بسم الله الرحمن الرحیم




@MatlabOnline
کانال تخصصی
آکادمی Matlab دانشگاه تهران
آموزش تخصصی کلیه ToolBox ها.
آموزش Matlab 2016
با عضویت هزاران نخبه دانشگاهی

# فیلتر کردن سیگنال بوسیله FPGA

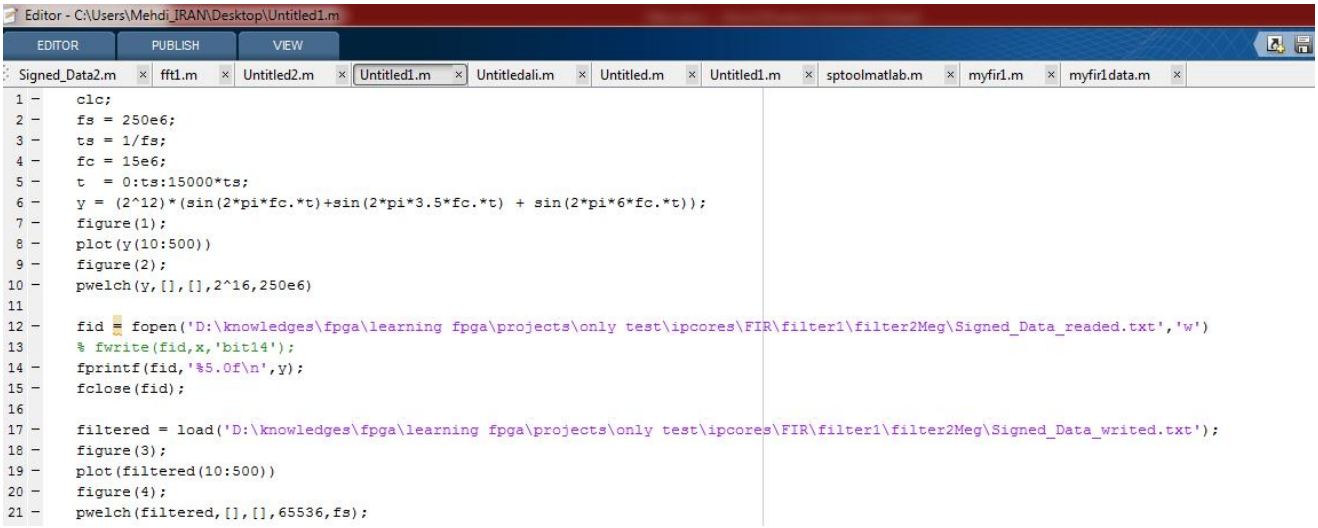
# و مقایسه نتایج با فیلتر کردن سیگنال بوسیله MATLAB

## مقدمه:

در این پروژه قصد داریم سیگنال تولید شده در MATLAB را توسط Xilinx ISE خوانده و از IP Core فیلتر FIR عبور دهیم و نتیجه فیلتر شده را پس از ذخیره‌سازی در فایل متنی توسط Xilinx ISE در MATLAB مشاهده کنیم.

# تولید سیگنال در MATLAB:

در کدهای زیر سه سیگنال سینوسی با فرکانس‌های 10MHz، 35MHz و 60MHz را با هم جمع میکنیم و علاوه بر نمایش در حوزه فرکانس نتیجه حاصل را در فایل متنی Signed_data_readed.txt ذخیره می‌کنیم.



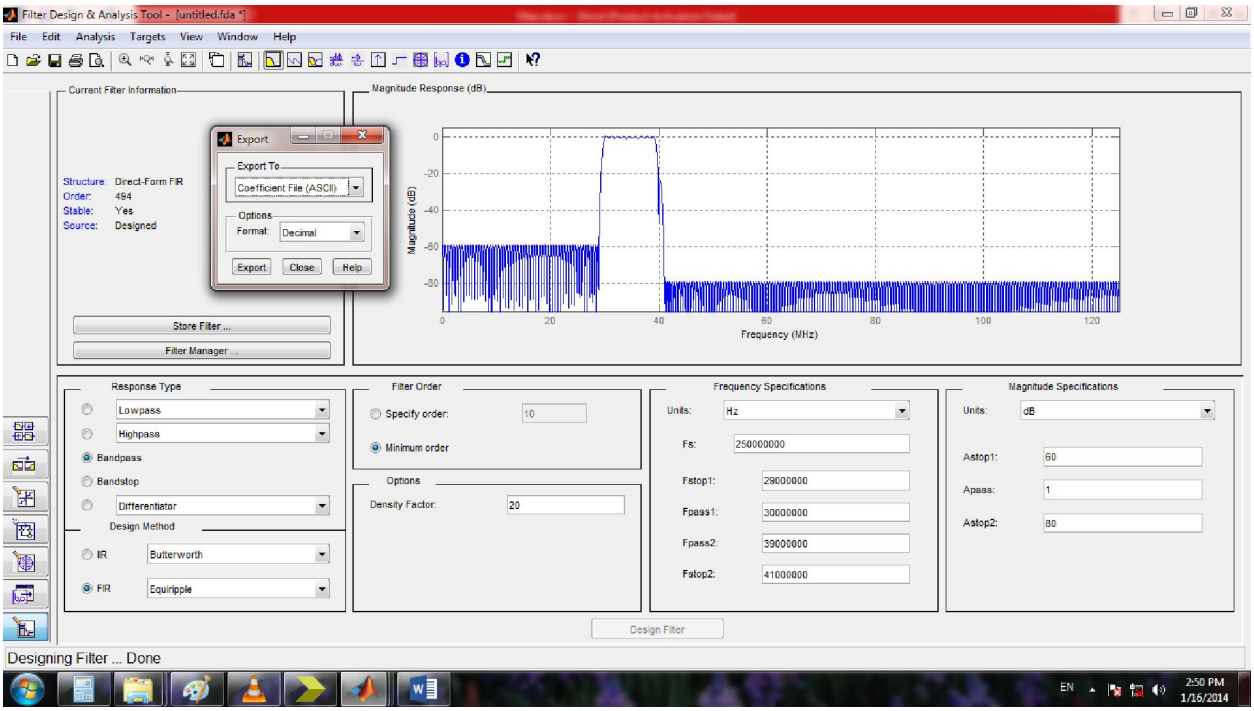

برای اینکه نتایج ذخیره شده در فایل متنی قابل فراخوانی از Xilinx ISE باشد در خط 14 سیگنال را طوری ذخیره می کنیم که دارای 5 رقم صحیح و 0 رقم اعشاری(%5.0f )باشد و هر عدد در سطری مجزا (\n) باشد.

# طراحی فیلتر با fdatool:

برای تنظیم کردن فیلتر FIR در Xilinx ISE باید ضرایب فیلتر را بدست آوریم بنابراین دستور fdatool را در MATLAB command window وارد میکنیم تا پنجره زیر ظاهر شود.

از آنجایی که می‌خواهیم از سه سیگنال سینوسی فقط سیگنال سینوسی 35MHz را عبور دهیم پس فیلتر Bandpass را انتخاب می‌کنیم و تنظیمات زیر را همانند شکل زیر در آن اعمال می‌کنیم و سپس دکمه Design Filter را می‌زنیم.

از منوی file زیرمنوی Export را می‌زنیم. این پنجره را همانند شکل تنظیم می‌کنیم و دکمه Export را می‌زنیم تا ضرایب فیلتر در فایلی با پسوند coe. ذخیره شوند.



