International Journal of Engineering Research
Volume No.5 Issue: Special 5, pp:992-1128

ISSN:2319-6890)(online),2347-5013(print)
20 May 2016

FIR Filter Implementation on FPGA

Sriraksha S,Vasudha R and Jayashree Rajashekar

UG Scholars, Department of ECE

Cambridge Institute of Technology

K R Puram, Bangalore-560036.

Mr.Veerappa S C

Asst. Prof., Department of ECE
CiTech, Bangalore**.**
srirakshamurthy@gmail.com, veereshc@gmail.com

Abstract—Direct digital frequency synthesis is increasingly welcomed in modern communication systems and precise electronic systems. DDS output spectrum contains the fundamental plus aliased signals (images) that occur at integer multiples of the system clock frequency ± the selected output frequency. The frequency band where images exist is much richer in spurious signals and therefore, more hostile in terms of SFDR. There is a requirement to filter out these frequencies to improve the SFDR. A low pass FIR filter is designed and realized in field programmable gate arrays (FPGA) for real-time filtering applications. The coefficients are computed through the Blackman-Harris window technique using MATLAB. The design is coded in VHDL to cope with the parallelism of digital hardware. Simulation, compilation and synthesis is done to verify the validity of the design outputs. The observed output is compared with the calculated output results from MATLAB implementation that confirms the effectiveness of the design.

Keywords—Blackman-Harris,SFDR,DDS,FPGA

I INTRODUCTION

In signal processing, the function of a filter is to remove unwanted parts of the signal, such as random noise, or to extract useful parts of the signal, such as the components lying within a certain frequency range. In signal processing, a digital filter is a system that performs mathematical operations on a sampled, discrete-time signal to reduce or enhance certain aspects of that signal. Digital filters are used for two general purposes: (a) separation of signals that have been combined, and (b) restoration of signals that have been distorted in some way. FIR filters are one of the primary types of filters used in Digital Signal Processing. FIR filters are said to be finite because they do not have any feedback. Therefore, if we send an impulse through the system (a single spike) then the output will invariably become zero as soon as the impulse runs through the filter. The basic characteristics of Finite Impulse Response (FIR) filters are linear phase characteristic, high filter order (more complex circuits)and stability.

ADVANTAGES OF FIR OVER IIR

a)They can easily be designed to be "linear phase" (and usually are). Put simply, linear-phase filters delay the input signal but don’t distort its phase.
b) They are simple to implement. On most DSP microprocessors, the FIR calculation can be done by looping a single instruction.
c) They are suitable for multi-rate applications. By multi-rate, we mean either "decimation" (reducing the sampling rate), "interpolation" (increasing the sampling rate), or both. Whether decimating or interpolating, the use of FIR filters allows some of the calculations to be omitted, thus providing an important computational efficiency. In contrast, if IIR filters are used, each output must be individually calculated, even if it that output will discarded (so the feedback will be incorporated into the filter).

II FUNCTIONAL BLOCK DIAGRAM

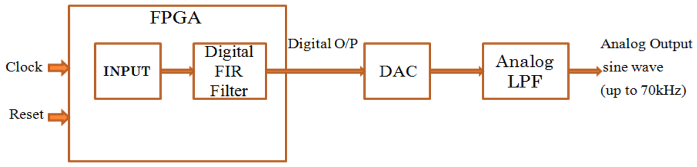


Figure.1 Functional Block Diagram

A clock of frequency 20 MHz is given as the input to the FPGA. The interface FPGA provides reset signals to the user programmable FPGAs in the Xtreme DSP Development Kit-IV. The Virtex-4 FPGA resets are controlled from the Interface FPGA, via FUSE software. Both of these resets are active low.The Xtreme DSP Development Kit-IV features these three

doi : 10.17950/ijer/v5i5/018

ISSN:2319-6890)(online),2347-5013(print)
20 May 2016

Xilinx FPGAs: Virtex-4 User FPGA, Virtex-II FPGA for clock management and a Spartan-II Interface FPGA. The Virtex-4 device is intended to be used for the main part of a user's design. The FIR Filter coded in VHDL is tested on Virtex-4 FPGA. The input samples are generated using MATLAB.

