



Review Article

A Comparative Study on FIR Filters for Reconfigurable Applications

Moorthi Kiruban, Raja Jayamani

Department of ECE, Adhiparasakthi Engineering College, Melmaruvathur, Tamilnadu, India

Email address:

murthy2930@gmail.com (M. Kiruban), s.t.jayamani@gmail.com (R. Jayamani)

To cite this article:

Moorthi Kiruban, Raja Jayamani. A Comparative Study on FIR Filters for Reconfigurable Applications. *International Journal of Information and Communication Sciences*. Vol. 3, No. 1, 2018, pp. 1-10. doi: 10.11648/j.ijics.20180301.11

Received: May 8, 2017; **Accepted:** November 22, 2017; **Published:** January 19, 2018

Abstract: Reconfigurability and low complexity are the two key requirements for finite impulse response (FIR) filters employed in multi standard wireless communication systems. In this article, a comparative study of various adaptive filter architectures, which includes BCSE architecture, constant shift method, programmable shift method, multiple constant method, DA based method are presented. This paper aims at study on efficient adaptive filter architecture in terms of area, EPS (energy per sample) and power. By comparing various methods it is observed that MCM structure involves significantly less area delay product and less energy per sample than the existing block implementation methods of direct- form structure for medium or large filter lengths. The MCM structure involves 14% less ADP and 13% less EPS than that of the existing direct- form block FIR structure.

Keywords: Reconfigurable Architecture, Multiple Constant Multiplication, BCSE Algorithm, Constant Shift Method, Programmable Shift Method, Distributed Arithmetic

1. Introduction

The current research is highly focused in the area of adaptive signal processing applications such as channel equalization, channelization, matched filtering and pulse shaping, etc. These applications require higher order digital filter thereby making hardware complex. Very often these filters need to support high speed digital communication. The number of multiplications and additions required for each filter output, however, increases linearly with filter output. In adaptive FIR filter architecture, the direct implementation of N-tap filter requires N MAC operations, which are too expensive in hardware implementations due to its logic complexity and area constraint.

2. Reconfigurable Architectures

The two important considerations for finite impulse response filter are reconfigurability and low complexity for any multi-standard communication systems. Reconfigurable

FIR filter has been developed for software defined radio (SDR) technology and the flexibility of transceiver design makes it possible to design in digital domain.

Following are the some of the methods used for reconfiguring the FIR filters.

2.1. Multiple Constants Method

The MCM method reduces the complexity by common sub expression sharing, when a given input is multiplied with a set of constants [1]. This scheme is suitable for implementing large order FIR filters with fixed coefficients. But, this can be formed only in the transpose form configuration of FIR filters as shown in figure 1.

In many applications the coefficients of FIR filters remain fixed, where as in some other applications, like SDR it requires separate FIR filters of different specifications to extract narrow-band channels from the wideband RF front end. To support multi-standard wireless communication these FIR filters need to be reconfigurable.

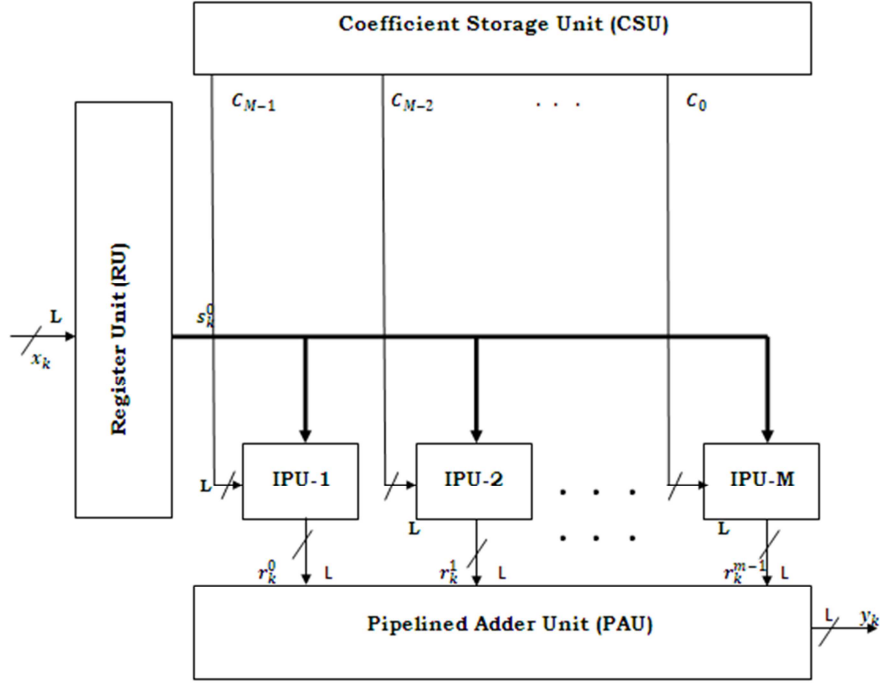


Figure 1. Transposed Form Block form FIR Filter for Reconfigurable Applications.

