Research Journal of Applied Sciences, Engineering and Technology 10(8): 890-894, 2015
DOI:10.19026/rjaset.10.2444
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: January 13, 2015 Accepted: March 25, 2015 Published: July 20, 2015

**Research Article**

# Design and Implementation of Reconfigurable FIR Filter using Common Sub-expression Elimination Method

[1]R. Dhayabarani and [2]R.S.D. Wahidabanu
[1]Department of Electronics and Communication Engineering, V.S.B. Engineering College, Karur, India
[2]Department of Electronics and Communication Engineering, Government College of Engineering, Salem, India

**Abstract:** The Design of Software Radio (SDR) will require a sharp filter of different bandwidth to fine tune the desired channel. This requires a large number of filter coefficients and different computational resources. In this study the novel coefficient optimization algorithm is proposed to reduce the area and power of finite Impulse Response Filter (FIR) design. In general filter coefficients will be encoded into the fewest possible nonzero bits using canonic signed digit expression. The proposed algorithm will share the Common Subexpressions (CS) and reduce the usage of replicating operations by involving CSD coefficients. The effectiveness of the proposed Binary Signed Sub coefficient (BSS) algorithm is identified and confirmed by filters using Collision Detection Multiple Access (CDMA), 121-tap high pass band, 105 and 325-tap low pass bands as benchmarks. The proposed algorithm will increase the number of common sub expression and realize the implementation procedure with low complexity. The experimental result shows our proposed algorithm will be a better one than the earlier with the smaller combinational logic and better throughput/area.

**Keywords:** Binary Signed Subcoefficient (BSS), reconfigurable filter, Software Radio (SDR), subcoefficient

## INTRODUCTION

Finite Impulse Response (FIR) filters are widely used in and Digital Signal Processing (DSP) and Software Radio (SDR) applications due to their phase property and stability. And the complexity of FIR filter is dominated by the multiplication of the large number of coefficients with the filter input, which leads to excessive area, delay and power consumption shown in Fig. 1.

This operation is known as Multiple Common Multiplications (MCM). Hence the previous work is mainly focused on the implementation of MCM operation by considering area, delay and power parameters. Although various efficient multiplier architectures, such as (Wallace, 1964), Modified Booth (Gallagher and Swartzlander, 1994) Binary array (Costa *et al*., 2002), multipliers has been proposed, which is not necessary in MCM operation of FIR filter design. Hence the multiplication of filter coefficient with the filter input is implemented using the Shift-addition architecture (Nguyen and Chatterjee, 2000) and each multiplication is realized using addition/subtraction and shifting operations. The main bottleneck of FIR filter implementation is coefficient multiplier. The coefficient in non-reconfigurable filters are constant and Canonical Signed Digit (CSD) representation (Avizienis, 1961) and common sub-expression elimination (Hartley, 1996) is useful to implement multiplier with low-complexity. While designing reconfigurable and programmable application filter design is not constant and not easy to find the common sub-expression elimination in the coefficients. Several implementations have been proposed to design reconfigurable and programmable FIR filters. Solla and Vainio (2002) programmable Multiply Accumulate (MAC) based filter processor has been proposed, which has large delay, large area and power requirements. After that CSD based reconfigurable FIR filter architecture has been proposed, where its area and power consumption are also high. Another two efficient reconfigurable FIR filter approach has been implemented in Park *et al*. (2004) and Mahesh and Vinod (2010). These proposals have been focused on the implementation of FIR filter coefficients and the products of input data with these fixed sub coefficients. These products are distributed in the chip area instead of Processing Element (PE) to compose the coefficient multiplication. This is done by splitting the multiplier coefficients into global and local units as described in Park *et al*. (2004), where Split CSD (SCSD) representation was involved to reduce the number of adder in coefficient FIR filter implementation. In this study, signed sub-coefficient based new coefficient representation has been

**Corresponding Author:** R. Dhayabarani, Department of Electronics and Communication Engineering, V.S.B. Engineering College, Karur, India

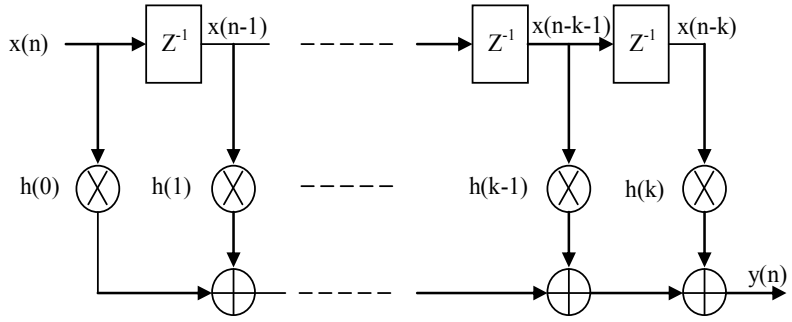


Fig. 1: Implementation of direct form FIR filter

proposed, which is suitable to implement the reconfigurable FIR filter.

