testbench文件。基本同样的操作，在simulate source 这里添加测试文件testbench.v 文件，以下不再重复图示了。



其code如下:

```

module testbench();
    reg        clk, reset, regWrite;
    reg  [4:0]  RW, RA, RB;
    reg  [31:0] busW;
    wire [31:0] busA, busB;

```



```

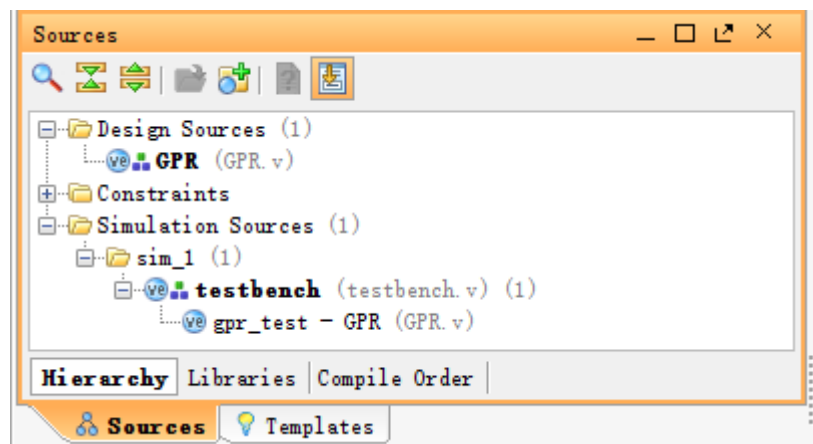
    GPR gpr_test(.I_clk(clk), .I_reset(reset), .I_RegWrite(regWrite), .I_RW(RW), .I_RA(RA),
    .I_RB(RB), .I_busW(busW),
    .O_busA(busA), .O_busB(busB));

    always
        #100 clk = ~clk;

    initial
    begin
        #0 reset = 1; clk = 0; regWrite=0; RA=0; RB=0;
        #100 reset = 0;
        #200 regWrite=1; busW=2; RW=0;
        #200 regWrite=1; busW=3; RW=1; RA=1;
        #200 regWrite=1; busW=4; RW=2; RB=2;
        #10000 $finish;
    end
endmodule

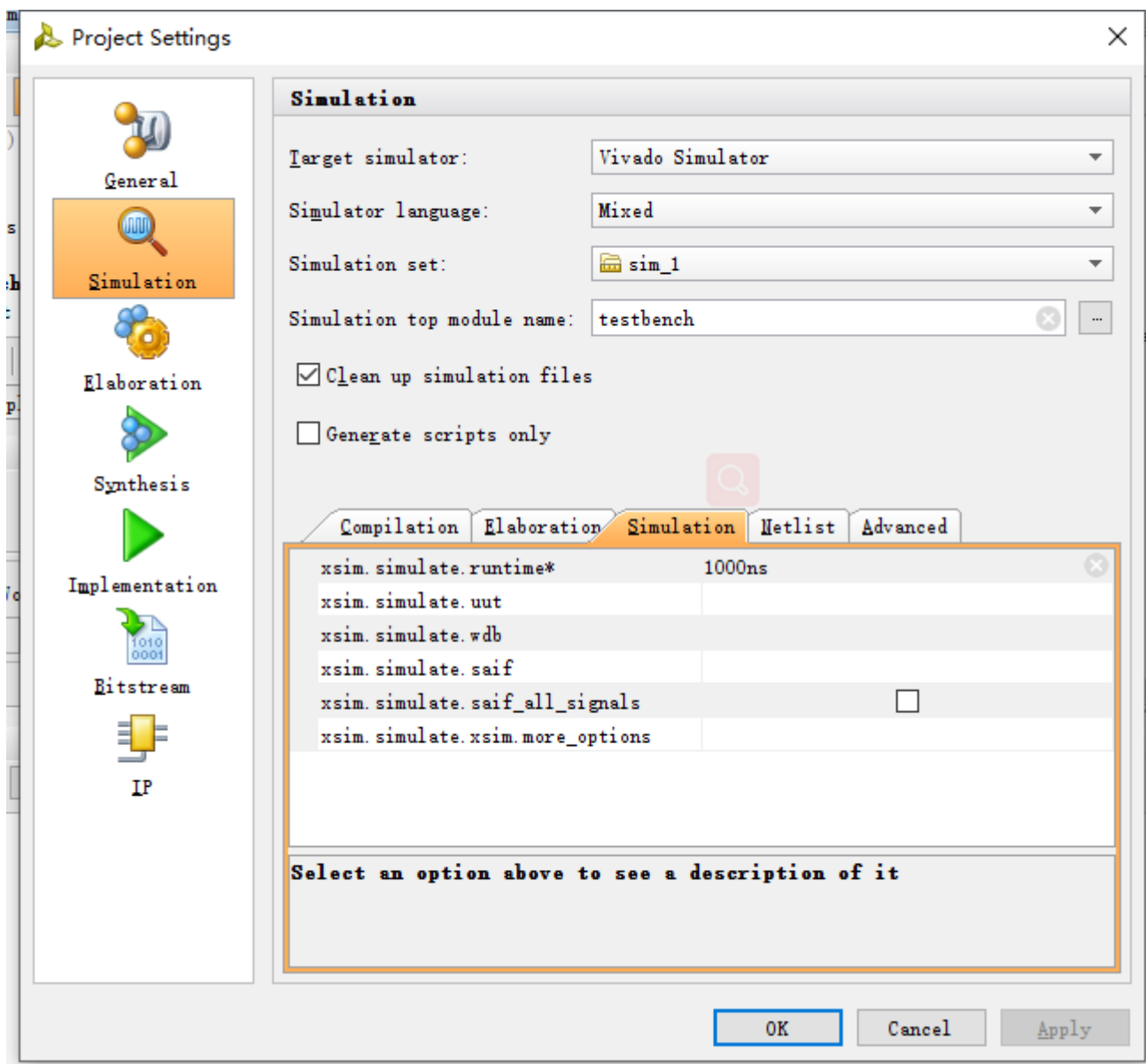
```

