

Group CSD(GCSD) 곱셈기를 이용한 Time-Multiplexed FIR 필터 설계

논문
59-2-37

Time-Multiplexed FIR Filter Design Using Group CSD(GCSD) Multipliers

전 창 하[†] · 서 동 현* · 정 진 균** · 김 용 은*** · 이 철 동[§]

(Chang-Ha Jeon · Dong-Hyun Seo · Jin-Gyun Chung · Yong-Eun Kim · Chul-Dong Lee)

Abstract - Multiplication is a fundamental arithmetic operation in many digital signal processing (DSP) and communication algorithms. The group CSD (GCSD) multiplier was recently proposed based on the variation of canonical signed digit (CSD) encoding and partial product sharing. This multiplier provides an efficient design when the multiplications are performed only with a few predetermined coefficients (e.g., FFT). In this paper, it is shown that, by exploiting the characteristics of the filter coefficients, GCSD multipliers can be used for the efficient implementation of time-multiplexed FIR filters.

Key Words : GCSD multiplier, Modified booth multiplier, Time-multiplexed FIR filter

1. 서 론

곱셈기는 디지털 신호처리 및 통신시스템의 주요 블록중 하나로서 긴 계산시간 및 큰 면적을 차지하는 연산자이다. 따라서 효율적인 곱셈기의 구현을 위해서 다양한 곱셈기 설계 방법들이 제안되었다.

곱셈계수가 고정되어 있지 않고 프로그래머블한 경우 modified Booth 곱셈기가 주로 사용된다[1]. 그룹 CSD 곱셈기는 프로그래머블 곱셈기에 사용되는 곱셈계수의 종류가 미리 정해져 있고, 곱셈계수의 수가 많지 않은 통신 및 DSP 응용의 VLSI 시스템 구현에 효율적으로 사용하기 위해 최근 제안된 곱셈기이다[2-3]. 프로그래머블 곱셈기에 사용되는 곱셈계수의 종류가 미리 정해져 있고, 곱셈계수의 수가 많지 않은 경우 modified Booth 곱셈기를 수정함으로써 효율적으로 구현할 수 있는 방법도 제안되었으나[4] 부분곱(PP)의 줄 수를 감소시키는 측면에서 CSD 기반의 방법이 더 우수함이 알려져 있다[3].

GCSD 곱셈기는 특정 그룹의 곱셈계수에 대해 CSD 코딩[5]과 부분곱 공유방법을 결합함으로써 생성되는 부분곱의 개수를 감소시킨다. 곱셈계수의 워드길이 W 일 경우 항상 $W/2$ 개의 부분곱을 생성하는 modified Booth 곱셈기와 달리 GCSD 곱셈기는 곱셈계수들의 특성에 따라 $W/2$ 개, 또는 그

이하의 부분곱을 생성함으로써 효율적인 곱셈기의 구현이 가능하다.

Modified Booth 곱셈기는 2의 홀수 차수의 weight를 갖는 digit만 nonzero 값을 가질 수 있다는 제약 조건하에 주어진 곱셈 계수를 코딩함으로써 부분곱의 계수를 $W/2$ 로 감소시킨다. 반면, GCSD 곱셈기는 다음과 같은 세 가지 제약 조건하에 주어진 곱셈계수들을 테이블로 정렬한 뒤 코딩함으로써 생성되는 부분곱의 계수를 $W/2$ 이하로 감소시킬 수 있다: 1) 계수의 이웃한 bit들을 grouping하고 (단, $2 \leq$ 그룹사이즈), 2) 한 group에서 하나의 PP만 생성하고, 3) 전체적으로 group 수를 최소화한다. GCSD 곱셈기는 그룹 사이즈가 2보다 크거나 같으므로 worst case(그룹사이즈=2)인 경우 modified Booth 곱셈기와 PP 수가 같고, 그 외의 경우는 PP수가 더 적다.

