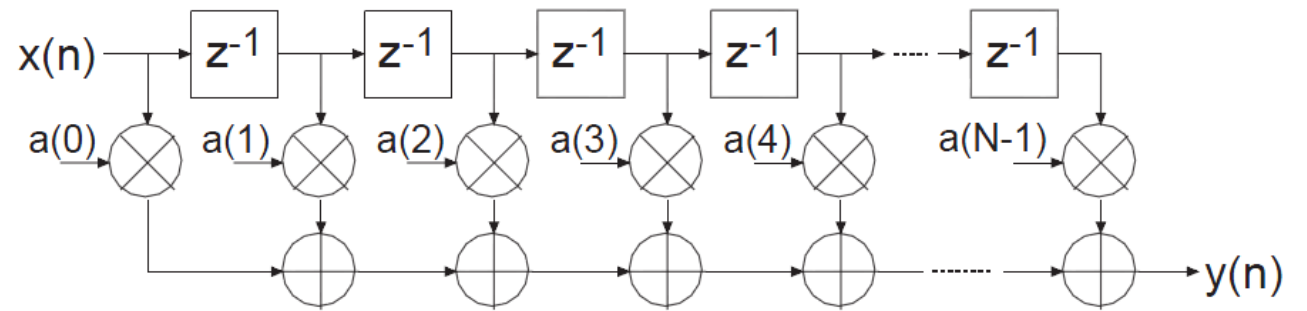


MUNI  
FI

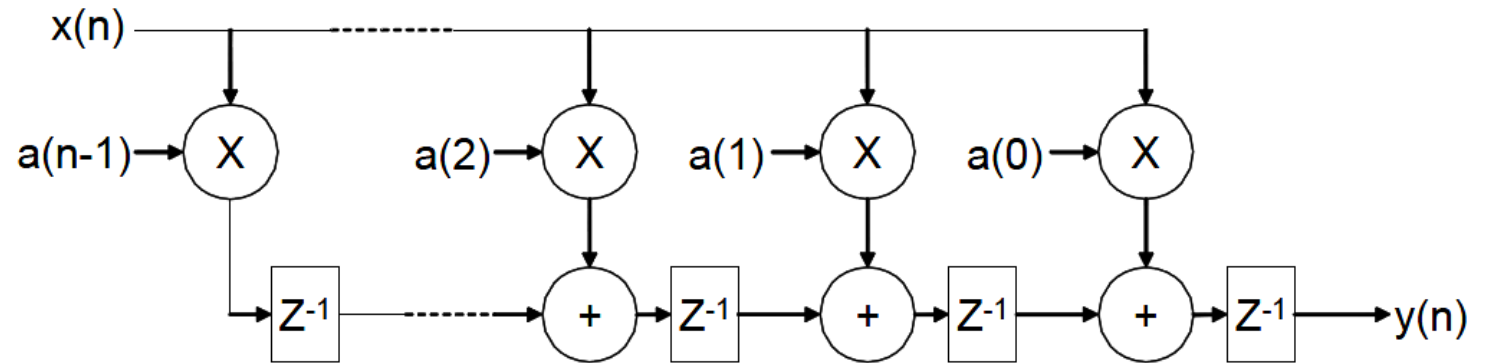
# PA221: FIR Filter Verification

# FIR Filter

$F_{MAX} = ?$



Conventional Tapped Delay Line FIR Filter Representation



Transposed Direct-Form

$$y[n] = a[0] x[n] + a[1] x[n - 1] + \dots + a[N] x[n - N] = \sum_{i=0}^N a[i] x[n - i]$$

# Assignment

## FIR filter design and verification

- **Design** a FIR filter according to specification
  - use Matlab to generate filter coefficients
  - use Vivado IP core wizard to generate FIR filter module
- **Verify** the FIR functionality (automatic testbench)
  - read input data from a file and feed them to the filter
  - read reference output data from a file and compare it with actual filter output
  - report results of the verification to a **log** file, including PASS/FAIL message and total number of errors detected

# FIR Filter Parameters

- Entity (IP core) name FIR\_50k
- Low-pass FIR, Equiripple
- $F_{\text{pass}} = 50 \text{ kHz}$ ,  $F_{\text{stop}} = 200 \text{ kHz}$
- $A_{\text{pass}} = 1 \text{ dB}$ ,  $A_{\text{stop}} = 40 \text{ dB}$
- Sampling frequency 6.25 MHz
- Clock frequency 50 MHz ( $\Rightarrow$  8 clock cycles per sample)
- Input: 9b, no fractional part (range -256 to +255)
- Output: 9b, no fractional part (range -256 to +255)

