Hardware Architecture for CORDIC with Improved Rotation Strategy

Yu Zhang¹, Yun-Ting Liao², Kui-Ting Chen³ and Takaaki Baba⁴

¹,²,⁴Graduate School of Information, Production and Systems, Waseda University
³Research Center of Information, Production and Systems, Waseda University

2-7 Hibikino, Wakamatsu-ku, Kitakyushu-shi, Fukuoka 808-0135, Japan

Phone: +81-93-692-5017   FAX: +81-93-692-5021

E-mail: ¹zymiaow@toki.waseda.jp, ²jesse_liao@fuji.waseda.jp, ³nore@aoni.waseda.jp, ⁴tbaba@waseda.jp

Abstract

For many mathematical applications, the conventional coordinate rotation digital computer (CORDIC) algorithm can achieve high efficiency. However, the iterative procedure of the conventional CORDIC algorithm is inefficient owing to its rotation strategy. Chen et al. proposed a CORDIC algorithm with an improved rotation strategy to reduce the number of unnecessary iterations of the CORDIC algorithm. As a result, the calculation speed can be improved to four times than that of the conventional CORDIC hardware when the improved rotation strategy can finish its own functions within one clock cycle. However, the complexity of the improved rotation strategy is greatly increased, making it difficult to finish each rotation within one clock cycle. To overcome this difficulty, a hardware architecture for CORDIC with an improved rotation strategy is proposed to ensure that the improved rotation strategy can be finished within one clock cycle.

1. Introduction

The conventional coordinate rotation digital computer (CORDIC), originally proposed by Volder in 1959 [1] and later generalized by Walther [2], is a well-known and efficient iterative algorithm. The CORDIC algorithm is widely used in hardware implementation because of its advantage that it only exploits the operations of addition and bit-shifting to perform various mathematical functions. The general conventional CORDIC hardware architectures are the state machine structure and the pipelined structure. Conventional CORDIC architectures are easily implemented and can work under a high frequency. As a trade-off, there are too many iterative procedures to achieve a high calculation speed for modern complex real-time applications. In such a situation, the total number of iterations must be reduced. To overcome this difficulty, a CORDIC algorithm with an improved rotation strategy has been introduced [3]. The improved CORDIC divides the conventional rotation approach into two rotation functions to reduce the number of unnecessary iterations of the CORDIC algorithm, as shown in Fig.1. The coarse rotation function can approximate the target angle within one iteration. In the precise rotation function, a greedy algorithm is adopted to reduce the number of iterations. However, hardware implementation of the CORDIC algorithm with the improved rotation strategy is extremely complex since it is difficult to finish each rotation within one clock cycle.

To overcome the above difficulties, a new CORDIC hardware architecture with an improved rotation strategy is proposed. The proposed structure aims to reduce the complexity of the CORDIC rotation strategy and ensures that the improved rotation strategy can be finished within one clock cycle.

The rest of this paper is organized as follows. In Sect.2, the concept of the CORDIC algorithm, the conventional CORDIC architecture and the CORDIC algorithm with the improved rotation strategy are introduced. In Sect.3, the details of the proposed CORDIC hardware with the improved rotation strategy will be introduced. In Sect.4, the results of simulations comparing the pro-
posed and existing architectures are provided. Finally, Sect.5 comprises a summary and the conclusions of this research.

2. Related Works

2.1 CORDIC algorithm

The CORDIC algorithm is an iterative algorithm for computing some basic operation functions and arithmetic operations. The basic idea of the CORDIC algorithm is to approximate the desired rotation angle by continuous rotation using a series of operation-related angles. The equations of the unified CORDIC algorithm are given by

\[ X_{i+1} = X_i - m \sigma_i 2^{-i} Y_i \]  \hspace{1cm} (1)
\[ Y_{i+1} = Y_i - \sigma_i 2^{-i} X_i \]  \hspace{1cm} (2)
\[ Z_{i+1} = Z_i - \sigma_i \alpha_i \]  \hspace{1cm} (3)
\[ K = \prod_{i=0}^{n} \cos(\alpha_i), K^i = \prod_{i=1}^{n} \cosh(\alpha_i) \]  \hspace{1cm} (4)

where \( m \in \{1, -1, 0\} \) depending on the type of coordinates: \( m=1 \) for circular coordinates, \( m=0 \) for linear coordinates and \( m=-1 \) for hyperbolic coordinates. \( \sigma_i \) indicates the direction of rotation in each iteration. \( \sigma_i = 1 \) and \( \sigma_i = -1 \), which are decided by the situation of iteration \( i \).

In Eq. (4), \( K \) is the scale factor, which is applied at the start or end of the iterations to normalize the result. \( n \) is the total number of iterations.

