A low-voltage micropower asynchronous multiplier with shift-add multiplication approach

Gwee, Bah Hwee; Chang, Joseph Sylvester; Shi, Yiqiong; Chua, Chien Chung; Chong, Kwen-Siong

2009

Gwee, B. H., Chang, J. S., Shi, Y., Chua, C. C., & Chong, K. S. (2009). A low-voltage micropower asynchronous multiplier with shift-add multiplication approach. *IEEE Transactions on Circuits and Systems I*, 56 (7), 1349-1359.

<https://hdl.handle.net/10356/91586>

<https://doi.org/10.1109/TCSI.2008.2006649>

© 2009 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE. This material is presented to ensure timely dissemination of scholarly and technical work. Copyright and all rights therein are retained by authors or by other copyright holders. All persons copying this information are expected to adhere to the terms and constraints invoked by each author's copyright. In most cases, these works may not be reposted without the explicit permission of the copyright holder. <http://www.ieee.org/portal/site> This material is presented to ensure timely dissemination of scholarly and technical work. Copyright and all rights therein are retained by authors or by other copyright holders. All persons copying this information are expected to adhere to the terms and constraints invoked by each author's copyright. In most cases, these works may not be reposted without the explicit permission of the copyright holder.

A Low-Voltage Micropower Asynchronous Multiplier With Shift–Add Multiplication Approach

Bah-Hwee Gwee, *Senior Member, IEEE*, Joseph S. Chang, Yiqiong Shi, Chien-Chung Chua, and Kwen-Siong Chong

Abstract—The design of a low-voltage micropower asynchronous (async) signed truncated multiplier based on a shift–add structure for power-critical applications such as the low-clock-rate (<4 MHz) hearing aids is described. The emphases of the design are micropower operation and small IC area, and these attributes are achieved in several ways. First, a maximum of three signed power-of-two terms accompanied with sign magnitude data representation is used for the multiplier operand. Second, the least significant partial products are truncated to yield a 16-bit signed product. An error correction methodology is proposed to mitigate, where appropriate, the arising truncation errors. The errors arising from truncation and the effectiveness of the error correction are analytically derived. Third, a low-power shifter design and an internal latch adder are adopted. Finally, a power-efficient speculative delay line is proposed to time the async operation of the various circuit modules. A comparison with competing synchronous and async designs shows that the proposed design features the lowest power dissipation ($5.86 \mu\text{W}$ at 1.1 V and 1 MHz) and a very competitive IC area (0.08 mm^2 using a $0.35\text{-}\mu\text{m}$ CMOS process). The application of the proposed multiplier for realizing a digital filter for a hearing aid is given.

Index Terms—Asynchronous (async) circuits, finite-impulse response (FIR) filter, low power, shift–add multiplier.

I. INTRODUCTION

THE CRITICAL parameters of several portable electronic devices, such as hearing instruments (hearing aids), pacemakers, intracranial pressure monitors, biosensors, etc., all with low-to-mid signal processing demands, include low-voltage (~ 1.1 V) micropower (< 1 mA) operation and small IC areas. In view of the tight power constraints, the processor used is often clocked at a low 1–2-MHz rate, and the signal conditioning circuits therein are also efficient, for example, Class D amplifiers [1]–[3].

The asynchronous (async) approach, as opposed to the prevalent synchronous (sync) approach, is an emerging design approach that potentially offers lower power dissipation and higher speed. The basic premise for these potential attributes are that

the signaling (protocol) between the different async modules is localized instead of a global clock and that spurious/glitch switching is low. Nevertheless, only few practical applications with lower power (with respect to their sync counterparts) have been demonstrated [4], [5] to date, and the exiguity of async designs is, in part, due to the nascency of sophisticated electronic design automation tools, the increased difficulty in design, verification, and testing, and the lack of commercial async cell libraries.

Of the cells in a cell library, the multiplier is usually the most complex and largest cell and dissipates the largest power in mathematically intensive tasks. In applications where the critical parameters are power dissipation and IC area and where the speed is low (< 4 MHz), the shift–add multiplication approach is a worthy design alternative to regular multiplier designs [6]. In this approach, there is no specific multiplication unit, multiplication is instead achieved by hardwired shifters and adders, and programmability is often restricted. For example, a relatively high speed programmable multiplierless finite-impulse response (FIR) filter [7] used a maximum of three canonical signed digits (CSDs). However, the range of the CSD coefficients therein are restricted, and as they are not always with three nonzero digits, some power is unnecessarily dissipated to perform an addition with zero, and undesirable spurious (glitch) switching is often generated along the addition path. The constant coefficient multiplication approach based on table lookup technique [8] requires a large memory for long wordlengths and therefore dissipates relatively high power. An async breakpoint multiplier [9] for a hearing instrument used a maximum of three logic “1” bits for the FIR filter coefficient. Although this design is micropower and features simple circuit complexity, the limitation of three “1”s renders the breakpoint multiplier highly restrictive.

In this paper, we propose a 16×16 -bit async shift–add multiplier [10] for a hearing instrument FIR filter, whose critical parameters include low-voltage (1.1 V) micropower ($5.86 \mu\text{W}$ at 1 MHz, $0.35\text{-}\mu\text{m}$ CMOS) operation and a small IC area. The proposed multiplier handles up to three signed power-of-two (SPT) terms and is unique as it utilizes a sign magnitude (SM) data representation (as opposed to two’s complement). By being able to handle three SPT terms, the proposed design is less restricted than several reported designs, for example, [9], and its primary novelty is its micropower operation—arguably the lowest multiplier design (both sync and async) reported to date and one of the smallest IC area requirement. Micropower operation is achieved in several ways. First, by employing an SM data representation, there is less power-wasting switching activities [11]. Second, a low-power transmission gate shifter structure is

Manuscript received November 28, 2006; revised January 24, 2008. First published November 11, 2008; current version published July 01, 2009. This paper was recommended by Associate Editor V. Kursun.

B.-H. Gwee and J. S. Chang are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore (e-mail: ebhgwee@ntu.edu.sg; ejschang@ntu.edu.sg).

Y. Shi is with the Integrated Systems Research Laboratory, School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore (e-mail: shiy0003@ntu.edu.sg).

C.-C. Chua is with STMicroelectronics, Singapore 554574, Singapore.

K.-S. Chong is with Temasek Laboratories@NTU, Nanyang Technological University, Singapore 639798, Singapore (e-mail: ks-chong@ntu.edu.sg).

