

17. IP 核使用实验教程-PLL

17. 1PGX-Nano 开发板简介

PGX-Nano 开发板有一个 50M 的板载有源晶振，可以提供给 FPGA 板卡使用。

17. 2实验目的

利用 PLL IP 核产生多路时钟：并通过仿真查看波形

第一路频率为 100MHz，相位差为 0，占空比为 50%的单端信号；

第一路频率为 200MHz，相位差为 0，占空比为 50%的差分信号；

第三路频率为 100MHz，相位差为 50，占空比为 50%的单端信号；

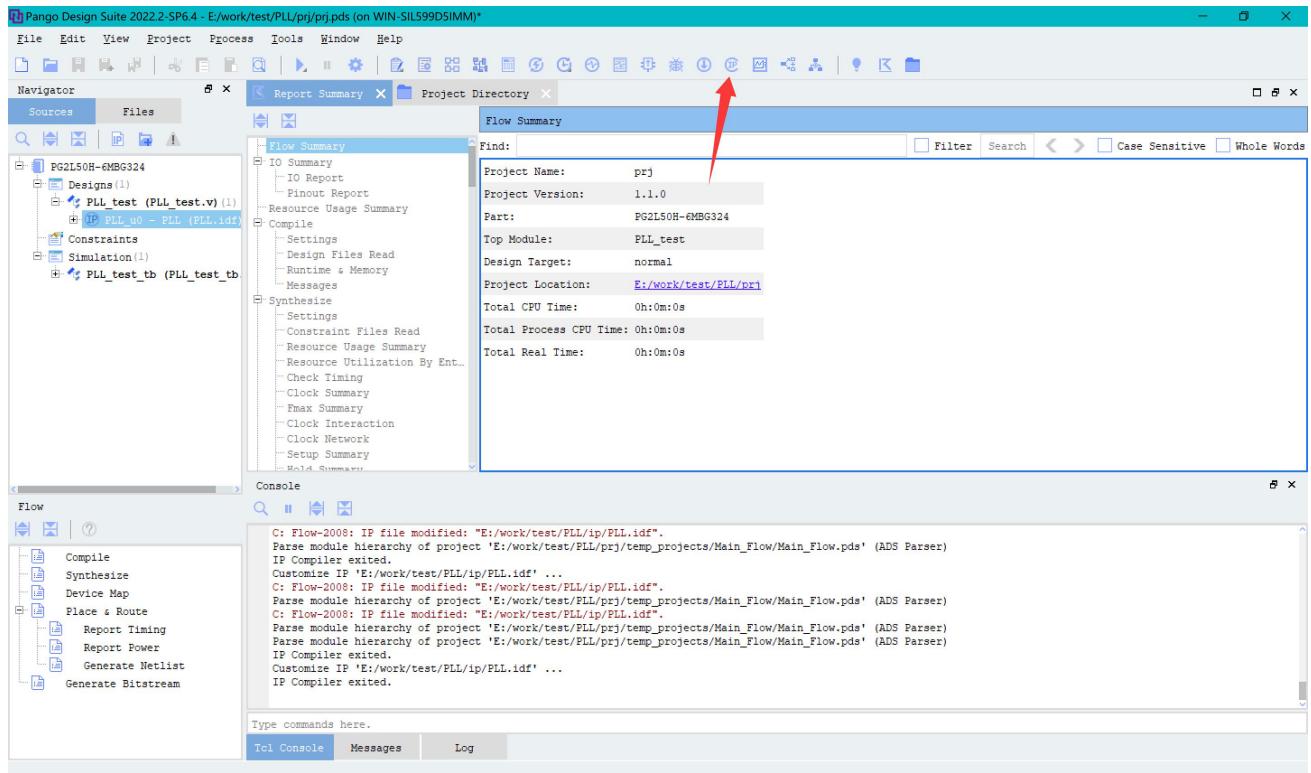
第四路频率为 200MHz，相位差为 0，占空比为 75%的单端信号。

17. 3实验原理