编写好testbench后的正确结构如下：



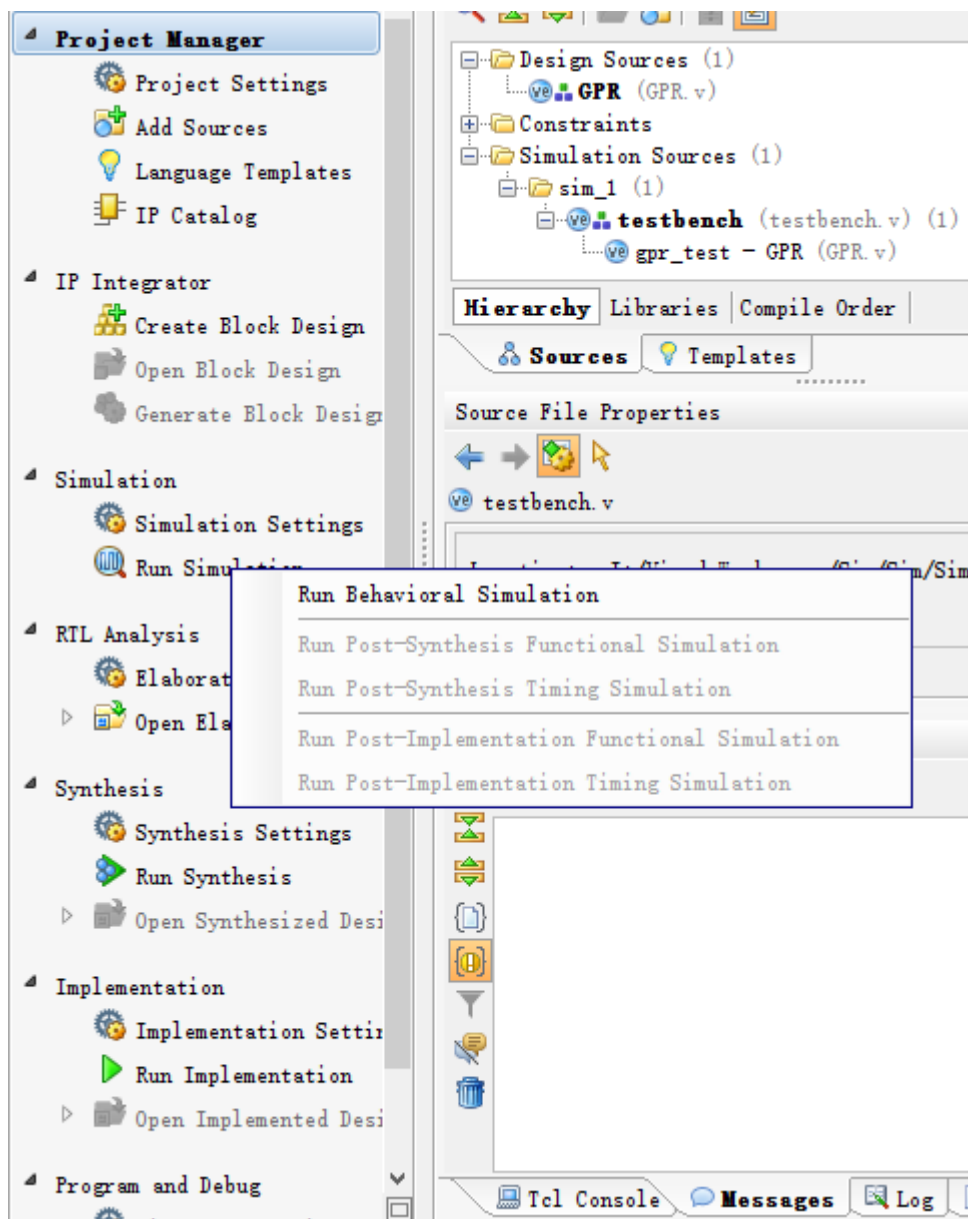
仿真

1. 仿真设置：点击主菜单 Flow->Simulation Settings 或者界面左边 Project Manager 下的 Simulation Settings 出现如下设置界面。

