

# Designing a Microcontroller-Based Low-Pass, High-Pass and Band-Pass Digital Filter with Op-Amp

Tochukwu Chiagunye, Ilo Somtoochukw and Eze Aru Okereke  
tchiagunye2@yahoo.com, somtofrancis@gmail.com and okezar@yahoo.com  
Computer Engineering Department,  
Michael Okpara University of Agriculture, Umudike, Abia State, Nigeria

**Abstract**—This paper presented the mathematical model and the Microcontroller based design approach to Low-pass, High-pass and Band-pass Digital Filter. The derived system was simulated with the aid of MATLAB 7.5. Here the required system equations and expressions were incorporated into the program code and operational flowchart were designed for the sequential line of code. The codes were written in MATLAB 7.5, thereby deriving the M-file that constituted the system. Assembly language program is used to program the PIC Microcontroller to implement the filtering task.

**Keywords**—Microcontroller, Filter, Low-pass, High-pass, Band-pass, Assembly Language

## 1.0 INTRODUCTION.

Digital filters are a very important part of digital signal processors (DSP). In fact, their extraordinary performance is one of the key reasons that DSP has become so popular. Filters have two uses: signal separation and signal restoration. Signal separation is needed when a signal has been contaminated with interference, noise, or other signals [1]. For example, imagine a device for measuring the electrical activity of a baby's heart while still in the womb, the raw signal will likely be corrupted by the breathing and heartbeat of the mother. A filter might be used to separate these signals so that these can be individually analyzed. Signal restoration is used when a signal has been distorted in some way. For example, an audio recording made with poor equipment may be filtered to better represent the sound as it actually occurred. Another example is the deblurring of an image acquired with an improperly focused lens, or a shaky camera. Also, noise may occur from wind or rain at an outdoor music presentation. Filtering out sinusoidal components of the signal that occur at frequencies that cannot be produced by the music itself results in recording the music with little wind and rain noise. Sometimes the signal is corrupted not by noise, but by other signal frequencies that are of no interest. If the signal is an electronic measurement of a brain wave obtained by using probes applied externally to the head, other electronic signals are picked up by the probes, but the physician may be interested only in signals occurring at a particular frequency. By using digital filtering, the signals of interest only can be presented to the physician.

These problems can be solved with either analog or digital filters. Which is better? Analog filters are cheap, fast, and have a large dynamic range in both amplitude and frequency. Digital filters, in comparison, are vastly superior in the level of performance that can be achieved. For example, Some low-pass digital filter has a gain of  $1 \pm 0.0002$  from DC to 1000 Hz, and a gain of less than 0.0002 for frequency above 1001 Hz. The entire transition occurs within only 1 Hz. Digital filters can achieve thousands of times better performance than analog filters. This makes a dramatic difference in how filtering problems are approached. With analog filters, the emphasis is on handling limitations of the electronics, such as the accuracy and stability of the resistors and capacitors. In comparison, digital filters are so good that the performance of the filter is frequently ignored. The emphasis shifts to the limitations of the signals, and the theoretical issues regarding their processing [1].

Applications involving processing of signals from external analog sources/sensors usually require some kind of digital filtering. For extremely high filter performance, Digital Signal Processors (DSP) is usually chosen, but in many cases these are too expensive to use. Then 8- or 16-bit Microcontrollers unit come into the picture [2]. These are inexpensive, efficient, and have all the required I/O features and communication modules that DSP(Digital Signal Processor) seldom have. The PIC (Peripheral Interface Controllers) microcontrollers are excellent for signal processing applications due to their powerful architecture, strong instruction set and built-in multi-channel 10-bit Analog to Digital Converter (ADC)[2].

## 2.0 Materials and Methods

The system was initiated with the realization of the block diagram shown in fig 1 below. The block diagram of fig 1 below, shows the PIC microcontroller as the major unit of the project, since the Analog to Digital conversion takes place inside the Microcontroller and all the filtering processes also happens in the Microcontroller unit. The microcontroller has an ADC inbuilt, this is one reason that makes PIC Derivatives of the microcontroller excellent for signal processing and analysis.

The visual output from the microcontroller is sent to the LCD and the audio output is sent to the speakers. The tone generator unit is made of a 555 IC configured in astable mode. It generates a signal of

1KHz which is fed to the microcontroller for processing.

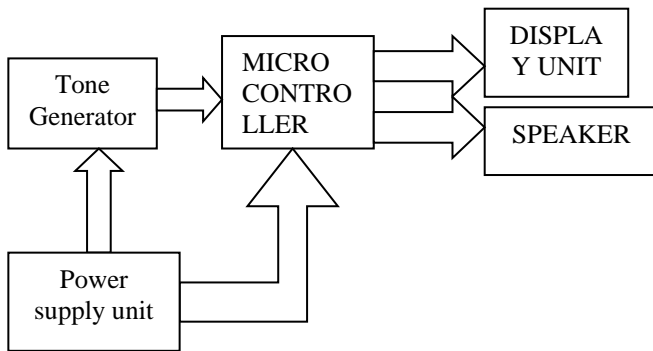


Fig 1: The block diagram of the system hardware unit.

