2022

Efficient and Accurate CORDIC Pipelined Architecture Chip Design Based on Binomial Approximation for Biped Robot

Rih-Lung Chung

Yen Hsueh

Shih-Lun Chen

Patricia Angela R. Abu

electronics

MDPI

Article

Efficient and Accurate CORDIC Pipelined Architecture Chip Design Based on Binomial Approximation for Biped Robot

Rih-Lung Chung [1,*], Yen Hsueh [1], Shih-Lun Chen [1] and Patricia Angela R. Abu [2]

[1] Department of Electronic Engineering, Chung Yuan Christian University, Chung Li City 320, Taiwan; lintn222@gmail.com (Y.H.); chrischen@cycu.edu.tw (S.-L.C.)
[2] Department of Information Systems and Computer Science, Ateneo de Manila University, Quezon City 1108, Philippines; pabu@ateneo.edu
* Correspondence: rlchung@cycu.edu.tw; Tel.: +886-3-265-4605

Abstract: Recently, much research has focused on the design of biped robots with stable and smooth walking ability, identical to human beings, and thus, in the coming years, biped robots will accomplish rescue or exploration tasks in challenging environments. To achieve this goal, one of the important problems is to design a chip for real-time calculation of moving length and rotation angle of the biped robot. This paper presents an efficient and accurate coordinate rotation digital computer (CORDIC)-based efficient chip design to calculate the moving length and rotation angle for each step of the biped robot. In a previous work, the hardware cost of the accurate CORDIC-based algorithm of biped robots was primarily limited by the scale-factor architecture. To solve this problem, a binomial approximation was carefully employed for computing the scale-factor. In doing so, the CORDIC-based architecture can achieve similar accuracy but with fewer iterations, thus reducing hardware cost. Hence, incorporating CORDIC-based architecture with binomial approximation, pipelined architecture, and hardware sharing machines, this paper proposes a novel efficient and accurate CORDIC-based chip design by using an iterative pipelining architecture for biped robots. In this design, only low-complexity shift and add operators were used for realizing efficient hardware architecture and achieving the real-time computation of lengths and angles for biped robots. Compared with current designs, this work reduced hardware cost by 7.2%, decreased average errors by 94.5%, and improved average executing performance by 31.5%, when computing ten angles of biped robots.

Keywords: biped robots; binomial approximation; coordinate rotation digital computer (CORDIC); field programmable gate array (FPGA); inverse kinematics; pipeline

Citation: Chung, R.-L.; Hsueh, Y.; Chen, S.-L.; Abu, P.A.R. Efficient and Accurate CORDIC Pipelined Architecture Chip Design Based on Binomial Approximation for Biped Robot. *Electronics* **2022**, *11*, 1701. https://doi.org/10.3390/electronics11111701

Academic Editor: Spyridon Nikolaidis

Received: 4 April 2022
Accepted: 23 May 2022
Published: 26 May 2022

1. Introduction

Currently, robots are applied in many different realms, such as agriculture, medical surgery, disaster relief, and robotic exploration, among others. In the area of agriculture, several human tasks can be replaced by robots for fulfilling agricultural tasks. The agricultural robots are designed to move in the scheduled route and to assemble crops in the specified location. Therefore, agricultural robots require abilities with high-speed motion and precise calculation. Concerning medical surgical application, robotic arms will be designed to manipulate precise surgery in narrow operating rooms. Hence, high-performance chips with high accuracy and high-speed execution time are needed to control the delicate robotic arms. For rescue applications, biped robots, for example, are needed to be designed to learn the movements of humans for disaster relief. To achieve this goal, one important concern is that stable movement should be carefully executed. Thus, the authors proposed and implemented the reinforcement-learning method for biped robots to achieve real-time, continuous dynamic walking with stable balance control [1,2]. Additionally, considering the biped humanoid robot to maintain balance during walking and running, the multiaxis

force–torque sensor is essential and widely used. Kim reviewed multiple types of the multiaxis force–torque sensors used in humanoid robots based on the understanding of biped walking, zero-moment point, and ground-reaction force [3]. Moreover, it is necessary for biped robots to calculate multiple parameters of moving lengths and rotation angles in a short and simultaneous time. Concerning robotic exploration, the coordinate rotation digital computer (CORDIC) algorithm was applied to robotic exploration where the path of robots is designed to avoid obstacles [4,5]. Summarizing, regardless the robotic application, it is necessary to design chips efficiently to compute the moving distances and rotation angles of robots.

A CORDIC-based algorithm is an efficient solution to calculate moving distances and rotation angles with extremely low complexity. It is realized by a sequence of rotation matrix with scale factor when computing moving distances [6]. Moreover, it only uses shifters and adders to compute the trigonometric and hyperbolic functions. In this paper, the CORDIC algorithm with this low complexity is employed to calculate the square root of the sum of two squares, the square root of the difference of two squares, and the arctangent function for biped robots. Therefore, it is possible to compute moving distances and rotation angles of robotic systems in a short time. Recently, energy-efficiency/area-efficiency strategy was the crucial methodology in hardware design [7–11]. In [7], an energy-efficiency design of an image processing technique was proposed for wireless sensor networks. An energy-efficiency modular exponentiation design for public-key cryptography was presented in [8]. In [9], an area-efficient reconfigurable CORDIC with multimode and multitrajectory operations was proposed for the application of communications systems and signal processing. An area-efficient, two-dimensional interpolation architecture based on cubic convolution was proposed in [10], which was developed to calculate the square-root function. In [11], the authors proposed an efficient hardware architecture for fast-Fourier-transform (FFT) implementation where the modified CORDIC (m-CORDIC) algorithm was employed to replace the complex multipliers to reduce complexity significantly.

To achieve high performance of the CORDIC-based algorithm, several research works use the field-programmable gate array (FPGA) and very large-scaled integration (VLSI) architecture to implement high-speed computation [1,12–15]. In [12], efficient implementations of a family of CORDIC algorithms by using FPGA were proposed, where the radix-4 CORDIC, bit-parallel CORDIC, and bit-serial CORDIC were realized and compared. Next, a novel Loeffler discrete cosine transform (DCT) based on a recursive CORDIC architecture was proposed by Chung et al. for reducing the memory demand and increasing image quality of the two-dimensional DCT signal analyzer [13]. In [14], the authors designed a sorted QR decomposition hardware with a CORDIC-based Givens rotation (GR) structure and highly pipelined architecture for high-speed wireless communications with multiple transmit/receive antennas. Then, Pilato et al. proposed a high-accuracy VLSI architecture based on CORDIC structure and fast magnitude estimation for calculating arctangent function [15]. Then, Chung et al. proposed an efficient and high-performance CORDIC-based algorithm for calculating lengths and angles of biped robots with the FPGA implementation [1]. In the design, hardware-sharing-machine technique combined with pipelined structure was utilized for computing square-root and arctangent operators to increase computational accuracy and decrease hardware area efficiently.

According to the aforementioned discussion, the CORDIC-based algorithm is widely used in many kinds of engineering applications to achieve high-performance hardware design. However, to the best of our knowledge, the study on increasing both performance and accuracy of CORDIC-based architecture for biped robots remains limited. Therefore, it is important to develop a high performance, high accuracy, and low complexity CORDIC-based algorithm and its FPGA implementation for biped robots. Hence, this study proposes an efficient and accurate CORDIC-based FPGA design for biped robots based on binomial approximation, pipelined architecture, and hardware-sharing machines. The remainder of the paper is organized as follows. Section 2 describes the biped robot model and the conventional CORIDC algorithm. In Section 3, the hardware architecture with a cost-

efficient and hardware-oriented CORDIC-based algorithm is proposed for biped robots. In Section 4, hardware simulation results of the CORDIC-based algorithm for biped robots are demonstrated. Section 5 describes and provides a discussion of the simulation results. Finally, concluding remarks are made in Section 6.

