

Hybrid CORDIC Algorithms

Shaoyun Wang, Vincenzo Piuri, *Senior Member, IEEE*,
and Earl E. Swartzlander, Jr., *Fellow, IEEE*

Abstract—Each coordinate rotation digital computer iteration selects the rotation direction by analyzing the results of the previous iteration. In this paper, we introduce two arctangent radices and show that about 2/3 of the rotation directions can be derived in parallel without any error. Some architectures exploiting these strategies are proposed.

Index Terms—CORDIC algorithm, elementary function, Givens transformation, planary rotator, CORDIC architecture.

1 INTRODUCTION

THE COordinate Rotation Digital Computer (CORDIC) algorithm is a well-known and widely-studied iterative technique [1], [2] for evaluating many basic arithmetic operations and mathematical functions. It is a bit-recursive implementation of the forward and backward Givens rotations [1], [2]. The basic CORDIC iteration equations at the i th step are [2]:

$$\begin{cases} x_{i+1} = k_i(x_i - m\sigma_i 2^{-S(m,i)} y_i) \\ y_{i+1} = k_i(y_i + \sigma_i 2^{-S(m,i)} x_i) \\ z_{i+1} = z_i - \sigma_i \alpha_{1,i} \end{cases} \quad (1)$$

where the iterations are performed for $\forall i, i = 0, 1, 2, \dots, N-1$. The *coordinate parameter* m identifies the coordinate system type (namely, circular, linear, and hyperbolic coordinates for m equal to 1, 0, and -1 , respectively). The *rotation direction* σ_i for rotation is $\sigma_i = -\text{sign}(z_i)$, while, for vectoring, it is $\sigma_i = -\text{sign}(x_i y_i)$. For $m = 1$, the *shift sequence* $S(1, i)$ is defined by $S(1, i) = 0, 1, 2, 3, 4, 5, \dots$. The *rotation angle* $\alpha_{1,i}$ is given by $\alpha_{1,i} = \arctan 2^{-S(1,i)} = \arctan 2^{-i}$. For each rotation performed, there is a *scale factor* $k_{1,i}$ which corrects the amplification introduced by the linearized "rotation" in the x and y coordinates. With the above assumptions, $k_i = \sqrt{1 + \sigma_i^2 2^{-2i}}$ for the i th iteration. After all iterations, the total scale factor is $K = \prod_{i=0}^{N-1} k_i = \prod_{i=0}^{N-1} \sqrt{1 + \sigma_i^2 2^{-2i}}$.

The functions directly computed by these CORDIC iteration equations, according to the selected value of the coordinate parameter, are summarized in Table 1. With the appropriate initial values of x , y , and z , we can compute various elementary functions.

TABLE 1
OUTPUTS OF THE CORDIC ALGORITHM

Coordinate System	rotation $z_n \rightarrow 0$	vectoring $y_n \rightarrow 0$
Circular $m = 1$	$x_n = \frac{1}{k_1}(x \cos z - y \sin z)$ $y_n = \frac{1}{k_1}(y \cos z + x \sin z)$	$x_n = \frac{1}{k_1}(x^2 + y^2)^{\frac{1}{2}}$ $z_n = z - \tan^{-1}\left(\frac{y}{x}\right)$
Linear $m = 0$	$x_n = x$ $y_n = y + x \times z$	$x_n = x$ $z_n = z + \frac{y}{x}$
Hyperbolic $m = -1$	$x_n = \frac{1}{k_1}(x \cosh z + y \sinh z)$ $y_n = \frac{1}{k_1}(y \cosh z + x \sinh z)$	$x_n = \frac{1}{k_1}(x^2 - y^2)^{\frac{1}{2}}$ $z_n = z + \tanh^{-1}\left(\frac{y}{x}\right)$

Many papers have been published on the design of advanced CORDIC processors to enhance the performance or to reduce the circuit complexity [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16]. The system latency has been reduced or the throughput has been increased by adopting general digital design techniques [4], [5], [6], [7]. An approach to anticipate the direction of each CORDIC rotation with some possible limited error has also been introduced [5]. The error in the result can be avoided either by appending rotations [8] or by limiting the input range [9]. Null rotations are used to enhance the convergence speed of the algorithm with the side effect of a variable scale factor, depending on the specific sequence of rotations actually applied [10], [11]. If null rotation steps are obtained by using micro-rotations [12], a constant scale factor is preserved, but the number of basic micro-iterations is twice the number of conventional iterations.