본 논문에서는 2절에서 time-multiplexed 필터구조에 대해 설명하고, 3절에서는 GCSD 곱셈기의 time-multiplexed FIR 필터 응용 방법을 제안한다. 끝으로 4절에서 결론을 맺는다.

2. Time-multiplexed 필터 구조

FIR 필터는 그림 1과 같이 직접형과 전치형의 기본적인 구조가 있다. 직접형은 전력 소모가 적고 구현이 쉽다는 장점이 있지만 속도가 느린 단점이 있다. 전치형 구조는 그림 1(b)와 같이 각 곱셈기와 덧셈기 사이에 레지스터를 삽입함으로써 최대지연시간을 줄일 수 있다. 그러나 구현면에서 직접형보다 복잡하고 필터 탭수가 증가할수록 데이터 입력 버스의 캐패시턴스가 증가되어 필터 성능을 제약하는 단점을 가진다.

그림 1과 같은 시스템에서 FIR필터의 차수가 증가함에

† 교신저자, 정회원 : 전북대 전기전자컴퓨터공학부 석사과정
E-mail : jsure@chonbuk.ac.kr

* 비 회원 : 전북대 전기전자컴퓨터공학부 석사과정

** 비 회원 : 전북대 전기전자컴퓨터공학부 교수

*** 비 회원 : 자동차부품연구소 연구원

§ 비 회원 : 전북임베디드시스템연구소 연구원

접수일자 : 2009년 11월 19일

최종완료 : 2010년 12월 23일

따라 곱셈기의 개수도 증가하게 된다. 곱셈기는 디지털 시스템에서 많은 면적을 차지하기 때문에 time-multiplexed FIR 필터에서는 그림 2와 같이 곱셈계수를 그룹핑 시켜서 사용되는 곱셈기 개수를 감소시킨다.

대부분의 FIR 필터의 계수는 첫 번째 계수 W_{first} 와 마지막 계수 W_{last} 의 부근에서 절대값의 크기가 작고 가운데 계수 $W_{(first-last)/2}$ 부근에서는 절대값의 크기가 큰 특징을 가진다. 그림 3은 통과 대역이 $0 \sim 0.3\pi$ 이고 천이대역이 0.05π 인 92차 저대역통과 필터의 주파수 응답과 임펄스 응답을 나타낸다. 그림 3(b)에서 중앙 부분의 절대값이 가장 크고 중앙에서 멀어질수록 계수 값의 크기가 작아짐을 알 수 있다.

따라서, 그림 2와 같은 time-multiplexed FIR 필터 구조에서 각 그룹의 세 계수 $\{W_i, W_{i+1}, W_{i+2}\}$ 는 서로 유사한 크기를 갖는다. 이러한 특징을 이용하여 3절에서 보이는 바와 같이 time-multiplexed FIR 필터를 더욱 효율적으로 구현할 수 있다.

그림 3의 FIR 필터 구조에서 계수를 공유하는 프로그래머블 곱셈기는 일반적으로 modified Booth 곱셈기를 사용하여 구현할 수 있으며 modified Booth 곱셈기는 계수의 워드 길이가 W 비트일 때 $W/2$ 개의 부분곱을 생성한다.

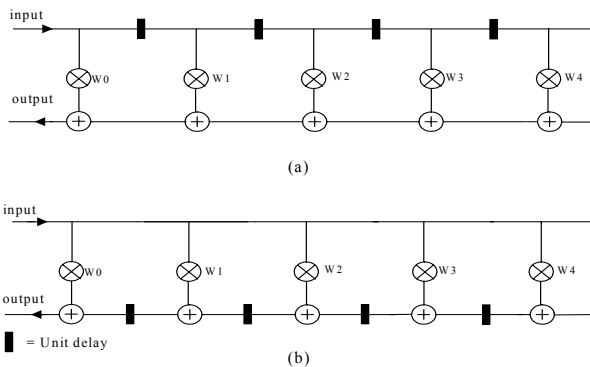


그림 1 직접형과 전치형의 FIR 필터구조: (a) 직접형 구조, (b) 전치형 구조

Fig. 1 FIR filter structures: (a) direct form structure, and (b) transposed form structure

