IEEE Instrumentation and Measurement
Technology Conference
Budapest, Hungary, May 21–23, 2001

Design of a CORDIC Processor for Mixed-Signal A/D Conversion

M. B. Yeary
Department of Electrical Engineering
Texas A&M University, College Station, TX 77843-3128, USA
Phone: +1 979-862-1640, email: mbyeary@ee.tamu.edu

Abstract *– This paper proposes a new method which offers a high level of synchronization between a source, which is primarily digital, that generates a test signal and the ADC that will sample it. By using using a single clock to control the source, a clock divider may be used to derive a clock that will trigger an ADC at the appropriate times to produce a coherently sampled data set. Thus the timing of the waveform and the ADC will be accurately synchronized; moreover, since test time is a valuable commodity, a predictable number of clock cycles can be issued in order to generate a sampled data set. A computer simulation is given which fully characterizes the theoretical aspects of this paper. In addition, selected laboratory measurements are also given for discussion.*

Keywords *– A/D conversion, CORDIC, coherent*

I. INTRODUCTION

The problem of establishing a reliable source for various aspects of mixed signal testing has been prevalent for many years. One particular application in which a well defined clock source is important to synchronize a signal under test and it's accompanying ADC strobe is beat frequency testing [1],[2]. The problem typically stems from the use of analog methods to generate sinusoidal test signals. Digital frequency synthesizers offer advantages over analog methods since they possess lower phase noise, finer frequency resolution, and the ability to rapidly change the output frequency [3],[4]. Moreover, more sophisticated algorithms may be implemented digitally which would be very difficult to achieve with analog circuits [5].

To generate sinusoids, some authors choose to store the values of a sinusoid in ROM (read only memory), but the spectral purity depends on the number of stored values. Increasing the number of stored values, increases the power consumption and space requirements [6]. This method also suffers from the fact that addressing the ROM at high clock rates is troublesome [7]. Other methods to generate sinusoids digitally have also been developed. Fliege [8] proposed a digital oscillator, but one of the main limitations of this approach is the requirement frequent initialization. In his classic paper, Volder first introduced CORDIC algorithm in 1959 [9]. Since then, the CORDIC algorithm has proven to be a very reliable computational method in the area of digital computer arithmetic.

In this paper, a waveform generation technique is proposed that takes advantage of a CORDIC processor. This strategy is ideally suited for this application since the accuracy of the signal is a function of the number of digital building blocks, and not a function of analog devices such as capacitors, op-amps, etc. These items are known to have electrical properties that vary over time. The CORDIC processor only relies on delay elements, digital addition/subtraction circuits, and bit shifters in order to generate the sinusoidal signals [10][11].

II. CORDIC PROCESSOR DESIGN

To develop a digital system that will serve as a device that will serve as a tone generator, vector rotations will be considered first. Leithold [12] has suggested rotating the coordinates of an axis by α degrees using Equations (1) and (2), moreover, making use of this axis rotation has also served as a basis for digital filter design [13]. x' represents the rotated x-coordinate, y' represents the rotated y-coordinate.

$$x' = x\cos(\alpha) - y\sin(\alpha) \tag{1}$$

$$y' = x\sin(\alpha) + y\cos(\alpha) \tag{2}$$

Thus the vector, $\vec{v}$, may be rotated by an arbitrary angle, and the new coordinates of $\vec{v}'$ are summarized by the transformation

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

0-7803-6646-8/01/$10.00 ©2001 IEEE

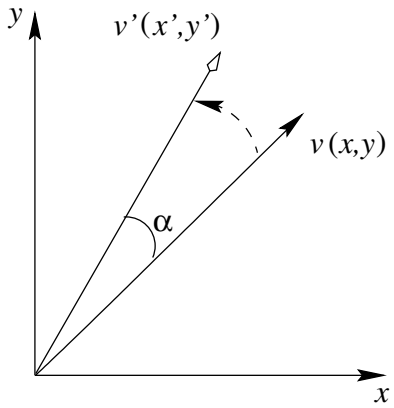

Fig. 1. Rotating a vector, $\vec{v}$, by α degrees in the counter-clockwise direction to produce $\vec{v}'$

$$= \cos(\alpha) \begin{bmatrix} 1 & -\tan(\alpha) \\ \tan(\alpha) & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (3)$$

