New Distributed Arithmetic Algorithm for Low-Power FIR Filter Implementation

Sangyun Hwang, *Student Member, IEEE*, Gunhee Han, *Member, IEEE*, Sungho Kang, *Member, IEEE*, and Jaeseok Kim, *Member, IEEE*

***Abstract*—This letter proposes a new distributed arithmetic (DA) algorithm for low-power finite-impulse response (FIR) filter implementation. The characteristic of the proposed algorithm is that the FIR filters using the proposed algorithm do not need to employ two's complement representation in lookup tables as well as multiply-and-accumulation blocks. Thus, the proposed algorithm can minimize the dynamic power consumption of the FIR filters. The experimental results show that the lowpass FIR filter using the proposed algorithm achieves 29% and 26% power consumption reduction compared to that using the conventional algorithm for zero-mean random inputs and speech inputs, respectively.**

***Index Terms*—Distributed arithmetic, lookup table (LUT), low-power finite-impulse response (FIR) filter.**

I. Introduction

FINITE IMPULSE response (FIR) filters using distributed arithmetic (DA) are widely used to implement pulse-shaping filters, digital phase-locked loop (PLL) frequency synchronizers, discrete cosine transform (DCT) cores, and so forth in hand-held applications where low-power consumption is required. DA is a bit-serial operation that computes the inner product of two vectors (one of which is a constant) in parallel [1], [2]. DA eliminates the need for multiply operations by using lookup tables (LUTs). In general, the FIR filters using DA consist of multiple LUTs and multiply-and-accumulation (MAC) blocks [1].

In most digital circuits, two's complement is used to represent numbers, since arithmetic operations are easy to perform. However, the use of two's complement representation can increase the dynamic power consumption of the digital circuits when the signals being processed in the digital circuits switch frequently around zero and do not utilize the entire bit size, as presented in [3]. One approach to minimize the dynamic power consumption of the digital circuits is to use the sign magnitude representation [3]. Although it can be adopted into the FIR filters to reduce power consumption, the hardware complexity of the FIR filters can increase, since the positive and negative arithmetic operations are performed separately in the sign magnitude representation. Thus, in this letter, a new DA algorithm is proposed to reduce the power consumption of the FIR filters without using the sign magnitude representation.

Manuscript received April 27, 2003; revised July 29, 2003. This work was supported by the Korea Research Foundation under Grant KRF 2000–042-E00029. Computer-aided drawing tools were supported by the IC Design Education Center. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Richard C. Kavanagh.

The authors are with the Department of Electrical and Electronic Engineering, Yonsei University, Seoul 120-749, Korea (e-mail: sea@asic.yonsei.ac.kr; jskim@asic.yonsei.ac.kr).

Digital Object Identifier 10.1109/LSP.2004.824029

II. Conventional DA Algorithm for FIR Filter Implementation

In this letter, the notations of [1] are used to express following equations. The conventional DA algorithm with offset-binary coding for an N-tap FIR filter is given by [1]

$$\begin{aligned} Y &= \sum_{i=0}^{N-1} c_i x_i = \frac{1}{2}\sum_{j=0}^{W-1}\left(\sum_{i=0}^{N-1} c_i d_{i,W-1-j}\right)2^{-j} \\ &\quad - \frac{1}{2}\sum_{i=0}^{N-1} c_i 2^{-(W-1)} \\ &= \frac{1}{2}\sum_{j=0}^{W-1}\underbrace{\left\{d_{0,W-1-j}\left(c_0 + \sum_{i=1}^{N-1} c_i d_{0,W-1-j} d_{i,W-1-j}\right)\right\}}_{a}2^{-j} \\ &\quad - \frac{1}{2}\sum_{i=0}^{N-1} c_i 2^{-(W-1)} \end{aligned} \tag{1}$$

where $\{c_i\}$'s are coefficients and $\{x_i\}$'s are W-bit filter inputs using two's complement representation. $d_{i,j}$ is obtained as

