



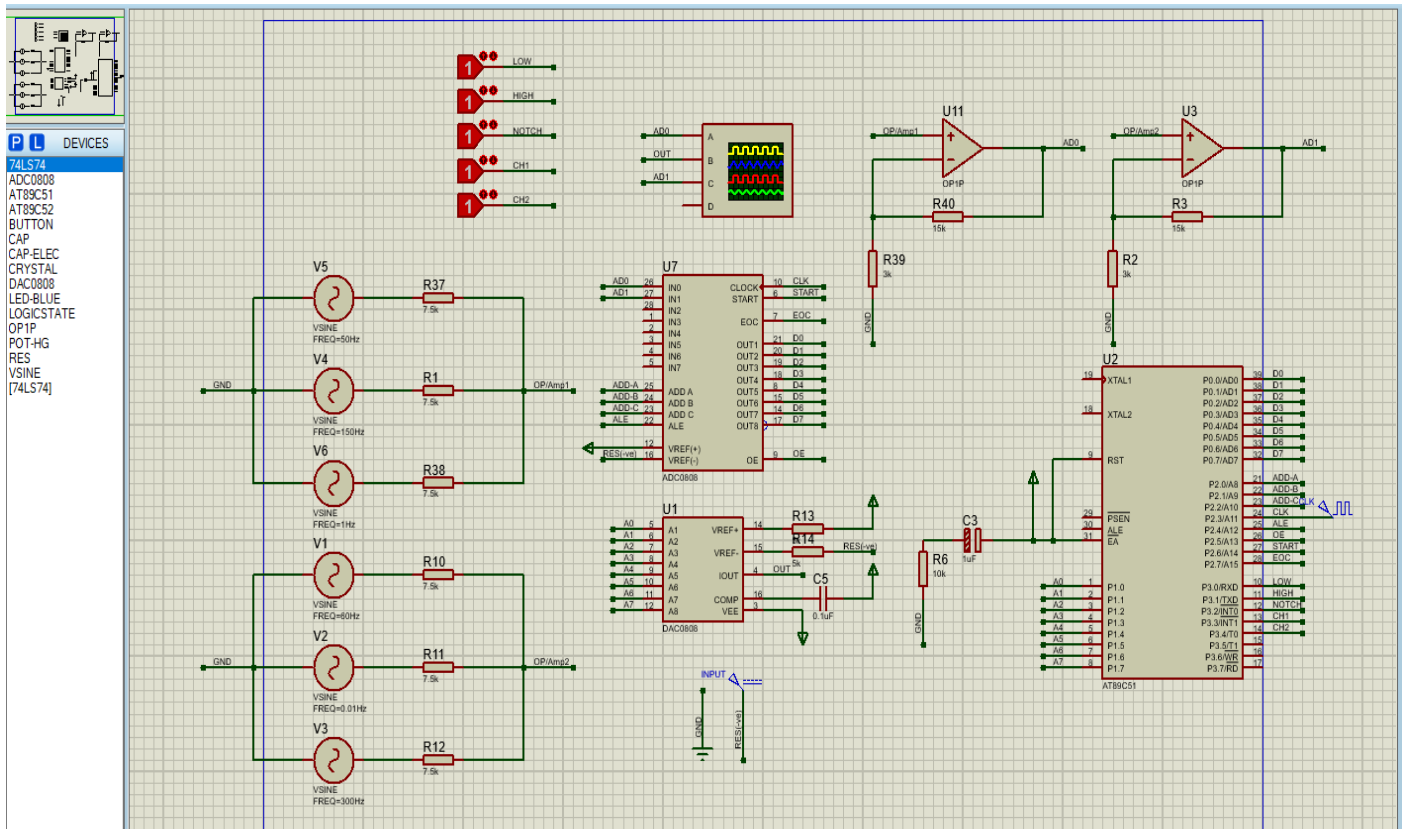
Research Project (Digital Filtering using C8051F020)