Two sine waves of frequencies 40 kHz and 400 kHz are combined to generate thousand samples. These frequencies have been chosen since one frequency is below the cut-off and the other is above the cut-off. The samples obtained are put in a look up table and used in VHDL code.

These samples are passed through the digital FIR filter to check its functionality. The digital FIR filter with a cut-off frequency of 70 kHz and sampling frequency of 2 MHz is designed. It effectively filters the frequencies above the cut-off frequency. In this case, sine wave of 400 kHz is eliminated and sine wave of 40 kHz is retained. A digital output is obtained which is fed to the DAC.A digital-to-analog converter (DAC) is a function that converts digital data (usually binary) into an analog signal.

A low-pass filter is a filter that passes signals with a frequency lower than a certain frequency and attenuates signals with frequencies higher than the cut-off frequency. The low pass filter used is an RC filter with R=2.2 kΩ and C= 1kpF.It consists of a resistor in series with a load, and a capacitor in parallel with the load.An analog sine wave of frequency up to 70 kHz and spurious free dynamic range of -35 dB is obtained as the output.

.

III IMPLEMENTATION IN MATLAB

Implementation: This involves producing the software code and/or hardware and performing the actual filtering.

The design of a digital filter involves following five steps:

(a) Filter specification: This may include stating the type of filter, for example low pass filter, the desired amplitude and/or phase responses and the tolerances, the sampling frequency, the word length of the input data.

(b) Filter coefficient calculation: The coefficient of a transfer function H(z) is determined in is this step, which will satisfy the given specification. The choice of coefficient calculation method will be influenced by several factors. The most important of which are the critical requirements i.e. specification. The window, optimal and frequency sampling method are the most commonly used.

(c) Realization: This involves converting the transfer function into a suitable filter network or structure.

(d) Analysis of finite word length effects: The effect of quantizing the filter coefficients and input data as well as the effect of carrying out the filtering

(e) Implementation: This involves producing the software code and/or hardware and performing the actual filtering.

The specifications of the FIR filter designed are as follows:

Type of filter: Low pass

Filter Structure: Direct form-I

Order: 20

Cut-off frequency: 70 kHz

Sampling Frequency: 2 MHz

Number of Taps: 21 (Quantized, Fixed-point)

Window: Blackman-Harris

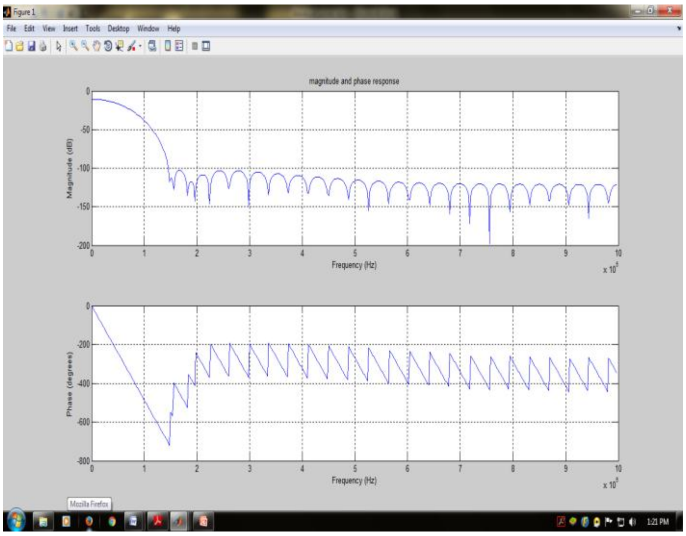


Figure.2 Magnitude and Phase response of low pass FIR filter in MATLAB

IV IMPLEMENTATION IN VHDL

The FIR filter designed consists of 20 adders, 21 multipliers and 20 D-flip flops. Each input sample (16 bit) is multiplied with a filter co-efficient (16 bit). The delayed version of the input and the second co-efficient are multiplied. These are added and the sum (32 bit) and carry obtained are passed onto the next stage. This process continues for 21 stages and occurs with every rising edge of clock.