## RECONFIGURABLE FIR FILTER ARCHITECTURE REVIEW

An efficient way to reduce the complexity in FIR filter implementation is done by the partial product technique which leads the filter coefficients will be partitioned to the small partial coefficients. To implement partial production the implementation of FIR filter we have to consider the N-tap FIR filter, where the transform function is described below:

$$y_n = \sum_{i=0}^{N-1} h_i \,.\, x[n-i] \quad (1)$$

where, $h_i$ is the $i^{th}$ coefficient and assumed in w-bit word length. We can divide the each filter coefficients to the k equal portions with each one m-bit (m = w/k). Then the $h_i$ coefficient will be written in terms of divided portions:

$$h_i = d_{i,0} + d_{i,1} \mathrm{X}\, 2^m + d_{i,2} \mathrm{X}\, 2^{2m} + \cdots + d_{i,k-1} \mathrm{X}\, 2^{(k-1)m} = \sum_{j=0}^{k-1} d_{i,j} \,.\, 2^{mj} \quad (2)$$

where, $d_{i,j}$ is the $j^{th}$ portion of the $i^{th}$ coefficient as the sub coefficient. From the above Eq. (1) the transfer function for the filter coefficient will be written as:

$$y[n] = \sum_{i=0}^{N-1} \sum_{j=0}^{k-1} d_{i,j} \,.\, 2^{mj} \,.\, x[n-i] = i=0N-1j=0k-1d,ji.\ xn-i.\ 2mj \quad (3)$$

As shown in the above equation, it is clear that the y [n] is composed by the sum, partial products and multiplication by power of two operations. The $d_{i,j}$.x [n-i] term is a partial product and can be implemented by simple sum/shift operation in pre-computing block. In the literature the relationship between the coefficients has been found and checked with an algorithm to extract their common factors. But this algorithm commonly has the low complexity. Pasko *et al*. (1999), algorithm is used to perform the global search but consumes so much time. The complexity of implementation is searching common factors using low complexity method.

## PROPOSED RECONFIGURABLE FIR FILTER ARCHITECTURE

We have proposed Binary Signed Sub coefficient (BSS) representation, which increases the number of common sub expression and realize the implementation process with low complexity. The proposed coefficient technique helps to reduce the size of the computer usage and reduce the required multiplexer blocks to half of the conventional design. This algorithm representation is based on the coefficient representation method and uses partial signed coefficient to represent the coefficient.

To explain the details of the proposed algorithm, let us assume $h = d_1 d_0$ as a coefficient with two m-bit sub coefficient $d_1$ and $d_0$ ($h = d_1 \mathrm{x}\ 2^m + d_0$). If $d_0$ be greater than $2^{m-1}$, its two's complement ($d_0$') is smaller than $2^{m-1}$ and we can write that:

$$h = (d_1\, X\, 2^m) + (2^m - d_0') = (d_1 + 1)\, X\, 2^m - d_0' \quad (4)$$

We can represent h as $(d_1 + 1)\bar{d}_0$, while all its sub coefficient values are smaller than $2^{m-1}$. Instead of adding the subtracting operation is applied to compose h coefficient. If the same procedure is applied to all sub coefficient, the absolute value of all sub coefficient in BSS representation will be in $0 \sim 2^{m-1}-1$ range. With this implementation the hardware requirement for the preprocessor and multiplexer will be reduced to half of the conventional design. The flowchart representation to convert the FIR filter coefficient to the proposed BSS representation is shown in Fig. 2. And it is assumed that $h_i$ is the $i^{th}$ coefficient of the filter with w-bit wordlength. And the coefficient is partitioned into k subcoefficient ($d_{i,0}$, $d_{i,1}$,…….$d_{i,k-1}$). The conversation algorithm starts from least significant subcoefficient ($d_{i,0}$). If the subcoefficient is smaller than $2^{m-1}$ nothing will be processed. But if the subcoefficient implementation is greater than $2^{m-1}$, its two's complement value will be calculated by ($C_{i,0} = 2^{m-1} - d_{i,0}$) and assigned by (m-1) new bits representation.

