SIMULATION AND SYNTHESIS OF FILTER

Vishwajit K. Barbudhe,
Assistant Professor
Jagadambha college of Engineering & Technology
Master of Engineering (ME), Electronics & Telecommunication Engineering Department,
Yavatmal, India

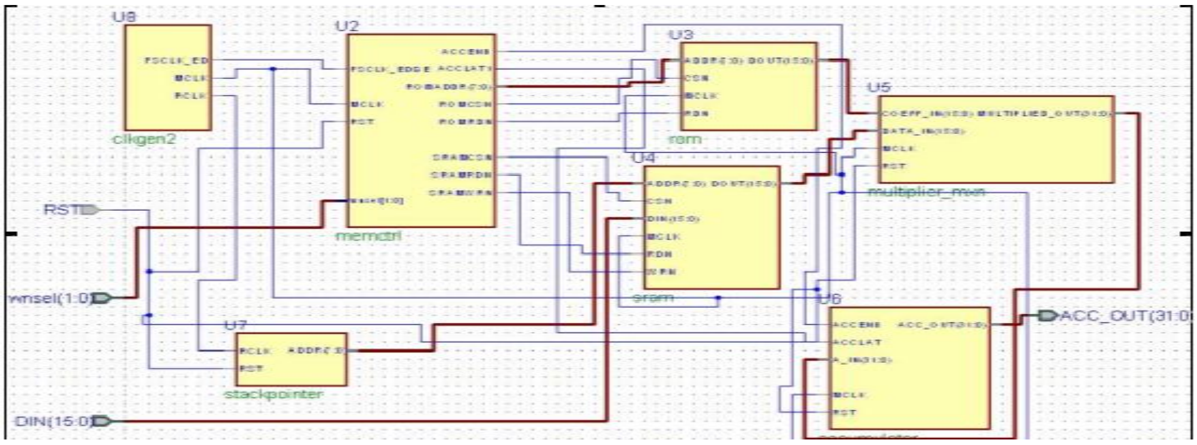
ABSTRACT

This simulation model will incorporate a mixture of both Behavioral as well as Structural modeling constructs that were briefly described above. The above aim has been furthered by taking a stepwise approach, that is, behavioral as well as dataflow simulation models of less complex devices such as program counter, memory controller, memory modules (SRAM & ROM), multiplier, accumulator and stack pointer have been developed and these models would be integrated together in the FIR filter model. All FIR filters support low pass, high pass, band pass, and band stop options. The FIR filters Rectangular, Bartlett, Hanning, Hamming, Blackman, Kaiser, and Dolph-Chebyshev are all Window FIR filters. The name "Window" comes from the fact that these filters are created by scaling a sinc (sin(x)/x) pattern with a window to produce the desired frequency effect.

Introduction:

Due to increased complexity and size of digital systems, the use of sophisticated design entry, verification and automatic hardware generation tools is rapidly gaining ground. One of the newest additions to this design methodology is the introduction of Hardware Description Languages (HDL). In a hardware design environment, the digital design can be represented as a HDL source file or it can be systematically captured. In this paper an attempt has been made to develop a simulation model of an FIR filter using VHDL. This simulation model will incorporate a mixture of both Behavioral as well as Structural modeling constructs that were briefly described above. The above aim has been furthered by taking a stepwise approach, that is, behavioral as well as dataflow simulation models of less complex devices such as program counter, memory controller, memory modules (SRAM & ROM), multiplier, accumulator and stack pointer have been developed and these models would be integrated together in the FIR filter model.

Block Diagram:



One of the best uses of VHDL today is to synthesize ASIC and FPGA devices. Synthesis is an automatic method of converting a higher level of abstraction to a lower level of abstraction. There are several synthesis tools available currently. In our project we have used the commercially available Exemplar Logic Leonardo Spectrum synthesis tool. It converts Register Transfer Level (RTL) descriptions to gate level netlists. These gate level netlists consist of interconnected gate level macro cells. These gate level netlists currently can be optimized for area, speed, testability, and so on. Models for the gate level cells are contained in technology libraries.

