

MULTIPLE CONSTANT MULTIPLICATION TECHNIQUE FOR CONFIGURABLE FINITE IMPULSE RESPONSE FILTER DESIGN

*Madipalli sumalatha Assistant professor, ECE.
Siddhartha Institute of Technology & Sciences, Hyderabad.*

Abstract--- This paper highlights the design of finite impulse response (FIR) filter with the help of an evolutionary optimization technique, particularly multiple constant multiplication. The Finite Impulse Response (FIR) Filter is that the necessary part for coming up with an associated economical digital signal processing system. Transpose kind finite impulse response (FIR) filter is of course a pipelined structure that supports the multiple constant multiplications (MCM) technique however direct form FIR filter structure doesn't support MCM technique. The MCM is more practical in Transpose form once the common operand is multiple with the set of constant coefficient that cut back the procedure delay. The implementation of MCM technique is simpler in fixed coefficient Transpose form FIR filter however complicated in configurable coefficient. For this purpose, the coefficient of the filter are encoded by Canonical Signed Digit (CSD). FIR filter supported canonical signed digits illustration of coefficient so as to attenuate the power consumption and quick implementation of the filter. Performance of the proposed filter has been analyzed in terms of its area, power and speed. The look of FIR filter is planned to implement using Xilinx tool.

Keywords--- Multiple constant multiplications (MCM), Canonical Signed Digit (CSD), FIR filter.

I. INTRODUCTION

Finite Impulse Response (FIR) computerized channel is loosely utilized in a very digital signal processing applications, as an example, speech processing, loud speaker equalization, echo cancellation, reconciling noise cancellation, and varied communication applications, as well as software-defined radio (SDR) etc. vast numbers of those applications need FIR channels of considerable request to fulfill the demanding frequency specifications. All the time these channels ought to facilitate high sampling rate for quick advanced correspondence. The amount of multiplications and additions needed for every filter output, yet, increments directly within the filter order. Since there's no excess calculation accessible in the FIR filter algorithm, in progress usage of a considerable request FIR filter in associated quality obliged condition may be a testing errand. Filter coefficient all the time stay steady and better-known a priori in signal processing applications. This feature has been utilized to scale back the complexity of

realization of multiplications. design for economical realization of FIR filters multiple constant multiplication (MCM) strategies. The MCM strategy lessens the number of additions required for the realization of multiplications by common sub expression sharing, once a given input is multiply with a meeting of constants. The MCM conspire is more successful, once a typical operand is accumulated with additional range of constants. Hence, the MCM conspire is suitable for the usage of vast request FIR filters with fixed coefficient. In any case, MCM block is formed simply within the transpose configuration of FIR filters.

The derivation of block-based FIR structure is simply once direct-form configuration is used, although the transpose form configuration doesn't foursquare support block making ready. Be that because it might, to require the procedure most well-likes posture of the MCM, FIR channel is needed to be acknowledged by transpose form configuration. There are a many applications, as an example, SDR channelizer, wherever FIR filters is ought to be actual in a very reconfigurable instrumentation to assist multi normal remote correspondence. Many structures are counseled for economical realization of FIR utilizing constant multiplication schemes. a canonic sign digit (CSD)-primarily based FIR filter, wherever the nonzero CSD values square measure altered to minimize the accuracy of filter coefficient while not essential impactation filter conduct.

In this paper, we have a tendency to explore the likelihood of realization of block FIR filter in transpose form configuration so as to require advantage of the MCM schemes for configurable applications. The rest of this paper is organized as follows. In Section II, FIR filter are presented. Multiple Constant Multiplication (MCM) approach is presented in Section III. The proposed architectures for configurable applications are presented in Section IV. Finally, the conclusion is drawn in Section V.

II. FIR FILTER

Filters will primarily, signals break and restoration. These tasks are necessary for signals processing. Break is appropriate if interference of noise or additional signals pollutes signals and apology is accessible if a signal is askew by some suggests that. Of these tasks crave absolute frequency specifications to get rid of noise which might be accomplished with higher order of FIR filter. However with

accretion order, the addition operations endure accretion arch to complication in calculation. FIR clarify has propulsion acknowledgment of certain continuance as a result of it resolves to null in certain time. FIR clarify action is annihilation however sum of past, present and maybe approaching outputs. The equation of FIR clarify is declared by,

$$y(n) = \sum_{k=0}^{N-1} x(n-k)h(k)$$

Where $x(n)$ is input sample, $Y(n)$ is output sample, $h(k)$ is filter coefficient, N is filter order. Eq.1 represents FIR filter of order N as each value of output is weighted sum of most recent input values. The appellation $x(n-k)$ in (1) represents delayed input to multiplication process generally called taps which define order of filter in altered forms of FIR filter structures.

