An Accuracy Reconfigurable Vector Accelerator based on Approximate Logarithmic Multipliers

Abstract— The logarithmic approximate multiplier proposed by Mitchell provides an efficient alternative to accurate multipliers in terms of area and power consumption. However, its maximum error of 11.1% makes it difficult to deploy in applications requiring high accuracy. To widely reduce the error of the Mitchell multiplier, this paper proposes a novel operand decomposition method which decomposes one operand into multiple operands and calculates them using multiple Mitchell multipliers. Based on this operand decomposition, this paper also proposes an accuracy reconfigurable vector accelerator which can provide a required computational accuracy with a high parallelism. The proposed vector accelerator dramatically reduces the area by more than half from the accurate multiplier array while satisfying the required accuracy for various applications. The experimental results show that our proposed vector accelerator behaves well in image processing and robot localization.

I. Introduction

Modern microprocessors mostly integrate vector multiplication units to cope with intensive computing requirements. However, a multiplier is also known as a power-hungry and area-expensive unit in a circuit. In applications involving a large number of multi-bit data or floating-point multiplication, the area and power dissipation of multipliers become dominant.

On the other hand, in many applications, the result of multiplication does not need to be accurate. For applications that require massive parallelism, a multiplier array with smaller area and higher speed is needed [1]. Thus, many approximation methods are proposed to improve the performance of multiplication. For example, approximate unsigned multipliers have been applied to some image processing applications, like sharpening and smoothing [2]-[5]. These researchers' proposals focus on bitwise truncation or compression of partial products of Wallace trees, with limited effect on the reduction of the overall area and energy consumption of the multiplier.

The logarithmic approximate multiplier proposed by Mitchell [6] converts multiplication to simpler shift and addition operations. Mitchell multiplier has been experimentally confirmed that it can significantly reduce the area to as low as 34% of an accurate multiplier. It is very effective for applications such as deep neural networks (DNN) that include massive parallel computing and do not necessarily achieve a high computational accuracy [7] [8]. However, for those applications that require a specific degree of accuracy and high parallelism simultaneously, the accuracy of the Mitchell multiplier needs improvement. One naive approach to achieving this goal is integrating both the accurate but not highly parallelized multiplier and the highly parallelized approximate multiplier on a chip and selectively using them depending on the target application and situation. However, since chip area is limited, integrating the two types of multipliers impairs the high area efficiency of the Mitchell multipliers and significantly reduces the degree of parallelism for the multiplication.

This paper first proposes a novel operand decomposition (OD) method which is more accurate than existing methods [9] [10]. Based on our OD method, this paper also proposes an accuracy reconfigurable vector accelerator for massive parallel multiplication and multiply-accumulate computation. The key idea of our proposal is a novel OD method combing with the Mitchell approximate multiplication [6]. Thanks to the proposed reconfigurable unit exploiting our OD mechanism, the area can be dramatically reduced from the accurate multiplier array while satisfying the required accuracy for various applications. The main contributions of our proposal include:

- Multiple accuracies can be reconfigured by different OD modes to suit various usage scenarios.
- The area is greatly reduced compared to the accurate multiplier, effectively improving the parallelism of operations with the same limited circuit area.
- The accuracy is significantly improved over the original logarithmic approximate multiplication.

We show that the proposed accuracy reconfigurable vector accelerator can be deployed in image smoothing and robot simultaneous localization and mapping (SLAM) applications with good performance. We can use the same accelerator by reconfiguring the accuracy to obtain different quality images as needed, or select different modes in robot localization and mapping to obtain the desired results, with high parallelism and small area overhead.

The rest of this paper is organized as follows. Section II reviews the conventional Mitchell logarithmic approximate multiplication and the original operand decomposition (OOD) approaches. Section III proposes the novel OD method, which is essential for the proposed reconfigurable vector accelerator, then gives the detail and advantages of the "accuracy reconfigurable" feature of the proposed vector accelerator. The experimental results regarding the accuracy improvement and area reduction are shown in Sect. IV. Section V describes the applicability of the proposed vector accelerator to image smoothing and robot SLAM applications. Lastly, concluding remarks are given in Sect. VI.

II. Related Work

This section summarizes the existing related works, including the original Mitchell algorithm and OOD methods, and analyzes their principles and shortcomings.

A. Mitchell Algorithm

In this section, we briefly describe the Mitchell algorithm based on logarithmic approximate multiplication.

