

Power-Efficient Pipelined Reconfigurable Fixed-Width Baugh-Wooley Multipliers

Jin-Hao Tu and Lan-Da Van, *Member, IEEE*

Abstract—In this paper, we propose a pipelined reconfigurable fixed-width Baugh-Wooley multiplier design framework that provides four configuration modes (CMs): $n \times n$ fixed-width multiplier, two $n/2 \times n/2$ fixed-width multipliers, $n/2 \times n/2$ full-precision multiplier, and two $n/4 \times n/4$ full-precision multipliers. Furthermore, low-power schemes including gated clock and zero input techniques are employed to achieve the power-efficient pipelined reconfigurable design. The presented power-efficient pipelined reconfigurable fixed-width multiplier design not only generates a family of widely used multipliers but also leads to 10.59, 21.7, 28.84, and 31.58 percent power saving, on average, for $n = 8, 16, 24$, and 32 , respectively, compared with that of the pipelined reconfigurable fixed-width multiplier without using the low-power schemes. On the other hand, compared with non-reconfigurable pipelined multiplier, we can save 0.81, 12.46, 17.93, and 23.2 percent power consumption, respectively, for $n = 8, 16, 24$, and 32 .

Index Terms—Baugh-Wooley algorithm, full-precision multiplier, fixed-width multiplier, pipeline, power efficient, and reconfigurable.

1 INTRODUCTION

As growing demands on portable computing and communication systems, the power-efficient multiplier plays an important role of very large-scale integration (VLSI) systems. Among these multipliers, the basic multiplication either follows the Baugh-Wooley [1] or the Booth [2], [3] algorithms. In many digital signal processing (DSP) algorithms such as digital filters, discrete cosine transform (DCT), and wavelet transform, it is desirable to provide full-precision multiplication [4], [5], [6], [7] and fixed-width multiplication [8], [9], [10], [11], [12], [13], [14], [15], [16] that produces n -bit output product with n -bit multiplier and n -bit multiplicand with low error. A fixed-width multiplier (also referred to as single-precision multiplier) with area and power saving can be achieved either by directly truncating n least significant columns and preserving n most significant columns or by other efficient methods [8], [9], [10], [11], [12], [13], [14], [15], [16]. By the former method, significant errors will be incurred since no error compensation is considered. Thus, the latter schemes explore issues on low error and small area. Lim [8] first utilized statistical techniques to estimate and simulate the error compensation bias. However, in his analysis, the reduction and rounding errors are separately treated such that this scheme does not lead to an accurate enough error compensation bias. Note that two sources of error for the fixed-width multiplier are the reduction and rounding errors. In [9], [10], the presented work improved the error compensation bias to be more accurate and practical since the reduction and rounding errors are concurrently treated. Later, in [11], [12], [13], [14], [15], [16], many researchers analyzed an adaptive error compensation bias

under keeping $n + w$ most significant columns and proposed various fixed-width multipliers. On the other hand, much work, recently, focuses on constructing reconfigurable full-precision multipliers [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27]. In [17], [18], [19], [20], [21], [22], one reconfigurable full-precision multiplier has been proposed by the subword partitioning technique, where one $n \times n$, two $n/2 \times n/2$, or four $n/4 \times n/4$ full-precision multiplications can be performed. In [23], [24], [25], a reconfigurable full-precision multiplier consists of an array of 4×4 or 8×8 small multipliers, where the multiplier in [24] has more configuration functions than that of [23], [25]. The reconfigurable architecture [24] can provide multiple 4×4 , 8×8 , 16×16 , 32×32 , and 64×64 operations and support multiplication, MAC, addition, and data format conversion. Due to so many reconfiguration functions and variable pipeline stages, the architecture [24] leads to larger hardware design complexity. The low-power multiplier designs are debated in [26], [27], [28]. In [26], a 2D pipeline gating technique is employed to design a power-aware array multiplier that is adaptive to the high- or low-resolution operations. In [27], the power cutoff technique is employed to reduce power consumption when lower resolution multiplication is demanded. Note that the conventional reconfigurable multiplier designs [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27] are based on the full-precision multiplier infrastructure to generate the full-precision multipliers. However, it can be seen that the full-precision multiplier is much more cost-ineffective than the fixed-width multipliers [16]. In [28], a Baugh-Wooley multiplier made use of the dynamic range detection unit and truncated multiplication technique to save power consumption. Nevertheless, the proposed multiplier provided only truncated output precisions under $n \times n$ truncated multiplication and didn't discuss how to generate the full-precision multipliers and other fixed-width-type multipliers. To the best of our knowledge, we are the first one to explore the power-efficient pipelined reconfigurable fixed-width multiplier and discuss how to reconfigure the structure to generate a family of useful fixed-width and full-precision multipliers.

- The authors are with the Department of Computer Science, National Chiao Tung University, 1001 Ta Hsueh Road, Hsinchu 300, Taiwan, ROC. E-mail: ldvan@cs.nctu.edu.tw.

Manuscript received 30 Jan. 2008; revised 8 July 2008; accepted 7 Oct. 2008; published online 30 June 2009.

Recommended for acceptance by G. Constantinides.

For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number TC-2008-01-0043. Digital Object Identifier no. 10.1109/TC.2009.89.

This work is intended to provide four useful arithmetic functions by reconfiguring the low-power fixed-width multiplier structure. The four configuration modes (CMs) include:

1. $n \times n$ fixed-width multiplier,
2. two $n/2 \times n/2$ fixed-width multipliers,
3. $n/2 \times n/2$ full-precision multiplier, and
4. two $n/4 \times n/4$ full-precision multipliers.

The rest of the paper is organized as follows: The Baugh-Wooley array multiplier and subword multiplication are briefly reviewed in Section 2. In Section 3, a pipelined reconfigurable fixed-width multiplication engine with four CMs is presented. In Section 4, power reduction schemes are proposed to achieve the power-efficient pipelined fixed-width multiplier. The comparison results in terms of power reduction and area cost for $n = 8, 16, 24$, and 32 are presented in Section 5. Also, we present the main design differences among the various reconfigurable multipliers in qualitative way. Last, brief statements conclude the presentation of this paper.

2 FUNDAMENTALS OF BAUGH-WOOLEY MULTIPLIER AND SUBWORD MULTIPLICATION

Considering two 2s-complement integer operands, we can, respectively, represent an n -bit multiplicand X and an n -bit multiplier Y as follows:

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

$$Y = -y_{n-1}2^{n-1} + \sum_{i=0}^{n-2} y_i 2^i, \quad (2)$$

where $x_i, y_i \in \{0, 1\}$. The $2n$ -bit full-precision product P_{FP} can be written as

$$\begin{aligned} P_{FP} &= X \cdot Y \\ &= x_{n-1}y_{n-1}2^{2n-2} + \sum_{i=0}^{n-2} \sum_{j=0}^{n-2} x_i y_j 2^{i+j} \\ &\quad + 2^{n-1} \left(-2^{n-1} + \sum_{j=0}^{n-2} \overline{x_{n-1}y_j} 2^j + 1 \right) \\ &\quad + 2^{n-1} \left(-2^{n-1} + \sum_{i=0}^{n-2} \overline{y_{n-1}x_i} 2^i + 1 \right). \end{aligned} \quad (3)$$

Equation (3) represents the Baugh-Wooley algorithm [1], [4], [5] in which this array multiplier sums partial product bits corresponding to each weighting. The partial product array for $n \times n$ 2s-complement multiplication are depicted in Fig. 1 [16], where notation w means to keep $n + w$ most significant columns of the partial products for fixed-width multiplications. If $w = n$, the fixed-width multiplier becomes a full-precision multiplier. In this paper, we would like to reconfigure the fixed-width multiplication engine to generate four useful multipliers under the limited hardware resource.

