

Design of Digit-Serial FIR Filters Using Mag Adder Graph Multiplier

K. SWAPNA¹, M. SUNIL BABU²

¹PG Scholar, Electronics and Communication Engineering, vasireddy venkatadri institute of technology, AP, India
²Assosiatet Professor, Electronics and Communication Engineering, vasireddy venkatadri institute of technology, AP, India Email: swapna1241@gmail.com, sunil.babu.m@gmail.com

Abstract—Finite Impulse Response (FIR) filters are widely applied in multistandard wireless communications. A novel 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. In digit-serial MCM design that offers low complexity MCM operations that offers a low delay. In this previous design a MCM operations performed by CSE algorithm but it occupies large and delay area. In MCM design based Graph algorithm provides low area and delay. In this we proposed a MAG multiplier based on graph architecture for implementing low complexity higher order FIR filters.

Keywords: FIR-finite impulse response filters, CSD – Canonic Signed-Digit multiplier, CSE- common subexpression elimination algorithms, MSD – Minimum Signed-Digit multiplier, MAG-Minimum added Graph Multiplier.

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 filter 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. This is 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.

Although area-, delay-, and power-efficient multiplier architectures, such as Wallace [2] and modified Booth [3] multipliers, have been proposed, the full flexibility of 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)].

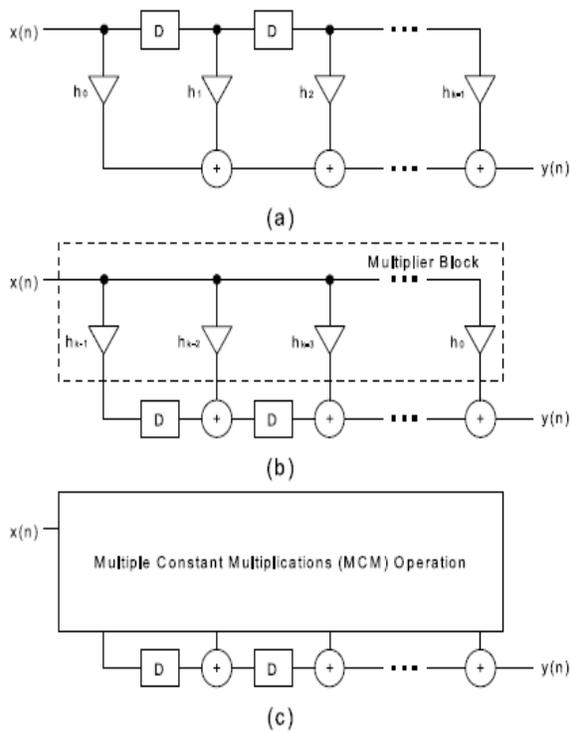


Fig. 1. FIR Filter implementations (a)Direct form. (b)Transposed form. (c)Transposed form with an MCM block

For the shift-adds implementation of constant multiplications, a straightforward method, generally known as digit based 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).

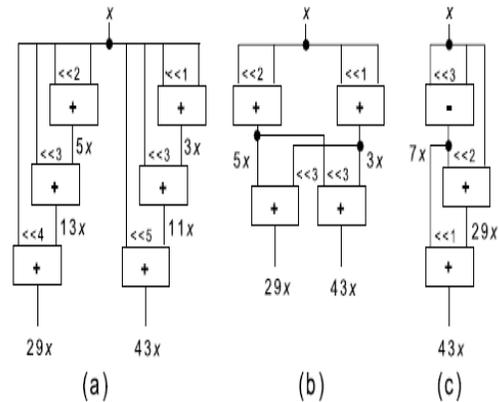


Fig.2.Shift-adds implementations of $29x$ and $43x$ (a) Without partial product sharing [6] and with partial product sharing. (b) Without CSE algorithm [9]. (c) Exact GB algorithm

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 subexpressions 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 a



constant, 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 $3x = (11)\text{bin}_x$ and $5x = (101)\text{bin}_x$ 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 $7x = (111)\text{bin}_x$ cannot be extracted from the binary representation of $43x$ 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 word length, digit-serial architectures offer alternative low complexity designs when compared to bit-parallel architectures. However, the shifts require the use of D flip-flops, as opposed 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.

In this paper, we initially determine the gate-level implementation costs of digit-serial addition, subtraction, and left shift operations used in the shift-adds design of digit-serial MCM operations. Then, we introduce the exact CSE algorithm [15] that

formalizes the gate-level area optimization problem as a 0–1 integer linear programming (ILP) problem when constants are defined under a particular number representation. We also present a new optimization model that reduces the 0–1 ILP problem size significantly and, consequently, the runtime of a generic 0–1 ILP solver. Since there are still instances which the exact CSE algorithm cannot handle, we describe the approximate GB algorithm [16] that iteratively finds the “best” partial product which leads to the optimal area in digit-serial MCM design at the gate level.

