양방향 알고리즘을 이용한 2의 보수 표현 기법에 의한 디지털 필터의 설계에 관한 연구

이영석*

Study on Design of Digital filter by 2's Complement Representation using Bidirectional algorithm

Youngseock LEE*

요 약

디지털 신호 처리 분야에서 디지털 필터는 필수불가결한 요소이다. 디지털 필터는 이진수의 곱셈 및 덧셈을 기본으로 하는 것으로 많은 연산량을 필요로 한다. 디지털 필터 내의 곱셈기는 VLSI 기술을 이용한 디지털 필터의 설계에 있어 반도체 칩 내부의 넓은 영역을 차지하고 전력의 대부분을 소비하며, 필터의 critical path를 결정하여 필터의 성능을 결정하는 중요한 요소로서 작용 한다. 본 연구에서는 특히 디지털 필터의 복잡성(complexity)를 해소하고 critical path를 줄여 필터의 연산 속도를 증가시키기 위한 방법으로 2의 보수로서 표현되는 이진수를 CSD(canonical signed digit)와 MSD(minimal signed digit)로 동시에 변환하여 표현하는 방법을 제안하였다. 제안된 방법은 VHDL로 구현하고 임의의 필터에 적용하여 필터의 critical path가 감소하는 것을 보였다.

ABSTRACT

The digital filter is essential element in digital signal processing area. It needs a high computational burden caused by multiplying and adding. The multiplier in digital filter is a dominant element, which occupies an wide area at the field of VLSI design, needs high power-consuming and also decides critical path that affects to filter performance. In this paper we proposed the simultaneous transform method which is represented 2's complementary representation to CSD and MSD representation to solve a complexity problem and to improve a computational speed. The performance of proposed method was implemented in VHDL and applied to an digital filters, was evaluated the decreasing of critical path delay.

Key word : Digital filter, CSD, MSD, critical path, VHDL

Ⅰ. 서 론

디지털 필터는 다양한 신호 처리 분야에서 필수 불가결하게 사용하는 요소이다. 디지털 필터는 기본적으로 덧셈기와 곱셈기를 이용하여 구현할 수 있으며, VLSI 설계에 있어 곱셈기는 반도체 내부 회로에서 많은 면적을 차지하고 높은 전력을 소비하며 VLSI 설계 및 생산에서

* 청운대학교 디지털방송공학과(yslee@chungwoon.ac.kr)

접수일자 : 2009.01.30
완료일자 : 2009.02.13
접수번호 : KIIECT2009-01-13

많은 비용을 소모되는 부분이다. 또한 많은 연산량으로 인하여 반도체 회로의 입력 신호로부터 출력 신호가 나오기 까지 반드시 소요되는 critical path 지연(delay)에 큰 영향을 미친다. 일반적으로 이진수의 곱셈은 이진수를 1 비트씩 이동시키면서 부분 곱들(partial products)을 구하고 이들을 더하여 구할 수 있다. 따라서 일반적인 곱셈기에서 N 비트의 곱셈을 수행하는 경우에 N번의 곱셈과 N-1번의 덧셈 연산을 필요로 한다. 곱셈기의 연산 부담을 줄이기 위한 방법으로서 Wallace tree 구조나 carry save adder tree 구조 등이 제안되었으나, 보다 기본적인 문제는 N 비트의 이진수에 존재하는 이진수 1의 개수에 관한 것이다. 만약 N 비트의 이진수 안에 있는 1의 개수를 줄일 수 있는 새로운 숫자 표현 방법이 있다면 곱셈기의 연산량을 줄일 수 있을 것이다. Lim 등[1]은 2의 보수로 표현되는 N 비트의 이진수를 부호 비트와 절대값으로 표시되는 2진수의 표현 방법을 개발하여 이진수의 1의 개수를 줄이는 알고리즘을 개발 하였으며, Das 등[2]은 CSD(canonical signed digit) 방법을 이용하여 이진수의 1의 개수를 줄이고 0의 개수를 늘려 곱셈기의 연산 부담을 줄일 수 있는 알고리즘을 VLSI 설계에 응용하였다. CSD 표현 방법은 이진수를 -1, 0, 1을 이용하여 나타내는 방법으로서 인접한 non-zero 비트를 허용하지 않는 표현 기법이다. 평균적으로 non-zero 비트를 33% 정도 줄일 수 있는 것으로 알려져 있다[3]. 이는 곱셈기의 부분곱 및 덧셈 연산을 약 1/3로 줄일 수 있다는 것을 의미한다. 그러나 CSD 연산을 수행하면 소비되는 시간 지연으로 인한 critical path 문제는 해결 될 수 없다. 본 연구에서는 디지털 필터의 critical path의 지연 문제를 해결하기 위하여 이진수의 CSD 표현 방법 및 Lim 등에 의해 제안된 MSD(minimum signed digit) 표현 방법을 동시에 수행하는 방법을 제안하였다. 제안된 방법은 디지털 필터에 적용하여 제안한 방법이 critical path 지연이 감소하는 것을 확인할 수 있었다.

