A 400-MHz Processor for the Conversion of Rectangular to Polar Coordinates in 0.25-μm CMOS

David D. Hwang, *Student Member, IEEE*, Dengwei Fu, and Alan N. Willson, Jr., *Fellow, IEEE*

***Abstract*—This paper describes the architecture and IC implementation of a rectangular-to-polar coordinate converter for digital communication applications. The architecture core uses small lookup ROMs, fast multipliers, and a single angle-rotation stage. Area and latency are reduced in comparison with traditional methods. The processor, implemented in 0.25-μm five-metal CMOS, has 14-b in-phase and quadrature channel inputs and 15-b magnitude and phase channel outputs. The phase and magnitude calculations have a maximum error of 0.00024 (0.0078% of π) and 0.03 (1% of $2\sqrt{2}$), respectively. Computational latency is 19 cycles, and power dissipation is 470 mW at 2.5 V and 406 MHz (Mconversions/s).**

***Index Terms*—Application-specific integrated circuits, coordinate conversion, CORDIC, digital communication, synchronization.**

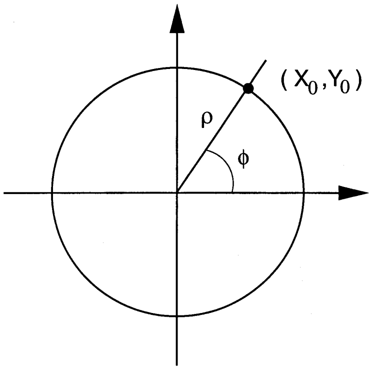

Fig. 1. Rectangular-to-polar conversion.

I. Introduction

MANY DIGITAL communication applications require the efficient conversion of rectangular to polar coordinates, i.e., a complex number represented as a vector with I and Q channels must be converted into polar representation (ρ, ϕ) using $\rho = \sqrt{X_0^2 + Y_0^2}$ and $\phi = \arctan(Y_0/X_0)$, as shown in Fig. 1. Often, only the phase must be precisely calculated from the input data; the magnitude, if required at all, is only used for coarse scaling.

The processor, along with a corresponding polar-to-rectangular processor [1], can implement the M-ary phase shift keying (PSK) receiver in [2], which requires high-precision wordlengths. (See also the applications mentioned in [3].) In CDMA and UMTS base-station applications, the processor could provide coordinate transformations for downconversion, as does the 24-b 95-MSamples/s rectangular-to-polar module embedded within [4]. The general wide-band PM/FM demodulators described in [5] require the extraction of the phase of a complex number. In the receivers discussed in [6] and [7], rapid phase extraction for clock and carrier synchronization is crucial, particularly if a latency-sensitive feedback loop is employed. The 16-b 50-MHz coordinate converter in [8] would be able to perform this computation with a latency of 18 computational cycles. Our chip could be overclocked eight times in such a system, providing an overall latency of only $\lceil 19/8 \rceil = 3$ computational cycles at the 50-MHz clock rate.

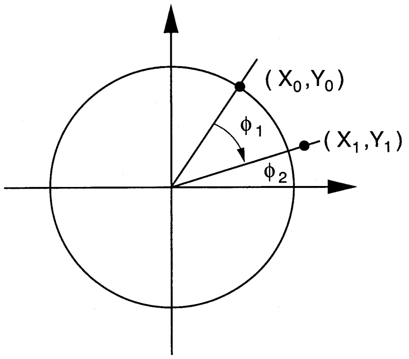

Fig. 2. Two-stage phase calculation.

II. Prior Art

In many of the target applications, the essential component of the algorithm is the extraction of the phase. There are several common implementations for such a phase extraction, which can be grouped into three categories: 1) ROM-based approaches; 2) multiplierless approaches with limited memory, such as CORDIC; and 3) modified CORDIC solutions.

The simplest method for complex value phase extraction employs a ROM lookup table [5] in which the X_0 and Y_0 signals access a ROM module which produces the phase ϕ. This method provides the lowest latency of computation—a single memory access—however, it is useful only for small-wordlength applications since ROM size increases exponentially $O(2^{2N})$ with input wordlength N. A quotient and ROM technique calculates the quotient $q = Y_0/X_0$ and uses it to index the ROM table

Manuscript received September 27, 2002; revised July 2, 2003. This work was supported by Analog Devices, Broadcom Corporation, GlobeSpan, and Rockwell Scientific under UC MICRO Grant 00-104, Grant 01-100, and Grant 02-085. The work of D. D. Hwang was supported in part by the Fannie and John Hertz Foundation.