In this paper, we analyze the parallelism which is intrinsic in the CORDIC algorithm for the rotation mode only. For simplicity's sake, we consider only the circular coordinates; similar schemes have been developed also for linear and hyperbolic coordinates [13]. As in many applications requiring function generation, we suppose that the rotation angle is given at run time in the standard binary representation, i.e., it is not the result of a previous vectoring operation. We generalize the approach of [5] by abstracting from a specific architecture and from a single optimization goal. In Section 2, we show that, in a CORDIC processor having significant word size equal to N (not including the sign bit),

- S. Wang and E.E. Swartzlander, Jr., are with the Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX 78712. E-mail: e.swartzlander@computer.org.
- V. Piuri is with the Department of Electronics and Information, Politecnico di Milano, 20133 Milano, Italy.

Manuscript received 13 Oct. 1995; revised 21 July 1996.

For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number 105636.

the rotation directions from iteration $N = \left\lceil \frac{n - \log_2 3}{3} \right\rceil$ on can be derived in parallel for the rotation mode without any error. This result is exploited by introducing the hybrid radix sets and the Hybrid CORDIC Algorithms. In Section 3, we derive families of architectures providing alternatives to the designer for a better trade-off analysis between circuit complexity and performance.

2 THE HYBRID CORDIC ALGORITHMS

The essence of the CORDIC algorithm is a representation of an arbitrary angle by using a set of constant angles $\{\alpha_i\}$ instead of a normal radix. This set is acting like a radix. Therefore, we may choose another set to represent the same angle just like what we do in representing numbers with different radices.

The initial angle θ is represented by a set of arctangent constants, called Arc Tangent Radix (ATR) [1]. For the circular coordinates, the radix set is:

$$\{\alpha_0, \alpha_1, \dots, \alpha_{N-1}\} = \{\arctan 2^0, \arctan 2^{-1}, \dots, \arctan 2^{-N+1}\}, \quad (2)$$

which satisfies the CORDIC convergence theorem [2]:

$$\alpha_i - \sum_{j=i+1}^{N-1} \alpha_j < \alpha_{N-1}. \quad (3)$$

In order to determine the actual value of σ_i for any i , all the previous iterations have to be computed before the evaluation of σ_i . This sequential dependence of the radix system induces two bottlenecks in the CORDIC algorithm with respect to parallelization: the generation of the rotation directions and the application of the computed rotations. This second operation cannot be parallelized completely since each rotation must be applied to the vector generated by the previous rotations. Some partial parallelization is possible by merging two conventional CORDIC iterations into the same cycle, but the accuracy is slightly affected [14]; merging cannot be extended to a higher number of conventional CORDIC iterations since the induced error becomes unacceptable. Also, in the low latency architecture in [5], the rotations are applied sequentially after parallel generation of the corresponding group of rotation directions. A completely parallel and error-free generation of the rotation directions is not possible. In fact, the rotation directions are generated in parallel only within the group of the associated CORDIC iterations; but one group can be generated only after the completion of the previous group.

In this paper, we show that the computation of the rotation directions can be partially parallelized without affecting the accuracy by introducing new hybrid radix systems.

The Hybrid Radix Sets

The circular ATR values approach the radix-2 coefficients progressively for increasing values of the CORDIC iteration index. In fact,

