

고정길이 Booth-Folding 제곱기 디자인

정회원 조 경 주*, 정 진 균*

Fixed-Width Booth-Folding Squarer Design

Kyung-Ju Cho*, Jin-Gyun Chung* *Regular Members*

요 약

본 논문은 W 비트 입력으로부터 W 비트를 출력하는 고정길이 제곱기의 디자인 방법을 제안한다. 효율적으로 양자화 오차를 보상하기 위해 modified Booth 인코더의 출력정보를 이용하여 오차보상 바이어스를 생성한다. 절사(truncation)된 부분이 양자화 오차에 미치는 영향에 따라 두 그룹(major/minor group)으로 나누고, 각 그룹에 서로 다른 오차보상 방법을 적용하여 절사된 부분을 보상한다. 시뮬레이션을 통해 제안한 오차보상 방법이 기존의 방법에 비해 절대 양자화 최대오차, 평균오차, 분산과 같은 성능 비교 파라미터에서 각각 30%, 24%, 43% 정도 적음을 보인다. 또한, 제안한 고정길이 제곱기는 이상적인 제곱기보다 면적과 전력소모 면에서 각각 약 28%, 27% 까지 적음을 보인다.

Key Words : Squarer, fixed-width, Booth-folding, quantization error.

ABSTRACT

This paper presents a design method for fixed-width squarer that receives a W -bit input and produces a W -bit squared product. To efficiently compensate for the quantization error, modified Booth encoder signals (not multiplier coefficients) are used for the generation of error compensation bias. The truncated bits are divided into two groups (major/minor group) depending upon their effects on the quantization error. Then, different error compensation methods are applied to each group. By simulations, it is shown that the performance of the proposed method is close to that of the rounding method and much better than that of the truncation method and conventional method. It is also shown that the proposed method leads to up to 28% and 27% reduction in area and power consumption compared with the ideal squarers, respectively.

I. 서 론

제곱 연산은 일반적인 곱셈 연산과 달리 승수와 피승수가 같은 곱셈으로 벡터 양자화, 에러 정정, 이미지 압축, 패턴 인식, OFDM 시스템에서 FEQ (Frequency Domain Equalizer)와 같은 많은 DSP 응용에 자주 사용된다. 따라서 다양한 제곱기의 구조가 연구되었다^[1-3].

Divide-and-conquer 기법^[1]과 folding과 부분곱 비

트의 재배열^[2]을 이용하여 부분곱의 최고 높이를 줄이는 방법이 제안되었다. 특히, 제곱 연산에 Booth 인코딩 기법을 도입하여 기존 구조보다 부분곱의 수를 약 50% 정도 줄인 효율적인 구조가 제안되었다^[3].

많은 멀티미디어와 DSP 응용에서 입력과 출력의 데이터 워드길이(W)가 같은 고정길이 곱셈기가 요구된다. 예를 들어, 승수와 피승수가 각각 W 비트 일 때 $(2W-1)$ 비트의 곱셈 출력을 얻는데, 하위 LSB

* 전북대학교 전자정보공학부 정보통신공학과 ({kjcho, jgchung}@chonbuk.ac.kr)

논문번호 : KICS2005-06-246, 접수일자 : 2005년 6월 17일

※ 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 육성·지원 사업의 연구결과로 수행되었음

(Least Significant Bit)의 (W-1) 비트를 절사(truncation)한 후 W 비트로 양자화 한다.

일반적인 고정길이 곱셈기 디자인 방법은 하위 LSB의 (W-1)비트에 해당하는 adder cell을 생략하고 확률적인 추정을 바탕으로 생성된 오차보상 바이어스를 더해줌으로써 양자화 오차를 보상한다. 기존 연구에서 Baugh-Wooley, Modified Booth, CSD (Canonic Signed Digit) 곱셈기의 각 구조에 적합한 다양한 고정길이 곱셈기 디자인 방법이 제안되었다^{[4]-[7]}. 하지만, 고정길이 제곱기에 관한 연구는 거의 없다. 따라서 제곱기의 구조에 적합한 고정길이 오차보상 방법에 관한 연구가 요구된다.

본 논문에서는 참고문헌 [8]을 바탕으로 Booth-folding 제곱기의 구조를 기반으로 하는 고정길이 제곱기 디자인 방법을 제안한다.