The Mitchell algorithm only requires shift and addition operations [6]. First, we find the positions k_A and k_B, which are the positions of the leading "1" bit in the two inputs A and B, respectively. The position can be identified by shifting the input bits left until the most significant bit is a "1" with a leading "1" detector. The remaining bits after the leading "1" are used as mantissa m_A and m_B. Then, we combine k and m into floating-point format f as shown in following equations.

$$f_A = k_A.m_A, \tag{1}$$
$$f_B = k_B.m_B. \tag{2}$$

Next, we add the two approximate logarithmic values to get f_{AB}, which is the logarithm of $A \times B$.

$$\begin{aligned} Sum = f_{AB} &= f_A + f_B \\ &= k_A.m_A + k_B.m_B = k_{AB}.m_{AB}. \end{aligned} \tag{3}$$

Lastly, we decode f_{AB} to get the approximate result of $A \times B$. The decoding method concatenates f_{AB} after leading "1" and shifts left to make leading "1" to position k_{AB}.

Whereas the multiplier based on Mitchell's algorithm reduces the area compared to the accurate one, it suffers from the computational accuracy. The maximum error of the traditional Mitchell algorithm is 11.11%, which occurs when the mantissa part is 0.5. The average error is 3.83%. The approximate product is always less than the exact value, which may cause an accumulation and widening of the error.

In order to improve the accuracy of the traditional Mitchell's algorithm, some improved methods have been proposed. Babic et al. [11] proposed an iteration based logarithmic multiplier, reducing the maximum error of the approximate multiplication to less than 0.5% (3 correction circuits). However, the parallel implementation with one correction circuit doubles the area compared to the Mitchell multiplier. Ansari et al. [12] proposed to round operand to its nearest power of two instead of the highest power of two that is smaller than or equal to the operand. Both of their proposed multipliers improved the multiplication accuracy. Alla et al. [13] proposed a method that combines Mitchell's approximation with a hardware pruning technique, producing an area-efficient multiplier architecture without compromising accuracy. In addition, the approximate logarithmic multiplier with two-stage operand trimming proposed by Pilipovic et al. [14] further reduces the area and power consumption compared to the Mitchell multiplier, but as a cost, it has a larger error than the Mitchell multiplier.

B. Original Operand Decomposition (OOD)

Mahalingam et al. [9] proposed an operand decomposition method to improve the accuracy of Mitchell's logarithmic approximation.

Assume that the two n-bit inputs of the multiplication are X ($X = \{x_{n-1}x_{n-2}\dots x_2x_1x_0\}$) and Y ($Y = \{y_{n-1}y_{n-2}\dots y_2y_1y_0\}$). Decompose X and Y into four operands $A = \{a_{n-1}a_{n-2}\dots a_2a_1a_0\}$, $B = \{b_{n-1}b_{n-2}\dots b_2b_1b_0\}$, $C = \{c_{n-1}c_{n-2}\dots c_2c_1c_0\}$ and $D = \{d_{n-1}d_{n-2}\dots d_2d_1d_0\}$. Among them, $a_i = x_i \vee y_i$, $b_i = x_i \wedge y_i$, $c_i = \bar{x_i} \wedge y_i$, $d_i = x_i \wedge \bar{y_i}$. The final product is calculated by (4):

$$X \times Y = (C \times D) + (A \times B). \tag{4}$$

The advantage of this method is to reduce the number of "1"s in every decomposed operand. Therefore, the switching power dissipated for the multiplication operation can be reduced. Although the OOD method slightly reduces the average error of the Mitchell algorithm, the maximum error does not decrease.

Nandan et al. [10] proposed a variant of the OOD method with the main objective of improving the delay and power consumption of the circuit, but with limited improvement in accuracy.

The above solutions have solved the problem of excessively large area of the accurate multiplier, and some other methods offer a few degrees of improvement in accuracy. However, the above methods do not achieve the requirements of multiple accuracy reconfigurable, and cannot adapt to the application scenarios with high parallelism and various accuracies.

III. Proposed Approach

First, this section proposes the novel OD method. Next, we introduce the working mode implementation on the proposed vector accelerator, and then highlight the advantages of the accuracy reconfigurable vector accelerator.

A. Proposed Operand Decomposition

Fig. 1(a) shows a non-Decomposition configuration which calculates four multiplication in parallel using a 4-Mitchell multiplier array. Since there is no decomposition of operands and one multiplication remains as it is, we refer to this mode as OD-1. Similarly, the proposal to decompose a multiplication into two or four ways will be referred as OD-2 and OD-4 respectively, which are shown in Fig. 1(b) and Fig. 1(c), respectively. Let us then explain how the proposed OD method improves the accuracy of approximate multiplication by reconfiguring the original Mitchell based configuration shown in Fig. 1(a). This proposed OD method is suitable for realizing the accuracy reconfigurable feature highlighted in Sect. III.B.

First, we explain the proposed OD-2 method as illustrated in Fig. 1(b). For given two binary operand X_1 and Y_1, the OD-2 method finds the position k_{X_1} of the leading "1" of X_1, and fills k_{X_1} "0"s after "1". As a result, the first decomposed number of X_{11} can be obtained. The second number is the remaining part of X_1 (X_1'), which removes the leading "1" from X_1. The method calculates X_1' using (5).