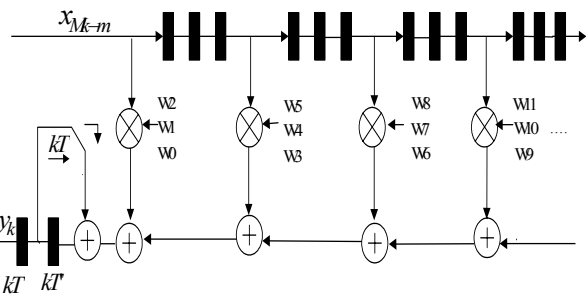
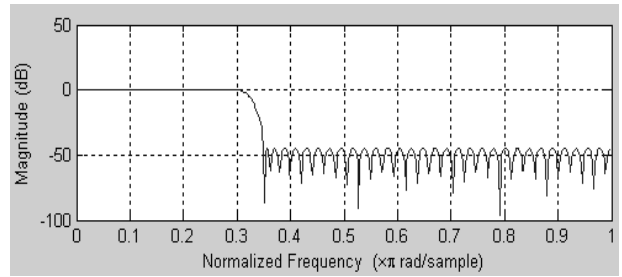
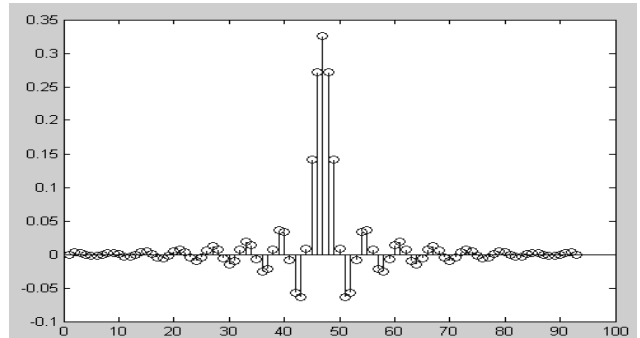


그림 2 Time-multiplexed FIR 필터 구조

Fig. 2 Time-multiplexed FIR filter structure



(a)



(b)

그림 3 통과대역이 $0 \sim 0.3\pi$ 이고 천이대역이 0.05π 인 92차 저대역통과 필터: (a) 주파수 응답, (b) 임펄스 응답

Fig. 3 92nd order low pass filter with passband width of $0 \sim 0.3\pi$ and transitionband of 0.05π : (a) frequency response, and (b) impulse response

3. GCSD 곱셈기의 time-multiplexed FIR 필터 응용

3.1 GCSD 곱셈기

Modified Booth 곱셈기는 그룹핑 되는 계수들의 특성을 고려하지 않는 반면, GCSD 곱셈기를 사용하면 그룹핑 되는 계수들의 특성을 고려한 효율적인 곱셈기를 구현 할 수 있다. GCSD 곱셈기는 modified Booth 곱셈 방식보다 부분곱의 줄 수를 감소시키기 위해 먼저 2의 보수 형태의 모든 곱셈계수를 테이블 형태로 정리한 후 CSD 계수로 변환한다. 변환된 테이블의 처음 컬럼부터 시작하여 각 행에 포함된 nonzero 디지털의 개수가 1개 이하가 되면서 가능한 한 많은 컬럼이 포함되도록 그룹을 생성한다. 이와 같은 조건으로 그룹을 생성하면 각 그룹에서는 한줄의 부분곱이 생성되고, 생성된 부분곱들을 더함으로써 곱셈 결과를 얻는다.

