

직접 디지털 주파수 합성을 위한 효율적인 ROM 압축 방법

(Effective ROM Compression Methods for Direct Digital Frequency Synthesis)

이진철[†] 신현철^{**}
(Jinchoul Lee) (Hyunchul Shin)

요약 본 논문에서는 디지털 주파수 합성기의 구조에 대하여 연구하였다. 디지털 주파수 합성기는 높은 스펙트럼 순도로 빠른 주파수 전환이 가능하고, 현대의 확산 스펙트럼 무선통신 시스템에 널리 사용된다. 롬 기반의 디지털 주파수 합성기는 싸인 파형의 크기를 저장한 롬 테이블을 사용한다. 본 논문에서는 롬의 크기를 줄이는 세 가지 새로운 기술을 제안하였다. 새로운 기법 중 한 가지는 여러 개의 계층적 구조를 사용하였다. 다른 기법들은 계층적 롬 구조를 간단한 보간 기법으로 결합하였다. 이러한 기법으로 12 비트의 싸인 파형을 생성하였다. 실험 결과 새롭게 제안한 기법은 기존 방법[1]에 비해 ROM 크기를 24%까지 줄일 수 있다.

키워드 : 직접 디지털 주파수 합성, 롬 압축

Abstract An architecture of direct digital frequency synthesizers (DDFS) is studied in this paper. The Direct digital frequency synthesizers (DDFS) provide fast frequency switching with high spectral purity and are widely used in modern spread spectrum wireless communication systems. ROM-based DDFS uses a ROM lookup table to store the amplitude of a sine wave. In this paper, we suggest three new techniques to reduce the ROM size. One new technique uses more number of hierarchical levels in ROM structures. Another techniques use simple interpolation techniques combined with hierarchical ROM structures. A 12 bit sine wave is generated by using these techniques. Experimental results show that the new proposed techniques can reduce the required ROM size by up to 24%, when compared to that of a resent method[1].

Key words : Direct Digital Frequency Synthesizer, ROM Compression

1. 서론

주파수 합성기는 기준 주파수로부터 다양한 주파수 신호를 발생하는 장치로 현대 통신 시스템에 널리 사용된다. 기존의 주파수 합성기는 PLL(Phase Locked Loop)을 바탕으로 구현되었는데, 이를 이용한 주파수 합성기는 본질적으로 피드백 루프를 가지고 있기 때문에 빠른 주파수 변환을 하기 어렵다. 이에 새로이 등장한 개념이 Direct Digital Frequency Synthesizer (DDFS)이다.

DDFS는 주어진 위상 값으로부터 직접 주파수 신호

를 발생시키고 피드백 루프가 없기 때문에 고속으로 주파수 변환을 하는데 적합하다.

DDFS에 사용되는 ROM의 크기를 줄이면 면적의 감소는 물론 소모 전력이 줄고 동작속도가 증가한다. 본 논문에서는 ROM을 사용하는 DDFS에서 ROM의 크기를 압축하는 기술을 제안한다.

2. ROM을 사용하는 DDFS

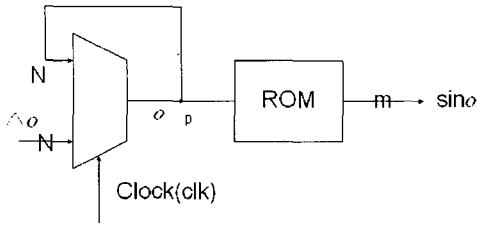
DDFS는 기본적으로 phase accumulator와 ROM table로 구성되어 있다[2]. 그림 1(a)의 DDFS의 기본 구조에서 보면 N bit의 frequency input word $\Delta\theta$ 가 clock 주파수 f_{clk} 에 의해 선형 위상(θ)을 생성하고 P bit의 accumulator 출력이 sine 파형을 생성하기 위한 ROM lookup table의 Address로 사용되게 된다. 식 (1)과 (2)는 DDFS의 minimum frequency resolution

