

# CORDIC을 이용한 디지털 Quadrature 복조기의 VLSI 구현

정회원 남 승 현\*, 성 원 용\*\*

## VLSI Implementation of CORDIC-Based Digital Quadrature Demodulator

Seung Hyeon Nahm\*, Won Yong Sung\*\* *Regular Members*

\*본 논문은 1995년도 교육부 학술 연구 조성비에 의하여 연구되었음(과제번호 : ISRC 95-2-2018).

### 요 약

디지털 quadrature 복조기는 디지털 통신 시스템에서 변조된 신호의 정확한 위상 복조를 위해 꼭 필요하다. 기존의 방법들은 주로 DDFS (Direct Digital Frequency Synthesizer)를 이용하여 캐리어를 발생시킨 후에 승산기를 이용하여 복조를 수행하였다. 그리고, DDFS에는 주로 ROM (Read Only Memory)을 사용하였는데, 높은 속도와 정확도를 요구하는 경우 ROM의 속도와 크기가 제한이 될 수 있다. 이러한 점을 극복하기 위하여 CORDIC (COordinate Rotation Digital Computer) 알고리즘을 사용하여 주파수 합성은 물론 캐리어 복조까지 수행하는 방식을 제안하였다. 최적의 하드웨어 구현을 위해 제한된 단어길이에 의한 영향을 분석하였으며, 하드웨어 비용면에서 ROM을 사용하는 방법과 비교한 결과 약 1/3 정도로 면적이 줄었다. 제안된 구조를 이용한 전주문형 VLSI 구현 결과를 보인다.

### ABSTRACT

Digital quadrature demodulator is needed for the coherent demodulation in the digital communication systems such as Binary Phase-Shift-Keying, Quadrature Phase-Shift-Keying, and Quadrature Amplitude Modulation. Conventionally, the DDFS (Direct Digital Frequency Synthesizer) is used for generating the carrier signal and separate multipliers are used for mixing. And the DDFS is implemented using the ROM (Read Only Memory), which can be a bottle-neck when the fast-speed and small-area implementation is required. A new architecture is developed, which employs the circular rotation mode of the CORDIC algorithm for signal mixing as well as carrier generation. To optimize the hardware design parameters, the finite word-length effects of the proposed implementation architecture are analyzed in comparison with a conventional ROM-based architecture. The hardware costs are also estimated, which showed that the proposed architecture occupies only a third of the area of the conventional ROM-based architecture for the same performance. A full-custom VLSI is developed using the proposed architecture.

\* 고등기술연구원 전자통신연구실

\*\* 서울대학교 전기공학부

論文番號 : 96403-1223

接受日字 : 1996年 12月 23日

### I. 서론

CMOS 집적회로 기술의 발달로 통신용 시스템의 구현에 있어서 많은 아날로그 회로가 디지털 회로로 대체되고 있다. 디지털 회로로 구현함으로써 공정이 까다롭고 설계의 많은 know-how가 필요한 아날로그 방식의 단점을 극복할 수 있다. 그리고, 여러가지 전송 속도와 변조 방식, 필터 특성을 갖도록 유연한 구조로 구현할 수 있다.

Quadrature 복조기는 BPSK (Binary Phase Shift Keying)와 QPSK (Quadrature Phase Shift Keying), QAM (Quadrature Amplitude Modulation)과 같은 변조 방식을 사용하는 디지털 통신 시스템의 수신기에서 채널에 가장 가까이 위치하는 부분이다[1][2]. 대역확산 통신 시스템에서도 확산된 신호를 BPSK 또는 QPSK로 변조하여 송신하므로, 수신기에서 quadrature 복조기를 사용하여 기저대역 신호로 변환한 후에 역확산을 수행하게 된다[3][4]. Quadrature 복조기를 구현하기 위한 예전의 방법은 아날로그 주파수 mixer를 사용하는 것이었는데, 앞서 언급한 디지털 구현의 장점 때문에 최근에는 아날로그 회로로 중간 주파수 (Intermediate Frequency) 대역으로 주파수를 낮춘 후, 디지털 quadrature 복조기를 사용하여 기저대역 신호를 추출하는 방식을 사용하는 경향이 크다. 그림 1에 디지털 quadrature 복조기를 보였는데, 무선 채널에서는 일반적으로 중간 주파수 (Intermediate Frequency) 대역에서 기저대역으로, 유선 채널에서는 통과대역에서 기저대역으로의 복조에 사용된다. Quadrature 복조를 위해 sine/cosine파를 발생시키는 디지털 발진기가 필요한데, 기존에는 보통 DDFS (Direct Digital Frequency Synthesizer)를 사용하였다[3][4][5].

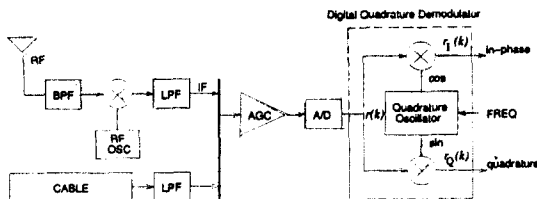


그림 1. 디지털 quadrature 복조기

주로 PLL (Phase Locked Loop)을 쓰는 아날로그

방식과 비교할 때 DDFS는 빠른 연속 위상전이가 반응 (fast continuous-phase switching response)과 높은 주파수 해상도 (fine frequency resolution), 넓은 대역폭, 좋은 spectral purity 등의 우수한 성능을 보인다. 그동안 고성능, 고속의 DDFS를 적은 chip 면적에 구현하기 위한 연구가 진행되어 왔는데, 그 대부분은 ROM (Read Only Memory)을 이용한 색인표 (table)에 sine/cosine 값들을 저장하는 방식이었다[5][6][7]. 한편, 주파수의 정밀도를 높이기 위해서는 크기가 큰 ROM이 필요한데, 이는 면적을 많이 차지할 뿐만 아니라 속도에도 제한을 주게 된다. 또 다른 접근 방법으로 각회전 알고리즘을 사용하는 것이 있는데, 극 좌표계로 표현된 값을 직교 좌표계로 변환함으로써 sine/cosine 파를 발생시킨다[6][8]. 이 방법은 쉬프트와 덧셈만으로 구성되므로 파이프라인 등의 기법으로 속도를 상당히 높일 수 있다.

그림 1의 quadrature 복조기를 구현하기 위해서는 앞에서 언급한 DDFS로부터 sine/cosine파를 발생시키고, 별도의 곱셈기를 통해 캐리어 복조를 수행해야 한다[3][4][5]. 본 연구에서는, 직교 좌표계의 circular rotation을 sine/cosine파의 발생 뿐만 아니라 주파수 mixing에도 사용하는 새로운 구조에 대하여 연구하였다. 제 2장에서는 기본적인 원리에 대해 설명하고, circular rotation을 위한 CORDIC (COordinate Rotation Digital Computer) 알고리즘에 대해 설명한다[9][10]. 제 3장에서는 최적의 하드웨어 구현을 위하여 기존의 ROM을 사용한 구조와 제안된 CORDIC을 사용한 구조의 양자화 효과를 분석하였다. 제 4장에서는 하드웨어 비용과 동작 속도를 추정 계산하여 서로 비교하였다. 제 5장에서는 제안된 구조를 사용하여 quadrature 복조기를 설계하여 디지털 Costas loop의 VLSI 구현에 응용한 예를 보이고, 그에 따른 하드웨어 설계에 대해 논의한다. 제 6장에서 결론을 맺는다.