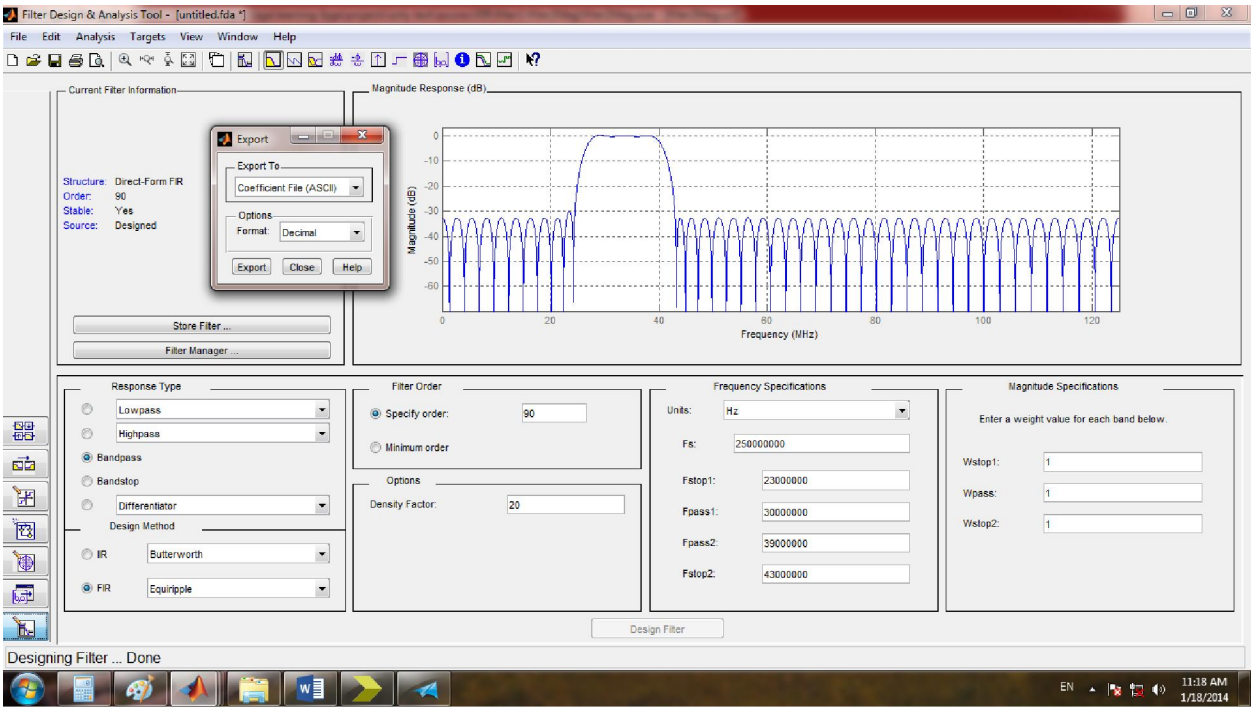

در شکل بالا مرتبه فیلتر بالا است و فیلتر در لبه‌ها تیز است و ممکن است در ساخت bit فایل در قسمت MAP ممکن است خطا ایجاد شود بنابراین همانطور که در شکل زیر دیده می‌شود ما فاصله Fstop1 را از Fpass1 و همچنین فاصله Fstop2 را از Fpass2 را زیاد کردیم و همچنین مرتبه فیلتر را کاهش دادیم.



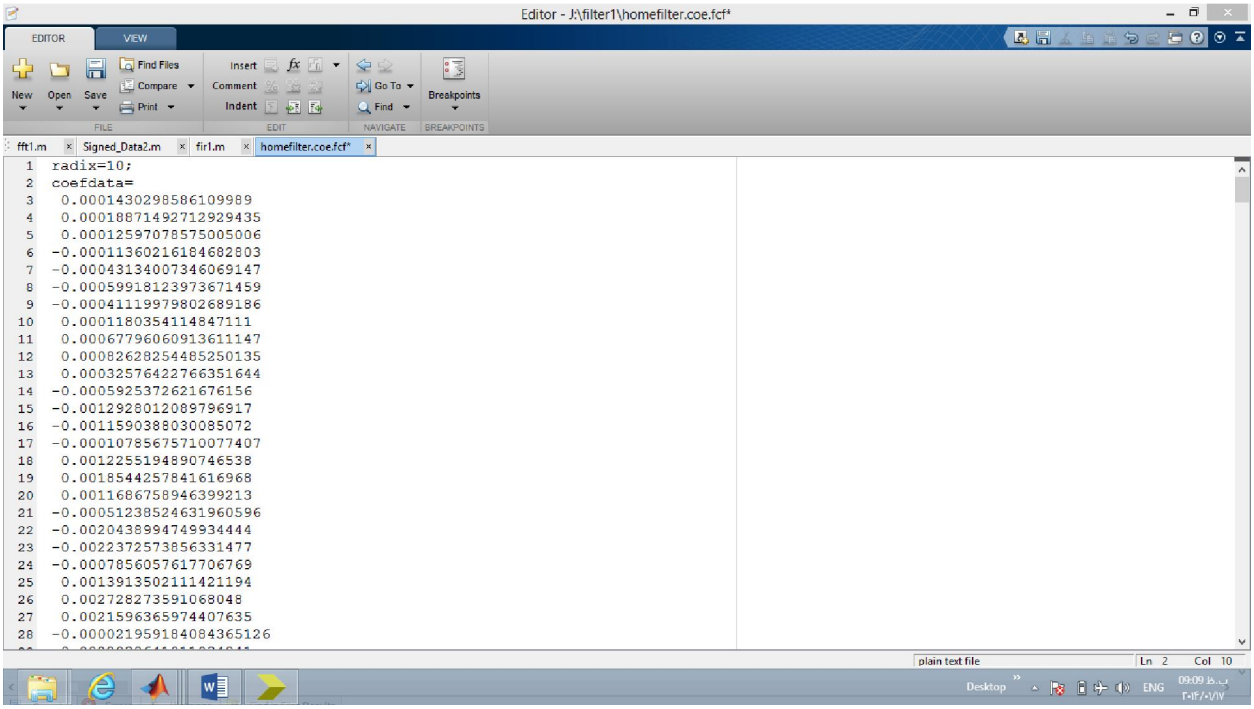


@MatlabOnline

کانال تخصصی
آکادمی Matlab دانشگاه تهران
آموزش تخصصی کلیه ToolBox ها.
آموزش Matlab 2016
با عضویت هزاران نخبه دانشگاهی




در ابتدای این فایل که ضرایب در آن ذخیره شده‌اند تغییراتی اعمال می‌کنیم تا همانند شکل زیر شود.

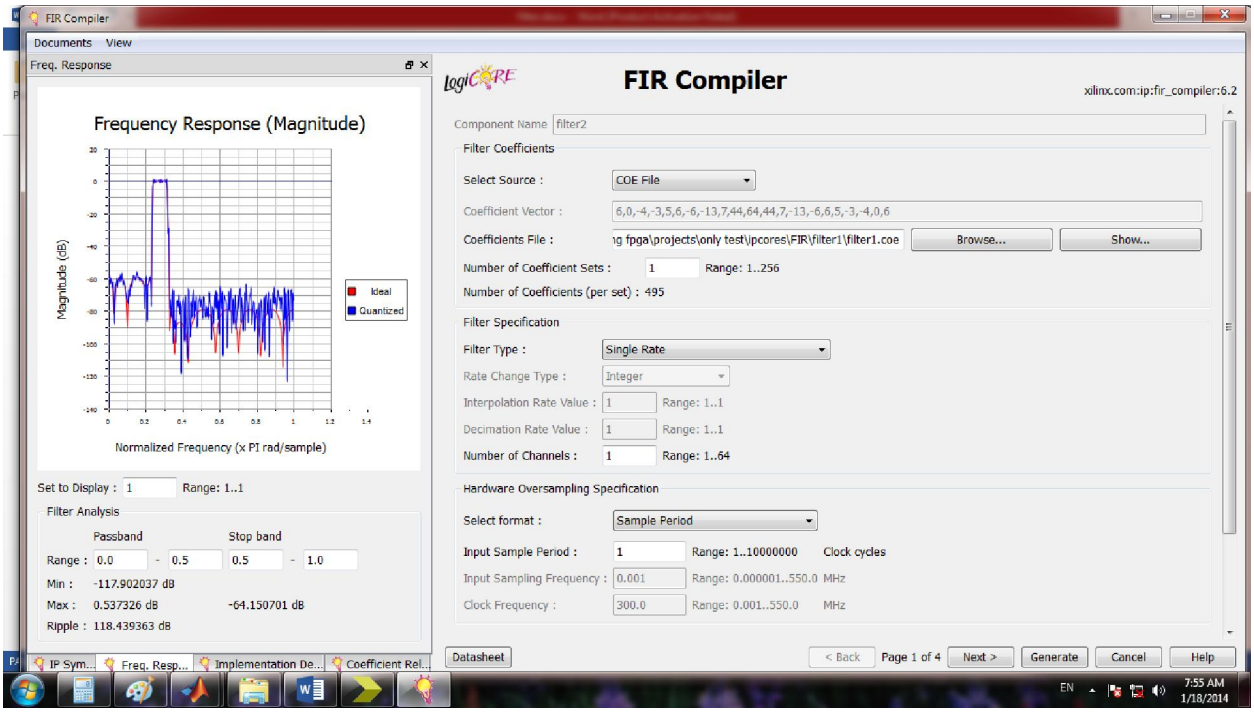



حال این فایل برای استفاده در FIR IP Core آماده شده است.

