Booth 인코더 출력을 이용한 저오차 고정길이 modified Booth 곱셈기 설계

정회원 조 경 주, 김 원 관, 정 진 균

Design of Low-error Fixed-width Modified Booth Multiplier Using Booth Encoder Outputs

Kyung-Ju Cho, Won-Kuan Kim, Jin-Gyun Chung *Regular Members*

요 약

본 논문은 워드길이가 W 비트인 입력으로부터 W 비트를 출력하는 고정길이 modified Booth 곱셈기에 대한 오차보상 방법을 설명한다. 효율적으로 양자화 오차를 보상하기 위해 Booth 인코더의 출력정보를 이용하여 오차보상 바이어스를 생성한다. 절단된 부분이 양자화 오차에 미치는 영향에 따라 두 그룹(major or minor group)으로 나누고, 각 그룹에 서로 다른 오차보상 방법을 적용한다. 기존 방법과 비교하여 제안한 방법이 오차보상 바이어스를 생성하는 회로의 하드웨어 오버헤드는 비슷하면서 약 50% 정도 양자화 오차가 적음을 시뮬레이션을 통해 보인다. 또한, 면적과 전력소모 면에서 제안한 고정길이 곱셈기가 이상적인 곱셈기 보다 약 40% 정도 적게 나타났다.

ABSTRACT

This paper presents an error compensation method for a fixed-width modified Booth multiplier that receives a W-bit input and produces a W-bit product. To efficiently compensate for the quantization error, Booth encoder outputs (*not multiplier coefficients*) are used for the generation of error compensation bias. The truncated bits are divided into two groups depending upon their effects on the quantization error. Then, different error compensation methods are applied to each group. By simulations, it is shown that quantization error can be reduced up to 50% by the proposed error compensation method compared with the existing method with approximately the same hardware overhead in the bias generation circuit. It is also shown that the proposed method leads to up to 40% reduction in area and power consumption of a multiplier compared with the ideal multiplier.

Key Words : modified Booth multiplier; quantization error; fixed-width multiplier.

Ⅰ. 서 론

많은 멀티미디어와 DSP 응용에서 입력과 출력의 데이터 길이가 같은 고정길이 곱셈기가 요구된다. 예를 들어, 승수와 피승수가 각각 W 비트일 때 $(2W-1)$비트의 곱셈 출력을 얻는데, 하위 LSB(Least Significant Bit)의 $(W-1)$비트를 생략하고 W 비트로 양자화 한다.

일반적인 고정길이 곱셈기의 디자인 방법은 하위 LSB의 $(W-1)$비트에 해당하는 adder cell(Least-significant Part에 해당)을 생략하고, 확률적인 추정에 근거하여 생성된 오차보상 바이어스를 남은 adder cell(Most-significant Part에 해당)에 더함으로써 양자화 오차를 보상한다.

Baugh-Wooley 곱셈기의 경우, 확률적 통계를 이용하거나 인덱스를 사용하여 오차보상 바이어스를 생성하는 디자인 방법이 제안되었다[1-4]. 참고문헌

* 전북대학교 전자정보공학부(kjcho, wkkim, jgchung@vlsidsp.chonbuk.ac.kr)
논문번호 : 030192-0509, 접수일자 : 2003년 5월 9일

[1]에서는 절단(truncation) 후 남은 adder cell(*MP*에 해당)에 고정 바이어스를 더하는 방법이 제안되었다. 이 방법의 오차 보상 바이어스를 생성하는 회로는 간단하지만 입력신호에 따라 적절히 변하는 바이어스 값이 아니므로 상당한 양자화 오차가 발생한다. 입력신호에 따른 영향을 고려하기 위해 인덱스를 사용한 바이어스 생성방법이 참고문헌 [2]와 [3]에서 제안되었다. 인덱스를 사용함으로써 양자화 오차를 줄일 수 있었지만, 두 가지 특별한 조건만을 고려하였기 때문에 여전히 한계를 지니고 있다. 참고문헌 [4]에서 통계적 분석과 선형귀환 분석(linear regression analysis)을 이용한 오차보상 방법을 제안하고, modified Booth 곱셈기에도 적용하였다.

고정길이 곱셈기에 대한 대부분의 연구는 Baugh-Wooley 곱셈기를 대상으로 하는 것이었고, 그 방법을 modified Booth 곱셈기에 적용하는 정도였다. 따라서 modified Booth 곱셈기의 구조에 적합한 오차보상 방법의 연구가 필요하다.

