Scientific Journal Impact Factor (SJIF) : 1.711
International Journal of Modern Trends in Engineering and Research
ISSN (ONLINE) : 2349 - 9745 ISSN (PRINT) : 2393 - 8161

HIGH PERFORMANCE FIR FILTER USING BIT-PAIR RECODED AND FUSED ADD-MULTIPLIER

Aruna Devi.P[1]
[1]*VLSI Design, Sasurie College of Engineering*

Abstract- DSP plays an important role in development of modern consumer electronics. Large number of arithmetic operations is carried out in DSP applications. To optimize the design of the fused Add-Multiply (FAM) operator for increasing performance, the technique - Bit pair recoding is implemented in FAM as direct recoding of the sum of two numbers. An efficient and structured recoding technique and their three different schemes by incorporating them in FAM designs are explored. The comparison of the three different schemes: Sum-to-Bit-pair recoding FAM design is performed and their significant reductions in terms of critical delay, hardware complexity and power consumption of the FAM unit is obtained. The delay present in the existing FAM design is overcome by faithfully rounding and Truncation method and the proposed design is finally implemented in FIR filter and their parameters are analyzed through FPGA.

Index terms- Modified Booth Recoding, FIR Filter, faithful rounding, truncated multiplier, VLSI design.

I. INTRODUCTION

Real-valued digital FIR filters form the foundation for many digital signal processing applications. FIR filters are a unique category of digital filters. They are non-recursive type of filter where the present output depends on the present input sample and the previous samples. A general FIR filter of order *M* can be expressed as,

$$y[n] = \sum_{k=0}^{N-1} b_k . x[n-k] \qquad (1)$$

. The basic function needed to implement a FIR filter is the multiply-and-accumulate (MAC) operation. Multiply–accumulate operation is a common step that computes the product of two numbers and adds that product to an accumulator. The hardware part that performs the operation is known as a multiplier–accumulator (MAC, or MAC unit); the operation itself is also often called a MAC or a MAC operation. Numerous architectures have been projected to optimize the concert of the MAC operation in terms of critical path delay, area and power consumption; however which increases the MAC components [1], [8].

To overcome this, Fused Add-Multiply is optimized as the direct recoding of sum of two numbers in its Bit pair recoding form which reduces delay, area and power consumption [1], [9]. The three recoding schemes of sum-to-Bit pair recoding are investigated which increases the area, power and delay. The proposed technique decreases the area, power and delay by implementing the FIR filters based on the direct structure with faithfully rounded and truncated multipliers [7]. This paper is describes the Multiple Constant multiplication /accumulation (MCMA) module, which is realized by accumulating all the partial products (PPs) where unnecessary PP bits (PPBs) are removed without affecting the final exactitude of the outputs [5], [10]. The bit widths of all the filter coefficients are minimized using non uniform quantization with unequal word lengths in order to reduce the hardware cost while still agreeable the specification of the frequency response.

The rest of the paper is organized as follows: In Section II discusses the motivation and presents a technical background for Bit Pair Recoding, FAM units, Sum- to –Modified Booth. In Section III, describes the proposed *S-BP* recoding scheme with rounding and truncation is presented. In Section IV, compares the experimental evaluation clearly by identifying the advantages of the proposed schemes with respect to area complexity, critical delay and power dissipation. Section V, concludes our work.

II. OVERVIEW of BIT PAIR RECODING and FUSED ADD-MULTIPLIER

A. Bit Pair Recoding

Bit pair recoding can also be called as Modified Booth algorithm. It is possible to reduce the number of partial products by half, by using the technique of radix 4 Bit pair recoding [2], [4]. The basic proposals is that, instead of shifting and adding for every column of the multiplier term and multiplying by 1 or 0, choose only every second column and multiply by ±1, ±2, or 0, to obtain the same results. Radix-4 booth encoder performs the course of action of encoding the multiplicand based on multiplier bits. It will compare 3 bits with overlapping technique at a time. Grouping starts from the LSB, and the first block only uses two bits of the multiplier and assumes a zero for the third bit as shown by Fig 1.

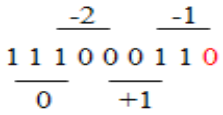

Figure 1. Bit Pairing as per Bit pair recoding.

