Constant Multiplication and its Existing Techniques

Tonjam Gunendra Singh[1], Kuldeep Thingbaijam[2], Basanta Nameirakpam[3]

[1] Department of E&CE, Manipur Institute of Technology, Manipur University, Imphal, INDIA

tgunen@nielit.gov.in, kuldeepthingbaijam@gmail.com, basanta_n@rediffmail.com

DOI: 10.29322/IJSRP.9.01.2019.p8594
http://dx.doi.org/10.29322/IJSRP.9.01.2019.p8594

***Abstract*: -** Finite impulse response (FIR) filtering is a ubiquitous operation in digital signal processing systems and is generally implemented in full custom circuits due to high-speed and low-power design requirements. The complexity of an FIR filter is dominated by the multiplication of a large number of filter coefficients by the filter input or its time-shifted versions. Over the years, many high-level synthesis algorithms and filter architectures have been introduced in order to design FIR filters efficiently. The aim of this article is to discuss about constant multiplication, its types and the existing constant multiplication techniques.

***Keywords*:-** Common subexpression elimination, constant multiplication, digit recoding, directed acyclic graph, FIR filter, hybrid algorithm, multiple constant multiplication, single constant multiplication.

I. INTRODUCTION

An FIR operation is described by the following equation:

$$y(n)=\sum_{k=0}^{M-1}h(k)x(n-k) \qquad (1)$$

where $h(k)$, $k=0,1,2...M-1$ are the impulse response coefficients and M is the filter length (number of coefficients). According to Equation (1), the output $y(n)$ depends only on the present and past input values. FIR filters are popular because of its advantages such as linear phase response.

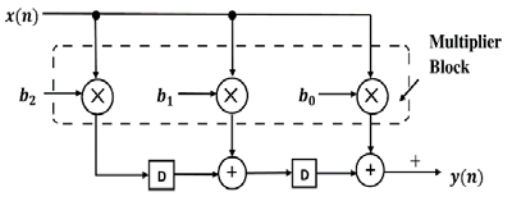

Figure *1. Transposed Direct – Form*

Here an input is multiplied by all the coefficients in a multiplier block at the same time. The multiplications are performed in parallel which are suitable for FPGA implementation. Further, it allows common intermediate multiplication terms to be shared among the constant multiplications.

The fundamental operations of a Finite Impulse Response (FIR) digital filter are multiply and accumulate (MAC). DSP systems are limited in terms of speed, area and power. Often these are antagonistic constraints and a trade-off depending on the design goals. So instead of using purpose multiplier where any two variables can be multiplied, multiplierless techniques are introduced. Here, the multiplication by the fixed coefficients can be implemented using shift and add/subtract operations only. Shifts can be implemented in hardware using wires and hence incur no cost. Furthermore, the number of adders can be reduced if the partial products are shared among the coefficients. This can reduce cost such as area and power.

II. TYPES OF CONSTANT MULTIPLICATIONS

1. SCM (SINGLE CONSTANT MULTIPLICATIONS)

SCM is the implementation of multiplication by a single constant. Here, the multiplication $y=cx$ of a variable x by a constant c is decomposed into additions, subtractions and shifts. The objective of SCM is to find the minimum number of operations.

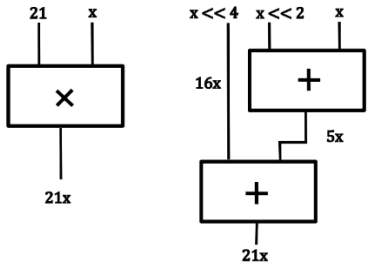

Figure *2. Multiplication using add and shift*

Consider the multiplication $21x$ .This can be expressed using adds and shifts as $21x=10101_2\,x=\ x\ <<4+x<<2+x$. A partial product is formed after every addition. This

decomposition of the multiplication can be improved after encoding the constant into representations such RADIX- 2^r

2. MCM (MULTIPLE CONSTANT MULTIPLICATIONS)
The SCM can be extended where a variable x is multiplied by several constants $c_1, \ldots c_n$ in parallel i.e. $y_i = c_i x$ with $1 \le i \le n$.

