实验三、数字FIR滤波器设计与实现

1. 实验目的
2. 学习Verilog基本语法；
3. 巩固VIVADO环境下Verilog编程设计的基础；
4. 掌握Debug使用方法；
5. 掌握数字FIR滤波器实现过程及使用Verilog语言实现FIR滤波器设计；
6. 掌握用MATLAB工具fdatool获取滤波器系数。
7. 实验内容

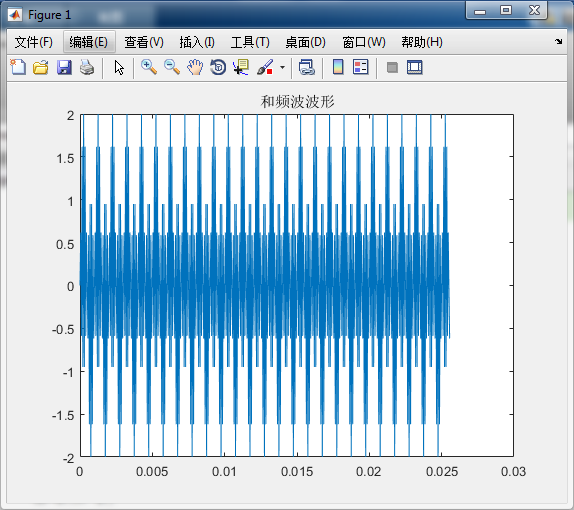
使用Verilog语言，实现数字FIR滤波器设计与实现，在开发板上验证。

三、实验要求

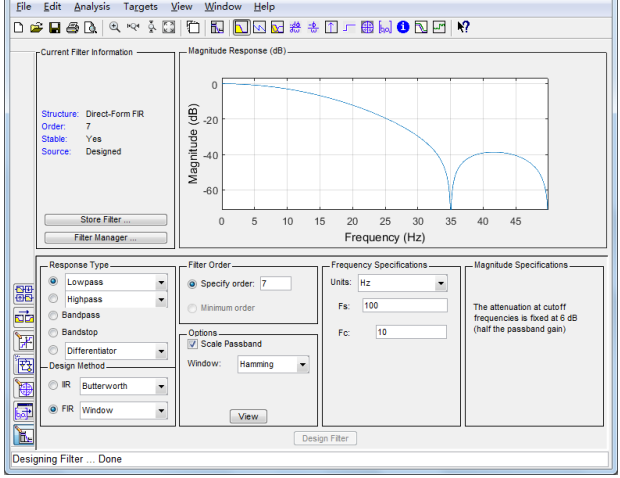
1. 通过MATLAB产生混合信号，并通过fdatool工具设计滤波器进行滤波仿真；
2. 将产生的信号量化，并保存为coe文件，将设计的滤波器系数保存为coe文件；
3. 通过Verilog实现FIR滤波器设计并对产生的信号滤波；
4. 上板验证，对比与MATLAB滤波的效果。

四、MATLAB实现过程

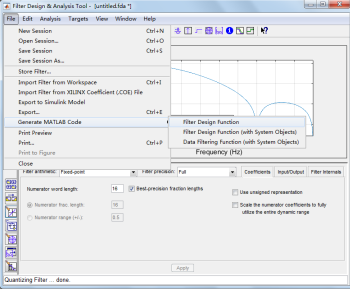
1. 打开 MATLAB 软件，产生混合信号。

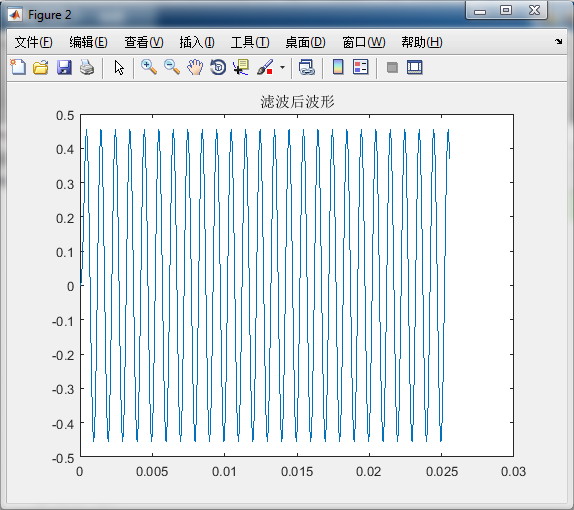


2.在命令行窗口输入fdatool进入滤波器设计界面，滤波器阶数和采样频率，通带频率按各自信号选取，其余按照此界面设置。

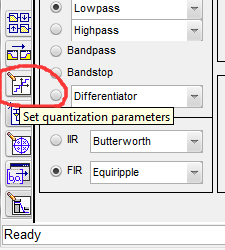


3.设计好后，点击File-Generate MATLAB code-Filter Design function,生成文件，并对混合信号滤波看效果。

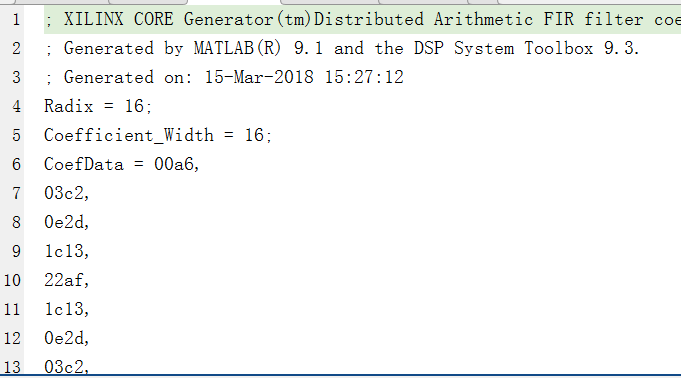




1. 点击工具栏红色按钮进行系数量化，量化位宽自己设定。



6．量化完后，点击Targets – XILINX (COE) File生成滤波器coe文件，并查看。



7.最后如果滤波器符合要求，对混合信号进行量化并保存为coe文件。实现过程如下：

width=16;

