© 2017 IJSRSET | Volume 3 | Issue 1 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

An Efficient Constant Multiplier Architecture for Realizing Fixed Point Reconfigurable FIR filter

Amutha M
Department of Electronics and Communication Engineering, Thanthai Periyar Govt. institute of technology, Vellore, Tamilnadu, India

ABSTRACT

This brief proposes one new algorithm called Vertical-Horizontal Binary Common Sub-expression Elimination (VHBCSE) algorithm for designing an efficient constant multiplier architecture for fixed point reconfigurable Finite Impulse Response filter. In a reconfigurable FIR filter, the coefficients can dynamically change in real time and are thus inevitable in multistandard wireless communication systems. In the proposed algorithm, the filter coefficients are first pre-analyzed using BCSE algorithm to avoid complete redundancy in coefficient multiplication and then different lengths of Binary Common Sub-expression Elimination algorithms are applied in different layers of Multiplier adder tree to eliminate redundant computations for common sub-expressions in producing the multiplication result and hence reduce the hardware and power consumption. The use of carry save adders for much faster operation and the use of signed decimal format data representation make the design most appropriate for designing any order efficient reconfigurable systems. The Efficiency of the proposed design is shown by comparing ASIC and FPGA implementation results of the present design viz., power, area and speed with the best existing reconfigurable FIR filter implementations in the literature. Xilinx 9.2i ISE, MATLAB Filter Visualization tool and Cadence tool with Faraday 90nm CMOS technology library are used for the synthesizing purposes.

Keywords : Reconfigurability, VHBCSE algorithm, Carry save adder, Signed decimal format data representation, Multiplier adder tree, Constant multiplier, finite impulse response filter, Multistandard system, VLSI design

I. INTRODUCTION

The Finite Impulse Response (FIR) filters are a class of digital filter whose response to any finite length input is of finite duration and are widely employed as a principal component in digital signal, image and video processing systems. Specifically, they find extensive application in wireless communication systems for various applications such as channelization, channel equalization, pulse shaping, matched filtering, echo cancellation, etc. because of their useful properties of absolute stability and linear phase. In such applications, the filters must be of higher order in order to achieve sharp transition band and stringent adjacent channel attenuation. As the complexity of implementation grows with the filter order and the precision of computation, real-time realization of these filters with desired level of accuracy becomes a challenging task. So FIR filter architectures for high order, high speed and low power applications are a need for the day.

Recently, the benefits and anticipated opportunities for Software defined radio (SDR) technology are having a significant impact on the wireless industry. The basic concept of an SDR is to implement some or all of the radio’s operating functions through modifiable software or firmware such as digital signal processors operating on programmable processing technologies. The use of these technologies enables new wireless features and capabilities to be added to existing radio systems without requiring new hardware which will allow several different air interfaces to be processed on a generic platform to support multistandard systems [1]. So, dynamic reconfigurability of the receiver to operate with different wireless standards is the principal requirement in an SDR system to provide flexibility in operation. Moreover, systems like multi-standard video codec [2] and multi-standard digital up/down converter [14] need a reconfigurable FIR filter with dynamically programmable filter coefficients and lengths which may vary in real time with different standards in a portable computing platform. The use of fully programmable filter processors implementing the reconfigurable FIR filter increases the area and power consumption [3]. Thus the realization of an efficient dedicated hardware architecture for reconfigurable FIR filter is very much needed for its significant applicability in any multistandard system.

In any FIR filter, the coefficient multiplier is the major constraint which defines the performance of the desired filter. The system's performance is generally determined by the performance of the multiplier because it is generally the slowest, most area and power consuming element in the system. Hence, optimizing the speed, area and power consumption of the multiplier is a major design issue. So far, one of the most efficient ways to simplify the multiplication process is to realize it using shifts and add operations. We call such a structure as the multiplier adder tree (MAT). The term MAT refers to the tree structured adders used to implement the multiplication operation. As shifts are less expensive in terms of hardware implementation, the design problem is on reducing of number of addition operations performed to obtain the multiplication result. Many designs have been proposed in literature to implement reconfigurable FIR filters. In [17], an approach based on information theory is presented. It uses a graph structure and the set of coefficients is partitioned into symbols. In [16], a design based on the concept of faithfully rounded truncated multipliers is presented. Both the designs are based on the multiple constant multiplication technique. Although the area costs for the two designs are reduced, the speed performance of both is moderate with increased critical path delay. In [14], a look up table (LUT) multiplier adder approach is presented where memory elements are used to store all the possible values of products of the filter coefficients. In [15], a distributed arithmetic (DA) based reconfigurable FIR filter implementation is presented. In both the implementations, the memory requirement increases with the filter order and real-time implementation of these filters of large orders is a challenging task.

