I J C T A, 9(13) 2016, pp. 5901-5909

Constant Multiplier Architecture Based on Vertical-Horizontal Binary Common Sub-Expression Elimination for FIR Filter

[1]P. Radhika, [2]G V K Chaitanya and [3]T. Vigneswaran

ABSTRACT

This paper presents an efficient constant multiplier architecture based on vertical-horizontal binary common sub-expression elimination (VHBCSE) algorithm for designing a reconfigurable finite impulse response (FIR) filter whose coefficients can rapidly change in real time. To design an efficient reconfigurable FIR filter, according to the proposed VHBCSE algorithm, 2-bit binary common sub-expression elimination(BCSE) algorithm has been applied vertically across adjacent coefficients on the 2-D space of the coefficient matrix initially, followed by applying variable-bit BCSE algorithm horizontally within each coefficient. Xilinx implementation results of multiplier show that the proposed VHBCSE algorithm is also successful in reducing the average power consumption by 9.9% along with an improvement in the area power product (APP) by 35% compared to 2-bit BCSE algorithm.

Index Terms: BCSE algorithm, MCM, reconfigurable FIR filter.

I. INTRODUCTION

FIR FILTER HAS wide application as the key component in any digital signal processing, image and video processing, wireless communication, and biomedical signal processing systems. Moreover, systems like Software Defined Radio (SDR) [1] and multi-standard video codec [2] need a reconfigurable FIR filter with dynamically programmable filter coefficients, interpolation factors and lengths which may vary according to the specification of different standards in a portable computing platform. Significant applicability of an efficient reconfigurable FIR filter motivates the system designer to develop the chip with low cost, power, and area along with the capability to operate at very high speed. In any FIR filter, the multiplier is the major constraint which defines the performance of the desired filter. Therefore, over the past three decades, design of an efficient hardware architecture for fixed point FIR filter has been considered as the major research focus as reported in published literatures [3]–[6].

In FIR filter, the multiplication operation is performed between one particular variable (the input) and many constants (the coefficients) and known as the multiple constant multiplication (MCM).

The algorithms proposed earlier to implement this MCM for an efficient FIR filter design can be categorized in two main groups: 1) graph based algorithms and 2) common sub-expression elimination (CSE) algorithms [7]–[8]. Most of these graph based or CSE algorithms presented earlier are used to obtain efficient FIR filter hardware architecture by running the algorithms on a particular (fixed) set of coefficients for some time (a couple of hours to days) on a highly efficient computing platform (like using 1–20 number of 3.2 GHz computers in parallel mode as mentioned in [4]). However, FIR filter implementation employing effective MCM design by running these algorithms on a fixed set of coefficients is not suitable for the application like SDR system because of the following two reasons:

[1] Assistant Professor, SRM University,
[2] M.Tech VLSI Design, SRM University,
[3] Professor, VIT University

1) coefficient of the filters in SDR system are dynamically programmable based on requirement of different standards and
2) highly computationally efficient platform needed for those algorithms is unaffordable in SDR system. Some techniques have been introduced for efficient reconfigurable constant multiplier design [9], [10] for any application where the filter's coefficients are changing in real time e.g. multi-standard digital up/down converter.

Binary common sub-expression elimination (BCSE) algorithm is one of those techniques, which introduces the concept of eliminating the common sub-expression in binary form for designing an efficient constant multiplier, and is thus applicable for reconfigurable FIR filters with low complexity [5]. However, the choice of the length of the binary common sub-expressions (BCSs) in [5] makes the design inefficient by increasing the adder step and the hardware cost. The efficiency in terms of speed, power, and area of the constant multiplier has been increased in the work presented in [6] while designing one reconfigurable FIR filter for multi-standard DUC by choosing 2-bit long BCS judiciously. Choice of the BCS of fixed length (3-bitor2-bit) in the earlier proposed BCSE algorithm based reconfigurable FIR filter designs [5], [6] leaves a scope to optimize the designed filter by considering the BCS across the adjacent coefficients as well as within a single coefficient. The convention considered for representing the input and the coefficient of the earlier designed FIR filter [5], [6] assigned magnitude format also gives a scope to modify the data representation to signed decimal number for wider applicability of the proposed FIR filter in any systems. On studying the above-mentioned literatures, it has been realized that the development of an efficient reconfigurable constant multiplier is very much needed for its applicability in any reconfigurable system. The organization of the paper is as follows. Basic concepts along with the complexity analyses of fixed bit BCSE (FBCSE), i.e., 2- bit BCSE algorithms proposed in the earlier literatures have been discussed. The Step wise flow chart along with the complexity analysis of the pro-posed VHBCSE algorithm based constant multiplier has been presented. Step wise flow chart along with the complexity analysis of the pro-posed VHBCSE algorithm based constant multiplier has been presented.

