Design of Efficient Graph Based Algorithm with Modified Carry save Adder

Mr.M.Sivakumar, Mr.R.Seshadri, Dr.S.Ramakrishnan
PG Scholar, Applied electronics, Arunai Engineering College, Tiruvannamalai, India
Assistant Professor, Department of ECE, Arunai Engineering College, Tiruvannamalai, India
Professor, Department of IT, Sakthi Mariamman Engineering College, Chennai, India

***Abstract*—** In the field of digital signal processing, many well-organized algorithms and architectures are introduced for the design of low complexity bit parallel multiple constant multiplication (MCM) operation which take over the complica-tion of many digital signal processing systems. Little courtesy has been given to the digit serial MCM intention. In this paper we discourse the difficulty of optimizing the gate level delay in digit serial MCM designs by modifying the carry save adder and implementing it to the graph based (GB) algorithm. Here delay and power can be highly reduced albeit at the cost of an increased area. Experimental effects show the capability of the proposed optimization algorithms and of the digit-serial MCM architectures in the design of digit-serial MCM operations and finite impulse response filters. The proposed method reduces the delay to achieve higher speed than the existing method.

***Index Terms*—** Graph based (GB) algorithm, modified carry save adder (MCSA), finite impulse response (FIR) filters, gate-level delay optimization, multiple constant multiplications.

I. INTRODUCTION

General process for carrying multiplication by a constant multiplication is by consuming an arrangement of shift and adds. It is conceivable to modify this method by permitting subtraction as well. This method reduces the shift and adds op-eration needed for the design of finite impulse response (FIR) filters. These filters are digital filters whose response to an unit impulse (unit sample function) is finite in interval. This is in divergence to infinite impulse response (IIR) filters. Finite impulse response (FIR) filters are great importance in digital signal processing system.

The direct and transposed-form finite impulse response (FIR) filter implementations are illustrated in Fig. 1.(a) and (b) correspondingly. Although both architectures have similar diff-iculties in hardware, the transposed form is generally preferred because of its higher performance and power efficiency. The multiplication of filter coefficients with filter inputs is recog-nized has major effects on the complexity and performance of the design because a large number of constant multiplications are required. This is commonly recognized as the multiple constant multiplications operation and performance blockage in many other DSP systems such as the fast Fourier transforms, discrete cosine transforms (DCTs) and error-correcting codes.

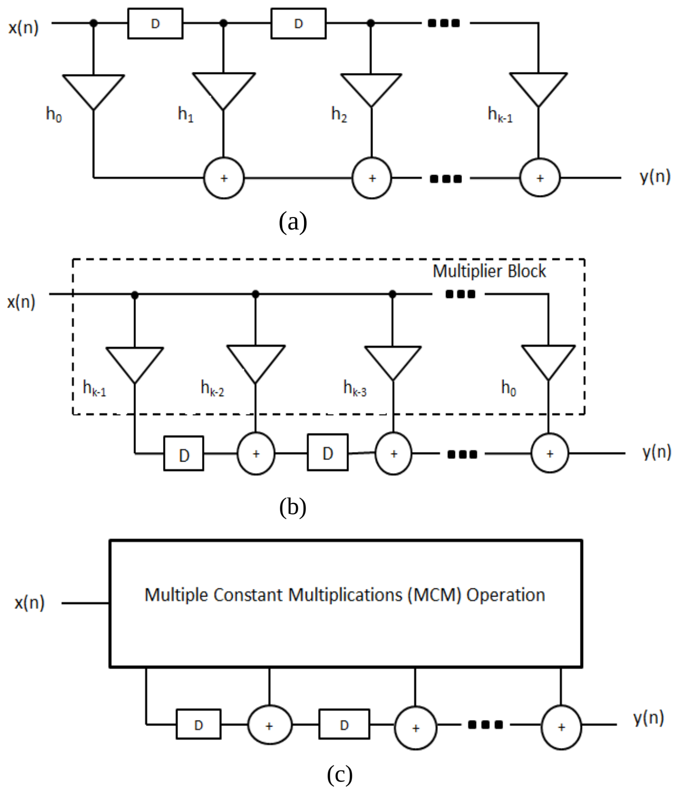

Fig.1.FIR filter implementations. (a) Direct form. (b) Transposed form with generic multipliers. (c) Transposed form with MCM block

