1998 International Computer Symposium
Workshop on Computer Architecture
December 17-19, 1998, N.C.K.U., Tainan, Taiwan, R.O.C.

Design, Implementation and Error Analysis of Redundant CORDIC Processors for Fast Vector Rotation and Trigonometric Function Evaluation *

Shen-Fu Hsiao Chung-Yi Liu Jen-Yin Chen
Institute of Computer and Information Engineering
National Sun Yat-Sen University
Kaohsiung, Taiwan

Abstract

A new high-speed redundant CORDIC processor is designed and implemented based on the double rotation method, which turns out to be the two-dimensional (2-D) Householder CORDIC, a special case of the generalized Householder CORDIC in the 2-D Euclidean vector space. The new processor has the advantages of regular structure and high throughput rate. The pipelined structure with radix-2 signed-digit (SD) redundant arithmetic is adopted to reduce the carry-propagation delay of the adders while the digit-serial structure alleviates the burden of the hardware cost and I/O requirement. Compared to previously proposed designs, the new CORDIC processor preserves the constant scaling factor, an important merit of the original CORDIC, and thus does not require any complicated division or square-root operations for variable scaling factor calculation. Practical VLSI chip implementation of the fixed-point redundant CORDIC processor using 0.6um standard cell library is given including detailed numerical error analysis.

1 Introduction

Advances in VLSI technology have stimulated great interest in developing special-purpose hardware for applications requiring high computing speed. An arithmetic unit known as a CORDIC (COordinate Rotation DIgital Computer) unit [1], has received much attention since it enables the efficient implementation of various types of rotations using simple hardware components, mainly adders and shifters. Much research has been directed to the design of CORDIC-based parallel algorithms and architectures to perform basic matrix computations such as QR decomposition, linear system solution, eigenvalue decomposition and singular value decomposition [2]. Other interested applications of CORDIC processors include coordinate transformation in computer graphics and trigonometric and hyperbolic function evaluations in the arithmetic unit of microprocessors [3]. Recently, the CORDIC algorithm is extended to higher dimensional spaces with applications mainly in parallel computations of general complex matrices on processor arrays [4][5].

Recently, Ercegovac and Lang have proposed using redundant signed-digit (SD) adders [6] with on-line arithmetic to replace the conventional binary adders in order to reduce the inherent carry delay [7]. However, the redundant on-line CORDIC destroys the nice property of the *constant* scaling factor in the original CORDIC due to a different sign selection rule, and hence the multiplication of the non-constant scaling factor requires special attention. Several approaches have been proposed to overcome the non-constant scaling factor problem. For example, Ercegovac and Lang used additional complicated hardware for the non-constant scaling factor calculation and correction [7]. Takagi, *et al.* , proposed the double rotation method and correcting rotation method to preserve the scaling factor for CORDIC vector rotation, but they did not extend to CORDIC angle calculation [8]. Lee and Lang inserted additional correcting iterations into the original CORDIC iterations in order to preserve the constant scaling factor [9]. Dawid and Meyr proposed a transformed CORDIC algorithm, called differential CORDIC (DCORDIC), which achieves fast redundant CORDIC implementation on pipelined digit-parallel structure at the cost of more pre-skew latches for the input signals [10]. Besides, the DCORDIC algorithm is not suited to digit-serial structure due to the high initial delay in each iteration.

In this paper, a new redundant on-line CORDIC processor with constant scaling factor and regular structure is presented based on the double-rotation method, which turns out to be the 2-D Householder CORDIC, a special case of the recently proposed multi-dimensional CORDIC with rational iterative scaling factors [4]. The processor not only preserves the constant scaling factor, but also keeps the nice

*This work was supported by the National Science Council of R.O.C. Taiwan under contract NSC 85-2215-E-110-009.

property of regular CORDIC structure, an important issue in VLSI implementation. Both angle calculation and vector rotation modes are readily performed in the redundant CORDIC processor. VLSI implementation of the processor based on 0.6 μm standard cell library is presented. In order to derive a practical guideline for the design of the fixed-point redundant CORDIC processor, we analyze the numerical errors of the implementation by considering the round-off errors, angle approximation errors and scaling factor decomposition errors in both angle calculation and vector rotation modes.

2 CORDIC Algorithms

2.1 Original CORDIC

According to the original CORDIC algorithm [1] a plane rotation with rotation angle θ can be decomposed into product of n elementary rotations:

$$\prod_{i=0}^{n-1} R(\sigma_i) = \prod_{i=0}^{n-1} \frac{1}{\sqrt{1+2^{-2i}}} \begin{pmatrix} 1 & \sigma_i 2^{-i} \\ -\sigma_i 2^{-i} & 1 \end{pmatrix} \tag{1}$$

The elementary rotation matrix $R(\sigma_i)$ represents the rotation of angle $\theta_i = \tan^{-1}(2^{-i})$ at the i-th iteration and $\sigma_i \in \{1, -1\}$ determines the direction of the rotation. An *unscaled* CORDIC iteration consists in applying the elementary rotation without performing the iterative scaling by $1/\sqrt{1+2^{-2i}}$. Thus, by adding another recurrence for accumulation of the elementary rotation angles θ_i, the i-th unscaled CORDIC iteration is expressed, with $i = 0, 1, \cdots, n-1$, as

$$\begin{aligned} x(i+1) &= x(i) + \sigma_i 2^{-i} y(i) \\ y(i+1) &= y(i) - \sigma_i 2^{-i} x(i) \\ z(i+1) &= z(i) + \sigma_i \theta_i \text{ with } \theta_i = \tan^{-1}(2^{-i}) \end{aligned} \tag{2}$$

where n, the number of unscaled CORDIC iterations, determines the resolution of the CORDIC operation. The above unscaled CORDIC iteration can be realized by shift-and-add operations performed concurrently on a CORDIC processor.