Filters

The term filter is commonly used to describe a device that discriminates, according to some attribute of the objects applied at its input, what passes through it. For example, an air filter allows air to pass through it but prevents dust particles that are present in the air from passing through. An oil filter performs a similar function, with the exception that oil is the substance allowed to pass through the filter, while particles of dirt are collected at the input to the filter and prevented from passing through. In photography, an ultraviolet filter is often used to prevent ultraviolet light, which is present in sunlight and which is not a part of visible light, from passing through and affecting the chemicals on the film.

Frequency Response of Filters

A frequency-selective filter is a device that passes a certain range of frequencies and blocks the rest. The range of frequencies passed defines the passband, and the range of frequencies blocked defines the stopband. The band-edge frequencies are called the cutoff frequencies. Ideally, a filter should have perfect transmission in the passband and perfect rejection in the stopband. The phase and group delay of a system whose transfer function has Magnitude= H(ω) and Phase= Φ(ω) are defined as:

$$t_p = -\frac{\Phi(\omega)}{\omega} \qquad \text{(Phase delay)}$$

$$\tau_g(\omega) = -\frac{d\Phi(\omega)}{d(\omega)} \qquad \text{(Group delay)}$$

If Φ(ω)varies linearly with frequency, t_p and τ_g are not only constant but also equal. They are simple to implement. On most DSP microprocessors, the FIR calculation can be done by looping a single instruction. They are suited to multi-rate applications. By multi-rate, we mean "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). They 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. They 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.) This is an important consideration when using fixed-point DSP's, because it makes the implementation much simpler.

Direct-Form Structure

The direct-form realization follows immediately from the non-recursive difference equation

$$y(n) = \sum_{k=0}^{M-1} h(k)x(n-k)$$

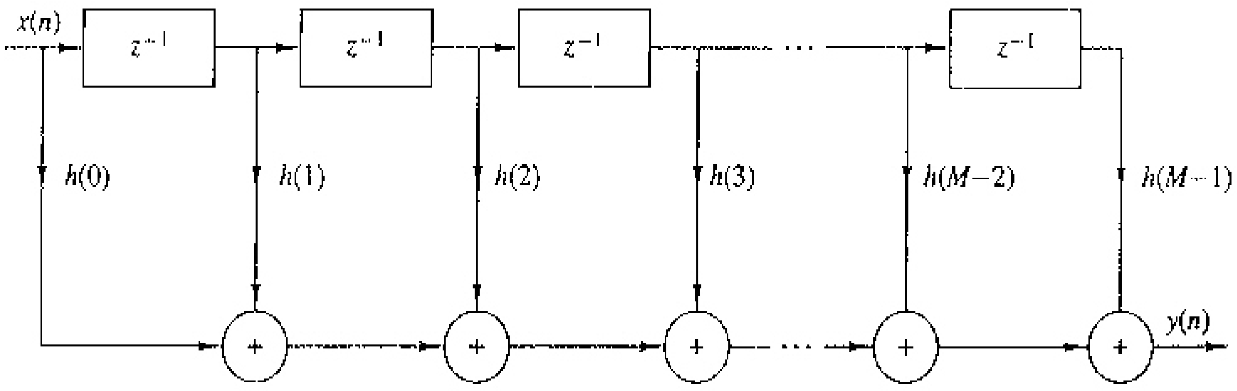


Fig ***Direct-form realization of Fir system***

We observe that this structure requires M-1 memory locations for storing the M-1 previous inputs and has a complexity of M multiplications and M-1 additions per output point. Since the output consists of a weighted linear combination of M-1 past values of the input and the weighted current value of the input, the structure in figure resembles a tapped delay line or a transversal system. Consequently, the direct-form realization is often called a transversal or tapped-delay-line filter.

Observations:

- There are 'M-1' delay blocks, so this is a canonic structure.
- Input signal is delayed 'M-1' times; so to store this delayed input signal 'M-1' memory locations are required.
- This structure has 'M-1' additions and 'M' multiplications.
- Output y(n) is a weighted linear combination of present input and past inputs.