### II. CORDIC 알고리즘을 사용한 Quadrature 복조

Quadrature 복조는 수신된 신호에서 in-phase 성분과 quadrature 성분을 추출하는 과정으로서, 캐리어와 같은 주파수와 위상을 갖는 sine/cosine파를 입력 신호와 곱함으로써 구현할 수 있다. 디지털 quadrature 복조는 그림 1과 같이 A/D (Analog to Digital) 변환

기를 통해 디지털화된 입력 신호에 대해서 이 과정을 수행하는 것이다. 그림 1에서 in-phase 성분과 quadrature 성분,  $r_I(k)$ 와  $r_Q(k)$ 는 다음과 같이 계산된다.

$$r_I(k) = Kr(k) \cos [\omega_0 k + \theta(k)] \quad (1)$$

$$r_Q(k) = Kr(k) \sin [\omega_0 k + \theta(k)] \quad (2)$$

여기에서  $\omega_0 k + \theta(k)$ 는 캐리어 신호의 위상이고,  $K$ 는 mixing 이득이다. 이 계산을 하기 위하여,  $\omega_0 k + \theta(k)$ 의 위상을 갖는 sine/cosine파를 발생시켜야 하고, 입력 신호  $r(k)$ 와의 곱셈을 수행해야 한다. 이 과정은 벡터 연산으로 표현하면, 다음과 같이  $[r(k) \ 0]^T$ 의 circular rotation으로 간주할 수 있다.<sup>1)</sup>

$$\begin{aligned} \begin{bmatrix} r_I(k) \\ r_Q(k) \end{bmatrix} &= \begin{bmatrix} Kr(k) \cos A(k) \\ Kr(k) \sin A(k) \end{bmatrix} \\ &= K \begin{bmatrix} \cos A(k) & -\sin A(k) \\ \sin A(k) & \cos A(k) \end{bmatrix} \begin{bmatrix} r(k) \\ 0 \end{bmatrix} \end{aligned} \quad (3)$$

여기에서  $A(k)$ 는 회전각  $\omega_0 k + \theta(k)$ 를 나타낸다. 그림 2-(가)에 circular rotator를 사용한 디지털 quadrature 복조기의 제안된 구조를 보였다.

입력신호가 RF 대역에서 이미 in-phase와 quadrature 성분으로 분리되어서 들어오는 경우에는, 다음과 같이 복소수 곱셈을 수행해야 한다[5][11].

$$\begin{aligned} r_I(k) + jr_Q(k) &= [r_{real}(k) + jr_{imag}(k)] \\ &\cdot K[\cos A(k) + j\sin A(k)] \end{aligned} \quad (4)$$

이 복소수 연산도 벡터 형태로 표현하면, 다음과 같이  $[r_{real}(k) \ r_{imag}(k)]^T$ 의 circular rotation으로 볼 수 있다.

$$\begin{bmatrix} r_I(k) \\ r_Q(k) \end{bmatrix} = K \begin{bmatrix} \cos A(k) & -\sin A(k) \\ \sin A(k) & \cos A(k) \end{bmatrix} \begin{bmatrix} r_{real}(k) \\ r_{imag}(k) \end{bmatrix} \quad (5)$$

그림 2-(나)에 circular rotator를 사용한 복소수 입력을 위한 디지털 quadrature 복조기의 제안된 구조를

보였다.

Circular rotation은 CORDIC 알고리즘을 이용하여 효율적으로 구현할 수 있다. CORDIC 알고리즘은 여러가지 기본 함수들을 반복연산을 통해 계산하는 방법으로서, 복잡한 디지털 신호처리 알고리즘을 구현 하는데 많이 응용되고 있다[9][10]. Circular rotation을 위한 CORDIC 알고리즘의 기본 알고리즘은 다음과 같다[6][9]. 직교 좌표  $[x_{in} \ y_{in}]$ 과 회전각  $z_{in}$ 이 주어졌을 때, 다음과 같이 변수들을 초기화한다.

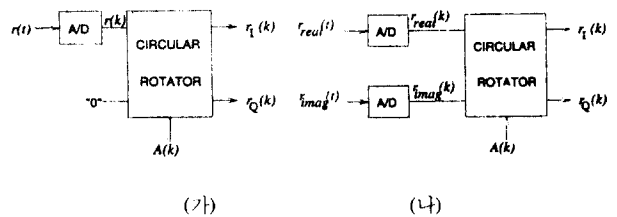


그림 2. Circular rotator를 사용한 디지털 quadrature 복조기 : (가) 실수 입력 (나) 복소수 입력 신호

$$x(0) = -y_{in} \text{sign}(z_{in}) \quad (6)$$

$$y(0) = +x_{in} \text{sign}(z_{in}) \quad (7)$$

$$z(0) = z_{in} - \text{sign}(z_{in}) \cdot 2 \arctan(1) \quad (8)$$

$i = 0, 1, 2, \dots, n-1$ 에 대해 다음과 같은 계산을 반복한다.

$$x(i+1) = x(i) - \mu(i) y(i) 2^{-i} \quad (9)$$

$$y(i+1) = y(i) + \mu(i) x(i) 2^{-i} \quad (10)$$

$$z(i+1) = z(i) - \mu(i) y(i) a(i) \quad (11)$$

여기에서  $\mu(i)$ 는  $z(i)$ 의 부호이고,  $a(i) = \arctan(2^{-i})$ 는 미리 정해진 기본각들이다. 이 circular rotation 과정은 다음과 같이 벡터 형태로 나타낼 수 있다.

$$v(i+1) = P(i) v(i) \quad (12)$$

여기에서  $v(i) = [x(i) \ y(i)]^T$ 는  $i$ 번째 반복에서의 rotation 벡터이고,  $P(i)$ 는 정규화하지 않은 rotation 행렬로서 다음과 같다.

1)  $[\cdot]^T$ 는 vector transpose를 나타낸다.

$$P(i) = \begin{bmatrix} 1 & -\mu(i)2^{-i} \\ \mu(i)2^{-i} & 1 \end{bmatrix} \quad (13)$$

$$= \sqrt{1+2^{-2i}} \begin{bmatrix} \cos a(i) & -\mu(i)\sin a(i) \\ \mu(i)\sin a(i) & \cos a(i) \end{bmatrix}$$

Rotation 벡터는  $n$ 번 반복 후 다음과 같은 값을 갖는다.

$$v(n) = K(n) \begin{bmatrix} \cos\{z_{i,n} - z(n)\} & -\sin\{z_{i,n} - z(n)\} \\ \sin\{z_{i,n} - z(n)\} & \cos\{z_{i,n} - z(n)\} \end{bmatrix} \begin{bmatrix} x_{i,n} \\ y_{i,n} \end{bmatrix} \quad (14)$$

여기에서  $K(n) = \prod_{i=0}^{n-1} \sqrt{1+2^{-2i}}$ 로서  $n$ 에 따라 정해지는 일정한 값이고,  $z(n) = z(0) - \sum_{i=0}^{n-1} \mu(i)a(i)$ 이다. Quadrature 복조를 위한 circular rotation은 이 식을 통하여 구현할 수 있다.  $K(n)$ 은 상수이므로 이 응용에서는 mixing 이득으로 보아서 정규화할 필요가 없다. CORDIC 알고리즘을 구현하기 위해서는 각 반복 계산을 하나의 덧셈기를 이용하여 순차적으로 수행하는 직렬 구조 혹은 단 수만큼의 덧셈기를 이용하는 병렬 구조를 사용할 수 있는데, 본 연구에서는 고속 응용 분야를 목표로 병렬 구조를 채용하였다. 병렬 구조의 CORDIC 프로세서는 그림 3과 같이 각 회전부 (Angle Rotation Block)와 각도 계산부 (Angle Computation Block)로 나누어 구현할 수 있다. 각 회전부는 매 반복 회전마다의 직교 좌표  $v(i)$ 를 계산하고, 각도 계산부는 회전방향 제어신호를 발생시키기 위하여  $z(i)$ 를 계산한다.

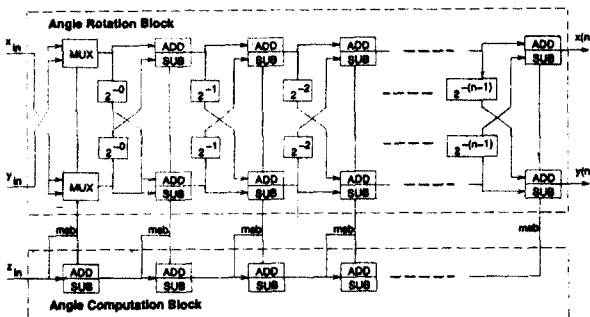


그림 3. 병렬 구조의 CORDIC 프로세서

### III. 제한된 단어길이의 영향 해석

디지털 신호처리에서 제한된 단어길이에 의한 영향의 분석은 경제적인 하드웨어 구현을 위해 꼭 필요하다. 이 경우 필요로 하는 성능의 구현을 위해 최소한의 비용, 일반적으로 게이트 수를 소모하도록 단어 길이들을 결정해야 한다. 본 장에서는 새롭게 제안된 CORDIC을 사용한 디지털 quadrature 복조기에서 제한된 단어길이가 성능에 미치는 영향을 분석하였다. 기존의 ROM을 사용한 구조에 대해서도 비슷한 분석을 하여 두 가지를 비교할 수 있도록 하였다.

