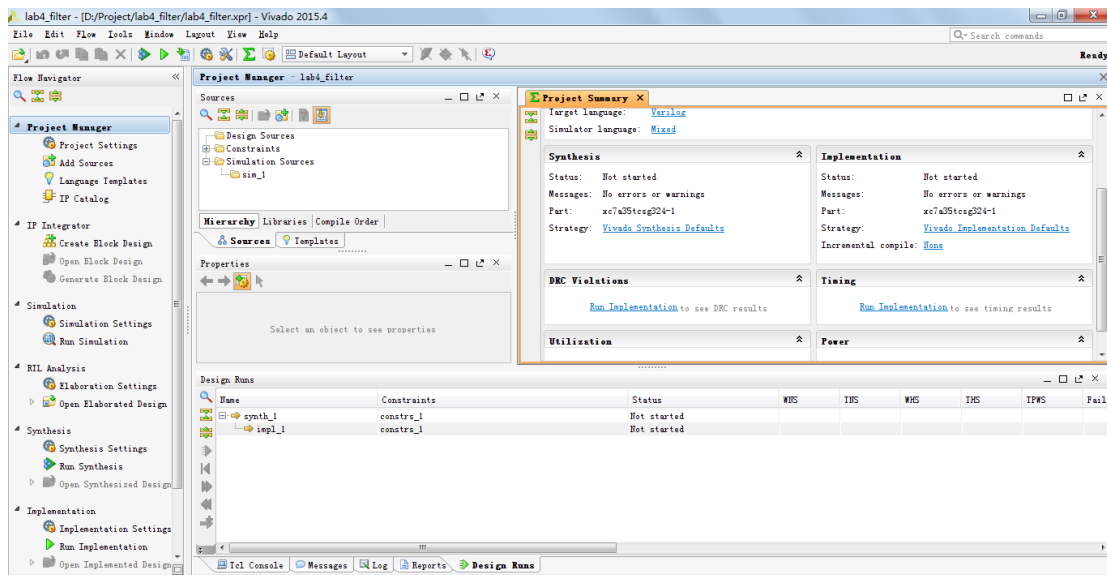


## 实验四、FIR 数字滤波器设计

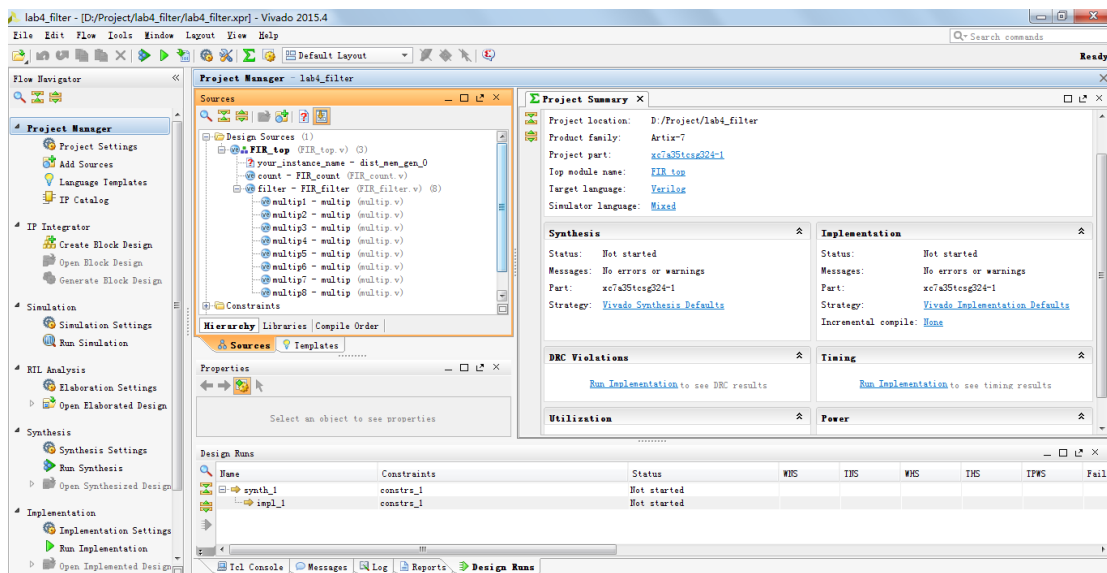
### 一、创建工程

1、仿照实验一的流程，创建一个新的工程“lab4\_filter”。



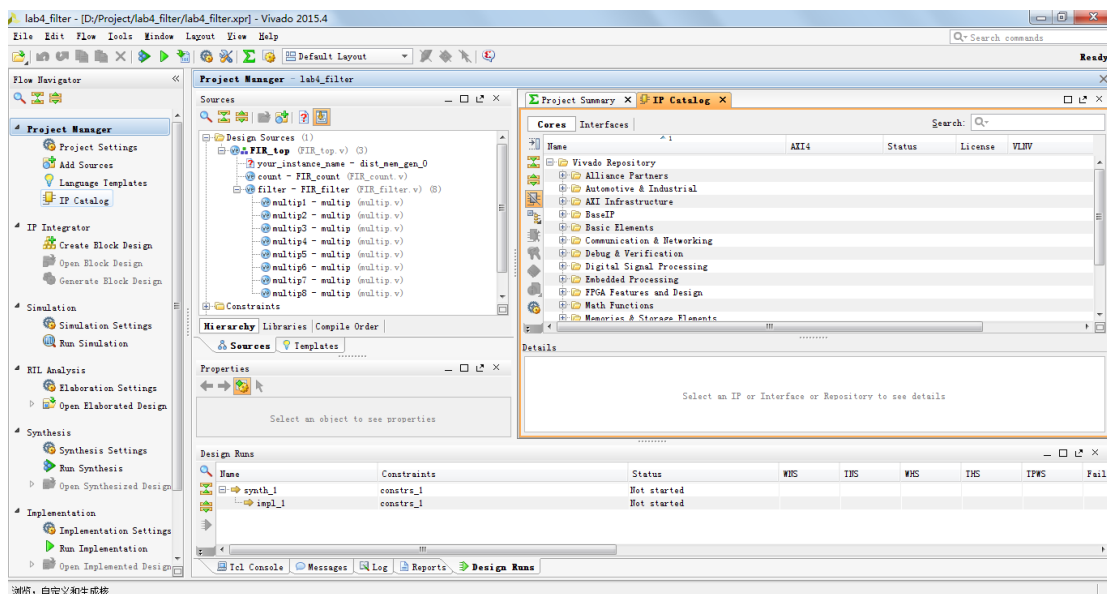
### 二、添加源文件

2、仿照实验二的流程，添加各模块的源文件（顶层模块 FIR\_top.v、计数模块 FIR\_count.v、滤波模块 FIR\_filter.v）。

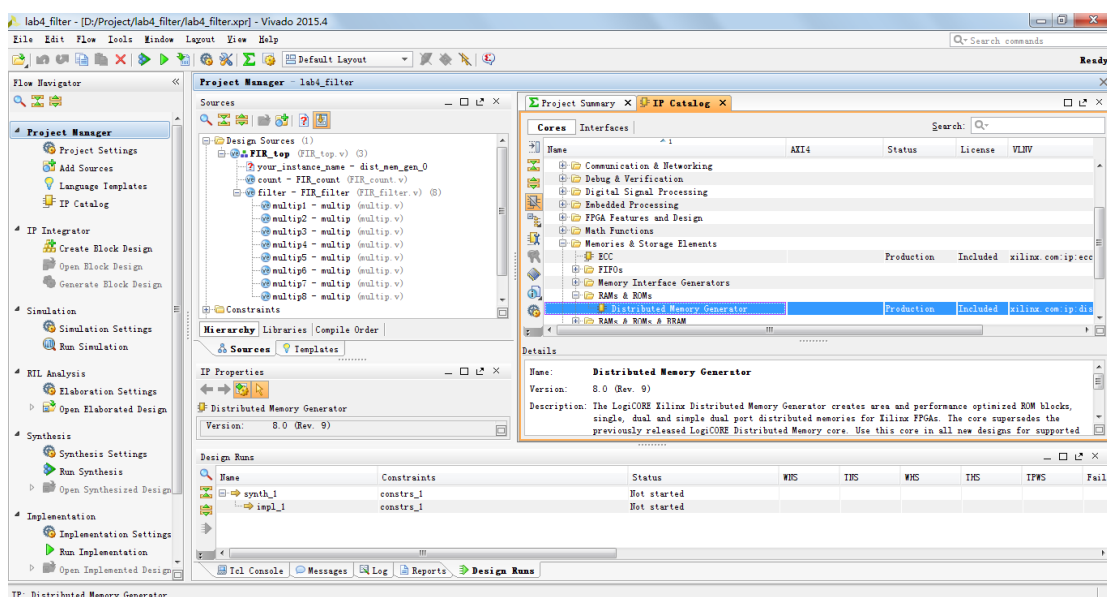


3、通过 IP 核调用添加 ROM 模块（Dist\_men\_gen\_0.v）

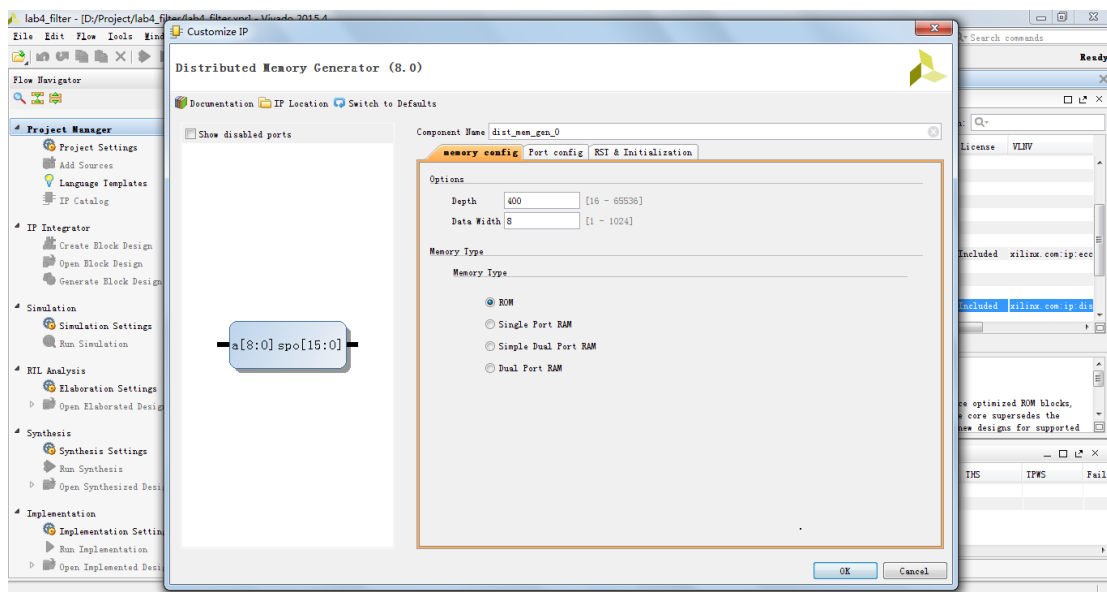
3.1、在“Flow Navigator”一栏中点击“Project Manager”下的“IP Catlog”。



