International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

# COPY RIGHT



IJIEMR Transactions, online available on 7th Apr 2018. Link :http://www.ijiemr.org/downloads.php?vol=Volume-7&issue=ISSUE-04

Title: FIR FILTER ARCHITECTURE FOR HIGH PERFORMANCE

Volume 07, Issue 04, Pages: 35–41.
Paper Authors

**PATIL SINDHU,DR.N.A.V.PRASAD**
S.V. Institute of Technology, Anantapur


USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per UGC Guidelines We Are Providing A Electronic Bar Code

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

# FIR FILTER ARCHITECTURE FOR HIGH PERFORMANCE

**PATIL SINDHU[1],DR.N.A.V.PRASAD[2]**
[1]PG Scholar, Department of ECE,S.V. Institute of Technology, Anantapur
[2]Professor & HOD, Department of ECE,S.V. Institute of Technology, Anantapur

**Abstract:** Transpose form finite-impulse response (FIR) filters are inherently pipelined and support multiple constant multiplications (MCM) technique that results in significant saving of computation. However, transpose form configuration does not directly support the block processing unlike direct form configuration. In this paper, we explore the possibility of realization of block FIR filter in transpose form configuration for area-delay efficient realization of large order FIR filters for both fixed and reconfigurable applications. Based on a detailed computational analysis of transpose form configuration of FIR filter, we have derived a flow graph for transpose form block FIR filter with optimized register complexity. A generalized block formulation is presented for transpose form FIR filter. We have derived a general multiplier-based architecture for the proposed transpose form block filter for reconfigurable applications. A low-complexity design using the MCM scheme is also presented for the block implementation of fixed FIR filters. The proposed structure involves significantly less area delay product (ADP) and less energy per sample (EPS) than the existing block implementation of direct-form structure for medium or large filter lengths

**Index Terms**—0–1 integer linear programming (ILP), digit-serial arithmetic, finite impulse response (FIR) filters, CSE , multiple constant multiplications.

## 1. INTRODUCTION

**F**INITE 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 [1]. These are mainly consists of multiplication of a vector of input samples with a set of constant coefficients is known as MCM operations.Multiple constant multiplications (MCM) are typical operations in digital signal processing (DSP) as well as in the design of finite-impulse-response (FIR) filters, as shown in Fig.1 (a).Multiplications are expensive in terms of area and power consumption, when implemented in hardware. The relative cost of an adder and a multiplier in hardware, depends on the adder and multiplier architectures. For example, a *k x k* array multiplier has k times the logic (and area) and twice the latency of the slowest ripple carry adder. Since the values of the coefficients are known beforehand, the full flexibility of a multiplier is not necessary, and it can be more efficiently implemented by converting it into a sequence of additions/subtractions and shift operations are shown in fig.1 (b).

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

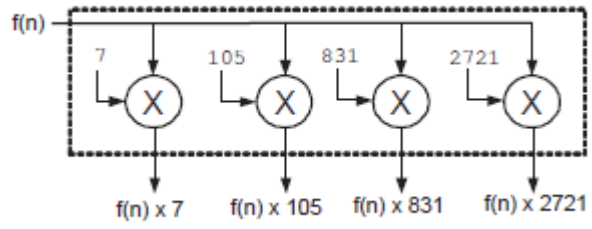


Figure 1(a): A multiplier-based MCM example

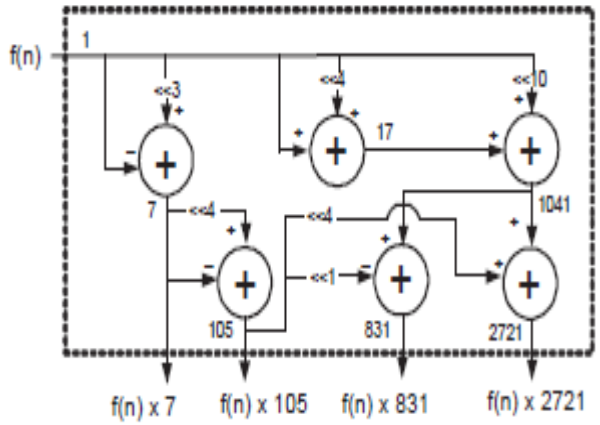


Figure 1(b): A multiplierless-based MCM example

For the shift-adds implementation of constant multiplications, a straightforward method, generally known as digit based recoding [2], 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 43*x*. Their decompositions in binary are listed as follows:

