

FPGA Optimized Low Power And High Speed FIR Filter Structures For DSP Applications

1. N. Parijatha, 2. K. R. Anusha Hinduja, 3. A. Chandra Shaker

1. Department of ECE, SRTIST, Ramananda Nagar, Nalgonda(dt),A.P,India.

2. Department of ECE, SRTIST, Ramananda Nagar, Nalgonda(dt),A.P,India.

3. Department of ECE, SRTIST, Ramananda Nagar, Nalgonda(dt),A.P,India.

Abstract

Digital filters are typically used to modify or alter the attributes of a signal in the time or frequency domain. The most common digital filters is the linear time-invariant (LTI) filter. LTI digital filters are generally classified as being finite impulse response (FIR) or infinite impulse response (IIR). In this project, various FIR filter structures will be studied and implemented in VHDL. The tapped line delay structure and transposed structure of FIR filter realization will be studied. The reduced hardware structures by using the symmetry property will also be studied and implemented. Basic arithmetic blocks to carry out DSP on FPGAs will be discussed. The very popular LUT based approach for arithmetic circuit implementation will be presented. The conventional PDSF MAC and Distributed arithmetic MAC units will be implemented and their performance will be compared. Usage of Pipelining in multipliers for improving the speed will also be discussed. The DA (Distributed arithmetic) based FIR filter will be studied and implemented. By using arithmetic building blocks, various structures of FIR filters will be implemented in VHDL. The ModelSim XE simulator will be used to simulate the design at various stages. Xilinx synthesis tool (XST) will be used to synthesize the design for spartan3E family FPGA (XC3S500E). Xilinx Placement & Routing tools will be used for backend, design optimization and I/O routing.

1. 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. The following block diagram illustrates the basic idea.



Fig 1.1 Filter basic Functionality

There are two main kinds of filter, analog and digital. They are quite different in their physical makeup and in how they work. An analog filter uses analog electronic circuits made up from components such as resistors, capacitors and op-amps to produce the required filtering effect. Such filter circuits are widely used in such applications as noise reduction, video signal enhancement, graphic equalizers in hi-fi systems, and many other areas. There are well-established standard techniques for designing an analog filter circuit for a given requirement at all stages, the signal being filtered is an electrical voltage or current which is the direct analogue of the physical quantity (e.g. a sound or video signal or transducer output) involved. A digital filter uses a digital processor to perform numerical calculations on sampled values of the signal. The processor may be a general-purpose computer such as a PC, or a specialized DSP (Digital Signal Processor) chip.

Finite Impulse Response (FIR) filters

An FIR with constant coefficients is an LTI digital filter. The output of an FIR of order or length L , to an input time-series $x[n]$, is given by a finite version of the convolution sum given by the following equation

$$y[n] = x[n] * f[n] = \sum_{k=0}^{L-1} x[k]f[n-k]$$

where $f[0] \neq 0$ through $f[L-1] \neq 0$ are the filter's L coefficients. They also correspond to the FIR's impulse response. For LTI systems it is sometimes more convenient to express in the z-domain with

$$Y[z] = F(z)X(z)$$

Where $F(z)$ is the FIR's transfer function defined in the z-domain by The roots of polynomial $F(z)$ in above equation define the zeros of the filter. The presence of only zeros is the reason that FIRs are sometimes called all zero filters.

The L^{th} order LTI FIR filter is graphically interpreted in Figure. It can be seen to consist of a collection of a "tapped delay line," adders, and multipliers. One of the operands presented to each multiplier is an FIR coefficient, often referred to as a "tap weight" for obvious reasons. Historically, the FIR filter is also known by the name "transversal filter," suggesting its "tapped delay line" structure.