예를 들어 그림 2에서 계수 그룹 $\{W_0, W_1, W_2\}$ 이 $\{-0.005, 0.01, -0.02\}$ 일 때 각 계수를 14비트의 2의 보수로 표현하면 표 1과 같다. 표 1의 계수를 CSD 방식을 이용하여 표현하면 표 2와 같다. 표 2의 CSD 계수에 계수 그룹핑 알고리즘을 적용하면 표 3과 같이 4개의 그룹으로 그룹핑 할 수 있다. 따라서 각 그룹에서 한줄의 부분곱이 생성되므로 7줄의 부분곱이 생성되는 modified Booth 곱셈기에 비해 42.8%의 부분곱 줄 수를 감소시킬 수 있다.

표 1 계수들의 2의 보수 표현

Table 1 2's complement representation of the coefficients

비트 No.	2의 보수 표현													
	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-0.005	1	1	1	1	1	1	1	0	1	0	1	1	1	1
0.01	0	0	0	0	0	0	1	0	1	0	0	0	1	1
-0.02	1	1	1	1	1	0	1	0	1	1	1	0	0	1

표 2 표 1의 계수들의 CSD 표현 값

Table 2 CSD representation of the coefficients in Table 1

비트 No.	CSD 표현													
	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-0.005	0	0	0	0	0	0	0	0	0	-1	0	0	0	-1
0.01	0	0	0	0	0	0	1	0	1	0	0	1	0	-1
-0.02	0	0	0	0	0	0	0	-1	0	0	0	-1	0	1

표 3 표 2의 CSD 계수의 그룹핑

Table 3 Grouping of the CSD coefficients in Table 2

비트 No.	CSD 표현													
	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-0.005	0	0	0	0	0	0	0	0	0	-1	0	0	0	-1
0.01	0	0	0	0	0	0	1	0	1	0	0	1	0	-1
-0.02	0	0	0	0	0	0	0	-1	0	0	0	-1	0	1
그룹	G3				G2				G1		G0			

부분 곱의 줄 수가 감소되면 하드웨어의 감소뿐 아니라 곱셈기의 계산시간이 감소된다. 예를 들어, 그림 4는 부분곱 줄 수가 4인 곱셈기와 3인 곱셈기의 구조를 비교한 그림이다. 그림에서 알 수 있듯이 줄 수가 4인 곱셈기 보다 3인 곱셈기는 FA(Full Adder)가 1줄 차이가 나는 것을 확인 할 수 있다. 즉, 부분곱의 줄 수를 n 줄 감소시킴으로써 n 개의 FA에 해당하는 지연시간을 감소 시킬 수 있다.

필터 계수의 절대 값의 크기가 작으면 CSD 표현방식의 특징으로 인해 계수 그룹의 MSB 부분이 0이 된다. 따라서 필터 계수의 절대 값의 크기가 작은 그룹계수가 있을 때 GCSD 곱셈기를 이용하면 효율적인 설계가 가능하다.

3.2 Time-multiplexed FIR 필터 응용 예

그림 5는 데시메이션 필터의 예이며, 콤필터와 FIR 1차 및 FIR 2차 필터로 구성된다[6]. 표 4는 stage 3의 FIR 필터 계수이며, M8을 제외하고는 4개의 필터 계수가 하나의 곱셈기를 공유하는 구조이다.

표 5는 표 4를 GCSD 방법으로 그룹화하여 표현한 필터 계수이다. 표 5에서 FIR필터 계수의 특성으로 인해 계수들

의 MSB에 0이 밀집되어 있음을 알 수 있다. 따라서 GCSD 곱셈기를 이용하면 부분곱 줄 수가 감소하게 된다. 표 6은 modified Booth 곱셈 방법과 GCSD 방법에 의한 부분곱 줄 수의 비교를 나타낸다. 곱셈계수의 워드길이가 W 일 경우 항상 $W/2$ 개의 부분곱을 생성하는 modified Booth 곱셈기와 달리 GCSD 곱셈기는 곱셈계수들의 종류에 따라 $W/2$ 개, 또는 그 이하의 부분곱을 생성함으로써 modified Booth 곱셈기를 사용한 것보다 부분곱의 줄 수를 최대 62.5%까지 감소 시키며, 평균적으로 약 42%정도 감소시킴을 보인다.