본 논문은 다음과 같이 구성된다. II 장에서 Booth-Folding 기법을 이용한 제곱기의 구조를 소개하고, III 장에서 제안하는 고정길이 제곱기의 구조와 오차보상 방법을 설명한다. IV 장에서 근사캐리 생성 절차를 제안하고, V 장에서 다양한 시뮬레이션을 통해 제안한 방법의 성능을 분석한다. 마지막으로 VI 장에서 간단히 결론을 맺는다.

II. Booth-Folding을 이용한 제곱기

2의 보수로 표현되고 워드길이가 W 비트인 X를 고려하자.

$$X = -x_{W-1} + \sum_{i=1}^{W-1} x_{W-1-i} 2^{-i} \quad (1)$$

곱셈기 디자인 시 부분곱의 수를 줄이기 위해 modified Booth 알고리즘이 자주 사용된다. 식 (1)의 X를 modified Booth 인코딩하면 다음과 같이 표현할 수 있다.

$$X = \sum_{i=0}^{W/2-1} x'_i 2^{-(2i+1)} \quad (2)$$

여기서, $x'_i = -2x_{2i+1} + x_{2i} + x_{2i-1}$ 이다. 표 1은 Modified Booth 인코딩 표이다. 예를 들어, W=10인 X는 다음과 같이 표현된다.

$$X = x_9 \cdot x_8 x_7 x_6 x_5 x_4 x_3 x_2 x_1 x_0 \\ = x_4' 2^{-1} + x_3' 2^{-3} + x_2' 2^{-5} + x_1' 2^{-7} + x_0' 2^{-9} \quad (3)$$

X의 제곱은 그림 1의 중간 부분처럼 modified Booth

표 1. Modified Booth 인코딩 표

x_{2i+1}	x_{2i}	x_{2i-1}	x'_i	$X_{sel,i}$	$2X_{sel,i}$	NEG_i
0	0	0	0	0	0	0
0	0	1	1	1	0	0
0	1	0	1	1	0	0
0	1	1	2	0	1	0
1	0	0	-2	0	1	1
1	0	1	-1	1	0	1
1	1	0	-1	1	0	1
1	1	1	0	0	0	0

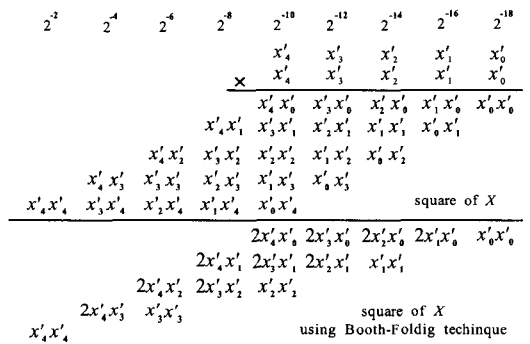


그림 1. W=10인 경우, Booth-Folding에 의한 제곱연산

인코딩의 곱으로 계산할 수 있다. 또한, folding 기법을 이용하면 그림 1의 아래 부분과 같이 부분곱의 수를 약 반으로 줄일 수 있다.

참고문헌 [3]에 따르면 W=10인 X의 제곱은 다음과 같이 표현할 수 있다.

$$X^2 = C_4 2^{-2} + (P_3 2^3 + C_3) 2^{-6} + (P_2 2^3 + C_2) 2^{-10} \\ + (P_1 2^3 + C_1) 2^{-14} + (P_0 2^3 + C_0) 2^{-18} \quad (4)$$

여기서, C_i, P_i, Q_i 는 다음과 같다.

$$C_i = x'_i \times x'_i \text{ for } i = 0, \dots, 4 \\ P_i = Q_i \times x'_i + x_{2i+1} \times x'_i \\ Q_i = (q_{i,(7-2i)} \cdot q_{i,(6-2i)} \cdots q_{i,0}) 2^{(7-2i)} \\ = (x_9 \cdot x_8 \cdots x_{2i+2}) 2^{(7-2i)} \text{ for } i = 0, \dots, 3 \quad (5)$$

X의 제곱을 계산하기 위해 더해져야 할 모든 부분곱(P_i, C_i 항)들의 배열을 그림 2(a)에 나타내었다. Booth 인코딩된 디지털 x' 는 다섯 개의 값 {0, ±1, ±2} 중에서 하나를 가지므로 그림 2(a)에서 C_i 항들은 세 개의 값 {0, 1, 4} 중에서 하나가 된다. 따

