International Journal of Computer and Communication Technology

Volume 6 | Issue 2 Article 7

April 2015

DESIGN OF FIR FILTER USING LOOK UP TABLE AND MEMORY BASED APPROACH

A. SIRISHA
Department of Electronics and Communication Engineering, Chirala Engineering College, Chirala.,
asirisha@gmail.com

P. BALANAGU
Department of Electronics and Communication Engineering, Chirala Engineering College, Chirala.,
pbalanagu@gmail.com

N. SURESH BABU
Department of Electronics and Communication Engineering, Chirala Engineering College, Chirala.,
nsureshbabu@gmail.com

DESIGN OF FIR FILTER USING LOOK UP TABLE AND MEMORY BASED APPROACH

A.SIRISHA, P.BALANAGU & N.SURESH BABU

Department of Electronics and Communication Engineering,
Chirala Engineering College, Chirala.

Abstract: The main objective of this Paper is to develop Finite Impulse Response (FIR) Filter using look-up table (LUT) and memory based processor. The configuration is to reduce memory size, logic gate count and improve the speed of operation. Digital filters are becoming ubiquitous in audio applications. As a result, good digital filter performance is important to audio system design. Digital filters differ from conventional analog filters by their use of finite precision to represent signals and coefficients and finite precision arithmetic to compute the filter response. In this Paper, FIR filter is implemented in Xilinx ise using VHDL language. VHDL coding for the FIR filter is implemented in this Paper and waveforms are observed through simulation.

Keywords: *Memory based, Look up table based, FIR filter.*

1. INTRODUCTION

Filter is a frequency selective network. It passes a band of frequencies while attenuating the others. Filters are classified as analog and digital depending on nature of inputs and outputs. Filters are further classified as finite impulse response and infinite impulse response filters depending on impulse response.

Digital filters are used extensively in all areas of electronic industry. This is because digital filters have the potential to attain much better signal to noise ratios than analog filters and at each intermediate stage the analog filter adds more noise to the signal, the digital filter performs noiseless mathematical operations at each intermediate step in the transform. The digital filters have emerged as a strong option for removing noise, shaping spectrum, and minimizing inter-symbol interference in communication architectures. These filters have become popular because their precise reproducibility allows design engineers to achieve performance levels that are difficult to obtain with analog filters
Digital Filters can be constructed from 3 fundamental mathematical operations.

- Addition (or subtraction)
- Multiplication (normally of a signal by a constant)
- Time Delay i.e: delaying a digital signal by one or more sample periods
-

Figure 1 shows a graphical means of describing a digital filter whereby the behavior of the filter is described by using the mathematical operations mentioned above.

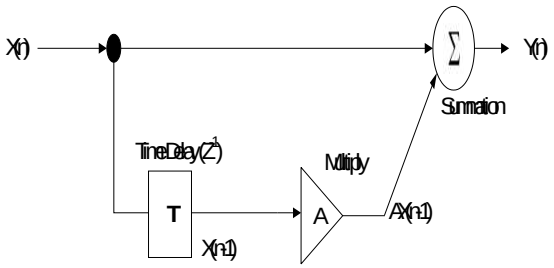

Figure 1 Block diagram of a Simple Digital Filter

The Impulse Response of a digital filter, *h(n)* is the response of the filter to an input consisting of the unit impulse function, $\delta(n)$. If the impulse response of a system is known, it is possible to calculate the system response for any input sequence $x(n)$. By definition, the unit impulse is applied to a system at sample index n=0. So, the impulse response is non-zero only for values of n greater than or equal to zero i.e *h(n)* is zero for n<0. This impulse response is said to be causal otherwise the system would be producing a response before an input has been applied. It is known from the time-invariance property of a Linear Time Invariant System that the response of a system to a delayed unit impulse $\delta(n-k)$ will be a delayed version of the unit impulse, i.e $h(n-k)$. It is also known from the linearity property that the response of a system to a weighted sum of inputs will be a weighted sum of responses of the system to each of the individual inputs. Therefore, the response of a system to an arbitrary input $x(n)$ can be written as follows:

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k)$$

IIR Vs FIR: FIR and IIR filters are the two common filter forms. A drawback of IIR filters is that the closed-form IIR designs are preliminary limited to low pass, band pass, and high pass filters. Furthermore, these designs generally disregard the phase response of the filter. For example, with a

relatively simple computational procedure excellent amplitude response characteristics may be obtained with an elliptic low pass filter while the phase response will be very nonlinear.

