FPGA based Speed Efficient Decimator using Distributed Arithmetic Algorithm

Rajesh Mehra
Assoc. Professor (ECE Deptt.)
NITTTR, Sec - 26, Chandigarh
India

Lajwanti Singh
Assist. Professor (ECE Deptt.)
AICE, Jaipur
India

ABSTRACT
In this paper, an efficient FPGA implementation of a multipliers less decimator is presented for wireless application. DA has been used to implement a decimator taking advantage of embedded LUT based structure of FPGAs. Speed and area efficient solution is designed using half band polyphase decomposition FIR structure. The proposed decimator has been designed with MATLAB and synthesized with Xilinx synthesis tool (XST)10.1 and implemented on Spartan-3E based 3s500efg.320-4 FPGA device. Improvement of 28% in speed and 50% in area has been observed as compared to MAC based approach.

Keywords
DA, Decimator, DSP, FIR, FPGA, LUT, XST

1. INTRODUCTION
In the last few years, there has been a growing trend to implement DSP functions in Field Programmable Gate Arrays (FPGAs), which offer a balanced solution in comparison with traditional devices. Although ASICs and DSP chips have been the traditional solution for high performance applications, now the technology and the market are imposing new rules. On one hand, high development costs and time-to-market factors associated with ASICs can be prohibitive for certain applications and, on the other hand, programmable DSP processors can be unable to reach a desired performance due to their sequential-execution architecture. In this context, FPGAs offer a very attractive solution that balance high flexibility, time-to-market, cost and performance [1].

A Field Programmable Gate Array or FPGA is a silicon chip containing of, but not limited to an array of configurable logic blocks (CLBs). Nowadays, FPGAs as cost-effective integrated tools have many applications in the field of communication and a growing range of other areas. Using FPGAs for hardware acceleration in wireless applications offers extensive processing power to realize promised portability of waveforms and reconfigurability [2]. Key advantages of FPGA are:

i. Their architectures are well suited for highly parallel implementation of DSP functions to enhance system performance.

ii. User programmability allows designers to trade-off device area vs. performance by selecting the appropriate level of parallelism to implement their functions. By programming the FPGA to use more on-chip resources, designers can achieve higher performance. By using less resources (and accepting a corresponding lower performance), designers can optimize the design for low cost.

iii. The other added advantage of FPGAs is its ability to integrate system logic. There are still plenty of resources left to handle traditional FPGA tasks such as the interfaces between other devices on the PC board. Both the system logic and the DSP functions can be implemented in one cost-effective fabric.

The evolution and growth in telecommunication system and services run in parallel to integrated circuit design techniques evolution. New fabrication process allows a new functionalities and higher performance. The design of the transmission side of a digital communication system based on the use of specialized digital up converters (DUC) where baseband processing is performed by a high performance DSP and up conversion to IF is performed by a digital chip versus FPGA based design. Similarly in receiver side of digital communication system, digital down converter (DDC) is used [3].

Discrete time systems with unequal sampling rates at various parts are called multi rate systems. Multi-rate systems are building blocks commonly used in digital signal processing (DSP). Their function is to alter the rate of the discrete-time signals, by adding or deleting a portion of the signal samples [4]. The rate conversion requirement leads to production of undesired signals associated with aliasing and imaging errors. So some kind of filter should be placed to attenuate these errors.

FIR filter is a finite impulse response filter which is non-recursive in nature. That is, there is no feedback involved. The impulse response of an FIR filter will eventually reach zero. FIR filter is stable and has linear phase. It depends only on inputs and consists of only zeroes [5]. As a DSP function, FIR are widely used in FPGA implementations. If very high sampling rates are required, full-parallel hardware must be used where every clock edge feeds a new input sample and produces a new output sample [6]. In case fully parallel implementation is not possible then partly serial approach can be adopted to enhance the system performance. Such filters can be implemented on FPGAs using combinations of the general purpose logic fabric, onboard RAM and embedded arithmetic hardware.