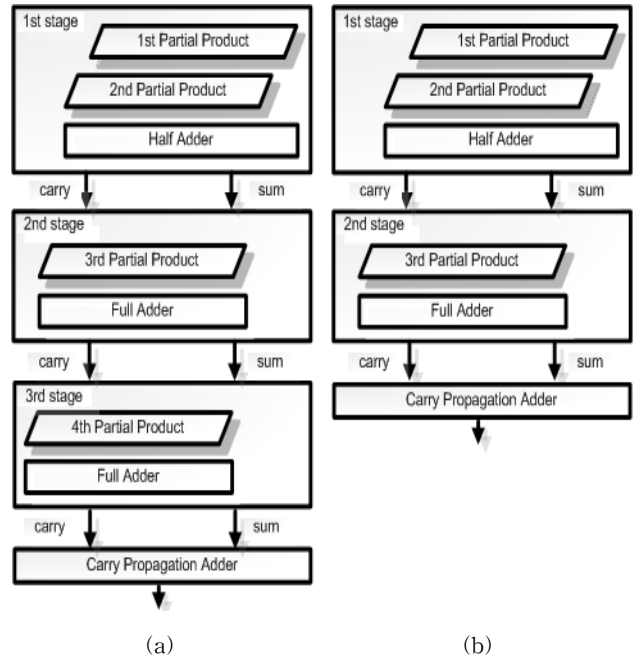


그림 4 곱셈기의 구조: (a) 부분곱이 4줄일 경우, (b) 부분곱이 3줄일 경우

Fig. 4 Multiplier structure: (a) 4 lines of partial products, (b) 3 lines of partial products

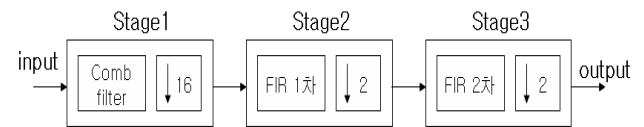


그림 5 데시메이션 필터 구조

Fig. 5 Decimation filter structure

표 4 그림 5의 stage 3 FIR 필터 16 비트 계수

Table 4 FIR filter in stage 3 with 16bit coefficients in Fig 5

곱셈기	필터계수	2진수로 표현한 필터 계수																	
		h0	h1	h2	h3	h4	h5	h6	h7	h8	h9	h10	h11	h12	h13	h14	h15		
M1	h0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	
	h1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0
	h2	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1
	h3	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1
M2	h4	1	1	1	1	1	1	1	1	1	1	0	0	1	1	0	1	1	1
	h5	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	1
	h6	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0
	h7	1	1	1	1	1	1	1	1	1	1	1	0	1	0	0	0	0	1

M3	h8	1	1	1	1	1	1	1	1	0	1	1	1	1	0	0	1
	h9	0	0	0	0	0	0	0	0	1	0	0	1	1	1	1	0
	h10	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1
	h11	1	1	1	1	1	1	1	1	0	0	0	0	1	0	1	1
M4	h12	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0
	h13	0	0	0	0	0	0	0	1	0	1	1	0	0	0	1	1
	h14	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0
	h15	1	1	1	1	1	1	1	0	0	0	0	1	1	0	1	1
M5	h16	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
	h17	0	0	0	0	0	0	1	0	0	1	1	1	0	1	1	1
	h18	1	1	1	1	1	1	1	1	0	1	0	0	0	1	0	1
	h19	1	1	1	1	1	1	0	0	1	1	1	1	0	0	0	0
M6	h20	0	0	0	0	0	0	1	1	1	0	0	1	1	1	1	0
	h21	0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	0
	h22	1	1	1	1	1	1	0	0	1	0	0	0	0	0	0	0
	h23	1	1	1	1	1	0	1	1	1	1	0	1	0	1	1	1
M7	h24	0	0	0	0	0	1	1	0	0	1	0	1	1	1	1	1
	h25	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	1
	h26	1	1	1	1	0	0	1	1	1	0	0	0	1	0	1	0
	h27	1	1	1	1	1	0	1	1	0	0	1	0	1	0	1	0
M8	h28	0	0	1	0	1	0	0	0	1	0	1	1	1	1	0	0
	h29	0	1	0	0	0	1	0	0	1	1	1	0	1	0	1	1