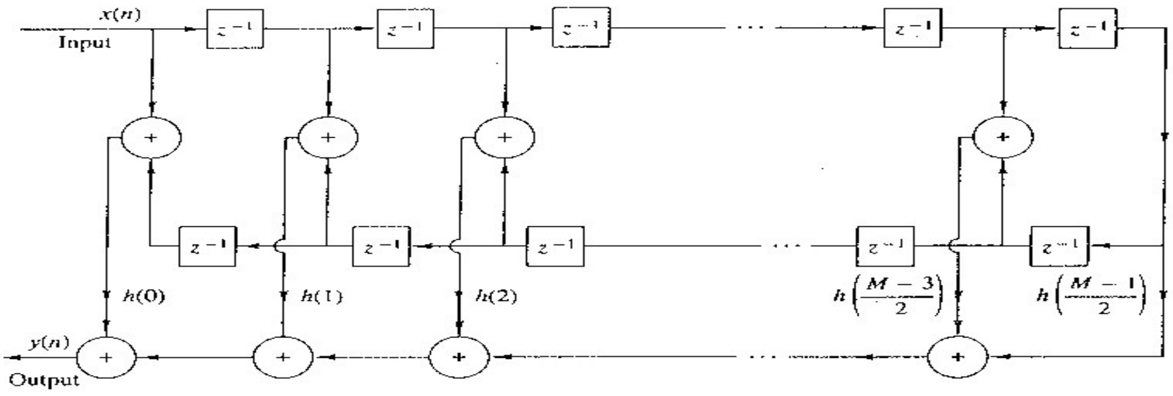


Fig ***Direct-form realization for M odd***

When The FIR system has linear phase, unit sample response of the system satisfies either the symmetry or asymmetry condition

$$h(n) = \pm h(M-1-n)$$

For such a system, the number of multiplications is reduced from M to M/2 for even M and to (M-1)/2 for odd M. The structure illustrated above has odd M.

Characteristics of Window Functions

The amplitude response of symmetric, finite-duration windows invariably shows a main lobe and decaying side lobes that may be entirely positive or that may alternate in sign. The spectral measures for a typical window are illustrated in Figure

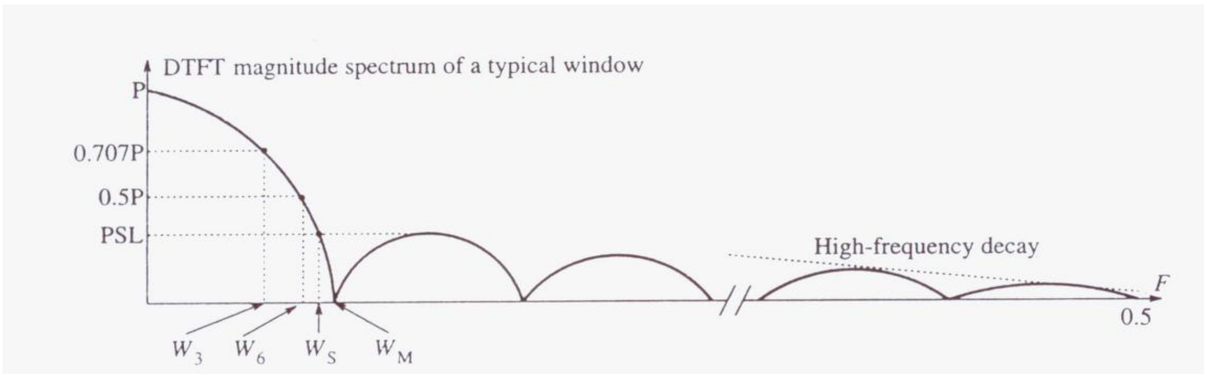


Fig. ***Magnitude spectrum of a typical window***

Amplitude-based measures for a window include the peak sidelobe level (PSL), usually in decibels, the decay rate Ds in dB/dec. Frequency-based measures include the mainlobe width W_M, the 3-dB and 6-dB widths (W_3 and W_6), and the width Ws to reach the peak sidelobe level. The windows commonly used in FIR filter design and their spectral features are listed in the Table and illustrated in Figure. As the window length N increases, the width parameters decrease, but the peak sidelobe level remains more or less constant. Ideally, the spectrum of a window should approximate an impulse and be confined to as narrow a mainlobe as possible, with as little energy in the sidelobes as possible.