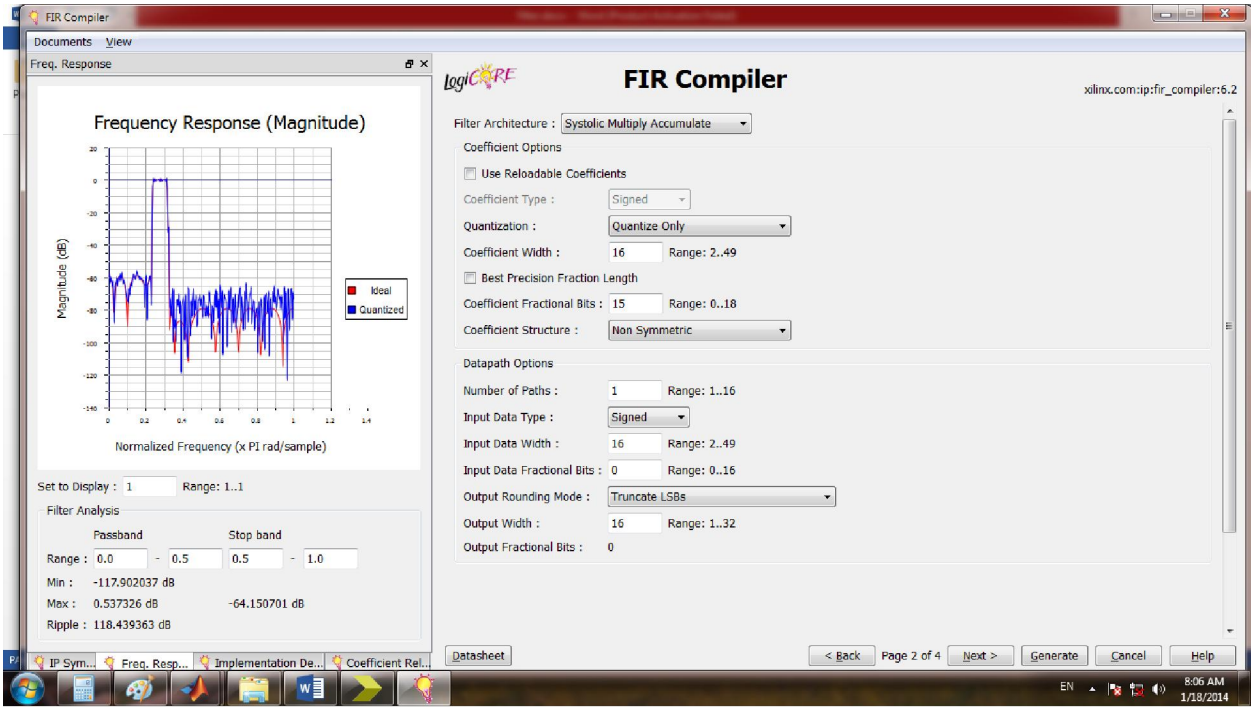
# تولید کدهای VHDL جهت اعمال فیلتر بر روی سیگنال:

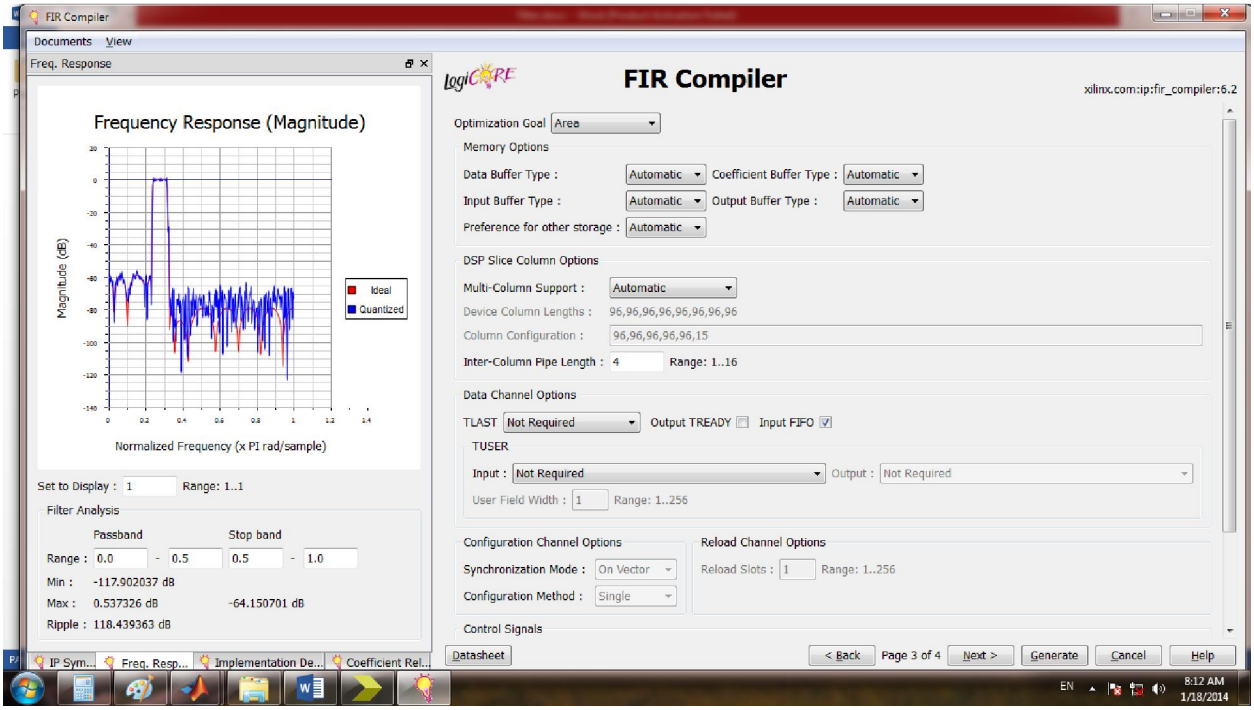
در Xilinx ISE پروژه جدیدی را درست کرده و ابتدا یک FIR IP Core درست کرده و سپس بصورت زیر تنظیم می‌کنیم.

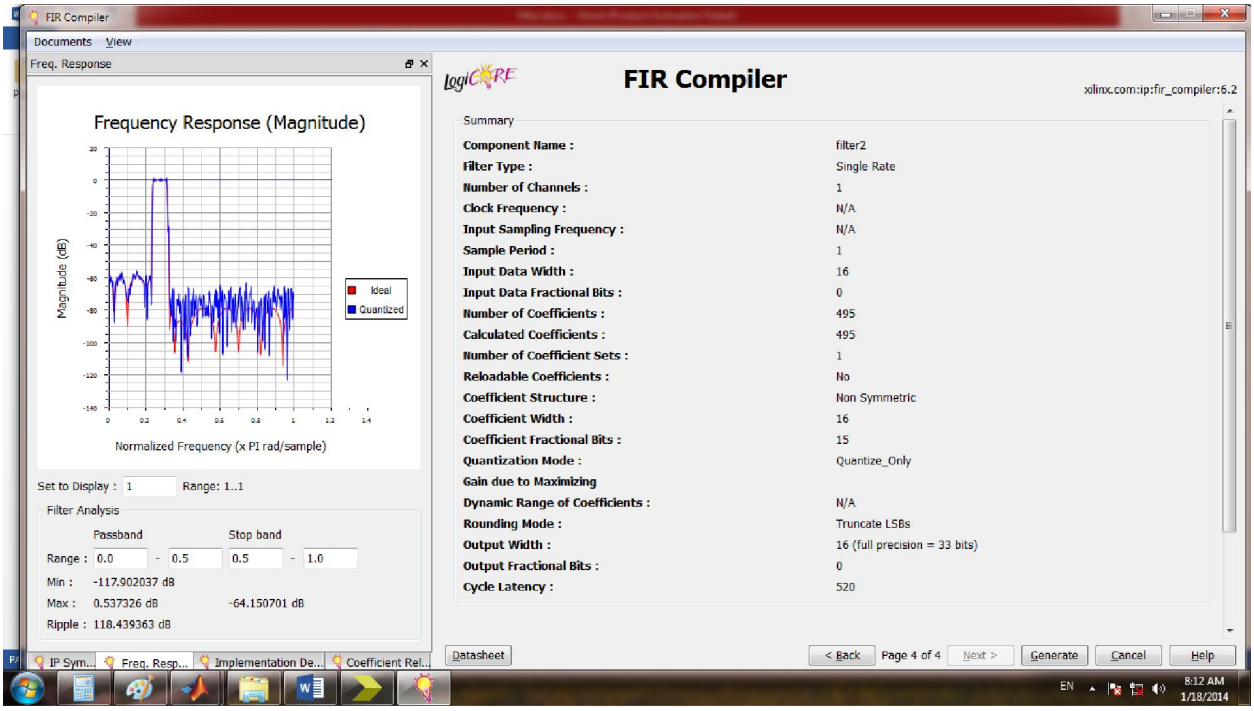
در صفحه اول نوع دریافت ضرایب را COE File انتخاب و آدرس فایل ذخیره کننده ضرایب را وارد میکنیم.