II. 이진수의 CSD 표현

CSD는 최소 해밍 거리(minimum Hamming distance)를 갖는 부호 있는 디지트 표현(signed digit representation)이다, 따라서 2진수를 CSD로 표현 할 경우 2진수에서 1의 개수를 최소화할 수 있으며 인접한 비트들 사이에 non-zero 값을 허용하지 않는다. 2의 보수를 CSD 표현으로 나타낼 경우에 각 코드값은 표 1과 같이 나타낼 수 있다.

표 1. 2의 보수의 CSD 표현.
Table. 1 CSD representation of 2's complementary number.

b_{i+1}	b_i	c_i	a_i	c_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	0	0
1	0	1	-1	1
1	1	0	-1	1
1	1	1	0	1

표 1 계속

표 1로부터 b_i들은 CSD로 변환 할 2의 보수들이고 a_i는 CSD로 변환된 수이다. c_i는 $i-1$ 과정에서 생성된 올림수(carry)를 나타내고 c_{i+1}은 i 번째 과정에서 발생한 올림수를 나타내며 $i+1$번째 과정에서 b_{i+1}과 b_{i+2}을 이용하여 연산을 수행하는 경우에 사용된다. CSD 알고리즘을 수행하는 과정은 2의 보수를 LSB(least significant bit)로부터 MSB(most significant bit)로 2비트를 쌍으로 하여 1비트씩 윈도우를 옮겨가면서 표 1에 나타난 논리식을 통하여 수행되며 그림 1과 같이 나타낼 수 있다.

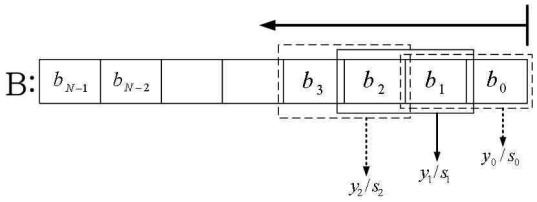

그림 1. CSD 알고리즘의 구성도.
Fig. 1. Block diagram of CSD algorithm.

CSD 알고리즘의 문제점은 그림 1과 같은 과정을 통하여 2비트씩 작을 이루어 오른쪽에서 왼쪽으로 자리가 이동하는 경우에 올림수의 전파(propagation) 과정에서 발생하는 리플(ripple)이 critical path에 영향을 미친다는 점이다. 그러므로 비트 수가 증가하면 증가할수록 올림수 리플에 의한 전파 지연이 증가하게 된다.

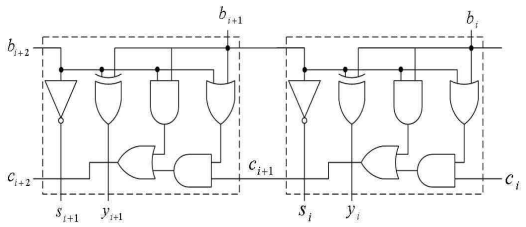

그림 2. 2의 보수의 CSD 알고리즘을 수행하기 위한 디지털 회로.
Fig. 2. Digital circuit of CSD algorithm for 2's complement number.

그림 2는 표 1에 나타낸 논리식을 이용하여 구현한 디지털 회로를 나타내고 있다. 점선으로 나타낸 박스 부분이 그림 1의 윈도우에 해당하는 부분으로서 윈도우가 좌측에서 우축으로 이동하는 경우 그림 2의 회로에서 올림수 c_{i+1}이 왼쪽의 점선으로 나타낸 박스로 이동해야만 왼쪽의 회로에서 출력을 나타낼 수 있다. 그러므로 올림수에 의한 전파 지연을 최소화 하면서 CSD표현 방법에 의해 non-zero 성분을 최소화 할 수 있는 알고리즘의 구성이 요구된다.

