

# ANALYSIS OF VARIOUS MCM ALGORITHMS FOR RECONFIGURABLE RRC FIR FILTER

M.Ragavi<sup>1</sup>, J.Harirajkumar<sup>2</sup>

<sup>1</sup>PG Scholar, Department of ECE, Sona College of Technology, TamilNadu, India

<sup>2</sup>Associate Professor, Department of ECE, Sona College of Technology, TamilNadu, India

## Abstract

Low complexity and power consumption are the key concerns while designing reconfigurable pulse shaping FIR filter for multistandard wireless communication system. In FIR filter, the single input to be multiplied by a set of coefficients known as multiple constant multiplications. This multiple constant multiplication becomes an obstruction in many applications. To overcome that, Digit Based Recoding, Canonic Sign Digit, Common Subexpression Elimination and Binary Common Subexpression Elimination algorithms are used to optimize the number of addition and subtraction operations. While designing these MCM algorithms in the architecture of RRC FIR filter, Binary Common Subexpression Elimination (BCSE) algorithm provides the better performance in terms of area and power.

**Keywords:** Multiple Constant Multiplication (MCM), Root Raised Cosine Filter (RRC), Canonic Sign Digit (CSD), Multiple Sign Digit (MSD), Common Subexpression Elimination (CSE)

\*\*\*

## 1. INTRODUCTION

FIR filter plays a vital role in the emerging wireless communication and DSP applications. Root raise cosine filter is the most popular pulse shaping technique used in mobile communication. A raised cosine filter belongs to the class of filters which satisfy the Nyquist criterion. The two conflicting requirements in telecommunication are the demand for high data rates per channel (or user) and need for more channels. Apparently as the channel bandwidth is increased to provide higher data rates, the number of channels to be allocated in a fixed spectrum must be reduced. Tackling these two contrary requirements at the same time leads to the development of RRC filters.

Hence, the multiplication of filter coefficients with the input data is generally implemented under a shift-adds architecture for the reduction in hardware. The normal shift and add implementation leads to the inclusion of maximum number of operations in the gate level design of the filter. To overcome that distinctive methods are there to implement the constant multiplication.

## 2. METHODS FOR OPTIMISATION OF MCM

### 2.1 Digit Based Recoding

The straight forward implementation of shift and add in the constant multiplication is called digit based recoding. In this, the number representation is usually binary. The binary representations are {0,1}. The binary value '1' represents that the variables are shifted according to its bit position. Then the shifted variables are added to get the final computation output.

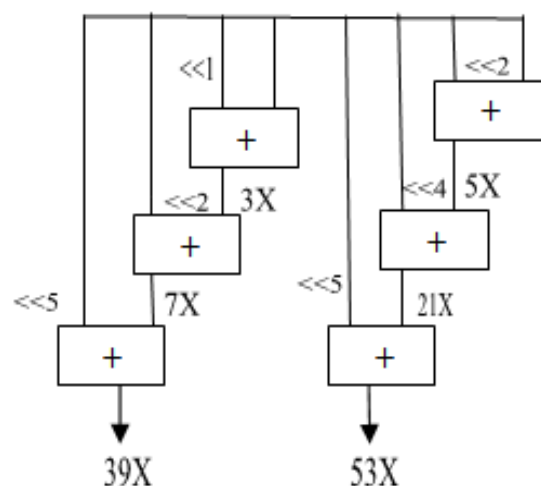
For example,

$$39x = (100111)_{\text{bin}} x = x \ll 5 + x \ll 2 + x \ll 1 + x$$

$$53x = (110101)_{\text{bin}} x = x \ll 5 + x \ll 4 + x \ll 2 + x$$

and it requires 6 addition operations shown in Fig.1

The digit based recoding technique does not allow the sharing of partial product [1-2]. So the area and power will be reduced at the gate level design. But the MCM problem will be overcome by minimizing the number of operations in the concept of partial product sharing. If the partial products are shared in the shift and add implementation, the MCM problem will be optimized.



**Fig -1:** shift and add implementation of 39x and 53x without partial product sharing

### 2.2. Canonic Sign Digit Representation (CSD)

Normally binary representation is used to represent the numerical values. Although this is the number representation of choice for digital arithmetic, alternative representations can also provide advantages when multiplications are implemented with constant shift-adds. Canonic Sign Digit Representation (CSD) is a signed digit system with digit set  $\{-1, 0, +1\}$ . This CSD representation [3] is widely used in multiplierless implementations, because it reduces the hardware usage due to its enlarged number of non-zero digits being when compared with binary representation.

