

A Parallel Area Delay Efficient Interpolation Filter Architecture

^[1]Anusha Ajayan, ^[2]Rafeekha M J

^[1] PG Student [VLSI & ES] ^[2] Assistant professor,

Department of ECE, TKM Institute of Technology, Kollam

^[1] anushaajayan899@gmail.com ^[2] rafeekhauvais2@gmail.com

Abstract:-- Interpolators are widely used in digital signal processing to increase the sampling rate digitally. A multi-standard Software Defined Radio (SDR) system involves interpolation with different filter coefficients, filter length and up-sampling factors to meet the stringent frequency specification. An SDR receiver consumes huge amount of resource when these interpolators are implemented individually in a hardware circuit. A reconfigurable Finite Impulse Response (FIR) interpolation filter is suitable for a resource and power constrained multi-standard SDR receiver. Now-a-days interpolation filter architecture with a few multipliers or without any multipliers are available. Area complexity, irregular dataflow and low hardware utilization efficiency are the major disadvantages of these architectures. In this work, a new parallel multiplier based reconfigurable structure is derived for interpolation filter. Elimination of redundancy and producing multiple outputs without reconfiguration are the features of this architecture. To validate the design, code can be developed using VHDL in Xilinx ISE Design Suite 13.2 and to be simulated in ModelSim SE 6.3f. The Xilinx synthesized result shows that, this architecture has less area, delay and Area-Delay Product (ADP) compared to the other existing architectures.

Index Terms—FIR filters, Interpolation, Software Defined Radio.

I. INTRODUCTION

The key requirement in designing any communication or information bearing device is its compactness. The fundamental the idea that behind the Software Defined Radio technology is that to replace all the analog processing system with the digital processing system so as to get the advantage of flexibility. In a multistandard SDR system it has to works with different communication specifications. For a multistandard communication having different specifications it requires separate filters, modulators, demodulators etc. If separate hardwares are used for each specification it will consume huge area and as a result of this power consumption of the entire system increases drastically. So reconfiguration is an essential factor. Here in this paper reconfigurable interpolation filter architecture is presented.

The process of upsampling the baseband signal followed by filtering of the signal is termed as interpolation. Whenever there is a need of changing from one sampling rate to another, Interpolation is very much essential. It is also called as upsampling or zero stuffing that means, inserting zero-valued samples between original input samples inorder to increase the sampling rate. Using a sample rate converter, the base band signal will be interfered with undesired signals. As a result of this distorted signals are produced at the output

side. These undesired components are removed through filtering. Distortions may arise due to upsampling. Distortions vigorously increase during the upcoming stages. Filtering the up sampled signal will remove distortions.

II. RELATED WORKS

During the last decade, several multiplier and multiplier less designs have been suggested for efficient hardware realization of reconfigurable FIR filters and filter-banks for SDR channelization. But we do not find much work on reconfigurable interpolation filter architecture except a few. A single rate (fixed up-sampling factor) FIR interpolation filter can be implemented using a FIR structure. This could be the reason for non availability of any specific design in the literature for reconfigurable FIR interpolation filter. However, a single rate interpolation filter operate at P times higher sampling rate than the input sampling frequency and requires N filter parameters to compute each output, where P is the upsampling factor. On the other hand, a poly-phase based multirate interpolation filter operates at the input sampling rate and compute P outputs using P sub-filters each having N/P filter parameters. Therefore, a multi-rate interpolation filter structure is more hardware efficient than the single rate interpolation filter structure. The existing reconfigurable FIR filter structures are efficient for channelizer, but they do not offer an efficient computing structure for reconfigurable interpolation filter.

A few multipliers-less designs are proposed for interpolation filter. Area complexity can be reduced by using the symmetric property of PSF and a LUT decomposition scheme. In addition to this, LUT sharing of in-phase and quadrature-phase filters are used to save LUT words which offer a significant saving in area complexity of the interpolation filter. Both these designs cannot be reconfigured for up-sampling factor other than 4, and for different filter specifications. A distributed arithmetic (DA)-based reconfigurable FIR interpolation filter architecture is proposed in [7]. The DA-LUT stores partial results of all the sub-filter outputs of interpolation filter with three different interpolation factors. As a result of this, the structure requires a large size DA-LUT which is not suitable for single chip realization. Recently, Hatai [3] have proposed a reconfigurable FIR interpolation filter design similar to using LUT-less DA technique to reduce the area complexity. Coefficient-vector of the desired interpolation filter are selected using an array of multiplexers. The structure uses AND-gates, multiplexer and adders to implement the DA-LUT and computes a sub-filter output of the interpolation filter in bit-serial manner. It involves less area than the previously proposed structures and supports base-band signal of low-sampling rates. Besides, the structure has a large overhead complexity (in terms of multiplexer and registers) for its reconfigurable feature.