$$d_{i,j} = \begin{cases} -\left(x_{i,W-1} - \bar{x}_{i,W-1}\right), & \text{when } j = W-1 \\ x_{i,j} - \bar{x}_{i,j}, & \text{otherwise} \end{cases} \tag{2}$$

where $x_{i,W-1}$ denotes the $(W-1)$th bit of x_i and $d_{i,j} \in \{-1, 1\}$. In (1), a is implemented by a 2^N-word LUT, while the second term of (1) is a constant value. Thus, the FIR filter can be implemented with one 2^N-word LUT and one MAC block. The MAC block performs as $\sum_{j=0}^{W-1} m_j 2^{-j}$, where m_j is the jth input of the MAC block. As shown in (1), in the conventional DA algorithm, the 2^N-word LUT can be implemented with a 2^{N-1}-word LUT with a 2×1 multiplexer (MUX) and a sign conversion block, since $d_{i,j}$ is ± 1 [1]. The 2×1 MUX selects one value between $\{d_{1,W-1-j}, \ldots, d_{N-1,W-1-j}\}$ and $-1 \times \{d_{1,W-1-j}, \ldots, d_{N-1,W-1-j}\}$ according to $d_{0,W-1-j}$. The output of the 2×1 MUX becomes the input of the 2^{N-1}-word LUT. The sign conversion block performs two's complement of the output of the 2^{N-1}-word LUT when $d_{0,W-1-j} = -1$. To reduce the hardware complexity of the 2^N-word LUT, the LUT decomposition can be used [1], [2]. If the LUT decomposition is adopted into the 2^N-word LUT, the 2^N-word LUT can be implemented with multiple smaller LUTs and adders. If the N input elements of the 2^N-word LUT

are divided into M datasets, the 2^N-word LUT can be designed with $M2^{N/M-1}$-word LUTs with M 2×1 MUXs, M sign conversion blocks, and $(M-1)$ adders.

III. Proposed DA Algorithm for FIR Filter Implementation

The objective of the proposed DA algorithm is the minimization of dynamic power consumption of FIR filters. In the proposed algorithm, since $d_{i,j}$ is -1 or 1, the N-tap FIR filter is given by

$$\begin{aligned} Y &= \frac{1}{2}\sum_{j=0}^{W-1}\left(\sum_{i=0}^{N-1} c_i d_{i,W-1-j}\right)2^{-j} - \frac{1}{2}\sum_{i=0}^{N-1} c_i 2^{-(W-1)} \\ &= \frac{1}{2}\sum_{j=0}^{W-1}\left\{\sum_{i=0}^{N-1}\left(|c_i| - 2|c_i|\,u_i^j\right)\right\}2^{-j} - \frac{1}{2}\sum_{i=0}^{N-1} c_i 2^{-(W-1)} \\ &= -\sum_{j=0}^{W-1}\underbrace{\left(\sum_{i=0}^{N-1}|c_i|\,u_i^j\right)}_{a}2^{-j} \\ &\quad + \frac{1}{2}\left(\sum_{j=0}^{W-1}2^{-j}\sum_{i=0}^{N-1}|c_i| - \sum_{i=0}^{N-1} c_i 2^{-(W-1)}\right) \end{aligned} \tag{3}$$

where ${u_i}^j$ is expressed as

$$u_i^j = \begin{cases} 0, & \text{when } \operatorname{sign}(c_i)d_{i,W-1-j} = 1 \\ 1, & \text{when } \operatorname{sign}(c_i)d_{i,W-1-j} = -1 \end{cases} \tag{4}$$