Radix-4 Bit pair algorithm is given below:

(1) Extend the sign bit 1 position if necessary to ensure that n is even.
(2) Add a 0 to the right of the LSB of the multiplier.
(3) According to the value of each vector, each Partial Product will be 0, +y, –y, +2y or –2y.

The benefit of this method is the halving of the number of partial products. This is significant in circuit design as it relates to the propagation delay in the running of the circuit, the complexity and power consumption of its realization. The efficient operation of Radix-4 Bit pair encoder is shown in the Table 1. It consists of eight different types of states and during these states, the outcomes, which are multiplication of multiplicand with 0,-1 and -2 consecutively.

Table 1. BIT PAIR RECODING TABLE for RADIX-4.

Binary			y_j^{MB}	Bit pair Encoding			Input Carry $c_{in,j}$
y_{2k+1}	y_{2k}	y_{2k-1}		Sign = s_j	x1 = one_j	x2 = two_j	
0	0	0	0	0	0	0	0
0	0	1	+1	0	1	0	0
0	1	0	+1	0	1	0	0
0	1	1	+2	0	0	1	0
1	0	0	-2	1	0	1	1
1	0	1	-1	1	1	0	1
1	1	0	-1	1	1	0	1
1	1	1	0	1	0	0	0

Let us consider the multiplication of 2's complement numbers X and Y with each number consisting of n = 2k bits. The multiplicand Y can be represented in Bit pair form as:

$$Y = (y_{n-1}y_{n-2} \ldots\ldots y_1 y_0)_2 = -y_{2k-1}2^{2k-1} + \sum_{i=0}^{2k-2} y_i 2^i$$

$$= [y_{k-1}^{BP} y_{k-2}^{BP} \ldots\ldots y_1^{BP} y_0^{BP}]_{BP} = \sum_{J=0}^{K-1} y_j^{MB}.2^{2J} \qquad (2)$$

$$y_j^{BP} = -2y_{2j+1} + y_{2j} + y_{2j-1} \qquad (3)$$

Digits, $y_j^{BP} \in \{-2, -1, 0, +1, +2\}$, 0≤j≤k-1 correspond to the three consecutive bits y_{2j+1}, y_{2j} and y_{2j-1} with one bit overlapped and taking into consideration that $y_{-1} = 0$. Each digit is represented by three bits named *s*, *one* and *two*. If the digit is negative, then the sign bit will be equal to 1 or else the sign bit will be equal to 0. Signal *one* shows if the absolute value of a digit is equal to 1 (*one* = 1) or not (*one* = 0). Signal *two* shows if the absolute value of a digit is equal to 2 (*two = 1*) or not (*two = 0*). Using these three bits we calculate the Bit pair digits y_j^{BP} by the following relation:

$$y_j^{BP} = (-1)^{s_j}.[one_j + 2.two_j] \qquad (4)$$

Equation (5), (6) & (7) shows the Boolean equations on which the implementation of the Bit pair encoding signals which is shown in the Figure 2.

$$one_j = y_{2j-1} \oplus y_{2j} \qquad (5)$$

$$two_j = (y_{2j-1} \oplus y_{2j})(\overline{one_j}) \qquad (6)$$

$$s_j = y_{2j+1} \qquad (7)$$

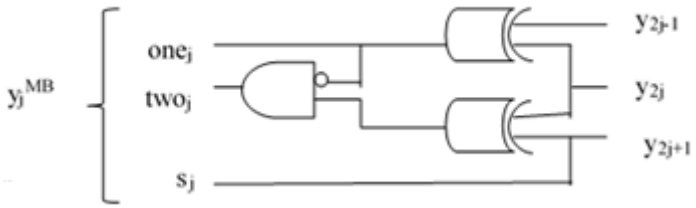

Figure 2. Bit pair Encoding Signals

B. FUSED ADD-MULTIPLY

An optimized propose of the AM operator is based on the fusion of the adder and the Bit pair encoding unit into a single block by direct recoding of the sum Y=A+B to its bit pair recoding representation. The fused Add-Multiply (FAM) component contains only one adder at the end. As a result, momentous area savings are observed and the critical path delay of the recoding process is reduced and decoupled from the bit-width of its inputs. A new technique for direct recoding of two numbers in the Bit pair image of their sum is shown in the Figure 3.

