논문 2007-44SD-9-7

그룹 곱셈 계수를 위한 Modified CSD 그룹 곱셈기 디자인

(Modified CSD Group Multiplier Design for Predetermined Coefficient Groups)

김 용 은*, 허 일 남*, 정 진 균**

(Yong-Eun Kim, Yi-Nan Xu, and Jin-Gyun Chung)

요 약

Fast Fourier Transform (FFT)과 같은 디지털 신호처리 응용에서는 계수가 미리 정해진 특정 그룹의 곱셈기를 사용한다. 본 논문에서는 수정된 CSD 알고리즘 및 부분곱 공유 알고리즘을 기반으로 계수가 미리 정해진 특정 그룹의 곱셈 계수를 위한 효율적인 곱셈기 설계 방법을 제안한다. 제안한 알고리즘을 direct digital frequency synthesizer (DDFS)에 사용되는 sine/cosine 생성회로 및 128 point radix-24 FFT에 사용되는 곱셈기에 적용하였을 경우 기존 곱셈에 비하여 면적, 소비전력, 속도에서 최대 34%의 이득이 있음을 CAD 시뮬레이션을 통해 보인다.

Abstract

Some digital signal processing applications, such as FFT, request multiplications with a group (or, groups) of a few predetermined coefficients. In this paper, based on the modified CSD algorithm, an efficient multiplier design method for predetermined coefficient groups is proposed. In the multiplier design for sine-cosine generator used in direct digital frequency synthesizer(DDFS), and in the multiplier design used in 128 point radix-2^4 FFT, it is shown that the area, power and delay time can be reduced up to 34%.

Keywords : Group Multiplier, Modified CSD, FFT, DDFS

Ⅰ. 서 론

곱셈기는 디지털 신호처리(DSP)의 기본 연산 회로로서 많은 분야에 이용된다. 등화기와 같은 응용에서 곱셈기의 계수는 선로의 환경에 따라 계속 업데이트 되며 사용될 계수를 미리 예측하기 어렵다. 그러나 FFT와 같은 응용에서는 사용되는 곱셈 계수들이 미리 정해지게 되며 몇 개의 곱셈계수들이 하나의 곱셈기를 공유하여 연산을 수행할 경우 Modified Booth 곱셈기가 주로 사용된다[1].

곱셈기의 계수가 고정되어 있을 경우에는 면적과 파워를 줄이기 위해 계수를 canonic signed digit(CSD) 표현으로 인코딩 하는 방식이 널리 사용된다[2~3]. CSD 코딩방식을 사용할 경우 곱셈계수의 nonzero 디지트 개수를 최소화 할 수 있으므로 곱셈기를 shift-add 방식으로 구현할 경우 효율적이다.

Ⅱ. 기존의 Radix-2^4 FFT의 곱셈블록

128 point radix-2^4 FFT의 두 번째 stage에서는 sin($\pi/8$), cos($\pi/8$), cos($\pi/4$)의 세 계수가 곱해진다[4]. 그림 1은 FFT에서 세 계수가 하나의 그룹으로서 곱셈기를 공유하여 곱해지는 과정을 나타낸다.

그림 1에서 곱셈기는 일반적으로 Modified Booth 곱셈기를 사용한다. 그림 1의 계수를 14 비트의 2의 보수로 표현하면 표 1과 같다. 계수가 14 비트 이므로 7 줄

* 학생회원, ** 정회원, 전북대학교 전자정보공학부
(Div. of Electronic & Information Engineering Chonbuk University)

※ 이 연구에 참여한 연구자는 2단계 BK21사업의 지원비를 받았음, This work was supported by the second stage of Brain Korea 21 Project.
접수일자: 2007년5월29일, 수정완료일: 2007년8월16일

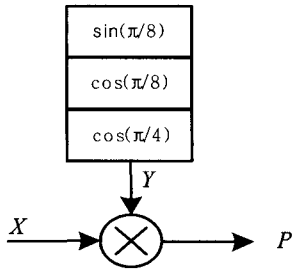


그림 1. FFT에서 계수와 곱해지는 곱셈블록
Fig. 1. Multiplication block used in FFT.

