Article

An Accuracy-Improved Fixed-Width Booth Multiplier Enabling Bit-Width Adaptive Truncation Error Compensation

Song-Nien Tang *, Jen-Chien Liao, Chen-Kai Chiu, Pei-Tong Ku and Yen-Shuo Chen

Information and Computer Engineering Department, Chung Yuan Christian University, Taoyuan 32023, Taiwan; liaojack1233@gmail.com (J.-C.L.); nike7014@gmail.com (C.-K.C.); jin0917362909@gmail.com (P.-T.K.); daniel0319@livemail.tw (Y.-S.C.)

* Correspondence: sntang@ice.cycu.edu.tw; Tel.: +886-3-26654731

Abstract: Fixed-width Booth multipliers (FWBMs) generate a product with the same bit width as the operand and have been extensively employed in many digital systems. Various truncation error compensation (TEC) schemes have been presented for FWBM designs, aiming to reduce hardware costs while preserving operation accuracy. In general, the existing TEC methods function adequately for an exact bit width of the operand but fail to consider the TEC effect for FWBM inputs with various bit-width levels. To address this issue, we propose a bit-width adaptive TEC (BWATEC) scheme for providing high-accuracy TEC functions that are adaptive to the multiple L' -bit numerical ranges of input data for an L -bit FWBM ($L' \leq L$). We also present adjustable architecture for a 16-bit FWBM to enable the proposed BWATEC scheme and evaluate the hardware performance, using the TSMC 40 nm standard cell library. Relative to the contrast 16-bit FWBM approaches that use state-of-the-art TEC methods, the proposed BWATEC-enabled FWBM design can achieve reductions in the area-delay-error product of 7.9–50.9%, 17.1–69.5%, 29.9–82.2%, and 100% for the 14-bit, 12-bit, 10-bit, and 8-bit inputs, respectively. Moreover, the resultant 16-bit FWBM with BWATEC was verified by using the field-programmable gate array for convolutional neural network acceleration.

Keywords: Booth multiplier; fixed-width; truncation error compensation; bit-width adaptive

Citation: Tang, S.-N.; Liao, J.-C.; Chiu, C.-K.; Ku, P.-T.; Chen, Y.-S. An Accuracy-Improved Fixed-Width Booth Multiplier Enabling Bit-Width Adaptive Truncation Error Compensation. *Electronics* **2021**, *10*, 2511. <https://doi.org/10.3390/electronics10202511>

Academic Editor: Spiros Nikolaidis

Received: 26 August 2021

Accepted: 13 October 2021

Published: 15 October 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Multipliers are widely used in many digital operation systems. To limit bit-width increases in data paths, fixed-width multipliers are accordingly employed as arithmetic modules for digital signal processing, communication baseband operations, and neural network acceleration [1–4]. L -bit fixed-width multipliers generate the same L -bit output width as the L -bit operand, of which the Baugh–Wooley (array) multiplier and Booth multiplier are two of the most popular types. Two convenient approaches to fixed-width Baugh–Wooley or Booth multipliers are post-truncation (PT) and direct-truncation (DT). The PT method calculates all partial products and rounds the $2L$ -bit full-width product to the L most significant bits (MSBs) to achieve high accuracy, but the hardware costs are high. The DT method truncates the partial products related to the least significant bits (LSBs) of the $2L$ -bit full-width product to reduce the hardware costs, but the accuracy is very low.

Considering both operation accuracy and hardware complexity, several schemes based on truncation error compensation (TEC) have been presented for fixed-width Baugh–Wooley multipliers [5–8] or fixed-width Booth multipliers (FWBMs) [9–22]. The Booth multiplier has benefits in achieving high hardware efficiency because the number of rows of partial products is significantly reduced [23,24]. Moreover, the lower level of truncated partial products for Booth multipliers profits the fixed-width operation accuracy [17,18]. Therefore, the FWBM that enables a kind of TEC scheme is discussed in this study. A number of TEC schemes for FWBMs have been presented [9–22]. In general, TEC schemes for FWBMs obtain a TEC value (bias) based on computer simulation [9–14] or probability-based estimation [14–22] to compensate for the truncation error associated with the curtailed

partial products. For simulation-based methods, the work in Reference [9] used linear regression analysis and simulation to generate bias values. In Reference [10], bias terms were generated based on simulation outcomes, and simplified through the Karnaugh map processing. Moreover, the authors of References [11,12] derived formulas in a closed form for TEC biasing based on simulation results. In References [12–14], the bias terms through a simulation were determined utilizing the Booth encoded results to further improve accuracy. The computer simulation methods presented in References [9–14] are applicable but generally consume exhaustive simulation time to obtain the TEC bias value. Instead of exhaustive simulation, the authors of References [14–22] also presented the probability-based scheme to derive the TEC function. In this study, we aim to design the TEC-enabled FWBM based on the probabilistic estimation method. More literature reviews for the state-of-the-art FWBM design that uses the probability-based TEC scheme [14–22] are discussed in Section 2.2.

In conventional TEC methods for FWBMs, TEC functions are generally operated based on a certain particular bit width of the FWBM operand. However, in practical applications, an L -bit FWBM might need to process input patterns with various L' -bit widths ($L' \leq L$; L and L' are generally even). For example, a 16-bit FWBM might be employed to operate with 16-bit, 14-bit, 12-bit, 10-bit, or 8-bit numerical input patterns, as specified by different situations. Such an operation can be practically performed in several applications. Taking the convolutional neural network (CNN) as an example, an accelerator may employ an FWBM to process input data that have different settable bit widths from different CNN models or layers. Moreover, FWBMs used in a shared digital filter might operate with input data whose levels are various for multiple analog modules. To the best of our knowledge, no previous study has developed a TEC scheme for such an FWBM design to offer adaptive TEC biasing for various bit widths of input data. In this study, we propose a bit-width adaptive TEC (BWATEC) scheme for providing an adjustable TEC bias for the diverse bit widths of input patterns. For an L -bit FWBM, the proposed BWATEC method can enable a tailored and high-accuracy TEC function for each case of the L' -bit input pattern (where $L' \leq L$). In addition, an FWBM design for enabling the BWATEC is proposed based on a reconfigurable bias circuit with high hardware efficiency.

The remainder of this paper is organized as follows. Section 2 briefly introduces the background of FWBMs and the conventional probability-based TEC schemes for FWBMs. Section 3 outlines the proposed BWATEC scheme and its operations. In Section 4, the architecture of a 16-bit FWBM enabling the proposed BWATEC scheme is described. Section 5 evaluates the accuracy and hardware performances of our design and reports the experimental results, using a system-on-chip (SoC) field-programmable gate array (FPGA) platform. Finally, the conclusions are highlighted in Section 6.

2. Preliminaries and Design Issues

Some abbreviations and acronym words frequently used in this study are tabulated in Table 1 for convenient reference.

Table 1. Mapping of abbreviations and acronym words.

FWBM	Fixed-Width Booth Multiplier
TEC	Truncation error compensation
BWATEC	Bit-width adaptive truncation error compensation
P.P.	Partial products
MP/TP	Main part/truncation part
MSC	Most significant column
CNN	Convolutional neural network
ECG	Electrocardiogram

Table 1. Cont.