Digital Object Identifier 10.1109/TCSI.2008.2006649

adopted [12]. Third, the partial product terms are truncated to obtain a fixed 16-bit signed product, thereby reducing $\sim 50\%$ of the hardware. Fourth, our latch adder (LA) [13], [14] is employed, and as it features an adder with an integrated latch, it dissipates less power and occupies less IC area than the usual independent adder and independent latch. Fifth, a transparent latch (T-Latch) is proposed and used to block unnecessary switching. Finally, a novel power-efficient speculative delay line is proposed and used to time the multiplier asynchronously. In this paper, the error arising from truncation is analyzed, and a simple error-correction algorithm to mitigate the effects of truncation is proposed—to our knowledge, the quantization error of a truncated multiplier with SPT terms embodying an SM representation remains uninvestigated in literature. The proposed design is verified by simulations and on the basis of measurements on a prototype IC. The merits of the design are demonstrated by comparing it against other reported sync and async multiplier designs, specifically its lowest power dissipation and its very competitive IC area requirement. We also show that a FIR filter embodying the proposed multiplier yields a magnitude response appropriate for many applications, including hearing instruments.

This paper is organized as follows. Section II describes the SM shift-add multiplication based on three SPT terms. An analytical study of the truncation error and correction based on three SPT terms is presented in Section III. The hardware implementation of the proposed multiplier is thereafter presented in Section IV. The simulation and measurement results are presented in Section V. Finally, conclusions are drawn in Section VI.

II. SHIFT-ADD MULTIPLIER DESIGN

Let X and Y be the N -bit multiplicand and multiplier operands, respectively, and they adopt the SM representation

$$X = (-1)^{x_{N-1}} \sum_{i=0}^{N-2} (x_i 2^{-N+i+1}) \quad (1a)$$

$$Y = (-1)^{y_{N-1}} \sum_{j=0}^{N-2} (y_j 2^{-N+j+1}) \quad (1b)$$

where i is the bit position of the multiplicand operand, j is the bit position of the multiplier operand, and $x_i, y_j \in \{0, 1\}$.

The multiplication product Z is as follows:

$$\begin{aligned} z = X \cdot Y &= \left[(-1)^{x_{N-1}} \sum_{i=0}^{N-2} (x_i 2^{-N+i+1}) \right] \\ &\cdot \left[(-1)^{y_{N-1}} \sum_{j=0}^{N-2} (y_j 2^{-N+j+1}) \right] \\ &= (-1)^{x_{N-1}} (-1)^{y_{N-1}} \sum_{i=0}^{N-2} \sum_{j=0}^{N-2} (x_i y_j 2^{-N+i+1} 2^{-N+j+1}) \\ &= (-1)^{z_{2N-2}} \sum_{k=0}^{2N-3} (z_k 2^{-2N+k+2}) \end{aligned} \quad (2)$$

where k is the bit position of the product and $z_k \in \{0, 1\}$.

By using the shift-add approach, the multiplier operand can be expressed as a minimum combination of several SPT terms, and the multiplication process simply involves binary shift and add/subtract operations on the multiplicand operand. The multiplier operand expressed in SPT terms is as follows:

$$Y = \sum_{m=1}^L \left[y(m) 2^{-f(m)} \right] \quad (3)$$

where $y(m) \in \{-1, 1\}$ is the m th SPT term in the multiplier operand, $f(m)$ is an integer representing the weight of the m th SPT term, and L is the total number of SPT terms in the multiplier operand.

It has been shown [15] that three or less SPT terms are usually sufficient for many filter applications, and on this basis, $\max(L) = 3$ is adopted in the proposed multiplier.

To simplify the hardware for a signed multiplication with SM data representation, we modify the SPT representation of the multiplier operand in (3) with sign bit extraction

$$Y = (-1)^{SG} \left(2^{-f(L)} + 2^{-f(m)} \sum_{m=1}^{L-1} \left[\frac{y(m)}{y(L)} \right] \right) \quad (4)$$

where $SG = \begin{cases} 1, & \text{for } y(L) = -1 \\ 0, & \text{for } y(L) = 1 \end{cases}$ is the sign bit extracted from the most significant SPT term and determines the sign of the multiplier operand.

The extracted sign bit and the sign of the multiplicand operand are used to determine the sign of the product with a simple XOR gate. Note that, as the sign of the first partial product generated is always positive, the adder/subtractor circuit can be simplified.

To further reduce power dissipation and using the modified representation in (4), we truncate the least significant partial products during multiplication, at the expense of quantization error, to obtain a 16-bit product

$$\begin{aligned} Z &= \left[(-1)^{x_{N-1}} \sum_{i=0}^{N-2} x_i 2^{-N+i+1} \right] \\ &\cdot \left\{ (-1)^{SG} \sum_{m=1}^L \left[\frac{y(m)}{y(L)} 2^{-f(m)} \right] \right\} \\ &= (-1)^{x_{N-1}} (-1)^{SG} \sum_{i=0}^{N-2} \sum_{m=1}^L \left(x_i \frac{y(m)}{y(L)} 2^{-N+i+1} 2^{-f(m)} \right) \\ &\cong (-1)^{x_{N-1}} (-1)^{SG} \sum_{i=0}^{N-2} \sum_{m=1}^L \left(t(x_i, y(m)) 2^{-N+i+1} 2^{-f(m)} \right) \end{aligned} \quad (5)$$

where

$$t(x_i, y(m)) = \begin{cases} x_i \frac{y(m)}{y(L)}, & \text{if } N - i - 1 + f(m) \leq 15 \\ 0, & \text{else.} \end{cases}$$

The quantization error arising from the truncation of the least significant portion of the partial products for a standard multiplier with a two's complement representation is well established [16]. However, in this case where the multiplier is designed with

the same truncation and with SPT terms that embody an SM representation, the quantization error remains, to our knowledge, uninvestigated. Interestingly, as the multiplier operand is limited to three SPT terms (there is a maximum of three rows of partial products), the error caused by truncating the positive and negative SPT terms of these partial products has opposite effects and somewhat negates each other's error. Practically, this implies that, when the accumulation instruction is executed in a FIR filter, a smaller quantization error results. We will now analytically investigate and derive the variance of these errors. We also propose a simple error-correction algorithm to shape the variance of the error, hence mitigating the worst case errors.

III. TRUNCATION ERROR ANALYSIS AND CORRECTION

In this section, by means of statistical analysis, the error arising from the truncation of the least significant portion of the partial products is derived. The error analysis here is different from that of the standard design [16] because SPT terms with an SM representation are used.

As a preamble to the analysis, note the following. First, the sign of the partial products generated can either be positive or negative, hence possibly requiring both addition and subtraction operations to compute the final product. Second, the number of addition/subtraction operations used for each multiplication is a function of the number of SPT terms L in the multiplier operand. Finally, the maximum number of SPT terms allowed is three in the proposed multiplier.

