ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Monthly Peer Reviewed Open Access International e-Journal

An Efficient Design of Memory-Based Realization of FIR Digital Filter

Ch.V.Ramesh Babu
M.Tech Student,
Rise Gandhi Group Of Institutions.

S. Chandrasekhar
Associate Professor,
Rise Gandhi Group Of Institutions.

Abstract:

In the last two decades, many efficient algorithms and architectures have been introduced for the design of low complexity bit-parallel multiple constant multiplications (MCM) operation which dominates the complexity of many digital signal processing systems. On the other hand, little attention has been given to the digit-serial MCM design that offers alternative low complexity MCM operations albeit at the cost of an increased delay.

In this paper, we address the problem of optimizing the gate-level area in digit-serial MCM designs and introduce high level synthesis algorithms, design architectures, and a computer aided design tool. Experimental results show the efficiency of the proposed optimization algorithms and of the digit-serial MCM architectures in the design of digit-serial MCM operations and finite impulse response filters.

Index Terms:

0–1 integer linear programming (ILP), digit-serial arithmetic, finite impulse response (FIR) filters, gate-level area optimization, multiple constant multiplications.

I. INTRODUCTION:

FINITE impulse response (FIR) filters are of great importance in digital signal processing (DSP) systems since their characteristics in linear-phase and feed-forward implementations make them very useful for building stable high- performance filters. The direct and transposed-form FIR filter implementations are illustrated in Fig. 1(a) and (b), respectively.

Although both architectures have similar complexity in hardware, the transposed form is generally preferred because of its higher performance and power efficiency [1].

The multiplier block of the digital FIR ilter in its transposed form [Fig. 1(b)], where the multiplication of filter coefficients with the filter input is realized, has significant impact on the complexity and performance of the design because a large number of constant multiplications are required.

Thisis generally known as the multiple constant multiplications (MCM) operation and is also a central operation and performance bottleneck in many other DSP systems such as fast Fourier transforms, discrete cosine transforms (DCTs), and error-correcting codes.

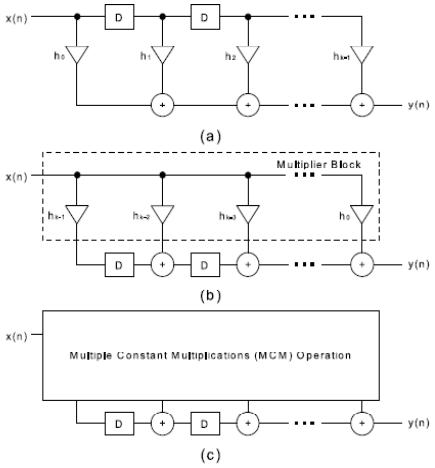


Fig. 1. FIR filter implementations. (a) Direct form. (b) Transposed form

Although area-, delay-, and power-efficient multiplier architectures, such as Wallace [2] and modified Booth [3] multipliers, have been proposed, the full flexibility of a multiplier is not necessary for the constant multiplications, since filter coefficients are fixed and determined beforehand by the DSP algorithms [4].

Hence, the multiplication of filter coefficients with the input data is generally implemented under a shift adds architecture [5], where each constant multiplication is realized using addition/subtraction and shift operations in an MCM operation [Fig. 1(c)].

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Monthly Peer Reviewed Open Access International e-Journal

For the shift-adds implementation of constant multiplications, a straightforward method, generally known as digitbased recoding [6], initially defines the constants in binary. Then, for each "1" in the binary representation of the constant, according to its bit position, it shifts the variable and adds up the shifted variables to obtain the result. As a simple example, consider the constant multiplications 29x and 43x. Their decompositions in binary are listed as follows:

$$29x = (11101)_{\text{bin}}x = x \ll 4 + x \ll 3 + x \ll 2 + x$$
$$43x = (101011)_{\text{bin}}x = x \ll 5 + x \ll 3 + x \ll 1 + x$$

which requires six addition operations as illustrated in Fig. 2(a).

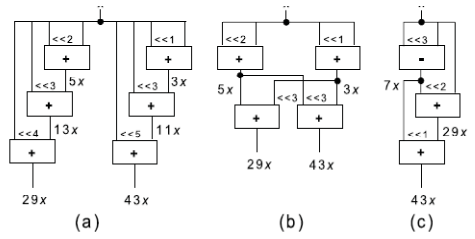


Fig. 2. Shift-adds implementations of 29x and 43x. (a) Without partial product sharing [6] and with partial product sharing. (b) Exact CSE algorithm [9]. (c) Exact GB algorithm [12].