III. EXISTING SYSTEM

A few multipliers-less designs are proposed for interpolation filter. Symmetric property of PSF and a LUT decomposition scheme are used first to reduce the area complexity of 1:4 interpolation filter. In addition to this, LUT sharing of in phase and quadrature-phase filters are used to save LUT words which offers a significant saving in area complexity of the interpolation filter. These architectures cannot be reconfigured for up-sampling factor other than 4. A Distributed Arithmetic (DA) based reconfigurable FIR interpolation filter architecture was then proposed. The DA-LUT stores partial results of all the sub-filter outputs of interpolation filter with three different interpolation factors.

Therefore, the structure requires a large size DA-LUT. Then proposed a reconfigurable FIR interpolation filter design similar using LUT-less DA [3] technique to reduce the area complexity. Here coefficient-vectors are selected using multiplexer arrays. The structure uses AND-gates, multiplexer and adders to implement the DA-LUT and computes a sub-filter output of the interpolation filter in bit-serial manner. It involves less area complexity while it supports base-band signal of low sampling factors. Besides,

the structure has a large overhead complexity for its reconfigurable feature.

IV. PROPOSED SYSTEM

Since, reuse of partial result favors parallel computation of interpolation filter outputs for different up-sampling factors, the data-selector unit can be avoided in the reconfigurable architecture without any extra cost. Overall, a parallel reconfigurable architecture can be designed using the partial result generation unit and the reconfigurable adder unit. Using the block-processing scheme the reconfigurable adder unit

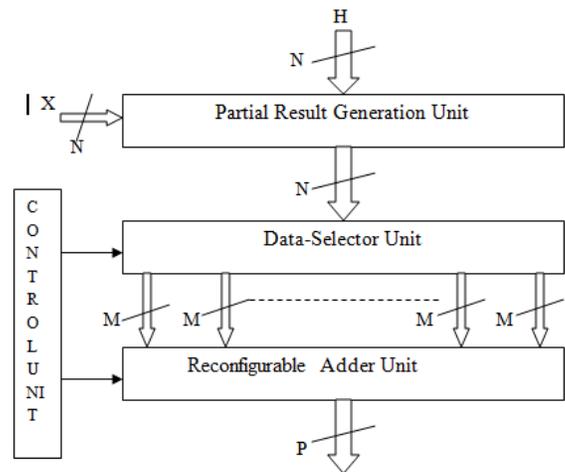


Fig.1. Existing system architecture

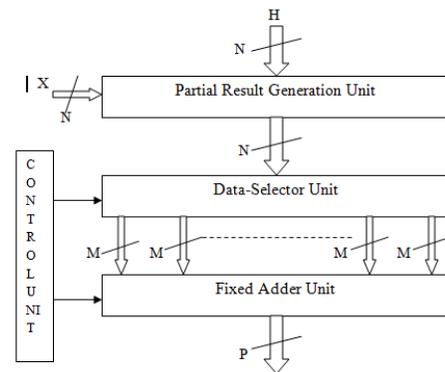


Fig.2. Proposed system

Can be replaced by a fixed adder-unit comprising of N adders. The overall architecture of the proposed system is shown figure.

A. Block Diagram

The basic block diagram for the reconfigurable interpolation filter is shown in the figure: 3 given below. The overall working flow is roughly described as follows. Based on the filter specification (interpolation factors), co-efficient are first generated by using the Filter Design and Analysis Tool (FDA Tool) in the MATLAB. Next the filter specifications are applied to the register arrays and produces the input vectors. At first these input vectors and coefficients are multiplied. Then the multiplied outputs are added together to get the final output.