Moreover, many DSP and computer applications demand to operate at lower resolution, where the data can be expressed in a halfword length [17], [18], [19], [20], [21], [22]. Generally, applying the subword multiplication

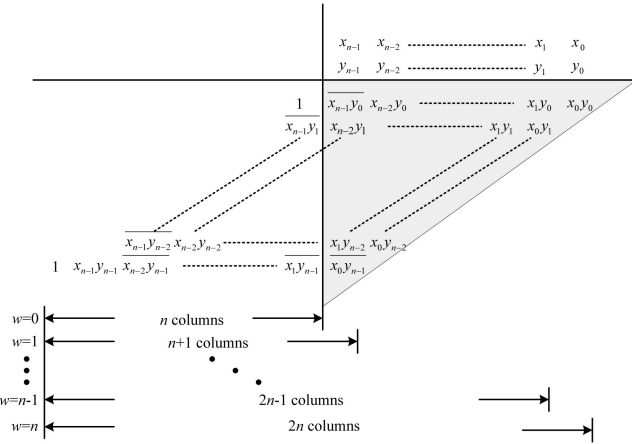


Fig. 1. Partial product array diagram for an $n \times n$ Baugh-Wooley multiplier.

scheme, we can partition an n -bit operand into two independent $n/2$ -bit operands or four independent $n/4$ -bit operands; hence, the subword multiplier can perform not only $n \times n$ full-precision multiplication but also two $n/2 \times n/2$ or four $n/4 \times n/4$ full-precision multiplications in parallel. Fig. 2 illustrates subword multiplication and the partial product array distribution [17], [18], [19], [20], [21], [22]. In Fig. 2a, two n -bit operands X and Y are partitioned into two independent pairs of $n/2$ -bit subwords, and then the two pairs of $n/2$ -bit subwords are multiplied to produce two independent n -bit products: $P_1 = X_1 Y_1$ and $P_0 = X_0 Y_0$, where the partial product array distribution is addressed in Fig. 2b. On the other hand, $n/4 \times n/4$ subword multiplication and the partial product array distribution are illustrated in Figs. 2c and 2d, respectively. To the best of our knowledge, the current subword scheme is applied only to full-precision multiplication based on the full-precision multiplier infrastructure. In the following section, we will extend this subword scheme to fixed-width and full-precision multiplication using the fixed-width prototype multiplier.

3 DESIGN OF RECONFIGURABLE FIXED-WIDTH MULTIPLIER

In this section, we begin to demonstrate how to generate four different multipliers under the limited hardware resource of the fixed-width multiplier. In this paper, we use the fixed-width multiplier in Fig. 3 as our reconfigurable multiplier prototype instead of the full-precision multiplier structure, where the fixed-width multiplier truncates partial products of the least significant part (LSP) as shown in the dashed line region of Fig. 3. In Fig. 3, three modules denoted by MUL1, MUL2, and MUL3 are used to reconfigure the following four different multipliers as listed in Table 1 through the corresponding four CMs. Thus, the proposed reconfigurable fixed-width multiplier employing MUL1, MUL2, and MUL3 is essentially different from the full-precision one [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27]. Without loss of generality, we use $n = 8$ to investigate each CM case in the following.

3.1 CM1: $n \times n$ Fixed-Width Multiplier

Configuration mode 1 (CM1) is in charge of operating $n \times n$ fixed-width multiplication that receives two n -bit numbers

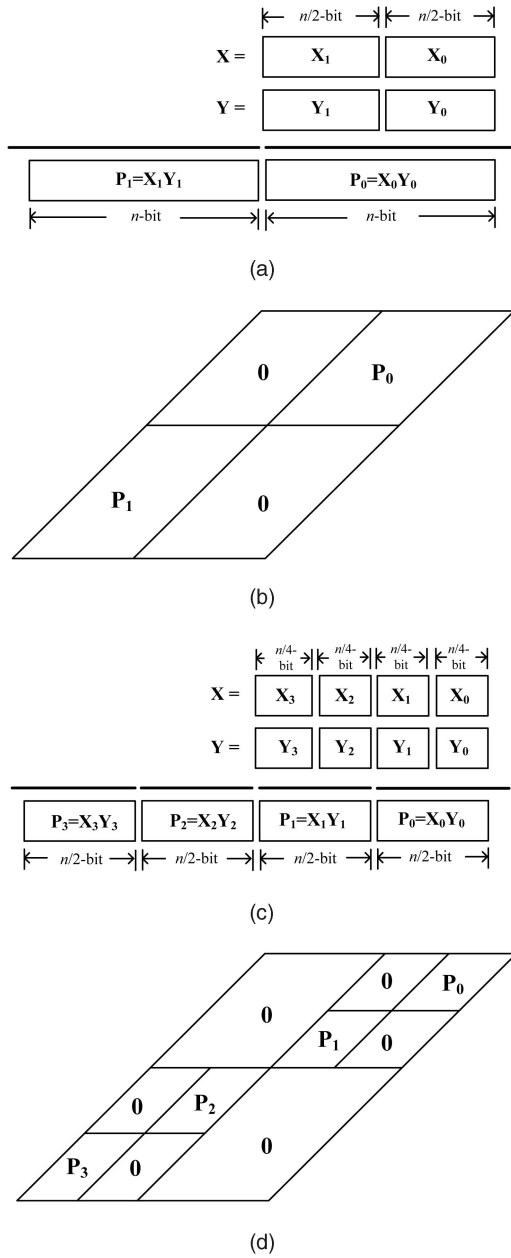


Fig. 2. Subword multiplication: (a) two $n/2 \times n/2$ multiplications, (b) two $n/2 \times n/2$ partial product array distributions, (c) four $n/4 \times n/4$ multiplications, and (d) four $n/4 \times n/4$ partial product array distributions.

and produces an n -bit product. It is known that the various fixed-width multipliers with adaptive compensation biases have been widely discussed in [11], [12], [13], [14], [15], [16]. Herein, regarding the trade-offs of the error and area cost in [16], we choose $w = 1$ (i.e., keeping $n + 1$ most significant columns) and $Q = 0$ for the prototype multiplier structure in CM1, where Q has been clearly defined in [16]. Since CM1 is confined to $w = 1$, the partial product array diagram as shown in Fig. 4a with $n = 8$ can easily be obtained from Fig. 1. As mentioned above in this section, the rest partial products are decomposed into three multiplication modules MUL1, MUL2, and MUL3 as depicted in Fig. 4b. The partial products of the three blocks are summed up independently and then the three summations are added together to produce final product. Throughout this paper, in order to completely achieve four configuration modes, we provide

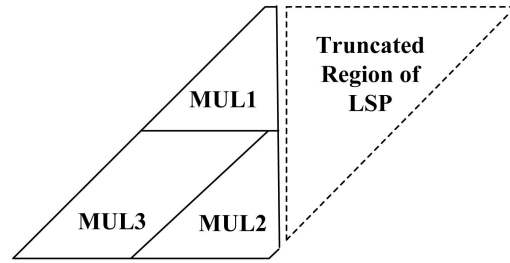


Fig. 3. Prototype structure of the proposed reconfigurable fixed-width multiplier involving MUL1, MUL2, MUL3, and discarding truncated region of LSP.

five configuration parameters CP_0 , CP_1 , CP_2 , CP_3 , and CP_4 combining with the proper partial product setting to generate other multipliers. In CM1, CP_0 , CP_1 , CP_2 , CP_3 , and CP_4 are set to 0 as shown in Fig. 4c.

