논문 2021-16-30

Highly Accurate Approximate Multiplier using Heterogeneous Inexact 4-2 Compressors for Error-resilient Applications

Jaewoo Lee, HyunJin Kim*

Abstract : We propose a novel, highly accurate approximate multiplier using different types of inexact 4-2 compressors. The importance of low hardware costs leads us to develop approximate multiplication for error-resilient applications. Several rules are developed when selecting a topology for designing the proposed multiplier. Our highly accurate multiplier design considers the different error characteristics of adopted compressors, which achieves a good error distribution, including a low relative error of 0.02% in the 8-bit multiplication. Our analysis shows that the proposed multiplier significantly reduces power consumption and area by 45% and 26%, compared with the exact multiplier. Notably, a trade-off relationship between error characteristics and hardware costs can be achieved when considering those of existing highly accurate approximate multipliers. In the image blending, edge detection and image sharpening applications, the proposed 8-bit approximate multiplier shows better performance in terms of image quality metrics compared with other highly accurate approximate multipliers.

Keywords : Approximate computing, Approximate multiplier, Approximate image processing, Inexact compressor, Error-resilient application

Ⅰ. Introduction

Growing demands in mobile and embedded applications have driven the need for implementations with low power consumption and hardware costs. Approximate computing has emerged to meet the need for low-cost implementations [1]. In the last years, several approximate multipliers using inexact compressors and have been proposed. When replacing exact compressors with inexact ones in a multiplier design, mulitiplier outputs are erroneous at the cost of simplifying the multiplier design. Thus, the approximate multipliers adopting inexact compressors cannot avoid a loss of accuracy. Nevertheless, applications that tolerate some degree of error can accept the approximate multiplication outputs.

One of the error-resilient applications can be image processing in that erroneous outputs may not go beyond the limits of human perceptions. For example, the convolution adopts convolutional filters computing the sum of the products, so a lot of multiplication operations are required. The error neutralization may happen when negative and positive errors are summated. Therefore, approximate multiplication can be an efficient option for the convolution along with reducing hardware costs. In the image blending, each pixel in the same location is approximately multiplied, which could not be out of bounds in human perceptions when noises are acceptable.

However, several approximate multipliers using inexact compressors do not show good error-resilient characteristics considering bad image processing results. As can be expected, the greater the hardware cost reduction, the lower the accuracy in general. An approximate multiplier for a target application can be selected by considering the trade-off relationship. In this paper, the highly accurate approximate multiplier is proposed to provide good output image qualities.

We present a novel approximate multiplier using two kinds of inexact 4-2 compressors with low error probabilities. The two inexact 4-2 compressors have different biased error characteristics. We employed these compressors to maximize the neutralization of negative and positive errors in each compressor, making the sum of the error biases on both sides close to zero. It is noted that both different types of compressors share an error-prone condition so that the design rule configures

*Corresponding Author (hyunjin2.kim@gmail.com)

Received: Aug. 23, 2021, Revised: Sep. 23, 2021, Accepted: Sep. 24, 2021

Jaewoo Lee: School of EEE, Dankook University (B.S.)
HyunJin Kim: School of EEE, Dankook University (Assoc. Prof.)

※ This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No.2021R1F1A1048054). The EDA tool was supported by the IC Design Education Center(IDEC), Korea. This research was results of a study on the "HPC Support" Project, supported by the 'Ministry of Science and ICT' and NIPA.

ⓒ IEMEK J. Embed. Sys. Appl. 2021 Oct. 16(5) 233-240
ISSN : 1975-5066
http://dx.doi.org/10.14372/IEMEK.2021.16.5.233

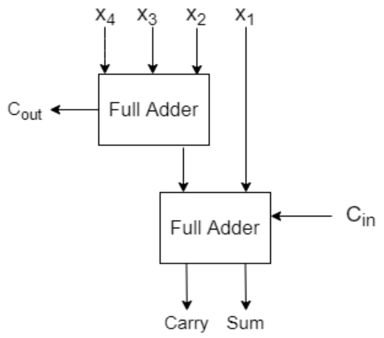


Fig. 1. Exact 4-2 compressors.

the topology in a way that reduces the error occurrence considering the different probability of each operand value. Notably, the relative error is significantly reduced by adjusting the topology. Our analysis shows that the proposed multiplier reduces hardware area costs by 26.1% and power consumptions by 27.0%, compared to the exact multiplier while keeping high accuracy. Besides, the proposed design shows good trade-off relationship between the accuracy and hardware cost. Due to its high accuracy, the image blending, edge detection and image sharpening results using the proposed multiplier show negligible difference from those using the exact multiplier.