FWBM	Fixed-Width Booth Multiplier
HW/SW	Hardware/software
SoC-FPGA	System-on-chip field-programmable gate array
ZP	Zero-Padding
R _Z /R _H /R _D	Zero region/hybrid region/deterministic-only region
ADEP	Area-delay-error product

2.1. Fixed-Width Booth Multiplier (FWBM)

Let A and B be two L -bit 2's complement operands, represented by " $a_{L-1}, a_L, \dots, a_1, a_0$ " and " $b_{L-1}, b_L, \dots, b_1, b_0$ " with the values shown below, respectively.

$$A = -a_{L-1} \cdot 2^{L-1} + \sum_{i=0}^{L-2} a_i \cdot 2^i \quad B = -b_{L-1} \cdot 2^{L-1} + \sum_{i=0}^{L-2} b_i \cdot 2^i \quad (1)$$

The Booth encoding maps three consecutive terms, b_{2j+1} , b_{2j} , and b_{2j-1} into d_j , as tabulated in Table 2. The d_j value can be associated with $(b_{2j+1}, b_{2j}, b_{2j-1})$ terms as expressed in Equation (2), where $Q = (1/2) \times L$. As a result, a $2L$ -bit full-width product (FP) for $A \times B$ can be obtained as shown in Equation (3).

$$B = \sum_{j=0}^{Q-1} d_j \cdot 2^{2j}, \quad d_j = -2b_{2j+1} + b_{2j} + b_{2j-1} \quad (2)$$

$$FP = \left(\sum_{j=0}^{Q-1} d_j \cdot 2^{2j} \right) \times \left(-a_{L-1} \cdot 2^{L-1} + \sum_{i=0}^{L-2} a_i \cdot 2^i \right) \quad (3)$$

Table 2. Mapping results for the Booth encoder and partial products.

$(b_{2j+1} \ b_{2j} \ b_{2j-1})$	d_j	$p_{L,j} \ p_{L-1,j} \ p_{L-2,j} \ \cdots \ p_{2,j} \ p_{1,j} \ p_{0,j}$	n_j
(0 0 0)/(1 1 1)	0	0 0 0 $\cdots$ 0 0 0	0
(0 0 1)/(0 1 0)	1	a_{L-1} a_{L-1} a_{L-2} $\cdots$ a_2 a_1 a_0	0
(1 0 1)/(1 1 0)	-1	$\overline{a_{L-1}}$ $\overline{a_{L-1}}$ $\overline{a_{L-2}}$ $\cdots$ $\overline{a_2}$ $\overline{a_1}$ $\overline{a_0}$	1
(0 1 1)	2	a_{L-1} a_{L-2} a_{L-3} $\cdots$ a_1 a_0 0	0
(1 0 0)	-2	$\overline{a_{L-1}}$ $\overline{a_{L-2}}$ $\overline{a_{L-3}}$ $\cdots$ $\overline{a_1}$ $\overline{a_0}$ 1	1

Using binary arithmetic for $A \times B$, the partial products (P.P.) for each d_j can be derived in terms of a_i (i is from 0 to $L - 1$), 0, or 1, as shown for the values of $p_{i,j}$, and n_j in Table 2. Based on the P.P. terms in Table 2, Figure 1 depicts the structure of the P.P. array for an example of a 16-bit ($L = 16$) $A \times B$ full-width Booth multiplier. As shown in Figure 1, all P.P. terms can be divided into two groups: the main part (MP) and truncation part (TP). The P.P. in the MP are calculated to generate the product, whereas the TP includes the P.P. for computing the rounded L LSBs of the full-width product. The TP can be further divided into the TP_{major} and TP_{minor} subgroups. As indicated in Figure 1, TP_{major} contains the P.P. in the most significant column (MSC) of the TP, which dominates the accuracy of the carry from the TP toward the MP. In general, the accuracy can be improved by increasing the column range for TP_{major} [17–20]. However, a TEC function based on the MSC TP_{major} with one MSC usually offers adequate accuracy in many applications and the use of one-MSC TP_{major} sufficiently serves as a baseline to evaluate the performances for different TEC schemes [12,14,16,22]. Thus, this study adopted the one-MSC TP_{major} to develop and evaluate our BWATEC scheme and FWBM design. In an FWBM design with

TEC, TP_{major} is reserved for calculation, whereas TP_{minor} is truncated, and an estimated bias is adopted to compensate for the truncation error [9–22]. Therefore, an L -bit FWBM with TEC produces an L -bit quantized FP_q result, as expressed in Equation (4), where B_{TEC} indicates the estimated bias value for TEC, TP_{major} is mapped to the 2^{-1} digit and $R\{\cdot\}$ is the rounding operation.

$$FP_q = MP + \sigma \cdot 2^L, \sigma = R\{TP_{major} + B_{TEC}\} \quad (4)$$

With regard to an L -bit FWBM whose operands can be assigned to the input data of multiple prespecified L' -bit width ($L' \leq L$), the L' -bit input patterns are necessarily left-shifted by $(L-L')$ bits and are padded with zeros (i.e., Zero-Padding bits) to form the L -bit operand. The aforementioned processing for $L = 16$ is also described in Figure 1 for input patterns with multiple 14-bit, 12-bit, 10-bit, or 8-bit (i.e., L' -bit) widths.

Figure 1. Partial product ($P.P.$) array structure for a 16-bit full-width Booth multiplier with multiple 16-to-8-bit numerical ranges of input patterns.

2.2. Probability-Based TEC Schemes for FWBMs

Several FWBM designs with probability-based TEC have been presented [14–22]. The authors of Reference [14] presented the probability-based scheme, together with their simulation-based works. Similarly, the work in Reference [15] used the expected value for $P.P.$ to derive bias values. Furthermore, the probabilistic analysis methods [16–22] derived closed formulas of the TEC function based on the expected value or the conditional probability for $P.P.$ terms. In Reference [16], the expected values for two groups of TP_{minor} (i.e., the n_j terms in Table 2 equals to 0 or 1) were respectively derived to obtain the probabilistic estimation bias when one-MSB TP_{major} is specified. In addition, a generalized probabilistic estimation bias (GPEB) method [17] further enhanced the work in Reference [16] for the cases of TP_{major} containing more $P.P.$ columns. Using the GPEB methods [16,17], a simple TEC function of a 1-bit or 2-bit constant value was derived. The work in Reference [18] presented a TEC scheme based on the conditional probability depending on non-zero Booth encoder outputs (i.e., $d_j^1 = 0$ in Table 2) for each row of TP_{minor} . In Reference [19], a more complex method based on [18] was presented by using a conditional probability model for multiple TP_{minor} rows. Such a design [19] slightly improved accuracy but increased hardware overheads. The authors of Reference [20] considered both expected values and conditional probability to progress a bias function improving accuracy and area based on the probability and computer simulation (PACS). In Reference [21], the concept of data scaling was presented and applied to conventional TEC-adapted FWBM designs for improving accuracy. A Booth-encoded sign-digit-based conditional probability (BSCP) method was presented in Reference [22] for the case of one-MSB TP_{major} . The work in Reference [22] further took advantage of the sign of non-zero Booth encoder results to generate a TEC function achieving relatively high accuracy.

Figure 3. Schematic for the TP_{minor} contents of a 16-bit Booth multiplier for other L' -bit inputs: (a) $L' = 16$ and (b) $L' = 8$.

By mapping TP_{major} to the 2^{-1} digit (i.e., the MSB of TP_{minor} is 2^{-2}), the expected value of all $P.P.$ for the j th-row TP_{minor} in the R_H , $E[TP_{minor,j}^{(H)}]$, can be calculated as Equation (5), where ns is the number of s_j (refer to Figure 2). Based on Equation (5) and the mapping contents in Table 2, the values of $E[TP_{minor,j}^{(H)}]$ and (n_j, s_j, e_j) according to d_j are listed in Table 3.

$$2^{2j-16} \cdot n_j + \sum_{\alpha=0}^{ns-1} 2^{2j+\alpha-16} \cdot s_j + 2^{2j+ns-16} \cdot e_j + \sum_{\beta=ns+1}^{14-2j} 2^{2j+\beta-16} \cdot 1/2 \quad (5)$$

Table 3. Values of (n_j, s_j, e_j) and according to d_j .