where $\operatorname{sign}(x)$ denotes the sign of x. In the proposed algorithm, a of (3) is implemented with a 2^N-word LUT that has an input of $\{d_{0,W-1-j}, d_{1,W-1-j}, \ldots, d_{N-1,W-1-j}\}$ similar to that of the conventional algorithm. Thus, the proposed algorithm does not require additional hardware blocks for the operations of ${u_i}^j$. The main characteristic of the 2^N-word LUT is to generate only nonnegative values. Table I shows the LUT examples of the conventional and the proposed algorithms where $y = c_1x_1 + c_2x_2 (c_1 = -1, c_2 = 2)$. In the proposed algorithm, the equation $y = -x_1 + 2x_2$ is written as $y = 3 - 2\,y_p$, where $y_p = u_1 + 2u_2$. u_1 is calculated from $\operatorname{sign}(c_1) \times x_1$, while the value of u_2 is determined by $\operatorname{sign}(c_2) \times x_2$. The LUT for $y_p (= u_1 + 2u_2)$ is shown in Table I, where the proposed LUT generates only nonnegative values. Since a of (3) is implemented with a 2^N-word LUT and the second term of (3) is a constant value, the N-tap FIR filter using the proposed algorithm can be implemented by one 2^N-word LUT and one MAC block similar to that using the conventional algorithm. The LUT decomposition of [1] and [2] can be also used to reduce the hardware complexity of the proposed 2^N-word LUT. If the LUT decomposition is adopted into the 2^N-word LUT, the 2^N-word LUT can be implemented using $M2^{N/M}$-word LUTs and $(M-1)$ adders in case the N input elements of the 2^N-word LUT are divided into M datasets.

Since the proposed LUTs generate only nonnegative values, the FIR filter using the proposed algorithm does not require two's complement representation in the operations of the FIR filter except in the final subtraction between the first and the second term of (3). Thus, the proposed algorithm can minimize the dynamic power dissipation of the FIR filter.

TABLE I
Lookup Table Examples of the Conventional and the Proposed Algorithms Where $y = -x_1 + 2x_2 = 3 - 2(u_1 + 2\,u_2) = 3 - 2\,y_p$

x_1	x_2	Conventional 2^2-word LUT contents (=y)	Proposed 2^2-word LUT contents ($=y_p$)
1	1	1	1 (u_1=1, u_2=0)
1	-1	-3	3 (u_1=1, u_2=1)
-1	1	3	0 (u_1=0, u_2=0)
-1	-1	-1	2 (u_1=0, u_2=1)

IV. Experimental Results and Conclusion

To compare the hardware complexity and the power consumption between the conventional and the proposed algorithms, two lowpass FIR filters are implemented: one for the conventional algorithm and the other for the proposed algorithm. The specifications of the two FIR filters are as follows. If the input signal frequency is f_s, the passband edge frequency and the stopband edge frequency of the FIR filters are $0.26f_s$ and $0.37f_s$, respectively. The peak ripple value in the passband is 0.7 dB, and the minimum attenuation in the stopband is 50 dB. The number of taps required for above filter specifications is 48. The bit size of the coefficients is 12, and the input bit size of the FIR filters is set to 16 ($W = 16$). In this letter, two input signals, which are random inputs with zero mean and speech inputs, are used to compare the power consumption between the conventional and the proposed FIR filters.

Fig. 1 shows the block diagram of the conventional and the proposed 48-tap FIR filters. In Fig. 1, the difference between the two filters is the constant and the gain value in the 'A' region as shown in (1) and (3). In Fig. 1, K_{conv} and K_{prop} are the constant values of the second terms in (1) and (3). The clock frequency of the Clk16 is 16 times faster than that of the Clk1. Since the 48-tap FIR filter of Fig. 1 operates at the clock rate of Clk16 and $W = 16$, the FIR filter is implemented using one 2^{48}-word LUT, one MAC block, and one adder. The operations of the FIR filter are as follows: each bit of the filter input ($= x_{i,j}$) is first converted to $d_{i,j}$ by using (2). In this letter, the value of $d_{i,j}$ is represented by mapping 0 for 1 and 1 for -1. $d_{i,j}$ is delivered into the s2p_reg48_j, which is a 48-bit shift register. The 48-bit 16×1 MUX selects one among the 16 outputs of the 48-bit shift registers according to the output of the four-bit counter. The 48-bit output of the 48-bit 16×1 MUX is used as the input of the 2^{48}-word LUT. The MAC block performs the MAC operations of the outputs of the 2^{48}-word LUT during 16 Clk16 cycles. After the 16 Clk16 cycles, a filter output is calculated from the output of the MAC block and the constant values K_{conv} or K_{prop}.