Even though area, delay and power efficient multiplier architectures have been proposed, the full flexibility of a mul-tiplier is not necessary for the constant multiplications, since filter coefficients are fixed and determined beforehand by the DSP algorithm. Hence filter coefficients multiplication with input data is generally implemented under a shift-adds archit-ecture where each constant multiplication is realized using a addition/subtraction and shift operation in an MCM operation.

For shift-adds implementation of constant multiplications, a straight forward method and is generally known as digit based recoding, initially defines the constants in binary. Then for each “1” in the binary representation of the constant, according to its bit position, it shifts variables and adds up the shifted variables to obtain the result. As a simple example, consider the constant multiplication 29x and

43x. Their decomposition in binary are listed as follows:

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

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

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

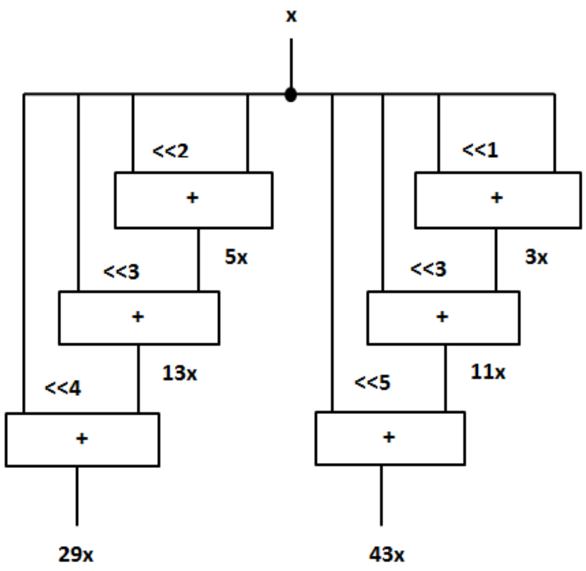

Fig. 2.(a)Shift adds implementation of 29x and 43x.

From the above figure, the shift-adds implementations of 29x and 43x which requires six addition operations for the constant multiplications. The graph based algorithm which is introduced to reduce the addition operations in the shift adds implementations of constant multiplications. These 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 shift adds imple-mentation of constant multiplication. These name implies that the operation is minimized by the reduction of graph accor-ding to the shift adds implementations of constant multipli-cations in the FIR filters. From this technique the number of repeated sub-expressions can be reduced maximum and produce the efficient result as compared with shift and adds techniques. To ensure this condition Fig.2(b) is compared with Fig.2(a). The graph based algorithm finds a better solution with minimum number of operations by sharing the common partial product 7x in both multiplications. The graph based algorithm for the implementation of 29x and 43x is given in following Fig. 2(b).

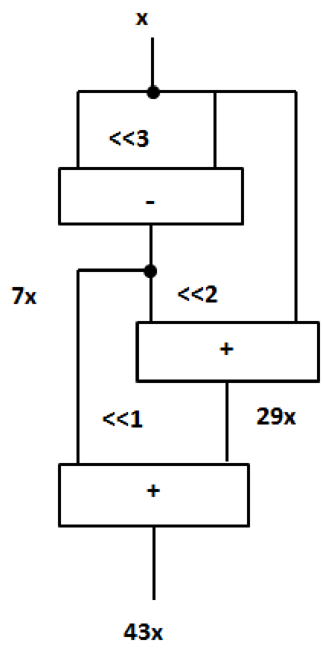

Fig. 2.(b)Graph based algorithm of 29x and 43x.