The overall scaling factor $K = \prod_{i=0}^{n-1} 1/\sqrt{1+2^{-2i}}$ is a constant as long as n is predefined and can be decomposed *multiplicatively* into several simple factors such that the multiplication of a 2-D vector by each simple factor is also implemented by two concurrent shift-and-add operations, called a *scaling CORDIC iteration*, which can also be realized using hardware resource available on the same CORDIC processor.

CORDIC algorithm has two function modes: *angle calculation* and *vector rotation*. In the angle calculation mode, a sequence of n unscaled rotations is applied to a vector $[x\ y]^T$ to bring it along the first canonical axis. The control signs $\{\sigma_i, 0 \le i \le n-1\}$ which encode the rotation angle $\theta = \tan^{-1}(x/y)$ are evaluated according to $\sigma_i = sign[x(i) \cdot y(i)]$ so that the accumulation of the elementary rotation angles $\sum_{i=0}^{n-1} \sigma_i \theta_i = \sum_{i=0}^{n-1} \sigma_i \tan^{-1}(2^{-i})$ approximates $\tan^{-1}(y/x)$, the angle between the initial vector and the first axis.

In the vector rotation mode, the given rotation angle θ is decomposed iteratively into $\theta = \sum_{i=0}^{n-1} \sigma_i \tan^{-1}(2^{-i})$, and the control signs σ_i are used to rotate vector $[x\ y]^T$ through the same angle. In many signal processing applications, the control signs σ_i are often made available as the result of a preliminary CORDIC evaluation and thus the angle decomposition is not necessary.

2.2 2-D Householder CORDIC

Recently a family of generalized multi-dimensional CORDIC algorithm, called Householder CORDIC algorithm, is presented with applications to several important digital signal processing operations [4]. For example, using the 4-D Householder CORDIC, the computation speed of complex singular value decomposition is increased by at least five times compared to the fastest alternative [5]. The i-th elementary rotation matrix of the m-D Euclidean Householder CORDIC algorithm can be written as

$$R_m(s_i) = E_m \left(I_m - 2\frac{u_i u_i^T}{u_i^T u_i} \right)$$

where

$$E_m = I_m - 2\frac{e_1 e_1^T}{e_1^T e_1} \text{ with } e_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

$$u_i = \begin{bmatrix} 1 \\ s_i t_i \end{bmatrix} = \begin{bmatrix} 1 \\ \sigma_{1,i} t_i \\ \vdots \\ \sigma_{m-1,i} t_i \end{bmatrix}, \quad t_i = 2^{-i}.$$

In the 2-D vector space where $e_1 = [1\ \ 0]^T$ and $u_i = [1\ \ \sigma_i 2^{-i}]^T$, the Householder CORDIC elementary rotation matrix becomes

$$R(\sigma_i) = \frac{1}{1+2^{-2i}} \begin{pmatrix} 1-2^{-2i} & \sigma_i 2^{-i+1} \\ -\sigma_i 2^{-i+1} & 1-2^{-2i} \end{pmatrix} \tag{3}$$

where $\sigma_i \in \{1, -1\}$. This variant of the original 2-D CORDIC algorithm, called the 2-D Householder CORDIC, in fact performs double rotations in each CORDIC iteration. Similar to eqn. (2), an unscaled 2-D Householder CORDIC iteration can be written as $(i = 1, 2, \cdots, n)$

$$\begin{aligned} x(i+1) &= [1-2^{-2i}]x(i) + \sigma_i 2^{-i+1} y(i) \\ y(i+1) &= -\sigma_i 2^{-i+1} x(i) + [1-2^{-2i}] y(i), \\ z(i+1) &= z(i) + \sigma_i \theta_i \text{ with } \theta_i = 2\tan^{-1}(2^{-i}) \end{aligned} \tag{4}$$

where the i-th elementary rotation angle θ_i for the 2-D Householder CORDIC is $2\tan^{-1}(2^{-i})$ instead of $\tan^{-1}(2^{-i})$ as in the original CORDIC, and the iteration index i runs from 1 to n instead of 0 to $n-1$.

Note that the iterative scaling factor $1/(1+2^{-2i})$ in eqn. (3) is a rational function instead of, $1/\sqrt{1+2^{-2i}}$, the *square-root* of a rational function as in the original CORDIC. Similar to the original CORDIC, the overall scaling factor $K=\prod_{i=1}^{n} 1/(1+2^{-2i})$ is a constant as long as n is predetermined, and can be decomposed into products of factors such that the multiplication of a vector component by each factor can be implemented by the shifters, 3-to-2 carry-save adders (CSAs) and adders available in the same 2-D Householder processor. In fact, an n-D (for any $n \geq 2$) Householder CORDIC can be easily realized using redundant arithmetic without changing the overall scaling factor since the iterative scaling factor is rational. The double-rotation 2-D Householder CORDIC is only a special of case of the more generalized Householder CORDIC.