2. Biped Robot Model and CORIDC Algorithm

2.1. Biped Robot Model

In terms of robotic feet, robots can be divided into wheeled robots, crawler-type robots, and biped robots. Among them, biped robots are the most challenging to design because the step length and the rotation angle of each step of a biped robot need to be calculated. Biped robots can be used in rescue and exploration tasks. Therefore, biped robots have caught the attention of many researchers [16–18]. In [16], the authors proposed a new humanoid biped robot with an efficient energy usage mechanism to achieve versatility, efficient mobility, and high endurance. Then, to design the biped robot with a stable and smooth walk, the authors proposed the reinforcement Q-learning mechanism with an automatic training platform to acquire a straightforward gait pattern for biped robots [17]. In the implementation of biped robots, it is the core basis to calculate the step length and rotation angle. In [18], the authors built a mathematical model for calculating the necessary ten rotation angles and associated moving distance of the biped robot, as depicted in Figure 1. In Figure 1, the ten angles $\theta_1 - \theta_{10}$ in the front view and side view of a biped robot are plotted. The symbol $\theta_a(\theta_b)$ is denoted by two different angles θ_a and θ_b of the biped robots. The θ_a is denoted by the angle of the right leg of biped robots for $a = 1, 2, 5, 6, 7$. The θ_b is denoted by the angle of the left leg of biped robots for $b = 3, 4, 8, 9, 10$. To be specific, in Figure 1a, $\theta_1(\theta_3)$ denotes the hip angle of the right (left) leg in the front view of biped robots, and $\theta_2(\theta_4)$ denotes the shank angle of right (left) leg in the front view. In Figure 1b, $\theta_5(\theta_8)$ denotes the hip angle of right (left) leg in the side view of biped robots, $\theta_6(\theta_9)$ denotes the knee angle of right (left) leg in the side view, and $\theta_7(\theta_{10})$ denotes the shank angle of right (left) leg in the side view. To summarize, both the right leg and the left leg have five rotation angles because of the symmetry of biped robots. The angles $\{\theta_1, \theta_2, \theta_5, \theta_6, \theta_7\}$ describe the right leg position, and the angles $\{\theta_3, \theta_4, \theta_8, \theta_9, \theta_{10}\}$ describe the left leg position. Therefore, in this study, only the equations for the five rotation angles $\{\theta_1, \theta_2, \theta_5, \theta_6, \theta_7\}$ of the right leg are used as the basis. The equations for the other five rotation angles $\{\theta_3, \theta_4, \theta_8, \theta_9, \theta_{10}\}$ for the left leg can be extended straightforwardly. In Figure 1, subscripts $_R$ and $_L$ are used to distinguish the right or left leg, respectively. Thus, $x_R(x_L)$ denotes the length of the right (left) leg in the x-axis from the hip position, $y_R(y_L)$ denotes the length of the right (left) leg in the y-axis from the hip position, and $z_R(z_L)$ denotes the length of the right (left) leg in the z-axis from the hip position. Moreover, $l_{R1}(l_{L1})$ and $l_{R2}(l_{L2})$ denote the thigh length of the right (left) leg and the calf length of the right (left) leg, respectively. Equations (1)–(5) list the calculation for the five angles $\{\theta_1, \theta_2, \theta_5, \theta_6, \theta_7\}$ of the biped robot right leg in terms of $\{x_R,\ y_R, z_R, l_{R1}, l_{R2}\}$ [18]. The calculation for the other five angles $\{\theta_3, \theta_4, \theta_8, \theta_9, \theta_{10}\}$ of the biped robot left leg in terms of $\{x_L,\ y_L, z_L, l_{L1}, l_{L2}\}$ can be obtained very straightforwardly from Equations (1)–(5).

$$\theta_1 = \tan^{-1}\left(\frac{-z_R}{y_R}\right) \tag{1}$$

$$\theta_2 = \pi - \theta_1 \tag{2}$$

$$\theta_5 = 2\tan^{-1}\left(\frac{l_R^2 + l_{R1}^2 - l_{R2}^2}{2l_{R1}l_R + \sqrt{(2l_{R1}l_R)^2 + \left(l_R^2 + l_{R1}^2 - l_{R2}^2\right)^2}}\right) - 2\tan^{-1}\left(\frac{x_R}{l_R + (-z_R)}\right) \tag{3}$$

$$\theta_6 = 2\tan^{-1}\left(\frac{\sqrt{(2l_{R1}l_{R2})^2 - \left(l_R^2 - l_{R1}^2 - l_{R2}^2\right)^2}}{2l_{R1}l_{R2} + \left(l_R^2 - l_{R1}^2 - l_{R2}^2\right)}\right) \tag{4}$$

$$\theta_7 = \pi - \theta_5 - \theta_6 \tag{5}$$

where $l_R = \sqrt{x_R^2 + (-z_R)^2}$.

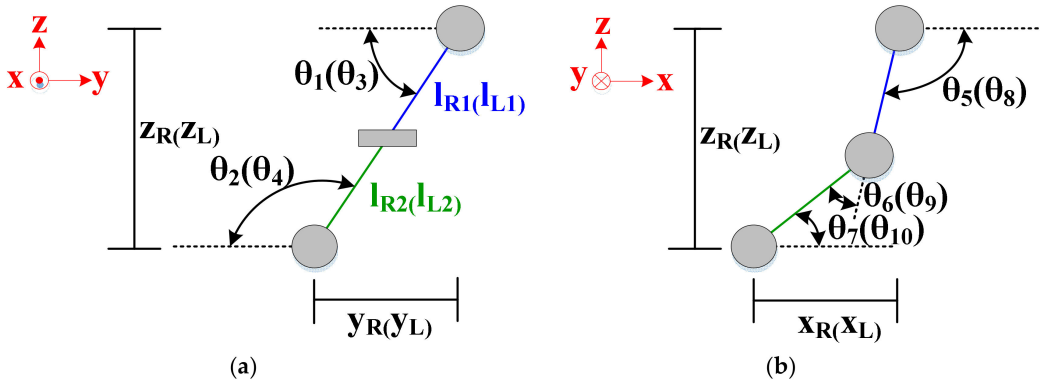

Figure 1. Views of biped robots: (**a**) front view and (**b**) side view.

2.2. CORDIC Algorithm

The CORDIC algorithm is an efficient VLSI algorithm architecture proposed by Volder in 1959 [19]. It can be used to calculate the lengths and angles in the *xy*-plane. The CORDIC architecture only needs add and shift circuits, and can therefore avoid the use of multipliers. The CORDIC algorithm is widely used in different applications such as DCT in image processing [13], FFT in digital signal processing [11], digital frequency synthesizer (DFS) in digital modulation [20], and calculation of moving distances and rotation angles for biped robots [1,18]. To realize the computation of the ten angles of biped robots described above, the CORDIC algorithm of vectoring mode can be applied to significantly reduce the computing complexity.