2.2 Conventional hardware of CORDIC

Andraka presented a survey of the CORDIC algorithm based on field-programmable gate array (FPGA) implementation [4]. Two basic structures are used to implement a conventional CORDIC hardware architecture. A more compact state machine is easy to realize while a high-speed, fully pipelined processor meets the requirements for a high speed. For the state machine structure and pipeline structure, the iterators are the same. The conventional CORDIC iterator is composed of three full adders/subtractors, two shifters, and an elementary angle list of values of \( \tan^{-1} 2^{-i} \), as illustrated in Fig.2. The sign bit of \( Z_i \) is used for determining the value of \( \sigma_i \) in Eqs. (1) to (3) and controlling the direction of the CORDIC rotation. However, in such a design, one iterator is simply duplicated from the basic equations. It only operates additions or bit-shifting in each clock cycle, which means that each clock cycle includes redundancy. In addition, the conventional architecture is only translates the basic equations without considering a quick approach for finding rotation angles.

2.3 CORDIC with improved rotation strategy

Chen et al. [3] improved the CORDIC algorithm by adopting a greedy algorithm. The simulation results proved that the CORDIC algorithm can provide a high calculation ability by greatly reducing the total number of iterations. The CORDIC algorithm with the improved rotation strategy divides the conventional rotation function into a coarse rotation function and a precise rotation function. The proposed coarse rotation function employs a central angle list to approximate the target angle after one iteration. In the precise rotation function, the greedy algorithm is terminated when the deviation between the target angle and the rotated angle is less than a specified threshold.

However, the hardware architecture for CORDIC with the improved rotation strategy is extremely complex if it is directly implemented. A central angle list is required for the iterative procedure, which means that the memory cost is larger than that in the conventional approaches. Concerning the proposed precise rotation function, each elementary angle has to be sequentially inspected by the greedy algorithm to select the appropriate rotation angle. In such a situation, many comparisons must be performed in each clock cycle both for the greedy algorithm and for determining the deviation. As a result, it is almost impossible to finish the precise rotation function within one clock cycle. In other words, this kind of hardware pipeline structure must encounter data hazards, which means that it is inapplicable.

3. Proposed CORDIC Hardware

To overcome the above difficulties, a CORDIC hardware architecture with an improved rotation strategy is proposed. The proposed hardware consists of two main architectures, which are a coarse rotation block (CRB) and a precise rotation block (PRB) as shown in Fig.3. It is a new way of performing the conventional iterative rotation and provides a quicker approach to finding rotation angles. These two blocks can greatly reduce the number of iterations.
Furthermore, to achieve a high calculation performance, the proposed hardware adopts an eight-stage pipeline to increase the calculation speed of the hardware, as illustrated in Fig. 4. For the purpose of pipeline execution without encountering data hazards, the key point of the proposed hardware is to ensure that the calculations of the coarse rotation and precise rotation can both be finished within one clock cycle.

### 3.1 Coarse rotation block

To obtain a quicker way of finding rotation angles, the elementary angle list of rotations, which comprises values of $\arctan(2^{-i})$ or $\text{arctanh}(2^{-i})$, must be changed. The proposed coarse rotation block employs a central angle list, which is a new look-up table, to approximate the target angle within one clock cycle. This can reduce the number of clock cycles by about 2-3 compared with the conventional rotation strategy.

Figure 5 shows the Verilog implementation architecture of the proposed coarse rotation block. This architecture consists of a memory with a central angle list, which contains a new rotation angle list. The proposed list divides the angles from 0 to 90° into 13 sections. Once an appropriate angle section containing the target angle is chosen, the target angle is rotated to the central angle of the section. The selected central angle is the output of the coarse rotation block, which is also the input of the precise rotation block. On the basis of the simulation results of [3], the proposed architecture employs a 13 to 1 multiplexer so as to complete the above operation of the coarse rotation block. On the basis of the simulation results of [3], the proposed architecture employs a 13 to 1 multiplexer so as to complete the above operation of the coarse rotation block. When the central angle list is reformed or increased, the proposed architecture can still finish the coarse rotation function within one clock cycle by adjusting the size of the multiplexer. Therefore, this proposed coarse rotation block is generic for CORDIC hardware architecture.

### 3.2 Precise rotation block

The precise rotation block exploits a greedy algorithm to perform a micro-rotation to minimize the deviation between the target angle and the rotation angles. The algorithm terminates while the deviation cannot be minimalized. The greedy algorithm chooses the best rotation angle at each step. It incorporates deviation judgement given by the following equation:

$$\text{deviation} = \min |d(\theta) - \theta^i|$$

where $d(\theta)$ is the target value of the rotation function to be rotated at each step and $\theta^i$ is the rotation angle from the elementary rotation angle list. This approach can reduce the rotation time.

