A New algorithm for multiple constant multiplications with low power consumption

A RAGHAVA RAJU[1] **SUGGUNA LAKSHMI LAVANYA**[2]

raghavaraju26@gmail.com[1] lavanyalakshmi40@gmail.com[2]

[1]PG Scholar, Dept of ECE, Chebrolu Engineering College, Chebrolu, Guntur, A.P, India.

[2]Associate Professor, Guide, Dept of ECE, Chebrolu Engineering College, Chebrolu, Guntur, A.P, India.

ABSTRACT:

Multiplications are costly operations in FIR filters. But for any given filter, the filter weights are constants. Several techniques have been developed over the years for the efficient realization of constant multiplications by a network of add/subtract-shift operations [6]. Constant multiplication methods are broadly of two types, (i) single constant multiplication (SCM) methods and (ii) multiple constant multiplication (MCM) methods. In this brief, Radix-2^{r}arithmetic is applied to the multiple constant multiplication (MCM) problems. Given a number M of nonnegative constants with a bit length N, we determine the analytic formulas for the maximum number of additions, the average number of additions, and the maximum number of cascaded additions forming the critical path. We get the first proven bounds known so far for MCM. In addition to being fully predictable with respect to the problem size (M, N), the RADIX-2r MCM heuristic exhibits sub linear runtime complexity O(M×N/r),where r is a function of (M, N). For high complexity problems, it is most likely the only one that is even feasible to run. Another merit is that it has the shortest adder depth in comparison with the best published MCM algorithms.

Index Terms—High-speed and low-power design, linear time invariant (LTI) systems, multiplier less single/multiple constant multiplication (SCM/MCM), Radix-2^r arithmetic.

1. INTRODUCTION:

A great deal of research has been done to develop effective algorithms to identify the optimal set of non-redundant sub expressions to achieve the minimum number of logic operators and the minimum logic depth of the MCM. Irrespective of differences in methodology and the level of optimality, in all these works, after the common sub expression terms are determined and the ADD/SUB network of non-redundant sub expressions (or terms) is formed, the product value corresponding to each of the coefficients is computed by an adder-tree that sums up its relevant terms. Two adder-trees are formed, for computing the product of a pair of coefficients using shifted versions of unique CS terms ‘1’, ‘10-1’ and ‘1001’ from the term-networks or sub expression-networks.

A brief characterization study of the number of ADD/SUB operators in different parts of arbitrary filters using 2-bit recursive common sub expression elimination for MCMs reveals that the number of operators used to form the adder-tree networks is very significant, sometimes a few times more, compared to that in the term networks. While the main research focus of MCM is on more effective common sub expression sharing techniques, optimizations on adder-trees are largely omitted. Irrespective of which common sub expression identification algorithm is used, the formation of an adder-tree is commonly handled by the same tree-height minimization algorithm that guarantees the height of generated adder-tree is the minimum at the operator level. In this paper, we present a method of derivation of equivalent adder-trees to minimize the adder tree resource. We have developed the cost model of the shift ADD/SUB network by bit-level analysis, which could be reduced by suitable scheduling of operations on the adder-tree. We find that significant area and power reduction (up to 15% and 11.6% respectively with an average of 8.46% and 5.96%) can be achieved on top of already optimized MCM blocks.

Multiple-Constant multiplication:

The multiple-constant multiplication (MCM) problem is to determine a shift-and-add network that can realize multiplication of a single input, x(n), with a set of coefficients, H. It is sufficient to only consider odd integer coefficients, since even and fractional coefficients can be obtained by an appropriate shift operation. The sign of the coefficient can also in most cases be compensated for in other parts of the implementation. Hence, the coefficient set, C, which is the input to the MCM algorithm, as illustrated in Fig. 1, is assumed to only contain unique positive odd integers. The shift- and-add networks are often illustrated using the directed acyclic graph representation of multiplication introduced. Each node, except for the input node, corresponds to an addition/subtraction and each edge corresponds to a shift operation, i.e., a multiplication by a power-of-two. The nodes are assigned values, which are referred to as fundamentals. A fundamental, fi, is computed from two other fundamentals fj and fk as

$$f_i = e_j f_j + e_k f_k \qquad (1)$$

Figure1. Design path for MCM blocks.