In this proposed different graph based multipliers types i.e. CSD – canonic signed-digit multiplier, MSD – minimum signed-digit multiplier, MAG – minimum adder graph multiplier, CSDAG – csd adder graph multiplier. They are used for low complexity, low power and low area applications.

The section II explains the complexity of serial constant multipliers. Section III explains graph based multipliers. Section IV explains results and analysis.

II.COMPLEXITY OF SERIAL CONSTANT MULTIPLIERS

In this chapter, the possibilities to minimize the complexity of bit-serial single-constant multipliers are investigated [57]. This is done in terms of the required number of building blocks, which includes adders and shifts. The multipliers are described using a graph representation. It is shown that a minimum set of graphs, required to obtain optimal results given certain restrictions, can be found. In the case of single-constant multipliers, the number of possible solutions can be limited because of the finite number of graph topologies. However, if a shift-and-add network realizing several coefficients is

required, a multiple-constant multiplication (MCM) problem is obtained.

Different heuristic algorithms can then be used to reduce the complexity, by utilizing the redundancy between the coefficients. Two algorithms suitable to achieve efficient realization of MCM using serial arithmetic are presented [56],[62],[66]. It is shown that the new algorithms reduce the total complexity significantly. Furthermore, we study the trade-offs in implementations of FIR filters using MCM and digit-serial arithmetic. Comparisons considering area, speed, and energy consumption, with respect to the digit-size, are performed [61],[67].

III. Graph Multipliers

In this section, different types of single-constant graph multipliers will be defined, with respect to constraints on adder cost and throughput. Furthermore, the possibilities to exclude some graphs from the search space are examined. The investigation covers all coefficients up to 4095 and all types of graph multipliers containing up to four adders. All possible graphs, using the representation discussed in Section 3.1, for adder costs from 1 to 4 are presented in Fig. 2.1 [24]. Note that although bit-serial arithmetic will be assumed for the multipliers, results considering adder and flip-flop costs are generally also valid for any digit-serial implementation. However, the numbers of registers that are required to perform pipelining depend on the digit-size. Furthermore, the cost difference between adders and shifts becomes higher for larger digit-sizes, since the number of full adders increases linearly while the number of flip-flops is constant. Hence, such trade-offs are mainly of interest for small digit-sizes.

3.1 Multiplier Types

Different multiplier types can be defined based on the requirements considering adder

cost, flip-flop cost, and pipelining. The types that will be discussed here are described in the following.

- **CSD – Canonic Signed-Digit multiplier**

Multiplier based on the CSD representation, as discussed in Section 4.1, with an adder cost equal to one less than the number of nonzero digits.

- **MSD – Minimum Signed-Digit multiplier**

Similar to the CSD multiplier and requires the same number of adders, but can in some cases decrease the flip-flop cost by using other MSD representations, which were discussed in Section 3.1.

- **MAG – Minimum Adder Graph multiplier**

Graph multiplier that is based on any of the topologies in Fig. 4.1 and, for any given coefficient, has the lowest possible adder cost.

Example

To describe the difference between the defined multiplier types, corresponding realizations of the coefficient 2813, which has the CSD representation 101010000101, are shown in Fig. 4(a). There are other possible solutions for all types except the CSD multiplier. However, note that the values corresponding to the nonzero digits in the CSD representation can be added in different orders, resulting in other structures. Since this may eliminate the pipeline feature, the basic structure used in Fig. 4 (b) will be assumed for CSD multipliers. The adder costs for the multipliers in figs. 4 (a), (b), (c), and are 4, 4, and 3 respectively.

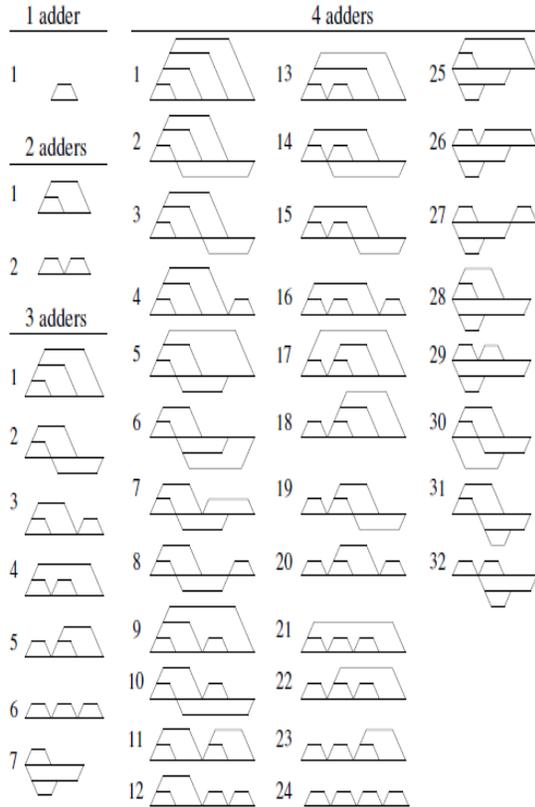


Figure.3. Possible graph topologies for an adder cost up to four.

