SAJET

South Asian Journal of Engineering and Technology Vol.2, No.18 (2016) 61–70

**ISSN No: 2454-9614**

# Delay and Area Optimization using Constant Multiplier

Viju.R.J, C.Bagavathi*

*[a)]Department of Electronics and Communication Engineering Sri Krishna College of Engineering and Technology Erode, Tamilnadu, India.*

*Corresponding Author: Viju.R.J*

E-mail: ***r.j.viju5@gmail.com,***

**Received: 10/11/2015, Revised: 11/12/2015 and Accepted: 10/03/2016**

**Abstract**

Reducing the delay and area is a major concept in VLSI architecture. In this project, a two-step optimization technique for designing a reconfigurable VLSI architecture of an interpolation filter for multi standard Digital Up Converter (DUC) with reduced delay and area consumption is proposed. Initially, the number of multiplication per input sample and number of additions per input sample are reduced when compared to the design of root-raised-cosine Finite Impulse Response filter for multi standard DUC. Multipliers are the basic element of any filter. Hence, a 2-bit binary common sub-expression based (BCS) elimination algorithm has been proposed to design such constant multipliers. This technique has succeeded in reducing the delay, area and power usage along with 36% improvement in operating frequency compared to the earlier 3-bit BCS-based technique for designing the multi-standard DUC

*Index terms— FIR filter, Digital Up Converter, Binary Common Sub-expression*

**Reviewed by* **ICETSET'16** *organizing committee*

## .1. Introduction

### *1.1. General*

The explosive development in Very Large Scale Integrated (VLSI) circuits in recent years has become the pacing force in the development of new communication systems. In modern telecommunication system the major requirement in communication devices are high data transfer rate and high channel capacity. For satisfying the above requirements, a concept known as Software Defined Radio (SDR) **[2]** has been introduced. Different problems are encountered wile designing VLSI architecture. The circuits designed may be general purpose integrated circuits such as microprocessor, digital signal processor and memories. They are characterized by a wide range of applications. The most important entities are, speed, area, delay, design time and power dissipation.

### *1.2 Software Defined Radio (SDR)*

Software Defined Radio (SDR) **[2]** is a radio communication system where components that have been typically implemented in hardware are instead implemented by means of software on a personal computer or

embedded system. SDR software performs all of the demodulation, filtering, and signal enhancement. In an SDR system, multiple standards can be realized in a single chip by providing a programmable channel select filter at the baseband level.

Different standards have different channel bandwidths, sampling rates, carrier to noise ratio, blocking andinterference profiles. Hence, a design of low power, low

area and low complexity reconfigurable channel filter for data rate conversion in SDR system is necessary.

*1.3 Digital Up Converter (DUC)*

The digital up converter (DUC) is a device which converts digital baseband signal up to a pass band signal. The input signal is sampled at a relatively low sampling rate. This baseband signal is filtered and converted to a higher sampling rate and then modulated with a carrier signal generated from the direct digital synthesizer. The DUC can be extensively used in wireless and wire line communication systems.

*1.4 FIR Filter*

A finite impulse response (FIR) [1] filter is a filter structure that can be used to implement almost any sort of frequency response digitally. An FIR filter is usually implemented by using a series of delays, multipliers and adders to create the filters output. The process of selecting the filters length and coefficients are called filter design. The longer the filter, the more finely the response can be tuned. An FIR filter can produce weighted average of the N most recent inputs.

*1.5 Canonical Signed Digit (CSD)*

CSD is a special manner for encoding a value in a signed digit representation, which itself is a non-unique representation and allows one number to be represented in many ways. Probability of digit being zero is close to 66% and leads to efficient implementations of add/subtract networks. CSD is obtained by transforming every sequence of zero followed by ones (011...1) into + followed by zeros and the least significant bit by - (+0…0- ).