The CORDIC algorithm can be divided into vectoring mode and rotation mode. The former is to calculate the length, and the latter is to calculate the angle. Moreover, the vectoring mode of the CORDIC algorithm can also be divided into circular vectoring (CV) mode and hyperbolic vectoring (HV) mode. The CV mode can be used to calculate the square root of the sum of two squares while the HV mode can be used to calculate the square root of the difference of two squares. The CORDIC algorithm in the vectoring mode is summarized in Equations (6)–(8), where (x_i,y_i) denotes the coordinate point in the *xy*-plane at the *i*th iteration of the CORDIC algorithm, for $i = 0, 1, \ldots, N$, where N is the number of iterations of the CORDIC algorithm. Moreover, ω_i denotes the rotation angle at the *i*th iteration and the special magnitude of rotation angle at the *i*th iteration can be computed using $\alpha_i = \tan^{-1} 2^{-i}$ (in degrees), which can be previously stored in the lookup table (LUT) for hardware implementation. Equations (6)–(8) are the general expressions for the vectoring mode of the CORDIC algorithm. When $m = 1$, the CV mode is chosen; otherwise, when $m = -1$, HV mode is chosen. In the algorithm, the sign number $\sigma_i \in \{-1, +1\}$ is determined by the negative sign value of y_i:

$$x_{i+1} = x_i - m \cdot \sigma_i \cdot 2^{-i} \cdot y_i \tag{6}$$

$$y_{i+1} = y_i + m \cdot \sigma_i \cdot 2^{-i} \cdot x_i \tag{7}$$

$$\omega_{i+1} = \omega_i - \sigma_i \cdot \alpha_i \tag{8}$$

In the CV mode, the square root of the sum of squares can be calculated. After performing sufficient iterations N on the CORDIC algorithm, the square root of the sum of squares can be obtained by x_N given in (6), but with a scaling factor to modify the final result because the CV mode in the CORDIC algorithm enlarges the calculation of the square-root of the sum of squares value. That is, the square root of the sum of squares value is given by $\sqrt{x_0^2 + y_0^2} = K_C x_N$, where (x_0, y_0) is the coordinate point in the *xy*-plane at the initialization time, x_N is the value of x_i at the Nth iteration, and K_C is the scale factor

in the CV mode in the CORDIC algorithm. When N increases sufficiently, for example, $N = 6$, Kc approaches a constant value of 0.6073. At the same time, the rotation angle at the Nth iteration can also be calculated as $\omega_N = \omega_0 + \tan^{-1} y_0/x_0$, where ω_0 is the angle at the initialization time. Finally, the square root of the difference of squares value can be obtained by x_N in the HV mode in Equation (6), but with K_H, a scale factor in the HV mode. That is, the square root of the difference of squares value is given by $\sqrt{x_0^2 - y_0^2} = K_H x_N$. When N increases sufficiently, K_H approaches a constant value of 1.2076.

2.3. Binomial Approximation for Scale Factor

To achieve enough accuracy for the scale factor, the iteration number is relatively high in the conventional folded-type CORDIC algorithm and it causes a high demand in hardware cost. To solve this problem, a binomial approximation technique is elegantly used here for simplifying the calculation of the scale factor of the CORDIC algorithm. First, the binomial expansion of the nth-order polynomial $(1+x)^n$ is expressed by

$$\begin{aligned}(1+x)^n &= C_0^n 1^n x^0 + C_1^n 1^{n-1} x^1 + \cdots + C_{n-1}^n 1^1 x^{n-1} + C_n^n 1^0 x^n \\ &= \tfrac{n!}{0!n!} x^0 + \tfrac{n!}{1!(n-1)!} x^1 + \cdots + \tfrac{n!}{(n-1)!1!} x^{n-1} + \tfrac{n!}{n!0!} x^n \\ &= 1 + nx + \cdots + nx^{n-1} + x^n\end{aligned} \tag{9}$$

where the binomial coefficient is given by $C_k^n = \frac{n!}{k!(n-k)!}$. Then, the binomial expansion can be approximated by Equation (10) when the value of x is sufficiently small:

$$(1+x)^n \approx 1 + nx \tag{10}$$

The scale factor of the CORDIC algorithm in CV mode at the ith iteration is K_i, given by

$$K_i = \left(1 + 2^{-2i}\right)^{-1/2} \tag{11}$$

Substituting $x = 2^{-2i}$ and $n = -1/2$ into Equation (10), the approximation value of K_i at the ith iteration can be obtained by

$$K_i \approx 1 - 2^{-2i-1} \tag{12}$$

Finally, Equation (13) shows the rotation matrix at the ith iteration, which is used to calculate the length in the vectoring mode [6]:

$$\mathbf{R}_i = K_i \begin{bmatrix} 1 & -\sigma_i 2^{-i} \\ \sigma_i 2^{-i} & 1 \end{bmatrix} \tag{13}$$

By using the binomial approximation of the scale factor given in Equation (12) instead of using the scale factor given in Equation (11), we can observe that the rotation matrix given in Equation (13) can only be obtained by using adders and shifters. In doing so, the same accuracy of the final scale factor can be obtained with fewer iterations. Therefore, the benefits of using the binomial approximation include the decrease in the latency, the reduction in the computing complexity, and the improvement of the performance for hardware implementation. To summarize, the proposed algorithm uses simple operators and support from the fast convergence speed of the scale factor with binomial approximation for improving the accuracy and decreasing hardware area. In so doing, an efficient CORDIC architecture is provided for FPGA implementation.

3. Hardware Architecture

From Equations (1)–(5), it can be observed that the calculation of sum of squares, difference of two squares, and arctangent function are needed to calculate all the angles of biped robots. Therefore, in this study, vectoring mode and rotation mode of CORDIC algorithms were utilized to complete the calculation of Equations (1)–(5). In doing so,

extensive hardware resources can be saved. Figures 2 and 3 depict the CV mode and HV mode of CORDIC algorithms, respectively. In Figure 2, the inputs are x, y, 0, and the outputs give the square root of the sum of squares $\sqrt{x^2+y^2}$ and the arctangent function $\tan^{-1} y/x$. In Figure 3, the inputs are x and y, and the outputs give the square root of the difference of two squares $\sqrt{x^2-y^2}$. Based on Equations (1) and (2) and the CV mode in Figure 2, the hardware architecture for calculation of $\{\theta_1, \theta_2\}$ for a biped robot are shown in Figure 4. Next, based on Equation (3), CV mode in Figure 2, and HV mode in Figure 3, the hardware architecture for calculation of θ_5 for a biped robot is shown in Figure 5. Based on Equation (4), CV mode in Figure 2, and HV mode in Figure 3, the hardware architecture for calculation of θ_6 for a biped robot is shown in Figure 6. Finally, based on Equation (5), θ_5 from Figure 5, and θ_6 from Figure 6, the hardware architecture for calculation of θ_7 for a biped robot is shown in Figure 7.

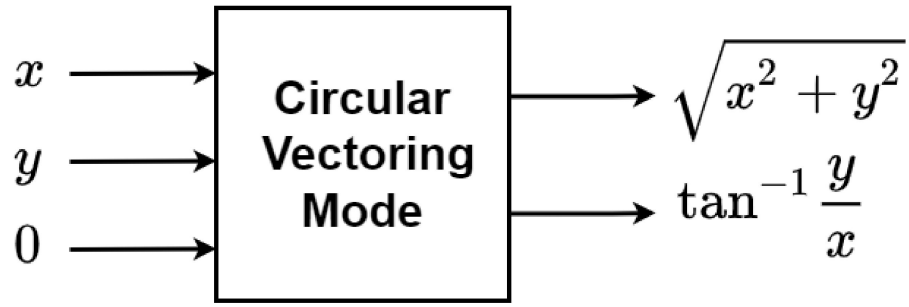

Figure 2. Circular vectoring (CV) mode of the CORDIC algorithm.

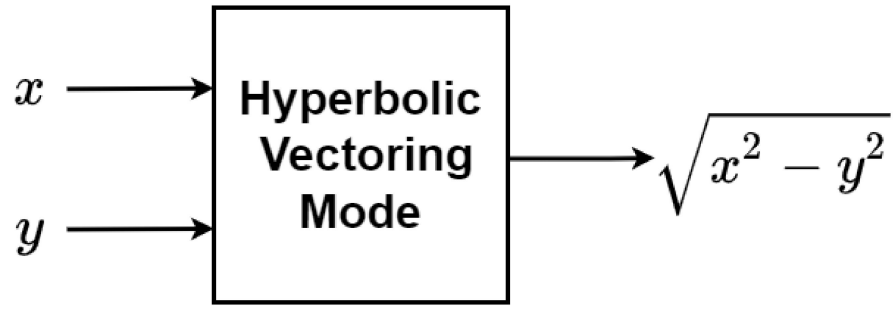

Figure 3. Hyperbolic vectoring (HV) mode of the CORDIC algorithm.