A FIR Filter Structure

The basic building blocks of FIR filter are multiplier, adders and signal delay. Multipliers ought to be quick enough in order that overall throughput shouldn't suffer. Adders are utilized in combination of multipliers and delays are used to store sample value in memory for one sample clock cycle. Blocks conjointly ought to be able to also need to be able to store filter constant in memory. The most usually used structures for filter are of two sorts, direct form and transposed form.

- Direct form
 - Several signals multiplied by constants and additional later on.
 - For accomplishing high throughput and reduction in delay of adder it needed further pipeline register between adders.

The data-flow graphs (DFG) of direct form FIR filter for filter length $N=6$ as shows below.

- Transposed form
 - Already have register between adders.
 - It gains high throughput without additional pipeline registers.
 - Filter suitable for MCM operations as single input is multiplied with number of constants.
 - It offers higher operating frequency to support higher sampling rate.

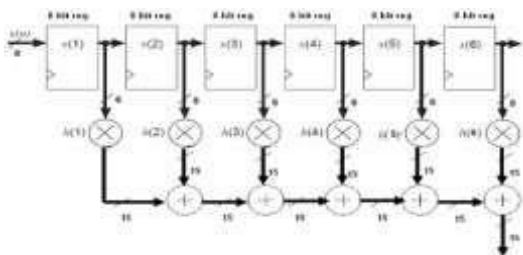


Fig.1. DFG 1 of direct form structure for $N=6$.

The data-flow graphs (DFG) of transpose form FIR filter for filter length $N=6$ as shows below.

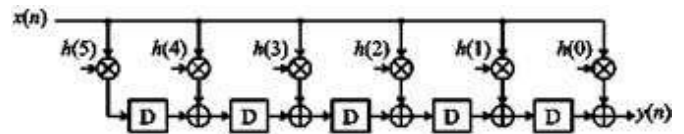


Fig.2. DFG 2 of transpose form structure for $N=6$. For output $y(n)$

The data-flow graphs (DFG) of transpose form FIR filter for filter length $N=6$ as shows below.

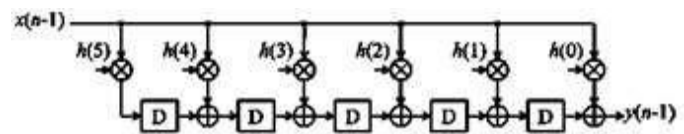


Fig.3. DFG 3 of transpose form structure for $N=6$. For output $y(n-1)$

The block transpose form configuration of block FIR filter (filter length, $L=2$ and filter order, $N=6$).

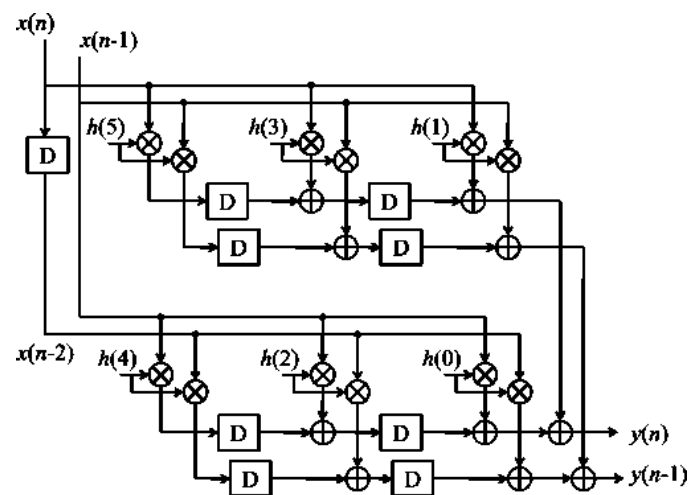


Fig.4. DFG 4 of transpose form structure for $L=2$ and $N=6$.