B. Input Vector Generation Unit

The Vector Generation Unit (VGU) receives one input-block in each cycle and generates (N/P1) input-vectors of size (L/P1) each in parallel, where P1 is the smallest up-sampling factor from a set of q different up-sampling factors to be realized by the reconfigurable architecture. Internal structure of the VGU is shown in figure: 4. It is comprised of (N-1) registers. The VGU receives a block of input samples in every cycle and produces 8 data-vectors. The block of inputs is determined by using the block formulation method[1].

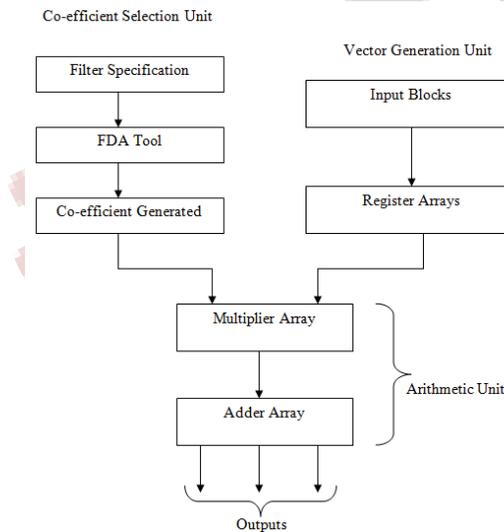


Fig.3. Basic block diagram

C. Co-efficient Generation Unit

First the filter specifications are given to the Filter Design Tool in the MATLAB. It produces the co-efficients. These coefficient are used directly into the VHDL coding. The Coefficient Generation Unit(CSU) is comprised of N number of J :1 MUXes or N number of ROM LUTs of depth J word 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 for J=4,

otherwise the ROM based CSU is preferred. The required coefficient-vector of a particular interpolation filter is selected in one cycle from the CSU.

D. Arithmetic unit

The structure of Arithmetic Unit (AU) is shown in figure: 5 having interpolation factors (IF₂; IF₄; IF₈) and for block size L=4, and filter length N=16. It is comprised of (N/P₁) Multiplier Units (MUs) and ((N/P₁)-1 = 7) Adder Units (ADU). Each Multiplier Unit receives an (LP₁) point input-vector from the VGU and a short P₁-point coefficient-vector C_m from the CSU, and calculates one partial filter output-vector (Z_{k,m}) of size (N=P₁).

The partial output-vectors (Z_{k,0}, Z_{k,4}), (Z_{k,1}, Z_{k,5}), (Z_{k,2}, Z_{k,6}) and (Z_{k,3}, Z_{k,7}) added in four separate ADUs (ADU₁, ADU₂, ADU₃, ADU₄) to compute filter output-blocks (Y⁰⁰_k, Y⁰¹_k, Y¹⁰_k, Y¹¹_k) of IF₈. For IF₄, the output-vectors (Y⁰⁰_k, Y⁰¹_k, Y¹⁰_k, Y¹¹_k) represents its partial filter output. The adders ADU₅ and ADU₆ adds the partial output-vectors. As a result the complete filter output vectors of IF₄ (Y⁰_k, Y¹_k). Similarly, the output vectors of IF₄ represent the partial filter outputs of IF₂. Then these output vectors are added in ADU₇ to get the output vector Y_k of IF₂.

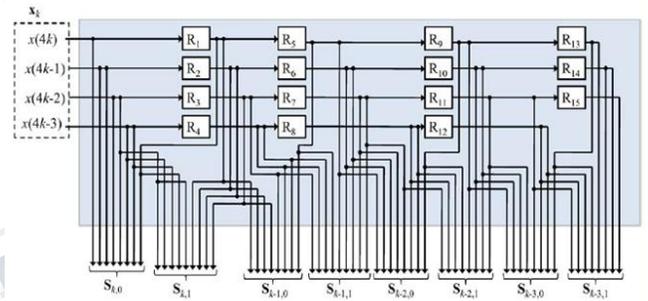


Fig.4. Vector Generation Unit

V. EXPERIMENTAL RESULTS

The three basic modules are synthesized using VHDL in Xilinx ISE Design. Then simulated using Modelsim SE 6.3f simulator.

