ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 6, Issue 11, November 2017

Reconfigurable Interpolation Filter Architecture Design

Satheesh Kumar S[1], Venugopal P[2], Srinivas Koppu[3], Rajeev Pankaj N[4], Rajesh Kaluri[4], Madhusudhana Rao[5]

Assistant Professor (Senior), School of Electronics Engineering, VIT University, Vellore, Tamilnadu, India[1]

Assistant Professor (Senior), School of Electronics Engineering, VIT University, Vellore, Tamilnadu, India[2]

Assistant Professor (Senior), School of Information Technology and Engineering, VIT University, Tamilnadu, India[3,5]

Assistant Professor (Senior), School of Electronics Engineering, VIT University, Vellore, Tamilnadu, India[4]

Lecture, Department of Information Technology, Higher college of Technology, Muscat, Oman[5]

ABSTRACT: In VLSI implementation the demand for high speed and delay efficient interpolation filters are required for many applications In a poly-phase based interpolation filter, the input matrix size and coefficient matrix size is given by (P×M), where P is the up-sampling factor and M =N/P and, N is the filter length. N is kept constant, so that the input-matrix and the coefficient-matrix resizes only when P changes. In this paper, an analysis of interpolation filter output computation for multiple up-sampling factors is made and it identifies the redundant partial results and reuses it there by saving the area. With the help of block formulation mechanism the partial results can be reused for different up-sampling factors. Using the block formulation mechanism, a parallel multiplier-based reconfigurable architecture is derived for interpolation filter. Using this architecture it is possible to compute the filter outputs in parallel for multiple up-sampling factors without using any extra resources. Besides, the proposed structure has significantly less register complexity. The logic synthesis of this architecture is performed using RC compiler in Cadence for gpdk90nm technology.

KEYWORDS: Full parallel filter, Interpolation filter, up-sampling factor, reconfigurable.

I.INTRODUCTION

In digital signal processing (DSP) systems to increase the sampling rate digitally interpolators are used. It comprises an up-sampler and an anti-imaging (Interpolation) filter. Sampling rate of base-band signal can be changed using up sampler. So that interpolation filter suppress the undesired interference effect resulted [1]. Due to high inter-symbol interference (ISI) rejection ratio and high bandwidth limitation Pulse shaping filters (PSFs) are used as interpolation filter. In a base band signal, Interpolation filter have different constraints.

Multi-standard SDR applications involve interpolators with different filter coefficients, filter-lengths and up-sampling factors to meet the specifications of different communication standards [3]. For example: Universal Mobile Telecommunication Standard (UMTS) uses interpolators with interpolation factors (4, 8, and 16), and filter lengths (25, 49, 97) respectively when the interpolators are implemented individually in a hardwired circuit. Multi-standard SDR receiver support different up-sampling factors as well as filter specifications. So reconfigurable finite impulse response (FIR) filters are used to meet these constraints since reconfigurable FIR filters have deficient hardware realizations and filter banks for SDR channels. Single rate FIR interpolation filter can be implemented using FIR structure. So that non availability of any specific design for reconfigurable interpolation filter is not possible. Single rate interpolation filter requires P times higher the sampling rate than the input sampling frequency and N filter parameters to compute the output. Where, P is up sampling factor. The previous version is used for channelizer not for effective computing structure.

An interpolation filter involves an input matrix of size (P×M) where M=N/P; N=filter length; P=up sampling factor. Depends on up sampling factor, number of sub filters required for reconfigurable interpolation filter for full parallel

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 6, Issue 11, November 2017

structure. Because of lowest sampling rates in full parallel architecture, number of sub samples remained unused. Similarly for highest sampling rate, numbers of sub samples are partially utilized to overcome this problem I am implementing folded architecture instead of full parallel architecture.