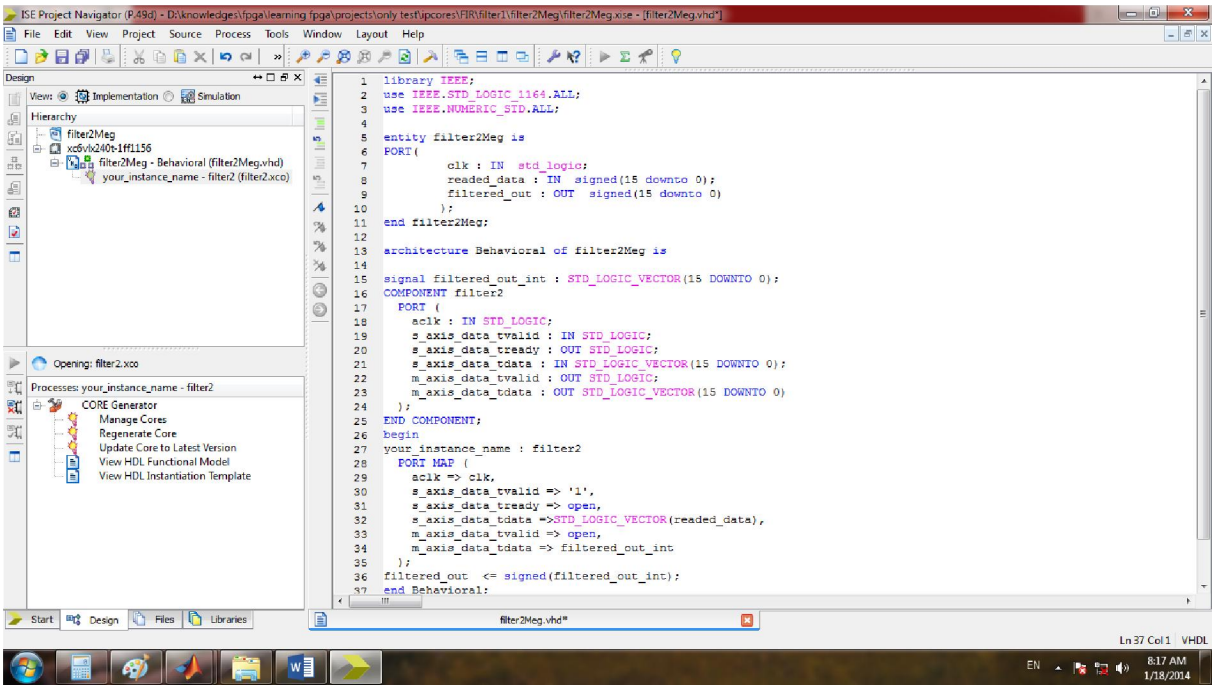

اگر به ارقام ضرایب توجه شود مشاهده می‌شود که ارقامی بین 1 و1- هستند بنابراین در صفحه دوم برای بدست آوردن دقتی نزدیک به دقت ضرایب ایده‌آل توسط ضرایب کوانتیزه شده، مقدار coefficient fractional bit را (15) نزدیک به مقدار coefficient width (16) قرار می‌دهیم. صفحه سوم و چهارم را بدون تغییر باقی می‌گذاریم.


FIR Compiler
Documents View
Freq. Response
Frequency Response (Magnitude)
Magnitude (dB)
Ideal
Quantized
Normalized Frequency (x PI rad/sample)
Set to Display : 1 Range: 1..1
Filter Analysis
Passband
Stop band
Range : 0.0 - 0.5 0.5 - 1.0
Min : -117.902037 dB
Max : 0.537326 dB -64.150701 dB
Ripple : 118.439363 dB
IP Sym...
Freq. Resp...
Implementation De...
Coefficient Rel...
FIR Compiler
xilinx.com:ip:fir_compiler:6.2
Filter Architecture : Systolic Multiply Accumulate
Coefficient Options
Use Reloadable Coefficients
Coefficient Type : Signed
Quantization : Quantize Only
Coefficient Width : 16 Range: 2..49
Best Precision Fraction Length
Coefficient Fractional Bits : 15 Range: 0..18
Coefficient Structure : Non Symmetric
Datapath Options
Number of Paths : 1 Range: 1..16
Input Data Type : Signed
Input Data Width : 16 Range: 2..49
Input Data Fractional Bits : 0 Range: 0..16
Output Rounding Mode : Truncate LSBs
Output Width : 16 Range: 1..32
Output Fractional Bits : 0
Datasheet
< Back
Page 2 of 4
Next >
Generate
Cancel
Help
8:06 AM
1/18/2014


FIR Compiler
Documents View
Freq. Response
Frequency Response (Magnitude)
Magnitude (dB)
Ideal
Quantized
Normalized Frequency (x PI rad/sample)
Set to Display : 1 Range: 1..1
Filter Analysis
Passband
Stop band
Range : 0.0 - 0.5 0.5 - 1.0
Min : -117.902037 dB
Max : 0.537326 dB -64.150701 dB
Ripple : 118.439363 dB
IP Sym...
Freq. Resp...
Implementation De...
Coefficient Rel...
FIR Compiler
xilinx.com:ip:fir_compiler:6.2
Optimization Goal Area
Memory Options
Data Buffer Type : Automatic
Coefficient Buffer Type : Automatic
Input Buffer Type : Automatic
Output Buffer Type : Automatic
Preference for other storage : Automatic
DSP Slice Column Options
Multi-Column Support : Automatic
Device Column Lengths : 96,96,96,96,96,96,96,96
Column Configuration : 96,96,96,96,96,15
Inter-Column Pipe Length : 4 Range: 1..16
Data Channel Options
TLAST Not Required
Output TREADY
Input FIFO
TUSER
Input : Not Required
Output : Not Required
User Field Width : 1 Range: 1..256
Configuration Channel Options
Synchronization Mode : On Vector
Configuration Method : Single
Reload Channel Options
Reload Slots : 1 Range: 1..256
Control Signals
Datasheet
< Back
Page 3 of 4
Next >
Generate
Cancel
Help
8:12 AM
1/18/2014



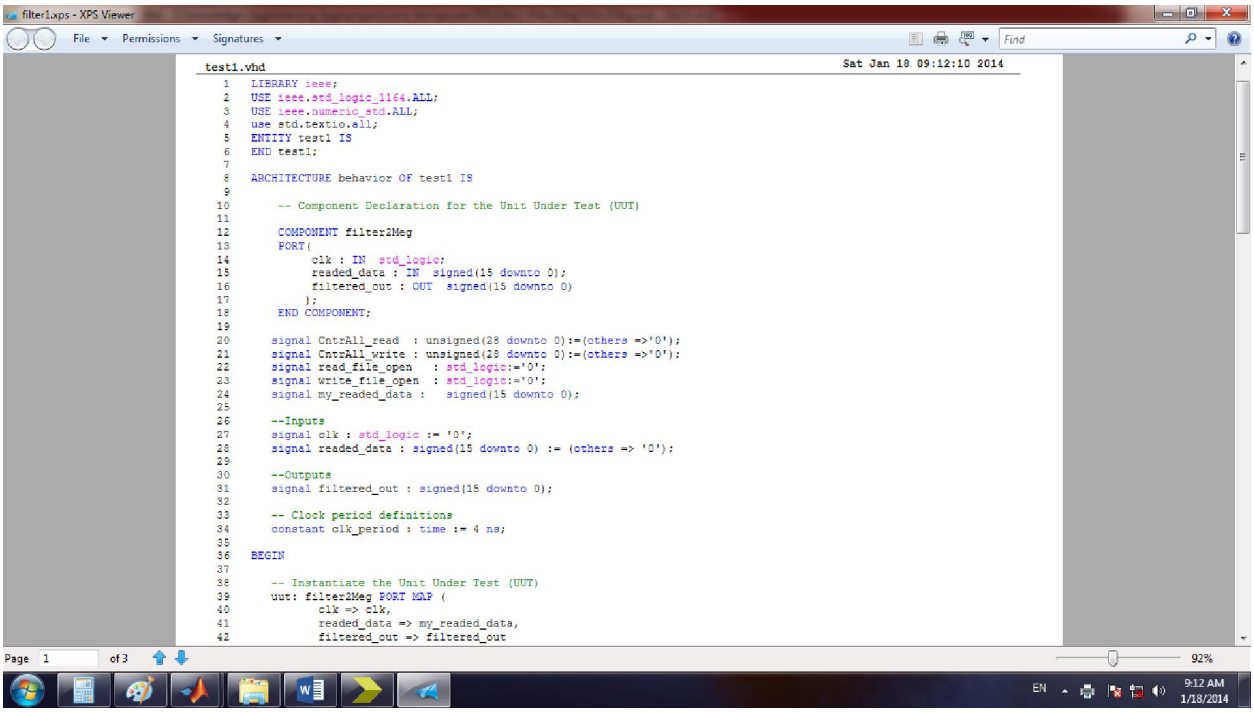

پس از اتمام تنظیمات FIR IP Core ،سورس جدیدی را بعنوان فایل اصلی درست می‌کنیم و همانند زیر FIR IP Core را به آن الحاق می‌کنیم.