D. D. Hwang is with the Department of Electrical Engineering, University of California, Los Angeles, CA 90095 USA (e-mail: dhwang@icsl.ucla.edu).

D. Fu was with the Department of Electrical Engineering, University of California, Los Angeles, CA 90095 USA. He is now with NVIDIA Corporation, Santa Clara, CA 95050 USA (e-mail: dfu@nvidia.com).

A. N. Willson, Jr., is with the Integrated Circuits and Systems Laboratory, Department of Electrical Engineering, University of California, Los Angeles, CA 90095 USA (e-mail: willson@ee.ucla.edu).

Digital Object Identifier 10.1109/JSSC.2003.817588

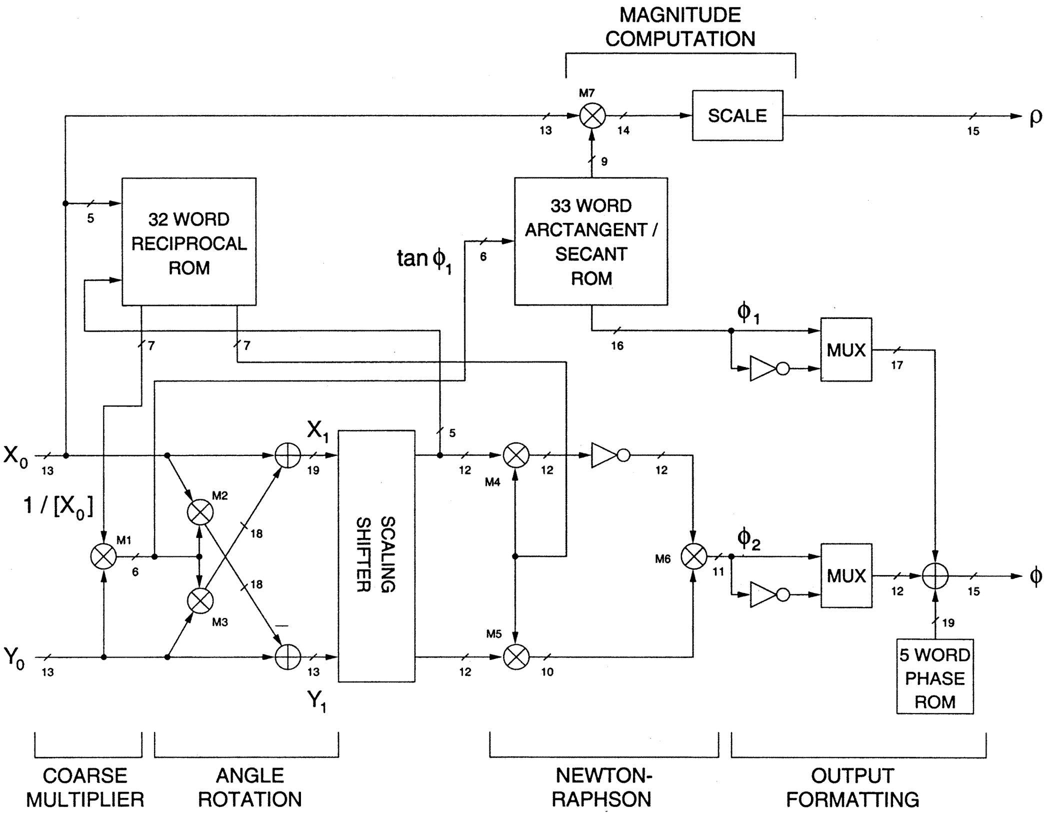

Fig. 3. Converter architecture.

which, in turn, produces ϕ [5]. This method lessens but still retains the complexity problem. Obviously, other techniques are needed for larger wordlength applications.