From the above figure addition operation is reduced by introducing subtraction operation and sharing the sub-expression of constant multiplication. Note that the partial product 7x cannot be extracted from the binary representation of 43x in the shift and adds implementation of constant multiplication. In this project we propose the modified carry save adder in the graph based algorithm to increase the gate level speed of constant multiplications of FIR filters. Previously adder is used in the graph based algorithm for the operation of constant multiplications in FIR filters. The proposed method comparatively shows the best result in the gate level delay of constant multiplications. Then for the implementaion of carry save adder in the graph based algorithm is made efficient compared with the existing method of graph based algorithm.

Experimental results on a comprehensive set of instances shows that the solutions of algorithms introduced in this project lead to significant improvements of speed in the graph based algorithm designs, compared to those obtained using the algorithms designed for the multiple constant multiplication (MCM) problem. A large number of constant multiplications are required when filter coefficients are multiplied with the filter inputs, these large number of constant multiplications are refered as multiple constant multiplication (MCM). The backround concepts explains about the implementation of modified carry save adder in the graph based algorithm and discusses about the result compared with the implementation of adder in the graph based algorithm. Finally the result gives the better solution.

The rest of this paper proceeds as follows. Section II gives the background concepts. The experimental results are presented in Section III and conclusions in Section IV.

II. BACKGROUND

(A) Number Representation

The binary representation decomposes a number in a set of additions of powers of 2. The representation of numbers using a signed digit system makes use of positive and negative digits, {1, 0, −1}. The canonical signed digit (CSD) representation is a signed digit system that has a unique representation for each number and verifies the following main properties: 1) two nonzero digits are not adjacent; and 2) the number of nonzero digits is minimum. Any n digit number in CSD has at most [(n+1)/2] nonzero digit and, on average, the number of nonzero digits is reduced by 33% when compared to binary. The minimal signed digit (MSD) representation is obtained by dropping the first property of the CSD representations under MSD, including its CSD representation, but all with a minimum number of nonzero digits.

Consider the constant 23 defined in six bits. Its binary representation 010111 includes four nonzero digits it is represented as 101001 in CSD, and both 101001 and 011001 denote 23 in MSD using three nonzero digits (where 1 stands for -1).

(B) Modified Carry Save Adder

The 16-bit conventional CSA is shown in Fig 3. It has 17-half adders and 15-full adders. Since the ripple carry adder (RCA) is used in the final stage, this structure yields large carry propagation delay. To reduce this delay, the final stage of CSA is divided into 5 groups as shown in Fig 4.

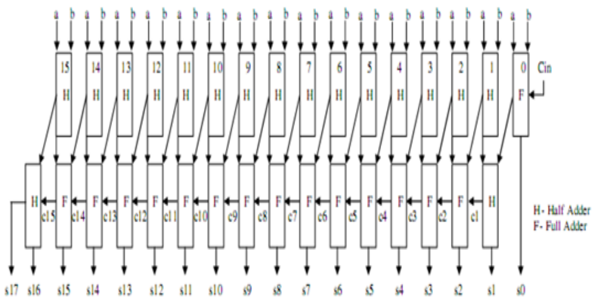

Fig. 3. Conventional carry save adder

The first group includes n 2log 1+ -bit value and other groups includes n 2 log –bit value, where n is the bit size of the adder. The first group of output s[4:0] are directly assigned as the final output; the second group {c7,x[7:5]} manipulates the partial result by considering c4 is 0; the third group {c10,x[10:8]} manipulates the partial result by considering c7 is 0; the fourth group {c13,x[13:11]} manipulates the partial result by considering c10 is 0 and the fifth group x[17:14] manipulates the partial result by considering c13 is 0. By this method carry save adder is modified for GB algorithm.

