ADA106
实验教程

Rev. 1.0

版本记录

版本	时间	描述
Rev. 1.0	2019-03-18	第一版发布

版权声明：

免责声明

目录

本实验中, 程序要输出 8 位的正弦波数据作为 ADA106 模块的 DA 数字输入，产生一个正弦波信号, 再将这个正弦波信号通过 SMA 线输入 AD 的模拟口, FPGA 板卡采集 AD 的数据，通过内部 ILA 显示采集的数据

1. 制作 coe 文件

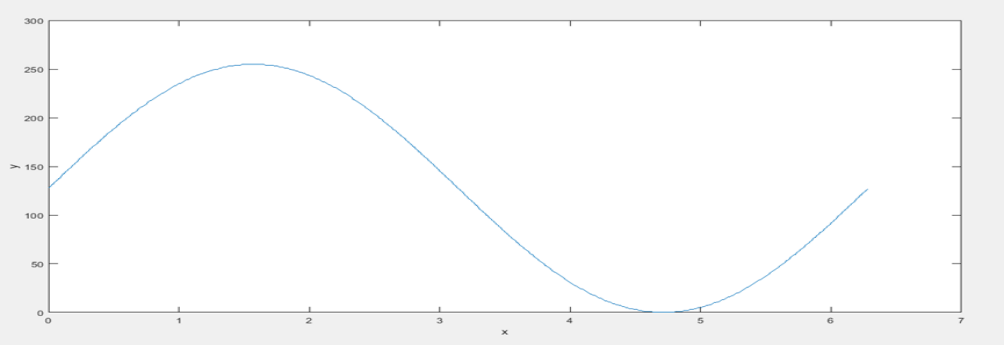
在 matlab 中

```
X=linspace(0,2*pi,512);
Y=(255/2)*(sin(X)+1);
Y=uint8(Y);
plot(X,Y);
dlmwrite('sin512.txt',Y);//保存数据为文本文件，数据之间以逗号结尾
MEMORY_INITIALIZATION_RADIX=16;
MEMORY_INITIALIZATION_VECTOR=
```

更改文件 sin512.txt 后缀为.coe，得到 sin512.coe

sin512.coe

```
MEMORY_INITIALIZATION_RADIX=16;
MEMORY_INITIALIZATION_VECTOR=
80,
81,
83,
84,
86,
87,
89,
8A,
8C,
8E,
8F,
91,
92,
94,
95,
97,
98,
9A,
9B,
9D,
9E,
A0,
A2,
```

