

# Frequency Spectrum Based Low-Area Low-Power Parallel FIR Filter Design

**Jin-Gyun Chung**

*Division of Electronic and Information Engineering, Chonbuk National University, Chonju 561-756, Korea  
Email: jgchung@moak.chonbuk.ac.kr*

**Keshab K. Parhi**

*Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455, USA  
Email: parhi@ece.umn.edu*

*Received 11 July 2001 and in revised form 15 May 2002*

Parallel (or block) FIR digital filters can be used either for high-speed or low-power (with reduced supply voltage) applications. Traditional parallel filter implementations cause linear increase in the hardware cost with respect to the block size. Recently, an efficient parallel FIR filter implementation technique requiring a less-than linear increase in the hardware cost was proposed. This paper makes two contributions. First, the filter spectrum characteristics are exploited to select the best fast filter structures. Second, a novel block filter quantization algorithm is introduced. Using filter benchmarks, it is shown that the use of the appropriate fast FIR filter structures and the proposed quantization scheme can result in reduction in the number of binary adders up to 20%.

**Keywords and phrases:** parallel FIR filter, quantization, fast FIR algorithm, canonic signed digit.

## 1. INTRODUCTION

Finite impulse response (FIR) filters are widely used in various DSP applications. In some applications, the FIR filter circuit must be able to operate at high sample rates, while in other applications, the FIR filter circuit must be a low-power circuit operating at moderate sample rates. The low-power or low-area techniques developed specifically for digital filters can be found in [1, 2, 3, 4, 5, 6, 7].

Parallel (or block) processing can be applied to digital FIR filters to either increase the effective throughput or reduce the power consumption of the original filter. While sequential FIR filter implementation has been given extensive consideration, very little work has been done that deals directly with reducing the hardware complexity or power consumption of parallel FIR filters.

Traditionally, the application of parallel processing to an FIR filter involves the replication of the hardware units that exist in the original filter. If the area required by the original circuit is  $A$ , then the  $L$ -parallel circuit requires an area of  $L \times A$ . Recently, an efficient parallel FIR filter implementation technique requiring a less-than linear increase in the hardware cost was proposed using FFAs (fast FIR Algorithms) [8].

In [9], it was shown that the power consumption of arithmetic units can be reduced if statistical properties of the input signals are exploited. In this paper, based on [10], it is

shown that the hardware cost can be reduced by exploiting the frequency spectrum characteristics of the given transfer function. This is achieved by selecting appropriate FFA structures out of many possible FFA structures all of whom have similar hardware complexity at the word-level. However, their complexity can differ significantly at the bit-level. For example, in narrowband low-pass filters, the signs of consecutive unit sample response values do not change much and therefore their difference can require fewer number of bits than their sum. This favors the use of a parallel structure which requires subfilters which require difference of consecutive unit sample response values as opposed to sum.

In addition to the appropriate selection of FFA structures, proper quantization of subfilters is important for low-power or low hardware cost implementation of parallel FIR filters. It is shown in [5, 6, 7] that if the filter coefficients are first scaled before the quantization process is performed, the resulting filter will have much better frequency-space characteristics. When the quantized filter is implemented, a postprocessing scale factor (PPSF) is used to properly adjust the magnitude of the filter output. In cases where large levels of parallelism are used, the number of required subfilters is large, and consequently the PPSFs can contribute to a significant amount of hardware overhead. In [8], PPSFs are restricted to a set of simple values to reduce the hardware overhead due to PPSFs. Since the original PPSF is replaced with the new simple PPSF

that is the nearest in value, the quantized filter coefficients must also be properly modified. However, this approach is not guaranteed to give optimal quantized coefficients since already quantized coefficients are modified again. To avoid this problem, we propose *look-ahead maximum absolute difference* (LMAD) quantization algorithm, which gives optimal quantized coefficients for a given simple PPSF value.

In Section 2, FFAs are briefly reviewed. Also, frequency spectrum related hardware complexities for different types of FFAs are discussed. Section 3 presents a quantization method suitable for block FIR filters. Section 4 presents several block filter design examples.

## 2. FAST FIR ALGORITHMS

Consider the general formulation of a length- $N$  FIR filter,

$$y_n = \sum_{i=0}^{N-1} h_i x_{n-i} \quad n = 0, 1, 2, \dots, \infty, \quad (1)$$

where  $\{x_i\}$  is an infinite length input sequence and  $\{h_i\}$  are the length- $N$  FIR filter coefficients. Then the polyphase representation of a traditional  $L$ -parallel FIR filter [11] can be expressed as

$$\sum_{i=0}^{L-1} Y_i(z^L) z^{-i} = \sum_{j=0}^{L-1} H_j(z^L) z^{-j} \sum_{k=0}^{L-1} X_k(z^L) z^{-k}, \quad (2)$$

where  $Y_i(z) = \sum_{m=0}^{\infty} z^{-m} y_{mL+i}$ ,  $H_i(z) = \sum_{m=0}^{N/L-1} z^{-m} h_{mL+i}$ ,  $X_i(z) = \sum_{m=0}^{\infty} z^{-m} x_{mL+i}$ , for  $i = 0, 1, \dots, L - 1$ . This block FIR filtering equation shows that the parallel FIR filter can be realized using  $L^2$ -FIR filters of length  $N/L$ . This linear complexity can be reduced using various FFA structures.

### 2.1. $2 \times 2$ ( $L = 2$ ) FFAs

From (2) with  $L = 2$ , we have

$$\begin{aligned} Y_0 + z^{-1}Y_1 &= (H_0 + z^{-1}H_1)(X_0 + z^{-1}X_1), \\ &= H_0X_0 + z^{-1}(H_0X_1 + H_1X_0) + z^{-2}H_1X_1, \end{aligned} \quad (3)$$

which implies that

$$\begin{aligned} Y_0 &= H_0X_0 + z^{-2}H_1X_1, \\ Y_1 &= H_0X_1 + H_1X_0. \end{aligned} \quad (4)$$

Direct implementation of (4) is shown in Figure 1. This structure computes a block of 2 outputs using 4 length  $N/2$  FIR filters and 2 postprocessing additions, which requires  $2N$  multipliers and  $2N - 2$  adders.

