ASIAN JOURNAL OF PHARMACEUTICAL AND CLINICAL RESEARCH

INNOVARE ACADEMIC SCIENCES
Knowledge to Innovation

Special Issue (April)

Online - 2455-3891
Print - 0974-2441

Full Proceeding Paper

EFFICIENT FINITE IMPULSE RESPONSE FILTER ARCHITECTURE USING MULTIPLE CONSTANT MULTIPLICATION AND COMMON SUB-EXPRESSION ELIMINATION TECHNIQUES

BHARGAV SHUKLA, AUGUSTA SOPHY BEULET P*

School of Electronics Engineering, VIT University, Chennai, Tamil Nadu, India. Email: sophyguna@gmail.com

Received: 23 January 2017, Revised and Accepted: 03 March 2017

ABSTRACT

Objective: This paper introduces the computationally efficient, low power, high speed partial reconfigurable Finite impulse response (FIR) filter design using multiple constant multiplication technique (MCM). The complexity of many digital signal processing (DSP) systems is reduced by multiple constant multiplication operation.

Methods: Multiple constant multiplications (MCM) along with methods like common sub expression elimination (CSE) is used for the better performance of digital signal processing systems. This paper introduces a CSE operation of finite Impulse response filter design which is solved with decreased number of operators.

Results: Using these techniques shows that the area efficiency is increased when compared with designs based on direct form implementation of FIR filters.

Conclusion: This method has achieved minimum area and delay with a high process rate. The future work will be for using CSE for 0-1 ILP.

Keywords: Filters, Multiple constant multiplication, Common sub-expression elimination, Finite impulse response, Canonical signed digit, Binary signed digit, Horizontal sub-expression elimination, Vertical sub-expression elimination.

© 2017 The Authors. Published by Innovare Academic Sciences Pvt Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4. 0/) DOI: http://dx.doi.org/10.22159/ajpcr.2017.v10s1.19752

INTRODUCTION

Finite impulse response (FIR) filters are very most important in digital signal processing (DSP) systems since their liner characteristic implementations make them very useful for building stable, high-performance filters. The direct form and transposed-form FIR filter implementations are as shown in Fig. 1a and b, respectively. Although hardware wise both architectures have similar complexity but the transposed form is generally preferred because of its better performance and power efficiency [1].

The multiplier block of the digital FIR filter in its transposed form Fig. 1b, where the multiplication of the filter coefficients with the filter input is multiple constant multiplication (MCM) is defined as the process of finding the minimum number of addition/subtraction operations. It is arithmetic operation that multiplies a set of fixed point constants with the same fixed-point variable x.

MCM TECHNIQUE

MCM is a simple way of realizing the constant multiplications using a shift-adds architecture [2]. It is first to define the constants under a particular number representation, and second, for the non-zero digits in the representation of the constant, is to shift the input variable according to the digit positions and add or subtract the shifted variable with respect to the digit values. As simple example, consider the constant multiplications 22x and 44x. Their decompositions are in MCM are listed as follows as shown in Fig. 2 [3].

22x = (10110) bin x = x>>1+x>>2+x>>4

44x = (101100) bin x = x>>2+x>>3+x>>5

COMMON SUB-EXPRESSION ELIMINATION (CSE) TECHNIQUE

To reduce the number of operators, first use a CSE algorithm with a based on the canonical signed digit (CSD) representation of filter coefficients for implementing low complexity FIR filters [4]. In CSDs, it is difficult to deal with "+" and "−" [5]. The observation is that the number of unpaired bits is considerably few for binary coefficients compared to CSD coefficients, particularly for higher order FIR filters.

The binary CSE (BCSE) algorithm deals with elimination of unnecessary BCSE that occurs within the coefficients. The BCSE technique focuses on eliminating unnecessary computations in coefficient multipliers by reusing the most common binary bit patterns (binary common sub-expressions [BCSs]) present in coefficients [6]. The number of BCSs that can be formed in an n-bit binary number is 2^n - (n+1).