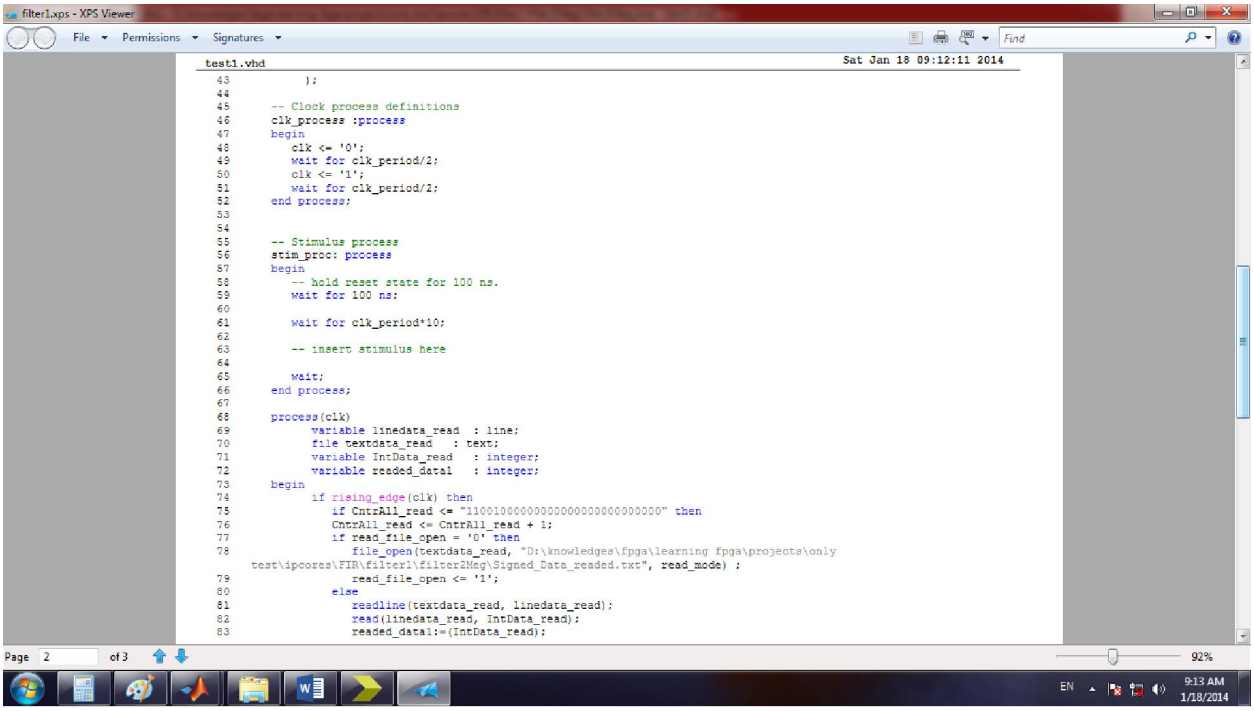



```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  use IEEE.NUMERIC_STD.ALL;
4
5  entity filter2Meg is
6  PORT(
7          clk : IN  std_logic;
8          readed_data : IN  signed(15 downto 0);
9          filtered_out : OUT  signed(15 downto 0)
10        );
11 end filter2Meg;
12
13 architecture Behavioral of filter2Meg is
14
15 signal filtered_out_int : STD_LOGIC_VECTOR(15 DOWNTO 0);
16 COMPONENT filter2
17   PORT (
18     aclk : IN STD_LOGIC;
19     s_axis_data_tvalid : IN STD_LOGIC;
20     s_axis_data_tready : OUT STD_LOGIC;
21     s_axis_data_tdata : IN STD_LOGIC_VECTOR(15 DOWNTO 0);
22     m_axis_data_tvalid : OUT STD_LOGIC;
23     m_axis_data_tdata : OUT STD_LOGIC_VECTOR(15 DOWNTO 0)
24   );
25 END COMPONENT;
26 begin
27 your_instance_name : filter2
28   PORT MAP (
29     aclk => clk,
30     s_axis_data_tvalid => '1',
31     s_axis_data_tready => open,
32     s_axis_data_tdata =>STD_LOGIC_VECTOR(readed_data),
33     m_axis_data_tvalid => open,
34     m_axis_data_tdata => filtered_out_int
35   );
36 filtered_out  <= signed(filtered_out_int);
37 end Behavioral;
```

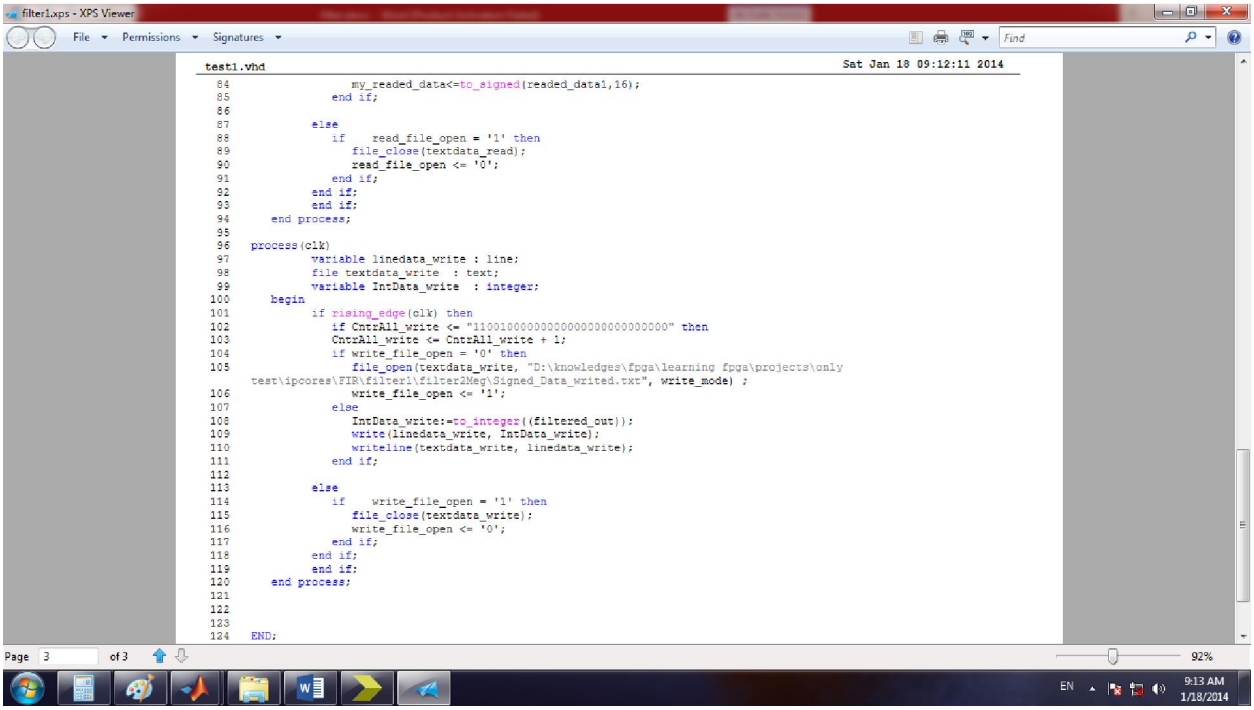
همانطور که مشاهده می‌شود داده ورودی از نوع signed در خط 32 به ورودی FIR IP Core اعمال می‌شود و پس از فیلتر شدن طی خطوط 34 و 36 به خروجی تحویل داده خواهد شد.

در ادامه باید برای داشتن ورودی یعنی گرفتن سیگنال تولید شده در MATLAB و ذخیره داده های فیلتر شده در خروجی FIR IP Core ،یک فایل TEST بصورت زیر درست می‌کنیم.