By imposing the condition that $\tan(\alpha) = 2^{-i}$, then multiplication of x and y by $\tan(\alpha)$ reduces to a multiplierless hardware bit shift and add operation. Thus rotating a vector by $\tan^{-1}(2^{-i})$ degrees may accomplished in an efficient manner. Subsequently, $\alpha = \tan^{-1}(2^{-i})$, and the constant $\cos(\tan^{-1}(2^{-i}))$ may be stored in a lookup-table or hardwired. By allowing the variable i to assume integer values, α also assumes particular values. Some examples are as follows: $i = 0$, $\alpha = 45°$; $i = 1$, $\alpha = 26.5651°$; $i = 2$, $\alpha = 14.0362°$; $i = 3$, $\alpha = 7.1250°$; $i = 4$, $\alpha = 3.5763°$; etc.

Given a vector which is at an arbitrary angle, for instance $\theta = 60°$, as depicted in Figure 2, it is desired to compute $\sin(\theta)$. The first step is to successively rotate a vector in decreasingly small steps so that it will lie in the direction of the vector at angle θ. A vector, which initially approximates the vector pointing in the direction of the vector at angle θ, is assigned to lie on the x axis. Since a decreasing series of angles required to ensure convergence [10], the vector is rotated by $-90°$, as indicated by rotation "1" on the graph. Following this, the vector may be rotated by increments of $\pm\tan^{-1}(2^{-i})$ degrees. Choosing + or − is governed by whether or not the rotated vector is getting closer to the vector pointing in the θ direction. Therefore, the vector is rotated by $+45°$, then by $-26.5651°$, then by $+14.0362°$, then by $-7.1250°$, then by $+3.5763°$. After these five rotations, the approximating vector lies at an angle of $61.0776°$, which is reasonably close to $\theta = 60°$ for only five iterations.

The ability to rotate a vector is the basis of the CORDIC (COordinate Rotation Digital Computer) algorithm proposed by Volder [9] which can be used to compute trigonometric functions. Thus by scaling y' by $\cos(\tan^{-1}(2^{-i}))$ each time a rotation occurs, then an approximation for $\sin(\theta)$ occurs.

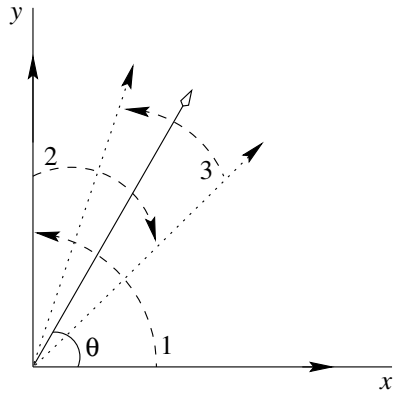

Fig. 2. Successive rotations of a vector are employed in the determination of $\sin(\theta)$. In this example, 3 rotations are depicted, but several more are needed for greater accuracy in the iterative process.

From the above discussion, it is possible to iteratively calculate the sine of an angle. The following equations describe this procedure, provided that y'_{i+1} is scaled by dividing by $\cos(\tan^{-1}(2^{-i}))$, for each iteration i. Following an initial rotation of $-90°$, the recursive equations for $i = 1, 2, ..., N$ are

$$x'_{i+1} = x'_i - y'_i \cdot d_i \cdot 2^{-i} \quad (4)$$

$$y'_{i+1} = y'_i + x'_i \cdot d_i \cdot 2^{-i} \quad (5)$$

$$w_{i+1} = w_i - d_i \cdot \tan^{-1}(2^{-i}) \quad (6)$$

$$d_i = \text{sign}(w_i) = \begin{cases} +1 \ , \ w_i \geq 0 \\ -1 \ , \ w_i < 0 \end{cases} \quad . \quad (7)$$

A signal flow graph can be used to illustrate the rotation of each iteration. This particular structure is known as a pipelined architecture [14],[10] and is illustrated in Figure 3. Each stage of the unit will produce a particular amount of rotation, according to $\tan^{-1}(2^{-i})$. Moreover, the hardware is strictly comprised of digital elements.