II.BLOCK FORMULATION OF INTERPOLATION

Consider a FIR interpolation filter having up-sampling factor p. Let it processes the input block having l input samples. It will generate the p filter outputs having block size l for every cycle. The k^{th} cycle filter output computation can be written as,

$$\begin{bmatrix} y^0{}_k \\ y^1{}_k \\ \cdot \\ y^{L-1}{}_k \end{bmatrix} = \begin{bmatrix} s^0{}_{k,0} & \cdot & \cdot & s^0{}_{k,M-1} \\ s^1{}_{k,0} & \cdot & \cdot & s^1{}_{k,M-1} \\ \cdot & \cdot & \cdot & \cdot \\ s^{L-1}{}_{k,0} & \cdot & \cdot & s^{L-1}{}_{k,M-1} \end{bmatrix} \otimes \begin{bmatrix} c0 & \cdot & \cdot & cM-1 \\ c0 & \cdot & \cdot & cM-1 \\ \cdot & \cdot & \cdot & \cdot \\ c0 & \cdot & \cdot & cM-1 \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} y^0{}_k \\ y^1{}_k \\ \cdot \\ y^{L-1}{}_k \end{bmatrix} = \begin{bmatrix} s^0{}_{k,0} & \cdot & \cdot & s^0{}_{k,M-1} \\ s^1{}_{k,0} & \cdot & \cdot & s^1{}_{k,M-1} \\ \cdot & \cdot & \cdot & \cdot \\ s^{L-1}{}_{k,0} & \cdot & \cdot & s^{L-1}{}_{k,M-1} \end{bmatrix} \otimes \begin{bmatrix} c0 & \cdot & \cdot & cM-1 \\ c0 & \cdot & \cdot & cM-1 \\ \cdot & \cdot & \cdot & \cdot \\ c0 & \cdot & \cdot & cM-1 \end{bmatrix} \quad (2)$$

Where,

$$S^l_{k,i} = [x(kl - l - pi), \ldots, x(kl - l - pi - p + 1)]^T \quad (2a)$$

$$Y^l_k = [y(pkl - lp), \ldots, y(pkl - lp - 1)]^T \quad (2b)$$

A. Analysis of Interpolation Filter Block Formulation

Consider filter length, n=8 and l=4, For p=2, filter output can be written as,

$$\begin{bmatrix} y^0{}_k \\ y^1{}_k \\ y^2{}_k \\ y^3{}_k \end{bmatrix} = \begin{bmatrix} s^0{}_{k,0} & s^0{}_{k,1} & s^0{}_{k,2} & s^0{}_{k,3} \\ s^1{}_{k,0} & s^1{}_{k,1} & s^1{}_{k,2} & s^1{}_{k,3} \\ s^2{}_{k,0} & s^2{}_{k,1} & s^2{}_{k,2} & s^2{}_{k,3} \\ s^3{}_{k,0} & s^3{}_{k,1} & s^3{}_{k,2} & s^3{}_{k,3} \end{bmatrix} \otimes \begin{bmatrix} c0 & c1 & c2 & c3 \\ c0 & c1 & c2 & c3 \\ c0 & c1 & c2 & c3 \\ c0 & c1 & c2 & c3 \end{bmatrix} \quad (3)$$

$$c_i = [h(2i), h(2i+1)] \quad (4)$$

For p=4, filter output can be written as,

$$\begin{bmatrix} y^0{}_k \\ y^1{}_k \\ y^0{}_k \\ y^0{}_k \end{bmatrix} = \begin{bmatrix} r^0{}_{k,0} & r^0{}_{k,1} \\ r^1{}_{k,0} & r^1{}_{k,1} \\ r^2{}_{k,0} & r^2{}_{k,1} \\ r^3{}_{k,0} & r^3{}_{k,1} \end{bmatrix} \otimes \begin{bmatrix} d_0 & d_1 \\ d_0 & d_1 \\ d_0 & d_1 \\ d_0 & d_1 \end{bmatrix} \quad (5)$$

Where,

$$r^l{}_{k,i} = [s^l_{k,2i}, s^l_{k,2i+1}]^t \quad (6a)$$

$$d_i = [c_{2i}, c_{2i+1}]^t \quad (6b)$$

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 6, Issue 11, November 2017

Rewriting the output in split form as,

$$y_k = [y^{l0}, y^{l1}]^t \tag{7a}$$

$$y^{l0} = [y(16k - 4l), y(16k - 4l-1)]^t \tag{7b}$$

$$y^{l0} = [y(16k - 4l-2), y(16k - 4l -3)]^t \qquad \text{For } l = 0, 1, 2, 3 \tag{7c}$$

