





## Design and Implementation of Area and Power Efficient Reconfigurable FIR Filter with Low Complexity Coefficients

Baboji KILLADI<sup>1</sup> , Sridevi SRIADIBHATLA<sup>1,\*</sup> 

<sup>1</sup>School of Electronics Engineering, Vellore Institute of Technology, Vellore, India

### Article Info

Received: 13/08/2018

Accepted: 19/02/2019

### Keywords

BCSE algorithm  
Binary complexity  
Reconfigurable FIR filter  
VHBCSE algorithm

### Abstract

This paper presents the design and implementation of area and power efficient reconfigurable finite impulse response (FIR) filter. We present a method for designing a reconfigurable filter with low binary complexity coefficients (LBCC) and thus to optimize the filter while satisfying the design specifications. The total number of non zero binary bits is taken as a measure of the binary complexity (BC) of a coefficient. We propose two implementation architectures namely signed-magnitude architecture (SMA) and signed-decimal architecture (SDA) which are based on 3-bit binary common sub expression elimination (BCSE) algorithm and vertical horizontal BCSE (VHBCSE) algorithm respectively. SMA and SDA reduce the redundant computations of the coefficient multiplications in the filter. The proposed filters are synthesized on tsmc 65nm CMOS technology. The synthesis results show that the proposed filters are area and power efficient when compared with the existing ones.

## 1. INTRODUCTION

Finite impulse response (FIR) filters are most commonly used in applications like channelization, channel equalization and pulse shaping because of their powerful design methods, ease of implementations, underlying stability and linear-phase properties. With increasing demand in mobile systems, low power and high speed implementation of FIR filters is emerged as an imperative requirement. With rapid developments in the field of software defined radio (SDR) [1-5] technology, realization of reconfigurable FIR filters has been focused. Flexibility obtained through reconfigurable digital filters which occupied the position of analog signal processing systems, enabled SDR [1-5] to support multiple wireless communication standards. Each communication standard supported by SDR [1-5] has a distinct filter described by a unique coefficient set. When the SDR [1-5] system changes its mode of operation to a different standard, the coefficient set corresponding to that standard need to be loaded. Hence the reconfigurable FIR filter whose coefficients change dynamically plays a paramount role in SDR [1-5] systems and thus low power and high throughput realization of reconfigurable filters is the current area of interest.

Many researchers have focused on implementing reconfigurable FIR filters with low power, low area and high speed. A canonic signed digit (CSD) [6-8] based digit reconfigurable FIR filter architecture is introduced in [9]. Here, the authors focused on reducing the complexity of the FIR filter by reducing the precision of the coefficients while ensuring no changes in the performance. But this architecture consumed more power and utilized more hardware for its realization. Tang et al., [10] presented a high speed programmable CSD [6-8] based FIR filter. This architecture is comprised of booth encoding scheme, Wallace adder tree and carry look-ahead adder. It offered high speed while consuming more power. The authors in [11] presented filtering operation as vector scaling operation. The concept is to pre-evaluate the values such as  $x$ ,  $3x$ ,  $5x$ ,  $7x$ ,  $9x$ ,  $11x$ ,  $13x$  and  $15x$  where  $x$  is the input and thereafter reuse these pre-

evaluations effectively using multiplexers. The presence of multiplexers facilitates the reconfigurability [11]. The method proposed in [11] was modified in [12] where a new carry-save adder and conditional capture flip-flops were used to improve the power and performance. The architectures [9-12] were effective only for lower order filters.

Multiple constant multiplications (MCM) [13-14] based methods which were suitable for implementation of both lower order and higher order filters were discussed in [15-17]. MCM [13-14] involve multiplications of one variable with many constants. The authors in [15] used 3-bit binary common sub expression (BCSE) algorithm to implement the reconfigurable filters. BCSE algorithm reused the frequent binary common sub expressions (BCSs) to eliminate redundant computations in coefficient multiplications and thus it helped to implement the efficient reconfigurable filters. I. Hatai et al., in [16], further improved the reconfigurable filter by considering 2-bit BCSE algorithm which eliminated the redundant adder presented in coefficient multiplications. The reconfigurable filter was further improved in [17] by vertical horizontal BCSE (VHBCSE) algorithm which eliminates the redundant computations within the coefficients as well as among the adjacent coefficients.

In [15-17], the filters are designed using Parks-McClellan algorithm (`firpm` command in MATLAB) and implemented using shift-add architectures. The methodologies of the filters in [15] and [17] are shown in Figure 1. The authors focused on optimizing the reconfigurable filters using sub expression elimination algorithm based architectures. They did not focus on the optimization that can be achieved in filter design phase.

In the present work, we investigate a method which optimizes the filter in design phase. The methodology of the proposed filter is shown in Figure 2. It is initiated by the design of a nominal filter for given specifications followed by greedy randomized heuristic [18] which updates the coefficients of the nominal filter consecutively in random manner for filter design with low binary complexity coefficients (LBCC). The heuristic runs for several times. The best results i.e., coefficients with much smaller binary complexities is taken as final filter design. It completes the design of the proposed filter. Thereafter proposed filter is implemented with 3-bit BCSE and VHBCSE architectures of [15] and [17]. The difference between the proposed filter and filters in [15] and [17] majorly lies in design process.

The two reconfigurable filter architectures proposed in the present work are signed-magnitude architecture (SMA) and signed-decimal architecture (SDA). SMA supports signed-magnitude representation where as SDA supports signed-decimal representation.

The organization of the paper is as follows. Section 2 presents the proposed design of the reconfigurable FIR filter with LBCC. The proposed implementation architectures are explained in Section 3. Simulation and synthesis results are discussed in Section 4. Section 5 explains the conclusion.

## 2. PROPOSED DESIGN OF THE RECONFIGURABLE FIR FILTER WITH LBCC

Let us consider the design of a FIR filter for given specifications such as pass band frequency ( $\omega_p$ ), stop band frequency ( $\omega_s$ ), pass band ripple and stop band attenuation. A filter design is described by its coefficient vector  $\theta$  which is a vector of  $q$  real coefficients i.e.,  $\theta \in R^q$ .

The aim of the proposed design is to determine the coefficient vector  $\theta$  with lowest binary complexity (BC) while satisfying the given specifications.

The BC of a coefficient vector  $\theta$  is defined as the total number of non zero bits in its coefficients and it is measured by the function [18]  $\Phi: R^q \rightarrow R$

$$\Phi_{bin}(\theta) = \sum_{i=1}^q \phi_{bin}(\theta_i) \quad (1)$$

Where  $\phi_{bin}(\theta_i)$  provides, the BC of the  $i^{th}$  coefficient of the coefficient vector  $\theta$  and is given by

$$\phi_{bin}(\theta_i) = \sum_{i=-U}^V b_i \tag{2}$$

Where  $U$  and  $V$  are non negative integers and  $b_i$  are the bits in the binary expansion of  $\theta_i$ .  $U + 1$  and  $V$  give the number of bits in integer and fractional parts of the binary expansion of  $\theta_i$  respectively.