As mentioned earlier, the LUT decomposition can be used to reduce the hardware complexity of the 2^{48}-word LUT. As shown in Fig. 1, if the 48 input bits of the 2^{48}-word LUT are divided into four datasets, the 2^{48}-word LUT can be implemented using four 2^{12}-word LUTs and three adders. The 2^{12}-word LUTs are also designed with multiples of smaller LUTs and adders. If the

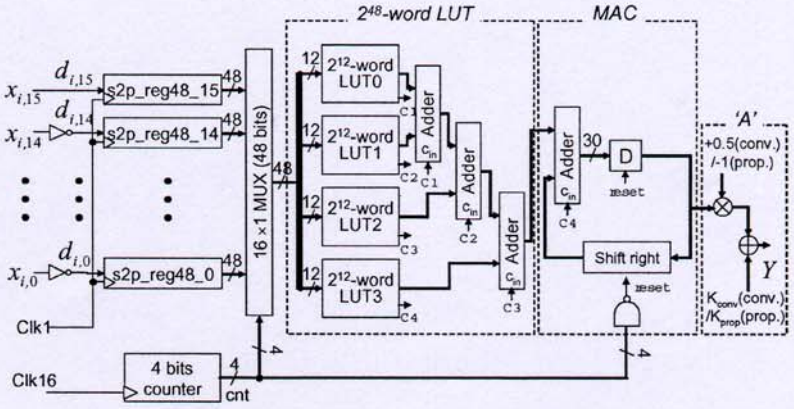

Fig. 1. Block diagrams of the conventional and the proposed 48-tap FIR filters ($W = 16$, $N = 48$).

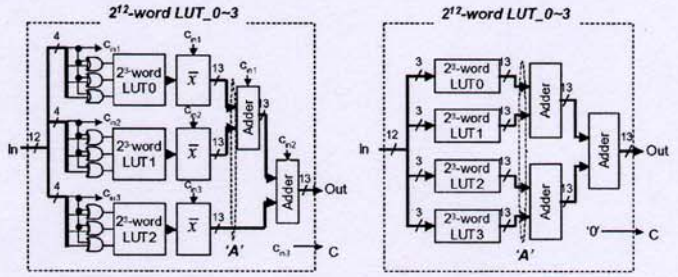

Fig. 2. The 2^{12}-word lookup tables for the conventional and the proposed 48-tap FIR filters. (a) Conventional 2^{12}-word LUT [1]. (b) Proposed 2^{12}-word LUT.

2^{12}-word LUTs are implemented using multiples of 2^3-word LUTs and adders, the 12 input bits of the 2^{12}-word LUTs should be divided into three datasets and four datasets for the conventional FIR filter and the proposed one, respectively. Fig. 2 shows the 2^{12}-word LUTs for the conventional and the proposed 48-tap FIR filters. In Fig. 2(a), $\bar{x}$ represents a one's complementer. As shown in Fig. 2(a), the 2^{12}-word LUT for the conventional algorithm requires three 2^3-word LUTs with three 2×1 MUXs, three one's complementers, and two adders to implement the 2^{12}-word LUT. The 2×1 MUX is implemented using XOR gates as shown in Fig. 2(a). The one's complementer performs the bitwise negation of its input signal when its control bit is 1. Although a sign conversion block is implemented with a one's complementer and an incrementer, the incrementer for the sign conversion block may not be required, since the carry input of an adder is used to perform the increment operation shown in Fig. 2(a).

Fig. 2(b) shows the 2^{12}-word LUT for the proposed 48-tap FIR filter. In the proposed algorithm, four 2^3-word LUTs and three adders are required to implement the 2^{12}-word LUT. Unlike the conventional 2^{12}-word LUT, the proposed one requires an additional 2^3-word LUT and one adder. However, the proposed 2^{12}-word LUT does not require three 2×1 MUXs and three one's complementers of the conventional one.