Substituting (6) and (7) in (5) we get,

$$\begin{bmatrix} y^{00}_k \\ y^{01}_k \\ y^{10}_k \\ y^{11}_k \\ y^{20}_k \\ y^{21}_k \\ y^{30}_k \\ y^{31}_k \end{bmatrix} = \begin{bmatrix} s^0_{k,0} & s^0_{k,2} \\ s^0_{k,1} & s^0_{k,3} \\ s^1_{k,0} & s^1_{k,2} \\ s^1_{k,1} & s^1_{k,3} \\ s^2_{k,0} & s^2_{k,2} \\ s^2_{k,1} & s^2_{k,3} \\ s^3_{k,0} & s^3_{k,2} \\ s^3_{k,1} & s^3_{k,3} \end{bmatrix} \otimes \begin{bmatrix} c0 & c2 \\ c1 & c3 \\ c0 & c2 \\ c1 & c3 \\ c0 & c2 \\ c1 & c3 \\ c0 & c2 \\ c1 & c3 \end{bmatrix} \tag{8}$$

By observing (3) and (8), it is clear that for p=2 and p=4, interpolation filter is using same input vectors ($s^l_{k,i}$) and coefficient vectors (c_i) to compute the filter output (y_k). So the partial results of (8) form the redundant set. We can reuse the partial results of (3) to get the partial results of (8). However the input vectors and coefficients appear to be at different locations but they appear in specific pattern. To reuse the partial results, modified block formulation is used.

B. Reconfigurable Interpolation Filter using Modified Block Formulation

In modified block formulation, we use a input vector

$$s^l_{k,2j+i} \text{ as } v^l_{k-j,i} = s^l_{k,2j+i} \tag{9}$$

Using (9) we can write input vectors as,

$$V^l_{k,0} = s^l_{k,0} \qquad v^l_{k,1} = s^l_{k,1} \qquad v^l_{k-1,0} = s^l_{k,2} \qquad v^l_{k-1,1} = s^l_{k,3}$$

Substituting (9) in (6) we will get,

$$\begin{bmatrix} y^0_k \\ y^1_k \\ y^2_k \\ y^3_k \end{bmatrix} = \begin{bmatrix} v^0_{k,0} & v^0_{k-1,0} \\ v^1_{k,0} & v^1_{k-1,0} \\ v^2_{k,0} & v^2_{k-1,0} \\ v^3_{k,0} & v^3_{k-1,0} \end{bmatrix} \otimes \begin{bmatrix} c0 & c2 \\ c0 & c2 \\ c0 & c2 \\ c0 & c2 \end{bmatrix} + \begin{bmatrix} v^0_{k,1} & v^0_{k-1,1} \\ v^1_{k,1} & v^1_{k-1,1} \\ v^2_{k,1} & v^2_{k-1,1} \\ v^3_{k,1} & v^3_{k-1,1} \end{bmatrix} \otimes \begin{bmatrix} c1 & c3 \\ c1 & c3 \\ c1 & c3 \\ c1 & c3 \end{bmatrix} \tag{10}$$

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 6, Issue 11, November 2017

Similarly, substituting (9) in (8) we will get,

$$\begin{bmatrix} y^{00}{}_k \\ y^{10}{}_k \\ y^{20}{}_k \\ y^{30}{}_k \end{bmatrix} = \begin{bmatrix} v^0{}_{k,0} & v^0{}_{k-1,0} \\ v^1{}_{k,0} & v^1{}_{k-1,0} \\ v^2{}_{k,0} & v^2{}_{k-1,0} \\ v^3{}_{k,0} & v^3{}_{k-1,0} \end{bmatrix} \otimes \begin{bmatrix} c0 & c2 \\ c0 & c2 \\ c0 & c2 \\ c0 & c2 \end{bmatrix} \tag{11a}$$

$$\begin{bmatrix} y^{01}{}_k \\ y^{11}{}_k \\ y^{21}{}_k \\ y^{31}{}_k \end{bmatrix} = \begin{bmatrix} v^0{}_{k,1} & v^0{}_{k-1,1} \\ v^1{}_{k,1} & v^1{}_{k-1,1} \\ v^2{}_{k,1} & v^2{}_{k-1,1} \\ v^3{}_{k,1} & v^3{}_{k-1,1} \end{bmatrix} \otimes \begin{bmatrix} c1 & c3 \\ c1 & c3 \\ c1 & c3 \\ c1 & c3 \end{bmatrix} \tag{11b}$$

So, using modified block formulation the partial results of an interpolation filter for different up-sampling factors can be reused. Interpolation filter must have minimum filter length to take the use of proposed architecture. The input block size should be equal to one of its up sampling factor.