Thus in order to generate a sinusoidal signal, θ is repeatedly from from 0 to π, so that repeating sinusoid can be produced. This can be expedited with a register. By initializing the contents of an M bit register to contain exclusively zeros and by incrementally adding W to this register in a step-wise fashion, a periodic pattern can be created. It is assumed that the carry out bit of the most significant bit is discarded. As an example, suppose that an $M = 3$ bit register which is initialized to 000 is given.

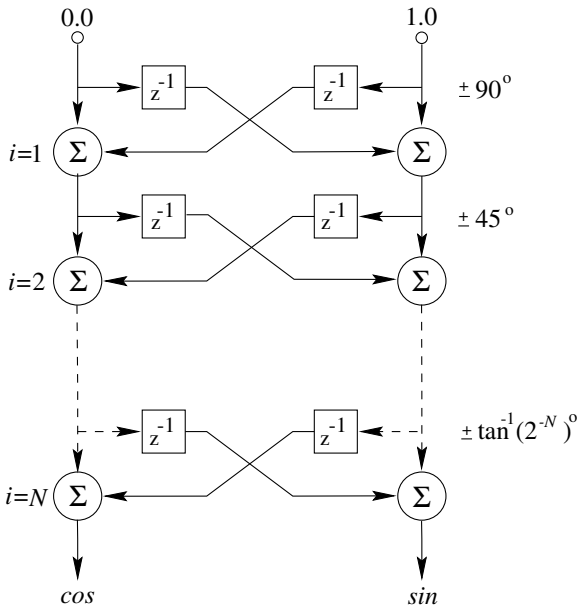

Fig. 3. Signal Flow Graph of the CORDIC Processor

By repeatedly adding $W = 001$ to the contents of the register, it will cycle through all of its other $(2^3 - 1)$ combinations and return to 000. Thus by appropriately scaling the contents of the register, θ will increment from 0 to π, and repeat continuously. As depicted in Figure 4, upon every increment of the clock, "clk", the output of a register can be used to provide an updated θ to the CORDIC processor.

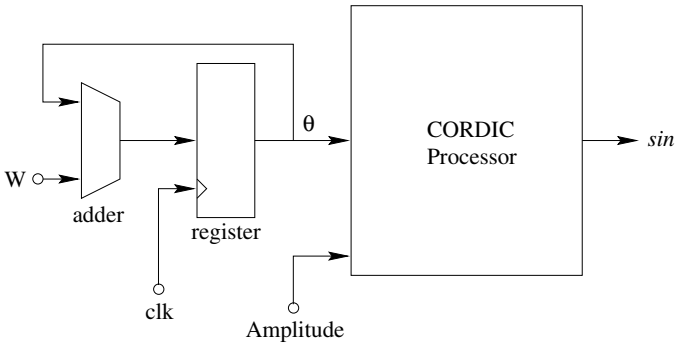

Fig. 4. CORDIC Processor Employed to Produce a Sinusoid

III. APPLICATIONS TO MIXED-SIGNAL TEST

The results section of this paper is organized as follows. Subsection A provides background information on how the theory of the proposed technique, which is supported by a simulation. While Subsection B discusses its application in the laboratory.

