Hardware Implementation of DSP Filter on FPGAs

Apurva Singh Chauhan
HCST,Mathura
Mathura,U.P.India

A.Mukund Lal
HCST,Mathura
Mathura,U.P. India

Varun Maheshwari
Dayalbagh Educational Institute
Agra,U.P. India

D.Bhagwan Das, PhD.
Dayalbagh Educational Institute
Agra,U.P. India

ABSTRACT

This paper describes the hardware implemention of Digital signal processing filters on FPGAs (Field programmable gate array). Traditionally, digital signal processing (DSP) algorithms are implemented using general purpose programmable DSP chips for low-rate applications. Alternatively, special purpose fixed function DSP chipsets and application specific integrated circuits (ASICs) are used for high-performance application. Technological amelioration by Xilinx on FPGAs in the past few years has opened new paths for engineers to design various applications on FPGAs. The FPGAs assert the high explicitness of the ASIC while avoiding its high development cost and its inability to accommodate design modifications after production. Highly adaptable and design-flexibility, FPGAs provide optimal device utilization through conservation of board space and system power important advantages not available with many stand-alone DSP chips.

Keywords*:* Digital Filter, Digital Signal Processing, FPGA

1. INTRODUCTION

Digital signal processing is an important area where FPGAs have found many applications in recent years. A major revolution in filter design has taken place over the past few years. FPGA contains over a million equivalent logic blocks (logic gates and tens of thousands of flip-flops).This means that it is not possible to use traditional methods of logic design involving the drawing of logic diagrams when the digital circuit may contain thousands of gates[1]. The reality is that today digital systems are designed by writing software in the form of hardware description languages (HDLs) [2, 3]. When using the hardware description languages the designer typically describes the behavior of the logic circuit rather than writing traditional Boolean logic equations. Computer-aided design tools are used to both simulate VHDL design and to synthesize the design to actual hardware [4-8].

This paper explains the process of designing a DSP filter in VHDL using MATLAB(R2009a). Coefficients of filters are generated in MATLAB. Finally the design is implemented on Xilinx Spartan-3 FPGA Kit.

2. DIGITAL FILTERS

Digital Filter is numerical procedure or algorithm that transforms a given sequence of numbers into a second sequence that has some more desirable properties. Figure 1 shows the block diagram of digital filter.

Fig 1: Block diagram of Digital Filter

A digital filter is better conceptualized in the frequency domain. The filter implementation simply performs a convolution of the time domain impulse response and the sampled signal. A filter is designed with a frequency domain impulse response which is as close to the desired ideal response as can be generated given the constraints of the implementation. The frequency domain impulse response is then transformed into a time domain impulse response that is the coefficients of the filter [9, 10].

Digital filters are more size and powers efficient than analog filters in various applications as linear phase, very high stop band attenuation, very low pass band ripple. Filter's response must be programmable or adaptive; the filter must manipulate phase and, very low shape factors.

There are two basic types of digital filters, Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) filters. The general form digital filter difference equation is shown in equation (1):

$$y(n) = \sum_{i=0}^{N} a_i \, x(n-i) - \sum_{i=1}^{N} b_i \, y(n-i) \quad (1)$$

Where y(n) is the current filter output, the y(n-i) are previous filter outputs, the x(n-i) are current or previous filter inputs, the a_i are the filter's feed forward coefficients. Corresponding to the zeros of the filter, the b_i are the filter's feedback coefficients corresponding to the poles of the filter and N is the filter's order. IIR filters have one or more nonzero feedback coefficients. As the result of the feedback term, if the filter has one or more poles, once the filter has been excited with an impulse there is always an output. FIR filters have no non-zero feedback coefficient. That is, the filter has only zeros, and once it has been excited with an impulse, the output is present only for a finite (N) number of computational cycles [11].

3. IMPLEMENTATION OF DSP FILTERS ON FPGA

The advantage of FPGAs over conventional DSPs to perform digital signal processing is their capability to exploit parallelism, i.e., replication of hardware functions that operate concurrently in different parts of the chip. Another advantage

of FPGAs is the flexibility for trading off between area and speed.