3.2 CM2: Two $n/2 \times n/2$ Fixed-Width Multipliers

Configuration mode 2 (CM2) plays a role of concurrently performing two $n/2 \times n/2$ fixed-width multiplications. In this configuration mode, we need two copies of hardware resource to implement CM2. First, we have to determine which multiplier modules are suitable for two $n/2 \times n/2$ fixed-width multiplications under the constraint of the minimum number of modules and partial product configuration settings. It is manifest that MUL1 and MUL2 are suitable for two $n/2 \times n/2$ fixed-width multiplications. Due to the use of MUL1 and MUL2, the corresponding fixed-width subword operation of CM2 is illustrated in Fig. 5, where two subword products are X_1Y_0 and X_0Y_1 , and each fixed-width multiplication has $n/2$ -bit wide output. If we choose MUL3 for X_1Y_1 and either MUL1 for X_1Y_0 or MUL2 for X_0Y_1 , we can find that it is difficult to implement two input-independent fixed-width multipliers due to the same X_1 or Y_1 . Even though we can carry out one $n/2 \times n/2$ fixed-width multiplier from partial products of X_1Y_1 , larger number of configuration parameters are needed. That means lower flexibility and larger numbers of parameter settings are incurred. Once deciding the fixed-width subword product candidates, we can depict the partial product array diagram using MUL1 and MUL2 in Fig. 6a, where the partial products circled by dot-line are needed to be reconfigured in comparison with CM1. In Fig. 6a, compared with partial products of MUL1 and MUL2 of CM1, x_4y_3 , x_5y_3 , x_6y_3 , x_7y_3 , x_3y_4 , x_3y_5 , x_3y_6 , and $\overline{x_3y_7}$ are complemented, x_3y_3 is configured to zero. The configuration parameters of CM2 can be set as addressed in Fig. 6b, where CP_0 , CP_1 , and CP_2 are set to 1. The rest partial products are unchanged.

TABLE 1
Proposed Four Configuration Modes of the Reconfigurable Fixed-Width Baugh-Wooley Multiplier

Configuration Mode (CM)	Function Descriptions
CM1	$n \times n$ fixed-width multiplier
CM2	two $n/2 \times n/2$ fixed-width multipliers
CM3	$n/2 \times n/2$ full-precision multiplier
CM4	two $n/4 \times n/4$ full-precision multipliers

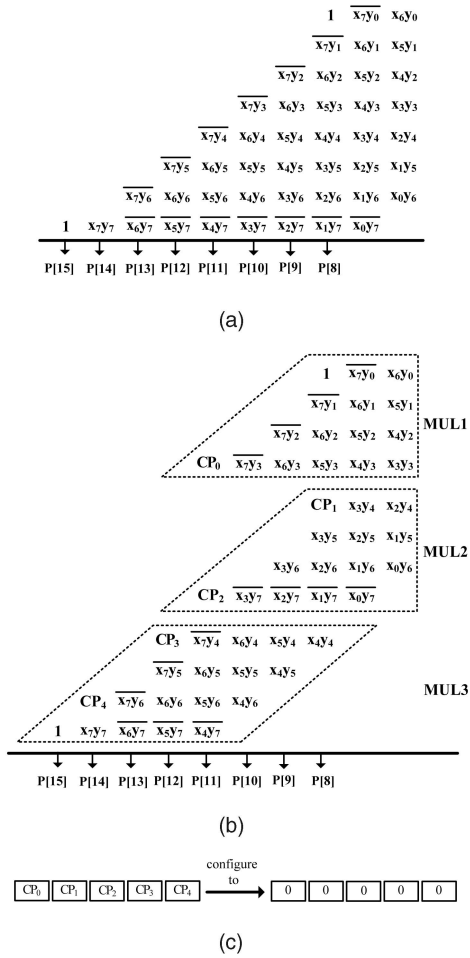


Fig. 4. (a) Partial product array diagram for $n \times n$ fixed-width multiplication; (b) proposed partial product array diagram using MUL1, MUL2, and MUL3 for CM1; and (c) configuration parameter settings.

3.3 CM3: $n/2 \times n/2$ Full-Precision Multiplier

Configuration mode 3 (CM3) serves as performing an $n/2 \times n/2$ full-precision multiplication. In behavior similar to that in CM2, the design procedures can be stated as follows: First, we have to determine which modules are suitable for $n/2 \times n/2$ full-precision multiplications with the minimum number of modules and partial product configuration settings. Under these constraints, since the proposed reconfigurable structure to implement full-precision multiplication is based on the fixed-width multiplier fabric, we can observe that just only one module, MUL3, can meet. Thus, the partial product array diagram of the MUL3 is depicted in Fig. 7, where CP_3 and CP_4 are set to 1 and 0, respectively.

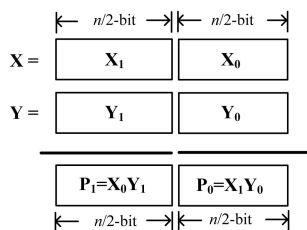


Fig. 5. Subword operation for two $n/2 \times n/2$ fixed-width multiplications.

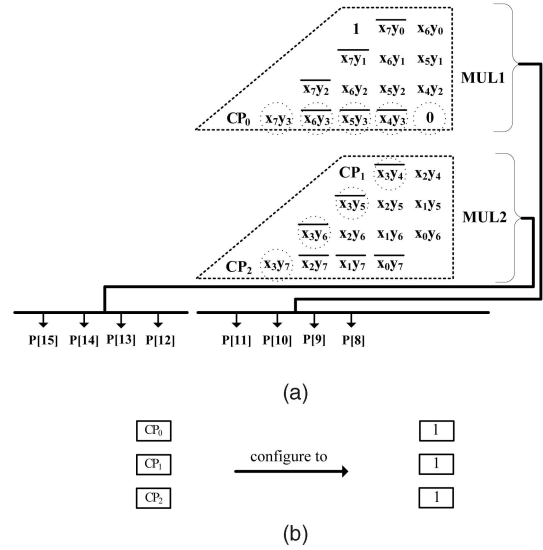


Fig. 6. (a) Proposed partial product array diagram for CM2, and (b) configuration parameter settings.

3.4 CM4: Two $n/4 \times n/4$ Full-Precision Multipliers

Configuration mode 4 (CM4) widely used in lower resolution operation serves as performing two $n/4 \times n/4$ full-precision multiplications. Under the minimum number of modules and partial product configuration setting constraints, we make use of the MUL3 to fulfill the CM4 operation. Due to the use of MUL3, the corresponding subword operation of CM4 is illustrated in Fig. 8, where two subword products are $X_2 Y_2$ and $X_3 Y_3$, and each fixed-width multiplication has $n/2$ -bit wide output. Then, the partial product array diagram of two $n/4 \times n/4$ full-precision multipliers can be obtained in Fig. 9a. In Fig. 9a, compared with partial products of the MUL3 of CM1, $x_5 y_4$ and $x_4 y_5$ are complemented, $x_6 y_4$ and $x_6 y_5$ are configured to one, and $\overline{x_7 y_4}$, $\overline{x_7 y_5}$, $x_4 y_6$, $x_5 y_6$, $\overline{x_4 y_7}$, and $\overline{x_5 y_7}$ are configured to zero. The configuration parameters of CM4 can be set as addressed in Fig. 9b, where CP_3 and CP_4 are set to 0 and 1, respectively. The rest partial products are unchanged.