$$\lim_{k \rightarrow \infty} \frac{\tan 2^{-k}}{2^{-k}} = 1.$$

Figs. 1 and 2 compare the circular ATRs and the radix-2

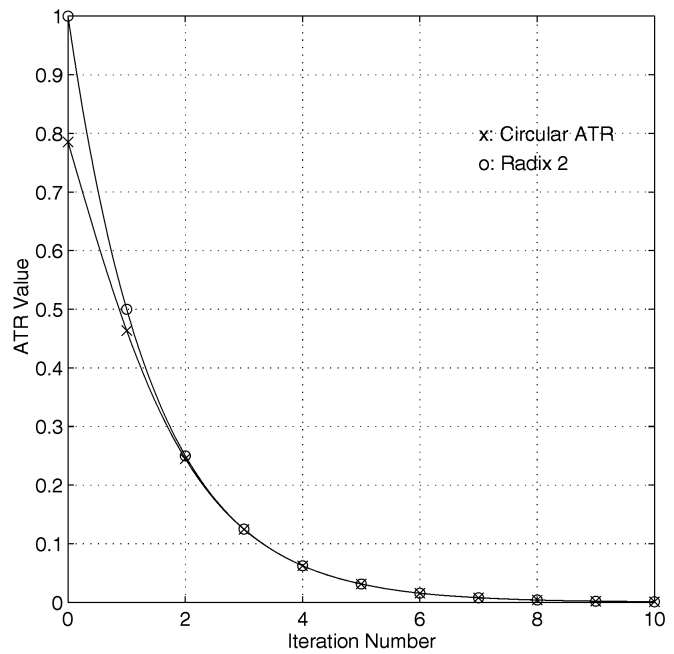


Fig. 1. Radix-2 and circular ATRs: the actual values.

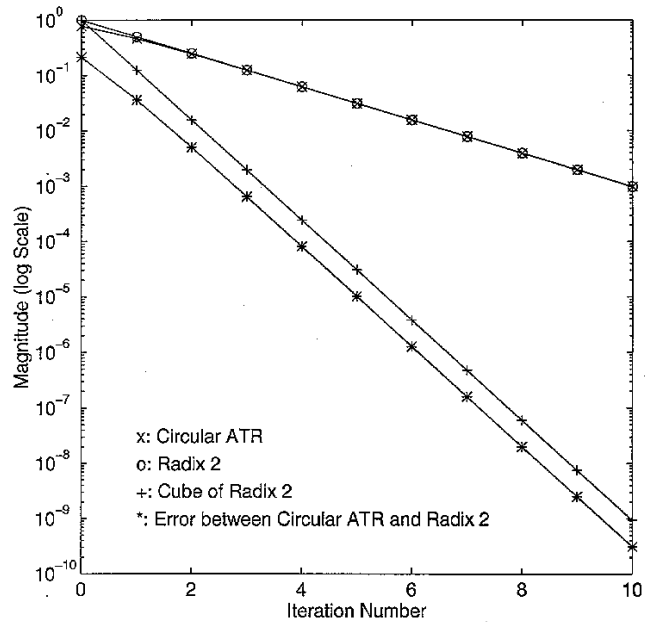


Fig. 2. Radix-2 and circular ATRs: the approximation error.

ATR. After the first few iterations, the error is less than the cube of the radix-2 ATR values. If the iteration index i is large enough, the error in assuming that radix-2 coefficients are the same as circular ATRs is negligible in the finite precision N -bit representation. Therefore, if radix-2 is adopted as the radix set for the least significant part of the rotation directions, instead of the circular ATRs, the least significant part of the rotation directions will have about the same precision as the corresponding conventional ATRs.

Based on this remark, we define the *Mixed-Hybrid Radix Set*:

Mixed-Hybrid Circular ATR =

$$\overbrace{\{\arctan 2^{-0}, \arctan 2^{-1}, \dots, \arctan 2^{-n+1}\}}^{\text{most significant part}}, \underbrace{\{2^{-n}, \dots, 2^{-N+1}\}}_{\text{least significant}}$$

The name emphasizes that it is obtained by mixing the conventional circular ATR (for the most significant part) with the conventional linear ATR (for the least significant part). The resulting algorithm is called the *Mixed-Hybrid CORDIC Algorithm*. The CORDIC iterations related to the most significant part are performed as in the conventional CORDIC algorithm. In order to avoid errors, each rotation direction can be evaluated only after the previous iteration has been completed. The iterations related to the least significant part can be simplified as in the linear coordinate system. All rotation directions associated with such iterations are available in parallel, since each of them is given by the corresponding bit of the residue rotation angle (1 and 0 identify positive and negative rotation, respectively). All the above rotations are applied sequentially without any error or can use merged CORDIC algorithms [5], [14] with some errors. It is worthwhile to note that application of the CORDIC iterations related to the most significant part of θ may change the value of the bits of the least significant part with respect to their initial value in θ . As a consequence, the parallel evaluation of the directions associated with the least significant part can be performed only after rotations of the most significant one.