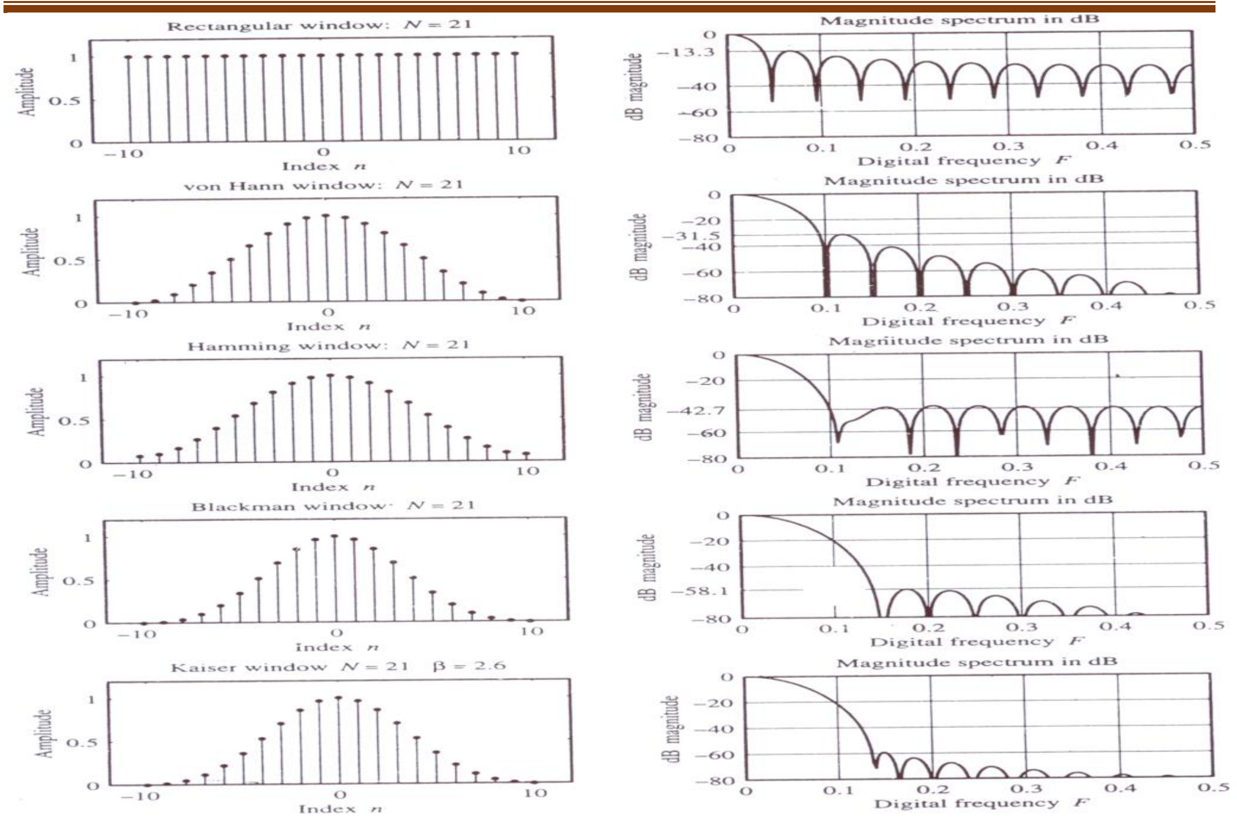


Fig. ***Spectral characteristics of some windows***

What Windowing Means

Symmetric windowing of the impulse response of an ideal filter is accomplished by windows that are themselves symmetric, as illustrated in Figure.

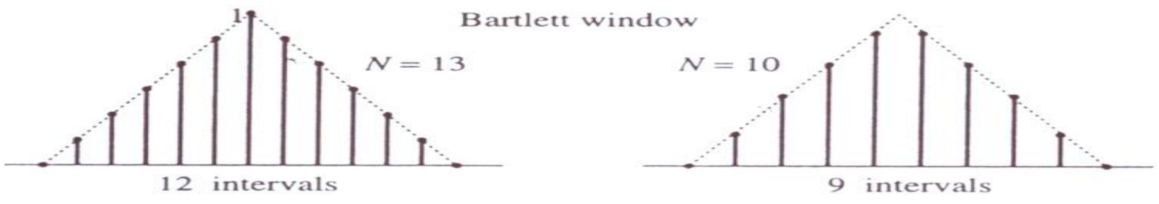


Fig. ***Bartlett window***

An N-point FIR window uses N - 1 intervals to generate N samples (including both end samples). Once selected, the window sequence must be symmetrically positioned with respect to the symmetric impulse response sequence. Windowing is then simply a point-wise multiplication of the two sequences.The symmetrically windowed impulse response of an ideal lowpass filter may be written as