본 논문에서는 효율적인 고정길이 modified Booth 곱셈기 디자인 방법을 설명한다. 하드웨어 복잡도가 적은 효율적인 양자화 오차보상 바이어스를 생성하기 위해 Booth 인코더의 출력 정보를 이용한다. 또한, 절단된 부분이 양자화 오차에 미치는 영향에 따라 두 그룹(major or minor)으로 나누고, 각 그룹에 서로 다른 오차보상 방법을 적용한다.

본 논문은 Ⅱ장에서 modified Booth 곱셈기와 고정길이 곱셈기의 구조를 간단히 설명하고, Ⅲ장에서 modified Booth 곱셈기에 대한 새로운 오차보상 방법을 제안한다. Ⅳ장에서 approximate carry 생성 절차를 제안하고, Ⅴ장에서 기존의 방법과 제안한 방법의 절대 양자화 오차와 면적과 전력소모에 대한 성능을 비교한다. 마지막으로 Ⅵ장에서 간단히 결론을 맺는다.

Ⅱ. Modified Booth 곱셈기와 고정길이 곱셈기

본 절에서는 modified Booth 곱셈기의 특징에 대해 간단히 살펴보고 고정길이 곱셈기의 구조에 대해서 설명한다.

2.1 Modified Booth 곱셈기

Modified Booth 코딩은 부분곱의 수를 줄이기 위한 기법으로 가장 널리 쓰이는 방법 중의 하나이다[5]. 워드길이가 W 비트이고 2의 보수 성질을 갖는 두 분수(fractional number) X와 Y의 곱셈을 고려하자.

표 1. Modified Booth 인코딩 표

y_{2i+1}	y_{2i}	y_{2i-1}	y'_i	X_{sel}	$2X_{sel}$	NEG
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	1

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

Modified Booth 코딩에 의해 식 (1)의 Y는 다음과 같이 표현된다.

$$Y = \sum_{i=0}^{W/2-1} y'_{W/2-1-i}\, 2^{-(2i+1)} \quad (2)$$

여기서

$$y_i' = -2y_{2i+1} + y_{2i} + y_{2i-1} \quad (3)$$

이다. Modified Booth 코딩에서 y_{-1}은 항상 0이고 W는 짝수라고 가정한다.

표 1에 식 (3)에 대한 Modified Booth 인코딩을 요약하고, 그림 1에 modified Booth 인코더 회로를 나타내었다. 또한, 그림 2에 W=8인 경우의 modified Booth 곱셈기의 부분곱들과 각 부분곱의 비트 생성회로를 나타내었다. 그림 2에서 상수 0101011은 부호확장을 제거하기 위한 compensation vector이다.

2.2 고정길이 곱셈기

그림 2(a)의 Modified Booth 곱셈기에 대한 부분곱들을 그림 3과 같이 *MP*와 *LP* 영역으로 나눌 수 있다. 좀더 효율적인 에러보상 바이어스를 생성하기 위해서 다시 *LP* 영역에서 가장 큰 weight를 가지는 LP_{major}와 그 나머지 부분인 LP_{minor}로 *LP* 영역을 나눈다.

(2W-1) 비트의 이상적인 곱 P_I는 다음과 같이 표현 할 수 있다.

$$P_I = S\_MP + S\_LP \tag{4}$$

여기서

$$S\_MP = \sum_{2W-2\geq 2i+j\geq W-1} p_{i,j}\, 2^{-(2W-2-2i-j)} \tag{5}$$

이며

$$S\_LP = \sum_{W-1\geq 2i+j\geq 0} p_{i,j}\, 2^{-(2W-2-2i-j)} + \sum_{i=0}^{W/2-1} n_{i,0}\, 2^{-(2W-2-2i)} \tag{6}$$