29*x* = (11101)bin*x* = x<<4 + x<<3 + x<<2 + *x*

43*x* =(101011)bin*x* = x<<5 + x<<3 + x<<1 + *x*

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

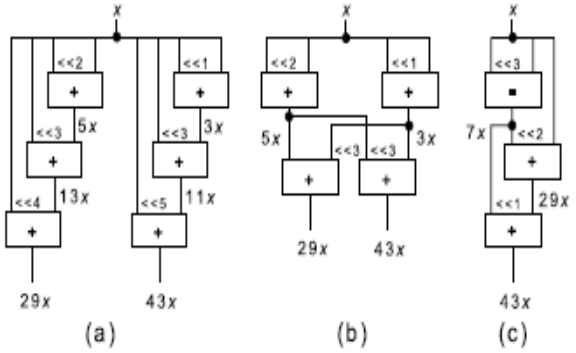


Fig. 2. Shift-adds implementations of 29*x* and 43*x*. (a) Without partial product sharing [2] and with partial product sharing. (b) Exact CSE algorithm [5]. (c) Exact GB algorithm [6].

The algorithms designed for the MCM problem can be categorized in two classes: common sub expression elimination (CSE) algorithms [3]–[5] and graph-based (GB) algorithm [6]–[8].The proposed algorithm that optimally solves this maximal sharing problem. This problem has been the subject of extensive research in recent years. Two key strategies have had a large impact in the optimization of MCMs. One is to consider not only adders, but also subtracter to combine partial terms, thus increasing the opportunity for the sharing of common sub expressions.The second is the usage of the Canonical Sign Digit (CSD) representation for the coefficients. This representation minimizes the number of non-zero digits; hence the maximal sub expression sharing search starts from a minimal level of complexity.In a recent paper, Park [9] propose the usage of a Minimal Signed Digit (MSD) representation for the coefficients. The MSD representation is obtained from the CSD representation by relaxing the requirement that there cannot be two consecutive non-zero digits. Under the

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

MSD representation, a given numerical value can have multiple representations. However, in all of them, the number of non-zero digits is the same as the CSD representation. The algorithm proposed in [9] exploits the redundancy of the MSD representation by choosing the MSD instance that leads to a maximal sharing in the implementation of efficient FIR filters.To the best of our knowledge, all previous solutions to this problem have been heuristic, providing no indication as to how far from the optimum their solution is. We propose an exact algorithm that is feasible for many real situations. We model this problem as a Boolean network that covers all possible partial terms which may be used to generate the set of coefficients in the MCM instance. The inputs to this network are shifted versions of the value that serves as input to the MCM operation. Each adder and subtracter used to generate a given partial term is represented as an AND gate. All partial terms that represent the same numerical value are ORed together. There is a single output which is an AND over all the coefficients in the MCM. We cast this problem into a 0-1 Integer Linear Programming (ILP) problem by requiring: that the output is asserted, meaning that all coefficients are covered by the set of partial terms found; while minimizing the total number of AND gates that evaluate to one, *i.e.*, the number of adders/subtracters. We have applied this algorithm to coefficients represented in binary, CSD and MSD representations. Note that the redundancy of the MSD representation can be readily incorporated in our model, where the equivalent MSD representations are simply new inputs to the OR gate that generates a given coefficient.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 [6] 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 [5].