#### Example

In binary, 39 as 100111 – four non zero digits

In CSD, 39 as 101000 it reduces the number of non zero digits.

### 2.3. Common Subexpression Elimination (CSE)

The Common Subexpression Elimination (CSE)[4-5] deals with the elimination of common subexpressions within the coefficients. The Common subexpressions are equivalent to the common digit patterns.

$$F_1 = 39 * X = (100111) * X = X \lll 5 + X \lll 2 + X \lll 1 + X$$

$$F_2 = 53 * X = (110101) * X = X \lll 5 + X \lll 4 + X \lll 2 + X$$

$X + X \lll 2 + X \lll 5$  is a common term. So assign common term to  $D_1$

$$D_1 = X + X \lll 2 + X \lll 5$$

$$F_1 = D_1 + X \lll 1$$

$$F_2 = D_1 + X \lll 4$$

The decompositions of  $39x$  and  $53x$  in binary are listed as follows is shown in Fig.2

$$39x = (100111)_{bin} X = X \lll 5 + X \lll 2 + X \lll 1 + X$$

$$53x = (110101)_{bin} X = X \lll 5 + X \lll 4 + X \lll 2 + X$$

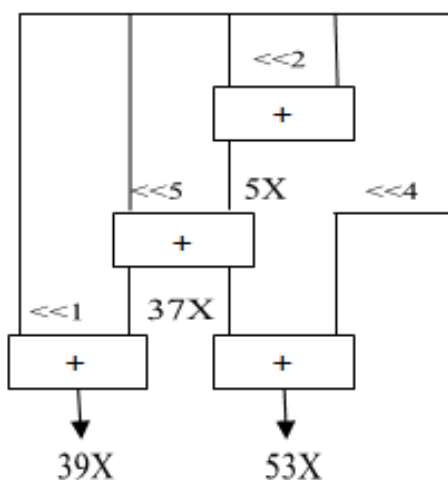


Fig -2: shift and add implementation with partial product sharing

In CSD-CSE, the goal is to identify miscellaneous occurrences of identical bit patterns that are present in the CSD representation of coefficients, and eliminate these

excessive multiplications. Drawbacks of CSD-CSE are Logical depth will be higher. Logical depth is a critical path that mainly depends on number of addition operations in a chain.

### 3. ARCHITECTURE OF ROOT RAISED COSINE FIR FILTER

The Binary Common Subexpression Elimination (BCSE) [6-7] technique focuses on eliminating redundant computations in coefficient multipliers by reusing the most common binary bit patterns (BCSs) present in coefficients. An n-bit binary number can form  $2^n - (n + 1)$  BCSs among themselves.

In the architecture of RRC FIR filter design [8] shown in Fig.3, 2-bit binary common sub expression (BCS)-based BCS elimination algorithm has been used to design an efficient constant multiplier, which is the basic component of any filter.

This technique has succeeded in initially reduces the number of multiplications per input sample and additions per input sample in comparison with individual implementation of each standard's filter while designing a root-raised-cosine finite-impulse response filter for Multistandard DUC for three different standards.

In this architecture, 2-bit BCSs ranging from 00 to 11 have been considered. Within four of these BCSs, an adder is required only for the pattern 11. This facilitates in reduction of hardware and improvement in speed while performing the constant multiplication.

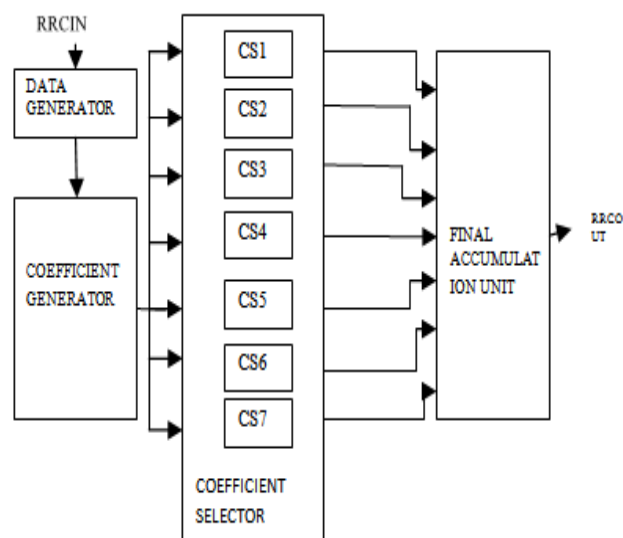


Fig -3: Architecture of RRC FIR filter

#### 3.1. Data Generator (DG) Block

DG block is used to sample the input data (RRCIN) depending on the selected value of the interpolation factor selection parameter. From the design point of view, it has been observed that 25, 37, and 49-tap filters with