Team Number: 4

Team Members:

| Name | Section | BN | Email |
|-------------------------------|---------|----|--|
| Gehad Mohamed Ahmed | 1 | 27 | gehad.mohamed887@gmail.com |
| AmanyBahaa El-Din | 1 | 18 | amanyb658@gmail.com |
| Samar AlaaSobhyTaher | 1 | 41 | smrtaher3@gmail.com |
| Mustafa Yehia Mohamed sayed | 2 | 33 | Mustafayehia4@gmail.com |
| Mustafa Mohamed Tufik Mahmoud | 2 | 32 | Mostafa.mohamed15397@gmail.com |

Circuit schematic:



Design description:

- Input signal → User have up to two channels to select which input will be entered to ADC
 - Weighted Summer Signal includes different frequencies (minimum frequency = 1 Hz, then 50 Hz and the maximum frequency = 150 Hz),
 - Weighted Summer Signal includes different frequencies (minimum frequency = 0.01 Hz, then 60 Hz and the maximum frequency = 300 Hz)
- Oscilloscope displays the input signal, which gets into the ADC in pin0/pin1 (in our simulation) through the address lines (ADD-A, ADD-B, ADD-C).
 - ADD-A is LSB
 - ADD-C is MSB
 - When ADD-A, ADD-B and ADD-C all equals to 0 this means we select the first channel as input to ADC.
 - When ADD-A = 1, ADD-B = 0 and ADD-C = 0 this means we select the second channel as input to ADC.

- Clock → Starts with getting the input signal into ADC (Enter the signal in ADC takes one clock).
 - o We used an external clock for 8051 to speed up its operation (500 KHz).
- ADC Mode → Start of Conversion then (High-Low, High-Low) then End of Conversion.
- Output-Enable → Is set to High to read digital data from ADC which is sent to DAC from 8051 micro-controller after filtration.
- Filtering → occurs when digital signal data get into micro-controller using FIR filters, then the filtered signal data get out from pins of micro-controller to pins of DAC.
- Oscilloscope → the input and the output signal are viewed through the Oscilloscope.

Description for the digital filters:

We used the FIR (Finite Impulse Response) digital filter for filtration of the digital signal, which gets out from ADC then gets into micro-controller 8051, If you put in an impulse, that is, a single (1) sample followed by many (0) samples, zeroes will come out after the (1) sample has made its way through the delay line of the filter. Why did we use FIR digital filter for our desired filtration? Because its impulse response is finite as a result of there is no feedback in the FIR. A lack of feedback guarantees that the impulse response will be finite. The process of selecting the filter's length and coefficients is called filter design. The goal is to set those parameters such that certain desired stopband and passband parameters will result from running the filter. The impulse response of an Nth-order discrete-time FIR filter lasts for N+1 samples. A higher order means better response. For a discrete-time FIR filter, the output is a weighted sum of the current and a finite number of previous input values. The operation is described by the following equation, which defines the output sequence $y[n]$ in terms of its input sequence $x[n]$:

$$y[n] = (b_0 * x[n] + b_1 * x[n-1] + + b_N * x[n-N])$$

$$y[n] = \sum_{i=0}^N (b_i * x[n - i])$$

- $x[n] \rightarrow$ input signal
- $y[n] \rightarrow$ output signal
- $b_i \rightarrow$ filter coefficients
- $N \rightarrow$ filter order

FIR's order in our design (N) \rightarrow 4.

Sampling Rate in our design \rightarrow is 16000.

Micro-controller's clock is set \rightarrow 500k the same for ADC and DAC.

We have used the FDA-tool in matlab to design the required filters through it, we gave this tool the kind of filter (High pass filter, Low pass filter or Notch filter). In our project we designed these three filters, we put the sampling frequency, the kind of filter and the order for this filter (N) [N=4, for our designs] to get (N+1) (5) (number of samples) in this tool. After these steps, the shape of response will be shown which resulted from these inputs in designing filter. Then, you will get some coefficients ($b(i)$) and by using their values in the previous equation we get the new values of digital signal after filtration.