The MCM structure for FIR filter for block size $L=4$ is shown in figure 2. The MCM structure has six MCM blocks correspond to six input samples. Each MCM block produces the necessary product terms as listed in table 1. The sub

expression of MCM blocks are shifted added in the adder network to produce the inner product values ($r_{l,m}$), for $0 \leq l \leq L-1$ and $0 \leq m \leq (N/L-1)$ corresponds to matrix product of $R = s_k^0 \cdot C$.

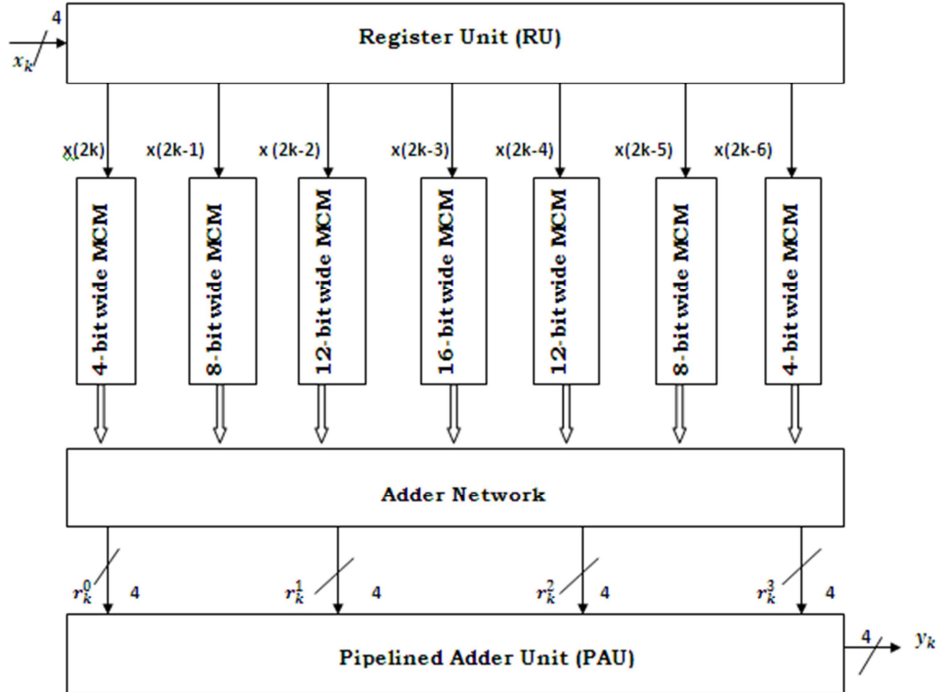


Figure 2. MCM based Structure for fixed FIR filter of block size $L=4$ and Filter length $N=16$.

2.2. BCSE Algorithm

This algorithm deals with the elimination of redundant binary common sub expression (BCSs) that occur within the

coefficients [2]. The BCSE technique eliminates redundant computations in coefficient multipliers by reusing the most common binary bit patterns (BCSs) present in coefficients. An n bit binary number can form $2^n - (n+1)$ BCSs among

themselves, $2^{n-1}-1$ adders are required for all the possible n-bit the binary sub expressions. The number of adders needed to implement the coefficient multipliers using the binary

representation-based BCSE is considerably less than CSD-based CSE methods. This architecture is based on the transposed direct form as shown in figure 3.

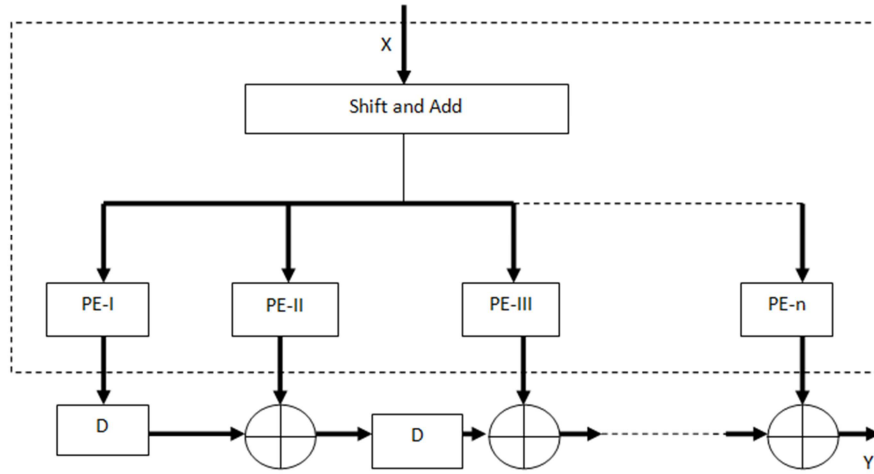


Figure 3. Transposed direct form of an FIR filter.

In transposed direct form, the coefficient multipliers share the same input and hence commonly known as multiplier block (MB). By exploiting the redundancy in MCM, the complexity can be reduced in MB block for FIR filter implementations by using BCSE.

In figure 3 PE-i represents the processing elements corresponds to the i^{th} coefficient. The multiplication operation is done by the shift and adds units that are present in processing element. The basic architecture of the PE is shown in figure 5.

The functions of different blocks of the PE are explained below.