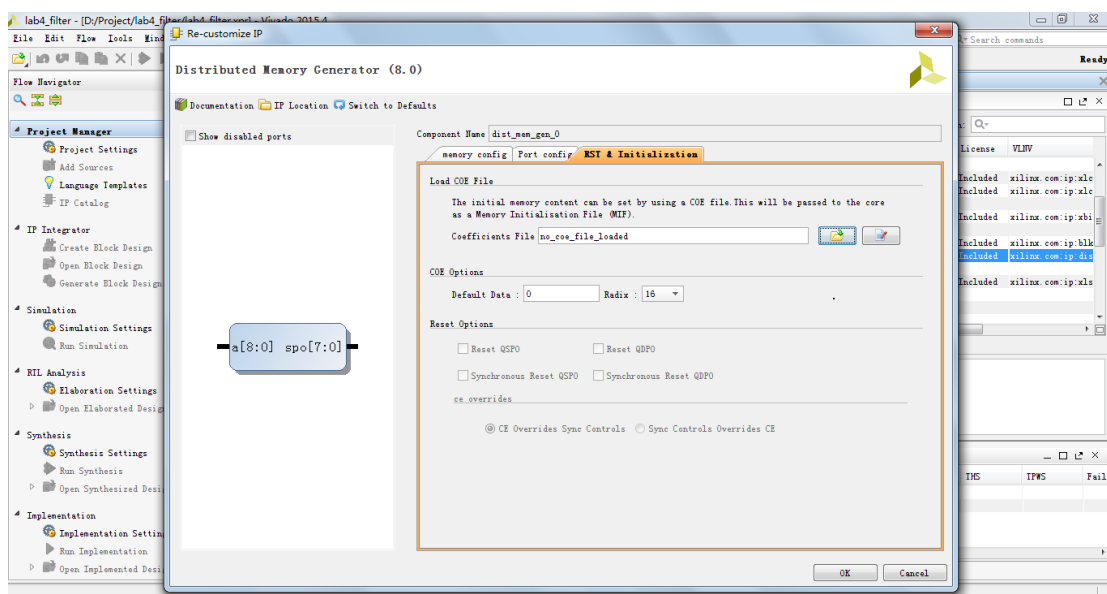
3.2、在“IP Catalog”窗口中，找到“Memories & Storage Elements/ RAMs & ROMs/ Distributed Memory Generator”，双击打开。

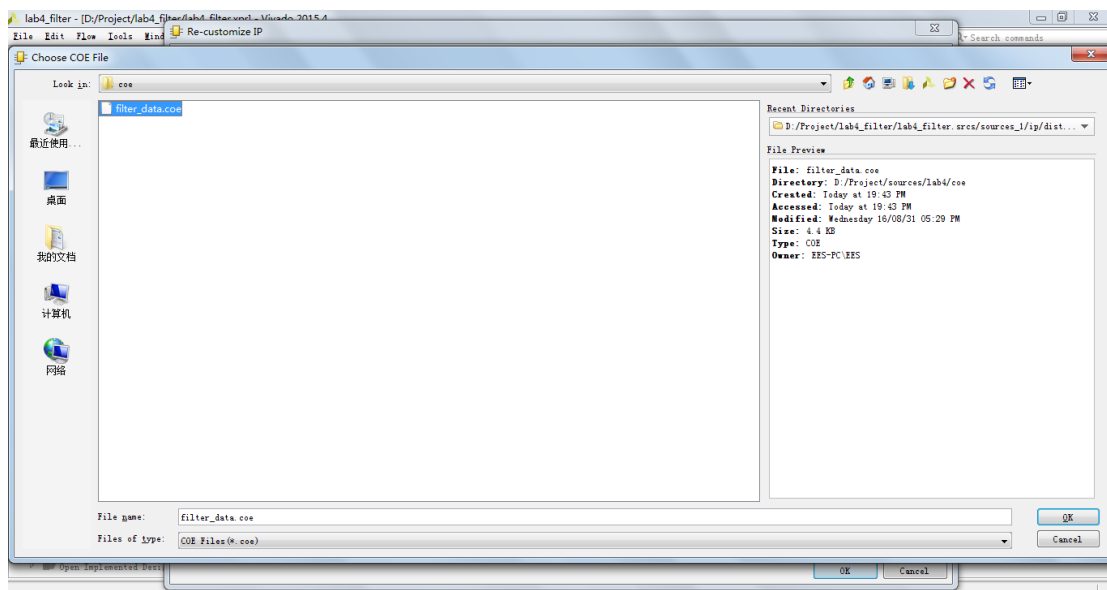


3.3、在“Customize IP”界面的“memory config”标签页下，“Memory Type”选择“ROM”，“Options”中的“Depth”一栏输入400，“Data Width”一栏输入8。

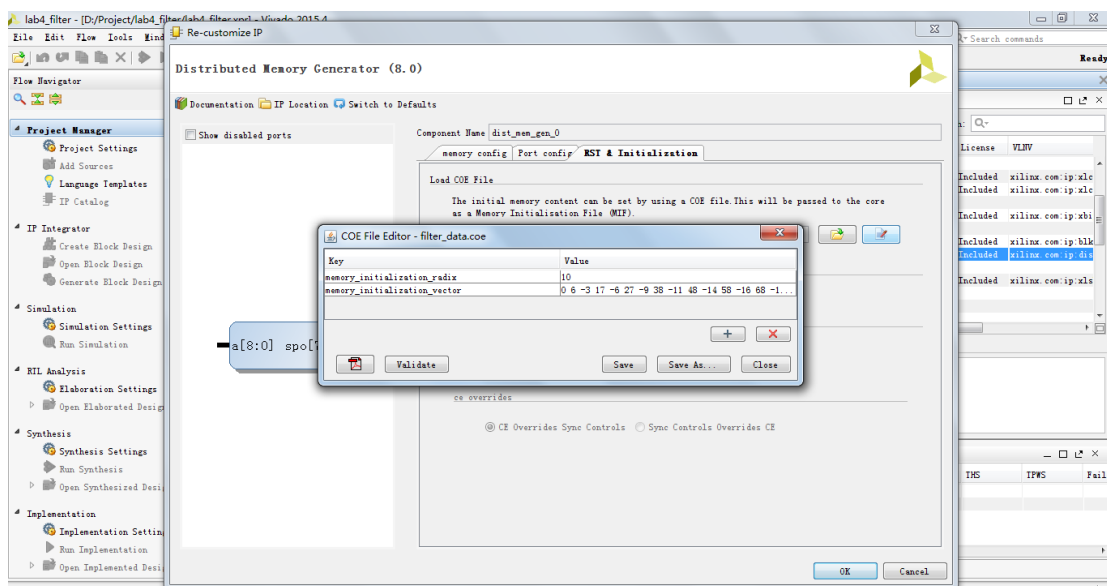


3.4、点击进入“RST & Initialization”标签页，在“Coefficients File”一栏通过点击文件夹图标“Browse”，选择后缀名“.coe”的文件（用于加载到ROM中），点击“OK”。