The proposed pipelined reconfigurable structure for $n = 8$ is depicted in Fig. 10a, where ADD and MUX denote an adder and a multiplexer, respectively. The detailed diagrams of the corresponding MUL1, MUL2, and MUL3 are exposed in Figs. 10b, 10c, and 10d, respectively, where A, ND, HA, and FA denote an AND gate, a NAND gate, a half adder, and a

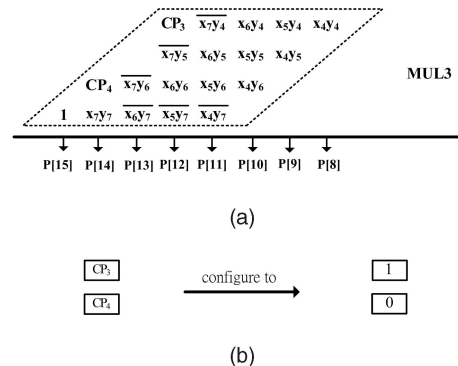


Fig. 7. (a) Proposed partial product array diagram for CM3, and (b) configuration parameter settings.

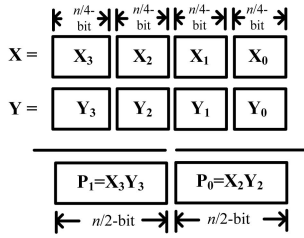
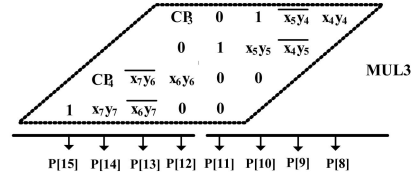


Fig. 8. Subword operation for two $n/4 \times n/4$ full-precision multiplications.

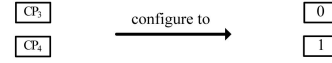
full adder, respectively; and the logic diagrams of the other processing elements are depicted in Fig. 10e. The overall structure in Fig. 10a is partitioned into three stages. The first stage is responsible for decoding the operation (OP) code to generate control signals for the next stage, where the truth table of this decoder is listed in Table 2. According to the control signals, we can manipulate three multiplication modules involving MUL1, MUL2, and MUL3 at the second stage. As shown in Figs. 10b, 10c, and 10d, since CM1 and CM2 enable MUL1 and MUL2 to compute at the same time, $t[2]$ are used to configure MUL1 and MUL2 for correct function. Similarly, since CM1, CM3, and CM4 need to enable MUL3, $t[1]$ and $t[0]$ with the values of $\{00, 10, 01\}$ are used to configure the MUL3 in accordance with three different modes. As a consequence, CP_0, CP_1 , and CP_2 can be implemented by $t[2]$, CP_3 and CP_4 can be realized by $t[1]$ and $t[0]$, respectively. In another viewpoint, from configuration parameter settings as shown in Figs. 4c, 6b, 7b, and 9b, we can easily follow the above CP implementation.

A multiplexer at the second stage selects the output of MUL3 or the concatenation output of MUL1 and MUL2, and this design will be beneficial for power saving discussed in the next section. For CM1, since we have three multiplier modules to implement $n \times n$ fixed-width multiplication for Type 1 with $\theta_{Q=0,w=1}$ [16], two adaptive compensation biases of MUL1 and MUL2 are needed to carefully control. According to the binary thresholding mentioned in [16], if each adaptive compensation bias adds a constant $K = 1/2$ for $\theta_{Q=0,w=1} = 0$, the two adaptive compensation biases are not equivalent to the compensation design as shown in [16, Fig. 5]. Thus, the design will lead to larger error for CM1 than that of adding a constant $K = 1/2$ one time. Herein, we propose subcalibration-circuit 1 (SCC1) and subcalibration-circuit 2 (SCC2) to keep away from double constant addition and to achieve this reconfiguration for $n \times n$ and $n/2 \times n/2$ fixed-width multiplications. The logic diagram of SCC1 and SCC2 as shown in Fig. 10e is little area overhead, where the truth table of SCC1 and SCC2 is tabulated in Table 3. For CM1, if $K_{m1} = 1$ and $K_{m2} = 1$ (i.e., $\theta_{Q=0,w=1} = 0$), then $SCC1 = 1$ and $SCC2 = 0$ to avoid double addition of constant $K = 1/2$. Otherwise, $SCC1 = 0$ and $SCC2 = 0$ since $\theta_{Q=0,w=1} \neq 0$. For CM2, two independent $n/2 \times n/2$ multipliers are operated in parallel. Thus, SCC1 and SCC2 follow the values of K_{m1} and K_{m2} (i.e., $SCC1 = K_{m1}$ and $SCC2 = K_{m2}$).

The third stage is in charge of accumulating the output values of MUL1, MUL2, and MUL3 for CM1 and selecting output of final product according to four CMs. In Fig. 10a, ADD1 adds the output of MUL1 and MUL2; however, the output bits of ADD1 only include carryout and ignore least significant bit due to the fixed-width output. For example, originally, $A[3:0] + B[3:0]$ will produce $\{\text{carryout}, C[3:0]\}$, but



(a)



(b)

Fig. 9. (a) Proposed partial product array diagram for CM4, and (b) configuration parameter settings.

we only need $\{\text{carryout}, C[3:1]\}$. ADD2 adds the output of ADD1 and the output of the multiplexer at the second stage to achieve CM1. We make use of the control signal $t[3]$ to determine the final correct product among different CMs. Note that the proposed reconfigurable methodology and concept can be applied to the larger bit width and used to increase configuration modes such as $n/8 \times n/8$ and $n/16 \times n/16$ multipliers while the larger word length is given. For example, from the above analysis, the conventional full-precision subword multiplication schemes [17], [18], [19], [20], [21], [22] can be applied to MUL3 to increase configuration modes including four $n/8 \times n/8$, eight $n/16 \times n/16$ full-precision multipliers, and so forth, according to the larger input word length n . On the other hand, although we discuss only 2s-complement multiplication in this paper, this reconfigurable concept can easily be extended to unsigned array multiplier.

4 DESIGN OF POWER-EFFICIENT RECONFIGURABLE FIXED-WIDTH MULTIPLIER

In this section, we further discuss how to design a power-efficient pipelined reconfigurable multiplier. As mentioned in Section 3, the multiplications of CM2, CM3, and CM4 are of power-inefficient because they invoke all hardware resource to compute. It is desirable to apply low-power schemes such that the proposed reconfigurable fixed-width multiplier possesses power-efficient capability. We apply low-power schemes including clock gating and zero input techniques to achieve power saving.

4.1 Clock Gating for the Second and Third Stages

The clock gating scheme is applied to the registers at the second and third stages of Fig. 11 in order to reduce unnecessary transitions. According to the following rules, we are able to disable the corresponding pipeline registers for power saving.