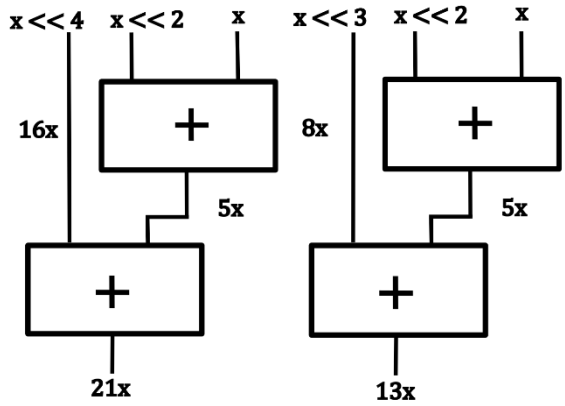

Figure *3. Multiplications of two constants* $21x$ *and* $13x$

SCM algorithms can solve MCM problems and vice-versa. However, there is a difference: in SCM intermediate terms can be shared within the constant, whereas; in MCM intermediate terms can be shared within and between constants. Thus, SCM algorithm can be fine-tuned to solve the SCM problem better. This sharing results in greater potential savings as the number of constants increases.

III. EXISTING MCM ALGORITHMS

1. DIGIT RECODING
In these algorithms, the number is recoded into representations such as canonic signed digit (CSD), minimal signed-digit (MSD), double base number system (DBNS) and RADIX- 2^r . The decomposition of the constant is obtained directly from the digit representation. These are the fastest and they have low computational cost. Canonic signed-digit (CSD) representation is a signed- digit (SD) system with the digit set $\{1, 0, -1\}$.-1 is generally represented as $\bar{1}$. CSD has the following properties - adjacent bits are non-zero; CSD representation is unique; CSD contains minimum number of non-zero bits; on average CSD contains about 33% fewer non- zero digits then the binary two's complement representation. For an N -bit constant, the number of additions is bounded by $(N+1)/2$ and tends asymptotically to an average of $N/3+1/9$. These results in a saving of 33% in add operations. CSD is widely used for constant multiplications even though a number of heuristics exist.

2. COMMON SUBEXPRESSION ELIMINATION
Subexpression elimination is applied to multiplication of a variable with a set of constants at the same time. Hartley (1991) introduced CSE representation which allowed the use of pattern search algorithms. CSE algorithms are performed on constants represented by a number representation such as SD and DBNS. It then examines all possible subexpressions i.e. the shift and add operations of constant implementations and finds redundant operations. The 'best', which is usually the most common, is selected to be shared among the constant multiplications.

An exact CSE algorithm to Hartley (1996) is presented in (Aksoy et al., 2014) which can handle binary, CSD and MSD representations and can also specify delay constraint. It showed that binary representation provides greater sharing of patterns allowing better are-efficient implementations whereas MSD is to be used while minimizing delay.

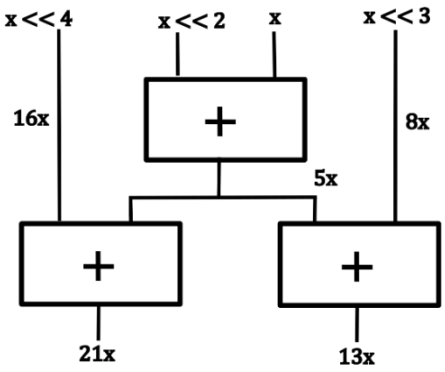

Figure *4. Common Subexpression Elimination*

Figure 4 shows how partial terms can be shared among the coefficient multiplications. Here, a common sub pattern or subexpression $5x$ is found in the representation of the constant multiplications $21x$ and $13x$.

3. DIRECTED ACYCLIC GRAPH
Multiplication by a constant which is decomposed into shifts and adders is represented using directed acyclic graphs. Each vertex has an in-degree of two except for one vertex which has in-degree zero. That vertex is referred to as input vertex or source vertex and corresponds to the input of the multiplier. Each vertex has out degree larger than or equal to one except for one which has out-degree zero. That vertex is referred to as the output vertex and corresponds to the output of the multiplier. Each edge is assigned a value of $\pm 2^n$ representing a multiplication of the value of the initial vertex of that edge by a value and corresponds to a shift and a subsequent addition or subtraction. Each node represents one addition or subtraction depending on whether an incoming node is labeled as $+2^n$ or -2^n .