However, the digit-based recoding technique does not exploit the sharing of common partial products, which allows great reductions in the number of operations and, consequently, in area and power dissipation of the MCM design at the gate level. Hence, the fundamental optimization problem, called the MCM problem, is defined as finding the minimum number of addition and subtraction operations that implement the constant multiplications. Note that, in bit-parallel design of constant multiplications, shifts can be realized using only wires in hardware without representing any area cost.

The algorithms designed for the MCM problem can be categorized in two classes: common subexpression elimination (CSE) algorithms [7]–[9] and graph-based (GB) techniques [10]–[12]. 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" subexpression, 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 aconstant, yielding better solutions than the CSE algorithms, as shown in [11] and [12].

Returning to our example in Fig. 2, the exact CSE algorithm of [9] gives a solution with four operations by finding the most common partial products 3x = (11)binx and 5x = (101)binx when constants are defined under binary, as illustrated in Fig. 2(b).

On the other hand, the exact GB algorithm [12] finds a solution with the minimum number of operations by sharing the common partial product 7x in both multiplications, as shown in Fig. 2(c). Note that the partial product 7x =(111)binx cannot be extracted from the binary representationof 43x in the exact CSE algorithm [9].

However, all these algorithms assume that the input data x is processed in parallel. On the other hand, in digit-serial arithmetic, the data words are divided into digit sets, consisting of d bits, that are processed one at a time [13]. Since digit serial operators occupy less area and are independent of the data wordlength, digit-serial architectures offer alternative lowcomplexity designs when compared to bit-parallel architectures.

However, the shifts require the use of D flip-flops, asopposed to the bit-parallel MCM design where they are free in terms of hardware. Hence, the high-level algorithms should take into account the sharing of shift operations as well as the sharing of addition/subtraction operations in digit-serial MCM design.

Furthermore, finding the minimum number of operations realizing an MCM operation does not always yield an MCM design with optimal area at the gate level [14]. Hence, the high-level algorithms should consider the implementation cost of each digit-serial operation at the gate level.

II. LUT DESIGN FOR MEMORY-BASED MULTIPLICATION:

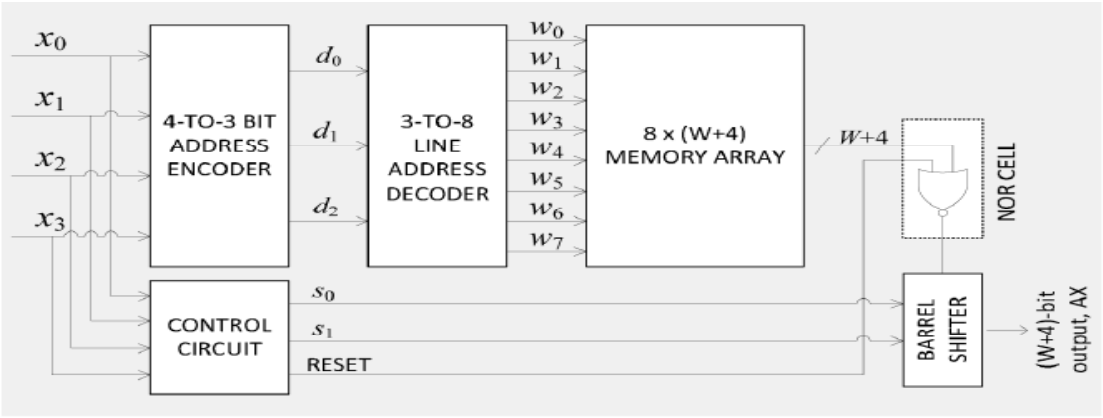
The basic principle of memory-based multiplication is depicted in Fig. 2. Let be a fixed coefficient and be an input word to be multiplied with . If we assume to be an unsigned binary number of word-length , there can be possible values of , and accordingly, there can be possible values of product . Therefore, for the conventional implementation of memory-based multiplication [24], a memory unit of words is required to be used as look-up-table consisting of pre-computed product

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Monthly Peer Reviewed Open Access International e-Journal