In the FAM design, the multiplier is a parallel one based on the Bit pair algorithm. Let us consider the product X.Y. The term multiplicand, Y= $Y_{n-1}Y_{n-2}\ldots\ldots Y_2Y_1Y_0$ is encoded based on the Bit pair

algorithm and multiplied with X= $X_{n-1}X_{n-2}\ldots\ldots X_2X_1X_0$. Both and consist of n=2k bits and are in 2's complement form.

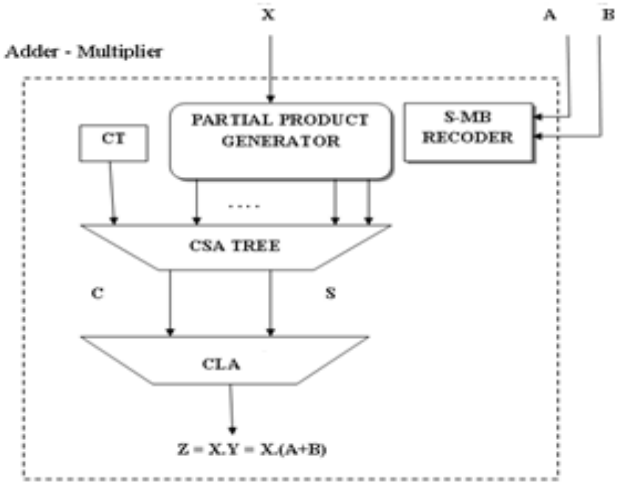

Figure 3. Fused Design of the AM Operator

Equation (8) describes the invention of the k partial products:

$$PP_j = X.y_j^{BP} = p_{j,n}.2^n + \sum_{i=0}^{n-1} p_{j,i}.2^i \qquad (8)$$

The generation of the i-th bit $p_{j,i}$ of the partial product PP_j is based on the next logical expression.

$$p_{j,i} = ((x_i \oplus s_j) \wedge one_j \vee \left((x_{i-1} \oplus s_i) \wedge two_j\right) \qquad (9)$$

For the calculation of the LSB and MSB of the partial product, consider $x_{-1} = 0$ and $x_n = x_{n-1}$ respectively. If n= 2k+1, then the number of the resulting partial products is [n/2] +1 = k+1 and the most significant Bit pair digit is formed based on sign extension of the initial 2's complement number. After the partial products are generated, they are added, accurately weighted, through a Wallace Carry-Save Adder (CSA) tree along with the Correction Term (CT) which is given by the following equations:

$$Z = X.Y = CT + \sum_{j=0}^{k-1} PP_j 2^{2j} \qquad (10)$$

CT = CT (low) + CT (high)

$$CT = \sum_{j=0}^{k-1} c_{in,j}\, 2^{2j} + 2^n(1 + \sum_{j=0}^{k-1} 2^{2j+1}) \qquad (11)$$

where, $c_{in,j} = one_j \vee two_j \wedge s_j$. At last, the carry-save output of the Wallace CSA tree is leaded to a fast Carry Look Ahead (CLA) adder to form the final result Z= X.Y.

III. SUM TO BIT PAIR RECODING TECHNIQUE (S-BP) IN FIR FILTER

A. SUM TO BIT PAIR RECODING:

In *S-BP* recoding technique, the sum of two successive bits of the inputs (a_{2j}, a_{2j+1}), (b_{2j}, b_{2j+1}) is recoded into one BP digit y_j^{BP}. From (3), three bits are included in forming a BP digit. The most significant of them is negatively weighted while the two least significant of them have positive

weight. In order to transform the two abovementioned pairs of bits in BP form signed-bit arithmetic is used. For this function, a signed Half Adders (HA) and Full Adders (FA) are developed by considering their inputs and outputs to be signed.
Two types of signed HAs which are referred as HA* and HA** are used. Considering that p, q are the binary inputs and c, s are the outputs carry and sum respectively, HA* which implements the relation 2c –s = p+q where the bit sum is considered negative and, as a result, the output takes one of the values {0, +1, +2}. The dual implementation of HA* is implemented by inverting the signs of all inputs and outputs and, as a result, and it changes the output values to {-2, -1, 0}. The Boolean equation for both the Half adders is given by $c = p \vee q$ and $s = p \oplus q$. In the case of HA**, the input p is negatively signed and q is positively signed and HA** implements the relation 2-c-s = -p+q and the output values become {-1, 0, +1}. The Boolean equation for the HA** operation is given by $c = \bar{p} \wedge q$ and $s = p \oplus q$.

