A CORDIC-BASED DIGITAL QUADRATURE MIXER : COMPARISON WITH A ROM-BASED ARCHITECTURE

Seunghyeon Nahm, Kyungtae Han, and Wonyong Sung

School of Electrical Engineering, Seoul National University

ABSTRACT

This paper describes a CORDIC-based digital quadrature mixer in comparison with a ROM-based architecture. This architecture employs the circular rotation mode of the CORDIC algorithm for signal mixing as well as carrier generation, while the ROM-based architecture requires an additional complex multiplier for signal mixing. To optimize the hardware design parameters, the finite word-length effects of two implementation architectures are analyzed numerically. In addition, the simulation-based word-length determination is also conducted for the application to QPSK and QAM demodulators. The hardware costs of them are estimated, which shows that the CORDIC-based architecture occupies only a third of the area of the ROM-based architecture.

1. INTRODUCTION

The quadrature modulator and demodulator is the front-end of various modulation schemes such as BPSK (Binary Phase Shift Keying), QPSK (Quadrature Phase Shift Keying), and QAM (Quadrature Amplitude Modulation). In the receiver, the quadrature mixer should be able to mix an input signal with sinusoids having a fine frequency resolution for coherent demodulation [1]. Figure 1 shows the digital quadrature mixer which is used for IF (Intermediate Frequency) to baseband down-conversion in a wireless channel or passband to baseband demodulation in a wire-based channel. A DDFS (Direct Digital Frequency Synthesizer) has been used conventionally for implementing the digital quadrature oscillator [1][2].

There have been many efforts for implementing an efficient DDFS, and most of them employ a ROM (Read Only Memory) look-up table that stores sine-wave samples [2][3][4]. However, this approach not only needs a large ROM size for obtaining a fine frequency resolution but also has a speed limit because a large ROM is slow. The straightforward implementation according to Fig. 1 needs a DDFS for generating quadrature signals and separate multipliers for signal mixing [1][2]. The overall quadrature

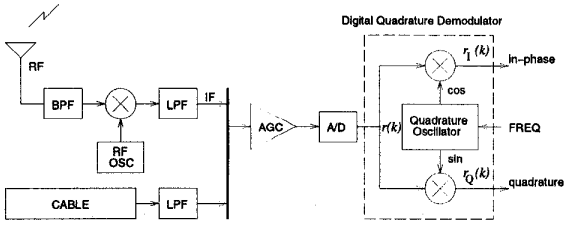


Figure 1: The digital quadrature mixer

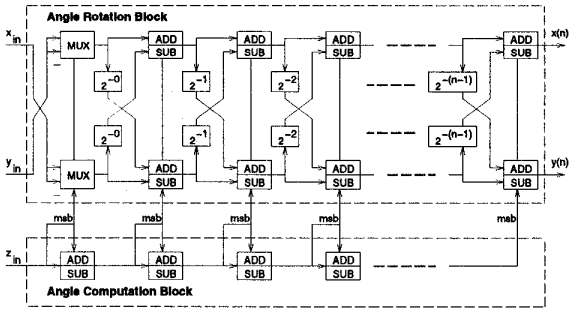


Figure 2: The CORDIC processor

mixing can also be implemented using the CORDIC algorithm. Using this algorithm, the digital phase-locked loop [5] or the digital costas loop [6] can be easily implemented. However, the practical comparison of the hardware cost and the speed is needed for the application to real systems.

In this paper, we analyze the optimum word-lengths for implementing quadrature mixers using both ROM and CORDIC-based algorithms, evaluate the speed of them, and compare the hardware cost for the VLSI implementation.

2. CORDIC-BASED DIGITAL QUADRATURE MIXER

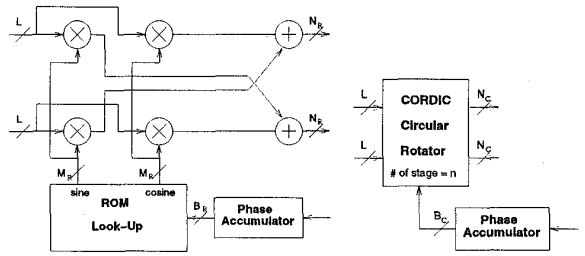
The quadrature mixing needs the generation of cosine and sine waves having the phase of $\omega_0 k + \theta(k)$ and multiplications with the input signal, $r(k)$. This can be interpreted as a circular rotation of $[r(k)\ 0]^t$. If the input signal has been split into the real and imaginary components prior to the down-conversion, a complex mixer is needed for performing true single side-band down-conversion to baseband [2].