Modified block formulation for a filter length, N=16 and block size of l=4 for up-sampling factor p=2, 4 and 8 are given below.

a) For p = 2

$$\begin{aligned} \begin{bmatrix} y^{00}{}_k \\ y^{10}{}_k \\ y^{20}{}_k \\ y^{30}{}_k \end{bmatrix} = & \begin{bmatrix} v^0{}_{k,0} & v^0{}_{k-2,0} \\ v^1{}_{k,0} & v^1{}_{k-2,0} \\ v^2{}_{k,0} & v^2{}_{k-2,0} \\ v^3{}_{k,0} & v^3{}_{k-2,0} \end{bmatrix} \otimes \begin{bmatrix} c0 & c4 \\ c0 & c4 \\ c0 & c4 \\ c0 & c4 \end{bmatrix} \\ & + \begin{bmatrix} v^0{}_{k-1,0} & v^0{}_{k-3,0} \\ v^1{}_{k-1,0} & v^1{}_{k-3,0} \\ v^2{}_{k-1,0} & v^2{}_{k-3,0} \\ v^3{}_{k-1,0} & v^3{}_{k-3,0} \end{bmatrix} \otimes \begin{bmatrix} c2 & c6 \\ c2 & c6 \\ c2 & c6 \\ c2 & c6 \end{bmatrix} \\ & + \begin{bmatrix} v^0{}_{k,1} & v^0{}_{k-2,1} \\ v^1{}_{k,1} & v^1{}_{k-2,1} \\ v^2{}_{k,1} & v^2{}_{k-2,1} \\ v^3{}_{k,1} & v^3{}_{k-2,1} \end{bmatrix} \otimes \begin{bmatrix} c1 & c5 \\ c1 & c5 \\ c1 & c5 \\ c1 & c5 \end{bmatrix} \\ & + \begin{bmatrix} v^0{}_{k-1,1} & v^0{}_{k-3,1} \\ v^1{}_{k-1,1} & v^1{}_{k-3,1} \\ v^2{}_{k-1,1} & v^2{}_{k-3,1} \\ v^3{}_{k-1,1} & v^3{}_{k-3,1} \end{bmatrix} \otimes \begin{bmatrix} c3 & c7 \\ c3 & c7 \\ c3 & c7 \\ c3 & c7 \end{bmatrix} \end{aligned} \tag{12}$$

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 6, Issue 11, November 2017

III. PROPOSED RECONFIGURABLE INTERPOLATION FILTER ARCHITECTURE

The configurable architecture consist of three units such as i) coefficient selection unit (CSU) ii) input vector generation unit (VGU) iii) arithmetic unit (AU). CSU unit contains n number of mux`s of depth with j words each. Where, n is filter length and j is the number of interpolation filters of different co- efficient vector to be realized in the re-configurable architecture. The required coefficient vector is selected in one cycle from the CSU.

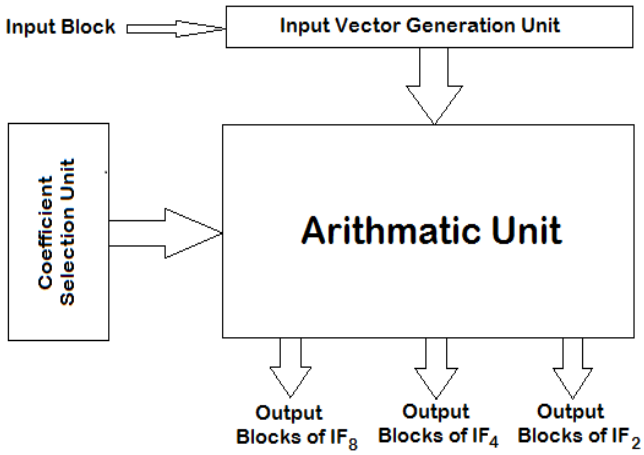

Fig.1: Proposed Reconfigurable Architecture for Interpolation Filters of different Up-sampling factors