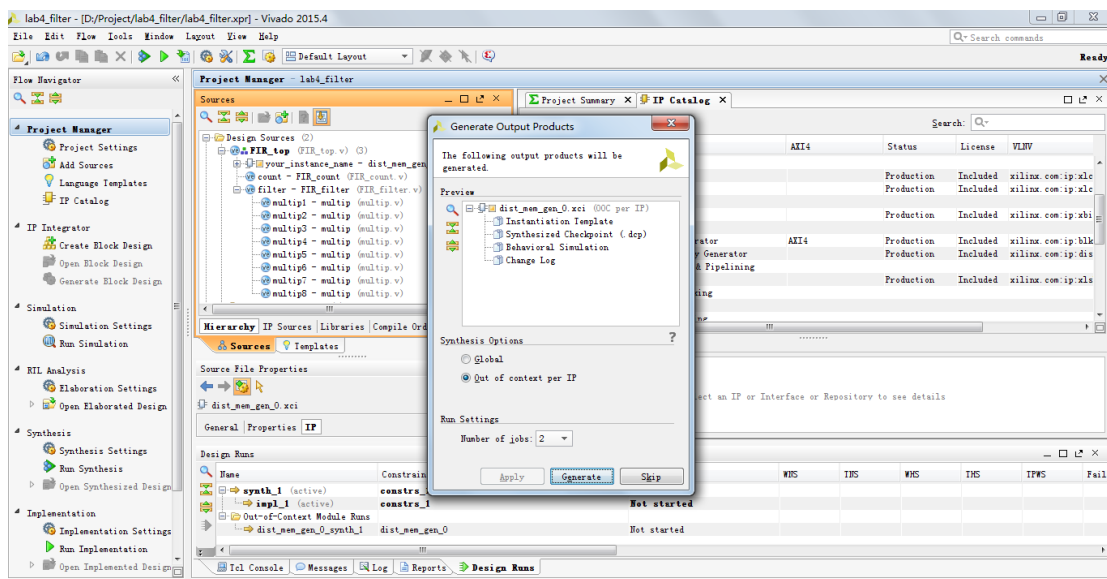




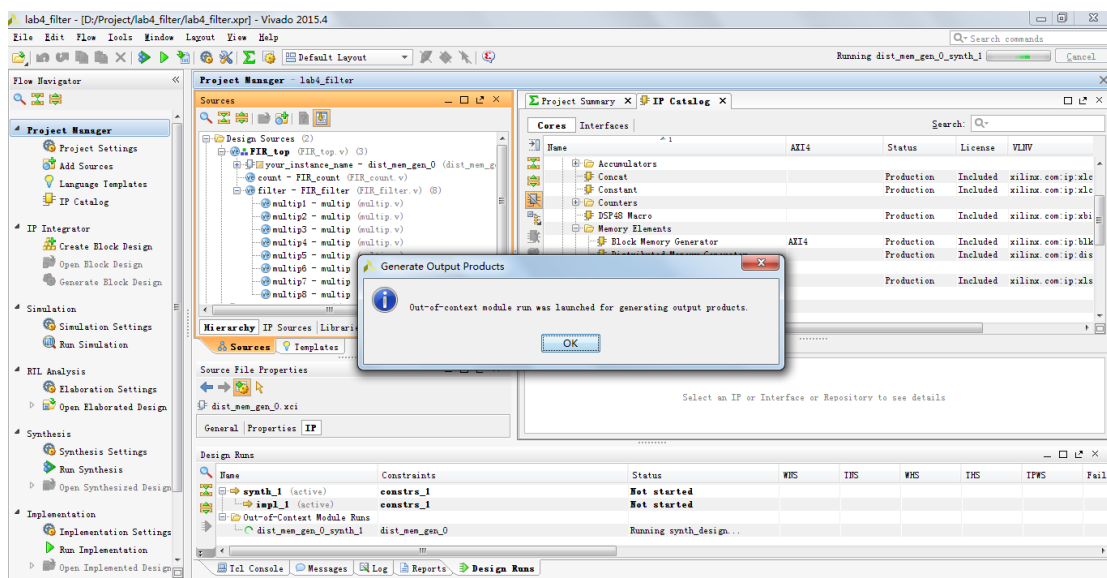
3.5、点击“Coefficients File”一栏最右侧的图标“Edit”，查看或修改初始化文件的内容。查看完毕后点击“Close”，再点击“OK”。