A Tone Generator of 1 Hz was realized with a 555 timer IC configured in Astable mode to generate a 1Hz signal which was sent to the microcontroller for conversion and subsequent filtering. The PIC microcontroller with an inbuilt ADC was used to convert the analog signal from the tone generator into digital signal. The MCU uses the stored assembly program to manipulate this digitized signal, performing filtering, sending same to an inbuilt DAC for digital to analog conversion. Assembly language was used to develop codes for the microcontroller to enable it read the values sent by the Tone generator (555 IC) so as to take appropriate action.

The system realized was modeled and analyzed with Matlab 7.5. The generated Transfer Function of the Op-Amp filter circuit is fed into the Matlab. Here, the values of the parameters; resistance and capacitance were assigned to the model based on the design considerations. The Transfer Function was fed directly into the MATLAB, the op amp \_tf command was then used to generate the Transfer Function values. For example, given the Transfer Function of a band pass filter as  $E_o/E_i = -[C_1S/R_1C_1C_2S][R_2/R_2C_2S+1]$  The following code are applied to MATLAB:

```

>> R1 = 20000; R2 = 500; C1 = 0.1E-6; C2 = 0.1E-6;
>> s = tf('s')
Transfer function:
s

>> opamp_tf =
-(C1*s/(R1*C1*s+1))*(R2/(R2*C2*s+1))

Transfer function:
-5e-005 s

-----
1e-007 s^2 + 0.00205 s + 1
    
```

Here the MATLAB BODE command, bode(opamp\_tf) is used to plot the Frequency Response of the system. The approach that was adopted in the implementation of the project was the method that allows the representation of the Transfer Function into numerator[num] and denominator[denum] values and input same into the MATLAB.

### 3.0 System Design

#### 3.1 System Specification.

The Microcontroller-based digital filter is a system that helps users to continuously monitor the filtering action at different inputs levels with the help of an LCD and an audio speaker. The system derived is able to issue a signal output that will be used to analyze the type of filter produced at various input levels.

During its operation, a maximum frequency value (the cutoff frequency)  $f_c$  of 1KHz is set to determine the pass-band and the stop-band of each of the filter. For the low pass, the pass-band is set between  $0 < f_c$  and stop-band is set above  $f_c$ , high-pass, pass-band is above  $f_c$  and stop-band is between 0 and  $f_c$  and band-pass is set to frequencies between the pass band of low and high pass. For this project the cut-off frequencies for the various filter types of reference is keyed into the microcontroller during Microcontroller programming.

#### 3.2 Power Supply Design

The power requirement for all the components used in this work falls within the 5Vdc and 9Vdc range. Therefore a suitable 5/9Vdc supply is designed by using a 9V battery, voltage regulators, capacitors and resistors of varying specifications. The input voltage to the system is obtained from the 9Vdc battery which is used in order to maintain systems portability and reduce fear of constant power outage. The output from the battery is fed to the MC7505 5-Vdc regulator to provide a fixed 5Vdc output for the system. The power supply circuit can handle up to 1A of current, depending on the current rating of the battery. The block diagram of the regulated power is shown in fig. 2, the graph shows what happens after each block.

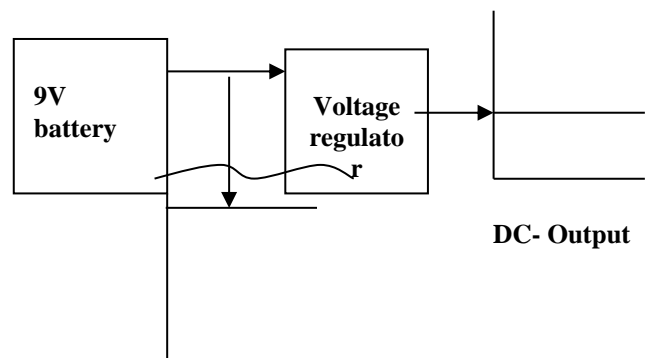


Fig 2: The block diagram of regulated voltage

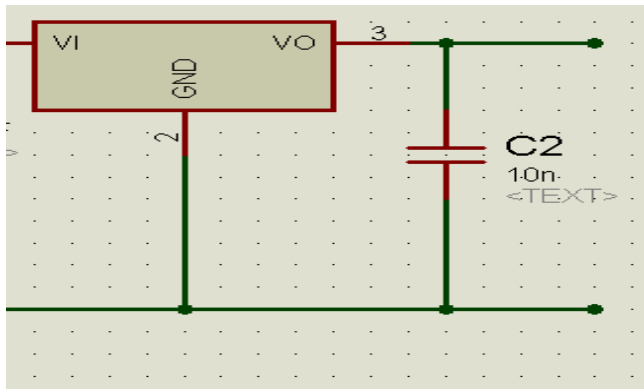


Fig 3: The schematic diagram of the Power supply  
 The output D.C. voltage and current are given by;