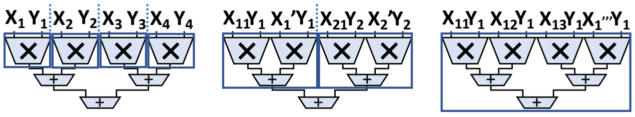
$$X_1' = \bar{X_{11}} \wedge X_1. \tag{5}$$

Then $X_1 \times Y_1$ can be calculated by (6).

$$X_1 \times Y_1 = (X_{11} + X_1') \times Y_1 = X_{11} \times Y_1 + X_1' \times Y_1. \tag{6}$$

Since it is obvious that X_{11} is the exponent of 2, the mantissa part of the approximate value of $\log_2 X_{11}$ is 0. When applying the Mitchell algorithm to $X_{11} \times Y_1$, only the integer part of X_{11} is added to the approximate value of $\log_2 Y_1$. Therefore, decoding the sum of the two logarithms is equivalent to shifting Y_1 to the left by k_{X_1} bits. This shifting left operation does not involve any errors. Meanwhile, since X_1' is a smaller number than X_1, $X_1' \times Y_1$ produces a smaller error when applying the Mitchell algorithm than applying it to $X_1 \times Y_1$ directly. Therefore, the proposed OD reduces the overall error for the multiplication.

Decomposing a smaller operand of X_1 and Y_1 can further improves the accuracy. By decomposing the smaller operand (suppose Y_1) into a power of 2 (Y_{11}) and a much smaller operand (Y_1'), we expect Y_1' to contain fewer "1"s, resulting in a smaller error in the approximate logarithm operation, so that the result of Mitchell approximate multiplication of Y_1' and X_1 will produce smaller errors that minimize the effect of the product error.



(a) OD-1 (b) OD-2 (c) OD-4

Fig. 1. OD-1/2/4 mode of the proposed vector accelerator.

The operand X_1 can also be decomposed into four operands (OD-4), which corresponds to Fig. 1(c). X_1' generated after the initial decomposition is decomposed into X_{12} and X_1'', then decompose X_1'' into X_{13} and X_1'''. Therefore, $X_1 \times Y_1$ can be obtained by (7).

$$\begin{aligned} X_1 \times Y_1 &= (X_{11} + X_{12} + X_{13} + X_1''') \times Y_1 \\ &= X_{11} \times Y_1 + X_{12} \times Y_1 + X_{13} \times Y_1 + X_1''' \times Y_1. \end{aligned} \tag{7}$$

$X_{11} \times Y_1$, $X_{12} \times Y_1$, and $X_{13} \times Y_1$ do not produce any errors when applying the Mitchell algorithm, and X_1''' occupies a much smaller proportion of the original operand X_1. Therefore, $X_1''' \times Y_1$ produces a much smaller error, which ultimately makes the overall result more accurate. Consequently,compared to the original Mitchell multiplication and the proposed OD-2 method, further reduces overall error. In Sect. IV, we experimentally show the accuracy improvement effect thanks to the proposed OD method. Another key point of the proposed OD method is that the most of the parts in Fig. 1(a) can be reused in the configuration shown in Fig. 1(b) and Fig. 1(c). In other words, just providing the decomposed operands to the configuration improves the approximation accuracy. The accuracy reconfigurable feature in Sect. III.B is achieved by switching between the above three modes.

B. Accuracy Reconfigurable Feature

The vector accelerator proposed in this paper satisfies both the high accuracy and high parallelism with only small area overhead, and supports on-demand accuracy reconfiguration. The key idea is illustrated in Fig. 2. The conventional accurate multiplier (black) has the highest accuracy but occupies the largest area. Namely, only a small amount of accurate multipliers can be integrated in the limited circuit area. In contrast, the original Mitchell approximate multiplier (red) greatly reduces the area and increases the parallelism of the multiplier array. However, its error sometimes have the difficulty coping with the need for high accuracy. Therefore, integrating one of the two parallel computing units in the circuit alone cannot meet the needs of both high parallelism and high accuracy. In addition, integrating both at the same time makes the area overhead more than worthwhile.