III. MCM TECHNIQUE

In any FIR filter, the multiplier is the major constraint that defines the performance of the required filter. Therefore, over the past 3 decades, style of associate economical hardware design for fixed point FIR filter has been thought-about because the major analysis focus as reported in published literatures. To get optimized solutions, the first coefficients are multiplied by constant value. In FIR filter, the multiplication operation is performed between one explicit variable (the input) and plenty of constants (the coefficient). This can be done by using multiplier block referred to as Multiple Constant Multiplication (MCM) followed by accumulation of all products.

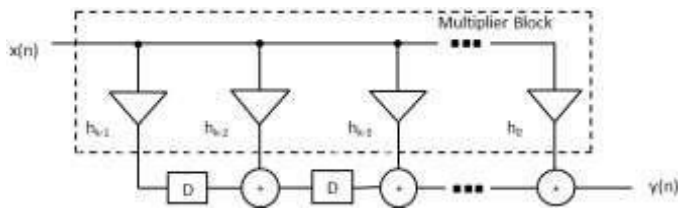


Fig.5 .FIR filter implementations of Transposed form with generic multipliers

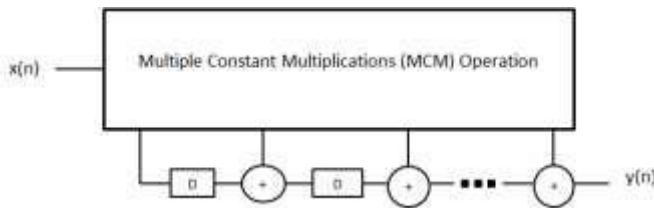


Fig.6 Transposed form with MCM block

The transposed-form finite impulse response (FIR) filter implementations are illustrated in Fig.5 correspondingly. Though each architectures have similar difficulties in hardware, the transposed type is mostly most popular attributable to its higher performance and power potency

The multiplication of filter coefficients with filter inputs is recognized has major effect on the complexity and performance of the look as a result of an oversized variety of constant multiplication are needed. This is often ordinarily recognized because the multiple constant multiplication operation. Even though area, delay and power economical multiplier architecture are projected, the total flexibility of a multiplier isn't necessary for the constant multiplications, since filter coefficients are mounted and determined beforehand by the CSD algorithmic rule. Thence filter coefficients multiplication with input file is mostly enforced underneath a typical sub expression constant sharing wherever every constant multiplication expression constant sharing wherever every constant multiplication is realized victimization CSD operation in an MCM operation.

IV. EXISTING METHOD

The design of block FIR filter for Reconfigurable applications is shown within the Fig.7 for block size L . The most blocks are one Register Unit (RU), one Coefficient Storage Unit (CSU), one Pipeline Adder Unit (PAU), and M number of Inner Product Units (IPUs).

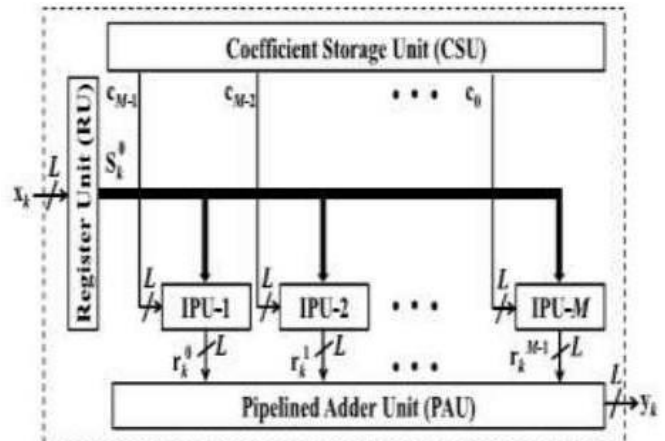


Fig.7. Block of reconfigurable coefficient FIR filter (L, M).

a. Coefficient Storage Unit (CSU)

The Coefficient Storage Unit (CSU) is employed to store the coefficient of all the filters. These coefficients are utilized in the reconfigurable applications. It's N Read Only Memory (ROM) Lookup Tables (LUTs) where N is that the length of the filter ($N=ML$).

b. Register Unit (RU)

The Register Unit (RU) is employed for storing the input samples is shown in Fig.8. It contains $(L-1)$ registers. Throughout the cycle, the register unit accepts input sample and computes L rows of in parallel. The outputs from the RU are given as inputs to M Inner Product Units.