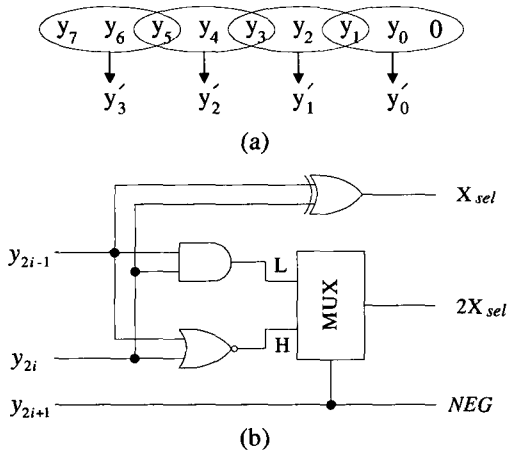


그림 1. Modified Booth 인코더: (a) 승수의 그룹핑 (W=8), (b) 인코더 회로

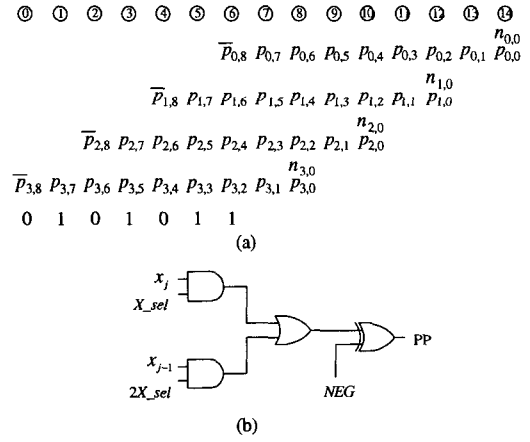


그림 2. Modified Booth 곱셈기의 부분곱(W=8): (a) 부분곱 배열, (b) 부분곱 비트 생성회로

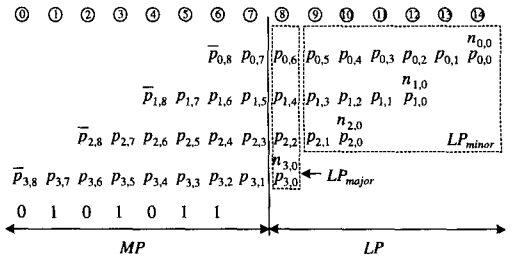


그림 3. Modified Booth 곱셈기의 MP와 LP 영역

이다. 즉, S_MP와 S_LP는 각각 MP 영역과 LP 영역에 포함된 항들의 합을 나타낸다.

일반적인 고정길이 곱셈기는 S_LP에 대해 소요되는 adder cell을 생략하고, 확률적인 추정에 근거하여 생략된 adder cell을 대신하는 적절한 바이어스를 S_MP에 더해준다. 따라서 W 비트로 양자화된 곱 P_Q는 다음과 같이 표현 될 수 있다.

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

여기서 σ는 오차보상 바이어스이다. 즉, LP 영역에서 MP 영역으로 전파되는 approximate carry 신호이다.

참고문헌 [4]에서 LP 영역으로부터 생성된 carry 신호의 합을 다음과 같이 정의 하였다.

$$\sigma_{[4]} = \lfloor \frac{1}{2}\beta + \lambda \rfloor \tag{8}$$

여기서 β는 LP_{major} 원소의 합이고 λ는 LP_{minor} 원소의 합이다. 여기서, $\lfloor t \rfloor$는 t 보다 작거나 같은 최대 정수를 의미한다. 주어진 β에 대해 통계적 분석 방법을 이용하여 $\sigma_{[4]}$의 값을 분석한 결과, 최선의 $\sigma_{[4]}$ 선택은 β가 된다고 결정하였다. 예를 들어, W=8인 경우에 대한 고정길이 modified Booth 곱셈기의 $\sigma_{[4]}$는 다음과 같이 계산된다.

$$\sigma_{[4]} = p_{0,6} + p_{1,4} + p_{2,2} + p_{3,0} + n_{3,0} \tag{9}$$

이 방법의 $\sigma_{[4]}$는 LP_{minor}에서 생성되는 carry를 포함하지 않으므로 상당히 큰 오차가 발생한다.

Ⅲ. 제안하는 오차보상 방법

그림 3으로부터 S_LP는 다음과 같이 표현 할 수 있다.

$$S\_LP = S\_LP_{major} + S\_LP_{minor} \tag{10}$$

S_LP'를 다음과 같이 정의하면

$$S\_LP' = S\_LP_{major} \times 2^{W} \tag{11}$$

이다. 여기서 $S_LP'_{major}$와 $S_LP'_{minor}$는 다음과 같다.

$$S\_LP'_{major} = p_{0,6} + p_{1,4} + p_{2,2} + p_{3,0} + n_{3,0} \tag{12}$$

$$\begin{aligned} S\_LP'_{minor} &= 2^{-1}(p_{0,5} + p_{1,3} + p_{2,1}) \\ &+ 2^{-2}(p_{0,4} + p_{1,2} + p_{2,0} + n_{2,0}) \\ &+ 2^{-3}(p_{0,3} + p_{1,1}) + 2^{-4}(p_{0,2} + p_{1,0} + n_{1,0}) \\ &+ 2^{-5}(p_{0,1}) + 2^{-6}(p_{0,0} + n_{0,0}) \end{aligned} \tag{13}$$

식 (12)와 (13)으로부터 알 수 있듯이 LP_{major}는 LP 영역에서 가장 큰 weight를 가지므로 LP 영역에서 생성되는 carry 신호에 주된 영향을 준다.

