Transposed Structure Design of FIR Filter using VHDL

Hanny Kumar¹ and Kamal Kumar²

Abstract

The paper focuses on the design of the Finite Impulse response Filter (FIR) Filter using VHDL programming language. In the FIR filter design the two input sequence $x(n)$ and $h(n)$ are considered of length $M = 4$ and $N = 4$ for both respectively. The output of FIR filter is the convolution of two sequences with the tap length $(M + N - 1 = 7)$. The design considerations are followed to test many input sequences. The transposed structure is considered for the design of FIR filter to optimize the delay. The design in developed in Xilinx 14.2 software and waveform are simulated in modelsim10.1 software. The design is also synthesized on SPARTAN -3E FPGA.

Keywords

Finite Impulse Response (FIR), Field Programmable Gate Array (FPGA), Very High Speed Integrated Circuit Hardware Description language (VHDL)

Introduction

The FIR filter is one of the moving averages (MA) or all zero filters that accepts present input and past input samples and produces the output results. In the practical way the designing of the FIR filter is easy to control the multiple applications of DSP. The filter is completely capable to provide linear phase because of its behavior of linear and arbitrary can approximate frequency response. It is non-recursive in nature and the design uses the delay elements and weight coefficients. The filter does not have the feedback so it is sable filter. The differential equation is given below which can be used for the direct and transposed structure of the FIR filter.
The another category of the filter is infinite impulse response (IIR) filter which has pole and zero auto regressive nature or moving average (ARMA). The calculation of the filter depends on the present samples and past samples both. The filter has the feedback this nature of the filter provides that the filter has impulse response which stays for infinite period or duration. The major advantage of the IIR filter is that the filter can be designed in a simple way. Different types of filter which lies in the frequency selective range can be designed using equation based closed form. Another benefit of the IIR filter is that it has computation economy with designing and implementation point of view. The frequency response of the filter can be achieved for high frequency selection with the help of recursive coefficients.

The IIR filter cannot give the output response in linear phase and its hardware implementation is more complex in nature. Moreover, in some cases phase characteristics are not a major concern then the magnitude response can be analyzed and implemented in an efficient manner using IIR filter. The equation below gives difference equation to explore the structure of IIR filter.

\[ y[n] = 2x[n] + 5x[n - 1] + 7x[n - 2] + 4y[n - 1] + 3y[n - 2] \]

**Figure 2: IIR filter structure**

**Frequency Response of FIR Filter**

The filter in frequency response is an outcome of the spectrum of a time invariant and linear phase filter, it results the product of the filter frequency response and input signal spectrum. The frequency response of such filter specifies that the filter must determine...
the band of frequency in which it can work with the input signal of the filter and it can pass
the signal will not reject automatically.

![Passband and Stopband of FIR filter](image)

*Figure 3: Passband and Stopband of FIR filter*

The desired portion of the filter can be chosen from the frequency response and the
design engineers of the filter. The data from the time domain to frequency domain can be
transformed using Fast Fourier Transform (FFT) filter directly in frequency domain that
complete pass the desired frequency and removes the unwanted portion. The inverse
transform returns back to the time domain with the help of inverse fast Fourier transform
(IFFT).

![Domain presentation of FIR filter](image)

*Figure 4: Domain presentation of FIR filter*

The outputs of the IFFT are then concatenated together if necessary. Direct Frequency
Domain filtering is NOT fast convolution (which also uses the FFT & IFFT to filter the
signal in conjunction with the Overlap Add and Overlap Save algorithms). Direct
Frequency Domain filters are also known as Fourier Filters. Let us first compare a Fourier
Filter with a conventional Finite Impulse Response (FIR) filter. Both are linear filters —
this means that the outputs of both filters are stable and scale linearly with the input.

![Frequency Domain characteristics of a low pass FIR filter](image)

*Figure 5: Frequency Domain characteristics of a low pass FIR filter*
All practical FIR filters have loss characteristics, as seen in the figure above. These loss characteristics can be reduced by increasing the number of taps or coefficients of the filter, but can never be totally eliminated.

**Filter Design**

The FIR filter design can be understood with the help of following example of two sequences Design Considerations. The output of the FIR filter output can be determined using the following equation:

\[ y[n] = \sum_{k=0}^{N-1} h[k] \cdot x[n-k] \]

Where,

- \( x[k] \) = Presents the input samples of FIR filter

- \( h[k] \) = Weight coefficients of FIR filter under frequency response, and

- \( y[n] \) = Output samples of FIR filter.

Example: \( x(n) = \{1,1,1,1\} \) and \( h(n) = \{1,2,3,4\} \) then \( Y(n) = x(n) \otimes h(n) \)

\[
\begin{array}{c|cccc}
H & 1 & 1 & 1 & 1 \\
\hline
1 & 1 & 1 & 1 & 1 \\
2 & 2 & 2 & 2 & 2 \\
3 & 3 & 3 & 3 & 3 \\
4 & 4 & 4 & 4 & 4 \\
\end{array}
\]

*Figure 6: Linear convolution as FIR filter example*

FIR filter output

\( Y(n) = \{1, 3, 6, 10, 9, 7, 4\} \)

The implementation of the FIR filter is generally followed by using the linear convolution.

**Simulation Results**

The modesim waveform of devoped FIR filter is shown in fig. 7. The input sequeuces of the filter design are \( x(n) \), \( x(n-1) \), \( x(n-2) \) and \( x(n-3) \) followsthe tap coefficients values \( h(0) \), \( h(1) \), \( h(2) \) and \( h(3) \) as weight coefficients. The output samples of the FIR filter are presented using the FIR filter output FIR1, FIR2, FIR3, FIR4, FIR5, FIR6 and FIR7.
Transposed Structure Design of FIR Filter using VHDL.

Figure 7: Modelsim Simulation of FIR Filter

Figure 8: Modelsim Simulation of FIR in Binary
Conclusions

The design and implementation of FIR filter is done successfully using VHDL programming language. The design is tested for several test inputs and verified on SPARTAN 3E FPGA. The tap length of FIR filter is 7. The design is following the transposed structure of FIR filter. The direct form of realization takes more delay in comparison to transpose form of FIR filter. In the design the time required to develop the design is 15.00 ns. In future the design of FIR filter can be designed for large number of tap FIR filter.

References


This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/).

© 2016 by the Authors. Licensed by HCTL Open, India.