

작은 오차를 갖는 절사형 Booth 승산기

정해현*, 박종화**, 신경욱*

금오공과대학교 전자공학부*, 서두로직(주) EDA 사업부**

A Low-Error Truncated Booth Multiplier

Hae-Hyun Jung*, Jong-Wha Park**, Kyung-Wook Shin*

* Kumoh National University of Technology, ** Seodu Logic Inc.

요 약

N-비트×N-비트 승산에서 승산결과 2N-비트 중 상위 N-비트만을 출력하는 절사형(truncated) Booth 승산기의 절사오차 최소화를 위한 효율적인 오차보상 방법을 제안하였다. 제안된 방법을 적용하여 작은 칩 면적과 저전력 특성을 갖는 절사형 승산기를 설계하고 면적, 절사오차 등을 기존의 방식과 비교하였다. 제안된 절사형 Booth 승산기는 승산결과의 하위 N-비트를 계산하는 회로를 생략하므로, 절사되지 않은 일반 승산기에 비해 게이트 수가 약 35%~40% 정도 감소한다. 본 논문에서 설계된 절사형 Booth 승산기는 기존의 고정 오차보상 방법을 적용한 경우에 비해 평균오차를 약 30%~40% 정도 줄일 수 있다.

ABSTRACT

This paper describes an efficient error-compensation technique for designing a low-error truncated Booth multiplier that receives two N-bit numbers and produces an N-bit product by eliminating the N least-significant bits. Applying the proposed method, a truncated Booth multiplier for area-efficient and low-power applications has been designed, and its performance (truncation error, area) was analyzed. Since the truncated Booth multiplier omits about half the partial product generators and adders, it has an area reduction by about 35%~40%, compared with non-truncated parallel multipliers. Error analysis shows that the proposed approach reduces the average truncation error by approximately 30%~40%, compared with conventional methods.

1. 서 론

승산기는 통신 및 디지털 신호처리 프로세서, 디지털 필터, 마이크로 프로세서 등의 핵심 구성 요소이며, 칩의 면적과 동작 속도에 큰 영향을 미치는 연산장치이다. 두 이진수에 대한 승산은 부분곱 생성과 생성된 부분곱의 가산으로 이루어지며, 부분곱 생성과 가산 방식에 따라 매우 다양한 승산기 구현 방법들이 제안되고 있다. 널리 사용되고 있는 승산기 구조는 전가산기 배열을 이용하여 부분곱을 가산하는 배열 승산기와 수정형 Booth 알고리즘^[1,2]을 이용하여 부분곱의 수를 줄이는 Booth 승산기 등이 있다. 일반적으로, 승수와 피승수가 각각 N-비트인 승산의 결과는 2N-비트가 되며, 승산기의 비트 수가 커질수록 부분곱의 수와 부분곱의 비트 수가 증가하므로 승산기 회로의 복잡도, 승산시간, 전력소모 등이 크게 증가한다.^[3]

데이터 패스의 비트 수가 N-비트로 고정된 디지털 신호처리 분야에서는 승산 결과의 하위 N-비트를 절사(truncation)하고 상위 N-비트만을 사용한다. 이때, 승

산결과를 절사하는 방법은 크게 두 가지로 나눌 수 있다. 첫째, 2N-비트의 승산 결과를 정확하게 계산한 후, 하위 N-비트를 잘라버리는 방법은 절사 오차는 작지만 승산 결과의 하위 N-비트를 계산하는 회로가 그대로 사용되므로 일반 승산기와 동일한 면적과 전력소모를 갖는 단점이 있다. 둘째, 승산 결과의 하위 N-비트를 계산하는 회로를 생략하여 칩 면적과 전력소모를 일반 승산기에 비해 거의 절반으로 줄이는 방법이 있으며, 이를 절사형 승산기(truncated multiplier)라고 한다. 그러나, 절사형 승산기는 절사된 하위 N-비트에 대한 연산이 생략되므로 승산 결과의 절사 오차가 커지게 되며, 따라서 절사 오차를 보상해 주기 위한 적절한 방법이 필요하다.