2.2.1. Shift and Add Unit

The complexities of multiplication are reduced by shift and add operations. Figure 4 shows the architecture of shift and add unit. In both CSM and PSM architecture 3-bit BCSs shift and add unit is used.

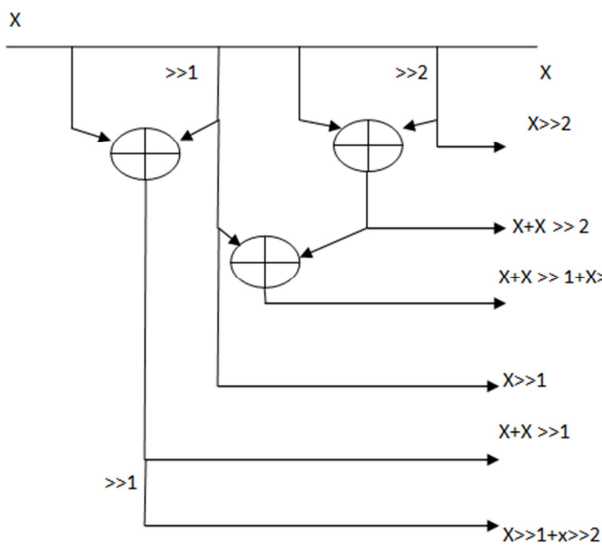


Figure 4. Architecture of shift and add unit.

2.2.2. Multiplexer Unit

The output from the shift and add unit are properly selected by the multiplexer units and they share the outputs of the shift and add unit. Totally 8/4 inputs from the shift and add unit are applied to the inputs to the multiplexers and hence 8: ¼:1 multiplexer unit is used in this architecture. The filter coefficients in LUT are used as select signals of the multiplexer. In CSM method, the coefficients are directly stored in the LUTs without any modification where as in PSM, the coded coefficients are stored.

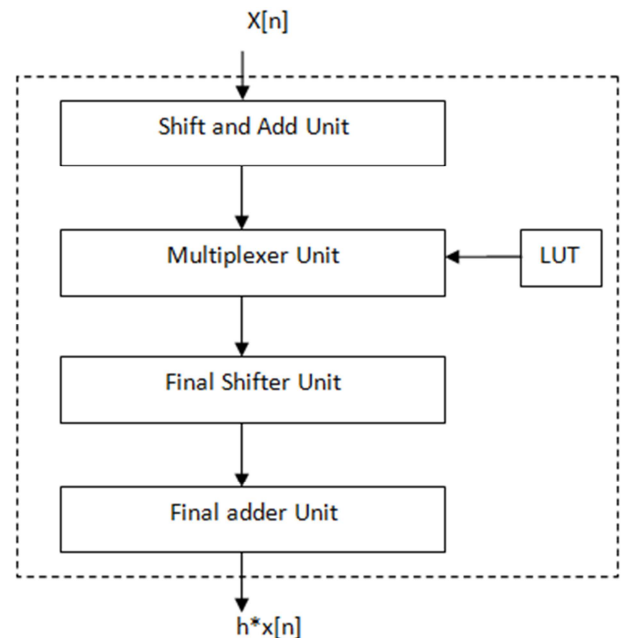


Figure 5. Basic Architecture of processing elements.

2.2.3. The Final Shifter Unit

After the computation of intermediate additions the final shifter unit will perform the shifting operations. In the CSM, the final shifts are constants and hence no PS is required. In

the PSM, PS is used.

2.2.4. Final Adder Unit

This unit will compute the sum of all intermediate additions.

2.3. Constant Shift Method

For high speed filter CSM method is used with slight increase in area and power. In the CSM architecture, the coefficients are stored directly in the LUT. These coefficients are partitioned into groups of 3-bits and are used as the select signals for the multiplexers. The number of multiplexers units required is $\lceil n/3 \rceil$, where n is the word length of the filter coefficient [8].

The CSM can be explained with the help of an 8-bit coefficient $h = "0.11111111."$

Maximum number of additions and shifts are required in case of this non-zero coefficient. In this case, $n=8$, and

therefore the number of multiplexers required is 3. The output $y = h * x$ is expressed as

$$Y = 2^{-1}x + 2^{-2}x + 2^{-3}x + 2^{-4}x + 2^{-5}x + 2^{-6}x + 2^{-7}x + 2^{-8}x \quad (1)$$

By portioning into groups of three bits from MSB,

$$Y = 2^{-1} (x + 2^{-1}x + 2^{-2}x + 2^{-3}x + 2^{-4}x + 2^{-5}x + 2^{-6}x) \quad (2)$$

$$h = 2^{-1} (x + 2^{-1}x + 2^{-2}x) + 2^{-3} (x + 2^{-1}x + 2^{-2}x) + 2^{-6}x (x + 2^{-1}x) \quad (3)$$

Note that the terms $x + 2^{-1}x + 2^{-2}x$ and $x + 2^{-1}x$ can be obtained from the shift and add unit. Then by using the three multiplexers (mux), two 8:1 mux for the first two 3-bit groups and one 4:1 mux for the last two bits of the filter coefficients, the intermediate sums shown inside the brackets of (3) can be obtained. The final shifter unit will perform the shift operations 2^{-1} , 2^{-3} and 2^{-6} . The basic Architecture of PE for CSM is shown in figure 6.