This implies that it is possible to save either two shifts, or one adder and one Shift compared to the CSD multiplier.4 (b) and (c) with an extra cost of 0 and 1 register, respectively. Note that the flip-flop cost will include both shifts and pipelining registers, since both correspond to a single flip-flop in bit-serial arithmetic.

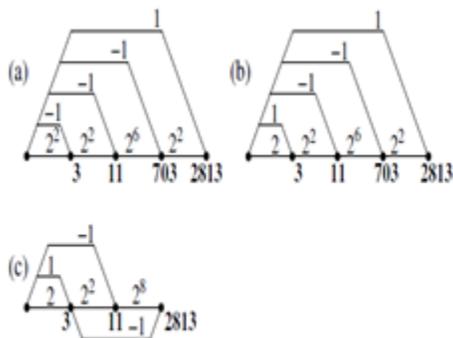


Figure.4. Different realizations of the coefficient 2813. (a) CSD, (b) MSD, (c) MA G

IV.RESULTS AND ANALYSIS

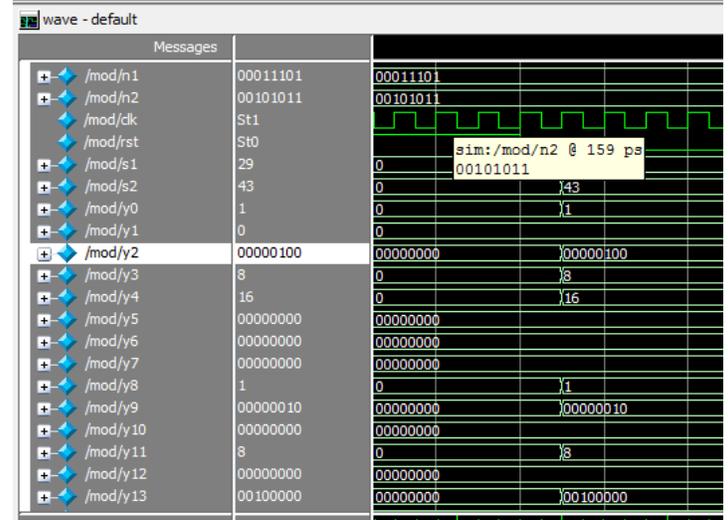


Fig.5.Waveform Results of GB Algorithm.

TABLE 1

The comparison Result of CSE and BE Algorithm

Algorithm name	Delay(ns)	Area(%)
CSE	2.780	33
GB	2.58	30

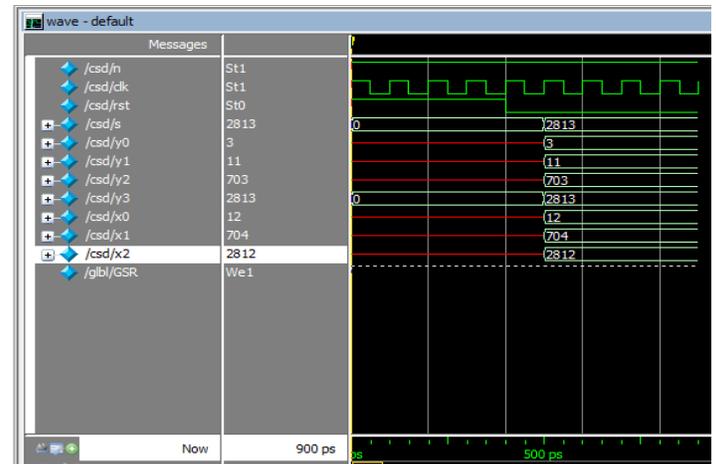


Fig.6. Waveform Results of CSDAG Algorithm

TABLE 2

The Comparison Result of CSD and MAG

Graph based Multipliers	Delay(ns)	Area (%)
CSD	2.58	24
MAG	2.58	19

The comparison tables and waveforms show the analysis of different algorithms and different graph based multipliers. In this BE algorithm shows an efficient results compare to CSE algorithms. Again analyses in graph based multipliers i.e. CSD, MSD, and MAG .the MAG shows good area in table 2.