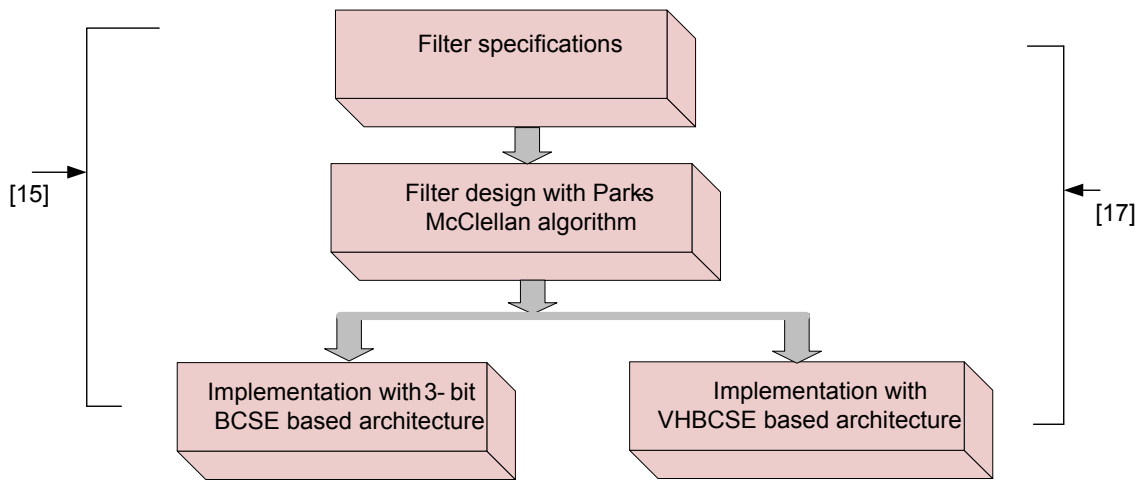


Figure 1. Methodology of filters in [15] and [17]

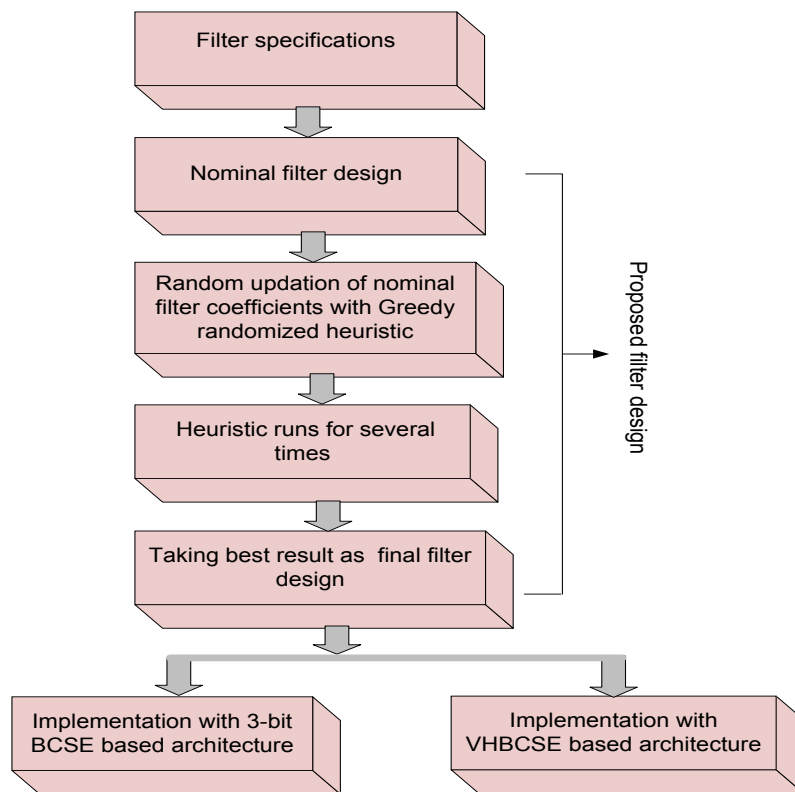


Figure 2. Methodology of proposed filter

## 2.1. Design Procedure

Initially a nominal filter described by its coefficient vector  $\theta_{nom} \in R^q$  is formed. The coefficients of the nominal filter are obtained using convex optimization [19] while maximizing stop band attenuation corresponding to the maximum pass band ripple. Thereafter greedy randomized heuristic [18] is used to determine  $\theta$  with least BC from  $\theta_{nom}$ . It starts by taking the coefficients of the nominal filter. It is formulated into passes and in each pass the individual coefficients are greedily truncated sequentially in random order for low binary complexity. The heuristic stops when the coefficient vector  $\theta$  does not alter around one pass. As this algorithm is random and can be converged differently in different runs, it is repeated for many times and the coefficient vector  $\theta$  with least BC is taken as final filter design.

Let us consider the design of a low-pass filter of length 31 with passband, stopband, maximum passband ripple and minimum stopband attenuation given by  $[0, 0.2\pi]$ ,  $[0.3\pi, \pi]$ ,  $\pm 0.2\text{dB}$  and  $28\text{dB}$  respectively. The coefficients of the nominal filter are obtained using convex optimization [19] by maximizing the minimum stop band attenuation corresponding to the maximum pass band ripple of  $\pm 0.17\text{dB}$ .

$V=16$  fractional bits are used to represent the coefficients. The BC of the nominal filter is  $\Phi_{bin}(\theta_{nom}) = 171$ . The greedy randomized heuristic [18] runs for 100 times. The best result of complexity  $\Phi_{bin}(\theta) = 107$  is the proposed filter design. It is about 3.45 non zero bits per coefficient.

In this way, specifications of the multiple communication standards supported by reconfigurable filter are individually designed for low complexity.

We designed FIR filters of taps 17, 19 and 31, each with four different specifications. We use three different coefficient lengths for the simulations. The results are tabulated from Table 1 through Table 5.

Table 1 shows that for a 19-tap filter with coefficient length of 16 bits, the proposed filter design offers an average reduction of 46.92% in BC over [15]. In a similar way, from Table 2, it can be observed that the proposed design of 31-tap filter with coefficient length of 16 bits allows average reduction of 48% in BC when compared to [15]. Table 3 provides simulation results of 31-tap filter with 8 bit coefficient precision in comparison to [15]. From Table 3, it is evident that the BC of the proposed design is 18.65% less than that of [15]. From the Tables 2 and 3, we can notice that the average reduction in BC is decreasing with the coefficient precision.

**Table 1.** Results of 19-tap filter design with coefficient precision of 16

Pass band frequency ( $\omega_p$ ), Stop band frequency ( $\omega_s$ )	Filter design BC [15]	Nominal filter BC	Proposed filter BC	%Reduction in BC
$\omega_p = 0.1\pi, \omega_s = 0.12\pi$	128	142	48	62.50
$\omega_p = 0.15\pi, \omega_s = 0.2\pi$	128	120	84	34.38
$\omega_p = 0.2\pi, \omega_s = 0.22\pi$	125	127	69	44.80
$\omega_p = 0.2\pi, \omega_s = 0.3\pi$	137	114	74	45.98
				Avg = 46.92

**Table 2.** Results of 31-tap filter design with coefficient precision of 16

Pass band frequency ( $\omega_p$ ), Stop band frequency ( $\omega_s$ )	Filter design BC [15]	Nominal filter BC	Proposed filter BC	%Reduction in BC
$\omega_p = 0.1\pi, \omega_s = 0.12\pi$	180	185	89	50.55
$\omega_p = 0.15\pi, \omega_s = 0.2\pi$	168	197	83	50.59
$\omega_p = 0.2\pi, \omega_s = 0.22\pi$	194	207	91	53.09
$\omega_p = 0.2\pi, \omega_s = 0.3\pi$	172	171	107	37.79
				Avg = 48.01

Similarly the results in Table 4 demonstrates that the proposed design of 17-tap filter with coefficient precision of 10 bits provides the average reduction in the BC by 29.62% when compared with [15].