Compared with these integration strategies, our proposal (blue) has an advantage that, by making a trade-off between accuracy and parallelism, it can meet the usage requirements of various accuracies, keeping high parallelism with only small area overhead. The key idea behind of this accuracy reconfiguration is the OD proposed in the Sect. III.A, which improves the accuracy by decomposing one multiplication into two or four ways. More specifically, in the OD-1 mode (“1” in Fig. 2, which corresponds to Fig. 1(a)), the accuracy is identical with that of the original Mitchell multiplier, but with a slight increase in the occupied area due to the decomposition circuit. OD-2 mode (“2” in Fig. 2, which corresponds to Fig. 1(b)) makes one multiplication to use two approximate multipliers, and the accuracy can be greatly improved. OD-4 mode (“4” in Fig. 2, which corrsponds to Fig. 1(c)) gives an accuracy comparable to that of the accurate multiplier, but at the expense of the most parallelism. Also, by integrating the proposed vector accelerator into the processor in the form of an extended instruction set, the accuracy can be reconfigured at the software level, making the proposed vector accelerator generalizable to any usage scenario.

IV. Evaluation of the Vector Accelerator

In this section, we experimentally evaluate the accuracy, area, and critical path delay of the proposed vector accelerator, and compare with the conventional Mitchell multiplier and OOD implementation. First, we evaluate the accuracy of the proposed vector accelerator. We use our proposal based on OD-2, OD-4, the OOD method in [9], and the original Mitchell method (which is the same in principle as OD-1) in [6]. Using the above multiplication, we implement the multiplier whose input is 8-bit, 16-bit, and 32-bit unsigned integers, respectively.

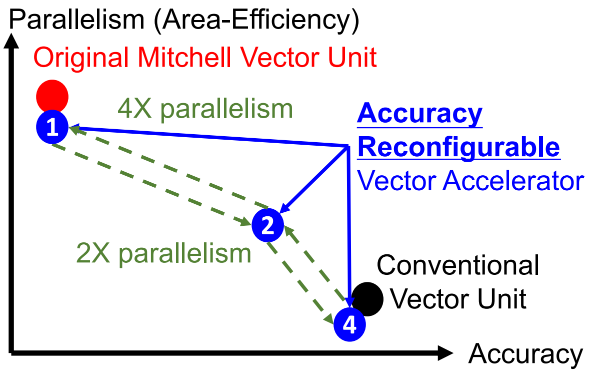


Fig. 2. The trade-off between accuracy and parallelism.

We use mean relative error distance (MRED) [15] as a criteria for the accuracy evaluation:

$$MRED = \frac{1}{N} \sum_{i=1}^{N} \frac{|P_i - P_i'|}{P_i}, \tag{8}$$

where N is the number of test cases of the multiplication, i is the index of each test case, P_i is the accurate product of the i-th test case, and P_i' is the approximate product of the i-th test case.

Table I shows the evaluation results of the maximum error and MRED in various situations. Compared with the original Mitchell multiplier (OD-1), the maximum errors of the proposed OD method are only 4.81% (OD-2) and 1.10% (OD-4), and the MRED are only 1.05% (OD-2) and 0.10% (OD-4), which has been significantly improved up to a factor of 30 compared to the MRED of Mitchell multiplier.

Then, we compare the circuit area and critical path delay. We use a commercial logic synthesis tool to obtain the area and timing reports for the proposed accuracy reconfigurable vector accelerator. At the same time, we synthesize the vector

TABLE I
Accuracy of 8-bit, 16-bit and 32-bit integer multiplication

No. of bits	Method	Max Error (%)	MRED (%)
8	**Mitchell (OD-1)** [6]	11.11	3.67
	OOD [9]	11.11	1.88
	Proposed OD-2	4.81	0.89
	Proposed OD-4	1.08	0.03
16	**Mitchell (OD-1)** [6]	11.11	3.84
	OOD [9]	11.11	2.17
	Proposed OD-2	4.81	1.05
	Proposed OD-4	1.10	0.10
32	**Mitchell (OD-1)** [6]	11.11	3.84
	OOD [9]	11.11	2.18
	Proposed OD-2	4.81	1.05
	Proposed OD-4	1.10	0.10

TABLE II
AREA AND CRITICAL PATH DELAY OF ACCURATE, MITCHELL, OOD UNIT AND PROPOSED VECTOR ACCELERATOR

No. of bits	**32-bit**	
	Area (μm^2)	**Delay** (ns)
Accurate	117,768.85	1,438.19
Mitchell [6]	36,581.71	1,567.51
OOD [9]	36,581.71	1,567.51
Proposed	55,071.58	2,190.28
Proposed[1]	50,504.28	1,942.02
Proposed[2]	40,236.36	1,934.48

[1] Without operand comparison

[2] Without operand comparison, OD-1 and OD-2 only

unit using eight accurate multipliers and the vector unit with eight Mitchell multipliers only. We also synthesize a vector unit with eight Mitchell multipliers using the OOD method in [9]. We use a commercial standard cell library based on 55 nm CMOS process technology. We evaluate and compare the area and delay of the above synthesized circuits.