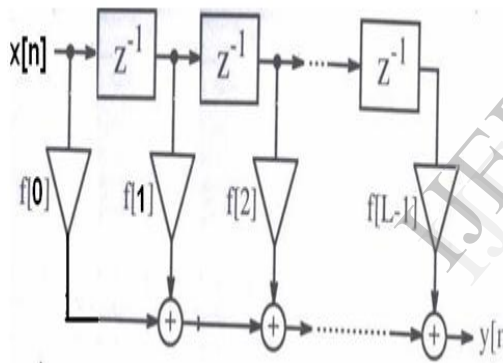


Fig 1.2 L tap FIR filter

The above represented structure for FIR filter is called Direct Form FIR structures. The other structures such as transposed, symmetric are present.

2. Implementation of FIR filter structures on FPGA

Direct form FIR filter

The direct form of FIR filter is standard linear convolution, which described the output as convolution of input and impulse response of the filter.

$$y[n] = x[n]*c[n] = \sum_k x[k]c[n-k] = \sum_k c[k]x[n-k].$$

where $c[n]$ values represent filter coefficients, and $x[n]$ represents the input samples. The figure shows the direct form FIR structure

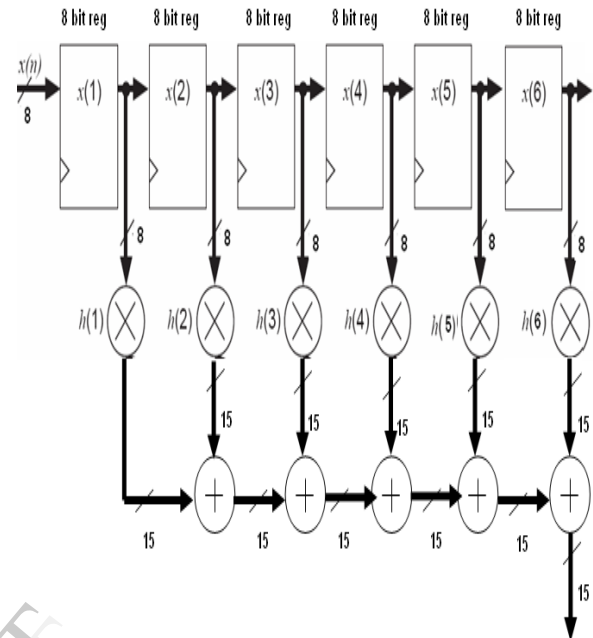


Fig 2.1 Direct form 6-tap FIR filter .

Transposed Form FIR Structure

A variation of the direct FIR model is called the transposed FIR filter. It can be constructed from the FIR filter in Fig by:

- Exchanging the input and output
- Inverting the direction of signal flow
- Substituting an adder by a fork, and vice versa

A transposed FIR filter is shown in Fig and is, in general, the preferred implementation of an FIR filter. The benefit of this filter is that there is no need for an extra pipeline stage for the adder (tree) of the products to achieve high throughput.

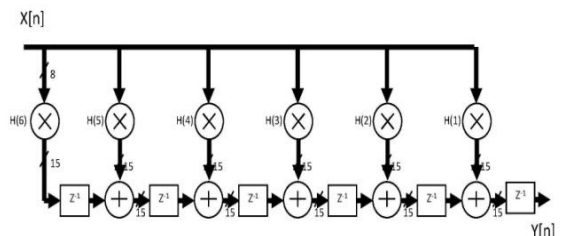


Fig 2.2 Transposed form structure for FIR filter.

Since functionality wise the transposed structure is same as direct form structure the

impulse response is same for both. Hence the transposed form filter behaviour is same as direct form for any input.

