Field Programmable Gate Array Implementation and Testing of a Minimum-phase Finite Impulse Response Filter

P. K. Gaikwad
Department of Electronics
Willingdon College,
Sangli, India
e-mail: pawangaikwad2003 {at} yahoo.co.in

Abstract— **The present paper depicts a prototype developed to test and verify functioning of digital filters easily. The minimum-phase Finite Impulse Response (FIR) filter was a target-filter to develop such a system. The experimental setup was technologically advanced in such a way that, it compares the magnitude and phase responses of a filter; shown by Frequency Visualization tool (*fvtool*) of the MATLAB software with the same filter responses observed on a Digital Storage Oscilloscope in real-time operations. For this research work, an important facility provided in the MATLAB software to generating Hardware Description Language (HDL) for a given filter objects was deployed. A Very High Speed Integrated Circuit HDL (VHDL) soft Intellectual Property (IP) core was generated for the minimum-phase FIR filter, and instantiated it in a top level entity, along with a clock divider VHDL module; capable of providing proper sampling clock to the filter. For implementation of this core, the Xilinx Field Programmable Gate Array (FPGA) Virtex-5 device was deployed. Around this reconfigurable device, there was an Advanced RISC Machine (ARM) microcontroller, to provide inter-conversion between Analogue and Digital signals; essential for the filter testing.**

Keywords-FPGA; minimum-phase FIR filter; MATLAB; fvtool; ARM microcontroller

I. INTRODUCTION

As a digital filter designer uses the higher filter-order to achieve a minimum transition width, then the bottleneck arises that, filter response delay becomes half of the filter-order. In many control applications such delay is not tolerable. However, in such cases, a smaller group delay can be realized in the pass-band region by using minimum-phase Finite Impulse Response (FIR) filters, which is defined as: ‘a filter obtained by selecting the zeros to lie on or inside the unit circle’[1].

A filtering process in a control loop typically requires the response to an input stimulus to be as quick as possible. A large delay in the filtering process can cause a negative feedback control loop to become unstable. Minimum phase filters can achieve better results in terms of a lower ripple or lower order for a given magnitude specification than equivalent linear phase filters. Therefore, minimum phase can be a better choice than linear phase in applications where the phase response is not constrained [2].

In view point of application of minimum-phase filters the problem of modeling head-related transfer functions (HRTFs) is addressed in [3]. Traditionally HRTFs are approximated in real-time applications using minimum-phase reconstruction and various digital filter design techniques, yielding FIR or IIR structures. In that work, binaural auditory modeling has been applied to HRTF analysis, and filter design methods have been compared from the auditory perception point of view. Applicable perceptually valid filter design techniques were presented, and listening test results for localization and timbre degradation were discussed in [3].

The present research work depicts the real-time performance analysis of the minimum-phase FIR filter. It was verified against its response shown by the Frequency Visualization Tool (*fvtool*); available with the MATLAB software. For hardware implementation of this filter an outstanding facility of MATLAB has been deployed; which generates Hardware Description Languages (HDLs) like Very High Speed Integrated Circuit HDLs (i.e. VHDL) or Verilog for the filter object, specified in the MATLAB file. Therefore, a VHDL source code was generated for minimum-phase FIR filter using *generatehdl* instruction in MATLAB file [4]. It was further instantiated in a top level VHDL module. In addition to the filter core, the top level entity was including a clock divider module; which provides a desired sampling clock to the filter.

To simulate the filter module a VHDL entity called as test bench (producing stimulus for the filter module) was generated using the same MATLAB file [4]. Further, the top level VHDL entity (including the filter and clock divider modules) was synthesized using Register Transfer Level (RTL) synthesis process of an Electronic Design Automation (EDA) tool from Xilinx Integrated Software Environment (ISE). A bit-stream file having *.bit* extension was generated using the *generate program file* process of Xilinx ISE design flow, to download it into the FPGA device via the USB cable, attached to the Personal Computer.

The MATLAB file [4] provides digital filter’s world-lengths of 10 bit for both input and output lines. Therefore, a

microcontroller, containing Analogue to Digital Convertor (ADC) was to provide a 10 bit digital input to such a filter. The filtered digital output from FPGA was fed-back into the same microcontroller chip; having a 10 bit Digital to Analogue Converter (DAC). Both of the analogue waveforms (i.e. input and output to the filter), were plugged into the Digital Storage Oscilloscope (DSO) and thus tested the functioning of FIR filter of minimum-phase response.

