Reconfigurable Interpolation Filter With Efficient Delay Profile Using Improved Multiplication Algorithm

Sumee Sahu[1], Prof. Kamna Mishra[2]

[1]*M. Tech. Scholar,* [2]*Guide*
Department of Electronics and Communication Engineering, LNCT, Bhopal

Abstract - In various modern digital signal processing applications, sampling rate needs to be increased digitally to compensate the inter symbol interference and to deal with the high bandwidth limitations. Such applications incorporate echo cancellation, modeling human voice pitch, audio and sound signal analysis, sampling rate conversion, and timing synchronization. The proposed design builds upon Reconfigurable Interpolation Filter with Efficient Delay Profile using Improved Multiplication Algorithm. The proposed design is to be order-scalable, which will permit widely varying minimum clock period delay and this was achieved with the help of dadda multiplication algorithm which is faster compared to counterparts.. The implementation of proposed work has done on Xilinx 13.1 ISE. Outcomes of proposed work has analyzed and compared with previous with respect to minimum clock period(MCP).

Keywords- Reconfigurable Interpolation filter, Multiplication algorithm, digital filter, Delay efficient design.

I. INTRODUCTION

The process of digital signal interpolation is fundamental to signal processing. It is used in many contexts, most typically for conversion between sampling rates. This work explores efficient designs of digital interpolation systems for integer upsample factors.

Interpolation of a signal by an integer upsample factor can be accomplished by processing the signal, x[n], with the cascade of an expander and low-pass filter, as shown in Fig. 1.1. If the input signal x[n] has sampling frequency f, this results in the upsampled and interpolated output signal y[n] at the increased sampling frequency Lf. More complex interpolation systems can be designed as the cascade of multiple expanders and low-pass filters. A system containing two expanders and two low-pass filters is shown in Fig. 1.2.

If the parameters of the cascaded interpolator in Figure 1-2 are chosen correctly, namely $L_1 . L_2 = L$ and with appropriate choice of LPF_1 LPF_2, then this system will perform then perform equivalent interpolation to the system in Figure 1.1. In this case, assuming input sampling frequency f, the interpolated output signals y[n] and z[n] both have sampling frequencies Lf, and more specifically y[n] = z[n]. Thus, these two systems are distinct designs accomplishing the same interpolation, and can be compared in terms of computational efficiency.

Analog to-Digital Converters (ADCs) are enter design obstructs in present day communication frameworks. With enormous advances in CMOS creation innovation, most signal-processing functions are being moved to the digital space for a low power, ease, high return and profoundly reconfigurable execution.

The achievable inspecting g rate and determination of an ADC in a specific innovation is managed by the rate at which exchanged capacitor circuits in the ADC can be worked. To achieve higher sampling rates, many ADCs sampling at lower rates can be used in parallel. Exploiting parallelism in A/D conversion can be done either in the time or frequency domains.

Figure 1.1 Interpolation systems consisting of an expander and low-pass filter.

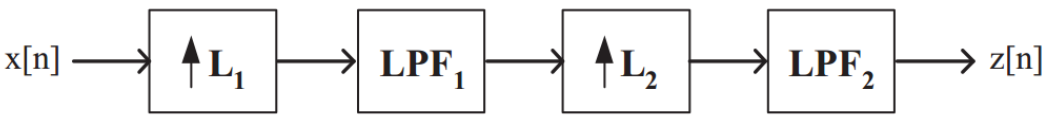

Figure 1.2 Interpolation systems consisting of the cascade of two expanders and two low-pass filters.