Among the other approaches, the Common Subexpression Elimination (CSE) algorithms have been used as a powerful tool for eliminating hardware redundancies and reduce area and power consumption, especially for higher order filters in low complexity fixed point FIR filter implementations [12]. The main idea of CSE is to detect instances of identical bit patterns that are present in any particular representation of the coefficients and to eliminate computations for those redundant bit patterns by reusing the results between the common sub-expressions (CSs) but with appropriate shifts in bit positions.

Two classes of common sub-expressions occur in the filter coefficients, called the horizontal common sub-expressions (HCS) and the vertical common sub-expressions (VCS). CSE algorithms based on several different representations of filter coefficients have been proposed in literature. The methods in [4] – [10] used canonical signed digit (CSD) representation to eliminate the identical bit patterns within the coefficient known as Horizontal common sub-expression elimination (HCSE) and across adjacent coefficients known as Vertical common sub-expression elimination (VCSE) [11]. The CSD-based CSE methods suffer from the drawback that the symmetry of FIR filter coefficients cannot be completely exploited when the bits in VCSs are of opposite sign. As a result, additional adders are required to obtain the symmetric part of the coefficients when more than one VCS with bits of opposite sign exist. Also, they require additional memory for storing the sign bit and involve subtraction operation which can be a constraint in realizing reconfigurable filters. In [8], minimal signed digit (MSD) representation was used for the filter coefficients. The MSD representation also suffers from the same drawbacks as CSD and also it can have more than one representation for the same decimal or binary representation.

In [19], a CSE algorithm based on binary representation of coefficients for realizing higher order FIR filters is presented. The authors show that the number of bits that do not form part of CSs is considerably less than that with CSD representation, especially for higher order FIR filters and thus the binary based CSE method is more efficient in reducing the number of adders needed to realize the multipliers. In [12], [13], a CSE technique based on binary representation of coefficients called binary common subexpression elimination (BCSE) which can completely exploit the symmetry of FIR filter coefficients was proposed. This technique uses signed magnitude format for representing the input and filter coefficient, which makes the system inappropriate for higher and lower order systems. They consider only fixed bit binary CSs (BCSs) and improper choice of the length of the BCS may lead to inefficient utilization of hardware. They attempt to eliminate redundant computations only vertically and optimization techniques are needed to reduce the probability of use of adders in the MAT.

Based on the above discussion, it has been realized that the design of an efficient hardware architecture for the constant multiplier in reconfigurable FIR filter is very much needed for its wide applicability in any reconfigurable system. In this paper, we propose one new BCSE algorithm which is a combination of vertical and horizontal BCSE algorithms called vertical horizontal binary common subexpression elimination (VHBCSE) algorithm for designing an efficient constant multiplier filter. According to the proposed algorithm, the 2-bit VCSE has been applied across the adjacent coefficients to generate the partial products and then conditional 4-bit and 8-bit HCSE are applied within the coefficients in the successive layers of the MAT to sum up the partial products. And the suitability of the designed architecture for synthesis of fixed point reconfigurable FIR filter is also established.

This paper is organised as follows: In section II, a detailed description of the proposed algorithm is presented. In section III, the hardware architecture of the proposed reconfigurable constant multiplier architecture has been described. In section IV, the architecture of the proposed fixed point reconfigurable FIR filter is presented. The simulation and synthesis results and discussions are provided in section V. Finally, the conclusion is drawn in section VI.

II. METHODS AND MATERIAL

1. Proposed Algorithm

The VHBCSE algorithm considers the coefficients in binary form and uses a combination of vertical and horizontal BCSE algorithms to design an efficient reconfigurable constant multiplier architecture. The goal of VHBCSE algorithm is to produce an area and power efficient reconfigurable constant multiplier by eliminating more redundant computations in the shift and add based constant multiplier architecture. According to the proposed algorithm, firstly the 2-bit VCSE has been applied on the coefficients and then 4-bit and 8-bit HCSE respectively, are applied in the successive layers of the MAT to generate the multiplication result. The proposed algorithm considers the signed decimal format data representation which is widely used by most of the systems for both the inputs and the filter coefficients.

For our multiplier design, we have taken the length of the input (x) and coefficient (h) as 16-bit and 17-bit respectively, while the output is assumed to be 17-bit long with MSB reserved the sign bit. Herein, the sampled inputs are stored in the register and the filter coefficients are stored in the LUT. Instead of directly storing the coefficients in the LUT, the coefficients are preanalyzed using BCSE algorithm to avoid complete redundancy in coefficient multiplication. The proposed algorithm consists of the following steps:

1) Get the input ‘x’ of 16 bits.
2) Get the filter coefficients of 17 bits in the signed decimal format.
3) Preanalyze the coefficients using the BCSE algorithm for repetition and store them in LUT.
4) Retrieve the coefficient ‘h’ from LUT and produce the multiplexed coefficient (MC) ‘hm’ of 16 bits which represents the magnitude of ‘h’.
5) Partition the MC into fixed groups of 2 bits each and use them as the select lines to eight (4:1) multiplexers. Apply 2-bit VCSE as follows:
    a. Precompute the partial products corresponding to all the 2-bit BCSs ranging from “00” to “11”. Partial product corresponding to the exact bit position of the 2-bit BCS in MC can be obtained by right shifting the precomputed partial product by appropriate number of bits.
    b. Based on the 2-bit group, the multiplexer selects the correct partial product from the precomputed values. By this way, eight partial products are generated.
    6) Now perform controlled addition of the generated partial products based on 4-bit BCSE. This can be done as follows:
    a. Partition the MC into groups of 4 bits.
    b. Produce the output corresponding to each of the 4-bit group one by one individually by adding the appropriate partial products generated in the earlier step.
    c. If any of the 4-bit group is redundant, skip the computations for that group. Use the result obtained for the same bit pattern in the earlier group shifted to the appropriate number of bits. By this way, four adder outputs are obtained.

7) Now perform controlled addition of the four results obtained in step-6 based on 8-bit BCSE to produce two adder outputs.
8) Now add the two outputs obtained in step-7 to produce the final addition result. This represents the magnitude of the multiplication result x*h.
9) Now convert this result into signed decimal format based on MSB of the LUT coefficient h. This represents the final multiplication result. Store this result in the register.

2. Hardware Architecture of The Proposed Constant Multiplier

In this section, hardware architecture of the constant multiplier based on the proposed VHBCSE algorithm that incorporates reconfigurability into the FIR filter architecture is given. The overall architecture of the constant multiplier is shown in Fig. 1. Let us discuss the functionality of each of the blocks in detail.