그림 4는 제안한 고정길이 modified Booth 곱셈기의 구조를 나타낸다. 제안한 오차보상 바이어스는 다음과 같이 정의한다.

$$\sigma_{prop} = C_E[S\_LP'_{major} + C_A[S\_LP'_{minor}]] \tag{14}$$

여기서 $C_E[t]$와 $C_A[t]$는 각각 t에 대한 exact carry 값과 approximate carry 값으로 정의된다. 식 (14)에서 $C_A[S_LP'_{minor}]$는 LP_{minor}로부터 LP_{major}로 전파되는 approximate carry 값이다.

위에서 설명한 고정길이 Modified Booth 곱셈기의 오차보상 바이어스 계산절차를 요약하면 다음과 같다.

1. LP 영역을 LP_{major}와 LP_{minor}로 나눈다.
2. LP_{minor}로부터 approximate carry 값을 계산한다.
3. LP_{major}에 approximate carry 값을 더한다.
4. 단계 3의 덧셈 후 생성된 carry 값이 에러보상 바이어스이다.

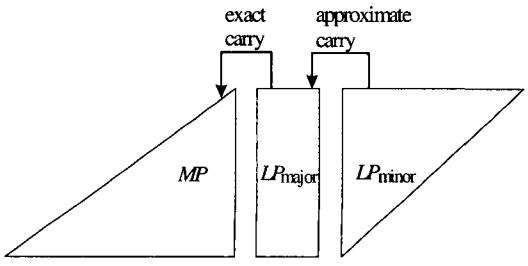


그림 4. 제안한 고정길이 Booth 곱셈기 구조

Ⅳ. Approximate Carry 생성

부분곱 생성될 때 주된 정보를 제공하는 파라미터 y_i''를 다음과 같이 정의하자.

$$y_i'' = \begin{cases} 1, & \text{if } y_i' \neq 0 \\ 0, & otherwise \end{cases} \tag{15}$$

표 1로부터 y_i''는 다음과 같이 간단히 계산된다.

$$y_i'' = X_{i\_sel} \vee 2X_{i\_sel} \tag{16}$$

여기서 $\vee$는 OR 연산자이다.

예를 들면, W=8인 경우에 $y_3''y_2''y_1''y_0''$=0001일 때 가질 수 있는 승수는 오직 3개로서 다음과 같다.

$$\begin{aligned} &00000001(0) \rightarrow y_3'y_2'y_1'y_0' = 0001 \\ &11111110(0) \rightarrow y_3'y_2'y_1'y_0' = 000\,\overline{2} \\ &11111111(0) \rightarrow y_3'y_2'y_1'y_0' = 000\,\overline{1} \end{aligned} \tag{17}$$

그림 5는 $y_3''y_2''y_1''y_0''$ = 0001일 때 가질 수 있는 3개의 승수에 대한 부분곱을 나타낸다. 그림 5로부터 파라미터 y_i''가 1이고 입력 X의 각 비트가 균일한 확률 분포를 가진다고 가정하면 $E[x_i=1]$ = 0.5이다.

그림 3에서 y_3'에 의해 생성되는 부분곱은 LP_{minor}에 포함되는 원소가 존재하지 않으므로 y_3'은 $E[S_LP'_{minor}]$에 전혀 영향을 주지 않음을 알 수 있다. 그림 3에서 $y_2''y_1''y_0''$ = 100일 때, 식 (13)을 이용하여 $E[S_LP'_{minor}]$를 계산하면 다음과 같다.

$$\begin{aligned} E[S\_LP'_{minor}] &= E[2^{-1}(p_{2,1}) + 2^{-2}(p_{2,0} + n_{2,0})] \\ &= 2^{-1}(2^{-1}) + 2^{-2}(2^{-1} + 2^{-1}) \\ &= 2^{-1} \end{aligned} \tag{18}$$

⓪	①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩	⑪	⑫	⑬	⑭
$y_3'y_2'y_1'y_0'=0001$														0
						x_7	x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0
$y_3'y_2'y_1'y_0'=000\overline{1}$														1
						$\overline{x}_7$	$\overline{x}_7$	$\overline{x}_6$	$\overline{x}_5$	$\overline{x}_4$	$\overline{x}_3$	$\overline{x}_2$	$\overline{x}_1$	$\overline{x}_0$
$y_3'y_2'y_1'y_0'=000\overline{2}$														1
						$\overline{x}_7$	$\overline{x}_6$	$\overline{x}_5$	$\overline{x}_4$	$\overline{x}_3$	$\overline{x}_2$	$\overline{x}_1$	$\overline{x}_0$	1
						MP					*LP*			