SDRs require filters of different bandwidths or a tunable filter to process the many different bandwidth signals. A filter can be implemented on reconfigurable hardware, e.g., FPGA that enables a complete reconfiguration for different air interfaces, or as a common filter that is parameterizable for all the required channel filtering. A reconfigurable

filter is an easy solution, obtainable simply by reconfiguring the FPGA with filter functionality. A parameterizable filter needs sophisticated solutions and the conventional FIR filter designs are not well suited. This is because the filter's length, i.e., number of taps varies inversely with the fractional bandwidth fBW/fs, and thus the required computational resources (such as data registers, multipliers, and adders) are different for filters with different lengths. Implementation considerations favor filters that are implemented with a fixed number of multipliers rather than with a varying number of multipliers application of multirate filters can provide a parametric filter solution by changing the rate factors. a variable bandwidth FIR filter architecture with fixed computational resources. His solution used arbitrary interpolators at the input and output of a fixed bandwidth filter, and by changing the input and output rate by the arbitrary interpolator, the required bandwidth is achieved. The variable bandwidth filters can also be realized by frequency masking filters, tunable IIR filters, low-pass to low- pass transformation tunable FIR filters, and a number of other solutions. Many of these options have a constant computational complexity but none preserves linear phase. Harris also presented a selectable bandwidth filter by using a pair of M-path analysis and synthesis filters. The bandwidth is changed by masking, enabling or disabling, the connections between the analysis and the synthesis filter.

II. DIGITAL FILTERS

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 in figure 2.1 illustrates the basic idea.

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.

Figure 2.1 Basic idea of a filter.

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. The analog input signal must first be sampled and digitized using an ADC (analog to digital converter). The resulting binary numbers, representing successive sampled values of the input signal, are transferred to the processor, which carries out numerical calculations on them. These calculations typically involve multiplying the input values by constants and adding the products together. If necessary, the results of these calculations, which now represent sampled values of the filtered signal, are output through a DAC (digital to analog converter) to convert the signal back to analog form. In a digital filter, the signal is represented by a sequence of numbers, rather than a voltage or current. The figure 2.2 shows the basic setup of such a system.

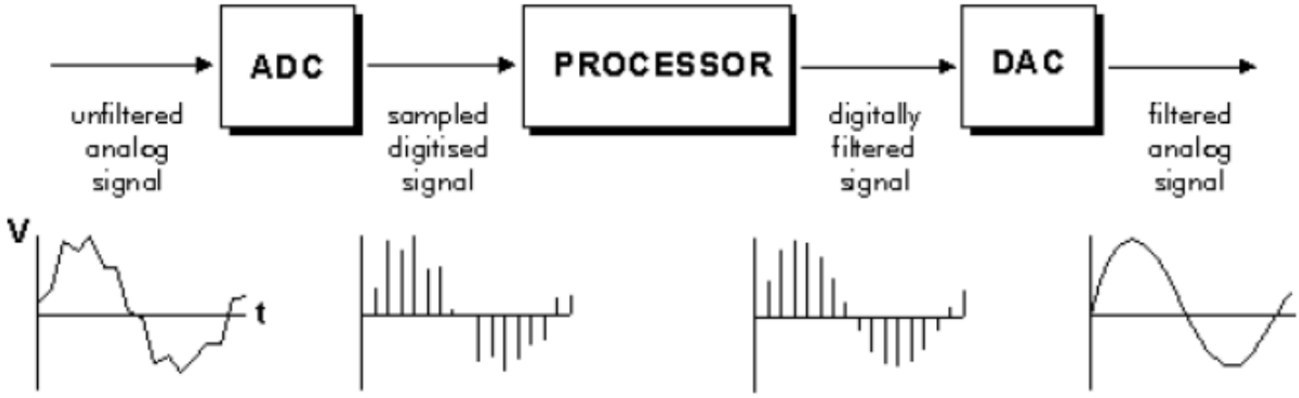

Figure 2.2 Basic Setup of a digital filter.

III. PROPOSED ARCHITECTURE

A variety of filter designs are considered for use in interpolation systems. The broadest distinction lies between FIR and IIR filter designs, which impact the use of polyphase implementations, and hence the computational cost of a system. The class of FIR filters is further divided into linear- phase and minimum-phase filters. In Proposed work a delay efficient reconfigurable interpolation filter has been implemented using Xilinx ISE.