y2=round(y.\*2^(width-3));%量化Q13

fig=fopen('signal.coe','w');

fprintf(fig,'memory\_initialization\_radix=10;\n memory\_initialization\_vector = \n');

for i = 1:N %将数据转换为补码表示

if y2(i) >= 0

y2(i) = y2(i);

else

y2(i) = 2^width + y2(i);%负整数补码表示（相当于对数据位宽n做2^(n+1)做补）

end %大于2^15-1=32767的是负数；小于32767的是正数

end

for i=1:N

if i==N

fprintf(fig,'%d,',y2(i));

else

fprintf(fig,'%d,\r\n',y2(i));

end

end

fclose(fig);

五、FPGA实现过程

1.建立新工程后，添加ROM IP核保存MATLAB生成的混合信号。

2.进行滤波器实现设计。

3.将混合信号滤波输出。

4.建立仿真文件，仿真观察是否与MATLAB滤波效果一致。

实现代码顶层文件如下：

module top\_fir(

input clk,

input rst,

output [15:0] fir\_out

);

wire [15:0] spo;

wire [8:0] address;

wire clk\_20k;

dist\_mem\_gen\_0 your\_instance\_name (

.a(address), // input wire [8 : 0] a

.spo(spo) // output wire [15 : 0] spo

);

address address\_ins(

.clk\_20k(clk\_20k),

.rst(rst),

.address(address)

);

clk\_20 clk\_20\_ins(

.clk(clk),

.rst(rst),

.clk\_20k(clk\_20k)

);

filter filter\_ins(

.clk(clk),

.clk\_20k(clk\_20k),

.rst(rst),

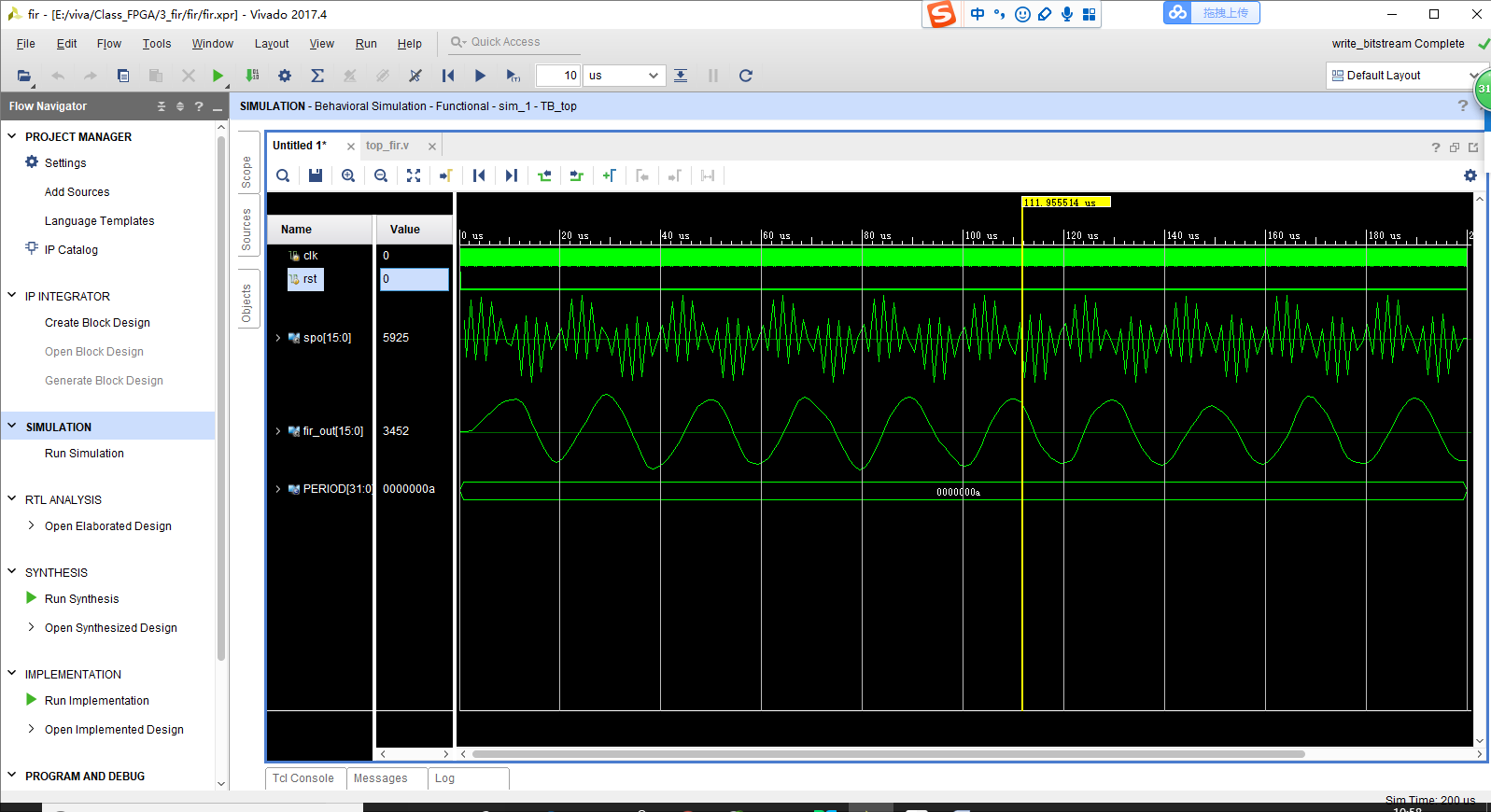
.fir\_in(spo),

.fir\_out(fir\_out)