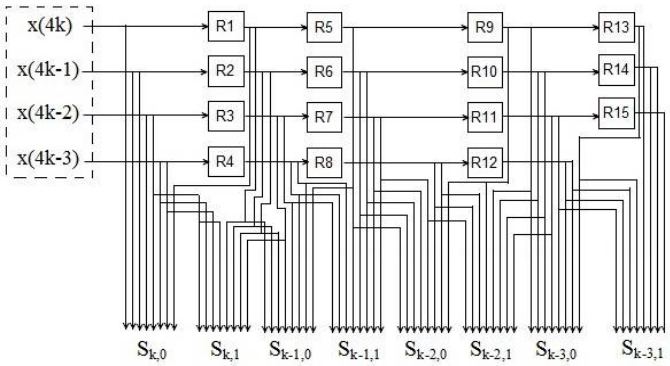

Fig.2 Internal structure of the vector generation unit (VGU)

The vector generation unit (VGU) takes one input block in each cycle and generates (N/P_I) input vectors of size (LP_I) each in parallel where P_I is the smallest up sampling factor from set of different set of up sampling factors to be realized by reconfigurable architecture. The internal structure of vector generation unit for set of up sampling factors {P_1=2, P_2=4, P_3=8} and corresponding filter length is N=16 and block size l=4 .and it contains (N-1) registers. As shown in Fig.2.Those unit receives a block of input samples in every cycle and produces 8 data vectors { $s_{k,0}$; $s_{k,1}$; $s_{k-1,0}$; $s_{k-1,0}$; $s_{k,-1,1}$; $s_{k-2,0}$; $s_{k-2,1}$; $s_{k-3,0}$; $s_{k-3,1}$}; where $s_{k-i,i}$ = [$v^0_{k-i,j}$; $v^1_{k-i,j}$; $v^2_{k-i,j}$; $v^3_{k-i,j}$;] and $V^1_{k-j,i} = s_{k,2j+i}$ {x(4k-l-4j-2i), x(4k-l-4j-2i-1)}

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 6, Issue 11, November 2017

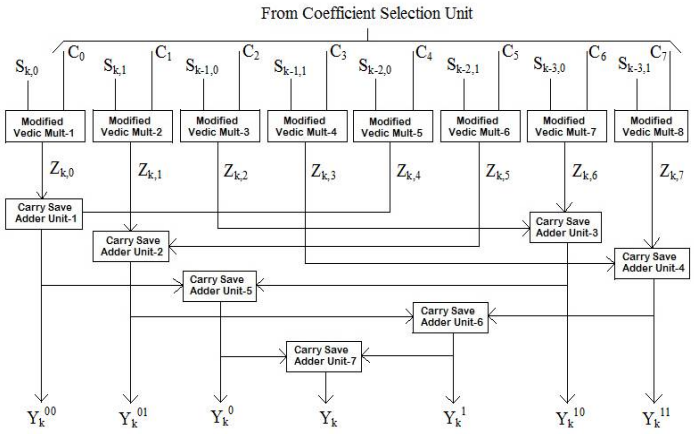

Fig.3 Structure of the arithmetic unit (AU)

The arithmetic unit (AU) comprised of [(N/p1) =8] multiplier units receives an, [(N/p1-1)=7] adder units (ADU) and the structure of ADU is shown in figure . For a set of up sampling factors {p1=2; p2=4; p3=8} Filter length N=16 and block size L=4. Each multiplier unit receives an Lp1 point unit vector $s_{k-j,i}$ from the VGU and p1 point co-efficient vector from the coefficient selection unit (CSU) and produce the output as one partial filter output vector Z_{K-m} of size (N/p1) for 0<=j<=l-1; 0<=j<=p1-1 and 0<=m<=lp1-1.

