Global Journal of Researches in Engineering: F
Electrical and Electronics Engineering
Volume 14 Issue 9 Version 1.0 Year 2014
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 2249-4596 & Print ISSN: 0975-5861

An Efficient Implementation of Digit FIR Filters using Memory based Realization

By Maloth Santhoshi, Mrs. E. P. Vanetha & P. Srikanth
Vardhaman Colloge of Engineering, India

Abstract- The main contribution of this paper is an exact common sub expression elimination algorithm for the optimum sharing of partial terms in multiple constant multiplications (MCMs).Although many efficient high-level algorithms have been proposed for the realization of Multiple Constant Multiplications (MCM) using the fewest number of addition and subtraction operations, they do not consider the low-level implementation issues that directly affect the area, delay, and power dissipation of the MCM design. It is found that the proposed LUT-based multiplier involves comparable area and time complexity for a word size of 8 bits, but for higher word sizes, it involves significantly less area and less multiplication time than the canonical-signed-digit (CSD)-based multipliers we have proposed the anti symmetric product coding (APC) and odd-multiple-storage (OMS) techniques for lookup-table (LUT) design for memory-based multipliers to be used in digital signal processing applications. It was observed that the proposed algorithm obtains better solutions in terms of area than the algorithms designed for the MCM problem and the optimization of area problem in a digit-serial MCM operation at gate-level.

Keywords: *digital signal processing (DSP), finite impulse response (FIR) filter, multiple constant multiplication, lut-based computing, VLSI design.*

GJRE-F Classification : *FOR Code: 090699*

Strictly as per the compliance and regulations of :

An Efficient Implementation of Digit FIR Filters using Memory based Realization

Maloth Santhoshi [α], Mrs. E. P. Vanetha [σ] & P. Srikanth [ρ]

Abstract- The main contribution of this paper is an exact common sub expression elimination algorithm for the optimum sharing of partial terms in multiple constant multiplications (MCMs).Although many efficient high-level algorithms have been proposed for the realization of Multiple Constant Multiplications (MCM) using the fewest number of addition and subtraction operations, they do not consider the low-level implementation issues that directly affect the area, delay, and power dissipation of the MCM design. It is found that the proposed LUT-based multiplier involves comparable area and time complexity for a word size of 8 bits, but for higher word sizes, it involves significantly less area and less multiplication time than the canonical-signed-digit (CSD)-based multipliers we have proposed the anti symmetric product coding (APC) and odd-multiple-storage (OMS) techniques for lookup-table (LUT) design for memory-based multipliers to be used in digital signal processing applications. It was observed that the proposed algorithm obtains better solutions in terms of area than the algorithms designed for the MCM problem and the optimization of area problem in a digit-serial MCM operation at gate-level.

Keywords: digital signal processing (DSP), finite impulse response (FIR) filter, multiple constant multiplication, lut-based computing, VLSI design.

I. Introduction

There are many hand-held products that include digital signal processing (DSP), for example, cellular phones and hearing aids. For this type of portable equipment a long battery life time and low battery weight is desirable.

Author α: Vardhaman Colloge of Engineering, Hyderabad India. e-mail: maloth.santhoshi@gmail.com
Author σ: Vardhaman Colloge of Engineering, Hyderabad India. e-mail: vanetha.ece4@gmail.com
Author ρ: e-mail: parshasrikanth5@gmail.com

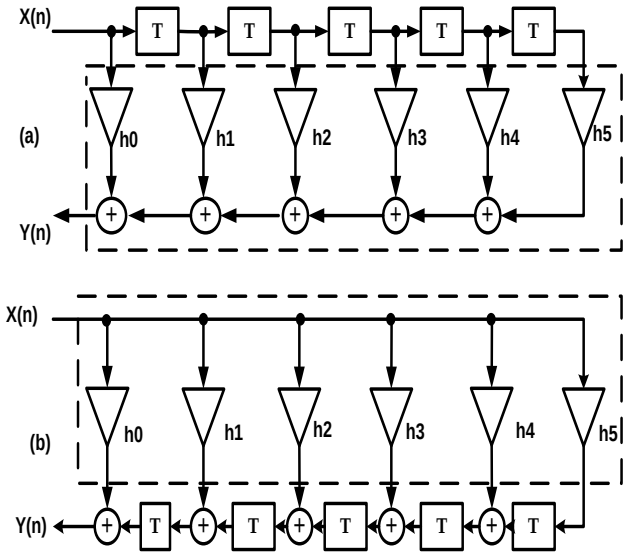

Fig. 1. (a) Direct and (b) Transposed direct form

To obtain this the circuit must have low power consumption. The main issue in this thesis is to minimize the energy consumption per operation for the arithmetic parts of DSP circuits, such as digital filters. More specific, the focus will be on single and multiple-constant multiplication using serial arithmetic.

a) Fir Filter Design

If the impulse response becomes zero after a finite number of samples it is a finite-length impulse response (FIR) filter.