그림 5. y3″ y2″ y1″ y0″ = 0001일 경우에 대응되는 3개의 부분곱들(W=8)

같은 방법을 적용하여 $y_2''y_1''y_0''$ = 010과 $y_2''y_1''y_0''$ = 001인 경우에 계산하면 $E[S\_LP'_{minor}]$ = 0.5이다. 따라서 W = 8인 경우에 $E[S_LP'_{minor}]$는 다음과 같이 표현할 수 있다.

$$E[S\_LP'_{minor}] = 2^{-1}(y_2'' + y_1'' + y_0'') \quad (19)$$

다양한 W에 대해 식 (19)는 다음과 같이 확장된다..

$$E[S\_LP'_{minor}] = 2^{-1}\sum_{i=0}^{W/2-1} y_i'' \quad (20)$$

본 논문에서는 $E[S_LP'_{minor}]$의 라운드된 값을 approximate carry로 정의한다.

예제 1: W = 10인 경우에 $E[S_LP'_{minor}]$의 라운드된 값을 고려하자. 식 (20)으로부터 W=10인 경우의 $E[S_LP'_{minor}]$는 다음과 같다.

$$E[S\_LP'_{minor}] = 2^{-1}(y_3'' + y_2'' + y_1'' + y_0'') \quad (21)$$

$E[S_LP'_{minor}]$의 라운드된 최대 값은 2이므로 라운드된 값을 표현하기 위해 2개의 신호가 필요하다. 식 (21)에서 y_i''가 최소한 한 개 이상의 '1' 값을 가지면 라운드된 값은 1보다 크거나 같고, y_i''가 3개 이상의 '1' 값을 가지면 라운드된 값은 2이다. 표 2와 같은 표기법을 사용하고 위의 규칙을 적용하면 그림 6와 같은 카르노 맵을 얻을 수 있다. 따라서 a_carry_0과 a_carry_1 신호는 각 맵으로부터 다음과 같이 계산된다.

$$\begin{aligned} a\_carry\_0 &= y_3''\vee y_2''\vee y_1''\vee y_0'' \\ a\_carry\_1 &= y_3''y_2''(y_1''\vee y_0'')\vee y_1''y_0''(y_3''\vee y_2'') \end{aligned} \quad (22)$$

식 (22)은 그림 7과 같이 구현된다. a_carry_0과 a_carry_1 값이 LP_{major}에 더해진 후 생성된 carry 신호는 오차보상 바이어스로서 MP 영역에 더해진다. □

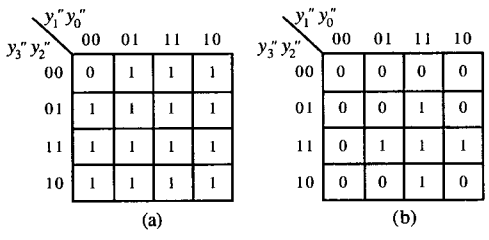


그림 6. Approximate carry 신호의 카르노 맵(W=10): (a) a_carry_0, (b) a_carry_1

표 2. Approximate carry 값의 표현

Rounded value	a_carry_0	a_carry_1
0	0	0
1	1	0
2	1	1

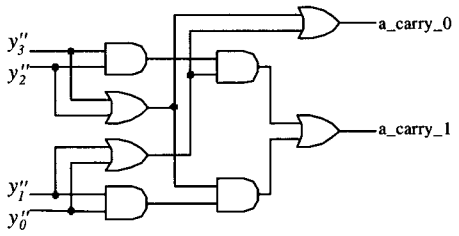


그림 7. Approximate carry 생성회로(W=10)

Approximate carry 생성절차(ACGP Ⅰ)는 다음과 같이 요약된다.

1. 주어진 W에 대하여 approximate carry 수는 $N_{AC} = \lfloor W/4 \rfloor$로 결정된다.
2. Approximate carry 신호를 a_carry_0, a_carry_1,…, a_carry_(N_{AC}-1)로 정의한다.
3. Approximate carry 신호 a_carry_i는 $y_{W/2-2}''$, $y_{W/2-3}''$, …, y_0'' 중에서 최소한 1이 (2i+1)이면 1로 결정된다.
4. 단계 3의 결과로부터 카르노 맵을 이용하여 approximate carry 신호를 디자인한다.

