DBpia

논문 02-27-6C-11 한국통신학회논문지 '02-6 Vol.27 No.6C

전파캐리의 선택에 의한 부호확장 오버헤드의 감소

정회원 조 경 주*, 김 명 순**, 유 경 주***, 정 진 균*

Sign-Extension Overhead Reduction by Propagated-Carry Selection

Kyung-Ju Cho*, Myung-Soon Kim**, Kyung-Ju Yoo***, Jin-Gyun Chung* *Regular Members*

요 약

고정계수를 갖는 곱셈기의 구현 시 면적과 전력소모를 줄이기 위해서 곱셈계수를 CSD(Canonic Signed Digit) 형태로 표현 할 수 있다. CSD 계수의 1 또는 -1의 위치에 따라 부분곱들을 시프트 하여 더할 때 모든 부분곱들의 부호확장이 필요하며 이로 인해 하드웨어의 오버헤드가 증가하게된다. 본 논문에서는 부호확장 부분에서의 캐리전파를 적절히 조절함으로써 부호확장으로 인한 오버헤드를 조절 할 수 있다는 사실을 이용하여 새로운 부호확장 오버헤드감소 방법을 제시한다. 또한 CSD 곱셈기에 적합한 고정길이 곱셈기의 구조를 제시하고 전파캐리선택 절차를 이용한 부호확장 제거방법과 결합함으로서 CSD 곱셈기를 효율적으로 구현할 수 있음을 보인다. 이 곱셈기의 응용으로서 SSB/BPSK-DS/CDMA 전송방식에 사용되는 힐버트 트랜스포머를 43탭 FIR 필터로 구현하고 기존의 compensation 벡터방법과 비교하여 nonzero 비트수에 따라 약 16~28%의 부호확장 오버헤드를 줄일 수 있음을 보인다.

ABSTRACT

To reduce the area and power consumption in constant coefficient multiplications, the constant coefficient can be encoded using canonic signed digit(CSD) representation. When the partial product terms are added depending on the nonzero bit(1 or -1) positions in the CSD-encoded multiplier, all sign bits are properly extended before the addition takes place. In this paper, to reduce the overhead due to sign extension, a new method is proposed based on the fact that carry propagation in the sign extension part can be controlled such that a desired input bit can be propagated as a carry. Also, a fixed-width multiplier design method suitable for CSD multiplication is proposed. As an application, 43-tap Hilbert transformer for SSB/BPSK-DS/CDMA is implemented. It is shown that, about 16~28% adders can be saved by the proposed method compared with the conventional methods.

I. 서 론

신호처리 알고리즘의 하드웨어 구현 시 가장 큰 면적 및 전력소모를 차지하는 회로 중의 하나가 곱셈기이며, 곱셈기를 효율적으로 구현하기 위한 많은 연구가 진행되었다. 고정계수를 갖는 곱셈기의 경우 계수를 CSD 형태로 표현함으로써 계수 중 0이 아닌 디지트의 개수를 감소시킬 수 있고 이에 따라 곱셈기의 면적 및 전력소모를 감소시킬 수 있다[1].

CSD 곱셈에서는 계수 중 1이나 -1의 위치에 따라 부분곱들을 시프트하여 더하며 이 때 부호확장이 선행되어야 하는데 이로 인해 하드웨어의 복잡도와 부호비트에 해당하는 데이터버스의 로드가 증가하게 된다. 이를 줄이기 위해 compensation 벡터를 만들어 부분곱의 합에 더해주는 방법이 사용되었다[2].

많은 멀티미디어와 DSP 응용에서 곱셈 연산은 입력과 출력 데이터의 길이가 같은 고정길이 곱셈기가 요구된다. 예를 들면 W 비트의 승수와 W 비

* 전북대학교 전자정보공학부(kjcho@vlsidsp.chonbuk.ac.kr, jgchung@moak.chonbuk.ac.kr),
** 한국전자통신연구원 무선방송연구소 무선 LAN팀(mskim75@etri.re.kr)
*** (주)로직메카부설멀티미디어통신연구소(raceyou@logicmeca.com)
논문번호 : 010228-0827, 접수일자 : 2001년 8월 27일

Copyright (C) 2003 NuriMedia Co., Ltd.
www.dbpia.co.kr

트의 피승수의 곱셈 출력은 (2*W*-1)비트인데 LSB (Least Significant Bit)로부터 (*W*-1)비트를 버림(truncation)하여 양자화된 *W* 비트를 출력하게 된다. 전형적인 고정길이 곱셈기 디자인 방법에서는 adder cell의 반을 생략한후 확률적 추정에 근거한 적절한 보상 바이어스를 더하여 보상하는 방식을 사용하며 일반적인 병렬 곱셈기와 비교하여 약 50%의 면적을 줄일 수 있다[3-5].