3 Redundant CORDIC

3.1 Original Redundant CORDIC

As mentioned before, a complete CORDIC operation requires n unscaled CORDIC iterations plus s scaling iterations. In order to build a *pipelined* CORDIC processor, the iterative CORDIC structure may be unfolded into $(n+s)$ pipelined stages where each stage realizes either an unscaled CORDIC iteration or a scaling iteration. In this pipelined structure, the costly barrel shifters are now replaced by simple hardwired interconnections due to the fixed number of right-shift bits required in each stage. Since the carry-propagation delay of the adder in each stage decides the throughput rate of the pipelined CORDIC processor, one may wish to design a carry-free adder with the carry delay independent of the wordlength. The signed-digit (SD) adders proposed in [6] has been employed in a recently proposed redundant on-line CORDIC [7] where the adders in each stage of the pipelined CORDIC processor are implemented by *digit-serial* SD adders (SDAs) with the inputs starting from the most significant digits (MSDs). In this redundant on-line implementation, the right-shift wiring is replaced by simple delay elements. By substituting $y(i)$ in eqn. (2) by $w(i)=2^{i}y(i)$, we can rewrite the unscaled CORDIC iterations for angle calculation as

$$\begin{array}{l} x(i+1)=x(i)+\sigma_i 2^{-2i}w(i) \\ w(i+1)=2[w(i)-\sigma_i x(i)] \\ z(i+1)=z(i)+\sigma_i \tan^{-1}(2^{-i}) \end{array} ,$$

$$\sigma_i=\begin{cases} 1 & \text{if } \hat{w}(i)>0 \\ 0 & \text{if } \hat{w}(i)=0 \\ -1 & \text{if } \hat{w}(i)<0 \end{cases}, \quad i=0,1,2,\cdots,n-1. \tag{5}$$

The sign selection σ_i is based on the estimate $\hat{w}(i)$, the first several MSDs of $w(i)$, due to the requirement of a fast sign selection rule in the on-line CORDIC. It is shown in [7] that to assure convergence, the sign selection should include 0 in addition to 1, and -1. Here, $x(0)$ and $y(0)=w(0)$ are respectively the x and y components of the initial input vector, one of them normalized, and $x(0)>0$.

Compared to the original CORDIC, the major change in the redundant CORDIC is that the control signs σ_i belong to the set $\in \{-1,0,1\}$ instead of $\{1,-1\}$. The scaling factor of the redundant online CORDIC with the above sign selection rule is $K=\prod_{i=0}^{n-1}\frac{1}{\sqrt{1+\sigma_i^2 2^{-2i}}}$, which is no longer a constant, depending on whether σ_i is zero or not. Thus, on-line calculation of the variable scaling factor is needed. Lee and Lang presented a strategy which restricts the sign selection range to only $\{-1,1\}$ but the approach calls for additional correcting iterations in order to make the algorithm converge [9].

3.2 Redundant 2-D Householder CORDIC

As can be seen in eqn. (3), the control signs σ_i of the 2-D Householder CORDIC rotation matrix are either 1 or -1. If the selection rule includes the selection of zero, as required in the redundant implementation, and the iterative scaling by $(1+2^{-2i})$ is performed in the case of $\sigma_i=0$, the overall scaling factor $K=\prod_{i=1}^{n}1/(1+2^{-2i})$ remains constant independent of the choice of σ_i. This constant-factor approach is not applicable to the original CORDIC because, when $\sigma_i=0$, the redundant 2-D square-root CORDIC requires the iterative scaling of $\sqrt{1+2^{-2i}}$ which calls for complicated square-root operation.

Assume that the initial vector $[x(1)y(1)]^T$ be in the range of $[-\pi/4,\pi/4]$, and $x(1),y(1)$ be normalized so that $1/2 \leq x(1) < 1$. Let $w(i)=2^{i-1}y(i), i=1,2,\cdots,n$. The estimate $\hat{w}(i)=w^{(0)}.(i)w^{(1)}(i)w^{(2)}(i)$ is chosen to consist of three digits, one integer digit $w^{(0)}(i)$ and two fractional digits $w^{(1)}(i)w^{(2)}(i)$. Based on the 2-D Householder CORDIC in eqn. (4), a new redundant CORDIC for angle calculation can be expressed as

Angle Calculation Mode ($i=1,2,\cdots,n$):

$$\begin{array}{l} x(i+1)=[1-\beta_i 2^{-2i-2}]x(i)+\alpha_i 2^{-2i+1}w(i) \\ w(i+1)=2[1-\beta_i 2^{-2i-2}]w(i)-\alpha_i x(i) \\ z(i+1)=z(i)+\alpha_i 2\tan^{-1}(2^{-i-1}) \end{array} ,$$

$$(\alpha_i,\beta_i)=\begin{cases} (1,1) & \text{for } \hat{w}(i)>0 \\ (0,-1) & \text{for } \hat{w}(i)=0 \\ (-1,1) & \text{for } \hat{w}(i)<0 \end{cases}, \tag{6}$$

where $\alpha_i = \sigma_i \in \{1, 0, -1\}$ and the new sign β_i selects either the unscaled 2-D Householder CORDIC operation (when β_i=1) or the multiplication by $1 + 2^{-2i}$ (when $\beta_i = -1$). As observed in [8], special computation rule for the three most significant integer digits is required to generate a redundant representation of $w(i)$ with the MSD located at the first integer digit position. The inclusion of the first integer digit $w^{(0)}$ in the estimation $\hat{w}(i)$ is due to the fact that $|w(i)| < 1$. The time spent on sign selection is relatively small compared to that on other operations.

When $i \geq n/2 + 1$, the x and w recurrences of eqn. (6) degenerates, within accuracy of n fractional bits, to a much simpler form

$$\begin{aligned} x(i+1) &= x(i) \\ w(i+1) &= 2w(i) - \alpha_i x(i) \end{aligned}$$

since the other terms are negligible. Hence the structure for the second half pipelined stages is much simpler than that for the first half pipelined stages.

