Middle-East Journal of Scientific Research 19 (6): 875-879, 2014
ISSN 1990-9233

DOI: 10.5829/idosi.mejsr.2014.19.6.1457

Implementation of Graphic Equalizer Using ADSP-BF533

A. Geetha

Department of ECE,
Bharath Univeristy, Chennai-73, India

Abstract: This paper considers the implementation of real time audio applications and also offers programming techniques to create DSP algorithms which found in today's professional and consumer audio equipment. The effect includes filtering operation, audio effect of volume control and channel movement. The various effects are controlled by pressing push buttons (PF8-PF11) on ADSP-BF533 EZ-KIT LITE. Depending on the code implementation, the above buttons are pressed. The output is discussed with spectrum and the LED display.

Key words: DSP algorithms · Audio applications · Consumer audio equipment · ADSP-BF533 EZ-KIT LITE

INTRODUCTION

Angraphic equalizer is a processor which is designed to deliberately alter the tonal quality of audio passing through it. It does this by using a number of filter circuits, which are capable of applying gain to audio signals within specific frequency ranges-both positive gain, referred to as 'boost' and negative gain, referred to as 'cut'. It presents a graphic representation of volume and other characteristics of a sound. graphicequalizer is used for signal processing which help in finalizing the sound from audio system. A few things that could be improved; better tone quality so there is less clicking, greater functionality through the push button. The simplest filter circuits in common use within equalizers are high-pass and low-pass filters. a high-pass filter progressively reduces the level of any audio frequencies below a user-specified 'cutoff' frequency, while leaving the level of those above this point comparatively unchanged. On the other hand, the low-pass filter reduces the level of frequencies above the cutoff point, leaving those below comparatively unchanged. This paper describes the implementation of graphic equalizer using the Black fin ADSP-BF533 EZ KIT LITE and VisualDSP++5.0.The advantage of using graphicequalizer relates to the Manipulation or control the pace and according to user preferences. The goal of this project is to be able to distinctly hear the different tones being produced by different types of filters. These different types of filters were controlled by PF push buttons on the ADSP-BF533 EZ-KIT LITE board [1-2].

Audio Interface: The audio input signal from PC or through microphone is given to the processor audio jack in. The audio input is stored in AD1836 buffer. The audio codec uses both the primary and secondary data transmit and receive pins to audio input and output data from audio inputs and outputs. The AD1836 audio codec provides three channels of stereo audio output and two channels of multichannel 96 kHz input. The SPORT0 interface of the processor links with the stereo audio data input and output pins of the AD 1836 codec. The processor is capable of transferring data to the audio codec in time-division multiplexed (TDM) or two-wire interface (TWI) mode. The TWI mode allows the codec to operate at a 96 kHz sample rate but limits the output channels to two. The TDM mode can operate at a maximum of 48 kHz sample rate but allows simultaneous use of all input and output channels [3-5].

Board Set up: The board switch 9 pins demonstrated in fig 2. the switch's position 1 through 4 connects the PF_x(x is 8-11) pins of the processor with the push buttons. In addition the position 5 and 6 connect the transmit and receive frame syncs and SPORT0 clock which is used for the audio decoder.

Filter Coefficients: Infinite Impulse response filters have an impulse response function that has an infinite extension. Algebraically, IIR filters can be represented in the form:

Corresponding Author: A. Geetha, Department of ECE, Bharath Univeristy, Chennai-73, India.