$$V_{d.c} = 2V_m/\pi \quad 1$$

$$I_{d.c} = V_{d.c}/R_L = 2V_m/\pi R_L = 2I_m/\pi \quad 2$$

Therefore, the D.C. output voltage,  $V_{d.c} = 2V_m/\pi = 2 * V_m/\pi$

$$V_{dc} * \pi = 2 * V_m \quad 3$$

$$V_m = V_{dc} * \pi/2$$

$$= 14.14V$$

Assuming a load resistance of 1000Ω, the d.c. power delivered to the load is

$$P_{d.c} = V_{d.c}^2/R_L \quad 4$$

$$= 9.0^2/1000 = 0.081 W$$

The 555 timer configured in Astable mode served as a tone generator . It outputs a continuous stream of rectangular pulses having a specified frequency. Here, the centre frequency =  $1.44 / (R_1 + 2 R_2) C$ .

$$F_c = 1.44 / (R_1 + 2R_2) C \quad 5$$

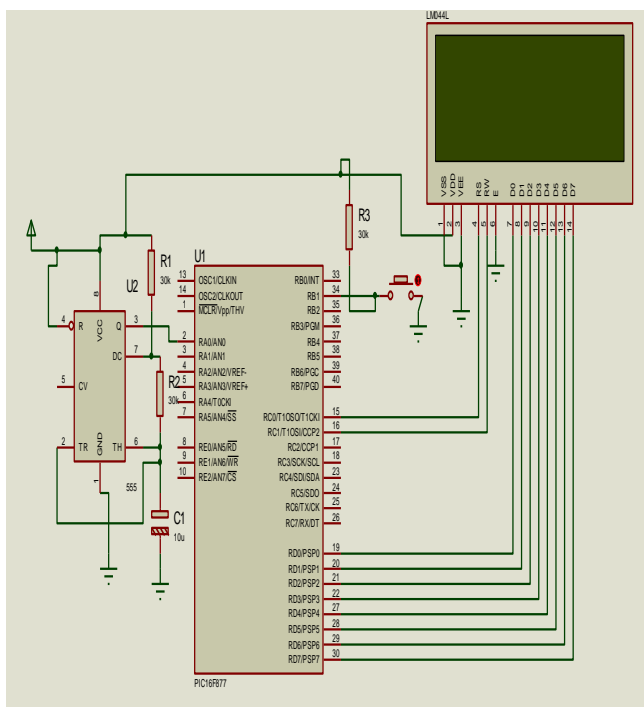


Fig 4: Circuit Diagram

For this project , R1 is selected to be 20K and R2 is 500 , the cutoff frequency of 1KHz, then

$$1000 = (1.44) / (20000 + 1000)C \quad 6$$

$$1000 = (1.44) / (21000)C \quad 7$$

$$C = (1.44) / (21000000) \quad .8$$

$$C = 0.1E-6$$

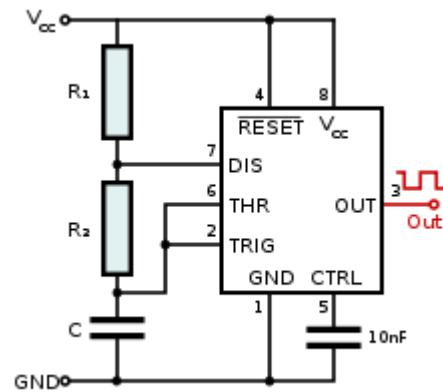


Fig 5: The schematic diagram of 555 IC in astable mode.

The output (pin 3) is fed to the microcontroller.

### 3.3 The Filtering Unit.

The Microcontroller is the central filtering unit in the project. The Microcontroller used is PIC 16F877A, Fig. 6 below shows the pin-out of the PIC 16F877A with minimum configuration.

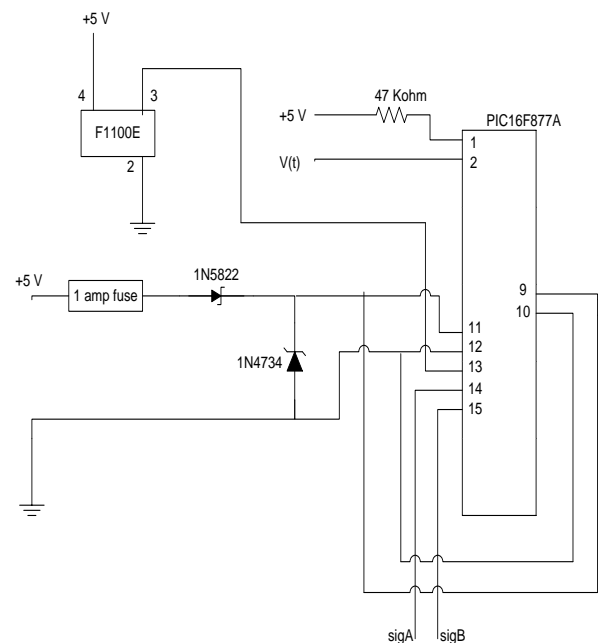


Figure 6: The schematic diagram of PIC Microcontroller

### 3.4 The PIC microcontroller Software design unit

This filtering process is basically software controlled. An assemble language program is used to generate the necessary filtering source codes which

are compiled using MIDE to generate the necessary machine language codes for the PIC microcontroller. A block diagram of program modules and structure is shown in Fig 7.