Where ej and ek are edge values, as illustrated in Fig. 2 (a). The obtained signal value in this node will then be fi times the input signal. As an example, consider the coefficient set C= {33, 57}. The coefficient 33 can be obtained directly from the input as 1 + 25. However, the coefficient 57 cannot be realized and a new node must therefore be included. For example, the value 3 solves the problem, as shown in Fig. 2 (b). Hence, the set of extra fundamentals is E= {3}, and the total fundamental set F=C∪E= {3, 33, 57} is the output of the MCM algorithm.

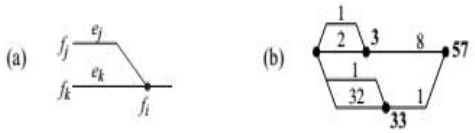

Figure2. (a) Graph representation. (b) Shift-and-add MCM block.

A. Scaling

All nodes in the MCM block are explicitly scaled using safe scaling, i.e., there will never be an overflow and the output word length is just enough to represent all possible outputs. Quantization is not considered, and, hence, full precision is kept throughout the MCM block. Using safe scaling, the number of output bits, Wi, from the add operation associated with the fundamental fi is

$$W_i = W_0 + \lceil \log_2(|f_i|) \rceil \quad (2$$

Where W0 is the word length at the input, x(n), of the MCM block.

B. Bit-Level Optimization

For each fundamental fi obtained according to (1), there are two possibilities associated with the magnitude of the edge values, ej and ek .In the first case the value at one of the input nodes is left shifted at least once while the significance of the other value is unchanged (otherwise the result would not be odd). The shift operation is, for simplicity, always associated with the input node fj. The second case occurs when the magnitude of both edge values is less than one, i.e., two right shifts are performed. In order to obtain an odd integer fundamental, the edge values must then be of equal significance. Hence, we have

$$\begin{aligned} &|e_j| > 1, |e_k| = 1 \quad \text{(one left shift) or} \\ &|e_j| = |e_k| < 1 \quad \text{(two right shifts)} \end{aligned} \quad (3)$$

When signs are also considered, this leads to the five different cases illustrated in Fig. 4. The hardware complexity for these cases can be determined by counting the number of full adder (FA) cells required to realize each word level (WL) adder. The number of overhead full adders, defined as the difference between the total number of full adders, n FA, i, and the input word length, W0, is for the fundamental fi computed

$$n_{FA,i}^{OH} = n_{FA,i} - W_0 = \begin{cases} \lceil \log_2(|f_i|) \rceil - \log_2(|e_j|), & |e_j| > 1 \\ \lceil \log_2(|f_i|) \rceil, & |e_j| < 1 \end{cases} \quad (4)$$

Furthermore, it was also shown that the use of half adders, which would be required in a direct implementation of the case in Fig. 4 (c), can be eliminated by changing the signs of the edge weights.

C. Interconnection

It can be shown that it is preferable to separate the tasks of finding the fundamentals, F, and to find a suitable interconnection graph, G, as illustrated by the state chart in Fig.1. The reason is simply that it is easier to create a good interconnection when all information is available, i.e., after the complete MCM problem has been solved. The focus was on this problem. It was shown that an interconnection graph, G, where all nodes have minimum depth for a given fundamental set, F, can be obtained using the following algorithm:

1. Initialize a graph G that only contains the input node.

2. Add to G all fundamentals, fi ∈F, which can be obtained from the nodes in G, i.e., by adding/subtracting any two realized fundamentals (only the input in the first iteration) according to (1).

3. Repeat step 2 until all fundamentals in F have been realized. By applying this algorithm, one level at a time is added to the graph, i.e., the first time step 2 is executed all realized nodes will have depth 1, the second time depth 2 and so on. This assures that all fundamentals are realized at minimum depth given F. Hence, the adder depth can, if possible, only be reduced by adding or changing the extra fundamentals, i.e., by using a different MCM algorithm in the preceding step. By considering different interconnections, it is possible to make sure that each fundamental is realized using a minimum number of overhead full adders according to (4), to also optimize the complexity.