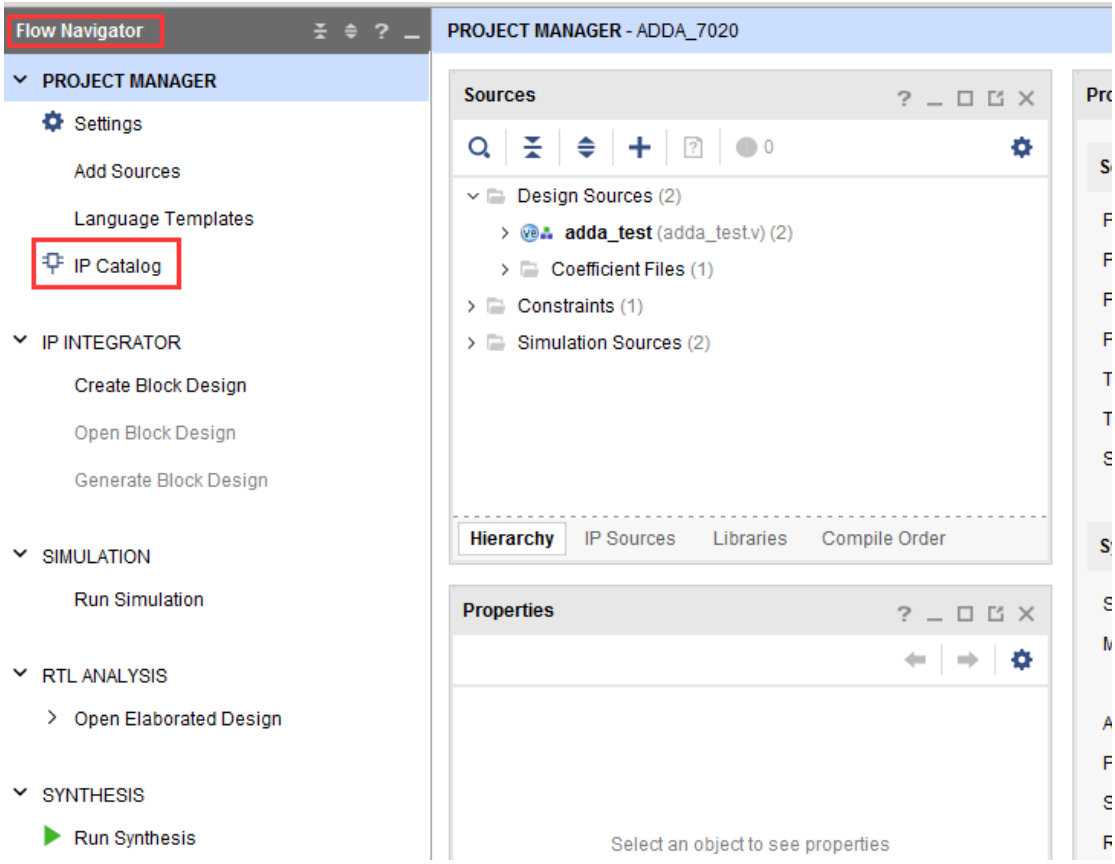


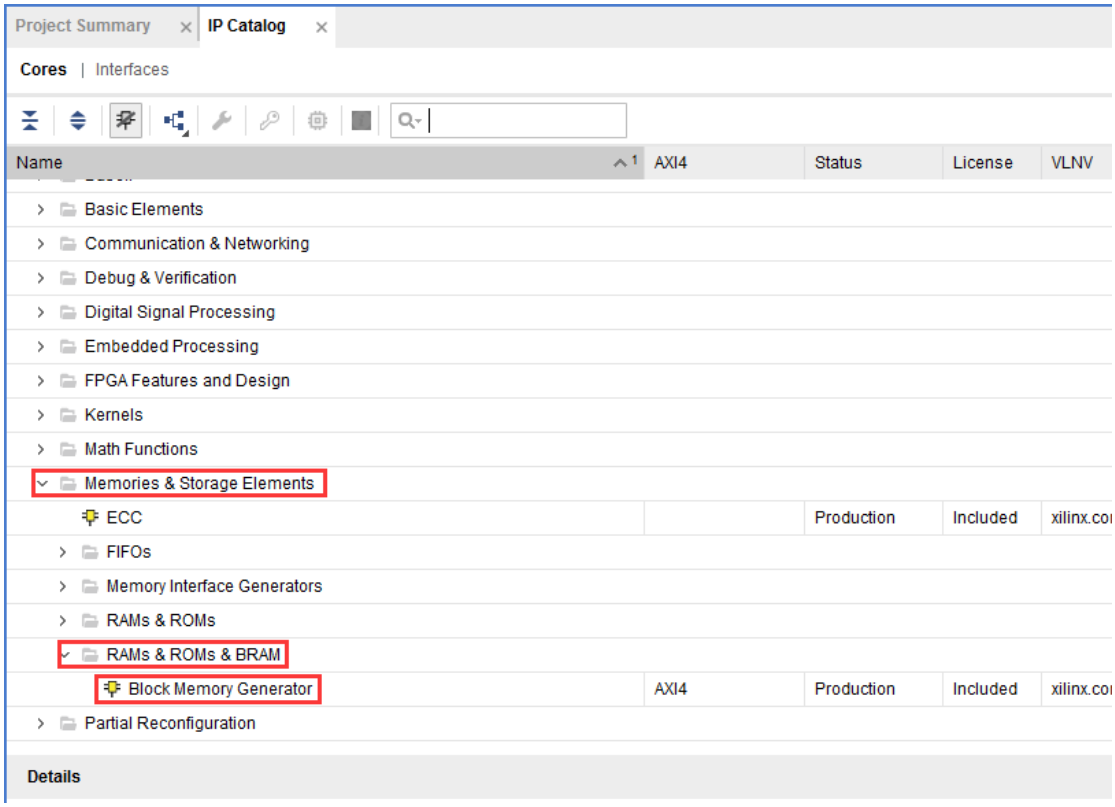
2. 新建工程,工程名为 ADDA_7020

点击 File -> New Project

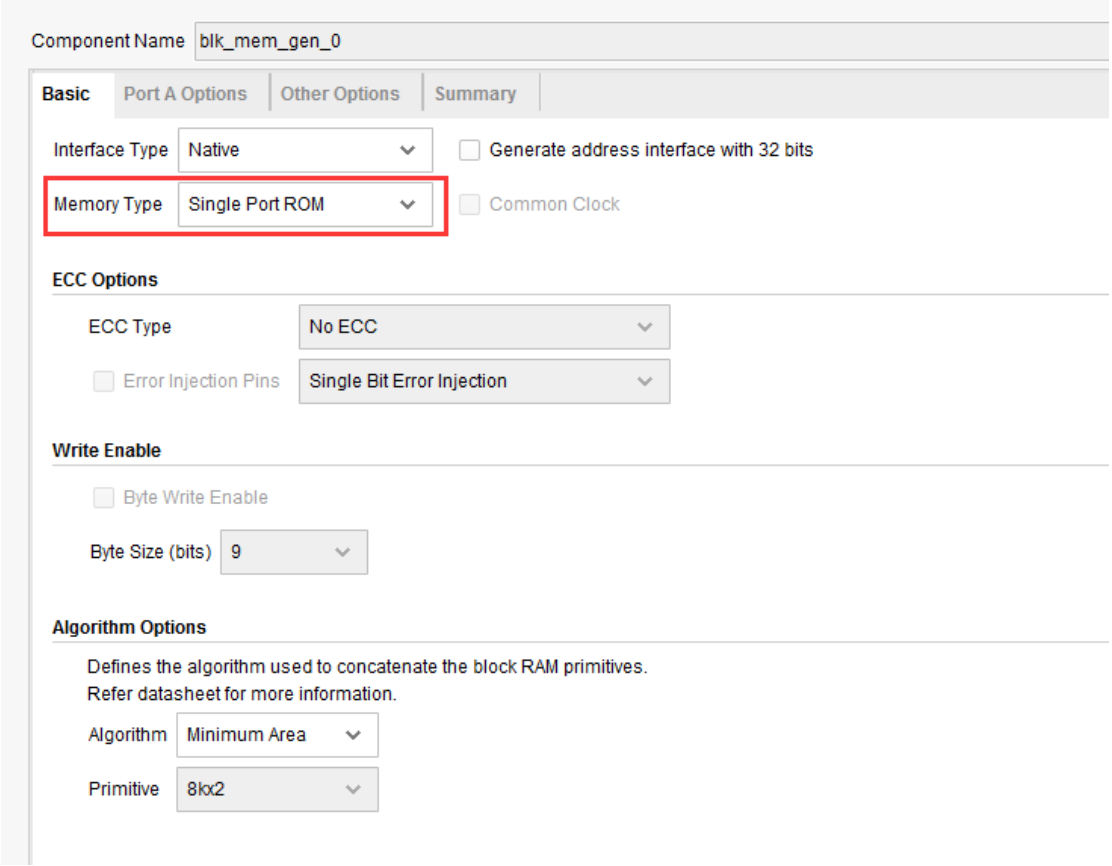
3. 添加 ROM IP

点击 Flow Navigator -> PROJECT MANAGER -> IP Catalog ->Memories & Storage Elements ->RAMs & ROMs & BROAM