[†] 비회원 : 삼성전자 통신연구소 연구원
jinchoul.lee@samsung.com

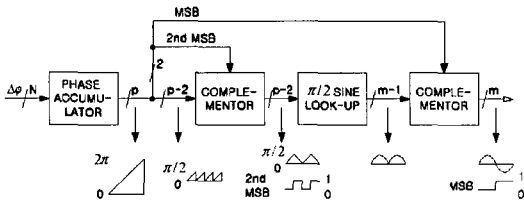
^{**} 종신회원 : 한양대학교 전자전기계어계측공학과 교수
shin@hanyang.ac.kr

논문접수 : 2004년 5월 19일

실사완료 : 2004년 6월 22일



(a) DDS의 기본 구조



(b) Quadrant Method를 이용한 DDS

그림 1 DDS의 구조

$$\Delta f = \frac{f_{clk}}{2^N} \quad (1)$$

$$f_{out} = \frac{\Delta \phi f_{clk}}{2^N} \quad (2)$$

과 output frequency이다.

그림 1(b)의 구조는 일반적으로 사용되고 있는 quadrant method를 이용한 DDS의 구조이다. quadrant method는 sine 파형의 대칭성을 사용하여 $\pi/2$ rad의 sine 파형만을 이용해서 2π rad의 sine 파형을 생성하는 방법이다. 이렇게 함으로써, ROM lookup table의 용량을 줄일 수 있다[3].

예를 들어 그림 1(b)의 DDS가 32 bit의 frequency input word와 12 bit의 output data로 구성되어 있다고 하자($p = 14$ and $m = 11$).

우선 잘 알려진 Sunderland[4]의 ROM 압축 방법을 이용하여 ROM의 크기를 줄이는 방법을 설명한다. Sunderland 방식은 ROM의 address로 들어오는 sine 파형의 위상을 $\phi(n) = \alpha + \beta + \gamma$ 로 표현한다. 이것은 12 bit의 위상을 최 상위 4 bits인 α , 중간 4 bits인 β , 최 하위 4 bits인 γ 로 나누어 표현하는 것이다. 이 sine 함수는 식 (3)과 같이 표현된다.

$$\begin{aligned} \sin\left(\frac{\pi}{2}(\alpha + \beta + \gamma)\right) &= \sin\left(\frac{\pi}{2}(\alpha + \beta)\right)\cos\left(\frac{\pi}{2}\gamma\right) + \cos\left(\frac{\pi}{2}(\alpha + \beta)\right)\sin\left(\frac{\pi}{2}\gamma\right) \\ &\equiv \sin\left(\frac{\pi}{2}(\alpha + \beta)\right) + \cos\left(\frac{\pi}{2}\alpha\right)\sin\left(\frac{\pi}{2}\gamma\right) \end{aligned} \quad (3)$$

이 방법은 ROM을 Coarse ROM $\sin\left(\frac{\pi}{2}(\alpha + \beta)\right)$ 과 Fine ROM $\cos\left(\frac{\pi}{2}\alpha\right)\sin\left(\frac{\pi}{2}\gamma\right)$ 으로 나누어 표현함으

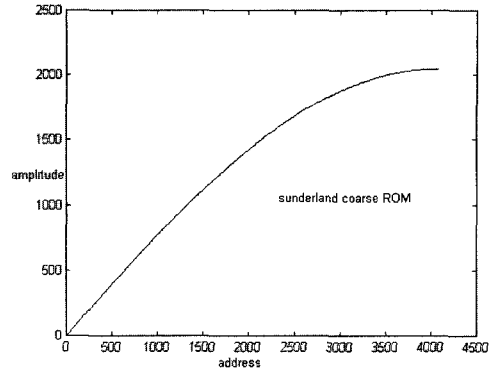


그림 2 Sunderland 방법의 Coarse ROM

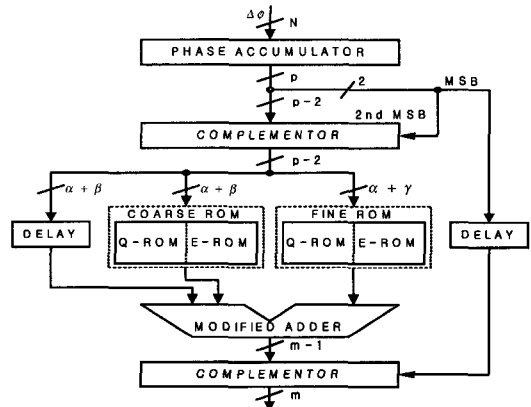


그림 3 DDS 블록도

로써 크기를 줄일 수 있다. 그림 2는 Sunderland 방식에서 Coarse ROM data의 크기를 나타낸다.