Ⅱ. Related Works

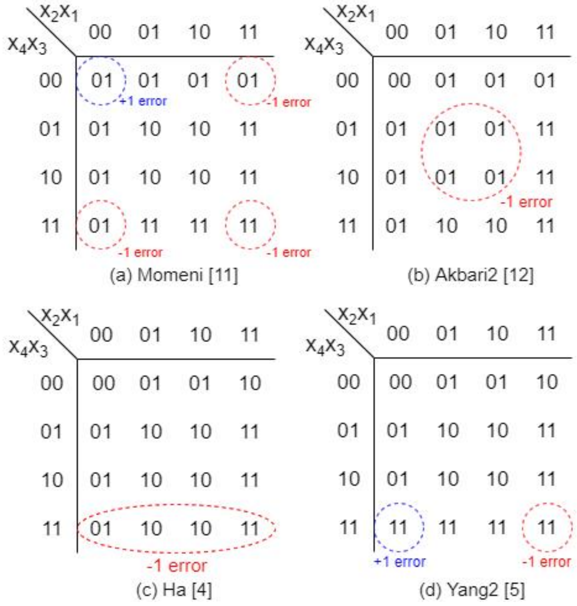
1. Exact Compressor

Compressors are used to take multiple inputs and express the sum of them in fewer outputs. For example, n-2 compressor outputs the sum of n inputs as 2 output bits. The full adder can be viewed as a 3-2 compressor. The conventional multipliers using the carry-save unit adopt only full and half adders, which proceed with the partial product reduction [2, 3].

While 3-2 compressions using full adders need 5 steps in 8-bit multiplication, it is known that 4-2 compression can reduce the number of steps in partial product reductions with efficient hardware costs. The 4-2 compressor can have 4-bit input and 2-bit output. Approximate multipliers using inexact 4-2 compressors requires only 3 steps to produce final multiplication outputs. Exact 4-2 compressor uses two full adders in Fig. 1 [4].

2. Inexact 4-2 Compressors and Partial Product Reductions

Unlike exact 4-2 compressors, inexact 4-2 compressors have only 4-bit input, x_1, x_2, x_3, x_4 and 2-bit output denoted as Carry and Sum. In all inexact 4-2 compressors, a negative error cannot be avoided when $x_4x_3x_2x_1 = 1111_2$, because 2 bits cannot represent the value of 4_{10}. Besides, an inexact 4-2 compressor has error entries depending on its truth table. Nevertheless, inexact 4-2 compressors can be attractive because of their low hardware costs.



(a) Momeni [11]

X4X3 \ X2X1	00	01	10	11
00	01	01	01	01
01	01	10	10	11
10	01	10	10	11
11	01	11	11	11

(b) Akbari2 [12]

X4X3 \ X2X1	00	01	10	11
00	00	01	01	01
01	01	01	01	11
10	01	01	01	11
11	01	10	10	11

(c) Ha [4]

X4X3 \ X2X1	00	01	10	11
00	00	01	01	10
01	01	10	10	11
10	01	10	10	11
11	01	10	10	11

(d) Yang2 [5]

X4X3 \ X2X1	00	01	10	11
00	00	01	01	10
01	01	10	10	11
10	01	10	10	11
11	11	11	11	11

+1 error
-1 error
-1 error
-1 error
-1 error
-1 error
+1 error
-1 error

Fig. 2. Truth tables of inexact 4-2 compressors.

X4X3 \ X2X1	00	01	10	11
00	$\frac{81}{256}$	$\frac{27}{256}$	$\frac{27}{256}$	$\frac{9}{256}$
01	$\frac{27}{256}$	$\frac{9}{256}$	$\frac{9}{256}$	$\frac{3}{256}$
10	$\frac{27}{256}$	$\frac{9}{256}$	$\frac{9}{256}$	$\frac{3}{256}$
11	$\frac{9}{256}$	$\frac{3}{256}$	$\frac{3}{256}$	$\frac{1}{256}$

Fig. 3. Probability table for 4-bit inputs.