Symmetric Fir Filter Structure

The chosen FIR impulse response is symmetric, i.e., $h(1)=h(6)$, $h(2)=h(5)$, and $h(3)=h(4)$. The output equation can be modified as below

$$Y[n] = (x[n] + x[n-5]) * h(1) + (x[n-1] + x[n-2]) * h(2) + (x[n-3] + x[n-4]) * h(3)$$

This approach leads to more efficient structure, with only three multiplications shown in figure

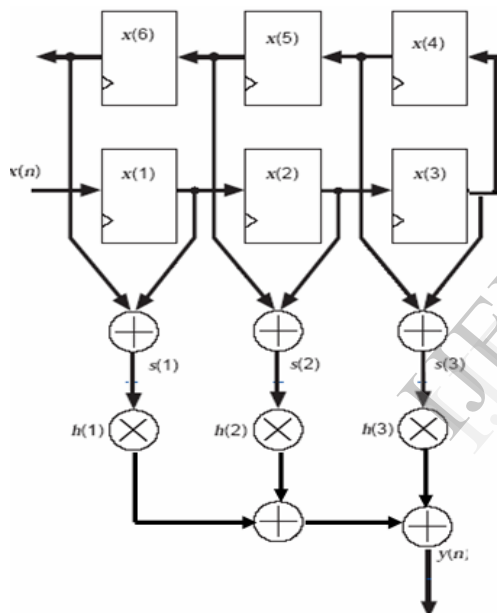


Fig 2.3 symmetric structure FIR filter

Since functionality wise the symmetric structure is same as direct form structure the impulse response is same for both. Hence the transposed form filter behaviour is same as direct form for any input.

Fir Filter Using Distributed Arithmetic

A completely different FIR architecture is based on the distributed arithmetic (DA) concept introduced. In contrast to conventional sum-of-products architecture, in distributed arithmetic we always compute the sum of products of a specific bit b over all coefficients in one step. This is computed using a small table and an accumulator with shifter. The structure of Distributed Arithmetic FIR filter is shown in the below figure.

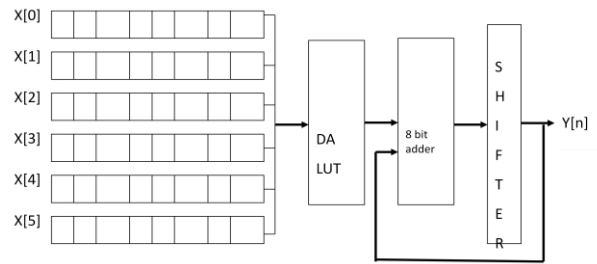


Fig 2.4 DA FIR filter structure

The shift and add unit which is main logic in Distributed arithmetic is implemented in structural style using n bit adder and PIPO register as components. The shift and add circuit shift the existing number one bit to right and adds the next number to it. The top level FIR filter module is implemented in structural style using Look up Table and shift-and-add circuit as components.

3. RESULTS

Simulation of Direct form FIR Structure

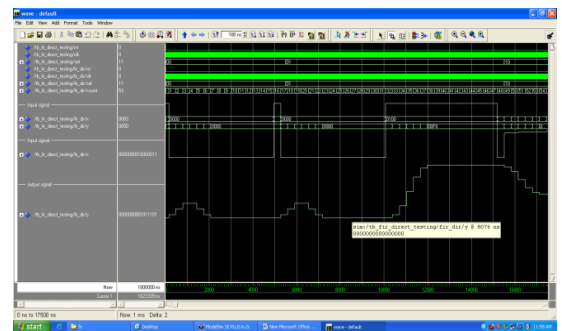


Fig 3.1 Unit impulse and unit step response of Direct form FIR Filter

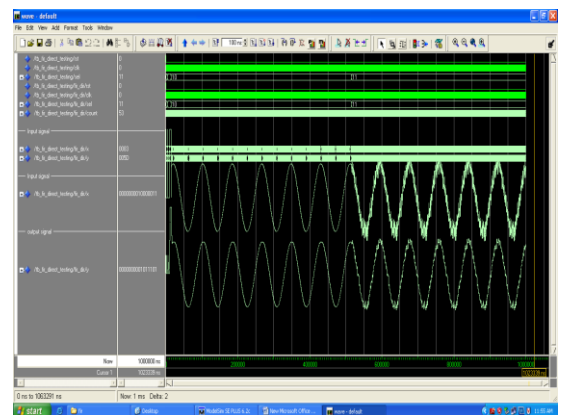


Fig 3.2 Sine wave and sine wave with noise response of Direct form FIR Filter.