To increase the performance, the computation of the most significant part must be completely separated from the least significant one while preserving full accuracy. To achieve this goal, we partition the binary representation of $\theta = \sum_{i=0}^{N-1} \theta_i 2^{-i}$ (θ_i being the bits in the N -bit binary representation) in two parts: the n most-significant bits and the remaining $N - n$ bits. Each of these groups of bits is the binary representation of an angle, namely:

$$\theta_H = \sum_{i=0}^{n-1} \theta_i 2^{-i} \quad \theta_L = \sum_{i=n}^{N-1} \theta_i 2^{-i}$$

The rotation θ is partitioned in two rotations: θ_H is the most relevant, while θ_L adjusts the position of the rotated vector to achieve the final expected (or nearby) position. θ_L always has zeros in the n most significant bits. The angle θ_H is completely independent from θ_L only if it does not change the $N - n$ least significant bits of the residue rotation angle with respect to θ_L . To achieve this goal, the exact rotation θ_H must be performed separately from θ_L . To avoid a possible increase in the number of the rotation iterations and to speed up the whole algorithm, we modify the radix set and the traditional CORDIC algorithm by assuming that the whole rotation θ_H is performed in only one iteration. This is not an actual limit since, when n is not too large, the iteration can be realized by using a ROM-based architecture.

Next, we introduce the *Partitioned-Hybrid Radix Set* and the *Partitioned-Hybrid CORDIC Algorithm*:

Partitioned-Hybrid Circular ATR =

$$\overbrace{\{\arctan 2^{-n+1}\}}^{\text{most significant part}}, \underbrace{\{2^{-n}, \dots, 2^{-N+1}\}}_{\text{least significant part}}$$

The least significant part is still obtained by approximating the circular ATR by the radix-2. The rotation directions for this part are, as usual, given by $\sigma_i = \{-1, 1\}$ being $i = n, n + 1, \dots, N - 1$. Their parallel generation and their application are performed as in the Mixed-Hybrid case. The most significant part is compressed into only one radix to formally describe the single iteration for θ_H . For σ_{n-1} , the definition of rotation direction is extended to represent any possible angle θ_H by compressing the circular ATR iterations from the initial one to the $(n - 1)$ th one, without modifying θ_L . Formally, it is:

$$\sigma_{n-1} = \frac{\theta_H}{\arctan 2^{-n+1}} = \frac{\sum_{i=0}^{n-1} \theta_i 2^{-i}}{\arctan 2^{-n+1}}$$

The rotation angle θ_H for this preliminary rotation is directly available from θ . Its application cannot be performed by using (1), since more than one angle value is actually taken into account. The results must be generated—directly and exactly—from the value of θ_H , e.g., by using such an angle to select x_{n-1} and y_{n-1} in a lookup table, including the scale factor.

For both of the Hybrid CORDIC Algorithms, convergence is guaranteed, since they are correct with respect to the condition imposed by (3) for the circular coordinates or similar for the linear case, as stated by Theorem 1.

THEOREM 1. *The Hybrid CORDIC Algorithms satisfy the CORDIC convergence theorem.*

PROOF. In the Hybrid CORDIC Algorithms, the CORDIC convergence theorem is satisfied by the angles α_i having $i \geq n$ since they coincide with the conventional linear ATRs.

Consider now the angles having $i < n$. In the conventional circular CORDIC Algorithm, (3) is:

$$A = \arctan 2^{-i} - \sum_{j=i+1}^{n-1} \arctan 2^{-j} - \sum_{j=n}^{N-1} \arctan 2^{-j} < \arctan 2^{-N+1} = B.$$

Since $\arctan 2^{-j} < 2^{-j}$ (for any $j \geq 0$), it is:

$$\arctan 2^{-i} - \sum_{j=i+1}^{n-1} \arctan 2^{-j} - \sum_{j=n}^{N-1} 2^{-j} < A < B < 2^{-N+1}.$$

The first and the last terms of the inequality are the terms of (3) for the angles of the most significant part in the Mixed-Hybrid CORDIC Algorithm; as a consequence, this algorithm satisfies the CORDIC convergence theorem.

The most significant part of the Partitioned-Hybrid Radix set is a complete and exact rotation by the angle θ_H performed in one step: As a consequence, the algorithm converges to the expected angle θ if it is able to converge to the angle $\theta_L = \theta - \theta_H$, i.e., the angle related to the sequence of iterations for $i \geq n$. Since we have shown above that these iterations satisfy the convergence theorem, the whole Partitioned-Hybrid

CORDIC Algorithm satisfies the CORDIC convergence theorem. \square

Partitioning the Radix Set

For both of the Hybrid Radix sets, we need to identify the value of n that preserves full accuracy. Here, we show that they are identical to the corresponding radix-2 coefficients when $n \approx \frac{1}{3}N$.