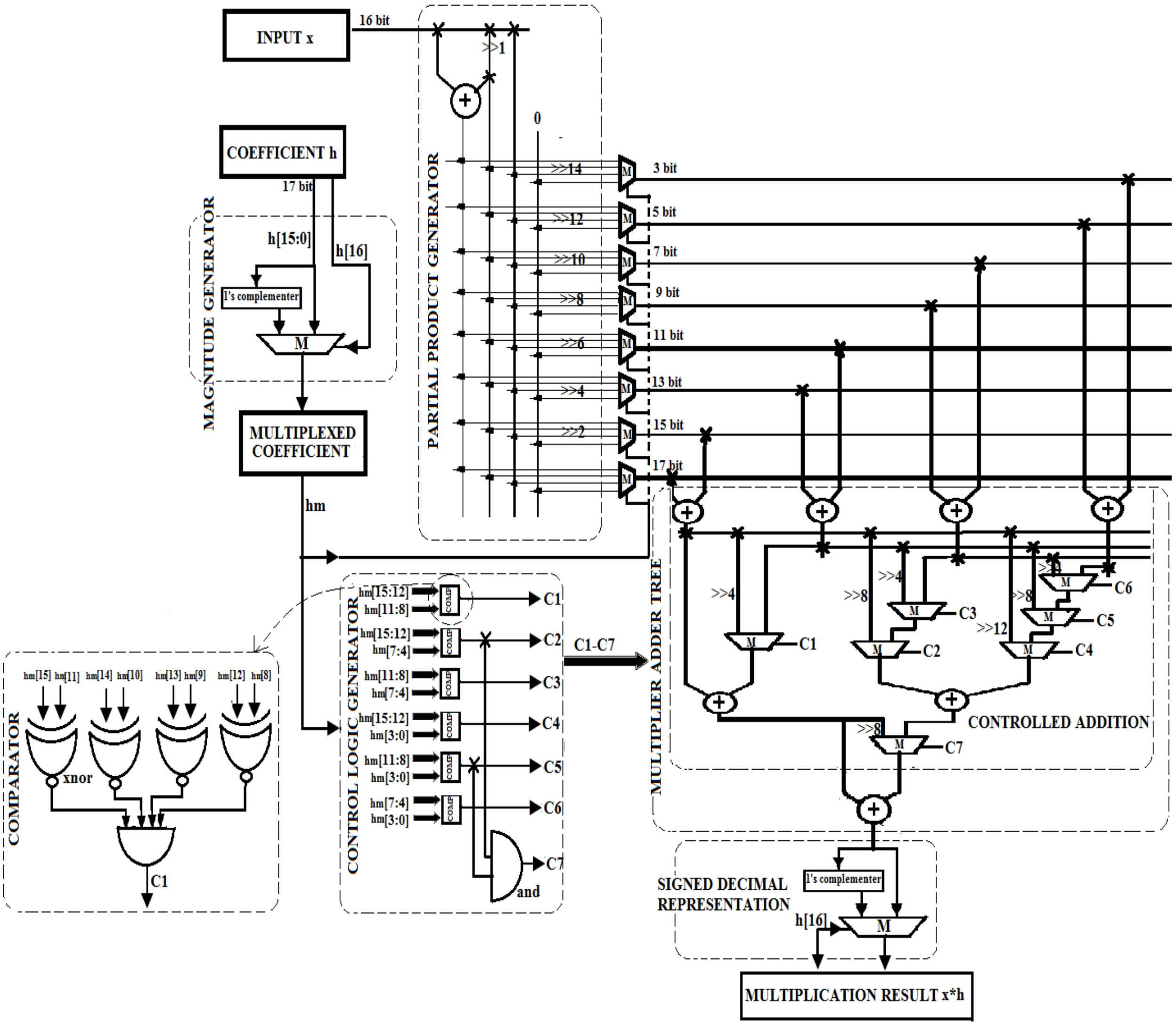


Figure 1. Proposed Constant multiplier architecture

The architecture has a pre-analysis part in which the filter coefficients are pre-analyzed using BCSE algorithm to avoid complete redundancy in coefficient multiplication and the resulting coefficients are stored in the LUT in signed decimal format.

A. Magnitude Generator

This block generates the magnitude of the filter coefficient represented in signed decimal format. Depending on the value of the sign bit (MSB) of the coefficient, one 16-bit 2:1 multiplexer produces either the original coefficient (excluding MSB) or its inverted form. We call this as the multiplexed coefficient hm.

B. Partial Product Generator (PPG)

This represents the shared block between all the constant multipliers of the FIR filter architecture. This unit uses shift and add technique to generate the partial products. The BCSs based shift and add unit is used which precomputes partial products corresponding to all the 2-bit BCSs ranging from “00” to “11”. Of these four BCSs, an adder is required only for the BCS “11”. This facilitates speed improvement and hardware reduction when performing the multiplication operation.

Partial product corresponding to exact bit position of the 2-bit BCS in hm can be obtained by right shifting the precomputed partial product by appropriate number of bits. For the coefficient of 16-bit length, for each precomputed partial product, 8 partial products of 17, 15, 13, 11, 9, 7, 5 and 3 bits will be generated by right shifting it by 0, 2, 4, 6, 8, 10, 12 and 14 bits respectively. Since the shifts are constant irrespective of the coefficients, they can be hardwired resulting in high speed operation. The cost of the PPG unit is independent of the filter length as the same PPG unit is shared by all the coefficients at each constant multiplier.

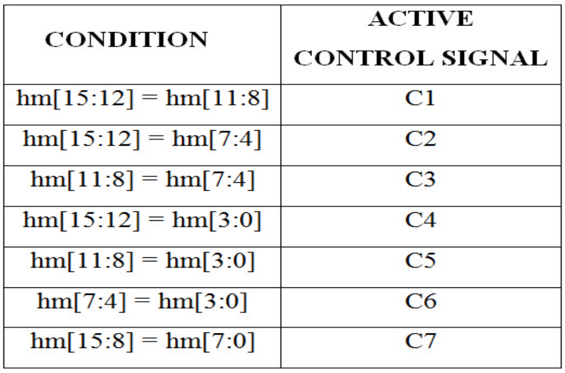
C. Multiplexer unit

Depending on the coefficient’s binary value, the multiplexer unit will select the appropriate data from the PPG unit. For the coefficient of 16-bit length, eight 4:1 multiplexers are used to select the appropriate partial product from the precomputed values according to the 2-bit BCSE algorithm applied vertically on the MAT. All the multiplexers share the output the PPG unit. The multiplexed coefficient is split into groups of two bits each and each group forms the select line to the corresponding multiplexer. The widths of these 8 multiplexers are 17, 15, 13, 11, 9, 7, 5 and 3 bits each instead of 16 bits for all. This would reduce the hardware and power consumption. By this way, eight partial products each of length 17, 15, 13, 11, 9, 7, 5 and 3 bits respectively are generated.

D. Control Logic Generator (CLG)

This block takes the multiplexed coefficient (hm[15:0]) as its input and groups it into one of 4-bit each (hm[15:12], hm[11:8], hm[7:4], and hm[3:0]) and another of 8-bit each (hm[15:8], hm[7:0]) and produces 7 control signals depending on the equality check for 7 different cases as given in table I. These control signals are used to control the addition of partial products generated by the PPG unit according to 4-bit and 8-bit HCSE respectively at successive layers of the MAT. Comparators made of XNOR-AND gates can be used to perform the equality check for each case.

TABLE I. CONTROL LOGIC FOR MAT



CONDITION	ACTIVE CONTROL SIGNAL
hm[15:12] = hm[11:8]	C1
hm[15:12] = hm[7:4]	C2
hm[11:8] = hm[7:4]	C3
hm[15:12] = hm[3:0]	C4
hm[11:8] = hm[3:0]	C5
hm[7:4] = hm[3:0]	C6
hm[15:8] = hm[7:0]	C7

E. Multiplier Adder Tree

In this block, the partial products generated from the PPG unit followed by the multiplexer unit are summed up using tree structured adders to produce the multiplication result. Adders of different word lengths are required for different binary weights of the multiplexer outputs. The adding process is performed in three steps. In the first two steps, instead of direct addition of the partial products, controlled addition operations are performed according to the proposed algorithm based on the control signals generated by the CLG unit. And in the third step, the intermediate adder results are summed up to produce the final multiplication result.

