Regular paper | Journal of the KIECS. pp. 173-180, vol. 17, no. 1, Feb. 28. 2022, t. 111, pISSN 1975-8170 | eISSN 2288-2189
http://dx.doi.org/10.13067/JKIECS.2022.17.1.173

효율적인 4-2 Compressor와 보상 특성을 갖는 근사 곱셈기

김석* · 서호성** · 김수** · 김대익***

Approximate Multiplier With Efficient 4-2 Compressor and Compensation Characteristic

Seok Kim* · Ho-Sung Seo** · Su Kim** · Dae-Ik Kim***

요 약

근사 컴퓨팅은 효율적인 하드웨어 컴퓨팅 시스템을 설계하기 위한 유망한 방법이다. 근사 곱셈은 고성능, 저전력 컴퓨팅을 위한 근사 계산 방식에 사용되는 핵심적인 연산이다. 근사 4-2 compressor는 근사 곱셈을 위한 효율적인 하드웨어 회로를 구현할 수 있다. 본 논문에서는 저면적, 저전력 특성을 갖는 근사 곱셈기를 제안하였다. 근사 곱셈기 구조는 정확한 영역, 근사 영역, 상수 수정 영역의 세 영역으로 나누어진다. 새로운 4:2 근사 compressor를 사용하여 근사 영역의 부분 곱 축소를 단순화하고, 간단한 오류 수정 방식을 사용하여 근사로 인한 오류를 보상한다. 상수 수정 영역은 오차를 줄이기 위해 확률 분석을 통한 상수를 사용하였다. 8×8 곱셈기에 대한 실험 결과, 제안한 근사 곱셈기는 기존의 4-2 compressor 기반의 근사 곱셈기보다 적은 면적을 요구하면서 적은 전력을 소비함을 보였다.

ABSTRACT

Approximate Computing is a promising method for designing hardware-efficient computing systems. Approximate multiplication is one of key operations used in approximate computing methods for high performance and low power computing. An approximate 4-2 compressor can implement hardware-efficient circuits for approximate multiplication. In this paper, we propose an approximate multiplier with low area and low power characteristics. The proposed approximate multiplier architecture is segmented into three portions; an exact region, an approximate region, and a constant correction region. Partial product reduction in the approximation region are simplified using a new 4:2 approximate compressor, and the error due to approximation is compensated using a simple error correction scheme. Constant correction region uses a constant calculated with probabilistic analysis for reducing error. Experimental results of 8×8 multiplier show that the proposed design requires less area, and consumes less power than conventional 4-2 compressor-based approximate multiplier.

키워드

Approximate Multiplier, Approximate 4-2 Compressor, Partial Product Reduction, Constant Correction, Hardware Efficient
근사 곱셈기, 근사 4-2 Compressor, 부분 곱 축소, 상수 수정, 하드웨어 효율

* 건양대학교 의공학부 (kims@konyang.ac.kr)
** 전남대학교 전기전자통신컴퓨터공학부
(216166@jnu.ac.kr, 182030@jnu.ac.kr)
*** 교신저자 : 전남대학교 전기전자통신컴퓨터공학부
· 접 수 일 : 2021. 11. 29
· 수정완료일 : 2022. 01. 08
· 게재확정일 : 2022. 02. 17
· Received : Nov. 29, 2021, Revised : Jan. 08, 2022, Accepted : Feb. 17, 2022
· Corresponding Author : Dae-Ik Kim
School of Electrical, Electronic Communication, and Computer Engr.,
Chonnam National University,
Email : daeik@jnu.ac.kr

Ⅰ. 서 론

근사 컴퓨팅은 디지털 시스템의 전력, 속도, 면적에서 성능을 향상하기 위해 정확한 계산의 요구 사항을 완화하는 새로운 설계 방식이다[1]. 이 접근 방식은 전력과 속도가 주된 요소인 임베디드 시스템과 이동형 무선통신 시스템에 사용할 수 있다[2]. 이와 더불어 근사 컴퓨팅은 멀티미디어 처리, 데이터 마이닝, 기계학습 등에 오류 회복을 위해 적용할 수 있다.

곱셈기는 필터링과 CNN (Convolutional Neural Network) 등을 응용하는 마이크로프로세서, 디지털 신호처리장치, 임베디드 시스템에 필수적인 구성요소이지만, 회로가 복잡하고 전력을 많이 소비하는 디지털 회로이다[3]. 따라서 이러한 문제를 해결하기 위해 최근 근사 곱셈기에 대한 연구가 활발하게 이루어지고 있다[4].

n 비트를 갖는 두 수를 곱하는 곱셈기는 일반적으로 아래와 같이 세 단계로 나누어 구현한다[5].

1) 피승수 (multiplicand)의 비트는 n개의 부분 곱항(n partial product term)의 집합을 생성하기 위해 승수 (multiplier)의 비트와 AND 연산을 수행한다. 이 단계는 부분 곱을 생성하기 때문에, 부분 곱 생성 (PPG; Partial Product Generation) 단계라고 한다.