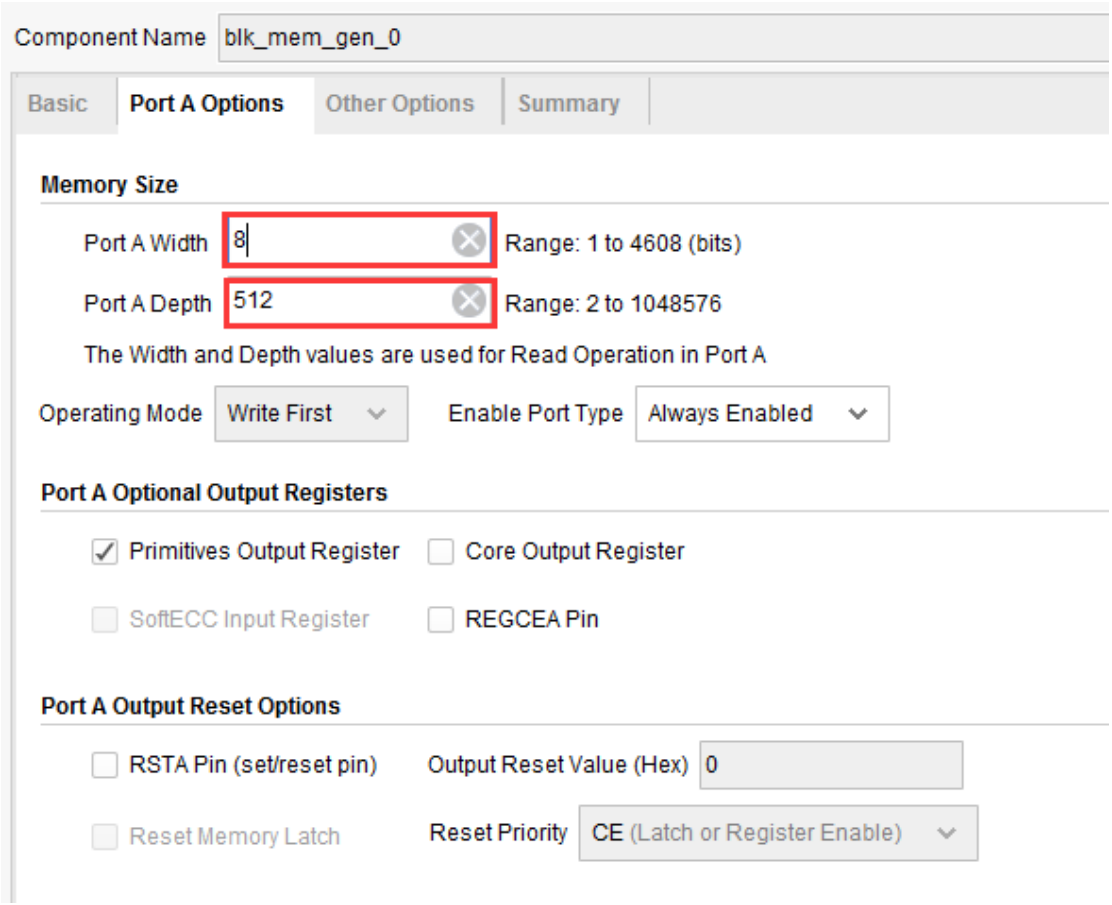




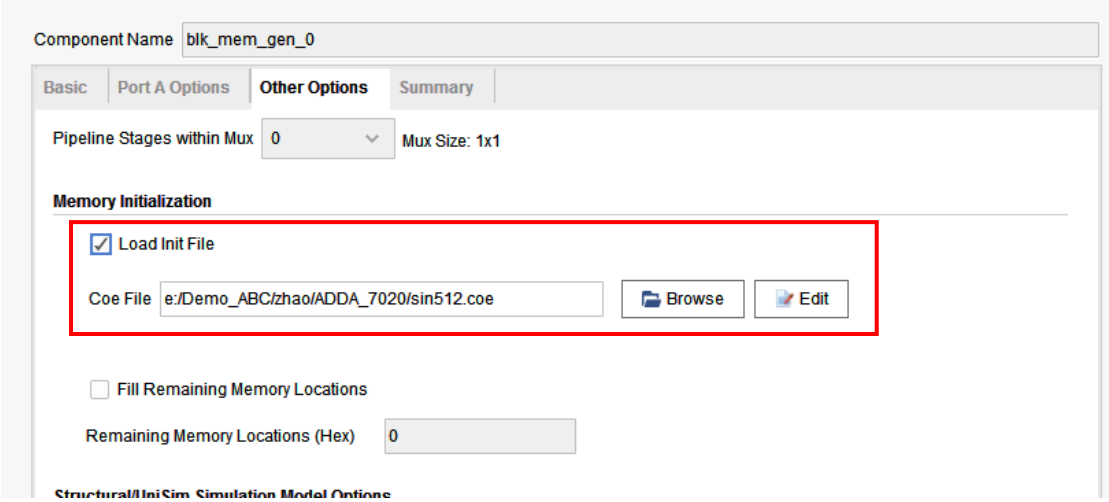
Memory Type 选择 Single Port ROM



Port A Width 选择 8, Port A Depth 选择 512



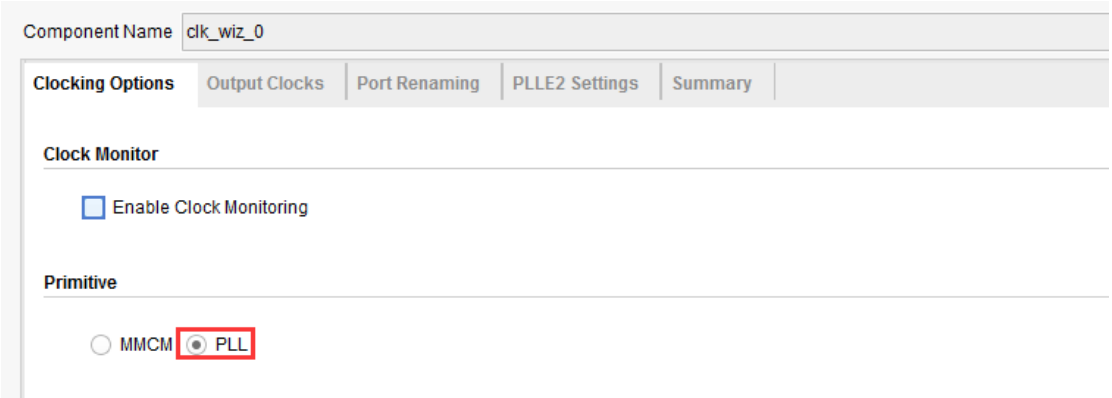