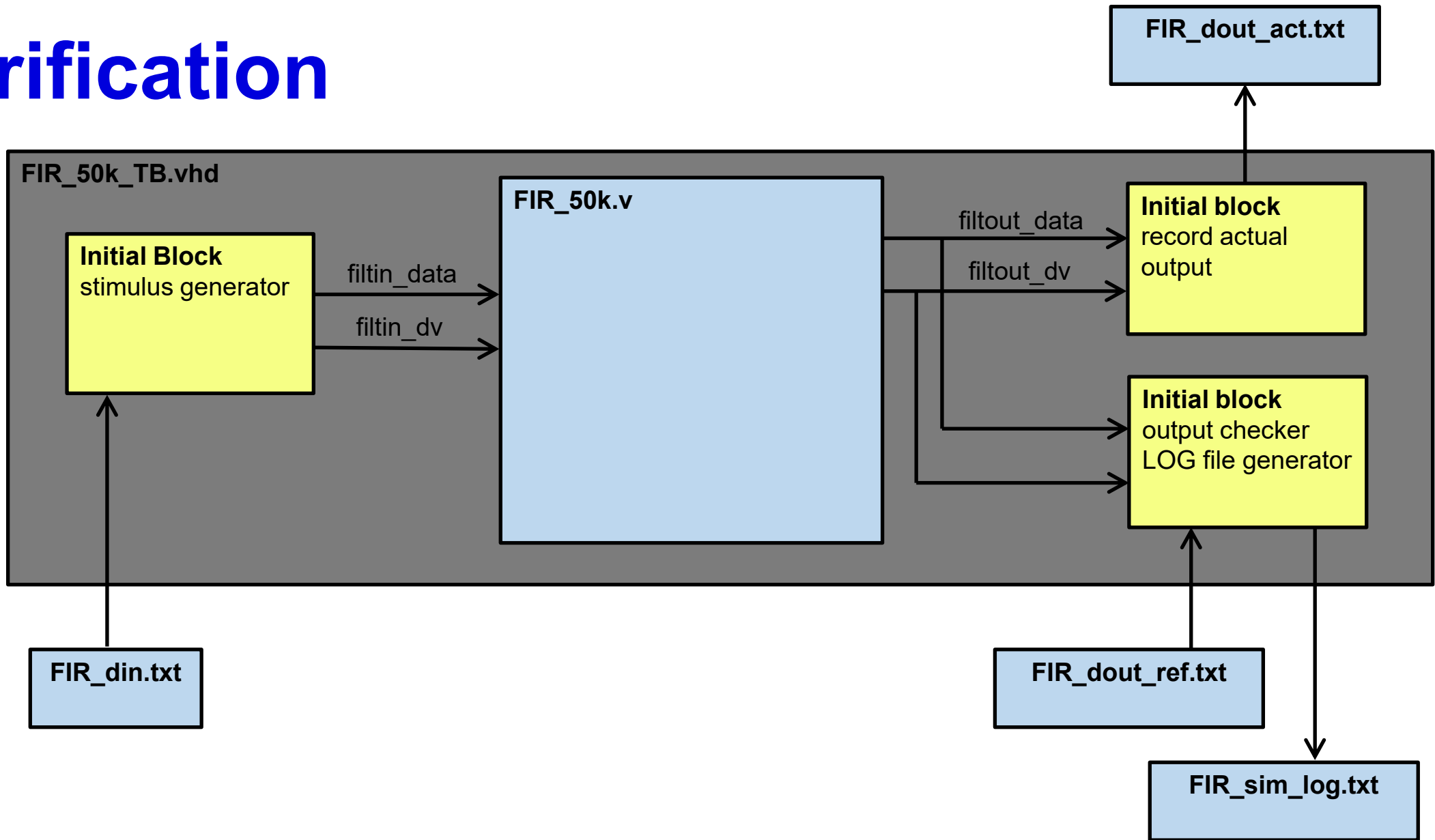
# Verification: Self-testing testbench

- Automatically generates stimuli for the DUT (design under test)
- Automatically verifies correctness of the DUT outputs
- Generates a LOG file with simulation results