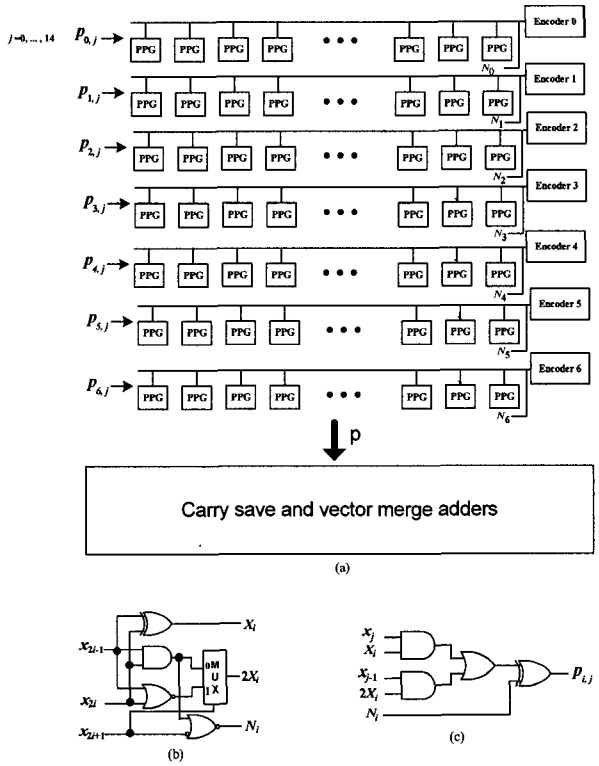


그림 2. Modified Booth 곱셈기: (a)전체 블록도, (b)인코더, (c)부분곱 발생회로
Fig. 2. Modified Booth Multiplier. (a)Top block diagram, (b)Incoder, (c)Partial product generator

표 1. 그림 1에 사용되는 곱셈 계수의 2의 보수 표현 값
Table 1. 2's complement representation of the multiplication coefficient in Fig. 1. (word-length = 14)

	2의 보수 표현													
비트 No.	13	12	11	10	9	8	7	6	5	4	3	2	1	0
cos(π/4)	0	1	0	1	1	0	1	0	1	0	0	0	0	0
sin(π/8)	0	0	1	1	0	0	0	0	1	1	1	1	1	1
cos(π/8)	0	1	1	1	0	1	1	0	0	1	0	0	0	0

의 부분곱이 생성되며 전체적인 곱셈기의 구조는 그림 2와 같다. 그림 2-(a)는 전체 블록도 이며 인코더

표 2. 표 1의 계수의 CSD 표현 값
Tabel 2. CSD representation of the coefficients in Table 1.

	CSD 표현													
비트 No.	13	12	11	10	9	8	7	6	5	4	3	2	1	0
cos(π/4)	1	0	0	0	-1	0	-1	0	0	1	0	0	0	0
sin(π/8)	0	1	0	-1	0	0	0	1	0	0	0	0	0	-1
cos(π/8)	1	0	-1	0	-1	0	1	0	1	0	0	0	0	0

(encoder) 에서 생성된 값이 부분곱 생성기 (PPG)를 제어하여 최종 부분곱이 생성되고 부분곱들은 덧셈블럭에서 더해지게 된다. 그림 2-(b)는 인코더이며 그림 2-(c)는 부분곱 생성기이다.

III. 효율적인 그룹 곱셈기 설계방법

1. 부분곱 줄 수 감소 방법

표 1의 계수를 CSD 알고리즘을 이용하여 나타내면 표 2와 같다. 표 1에는 nonzero 비트가 19개이지만 CSD 알고리즘을 적용함으로서 13개로 감소되었음을 알 수 있다. CSD 계수를 shift와 add 방식으로 구현할 경우 부분곱의 줄 수는 각 계수의 nonzero 디지트 수와 같다. 본 논문에서는 부분곱의 줄 수를 더욱 감소시키기 위하여 곱셈 계수의 그룹화 알고리즘을 제안한다.

곱셈 계수를 그룹화 하는 원칙은 한 그룹내에서 덧셈기 사용을 피하기 위하여 그룹화 된 각 행에 nonzero 디지트 개수가 1이하가 되도록 하는 것이다. 예를 들어, 표 3에서와 같이 열 13과 12를 하나의 그룹 P_4로 묶는다. P_4에는 각 행에 nonzero 디지트가 1개씩만 포함되어 있다. 만약 열 11을 P_4에 포함 시킬 경우 세 번째 행에 nonzero 디지트가 두 개가 되므로 열 11은 다음 그룹에 포함 되어야 한다. 반면, P_0의 경우 열 5에서 0까지를 포함해도 각 행에 nonzero 디지트가 하나씩 밖에 포함되지 않으므로 하나의 그룹으로 묶을 수 있다.