II. CONCEPTS AND COMPLEXITY ANALYSES OF BCSE ALGORITHMS

Considering the coefficients in binary pattern, the BCSE algorithms described in [5], [6] attempt to eliminate the redundant computation vertically by considering 3-bit or 2-bit BCS present across the adjacent coefficients. As defined and explained in [10] and [11], horizontal BCSE algorithm utilizes CSs occurring within each coefficient to get rid of redundant computations, while vertical BCSE uses CSs found across adjacent coefficients to eliminate redundant computations. According to BCSE algorithm a total of BCSs can be formed out of an n-bit binary number and the number of adders required to generate the partial products for n-bit BCS is [10]. In general, the adder step in BCSE algorithm which defines the critical path can be calculated as , where n is the number of non-zero elements present within the coefficients.

In a reconfigurable constant multiplier, the coefficient values can be dynamically programmable. Therefore, the idea behind the reconfigurable multiplier is to consider the worst case (which involves the largest number of addition steps) where by all the relatively better cases will also be taken care of. Hence, considering a reconfigurable multiplier having 16-bit input (X) and the 16-bit coefficient (H), the worst case condition will occur for the coefficient of values 16'HFFFF. Shift and add based multiplication operation between the inputs (X) with this coefficient (16'HFFFF) values can be written as

$$X * H = \frac{X}{2} + \frac{X}{4} + \frac{X}{8} + \frac{X}{16} + \frac{X}{32} + \frac{X}{64} + \frac{X}{128} + \frac{X}{256} + \frac{X}{512} + \frac{X}{1024} + \frac{X}{2048} + \frac{X}{4096} + \frac{X}{8192} + \frac{X}{16384} + \frac{X}{32768} + \frac{X}{65636} \quad (1)$$