A. Theory and Simulation

As reported in the current liturature, the coherence property of a sampled data set is of prime importance [15][16][17]. By definition, coherency is an important property that eliminates additional unwanted discontinuities in a data set which introduce unwanted artifacts in the signal's spectral content. Thus, coherence in a sampled waveform is preferred in order to use the Discrete Fast Fourier Transform (DFFT) which is commonly used for data analysis. As noted by Mielke [18], in order to have a coherently sampled waveform, the frequency of the incoming sinusoid and the sampling speed must be accurately set according to

$$\frac{L_c}{L_s} = \frac{f_t}{f_{adc}} \tag{8}$$

where L_c is the number of cycles of an analog signal under test whose frequency is f_t Hertz and L_s is the number of samples taken at an analog-to-digital frequency of f_{adc} Hertz. In addition to (8), the variables L_c and L_s must be integers. This paper proposes to satisfy these constraints by scaling the master clock speed, f_{clk}, appropriately, as depicted in Figure 5. Since the frequency of the CORDIC processor's output depends on frequency of "clk", a predictable f_t and f_{adc} can be derived from f_{clk} which will satisfy (8).

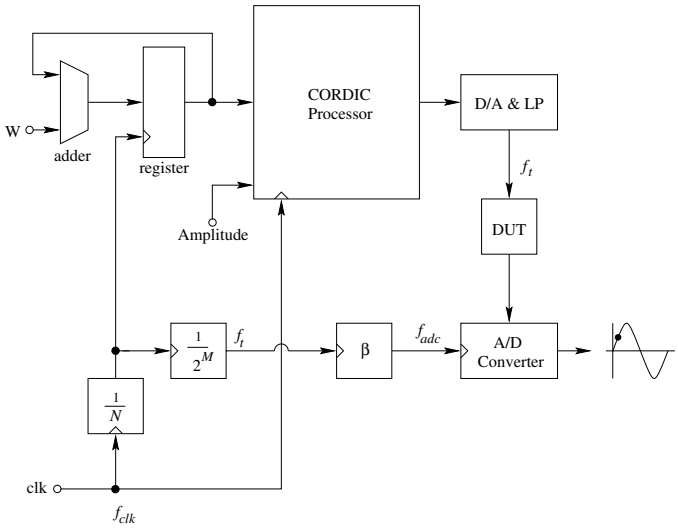

Fig. 5. By Appropriately Establishing a Network of Clock Signals, a CORDIC Processor can be Applied to Mixed-Signal A/D Conversion

The variable β was established to represent a clock divider that will ensure the sampling speed of the ADC is slightly less that the speed of the analog signal to be sampled. (Since undersampling is employed, it is assumed that the analog signal under test is periodic over L_c cycles). Thus

$$\beta = 1 - p2^{-M} \quad . \tag{9}$$

The integer p lies in the range of $1 \leq p \leq (1 + 2^{M-1})$, the upper bound is established since more than two samples are required to reconstruct an analog signal.

To convince the reader that this approach to undersampled A/D conversion works, the following simulation experiment is considered. The variables M, W, and N were established in an attempt to produce a spectrally pure sinusoid. Thus, M was assigned to be 8, and it was incremented by $W = 00000001$. N was set to 9, which produced a fine resolution of $\tan^{-1}(2^{-9}) = 0.1119°$ within the CORDIC processor. The variable p was set to 16, which would mandate that 16 cycles of the sinusoid to be generated to complete the sampling process. Figure 6 depicts the output of the sinusoidal generator, with several samples. To create $L_s = 16$ samples for a 16 point DFFT, precisely $NL_s2^M = 36864$ clock cycles were employed to create a perfectly coherent sample set, as depicted in Figure 7. It was assumed that each of the N rotations require one clock cycle. The result also assume that a full A/D conversion occurs at each strobe of the A/D converter. It is also assumed that the ADC conversion time is longer than the DAC settling time. It should be noted that the LP filter in Figure 5 is typically a source of a delay (linear distortion), but is reduced to zero in this simulation.