## 2. Digit-Serial Arithmetic

In digit-serial arithmetic, data words are divided into digits, with a digit size of $d$ bits, which are processed in one clock cycle. The special cases of the digit-serial computation, called bit-serial and bit-parallel processing, occur when the digit size $d$ is equal to 1 and input data word length, respectively. The digit-serial computation plays an important role when the bit-serial implementations cannot meet delay requirements and the bit-parallel designs require excessive hardware. Thus, an optimal tradeoff between area and delay can be obtained by changing the digit size parameter ($d$). The fundamental digit-serial operations were introduced in [8]. The digit-serial addition, subtraction, and left shift operations are depicted in Figure 3 when $d$ is equal to 3. Notice from Figure 3(a) that in a digit-serial addition operation, in general, the number of required full adders (FAs) is equal to $d$ and the number of necessary D flip-flops is always 1. The subtraction operation (Figure 3(b)) is

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

implemented using 2's complement, requiring the initialization of the D flip-flop with 1 and additional *d* inverter gates with respect to the digit-serial addition operation.

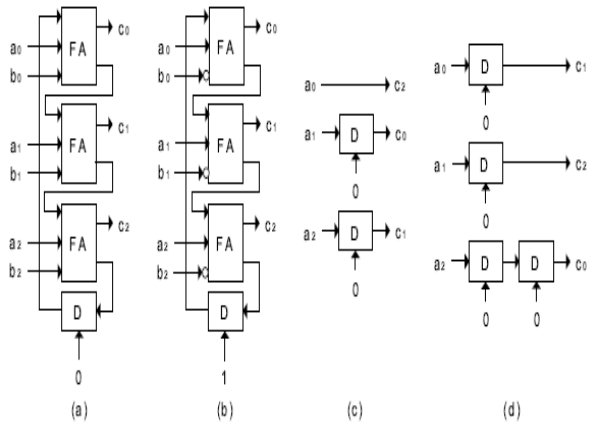


Figure 3: The digit-serial operations when *d* is 3: (a) addition operation; (b) subtraction operation; (c) left shift by 2 times;(d)left shift by 4 times.

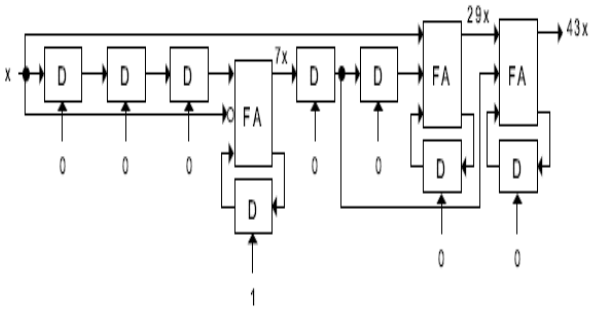


Figure 4: Bit-serial realization of shift-adds implementation of29*x* and 43*x* given in Figure 2(c).

As an example on digit-serial realization of constant multiplications under the shift-adds architecture, Figure 4 illustrates the bit-serial implementation of 29*x* and 43*x* obtained by the exact GB algorithm [2] given in Figure 2(c). The network includes 2 bit serial additions, 1 bit-serial subtraction, and 5 D flip-flops for all the left shift operations. Observe from Figure 4 that at each clock cycle, one bit of the input data *x* is applied to the network input and one bit of the constant multiplication output is computed. Note that the digit-serial design of the MCM operation occupies significantly less area when compared to its bit-parallel design and the area of the design is not dependent on the bit-width of the input data. However, the latency of the MCM computation is increased due to the serial processing. Suppose that *x* is a 16-bit input value. To obtain the actual output of 29*x* and 43*x* in the bit-serial network of Figure 4, 21 and 22 clock cycles are required respectively. Thus, necessary bits must be appended to the input data *x*, *i.e.*, 0s, if *x* is an unsigned input or sign bits, otherwise. Moreover, in the case of the conversion of the outputs obtained in digit-serial to the bit parallel format, storage elements and control logic are required.Note that while the sharing of addition/subtraction operations reduces the complexity of the digit-serial MCM design (since each addition and subtraction operation requires a digit-serial operation), the sharing of shift operations for a constant multiplication reduces the number of D flip-flops, and consequently, the design area. Observe from Figure 4 that two D flip-flops cascaded serially to generate the left shift of 7*x* by two can also generate the left shift of 7*x* by one without adding any hardware cost.The exact CSE algorithms that formalize the MCM problem as a 0–1 ILP problem were introduced in [23] and [24]. In these algorithms, the target constants are defined under a number representation and all possible implementations of constant multiplications are extracted from the representations of constants. The problem reduction and model simplification techniques for the exact CSE