For example, 3-bit binary representation can form four BCSs, which are [0 1 1], [1 0 1], [1 1 0], and [1 1 1]. These BCSs can be expressed as:

$$[0\ 1\ 1]=x_2=2^{-1}x+2^{-2}x \quad (1)$$

$$[1\ 0\ 1]=x_3=x+2^{-2}x \quad (2)$$

$$[1\ 1\ 0]=x_4=x+2^{-1}x \quad (3)$$

$$[1\ 1\ 1]=x_5=x+2^{-1}x+2^{-2}x \quad (4)$$

Where, x is the input signal.

Note that other BCSs such as [0 0 1], [0 1 0], and [1 0 0] do not require any adder for implementation as they have only one nonzero bit. A simple realization of above BCSs would require five adders.

However, x2 can be obtained from x4 by right shift operation:

$$X_2=2^{-1}x+2^{-2}x=2^{-1}(x+2^{-1}x)=2^{-1}x_4 \quad (5)$$

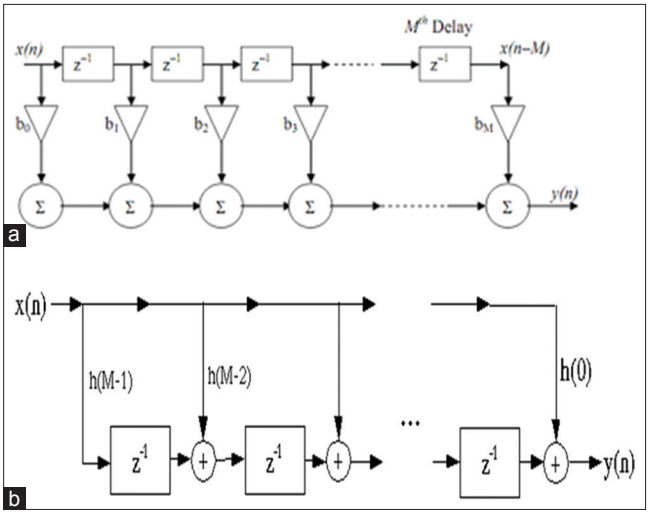

Fig. 1: Finite impulse response implementation (a) direct form (b) transposed form with generic multipliers obtained and has significant impact on the complexity and performance of the design because it requires large number of multiplier constant

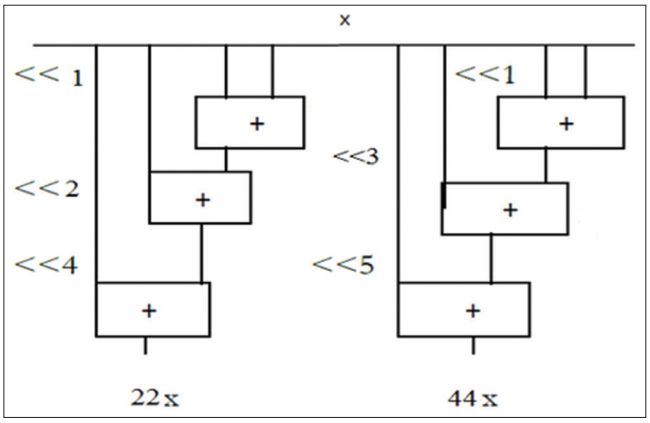

Fig. 2: Multiple constant multiplication six adders

Also x_5 can be obtained from x_4 using an adder:

$$X_5=x+2^{-1}x+2^{-2}x=x_4+2^{-2}x \quad (6)$$

Thus, only three adders are required to perceive the BCs x_2-x_5.

Greedy CSE algorithm

The new CSE algorithm combines three techniques, binary horizontal sub-expression elimination, binary vertical sub-expression elimination and hardwiring of the final stages, which reduces the number of adders. This technique focuses on eliminating redundancy in coefficient multiplier [7]. For example, x_3-x_6 are formed from the binary representation of coefficient as follows:

$$[0\ 1\ 1]=x_3=2^{-1}x_1+2^{-2}x_1 \quad (7)$$

$$[1\ 0\ 1]=x_4=x_1+2^{-2}x_1 \quad (8)$$

$$[1\ 1\ 0]=x_5=x_1+2^{-1}x_1 \quad (9)$$

$$[1\ 1\ 1]=x_6=x_1+2^{-1}x_1+2^{-2}x_1 \quad (10)$$

A direct realization of the binary HCSs (BHCSs) Equations 7 to 10 would require 5 adders. However, as x_5 can be obtained from x_3 by a shift operation and x_6 from x_5 using an adder, only three adders are required the BHCSs.