From the aforementioned preamble, consider now the analysis. The total probability for the three SPT terms is as follows:

$$P_0 + P_1 + P_2 + P_3 = 1 \quad (6)$$

where P_0 , P_1 , P_2 , and P_3 are the probabilities of the multiplier operand having zero, one, two, or three SPT terms.

For a 16×16 -bit multiplication with an SM data representation, the magnitude of the final product without truncation is represented by two segments: Z_h comprising the 15 MSBs and Z_l comprising the 15 LSBs. In the case of truncation where the least significant partial products are discarded and the final product is the 15-bit Z_t , the absolute value of the truncation error ε is as follows:

$$\varepsilon = |Z_h - Z_t|. \quad (7)$$

Assuming that both the multiplicand and multiplier operands have equal probability to be positive or negative (i.e., the product can be positive or negative with equal probability), the mean value of the truncation error is then

$$\bar{\varepsilon} = \frac{1}{2}\varepsilon + \frac{1}{2}(-\varepsilon) \quad (8)$$

and is always zero. Consider now, in turn, the variance of the truncation error σ^2 for four different numbers of SPT terms $L = 0, 1, 2$, and 3.

Case 1: $L = 0$: The multiplier operand contains zero SPT term. As $Z_t = Z_h = 0$, $\varepsilon = 0$, and $\sigma^2 = 0$.

Case 2: $L = 1$: The multiplier operand contains one SPT term. Due to the sign bit extraction, the first nonzero bit in

$$\begin{array}{r|rrrr} & c_{14} & c_{13} & c_{12} & \dots & c_1 & c_0 \\ f_{29} & f_{28} & \dots & f_{16} & f_{15} & f_{14} & f_{13} & \dots & f_2 & f_1 & f_0 \\ + & g_{29} & g_{28} & \dots & g_{16} & g_{15} & g_{14} & g_{13} & \dots & g_2 & g_1 & g_0 \\ \hline z_{29} & z_{28} & \dots & z_{16} & z_{15} & z_{14} & z_{13} & \dots & z_2 & z_1 & z_0 \end{array}$$

Fig. 1. Addition of the two partial products.

Fig. 2. Variance of truncation error σ^2 for the case of $L = 2$ with and without correction.

the multiplier operand would be positive. The final product is simply a shifted version of the multiplicand operand, and no addition/subtraction operation is required. Hence, as $Z_t = Z_h$, $\varepsilon = 0$, and $\sigma^2 = 0$.

Case 3: $L = 2$: The multiplier operand contains two SPT terms with the first nonzero bit being always positive, and two partial products will be generated. As the sign of the second nonzero bit is random, the addition or subtraction operation between the two partial products has equal probability.

Consider first the addition operation shown in Fig. 1, where the two partial products are $F = f_{29}, \dots, f_0$ and $G = g_{29}, \dots, g_0$, the final product is $Z = z_{29}, \dots, z_0$, and c_n is the carry bit generated by $f_n + g_n$ for $n = 0, \dots, 14$. It can be seen that, if the least significant partial products are truncated, the carry bit c_{14} is zero. Hence

$$\varepsilon = \begin{cases} z_{15}, & \text{for } c_{14} = 1 \\ 0, & \text{for } c_{14} = 0 \end{cases} \quad (9)$$

where z_{15} is the LSB of the final multiplication product.

In the case of a subtraction operation, the truncation error depends on the borrow bit generated by bit 14 of the partial products (i.e., b_{14} is the borrow bit generated by f_{14} and g_{14} in the subtraction operation) and is as follows:

$$\varepsilon = \begin{cases} z_{15}, & \text{for } b_{14} = 1 \\ 0, & \text{for } b_{14} = 0. \end{cases} \quad (10)$$

In short, σ^2 depends on the probability of c_{14} and b_{14} being generated.

Following the derivations given in the Appendix, the continuous line in Fig. 2 shows σ^2 for the case of $L = 2$ as a function of P_x , where P_x is the probability of "1"s in bit 14, $\dots$, bit 0 in the multiplicand operand. As expected, σ^2 increases as P_x

Fig. 3. Variance of truncation error σ^2 for the case of $L = 3$ with and without correction.

increases. This is because an increase in P_x increases the probability of the number of “1”s in the partial product and, hence, a higher probability of generating c_{14} .

To reduce the variance, we propose a simple correction scheme that involves adding a 1 to the LSB for $P_x > 0.5$. Hardwarewise, this is obtained by connecting the carry-in of the first adder to V_{DD} (the subtraction operation is unaffected). As shown by the dashed line in Fig. 2, the worst case variance of the truncation error without the correction scheme is reduced by 46% from 0.96 LSB^2 to 0.5 LSB^2 when the proposed correction scheme is employed.

Case 4: $L = 3$: The multiplier operand contains three SPT terms with the first nonzero bit being always positive, and three partial products will be generated. As the sign of the second and third nonzero bits is random, the four possible operation combinations (each with equal probability of 25%) among the three partial products are the following:

- 1) Partial Product 1 + Partial Product 2 + Partial Product 3;
- 2) Partial Product 1 + Partial Product 2 – Partial Product 3;
- 3) Partial Product 1 – Partial Product 2 + Partial Product 3;
- 4) Partial Product 1 – Partial Product 2 – Partial Product 3.

Following the derivations given in the Appendix, Fig. 3 (continuous line) is a plot of (26) and shows σ^2 for the case of $L = 3$ as a function of P_x . As expected and as in the case for $L = 2$ for the same reason, the variance increases as P_x increases.

As in the case $L = 2$, our simple proposed correction scheme of adding a 1 to the LSB may be applied for $P_x > 0.5$. As shown by the dashed line in Fig. 3, the worst case variance of the truncation error is reduced by $\sim 26\%$ from 1.95 LSB^2 to 1.45 LSB^2 . In summary, for cases $L = 2$ and $L = 3$, the worst case variance of the truncation error can be large, and by means of the simple correction scheme, the worst case variance is significantly reduced.