In addition, two types of signed FAs which are referred as FA* and FA** are designed. Assuming that p, q and c_i are the binary inputs and c_0, s are the output carry and sum respectively, FA* implements the relation $2c_0 = p\text{-}q+c_i$ where the bits s and q are considered negatively signed and the output values of FA* are {-1, 0, +1, +2}. The Boolean equation for the FA* is given $c_0 = (p \vee \bar{q} \wedge c_i) \vee (p \wedge \bar{q})$ and $s = p \oplus q \oplus c_i$. In the case of FA**, the two inputs p, q are negatively signed and FA** implements the relation $-2c_0+s = -p-q+c_i$. And the output values become {-2, -1, 0, +1}. The Boolean equation for the FA** operation is given by $c_0 = (p \vee q \wedge \bar{c_i}) \vee (p \wedge q)$ and $s = p \oplus q \oplus c_i$.

The three schemes of the S-BP Recoding technique are designed using usual and signed HAs and FAs. All the three schemes can be applied in either signed or unsigned numbers which consist of odd or even number of bits [6], [9]. In all schemes, assume that both inputs and are in 2's complement form and consist of 2k bits in case of even or 2k+1 bits in case of odd bit-width. These techniques are recoded into the sum of two numbers into BP form.

1) S-BP1 Recoding Scheme

The first scheme of the existing recoding technique is referred as *S-BP*1 and is described in detail in Figure 4 for even and Figure 5 for odd bit-width of input numbers.

$$Y = A + B = y_k 2^k + \sum_{j=0}^{k-1} y_j^{BP} 2^{2j}$$

where, $y_j^{BP} = -2s_{2j+1} + s_{2j} + c_{2j}$ (12)

Considering the initial values $b_{-1}=0$ and $c_0 = 0$, the BP digits $y_j^{BP}, 0 \le j \le k-1$, of (12) is based on s_{2j+1} and s_{2j} are extracted. A usual FA with inputs a_{2j}, b_{2j} and b_{2j-1} produces the carry $c_{2j+1} = (a_{2j} \wedge b_{2j}) \vee (b_{2j-1} \wedge (a_{2j} \vee b_{2j}))$ and the sums $s_{2j} = a_{2j} \oplus b_{2j} \oplus b_{2j+1}$. As the bit s_{2j+1} needs to be negatively signed, then use a FA* with inputs a_{2j+1}, b_{2j+1} (-) and c_{2j+1}, which produces the carry, $c_{2j+2} = (a_{2j+1} \wedge \overline{b_{2j+1}}) \vee (c_{2j+1} \wedge (a_{2j+1} \vee \overline{b_{2j+1}})$ and the sum, $s_{2j+1} = a_{2j+1} \oplus b_{2j+1} \oplus c_{2j+1}$ in negative sign. Note that, b_{2j+1} is driven to the FA* as negatively signed.

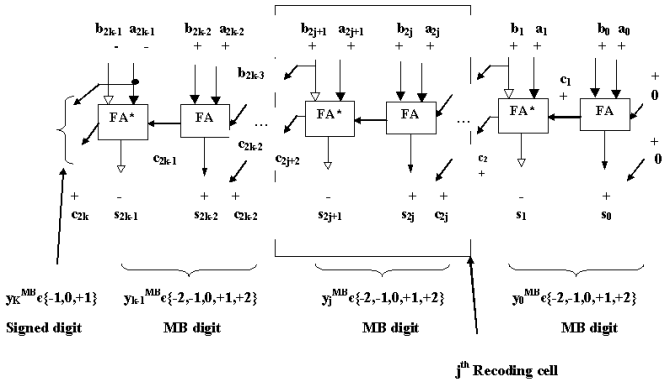

Figure 4. S-BP1 recoding scheme for even number of bits.

In the first case, the bit-width is even; the MSD $y_{k,even}^{SD}$ is a signed digit and is given by $y_{k,even}^{SD} = a_{2k-1} + c_{2k}$.