Table 1: Design utilization of summary using FIR direct form

Device utilization summary			
Logic utilization	**Used**	**Available**	**Utilization (%)**
Number of slice flip flops	66	4,896	1
Number of 4 input LUTs	64	4,896	1
Number of occupied Slices	41	2,448	1
Number of slices containing only related logic	41	41	100
Number of slices containing unrelated logic	0	41	0
Total number of 4 input LUTs	71	4,896	1
Number used as logic	62		
Number used as a route-thru	6		
Number used as shift registers	2		
Number of bonded IOBs	25	108	23
Number of BUFGMUX	1	24	4
Number of MULTI18X18SIOs	3	12	25
Average fanout of non-clock nets	1.49		

FIR: Finite impulse response, LUT: Look up table, IOBs: Input output blocks

Table 2: Design utilization of summary using MCM and CSE

Device utilization summary (estimated values)			
Logic utilization	**Used**	**Available**	**Utilization (%)**
Number of slices	38	2448	1
Number of slice flip flops	66	4896	1
Number of 4 input LUTs	66	4896	1
Number of bonded IOBs	25	108	23
Number of MULTI18X18SIOs	3	12	25
Number of GCLKs	1	21	4

MCM: Multiple constant multiplication, CSE: Common sub-expression elimination, GCLKs: Global clock, LUTs: Look up tables, IOBs: Input output blocks

Table 3: Detailed dimensions of direct form MCM and CSE

Techniques used	Delay	Number of operators
Direct form	16.147ns	381
MCM and CSE	9.892ns	192

MCM: Multiple constant multiplication, CSE: Common sub-expression elimination, ns: Not significant

$$X_3=2^{-1}x_1+2^{-2}x_1=2^{-1}(x_1+2^{-1}x_1)=2^{-1}x_5 \quad (11)$$

$$X_6=x_1+2^{-1}x_1+2^{-2}x_1=x_5+2^{-2}x_1=x_5+2^{-2}x_1 \quad (12)$$

LITERATURE SURVEY

There is a lot of work has been done in this field using different techniques and motives to provide the best system. The literature survey of different researchers has been given below:

Soderstrand *et al.* [4] proposed a hardware optimization technique based on minimum adder CSD multiplier blocks is combined with a technique for trading adders to reduce hardware requirements for FIR filter coefficients. Noise free filters can only be achieved using FIR filters because FIR filters can always designed using a sufficient number of bits in the multipliers that rounding or truncation after multiple is not necessary. Thus, an idea is carried out by increasing the filter order and decreasing the bits to minimize the hardware

requirement. This technique is implemented in field programmable gate array (FPGA) which is an integrated circuit, and it requires high cost.

Vijay *et al.* [5] represented the complexity of FIR filters and reduced using a number of adders and subtractors for implementation of