절사형 승산기의 오차 보상 방법은 확률적 접근 방법에 기초한 고정 오차 보상 방법과 근사화된 캐리 입력 신호(바이어스 신호)를 만들어 이용하는 가변 오차보상 방법의 두 가지로 구분된다. Kidambi에 의해 처음으로 제안된 고정 오차 보상 방법은 절사된 하위 N-비트의 부분곱에서 발생할 수 있는 캐리 신호를 확률적

으로 분석하여 고정된 보상값을 구하는 방법이다⁴⁾. 이 방법은 오차보상을 위해 추가적인 회로가 필요 없다는 장점은 있으나, 오차보상 후에 절사 오차가 비교적 크다는 단점을 갖는다. 최근, Kidambi의 고정 오차보상 방법을 Booth 승산기에 적용한 결과도 발표되었다⁵⁾. 한편, Jou는 절사된 부분곱의 하위 비트 중 일부를 이용하여 근사화된 캐리 신호(바이어스)를 생성하고, 이를 가산하는 진보된 형태의 오차 보상 방법을 배열 승산기 구조에 대해 제안하였다⁶⁾. 이 방법은 Kidambi가 제안한 고정 오차 보상 방법에 비해 절사 오차를 크게 줄일 수 있다는 장점을 가지며, 바이어스 생성 회로도 매우 단순하다. 그러나, 배열 구조의 절사형 승산기에만 적용될 수 있으며, 절사형 Booth 승산기의 오차 보상에는 직접 적용될 수 없다.

본 논문에서는 문헌 [6]의 오차 보상 개념을 절사형 Booth 승산기에 적용할 수 있도록 변형된 방법을 제안하였으며, 이를 적용한 승산기 설계에 대해 기술한다.

II. 절사형 Booth 승산기의 오차 보상

일반적으로, 승산기는 승수의 비트 수 만큼 부분곱을 생성하여 가산하는 배열 승산기와 승수에 대한 리코딩을 통해 부분곱의 수를 줄이는 Booth 승산기로 구분되며, 따라서 절사형 승산기도 위의 두 가지 방법으로 구현할 수 있다. 본 장에서는 N-비트×N-비트의 승산 결과 2N-비트 중, 하위 N-비트를 잘라버리고 상위 N-비트만을 취하는 절사형 승산기의 오차 보상 방법에 대해 기술한다. 편의상, N=8인 경우를 예로 들어 설명한다.

절사형 배열 승산은 그림 1에 표시된 cut-line 우측의 하위 N-비트 부분을 잘라 버리고 부분곱의 나머지 상위 N-비트와 오차 보상을 위한 바이어스 신호를 가산하여 N-비트의 승산 결과를 얻는다. 이에 대한 오차 보상은 아래의 식(1)과 같이 표현된다⁶⁾.

$$\alpha_{n-1} \cong \bar{\alpha}' = \begin{cases} \sum_{\substack{i+j=n-1 \\ i,j \neq n-1}} x_i y_j + 1, & \text{if } \beta = 0 \\ \sum_{\substack{i+j=n-1 \\ i,j \neq n-1}} x_i y_j, & \text{if } \beta > 0 \end{cases} \quad (1)$$

단, $\beta = \sum_{i=1}^N x_{N-1-i} y_i$ 이며, 절사된 부분의 최상위 비트에서 발생하는 캐리의 합을 나타낸다. N=8인 경우, 오차 보상에 사용될 바이어스 신호는 식(2)에 의해 6개의 OR 게이트와 1개의 NOR 게이트로 생성된다⁶⁾.

i) $\beta > 0$ 인 경우 ;

$$\alpha_{n-1} = (x_7 y_0 + x_6 y_1 + x_5 y_2 + x_4 y_3 + x_3 y_4 + x_2 y_5 + x_1 y_6 + x_0 y_7) \cdot 2^8 \quad (2-a)$$

ii) $\beta = 0$ 인 경우 ;

$$\alpha_{n-1} = (x_7 y_0 + x_6 y_1 + x_5 y_2 + x_4 y_3 + x_3 y_4 + x_2 y_5 + x_1 y_6 + x_0 y_7 + 1) \cdot 2^8 \quad (2-b)$$

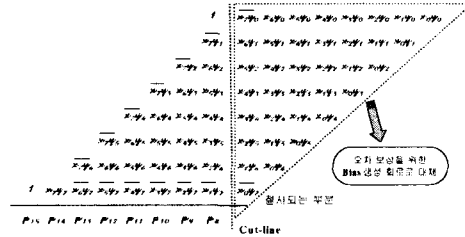
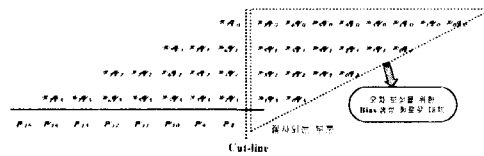
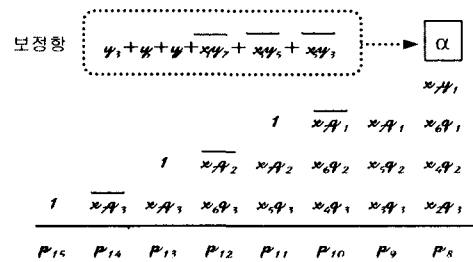


