Scientific Journal of Impact Factor (SJIF) : 3.134

ISSN (Print): 2348-6406
ISSN (Online): 2348-4470

International Journal of Advance Engineering and Research Development

# A Novel Architecture to Implement a Higher Order FIR Filter for Video Signal Demodulation in a Satellite Link

Srividya P[1],Dr.K.R.Nataraj[2]

[1]*Department of ECE, SJBIT, psrividya@sjbit.edu.in*
[2]*Department of ECE, SJBIT, krnataraj@sjbit.edu.in*

***Abstract*** *—The Finite Impulse Response (FIR) filters are widely used in the satellite link to demodulate signals because of its linear phase property and its simple and regular structure compared to Infinite Impulse Response (IIR) filters. This is a basic requirement for applications in audio and video filtering. However to achieve the same frequency specification, FIR filters require higher number of taps compared to IIR filters. This paper proposes an architecture which is a modification of existing Distributed Arithmetic (DA) based FIR filter architecture. It helps in reducing the look up table size required by a DA based filter. It uses multiplexers and adders to implement the Multiply And Accumulate function (MAC). The proposed architecture helps to reduce area, power and increases the computation speed.*

***Keywords****-IIR, FIR, Multiply And Accumulate function*

## I. INTRODUCTION

To improve performance in highly challenging sectors like satellite communication more sophisticated processing algorithms are required. The signal processing algorithms must offer flexibility, improved service quality and reliability. The above objectives should be achieved by taking into account the application dependent system constraints, including complexity, cost, power consumption, speed and design re-usability.

Onboard processing capability is provided in new-generation satellites to make them an intelligent unit than a mere repeater. This enables the satellite to condition, amplify, or reformat received uplink data and route the data to specified locations. A type of Onboard processing, called Base-Band (BB) processing takes place by down-converting, demodulating, detecting the signal and reconstructing the binary information stream to correct any errors that might have been encountered during the uplink.

The implementation of narrow transition band FIR filters are very costly as it requires considerably more arithmetic operations and hardware components such as multipliers, adders and delay elements. This demands more execution time and memory [1].

To resolve the problems of FIR filter implementation, several multiplier less techniques like conversion based approach, memory based approach or Look Up Table (LUT) based approach are proposed.

Among all the methods available, Distributed Arithmetic (DA) was proved to be very efficient solution especially for LUT based FPGA architectures [2-3]. Several modifications for DA architecture to implement FIR filters with smaller resource usage were suggested [4-6].

## II. DISTRIBUTED ARITHMETIC BASED TECHNIQUE

The DA architecture yields better area, power and speed trade off balance when compared to the other available filter architectures for FPGA implementation.

DA is used as the architecture for implementing FIR filters without multipliers. In DA technique, the sum of products computation required for FIR filters is realized using LUTs, shifters and adders. These operations can be mapped efficiently onto an FPGA. Hence DA is a desired architecture on these devices.

DA is used to design bit-level architectures for vector to vector multiplications. In distributed arithmetic, each word in the vector is represented as a binary number, the multiplications are reordered and mixed, such that the arithmetic becomes “distributed” throughout the structure .

DA relies on the fact that the filter coefficients, Ak are known, so multiplying Akxk becomes a multiplication with a constant. This is a prerequisite for a DA design. It is a powerful technique for reducing the size of a parallel multiply-accumulate hardware that is well suited to FPGA designs.

To reduce the hardware complexity substantially, a symmetrical filter structure can be used. It halves the number of coefficients. But one additional clock cycle is taken before the output is obtained. The area saving obtained by using DA is about 50%.

Partitioning the DA based filter reduces the single look up table into a desired number of LUTs. The computations are broken down into smaller elements that can be executed in parallel. Greater speed benefits can be attained especially for higher precision calculations. The area occupied by parallel DA based filter is more compared to serial filter.