A technique suited for larger wordlength applications is a multiplierless approach that requires limited memory: CORDIC (Coordinate Rotation Digital Computer) [9], [10]. However, since an N-bit input requires approximately N stages, the latency in CORDIC increases linearly with required precision. In addition, being inherently sequential, with the input of one stage depending on the result from the previous stage, parallelizing CORDIC-based systems can be difficult. Wordlength considerations also affect CORDIC structures. Each CORDIC stage truncates the intermediate (X_i, Y_i) values to a fixed wordlength, causing truncation errors to propagate to the output. To insure small errors, internal wordlengths must be larger than the output precision. In the CORDIC-based system of [11], 20-b internal wordlengths are required for a 14-b output, 14-stage design. Though CORDIC's multiplierless feature reduces complexity, both latency and internal wordlengths increase with increased precision requirements. Moreover, multiplier techniques such as Booth-encoding cannot conveniently be employed.

Hybrid approaches which incorporate modified CORDIC structures along with multipliers and/or memory attempt to strike a compromise between complexity and latency. An interpolation method of [12] divides an N-rotation CORDIC into two computational stages: a coarse stage replaces $N-k$ CORDIC rotations with a lookup table and two multipliers, while a fine stage uses CORDIC for the remaining k rotations. A Taylor series method (see also [12]) implements a coarse stage with CORDIC for the first $k < N$ rotations and a fine stage computing the residual angle using Taylor series approximations as a single rotation via two multiplications and without a trigonometric lookup table. Another two-stage approach is presented in [13], where the coarse stage uses CORDIC to perform $k < N$ rotations, then the remaining angle $\arctan(Y_k/X_k)$ is approximated by its first-order Taylor series Y_k/X_k, computed either by convergence division or by further CORDIC rotations and a ROM-based division. Similar hybrid techniques are presented in [14]. In modified CORDIC approaches, although the number of rotations is reduced, latency still grows with N (and indirectly, k) due to the dependence on CORDIC to perform $k < N$ rotations.

III. Algorithm and Architecture

The method described here implements a two-stage phase computation using a fixed number of rotations (one) regardless of precision requirements. The coarse stage uses two ROMs and a multiplier to calculate ϕ_1, an approximation of ϕ. It starts by rotating (X_0, Y_0) toward the X axis by the computed angle ϕ_1 using a two-multiplier butterfly structure, resulting in the vector (X_1, Y_1) of Fig. 2. At this point, the residual angle $\phi_2 = \arctan(Y_1/X_1)$ can be approximated by Y_1/X_1. The fine stage performs this division to obtain ϕ_2 via a reciprocal ROM and multiplier followed by a Newton–Rhapson sharpening stage using two multipliers. The final value $\phi = \phi_1 + \phi_2$. (A magnitude calculation stage operates in parallel with the fine phase computation stage.) The structure of the core processor, including all stages except for the input formatting stage, is shown in Fig. 3.

Rigorous analysis [15] demonstrates that the ROM sizes are on the order of $2^{(N+1)/3}$ words (32 words for $N = 14$). The six multipliers are not $N \times N$ multipliers with full output precision, but rather vary in size from $[(N-2)/3+2] \times [N-1]$ to $[N-4] \times [2(N-2)/3+4]$ (6×13 to 10×12 for $N = 14$) with truncation/rounding operations performed at the multiplier outputs. The proposed architecture's scaling properties should also be noted. Since the ROMs in the structure are on the order of $2^{(N+1)/3}$ words, many of the input wordlengths and multiplier wordlengths increase with N by a factor of $\lceil (N-2)/3 \rceil$. For example, if the input wordlength increases by three bits, the ROM complexity would only increase by a factor of two. (In comparison, the complexity would increase by a factor of $2^6 = 64$ or $2^3 = 8$ in the prior art ROM-based and quotient-based methods, respectively.) Also, for the method presented here, many internal and multiplier wordlengths would increase by only one bit for a 3-b input wordlength increase. Traditional CORDIC would require a greater increase of wordlength to compensate for truncation errors; for example, three bits of additional internal wordlength are required (going from 20 b to 23 b) as precision increases from $N = 14$ to $N = 17$ in the CORDIC-like circuit of [11]. CORDIC would also require three additional rotation stages, increasing latency. Modified CORDIC algorithms often require an increased number of rotation stages as wordlength precision increases.

