IMPLEMENTATION OF VHDL TO DESIGN 64 TAP FIR FILTER

Aayush Khandelwal[1], Shweta Agrawal[2]
[1]Research Scholar, [2]Assistant Professor
[1,2]Department Of Electronics & Communications, SRCEM, Banmore, M.P. (INDIA)

ABSTRACT
VHDL (VHSIC Hardware Description Language) is a hardware description language used in electronic design automation to describe digital and mixed-signal systems such as FPGA, DSP Systems and ICs. VHDL can also be used as a general purpose parallel programming language. In this paper, VHDL is used to design 64 tap FIR filter.
Keywords: VHDL, 64 tap, FIR Filter

INTRODUCTION

FILTER

In signal processing, a **filter** is a device or process that removes some unwanted components or features from a signal. Filtering is a class of signal processing, the defining feature of filters being the complete or partial suppression of some aspect of the signal. Most often, this means removing some frequencies or frequency bands. However, filters do not exclusively act in the frequency domain; especially in the field of image processing many other targets for filtering exist.

There are many different bases of classifying filters and these overlap in many different ways; there is no simple hierarchical classification. Filters may be:

- non-linear or linear
- time-variant or time-invariant , also known as shift invariance. If the filter operates in a spatial domain then the characterization is space invariance.
- causal or not-causal: A filter is non-causal if its present output depends on future input. Filters processing time-domain signals in real time must be causal, but not filters acting on spatial domain signals or deferred-time processing of time-domain signals.
- analog or digital
- discrete-time (sampled) or continuous-time
- passive or active type of continuous-time filter
- infinite impulse response (IIR) or finite impulse response (FIR) type of discrete-time or digital filter.

FIR FILTER

"FIR" means "Finite Impulse Response". If we put in an impulse, that is, a single "1" sample followed by many "0" samples, zeroes will come out after the "1" sample has made its way through the delay line of the filter.

In the common case, the impulse response is finite because there is no feedback in the FIR. A lack of feedback guarantees that the impulse response will be finite. Therefore, the term "finite impulse response" is nearly synonymous with "no feedback".

However, if feedback is employed yet the impulse response is finite, the filter still is a FIR. An example is the moving average filter, in which the nth prior sample is subtracted (fed back) each time a new sample comes in. This filter has a finite impulse response even though it uses feedback: after N samples of an impulse, the output will always be zero.

EXPERIMENTAL WORK

General DescriptionThe designed FIR filter implements the function,

$$H(z) = \sum_{n=0}^{o-1} b[n] \cdot z^{-n}$$

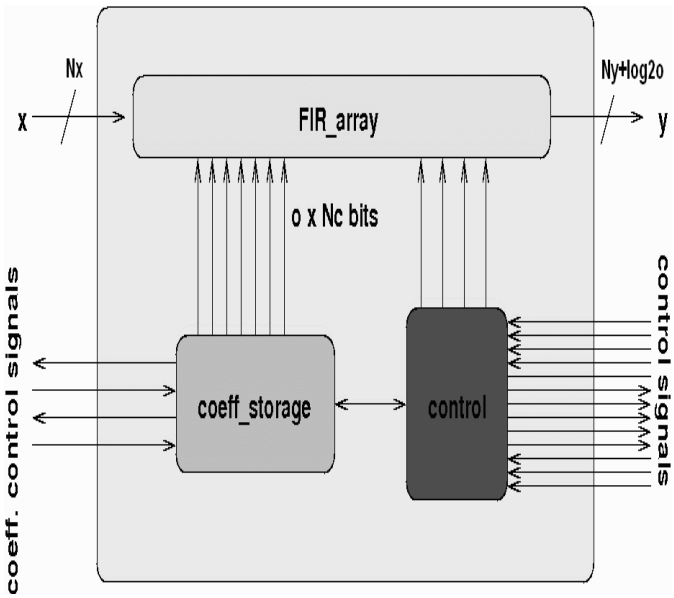
The architecture of the filter is fully sequential using one time-multiplexed multiplier and one adder. The outputs of filter are registered. Coefficients b are stored in the internal register array and can be read/written using bus.