In designing filters and other signal-processing system that pass some portion of the frequency band undistorted, it is desirable to have approximately constant frequency response magnitude and zero phases in that band. For casual systems, zero phases are not attainable, and consequently, some phase distortion must be allowed. As the effect of linear phase with integer slope is a simple time shift.

A nonlinear phase, on the other hand, can have a major effect on the shape of a signal, even when the frequency-response magnitude is constant. Thus, in many situations it is particularly desirable to design systems to have exactly or approximately linear phase.

Compared to IIR filers, FIR filters can have precise linear phase. Also, in the case of FIR filters, closed-form design equations do not exist. While the window Method can be applied in a straightforward manner, some iteration may be necessary to meet a prescribed specification.

The window method and most algorithmic methods afford the possibility of approximating more arbitrary frequency response Characteristics with little more difficulty than is encountered in the design of low pass filters. Also, it appears that the design problem for FIR filters is much more under control than the IIR design problem because there is an optimality theorem for FIR filters that is meaningful in a wide range of practical situations.

The magnitude and phase plots provide an estimate of how the filter will perform; however, to determine the true response, the filter must be simulated in a system model using either calculated or recorded input data. The creation and analysis of representative data can be a complex task. Most of the filter algorithms require multiplication and addition in real-time. The unit carrying out this function is called MAC (multiply accumulate).

The better MAC the better performance can be obtained. Once a correct filter response has been determined and a coefficient table has been generated, the second step is to design the hardware architecture. The hardware designer must choose between area, performance, quantization, architecture, and response.

2. FIR IMPLEMENTATION

Finite Impulse Response (FIR) filters are one of two primary types of filters used in DSP, the other type being Infinite Impulse Response Filters (IIR) filters.

The impulse response of an FIR filter is “finite" because there is no feedback in the filter. Compared to IIR filters, FIR filters offer the following advantages:

They can easily be designed to be "Linear Phase”. Linear-Phase filters delay the input signal, but don’t distort its phase.

They are simple to implement. On most DSP microprocessors, looping a single instruction can do the FIR calculation.

FIR filters are suited to multi-rate applications. i.e: reducing the sampling rate (decimation) or increasing the sampling rate (interpolation), 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 that output will be discarded (so the feedback will be incorporated into the filter).

FIR filters have desirable numeric properties. In practice, all DSP filters must be implemented using "finite-precision" arithmetic, that is, a limited number of bits. The use of finite-precision arithmetic in IIR filters can cause significant problems due to the use of feedback, but FIR filters have no feedback, so they can usually be implemented using fewer bits, and the designer has fewer practical problems to solve related to non-ideal arithmetic.

FIR filters can be implemented using fractional arithmetic. Unlike IIR filters, it is always possible to implement a FIR filter using coefficients with magnitude of less than 1.0. (The overall gain of the FIR filter can be adjusted at its output, if desired.)

A disadvantage of using FIR filters is that they require more co-efficients than an IIR filter in order to implement the same frequency response, therefore needing more memory and more hardware resources to carry out mathematical operations.

Some terms used in describing FIR filters are as follows:

Impulse Response - The "impulse response" of an FIR filter is the set of FIR coefficients. Putting an impulse into a FIR filter which consists of a "1" sample followed by many "0" samples, the output of the filter will be the set of coefficients, as the 1 sample moves past each coefficient in turn to form the output.