);

Endmodule

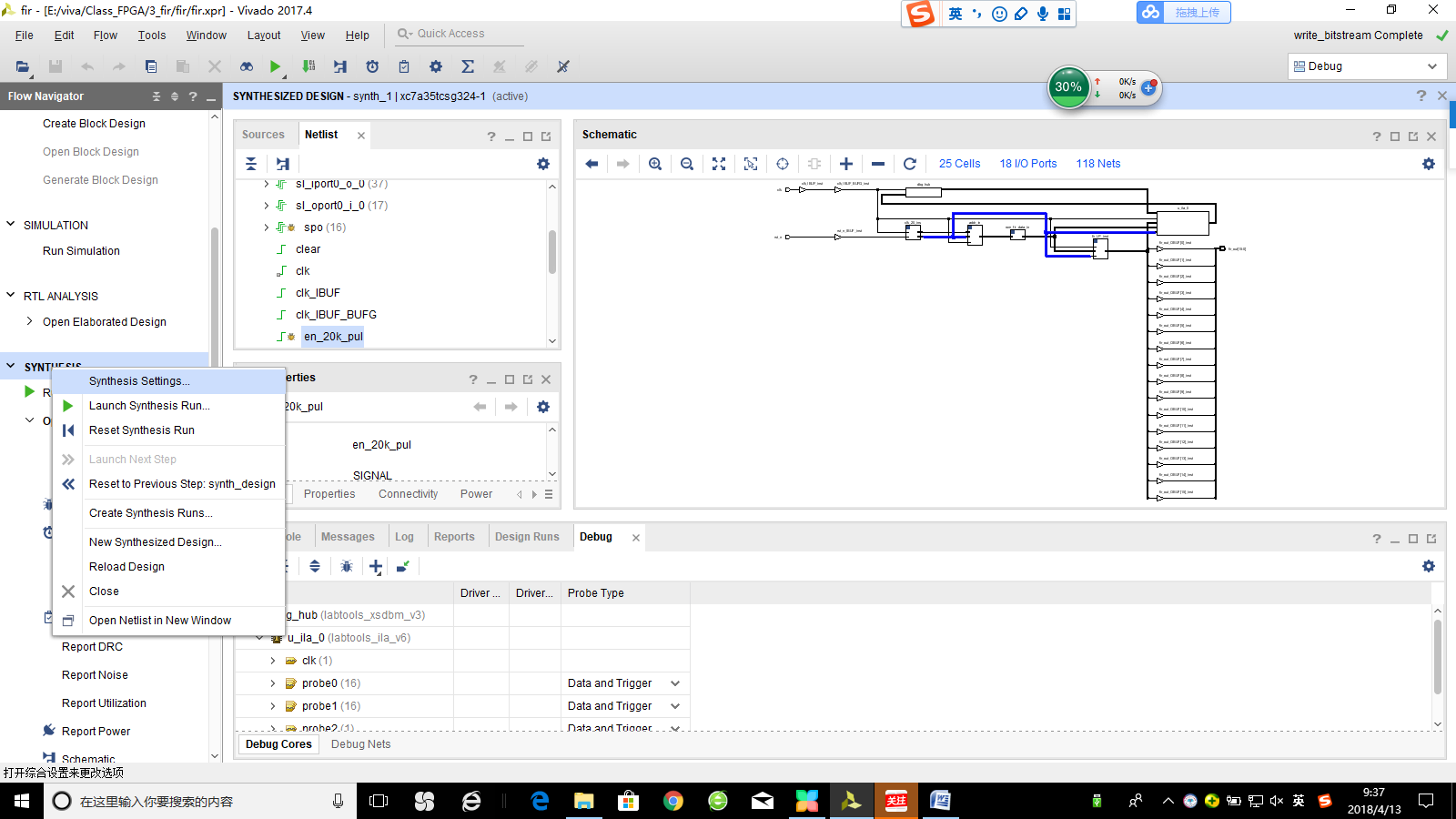
仿真效果如下：

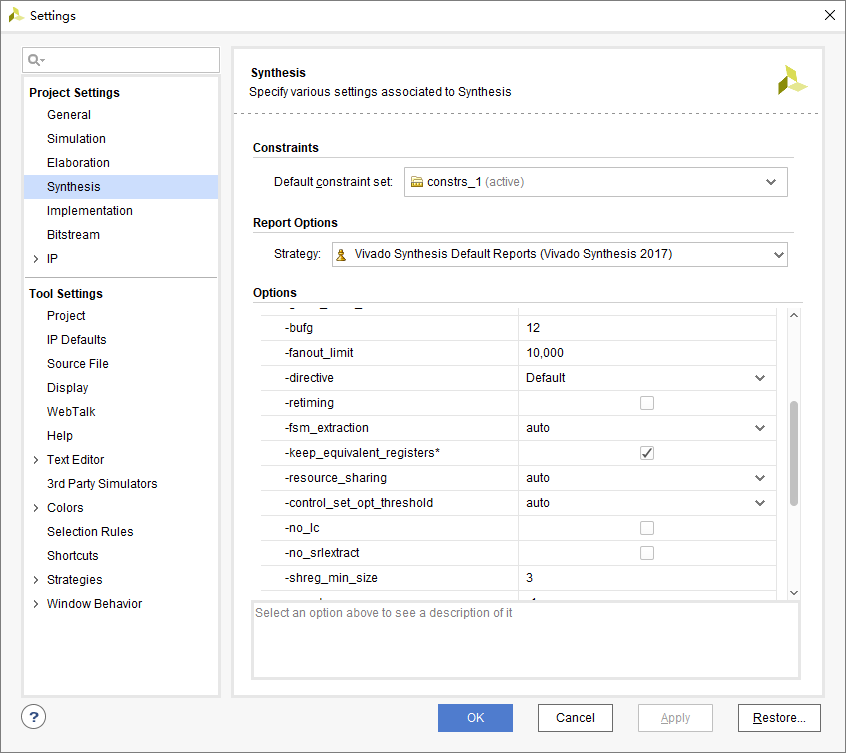