Figure 3.1 demonstrated the Top Module of Architecture of Filter Length 64. To improve delay an improved multiplication algorithm has been used in this work. There are for sub models are there in proposed work CSU , AU, VGU, as demonstrated in figure 3.2 RTL Schematic. coefficient selection unit (CSU), ii) input-vector generation unit (VGU), and iii) arithmetic-unit. The CSU is comprised of N number of J:1 MUXes or N number of ROM LUTs of depth J words each, where N is the filter length and J is the number of interpolation filters of different coefficient vector to be realized in the reconfigurable architecture. To avoid longer critical path delay, MUX-based CSU is used; otherwise the ROM-based CSU is preferred. The required coefficient-vector of a particular interpolation filter is selected in one cycle from the CSU.

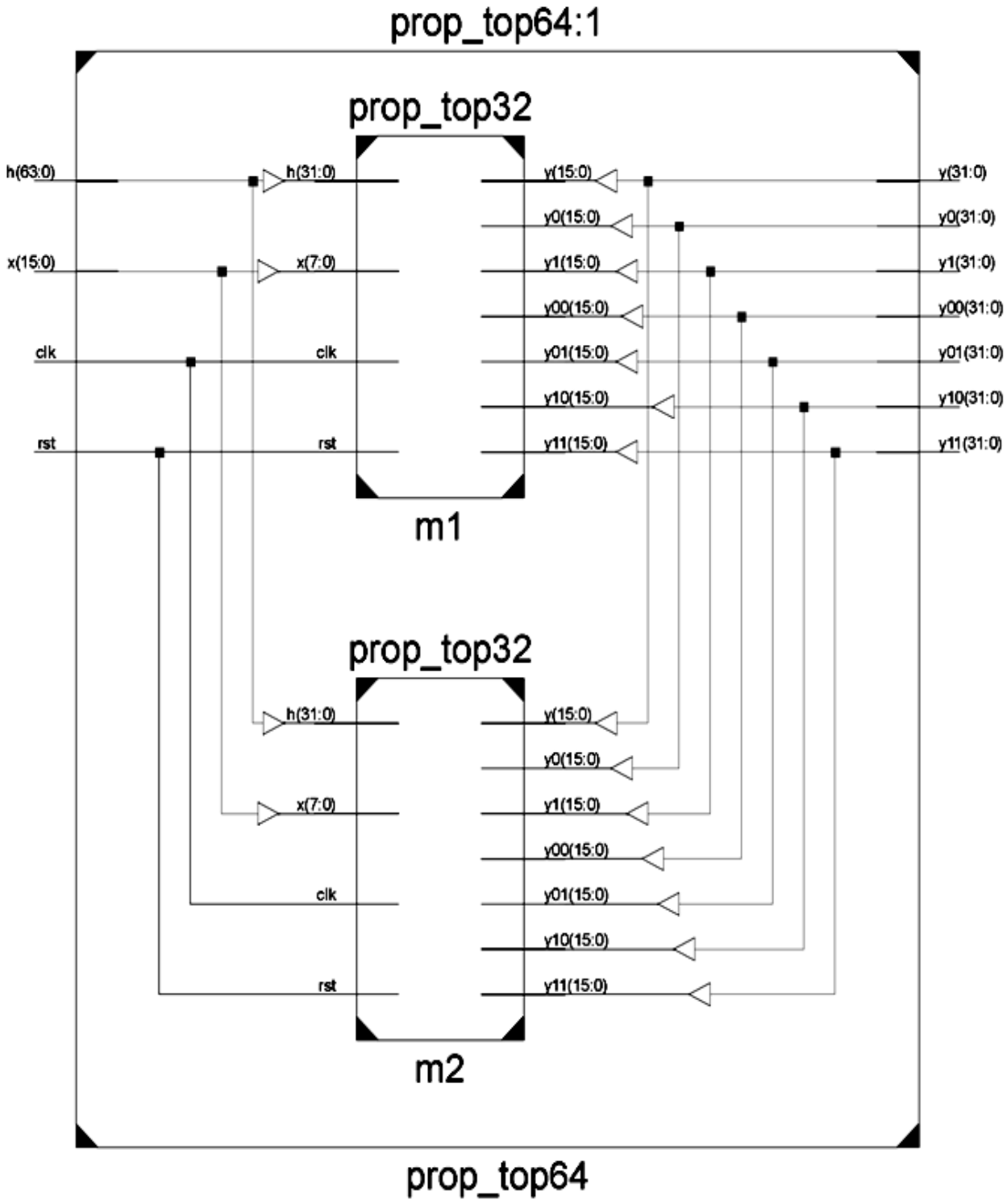

Figure 3.1 RTL Schematic of Top Module of Architecture of Filter Length 64

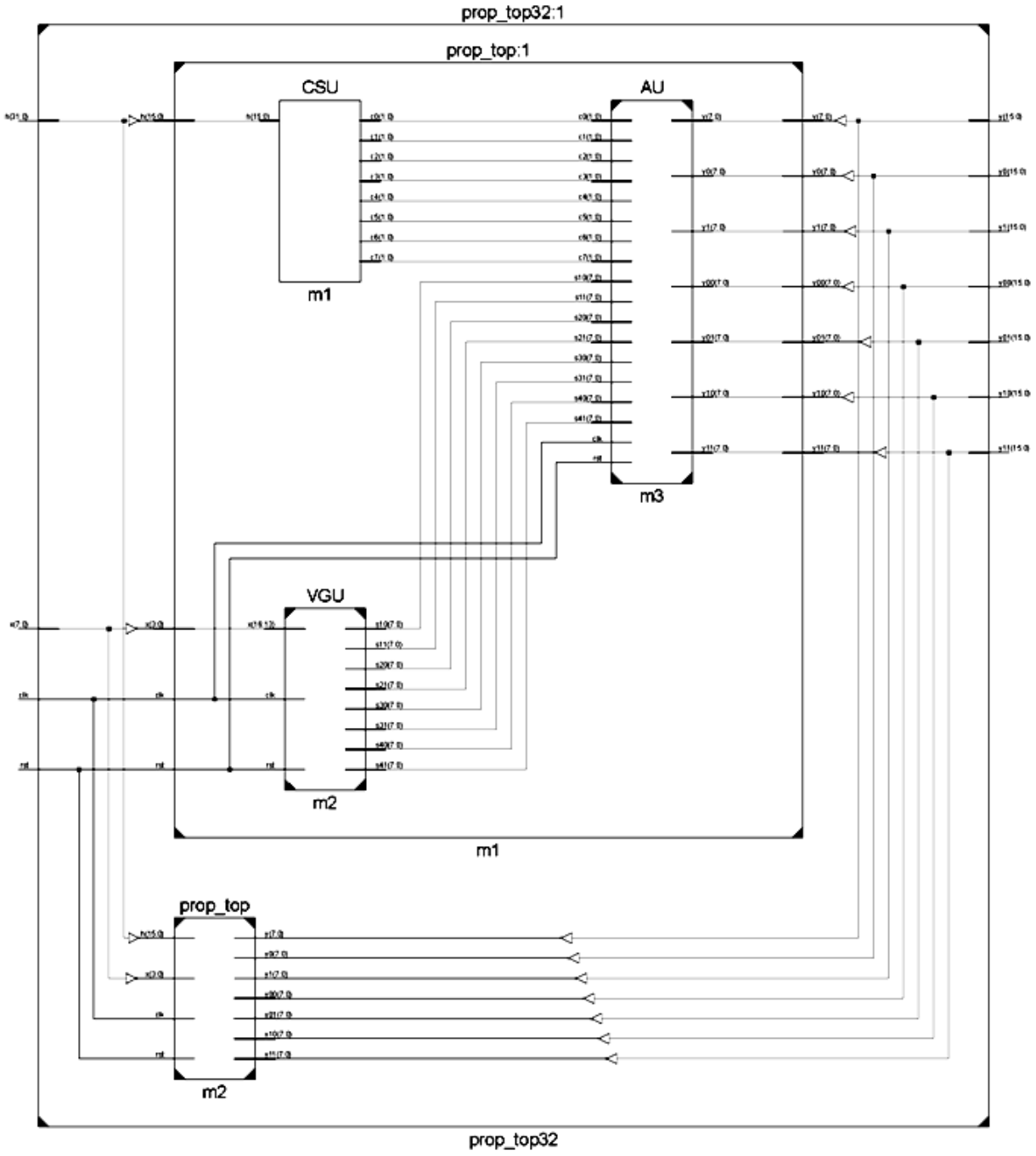

Figure 3.2 RTL Schematic of Sub Modules of Architecture of Filter Length 32.

