Design & Implementation of FIR Filters using On-Board ADC-DAC & FPGA

Prof. Sashank Shekhar Pujari, Prangya Paramita Muduli, Amruta Panda,Rasmita Badhai,Sofia Nayak, Yougajyoty Sahoo,
P.G department of embedded system design
Sambalpur University Institute of Information Technology
Jyotivihar, Burla, India.
sspujari@suiit.ac.in,
prangya.muduli@suiit.ac.in,
amruta.panda@suiit.ac.in
rasmita.badhai@suiit.ac.in
sofia.nayak@suiit.ac.in
yougajyoty.sahoo@suiit.ac.in

***Abstract*— Electronics industry is very prodigiously moving towards digital platform, but the world is analog in nature, so when any analog signal needs to be processed in digital platform it should be converted to digital with the help of analog to digital converter. After processing through digital platform by the help of DAC it will be again converted to analog format. Here the digital platform is ALTERA CYCLONE-II FPGA. FIR filters are used in every aspect of present-day technology because filtering is one of the basic tools of information acquisition and manipulation. Different types of digital FIR filters are implemented on external signal by using VERILOG HDL language over FPGA. Further model based design approach using MATLAB and Simulink is taken into account for optimized designing and resource computation.**

Keywords- ***fpga;adc;dac;fir;lpf;hpf;bpf;bsf;altera cyclone-ii ;dsp.***

I. INTRODUCTION

Signal processing has been used to transform or manipulate both analog and digital signal. Digital signal processing becomes a more mature technology after the development of programmable DSP processor. This type of digital platform is widely accepted because of its speed and accuracy. A digital platform like FPGA plays a vital role in signal processing than DSP processor because of the four basic reasons such as: ability to handle very high computational workloads ,offload compute intensive works from DSP processor, customize architecture to suit ideal algorithm, reduce system cost. So in an FPGA board a lot of operation can be performed starting from simple addition to complex calculation like s-transform and wavelet-transform. Model based design approach is utilized in this project by using MATLAB and Simulink. MATLAB helps in simulating system behavior, generating software for prototyping and production. A Simulink model is also used here for observing different types of filtering effects because it provides continuous test and verification of embedded system.

The paper is organized as follows; in section II Signal and system and FPGA as a subject is introduced. System level system description is covered in section-III. Design approach using Model based design and Simulation is described in section-IV. Section-V covers implementation of hardware and firmware .Different filter observation is tabulated in section – VI. Concluding remark and extending the scope of paper is given in section-VII followed by acknowledgement and references.

II. SUBJECT INTIATION

As the basic aim behind the project is to process signal on FPGA, so It begins with foundation study on signal and system and basics of FPGA. As FPGA need to be programmed so acquaintance with VERILOG /VHDL is also a requisite. Here VERILOG language is used, which is not highly syntactic and structural like VHDL. To write codes in Verilog, concept of basic logic design and digital system design is necessary. First small interfacings like led, LCD, keyboard should be practiced on FPGA. Lab should be pre planned and result oriented. Various vendors are providing IDE to work on FPGA such as QUARTUS-II for ALTERA DE2 kit, Xilinx-ISE web pack for SPARTAN-3E FPGA board. Xilinx and Altera provides Language template with proper syntax for efficient coding.

FDA TOOL in MAT LAB provided by Mathworks is highly useful in designing different order filter. Automatic code generation by HDL coder of MATLAB is an important procedure to learn. Design knowledge of Simulink is also essential. Again the basics of filters such as LPF, HPF, BPF, BSF is required. For different type of design requirement the help of FPGA IP core is taken.

Finally to process analog signals fundamental understanding of ADC and DAC is also required.

ISBN No.978-1-4799-3834-6/14/$31.00©2014 IEEE

III. System level Description

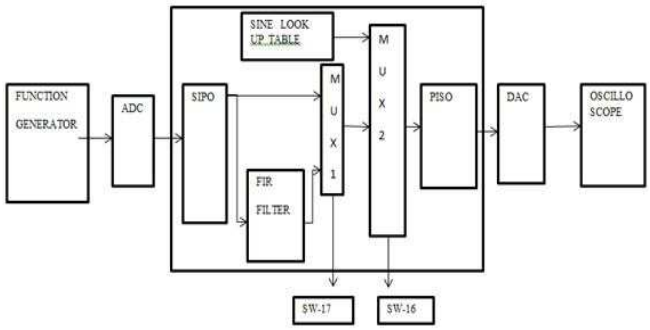

Figure-1. Top Level Block Diagram

The top level block diagram shown in figure-1 comprises of central cyclone-II FPGA interfaced with peripherals. Here the peripherals are on board audio codec WM-8731, function generator and oscilloscope. This is a basic set-up required for any type of signal processing on FPGA, be it an electrical signal or biomedical signal.