The divided groups are listed below:

i). {c4,s[4:0]}
ii). {c7,x[7:5]}
iii). {c10,x[10:8]}
iv). {c13,x[13:11]}
v). x[17:14]

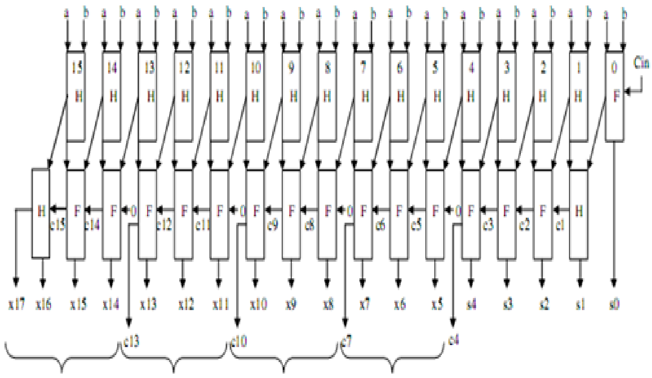

Fig. 4. Modified carry save adder

Depending on c4 of the first group, the second group mux gives the final result without the carry propagation delay from c4 to c7; depending on c7 of the second group final result, the third group mux gives the final result without the carry propagation delay from c7 to c10; depending on c10 of the third group final result, the fourth group mux gives the final result without the carry propagation delay from c10 to c13 and depending on c13 of the fourth group final result, the fifth group mux gives the final result without the carry propagation delay from c13 to s17.

The main advantage of this logic is that each group computes the partial results in parallel and the multiplexers are ready to give the final result “immediately” with the minimum delay of the mux. When the C_{in} of each group arrives, the final result will be determined “immediately”. Thus the maximum delay is reduced in the carry propagation path. This same logic has used for 32 and 64-bit adder structures to achieve higher speeds. The area indicates the total cell area of the design and the total power is sum of leakage power, internal power, net power and dynamic power. The proposed result shows that the CLA and CSA have reduced area than MCSA. But the speed of the MCSA architecture has significantly improved when compared to the conventional CSA and CLA. In graph based algorithm MCSA is used for the substitution of addition operation to increase the speed and power, because MCSA is more advance than CSA and CLA.

III. EXPERIMENTAL RESULT

The experimental result is given for the implementation of modified carry save adder in the graph based algorithm. Comparision of modified carry save adder with the normal adder is shown in table 1. The synthesis result is given below for the implementation of normal adder in the graph based algorithm. These results which gives the clear idea about the graph based algorithm.

SYNTHESIS RESULT

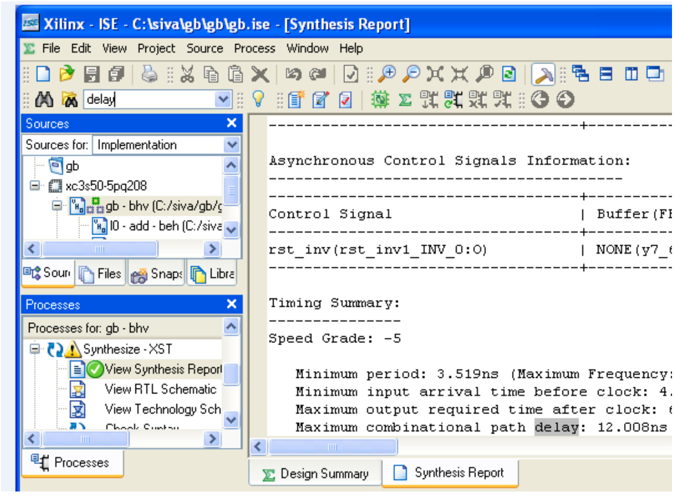

Fig. 5 synthesis result for delay.

The synthesis result for delay is given in the above fig. 5.
It shows the required delay for the implementation of normal adder in graph based algorithm. From the above result, to increase the speed by reducing delay, the modified carry save adder is used in graph based algorithm. The power can also be redused in MCSA.The flow chart and the table is given for the comparision of experimental results for normal adder with modified carry save adder in the implementation of graph based (GB) algorithm is shown as follows.

Table. 1

Work	GB algorithm using normal adder	GB algorithm using Modified carry save adder
power	0.222	0.215
Delay	12.008ns	6.216ns

From table.1 the comparision of an implementation of modified carry save adder in the graph based algorithm with the normal adder is shown. Here we can reduce the delay and increase the speed and we can also reduce the power as shown in above table. From the flow chart the reduction of power and delay can be achieved greatly as shown. The graph based algorithm is given for the efficient carry save adder and by the reduction of delay we can improve the speed of the FIR Filter from the implementation of graph based algorithm. The proposed result shows that the CLA and CSA have reduced area than MCSA in graph based (GB) algorithm. But the speed of the MCSA architecture has been significantly improved and has the least value of power-delay product compared to the conventional CSA and CLA.