그림 4에 CORDIC을 사용한 방법과 ROM을 사용하는 방법의 하드웨어 모델을 보였다. 분석의 일반화를

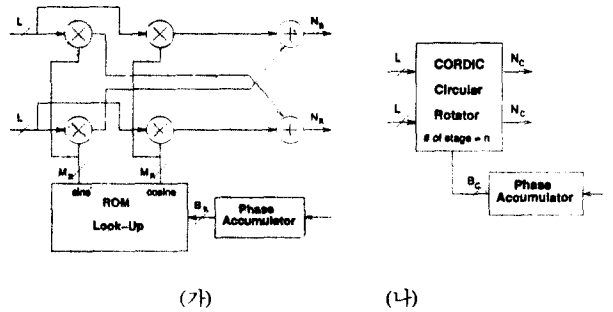


그림 4. 디지털 quadrature 복조기의 하드웨어 모델 : (가) ROM을 사용한 구조 (나) CORDIC을 사용한 구조

위해 복소수 입력을 가정하였고, 두가지 구조 모두 입력 신호의 단어길이를  $L$ 이라 하였다. 주파수 제어를 위해 사용하는 위상 누적기는 두 가지 구조에서 같으므로 위상 신호 출력으로부터 분석을 시작하였다. 위상 누적기의 출력은 디지털 quadrature 발진기에서 요구되는 위상 정밀도에 따라  $B_C$ 와  $B_R$  bit로 버림 된다. 2) ROM을 사용한 구조에서는 위상 입력을 단어길이  $B_R$ 로 양자화할 때와 sine/cosine과의 값을 단어길이  $M_R$ 로 양자화할 때 오차가 발생한다. CORDIC을 사용한 구조에서는 회전하는 단 수  $n$ 의 제한과 각도 계산부의 단어길이  $B_C$ 의 제한에 의해 발생하는 각 근사화 오차와, 각 회전부의 단어길이  $N_C$ 의 제한에

2) Subscript C는 CORDIC을 사용한 구조를 의미하고, subscript R을 ROM을 사용한 구조를 의미한다.

이한 반올림 오차가 발생한다. 디지털 quadrature 복조기의 제한된 단어길이의 영향은, 발생시키는 sine/cosine과 자체의 정밀도보다 mixing된 출력의 정밀도라는 측면에서 해석하여야 한다.

### 3.1 CORDIC을 사용한 구조

CORDIC 알고리즘의 양자화 효과에 대하여는 Hu[9]의 연구 결과가 있는데, 본 연구에서는 디지털 quadrature 복조기의 구현에 맞도록 더욱 구체화하여 실제 하드웨어 설계 사양의 최적화에 적합하도록 분석하였다. 각도 계산부와 각 회전부에서 발생하는 양자화 영향들을 나누어 설명한다.

#### 3.1.1 각 회전부(Angle Rotation Block)의 양자화 영향

CORDIC 알고리즘에서 각 회전부의 양자화 영향으로는 제한된 단 수에 의한 각 근사화 오차와 각 회전 계산 과정의 반올림 오차가 있다. CORDIC 알고리즘에서 회전각  $A$ 는 다음과 같이  $n$  개의 기본각  $\{a(i)\}$ 의 선형 조합으로 근사된다.

$$A = \text{sign}(A) \cdot 2 \arctan(1) + \sum_{i=0}^{n-1} \mu(i) a(i) + \delta \quad (15)$$

여기에서  $\delta$ 는 각 근사화에 의한 오차이다. CORDIC 알고리즘의 수렴 조건으로부터  $\delta$ 는 다음과 같은 범위를 갖는다.

$$|\delta| \leq a(n-1) = \arctan(2^{-n+1}) \quad (16)$$

$\delta=0$ 일 때의  $v(n)$ 의 이상적인 값을  $v^*(n)$ 이라 하면, 계산된 값  $v(n)$ 는 다음과 같이 표현된다.<sup>3)</sup>

$$v(n) = \begin{bmatrix} \cos \delta & \sin \delta \\ -\sin \delta & \cos \delta \end{bmatrix} v^*(n) \equiv Dv^*(n) \quad (18)$$

그러므로, 각 근사화에 의한 양자화 오차는 다음과 같이 계산된다.

$$v(n) - v^*(n) = (D-I)v^*(n) \quad (18)$$

3) 각 근사화 오차를 해석할 때는 반올림 오차가 없다고 가정한다.

이로부터 각 근사화에 의한 오차한계는 다음과 같이 추정할 수 있다[9].<sup>4)</sup>

$$|v(n) - v^*(n)| \leq \|(D-I)\| \cdot |v^*(n)| \leq |\delta| \|v^*(n)\| \leq a(n-1) |v^*(n)| \quad (19)$$

입력 단어길이가  $L$ 이라면, 입력 좌표는  $|x_{i,n}| \leq 2^{L-1}$ ,  $|y_{i,n}| \leq 2^{L-1}$ 로 크기가 제한되므로, 식 (19)의 오차한계는 다음과 같이 입력 단어길이를 표현된다.

$$|v(n) - v^*(n)| \leq a(n-1) |v^*(n)| \leq a(n-1) K(n) \sqrt{2} \cdot 2^{L-1} \quad (20)$$

반올림 오차를 해석하기 위하여,  $i$ 번째 계산에서 반올림을 하기 전의 값을  $\hat{v}(i)$ 라 하고, 다음과 같이 양자화 연산자  $Q[\cdot]$ 를 정의하자.

$$Q[\hat{v}(i)] = \hat{v}(i) + e(i) \quad (21)$$

여기에서  $e(i)$ 는 반올림에 의한 오차 벡터  $[e_x(i) e_y(i)]^T$ 이다. 고정소수점 연산에서 반올림 오차는  $|e_x(i)| \leq \epsilon$ ,  $|e_y(i)| \leq \epsilon$ 로 크기가 제한되는데,  $\epsilon$ 은 하드웨어 정밀도에 의해 결정된다. 고정소수점 표현에서 소수 부분이  $M$ 자리라면,  $\epsilon = 2^{-(M+1)}$ 이 된다. 각 단의 반올림 오차는 다음 단으로 전파되는데, 이 오차의 전파식을 다음과 같이 쓸 수 있다[9].

$$f(n) \equiv Q[\hat{v}(n)] - v(n) = e(n) + \sum_{i=0}^{n-1} B(i) e(i) \quad (22)$$

여기에서  $B(i) \equiv \prod_{j=i}^{n-1} P(j)$ 이다. 그러므로, 반올림 오차는 다음과 같이 제한된다.

$$\begin{aligned} |f(n)| &\leq |e(n)| + \sum_{i=0}^{n-1} |B(i) e(i)| \\ &\leq |e(n)| + \sum_{i=0}^{n-1} \|B(i)\| \cdot |e(i)| \\ &\leq \sqrt{2} \epsilon \left( 1 + \sum_{i=2}^{n-1} \|B(i)\| \right) \end{aligned} \quad (23)$$

4)  $\|\cdot\|$ 와  $|\cdot|$ 는 각각 행렬과 벡터의  $l_2$  norm을 뜻한다.

$\|(D-I)\| = \sqrt{(\cos \delta - 1)^2 + \sin^2 \delta} = 2 \sin |\delta|/2$ .

초기에는 반올림 오차가 없기 때문에  $e(0)=0$ 이고, 첫 번째 회전에서는 shift-down이 없으므로  $e(1)=0$ 이다.

위의 유도 과정에서 입력과 내부 계산 결과는 모두 같은 단어길이를 갖는다고 가정한 것인데, 이것은 하나의 연산기를 반복해서 사용하는 경우에 적합한 분석 방법이다. 디지털 통신 시스템에서는, 하드웨어를 줄이기 위하여 아날로그 입력 신호를 가능한 한 짧은 단어길이를 양자화하고, 연산에 의해 오차가 누적되는 것을 막기 위하여 내부 중간 계산 결과들은 그보다 긴 단어길이를 갖도록 한다. 그러므로 양자화 오차 분석도 이러한 하드웨어 구조에 맞게 수정해야 한다.