예제 2: 워드길이 W=12인 modified Booth 곱셈기의 경우, approximate carry 수 N_{AC}는 3이므로 a_carry_0, a_carry_1, a_carry_2의 신호가 필요하다. ACGP Ⅰ의 단계 3과 4로부터 다음과 같은 approximate carry 신호를 얻을 수 있다. □

$$\begin{aligned} a\_carry\_0 &= y_4''\vee y_3''\vee y_2''\vee y_1''\vee y_0'' \\ a\_carry\_1 &= y_4''\{y_3''y_2''\vee y_1''y_0''\vee(y_3''\vee y_2'')(y_1''\vee y_0'')\} \\ &\quad \vee y_3''y_2''(y_1''\vee y_0'')\vee y_1''y_0''(y_3''\vee y_2'') \\ a\_carry\_2 &= y_4''y_3''y_2''y_1''y_0'' \end{aligned} \quad (23)$$

ACGP Ⅰ의 방법은 카르노 맵을 사용하여 approximate carry 생성회로를 디자인한다. 따라서 워드길이 W가 증가할수록 카르노 맵의 복잡도는 지수적으로 증가한다. 또한, approximate carry 생성회로의 계산시간이 길어지므로 고정길이 곱셈기의 critical path가 길어지는 단점이 있다.

한편, $E[S_LP'_{minor}]$의 라운드된 값은 a_carry_i의 합과 같으며 다음과 같이 쓸 수 있다.

$$\sum_{i=0}^{N_{AC}-1} a\_carry\_i = \left\{ \sum_{i=0}^{W/2-1} \frac{y_i''}{2} \right\}_r \qquad (24)$$

워드길이 W의 증가로 인한 단점을 보안하기 위해 식 (24)에 근거하여 approximate carry 생성절차 (ACGP Ⅱ)를 제안한다.

1. 부분곱 생성 시 주된 정보를 제공하는 파라미터 $\{y_{W/2-2}'', y_{W/2-3}'', \cdots, y_0''\}$을 세 개의 그룹으로 나눈다. 신호 수가 $3N+k$이면 마지막 그룹의 신호 개수는 k이다. $3N$ 개의 신호는 N 개의 FA(Full Adder)를 사용하여 더한다. k=2이면 HA(Half Adder)로 더하고, k=1이면 다음 stage로 넘긴다. 각 adder로부터 N(or, N+1 for k=2) 개의 approximate carry 신호가 생성된다.
2. 단계 1과 같은 원리를 이용하여 stage 1에서 생성된 sum 신호를 더한다. 이때 각 adder로부터 approximate carry 신호가 생성된다. 새로 생성된 sum 신호는 다음 stage로 넘긴다.
3. Sum 신호가 1개가 될 때까지 step 1, 2를 반복한다. Adder의 총 수는 approximate carry 수 N_{AC} 와 같다.
4. 라운드 연산을 위해 1을 마지막 adder에 더한다.

워드길이 W = 8, 10, 14에 대한 ACGP Ⅱ로부터 생성된 approximate carry 신호를 그림 8에 나타내었다. 그림 8로부터 워드길이 W가 증가해도 approximate carry에 대한 계산시간이 W에 비례해서 증가하지 않을 뿐만 아니라 카르노 맵을 이용하여 디자인하는 방법보다 간단함을 알 수 있다.