Multiple Constant Multiplication (MCM) is an arithmetic operation that multiplies a set of fixed-point constants $\{C_0, C_1, C_2, \cdots, C_{M-1}\}$ with the same fixed-point variable X. From a circuit point of view, MCM dominates the complexity of the whole category of Linear Time Invariant (LTI) systems, such as, FIR/IIR filters, DSP transforms (DCT, DFT, Walsh …), LTI controllers, crypto-systems, etc. To be efficiently implemented, MCM must avoid costly multipliers. The hardware alternative must be multiplier less, i.e., using only additions, subtractions, and shifts. Therefore, the MCM problem is defined as the process of finding the minimum number of addition/subtraction operations. The computational complexity of MCM is conjectured to be NP-hard.

2. LITERATURE SURVEY:

The complexity of digital filter is greatly influenced by the no of multiplication needed in the multiplier block. The complexity can be greatly reduced if efficient number system is used. Proposed an efficient MSD representation which can provide no of forms which has minimum no of nonzero digits for the constant.

The multiplier block of digital FIR filter has significant impact on the complexity and performance of the design because large numbers of constant multiplications are required to achieve high through-put and the multiplication operation is considered to be expensive as it occupies significant area Proposed efficient shift-add design of digit serial multiplications and yielded significant area and power reduction than those compared with the multiplier blocks which are implemented by using digit serial constant multiplier. Hence the multiplications of input data with filter coefficients is done using shift-add architectures where each constant multiplication is realized by using addition/subtraction and shift operation in Multiple Constant Multiplication(MCM) block.

Title: Multiple constant multiplications: efficient and versatile framework and algorithms for exploring common sub expression elimination

Many applications in DSP, telecommunications, graphics, and control have computations that either involve a large number of multiplications of one variable with several constants, or an easily be transformed to that form. A proper optimization of this part of the computation, which we call the multiple constant multiplication (MCM) problem, often results in a significant improvement in several key design metrics, such as throughput, area, and power. However, until now little attention has been paid to the MCM problem.

After defining the MCM problem, we introduce an effective problem formulation for solving it where first the minimum number of shifts that are needed is computed, and then the number of additions is minimized using common sub expression elimination.

The algorithm for common sub expression elimination is based on an iterative pair wise matching heuristic. The power of the MCM approach is augmented by preprocessing the computation structure with a new scaling transformation that reduces the number of shifts and additions.

An efficient branch and bound algorithm for applying the scaling transformation has also been developed. The flexibility of the MCM problem formulation enables the application of the iterative pair wise matching algorithm to several other important and common high level synthesis tasks, such as the minimization of the number of operations in constant matrix-vector multiplications, linear transforms, and single and multiple polynomial evaluations. All applications are illustrated by a number of benchmarks.

Title: Global Optimization of Common Sub expressions for Multiplier less Synthesis of Multiple Constant Multiplications

In the context of multiple constant multiplication (MCM) design, we propose a novel common sub expression elimination (CSE) algorithm that models the optimal synthesis of coefficients into a 0-1 mixed-integer linear programming (MILP) problem. A time delay constraint is included for synthesis. We also propose coefficient decompositions that combine all minimal signed digit (MSD) representations and the shifted sum (difference) of coefficients. In the examples we demonstrate, the proposed solution space further reduces the number of adders/subtractors in the MCM synthesis.

3. PROPOSED SYSTEM RADIX-2^{r}MCM

A non negative N-bit constant C is expressed in Radix-2^{r}as

$$C = \sum_{j=0}^{(N+1)/r-1} \left(c_{rj-1} + 2^0 c_{rj} + 2^1 c_{rj+1} + 2^2 c_{rj+2} + \cdots + 2^{r-2} c_{rj+r-2} - 2^{r-1} c_{rj+r-1}\right) \times 2^{rj} = \sum_{j=0}^{(N+1)/r-1} Q_j \times 2^{rj} \quad (1)$$

where $c_{-1} = c_N = 0$ and $r \in N^*$. In (1), the two's complement representation of Cis split into [(N+1)/r] slices (Qj),each of r+1bit length.