이 방법은 그림 3과 같이 ROM을 $2^{\alpha+\beta}$ word의 coarse ROM과 $2^{\alpha+\gamma}$ word의 Fine ROM으로 나누어 표현함으로써 크기를 줄일 수 있다.

다음으로 sine phase difference algorithm[5]은 식 (4)와 같은 방식을 Sunderland coarse ROM에 사용하여서 amplitude를 줄인다.

$$f(\phi) = \sin\left(\frac{\pi\phi}{2}\right) - \phi \quad (4)$$

그림 4는 Sunderland 방식 coarse ROM의 값인 $\sin\left(\frac{\pi}{2}(\alpha + \beta)\right)$ 를 대신하여서 $\sin\left(\frac{\pi}{2}(\alpha + \beta)\right) - (\alpha + \beta)$ 값을 저장하면, 결과적으로 신호의 최대 amplitude가 1/4 이하로 줄어들기 때문에, ROM에 저장되는 data의 크기가 2 bits 줄어드는 것을 보여준다. 이 방법에서는 추후에 data에서 빠진 $(\alpha + \beta)$ 값을 더해주는 덧셈기가 필요하다.

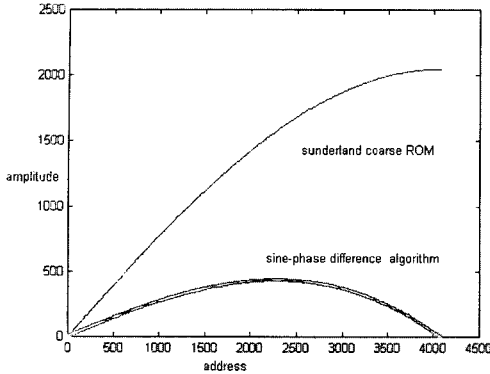


그림 4 Sine_phase difference algorithm

[1]에서는 ROM의 압축 정도를 높이기 위하여 Sunderland 방식[4]을 개선하여, 그림 3에서 볼 수 있듯이 두 ROM을 압축하기 위해서 각 ROM을 quantization ROM(Q-ROM)과 error ROM(E-ROM) 두 부분으로 나눈다. Q-ROM은 각 ROM에 MSB 몇 bit를 저장하고, E-ROM은 원래의 ROM과 Q-ROM 사이의 error 값을 저장한다[1].

그밖에 DDFS(Direct Digital Frequency Synthesizer)의 구조로는 Taylor Series를 이용한 방법과 CORDIC에 기초한 방법 등이 있다. Taylor Series Approximation을 이용한 방법으로는 sine 함수를 Taylor 급수 전개하여 3~4항까지만 구하는 방식으로 일부는 곱셈기를 사용하고 일부는 ROM에서 lookup하는 형태로서 uncompressed 방식에 비해 53:1 정도로 ROM을 압축 할 수 있다[6]. Taylor Series expansion 방법은 $\sin x = \sin(a) + k_1(x-a)\cos(a) - k_2(x-a)^2\sin(a) / 2 + \dots$ 의 전개식을 이용한 방법으로 이 방식의 장점은 ROM을 이용하는 방식에 비해 작은 ROM을 갖게 되고 작은 ROM을 사용함으로써 빠른 동작을 할 수 있다는 점이다. 단점으로는 곱셈기가 추가로 필요하고 sine 하나만 구할 때도 cosine을 저장하기 위한 ROM이 필요하다.

CORDIC-based architecture에서는 sine을 ROM에 의한 lookup 방식을 사용하지 않고, 직접 CORDIC 연산을 통해 구한다[7].

이 방식의 장점은 ROM의 크기에 제한을 받지 않고 원하는 정밀도를 얻을 수 있고 sine과 cosine을 동시에 발생시킬 수 있고 파이프라인 사용으로 동작 속도를 향상시킬 수 있다는 점이며 단점은 회로가 복잡해지고 전력 소모가 크다는 것이다.