If (4) is written in a different form, the  $(2 \times 2)$  FFA0 (FFA-type 0) is obtained,

$$\begin{aligned} Y_0 &= H_0X_0 + z^{-2}H_1X_1, \\ Y_1 &= H_{0+1}X_{0+1} - H_0X_0 - H_1X_1, \end{aligned} \quad (5)$$

where  $H_{i+j} = H_i + H_j$  and  $X_{i+j} = X_i + X_j$ . Implementation of (5) is shown in Figure 2. This structure computes a block of

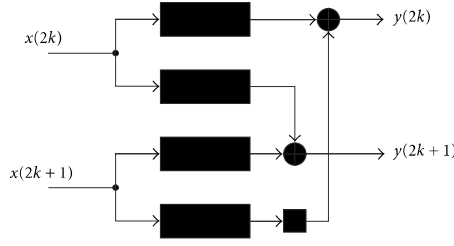


FIGURE 1: Traditional 2-parallel FIR filter.

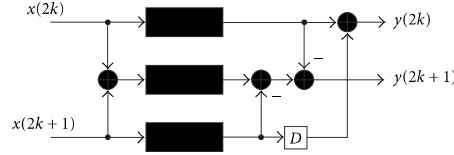


FIGURE 2: 2-parallel FIR filter using FFA0.

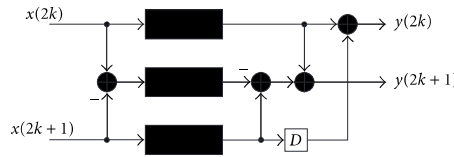


FIGURE 3: 2-parallel FIR filter using FFA1.

2 outputs using 3 length  $N/2$  FIR filters and 4 preprocessing and postprocessing additions, which requires  $3N/2$  multipliers and  $3(N/2 - 1) + 4$  adders.

By a simple modification of (5), the following FFA1 (FFA-type 1) is derived [11],

$$\begin{aligned} Y_0 &= H_0X_0 + z^{-2}H_1X_1, \\ Y_1 &= -H_{0-1}X_{0-1} + H_0X_0 + H_1X_1. \end{aligned} \quad (6)$$

In (6),  $H_{0-1} = H_0 - H_1$  and  $X_{0-1} = X_0 - X_1$ . The structure derived by FFA1 is shown in Figure 3. The structures derived by FFA0 and FFA1 are essentially the same except some sign changes. Notice that, in FFA1,  $H_{0-1}$  is used instead of  $H_{0+1}$ .

When an FIR filter is implemented using a multiplierless approach, the hardware complexity is directly proportional to the number of nonzero bits in the filter coefficients. If the signs of the given impulse response sequences do not change frequently as in the narrowband low-pass filter cases, the coefficient magnitudes of  $H_0 + H_1$  are likely to be larger than those of  $H_0 - H_1$ . Then,  $H_0 + H_1$  has more nonzero bits in the coefficients than  $H_0 - H_1$ . (See examples in Section 4.) If the signs of the given impulse sequences change frequently as in the wide-band low-pass filter cases,  $H_0 - H_1$  is likely to have more nonzero bits in the coefficients than

$H_0 + H_1$ . Thus, to achieve minimum hardware cost, it is necessary to select either FFA0 or FFA1 depending upon the frequency spectrum specifications.

### 2.2. $3 \times 3$ ( $L = 3$ ) FFAs

The  $(3 \times 3)$  FFA produces a parallel filtering structure of block size 3. From (2) with  $L = 3$ , we have

$$\begin{aligned} Y_0 &= H_0X_0 + z^{-3}(H_1X_2 + H_2X_1), \\ Y_1 &= (H_0X_1 + H_1X_0) + z^{-3}H_2X_2, \\ Y_2 &= H_0X_2 + H_1X_1 + H_2X_0. \end{aligned} \quad (7)$$

Direct implementation of (7) computes a block of 3 outputs using 9 length  $N/3$  FIR filters and 6 postprocessing additions, which requires  $3N$  multipliers and  $3N - 3$  adders.

By a similar approach as in  $(2 \times 2)$  FFA0, following  $(3 \times 3)$  FFA0 is obtained,

$$\begin{aligned} Y_0 &= H_0X_0 - z^{-3}H_2X_2 + z^{-3}[H_{1+2}X_{1+2} - H_1X_1], \\ Y_1 &= [H_{0+1}X_{0+1} - H_1X_1] - [H_0X_0 - z^{-3}H_2X_2], \\ Y_2 &= H_{0+1+2}X_{0+1+2} - [H_{0+1}X_{0+1} - H_1X_1] - [H_{1+2}X_{1+2} - H_1X_1]. \end{aligned} \quad (8)$$

Figure 4 shows the filtering structure that results from the  $(3 \times 3)$  FFA0. This structure computes a block of 3 outputs using 6 length  $N/3$  FIR filters and 10 preprocessing and postprocessing additions, which requires  $6(N/3)$  multipliers and  $6(N/3 - 1) + 10$  adders. Notice that  $(3 \times 3)$  FFA0 structure provides a saving of approximately 33% over the traditional structure.

The  $(3 \times 3)$  FFA1 structure can be obtained by modifying (8) as follows:

$$\begin{aligned} Y_0 &= H_0X_0 + z^{-3}H_2X_2 - z^{-3}[H_{2-1}X_{2-1} - H_1X_1], \\ Y_1 &= -[H_{0-1}X_{0-1} - H_1X_1] + [H_0X_0 + z^{-3}H_2X_2], \\ Y_2 &= H_{0-1+2}X_{0-1+2} - [H_{0-1}X_{0-1} - H_1X_1] - [H_{2-1}X_{2-1} - H_1X_1]. \end{aligned} \quad (9)$$

Figure 5 shows the filtering structure that results from the  $(3 \times 3)$  FFA1.