From the results, it is noticed that the binary complexities (BCs) of the proposed design are much smaller than that of [15]. As the implementation of the filters with smaller BCs consume less power and occupies less area than the filters with higher BCs, the proposed filter implementations consumes less power and smaller area during high level synthesis when compared to [15].

**Table 3.** Results of 31-tap filter design with coefficient precision of 8

Pass band frequency ( $\omega_p$ ), Stop band frequency ( $\omega_s$ )	Filter design BC [15]	Nominal filter BC	Proposed filter BC	%Reduction in BC
$\omega_p = 0.1\pi, \omega_s = 0.12\pi$	63	65	51	19.04
$\omega_p = 0.15\pi, \omega_s = 0.2\pi$	57	66	46	19.29
$\omega_p = 0.2\pi, \omega_s = 0.22\pi$	80	78	59	26.25
$\omega_p = 0.2\pi, \omega_s = 0.3\pi$	50	45	45	10.00
				Avg = 18.65

**Table 4.** Results of 17-tap filter design with coefficient precision of 10

Pass band frequency ( $\omega_p$ ), Stop band frequency ( $\omega_s$ )	Filter design BC [15]	Nominal filter BC	Proposed filter BC	%Reduction in BC
$\omega_p = 0.1\pi, \omega_s = 0.12\pi$	72	68	40	44.44
$\omega_p = 0.15\pi, \omega_s = 0.2\pi$	61	64	44	27.86
$\omega_p = 0.2\pi, \omega_s = 0.22\pi$	66	63	43	34.84
$\omega_p = 0.2\pi, \omega_s = 0.3\pi$	53	65	47	11.32
				Avg = 29.62

The results also demonstrate that higher reductions in BCs are possible for higher coefficient lengths. Hence the proposed design is more effective for the implementation of filters with higher coefficient lengths. It can be seen with the synthesis results in the next section.

The resulting low complexity coefficient vectors from the proposed design are placed in the look up tables (LUTs) of the proposed architectures to perform the required filtering operation.

### 3. PROPOSED RECONFIGURABLE FILTER ARCHITECTURES

Here we propose a reconfigurable filter architecture that supports  $n$  communication standards as shown in Figure 3. The traditional transposed form FIR filter is application specific where coefficient set is fixed. To achieve reconfigurability, the coefficient set of the filter should be changed depending upon the selection of a specific standard. The selection of a given standard and dynamic loading of corresponding coefficient set from LUTs is enabled by a mode select input  $M\_sel$  of length  $j = \lceil \log_2 n \rceil$ . We use a coefficient segmentation procedure to select the coefficients to be stored in each LUT.

#### 3.1. Coefficient Segmentation

In order to do the coefficient segmentation, a matrix  $K$  of size  $n \times m$  is formed where  $n$  is the number of standards and  $m$  is the filter length. The matrix is arranged such that each row contains the coefficient set of one specification. The matrix is given below.

$$K = \begin{bmatrix} C_{1,1} & C_{1,2} & \cdots & C_{1,m} \\ C_{2,1} & C_{2,2} & \cdots & C_{2,m} \\ \vdots & \vdots & \cdots & \vdots \\ C_{n,1} & C_{n,2} & \cdots & C_{n,m} \end{bmatrix} \quad (3)$$

The columns of a matrix  $K$  are reversed with the help of Exchange matrix  $E$ . In order to do this, the matrix  $K$  in Equation (3) is multiplied with the Exchange matrix  $E$  of size  $m \times m$ . The resulting matrix  $P$  is known as the coefficient segmentation matrix and is given by

$$P = K \times E = \begin{bmatrix} C_{1,m} & C_{1,m-1} & \dots & C_{1,1} \\ C_{2,m} & C_{2,m-1} & \dots & C_{2,1} \\ \vdots & \vdots & \dots & \vdots \\ C_{n,m} & C_{n,m-1} & \dots & C_{n,1} \end{bmatrix} \tag{4}$$

Where  $C_{1,1}, C_{1,2}, C_{1,3}, \dots, C_{1,k}, \dots, C_{1,m}$  indicates the coefficients of the first standard,  $C_{2,1}, C_{2,2}, C_{2,3}, \dots, C_{2,k}, \dots, C_{2,m}$  indicates the second standard coefficients and so on. Each column is split into one segment. Thus the total matrix is split into  $m$  segments. Each segment is a column matrix of size  $n \times 1$ . The coefficients in  $j^{th}$  segment of coefficient segmentation matrix will be stored in  $j^{th}$  LUT of the proposed filter.

Shift-add unit (SAU) is used to generate partial products of the multiplication and processing element (PE) is used to perform the multiplication operation with the help of SAU. PE -  $i$  performs  $i^{th}$  coefficient multiplication with the help of  $i^{th}$  LUT and SAU.

The architectures of the SAU and PE which are different for proposed SMA and SDA are explained below.

