Analysis of Adder-Trees for Multiple Constant Multiplication

Kaushiki Sharma[1], Prof. Ashish Raghuwanshi[2]

[1]M-Tech Research Scholar, [2]Research Guide Department of Electronics & Communication Engineering, IES, Bhopal

Abstract: To minimize the energy consumption per operation for the arithmetic parts of DSP circuits, such as digital filters. More specific, the focus is on single- and multiple-constant multiplication using serial arithmetic. The possibility to reduce the complexity and energy consumption is investigated. Multiplying by known constants is a common operation in many digital signal processing (DSP) algorithms. High performance DSP systems are implemented in custom hardware, in which the designer has the ability to choose which logic elements will be used to perform the computation. By exploiting the properties of binary multiplication, it is possible to realize constant multiplication with fewer logic resources than required by a generic multiplier.

Keyword:Critical path, genetic algorithm, high-speed, multiple constant multiplications (MCM).

I. INTRODUCTION

Multiplying a variable by a set of known constant coefficients is a common operation in many digital signal processing (DSP) algorithms. Compared to other common operations in DSP algorithms, such as addition, subtraction, using delay elements, etc., multiplication is generally the most expensive. There is a trade-off between the amount of logic resources used (i.e. the amount of silicon in the integrated circuit) and how fast the computation can be done. Compared to most of the other operations, multiplication requires more time given the same amount of logic resources and it requires more logic resources under the constraint that each operation must be completed within the same amount of time.

Serial Arithmetic

In digit-serial arithmetic, each data word is divided into digits that are processed one digit at a time. The number of bits in each digit is the digit-size, d. This provides a trade-off between area, speed, and energy consumption. For the special case where d equals the data word length we have bit-parallel processing and when d equals one we have bit-serial processing.

Digit-serial processing elements can be derived either by unfolding bit-serial processing elements or by folding bit-parallel processing elements. In Fig. 1.1, a digit-serial adder, subtractor, and shift operation is shown, respectively. These are the operations that are required to implement constant multiplication.

It is clear that serial architectures with a small digit-size have the advantage of area efficient processing elements. How speed and energy consumption depend on the digit-size is not as obvious. One main difference compared to parallel arithmetic is that the shift operations can be hardwired, i.e., without any flip-flops, in a bit-parallel architecture. However, the flip-flops included in serial shifts have the benefit to reduce the glitch propagation between subsequent adders/subtractors. To further prevent glitches pipelining can be introduced, which also increases the throughput. Note that fewer registers are required for pipelining in serialarithmetic compared to the parallel case. For example, in bit-serial arithmetic only one flip-flop is required for each pipelining stage and, in addition, the available shift operations can be used to obtain an improved design, i.e., with a shorter critical path.

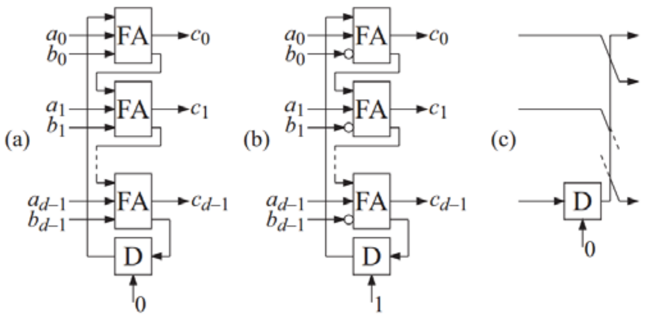

Fig. 1.1 Digit-serial (a) adder, (b) subtractors, and (c) left shift.

A general multiplier is needed if one performs multiplication between two arbitrary variables. However, when multiplying by a known constant, can exploit the properties of binary multiplication in order to obtain a less expensive logic circuit that is functionally equivalent to simply asserting the constant on one input of a general multiplier. In many cases, using a cheaper implementation for only multiplication still results in significant savings when considering the entire logic circuit because multiplication is relatively expensive. Furthermore, multiplication could be the dominant operation, depending on the application.

There are many hand-held products that include digital signal processing (DSP), for example, cellular phones and hearing aids. For this type of portable equipment a long battery life time and low battery weight is desirable. To obtain this the circuit must have low power consumption.

Constant Multiplication

Multiplication with a constant is commonly used in DSP circuits, such as digital filters. It is possible to use shift-and-add operations to efficiently implement this type of multiplication, i.e., shifts, adders and subtractors are used instead of a general multiplier. As the complexity is similar for adders and subtractors both will be referred to as adders, and adder cost will be used to denote the total number of adders/subtractors. A serial shift operation requires one flip-flop, as seen in Fig. 1.1 (c),hence, the number of shifts is referred to as flip-flop cost.