Similar to the derivation for CORDIC angle calculation, by substituting $z(i)$ by $w(i) = 2^{i-1}z(i)$ and letting $\hat{w}(i) = .w^{(1)}(i)w^{(2)}(i)w^{(3)}(i)$, the redundant 2-D Householder CORDIC vector rotation for input angle in the range of $[-\pi/4, \pi/4]$ can be expressed as

Vector Rotation Mode ($i = 1, 2, \cdots, n$):

$$\begin{aligned} x(i+1) &= [1 - \beta_i 2^{-2i-2i}]x(i) + \alpha_i 2^{-i}y(i) \\ y(i+1) &= -\alpha_i 2^{-i}x(i) + [1 - \beta_i 2^{-2i-2}]y(i), \\ w(i+1) &= 2w(i) + \alpha_i 2^{i+1}\tan^{-1}(2^{-i-1}) \end{aligned}$$

$$(\alpha_i, \beta_i) = \begin{cases} (-1, 1) & \text{for } \hat{w}(i) > 0 \\ (0, -1) & \text{for } \hat{w}(i) = 0 \\ (1, 1) & \text{for } \hat{w}(i) < 0 \end{cases} \tag{7}$$

where the control sign tuples $(\alpha_i, \beta_i), 1 \leq i \leq n$, are selected to force to zero the z component Here, the first three fractional digits are required in the estimation $\hat{w}(i)$. The proof of convergence can be found in [8]. As in the CORDIC angle calculation, the generation of $w(i)$ needs special processing in order to obtain a redundant representation of $w(i)$ with the MSD in the first fractional position. Again, when $i \geq n/2+1$, the x and y recurrences of eqn. (7) reduces to

$$\begin{aligned} x(i+1) &= x(i) + \alpha_i 2^{-i}y(i) \\ y(i+1) &= -\alpha_i 2^{-i}x(i) + y(i). \end{aligned}$$

where the control sign β_i disappears.

3.3 Comparison

Several methods for redundant constant-factor CORDIC implementations have been published recently [8][9][10]. Tab. 1 compares the speed and area performance of the pipelined (unfolded) *digit-parallel* implementation of these methods if we neglect the time for the scaling iterations and the output SD conversion. The addition of two redundant SD numbers takes about the delay of two full-adders (FAs). Using the (p, n) coding for the SD numbers, the addition of three SD numbers can be performed using two 3-to-2 CSAs (one for the postive bits, the other for the negative bits) followed by a two-input SDA, and thus takes the delay of three FAs only. In the comparison, an FA is assumed to take one cycle (roughly equal to the delay of two 2-input XOR gates).

In [8], the double-rotation method is applied to CORDIC vector rotation (VR) only. Our 2-D Householder CORDIC extends to the angle calculation (AC) mode. The sign selection is assumed to take 0.5 cycle. By recognizing the fact of the degenerated form for the second half iterations, the total delay for the CORDIC VR is $(0.5+1+2) \times n/2 + (0.5+2) \times n/2 = 3n$ cycles. The same result can be obtained for the CORDIC AC. The major hardware for the first half stages in the pipelined digit-parallel structure is $(3 \times \text{SDA} + 4 \times \text{CSA}) \times n/2$ while the major hardware for the second half stages is $(3 \times \text{SDA}) \times n/2$. Thus the total major hardware is $3n \times \text{SDA} + 2n \times \text{CSA}$.

In [9], the number of extra correcting iterations depends on the accuracy of the estimation in the sign selection. About $0.25n$ additional correcting iterations are required if six fractional digits are used in estimation, and the corresponding sign selection is assumed to take one cycle in addition to the two cycles for the SD addition. Thus, a total of $(1+2) \times 1.25n = 3.75n$ cycles are required. The major hardware of the arithmetic units is $(3 \times \text{SDA}) \times 1.25n = 3.75n \times \text{SDA}$.

The DCORDIC in [10] requires an initial delay of about n cycles to calculate the absolute value and the exact sign of the first steering variable. As shown in [10], each DCORDIC VR iteration requires 0.5 cycle for sign selection plus 2 cycles for the SD addition, leading to a total of $n + 2.5n = 3.5n$ cycles. In AC mode, only three cycles (two cycles for the SD addition and one cycle for absolute value computation) are required in each iteration. Thus, it takes about $n + 3n = 4n$ cycles to perform DCORDIC AC. The total hardware for the arithmetic units in the DCORDIC is $3n \times \text{SDA} + n \times \text{ABS}$ where ABS denotes a unit of absolute value and sign calculation. Note that since the absolute value and sign calculation is required in each DCORDIC iteration before deciding the addition or subtraction of that iteration, the DCORDIC is not suited to iterative (folded) structure or digit-serial implementation.

Although the comparison in Tab. 1 is for pipelined digit-parallel structure, the VLSI implementation in the next section is based on the pipelined on-line (digit-serial) structure in order to reduce the the hardware and I/O cost.