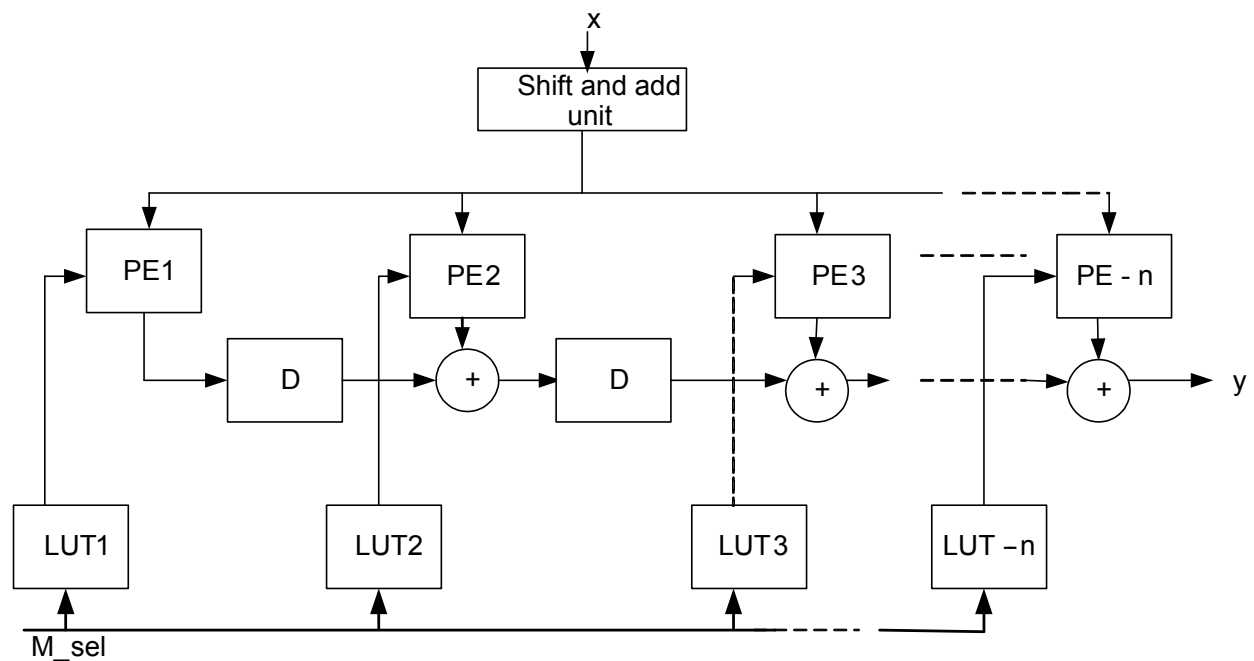


Figure 3. Proposed reconfigurable FIR filter architecture

### 3.2. Architecture of SMA

SMA employs 3-bit BCSE [15] algorithm while computing the coefficient multiplications. The algorithm reuses the frequent 3-bit BCSs to remove redundant computations in coefficient multiplications. Signed-magnitude format is used for both inputs and coefficients. The coefficients in LUT are split into 3-bit groups and given as selection lines to the multiplexers in PE. The architecture of PE is shown in Figure 4.

Coefficient precision is considered as 16 bits. Each 16-bit coefficient  $h[15:0]$  is stored as 17-bit coefficient  $h[16:0]$  in LUTs where MSB bit represents the sign bit and the remaining 16-bits represent the magnitude bits. Each coefficient except MSB is divided into five three bit groups and one 1-bit group. The first five three bit groups are given as selection lines to the five  $8 \times 1$  multiplexers (M1-M5) and the remaining bit

forms the selection line to the 2x1 multiplexer (M6). These multiplexers provide the appropriate BCSs from SAU depending on coefficient binary value.

SAU shown in Figure 5 computes the 3-bit BCSs of the coefficient multiplication with single input variable  $x$  and is given by  $[0\ 0\ 1] = C_1 = 2^{-2}x$ ,  $[0\ 1\ 0] = C_2 = 2^{-1}x$ ,  $[0\ 1\ 1] = C_3 = 2^{-1}x + 2^{-2}x$ ,  $[1\ 0\ 0] = C_4 = x$ ,  $[1\ 0\ 1] = C_5 = x + 2^{-2}x$ ,  $[1\ 1\ 0] = C_6 = x + 2^{-1}x$  and  $[1\ 1\ 1] = C_7 = x + 2^{-1}x + 2^{-2}x$ .

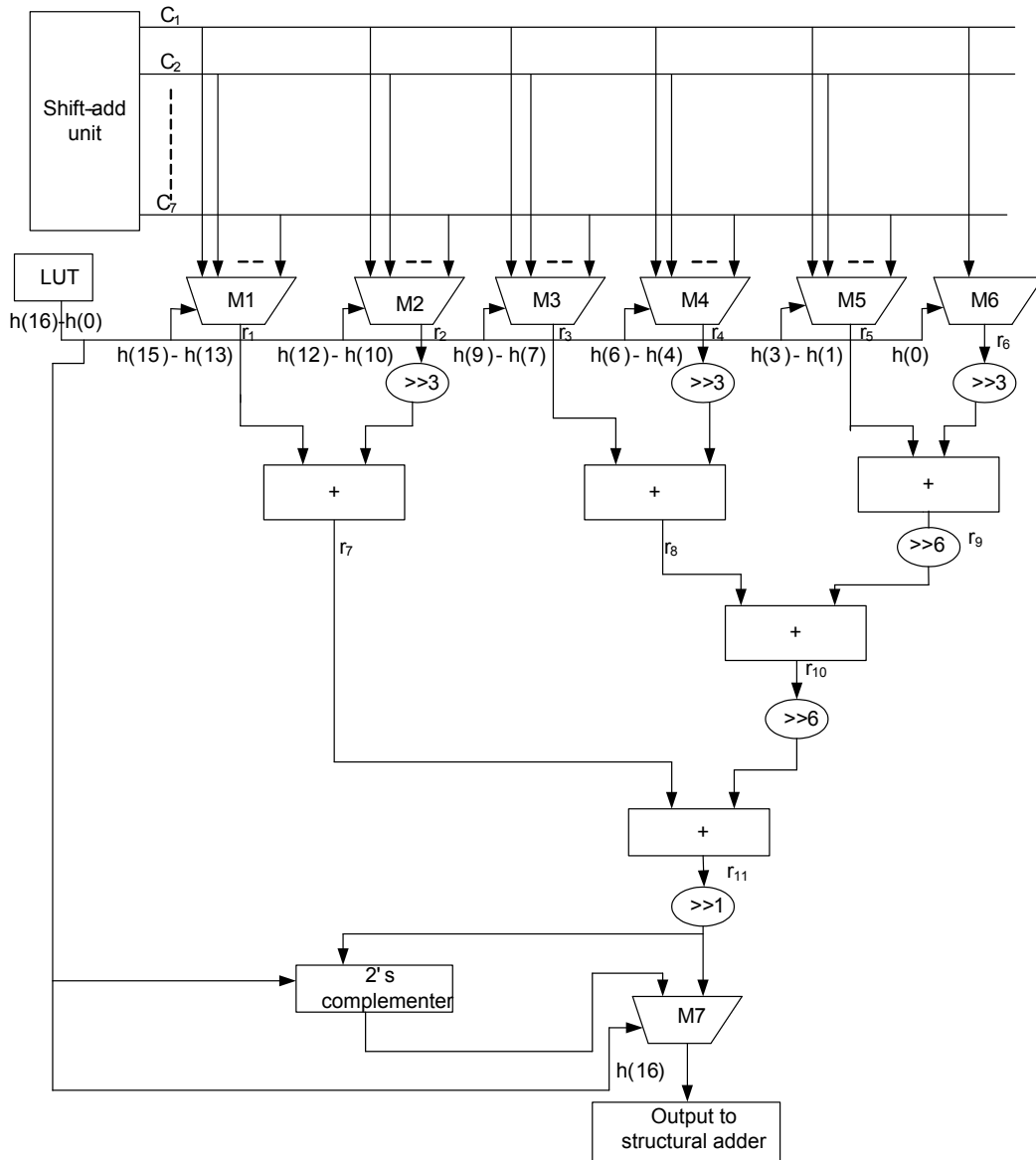


Figure 4. PE architecture of SMA [15]

Adders are represented with  $\Sigma$  operators and shifters are represented with shift operators ( $\gg$ ) along with their shifting values. Shifters provide the required shifts to the intermediate terms and adders add the intermediate terms to produce multiplication output. Two input multiplexer (M7) provides the true value or complemented value of output depending on the sign bit  $h$  [16].