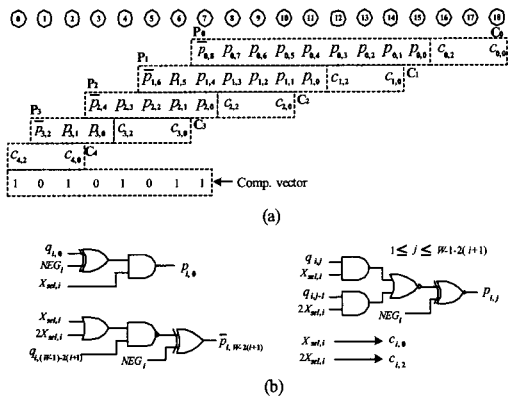


그림 2. W=10인 경우, Booth-folding 제곱기의 부분곱 (a) 부분곱 배열, (b) 부분곱 비트 생성회로

라서 \$c_{i,1}\$은 항상 0 이다. 그림 2(a)에서 상수 백터 (10101011)은 부호확장을 제거하기 위한 보상벡터 (compensation vector)이다. 그림 2(b)는 각 부분곱의 비트들을 생성하는 회로이다.

III. 고정길이 제곱기의 구조와 오차보상 방법

3.1 고정길이 제곱기의 구조

\$W=10\$인 제곱기의 부분곱 배열을 그림 3과 같이 MP(Most Significant Part)와 LP(Least Significant Part) 부분으로 나눈다. 보다 효율적인 오차보상 바이어스를 생성하기 위해 LP 부분을 \$LP_{major}\$와 \$LP_{minor}\$로 나눈다. \$(2W-1)\$비트의 이상적인 곱 \$P_I\$는 다음과 같이 표현할 수 있다.

$$P_I = S_MP + S_LP \quad (6)$$

여기서, \$S_MP\$와 \$S_LP\$는 각각 MP와 LP에 포함된 부분곱 비트들의 합이다.

일반적인 고정길이 곱셈기는 \$S_LP\$의 계산을 위해 필요한 adder cell을 생략하고, 확실적인 추정에 근거하여 생략된 adder cell을 대신할 적절한 바이어스를 \$S_MP\$에 더해준다. 따라서 \$W\$ 비트로 양자화된 곱 \$P_Q\$는 다음과 같이 표현할 수 있다.

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

여기서 \$\sigma\$는 오차보상 바이어스로서 LP로부터 MP로 전파되는 근사캐리(approximate carry) 신호이다.

3.2 제안하는 오차보상 방법

그림 3에서 \$S_LP\$는 다음과 같이 표현할 수 있다.

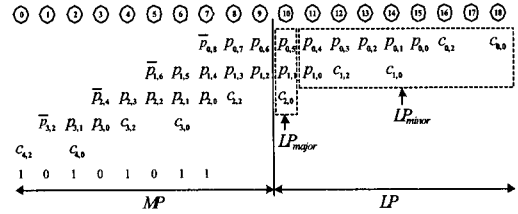


그림 3. W=10인 경우, 제곱기의 MP와 LP

$$S_LP = S_LP_{major} + S_LP_{minor} \quad (8)$$

여기서, \$S_LP_{major}\$와 \$S_LP_{minor}\$는 다음과 같다.

$$\begin{aligned} S_LP_{major} &= p_{0,5} + p_{1,1} + c_{2,0} \\ S_LP_{minor} &= 2^{-1}(p_{0,4} + p_{1,0}) + 2^{-2}(p_{0,3} + c_{1,2}) \\ &\quad + 2^{-3}(p_{0,2}) + 2^{-4}(p_{0,1} + c_{1,0}) \\ &\quad + 2^{-5}(p_{0,0}) + 2^{-6}(c_{0,2}) + 2^{-8}(c_{0,0}) \end{aligned} \quad (9)$$

\$LP_{major}\$는 LP에서 가장 큰 weight를 가지므로 \$LP_{minor}\$에 비해 LP에서 생성되는 캐리 신호에 훨씬 큰 영향을 준다. 이 점을 고려하여 본 논문에서는 그림 4와 같은 고정길이 제곱기의 구조를 제안하며 오차보상 바이어스는 다음과 같이 정의된다.

$$\sigma_{prop} = C_E[S_LP_{major} + C_A[S_LP_{minor}] + 1] \quad (10)$$

여기서, \$C_E[t]\$와 \$C_A[t]\$는 각각 \$t\$에 대한 정확한 캐리 값과 근사캐리 값을 의미한다. 또한, 상수 1의 덧셈은 \$C_E[S_LP_{major} + C_A[S_LP_{minor}]]\$의 연산 후 발생한 합 신호를 반올림하기 위한 것이다. 식 (10)에서 \$C_A[S_LP_{minor}]\$는 \$LP_{minor}\$에서 \$LP_{major}\$로 전파되는 근사 캐리 값이다.