Table II shows the comparison results. The area of the Mitchell vector unit is only 31.1% of the accurate vector unit. The OOD unit using bit manipulation has no change in area and delay compared to the Mitchell unit, but the degree of parallelism is fixed by half. The area of the proposed accuracy reconfigurable vector accelerator is 55,071.58 μm^2, which is slightly increased compared to the Mitchell unit, but still only 46.76% of the accurate unit, achieving a significant saving. If we ignore the accuracy gain from decomposing smaller oprands, the area is 50,504.28 μm^2, which is only 42.88% of the accurate unit. If we further simplify the accelerator to which with only two reconfigurable accuracy-parallel modes (e.g. OD-1 and OD-2 mode), the area is even reduced to 34.17% of the accurate unit. With this area saving, we can accelerate the vector multiplication or multiply-accumulate operation by integrating higher parallel vector unit on a chip. On the other hand, due to the decomposition circuit, the critical path delay has increased by up to 52.29% from the accurate unit, which is one of our primary future works.

V. APPLICATION CASE STUDIES

In this section, we discuss the applicability of the proposed accuracy reconfigurable vector accelerator in the image smoothing and Simultaneous Localization and Mapping (SLAM). In these applications, we compare the performance of the proposed accuracy reconfigurable vector accelerator in OD-2 mode, OD-4 mode, and OD-1 mode (which is parallel Mitchell multipliers) with the parallel accurate multipliers.

A. Image Smoothing

Image smoothing [16] is an application that reduces the noise from an image by convolving the image using a smoothing kernel. The smoothing kernel is a square matrix with each value conforming to a Gaussian distribution:

$$H(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \tag{9}$$

where $H(x,y)$ represents the value of the x-th row and y-th column in the smoothing kernel, and σ^2 is variance of the kernel. By multiplying and accumulating the value of each pixel of the target image and its surrounding pixels with the smoothing kernel, and then averaging, we can obtain the value of the pixel after the smoothing process.

TABLE III
SSIM AND PSNR IN THE IMAGE SMOOTHING

	Color		**Grayscale**	
	SSIM	**PSNR**	**SSIM**	**PSNR**
OD-1	0.973	20.603	0.990	26.599
OD-2	0.984	22.919	0.995	29.132
OD-4	0.986	23.628	0.996	29.887

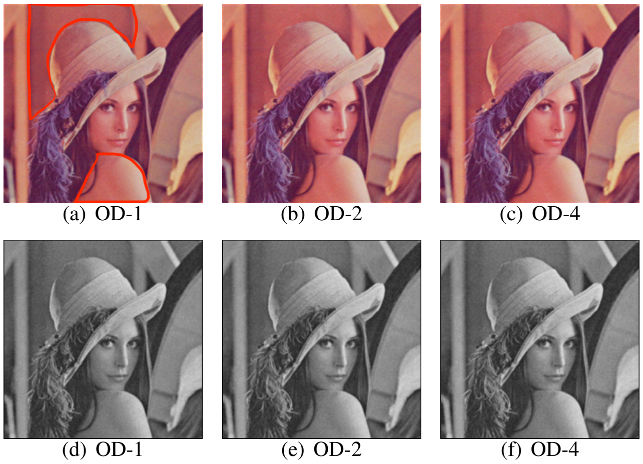

(a) OD-1 (b) OD-2 (c) OD-4
(d) OD-1 (e) OD-2 (f) OD-4

Fig. 3. Smoothed color and grayscale Lenna images. The color image using OD-1 has dull colors and severe distortion (Especially in the red line box area), while images smoothed using the proposed OD-2 or OD-4 mode are brighter. In the case of grayscale image, the difference is not significant.

We quantify the floating-point smoothing kernel to the unsigned fixed-point number with 8 fractional bits as $\lfloor H \cdot 2^8 \rfloor / 2^8$, where H is the floating-point value of smoothing kernel. As image quality metrics, we select the Structural Similarity Index (SSIM) [17] and the Peak-Signal-to-Noise Ratio (PSNR). We evaluate the SSIM and PSNR between the images smoothed with the accuracy reconfigurable vector unit and accurate parallel multipliers as the criteria for the evaluation. As test images, we use the color and grayscale Lenna images of 512×512 pixels with random noise.

Table III shows the results of image smoothing. Compared to the image smoothed using OD-1 mode, the PSNR of the image using OD-2 mode improved by 11.2% (color) and 9.5% (grayscale). The OD-4 mode further improves the accuracy, e.g. the PSNR improved by 14.7% (color) and 12.4% (grayscale) compared to OD-1 mode. Fig. 3 visualizes the smoothed Lenna image. In the case of color image smoothing, the tone of the low PSNR and low SSIM image smoothed by parallel Mitchell multipliers with the lowest accuracy is darker, especially the red line box area in Fig. 3(a). Meanwhile, the images smoothed using the OD-2 or OD-4 mode with improved accuracy are brighter and closer to the original color of the image. Therefore, it is recommended to use OD-2 or OD-4 for better results in cases where high accuracy is required. On the other hand, with grayscale images, there is no significant difference between three modes, and hence, the OD-1 mode can be selected. In this case study, we can reconfigure the accuracy as needed to obtain good smoothing results for color and grayscale images, respectively, while ensuring high parallelism in the computation.