IV. SYNTHESIS RESULTS

Proposed work has been implemented and simulated using Xilinx 13.1 ISE the performance and implementation results of proposed work are given in figure 4.2 Device utilization summary. A comparative analysis of proposed work with existing work has given in table 1 Synthesis Results of Proposed and Previous Architectures. The performance of proposed work is better than the existing work.

Figure 4.1 has give the Top Module of the Architecture of the propose system.

Table 1: Synthesis Results of Proposed and Previous Architectures.

Structures	Filter Length	Minimum Clock Period (MCP)	Area
Previous [1]	32	1.38 ns	230512.68 um^2
Proposed	32	1.153ns	158 Slice Registers 352 Slice LUTs
Proposed	64	1.153ns	316 Slice Registers 704 Slice LUTs

Fig. 4.3 shows the graphical comparison if Minimum Clock Period (MCP) with previous work. Figure 4.4 has given Minimum Clock Period (MCP) of Proposed Architecture of Filter Length 32. Minimum Clock Period (MCP) of Proposed Architecture of Filter Length 64. Figure 4.5 Minimum Clock Period (MCP) of Proposed Architecture of Filter Length 64.

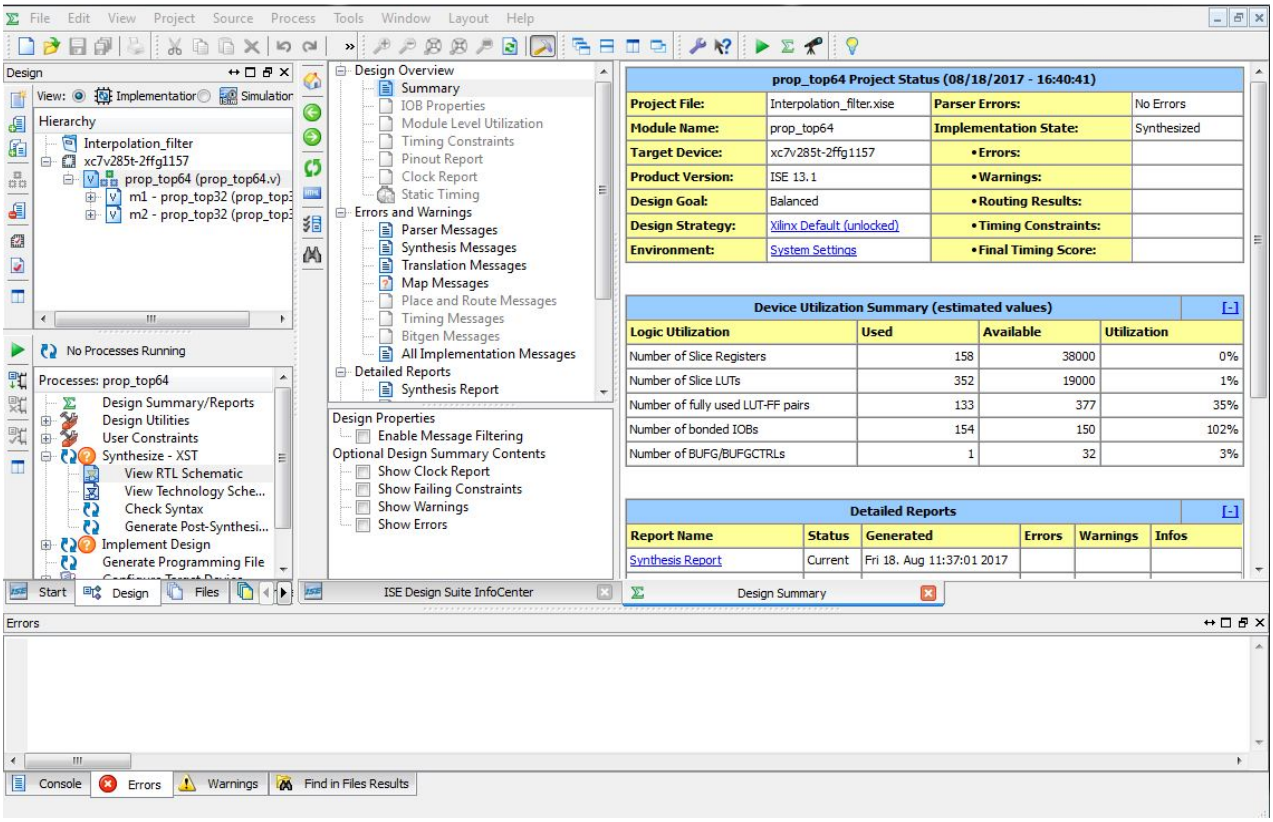

Figure 4.1 Top Module of the Architecture.

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	158	357600	0%
Number of Slice LUTs	352	178800	0%
Number of fully used LUT-FF pairs	133	377	35%
Number of bonded IOBs	154	600	25%
Number of BUFG/BUFGCTRLs	1	32	3%

Figure 4.2 Device Utilization of Proposed Architecture of Filter Length 32.

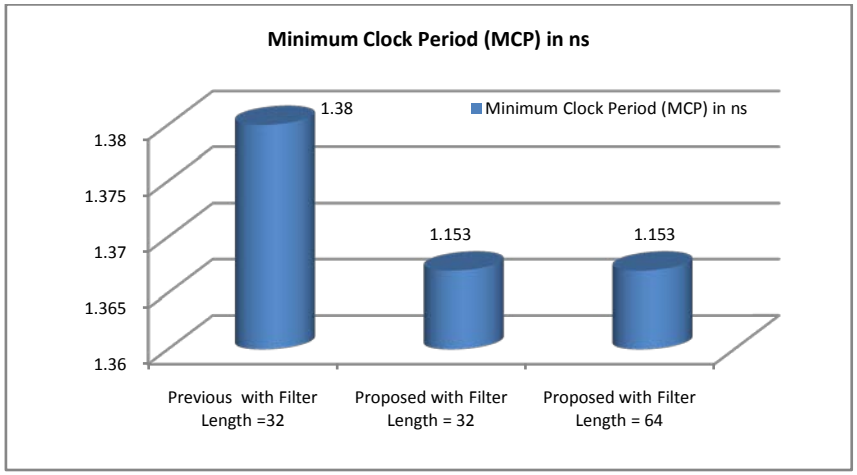

Figure 4.3 Comparison of MCP.