II. Filter Response Analysis using MATLAB Frequency Visualization Tool (FVTOOL)

The MATLAB file [4] provides various FIR-filter design parameters, like sampling frequency, Nyquist criteria, pass-band frequency range, phase selection; required in the *firgr* instruction. It also provides set of instructions for filter-response analysis using Frequency Visualization Tool (*fvtool*). The Fig. 1 illustrates magnitude response (in dB) for a minimum-phase FIR filter.

There are two imperative points of the response curve on either sides of the pass-band, revealing the upper and lower cut-off frequency values; associated with their –3dB magnitudes. The lower cut-off frequency provided in the MATLAB file [4] is 19 KHz; whereas, in Fig. 1 it was marked on 18.97 KHz (approximately 19 KHz). Another marking point is along the upper cut-off frequency of 29 KHz.

The phase response, with respect to the frequency is also shown in Fig. 2. It indicates that, for a lower cut-off frequency (19.00 KHz) the filter produces the phase response of 3.088 radians. On the other hand, the phase response of -3.180 radians was observed at the upper-cut-off frequency (29.01 KHz). A random frequency of 23.23 KHz (from the pass-band) was selected to note the phase response, and it shown 0.28 radians. It discloses that, during the pass-band, there was a decrease in the phase-response with respect to the increase in the input frequency. On the contrary, there are more phase-variations, during outside the pass-band.

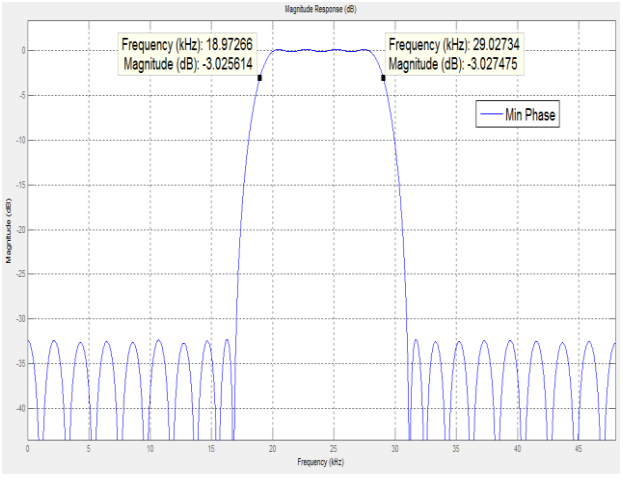

Figure 1. Magnitude response analysis curve for minimum-phase FIR filter

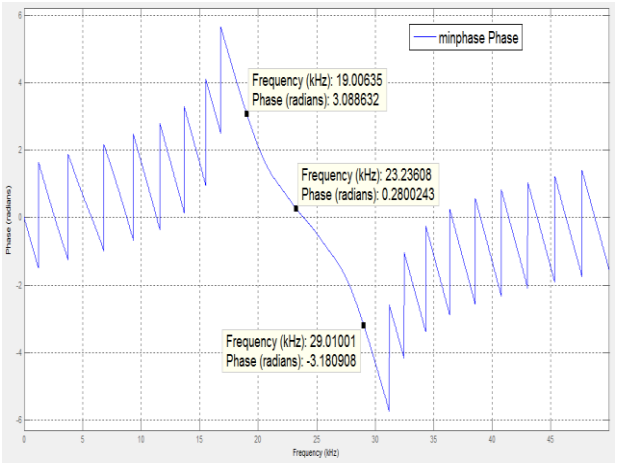

Figure 2. Phase response of minimum-phase FIR filter

The collection of simulation models is commonly called as *testbench* [5]. The MATLAB file [4], not only provides generating the VHDL code for a filter object, but also generates a testbench (another VHDL source code) to simulate the VHDL-filter module. The Fig. 3 illustrates the simulation result of a minimum-phase FIR filter, designed in VHDL. It depicts that, only during the pass-band frequencies (19 KHz to 29 KHz), the significant values of input magnitudes are produced at the output of the filter.