FIR filter is expressed as

$$y = \sum_{k=1}^{K} A_k\, x_k \qquad (1)$$

$x_k$can be expressed as

$$x_k = -b_{k0} \sum_{n=1}^{N-1} b_{kn}\, 2^{-n} \qquad (2)$$

By proper rearrangement, filter output can be written as

$$y = -\sum_{k=1}^{K} A_k * (b_{k0}) \sum_{n=1}^{N-1} \left[ \sum_{k=1}^{K} A_k * b_{kn} \right] 2^{-n} \qquad (3)$$

The DA based filter includes three units: shift register unit, the DA LUT unit and the adder/shifter unit.

## III. PROPOSED FILTER ARCHITECTURE

In DA based technique, the summation$\sum_{k=1}^{K} A_k\, b_{kn}$in equation (3) is pre calculated for all the possible combinations. When the coefficients of $c_k$are known and theinput values are either 1 or 0, then each SOP is just a combination of the $c_k$'s for which a true table can be constructed. This is called a Look up table (LUT).The LUT has $2^k$words addressed by k-bits.

The number of co-efficient for a higher order filter is more. In a DA based technique, constructing a LUT for all possible combination is a tedious and time consuming process.

The proposed filter architecture makes use of multiplexers and adders instead of LUTs. This reduces the number of computations required to create a LUT and increases the processing speed.

### *A. Algorithm of the proposed filter architecture*

1. Obtain the filter coefficients and the filter order using Matlab. Parks-McClellan algorithm is used. Convert the floating point filter coefficients into integers.
2. Convert the filter input into its equivalent binary value.
3. Divide the total number of filter coefficients into groups. Each group should have 2 values.
4. Start with LSB input bits of group1.
5. According to the combination of 0's and 1's add the first 2 coefficients.
6. Consider the input bits of group2 of same position.
7. According to the combination of 0's and 1's add the next two coefficients to the sum obtained in step5.
8. Repeat steps 6 and 7 till all the groups are considered.
9. Multiply the sum obtained by its place value.
10. Now move on to next bit position. Consider bits of group 1.
11. Repeat steps 5 to 8 till all the input bits are considered. After computing the sum of each bit position, multiply the sum obtained by its place value and adds it to sum obtained in step 9.

The block diagram of the proposed filter architecture is shown in Fig.1. The decision device is used to compare filter coefficients. Finally $Y_B$ obtained after each bit position is added to get the final output of the filter.

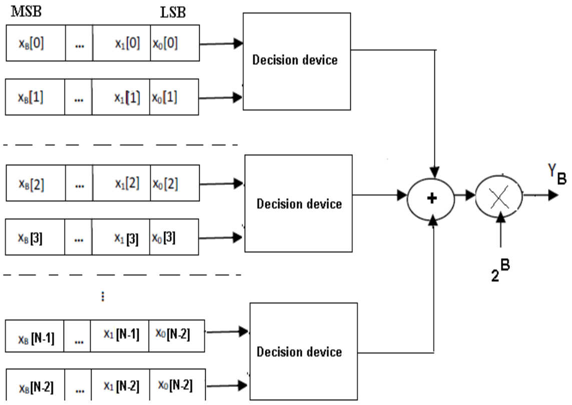

MSB
LSB
$x_B[0]$ ... $x_1[0]$ $x_0[0]$
$x_B[1]$ ... $x_1[1]$ $x_0[1]$
$x_B[2]$ ... $x_1[2]$ $x_0[2]$
$x_B[3]$ ... $x_1[3]$ $x_0[3]$
$x_B[N-1]$ ... $x_1[N-1]$ $x_0[N-2]$
$x_B[N-2]$ ... $x_1[N-2]$ $x_0[N-2]$
Decision device
Decision device
Decision device
+
$Y_B$
$2^B$

*Fig.1. Block diagram of proposed filter architecture.*

## IV. PARKS-MCCLELLAN ALGORITHM

It is an iterative algorithm to design and implement efficient and optimal FIR filters. To find the optimal filter coefficients it uses an indirect method.The algorithm minimizes the error in the pass and stop bands by utilizing the Chebyshev approximation.

The design of an FIR filter using Parks-McClellan algorithm is a two-step process.

1. Estimate the order of the optimal Parks-McClellan FIR filter to meet the design specifications using the remezord command.
2. Actual design of the filter using remez command. This gives the coefficients of the filter.