FPGA offers the significant benefits of being readily programmable. FPGA can be reprogrammed, giving designers multiple opportunities to tweak there circuits. FPGA consists of an array of logic blocks that are configured using software and programmable input/output blocks surround the logic blocks. Both are connected by programmable interconnects, Offers millions of gates of logic capacity operate at 300MHz. FPGA offers all of the features needed to implement most complex designs [12]. Dedicated memory blocks can be configured as basic single-port RAMs, ROMs, FIFOs. FPGAs are system building resource for systems such as high-speed serial Input/output, arithmetic modules, embedded processors, and large amount of memory. Figure 2 shows the design flow of the FPGA [13].

This is the entire process for designing a device. As specification allows each engineer to understand the entire design and his or her piece of it. It allows the engineer to design the correct interface to the rest of the pieces of the chip. It also saves time and misunderstanding.. For smaller chips, schematic entry is often the method of choice, especially if the design engineer is already familiar with the tools. For larger designs, however, a hardware description language (HDL) such as Verilog or VHDL is used because of its portability, flexibility, and readability. Synthesis tool has recommended or mandatory methods of designing hardware so that it can correctly perform the desired operation. There will be much iteration of design and simulation in order to get the correct functionality and timing analysis.

.

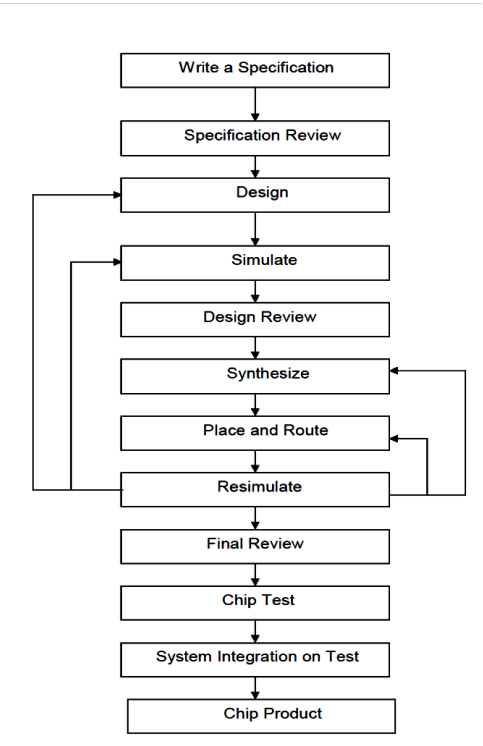

Fig 2: Steps to implement the coding on FPGA

A basic step involves designing Digital filter:

1. Determine a desired response or a set of desired response.
2. Select a class of filter for approximating the desired response. In our design we implemented FIR filters.
3. .Establish a criterion for the response of a filter in the selected class compared to the desired response.
4. Synthesize the filter using a proper structure and a proper implementation form.
5. Analyse the filter performance.

To completely describe digital filters, three basic elements (or building blocks) are needed; an adder, a multiplier, and a delay element. FIR filter order is N-I, length is N. System output depends on a function of the input, and has no direct relationship with the past output, it does not contain feedback branch. y(n) is the output. And x (n-i) is the input sample sequence on the nth times. h(i) is the i-th level filter tap coefficient, L is the number of the filter tap. Its basic structure is shown in Figure 3. It can find that the structure of FIR filter in hardware is mainly composed of the shift register, adder and multiplier.

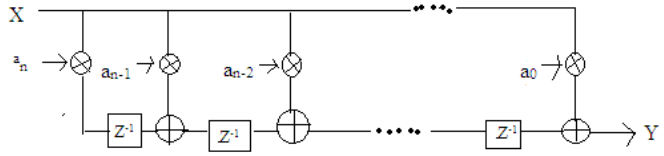

Fig 3: Structure of digital filter

The multiplier is a gain element, and it multiplies the input signal by a constant. The constant which is being added is the coefficient terms of the fir filter. Below Figure 4 shows the design flow for multiplier.