Inexact 4-2 compressors in [5-7], and [8] are denoted as Momeni, Akbari2, Ha and Yang2, respectively, in the latter part of this paper. The truth tables of these inexact compressors are shown in Fig. 2. The implementation of Momeni [5] and Akbari2 [6] can be simple because of their high error rate. On the other hand, Ha [7] and Yang2 [8] are used to implement highly accurate approximate multipliers due to their low error rate [9].

When the approximate multiplier adopts the inexact compressors, partial products are used as inputs of the compressors. Let us assume that evenly distributed two integer values are multiplied. Because each bit of the integer inputs has the probabilities of being 0 and being 1 each equal to 0.5, then the probability of a partial product being 1 would be 0.25. If the partial products are used as inputs of the compressors, the probabilities of having inputs $x_4x_3x_2x_1 = 0000, \cdots, 1111$ in a compressor are shown in Fig. 3. The error probability can be calculated by summing the multiplication with the error in Fig. 2 and the probability in Fig. 3 for each input. When the error probability is denoted as B, Bs of Ha and Yang2 are calculated as:

$$B(Ha) = -1 \times \left(\frac{9}{256} + \frac{3}{256} + \frac{3}{256} + \frac{1}{256}\right) = -\frac{16}{256}, \tag{1}$$

$$B(Yang2) = (+1) \times \frac{9}{256} + (-1) \times \frac{1}{256} = +\frac{8}{256}. \tag{2}$$

The positive error probability of Yang2 results in a positive bias in the error distribution. Although Momeni and Ha have the same number of error cases, the values of the probability table and the truth tables show that Momeni has a much higher error probability when evenly distributed values are used as inputs. Momeni, Akbari2, Ha, and Yang2 have the error probabilities of -0.391, -0.141, -0.063, +0.039, respectively. For evenly distributed inputs, it is noted that Ha and Yang2 can have small error probabilities.

3. Partial Product Reduction Scheme

Assuming that the two n-bit inputs in n-bit multiplication are denoted as $A = a_n a_{n-1} a_{n-2} \cdots a_1 a_0$ and $B = b_n b_{n-1} b_{n-2} \cdots b_1 b_0$, the multiplication, $A \times B$, is formulated as follows:

$$A \times B = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} 2^{i+j} a_i b_j. \tag{3}$$

An intuitive way to show it is the partial product matrix (PPM). The PPM is reduced using partial products reduction schemes until no more than two reduced partial products remain in each column. The remaining partial products are turned into the final output with carry propagate addition.

Assuming inexact 4-2 compressors are used for 8-bit multiplication, the reduction scheme requires 3 steps as shown in Fig. 4. Each dot denotes the partial products. The boxes indicate adders and compressors. In Fig. 4, the grids represent partial products of Dadda multiplier. First, 8×8 partial products are generated, with up to 8 partial products in each column $2^{15} \sim 2^0$. The number of partial products is reduced to 4 or less in each column with inexact 4-2 compressors in step 1. The number of reduced partial products from step 1 are again reduced to 2 or less with inexact compressors in step 2, and summed to a 15-bit output with the carry-propagate adder in step 3.

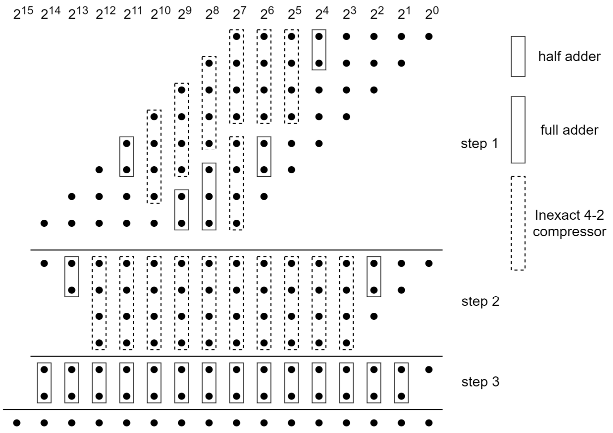
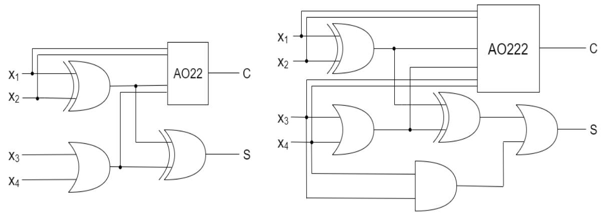


Fig. 4. Reduction scheme with inexact 4-2 compressors.