# FIR Filter Parameters

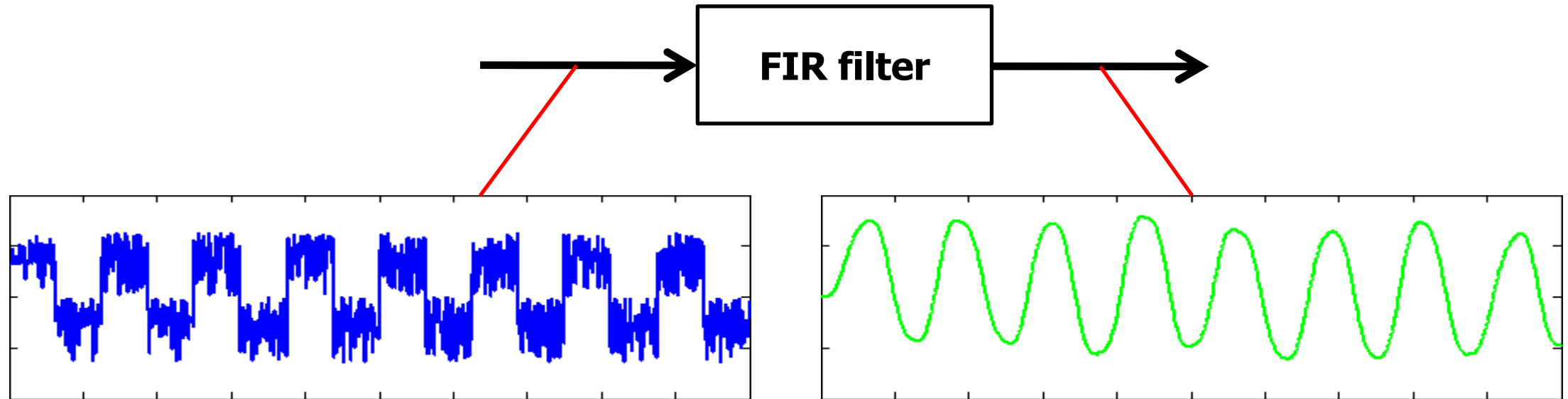
- Instantiate the design under test (DUT; FIR\_50k)
- Create an initial block that reads data from **FIR\_din.txt** file and feeds the data to DUT. Note that there should be one sample for each 8 clock cycles (50 MHz clock at 6.25 MHz sampling frequency).
- Create an initial block that writes data to **FIR\_dout\_act.txt** file whenever FIR output valid signal is asserted. Each sample on a line.
- Create an initial block that reads data from **FIR\_dout\_ref.txt** file and verifies data on the DUT output (whenever FIR output valid signal is asserted). Any discrepancy is reported to both a textual **log** file and simulator console as an error.
- Correctly finish the simulation (do not run forever).

# Verification



# Expected Waveforms

- See the analog waveforms in the ModelSim / Questa.