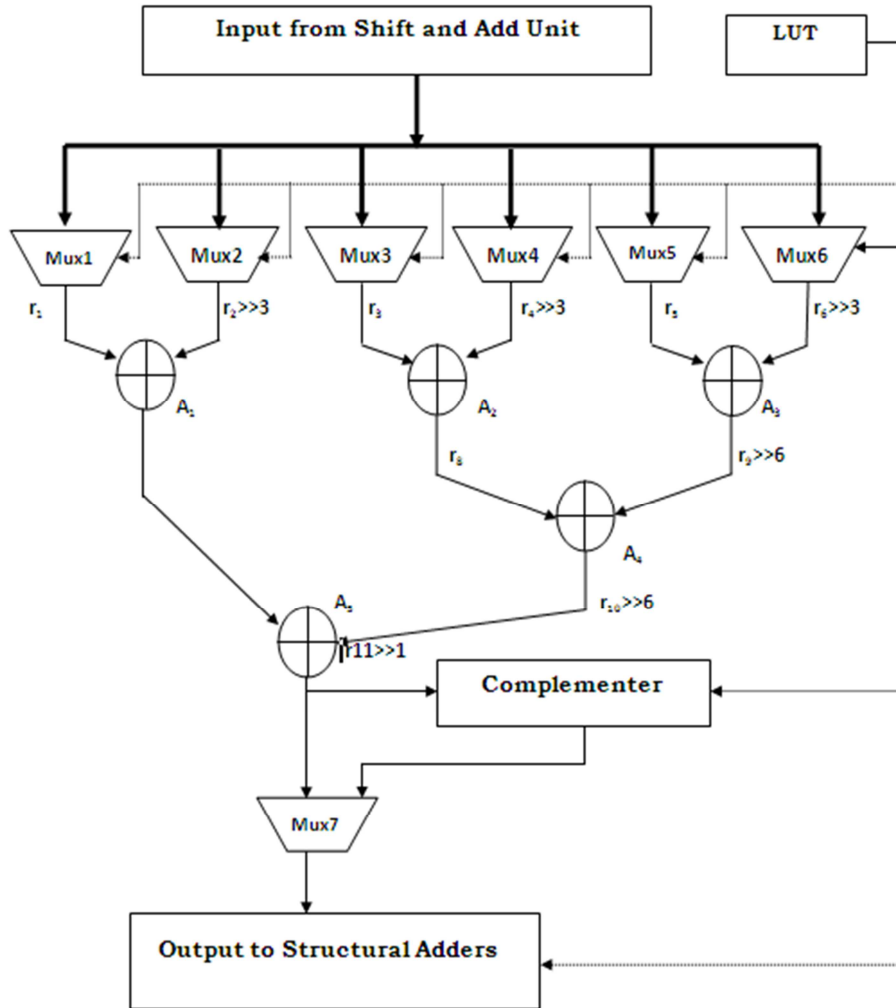


Figure 6. Architecture of PE for CSM.

2.4. Programmable Shift Method

This method is the reconfigurable version of the BCSE algorithm [8]. The PSM has a pre-analysis part in which the

filter coefficients are analyzed using the BCSE algorithm. Thus the redundant computations are eliminated using the BCSs and the resulting coefficients in a coded format are stored in the LUT. The shift and the add unit is identical to for

both PSM and CSM. After the application of BCSE algorithm the number of multiplexer units required for filter coefficients can be obtained. The architecture of PE for PSM is shown in figure 7. The coefficient word length is fixed as 16 bits. After the statistical analysis with different filter lengths it was found that maximum number of non-zero operands is 5 for any coefficients and the number of multipliers as 5. The LUT consists of two rows of 18 bits for each coefficient of the form DDDDXDDDDXXDDDDXX and SDDDXDDDDXXMMML, where "S" represents the sign bit, "DDDD" represents the shift values from 2^0 to 2^{-15} and "XX" represent the input "x" or the BCSs obtained from the shift and add unit. In the coded format, XX= "01" represents "x", "10" represents $x+2^{-1}x$, "11" represents $x+2^{-2}x$, and "00" represents $x+2^{-1}x+2^{-2}x$, respectively. Thus, the two rows can store up to five operands which is the worst case number of operands for a 16-bit coefficients, the number of operand is less than the worst case number of operands, 5. In that case "MMML" can be used to avoid unnecessary additions. The values "MMMM" will be given as select signal to the Mux 6 and "L" to Mux 8. "MMML" indicates the presence of five operands. A "1" in each position indicates the presence of each operand. Thus for all operands to be present will be indicated by "MMML"="11111". This means the Mux6 will select the output of adder, A_4 and Mux8 will select the output of adder, A_2 . If only one operand is present, "MMML"="10000".

This means the Mux8 will select the output of PS, shr4 and Mux6 will select the output of PS, shr1.