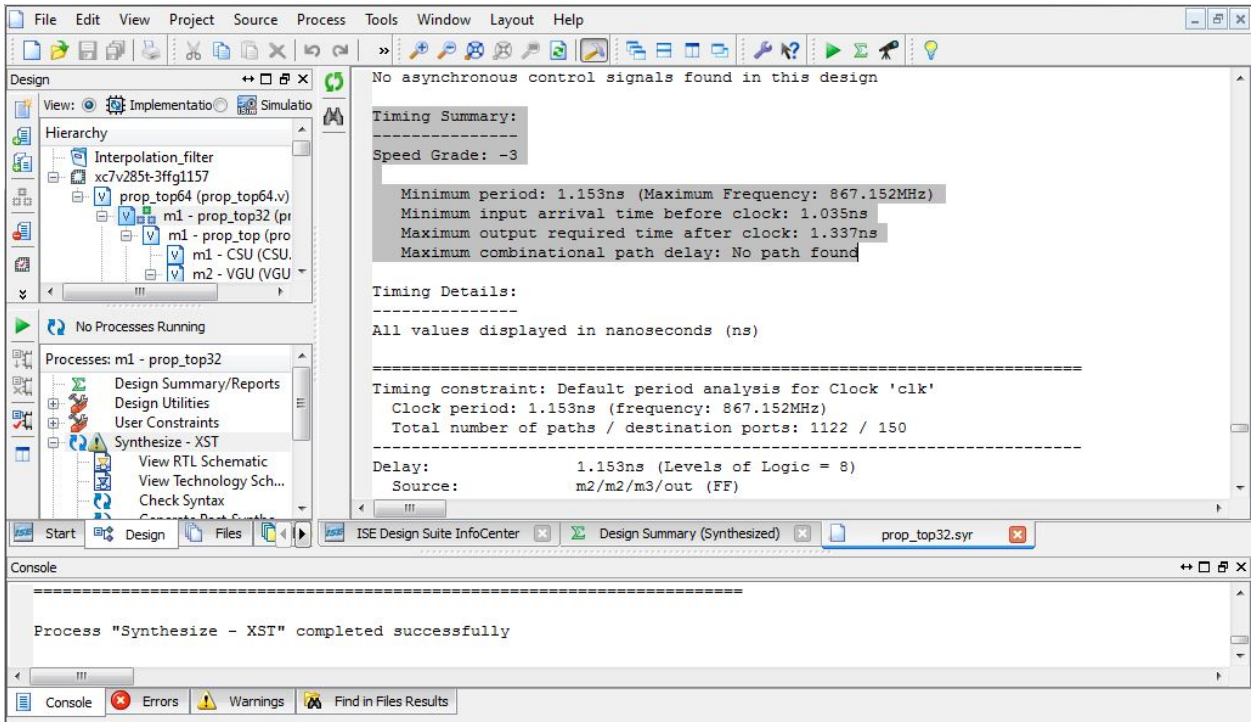

Figure 4.4 Minimum Clock Period (MCP) of Proposed Architecture of Filter Length 32.