Single-Constant Multiplication

The general design of a multiplier is shown in Fig. 1.2. The input data, X, is multiplied with a specific coefficient, α, and the output, Y, is the result.

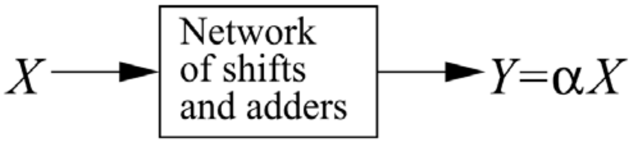

Fig. 1.2 The principle of single-constant multiplication.

However, multipliers can in many cases be implemented more efficiently using other different structures that require fewer operations. Most existing work has focused on minimizing the added cost, while shifts are assumed free as they can be hardwired in the implementation. This is true for bit-parallel arithmetic. However, in serial arithmetic shift operations require flip-flops, and therefore have to be taken into account.

Multiple-Constant Multiplication

In some applications one signal is to be multiplied with several coefficients, as shown in Fig. 1.3. An example of this is the transposed direct form FIR filter where a multiplier block is used, as marked by the dashed box in Fig. 1.1 (b). A simple method to realize multiplier blocks is to implement each multiplier separately, for example, using the CSD representation. However, multiplier blocks can be effectively implemented using structures that make use of redundant partial results between the coefficients, and thereby reduce the required number of components.

This problem has received considerable attention during the last decade and is referred to as multiple-constant multiplication (MCM). The MCM algorithms can be divided into three groups based on the operation of the algorithm: subexpression sharinggraph based and difference methods. Most work has focused on minimizing the number of adders. However, for example, logic depth andpower consumption have also been considered. An algorithm that considers the number of shifts may yield digit-serial filter implementations with smaller overall complexity.

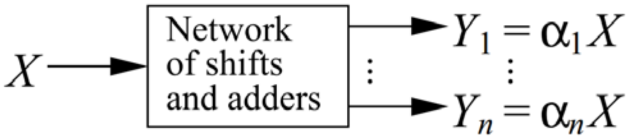

Fig. 1.3 The principle of multiple-constant multiplication.

Applications of Constant Multiplication

Multiplication by a set of constants occurs when multiplying by a constant vector or a constant matrix. For example, the dot product $a{\cdot}b$ gives the scalar projection of a on to b (or vice versa). Multiplication by a constant matrix is nothing more than performing the dot product between several constant vectors (which collectively form a matrix) and a variable vector (the elements of this vector are the inputs). Multiplying by a constant matrix can thus be regarded as a linear transformation of coordinates, which is used in many applications. For example, the conversion from the RGB (red, green, blue) color space to the YUV color space (Y represents brightness s, U and V represent Chroma) involves multiplication by a constant 3x3 matrix. Because the human eye is more sensitive to brightness than coloring (chroma), we can compress the information in the U and V components with only a minor perceived distortion. This is exploited in JPEG and MPEG for compressing images and video, respectively. However, most display devices require color to be provided in the RGB format, hence the need for color space conversion. Any application which involves a predefined linear transformation of coordinates will use multiplication by a set of constants.

Types of Multiplier

Depending on the requirements considering adder cost, flip-flop cost, and pipelining, different multiplier types can be defined. The types that will be discussed are described in the following.

• *CSD – Canonic Signed-Digit multiplier*

Multiplier based on the CSD representation.

• *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,

• *MAG – Minimum Adder Graph multiplier*

Graph multiplier that is based on any of the topologies in Fig. 1.3 and, for any given coefficient, has the lowest possible adder cost.

• *CSDAG – CSD Adder Graph multiplier*

Similar to the MAG multiplier, but may use the same number of adders as corresponding CSD/MSD multiplier, and can by that lower the flip-flop cost.

• *PL MAG/PL CSDAG – Pipelined graph multiplier*

In a pipelined bit-serial graph multiplier, there is at least one intermediate flip-flop between adders. This property, which is always obtained for CSD/MSD multipliers, gives high throughput.

II. LETERATURE SURVEY

SR. NO.	TITLE	AUTHORS	YEAR	METHODOLOGY
1	Fine-Grained Critical Path Analysis and Optimization for Area-Time Efficient Realization of Multiple Constant Multiplications	X. Lou, Y. J. Yu and P. K. Meher	March 2015	In this research work, critical path of multiple constant multiplications (MCM) block is analyzed precisely and optimized for high-speed and low-complexity implementation.
2	Design of high-speed multiplier less filters using a non-recursive signed common subexpression algorithm	M. Martinez-Peiro, E. I. Boemo and L. Wanhammar	Mar 2002.	A new algorithm called no recursive signed common sub expression elimination (NR-SCSE) is discussed
3	A new algorithm for elimination of common subexpressions	R. Pasko, P. Schaumont, V. Derudder, S. Vernalde and D. Durackova	Jan 1999	A new solution of the multiple constant multiplication problem based on the common subexpression elimination technique is presented.
4	ILP modelling of the common subexpression sharing problem	O. Gustafsson and L. Wanhammar	2002	Modelling the sub expression sharing problem using integer linear programming (ILP) an optimal solution can be found.
5	Contention resolution algorithm for common subexpression elimination in digital filter design	FeiXu, Chip-Hong Chang and Ching-Chuen Jong	Oct. 2005	a new algorithm, called contention resolution algorithm for weight-two sub expressions (CRA-2)