Here an external sine wave is generated using function generator and is given to ADC. Data coming from ADC are stored in serial in parallel out shift register. A MUX1 is connected to the Serial In Parallel Out(SIPO) register which is controlled by the switch 17 .The same output from the SIPO is passed through different type of basic FIR filter. The output of basic FIR Filter is again a input to the MUX2.In side FPGA block MUX are designed using Verilog program. Previously inside the ROM of FPGA a sine lookup table is there which is required for generating internal sine wave. The output from MUX1 and the sine look up table is given as input to the MUX2. This MUX2 is controlled by switch 16 present on DE2 board. The output from MUX2 is fed as input to the Parallel in Serial out (PISO) register. The output of PISO register is then converted to analog format by DAC and finally displayed on oscilloscope.

The Verilog program is designed in such a way that when switch16 and 17 are OFF (‘0’) then internal sine wave is generated, via DAC it will be displayed on oscilloscope. When switch 16 is ON (‘1’) and switch 17 is OFF (‘0’) at that time also internal sine wave is generated. When switch 16 is OFF and only Switch 17 is ON then external sine wave is displayed on oscilloscope. When both switches are in ON condition external signal coming from function generator is passed through FIR Filter and filtered output is displayed on oscilloscope.

This FIR filter can be any of the 4 FIR type of filter such as LPF, HPF, BPF and BSF. With the increase of filter order and changing its frequency and density factor of filter design the response of filter can be observed and can be refined according to our need

The peripherals are described as follows:

- FUNCTION GENERATOR:

A function generator of frequency starting from 1 KHz to 10 MHz is taken. Because, within the above range all signals come which will be analyzed using lab environment.

- OSCILLOSCOPE:

A digital storage oscilloscope is taken to display the results.

- ADC and DAC

WM8731 is a low power audio codec[2] which is present in DE2 board. .The codec includes line and microphone inputs to the on-board sigma-delta ADC, line and headphone out from on-board sigma-delta DAC, a crystal oscillator, configurable digital audio interface and a choice of 2 or 3 wires MPU serial controls interface. .Line inputs have logarithmic volume level adjustment and mute. The ADC is a high-quality multi-bit high order oversampling architecture delivering optimum performance at low power application. The output from the ADC is available in the digital audio interface. The on-board DAC accepts digital data from digital audio interface. Digital filter de-emphasis at 32 KHz, 44.1 KHz and 48 KHz can be applied to digital data under software control. The below figure-2 is the internal architecture of audio codec WM8731

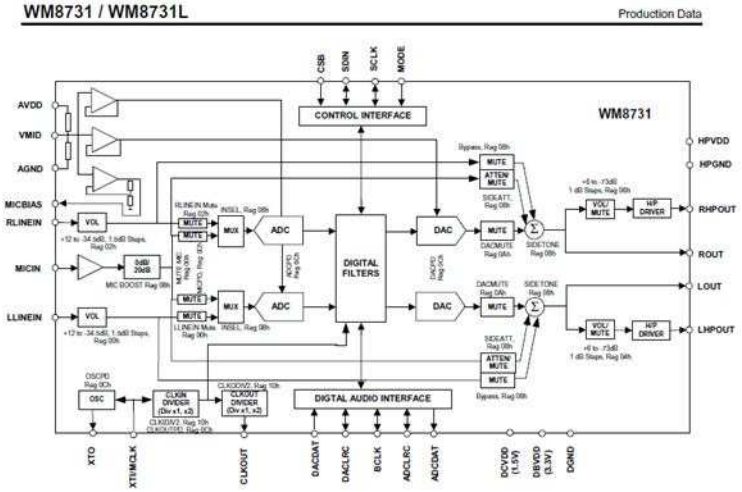

Figure-2 Internal architecture of Audio codec

The line inputs in the codec are of high impedance and low capacitance, thus very suitable for receiving line level signals. The use of multi-bit feedback and high oversampling rate in ADC reduces the effect of jitter and high frequency noise. The digital data from ADC is fed to ADC high pass filter. The ADC filter performs 24 bit signal processing to convert the raw multi-bit over sampled data from ADC to correct sampling frequency output to digital audio interface. The DAC filter also performs 24 bit signal processing to convert the incoming data from digital audio interface at the specified sample rate to multi bit over sampled data for processing by analog DAC. The DAC used here is also multi-bit sigma delta converter. Both the line output is also of low impedance which will be taken to oscilloscope for different observation.