III. Development of Soft IP Core for FPGA Implementation of Minimum-phase FIR Filter

An astonishing facility provided in MATLAB is *generatehdl*. It is applicable for those who are interested not only too stuck-up with the simulations; but are eager to hardwire the filter-objects into the reconfigurable devices like Field Programmable Gate Arrays (FPGAs), and monitor their real-time performance. In the present research work, a VHDL entity was generated using *generatehdl* command in the MATLAB file [4].

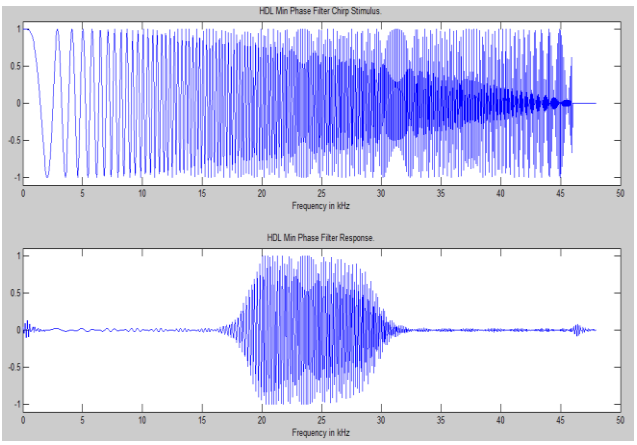

Figure 3. Simulation results obtained by using VHDL testbench; that provides input stimulus to the VHDL filter module

It requires mentioning the corresponding HDL that is, VHDL or Verilog, as a part of the *generatehdl* instruction itself. In this work, VHDL source code was chosen for the minimum-phase FIR filter object, and MATLAB file [4] was executed. A VHDL entity containing 10 bit input and 10 bit output for the digital filter was produced. It also includes some special control signals like *reset* and *clock-enable*. It has *sampling-clock* to sample the digital data settled at the filter input. Following are the lines of VHDL code, illustrating the entity structure of minimum-phase FIR filter.

```
ENTITY hdlminphasefilt IS
PORT(  clk          : IN   std_logic;
       clk_enable : IN   std_logic;
       reset        : IN   std_logic;
       filter_in    : IN  std_logic_vector(9 DOWNTO 0);
       filter_out   : OUT std_logic_vector(9 DOWNTO 0) );
END hdlminphasefilt;
```

The sampling clock of 100 KHz was chosen in MATLAB file [4]. Therefore, the same frequency signal was provide at the *clk* input of this entity. However, the Xilinx FPGA Virtex-5 board, developed by Digilent Inc., provides the on-board clock source of 100 MHz. To generate the sampling frequency signal for the filter module, another clock divider VHDL core, as given in [6], was deployed. The clock divider and filter entities, both were integrated and instantiated in a top level VHDL soft Intellectual Property (IP) core named as *allphasefilters*.

The top level VHDL core contains two components named as *clkDivMain* and *hdlminphasefilt*; connected to the top level entity *allphasefilters* using the VHDL *port map* commands. A clock input of 100 MHz; associated to the top level module was connected to the clock input of clock divider module *clkDivMain*. Output of this entity was further connected to an internal signal clk100kHz_int of entity *allphasefilters*. Thus, output of clock divider entity generates a 100 KHz signal when 100 MHz is input to it. The 100 KHz clock was directly given as a filter's sampling clock input. Such an arrangement of 100 KHz clock to the filter was made, because sampling frequency of 100 KHz was chosen during the filter designing in the MATLAB file [4] itself.