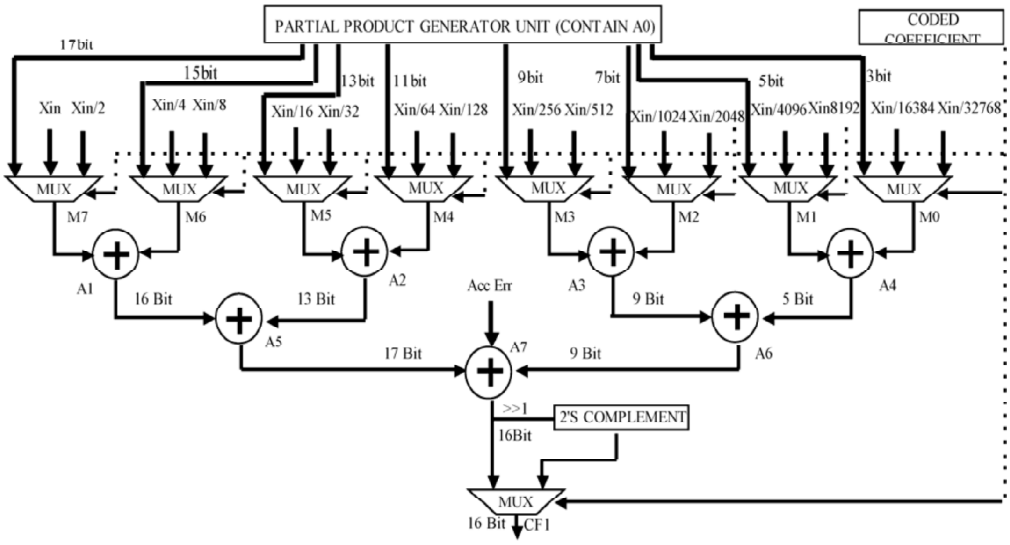


Figure 1: Reconfigurable constant multiplier using 2-bit bcse algorithm

2-Bit BCSE Algorithm [6]: Considering a 2-bit BCS, the partial-product generated from each BCS will be as

$$X1 = X + \frac{X}{2} \tag{2}$$

Substituting (2) in (1) gives

$$X * H = \frac{X}{2} + \frac{X}{4} + \frac{X}{8} + \frac{X}{16} + \frac{X}{32} + \frac{X}{64} + \frac{X}{128} + \frac{X}{256} + \frac{X}{512} + \frac{X}{1024} + \frac{X}{2048} + \frac{X}{4096} + \frac{X}{8192} + \frac{X}{16384} + \frac{X}{32768} \tag{3}$$

Fig. 1 depicts a reconfigurable constant multiplier that can be realized in hardware using (3). The eight terms appearing on the right hand side of (3) correspond to the eight partial products (shown as M7-M0 in Fig. 1) generated by the 2-bit BCSE algorithm given in [6]. These are summed up by the multiplier adder tree (MAT) (shown as A1-A7 in Fig. 1), leading to the product according to (3). The name MAT stems from the tree-like configuration of adders used to realize multiplication as depicted in Fig. 1.

Proposed Solutions: The algorithm proposed in this paper to solve the above addressed problems consists of the following steps:

1) At first, the filter coefficient has been multiplexed between its original and complemented values depending on the most significant bit (MSB) of the coefficient to support the signed decimal data representation. This technique helps in reducing the hardware complexity when the coefficients consist of small negative decimal numbers.
2) According to the proposed algorithm in layer-1 of MAT, the 2-bit BCSE has been applied vertically followed by conditional 4-bit and 8-bit BCSEs horizontally in layer-2 and layer-3 respectively to find out the common sub-expressions (CSs) present within the coefficients. This technique helps in solving the additional hardware consumption problem by eliminating more CSs.
3) Extending the BCSE in the lower level or applying the BCSE horizontally will reduce the probability of use of the MB adders present in the lower levels of the MAT. This will reduce the power consumption to a great extent by lowering the switching activities of these MB adders.

4) Our proposed technique can solve the problem of high power and area consumption problem for both the cases, viz. small valued negative coefficient and high valued positive coefficient. The proposed design also supports the data representation in signed decimal format. Apart from that, area and power efficient reconfigurable FIR filter can be obtained during the high level synthesis procedure using constant multiplier based on proposed VHBCSE algorithm. All of these observations make the proposed constant multiplier an excellent candidate for designing any (higher or lower) order efficient reconfigurable FIR filter.