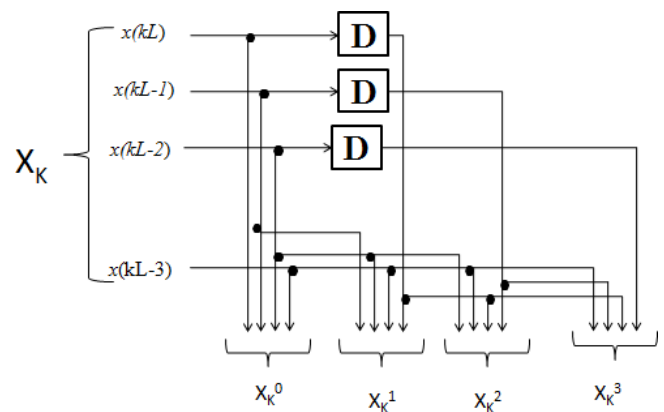


Fig.8. Internal structure of Register Unit (RU) for block size $L=4$.

c. Inner Product Unit (IPU)

The Inner Product Unit (IPU) is used to perform a multiplication operation of with the small weight vector cm is shown in Fig.9.

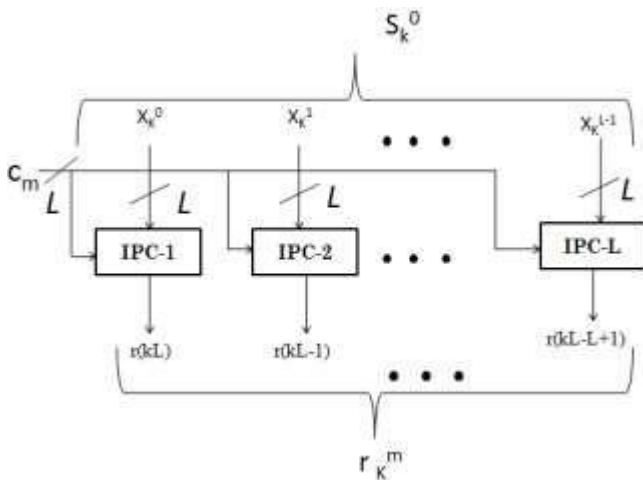


Fig.9. Internal structure of () IPU.

The M Inner Products Units accepts L rows of from the RU and M small weight vectors from the CSU. Every Inner Product Unit contains L number of Inner Product Cells (IPCs) which performs L inner product computations of L rows of with constant vector c_m and produces a block of L number of partial inner products. All the four IPU's work at the same time and M blocks of a result are obtained.

d. Inner Product Cell (IPC)

The internal structure of $(l+1)$ th IPC is shown in Fig.10. The Inner Product Cell (IPC) accepts $(l+1)$ th row of S_k^0 and small weight vector c_m and manufacture a partial result of inner product $r(kL-l)$, for $0 \leq l \leq L-1$. Every IPC consists of L multipliers and $(L-1)$ number of adders

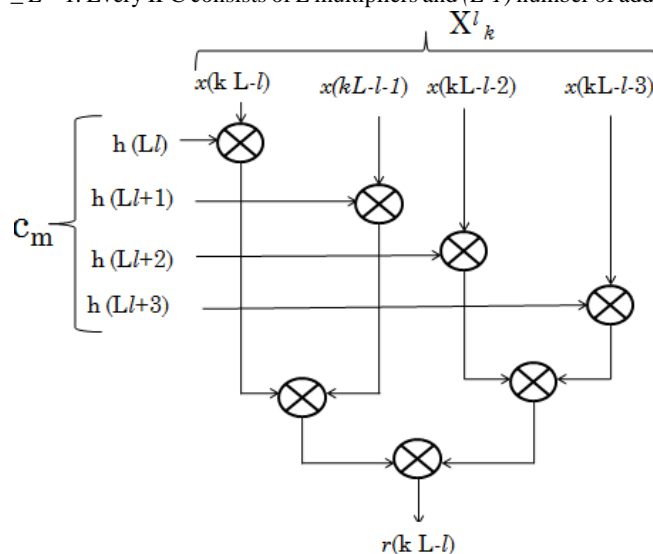


Fig.10. Internal structure of $(l+1)$ th IPC.

e. Pipelined Adder Unit (PAU)

The Pipelined Adder Unit (PAU) receives partial products from all the M IPU's is shown in Fig.11.