표 3은 이와 같은 원칙에 따라 표 2를 5개의 그룹으로 묶을 수 있음을 보여주며 곱셈 수행 시 5줄의 부분곱을 가지게 된다. Modified Booth 알고리즘을 사용할 경우 14 bit의 곱셈 계수는 7줄의 부분곱을 가지게 되므로 제안한 부분곱 그룹화를 적용함으로서 2줄의 부분곱을 감소시킬 수 있다.

CSD 계수는 연속해서 nonzero 디지트가 올 수 없으므로 제안하는 그룹화 알고리즘은 한 그룹이 최소 2열 이상을 포함하게 된다. 또한 각 그룹에서는 한 줄의 부분곱만을 생성하므로 부분곱의 줄 수는 곱셈 계수의 워

표 3. 표 2의 CSD 계수의 그룹화
Table 3. CSD coefficient grouping of Table 2.

	CSD 표현													
비트 No.	13	12	11	10	9	8	7	6	5	4	3	2	1	0
sin(π/8)	1	0	0	0	-1	0	-1	0	0	1	0	0	0	0
cos(π/4)	0	1	0	-1	0	0	0	1	0	0	0	0	0	-1
cos(π/8)	1	0	-1	0	-1	0	1	0	1	0	0	0	0	0
부분곱	P_4		P_3		P_2		P_1		P_0					

드 길이가 W 일 경우 $W/2$ 보다 작거나 같다. 따라서 생성되는 부분곱의 줄 수는 Modified Booth 부분곱의 줄 수 보다 항상 작거나 같다.

제안하는 곱셈 계수의 그룹화 알고리즘을 요약하면 다음과 같다.

곱셈계수 그룹화 알고리즘

1. 주어진 곱셈 계수의 개수가 N_c, 워드길이가 N_W 일 때 각 곱셈 계수를 CSD로 변환한 후 $N_c \times N_W$ 의 행렬로 표현한다.
2. 맨 처음 열부터 시작하여 맨 끝열에 이르기 까지 각 행에 nonzero 디지트 수가 1이하가 되면서 가능한 많은 수의 열을 포함 하도록 각 그룹을 차례대로 정의한다.

부분곱을 그룹화 할 때 각 행에 nonzero 디지트 수가 1이하이므로 입력 x를 nonozero 디지트의 위치에 따라 shift 하여 계수에 따른 부분곱을 생성 시킬 수 있다. 또한, 표 3에서 처럼 -1이 곱해져서 입력 x가 2의 보수가 되어야 할 때와 부분곱이 0이 되어야 하는 경우가 있으므로 부분곱 생성회로를 구현할 때 부분곱을 shift 하는 부분은 MUX를, 부분곱이 음수를 가져야 할 때는 XOR 게이트를, 부분곱이 0이 되어야 할 때에는 AND 게이트를 사용하여 처리한다. 그림 2는 표 3의 부분곱 P_3와 P_0를 생성하기 위한 부분곱 생성회로이다. P_3의 경우 두 열을 포함하므로 MUX 입력은 x_j와 x_{j-1}이 된다. 반면 P_0는 6열을 포함하고 nonzero 디지트는 P_0의 첫 번째, 두 번째와 여섯 번째 열에만 존재하므로 mux 입력은 x_j, x_{j-1}, x_{j-5}가 된다.

2. 부분곱 생성회로의 공유

표 3에서 두 그룹의 nonzero 디지트의 절대값 패턴이 같다면 부분곱 생성회로를 공유할 수 있다. 즉, 입력의 shift되는 패턴이 같다면 그림 2의 MUX회로가 공유 될 수 있다. 예를 들어 표 3에서 P_4와 P_1은 계수에 따라 차례대로 $(2x, x, 2x)$의 입력을 제공해야 하기 때문에 (-1은 1로 간주하고 추후 부분곱을 반전 시킨다) shift 역할을 하는 MUX 회로를 공유할 수 있다. MUX회로를 공유하면 x의 bit 수만큼의 MUX 회로가 줄어든다. 부분곱을 더욱 많이 공유하기 위하여 다음과 같은 방법을 제안한다.

부분곱에서 0을 출력할 때 그림 2-(a)와 같이 AND gate를 이용하면 어떤 값도 최종적으로 0으로 만들 수 있기 때문에 표 3의 0인 원소들은 don't care로 간주하여 처리하고 추후에 0으로 만든다. 즉, 표 3의 0을 부분곱 회로가 공유될 수 있는 값으로 바꾼다. 표 4는 표 3에서 부분곱 생성회로가 더 공유될 수 있도록 바꾸어준 결과이다. 바뀐 부분은 괄호로 나타내었다.