	<i>P.P. Values</i>	$E[TP_{minor,j}^{(H)}]$ Values
$d_j = 0$	(all $P.P. = 0$)	zero
$d_j = 1$	$(n_j = 0; s_j = 0; e_j = 1/2)$	$2^{-2} - 2^{-16+2j+ns-1}$
$d_j = -1$	$(n_j = 1; s_j = 1; e_j = 1/2)$	$2^{-2} + 2^{-16+2j+ns-1}$
$d_j = 2$	$(n_j = 0; s_j = 0; e_j = 0)$	$2^{-2} - 2^{-16+2j+ns}$
$d_j = -2$	$(n_j = 1; s_j = 1; e_j = 1)$	$2^{-2} + 2^{-16+2j+ns}$

The $E[TP_{minor,j}^{(H)}]$ values in Table 3 can be summarized by using the following expression, where a variable δ_j is defined by $\delta_j = 1$ for $d_j \neq 0$; otherwise, $\delta_j = 0$.

$$E[TP_{minor,j}^{(H)}] = 2^{-2} \cdot \delta_j - d_j \cdot 2^{-16+2j+ns-1} \quad (6)$$

From observing the R_H region for $L' = 14, 12$, and 10 in Figure 2, it can be found that the R_H includes the m th to the k th row of TP_{minor} , in which $m = ns/2$ and $k = 7 - (ns/2)$. By summing $E[TP_{minor,j}^{(H)}]$ for all rows in the R_H , an overall $E[TP_{minor}^{(H)}]$ can be obtained as follows:

$$E[TP_{minor}^{(H)}] = \sum_{j=m}^k E[TP_{minor,j}^{(H)}] = \sum_{j=m}^k 2^{-2} \cdot \delta_j - \sum_{j=m}^k d_j \cdot 2^{-16+2j+ns-1} \quad (7)$$

(a)

For an FWBM with TEC, the result of Equation (7) can be viewed as an ideal bias for the truncated TP_{minor} in the R_H ; however, the calculation of (a) in Equation (7) is complex. The bottom row in the R_H (i.e., the k th row; $j = k$) dominates the final calculation result. Moreover, the result of (a) in Equation (7) can be rounded to the 2^{-2} digit to be arithmetically added to δ_j . Therefore, Equation (7) can be approximated by Equation (8) by

simplifying the (a) part to a $\sigma \cdot 2^{-2}$ term, where $R_{-2}\{.\}$ represents rounding a value to the 2^{-2} digit.

$$E[TP_{minor}^{(H)}] \cong \sum_{j=m}^k (2^{-2} \cdot \delta_j) - R_{-2}\{d_k \cdot 2^{-3}\} \cong \sum_{j=m}^k (2^{-2} \cdot \delta_j) + \sigma \cdot 2^{-2} \quad (8)$$

$\sigma = 1, d_k < 0; \sigma = -1, d_k > 0; \sigma = 0, d_k = 0$

However, the subtraction arithmetic for the 2^{-2} (i.e., $\sigma = -1$ in Equation (8)) is also an issue in a *P.P.* array. This issue can be resolved by taking advantage of the following operational features. When d_k is negative, both δ_k and σ are equal to 1; thus, a carry of “1” can be added to the 2^{-1} digit. If d_k is positive, δ_k is 1, whereas σ is -1 . Thus, δ_k can be eliminated at the 2^{-2} digit, owing to the offset by σ . As a result, Equation (8) can be further calculated by using Equation (9), where a variable, γ , is operated at the 2^{-1} digit only with an addition.

$$E[TP_{minor}^{(H)}] \cong \sum_{j=m}^{k-1} (2^{-2} \cdot \delta_j) + 2^{-1} \cdot \gamma ; \gamma = \begin{cases} 1, d_k < 0 \\ 0, d_k \geq 0 \end{cases} \quad (9)$$

As shown in Figure 2a–c, the R_D region only comprises *P.P.* in terms of s_j and n_j , as the number of *P.P.* within a row in the R_D is less than the number of ZP bits (Figure 1). Similar to s_j in the R_H , the s_j terms in the R_D are also *P.P.* obtained from the ZP bits of the *A* operand and are equal to the d_j -dependent deterministic “1” or “0” values. Setting s_j and n_j to “1” (for $d_j < 0$) or “0” (for $d_j \geq 0$), the actual value of all *P.P.* for the j th row of $TP_{minor}^{(D)}$ in the R_D , i.e., $E[TP_{minor,j}^{(D)}]$, can be obtained by the following derivation. An accumulated result from the j th row in the R_D is introduced to the 2^{-1} digit for negative d_j values.

$$E[TP_{minor,j}^{(D)}] = 2^{-16+2j} \cdot (n_j + s_j) + \dots + 2^{-2} \cdot s_j = \begin{cases} 0, d_j \geq 0 \\ 2^{-1}, d_j < 0 \end{cases} \quad (10)$$

For the design examples illustrated in Figure 2a–c, the R_D region includes rows with indexes from $k+1$ to $Q-1$, where Q equals 8 for the case of $L = 16$. The variable Q is defined as $Q = L/2$, which is the number of rows in a *P.P.* array. Thus, a global $E[TP_{minor}^{(D)}]$ can be derived, as shown in Equation (11), in which the variable λ_j is defined by $\lambda_j = 1$ for $d_j < 0$ and $\lambda_j = 0$ for $d_j \geq 0$, corresponding to the execution results of Equation (10) for each row.

$$E[TP_{minor}^{(D)}] = \sum_{j=k+1}^{Q-1} TP_{minor,j}^{(D)} = \sum_{j=k+1}^{Q-1} (2^{-1} \cdot \lambda_j) \quad (11)$$

3.2. BWATEC Synthesis and Operations

For FWBMs with TEC, $E[TP_{minor}^{(H)}]$ and $E[TP_{minor}^{(D)}]$ values obtained by using Equations (9) and (11) can be employed as the TEC bias to compensate for the truncated *P.P.* of TP_{minor} in the R_H and R_D , respectively. Moreover, the operations of Equations (9) and (11) are different from the input bit width (L'), as well as the contents and range of the R_H and R_D regions (Figure 2). In addition to the cases of $L' = 14, 12$, and 10 as shown in Figure 2a–c, the proposed schemes based on Equations (9) and (11) can also be applied to the conditions of $L' = 16$ and 8 as shown in Figure 3. For $L' = 16$, TP_{minor} only has the R_H region, and we can use Equation (9) to generate the TEC bias. Alternatively, when $L' = 8$, Equation (11) is used, as only the R_D region is calculated. In practice, the TEC function for $L' = 16$ can be further improved. From Figure 1, we can use deterministic $p_{0,7}$ and n_7 to operate with δ_j at the 2^{-2} digit; thus, a more precise carry can be added at the 2^{-1} digit, instead of adding γ in Equation (9). The efficient use of Equations (9) and (11), as associated with multiple combinations of deterministic and probabilistic data, achieves the aims of the proposed BWATEC scheme. Considering a 16-bit FWM, Figure 4 illustrates the TEC operations by using the proposed BWATEC scheme for various L' -bit inputs.

Figure 4. Schematic of truncation error compensation (TEC) operations that use the proposed bit-width adaptive TEC (BWATEC) scheme for various L' -bit input patterns of a 16-bit fixed-width Booth multiplier (FWBM).