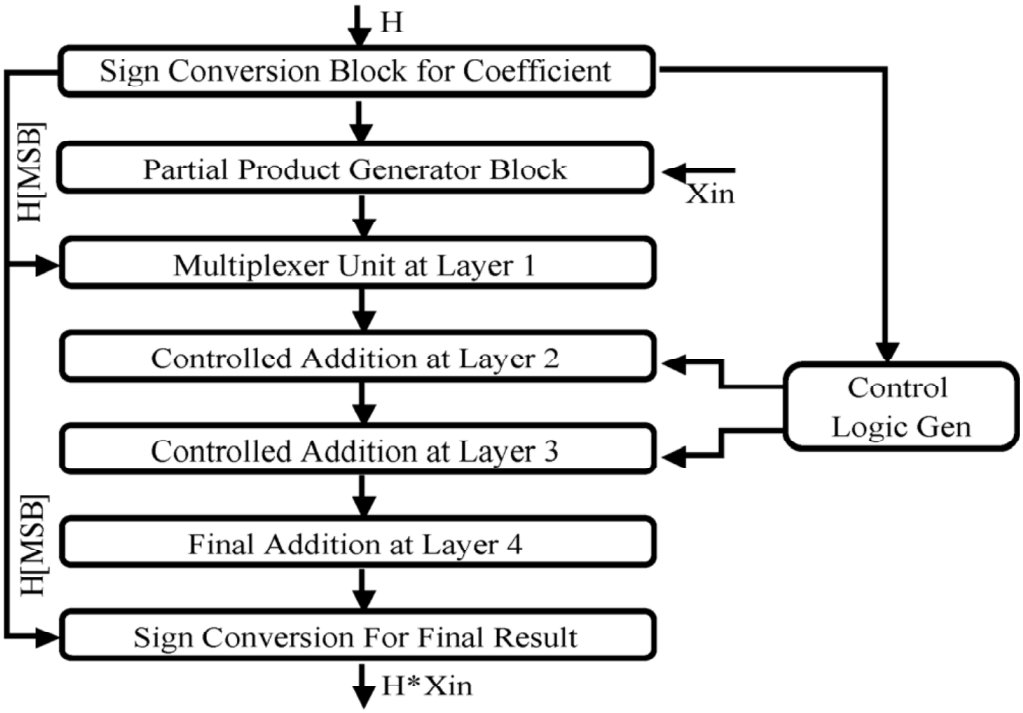


Figure 2: Data flow diagram of the CM using VHBCSE algorithm

III. PROPOSED VERTICAL-HORIZONTAL BCSE ALGORITHM

Vertical and horizontal BCSEs are the two types of BCSE used for eliminating the BCSs present across the adjacent coefficients and within the coefficients respectively in any BCSE method. Vertical BCSE produces more effective BCS elimination than the horizontal BCSE as shown in [12]. However, this paper proposes one new BCSE algorithm which is a combination of vertical and horizontal BCSE for designing an efficient reconfigurable FIR filter. In our proposed algorithm, a 2-bit vertical BCSE has been applied first on the adjacent coefficient, followed by 4-bit and 8-bit horizontal BCSEs to detect and eliminate as many BCSs as possible which are present within each of the coefficient. The procedure of application of the proposed algorithm for designing the above mentioned (in Section II) 8-Tap symmetric FIR filter has been depicted graphically in Fig. 4. Here it can be noted that the coefficients of the FIR filter form a 2-D matrix where each row represents a single coefficient and the columns correspond to individual bits of the coefficients. Application of 2-bit VCSE to these filter coefficients to generate the partial products requires one adder of 17 full adder cells. Application of 4-bit and 8-bit HCSE considering H0 as one input to the designed multiplier finds no match for the 4-bit and 8-bit BCS within the H0 coefficient. Therefore this multiplier requires 1, 2, 1, 2, and 1 number of adders consisting of 17-bit, 16-bit, 13-bit, 9-bit, and 5-bit respectively, a total number of 85 full adder cells to sum up the partial products. Whereas, the constant multiplier considering the coefficient H1, H2, and H3 as one input can be implemented by using only 3 adders each consisting of 1 adder of 17-bit length and 2 adders of 16-bit length as 4-bit and 8-bit BCS present within each of these coefficients, requires a total number of 49 full adder cells. Hence, total requirement of the full

adder cells amounts to 249 (=1 17 (for the partial product generation of 2-bit BCs)+85(H0)+49 (H1)+49 (H2)+49 (H3))

IV. ARCHITECTURE OF THE PROPOSED VHBCSE ALGORITHM BASED CONSTANT MULTIPLIER

The data flow diagram of the proposed vertical horizontal BCSE algorithm based constant multiplier (CM) design is shown in Fig. 2. The designed multiplier considers the length of the input (Xin) and coefficient (H) as 16-bit and 17-bit respectively while the output is assumed to be 16-bit long.