LEMMA 1. *In an N -bit fixed-point CORDIC processor, the smallest index i^* , which identifies a circular ATR α_{i^*} having the same N -bit representation as its trigonometric tangent (absolute error $\leq 2^{-N}$) is*

$$i^* = \left\lceil \frac{N - \log_2 3}{3} \right\rceil.$$

PROOF. Consider the circular ATR $\alpha_i = \arctan 2^{-i}$. The error ϵ_i introduced because of assuming such an angle identical to its trigonometric tangent, is

$$\epsilon_i = 2^{-i} - \alpha_i = 2^{-i} - \arctan 2^{-i}.$$

By developing the arctan function in its Taylor series, we have:

$$\begin{aligned} \epsilon_i &= 2^{-i} - \left[2^{-i} - \frac{1}{3}(2^{-i})^3 + \frac{1}{5}(2^{-i})^5 - \dots \right] \\ &= \frac{1}{3}(2^{-i})^3 - \frac{1}{5}(2^{-i})^5 + \dots \end{aligned}$$

Obviously, we have,

$$\frac{1}{3}2^{-3i} > \epsilon_i = \frac{1}{3}2^{-3i} - \frac{1}{5}2^{-5i} + \dots$$

The smallest number that can be represented by N -bit fixed-point number is 2^{-N+1} , when the fractional part is composed of $N-1$ bits. Since we are looking for an ATR index i which is identical to its tangent, the error induced by our approximation is required to be smaller than the smallest angle representable in the processor word. Therefore, we require that:

$$2^{-N} \geq \frac{1}{3}2^{-3i} > \epsilon_i = \frac{1}{3}2^{-3i} - \frac{1}{5}2^{-5i} + \dots$$

By solving with respect to i :

$$\begin{aligned} \frac{1}{3}2^{-3i} &\leq 2^{-N} \\ 3i + \log_2 3 &\geq N \\ i &\geq \frac{N - \log_2 3}{3} \end{aligned}$$

The minimum value i^* of the index i that satisfies this relationship is $i^* = \left\lceil \frac{N - \log_2 3}{3} \right\rceil$. \square

THEOREM 2. *In an N -bit fixed-point CORDIC processor for the circular coordinate system, the rotation directions that cannot be generated in parallel in the Hybrid CORDIC Algorithms without any errors, comparing to the conventional CORDIC algorithm, are the first n iterations:*

$$n = \left\lceil \frac{N - \log_2 3}{3} \right\rceil.$$

PROOF. According to Lemma 1, in the conventional CORDIC algorithm for the circular coordinate system, all ATR α_i having index i greater or equal to $\left\lceil \frac{N - \log_2 3}{3} \right\rceil$

can be replaced by the corresponding linear ATR 2^{-i} without affecting the accuracy, being identical to the representation in the N -bit processor word.

The ATRs of the least significant part are now identical to corresponding radices in the linear case. Therefore, we can apply the standard technique for parallel generation of the rotation directions for the ATRs of the least significant part only.

Consider now the CORDIC iterations starting from the initial one ($i=0$) to the one identified by the index $\left\lceil \frac{N - \log_2 3}{3} \right\rceil - 1$. Since the ATRs corresponding to these iterations cannot be replaced by the corresponding linear ATR 2^{-i} without affecting the accuracy, the related $\left\lceil \frac{N - \log_2 3}{3} \right\rceil$ rotation directions cannot be generated in parallel. \square

About $\frac{1}{3}N$ iterations must be computed in a strictly sequential way, both for the direction generation and the rotation, without affecting accuracy. If a little error is acceptable, we can apply the solutions proposed in [5] and [14] to this first set of iterations. The subsequent rotation directions can be generated in parallel since they coincide with the representation of the residue rotation angle remaining after the first sequential CORDIC iterations.

3 ARCHITECTURES FOR THE HYBRID CORDIC ALGORITHMS

The previous analysis, as well as the extension to the hyperbolic coordinate system [13], can be exploited to design efficient architectures for trigonometric and transcendental function generators. The architectures for these CORDIC algorithms consist of cascading two CORDIC subprocessors: one for each ATR subset. This obviously leads to a higher circuit complexity, but the latency and the throughput may be enhanced.