In the step-1, the partial products selected by the multiplexer unit are added based on 4-bit BCSE controlled by the signals C1 through C6 to produce four adder results. In the step-2, the results of step-1 are added based on 8-bit BCSE to produce two adder results controlled by the control signal C7. The main advantage of the controlled addition is that some of the adders will be unloaded resulting in less area and power consumption compared to that for direct addition. And in the step-3, the two results from step-2 are added together to produce the multiplication output. For the addition operations in all the three steps, carry save adders are used which are more power efficient with improved speed of operation. Instead of bit wise addition, since we use BCS wise addition, the same hardware architecture can be used for different filter specifications with different coefficient sets to achieve the necessary reconfigurability.

F. Signed decimal format data representation

The multiplication result obtained from the CA block is then converted into signed decimal format data representation. A multiplexer is used to complement the output in case of a negative coefficient whose select signal is the sign bit of the filter coefficient. This represents the final multiplication result.

3. Architecture of The Proposed Reconfigurable Fir Filter

In this section, the architecture of the reconfigurable FIR filter based on the constant multiplier architecture proposed in section III is presented. The architecture is based on the transposed direct form FIR filter structure and is shown for a 4-tap FIR filter in Fig. 2.

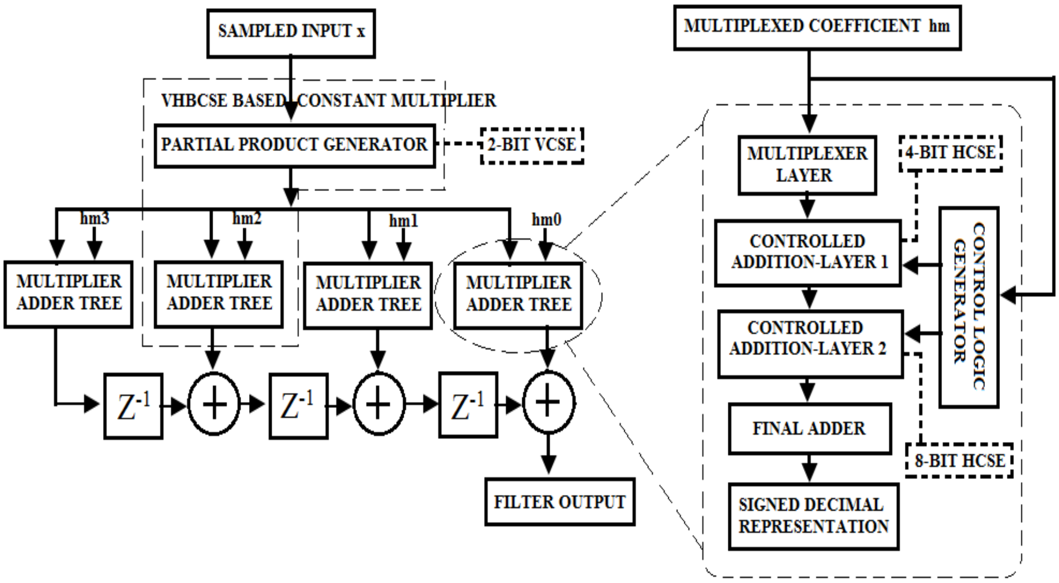


Figure 2. Proposed Reconfigurable FIR filter architecture

The dotted portion in the figure represents the VHBCSE based constant multiplier. The Partial product generator block represents the shared block between all the constant multipliers. It takes the sampled input x as input and computes the partial products for all possible 2-bit BCSs using BCSE based shift and add technique. The precomputed partial products are then sent to the Multiplier adder tree blocks for each multiplexed filter coefficient. Each of the blocks includes the multiplexer layer, controlled addition layers and the final data representation block. The multiplexer layer will select the exact partial products based on the multiplexed filter coefficient, which are then summed up using controlled addition layers based on 4-bit VCSE and 8-bit VCSE respectively. The multiplication result from the final adder unit is then converted into signed decimal format for use with a wide range of systems. The multiplication results from all the multiplier adder tree blocks are then accumulated using the final accumulation unit. The data accumulation unit has a chain of three registers and three adders and produces the final filter output.

III. RESULTS AND DISCUSSION

In this section, we provide the synthesis results of the proposed constant multiplier architecture and the reconfigurable FIR filter architecture based on the proposed constant multiplier architecture. We also prove the efficiency of the proposed design by comparing its performance measures with several other reported works in literature. The proposed VHBCSE based constant multiplier architecture shown in Fig. 1 has been coded using Verilog and synthesized in the FPGA platform using Xilinx ISE 9.2i synthesis tool and in the ASIC platform using Cadence tool. The results of the synthesis are shown in table II.