The remainder of this paper is organized as follows: First the concept of Decimators and Distributed arithmetic algorithm is described. A brief discussion of decimator requirement, polyphase structure and MATLAB based simulation of equiripple based decimator is presented. Then FPGA implementation of proposed decimator is given with comparison of resources and speed. Some final remarks are given in the conclusion.

2. DECIMATOR

Sampling rate conversion (SRC) is important part of today's digital signal processing systems and is one of the most frequent and useful tasks in the field of communication. SRC involves resampling which itself causes aliasing and imaging. The elimination of the detrimental effects of these two fundamental characteristics of resampling requires filtering [7]. Interpolators and decimators are utilized to increase or decrease the sampling rate.

With Decimation, the sample rate of the discrete signal xi[k] can be decreased by the integer factor M. In the simplest case we can take every Mth sample of xi[k].

$$xo[k] = xi[k \cdot M]$$

This is a sampling process and can therefore cause aliasing. To prevent information loss below Fo, we have to reduce the information selectively. This is done by a low pass filter hM[k] of cut off frequency FCM. For the convolution in the time domain a filter sample rate of Fi is needed (Tf = Ti).

$$FCM = F_o/2 = F_i/2M$$

$$HM(f) = \begin{cases} 1 \text{ for FCM} \geq |f| \\ 0 \text{ for FCM} \leq |f| \leq 2 \text{ FCM} \end{cases}$$

By removing samples, the average signal energy per sample is not changed and no amplification in the filter's pass band is needed. It is not necessary to filter xi over the whole time axis, since only every Mth sample is needed afterwards. Decimation can be described by the following block diagram:

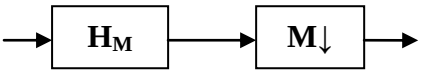


Fig 1: Decimation process

3. DALUT ALGORITHM

DALUT algorithm is an efficient method for computing inner products when one of the input vectors is fixed. It uses look-up tables and accumulators instead of multipliers for computing inner products and has been widely used in many DSP applications such as DFT, DCT, convolution, and digital filters [8]. This technique, first proposed by Croisier is a multiplier-less architecture that is based on an efficient partition of the function in partial terms using 2's complement binary representation of data. The partial terms can be pre-computed and stored in LUTs. The flexibility of this algorithm on FPGAs permits everything from bit-serial implementations to pipelined or full-parallel versions of the scheme, which can greatly improve the design performance.

The multiplier less distributed arithmetic (DA)-based technique has gained substantial popularity, Due to its high-throughput processing capability and increased regularity, results in cost-effective and area-time efficient computing structures. The main operations required for DA-based computation of inner product are a sequence of lookup table (LUT) accesses followed by shift accumulation operations of the LUT output. DA-based computation is well suited for FPGA realization, because the LUT as well as the shift-add operations, can be efficiently mapped to the LUT-based FPGA logic structures [9].

Multiplier-less schemes can be classified in two categories according to how they manipulate the filter coefficients for the multiply operation. In first type of multiplier-less technique, the coefficients are transformed to other numeric representations whose hardware implementation or manipulation is more efficient than the traditional binary representation such as Canonic Sign Digit (CSD) method, in which coefficients are represented by a combination of powers of two in such a way that multiplication can be simply implemented with adder/subtractors and shifters [9]. The second type of multiplier-less method involves the use of memories (RAMs, ROMs) or Look-Up Tables (LUTs) to store pre-computed values of coefficient operations.

In FIR filtering, one of the convolving sequences is derived from the input samples while the other sequence is derived from the fixed impulse response coefficients of the filter. This behavior of the FIR filter makes it possible to use DA-based technique for memory-based realization. It yields faster output compared with the multiplier-accumulator-based designs because it stores the pre computed partial results in the memory elements, which can be read out and accumulated to obtain the desired result. The memory requirement of DA-based implementation for FIR filters, however, increases exponentially with the filter order [10].