interpolation factors of four, six, and eight constitute a branch filter of seven taps. This indicates that to generate the full filter response, seven subfilters are required for multiplication of the filter coefficients with the input sequence.

### 3.2. Coefficient Generator (CG) Block

The CG block performs the multiplication between the inputs and the filter coefficients. The two-phase optimization technique is proposed, which helps in reducing the hardware usage by a considerable amount to facilitate reconfigurable FIR filter implementation with low computation time and low complexity. The data flow diagram of CG block for programmable coefficient sets is shown in Fig.4

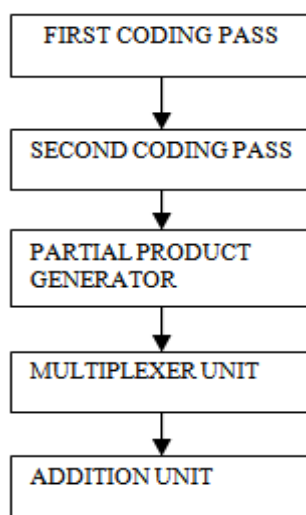


Fig -4: Flow diagram of CG Block

#### 3.2.1 First Coding Pass (FCP)

In one FCP block, two sets of 25, 27, and 49 tap filter coefficients differing only by roll-off-factor are the inputs. Inside the FCP block, three coding pass (CP) blocks are running in parallel for three different interpolation factors. Occurrence of matching between all bits is explored vertically between two coefficients of same length filter.

#### 3.2.2 Second Coding Pass (SCP)

The outputs from FCP block are three sets of coded coefficients and these coded coefficients are passing through another CP block to get the final coefficient set. In the SCP, the common terms present vertically in between these three coded coefficient sets have been found out based on the selected interpolation factor.

#### 3.2.3 Partial Product Generator (PPG) Unit

Shift-and-add method is used to generate the partial product during the multiplication operation between the input data ( $X_{in}$ ) and the filter coefficients. In BCSE technique, realizations of the common subexpression using shift-and-add method eliminates the common term present in a coefficient

#### 3.2.4. Multiplexer Unit

Depending on the coded coefficients, the multiplexer unit will select the appropriate data generated from the PPG unit. The BCS of length 2 bits would require eight 4:1 multiplexer units to produce the partial product that will be added to perform the multiplication operation considering the coefficient word length of 16 bits each.

#### 3.2.5. Addition Unit

Addition unit performs the task of summing all the outputs of the PPG block followed by eight multiplexer units. Different word length adders are required for different binary weights. The outputs from the eight multiplexers are added together. The output of the final adder passes through a two's complement circuit. The final output from this addition unit depends on the sign magnitude bit of the coded coefficient set.

### 3.3. Coefficient Selector (CS) Block

In the reconfigurable FIR filter, the CS block is used to steer proper data to the final accumulation block depending on the corresponding interpolation factor parameter. It takes the input from the CG block.

### 3.4. Final Data Accumulation Unit (FA)

The reconfigurable FIR filter is based on transposed direct form architecture. The final accumulation block has a chain of six adders and six registers as there is seven subfilters.