**2.Existing Methodology**

*2.1Description*

To improve the power consumption, a combination of symmetrical retimed direct form architecture, balanced modular architecture, separated signed processing architecture, and modified Canonical Signed Digit (CSD) technique-based Finite Impulse Response (FIR) filter **[5]** have been used. A multiplier-less FIR interpolator with smaller area are used in which the efficient use of Lookup Tables (LUTs) in the design helps to reduce the power and area. An area-delay-power efficient FIR filter **[5]** by systolic decomposition of Distributed Arithmetic (DA)-based inner-product computation has been used. Based on modified DA technique, high-speed and medium-speed FIR filter architectures are developed. The high-speed FIR filter architecture where the LUTs are working in parallel draws a very high current and involves huge area consumption.

Common Sub-expression Elimination (CSE) **[6]** technique, in which multiplication operations between the

constant coefficients and inputs are performed by shift and add operations, is known to be most recently used technique. The number of addition operations used to perform the multiplication operation defines the Logic Depth (LD) or the critical path of the circuit.

CSE algorithm is a useful solution in achieving less hardware footprint for implementing higher order digital filters. A low complexity architecture based on Binary CSE (BCSE) algorithm consumes less hardware and power than those of CSD-CSE method using a common constant/programmable shift-and-add block **[6].**

Two different types of architecture used for the addition and shift unit,

1. Constant Shift Method (CSM)
2. Programmable Shift Method (PSM)

*2.2 Constant Shift Method (CSM)*

In coded Shift Method, LUT is used to store the coefficients. These coefficients are partitioned into groups of 3-bits and used to select the signal from the multiplexers. Various types of multiplexer are required is [n/3], where n is the wordlength of the filter coefficients. With the help of 8-bit coefficient the CSM can be explained in 'h = 0.11111111'. Here, h is the worst case 8-bit coefficients hence all the bits are nonzero and needs the addition and shift operation is maximized. Here using 8-bit, that is n = 8, number of multiplexers required is 3.

$Y= 2^{-1} x + 2^{-2}x + 2^{-3}x + 2^{-4}x + 2^{-5}x + 2^{-6}x + 2^{-7}x + 2^{-8}x.$ (2.1)

$h = 2^{-1}(x + 2^{-1} x +2^{-2} x+ 2^{-3} x+2^{-4} x+2^{-5} x +2^{-6} x+2^{-7} x).$ (2.2)

Then the shift operation is performed by the final shifter. The circuit does not required programmable shifters. Wordlength of the coefficient is 16 bits. In LUT, the filter coefficients are stored for the sign bit. The first bit is used to represent the integer part of the coefficients remaining bits are used to represent the fractional part of the coefficients. There are 3-bits with 8combination are possible. Mux1 to Mux7 are used. Output needs to be complemented in Mux7 hence it is a 2:1 Mux. Fewer number of shift and add unit has been proposed in 3-bit BCSE than 4-bit BCSE. Four 2:1 multiplexers is equivalent to one 8:1 multiplexer. In CSM approach, the coefficients are stored in LUT hence coefficient multiplication is not avoided.

*2.3 Architecture of CSM*

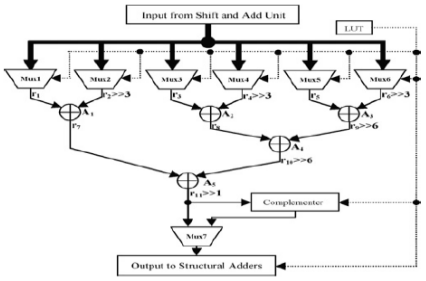


Fig. 1 Architecture of CSM

*2.4 Programmable Shift Method (PSM)*

In this section,reconfigurability is incorporated into BCSE and the filter coefficients are analyzed using BCSE. Resulting coefficients are stored in LUT. Multiplexer units are obtained from the filter coefficients. After considering the number of nonzero operands, thenumber of multiplexers are selected. Worst case coefficient is defined considering the maximum number of operands. There are two advantages in this method.The number of addition is reduced compared to CSM and it offers the flexibility of changing the wordlength. The coefficient wordlength of proposed PSM architecture changed dynamically.