입력의 단어길이가  $L$ 이라고 할 때, 계산 도중 scale factor,  $K(n)$ , 만큼 커지는 값을 수용할 수 있도록 각 회전부의 내부 단어길이는 기본적으로 1 bit 부호 확장을 시켜야 한다[6]. CORDIC 알고리즘에서는 곱셈이 간단한 shift-down 연산들로 대체되었기 때문에 가장 아래 자리에 guard bit를 뒀으로써 반올림 오차를 줄일 수 있다. 이 때, guard bit의 수  $M_C$ 가 전체 shift-down 자리수  $\sum_{i=0}^{n-1} i$ 보다 클 필요는 없다. 그리고, 내부 단어길이는 전체 단 수보다는 커야만 shift-down해서 더하는 연산이 의미가 있게 된다. 그러므로, 내부 단어길이  $N_C$ 는 다음과 같은 범위를 가질 수 있다.

$$n \leq N_C = L + M_C + 1 \leq L + \frac{n(n-1)}{2} + 1 \quad (24)$$

$M_C$  개의 guard bit가 첨가되었을 때, 전체 shift-down 자리수가  $M_C$ 를 넘지 않는 동안은 반올림 오차가 발생하지 않는다. 즉,  $m(m-1)/2 \leq M_C$ 를 만족시키는 최대의  $m$ 에 대해서는 다음과 같은 식을 만족시킨다.

$$e(j) = 0 \quad j = 0, 1, 2, \dots, m \quad (25)$$

입력이 정수부로만 이루어져 있다면, 첨가되는  $M_C$  bit는 소수부로 볼 수 있다. 따라서, 반올림 오차한계는 단 수에 따라 다음과 같이 계산할 수 있다.

$n \leq m$ 일 때는 반올림 오차가 발생하지 않는다.

$$|f(n)| = 0 \quad (26)$$

$n = m + 1$ 이면, 반올림 오차가 마지막 단에서만 발생한다.

$$|f(n)| \leq \frac{\sqrt{2}}{2^{M_C+1}} \quad (27)$$

$n = m + 2$ 일 때는 전과되는 오차도 고려해야 한다.

$$|f(n)| \leq \frac{\sqrt{2}}{2^{M_C+1}} \left( 1 + \sum_{i=m+1}^n \|B(i)\| \right) \quad (28)$$

전체 양자화 오차를 해석하기 위하여,  $\tilde{v}(n)$  ( $= Q[\hat{v}(n)]$ )을 각 근사화 오차와 반올림 오차를 포함하는 계산 결과라 하자. 그러면,  $\tilde{v}(n)$ 은 다음과 같이 표현될 수 있다.

$$\tilde{v}(n) = v(n) + f(n) = Dv^*(n) + f(n) \quad (29)$$

그러므로, 전체 양자화 오차한계는 다음과 같다.

$$|\tilde{v}(n) - v^*(n)| \leq \|D - I\| \cdot |v^*(n)| + |f(n)| \quad (30)$$

이것은 CORDIC 알고리즘에서 발생하는 오차의 상한이므로, 하드웨어 비용과 성능의 trade-off를 고려하여 최적의  $n$ 과  $M_C$ 를 결정해야 한다. 이 해석 결과를 이용하여 그림 5에  $L=8$ 인 경우의 CORDIC 오차한계를 나타내었는데, 여기에서 오차한계는  $K(n)$ 으로 정규화하였다.

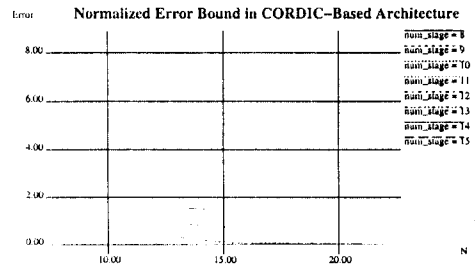


그림 5. CORDIC을 사용한 구조의 오차한계 ( $L=8$ )

3.1.2 각도 계산부(Angle Computation Block)의 양자화 영향

실제 하드웨어 구현에서는, 회전각  $A$ 가 먼저 디지털 각도  $Z_{i,n}$ 으로 양자화된 후 식 (15)의 기본각들의 선형 조합으로 근사된다.  $2^{B_c}$ 이  $2\pi$ 를 표현하도록 각도를  $B_c$  bit의 2의 보수로 정규화시킬 수 있다[6]. 그러면, 초기 각도 양자화 오차는 반올림을 사용하는 것으로 가정할 때 다음과 같이 제한된다.

$$|A - z_{i,n}| \leq \frac{\pi}{2^{B_c}} \quad (31)$$

하드웨어 구현을 위해서는 회전각  $A$  뿐만 아니라 기본각  $\{a(i)\}$ 도 양자화되어야 하고 실제 각도 계산은 이 값들을 사용하여 수행된다. 이에 의해 발생하는 오차는 회전 방향에 영향을 주어서 다른 값으로 수렴하게 할 수도 있다. CORDIC 알고리즘에서 각도 계산 결과  $z(n)$ 은  $n$  번의 반복 계산을 통해 다음과 같은 값을 갖는데, 이것은 0으로 수렴하도록 되어 있다.

$$z(n) = A - \text{sign}(A) \cdot 2 \cdot \arctan(1) - \sum_{i=0}^{n-1} \mu(i) a(i) \quad (32)$$

하지만, 실제 계산은 양자화된 값들로 수행되어 다음과 같은 결과를 얻는다.

$$\hat{z}(n) = z_{i,n} - \text{sign}(z_{i,n}) \cdot 2 \cdot \arctan(1) - \sum_{i=0}^{n-1} \hat{\mu}(i) \hat{a}(i) \quad (32)$$

여기에서  $\{\hat{a}(i)\}$ 는 양자화된 기본각들이고  $\{\hat{\mu}(i)\}$ 는  $\{\hat{a}(i)\}$ 를 사용하여 계산된 실제 회전방향 제어신호이다.  $i=0, 1, 2, \dots, n-2$ 에서 다음과 같은 부등식을 만족하면,  $\hat{z}(n)$ 은  $|\hat{z}(n)| \leq \hat{a}(n-1)$ 로 제한된다[9].

$$\hat{a}(i) - \sum_{j=i+1}^{n-1} \hat{a}(j) \leq \hat{a}(n-1) \quad (34)$$

반면에, 각 회전은 원래의 기본각  $\{a(i)\}$ 를 사용하여 수행되므로 실제 회전각  $A^*$ 는 다음과 같다.

$$A^* = \text{sign}(z_{i,n}) \cdot 2 \cdot \arctan(1) + \sum_{i=0}^{n-1} \hat{\mu}(i) a(i) \quad (35)$$

그러므로 실제 각 근사화 오차  $z^*(n)$ 는 다음과 같이 유도된다.

$$z^*(n) = A - \text{sign}(z_{i,n}) \cdot 2 \cdot \arctan(1) - \sum_{i=0}^{n-1} \hat{\mu}(i) a(i) \quad (36)$$

이것은 다음과 같이 실제 각도 계산 결과  $\hat{z}(n)$ 로 표현될 수 있다.

$$z^*(n) = \hat{z}(n) + A - z_{i,n} - \sum_{i=0}^{n-1} \hat{\mu}(i) [a(i) - \hat{a}(i)] \quad (37)$$

그러면, 실제 각 근사화 오차는 다음과 같이 제한된다.5)

$$|\delta| = |z^*(n)| \leq a(n-1) + |A - z_{i,n}| + \sum_{i=0}^{n-1} |a(i) - \hat{a}(i)| \quad (38)$$

하드웨어 구현을 위해서는, 이 각도 계산 오차도 고려되어야 한다. 그림 6에 모의 실험 결과에서 나타난

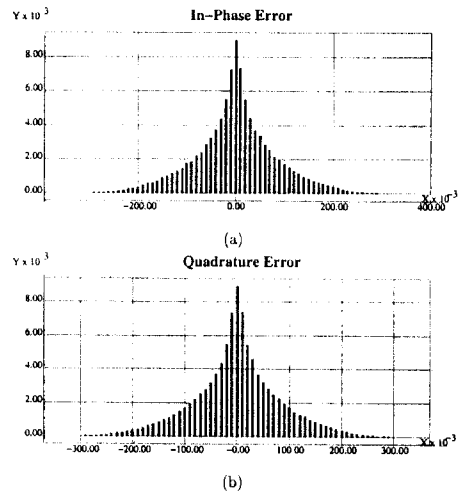


그림 6. CORDIC을 사용한 구조의 모의실험 결과에서 나타난 오차 histogram : (가) in-phase 성분 (나) quadrature 성분

5) 마지막 단의 각도 계산은 실제로 하드웨어로 구현할 필요가 없으므로  $\hat{a}(n-1) = a(n-1)$ 이라고 할 수 있다.