And signed bit representation of subcoefficient is also assigned to "1" to extract the subtracting

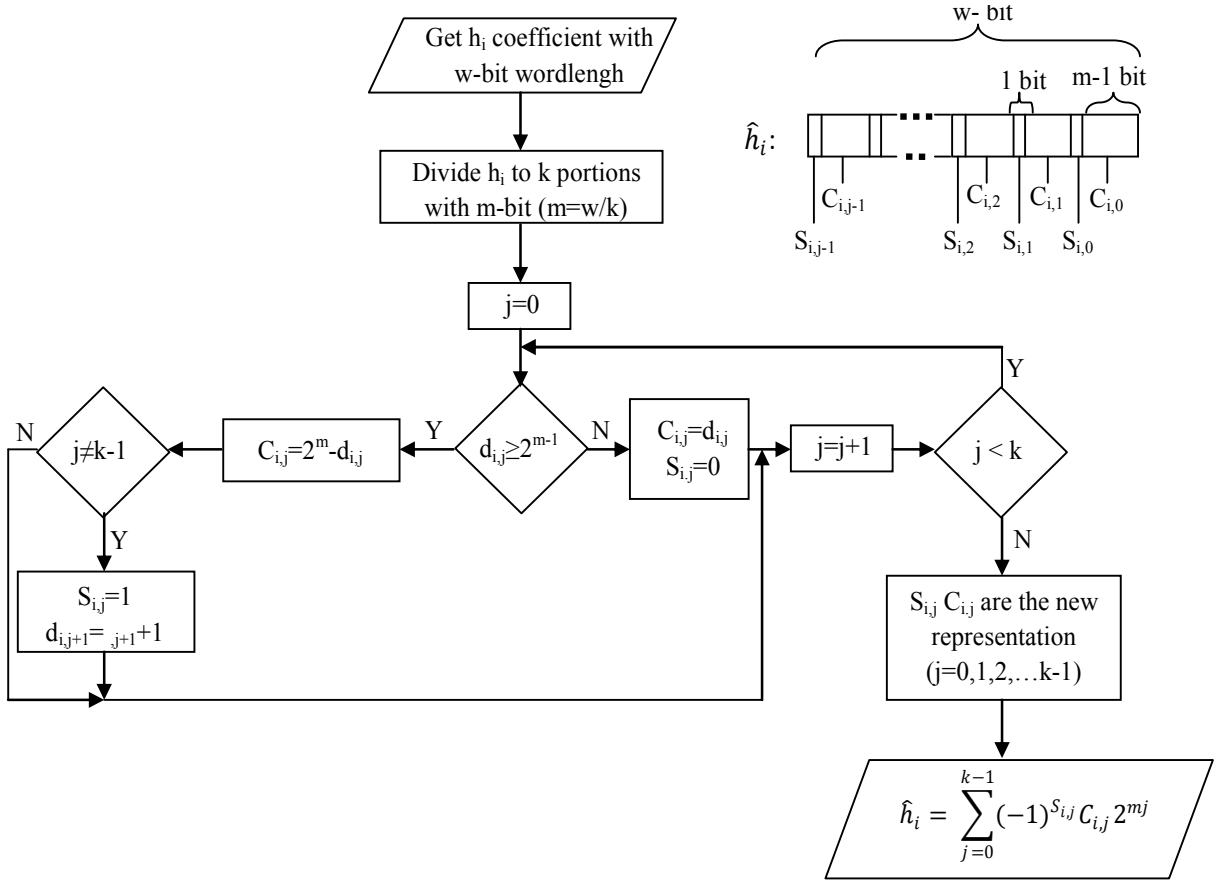


Fig. 2: Flowchart of Binary Signed Subcoefficient (BSS) representation

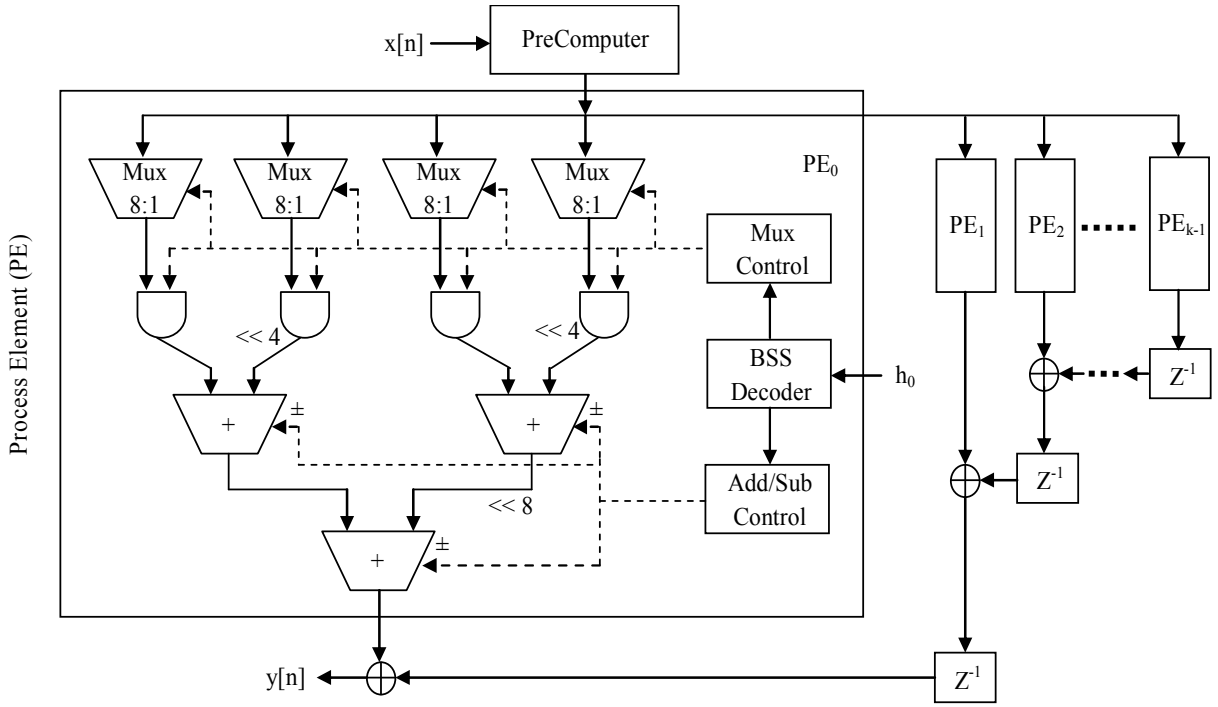


Fig. 3: Reconfigurable FIR filter architecture for 4-bit BSS representation

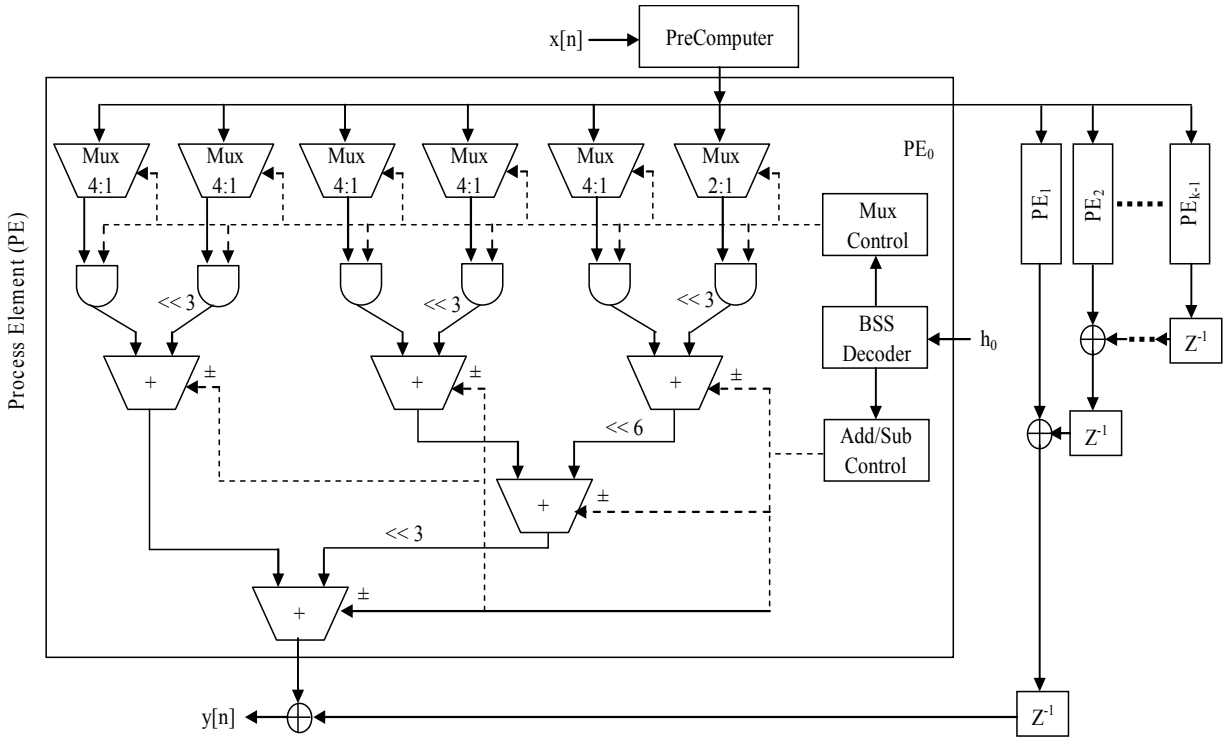


Fig. 4: Reconfigurable FIR filter architecture for 3-bit BSS representation

operation. After that, one higher significant subcoefficient value will be incremented by one as shown in Eq. (4). The same procedure is repeated to other subcoefficient to generate the new signed representation of all subcoefficients (Fig. 2). Finally the subcoefficient value for k will be generated as discussed in the equation below:

$$\hat{h}_i = \sum_{j=0}^{k-1} (-1)^{S_{i,j}} C_{i,j} \cdot 2^{mj} \quad (5)$$

From the above equation the representation of proposed Binary Signed Subcoefficient (BSS) has been designed using two-reconfigurable FIR filter.

These two FIR filter has 8-bit wordlength of input data and 16-bit wordlength of filter coefficients. And the architectural representation for 4-bit Binary Signed Subcoefficient (BSS) filter is shown in Fig. 3. From the figure the filter coefficient is divided into four 4-bit subcoefficient and its BSS representation are generated by a BSS decoder block which implements the conversion of coefficient representation as shown in Fig. 2 flowchart. In wordlength each partial products are calculated by preprocessor block using shift/sum operation and products