## V. IMPLEMENTATION METHODOLOGY AND RESULTS

The Quadrature Carrier Multiplexed scheme is used. This scheme enables two Double Side band suppressed carrier modulated waves (resulting from the application of two physically independent message signals) to occupy the same channel bandwidth, and yet it allows for the separation of the two message signals at the receiver output. It is therefore a bandwidth-conservation scheme. A block diagram of the Quadrature Carrier Multiplexing system is shown in Fig.2.

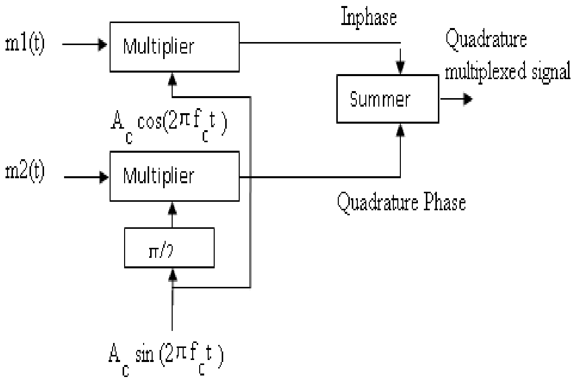


*Fig. 2. Quadrature Multiplexing of two signals m1(t) and m2(t).*

The transmitter part of the system, involves the use of two separate product modulators that are supplied with two carrier waves of the same frequency but differing in phase by -90 degrees. The transmitted signal s(t) consists of the sum these two product modulator outputs, as shown by

$$s(t) = A_c m_1(t) \cos(2\pi f_c t) + A_c m_2(t) \sin(2\pi f_c t) \text{ --- } (4)$$

Where $m_1(t)$ and $m_2(t)$ denote the two different message signals applied to the product modulators. Thus s(t) occupies a channel bandwidth of 2W centred at the carrier frequency $f_c$, where W is the message bandwidth of $m_1(t)$ and $m_2(t)$. According to equation (4), $A_c m_1(t)$ can be viewed as the in-phase component of the multiplexed band pass signal s(t) and $A_c m_2(t)$ as its quadrature component.

The receiver part of the system is shown in Fig.3. The multiplexed signal s(t) is applied to two separate coherent detectors that are supplied with two local carriers of same frequency, but differing in phase by -90 degrees. The output of the top detector is ½ $A_c^1 m_1(t)$, whereas the output of the bottom detector is ½ $A_c^1 m_2(t)$. For the system to operate satisfactorily it is important to maintain the correct phase and frequency relationships between the local oscillators used in transmitter and receiver parts of the system.

The message signals considered for analysis are 10KHz and 20KHz and is shown in Fig.4. The sampling frequency of 1.2 MHz and carrier frequency of 1 MHz is used.

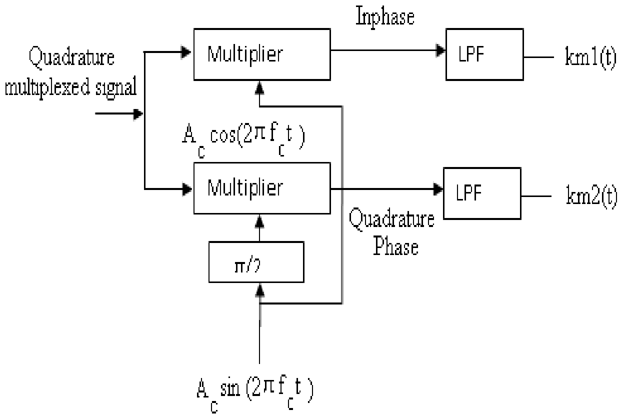


***Fig. 3. Coherent detection of Quadrature Multiplexed signals.***

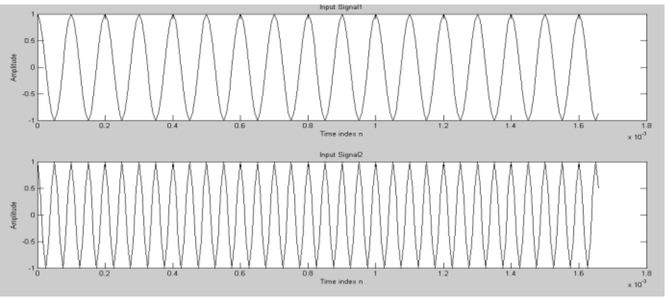


***Fig. 4. Message signals considered for analysis.***

The modulator and demodulator are coded in Matlab. The Low pass filter coefficients and the filter order is calculated using Parks-McClellan algorithm. The filter response for cut-off of 10 KHz and 20 KHz is shown in Fig.5. The demodulator output obtained in Matlab is shown in Fig.6.