We propose the following  $(3 \times 3)$  FFA2 structure which is efficient when the coefficient magnitudes of  $H_{0-2}$  are smaller than those of  $H_{0-1+2}$  or  $H_{0+1+2}$ ,

$$\begin{aligned} Y_0 &= H_0X_0 + z^{-3}(H_2X_2 + H_1X_1 - H_{2-1}X_{2-1}), \\ Y_1 &= -H_{0-1}X_{0-1} + H_1X_1 + H_0X_0 + z^{-3}H_2X_2, \\ Y_2 &= -H_{0-2}X_{0-2} + H_0X_0 + H_1X_1 + H_2X_2. \end{aligned} \quad (10)$$

Figure 6 shows the filtering structure that results from the  $(3 \times 3)$  FFA2.

### 2.3. Cascading FFAs

The  $(2 \times 2)$  and  $(3 \times 3)$  FFAs can be cascaded together to achieve higher levels of parallelism. The cascading of FFAs is a straightforward extension of the original FFA application [8]. For example, an  $(m \times m)$  FFA can be cascaded with an

$(n \times n)$  FFA to produce an  $(m \times n)$ -parallel filtering structure. The set of FIR filters that result from the application of the  $(m \times m)$  FFA are further decomposed, one at a time, by the application of the  $(n \times n)$  FFA. The resulting set of filters will be of length  $N/(m \times n)$ .

For example, the  $(4 \times 4)$  FFA can be obtained by first applying the  $(2 \times 2)$  FFA0 to (2) and then applying the  $(2 \times 2)$  FFA0 or the  $(2 \times 2)$  FFA1 to each of the filtering operations that result from the first application of the FFA0. The resulting  $(4 \times 4)$  FFA structure is shown in Figure 7. Each filter block  $F_0$ ,  $F_0 + F_1$ , and  $F_1$  represents a  $(2 \times 2)$  FFA structure and can be replaced separately by either  $(2 \times 2)$  FFA0 or  $(2 \times 2)$  FFA1. Each filter block  $F_0$ ,  $F_0 + F_1$ , and  $F_1$  is composed of three subfilters as follows:

- (i)  $F_0 : H_0, H_2, H_0 \pm H_2$ ,
- (ii)  $F_0 + F_1 : H_0 + H_1, H_2 + H_3, (H_0 + H_1) \pm (H_2 + H_3)$ ,
- (iii)  $F_1 : H_1, H_3, H_1 \pm H_3$ ,

where

$$\pm = \begin{cases} +, & \text{for FFA0,} \\ -, & \text{for FFA1.} \end{cases} \quad (11)$$

When the filter block  $F_0 + F_1$  is implemented using FFA1 structure, the subfilters are  $H_{0+1}$ ,  $H_{2+3}$ , and  $H_{0+1} - H_{2+3}$ . Thus, even though FFA1 structure is used for slowly varying impulse response sequences, optimum performance is not guaranteed. In this case, better performance can be obtained by using the FFA1' shown in Figure 8. Since the subfilters in FFA1' are  $H_{0-1}$ ,  $H_{2-3}$ , and  $H_{0-1} - H_{2-3}$ , the FFA1' gives smaller number of nonzero bits than FFA1 for the case of slowly varying impulse response sequences. Notice that the FFA1' structure can be derived by first applying the  $(2 \times 2)$  FFA1 (instead of the  $(2 \times 2)$  FFA0) to (2). When the filter block  $F_0 + F_1$  in Figure 7 is replaced by FFA1' in Figure 8, it can be shown that the outputs are  $y(4k)$ ,  $-y(4k + 1)$ ,  $y(4k + 2)$ , and  $-y(4k + 3)$ .

### 2.4. Selection of FFA types

For given length  $N$  unit sample response values  $\{h_i\}$  and block size  $L$ , the selection of best FFA type can be roughly determined by comparing the signs of the values in subfilters  $H_0, H_1, \dots, H_{L-1}$ .

For example, in the case of  $L = 2$  and even  $N$ ,  $H_0$ , and  $H_1$  are

$$\begin{aligned} H_0 &= \{h_0, h_2, \dots, h_{N-2}\}, \\ H_1 &= \{h_1, h_3, \dots, h_{N-1}\}. \end{aligned} \quad (12)$$

From (12), the  $i$ th value of  $H_0$  can be paired with the  $i$ th value of  $H_1$  as  $(h_0, h_1), (h_2, h_3), \dots, (h_{N-2}, h_{N-1})$ . Comparing the signs of the values in each pair, the number of pairs with opposite signs and the number of pairs with the same signs can be determined. If the number of pairs with opposite signs is larger than the number of pairs with the same signs,  $H_0 + H_1$  is likely to be more efficient than  $H_0 - H_1$ . The sign-comparing procedure can be extended to any block size of  $L$  with appropriate modifications.

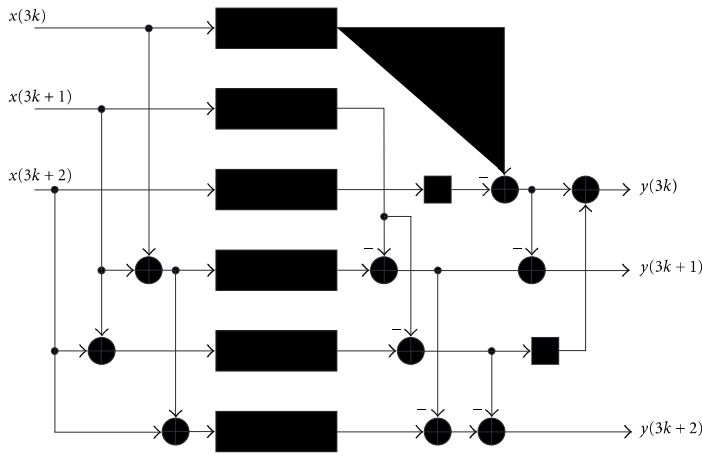


FIGURE 4: 3-parallel FIR filter using FFA0.