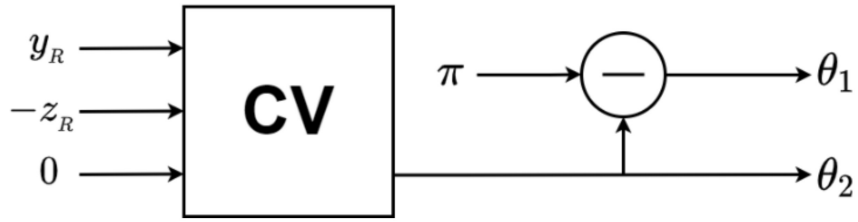

Figure 4. Hardware architecture for calculation of θ_1 and θ_2.

Based on Figures 2–7, the whole block diagram for implementing all the angles of the biped robot can be constructed. To reduce the hardware area, we also carefully incorporate CV with a hardware sharing machine to build the "CV-Mode Hardware Sharing Machine", which is mentioned in Section 3.1. HV with a hardware sharing machine was also incorporated to build the "HV-Mode Hardware Sharing Machine", which is mentioned in Section 3.2.

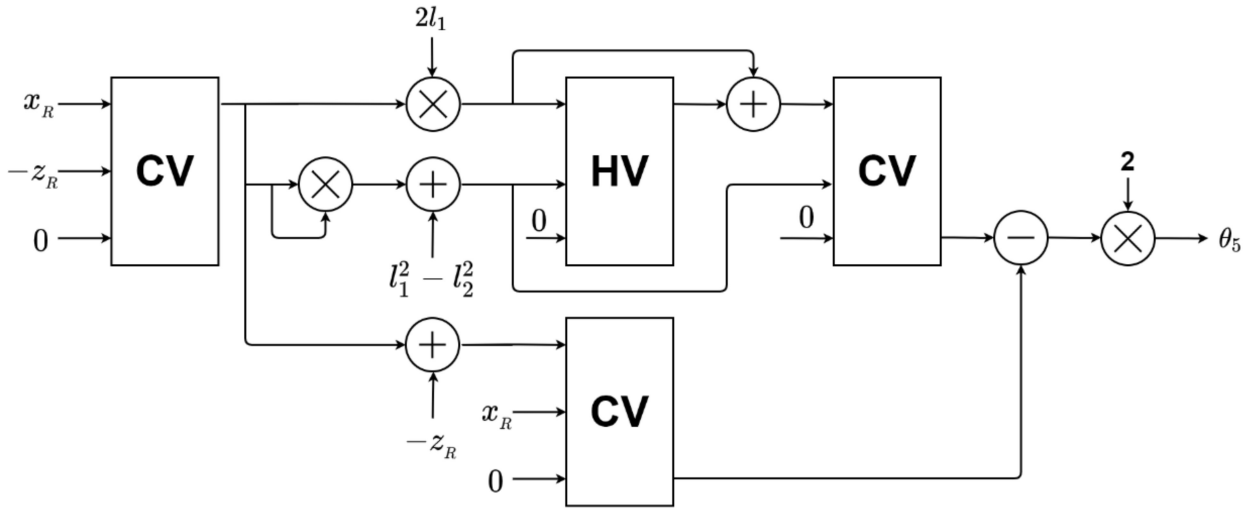

Figure 5. Hardware architecture for calculation of θ_5.

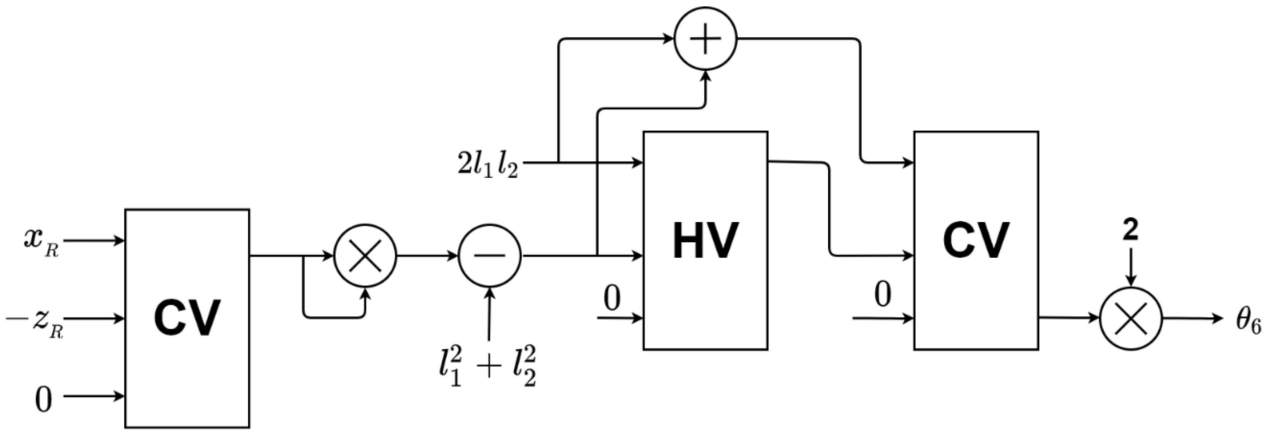

Figure 6. Hardware architecture for calculation of θ_6.

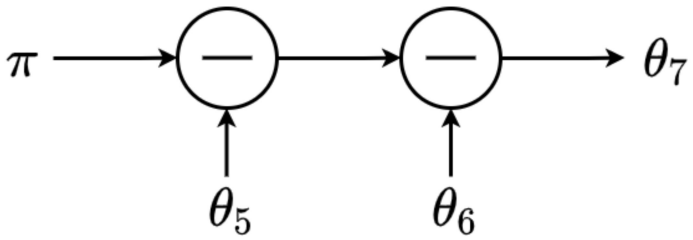

Figure 7. Hardware architecture for calculation of θ_7.

3.1. CV-Mode Hardware Sharing Machine

In this subsection, we apply the hardware sharing machine to the CV-mode design of the CORDIC algorithm, shown in Figure 2, for reducing the hardware area. Figure 8 illustrates the block diagram of the CV component of our architecture with the hardware sharing machine of the CORDIC algorithm. It can be observed in the proposed design that the CV-mode hardware sharing machine contains 18 copies of the CV-mode module combined with a CV scale factor and an error-correct module. By using this design, the high accuracy values of the square root of the sum of squares and arctangent functions can be obtained. To reduce the hardware area of the CV mode, the hardware sharing machine is used. Figure 9 illustrates the detailed architecture of the CV-mode module and CV scale factor with the hardware sharing machine shown in Figure 8, where the former is plotted in the red box and the latter in the blue box. The CV-mode hardware sharing

machine was realized according to the proposed novel binomial approximation algorithm where the scale factor is calculated in Equation (12). The simplified version of the CV scale factor can be realized only by shifters and adders. The hardware sharing machine was employed to implement the scaling-factor generation module with the proposed binomial approximation for hardware cost reduction. The CV-mode hardware sharing machine was developed based on the method of iterative design. Moreover, the pipelined architecture with operator simplification technique was used to enhance the executing efficiency and to reduce hardware area. The proposed design is composed of add and shift operators only, and is realized by a six-stage pipelined architecture which can efficiently improve the performance and reduce the hardware area. The error correction circuit is utilized to scale the value of the iterative output of the CV-mode hardware sharing machine, as shown in Figure 10. Although this design is composed of 15 adders greater than the 9 adders in [1], there only 18 iterations were required, which is less than the 24 iterations in [1], and achieved fast convergence of the scale factor.

