FPGA BASED HARDWARE LEVEL ANALYSIS OF INVERSE SINC FILTERS

P. K. Gaikwad

Department of Electronics, Willingdon College, Sangli, (M.S.), INDIA

pawangaikwad2003@yahoo.co.in

Abstract

This paper focuses on effect of cut-off frequency on the transition curve of the Inverse Sinc filter. The MATLAB Frequency Visualisation Tool, fvtool, was used to analyze the transition bandwidth variation in response to the change in cut off frequency of the digital filter. One of the imperative facilities provided in MATLAB software, to generate the Hardware Description Language (HDL) for the filter object i.e. '*generatehdl*' command, was used and the digital filter was hardwired in a reconfigurable device: Field Programmable Gate Array (FPGA). Using this MATLAB function the Soft Intellectual Proprietary (IP) Core was generated in Very High Speed Integrated Circuit Hardware Description Language (VHDL) and instantiated it in a top level module. For testing purpose of the filter at analogue front, using Digital Storage Oscilloscope (DSO), the Advanced RISC Machine (ARM) device was interfaced with the Xilinx FPGA Spartan 3E board: Nexys2. Every time the filter's cut off frequency was varied and new VHDL code was generated to implement in the FPGA and test it at the analogue front. The frequency response curves for such filters were analyzed at software as well as hardware levels. It was observed that, moving back to the cut off frequency from desired, there was a diversification from expected behaviour of the Inverse Sinc filter.

***Keywords**:* ARM Microcontroller, FPGA, fvtool, Inverse Sinc, MATLAB.

1. Introduction

The present paper deals with Field Programmable Gate Array (FPGA) implementation of Inverse Sinc filter. The paper also focuses on creating Very High Speed Integrated Circuit Hardware Description Language (VHDL) code using MATLAB tool. The Soft Intellectual Proprietary (IP) Core was generated by *generatehdl* command provided in MATLAB software. The source code for the filter generated by MATLAB tool was instantiated using structural modelling style of the VHDL architecture. As a part of the Xilinx Integrated System Environment (ISE) design flow, the filter core was further synthesised and implemented to generate the bits-stream for the project. The bit file was further downloaded in the Xilinx FPGA device Spartan 3E for real time verification of behavioural study of the filter core.

The reason behind choosing the Inverse Sinc filter for its realization in FPGA is that it has various applications to design an embedded system that requires enhanced high frequency performance of the filter. To obtain the analogue signal from the output of a digital filter the Digital to Analogue Converters (DACs) are deployed. However, as reported by (KEN YANG, 2006), the DAC's frequency response is not flat; it attenuates the analogue output at higher frequencies. At 80% of $f_{NYQUIST}$, for instance ($f_{NYQUIST}$ $f_S/2$), the frequency response attenuates by 2.42 dB. That amount of loss is unacceptable for some broadband applications requiring a flat frequency response. Fortunately, however, several techniques can cope with the non-flat frequency response of a DAC. These techniques include increasing the DAC's update rate using interpolation techniques, pre-equalization filtering, and post-equalization filtering, all of which reduce or eliminate the effects of the Sinc roll-off.

The present work concentrates on study of effect of cut off frequencies on the transition curve. The *.m* file provided in (Documentation Centre, 2013) was modified to vary the filter's cut-off frequency and generated different HDL codes. Using MATLAB frequency visualization tool, fvtool, it was observed that, when cut off frequency was a minimum, the frequency response curve was stable during the pass band. Also there was a slow rolling-off from near 0 dB to -66dB. On the contrary, when cut-off frequency was increased, there was a fast roll-off taking place, but its pass band reason was not stable as compared with the low cut-off frequency response curve, also there was increase in the magnitude during the same band. The frequency curves were monitored on fvtool, available in the MATLAB software. The same responses were observed at the hardware front. Every time, when there was a change in cut-off frequency, the separate VHDL code was generated using MATLAB. And such codes were implemented in FPGA for their realization. For verification of the working of these all filters; with different cut-off frequencies, their frequency responses were observed on digital storage oscilloscope (DSO).