IV. HARDWARE IMPLEMENTATION

Fig. 4 shows the circuit block diagram of the proposed multiplier with a maximum of three SPT terms. The inputs are the multiplicand operands, and the control signals are *REQ* (Request—to initiate multiplication), *EN1* (Enable1—the second SPT term exists, thereby enabling the processing of

the second SPT term), *EN2* (Enable2—the third SPT term exists, thereby enabling the processing of the third SPT term), *CTL1* (Control1—to control the shift operation of SHIFT MODULE1), *CTL2* (Control2—to control the shift operation of SHIFT MODULE2), *CTL3* (Control3—to control the shift operation of SHIFT MODULE3), *SUB1* (Subtract1—the second SPT term is negative, thereby performing subtraction operation in ADD/SUB1), *SUB2* (Subtract2—the third SPT term is negative, thereby performing subtraction operation in ADD/SUB2), *CORRECTION* (see later), and *SG* (the sign bit). The outputs are *PRODUCT* [15:0] and *ACK* (Acknowledge—to indicate the completion of multiplication). Note that, because the multiplier operand can be predecoded and stored directly as control signals (rather than actual 16-bit coefficients in memory), a decoding circuit hardware is not required. This approach results in simpler hardware as there is no need to convert the coefficients to control signals.

The async approach is adopted in the multiplier primarily for control simplicity and for low power dissipation. These are attributed to two reasons. First, the async circuit is essentially a self-timed circuit with inherent timing control. By means of our earlier proposed LAs [13], [14] and the async operation, the LAs therein are controlled adaptively (for four different cases) by a novel speculative delay line (see later) to control the async multiplier asynchronously. Note that no additional latch (as a separate circuit element) is required to latch the intermediate results, hence resulting in low hardware overhead and low power dissipation. Second, fine-grain gating is innate in async circuits. By means of async operation, the proposed multiplier operates such that only the modules that are necessary during the multiplication process are enabled and the remaining modules are disabled, thereby blocking unnecessary switching. In this fashion, power dissipation is reduced and also partly because glitch/spurious switching is virtually eliminated.

The overall hardware is simple, comprising only three shift modules with a T-Latch at the input, two LAs/subtractors (in a carry-ripple structure), the speculative delay line, and an output multiplexer. All circuit modules herein handle 15-bit data while the sign bit is determined from a simple XOR gate. The applicability of the correction scheme depends on the statistical range of the input signals. Where the input signal range is high, the correction scheme should be applied, and this simply involves connecting the “*CORRECTION*” signal (the carry-in signal in ADD/SUB1) to logic high—there is no hardware cost. We will further discuss how to practically handle the errors arising from truncation and the use of SPT terms in an example in Section V.

The operation of the proposed multiplier will now be described with some emphasis on the pertinent design approaches adopted to reduce power dissipation. When the *REQ* is asserted to initialize a multiplication process, the T-Latch will first synchronize the multiplicand operand (see Fig. 4) to the three shift modules, and each module handles one SPT term. The SPT terms are weighted more heavily from Shift Module 1 to Shift Module 3. As it is known that Shift Module 1 always generates an output greater than Shift Modules 2 and 3, subtraction can be performed without the need for magnitude comparison—the circuit path can hence be simple and the power dissipation low. Put simply, a comparator is not required for the SM subtraction,

Fig. 4. Architecture of the proposed async shift-add multiplier.

Fig. 5. (a) Shifter design. (b) T-Latch.

and the probability of a high degree of switching in the proposed design is low. When the multiplication process is complete, the *ACK* signal will be generated. The delay of the multiplier is considered from *REQ* = 1 to *ACK* = 1.

The shifter modules shown in Fig. 5(a) are constructed with a four-level logarithmic structure transmission-gate multiplexer (with a low-power stage arrangement [12]) to enable the input operands to shift to 16 different positions. The simple T-Latches

shown in Fig. 5(b) are employed at the input of each shifter module to synchronize the multiplicand input to reduce spurious switching. The shifters are also buffered to maintain signal integrity (drive).

Our LAs [without the weak inverter in the carry-out signal as shown in Fig. 6(a)] [13], [14] are employed, resulting in lower power dissipation than a conventional design. This is because only the Sum signal needs to be connected to the next module; hence, the weak inverter is only required in the Sum signal to latch the data (when the *EA* signal is inactivated). As speed is not a critical parameter (< 4 -MHz clock rate), small transistor aspect ratios ($W/L = 2$) in the LAs are selected, to keep the load capacitances small. These LAs are timed and enabled after a certain delay determined from the delay line associated with the LAs. This is to reduce spurious switching due to the poorly synchronized outputs from the different shift modules to the ADD/SUB modules.

Fig. 6(b) shows the proposed power-efficient speculative delay line that controls the async operation of the proposed multiplier. The novelty here is the matching of the delay to the specificities of the input signal. Specifically, a longer delay is allocated for a larger number of SPT multiplications, i.e., the delay allowed for three SPT terms is longest and shortest for one SPT term (and zero delay for zero SPT term). The single worst case delay is partitioned into three completion delays and is enabled chronologically. As the enable control signals,

Fig. 6. (a) LA. (b) Proposed speculative delay line.

$EN1$ and $EN2$, are stored in the control words, additional abortion circuits [17] are not required to select the respective completion signal. The hardware design is hence efficient in the sense that the AND gates that enable the ADD/SUB modules are used as part of the delay while simultaneously serving as blocking gates. The latter ensures that the REQ signal does not unnecessarily toggle the entire delay line on all occasions.

Fig. 7(a)–(c) shows the waveform of the pertinent control signals for the three different cases ($L = 1, 2$, and 3) in the proposed multiplier operating with a four-phase async handshake protocol. For case $L = 1$, no addition is required (the LAs are disabled); hence, the delay is the fastest [see Fig. 7(a)]. For case $L = 2$, only two SPT terms are required to be added. In this case [see Fig. 7(b)], $EN1$ is coded to 1 to generate the $EA1$ signal to enable the LAs in the ADD/SUB1, and LAs in the ADD/SUB2 remain disabled. For case $L = 3$, all three SPT terms have to be added, and its delay is the longest. In this case [see Fig. 7(c)], both $EN1$ and $EN2$ are coded to 1, and the LAs in the ADD/SUB1 will first be enabled ($EA1 = 1$), subsequently followed by the LAs in the ADD/SUB2 ($EA2 = 1$). For all cases, when the $REQ = "0"$, the ACK signal is reverted to "0."

The different ways in which these modules are enabled allow for reducing spurious switching and increase the speed of the

multiplier. For example, in a single shift operation ($L = 1$), ADDER DELAY 1 and ADDER DELAY 2 are disabled, and when the REQ is asserted, only the SHIFT DELAY portion toggles. Subsequently, after the SHIFT DELAY, the acknowledge signal ACK is generated.

Note that the operation in ADD/SUB 1 is usually completed earlier than the worst case delay. ADDER DELAY 1 in Fig. 6 is designed to be faster than ADDER DELAY 2. This allows ADD/SUB2 to operate earlier, thereby reducing the overall worst case delay. Furthermore, the reset cycle can be completed in a shorter time, because one of the inputs of the AND gate before the ADDER DELAY 2 is connected to REQ .

