International Journal of LNCT, Vol 2(4)
ISSN (Online): 2456-9895

Review of high speed integrated circuit of finite impulse response filter

Piyush Gour[1] and Santosh Jha[2]
M. Tech. Scholar, Department of Electronics and Communication, LNCT, Bhopal[1]
Assistant Professor, Department of Electronics and Communication, LNCT, Bhopal[2]

Abstract
This paper presents efficient modified distributed arithmetic (MDA)-based approaches for low delay reconfigurable implementation of finite impulse response (FIR) filters whose filter coefficients change during runtime. Conventionally, for reconfigurable DA-based implementation of FIR filter, the lookup tables (LUTs) are required to be implemented in ROM; and the ROM-based LUT is found to be costly for application specific integrated circuit (ASIC) implementation. Therefore, a shared-LUT design is proposed to realize the MDA computation. Instead of using separate registers to store the possible results of partial inner products for DA processing of different bit positions, registers are shared by the DA units for bit slices of different weightage. The all design implemented in Xilinx Virtex-5 FPGA device (XC5VSX95T-1FF1136).

Keywords
Finite impulse response (FIR), Look up table (LUT), Modified distributive arithmetic technique.

1.Introduction
Reconfigurable finite impulse response (FIR) filter whose filter coefficients change dynamically during runtime plays an important role in the software defined radio (SDR) systems [1], [2], multi-channel filters [3], and digital up/down converters [4]. However, the well-known multiple constant multiplication (MCM)-based technique [5] which is widely used for the implementation of FIR filters cannot be used when the filter coefficients change dynamically. On the other hand, general multiplier-based structure requires a large chip area, and consequently enforces limitation on the maximum possible order of the filter that can be realized for high-throughput applications. Distributed arithmetic (DA)-based technique [6] has gained substantial popularity, in recent years, for their high throughput processing capability and increased regularity which result in cost-effective and area-time efficient computing structures. The main operations required for DA-based computation are a sequence of lookup-table (LUT) accesses followed by shift-accumulation operations of the LUT output. The conventional DA implementation used for the implementation of FIR filter assumes that impulse response coefficients are fixed and this behavior makes it possible to use ROM-based LUTs. The memory requirement for DA-based implementation of FIR channels, be that as it may, increments exponentially with the channel arrange. To dispose of the issue of such vast memory prerequisite, systolic decay methods are proposed by Meher et al. for DA-based execution of long-length convolutions and FIR channel of substantial requests [7], [8]. For reconfigurable DA-based FIR channel whose channel coefficients change progressively, we have to utilize rewritable RAM based LUT [9] rather than ROM-based LUT. Another approach is to store the coefficients in the simple area by utilizing serial advanced to-simple converters bringing about blended flag engineering [10]. We likewise find many takes a shot at DA based usage of versatile filters [11], [12] where the coefficients change at every cycle. In this paper, we present efficient schemes for the optimized shared-LUT implementation of reconfigurable FIR filters using DA technique where LUTs are shared by the DA units for bit slices of different weightage. Also, the filter coefficients can be changed dynamically in runtime with very small reconfiguration latency. In the next section, we briefly discuss the mathematical background of DA-based implementation of FIR filter[13].

2.Literature review
Basant Kumar Mohanty et al. [1], transpose shape limited drive reaction (FIR) channels are intrinsically pipelined and bolster various steady increases (MCM) procedure that outcomes in noteworthy sparing of calculation. Be that as it may, transpose shape design does not specifically bolster the piece handling not at all like direct frame arrangement. In this paper, we investigate the likelihood of acknowledgment of square FIR channel in transpose frame design for territory defer effective acknowledgment of huge request FIR channels for both settled and reconfigurable applications. In view of a nitty gritty computational examination of transpose frame design of FIR channel, we have determined a stream diagram for transpose shape square FIR channel with enhanced enlist unpredictability. A summed up piece plan is displayed for transpose shape FIR channel.