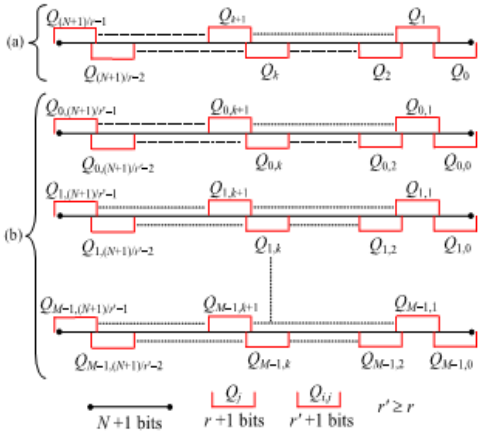

Figure 3: Slice partitioning of N-bit constants in Radix-2^r. (a) RADIX-2^r SCM. (b) RADIX-2 r MCM.

Each pair of two contiguous slices has one overlapping bit. A digit set DS(2r)corresponds to (1), such that

$Q_j \in DS(2^r) = \{-2^{r-1}, -2^{r-1}+1, \ldots\ldots\text{-}1,0,1,2^{r-1}\text{-}1,2^{r-1}\}$

The sign of the Qj term is given by the c_{rj+r-1} bit, and $|Q_j| = 2^{k_j} \times m_j$, with $k_j \in \{0, 1, 2, \ldots, r-1\}$ and $m_j \in OM(2^r) \cup \{0, 1\}$, where $OM(2^r) = \{3, 5, 7, \ldots, 2^{r-1} - 1\}$. $OM(2^r)$ is the set of odd positive digits in Radix-2^r recoding, with $|OM(2^r)| = 2^{r-2} - 1$.

RADIX-2^r SCM can be easily extended to MCM. In MCM, a single variable X is simultaneously multiplied by a set of M constants {C0,C1,C2,...,CM−1}, having all the same bit size N. In RADIX-2^r MCM, each constant Ci is split into $\lceil (N+1)/r' \rceil$ slices(Qij), each slice of bit length r^1+1[see Fig. 3(b)]. Thus, the maximal number of partial products (PPs)is M×[(N+1)/r^1]plus a maximum of $2^{r'-2} - 1$nontrivial

PP{3×X,5×X,7×X,..., $(2^{r'-1} - 1) \times X$ }hat can be invoked during the PP generation process. Aided by Fig. 3(b) and using the same reasoning based on [1, Th. 1], we can easily demonstrate that the maximum number of additions (Upb)in RADIX-2^r MCM is

$$Upb(r') = M \times \lceil (N+1)/r' \rceil + 2^{r'-2} - 1 - M.$$

$Upb(r')$ is minimal for $r' = 2 \cdot W(\sqrt{M \cdot (N+1) \cdot \log(2)})/\log(2)$.

Note that $r' \geq r$ (see Fig. 1) due to the product M×(N+1). In fact, RADIX-2rSCM is a particular case ofRADIX-2r MCM for M=1. Pursuing also the same reasoning developed, we can straightforwardly derive the analytic expressions of Avg and Ath(see Table II). However, in real-life applications of MCM, for instance, in the transposed form of a FIR filter, the coefficients will most likely have different bit sizes. Assume that, for each constant Ci corresponds a bit size Ni, the total number of PPs for a set of M constants will be equal to M−1 i=0 (Ni+1)/r .

Likewise, taking this fact into account, we can easily prove the analytic equations in Table III. Unlike existing MCM algorithms, in RADIX-2 r MCM, each constant is implemented apart, independently from the others. However, all constants share the same set of nontrivial PPs. This is illustrated by the following MCM example: C0=(84AB5)H, C1=(64AB55)H,andC2= (5959595B)H. To the constantsC0, C1,andC2 corresponds the bit sizes N0=20, N1=23,andN2=31, respectively. Thus, for 2 i=0(Ni+1)=77,ther =r1 formula in Table III gives r' =4, which is the value that minimizes Upb.

Hence, the solution given by the online version of RADIX-2 r MCM is

$C0 \times X = X \times 2^{19} + X1 \times 2^{12} - X1 \times 2^{8} - X1 \times 2^{4} + X1$

$C1 \times X = X0 \times 2^{21} + X1 \times 2^{16} - X1 \times 2^{12} - X1 \times 2^{8} + X1 \times 2^{4} + X1$

$C2 \times X = X0 \times 2^{29} - X2 \times 2^{24} + X0 \times 2^{21} - X2 \times 2^{16} + X0 \times 2^{13} - X2 \times 2^{8} + X0 \times 2^{5} - X1$
withX0=3×X=X×2+X, X1=5×X=X×2 2+ X,andX2=7×X=X×2 3−X.