TABLE II IMPLEMENTATION RESULTS OF THE PROPOSED CONSTANT MULTIPLIER

FPGA Platform	Number of Slice LUTs	312
	Number of Bonded IOBs	50
ASIC Platform	Area consumption (um²)	2887
	Power consumption (mw)	0.156

The proposed reconfigurable FIR filter architecture shown in Fig. 2 has also been coded using Verilog and synthesized in the FPGA platform using Xilinx ISE 9.2i synthesis tool. Table III depicts the results of the implementation of the proposed design and those of other designs on FPGA devices for different filter descriptions.

TABLE III
FPGA IMPLEMENTATION RESULTS FOR DIFFERENT DESIGNS

Target FPGA Device	Refr.	Method used	Filter description		Slice Reg.	Slice LUT	NSLUT	% Impr. in NSLUT
			#Taps	WL				
Virtex-2 xc2v500	**Prop.**	**Prop.**	**20**	**16x17**	**317**	**848**	**42.4**	
	[13]	2- BCSE	49	16x17	N/M	1742	35.55	-19.27
	[12]	3- BCSE	20	16x17	N/M	1521	76.05	44.24
Virtex-5 xc5vlx50	**Prop.**	**Prop.**	**16**	**16x17**	**255**	**723**	**36.15**	
	[15]	DA based	16	8x8	1120	755	188.75	80.84
	[18]	MAC based	16	8x8	970	806	201.5	82.21

Since different authors considered different filter descriptions, for a fair and straightforward comparison, normalized Slice LUT (NSLUT) based on (1) has been considered.

$$\text{NSLUT} = \frac{\text{SLUT}}{\#\,\text{Taps}} \text{ X } \frac{16}{\text{Input\_WL}} \text{ X } \frac{16}{\text{coeff\_WL}} \qquad (1)$$

The results indicated in table III depict that the proposed design possesses 44.24% less number of NSLUT than the work reported in [12] based on 3-bit BCSE and 19.27% more NSLUT than the work presented in [13] based on 2-bit BCSE when implemented on the Virtex-2 FPGA device. The increase in the number of NSLUT as compared with the 2-bit FBCSE algorithm based design [13] is due to the modifications incorporated in the architecture for supporting signed decimal format data representation for its wide applicability in several systems and the additional hardware required for CLG block for the controlled addition of partial products and the offline pre-analysis part which eliminates complete redundancy of the coefficients. Irrespective of all these modifications, the proposed design achieves 80.84% and 82.21% less NSLUT than the DA based method [15] and the MAC based design [18] respectively when implemented on Virtex-5 FPGA device.

To prove the efficiency of the proposed design, we show the performance measures of the present design viz., power, area and speed in comparison with the best existing reconfigurable FIR filter implementations in the literature. For the purpose of illustration, we have designed a 20 tap linear phase symmetric reconfigurable low pass FIR filter with passband and stopband frequencies of 0.15π rad/sec and 0.25π rad/sec respectively based on the proposed algorithm. We have considered a narrow transition band FIR filter because higher order filters intended for mobile systems should have stringent channel attenuation specifications and correspondingly sharp transition bands.

The filter coefficients are generated using MATLAB Filter Visualization Tool and are directly incorporated in the Verilog code for FIR filter. Cadence tool along with 90nm CMOS technology library is used to calculate the performance measures. The performance comparison results for different designs on ASIC platform are presented in table IV.

For the reason already stated, we have normalized the area and power for per tap based on the wordlength of the input and the coefficients using (2) and (3).

$$\text{A/tap} = \frac{\text{Total Area}}{\#\,\text{Taps}} \text{ X } \frac{16}{\text{Input\_WL}} \text{ X } \frac{16}{\text{coeff\_WL}} \qquad (2)$$

$$\text{P/tap} = \frac{\text{Total Power}}{\#\,\text{Taps}} \text{ X } \frac{16}{\text{Input\_WL}} \text{ X } \frac{16}{\text{coeff\_WL}} \qquad (3)$$

Table IV shows that our proposed design achieves 72.8%, 74.58%, 66.79% and 66.37% less maximum sampling period (MSP) than the designs presented in [13], [16], [17] and [14] respectively by using efficient carry save adders for the MAT. Our proposed design has 1.5% more MSP than the work presented in [15]. But it can be seen that the filter in [15] was designed considering 8-bit input and coefficient wordlength, whereas our proposed design considers 16 bit input and 17-bit coefficient wordlength. Table IV also reveals that our proposed design has succeeded in reducing the area/tap by 68.68%, 78.21%, 12.21%, 44.27%, 87.31% and 90.68% than the designs presented in [13], [12], [16], [17], [15] and [14] respectively and also in terms of power/tap by 79.93%, 85.99%, 43.1%, 59.7%, 98.5% and 85.9% than the designs presented in [13], [12], [16], [17], [15] and [14] respectively.