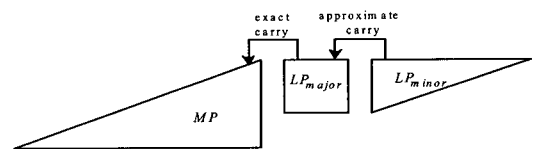


그림 4. 제안하는 고정길이 제곱기의 구조

IV. 근사 캐리 생성

Booth 인코더 출력에 의한 새로운 파라미터 \$x''_i\$를 다음과 같이 정의하자.

$$x''_i = \begin{cases} 1, & \text{if } x'_i \neq 0 \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

표 1로부터 x''_i 는 다음과 같이 간단히 계산된다.

$$x''_i = X_{sel,i} \vee 2X_{sel,i} \quad (12)$$

여기서, \vee 는 OR 연산자이다.

$x''_i=1$ 이면, 표 1로부터 $E[c_{i,0}]=2/3$, $E[c_{i,2}]=1/3$ 이 됨을 보일 수 있으며 $E[p_{i,j}]=1/2$ 이다. 여기서 $E[t]$ 는 t 에 대한 기대치를 의미한다.

예제 1: $W=10$ 인 경우, $E[S_{LP_{minor}}]$ 의 값을 고려하자. 그림 3에서 볼 수 있듯이 x'_4, x'_3, x'_2 에 의해 생성되는 부분곱 중 LP_{minor} 에 포함되는 원소는 없기 때문에 x'_4, x'_3, x'_2 는 $E[S_{LP_{minor}}]$ 에 전혀 영향을 주지 않는다. 식 (9)를 이용하여 $x''_1x''_0=01$, $x''_1x''_0=10$ 인 경우에 대해 $E[S_{LP_{minor}}]$ 을 계산하면 표 2와 같다. 따라서 $W=10$ 인 경우의 $E[S_{LP_{minor}}]$ 는 다음과 같다.

$$E[S_{LP_{minor}}] = 2^{-1} \{ (1-2^{-2})x''_1 + (1-2^{-6})x''_0 \} \quad (13)$$

본 논문에서는 $E[S_{LP_{minor}}]$ 의 반올림 값을 근사 캐리로 정의한다. 식 (13)에서 $x''_1x''_0=11$ 일 때 $E[S_{LP_{minor}}]$ 의 반올림 값은 1이고, 그 외의 경우는 0이다. 따라서 $E[S_{LP_{minor}}]$ 의 반올림 값을 표현하기 위해 1개의 신호가 필요하고 그 신호는 다음과 같다.

$$a_carry_0 = (x''_1 \wedge x''_0) \quad (14)$$

여기서, 연산자 \wedge 는 AND 연산이다. 그림 5에 식 (14)에 대한 구현과 전체 오차보상 바이어스 생성회

표 2. $W=10$ 인 경우 $E[S_{LP_{minor}}]$ 의 값

$x''_1 \ x''_0$	$E[S_{LP_{minor}}]$
0 1	$E[2^{-1}p_{0,4}+2^{-2}p_{0,3}+\dots+2^{-5}p_{0,0}+2^{-6}c_{0,2}+2^{-8}c_{0,0}] = 2^{-1}(1-2^{-6})$
1 0	$E[2^{-1}p_{1,0}+2^{-2}c_{1,2}+2^{-4}c_{1,0}] = 2^{-1}(1-2^{-2})$