- CYCLONE-II FPGA

ISBN No.978-1-4799-3834-6/14/$31.00©2014 IEEE

Altera's DE2 board [2]i.e. the development and education board with CYCLONE-II FPGA has been designed to provide an ideal vehicle for learning about digital logic and computer organization in a laboratory setting. All important component of the board are connected to the pins of this chip, allowing user to configure various components as desired. It is operated by 9 volt power adapter and a USB cable. Software provided with DE2 features the Quartus-ii web edition design tools.DE2 board is connected to computer via USB cable, before which USB BLASTER driver should be pre-installed. Once the connection is set-up the DE2 board is controlled by help of computer.

IV. DESIGN AND SIMULATION

The total code is designed using Quartus-ii v 7.2,[4] provided by ALTERA and the FDA TOOL provided by MATLAB, Math works. Here a structural modeling is done by component instantiation. In the top level i.e. DE2_TV all the required components are declared such as VGA_AUDIO_PLL, I^2c_AV_Configuration, Audio_DAC unit and Basic_FIR. With the help of ALTERA IP core that is VGA_AUDIO_PLL a clock of 18.4 MHz is derived. For I^2c 50MHz on board clock is used. On board audio codec WM8731 is controlled by I^2c bus interface which is connected to the pins of cyclone-ii FPGA. Inside this communication bus I^2c_controller is there which is programmed to send the data in master slave mode. A register in side this configuration block is programmed so as to change the function of audio-codec. Total 11 cases need to be programmed.

This register has 16 bit length among which B9 to B15 is for address bit and from B0 to B8 is data bit. Looking into the datasheet of WM8731 it needs to be programmed. Analog audio path should be designed by taking DAC SELECT and MIC DESELECT. Line input taken from function generator is given to ADC. The output from ADC by passing through a high pass filter goes to the digital audio interface. Digital audio interface takes data from the internal ADC digital filter and places it on the ADC DAT output. The ADC LRC is an alignment clock that controls whether left or right channel data is present on ADCDAT lines. Both ADCLRC and ADCDAT are synchronous with BCLK (bit stream clock) with each data bit transition signified by a BCLK high to low transition. Then AUDIO-DAC is so designed that external data coming out are stored in SIPO register. A sine lookup table is designed to generate internal sine wave. Data from PISO register will be entered to DAC. The digital audio interface also receives digital audio data from internal DAC digital filter in the DACDAT input. Both DACLRC and DACDAT are synchronous with BCLK and each bit transition signified by BCLK high to low transition. Finally the basic FIR filter is designed in two ways. This FIR filter includes low pass filter, high pass filter, band pass filter and band stop filter. FDA TOOL[5] in MAT LAB is a Graphical User Interface (GUI) that allows you to design or import, and analyse digital FIR and IIR filters. As you write FDATOOL in command window of MAT LAB filter design window appears. There one has to provide response type i.e. which type of filter you want to design and need to suggest the design method whether FIR or IIR filter. There by one has to provide the filter order, and different frequency specification according to the need of your design. For e.g. in this project we have taken 1 KHz as Fpass and 5 KHz as Fstop. Then clicking on design filter the required filter is designed. The designed filter is stored by giving the name of the filter on the store option from the drop down menu of file. After that filters need to be quantized. There after clicking on quantizing option specify filter arithmetic as fixed point because of ease of computation in FPGA. Then select specify all from the filter precision and click on filter internals. Then choose rounding mode as floor and overflow mode as saturate. Then click on apply option of that window. Then from the menu bar of FDA TOOL filter coefficients can be taken by exporting them to work space. Then by writing q= quantizer on command window it will directly specify Data Mode = fixed, Round Mode = floor, Overflow Mode = saturate, Format = [16 15]. Then specifying the variable name in which coefficients are stored in work space the coefficients will appear on command window in quantized format. Then by writing the command num2bin (q, variable name) it will give the coefficients in binary format.

In the model based design approach filter can be designed taking the help of MAT LAB HDL CODER. By following these steps it can be designed. After quantizing, when filter is fully designed; then from the menu bar of FDA TOOL click on target option and choose generate HDL code from the drop down menu. There by specify the language VHDL/VERILOG in which code will be generated and specify the target area at which the generated code will be stored and click on the generate option to generate the code. But you need to round the value of the coefficient.