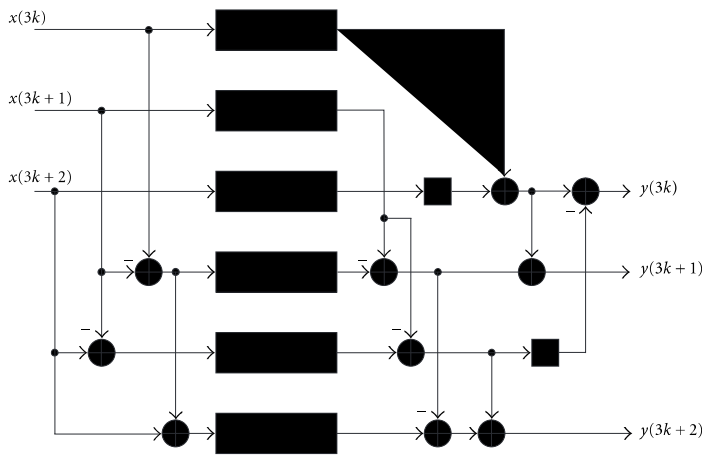


FIGURE 5: 3-parallel FIR filter using FFA1.

**3. LOOK-AHEAD MAD QUANTIZATION**

It is shown in [5, 6, 7] that if the filter coefficients are first scaled before the quantization process is performed, the resulting filter will have much better frequency-space characteristics. The NUS algorithm [6] employs a scalable quantization process. To begin the process, the ideal filter is normalized so that the largest coefficient has an absolute value of 1. The normalized ideal filter is then multiplied by a variable scale factor (VSF). The VSF steps through the range of numbers from 0.4375 to 1.13 with a step size of  $2^{-W}$ , where  $W$  is the coefficient word length. Signed power-of-two (SPT) terms are then allocated to the quantized filter coefficient that

represents the largest absolute difference between the scaled ideal filter and the quantized filter. The NUS algorithm iteratively allocates SPT terms until the desired number of SPT terms is allocated or until the desired NPR, normalized peak ripple, specification is met. Once the allocation of terms stops, the NPR is calculated. The process is then repeated for a new scale factor. The quantized filter leading to the minimum NPR is chosen.

In parallel FIR filters, the NPR cannot be used as a selection criteria for choosing the best quantized filter since pass-band/stopband ripples cannot be defined for the set of sub-filters obtained by the application of FFAs. In [8], it is shown that the maximum absolute difference (MAD) between the

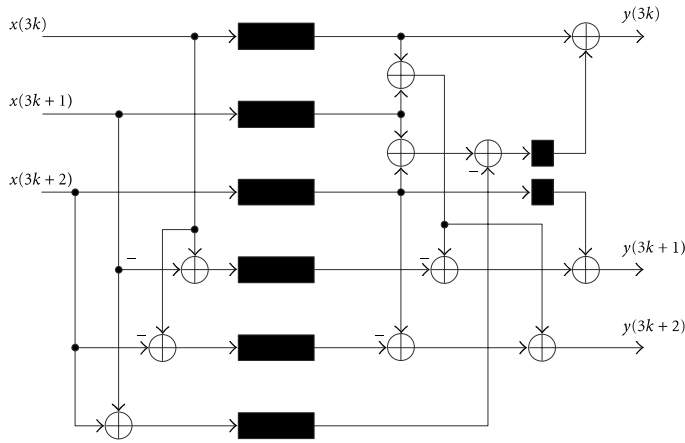


FIGURE 6: 3-parallel FIR filter using FFA2.

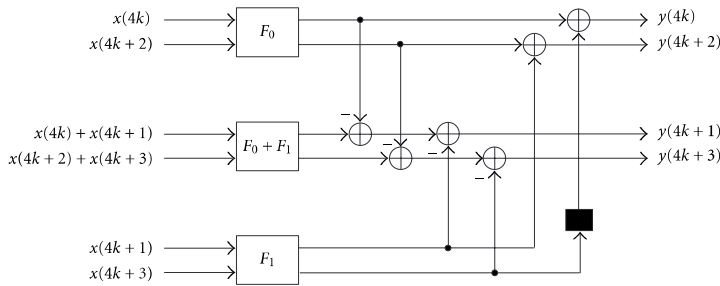


FIGURE 7: 4-parallel FIR filter structure.

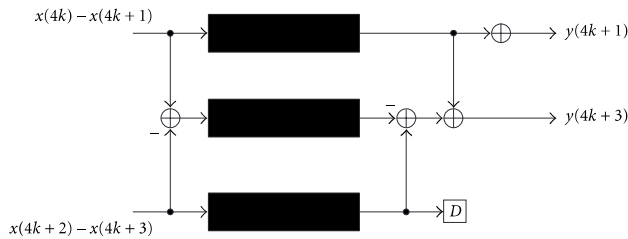


FIGURE 8: FFA1' structure.

frequency responses of the ideal filter and the quantized filter can be used as an efficient selection criteria for parallel filters.

When the quantized filter is implemented, a postprocessing scale factor (PPSF) is used to properly adjust the magnitude of the filter output. The PPSF is calculated as

$$PPSF = \frac{\text{Max}[\text{Absolute}(\text{Ideal Filter Coeffs.})]}{VSF}. \quad (13)$$

In the cases where large levels of parallelism are used, the PPSFs can contribute to a significant amount of hardware overhead. In [8], to reduce this hardware overhead the PPSFs are restricted to the following set of values: {0.125, 0.25, 0.375, 0.5, 0.625, 0.75, 0.875, 1}. The original PPSF is replaced with the new PPSF that is the nearest in value. Since the scale factor of the quantized filter is shifted in value, the quantized coefficients must also be properly shifted

```

For each filter section in the parallel FIR filter
{
  Normalize the set of filter coefficients so that the magnitude
  of the largest coefficient is 1;
  For VSF = Lower Scale:Step Size:Upper Scale,
  {
    Compute PPSF by (13);
    Convert PPSF into Canonic Signed Digit form;
    If (No. of nonzero bits in PPSF) < prespecified value,
    {
      Scale normalized coefficients with VSF;
      Quantize the scaled coefficients using SPT term
      allocation scheme in NUS algorithm;
      Calculate MAD between the frequency responses
      of the ideal and quantized filters;
    }
  }
  Choose scale factor that leads to the minimum MAD;
}
    
```

ALGORITHM 1: Look-ahead MAD quantization.