To finish the greedy algorithm within one clock cycle, a parallel greedy structure is proposed that applies the characteristic of the conventional CORDIC elementary angle list, as illustrated in Fig. 6. The proposed architecture mainly includes an elementary angle list, a multiplexer, a comparator and a deviation judgement block. The elementary angle list consists of values of $\arctan(2^i)$. For the part involving the elementary angle list, the greedy algorithm in the proposed precise rotation block does not need to check all bit values of each elementary angle. Through observing the location of the first high bit for all elementary angles, the operation of the greedy algorithm can be rapidly finished without performing subtractions between every rotation. A multiplexer is used to decide the interval of the rotation angle. The comparator decides which element angle needs to be rotated. Then the deviation judgement block controls the total rotation time of the precise rotation block. The proposed architecture can successfully finish all the precise rotation function within one clock cycle. In addition,
can greatly improve the calculation performance by efficiently using the waiting time of a clock cycle.

4. Simulation Results

The proposed architecture is directly implemented on a Cyclone IV GX EP4CGX22CF19C6 development board with Verilog hardware description language in Quartus II 9.0. This paper applies CORDIC in rotation mode under circular coordinates. Therefore, the results of sine, cosine and tangent functions are given in the simulation results. All the experiments are performed under a 50 MHz system clock (1 clock cycle is 20 ns). The experimental parameters of the simulation are shown in Table 1. The total number of angle sections is 13. The angles of the side section and internal section are 3.75° and 7.5°, respectively. All parameters of the CORDIC algorithm are set to the same condition for each elementary angle number. The target angle is changed from 0.0 to 90.0° in increments of 0.1°.

The performance is compared with that of existing architectures [5], [6]. In the Table 2, the proposed CORDIC hardware just takes 7.277 ns to finish the precise rotation block. This architecture reduces the maximum number of iterations from 32 to 7 time, the improvement ratio is 78.1%. The mean iteration time is reduced by 84.1%. The experimental results demonstrate that the average calculation speed is 400% faster than that of the existing architecture [5], [6]. The maximum clock rate of the proposed architecture is 137.42 MHz.

5. Conclusions

We proposed a hardware architecture for the CORDIC algorithm with an improved rotation strategy based on FPGA technology. In contrast to the conventional method, the proposed architecture employs a coarse rotation block and a precise rotation block to ensure that the improved rotation strategy can be finished within one clock cycle. The simulation results prove that the proposed architecture just takes 5.381 ns to finish the coarse rotation function and 7.277 ns to finish the precise rotation function under a 50 MHz system clock. Overall, the proposed architecture can complete the procedures of the improved rotation strategy within one clock, and the speed of the proposed structure is increased by 400% compared with the conventional hardware architecture.

Acknowledgments

This work was supported by a Japan Society for the Promotion of Science (JSPS) Grant-in-Aid for Scientific Research (No. 26280017) and a Grant-in-Aid for Young Scientists (No. 15K21435).

Table 1: Summary of simulation parameters

| Parameters                     | Values |
|--------------------------------|--------|
| System clock                  | 50MHz  |
| Number of elementary angles   | 32     |
| Number of side sections       | 2      |
| Number of internal sections   | 11     |
| Angle of side section         | 3.75°  |
| Angle of internal section     | 7.5°   |
| Interval of target angle      | [0.0°, 90.0°] |

Table 2: Comparison of simulation results

|                      | Existing Architectures [5], [6] | Proposed Architecture | Improvement Ratio |
|----------------------|--------------------------------|-----------------------|-------------------|
| Coarse rotation      | —                              | 5.381 ns              | —                 |
| Precise rotation     | —                              | 7.277 ns              | —                 |
| Max. iteration       | 32                             | 7                     | 78.1%             |
| Mean iteration       | 32                             | 5.09                  | 84.1%             |
| Avg.                 | 640 ns                         | 160 ns                | 400%              |

References

[1] J. Volder: The CORDIC trigonometric computing technique, IRE Transactions on Electronic Computing, Vol. EC-8, pp. 330-334, 1959.
[2] J. S. Walther: A unified algorithm for elementary functions, Proc. Spring. Joint Computer Conf., pp. 379-385, 1971.
[3] K.-T. Chen, K. Fan, X. Han and T. Baba: A CORDIC algorithm with improved rotation strategy for embedded applications, J. of Industrial and Intelligent Information, Vol. 3, No. 4, pp. 274-279, 2015.
[4] R. Andraka: A survey of CORDIC algorithm for FPGA based computers, Proc. 1998 ACM/SIGDA 6th Int. Symp. on Field Programmable Gate Arrays, pp. 191-200, 1998.
[5] M. Ristovic, S. Lubura and D. Jokic: Implementation of CORDIC algorithm on FPGA altera cyclone, 20th Telecommunications Forum, pp. 875-878, 2012.
[6] K.-T. Chen, M.-L. Jia, X. Wu and T. Baba: A programmable architecture of CORDIC for accelerating embedded processor, Journal of Signal Processing, Vol. 17, No. 6, pp. 247-254, 2013.