X. Lou, Y. J. Yu and P. K. Meher, [1] in this research work, critical path of multiple constant multiplication (MCM) block is analyzed precisely and optimized for high-speed and low-complexity implementation. A delay model based on signal propagation path is proposed for more precise estimation of critical path delay of MCM blocks than the conventional adder depth and the number of cascaded full adders. A dual

objective configuration optimization (DOCO) algorithm is developed to optimize the shift-add network configuration to derive high-speed and low-complexity implementation of the MCM block for a given fundamental set along with a corresponding additional fundamental set. A genetic algorithm (GA)-based technique is further proposed to search for optimum additional fundamentals. In the evolution process of GA, the DOCO is applied to each searched additional fundamental set to optimize the configuration of the corresponding shift-add network. Experimental results show that the proposed GA-based technique reduces the critical path delay, area, power consumption, area delay product and power delay product by 32.8%, 4.2%, 5.8%, 38.3%, and 41.0%, respectively, over other existing optimization methods.

M. Martinez-Peiro, E. I. Boemo and L. Wanhammar, [2] In this work, a new algorithm called no recursive signed common sub expression elimination (NR-SCSE) is discussed, and several applications in the area of multiplier less finite-impulse response (FIR) filters are developed. While the recursive utilization of a common sub expression generates a high logic depth into the digital structure, the NR-SCSE algorithm allows the designer to overcome this problem by using each sub expression once. The research work presents a complete description of the algorithm, and a comparison with two other well-known options: the graph synthesis, and the classical common sub expression elimination technique. Main results show that the NR-SCSE implementations of several benchmark circuits offer the best relation between occupied area and logic depth respect to the previous values published in the technical literature.

R. Pasko, P. Schaumont, V. Derudder, S. Vernalde and D. Durackova, [3] The problem of an efficient hardware implementation of multiplications with one or more constants is encountered in many different digital signal-processing areas, such as image processing or digital filter optimization. In a more general form, this is a problem of common sub expression elimination, and as such it also occurs in compiler optimization and many high-level synthesis tasks. An efficient solution of this problem can yield significant improvements in important design parameters like implementation area or power consumption. In this research work, a new solution of the multiple constant multiplication problem based on the common subexpression elimination technique is presented. The performance of their method is demonstrated primarily on a finite-duration impulse response filter design. The idea is to implement a set of constant multiplications as a set of add-shift operations and to optimize these with respect to the common sub expressions afterwards. Authors show that the number of add/subtract operations can be reduced significantly this way. The applicability of the presented algorithm to the different high-level synthesis tasks is also indicated. Benchmarks demonstrating the algorithm's efficiency are included as well.

O. Gustafsson and L. Wanhammar, [4] Subexpression sharing is an important implementation issue when one data is multiplied with many constants or a sum of products is computed. By modelling the sub expression sharing problem using integer linear programming (ILP) an optimal solution can be found. Further, the model can be directly incorporated with the design of algorithms that have linear design constraints, e.g., linear-phase FIR filters. The proposed method is compared with previously reported algorithms. It produces better results than other sub expression sharing methods, even though it is still not comparable with the optimal method based on graph representation. However, the possibility to expand the ILP model beyond sub expression sharing is discussed. This would then produce identical results to the optimal adder graph method.

FeiXu, Chip-Hong Chang and Ching-Chuen Jong,[5] In this research work, a new algorithm, called contention resolution algorithm for weight-two subexpressions (CRA-2), based on an ingenious graph synthesis approach has been developed for the common subexpression elimination of the multiplication block of digital filter structures. CRA-2 provides a leeway to break away from the local minimum and the flexibility of varying optimization options through a new admissibility graph. It manages two-bit common subexpressions and aims at achieving the minimal logic depth as the primary goal. The performances of their proposed algorithm are analyzed and evaluated based on benchmarked finite-impulse-response filters and randomly generated data. It is demonstrated that CRA-2 achieves the shortest logic depth with significant reduction in the number of logic operators compared with other reported algorithms.