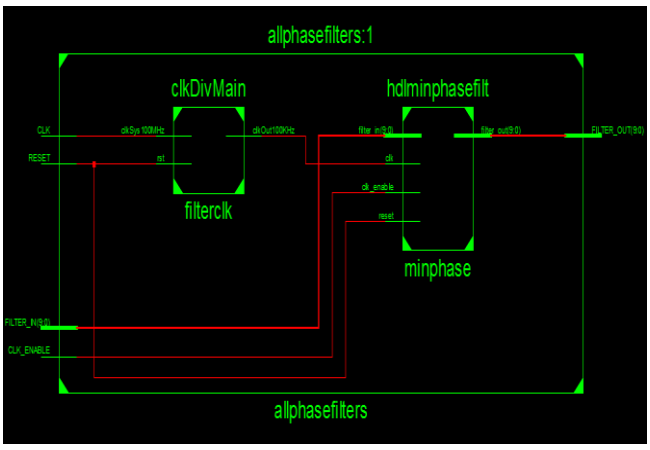

Figure 4. Register Transfer Level RTL view of VHDL entity allphasefilters

IV. Synthesis Results of the Soft IP Core Developed for Implementation of Minimum-phase FIR filter

Synthesis is the process of constructing a gate level netlist from a register-transfer level model of a circuit described in a HDL [7]. Therefore, prior to the implementation of soft IP core into the FPGA device, a synthesis process of an Electronic Design Automation (EDA) tool, from Xilinx Integrated Software Environment (ISE) was performed. There were various synthesis results produced by Xilinx ISE tool for the minimum-phase FIR filter code. Two of them are reported in this research paper.

A. RTL Level Synthesis

The Register Transfer Level (RTL) Synthesis results of the VHDL entity named as *allphasefilters*, is as shown in Fig. 4. It comprises two internal cores like *clkDivMain* and *hdlminphasefilt*; whose instance names are *filterclk* and *minphase* respectively. The Fig. 4 also displays that, the output of *clkDivMain* module, named as *clkOut100KHz* is given to *clk* input of *hdlminphasefilt* core.

B. Synthesis Report

The Advanced HDL Synthesis Report and Device utilization summary of the top level module *allphasefilters* are shown in Table I and Table II, respectively.

V. FPGA Interfacing with Microcontroller for Real-time Testing of the Filter

The analogue signals were monitored at the input and output of digital filter by converting them into their digital equivalents, using an Advanced RISC Machine (ARM) microcontroller device LPC2148. A frequency generator from laboratory was installed to provide analogue signals at the input of 10 bit ADC; hosted in the microcontroller. The 10 bit digital signal was hence generated and fed to input of Xilinx FPGA Virtex-5 device; embedded on the development board, provided by Digilent Inc. The details of pin connections between ARM7 device and FPGA are illustrated in Fig. 5. The microcontroller ADC output was taken at its General Purpose Input Output (GPIO) pins from P1.16 through P1.25.

TABLE I. Advanced HDL Synthesis Report with Macro Stastics for Top Level Module

# Multipliers	44
10x15-bit multiplier	44
# Adders/Subtractors	45
11-bit adder	1
36-bit adder	44
# Counters	1
16-bit up counter	1
# Registers and F/Fs	1561

TABLE II. DEVICE UTILIZTION SUMMARY FOR XILINX FPGA VIRTEX-5 DEVICE: 5VLX50TFF1136-3

Slice Logic Utilization:	**Used**	**Available**	**Percentage**
Number of Slice Registers	1358	28800	4%
Number of Slice LUTs	3083	28800	10%
Number used as Logic	3083	28800	10%
Slice Logic Distribution:			
No. of LUT F/F pairs used	3095	--	--
No. with an unused F/F	1737	3095	56%
No. with an unused LUT	12	3095	0%
No. of fully used LUT-FF pairs	1346	3095	43%
No. of unique control sets	3	--	--
IO Utilization:			
Number of IOs	23	--	--
Number of bonded IOBs	23	480	4%
Specific Feature Utilization			
No. of BUFG/BUFGCTRLs	2	32	6%
Number of DSP48Es	34	48	70%

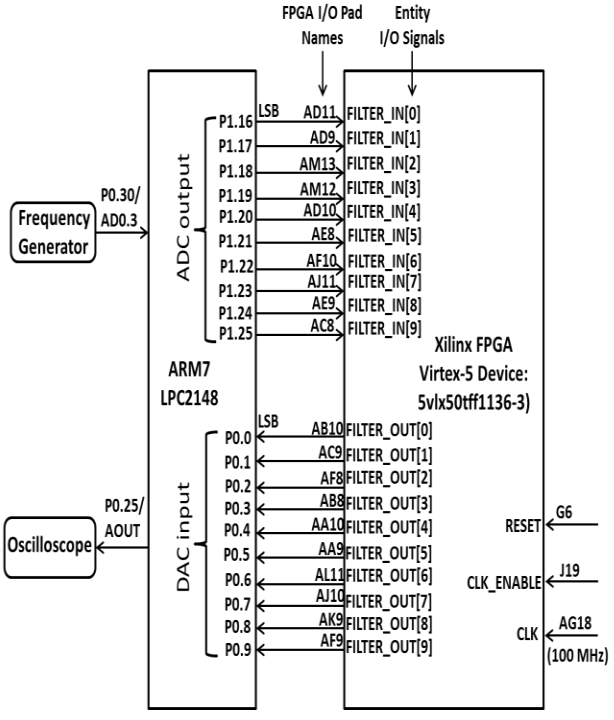

Figure 5. Interfacing details of Advanced RISC Machine (ARM) microcontroller with the Xilinx FPGA Virtex-5