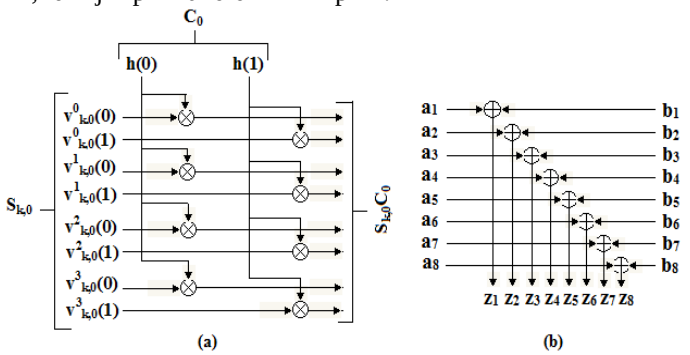

Fig.5: (a) Internal structure of multiplier unit (MU). (b) Internal structure of adder-unit (ADU)

The internal structure of first multiplier unit and adder unit is shown in figure 4. The partial vectors ($z_{k,0}$, $z_{k,4}$), ($z_{k,1}$, $z_{k,5}$), ($z_{k,2}$, $z_{k,6}$), ($z_{k,0}$, $z_{k,4}$), are added in four different ADU`s to compute filter output blocks y_k^{00}, y_k^{01}, y_k^{10}, y_k^{11}, of if8 and where and y_k^{ij} = [$y_k^{0ij,}$ $y_k^{1ij,}$, $y_k^{2ij,}$, y_k^{3ij}] and y_k^{lij} ={y(32k-81-4i-2j);y(32k-81-4j-2j-1)} for 0<=l<=3, i=o,1and j=0,1 The output vectors also represents the partial filter output of (if4) and these partial output vectors added in ADU`S and to obtain complete output vectors($y_k^{0,}$ y_k^{1}) of if4 where (y_k^{lj} = [$y_k^{0j,}$ $y_k^{1j,}$, $y_k^{2j,}$, y_k^{3j}] and y_k^{lj} ={y(16k-8m-2j);y(32k-81-4j-2j-1)} for 0<=l<=3, j=0,1 Similarly the output vectors ($y_k^{0,}$ y_k^{1}) also represents the partial filters output of if 2 added in ADU to obtain complete filter output vector (y_k) of IF2 where (y_k=[$y^0{}_k$, $y^1{}_k$, $y^2{}_k$, $y^3{}_k$] and y_k^i={y(8k-2l);y(8k-81-2l-1)} for 0<=l<=3.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 6, Issue 11, November 2017

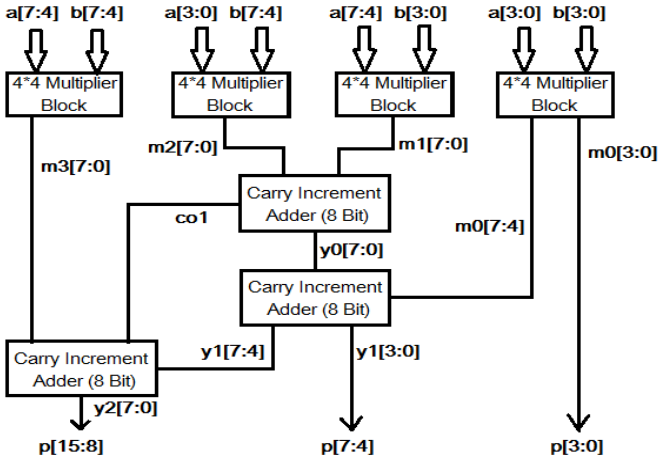

Fig.4 8-bit Modified Vedic Multiplier

Reconfigurable architecture reuses the partial results for parallel computations of filter outputs of different sampling factor. A different coefficient vector of desired filter is selected from the CSU and given to the arithmetic unit to perform filter computation in the case when there is need to change filter specifications. This proposed architecture produces filter output at multiple sampling frequency of an input sampling frequency.

IV. SIMULATION RESULTS

Reconfigurable interpolation filter architecture is designed in Verilog HDL. The simulation results for filter length of 16 and block size of 4 for three different up-sampling factors i.e. P=2, 4 and 8 is given below.

