



Area-Delay Efficient Reconfigurable FIR Filter Architecture

Nisha C.K.¹, Pramod P.²

Student, M.Tech, VLSI Design & Signal Processing, LBS College of Engineering, Kasaragod, Kerala, India¹

Assistant Professor, ECE Department, LBS College of Engineering, Kasaragod, Kerala, India²

Abstract: Transpose form finite impulse response (FIR) filter based on block formulation method can be used for reconfigurable applications. This reconfigurable FIR filter architecture realization is area, delay and power efficient. Transpose form finite impulse response filters are inherently pipelined and support multiple constant multiplication (MCM) technique that results in significant saving of computation. By block formulation method data samples in fixed size blocks are processed consecutively. General multiplier based architecture for transpose form configuration of filter, for which efficient multiplication can be performed by dadda multiplier in terms of area, delay and power. Reconfigurable FIR filter is implemented using VHDL language by Xilinx software.

Keywords: Transpose form, Block formulation, Reconfigurable, Multiple Constant Multiplication, Dadda multiplier.

I. INTRODUCTION

Finite impulse response (FIR) digital filter have wide range of digital signal processing applications, such as speech processing, loud speaker equalization, echo cancellation and communication applications including Software Defined Radio (SDR). These applications require FIR filters of large order to meet frequency specifications and need to support high sampling rate for high speed digital communication. When FIR filter order increases the number of multiplication and addition also increases linearly for obtaining filter output. Since there is no redundant computation available in FIR filter algorithm. Efficient realization of FIR filter can be performed using distributed arithmetic (DA) and multiple constant multiplication (MCM) methods. DA based designs are lookup tables (LUTs) to store pre-computed results to reduce the computational complexity. The MCM method helps to reduce the number of additions required for the realization of multiplication by common sub-expression sharing when a given input is multiplied with a set of filter coefficients. The MCM scheme is more effective, when a common operand is multiplied with more number of constants and it is suitable for implementation of large order FIR filter with fixed coefficients. But, MCM blocks can be formed only in FIR filter in transpose form configuration. Throughput of hardware structure can be increased by block processing method. Therefore it helps for improvement in area delay parameters. In direct form FIR filter configuration block processing is simple and straight forward, whereas the transpose form configuration does not directly support block formulation. FIR filter in transpose form configuration can be realized with the help of MCM method to support block processing method.

Transpose form structures are inherently pipelined and supposed to offer higher operating frequency to support higher sampling rate.

Chen and Chiueh [1], have proposed a canonic sign digit (CSD) based reconfigurable FIR filter, where the non-zero CSD values are modified to reduce the precision of filter coefficients without significant impact on filter behaviour. But in which the reconfiguration overhead is significantly large and does not provide area-delay efficient structure. The architectures are more suitable for lower order filters and not appropriate for channel filters due to their large area complexity. Constant shift method (CSM) and programmable shift method are used for RFIR filters, specifically for SDR channelizer. DA based reconfigurable finite impulse response filter architecture has been proposed by Park and Meher [2]. The FIR filter structure with multiplier uses either direct form configuration or transpose form configuration. But, the multiplier less structures use transpose form configuration, whereas the DA based structures uses direct form configuration.

Reconfigurable FIR filter architecture for implementing finite impulse response filter has been implemented by A. Umasankar [3], with low complexity. Reconfigurable FIR architecture can be implemented with constant shift method and programmable shift method. This method is used for transposed form FIR filter. Processing element performs coefficient multiplication operation with the help of shift and add unit. Binary common sub-expression based shift and add method is used here. It consists of two methods, constant shift method (CSM) and programmable shift method (PSM). CSM helps for high speed operation and PSM helps for low area and low power reconfigurable FIR filter. Efficient digital reconfigurable finite impulse response filter architecture is realized by Md. Zameeruddin [4], and has



proposed efficient method for lookup table design in memory based FIR filter. In lookup table based approach the memory elements store all the possible values of product of filter coefficients. It consists of four-three bit address encoder, three-eight line address decoder, memory array of eight words, NOR cell, control circuit and barrel shifter. LUT based multiplication used to reduce memory size. It has less area, low latency and high throughput also.