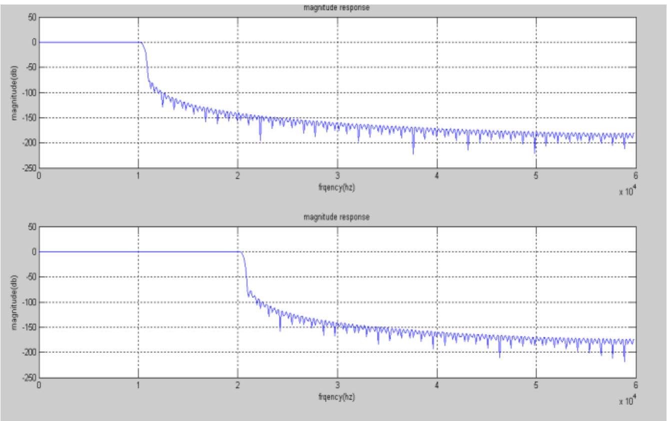


***Fig. 5.Low Pass Filter Response.***

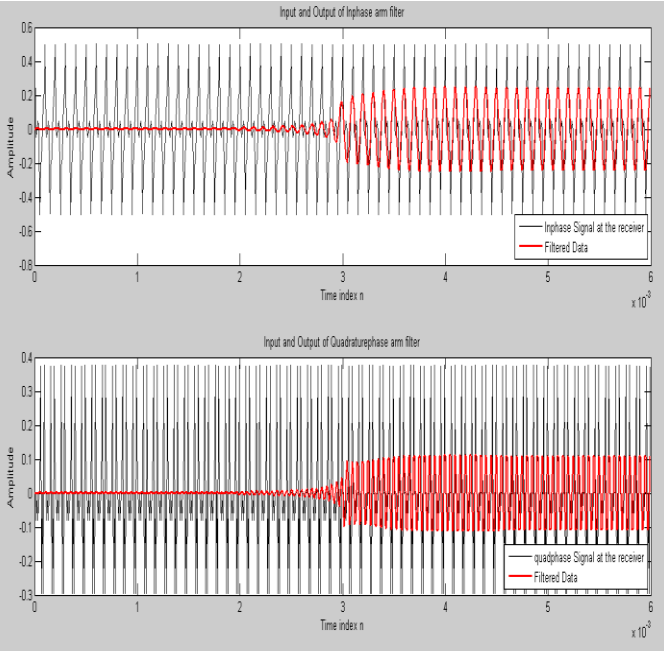


***Fig. 6.Matlab output of the Demodulator.***

The low pass filter using the proposed architecture is coded in VHDL and RTL schematic of the same is shown in Fig.7. Modelsim simulator is used to verify the design and the output obtained in the simulator for Inphase arm is shown in Fig.8. The filter output obtained from VHDL coding is stored in an external file and these integer values are read in Matlab. The outputs from Matlab and VHDL are compared as shown in Fig.9 and they match in their frequency. The integer equivalent of output obtained in Matlab is shown in Fig.10. It is same as the integer values obtained in Modelsim as shown in Fig.8. Last few values of the output are also compared and shown in Fig.11. The Matlab and VHDL filter outputs for quadrature phase arm is also shown in Fig.12.