그림 1. 절사형 배열 승산 (8-b의 경우)

한편, 절사형 Booth 승산도 동일한 방식으로 그림 2(a)에 표시된 cut-line 우측의 하위 N-비트 부분을 잘라버리고 부분곱의 나머지 상위 비트만을 가산하여 8-비트의 승산 결과를 얻는다. 그러나 Booth 승산의 경우, 승수 3-비트 단위로 Booth 인코딩 되어 부분곱이 생성되므로 인접한 부분곱들은 서로 독립적이지 않다. 따라서, 그림 2(a)와 같은 cut-line으로 절사된 Booth 승산에 식(1)과 식(2)에 정의된 오차 보상 방법을 적용하면 절사 오차가 매우 커지게 된다.



(a) Booth 승산의 부분곱과 cut-line



(b) 제안된 방법에 의한 절사형 Booth 승산

그림 2. 절사형 Booth 승산 (8-b의 경우)

본 논문에서는 이와 같은 문제점을 해결하기 위해 Booth 승산의 부분곱과 배열 승산의 부분곱 사이의 관계에 대한 분석을 통해 절사형 Booth 승산기의 효율적인 오차 보상방법을 고안하였다. 이를 위해, 그림 2(a)의 cut-line으로 절사된 부분곱을 Booth 인코딩의 역과 정으로 분해한 후, 이로부터 보정항 α 를 유도함으로써

써 그림 2(b)와 같은 절사형 Booth 승산의 변형된 부분곱을 얻었으며, 따라서 식(1)과 식(2)에 정의된 오차 보상 방법을 적용할 수 있도록 하였다.

보정항 α 는 다음과 같은 두 가지를 고려하여 생성된다. 첫째, Booth 인코딩에 의해 생성되는 음의 부분곱 $(-X, -2X)$ 을 처리하기 위해, 절사 승산기의 최하위 비트 위치에 Booth 인코더에서 생성되는 부호 벡터 (y_3, y_5, y_7) 를 가산해야 한다. 둘째, 그림 2(a)에 표시된 cut-line으로 절사된 부분곱의 최하위 비트 위치에 포함되는 $(x_1 y_7 + x_3 y_5 + x_5 y_3) \cdot 2^8$ 항을 제거해야 한다. 따라서, 그림 2(b)와 같은 절사형 Booth 승산이 얻어진다.

한편, 그림 2(a)의 cut-line으로 절사된 부분곱에는 $(x_6 y_1 + x_4 y_3 + x_2 y_5) \cdot 2^8$ 가 포함되어 있으므로, 오차 보상을 위한 바이어스 회로에서 제거되며, 따라서 절사형 배열 승산기에 비해 오차 보상회로가 단순해진다.

III. 절사형 Booth 승산기 설계

본 장에서는 2장에서 설명된 오차 보상 방법을 적용한 절사형 Booth 승산기 설계에 대해 기술한다. 그림 3은 오차 보상 회로를 갖는 8-비트×8-비트 절사형 Booth 승산기의 구조를 나타낸 것이며, 부분곱 생성블록, 부분곱 가산회로, 오차보상 회로 등으로 구성된다.

부분곱 생성블록은 승수에 대한 Booth 인코딩을 수행하는 인코더 블록과 여기서 생성된 3-b 신호와 피승수를 받아 부분곱을 생성하는 디코더 블록으로 구성된다. 디코더 블록에서는 절사된 부분곱의 최하위 비트 위치(즉, 2^8 위치)에 가산될 보정항 α 가 생성된다. 한편, 첫 번째 부분곱 $x_7 y_1$ 은 Booth 인코더/디코더 블록을 거치지 않고 AND 게이트를 이용하여 직접 생성된다. 생성된 부분곱은 캐리 보존 가산기(carry-save adder)에 의해 가산되어 최종 승산결과가 얻어진다.