(a) Ha compressor [4]

(b) Yang2 compressor [5]

Fig. 5. Schematic of inexact 4-2 compressors used in the proposed multiplier.

Ⅲ. Proposed Design

1. Inexact Compressors in the Proposed Multiplier

For designing a more accurate approximate multiplier, we use two different types of inexact compressors, Ha and Yang2. The schematics of the inexact compressors, Ha and Yang2, are shown in Fig. 5. From [10], it is expected that the multiplier design adopting different types of compressors could provide better error characteristics. Compared with Momeni and Akbari2 in Fig. 2, Ha and Yang2 have considerably low error probabilities, so that the compressor can be suitable for our highly accurate approximate multiplier design. It is motivated that the different error characteristics of the inexact compressors can help enhance the performance by increasing error neutralization. Besides, all errors of both Ha and Yang2 compressors occur when $x_4x_3 = 11$, as shown in Fig. 2. In the following, the proposed multiplier design considers the input value that can occur errors and different error probabilities of the two different compressors.

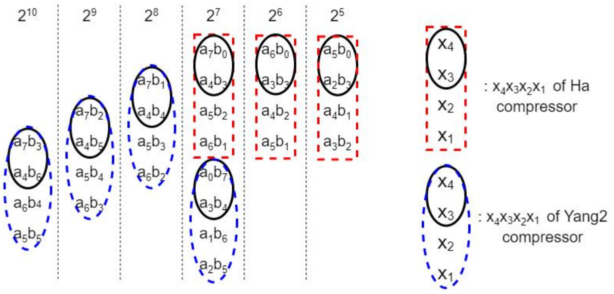


Fig. 6. Proposed topology design in step 1.

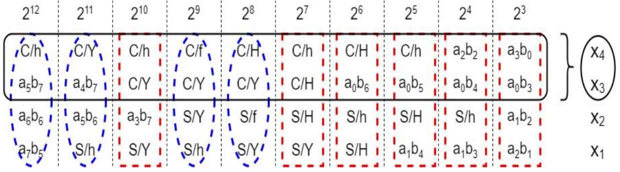


Fig. 7. Proposed topology design in step 2.

2. Proposed Multiplier Design

In order to neutralize the negative and positive biases of the two compressors close to zero, Ha compressors, which have larger bias values, are usually placed in the least significant bit (LSB) and Yang2 compressors are placed in the most significant bit (MSB) as shown in Fig. 6.

In the proposed design, the error rate and relative error are further reduced by applying the following rules:

Rule 1) Use the two partial products with the two lowest probabilities of being 1 as input x_4x_3 of a compressor because errors occur when $x_4x_3 = 11$.

Rule 2) If inputs of a compressor have the equal probabilities of being 1, set the topology in way that the output have the highest expected value when an error occurs.

Rule 1 reduces the error occurrence and Rule 2 helps the proposed design have the lower relative error. Let us assume each bit of multiplication inputs $A = a_7a_6 ... a_0$ and $B = b_7b_6 \cdots b_0$ are uniformly distributed into 0 and 1. The probability that any partial products of step 1, a_ib_j, is 1 becomes 0.25. Therefore, in step 1, the topology is selected by Rule 2.

As an example of Rule 2, two of the four inputs $(a_7b_{0,}\ a_6b_{1,}\ a_5b_{2,}\ a_4b_3)$ of the Ha compressor at column 2^7 in Fig. 6 would be x_4 and x_3. Let us assume that a_7b_0, a_4b_3 are used as x_4 and x_3. If an error occurs in this compressor, it means that $a_7b_0 = a_4b_3 = 1$, which implies $a_7 = b_0 = a_4 = b_3 = 1$. The case where $a_7b_0 = a_4b_3 = 1$ has a larger expected value than any other case where any two inputs are x_4x_3, because $a_7 = b_3 = 1$ is the largest positive factor for the expected value in this case. The expected value in case of an error can be calculated by expecting a value of 0.5 for all other input bits of A and B, except for the four input bits that are elements of x_4x_3. Fig. 6 shows $x_4x_3x_2x_1$ of inexact compressors in step 1.

In step 2, Sum and Carry of half adder, full adder, Ha compressor, and Yang2 compressor all have different probabilities of being 1. There are also partial products that are taken over directly to step 2 without being the inputs of adders or compressors in step 1.