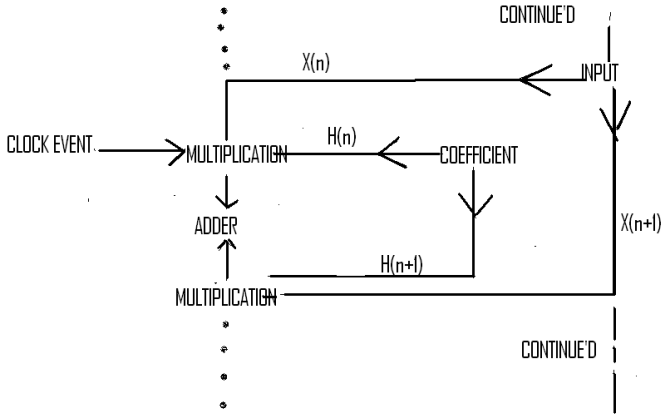

Fig 4: Design flow for adder

The adder has two inputs and one output, and it simply adds the two inputs together. This added adds the previous value of the multiplication with the present value of the multiplication. Figure 5 shows the design flow for adder

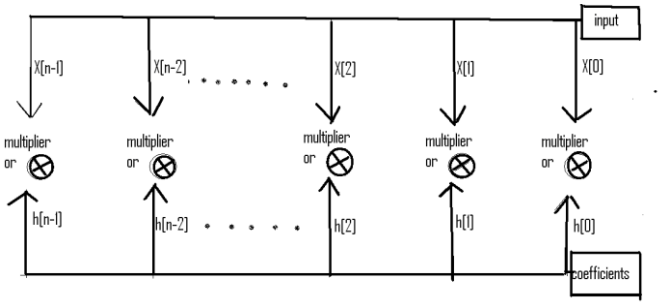

Fig 5: Design flow for multiplier

The whole VHDL codes completes in the three processes. Firstly initialization the input values of signal. Second is the multiplication of the input value with coefficients of the filter and finally the addition of the multiplied terms generated the output. Coefficient of the digital FIR filter is generated in MATLAB. Figure 6 shows the flow chart of programming.

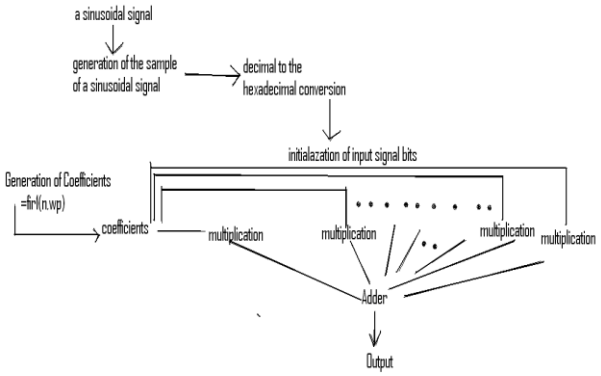

Fig 6: Flow Chart of Programming

The VHDL code is implemented on Spartan-3 Starter FPGA Kit board and Snap shot of Spartan-3 FPGA Kit is shown in Figure 7 includes the following components and features[14]

- 200,000-gate Xilinx Spartan-3 XC3S400 FPGA in a 256-ball thin Ball Grid Array
- package (XC3S400FT256)
- 4,320 logic cell equivalents
- 9-pin RS-232 Serial Port
- DB9 9-pin female connector (DCE connector)
- RS-232 transceiver/level translator

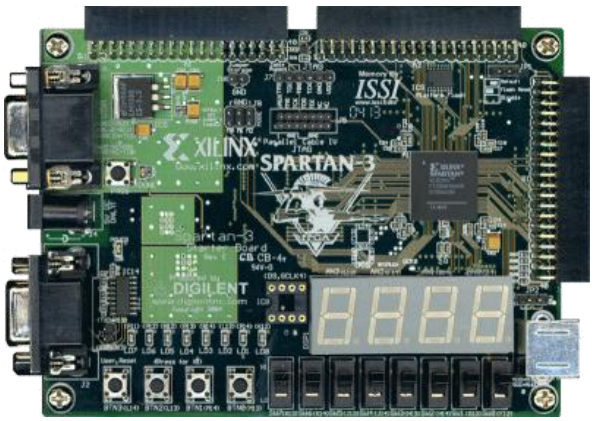

Fig 7: Snap shot of Spartan-3 FPGA Kit

4. EXPERIMENTAL ANALYSIS

For the sake of implementation the following specifications are considered for low pass FIR model of filter:

- Cut off frequency =50 Hz
- Sampling frequency =1.6 kHz
- Order of the filter =32

There are basically two inputs should be here in the digital form for the digital FIR filter designing.