Our filters frequencies:

High pass filter → for 1 Hz

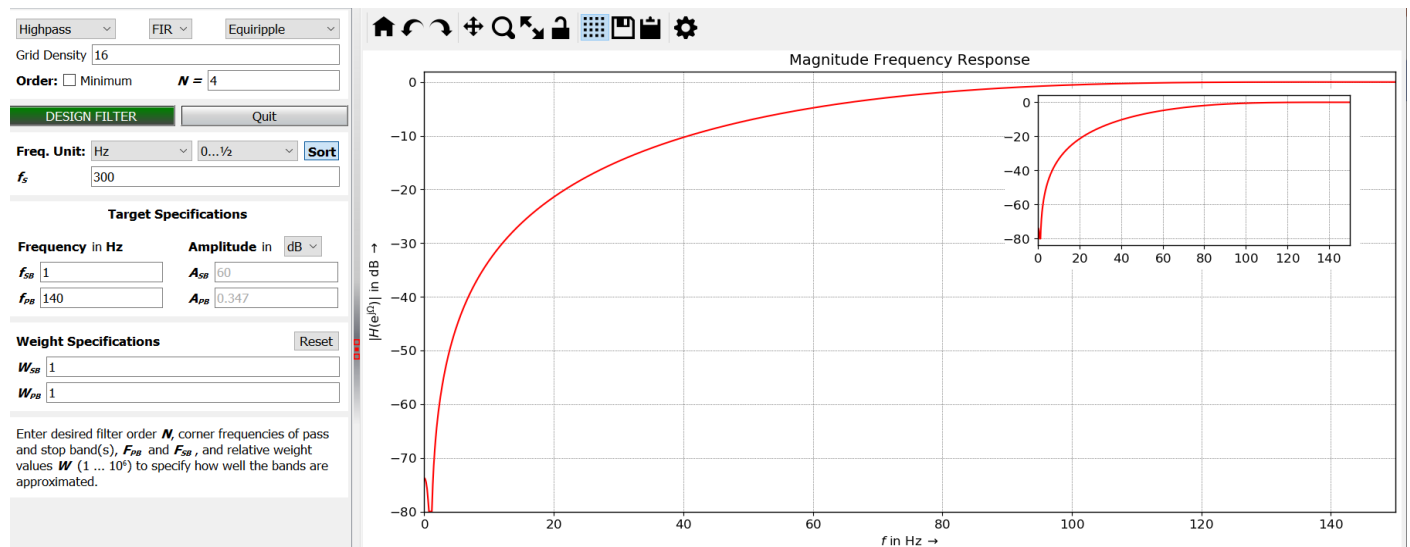
Low pass filter → for 150 Hz

Notch filter → for 50 Hz.

Filters response using FDA-Tool:

High pass filter:

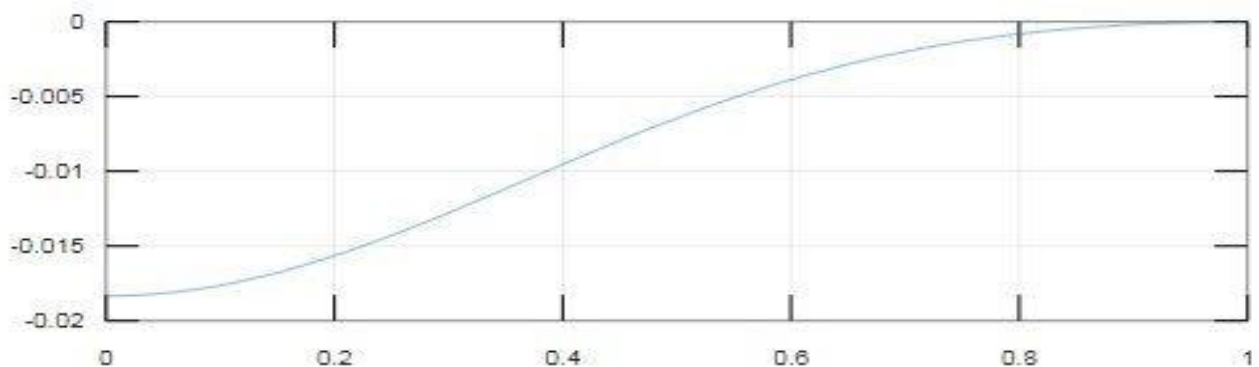
High pass filter for N (Order) = 4, $f_s = 300$ Hz, $f_{\text{cutoff}} = 1$ HZ