通过配置 FPGA 硬件资源上的 IP 硬核 PLL 生成所需要的时钟

17. 4 IP 核配置

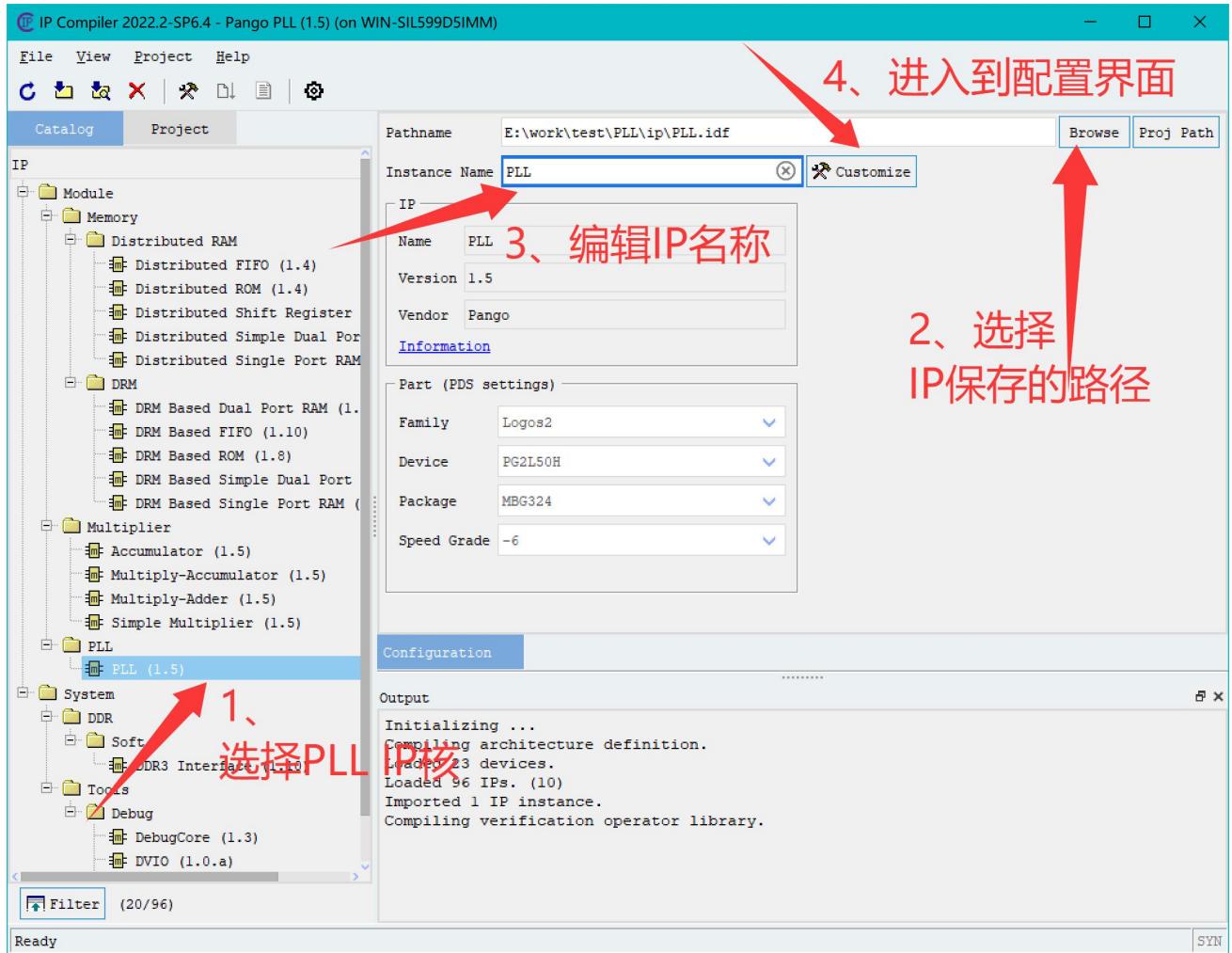
进入 PDS 软件之后点击 IP 按钮进入 IP 选择界面



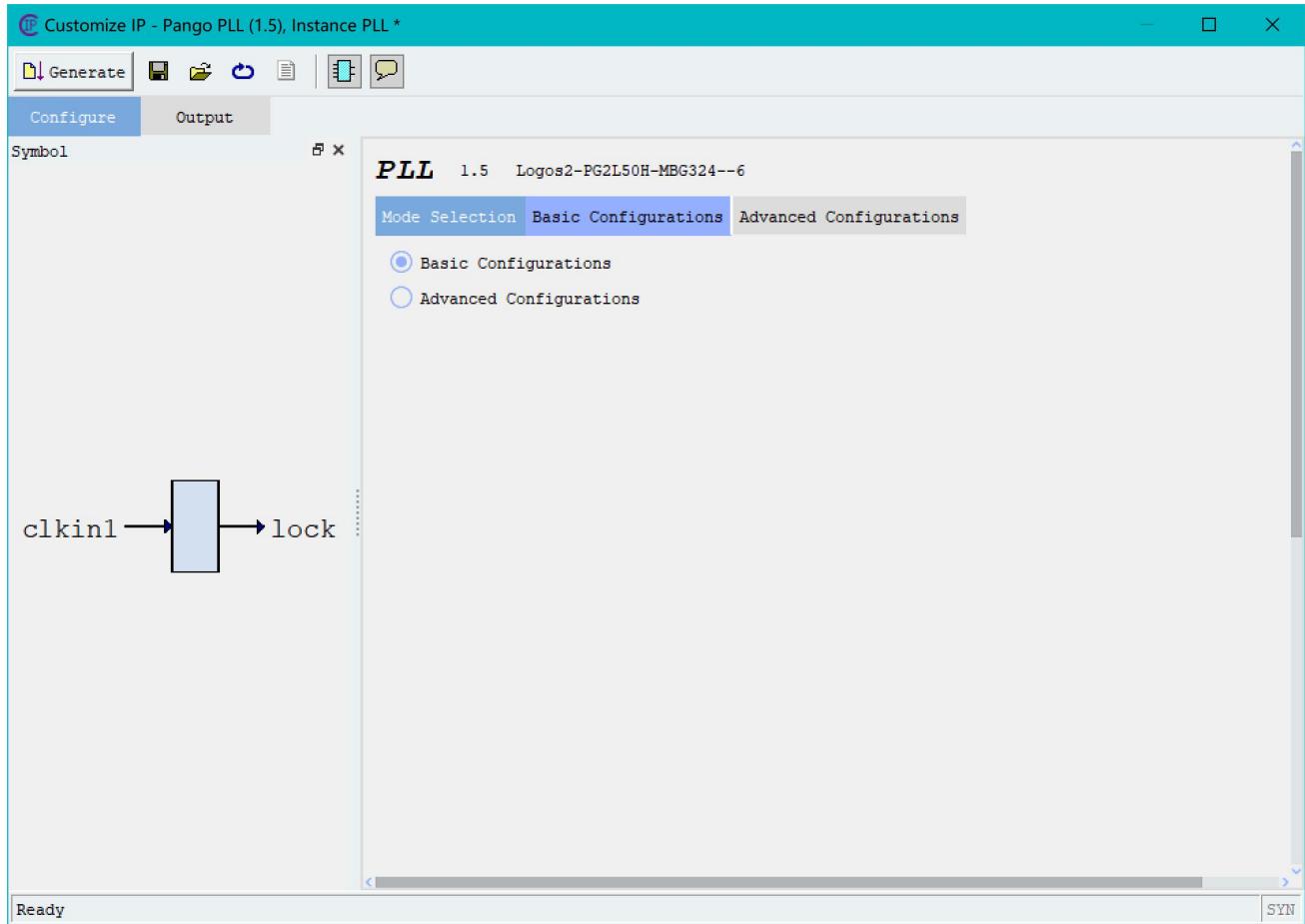
1、在最左侧选择 IP 为 PLL；

2、在 Pathname 这里选择 IP 所要存放的路径；

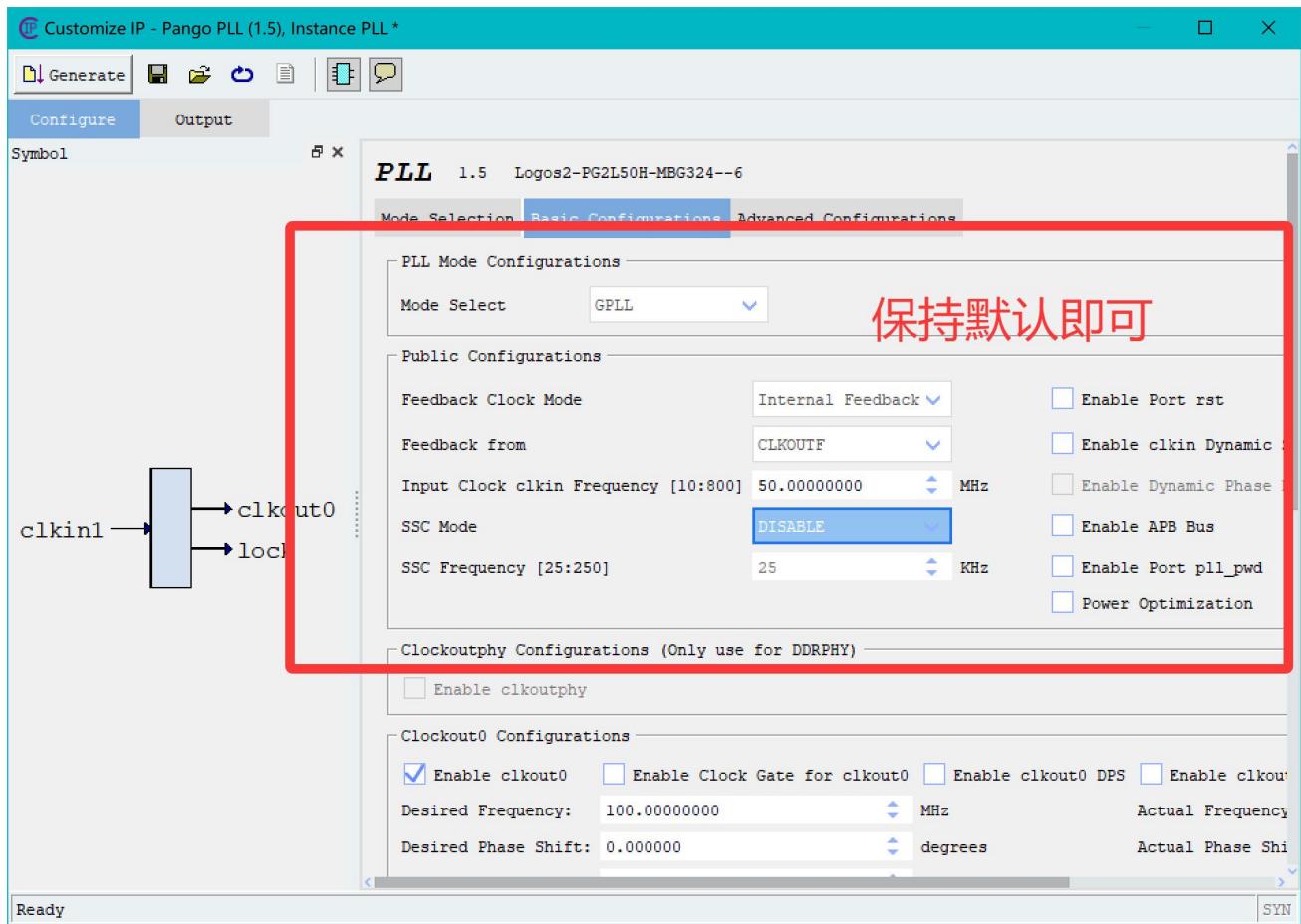
- 3、在 Instance Name 这里填写用户自定义的 IP 名称，该名称为最终 IP 核被调用的名称
- 4、点击 Customize 进入 IP 核配置界面