滤波后信号周期为1ms,与MATLAB仿真一致。

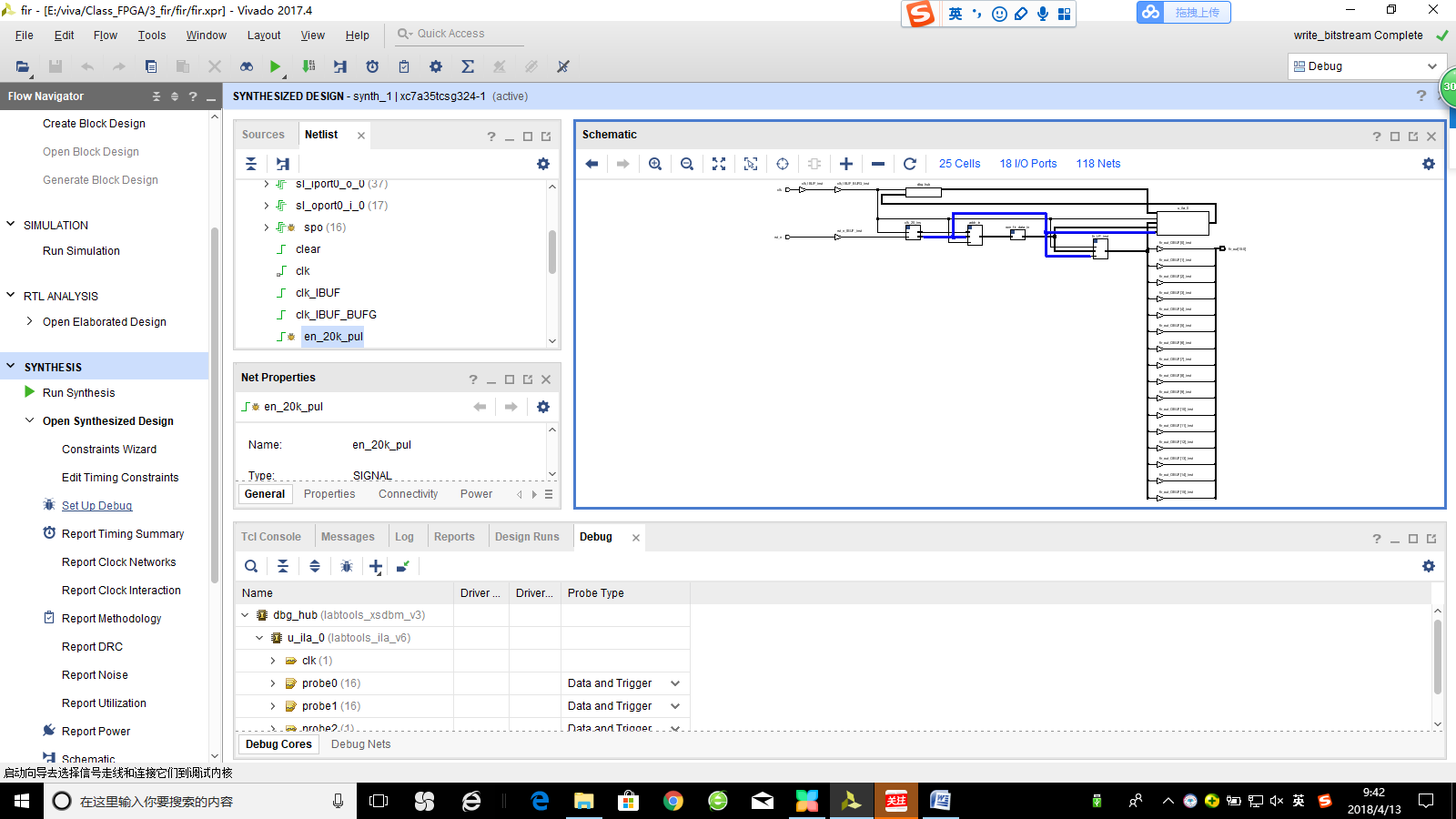
六、Debug调试方法

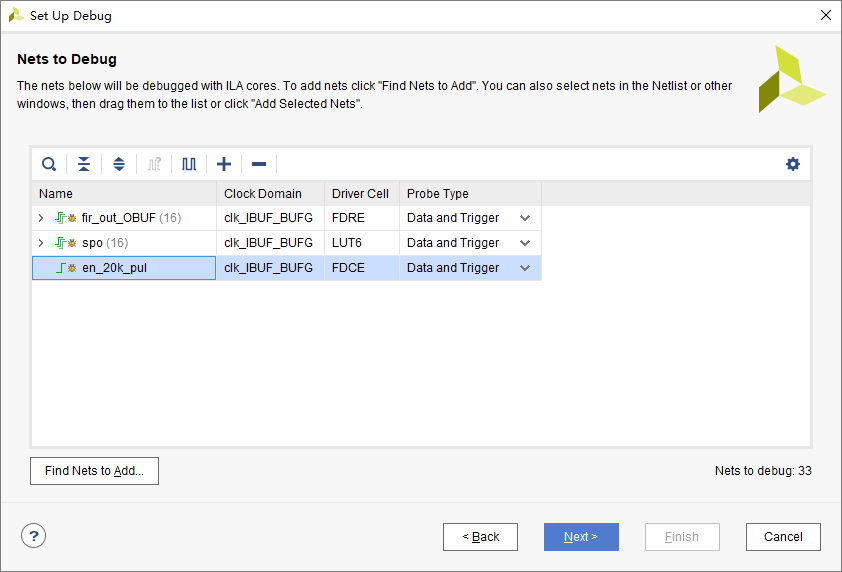