The circular rotation can be implemented efficiently using a CORDIC algorithm, which is an iterative arithmetic algorithm for computing many elementary functions [7]. There is no need of scale factor normalization since a constant gain is allowed in this application. In a parallel hardware implementation, a CORDIC processor is composed of the angle rotation and the angle computation blocks as shown in Fig. 2. The former computes the rotated position in the rectangular coordinate for each iterative rotation, and the latter generates the direction control signals resulting from the computation of the rotation angle.

0-7803-4455-3/98/$10.00 © 1998 IEEE

3. FINITE WORD-LENGTH EFFECTS

For the comparison of the CORDIC and the ROM-based architectures, the finite word-length effects are analyzed. The hardware models of them are shown in Fig. 3. Both architectures have an input word-length, L, which is the A/D converter precision, and they use the same phase accumulator for the frequency control. The output of the phase accumulator is truncated to B_C, or B_R bits, to provide the required phase precision to the digital quadrature oscillator. In the ROM-based architecture, the quadrature wave output of the ROM are quantized to M_R bits and $L \times M_R$-bit multiplication is followed for the signal mixing. In the CORDIC-based architecture, the angle is approximated by n elementary rotations, whose direction is calculated in B_C bit arithmetic, and the computation for the elementary rotation is conducted by N_C bit addition and subtraction operations. In any case, the finite word-length effects should be analyzed for the final mixed result not for the quadrature signals.



(a) (b)

Figure 3: Hardware models of the digital quadrature mixer; (a) a ROM-based and (b) a CORDIC-based architecture

3.1. Analysis-based Word-length Optimization

For the CORDIC-based architecture, Hu [7]'s study on the numerical error bound is employed with some modifications. These modifications consider that the input signal is represented by finite number of bits, thus the intermediate results of the angle rotation block can be designed not to have any quantization noise by adding guard bits, and the phase input and the elementary angles are also quantized, thus the word-length for the angle computation block should be larger than the number of stages. The error bound can be controlled by the word-length of the angle rotation block, and the minimum error bound is confined by the number of stages.

For the ROM-based architecture, the main error source is the input phase quantization, and the multiplicative error from the multiplication of the input signal and the ROM output should also be considered. In this paper, the details of the analyses are not presented and only the results are shown.

For the comparison of two architectures, the strategy of maintaining the number of significant bits of the input is used. Using this strategy, the computed result is equivalent to the value quantized by the same word-length after an ideal demodulation. The outputs are assumed to be normalized by the mixer gain.

Table 1 shows the determined optimum word-lengths of two types of digital quadrature mixer, where the previously mentioned performance criterion is used, that is, the upper bound of 0.5 regarding the input word-length as an integer part in a fixed-point data representation.

Table 1: Optimum word-lengths of the digital quadrature mixer

Input WL	ROM-Based Precision 0.5			CORDIC-Based Precision 0.5		
L	B_R	M_R	N_R	B_C	n	N_C
4	7	6	10	10	6	9
5	8	7	12	11	7	10
6	9	8	14	12	8	11
7	10	9	16	13	9	13
8	11	10	18	14	10	14
9	12	11	20	15	11	15
10	13	12	22	17	12	16

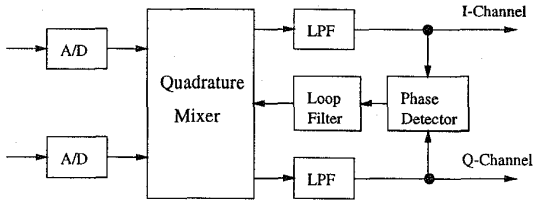


Figure 4: Block diagram for the simulation-based word-length optimization

3.2. Simulation-based Word-length Optimization

We conducted a simulation-based word-length optimization because keeping the effective numerical precision is not the exact design objective for digital communication systems. The final design objective of the receiver is the BER (Bit Error Rate), but the Monte Carlo simulation will require millions of data bits for measuring the BER under 10^{-5}. Due to this difficulty, we use the output SNR (Signal-to-Noise Ratio) instead of BER as a gauge of the system performance [8]. Figure 4 shows the simulation environment for the word-length determination.

In order to find the optimum word-lengths that satisfy the system performance while minimizing the hardware cost, the simulation based word-length optimization method is employed [9]. First, the minimum word-length for each signal, or the critical word-length, is determined. The critical word-length of a signal is the smallest word-length guaranteeing the system performance when all the other signals have enough precision. Then, from this lower bound of word-lengths, the set of the optimal word-lengths that minimizes the hardware cost while satisfying the given specifications is determined. In this search procedure, we used the fast search algorithm [9].

Figure 5 shows the output SNR according to the number of bits for each signal in the CORDIC-based implementation of the QPSK demodulator. The input signal is assumed to have 8 dB of SNR.

Table 2 shows the results of the simulation-based word-length optimization for QPSK, QAM16, and QAM64. The design criterion is the SER (Symbol Error Rate) of 10^{-5}.