The proposed algorithm can be considered a modified CORDIC-like solution in that it possesses rotations, memory, and multipliers. It is unique in that it requires one fixed rotation using fast multipliers and memory rather than the N multiplierless rotations of pure CORDIC or the $k < N$ multiplierless rotations preceded/followed by fast multipliers (coupled with memory) of the modified CORDIC solutions. Having a fixed number of stages, latency can be reduced regardless of the required precision by using efficient multiplier structures (modified Booth). Each of the architecture stages is described in further detail in the next sections.

A. Coarse Phase Computation Stage

This stage produces a coarse estimate ϕ_1 of ϕ. Initially, the five MSBs of X_0, denoted $[X_0]$, are used to access a 32-word ROM which performs reciprocation, producing $1/[X_0]$. The tangent of ϕ_1 is calculated by multiplying this value by Y_0 and rounding to five fractional bits, obtaining $\tan\phi_1 = Y_0 \times 1/[X_0]$. Finally, $\tan\phi_1$ is used to access a 33-word arctangent ROM which produces the coarse phase estimate ϕ_1.

B. Fine Phase Computation Stage

This stage is used to calculate the residual angle $\phi_2 = \phi - \phi_1$. It begins by rotating the vector (X_0, Y_0) clockwise by ϕ_1 to produce the vector (X_1, Y_1) lying ϕ_2 radians above the X axis, as shown in Fig. 2. (Note, however, that ϕ_2 can be positive or negative.) This butterfly rotation can be described as

$$\begin{bmatrix} X_1 \\ Y_1 \end{bmatrix} = \begin{bmatrix} 1 & \tan\phi_1 \\ -\tan\phi_1 & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \end{bmatrix}. \qquad (1)$$

We note that the value of $\tan\phi_1$ has been calculated in the coarse stage. The butterfly stage also operates in parallel with the arctangent ROM of the coarse stage, reducing latency. Once (X_1, Y_1) is calculated, the fine stage proceeds to calculate the value of $\phi_2 = \arctan(Y_1/X_1)$. It can be shown that if $|Y_1/X_1| < 2^{(N-2)/3}$ then $\phi_2 \approx Y_1/X_1$ with sufficiently minimal loss of accuracy [15].

Y_1/X_1 is calculated by using an approximation term multiplied by a Newton–Raphson sharpening term. The approximation term is obtained by using the same reciprocal ROM technique (and, if desired, the same reciprocal ROM itself) as in the coarse stage and is represented as $Y_1 \times 1/[X_1]$. However, the approximation term does not meet the accuracy requirements of the fine stage. To sharpen the accuracy, one iteration step of the Newton–Raphson method [16] is applied. Using $1/[X_1]$ as an initial guess for the zero of the function $f(x) \equiv 1/x - X_1$, a sufficiently accurate result is obtained with one iteration of $x_{k+1} = x_k - f(x_k)/f'(x_k)$. Hence, the fine angle is computed as

$$\phi_2 = \left(Y_1 \times \frac{1}{[X_1]}\right) \times \left(2 - X_1 \times \frac{1}{[X_1]}\right) \qquad (2)$$

where $(Y_1 \times 1/[X_1])$ is the approximation term and $(2 - X_1 \times 1/[X_1])$ is the sharpening term. In terms of architecture, the reciprocal ROM output $1/[X_1]$ is multiplied by both Y_1 (to form the approximation term) and X_1. The sharpening term is calculated by simply inverting the bits of $X_1 \times 1/[X_1]$ (i.e., using its one's complement) which approximates $2 - X_1 \times 1/[X_1]$ with an error much less than the final rounding of ϕ. The approximation term is multiplied by the sharpening term to produce ϕ_2.

C. Input Formatting and Quadrant Mapping Stages

Before the phase computation stages, an input formatting stage is employed to map the original circuit inputs into values suitable for the conversion algorithm [15]. For highest phase accuracy, the inputs to the IC, X and Y, must be mapped to a corresponding X_0 and Y_0 within the first octant in the range $0 \le X_0 < 2$ and $0 \le Y_0 \le X_0$. After the phase computation stages, the quadrant mapping stage forms the sum $\phi = \phi_1 + \phi_2$ and remaps this phase to the original quadrant. The quadrant mapping stage uses two multiplexers as well as a five-word phase ROM which stores the rounded values of 0, π, $-\pi$, $\pi/2$, and $-\pi/2$, as well as an additional rounding bit.