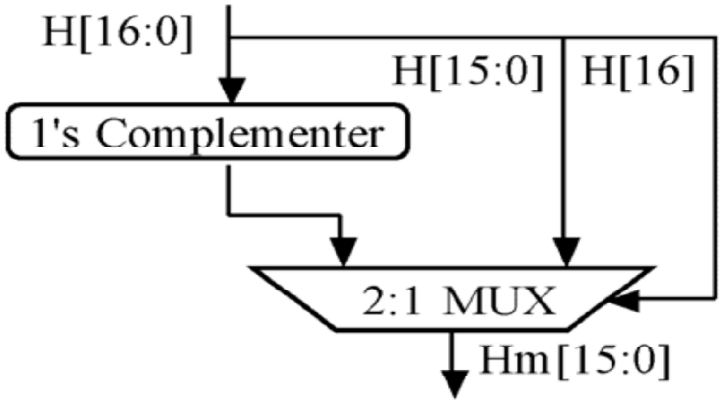


Figure 3: Hardware architecture of the sign conversion block

Here in, the sampled inputs are stored in the register first and then the coefficients are stored directly in the LUTs. Functionality along with hardware architecture of different blocks of the designed VHBCSE based multiplier are explained below in details.

1) *Sign Conversion Block:* Sign conversion block is needed to support the signed decimal format data representation for both the input and the coefficient. The architecture of the sign conversion block is shown in Fig. 2. There is one 1's complement circuit to generate the inverted version of the 16-bit (excluding MSB) coefficient. One 16-bit 2:1 multiplexer produces the multiplexed coefficients depending on the value of the most significant bit (MSB) of the coefficient. For negative value of the original coefficient, the multiplexed coefficient will be in the inverted form; otherwise it will be as it is.

2) *Partial Product Generator (PPG):* In BCSE method, shift and add based technique has been used to generate the partial product which will be summed up in the following steps/layers for producing the final multiplication result. Choice of the size of the BCS defines the number of partial products. In the proposed algorithm in the layer-1, 2-bit binary common sub-expressions (BCSs) ranging from "00" to "11" have been considered, which will produce 4 partial products. But, within four of these BCSs, a single adder (A0) will be required to generate the partial product only for the pattern

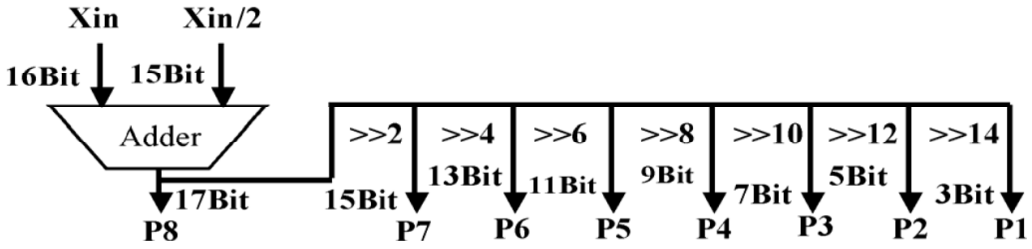


Figure 4: Block diagram of the partial product generator

"11"; the rest will be generated by hardwired shifting. For the coefficient of 16-bit length, 8 partial products of 17, 15, 13, 11, 9, 7, 5, and 3 bits (P8-P1) will be generated by right shifting the first partial product (P8) by 0, 2, 4, 6, 8, 10, 12, and 14 bits binary value. The architecture of this block is shown in Fig. 4.