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

algorithms were presented in [9]. The exact GB algorithms that search for a solution with the minimum number of operations in breadth-first and depth-first manners were introduced in [12]. Efficient GB algorithms that includes two parts, i.e., optimal and heuristic, were introduced in [10]–[12]. In their optimal parts, each target constant that can be implemented with a single operation is synthesized. If there exist unimplemented elements left in the target set,then they switch to their heuristic parts where the required intermediate constants are found. The RAG-n algorithm [10] initially chooses a single unimplemented target constant with the smallest single coefficient cost evaluated by the algorithm and then synthesizes it with a single operation including one(two) intermediate constant(s) that has(have) the smallest value in its heuristic part. The Hcub algorithm [11] selects a single intermediate constant that yields the best cumulative benefit over all unimplemented target constants for the implementation of each target constant. The approximate algorithm [12] computes all possible intermediate constants that can be synthesized with the current set of implemented constants using a single operation and chooses the one that leads to the largest number of synthesized target constants. For the MCM-DS problem, the GB algorithms based on RAG-n were introduced. The RSAG-n algorithm chooses the intermediate constant(s) that require the minimum number of shifts. The RASG-n algorithm selects the intermediate constant(s) with the minimum cost value as done in RAG-n, but if there are more than one possible intermediate constant, it favors the one that requires the minimum number of shifts.

## 3. SIMULATION RESULTS

GB algorithm can be applied for any coefficient pair combinations. Hence GB algorithm is used and number of operations is reduced drastically than other algorithms.

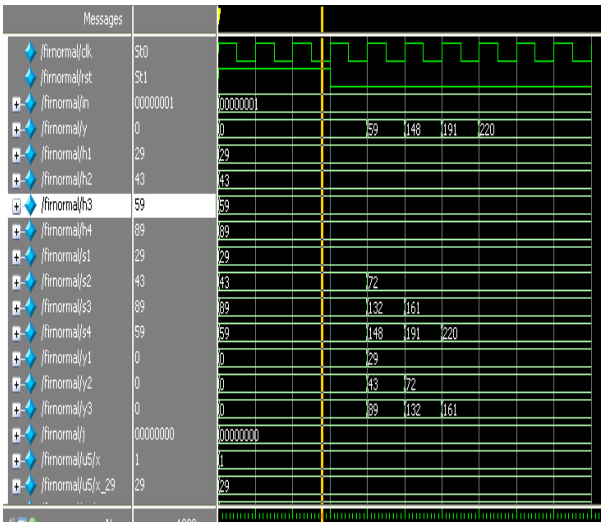


**Fig5.** Output for FIR filter with digit based recoding algorithm

Four filter coefficient 29,43,59,89 values are taken for digit serial FIR filter design. X(n) is taken as a input sequence and Y(n) is taken as output sequence.

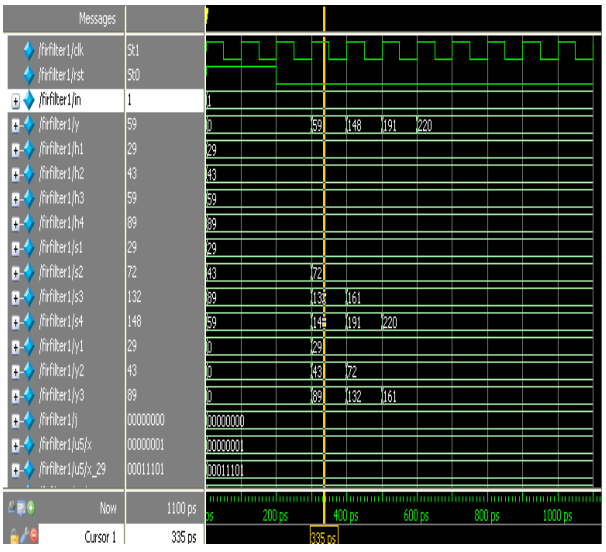


**Fig6.** Output for FIR filter with CSE algorithm

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

Fig.5 shows 4 tap FIR filter with without partial product sharing (Digit based recoding) algorithm and Fig.6 displays 4 tap filter with CSE algorithm.