오차의 histogram을 보였다. 여기에서 정규화된 오차가 0.5 보다 작도록 하기 위하여  $L=8$ ,  $B_C=16$ ,  $n=10$ ,  $M_C=5$ 를 사용하였는데, 이것은 각도 계산 오차도 고려하여 결정된 값들이다.

### 3.2 ROM을 사용한 구조

ROM을 사용한 DDFS는 각도 양자화와 출력 양자화의 두 가지 오차 발생원이 있다. 각도  $A$ 는 다음과 같이 양자화된다.

$$A = Q[A] + \delta \quad (39)$$

$\delta$ 는 반올림에 의한 오차로서 각도 입력의 단어길이가  $B_R$  bit일 때  $|\delta| \leq \pi/2^{B_R}$ 로 제한된다. 그러면, DDFS의 sine/cosine과 출력은 다음과 같다.

$$\cos(A - \delta) + e_x \quad (40)$$

$$\sin(A - \delta) + e_y \quad (41)$$

여기에서  $e_x$ 와  $e_y$ 는 출력의 반올림 오차로서, DDFS의 출력 단어길이가  $M_R$  bit일 때  $|e_x| \leq \epsilon$ ,  $|e_y| \leq \epsilon$ 를 만족하며,  $\epsilon = 1/(2^{M_R} - 1)$ 이다.<sup>6)</sup>

Quadrature 복조기의 출력은 다음과 같다.

$$\begin{aligned} \begin{bmatrix} r_I \\ r_Q \end{bmatrix} &= \begin{bmatrix} \cos(A - \delta) + e_x & -\sin(A - \delta) - e_y \\ \sin(A - \delta) + e_y & \cos(A - \delta) + e_x \end{bmatrix} \begin{bmatrix} r_{real} \\ r_{imag} \end{bmatrix} \\ &= \begin{bmatrix} \cos(A - \delta) & -\sin(A - \delta) \\ \sin(A - \delta) & \cos(A - \delta) \end{bmatrix} \begin{bmatrix} r_{real} \\ r_{imag} \end{bmatrix} \\ &\quad + \begin{bmatrix} e_x & -e_y \\ e_y & e_x \end{bmatrix} \begin{bmatrix} r_{real} \\ r_{imag} \end{bmatrix} \\ &= \begin{bmatrix} \cos \delta & \sin \delta \\ -\sin \delta & \cos \delta \end{bmatrix} \begin{bmatrix} \cos A & -\sin A \\ \sin A & \cos A \end{bmatrix} \begin{bmatrix} r_{real} \\ r_{imag} \end{bmatrix} \\ &\quad + \begin{bmatrix} e_x & -e_y \\ e_y & e_x \end{bmatrix} \begin{bmatrix} r_{real} \\ r_{imag} \end{bmatrix} \quad (42) \end{aligned}$$

위 식은 벡터 형태로 다음과 같이 쓸 수 있다.

6) 1과 -1은 각각  $2^{M_R-1}-1$ 와  $-2^{M_R-1}+1$ 로 나타내어야 한다.

$$\begin{aligned} \tilde{v} &= \begin{bmatrix} \cos \delta & \sin \delta \\ -\sin \delta & \cos \delta \end{bmatrix} \begin{bmatrix} \cos A & -\sin A \\ \sin A & \cos A \end{bmatrix} r \\ &\quad + \begin{bmatrix} e_x & -e_y \\ e_y & e_x \end{bmatrix} r \\ &= v + Er \quad (43) \end{aligned}$$

여기에서  $\tilde{v} = [r_I \ r_Q]$ 이고  $r = [r_{real} \ r_{imag}]^T$ 이다. 위 식에서 첫째 항은 각도의 양자화 오차를 포함하고 있고, 둘째 항은 반올림 오차를 포함하고 있는데, 이 두 가지 항을 분리하여 분석한다.

$\delta=0$ 일 때  $v$ 의 이상적인 결과를  $v^*$ 라 하자. 그러면,  $v$ 의 실제 계산 결과는 다음과 같이 표현된다.<sup>7)</sup>

$$v = \begin{bmatrix} \cos \delta & \sin \delta \\ -\sin \delta & \cos \delta \end{bmatrix} v^* \equiv Dv^* \quad (44)$$

그러므로, 각도 양자화에 의한 오차는 다음과 같다.

$$v - v^* = (D - I)v^* \quad (45)$$

오차한계는 다음과 같이 추정된다.

$$\|v - v^*\| \leq \|(D - I)\| \cdot \|v^*\| \leq \frac{\pi}{2^{B_R}} \|v^*\| \quad (46)$$

입력 단어길이가  $L$ 일 때, 입력은  $|r_{real}| \leq 2^{L-1}$ 와  $|r_{imag}| \leq 2^{L-1}$ 로 제한되므로, 식 (46)의 오차한계는 다음과 같이 입력 단어길이를 표현할 수 있다.

$$\|v - v^*\| \leq \frac{\pi}{2^{B_R}} \|v^*\| \leq \sqrt{2}\pi \cdot 2^{L-B_R-1} \quad (47)$$

반올림 오차는 다음과 같이 제한됨을 쉽게 유도할 수 있다.

$$\|Er\| \leq \|E\| \cdot \|r\| \leq \frac{\sqrt{2}}{2^{M_R-1}} \cdot 2^{L-1} \quad (48)$$

7) 각도 양자화 오차를 해석할 때는 반올림 오차를 고려하지 않는다.



두 가지 오차를 포함하는 전체 양자화 오차한계는 다음과 같다.

$$\begin{aligned}
 |\tilde{v} - v^*| &\leq \|(D-I)\| \cdot |v^*| + E_{\tau} \\
 &\leq \sqrt{2} \cdot \pi \cdot 2^{L-B_k-1} + \frac{\sqrt{2}}{2^{B_k-1}} \cdot 2^{L-1} \quad (46) \\
 &= \sqrt{2} \cdot 2^{L-1} \left( \frac{\pi}{2^{B_k}} + \frac{\sqrt{2}}{2^{B_k-1}} \right)
 \end{aligned}$$

그림 7에  $L=8$ 인 경우의 오차한계를 계산한 결과를 보였다. 여기에서 오차한계는  $2^{M_k-1}/(2^{M_k}-1)$ 를 곱함으로써 정규화하였다. Random한 입력과 위상으로 모의 실험하여 이 분석을 확인하였는데, 그림 8에 모의 실험 결과에서 나타난 오차의 histogram을 보였다. 정규화된 오차가 0.5 보다 작도록 하기 위하여  $L=8$ ,  $B_R=11$ ,  $M_R=10$ 로 결정된 단어길이들을 사용하였다.

#### IV. 하드웨어 비용/속도 분석 및 비교

CORDIC을 사용한 구조와 ROM을 사용한 구조의 하드웨어 효율성을 비교하기 위하여 각각의 하드웨어 비용과 속도를 분석하였다. 공정한 비교를 위해서는 합리적인 성능 평가 기준이 마련되어야 하는데, 여기에서는 quadrature 복조기를 통과했을 때 신호의 유효 숫자를 유지시킬 수 있는 정도의 성능을 갖도록 설계 사양을 맞추었다. 그러면, 계산된 결과는, 그림 9에 보인바와 같이, 이상적인 복조를 한 후에 같은 단어 길이로 양자화한 것과 동등하게 된다. 출력은 mixer 이득으로 정규화한 것으로 가정하였는데, CORDIC을 사용한 구조에서는  $K(n)$ 이고 ROM을 사용한 구조에

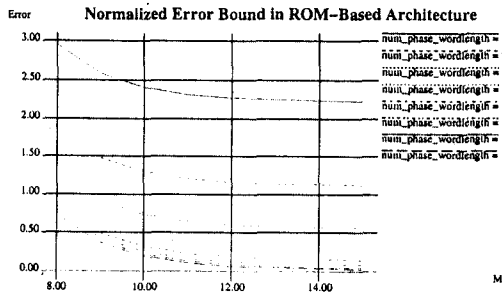


그림 7. ROM을 사용한 구조의 오차한계 ( $L=8$ )