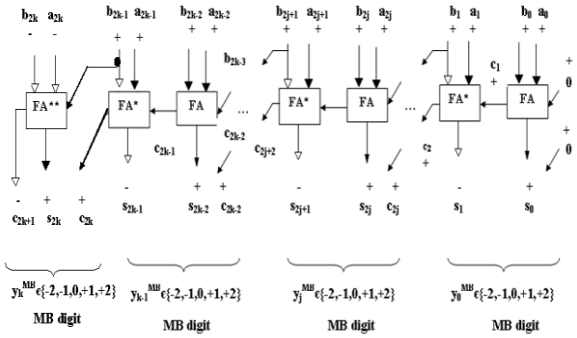

Figure 5. S-BP1 recoding scheme for odd number of bits.

In the second case, the bit-width is odd; the MSD $y_{k,odd}^{SD}$ is a MB digit that is formed based on c_{2k+1}, s_{2k} and c_{2k}. The carry $c_{2k+1}(-)$ and the sum s_{2k} are produced by a FA** with inputs $a_{2k}(-)$, $b_{2k}(-)$ and b_{2k-1}. The critical path delay is given by the equation: $T_{S-BP1} = T_{FA,Carry} + T_{FA*,sum}$, where $T_{FA,Carry}$ is the delay of shaping the output carry of a usual FA and $T_{FA,Carry}$ is the delay of forming the sum of a signed FA*.

2) *S-BP2 Recoding Scheme*

The second approach of the existing recoding technique, *S-BP*2, is described in Figure 6 for even and Figure 7 for odd bit-width of input numbers. Considering the initial values $c_{0,1}$=0 and $c_{0,2}$=0, the BP digits $y_j^{BP}, 0 \le j \le k-1$, of (12) is based on s_{2j+1}, s_{2j} and $c_{2j,2}$. A usual FA with the inputs a_{2j}, b_{2j} and $c_{2j,1}$ is used to produce the carry c_{2j+1} and the sum s_{2j}. The bit $c_{2j,1}$ is the output carry of a usual HA and has the bits a_{2j-1}, b_{2j-1} as inputs. The bit s_{2j+1} is the output sum of a HA* in which we drive c_{2j+1} and the sum produced by a usual HA with the bits a_{2j+1}, b_{2j+1} as inputs. The HA* is used in order to produce the (-) sum $s_{2j+1} = a_{2j+1} \oplus b_{2j+1} \oplus c_{2j+1}$ and its carry outputs is given by $c_{2j+2,2} = c_{2j+1} \vee (a_{2j+1} \oplus b_{2j+1})$.

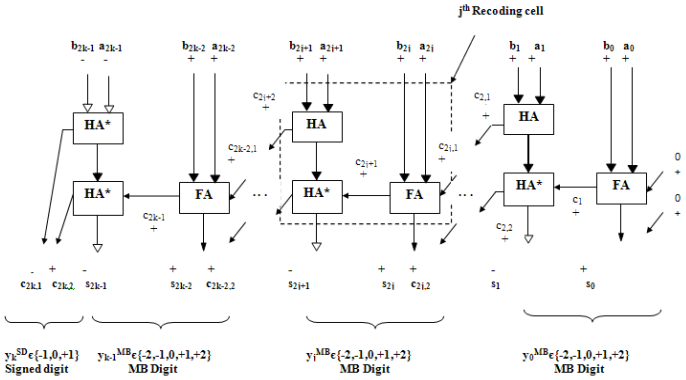

Figure 6. S-BP2 recoding scheme for even number of bits.

In the first case, the bit-width is even; the MSD $y_{k,even}^{SD}$ is a signed digit and is given by $y_{k,even}^{SD} = -c_{2k,1} + c_{2k,2}$.

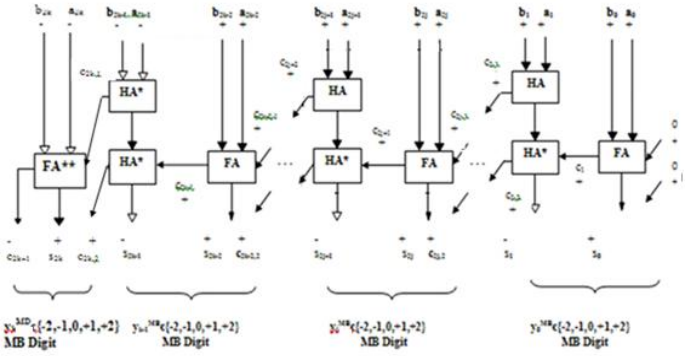

Figure 7 S-BP2 recoding scheme for odd number of bits.