We have determined a general multiplier-based engineering for the proposed transpose frame square channel for reconfigurable applications. A low-multifaceted nature configuration utilizing the MCM plot is likewise displayed for the square execution of settled FIR channels. The proposed structure involves significantly less area delay product (ADP) and less energy per sample (EPS) than the existing block implementation of direct-form structure for medium or large filter lengths, while for the short-length filters, the block implementation of direct-form FIR structure has less ADP and less EPS than the proposed structure. Application specific integrated circuit synthesis result shows that the proposed structure for block size 4 and filter length 64 involves 42% less ADP and 40% less EPS than the best available FIR filter structure proposed for reconfigurable applications.

In to this concise proposes a two-advance enhancement procedure for planning a reconfigurable VLSI engineering of an interjection channel for multi standard computerized up converter (DUC) to diminish the power and territory utilization. The proposed method at first decreases the quantity of increases per input test and augmentations per input test by 83% in examination with singular usage of every standard's channel while planning a root-raised-cosine limited drive reaction channel for multi standard DUC for three unique norms. In the following stage, a 2-bit parallel normal subexpression (BCS)- based BCS end calculation has been proposed to outline an effective steady multiplier, which is the fundamental component of any channel. This strategy has prevailing with regards to lessening the region and power use by 41% and 38%, separately, alongside 36% change in working recurrence over a 3-bit BCS-based procedure detailed prior, and can be viewed as more fitting for outlining the multi standard DUC[2].

Sang Yoon Park et al. [3], this paper presents effective dispersed number juggling (DA)- based methodologies for high-throughput reconfigurable execution of limited drive reaction (FIR) channels whose channel coefficients change amid runtime. Ordinarily, for reconfigurable DA-based execution of FIR channel, the query tables (LUTs) are required to be actualized in RAM; and the RAM-based LUT is observed to be expensive for ASIC usage. In this manner, a common LUT configuration is proposed to understand the DA calculation. Rather than utilizing separate registers to store the conceivable aftereffects of fractional inward items for DA preparing of various piece positions, registers are shared by the DA units for bit cuts of various weightage. The proposed configuration has about 68% and 58% less zone defer item, and 78% and 59% less vitality for each example than DA-based systolic structure and convey spared viper (CSA)- based structure, individually for the ASIC usage. A DRAM-based outline is additionally proposed for the FPGA execution of the reconfigurable FIR channel which underpins up to 91 MHz input examining recurrence, and offers 54% and 29% less the quantity of cuts than the systolic structure and the CSA-based structure, separately when actualized in Xilinx Virtex-5 FPGA gadget (XC5VSX95T-1FF1136).

Basant K. Mohanty et al. [4], we have analyzed memory footprint and combinational complexity to arrive at a systematic design strategy to derive area-delay-power-efficient architectures for two-dimensional (2-D) finite impulse response (FIR) filters. We have presented novel block based structures for separable and non-separable filters with less memory footprint by memory sharing and memory-reuse along with appropriate scheduling of computations and design of storage architecture. The proposed structures involve times less storage per output (SPO), and nearly times less energy consumption per output (EPO) compared with the existing structures, where is the input block-size. They involve times more arithmetic resources than the best of the corresponding existing structures, and produce times more throughput with less memory band-width (MBW) than others. We have also proposed separate generic structures for separable and non-separable filter-banks, and a unified structure of filter-bank constituting symmetric and general filters. The proposed unified structure for 6 parallel filters involves nearly times more multipliers, times more adders, less registers than similar existing unified structure, and computes times more filter outputs per cycle with times less MBW than the existing design, where is FIR filter size in each dimension. ASIC synthesis result shows that for filter size (4 4), input-block size, and image-size (512 512), proposed block-based non-separable and generic non-separable structures, respectively, involve 5.95 times and 11.25 times less area-delay-product (ADP), and 5.81 times and 15.63 times less EPO than the corresponding existing structures. The proposed unified structure involves 4.64 times less ADP and 9.78 times less EPO than the corresponding existing structure.