*2.5 Architecture of PSM*

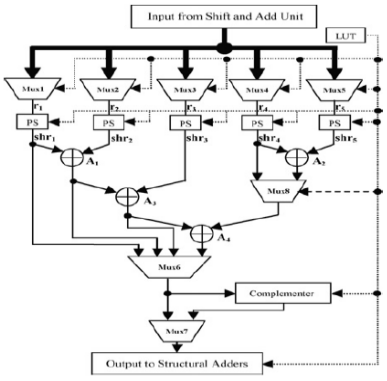


Fig. 2 Architecture of PSM

*2.6 Disadvantages*

The major disadvantages of the existing system are listed below:

1. The constant shift multiplication-based FIR filter design involves use of redundant adder in the multiplier block. This additional hardware usage consumes more area and power, and makes the design unsuitable for SDR [6] system where low power and low area consumptions are the key concerns.
2. The reduction in power has been achieved by compromising with the speed of operation that makes the symmetrical retimed direct form architecture unsuitable for the SDR **[2]** system.

## 3. Proposed Methodology

*3.1 Proposed System*

To overcome the previously discussed disadvantages, a new reconfigurable architecture has been proposed in this project for initial reduction of Multiplications Per Input Sample (MPIS**) [8]** and Additions Per Input Sample (APIS**) [1]** and subsequent reduction of hardware and power by designing an efficient constant multiplier using 2-bit Binary Common Sub-expression (BCS).The use of 2-bit BCS will lead to a good amount of saving in the propagation delay when compared with the 3-bit BCS based constant multiplier design **[9].**

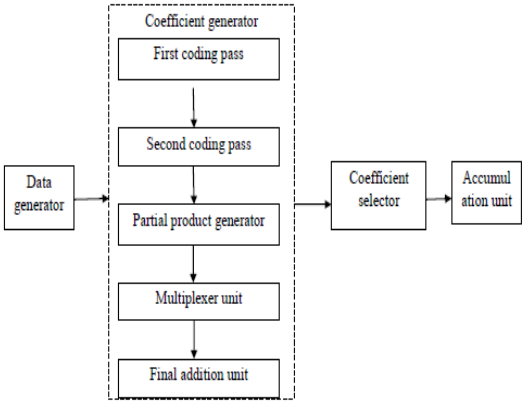


Fig. 3 Simplified Block Diagram

*3.2 Architectural Design*

The reconfigurable architecture of FIR **[2]** interpolation filter based on the proposed method is shown below

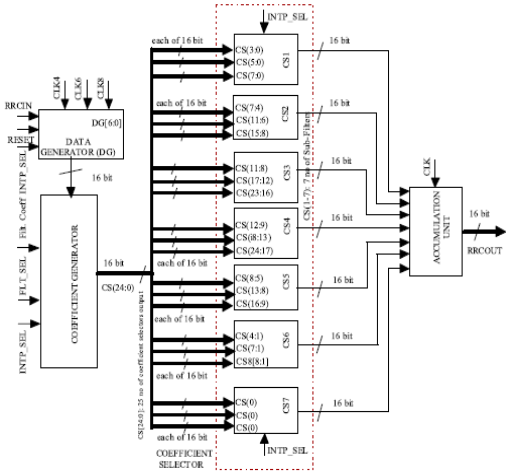


Fig. 4 Architecture of reconfigurable RRC filter

The proposed reconfigurable RRC filter architecture consists of the following major modules. They are Data Generator (DG), a Coefficient Generator (CG), a Coefficient Selector (CS) and an Accumulation Unit block (FA).

*3.3 Data Generator (DG)*

DG block is used to sample the input data (RRCIN) depending on the selected value of the interpolation factor selection parameter (INTP_SEL). 25-, 37- and 49- tap filters with interpolation factors of four, six, and eight constitute a branch filter of seven taps. i.e.)