In the second case, the bit-width is odd; the MSD $y_{k,odd}^{SD}$ is a MB digit that is formed based on c_{2k+1}, s_{2k} and $c_{2k,2}$. The carry $c_{2k+1}(-)$ and the sum s_{2k} are produced by a FA** with inputs $a_{2k}(-)$, $b_{2k}(-)$ and $c_{2k,1}$. The critical path delay is given by the equation$T_{S-BP2} = T_{HA,Carry} + T_{FA,Carry} + T_{HA*,sum}$, where $T_{HA,Carry}$ and $T_{FA,Carry}$ are the delays of shaping the output carry of a usual HA & FA respectively and $T_{FA,Carry}$ is the delay of forming the sum of a signed HA*.

3) S-BP3 Recoding Scheme

The third approach of the existing recoding technique, *S-BP*3, is described in Figure 8 for even and Figure 9 for odd bit-width of input numbers. Considering the initial values $c_{0,1}$=0 and $c_{0,2}$=0, the BP digits $y_j^{BP}, 0 \le j \le k-1$, of (12) is based on s_{2j+1}, s_{2j} and $c_{2j,2}$. A usual FA with the inputs a_{2j}, b_{2j} and $c_{2j,1}$ is used to produce the carry c_{2j+1} and the sum s_{2j}. The bit $c_{2j,1}$ is the output carry of a HA* and has the bits a_{2j-1}, b_{2j-1} as inputs. The negative signed bit s_{2j+1} is the output sum of a HA** is drive c_{2j+1} and the sum produced by a HA* with the bits a_{2j+1}, b_{2j+1} as inputs. The HA** is used in order to produce the sum $s_{2j+1} = a_{2j+1} \oplus b_{2j+1} \oplus c_{2j+1}$ and its carry outputs is given by$c_{2j+2,2} = c_{2j+1} \wedge \overline{(a_{2j+1} \oplus b_{2j+1})}$.

In case of even number of bits a_{2k-1} and b_{2k-1} are negatively weighted and the dual implementation of the HA* is used. As a result, the output sum of the HA* becomes positively weighted and the HA** that follows has to be replaced with a HA*.

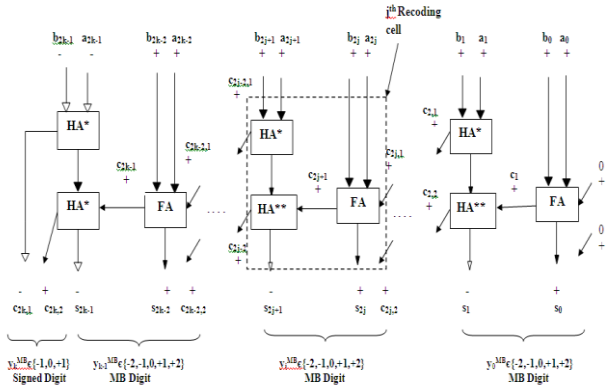

Figure 8 S-BP3 recoding scheme for even number of bits.

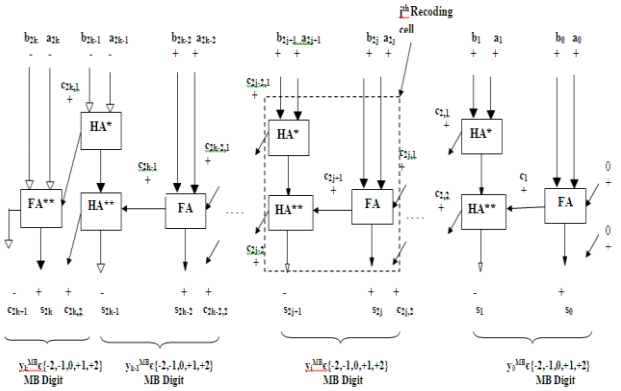

Figure 9 S-BP3 recoding scheme for odd number of bits.

The critical path delay of *S-BP*3 recoding is given by$T_{S-BP3} = T_{HA*,Carry} + T_{FA,Carry} + T_{HA**,sum}$, where $T_{HA*,Carry}$ and $T_{FA,Carry}$ are the delays of shaping the output carry of a signed HA* and FA respectively and $T_{HA**,sum}$ is the delay of forming the sum of a signed HA**.