3.3. Design Scalability

Taking the 16-bit FWBM example as a base, the deduced processing can also be applied to general L -bit FWBM designs (i.e., L is a scalable number other than 16). In general, an L -bit Booth multiplier is operated based on L of an even number. Considering the scalability of the proposed design, the aimed L -bit FWBMs can be categorized into two kinds of specifications. One is $L = 2n$, and n is an even integer; thus, the number of $P.P.$ rows (i.e., the Q value) is even based on $Q = L/2$. The other is $L = 2n$, and n is an odd integer; thus, the number of $P.P.$ rows is odd. Referring to the contents associated with Figures 2 and 3, the TP_{minor} $P.P.$ corresponding to the $R_Z/R_H/R_D$ regions are illustrated for the design case of an L -bit Booth multiplier ($L = 16$) with various L' -bit inputs ($L' = 16, 14, 12, 10$, and 8), which has even (i.e., 8) $P.P.$ rows. Moreover, the contents related to Figure 4 illustrate the proposed BWATEC operation for a 16-bit FWBM. As extension based on the illustration for the case of $L = 16$, Figure 5 depicts the $R_Z/R_H/R_D$ distribution of TP_{minor} rows of a general L -bit Booth multiplier (i.e., L is scalable) for different L' -bit inputs, and Figure 5a,b illustrates the specification of “ $L = 2n$ (n and Q are even; even rows)” and “ $L = 2n$ (n and Q are odd; odd rows)”, respectively. In Figure 5a,b, the value shown inside each $R_Z/R_H/R_D$ block represents the number of rows in that region and refers to the ceiling operator.

Figure 5. Schematic for the $R_Z/R_H/R_D$ distribution of TP_{minor} rows of a general L -bit Booth multiplier for various L' -bit inputs: (a) $L = 2n$, and n is even; (b) $L = 2n$, and n is odd.

As indicated in the previous section, the proposed BWATEC operations for the case of a 16-bit FWBM (i.e., Figure 4) can be synthesized based on the contents in Figures 2 and 3 in common with Equations (9) and (11). By analogy with the derivation for the contents in Figure 4, the proposed BWATEC operations for various L' -bit input patterns of a general L -bit FWBM can be similarly synthesized based on Figure 5, Equations (9) and (11), as described in Figure 6, where the two specifications of $L = 2n$ (n is even or odd) are also respectively illustrated. The contents in Figures 5 and 6 further address the $R_Z/R_H/R_D$ distribution and BWATEC operations of a general L -bit FWBM operated with L' -bit inputs in small L' values. For the cases of " $L' \leq L/2-2$ ($L = 2n$; n is even)" or " $L' \leq L/2-1$ ($L = 2n$; n is odd)", only the R_Z and R_D regions are included and the range of R_D is reduced with smaller L' values. Such conditions allow only the R_D $P.P.$ to be calculated to obtain the TEC bias, as expressed in Equation (12), which is an extended form based on Equation (11).

$$E[TP_{minor}^{(D)}] = \sum_{j=t}^{Q-1} (2^{-1} \cdot \lambda_j); t = \begin{cases} Q/2 + 1, & L' \leq L/2 - 2, Q \text{ is even} \\ \lceil Q/2 \rceil + 1, & L' \leq L/2 - 1, Q \text{ is odd} \end{cases} \quad (12)$$

Figure 6. Schematic of BWATEC operations for various L' -bit input patterns of a general L -bit FWBM: (a) $L = 2n$, and n is even; (b) $L = 2n$, and n is odd.

4. Proposed BWATEC-Enabled FWBM Architecture

There is also a need for an FWBM design with an efficient architecture for enabling the proposed BWATEC scheme. Figure 7 describes the hardware architecture of a 16-bit FWBM example enabling the BWATEC functions. As shown in Figure 7, the $P.P.$ values are first produced through the Booth Encoder, and the $P.P.$ Generator operates on two operands of A and B , which are already padded with ZP bits according to the prespecified bit width, L' of input patterns (Figure 1). Depending on L' , the BWATEC-associated δ_j , γ , and λ_j terms are also set and sent to the $P.P.$ array, along with the $P.P.$ terms. The carry-save adder/carry-propagation adder (CSA/CPA) unit performs the array operations for the MP , TP_{major} , and BWATEC biasing. Right shifting of bits can be optionally executed at the CPA output depending on the practical system design. As detailed in Figure 7, we used four groups (i.e., M_1 , M_2 , M_3 , and M_4) of multiplexers controlled by the setting of L' to enable data selection of the carry of δ_j accumulation, γ , λ_j , and "0" for the BWATEC operations described in Figure 4. A switch is also employed for selecting γ or an extra carry contributed by the addition of $p_{0,7}$ and n_7 for $L' = 16$. Based on the configuration shown in Figure 7, similar approaches can be used to deduce the FWBM design for other bit widths of the operand. As a result, the BWATEC-enabled FWBM can be realized by using the originally required $P.P.$ elements with additional multiplexers (incl. a switch) and control logics for adjustable TEC operations.

Figure 7. Hardware architecture of 16-bit FWBM that uses the proposed BWATEC scheme.

Considering a general L -bit FWBM (L is scalable) enabling the proposed BWATEC scheme, we see that its hardware configuration for TEC biasing can also be developed based on the BWATEC operation shown in Figure 6, as the approach for the 16-bit FWBM example (refer to Figures 4 and 7). The hardware structure for BWATEC biasing of a general L -bit FWBM is described in Figure 8a,b for the two specifications of “ $L = 2n$ (n is even)” and “ $L = 2n$ (n is odd)”, respectively. As indicated in Figure 8, the addition of biasing element is performed by using full adders (FAs) or half adders (HAs). The mandatory multiplexers (i.e., MUX1 in Figure 8) are employed to select the carry of δ accumulation or the γ and λ terms based on the BWATEC operations in Figure 6, and the optional multiplexers (i.e., MUX2) can allow unadded δ terms to be “0” for energy efficiency. If the devised FWBM is specified to process L' -bit inputs with small L' values, corresponding levels of multiplexers (i.e., MUX3) might be employed to mask the uncalculated γ or λ terms (refer to Figure 6) as shown in Figure 8. Based on the configuration of a $P.P.$ array and the BWATEC biasing (refer to Figure 8), the hardware (HW) resource usage in the number of FAs, HAs, and multipliers (i.e., MUX1, 2, and 3) of a general L -bit FWBM using the proposed BWATEC scheme for various L' -bit inputs are listed in Table 4.

Table 4. HW resources usage of a general L -bit FWBM for various L' -bit inputs using the BWATEC scheme ($Q = L/2$).

MP/TP _{major} HW Resources		BWATEC Biasing HW Resources			
#FA/HA		#FA/HA	#MUX1	#MUX2	#MUX3
$\frac{Q}{2} \times (4 + L) + (Q - 1)$	$L = 2n$ (even n)	$Q/2$	$Q/2$	$\lceil Q/2 - 1 \rceil$	$(1/2) \times (L/2 - L')$
	$L = 2n$ (odd n)	$\lceil Q/2 \rceil$	$\lceil Q/2 \rceil$	$\lceil Q/2 - 1 \rceil$	$\lceil (1/2) \times (L/2 - L') \rceil + 1$

Figure 8. Hardware configuration for BWATEC biasing of a general L -bit FWBM: (a) $L = 2n$, and n is even; (b) $L = 2n$, and n is odd.

5. Evaluations and Experiments

Considering 16-bit FWBM designs with TEC based on one-MSC TP_{major} , this section evaluates the accuracy and hardware performances for the proposed design and several representative works in previous studies. Moreover, the 16-bit FWBM with BWATEC was verified through the SoC-FPGA implementation for CNN inference operations.