[25/4] = [37/6] = [49/8] = 7.

This indicates that, to generate the full filter response seven sub filters are required for multiplication of the filter coefficients with input sequence.

*3.4 Coefficient Generator (CG)*

The CG block performs the multiplication between the inputs and the filter coefficients. The proposed two phase optimization technique helps in reducing the hardware usage by a considerable amount to facilitate reconfigurable FIR filter [5] implementation with low computation time and low complexity.The flow diagram of CG block for programmable coefficient sets is shown below in figure 4.3. The code generator includes the following blocks namely, First Coding pass (FCP), Second Coding Pass (SCP), Partial Product Generator (PPG), Multiplexer Unit (MU) and Final Addition (FA) block.The functionality of each block represented in fig 3.2.2 is described as follows.

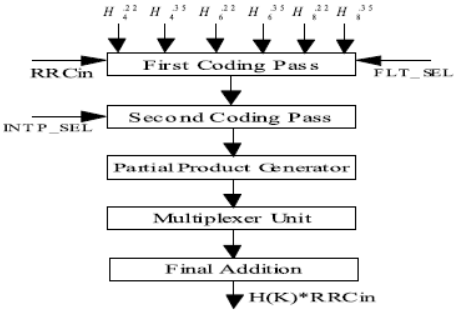


Fig. 5CoefficientGenerator Block

*3.5 First Coding Pass (FCP)*

The input to the FCP block are two sets of 25-, 27-, and 49- tap filter coefficients which differ only by roll off factor. Three coding blocks are running in parallel for three different interpolation factors inside the FCP block. Matching between all bits is explored vertically between two coefficients of same length filter.

The architecture for implementation of First Coding Pass is shown below in fig 3.2.3

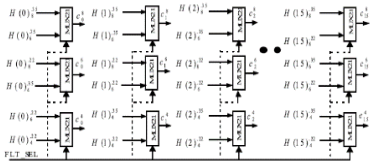

Fig. 6Architecture of FCP block

In the FCP block, the coefficient sets are multiplexed through one 2:1 multiplexer, where one control parameter (FLT_SEL) selects the desired filter depending on the roll off factor. This multiplexing technique helps in decreasing the requirement of the multiplier by 50% as the total number of coefficients is 111 instead of the initial requirement of 222.

*3.6 Second Coding Pass (SCP)*

In the Second Code Passing (SCP), the coefficients obtained from the FCP block are passed through another set of multiplexers; where one control parameter (INTP_SEL) selects the desired filter depending on the interpolation factor. The total number of filter coefficients that will process further from the earlier requirement of 111 is reduced using this technique.

The outputs from FCP block are three sets of coded coefficients that are 13, 19, and 25 in number and are

passed through another CP block to get the final coefficient set. In this block, the common terms present vertically in between these three coded coefficient sets have been found out and coded accordingly.The architecture view of Second code pass block is shown below in fig-3.2.4,

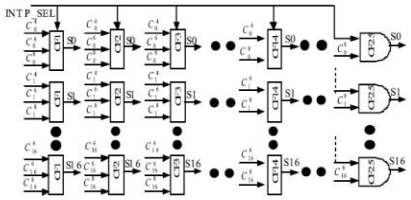

Fig. 6 Architecture of SCP block

The combination of FCP and SCP steps reduces the requirement of MPIS from 42 to 7 and APIS from 36 to 6, which facilitates 83.3% improvement of this design. APIS and MPIS can be further reduced by considering more filters of different specifications.

*3.7.Partial Product Generator (PPG)*

The partial product during the multiplication operation between the input data ($X_{in}$) and the filter coefficients is generated using shift and add method. In BCSE technique, realizations of the common sub expression using shift and add method **[6]** eliminates the common term present in a coefficient. The architecture for implementation of PPG block is shown below in fig 3.2.5.