For a given specification the filter order, N, is usually much higher for an FIR filter than for an IIR filter. However, FIR filters can be guaranteed to be stable and to have a linear phase response, which corresponds to constant group delay.

Different realizations of a fifth-order (five tap) FIR filter. (a) Direct form and (b) transposed direct form. It is not recommended to use recursive algorithms to realize FIR filters because of stability problems. Hence, here all coefficients bk in (1.1) is assumed to be zero. If an impulse is applied at the input each output sample will be equal to the corresponding coefficient ak, i.e., the impulse response is the same as the coefficients. The transfer function of an Nth-order FIR filter can then be written as

$$H(z) = \sum_{k=n}^{N} h(k)\, z^{-k}$$

A direct realization for N = 5 is shown in Fig. 1(a). This filter structure is referred to as a direct form FIR filter. If the signal flow graph is transposed the filter structure in Fig. 1.1 (b) is obtained, referred to as transposed direct form [2]. The dashed boxes in Figs. 1 (a) and (b) mark a sum-of-product block and a multiplier block, respectively. In both cases, the part that is not included in the dashed box is referred to as the delay section and the adders in Fig.1 (b) are called structural adders.

b) Bit-Serial Multiplier

In bit-serial arithmetic, the numbers are commonly processed with the least significant bit first. In bit-serial addition the two's complement numbers are added sequentially bit-by-bit by a full adder controlled by the clock clk as shown in Fig. 2.a. The full adder produces a sum bit and a carry bit. The carry has to be saved by the D flip-flop in order to be added to the input words on the next higher significant level in the next clock cycle. At the start of the computation, the input carry bit should be cleared, which is controlled by the signal. This circuit is called Carry-Save Adder (CSA) since there is no carry propagation. Hence the circuit can be used at high clock rates compared to parallel adders. The sum will be available after the propagation time of the full adder t add 0, hence this adder is of latency model order 0. An adder with LM 1 is realized by pipelining with a D flip-flop at the sum output shown in Fig.2.b. The minimum clock period is determined by the delay of the full adder and the propagation delay of the D flip-flop, t add 1.

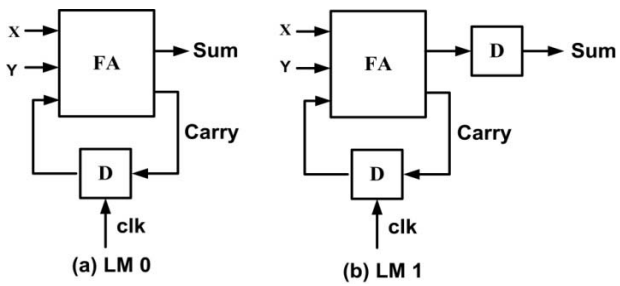

Fig. 2 : Bit-serial adders with LM 0 and LM 1

The algorithms designed for the MCM problem can be categorized in two classes: common sub expression elimination (CSE) algorithms [1]–[4] and graph-based (GB) techniques [2]–[4]. The CSE algorithms initially extract all possible sub expressions from the representations of the constants when they are defined under binary, canonical signed digit (CSD) [7], or minimal signed digit (MSD) [8]. Then, they find the "best" sub expression, generally the most common, to be shared among the constant multiplications. The GB methods are not limited to any particular number representation and consider a larger number of alternative implementations of a constant, yielding better solutions than the CSE algorithms. There is a disadvantage by implementing these algorithm the circuit complicity increases.

In order to avoid costly multipliers, most prior hardware implementations of digital FIR filters can be divided into two categories: multiplier less based and memory based. Multiplier less-based designs realize MCM with shift-and add operations and share the common sub operations using canonical signed digit (CSD) recoding and common sub expression elimination (CSE) to minimize the Adder cost of MCM.