Another method of designing filter is by taking the help of IP of Altera and there by taking filter coefficients from FDA TOOL in the following manner. First code is written in VERILOG language. Here IP of Altera will be utilized i.e. the Altmult_add. In this IP four multipliers are there so, when we are designing for 8th order filter there we need to cascade two of these IP. Similarly when designing for higher order filter required no of multipliers can be taken by cascading them. And for adding them another IP named LPM_ADD_SUB is used. One ADDER and two MAC units are used in 8thorder filter. Two adders and three MAC Units are used in 12^{th} order filter. Three adders and four MAC units are used in 16^{th} order filter.

This design is simulated using ALTERA simulation environment. By providing coefficient of 8th order filter the output waveform is first analyzed. Most of the information is lost here by 8th order filter. Again to get brick size filter(strict cut off point) filter order is gradually increased from 8 to 12,16,32 and the wave forms are analyzed.

ISBN No.978-1-4799-3834-6/14/$31.00©2014 IEEE

The following in Figure-3 is the Simulink model of Filter design. There are four main stages of the Simulink model. They are Audio source file:-Reads multi-media files containing audio, video, or audio and video data. Audio filter: it is a filter design tool which can design different type of filter and also the filter operation. Spectrum analyzer: it is the device in which frequency spectrum of signal can be observed. Headphone and speaker: the headphone block is used here for listening audio output. Different filters are changed and accordingly the output using headphone and through spectrum analyzer can be observed.

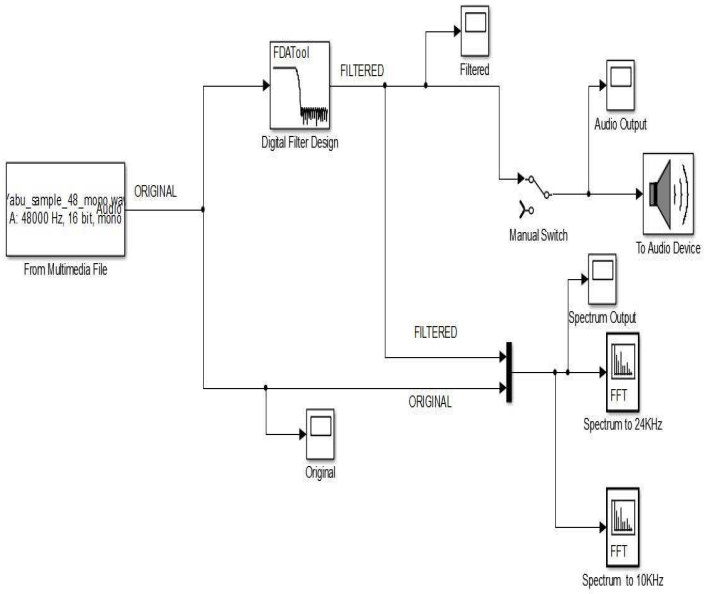

Figure-3 Simulink model of the project

In the model based design approach i.e. filtered output is observed in scope. Here different filters such as lpf, hpf, bpf, bsf filters are taken . Changes are observed from the spectrum output from different filters

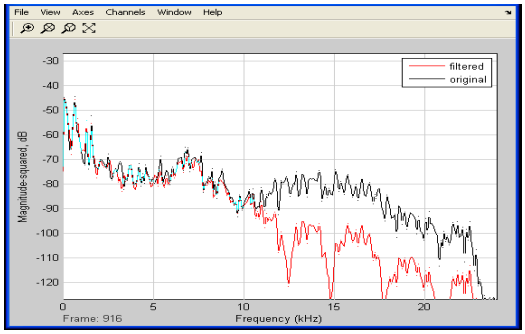

Figure-4: spectrum output of a low pass filter

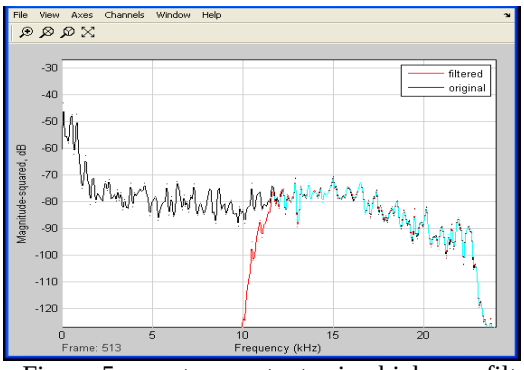

Figure-5 spectrum output using high pass filter.

