

# Design of FIR Low Pass Filter using Vaman

Priyanka - EE21MTECH12002

Download all codes from

[https://github.com/PeriPriyanka/FPGA\\_5811/Project/codes](https://github.com/PeriPriyanka/FPGA_5811/Project/codes)

## 1 DESIGN OF FIR FILTER USING VAMAN

FIR filter are Finite impulse response filters. The impulse response of the FIR is for finite duration of time. The filter of infinite impulse response needs to truncate to finite impulse response. One of the methods for truncation is applying a window. Windowing can be considered as a sort of amplitude envelope that gets multiplied with infinite impulse response to give finite impulse response.

$$h(n) = h_d(n) * w(n) \quad (1.0.1)$$

Where,  $h(n)$  is finite impulse response,  $h_d(n)$  is infinite impulse response,  $w(n)$  is window function.

### 1.1 Steps to design a low pass filter

- 1) Design an Infinite impulse response low pass filter.

$$h_d(n) = \begin{cases} \frac{\sin(w_c n)}{\pi n} & n = -\frac{(N-1)}{2} \text{ to } \frac{N-1}{2} \\ \frac{w_c}{\pi} & n = 0 \end{cases} \quad (1.1.1)$$

or

$$h_d(n) = \begin{cases} \frac{\sin(w_c(n-\tau))}{\pi(n-\tau)} & n = 0 \text{ to } (N-1) \\ \frac{w_c}{\pi} & n = \tau \end{cases} \quad (1.1.2)$$

where center of symmetry,  $\tau = (N-1)/2$  and  $w_c$  is digital cutoff frequency.

- 2) Design of a Hamming window

$$w_h(n) = \begin{cases} 0.54 - 0.46 \cos(\frac{2\pi n}{N-1}) & n = 0 \text{ to } N-1 \\ 0.54 + 0.46 \cos(\frac{2\pi n}{N-1}) & n = -\frac{(N-1)}{2} \text{ to } \frac{N-1}{2} \end{cases} \quad (1.1.3)$$

- 3) Design a finite impulse response low pass filter

$$h(n) = h_d(n) * w_h(n) \quad (1.1.4)$$

## 2 DESIGN OF FILTER IN VAMAN

An FIR Low pass filter designed by accessing the cutoff frequency from ESP32. ESP32 accesses the cutoff frequency from web server through WiFi module and passes the data to arm. Arm has been coded in a way to access the data from ESP32, produce the filter coefficients and returns the data to ESP32 via UART Serial communication. The ESP32 then sends the received information of filter coefficients to web server with the help of WiFi module.

### 2.1 Coding in ESP32

Get the web server code of ESP32 from

[https://github.com/PeriPriyanka/FPGA\\_5811/Project/codes/ESP32/src/main.cpp](https://github.com/PeriPriyanka/FPGA_5811/Project/codes/ESP32/src/main.cpp)

- Input to the ESP32 is given through WiFi module by web server. To run ESP32 by web server, download the web server code from above link.
- Run the command "pio run" for compilation in ESP32 directory. Check whether all the necessary files like "platformio.ini" are included in the folder ESP32. Include the necessary libraries, if needed for web server compilation.
- After compilation, for flashing the code, ESP32 of vaman should be flashed with UART. Take a USB-UART connector.
- Connect the ground,  $T_x$  and  $R_x$  of USB-UART to ground,  $R_x$  and  $T_x$  of Vaman respectively.
- The I0 pin and EN pin are ground before flashing. While flashing remove EN pin from ground and keep it floating.
- The code will be flashed. After flashing connect the I0 pin to vcc.
- To observe the output in serial monitor, open the serial monitor and IP address of ESP32 is observed on the serial monitor.
- Copy the IP address and paste in the browse. The server interface will be opened.