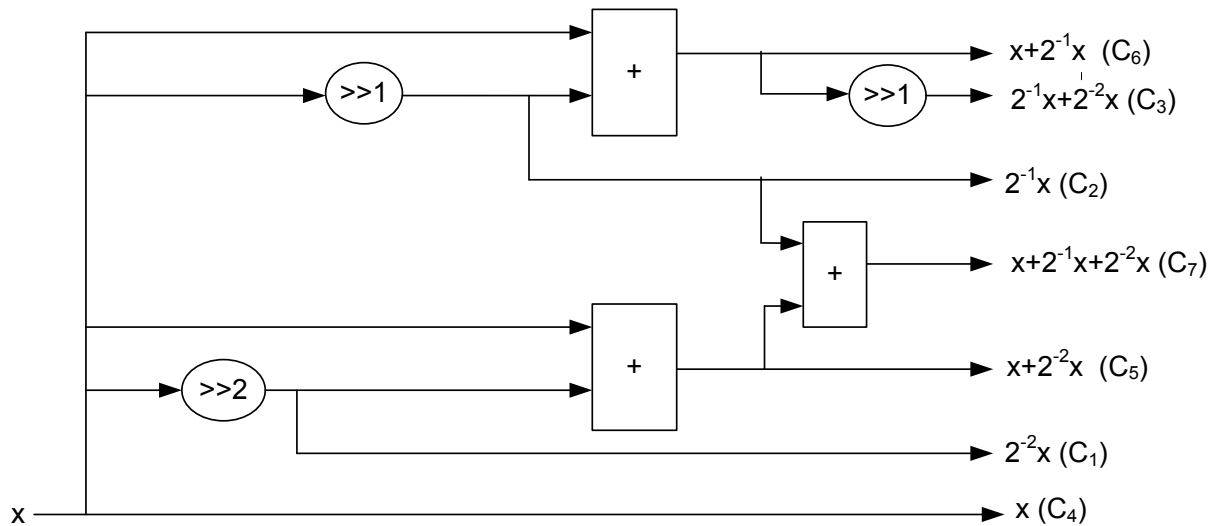
The following example describes the operation of the multiplication.

Let us consider a coefficient  $h = 0.11111111$ . The output  $y$  is given by

$$y = x * h = 2^{-1}x + 2^{-2}x + 2^{-3}x + 2^{-4}x + 2^{-5}x + 2^{-6}x + 2^{-7}x + 2^{-8}x \tag{5}$$

Considering 3-bit BCSs from most significant bit (MSB) of Eq. (5),  $y$  can be rewritten as

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



**Figure 5.** SAU architecture of SMA

The intermediate terms  $x + 2^{-1}x + 2^{-2}x$ ,  $x + 2^{-1}x$  in Equation (6) which are generated from the SAU, can be obtained by two  $8 \times 1$  multiplexers and one  $4 \times 1$  multiplexer. Shifters provide the required shifts of  $2^{-1}$ ,  $2^{-3}$  and  $2^{-6}$  to the intermediate terms and adders add the intermediate terms to produce the output  $y$ .

### 3.3. Architecture of SDA

SDA utilizes VHBCSE [17] algorithm which is a combination of horizontal BCSE (HBCSE) and vertical BCSE (VBCSE) to compute coefficient multiplications. VHBCSE [17] employs 2-bit VBCSE algorithm to eliminate redundant computations on adjacent coefficients followed by 4-bit and 8-bit HBCSE algorithms to eliminate redundant computations within the coefficients.

Architecture of PE is presented in Figure 6. The structure is shown for inputs and coefficients of 16-bit precision. PE gets the inputs from SAU shown in Figure 7. SAU generates the partial products of the shift-add based 16-bit coefficient multiplication while considering 2-bit BCSs 00, 01, 10 and 11. All the BCSs 00, 01, 10 and 11 can be realized with a single adder as BCS 11 alone needs an adder for its realization. Hence the partial products corresponding to 11 in coefficient multiplication are generated in SAU. These products will be added in further steps in PE to produce the final multiplication output.

Sign conversion block in Figure 8 changes the format of the coefficients in LUTs from sign-magnitude format to sign-decimal format (one's complement). It accommodates one's complement circuit to invert the coefficient  $M[15:0]$  except MSB  $M[16]$ . Two input multiplexer provides the multiplexed coefficient  $M_m[15:0]$  depending on the sign bit which is either the true value or one's complemented value of the coefficient. The multiplexed coefficient  $M_m[15:0]$  is split into four bit groups ( $M_m[15:12]$ ,  $M_m[11:8]$ ,  $M_m[7:4]$ ,  $M_m[3:0]$ ) and eight bit groups ( $M_m[15:8]$ ,  $M_m[7:0]$ ). These four and eight bit groups are given as inputs to the control logic generator. Control logic generator checks the equality of four, eight groups and provides 7 control signals as shown in Figure 9. These control signals control the addition operations at layer 2 and layer 3.



The coefficients are split into 2 bit groups and given as selection lines to the multiplexer. Since the coefficient length is 16, eight multiplexers are used at layer 1 as depicted in Figure 6 to produce the eight correct partial products  $r_0-r_7$  according to the coefficient binary value. At layer 2, as shown in Figure 10, four adders are used to add the eight partial products. Controlled additions are performed depending on the control signals  $CL_1-CL_6$  which are generated by 4-bit BCSE algorithms. In controlled additions, the addition operation is performed if the control signal is 0 otherwise the shifted version of other addition is taken as the result. The four intermediate sums  $r_8-r_{11}$  are added as shown in Figure 11 with the help of two adders in layer 3. Controlled addition is performed based on control signal  $CL_7$  which is generated by 8-bit BCSE algorithm. In layer 4, two outputs  $r_{12}$  and  $r_{13}$  from layer 3 is added and followed by one right shift operation to form the output  $y_1$ . Depending on the sign bit, the output is taken as  $y_1$  or two's complemented version of  $y_1$ .