- I/P signal which can be modify in accordance.
- Coefficient of the filter stages

The I/P signal is sine wave of fundamental frequency i.e. 50 Hz. We had taken eight samples of sine wave.

Table 1 Inputs for 50 Hz wave

Inputs	decimal	hexadecimal
X1 (n)	0.0037	0D
X2 (n)	0.0317	14
X3 (n)	0.1255	1B
X4 (n)	0.2498	22
X5 (n)	0.3084	27
X6 (n)	0.2498	2C
X7 (n)	0.1255	30
X8 (n)	0.0317	32

The table 2 shows the digitized form of coefficients

Table 2 Coefficients for 50 Hz wave

Coefficients	decimal	hexadecimal
H1 (n)	0.0037	06
H2 (n)	0.0317	20
H3 (n)	0.1255	6F
H4 (n)	0.2498	D2
H5 (n)	0.3084	FF
H6 (n)	0.2498	D2
H7 (n)	0.1255	6F
H8 (n)	0.0317	20

The given Fig 8 shows the Magnitude and phase response of filter

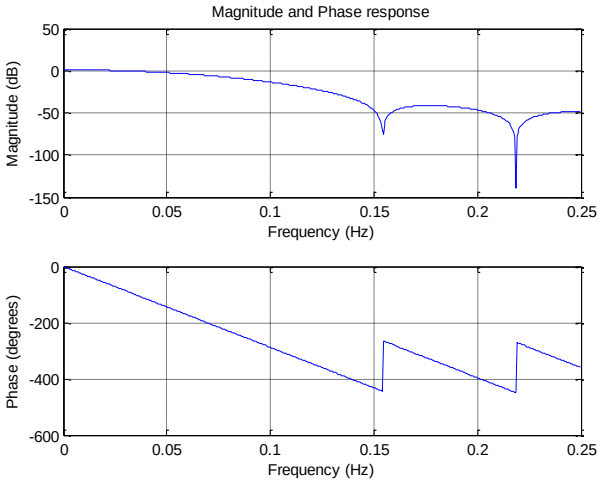

Fig 8: Magnitude and phase response of filter

Figure 10 and Figure 11 shows the input and output waveforms of the filter

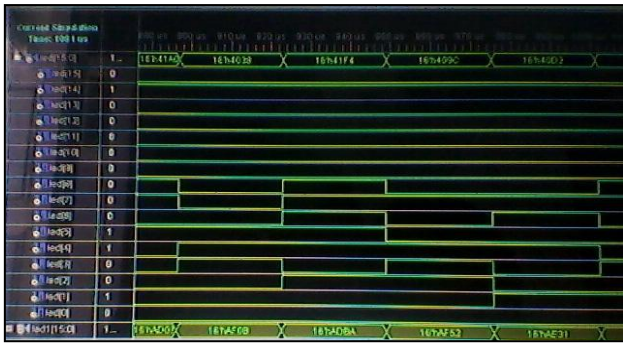

Fig 9: Input waveform of Low Pass Filter

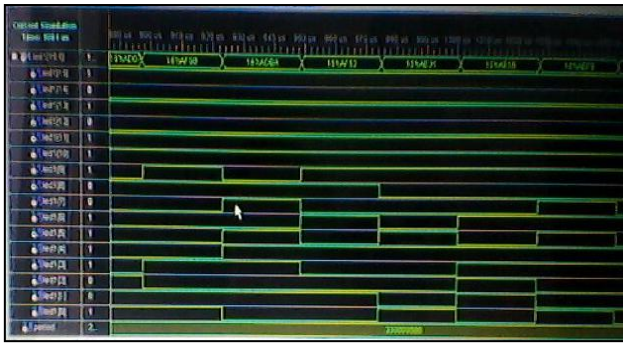

Fig 10: Output waveform of Low Pass Filter

5. RESULT AND CONCLUSIONS

The filter specifications are real world used to design the filter coefficients. These coefficients are used to implement filter on Xilinx FPGA Spartan 3 kit using Xilinx ISE 9.2i.