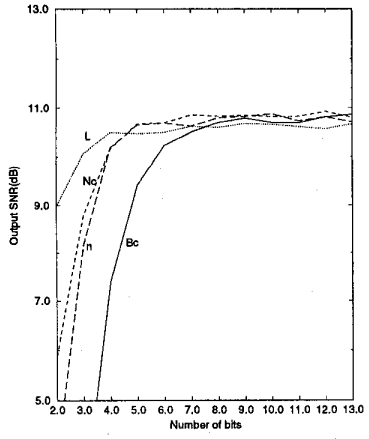


Figure 5: Output SNR versus number of bits

Table 2: Optimum word-length determined by simulation

		ROM-Based			CORDIC-Based		
		SER = 10^{-5}			SER = 10^{-5}		
MOD	L	B_R	M_R	N_R	B_C	n	N_C
QPSK	5	7	5	6	7	6	6
QAM16	6	8	7	7	8	7	7
QAM64	7	9	8	8	9	8	8

4. HARDWARE COST AND SPEED ESTIMATION

4.1. Hardware Cost

VTI 1.0μm cell library is used for estimating the hardware cost [10]. Since a ROM and a multiplier can be efficiently implemented utilizing their regular structures, the number of transistors or the gate counts cannot be used for reasonable comparison. Thus, the areas occupied by the macrocells are estimated using a cell compiler library.

A direct implementation of a ROM-based architecture shown in Fig. 3-(a) needs two ROM's which store all the sine and cosine samples, and four multipliers and two adders. The size of the ROM's can be reduced without affecting the precision by using the eighth-wave symmetry of the quadrature waves. The sine and cosine samples ranging from 0 to $\pi/4$ are only needed and the sign of the outputs can be determined by the phase input. Therefore, the address is reduced from B_R to $B_R - 3$ bits and the output from M_R to $M_R - 1$ bits. Several techniques for the ROM reduction can also be used when the size is large. However, the intensive application of such techniques are so complex, thus no additional ROM reduction technique is assumed. The size of multipliers cannot be reduced, since it has a large effect on the output precision. However, the size of adders can be reduced to a required amount for the next computation.

A CORDIC-based architecture is composed of two blocks: an angle rotation block and an angle computation block. The former consists of an adder/subtracter array and fixed hard-wired shifts. The latter also has an adder/subtracter array for adding or subtracting pre-determined values, which can be implemented using simpler gates than full adders. And there is no need of computation at the first and the last stages, and the rest of the stages only require $B_C - 3$ bit addition/subtraction with fixed values [4]. Table 3 shows the hardware requirement of a ROM-based and a CORDIC-based architectures.

Table 3: Hardware requirements

Hardware Unit	Number of Units	Size
ROM-Based Architecture		
ROM	2	$2^{B_R-3} \times (M_R - 1)$
Multiplier	4	$L \times M_R$
Adder	2	$N_R \leq L + M_R$
CORDIC-Based Architecture		
Adder/Subtracter	$2 \times n$	N_C
Adder/Subtracter	$n - 2$	$B_C - 3$

Table 4: Cost comparison using analysis-based optimization

Input WL	ROM-Based	CORDIC-Based
L	Area	Area
4	6799687	2422000
5	9959408	3287000
6	10947340	4394200
7	14877652	5501400
8	16376272	6643200
9	21519260	8044500
10	24452392	9584200

The hardware cost is estimated in Table 4 using the analysis-based word-length optimization results. Note that the unit is λ^2 ($\lambda = 0.5\mu$m) in the VTI 1.0 μm cell library. Table 5 also shows the required hardware cost of the two implementation methods using the simulation-based word-length optimization results in the QPSK, QAM16, and QAM64 systems. The hardware cost comparison results show that the CORDIC-based implementation requires only about a third of the area needed for the ROM-based one at both the analysis-based and the simulation-based word-length optimization methods.

4.2. Speed

The critical path of the ROM-based architecture is the path from the ROM address inputs to the complex multiplier outputs. Thus, the total delay is the sum of the ROM output delay, one multiplier, and an adder delay. However, the critical path of the CORDIC-based architecture is somewhat complex. Figure 6 shows the crit-

Table 5: Cost comparison using simulation-based optimization

MOD	ROM-Based	CORDIC-Based
QPSK	6059720	1522400 (25.12 %)
QAM16	7385598	2127900 (28.81 %)
QAM64	10770880	2837200 (26.34 %)