To confirm the merit of the proposed LUT, the transition rates for the outputs of the conventional and the proposed 2^3-word LUTs are estimated and compared using C simulations with 10 000 randomly generated 16-bit input signals before implementing the conventional and the proposed FIR filters. In this letter, the transition rate represents the average number of signal transitions ($0 \rightarrow 1$ or $1 \rightarrow 0$) per one Clk16 cycle. Fig. 3 shows the average transition rates for the 13-bit outputs of 2^3-word LUTs of the conventional and the proposed 48-tap FIR filters. The transition rate of each bit is averaged over all outputs of 2^3-word LUTs. As shown in Fig. 3, the 13-bit outputs of the proposed 2^3-word LUTs have lower transition rates than that of the conventional 2^3-word LUTs, since the proposed LUTs do not require two's complement representation. Thus, the dynamic power of the proposed 2^3-word LUTs as well as that of the adders following them can be less than that of the conventional ones, since the adders do not require two's complement representation in the proposed algorithm [3].

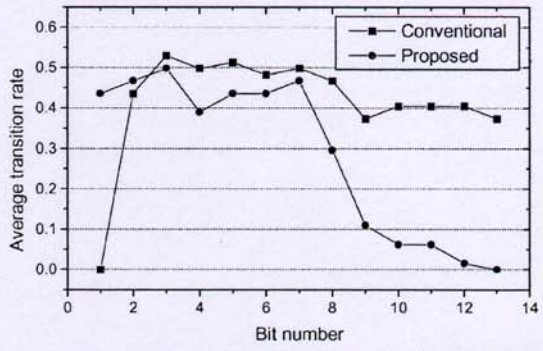

Fig. 3. Average transition rates for 13-bit outputs of 2^3-word LUTs of the conventional and the proposed 48-tap FIR filters.

TABLE II
COMPARISON OF HARDWARE COMPLEXITY AND POWER CONSUMPTION OF THE CONVENTIONAL AND THE PROPOSED 48-TAP FIR FILTERS

		Conventional filter	Proposed filter
Gate count		10,647	10,344
Power (mW)	Random input (1.2288 MHz)	12.1659	8.6962
	Speech input (44.1 kHz)	0.3877	0.2883

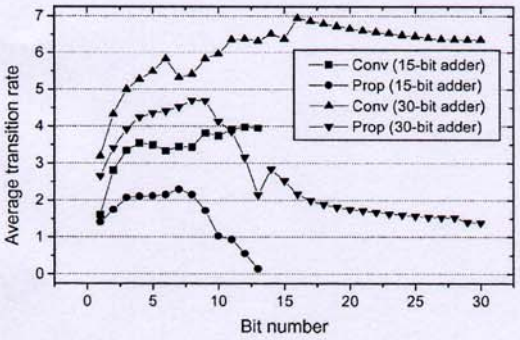

Fig. 4. Average transition rates for the output bits of 13-bit adders and 30-bit adders in 2^{12}-word LUTs and MAC blocks of the conventional and the proposed 48-tap FIR filters.

In the proposed algorithm, the FIR filter does not require two's complement representation and perform sign extension in the MAC block, since the input of the MAC block is always nonnegative. Thus, it can reduce the dynamic power consumption of the MAC block without using the sign magnitude implementation.

The conventional and the proposed 48-tap FIR filters are designed in hardware description language and synthesized to gate-level circuits with a 0.25-μm 2.5-V standard cell library [4]. The synthesized gate-level netlists are used for power consumption simulation using DesignPower (Synopsys, Inc.) [5]. The DesignPower performs gate-level circuit simulations to measure the power consumption of the two FIR filters: 2500 randomly generated 16-bit input signals and 120 000 16-bit speech input signals are used as input stimuli for the power consumption simulation. The clock frequency of the random signals and the speech signals are 1.2288 MHz and 44.1 kHz, respectively. The DesignPower can estimate the total power, which consists of net switching power, cell internal power, and leakage power by using the standard cell library supporting the power estimation [4]. The estimated total power consumption P_{total} is given by