In summary, the proposed speculative delay line improves the performance of the async multiplier and is more power efficient (over reported delay lines).

V. RESULTS

The proposed async shift-add multiplier is verified by means of HSPICE and NANOSIM computer simulations at 1.1 V and 1 MHz and on the basis of measurements on a prototype IC embodying the proposed multiplier. The microphotograph of the prototype IC is shown in Fig. 8.

Fig. 7. Waveforms for the control signals for different cases in the proposed multiplier. (a) Case $L = 1$. (b) Case $L = 2$. (c) Case $L = 3$.

Fig. 8. Microphotograph of the prototype IC of the proposed multiplier.

Table I depicts a tabulation of several parameters of the proposed async multiplier against the reported designs in [5], [14], and [18], and for the sake of clarity, the parameters are normalized with respect to the proposed multiplier. For a fair comparison, the reported designs are simulated with the same conditions and with the same IC fabrication process parameters.

Five hundred sets of random multiplicand and multiplier operands are input to the different multipliers except in the case of the breakpoint multiplier where the random multiplier operands have a maximum of three “1”s to accommodate its very restricted data representation. Note that the multiplicand operands applied to the proposed design are represented in SM while their equivalents (operands) are represented in two’s complement for the other multipliers. To emulate the operating conditions of a typical FIR filter, the ratio of three, two, and one SPT terms is 63%, 12%, and 25%, respectively.

TABLE I
PRELAYOUT-POWER, AVERAGE SWITCHING, NUMBER OF TRANSISTORS, AVERAGE DELAY, AND PDP AT $V_{DD} = 1.1$ V AT 1 MHz (NORMALIZED TO THE PROPOSED DESIGN)

	Power	Average Switching	Area	Average Delay	PDP
Proposed Async	1.00	1.00	1.00	1.00	1.00
Async Bk-pt [9]	1.46	1.42	0.78	0.72	1.05
Sync Trunc [14]	1.63	1.46	1.71	0.51	0.83
Sync [18]	2.56	2.26	3.18	0.71	1.79

TABLE II
COMPARISON OF THE AVERAGE DELAY FOR VARIOUS NUMBER OF SPT TERMS

	Average Delay (ns)		
	Case 1*	Case 2*	Case 3*
Proposed Async	116	104	75
Async Bk-pt [9]	84	81	69
Sync Trunc [14]	59 [#]	59 [#]	59 [#]
Sync [18]	82 [#]	82 [#]	82 [#]

* Case 1: Three SPTs (63%), two SPTs (12%), one SPT (25%)

Case 2: Three SPTs (25%), two SPTs (50%), one SPT (25%)

Case 3: Three SPTs (25.6%), two SPTs (17.6%), one SPT (56.8%)

[#] Worst case delay for sync clock system

From Table I, the following remarks are made.

- 1) The proposed async multiplier features the lowest power dissipation, and this is also reflected in its smallest average number of transistor switchings. The Sync multiplier, as expected, dissipates the highest power, partly because it is a full untruncated 16×16 -bit multiplier. The async truncated design also features a lower power dissipation ($\sim 39\%$) than that of the sync truncated multiplier due to its simpler structure and reduced spurious switching. However, the sync truncated multiplier is more general (with a full 16-bit multiplier operand data representation). The proposed design also dissipates lower power ($\sim 31\%$) than that of the reported breakpoint multiplier, despite the former being far more general and featuring less quantization error than the latter. The lower power dissipation attribute is, in part, due to the LAs and the power-efficient shifter structure.
- 2) The proposed async multiplier requires a relatively small number of transistors, translating to a relatively small IC area— $\sim 69\%$ and $\sim 42\%$ less than the Sync multiplier and Sync truncated multiplier, respectively. When compared with the highly restricted large quantization error async breakpoint multiplier, the proposed design is $\sim 28\%$ larger in the IC area. In short, the proposed multiplier features a very competitive IC area attribute.
- 3) The proposed async multiplier has the longest delay of all the multipliers. The average delay is 116 ns, and the cycle delay is 175 ns (including average reset delay), translating to a 5.7-MHz operation. This delay is inconsequential as the intended application of the proposed multiplier is for low speed (< 4 MHz) applications.

Table II depicts the effectiveness of the proposed speculative delay line that reduces the average delay depending on the distribution of the number of SPT terms. As delineated earlier, the

TABLE III
MEASUREMENTS ON THE PROTOTYPE IC EMBODYING THE PROPOSED
MULTIPLIER AT 1.1 V AND 1 MHz

Process	0.35 μm dual-poly four-metal CMOS process
Size of inputs	16 $\times$ 16-bit
Size of product	16-bit
Power measurement	5.86 μW
Delay measurement	222 ns
Power-delay-product	1.3 pJ
IC area	0.08 mm ²

Fig. 9. Frequency response of a low-pass FIR filter based on the standard and proposed multipliers.

speed of the multiplication process of the proposed design increases with a smaller number of SPT terms. Specifically, in Case 3, the delay is 65% of Case 1, and this compares favorably against the highly restrictive async breakpoint multiplier [9] design whose delay is 82% higher. In the case of the sync multipliers, the delay is independent of the number of SPT terms, and the delay is timed to accommodate the critical path(s).

Table III summarizes the parameters of the proposed multiplier on the basis of measurements on prototype ICs at 1.1 V and 1 MHz. The measurement shows that the power dissipation of the prototype is 5.86 μW . The measured average delay is 222 ns (in addition to 17.6 ns for the reset phase and this translates to 4.17 MHz). This delay is $\sim 25\%$ longer than simulations, and we attribute this to the unoptimized layout. The worst case delay has 300 ns for evaluation and 20 ns for reset and, hence, is a 320-ns cycle delay in total (translates to 3.13 MHz), and this still appropriate for a hearing instrument application whose clock rate is generally 1–2 MHz.

To demonstrate an application of the proposed shift-add multiplier, a practical 20-tap FIR low-pass filter embodying the proposed multiplier is simulated. Fig. 9 shows its frequency response (where the coefficients are based on three SPT terms and

TABLE IV
COMPARISON OF THE STOPBAND ATTENUATION AND POWER OF
VARIOUS FILTERS

Filter	Stop-Band Attenuation	Normalized Power
20-tap Standard	40dB	1
20-tap (Shift/Add Multiplier)	35dB	0.47
25-tap (Shift/Add Multiplier)	40dB	0.59

are truncated) and that of a standard filter based on a standard multiplier (using standard untruncated coefficients). The results are calculated using Monte Carlo simulations with 2000 sets of random data.