进入到 IP 配置界面后我们在 Mode Selection 这栏选择基本配置，对于我们一般的使用而言基本配置就可以满足需要，我们保持默认即可

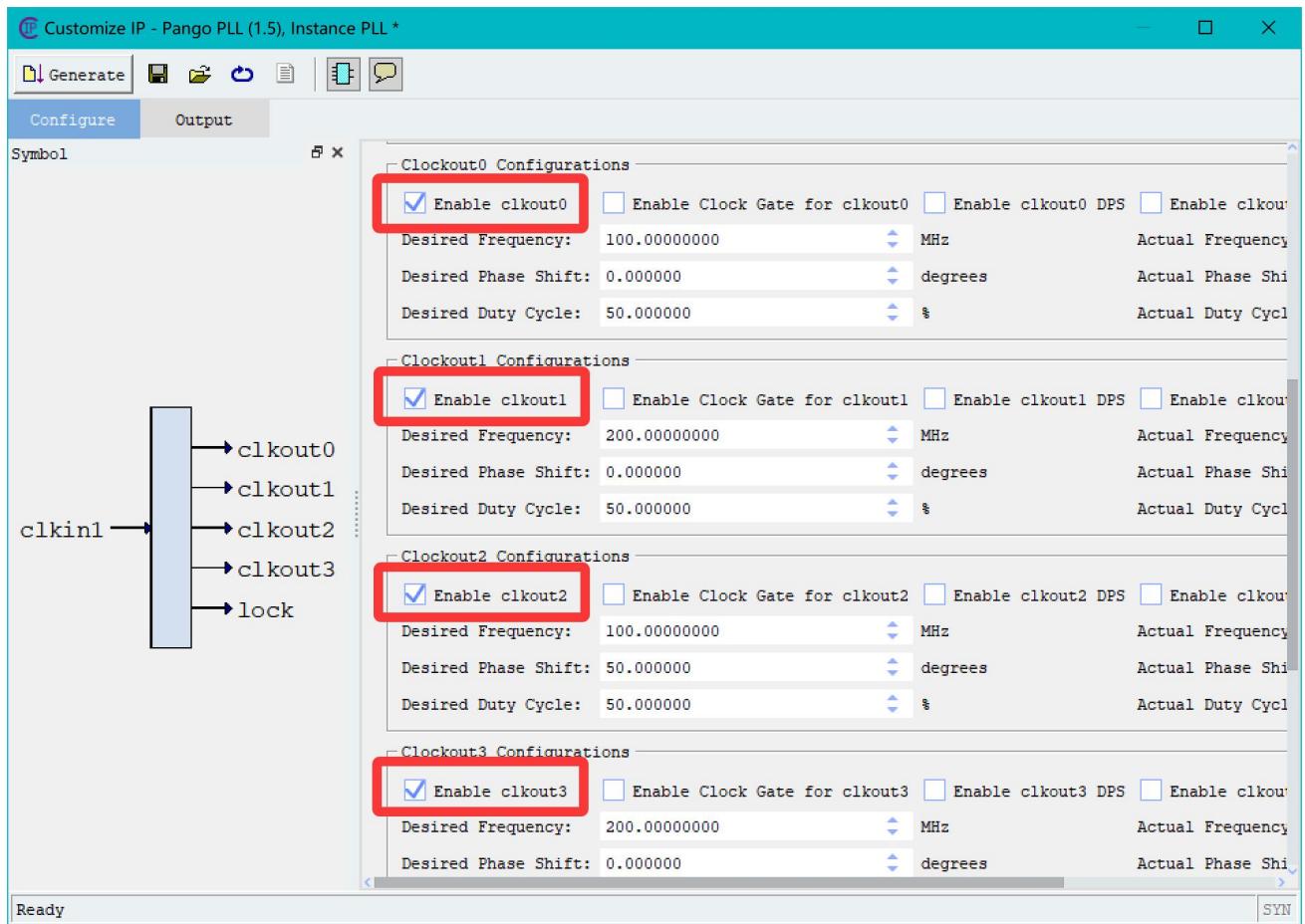


下一个配置界面是我们的重头戏，需要配置我们所需要的时钟频率，相位差，占空比，等等。开发板的 50MHz 时钟信号作为 PLL 的时钟输入，因此上面这些选项我们保持默认即可，详细的解释可以参考手册。