- Allowing frequencies above 1 HZ to be passed

Frequency response Using Octave:

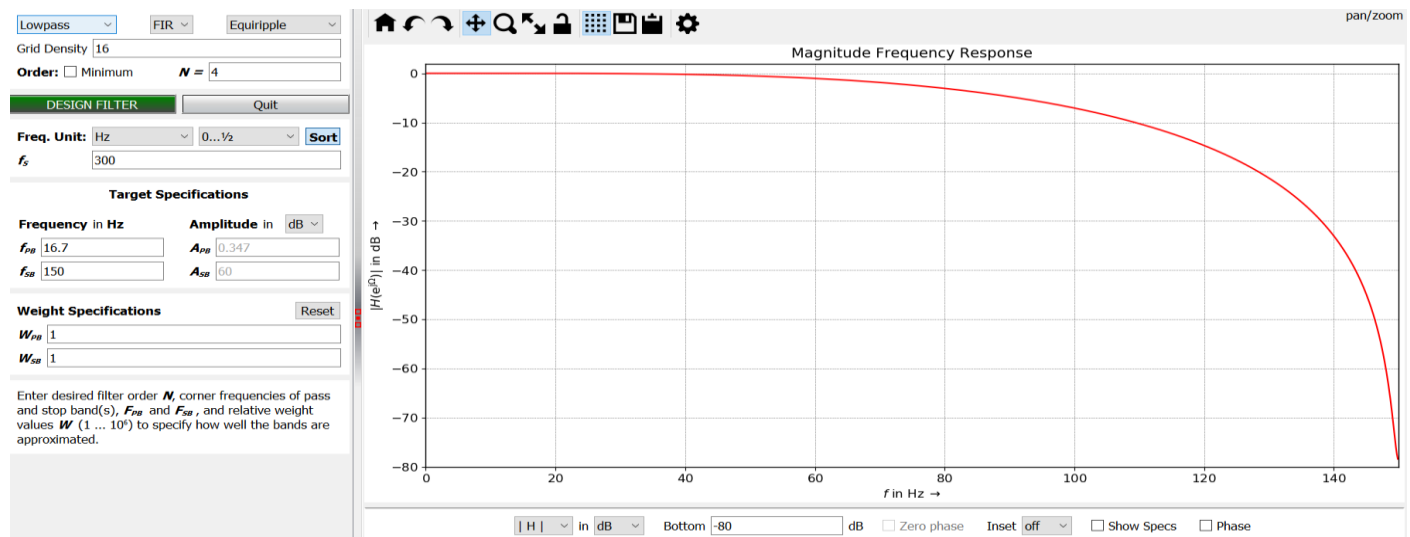
```
octave:5> h = fir1(4, 0.000125, 'high');  
octave:6> freqz(h)
```



- `fir1` ((filter's order (4)), ($\frac{f_{\text{cutoff}}}{\frac{1}{2} * \text{samplerate}}$), (filter's type)).