TABLE IV
ASIC IMPLEMENTATION RESULTS FOR DIFFERENT DESIGNS

Refr.	Method used	Filter description		MSP (ns)	Area (um^2)	Area /tap	Power (mw)	Power /tap	% Improvement in		
		#Taps	WL						MSP	A/tap	P/tap
Our work	**Prop.**	**20**	**16x17**	**1.604**	**16961**	**798**	**0.615**	**0.029**			
[13]	2-BCSE	16	16x17	5.908	43317	2548	2.458	0.145	72.8	68.68	79.93
[12]	3-BCSE	16	16x17	N/M	62271	3663	3.533	0.207	-	78.21	85.99
[16]	MCM	28	12x12	6.31	14307	909	0.81	0.051	74.58	12.21	43.1
[17]	IT	28	12x12	4.83	22533	1432	1.14	0.072	66.79	44.27	59.7
[15]	DA	16	8x8	1.58	25163	6290	8.25	2.06	-1.5	87.31	98.5
[14]	LUT	16	16x8	4.77	68558	8569	1.66	0.207	66.37	90.68	85.9

From table IV, the percentage improvement in efficiency of proposed design over the designs presented in [13], [12], [16], [17], [15] and [14] in terms of area, power and speed respectively are depicted graphically in Figs. 3, 4 and 5.

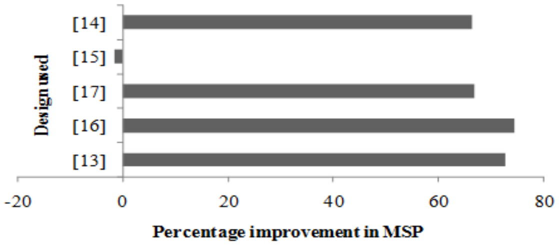


Figure 3. Comparative results of MSP

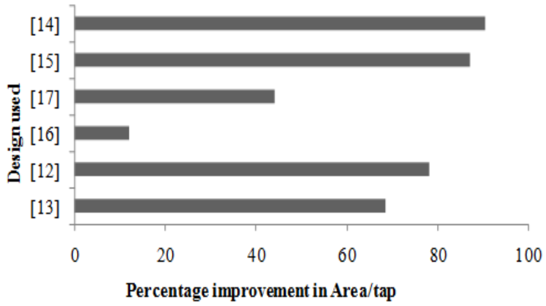


Figure 4. Comparative results of Area/tap

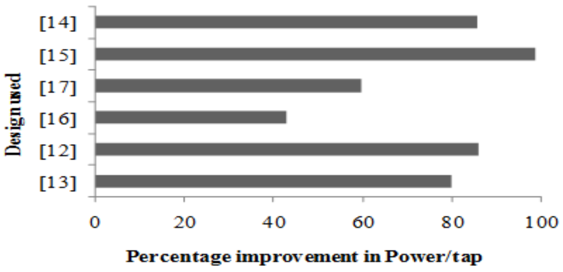


Figure 5. Comparative results of Power/tap

The proposed VHBCSE technique is compared with fixed bit vertical BSCE algorithm (FBCSE), namely 2-bit BCSE algorithm proposed in the recent work [13] and the reduction in power consumption with an increase in filter length is examined. We have redesigned the reconfigurable architecture based on [13] and applied it to a linear phase low pass FIR filter for different filter lengths. The passband and stopband frequencies of the filter are chosen to be 0.15π rad/sec and 0.25π rad/sec respectively, and the coefficients are coded using 16-bit binary precision code. The results of the comparison of the dynamic power consumption of the proposed design with that of the FBCSE based design [13] on ASIC platform are shown in table V.

TABLE V
COMPARISON OF POWER CONSUMPTION RESULTS ON ASIC PLATFORM

# Taps	Power (mw)		% Impr. in power than [13]
	Our work	FBCSE [13]	
8	0.279	0.989	71.78
12	0.474	1.7605	73.07
16	0.6129	2.4579	75.06
20	0.615	2.509	75.49
24	0.9438	3.9505	76.11
28	1.081	4.623	76.61
40	1.212	8.4495	85.66
64	1.479	13.6692	89.18

The percentage improvement in power consumption of the proposed design over the FBCSE based design [13] obtained from table 5 is depicted in Fig. 6.

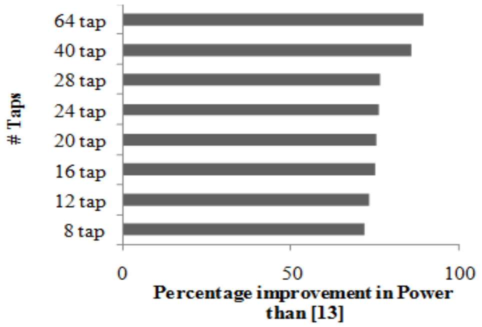


Figure 6. Comparative results of Power consumption for different filter lengths

Fig. 6 clearly reveals that the percentage improvement in power consumption for the proposed design over the FBCSE based design [13] increases as the filter length increases because a large number of common sub-expressions can be found for substantially more number of filter coefficients. This becomes highly significant because the channel filters in wireless communication transceivers are of very high orders. Hence our proposed architecture gains the upper hand for such higher order filters.