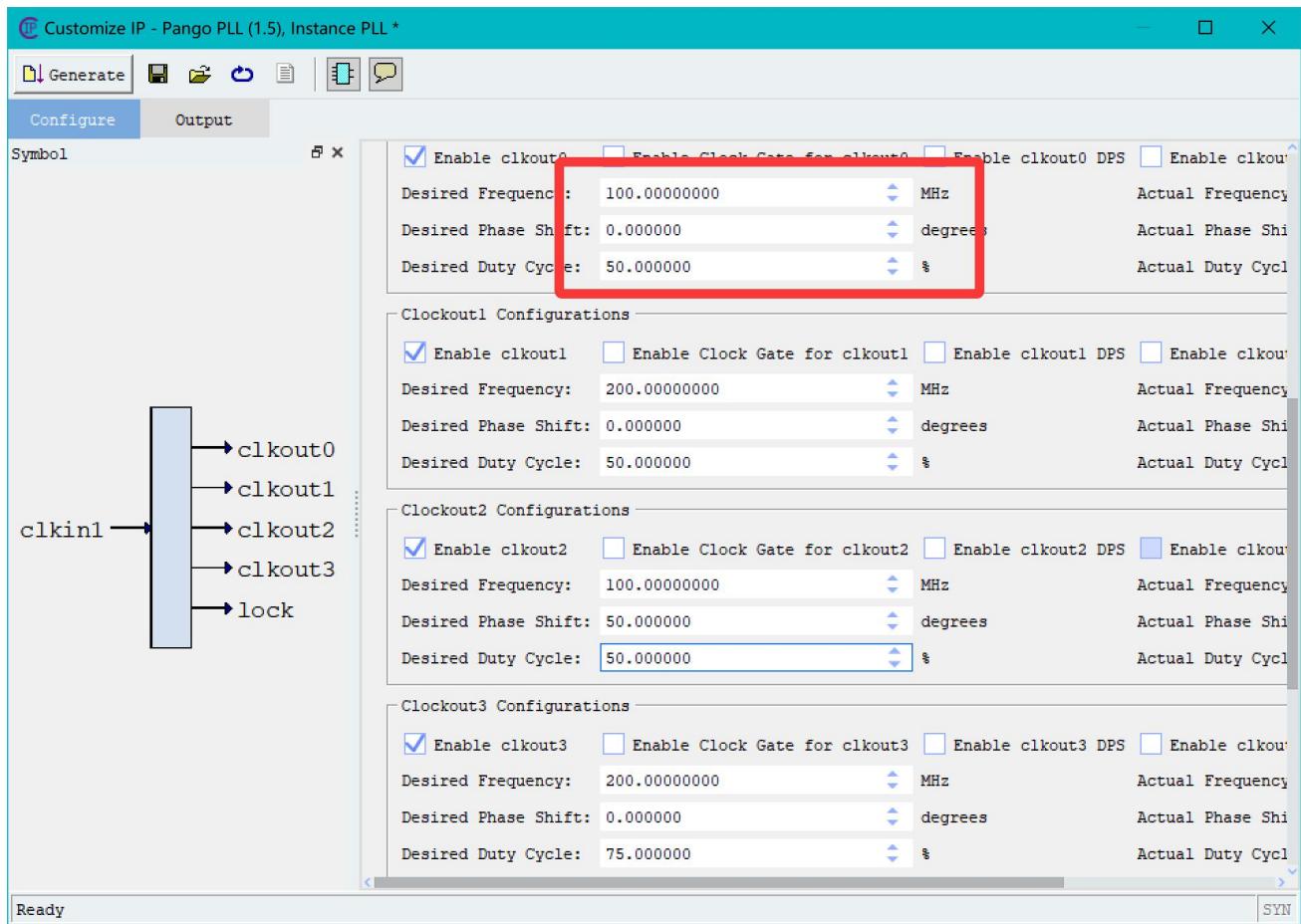


接下来配置我们需要输出几路时钟以及输出时钟的参数

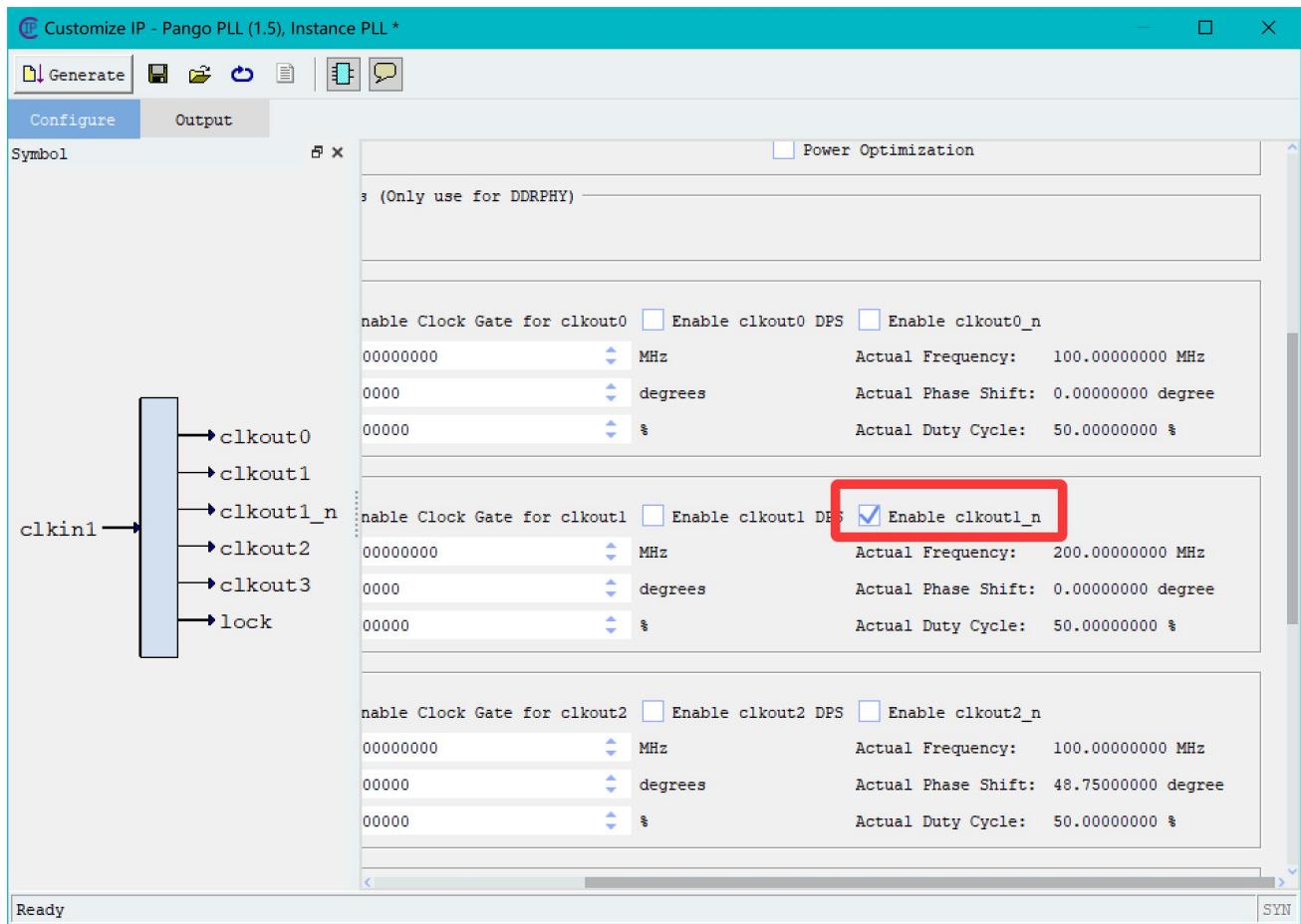
我们需要四路时钟，因此我们 Enable clkout0/clkout1/clkout2/clkout3