Low pass filter:

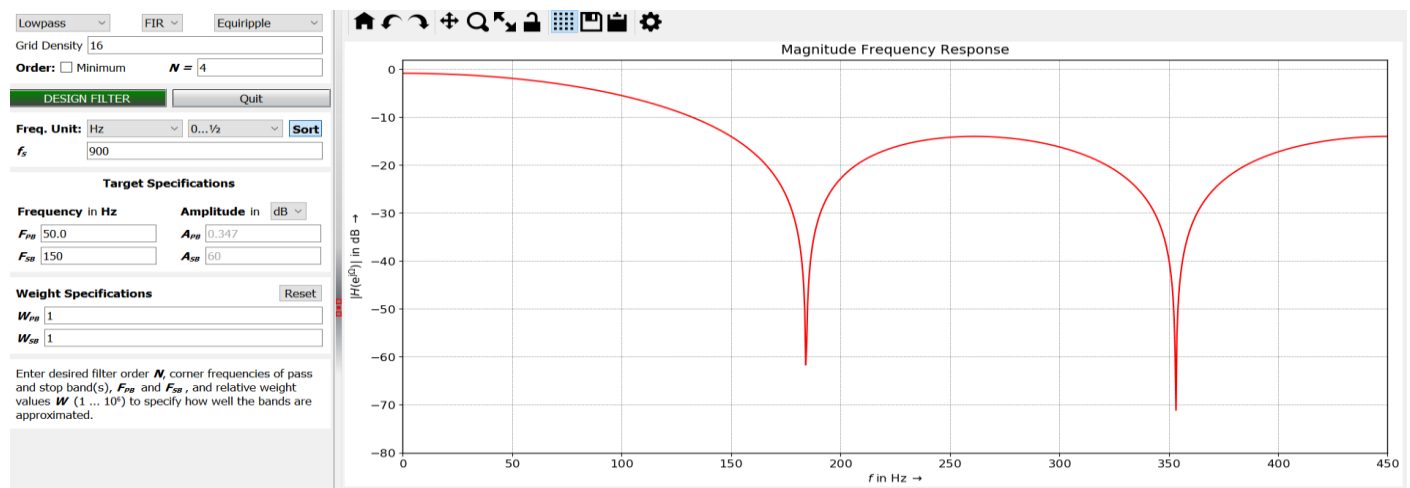
Low pass filter for $N = 4$, $f_s = 300$ Hz, $f_{\text{cutoff}} = 150$ HZ:



- Allowing frequencies below 150 Hz to be passed.

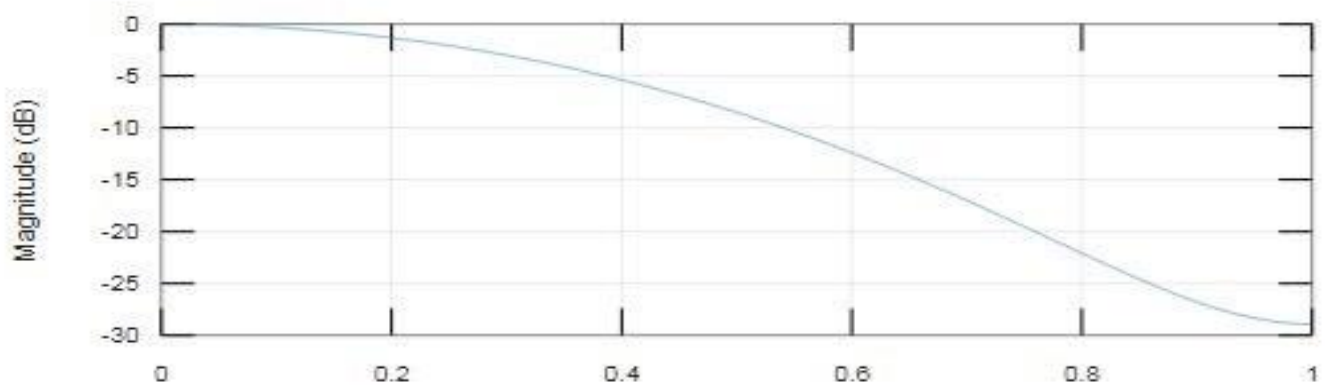
Low pass filter for $N = 4$, $f_s = 300$ Hz, $f_{\text{cutoff}} = 150$ HZ:

- We used a low order \rightarrow to be matched with the capacity of 8051 micro-controller.
- Disadvantage from using low orders \rightarrow low in accuracy of the expected output.