Note that the online version offers three solutions. The one given above corresponds to the optimization of adder depth.

We introduced a variant of RADIX-2^r called R3. It has a better Avg with the sameUpbandAth.R3MCMgives

$C0 \times X = X \times 2^{19} + U75 \times 2^{8} - U75, U75 = X1 \times 2^{4} - X1$

$C1\times X=U101\times 2^{16}$ $-U85\times 2^{8}$ $+U85, U101=X0\times 2^{5}-X1$ $U85=X1\times 2^{4}+X1$

$C2\times X=U89\times 2^{24}+U89\times 2^{16}+U89\times 2^{8}+U91$ $U89=X0\times 2^{5}-X2, U91=X0\times 2^{5}-X1.$

These two solutions are compared in Table IV with the ones provided by the most efficient MCM algorithms. Note that the canonical signed digit (CSD) representation, RADIX-2^{r}, and R3 MCM, as digit-recoding algorithms, allow both serial and parallel implementations, whereas Hcub, BHM, and Lefèvre's common subpattern (CSP) are limited to serial implementation only due to the shared terms. This issue will be detailed further in the next section.

4. RESULTS

4.1 SIMULATION RESULT:

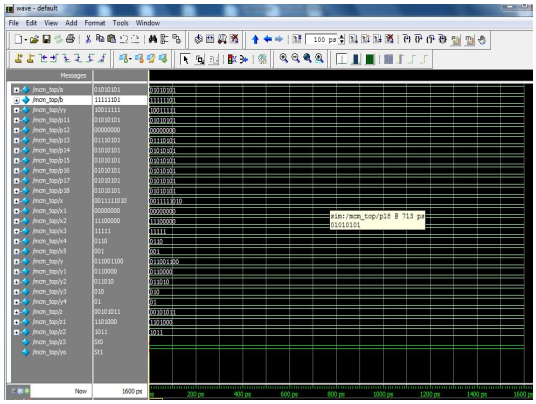

Figure4: Simulation results for MCM-top

4.2 SYNTHESIS RESULTS:

The developed project is simulated and verified their functionality. Once the functional verification is done, the RTL model is taken to the synthesis process using the Xilinx ISE tool. In synthesis process, the RTL model will be converted to the gate level netlist mapped to a specific technology library. Here in this Spartan 3 family, many different devices were available in the Xilinx ISE tool. In order to synthesis this design the device named as "XC3S500E" has been chosen and the package as "FG320" with the device speed such as "-4". This design is synthesized and its results were analyzed as follows

TECHNOLOGY SCHEMATIC;

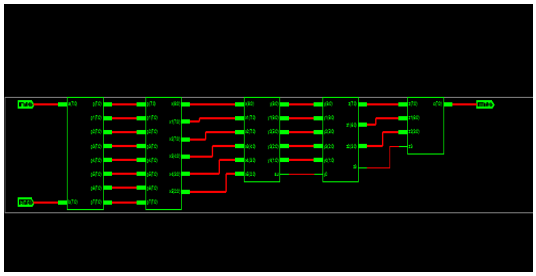

Figure 5: Technology schematic for MCM

DESIGN SUMMARY:

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	45	3584	1%
Number of 4 input LUTs	79	7168	1%
Number of bonded IOBs	24	221	10%