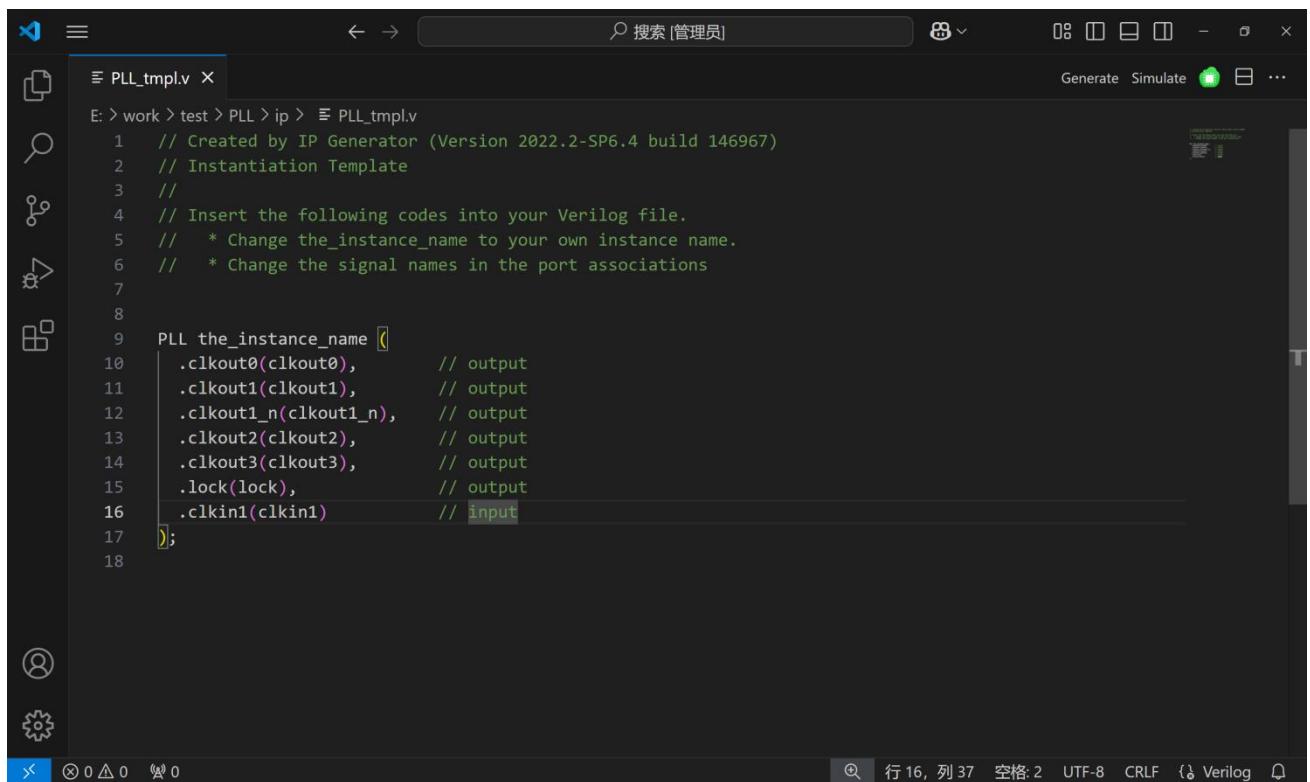
我们所需要的时钟频率，相位，占空比在下图中配置



在右侧还可以选择我们输出的时钟是否为差分时钟



其余选项我们保持默认即可，常用的就这几个，我们点击左上角的 Generate 生成我们的 IP，生成之后自动弹出我们的 IP 例化模板，我们只需要复制即可完成 IP 核的调用

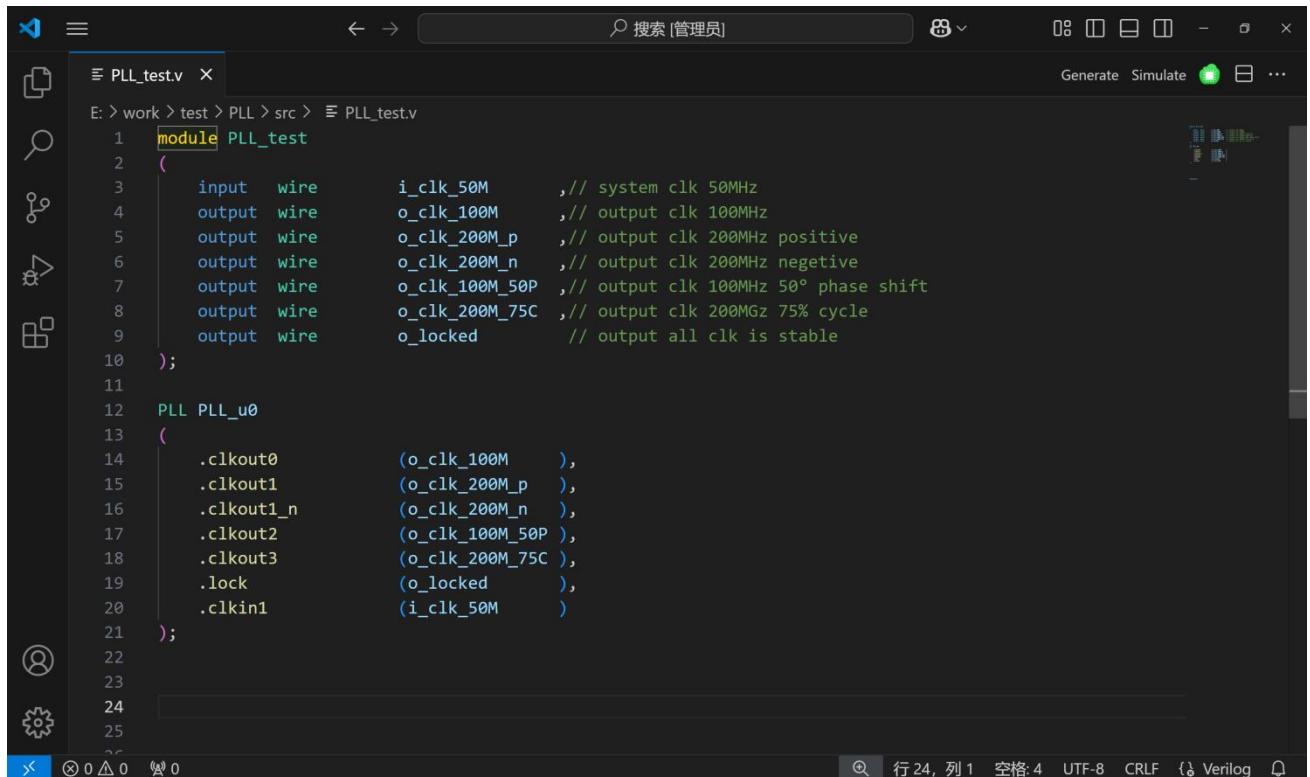


```

E: > work > test > PLL > ip > .PLL_tmp1.v
1 // Created by IP Generator (Version 2022.2-SP6.4 build 146967)
2 // Instantiation Template
3 //
4 // Insert the following codes into your Verilog file.
5 // * Change the_instance_name to your own instance name.
6 // * Change the signal names in the port associations
7
8
9 PLL the_instance_name [
10   .clkout0(clkout0),           // output
11   .clkout1(clkout1),           // output
12   .clkout1_n(clkout1_n),       // output
13   .clkout2(clkout2),           // output
14   .clkout3(clkout3),           // output
15   .lock(lock),                // output
16   .clkin1(clkin1)             // input
17 ];
18