4 VLSI Implementation

According to eqns. (6)(7), the i-th pipelined stage ($i \leq n/2$) of the redundant on-line 2-D Householder CORDIC processor is derived as shown in Fig. 1 where each wire corresponds to a signed-digit represented by two binary bits. The delay elements of various size implement the shifting operation for proper digit alignment. The multiplexers select the left input during the angle calculation mode and the right input during the vector rotation mode. The sign selection unit scans three MSDs of $w(i)$ and generates appropriate sign tuple (α_i, β_i). The 3-to-2 carry-save adders (CSAs) perform fast addition for the three inputs and generate two outputs for the on-line SDAs followed. The group of delay elements of size 2 accounts for the delay of the sign selection unit and the CSAs. One cycle is assigned for the sign selection and the CSA; the operation of SDA takes another separated cycle. The major component in each redundant on-line CORDIC stage is the on-line SDA. Since the carry of a radix-2 SD addition propagates at most two digits [6], three consecutive digits are required to generate one sum digit, as shown in Fig. 2(a). The structure of an on-line SDA is illustrated in Fig. 2(b). Adjustment of the SD number for $w(i)$ is required to transform the MSDs of $1\bar{1}$ into 01, and to transform the MSDs of $\bar{1}1$ into $0\bar{1}$. The multiplication by 2 in eqns. (6)(7) is realized by one-digit left-shift, which is equivalent to insert a unit delay to every other component.

We practically implemented a redundant digit-serial 2-D Householder CORDIC processor using the cell-based design methodology. The design flow begins with register-transfer level (RTL) Verilog description of the redundant on-line 2-D Householder CORDIC. Then, the Synopsys high-level synthesis tool is used to transform the RTL Verilog code into gate-level description based on the Compass 0.6μm cell library. The pre-layout timing and area estimation is generated at this stage. Afterward, the gate-level Verilog code is read into the Cadence environment for automatic placement and routing to create the physical layout. Finally, post-layout simulation is done in order to acquire more accurate timing and power estimation. In order to supply necessary input stimulus, we also wrote a hardware simulator which generates random input patterns and the corresponding output patterns for examining the correctness of the simulation.

An 8-bit redundant on-line CORDIC processor has been implemented which contains 14 pipelined stages including 12 unscaled CORDIC stages ($n = 12$), two scaling stages, and an on-the-fly SD converter to transform the digit-serial SD outputs to the conventional bit-parallel 2's complement format. The reason for the choice of the iteration number $n = 12$ will be discussed in Sec. 5 of numerical error analysis. As shown in Tab. 2, the processor has a core area of 2700 μm × 2600 μm and can operate at a maximum frequency of more than 100 MHz with power dissipation less than 1W using 5V power supply.

We also synthesized a digit-parallel redundant CORDIC processor where the SDAs in each pipelined stage perform the digit-parallel addition. The synthesized chip information of the word-level redundant CORDIC processor is also included in Tab. 2. Although the digit-parallel implementation has higher throughput, it requires more hardware area, more I/O pins, and consumes more power compared to the digit-serial implementation.

5 Numerical Error Analysis

In this section, a statistical approach to the analysis of the round-off errors for the *fixed-point* redundant 2-D Householder CORDIC processor is presented from which practical guidelines for the design of CORDIC processors are deduced. In [11], Hu proposed a worst-case error analysis method for original CORDIC operations only in vector rotation modes. Kota and Cavallaro extended the analysis to angle calculation mode [12]. Both papers did not take into consideration the errors caused by the implementation of the scaling.

5.1 Modeling of Round-Off Errors

The round-off errors due to the truncation of the right-shifted operand in the i-th unscaled 2-D Householder CORDIC iteration can be modeled as

$$\begin{aligned} v_{i+1} &= \begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} \\ &= \begin{pmatrix} 1-2^{-2i} & \sigma_i 2^{-i+1} \\ -\sigma_i 2^{-i+1} & 1-2^{-2i} \end{pmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} e_i^x \\ e_i^y \end{bmatrix} \\ &= U_i v_i + \mathbf{e}_i, \qquad 1 \leq i \leq n \end{aligned} \tag{8}$$

where e_i^x and e_i^y are the round-off errors due to the truncation of the shifted operands and n is the total number of unscaled CORDIC iterations.

Similarly, the round-off error in a scaling iteration is expressed as

$$\begin{aligned} v_{i+1} &= \begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} \\ &= \begin{pmatrix} 1+\alpha_i 2^{-i+1}+\beta_i 2^{-2i} & 0 \\ 0 & 1+\alpha_i 2^{-i+1}+\beta_i 2^{-2i} \end{pmatrix} \\ &\quad \times \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} e_i^x \\ e_i^y \end{bmatrix} \\ &= C_i v_i + \mathbf{e}_i, \qquad n+1 \leq i \leq n+s \end{aligned} \tag{9}$$

Combining eqns. (8)(10), we have

$$v_{i+1} = M_i v_i + \mathbf{e}_i,\ M_i = \begin{cases} U_i & 1 \leq i \leq n \\ C_i & n+1 \leq i \leq n+s \end{cases}. \tag{10}$$

To simplify the analysis, we assume that the error sources e_i^x and e_i^y are statistically independent random variables with mean μ_e and variance σ_e^2. For a fixed-point system with b fractional bits and downward-direct truncation, $\mu_e = 2^{-b-1}$ and $\sigma_e^2 = 2^{-2b}/12$. First, consider the effect of the i-th stage noise source $\mathbf{e}_i = [e_i^x\ e_i^y]^T$ on the final outputs and set the other noise sources $e_j^x = e_j^y = 0, j \neq i$. The outputs at the final stage due to e_i^x, e_i^y, represented by the vector $\mathbf{f}_i = [f_i^x\ f_i^y]^T$, can be expressed as

$$\mathbf{f}_i = \prod_{j=i+1}^{n+s} M_j \mathbf{e}_i \equiv P_i \mathbf{e}_i, \tag{11}$$

where $P_i = \prod_{j=i+1}^{n+s} M_j$ is the transfer matrix from the outputs of stage i to the final outputs. The mean of vector $\mathbf{f}_i$ is

$$\begin{aligned}\mu_i &= \begin{bmatrix} \mu_i^x \\ \mu_i^y \end{bmatrix} = E(\mathbf{f}_i) = E(\prod_{j=i+1}^{n+s} M_j \mathbf{e}_i) \\ &= \prod_{j=i+1}^{n+s} M_j E(\mathbf{e}_i) = P_i \begin{bmatrix} \mu_e \\ \mu_e \end{bmatrix}. \end{aligned} \tag{12}$$