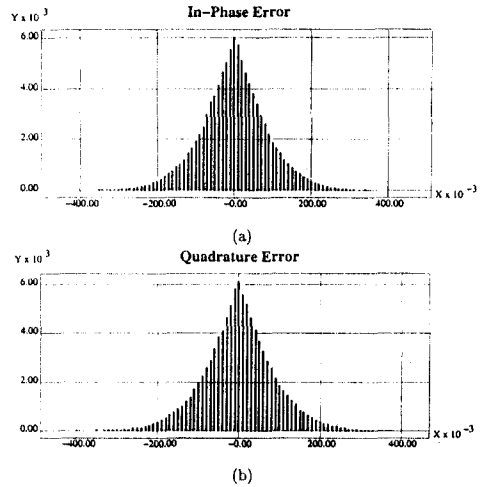


그림 8. ROM을 사용한 구조의 모의실험 결과에서 나타난 오차 histogram : (가) in-phase 성분 (나) quadrature 성분

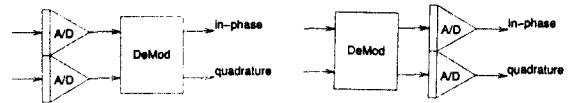


그림 9. 목표 오차한계를 정하기 위한 양자화 모델

서는  $2^{M_k-1}/(2^{M_k}-1)$ 이다.

#### 4.1 하드웨어 비용

하드웨어 비용을 추정하기 위하여 VTI 1.0  $\mu\text{m}$  cell library를 사용하였다. ROM과 곱셈기는 규칙적인 구조로 효율적으로 구현 가능하기 때문에 transistor 수나 gate 수는 공정한 비교 대상이 못 된다. 그래서, macro-cell들이 차지하는 면적을 cell compiler library를 참조하여 표 1와 같이 계산하였다. 여기에서 단위는  $\lambda (=0.5\mu\text{m})$ 이다[12][13]. 그림 4-(가)의 ROM을 사용한 구조를 그대로 구현하자면, sine과 cosine 값을 저장하기 위하여 두 개의 ROM이 필요하고 복소수 연산을 위하여 네 개의 곱셈기와 두 개의 덧셈기가 필요하다. 이 때 sine/cosine파의 대칭성을 이용하여 ROM의 크기는 정밀도에 영향을 주지 않으면서도 1/8로 줄일 수 있다. 즉, 0에서  $\pi/4$ 까지의 sine과 cosine 값들만

저장하면 나머지는 대칭성을 이용하여 구할 수 있으므로, ROM의 주소가  $B_R$  bit에서  $B_R-3$  bit로 준다. 또한, 출력의 부호는 각도 입력으로부터 정해질 수 있어서 부호를 나타내는 bit가 필요 없으므로, 출력의 단어길이가  $M_R$  bit에서  $M_R-1$  bit로 준다. 크기가 클 때는 ROM의 크기를 줄이기 위한 일반적인 기술들을 더 쓸 수 있지만, 부가 회로가 첨가되어서 추정의 일반화가 어렵기 때문에, 여기에서는 더 이상의 ROM의 크기를 줄이는 기술은 사용하지 않는 것으로 가정하였다. 곱셈기는 출력 정밀도에 큰 영향을 미치므로 크기를 줄일 수 없지만, 덧셈기의 크기는 반올림 회로를 이용하여 다음 계산에 필요한 만큼의 정밀도로 줄일 수 있다. 표 2에 ROM을 사용한 구조에 필요한 하드웨어를 요약하였다.

CORDIC을 사용한 구조는 각 회전부와 각도 계산부의 두 개의 블록으로 구성된다. 각 회전부는 덧셈기/뺄셈기 (adder/subtractor) 배열과 고정된 hard-wired shift들로 구성되어 있다. 각도 계산부도 덧셈기/뺄셈기 배열로 구성되는데, 여기에는 고정된 값과의 계산

표 1. 하드웨어 macrocell들의 면적 추정

$2^N \times M$ -bit ROM	$\begin{aligned} \text{col} &= 3 \\ \text{row} &= 2^N \cdot \text{col} \\ x\_dec &= \begin{cases} 30.5, & N - \text{col} - 2 \leq 2 \\ 324.0, & 2 < N - \text{col} - 2 \leq 4 \\ 340.5, & 4 < N - \text{col} - 2 \leq 6 \end{cases} \\ \text{Height} &= 8.5 \cdot \text{row} + 647 + 17 \\ \text{Width} &= 67.5 \cdot M \cdot 2^{N-3} \\ &\quad + 16(N - \text{col} - 2) \\ &\quad + x\_dec + 172.5 \end{aligned}$
$N \times M$ multiplier	$\begin{aligned} \text{Height} &= 324(\lceil M/2 \rceil - 2) + 794 \\ \text{Width} &= 167(N + 1) + 536.5 \end{aligned}$
$N$ bit adder	$\begin{aligned} \text{Height} &= 86.5 \\ \text{Width} &= 170N \end{aligned}$
$N$ bit adder/subtractor	$\begin{aligned} \text{Height} &= 86.5 \\ \text{Width} &= 200N \end{aligned}$

표 2. ROM을 사용한 구조에서 요구되는 하드웨어

Hardware Unit	Number of Units	Size
ROM	2	$2^{B_R-3} \times (M_R - 1)$
Multiplier	4	$L \times M_R$
Adder	2	$N_R \leq L + M_R$

만 수행하면 되므로 전가산기보다 간단한 논리회로로 구현할 수 있다. 그리고, 첫번째와 마지막 단계에서는 계산이 필요 없고, 나머지 단계에서도  $B_C-3$  bit 덧셈/뺄셈만이 필요 하대기. 표 3에 CORDIC을 사용한 구조에 필요한 하드웨어들을 요약하였다.

정밀도의 요구사항이 정해지면, 제 3장의 양자화 효과에 대한 분석을 이용하여 단어길이들을 결정할 수 있다. 표 4에 두 가지 구현 방법에 대해 최적의 단어 길이들을 구해서 정리하였다. 이 때 필요한 하드웨어 비용을 추정하여 표 5에 정리하였다. 여기에서 ROM을 사용한 구조의 출력 단어길이는 CORDIC을 사용한

표 3. CORDIC을 사용한 구조에서 요구되는 하드웨어

Hardware Unit	Number of Unit	Size
Adder/Subtractor	$2 \times n$	$N_C$
Adder/Subtractor(Fixed)	$n-2$	$B_C-2$

표 4. 디지털 quadrature 복조기의 최적 단어길이

Input WL	ROM-Based			CORDIC-Based			
	Precision 0.5			Precision 0.5			
$L$	$B_R$	$M_R$	$N_R$	$B_C$	$n$	$M_C$	$N_C$
3	6	5	8	9	5	3	7
4	7	6	10	10	6	4	9
5	8	7	12	11	7	4	10
6	9	8	14	14	8	4	11
7	10	9	16	13	9	5	13
8	11	10	18	14	10	5	14
9	12	11	20	15	11	5	15
10	13	12	22	17	12	5	16

표 5. 디지털 quadrature 복조기의 하드웨어 비용 (단위:  $\lambda^2$ )

$L$	ROM-Based			CORDIC-Based		
	ROM	CmplxMult	Total	AngleRot	AngleComp	Total
3	963020	5121556	6084576	1211000	415200	1626200
4	1088919	5710768	6799687	1868400	553600	2422000
5	1232668	8726740	9959408	2422000	865000	3287000
6	1414956	9532384	10947340	3044800	1349400	4394200
7	1680000	13197652	14877652	4048200	1453200	5501400
8	2156544	14219728	16376272	4844000	1799200	6643200
9	2984968	18534292	21519260	5709000	2335500	8044500
10	4679592	19772800	24452392	6643200	2941000	9584200

구조와 같게 되도록 덧셈기의 단어길이를 조정하였다. CORDIC을 사용한 디지털 quadrature 복조기가 ROM을 사용한 방식에 비해 약 1/3 정도의 작은 면적을 갖는 것을 볼 수 있다.