3. 새로운 ROM 압축 방법들

3.1 Hierarchical ROM Division Technique(HRDT)

[1]에 따라 나누어진 coarse ROM의 Q ROM을 CQ ROM과 CE ROM으로 다시 한번 나누어서 표현함으로써 ROM의 크기를 줄였고 fine ROM을 FQ와 FE ROM으로 나누어 구현 방식을 그림 5와 같이 변형시킴으로써 전체적인 ROM의 크기를 줄이는 방법을 사용하였다. 표 1에서는 기존의 방법과 ROM의 크기를 비교하였다. 본 HRDT 방법은 [1]보다 ROM의 bit수를 7% 감소시킨다.

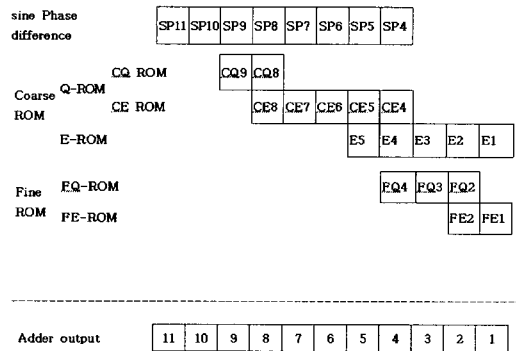


그림 5 HRDT의 ROM 데이터

표 1 제안한 HRDT의 ROM 크기

	12 bit output			
	[1]		제안한 HRDT	
Coarse ROM	Q	2 ⁶ *6	CQ	2 ³ *2
	E	2 ⁶ *5	CE	2 ⁶ *5
Fine ROM	Q	2 ⁵ *2	FQ	2 ⁶ *3
	E	2 ⁸ *3	FE	2 ⁸ *2
Total ROM size	2496		2320	

3.2 ROM Decomposition with 0th Order Approximation(RD0)

Sunderland 방법[4]에 따라 나누어진 Coarse ROM에 sine_phase difference algorithm을 대신하여, π/2 구간을 8블록으로 나누어서 각 블록의 최초 값을 저장하여 그 값을 뺀 차만 저장한다. 각 블록은 식(5)로 표현된다.

$$f(\phi) = \sin\left(\frac{\pi\phi}{2}\right) - \sin\left(\frac{\pi * n}{2 * 8}\right) \quad (n=0,1,2...7) \quad (5)$$

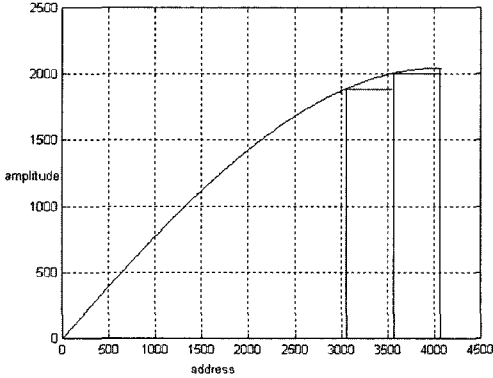
실험결과 6번째 블록까지는 sine_phase difference algorithm를 사용하여 quantization ROM과 error ROM으로 압축하는 것이 유리하였다. 그러나 7번째, 8번째 블록에서는 RD0 방법으로 Q ROM과 E ROM으

로 압축하는 것이 각각 24bit, 80bit의 이득이 생긴다.

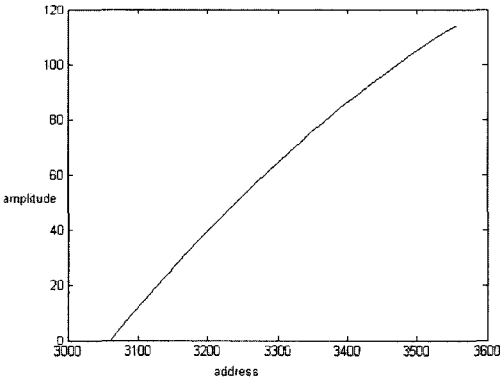
그러므로 6번째 블록까지는 sine_phase difference algorithm를, 7, 8번째 블록은 RD0 방법을 사용할 경우 104bit의 이득이 생긴다. 또한 3.1절에 소개한 HRDT의

방식과 같이 fine ROM을 FQ ROM과 FE ROM으로 나누면, [1]보다 128bit의 이득을 얻을 수 있어서 총 232bit를 줄일 수 있다.