1、设置综合选项





2、set\_up debug



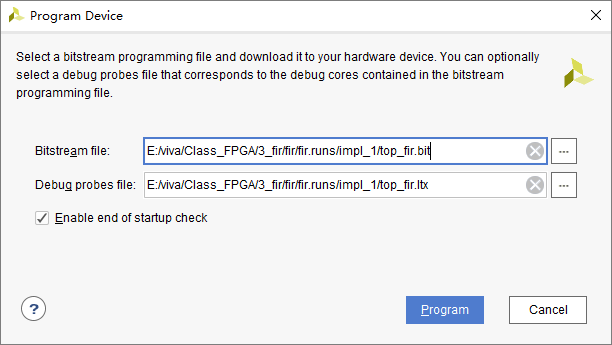


添加滤波器输入信号（spo），输入使能信号（en\_20k\_pul）和输出信号。

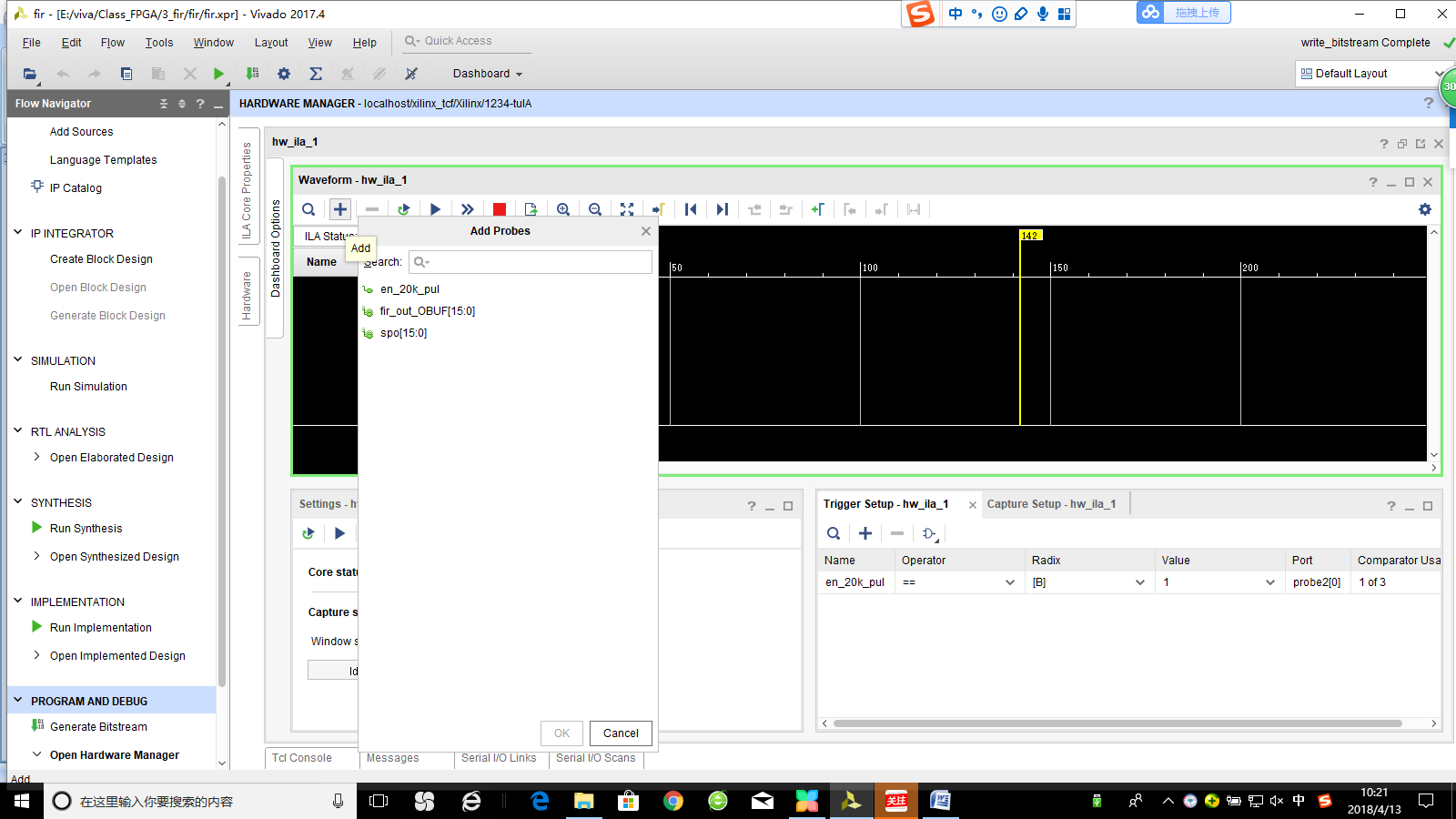
保存并重新编译工程，生成bit文件。

3、Debug 调试

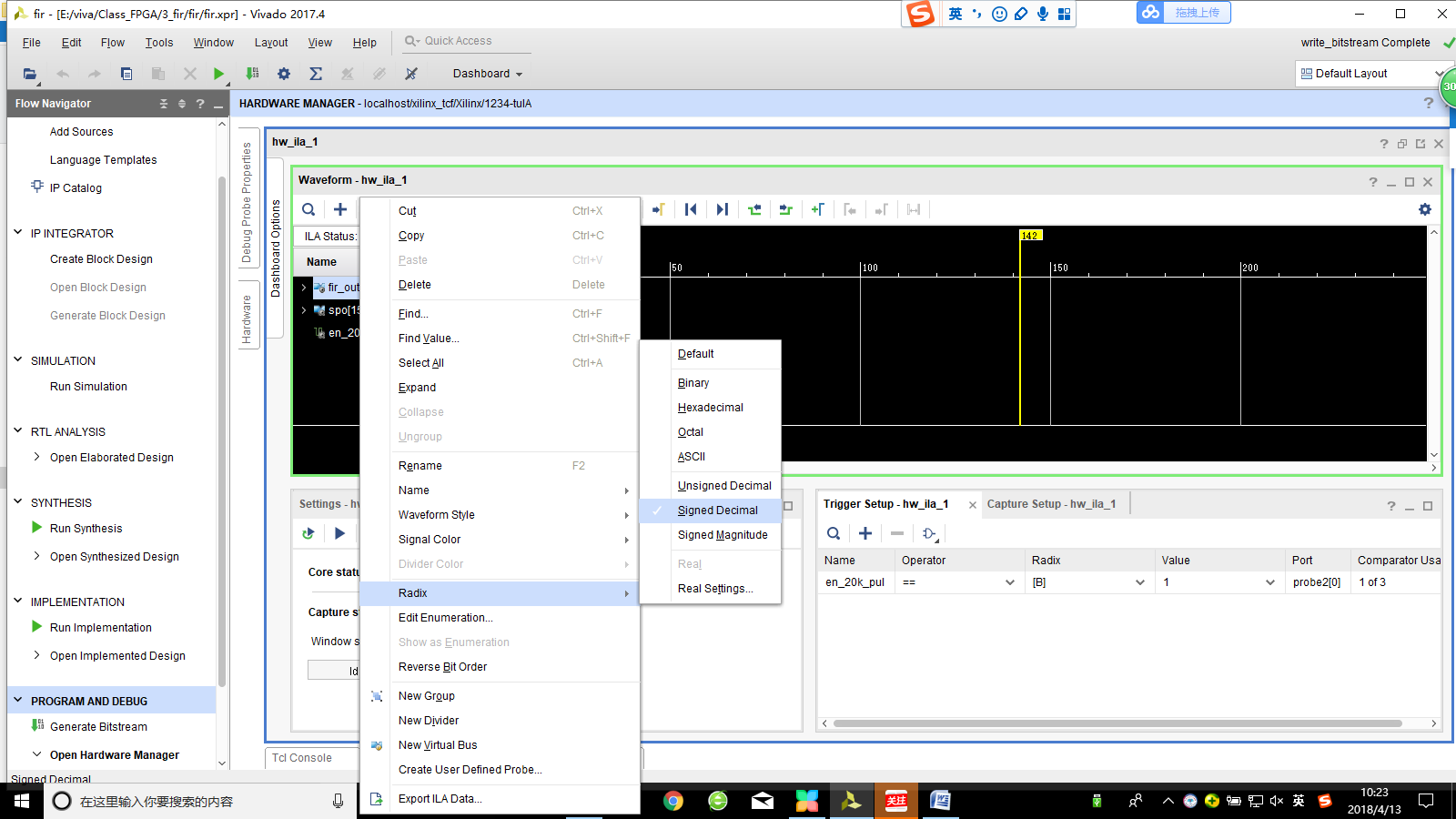