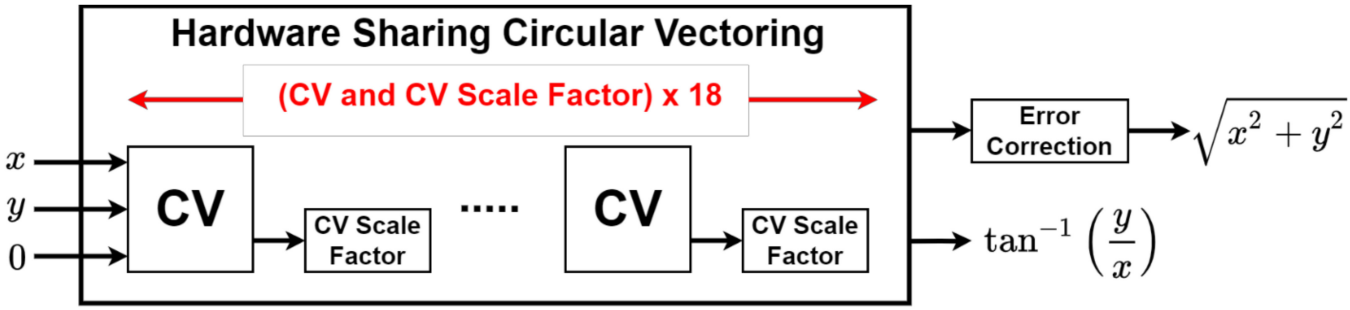

Figure 8. The CV component of our architecture with hardware sharing machine.

3.2. HV1-/HV2-Mode Hardware Sharing Machine

In this subsection, we apply the hardware sharing machine to the design of HV mode of the CORDIC algorithm, shown in Figure 3, for reducing the hardware area. Figure 11 illustrates the block diagram of the HV component of our architecture with the hardware sharing machine of the CORDIC algorithm. In the proposed design, it is efficient to separate an HV into two types, HV1 and HV2 [1]. It can be observed in the proposed design that the HV-mode hardware sharing machine contains 6 copies of the HV1-mode module and 18 copies of the HV2-mode module combined with an HV scale factor. By using this design, the high accuracy values of the square root of the difference of two squares can be obtained. To achieve the target of high performance and low computation, the proposed HV1 and HV2 designs were realized with a hardware sharing machine and pipelined architecture based on [1]. For convenience, the design of the HV1-mode hardware sharing machine proposed by [1] is replotted in Figure 12, and the design of the HV2-mode hardware sharing machine proposed by [1] is replotted in Figure 13. It uses low-complexity operators such as adders and shifters to realize HV1 and HV2. Hence, the proposed HV1-/HV2-mode hardware sharing machine designs have the benefits of low cost and high performance. In order to advance the performance of the HV1-/HV2-mode hardware sharing machine in [1], an initial controller was developed in this study to control the first six iterations in the initial process. Hence, the proposed hardware sharing machine HV1 design had the ability of error correction in the initial iteration. In addition, a simplified architecture of the remaining 12 iterations required by repeating a specific i for each iteration was presented in [1]. Furthermore, the design of an HV scale factor is plotted in Figure 14, which is used to modify the final result to be the correct magnitude. In doing so, the performance of the HV1-/HV2-mode hardware sharing machine with an HV scale factor circuit can achieve high accuracy with low complexity.

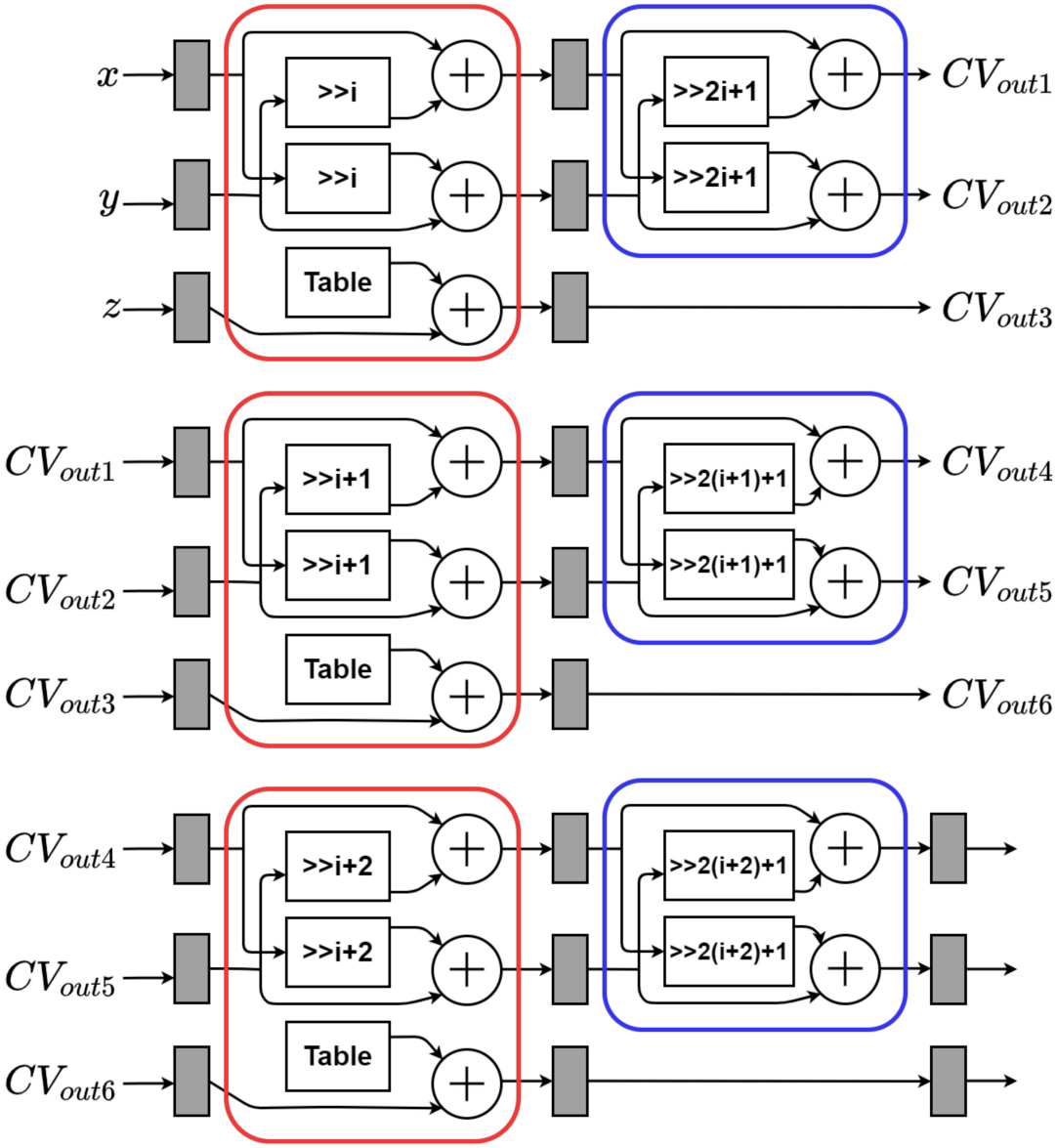

Figure 9. Design of the CV-mode hardware sharing machine.