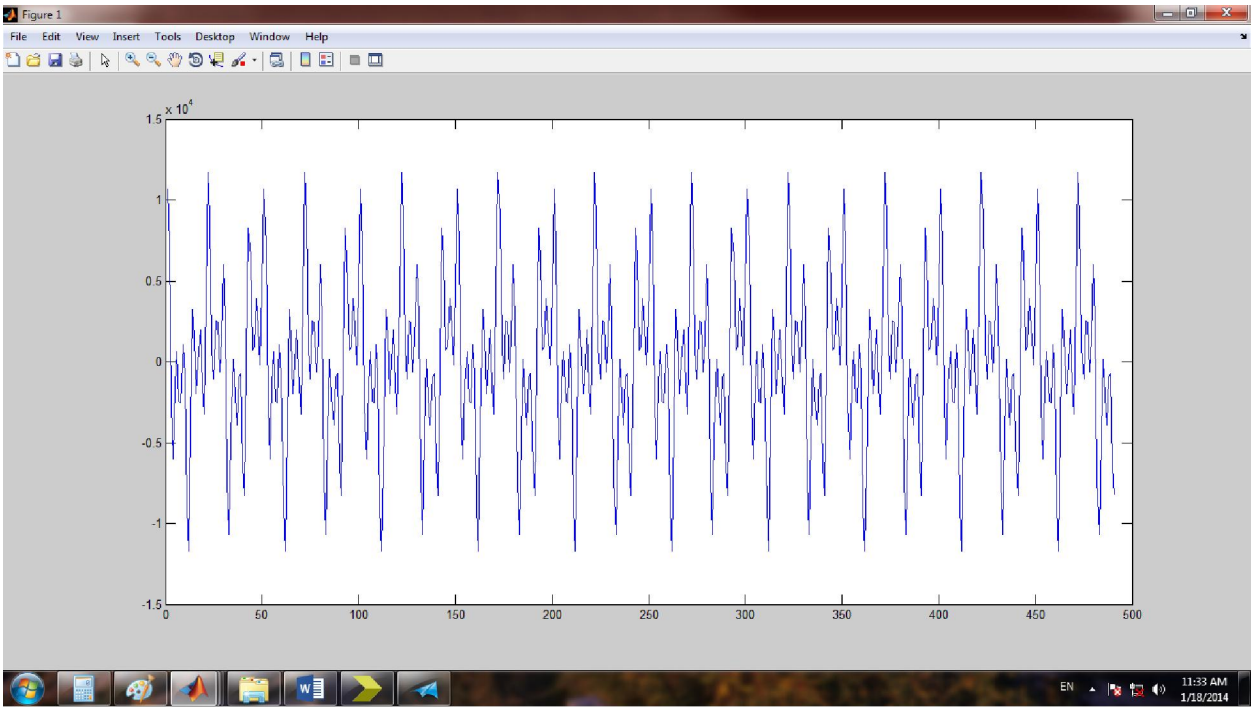

```
43          );
44
45      -- Clock process definitions
46      clk_process :process
47      begin
48         clk <= '0';
49         wait for clk_period/2;
50         clk <= '1';
51         wait for clk_period/2;
52      end process;
53
54
55      -- Stimulus process
56      stim_proc: process
57      begin
58         -- hold reset state for 100 ns.
59         wait for 100 ns;
60
61         wait for clk_period*10;
62
63         -- insert stimulus here
64
65         wait;
66      end process;
67
68      process(clk)
69            variable linedata_read  : line;
70            file textdata_read   : text;
71            variable IntData_read   : integer;
72            variable readed_data1   : integer;
73      begin
74            if rising_edge(clk) then
75               if CntrAll_read <= "11001000000000000000000000000" then
76               CntrAll_read <= CntrAll_read + 1;
77               if read_file_open = '0' then
78                  file_open(textdata_read, "D:\knowledges\fpga\learning fpga\projects\only
     test\ipcores\FIR\filter1\filter2Meg\Signed_Data_readed.txt", read_mode) ;
79                  read_file_open <= '1';
80               else
81                  readline(textdata_read, linedata_read);
82                  read(linedata_read, IntData_read);
83                  readed_data1:=(IntData_read);
```

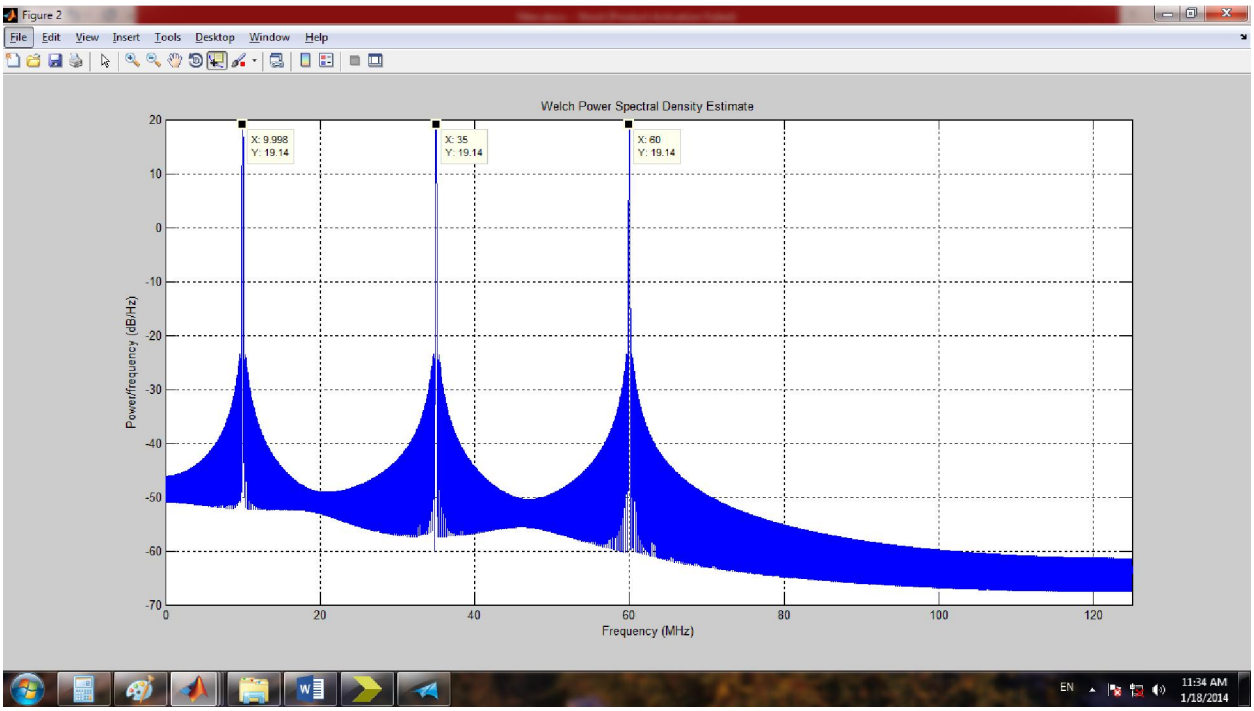


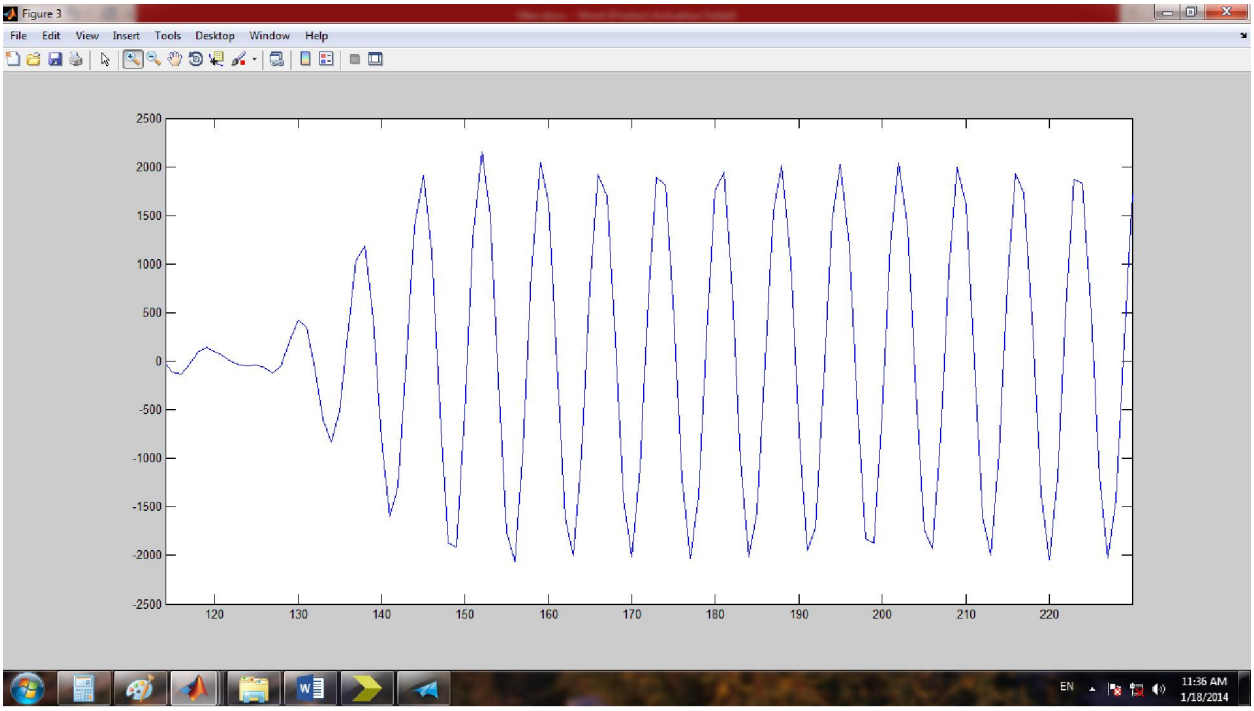

```
84                   my_readed_data<=to_signed(readed_data1,16);
85               end if;
86
87            else
88               if    read_file_open = '1' then
89                  file_close(textdata_read);
90                  read_file_open <= '0';
91               end if;
92            end if;
93            end if;
94      end process;
95
96   process(clk)
97            variable linedata_write : line;
98            file textdata_write  : text;
99            variable IntData_write  : integer;
100     begin
101           if rising_edge(clk) then
102              if CntrAll_write <= "11001000000000000000000000000" then
103              CntrAll_write <= CntrAll_write + 1;
104              if write_file_open = '0' then
105                 file_open(textdata_write, "D:\knowledges\fpga\learning fpga\projects\only
     test\ipcores\FIR\filter1\filter2Meg\Signed_Data_writed.txt", write_mode) ;
106                 write_file_open <= '1';
107              else
108                 IntData_write:=to_integer((filtered_out));
109                 write(linedata_write, IntData_write);
110                 writeline(textdata_write, linedata_write);
111              end if;
112
113           else
114              if    write_file_open = '1' then
115                 file_close(textdata_write);
116                 write_file_open <= '0';
117              end if;
118           end if;
119           end if;
120     end process;
121
122
123
124  END;
```