본 논문에서는 compensation 벡터보다 더욱 효율적인 부호확장 오버헤드감소 방법과 CSD 곱셈기에 적합한 새로운 고정길이 곱셈기의 구조를 제안한다. Ⅱ절에서 전파캐리선택에 의한 새로운 부호확장 오버헤드감소 방법을 제안하고, Ⅲ절에서 CSD 곱셈기에 적합한 새로운 고정길이 곱셈기의 구조를 제시한다. Ⅳ절에서 Ⅱ절과 Ⅲ절의 방법을 응용하여 SSB/BPSK-DS/ CDMA 전송방식에 사용되는 힐버트 트랜스포머를 43탭 FIR 필터로 구현하고 각 곱셈기를 기존의 compensation 벡터방법과 비교한 후 Ⅴ절에서 결론을 맺는다.

Ⅱ. 부호확장 오버헤드감소 방법

이 절에서는 기존의 compensaton 벡터방법을 소개하며, 새로운 전파캐리선택 절차를 제시하고 이를 이용한 부호확장 오버헤드감소 방법을 제안한다.

2.1. Compensation 벡터 방법

Compensation 벡터방법에서는 각 시프트된 데이터워드를 두 벡터의 합으로 표현한다. 예를 들어 $x_3.x_2x_1x_0 \times 2^{-3}$은 다음과 같이 표현 될 수 있다.

$$x_3.x_2x_1x_0 \times 2^{-3} = 0.00\overline{x_3}x_2x_1x_0 + 1.111 \quad (1)$$

식 (1)에서 $\overline{x_3}$는 x_3의 complement를 나타낸다. 각 시프트된 데이터워드를 식 (1)과 같이 표현할 때 1로만 구성된 벡터들이 나타나는데 이러한 벡터들을 미리 더함으로써 하나의 벡터로 표현 할 수 있다. 따라서 부호확장으로 인한 오버헤드는 하나의 compensation 벡터를 더하는데 필요한 오버헤드로 감소된다.

다음 계산에 대한 compensation 벡터의 적용 예를 그림 1에 보였다.

$$Y = x_9.x_8x_7x_6x_5x_4x_3x_2x_1x_0 \times 1.0\overline{1}0000\overline{1}01 \quad (2)$$

Compensation 벡터와의 덧셈 시 회로를 더욱 간단히 하기 위하여 다음 식을 사용 할 수 있다.

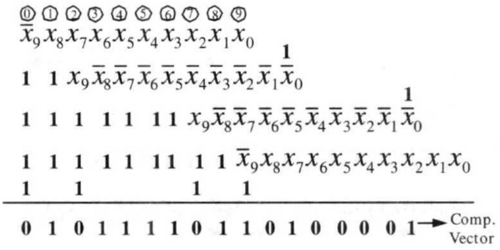

그림 1. 식 (2)에 대한 compensation 벡터방법의 적용.

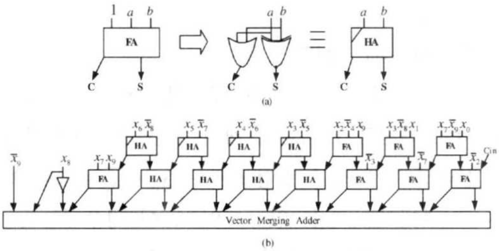

그림 2. 그림 1의 처음 10 컬럼에 해당하는 구현.

표 1. 예측 가능한 sum과 carry 비트

	Input bits	Sum	Carry
Case 1	$a\ a\ a$	a	a
Case 2	$a\ a\ \overline{a}$	$\overline{a}$	a
Case 3	$a\ a\ b$	b	a
Case 4	$a\ \overline{a}\ b$	$\overline{b}$	b

$$1 + a + b \Rightarrow \text{Sum} = \overline{\text{Sum}(a+b)},\ \text{Carry} = a \vee b \quad (3)$$

식 (3)에서 ∨는 OR를 나타낸다. 식 (3)은 그림 2(a)와 같이 구현될 수 있다. 그림 2(b)는 그림 1의 컬럼 0 ~ 9에 해당하는 구현 예를 보여준다.

2.2. 예측 가능한 덧셈

전가산기는 세 비트를 입력으로 받아서 sum과 carry를 계산하여 출력하는 회로이다. 만일 세 입력 비트가 서로 독립적이지 않은 표 1과 같은 경우에는 sum과 carry를 계산 없이 바로 구할 수 있다.