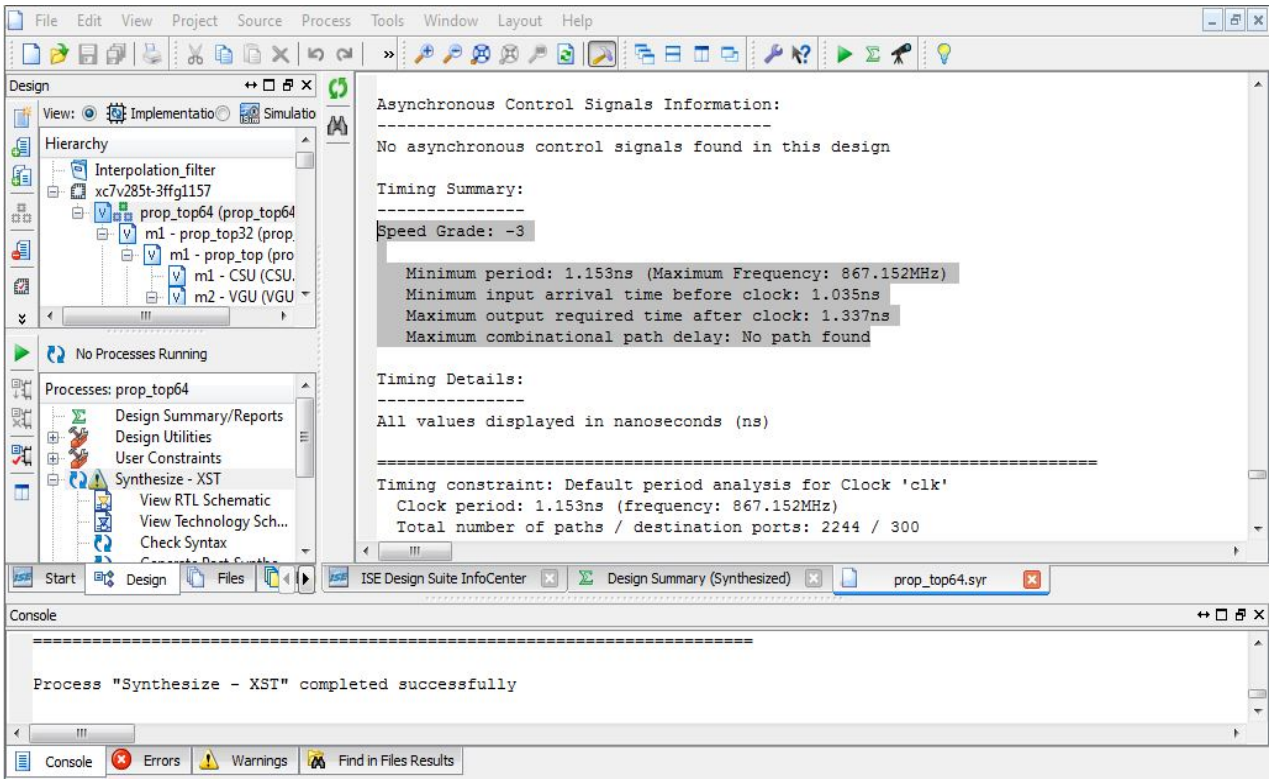

Figure 4.5 Minimum Clock Period (MCP) of Proposed Architecture of Filter Length 64

V. CONCLUSION

The fundamental goal of designing a digital filter is to design a filter that can reconstruct an interpolated continuous input signal and delay the reconstructed signal by a desired fractional delay. A digital reconfigurable interpolation filter is used in many modern digital signal processing applications, such as echo cancellation, modeling human voice pitch, musical signal analysis, and timing synchronization. The proposed filter scales in filter order to permit widely varying fractional delay values with high accuracy. This prototype was compared to a targeted high performance implementation platform using a Xilinx FPGA paired with a hard processing core via a high-speed bus. As part of the future work of this project, the software/hardware hybrid filter architecture can be implemented using a Xilinx FPGA with an integrated hard

processor or an embedded processor with a high-speed bus.

REFERENCES

[1] B. K. Mohanty, "Novel Block-Formulation and Area-Delay-Efficient Reconfigurable Interpolation Filter Architecture for Multi-Standard SDR Applications," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 62, no. 1, pp. 283-291, Jan. 2015.

[2] C. M. Diniz, M. Beck Fonseca, E. Costa and S. Bampi, "Enhancing a HEVC interpolation filter hardware architecture with efficient adder compressors," 2015 IEEE 13th International New Circuits and Systems Conference (NEWCAS), Grenoble, 2015, pp. 1-4.