Ⅲ. 이진수의 MSD 표현

Lim 등에 의해 연구된 MSD 표현 방법은 CSD 알고리즘과 달리 그림 3과 같이 2진수의 MSB로부터 LSB로 3비트를 한 쌍으로 하여 1비트씩 이동하면서 변환을 수행한다.

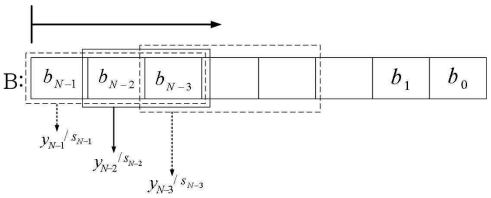

그림 3. Lim의 MSD 알고리즘의 구성도.
Fig. 3. Block diagram of Lim's MSD algorithm.

Lim의 MSD 표현 방법은 식 (1)과 같이 나타낸 2의 보수를 식 (2)와 같은 표현으로 나타내고자 하는 것으로서 회귀적인(recursive) 방법을 이용하여 구할 수 있으며 표 2와 같은 입력 및 결과 식을 갖는다.

$$x = -b(B-1)2^{B-1} + \sum_{r=0}^{B-2} b(r)2^r, b(r) = 0,1 \quad (1)$$

$$y_m = -c_m(B-1)2^{B-1} + \sum_{r=0}^{B-2} c_m(r)2^r, c_m(r) = 0,1 \quad (2)$$

위 식들로부터 B는 2진수의 비트 수를 나타내고 y_m은 m번째 회귀를 한 MSD의 출력을 나타낸다. MSD의 출력은 표 1의 CSD의 표현과 마찬가지로 논리표로서 표 2와 같이 나타낼 수 있다[4].

표 2. Lim의 2의 보수에 대한 MSD 표현.
Table. 2. Lim's MSD representation of 2's complement number.

d_i	b_i	b_{i-1}	b_{i-2}	y_i	d_{i-1}
0	0	0	x	0	0
0	0	1	0	0	1
0	0	1	1	1	1

0	1	0	0	-1	1
0	1	0	1	0	1
0	1	1	x	0	0
1	0	0	x	0	0
1	0	1	x	-1	0
1	1	0	x	1	0
1	1	1	x	0	0

그림 4는 표 2의 논리식을 바탕으로 하여 3 비트의 입력 신호 x_{i-1}, x, x_{i+1}에 대한 출력을 나타내고 있다. 출력 신호 s_i는 각 비트에서의 부호를 나타내며 $s_i = 0$이면 음수를, $s_i = 1$이면 음수를 나타낸다. y_i는 0 또는 1로 표현된다. 따라서 입력 신호 x_{i-1}, x, x_{i+1}에 대한 출력은 부호를 나타내는 s_i와 값을 나타내는 y_i로 나타낼 수 있다. 또한 점선으로 나타낸 부분은 1 비트를 MSD를 이용하여 변환시키기 위한 회로셀을 나타내고 있다.

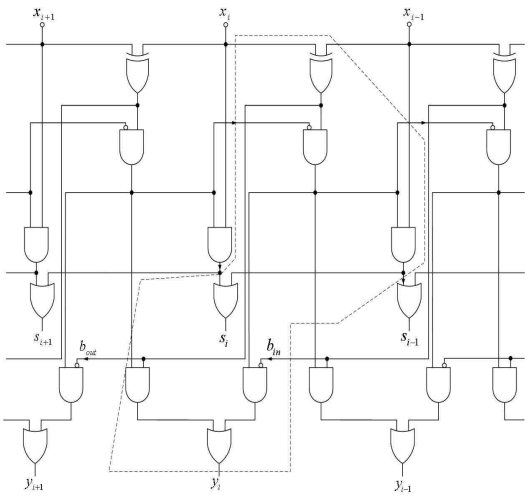

그림 4. Lim의 MSD 알고리즘의 구성도.
Fig. 4. Block diagram of Lim's MSD algorithm.

Ⅳ. 양방향 알고리즘

Critical path를 감소시켜 변환 속도를 높이기 위하여 본 연구에서는 양방향 알고리즘을 적용하였다. 적용된 알고리즘은 MSB로부터 LSB 방향으로 진행되는 MSD 표현 방법과 LSB로부터 MSB 방향으로 진행하는 CSD 표현 방법을 조합하여 그림 5와 같이 동시에 두 종류의 변환이 진행하도록 하는 것이다.