식 (2)에 해당하는 부호확장된 부분곱들은 그림 3과 같다. 원하는 곱셈 결과를 얻기 위해서는 부호확장 부분도 덧셈에 포함되어야한다. 그러나 그림 3에서 알 수 있듯이 처음 3 컬럼은 표1의 Case 2에 해당하고, 다음 5 컬럼은 Case 4에 해당하므로 이들 컬럼의 덧셈 결과는 간단하게 얻을 수 있다. 이러한 사실을 이용하여 새로운 부호확장제거 방법을 제시한다.

Copyright (C) 2003 NuriMedia Co., Ltd.
www.dbpia.co.kr

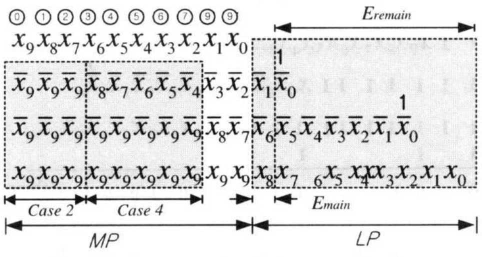

그림 3. 식 (2)에 해당하는 단순 부호확장된 부분곱.

2.3. 전파캐리 선택

그림 3의 컬럼 6을 고려해보면 $\overline{x_5}+\overline{x_9}+x_9$은 Case 4에 해당하기 때문에 캐리는 $\overline{x_5}$가 되어 컬럼 5로 전파된다. 그러나 컬럼 6에서 $x_3+\overline{x_9}+x_9$도 역시 Case 4에 해당함을 알 수 있고, 이 경우에는 x_3가 캐리로 전파된다. 또한, 컬럼 7로부터 전파되어 온 캐리를 컬럼 5로 다시 전파 되도록 선택할 수도 있다. 하드웨어를 감소시키기 위해서는 가능한 모든 경우를 다 고려하여 전파캐리를 선택하여야한다. 이를 위해 다음의 전파캐리선택 절차를 사용한다.

전파캐리선택 절차

주어진 컬럼 i에 대해 가능한 모든 전파캐리 후보들을 결정하여 $p_0, p_1, \ldots, p_M$이라고 한다. 각 전파 캐리 후보 p_j에 대해 다음 조건을 만족하는 가장 큰 컬럼번호 n_j를 결정한다.

1. $n_j < i$,
2. p_j, 또는 $\overline{p_j}$가 컬럼 n_j에 나타난다.

가장 큰 컬럼번호 n_j를 갖는 p_j를 컬럼 i에 대한 전파캐리로 선택한다. 만일 어떠한 캐리후보도 위의 조건을 만족하는 컬럼을 갖지 못하면 임의로 전파캐리를 선택한다.

전파캐리선택 절차를 효과적으로 적용하기 위하여 단순 부호확장과 compensation 벡터방법을 혼합할 수 있다. 이를 이용하여 본 논문에서 제안하는 부호확장 오버헤드감소 방법은 다음과 같다.

2.4 전파캐리선택 절차를 이용한 부호확장 오버헤드감소 방법

1. 단순 부호확장(그림 3 참조)을 모든 N 개의 부분곱에 적용한다. (N = 부분 곱의 개수)
2. 다음 과정을 i = 0 ~ N 동안 N+1 회 반복한다.
   - 마지막 i 개의 부분곱에만 compensation 벡터 방법을 적용하고 처음 N-i개의 부분곱은 변화 시키지 않는다.
   - 새로운 N 개의 부분곱에 전파캐리선택 절차를 적용한다.
   - 구현비용을 계산한다.
3. 구현비용이 가장 적은 구현을 선택한다.

예제 1: 다음 부분곱들의 덧셈을 고려하자.

$$\begin{array}{ccc} x_4 & x_3 & x_2 \\ \overline{x_6} & \overline{x_5} & \overline{x_4} \\ \overline{x_9} & \overline{x_9} & \overline{x_9} \\ x_9 & x_9 & x_9 \\ & & c_{in} \end{array}$$

위의 부분곱들은 그림3의 컬럼 5~7에 해당하며 c_{in}은 그 전 컬럼으로부터 전파되어온 캐리를 나타낸다. 전파캐리선택 절차는 마지막 컬럼부터 시작하며, 마지막 컬럼에서 전파캐리의 후보는 x_2와 $\overline{x_4}$이다. (c_{in}은 미리 결정할 수 있는 값이 아니므로 전파캐리의 후보에서 제외한다.) x_2나 $\overline{x_2}$는 마지막 컬럼의 왼쪽에 있는 컬럼들에 나타나지 않는 반면, x_4는 컬럼 0에 나타나므로 $\overline{x_4}$를 전파 캐리로 선택한다.