3.6、在弹出的对话框中选择“Out of context per IP”，再点击“Generate”。

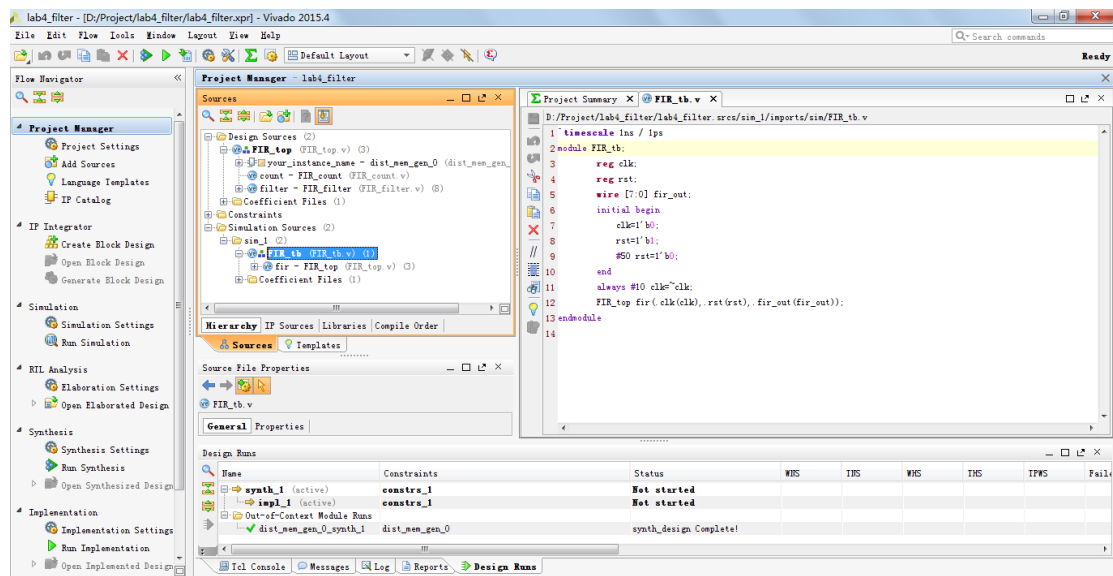


3.7、在弹出的对话框中点击“OK”。

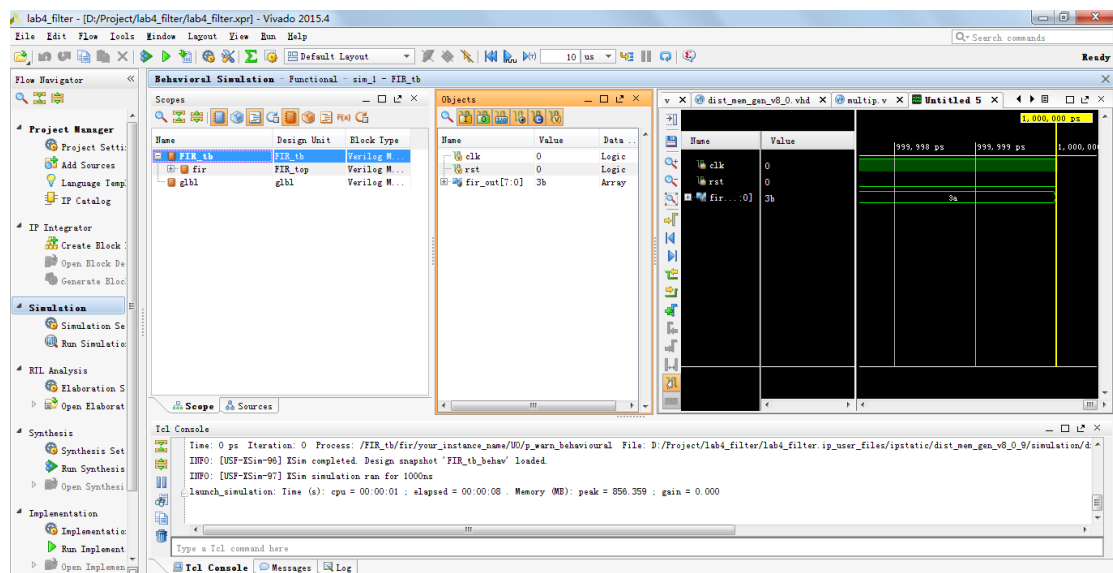


### 三、仿真

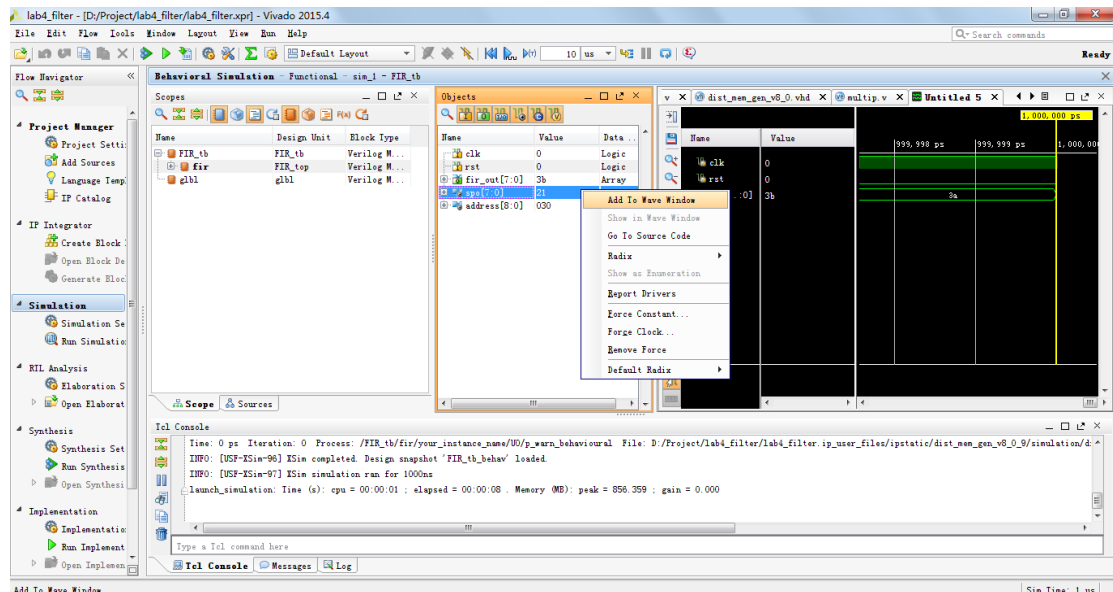
4、仿照实验二的流程，添加仿真文件 FIR\_tb.v。



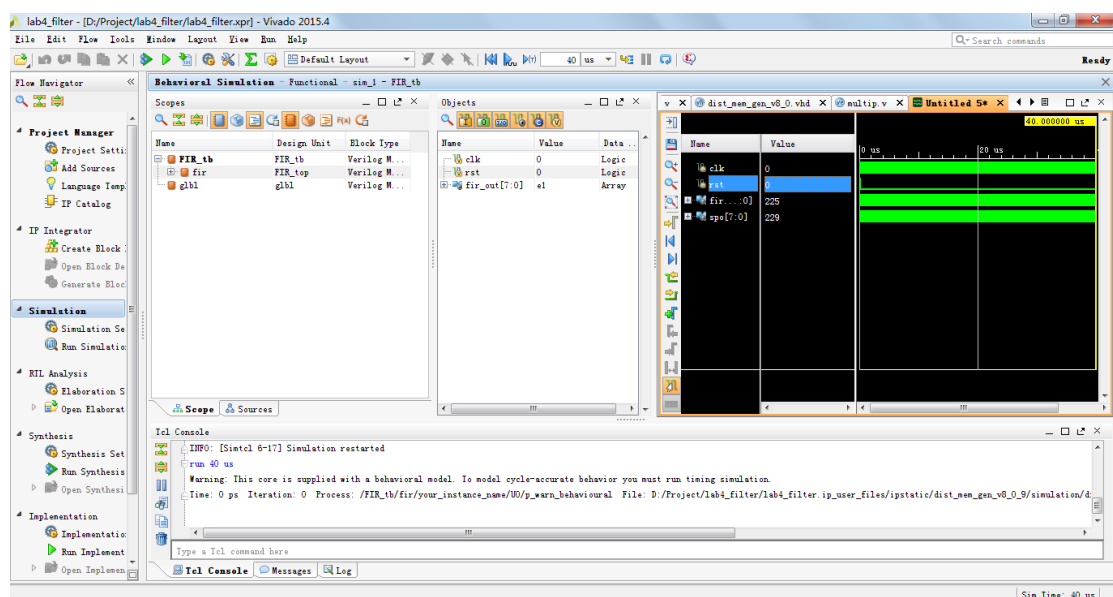
5、在左侧“Flow Navigator”一栏中的“Simulation”下点击“Run Simulation”，选择“Run Behavior Simulation”，进入仿真界面。



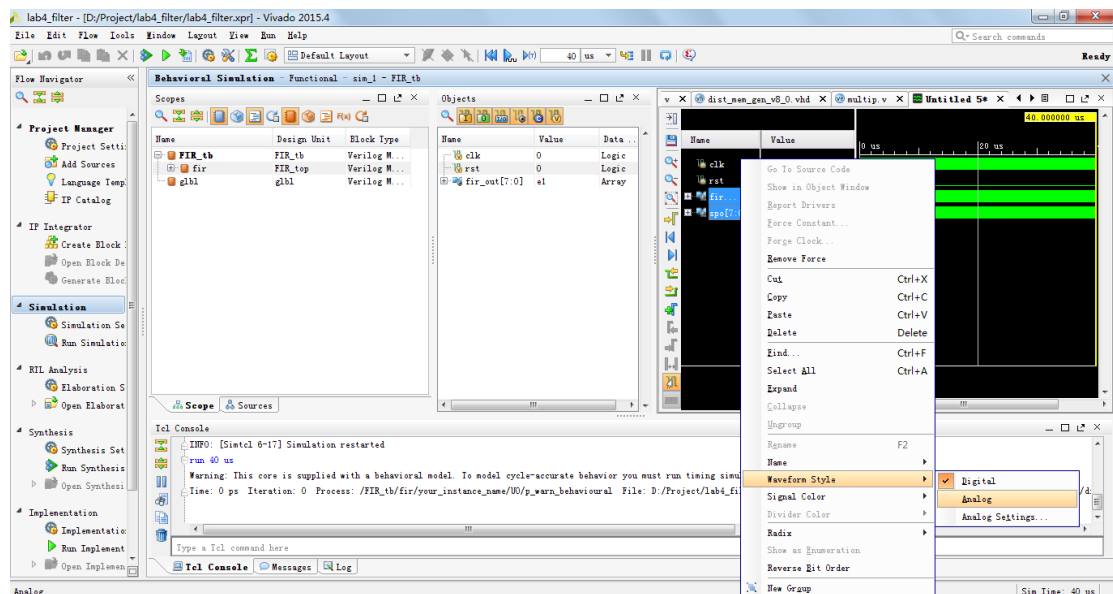
6、在“Scopes”窗口选中“FIR\_tb/fir”，在右侧的“Objects”窗口中选择“spo[7:0]”信号，右击选择“Add To Wave Window”。



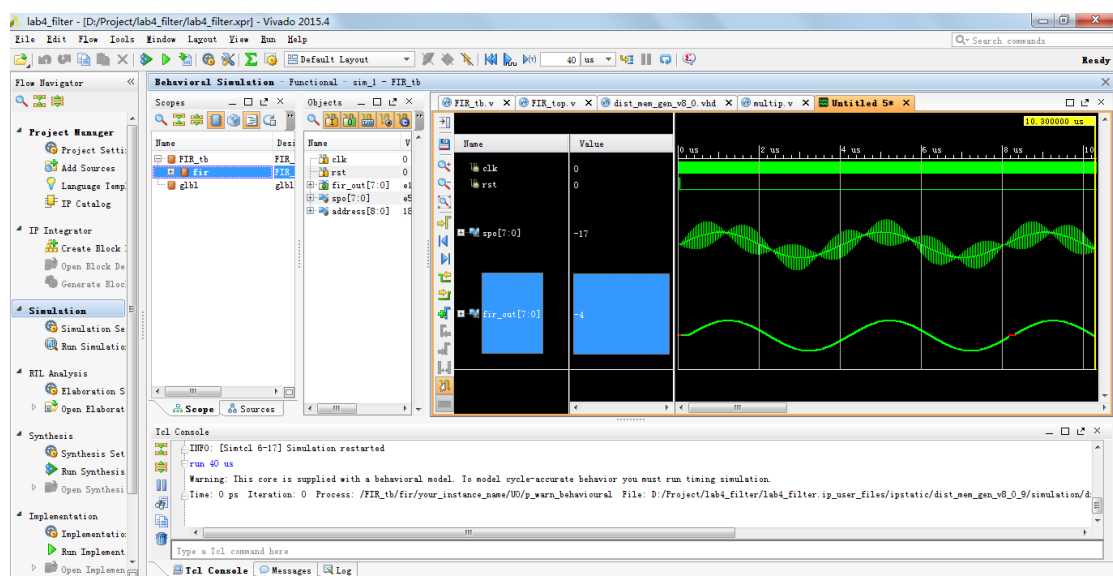
7、在波形窗口上方的工具栏中点击“Restart”，然后在其右侧运行时间一栏中输入 40  $\mu$  s，点击“Run for 40  $\mu$  s”。



8、将未经滤波的波形数据“spo[7:0]”和输出信号“fir\_out[7:0]”以模拟波形的形式显示。在波形窗口选中“fir\_out[7:0]”及“spo[7:0]”，单击右键，在弹出的选项中选择“Radix->Signed Decimal”和“Waveform Style->Analog”。