4. CONCLUSION

In the above sections a 6 tap FIR filter is implemented in 4 different structures which are Direct form, Transposed form, Symmetric form, Distributed arithmetic based FIR filters. The functionality of these structures was verified and found same for all. The difference in implementation results only performance wise difference but filtering action will be same.

By using arithmetic building blocks, various structures of FIR filters are implemented in VHDL. The ModelSimSE 6.2C simulator has been used to simulate the design at various stages.

Xilinx synthesis tool (Xilinx ISE 9.2i version) has been used to synthesize the design for Spartan 3E family FPGA (XC3S500E). On chip verification is done by Xilinx chip scope pro 9.2i tool.

In the implementation it is assumed that the coefficients are constant and fixed. Especially in Distributed Arithmetic type the complete Look up Table contents are calculated based on the coefficients. This assumption is true and applicable for several practical applications. The present work implements few FIR filter structures on spartan3E FPGAs but analyzing the trade-off between performance and chip area.

As the present implemented structures makes use of only VHDL constructs, hence can be ported on any FPGA family. The work also brings out fundamental design goals in FPGA based design. The Distributed Arithmetic usage and its advantages over conventional arithmetic are clearly discussed.

5. FUTURE SCOPE

- (1) The implemented FIR structures at code level can be modified to make full benefit from the FPGA, such as using fast carry chains, Embedded Array Blocks etc.
- (2) To achieve the peak performance fully parallel pipelined version can be implemented
- (3) In the present work the DA based FIR filter is implemented with one LUT and without pipelining. This can be extended to full parallel implementation with more than one LUTs for high speed applications. The following figure shows a possible architecture.

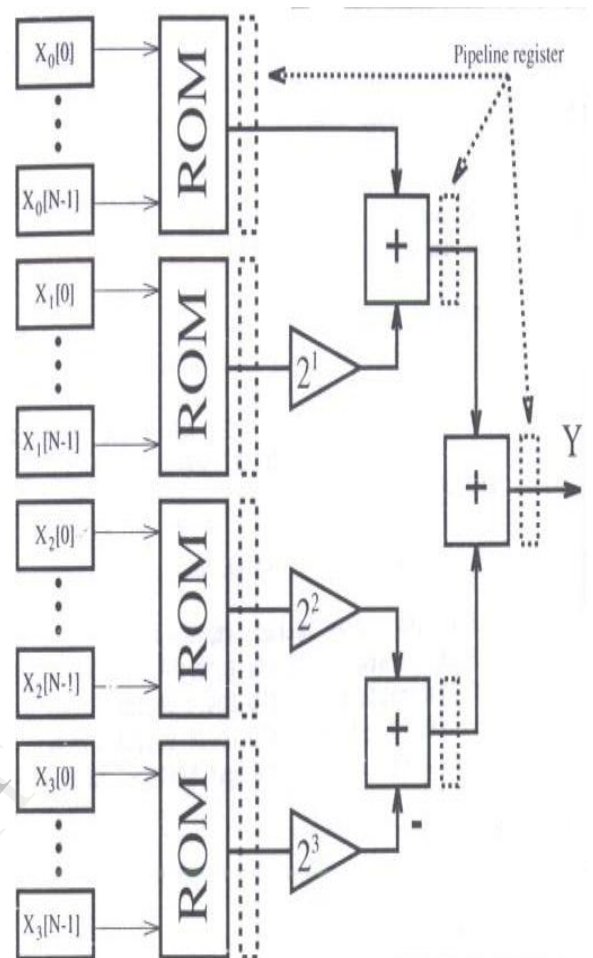


Fig 5.1 Parallel implementation of Distributed Arithmetic FIR filter.