Coe File 选择 步骤 1 制作的 coe 文件



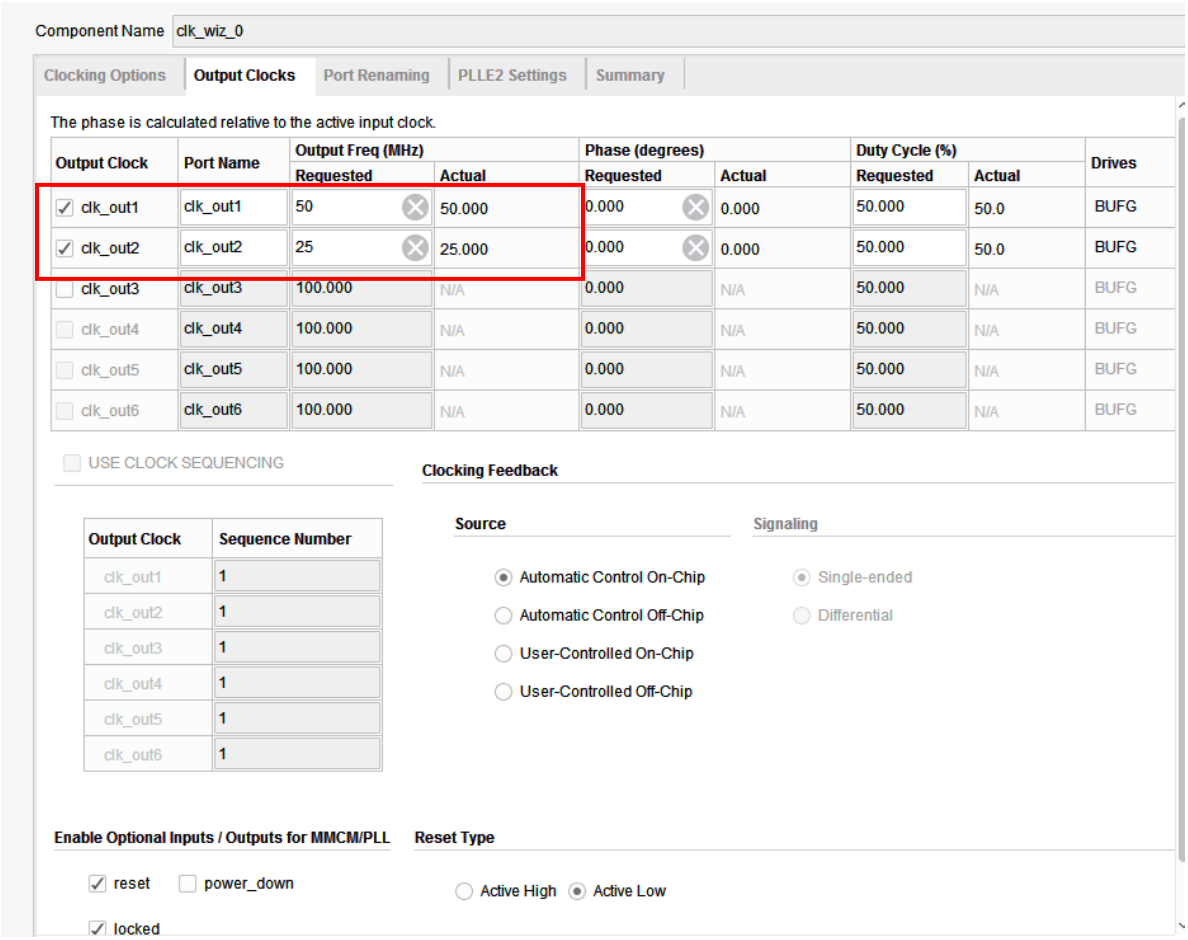
4. 添加 PLL IP

点击 Flow Navigator -> PROJECT MANAGER -> IP Catalog ->FPGA Features and Design -> Clocking -> Clocking Wizard，将出现图窗口，并作如下设置