are distributed to each tap of subcoefficient block. But there are so many partial products will be generated from the product separation. From the products the required products are selected by 8:1 multiplexer and these blocks are controlled by MUX control block. So for mux block selection sixteen partial products and four 16:1 multiplexer is needed to design conventional reconfigurable FIR filter architecture. After the selection process four partial products will be selected from the PE block, the hardwired shift operation will be combined with the addition/subtraction operation which is controlled by the arithmetic control block.

To simplify the implementation of multiplication by zero for each subcoefficient, the MUX control block will be enabled for multiplexer block followed by AND gates. Another 3-bit subcoefficient implementation of BSS representation of the reconfigurable FIR filter is shown in Fig. 4. And the partition process of 16-bit filter is different from the normal partitioning method. IN this implementation the 16-bit filter coefficient is partitioned into five 3-bit portions and the remaining $16^{th}$ bit will be considered as a sign bit.

For this implementation four partial products are calculated by preprocessor block and five 4:1 multiplexer and one 2:1 multiplexer are used to select the proper subcoefficient sequence with the desired tap coefficient multiplication. And it is noticeable that, to design the conventional reconfigurable FIR filter in the same partitioning condition the eight partial products in preprocessor block and five 8:1 multiplexer is needed. With this implementation procedure the proposed Binary Signed Subcoefficient (BSS) will decrease the size of the multiplexer and preprocessor process to half of the conventional design. This implementation will relax the partial product distribution bus, which is generated by preprocessor block.

Table 1: Comparison of FPGA resources between conventional architecture and proposed architecture

| | Conventional 4-bit representation | Conventional 3-bit representation | Proposed 4-bit BSS representation | Proposed 3-bit BSS representation |
|---|---|---|---|---|
| Total logic elements | 2976 | 1848 | 2184 | 1618 |
| Preprocessor logic elements | 45 | 18 | 19 | 4 |
| Processing Element (PE) logic elements | 302 | 214 | 199 | 138 |

## EXPERIMENTAL RESULTS

Using the proposed Binary Signed Subcoefficient (BSS) representation, two reconfigurable 10 taps FIR filters (4 and 3 bit BSS filter as shown in Fig. 3 and 4, respectively) is implemented on an Altera Stratix FPGA device, utilizing Quartus II software. These implementation results show better efficiency than the conventional method. To compare the complexity and hardware usage of the proposed reconfigurable FIR filters with the conventional work, another reconfigurable filter is designed and implemented based on the architectures of Park *et al*. (2004) for 4 bit subcoefficient and Mahesh and Vinod (2010) for 3 bit subcoefficient. The comparison between hardware usages of the proposed filter with the conventional architecture (Park *et al*., 2004) is shown in Table 1.