1. If CM1 is performed, the input register of MUL1, MUL2, or MUL3 is conditionally disabled (i.e., referred to gated register in Fig. 11). The disable conditions depend on which input value of the register is zero.

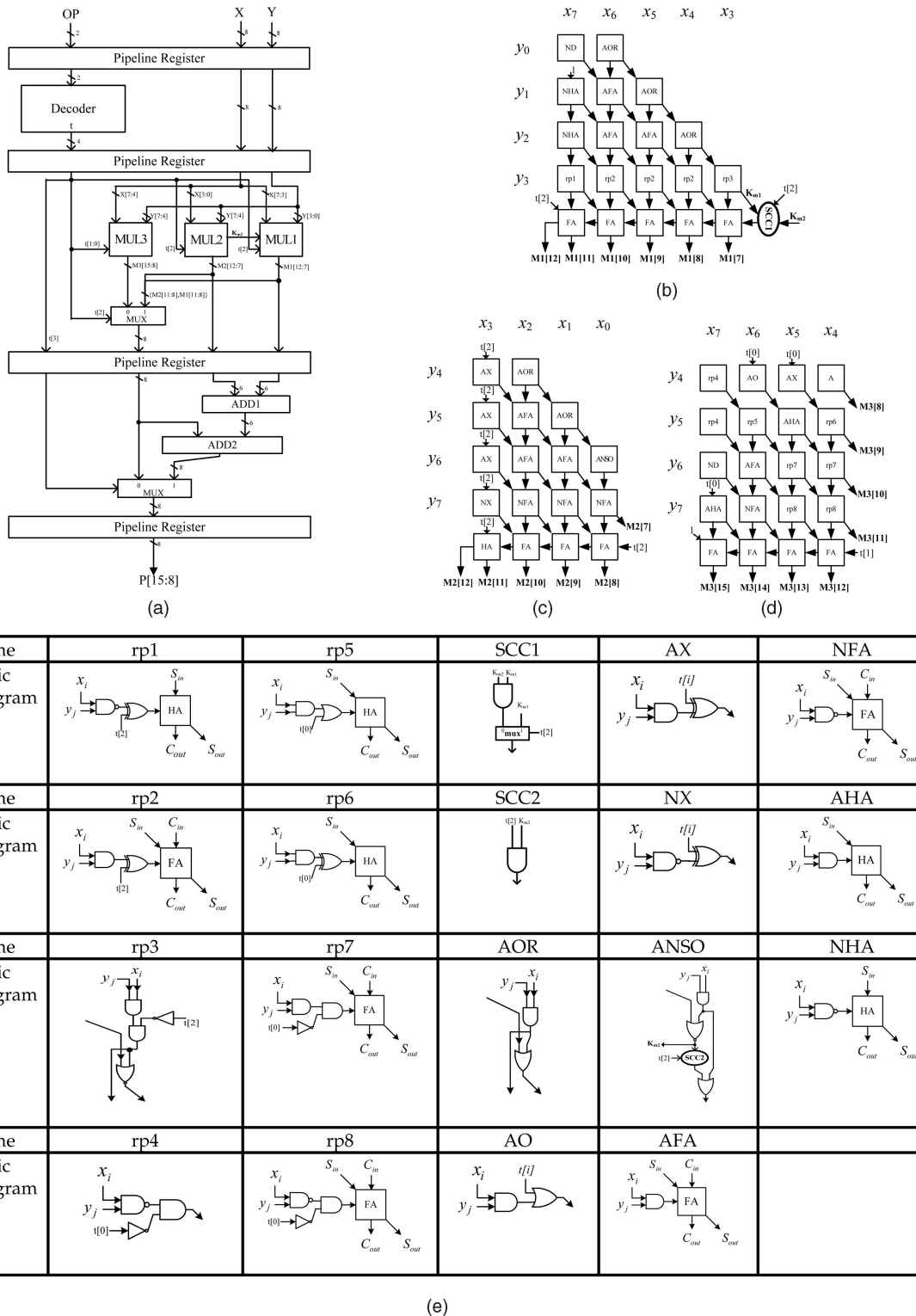


Fig. 10. (a) Proposed pipelined reconfigurable multiplier, (b) structure of MUL1, (c) structure of MUL2, (d) structure of MUL3, and (e) logic diagrams of the other processing elements.

2. If CM2 is performed, input registers of MUL3 and ADD1 can be disabled.
3. If CM3 is performed, input registers of MUL1, MUL2, and ADD1 can be disabled.
4. If CM4 is performed, input registers of MUL1, MUL2, and ADD1 can be disabled.

The penalty of this scheme is the hardware overhead. The overhead covers the duplicated input registers so as to achieve the gated register for each multiplication module. If no duplicated input register is considered, for example, CM2 with disabling MUL3 (i.e., input registers for $X[7:4]$ and $Y[7:4]$ are disabled), the outputs of MUL1 and MUL2 must be

TABLE 2
Truth Table of Decoder

OP[1:0]	t[3]	t[2]	t[1]	t[0]
00(CM1)	1	0	0	0
01(CM2)	0	1	0	0
10(CM3)	0	0	1	0
11(CM4)	0	0	0	1

wrong because MUL1 and MUL2 need $X[7:4]$ and $Y[7:4]$, respectively, to generate the product. Hence, we duplicate input register for $X[7:4]$ and $Y[7:4]$ such that the input registers of MUL1, MUL2, and MUL3 are separated in Fig. 11.

Furthermore, in CM1, since the inputs of MUL1, MUL2, and MUL3 are duplicated, we can detect zero values of input data to disable the multiplication module. The conditions of zero value of the input are described in the following:

1. If $X[7:4]$ is zero, input registers of MUL1 and MUL3 can be disabled.
2. If $X[3:0]$ is zero, input registers of MUL2 can be disabled.
3. If $Y[7:4]$ is zero, input registers of MUL2 and MUL3 can be disabled.
4. If $Y[3:0]$ is zero, input registers of MUL1 can be disabled.

Note that although one of the input operands is zero, the product of multiplication module is not equal to zero. Because some partial products are inverted as shown in Fig. 4b, the actual product outputs of the disabled MUL3 and MUL2 should be $(11110000)_2$ and $(001111)_2$, respectively. MUL1 is more particular since we must concern with partial product x_3y_3 and K_{m2} . Let us consider the following cases (\wedge, \vee denote AND and OR operators, respectively):

1. If $x_3y_3 = 0$ and $K_{m2} = 0$, the output of SCC1 is 0 such that MUL1 produces $(010001)_2$.
2. If $x_3y_3 = 0$ and $K_{m2} = 1$, the output of SCC1 is 1 such that MUL1 produces $(010010)_2$.
3. If $x_3y_3 = 1$ and $K_{m2} = 0$, the output of SCC1 is 0 such that MUL1 produces $(010010)_2$.
4. If $x_3y_3 = 1$ and $K_{m2} = 1$, the output of SCC1 is 0 such that MUL1 produces $(010010)_2$.