Primitive 选择 PLL

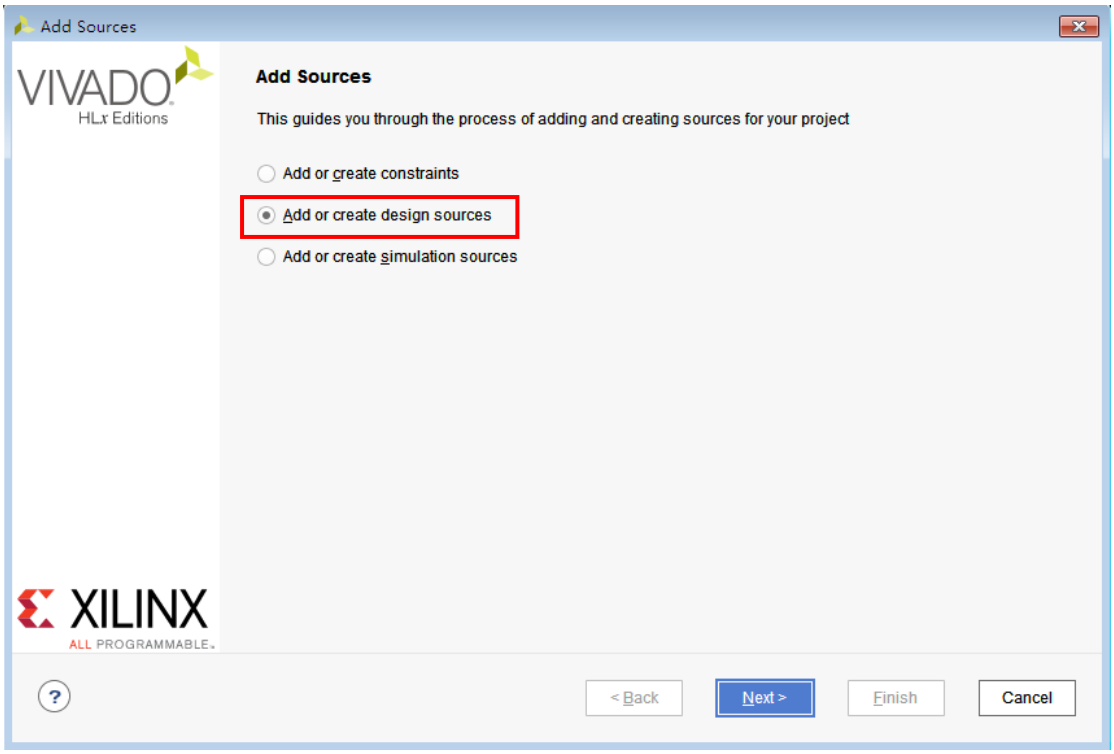


Output Clock 选择 clk_out1 50MHZ clk_out2 25MHZ, Reset Type 选择 Active Low

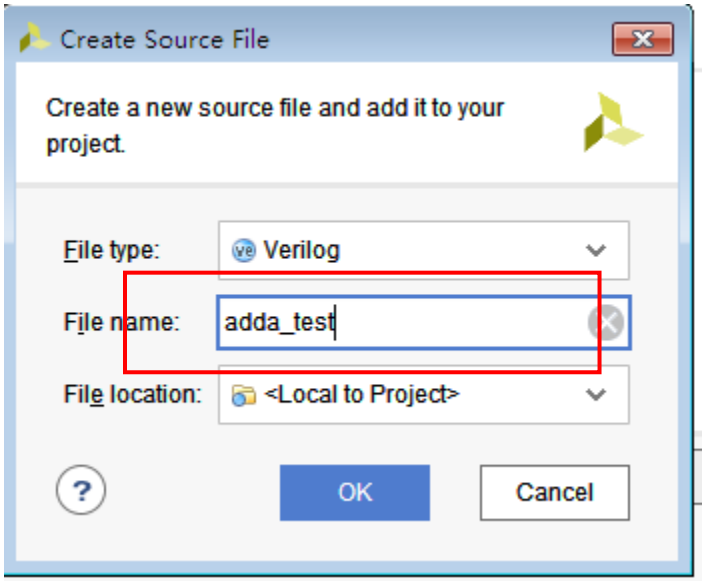


5. 编写 ADDA_7020 的顶层 verilog 程序

左击 SOURCES-> Design Source ->Add Source



点击 Create File,文件名填 adda_test,



adda_test.v 程序如下：

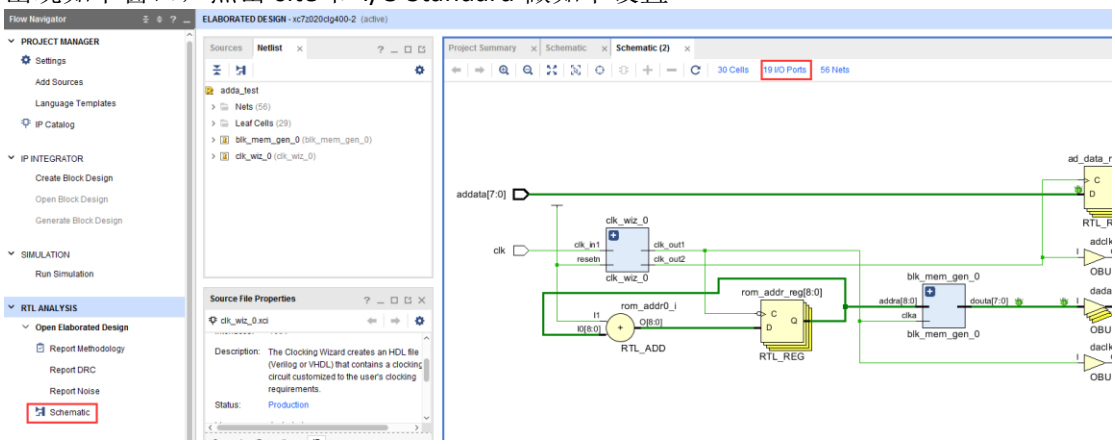
```
`timescale 1ns / 1ps
module adda_test(
```

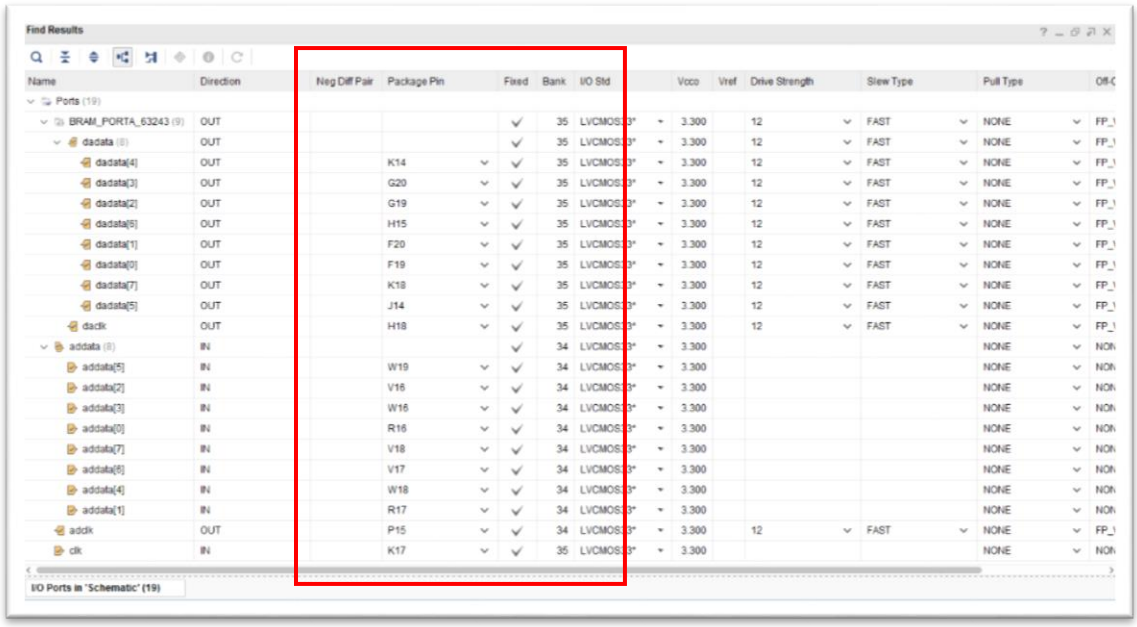
```
    input clk,          //fpga clock

    output daclk,
     (* mark_debug = "TRUE"*)
    output [7:0] dadata,      //DA data

    output adclk,
     (* mark_debug = "TRUE"*)
     input  [7:0] addata       //AD data

   );
reg [8:0] rom_addr;

(*DONT_TOUCH="{TRUE|FALSE}"*)reg [7:0] ad_data;

wire [7:0] rom_data;
wire clk_25;
wire clk_50;

assign dadata=rom_data;
assign daclk=clk_50;
assign adclk=clk_25;

//DA output sin waveform
always @(negedge clk_50)
begin
    rom_addr <= rom_addr + 1'b1 ;
end

always @(posedge clk_25)
begin
     ad_data <= addata ;
end


blk_mem_gen_0 blk_mem_gen_0 (
  .clka(clk_50), // input clka
  .addra(rom_addr), // input [8 : 0] addra
  .douta(rom_data) // output [7 : 0] douta
);


clk_wiz_0 clk_wiz_0
  (// Clock in ports
   .clk_in1(clk),    //IN
   // Clock out ports
   .clk_out1(clk_50),     //OUT
```

```
    .clk_out2(clk_25),    // OUT
    // Status and control signals
    .resetn(1'b1),// IN
    .locked());     // OUT

Endmodule
```