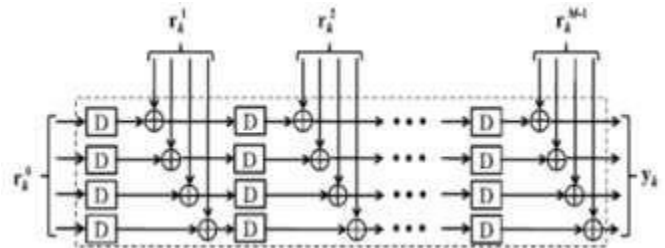


Fig.11. Internal structure of Pipelined Adder Unit (PAU)

The PAU involves $(M-1)$ adders and the same number of registers, wherever every register includes breadth of $(B+ B_-)$, B , and B_- respectively, being the bit width of input sample and filter coefficient.

V. PROPOSED METHODOLOGY

In this section, we have a tendency to present a structure of block FIR filter for such configurable applications. During this section, we have tendency to discuss the implementation of block FIR filter, hence filter coefficients multiplication with input file is mostly enforced underneath a typical sub expression constant sharing wherever every constant multiplication is realized victimization CSD operation in an MCM operation.

a. Canonical signed digit

CSD is a remarkable resolution in implementing economical multipliers, significantly once multiplying are signaling by a continuing multiplicand.

CSD is a representation in which the number of non-zero digits becomes minimal. To achieve this CSD uses a ternary number system which uses a new digit -1 denoted by $\bar{1}$. it uses both additions and subtractions. This provides additional flexibility in implementing arithmetic functions. It is particularly helpful in dealing with numbers which have a string of 1s. In these cases CSD replaces several shift and add operations resulting from the string of 1s with two shifts and one subtract operation. a CSD number is quicker than a standard binary number. However, before mistreatment CSD illustration we want to convert the number from binary to CSD. once one in all the multiplicands is fastened, we are able to simply calculate the CSD equivalent of the constant and implement it.

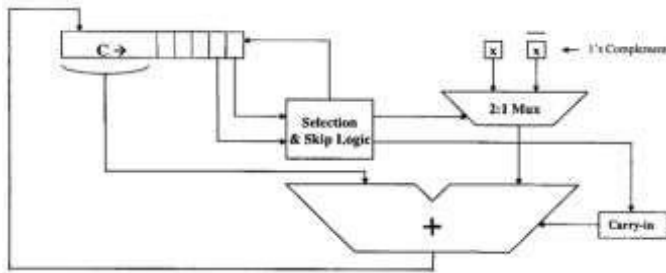


Fig.12 Proposed canonical signed digit multiplier.

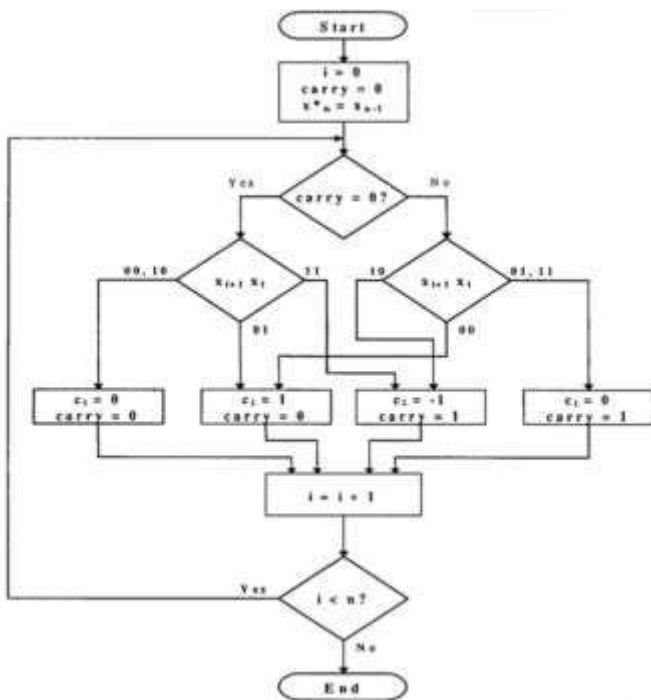


Fig.13 The algorithm flowchart to convert binary number to CSD.

