Abstract: In this paper, FPGA realization of MUX based multiplier and odd multiple scheme architectures are proposed for FIR filter and discussed in terms of complexity. In digital filter implementation, the multiplier usage is avoided by using MUX based multiplier and Look Up Table (LUT) based multiplier. These multipliers are used for constructing direct form FIR filters with signed number representation. The two architectures have been implemented using Verilog and synthesized using SPARTRAN XC3S500E. The performance is analyzed for mux based and LUT based Multiplier.

Keywords:-FIR Filter, Look up Table, Reconfigurable Architecture, Distributed Arithmetic.

I. INTRODUCTION
Finite Impulse Response (FIR) digital filter is widely used as a basic block in signal and image processing applications. The number of multiply-accumulate (MAC) operations required per filter output increases linearly with the order of filter, but implementation of higher order filters in real-time is another challenging task. Recently with the advent of software defined radio (SDR), the research has been concentrated on realization of FIR filters[1][2][3][4] mainly due to its high flexibility and low complexity[5][6][7]. The digit-based reconfigurable architecture presented in [3] provides a flexible and low power solution with a wide range of precision and tap length of FIR filters. Conventionally, the FIR filters are designed based on programmable multiply-accumulate [MAC] architecture [6] and systolic architecture [7]. The performances of the designs are analyzed in terms of hardware complexity, power consumption and throughput. In [6], the programmable MAC architectures consume low power with reduced supply voltage and it requires large area. In [7], even though systolic based architecture reduces the complexity, it increases the latency when the order of the filter increases. Several attempts have been made and it continued to develop low-complexity dedicated VLSI systems for these filters. There are several issues in the hardware implementation of Digital filters. The direct implementation of N-tap FIR filters which requires N MAC operations are too expensive to implement in hardware due to its logic complexity and area requirement. Distributed Arithmetic (DA) and LUT constitute memory-based techniques. Among them, DA is applied for inner product computation [8]-[17], LUT provides computation of multiplication [18]-[24]. In the LUT based approach, multiplications of input values with a fixed coefficient and consisting of all possible pre-computed product values corresponding to all possible values of input multiplicand, while in the DA-based approach, an LUT is used to store all possible values of inner-products of a fixed-point vector. If the inner-products are implemented directly, the memory-size of LUT-multiplier based implementation increases exponentially with the input word length, whereas the memory size of the DA-based approach increases exponentially with the inner-product-length.

It is observed that the memory-size reduction is achieved by such decompositions is accompanied by raise in latency as well as the number of adders and latches. In this paper FPGA realization of the two FIR filter architectures such as Odd multiple storage scheme [4] and MUX based multiplier are discussed.
Both Odd multiple storage scheme and MUX based multiplier have multiplier less architecture to reduce complexity in the design, by manipulating the odd multiples of the fixed coefficient in LUT design, and general multipliers replaced by shift and add operations respectively.

The performances of the proposed architectures are analyzed in terms of area and time complexities by varying the number of taps.

The rest of the paper is organized as follows. Section II describes the details of Multiplier less architecture design of FIR Filter and their implementation. The performance of the design are analyzed and discussed in Section III. Finally, Section IV concludes the paper.

Figure-1: General structure of an FIR filter

II. FIR FILTERS

In signal processing, the filter is used to remove some unwanted component or feature from a signal thereby improving the quality of signal. It alters the amplitude and/or phase characteristics of a signal in a desired manner with respect to frequency. The primary function of filter are – to confine a signal into a prescribed frequency band, to decompose a signal into two or more sub-bands, to modify the frequency spectrum of a signal and to model the input output relationship of a system. Filters are extensively used in signal processing and communication system in applications like noise reduction, echo cancellation, image enhancement, speech and waveform synthesis etc. There are two main kind of filter: analog and digital filter. Analog filter has analog signal at both its input & output and are made up from components such as resistors, capacitors and op amps to produce the required filtering effect. Such filters are fast and simple to realize but are little stable, sensitive to temperature variations and expensive to realize in large amounts.

Digital filter on the other hand uses digital processor to perform numerical calculations on sampled values of the signal and eliminate the problems associated with their classical analog counterparts, thus are preferably used in place of analog filter [1]. Broadly, digital filters are classified as: Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) filter. FIR filters have linear phase, stability, fewer finite precision errors, and efficient implementation hence preferred over IIR filter.

FINITE IMPULSE RESPONSE FILTER

Finite impulse response (FIR) filters are a class of digital filters that have a finite impulse response and are among one of the primary types of filter used in DSP and communication system [3]. They do not have any feedback and therefore if excited by impulse response, the output will invariably become zero. The input-output relationship of FIR filter is given by (1) $y[n] = (\sum_{k=0}^{N-1} p(k) x[n-k])$ (1) where, $p(k), k = 0,1,2,3……..N-1$ are the impulse response coefficients of the filter. N is the filter length that is number of coefficients.