B. Simultaneous Localization and Mapping

Simultaneous Localization and Mapping is the computational problem of constructing or updating a map of an unknown environment while simultaneously estimating an robot's pose (position, orientation, etc.) within it. The researches of [18] and [19] have high expectation to the possibility and prospect of applying approximate calculation in the field of SLAM. We use a laser-based SLAM application named LittleSLAM proposed by Tomono [20] to evaluate the performance of the proposed accuracy reconfigurable vector accelerator. The core approach to robot localization in LittleSLAM is the Iterative Closest Point method based on the scan matching. It estimates the rigid-body transformation of the robot by matching the scans of the environmental landmarks by the robot's mounted sensors. In this process, the point cloud in the current scan frame needs to be matched to the reference frame. After that, gradient descent method is used to find the robot's pose that minimizes the cost function. These two processes are repeated iteratively until the decrease of the cost function is less than a threshold, then the gradient descent method is considered to converge, i.e., the most probable pose of the robot is found. Finally, all points in the current scan frame are multiplied with the coordinate transformation matrix to be added to the global map.

The SLAM processing involves a vast number of multiplications in many part. However, we anticipate that some parts are resilient to errors introduced by proposed accuracy reconfigurable vector accelerator. We deploy accuracy reconfigurable vector accelerator on two parts of the SLAM program to evaluate the performance of the proposed vector accelerator in simultaneous localization and mapping, respectively: (1) The square operation in the cost function of simultaneous localization:

$$G(tx, ty, th) = \sum_{i=0}^{N} [(x_{ci} - x_{ri})^2 + (y_{ci} - y_{ri})^2], \qquad (10)$$

where tx and ty represent the robot's position and th represents the robot's orientation. (x_{ci}, y_{ci}) and (x_{ri}, y_{ri}) are the coordinate of i-th point pair in the current scan frame and reference scan frame, respectively. N is the total number of associated point pair. (2) Matrix multiplication in the coordinate transformation step of map generation:

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} \cos(th) & -\sin(th) \\ \sin(th) & \cos(th) \end{pmatrix} \begin{pmatrix} x_{li} \\ y_{li} \end{pmatrix} + \begin{pmatrix} tx \\ ty \end{pmatrix}, \qquad (11)$$

where (x_i, y_i) is the coordinate of the i-th point in generated map under global coordinate system. (tx, ty, th) is the estimated robot's pose. (x_{li}, y_{li}) is the coordinate of the i-th point under robot coordinate system. We implement the accurate reconfigurable vector accelerator in C++ language to replace the accurate multipliers. We use the two datasets, i.e., corridor and hall, that come with the program as test inputs.

Table IV shows the maximum and mean distances between the robot coordinates estimated by performing the square operation in the cost function using the proposed accuracy reconfigurable vector accelerator and the accurate multipliers. The choice of accuracy mode has no significant effect on the convergence position of the gradient descent method using iterations. In general, using the OD-1 mode of the accuracy reconfigurable vector accelerator yields an estimated pose trajectory that is quite close to that of using the accurate multipliers. In the hall dataset scenario, the maximum error of 19.27 cm is only 0.43% of the total walking distance of 44.64 m, while in the corridor dataset, maximum error (30.36 cm) is only 0.46% of the total walking distance of 66.05 m. Fig. 4 shows the estimated robot pose trajectory under the hall dataset. The orientation of the robot is omitted. The dots in the figure represent the estimated robot pose corresponding to each scan frame. The zoomed area shows the location where the maximum error occurs. The trajectory using OD-1 and that using the accurate multipliers in the Fig. 4 overlap to a high degree, so that sufficient accuracy can be obtained by using OD-1 mode in the simultaneous localization part of SLAM.

TABLE IV
MAXIMUM AND MEAN DIFFERENCE IN ROBOT LOCALIZATION

	hall		**corridor**	
	Max (cm)	**Mean** (cm)	**Max** (cm)	**Mean** (cm)
OD-1	19.27	10.74	30.36	18.19
OD-2	13.40	9.51	15.72	5.74
OD-4	21.97	14.04	11.92	4.49

TABLE V
MAXIMUM AND MEAN DIFFERENCE IN MAPPING

	hall		**corridor**	
	Max (cm)	**Mean** (cm)	**Max** (cm)	**Mean** (cm)
OD-1	59.79	11.27	46.42	1.83
OD-2	27.50	3.25	25.12	1.00
OD-4	5.77	0.29	6.32	0.52