Fig. 9 shows that the stopband attenuation of the (20-tap) filter embodying the proposed multiplier is, on average, 5 dB less than that of the standard filter. On the basis of simulations, the power advantage of the proposed filter is 53% over the standard filter. If it is desired to obtain the same 40-dB stopband attenuation as the standard design, it would be necessary to increase the number of taps in our design by five taps to a 25-tap design. In this case, the power advantage of the proposed design is still very worthy—41% better than the standard design, as depicted in Table IV. In summary, on the basis of significant power and areas savings, we recommend the proposed multiplier to be used in filters with the same number of taps (as standard filters) if the system can tolerate a more relaxed attenuation and in filters with an increased number of taps if otherwise.

VI. CONCLUSION

A low-voltage micropower 16 $\times$ 16-bit async signed truncated multiplier has been proposed and designed based on the shift-add structure. To reduce the power dissipation, a number of low-power techniques at the system level (SM data and the sum of SPT terms representation) and low-power circuit design methodologies have been proposed and adopted. The quantization error arising from the truncation and with the limited number of SPT terms has been quantized, and an error-correction scheme to mitigate this error has been proposed. It has been shown that the proposed design depicts the lowest power dissipation compared with reported designs and that its IC area requirement is very competitive.

APPENDIX

With reference to the discussion for Case 3 ($L = 2$) in Section III, the probability of c_{14} and b_{14} being generated, followed by the variance of the truncation error σ^2 in this case will be derived. Fig. 1 shows that $c_n = 1$ if at least two of the three bits in f_n , g_n , and $c_{n-1}|_{n \neq 0}$ equal to 1. The probability of c_n being equal to 1 for $0 \leq n \leq 14$ is given in

$$P(c_n = 1) = P(f_n = 1)P(g_n = 1) + P(c_{n-1} = 1) \\ \times [P(f_n = 1)P(g_n = 0) \\ + P(f_n = 0)P(g_n = 1)], \\ \text{for } 1 \leq n \leq 14 \quad (11a)$$

$$P(c_n = 1) = P(f_0 = 1)P(g_0 = 1), \quad \text{for } n = 0 \quad (11b)$$

Fig. 10. Probability of f_n being in use.

where $P(f_n = 0) = 1 - P(f_n = 1)$.

As f_n and g_n are simply shifted versions of the multiplicand operand, only 15 out of the 30 bits are actually used, and the remaining are zeros. If P_x is the probability of each bit of the multiplicand operand being 1, the probabilities of f_n and g_n being 1 are, respectively

$$P(f_n = 1) = P(f_n \text{ in use})P_x \quad (12a)$$

$$P(g_n = 1) = P(g_n \text{ in use})P_x. \quad (12b)$$

To determine the probability of f_n being used $P(f_n \text{ in use})$, consider Fig. 10. Row 0 in Fig. 10 is the original multiplicand operand, and the following 15 rows are the shifted versions of the multiplicand operand. The first partial product can occupy any row from Rows 1 to 14 but not Row 15. This is because the first nonzero bit in the multiplier operand can be bit 1 to 14 with equal probability and can be bit 0 with zero probability—the latter case is, in fact, Case 2 ($L = 1$) because there would only be one nonzero bit. In other words, f_0 is not in use, and the probability of f_n being in use increases as n increases for $1 \leq n \leq 14$. This can be observed from the number of dots in each column in Fig. 10, and the probability of f_n being in use is hence simply

$$P(f_n \text{ in use}) = n/14, \quad \text{for } 0 \leq n \leq 14. \quad (13)$$

The second partial product also occupies one of the rows in Fig. 10 and is always located below the row occupied by the first partial product. For a given row position occupied by the first partial product, the probability of the second partial product occupying a row is easily derived and is tabulated in Table V. For example, if the first partial product is in Row 3, there is equal probability ($P = 1/12$) for the second partial product to occupy one of the rows from Rows 4 to 15.

Consider now the probability of g_n being used $P(g_n \text{ in use})$. For g_n to be in use, the second partial product must occupy a row below Row $(14 - n)$ for $1 \leq n \leq 14$, i.e.,

$$P(g_n \text{ in use}) = \begin{cases} \sum_{r=0}^n \left(\frac{1}{14} \sum_{s=0}^{13-r} \frac{1}{14-s} \right), & \text{for } 0 \leq n \leq 13 \\ 1, & \text{for } n = 14. \end{cases} \quad (14)$$

Using (11)–(14), the probability of c_n being generated $P(c_n = 1)$ in an addition operation can be determined.

Similarly, the probability of b_n being generated $P(b_n = 1)$ in the subtraction operation is given in

$$P(b_n = 1) = P(f_n = 0)P(g_n = 1) + P(b_{n-1} = 1) \times [P(f_n = 0)P(g_n = 0) + P(f_n = 1)P(g_n = 1)], \quad \text{for } 1 \leq n \leq 14 \quad (15a)$$

$$P(b_n = 1) = P(f_0 = 0)P(g_0 = 1), \quad \text{for } n = 0 \quad (15b)$$

where $P(f_n = 0) = 1 - P(f_n = 1)$.

Finally, given that the operation between the two partial products can be either an addition or subtraction with equal probability, the variance of the truncation error is given in the following and shown in Fig. 2:

$$\sigma^2 = E[\varepsilon^2] - (\bar{\varepsilon})^2 = E[\varepsilon^2] = z_{15}^2 \frac{1}{2} [P(c_{14} = 1) + P(b_{14} = 1)]. \quad (16)$$

With reference to the discussion for Case 4 ($L = 3$) in Section III, consider combination (1) where there are only additions. It can be easily shown that the worst case truncation error is given by

$$\varepsilon = \begin{cases} 2z_{15}, & \text{for } c_{14} = 2 \\ z_{15}, & \text{for } c_{14} = 1 \\ 0, & \text{for } c_{14} = 0 \end{cases} \quad (17)$$

where z_{15} is the LSB of the final multiplication product.