4.2 동작 속도

Macrocell들의 동작 속도를 추정하기 위하여 표 6의 VTI 1.0  $\mu\text{m}$  cell library를 사용하였다[12][13]. ROM과 multiplier의 지연시간은 표에 주어진 바와 같이 크기에 대한 식으로 주어지지만, adder/subtractor의 지연시간은 data path library로부터 추정된 것이기 때문에 단어길이가 긴 경우 조금 과장되게 추정될 가능성이 있다.

표 6. Speed estimation of the hardware macrocells

$2^N \times M$ - bit ROM	precharge time (표 7) +access time (표 8)
$N \times M$ multiplier	propDelay for square multipliers ( $N=M$ ) = $(N^{1/2} \times 7.85) + 2.9$ propDelay for non-square multipliers ( $N > M$ ) = (square propDelay) $\times (M/N)^{1/4}$
$N$ - bit adder/subtractor	$N$ (ns)

표 7. ROM precharge time

N	M			
	4	8	16	32
6	11	12	13	18
7	12	13	14	19
8	13	14	17	20
9	14	15	18	21
10	18	19	20	23

표 8. ROM access time

N	M			
	4	8	16	32
6	6	7	8	11
7	6	7	8	11
8	6	7	8	11
9	8	8	9	12
10	14	14	15	18

ROM을 사용하는 구조에서는 ROM의 주소 입력으로부터 복소수 곱셈기의 출력까지가 임계경로 (critical path)가 된다. 그러므로 전체 지연시간은 ROM의 출력 지연시간과 하나의 곱셈기, 하나의 덧셈기의 지연시간들을 더한 시간으로서 표 9에 나타내었다.

표 9. Speed of the digital quadrature mixer (ns)

Input WL L	ROM-Based			CORDIC-Based		
	ROM	CmplxMult	Total	AngleRot	AngleComp	Total
6	19	25.5	44.5	35	39	43
7	20	27.0	47.0	44	49	56
8	23	28.5	51.5	54	60	69
9	26	29.5	55.5	65	72	85
10	34	31.0	65.0	77	85	109

CORDIC을 사용하는 구조에서의 임계경로는 그림 10에 보인 바와 같이 상당히 복잡한데, 그림의 각 회전부의 각 단은 X block과 Y block을 번갈아서 나타낸다. 각 회전부는 right shift 동작을 하므로 그림에서와 같이 앞 단의 최상위 자리에서부터 현재 단의 최상위 자리까지의 carry 전파가 필요하다. 그리고 단이 나아갈수록 전파되는 길이가 길어진다. 각 계산부에서는 각 단에서 앞 단으로부터 계산된 부호가 필요하므

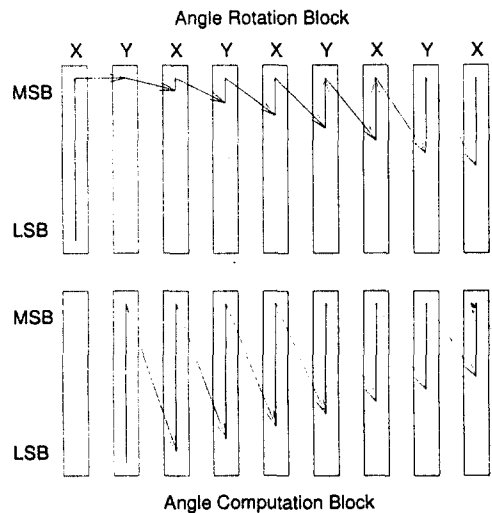


그림 10. Critical path of the CORDIC-based architecture

로 앞 단의 계산이 끝날 때까지 기다려야 한다. 그런데 단이 나아갈수록 계산되는 각도가 작아지므로 부호를 검출하는데 걸리는 시간이 짧아지게 된다. 각회전부는 각 단에서 각계산부의 계산이 끝나야만 동작을 할 수 있다. 그러므로 전체적인 임계경로는 이 두 부분의 더 긴 경로로 이루어진다. 표 9에 추정된 결과는 이러한 점을 고려한 값들이다.

Ripple-carry adder를 사용할 경우 CORDIC을 사용하는 구조는 ROM을 사용하는 구조에 비해 동작 속도가 느리다. 이것은 높은 precision을 요구하는 경우 더욱 뚜렷해진다. 하지만 CORDIC processor는 동작 속도를 증가시키기 위한 여러 가지 방법들이 있다. 각회전부의 carry 전파는 carry-save adder를 사용하여 면적을 두 배 정도로 늘림으로써 완전히 없앨 수 있다. Pipelining을 함으로써 CORDIC processor는 각 계산부의 한 단 정도의 throughput을 낼 수도 있다. 또한 direction sequence를 계산하는 새로운 알고리즘들을 사용하면 bit-level까지 pipelining을 할 수도 있다[14]. 반면에 ROM을 사용하는 구조는 pipelining을 하더라도 ROM의 속도에 의해 제한이 되게 된다. Feedback loop 안에 들어있어서 pipelining을 할 수 없는 경우에도 새로운 각계산 알고리즘과 각회전부에서 carry-save adder를 사용함으로써 매우 높은 속도를 낼 수 있다.

### V. CORDIC을 이용한 디지털 Costas Loop의 VLSI 구현

새롭게 제안된 CORDIC을 사용한 디지털 quadrature 복조기를 이용하여 BPSK 대역확산 통신용 수신기의 일부분인 디지털 Costas loop을 구현하기 위한 VLSI를 설계하였다. 디지털 Costas loop은 BPSK 변조된 신호의 동기 복조를 위해 효율적으로 사용될 수 있는 일종의 위상 추적 회로로서 그림 11에 블록도를 보였다. 이를 모두 디지털 회로로 구현하기 위하여 각 부분을 제한된 단어길이를 갖는 디지털 연산자로 대체하여 모의실험을 통해 최적의 설계사양들을 정하였다.

그림 11의 점선 부분이 디지털 quadrature 복조기로서, 병렬 CORDIC 프로세서 구조를 사용하여 구현하였다. 이 시스템은 그림 2-(가)와 같이 실수 입력이 들어오며, 입력 단어길이  $L$ 은 모의실험을 통해 4 bit로 결정하였다. 이 시스템에는  $y_{in}$ 이 0이므로 식 (20)의 각 근사화 오차한계는 다음과 같이 수정되어야 한다.

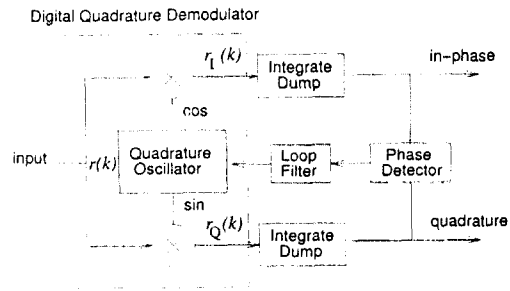


그림 11. 디지털 Costas loop 블록도

$$|v(n) - v^*(n)| \leq a(n-1) |v^*(n)| \leq a(n-1) K(n) 2^{L-1} \quad (50)$$

수정된 오차한계를 이용하여 최적의 단어길이를 결정하는데, 내부 단어길이를 하나 늘리는 것보다 단수를 늘리는 것에 하드웨어 비용이 더 필요하므로 단수  $n$ 을 먼저 결정해야 한다. 여기에서는 정규화된 CORDIC 오차가 0.5 보다 작게 하는 것을 목표로 하여  $n=6, N_C=8, B_C=8$ 로 결정하였다.