$$h_{\omega}[n] = 2F_c \operatorname{sinc}(2nF_c)\omega[n] \qquad -0.5(N-1) \le n \le 0.5(N-1)$$

The Window Technique

The window technique is best described in terms of a specific example. We want to design a symmetric lowpass linear-phase FIR filter having a desired frequency response

$$H_d(\omega) = 1e^{-j\omega(N-1)/2}, \quad 0 \le |\omega| \le \omega_c$$
$$= 0 \quad , \quad \text{otherwise}$$

A delay of (N-1) / 2 units is incorporated into $H_d(\omega)$ in anticipation of forcing the filter to be of length N. The corresponding unit sample response $h_d(n)$ is non-causal and infinite in duration. If we multiply $h_d(n)$ by the rectangular window sequence,

$$W(n) = 1, \quad n = 0,1,\ldots,N-1$$
$$= 0, \quad \text{otherwise}$$

we obtain an FIR filter of length N having the unit sample response

$$h(n) = \frac{\sin \omega_c \left(n - \frac{N-1}{2}\right)}{\pi\left(n - \frac{N-1}{2}\right)} \qquad ,0 \le n \le N-1 \ \ , n \ne (N-1) / 2$$

If M is selected to be odd, the value of h(n) at n = (N-1)/2 is $h[(N-1) / 2] = \omega_c / \pi$

The magnitude of the frequency response $H(\omega)$ of this filter is illustrated in figure for N = 61 and N = 101.

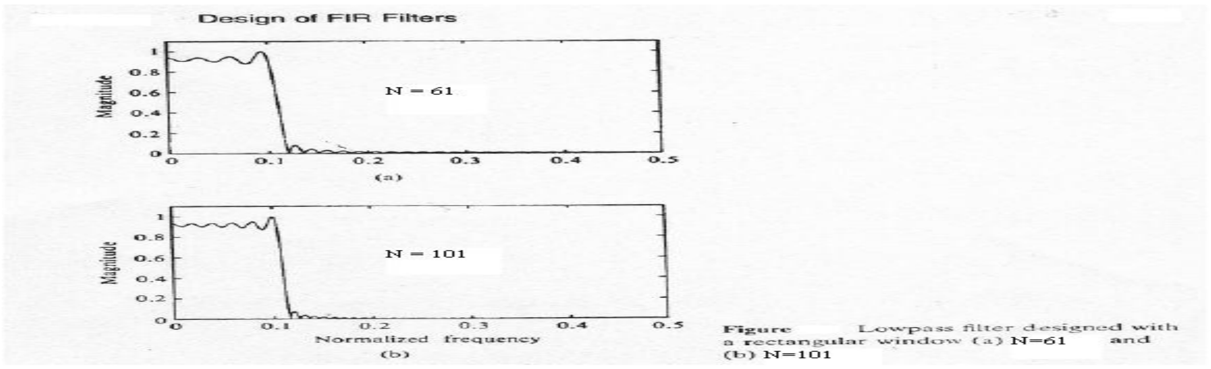


Fig. magnitude of the frequency response H(ω) of this filter

We observe that relatively large oscillations or ripples occur near the band edge of the filter. The oscillations increase in frequency as N increases, but they do not diminish in amplitude. These large oscillations are the direct result of the large sidelobes existing in the frequency characteristic $W(\omega)$ of the rectangular window. As this window function is convolved with the desired frequency response characteristic $H_d(\omega)$, the oscillations occur as the large constant area sidelobes of $W(\omega)$ move across the discontinuity that exists in $H_d(\omega)$. Since the equation

$$H_d(\omega) = \sum_{h=0}^{\infty} h_d(n)\, e^{-j\omega n}$$

is basically a Fourier series representation of $H_d(\omega)$, the multiplication of $h_d(n)$ with a rectangular window is identical to truncating the Fourier series representation of the desired filter characteristic $H_d(\omega)$. The truncation of the Fourier series is known to introduce ripples in the frequency response characteristic $H(\omega)$ due to the non-uniform convergence of the Fourier series at a discontinuity. The oscillatory behavior near the band edge of the filter is called the Gibbs phenomenon.