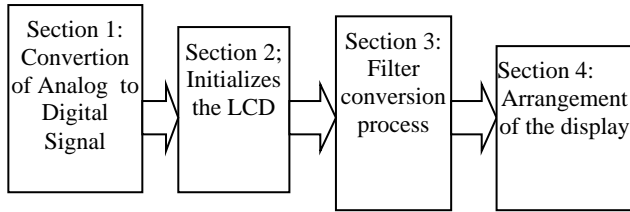


Fig 7: The System Software Structure

### 3.5 The display unit

The PIC microcontroller displays the output of the filter signal using the Hitachi HD44780-based character LCD module shown in fig 8 below. Here PIC16F877A microcontroller is connected to HD44780 LCD in 4-bit interface data. Here, only four bus lines (DB4 to DB7) are used for data transfer. The data transfer between the HD44780 and the PIC16F877A is completed after the 4-bit data has been transferred twice. As for the order of data transfer, the four high order bits (for 8-bit operation, DB4 to DB7) are transferred before the four lower order bits (for 8-bit operation, DB0 to DB3). Any character on HD44780 LCD is displayed by sending its respective code.

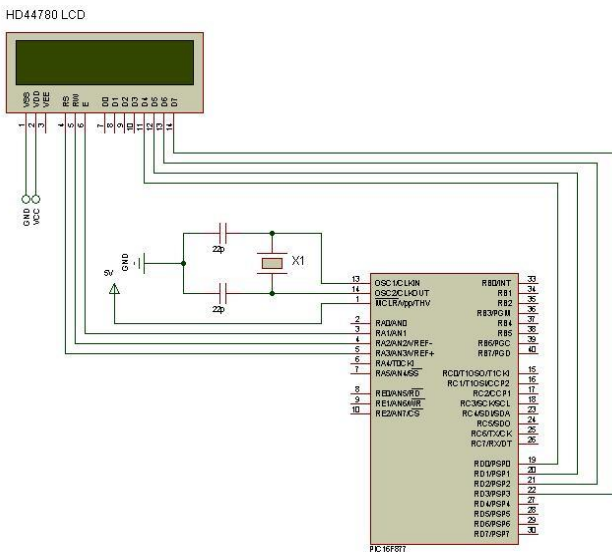


Fig 8: The schematic diagram of the display unit.

### 3.6 The Software Subsystem Design

MATLAB 7.5 is an excellent tool for simulating, analyzing and monitoring the filtering performance of filter systems. In this section, we present the three basic digital filters designed using the top-bottom design method, modeling their individual Op-Amp Filter circuit, generating their Transfer Functions using Complex impedance modeling technique. The transfer function will be used in the implementation of the various digital filter using Matlab 7.5.

### 3.7 System modeling

Using complex impedance analysis, the time domain parameters of an Op-Amp filters are converted into their equivalent frequency domain equivalence starting from the basic characteristics as governed by the Ohm's Law. Subsequently, using Laplace transforms, the impedance of these linear passive elements as functions of the Laplace variable,  $S$ . Ohm's Law in the frequency domain is  $E(s) = I(s)Z(s)$ . Frequently for convenience, the  $(s)$  is dropped.

Table 1: Basic Voltage-Current Relationships

Element	Time Domain Voltage-Current Relationship	Frequency Domain Voltage-Current Relationship	Complex Impedance Voltage-Current Relationship
Resistor	$e = Ri$	$E(s) = RI(s)$	$Z(s) = R$
Capacitor	$e = \frac{1}{C} \int idt$	$E(s) = \frac{1}{Cs} I(s)$	$Z(s) = \frac{1}{Cs}$
Inductor	$e = L \frac{di}{dt}$	$E(s) = Ls I(s)$	$Z(s) = Ls$