V.CONCLUSION

The proposed new approach is MAG for implementing reconfigurable higher order filters with low complexity. The proposed MAG method make use of architecture with fixed number of multiplexers and the reduction in complexity is achieved by applying the graph based algorithm. The MAG architecture results in high speed filters and low area and thus low power filter implementations. The MAG also provides the flexibility of changing the filter coefficient word lengths dynamically. The proposed reconfigurable architectures can be easily modified to employ any graph based (GB) method, which results in architectures that offers good area and power reductions and speed improvement reconfigurable FIR filter implementations.

REFERENCES

- [1] L. Wanhammar, *DSP Integrated Circuits*. New York: Academic, 1999.
- [2] C. Wallace, "A suggestion for a fast multiplier," *IEEE Trans. Electron. Comput.*, vol. 13, no. 1, pp. 14–17, Feb. 1964.
- [3] W. Gallagher and E. Swartz lander, "High radix booth multipliers using reduced area adder trees," in *Proc. Asilomar Conf. Signals, Syst. Comput.*, vol. 1. Pacific Grove, CA, Oct.–Nov. 1994, pp. 545–549.
- [4] J. McClellan, T. Parks, and L. Rabiner, "A computer program for designing optimum FIR linear phase digital filters," *IEEE Trans. Audio Electroacoust.*, vol. 21, no. 6, pp. 506–526, Dec. 1973.
- [5] H. Nguyen and A. Chatterjee, "Number-splitting with shift-and-add decomposition for power and hardware optimization in linear DSP synthesis," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 8, no. 4, pp. 419–424, Aug. 2000.
- [6] M. Ercegovac and T. Lang, *Digital Arithmetic*. San Mateo, CA: Morgan Kaufmann, 2003.
- [7] R. Hartley, "Subexpression sharing in filters using canonic signed digit multipliers," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 43, no. 10, pp. 677–688, Oct. 1996.
- [8] I.-C. Park and H.-J. Kang, "Digital filter synthesis based on minimal signed digit representation," in *Proc. DAC*, 2001, pp. 468–473.
- [9] 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.

[10] A. Dempster and M. Macleod, “Use of minimum-adder multiplier blocks in FIR digital filters,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 42, no. 9, pp. 569–577, Sep. 1995.

[11] Y. Voronenko and M. Püschel, “Multiplierless multiple constant multiplication,” *ACM Trans. Algor.*, vol. 3, no. 2, pp. 1–39, May 2007.

[12] L. Aksoy, E. Gunes, and P. Flores, “Search algorithms for the multiple constant multiplications problem: Exact and approximate,” *J. Microprocess. Microsyst.*, vol. 34, no. 5, pp. 151–162, Aug. 2010.

[13] R. Hartley and K. Parhi, *Digit-Serial Computation*. Norwell, MA: Kluwer, 1995.

[14] 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.

[15] 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.

[16] 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.

[17] B. Parhami, *Computer Arithmetic: Algorithms and Hardware Designs*. New York: Oxford Univ. Press, 2000.

[18] K. Johansson, O. Gustafsson, A. G. Dempster, and L. Wanhammar, “Algorithm to reduce the number of shifts and additions in multiplier blocks using serial arithmetic,”

in *Proc. IEEE Mediterranean Electrotechnical Conf.*, Dubrovnik, Croatia, May 12–15, 2004, vol. 1, pp. 197–200.

[19] K. Johansson, O. Gustafsson, and L. Wanhammar, “Implementation of low-complexity FIR filters using serial arithmetic,” in *Proc. IEEE Int. Symp. Circuits Syst.*, Kobe, Japan, May 23–26, 2005, vol. 2, pp. 1449–1452.

[20] K. Johansson, O. Gustafsson, A. G. Dempster, and L. Wanhammar, “Trade-offs in low power multiplier blocks using serial arithmetic,” in *Proc. National Conf. Radio Science (RVK)*, Linköping, Sweden, June 14–16, 2005, pp. 271–274.

[21] K. Johansson, O. Gustafsson, and L. Wanhammar, “Trade-offs in multiplier block algorithms for low power digit-serial FIR filters,” in *Proc. WSEAS Int. Conf. Circuits*, Vouliagmeni, Greece, July 10–12, 2006

[22] K. Johansson, O. Gustafsson, and L. Wanhammar, “Multiple constant multiplication for digit-serial implementation of low power FIR filters,” *WSEAS Trans. Circuits Syst.*, vol. 5, no. 7, pp. 1001–1008, July 2006..

[24] A. G. Dempster and M. D. Macleod, “Constant integer multiplication using minimum adders,” *IEE Proc. Circuits Devices Syst.*, vol. 141.