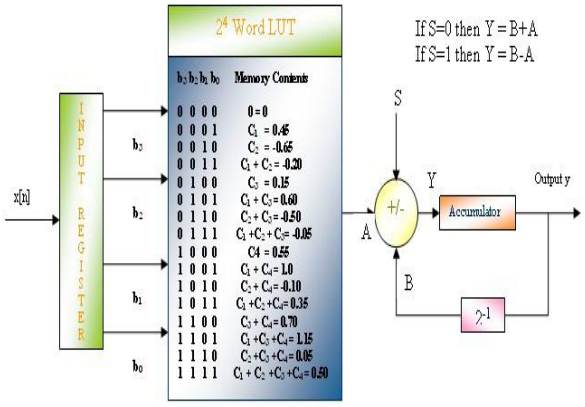


Fig 2: LUT based Multiplier-less Implementation [8]

4. MATLAB BASED SIMULATION

When a decimation filter is designed, it is important to decide which type of filters will be used and where decimation will occur. Design flow for decimator filter implementation is shown in figure 3. We go through the same design flow to synthesize and simulate the proposed decimator structure. Polyphase decomposition is used to implement parallel processing for decimation.

Following steps are followed to implement the proposed structure:

i. Define the filter specifications such as order, filter design, cut-off frequency and sampling frequency.

ii. Calculate the filter coefficient using MATLAB FDA Tool.

iii. Apply DA Algorithm on filter Coefficients.

iv. Configure the target FPGA, Virtex 2 based 3s500efg320-4 kit for real time debugging.

v. Check the DA Filter coefficients on target FPGA Chip.

vi. Configure the target FPGA.

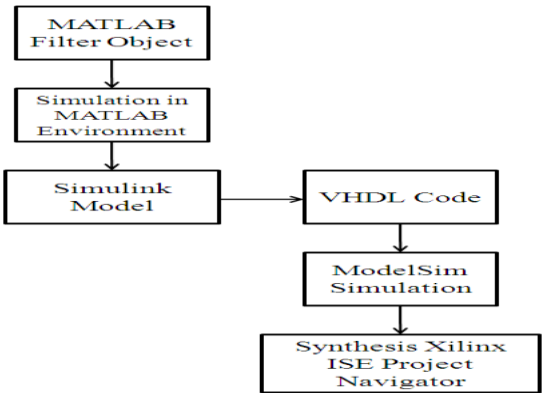
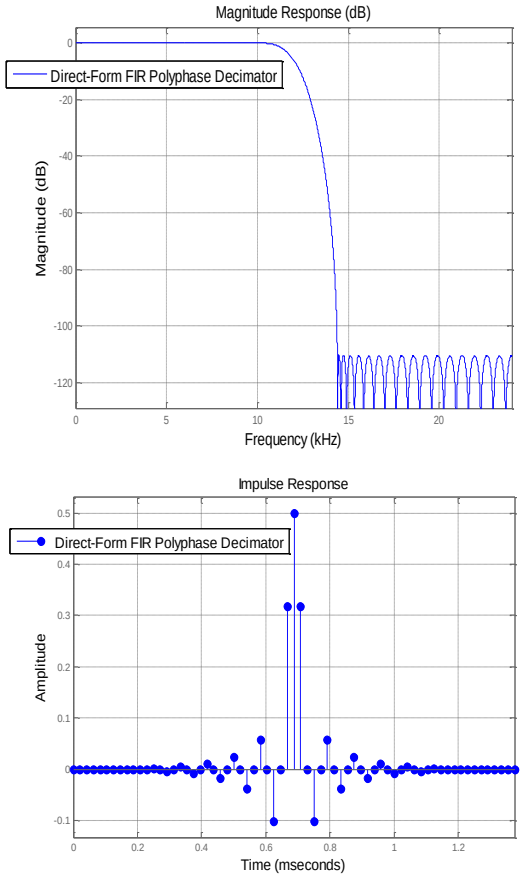


Fig 3: Decimation Filter Implementation: Design Flow

Equiripple based half band polyphase decimator is designed and implemented using MATLAB. Order of the proposed decimator filter is 66 & implementation cost has been calculated in terms & multiplies and MIPS. The Symmetric structure has shown reduced consumed number of multipliers. The Symmetric direct form FIR filter has consumed 50% less multipliers compared to transferred direct form FIR filter for providing cost and area efficiency [12].