Computation sharing programmable digital finite impulse response filter has been proposed by Jongsun Park [5], for low power high performance applications. Architecture is based on a computation sharing multiplier (CSHM). Multiplication is performed by vector scaling operation. It helps for reduction of redundant computation in FIR filtering operation by CSHM. Multiplication in vector scaling operation is significantly simplified to add and shift operations of alphabet multiplied by input samples. Low power FIR filters design using MCM and accumulate with modified booth algorithm has been proposed in [6]. This architecture is based on multiple constant multiplications (MCM) accumulate unit. Multiplier block performs multiplication operation consists of partial product generation, partial product compression, truncation and final addition units. It is used for low power and low area applications. But it has losses due to truncation. Digital finite impulse response filter has been proposed by A.D Wankhade [7], based on truncated multiplier. Digital FIR filter in direct form architecture is implemented by MCM method with truncated multiplier. Truncated multiplier reduces the number of partial product bits in the multiplication. Bit width and hardware resources are reduced by using truncated multiplier.

The rest of this paper is as follows. Section II discusses proposed method of reconfigurable FIR filter architecture. In section III, result analysis is explained. Finally, conclusion is described in section IV.

II. PROPOSED METHOD

The proposed structure for block FIR filter for reconfigurable application is shown in figure 1 for block size of L=4 and filter length N=16. It consists of one register unit (RU), one coefficient selection unit (CSU), M number of inner product unit (IPU) and one pipelined adder unit (PAU). Block processing helps for the low power implementation of FIR filters. In this method, number of sequence in blocks of fixed size, as L is processed consecutively.

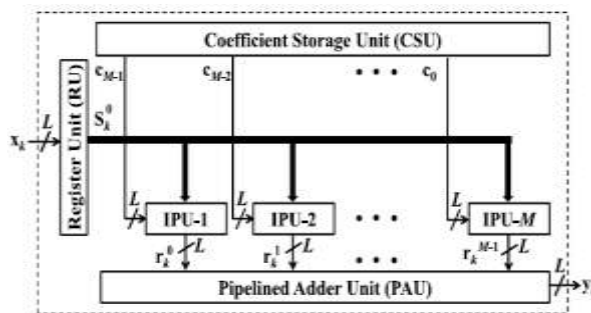


Fig. 1. Reconfigurable FIR filter architecture

In each cycle the proposed structure receives a block of L input samples and produces a block of L filter outputs. The CSU stores coefficient of all filters to be used for reconfigurable applications. It is implemented using multiplexers. At a time CSU can store four set of coefficients. From which based on requirement set of coefficients are selected.

The register unit (RU) helps for the block formulation of input data samples. It receives X_k during the K^{th} cycle and produces L rows of S_k^0 in parallel. L rows of S_k^0 are transmitted to M inner product unit (IPU) of the structure. The M IPU also receives M short weight vectors from CSU. Each IPU performs matrix vector product of S_k^0 with the short weight vector C_m , and computes a block of L partial filter outputs.

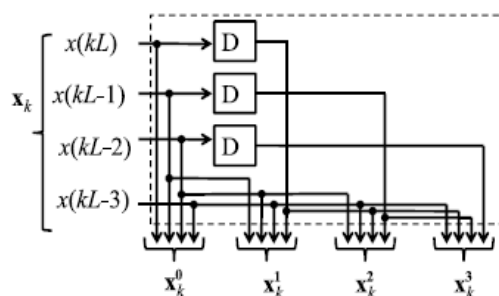


Fig. 2. Register Unit



Figure 2 shows the structure of a register unit. It helps for block formulation for a block size of $L=4$. Inner product unit consists of L number of L point inner product cells (IPC). The IPC receives S_k^0 and the coefficient vector C_m , and computes a partial result of inner product.

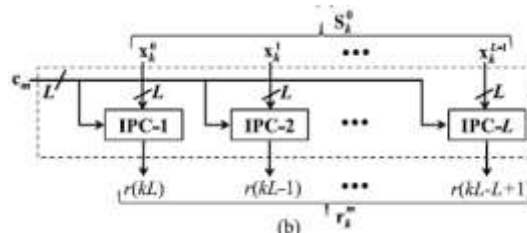


Fig. 3. Inner Product Unit

Figure 3 shows the diagram of inner product unit. All IPU works in parallel and produces of M blocks of result (r_k^m). An FIR filter works by multiplying an array of the most recent n data samples by an array of constants (called the tap coefficients), and summing the elements of the resulting array. Therefore the unit is called as inner product unit.