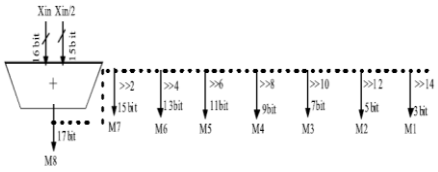

Fig. 7 Architecture of PPG block

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 reduction in hardware and improvement in speed while performing the multiplication operation.

*BCSE Technique*

Instead of 3 bit BCSE technique, 2 bit BCSE technique [6] is used. In this technique the Logic Depth can be defined as,

$LD_{2BCS}= [Log_2 2] + Log_2 [16/2] = 4$ (3.1)

Where the term $log_2 2$ is due to the two bit BCS and the term $log_2 [16/2]$ is due to 16 bit word length of coefficients. Also the Propagation Delay can be defined as

$T_{2BCS}=4 \times t_{add} + t_{4:1mux} + t_{acc}$ **(3.2)**

Where,$t_{add}$is the delay of each adder used in the constant multiplier.

$t_{4:1mux}$ is the delay for the 4:1 multiplexer.

$t_{acc}$is the delay for final adder in the delay chain of FIR filter.

The 2 bit BCS leads to a good amount of saving in the propagation delay compared with the 3 bit BCS based constant multiplier design. In any FIR filter, the multiplication operation [1] between the inputs and the coefficients with word length 16 bits can be given as

$y1=x1+2^{-1}x1+2^{-2}x1+2^{-3}x1 +\ldots+2^{-14}x1+2^{-15}x1$ **(3.3)**

By considering 2 bit BCS technique i.e.) $x2 = x1 + 2^{-1}x1$, the above equation (3.3) rewritten as

$y1=x2+2^{-2}x2+2^{-4}x2 +\ldots +2^{-10}x2 +2^{-12}x2 + 2^{-14}x2$(3.4)

In the proposed architecture, the shift add unit has been grouped in eight preshifted values of $2N + 1$ bit, where $N = 8, 7, 6, 5, 4, 3, 2, 1$ to implement the above equation **(4.4).** This will help in reducing the multiplexer and the adder width. As this shifting is done prior to the addition operation, the maximum error due to truncation has been precalculated and added in the final addition operation of the constant multiplier block.

*3.8 Multiplexer Unit*

The multiplexer unit selects the appropriate data generated from the PPG unit depending on the coded coefficients. Eight 3.3.5 multiplexer units are required to produce the partial product for BCS of length 2 bits. The architecture of Multiplexer Unit is shown below in fig 3.2.7. The obtained partial product will be added to perform the multiplication operation. The coefficient word length of 16 bits each is considered for this operation.

*3.9 Final Addition Unit*

The outputs of the PPG block followed by eight multiplexer units are summed together in the Addition Unit. The output from eight multiplexers is M7 – M0. The obtained final output is passed through a two's complementary circuit. The architecture of final addition is shown above in fig 3.2.7

Based on the different binary weights, different word length adders are required. The resultant output from the addition unit depends on the sign magnitude bit of the coded coefficient set.

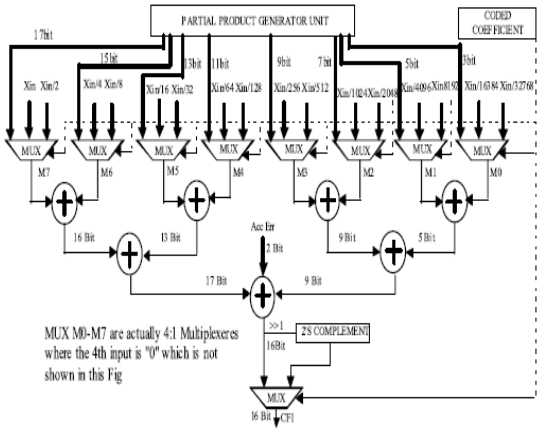


Fig. 8 Architecture of multiplexer and final addition unit

*3.10 Coefficient Selector*

In the reconfigurable FIR filter [5], depending on the corresponding interpolation factor the coefficient selector block steer the proper data to the final accumulation unit. The hardware architecture of code selector block is shown below in fig 3.2.8,