b. Coefficient multiplication

This methodology uses common sub-expression sharing algorithm across all multiplications that cut back range of arithmetic operations to calculate inner products. This eventually reduces procedural complexity. Coefficient are represented in Canonical Sign Digit form (CSD) that's utilized in multiplication. In CSD, every digit takes the value from $\{-1, 0, 1\}$. In add shift network, every digit takes the value from $\{-1, 0, 1\}$. In add shift network, every non-zero bit represent addition or subtraction. Here 1 represents addition and -1 represent subtraction. The CSD illustration has around 32% fewer non-zero bits, thus range of operation needed for multiplications are reduced. Further complexity of multiplication is reduced by identifying common sub-expression from constant coefficient. A Pattern consisting of two non-zero digits having form like $\{100\dots 1\}$ or $\{100\dots 0-1\}$ are known and computation of all occurrences of constant sequence shared by single sub graph to cut back computation in MCM.

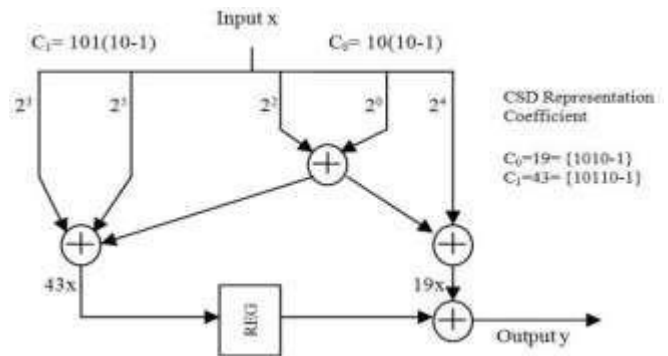


Fig.14 MCM design using common sub-expressions elimination by CSD representation of coefficient

Fig.14 represents example of common sub-expression elimination for MCM of FIR filter. Firstly, coefficient are expressed in CSD representation to scale back add/shift operations. Then common sub-expressions from the like coefficient are identified and shared. Like as in Fig.1 $\{10-1\}$ is common sub expression shared to reduce number of add/shift operations. This helps in reducing MCM procedural computational complexity. This methodology uses L method block FIR filter acceptable L input samples and giving L output samples in parallel.

c. Proposed design

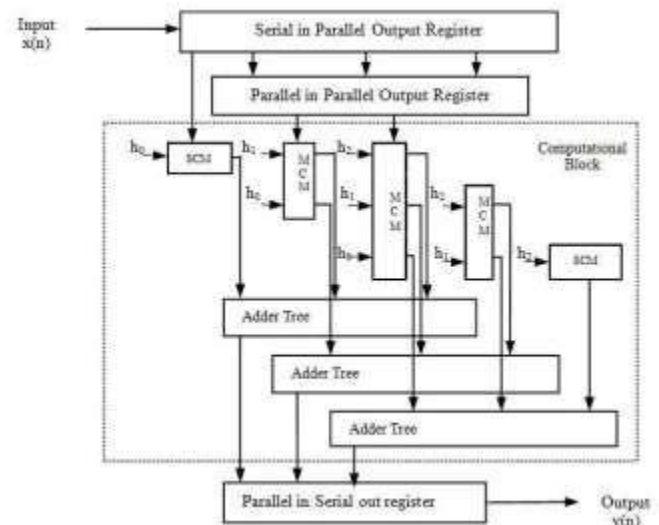


Fig.15 Design of 4 way 3-tap FIR filter

Fig.15 represents 3-way tap FIR filter. Here input is provided to Serial Input Parallel Output (SIPO) register then this input is provided to MCM block wherever coefficient are multiplied with input. Then these are summed in adder tree. Finally, output is taken out through PISO register. During this technique, new input samples $x(Lk-n)$ are applied where n ranges from $n=0$ to $N-1$. At the same time, recent input samples are applied for computation. Therefore at finish of computation filter output $y(Lk+n)$ is calculated wherever L is block length. It is ascertained that input samples apart from previous and current one are multiplied at same time by multiple

coefficient which specify utilization of multiple constant multiplication design methodology in computation.