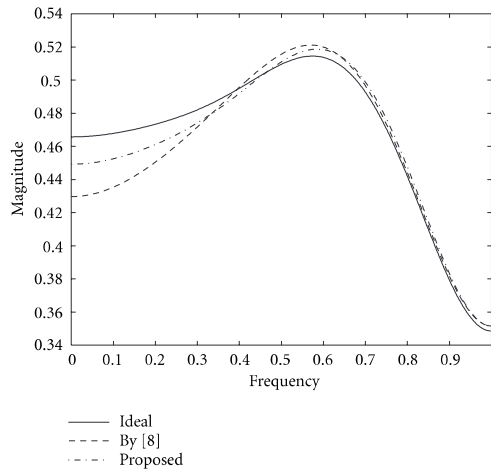


FIGURE 9: Frequency responses of Example 1.

in value. This is accomplished using the following three steps:

- (i) determine effective coefficients, effective coeffs. = quantized coeffs.  $\times$  PPSF;
- (ii) determine shifted coefficients with new PPSF, shifted coeffs. = effective coeffs./new PPSF;
- (iii) quantize the shifted coefficients.

However, the above steps are not guaranteed to give optimal quantized coefficients for the new PPSF value. The reason is that the quantization in (iii) is performed on the already quantized coefficients.

To avoid this problem, LMAD quantization algorithm is proposed. In the proposed algorithm, the PPSF for a given VSF is computed by (13) before the quantization step begins. If the number of nonzero bits in the computed PPSF is less than a prespecified value, then the normalized coefficients are scaled by the VSF and the scaled coefficients are quantized. Otherwise, the procedure is repeated for the next VSF value.

In [8], the number of nonzero bits in PPSF is fixed. However, in the proposed approach, the number of nonzero bits in PPSF can be varied and the PPSF value giving the best performance can be selected. From our simulation experience, increasing the number of nonzero bits in PPSF more than three does not improve the numerical performance significantly.

*Example 1.* Consider an ideal filter section with the following coefficients [8]: ideal coeffs. =  $\{-.0648 .1404 .4328 -.0818 .0391\}$ . In [8], these coefficients are quantized using word length of 7 bits to the following values by the scalable MAD quantization algorithm:  $\{-.109375 .203125 .6875 -.140625 .046875\}$  with PPSF = 0.625. The computed MAD value is 0.0360125. For comparison, the ideal coefficients are quantized using the proposed algorithm with PPSF =

TABLE 1: The number of adders by the method used in [8] and by the proposed method. The numbers inside parentheses denote the FFA types used for each case.

	24-Tap FIR		72-Tap FIR	
	By [8]	Proposed	By [8]	Proposed
$L = 1$	56 (0)	49 (0)	125 (0)	96 (0)
$L = 2$	74 (0)	54 (1)	192 (0)	173 (1)
$L = 3$	119 (0)	99 (1)	293 (0)	272 (1)
$L = 4$	133 (0-0-0)	123 (0-1-0)	313 (0-0-0)	303 (1-1-1)

0.625. The quantized coefficients are  $\{-.109375 .21875 .6875 -.140625 .0625\}$ . The computed MAD value is 0.01648125. Notice that the MAD value by the proposed method is only 45% of the MAD value in [8]. Frequency responses are compared in Figure 9.

Table 1 shows that, for the two low-pass FIR filter examples in [8], the proposed method can save up to 24% of adders. In [8], only FFA type 0 is used for each value of  $L$ . However, as can be seen from Table 1, better results are obtained by selecting FFA type(s) properly for each  $L$ .

*Example 2.* In this example, the hardware saving by the appropriate selection of FFA structures is compared with the hardware saving by the proposed LMAD quantization scheme using a simple low-pass filter with filter order = 7, passband edge =  $0.1\pi$ , maximum passband ripple = 0.02 dB, stopband edge =  $0.3\pi$ , and minimum stopband attenuation = 22 dB. In this example, only block size of 2 ( $L = 2$ ) is considered.

Table 2 shows the filter coefficients obtained by FFA0 without scaling. Table 3 shows the filter coefficients obtained

TABLE 2: Filter coefficients (canonic signed digit format) and the number of nonzero bits for FFA0 without scaling (word-length = 8).

	$H_0$	$H_{0+1}$	$H_1$
Coefficients	0 0 0 0 1 0 0 1	0 0 1 0 $\bar{1}$ 0 0 1	0 0 0 1 0 0 0 $\bar{1}$
	0 0 0 1 0 1 0 0	0 1 0 $\bar{1}$ 0 $\bar{1}$ 0 1	0 0 1 0 $\bar{1}$ 0 0 0
	0 0 1 0 $\bar{1}$ 0 0 0	0 1 0 $\bar{1}$ 0 $\bar{1}$ 0 1	0 0 0 1 0 1 0 0
	0 0 0 1 0 0 0 $\bar{1}$	0 0 1 0 $\bar{1}$ 0 0 1	0 0 0 0 1 0 0 1
Nonzero bits	8	14	8

TABLE 3: Filter coefficients and the number of nonzero bits for FFA0 with LMAD scaling (word-length = 7).

	$H_0$	$H_{0+1}$	$H_1$
Coefficients	0 0 1 0 0 1 0	0 0 1 0 1 0 0	0 1 0 0 0 $\bar{1}$ 0
	0 1 0 1 0 0 0	0 1 0 0 1 0 0	1 0 $\bar{1}$ 0 0 0 0
	1 0 $\bar{1}$ 0 0 0 0	0 1 0 0 1 0 0	0 1 0 1 0 0 0
	0 1 0 0 0 $\bar{1}$ 0	0 0 1 0 1 0 0	0 0 1 0 0 1 0
Nonzero bits	8	8	8

TABLE 4: Filter coefficients and the number of nonzero bits for FFA1 with LMAD scaling (word-length = 7).