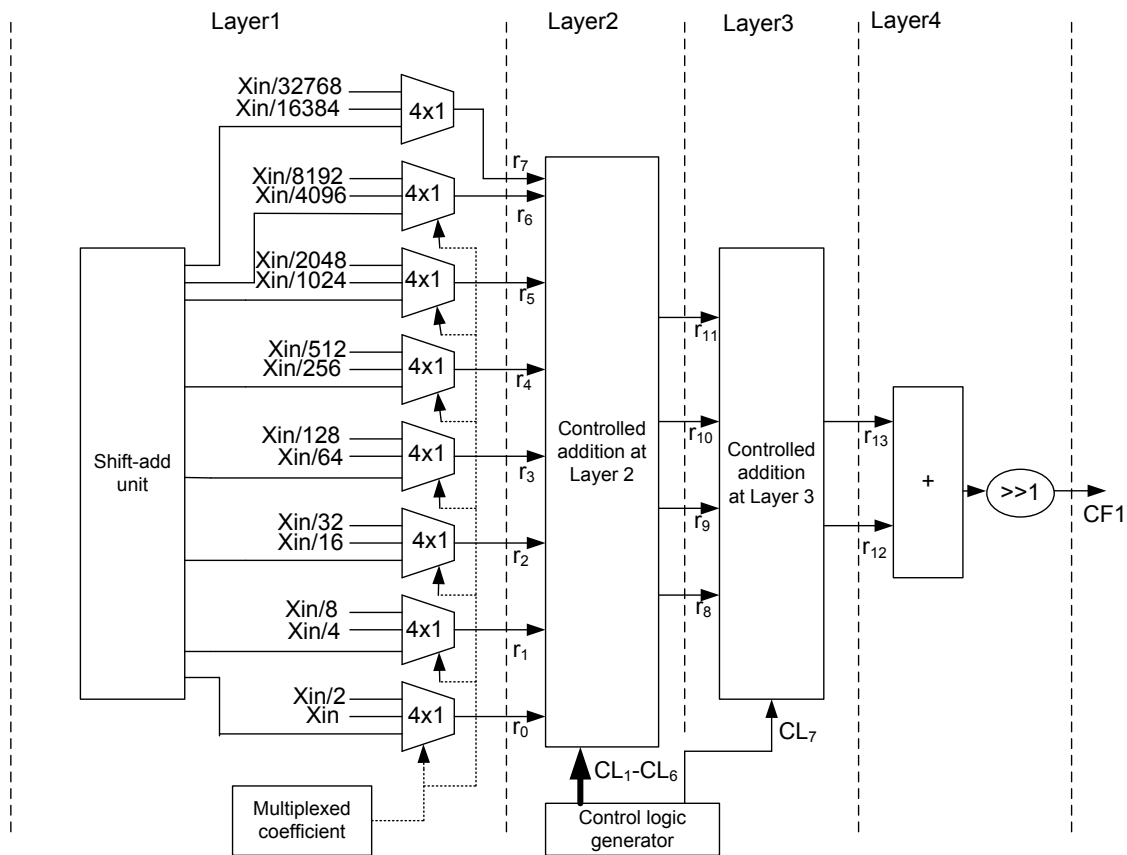


Figure 6. PE architecture of SDA [17]