Table V shows the maximum and mean distances between the map point generated by performing the coordinate transformation matrix multiplication using the proposed accuracy reconfigurable vector accelerator and the accurate multipliers. It can be seen that with the hall dataset, the map error using the OD-1 mode is up to more than 50 cm. The maps generated using the OD-1 mode produce a larger offset compared to the maps generated using the accurate multiplier. Meanwhile, as can be seen in the generated map of the hall dataset shown in Fig. 5, the map generated using the OD-1 is more heavily smearing, i.e., the representation of obstacles in the environment is not fine enough. The red part is where the error exceeds 50 cm compared to the map generated using the accurate multiplier array. The maximum errors (59.79 cm) occur at the two larger red dots in the lower right corner. The proposed OD-2 mode reduces the mean error to 28.8% of the OD-1 mode, making the generated maps much closer to those generated using the accurate multipliers. In the case of higher accuracy requirements, the OD-4 mode even reduces the mean error to within 1 cm. Therefore, in order to build finer maps without losing high parallelism, the OD-2 or OD-4 mode of the proposed vector accelerator is an good choice.

In this case study, if the accurate multiplier array is used, it is limited by the circuit area to achieve high parallelism calculation, which will inevitably increase the number of execution cycles and require significant amount of energy. Meanwhile, if only Mitchell multiplier array is used, it is not possible to obtain high accuracy localization and mapping results simultaneously when performing high parallelism operations. By applying the proposed vector accelerator in different parts of this SLAM application and configuring the different accuracies, it is possible to obtain high accuracy results for both robot localization and mapping at the same time, and to maintain high parallelism.

VI. CONCLUSIONS

This paper proposed a novel operand decomposition method which widely improves the accuracy of existing OD methods for multiplication. One of the inputs can be optionally decomposed into multiple operands, and the final product is ex-

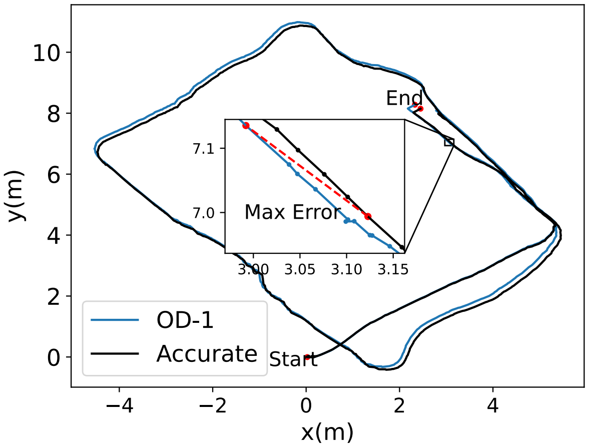


Fig. 4. Estimated robot's location trajectory (44.64 m) of hall dataset. The maximum error using OD-1 mode (19.27 cm) is only 0.43% of the total walking distance, which is small enough for localization.

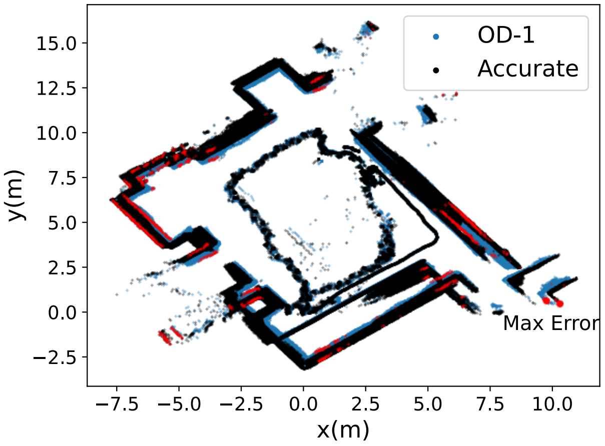


Fig. 5. Generated map of hall dataset. The red part indicates that the error exceeds 50 cm. The representation of obstacles in the map generated using the OD-1 mode is not fine enough, so the higher-accuracy mode needs to be configured to obtain finer map.

pressed as the sum of the product of the decomposed number and another input. Based on the proposed OD method, we also proposed the accuracy reconfigurable vector accelerator. Multiple accuracy can be obtained by different OD modes to suit various usage scenarios, while remaining high parallelism. Through the hardware evaluation, the area of the proposed accuracy reconfigurable vector accelerator can be significantly reduced to only 34.17% of the traditional accurate multiplier array.