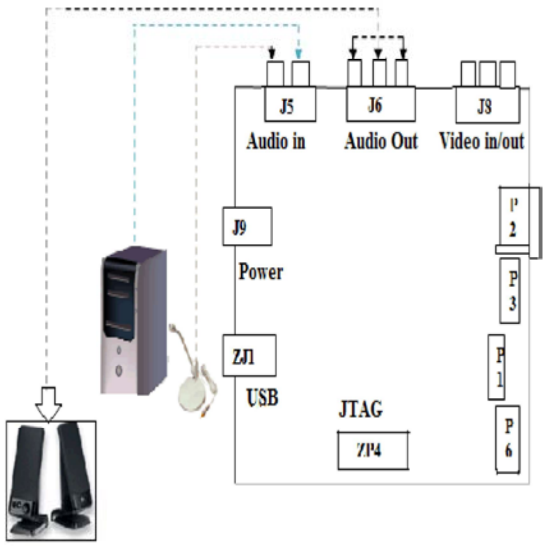


Fig. 1: The audio input is processed in dsp processor blackfin-533 and given out in headphone through audio output jack.

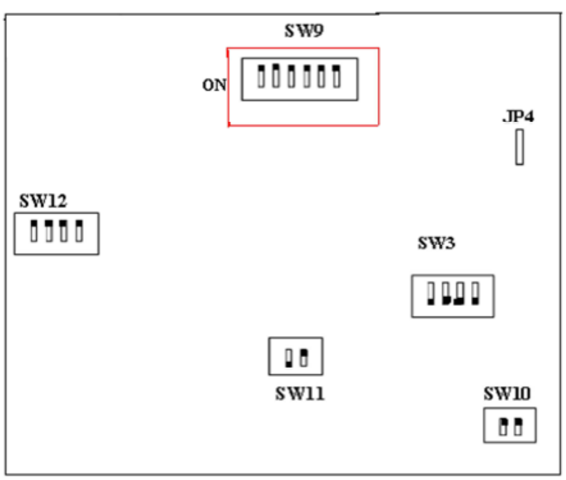


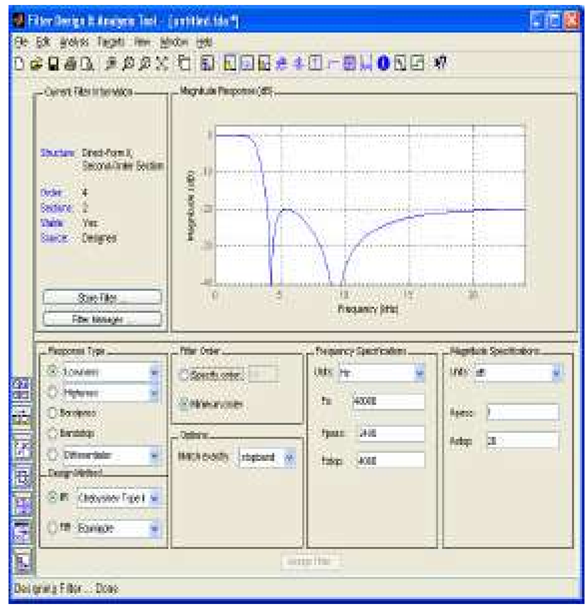
Fig. 2:

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

The IIR filters are programmed using the direct form-2 implementation process. This is shown below:

$$d[m] = x(m) - a_2 d[m-2] - a_1 d[m-1]$$
$$y[m] = b_2 d[m-2] + b_1 d[m-1] + b_0 d[m]$$

Where m represents the integer values and y[m] is the output being produced by the filter. These exact labels were not used in our code but the above is an example of how it could be implemented. Mat-lab was used to design all of the filters [9-12]. This was possible using the fdatool. All of the filters were designed using the Chebyshev Type 2 method. Below is an example of how the IIR filters designed in Matlab using the fdatool.



Code Implementation

Filter: The purpose of this paper was to implement audio filters which would produce distinct variations in sound. These filters included low pass, high pass, band-pass and band stop filters which are controlled by PF push buttons [10-17].