```

例化我们的 IP 模板并封装在 PLL_test 这个 module 内

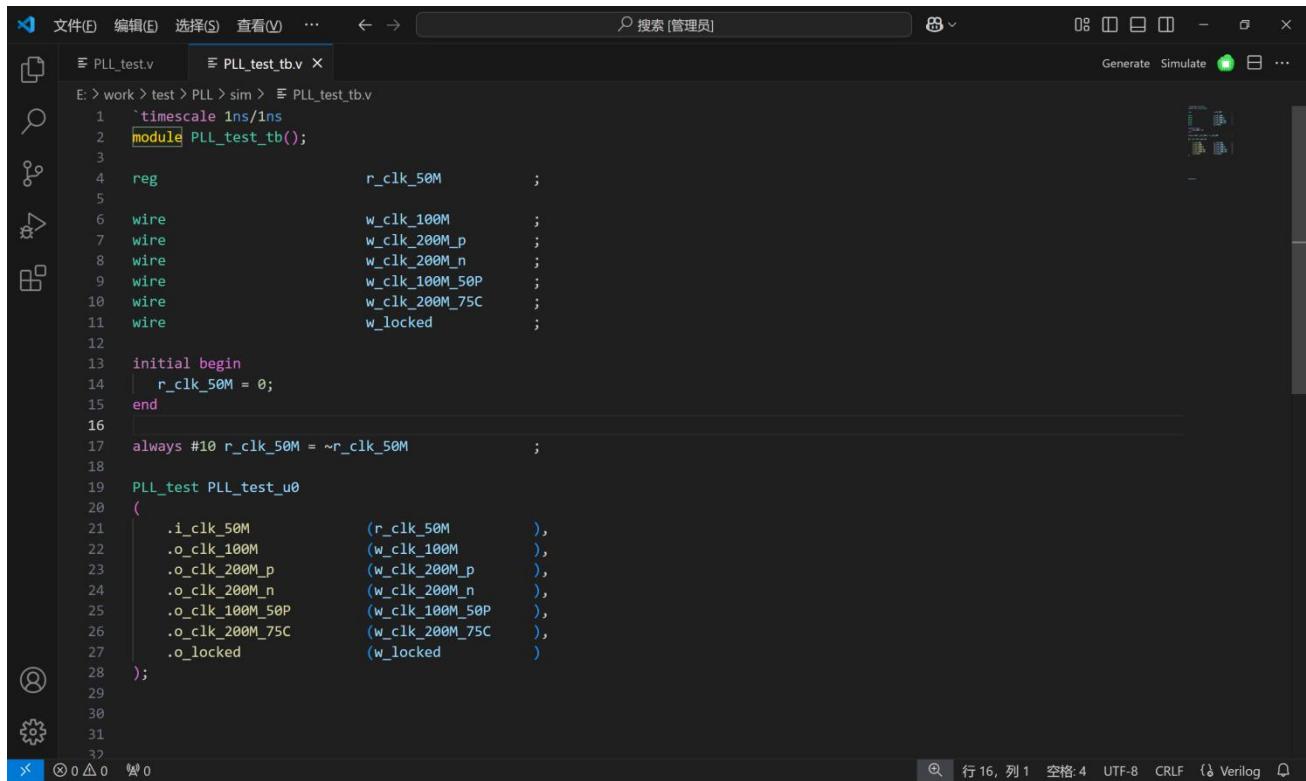


```

E: > work > test > PLL > src > .PLL_test.v
1 module PLL_test
2 (
3   input  wire    i_clk_50M      ,// system clk 50MHz
4   output wire    o_clk_100M    ,// output clk 100MHz
5   output wire    o_clk_200M_p   ,// output clk 200MHz positive
6   output wire    o_clk_200M_n   ,// output clk 200MHz negative
7   output wire    o_clk_100M_50P ,// output clk 100MHz 50° phase shift
8   output wire    o_clk_200M_75C ,// output clk 200MHz 75% cycle
9   output wire    o_locked      // output all clk is stable
10 );
11
12   PLL  PLL_u0
13   (
14     .clkout0      (o_clk_100M      ),
15     .clkout1      (o_clk_200M_p    ),
16     .clkout1_n    (o_clk_200M_n    ),
17     .clkout2      (o_clk_100M_50P ),
18     .clkout3      (o_clk_200M_75C ),
19     .lock         (o_locked      ),
20     .clkin1       (i_clk_50M      )
21 );
22
23
24
25

```

因为我们的模块只有一个 50MHz 的时钟作为输入端口，因此仿真文件只需要产生一个 50MHz 的激励即可。50MHz 转化为时间即 20ns，我们 10ns 进行一次翻转

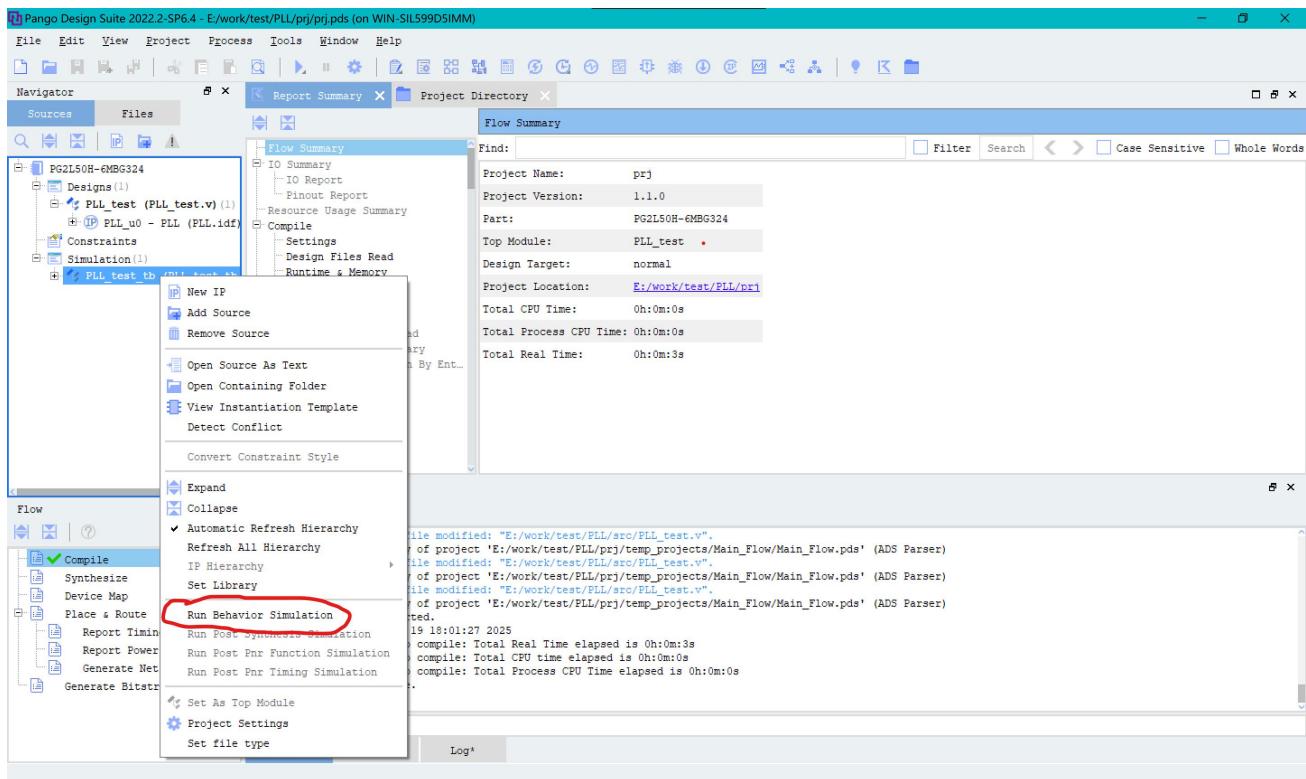


```

1 `timescale 1ns/1ns
2 module PLL_test_tb();
3
4 reg r_clk_50M;
5
6 wire w_clk_100M;
7 wire w_clk_200M_p;
8 wire w_clk_200M_n;
9 wire w_clk_100M_50P;
10 wire w_clk_200M_75C;
11 wire w_locked;
12
13 initial begin
14     r_clk_50M = 0;
15 end
16
17 always #10 r_clk_50M = ~r_clk_50M;
18
19 PLL_test PLL_test_u0
20 (
21     .i_clk_50M(r_clk_50M),
22     .o_clk_100M(w_clk_100M),
23     .o_clk_200M_p(w_clk_200M_p),
24     .o_clk_200M_n(w_clk_200M_n),
25     .o_clk_100M_50P(w_clk_100M_50P),
26     .o_clk_200M_75C(w_clk_200M_75C),
27     .o_locked(w_locked)
28 );
29
30
31
32