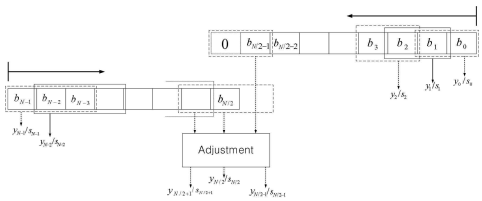

그림 5. 양방향 알고리즘의 구성도.
Fig. 5. Block diagram of bidirectinal algorithm.

두 표현 방법은 모두 최소 해밍 거리를 갖는 변환이기 때문에 두 종류의 변환을 진행하면서 non-zero 성분의 redundancy는 발생하지 않는다. 또한 CSD 알고리즘을 진행하면서 발생하는 올림수 리플의 양을 반으로 줄일 수 있기 때문에 CSD 알고리즘에 의해 발생하는 critical path를 감소시킬 수 있다. 양방향 알고리즘은 다음의 단계에 의하여 진행된다.

(1) 입력을 $B=\{b_{N-1},\cdots,b_1,b_0\}$로 나타내고, 출력을 $Y=\{y_{N-1},\cdots,y_1,y_0\}$로 나타낸다. 이때, 입력은 N 비트 2의 보수이고, 출력은 N 비트 MSD 표현 방식으로 나타난다고 가정한다.

(2) 입력 B를 B_L과 B_R로 나누어 B_L은 B의 상위 절반 비트들로, B_R은 하위 절반 비트들로 구성된다. 즉, $B_L=\{b_{N-1},\cdots b_{N/2}\}$ $B_R=\{b_{N/2-1},\cdots,b_0\}$.

(3) B_R의 MSB에 0을 삽입하여 양의 2의 보수로 나타내도록 한다. 즉, $B_R=\{0,b_{N/2-1},\cdots,b_0\}$로 수정한 다음 B_L은 Lim의 MSD 알고리즘을 수

행하고, B_R은 CSD 알고리즘을 수행하여 B_L과 B_R의 출력을 각각 Z_L과 Z_R로 나타낸다.

(4) Z_R의 비트 중 MSB를 제외한 값을 출력 Y의 오른쪽 출력 $Y_R^{'}$으로 나타낸다.

(5) Z_R의 LSB와 Z_L의 MSB를 더하고 더한 결과가 2이면 Z_L의 LSB 비트 및 LSB+1 비트에 $\{0,1\}$을 할당한다.

(6) $Y_L^{'}$에 Z_L을 할당하고 $Y^{'}=\{Y_L^{'},Y_R^{'}\}$으로 나타낸다.

(7) $Y_L^{'}$의 하위 2비트가 $\{-1,1\}$이면 $\{0,-1\}$로 할당한다.

(8) (7)에 의해 갱신된 $Y_L^{'}$의 LSB와 $Y_R^{'}$의 MSB가 $\{-1,1\}$이면 $\{0,1\}$을 할당한다.

(9) $Y=\{Y_L^{'},Y_R^{'}\}$로 출력을 나타낸다.

V. 실험 및 결과

Lim의 MSD 알고리즘과 CSD 알고리즘을 결합한 2의 보수에 대한 양방향 알고리즘은 16비트의 2의 보수에 적용할 수 있도록 VHDL로 구현하였다. 그리고 실험은 CSD 알고리즘 및 MSD 알고리즘을 단독으로 수행 하였을 때와 양방향 알고리즘을 수행하였을 때 지연 시간을 측정하고 비교하였다. 표 3은 실험을 위하여 사용한 디지털 필터의 계수 및 2의 보수, CSD 표현, Lim의 MSD 표현, 양방향 알고리즘 표현을 나타내며 $\bar{1}$은 -1을 나타낸다.

표 3. 디지털 필터 계수, 2의 보수, CSD, Lim의 MSD 및 양방향 알고리즘 표현.
Table. 3. Digital filter coefficients, 2's complement number, CSD, Lim' MSD and bidirectional algorithm representations.