We have successfully deployed the proposed accuracy reconfigurable vector accelerator in image smoothing and simultaneous localization and mapping applications. We experimentally confirmed that the proposed vector accelerator behaved well. In image smoothing applications, we can reconfigure the accuracy as needed to obtain a slightly lower quality image (OD-1 mode) or sacrifice a certain degree of parallelism to obtain a high quality image (OD-2 or OD-4). Meanwhile, in SLAM application, applying OD-1 mode of the proposed vector accelerator in simultaneous localization can achieve acceptable localization accuracy. In the case of mapping, the OD-2 or OD-4 mode of the proposed vector accelerator can be utilized to obtain a better quality map. Our proposal addresses both the low parallelism of accurate multipliers and the lack of accuracy of original logarithmic approximate multipliers, and achieves on-demand accuracy and high parallelism computation with only small area overhead.

References

[1] W. Liu, F. Lombardi, and M. Shulte, "A retrospective and prospective view of approximate computing," In *Proceedings of the IEEE*, vol. 108, no. 3, pp. 394-399, 2020.

[2] R. Zendegani, M. Kamal, M. Bahadori, A. Afzali-Kusha, and M. Pedram, "RoBA multiplier: A rounding-based approximate multi-plier for high-speed yet energy-efficient digital signal processing," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 2, pp. 393-401, 2017.

[3] H. Jiang, C. Liu, F. Lombardi, and J. Han, "Low-power approximate unsigned multipliers with configurable error recovery," *IEEE Trans.Circuits Syst. I, Reg. Papers*, vol. 66, no. 1, pp. 189-202, 2019.

[4] T. Yang, T. Ukezono, and T. Sato, "Low-power and high-speed approximate multiplier design with a tree compressor," In *Proceedings of the IEEE International Conference on Computer Design*, pp. 89-96, 2017.

[5] D. Esposito, A. G. M. Strollo, E. Napoli, D. Caro, and N. Petra, "Approximate multipliers based on new approximate compressors," *IEEE Trans. on Circuits and Systems I: Regular Papers*, vol. 65, no. 12, pp.4169-4182, 2018.

[6] J. N. Mitchell, "Computer multiplication and division using binary logarithms," *IRE Transactions on Electronic Computers* vol. EC-11, no. 4, pp. 512-517, 1962.

[7] M. S. Kim, A. A. D. Barrio, R. Hermida, and N. Bagherzadeh, "Low-power implementation of Mitchell's approximate logarithmic multiplication for convolutional neural networks," In *Proceedings of the 23rd Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 617-622, 2018.

[8] M. S. Kim, A. A. D. Barrio, L. T. Oliveira, R. Hermida, and N. Bagherzadeh, "Efficient Mitchell's approximate log multipliers for convolutional neural networks," *IEEE Transactions on Computers*, vol. 68, no. 5, pp. 660-675, 2019.

[9] V. Mahalingam and N. Ranganathan, "Improving accuracy in Mitchell's logarithmic multiplication using operand decomposition," *IEEE Transactions on Computers*, vol. 55, no. 12, pp. 1523-1535, 2006.

[10] D. Nandan, J. Kanungo, and A. Mahajan, "An efficient VLSI architecture design for logarithmic multiplication by using the improved operand decomposition," *Integration, the VLSI Journal*, vol. 58, pp. 134-141, 2017.

[11] Z. Babic, A. Avramović, and P. Bulić, "An iterative logarithmic multiplier," *Microprocessors and Microsystems*, vol. 35, no. 1, pp. 23-33, 2011.

[12] M. S. Ansari, B. F. Cockburn, and J. Han, "A hardware-efficient logarithmic multiplier with improved accuracy," In *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 928-931, 2019.

[13] N. Alla and S. E. Ahmed, "An area and delay efficient logarithmic multiplier," In *Proceedings of the International Conference on Contemporary Computing and Applications (IC3A)*, pp. 169-174, 2020.

[14] R. Pilipovic, P. Bulic, and U. Lotric, "A two-stage operand trimming approximate logarithmic multiplier," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 6, pp. 2535-2545, 2021.

[15] C. Liu, J. Han, and F. Lombardi, "A low-power, high-performance approximate multiplier with configurable partial error recovery," In *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1-4, 2014.

[16] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. Saddle River, NJ, USA: Prentice-Hall, 2006.

[17] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Imagequality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process*, vol. 13, no. 4, pp. 600-612, 2004.

[18] A. Agrawal et al., "Approximate computing: Challenges and opportunities," In *Proceedings of 2016 IEEE International Conference on Rebooting Computing (ICRC)*, pp. 1-8, 2016.

[19] J. Oh, J. Choi, G. C. Januario, and K. Gopalakrishnan, "Energy-efficient simultaneous localization and mapping via compounded approximate computing," In *Proceedings of 2016 IEEE International Workshop on Signal Processing Systems (SiPS)*, pp. 51-56, 2016.

[20] M. Tomono, "A scan matching method using Euclidean invariant signature for global localization and map building," In *Proceedings of IEEE International Conference on Robotics and Automation*, Vol. 1, pp. 866-871, 2004.