The covariance of $\mathbf{f}_i$ is

$$F_i = E((\mathbf{f}_i - \mu_i) = \sigma_e^2 P_i P_i^T. \tag{13}$$

Due to the orthogonality of the CORDIC rotation, $P_i P_i^T = \epsilon_i$ is a scaled identity matrix with

$$\epsilon_i = \begin{cases} \prod_{j=i+1}^{n}(1+2^{-2j})^2 \times \\ \prod_{j=n+1}^{n+s}(1+\alpha_j 2^{-j+1} + \beta_j 2^{-2j})^2 \\ \qquad 1 \leq i \leq n-1 \\ \prod_{j=i+1}^{n+s}(1+\alpha_j 2^{-j+1} + \beta_j 2^{-2j})^2 \\ \qquad n \leq i \leq n+s-1 \end{cases}. \tag{14}$$

Hence, the outputs f_i^x, f_i^y are independent with identical variance $\epsilon_i \sigma_e^2$.

Since the model is linear, the overall error at the output of the final stage is

$$\mathbf{f} = \begin{bmatrix} f^x \\ f^y \end{bmatrix} = \sum_{i=1}^{n+s} \mathbf{f}_i$$

where

$$f^x = \sum_{i=1}^{n+s} f_i^x, f^y = \sum_{i=1}^{n+s} f_i^y$$

with

$$f_{n+s}^x = e_{n+s}^x, f_{n+s}^y = e_{n+s}^y.$$

Thus, the mean of $\mathbf{f}$ is

$$\begin{aligned}\mu &= \begin{bmatrix} \mu^x \\ \mu^y \end{bmatrix} = E(\mathbf{f}) = \sum_{i=1}^{n+s} \mu_i \\ &= \sum_{i=1}^{n+s-1} P_i \begin{bmatrix} \mu_e \\ \mu_e \end{bmatrix} + \begin{bmatrix} \mu_e \\ \mu_e \end{bmatrix} \end{aligned} \tag{15}$$

and its covariance matrix is

$$\begin{aligned} E(\mathbf{f}\mathbf{f}^T) &= \sum_{i=1}^{n+s} F_i = \sigma_e^2 \sum_{i=1}^{n+s-1} P_i P_i^T + \sigma_e^2 I \\ &= \sum_{i=1}^{n+s-1} \epsilon_i \sigma_e^2 I + \sigma_e^2 I \equiv \epsilon \sigma_e^2 I \end{aligned} \tag{16}$$

where $\epsilon = \sum_{i=1}^{n+s-1} \epsilon_i + 1$. The random variable f^x can be approximated as a random variable of Gaussian distribution with mean μ^x and variance $\epsilon\sigma_e^2$ because $f^x = \sum_i f_i^x$ with f_i^x contributions independent and identically distributed. Hence, the probability that $|f^x - \mu^x| > 5\sqrt{\epsilon}\sigma_e$ is less than 0.00002. It is a reasonable approximation to take the overall worst-case round-off error as $\mu^x + 5\sqrt{\epsilon}\sigma_e \equiv 2^{-b_r}$ where b_r is defined as the effective fractional bit accuracy for the round-off errors. The strictly worse-case round-off errors as in [11] can be derived in a similar approach, but we found out that the above statistical approach gives bounds closer to the simulation results.

5.2 Angle Approximation Error

Another source of error, called *angle approximation error*, is introduced through the approximation of an angle θ by the sum $\tilde{\theta} = \sum_{i=1}^{n} \sigma_i \tilde{\theta}_i$ of the n *quantized* elementary rotation angles $\tilde{\theta}_i$. The angle approximation error has two sources: one from the finite number of the unscaled CORDIC iterations, n, and the other from the finite resolution of the quantized elementary angle. Assuming n unscaled CORDIC iterations, the difference between the true angle $\tilde{\theta}$ and the ideal angle θ is

$$\Delta\theta = |\theta - \tilde{\theta}| \leq \theta_n + \sum_{i=1}^{n} |\theta_i - \tilde{\theta}_i| \tag{17}$$

where $\theta_n = 2\tan^{-1}(2^{-n})$ is the error due to the finite number of CORDIC iterations and $|\theta_i - \tilde{\theta}_i|$ is the quantization error of the i-th elementary rotation angle. Let $v = [x\ y]^T$ be the ideal result after CORDIC rotations without angle approximation error and let $\tilde{v} = [\tilde{x}\ \tilde{y}]^T$ be the actual result. We have

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} = \begin{pmatrix} \cos\Delta\theta & \sin\Delta\theta \\ -\sin\Delta\theta & \cos\Delta\theta \end{pmatrix} \begin{bmatrix} x \\ y \end{bmatrix}.$$

The difference between v and $\tilde{v}$ is

$$\begin{aligned} \tilde{v} - v &= \begin{bmatrix} \tilde{x} - x \\ \tilde{y} - y \end{bmatrix} = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \\ &= \begin{pmatrix} \cos\Delta\theta - 1 & \sin\Delta\theta \\ -\sin\Delta\theta & \cos\Delta\theta - 1 \end{pmatrix} \begin{bmatrix} x \\ y \end{bmatrix}. \end{aligned}$$

An upper bound for $|\Delta x|$ is given by

$$\begin{aligned} |\Delta x| &\leq (|\sin\Delta\theta| + |\cos\Delta\theta - 1|) \cdot max(|x|, |y|) \\ &\simeq \Delta\theta \cdot max(|x|, |y|) \equiv 2^{-b_a} \end{aligned}$$

where b_a is defined as the effective number of fractional bits for $|\Delta x|$ due to the angle approximation error $\Delta\theta$. The same result can be derived for $|\Delta y|$.