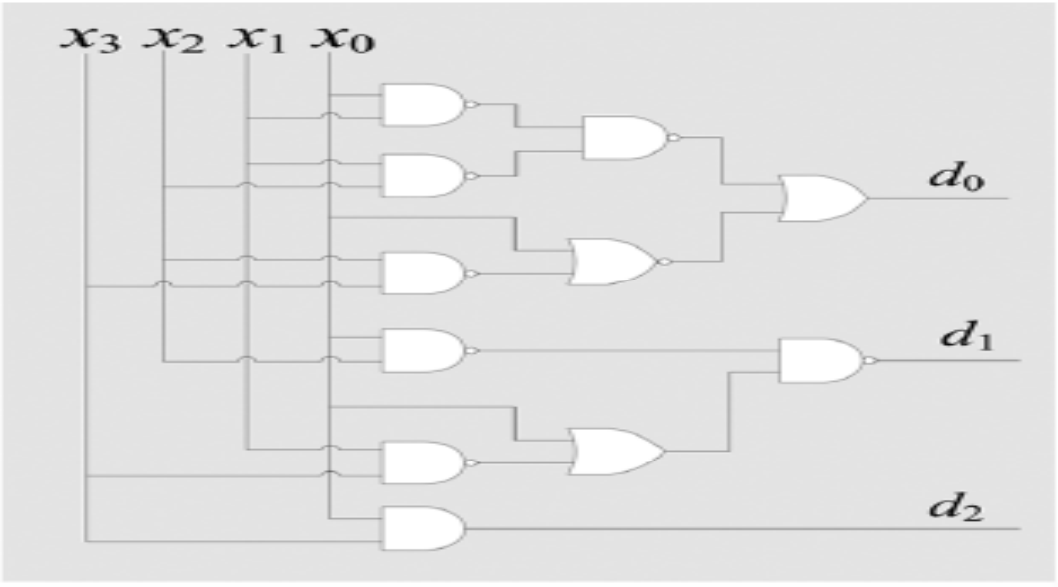
values corresponding to all possible values of The product-word , for , is stored at the memory location whose address is the same as the binary value of , such that if -bit binary value of is used as address for the memory-unit, then the corresponding product value is read-out from the memory.Although possible values of correspond to possible values of , recently we have shown that only words corresponding to the odd multiples of may only be stored in the LUT [30]. One of the possible product words is zero, while all the rest are even multiples of which could be derived by left-shift operations of one of the odd multiples of . We illustrate this in Table I for . At eight memory locations, eight odd multiples are stored are derived by left-shift operations of . Similarly, and are derived by left-shifting , while and are derived by left-shifting and , respectively. The address corresponds to , which can be obtained by resetting the LUT output. For an input multiplicand of word-size similarly, only odd multiple values need to be stored .

in the memory-core of the LUT, while the other non-zero values could be derived by left-shift operations of the stored values.Based on the above, an LUT for the multiplication of an -bit input with -bit coefficient is designed by the following strategy:

- A memory-unit of words of -bit width is used to store all the odd multiples of
- A barrel-shifter for producing a maximum of leftshifts is used to derive all the even multiples of .
- The -bit inputword is mapped to -bit LUT-address by an encoder.
- The control-bits for the barrel-shifter are derived by a control-circuit to perform the necessary shifts of the LUToutput. Besides, a RESET signal is generated by the same control circuit to reset the LUT output when .



(a)



(b)

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Monthly Peer Reviewed Open Access International e-Journal

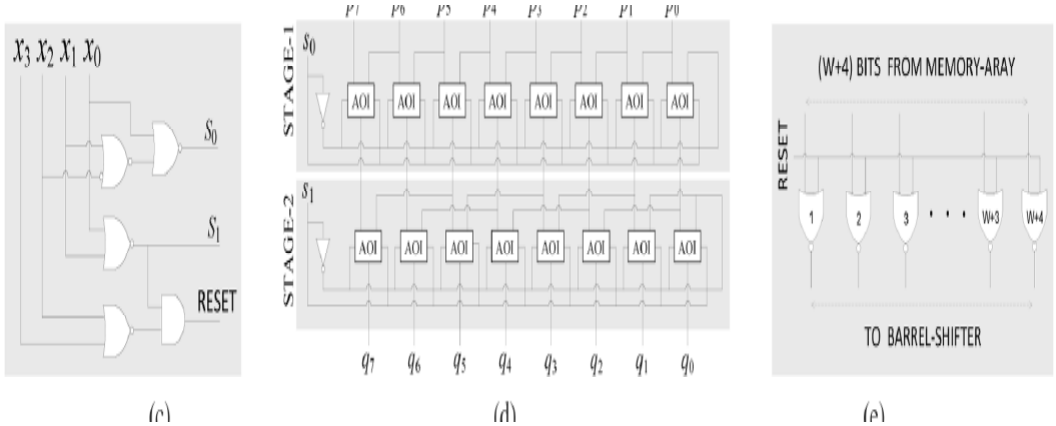


Fig. 3. Proposed LUT design for multiplication of -bit fixed coefficient, and 4-bit input operand, . The proposed -based multiplier. (b)The 4-to-3 bits input encoder. (c) Control circuit. (d) Two-stage logarithmic barrel-shifter e) Structure of the NOR-cell.