The coding can be explained as given below. Consider the positive coefficient h

$$h = [1010011001010011] \quad (4)$$

by using BCSE, substituting 2=[1 1], 3=[1 0 1], (1) becomes

$$h = [3000020003000020] \quad (5)$$

then (2) will be stored in the LUT as 000001101011011110 and 100111111010000000. It must be noted that as (2) has only four operands, the fifth operand values "DDDDXX" are substituted as 000000 and "MMML" as "11110". The XX values are given as select signals for Mux1 to Mux5. The values of DDDD are fed to corresponding PS. The multiplexer Mux6 and Mux8 will select the appropriate output in case the number of operands is less than 5. The use of Mux6 and Mux8 reduces the number of adders utilized by selecting the output from the appropriate adders as all the adders in the PE are not always needed. Mux 7 is used to complement the output in case of a negative coefficient and its select signal is the sign bit "S" of the coefficient. The PSM architecture has two advantages; it guarantees a reduced number of additions compared to CSM and it offers the flexibility of changing the word length of coefficients.

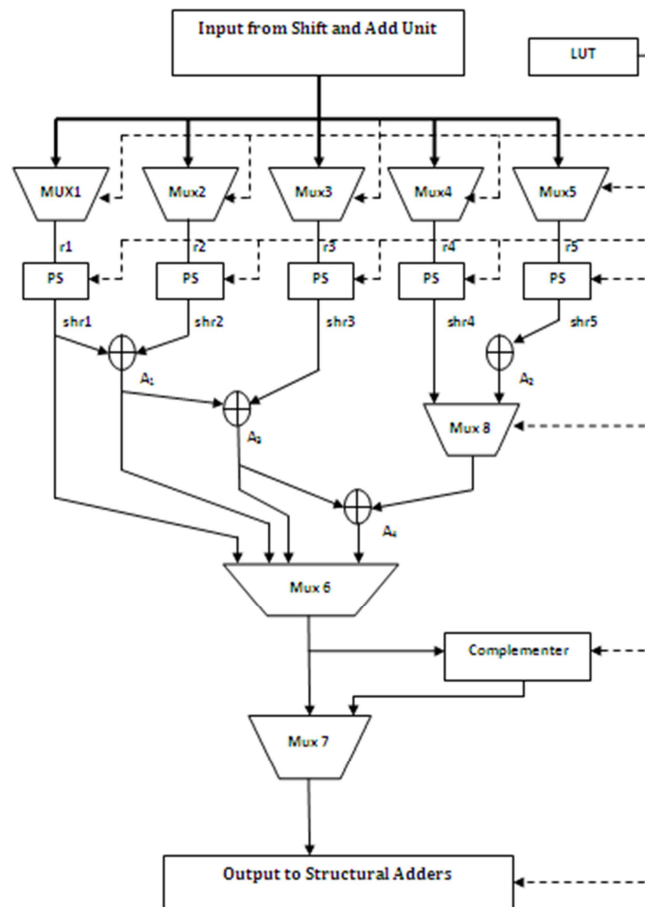


Figure 7. Architecture of PE for PSM.

2.5. Vertical- Horizontal BCSE Algorithm

The effective BCS elimination is done by the Vertical BCSE than the horizontal BCSE [2].

The detection and elimination of BCSs are done by applying 2 bit vertical BCSE followed by 4 bit and 8 bit horizontal BCSE. The data flow diagram of the vertical-Horizontal BCSE Algorithm based constant multiplier is shown in figure 8. The designed multiplier considers the length of the input (X_{in}) and coefficient (H) as 16-bit and 17-bit respectively while the output is assumed to be 16 bit long. The sampled inputs are stored in the register first and the coefficients are stored directly in the LUTs. The sign conversion block is needed to represent the input and

coefficient in signed decimal format. A 2 bit binary common sub-expression (BCSs) is used to reduce the multiplexer size which is used select the proper partial product from the partial product generator (PPG). Control logic generator converts the multiplexer output into 4 bit and 8 bit groups. Depending on the equality check it will generate 7 control signals. Depending on the coefficient's binary value multiplexer unit selects the appropriate data from the PPG unit. In layer 2, controlled addition will add the partial products generated from eight groups of 2-bit BCSs for final multiplication. Controlled addition at layer-3 will add the four multiplexed sums. Finally in layer-4 will add the two sums produced at previous layer.