### Microcontroller Program Flowchart

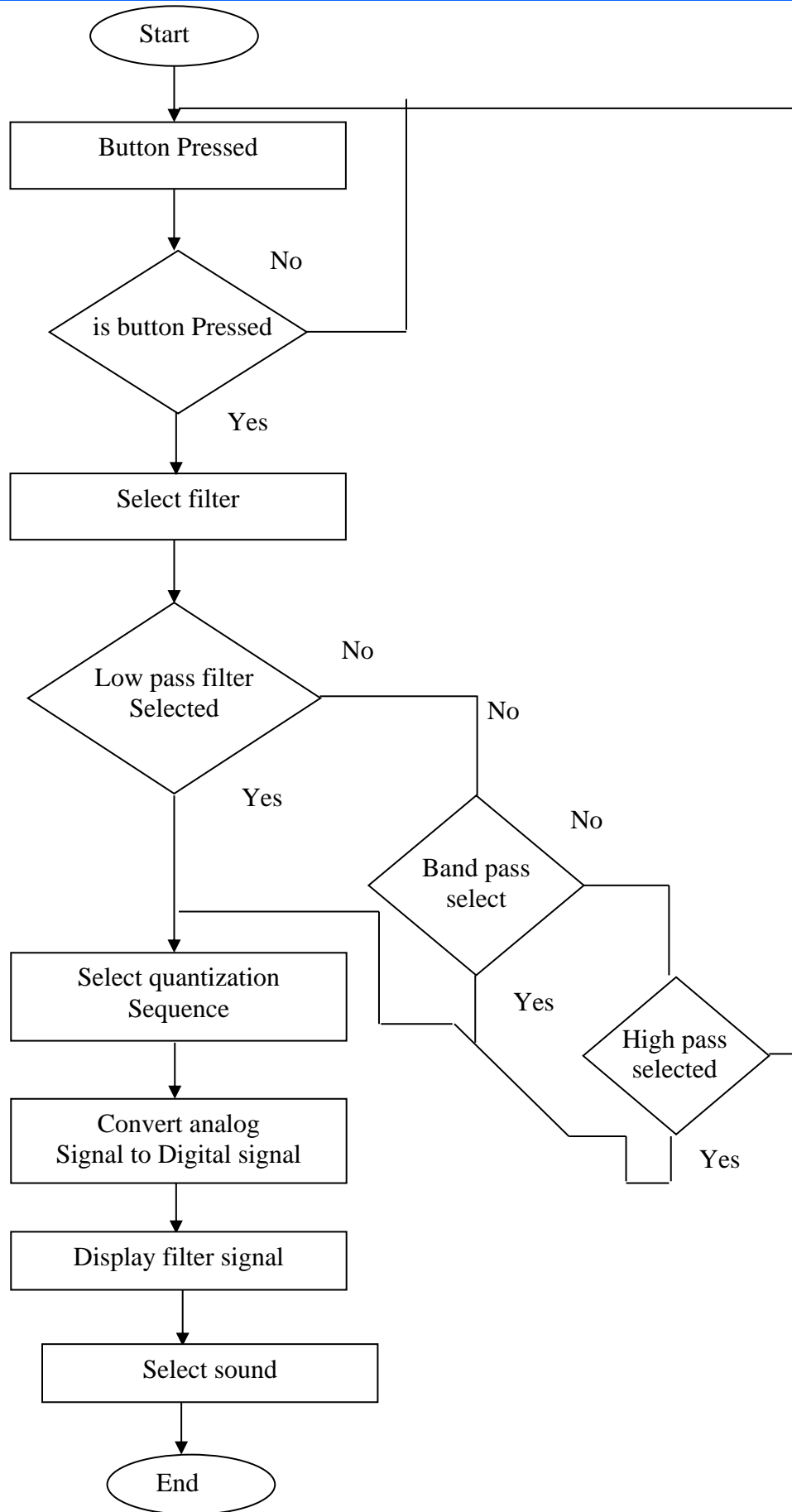


Figure 9: Program Flowchart

### 3.9 A single pole Low-pass Filter with Op-Amp

The fig.10 below show the circuit diagram of the Op-Amp filter which is also referred to as the integrator.

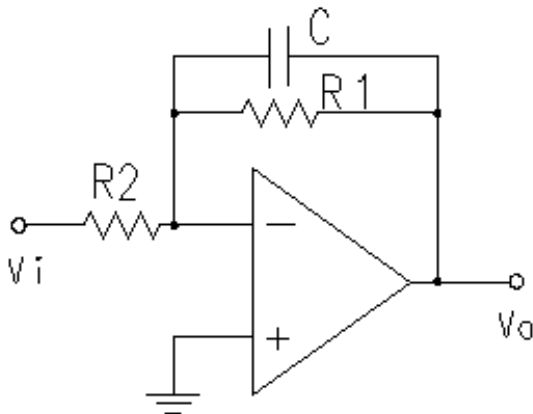


Fig 10: A Single pole low pass filter circuit

Complex impedance analysis is used to derive the Transfer Function of the circuit. Here, all elements in the circuit diagram to their equivalent complex impedances. In this case:

$R_1$ ,  $R_2$  and  $C_1$  are replaced with their equivalent complex impedance values, thus:

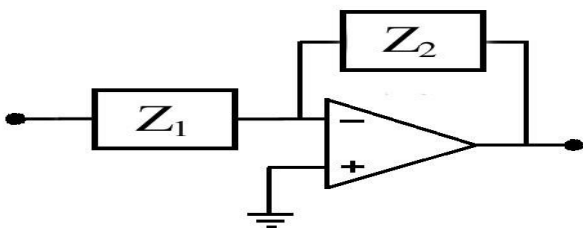


Figure 11: Complex Impedance representation

From the figure above, where  $Z_2$  is the parallel combination of  $R_1$  and  $C$  while  $Z_1=R_2$