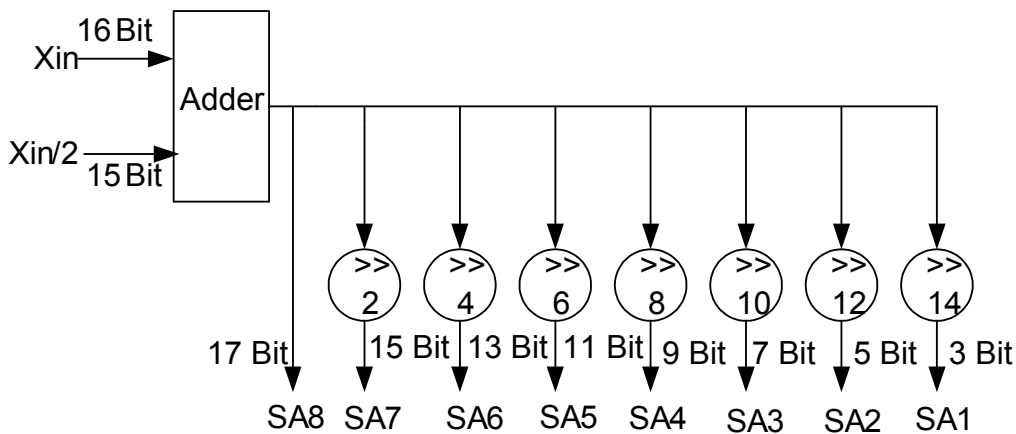


Figure 7. SAU architecture of SDA

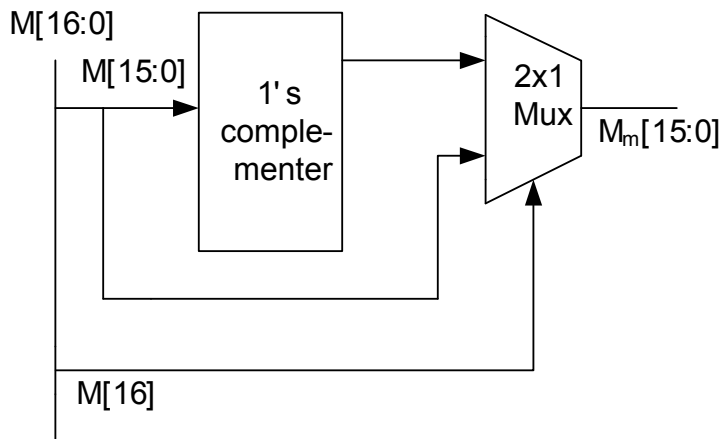


Figure 8. Sign Conversion Block

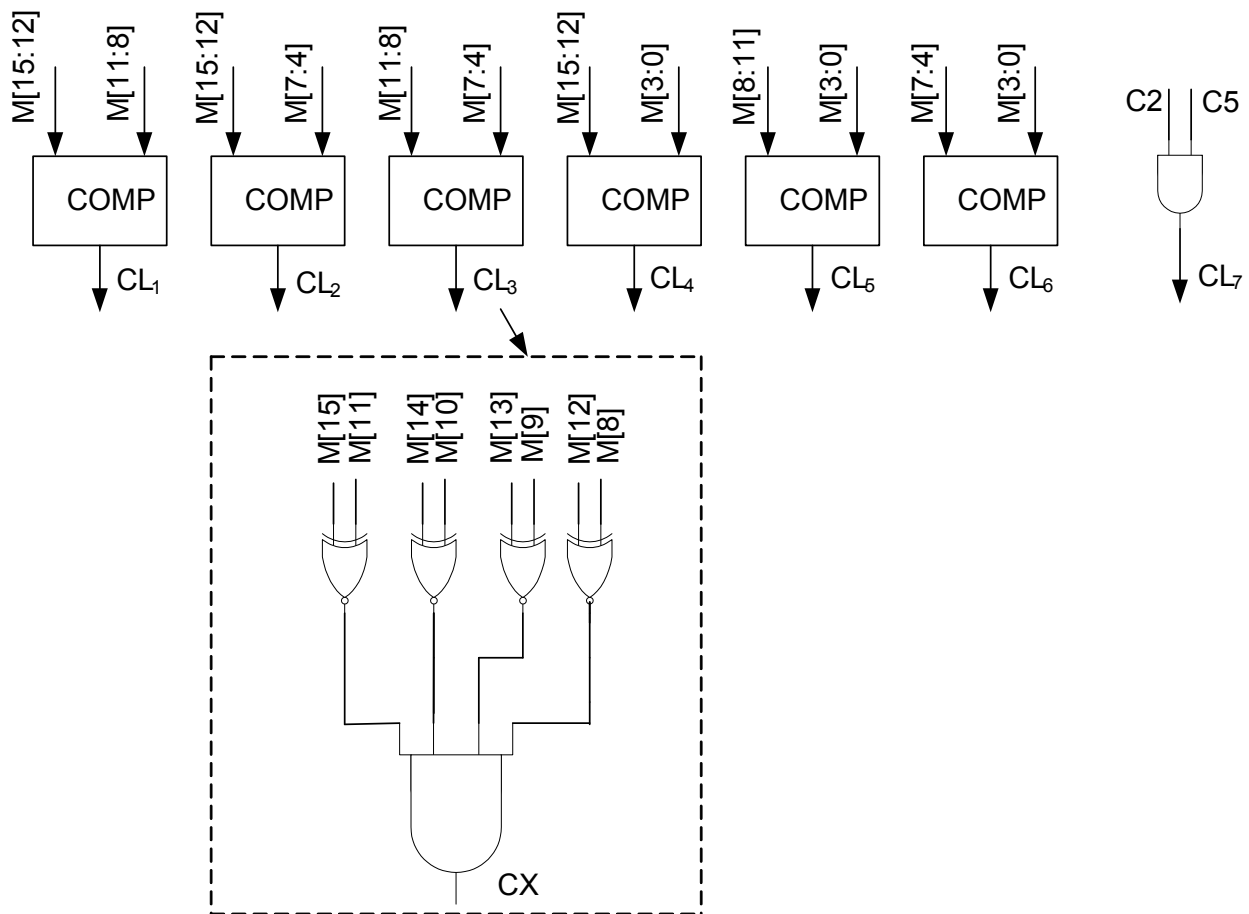


Figure 9. Control logic generator

#### 4. SIMULATION AND SYNTHESIS RESULTS

We have implemented low pass reconfigurable FIR filters of taps 19, 31 and 17 along with coefficient precisions 16-bit, 8-bit and 10-bit respectively with the following specifications.

- $\omega_p=0.1\pi, \omega_s=0.12\pi$
- $\omega_p=0.15\pi, \omega_s=0.2\pi$

- $\omega_p=0.2\pi, \omega_s=0.22\pi$
- $\omega_p=0.2\pi, \omega_s=0.3\pi$

Initially we designed the nominal filter by maximizing the stop band attenuation corresponding to the maximum pass band ripple and after getting suitable coefficients from initial design process, we used Greedy randomized heuristic [18] to determine the low complexity coefficients from the initial coefficients. The low complexity coefficients obtained from the second design phase are utilized to implement the filters.

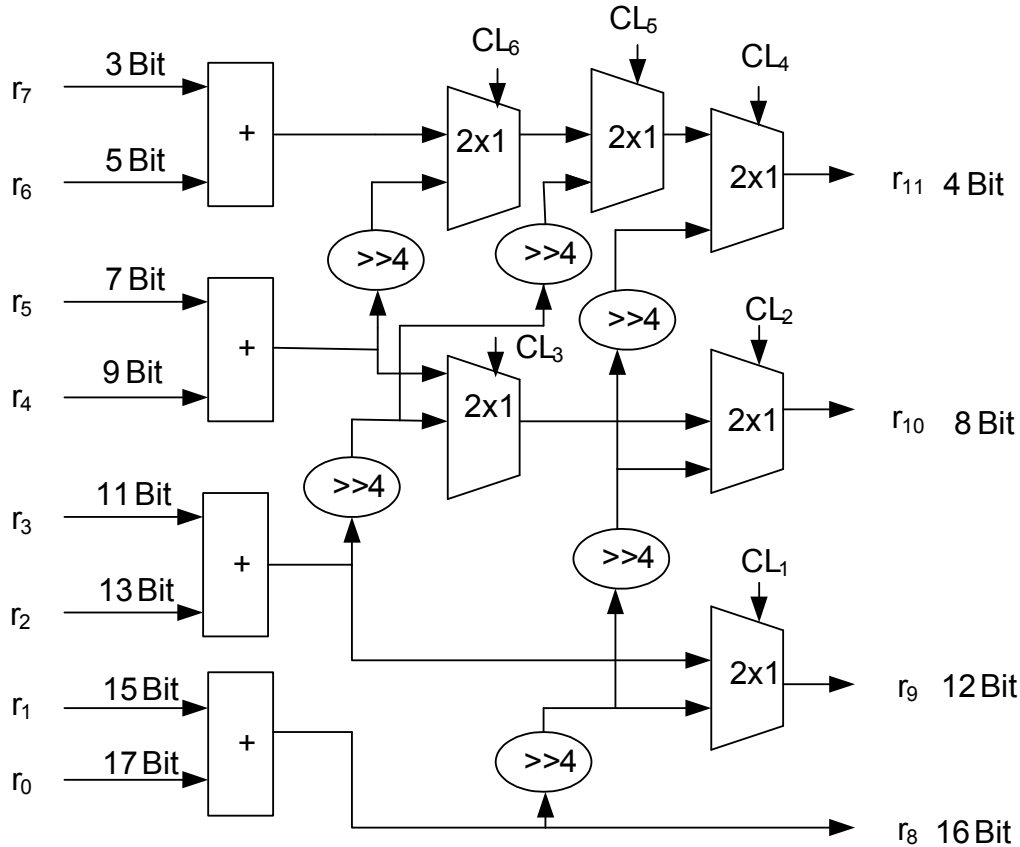


Figure 10. Addition at layer – 2

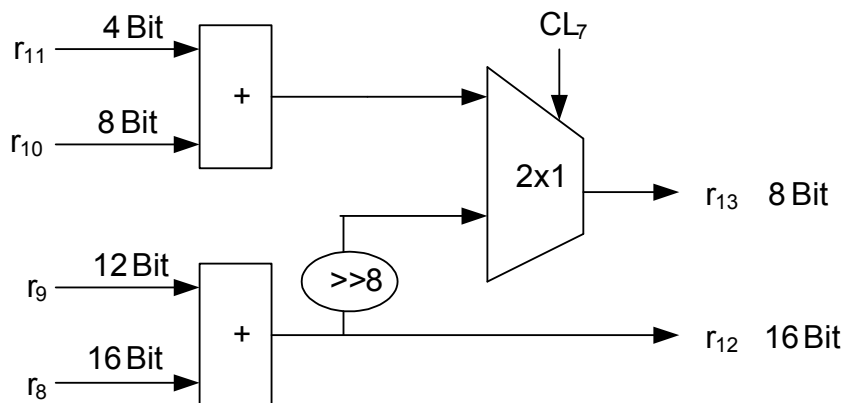


Figure 11. Addition at layer – 3

We simulated the proposed architectures SMA and SDA using Modelsim and synthesized with Synopsys 65nm CMOS technology. As the area, power and speed values of the synthesis results depend on filter coefficients, they vary for each specification. Hence we tabulated the average values. For fair comparison, we also synthesized Binary constant shifts method (BCSM) of [15] and VHBCSE of [17]. Then, all the proposed SMA and SDA structures are compared with [15] and [17] respectively.

Table 5 shows that the synthesis results of 19-tap, 16-bit coefficient length FIR filter. The results indicate that SMA offers 23.27% reduction in area and 33.92% reduction in dynamic power over [15]. Similarly the SMA offers reductions of 3.74% and 5.08% in area and power respectively while designing 31-tap, 8-bit coefficient length FIR filter over [15] and the results are shown in Table 6. The results in Table 7 shows that the proposed SMA consumes 14.34% less dynamic power over [15] with 10.19% reduction in area for 17-tap filter with coefficient precision of 10 bits.