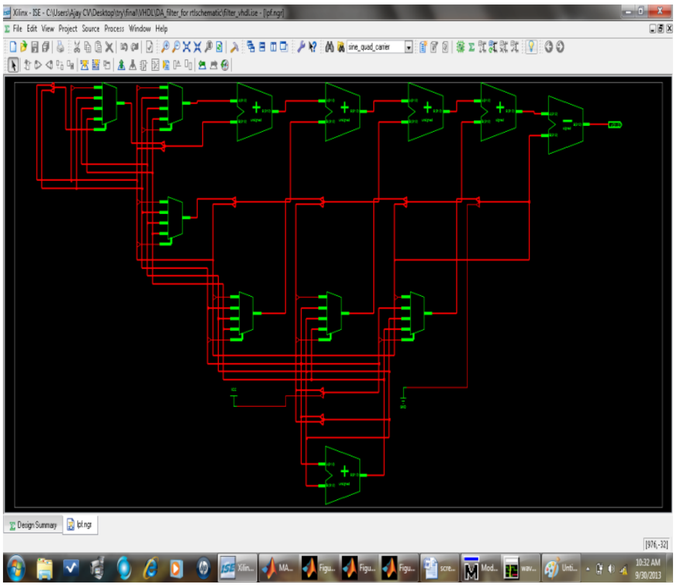
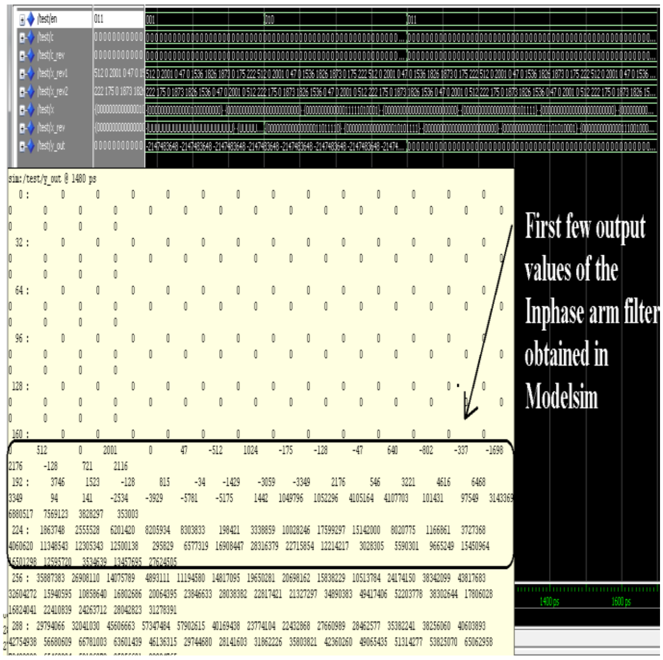


***Fig. 7. RTL Schematic of Proposed Filter Architecture .***



```
sim:/test/y_out @ 1480 ps
  0 :        0        0        0        0        0        0        0        0        0        0        0        0        0
0        0        0        0        0        0        0        0        0        0        0        0        0        0        0
0        0        0        0
 32 :        0        0        0        0        0        0        0        0        0        0        0        0        0
0        0        0        0        0        0        0        0        0        0        0        0        0        0        0
0        0        0        0
 64 :        0        0        0        0        0        0        0        0        0        0        0        0        0
0        0        0        0        0        0        0        0        0        0        0        0        0        0        0
0        0        0        0
 96 :        0        0        0        0        0        0        0        0        0        0        0        0        0
0        0        0        0        0        0        0        0        0        0        0        0        0        0        0
0        0        0        0
128 :        0        0        0        0        0        0        0        0        0        0        0        0        0
0        0        0        0        0        0        0        0        0        0        0        0        0        0        0
0        0        0        0
160 :        0        0        0        0        0        0        0        0        0        0        0        0        0
0      512        0     2001        0       47     -512     1024     -175     -128      -47      640     -802     -337    -1698
2176     -128      721     2116
192 :     3746     1523     -128      815      -34    -1429    -3059    -3349     2176      546     3221     4616     6468
3349       94      141    -2534    -3929    -5781    -5175     1442  1049796  1052296  4105164  4107703   101431    97549  3143369
6880517  7569123  3828297   353003
224 :  1863748  2555528  6201420  8205934  8303833   198421  3338859 10028246 17599297 15142000  8020775  1166861  3727368
4060620 11348543 12305343 12500138   295829  6577319 16908447 28316379 22715854 12214217  3028305  5590301  9665249 15450964
16501298 12595720  3534639 13457695 27624505
256 : 35887383 26908110 14075789  4893111 11194580 14817095 19650281 20698162 15838229 10513784 24174150 38342099 43817683
32604272 15940595 10858640 16802686 20064395 23846633 28038382 22817421 21327297 34890383 49417406 52203778 38302644 17806028
16824041 22410839 24263712 28042823 31278391
288 : 29794066 32041030 45606663 57347484 57902615 40169438 23774104 22432868 27660989 28462577 35382241 38256060 40603893
42754938 56680609 66781003 63601439 46136315 29744680 28141603 31862226 35803821 42360260 49065435 51314277 53825070 65062958
```