If PF8 is pressed once, LED 4 would come on and the low-pass filter would be initiated. If PF8 is pressed twice, LED 5 would come on and the high-pass filter would be initiated.

```
EX_INTERRUPT_HANDLER(FlagA_ISR)
{
// confirm interrupt handling
if (*pFIO_FLAG_C == 0x0100)
{
*pFIO_FLAG_C = 0x0100;
lp++;
if (lp == 1)
{
lowpassfilter();
}
if (hp==2)
{
highpassfilter();
}
} }
```

The PF8 is activated by using the code *pFIO_FLAG_C = 0x0100 as 0x0100 is the address for PF8. The same concept as shown above in the FlagA_ISR section which initiated PF8 was utilized to activate PF 9. If PF9 is pressed once, LED 7 would come on and the band-pass filter would be initiated [11]. If PF9 is pressed twice, LED 8 would come on and the band-stop filter would be initiated. These filters are processed in the Sport0_RX_ISR section of the ISR file.

If none of the button is pressed, the original audio signal will be transmitted to the speakers with no modification.

Audio Effect: The code is to control the volume. It takes the input values and shifts them by 8 to the right, then stores the value. The volume level for each speaker is scaled by 5000 and also stored [12]. These two values are then multiplied together, shifted 7 bits and sent to the proper output. Scale the volume of each speaker properly, as well as sending the modified values to the output.

```
//when PF10 is pressed the following values will be assigned//volume control will be added here void Process_Data(void)
{
//assigning input to the variable
volume Left =iChannel0 LeftIn<<8;
volumeRight = iChannel0RightIn<<8;
volumeLeft *=5000; //multiplying the input with again to act as a volume
volumeRight *=5000;
//assigning the modified input to the output
iChannel0LeftOut = volumeLeft>>7;
iChannel0RightOut = volumeRight>>7;
}
```

The code is processed using the function Sport0_RX_ISR, which is consequently executed after a complete frame of input data has been received. The new samples are stored in iChannel0LeftIn and iChannel0RightIn respectively. Then the functions in PROCCESS_DATA.C are called in which user code can be executed. After that, the processed values are copied from the variables I Channe l0 Left Out, iChannel0 Right Out, I Channel 1 Left Out and I Channel 1 Right Out into the DMA transmit buffer.

Channel Movement: When PF11 is pressed, the output signal would “move” from the left to the right, the LEDs would also move from the left to the right. Pressing PF11 one more time will stop the effect. Pressing PF11 for the third time, the output signal would “move” from the right to the left, the LEDs would also reflect this change. Pressing PF11 again would stop the effect. Then the process can be started again.

RESULTS AND DISCUSSION

The above spectrum plot explains various effects in generated tone signal. The fig 2 explains the original tone spectrum plot. The original spectrum plot is used for differentiate with the other spectrum plots of various filter signals. The fig3 explains low pass filter spectrum plot which was obtained by applying 5k cutoff frequency to original signal. The fig4 explains high pass spectrum plot which was obtained by applying 4k cutoff frequency to original signal. The fig5 is band pass spectrum plot which was obtained by applying 1-6k pass frequency to original signal and other frequencies are rejected due to band pass filter characteristics. The fig6 explains band stop spectrum plot which was obtained by applying 1-4k stop frequency