下载bit文件



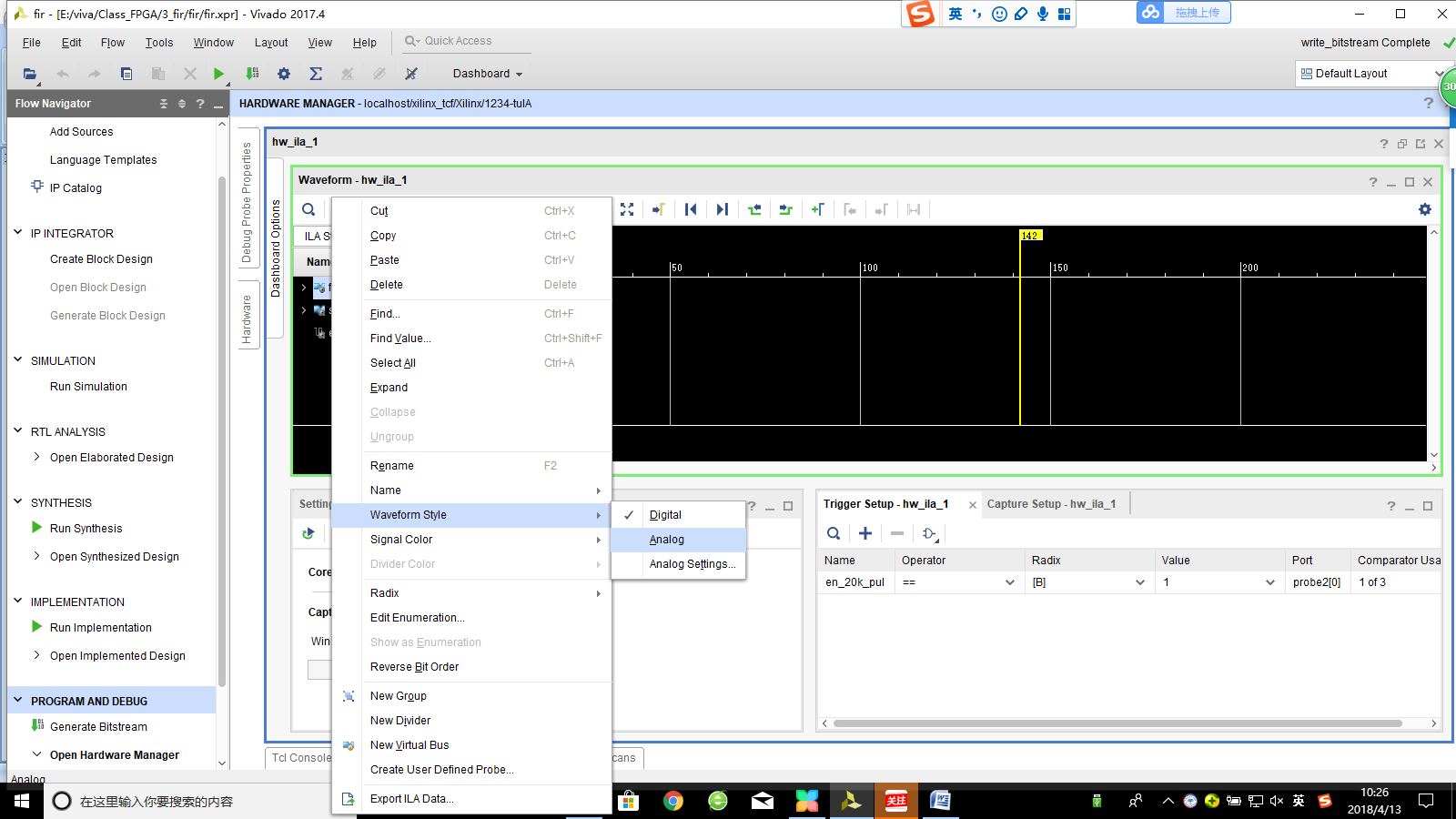
添加信号



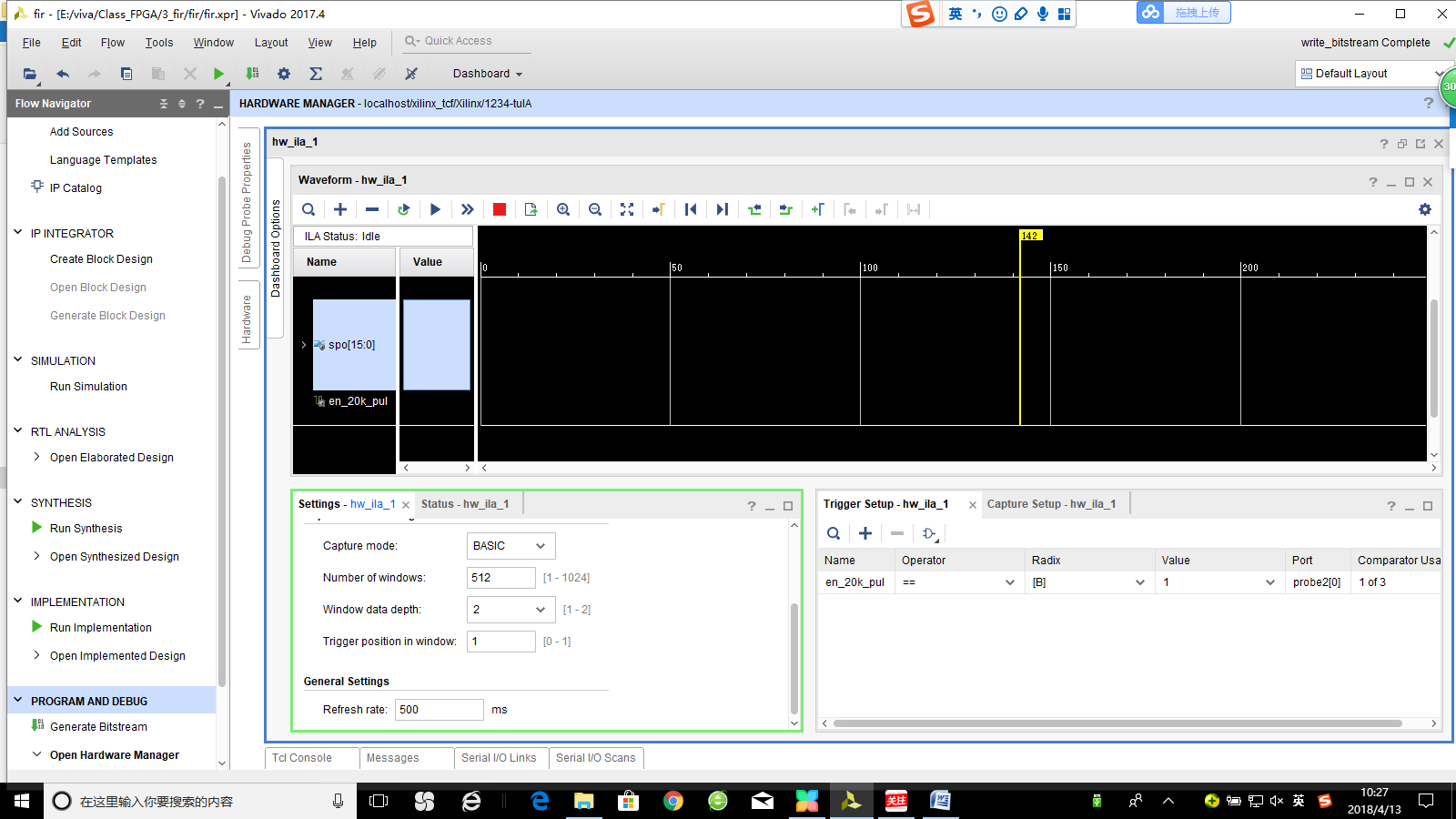