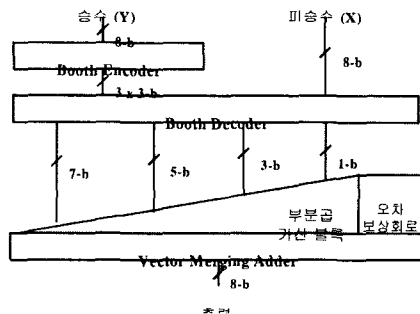


그림 3. 오차보상을 갖는 8-b×8-b 절사형 Booth 승산기의 구조

그림 4는 오차 보상 회로와 보정항 처리 회로가 포함된 부분곱 가산 블록의 내부 구성도이며, 부호확장

제거 기법을 적용하였다. 오차 보상 회로는 4-입력 OR 게이트 1개, 2-입력 OR 게이트 3개, 그리고 2-입력 NOR 게이트 1개로 구성되며, 부분곱 가산 블록의 최하위 비트(즉, 2^8 위치)로 입력되는 4개의 오차 보상 신호(바이어스 신호)를 생성한다. 오차 보상 회로는 매우 단순하므로, 승산기 전체 면적에서 차지하는 비율은 거의 무시할 수 있다. 한편, 보정항 처리 회로는 그림 2(b)에 표시된 보정항 α 와 첫 번째 부분곱 $x_7 y_1$ 을 가산하기 위해 사용된다.

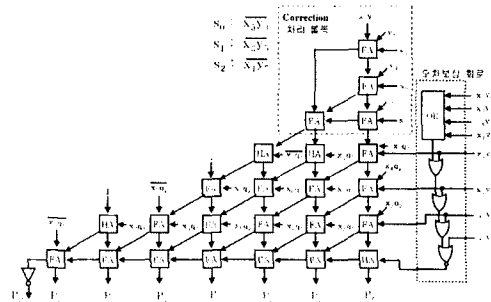
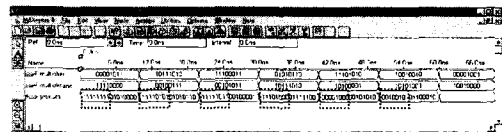
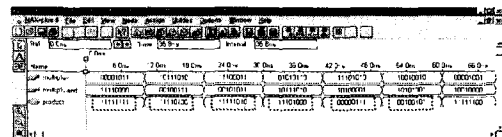


그림 4. 8-b×8-b 절사형 Booth 승산기의 오차보상과 보정항 처리 회로 및 부분곱 가산 블록

설계된 절사형 Booth 승산기는 VHDL을 이용하여 모델링되었으며, Max+Plus II와 C언어를 사용하여 검증하고, Synopsys를 사용하여 합성하였다. 그림 5는 논리 검증 결과를 보인 것이다. 그림 5(a)는 절사되지 않은 일반 승산기에서 16-비트의 승산결과가 출력되는 것을 보이고 있으며, 그림 5(b)는 본 논문에서 설계된 절사형 승산기의 동작을 보이고 있다. 일반 승산기의 출력 중 상위 8-비트(그림 5(a)에서 점선으로 표시된 부분)와 그림 5(b)에 나타난 8-비트의 승산결과를 비교하면, 일부 승산 결과의 최하위 비트 값에만 차이가 있어 절사 오차가 매우 작음을 알 수 있다.



(a) 절사되지 않은 일반 승산기의 승산 결과



(b) 제안된 방법으로 설계된 절사형 Booth 승산기의 승산 결과

그림 5. 8-b×8-b 승산기의 시뮬레이션 결과

절사형 승산기의 성능은 승산 결과에 대한 평균 오차와 분산 그리고 면적을 통해 평가될 수 있다. M 개의 데이터에 대한 절사 승산기의 평균오차 E 는 식(3)과 같이 정의된다.

$$E = \frac{1}{M} \sum_{i=1}^M (\Gamma_i - \Gamma'_i) \quad (3)$$

식(3)에서 Γ_i 는 절사되지 않은 승산 결과의 상위 N 비트의 값을 나타내며, Γ'_i 는 절사 승산기의 출력 N 비트의 값을 나타낸다.

표 1은 10,000개의 데이터에 대해 구해진 평균 오차를 비교한 것이다. 본 논문의 오차 보상 방법을 적용한 절사형 Booth 승산기는 오차 보상이 없는 절사 승산기에 비해 평균오차가 약 50%에서 70%까지 감소하며, 또한 문헌 [4]의 고정 오차 보상 방법을 적용한 승산기에 비해서도 약 40% 정도 감소한다. 본 논문의 방법은 승산기의 비트 수가 증가할수록 기존의 방법에 비해 평균 오차가 더욱 감소함을 알 수 있다.