Basant K. Mohanty et al. [5], in this paper, we present an efficient distributedarithmetic (DA) formulation for the implementation of block least mean square (BLMS) algorithm. The proposed DA-based outline utilizes a novel look-into table (LUT)-sharing method for the calculation of channel yields and weight-increase terms of BLMS calculation. Additionally, it offers noteworthy sparing of adders which constitute a noteworthy part of DA-based structures. Additionally, we have recommended a novel LUT-based weight refreshing plan for BLMS calculation, where just a single arrangement of LUTs out of sets should be altered in each emphasis, where , and are, individually, the channel length and info piece measure. In view of the proposed DA detailing, we have determined a parallel design for the execution of BLMS versatile computerized channel (ADF). Contrasted and the best of the current DA-based LMS structures, proposed one includes almost times adders and times LUT words, and offers nearly times throughput of the other. It requires nearly 25% more flip-flops and does not involve variable shifters like those of existing structures. It involves less LUT access per output (LAPO) than the existing structure for block-size higher than 4. For block-size 8 and filter length 64, the proposed structure involves 2.47 times more adders, 15% more flip-flops, 43% less LAPO than the best of existing structures, and offers 5.22 times higher throughput. The number of adders of the proposed structure does not increase proportionately with block size; and the number of flip-flops is independent of block-size. This is a major advantage of the proposed structure for reducing its area delay product (ADP); particularly, when a large order ADF is implemented for higher block-sizes. ASIC synthesis result shows that, the proposed structure for filter length 64, has almost 14% and 30% less ADP and 25% and 37% less EPO than the best of the existing structures for block size 4 and 8, respectively.

3.Distributive arithmetic technique

Distributed Arithmetic (DA) is a widely-used technique for implementing sum-of-products computations without the use of multipliers. Designers frequently use DA to build efficient Multiply-Accumulate Circuitry (MAC) for filters and other DSP applications. The main advantage of DA is its high computational efficiency. DA distributes multiply and accumulates operations across shifters; lookup tables (LUTs) and adders in such a way that conventional multipliers are not required.

Distributed arithmetic is an important algorithm for DSP applications. It is based on a bit level rearrangement of the multiply and accumulate operation to replace it with set of addition and shifting operations. The basic operations required are a sequence of table lookups, additions, subtractions and shifts of the input data sequence. The Look Up Table (LUT) stores all possible partial products over the filter coefficient space.

Assuming coefficients c[n] is known constants, and then y[n] can be rewritten as follows:

$$y[n] = \Sigma\, c[n] \cdot x[n] \quad n = 0, 1, \ldots, N\text{-}1 \tag{1}$$

Variable x[n] can be represented by:

$$x[n] = \Sigma\, x_b[n] \cdot 2^b \qquad b=0, 1, \ldots, B\text{-}1$$
$$x_b[n] \in [0, 1] \tag{2}$$

Where x_b [n] is the b^{th} bit of x[n] and B is the input width. Finally, the inner product can be rewritten as follows:

$$y = \Sigma\, c[n] \Sigma\, x_b[k] \cdot 2^b \tag{3}$$

$$\begin{aligned} &= c[0]\,(x_{B-1}[0]2^{B-1} + x_{B-2}[0]\,2^{B-2} + \ldots + x_0[0]\,2^0) \\ &\quad + c[1]\,(x_{B-1}[1]\,2^{B-1} + x\,x_{B-2}[1]\,2^{B-2} + \ldots + x_0[1]\,2^0) + \ldots \\ &\quad + c[N\text{-}1]\,(x_{B-1}[N\text{-}1]\,2^{B-1} + x_{B-2}[0]\,2^{B-2} + \ldots + x_0[N\text{-}1]\,2^0) \end{aligned} \tag{4}$$

$$\begin{aligned} &= (c[0]\,x_{B-1}[0] + c[1]\,x_{B-1} + \ldots + c[N\text{-}1]\,x_{B-1}[N\text{-}1])\,2^{B-1} + (c[0]\,x_{B-2}[0] \\ &\quad + c[1]\,x_{B-2}[1] + \ldots + c[N\text{-}1]\,x_{B-2}[N\text{-}1])\,2^{B-2} + \ldots + (c[0]\,x_0[0] + \\ &\quad c[1]\,x_0[1] + \ldots + c[N\text{-}1]\,x_0[N\text{-}1])\,2^0 \end{aligned} \tag{5}$$

$$= \Sigma\; 2^b\, \Sigma\, c[n] \cdot x_b[k]$$

Where n=0, 1… N-1 and b=0, 1… B-1
The coefficients in most of DSP applications for the multiply accumulate operation are constants.