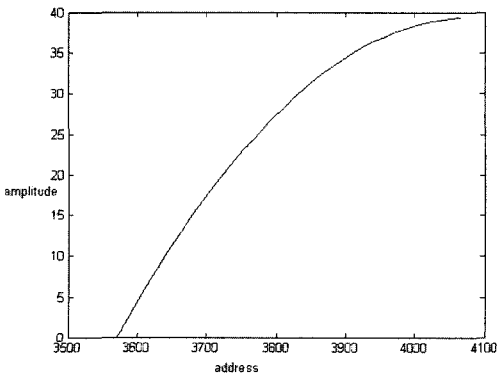
그림 6의 (a)는 [4]의 coarse ROM 데이터 크기를 나타내고, (b)와 (c)는 coarse ROM 데이터를 sine_phase difference algorithm[5]을 사용한 것보다 식 (5)를 사용하여 이득이 있는 블록(n=6,7)의 크기를 보여준다.



(a) Approximation in RD0 (Coarse ROM)



(b) 7번째 블록



(c) 8번째 블록

그림 6 Approximation in RD0

3.3 ROM Decomposition with 1st Order Approximation (RD1)

Sunderland 방법[4]에 따라 나누어진 Sunderland coarse ROM에 sine_phase difference algorithm를 대신하여 그림 7에서 보는 바와 같이 coarse ROM의 phase를 4블록으로 나눈다. address와 상수 값을 이용하여 각각의 블록에 sine_phase difference algorithm를 대신하여 식 (6), (7), (8), (9)과 같이 바꾸어서 quantization ROM과 error ROM으로 압축한다. 나누어진 구간은 $f(\phi)$ 값이 작아지도록, 즉 ROM에 저장되는 bit의 수가 작아지도록, 입력 값인 address와 address를 shift한 값과 상수를 이용한다. 식 (6)은 address와 address를 shift right한 값을, 식 (7), (8)에서는 address와 상수 값을, 식 (9)에서는 address를 두 번 shift right한 값과 상수를 사용하였다. 그림 7은 식 (6)~(9)을 사용한 coarse ROM 4개 블록의 ROM 크기를 나타낸다.

$$f(\phi) = \sin\left(\frac{\pi\phi}{2}\right) - (1.5 * \phi), \text{ for } 0 \leq \phi < 1/4 \quad (6)$$

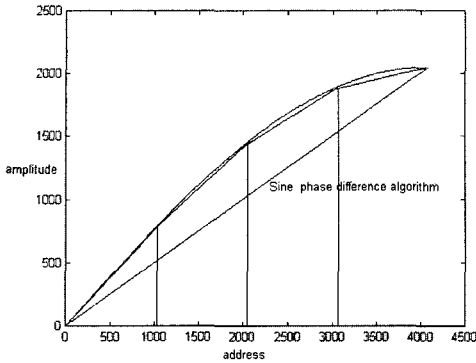
$$f(\phi) = \sin\left(\frac{\pi\phi}{2}\right) - (\phi + 278/2048), \text{ for } 1 \leq \phi < 1/2 \quad (7)$$

$$f(\phi) = \sin\left(\frac{\pi\phi}{2}\right) - (\phi + 356/2048), \text{ for } 1/2 \leq \phi < 3/4 \quad (8)$$

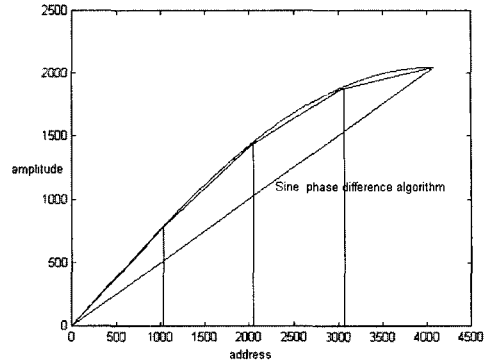
$$f(\phi) = \sin\left(\frac{\pi\phi}{2}\right) - (0.25\phi + 1510/2048), \text{ for } 3/4 \leq \phi < 1 \quad (9)$$

이러한 방법을 적용하여 quantization ROM과 error ROM으로 압축한 경우 sine_phase difference algorithm를 사용하여 quantization ROM과 error ROM으로 압축하였을 때 보다 coarse ROM의 크기가 1664bit에서 1200bit로 줄어들어 464bit의 이득이 생기고 각각의 블록은 표 2와 같은 크기를 갖는다.