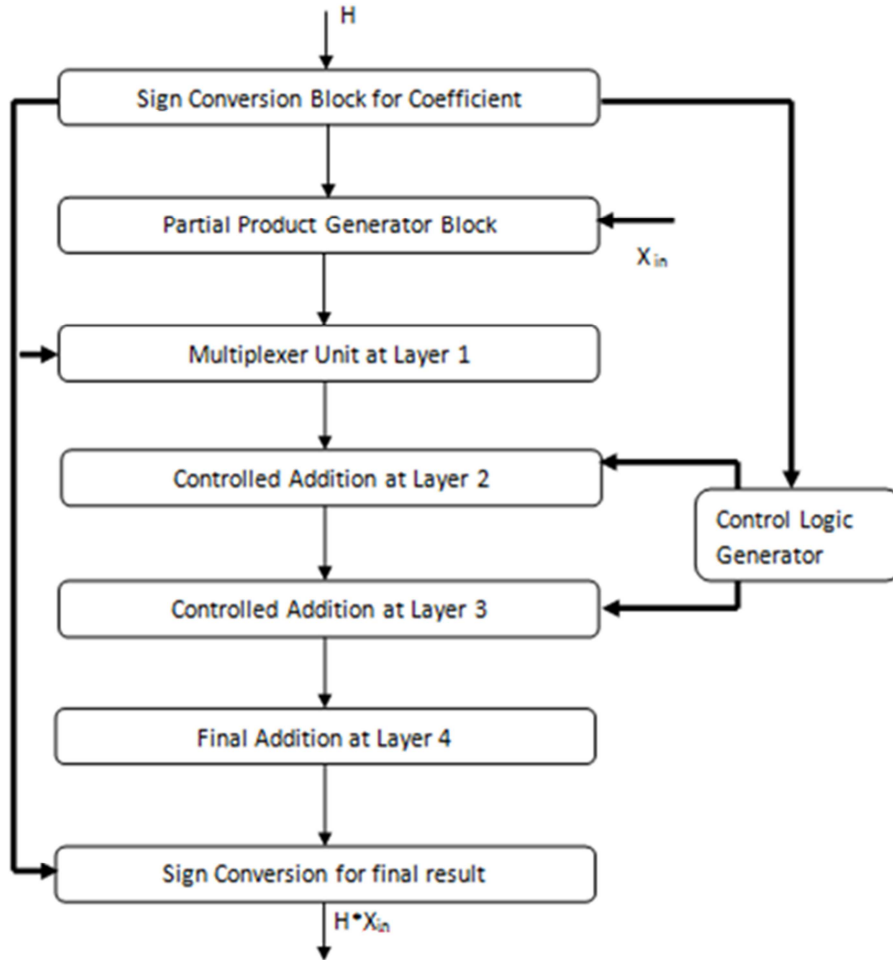


Figure 8. DFG of the CM using VHBCSE algorithm.

2.6. Distributed Arithmetic Based Reconfigurable FIR Digital Filter

This method is area-time efficient and cost-effective due to its high-throughput processing capability and increased regularity. The main operations required for DA-based computation are a sequence of look-up table accesses followed by shift-accumulation operations of the LUT output. In conventional DA implementation FIR filter assumes that impulse response coefficients are fixed, and makes it possible

to use ROM-based LUTs. The memory requirement for DA-based implementation of FIR filters, however, exponentially increases with filter order. To eliminate this large memory requirement, systolic decomposition technique is used for long length convolutions and FIR filter of large orders. For reconfiguration, RAM based LUT can be used for dynamic change in coefficients [6]-[7].

Registers are limited resource in FPGA since each LUT in many FPGA devices contains only two bits of registers. Therefore LUTs are required to be implemented by distributed

RAM (DRAM) for FPGA implementation. Using a DRAM to implement LUT for each bit slice will lead to very high resource consumption. Thus, decompose the partial inner-product generator into Q parallel sections and each section has R time- multiplexed operations corresponding to R bit slices. When L is a composite number given by L=RQ (R and Q are two positive integers)

$$y = \sum_{q=0}^{Q-1} 2^{-Rq} \left[\sum_{r=0}^{R-1} 2^{-r} \left(\sum_{p=0}^{P-1} S_{r+qR,p} \right) \right] \quad (6)$$

Figure 9 shows the time multiplexed DA-based FIR filter using DRAM. To implement equation (6) the structure has Q sections, and each sections consists of P DRAM-based DRPPGs (DRPPGs) and the PAT to calculate the rightmost summation, followed by shift-accumulator that performs over R cycles according to second summation. However, use of dual-port DRAM to reduce the total size of LUTs by half since two DRPPGs from two different sections can share the single DRAM.