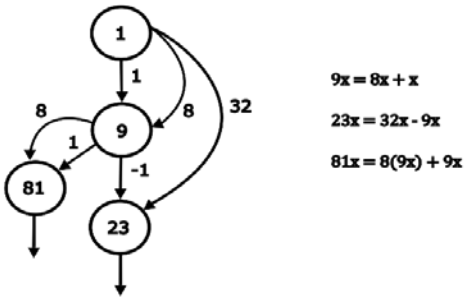

Figure 5. *Directed Acyclic Graph*

4. HYBRID ALGORITHM

Some algorithms like (BIGE) (Thong and Nicolici, 2011) combine features of CSE and DAG. It is an exact SCM algorithm based on DAG and CSE which guarantees optimality through an exhaustive search for constants up to 32 bits. The algorithms and heuristics presented above are designed mainly to minimize the number of adders or area. However, some algorithms consider both the minimum number of adders which represent area and adder-depth i.e. the number of adders in a maximal path which represents delay of the multiplier. The algorithm in (Kang and Park, 2001) allowed a trade-off between number of adders and the number of adder- depths i.e. between area and speed. Such algorithm showed lower resource usage with 54.1% reduction on average and similar performances compared to other adder graph methods.

IV. CONCLUSION

The fundamental operation of FIR filter and the reason of using multiplierless techniques in FIR filter has been discussed. Also, different types of constant multiplier and its different algorithms or techniques have been explained using suitable examples.

REFERENCES

[1] Pasko, R.,Schaumont, P.,Derudder, V.,Vernalde, S.,and Durackova, "*A new algorithm for elimination of common subexpressions*", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. **18**, Issue **1**, pp.**58–68**, **1999.**

[2] Dempster, A. G.,Demirsoy, S. S.,and Kale, "*Designing multiplier blocks with lowlogic depth*", IEEE International Symposium on Circuits and Systems, Vol. **5**, pp. **773–776**, **2002.**

[3] S. Rahimian Omam ,S.M.Fakhraie, and O. Shoaei "*Minimizing the adder cost in multiple constant multipliers*", IEICE Electronics Express, Vol.**3**, No.**14**, pp.**340–346**, **2006**.

[4] Aksoy, L., Gunes, E. and Flores, P."*Search algorithms for the multiple constant multiplications problem: exact and approximate"*, Elsevier Journal Microprocessor and Microsystems 34, pp.**151-162**, **2010**

[5] Y. Voronenko and M. P¨uschel, "*Multiplierless multiple constant multiplication*" ACM Trans. Alg., Vol. **3**, pp. **2**, **2007**.

[6] Martin Kumm, Martin Hardieck, Jens Willkomm, Peter Zipf , Uwe Meyer-Baese, "*Multiple Constant Multiplication with Ternary Adders*", IEEE, **2013**

[7] T. Sandhya Pridhini, Diana Aloshius, Aarthi Avanthiga, Rubesh Anand, "*Design of Multiple Constant Multiplication algorithm for FIR filter*", IJCSMC, Vol. **3**, Issue. **3**, pp.**438 – 444**, **2014.**

Authors Profile

Mr. *T. Gunendra Singh* pursed his Bachelor in Engineering in Electronics and Telecommunication Engineering in the year 1990 from NIT Silchar and currently pursuing Master of Technology in Electronics and Communication Engineering at Manipur Institute of Technology, Imphal. He worked at Manipur Electronics Development Corporation Ltd, Imphal, as Service Engineer from Sept 1990 till May 1998. He joined National Institute of Electronics and Information Technology (NIELIT), Imphal in the year 1998 as Senior Design Engineer and currently serving as Scientist "E" at the same Institute.

Mr. *Kuldeep Th.* pursed his Bachelor in Engineering in Electronics and Communication Engineering from Anna University, Tamil Nadu in the year 2008 and Master of Technology in VLSI Design and Embedded System from Visvesvaraya Technological University, Karnataka in the year 2013. He is currently serving as a faculty in National Institute of Electronics and Information Technology (NIELIT), Imphal since 2014.