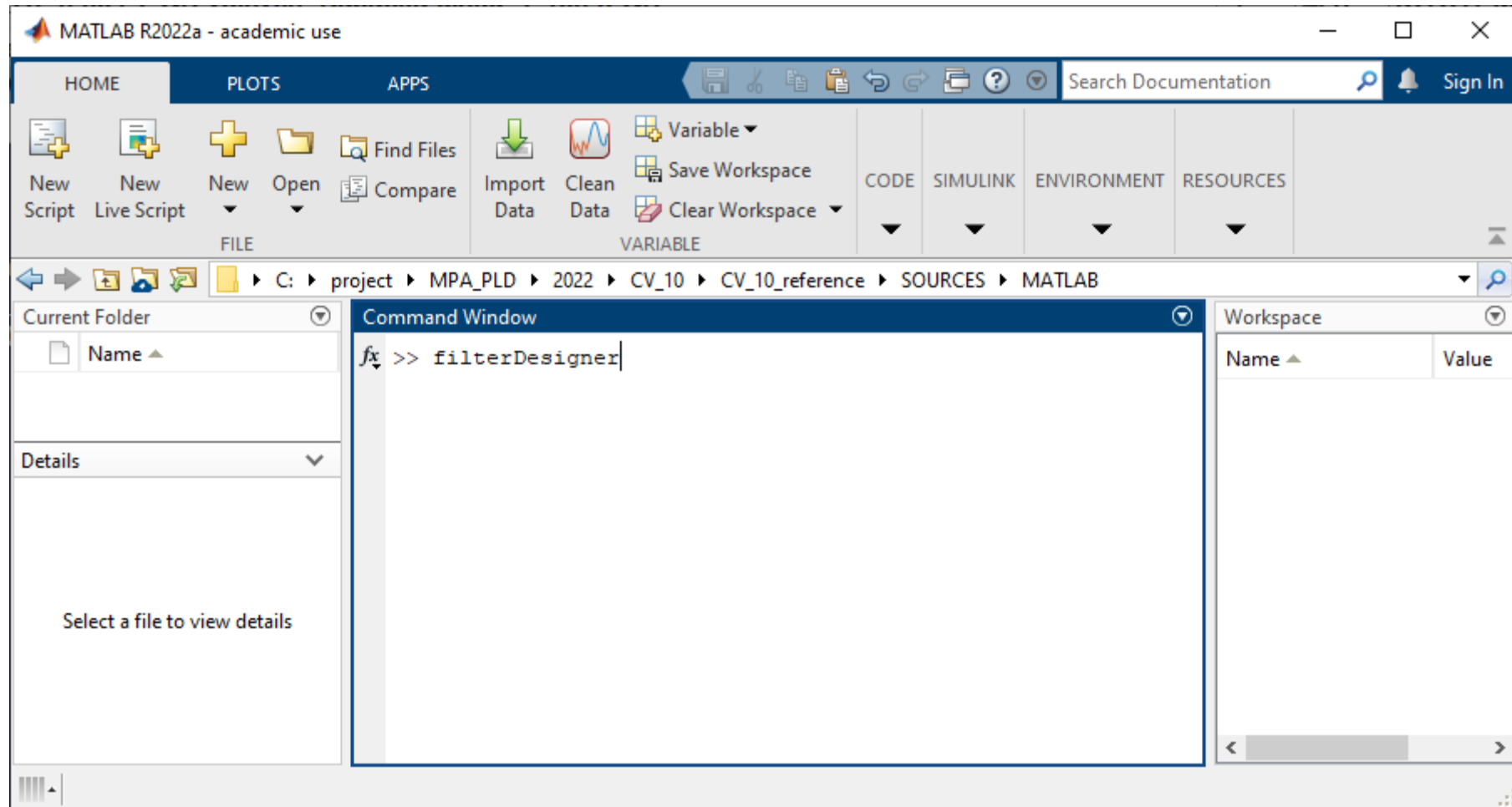




# **FIR filter design: filterDesigner**

Matlab Filter Design & Analysis Tool

# Matlab: Filter Design



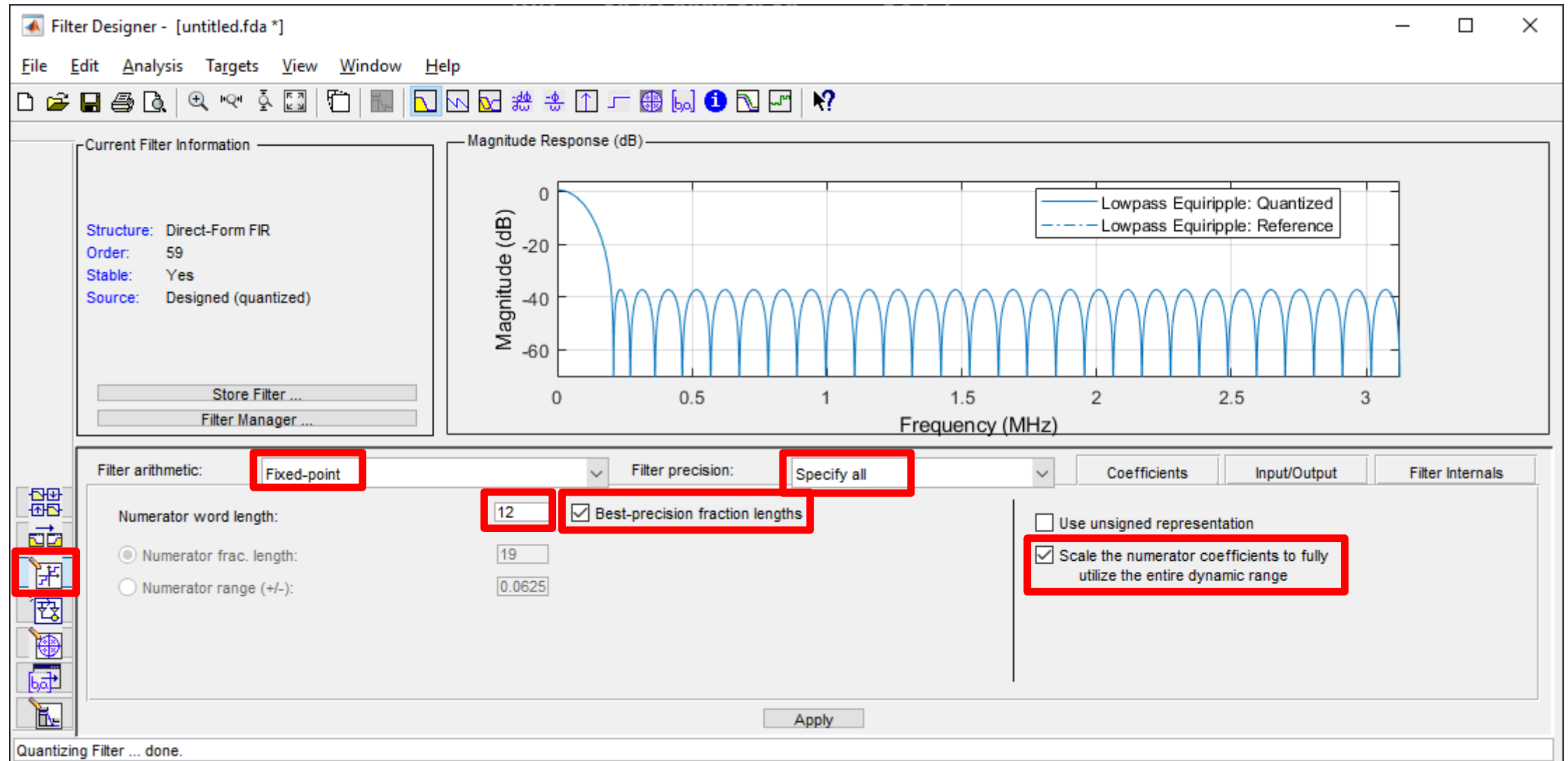
# Matlab: Filter Design

The screenshot displays the MATLAB Filter Designer window with the following components:

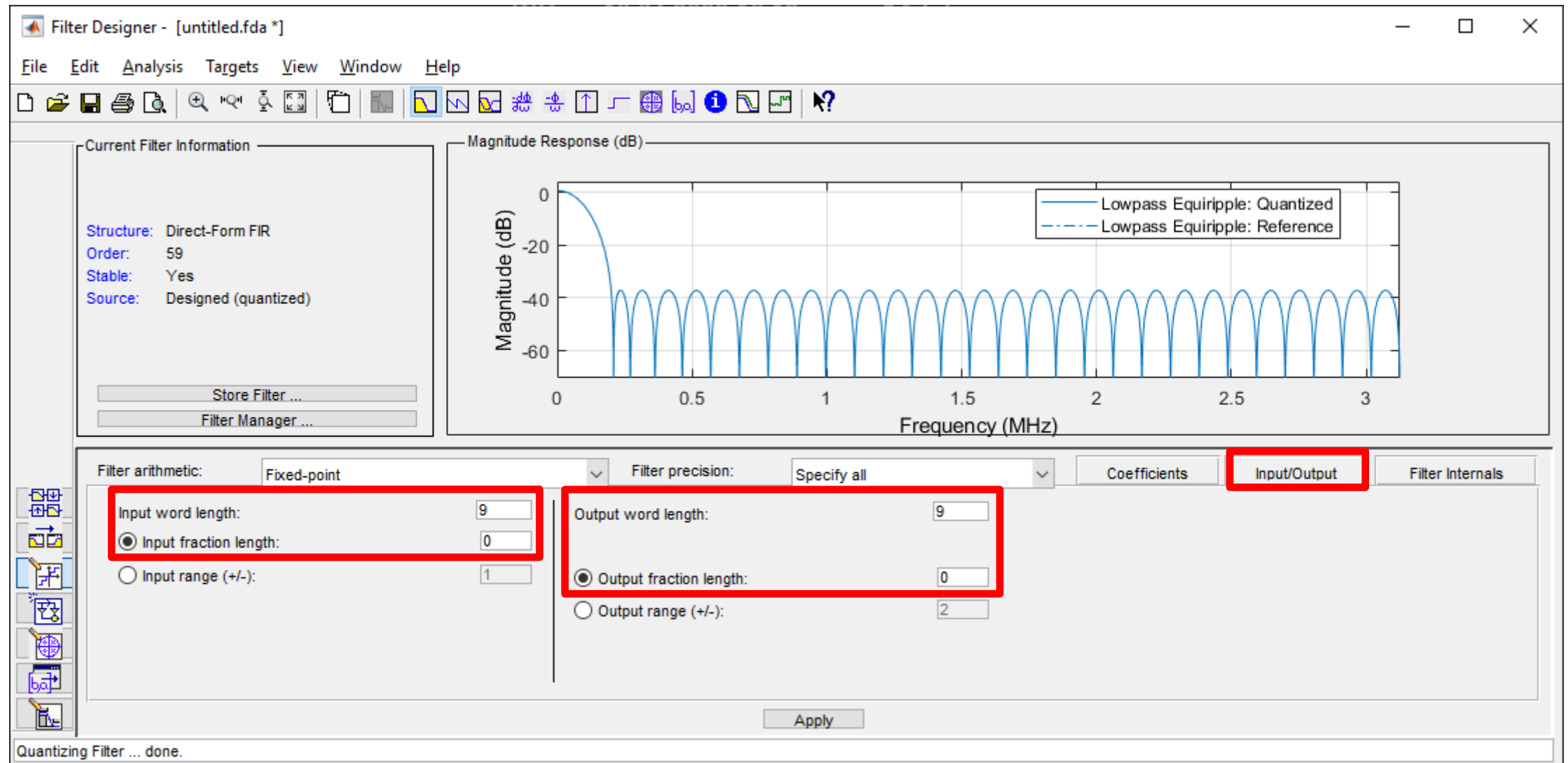
- Current Filter Information:**
  - Structure: Direct-Form FIR
  - Order: 59
  - Stable: Yes
  - Source: Designed
- Magnitude Response (dB):** A plot showing the magnitude response of the filter. The x-axis is Frequency (MHz) from 0 to 3, and the y-axis is Magnitude (dB) from 0 to -60. The plot shows a lowpass filter response with a passband near 0 dB and a stopband starting around 0.1 MHz, reaching approximately -40 dB.
- Response Type:**  Lowpass
- Filter Order:**  Specify order: 10;  Minimum order
- Options:** Density Factor: 20
- Frequency Specifications:**
  - Units: Hz
  - Fs: 6250000
  - Fpass: 50000
  - Fstop: 200000
- Magnitude Specifications:**
  - Units: dB
  - Apass: 1
  - Astop: 40
- Design Method:**  IIR Butterworth;  FIR Equiripple
- Buttons:** Store Filter..., Filter Manager..., Design Filter