In the decimator design MAC unit has been replaced with LUT unit which is proposed multiplier less technique. The size of the LUT grows exponentially with the order of the filter. Here 67 coefficients are divided in two sections with 34 and 33 coefficients respectively to perform using polyphase decomposition. Then 34 coefficients of one part have been processed by using (8 8 8 8 2) DALUT partitioning to limit the size of LUTs



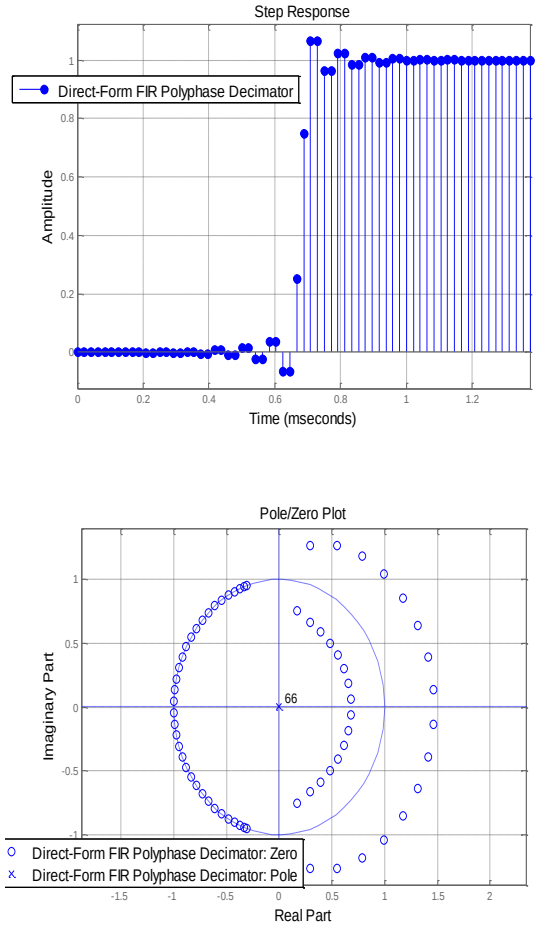


Fig4: Various Output Response of FIR Polyphase Decimator

5. FPGA IMPLEMANTATION

The multiplier based and multiplier less decimator are implemented and synthesized on Spartan-3E based 3s500efg320-4 target device. The modelsim based simulated output of the proposed decimator with 16 bit precision is shown in figure 5.

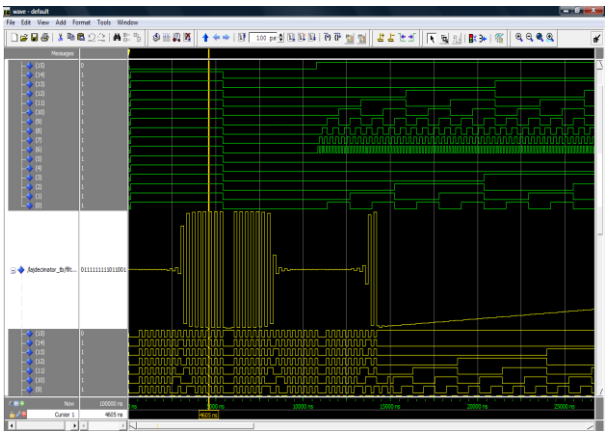


Fig 5: Simulated Decimator Output

```
Speed Grade: -4

   Minimum period: 15.660ns (Maximum Frequency: 63.857MHz)
   Minimum input arrival time before clock: 10.209ns
   Maximum output required time after clock: 4.283ns
   Maximum combinational path delay: No path found
```

Table 1 shows the area and speed comparision of both the techniques. The proposed DA based design shows 28% enhancement in speed by saving almost 50 % of the resouces as compared to MAC based design.