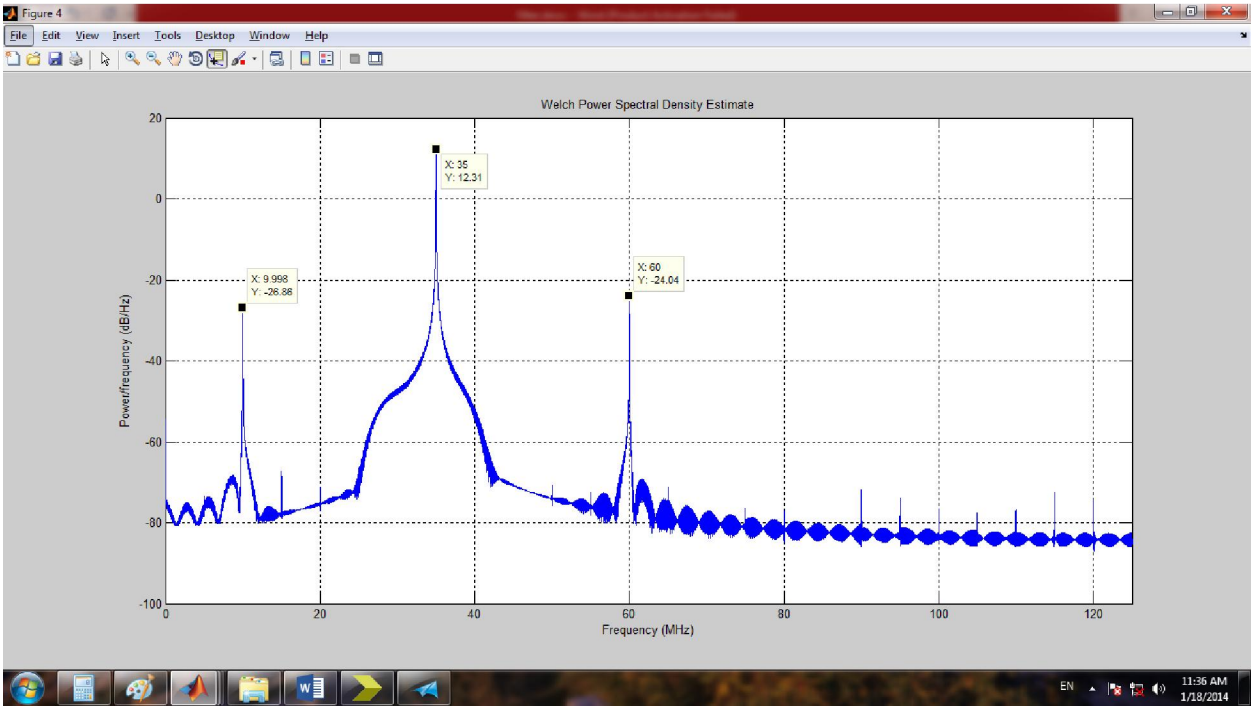
حال شبیه‌سازی را اجرا می‌کنیم تا طی خطوط 74 الی 94 داده‌هایی که توسط MATLAB ذخیره شده‌اند خوانده شوند و با توجه به Port Map به ورودی فایل اصلی تحویل داده شوند و پس از فیلتر شدن در آنجا و خارج شده سیگنال فیلتر شده از خروجی FIR IP Core این سیگنال طی خطوط 96 الی 120 در فایل متنی ذخیره شود.

حال طی خطوط 17 الی 21 در فایل MATLAB ،این داده‌های فیلتر شده در حوزه زمان و فرکانس رسم می‌شوند و مطابق شکل زیر می‌توانیم سیگنال اولیه و سیگنال فیلتر شده را در حوزه زمان و فرکانس با هم مقایسه کنیم.