Flow chart.1

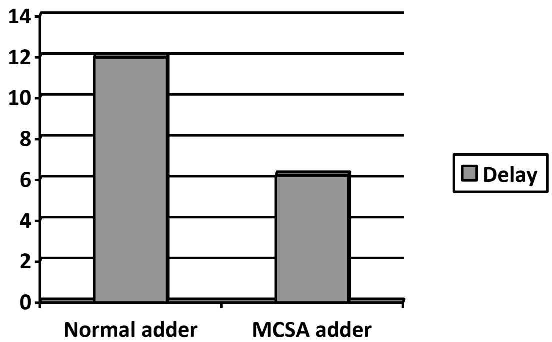

Fig. 6. Flow chart of comparision for delay

Flow chart .2

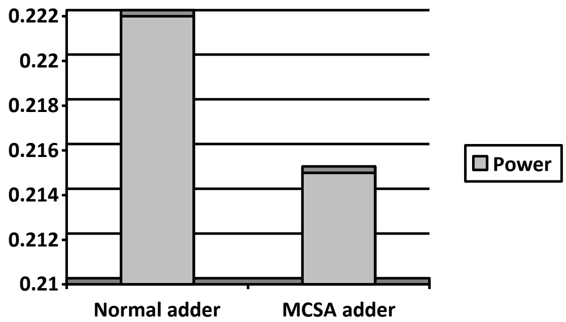

Fig.7. Flow chart of comparision for power

From the above given flow chart .1 and .2. Delay and power can be reduced as much as possible compared with the implemenation of normal adder in the graph based algorithm.

IV CONCLUSION

Thus in our project, by the implementation of modified carry save adder in the graph based algorithm, the delay is reduced maximum, and the power is also reduced as much as possible. The graph based (GB) algorithm is given for the improvement of shift and adds implementation of the constant multiplications. The graph based (GB) algorithm reduces the number of shift and adds operation. From the proposed technique by implementing the modified carry save adder (MCSA) in the graph based algorithm, as shown in flow chart, the delay is reduced maximum from 12.008ns to 6.216ns to achieve higher speeds and power is reduced from 0.222 to 0.215, from the normal adder as given in table.1. The proposed algorithm have been synthesized and implemented using xilinx 10.1. The speed of the MCSA architecture has significantly improved and power is also redused in the experimental result.

REFERENCES

[1] 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.

[2] C. Wallace, "A suggestion for a fast multiplier," IEEE Trans. Electron. Compute., vol. 13, no. 1, pp. 14–17, Feb. 1964.

[3] 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.

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

[5] K. Parhi, "A systematic approach for design of digit-serial signal processing architectures," IEEE Trans. Circuits Syst., vol. 38, no. 4, pp. 358–375, Apr. 1991.

[6] 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.

[7] 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] 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.

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

[11] P. Barth, "A Davis-Putnam based enumeration algorithm for linear pseudo-Boolean optimization," Max-Planck-Institut für Informatik, Saarbrücken, Germany, Tech. Rep. MPI-I-95-2-003, 1995.

[12] 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.

[13] 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.

[14] T. Larrabee, "Test pattern generation using Boolean satisfiability," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 11, no. 1, pp. 4–15, Jan. 1992.

[15] Y.-H. Ho, C.-U. Lei, H.-K. Kwan, and N. Wong, "Global optimization of common subexpressions for multiplierless synthesis of multiple constant multiplications," in Proc. ASPDAC, 2008, pp. 119–124.

[16] Levent aksoy, Cristiano lazzari, "Design of digit-serial FIR filters: Algorithm, Architectures, and a CAD tool," IEEE Trans. On very large scale integration (VLSI) systems, vol.21. No.3,march 2013.