Designing Filter ... Done

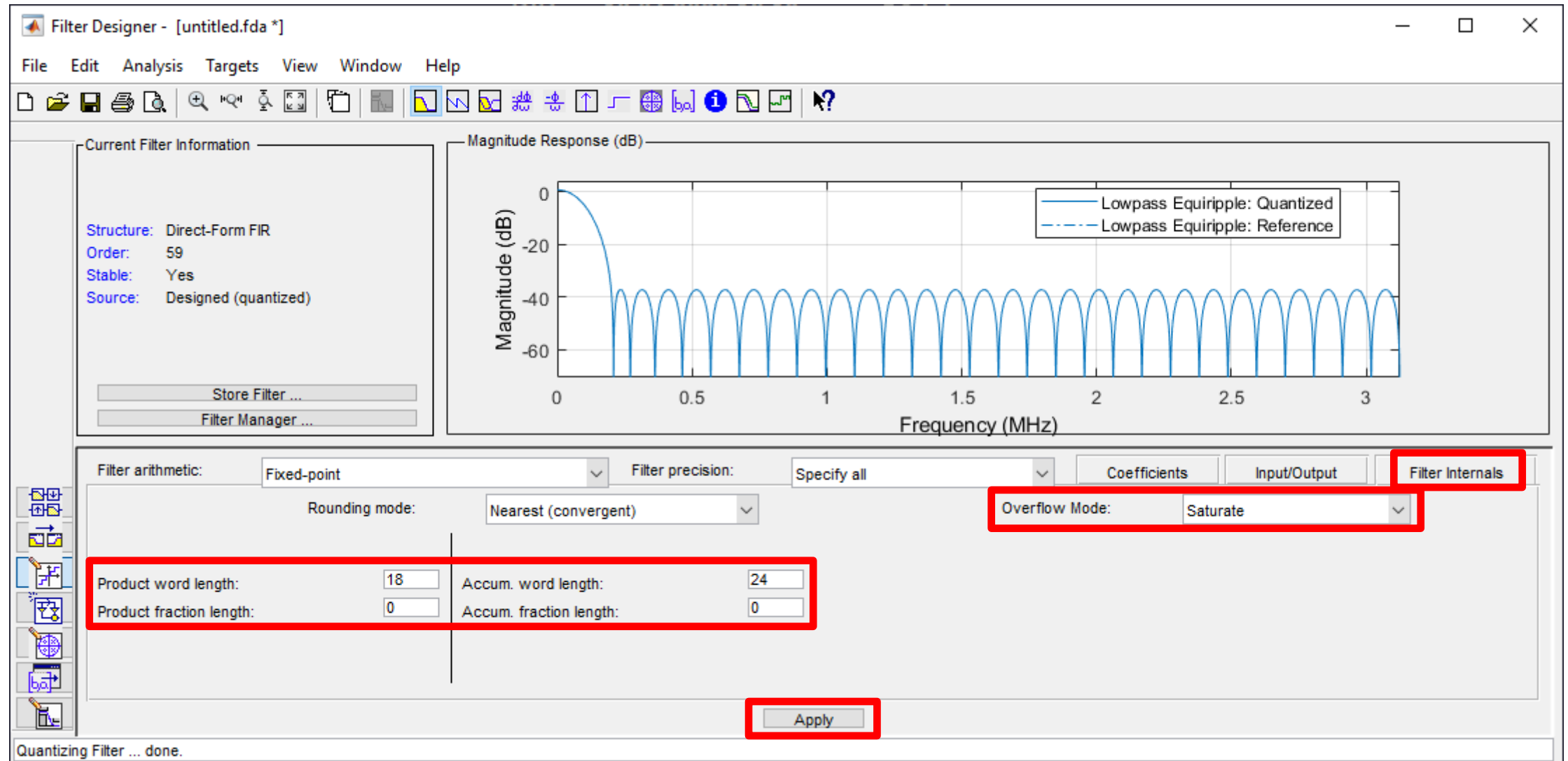
# Matlab: Filter Design



# Matlab: Filter Design



# Matlab: Filter Design

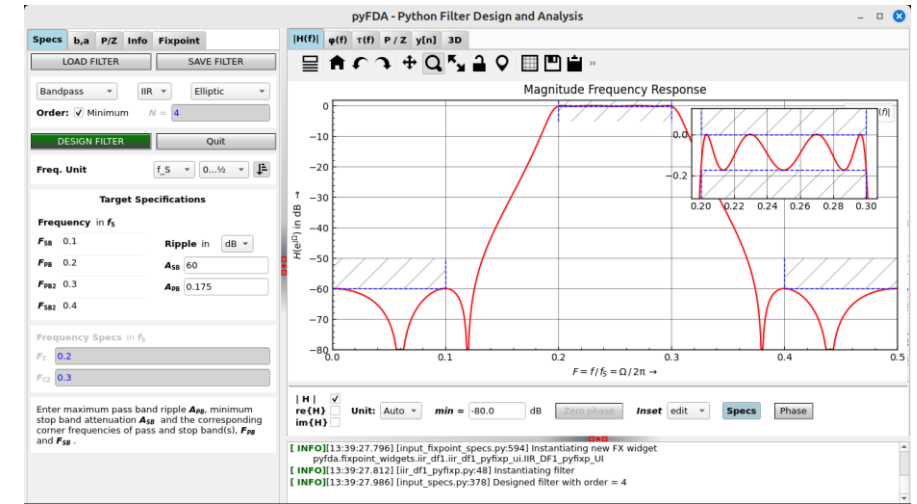


# Matlab: Filter Design

- Export as TXT or CSV file.
- Modify the file into the following format:
  - $c_0, c_1, c_2, \dots$
- If you have no Matlab, you can find the exported file in `w03_filter_template/sources/matlab/FIR_50k.fcf`

# Alternative Tools: Filter Design

- pyfda (<https://github.com/chipmuenk/pyfda>)
- T-Filter – online  
(<http://t-filter.engineerjs.com/>)
- FIR Filter Designer - online (<https://wirelesslibrary.labs.b-com.com/FIRfilterdesigner/#/>)
- GNU Octave (<https://www.allaboutcircuits.com/technical-articles/design-of-fir-filters-design-octave-matlab/>)
- rePhase (<https://www.minidsp.com/applications/advanced-tools/rephase-fir-tool>)





# **FIR II IP Core**

Quartus IP Core Library

# IP Core

Quartus Prime Lite Edition - C:/Users/jakral/ownCloud/MUNI/vyuka/PV221/Cviceni/w03\_filter/w03\_filter - w03\_filter

File Edit View Project Assignments Processing Tools Window Help

Search Intel FPGA

Project Navigator: Hierarchy

Entity: Instance

Cyclone V: 5CSEMA5F31C6

w03\_filter

filter:filter\_inst

Tasks: Compilation

Task: Compile Design, Analysis & Synthesis, Fitter (Place & Route)

Table of Contents: Flow Summary, Flow Settings, Flow Non-Default Global Settings, Flow Elapsed Time, Flow OS Summary, Flow Log, Analysis & Synthesis, Fitter, Assembler, Timing Analyzer, EDA Netlist Writer, Flow Messages, Flow Suppressed Messages

Flow Summary

Flow Status	Successful -
Quartus Prime Version	23.1std.0 Bu
Revision Name	w03_filter
Top-level Entity Name	w03_filter
Family	Cyclone V
Device	5CSEMA5F3
Timing Models	Final
Logic utilization (in ALMs)	1 / 32,070 (<
Total registers	0
Total pins	37 / 457 (8 9
Total virtual pins	0
Total block memory bits	0 / 4,065,280
Total DSP Blocks	0 / 87 (0 %)
Total HSSI RX PCSs	0

IP Catalog

Library

- Basic Functions
- DSP
  - Error Detection and Correction
  - Filters
    - FIR II**
  - Floating Point
  - Signal Generation
  - Transforms
- Interface Protocols
- Memory Interfaces and Controllers
- Processors and Peripherals