The conventional architecture uses 45 Logic Element in the preprocessor block while the proposed architecture uses 18 Logic Elements only. This Logic Element is reduced by decreasing the range of subcoefficient value of the Binary Signed Subcoefficient representation. The total FPGA resource usage of the proposed BSS filter has been reduced to 1848 Logic Elements, while the conventional filter has 2976 logic elements. From the comparison, the hardware resource usage of the proposed BSS representation and PE architecture will be reduced to 38% for 4-bit BSS. The implementation of the BSS decoder does not have any complexity and synthesis results show that it shows 19 logic elements only. The hardware usage of the proposed BSS filter is compared with CSM approach is shown in Table 1. And the hardware usage will be reduced to 21% in 3-bit BSS filter. In the implementation of higher order FIR filter design, the preprocessor hardware is negligible when compare with the overall filter hardware. And the filter resource usage is dominated by the Processor Elements (PE) blocks. As shown in Table 1, hardware usage of the PE block will be reduced to 38 and 21% for the proposed 4 and 3 bit BSS architecture respectively over the conventional approaches. These resource savings are achieved by reducing the multiplier blocks to half of the conventional approaches.

## CONCLUSION

Our proposed Binary Signed Subcoefficient (BSS) representation has been implemented in this study to reduce the hardware usage and complexity of the reconfigurable FIR filter. From the summary, the coefficient simplification in our proposed Binary Signed Subcoefficient (BSS) representation is useful to reduce the logic by splitting the coefficient into subcoefficient. Also reduces the size of the multiplexer and the preprocessor processing to half of the conventional design. In our proposed method two reconfigurable FIR filters were designed and synthesized with 3 and 4-bit subcoefficients. And the synthesis results of the two reconfigurable FIR filter architecture show that the reduction of resource usage will be reduced to 38 and 21% when compared with conventional two-state reconfigurable architecture.

## REFERENCES

Avizienis, A., 1961. Signed digit number representation for fast parallel arithmetic. IRE T. Electron. Comput., EC-10: 389-400.

Costa, E., S. Bampi and J. Monteiro, 2002. A new architecture for signed radix-2m pure array multipliers. Proceeding of IEEE International Conference on Computer Design: VLSI in Computers and Processors, pp: 112-117.

Gallagher, W.L. and E.E. Swartzlander, 1994. High radix booth multipliers using reduced area adder trees. Proceeding of Conference Record of the 28th Asilomar Conference on Signals, Systems and Computers, pp: 545-549.

Hartley, R.I., 1996. Subexpression sharing in filters using canonic signed digit multipliers. IEEE T. Circuits-II, 43(10): 677-688.

Mahesh, R. and A.P. Vinod, 2010. New reconfigurable architectures for implementing FIR filters with low complexity. IEEE T. Comput. Aid. D., 29(2): 275-288.

Nguyen, H.T. and A. Chatterjee, 2000. Number-splitting with shift-and-add decomposition for power and hardware optimization in linear DSP synthesis. IEEE T. VLSI Syst., 8(4): 419-424.

Park, J., W. Jeong, H. Mahmoodi-Meimand and Y. Wang, H. Choo and K. Roy, 2004. Computation sharing programmable FIR filter for low-power and high-performance applications. IEEE J. Solid-St. Circ., 39(2): 348-357.

Pasko, R., P. Schaumont, V. Derudder, S. Vernalde and D. Durackova, 1999. A new algorithm for elimination of common subexpressions. IEEE T. Comput. Aid. D., 18(1): 58-68.

Solla, T. and O. Vainio, 2002. Comparison of programmable FIR filter architectures for low power. Proceeding of 28th European Solid-State Circuits Conference, pp: 759-762.

Wallace, C.S., 1964. A suggestion for a fast multiplier. IEEE Trans. Electron., 13(1): 14-17.