5.1. Evaluations of Accuracy and Hardware Performances

For the accuracy performance, the signal-to-noise ratio (SNR) is the most important parameter and is defined as in Equation (13), where FP (refer to Equation (3)) is the product of the full-width Booth multiplier, and FP_q (refer to Equation (4)) is the product of the FWBM with TEC, DT, or PT. In Equation (12), the mean square error (MSE) is also defined.

$$SNR(\text{dB}) = 10 \times \log_{10} (E[FP^2] / E[(FP - FP_q)^2]); \quad MSE = E[(FP - FP_q)^2] / 2^{2L} \quad (13)$$

For comparison, we select state-of-the-art TEC schemes whose functions have a closed form, i.e., the generalized probabilistic estimation bias (GPEB) [16,17], probability estimation and computer simulation (PACS) [20], Booth-encoded sign-digit-based conditional probability (BSCP) [22], and SC-generator-based (SCG) [12] methods, as well as the DT and PT approaches. Table 5 presents the accuracy (i.e., the SNR) and hardware performances (area, critical-path delay, and power consumption) for a 16-bit FWBM using the aforementioned TEC and proposed BWATEC schemes, respectively.

Table 5. Comparison of accuracy and hardware performances of a 16-bit fixed-width Booth multiplier (FWBM) for PT, DT, and various truncation error compensation (TEC) schemes.

	PT	BSCP	PACS	GPEB	SCG	Ours	DT
Accuracy Performance ($L' = 16$)—SNR (dB)							
SNR	85.56	81.91	81.83	79.34	81.84	81.87	64.84
Hardware Performances—Area (μm^2)/Delay (ns)/Power (mW)							
Area	2294	1330	1301	1249	1325	1312	1098
Delay	3.62	3.25	3.24	3.20	3.24	3.27	2.96
Power	1075	615.1	585.2	562.3	612.9	591.0	486.4

In Table 5, the SNR results were obtained for operations of 16-bit data (i.e., $L' = 16$), based on the calculation of 30 K sets of 16-bit $A \times B$ Booth multiplication. Both the A and B operands were uncorrelated random 16-bit numbers with uniform distribution in statistics. The hardware parameters were provided by using the Synopsys Design Compiler, through logic synthesis with the TSMC 40 nm typical standard cell library for FWBM designs with no optional bit-shift processing at the output. According to References [12,22], we used a general sorting circuit based on Reference [12] for the BSCP and SCG designs in order to avoid the addition of negative digit values. In Table 5, the BSCP method achieves a better SNR than all other TEC-enabled designs. However, this result of the BSCP approach was obtained by using a complex TEC formula (i.e., Equation (19) in Reference [22]), and this function is difficult to be directly applied to a practical biasing circuit. The GPEB scheme outperforms other TEC-based works due to its use of a simple 1-bit or 2-bit constant TEC bias; however, the GPEB accuracy result is comparatively more reduced. Referring to the hardware parameters listed in Table 5, the results from the “area” and “power” items basically exhibit the same trend. To benchmark both the area efficiency and the accuracy, a design metric of area-delay-error product ($ADEP$), defined as “ $ADEP = \text{Area} \times \text{Delay} \times MSE$ ”, can be adopted to evaluate the overall design efficiency. As there is no TEC function involved in either DT or PT and the MSE magnitudes obtained by DT and PT are too extreme for the $ADEP$ evaluation, these two schemes are excluded from the $ADEP$ evaluation [22]. Table 6 lists the $ADEP$ results in percentage values (normalized to that of the GPEB case) for TEC-enabled designs.

Table 6. Comparison of $ADEP$ results of a 16-bit FWBM for various TEC schemes.

	BSCP	PACS	GPEB	SCG	Ours
$ADEP$ (%)	59.80%	59.35%	100%	60.32%	59.71%

As indicated in Table 6, the proposed design outperforms all listed schemes (i.e., a relatively small $ADEP$ value) except the PACS method. This is because additional data-selection multiplexers/switch and control logics are required in our design to enable the adjustment of the TEC function for multiple L' levels (refer to Figure 7). Such processing increases hardware costs and especially increases critical-path delay in our FWBM relative to other TEC designs, as shown in Table 5; however, the accuracy for $L' = 16$ in our case is comparatively improved by using deterministic $p_{0,7}$ and n_7 (refer to Section 3.2, Figure 4).

Nevertheless, the actual accuracy performance and hardware efficiency of the proposed design is manifested in the accuracy improvement for operations on L' -bit input patterns, giving $L' < 16$. Table 7 reports the SNR results for the TEC-enabled 16-bit FWBMs (i.e., works in Table 6) for operations of $L' = 14, 12, 10$, and 8. In Table 7, the SNR values were obtained based on the 16-bit product of FWBMs relative to the PT outcomes. As indicated in Table 7, our design achieves the highest SNR performances compared to other TEC-based designs for all listed L' cases because the proposed BWATEC scheme provides

more precise TEC biasing for various L' -bit inputs. In addition, higher SNR results can be achieved with smaller L' values by using the proposed design, due to more counts of deterministic R_Z/R_D elements. In practical designs, a slight improvement in the SNR results possibly results in an efficient enhancement in the system operation accuracy [21].

Table 7. SNR results (dB) of a 16-bit FWBM for various TEC schemes and L' -bit inputs.

	BSCP	PACS	GPEB	SCG	Ours
$L' = 14$	80.62	81.06	78.10	81.04	81.69
$L' = 12$	80.77	81.25	76.67	80.96	82.12
$L' = 10$	81.01	81.57	75.40	81.28	83.18
$L' = 8$	80.14	81.36	74.65	80.38	Inf.

Considering the overall design efficiency, Figure 9 illustrates the ADEP results from the TEC-based 16-bit FWBMs based on the MSE value relative to the PT products for operations of $L' = 14, 12, 10$, and 8, with ZP bits added to the input operand. In Figure 9, the ADEP values are normalized to the GPEB results, and the annotated percentage values represent the reduction of the ADEP achieved by the proposed design relative to all other listed methods. Figure 9 demonstrates that our scheme outperforms its contenders in terms of the ADEP values, achieving reductions of 7.9–50.9%, 17.1–69.5%, 29.9–82.2%, and 100% for the operations of input patterns with 14-bit, 12-bit, 10-bit, and 8-bit widths, respectively. Figure 9 shows that our design can achieve a more significant TEC effect with smaller specified L' values, as more ratios of deterministic values are used and associated with the R_H and R_D when using the proposed BWATEC scheme. For the case of $L' = 8$, our approach equivalently counts all $P.P.$ terms to obtain a full-width result that is the same as a PT outcome, and thus a 100% ADEP reduction can be achieved.

Figure 9. Normalized ADEP values of a 16-bit FWBM for various TEC schemes and L' .

As discussed in Sections 3.3 and 4, two specifications of “ $L = 2n$ (n is even)” and “ $L = 2n$ (n is odd)” are considered for the design scalability of an L -bit FWBM, using the proposed BWATEC scheme. Therefore, in addition to the case of $L = 16$ (for even n), another case of $L = 14$ (for odd n) was also evaluated for the ADEP performances in this section. Figure 10 illustrates the ADEP results from the TEC-enabled 14-bit FWBMs for operations of $L' = 12, 10, 8$, and 6, based on the same processing with that for the 16-bit FWBM evaluation. Figure 10 indicates that our design outperforms all other listed methods, achieving the significant ADEP reductions for the operations of inputs with 12-bit, 10-bit, 8-bit, and 6-bit

widths, respectively. Compared to the *ADEP* results for 16-bit FWBMs (Figure 9), the *ADEP* drops for all GPEB-excluded designs in relation to the GPEB base is reduced in Figure 10; however, the same trend of the *ADEP* reductions based on the relative value of L and L' is exhibited for our design associated with other TEC-based works.