	$H_0$	$H_{0-1}$	$H_1$
Coefficients	0 0 1 0 0 0 $\bar{1}$	$\bar{1}$ 0 1 0 0 0 0	0 1 0 $\bar{1}$ 0 0 1
	0 1 0 0 0 0 0	0 $\bar{1}$ 0 0 0 0 0	0 1 0 1 0 0 0
	0 1 0 1 0 0 0	0 1 0 0 0 0 0	0 1 0 0 0 0 0
	0 1 0 $\bar{1}$ 0 0 1	1 0 $\bar{1}$ 0 0 0 0	0 0 1 0 0 0 $\bar{1}$
Nonzero bits	8	6	8

by FFA0 with LMAD scaling. Notice that the filter coefficients by FFA0 with LMAD scaling satisfy the given specifications by word-length of 7 bits while the filter coefficients by FFA0 without scaling require word-length of 8 bits. The reduction of the word-length is due to the use of scaling factors. The PPSFs for the filter coefficients by FFA0 with LMAD scaling are 0010001( $H_0$ ), 0101000( $H_{0+1}$ ), and 0010001( $H_1$ ). Each PPSF contains two nonzero bits, which corresponds to the overhead of one adder. Table 4 shows the filter coefficients obtained by FFA1 with LMAD scaling. The PPSFs for the filter coefficients by FFA1 with LMAD scaling are 00101( $H_0$ ), 00001( $H_{0+1}$ ), and 00101( $H_1$ ). Frequency responses of ideal filter and the filter obtained by FFA1 quantized by LMAD are compared in Figure 10.

To compare the hardware savings by the quantization and the proper selection of FFA types, only  $H_{0+1}$  or  $H_{0-1}$  subfilters are considered. From Table 2, the number of nonzero bits for  $H_{0+1}$  of nonscaled FFA0 filter is 14 while the number of nonzero bits for  $H_{0+1}$  of scaled FFA0 filter is 10 (including PPSF). Thus, in addition to the word-length reduction, hardware saving of about 28% can be obtained by LMAD scaling.

From Table 4, the number of nonzero bits for  $H_{0-1}$  of scaled FFA1 filter is 7 (including PPSF). Thus, 22% further saving is obtained by the selection of proper filter type. Thus, in this example, about half of the saving is due to the LMAD quantization and the other half is due to proper filter type selection.

#### 4. DESIGN EXAMPLES

In this section, three design examples with various frequency specifications are given.

*Example 3.* Consider a narrowband low-pass filter with filter order = 35, passband edge =  $0.2\pi$ , maximum passband ripple = 0.185 dB, stopband edge =  $0.3\pi$ , and minimum stopband attenuation = 33.5 dB. As can be seen from Figure 11, the signs of the impulse response sequences (designed by the Remez exchange algorithm) change slowly.

For  $L = 2$ , according to the discussions in Section 2.4, the number of pairs with the same signs is 16, while the number of pairs with the opposite signs is only 2. Thus, FFA1 is more efficient than FFA0. By the LMAD quantization algorithm, the number of nonzero bits required for  $H_{0+1}$  is 42 but the number of nonzero bits required for  $H_{0-1}$  is 24. Thus the hardware cost of  $H_{0-1}$  is about 57% of the hardware cost of  $H_{0+1}$ . The frequency responses for  $L = 2$  are compared in Figure 12.

For  $L = 3$ , the number of pairs with the same signs in subfilter pairs  $\{H_0, H_1\}$ ,  $\{H_1, H_2\}$ , and  $\{H_{0+2}, H_1\}$  is 28 while the number of pairs with the opposite signs is 8. Also, the number of pairs with the same signs in subfilter pairs  $\{H_0, H_1\}$ ,  $\{H_1, H_2\}$ , and  $\{H_0, H_2\}$  is 12. Thus, FFA1 is the most efficient.

For  $L = 4$ , the number of pairs with the opposite signs in subfilter pair  $\{H_0, H_2\}$  is 7 while the number of pairs with

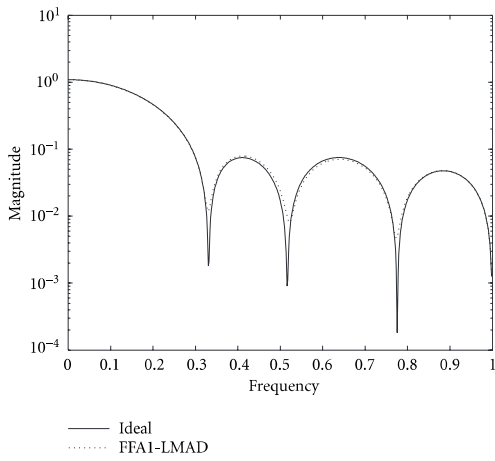


FIGURE 10: Frequency responses of ideal filter and the filter obtained by FFA1 quantized by LMAD.

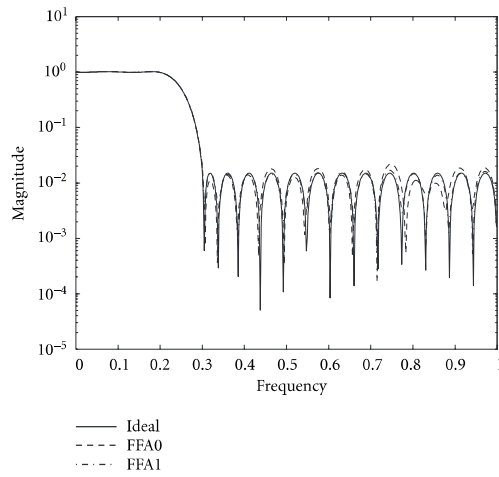


FIGURE 12: Frequency responses of Example 3.