6. 添加 I/O 约束

点击左侧边栏 RTL ANALYSIS 下的 Open elaborated design
在右侧的 Schematic 窗口下点击 I/O Ports
出现如下窗口，点击 site 和 I/O Standard 做如下设置



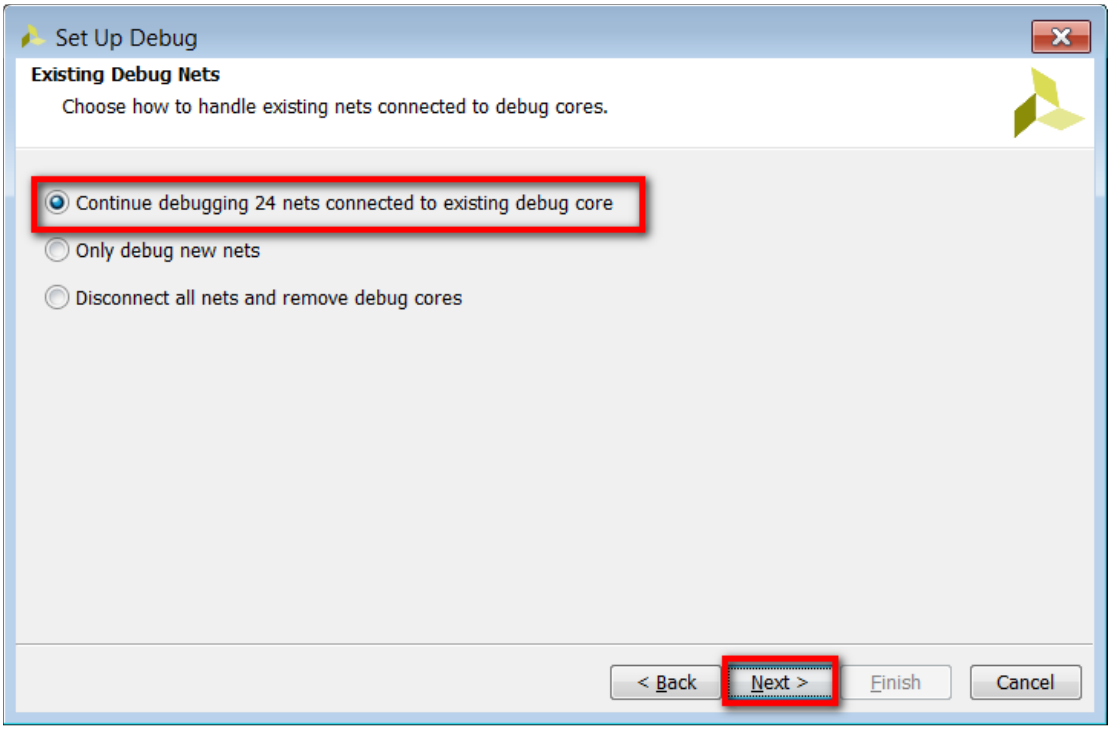


添加完成后保存，添加 XDC 文件命名为 ADDA_7020。

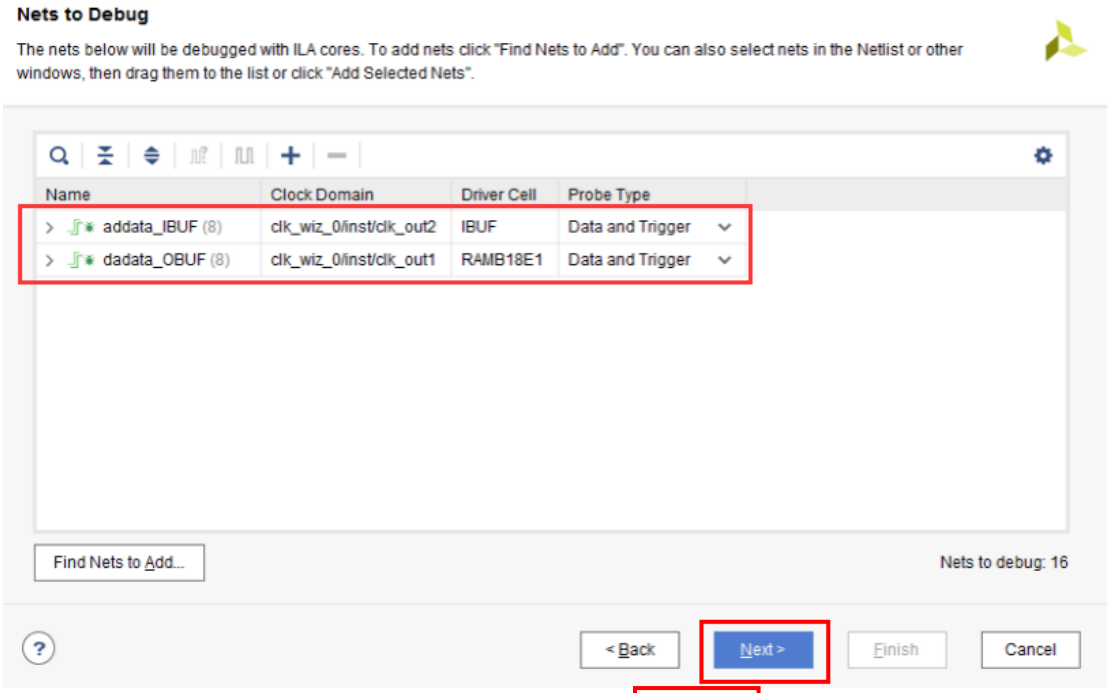
7. 设置 Debug

在左侧栏 synthesis 下 点击 Run Synthesis，
Synthesis Completed 弹窗口 点击 Cancel，
左侧边栏展开 Open Synthesized Design Set Up Debug，
如下图，点击 Next