Figure 10. Normalized *ADEP* values of a 14-bit FWBM for various TEC schemes and L' .

5.2. Design Verification and Experiments

5.2.1. CNN Acceleration Application

To verify an FWBM enabling the proposed BWATEC scheme, we implemented our design by using a SoC-FPGA platform and demonstrated the hardware acceleration for CNN inference operations. In a typical CNN accelerator, fixed-point operations are usually considered and a suitable bit width can be determined based on the CNN inference accuracy requirement [25,26]. Several studies have shown that the small bit width (e.g., 8-bit width or fewer) is sufficient for the model coefficients and operation precision, while preserve the inference accuracy [27–29]. However, a sufficiently high bit width (e.g., common 16 bits) is considered in several CNN accelerator approaches to ensure the precision required by various applications [30–32]. Moreover, different bit-widths can be specified for different CNN layers (e.g., the intermediate layers) to adjust the CNN performance [33,34]. Accordingly, several works have proposed CNN processing units that support operations with variable bit widths (e.g., 4/8/16-bit or 1-bit to 16-bit) [28,35–37]. In this study, an L -bit FWBM (e.g., our design example of a 16-bit FWBM) capable of processing input patterns with multiple L' -bit ($L' \leq L$) widths lends support to the aforementioned practical approaches.

5.2.2. SoC-FPGA Implementation

The employed SoC-FPGA-based platform uses a Xilinx Zynq-7000 SoC-FPGA device which integrates an ARM central processing unit (CPU) with the user-developed hardware side. Such a SoC-FPGA approach lends support to the CNN inference operations by using a software (SW)–hardware (HW) co-design scheme [38–40] by appropriately evaluating the SW–HW work division. For example, computation-expensive two-dimensional (2D) convolution is often accelerated at the HW side, while other low-effort CNN operations, such as maximum pooling, fully-connected (FC) layer execution and system controls are processed at the SW end [39,40]. Figure 11 shows the setup of our implementation that uses a SoC-FPGA approach based on the SW–HW co-design for CNN acceleration.

Figure 11. Schematic of the setup of our design implementation that uses a SoC-FPGA approach.

Referring to Figure 11, the division of HW and SW responsibilities in our CNN setup was as follows. The HW side was responsible for 2D (5×5) convolution, addition of an offset, activation function (i.e., rectified linear unit; ReLU), and maximum pooling, while the SW side performed residual low-complex operations (e.g., FC execution), HW operation mode setting, and system control. As depicted in Figure 11, the ARM CPU executes SW commands and communicates with the HW side through the AXI bus, and the data transferring between the external memory and the FPGA HW-side memory is executed through direct memory access (DMA). When the HW acceleration of each CNN layer was actuated, the feature map data, kernel weights, and control parameters were fetched from the external memory (e.g., DRAM) to the HW side via DMA transmission and stored in the block RAMs, data registers, and control registers, respectively. The 2D convolution accelerator then accessed those stored values for convolution operations and then sent the calculated result to the next module for the offset-addition, ReLU, and pooling operations. The final produced data of HW acceleration for each CNN layer were stored in the block RAMs and sent to the external memory through DMA for the follow-up SW processing.

In our design, the 2D (5×5) convolution accelerator employs 25 16-bit FWBMs with BWATEC, which can operate with multiple 16-bit, 14-bit, 12-bit, 10-bit, or 8-bit numerical input data. Depending on the tiling for each CNN layer, the block RAMs can be configured to store the data of input and output feature maps (images) with sizes from 32×32 to 128×128 . Table 8 lists the main HW resource usage on a Xilinx/Zynq-7000 SoC-FPGA device for our FPGA design, and the items include the lookup-table (LUT), flip-flop (FF), LUTRAM, and block RAM (BRAM) utilization. Table 8 also lists the HW performance of giga operations per second (GOPs), which is obtained by using a 50 MHz clock rate with values converted from the giga multiplication and addition operations [40].

Table 8. HW resources and performances for our FPGA implementation.

LUT Util.	LUTRAM Util.	FF Util.	BRAM Util.	GOPs
6786	62	2572	12.5	2.55

5.2.3. Electrocardiogram Classification Experiment

Based on our SoC-FPGA implementation and SW–HW co-design setup, an experiment was performed to demonstrate the electrocardiogram (ECG) classification. In this work, we used the standard MIT–BIH arrhythmia dataset [41] for the CNN model training and inference. To operate with ECG data by using a 2D CNN model [42,43], we transformed the one-dimensional MIT–BIH ECG signals into the (128×128) 2D ECG image by using the signal preprocessing technique presented in Reference [42]. Rather than the clinical ECG classification for seven or five arrhythmia classes [42,44], our experiment

merely classified ECG images into “normal” and “abnormal” heartbeats for wearable ECG monitor applications.

The experimental network for the aimed ECG classification was built up by using a simplified LeNet-5 CNN model [45]. As the contents summarized in Table 9, the built-up CNN model includes two convolution and maximum pooling layers, followed by the FC layers. Our CNN model was first determined by training process performed on a high-end computer in floating-point operations. For the CNN inference using an SW–HW co-design approach, the executions accelerated at the HW side were performed in fixed-point operations to achieve available overall accuracy (Table 9). To verify the proposed design, we implemented a 2D convolution unit consisting of 25 pcs 16-bit FWBMs with the proposed BWATEC function on the SoC-FPGA device. In our experiment, the same 2D convolution unit with one set of 16-bit FWBMs (25 pcs) was operated to perform the computation of two CNN layers. To demonstrate the BWATEC operations of our design, two phases of L' setting for the same set of 25 pcs 16-bit FWBMs (i.e., $L = 16$; L' -bit inputs) were adopted to execute two layers of CNN convolution execution. In phase 1, all 16-bit FWBMs in the 2D convolution unit were set to operate with 12-bit input data (i.e., $L' = 12$) for CNN layer-1 operations to consist with the numerical level of inputs. In phase 2, the same 16-bit FWBMs were set to process 16-bit input data (i.e., $L' = 16$) for CNN layer-2 operations to preserve the computation precision.

Table 9. Experimental CNN model architecture and SNR performances among layers.

Layers	Input Feature Map Size	Input Channel No.	Kernel Size	Output SNR (dB)
Input	128×128	1	—	—
1st Convolution	128×128	1	5×5	34.37
1st Max. Pooling	128×128	4	2×2	35.14
2nd Convolution	64×64	4	5×5	29.48
2nd Max. Pooling	64×64	8	2×2	29.86
FC	32×32	8	—	—

For evaluation, we also implemented contrast 2D convolution units composed of 16-bit FWBMs, using the BSCP, PACS, GPEB, and SCG TEC schemes on the same SoC-FPGA device. Table 10 lists the FPGA LUT resource utilization for a 2D convolution unit with various TEC-based FWM designs, using the aforementioned methods and our scheme. As shown in Table 10, our design achieves the medium level of area efficiency on FPGA, which is basically consist with the trend of area parameters listed in Table 7. However, the feature of our design for multiple setting of L' -bit operations lends support to the system development requiring flexible word lengths or improved accuracy. For a case study in addition to CNN acceleration, the devised 2D convolution unit can be restructured to realize a 25-tap finite impulse response (FIR) filter by inserting several multiplexers in the data paths of 25 pcs FWBMs. We also developed such an FIR with a slight HW overhead via FPGA implementation to use our design for digital signal processing applications.

Table 10. Comparison of lookup-table (LUT) resource usage for FPGA implementation of a 2D convolution unit for various TEC schemes.