같은 방법으로 컬럼 1에 대해서는 $\overline{x_4}$(컬럼 2로부터 전파된 캐리)를 전파캐리로 선택한다. 마지막으로 컬럼 0을 간략화 시키면 다음과 같은 단순화된 부분곱들을 얻는다.

$$\begin{array}{cccc} 1 & \overline{x_6} & x_3 & x_2 \\ & & \overline{x_5} & x_4 \\ & & x_4 & c_{in} \end{array}$$

그림 4(a)는 위의 부분곱들에 해당하는 회로구현의 예이다.

전파캐리선택 절차를 적용하지 않고 compensation 벡터방법을 적용하여 간략화 시키면 다음과 같은 부분곱을 얻는다 (그림 1의 컬럼 5~7 참조).

$$\begin{array}{ccc} x_4 & x_3 & x_2 \\ \overline{x_6} & \overline{x_5} & \overline{x_4} \\ 1 & 1 & x_9 \\ & & c_{in} \end{array}$$

효율적인 구현을 위해서 식 (3)을 사용하면 그림 4(b)와 같이 구현된다. 그림 4로부터 제안한 방법을 이용하여 3개의 HA (1FA = 2HA)를 줄일 수 있음을 알 수 있다. □

Copyright (C) 2003 NuriMedia Co., Ltd.

www.dbpia.co.kr

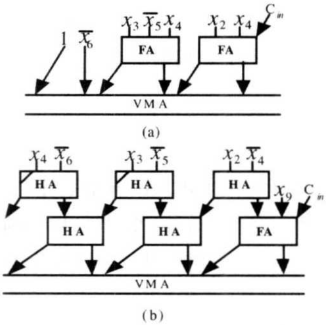

그림 4. 예제 1의 구현: (a)전파캐리선택 절차. (b) compensation 벡터.

예제 2: 제안한 방법에 의한 식 (2)의 구현을 고려하자.

먼저 단순 부호확장을 식 (2)에 적용하면 그림 3과 같은 부분곱을 얻고, 이 부분곱에 전파캐리선택 절차를 적용한 후 그 결과를 구현하면 그림 5와 같다. 다음, 그림 6과 같이 마지막 부분곱에만 compensation 벡터방법을 적용하고 전파캐리선택 절차를 적용한 후 구현비용을 계산한다. 다음에는 마지막 두 개의 부분곱에만 compensation 벡터방법을 적용하고 전파캐리선택 절차를 적용한 후 구현비용을 계산한다. 이러한 절차를 모든 4개의 부분곱이 compensation 벡터형태로 변환 될 때까지 반복한다. 마지막으로, 구현 비용이 가장 적은 구현을 선택한다.

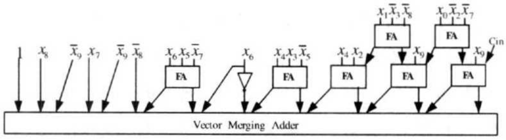

그림 5. 전파캐리 선택절차에 의한 그림 3의 처음 10 컬럼의 구현.

⓪ ① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨

$$x_9x_8x_7x_6x_5x_4x_3x_2x_1x_0$$

$$\bar{x}_9\bar{x}_9\bar{x}_9\bar{x}_8\bar{x}_7\bar{x}_6\bar{x}_5\bar{x}_4\bar{x}_3\bar{x}_2\bar{x}_1\overset{1}{\bar{x}_0}$$

$$\bar{x}_9\bar{x}_9\bar{x}_9\bar{x}_9\bar{x}_9\bar{x}_9\bar{x}_9\bar{x}_9\bar{x}_8\bar{x}_7\bar{x}_6\bar{x}_5\bar{x}_4\bar{x}_3\bar{x}_2\bar{x}_1\overset{1}{\bar{x}_0}$$

$$1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ \underset{1}{1}\ \bar{x}_9x_8x_7x_6x_5x_4x_3x_2x_1x_0$$

그림 6. 마지막 부분곱에만 compensation 벡터방법 적용 (i = 1).