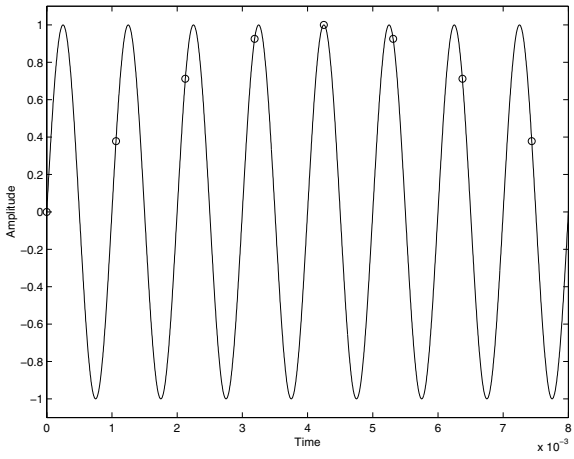

Fig. 6. Several Cycles of an Analog Signal with Accompanying Sample Points.

B. Laboratory Measurements

To illustrate how the conversion strategy works, an example has been prepared which employs a 16-bit fixed point Texas Instruments TMS320C6201 digital signal processor (DSP). Figure 8 depicts several cycles of the output of the CORDIC processor in conjunction with a 16 bit DAC and passive LP filter. The variables M, W, and N were established to be the same as they were in the previously mentioned simulation.

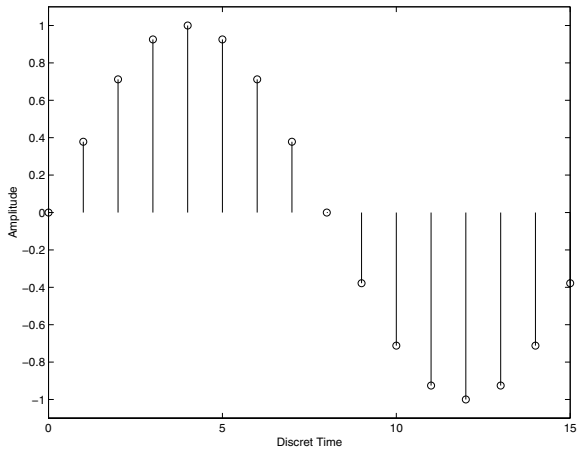

Fig. 7. The Full Cycle Set of Samples

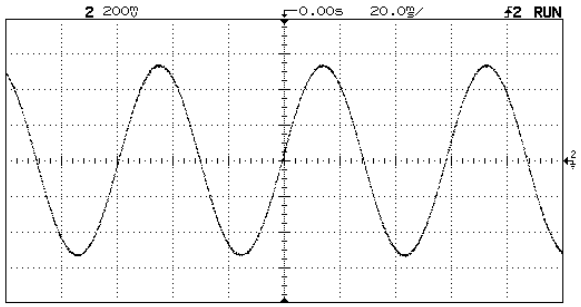

Fig. 8. Sinewave Generation via the CORDIC Processor

Figure 9 depicts a zoom-in view of a section of the sinewave as pictured in Figure 8. The DSP writes discrete data to the DAC, while the LP filter smoothly connects them. It should be noted that the LP filter in Figure 5 is a source of a delay (linear distortion), as well as any delay that is produced by the DUT. Therefore, this is accounted for by slightly delaying the strobe time of the DAC.

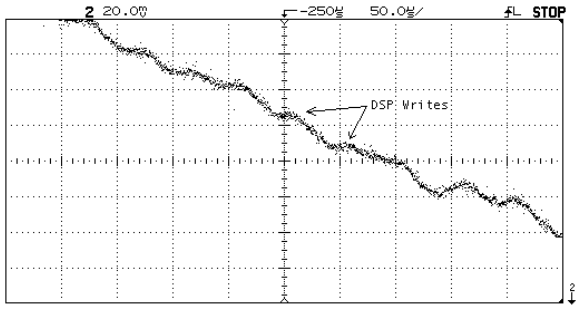

Fig. 9. Zoom-in view of the Sinusoidal Output following the LP Filter

IV. CONCLUSIONS

An analog-to-digital conversion technique was developed which employs a CORDIC processor. By doing this, it has been shown that a coherently sampled signal may be produced. Moreover, one of the main advantages of the new technique is that it only requires adders, multipliers, and registers to generate the sinusoidal signal. The duration of which may be precisely controlled by a digital clock. Since it is a well known fact that the amount of test time devoted to a DUT is critical, a predictable number of clock cycles may be issued to complete the sampling process. To our knowledge, this is the first time that such an all digital CORDIC processor has been applied in the area of mixed-signal testing.