TABLE 3
Truth Table of Subcalibration-Circuit1 (SCC1) and Subcalibration-Circuit2 (SCC2)

K_{m1} in Fig. 10(e)	K_{m2} in Fig. 10(e)	CM1		CM2	
		Output of SCC1	Output of SCC2	Output of SCC1	Output of SCC2
0	0	0	0	0	0
0	1	0	0	0	1
1	0	0	0	1	0
1	1	1	0	1	1

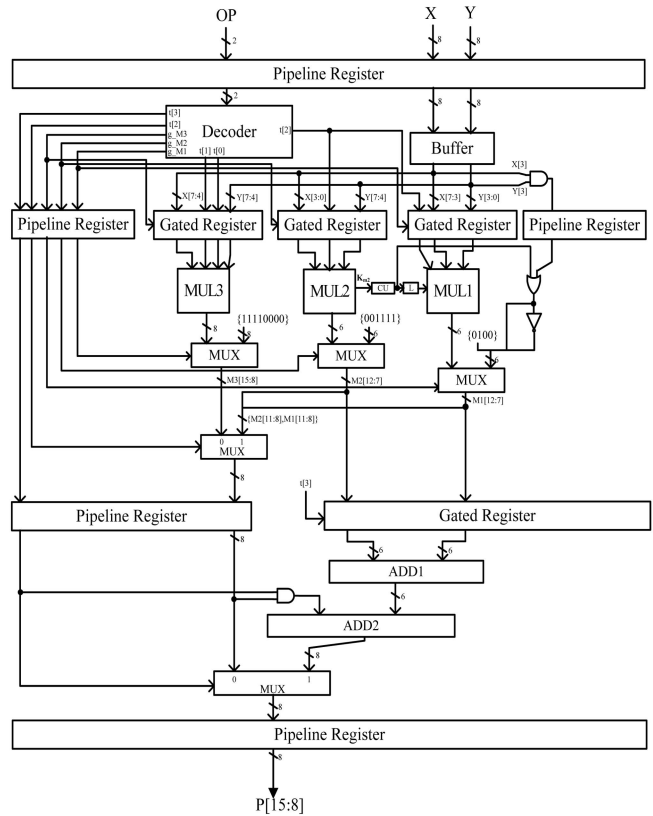


Fig. 11. Proposed power-efficient pipelined reconfigurable fixed-width multiplier.

Since we would like to disable MUL1, the inputs x_3y_3 and K_{m2} of MUL1 must be latched, and thus, the output signal of SCC1 will be unchanged. From the above four cases, the actual product of the disabled MUL1 is $(0100, x_3y_3 \vee K_{m2}, x_3y_3 \vee K_{m2})_2$ via logic operation of x_3y_3 and K_{m2} as shown in Fig. 11. On the other hand, the control unit (CU) in Fig. 11 is used to treat $K_{m2} = 1$ when MUL2 is disabled. The block denoted by L is a latch to keep present value when MUL1 is disabled. According to the above analysis, the signals g_M1, g_M2, g_M3 , and $t[3]$ are generated to control four gated registers and the former three signals are used to control three multiplexers of the actual product selection as shown in Fig. 11 such that low power consumption is achieved.

4.2 Zero Input for the Third Stage

Zero input scheme working for CM2, CM3, and CM4 is mainly aimed at providing zero input sequences for adder to keep value unchanged at the third stage of Fig. 11. If CM2, CM3, or CM4 is performed, we use AND gates to generate zero sequence and feed into the ADD2. In this case, for ADD1, we can use $t[3]$ as the control signal of the clock gating register to latch its input value. At the same time, for ADD2, one of the inputs comes from ADD1 that has been latched and we only need to set the other input to zero via AND operation with $t[3]$. Thus, we can further reduce the transition activity while the same CM is successively performed. On average, the gated clock and zero input schemes reduce around 98 and 2 percent of the total power reduction, respectively, since the latter scheme affects only ADD2 at the third stage.

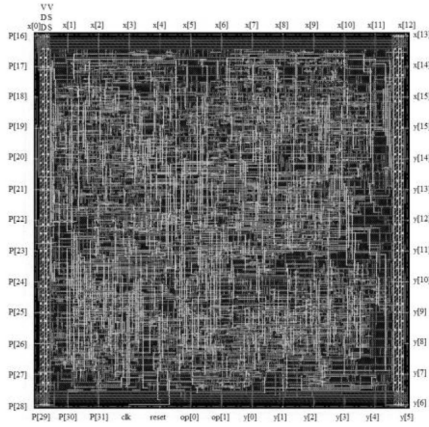


Fig. 12. Proposed power-efficient pipelined reconfigurable fixed-width multiplier layout for $n = 16$.

5 COMPARISON AND CHIP IMPLEMENTATION

In this section, we present the main differences among the various reconfigurable multipliers in qualitative way and show the power and area comparison results among power-efficient reconfigurable, nonpower-efficient reconfigurable, and non-reconfigurable pipelined fixed-width multipliers in quantitative behavior. The qualitative comparison results between the proposed reconfigurable multiplier and other existing reconfigurable multipliers are listed in Table 4. From Table 4, only the proposed reconfigurable multiplier uses the fixed-width multiplier infrastructure to generate fixed-width and full-precision multipliers. Thus, we can directly provide two useful precision multiplier outputs for DSP and computer applications. Other reconfigurable multipliers [17], [18], [19], [20], [21], [22], [23], [24], [25] apply the full-precision multiplier infrastructure to generate only full-precision multipliers. The proposed reconfigurable multiplier and other reconfigurable multipliers [17], [18], [19], [20], [21], [22], [23], [25] have compact design complexity in comparison with that of [24] because the multiplier in [24] needs to reconfigure more different function modes and pipeline stages. The number of operands of the proposed multiplier and published multipliers [17], [18], [19], [20],

[21], [22], [23], [24] is variable such that the designs can concurrently provide multiple lower resolution multiplications. Concerning the chip implementation, we adopt the cell-based design flow with Artisan standard cell library and implement the reconfigurable fixed-width multiplier in TSMC 0.18 μm CMOS process. Synopsys Design Compiler is employed to synthesize the RTL design of the proposed reconfigurable multiplier and Cadence SOC Encounter is adopted for placement and routing (P&R). The active chip layout area of the proposed power-efficient pipelined reconfigurable 16×16 fixed-width multiplier, as shown in Fig. 12, is $197.005 \mu\text{m} \times 196.56 \mu\text{m}$. Although we have mentioned the main differences in qualitative way as listed in Table 4, it is difficult to compare the performance with other previous reconfigurable multipliers [17], [18], [19], [20], [21], [22], [23], [24], [25] in quantitative way due to different CMs/functions, different numbers of CMs, different prototype multiplier infrastructures, and different targets. In order to show the power consumption and chip area comparison results in quantitative way, we reproduce the pipelined reconfigurable fixed-width multiplier without using low-power schemes (i.e., nonpower-efficient pipelined reconfigurable multiplier) and non-reconfigurable pipelined fixed-width multiplier for $n = 8, 16, 24$, and 32 . Note that the non-reconfigurable pipelined fixed-width multiplier uses four pipelined-register bands as shown in Fig. 10a to pipeline the fixed-width multiplier of [16]. Table 5 reveals the power consumption and chip area comparison among the non-reconfigurable pipelined fixed-width multiplier, power-efficient, nonpower-efficient pipelined reconfigurable fixed-width multipliers in different CMs. We measure the power consumption using 100,000 random input vectors via Synopsys PrimePower at 100 MHz with 1.8 V after RC extraction of the placed and routed netlists. From Table 5, in comparison with the power dissipation of the non-reconfigurable multiplier, the proposed one can achieve power reduction of 0.81, 12.46, 17.93, and 23.2 percent, on average, for $n = 8, 16, 24$, and 32 , respectively. In the same table, the proposed power-efficient pipelined reconfigurable fixed-width multiplier compared with the non-power-efficient one can save 10.59, 21.7, 28.84, and 31.58 percent power consumption, on average, respectively, for $n = 8, 16, 24$, and 32 . We can see that, for $n = 32$, the