본 예제에서는 그림 5의 구현이 가장 적은 비용을 차지함을 알 수 있으며 제안하는 방법에 의한 구조로 선택된다. 그림 5와 그림 2를 비교함으로써 제안하는 방법에 의해 8개의 HA를 절약할 수 있음을 알 수 있다. □

Ⅲ. 고정 길이 곱셈기

2의 보수로 표현된 W 비트의 두 수 X와 Y를 다음과 같이 정의하자.

$$X = -x_{W-1} + \sum_{i=1}^{W-1} x_{W-1-i}2^{-i}$$
$$Y = -y_{W-1} + \sum_{i=1}^{W-1} y_{W-1-i}2^{-i} \tag{4}$$

X와 Y의 이상적인 곱 P_I는 식 (5)와 같이 표현된다.

$$P_I = S_MP + S_LP \tag{5}$$

여기서 S_MP와 S_LP는 그림3에 보인바와 같은 MP와 LP부분의 부분곱들의 합이며 다음과 같다.

$$S_MP = -p_{2W-2} + \sum_{i=1}^{W-1} p_{2W-2-i}2^{-i}$$
$$S_LP = \sum_{i=W}^{2W-2} p_{2W-2-i}2^{-i} \tag{6}$$

일반적인 고정길이 곱셈기에서는 S_LP항을 생략하여 전체 adder cell의 반만 사용하고 확률적 추정에 근거한 적절한 바이어스를 더해줌으로써 오차를 보상한다. [4]와 [5]에서는 Baugh-Wooley 곱셈기에 대하여 적절한 σ의 계산방법을 제시했으며 본 논문에서는 CSD 곱셈기에 적합한 σ의 계산법을 제시한다.

식 (5)에서 S_LP항을 생략하고 바이어스를 더한 양자화된 곱 P_Q는 다음과 같이 표현된다.

$$P_Q = S_MP + \sigma \times 2^{-(W-1)} \tag{7}$$

σ를 LP로부터 MP에 전파되는 캐리의 추정치로 생각할 때 다음과 같이 표현할 수 있다.

$$\sigma = [LP'/2]_r \tag{8}$$

식 (8)에서 $LP' = LP \times 2^W$이며 $[t]_r$은 t에 대한 반올림 연산이다. W=10일 때 그림 3으로부터 $[LP'/2]_r$은 다음과 같이 나타낼 수 있다.

Copyright (C) 2003 NuriMedia Co., Ltd.

www.dbpia.co.kr

$$[LP'/2]_r = [2^{-1}(\overline{x_1}+\overline{x_6}+x_8)+2^{-2}(1+\overline{x_0}+\overline{x_5}+x_7)+2^{-3}(\overline{x_4}+x_6)+\cdots+2^{-9}x_0]_r \quad (9)$$

식 (9)를 직접 구현하여 σ를 계산하는 것은 효율적이지 못하므로 식 (9)의 근사화가 필요하다. 이를 위해 식 (9)에서 weight가 가장 큰 항들과 그 외의 항들을 구분하여 E_{main}과 E_{remain}을 다음과 같이 정의한다(그림 3참조).

$$\begin{aligned} E_{main} &= \overline{x_1}+\overline{x_6}+x_8 \\ E_{remain} &= 2^{-1}(1+\overline{x_0}+\overline{x_5}+x_7) \\ &\quad +2^{-2}(\overline{x_4}+x_6)+\cdots+2^{-8}x_0 \end{aligned} \quad (10)$$

식 (10)를 이용하여 σ를 표현하면 다음과 같다.

$$\sigma = \left[\frac{1}{2}(E_{main}+E_{remain})\right]_r \quad (11)$$

임의의 정수 θ_{index}를 사용하여 식 (11)은 다음과 같이 표현될 수 있다.

$$\begin{aligned} \sigma &= \theta_{\text{index}} + \left[\frac{1}{2}(E_{main}+E_{remain})-\theta_{\text{index}}\right]_r \\ &= \theta_{\text{index}} + [K]_r \end{aligned} \quad (12)$$

본 논문에서는 식 (12)의 계산을 용이하게 하고 에러를 최소화하기 위하여 θ_{index}와 K를 다음과 같이 정의한다.

$$\begin{aligned} \theta_{\text{index}} &= E_{main} \\ K &= E\left[\frac{1}{2}(E_{remain}-E_{main})\right] \end{aligned} \quad (13)$$

식 (13)에서 $E[t]$는 t의 기대치를 구하는 연산이다. 따라서 식 (12)의 근사치 $\sigma_{proposed}$는 다음과 같이 표현된다.

$$\sigma_{prop.} = E_{main} + \left[\frac{1}{2}E[E_{remain}-E_{main}]\right]_r \quad (14)$$

예제 3: 그림 3에서 W=10인 고정길이 곱셈기 구현 시

$$E_{main} = \overline{x_1}+\overline{x_6}+x_8 \quad (15)$$

이다. 입력 비트 x_i가 균일 분포를 갖는다고 가정할 때

$$\begin{aligned} \frac{1}{2}E[E_{remain}-E_{main}] &= 0.8779-0.75 \\ &= 0.1279 \end{aligned} \quad (16)$$

이며 0.1279의 반올림 값은 0이므로 오차보상 바이어스는 다음과 같이 결정된다.

$$\sigma_{prop.} = \overline{x_1}+\overline{x_6}+x_8 \quad (17)$$

따라서 LP는 FA 1개만으로 구현된다.