5.3 Overall Effective Error

As mentioned before, the scaling constant K is approximated by the product of s simple factors $\tilde{K} = \prod_{i=n+1}^{n+s}(1 + \alpha_i 2^{-i+1} + \beta_i 2^{-2i})$ implemented by the scaling CORDIC iterations. Hence, an additional error $\Delta K = |K - \tilde{K}|$ due to the scaling constant approximation must be taken into account. Combining the round-off error, the angle approximation error, and the scaling constant approximation error, we obtain the overall effective error

$$2^{-b_r} + 2^{-b_a} + 2^{-b_s} \equiv 2^{-b_{eff}} \qquad (18)$$

where b_r and b_a depend on the internal fractional bit accuracy (b) and the number of unscaled and scaling CORDIC iterations (n and s), while the scaling constant approximation error b_s is fixed for a particular scaling factor decomposition. Our objective is to find n, s and b such that the overall effective error $2^{-b_{eff}}$ does not exceed $2^{-b_{ext}}$ where b_{ext} is the prescribed external bit accuracy. The error analysis for the redundant CORDIC is quite similar to that in Secs. 5.1 and 5.2.

We wrote a hardware simulator which computes the worse-case numerical errors from 10^5 randomly generated input patterns. Tab. 3 compares the analysis results (b_{eff}) with the simulation results (b_{sim}). For single scaling iteration ($s = 1$) with the scaling factor $(1 - 2^{-2} - 2^{-6})$, the effective bit accuracy for the scaling factor approximation error is $b_s = 8.34$ which is not accurate enough when combining with other error sources. Thus, we use two scaling iterations $(1 - 2^{-2} - 2^{-6})(1 + 2^{-8} - 2^{-18})$ which gives approximation error less than $2^{-11.54}$. Boldface numbers indicate choices of b and n assuming the prescribed external bit accuracy is $b_{ext} = 8$. From Tab. 3, we know that 12 unscaled CORDIC iterations and 2 scaling iterations with 12 internal fractional bits are enough to achieve an external accuracy of 8 fractional bits. That is the reason why we select the numbers of the unscaled and scaling CORDIC iterations and the internal wordlength during our VLSI implementation in Sec. 4.

6 Conclusions

A new redundant on-line CORDIC with constant scaling factor was presented using the 2-D Householder CORDIC. Compared to previously proposed approaches, our method does not require complicated scaling factor calculation or extra correcting iterations, and can perform both CORDIC angle calculation and vector rotation. VLSI implementation of the processor using standard cell design methodology is presented. Numerical accuracy analysis of the fixed-point processor is also given.

References

[1] J.E. Volder, *The CORDIC Trigonometric Computing Technique*, IRE Trans. on Electronic Computers, Vol. EC-8, No. 3, pp. 330-334, Sept. 1959.

[2] N.D. Hemkumar and J.R. Cavallaro, *Redundant and On-Line CORDIC for Unitary Transformations*, IEEE Trans. Computers, pp. 941-954, Aug. 1994.

[3] D. Timmermann, *et. al.* , *A CMOS Floating-Point Vector-Arithmetic Unit*, IEEE Trans. Computers, Vol. 29, No. 5, pp. 634-639, May 1994.

[4] S.-F. Hsiao and J.-M. Delosme, *Householder CORDIC Algorithms*, IEEE Trans. Computers, Vol. 44, No. 8, pp. 990-1001, Aug. 1995.

[5] S.-F. Hsiao and J.-M. Delosme, *Parallel Singular Value Decomposition of Complex Matrices Using Multi-dimensional CORDIC Algorithms*, IEEE Trans. Signal Processing, Vol. 44, No. 3, pp. 685-697, Mar. 1996.

[6] A. Avizienis, *Signed-Digit Number Representations for Fast Parallel Arithmetic*, IRE Trans. Electronic Computers, Vol. EC-10, pp. 389-400, Sept. 1961.

[7] M. Ercegovac and T. Lang, *Redundant and On-Line CORDIC: Application to Matrix Triangularization and SVD*, IEEE Trans. Computers, Vol. 39, No. 6, pp. 725-740, June 1990.

[8] N. Takagi, T. Asada, and S. Yajima, *Redundant CORDIC Methods with a Constant Scale Factor for Sine and Cosine Computation*, IEEE Trans. Computers, Vol. 40, No. 9, pp. 989-995, Sept. 1991.

[9] J.-A. Lee and T. Lang, *Constant-Factor Redundant CORDIC for Angle Calculation and Rotation*, IEEE Trans. Computers, Vol. 41, No. 8, pp. 1016-1025, Aug. 1992.

[10] H. Dawid and H. Meyr, *The Differential CORDIC Algorithm: Constant Scale Factor Redundant Implementation without Correcting Iterations*, IEEE Trans. Computers, Vol. 45, No. 3, pp. 307-318, Mar. 1996.

[11] Y.H. Hu, *The Quantization Effects of the CORDIC Algorithm*, IEEE Trans. Signal Processing, Vol. 40, No. 4, pp. 834-844, Apr. 1992.