Therefore, with reference to op amp characteristics, the circuit is:

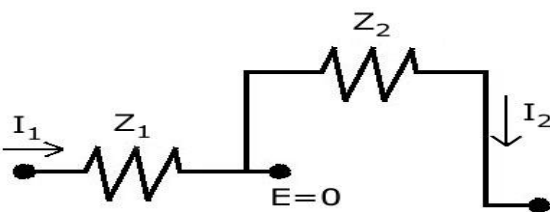


Fig 12: The reduced Op-Amp circuit

From the circuit above,

$$I_1(s) = E_i \frac{1}{Z_1}$$

Applying Ohm's Law to  $Z_2$ :

$$I_2(s) = (0 - E_o) \frac{1}{Z_2} = -E_o \frac{1}{Z_2} \tag{10}$$

Since  $I_1 = I_2$ :

$$E_i \frac{1}{Z_1} = -E_o \frac{1}{Z_2} \tag{11}$$

Rearranging:

$$\frac{E_o}{E_i} = -\frac{Z_2}{Z_1} \tag{12}$$

Substituting for the values of  $Z_1$  and  $Z_2$  gives the Transfer Function of the referenced FIR Digital filters is therefore computed as:

$$Z_2 = R_1 // C = \frac{R_1 * 1/sC}{R_1 + 1/sC} \tag{13}$$

Then dividing Num and Denum by  $sC$  gives;

$$Z_1 = R_2 \tag{14}$$

$$E_o/E_i = -(Z_2/Z_1)$$

Then substituting the values of  $Z_2$  and  $Z_1$  thus;

$$E_o/E_i = -[R_1 / (R_1 sC + 1)] / R_2 \tag{15}$$

Rearranging the equation gives

$$E_o/E_i = -[(1/R_2C)] / [s + (1/R_1C)] \tag{16}$$

### 3.10 The Single Pole High-pass Filter with Op-Amp

Applying similar top-down design method used for the low-pass filter, the transfer function of a high-pass filter can also be derived by modeling fig 11, using Complex Impedance analysis.

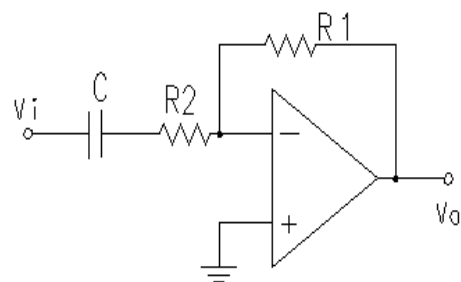


Fig 13: Single pole high-pass filter circuit.

The high-pass filter circuit can be represented in complex impedance notation as shown below.

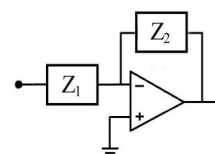


Fig14: complex impedance representation

The transfer function can be obtained by using the relationship as derived in section 12 above. Thus

$$\frac{E_o}{E_i} = -\frac{Z_2}{Z_1} \quad 17$$

From fig 14,  $Z_1 = R_2 + C$  and  $Z_2 = R_1$ . The transfer function will now be the substitution of their individual complex impedance values as shown below.

$$\begin{aligned} E_o/E_i &= -R_1/[R_2+(1/sC)] \\ &= [R_1]/[(R_2sC+1)/sC] \\ &= [R_1sC] / [(R_2sC+1)] \end{aligned} \quad 18$$

Rearranging the equation gives:

$$E_o/E_i = -(R_1/R_2)s / [s+(1/R_2C)] \dots \quad 19$$

The equation represents the transfer function of a single pole first order high-pass with Op-Amp.

### 3.11 The band-pass filter with Op-Amp

The band-pass filter obtains from the parallel combination of high-pass and low-pass filters as shown fig 15.

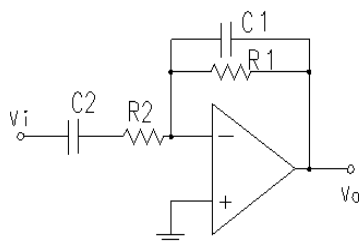


Figure 15: The band-pass circuit

The band-pass circuit can be reduced to as shown Fig 16.