By odd-multiple-storage scheme, for address-length 4, the LUT size is reduced to half by using a two-stage logarithmicbarrel-shifter and number of NOR gates, where is the word-length of the fixed multiplying coefficients. Three memory-based structures having unit throughput rate are designed further for the implementation of FIR filter. One of the structures is based on DA principle, and the other two are based on LUT-based multiplier using the conventional and the proposed LUT designs.

All the structures are found to have the same or nearly the same cycle periods, which depend on the implementation of adders, the word-length and the filter order. The conventional LUT-multiplier-based filter has nearly the same memory requirement and the same number of adders, and less number of input registers than the DA-based design at the cost of higher adder-widths.

REFERENCES:

[1] J. G. Proakis and D. G. Manolakis, Digital Signal Processing: Principles,Algorithms and Applications. Upper Saddle River, NJ: Prentice-Hall, 1996.

[2] G. Mirchandani, R. L. Zinser Jr., and J. B. Evans, "A new adaptive noise cancellation scheme in the presence of crosstalk [speech signals],"IEEE Trans. Circuits Syst. II, Analog. Digit. Signal Process.,vol. 39, no. 10, pp. 681–694, Oct. 1995.

[3] D. Xu and J. Chiu, "Design of a high-order FIR digital filtering and variable gain ranging seismic data acquisition system," in Proc. IEEE Southeastcon'93, Apr. 1993, p. 6.

[4] K. K. Parhi, VLSI Digital Signal Procesing Systems: Design and Implementation.New York: Wiley, 1999.

[5] H. H. Kha, H. D. Tuan, B.-N. Vo, and T. Q. Nguyen, "Symmetric orthogonal complex-valued filter bank design by semidefinite programming," IEEE Trans. Signal Process., vol. 55, no. 9, pp. 4405–4414, Sep 2007.

[6] H. H. Dam, A. Cantoni, K. L. Teo, and S. Nordholm, "FIR variable digital filter with signed power-of-two coefficients," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 54, no. 6, pp. 1348–1357, Jun. 2007.

[7] R. Mahesh and A. P. Vinod, "A new common subexpression elimination algorithm for realizing low-complexity higher order digital filters," IEEE Trans. Computer-Aided Ded. Integr. Circuits Syst., vol. 27, no. 2, pp. 217–229, Feb. 2008.

[8] International Technology Roadmap for Semiconductors, [Online]. Available: http://public.itrs.net/

[9] B. Prince, "Trends in scaled and nanotechnology memories," in Proc. IEEE Conf. Custom Integr. Circuits, Nov. 2005, pp. 7–.

[10] K. Itoh, S. Kimura, and T. Sakata, "VLSI memory technology: Current status and future trends," in Proc. 25th Eur. Solid-State Circuits Conference,{ ESSCIRC'99, Sep. 1999, pp. 3–10.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Monthly Peer Reviewed Open Access International e-Journal

[11] T. Furuyama, "Trends and challenges of large scale embedded memories," in Proc. IEEE Conf. Custom Integrated Circuits, Oct. 2004, pp. 449–456.

[12] D. G. Elliott, M. Stumm, W. M. Snelgrove, C. Cojocaru, and R. Mckenzie, "Computational RAM: Implementing processors in memory," IEEE Trans. Design Test Comput., vol. 16, no. 1, pp. 32–41, Jan. 1999.

[13] H.-R. Lee, C.-W. Jen, and C.-M. Liu, "On the design automation of the memory-based VLSI architectures for FIR filters," IEEE Trans. Consum. Electron., vol. 39, no. 3, pp. 619–629, Aug. 1993.

[14] S. A. White, "Applications of the distributed arithmetic to digital signal processing:Atutorial review," IEEE ASSP Mag., vol. 6, no. 3, pp. 5–19, Jul. 1989.

[15] M. Mehendale, S. D. Sherlekar, and G.Venkatesh, "Area-delay tradeoff in distributed arithmetic based implementation of FIR filters," in Proc.10th Int. Conf. VLSI Design, Jan. 1997, pp. 124–129.

[16] H. Yoo and D. V. Anderson, "Hardware-efficient distributed arithmetic architecture for high-order digital filters," in Proc. IEEE Int. Conf.Acoustics, Speech, Signal Processing, (ICASSP'05), Mar. 2005, vol. 5, pp. v/125–v/128.

[17] D. J. Allred, H. Yoo, V. Krishnan, W. Huang, and D. V. Anderson, "LMS adaptive filters using distributed arithmetic for high throughput," IEEE Trans. Circuits Systems I, Reg. Papers, vol. 52, no. 7, pp. 1327–1337, Jul. 2005.