3.3. Whole Hardware Architecture of the Proposed Efficient CORDIC Design with Hardware Sharing Machine

In this subsection, based on the aforementioned designs given in Figures 4–14, the whole hardware architecture of the proposed efficient CORDIC design with hardware sharing machine is proposed. In the proposed design, 8-bit input and 27-bit output are employed. From Figure 4, it can be observed that the precise calculation of $\{\theta_1, \theta_2\}$ can be obtained by using only one CV component. Then, from Figures 5–7, it can be observed that the precise calculation of $\{\theta_5, \theta_6, \theta_7\}$ can be obtained by using three CV components and one HV component. In the CV component with hardware sharing machine, it contains three CV modules combined with a CV factor circuit. In the HV component with hardware sharing machine, it contains one HV1 module and three HV2 modules. Finally, Figure 15 shows the proposed efficient 22-stage pipelined CORDIC design with hardware sharing machine where the scale factor of the CORDIC algorithm was realized using binomial approximation. In the figure, x denotes x_R (or x_L), y denotes y_R (or y_L), z denotes z_R (or z_L), l_1 denotes l_{R1} (or l_{L1}), and l_2 denotes l_{R2} (or l_{L2}) for notation simplicity. The design contains 3 hardware sharing machines, 1 controller, 18 adders, 17 shifters and 2 multipliers.

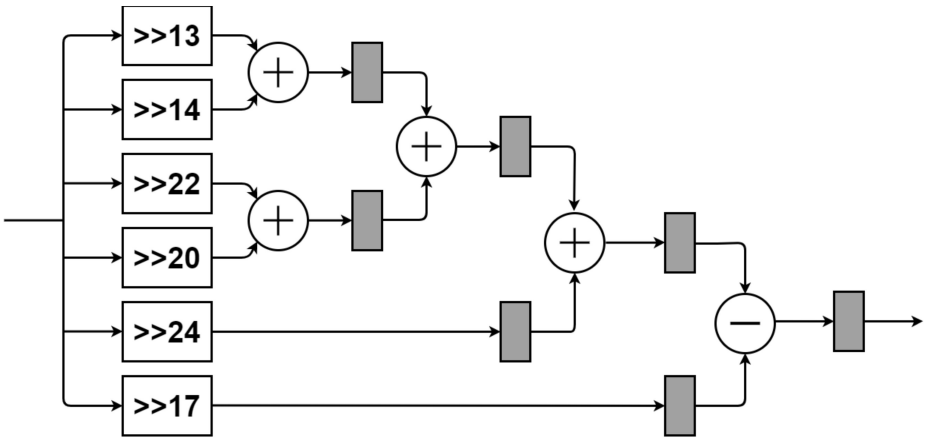

Figure 10. Error correction circuit for the CV-mode hardware sharing machine.

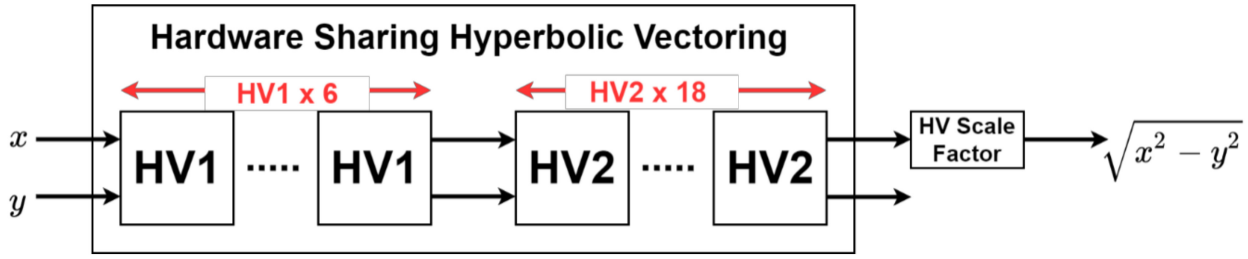

Figure 11. The HV component of our architecture with hardware sharing machine.

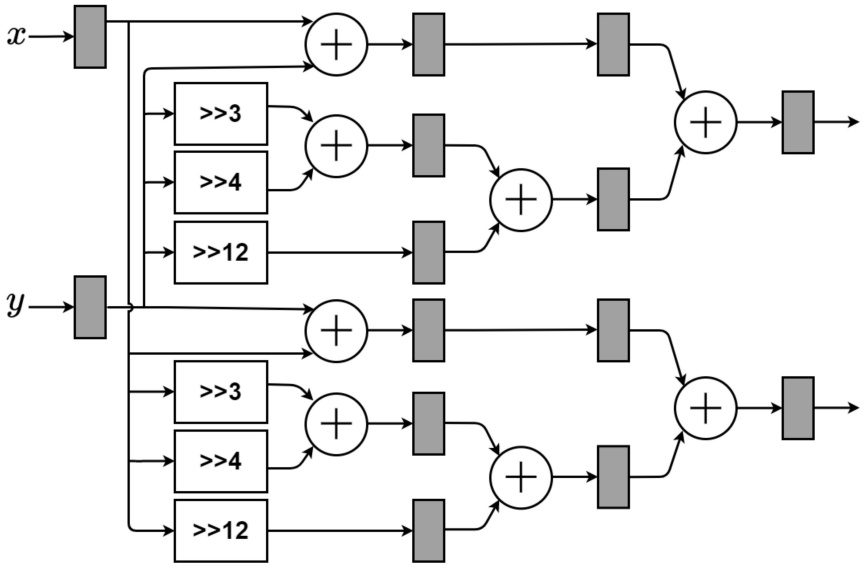

Figure 12. Design of the HV1-mode hardware sharing machine.

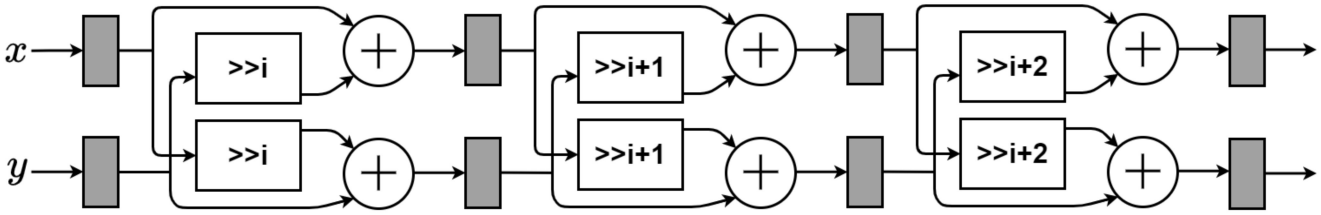

Figure 13. Design of the HV2-mode hardware sharing machine.

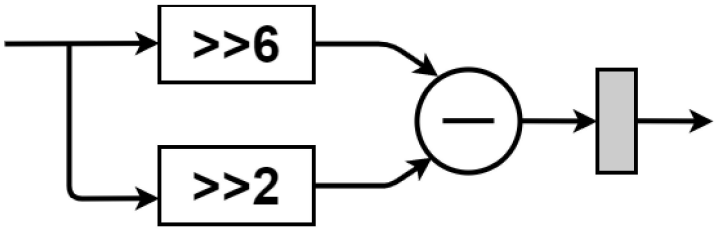

Figure 14. Design of HV scale factor.