To alleviate the presence of large oscillations in both the passband and the stopband, we should use a window function that contains a taper and decays toward zero gradually, instead of abruptly, as it occurs in a rectangular window. Figures 1 through 4, illustrate the frequency response of the resulting filter when some of the window functions listed in Table are used to taper $h_d(n)$. As illustrated in Fig. 2.19 (1 through 4), the window functions do indeed eliminate the ringing effects at the band edge and do result in lower sidelobes at the expense of an increase in the width of the transition band of the filter.

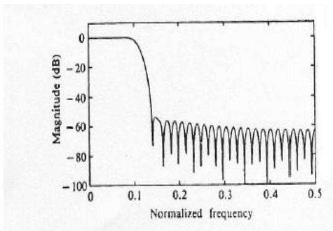


Figure 2 Lowpass FIR filter designed with Hamming window (N = 61)

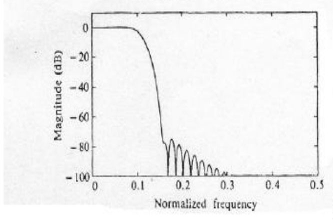


Figure 3 Lowpass FIR filter designed with Blackman window (N = 61)

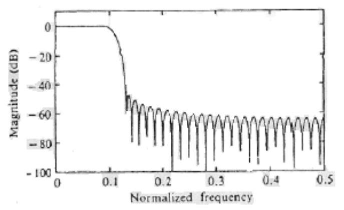


Figure 4 Lowpass FIR filter designed with Kaiser window (N = 61)

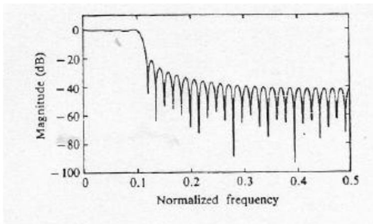


Figure 1 Lowpass FIR filter designed with rectangular window (N = 61)

Fig. Frequency response of designed filter

FPGA Design cycle

Define requirements: This stage is used for the design definition of the functions, required device, I/O pins locations and performance requirements.

VHDL coding entry: This stage describes the functional description of the architectures. Includes combinational logic design and RTL (Register Transfer Level) coding.

Synthesis: The *Synthesis* stage generates a netlist from the VHDL code. The netlist is a lower level abstraction of the code, which will be used for the place and route process.

Verification: This is used to verify the logical functionality of the circuit based on the netlist generated from the *Synthesis* stage.

Place and Route: This operation helps in creating the circuit layout using an automatic placement and routing tool. The main objective of placement is to ease routing of the design.

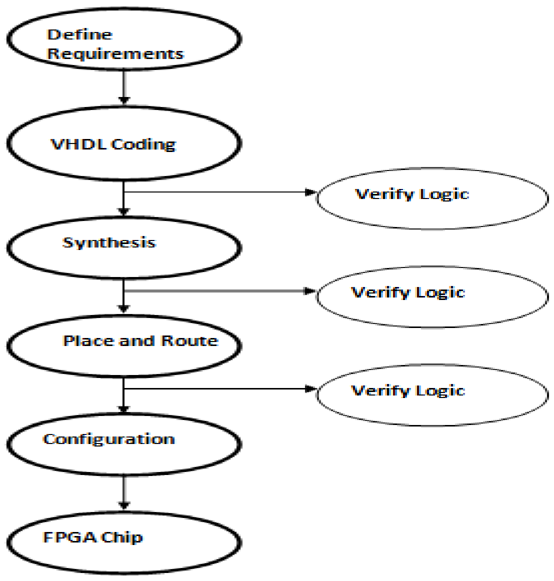


Figure *The FPGA design cycle*

Layout: This stage generally consists of a mask-layout of all circuit blocks that would be created using a Layout editor.