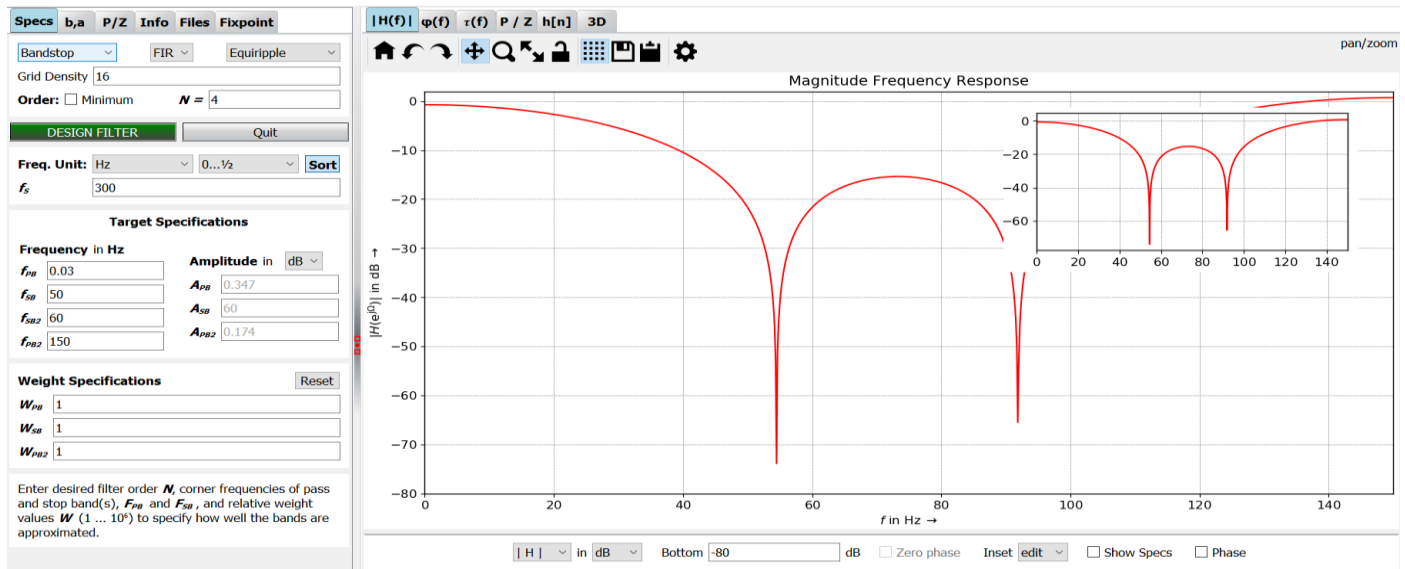
Frequency response Using Octave:

```
octave:3> l = fir1(4, 0.01875, 'low');  
octave:4> freqz(l)
```



Notch filter:

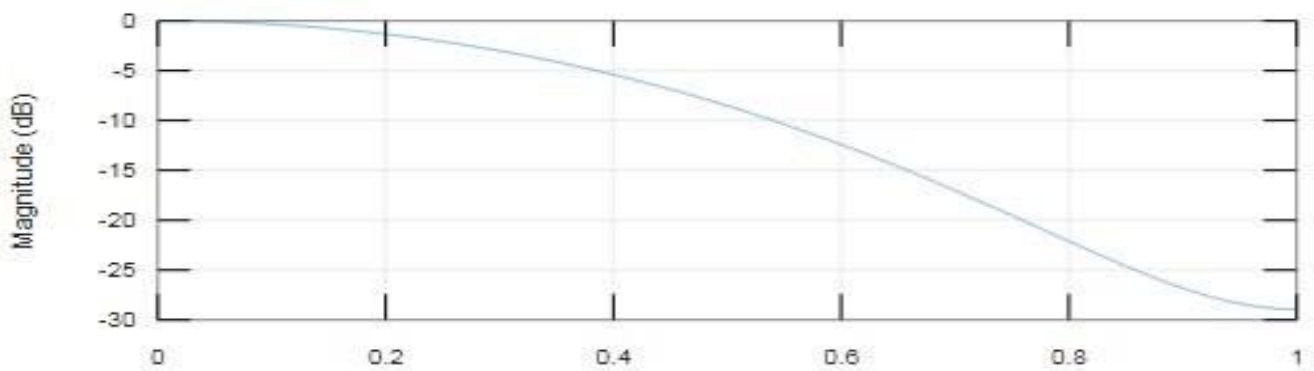
Notch filter for $N = 4$, $f_s = 300$ Hz, $f_{\text{cutoff}} = 50$ Hz:



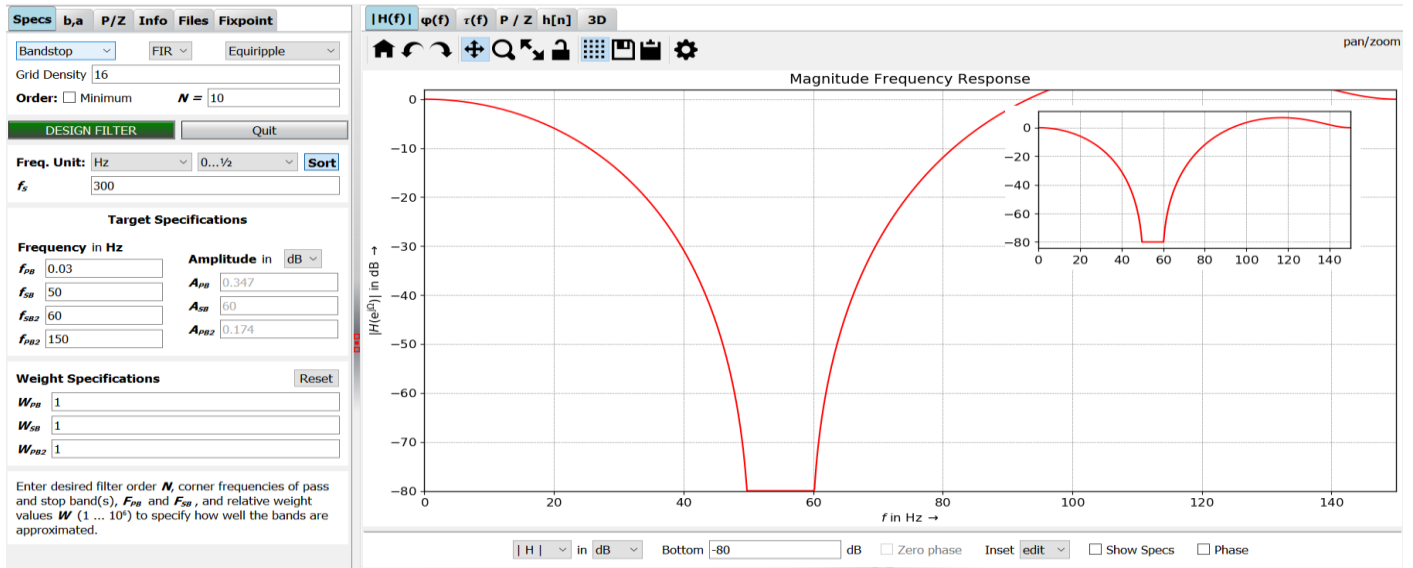
- The response for notch filter isn't accurate at all due to using low orders.
- Notch filter allows to pass all frequencies except frequency= 50 HZ.

Frequency response Using Octave:

```
octave:7> n = fir1(4, 0.00625, 'stop');
octave:8> freqz(n)
```



Notch filter for $N = 10$, $f_s = 300$ Hz, $f_{\text{cutoff}} = 50$ Hz:



- The response gets better for high order as what had been showed from our results using more than one tool in graphing to visualize the response of the designed filter.

Thanks

Dr/ Ahmed M. Ehab

Eng/ Sherif Sayed