제안한 $\sigma_{prop.}$의 성능을 평가하기 위해 절대에러(absolute error) ε을 다음과 같이 정의하자.

$$\varepsilon = |P_I - P_Q| \quad (18)$$

가능한 모든 1024개의 10비트 입력에 대해서 제안한 $\sigma_{prop.}$를 사용했을 때 ε의 평균 및 분산은 각각 8.419×10^{-4}과 3.584×10^{-7}이다. 근사화 없이 *S_MP*와 *S_LP*를 계산하고 그 출력 중 10비트를 취하였을 때 ε의 평균 및 분산은 각각 9.746×10^{-4}과 3.187×10^{-7} 이다. 이상으로부터 제안한 방법은 이상적인 곱셈기에서 출력을 취했을 때와 거의 동일한 오차특성을 얻으면서 LP에 해당하는 adder cell을 단 하나의 FA로 줄일 수 있음을 알 수 있다. □

예제 4: 본 예제에서는 표 2의 4가지 경우에 대한 부호확장 오버헤드감소 및 고정길이 곱셈기의 적용을 고려한다. 곱셈기의 계수는 CSD 형태이고 입력은 2의 보수라고 가정한다. 경우 i과 ii에서는 0이 아닌 디지트를 3개 갖는 모든 11디지트 CSD 계수를 고려하고, 경우 iii과 iv에서는 0이 아닌 디지트를 4개 갖는 모든 10디지트 CSD 계수에 대해 시뮬레이션을 수행한다.

부호확장 부분을 최소화하기 위해 각 경우에 대해 전파캐리선택 방법과 compensation 벡터방법을 적용한 후 각 곱셈의 LP부분을 제안한 고정길이 곱셈기 디자인 방법을 적용하여 단순화시킨다. 그림 7~10은 각 경우에 대하여 compensation 벡터방법 적용 시와 전파캐리선택 방법 적용시의 전가산기의 차이를 보여준다. 시뮬레이션 결과로부터 제안한 방법에 의해 하나의 곱셈 당 최고 5개까지의 FA를 절약할 수 있음을 알 수 있다. 전체적으로 제안한 방법에 의해 부호확장에 의한 오버헤드를 약 30%

표 2. 예제4의 4가지 경우.

	CSD 계수의 0이 아닌 디지트 수	CSD 계수의 워드 길이	입력의 워드 길이
i	3	11	11
ii	3	11	13
iii	4	10	10
iv	4	10	11

Copyright (C) 2003 NuriMedia Co., Ltd.
www.dbpia.co.kr

도 줄일 수 있다. □

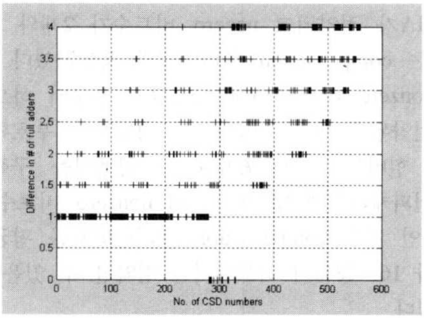

그림 7. 경우 i의 시뮬레이션 결과.

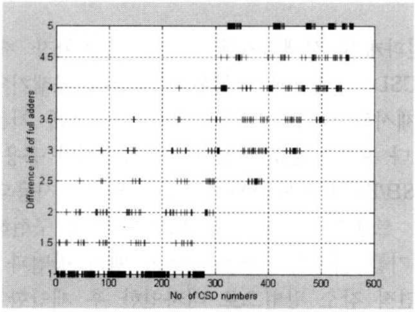

그림 8. 경우 ii의 시뮬레이션 결과.

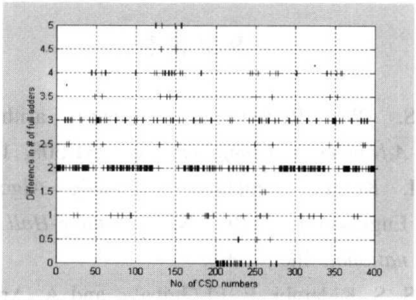

그림 9. 경우 iii의 시뮬레이션 결과.