Verification: After the layout phase, the final performance verification is based on the parameters obtained with layout extraction programs. This stage basically consists of simulations to verify the functionality and performance of the entire chip.

Fabrication and Testing

Once the design has been verified by simulation, the chip layout has been determined, and there is evidence that the manufactured chip can be properly tested, the design data file is transmitted to the fabrication facility.

Results

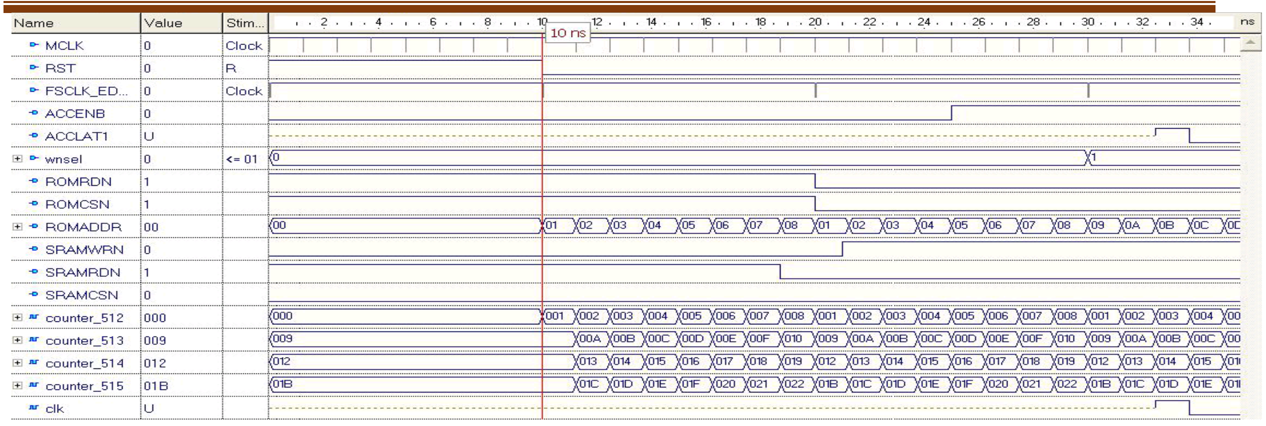
Finite Impulse Response (FIR) filters are used in every aspect of present day technology because filtering is one of the basic tools of information acquisition and manipulation. Digital filters form an integral part in digital signal processing applications. The FIR filter designed uses a window selection algorithm. The design can be easily modified to change the filter parameters according to the requirement. The different blocks were simulated in Active HDL and the waveforms verified the workability of the sub modules. The sub modules were integrated and the whole 8-tap 16-bit FIR filter was realized. The synthesis results were obtained on Leonardo spectrum for every block and they fit the technology. Renoir Graphics was used to generate the flow chart within the process declarations.

Simulated Waveforms

The initialize command launches elaboration and initialization of the simulation model. During elaboration, the simulator loads design units and builds the simulation model in the computer memory. During the initialization, all objects in the model (signals, variables, etc.) acquire their initial values (either default or explicitly specified) and all concurrent processes are executed once until their suspension. The simulator runs the process for the time specified in the run window. For our design we use steps of 1.25 nanoseconds.