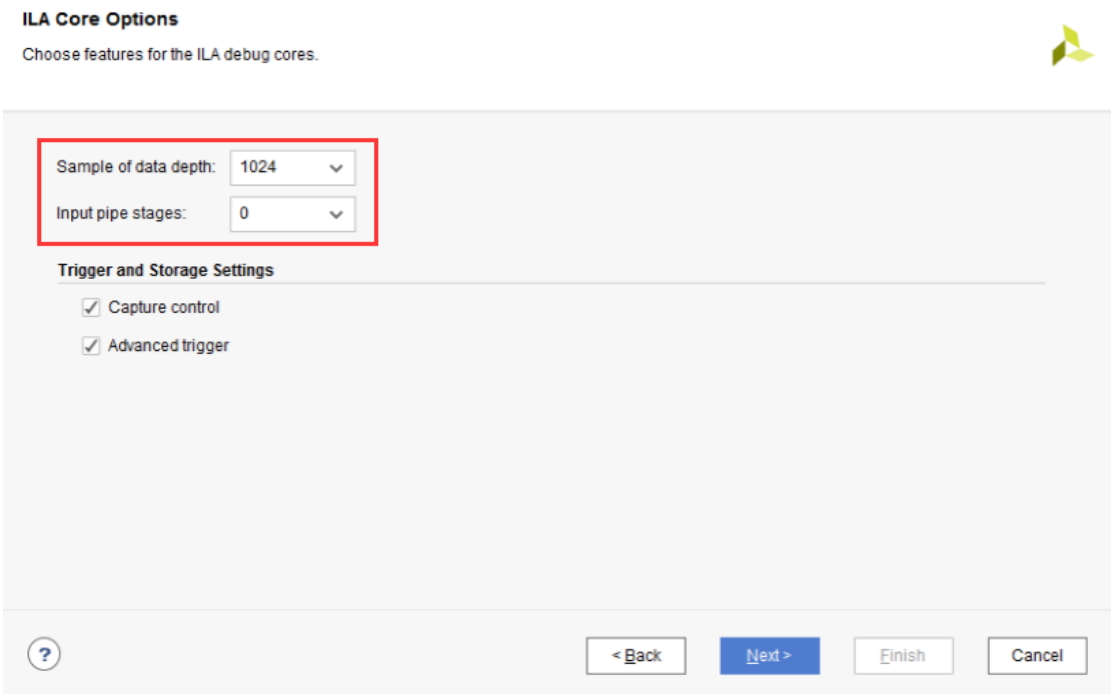
如下选择 Continue debugging， 点击 Next，



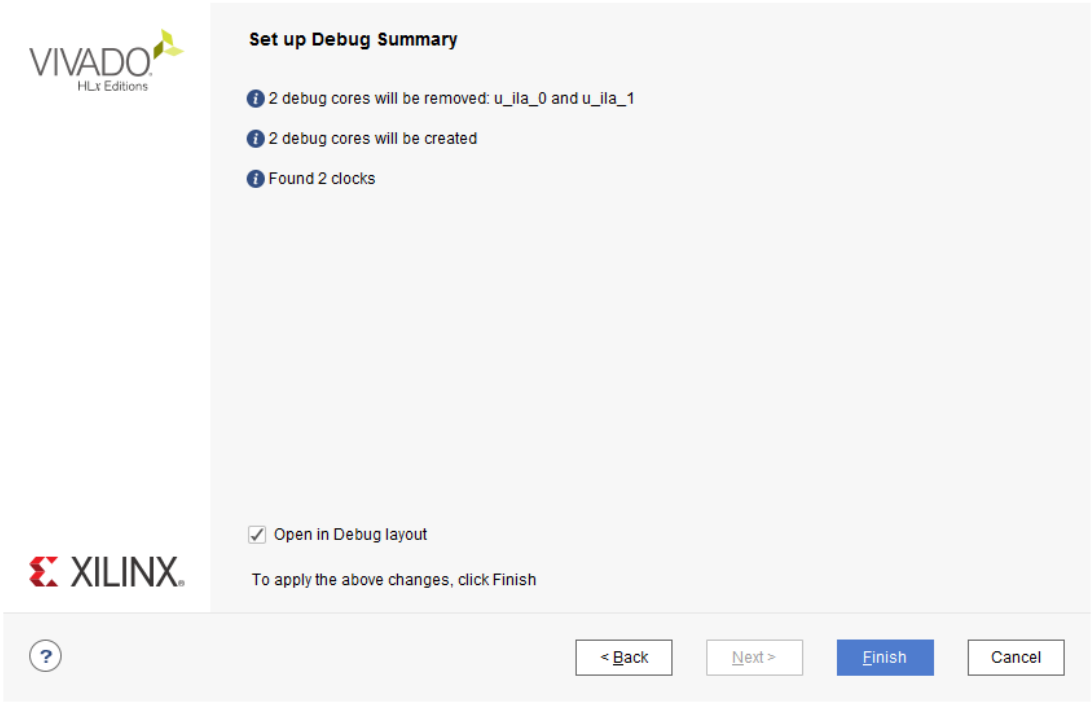
如下勾选，点击 Next，



可以看到，我们在源代码中 Mark 的信号，都出现在子这里。点击 next



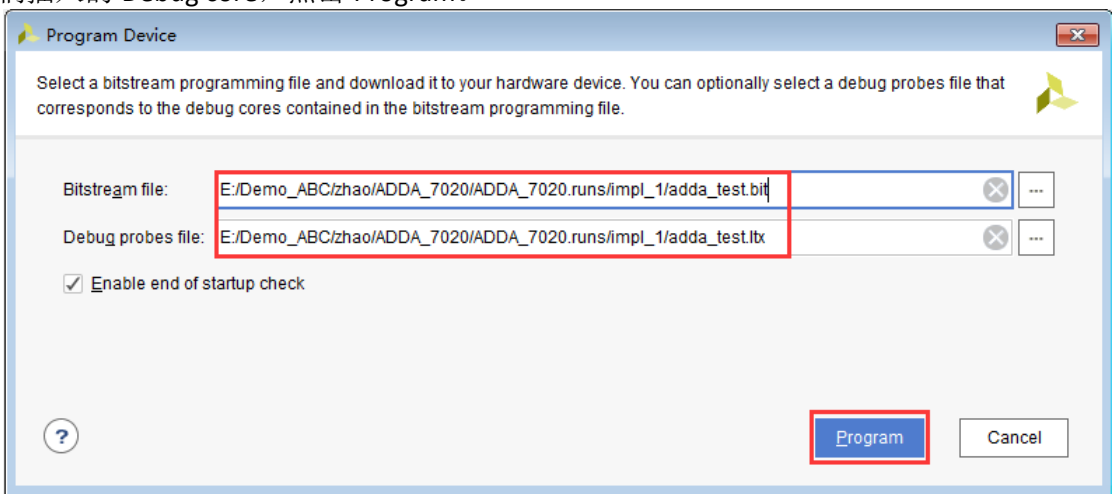
点击 Next



点击完成。在左侧边栏 Program and Debug 下点击 Generate Bitstream， Bitstream Generation Completed 弹窗点击 OK。

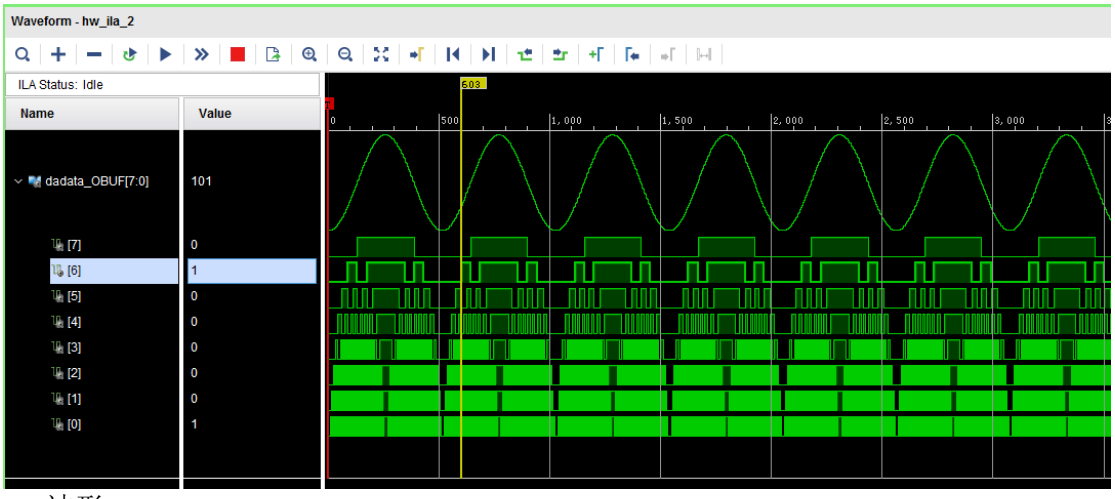
8. 调试

将 ADDA 子卡连接到开发板上 JM2 上，连接电脑 USB 到开发板 JTAG 口，确保开发板拨码开关 1，2 都在 ON 的位置，给开发板上电，回到左侧边栏，点击 Open