Filter coeff.	2's comp.	CSD	Lim's MSD	Bidirect.
0.0004	0000 0000 0000 0100	0000 0000 0000 0010	0000 0000 0000 0010	0000 0000 0000 0010
-0.0006	1111 1111 1111 1010	0000 0000 0000 $\bar{1}$010	0000 0000 0000 0$\bar{1}\bar{1}$0	0000 0000 0000 $\bar{1}$010
-0.0012	1111 1111 1111 0100	0000 0000 000$\bar{1}$ 0100	0000 0000 0000 $\bar{1}\bar{1}$00	0000 0000 000$\bar{1}$ 0100
-0.0008	1111 1111 1111 1000	0000 0000 0000 $\bar{1}$000	0000 0000 0000 $\bar{1}$000	0000 0000 0000 $\bar{1}$000
0.0007	0000 0000 0000 0111	0000 0000 0000 100$\bar{1}$	0000 0000 0000 100$\bar{1}$	0000 0000 0000 100$\bar{1}$
0.0024	0000 0000 0000 1000	0000 0000 0010 $\bar{1}$000	0000 0000 0010 $\bar{1}$000	0000 0000 0010 $\bar{1}$000
0.0022	0000 0000 0001 0110	0000 0000 0010 $\bar{1}$0$\bar{1}$0	0000 0000 0001 10$\bar{1}$0	0000 0000 0010 $\bar{1}$0$\bar{1}$0
-0.0008	1111 1111 1111 1000	0000 0000 0000 $\bar{1}$000	0000 0000 0000 $\bar{1}$000	0000 0000 0000 $\bar{1}$000
-0.0046	1111 1111 1101 0010	0000 0000 0$\bar{1}$01 0010	0000 0000 00$\bar{1}\bar{1}$ 0010	0000 0000 0$\bar{1}$01 0010
-0.0051	1111 1111 1100 1101	0000 0000 0$\bar{1}$01 0$\bar{1}$01	0000 0000 0$\bar{1}$01 00$\bar{1}\bar{1}$	0000 0000 0$\bar{1}$01 0$\bar{1}$01
0	0000 0000 0000 0000	0000 0000 0000 0000	0000 0000 0000 0000	0000 0000 0000 0000
0.0076	0000 0000 0100 1100	0000 0000 0101 0$\bar{1}$00	0000 0000 0101 0$\bar{1}$00	0000 0000 0101 0$\bar{1}$00
0.0102	0000 0000 0110 0110	0000 0000 10$\bar{1}$0 10$\bar{1}$0	0000 0000 10$\bar{1}$0 10$\bar{1}$0	0000 0000 10$\bar{1}$0 10$\bar{1}$0
0.0026	0000 0000 0001 1010	0000 0000 0010 $\bar{1}$010	0000 0000 0010 0$\bar{1}\bar{1}$0	0000 0000 0010 $\bar{1}$010
-0.0112	1111 1111 1001 0000	0000 0000 $\bar{1}$001 0000	0000 0000 $\bar{1}$001 0000	0000 0000 $\bar{1}$001 0000
-0.0185	1111 1111 0100 0111	0000 000$\bar{1}$ 0100 100$\bar{1}$	0000 0000 $\bar{1}\bar{1}$00 100$\bar{1}$	0000 000$\bar{1}$ 0100 100$\bar{1}$
-0.0086	1111 1111 1010 1010	0000 0000 $\bar{1}$010 1010	0000 0000 0$\bar{1}$0$\bar{1}$ 0$\bar{1}\bar{1}$0	0000 0000 $\bar{1}$010 1010
0.0148	0000 0000 1001 0100	0000 0000 1001 0100	0000 0000 1001 0100	0000 0000 1001 0100
0.0139	0000 0001 0011 1111	0000 0001 0100 000$\bar{1}$	0000 0001 0100 000$\bar{1}$	0000 0001 0100 000$\bar{1}$
0.0214	0000 0000 1101 0110	0000 0001 00$\bar{1}$0 $\bar{1}$0$\bar{1}$0	0000 0001 00$\bar{1}$0 $\bar{1}$0$\bar{1}$0	0000 0001 00$\bar{1}$0 $\bar{1}$0$\bar{1}$0
-0.018	1111 1111 0100 1100	0000 0001 0101 0$\bar{1}$00	0000 0000 $\bar{1}\bar{1}$01 0$\bar{1}$00	0000 0001 0101 0$\bar{1}$00
-0.0574	1111 1101 1100 0010	0000 00$\bar{1}$0 0$\bar{1}$00 0010	0000 00$\bar{1}$0 0$\bar{1}$00 0010	0000 00$\bar{1}$0 0$\bar{1}$00 0010
-0.0534	1111 1101 1110 1010	0000 00$\bar{1}$0 0$\bar{1}$00 1010	0000 00$\bar{1}$0 0$\bar{1}$00 1010	0000 00$\bar{1}$0 0$\bar{1}$00 1010
0.0201	0000 0000 1100 1001	0000 0001 00$\bar{1}$0 1001	0000 0001 00$\bar{1}$0 1001	0000 0001 00$\bar{1}$0 1001
0.1452	0000 0101 1010 1100	0000 10$\bar{1}$0 0$\bar{1}$0$\bar{1}$ 0$\bar{1}$00	0000 0110 0$\bar{1}$0$\bar{1}$ 0$\bar{1}$00	0000 10$\bar{1}$0 0$\bar{1}$0$\bar{1}$ 0$\bar{1}$00
0.2641	0000 1010 0101 0001	0000 1010 0101 0001	0000 1010 0101 0001	0000 1010 0101 0001
0.3129	0000 1100 0011 1001	0001 0$\bar{1}$00 0100 $\bar{1}$001	0001 0$\bar{1}$00 0100 $\bar{1}$001	0001 0$\bar{1}$00 0100 $\bar{1}$001