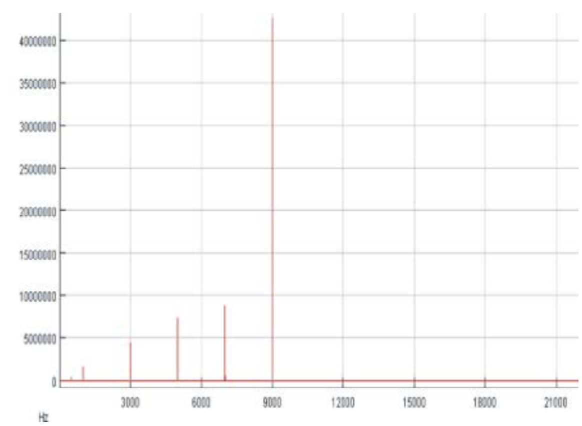


Fig. 3: Original signal

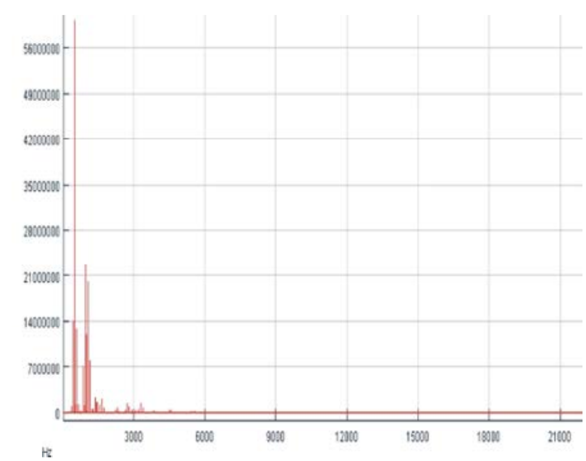


Fig. 4: Low pass signal

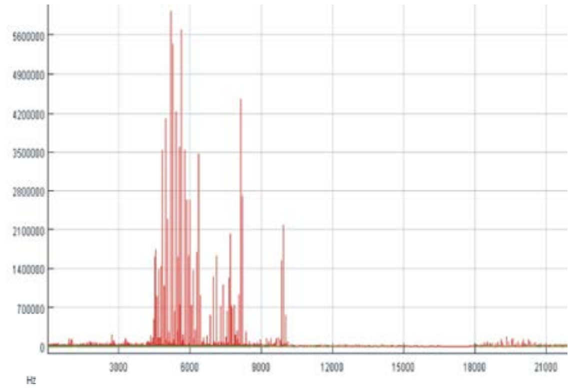


Fig. 5: High pass signal

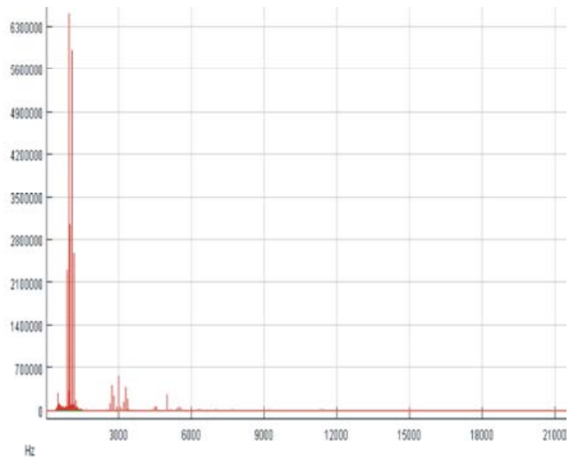


Fig. 6: Band pass signal

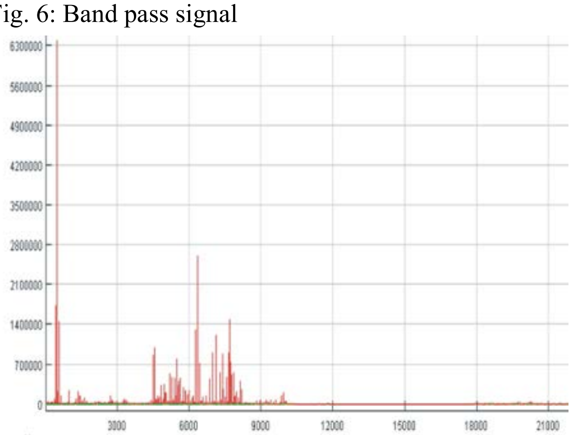


Fig. 7: Band stop signal

to original signal and other frequencies are allowed due to band stop filter characteristics. All spectrum plots were plotted with same size of clipping. The last spectrum plot was obtained by applying 3k low frequency to original audio signal which allows only low frequencies and other frequencies are rejected due to low pass filter characteristics.