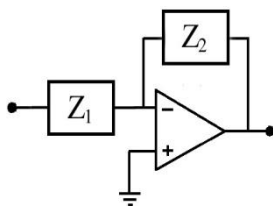


Figure 16: the complex impedance representation of the band-pass filter

Using complex impedance design analysis, the circuit can be modeled based on the Op-Amp assumptions to obtain the relationship shown.

$$\frac{E_o}{E_i} = -\frac{Z_2}{Z_1} \quad 20$$

From the circuit of fig 16,  $Z_2=R_1/C_1$  and  $Z_1=R_2 + C_2$ . The transfer function of a band-pass filter can be obtained by substituting their various complex impedance values to equation above;

$$E_o/E_i = - [(R_1*(1/sC_1)) / (R_2+(1/sC_2))] / [(R_2+(1/sC_2))] \quad 21$$

$$= [R_1/ (R_2 sC_1+1)] / [(R_2sC_2+1)/sC_2] \quad 22$$

and rearranging the expression is shown in equation 23.

$$\frac{E_o}{E_i} = -\left(\frac{C_1s}{R_1C_1s+1}\right)\left(\frac{R_2}{R_2C_2s+1}\right) \quad 23$$

## 4.0 System Simulation

Matlab 7.5 is deployed to help in the analysis and simulation of the modeled FIR filter circuits. The transfer function obtained for the various FIR digital filters are rearranged into vector format of denominator (den) and numerator (num). The resistor and the capacitor values are selected based on the design of the signal generation subsystem. The MATLAB program codes are generated and edited using m. file text editor. Here, the transfer functions obtained in the designs above are rearranged in a vector format to represent the den and num used in the MATLAB programming. The source codes and arrangements are used for the BODE plot which shows the relationship between the magnitude, phase and the frequency of the filter. The BODE plot as obtained after executing the source codes, are represented in the proceeding graphs.

### 4.1 Single pole First order low-pass filter simulation

The analysis of the behavior of the filter in time domain can be done by plotting the step response diagram of the filter, using the step function of MATLAB. To obtain the step response of the low-pass filter the program code is executed in the MATLAB 7.5. The plot obtained is also presented.

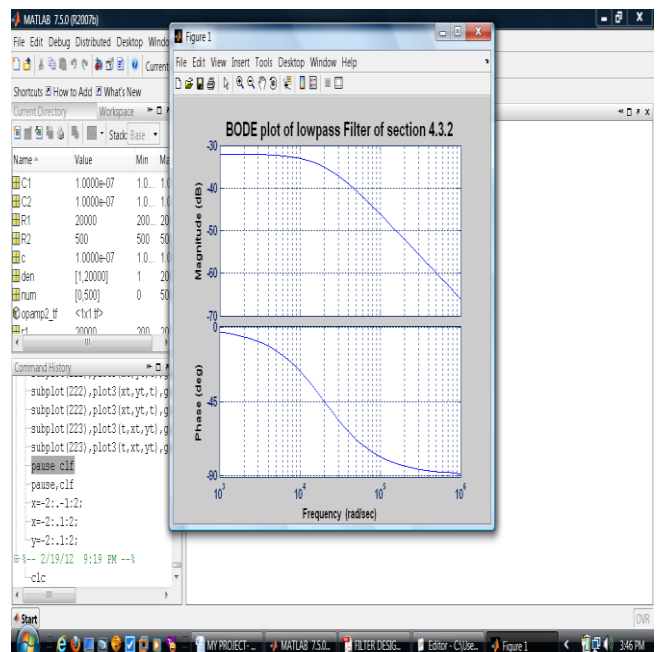


Fig 17: The bode plot of low-pass filter

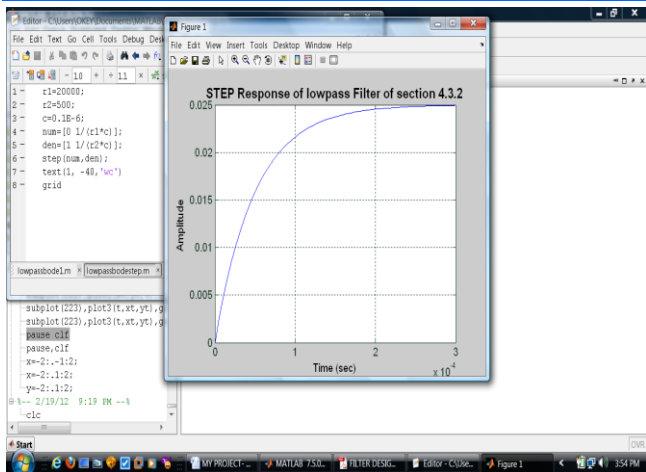


Fig 18: Step response of low-pass filter

The poles and zeros of the filters can also be calculated from MATLAB by using MATLAB pole and zero functions. The poles analysis is relevant since it is used to know if the simulated filter is stable or not based on the calculated poles. The one with all negative real part is stable as show in fig 18.