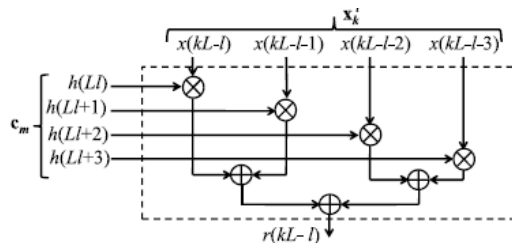


Fig. 4. Inner product cell

Internal diagram of inner product cell in inner product unit is shown in figure 4. Inner product cell consists of multiplication and addition unit. Here multiplication operation is performed by using a dadda multiplier in inner product cell between filter coefficients and data samples. Figure 5 shows a dadda multiplier. Dadda multiplier is a hardware multiplier similar to Wallace tree multiplier is faster and requires less number of gates. Steps for dadda multiplier is that step1: multiply each bit of one of the arguments, by each bit of the other, yielding n^2 results. Depending on position of multiplied bits, the wire carries different weights. Step 2: reduce the number of partial products to two by layers of full adders and half adders. Step 3: group the wires in two numbers and add them with a conventional adder. Take any three wires with the same weights and input them in to a full adder. The result will be the output wire of the same weights and an output wire with a higher weight for each three input wires. If there are two wires of same weight left, and the current number of output wire with that weight is equal to two, input them in to a half adder.

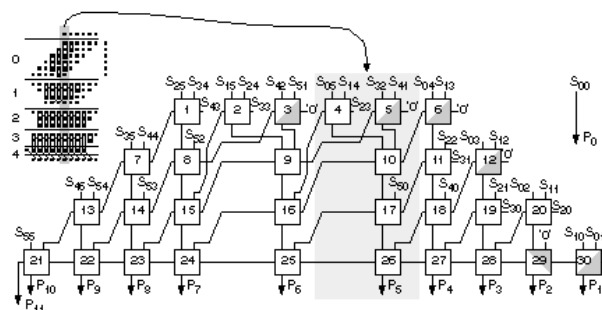


Fig. 5. Dadda Multiplier

Otherwise pass them through to the next layer. If there is just one wire left, connect it to the next layer. If there are two wires of the same weights lefts, and the current number of output wire with that weight is equal to one or two input them in to a half adder. Otherwise pass them through to the next layer. Consider a 6×6 dadda multiplier for multiplication between filter coefficients and input data samples. By using dadda multiplier delay, area and power consumption of the reconfigurable FIR filter is reduced.

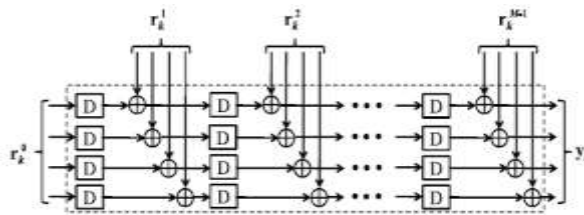


Fig. 6. Pipelined adder unit

Additions of partial inner products are performed by a pipelined adder unit, to obtain a block of L filter outputs. Figure 6 shows the architecture of pipelined adder unit.

III.RESULT ANALYSIS

Reconfigurable FIR filter structure in transpose form configuration is implemented using VHDL language for filter length 16 and block size L=4. Xilinx ISE web pack 14.1 software is used for design, synthesis and implementation. VHDL is used to describe the behaviour and structure of system and circuit design. The proposed structure is simulated on the Xilinx system generator. The simulated output waveforms are obtained with the help of Model Sim 10.3csoftware. Area, delay and power can be measured and comparisons are performed.