VI. CONCLUSION

Thus in our project, by the implementation of configurable finite impulse response within the canonical signed digit algorithm exploitation multiple constant multiplication, the delay is reduced most, and the power is additionally reduced the maximum amount as doable. The CSD algorithmic program is given for the development of shift and adds implementation of the constant multiplications. The CSD formula reduces the number of shift and adds operation. From the projected technique by implementing the configurable finite impulse response within the canonical signed digit algorithm, as shown in flow chart, the delay is reduced and to realize higher speeds and power is reduced. The projected algorithmic rule are synthesized and enforced victimization Xilinx 10.1.

REFERENCES

1. Basant K. Mohanty, Senior Member, IEEE, Pramod K. Meher, Senior Member, IEEE, Somaya Al-Maadeed, Senior Member, IEEE, and Abbas Amira, Senior Member, IEEE "Memory Footprint Reduction for Power-Efficient Realization of 2-D Finite Impulse Response Filters" IEEE Transactions On Circuits And Systems—I: Regular Papers, Vol. 61, No. 1, January 2014
2. Xin Lou, Student Member, IEEE, YaJun Yu, Senior Member, IEEE, and Pramod Kumar Meher, Senior Member, IEEE "Analysis and Optimization of Product-Accumulation Section for Efficient Implementation of FIR Filters" IEEE Transactions On Circuits And Systems—I: Regular Papers 2016 IEEE
3. Georgina Binoy Joseph, R. Devanathan, "A Mathematical Framework for Online Constant Coefficient Multiplication" International Journal of Engineering and Technology Innovation, vol. 7, no. 3, 2017, pp. 217 - 228
4. Kenny Johansson, Oscar Gustafsson, Linda S. DeBrunner, and Lars Wanhammar, "Minimum Adder Depth Multiple Constant Multiplication Algorithm for Low Power FIR Filters" 2011 IEEE
5. Mathias Faust and Chip-Hong Chang, "Bit-Parallel Multiple Constant Multiplication using Look-Up Tables on FPGA" 2011 IEEE
6. Mamta N. Ahuja Prachi Palsodkar "Design Of Efficient Add/Shift Algorithm For Multiple Constant Multiplication" 2015 Fifth International Conference on Communication Systems and Network Technologies
7. Basant Kumar Mohanty, Senior Member, IEEE, and Pramod Kumar Meher, Senior Member, IEEE "A High-Performance FIR Filter Architecture for Fixed and Reconfigurable Applications" IEEE Transactions On Very Large Scale Integration (VLSI) Systems, Vol. 24, No. 2, February 2017
8. Jongsun Park, Woopyo Jeong, Hamid Mahmoodi-Meimand, Student Member, IEEE, Yongtao Wang, Hunsoo Choo, and Kaushik Roy, Fellow, IEEE "Computation Sharing Programmable FIR Filter for Low-Power and High-Performance Applications" IEEE Journal Of Solid-State Circuits, Vol. 39, No. 2, February 2004
9. Tero Rissa, Riku Uusikartano, and Jarkko Niittylahti "Adaptive FIR Filter Architectures for Run-Time Reconfigurable FPGAs" 2002 IEEE
10. Zhangwen Tang, Jie Zhang and Hao Min "A High-Speed, Programmable, Csd Coefficient Fir Filter" IEEE Transactions on Consumer Electronics, Vol. 48, No. 4, November 2002
11. Manish B. Trimale Chilveri "A Review: FIR Filter Implementation" 2017 IEEE International Conference On Recent Trends In Electronics Information & Communication Technology, May 19-20, 2017, India
12. Deepa Yagain and A. Vijaya Krishna "Design of Synthesizable, Retimed Digital Filters Using FPGA Based Path Solvers with MCM Approach: Comparison and CAD Tool" Hindawi Publishing Corporation VLSI Design Volume 2014, Article ID 280701, 18 pages <http://dx.doi.org/10.1155/2014/280701>
13. Mitsuru Yamada Akinori Nishihara "High-speed FIR Digital Filter with CSD Coefficients Implemented on FPGA" 2001 IEEE.
14. Levent Aksoy, Student Member, IEEE, Eduardo da Costa, Paulo Flores, Member, IEEE, and José Monteiro, Member, IEEE "Exact and Approximate Algorithms for the Optimization of Area and Delay in Multiple Constant Multiplications" IEEE Transactions On Computer-Aided Design Of Integrated Circuits And Systems, Vol. 27, No. 6, June 2008.