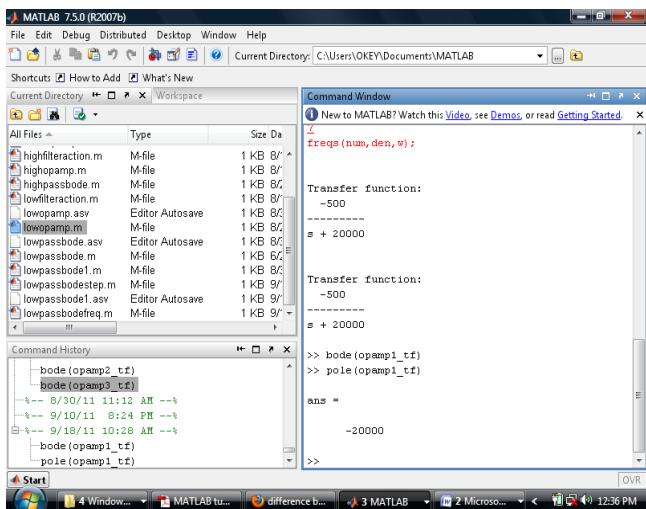


Fig 19: The pole representation of the low-pass filter

#### 4.2 Single Pole First order High-pass filter Simulation

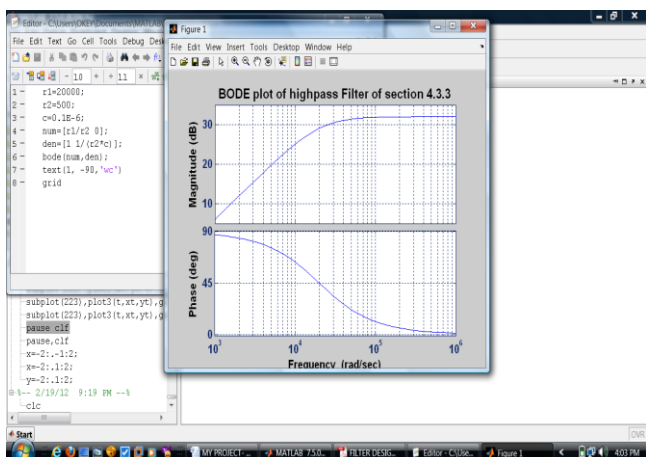


Figure 20: Bode plot of High-pass filter

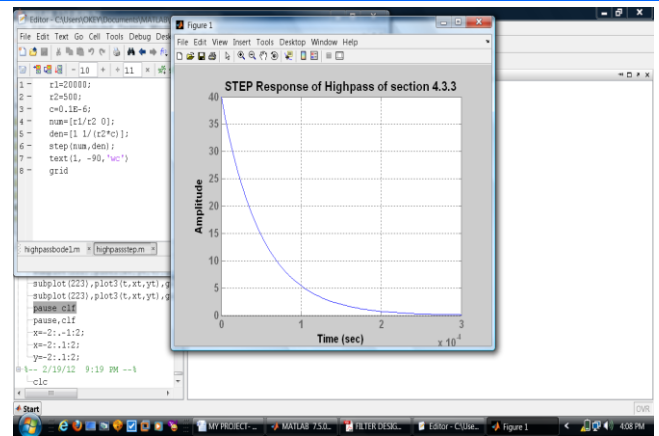


Figure 21: The step response of High-pass filter

From the filter pole calculation using the MATLAB pole function, one can conclude that the simulated filter is stable because it has one negative real part of -500 since it is a single pole filter.

#### 5.0 Recommendation and Conclusion

The design of a Microcontroller-based low-pass, high-pass and band-pass digital filter with op-amp has been achieved in this project. The design followed a detailed analysis of an existing system, modeling of new systems and subsequently designing the envisaged system. The end result is a Microcontroller-based filtering system whose output can also be monitored and analyzed using the Matlab 7.5 simulator. A signal processing operating system could be developed without great energy in running filters for function-specific signal processors. These devices can then be deployed in engineering labs for the training of staff and students. This program was developed using MATLAB 7.5, the use of a lower version to modify any part of this program could result to errors. Also, the presence of a strong graphic user interface platform possessed by Windows Vista also affects the kernel function application in a way that the processing is delayed for a varied time. This actually, will not affect the final result of the program. If for any reason, there happen to be a bug generated in the program procedure, a self-debugging event can be generated.

#### REFERENCES

1. Howstuffworks.inc,2012<<http://electronics.howstuffwork.com/microcontroller4.htm>>
2. Steven .W.Smith,2011<<http://www.dspguide.com/ch14.htm>>
3. Jullius.O.SmithIII,9<sup>th</sup> June2012,<[https://ccrma.stanford.edu/~jos/filters/Examples\\_Digital\\_Filters.html](https://ccrma.stanford.edu/~jos/filters/Examples_Digital_Filters.html)>
4. UBM.Tech,2012<<http://www.eetimes.com/discussion/beginner-s-corner/4024479/Introduction-to-Digital-Filters>>
5. The Mathworks.Using Simulink Version 6.The MathWorks Natick,MA.1990-2004



6. Hahn, Essential MATLAB for Scientists and Engineers, John Wiley & Sons, New York, NY, 1997

7. <http://www.asic-soc.blogspot.com/2009/01/digital-filter-implemntation-using-html>

8. MikroElektronika,2012,<<http://www.mikroe.com/eng/chapters/view/71/chapter-1-basic-concepts-of-digital-filtering-and-types-of-digital-filter>>

9. S. J. Chapman. MATLAB Programming for Engineers. Thomson, 2004

10. Dorf and Bishop, "Modern Signal processing systems", 10th Edition, Prentice Hall, 2005