9、调整界面布局，通过“Zoom Fit”、“Zoom In”及“Zoom Out”，将波形缩放到合适大小。

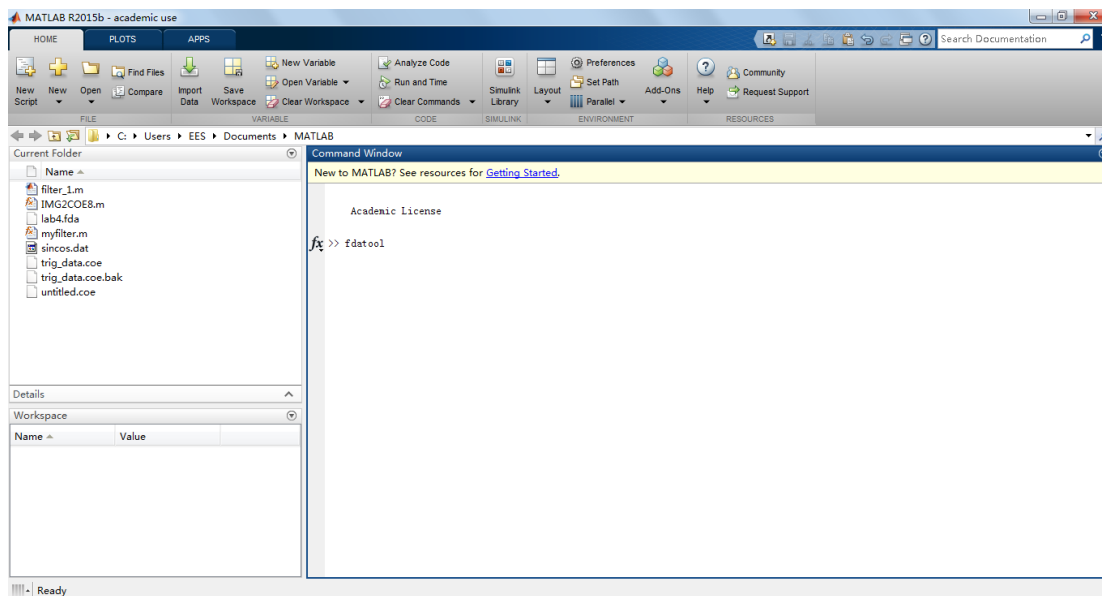


10、完成仿真后，即可编写约束文件，进行综合、实现、生成比特流并下载到开发板的 FPGA 芯片中。

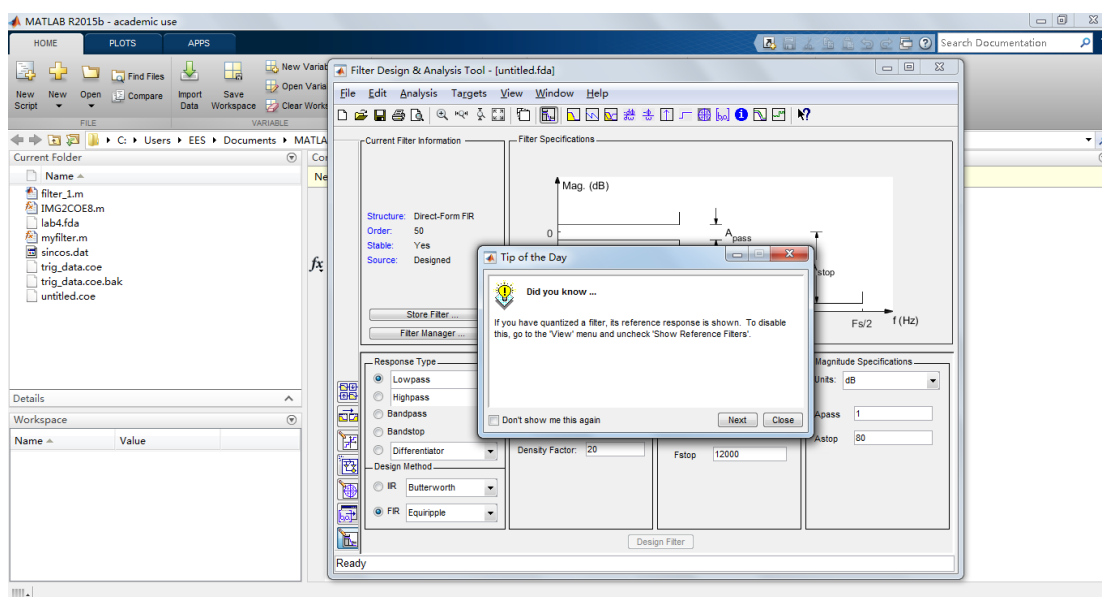


## 附录、用 Matlab 生成待滤波信号

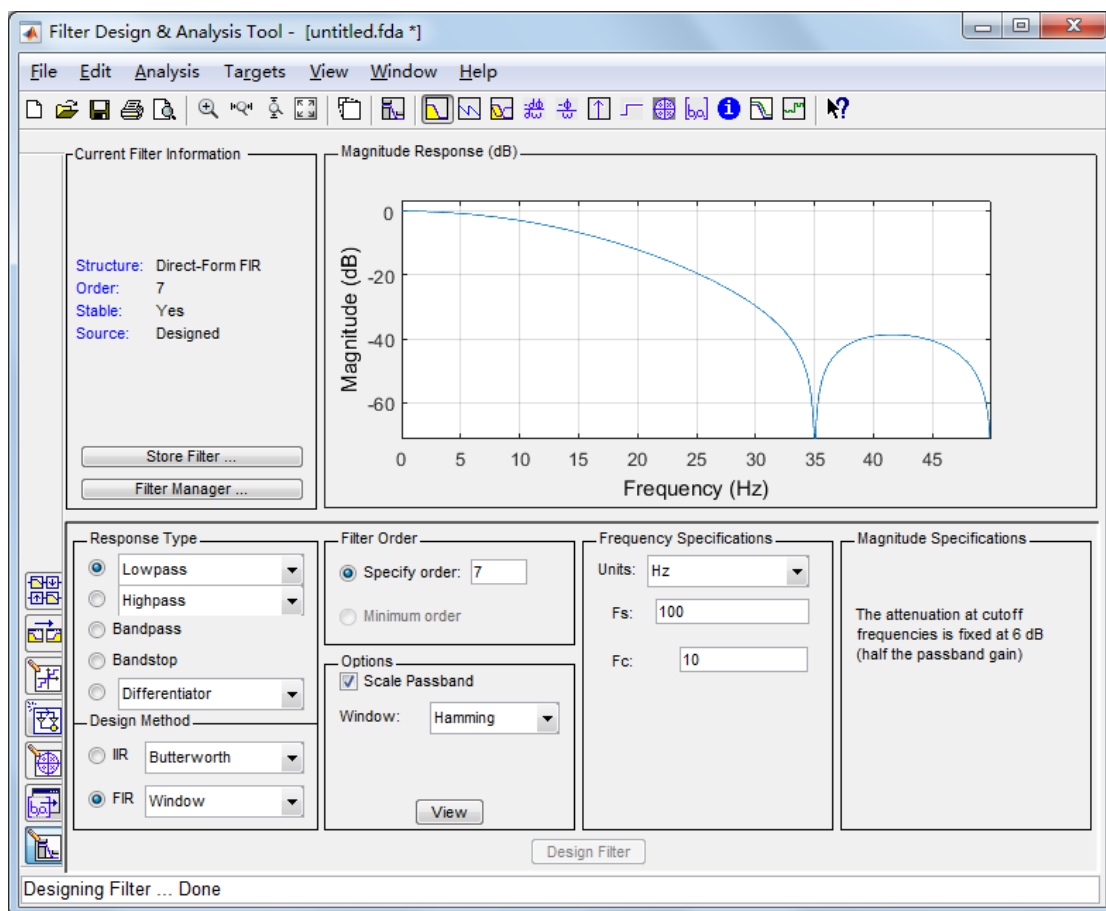
- 1、打开 MATLAB 软件，在命令窗口输入 `fdatool` 并回车，就会弹出滤波器设计工具。