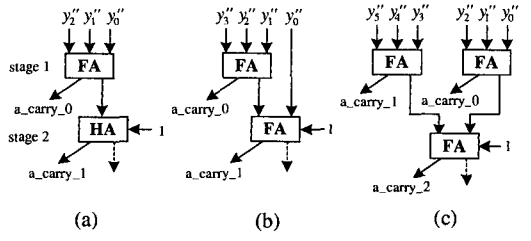


그림 8. ACGP Ⅱ에 의해 디자인된 approximate carry 생성회로: (a)W=8, (b)W=10, (c)W=14

Ⅴ. 제안한 곱셈기의 성능분석

이 장에서는 예제 1에서 디자인된 approximate

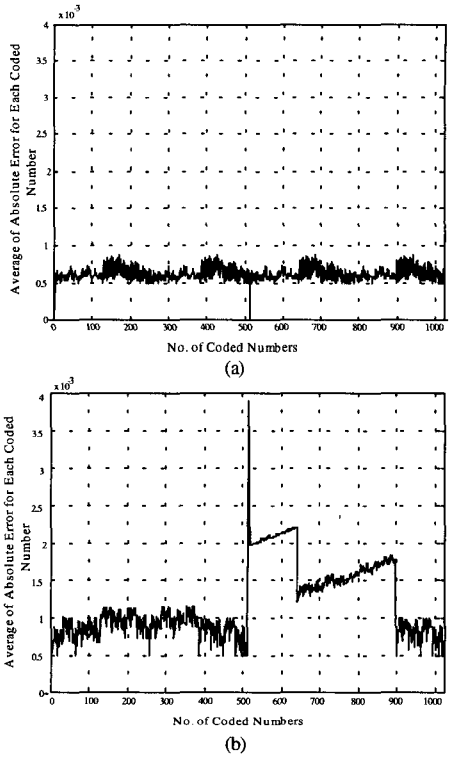


그림 9. 절대오차의 평균 비교(W=10): (a) 제안한 방법, (b) 참고문헌 [4]의 방법

carry와 식 (14)를 이용하여 오차보상 바이어스를 생성한 후 W = 10인 경우에 대해 기존의 방법과 제안한 고정길이 곱셈기의 성능을 평가한다. 모든 가능한 10 비트 계수 각각에 대해 모든 가능한 10 비트 입력을 고려하여 절대 양자화 오차 $\varepsilon(=|P_I - P_Q|)$의 평균을 계산하였다. 그림 9에 참고문헌 [4]의 방법과 제안된 방법의 절대오차 평균을 비교하였다. 그림 9(b)에 나타난 피크 점은 1000000000 × 1000000001의 계산 시 발생하는 오버플로 때문이다. 표 4에 다양한 양자화 방법의 절대오차의 평균과 분산을 비교하였다. 표 4에서 Round와 Truncation은 $(2W-1)$의 이상적인 곱 P_I로부터 W 비트로 각각 라운드하는 방법과 절단하는 방법이다. 또한, $\sigma_{[2]}$는 Baugh-Wooley 곱셈기에 대한 참고문헌 [2]의 오차보상 방법이다. 또한, 워드길이 W = 12, 16인 경우에 대해, 표 4와 5에서 다양한 양자화 방법의 결과를 비교하였다. 제안한 $\sigma_{prop.}$의 절대 양자화 오차(ε)의 평균은 $\sigma_{[2]}$와 $\sigma_{[4]}$의 절대 양자화 오차 평균의 약 50% 임을 알 수 있다.

제안한 방법과 참고문헌 [4] 방법의 면적과 전력 소모를 비교하기 위해 워드길이 W = 10, 12인 경

우의 고정길이 modified Booth 곱셈기를 VHDL(Very High Speed Integrated Circuit Hardware Description Language)로 구현하고, Synopsys의 class cell library를 사용하여 합성한 후 면적과 전력소모를 비교하였다. 표 6과 7에서 Ideal은 양자화하지 않은 이상적인 곱셈기이다. 표 6과 7로부터 고정길이 곱셈기는 이상적인 modified Booth 곱셈기의 면적과 전력소모를 약 40% 정도 절약할 수 있다.

이상의 결과로부터 제안한 고정길이 modified Booth 곱셈기의 면적과 전력소모는 참고문헌 [4] 방법과 거의 비슷하지만 양자화 에러는 약 50% 정도 적음을 알 수 있다.

표 3. 절대오차에 대한 다양한 양자화 방법의 평균과 분산 비교(W=10)

방법	평균	분산
Round	4.8733×10^{-4}	7.9337×10^{-8}
Truncation	9.6607×10^{-4}	3.1696×10^{-7}
$\sigma_{[2]}$	13.5830×10^{-4}	6.2840×10^{-7}
$\sigma_{[4]}$	12.1524×10^{-4}	8.2146×10^{-6}
$\sigma_{prop.}$	6.2841×10^{-4}	1.9412×10^{-7}

표 4. 절대오차에 대한 다양한 양자화 방법의 평균과 분산 비교(W=12)

방법	평균	분산
round	1.2201×10^{-4}	4.9743×10^{-9}
truncation	2.437×10^{-4}	1.9845×10^{-8}
$\sigma_{[2]}$	3.6084×10^{-4}	4.6719×10^{-8}
$\sigma_{[4]}$	3.1561×10^{-4}	5.2085×10^{-7}
$\sigma_{prop.}$	1.6276×10^{-4}	1.3271×10^{-8}

표 5. 절대오차에 대한 다양한 양자화 방법의 평균과 분산 비교 (W=16)

방법	평균	분산
round	7.5124×10-6	1.9530×10-11
truncation	1.4784×10-5	7.9612×10-11
$\sigma_{[2]}$	2.4267×10-5	2.3987×10-10
$\sigma_{[4]}$	1.9211×10-5	1.9174×10-10
$\sigma_{prop.}$	1.0726×10-5	6.3770×10-11

표 6. Synopsys 시뮬레이션 결과(W=10)