In [8] and [9], more area savings are achieved by jointly considering the optimization of coefficient quantization and CSE. Most multiplier less MCM-based FIR filter designs use the transposed structure to allow for cross-coefficient sharing and tend to be faster, particularly when the filter order is large. However, the area of delay elements is larger compared with that of the direct form due to the range expansion of the constant multiplications and the subsequent additions in the SAs. Memory-based FIR designs consist of two types of approaches: lookup table (LUT) methods and distributed arithmetic (DA) methods [4]–[5]. The LUT-based design stores in ROMs odd multiples of the input signal to realize the constant multiplications in MCM [11].The DA-based approaches recursively accumulate the bit-level partial results for the inner product computation in FIR filtering [4], [5].

The remainder of the paper is organized as follows. In Section II, we have presented the proposed design of Architecture. In Section III, We have described the modules used in the proposed design for LUT Based multiplier in Section IV, Simulation results and FIR filter are evaluated and compared Conclusions are presented in Section V.

II. Proposed Architecture

A conventional lookup-table (LUT)-based multiplier is shown in Fig.4, where A is a fixed coefficient, and X is an input word to be multiplied with A. Assuming X to be a positive binary number of word length L, there can be 2^L possible values of X , and accordingly, there can be 2^L possible values of product $C = A \cdot X$. Therefore, for memory-based Multiplication, an LUT of 2L words, consisting of precomputed product values corresponding to all possible values of X,

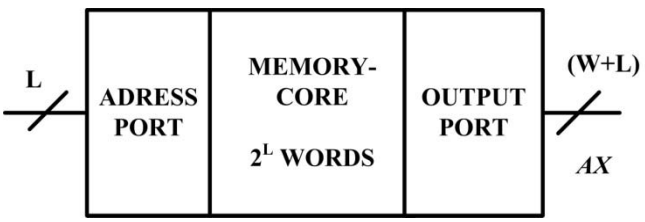

Fig. 3 : Conventional LUT-based multiplier

is The product word $A \cdot XI$ is stored at the location XI for $0 \leq XI \leq 2^L - 1$, such that if an L-bit binary value of XI is

used as the address for the LUT, then the corresponding product value $A \cdot XI$ is available as its output.

Memory based Computing: A class of dedicated systems, where the computational functions are performed by lookup tables (LUTs) instead of actual calculations .It close to human-like computing. Simple to design, and more regular compared with the multiply-accumulate structures. Involve less dynamic power consumption due to minimization of switching activities. It has potential for high-throughput and reduced latency implementation.

Memory-based computations examples Inner-product computation using the distributed arithmetic (DA) direct implementation of constant multiplications implementation of fixed and adaptive FIR filters and transforms well-suited for digital filtering and orthogonal. Transformations for digital signal processing and other applications: evaluation of trigonometric functions, sigmoid and other nonlinear function. The disadvantage of DA is filter with N coefficients the LUT has 2 N values. For higher order filter LUT size will increase, it required more memory space. The significant work on LUT optimization for memory-based multiplication, recently we have presented a new approach to LUT design, where only the odd multiples of the fixed coefficient are required to be stored [9], which we have referred to as the *odd-multiple-storage* (OMS).

The multiplication of any binary word X of size L, with a fixed coefficient A, instead of storing all the $2L$ possible values of $C = A \cdot X$, only $(2^L/2)$ words corresponding to the odd multiples of A may be stored in the LUT, while all the even multiples of A could be derived by left-shift operations of one of those odd multiples. Based on the above assumptions, the LUT for the multiplication of an L-bit input with a W–bit coefficient could be designed by the Following strategy.

a) A memory unit of $[(2^L/2) +1]$ words of (W+L) –bit width is used to store the product values, where the first $(2^L/2)$ words are odd multiples of A, and the last word is zero.

b) A barrel shifter for producing a maximum of (L-1) left shifts is used to derive all the even multiples of A.

Table 1 : Look up table to multiples a 4-bit word word*with a constant

Address Word, X	Product word	Address Word, X	Product word
0000	0	1000	8A
0001	A	1001	9A
0010	2A	1010	10A
0011	3A	1011	11A
0100	4A	1100	12A
0101	5A	1101	13A
0110	6A	1110	14A
0111	7A	1111	15A

c) The L-bit input word is mapped to the (L-1) bit address of the LUT by an address encoder, and control bits for the barrel shifter are derived by a control circuit.