**Table 5.** Synthesis results of proposed 19-tap SMA with 16-bit coefficient precision & 8-bit input precision

Parameters	BCSM [15]	Proposed SMA	%improvement over [15]
Area ( $\mu\text{m}^2$ )	25176.400	19316.400	23.27
Data required time (ns)	2.4	2.4	0.00
Dynamic power (mW)	5.637	3.725	33.92

**Table 6.** Synthesis results of proposed 31-tap SMA with 8-bit coefficient precision & 8-bit input precision

Parameters	BCSM [15]	Proposed SMA	%improvement over [15]
Area ( $\mu\text{m}^2$ )	21369.200	20570.800	3.74
Data required time (ns)	2.4	2.39	0.004
Dynamic power (mW)	3.999	3.796	5.08

**Table 7.** Synthesis results of proposed 17-tap SMA with 10-bit coefficient precision & 8-bit input precision

Parameters	BCSM [15]	Proposed SMA	%improvement over [15]
Area ( $\mu\text{m}^2$ )	14816.000	13306.800	10.19
Data required time (ns)	2.4	2.4	0.00
Dynamic power (mW)	3.082	2.640	14.34

Table 8 demonstrates the synthesis results of 19-tap filter with coefficient precision of 16 bits. It can be seen from Table 8 that the SDA has 4.21% reduction in area and 14.00% reduction in dynamic power over VHBCSE respectively over [17]. From Table 9, it can be seen that the SDA provides 23.39% subdual in required area overhead and 30.55% minimization in Dynamic power consumption respectively over [17] for a filter of 17-taps with 10-bit coefficient length.

**Table 8.** Synthesis results of proposed 19-tap SDA with 16-bit coefficient precision & 16-bit input precision

Parameters	VHBCSE [17]	Proposed SDA	%improvement over [17]
Area ( $\mu\text{m}^2$ )	31530.000	30203.200	4.21
Data required time (ns)	3.00	3.00	0.00
Dynamic power (mW)	6.827	5.871	14.00

**Table 9.** Synthesis results of proposed 17-tap SDA with 10-bit coefficient precision & 16-bit input precision

Parameters	VHBCSE [17]	Proposed SDA	%improvement over [17]
Area ( $\mu\text{m}^2$ )	26712.000	20464.800	23.39
Data required time (ns)	2.6	2.58	0.007
Dynamic power (mW)	6.166	4.282	30.55

From the results, it is evident that the proposed filters outperform the existing ones.

## 5. CONCLUSION

This paper presented the design and implementation of area and power efficient reconfigurable FIR filter. A methodology for designing filters with low binary complexity coefficients is introduced. Results of filter design with different specifications showed that the proposed method returns optimized filter designs with much reduced binary complexities. It is evident from the results that the efficiency of the algorithm increases with precision of the coefficients. We proposed two reconfigurable architectures namely SMA and SDA for efficient implementation of the filters. We synthesized the proposed filter on tsmc 65nm CMOS technology using Synopsys Design compiler. SMA offered up to 24% reduction in area and 33.91% reduction in dynamic power consumption respectively over those of BCSM. SDA provided a maximum minimization of 25.92% in area and 30.55% in dynamic power consumption as compared to VHBCSE. From the synthesis results, it can be concluded that the proposed reconfigurable filters are area and power efficient which made them suitable for SDR systems.

## CONFLICTS OF INTEREST

No conflict of interest was declared by the authors.

## REFERENCES

- [1] Hentschel, T., Fettweis, G., "Software radio receivers", CDMA techniques for third generation mobile systems, Springer, 257-283, (1999).
- [2] Hentschel, T., Henker, M., Fettweis, G., "The digital front-end of software radio terminals", IEEE Personal communications, 6(4): 40-46, (1999).
- [3] Dillinger, M., Madani, K., Alonistioti, N., Software defined radio: Architectures, systems and functions, John Wiley & Sons, (2005).
- [4] Singhal, S.K., Mohanty, B.K., "Efficient Parallel Architecture for Fixed-Coefficient and Variable-Coefficient FIR Filters Using Distributed Arithmetic", Journal of Circuits, Systems and Computers, 25(07): 1650073, (2016).
- [5] Hu, J., Huang, Z., Liu, C., Su, S., Zhou, J., "Design of Digital Channelizer Based on Source Number Estimation", Journal of Circuits, Systems and Computers, 25(02): 1650008, (2016).
- [6] Hewlitt, R.M., Swartzlantler, E.S., "Canonical signed digit representation for FIR digital filters", 2000 IEEE Workshop on Signal Processing Systems, SIPS 2000, Design and Implementation (Cat. No. 00TH8528), IEEE, 416-426, (2000).
- [7] Hashemian, R., "A new method for conversion of a 2's complement to canonic signed digit number system and its representation", Conference Record of the Thirtieth Asilomar Conference on Signals, Systems and Computers, IEEE, 904-907, (1996).
- [8] He, S., Torkelson, M., "FPGA implementation of FIR filters using pipelined bit-serial canonical signed digit multipliers", Proceedings of the IEEE Custom Integrated Circuits Conference-CICC'94, IEEE, 81-84, (1994).
- [9] Chen, K.H., Chiueh, T.D., "A low-power digit-based reconfigurable FIR filter", IEEE Transactions on Circuits and Systems II: Express Briefs, 53(8): 617-621, (2006).
- [10] Tang, Z., Zhang, J., Min, H., "A high-speed, programmable, CSD coefficient FIR filter", IEEE Transactions on Consumer Electronics, 48(4): 834-837, (2002).

- [11] Muhammad, K., Roy, K., “Reduced computational redundancy implementation of DSP algorithms using computation sharing vector scaling”, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 10(3): 292-300, (2002).
- [12] Park, J., Jeong, W., Mahmoodi-Meimand, H., Wang, Y., Choo, H., Roy, K., “Computation sharing programmable FIR filter for low-power and high-performance applications”, *IEEE Journal of solid-state Circuits*, 39(2): 348-357, (2004).
- [13] Voronenko, Y., Püschel, M., “Multiplierless multiple constant multiplication”, *ACM Transactions on Algorithms (TALG)*, 3(2): 11, (2007).
- [14] Gustafsson, O., Dempster, A.G., “On the use of multiple constant multiplication in polyphase FIR filters and filter banks”, *Proceedings of the 6th Nordic Signal Processing Symposium-NORSIG 2004*, 53-56, (2004).
- [15] Mahesh, R., Vinod, A.P., “New reconfigurable architectures for implementing FIR filters with low complexity”, *IEEE transactions on computer-aided design of integrated circuits and systems*, 29(2): 275-288, (2010).
- [16] Hatai, I., Chakrabarti, I., Banerjee, S., “An Efficient VLSI Architecture of a Reconfigurable Pulse-Shaping FIR Interpolation”, *IEEE Transactions on very large scale integration (VLSI) systems*, 23(6): 1150-1154, (2015).
- [17] Hatai, I., Chakrabarti, I., Banerjee, S., “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*, 62(4): 1071-1080, (2015).
- [18] Skaf, J., Boyd, S.P., “Filter design with low complexity coefficients”, *IEEE Transactions on Signal processing*, 56(7): 3162-3169, (2008).
- [19] Boyd, S., Vandenberghe, L., *Convex optimization*, Cambridge university press, (2004).