doi : 10.17950/ijer/v5i5/018

ISSN:2319-6890)(online),2347-5013(print)
20 May 2016

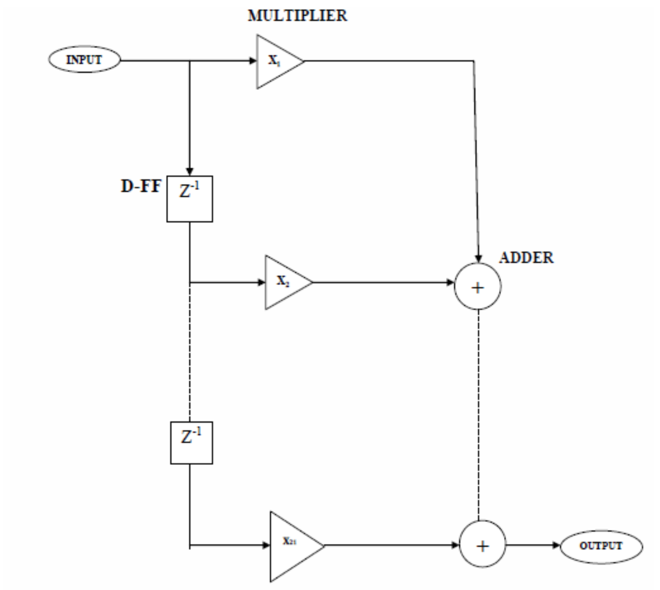


Figure.3 FIR filter realization

FLOW CHART OF FIR FILTER CODE

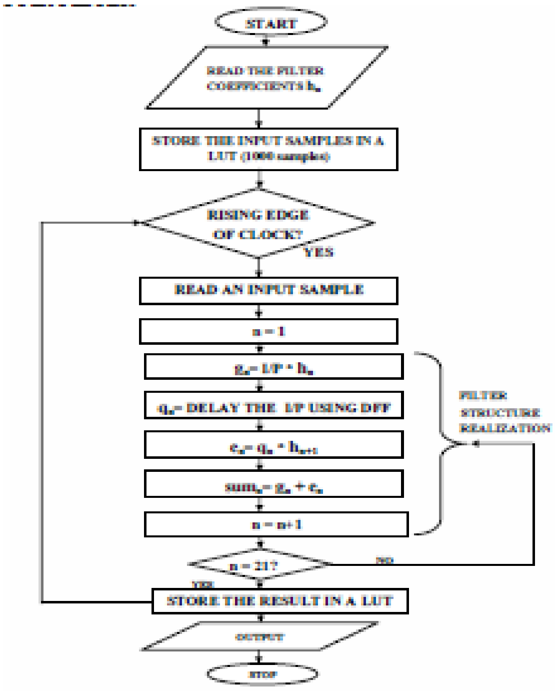

START
READ THE FILTER COEFFICIENTS h_n
STORE THE INPUT SAMPLES IN A LUT (1000 samples)
RISING EDGE OF CLOCK?
YES
READ AN INPUT SAMPLE
n = 1
g_n= I/P * h_n
q_n= DELAY THE I/P USING DFF
e_n= q_n * h_{n+1}
sum_n= g_n + e_n
n = n+1
FILTER STRUCTURE REALIZATION
n = 21?
NO
YES
STORE THE RESULT IN A LUT
OUTPUT
STOP

Figure.4 Flowchart of FIR filter code

Initially, the input samples are generated in MATLAB using “sin 40 kHz + sin 400 kHz”. The samples obtained are put in a look-up table and with every rising edge of clock each sample is processed for 21 stages. This process is continued until all the thousand samples are processed and the FIR filter is realized according to the structure. A sine wave of 40 kHz is obtained as the output in this case filtering the sine wave above cut-off frequency. It has a spurious free dynamic range of -42 dB.The output obtained is passed through an analog low pass filter to remove the distortions present in it.