	BSCP	PACS	GPEB	SCG	Ours
LUT Util.	5225	4907	4459	5206	4957

The ECG classification was checked by using a modified CNN inference model with the 2D convolution performed in our two-phase fixed-point operations. Moreover, the

experimental CNN operations (Table 9) were performed by using the SoC-FPGA based on our SW–HW co-design approach to obtain the inference results. For the bit-width setting (i.e., L') of two phases, the SW side would prepare the FWBM operands appropriately padded with ZP bits and set the BWATEC control for each round (i.e., layer 1 or 2) of CNN HW acceleration. The inference outcomes generated via the SoC-FPGA were further compared with the results generated by using the aforementioned fixed-point-operated CNN inference model for verification. After inference checking, the confusion matrix and performances of our ECG classification experiment are listed in Table 11. The performance results reported in Table 11 were obtained based on the accuracy (*Acc.*), sensitivity (*Sen.*), and specificity (*Spc.*) statistical metrics extracted from the confusion matrix [42,44]. The terms of TP, TN, FP, and FN denote true positive as “abnormal” (arrhythmia), true negative as “normal”, false positive as “abnormal”, and false negative as “normal” in the binary classification, respectively. The associated formulas are defined as follows:

$$Acc. = \frac{TP + TN}{TP + TN + FP + FN} \quad Sen. = \frac{TP}{TP + FN} \quad Spc. = \frac{TN}{TN + FP} \quad (14)$$

Table 11. Confusion matrix and performances of the experimental CNN for ECG classification.

<i>Prediction</i> → <i>Label</i> ↓	Abnormal	Normal	Metrics
Abnormal	27,577 (TP)	705 (FN)	97.5% (<i>Sen.</i>)
Normal	2266 (FP)	35,316 (TN)	94.0% (<i>Spc.</i>)
Metrics	95.5% (<i>Acc.</i>)		

6. Conclusions

In this paper, we presented a BWATEC scheme capable of providing an adjusted TEC function adaptive to various L' -bit input patterns of an L -bit FWBM, in which $L' \leq L$. Using different combinations of hybrid deterministic/probabilistic values associated with the R_H and R_D regions, the proposed BWATEC scheme can generate a tailored high-accuracy TEC bias for an L -bit FWBM, depending on the setting of L' (L and L' are scalable). An FWBM enabling the proposed BWATEC scheme can be realized by using a reconfigurable bias circuit in a $P.P.$ array with design scalability.

Taking a 16-bit FWBM as an example, we found that the approach using our BWATEC scheme exhibited design efficiency and different degrees of *ADEP* reduction for operations with 14-bit to 8-bit inputs, as compared to FWBM designs that used state-of-the-art TEC methods.

Moreover, the resultant 16-bit FWBM with BWATEC were verified by using the Xilinx Zynq-7000 SoC-FPGA based on the SW–HW co-design approach. The SoC-FPGA-based verification demonstrated the experimental CNN model for ECG classification.

Author Contributions: Conceptualization and methodology, S.-N.T.; review and editing, S.-N.T.; project administration, S.-N.T.; software and validation (experiment), J.-C.L., C.-K.C., P.-T.K. and Y.-S.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Ministry of Science and Technology of Taiwan, under project MOST 110-2221-E-033-039.

Informed Consent Statement: C-language simulation and Verilog modeling supports. We also thank Hong-Yu Ke and Jia-Nan Zhong for their works in the CNN model development for ECG classification.

Acknowledgments: The authors thank Yu-Shin Han and Jih-Hsiang Yeh for their works.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Parhi, K.K. *VLSI Digital Signal Processing Systems: Design and Implementation*, 1st ed.; Wiley: New York, NY, USA, 1999.
2. Lee, H.Y.; Park, I.C. Balanced Binary-Tree Decomposition for Area-Efficient Pipelined FFT Processing. *IEEE Trans. Circuits Syst. I Reg. Pap.* **2007**, *54*, 889–900. [\[CrossRef\]](#)
3. Chen, H.Y.; Lin, J.N.; Hu, H.S.; Jou, S.J. STBC-OFDM Downlink Baseband Receiver for Mobile WMAN. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2013**, *21*, 43–54. [\[CrossRef\]](#)
4. Tang, S.-N.; Han, Y.-S. A High-Accuracy Hardware-Efficient Multiply–Accumulate (MAC) Unit Based on Dual-Mode Truncation Error Compensation for CNNs. *IEEE Access* **2020**, *8*, 214716–214731. [\[CrossRef\]](#)
5. Van, L.-D.; Yang, C.-C. Generalized Low-Error Area-Efficient Fixed-Width Multipliers. *IEEE Trans. Circuits Syst. I Reg. Pap.* **2005**, *52*, 1608–1619. [\[CrossRef\]](#)
6. Tu, J.-H.; Van, L.-D. Power-efficient pipelined reconfigurable fixed-width Baugh-Wooley multipliers. *IEEE Trans. Comput.* **2009**, *58*, 1346–1355. [\[CrossRef\]](#)
7. Chang, C.-H.; Satzoda, R.K. A Low Error and High Performance Multiplexer-Based Truncated Multiplier. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2010**, *18*, 1767–1771. [\[CrossRef\]](#)
8. Petra, N.; Caro, D.D.; Garofalo, V.; Napoli, E.; Strollo, A.G.M. Design of Fixed-Width Multipliers with Linear Compensation Function. *IEEE Trans. Circuits Syst. I Reg. Pap.* **2011**, *58*, 947–960. [\[CrossRef\]](#)
9. Jou, S.-J.; Tsai, M.-H.; Tsao, Y.-L. Low-error reduced-width Booth multipliers for DSP applications. *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.* **2003**, *50*, 1470–1474. [\[CrossRef\]](#)
10. Chen, Y.-H.; Chang, T.-Y.; Jou, R.-Y. A statistical error-compensated Booth multipliers and its DCT applications. In Proceedings of the TENCON IEEE Region 10 Conference, Fukuoka, Japan, 21–24 November 2010; pp. 1146–1149. [\[CrossRef\]](#)
11. Song, M.A.; Van, L.D.; Kuo, S.Y. Adaptive low-error fixed-width Booth multipliers. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **2007**, *90*, 1180–1187. [\[CrossRef\]](#)
12. Wang, J.-P.; Kuang, S.-R.; Liang, S.-C. High-Accuracy Fixed-Width Modified Booth Multipliers for Lossy Applications. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2011**, *19*, 52–60. [\[CrossRef\]](#)
13. Kuang, S.-R.; Wang, J.-P.; Guo, C.-Y. Modified Booth Multipliers with a Regular Partial Product Array. *IEEE Trans. Circuits Syst. II Exp. Briefs* **2009**, *56*, 404–408. [\[CrossRef\]](#)
14. Cho, K.-J.; Lee, K.-C.; Chung, J.-G.; Parhi, K.K. Design of low-error fixed-width modified Booth multiplier. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2004**, *12*, 522–531. [\[CrossRef\]](#)
15. Juang, T.-B.; Hsiao, S.-F. Low-error carry-free fixed-width multipliers with low-cost compensation circuits. *IEEE Trans. Circuits Syst. II Exp. Briefs* **2005**, *52*, 299–303. [\[CrossRef\]](#)
16. Li, C.-Y.; Chen, Y.-H.; Chang, T.-Y.; Chen, J.-N. A Probabilistic Estimation Bias Circuit for Fixed-Width Booth Multiplier and Its DCT Applications. *IEEE Trans. Circuits Syst. II Exp. Briefs* **2011**, *58*, 215–219. [\[CrossRef\]](#)
17. Chen, Y.-H.; Li, C.-Y.; Chang, T.-Y. Area-Effective and Power-Efficient Fixed-Width Booth Multipliers Using Generalized Probabilistic Estimation Bias. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2011**, *1*, 277–288. [\[CrossRef\]](#)
18. Chen, Y.-H.; Chang, T.-Y. A High-Accuracy Adaptive Conditional Probability Estimator for Fixed-Width Booth Multipliers. *IEEE Trans. Circuits Syst. I Reg. Pap.* **2012**, *59*, 594–603. [\[CrossRef\]](#)
19. Chen, Y.-H. An Accuracy-Adjustment Fixed-Width Booth Multiplier Based on Multilevel Conditional Probability. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2015**, *23*, 203–207. [\[CrossRef\]](#)
20. He, W.-Q.; Chen, Y.-H.; Jou, S.-J. High-Accuracy Fixed-Width Booth Multipliers Based on Probability and Simulation. *IEEE Trans. Circuits Syst. I Reg. Pap.* **2015**, *62*, 2052–2061. [\[CrossRef\]](#)
21. Chen, Y.-H. Improvement of Accuracy of Fixed-Width Booth Multipliers Using Data Scaling Technology. *IEEE Trans. Circuits Syst. II Exp. Briefs* **2021**, *68*, 1018–1022. [\[CrossRef\]](#)
22. Zhang, Z.; He, Y. A Low-Error Energy-Efficient Fixed-Width Booth Multiplier with Sign-Digit-Based Conditional Probability Estimation. *IEEE Trans. Circuits Syst. II Express Briefs* **2018**, *65*, 236–240. [\[CrossRef\]](#)
23. Oklobdzija, V.G.; Villegier, D.; Liu, S.S. A method for speed optimized partial product reduction and generation of fast parallel multipliers using an algorithmic approach. *IEEE Trans. Comput.* **1996**, *45*, 294–306. [\[CrossRef\]](#)
24. He, Y.; Chang, C.-H. A New Redundant Binary Booth Encoding for Fast 2^n -Bit Multiplier Design. *IEEE Trans. Circuits Syst. I Reg. Pap.* **2009**, *56*, 1192–1201. [\[CrossRef\]](#)
25. Gong, L.; Wang, C.; Li, X.; Chen, H.; Zhou, X. MALOC: A Fully Pipelined FPGA Accelerator for Convolutional Neural Networks with All Layers Mapped on Chip. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **2018**, *37*, 2601–2612. [\[CrossRef\]](#)
26. Zhang, L.; Li, B.; Liu, Y.; Zhao, X.; Wang, Y.; Wu, J. FPGA Acceleration of CNNs-Based Malware Traffic Classification. *Electronics* **2020**, *9*, 1631. [\[CrossRef\]](#)
27. Moons, B.; Verhelst, M. An Energy-Efficient Precision-Scalable ConvNet Processor in 40-nm CMOS. *IEEE J. Solid-State Circuits* **2017**, *52*, 903–914. [\[CrossRef\]](#)
28. Camus, V.; Mei, L.; Enz, C.; Verhelst, M. Review and Benchmarking of Precision-Scalable Multiply-Accumulate Unit Architectures for Embedded Neural-Network Processing. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2019**, *9*, 697–711. [\[CrossRef\]](#)
29. Chen, Q.; Fu, Y.; Song, W.; Cheng, K.; Lu, Z.; Zhang, C.; Li, L. An Efficient Streaming Accelerator for Low Bit-Width Convolutional Neural Networks. *Electronics* **2019**, *8*, 371. [\[CrossRef\]](#)