Hardware Manager，在 Hardware Manager 窗口点击 Auto Connect 图标，自动检测到 Device 后，右键单击 xc7z020 选择 Program Device，出现如下弹窗，可见除了 Bitstream 外，还有一个 Debug probes 文件，这就是我们插入的 Debug core，点击 Program。

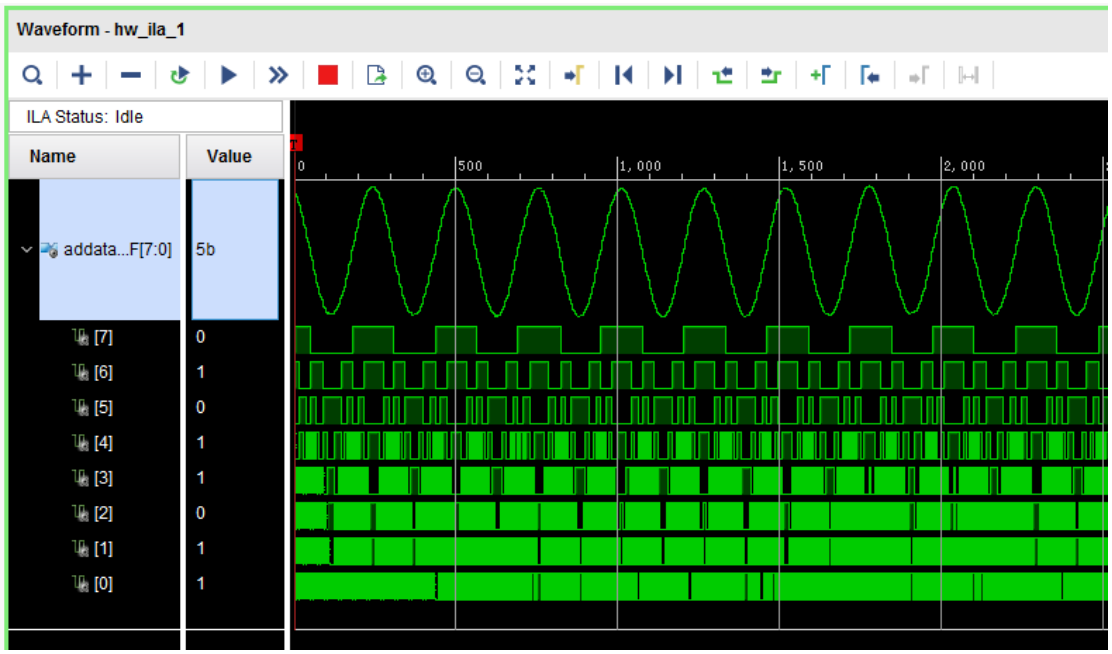


点击 program　出现如下窗口

点击 hw_ila_2 选择 DA 波形，点击 hw_ila_1 选择 AD 波形。
DA 波形，在出现如下窗口击运行箭头，将出



AD 波形


Waveform - hw_ila_1
ILA Status: Idle
Name
Value
addata...F[7:0]
5b
[7]
0
[6]
1
[5]
0
[4]
1
[3]
1
[2]
0
[1]
1
[0]
1
0
500
1,000
1,500
2,000