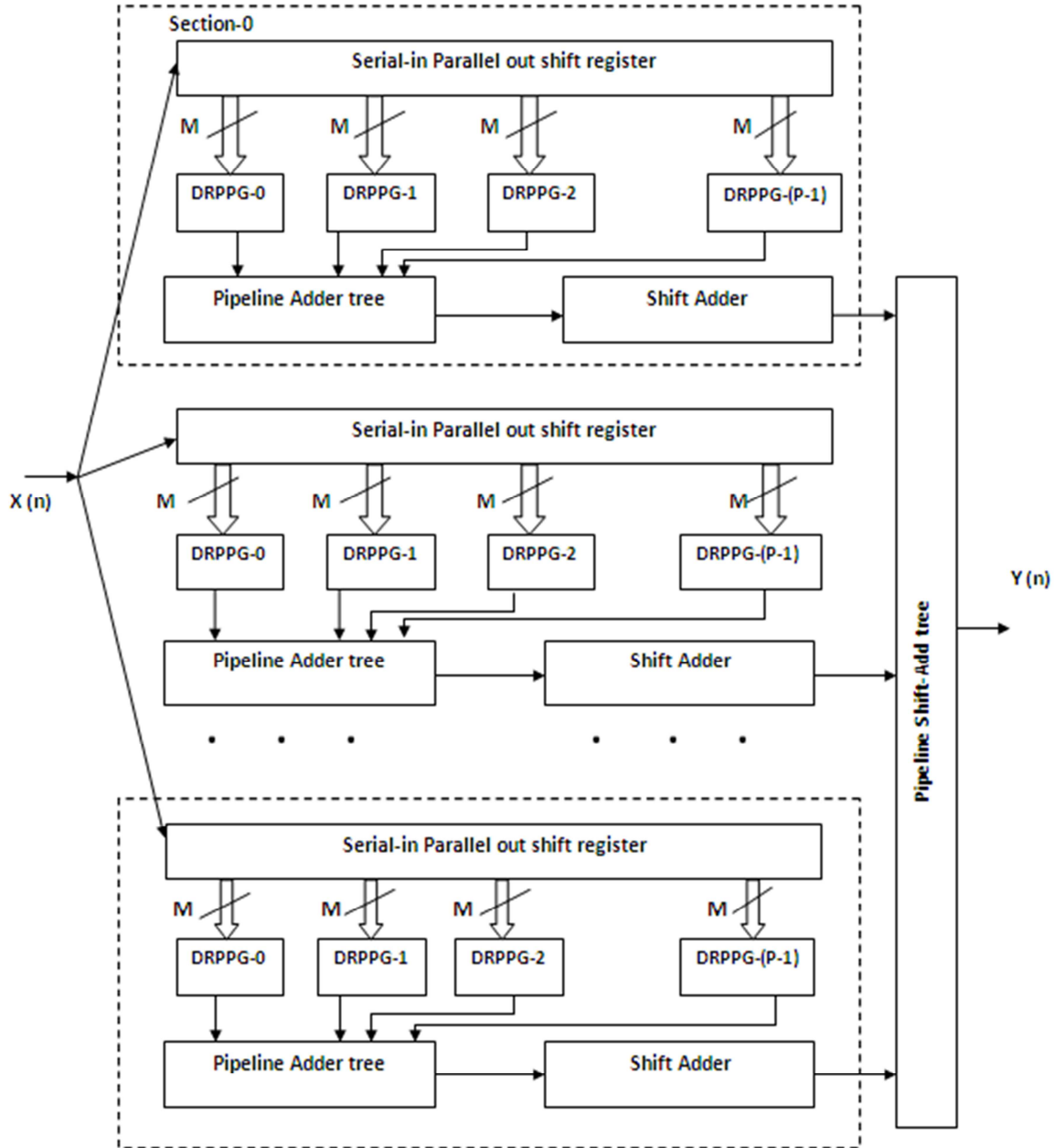


Figure 9. Structure of the DA-based FIR filter.

3. Results and Analysis

The objective of this analysis is to select an area, power and delay efficient FIR filters for reconfigurable applications. For high speed filters the CSM architecture is preferable. Whereas for low area and low power PSM architecture proved to be

better choice and also it provides flexibility in changing the filter coefficient word length dynamically. These reconfigurable FIR architectures can be modified to any MCM method. This method well adapts to the low complexity reconfigurable channel filters. Table 1 provides the theoretically estimated hardware and time complexities of

various structures.

In MCM scheme, to reduce the computational complexity horizontal and vertical sub expression elimination is done in block FIR filter. The comparison bar chart shown in the figure 10 and figure 11 shows this structure has less ADP¹ and less EPS². Table 2 shows the area, the minimum clock period

(MCP) and power estimates obtained from the synthesis reports. This MCM scheme has more area and power due to extra FFs. But it has less ECP (higher sampling frequency³). ADP varies directly with (ΔA - increase in area) and inversely with (ΔT -reduction in MCP).

Table 1. Theoretically estimated hardware and time complexities of various structures.

Structures	Filter Length	FF	Adder	Multipliers	Cycle period	Throughput
CSM [9]	16	360	114	0	$4T_{MUX}+4T_A+3T_{FA}$	1/T
	32	744	226	0	$4T_{MUX}+4T_A+3T_{FA}$	1/T
	64	1512	450	0	$4T_{MUX}+4T_A+3T_{FA}$	1/T
DA [10]	16	296	71	0	$2T_{MUX}+2T_A+2T_{FA}$	1/T
	32	432	143	0	$2T_{MUX}+2T_A+3T_{FA}$	1/T
	64	696	287	0	$2T_{MUX}+2T_A+4T_{FA}$	1/T
MCM [1] L=4	16	312	60	64	$T_M+T_A+2T_{FA}$	4/T
	32	696	124	128	$T_M+T_A+2T_{FA}$	4/T
	64	1464	252	252	$T_M+T_A+2T_{FA}$	4/T
MCM [1] L=8	16	248	120	120	$T_M+T_A+3T_{FA}$	8/T
	32	632	248	256	$T_M+T_A+3T_{FA}$	8/T
	64	1400	504	512	$T_M+T_A+3T_{FA}$	8/T

¹ADP= area/ Sampling frequency.

²EPS= power/ sampling frequency.

³sampling frequency= block size/MCP.

Table 2. Synthesis results for various structures showing MCP, sampling rate, area, and power for various filter lengths.