30. Chen, Y.; Krishna, T.; Emer, J.S.; Sze, V. Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks. *IEEE J. Solid-State Circuits* **2017**, *52*, 127–138. [CrossRef]
31. Du, L.; Du, Y.; Li, Y.; Su, J.; Kuan, Y.-C.; Liu, C.-C.; Chang, M.-C.F. A Reconfigurable Streaming Deep Convolutional Neural Network Accelerator for Internet of Things. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2018**, *65*, 198–208. [CrossRef]
32. Jo, J.; Cha, S.; Rho, D.; Park, I. DSIP: A Scalable Inference Accelerator for Convolutional Neural Networks. *IEEE J. Solid-State Circuits* **2018**, *53*, 605–618. [CrossRef]
33. Zhou, S.; Wu, Y.; Ni, Z.; Zhou, X.; Wen, H.; Zou, Y. DoReFa-Net: Training Low Bitwidth Convolutional Neural Networks with Low Bitwidth Gradients. *arXiv* **2016**, arXiv:1606.06160. Available online: <https://arxiv.org/abs/1606.06160v2> (accessed on 11 October 2021).
34. Park, E.; Ahn, J.; Yoo, S. Weighted-Entropy-Based Quantization for Deep Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017. [CrossRef]
35. Shin, D.; Lee, J.; Lee, J.; Yoo, H.J. DNPU: An 8.1 TOPS/W reconfigurable CNN-RNN processor for general-purpose deep neural networks. In Proceedings of the IEEE Int. Solid-State Circuits Conference (ISSCC), San Francisco, CA, USA, 5–9 February 2017; pp. 240–241. [CrossRef]
36. Garofalo, A.; Tagliavini, G.; Conti, F.; Rossi, D.; Benini, L. XpulpNN: Accelerating Quantized Neural Networks on RISC-V Processors Through ISA Extensions. In Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, France, 9–13 March 2020; pp. 186–191. [CrossRef]
37. Lee, J.; Kim, C.; Kang, S.; Shin, D.; Kim, S.; Yoo, H. UNPU: An Energy-Efficient Deep Neural Network Accelerator with Fully Variable Weight Bit Precision. *IEEE J. Solid-State Circuits* **2019**, *54*, 173–185. [CrossRef]
38. Han, Y.; Virupakshappa, K.; Vitor Silva Pinto, E.; Oruklu, E. Hardware/Software Co-Design of a Traffic Sign Recognition System Using Zynq FPGAs. *Electronics* **2015**, *4*, 1062–1089. [CrossRef]
39. Guo, K.; Han, S.; Yao, S.; Wang, Y.; Xie, Y.; Yang, H. Software-Hardware Codesign for Efficient Neural Network Acceleration. *IEEE Micro* **2017**, *37*, 18–25. [CrossRef]
40. Moini, S.; Alizadeh, B.; Emad, M.; Ebrahimpour, R. A Resource-Limited Hardware Accelerator for Convolutional Neural Networks in Embedded Vision Applications. *IEEE Trans. Circuits Syst. II Express Briefs* **2017**, *64*, 1217–1221. [CrossRef]
41. Moody, G.B.; Mark, R.G. The impact of the MIT-BIH arrhythmia database. *IEEE Eng. Med. Biol. Mag.* **2001**, *20*, 45–50. [CrossRef]
42. Ju, T.J.; Nguyen, H.M.; Kang, D.; Kim, D.; Kim, D.; Kim, Y.-H. ECG arrhythmia classification using a 2-D convolutional neural network. *arXiv* **2018**, arXiv:1804.06812. Available online: <https://arxiv.org/abs/1804.06812> (accessed on 11 October 2021).
43. Wu, Y.; Yang, F.; Liu, Y.; Zha, X.; Yuan, S. A Comparison of 1-D and 2-D Deep Convolutional Neural Networks in ECG Classification. *arXiv* **2018**, arXiv:1810.07088. Available online: <https://arxiv.org/abs/1810.07088> (accessed on 11 October 2021).
44. Saadatnejad, S.; Oveisi, M.; Hashemi, M. LSTM-Based ECG Classification for Continuous Monitoring on Personal Wearable Devices. *IEEE J. Biomed. Health Inform.* **2020**, *24*, 515–523. [CrossRef]
45. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]