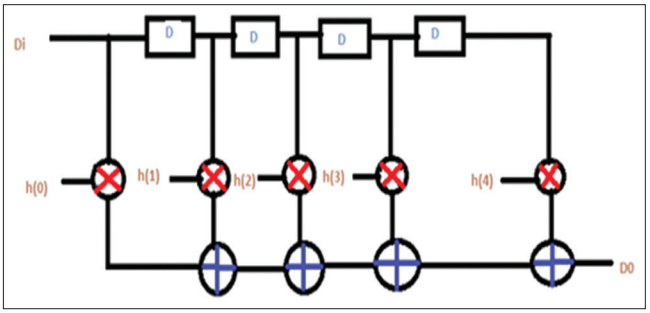

Fig. 3: 5-tap finite impulse response filter

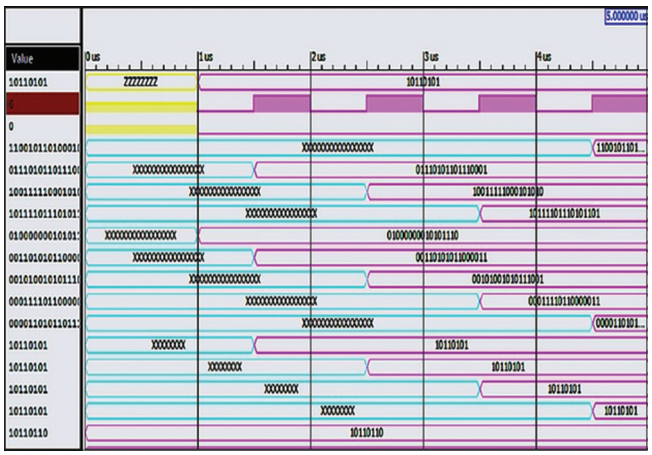

Fig. 4: Output of 5-tap finite impulse response filter with direct form

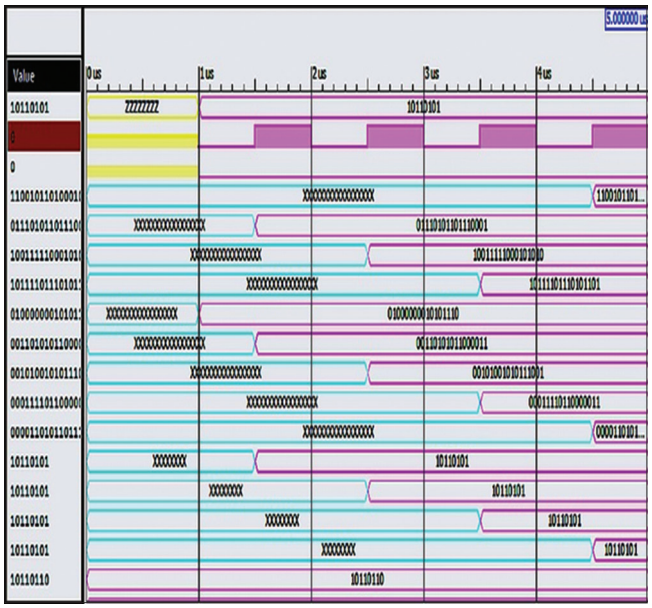

Fig. 5: Output of 5-tap finite impulse response filter with multiple constant multiplication and common sub-expression elimination technique

coefficient multipliers. A CSE algorithm with a based on the CSD representation of coefficients of filter for implementing low complexity FIR filters. In CSE, first, write the expression in binary form. According to bit position, shift the variable and added up the shifted variable. The next step is to maximize the groups of the sub expressions for the reduction of operators. CSDs used where there is the occurrence of consecutive non-zero digits. Steps of converting a binary number into canonical digits repeated again and again until there are no consecutive non-zero digits. Thus, it is very difficult to deal with CSDs.

Aksoy *et al.* [8] have represented an exact CSE for sharing terms in MCM. This algorithm deals with the Boolean networks to cover all terms which generate the set of coefficients in MCM. Linear programming is used to reduce the number of gates used in it. This algorithm handles the binary and CSD representations for the coefficients. This paper results the reducing in delay, but area of filter is increased. Hence, this algorithm is not efficient to decrease the number of operators.

Aksoy *et al.* [9] have represented the design of low complexity using bit-parallel MCM operation which reduces the complexity of many DSP systems. In digital-serial design, input data are divided into the number of bits. Then, it processed the data serially bit by bit but applies each bit in parallel. Digital-serial computation plays an important role when bit serial implementations cannot meet the delay requirements, then bits sends in parallel need more hardware. FIR filters which are under the shifts-adds architecture having significant area reduction as compared to that filter design which is implemented using digital serial constant multiplier. Some attention had been given to the digit-serial MCM design which offers low complexity MCM operation at the cost of an increased delay. Thus, it was a tradeoff between time and area of the filter.

Al-Hasani *et al.* [10] have represented CSE algorithm which is used to minimize the complexity of the MCM operation. MCM used binary signed digit (BSD) number system to design the coefficients. The BSD used in this paper which gives a better performance over CSD. BSD representation is used to find the possible decompositions which results the possibility of finding MCM realizations with minimum logic depth is increased because the sub expression space becomes large. In BSD, first, a decimal number is converted into binary digits. Each digit associated with a sign, positive, or negative. Hence, a decimal number can be represented in many ways using BSDs. Thus, BSD show redundancy.