4.Proposed methodology

The above strategy holds great just when we go for bring down request channels. For higher request channels, the extent of the LUT additionally increments exponentially with the request of the channel. For a channel with N coefficients, the LUT have 2N esteems. This thusly diminishes the execution. it to square transpose shape compose I arrangement of piece FIR channel. The DFG-3 can be retimed to acquire the DFG-4 of Figure 1, which is alluded to square transpose frame compose II arrangement.

Note that both sort I and sort II setups include a similar number of multipliers and adders, however type-II design includes almost L times less postpone components than those of sort I arrangement. We have, along these lines, utilized piece transpose shape compose II design to determine the proposed structure. In Section II-C, we introduce numerical detailing of square transpose shape compose II FIR channel for a summed up definition of the idea of piece based calculation of transpose frame FIR filers.

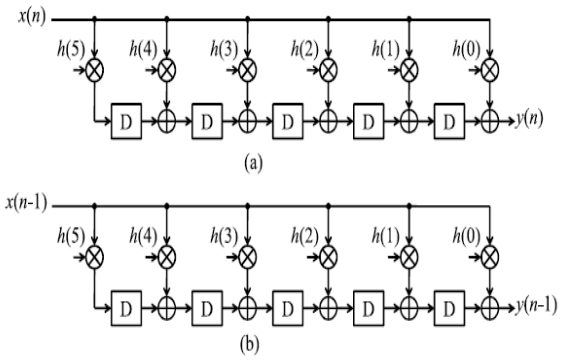

Figure 1 DFG of transpose form structure for $N = 6$. (a) DFG-1 for output $y(n)$. (b) DFG-2 for output $y(n-1)$

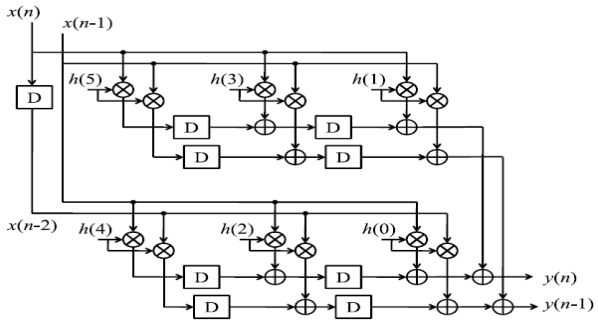

Figure 2 Merged DFG (DFG-3: transpose form type-I configuration for block FIR structure)

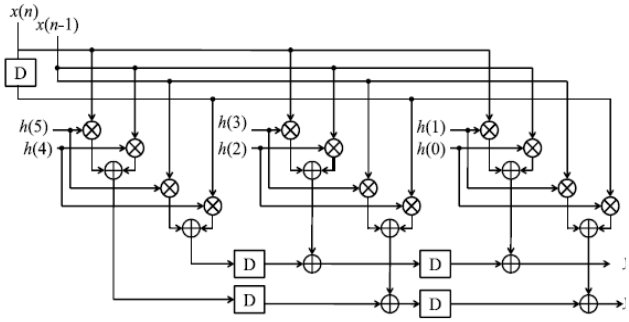

Figure 3 DFG-4 (retimed DFG-3) transpose form type-II configuration for block FIR structure