III. PROBLEM IDENTIFICATION

The critical path of MCM blocks are analyzed based on the signal path and a fine-grained delay model for CPD estimation had been proposed. Based on precise estimate of critical path, authors have proposed an algorithm named DOCO to optimize the shift-add network configuration of MCM blocks for the reduction of CPD complexity subject to an additional fundamental set. In order to find the optimum additional fundamentals for a given fundamental set, a GA-based search method is proposed. The DOCO algorithm has been adopted in the proposed GA-based technique to optimize the shift-add network configurations.

IV. CONCLUSION

The primary objective of this review has been to present a new type of partial product generation algorithm (Redundant Booth), to reduce the design to practice, and to show through study and analysis that this algorithm is competitive with other more commonly used algorithms when used for high performance implementations. That can minimize the energy consumption per operation for the arithmetic parts of DSP circuits, such as digital filters. More specific, the focus is on single- and multiple-constant multiplication using serial arithmetic. The possibility to reduce the complexity and energy consumption is investigated. The main difference between serial and parallel arithmetic, which is of interest here, is that a shift operation in serial arithmetic require a flip-flop, while it can be hardwired in parallel arithmetic.

REFRENCES

[1] X. Lou, Y. J. Yu and P. K. Meher, "Fine-Grained Critical Path Analysis and Optimization for Area-Time Efficient Realization of Multiple Constant Multiplications," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 62, no. 3, pp. 863-872, March 2015.

[2] M. Martinez-Peiro, E. I. Boemo and L. Wanhammar, "Design of high-speed multiplierless filters using a nonrecursive signed common subexpression algorithm," in IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, vol. 49, no. 3, pp. 196-203, Mar 2002.

[3] R. Pasko, P. Schaumont, V. Derudder, S. Vernalde and D. Durackova, "A new algorithm for elimination of common subexpressions," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 18, no. 1, pp. 58-68, Jan 1999.

[4] O. Gustafsson and L. Wanhammar, "ILP modelling of the common subexpression sharing problem," Electronics, Circuits

and Systems, 2002. 9th International Conference on, pp. 1171-1174 vol.3,2002.

[5] FeiXu, Chip-Hong Chang and Ching-Chuen Jong, "Contention resolution algorithm for common subexpression elimination in digital filter design," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 52, no. 10, pp. 695-700, Oct. 2005.

[6] L. Aksoy, E. Costa, P. Flores and J. Monteiro, "Optimization of area under a delay constraint in digital filter synthesis using SAT-based integer linear programming," 2006 43rd ACM/IEEE Design Automation Conference, San Francisco, CA, 2006, pp. 669-674.

[7] A. Hosangadi, F. Fallah and R. Kastner, "Reducing hardware complexity of linear DSP systems by iteratively eliminating two-term common subexpressions," Proceedings of the ASP-DAC 2005. Asia and South Pacific Design Automation Conference, 2005., 2005, pp. 523-528 Vol. 1.

[8] L. Aksoy, E. da Costa, P. Flores and J. Monteiro, "Exact and Approximate Algorithms for the Optimization of Area and Delay in Multiple Constant Multiplications," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 27, no. 6, pp. 1013-1026, June 2008.

[9] D. R. Bull and D. H. Horrocks, "Primitive operator digital filters," in IEE Proceedings G - Circuits, Devices and Systems, vol. 138, no. 3, pp. 401-412, June 1991.

[10] A. G. Dempster and M. D. Macleod, "Constant integer multiplication using minimum adders," in IEE Proceedings - Circuits, Devices and Systems, vol. 141, no. 5, pp. 407-413, Oct 1994.

[11 O. Gustafsson, A. G. Dempster and L. Wanhammar, "Extended results for minimum-adder constant integer multipliers," Circuits and Systems, 2002.ISCAS 2002. IEEE International Symposium on, 2002, pp. I-73-I-76 vol.1.

[12] A. G. Dempster and M. D. Macleod, "Use of minimum-adder multiplier blocks in FIR digital filters," in IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, vol. 42, no. 9, pp. 569-577, Sep 1995.

[13] F. Xu, J. J. Chen, C. H. Chang, and C. C. Jong, “A modified reduced adder graph algorithm for multiplier block minimization in digital filters,” in Proc . IEEEAs ia - Pacific Conf . Circuits Syst ., Dec. 2004, vol. 2, pp. 705–708.

[14] R. I. Hartley, “Optimization of canonic signed digit multipliers for filter design,” in Proc . IEEE Int .Symp.Circuits Syst., Singapore, Jun. 1991, vol. 4, pp. 1992–1995.

[15] D. Bull and D. Horrocks, “Primitive operator digital filters,” in IEE Proc. Inst. Elec. Eng. Circuits , Devices, Syst., Jun. 1991, vol. 138, no. 3, pp. 401–412.