표 5 표 4를 GCSD방법으로 그룹화하여 표현한 필터 계수
 Table 5 The filter coefficients grouped with GCSD method in Table 4

곱셈기	필터계수	2진수로 표현한 필터 계수																
M1	h0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	-1	0
	h1	0	0	0	0	0	0	0	0	0	1	0	0	-1	0	0	1	0
	h2	0	0	0	0	0	0	0	0	0	1	0	0	0	-1	0	0	0
	h3	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	1	0	0
M2	h4	0	0	0	0	0	0	0	0	-1	0	1	0	0	-1	0	0	-
	h5	0	0	0	0	0	0	0	0	0	1	0	-1	0	1	0	0	-
	h6	0	0	0	0	0	0	0	0	1	0	0	-1	0	1	0	0	0
	h7	0	0	0	0	0	0	0	0	-1	0	1	0	0	0	0	0	1
M3	h8	0	0	0	0	0	0	0	0	-1	0	0	0	-1	0	0	0	1
	h9	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	-1	0
	h10	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	-
M4	h11	0	0	0	0	0	0	0	-1	0	0	0	1	0	-1	0	0	-
	h12	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	-
	h13	0	0	0	0	0	0	1	0	-1	0	-1	0	0	1	0	0	-
	h14	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0
M5	h15	0	0	0	0	0	0	-1	0	0	0	1	0	0	-1	0	0	-
	h16	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
	h17	0	0	0	0	0	0	1	0	1	0	0	0	-1	0	0	0	-
	h18	0	0	0	0	0	0	0	-1	0	1	0	0	0	1	0	1	1
M6	h19	0	0	0	0	0	-1	0	1	0	0	0	-1	0	0	0	0	0
	h20	0	0	0	0	0	1	0	0	-1	0	1	0	0	0	-1	0	0
	h21	0	0	0	0	0	1	0	0	-1	0	1	0	0	1	0	0	0
	h22	0	0	0	0	0	-1	0	0	1	0	0	0	0	0	0	0	0
M7	h23	0	0	0	0	0	-1	0	0	0	0	-1	0	-1	0	0	0	-
	h24	0	0	0	0	1	0	-1	0	1	0	-1	0	0	0	0	0	-
	h25	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	1	1
	h26	0	0	0	-1	0	1	0	0	-1	0	0	0	1	0	1	0	1
M8	h27	0	0	0	0	0	-1	0	-1	0	1	0	-1	0	-1	0	0	0
	h28	0	0	1	0	1	0	0	0	1	0	-1	0	0	-1	0	0	0
	h29	0	1	0	0	0	1	0	1	0	0	0	-1	0	-1	0	0	-

표 6 Modified Booth 곱셈과 GCSD 곱셈의 부분곱 줄 수 비교

Table 6 Comparison of the number of partial product lines with modified Booth multiplier and GCSD multiplier

곱셈기	Modified Booth 부분곱 줄 개수	GCSD 부분곱 줄 개수	감소효율 (%)
M1	8	3	62.5
M2	8	4	50
M3	8	4	50
M4	8	5	37.5
M5	8	4	50
M6	8	5	37.5
M7	8	6	25
M8	8	6	25