The 10 bit data from ADC was further provided as stimulus for digital filter signals, named as FILTER_IN[0] through FILTER_IN[9] at the FPGA Input Output (I/O) pads, as shown in Fig. 5. The minimum-phase FIR filter response was generated at the FPGA output signals FILTER_OUT[0] through FILTER_OUT[9], emerging from the entity *allphasefilters*. To monitor the digital-filter output it did convert back into its analogue equivalent by means of the Digital to Analogue Converter (DAC) comprised in the same microcontroller device. The details of such analogue signals were studied on a Digital Storage Oscilloscope (DSO), in terms of magnitude and phase response of the filter.

VI. FINAL RESULTS OF TESTING THE ENTIRE SETUP

As given in [4], the minimum-phase FIR filter pass-band is 19 KHz to 29 KHz, and the same response was also observed using frequency visualization tool (*fvtool*), as shown in Fig. 1. The phase variation during the same pass-band is also shown in Fig. 2. The present research work proves and experiences the similar results by means of observing these responses on an oscilloscope. As shown in Fig. 1, during the pass-band, there is no magnitude attenuation as such; however at the upper cut-off frequency (29 KHz), the magnitude was attenuated by the order of -3dB. The present research work is for the development of a prototype; which can be useful to verify such filter-performance, against their simulation results; shown by frequency visualization tool of the MATLAB. For that, the author gives some snaps taken at the time of actual testing of the minimum-phase FIR filter; implemented in FPGA, and tested it by means of a microcontroller ARM.

The particular frequency of 23 KHz was tuned on the analogue signal-generator to find the similarity between two phase responses, like the one marked in the Fig. 2, and another was calculated from the oscilloscope itself. Fig.6 shows filter-input on channel 1 (CH1) of the DSO, and channel 2 (CH2) is used to display the signal pertaining filter-output.

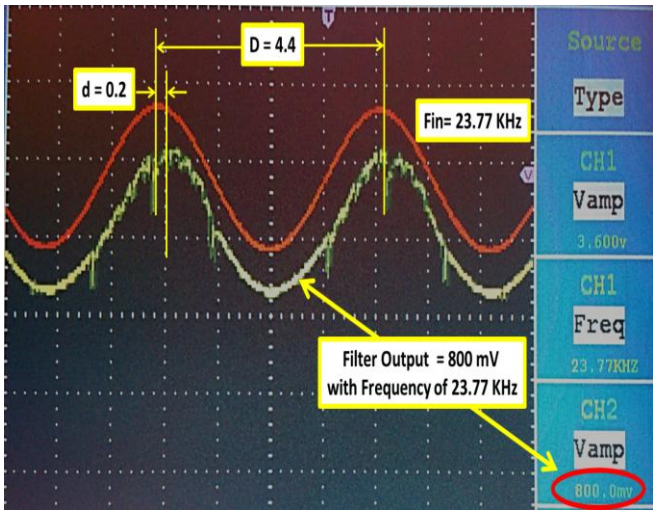

Figure 6. A photograph showing the filter input, output waveforms observed on oscilloscope, during the pass-band

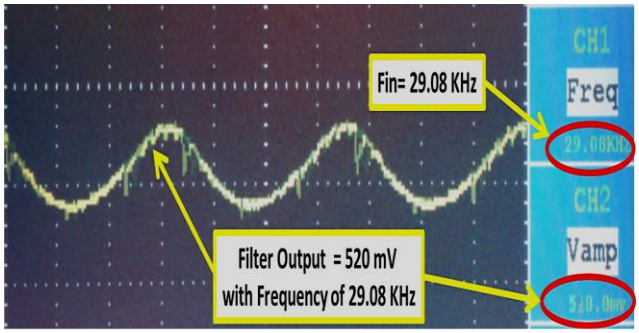

Figure 7. A photograph showing the filter output voltage relating to -3dB attenuation to that of pass-band frequency