## 4. RESULTS AND DISCUSSION

Simulation has been done in Xilinx ISE design 14.2 for synthesizing purposes. Then the synthesis is implemented in Xilinx's SPARTRAN 3E FPGA. This result shows the 6 tap RRC FIR filter of 16 bit coefficient using the method of BCSE algorithm is shown in Fig.5

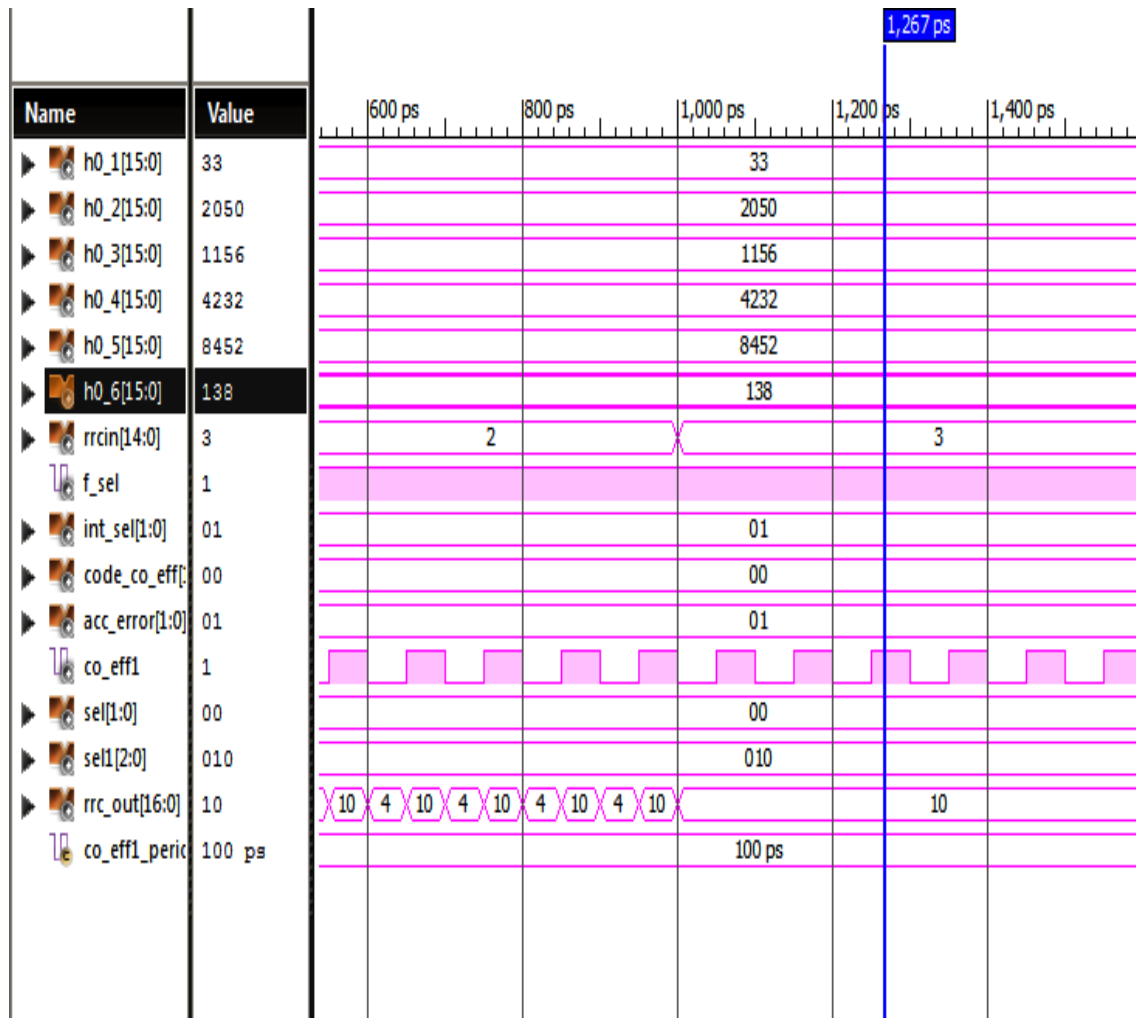


Fig -5: illustrates the implementation of RRC filter using 2-bit Binary Common Subexpression Elimination for the MCM optimization problem.