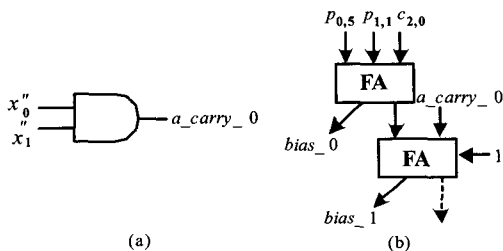


그림 5. $W=10$ 인 경우, 오차보상: (a) 근사캐리 신호, (b) 오차보상 바이어스 회로

를 나타내었다. 근사캐리 신호는 LP_{major} 에 더해지고 그 결과로 발생하는 캐리신호들을 MP 에 오차보상 바이어스로 더한다. □

표 3. $W=12$ 인 경우 $E[S_{LP_{minor}}]$ 의 값

$X_{sel,2} \ x''_1 \ x''_0$	$E[S_{LP'_{minor}}]$
0 0 1	$E[2^{-1}p_{0,6}+2^{-2}p_{0,5}+\dots+2^{-8}c_{0,2}+2^{-10}c_{0,0}] = 2^{-1}(1-2^{-8})$
0 1 0	$E[2^{-1}p_{1,4}+2^{-2}p_{1,3}+\dots+2^{-4}c_{1,2}+2^{-6}c_{1,0}] = 2^{-1}(1-2^{-4})$
1 0 0	$E[2^{-1}c_{2,0}] = 2^{-1}(1-2/3)$

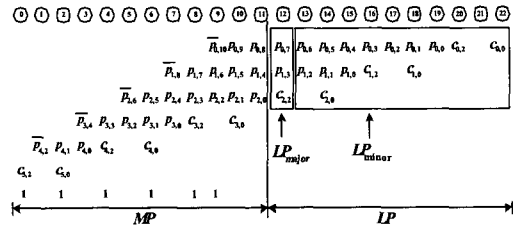


그림 6. $W=12$ 인 경우, 제곱기의 MP와 LP.

예제 2: $W=12$ 인 경우, $E[S_{LP_{minor}}]$ 를 고려하자. 그림 6은 $W=12$ 인 제곱기의 부분곱의 배열을 나타낸다. x'_5, x'_4, x'_3, x'_2 ($c_{2,0}$ 은 제외)는 $E[S_{LP_{minor}}]$ 에 전혀 영향을 주지 않는다.

그림 6으로부터 $S_{LP_{minor}}$ 는 다음과 같이 표현된다.

$$S_{LP_{minor}} = 2^{-1}(p_{0,6} + p_{1,2}) + 2^{-2}(p_{0,5} + p_{1,1} + c_{2,0}) + 2^{-3}(p_{0,4} + p_{1,0}) + 2^{-4}(p_{0,3} + c_{1,2}) + 2^{-5}(p_{0,2}) + 2^{-6}(p_{0,1} + c_{1,0}) + 2^{-7}(p_{0,0}) + 2^{-8}(c_{0,2}) + 2^{-10}(c_{0,0}) \quad (15)$$

식 (15)를 이용하여 $X_{sel,2}x''_1x''_0=001$, $X_{sel,2}x''_1x''_0=010$, $X_{sel,2}x''_1x''_0=100$ 인 경우에 대해 $E[S_{LP_{minor}}]$ 를 계산하면 표 3과 같다. 따라서 $W=12$ 인 경우의 $E[S_{LP_{minor}}]$ 는 다음과 같다.

$$E[S_{LP_{minor}}] = 2^{-1} \{ (1-2/3)X_{sel,2} + (1-2^{-4})x''_1 + (1-2^{-8})x''_0 \} \quad (16)$$

위의 두 예제의 결과를 확장하여 일반적인 워드

길이 W 에 대한 $E[S_{LP\ minor}]$ 는 다음과 같이 표현할 수 있다.

$$E[S_{LP\ minor}] = \begin{cases} 2^{-1} \sum_{k=0}^{m-1} (1 - 2^{-4(m-k)+2}) x''_k, & \text{for } W = 4m + 2 \\ 2^{-1} \left\{ (1 - 2^{-1}) X_{sel, (m-1)} \right. \\ \left. + \sum_{k=0}^{m-2} (1 - 2^{-4(m-1-k)}) x''_k \right\}, & \text{for } W = 4m \end{cases} \quad (17)$$

본 논문에서 제안하는 근사캐리 생성절차 (ACGP)는 다음과 같이 요약된다.

1. 주어진 $W=4m+j$ ($j=0, 2$)에 대해 근사캐리의 수(N_{AC})를 결정한다. 여기서, $N_{AC} = \lfloor W/8 \rfloor$ 이다.
2. 근사캐리 신호를 $a_carry_0, a_carry_1, \dots, a_carry_{(N_{AC}-1)}$ 로 정의한다.
3. 근사캐리 신호 a_carry_i 는 $x''_{(m-1)}, x''_{(m-2)}, \dots, x''_0$ 중에서 최소한 1이 $(2i+2)$ 이면 1로 결정된다. ($j=0$ 이면 $x''_{(m-1)}$ 대신 $X_{sel, (m-1)}$ 이 사용된다.)
4. Step 3의 결과로부터 카르노 맵을 이용하여 근사캐리 신호를 설계한다.