4) Unsigned Input Numbers

In case of unsigned bits, their most significant bits are positively signed. The modifications that made in all *S-BP* schemes for both cases of even (the two most significant digits change) and odd (only the most significant digit change) bit-width of A and B, concerning the signs of the most significant bits of A and B. The basic recoding block in all schemes remains unchanged.

B. FIR FILTER WITH ROUNDING AND TRUNCATION USING FAM TECHNIQUE

Filters are widely employed in signal processing and communication systems in applications such as noise reduction, radar, channel equalization, biomedical signal processing, video processing, audio processing and analysis of profitable and monetary data. A non-recursive filter has no response and its input-output relation. The output $y(m)$ of a non-recursive filter is a function only of the input signal $x(m)$. The retort of such a filter to an impulse consists of a finite sequence of $M+1$ sample, where M is the filter order. Hence, the filter is well-known as a *Finite-Duration Impulse Response* (FIR) filters. Other names for a non-recursive filter include all-zero filter, feed-forward filter or moving average (MA) filter a term usually used in statistical signal processing literature. There are different structures for realization of a digital filter. These structures offer various tradeoffs between complexity, cost of implementation, computational efficiency and stability. According to Eqn. (1), FIR filter of order N is characterized by N+ 1 coefficient and, in general, requires N+1 multipliers and N two input adders. The multipliers coefficients are precisely the coefficients of the transfer

function are called direct form structures. A direct form recognition of an FIR filter can be readily developed from the convolution sum description as indicated below for N=4.

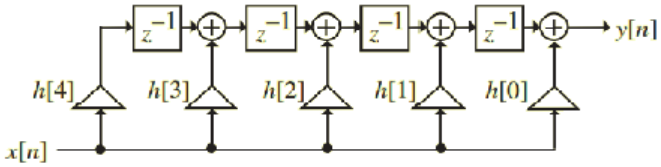

Figure 10. Direct Form FIR Filter

Both direct form structures are canonic with high opinion to delays. Multiply–accumulate operation is a common step that computes the product of two numbers and adds that product to an accumulator. For this purpose, Fused Add-Multiply is optimized as the direct recoding of sum of two numbers in its Bit pair recoding form which reduces delay, area and power consumption. An important design issue of FIR filter implementation is the optimization of the bit widths for filter coefficients, which has direct contact on the area cost of arithmetic units and registers. In addition, given that the bit widths after multiplications grow, many DSP applications do not need full-precision outputs. As an alternative, it is desirable to generate faithfully rounded outputs where the total error introduced in quantization and rounding is no more than one unit of the last place defined as the weighting of the least significant bit (LSB) of the outputs.

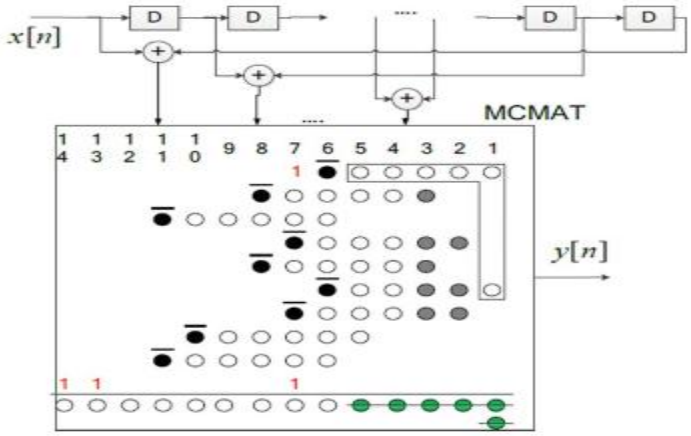

Figure 11 Rounding and Truncation

Implementations of Low-cost FIR filters are based on the direct structure in Figure 10 with faithfully rounded truncated multipliers are presented. The MCMA module is realized by accumulating all the partial products (PPs) where unnecessary PP bits (PPBs) are removed without affecting the final precision of the outputs. Instead of accumulating individual multiplication for every product, it is more proficient to collect all the PPs into a single PPB matrix with carry-save addition to reduce the height of the matrix to two, followed by a last carry propagation adder. For the addition of two bits, S-BP technique is presented to form BP representation. Then the BP form is multiplied with the input. The difference of individual multiplications and combined multiplication for A × B + C × D. The bit widths of all the filter coefficients are minimized using non uniform quantization with unequal word lengths in order to reduce the hardware cost while still satisfying the specification of the frequency response. In order to avoid the sign extension bits, we complement the sign bit of each PP row and add some bias constant using the property $\bar{s}$ =1- s, where s is the sign bit of a PP row. All the bias constants are collected into the very last row in the PPB matrix. The removal of unnecessary PPBs is composed of three processes: deletion, truncation, and rounding.

