LETTER

CSD-Based Programmable Multiplier Design for Predetermined Coefficient Groups

Yong-Eun KIM[†a)], **Kyung-Ju CHO**[††b)], *Nonmembers*, **Jin-Gyun CHUNG**[†c)], *Member*, *and* **Xinming HUANG**[†††d)], *Nonmember*

SUMMARY An efficient multiplier design method for predetermined coefficient groups is presented based on the variation of canonic signed digit (CSD) encoding and partial product sharing. By applications to radix-2^4 FFT structure and the pulse-shaping filter design used in CDMA, it is shown that the proposed method significantly reduces the area, propagation delay and power consumption compared with previous methods.

key words: *programmable CSD multiplier, partial products depth reduction, partial products sharing*

1. Introduction

In some DSP applications such as FFT, multiplications are performed only with a few predetermined coefficients which are time-varying in periodical order. In these applications, multipliers should have programmability. When a few coefficients share a multiplier, modified Booth encoding, which halves the number of partial products, is generally used. If the multiplier coefficient is a constant, the coefficient can be encoded such that it contains the fewest number of nonzero digits, which can be accomplished using CSD to reduce the area and power consumption [1], [2].

Subexpression elimination can be used to efficiently implement constant multipliers. The multiple constant multiplication (MCM) problem determines how subexpression elimination can be applied to a set of constant multipliers so that the number of additions required for the implementation is minimized [3]. However, subexpression elimination provides an efficient solution only when the multiplications between the input (as multiplicand) and each multiplier in the coefficient group need to be performed at the same time. In other words, subexpression elimination cannot be used if the multiplications are performed one after another in a periodical manner (e.g., multiplications in a pipelined FFT). Based on the modified Booth encoding, a multiplier design method for a group of predetermined coefficients has been recently proposed in [4].

Manuscript received May 15, 2009.
Manuscript revised July 9, 2009.

[†]The authors are with the Division of Electronics & Information Engr., Chonbuk National University, 664-14, Deokjindong 1-ga Deokjin-gu, Jeonju, 561-756, Korea.
[††]Corresponding author. The author is with Korea Association of Aids to Navigation (KAAN), IT Castle2 12F, Kasan Geumcheon-ku, Seoul, 153-768 Korea.
[†††]The author is with the Dept. of Electrical & Computer Engr., Worcester Polytechnic Institute, Worcester, MA, USA.
a) E-mail: dosto@chonbuk.ac.kr
b) E-mail: kjcho@chonbuk.ac.kr
c) E-mail: jgchung@chonbuk.ac.kr
d) E-mail: xhuang@ece.wpi.edu

DOI: 10.1587/transfun.E93.A.324

2. Partial Products Depth Reduction

Figure 1 shows the pipelined structure of N-point radix-2^4 FFT [5]. In the first and the third multiplication blocks, three coefficients {cos($\pi/8$), cos($\pi/4$), and sin($\pi/8$)} are multiplied by an input complex signal in periodical order as

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 \\ \cos(\pi/8) - j\sin(\pi/8) \\ \cos(\pi/4) - j\sin(\pi/4) \\ \sin(\pi/8) - j\cos(\pi/8) \end{bmatrix} (x_r + jx_i). \tag{1}$$

In general, the multiplications in (1) can be implemented using a programmable multiplier such as modified Booth multiplier.

If the coefficient word-length is W, the number of partial product (PP) rows obtained by modified Booth algorithm is $W/2$. To further reduce the number of PP rows, we propose a grouping algorithm for multiplication coefficients:

1. When the number of given 2's complement coefficients is N_c with the word length of N_w bits, the coefficients are arranged as an $N_c \times N_w$ table.
2. The coefficients in the table are converted to CSD representations.
3. Starting from the first column, a group is defined such that each row in a group contains at most one nonzero digit. A group should contain as many columns as possible so that the number of groups is minimized.

By applying the proposed grouping algorithm to these three coefficients in (1) with N_w=14, the CSD coefficient table with 5 groups is obtained as shown in Table 1. Each group G_i generates a corresponding partial product P_i. Thus, the multiplication result (Y) is obtained as

$$Y = P_4 + 2^{-2}P_3 + 2^{-4}P_2 + 2^{-6}P_1 + 2^{-8}P_0 \tag{2}$$

In Table 1, the number of PP rows required by the proposed algorithm is only 5, while modified Booth encoding requires 7 (=N_w/2) PP rows. Thus the proposed grouping algorithm reduces the number of PP rows by 2, which leads to lower power consumption and higher speed as can be seen from the simulation results in Table 4. Each group includes at least two columns by the proposed algorithm since CSD coding does not allow any consecutive nonzero digits. Thus, the number of PP rows generated by the proposed algorithm is always less than or equal to that of the modified Booth encoding.