FIR-DAC INTERFACE

The input samples are passed through FIR filter as well as DAC. Since FIR and DAC are interfaced, FIR filter effectively filters the samples with frequencies above cut-off and this output is passed through the DAC. The DAC output is a staircase waveform. In order to smoothen the waveform, the DAC output is passed through an analog low pass filter. It is then loaded onto the Xtreme DSP Development-IV kit. This kit consists of two independent DACs each having an isolated supply and ground planeThe DAC device has 14-bit resolution

The DAC device has 14-bit resolution. It has an internal Phase-Locked Loop (PLL) clock multiplier device feature. Initially, the user feeds 14-bits of data into the DAC. This data is latched into edge-triggered latches on the rising edge of the reference clock. It is interpolated by a factor of 2 by the digital filter. Finally it is sent to the 14-bit DAC.

IVRESULTS

1.SIMULATION RESULTS

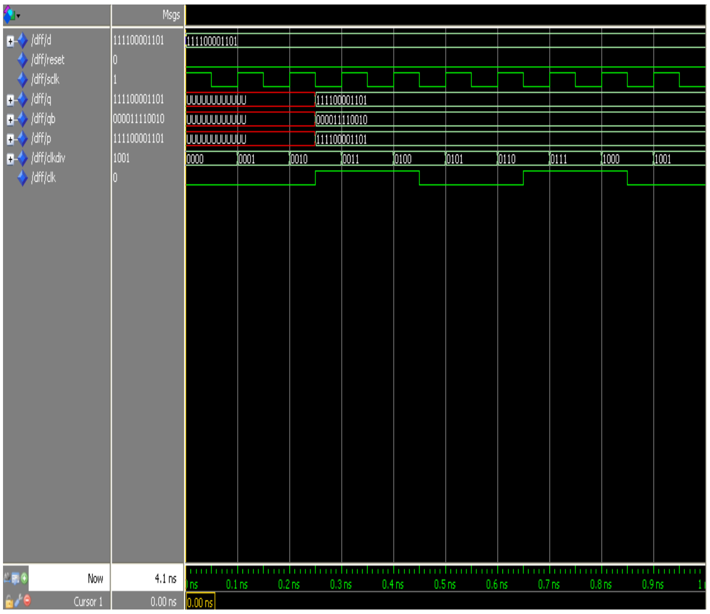
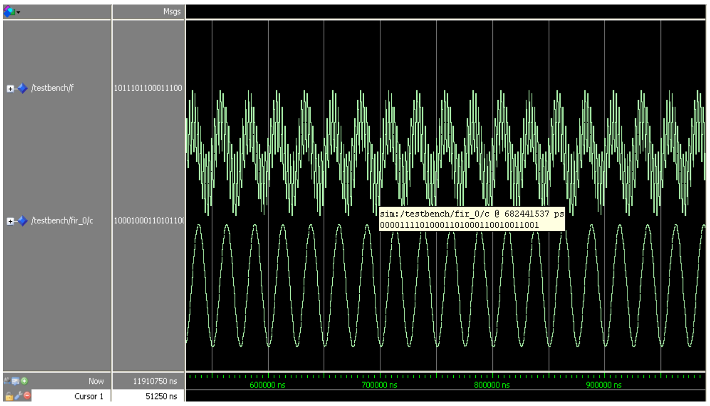
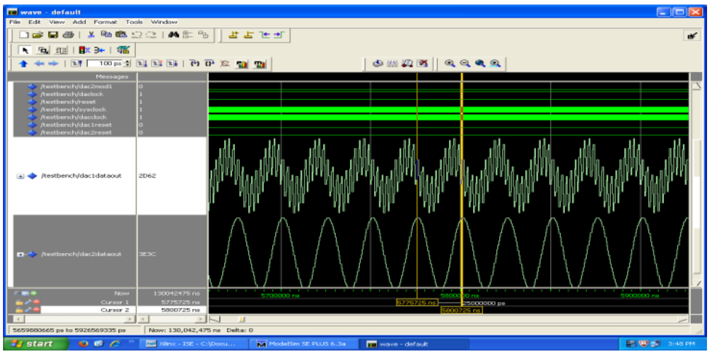


Figure.5 Output of D – FF



sim:/testbench/fir_0/c @ 682441537 ps
0000111101000110100011001001100l

Figure.6 Output of FIR filter designed.