D. Magnitude Computation Stage

The magnitude computation stage operates in parallel with the fine phase computation stage. The magnitude of a given complex number (X_0, Y_0) can be calculated as $\rho = X_0 \sec\phi$. Since the coarse angle ϕ_1 is an approximation of ϕ, similarly $X_0 \sec\phi_1$ approximates $X_0 \sec\phi$. Therefore, the arctangent ROM is expanded to store $\sec\phi_1$ as well as ϕ_1. In the magnitude computation stage, the value of X_0 is multiplied by the ROM output $\sec\phi_1$ to produce ρ. Since the original IC inputs X and Y were scaled, the value of ρ is also appropriately scaled. The final value of ρ has a maximum error of 0.03. For many applications, a high-precision magnitude ρ is not necessary. For those applications requiring higher precision, interpolation algorithms can be used in a modified design.

TABLE I
IC SPECIFICATIONS

Technology	0.25-μm, 5-metal, 2.5-V TSMC
Transistor Count	100 229
Core Area	0.484 mm^2
Package	84-Pin Kyocera Pin Grid Array
X, Y Input Wordlength	14 b
ρ, ϕ Output Wordlength	15 b
ϕ Maximum Error	< 0.00024 (0.0078% of π)
ρ Maximum Error	< 0.03 (1% of $2\sqrt{2}$)
Computational Latency	19 cycles
Maximum Frequency (2.5 V)	406 MHz
Power Dissipation (2.5 V)	470 mW
Maximum Frequency (1.8 V)	260 MHz
Power Dissipation (1.8 V)	140 mW

E. Hardware Optimization

Hardware optimization techniques are used to reduce area while retaining functionality. As shown in Fig. 3, there are three ROM modules in the system, having 33, 32, and 5 words. To implement each of the ROM modules would require the full-custom layout of each ROM, which would be costly in design time. However, since each ROM is small, each is easily implemented as a combinational logic circuit using standard cells. Internal wordlengths are also reduced. First, each bus is minimized in keeping within our error-bound analysis [15], producing the minimum wordlengths required for proper output accuracy. In addition, except the formatting stages, virtually every bus holds a nonnegative number. Taking advantage of this fact, these buses are encoded using positive instead of two's complement arithmetic, saving one bit on every bus. Also, we retain only information-bearing bits, dropping portions of buses known to be zero at all times.

IV. CHIP CHARACTERISTICS AND TEST RESULTS

The prototype IC [17] was implemented in a 0.25-μm five-metal 2.5-V TSMC CMOS technology using a library from Artisan Components. The IC inputs are 14-b two's complement X and Y in the range $-2 < X, Y < 2$, with two integer and twelve fractional bits. The IC outputs are 15-bit two's complement ρ and ϕ in the ranges $0 \leq \rho \leq 2\sqrt{2}$ and $-\pi \leq \phi \leq \pi$, each with three integer and twelve fractional bits. The circuit is packaged in an 84-pin Kyocera ceramic package. The core area is 0.484 mm^2 and consists of 100 229 transistors. The circuit was tested on a custom-designed PC board and employed built-in self-test (BIST) to verify the core processing speed and various random test vectors to verify functionality and accuracy. The IC operated at 2.5 V at a maximum frequency of 406 MHz (Mconversions/s), while dissipating 470 mW. The IC also operated at 1.8 V at a maximum frequency of 260 MHz (Mconversions/s), while dissipating 140 mW. The maximum error in the computation of ϕ was less than one output LSB, $2^{-12} = 0.000\,24$ $(0.0078\%$ of $\pi)$, and the maximum error in the computation of ρ was less than 0.03 (123 LSB, 1% of $2\sqrt{2}$). The total computational latency is 19 cycles. A summary of the test results and chip specifications is given in Table I. A photomicrograph of the prototype IC is shown in Fig. 4.