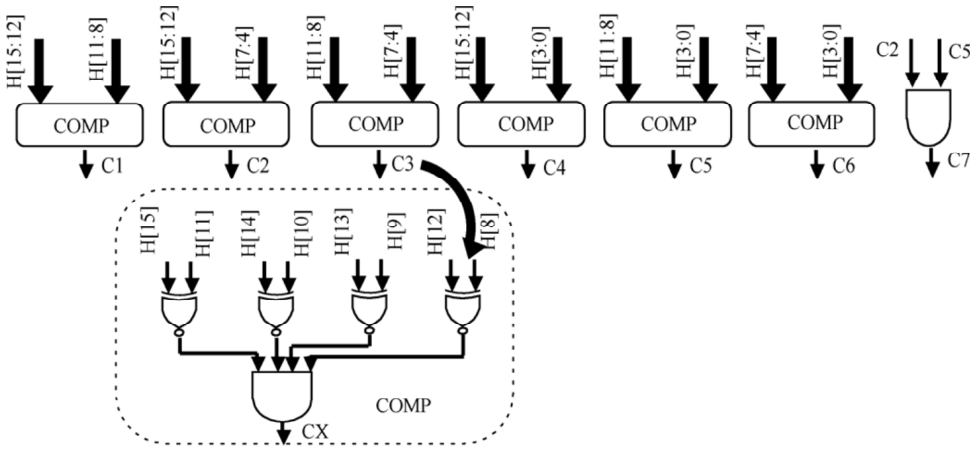


Figure 5: Block diagram of the control logic generator

3) *Control Logic (CL) Generator:* Control logic generator 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]). According to the algorithm mentioned in Section IV, the CL generator block will produce 7 control signals depending on the equality check for 7 different cases. The architecture for the control signal generator block is shown in Fig. 5. The control signal for 8-bit equality check is seen to be produced through the control signals generated from the 4-bit equality check.

4) Multiplexers Unit: The multiplexer unit is used to select the appropriate data generated from the PPG unit depending on the coefficient's binary value. At layer-1, eight 4:1 multiplexers are required to produce the partial products according to the 2-bit BCSE algorithm applied vertically on the MAT. The widths of these 8 multiplexers are 17, 15, 13, 11, 9, 7, 5, and 3-bit each instead of 16-bit for all, which would reduce the hardware and power consumption. The architecture of the multiplexers unit is shown in Fig. 8.

5) *Controlled Addition at Layer-2:* The partial products (PP) generated from eight groups of 2-bit BCSs are added up for the final multiplication results which have been performed in three layers.

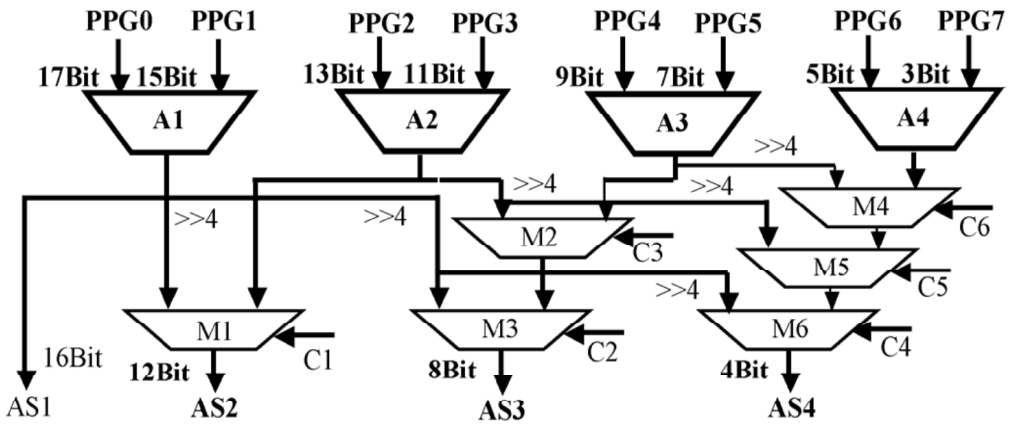


Figure 6: Architectural details of the controlled addition at layer-2 block.