3. Partial Products Sharing

If the nonzero digit locations of two groups are the same as in G_4 and G_1 in Table 1, the two groups can share PP generation circuits. The sign difference in the first rows of G_4 and G_1 can be taken care of later by additional complementing circuits. For any row in a group that contains only 0's, the corresponding PP is 0. In this case, the zero digits in the row can be changed to nonzero digits to share the partial product generation circuits, since the output value can be easily changed back to 0 using a control signal. The all zero rows of G_3 and G_2 in Table 1 can be changed as shown in Table 2, where the changed digits are denoted using parentheses. Figure 2 shows the partial product bit generation circuit with control signals, where S_i, N_i and Z_i are shift, negation, and zero control signals, respectively.

Table 3 shows a new representation of each group in Table 2 using control signals. Two changed digits in Table 2 are compensated by Z_3 and Z_2 signals. If the N_i signals in a group are all 1's, the XOR gate in Fig. 2 can be simply replaced by an inverter. Also, if the Z_i signals in a group are all 1's, the AND gate in Fig. 2 can be eliminated. In addition, if $S_i = S_j$, the input shift patterns are the same and the shifted inputs can be shared between G_i and G_j. Since the shifted inputs are shared, it is sufficient to store either S_i or S_j. Notice that S_4, S_3, S_2 and S_1 are the same in Table 3. Thus the shifted inputs are shared among groups G_4, G_3,G_2 and G_1 and we only need to store S_4. (For the shift operation by S_0, refer to the lower part of Fig. 3.)

In conventional approach, the coefficient look-up table (LUT) has 14 columns if the coefficient word-length is 14 bits. As an example, if a modified Booth multiplier is used, all the coefficients are represented in 14 bit 2's complement format and are stored in an LUT. Then, for each multiplication operation, the corresponding stored coefficient is read from the LUT and encoded using modified Booth algorithm for the multiplication. On the other hand, in the proposed approach, we store the encoded signals (not the coefficients in 2's complement format). Thus, if an N_i (or Z_i) signal always has a fixed encoded value (e.g., N_4, Z_4, N_3, N_2, Z_1, and Z_0 in Table 3), the N_i (or Z_i) signal can be implemented

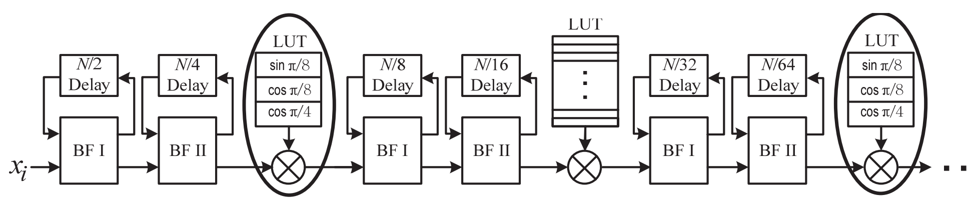

Fig. 1 Pipelined structure of radix-2^4 FFT.

Table 1 CSD coefficient table.

Column	13	12	11	10	9	8	7	6	5	4	3	2	1	0
cos(π/8)	1	0	0	0	-1	0	-1	0	0	1	0	0	0	0
cos(π/4)	1	0	-1	0	-1	0	1	0	1	0	0	0	0	0
sin(π/8)	0	1	0	-1	0	0	0	1	0	0	0	0	0	-1
Group	G_4		G_3		G_2		G_1		G_0					

Table 2 Modification of '0' in CSD coefficients.

Column	13	12	11	10	9	8	7	6	5	4	3	2	1	0
cos(π/8)	1	0	(-1)	0	-1	0	-1	0	0	1	0	0	0	0
cos(π/4)	1	0	-1	0	-1	0	1	0	1	0	0	0	0	0
sin(π/8)	0	1	0	-1	0	(-1)	0	1	0	0	0	0	0	-1
Group	G_4		G_3		G_2		G_1		G_0					

Table 3 New representation of CSD coefficients using control signals.

Group	G_4			G_3			G_2			G_1			G_0		
Control signals	S_4	N_4	Z_4	S_3	N_3	Z_3	S_2	N_2	Z_2	S_1	N_1	Z_1	S_0	N_0	Z_0
cos(π/8)	1	0	1	1	1	0	1	1	1	1	1	1	01	0	1
cos(π/4)	1	0	1	1	1	1	1	1	1	1	0	1	10	0	1
sin(π/8)	0	0	1	0	1	1	0	1	0	0	0	1	00	1	1

Table 4 Simulation results.