The architectural approach for the case of the Mixed-Hybrid CORDIC algorithm is given in Fig. 3. The first subprocessor uses the whole rotation angle θ and the initial vector coordinates x^* and y^* to generate z_m , x_m , and y_m at the end of the first n iterations. The rotation directions are generated sequentially, as in the conventional CORDIC algorithm, or predicted in parallel, with an error in the evaluation of the angle [5]. At the end of these iterations, the most significant part of z_n is zero. The second subprocessor operates starting from z_m , x_m , and y_m to generate x_N and y_N at the end of the remaining $N-n$ iterations.

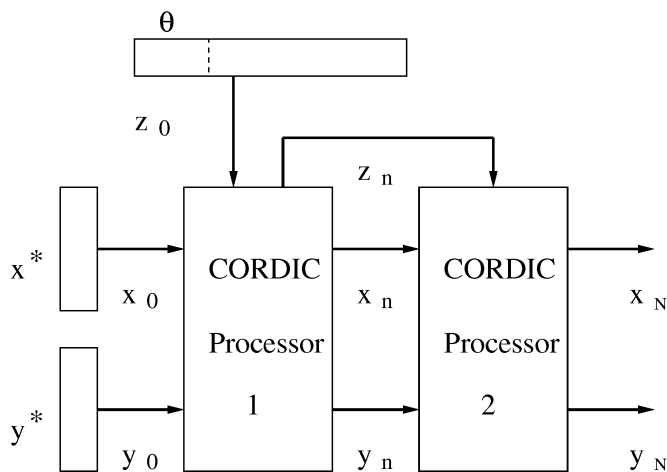


Fig. 3. The architecture for Mixed-Hybrid CORDIC algorithms.

A similar architecture can be designed for the case of the Partitioned-Hybrid CORDIC algorithm, as shown in Fig. 4. The first subprocessor operates a rotation of θ_H on x^* and y^* and generates x_n and y_n . The second subprocessor receives these intermediate rotated coordinates and applies the rotation θ_L to generate the final x_N and y_N .

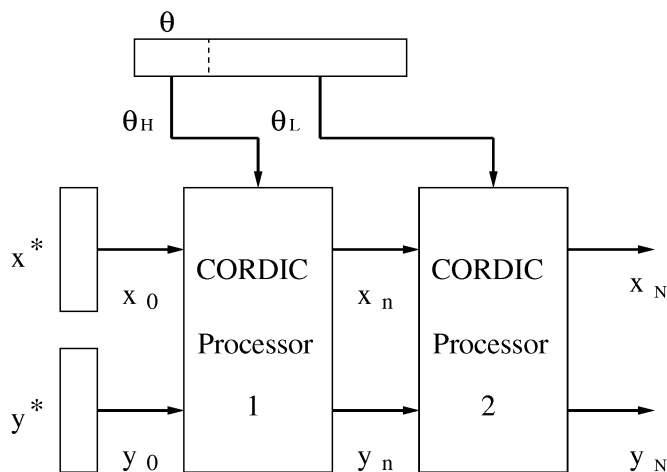


Fig. 4. The architecture for Partitioned-Hybrid CORDIC algorithms.

Consider the implementation of the first subprocessor working on the most significant part of the ATR set and performing the first n CORDIC iterations. The simplest approach consists of a ROM for the whole subprocessor, where the most significant part of θ is used as an address. This solution is feasible in terms of circuit complexity only for relatively small values of n , as occurs in several applications. The latency of the whole CORDIC processor is reduced to about 67 percent of the latency of a traditional processor performing N iterations. Alternative solutions can be envisioned for the Mixed-Hybrid CORDIC Algorithm by using a conventional CORDIC processor based on a sequential machine, a combinatoric (possibly pipelined) CORDIC processor, or combined CORDIC processors merging several iterations in a single cycle with possible ROM-based prediction of the rotation directions [15]. This

last class of architectures is an intermediate solution between the completely sequential and the purely combinatoric ones: It allows us to save some circuit complexity with respect to the combinatoric case, while achieving higher performance with respect to the sequential machine at a higher circuit complexity. In [13], we reported the detailed evaluation of these structures.

It is not realistic that we put the initial vector x_0 and y_0 into ROM when this architecture is used for planary rotations. However, we do have choices of when to do the multiplication. The first choice is to finish the evaluation of the sin and cos functions and do a multiplication, just like what we do with the scale factor. On the other hand, we can do the multiplication of the outputs of the ROM with x_0 and y_0 right after we get the ROM output. Either way, the multiplication cannot be eliminated. Of course, if a carry save adder is used, only one full adder delay is added in either case.