[12] K. Kota and J. R. Cavallaro, *Numerical Accuracy and Hardware Tradeoffs for CORDIC Arithmetic for Special-Purpose Processors*, IEEE Trans. Computers, Vol. 42, No. 7, pp. 769-779, July 1993.

methods	total delay		major hardware of arithmetic units
	VR	AC	
[14]	$3n$	-	$3n \times \text{SDA} + 2n \times \text{CSA}$
[15]	$4.5n$	$4.5n$	$3.75n \times \text{SDA}$
[16]	$3.5n$	$4n$	$3n \times \text{SDA} + n \times \text{ABS}$
ours	$3n$	$3n$	$3n \times \text{SDA} + 2n \times \text{CSA}$

Table 1: Comparison of several redundant constant-factor CORDIC methods in pipelined (unfolded) digit-parallel implementation.

Processor Architecture	digit-serial	digit-parallel
cycle time	6.7 ns	14 ns
latency	81 cycles	29 cycles
throughput (ns /sample)	80	14
internal word length	12	12
core size ($\mu m \times \mu m$)	2700 × 2600	3500 × 3800
power dissipation (at 50 MHz)	1 W	2.8 W

Table 2: Chip information of the digit-serial and digit-parallel redundant CORDIC processors.

b	n	vector rotation				angle calculation	
		b_{sim}	b_r	b_a	b_{eff}	b_{sim}	$b_{eff} = b_a$
8	9	5.75	4.89	7.33	4.63	6.35	4.92
	10	5.77	4.81	7.87	4.63	6.35	4.71
	11	5.75	4.73	8.25	4.59	6.45	4.71
	12	5.89	4.65	8.48	4.54	6.35	4.59
	13	5.87	4.58	8.61	4.49	6.42	4.47
9	9	6.61	5.89	7.57	5.47	6.90	5.84
	10	6.79	5.81	8.24	5.54	7.03	5.84
	11	6.83	5.72	8.75	5.53	7.03	5.76
	12	6.71	5.65	9.28	5.50	7.45	5.66
	13	6.88	5.58	9.30	5.45	6.79	5.53
10	9	7.57	6.89	7.81	6.24	7.68	6.55
	10	7.73	6.81	8.64	6.40	7.93	6.69
	11	7.81	6.73	9.35	6.46	8.01	6.69
	12	7.84	6.65	9.91	6.46	7.97	6.62
	13	7.71	6.58	10.29	6.40	7.93	6.52
11	9	7.67	7.89	7.93	6.85	8.01	7.21
	10	8.37	7.81	8.87	7.16	8.38	7.56
	11	8.65	7.73	9.75	7.32	8.77	7.70
	12	8.70	7.65	10.54	7.38	8.77	7.70
	13	8.59	7.58	11.18	7.38	8.78	7.63
12	9	8.07	8.89	7.96	7.26	8.03	7.50
	10	8.77	8.81	8.92	7.74	8.70	8.03
	11	9.25	8.72	9.84	8.03	9.14	8.33
	12	9.47	8.65	10.69	8.17	9.30	8.45
	13	9.58	8.58	11.43	8.22	9.34	8.45
$K \simeq (1 - 2 \cdot 2^{-3} - 2^{-6})(1 + 2 \cdot 2^{-9} - 2^{-18})$, $b_s = 11.84$.							

Table 3: The external bit accuracy (b_{sim} or b_{eff}) vs. the number of unscaled CORDIC iterations (n) and the internal bit accuracy (b).

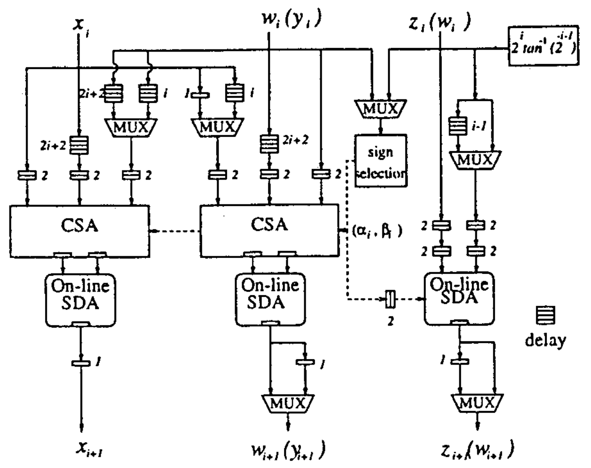

Figure 1: Implementation of the i-th pipelined stage ($i \le n/2$) for the redundant digit-serial 2-D Householder CORDIC processor. Each wire corresponds to a sign digit represented by two binary bits.

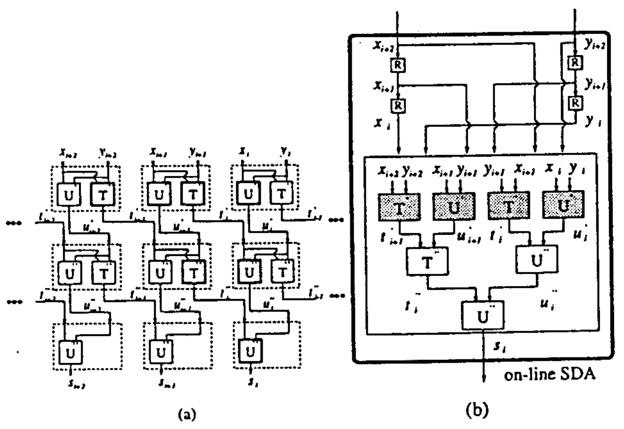

Figure 2: (a) The structure of an SD adder. (b) The implementation of an on-line SD adder.