References

[1] C. Coombs, *Electronic Instrument Handbook*. New York: McGraw-Hill, 1995.

[2] R. Tangelder, H. de Vries; R. Rosing, H. Kerkhoff, M. Sachdev, "Jitter and decision-level noise separation in A/D converters," *Proc. IMTC*, vol. 3, pp. 1558-1562, 1999.

[3] E. Grayver, B. Daneshrad, "Direct digital frequency synthesis using a modified CORDIC," *Proceedings of the IEEE International Symposium on Circuits and Systems*, vol. 5, pp. 241-244, 1998.

[4] J. Tierney, C. Rader, and B. Gold, "A digital frequency synthesizer," *IEEE Transactions on Audio Electroacoustics*, vol. AU-19, pp. 48-56, 1971.

[5] P.M. Grant, "Digital Signal Processing Part 1: Digital Filters and the DFT," *Electronics & Communication Engineering Journal*, Feb., pp. 13-21, 1993.

[6] J. Vankka and B. Daneshrad, "Methods of mapping from phase to sine amplitude in direct digital synthesis," *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, vol. 44, No. 2, pp. 526-534, 1997.

[7] A. Madisetti, A. Kwentus, A. Willson, "A 100-MHz, 16-b, direct digital frequency synthesizer with 100-dBc spurious-free dynamic range," *IEEE Journal of Solid-State Circuits*, vol. 34, No. 8, pp. 1034-1043, 1999.

[8] N. Fliege, "Complex digital oscillators and FSK modulators," *IEEE Transactions on Signal Processing*, vol. 40, pp. 333-342, 1992.

[9] J. Volder, "The CORDIC trigonometric computing technique," *IEEE Transactions on Computers*, vol. EC-8, pp. 330-334, 1959.

[10] G. Gielis, R. Plassche, and J. van Valberg, "A 540-MHz 10-b polar-to-cartesian converter," *IEEE Journal of Solid-State Circuits*, vol. 26, No. 11, pp. 1645-1650, 1991.

[11] S. Wang, V. Piuri, E. Swartzlander, Jr., "A unified view of CORDIC processor design," *IEEE 39th Midwest symposium on Circuits and Systems*, vol. 2, pp. 852-855, 1996.

[12] L. Leithold, *The Calculus*. New York, New York: Harper and Row, 5th edition, 1986.

[13] M. Yeary, *Design of an FIR filter for directional edge detection*. Master's Thesis, Texas A&M University, 1994.

[14] V. Considine, "CORDIC trigonometric function generator for DSP," *Proc. ICASSP*, vol. 4, pp. 2381-2384, 1989.

[15] P. Carbone and G. Chiorboli, "ADC sinewave histogram testing with quasi-coherent sampling," *Proc. IMTC*, vol. 1, pp. 108-113, 2000.

[16] A. Breitenbach and I. Kale, "An almost leakage free method for assessing $\Sigma - \Delta$ modulator spectra," *IEEE Transactions on Instrumentation and Measurement*, vol. 47, No. 1, pp. 40-44, 1998.

[17] L. Hsieh and A. Grochowski, "THD and SNR tests using the simplified volterra series with adaptive algorithms," *Proc. ITC*, pp. 364-369, 1995.

[18] J. Mielke, "Frequency domain testing of ADC's," *IEEE Design & Test of Computers*, vol. 13, No. 1, pp. 64-69, 1996.