As shown in Fig. 6, one complete oscillation (designated as D) of the input signal takes 4.4 divisions on the scale of the oscilloscope. On the other hand, the difference between peaks of input and output waveforms (named as d) is of the order of 0.2 divisions. The phase difference between input and output signals can be calculated as: (d/D) x 360 in degrees. Substituting values of D and d, it results (0.2/4.4)x360 = 16.36 degrees. Converting this value into its radians, it gives 0.28 radians; which resembles the value of phase response pointed at the frequency 23 KHz as shown in Fig.2. Hence the phase response minimum-phase FIR filter observed on DSO was compared with the one shown by *fvtool* of the MATLAB software.

To analyze the magnitude-response of the filter, Fig.6 and Fig.7 were compared. As shown in Fig.6, the magnitude-response of the filter for input frequency 23.77 KHz (i.e. during its pass-band) is 800 mV. On the other hand, at the upper cut-off of 29 KHz, the magnitude-response of the filter is 520 mV, as depicted in Fig. 7. Also, the -3dB value of the figure 800 mV can be calculated by multiplying it with 0.707; which results to 565.6 mV. This is close to the -3dB attenuated magnitude value (520 mV), shown in Fig.7.

To compare the performance of minimum-phase FIR filter with other FIR filters, the MATLAB file [4] was modified, by changing the program code line; as given below.

```
b  = firgr(44, f, a, w, 'minphase');
```

The details of variables in the above code line are explained well in the MATLAB file [4] itself.

In this code line, the word ‘minphase’ was replaced by ‘Ormsby’, ‘maxphase’, and ‘linearphase’; pertaining ‘zero-phase, ‘maximum-phase’, and ‘linear-phase’ FIR filters, respectively. It was found that, their phase-responses are different, during the same pass-band. The Zero-phase FIR filter produces a phase-response of 3.59 radians to -9.88 radians, during the lower cut-off to the upper cut-off frequencies. The maximum-phase FIR filter shows -1.23 radians to -23.86 radians phase variations, during the pass-band. The linear-phase FIR filter gives -1.13 radians to -14.96 radians phase response from 19 KHz to 29 KHz frequency band, that is, the pass-band.

ACKNOWLEDGMENT

The present paper deals with the development of a simple analysis prototype for digital filters. The MATLAB software played a vital role in this research paper by providing *generatehdl* command. The Xilinx FPGA Virtex-5 board; developed by Digilent Inc. was deployed easily for this research work; as it provides not only hardware resources on the board, but also there are various free soft IP cores available for testing its associated interfacings modules.

REFERENCES

[1] “Design of Minimum-Phase FIR Filters”, Viewed 20 May 2013 from http://www.cs.tut.fi/~ts/FIR_minimumphase.pdf.

[2] Help Document, “Minimum and Maximum Phase FIR Design (Digital Filter Design Toolkit)”, National Instruments Corporation, (2011), Viewed 28 May 2013, from http://zone.ni.com/reference/en-XX/help/371325F-01/lvdfdtconcepts/fir_mag_approx/.

[3] Huopaniemi, Jyri; Zacharov, Nick; Karjalainen, Matti, “Objective and Subjective Evaluation of Head-Related Transfer Function Filter Design”, AES E-Library, (April 1999), JAES Volume 47, Issue 4, pp. 218-239.

[4] MathWorks Document, “HDL Minimum Phase FIRT Filter”, Documentation Centre, Viewed 25 April 2013, from http://www.mathworks.in/help/hdlfilter/examples/hdl-minimum-phase-firt-filter.html.

[5] VHDL, Viewed 5 June 2013, from http://en.wikipedia.org/wiki/VHDL.

[6] Support Document, “Xilinx ISE demo project for the PmodAD2 and a PmodDA1”, Digilent, Inc., Pullman, WA 99163-0428, November 2011, DSD-0000321.

[7] J. Bhaskar, “Verilog HDL Synthesis A Practical Primer”, Star Galaxy Publishing, Allentown, PA 18103, (1999), p.1.