표 2는 제안된 방법으로 설계된 절사형 Booth 승산기의 면적(Synopsys로 합성된 게이트 수)을 일반 Booth 승산기 및 문헌 [6]의 절사형 배열 승산기와 비교한 것이다. 절사되지 않은 일반 승산기에 비해 약 35%의 면적 감소가 얻어지며, 문헌 [6]의 승산기와 비슷한 면적으로 구현된다. 표 1과 표 2의 결과로부터, 본 논문의 방법은 면적과 평균 오차에서 우수한 성능을 가지며, 승산기의 비트 수가 클수록 평균 오차 및 면적이 감소하므로 비트 수가 큰 경우에도 적합하다.

표 1. 절사 승산기의 평균오차 비교

비트 수	오차보상이 없는 절사 승산기	문헌 [4]의 승산기	본 논문의 승산기
8	130.5 (1)	187.89 (1.44)	65.04 (0.50)
12	4099.5 (1)	3927.92 (0.96)	1570.32 (0.38)
16	98308.5 (1)	74497.4 (0.76)	30403.7 (0.31)

표 2. 절사 승산기의 면적 비교

비트 수	일반 승산기	문헌 [6]의 승산기	본 논문의 승산기
8	568 (1)	370 (0.65)	374 (0.66)
12	1134 (1)	780 (0.69)	734 (0.65)
16	1865 (1)	1299 (0.70)	1158 (0.62)

IV. 결 론

일반적으로, 고정 비트 크기를 갖는 디지털 신호처리 응용에서는 승산 결과의 상위 비트 절반만 연산에 사용된다. 따라서, 승산기 설계 시에 부분곱의 하위 비트 절반에 대한 연산을 생략함으로써 승산기가 차지하는 면적과 전력소모를 줄이기 위한 절사형 승산기에 대한 관심이 높아지고 있다.

본 논문에서는 절사형 Booth 승산기의 절사 오차를 최소화하기 위한 오차 보상 방법을 제안하고, 이를 적용하여 설계된 승산기의 절사 오차, 면적 등을 분석하였다. 문헌 [4]의 절사 승산기와 비교하여 평균 오차가 약 40% 정도 감소하며, 절사되지 않은 일반 승산기에 비해 약 35%의 면적 감소가 얻어진다. 본 논문의 방법은 승산기의 비트 수가 클수록 기존의 방법에 비해 평균 오차 및 면적이 감소하므로 비트 수가 큰 절사형 승산기 설계에 적합할 것으로 평가된다. 따라서, 제안된 방법은 고정 비트 크기를 갖는 디지털 신호처리 회로의 면적 감소와 저전력 설계에 폭넓게 적용 가능하다.

참고문헌

- [1] A.D. Booth, "A signed binary multiplication technique," *Quarterly J. Mechanics, Appl. Math.*, vol. 4, Part 2, pp. 236-240, 1951.
- [2] L.P. Rubinfeld, "A proof of the modified Booth's algorithm for multiplication," *IEEE Trans. on Computers*, vol. C-24, no. 10, pp. 1014-1015, Oct. 1975.
- [3] G.K. Ma and F.J. Taylor, "Multiplier policies for digital signal processing," *IEEE ASSP Magazine*, pp. 6-20, Jan. 1990.
- [4] S.S. Kidambi, F. El-Guibaly, and A. Antoniou, "Area-Efficient Multipliers for Digital Signal Processing Applications," *IEEE Trans. on Circuits and Systems II*, vol. 43, no. 2, pp. 90-95, Feb. 1996.
- [5] 이광현, 임종석, "저전력 설계를 위한 전단된 Booth 곱셈기 구조," *대한전자공학회 논문지*, vol. 37, SD-9, pp. 55-64, Set. 2000.
- [6] J.M. Jou, S.R. Kuang, and R.D. Chen, "Design of Low-Error Fixed-Width Multipliers for DSP Application," *IEEE Trans. on Circuits and Systems II*, vol. 46, no. 6, pp. 836-842, Jun. 1999.

금오공과대학교 2001년도 교내학술연구비 지원과 반도체설계교육센터(IDEC)의 CAD Tool 지원에 의한 연구 결과의 일부임.