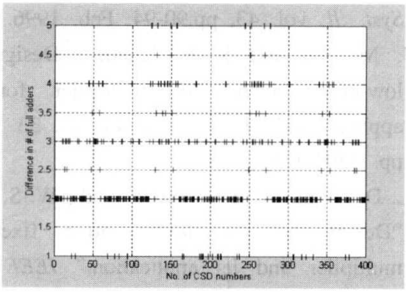

그림 10. 경우 iv의 시뮬레이션 결과.

Ⅳ. 힐버트 트랜스포머 구현

본 절에서는 II절 및 III절의 방법을 결합하여 SSB/BPSK- DS/CDMA용 힐버트 트랜스포머를 43 탭으로 디자인한다. 구현하는 힐버트 트랜스포머는 입력과 출력이 각각 10비트이고 정규화된 주파수 대역에서 통과대역은 0.03~0.97이며, 저지대역은 0~0.03, 0.97~1이다. 통과대역 리플은 최고 2dB를 넘지 않으며 저지대역 감쇠는 70dB이다.

필터계수가 anti-symmetric한 성질을 가지며 짝수 필터계수는 0이라는 성질을 가지므로 그림 11과 같은 folded direct form transpose 구조를 사용할 수 있다. 이 구조는 짧은 critical path (data broadcast delay +1 multiplier + 1 adder + 1 register setup time)를 가지는 장점이 있다.

그림 12~13에서 (a)는 제안하는 전파캐리선택 방법으로 구현한 경우이고 (b)는 compensation 벡터 방법을 사용하여 구현한 예이다.

여기서 $h_0 = 0.101000101$, $h_1 = 0.0100\bar{1}0\bar{1}0\bar{1}$ 이다. 제안한 방법을 적용하면 h_0의 경우

$E_{main} = x_0 + x_2 + x_6 + x_8$,

$E_{remain} = 2^{-1}(x_1 + x_5 + x_7) + \cdot \cdot \cdot + 2^{-8}x_0$ 이다.

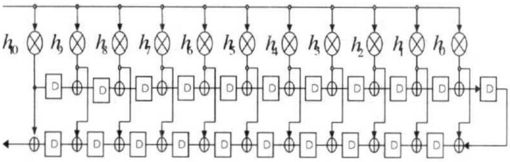

그림 11. Folded direct form transpose 구조.

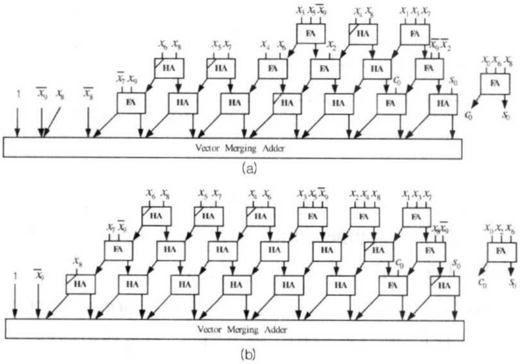

그림 12. 곱셈기 h_0의 구현: (a) 전파캐리선택 방법 사용. (b) compensation 벡터 사용.

Copyright (C) 2003 NuriMedia Co., Ltd.
www.dbpia.co.kr

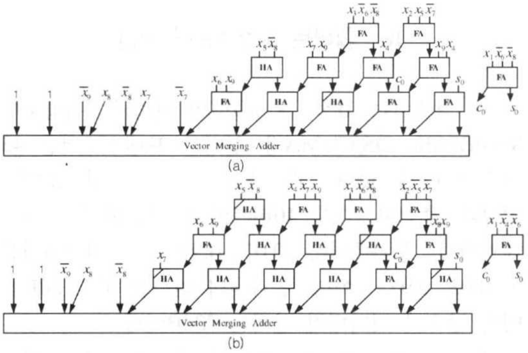

그림 13. 곱셈기 h_1의 구현: (a) 전파캐리선택 방법 사용. (b) compensation 벡터 사용.

또한 $\frac{1}{2}E[E_{remain}-E_{main}] = -0.3174$이므로 반올림한 값은 0이 되어 $\sigma_{prop.}=x_0+x_2+x_6+x_8$ 이다. h_1의 경우 $E_{main}=x_1+\overline{x_4}+\overline{x_6}+\overline{x_8}$, $E_{remain}=2^{-1}(x_0+\overline{x_3}+\overline{x_5}+\overline{x_7})+\cdot\cdot\cdot+2^{-8}x_0$이다. 또한 $\frac{1}{2}E[E_{remain}-E_{main}] = -0.1045$이므로 반올림한 값은 0이 되어 $\sigma_{prop.}=x_1+\overline{x_4}+\overline{x_6}+\overline{x_8}$ 이다.