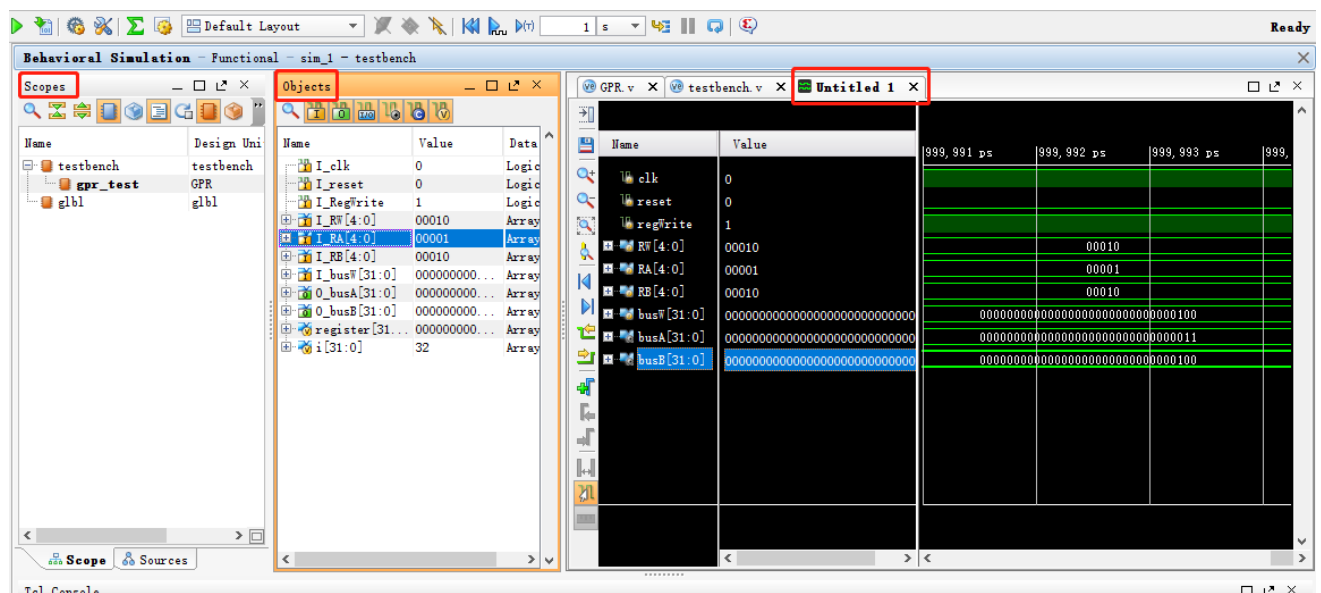


该界面可以配置仿真的相关信息，如：运行时时间，也可以在Target simulator中选择使用Modelsim等其他软件仿真。本文依旧使用默认配置仿真。

2.设置好之后， Flow->Run Simulation 或者左边直接点击，选择 Run Behavioral Simulation， 就是最上的那个选择。

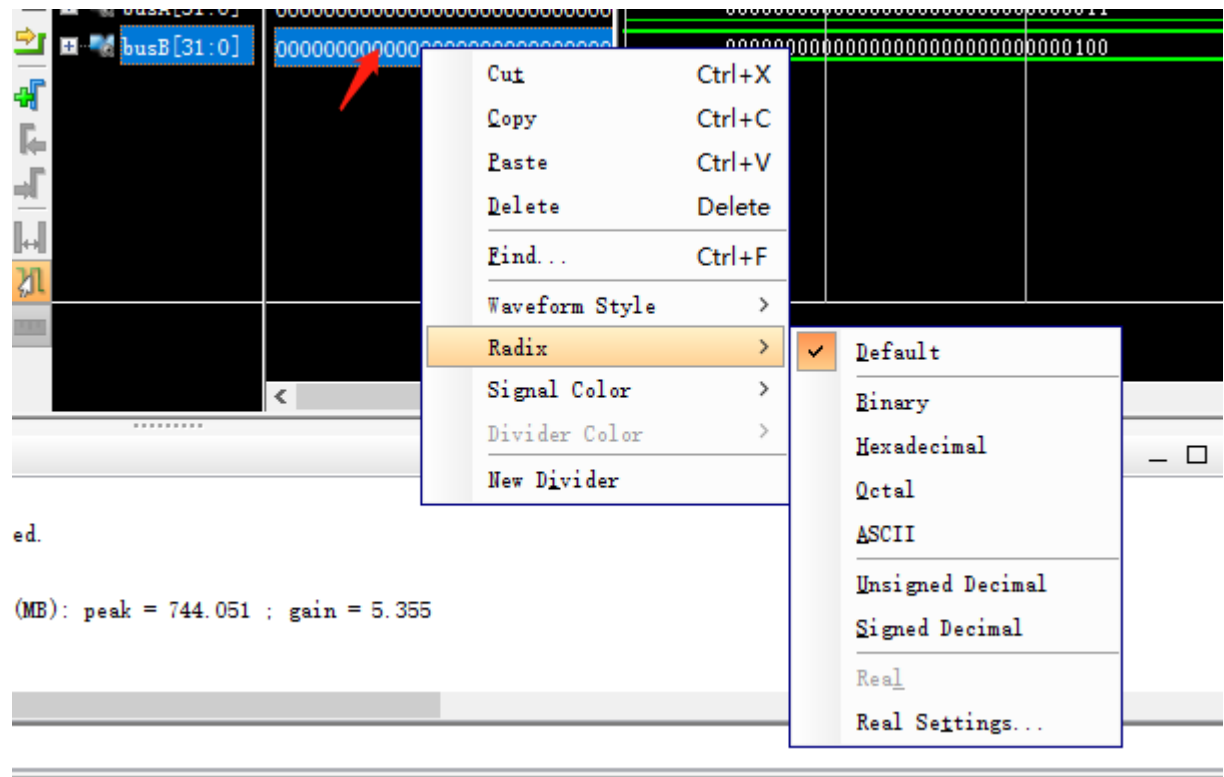


完成后，其界面如下：

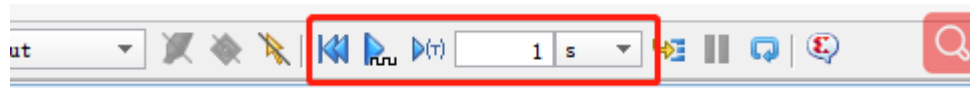


其中Scopes窗口对应于源文件选择，Object为选中源文件中module的reg、wire、变量和常量的选择，Untitled 1为仿真波形窗口。可以将Object中要检测的对象拖入到波形窗口中进行仿真监测。

右键相应的对象可以改变它（二、十、十六）进制的显示方式：



上方的工具栏可以选择Restart重置、Run All（运行到finish或stop）和间隔执行：



也可以在源文件设置打断点：