First few output values of the Inphase arm filter obtained in Modelsim

Fig. 8.Modelsim Output for Inphase arm filter.

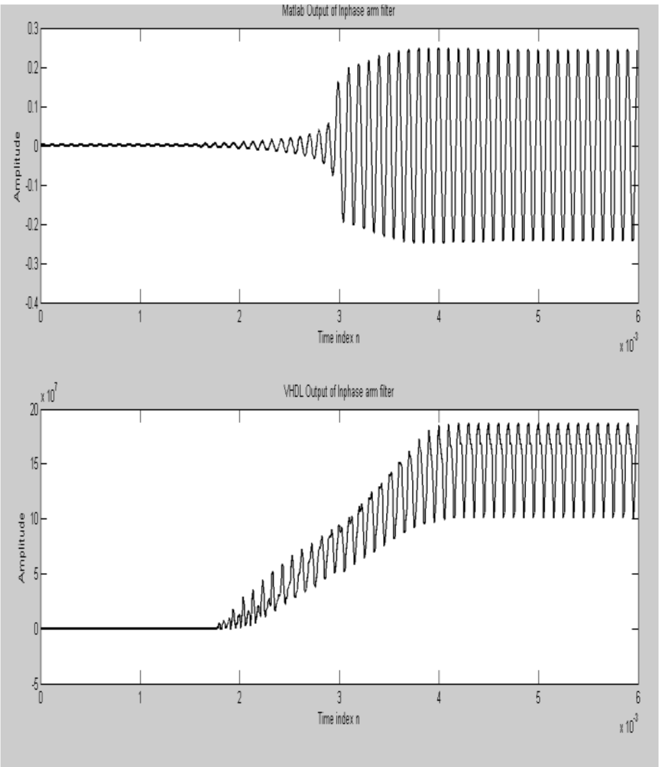
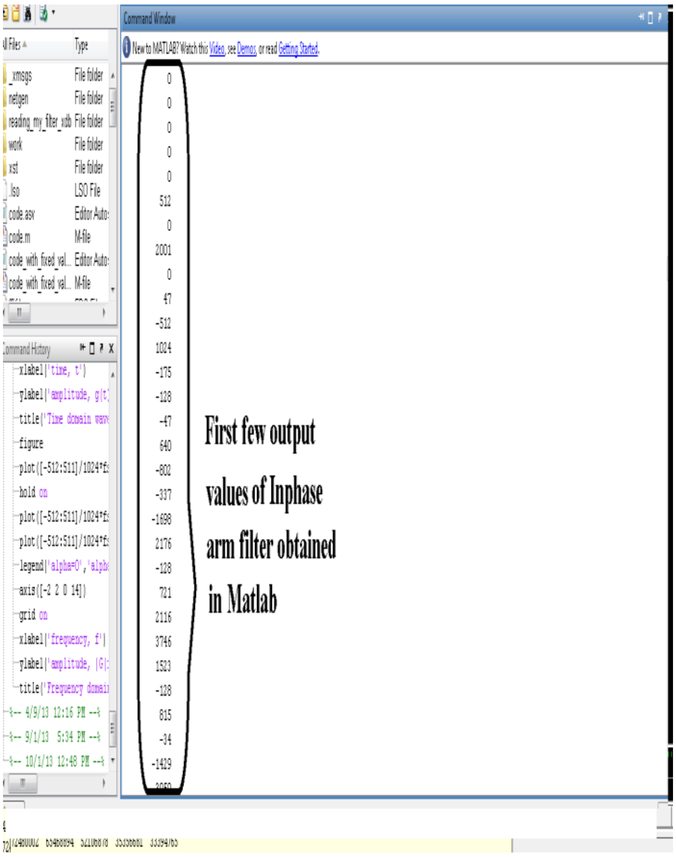


Fig. 9.Matlab and VHDL Filter output for Inphase arm.