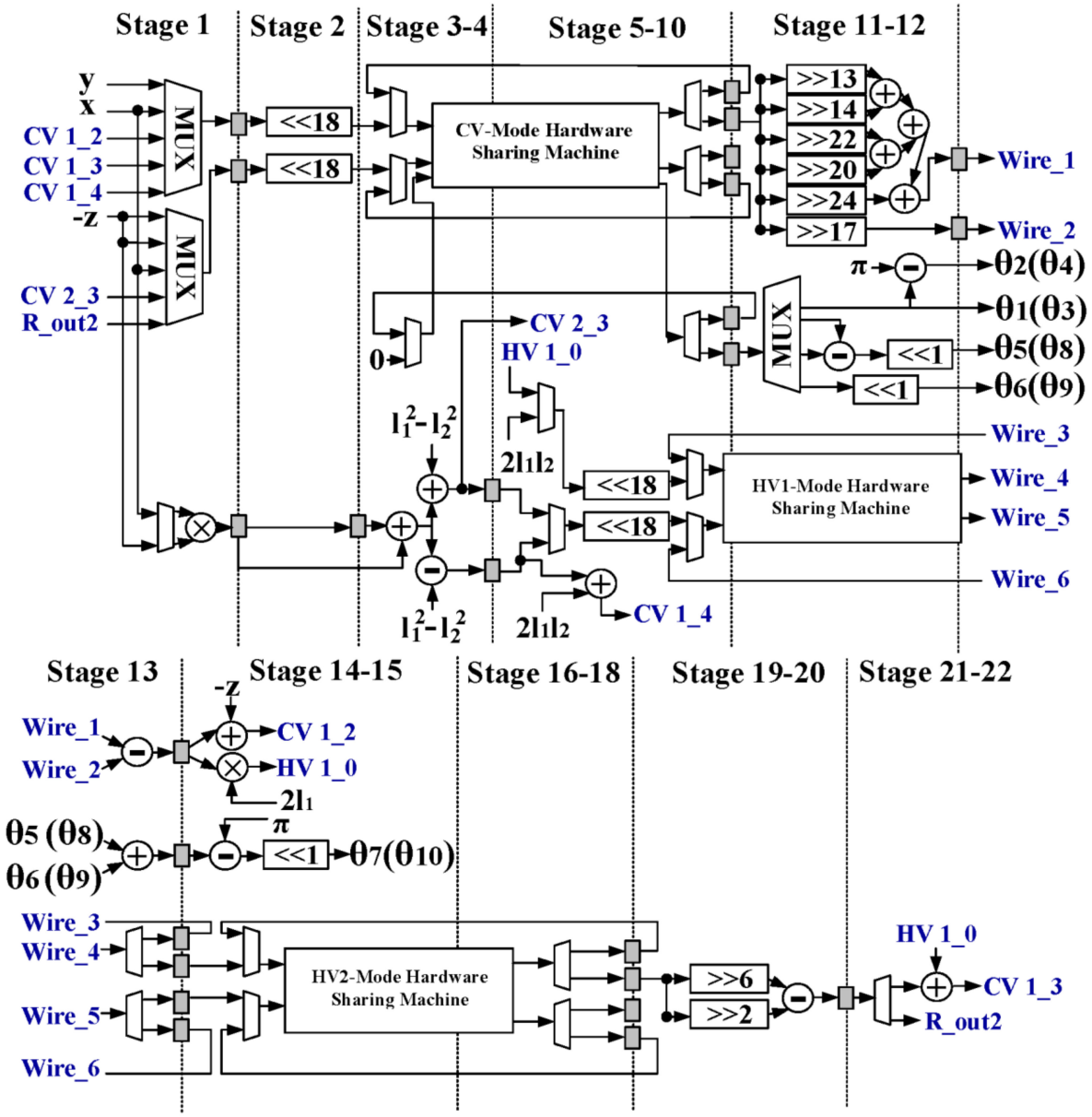

Figure 15. The proposed efficient 22-stage pipelined CORDIC design with hardware sharing machine.

4. Hardware Simulation Result

In this section, one example with several simulations is first presented to discuss the hardware simulation results. Then, in Tables 1 and 2 (Section 5), the results presented are the average results based on several hardware simulations and were not based on a specific corner case. In the hardware simulation, we use two selection modes according to the types of the rotation angles used to calculate all of the rotation angles of the biped robots.

The first type is to calculate angle1 (the calculation of θ_1) and angle2 (the calculation of θ_2). The second type is to calculate angle5 (the calculation of θ_5), angle6 (the calculation of θ_6), and angle7 (the calculation of θ_7). In this hardware design, to improve the accuracy of angle calculation, the input values x and y are enlarged 2^{18} times, which is shown in stage 2 (Figure 15). Then, the angle value of the special magnitude of rotation angle $\alpha_i = \tan^{-1} 2^{-i}$ in the lookup table (LUT) is enlarged 10^7 times, and therefore the final results of angle1, angle2, angle5, angle6, and angle7 are also enlarged 10^7 times. First, when three inputs x, y, and z in stage 1, shown in Figure 2, are set to 18, 55, and –63, respectively, for example, the accurate value of θ_1 can be obtained by Equation (1), where $-z_R$= 63 and y_R = 55, then $\theta_1 = \tan^{-1}(-z_R/y_R)$= 0.853091186091417 (radian/sec). Then, the accurate value of θ_2 can be obtained by Equation (2), where $\theta_2 = \pi - \theta_1$= 2.288501467498376 (radian/sec). With hardware simulation, the 10^7-times enlarged version of angle1 is equal to 8,530,863 and the 10^7-times enlarged version of angle2 is equal to 22,885,064. Compared to the hardware simulation result of angle1 divided by 10^7 to the accurate value of the θ_1, the error occurs at the fifth digit after the decimal point. On the other hand, compared to the hardware simulation result of angle2 divided by 10^7 to the accurate value of the θ_2, the error occurs at the sixth digit after the decimal point. Additionally, hardware simulation results of angle5, angle6, and angle7 for the biped robots were also conducted. After comparison, the maximum error between the accurate value and the hardware simulation results occurs at the fourth digit after the decimal point. To summarize, the proposed efficient 22-stage pipelined CORDIC design for calculating rotation angles of biped robots can achieve satisfactory accuracy vales.

Table 1. Comparison of AME between previous designs and this work.

	$\theta_1(\theta_3)$	$\theta_2(\theta_4)$	$\theta_5(\theta_8)$	$\theta_6(\theta_9)$	$\theta_7(\theta_{10})$	**Average**
[18]	5.7×10^{-4}	8.75×10^{-4}	1.924×10^{-2}	3.8×10^{-2}	2.715×10^{-2}	1.717×10^{-2}
[1]	1.98×10^{-5}	5.36×10^{-7}	2.39×10^{-2}	1.9×10^{-3}	2.3×10^{-3}	5.624×10^{-3}
This work	2.45×10^{-6}	5.63×10^{-6}	7.79×10^{-4}	7.54×10^{-4}	1.13×10^{-5}	3.11×10^{-4}

Table 2. Comparison of latency (μs) between previous designs and this work.

	$\theta_1(\theta_3)$	$\theta_2(\theta_4)$	$\theta_5(\theta_8)$	$\theta_6(\theta_9)$	$\theta_7(\theta_{10})$	**Average**
[18]	0.12	0.13	0.43	0.42	0.44	0.308
[1]	0.063	0.063	0.189	0.063	0.197	0.115
This work	0.043	0.043	0.129	0.043	0.136	0.0788

5. Results and Discussion

First, a more detailed comparison obtained the average results based on several hardware simulations, which are listed in Tables 1 and 2. Altera Cyclone-IV FPGA on the Quartus II platform was employed for evaluating the performance of the proposed design. The absolute maximum error (AME) is used as the performance index for evaluating the accuracy of CORDIC algorithms in previous studies [1,18]. Table 1 lists the comparison of AME with previous algorithms and this work. Since the proposed design in this study was realized based on the new binomial approximation algorithm to calculate the scaling factors, the accuracy result in this work increased to over 94.5% and 98.2% compared to the previous studies in [1,18], respectively. In addition, the iterating cycles obtained by using the proposed algorithm with binomial approximation can be reduced to 18, whereas, in the previous work, it needs 24 iterating cycles. To be specific, the proposed design required 6 cycles for computing the angles of $\theta_1(\theta_3)$, $\theta_2(\theta_4)$, and $\theta_6(\theta_9)$, 18 cycles for computing the $\theta_5(\theta_8)$, and 19 cycles for computing the $\theta_7(\theta_{10})$, which were significantly less than the previous designs [1,18].