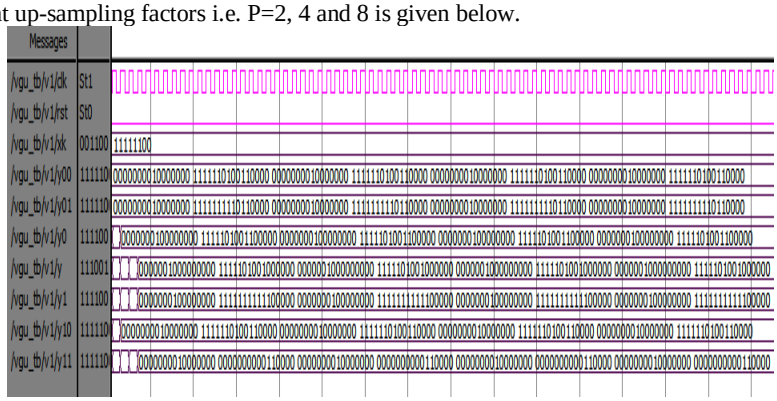

Fig.6: Simulation result

The logic synthesis of this architecture is performed using RC compiler in Cadence for gpdk 90nm technology. The synthesis results are shown in table below.

Parameter	*Filter Length*	*Power (mW)*	*Area (um^2)*	*No of Gates*
Existing	16	14.62	12248	1530
Proposed	16	12.77	12119	1498

Table 1: Logical Synthesis results

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 6, Issue 11, November 2017

V.CONCLUSION

In this paper we have analyzed the interpolation filter computations for different up sampling factors by identify and remove the redundant computations by reusing the partial factors. Modified block formulation technique is used to share partial results of parallel computation of filter outputs of different up sampling factors. This architecture computes the interpolation filter outputs for multiple up-sampling factors in parallel. So due to parallel implementation it is providing high speed response without any extra resources.

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, vol. 62, no. 1, January 2015
[2] R .Mahesh and A. P. Vinod, "Reconfigurable low area complexity filter bank architecture based on frequency response masking for non-uniform Channelization in software defined radio," IEEE Trans. Aerosp. Electron. Syst., vol. 47, no. 2, pp. 1241–1254, Apr. 2011.
[3] E. Buracchini, "The software radio concept," IEEE Commun. Mag., vol. 38, pp. 138–143, Sep. 2000.
[4] J.-K. Choi and S.-Y. Hwang, "Area-efficient pulse-shaping 1:4 interpolated FIR filter based on LUT partitioning," IEEE Electron. Lett., vol. 35, no. 18, pp. 1504–1505, Sep. 1999.
[5] Y. Son, K. Ryoo, and Y. Kim, "1:4 interpolation FIR filter," IEEE Electron. Lett., vol. 40, no. 25, pp. 1570–1572, Dec. 2004.
[6] 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. 1571–1577.
[7] K. H. Chen and T. D. Chiueh, "A low-power digit-based reconfigurable FIR filter," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 53, no. 8, pp. 617–621, Aug. 2006
[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 J. Solid State Circuits, vol. 39, no. 2, pp. 348–357, Feb. 2004.
[9] 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. 247–251.
[10] SreehariVeeramachaneni; Kirthi M Krishna; LingamneniAvinash; Sreekanth Reddy Puppala and M.B. Srinivas;"Novel Architectures for High-Speed and Low-Power 3-2, 4-2 and 5-2 Compressors," 20th International Conference on VLSI Design, 2007, pp.324-329, Jan.2007.
[11] K. Prasad and K. K. Parhi, "Low-power 4-2 and 5-2 compressors," in Proc. of the 35th Asilomar Conf. on Signals, Systems and Computers, vol. 1, 2001, pp. 129–133.
[12] S. F. Hsiao, M. R. Jiang, and J. S. Yeh, "Design of highspeed low-power 3-2 counter and 4-2 compressor for fast multipliers," IEEE Electron. Lett, vol. 34, no. 4, pp. 341–343,1998.
[13] M. Ramalatha, K. D. Dayalan , P. Dharani, and S. D.Priya, "High Speed Energy Efficient ALU Design using Vedic Multiplication Technique ," Lebanon , pp. 600-603, July 2009.
[14] Abhijeet Kumar, Dilip Kumar, Siddhi, "Hardware Implementation of 16*16 bit Multiplier and Square using Vedic Mathematics", Design Engineer, CDAC, Mohali.
[15] M. Ramalatha, K. Deena Dayalan, S. Deborah Priya, "High Speed Energy Efficient ALU Design using Vedic Multiplication Techniques," Advances in Computational Tools for Engineering Applications, 2009, IEEE Proc., pp 600-603.