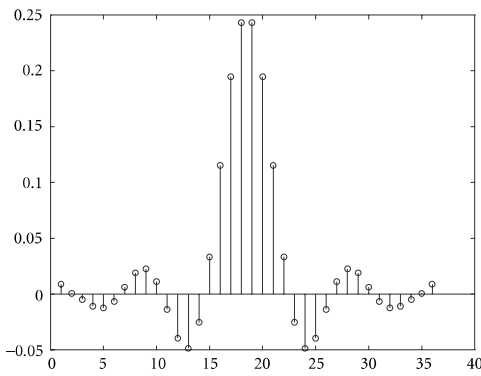


FIGURE 11: Ideal impulse response of Example 3.

the same signs is 2. Thus, FFA0 is the most efficient for  $F_0$ . The number of pairs with the opposite signs in the subfilter pair  $\{H_1, H_3\}$  is 7 while the number of pairs with the same signs is 2. Thus, FFA0 is the most efficient for  $F_1$ . By a similar procedure, it can be shown that FFA1' is the most efficient choice for  $F_0 + F_1$ .

The design results for  $L = 2, 3$ , and 4 are summarized in Table 5. For  $L = 2$  and  $L = 3$ , about 20% of the hardware can be saved by a proper choice of FFA types. However, for  $L = 4$ , only 7% of the hardware saving can be achieved by a proper choice of FFA types. The main reason is that the correlation of filter coefficients between subfilters is reduced as the block size increases.

*Example 4.* Consider a wideband low-pass filter with filter order = 62, passband edge =  $0.8\pi$ , maximum passband ripple =

TABLE 5: Total number of nonzero bits for Example 3 with different block size and various structures (word-length = 10).

$L = 2$		$L = 3$			$L = 4$		
FFA0	FFA1	FFA0	FFA1	FFA2	FFA0-0-0	FFA1-1-1	FFA0-1'-0
102	84	144	115	123	194	196	184

TABLE 6: Total number of nonzero bits for Example 4 with different block size and various structures (word-length = 9 for  $L = 2$  and word-length = 10 for  $L = 3$  and  $L = 4$ ).

$L = 2$		$L = 3$			$L = 4$		
FFA0	FFA1	FFA0	FFA1	FFA2	FFA1-1-1	FFA1-0-1	FFA0-1'-0
170	193	244	286	284	284	285	295

ripple = 0.27 dB, stopband edge =  $0.85\pi$ , and minimum stopband attenuation = 32.5 dB. As can be seen from Figure 13, the signs of the impulse response sequences change frequently. By the sign comparing procedure, the best FFA types are predicted as FFA0 ( $L = 2$ ), FFA0 ( $L = 3$ ), and FFA1-FFA1-FFA1 ( $L = 4$ ).

The design results for  $L = 2, 3$ , and 4 are summarized in Table 6. For  $L = 2$  and  $L = 3$ , about 12%–15% of the hardware can be saved by a proper choice of FFA types. For  $L = 4$ , 4% of the hardware saving can be achieved by a proper choice of FFA types.

*Example 5.* Consider a narrow bandpass filter with filter order = 86, passband =  $0.22\pi \sim 0.3\pi$ , maximum passband ripple = 0.19 dB, stopband =  $0 \sim 0.18\pi, 0.34\pi \sim \pi$ , and minimum stopband attenuation = 35 dB. Figure 14 shows



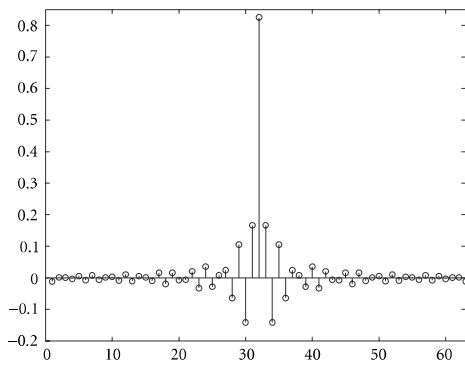


FIGURE 13: Ideal impulse response of Example 4.

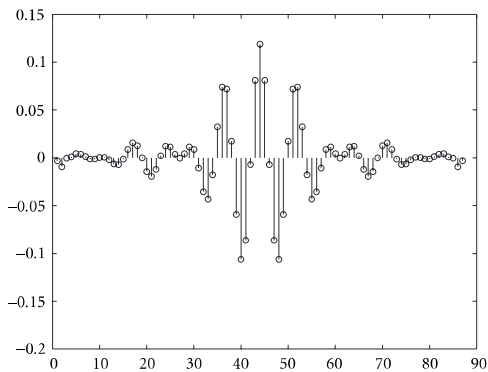


FIGURE 14: Ideal impulse response of Example 5.

the impulse response sequence. By the sign-comparing procedure, the best FFA types are predicted as FFA1 ( $L = 2$ ), FFA2 ( $L = 3$ ), and FFA0-FFA1'-FFA0 ( $L = 4$ ). The design results for  $L = 2, 3$ , and  $4$  are summarized in Table 7. For  $L = 2$  and  $L = 3$ , about 16%–18% of the hardware can be saved by a proper choice of FFA types. For  $L = 4$ , 4% of the hardware saving can be achieved by a proper choice of FFA types.

## 5. CONCLUSIONS

It has been shown that the hardware cost and power consumption of parallel FIR filters can be reduced significantly by exploiting the frequency spectrum characteristics. For example, in narrowband low-pass filters, the signs of consecutive unit sample response values do not change much and therefore their difference (FFA1) can require fewer number of bits than their sum (FFA0). In wideband low-pass filters, the signs of consecutive unit sample response values change frequently and therefore their sum (FFA0) can require fewer number of bits than their difference (FFA1). To determine

TABLE 7: Total number of nonzero bits for Example 5 with different block size and various structures (word-length = 12).