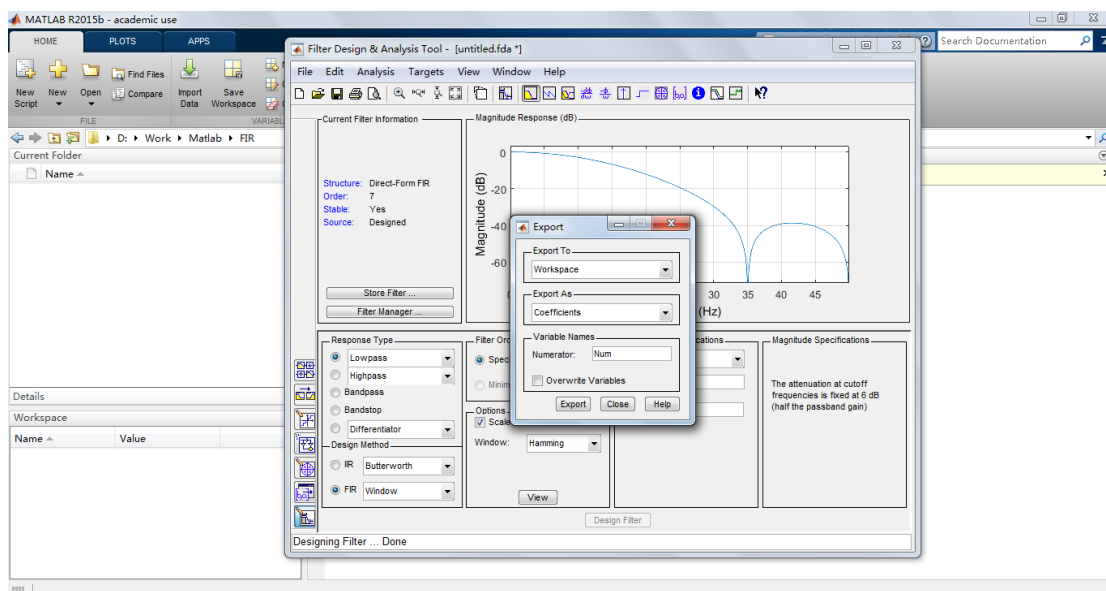
- 2、在弹出的提示窗口点击“Close”。



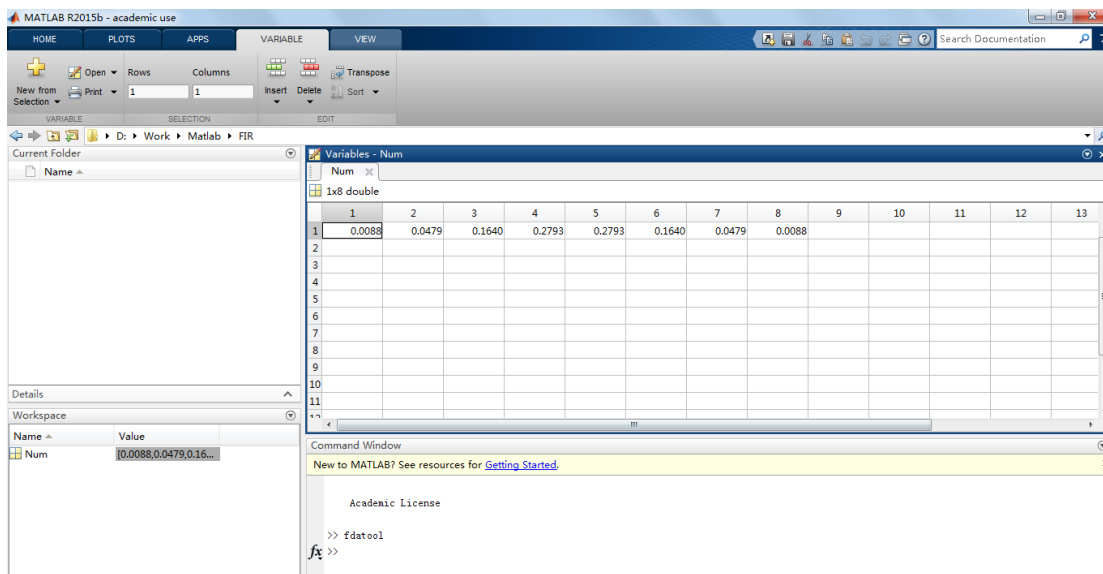
- 3、按照本次实验的要求，在响应类型 **Response Type** 中选择低通 Lowpass；设计方法 **Design Method** 中选择 FIR，并且选择用窗函数法 **Window** 进行 FIR 数字滤波器的设计。在 **Filter Order** 中选择 Specify order，在这里输入 7，注意这里输入的数值是所要设计的滤波器的阶数减 1。在 **Options** 中勾选 Scale Passband，并将 Window 选为 Hamming。在 **Frequency Specifications** 中单位选择 HZ，采样频率  $F_s$  输入值为 100，截止频率  $F_c$  中输入值为 10。点击 Design Filter 即可设计出所需的滤波器。



- 4、在 FDATool 工具界面中，点击 File 选择 Export，在弹出窗口中点击 Export，即可在 MATLAB 中生成所设计的滤波器的抽头系数。