Note that, in general, the probabilities for $c_n = 2, 1$, and 0 are given in

$$P(c_n = 2) = P(c_{n-1} = 1)P(f_n = 1)P(g_n = 1) \times P(h_n = 1) + P(c_{n-1} = 2) \times [P(f_n = 1)P(g_n = 1) + P(f_n = 1)P(g_n = 0) \times P(h_n = 1) + P(f_n = 0) \times P(g_n = 1)P(h_n = 1)], \quad \text{for } 1 \leq n \leq 14 \quad (18a)$$

$$P(c_n = 2) = 0, \quad \text{for } n = 0 \quad (18b)$$

$$P(c_n = 1) = P(c_{n-1} = 0) \times \{P(f_n = 1)P(g_n = 1) + P(h_n = 1)[P(f_n = 1)P(g_n = 0) + P(f_n = 0)P(g_n = 1)]\} + P(c_n = 1)[P(f_n = 1)P(h_n = 0) + P(h_n = 1)P(g_n = 0) + P(g_n = 1)P(f_n = 0)] + P(c_{n-1} = 2) \times \{P(f_n = 0)P(g_n = 0) + P(h_n = 0)[P(f_n = 1)P(g_n = 0) + P(f_n = 0)P(g_n = 1)]\}, \quad \text{for } 1 \leq n \leq 14 \quad (19a)$$

TABLE V
PROBABILITY OF THE SECOND PARTIAL PRODUCT OCCUPYING A GIVEN ROW

		Row position of the 2 nd partial product														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Row position of the 1 st partial product	1	0	1/14	1/14	1/14	1/14	1/14	1/14	1/14	1/14	1/14	1/14	1/14	1/14	1/14	1/14
	2	0	0	1/13	1/13	1/13	1/13	1/13	1/13	1/13	1/13	1/13	1/13	1/13	1/13	1/13
	3	0	0	0	1/12	1/12	1/12	1/12	1/12	1/12	1/12	1/12	1/12	1/12	1/12	1/12
	4	0	0	0	0	1/11	1/11	1/11	1/11	1/11	1/11	1/11	1/11	1/11	1/11	1/11
	5	0	0	0	0	0	1/10	1/10	1/10	1/10	1/10	1/10	1/10	1/10	1/10	1/10
	6	0	0	0	0	0	0	1/9	1/9	1/9	1/9	1/9	1/9	1/9	1/9	1/9
	7	0	0	0	0	0	0	0	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
	8	0	0	0	0	0	0	0	0	1/7	1/7	1/7	1/7	1/7	1/7	1/7
	9	0	0	0	0	0	0	0	0	0	1/6	1/6	1/6	1/6	1/6	1/6
	10	0	0	0	0	0	0	0	0	0	0	1/5	1/5	1/5	1/5	1/5
	11	0	0	0	0	0	0	0	0	0	0	0	1/4	1/4	1/4	1/4
	12	0	0	0	0	0	0	0	0	0	0	0	0	1/3	1/3	1/3
	13	0	0	0	0	0	0	0	0	0	0	0	0	0	1/2	1/2
	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

$$P(c_n = 1) = P(f_0 = 1)P(g_0 = 1) + P(h_0 = 1)[P(f_0 = 1)P(g_0 = 0) + P(f_0 = 0)P(g_0 = 1)],$$

for $n = 0$ (19b)

$$P(c_n = 0) = 1 - P(c_n = 1) - P(c_n = 2). \quad (20)$$

The probabilities of f_n , g_n , and h_n being equal to 1 can be easily derived

$$P(f_n = 1) = \begin{cases} \left(\frac{n-1}{13}\right)P_x, & \text{for } 1 \leq n \leq 14 \\ 0, & \text{for } n = 0 \end{cases} \quad (21a)$$

$$P(g_n = 1) = \begin{cases} P_x, & \text{for } n = 14 \\ \sum_{r=1}^n \left(\frac{1}{13} \sum_{s=0}^{13-r} \frac{1}{13-s}\right) P_x, & \text{for } 1 \leq n \leq 13 \\ 0, & \text{for } n = 0 \end{cases} \quad (21b)$$

$$P(h_n = 1) = \begin{cases} \sum_{r=1}^{15} \left[\frac{1}{13} \sum_{s=0}^{t-3} \left(\frac{1}{13-s} \sum_{r=s}^{t-3} \frac{1}{13-r} \right) \right] P_x, & \text{for } 0 \leq n \leq 12 \\ P_x, & \text{for } n = 13 \\ P_x, & \text{for } n = 14. \end{cases} \quad (21c)$$

Hence, the variance of the truncation error for the first combination (1) is given by

$$\sigma_{(1)}^2 = E[\varepsilon^2] = (2z_{15})^2 P(c_{14} = 2) + z_{15}^2 P(c_{14} = 1). \quad (22)$$

By using a similar analysis delineated for the first combination aforementioned, the variance of the truncation error for the other three combinations (2), (3), and (4) are, respectively

$$\sigma_{(2)}^2 = E[\varepsilon^2] = z_{15}^2 P(c_{14} = 1) + z_{15}^2 P(b_{14} = 1) \quad (23)$$

$$\sigma_{(3)}^2 = E[\varepsilon^2] = z_{15}^2 P(c_{14} = 1) + z_{15}^2 P(b_{14} = 1) \quad (24)$$

$$\sigma_{(4)}^2 = E[\varepsilon^2] = (2z_{15})^2 P(b_{14} = 2) + z_{15}^2 P(b_{14} = 1). \quad (25)$$

Using (22)–(25), the overall variance for the case of $L = 3$ is as follows and is shown in Fig. 3

$$\sigma^2 = \frac{1}{4} \left(\sigma_{(1)}^2 + \sigma_{(2)}^2 + \sigma_{(3)}^2 + \sigma_{(4)}^2 \right). \quad (26)$$

REFERENCES

- [1] B.-H. Gwee, J. S. Chang, and H. Y. Li, "A micropower low-distortion digital pulsewidth modulator for a digital Class D amplifier," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 49, no. 4, pp. 245–256, Apr. 2002.
- [2] W. Shu and J. S. Chang, "THD of closed-loop analog PWM Class D amplifier," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 6, pp. 1769–1777, Jul. 2008.
- [3] T. Ge and J. S. Chang, "Modeling and technique to improve PSRR and PS-IMD in analog PWM Class D amplifiers," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 55, no. 6, pp. 512–516, Jun. 2008.
- [4] K.-S. Chong, B.-H. Gwee, and J. S. Chang, "Energy-efficient synchronous-logic and asynchronous-logic FFT/IFFT processors," *IEEE J. Solid-State Circuits*, vol. 42, no. 9, pp. 2034–2045, Sep. 2007.
- [5] Y. W. Li, G. Patounakis, K. L. Shepard, and S. M. Nowick, "High-throughput asynchronous datapath with software-controlled voltage scaling," *IEEE J. Solid-State Circuits*, vol. 39, no. 4, pp. 704–708, Apr. 2004.
- [6] H. H. Dam, A. Cantoni, K. L. Teo, and S. Nordholm, "FIR variable digital filter with signed power-of-two coefficients," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 6, pp. 1348–1357, Jun. 2007.
- [7] Z. Tang, J. Zhang, and H. Min, "A high-speed, programmable, CSD coefficients FIR filter," *IEEE Trans. Consum. Electron.*, vol. 48, no. 4, pp. 834–837, Nov. 2002.
- [8] M. J. Wirthlin, "Constant coefficient multiplication using look-up tables," *J. VLSI Signal Process. Syst.*, vol. 36, no. 1, pp. 7–15, Jan. 2004.
- [9] L. S. Nielsen and J. Sparsø, "Designing asynchronous circuits for low-power: An IFIR filter bank for a digital hearing aid," *Proc. IEEE*, vol. 87, no. 2, pp. 268–281, Feb. 1999.
- [10] C. C. Chua, B.-H. Gwee, and J. S. Chang, "A low-voltage micropower asynchronous multiplier for a multiplierless FIR filter," in *Proc. IEEE Int. Symp. Circuits Syst.*, Bangkok, Thailand, 2003, vol. 5, pp. 381–384.
- [11] A. P. Chandrakasan, "Ultra low power digital signal processing," in *Proc. 9th Int. Conf. VLSI Des.*, 1995, pp. 352–357.