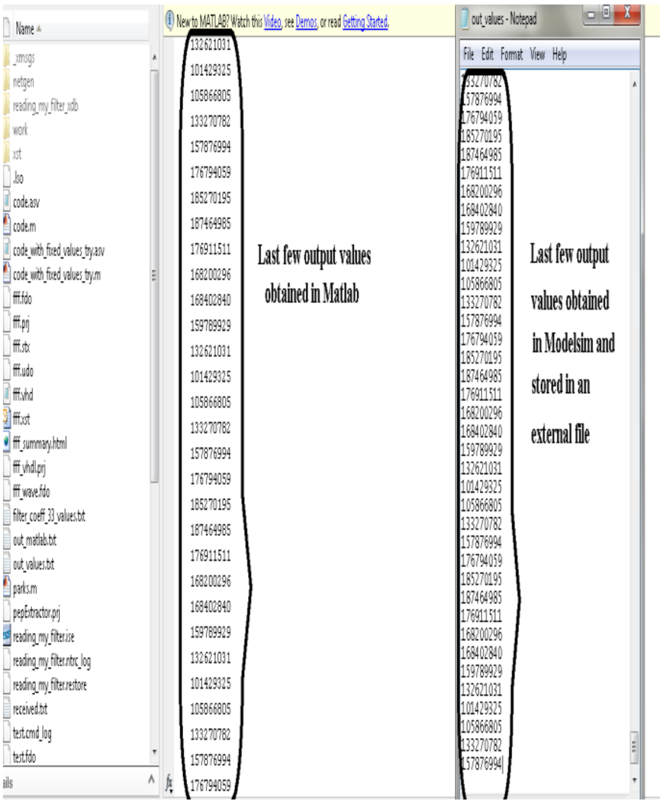


Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Command History
First few output values of Inphase arm filter obtained in Matlab

```
xlabel('time, t')
ylabel('amplitude, g(t'
title('Time domain wav
figure
plot([-512:511]/1024*f
hold on
plot([-512:511]/1024*f
plot([-512:511]/1024*f
legend('alpha=0','alph
axis([-2 2 0 14])
grid on
xlabel('frequency, f')
ylabel('amplitude, |G|
title('Frequency domai
%-- 8/9/13 12:16 PM --%
%-- 9/1/13  5:34 PM --%
%-- 10/1/13 12:48 PM --%
```

```
     0
     0
     0
     0
     0
   512
     0
  2001
     0
    47
  -512
  1024
  -175
  -128
   -47
   640
  -802
  -337
 -1698
  2176
  -128
   721
  2116
  3746
  1523
  -128
   815
   -34
 -1429
```

*Fig. 10. Filter output obtained in Matlab converted into integers.*



New to MATLAB? Watch this Video, see Demos, or read Getting Started.
out_values - Notepad
File Edit Format View Help
Last few output values obtained in Matlab
Last few output values obtained in Modelsim and stored in an external file

```
132621031
101429325
105866805
133270782
157876994
176794059
185270195
187464985
176911511
168200296
168402840
159789929
132621031
101429325
105866805
133270782
157876994
176794059
185270195
187464985
176911511
168200296
168402840
159789929
132621031
101429325
105866805
133270782
157876994
176794059
```

```
133270782
157876994
176794059
185270195
187464985
176911511
168200296
168402840
159789929
132621031
101429325
105866805
133270782
157876994
176794059
185270195
187464985
176911511
168200296
168402840
159789929
132621031
101429325
105866805
133270782
157876994
176794059
185270195
187464985
176911511
168200296
168402840
159789929
132621031
101429325
105866805
133270782
157876994
```