TABLE 4
Qualitative Comparison among Different Reconfigurable Multipliers

	[17-22]	[23]	[24]	[25]	This Work
Multiplication Infrastructure	Full-precision	Full-precision	Full-precision	Full-precision	Fixed-width
Precision Provided	Full-precision	Full-precision	Full-precision	Full-precision	Fixed-width, Full-precision
Complexity	Compact	Compact	Large	Compact	Compact
# Pipeline Stages	Non-pipelined	Fixed	Variable	Non-pipelined	Fixed
Function	Multiplication	Inner product	MAC, multiplication, addition, data format conversion	Multiplication	Multiplication
# Operands	Variable	Variable	Variable	Fixed	Variable

TABLE 5

Power Consumption and Chip Area Comparison among Non-Reconfigurable Pipelined Fixed-Width Multiplier, Power-Efficient, and Nonpower-Efficient Pipelined Reconfigurable Fixed-Width Multipliers for $n = 8, 16, 24,$ and 32

n	Power Consumption and Chip Area	Non-Reconfigurable Pipelined Fixed-Width Multiplier	Non-Power-Efficient Reconfigurable Pipelined Fixed-Width Multiplier	Power-Efficient Reconfigurable Pipelined Fixed-Width Multiplier
8	CM1 Power Consumption	1.609 mW	1.845 mW	2.145 mW
	CM2 Power Consumption	N/A	1.822 mW	1.607 mW
	CM3 Power Consumption	N/A	1.815 mW	1.389 mW
	CM4 Power Consumption	N/A	1.657 mW	1.242 mW
	Average Power Consumption	1.609 mW (100%)	1.785 mW (110.94%)	1.596 mW (99.19%)
	Active Chip Area	99.565 $\mu\text{m} \times 90.72 \mu\text{m}$ (100%)	106.71 $\mu\text{m} \times 105.84 \mu\text{m}$ (125.04%)	115.93 $\mu\text{m} \times 115.92 \mu\text{m}$ (148.78%)
16	CM1 Power Consumption	4.777 mW	5.474 mW	6.237 mW
	CM2 Power Consumption	N/A	5.659 mW	4.358 mW
	CM3 Power Consumption	N/A	5.432 mW	3.414 mW
	CM4 Power Consumption	N/A	4.799 mW	2.719 mW
	Average Power Consumption	4.777 mW (100%)	5.341 mW (111.81%)	4.182 mW (87.54%)
	Active Chip Area	165.665 $\mu\text{m} \times 156.24 \mu\text{m}$ (100%)	187.57 $\mu\text{m} \times 186.48 \mu\text{m}$ (135.14%)	197.005 $\mu\text{m} \times 196.56 \mu\text{m}$ (149.61%)
24	CM1 Power Consumption	10.56 mW	12.52 mW	13.23 mW
	CM2 Power Consumption	N/A	12.89 mW	8.846 mW
	CM3 Power Consumption	N/A	12.45 mW	7.042 mW
	CM4 Power Consumption	N/A	10.85 mW	5.551 mW
	Average Power Consumption	10.56 mW (100%)	12.18 mW (115.34%)	8.667 mW (82.07%)
	Active Chip Area	245.965 $\mu\text{m} \times 241.92 \mu\text{m}$ (100%)	285.715 $\mu\text{m} \times 277.2 \mu\text{m}$ (133.10%)	296.585 $\mu\text{m} \times 287.28 \mu\text{m}$ (143.19%)
32	CM1 Power Consumption	20.17 mW	23.37 mW	23.75 mW
	CM2 Power Consumption	N/A	23.88 mW	15.14 mW
	CM3 Power Consumption	N/A	23.30 mW	13.17 mW
	CM4 Power Consumption	N/A	20.00 mW	9.894 mW
	Average Power Consumption	20.17 mW (100%)	22.64 mW (112.25%)	15.49 mW (76.80%)
	Active Chip Area	333.345 $\mu\text{m} \times 327.6 \mu\text{m}$ (100%)	382.285 $\mu\text{m} \times 378.0 \mu\text{m}$ (132.32%)	389.115 $\mu\text{m} \times 383.04 \mu\text{m}$ (136.48%)

average power consumption of the proposed power-efficient reconfigurable multiplier leads to 23.2 and 31.58 percent power saving in comparison with that of the non-reconfigurable multiplier and nonpower-efficient multiplier, respectively. In the same case, although the proposed power-efficient reconfigurable multiplier has 36.48 and 3.14 percent more area than that of the non-reconfigurable and nonpower-efficient reconfigurable structures, respectively, the proposed architecture can certainly attain the largest power saving among three designs. It is emphasized that the non-reconfigurable multiplier cannot provide more than one configuration mode compared with the reconfigurable multiplier design. In CM2, CM3, and CM4, the presented

power-efficient reconfigurable multiplier outperforms the nonpower-efficient one in terms of power saving. The power consumption of CM1 of the power-efficient reconfigurable multiplier closely approaches that of the nonpower-efficient one while the length of n increases.

6 CONCLUSIONS

This paper presents a framework for the pipelined reconfigurable fixed-width Baugh-Wooley multiplier to generate a family of fixed-width and full-precision multipliers including CM1, CM2, CM3, and CM4. We make use of low-power schemes including gated clock and zero input techniques to

achieve power reduction of 0.81, 12.46, 17.93, and 23.2 percent, on average, compared with the non-reconfigurable multiplier for $n = 8, 16, 24,$ and $32,$ respectively. On the other hand, compared with the nonpower-efficient reconfigurable multiplier, we can save 10.59, 21.7, 28.84, and 31.58 percent power consumption, respectively, for $n = 8, 16, 24,$ and $32.$ The future work may cover as follows: One is to apply this reconfigurable design methodology to other arithmetic number systems and the other is to use this design in power-aware computer and DSP applications.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their constructive comments and suggestions. This work was supported in part by the National Science Council (NSC) Grants NSC-96-2220-E-009-038, NSC-96-2221-E-009-220, NSC-97-2220-E-009-055, and MOEA-96-EC-17-A-01-S1-048.