입력 x를 0으로 만들어 주는 제어 신호를 zero, complement 시켜주는 제어 신호를 neg, MUX 제어신호를 sel 이라고 하면 shift 제어신호 P_{4-sel}을 이용하여 부분곱 P_4를 먼저 생성한다. 표 4에서 P_3의 경우 P_4의 결과를 complement 시키고 P_3의 첫 행은 P_{3-zero}신호를 이용하여 0으로 만든다. 부분곱 P_2, P_1 역시 P_4의 부분곱 결과에 제어신호 neg, zero를 이용하여 반전 시키거나 0으로 만들어 생성시킬 수 있다.

전체적인 부분곱 생성회로는 그림 3과 같다. 그림 3을 이용하여 부분곱을 생성 할 때 그림 2의 부분곱 회로를 이용하여 부분곱을 생성할 때 보다 42(14×3)개의 MUX가 감소된다. 부분곱 생성 회로를 공유하는 알고리즘을 요약하면 다음과 같다.

부분곱 생성회로 공유 알고리즘

1. 계수의 행렬에서 총 그룹수가 Ng일 때 각 그룹을 $P_i(i = Ng-1, Ng-2, \ldots, 0)$ 라고 한다.
2. 각 그룹 P_i에 대해 동일한 nonzero 디지트 절대값 패턴을 갖는 그룹들의 수가 최대가 되도록 나머지 그룹의 0을 nonzero 디지트로 바꾸어 주고 동일한 nonzero 디지트 절대값 패턴을 갖는 그룹들의 수 S_i를 기록한다.
3. S_i가 최대일 때의 동일한 nonzero 디지트 절대값 패턴을 갖는 그룹들을 기록하고 계수 행렬에서 제외시킨다.
4. 계수 행렬의 나머지 그룹들에 대해 더 이상 그룹이 남지 않을 때까지 2~3을 반복한다.

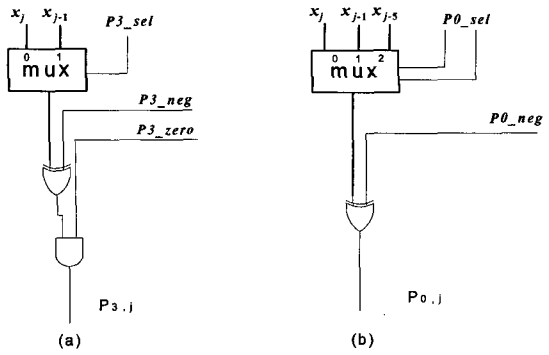


그림 2. P_3와 P_0의 부분곱 생성회로
Fig. 2. PP generation circuit for P_3 and P_0.

표 4. 공유 패턴수를 최대로하기 위해 '0'을 수정한 결과
Table 4. Change of 0's to reduce PP generation circuits.

계수	CSD 표현											
비트 No.	13	12	11	10	9	8	7	6	5	4	3	2	1	0
sin(π/8)	1	0	(-1)	0	-1	0	-1	0	0	1	0	0	0	0
cos(π/4)	0	1	0	-1	0	(-1)	0	1	0	0	0	0	0	-1
cos(π/8)	1	0	-1	0	-1	0	1	0	1	0	0	0	0	0
부분곱	P_4		P_3		P_2		P_1		P_0					

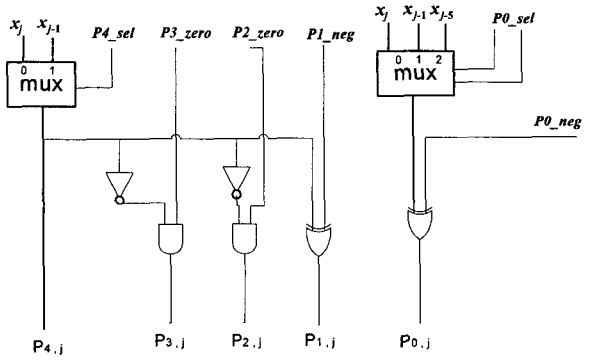


그림 3. 부분곱 생성 회로
Fig. 3. PP generation circuit for partial product.