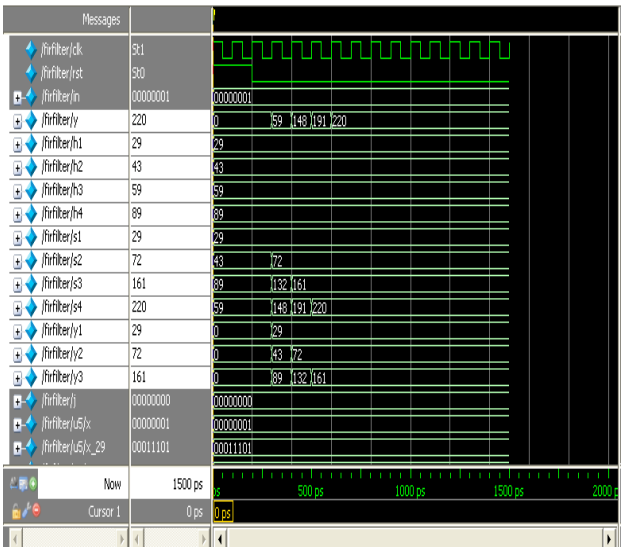


**Fig7.** Output for FIR filter with GB algorithm

Fig.7 displays 4 tap filter with GB algorithm. This simulation result was displayed by modelsim software. These are the simulation results displayed by modelsim software.

### 3.1. FIR FILTER DEVICE UTILIZATION REPORT

**Table -1: Delay and Gate Count Comparison**

| FIR Filter | Delay | Device Utilization |
|---|---|---|
| Normal method | 14.203 | 323 |
| CSE Algorithm | 15.528 | 321 |
| GB Algorithm | 12.875 | 299 |

## 4. CONCLUSION

Thus the implementation of digit serial FIR filter was implemented with low complexity MCM architectures using GB algorithm. Device utilization and delay values are compared for hardware implementation. Hence this MCM approach drastically reduces the system complexity, area and delay and FPGA hardware real time implementation has performed with spartan3 version. Future enhancement of this paper is to design MCM architecture with more coefficient pairs for FIR filter implementation.

## 5. REFERENCES

[1] L. Wanhammar, *DSP Integrated Circuits*. New York: Academic, 1999.

[2] M. Ercegovac and T. Lang, *Digital Arithmetic*. San Mateo, CA: Morgan Kaufmann, 2003.

[3] 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.

[4] I.-C. Park and H.-J. Kang, "Digital filter synthesis based on minimal signed digit representation," in *Proc. DAC*, 2001, pp. 468–473.

[5] 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.

[8] 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.

[9] P. Cappello and K. Steiglitz. Some Complexity Issues in Digital Signal Processing. *IEEE Trans. on Acoustics, Speech, and Signal Processing*,32(5):1037.1041, October 1984.

[10] R. Hartley and P. Corbett. Digit-Serial Processing Techniques. *IEEE TCAS II*, 37(6):707-719, 1990.

[11] L. Aksoy, E. Gunes, and P. Flores. Search Algorithms for the Multiple Constant Multiplications Problem: Exact and Approximate. *Elsevier Journal on Microprocessors and Microsystems*, 34:151-162,
2010.

[12] 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.

[13] P. Flores, J. Monteiro, and E. Costa, "An exact algorithm for the maximal sharing of partial terms in multiple constant multiplications," in *Proc. Int. Conf. Comput.-Aided Design*, Nov. 2005, pp. 13–16.