- 5、结果可以在 MATLAB 的 Workspace 中看到。



6、具体计算出的系数如下：

$$H(0)= 0.00875474290329771$$

$$H(1)= 0.0479488721685246$$

$$H(2)= 0.164024391089411$$


$$H(3)= 0.279271993838767$$

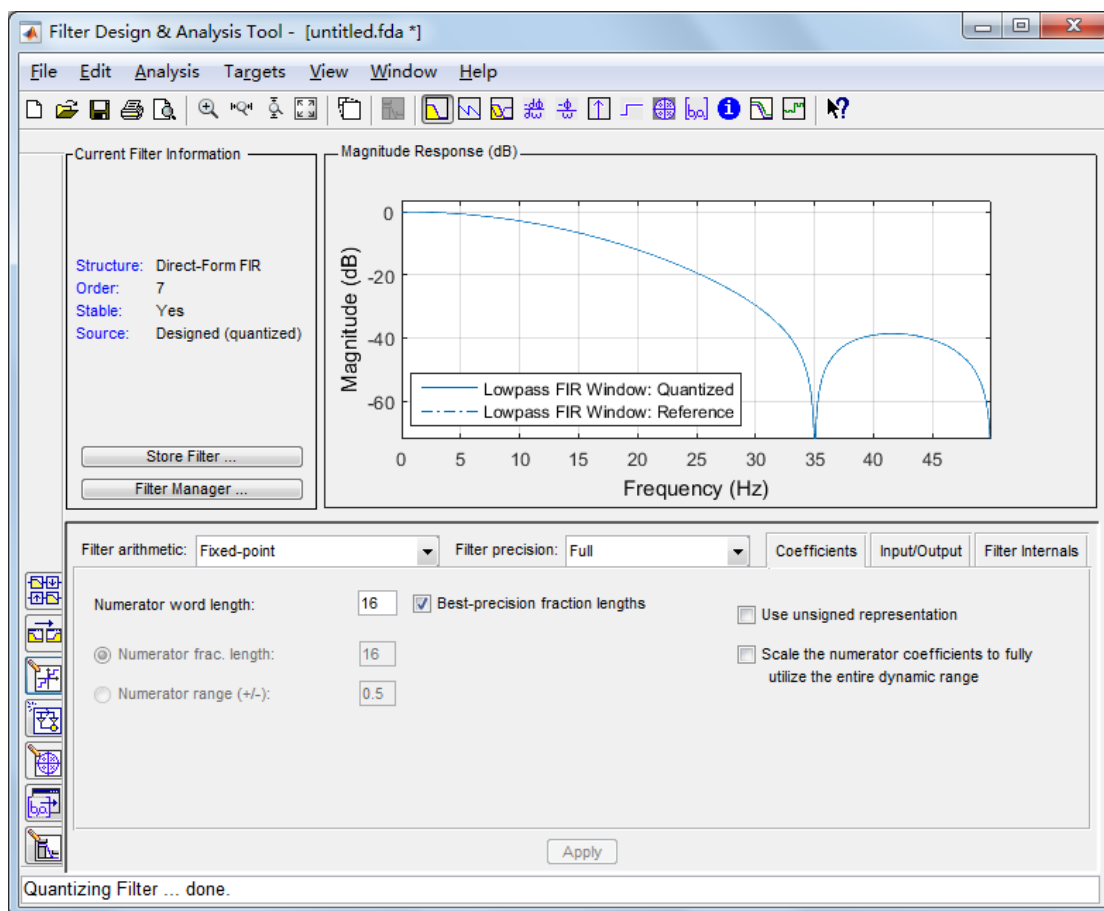
$$H(4)= 0.279271993838767$$

$$H(5)= 0.164024391089411$$

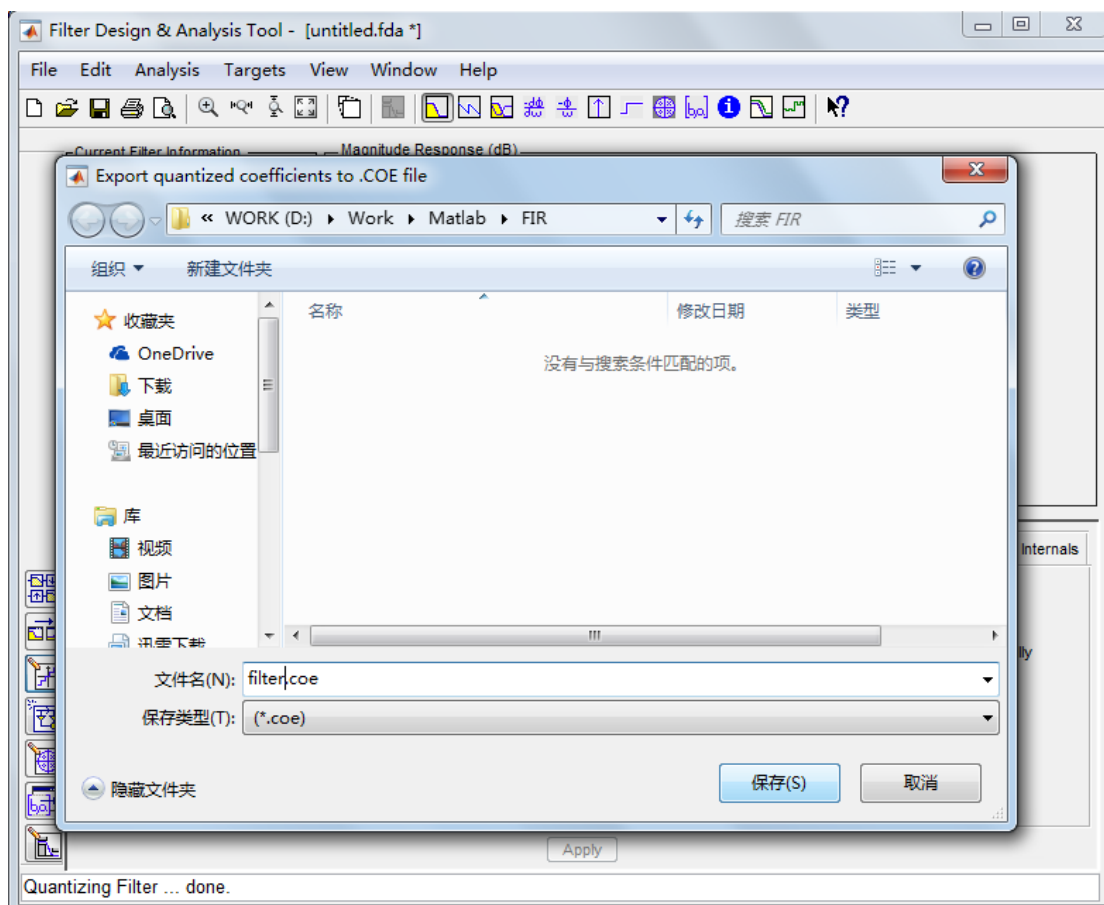
$$H(6)= 0.0479488721685246$$

$$H(7)= 0.00875474290329771$$

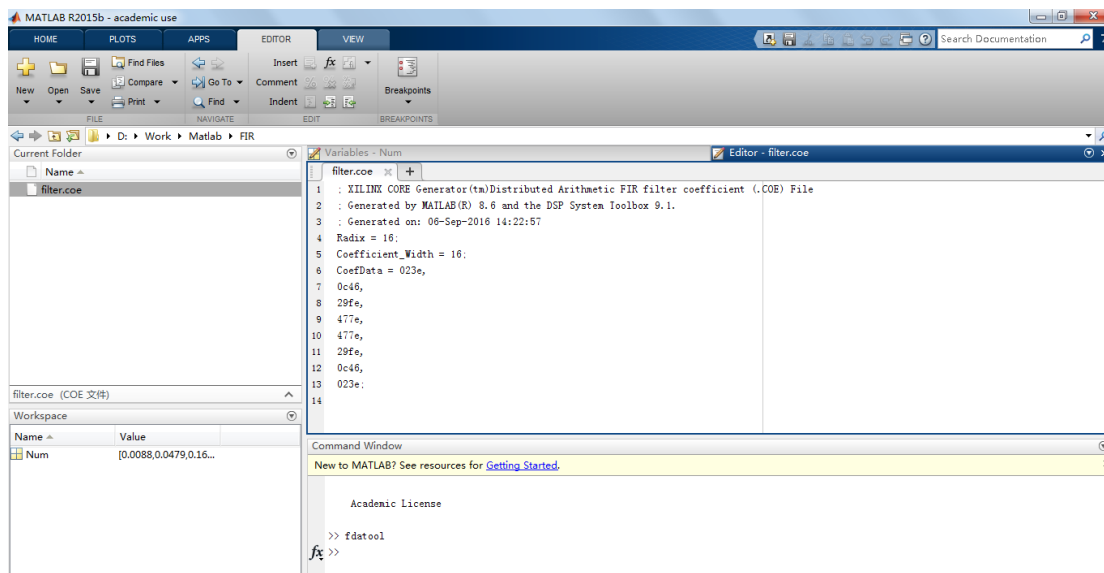
因为 FPGA 并不支持浮点数的运算，所以需要对抽头系数进行量化处理。在 FDATool 界面中，首先点击按钮 ，在 Filter arithmetic 中选择 Fixed-point，Number word length 中可以输入的是字长，当输入 8 时，点击 Apply，可以看到有较大的偏差。所以将数值改为 16。



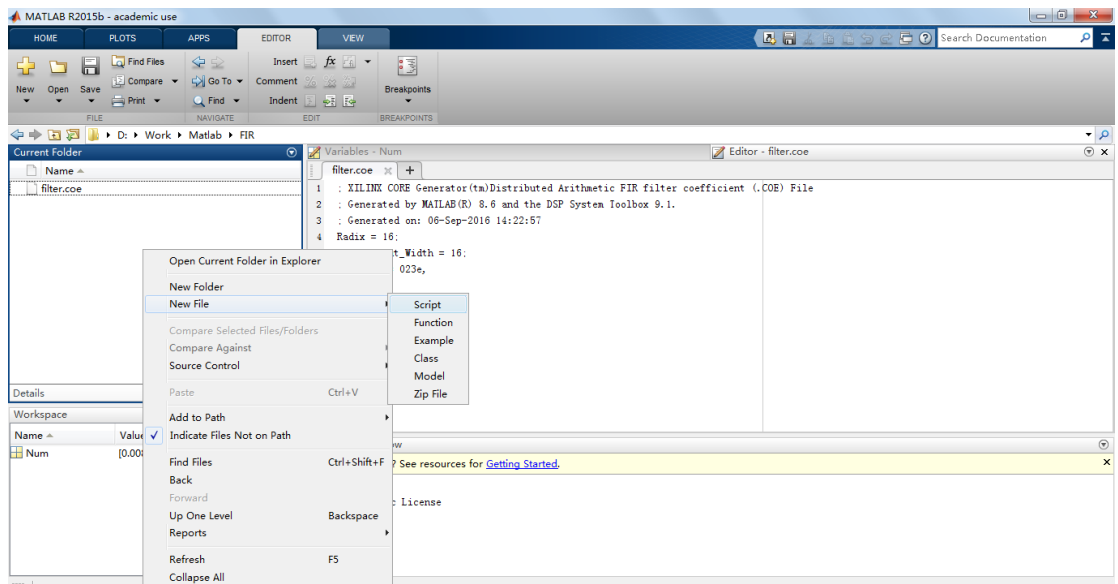
- 7、其次，点击 **Target**，在下拉菜单中选择 **XILINX coefficient(.COE) file** 即可生成以.COE 文件形式保存的经过量化的滤波器系数，当滤波器的阶数较高时可以在 vivado 中通过 ROM 的 IP 核读入，本次实验的滤波器阶数比较少，所以可以直接将结果复制到滤波器的 verilog 程序中。



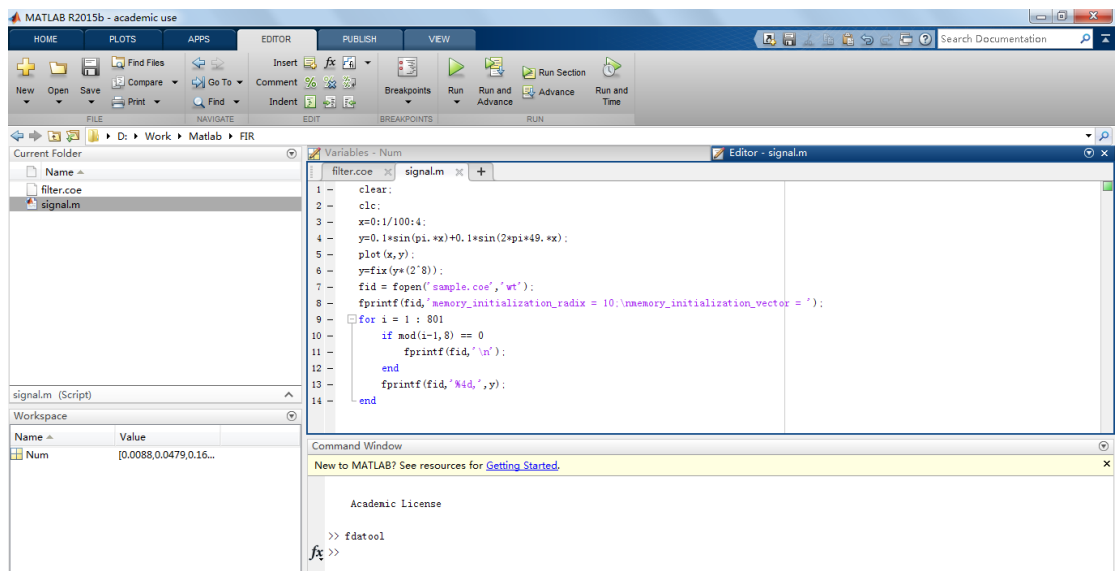
8、MATLAB 主界面可查看生成的 coe 文件内容。



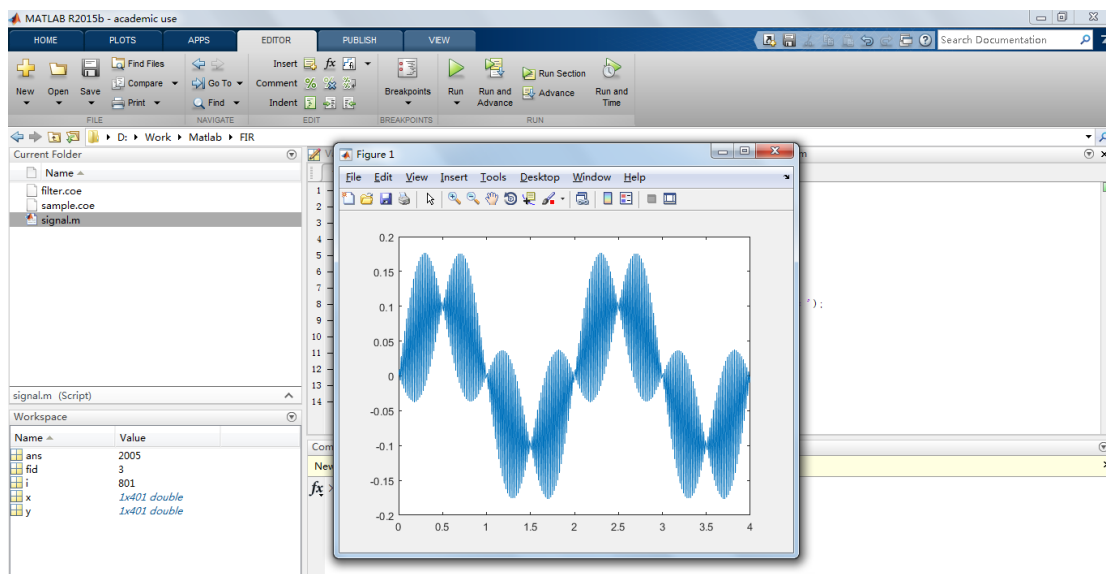
9、在 MATLAB 主界面的 Current Folder 窗口空白处右击，选择 New File->Script。