위의 절차를 예제 3을 통해 설명한다.

예제 3: $W=12$ 인 경우, 근사캐리 생성절차의 step 1에 의해 근사캐리의 수는 1이다. Step 3에 따라 $X_{sel,2}, x''_1, x''_0$ 중에서 두 개 이상 '1'의 값을 가지면 a_carry_0 신호는 1이 된다. 카르노 맵을 이용하면 다음과 같은 식을 구할 수 있다.

$$a_carry_0 = (x''_1 \wedge x''_0) \vee (X_{sel,2} \wedge (x''_1 \vee x''_0)) \quad (18)$$

□

V. 제안한 제공기의 성능 분석

$W=10$ 인 경우, 예제 1에서 디자인된 오차보상 바이어스를 고정길이 제공기에 적용한다. 정확한 실험을 위하여 모든 가능한 10 비트 입력에 대해 절대 양자화 오차 $\epsilon (= |P_I - P_O|)$ 를 계산한 후 최대오차, 평균오차, 오차의 분산을 구하였다. 표 4에 다양한 양자화 방법의 성능을 비교하였다. 표 4에서 Round와 Truncation은 $(2W-1)$ 의 이상적인 곱 P_I 로부터 W 비트로 각각 반올림과 절단하여 양자화한 방법을 나타내며, $\sigma_{[5]}$ 는 고정길이 modified Booth

곱셈기에 대한 참고문헌 [5]의 오차보상 방법을 고정길이 제공기에 적용한 방법이다. 워드길이 $W = 12, 14, 16$ 인 경우에 대해서 표 5~7에 다양한 양자화 방법의 성능을 비교하였다.

표 4~7로부터 제안한 방법의 $\sigma_{prop.}$ 이 절대 양자화 최대오차, 오차평균, 분산과 같은 성능비교 파라미터에서 $\sigma_{[5]}$ 의 방법보다 약 30%, 24%, 43% 적음을 알 수 있다.

제안한 방법과 참고문헌 [5] 방법의 면적과 전력 소모를 비교하기 위해 워드길이 $W = 10, 12, 16$ 인 경우의 고정길이 제공기를 VHDL로 구현하고, Synopsys의 class cell library를 사용하여 합성한 후 면적과

표 4. 다양한 양자화 방법의 평균과 분산 비교 ($W=10$)

	최대오차 ($\times 2^{-9}$)	평균 ($\times 2^{-9}$)	분산 ($\times 2^{-18}$)
Round	0.5	0.2422	0.0236
Truncation	0.9863	0.4406	0.0865
$\sigma_{[5]}$	1.3418	0.4349	0.1063
$\sigma_{prop.}$	0.9512	0.3171	0.0566

표 5. 다양한 양자화 방법의 평균과 분산 비교 ($W=12$)

	최대오차 ($\times 2^{-11}$)	평균 ($\times 2^{-11}$)	분산 ($\times 2^{-22}$)
Round	0.5	0.2461	0.0222
Truncation	0.9966	0.4695	0.0858
$\sigma_{[5]}$	1.2769	0.3548	0.0708
$\sigma_{prop.}$	0.9878	0.3115	0.0511

표 6. 다양한 양자화 방법의 평균과 분산 비교 ($W=14$)

	최대오차 ($\times 2^{-13}$)	평균 ($\times 2^{-13}$)	분산 ($\times 2^{-26}$)
Round	0.5	0.2480	0.0215
Truncation	0.9991	0.4846	0.0848
$\sigma_{[5]}$	1.7518	0.4839	0.1266
$\sigma_{prop.}$	0.9967	0.3163	0.0513

표 7. 다양한 양자화 방법의 평균과 분산 비교 ($W=16$)

	최대오차 ($\times 2^{-15}$)	평균 ($\times 2^{-15}$)	분산 ($\times 2^{-30}$)
Round	0.5	0.2490	0.0212
Truncation	0.9998	0.4922	0.0841
$\sigma_{[5]}$	1.6745	0.4107	0.0931
$\sigma_{prop.}$	1.2795	0.3303	0.0563