Consider now the second subprocessor. The ROM-based approach is never feasible since the ROM is unrealizable. Suitable solutions are variants of the pipelined and combined architectures [13], [15]. No rotation prediction is necessary since directions can be easily obtained by applying a trivial transformation to the angle representation. Pipelined and combined structures degenerate to a purely combinatoric circuit [5] when $N - n$ CORDIC iterations are performed in a single pipeline stage or clock cycle.

Even if the architectures for the Hybrid CORDIC Algorithms are very similar at a high abstraction level, their actual circuit complexity may be different, in particular, as Processor 1 is concerned. In fact, for example, the ROM realizing this processor in the Partitioned-Hybrid Algorithm must generate only x_n and y_n , while an additional memory bank is required in the Mixed case, since the residue angle z_n also needs to be evaluated. On the other hand, the Mixed case allows for a smaller implementation based on a sequential machine, even though the latency is increased and the throughput is reduced. The designer is thus provided with a wide range of alternative approaches so as to achieve an optimum trade-off between performance and circuit complexity for the specific application envisioned.

4 CONCLUSIONS

In this paper, we showed that, in a CORDIC processor for the circular coordinate system in the rotation mode, the rotation directions for approximately 2/3 of the iterations can be derived in parallel without introducing any error. We generalized the approach of [5] by extracting the arithmetic essence and presenting an abstract view of the partitioned CORDIC iterations. We also derived some families of architectures supporting the CORDIC computation at different levels of cost and performance.

ACKNOWLEDGMENTS

The authors are grateful to the anonymous referees for providing comments and suggestions that greatly improved this paper.

REFERENCES

- [1] J. Volder, "The CORDIC Trigonometric Computing Technique," *IRE Trans. Electronic Computers*, vol. 8, pp. 330-334, 1959.
- [2] J. Walther, "A Unified Algorithm for Elementary Functions," *Spring Joint Computer Conf. Proc.*, vol. 38, pp. 379-385, 1971.
- [3] J. Delosme, "VLSI Implementation of Rotations in Pseudo Euclidean Spaces," *Proc. Int'l Conf. Acoustic, Speech, and Signal Processing*, vol. 2, pp. 927-930, 1983.
- [4] R. Kundmund et al., "CORDIC Processor with Carry Save Architecture," *Proc. ESSCIRC '90*, pp. 193-196, 1990.
- [5] D. Timmermann, H. Hahn, and B. Hosticka, "Low Latency Time CORDIC Algorithms," *IEEE Trans. Computers*, vol. 41, pp. 1,010-1,015, 1992.
- [6] E. Depreettere, P. Dewilde, and R. Udo, "Pipelined CORDIC Architectures for Fast VLSI Filtering and Array Processing," *IEEE Trans. Signal Processing*, vol. 40, pp. 834-844, 1992.
- [7] H. Dawid and H. Meyr, "High Speed Bit-Level Pipelined Architectures for Redundant CORDIC Implementation," *Proc. Int'l Conf. Application Specific Array Processors*, pp. 358-372, 1992.
- [8] E. Antelo, J. Bruguera, J. Villalba, and E. Zapata, "Redundant CORDIC Rotator Based on Parallel Prediction," *Proc. Int'l Symp. Computer Arithmetic*, pp. 172-179, 1995.
- [9] P.W. Baker, "Suggestion for a Fast Binary Sine/Cosine Generator," *IEEE Trans. Computers*, vol. 25, pp. 1,134-1,136, 1976.
- [10] M. Ercegovic and T. Lang, "Redundant and On-Line CORDIC: Application to Matrix Triangularization and SVD," *IEEE Trans. Computers*, vol. 39, pp. 725-740, 1990.
- [11] S. Wang and E.E. Swartzlander Jr., "Critically Damped CORDIC Algorithm," *Proc. Midwest Symp. Circuits and Systems*, pp. 253-256, 1994.
- [12] 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, pp. 989-995, 1991.
- [13] S. Wang, V. Piuri, and E.E. Swartzlander Jr., "The Hybrid CORDIC Algorithms," Internal Report no. 96-058, Dept. of Electronics and Information, Politecnico di Milano, Milano, Italy, 1996.
- [14] S. Wang and E.E. Swartzlander Jr., "Merged CORDIC Algorithm," *Proc. Int'l Symp. Circuits and Systems*, pp. 1,988-1,991, 1995.
- [15] S. Wang, V. Piuri, and E.E. Swartzlander Jr., "A Unified View of CORDIC Processor Design," *Proc. Midwest Symp. Circuits and Systems*, 1996.
- [16] S. Wang, V. Piuri, and E.E. Swartzlander Jr., "Granularly-Pipelined CORDIC Processor for Sine and Cosine Generators," *Proc. Int'l Conf. Acoustic, Speech, and Signal Processing*, vol. 6, pp. 3,299-3,302, 1996.