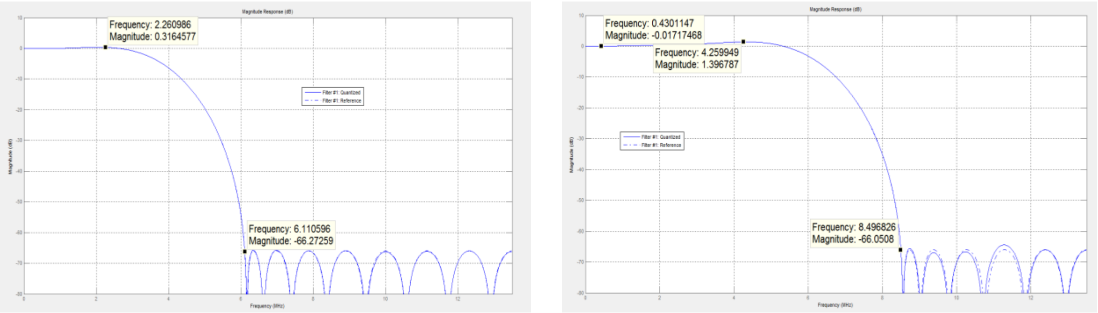
2. Frequency Response Curves for Inverse Sinc Filters with Cut-off Frequencies

As reported by (Kah-Howe Tan; Wen Fung Leong; Kadam, S.; Soderstrand, M.A. 2001), the MATLAB provides optimized HDL source code and facilitates savings ranging from 36% to 53% in FPGA resources which is achieved through a filter design program that simultaneously applies optimum scaling, careful selection of filter order and use of fixed-coefficient multipliers designed with CSD and/or DM techniques. The output of the program is a VHDL description of the optimized hardware that is suitable as input to the Synplify Pro computer program that generates highly optimized FPGA circuits for Xilinx and other FPGA's.

The MATLAB file (.m file) provided in (Documentation Centre, 2013) was used to design the Inverse Sinc filter as well as to perform re-quantization of the filter. The '*generatehdl'* command was used to create the VHDL source code for the same filter. The cut-off frequency in the filter code was varied from 2.26MHz to 7.26MHz and response curves for these filters were studied using fvtool, provided in the MATLAB software. The Figure 1 (a) through Figure 1 (d) shows variations of the cut-off frequencies and their response curves observed using fvtool from MATLAB.

It was observed that when cut-off frequency was 2.26 MHz, the pass band magnitude was 0.316 dB. As per the *.m* file provided in (Documentation Centre, 2013), the Stop band Attenuation was set to -66 dB, and in Figure 1(a), at this attenuation point, it is shown at the input frequency of the order of 6.11 MHz. It is the trailing point of the transition bandwidth of the response curve.

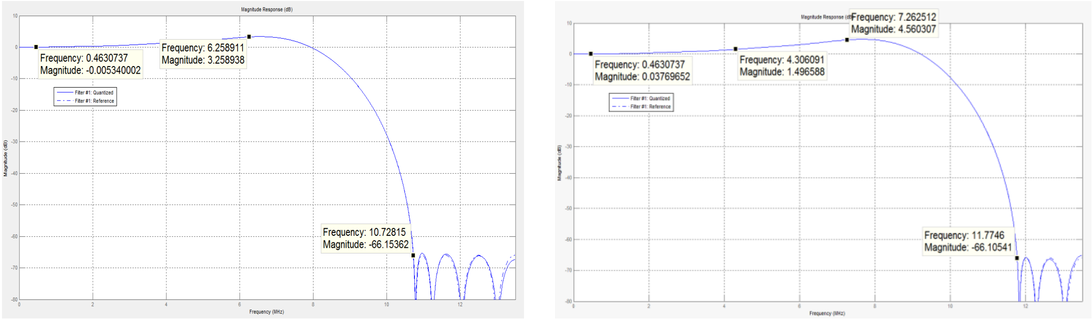
Figure 1 (b) illustrates that the response is slight modified due to increase in the cut-off frequency from 2.26 MHz to 4.26MHz. Its -66 dB attenuation was noted at 8.49 MHz. The transition from cut-off frequency to maximum attenuation was comparatively faster than the earlier case. The same situation took place while moving from 4.26 MHz to 6.26 MHz as shown in Figure 1 (c). The cut-off frequency of the order of 7.26 MHz was selected to analyze the frequency response curve as shown in Figure 1(d).