$L = 2$		$L = 3$			$L = 4$		
FFA0	FFA1	FFA0	FFA1	FFA2	FFA0-0-0	FFA1-1-1	FFA0-1'-0
299	252	413	345	339	461	474	458

the best FFA type for given impulse response sequence and block size  $L$ , a sign-comparing procedure was proposed. The usefulness of the proposed sign-comparing procedure was demonstrated by several examples. Also, the proposed look-ahead MAD quantization algorithm was shown to be very efficient for the implementation of parallel FIR filters.

Substructure sharing is the process of examining the hardware implementation of the filter coefficients and sharing the hardware units that are common among the filter coefficients. Using the substructure sharing techniques in [8], further savings in hardware cost and power consumption can be achieved.

Developing a similar approach to power reduction of adaptive FIR filters will be an interesting future research. Further research needs to be directed towards finite word-length analysis of these low-power parallel FIR filters.

## ACKNOWLEDGMENTS

This research was supported in part by Information and Communication Research Institute at Chonbuk National University and by NSF under grant number CCR-9988262.

## REFERENCES

- [1] J. W. Adams and A. N. Willson Jr., "Some efficient digital prefilter structures," *IEEE Trans. Circuits and Systems*, vol. 31, no. 3, pp. 260–265, 1984.
- [2] J. T. Ludwig, S. H. Nawab, and A. P. Chandrakasan, "Low-power digital filtering using approximate processing," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 3, pp. 395–400, 1996.
- [3] N. Sankaraya, K. Roy, and D. Bhattacharya, "Algorithms for low-power high speed FIR filter realization using differential coefficients," *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 44, no. 6, pp. 488–497, 1997.
- [4] N. R. Shanbhag and M. Goel, "Low-power adaptive filter architectures and their application to 51.84 Mb/s ATM-LAN," *IEEE Trans. Signal Processing*, vol. 45, no. 5, pp. 1276–1290, 1997.
- [5] H. Samuelli, "An improved search algorithm for the design of multiplierless FIR filters with powers-of-two coefficients," *IEEE Trans. Circuits and Systems*, vol. 36, no. 7, pp. 1044–1047, 1989.
- [6] D. Li, J. Song, and Y. C. Lim, "A polynomial-time algorithm for designing digital filters with power-of-two coefficients," in *Proc. IEEE International Symposium on Circuits and Systems*, vol. 1, pp. 84–87, Chicago, Ill, USA, May 1993.
- [7] C.-L. Chen, K.-Y. Khoo, and A. N. Willson Jr., "An improved polynomial-time algorithm for designing digital filters with power-of-two coefficients," in *Proc. IEEE International Symposium on Circuits and Systems*, vol. 1, pp. 223–226, Seattle, Wash, USA, 30 April–3 May 1995.

- [8] D. A. Parker and K. K. Parhi, "Low-area/power parallel FIR digital filter implementations," *Journal of VLSI Signal Processing*, vol. 17, no. 1, pp. 75–92, 1997.
- [9] M. Winzker, "Low-power arithmetic for the processing of video signals," *IEEE Trans. on VLSI Systems*, vol. 6, no. 3, pp. 493–497, 1998.
- [10] J.-G. Chung, Y.-B. Kim, H.-J. Jeong, K. K. Parhi, and Z. Wang, "Efficient parallel FIR filter implementations using frequency spectrum characteristics," in *Proc. IEEE International Symposium on Circuits and Systems*, vol. 5, pp. 483–486, Monterey, Calif, USA, 31 May–3 June 1998.
- [11] Z. J. Mou and P. Duhamel, "Short-length FIR filters and their use in the fast nonrecursive filtering," *IEEE Trans. Signal Processing*, vol. 39, no. 6, pp. 1322–1332, 1991.

**Jin-Gyun Chung** received his B.S. degree in electronic engineering from Chonbuk National University, Chonju, South Korea, in 1985 and the M.S. and Ph.D. degrees in electrical engineering from the University of Minnesota, Minneapolis, Minnesota, in 1991 and 1994, respectively. Since 1995, he has been with the Department of Electronic and Information Engineering at Chonbuk National University, where he is currently Associate Professor. His research interests are in the area of VLSI architectures and algorithms for signal processing and communication systems, which include the design of high-speed and low-power algorithms for digital filters, DSL systems, OFDM systems, and ultrasonic NDE systems.



**Keshab K. Parhi** is a distinguished McKnight University Professor of Electrical and Computer Engineering at the University of Minnesota, Minneapolis, where he also holds the Edgar F. Johnson Professorship. He received the B.Tech., M.S.E.E., and Ph.D. degrees from the Indian Institute of Technology, Kharagpur (India, 1982), the University of Pennsylvania, Philadelphia (1984), and the University of California at Berkeley (1988), respectively. His research interests include all aspects of physical layer VLSI implementations of broadband access systems. He is currently working on VLSI adaptive digital filters, equalizers and beamformers, error control coders and cryptography architectures, low-power digital systems, and computer arithmetic. He has published over 330 papers in these areas. He has authored the text book "VLSI Digital Signal Processing Systems" (Wiley, 1999) and coedited the reference book "Digital Signal Processing for Multimedia Systems" (Dekker, 1999). Dr. Parhi has been a Visiting Professor at the Delft University of Technology and at Lund University. He has been a Visiting Researcher at the NEC Corporation, Japan, and a Technical Director—DSP Systems in the Office of CTO at Broadcom Corporation in Irvine, Calif. Dr. Parhi has served on editorial boards of the IEEE Transactions on Circuits and Systems, Signal Processing, Circuits and Systems—Part II: Analog and Digital Signal Processing, VLSI Systems, and the IEEE Signal Processing Letters. He is an editor of the Journal of VLSI Signal Processing. He has received numerous best paper awards including the 2001 IEEE W. R. G. Baker prize paper award. He has been a Distinguished Lecturer (1994–1999) and a recipient of a Golden Jubilee medal (1999) from the IEEE Circuits and Systems Society. He received a Young Investigator Award from the National Science Foundation in 1992, and was elected a Fellow of IEEE in 1996.