표 4는 CSD, Lim의 MSD 및 양방향 알고리즘을 VHDL로 구현하고 임의의 입력에 대한 모의 실험을 수행하였을 때 입력에 대한 출력의 지연 시간을 나타내고 있다.

표 4. CSD, Lim의 MSD 및 양방향 알고리즘의 지연 시간 비교.
Table. 4. Comparison of delay time for CSD, Lim's MSD and bidirectional algorithm.

Method	CSD	Lim's MSD	Bidirect.
Delay[ns]	7.2	5.5	3.7

표 4의 결과로부터 양방향에서 동시에 변환을 수행하는 알고리즘이 CSD나 Lim의 MSD에 비하여 지연 시간이 감소 한 것을 관측할 수 있다. 그러므로 양방향 알고리즘은 디지털 필터의 구현에 있어 지연 시간을 감소 시켜 필터 시스템의 critical path를 줄일 수 있다. CSD 표현 방법을 이용하여 2의 보수를 나타내는 시스템의 경우 설계 및 구현은 그림 2와 같이 간단하지만 올림수 리플에 의해 critical path가 길어지는 단점이 있고, Lim의 MSD 표현 방법은 그림 3에서 나타내는 바와 같이 올림수 리플의 영향을 CSD 표현 방법에 비하여 상대적으로 덜 받기 때문에 시스템의 critical path는 감소하는 장점을 갖고 있으나, 시스템이 복잡해지면서 발생하는 반도체 내부의 영역의 증가 및 전력 소비 문제가 나타날 수 있다. 따라서 critical path 문제를 해결하면서 시스템의 복잡도를 줄이는 방법으로서 양방향 알고리즘은 유용한 방법이라 할 수 있다.

VI. 결 론

본 연구에서는 디지털 필터의 기본 구성 요소인 곱셈기 및 덧셈기 구현에서 발생하는 critical path를 감소시켜 필터 연산의 속도를 향상시키기 위한 방법으로서 비트 스트림의 MSB와 LSB로부터 동시에 시작되는 최소 해밍 거리를 갖는 2의 보수 표현 시스템을 개발 하였다. VHDL로 구현된 시스템은 필터 계수에 대한 출력을 통하여 최소 해밍 거리를 갖고 있음을 확인하였고, CSD, Lim의 MSD 및 양방향 알고리즘에 대하여 입력에 대한 출력의 지연 시간을 측정하여 critical path가 감소하여 시스템의 연산 속도가 증가하는 것을 확인하였다.

참 고 문 헌

[1] Y. C. Lim, J.B. Evance and B. Liu, "Decomposit- ion of binary integers into signed power of two terms," IEEE Trans. on Circuits and Sytems, vol. 38, no. 6, pp. 667-672, June 1991.

[2] S. K. Das and M. C. Pinotti, "Fast VLSI circuits for CSD coding and GNAF coding," Electronics Letters, vol. 32, no. 7, pp. 632-634, Mar 1996.

[3] H. L. Garner, "Number systems and arithmetic," Advanced in Computers, vol. 6, pp. 131-194, 1965.

[4] M. Joyes and s. M. Yen, "Optimal left to right binary signed digit recording," IEEE Trans. Computers, vol. 49, no. 7, pp. 740-748, 261-265, July 2000.

저자약력

이 영 석(Youngseock LEE)

1993년 서울시립대학교 전자공학과 졸업.
1995년 동대학원 전자공학과 졸업(공학석사).
1998년 동대학원 전자공학과 졸업(공학박사).
1998년~현재 청운대학교 디지털방송공학과 부교수.

<관심분야> VLSI신호처리, SoC, 임베디드 시스템