According to the BCSE algorithm [6] proposed earlier, layer-2 requires four addition (A1-A4) operations to sum up the eight PPs. Instead of direct addition of these PPs, the controlled addition operations are performed at layer 2 according to the proposed VHBCSE algorithm. These adders (A1-A4) are controlled depending on the control signals (C1-C6), which were generated based on 4-bit BCSE from the control signal generator block. The architecture of this block is shown in Fig. 6, which reveals that the propagation delay will be the maximum between the paths which has been used to generate AS2, AS3, AS4, i.e.,

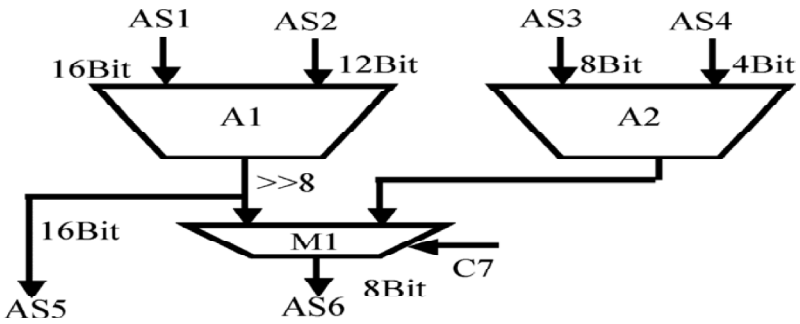


Figure 7: Hardware architecture of the controlled addition at layer-3.

6) *Controlled Addition at Layer-3:* The four multiplexed sums (AS1, AS2, AS3 and AS4) generated from layer-2 are now summed up in layer-3. In our algorithm, controlled additions are performed, instead of direct addition of these four sums as shown in Fig. 7. Hence, this addition (A6) is controlled by the control signal (C7) which has been generated based on 8-bit BCSE from the CS generator block. From Fig. 7 it is concluded that the propagation delay will be

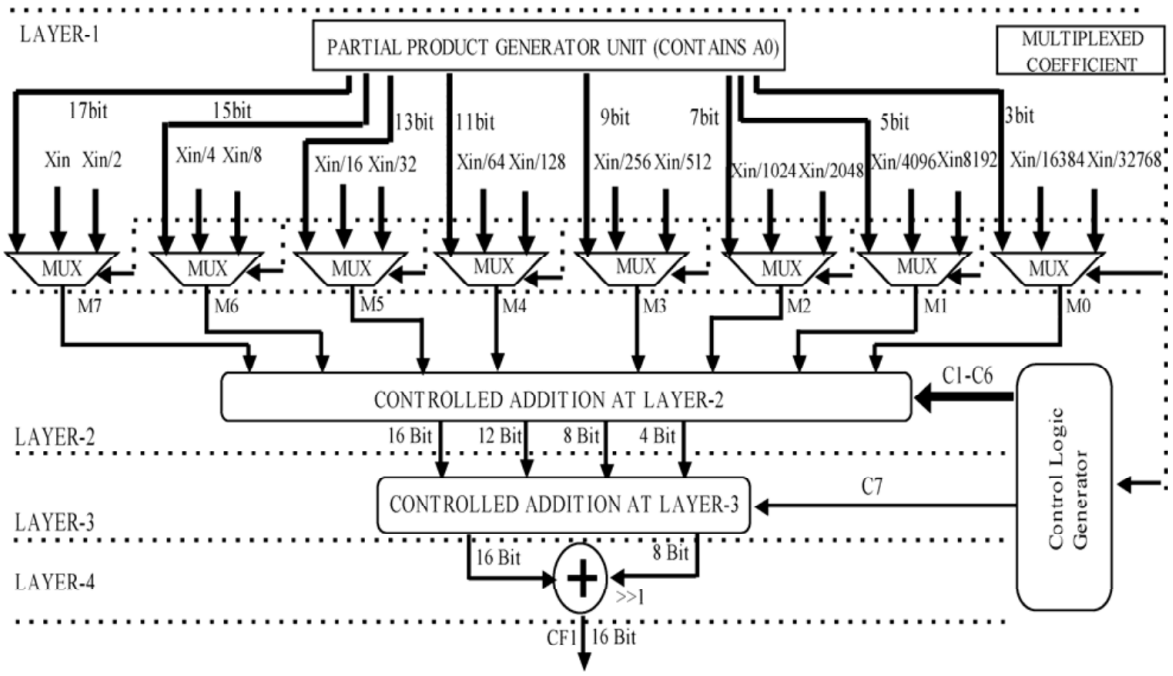
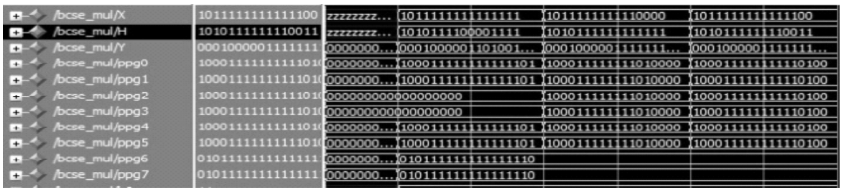