Table 1. Resource utilization

Logic Utilization	Multiplier Approach	Proposed Multiplier Less Approach
No. of Slices	1055 out of 4656 (22%)	566 out of 4656 (12%)
No. of Flip Flops	1210 out of 9312 (12%)	515 out of 9312 (5%)
Speed (MHz)	49.574	63.857

6. CONCLUSION

In this paper, an optimized half band polyphase decomposition technique has been presented to implement the decimator for multi-rate applications. FIR filter of order 66 has been designed using Symmetric direct form structure which shows 50% reduction in multipliers and MIPS as compared to transposed form. Distributed Algorithm has been used to further enhance the speed by taking advantage of LUT structure of target FPGA. The proposed multiplier less approach has shown an improvement of 28% in speed by saving almost 50% resources of target device as compared to multiplier based approach. So DA based symmetric structure is well suited for cost effective implementation of decimator design.

7. ACKNOWLEDGMENTS

The authors would also like to thank Director, National Institute of Technical Teachers' Training & Research, Chandigarh, India and Director AICE, Jaipur, Rajasthan, India for their inspiration and support throughout this research.

8. REFERENCES

[1] 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.

[2] A. Beygi, A. Mohammadi, A. Abrishamifar, "An FPGA Based Irrational Decimator for Digital Receivers", IEEE International Conference on Signal Processing and its Application, pp: 1 – 4, 2007.

[3] Santhosh Y N, Cyril P R, Namita Palacha, "Design and VLSI Implementation of Interpolators/Decimators for DUC/DDC", IEEE Conference on Emerging Trends in Engineering and Technology, pp: 755-759, 2010.

[4] Rajesh Mehra, Swapna Devi, "Optimized Design of Decimator for Alias Removal in Multirate DSP Applications", Proceedings of the 10th WSEAS International Conference on Wavelet Analysis and Multirate Systems, pp: 100-103, 2010.

[5] Sonika Gupta, Aman Panghal. "Performance Analysis of FIR Filter Design by Using Rectangular, Hanning and Hamming Windows Methods", IJARCSSE, Volume 2, Issue 6, June 2012

[6] Kanu Priya, Rajesh Mehra. "Area Efficient Design of FIR Filter Using Symmetric Structure", International Journal of Advanced Research in Computer and Communication Engineering, Volume 1, Issue 10, December 2012.

[7] G. J. Dolecek, F. Harris, "Design of CIC Compensator Filter in a Digital IF Receiver", IEEE International Conference on Communication and Information Technologies, pp: 638 – 643, 2008.

[8] Rajesh Mehra, Swapna Devi, "FPGA Based Design of High Performance Decimator using DALUT Algorithm", ACEEE International Journal on Signal and Image Processing, Volume 1, pp. 9-13, 2010.

[9] Pramod Kumar Meher, Shrutisagar Chandrasekara,

Abbes Amira, "FPGA Realization of FIR Filters by Efficient and Flexible Systolization Using Distributed Arithmetic" IEEE Transaction On Signal Processing, Volume 56, No.7, 2008.

[10] Rajesh Mehra, Ravinder Kaur, "FPGA based Efficient

Interpolator design using DALUT Algorithm", NeTCoM 2010, CSCP 01, pp. 51–62, 2011

[11] Rajesh Mehra, Lajwanti Singh, "Cost Analysis and Simulation of Decimator for Multirate Applications", International Journal of Computers and Technology, volume 11, pp. 2175-81, 2013.