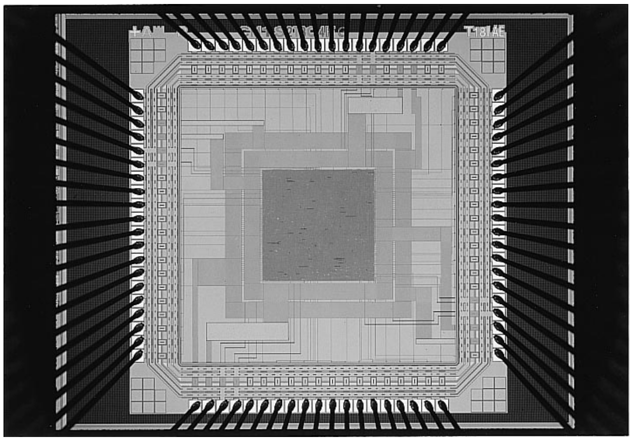

Fig. 4. IC photomicrograph.

V. CONCLUSION

An efficient rectangular-to-polar conversion algorithm has been introduced for digital communications applications. The conversion architecture is composed primarily of a two-stage phase computation and a parallel magnitude computation. A digital IC has been fabricated which implements the architecture in a 2.5-V 0.25-μm CMOS technology. As communications applications require increasing wordlengths and accuracy, the value of this algorithm may grow increasingly evident in comparison to ROM-based, traditional CORDIC, and modified CORDIC solutions.

REFERENCES

[1] D. Fu and A. N. Willson, Jr., "A high-speed processor for digital sine/cosine generation and angle rotation," in *Conf. Record 32nd Asilomar Conf. Signals, Systems, and Computers*, vol. 1, Nov. 1998, pp. 177–181.

[2] D. Critchlow, "The design and simulation of a modulatable direct digital synthesizer with non-iterative coordinate transformation and noise shaping filter," M.S. thesis, University of California, San Diego, 1989.

[3] Fairchild TMC2330A Datasheet, Oct. 2000.

[4] Intersil ISL5216 Datasheet, 2002.

[5] N. Boutin, "An arctangent type wideband PM/FM demodulator with improved performance," *IEEE Trans. Consumer Electron.*, vol. CE-38, pp. 5–9, Feb. 1992.

[6] M. P. Fitz and W. C. Lindsey, "Decision-directed burst-mode carrier synchronization techniques," *IEEE Trans. Commun.*, vol. 40, pp. 1644–1653, Oct. 1992.

[7] M. Andronico *et al.*, "A new algorithm for fast synchronization in a burst mode PSK demodulator," in *Proc. IEEE Int. Conf. Commun.*, vol. 3, June 1995, pp. 1641–1646.

[8] Logic Devices L2330 Datasheet, Sept. 2001.

[9] J. Volder, "The CORDIC trigonometric computing technique," *IRE Trans. Electron. Comput.*, vol. EC-8, pp. 330–334, Sept. 1959.

[10] A. Chen and S. Yang, "Reduced complexity CORDIC demodulator implementation for D-AMPS and digital IF-sampled receiver," in *Proc. IEEE Globecom*, vol. 3, Nov. 1998, pp. 1491–1496.

[11] A. Madisetti, A. Y. Kwentus, and A. N. Willson, Jr., "A 100-MHz 16-b direct digital frequency synthesizer with a 100-dBc spurious-free dynamic range," *IEEE J. Solid State Circuits*, vol. 34, pp. 1034–1043, Aug. 1999.

[12] H. M. Ahmed, "Efficient elementary function generation with multipliers," in *Proc. 9th Symp. Computer Arithmetic*, Sept. 1989, pp. 52–59.

[13] D. Timmermann, H. Hahn, and B. Hosticka, "Modified CORDIC algorithm with reduced iterations," *Electron. Lett.*, pp. 950–951, July 20, 1989.

[14] S. Wang, V. Piuri, and E. E. Swartzlander, “Hybrid CORDIC algorithms,” *IEEE Trans. Comput.*, vol. 46, pp. 1202–1207, Nov. 1997.

[15] D. Fu and A. N. Willson, Jr., “A high-speed processor for rectangular-to-polar conversion with applications in digital communications,” in *Proc. IEEE Globecom*, vol. 4, Dec. 1999, pp. 2172–2176.

[16] I. Koren, *Computer Arithmetic Algorithms*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[17] D. D. Hwang, D. Fu, and A. N. Willson, Jr., “A 400-MHz processor for the efficient conversion of rectangular to polar coordinates for digital communications applications,” in *Proc. IEEE Symp. VLSI Circuits*, June 2002, pp. 248–251.