Application	Method	Area (cell)	Propagation delay (*ns*)	Power (*u*W)
Radix-2^4 FFT	Booth Multiplier	12827 (1)	13.47 (1)	23.30 (1)
	[4]	10755 (0.84)	13.06 (0.97)	17.03 (0.73)
	Proposed	7577 (0.59)	11.89 (0.88)	12.86 (0.55)
Pulse-shaping filter	Booth Multiplier	46528 (1)	12.45 (1)	74.04 (1)
	[4]	23875 (0.51)	9.91 (0.80)	29.65 (0.40)
	Proposed	17341 (0.37)	9.43 (0.76)	22.56 (0.30)

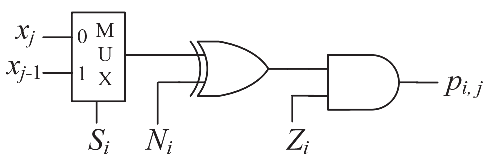

Fig. 2 Partial product bit generation circuit.

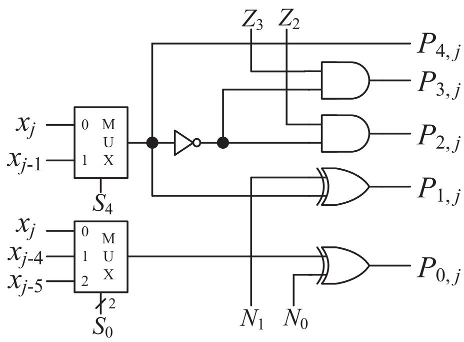

Fig. 3 Partial product generation circuit by Table 3.

directly without storing the values in an LUT. In addition, if some groups have the same shift pattern (e.g., S_4, S_3, S_2 and S_1 in Table 3), it is sufficient to store only one shift pattern. Thus, in Table 3, we need to store only 7 columns of control signals, S_4, S_0 (2 columns), Z_3, Z_2, N_1 and N_0. In this case, LUT size is reduced by 50% compared with conventional approaches. Figure 3 shows the total partial product generation circuit designed using Table 3. The generated PP bits are added using (2).

The sharing algorithm of PP generation circuits can be summarized as follows:

1. For each coefficient group obtained by the coefficient grouping algorithm, determine the nonzero digit location pattern by replacing each (−1) with 1 in the CSD coefficient table corresponding to the group.
2. For each group G_i, modify the element 0's in other groups such that the number of the nonzero digit location patterns equal to that of the group G_i is maximized. Then, the number of the same nonzero digit location patterns is recorded as K_i.
3. Choose the largest K_i. Record the groups sharing the same nonzero digit location pattern with group G_i. Remove the recorded groups from the table.
4. Repeat steps 2 and 3 until the table is empty.

4. Comparison of Performances

To evaluate the performance of the proposed method, the multipliers in radix-2^4 FFT and pulse-shaping filter used in CDMA [4] have been designed and synthesized using MagnaChip 0.18-μm CMOS technology. We employed Wallace tree for addition of partial products and carry look-ahead adder for the vector merging part. Table 4 shows the Synopsys simulation results. In FFT application, compared with the conventional modified Booth multiplier, the proposed method reduces the area, power consumption and propagation delay by 41%, 45% and 12%, respectively. The power consumption was estimated using Synopsys Design_analyzer. Also, in the pulse-shaping filter application, the proposed method reduces the area, power consumption and propagation delay by 63%, 70% and 24%, respectively. In addition, Table 4 shows proposed method performs much better than the modified Booth-based approach for predetermined coefficient groups [4].

5. Conclusions

Based on the proposed grouping and sharing algorithms, this letter presents an efficient multiplier design method for predetermined coefficient groups. The simulation results show that the proposed method can be successfully applied to the multiplication blocks in radix-2^4 FFT and the pulse-shaping filter in CDMA.

References

[1] R. Hartley, "Subexpression sharing in filter using canonic signed digit multipliers," IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., vol.43, no.10, pp.677–688, Oct. 1996.
[2] K.K. Parhi, VLSI Digital Signal Processing Systems, John Wiley & Sons, NY, 1999.
[3] M. Potkonjak, M.B. Srivastava, and A.P. Chandrakasan, "Multiple constant multiplications: Efficient and versatile framework and algorithms for exploring common subexpression elimination," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.15, no.2, pp.151–165, Feb. 1996.
[4] Y.E. Kim, K.J. Cho, and J.G. Chung, "Low power small area modified Booth multiplier design for predetermined coefficients," IEICE Trans. Fundamentals, vol.E90-A, no.3, pp.694–697, March 2007.
[5] J.Y. Oh and M.S. Lim, "New radix-2 to the 4th power pipeline FFT processor," IEICE Trans. Electron., vol.E88-C, no.8, pp.1740–1746, Aug. 2005.