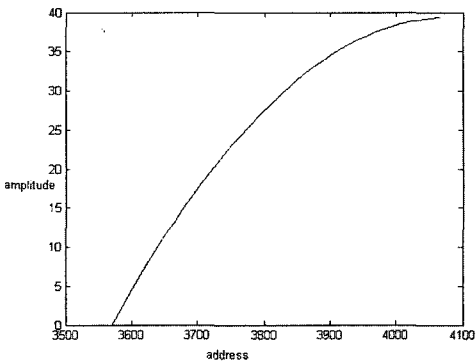
또한 Fine ROM을 FQ ROM과 FE ROM으로 나누면, [1]보다 128bit의 이득을 얻을 수 있어서 총 592bit의 ROM size를 줄일 수 있다. 표 2는 RD1의 4개 블록의 크기를 나타낸다.



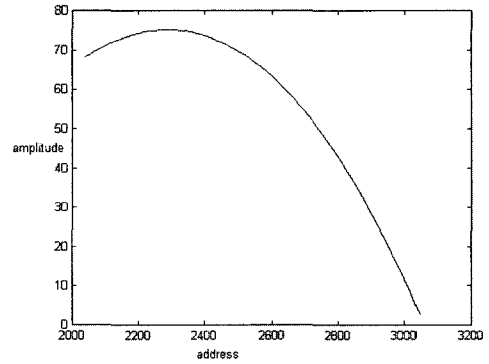
(a) 4블록으로 나누어진 Coarse ROM



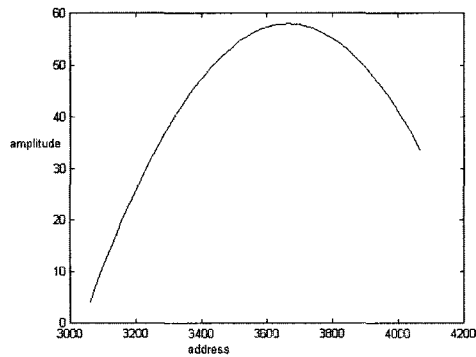
(b) 첫 번째 블록



(c) 두 번째 블록



(d) 세 번째 블록



(e) 네 번째 블록

그림 7 Approximation in RD1, $f(\theta)$

표 2 RD1의 각각의 블록의 크기

1번째	$2^4 * 4 + 2^6 * 2$
2번째	$2^4 * 4 + 2^6 * 5$
3번째	$2^4 * 4 + 2^6 * 4$
4번째	$2^4 * 3 + 2^6 * 4$
Total size	1200

4. 여러 구조의 추정 면적 비교

DDFS가 32 bit의 frequency input word와 12 bit의 output data로 구성되어 있고 $p = 14$ 이고 $m = 11$ 인 경우를 예로 들어 새롭게 제안한 방법들 (HRDT, RDO, RD1)과 Sunderland architecture[4], Nicholas archi-

표 3 각각의 ROM size비교

Compression Technique	ROM size 12 bit output
Sunderland architecture[4]	3840
Nicholas architecture[5]	3072
[1]	2496
HRDT	2320
RD0	2264
RD1	1904

texture[5], [1]의 ROM size를 표 3에 비교하였다. 모든 방법에서 식 (1)로 표현되는 minimum frequency resolution는 f_{clk} 와 2^N 의 변화가 없으므로 모두 같다.

본 논문에서 새롭게 제안한 HRDT, RD0, RD1는 [1]보다 각각 176bit, 232bit, 592bit의 ROM 크기를 줄였다. ROM의 크기를 비교하면, 본 논문에서 제안한 ROM Decomposition with 1st Order Approximation (RD1)이 가장 우수한 결과를 보여 주었으며 Sunderland[4] 방법에 비하여 50.4%, [1] 방법에 비하며 23.7% 개선된 결과를 보여준다.

ROM을 분할했을 때에는 나누어진 ROM의 값을 합하기 위한 덧셈기가 필요하다. 여러 개의 구간으로 나눈 RD0과 RD1은 적합한 구간을 선택하기 위해서 multiplexer가 필요하다.