A. Simulation of VGU

Here, first the input block is applied. The input block is declared as an array format. It can contain 4 inputs. Each input is 16 bits long. It consists of an array of delay elements. Here we take the delay element as D flip flop. Internally it is divided into 4 stages. The 4 set of output of delay elements (R1, R2, R3, R4), (R5, R6, R7, R8), (R9, R10, R11, R12) and (R13, R14, R15, R16) indicates four different stages such as stage 0, stage 1, stage 2 and stage 3 respectively. Each stage

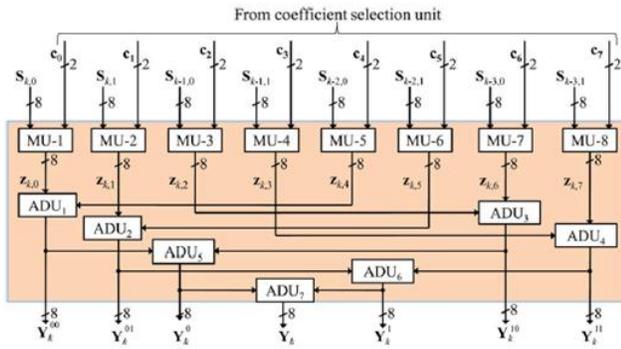


Fig.5. Arithmetic Unit

Consist of four 16 bits datas. The appropriate stages outputs are combined together to take the final output. The simulation results of the VGU are shown in the figure 6.

The co-efficients are generated by Filter Design and Analysis Tool. First open the MATLAB and type fda tool in the command window. Then select the create multirate filters icon. Select the filter type as interpolator, give the interpolation factor according to the specifications. Then set the sampling frequency. Co-efficient values can be obtained from the analysis menu in the menu bar. These can be directly used in the VHDL coding, by storing it in the LUT.

B. Simulation of AU

The AU is divided into Multiplier Array and Adder Array. The output vectors from the VGU and the coefficients from the CSU are given as the input of AU. First, the multiplier array part multiplies the inputs applied, then the adder unit adds the appropriate multiplier outputs. The simulation results of the AU is shown in the figure : 7.

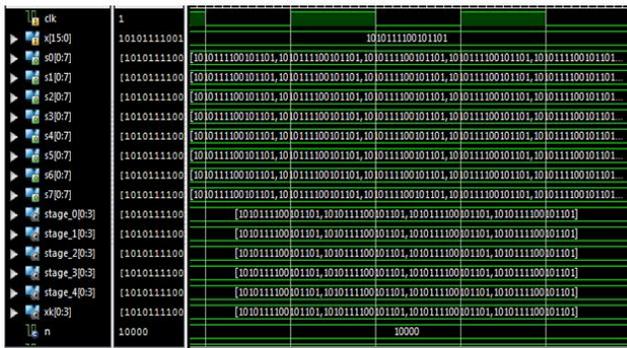


Fig.6. Simulation results of VGU.

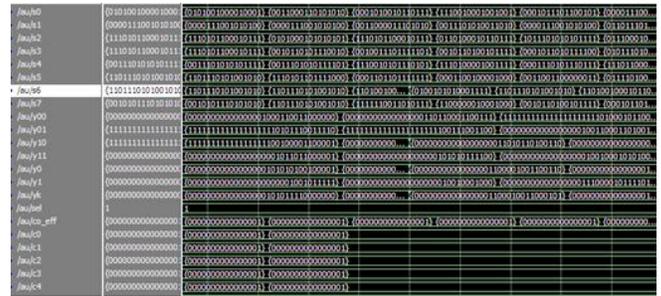


Fig.7. Simulation results of AU.