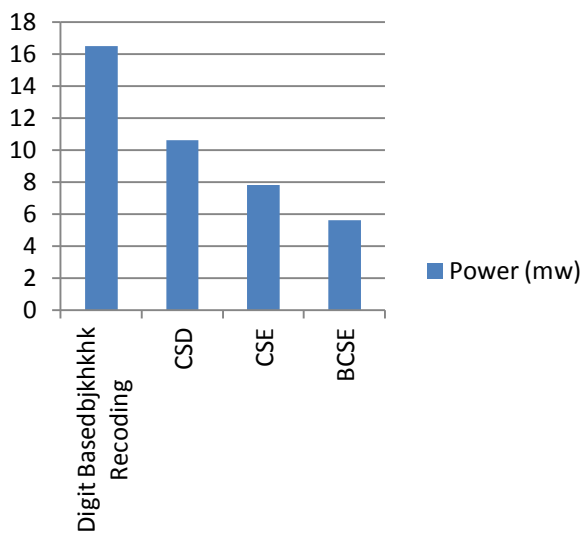


Chart-1: Comparison of Power

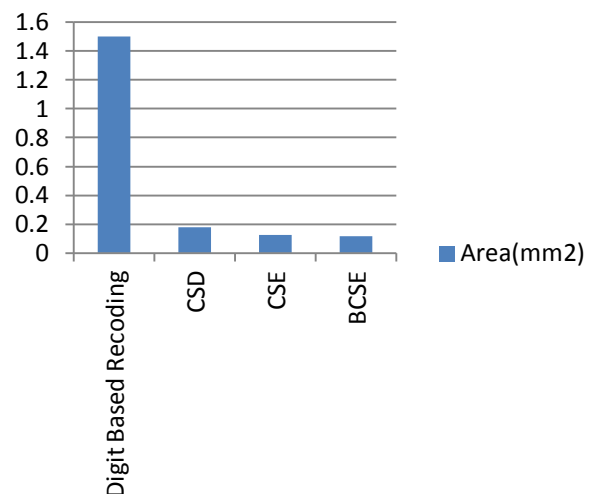
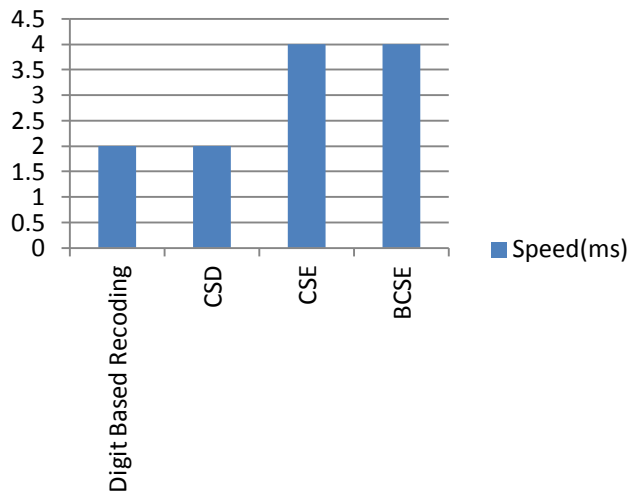


Chart -2: Comparison of Area



**Chart-3:** Comparison of Speed

From the above chart, the comparison has been made cleared that the area and power of BCSE algorithm is far better than other algorithms.

## 5. CONCLUSION

This architecture of Root Raise Cosine FIR filter which is an important component in multistandard wireless communication system consumes power and area of about 11.7% in BCSE algorithm. Furthermore improvement in the reduction of area and power along with maximum operating frequency of the design can be implemented by using graph based algorithm. This algorithm can be operated for different wordlength coefficients.

## REFERENCES

- [1]. M. Ercegovac and T. Lang. Digital Arithmetic. Morgan Kaufmann, 2003.
- [2]. K.-H. Chen and T.-D. Chieueh, "A low-power digit-based reconfigurable FIR filter," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 53, no. 8, pp. 617–621, Aug. 2006.
- [3]. K.-H. Chen and T.-D. Chieueh, "A low-power digit-based reconfigurable FIR filter," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 53, no. 8, pp. 617–621, Aug. 2006.
- [4]. Yuen H.A.H., Chi U.L., Hing-K.K., Ngai Wong, "Global optimization of common subexpressions for multiplierless synthesis of multiple constant multiplications" *IEEE Explore*, pp.119-124, 2008.
- [5]. Y. J. Yu and Y. C. Lim, "Optimization of linear phase FIR filters in dynamically expanding subexpressions space," *Circuits, Syst., Signal Process.*, vol. 29, no. 1, pp. 65–80, 2010
- [6]. 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.
- [7]. 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.

[8]. Indragith Hatai, Indragith chakrabharti and swapna Banerjee, "An Efficient VLSI Architecture of a Reconfigurable Pulse Shaping FIR Interpolation Filter for Multistandard DUC," *IEEE Transactions on Very Large Scale Integrated (VLSI) Systems*, April 2014