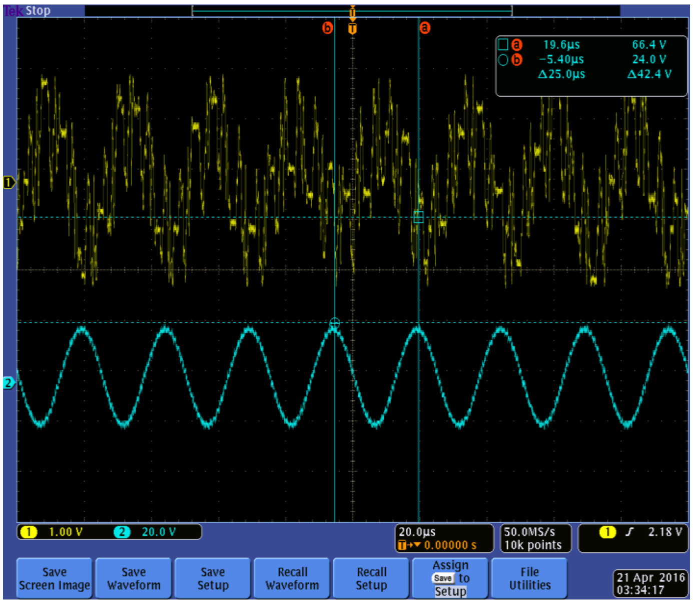
Signal	Value
/testbench/dac2mod1	0
/testbench/daclock	1
/testbench/reset	1
/testbench/sysclock	1
/testbench/dacclock	1
/testbench/dac1reset	0
/testbench/dac2reset	0
/testbench/dac1dataout	2D62
/testbench/dac2dataout	3E3C
Now	130042475 ns
Cursor 1	5775725 ns
Cursor 2	5800725 ns

5775725 ns 25000000 ps 5800725 ns

5659880665 ps to 5926569335 ps Now: 130,042,475 ns Delta: 0

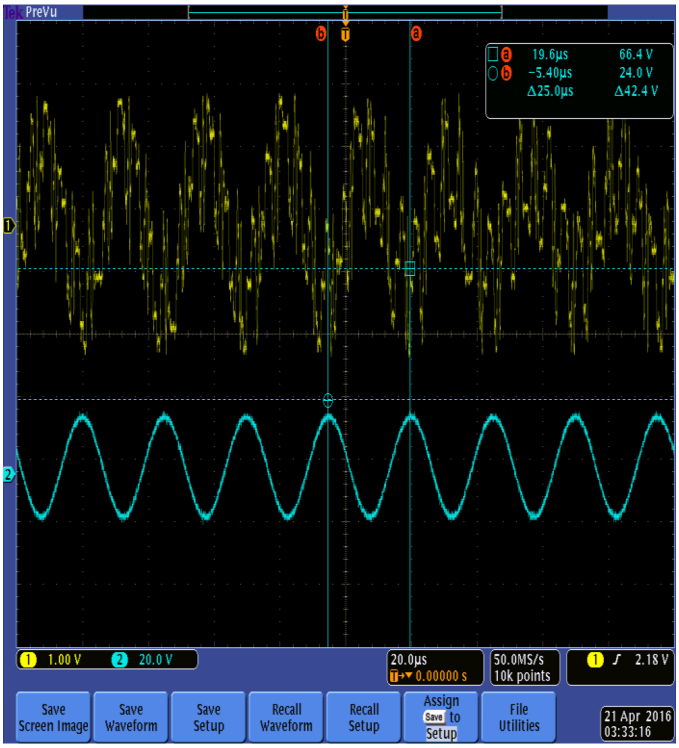
Figure.7 Behavioral Simulation of FIR-DAC interface