C. Final Simulation Results



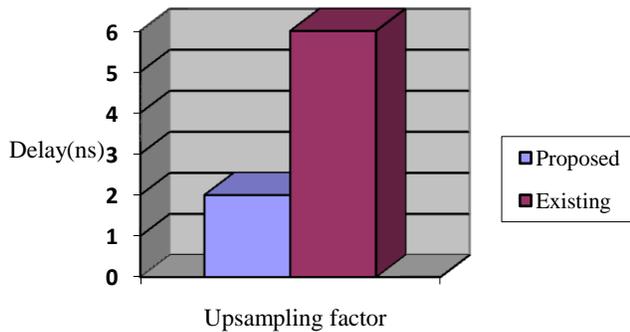


Fig.10. Delay comparison

VI. CONCLUSION

In this architecture, a new block formulation method is presented. By this architecture, the partial results are reused for parallel computation of filter outputs of different up-sampling factors. It does not require reconfiguration to compute filter outputs of a particular interpolation filter for different upsampling factors, and configured when there is a need to change the filter specification. In that case, a coefficient-vector of the desired filter is selected from the CSU and fed to the AU to perform the filter computation. The VGU and AU constitute the core of this structure and do not require any reconfiguration to change the filter computation. Therefore, the proposed architecture offers reconfigurability without using any overhead complexity unlike the existing reconfigurable architectures. It is always an advantage to realize the proposed architecture for the lowest up-sampling factor, and filter outputs of higher upsampling factors of a given set of up-sampling factors can be obtained in parallel without performing any extra computation. This filter outputs at multiple sampling frequency for an input sampling frequency is a unique feature of this architecture. The complexity of this architecture is independent of upsampling factor and it does not increase proportionately with the blocks-size. Therefore the area-delay efficiency of the proposed architecture is expected to be better for higher blocksizes. The entire architecture can be designed using VHDL language and synthesized in Xilinx ISE Design Suit 13.2 and simulated in ModelSim SE 6.3f.

REFERENCES

- [1] Basant Kumar Mohanty, Novel Block-Formulation and Area-Delay- Efficient Reconfigurable Interpolation Filter Architecture for Multi-Standard SDR Applications, IEEE Transactions On Circuits And Systems I: regular papers, volume:62,no.1,january 2015.
- [2] L. P. Usha, Naveen Kumar G. N, Design and implementation of Pulse- Shaping FIR Interpolation filter using BCSE Algorithm, International Journal of Engineering Research & Technology (IJERT)ISSN: 2278- 0181, volume:4,Issue:05,May-2015.
- [3] I. Hatai, I. Chakrabarti, and S. Banerjee, Reconfigurable architecture of RRC FIR interpolator for multi-standard digital up converter, in Proc. IEEE 27th Int. Symp. Parallel Distrib. Processing Workshops PhD Forum, 2013, pp.247251.
- [4] R. Mahesh and A. P. Vinod, Reconfigurable low area complexity filter bank architecture based on frequency response masking for nonuniform Channelization in software defined radio, IEEE,Trans.Aerosp. Electron. Syst, volume:47,no.2,pp.12411254,Apr.2011.
- [5] R. Mahesh and A. P. Vinod, New reconfigurable architectures for implementing FIR filters with low complexity, IEEE Trans.Comput-Aided Design Integr. Circuits Syst, volume:29,no.2,pp.275-288,Feb.2010.
- [6] R. Mahesh and A. P. Vinod, Reconfigurable frequency response masking filters for software radio channelization, IEEE Trans.Circuits Syst.II, Exp.Briefs, volume:55, no.3,pp.274278,Mar.2008.
- [7] G. C. Cardarilli, A. D. Re, M. Re, and L. Simone, Optimized QPSK modulator for DVB-S application, in Proc. IEEE Int. Symp. Circuits Syst.(ISCAS), 2006,pp.15711577.
- [8] J. Park, W. Jeong, H. Mahmoodi-Meimand, Y. Wang, H. Choo, and K. Roy, Computation sharing programmable FIR filter for low-power and high-performance applications, IEEE Journal Solid State Circuits, volume:39, no.2, pp.348357, Feb.2004.
- [9] T. Zhangwen, J. Zhang, and H. Min, A high-speed, programmable, CSD coefficient FIR filter, IEEE Trans.Consum.Electronics, volume: 48, no.4,pp.834837, Nov.2002.
- [10] N. Sankarayya, K. Roy, and D. Bhattacharya, Algorithms for lowpower high speed FIR filter realization using differential coefficients, IEEE Trans.CircuitsSyst.II, volume:44,pp.488497, June 1997.
- [11] R. I. Hartley, Subexpression sharing in filters using canonic signed digit multipliers, IEEE Trans.Circuits Syst.II, volume:43,no.10,pp.677688,Oct.1996.

- [12] M. Potkonjak, M. Srivastava, and A. P. Chandrakasan, Multiple constant multiplications: Efficient and versatile framework and algorithms for exploring common subexpression elimination, IEEE Trans.Computer-Aided Design, volume.15,pp.151165,Feb.1996.