Figure 6: Device utilization for MCM

TIMING REPORT:

```
Delay:               20.632ns (Levels of Logic = 17)
  Source:            a<4> (PAD)
  Destination:       yy<7> (PAD)

  Data Path: a<4> to yy<7>
                                Gate     Net
    Cell:in->out      fanout   Delay    Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             11    0.821    1.483  a_4_IBUF (a_4_IBUF)
     LUT2:I0->O             2    0.551    0.945  m1/p2<4>1 (p13<4>)
     LUT4:I2->O             2    0.551    1.216  m2/a4/Mxor_su_xo<1>1 (x
     LUT3:I0->O             1    0.551    0.996  m3/m1/ca_SW0 (N26)
     LUT4:I1->O             1    0.551    0.869  m3/m1/ca (y1<0>)
     LUT4:I2->O             2    0.551    1.072  m4/n1/Mxor_su_xo<3>1 (z
     LUT3:I1->O             1    0.551    1.140  m5/f1/Mxor_su_xo<1>1 (m
     LUT4:I0->O             2    0.551    0.000  m5/Madd_c_lut<0> (m5/Ma
     MUXCY:S->O             1    0.500    0.000  m5/Madd_c_cy<0> (m5/Mad
     MUXCY:CI->O            1    0.064    0.000  m5/Madd_c_cy<1> (m5/Mad
     MUXCY:CI->O            1    0.064    0.000  m5/Madd_c_cy<2> (m5/Mad
     MUXCY:CI->O            1    0.064    0.000  m5/Madd_c_cy<3> (m5/Mad
     MUXCY:CI->O            1    0.064    0.000  m5/Madd_c_cy<4> (m5/Mad
     MUXCY:CI->O            1    0.064    0.000  m5/Madd_c_cy<5> (m5/Mad
     MUXCY:CI->O            0    0.064    0.000  m5/Madd_c_cy<6> (m5/Mad
     XORCY:CI->O            1    0.904    0.801  m5/Madd_c_xor<7> (yy_7_
     OBUF:I->O                   5.644           yy_7_OBUF (yy<7>)
    ----------------------------------------
    Total                      20.632ns (12.110ns logic, 8.522ns route
                                        (58.7% logic, 41.3% route)
```

Figure 7: Timing report for MCM

5. CONCLUSION

A fully predictable and sub linear runtime MCM heuristic with the shortest adder depth has been developed (RADIX-2^r) and improved (R3). Its proven limits with an exact number of additions for the average, adder cost, and adder depth are the unique analytic bounds known so far for MCM. However, optimal bounds remain an open research problem.

REFERENCES

[1] A. K. Oudjida and N. Chaillet, "Radix-2 r arithmetic for multiplication by a constant,"IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 61, no. 5, pp. 349–353, May 2014.

[2] A. K. Oudjida, N. Chaillet, and M. L. Berrandjia, "Radix-2 r arithmetic for multiplication by a constant: Further results and improvements," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 62, no. 4, pp. 372–376, Apr. 2015.

[3] V. Lefèvre, "Multiplication by an integer constant," INRIA, Lyon, France, INRIA Res. Rep. 4192, May 2001.

[4] A. G. Dempster and M. D. Macleod, "Use of minimum adder multiplier blocks in FIR digital filters,"IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., vol. 42, no. 9, pp. 569–567, Sep. 1995.

[5] Y. Voronenko and M. Püschel, "Multiplierless multiple constant multiplication,"ACM Trans. Algorithms, vol. 3, no. 2, pp. 11, May 2007.

[6] A. K. Oudjida and M. L. Berrandjia, RADIX-2 r MCM Web Page, Jun. 2015.

[7] A. Avizienis, "Signed-digit number representation for fast parallel arithmetic,"IRE Trans. Electron. Comput., vol. EC-10, no. 3, pp. 389–400, Sep. 1961.

[8] O. Gustafsson, "Lower bounds for constant multiplication problems," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 54, no. 11, pp. 974–978, Nov. 2007.

[9] K. Johansson, O. Gustafsson, L. S. DeBrunner, and L. Wanhammar, "Minimum adder depth multiple constant multiplication algorithm for low power FIR filters," in Proc. IEEE ISCAS, Rio de Janeiro, Brazil, May 2011, pp. 1439–1442.

[10] L. Aksoy, P. Flores, and J. Monteiro, "Exact and approximate algorithms for the filter design optimization problem"IEEE Trans. Signal Process., vol. 63, no. 1, pp. 142–154, Jan. 2015.

[11] X. Lu, Y. J. Yu, and P. K. Meher, "Fine-grained critical path analysis and optimization for area-time efficient realization of multiple constant multiplications,"IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 62, no. 3, pp. 863–872, Mar. 2014.

[12] M. Kumm, M. Hardieck, J. Willkomm, P. Zipf, and U. Meyer-Baese, "Multiple constant multiplication with ternary adders," inProc. 23rd Int. Conf. FPL Appl., Porto, Portugal, Sep. 2013, pp. 1–8.

[13] L. Aksoy, P. Flores, and J. Monteiro, "Efficient design of FIR filters using hybrid multiple constant multiplications on FPGA," inProc. IEEE ICCD, Seoul, Korea, Oct. 2014, pp. 42–47