표 3. 각 곱셈기 H/W(HA기준)

nonzero 비트수	곱셈기	com.벡터 방법	전파캐리 선택절차	HA 차이
4	h_0	29	25	4
	h_1	24	20	4
	h_3	24	20	4
3	h_5	12	9	3
	h_6	10	7	3
	h_{10}	10	7	3
2	h_2	4	4	0
	h_7	4	4	0
	h_8	5	5	0
	h_9	5	5	0
1	h_4	0	0	0
합계		127	106	21

그림 12~13으로부터 전파캐리선택 방법을 사용한 곱셈기가 compensation 벡터방법을 사용한 곱셈기에 비해 효율적임을 알 수 있다.

필터를 구성하는 11개의 곱셈기를 nonzero 비트수에 따라 표 3과 같이 구분하고 전파캐리선택 방법과 compensation 벡터방법을 적용하여 구현하였을 때 하드웨어 소모정도를 HA기준으로 표 3에 보였다.

Nonzero 비트수가 3과 4인 경우 각각 3개와 4개의 HA가 절약되고 nozero 비트수가 2개와 1개인 경우는 compensation 벡터방법과 동일하였다. 따라서 nonzero 비트가 3개 이상일 경우부터 하드웨어가 절약됨을 알 수 있다. 두 방법으로 부호확장을 하여 힐버트 트랜스포머의 각 곱셈기를 구현했을 때 전파캐리선택 방법에 의해 nonzero 비트수가 4와 3인 경우 compensation 벡터방법보다 평균적으로 약 16~28%의 하드웨어를 절약할 수 있음을 알 수 있다.

V. 결 론

전파캐리 선택절차를 이용한 부호확장 제거방법 및 CSD 곱셈기에 적합한 고정길이 곱셈기의 구조를 제시하고 다양한 시뮬레이션을 통해 기존의 방법보다 효율적임을 보였다. 이 곱셈기의 응용으로서 SSB/BPSK-DS/CDMA 전송방식에 사용되는 힐버트 트랜스포머를 43탭 FIR 필터로 구현하고 각 곱셈기를 기존의 compensation 벡터 방법과 부호확장 면적 감소 방법으로 디자인한 후 제안한 방법에 의해 nonzero 비트수가 4와 3인 경우 약 16~28%의 부호확장 오버헤드를 줄일 수 있음을 보였다.

참 고 문 헌

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

[2] I. Koren, Computer Arithmetic Algorithms. Englewood Cliffs, NJ: *Prentice-Hall International*, Inc., 1993.

[3] S. S. Kidambi, F. El-Guibaly, and A. Antoniou, "Area efficient multipliers for digital signal processing application." *IEEE Trans. Circuits Syst. II*, vol. 43, pp.90-94, Feb. 1996.

[4] J. M. Jou and S. R. Kuang, "Design of a low-error fixed-width multiplier for DSP applications.", *Electron. Lett.*, vol. 33. no. 19, pp. 1597-1598, 1997.

[5] L.-D. Van, S.-S. Wang, and W.-S. Feng, "Design of the lower error fixed-width multiplier and its application." *IEEE Trans. Circuits Synst. II*, vol. 47, pp. 1112-1118, Oct. 2000.

Copyright (C) 2003 NuriMedia Co., Ltd.

www.dbpia.co.kr

조 경 주(Kyung-Ju Cho) 정회원

2000년 2월 : 원광대학교 전자공학과(공학사)
2002년 2월 : 전북대학교 정보통신학과(공학석사)
현재 : 전북대학교 대학원 정보통신공학과 박사과정
<주관심 분야> OFDM, VLSI 신호처리

김 명 순(Myung-Soon Kim) 정회원

제 25권 제 10A호 참조
1999년 2월 : 전북대학교 정보통신공학과(공학사)
2001년 2월 : 전북대학교 정보통신공학과(공학석사)
현재 : 한국전자통신연구원 무선방송연구소 무선 LAN팀
<주관심 분야> OFDM, VLSI 신호처리

유 경 주(Kyung-Ju Yoo) 정회원

제 25권 제 3B호 참조
1999년 2월 : 전북대학교 정보통신공학과(공학사)
2001년 2월 : 전북대학교 정보통신공학과(공학석사)
현재 : (주)로직메카부설멀티미디어통신연구소
<주관심 분야> VLSI 신호처리, CDMA

정 진 균(Jin-Gyun Chung) 정회원

제 25권 10A호 참조
현재 : 전북대학교 전자정보공학부 부교수
<주관심 분야> VLSI 신호처리, 고속 DSL 모뎀

Copyright (C) 2003 NuriMedia Co., Ltd.
www.dbpia.co.kr