Sum, Carry, half adder, full adder, Ha compressor, and Yang2 compressor will be referred as terms S, C, h, f, H and Y, respectively, and the probabilities will be denoted as P, taking the types of the output and the compressor as variables in the following (for example, the probability that Sum of half adders can be 1 will be denoted as $P(S/h)$).

$$
\begin{aligned}
&P(S/h) = p(x_2x_1 = 01) + p(x_2x_1 = 10) = 0.375 \\
&P(C/h) = p(x_2x_1 = 11) = \frac{1}{16} = 0.0625 \\
&P(S/f) = p(x_3x_2x_1 = 001 \mid 010 \mid 100 \mid 111) = 0.4375 \\
&P(C/f) = p(x_3x_2x_1 = 011 \mid 101 \mid 110 \mid 111) \fallingdotseq 0.1563 \\
&P(S/H) = p(x_4x_3x_2x_1 = 0001 \mid 0010 \mid 0100 \mid \dots \mid 1111) \\
&= \frac{27+27+27+3+27+3+9+1}{256} = \frac{124}{256} \fallingdotseq 0.4844 \\
&P(C/H) = p(x_4x_3x_2x_1 = 0101 \mid 0110 \mid 0111 \mid \dots \mid 1111) \qquad (4) \\
&= \frac{9+9+3+9+9+3+3+3+1}{256} = \frac{49}{256} \fallingdotseq 0.1914 \\
&P(S/Y) = p(x_4x_3x_2x_1 = 0001 \mid 0010 \mid \dots \mid 1110 \mid 1111) \\
&= \frac{27+27+27+3+27+3+9+3+3+1}{256} = \frac{130}{256} \fallingdotseq 0.5078 \\
&P(C/Y) = p(x_4x_3x_2x_1 = 0101 \mid \dots \mid 1100 \mid \dots \mid 1111) \\
&= \frac{9+9+3+9+9+3+9+3+3+1}{256} = \frac{58}{256} \fallingdotseq 0.2266
\end{aligned}
$$

For example, the inputs of Yang2 compressor located at 2^9 in step 2 are Sum of Yang2, Sum of the half adder, Carry of the full adder and Carry of Ha, and their probabilities of being 1 are 0.5078, 0.375, 0.1563 and 0.1914, respectively. Therefore, Carry of the full adder and Carry of Ha with the two lowest probabilities, 0.1563 and 0.1914, are chosen as x_4x_3 of the Yang compressor at column 2^9.

In this way, in step 2, Rule 1 is applied first and then Rule 2 is applied if there are more than one option to select two inputs with the two lowest probabilities of the inexact compressors. The topology selected in step 2 is shown in Fig. 7. The square dotted lines and the circle dotted lines represent the Ha compressors and Yang2 compressors, respectively, and the circle solid lines represent the x_4 and x_3 of each compressor in step 1 and step 2.

Ⅳ. Error Analysis

We compared the accuracies of the approximate multipliers in terms of several error metrics. Other approximate multipliers were made in the form of replacing the inexact compressors part in Fig. 4 with the corresponding compressors. The topologies of the multipliers only using Ha compressors and only using Yang2 compressors were the same with the proposed multiplier design. The outputs of the exact and approximate multipliers are denoted as M and M', respectively, and maximum output ($(2^8-1)^2$ in 8-bit unsigned multiplication) are donoted as $MaxOut$. The error metrics in [9] were adopted, which are defined as follows:

- ER (Error Rate) : the percentage of error occurrence.
- ED (Error Distance) = $|M-M'|$.
- MED (Mean Error Distance) : the average value of ED.
- NMED (Normalized Mean Error Distance) = $\frac{MED}{MaxOut}$.
- RED (Relative Error Distance) = $\frac{|M-M'|}{M}$ when $M\neq 0$.
- MRED (Mean Relative Error Distance) : the average value of RED.
- RERR (Relative Error) = $\frac{M'-M}{M}$.
- MRERR (Mean Relative Error) : the average value of RERR.
- PRED : the percentage that RED is higher than 2%
- PRED15 : the percentage that RED is higher than 15%

ER, NMED, MRED, MRERR and PRED are widely-used error metrics, and PRED15 is included to show whether RED exceeds 15% or not. These error metrics will be calculated including all 256^2 cases of 8-bit unsigned multiplication. All error metrics except for NMED were expressed in the form of percentages for ease of understanding.