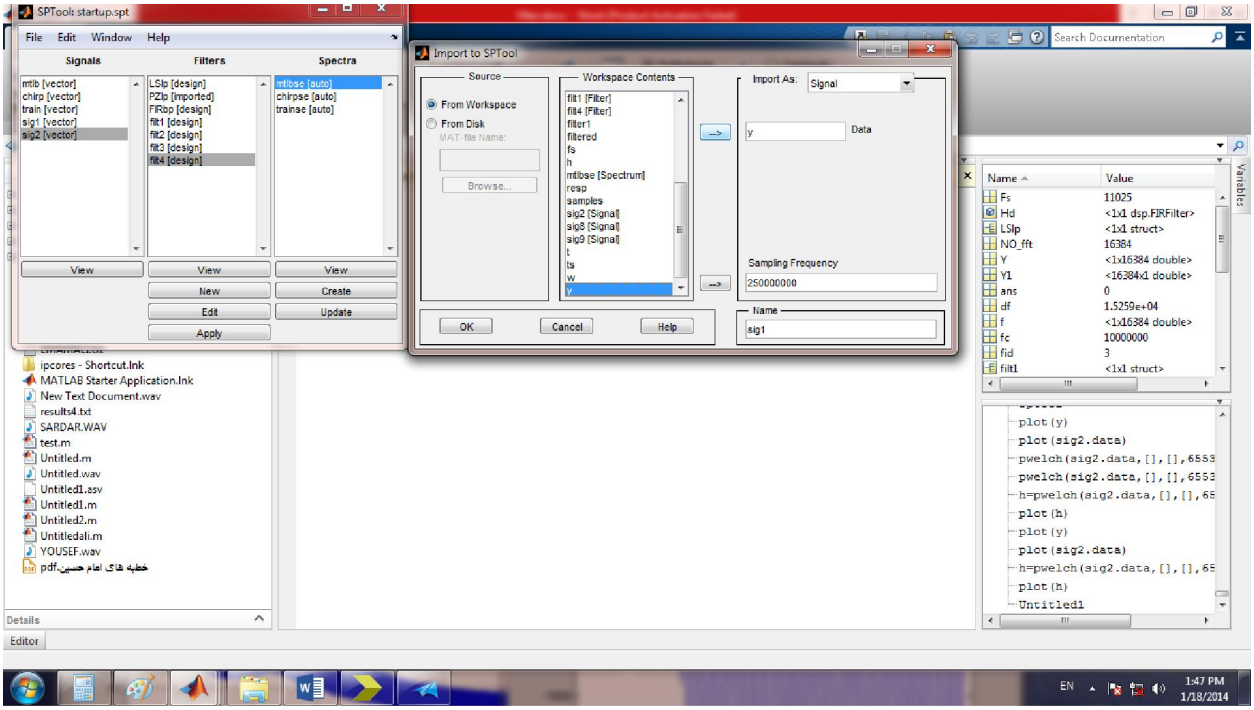



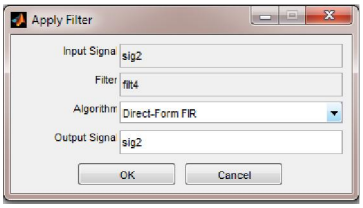


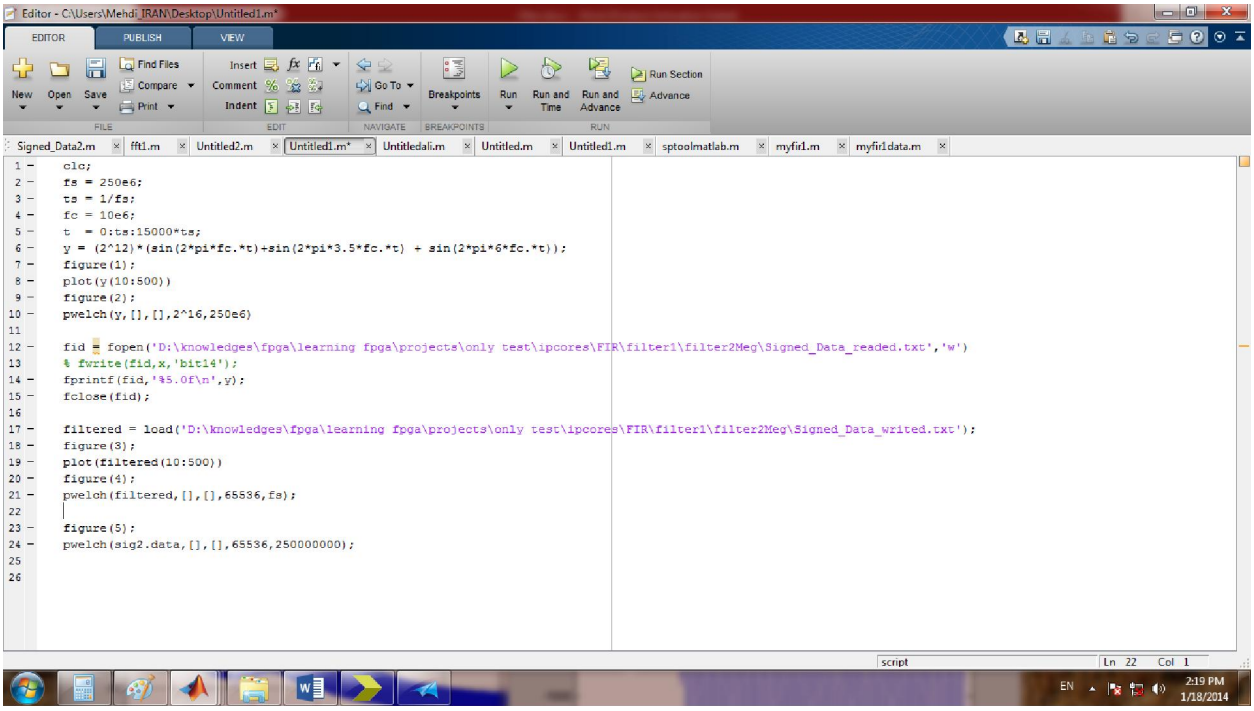

برای اینکه نتایج فیلتر کردن با FPGA را با نتایج فیلتر کردن با MATLAB مقایسه کنیم بصورت زیر از sptool در MATLAB استفاده می‌کنیم.

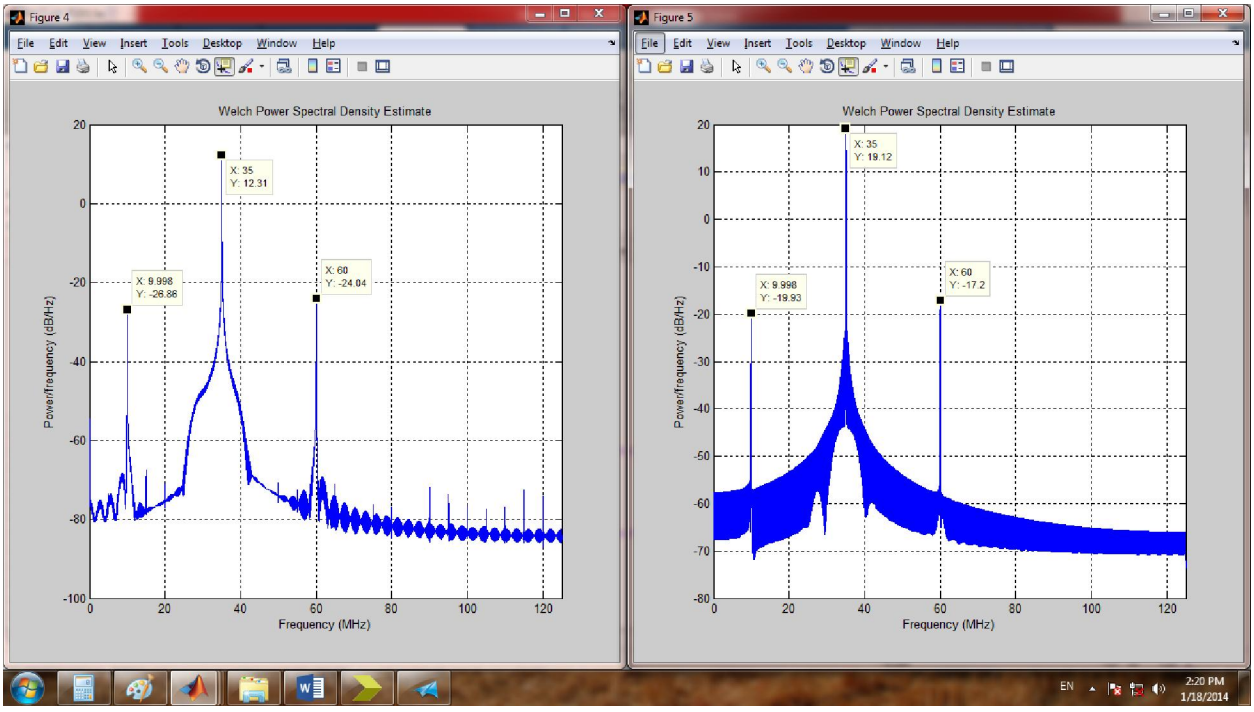
پس از یکبار اجرای برنامه mfile در MATLAB که منجر به تولید و ذخیره مقدار y در work space می‌شود دستور sptool را در MATLAB command window وارد می‌کنیم تا پنجره sptool بازشود. در منوی File آن Import را می‌زنیم و متغیر y را انتخاب می‌کنیم و Sampling Frequency را تنظیم و OKرا می‌زنیم تا مقدار y در sig1 قرار بگیرد. دوباره در قسمت Filters در sptool دکمه new را می‌زنیم تا در محیط fdatool همان مشخصات فیلتری را که در طراحی فیلتر FIR IP Core استفاده شده بود را تنظیم کنیم. پس از زدن design filter و بستن fdatool می‌بینیم که فیلتر جدید در قسمت Filters در sptool با نام filt4 تولید شده است. حال با انتخاب sig1 و filt4 و زدن دکمه Apply نتیجه فیلتر در sig2 ریخته می‌شود.




حال با انتخاب export از منوی File مقادیر sig2 در work space قرار می‌گیرد حال با اضافه کردن کدهای MATLAB زیر طی خطوط 23 و 24 در mfile و اجرای دوباره آن نتایج فیلترکردن در FPGA و MATLAB قابل مشاهده و مقایسه است.


Editor - C:\Users\Mehdi_IRAN\Desktop\Untitled1.m*
EDITOR
PUBLISH
VIEW
New
Open
Save
Find Files
Compare
Print
Insert
Comment
Indent
Go To
Find
Breakpoints
Run
Run and Time
Run and Advance
Run Section
Advance
Signed_Data2.m
fft1.m
Untitled2.m
Untitled1.m*
Untitledali.m
Untitled.m
Untitled1.m
sptoolmatlab.m
myfir1.m
myfir1data.m
1 clc;
2 fs = 250e6;
3 ts = 1/fs;
4 fc = 10e6;
5 t = 0:ts:15000*ts;
6 y = (2^12)*(sin(2*pi*fc.*t)+sin(2*pi*3.5*fc.*t) + sin(2*pi*6*fc.*t));
7 figure(1);
8 plot(y(10:500))
9 figure(2);
10 pwelch(y,[],[],2^16,250e6)
11
12 fid = fopen('D:\knowledges\fpga\learning fpga\projects\only test\ipcores\FIR\filter1\filter2Meg\Signed_Data_readed.txt','w')
13 % fwrite(fid,x,'bit14');
14 fprintf(fid,'%5.0f\n',y);
15 fclose(fid);
16
17 filtered = load('D:\knowledges\fpga\learning fpga\projects\only test\ipcores\FIR\filter1\filter2Meg\Signed_Data_writed.txt');
18 figure(3);
19 plot(filtered(10:500))
20 figure(4);
21 pwelch(filtered,[],[],65536,fs);
22
23 figure(5);
24 pwelch(sig2.data,[],[],65536,250000000);
25
26
script
Ln 22 Col 1
EN
2:19 PM
1/18/2014


Figure 4
File Edit View Insert Tools Desktop Window Help
Welch Power Spectral Density Estimate
X: 35
Y: 12.31
X: 9.998
Y: -26.86
X: 60
Y: -24.04
Power/frequency (dB/Hz)
Frequency (MHz)
Figure 5
File Edit View Insert Tools Desktop Window Help
Welch Power Spectral Density Estimate
X: 35
Y: 19.12
X: 9.998
Y: -19.93
X: 60
Y: -17.2
Power/frequency (dB/Hz)
Frequency (MHz)
EN
2:20 PM
1/18/2014

همانطور که مشاهده می‌شود نتایج تقریبا یکسان است و تفاوت از آنجا است که در FPGA ضرایب فیلتر کوانتیزه شده مورد استفاده قرار گرفته ولی در MATLAB ضرایب فیلتر کوانتیزه نشده استفاده شده است.

چه بسیار علومی که در سینه‌ها ماندند و بهمراه صاحبانشان به زیر خاک رفتند

پس در انتقال دانسته‌های خود به دوستان سستی نکنیم

اصغری

# پایان