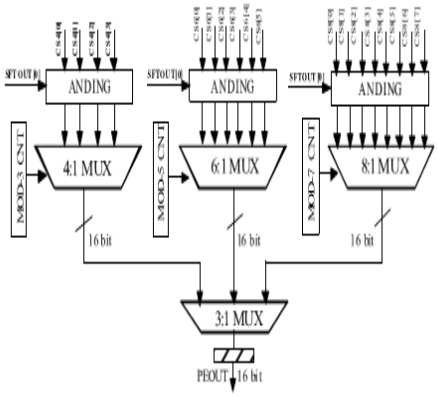


Fig. 9 Hardware architecture of CS block

The output of the Code Generator block is the input to the Coefficient Selector block.

*3.11 Final Accumulation Unit*

The reconfigurable FIR filter is based on transposed direct form architecture. As there are seven sub filters, the final accumulation data unit includes a chain of six adders and six registers.

**4. Simulation Result**

By comparing both the existing and proposed technique using various algorithm with 16 bit area, delay and power has to be optimized. Delay, area and powerhasbeen minimizedto41%, 34% & 38% respectively. Comparison are shown below,

TABLE 1
Comparison of simulation outputs

| S.NO | PARAMETERS | EXISTING METHOD | PROPOSED METHOD |
|---|---|---|---|
| 1 | DELAY | 16.683 ns | 11.703 ns |
| 2 | AREA | 9,967(gate count) | 7,384 (gate count) |
| 3 | POWER | 53.5 mW | 39 mW |

**5.Conclusion And Future Scope**

Different problems has been encountered while the reduction of Multiplication Per Input Sample and Reduction Per Input Sample. The area and delay has to be optimised for the constant multiplier.Binary Common Subexpression Elimination methods are implemented in final multiplication and addition unit to reduce the addition and multiplication unit.The number of gates has to be minimised for obtaining the minimum area and delay.

A constant multiplier architecture is used instead of using the Final Addition and Multiplication Unit. Number of addition and multiplication operation should minimized by using the 2-bit BCSE algorithm with the help of multiplexer unit in the aim of optimizing area and delay.

## References

[1] D. Shi and Y. J. Yu, "Design of linear phase FIR filters with high probability of achieving minimum number of adders," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 58, no. 1, pp. 126–136, Jan. 2011.

[2] J. Mitola, "The software radio architecture," IEEE Commun. Mag., vol. 33, no. 5, pp. 26–38, May 1995.

[3] J. Xie, J. He, and G. Tan, "FPGA realization of FIR filters for high-speed and medium-speed by using modified distributed arithmetic architectures," Microelectron. J., vol. 41, no. 6, pp. 365–371, Jun. 2010.

[4] O. Gustafsson, "Lower bounds for constant multiplication problems, "IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 54, no. 11, pp. 974–978, Nov. 2007.

[5] P. K. Meher, S. Chandrasekaran, and A. Amira, "FPGA realization of FIR filters by efficient and flexible systolization using distributed arithmetic," IEEE Trans. Signal Process, vol. 56, no. 7, pp. 3009–3017, Jul. 2008.

[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] S.-F. Hsiao, J.-H. Zhang Jian, and M.-C. Chen, "Low-cost FIR filter designs based on faithfully rounded truncated multiple constant multiplication/accumulation," IEEE Trans. Circuits Syst. II,Exp. Briefs, vol. 60, no. 5, pp. 287–291, May 2013.

[9] S.-J. Lee, J.-W. Choi, S. W. Kim, and J. Park, "A reconfigurable FIR filter architecture to trade off filter performance for dynamic power consumption," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 19, no. 12, pp. 2221–2228, Dec. 2011.