IV. CONCLUSION

We have reported a new algorithm called Vertical horizontal binary common sub-expression algorithm using signed decimal data representation for implementing a fixed point reconfigurable FIR filter. It is very evident from the implementation results that our proposed architecture outperforms in speed, area and power consumption when compared to numerous other reconfigurable FIR filter designs on ASIC and FPGA platforms, more specifically for higher order filters. Further, by combining the adders at the outputs of the multiplexers in several different ways and by employing techniques such as compression techniques for the adder tree, the proposed architecture can be optimized for best power saving and improved speed performance. Optimizability and the improvement in efficiency make the proposed reconfigurable FIR filter architecture best suited for the next generation multistandard wireless communication systems.

V. REFERENCES

[1]. T. Hentschel and G. Fettweis, "Software radio receivers," in CDMA Techniques for Third Generation Mobile Systems. Dordrecht, The Netherlands: Kluwer Academic, 1999, pp. 257–283.

[2]. J. L. Nunez-Yanez, T. Spiteri, and G. Vafiadis, "Multi-standard reconfigurable motion estimation processor for hybrid video codecs," IET Comput. Digit. Tech., vol. 5, no. 2, pp. 73–85, Mar. 2011.

[3]. D. Hwang, C. Mittelsteadt, and I. Verbauwhede, "Low power showdown: Comparison of five DSP platforms implementing an LPC speech codec," in Proc. IEEE Int. Conf. Acoust. Speech Signal Process., Salt Lake City, UT, May 2001, pp. 1125–1128.

[4]. M. Mehendale, S. D. Sherlekar, and G. Venkatesh, "Synthesis of multiplier-less FIR filters with minimum number of additions," in Proc. IEEE/ACM Int. Conf. Comput.-Aided Des., 1995, pp. 668–671.

[5]. M. Yagyu, A. Nishihara, and N. Fujii, "Fast FIR digital filter structures using minimal number of adders and its application to filter design," IEICE Trans. Fundam. Electron. Commun. Comput. Sci., vol. E79-A, no. 8, pp. 1120–1129, Aug. 1996.

[6]. R. I. Hartley, "Subexpression sharing in filters using canonic signed digit multipliers," IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., vol. 43, no. 10, pp. 677–688, Oct. 1996.

[7]. R. Pasko, P. Schaumont, V. Derudder, S. Vernalde, and D. Durackova, "A new algorithm for elimination of common subexpressions," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 18, no. 1, pp. 58–68, Jan. 1999.

[8]. C. Park and H. J. Kang, "Digital filter synthesis based on minimal signed digit representation," in Proc. DAC, Jun. 2001, pp. 468–473.

[9]. M. M. Peiro, E. I. Boemo, and L. Wanhammar, "Design of high-speed multiplierless filters using a nonrecursive signed common subexpression algorithm," IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., vol. 49, no. 3, pp. 196–203, Mar. 2002.

[10]. C. Y. Yao, H. H. Chen, C. J. Chien, and C. T. Hsu, "A novel common subexpression-elimination method for synthesizing fixed-point FIR filters," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 51, no. 11, pp. 2215–2221, Nov. 2004.

[11]. A. P. Vinod and E. M.-K. Lai, "Comparison of the horizontal and the vertical common subexpression elimination methods for realizing digital filters,"

in Proc. IEEE Int. Conf. Circuits Syst., May 2005, vol. 1, pp. 496–499.

[12]. 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.

[13]. Hatai, I. Chakrabarti, and S. Banerjee, "An efficient VLSI architecture of a reconfigurable pulse-shaping FIR interpolation filter for multi-standard DUC," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., May 2014 Online]. Available: http://ieeexplore.ieee.org

[14]. 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.

[15]. S. Y. Park and P. K. Meher, "Efficient FPGA and ASIC realizations of DA-based reconfigurable FIR digital filter," IEEE Trans. Circuits Syst. II, Exp. Brief, vol. 61, no. 7, pp. 511–515, Jul. 2014.

[16]. S. F. Hsiao, J. H. Z. Jian, and M. C. Chen, "Low-cost FIR filter designs based on faithfully rounded truncated multiple constant multiplication/ accumulation," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 60, no. 5, pp. 287–291, May 2013.

[17]. C. H. Chang, J. Chen, and A. P. Vinod, "Information theoretic approach to complexity reduction on FIR filter design," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 55, no. 8, pp. 2310–2321, Sep. 2008.

[18]. Xilinx Inc. LogiCORE IP FIR Compiler v5.0, San Jose, CA, 2010 Online]. Available: http://www.xilinx.com

[19]. 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. 217–229, Feb. 2008.