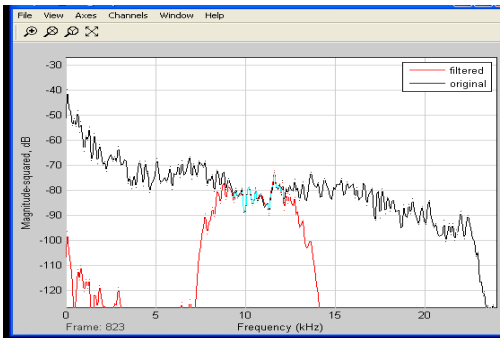

Figure-6 spectrum output using band pass filter.

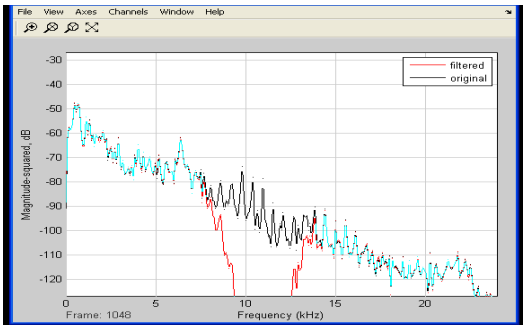

Figure-7 spectrum output using band stop filter

V. Hardware Implementation

ISBN No.978-1-4799-3834-6/14/$31.00©2014 IEEE

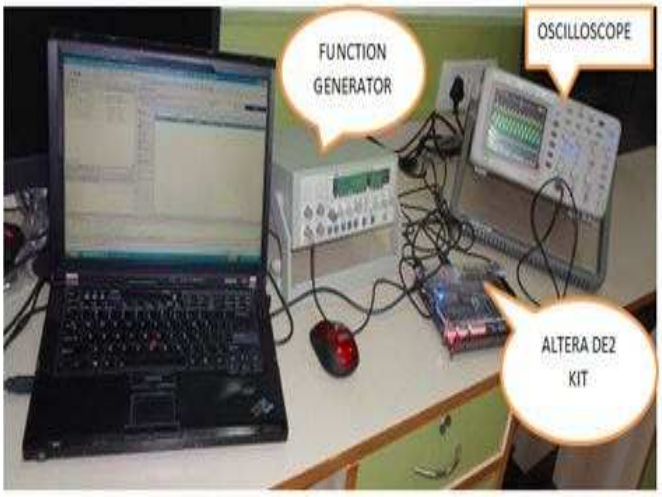

Figure-5: Hardware implementation Set-up

For the hardware set up (as Figure-5) first a signal from function generator is taken to the DE2 kit through 3.5 mm audio jack by inserting it to the line in. Similarly a jack is connected to line out. Output from line out is connected to the input of oscilloscope .FPGA is operated by 9v power supply. An USB blaster is connected to the laptop in which the required software is pre-installed. After successful compilation of the program a .SOF file will be generated. That files needs to be dumped through this USB cable by the help of programmer option of the Quartus-ii. A digital storage oscilloscope is taken to display the output properly.

VI. TESTING AND OBSEVATION

When the hardware set-up is ready according to the design then different testing and observation are taken. By switching off switch 16 and 17 internal sine wave is observed and making both on filtered sine wave will be observed. The four filters output with increasing their order (8,16,32,40) at different frequency and corresponding peak to peak voltage (Vpp) magnitude is also measured. The observations are as follows:

A. Low pass filter:

Filter is designed according to Fpass as 1 kHz and F-stop as 5 kHz. Then different higher orders are taken and magnitudes are tabulated here.

Frequency in Hz	**8th order Vpp**	**16th order Vpp**	**32nd order Vpp**	**40th order Vpp**
500	1.42V	1.64v	1.60v	840mv
1000	1.32v	1.50v	1.54v	760mv
2000	980mv	1.00v	1.12v	620mv
5000	180mv	80mv	60mv	60mv

Table No-1

B. High pass filter

This filter is also designed by taking Fstop= 1 KHz and Fpass=5 KHz

Frequency in HZ	**8th order Vpp**	**16th order Vpp**	**32nd order Vpp**
500	1.68v	120mv	160mv
1000	1.72v	160mv	120mv
2000	1.76v	240mv	160mv
5000	1.76v	560mv	240mv