III. Modules used in the Proposed Design for lut based Multiplier

a) Address Generator and Control Circuit

The address generation and control circuit used to produce the address d0d1d2d3. This address is given as the input to LUT component. The address generation circuit is generally used in conjunction with the control circuit which is used to produce the control signals s0and s1. The control signals are used in the subsequent blocks as can be seen from Fig. 3., for the multiplication of any binary word of size L, with a fixed coefficient A, instead of storing all the 2L possible values of C=A*X, only (2L/2) words corresponding to the odd multiples of A may be stored in the LUT, while all the even multiples of A could be derived by left-shift operations of one of those odd multiples. This can be achieved with of one of the modules i.e. Barrel Shifter.

b) Barrel Shifter Module

In Table 1, at eight memory locations, the eight odd multiples, $A \times (2i + 1)$, are stored. The even multiples 2A, 4A and 8A are derived by left-shift operations of A. 6A and 12A are derived by left shifting operation of 3A. 10A and 14A are derived by left shifting 5A and 7A, respectively. Three left-shift operations can be produced by a barrel shifter to derive all the multiplier.

c) LUT Component Module

The LUT component for multiplication of a 4-bit unsigned input consists of a set of eight odd multiple values of a fixed coefficient, say 4, i.e. 4, 12, 20, 28, and so on. Also, for 8-bit signed input, the LUT component has the above values as well as another set of odd multiple stored values such as 196, 200, and so on till 256. As a result, LUT size is considerably reduced. The design of hardware efficient and high throughput FIR filter has become much more demanding. In conventional design, however, the multipliers in the structure require a large portion of chip-area, and accordingly, the delay of the structure is large due to the large time required in multiplication. Multiplier less memory-based techniques [3]-[10] has been widely

used in many applications, in recent years, for their high throughput processing and cost-effective structures.

Memory based multiplier implementation of multiplications can be performed by using CAD tools computing. In this paper, a new approach to LUT implementation for memory-based multiplication was proposed [11]. Though the approach proposed in [11] is efficient in implementation, the approach can be improved further. For example, it is noticed that there was an address encoder and a control circuit in the architecture [11, Fig. 5], which may increase the chip area. Therefore, if we could improve the memory-based technique, we may get a hardware efficient structure for implementation. In this paper, we aim at presenting a new memory-based technique to replace the conventional direct LUT for hardware-efficient implementation. In FIR filtering, one of the convolving sequences is the input samples while the other is the fixed coefficients of the filter. This behavior of the FIR filter makes it possible for memory-based multiplication realization. It yields faster output compared with the multiplier-based designs because it stores the pre-computed results in the memory units, which can be read out and accumulated to obtain the result. This memory based technique has two major features. *First,* it suits well for any number of input word lengths. And any N number of coefficients can be multiplied and stored in a specified address location and it shows shifting operation of given variables. *Second,* the whole structure can be decomposed into a number of small units, which can be extended further to obtain a high-throughput structure for FIR filter implementation. Multiplication of an 8-bit input with a W-bit fixed coefficient can be performed through a pair of multiplications using a dual-port memory of 8 words (or two single-port memory units) along with a pair of decoders, encoders, NOR cells and barrel shifters as shown in Fig. 5. The shift-adder performs left-shift operation of the output of the barrel-shifter corresponding to more significant half of input by eight bit-locations, and adds that to the output of the other barrel-shifter.

The proposed system technique is it multiples the N no. of variables with N no. of coefficient and reduces the memory size and stores it in a specified location by performing the conditional operations of the given code instruction. For an 8-bit input variable in this system initially we have to initiate the input 8-bit input variable and 4-bit filter coefficient. The input variable and filter coefficient multiples and gives the product output. The product output stored address location and shifting operation can be seen in the simulation results.

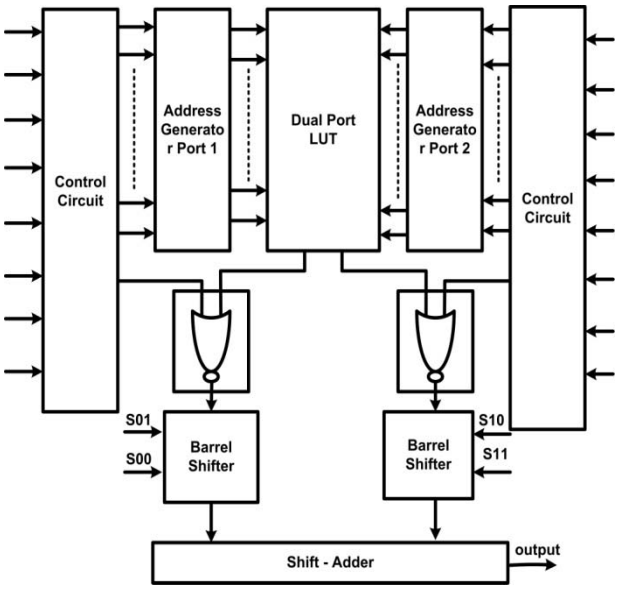

Fig. 5 : The shift-adder performs left-shift Operation of the output of the barrel-shifter