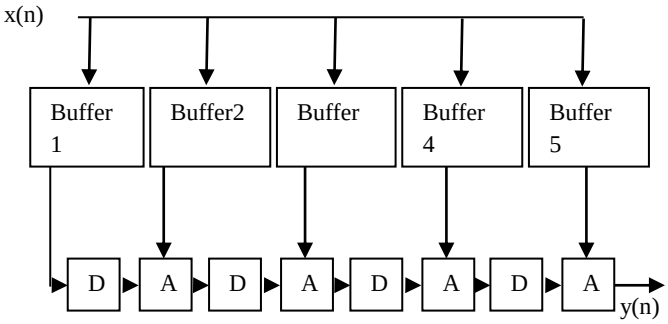

Figure 4 Block diagram of proposed methodology

5.Simulation result

The above strategy holds great just when we go for bring down request channels. For higher request channels, the measure of the LUT additionally increments exponentially with the request of the channel. For a channel with N coefficients, the LUT have 2N esteems. This thusly decreases the execution.

6.Conclusion

In this paper, we have investigated the likelihood of acknowledgment of piece FIR channels in transpose frame design for region postpones proficient acknowledgment of both settled and reconfigurable applications. A summed up piece definition is exhibited for transpose shape square FIR channel, and in light of that we have determined transpose frame piece channel for reconfigurable applications.
Limited Impulse Response channel assumes a vital part in numerous Digital Signal Processing applications. In this strategy, the multiplier less FIR channel is actualized utilizing Distributed Arithmetic which comprises of Look Up Table and afterward parceling is included. This engineering gives an effective region time control execution which includes altogether less idleness and less region defer intricacy when contrasted and existing structures for FIR Filter.

References

[1] Mohanty BK, Meher PK. A high-performance FIR filter architecture for fixed and reconfigurable applications. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2016; 24(2):444-52.
[2] Hatai I, Chakrabarti I, Banerjee S. An efficient VLSI architecture of a reconfigurable pulse-shaping FIR interpolation. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2015; 23(6):1150-4.
[3] Park SY, Meher PK. Efficient FPGA and ASIC realizations of a DA-based reconfigurable FIR digital filter. IEEE Transactions on Circuits and Systems II: Express Briefs. 2014; 61(7):511-5.

[4] Mohanty BK, Meher PK, Al-Maadeed S, Amira A. Memory footprint reduction for power-efficient realization of 2-D finite impulse response filters. IEEE Transactions on Circuits and Systems I: Regular Papers. 2014; 61(1):120-33.
[5] Mohanty BK, Meher PK. A high-performance energy-efficient architecture for FIR adaptive filter based on new distributed arithmetic formulation of block LMS algorithm. IEEE Transactions on Signal Processing. 2013; 61(4):921-32.
[6] Gokhale GR, Bahirgonde PD. Design of Vedic-multiplier using area-efficient carry select adder. In advances in computing, communications and informatics (ICACCI), international conference on 2015 (pp. 576-81). IEEE.
[7] Gokhale GR, Gokhale SR. Design of area and delay efficient Vedic multiplier using Carry Select Adder. In Information Processing (ICIP), International conference on 2015 (pp. 295-300). IEEE.
[8] Kumar UP, Goud AS, Radhika A. FPGA implementation of high speed 8-bit vedic multiplier using barrel shifter. In energy efficient technologies for sustainability (ICEETS), international conference on 2013 (pp. 14-17). IEEE.
[9] Kumar AA, Basha SM. Design and implementation of high speed 8 bit Vedic multiplier on FPGA.2015.
[10] Murali A, Padma GV, Saritha T. An optimized implementation of vedic multiplier using barrel shifter in FPGA technology. Journal of Innovative Engineering. 2014; 2(2).
[11] Khatri S, Jangid G. FPGA Implementation of 64-bit Fast multiplier using barrel shifter. International Journal for Research in Applied Science and Engineering Technology. 2014; 2:344-8.
[12] Billore TJ, Rotake DR. FPGA implementation of high speed 8 bit vedic multiplier using fast adders. Journal of VLSI and Signal Processing. 2014; 4(3):54-9.
[13] Kerur SS, Prakash Narchi JC, Kittur HM, Girish VA. Implementation of Vedic multiplier for digital signal processing. In international conference on VLSI, Communication & Instrumentation (ICVCI) 2011 (pp. 1-6).