Figure 11 shows the illustrative architecture of MCMA with truncation (MCMAT) that removes unnecessary PPBs [3], [7]. The white circles in the L-shape block represent the undeletable PPBs.

The gray circles represent deletion of the PPBs. After PP compression, the rounding of the resultant bits is denoted by crossed circles. The last row of the PPB matrix represents all the offset and bias constants required including the sign bit modifications.

IV. PERFORMANCE ANALYSIS

The 8 bit and 4 tab FIR filter implementations has been done based on usual add-multipliers (using three alternative techniques) and also the design was proposed based on Rounded Truncated multiplier (using three alternative techniques). Comparing the prior techniques, the proposed technique improves the area, power and time compared to prior techniques.

Table 2. PERFORMANCE ANALYSIS

Techniques using	EXISTING			PROPOSED		
	Time (ns)	Area	Power (mw)	Time (ns)	Area	Power (mw)
SMB1	20.128	6028	347	20.446	4145	248
SMB2	18.178	4576	288	15.952	3206	240
SMB3	17.388	4549	264	16.579	3131	234

V. CONCULSION

This paper focuses on low-cost FIR filter designs with Fused Add-Multiply technique. We propose a three S-BP recoding technique used in FIR filter with faithfully rounded and truncation and compare them to the existing S-BP technique incorporated with the FAM design, which leads to low area, cost and power consumption.

REFERENCES

[1] Amaricai A., Vladutiu M. and Boncalo.O. (2010), "*Design issues and implementations for floating-point divide-add fused," IEEE Trans. Circuits Syst. II*–Exp. Briefs, vol. 57, no. 4, pp. 295–299.

[2] Daumas M. and Matula D. W. (2000), *"A Booth multiplier accepting both a redundant or a non redundant input with no additional delay,"* in Proc. *IEEE Int. Conf. on Application-Specific Syst., Architectures, and Processors*, pp. 205–214.

[3] Ko H.-J. and Hsiao S.-F. (2011), "*Design and application of faithfully rounded and truncated multipliers with combined deletion, reduction, truncation, and rounding," IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 58, no. 5, pp. 304–308.

[4] Lyu C. N. and Matula D. W. (1995), *"Redundant binary Booth recoding,"* in Proc. 12th Symp. Comput. Arithmetic, pp. 50–57.

[5] Nikolaidis S., Karaolis E. and Kyriakis-Bitzaros E. D. (2000), *"Estimation of signal transition activity in FIR filters implemented by a MAC architecture," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst*., vol.19, no. 1, pp. 164–169.

[6] Seo Y.-H. and Kim D.-W. (2010), *"A new VLSI architecture of parallel multiplier–accumulator based on Radix-2 modified Booth algorithm". IEEE Trans. Very Large Scale Integr. (VLSI) Syst*., vol. 18, no. 2, pp.201–208.

[7] Shen-Fu Hsiao, Jun-Hong Zhang Jian, and Ming-Chih Chen (2013) *"Low-Cost FIR Filter Designs Based on Faithfully Rounded Truncated Multiple Constant Multiplication/Accumulation" IEEE Trans. on* Circuits Syst.*—II:* express briefs, vol. 60, no. 5.

[8] Swartzlander E. E. and Saleh H. H. M. (2012*), "FFT implementation with fused floating-point operations," IEEE Trans. Comput*., vol. 61, no. 2, pp. 284–288.

[9] Tsoumanis K., Xydis S., Efstathiou C., Moschopoulos N. and Pekmestzi K. (2014), *"An Optimized Modified Booth Recoder for Efficient Design of the Add-Multiply Operator, "IEEE Trans. Circuits Syst. I*–Regular paper, vol. 61, no. 4, pp. 1133-1143.

[10] F. Xu, C. H. Chang, and C. C. Jong, "Design of low-complexity FIR filters based on signed-powers-of-two coefficients with reusable common subexpressions," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, no. 10, pp. 1898–1907, Oct. 2007.