Table No-2

C. Band pass filter

This filter is designed by taking Fs= 48000Hz, Fs1=7200Hz, Fs2=14400Hz, Fp1= 9600Hz, Fp2=1200Hz

Frequency in KHz	**8th order Vpp in mv**	**12th order Vpp in mv**	**16th order Vpp in mv**	**32nd order Vpp in mv**
2.0	240	40	52	32
3.0	152	88	84	30
4.5	40	124	24	28
7.2	184	124	84	56
8.5	232	256	256	288
9.0	240	296	316	368
9.6	256	332	380	392
10	256	336	392	392
11	248	336	388	380
12	224	264	304	332
13	184	180	164	148
14.4	120	56	24	28

Table No-3

ISBN No.978-1-4799-3834-6/14/$31.00©2014 IEEE

D. Band stop filter

This filter is designed by taking Fs=48000Hz, Fp1=7200Hz,Fs1=9600Hz,Fs2=12000Hz,Fp2=14400Hz

Frequency in KHz	**8th order Vpp**	**16th order Vpp**	**32ndorder Vpp**
500	**3.20v**	**720mv**	760mv
1000	3.20v	720mv	760mv
2000	2.96v	600mv	640mv
5000	1.44v	520mv	480mv
8000	-------	--------	280mv
10000	480mv	120mv	120mv
13000	-------	--------	240mv
15000	720mv	320mv	320mv
20000	1.2v	-----	-------

Table No-4

From the above readings it is clear that the above band stop filter rejects 10khz signal. As order increases the band stop filters become more accurate.---- indicates that voltage values for this frequencies are not taken .

Let us have a comparison of the used resources

There is a limitations of implementing higher order filters by using Altera IP CORES, because the required no. of multipliers are not available in the specified device. But in MATLAB generated code there is not such type of limitations of multipliers for implementing higher order filters. So also the MATLAB generated code is much convenient and accurate. The code generated by MATLAB is portable that means the same code can be used for both Xilinx and Altera platform. It also consumes less resource as well as time.

Filter order	**Ip cores(no of multiplier)**	**Mat lab generated code(no of multipliers)**
8^{th}	16/70(23%)	18/70(26%)
16^{th}	32/70(46%)	34/70(49%)
32^{nd}	64/70(68%)	54/70(77%)
40^{th}	80/70(not possible)	70/70(100%)

Table No-5

VII. CONCLUSION AND FUTURESCOPE

The above project gave a basic detailed study and implementations of Signal Processing on FPGA, and so also different types of filtering method are being experienced. The project is developed with the help of many tools and software such as Mat lab & Simulink, FDA Tools, & Altera Quartus II etc. At last a Basic Signal Chain Established using FPGA is done thoroughly. Further using this signal chain numerous application such as FFT, wavelet transform, adaptive noise, system identification can be implemented on FPGA. This signal chain establishment is highly required before processing any signal on FPGA. Further this signal can be established in any high end FPGA and different IIR filter can be designed using the above said methods.

VIII. ACKNOWLEDGEMENT

Author wish to thank S O A university for providing advance lab. Without proper guidance at this lab this project could not be completed. Author is indebted to SUIIT for providing development tools and kits. Author is also thankful to the friends for their constant inspiration and support.

REFERENCES

[1] Magrath, A.J.; Clark, I.G.; Sandler, M.B.; , "Design and implementation of a FPGASigma-delta power DAC," Signal Processing Systems, 1997. SIPS 97 - Design and Implementation. 1997 IEEE Workshop on , pp.511-521, 3-5 Nov 1997.

[2] ALTERA," Audio Codec WM8731".

[3] ALTERA, "DE2 FPGA starter kit Board User Guide", Internet: http://www.altera.com/ Support/documentation/boards_and_kits/

[4] ALTERA,"Quartus-ii_v7.2 user manual".

[5] User guide of "Filter Design HDL Coder", Internet: http://www.mathworks.com/ documentation center

[6]Design example from www.altera.com/support

[7] Cheung, R.C.C.; Pun, K.P.; Yuen, S.C.L.; Tsoi, K.H.; Leong, P.H.W.;, "An FPGA-based Re-configurable 24-bit 96 kHz sigma-delta audio DAC," Field-Programmable Technology (FPT), 2003. Proceedings. 2003 IEEE International Conference on, vol., no., pp. 110- 117, 15-17 Dec. 2003

ISBN No.978-1-4799-3834-6/14/$31.00©2014 IEEE