2.HARDWARE TESTING RESULTS



a	19.6µs	66.4 V
b	−5.40µs	24.0 V
	Δ25.0µs	Δ42.4 V

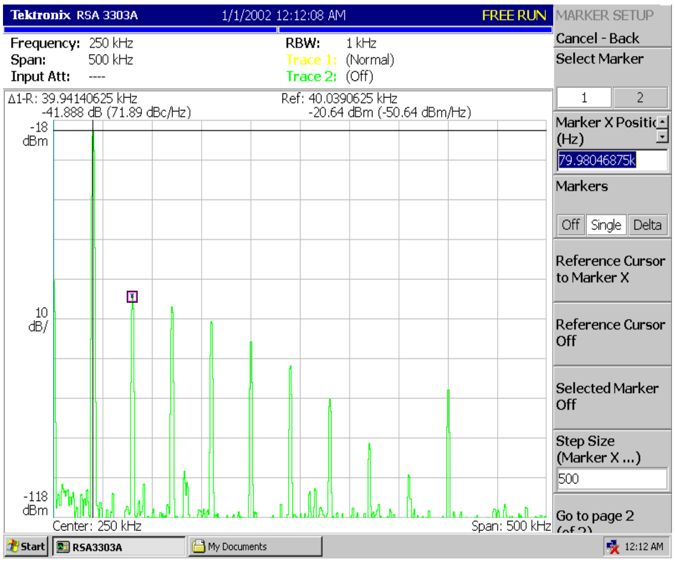
Figure.8 FIR filter output before passing through low pass filter



a	19.6µs	66.4 V
b	−5.40µs	24.0 V
	Δ25.0µs	Δ42.4 V

Figure.9 FIR filter output after passing through low pass filter

3.SPECTRUM ANALYZER OUTPUT



1/1/2002 12:12:08 AM

Frequency: 250 kHz
Span: 500 kHz
Input Att: ----
RBW: 1 kHz
Trace 1: (Normal)
Trace 2: (Off)

Δ1-R: 39.94140625 kHz
-41.888 dB (71.89 dBc/Hz)
Ref: 40.0390625 kHz
-20.64 dBm (-50.64 dBm/Hz)

Marker X Position (Hz): 79.98046875k
Step Size (Marker X ...): 500

Center: 250 kHz
Span: 500 kHz

V CONCLUSION

A low pass FIR filter with a cut-off frequency of 70 kHz has been designed in MATLAB using the windowing technique. The design was coded in VHDL and tested on FPGA. The samples were generated for a combination of two different sine waves with frequencies above and below 70 kHz. The filter designed effectively eliminates the frequencies above 70 kHz. The Spurious Free Dynamic Range was found to be -42 dB on the spectrum analyzer.

doi : 10.17950/ijer/v5i5/018

VI ACKNOWLEDGEMENT

We sincerely thank our guide Prof. Veerappa S C, Dept. of ECE for his unstilted support, valuable guidance and helpthroughout the project related work.We would like to thank all the faculty members and nonteaching staff ofDept. of Electronics and Communication, fortheir constant support.

REFERENCES

1) Digital Signal Processing, second edition by D. Ganesh Rao and Vineeta. P.Gejji2) Digital Signal Processing Using MATLAB, Third Edition by Vinay K. Ingle and John G.Proakis.
3)"Implementation of FIR Filter Using Efficient Window Function and its Application in Filtering a Speech Signal" by Saurabh Singh Rajput, Dr.S.S. Bhadauria
4) Xtreme DSP Development Kit-IV Reference Guide NT107-0272 - Issue 4
5) Design of FIR Filters by Elena Punskaya, www-sigproc.eng.cam.ac.uk/~op205