REFERENCES

- [1] C.R. Baugh and B.A. Wooley, "A Two's Complement Parallel Array Multiplication Algorithm," *IEEE Trans. Computers*, vol. 22, no. 12, pp. 1045-1047, Dec. 1973.
- [2] A.D. Booth, "A Signed Binary Multiplication Techniques," *Quarterly J. Mechanics and Applied Math.*, vol. 4, pp. 236-240, 1951.
- [3] O.L. MacSorley, "High-Speed Arithmetic in Binary Computer," *Proc. Conf. Institute of Radio Engineers (IRE '61)*, vol. 49, pp. 67-91, 1961.
- [4] K. Hwang, *Computer Arithmetic: Principles, Architecture, and Design*. John-Wiley, 1979.
- [5] F. Cavanagh, *Digital Computer Arithmetic: Design and Implementation*. McGraw-Hill, 1984.
- [6] M.D. Ercegovac and T. Lang, *Digital Arithmetic*. Morgan and Kaufmann, 2004.
- [7] S.L. Freeny, "Special-Purpose Hardware for Digital Filtering," *Proc. IEEE*, vol. 63, no. 4, pp. 633-647, Apr. 1975.
- [8] Y.C. Lim, "Single-Precision Multiplier with Reduced Circuit Complexity for Signal Processing Applications," *IEEE Trans. Computers*, vol. 41, no. 10, pp. 1333-1336, Oct. 1992.
- [9] M.J. Schulte and E.E. Swartzlander Jr., "Truncated Multiplication with Correction Constant," *Proc. Workshop Very Large Scale Integration (VLSI) Systems Signal Processing, VI*, pp. 388-396, 1993.
- [10] S.S. Kidambi, F. El-Guibaly, and A. Antoniou, "Area-Efficient Multipliers for Digital Signal Processing Applications," *IEEE Trans. Circuits and Systems*, vol. 43, no. 2, pp. 90-94, Feb. 1996.
- [11] E.J. King and E.E. Swartzlander Jr., "Data-Dependent Truncation Scheme for Parallel Multipliers," *Proc. 31st Asilomar Conf. Signals, Systems, and Computers*, vol. 2, pp. 1178-1182, 1997.
- [12] E.E. Swartzlander Jr., "Truncated Multiplication with Approximate Rounding," *Proc. 33rd Asilomar Conf. Signals, Systems, and Computers*, vol. 2, pp. 1480-1483, 1999.
- [13] J.M. Jou, S.R. Kuang, and R.D. Chen, "Design of Low-Error Fixed-Width Multiplier for DSP Applications," *IEEE Trans. Circuits and Systems*, vol. 46, no. 6, pp. 836-842, June 1999.
- [14] L.D. Van, S.S. Wang, and W.S. Feng, "Design of the Lower-Error Fixed-Width Multiplier and Its Application," *IEEE Trans. Circuits and Systems*, vol. 47, no. 10, pp. 1112-1118, Oct. 2000.
- [15] K.J. Cho, K.C. Lee, J.G. Chung, and K.K. Parhi, "Design Low-Error Fixed-Width Modified Booth Multiplier," *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, vol. 12, no. 5, pp. 522-531, May 2004.
- [16] L.D. Van and C.C. Yang, "Generalized Low-Error Area-Efficient Fixed-Width Multipliers," *IEEE Trans. Circuits and Systems I*, vol. 52, no. 8, pp. 1608-1619, Aug. 2005.
- [17] S. Krithivasan and M.J. Schulte, "Multiplier Architectures for Media Processing," *Proc. IEEE Asilomar Conf. Signals, Systems, and Computers*, vol. 2, pp. 2193-2197, Nov. 2003.
- [18] Y.-H. Huang, H.-P. Ma, M.-L. Liou, and T.-D. Chiueh, "A 1.1 G MAC/s Subword-Parallel Digital Signal Processor for Wireless Communication Applications," *IEEE J. Solid-State Circuits*, vol. 39, no. 1, pp. 169-183, Jan. 2004.
- [19] S. Krithivasan, M.J. Schulte, and J. Glossner, "A Subword-Parallel Multiplication and Sum-of-Squares Unit," *Proc. IEEE CS Ann. Symp. Very Large Scale Integration (VLSI) Systems*, pp. 273-274, Feb. 2004.
- [20] Y.-L. Tsao, W.-H. Chen, M.-H. Tan, M.-C. Lin, and S.-J. Jou, "Low-Power Embedded DSP Core for Communication Systems," *EURASIP J. Applied Signal Processing*, pp. 1355-1370, Jan. 2003.
- [21] D. Tan, A. Danysh, and M. Liebelt, "Multiple-Precision Fixed-Point Vector Multiply-Accumulator Using Shared Segmentation," *Proc. IEEE Symp. Computer Arithmetic*, pp. 12-19, June 2003.
- [22] C.L. Wey and J.F. Li, "Design of Reconfigurable Array Multipliers and Multiplier-Accumulators," *Proc. IEEE Asia-Pacific Conf. Circuits and Systems*, pp. 37-40, Dec. 2004.
- [23] R. Lin, "Reconfigurable Parallel Inner Product Processor Architecture," *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, vol. 9, no. 2, pp. 261-272, Apr. 2001.
- [24] K. Tatas, G. Koutroumpetis, D. Soudris, and A. Thanailakis, "Architecture Design of a Coarse-Grain Reconfigurable Multiply-Accumulate Unit for Data-Intensive Applications," *Integration, the VLSI J.*, vol. 40, pp. 74-93, Feb. 2007.
- [25] S.D. Haynes and P.Y.K. Cheung, "Configurable Multiplier Blocks for Embedding in FPGAs," *Electronics Letter*, vol. 34, no. 7, pp. 638-639, Apr. 1998.
- [26] J. Di and J.S. Yuan, "Run-Time Reconfigurable Power-Aware Pipelined Signed Array Multiplier Design," *Proc. IEEE Int'l Symp. Signals, Circuits, and Systems*, vol. 2, pp. 405-406, July 2003.
- [27] M. Sjalander, M. Drazdziulis, P. Larsson-Edefors, and H. Eriksson, "A Low-Leakage Twin-Precision Multiplier Using Reconfigurable Power Gating," *Proc. IEEE Int'l Symp. Circuits, and Systems*, vol. 2, pp. 1654-1657, May 2005.
- [28] S.-R. Kuang and J.-P. Wang, "Design of Power-Efficient Pipelined Truncated Multipliers with Various Output Precision," *IET Computers & Digital Techniques*, vol. 1, pp. 129-136, Mar. 2007.



Jin-Hao Tu received the BS degree from the National Changhua University of Education, Taiwan, in 2006, and the MS degree from the National Chiao Tung University (NCTU), Hsinchu, Taiwan, in 2008. His research interests are computer arithmetic and 3D graphics system design. In 2007, he was the corecipient of the third place of ARM Code-O-Rama Design Contest.



Lan-Da Van received the BS (honors) and MS degrees from Tatung Institute of Technology, Taipei, Taiwan, in 1995 and 1997, respectively, and the PhD degree from the National Taiwan University (NTU), Taipei, in 2001, all in electrical engineering. From 2001 to 2006, he was an associate researcher at the National Chip Implementation Center (CIC), Hsinchu, Taiwan. In February 2006, he joined the Faculty of Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan, where he is currently an assistant professor. His research interests are in VLSI algorithms, architectures, and chips for digital/biomedical signal processing, 3D graphics, and baseband communication systems. This includes the design of high-performance/power-aware/cost-effective graphics/DSP processors, adaptive filters, transform, computer arithmetic, and platform-based system-on-a-chip (SOC) designs. He has published 40 journal and conference papers in these areas. He was a recipient of the Chungghwa Picture Tube (CPT) and Motorola Fellowships in 1996 and 1997, respectively. He was an elected chairman of the IEEE NTU Student Branch in 2000. In 2002, he received the IEEE award for outstanding leadership and service to the IEEE NTU Student Branch. In 2005, he was a recipient of the Best Poster Award at iNEER Conference for Engineering Education and Research (iCEER). From 2009, he serves as the officer of IEEE Taipei Section. He has served as a reviewer for the *IEEE TCAS I*, the *IEEE TCAS II*, the *IEEE TCSVT*, the *IEEE TC*, the *IEEE TVLSI Systems*, the *IEEE TSP*, the *IEEE TMM*, and the *IEEE SPL*. He is a member of the IEEE.