As shown in Table 1, the multiplier with Momeni compressor had MRED and MRERR over 400%, which means that the approximate multiplication results have significant differences from the exact multiplication results.

Table 1. Error analysis of the 8-bit unsigned multipliers with adjusted topologies

Type	ER	NMED	MRED	MRERR	PRED	PRED15
Momeni	98.4%	0.0473	414%	410%	82.6%	50.2%
Akbari2	75.5%	0.0205	8.22%	-8.22%	57.0%	20.1%
Ha	40.0%	0.0062	1.61%	-1.61%	22.3%	0.9018%
Yang2	31.0%	0.0032	0.85%	0.36%	13.4%	0.0031%
Proposed	35.7%	0.0033	0.90%	0.02%	14.4%	0.0031%

Table 2. Error analysis of 8-bit unsigned multipliers with the unadjusted topologies

Type	ER	NMED	MRED	MRERR	PRED	PRED15
Ha	73.5%	0.0139	5.30%	-5.30%	54.6%	10.1%
Yang2	62,8%	0.0074	3.18%	2.72%	38.2%	5.12%
Proposed	69.3%	0.0073	3.10%	0.19%	40.2%	4.39%

Because Momeni compressor has an error when input is zero, large MREERs happened. Table 1 shows that 98.4% in the multiplication with Momeni compressors and 75.5% in the multiplication with Akbari2 compressors indicate the multipliers have significant error rates.

The proposed multiplier showed the lowest MRERR, because it was designed not to be biased to one side by cancelling opposing errors of the two heterogeneous compressors. On the other hand, the different multipliers used only one type of compressor, so MRERR depended on the bias of the compressor. The proposed multiplier has low MRED and PRED by applying Rule 2. Since the Ha compressor had a higher error probability than the Yang2 compressor, the multiplier with Ha compressor showed higher ER than the multiplier with Yang2 compressors. The proposed multiplier using both compressors showed ER in between them. The multipliers with Yang2 compressor and the proposed multiplier had similar NMED, MRED, MRERR, PRED, and PRED15, while the multiplier with Ha compressor was less accurate.

Table 2 shows the error metrics for the three multipliers including one with Ha compressor, one with Yang2 compressor and the proposed multiplier, where their topology was not adjusted according to the two rules. The topology was selected in a way that its x_4x_3 and x_2x_1 were the same as x_2x_1 and x_4x_3 of the proposed topology, respectively.

As shown in Table 1 and Table 2, the approximate multipliers that have the unadjusted topologies were much less accurate than the approximate multipliers that have the adjusted-topologies. Since the errors of the two compressors used in these multipliers are concentrated in the specific case (when $x_4x_3=11$), the error performance of each multiplier was very different depending on whether the topology was adjusted or not.

Ⅴ. Hardware Analysis

The 8-bit unsigned multipliers were coded using Verilog HDL and synthesized in 32nm standard cell library using Synopsys Design Compiler [11]. Table 3 shows the critical path, hardware area and power consumption of the multipliers. In the synthesis, 1.25 GHz was targeted for the toggle frequency. The multipliers using the Ha compressors and Yang2 compressors had the longest critical path, even longer than the exact multiplier. The multiplier using the Momeni compressors showed the best performance in the critical path, hardware area and power consumption and the multiplier using Akbari2 compressors was followed in the critical path and power consumption. In contrast, the multiplier using Yang2 compressors performed the worst. The multiplier using Ha compressors had hardware area close to that of the multiplier using Akbari2 compressors and power consumption similar to that of the proposed multiplier. The proposed multiplier showed a meaningfully reduced hardware area compared with the multiplier using Yang2 compressors.

Ⅵ. Applications

Image blending, edge detection and image sharpening were chosen as the applications to evaluate the performance of the approximate multipliers in Table 1 [9]. In the image blending, two 8-bit input images were multiplied by pixels at the same location, and the results were scaled back to the 8-bit outputs. The output image was superimposed by two input images. The peak signal to noise ratio ($PSNR$) and the mean structural similarity index ($MSSIM$) was used to assess the quality of the output images. $PSNR$ is calculated as follows:

$$PSNR = 10\log_{10}(\frac{MAX_I^2}{MSE}). \qquad (5)$$

MAX_I, the maximum value in the image, is 255 in 8-bit grayscale. And terms I and K denote the pixels of the exact image and the output image, respectively, MSE is defined as follows:

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j)-K(i,j)]^2. \qquad (6)$$

Table 3. Hardware comparison of 8-bit unsigned multipliers

Type	Critical path(ns)	Area(μm^2)	Power(μW)
Exact	0.44	125.07	37.212
Momeni	0.35	84.72	15.351
Akbari2	0.37	88.84	20.869
Ha	0.45	88.36	27.072
Yang2	0.45	104.96	28.896
Proposed	0.42	92.44	27.153

Table 4. Image blending results

	Cameraman Lena		Boat House		Girlface Bridge	
Type	MSSIM	PSNR	MSSIM	PSNR	MSSIM	PSNR
Momeni	72.18	24.23	79.72	25.90	61.62	23.13
Akbari2	95.01	32.88	93.12	31.64	95.92	35.52
Ha	98.54	41.11	98.36	40.30	99.53	46.12
Yang2	99.62	47.05	99.38	44.76	99.73	49.70
Proposed	99.62	47.52	99.40	45.14	99.73	49.98

MSE depends on how different the pixels of the exact image and the output image are. If the exact image and the output image are the same, the MSE becomes 0.

As shown in Table 4, the multiplier using Momeni compressors showed the worst performance in image blending. The multiplier using Ha compressors and the multiplier using Yang2 compressors had good $MSSIM$ and $PSNR$, but the latter was somewhat higher than the former. And the proposed multiplier showed the highest $MSSIM$ and $PSNR$ for all images. Fig. 8 shows the blended images of "Cameraman" and "Lena" with the exact and approximate multipliers.

Table 5 shows that the multiplication with Momeni compressor also had much lower $MSSIM$ than that with Akbari2 compressor in edge detection. In the edge detection, 5×5 Sobel convolution kernels in [12] were adopted. The convolutional outputs were normalized using the maximum and minimum values among them. It can be expected that the image processing with the multiplier using Momeni compressors would be significantly degraded

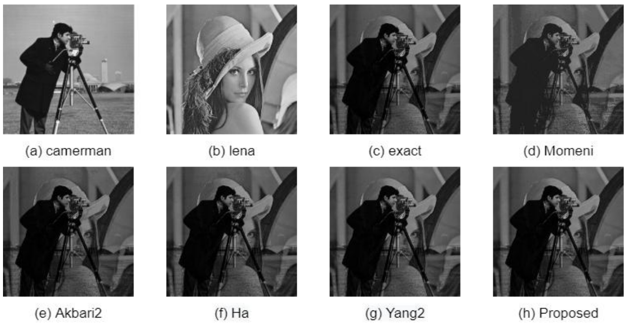

(a) camerman (b) lena (c) exact (d) Momeni (e) Akbari2 (f) Ha (g) Yang2 (h) Proposed

Fig. 8. Image blending results.

Table 5. Edge detection results

	Lena		Boat		House	
Type	MSSIM	PSNR	MSSIM	PSNR	MSSIM	PSNR
Momeni	19.29	18.68	13.65	17.34	22.19	17.17
Akbari2	80.46	28.96	83.89	27.70	81.43	27.40
Ha	100.00	×	100.00	×	100.00	×
Yang2	100.00	×	100.00	×	100.00	×
Proposed	100.00	×	100.00	×	100.00	×

Table 6. Image sharpening results

	Cameraman		Boat		House	
Type	MSSIM	PSNR	MSSIM	PSNR	MSSIM	PSNR
Momeni	29.76	8.68	33.67	8.75	48.45	9.66
Akbari2	96.80	22.71	97.74	27.15	98.23	26.68
Ha	99.85	38.08	99.91	44.20	99.90	44.93
Yang2	99.92	44.81	99.94	45.86	99.90	44.53
Proposed	99.89	41.06	99.93	45.91	99.90	45.29

in quality. The *PSNR* of the proposed multiplier, the multiplier with Ha compressor and the multiplier with Yang2 compressor were not defined in the edge detection, because the *MSE*s, the denominator in *PSNR*, of them were 0. It means that the output image using those multipliers with the proposed topology design was perfectly the same with the image using the exact multiplier, and that is why *MSSIM*s of them were 100.

As shown in Table 6, the multiplier using Momeni compressors still showed the worst performance due to its high error rate and the multiplier using Akbari2 compressors showed some improvement compared to edge detection, in image sharpening. The proposed multiplier generally showed outstanding performance similar to the multiplier using Yang2 compressors, and the multiplier using Ha compressors followed them.