*Fig. 11. Filter output for last few values obtained in Matlab and VHDL*

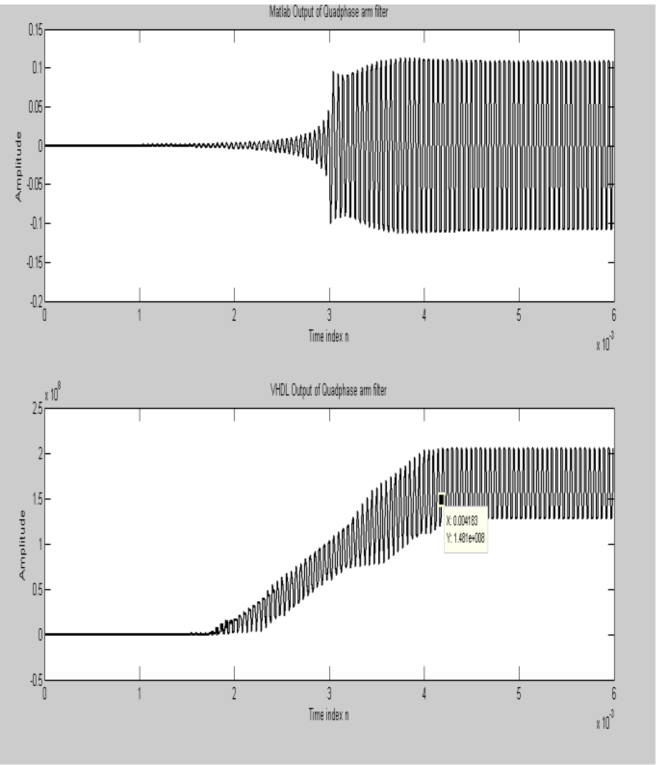


***Fig. 12.Matlab and VHDL Filter output for Quadrature phase arm.***

The target board used for the design implementation is Spartan3-XC3S400-PQ208.

## VI. CONCLUSION

The proposed FIR filter architecture suggests a modification to DA based approach. The filter avoids unnecessary calculations involved to create a LUT in DA method. This is useful to find the filter output especially when more number of inputs and filter coefficients are involved. Such a situation usually arises in satellite demodulators while demodulating video signals.

## REFERENCES

[1] P.Ramesh Babu, "Digital Signal Processing", SCITECH publications, 4th edition, 2012.

[2] Daniel J Allred, Walter Huang, Venkatesh Krishnan HeejongYoo and David V. Anderson, "An FPGA implementation for a High Throughput Adaptive Filter using Distributed Arithmetic", IEEE symposium on Field programmable custom computing machines, pp:324-325, 2004.

[3] T. Vigneswaran, P. Subbaramireddy, "Design of Digital FIR Filter Based on Dynamic Distributed Arithmetic Algorithm", Journal of applied Sciences, pp: 2908-2910, 2007.

[4] Wang Sen, Tang Bin, Zhu Jun, "Distributed Arithmetic for FIR Filter Design on FPGA", ICCCAS, pp:620-623, 2007.

[5] Patrick Longa, Ali Miri, "Area efficient FIR Filter Design on FPGAs using Distributed Arithmetic", IEEE International Symposium on Signal Processing and Information Technology, pp: 248-252, 2006.

[6] Narendra Singh Pal, Harjit Pal Singh, R.K.Sarin, Sarabjeet Singh, "Implementation of high speed FIR filter using serial and parallel Distributed Algorithm", IJCA, vol.7,pp:26-32, 2011.

[7] K.R Nataraj, S Ramachandran, B.S Nagabushana, "Development of Algorithm for demodulator for processing satellite data communication", IJCSNS, Vol 9, No 6, pp: 233-243, June 2009.

[8] K.R Nataraj, S Ramachandran, B.S Nagabushana, "Development of Algorithm, Architecture and FPGA implementation of demodulator for processing satellite data communication", IJCSNS, Vol 9, No 7, pp: 137-147, Ju1y 2009.

[9] Timotthpratt, " Satellite Communication ",Wiley publication,2008.

[10] Michae.O.Kolawole, "Satellite Communication Engineering", Marcel Dekker Inc, 2002.

Dennis roddy, "Satellite Communication", McGraw Hill, 3rd edition.