Example 1: 표 3에서 Ng = 5이고 P_4에 대해 P_3, P_2, P_1을 P_4와 동일하게 만들 수 있으므로 S_4=4이다. 반면 $S_3 = 1$, $S_2 = 1$, $S_1 = 4$, S_0=1이다. 따라서 P_4, P_3, P_2, P_1은 부분곱 생성시 MUX 회로를 공유할 수 있다. 다음 P_4, P_3, P_2, P_1을 제외하면 P_0만 남으며 P_0의 S_0는 1이 되어 최대값이 되므로 P_0를 제외한다. □

그림 4는 제안한 곱셈기의 전체 블록이다. 그림 4-(a)에서 부분곱의 제어신호를 저장하고 있으며 이 제어신호가 ROM에서 출력되면 새로운 부분곱 생성회로

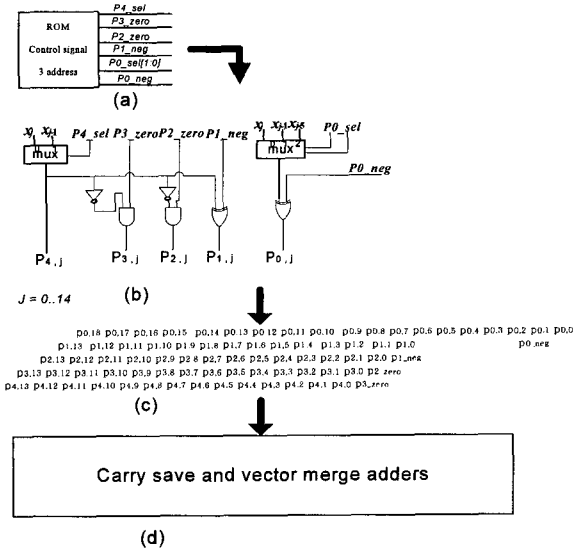


그림 4. 제안한 modified CSD 곱셈 구조
Fig. 4. Proposed modified CSD multiplier structure.

4-(b)로 입력되어 부분곱들을 생성한다. 부분곱들이 생성되면 이 부분곱들은 4-(d) 블록에서 더해진다. 기존의 곱셈기에 비하여 부분곱 생성회로가 간단하여 짐을 확인할 수 있고, 부분곱 줄 수도 감소됨을 알 수 있다.

Ⅳ. 시뮬레이션 및 비교

128 point radix-2^4 FFT에서 계수가 sin(π/8), cos(π/8), cos(π/4)인 곱셈 블록을 일반적인 Modified Booth 곱셈 설계 방법과 제안한 방법을 이용하여 설계하고 삼성 0.35um 라이브러리를 이용하여 Synopsys로 합성한 결과를 표 5에 보였다. Modified Booth / [5]의 방법에 비해 면적에서 각각 42.9% / 25.7% 이득을 얻으며 전력

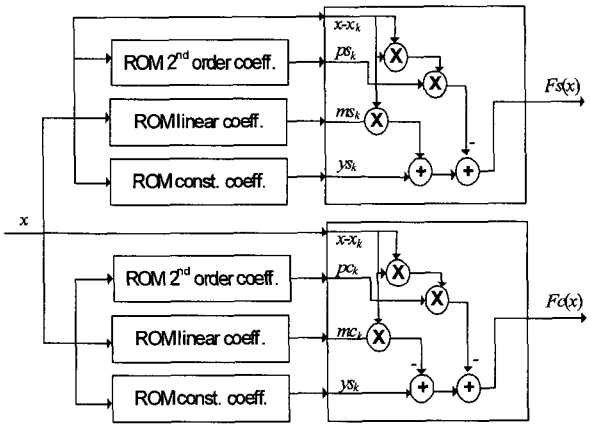


그림 5. Piecewise-quadratic sine and cosine 생성 블록
Fig. 5. Piecewise-quadratic sine and cosine generation block.

표 5. 입력이 14 bit 이고 계수가 sin(π/4), cos(π/4), cos(π/8) 일 때 Modified Booth, [5]와 제안한 방법으로 설계한 결과

Table 5. Design results of constant multipliers with 14bit input and coefficient {sin(π/4), cos(π/4), cos(π/8)} using conventional, [5] and proposed methods.

구 분	저장 계수:14 bit, 입력:14 bit		
	Modified Booth	[5] 방법	제안방법
면적(cell)	1293.1	993.3	738.3
파워(mW)	140.6	76.7	58.4
속도(ns)	10.2	10.5	8.2

표 6. Pck 의 계수

Table 6. Coefficient Pck.

	coefficient *pck*											
비트 No.	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	1	0	0	1	0	0	0	0	1
1	1	0	1	1	1	1	0	1	1	1	1	1

표 7. CSD로 변환한 Pck 의 계수

Table 7. CSD coded Pck.