$$P_{\text{total}} = C_L V_{DD} f^2 + I_{SC} V_{DD} + I_L V_{DD}. \quad (5)$$

In (5), C_L, V_{DD}, f, I_{SC}, and I_L represent load capacitance, supply voltage, operating frequency, short circuit current, and leakage current, respectively. The increase of transitions at each node in FIR filters causes the increase of C_L [3]. In (5), the first, the second, and the third term denote net switching power, cell internal power, and leakage power, respectively, where the net switching power and the cell internal power are referred to as the dynamic power.

Table II compares the hardware complexity and the power consumption of the conventional and the proposed FIR filters. In this letter, the gate count of a two-input NAND gate is 1. As shown in Table II, the hardware complexity of the proposed FIR filter is slightly smaller than that of the conventional FIR filter. There are two reasons for this. First, the hardware complexity of the three conventional 2^3-word LUTs with three 2×1 MUXs ($\cong 58$) is larger than that of the four proposed 2^3-word LUTs ($\cong 38$), since the conventional 2^3-word LUTs use two's complement representation. Second, the hardware complexity of three 13-bit one's complementers ($= 104$) is also slightly larger than that of the one 13-bit adder ($= 92$). Fig. 4 shows the average transition rates for the output bits of the 13-bit adders and 30-bit adders in the 2^{12}-word LUTs and MAC blocks of the conventional and the proposed FIR filters. The DesignPower is also used to estimate the transition rates for the output bits of adders in the two FIR filters by performing gate-level circuit simulations. The 2500 randomly generated 16-bit input signals, and 120 000 16-bit speech input signals are also used as input stimuli to estimate the transition rates. In Fig. 4, high transition rates larger than one result from spurious transitions (glitches). As shown in Fig. 4, the adder outputs of the proposed filter have lower transition rates than those of the conventional one. Thus, the proposed algorithm can significantly reduce the dynamic power consumption of the 2^{12}-word LUTs and MAC blocks. As shown in Table II, the proposed FIR filter consumes approximately 29% and 26% less power compared to the conventional one for the random input signals and the speech input signals, respectively, since the overall power dissipation of digital circuits using 0.25-μm technologies depends on their dynamic power dissipation. When the input signals are random, the power consumption reduction of the proposed FIR filter is maximized.

In conclusion, this letter proposes a new DA algorithm for low-power FIR implementation. The LUTs of the proposed algorithm generate only nonnegative values. Thus, if the LUT decomposition is used to reduce the hardware complexity of LUTs, the FIR filter using the proposed algorithm does not require two's complement representation in the LUTs and a MAC block. The characteristics of the proposed algorithm minimize the dynamic power consumption of FIR filters using DA. Thus, the proposed DA algorithm can be used for implementing FIR filters for hand-held applications that require low-power consumption.

REFERENCES

[1] K. K. Parhi, *VLSI Digital Signal Processing Systems: Design and Implementation.* New York: Wiley, 1999.
[2] S. A. White, "Applications of distributed arithmetic to digital signal processing: A tutorial review," *IEEE Acoust. Speech Signal Processing Mag.*, vol. 6, pp. 4–19, July 1989.
[3] A. P. Chandrakasan and R. W. Broderson, "Minimizing power consumption in digital CMOS circuits," *Proc. IEEE*, vol. 83, pp. 498–523, Apr. 1995.
[4] Taiwan Semiconductor Manufacturing Corp., *TSMC 0.25 μm 2.5 V CMOS Standard Cell Library Databook.* Hsinchu, Taiwan, R.O.C.: Taiwan Semiconductor Manufacturing Corp., 1999.
[5] A. C. McCormick, P. M. Grant, J. S. Thompson, T. Arslan, and A. T. Erdogan, "Low power receiver architectures for multi-carrier CDMA," *Proc. Inst. Elect. Eng. Circuits Devices Syst.*, vol. 149, no. 4, pp. 227–233, Aug. 2002.