Next, hardware simulation results listed in Table 2 are presented to show that the proposed design only spent 0.043 μs to obtain the angles $\theta_1(\theta_3)$, $\theta_2(\theta_4)$, and $\theta_6(\theta_9)$; 0.129 μs

to obtain the angles $\theta_5(\theta_8)$; and 0.136 µs to obtain the angles $\theta_7(\theta_{10})$. The average latency of the proposed design is 0.0788 µs, which means that a result could be produced in 0.0788 µs. With that, the calculated throughput of the proposed design is 12.69 Mega per result. Table 2 lists the comparison of the latency between the previous works and this work, where the proposed design in this study improved the execution performance by 31.5% and 74.4% relative to [1,18], respectively.

Finally, the comparisons of the computational resources and hardware costs in this work and the previous designs [1,18] are listed in Table 3. From Table 3, the proposed design with the hardware sharing technique consisted of 47 adders/subtractors and two multipliers. Comparing the proposed work to the previous design [1], this work can save one multiplier but adds only five adders/subtractors. On the basis of using the 171 NAND-equivalent gates for a adder/subtractor and 1773 NAND-equivalent gates for a multiplier synthesized by a TSMC 0.18 µm CMOS process, the proposed design had a size of 11.6 k gate counts, which indicates a reduction of 7.2% and 53.2% relative to [1,18], respectively. The power consumption of this work is 5.54 mW when operating at 100 MHz which was synthesized by Design Vision tool based on TSMC 0.18 µm CMOS process. Hence, the proposed design can achieve higher accuracy but with less latency than previous designs.

Table 3. Comparisons of computing resources and hardware costs between previous designs and this work.

	Implementation	Adders/Subtractors	Multipliers	Gate Counts	Power Consumption
[18]	FPGA	31	11	24.8K	N/A
[1]	TSMC 0.18 µmCMOS process	42	3	12.5K	5.97 mW @100 MHz
This wok	TSMC 0.18 µmCMOS process	47	2	11.6K	5.54 mW @100 MHz

6. Conclusions

This paper developed a new binomial approximation algorithm to calculate the scale factor of the CORDIC algorithm to achieve high accuracy and low complexity. Additionally, the efficient architecture of the CORDIC-based algorithm for calculating rotation angles of biped robot design was constructed. The architecture had benefits of lower cost and higher performance by using a pipeline and hardware sharing machine techniques. Therefore, the proposed architecture of the CORDIC-based algorithm is a promising candidate for efficiently calculating rotation angles of biped robots.

Author Contributions: Conceptualization, S.-L.C. and P.A.R.A.; data curation, R.-L.C. and Y.H.; formal analysis, Y.H.; funding acquisition, S.-L.C.; methodology, R.-L.C.; project administration, S.-L.C.; resources, S.-L.C.; supervision, R.-L.C.; validation, Y.H.; writing—original draft, R.-L.C., Y.H. and P.A.R.A.; writing—review and editing, R.-L.C., P.A.R.A. and S.-L.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Ministry of Science and Technology (MOST), Taiwan, under grant numbers of MOST-108-2628-E-033-001-MY3, MOST-110-2622-E-131-002, and the National Chip Implementation Center, Taiwan.

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

References

1. Chung, R.-L.; Zhang, Y.-Q.; Chen, S.-L. Fully pipelined CORDIC-based inverse kinematics FPGA design for biped robots. *Electron. Lett.* **2015**, *51*, 1241–1243. [CrossRef]
2. Lin, J.-L.; Hwang, K.-S.; Jiang, W.-C.; Chen, Y.-J. Gait balance and acceleration of a biped robot based on Q-Learning. *IEEE Access* **2016**, *4*, 2439–2449. [CrossRef]
3. Kim, J.-H. Multi-axis force-torque sensors for measuring zero-moment point in humanoid robots: A review. *IEEE Sens. J.* **2020**, *20*, 1126–1141. [CrossRef]
4. Vyas, P.; Vachhani, L.; Sridharan, K.; Pudi, V. CORDIC-based azimuth calculation and obstacle tracing via optimal sensor placement on a mobile robot. *IEEE/ASME Trans. Mechatron.* **2016**, *21*, 2317–2329. [CrossRef]

5. Vachhani, L.; Sridharan, K.; Meher, P.K. Efficient FPGA realization of CORDIC with application to robotic exploration. *IEEE Trans. Ind. Electron.* **2009**, *56*, 4915–4929. [CrossRef]
6. Meher, P.K.; Valls, J.; Juang, T.-B.; Sridharan, K.; Maharatna, K. 50 years of CORDIC: Algorithms, architectures, and applications. *IEEE Trans. Circuits Syst. I* **2009**, *56*, 1893–1906. [CrossRef]
7. Phamila, Y.A.V.; Amutha, R. Low complexity energy efficient very low bit-rate image compression scheme for wireless sensor network. *Inf. Processing Lett.* **2013**, *113*, 672–676. [CrossRef]
8. Satyanarayana, V.; Ramasubramanian, N. Energy efficient modular exponentiation for public-key cryptography based on bit forwarding techniques. *Inf. Processing Lett.* **2017**, *119*, 25–38.
9. Aggarwal, S.; Meher, P.K.; Khare, K. Concept, design, and implementation of reconfigurable CORDIC. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2016**, *24*, 1588–1592. [CrossRef]
10. Wang, D.; Ercegovac, M.D.; Zheng, N. Design of high-throughput fixed-point complex reciprocal/square-root unit. *IEEE Trans. Circuits Syst. II* **2010**, *57*, 627–631. [CrossRef]
11. Nguyen, H.N.; Khan, S.A.; Kim, C.-H.; Kim, J.-M. A pipelined FFT Processor using an optimal hybrid rotation scheme for complex multiplication: Design, FPGA implementation and analysis. *Electronics* **2018**, *7*, 137. [CrossRef]
12. Vadlamani, S.; Mahmoud, W. Comparison of CORDIC algorithm implementations on FPGA families. In Proceedings of the IEEE International Symposium on System Theory (SSST-2002), Huntsville, AL, USA, 19 March 2002; pp. 192–196.
13. Chung, R.-L.; Chen, C.-W.; Chen, C.-A.; Abu, P.A.R.; Chen, S.-L. VLSI implementation of a Cost-Efficient Loeffler DCT algorithm with recursive CORDIC for DCT-based encoder. *Electronics* **2021**, *10*, 862. [CrossRef]
14. Sun, L.; Wu, B.; Ye, T. Design and VLSI implementation of a reduced-complexity sorted QR decomposition for high-speed MIMO systems. *Electronics* **2020**, *9*, 1657. [CrossRef]
15. Pilato, L.; Fanucci, L.; Saponara, S. Real-time and high-accuracy arctangent computation using CORDIC and fast magnitude estimation. *Electronics* **2017**, *6*, 22. [CrossRef]
16. Hobart, C.G.; Mazumdar, A.; Spencer, S.J.; Quigley, M.; Smith, J.P.; Bertrand, S.; Pratt, J.; Kuehl, M.; Buerger, S.P. Achieving versatile energy efficiency with the WANDERER biped robot. *IEEE Trans. Robot.* **2020**, *36*, 959–966. [CrossRef]
17. Wong, C.-C.; Liu, C.-C.; Xiao, S.R.; Yang, H.-Y.; Lau, M.-C. Q-learning of straightforward gait pattern for humanoid robot based on automatic training platform. *Electronics* **2019**, *8*, 615. [CrossRef]
18. Wong, C.C.; Liu, C.C. FPGA realisation of inverse kinematics for biped robot based on CORDIC. *Electron. Lett.* **2013**, *49*, 332–334. [CrossRef]
19. Volder, J.E. The CORDIC trigonometric computing technique. *IRE Trans. Electron. Comput.* **1959**, *EC-8*, 330–334. [CrossRef]
20. Kajur, R.; Prasad, K.V. Hardware realization of GMSK system using pipelined CORDIC module on FPGA. *Appl. Inform. Cybern. Intell. Syst.* **2020**, 21–31. [CrossRef]