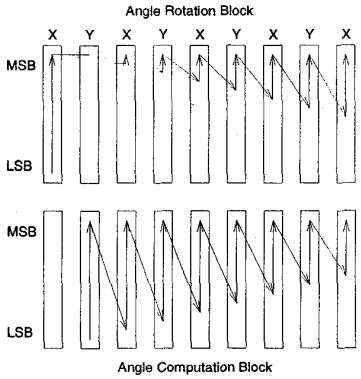


Figure 6: Critical path of the CORDIC-based architecture

ical path of the CORDIC-based architecture, where each stage of the rotation block represents X block and Y block alternatively. The angle rotation block has right shift operations, thus the carry propagation from the MSB of the previous stage to the MSB of the present one is needed as shown in Fig. 6, and this propagation becomes longer as the stage proceeds. The angle computation block requires the sign computed from the previous stage at every stage, thus has to wait until the computation of the previous stage is completed. However, the computed angle should be reduced as the stage proceeds, and the delay of the sign detection is also reduced accordingly. The rotation block can be operated only after the angle computation is completed at each stage. Thus, the total critical path is composed of the longer paths of these two blocks. Table 6 shows the estimated delay using the analysis-based word-length optimization results.

When the ripple-carry adder is used, the CORDIC-based architecture is slower than the ROM-based. This phenomenon is more noticeable as the precision is higher. However, the CORDIC processor has a potential to improve its speed. The carry propagation of the angle rotation block can be removed by employing a carry-save adder array although it doubles the area. When a pipelining scheme is employed, the speed of the CORDIC-based architecture can be as fast as one stage of angle computation block. If a fast direction sequence generation method is employed, the bit-level pipelining is also possible [6]. On the other hand, the speed of the ROM-based architecture is limited by the non-regular structure even if a pipelining scheme is employed.

Table 6: Speed of the digital quadrature mixer (ns)

Input WL	ROM-Based	CORDIC-Based
L	Delay	Delay
6	44.5	43
7	47.0	56
8	51.5	69
9	55.5	85
10	65.0	109

4.3. Comparison for Slow Applications

Many of the quadrature mixer need not be operated at the maximum speed, which can exceed 10 MHz according to Table 6. When the speed requirement becomes a half of the maximum, the CORDIC-based architecture can reduce the hardware accordingly because the CORDIC is an iterative algorithm. However, the ROM-size required in the ROM-based quadrature mixer can not be reduced. In the ROM-based architecture, the ROM requires approximately 16% of the total chip area in the full-speed implementation. Thus, the CORDIC-based implementation is more advantageous for low to medium throughput quadrature mixers.

5. CONCLUDING REMARKS

It is shown that the CORDIC-based digital quadrature mixer is a promising architecture in the view of the hardware cost and the design flexibility. The disadvantage is that the operation speed is slow for high precision, which can be overcome by pipelining or employing fast CORDIC algorithms [6].

6. REFERENCES

[1] Bong-Young Chung, Charles Chien, Henry Samueli, and Rajeev Jain, "Performance analysis of an all-digital BPSK direct-sequence spread-spectrum IF receiver architecture," *IEEE J. Selected Areas Commun.*, vol. 11, no. 7, pp. 1096–1107, Sep. 1993.

[2] Loke Kun Tan and Henry Samueli, "A 200 MHz quadrature digital synthesizer/mixer in 0.8-μm CMOS," *IEEE J. Solid-State Circuits*, vol. 30, no. 3, pp. 193–200, Mar. 1995.

[3] Gerard Gielis, Rudy van de Plassche, and Johan van Valburg, "A 540-MHz 10-b polar-to-cartesian converter," *IEEE J. Solid-State Circuits*, vol. 26, no. 11, pp. 1645–1650, Nov. 1991.

[4] Henry T. Nicholas III and Henry Samueli, "A 150-MHz direct digital frequency synthesizer in 1.25-μm CMOS with -90-dBc spurious performance," *IEEE J. Solid-State Circuits*, vol. 26, no. 12, pp. 1959–1969, Dec. 1991.

[5] Jarkko Vuori, "Implementation of a digital phase-locked loop using CORDIC algorithm," in *Proc. IEEE ISCAS '96*, 1996, pp. 164–167.

[6] Seunghyeon Nahm and Wonyong Sung, "A fast direction sequence generation method for CORDIC processors," in *Proc. IEEE ICASSP '97*, Apr. 1997, pp. 635–638.

[7] Yu Hen Hu, "The quantization effects of the CORDIC algorithm," *IEEE Trans. on Signal Processing*, pp. 834–844, Apr. 1992.

[8] Jen-Shi Wu, Ming-Luen Liou, Hsi-Pin Ma, and Tzi-Dar Chiueh, "A 2.6-V, 44-MHz all-digital QPSK direct-sequence spread-spectrum transceiver ic," *IEEE Journal of Solid-State Circuit*, pp. 1499–1510, Oct. 1997.

[9] Wonyong Sung and Ki-Il Kum, "Simulation-based word-length optimization method for fixed-point digital signal processing systems," *IEEE Trans. on Signal Processing*, pp. 3087–3090, Dec. 1995.

[10] VLSI Technologies, Inc., *1-Micron Cell Compiler Library*, Nov. 1991.