(a) Fc=2.26 MHz

(b) Fc=4.26 MHz

© 2013, IJCSMA All Rights Reserved, www.ijcsma.com



(c) Fc= 6.26 MHz

(d) Fc= 7.26 MHz

Figure 1: Inverse Sinc Filter Response Curves for Different Cut-off (Fc) Frequencies

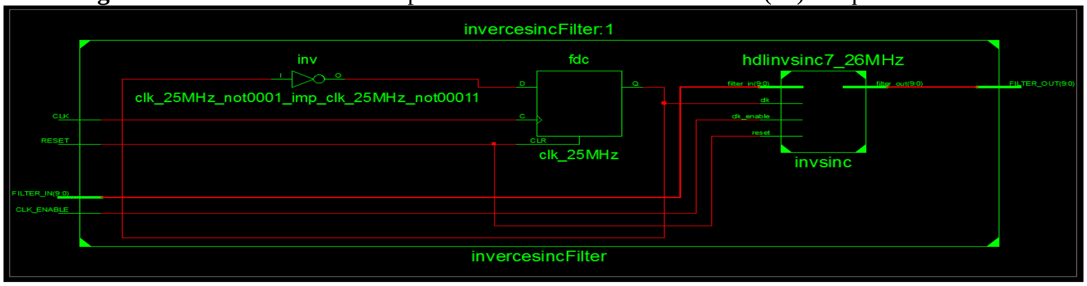


Figure 2: RTL Synthesis View of the Inverse Sinc Filter's VHDL Code

3. Synthesis Result of Digital Filter Soft IP Core

For every change in the filter's cut-off frequency, the new VHDL codes were generated. Such Soft IP Cores were instantiated in a top level VHDL module; one by one, and the process of synthesizing the code as well as performing the implementation was done successfully. There after the *.bit* file was generated using the 'Generate Programming File' option available in the Xilinx ISE Project Design flow. The Figure 2 illustrates Register Transfer Level (RTL) view of the synthesis results obtained by Xilinx ISE, 14.6 Version.

The Inverse Sinc filter Soft IP Core (hdlinvsinc7_26MHz), was instantiated in top level VHDL module; named as 'invercesincFilter' as depicted in Figure 2. The sampling frequency of 25 MHz was applied to the filter core. As given in a (Reference Manual, 2011), the FPGA board developed by Digilent Inc., is capable of providing onboard clock source of the order of 50MHz. On the other hand, the filter core requires the sampling clock source of 25 MHz. To achieve this necessary clock source of 25MHz, a clock divider was designed in the top level VHDL entity. On rising edge of the onboard clock (of 50 MHz), the internal signal declared in the top level entity was logically inverted, meaning thereby, the clock of half of the 50MHz frequency was generated and provided to the filter's module. To obtain more sharp cut-off and faster roll-off, the maximum cut off frequency could be set in the MATLAB file (Documentation Centre, 2013). But to follow the Nyquist theorem and design guide lines given in (Documentation Centre, 2013), the cut-off frequency was increased up to 7.26MHz.

The Figure 2 shows a RTL synthesis result, which depict a 10-bits output signal (FILTER_OUT); emerging from the filter core output. It has the input signals named as 'CLK', 'RESET', and 'CLK_ENABLE'. The 'CLK' is a global clock input to the top level entity. 'RESET' signal is an asynchronous signal used to reset the system; regardless of the events on the inputs, or clock, the system get reset. When the signal 'CLK_ENABLE' asserts to high, it enables the filter core to accept the input data from signal 'FILTER_IN'.

4. Hardware Implementation of the Filters for Testing at Analogue Front

The bitstream of the top level entity was generated and downloaded into the FPGA on-board's flash PROM. To perform real time testing of the digital filter at the analogue front, the Advanced RISC Machine (ARM) microcontroller device LCP2148 was deployed. Its analogue to digital converter, ADC and Digital to Analogue

Converter (DAC), both were set to perform 10 bit conversions. The details of the hardware setup essential to interface ARM7 microcontroller and FPGA Spartan3E board, Nexys2 are given by (Pawan K. Gaikwad, 2013). The Table 1 shows the magnitudes of gain in decibel (dB); obtained for different filters with different cut-off frequencies. Each filter was tested for the two input frequencies in association with the marking points shown in Figure 1. The first marking point on the response curve was at the cut off frequency, and the later one was on the attenuation point of -66 dB. In real time testing of these filters, filters have shown their performance; very close to the one shown by MATLAB fvtool. For example, as shown in Table 1, when input frequency was set to near cut off of the order of 2.2 MHz, the magnitude was observed using the DSO, and further the gain was calculated which result to 0.32dB. And to get tuned-up with the -66dB attenuation, it was necessary to set the input frequency about 6 MHz. The same phenomenon was continued for other filters with different cut-off frequencies. The summary of these filters is given in the Table 1.