(4) The work can be extended to design Adaptive Digital filters. Following section presents, a highly popular biomedical application with adaptive filtering.

Adaptive Digital Filters

There are instances when the condition that a filter input characteristics changes. It is advantageous for the filter to adapt to the changes in order to extract the desired signal. This brings about the term "adaptive" digital filter. In the area of adaptive digital filtering, most of the filters are realized using FIR filters because of their stability and simplicity. The adaptive noise cancellation circuit shown in Figure can be used in areas like bio-medical signal processing. A case in point is the merging of the heartbeat of the pregnant mother with that of the fainter heartbeat of the fetus. The adaptive noise cancellation technique can be used to filter out the mother's stronger heartbeat to make the fetal heartbeat more readily

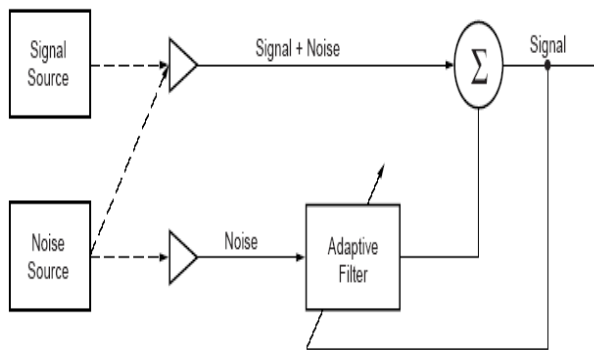


Fig 5.2 Adaptive Noise Canceling Filter

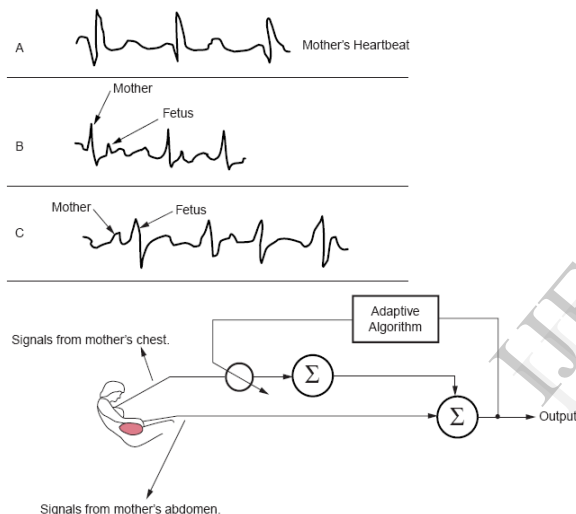


Fig 5.3 Adaptive Cancellation of Maternal ECG from Fetal ECG3

As above figure clearly shows, the collected signals from the chest and abdominal electrodes of the mother are adaptively filtered so that the mother's ECG signature is made less pronounced and the fetus' ECG signature is easier to recognize and analyze

The implemented FIR structures are based on fixed point sign magnitude number representation, these arithmetic block design can be extended to include the 2s complement number system. The floating point based FIR filter can also be implemented.

6. References

1. Digital Signal Processing with Field Programmable Gate Arrays, by U. Meyer-Baese, Springer Publications.
2. Practical FIR Filter Design in MATLAB, Revision 1.1, Ricardo A. Losada The Math Works, Inc. 3 Apple Hill Dr. Natick, MA 01760, USA
3. A Guide to Using Field Programmable Gate Arrays (FPGAs) for Application-Specific Digital Signal Processing Performance by Gregory Ray Goslin Digital Signal Processing Program Manager Xilinx, Inc. 2100 Logic Dr. San Jose, CA 95124
4. Optimization Techniques for Efficient Implementation of DSP in FPGAs. Mentor graphics Paper
5. Implementing FIR Filters in FLEX Devices, form altera application note73
6. ALTERA FLEX 10KE data sheet form www.altera.com