참고로 ROM의 size와 전가산기의 size를 Synopsys (Hynix 0.35um공정)로 합성한 결과 and_gate의 크기가 1.5 일 때, 반가산기의 크기는 4.5, 전가산기의 크기는 10.5 이고, 80bit ROM(address 2^4 data"1111")의 크기는 81.1이었다.

위의 Hynix 0.35um공정에 기초한 설계 data를 바탕으로 ROM의 bit당 크기는 1로, 반가산기의 크기는 4.5로, 1bit 전가산기의 크기는 10.5로 가정하고, ROM과 덧셈기의 추정 면적을 모두 더하여 기존의 방법과 제안한 방법들을 비교하면 표 4와 같다.

표 4 ROM과 덧셈기를 포함한 추정 면적

	ROM bit수	전가산기 개수	반가산기 개수	Total 면적
KYL2001[1]	2496	12	7	2653.5
HRDT	2320	13	8	2492.5
RD0	2264	15	27	2543.0
RD1	1904	39	30	2449.5

표 4에서 보인 바와 같이 ROM과 덧셈기의 면적을 합한 경우에도 본 논문에서 제안한 ROM Decompo-

sition with 1st Order Approximation (RD1)방법이 가장 작은 면적을 사용하였다.

5. 결론

DDFS는 현재의 통신 시스템에 널리 사용된다. 본 연구에서는 기존의 DDFS의 ROM 압축방법을 설명하였고, 새로운 압축 방법들을 제시하였다. 실험결과 제안한 방법 중에 RD1이 가장 좋은 결과를 나타내었다. 최적의 DDFS방식과 디자인 기법은 요구되는 파워소비, 속도, 면적에 의해 결정된다.

참고 문헌

- [1] B. Yang, K. Sung, Y. Kim, L. Kim, S. Han, and H. Yoo, "A Direct Digital Frequency Synthesizer Using A New ROM Compression Method," European Solid-State Circuit Conference 2001, pp. 385-390, Sep. 2001.
- [2] J.Tierney, C.M. Rader, and B. Gold, "A digital frequency synthesizer," IEEE Trans. on Audio Electroacoustic, vol. AU-19, pp. 48-56, mar. 1971.
- [3] A. Bellaouar et al., "Low-Power Direct Digital Frequency Synthesis for Wireless Communications," IEEE J. Solid-State Circuits, vol. 35, no. 3, pp. 385-390, Mar. 2000.
- [4] D. A Sunderland, et al., "CMOS/SOS Frequency Synthesizer LSI Circuit for Spread Spectrum Communications," IEEE J. Solid-State Circuits, vol. 19, no. 4, pp. 497-505, Aug. 1984.
- [5] H. T. Nicholas, et al., "The Optimization of Direct Digital Frequency Synthesizer Performance in the Presencs of Finite Word Length Effects," in Proc. 42nd Annu. Freq. Contr. Symp., pp.357-363, 1988.
- [6] L.A. Weaver, 'High Resolution Phase to Sine Amplitude Conversion',U.S. Patent 4 905 177, Feb. 1990.
- [7] M. PARK, "CORDIC-Based Direct Digital Frequency Synthesizer : Comparison with a ROM - Based Architecture in FPGA Implementation," IEICE Trans. FUNDAMENTAL, Vol. E83-A, No. 6, pp. 1282-1285, June 2000.



이진철

2002년 한양대학교 제어계측공학과 학사
 2004년 한양대학교 전자전기제어계측공
 학과 석사. 현재 삼성전자 통신연구소 연
 구원. 관심분야는 저전력 설계, 전력 추정



신현철

1978년 서울대학교 전자공학과 학사.
 1980년 한국과학기술원 전기 및 전자공
 학 석사. 1983년~1987년 U.C. Berkeley
 Ph.D. 1983년~1987년 Fulbright scho-
 larship. 1987년~1989년 MTS, AT&T
 Bell Lab's, Murray Hill N.J., USA.
 1989년~현재 한양대학교 교수. 1997년~현재 IDEC 한양대
 학교 지역센터 센터장. 관심분야는 CAD&VLSI, 통신용 반
 도체 설계, 저전력설계