Table 1: Variation of the Filters Gain Magnitudes (dB) for different cut off frequencies

Fc=2.26MHz		*Fc=4.26MHz*		*Fc=6.26MHz*		*Fc=7.26MHz*	
Fin (MHz)	Gain (dB)	Fin (MHz)	Gain (dB)	Fin (MHz)	Gain (dB)	Fin (MHz)	Gain (dB)
2.2	0.32	4.2	1.35	6.2	3.1	7.2	4.6
6.0	-65.7	8.5	-65.4	10.5	-65.3	12.0	-67.2

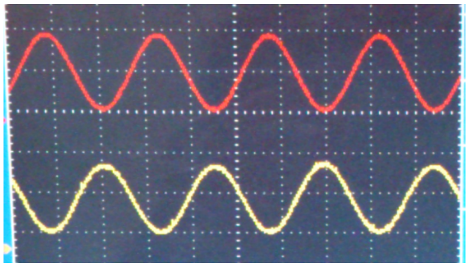


Figure 3: A photograph taken at the time of testing the filter at analogue front

Taking a horizontal look at Table 1, it reveals that, there is an increase in the magnitude for the increase in the cut off frequencies. In general observations during the pass band of a digital filter, there should be no hike in the magnitude beyond 0dB. In other words, the filter does not work as an amplifier, for any increase in the cut-off frequency. However, while designing the Inverse Sinc filter and monitoring its response curve, this hike in the magnitude becomes natural. Because the normal Sinc filter response gives decay in the amplitude on its cut-off frequency. And Inverse Sinc filter produces exactly opposite to the Sinc filter response curve. But if there is decrease in the desired cut-off frequency, the response curve would not show nature of the Inverse Sinc. This is what the main observation of the research work carried out here. The Figure 3 illustrates the waveforms during one of the tests carried out in the real time observations on a DSO.

4. Conclusion

The research paper shows the behaviour of the Inverse Sinc filter by focusing on the transition bandwidth; especially when there is a change in cut-off frequency. The role of the Sinc filter is to start dropping the magnitude of the gain, however in the Inverse Sinc filter it starts increasing this magnitude; when input frequency reaches the cut-off. But moving back from the cut-off; instead of the desired frequency, it changes the property of the Inverse Sinc filter. The FPGA based hardware implementation was possible due to the '*generatehdl*' function available in MATLAB tool. To verify the working of such filters, the ARM microcontroller's ADC and DAC; available on the single chip itself, were deployed in this research work.

References:

[1]Documentation Centre, 2013,"HDL Inverse Sinc Filter", MathWorks, retrieved from http://www.mathworks.in/help/hdlfilter/examples/hdl-inverse-sinc-filter.html
[2]Kah-Howe Tan; Wen Fung Leong; Kadam, S.; Soderstrand, M.A. 2001, "Public-domain Matlab program to generate highly optimized VHDL for FPGA implementation", Circuits and Systems, 2001. ISCAS 2001. The 2001 IEEE International Symposium on, Volume: 4, pp. 514 - 517
[3]KEN YANG, 2006, "Flatten DAC frequency response EQUALIZING TECHNIQUES CAN COPE WITH THE NONFLAT FREQUENCY RESPONSE OF A DAC", MAXIM INTEGRATED PRODUCTS, EDN pp.65-74
[4]Pawan K. Gaikwad, 2013, "Field Programmable Gate Array Implementation of Digital Filter of Highest-Possible Order and its Testing using Advanced Microcontroller", IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 1, Issue 4
[5]Reference Manual, 2011, "Digilent Nexys2 Board", Digilent, Inc., 215 E Main Suite D | Pullman, WA 99163, (509) 334 6306, Doc: 502-134