## 2.2 Coding in ARM

Get the web server code of ESP32 from

[https://github.com/PeriPriyanka/FPGA\\_5811/Project/codes/ARM/src/main.c](https://github.com/PeriPriyanka/FPGA_5811/Project/codes/ARM/src/main.c)

- Write the specified code in main.c for specific application.
- For compilation of ARM code, move to GCC\_Project directory and run "make" command. After compilation of code flashing along with top.bin, use the command:

```
python3 /Users/Priyanka/TinyFPGA-
Programmer-Application/tinyfpga-
programmer-gui.py --port /dev/cu.
usbmodem14101 --m4app /Users/
Priyanka/blink.bin --appfpga /Users/
Priyanka/top.bin --mode m4-fpga
```

- Replace "/Users/Priyanka" with the respective directory where all files are downloaded.
- Put the vaman in the boot mode for flashing. After flashing reset the vaman.

## 2.3 Interface between ESP32 and ARM

- For interfacing ESP32 and ARM, UART serial communication is used.
- Connect ground,  $T_x$  and  $R_x$  of ESP32 are connected to ground,  $R_x(45)$  and  $T_x(44)$  of the pygmy pins respectively.

### 3 RESULTS

- The interface of the server looks like below for a given IP address of ESP32. Give the cutoff value of the filter as input. Click submit button. The cutoff frequency is taken by ESP32 through WiFi and transfers it ARM via UART.
- The interface is as shown below.

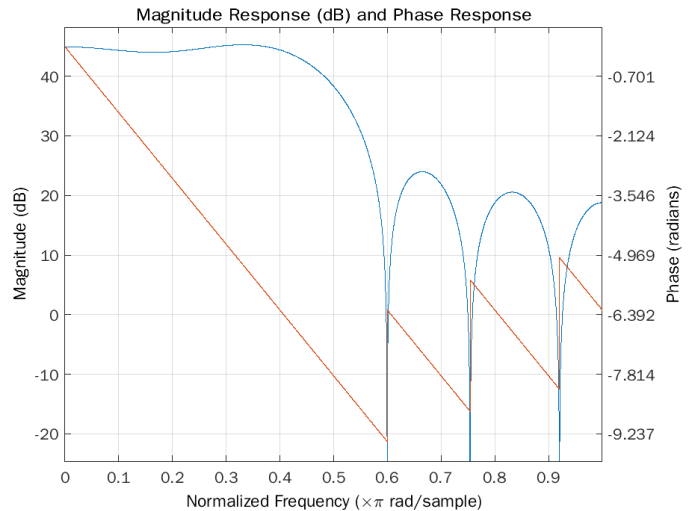
ESP Input Form
×
+

← → ↻
Not Secure | 192.168.134.237

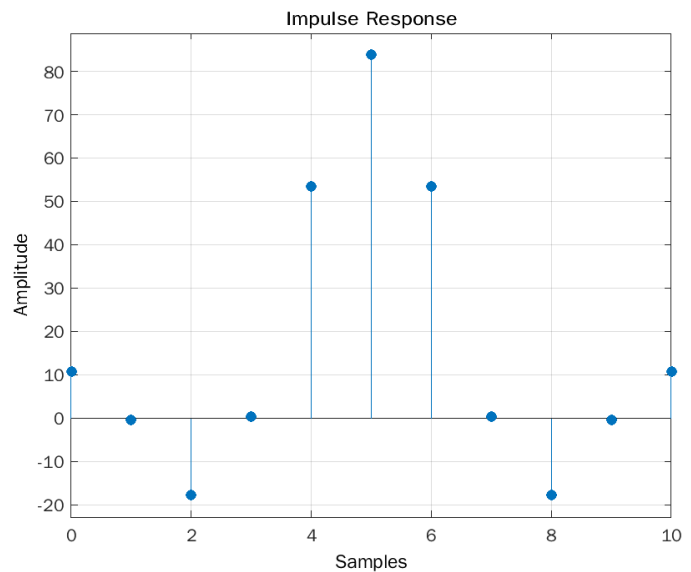
cutoff of the filter:

Output = 10.68 -0.4 -17.80 0.4 53.42 83.88 53.42 0.4 -17.80 -0.4 10.68 x

- Plotting the magnitude response, phase response and impulse response of the filter coefficients. The responses are shown below.
- The magnitude and phase response show the design filter is a Low pass filter.



- Blue gives magnitude response and red gives phase response.



### REFERENCES

- [1] John G. Proakis and Dimitris G. Manolakis, "Digital Signal Processing, Principles, Algorithms and Applications", 3<sup>rd</sup> edition.