将滤波器的输入和输出设置为有符号的十进制



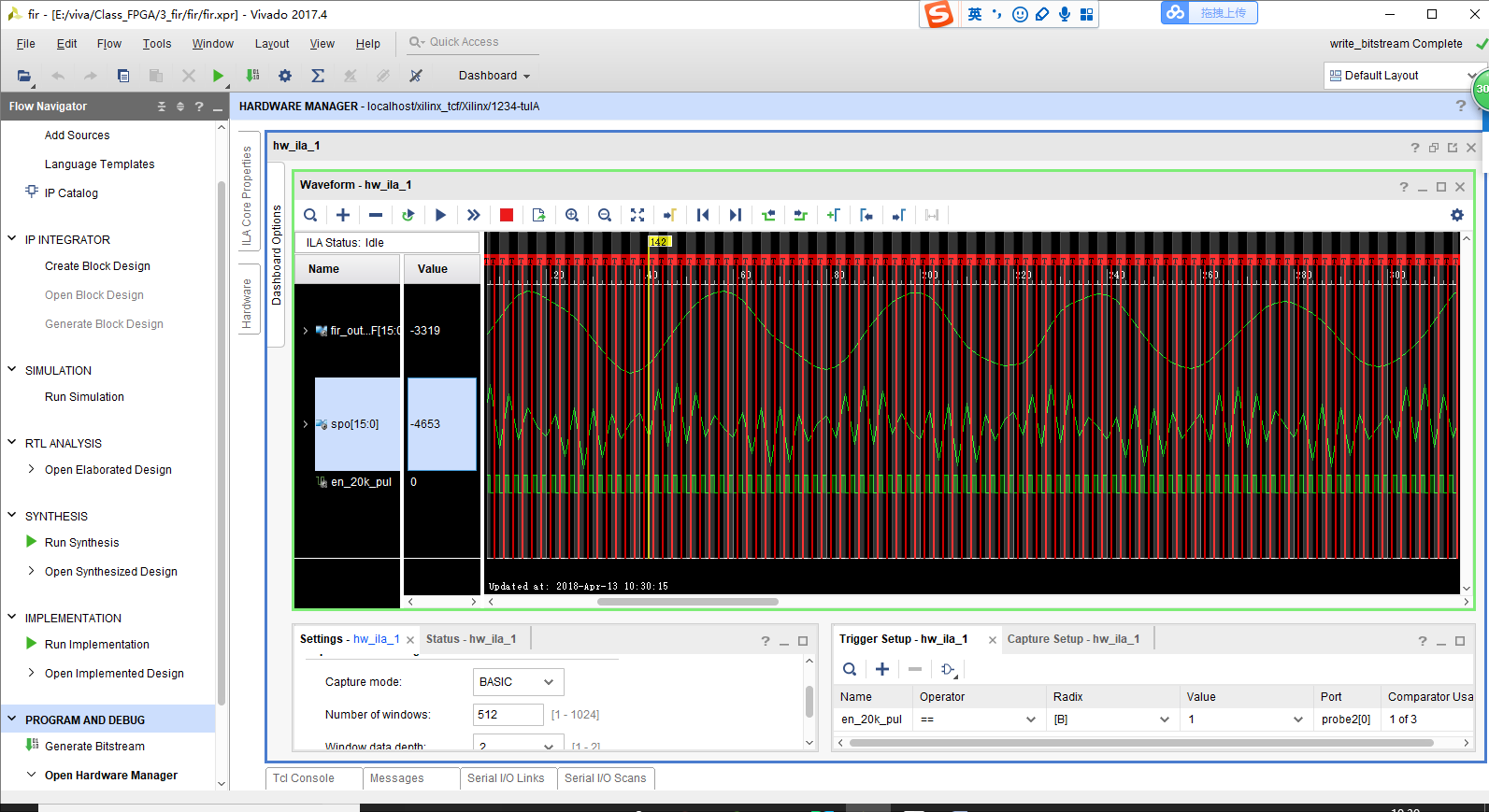
将滤波器的输入和输出设置为模拟显示



设置触发方式和条件



Debug 抓取信号



和仿真的波形对比。