Shaoyun Wang received the BS degree in physics and the MS degree in electronic engineering from Fudan University, Shanghai, People's Republic of China, in 1982 and 1985, respectively. He is currently a design engineer at Crystal Semiconductor Corporation, Austin, Texas, where he works on audio commercial products. His research interests are in application specific processing, computer arithmetic, CORDIC algorithms, signal processing architectures and implementations, and neural networks.

Previously, he was a lecturer in the Electronic Engineering Department at Fudan University, Shanghai, People's Republic of China, from 1985 to 1991, where he did research on real-time digital signal processing, distributed signal processing, and neural networks. He is pursuing the PhD degree in electrical engineering at the University of Texas at Austin with the support of a Crystal Semiconductor Fellowship.



Vincenzo Piuri obtained the DrIng degree in electronic engineering in 1984 and the PhD in computer engineering in 1989, from the Politecnico di Milano, Italy. He was an assistant professor in electrotechnics and computer science at the Politecnico di Milano and director of the Computer Center of the Department of Electronics and Information, Politecnico di Milano. Since 1992, he has been an associate professor in operating systems at the Politecnico di Milano and a consultant to several Italian public administrations for distributed information systems design.

His research interests include distributed and parallel computing systems, application-specific processing architectures, computer arithmetic, fault tolerance, data coding, neural network architectures, theory and applications of neural techniques for identification, prediction, and control of complex dynamic nonlinear systems. The original results of these researches have been published in more than 100 papers in book chapters, international journals, and proceedings of international conferences.

He served as special session organizer or program chair for several international conferences, as well as guest editor for special issues on neural technologies and architectures in the *International Journal on System Architectures*, the *International Journal on Microelectronic Systems Integration*, and the *International Journal on Computer-Aided Engineering*.

He is a senior member of the IEEE, IMACS, INNS, and AEI. He is secretary of the IEEE Computer Society, North Italy Chapter, and secretary of the IEEE Instrumentation and Measurement Society, North Italy Chapter. He is founding co-chair of the Technical Committee on Emergent Technologies of the IEEE Instrumentation and Measurement Society and founding co-chair of the Technical Committee on Fault Tolerance in VLSI Systems. He is member of the IMACS Technical Committee on Neural Networks.



Earl E. Swartzlander, Jr. is a professor of electrical and computer engineering at the University of Texas at Austin, where he holds the Schlumberger Centennial Chair in Engineering. His research interests are in application specific processing, the interaction between computer architecture and VLSI technology, and the history of calculators. He has made significant research contributions in computer arithmetic, VLSI development, and digital signal processor implementation. Previously, he was with TRW Defense and Space Systems from 1975 to 1990, where he held positions ranging from staff engineer to laboratory manager and, most recently, director of Independent Research and Development for the TRW Defense Systems Group.

He is the hardware area editor for *ACM Computer Reviews*, a subject area editor of the *Journal of Systems Architecture*, and editor of the Calculators column of the *IEEE Annals of the History of Computing*.

He obtained his doctorate in computer design with the support of a Howard Hughes Doctoral Fellowship. He is a fellow of the IEEE, an inaugural member of the IEEE Computer Society Golden Core, and is a registered professional engineer in Alabama, California, Colorado, and Texas. He has been honored as a Knight of the Imperial Russian Order of Saint John of Jerusalem Ecumenical Foundation (Knights of Malta), an Outstanding Electrical Engineer of Purdue University, a Distinguished Engineering Alumnus of Purdue University, and a Distinguished Engineering Alumnus of the University of Colorado.



本文献由“学霸图书馆-文献云下载”收集自网络，仅供学习交流使用。

学霸图书馆（www.xuebalib.com）是一个“整合众多图书馆数据库资源，提供一站式文献检索和下载服务”的24小时在线不限IP图书馆。

图书馆致力于便利、促进学习与科研，提供最强文献下载服务。

图书馆导航：

[图书馆首页](#) [文献云下载](#) [图书馆入口](#) [外文数据库大全](#) [疑难文献辅助工具](#)