	CSD 표현											
비트 No.	11	10	9	8	7	6	5	4	3	2	1	0
0	0	-1	0	-1	0	0	1	0	0	0	0	1
1	0	-1	0	0	0	0	-1	0	0	0	0	-1
PP	P3			P2			P1					P0

표 8. Pck의 계수와 곱해지는 곱셈 블럭을 기존방법과 제안한 방법으로 설계한 결과

Table 8. Design results of multiplication block with coefficient Pck using proposed methods.

구 분	저장 계수 : 12 bit		
	기존방법	[5] 방법	제안방법
면적(cell)	1000.2	507.4	468.7
파워(mW)	55.1	46.1	30.4
속도(ns)	9.9	7.1	7

소모면에서는 각각 58.5% / 23.9%, 속도면에서 각각 19.7% / 21.9% 이득이 있음을 보인다.

그림 5는 DDFS의 piecewise-quadratic sine and cosine 생성 블록이다[6]. 그림 4의 pck와 곱해지는 곱셈 블록 계수는 표 6과 같다. 표 7의 값은 CSD 알고리즘을 이용하여 표 6의 값을 변환한 것이고, 이를 제안한 알고리즘으로 구현한 결과는 표 8과 같다. Modified Booth / [5]의 방법에 비해 면적에서 각각 53.1% / 7.6% 이득을 얻으며 전력 소모면에서는 각각 44.8% / 34.1%, 속도면에서 각각 29.3% / 1.4% 이득이 있음을 보인다.

V. 결 론

본 논문에서는 미리 정해진 특정 그룹의 계수가 있을 때 제안한 알고리즘을 이용하여 부분곱 줄 수를 줄여 부분곱 전개시 full adder의 개수를 줄이고 부분곱 회로를 공유하여 MUX 개수를 줄이는 방법에 대해서 설명하였다. 제안한 알고리즘을 FFT와 DDFS의 sine/cosine 생성기의 곱셈 블록에 적용하여 파워, 면적, 속도에서 이득이 있음을 보였다.

참 고 문 헌

[1] O. L. MacSorley, "High speed arithmetic in binary computers", *Proc. IRE*, vol. 49, pp. 67-91, 1961.

[2] S. W. Reitwiesner, "Binary arithmetic," *Advances in Computers*, pp. 231 - 308, 1966.

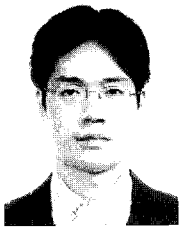
[3] 이은실, 김정범 "전류모드 CMOS 다치 논리회로를 이용한 32x32-bit Modified Booth 곱셈기 설계" 전자공학회논문지 제40권 SD편 제 12호, pp. 72-79 2003.12.

[4] J. Y. Oh and M. S. Lim, "New radix-2 to the 4th power pipeline FFT processor," *IEICE Trans.* vol. E88-C, no. 8, pp. 1740-1746, Aug. 2005.

[5] Y. E Kim, K. J. Cho, J. G. Chung "Low power small area modified Booth multiplier design for predetermined coefficients" *IEICE Trans.* vol. E90-A, pp. 694-697 Mar. 2007.

[6] De Caro. D and Strollo, A. G. M. "High-performance direct digital frequency sysnthesizers using piecewise-polynomial approximation", *IEEE Trans.* vol. 52, no. 2, pp. 324-337, Feb. 2005.

저 자 소 개



김 용 은(학생회원)
2005년 전북대학교 전자공학과 학사 졸업
2007년 전북대학교 정보통신 공학과 석사 졸업
2007년~현재 전북대학교 전자 정보공학부 박사
<주관심분야 : 통신, 신호처리, 반도체>

허 일 남(학생회원)
1999년 중국연변대학 전자정보 공학과 학사 졸업
2003년 전북대학교 물리기술학과 석사
2005년~현재 전북대학교 전자정보공학과 박사
<주관심분야 : 통신, 신호처리, 반도체>

정 진 균(정회원)
1985년 전북대학교 전자공학 학사 졸업
1989년 미국 미네소타 주립대학 전기공학 석사 졸업
1991년 미국 미네소타 주립대학 전기공학 박사 졸업
<주관심분야 : 통신, 컴퓨터, 신호처리, 반도체>