4. 결 론

본 논문에서는 DSP 응용 분야에서 널리 사용되는 time-multiplexed FIR필터에 GCSD 곱셈기를 적용함으로써 보다 효율적인 필터 구현이 가능함을 보였다. 데시메이션 필터에 적용한 결과 modified Booth 곱셈 방식에 비해 평균 약 42%의 감소효율이 있음을 보였다. 대부분의 FIR 필터는 근접한 필터 계수들끼리 유사한 크기를 가지므로, 제한한 방식을 time-multiplexed FIR 필터의 구현에 사용시 면적 및 속도 면에서 많은 향상이 있을 것으로 기대된다.

감사의 글

본 연구는 지식경제부 및 한국산업기술평가관리원의 IT산업원천기술개발사업[B1120-0901-0002, IT특화 연구소설립]과 2단계 BK21 사업 지원에 의하여 이루어진 연구로서, 관계부처에 감사 드립니다.

참 고 문 헌

[1] MacSorley, "High speed arithmetic in binary computers", Proc. IRE, vol. 49, pp. 67-91, 1961.
 [2] Y. E. Kim, S. H. Cho and J. G. Chung, "Modified CSD group multiplier design for predetermined coefficient groups", in Proc. IEEE ISCAS 2008, pp. 3362-3365, May, 2008.
 [3] Y. E. Kim, K. J. Cho, J. G. Chung and X. Huang "CSD-based programmable multiplier design for predetermined coefficient groups", IEICE Trans. (to appear).
 [4] Y. E. Kim, K. J. Cho and J. G. Chung "Low power small area modified Booth multiplier design for predetermined coefficients", IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. E90-A, pp. 694-697, Mar. 2007.

- [5] S. W. Reitwiesner, "Binary arithmetic," *Advances in Computers*, pp. 231-308, 1966.
- [6] 변산호, 류성영, 최영길, 노형동, 남현석, 노정진, "시그마-델타 A/D 컨버터용 디지털 테시메이션 필터 설계", *대한전자공학회 논문지*, pp. 34-45, 2007.



이철동 (李哲東)

1952년 2월 12일생. 1977년 경북대 전자공학과 졸업. 1989년 한양대 전자공학과 졸업(석사). 1989년 충북대 컴퓨터공학과 졸업(공학). 1994년~현재 전자부품연구원 센터장/본부장

Tel : 063-710-7710

E-mail : leecd@keti.re.kr

저 자 소 개

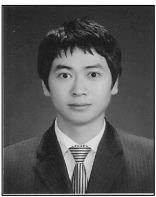


전창하 (全窓夏)

1981년 11월 18일생. 2009년 전북대 전자정보공학부 졸업. 2009년~현재 동 대학원 전기전자컴퓨터공학부 석사과정

Tel : 063-270-2466

E-mail : jsure@chonbuk.ac.kr



서동현 (徐東賢)

1981년 5월 15일생. 2009년 전북대 전자정보공학부 졸업. 2009년~현재 동 대학원 전기전자컴퓨터공학부 석사과정

Tel : 063-270-2466

E-mail : sdhyun@chonbuk.ac.kr



정진균 (鄭鎭均)

1963년 2월 15일생. 1985년 전북대 전자공학과 졸업. 1991년 미국 미네소타 대학교 전기공학과 졸업(공학). 1995년~현재 전북대 전기전자컴퓨터공학부 교수

Tel : 063-270-2466

E-mail : jgchung@chonbuk.ac.kr



김용은 (金溶恩)

1978년 3월 14일생. 2005년 전북대 전자공학과 졸업. 2007년 동 대학원 정보통신공학과 졸업(석사). 2007년~현재 동 대학원 전기전자컴퓨터공학부 박사과정. 2009~현재 자동차부품연구소 연구원

Tel : 041-559-3363

E-mail : yekim@katech.re.kr