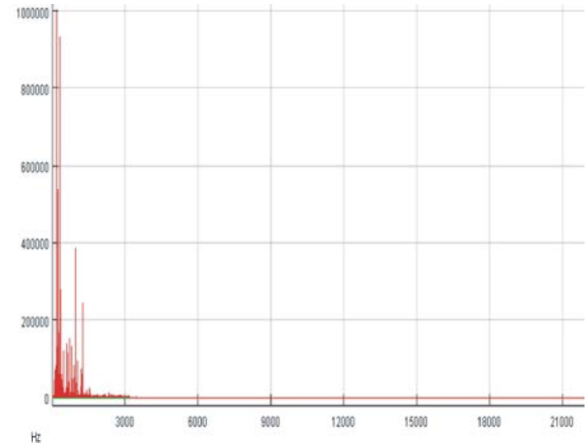


Fig. 8: Filtered Low pass signal for audio

REFERENCES

1. Parks, T.W. And J.H. Mcclellan, 0000. Chebychev Approximation For Non-Recursive Digital Filter With Linear Phase” IeeeTrans.Circuit Theory.
2. Gold, B. And C.M Rader, 1969. Digital Signal Processing Of Signals, Mcgraw-Hills, Newyork.
3. AES Recommended Practice for Digital Audio Engineering, Dec, 1985. Serial Transmission Format for Linearly Represented Digital Audio Data (AES3-1985 & ANSI S4.40-1985)," J. Audio Eng. Soc., 33: 975-984.
5. Regalia, P.A. and S.K. Mitra, 1987. Tunable Digital Frequency Response Equalization Filters, IEEE Trans. Acoust., Speech, Signal Process., ASSP, 35: 118.
6. Swami, M.N.S. and K.S. Thyagarajan, 1976. Digital Bandpass and Bandstop Filters with Variable Center Frequency and Bandwidth,” Proc. IEEE, 64: 1632.
7. Embedded Signal Processing with the Micro Signal Architecture, Woon-SengGan & Sen M. Kuo, Wiley, 2007.
8. The Scientist and Engineer's Guide to Digital Signal Processing By Steven.
9. Sundarraj, M., 2013. Study Of Compact Ventilator, Middle-East Journal of Scientific Research, ISSN:1990-9233, 16(12): 1741-1743.
10. Sundar Raj, M., T. Saravanan and R. Udayakumar, 2013. Energy Conservation Protocol for Real time traffic in wireless sensor networks, Middle-East Journal of Scientific Research, ISSN: 1990-923315, (12): 1822-1829.
11. Sundar Raj, M. and T.R. Vasuki, 2013. Automated Anesthesia Controlling System, Middle-East Journal of Scientific Research, ISSN: 1990-9233, 15(12): 1719-1723.

12. Sundar Raj, M., T. Saravanan and R. Udayakumar, 2013. Data Acquisition System based on CAN bus and ARM, Middle-East Journal of Scientific Research, ISSN:1990-9233, 15(12): 1857-1860.
13. Shafaq Sherazi and Habib Ahmad, 2014. Volatility of Stock Market and Capital Flow Middle-East Journal of Scientific Research, 19(5): 688-692.
14. Kishwar Sultana, Najm ul Hassan Khan and Khadija Shahid, 2013. Efficient Solvent Free Synthesis and X Ray Crystal Structure of Some Cyclic Moieties Containing N-Aryl Imide and Amide,Middle-East Journal of Scientific Research, 18(4): 438-443.
15. Monalisa Pattanayak and P.L. Nayak, 2013. Green Synthesis of Gold Nanoparticles Using Elettaria cardamomum (ELAICHI) Aqueous Extract World Journal of Nano Science & Technology, 2(1): 01-05.
16. Rajashree Chahataray and P.L. Nayak, 2013. Synthesis and Characterization of Conducting Polymers Multi Walled Carbon Nanotube-Chitosan Composites Coupled with Poly (P-Aminophenol) World Journal of Nano Science & Technology, 2(1): 18-25.
17. Umesh Kumar Parida S.K. Biswal, P.L. Nayak and B.K. Bindhani, 2013. Gold Nano Particles for Biomedical Applications, World Journal of Nano Science & Technology, 2(1): 47-57.