양자화 오차 해석 결과에 따르면, 그림 3의 CORDIC 프로세서 구조는 더욱 간단해질 수 있다. 첫번째 ADD/SUB 블록은 두 개의 4 bit 수를 더하거나 빼야 하므로, 5 bit의 전가산기로 충분하다. 같은 식으로 두번째, 세번째 단은 각각 6 bit, 8 bit의 전가산기가 필요하다. 하지만, down-shift된 bit들의 연산에는 전가산기를 사용할 필요가 없다. 그림 12에 세 번째 ADD/SUB 블록에서 전가산기 대신 반가산기를 사용한 회로를 보였다. 내부 단어길이를 8 bit로 정했기 때문에, 나머지 다른 ADD/SUB 블록들은 반올림 회로를 갖는 8 bit 전가산기로 구현해야 한다. 그림 13에 그 회로를 보였다. 각도 계산부는 항상 계산된 값의 크기가 앞 단 보다 작기 때문에, 모든 단에서 8 bit의 덧셈,

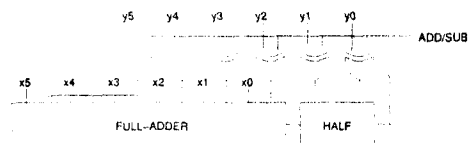


그림 12. Half-adder를 사용한 ADD/SUB 블록

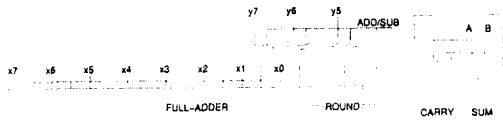


그림 13. 반올림 회로를 포함한 ADD/SUB 블럭

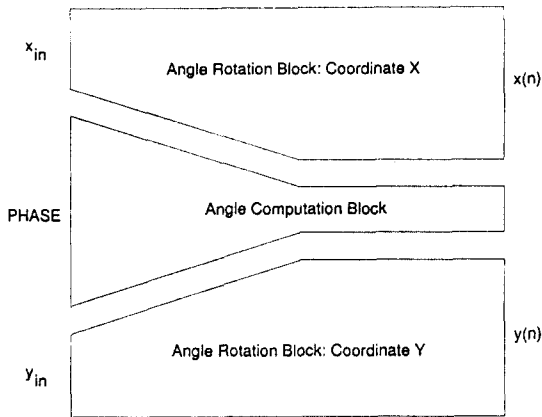


그림 14. 디지털 quadrature 복조기의 layout을 위한 floor plan

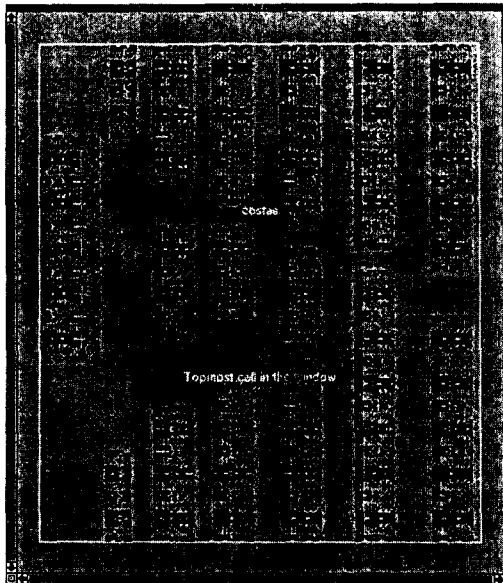


그림 15. 디지털 Costas Loop 회로의 layout

셀셈을 수행할 필요가 없다. 그리고 미리 정해진 값을 더하거나 빼기 때문에 전가산기 대신 간단한 회로로 대체할 수 있다[6]. 이러한 점들을 이용하여 그림 14와 같은 floor plan으로 compact한 layout이 가능하다.

그림 15에 전체 Costas loop 회로의 layout 결과를 보였다. 현대 0.8  $\mu\text{m}$  CMOS 공정을 사용하여 면적은 1.5 mm $\times$ 1.8 mm이고 동작 속도 (IF 주파수 $\times$ 4)는 50 MHz가 될 것으로 기대된다.

## VI. 결론

대역확산 통신은 물론 각종 유무선 디지털 통신 시스템의 핵심 부품인 quadrature 복조기를 디지털로 구현하기 위한 새로운 구조에 대하여 연구하였다. CORDIC 알고리즘의 circular rotation을 사용하여 sine/cosine파의 생성과 주파수 mixing을 한꺼번에 수행하도록 하였다. 기존에 주로 사용하던 ROM과 곱셈기를 사용하는 방법에 비하여 구조가 매우 규칙적이어서 설계가 용이하고 작은 면적에 구현할 수 있다. 목표 성능이 주어졌을 때 최적의 하드웨어 사양들을 결정할 수 있도록 제한된 단어길이에 대한 영향을 분석하였다. 비교를 위하여 ROM을 사용하는 방법에 대해서도 제한된 단어길이의 영향을 분석하였다. 하드웨어 효율을 비교하기 위하여 standard cell library를 사용하여 각각의 구조의 면적을 추정하였다. CORDIC을 사용하는 구조가 ROM과 곱셈기를 사용하는 구조에 비해 약 1/3 정도의 면적을 차지하는 반면 동작 속도에서는 불리함을 알 수 있었다. 하지만, CORDIC processor의 동작 속도는 pipelining 기법이나 새로운 빠른 알고리즘들로 극복될 수 있다. 제안하는 구조의 디지털 quadrature 복조기를 사용하여 디지털 Costas loop를 매우 작은 면적에 주문형 VLSI로 구현하였다.

## 참고 문헌

1. Edward A. Lee and David G. Messerschmitt, *Digital Communication*, Kluwer Academic Publishers, 101 Philip Drive, Assinippi Park, Norwell, MA 02061, USA, 1988.
2. John G. Proakis, *Digital Communications*, McGRAW-HILL, 1989.

3. 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.
4. Rajeev Jain, Henry Samueli, Paul T. Yang, Charles Chien, Gloria G. Chen, Linda K. Lau, Bong-Young Chung, and Etan G. Cohen, "Computer-aided design of a BPSK spread-spectrum chip set," *IEEE J. Solid-State Circuits*, vol. 27, no. 1, pp. 44-58, Jan. 1992.
5. Loke Kun Tan and Henry Samueli, "A 200 MHz quadrature digital synthesizer/mixer in 0.8- $\mu$ m CMOS," *IEEE J. Solid-State Circuits*, vol. 30, no. 3, pp. 193-200, Mar. 1995.
6. 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.
7. Henry T. Nicholas III and Henry Samueli, "A 150-MHz direct digital frequency synthesizer in 1.25- $\mu$ m CMOS with -90-dBc spurious performance," *IEEE J. Solid-State Circuits*, vol. 26, no. 12, pp. 1959-1969, Dec. 1991.
8. Avanindra Madiseti, Alan Kwentus, and Jr. Alan N. Willson, "A sine/cosine direct digital frequency synthesizer using an angle rotation algorithm," in *ISSCC 1995 Digest of Technical Papers*, Feb. 1995, pp. 262-263.
9. Yu Hen Hu, "The quantization effects of the CORDIC algorithm," *IEEE Trans. on Signal Processing*, pp. 834-844, Apr. 1992.
10. Yu Hen Hu, "CORDIC-based VLSI architectures for digital signal processing," *IEEE Signal Processing Magazine*, pp. 16-35, July 1992.
11. STANFORD TELECOM, *Digital, Fast Acquisition, Spread Spectrum Burst Processor STEL200A*, ASIC Custom Products Division, 1993.
12. VLSI Technologies, Inc., *1-Micron Cell Compiler Library*, Nov. 1991.
13. VLSI Technologies, Inc., *VSC370 Library*, Nov. 1991.

14. Seunghyeon Nahm and Wonyong Sung, "A fast direction sequence generation method for CORDIC processors," in *Proc. IEEE ICASSP '97*, Apr. 1997.



남 승 현(Seung Hyeon Nahm) 정회원  
 1969년 2월 22일생  
 1991년 : 서울대학교 제어계측공학과 졸업(공학사)  
 1993년 : 서울대학교 대학원 제어계측공학과 졸업(공학석사)  
 1998년 : 서울대학교 대학원 전기공학부 졸업(공학박사)  
 1997년 10월 ~ 현재 : 고등기술연구원 전자통신연구실 선임연구원  
 ※주관심분야: 디지털 신호처리, 디지털 통신 시스템의 구현  
 e-mail : namsh@iae.re.kr



성 원 웅(Won Yong Sung) 정회원  
 1978년 2월 : 서울대학교 전자공학과 졸업(공학사)  
 1980년 2월 : 한국과학원 전기 및 전자공학과 졸업(공학석사)  
 1980년 ~ 1983년 : 금성사 중앙연구소  
 1987년 7월 : 미국 University of California, Santa Barbara 전기 및 컴퓨터공학과 졸업(공학박사)  
 1989년 2월 ~ 현재 : 서울대학교 전기공학부 및 반도체공동연구소 부교수  
 1993년 ~ 1994년 : Comdisco Systems (현 Alta Group) 기술 고문, 반도체공동연구소 설계연구부장  
 ※주관심분야: 병렬처리 컴퓨터, VLSI를 이용한 고속 신호 처리  
 e-mail : wysung@dsp.snu.ac.kr