Tap - A FIR tap is a coefficient/delay pair. The number of FIR taps, often designated as "N" is an indication of ; the amount of memory required to implement the filter, the number of calculations required, and the amount of "filtering" the filter can do. More taps means more stop-band attenuation, less ripple, narrower filters, etc.

Multiply-Accumulate (MAC) - In a FIR context, a MAC is the operation of multiplying a coefficient by the corresponding delayed data sample and accumulating the result. FIR filters usually require one MAC per tap. Most DSP microprocessors implement the MAC operation in a single instruction cycle.

3. LUT BASED FIR DESIGN

It reduces the logic needed to implement MAC operator to only adders and shifters. It can be implemented on LUT "Look−up−table" based FPGAs without wasting resources. It has several implementation based on several trade-offs. Most FIR filters have constant operators so using this algorithm they can be implemented with very custom circuits. The advantage of this approach is its efficiency of mechanization.

Significant work has been done on efficient computation of sinusoidal transforms and filters.

Various algorithm-architecture co-designs have also been reported for efficient LUT-multiplier-based implementation of sinusoidal transforms. In an early paper had introduced a memory-based structure for the LUT multiplier based implementation of FIR filter.

But, we do not find any further work to improve the efficiency of LUT multiplier based implementation of FIR filter. In this paper, we aim at presenting two new approaches for designing the LUT for LUT-multiplier-based implementation, where the memory-size is reduced to nearly half of the conventional approach.

Besides, we find that instead of direct-form realization, transposed form realization of FIR filter is more efficient for the LUT multiplier based implementation. In the transposed form, a single segmented-memory core could be used instead of separate memory modules for individual multiplications in order to avoid the use of individual decoders for each of those separate modules.

It consists of memory-units for conventional LUT-based multiplica- tion, along with AS cells and a delay register.

During each cycle, all the 8 bits of current input sample are fed to all the LUT-multipliers in parallel as a pair of 4-bit addresses.

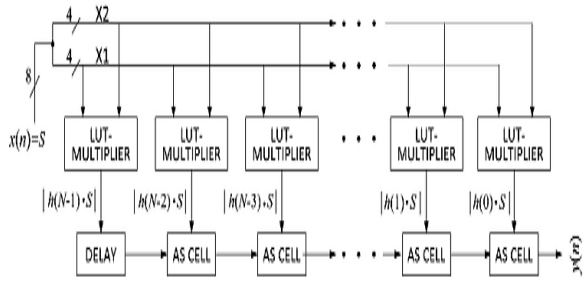

Figure 2 Conventional LUT-multiplier-based structure of an n-tap transposed form FIR

4. MEMORY BASED FIR DESIGN

Memory-based structures or Memory-based systems for those system. where memory elements like RAM or ROM is used either as a part or whole of an arithmetic unit. Memory-based structures are more regular compared with the multiply-accumulate structures, and have many other advantages, e.g., greater potential for high-throughput and reduced-latency implementation, (since the memory-access-time is much shorter than the usual multiplication-time) and are expected to have less dynamic power consumption due to less switching activities for memory-read operations compared to the conventional multipliers. Memory-based structures are well-suited for many digital signal processing (DSP) algorithms, which involve multiplication with a fixed set of coefficients. Several architectures have been reported in the literature for memory-based implementation of discrete sinusoidal transforms and digital filters for DSP applications.

It consists of a dual-port memory unit of size (consisting of 16 words of -bit width) and a shift-add (SA) cell. The SA cell shifts its right-input to left by four bit-locations and adds the shifted value with its other input to produce a -bit output.

The shift operation in the shift-add cells is hardwired with the adders, so that no additional shifters are required. The outputs of the multipliers are fed to the pipeline of AS cells in parallel. Each AS cell performs exactly the same function as that of the AS node of the DFG. It consists of either an adder or a sub tractor depending on whether the corresponding filter weight is positive or negative, respectively.