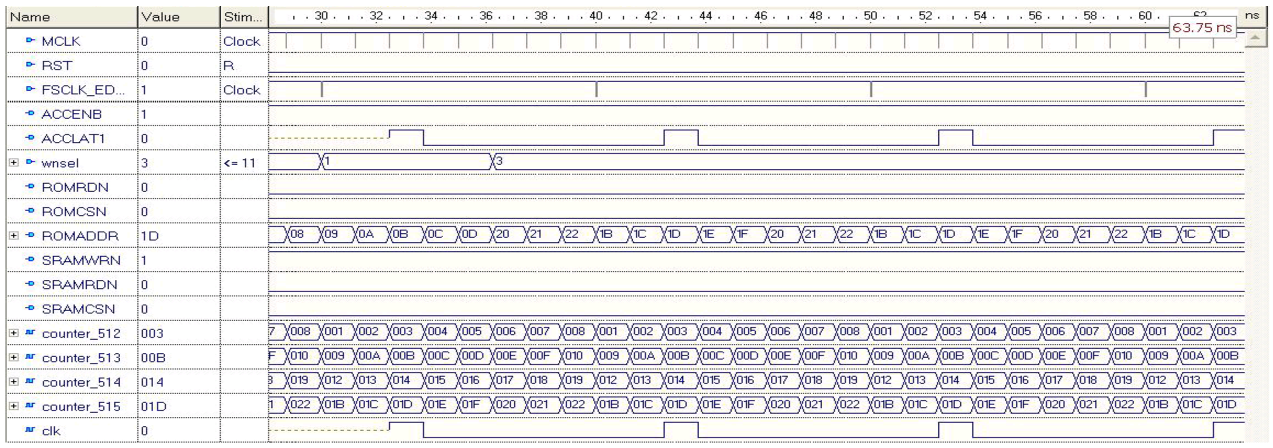
Memory Control Unit

The simulation of the memory control unit is shown on the next page. Initially the reset is kept asserted for 10ns during which the counters get loaded with the initial value. The window selection signal is kept at ‘00’ and this ensures that the addresses of the first set of coefficients in the ROM are generated. The ROM addresses are generated for the eight coefficients and after the last one it is reset to the first value. As the window selection signal is changed the addresses generated by the memory controller change accordingly. The red line shows the point where the system starts working.



Name	Value	Stim...
MCLK	0	Clock
RST	0	R
FSCLK_ED...	0	Clock
ACCENB	0	
ACCLAT1	U	
wnsel	0	<= 01
ROMRDN	1	
ROMCSN	1	
ROMADDR	00	
SRAMWRN	0	
SRAMRDN	1	
SRAMCSN	0	
counter_512	000	
counter_513	009	
counter_514	012	
counter_515	01B	
clk	U	

Fig Simulation result of the Memory Controller



Name	Value	Stim...
MCLK	0	Clock
RST	0	R
FSCLK_ED...	1	Clock
ACCENB	1	
ACCLAT1	0	
wnsel	3	<= 11
ROMRDN	0	
ROMCSN	0	
ROMADDR	1D	
SRAMWRN	1	
SRAMRDN	0	
SRAMCSN	0	
counter_512	003	
counter_513	00B	
counter_514	014	
counter_515	01D	
clk	0	

Fig Simulation Result of the Memory Controller (contd.)

References :

[1] lm,y.c.;and liu,b.,"design of cascade form fir Filters with discrete valued coefficients." *ieee* trans. Acousi speech *signal* pmcessing, vol 36 no 11, pp.1735- 1739,1988.
[2] parks, t. W.; and mcclellan, j. H. "chebyshev Approximation for nonrecursiv digital filters with linear Phase." *ieee* trans. Circuii theory vol 19, pp89-94. 1972.
[3] oppenhaim,a.v.digital signal pmcessing. Prentice Hall, 1975.
[4] bochnick, h.; anheier, w.. "f'ir filter design using Verilog and vhdl". Pmceedings *of* ihe nato advanced Study instirule *on* fundarnenials and standards *in* Hardware descripfion languages *11*, ciocco, barga, Italy.1993.
[5] khoo, k. *Y.;* kwentus a.; and willson, a.n. Jr., "an Efficient175 *mhz* programmablefir digital filter" Pmceeding *of* the *ieee h i* . Symp. Circuits and system Pp72-75, 1993.
[6]laskowski, *j;* and samueli, h., "a 150mhz 43tap Half-band fe? Digital filter in 1.2p cmos generated by Compiler. Pmceedings *of* the *ieee* cusi. *Ic* conference, 11.4.1-11.4.4
[7]evans, j.b., "an efficient fir filter architecture", Proceedings of ihe *ieee* 1ni.symposiurn *on* circuits and Systems, pp627-630, 1993.

[8]chen,s., mulgrew,b and grant, p m., "a clustering Technique for digital communications channel Equalization using radial basis function", *ieee* trans. Neural networks, vol4, pp570-578.

[9]mamun bin ibne reaz, md shahidul islam, md Shahiman sulaiman, md. Alauddin ali, "a multipurpose Fir filter design using vhdl, pmceedings *of* the *int. con/: on* robotics, *vision, in/:* and sigrtal.