Structures	N	Samp. Rate (GHz)	Area (μm^2)	Power (mW)
CSM [9]	16	0.78	67949	28.7
	32	0.77	133015	55.8
	64	0.76	264342	104.4
DA [10]	16	1.05	56325	24.9
	32	0.90	112446	47.3
	64	0.78	218351	85.6
VHBCSE [11]	16	0.2	44633	2.24
	32	0.2	87855	4.46
	64	0.2	175734	8.46
2 bit BCSE [11]	16	0.2	43317	2.46
	32	0.2	86735	5.25
	64	0.2	172542	13.66
3 bit BCSE [11]	16	0.2	62271	3.53
	32	0.2	124056	9.24
	64	0.2	258662	17.10
MCM [1] L=4	16	3.07	123489	59.3
	32	3.07	247350	108.2
	64	3.07	495437	201.1
MCM [1] L=8	16	5.71	232089	100.9
	32	5.71	476503	186.4
	64	5.71	957186	366.2

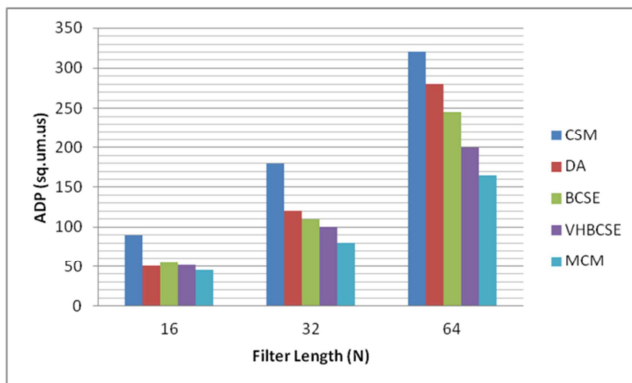


Figure 10. Comparison of ADP.

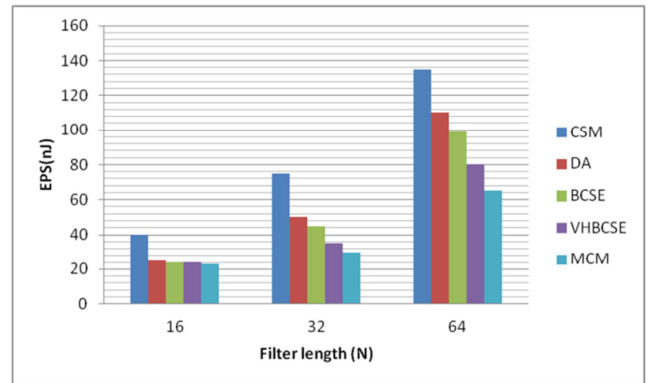


Figure 11. Comparison of EPS.

4. Conclusion

A brief review on area, power and delay efficient FIR filter for reconfigurable applications is presented. Performance comparison shows that the MCM blocks for horizontal and vertical sub expression elimination involves significantly less area delay product and less energy per sample than the existing block implementation methods of direct- form structure for medium or large filter lengths. While for the short length filters has less ADP and less EPS than the MCM block realization. The MCM structure involves 14% less ADP and 13% less EPS than that of the existing direct- form block FIR structure.

References

- [1] Basant Kumar Mohanty and Pramod Kumar Meher, "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 2016.
- [2] Indranil Hatai, Indrajit Chakrabarti and Swapna Banerjee, "An efficient constant multiplier architecture Based on vertical-horizontal binary common Sub-expression elimination algorithm for Reconfigurable fir filter synthesis," *IEEE transactions on circuits and systems—i: regular papers*, vol. 62, no. 4, April 2015.
- [3] Britto Pari J. and Joy Vasantha Rani S. P., "An optimized architecture For adaptive digital filter," *ARNP Journal of Engineering and Applied Sciences*, vol. 10, no. 11, June 2015.
- [4] 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.
- [5] 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.
- [6] M. Surya Prakash and Rafi Ahamed Shaik, "Low-Area and High-Throughput Architecture for an Adaptive Filter Using Distributed Arithmetic," *IEEE transactions on circuits and systems—II: express briefs*, vol. 60, no. 11, November 2013.
- [7] 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.
- [8] R. Mahesh and A. P. Vinod, "New reconfigurable architectures for implementing FIR filters with low complexity," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 29, no. 2, pp. 275–288, Feb. 2010.
- [9] 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.
- [10] R. Mahesh and A. P. Vinod, "A new common sub-expression elimination algorithm for realizing low-complexity higher order digital filters," *IEEE Trans. Computer Aided Design Integrated Circuits Syst.*, vol. 27, no. 2, pp. 217–219, Feb. 2008.
- [11] S. J. Darak, S. K. P. Gopi, V. A. Prasad, and E. Lai, "Low-complexity reconfigurable fast filter bank for multi-standard wireless receivers," *IEEE Trans. Very Large Scale Integration (VLSI) Syst.*, vol. 22, no. 5, pp. 1202–1206, May 2014.
- [12] C. Y. Yao, W. C. Hsia, and Y. H. Ho, "Designing hardware-efficient fixed-point FIR filters in an expanding sub-expression space," *IEEE Trans. Circuits and Systems I, Reg. Papers*, vol. 61, no. 1, pp. 202–212, Jan. 2014.
- [13] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms and Applications*. Upper Saddle River, NJ, USA: Prentice-Hall, 1996.
- [14] 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.