As the input is provided with the 50 Hz frequency samples to the filter model calculated it through the MATLAB software. Fig 10 shows the I/P and O/P response of the Low Pass Filter in ISE 9.2I SIMULATOR.

In this paper, an eight order low-pass and high pass and eight order band pass FIR filter is implemented in Spartan-III-xc3s400c-4ft256 FPGA. The transposed form structure of these filters is implemented. This approach gives a better performance than the common filter structures in terms of speed of operation, cost, and power consumption. In the transposed -form structure, N Shift Registers, N Adder and N+1 multipliers are used to realize the N order low pass , high pass and band pass filter. The designed filters can work for real time processing of any digital signal.

Table 3 shows the device utilization for the implementation of Filter.

Table 3 Device Utilization Summary for Low Pass Filter

Number of Slice	379 out of 3158 10%
Number of Slice Flip Flop	542 out of 7168 7%
Number of input LUT	553 out of 7168 7%
Number used as logic	517
Number used as Shift registers	36
Number of IOs	35
Number of bonded IOBs	33 out of 173 19%
IOB Flip Flips	16
Number of GCLKs	2 out 8 25%

6. REFERENCES

[1] Ray Goslin, Dr. San Jose, CA 95124 A Guide to Using Field Programmable Gate Arrays (FPGAs) for Application-Specific Digital Signal Processing Performance Gregory Digital Signal Processing Program Manager Xilinx, Inc. 2100 Logic .

[2] Mark S. Manalo and Ashkan Ashrafi "Implementing Filters on FPGAs" Department of Electrical and Computer Engineering Real-Time DSP and FPGA Development Lab

[3] Prabhat Ranjan(July 2008) "Implementation of FIR Filter on FPGA"Department of Electronics and Communication Thapar University,Punjab

[4] Joseph B. Evans "An Efficient FIR Filter Architecture" Telecommunications & Information Sciences Laboratory Department of Electrical & Computer Engineering University of Kansas Lawrence, KS 66045-2228

[5] D. E. Borth, I. A. Gerson, J. R. Haug, and C. D. Thompson (Apr 1988). A flexible adaptive FIR filter VLSI IC. IEEE Journ. Select. Areas Commun., SAC-6(3):494–503,.

[6] P. R. Cappello, editor 1984.VLSI Signal Processing. IEEE Press,

[7] J. B. Evans,Y. C. Lim, and B. Liu.(Apr 1990) A high speed programmable digital FIR filter. In IEEE Int. Conf. Acoust., Speech, Signal Processing,

[8] J. Gallia et al (Feb 1990).High-performance BiCMOS 100k-gate array. IEEE J. Solid State Circuits, SC-25(1):142–149.

[9] R. Hartley, P. Corbett, P. Jacob, and S. Karr. , May 1989 A high speed FIR filter designed by compiler. In IEEE Cust. IC Conf., pages 20.2.1–20.2.4.

[10] R. Rabiner1 and Ronald W. Schafer2 Introduction to Digital Speech Processing Lawrence 1 Rutgers University and University of California, Santa Barbara, USA, rabiner@ece.ucsb.edu 2 Hewlett-Packard Laboratories, Palo Alto, CA, USA

[11] NITSCHKE and GREGORY A. MILLER Digital filtering in EEG/ERP analysis: Some technical and empirical comparisons ACK B. university of Illinois at Urbana-Champaign, Urbana, Illinois and EDWIN W. COOK III University of Alabama, Birmingham, Alabama

[12] R.Boite and H. Leich,comments in (1984.) on 'A fast procedure to design equirriple minimum phase Fir filters.'IEEE Trans. Circuits Syst.CAS-31,503-504

[13] FPGA Architect - XilinxXC4000/Spartan. By ELANIX Inc.

[14] Spartan-3 Starter Kit Board User Guide UG130 (v1.1) May 13, 2005