This FIR model allows bit-level performance modelling. Note that the output word length is 22 bits. The input data and coefficients are both 8 bit word length:

An 8-bit by 8-bit multiply produces a 16 bit result,

There are 64 taps, $\log_2 N = 6$,

16 + 6 gives 22 bits for the output.



Top level schema of the FIR filter

Interface

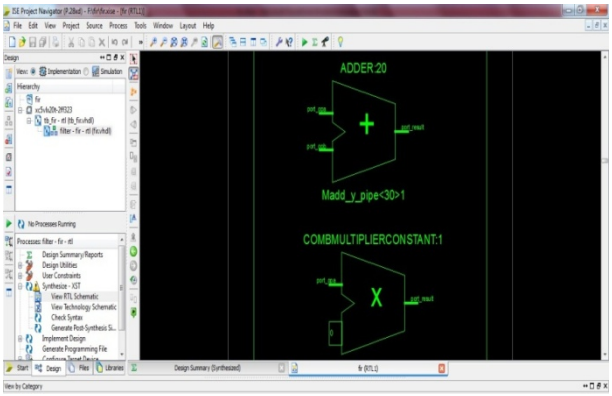
Generic	Type	Description
o	natural	The order of the filter
log2o	natural	Used for data path width – saturation margin
Nx	natural	Input data word width
Nc	natural	Coefficient data word width
Ny	natural	Output data word width

Co-efficient Addressing

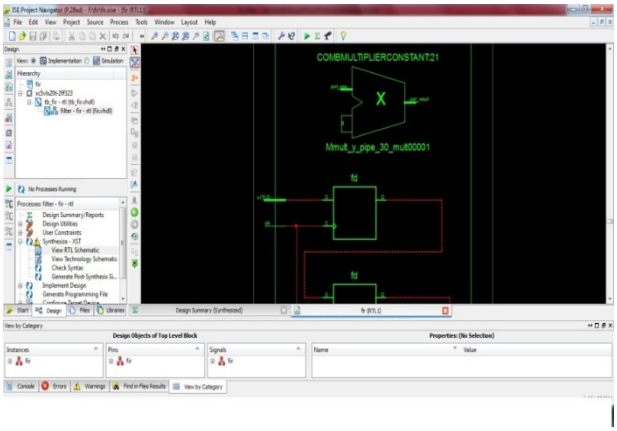
Address Bus	Mapped Co-efficent
FFEF	b[0]
FFED	b[1]
FFE8	b[2]
FFE6	b[3]

RTL DESCRIPTION

Top level circuit consists of three blocks. Adder, Multiplexer and D Flip Flop.



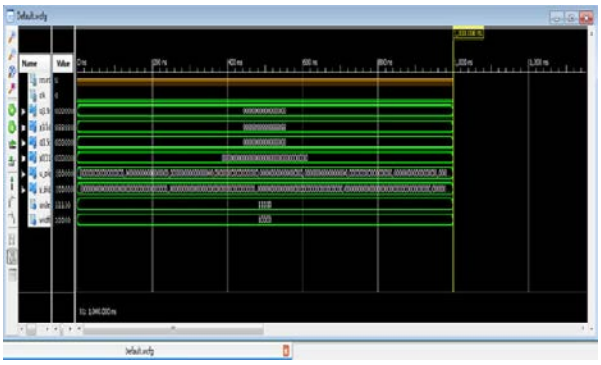
Schema1



Schema2

TEST BENCH VERIFICATION

Signal zero is set after to all pipeline registers at the input of the filter are written values. Then, by rising edge of clock, pipeline registers and also output and input registers are reset. The waveform of the test bench is in Figure.



CONCLUSION

The designing of the generic 64-tap FIR filter was successful using VHDL. The design is suitable for applications that require low power consumption and lesser area. The main advantage of the design is that they can be implemented as a reference model for the design of a synthesizable FIR filter.

REFERENCES

a. T.W. Parks and C.S. Burrus, Digital Filter Design. NewYork:Wiley,1987

b. Richard S. Juszkiewicz, An Analysis of Interpolated Finite Impulse Response Filters and Their improvements, IEEE Signal Processing Magazine, November2005

c. Xilinx FPGA datasheets,www.xilinx.com

d. J.G. Proakis and D.G. Manolakis, Digital Signal Processing-Principles, Algorithms and Applications New Delhi: Prentice-Hall, 2000