Aksoy *et al.* [11] have represented MCM which realizes the multiplication of constants by a variable. It can be implemented using generic multipliers and adders/subtractors. This algorithm is implemented by FPGA using linear programming formulation and graph based algorithm. The use of MCM design leads to FIR filter design requiring less number of operators, having less delay, and consuming less power with respect to those include only adders/subtractors. However, due to the further use of a large number of algorithms, this becomes more complex.

IMPLEMENTATION

In this, FIR filter is implemented using direct form. FIR filters direct form uses a large number of operators which increases the area of filters. In this different parameters such as area, delay is described using Xilinx. The proposed CSE method can be explained using the example of a 5-tap FIR filter coefficient (Figs. 3-5).

RESULTS

Table 1 shows the design utization of summary using FIR direct form.

Table 2 shows the design utilization of summary using MCM and CSE techniques. It shows the number of slices, LUTs (Look up Tables), flip-flops and IOBs (Input-Output Blocks) are used. GCLK refers to

Global Clock.

The comparision table for MCM and CSE techniques with respect to delay and number of operators is shown in Table 3.

CONCLUSION

The digital communication industry has grown at a very fast rate. With the fast development of digital communication system, different techniques used to optimize the FIR filters. They are commonly used in DSP system. Due to these reasons, the demand for FIR filters has increased. In communication devices, such as telephones, in which more noise is present, thus a large number of filters to be used in such devices to remove the noise. Thus, the purpose of this paperwork is to analyze and design the optimization techniques to reduce the area of filter and delay. So that, signal can process at fast rate. A direct form, MCM, and CSE techniques for the optimization of FIR filters are presented. Compare to many techniques, these techniques are designed based on a simple structure and suitable for optimization of FIR filters. The proposed techniques are considered to achieve minimum area and delay of filter using less number of operators, so it can give fast rate to process a signal. Future enhancement of this paper is to design MCM architecture using 0-1 ILP CSE logic.

REFERENCES

1. Wanhammar L. DSP Integrated Circuits. New York: Academic Press; 1999.
2. Ye WB, Yu YJ. Bit-level multiplierless FIR filter optimization incorporating sparse filter technique. IEEE Trans Circ Syst I 2014;61(11):3206-15.
3. Aksoy L, Costa E, Flores P, Monteiro J. Multiple tunable constant multiplications: Algorithms and applications. In: Proceeding IEEE ISCAS, November; 2012. p. 473-9.
4. Soderstrand MA, Johnson LG, Arichanthiran H, Hoque MD, Elangovan R. Reducing hardware requirement in FIR filter design. In: Proceeding IEEE ISCAS; 2000. p. 3275-8.
5. Vijay S, Vinod AP, Lai EM. A greedy common sub expression elimination algorithm for implementing FIR filters. In: Proceeding IEEE ISCAS, May; 2007. p. 3451-4.
6. Mahesh R, Vinod AP. A new common sub expression elimination algorithm for realizing low complexity higher order digital filters. IEEE Trans Comput Aided Des Integer Circ Syst 2008;27(2):217-9.
7. Vijay S, Vinod AP. A Greedy Common Sub Expression Elimination Algorithm for Implementing FIR Filters. IEEE 1-4244-0925-7/07. 2007.
8. Aksoy L, da Costa E, Flores P, Monteiro J. Exact and approximate algorithms for the optimization of area and delay in multiple constant multiplications. IEEE Trans Comput Aided Des Integr Circ Syst 2008;27(6):1013-26.
9. Aksoy L, Lazzari C, Costa E, Flores P, Monteiro J. Design of digit-serial FIR filters: Algorithms, architectures, and a CAD tool. IEEE TransVery Large Scale Integr VLSI Syst 2013;21(3):498-511.
10. Al-Hasani F, Hayes MP, Smith AB. A common sub expression elimination tree algorithm. IEEE Trans Circ Syst I 2013;60(9):2389-400.
11. Aksoy L, Flores P, Monteiro J. Efficient design of FIR filters using hybrid multiple constant multiplications on FPGA. In: Proceeding IEEE ISCAS; 2014.