Besides, each of the SA cells consists of a pipeline latch corresponding to the delays. The FIR filter structure takes one input sample in each clock cycle, and produces one filter output in each cycle. The first filter output is obtained after a latency of three cycles (1 cycle each for memory output, the SA cell and the last AS cell). But, the first outputs are not correct because they do not contain the contributions of all the filter coefficients. The correct output of this structure would thus be available after a latency of cycles.

5. RESULTS AND CONCLUSIONS

The waveforms for the lut based and memory based designs are shown below.

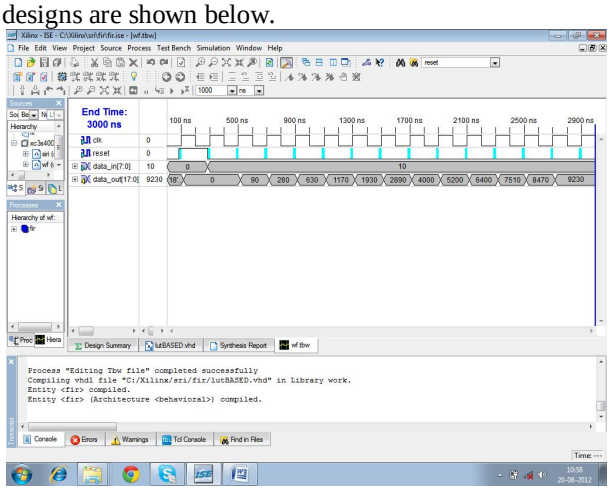

Figure 3 LUT Based FIR Design

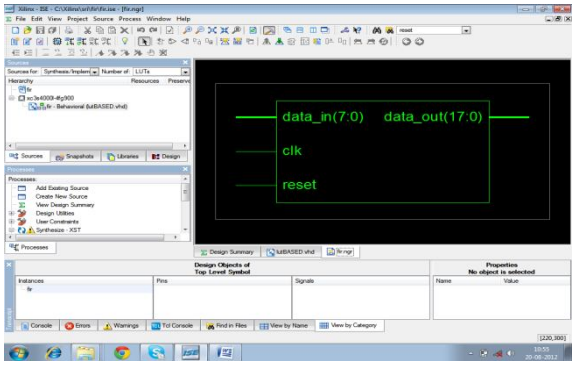

Figure 4 LUT Based FIR Schematic

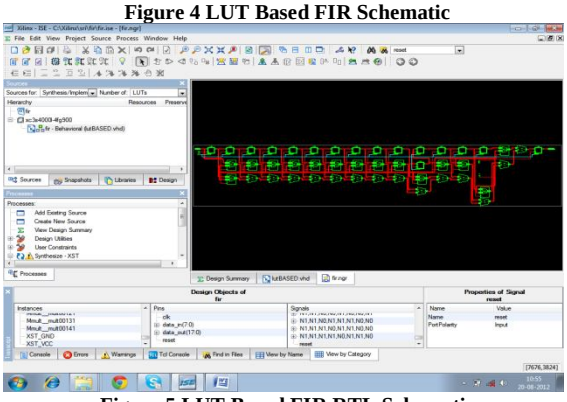

Figure 5 LUT Based FIR RTL Schematic

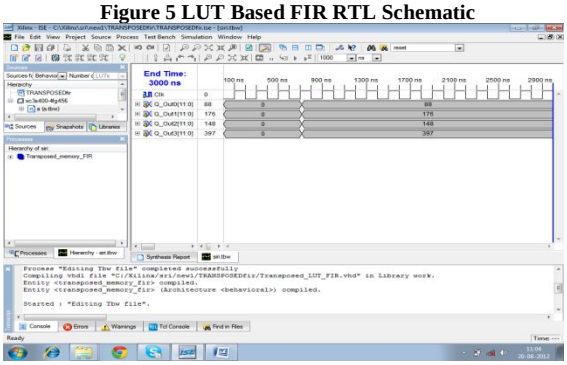

Figure 6 Memory Based FIR Design

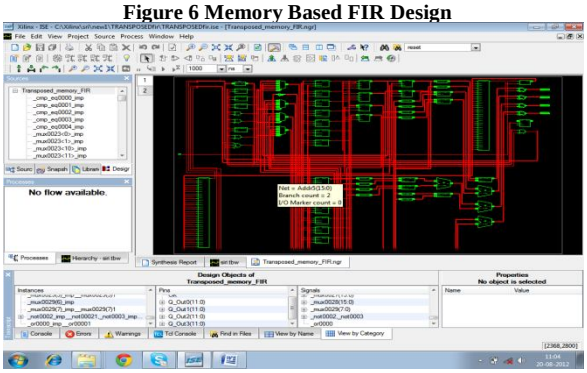

Figure 7 Memory based RTL schematic

The total memory usage for LUT based and Memory based designs are 229224 kilobytes and 203488 kilobytes. The time taken for evaluating the response for for LUT based and Memory based designs are 7.165ns and 14.927ns.

Acknowledgements

The authors would like to thank the anonymous reviewers for their comments which were very helpful in improving the quality and presentation of this paper.

REFERENCES:

[1] D. Xu and J. Chiu, "Design of a high-order FIR digital filtering and variable gain ranging seismic data acquisition system," in Proc. IEEE Southeastcon'93, Apr. 1993, p. 6.

[2] K. K. Parhi, VLSI Digital Signal Procesing Systems: Design and Implementation. New York: Wiley, 1999.

[3] H. H. Kha, H. D. Tuan, B.-N. Vo, and T. Q. Nguyen, "Symmetric orthogonal complex-valued filter bank design by semidefinite programming," IEEE Trans. Signal Process., vol. 55, no. 9, pp. 4405–4414, Sep. 2007.