TRANSPOSED DIRECT FORM OF AN FIR FILTER:

As shown in Fig. 1[1], FIR filtering operation performs the weighted summations of input sequences, called as convolution sum, which are frequently used to implement the frequency selective low-pass, high-pass, or band-pass filters. Generally, since the amount of computation and the corresponding power consumption of FIR filter are directly proportional to the filter order, if we can dynamically change the filter order by turning off some of multipliers, significant power savings can be achieved. However, performance degradation should be carefully considered when we
change the filter order.

Figure 2: Transposed direct form of an FIR filter

### III. EXISTING SYSTEMS

**MULTIPLIER LESS STRUCTURES BASED FOR FIR FILTER**

Figure 1 shows an implementation of N tap FIR filter with the usage of three elements namely adders, multipliers and delay elements. Let $X(n)$ and $Y(n)$ be the input and output sequences of the FIR filter respectively. Consider an N-tap FIR filter that can be formulated as

$$Y(n) = \sum_{k=0}^{N-1} h_k X(n-k)$$

where $h_k$ is the kth coefficient of the filter impulse response. In the above equation, the filter coefficients do not vary with regard to the input signal or the noise. This implies that the coefficients are fixed. Such filters whose coefficients are fixed are called as fixed filters or filters with fixed characteristics. In the filter the MAC structure and delay blocks are the main building blocks. The performance of the filter can be enhanced by the introduction of two architecture schemes which reduces the complexity and critical path. The two schemes discussed here are ODD and MUX based multiplier. A. Odd Multiple Storage Scheme This method proposed by [4] stores the product of odd multiple of coefficient and the input. The advantage of storing odd multiple is that even multiples can be obtained by a simple left shift operation. For any input vector of N bits the number of locations in the LUT is identified by $2^N$. However, the odd multiple storage scheme reduces the number of locations by a factor of 2. Considering an input vector of 8 bits, uses only 128($2^7$) unlike 256 locations of conventional LUT. Table I gives the complete insight about the look up table of odd multiple storage scheme. The 7-bit address (A6-A0) given in Table I identifies the stored values of LUT and it also gives a complete detail about the number of shifts to be performed for the corresponding input bits. Hence for the input bits'10' the LUT stored value A is shifted one time to get 2A. The control bits c2 cl c0 decides the number of shifts to be performed. The Figure 2 gives the proposed structure of the odd multiple storage schemes. The architecture has an 8 to 7 bit encoder which maps the 8-bit input word($x7:x0$) to a 7 bit address(A6:A0) according to the logical relations.
It chooses one output address value depending upon the output obtained from the encoder. The value obtained from the memory array is an odd multiple value of the co-efficient. In order to obtain the even multiple values, the shift operation is to be performed. The control unit generates the control bits according to the relations 3a to 3c.

![Figure-4-Proposed MUX based Multiplier architecture](image)

**Table- Mux Based Multiplier**

When the input word is 00000000 then the Reset (Rst) operation is performed by setting the reset bit which is given by the logical relation

\[ \text{Rst} = X \cdot \overline{X} \cdot \overline{X} \cdot \overline{X} \cdot \overline{X} \cdot \overline{X} \cdot \overline{X} \]

B. Proposed MUX based multiplier Figure 3 shows a shift and add multiplier implemented with the usage of namely adders, and MUX. Since the general multipliers are replaced by MUX, shifters and adders, referred to as multiplierless implementation. Consider an eight bit multiplication of signed numbers. Here the multiplication is carried out only with MUX and shifters. For example consider two signed 8-bit numbers x and c bits, represented in two's complement form.
IV. PROPOSED SYSTEMS

The proposed system was LUT based multiplierless FIR Filter which allows to reduce the maximum range of area so that the complexity will get reduced and increments the performance and computing will be very high. The following below are the results corresponding to proposed method.

a) RTL Block

b) RTL Logic Diagram

c) Waveform

d) Area Report

e) Delay Report

CONCLUSION

In this paper an efficient Low complexity based FIR filter architecture using MUX based multiplier and LUT based multiple scheme have been discussed and implemented effectively. The results of FIR filter architectures are analyzed and compared with respect to area and power. It is found that LUT multiple scheme occupies less area compared to MUX based FIR Filter architecture. Thus the proposed FIR filter architecture achieves low area and more flexibility and hence it is well suitable for VLSI implementation.

REFERENCES


[10] Xilinx Incorporation, "The Role of Distributed Arithmetic in FPGAbased signal Processing," Xilinx application notes, San Jose, CA