	게이트 수	전력소모(uW)
Ideal	811(100%)	1246.3(100%)
Fixed by σ[4]	501(61.8%)	761.1(61.0%)
Fixed by σprop.	506(62.4%)	781.8(62.7%)

표 7. Synopsys 시뮬레이션 결과(W=12)

	게이트 수	전력소모(mW)
Ideal	1079(100%)	1.6429(100%)
Fixed by $\sigma_{[4]}$	682(61.8%)	1.0440(63.6%)
Fixed by σprop.	717(62.4%)	1.0658(64.9%)

VI. 결 론

고정길이 modified Booth 곱셈기의 오차보상 바이어스를 생성하기 위해 부분곱을 MP와 LP 영역으로 나눈 후 LP 영역을 다시 LP_{major}와 LP_{minor} 그룹으로 나눠 각 그룹에 서로 다른 오차보상 방법을 적용하였다. 부분곱 생성할 때 주된 정보를 제공하는 파라미터 y_i''를 도입하여 LP_{minor}로부터 LP_{major}로 전파되는 approximate carry를 계산 후 LP_{major}에 더해 발생하는 carry를 오차보상 바이어스로 결정하는 새로운 고정길이 곱셈기 설계방법을 제안하였다.

시뮬레이션을 통해 제안한 고정길이 modified Booth 곱셈기가 참고문헌 [2]의 고정길이 Baugh-Wooley 곱셈기와 참고문헌 [4]의 고정길이 modified Booth 곱셈기보다 약 50% 정도 양자화 오차가 적으며, 좀더 라운딩된 값에 근접함을 알 수 있다. Synopsys Tool을 이용한 면적과 전력소모의 비교 시뮬레이션에서 참고문헌 [4]의 방법과 아주 근소한 차이가 남을 보였으며, 양자화하지 않은 이상적인 곱셈기와 비교하여 약 40%의 면적과 전력소모의 감소를 보였다. 따라서 제안한 고정길이 modified Booth 곱셈기를 통신 시스템 및 DSP 응용에 사용한다면 보다 정확하면서 면적과 전력소모가 적은 시스템 구현에 도움이 될 것으로 기대된다.

참 고 문 헌

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

[2] J. M. Jou and S. R. Kuang, R. D. Chen, "Design of a low-error fixed-width multipliers for DSP applications," *IEEE Trans. Circuits Syst. II*, vol. 46, no. 6, pp. 836-842, June 1999.

[3] L.-D. Van, S.-S. Wang, and W.-S. Feng, "Design of the lower error fixed-width multiplier and its application," *IEEE Trans. Circuits Syst. II*, vol. 47, no. 10 pp. 1112-1118, Oct. 2000.

[4] S. J. Jou and H. H. Wang, "Fixed-width multiplier for DSP application," in *Proceedings of 2000 ICCD*, (Austin, TX), pp. 318-322, Sept. 2000.

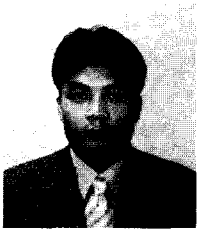
[5] O. L. MacSorly, "High speed arithmetic in binary computers", *Proc. IRE*, vol. 49, pp. 67-91, Jan, 1961.

[6] C. J. Nicol and P. Larsson, "A low-power multiplication for FIR filters," in *Proceedings of IEEE Int. Symp. Low Power Electronic and Design*, (Monterey, CA), pp. 76-79, Aug 1997.

[7] S. M. Kim, J. G. Chung, and K. K. Parhi, "Design of low error CSD Fixed-width multiplier", in *Proc. IEEE ISCAS*, (Phoenix, AZ), pp. 69-72, June 2002.

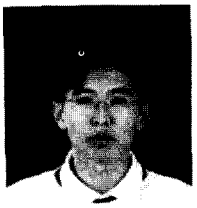
[8] K. J. Cho, K. C. Lee, J. G. Chung and K. K. Parhi, "Low Error Fixed-width Modified Booth Multiplier," in *Proc. of 2002 SIPS*, (San Diego, CA), pp. 45-50, Oct 2002.

조 경 주(Kyung-Ju Cho) 정회원



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

김 원 관(Won-Kan Kim) 정회원



2002년 2월 : 호원대학교 정보통신공학과(공학사)
현재 : 전북대학교 정보통신공학과 석사과정
<주관심 분야> OFDM, VLSI 신호처리

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 : 전북대학교 전자정보공학부 부교수
1996년 9월~현재 : 전북대학교 정보통신 연구소 연구원
2003년 2월~현재 : 전북대 부설 전북 실리콘밸리 교육센터장
<주관심 분야> VLSI 신호처리, 고속 DSL 모뎀