[4] H. H. Dam, A. Cantoni, K. L. Teo, and S. Nordholm, "FIR variable digital filter with signed power-of-two coefficients," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 54, no. 6, pp. 1348–1357, Jun. 2007.

[5] R. Mahesh and A. P. Vinod, "A new common subexpression elimination algorithm for realizing low-complexity higher order digital filters," IEEE Trans. Computer-Aided Ded. Integr. Circuits Syst., vol. 27, no. 2, pp. 217–229, Feb. 2008.

[6] International Technology Roadmap for Semiconductors, [Online]. Available: http://public.itrs.net/

[7] B. Prince, "Trends in scaled and nanotechnology memories," in Proc. IEEE Conf. Custom Integr. Circuits, Nov. 2005, pp. 7–10.

[8] K. Itoh, S. Kimura, and T. Sakata, "VLSI memory technology: Current status and future trends," in Proc. 25th Eur. Solid-State Circuits Conference,{ ESSCIRC'99, Sep. 1999, pp. 3–10.

[9] T. Furuyama, "Trends and challenges of large scale embedded memories," in Proc. IEEE Conf. Custom Integrated Circuits, Oct. 2004, pp. 449–456.

[10] D. G. Elliott, M. Stumm, W. M. Snelgrove, C. Cojocaru, and R. Mckenzie, "Computational RAM: Implementing processors in memory," IEEE Trans. Design Test Comput., vol. 16, no. 1, pp. 32–41,Jan. 1999.

Authors Profile:

A.Sirisha is pursuing M.Tech in VLSI &ES at Chirala Engineering College,Chirala.

P.Balanagu,M.Tech in Micro electronicsfrom VTU,Belgam Working as Assoc.prof in ECE Dept. CEC,Chirala. He has 9 years of teaching Experience.

Prof.N.Suresh Babu is Vice- Principal & HOD of ECE Dept in CEC,Chirala.He got his M.Tech in Microwave Engineering from Birla Innstitute of technology, Ranchi. He has 13 years of teaching Experience and 2 Years of Industrial Experience in various organisations .

◈◈◈