- [12] K. P. Acken, M. J. Irwin, and R. M. Owens, "Power comparisons for barrel shifters," in *Proc. IEEE Int. Symp. Low Power Electron. Des.*, 1996, pp. 209–212.
- [13] J. S. Chang, B.-H. Gwee, and K. S. Chong, "A digital multiplier with reduced spurious switching by means of latch adders," US Patent 7 206 801, Apr. 17, 2007.
- [14] K. S. Chong, B.-H. Gwee, and J. S. Chang, "A micropower low-voltage multiplier with reduced spurious switching," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 2, pp. 255–265, Feb. 2005.
- [15] M. Bhattacharya and T. Saramaki, "Some observations on multiplierless implementation of linear phase FIR filters," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2003, vol. 4, pp. 193–196.
- [16] S. S. Kidambi, F. El-Guibaly, and A. Antoniou, "Area-efficient multipliers for digital signal processing applications," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 43, no. 2, pp. 90–95, Feb. 1996.
- [17] S. M. Nowick, K. Y. Yun, P. A. Beerel, and A. E. Dooply, "Speculative completion for the design of high-performance asynchronous dynamic adders," in *Proc. 3rd Int. Symp. Adv. Res. Asynchronous Circuits Syst.*, 1997, pp. 210–223.
- [18] B. Parhami, *Computer Arithmetic, Algorithms and Hardware Designs*. London, U.K.: Oxford Univ. Press, 2000.

Bah-Hwee Gwee (S'93–M'97–SM'03) received the B.Eng. degree in electrical and electronic engineering from the University of Aberdeen, Aberdeen, U.K., in 1990 and the M.Eng. and Ph.D. degrees from Nanyang Technological University (NTU), Singapore, Singapore, in 1992 and 1998, respectively.

He was a Research Engineer in a National Science and Technology Board-funded project in NTU in collaboration with SEIKO—Human Interface Engineering Laboratory from 1990 to 1993. From 1995 to 1998, he was a Lecturer with Temasek Polytechnic, Singapore. He was an Assistant Professor with the School of Electrical and Electronic Engineering, NTU, from 1999 to 2005, where he has been an Associate Professor since 2005. He was the Principal Investigator of several project grants including the Association of Southeast Asian Nations–European Union University Network Programme, the NTU–Panasonic Collaboration, the NTU–Lingköping University Collaboration, and Defence Science and Technology Agency and Temasek Laboratories research projects. His research grants amount to a total of more than US\$3M. He has filed several patents in circuit design and has one U.S. patent granted. He has been serving as an Associate Editor for the *Journal of Circuits, Systems and Signal Processing* since 2007. His research interests include low-power asynchronous microprocessor and digital-signal-processor design, Class D amplifiers, and soft computing.

Dr. Gwee was the Chairman of the IEEE Singapore—Circuits and Systems Chapter in 2005 and 2006. He has been a member of the IEEE Circuits and Systems Society DSP, VLSI, BioCAS, and Life Sciences Technical Committees since 2004. He has served in the Organizing Committees for IEEE BioCAS 2004 and IEEE Asia Pacific Conference on Circuits and Systems (APCCAS) 2006, served as the Technical Program Chair for International Symposium on Integrated Circuits 2007, and served in the steering committee for IEEE APCCAS.

Joseph S. Chang received the B.Eng. degree in electrical and computer systems engineering from Monash University, Clayton, Australia, and the Ph.D. degree from the Department of Otolaryngology, Faculty of Medicine, University of Melbourne, Melbourne, Australia.

He is currently with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, Singapore, where he was previously the Associate Dean of the Research and Graduate Studies, College of Engineering. He is also an Adjunct Professor with the Texas A&M University, College Station. He has founded two high-technology spin-off companies in medical/electroacoustics and actively works in academic and industrial research programs. His research pertains to multidisciplinary biomedical and analog/digital electronics.

Dr. Chang served as a panel member of the Thematic Strategic Research Programme, Science and Engineering Research Council, Agency for Science Technology and Research. He is the Guest Editor for an issue of the PROCEEDINGS OF THE IEEE and an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I and II and serves as a coeditor of the IEEE CIRCUITS AND SYSTEMS MAGAZINE's "Open Column." He was the General Chair of the 11th International Symposium on Integrated Circuits, Systems and Devices (2004) and cochaired the recent IEEE–National Institutes of Health Biomedical Information Science and Technology Initiative Workshop (2007).

Yiqiong Shi received the B.Eng. degree in electrical and electronic engineering from Nanyang Technological University, Singapore, Singapore, in 2004, where she is currently working toward the Ph.D. degree in the Integrated Systems Research Laboratory, School of Electrical and Electronic Engineering.

Her research interests include asynchronous VLSI design, microprocessor and digital-signal-processor design, and computer-aided-design tools for analysis and verification.

Chien-Chung Chua received the B.Eng. and M.Eng. degrees in electrical and electronic engineering from Nanyang Technological University, Singapore, in 2001 and 2004, respectively.

He is currently a Senior Integrated Circuits Designer with STMicroelectronics, Singapore, Singapore. His research interests include low-power asynchronous VLSI and application-specific integrated circuit designs.

Kwen-Siong Chong received the B.Eng., M.Phil., and Ph.D. degrees in electrical and electronic engineering from Nanyang Technological University (NTU), Singapore, Singapore, in 2001, 2002, and 2007, respectively.

He is currently a Research Scientist with Temasek Laboratories@NTU. His research interests include asynchronous VLSI designs, low-voltage low-power VLSI circuits, audio signal processing, field-programmable gate array prototyping, and digital filterbank designs.