[3] X. Lian, W. Zhou, Z. Duan and R. Li, "An efficient interpolation filter VLSI architecture for HEVC standard," 2014 IEEE China Summit & International Conference on Signal and Information Processing (ChinaSIP), Xi'an, 2014, pp. 384-388.

[4] R. Mahesh and A. P. Vinod, "Reconfigurable Low Area Complexity Filter Bank Architecture Based on Frequency Response Masking for Nonuniform Channelization in Software Radio Receivers," in IEEE Transactions on Aerospace and Electronic Systems, vol. 47, no. 2, pp. 1241-1255, April 2011.

[5] R. Mahesh and A. P. Vinod, "New Reconfigurable Architectures for Implementing FIR Filters With Low Complexity," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 29, no. 2, pp. 275-288, Feb. 2010.

[6] R. Mahesh and A. P. Vinod, "Reconfigurable Frequency Response Masking Filters for Software Radio Channelization," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 55, no. 3, pp. 274-278, March 2008.

[7] K. H. Chen and T. D. Chiueh, "A Low-Power Digit-Based Reconfigurable FIR Filter," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 53, no. 8, pp. 617-621, Aug. 2006.

[8] Jongsun Park, Woopyo Jeong, H. Mahmoodi-Meimand, Yongtao Wang, H. Choo and K. Roy, "Computation sharing programmable FIR filter for low-power and high-performance applications," in IEEE Journal of Solid-State Circuits, vol. 39, no. 2, pp. 348-357, Feb. 2004.

[9] P.P.Vaidyanathan, MuItirate Systems and Filter Banks. Englewood Cliffs, NJ, USA: Prentice-Hall Signal Processing Series, 1993.

[10] N. A. Sheikholeslami andP. Kabal, “Generalized raised cosine filters,” IEEE Trans. Commun., vol. 47, no. 7, pp. 989-997, 1999.