```

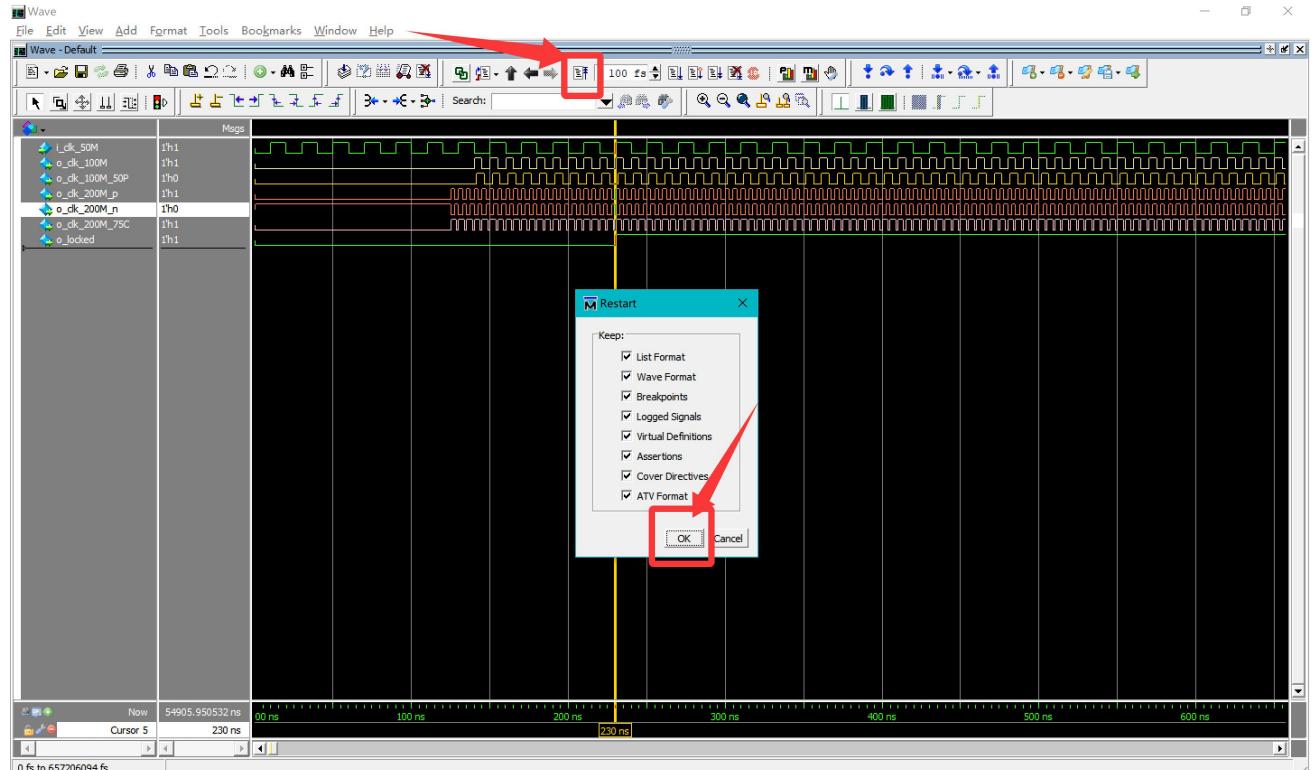
编写完所有代码后编译发现无报错，我们右键仿真文件进行仿真



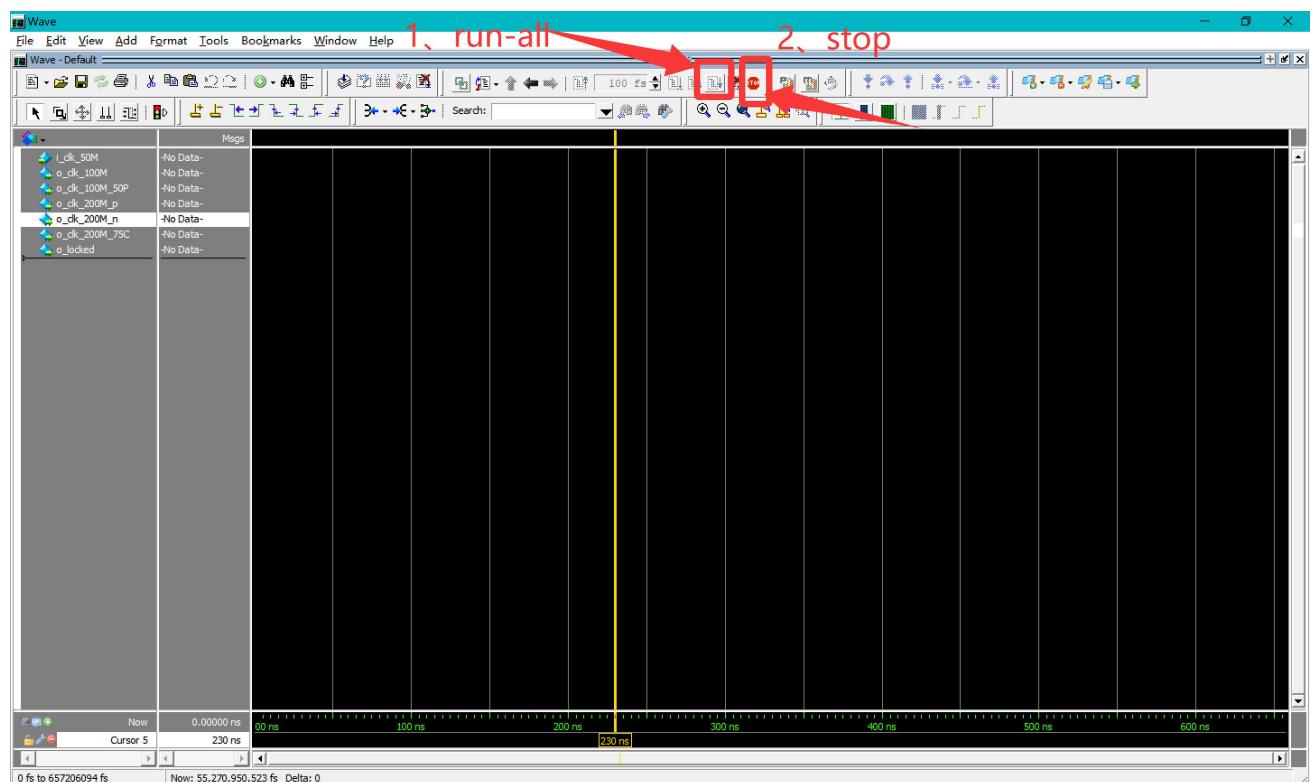
17.515.6 仿真现象

进入 modelsim 后我们如果想要查看模块信号的波形为不是仿真文件中的信号我们可以点

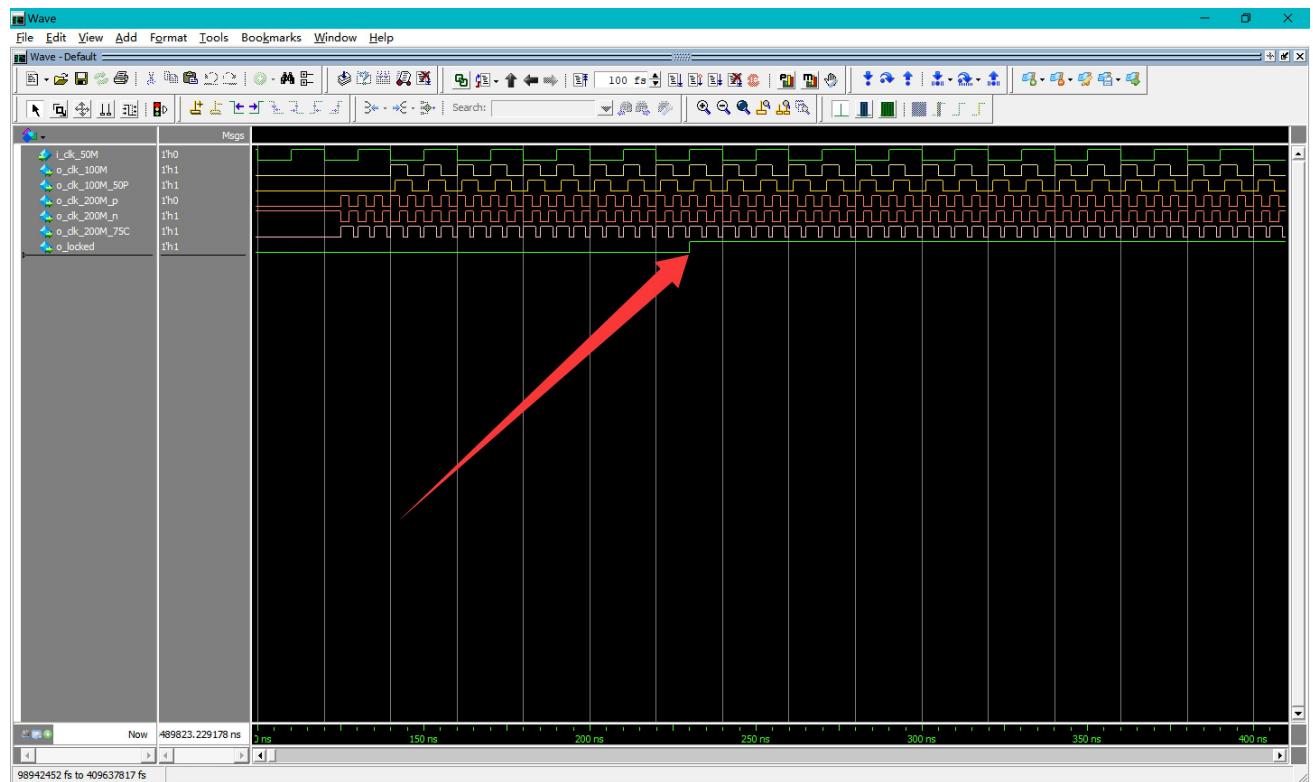
击要查看的模块，右侧是模块的信号，之后右键 Add wave 即可添加到波形中。注意我们刚刚加进来的波形部分没有加载，我们需要点击 restart 按钮复位一下，弹出的窗口点击 ok



复位之后我们需要开始仿真只需要点击 run-all 即可，停止只需点击旁边的红色按钮 (stop) 即可停止仿真



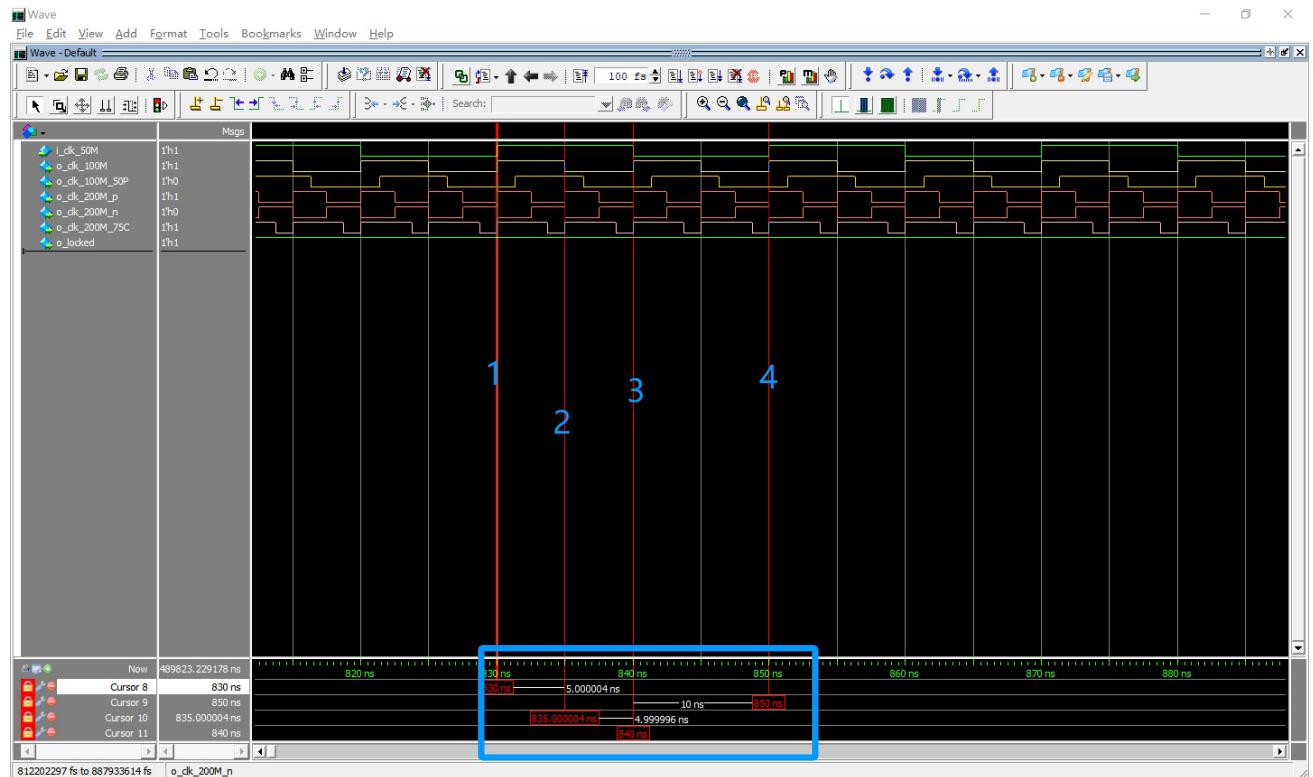
我们可以看到 PLL 的 lock 信号 (o_locked) 是在时钟稳定之后才拉高



1 到 4 之间是 i_clk_50M 的 cycle, 为 $5+5+10 = 20$ ns, 对应 50MHz, 数据正确;

1 到 3 之间是 o_clk_100M 的 cycle, 为 $5+5 = 10$ ns 对应 100MHz, 数据正确;

1 到 2 之间是 o_clk_200M 的 cycle, 为 5ns, 对应 200MHz, 数据正确;



下图中可以观测到 1 到 2 之间相位差为 50° ，时钟频率都为 100MHz。

3 到 6 为 200MHz 时钟的一个 cycle: 5ns;

3 到 4 是 o_clk_200M 的占空比, 为 2.5ns 即 50% , 数据正确;

3 到 5 是 $o_clk_200M_75C$ 的占空比, 为 $2.5+1.25 = 3.75\text{ns}$ 即 75% , 数据正确;

从波形中还可以观测到 $o_clk_200M_p$ 和 $o_clk_200M_n$ 互补, 说明为差分信号, 现象正确