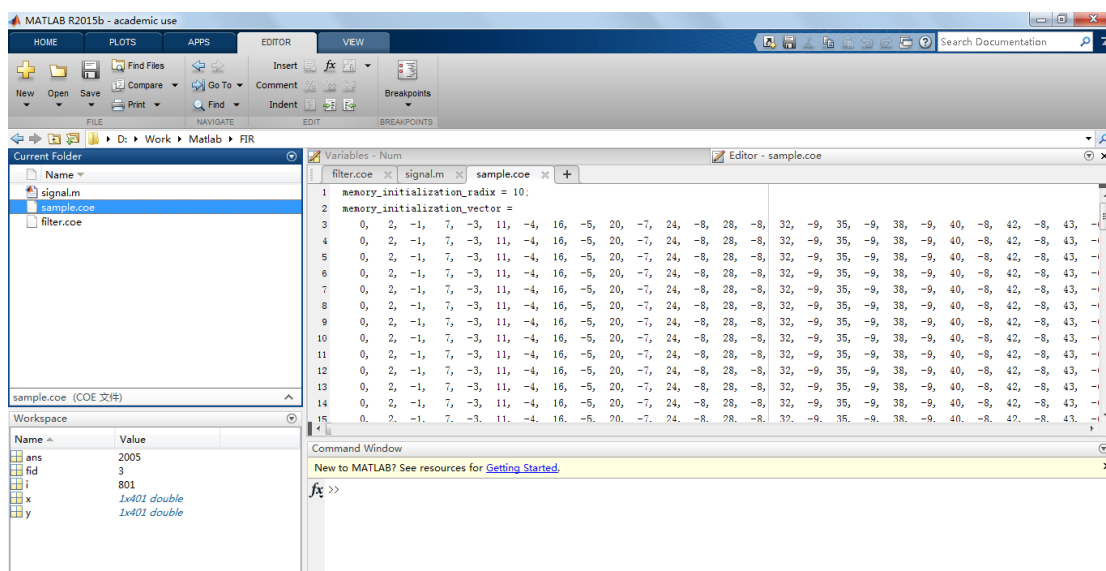
- 10、将新生成的后缀名为“.m”的文件重命名为“signal.m”，并双击打开，编写程序，在 MATLAB 中产生待滤波的波形，并进行采样量化后以.coe 文件形式输出。



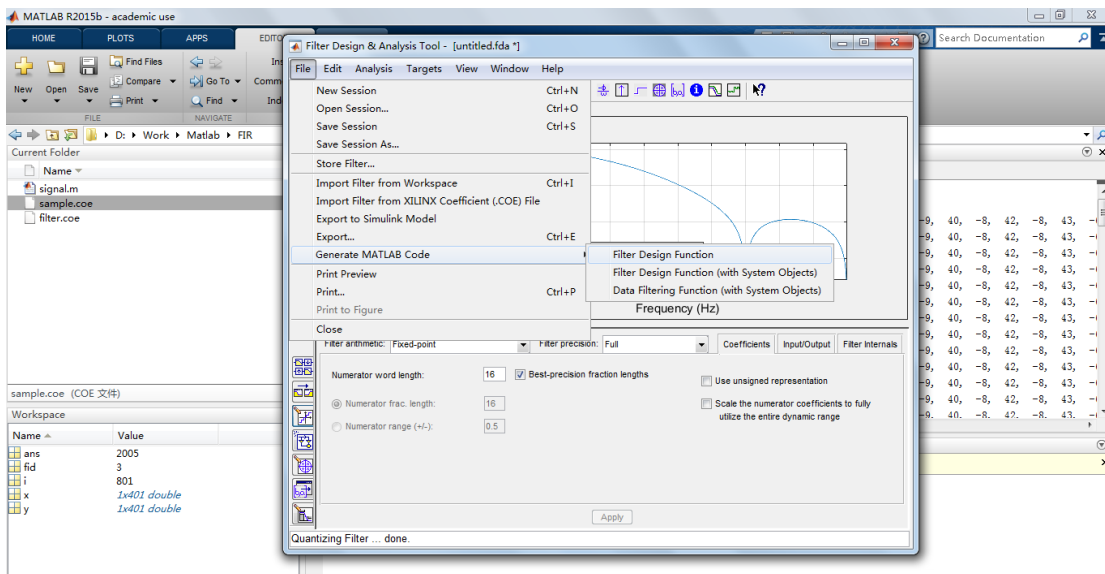
- 11、点击工具栏中的 Run，运行程序，可以看到弹出的波形窗口。



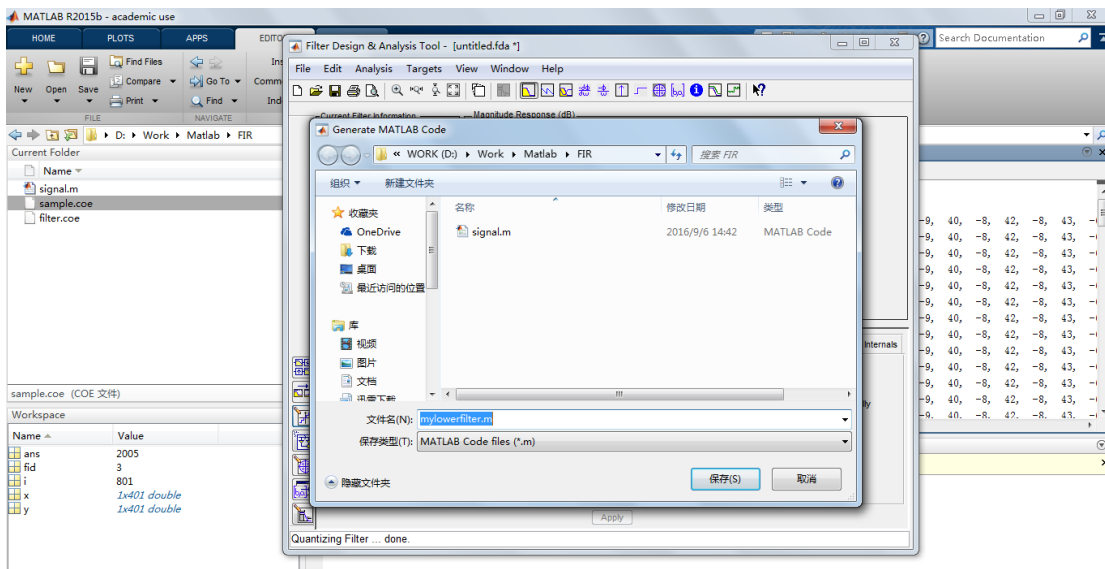
- 12、在 Current Folder 窗口中可以看到生成了名为“sample.coe”的文件，双击打开可查看其内容。在 vivado 中通过 ROM 的 IP 核调用该文件数据，即将波形数据存放在 ROM 中。



- 13、在 Fdatool 工具界面中，选择 File -> Generate MATLAB Code -> Filter Design Functions。

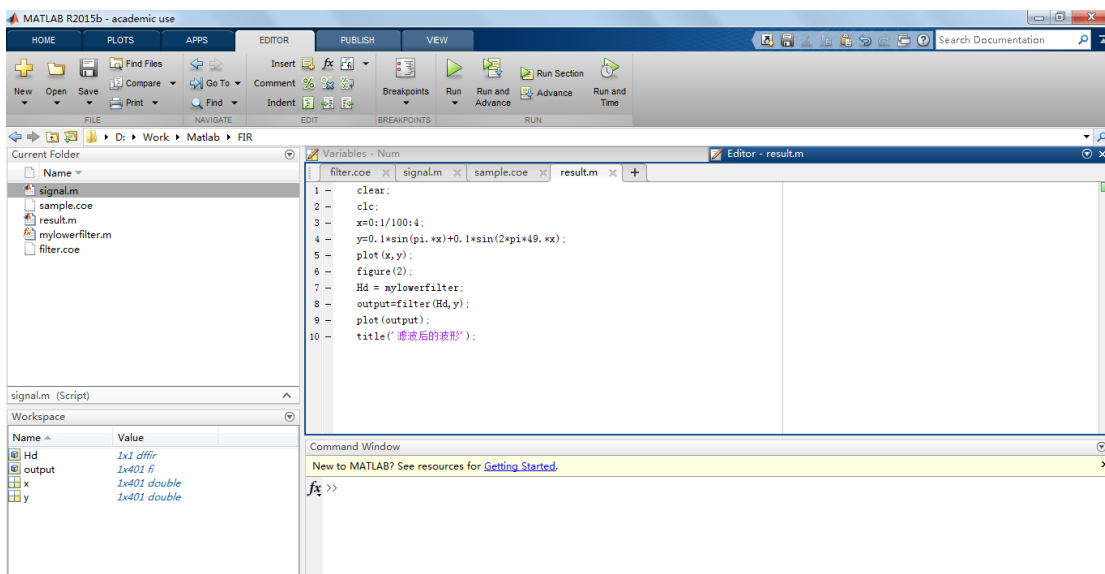


14、在弹出的窗口中将文件名修改为 mylowerfilter.m，点击保存。



15、在 MATLAB 主界面的 Current Folder 窗口中可以看到生成的 mylowerfilter.m 文件。  
在 Current Folder 窗口空白处右击，选择 New File->Script，将新生成的后缀名为“.m”的文件重命名为“result.m”，双击打开并编写程序。将之前生成的波形通过所设计的滤波器，并将输出的数据绘制成波形。





16、点击工具栏中的 Run, 运行程序, 可以看到滤波前后的波形, 通过对比输入输出波形, 观察滤波效果。