Ⅶ. Conclusion

This paper presents a new multiplier that can provide high accuracy and meaningfully reduced hardware cost with the proposed design topology. The key point is that MRERR is controllable by using heterogeneous compressors having different error characteristics. The accuracy can be improved by adjusting the design topology, especially when using compressors that share the error-prone condition. The proposed design has MRERR and PRED15 very close to zero, while having 27% less power consumption than using the exact multiplier.

There is no approximate multiplier that provides the best option for all applications. It is necessary to find the most efficient trade-off point between the accuracy and hardware cost, considering the required accuracy and detailed characteristics of the application. The proposed multiplier is suitable for applications requiring high precision while having error resilience.

References

[1] J. Han, M. Orshansky, “Approximate Computing: An Emerging Paradigm for Energy-efficient Design,” in Proc. 18th IEEE Eur. Test Symp., pp. 1 - 6, 2013.

[2] C. S. Wallace, “A Suggestion for a Fast Multiplier,” IEEE Trans. Electron. Comput., Vol. EC-13, No. 1, pp. 14 - 17, 1964.

[3] L. Dadda, “Some Schemes for Fast Serial Input Multipliers,” in Proc. IEEE 6th Symp. Comput. Arithmetic (ARITH), Vol. 34, pp. 349 - 356, 1983.

[4] S. Sowmiya, K. Stella, V. M. Senthilkumar, “Design and Analysis of 4-2 Compressor for Arithmetic Application,” Asian Journal of Applied Science and Technology (AJAST), Vol. 1, No. 1, pp. 106-109, 2017.

[5] A. Momeni, J. Han, P. Montuschi, F. Lombardi, “Design and Analysis of Approximate Compressors for Multiplication,” IEEE Transactions on Computers, Vol. 64, No. 4, pp. 984 - 994, 2015.

[6] O. Akbari, M. Kamal, A. Afzali-Kusha, M. Pedram, “Dual-quality 4:2 Compressors for Utilizing in Dynamic Accuracy Configurable Multipliers,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., Vol. 25, No. 4, pp. 1352 - 1361, 2017.

[7] M. Ha, S. Lee, "Multipliers With Approximate 4-2 Compressors and Error Recovery Modules," IEEE Embedded Systems Letters, Vol. 10, No. 1, pp. 6 - 9, 2017.

[8] Z. Yang, J. Han, F. Lombardi, “Approximate Compressors for Error-resilient Multiplier Design,” in Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Nanotechnol. Syst., pp. 183 - 186, 2015.

[9] A. G. M. Strollo, E. Napoli, D. De Caro, N. Petra, G. Di Meo, “Comparison and Extension of Approximate 4-2 Compressors for Low Power Approximate Multipliers,” IEEE Transactions on Circuits and Systems I: Regular Papers, Vol. 67, 2020.

[10] D. Esposito, A. G. M. Strollo, E. Napoli, D. De Caro, N. petra, "Approximate Multipliers Based on new Approximate Compressors," IEEE Transactions on Circuits and Systems I: Regular Papers, Vol. 65, No. 12, pp. 4169-4182, 2018.

[11] Kurup, Pran, Taher Abbasi. “Logic Synthesis Using Synopsys®,” Springer Science & Business Media, 2012.

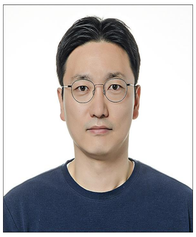
[12] “FilterSobel, Vol. 2, Image Processing, Intel Integrated Performance Primitives Developer Reference,” 2020.

Jaewoo Lee (이 재 우)

2015~Electronic and Electrical Engineering from Dankook University, Yongin, Republic of Korea (B.S. Candidate)

Field of Interests: Digital Circuit Design, Semiconductor Process Design
Email: cjswoto2005@naver.com

HyunJin Kim (김 현 진)



1997~Electrical Engineering from Yonsei University, Seoul, Republic of Korea (B.S.)
1999~Electrical Engineering from Yonsei University, Seoul, Republic of Korea (M.S.)
2010~Electrical and Electronic Engineering from Yonsei University, Seoul, Republic of Korea (Ph. D.)

Field of Interests: Neural Network Implementation and its Application, Pattern Matching Engine, Algorithm and Technique for Enhancing Networking Performance
Email: hyunjin2.kim@gmail.com