표 8. W=10, 12, 16인 경우, Synopsys 시뮬레이션 결과

	W=10		W=12		W=16	
	gate #	Power (μ W)	gate #	Power (μ W)	gate #	Power (μ W)
Ideal	458 (1)	0.696 (1)	633 (1)	1.018 (1)	1091 (1)	1.873 (1)
Fixed by $\sigma_{[s]}$	356 (0.78)	0.572 (0.82)	468 (0.74)	0.781 (0.77)	761 (0.70)	1.356 (0.72)
Fixed by $\sigma_{prop.}$	369 (0.80)	0.581 (0.83)	484 (0.76)	0.792 (0.78)	789 (0.72)	1.368 (0.73)

전력소모를 비교하였다. 표 8에서 Ideal은 (2W-1) 비트의 곱을 얻는 이상적인 곱셈기이다. 표 8로부터 고정길이 제곱기는 면적과 전력소모 면에서 이상적인 제곱기보다 각각 28%, 27% 까지 적음을 알 수 있다.

VI. 결론

Booth-folding 제곱기에 대한 고정길이 제곱기의 디자인 방법을 제안하였다. 효율적으로 양자화 오차를 보상하기 위해 절단된 부분을 major 그룹과 minor 그룹으로 나누고 각 그룹에 서로 다른 오차 보상 방법을 적용하였다. Minor 그룹에서는 modified Booth 인코더의 출력정보를 이용하여 근사캐리를 구하고, major 그룹에서는 정확한 캐리를 계산해서 절단된 부분을 보상하였다. 또한, 다양한 워드길이에 대해 근사캐리를 쉽게 구할 수 있도록 근사캐리 생성절차를 알고리즘화 하였다.

참고 문헌

[1] J. T. Yoo, K. F. Smith, and G. Gopalakrishnan, "A fast parallel squarer based on divider-and-conquer," *IEEE J. of Solid-State Circuits*, vol. 32, pp. 909-912, June 1997.

[2] R. K. Koagotla and W. R. Griesbach, "VLSI implementation of a 350 MHz 0.35um 8 bit merged squarer," *Electronic Letters*, vol. 34, pp. 47-48, Jan. 1998.

[3] A. G. M. Strollo and D. D. Caro, "Booth folding encoding for high performance squarer circuits," *IEEE Trans. CAS II*, vol. 50, pp. 250-254, May 2003.

[4] L. D. Van, S. S. Wang, and W. S. Feng, "Design of the lower error fixed-width mul-

tiplier and its application," *IEEE Trans. CAS II*, vol. 47, pp. 1112-1118, Oct. 2000.

[5] S. J. Jou, M. H. Tsai, and Y. L. Tsao, "Low-error reduced-width Booth multipliers for DSP applications," *IEEE Trans. CAS I*, vol. 50, pp. 1470-1474, Nov. 2003.

[6] S. M. Kim, J. G. Chung, and K. K. Parhi, "Low error fixed-width CSD multiplier with efficient sign extension," *IEEE Transactions on Circuits & Systems II*, vol. 50, pp. 984-993, Dec. 2003.

[7] K. J. Cho, K. C. Lee, J. G. Chung and K. K. Parhi, "Design of low error fixed-width modified Booth multiplier," *IEEE Trans. VLSI Systems*, vol. 12, pp. 522-531, May 2004.

[8] K. J. Cho, W. K. Kim, B. K. Kim, and J. G. Chung, "Design of low error fixed-width squarer," in *Proc. of 2003 IEEE SIPS*, (Seoul, Korea), pp. 213-218, Aug. 2003.

조 경 주 (Kyung-Ju Cho)

정회원



2000년 2월 원광대학교 전기공학부(공학사)

2002년 2월 전북대학교 정보통신공학(공학석사)

현재 전북대학교 정보통신공학과 박사과정

<관심분야> VLSI 신호처리,

OFDM, 고속 DSL 모델

정 진 균 (Jin-Gyun Chung)

정회원



1985년 2월 전북대학교 전자공학과(공학사)

1991년 12월 Univ. Minesota 전기공학과(공학석사)

1994년 12월 Univ. Minesota 전기공학과(공학박사)

1995년 3월~현재 전북대학교전

자정보공학부 부교수

1996년 9월~현재 전북대학교 정보통신 연구소 연구원
2003년 2월~2003년 11월 전북대학교 부설 전북 실리콘밸리 교육 센터장

2003년 11월~현재 전북대학교 IDEC 센터장

<관심분야> VLSI 신호처리, 고속 DSL 모델