The hardware operation is reduces the partial products, so the area is reduces and the speed or performance of the designs increase Similar for a 16-bit variable function it performs same operation. Additionally by using the clock performs of the circuit is having potential for high-throughput and reduced latency implementation Low complexity in the circuit design. A modified hardware-efficient approach for Memory based multiplication is proposed. The proposed approach is less hardware complexity than the existing memory less-based design for multiplication. Then the proposed approach is applied in the FIR filter. Thus it can be readily used as an IP core in a number of environments, especially for those high-order filters. Further work may concern about the more efficient design for multiplication.

IV. Simulations Results

Fig. 6 : Simulation result of 8 bit LUT based FIR filter

Fig. 7 : Simulation result of 16-bit LUT

Table 2 : Summary report of LUT

FIR filter design	No. of slices utilization out of 2448	No. of 4 input LUT out of 4896	Number of bonded IOBs out of 92
6-bit LUT based	6%	6	28
Proposed 8-bit memory based LUT	3%	3	28
16- bit memory based LUT	42%	36	39

V. Conclutions

In this paper, we have shown the possibility of using LUT based multipliers to implement the constant multiplication for DSP applications. It was observed that the proposed algorithm obtains better solutions in terms of area than the algorithms designed for the MCM problem and the optimization of area problem in a digit-serial MCM operation at gate-level. It was also shown that the realization of digit-serial FIR filters under the shift-adds architectures yields significant area and power reductions when compared to those whose multiplier blocks are implemented using digit-serial constant multipliers.

References Références Referencias

1. A Novel Architecture of LUT Design Optimization for DSP Application.
2. Hardware Efficient Approach for Memoryless-Based Multiplication and Its Application to FIR Filter "Jianjun He Institute of Information Science and Engineering, Central South University, Changsha, P.R. China Jiafeng XieInstitute of Information Science and Engineering, Central South University, Changsha, P.R. China Email: jftsa@yahoo.com.
3. L. Aksoy and C. Lazzari a "Design of Digit-Serial FIR Filters: Algorithms, Architectures, and a CAD Tool" re with INESC-ID, Lisbon 1000-029, Portugal (e-mail:levent@algos.inesc id.pt; lazzari@algos.-inesc-id.pt).
4. Low-Cost FIR Filter Designs Based on Faithfully Rounded Truncated Multiple Constant Multiplication Accumulation Shen-Fu Hsiao, Jun-Hong Zhang Jian, and Ming-Chih Chen.
5. L. Aksoy, E. Costa, P. Flores, and J. Monteiro, "Optimization of area in digital FIR filters using gate-level metrics," in Proc. DAC, 2007, pp.420–423.
6. L. Aksoy, C. Lazzari, E. Costa, P. Flores, and J. Monteiro, "Optimization of area in digit-serial multiple constant multiplications at gate-level," in Proc. ISCAS, 2011, pp. 2737–2740.
7. L. Aksoy, C. Lazzari, E. Costa, P. Flores, and J. Monteiro, "Efficient shift-adds design of digit-serial multiple constant multiplications," in Proc. Great Lakes Symp. VLSI, 2011, pp. 61–66.
8. L. Aksoy, E. Costa, P. Flores, and J. Monteiro, "Exact and approximate algorithms for the optimization of area and delay in multiple constant multiplications," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 6, pp. 1013–1026, Jun. 2008.
9. LUT Optimization for Memory-Based Computation Pramod Kumar Meher, *Senior Member, IEEE* Manuscript received June 25, 2009; revised November 14, 2009. Current version published April 21, 2010.
10. LUT Optimization Using Combined APC-OMS Technique For Memory-Based Computation Mrs.R.RAMYA#1 Mrs.S.SUDHA IJCAES - V-III – SI – NCIC'13 – AUG-2013.
11. Optimization of Area in Digit-Serial Multiple *Constant Multiplications at Gate-Level the Portuguese Foundation for Science and Technology* (FCT) research project PTDC/EIA-EIA/103532/2008.

Global Journal of Researches in Engineering (F) Volume XIV Issue IX Version I Year 2014
12

This page is intentionally left blank