Messages

Type	ID	Message
1	22036	For messages from NativeLink execution see the NativeLink log file c:/Users/jakral/ownCloud/MUNI/vyuka/PV221/Cviceni/w03_filter/w03_
1	22036	Successfully launched NativeLink simulation (quartus_sh -t "c:/intelfpga_lite/23.1std/quartus/common/tcl/internal/nativeLink/qnative
1	22036	For messages from NativeLink execution see the NativeLink log file c:/Users/jakral/ownCloud/MUNI/vyuka/PV221/Cviceni/w03_filter/w03_
1	22036	Successfully launched NativeLink simulation (quartus_sh -t "c:/intelfpga_lite/23.1std/quartus/common/tcl/internal/nativeLink/qnative
1	22036	For messages from NativeLink execution see the NativeLink log file c:/Users/jakral/ownCloud/MUNI/vyuka/PV221/Cviceni/w03_filter/w03_

System (36) Processing (185)

100% 00:00:41

# IP Core

Expected file format:

c0, c1, c2, ...

The screenshot shows the 'FIR II - filter' configuration tool. The 'Block Diagram' tab on the left shows a 'filter' block with three inputs: 'clk' (clock), 'rst' (reset), and 'avalon\_streaming\_sink' (avalon\_streaming). The 'Coefficients' tab on the right displays a 'Frequency Response (Magnitude)' graph. The graph shows a magnitude response in dB versus frequency in MHz, with a passband ripple between approximately 0.25 MHz and 3.25 MHz. The 'Import from file' button is highlighted with a red box. Below the graph is a table of coefficients:

Coeff No.	Original Value	Scaled Value	Fixed Value
0	-0.158203125	-0.1577918906	-323
1	0.0673828125	0.0669272106	137
2	0.0693359375	0.0688812897	141
3	0.080078125	0.079628725	163

At the bottom of the tool, an information bar displays: 'Info: filter: PhysChanIn 1, PhysChanOut 1, ChansPerPhyIn 1, ChansPerPhyOut 1, OutputFullBitWidth 27, Bankcount 1, CoefBitWidth 12'. The 'Cancel' and 'Finish' buttons are visible at the bottom right.

# IP Core

The screenshot shows a configuration window for an IP Core with several tabs: Filter Specification, Coefficient Settings, Coefficients, Input/Output Options (selected), Implementation Options, and Reconfigurability. The window is divided into two main sections: Input Options and Output Options.

**Input Options:**

- Input Type: Signed Binary (dropdown)
- Input Width: 9 bits (text input, highlighted with a red box)
- Input Fractional Width: 0 bits (text input)

**Output Options:**

- Output Type: Signed Binary (dropdown)
- Output Width: 9 bits (text input)
- Output Fractional Width: 0 bits (text input)
- Specifies whether to truncate or saturate the most significant bit (MSB): Truncation (dropdown)
- MSB Bits to Remove: 2 (text input, highlighted with a red box)
- Specifies whether to truncate or round the least significant bit (LSB): Truncation (dropdown)
- LSB Bits to Remove: 16 (text input, highlighted with a red box)
- outWidth: 27 bits (text input)
- Output Full Fractional Width: 0 bits (text input)

# IP Core

- Calculate number of expected DSP blocks:
  - 1) for non-symmetrical coefficients
  - 2) for symmetrical coefficients
- Verify the calculation on Implementation Options tab.

# Verilog: Reading Data from a File

```
integer fid_din;
reg [256:0] line; // buffer for line read from file

initial begin
    // open the file
    fid_din = $fopen("../../sources/data/FIR_din.txt", "r");
    if (fid_din == 0) begin
        $display("ERROR: File not found.");
        $finish;
    end

    // check EOF and try to read a line from the file
    while (!$feof(fid_din) && $fgets(line, fid_din) != 0) begin
        if ($sscanf(line, "%d", data) != 1) begin
            $display("ERROR: Failed to scan number from line.");
            $finish;
        end
    end
    $fclose(fid_din);
end
```

# Verilog: Writing Data to a File

```
integer fid_dout_act;

initial begin
    // open the file for writing
    fid_dout_act = $fopen("../../sources/data/FIR_dout_act.txt", "w");
    if (fid_dout_act == 0) begin
        $display("ERROR: File could not be opened.");
        $finish;
    end

    while (!sim_finished) begin
        // check if the data is valid
        if (dv) begin
            // write the data to file
            $fdisplay(fid_dout_act, "%d", data); // internally adds new line
        end
        #(CLK_PER); // wait for one clock period
    end

    $fclose(fid_dout_act); // close the file
end
```