Figure 8: Proposed Reconfigurable constant multiplier architecture.

7) *Final Addition on Layer-4:* This block performs the addition operation between the two sums (AS5-AS6) produced by layer-3 to finally produce the multiplication result between the input and the coefficient. The block diagram of the over-all constant multiplication is shown in Fig. 8.

V. SIMULATION RESULTS

Multiplier output by using 2bit BCSE



Multiplier output by using VHBCSE

Comparison table of 2bit BCSE and VHBCSE

Algorithm	*cells*	*Area*	*Total power*
2bit BCSE	37	152.410	2936.347
VHBCSE	36	152	2674.924

VI. CONCLUSION

This paper presents one new vertical-horizontal BCSE algorithm which removes the initial common sub-expressions (CSs) by applying 2-bit BCSE vertically. Further elimination of the CSs has been performed through finding the CSs present within the coefficients by applying BCSEs of different lengths horizontally to different layers of the shift and add based constant multiplier architecture. It has been shown that the proposed algorithm successfully reduces the average switching activities of the multiplier block adder by 7.5% while compared to those of 2-bit BCSE. The proposed VHBCSE algorithm is also successful in reducing the average power consumption by 9.9% and establishes improvements of efficiency of 35% in area power product (ADP) when compared to those of earlier proposed MCM algorithm based FIR filter.

REFERENCES

[1] 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 Integr. (VLSI) Syst.*, vol. 22, no. 5, pp. 1202–1206, May 2014.

[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] H. Samueli, "An improved search algorithm for the design of multiplier less FIR filters with power-of-two coefficients," *IEEE Trans. Circuits Syst.*, vol. 36, no. 7, pp. 1044–1047, Jul. 1989.

[4] C. Y. Yao, W. C. Hsia, and Y. H. Ho, "Designing hardware-efficient fixed-point FIR filters in an expanding subexpression space," *IEEE Trans. Circuits and Systems I, Reg. Papers*, vol. 61, no. 1, pp. 202–212, Jan. 2014.

[5] 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.

[6] I. 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

[7] Y. C. Lim, J. B. Evans, and B. Liu, "Decomposition of binary integers into signed power-of-two terms," *IEEE Trans. Circuits and Systems*, vol. 38, no. 6, pp. 667–672, Jun. 1991.

[8] Y. Pan and P. K. Meher, "Bit-level optimization of adder-trees for multiple constant multiplications for efficient FIR filter implementation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 2, pp. 455–462, Feb. 2014.

[9] A. P. Vinod, E. Lai, D. L. Maskell, and P. K. Meher, "An improved common subexpression elimination method for reducing logic operators in FIR filter implementations without increasing logic depth," *Integr. VLSI J.*, vol. 43, no. 1, pp. 124–135, Jan. 2010.

[10] A. P. Vinod and E. M. K. Lai, "An efficient coefficient partitioning algorithm for realizing low-complexity digital filters," *IEEE Trans. Comput. Aided Design Integr. Circuits Syst.*, vol. 24, no. 12, pp. 1936–1946, Dec. 2005.

[11] 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.

[12] Y. Jang and S. Yang, "Low-power CSD linear phase FIR filter structure using vertical common sub-expression," *Electron. Lett.*, vol. 38, no. 15, pp. 777–779, Jul. 2002.