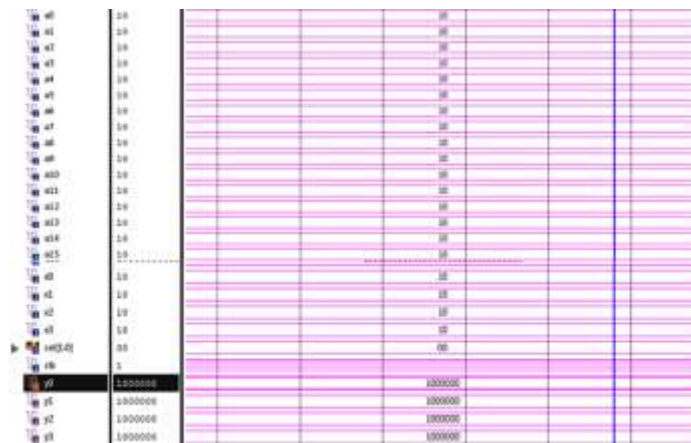


Fig. 7. Simulated waveform

For a filter length of N=16, and block size of L=4, digital FIR filter is implemented and simulated waveforms are shown in the figure7.

Table 1 Performance comparison

Parameters	Reconfigurable FIR filter architecture	Reconfigurable FIR filter with dadda multiplier
Delay (pS)	18243	3705
Power (mW)	954.14	49.04
Cell Area(x10 ³)	1777.31	204.64

Table 1 shows the performance comparison between reconfigurable FIR filter architecture with reconfigurable FIR filter with dadda multiplier. The comparison is performed based on parameters such as area, delay and power. FIR filter with dadda multiplier is efficient compared with FIR filter with ordinary multiplication in terms of area, delay and power for reconfigurable application for block size of L=4, and filter length of 16.

With the help of block formulation method it helps for parallelism of input data flow and output computations. So the reconfigurable filter can be used for higher order applications. The filter architecture has less area-delay product so the structure is small in size also. Because of high sampling rate FIR filter can be used for high speed applications with better performance. Reconfigurable finite impulse response filter has low power also by using block formulation method.



IV. CONCLUSION

Area-delay efficient block finite impulse response (FIR) filter in transpose form configuration can be realized for reconfigurable applications. A generalized block formulation is presented for transpose form block FIR filter, and based on that filter in transpose configuration is derived for reconfigurable applications. Performance comparison shows that the proposed structure involves significantly less area, delay and power from existing block form structure. Here area, delay and power can be reduced with the help of dadda multiplier and used for multiplication between filter coefficients and samples in inner product unit.

REFERENCES

- [1] K.-H. Chen and T.-D. Chiueh, "A low-power digit-based reconfigurable FIR filter," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 53, no. 8, pp. 617–621, Aug. 2006.
- [2] S. Y. Park and P. K. Meher, "Efficient FPGA and ASIC realizations of a DA-based reconfigurable FIR digital filter," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 61, no. 7, pp. 511–515, Jul. 2014.
- [3] R. Mahesh and A. P. Vinod, "A new common subexpression elimination algorithm for realizing low-complexity higher order digital filters," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 2, pp. 217–219, Feb. 2008.
- [4] P. K. Meher, "New approach to look-up-table design and memory based realization of FIR digital filter," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 3, pp. 592–603, Mar. 2010.
- [5] J. Park, W. Jeong, H. Mahmoodi-Meimand, Y. Wang, H. Choo, and K. Roy, "Computation sharing programmable FIR filter for low-power and high-performance applications," *IEEE J. Solid State Circuits*, vol. 39, no. 2, pp. 348–357, Feb. 2004.
- [6] B. K. Mohanty, P. K. Meher, S. Al-Maadeed, and A. Amira, "Memory footprint reduction for power-efficient realization of 2-D finite impulse response filters," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 1, pp. 120–133, Jan. 2014.
- [7] R. Mahesh and A. P. Vinod, "A new common subexpression elimination algorithm for realizing low-complexity higher order digital filters," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 2, pp. 217–219, Feb. 2008.
- [8] S. A. White, "Applications of distributed arithmetic to digital signal processing: A tutorial review," *IEEE ASSP Mag.*, vol. 6, no. 3, pp. 4–19, Jul. 1989.
- [9] B. K. Mohanty and P. K. Meher, "A high-performance energy-efficient architecture for FIR adaptive filter based on new distributed arithmetic formulation of block LMS algorithm," *IEEE Trans. Signal Process.*, vol. 61, no. 4, pp. 921–932, Feb. 2013.