2) n개의 부분 곱 항들은 적절한 방법을 통해 2개의 항으로 압축 또는 축소된다. 이 단계는 여러 개의 부분 곱 항을 축소하기 때문에, 부분 곱 축소 (PPR; Partial Product Reduction) 단계라고 한다.

3) 마지막 2개의 항은 리플 자리올림수 가산기 (Ripple carry adder) 또는 자리올림수 예측 가산기 (Carry look ahead adder) 등을 사용하여 최종 결과 값으로 산출된다.

PPG 단계는 일반적으로 AND 게이트를 사용하여 회로를 구현하지만, 부분 곱의 행을 줄이기 위해 booth 리코딩 (booth's recoding) 또는 수정 booth 리코딩 (Modified booth's recoding) 등을 적용한다.

PPG 단계에 근사 방안을 제안한 연구를 살펴보면 [7-9], 부분 곱의 AND-OR 인코딩을 사용하여 다른 정확도를 갖는 두 개의 4×4 곱셈기를 제안하였고, 16×16과 32×32 곱셈기에 이를 적용하였다[7]. 확률적 분석을 이용하여 하이브리드 부분 곱 어레이 기반의 4×4 곱셈기를 제안하였으며, 고성능의 근사 NOR 기반의 반가산기와 전가산기를 제안하여 4×4 곱셈기에 적용하였다[8]. 2-비트 디코더 로직을 사용하여 부분 곱 행의 개수를 반절로 줄이는 방법을 제안하였다[9]. 즉, 일반적인 8×8 곱셈기의 경우 8개의 부분 곱 행이 발생하는데, 새로운 디코더 로직을 사용하여 4개의 부분 곱 행이 발생한다.

곱셈기를 구현하는 세 단계에서 PPR을 수행하기 위한 단계 2)의 회로가 면적, 지연시간, 전력을 주로 소비한다[6]. PPR 단계는 주로 n-2 compressor로 구현되며, k-2 compressor, 전가산기, 반가산기 등이 이용된다. PPR을 위해 3-2 compressor인 자리올림수 저장 가산기 (Carry-save adder)를 사용하기도 하지만 4-2 compressor가 더 나은 성능을 보인다[6].

PPR에 근사 4-2 compress를 사용하면 정확도의 손실은 크지 않으면서 회로의 면적과 지연시간, 그리고 전력을 크게 줄일 수 있다. PPR 단계에 근사 방안을 제안한 연구를 살펴보면, 정확도의 손실이 있지만 적은 면적과 전력을 소모하는 두 가지 근사 4-2 compress를 제안하였다[10]. 전력의 손실이 있지만 높은 정확성을 갖는 근사 4-2 compress를 제안하였다[11]. 근사 4-2 compress를 개선하였고, PPR 구조를 절사 (truncation), 근사 (approximate), 정확한 (accurate) 영역으로 나누어 곱셈기를 구현하였다[5]. 절사 영역은 해당 곱셈 결과에 '0'을 모두 적용하였고, 근사 영역은 새로운 근사 4-2 compressor를 사용하였다. 또한 제안한 근사 compressor의 오차를 줄이기 위해 오차 수정 항을 PPR 구조의 레벨 1에서 레벨 2로 적용하였다. [5]에서 제안한 4-2 compressor를 개선하였고, PPR 구조에서 절사 영역을 모든 가능한 입력 조합의 평균치를 계산하여 상수로 교체하였다[12]. 또한 제안한 근사 compressor의 오차를 줄이기 위해 오차 수정 항을 PPR 구조의 레벨 1과 레벨 2에 모두 적용하였다.

본 논문에서는 근사 곱셈기의 면적과 전력을 개선할 수 있는 새로운 근사 4-2 compressor를 제안한다. 근사 곱셈기의 오차를 줄이기 위해 확률 분석을 통해 새로운 상수 기댓값을 하위 결과 영역 (Least significant result portion)에 적용하였고, 근사 4-2 compressor의 오차를 보상하는 구조를 제시하였다.

2장은 기존의 근사 곱셈기에 대해 알아보고, 3장은

새로운 근사 4-2 compressor, 상수 기댓값, 그리고 오차 보상 구조를 적용한 근사 곱셈기에 대해 제안한다. 4장에서는 기존의 근사 곱셈기와 제안한 근사 곱셈기의 오차와 하드웨어 성능을 비교하여 평가하고, 5장에서 결론을 맺는다.

II. 기존의 근사 곱셈기

2.1 정확한 4-2 compressor

고속 병렬 곱셈기에서 4-2 compressor는 부분 곱을 효율적으로 압축하는 과정을 빠르게 처리하는 부분 곱 합 트리의 기본적인 구성요소이며, 그림 1과 같이 두 개의 전가산기를 종속적으로 연결하여 구현한다[7]. 4-2 compressor는 5개의 입력 (*Q1, Q2, Q3, Q4, cin*)과 3개의 출력 (*sum, cout, carry*)으로 구성되고, 출력 논리식은 식(1)-(3)과 같이 표현한다. 여기에서 *cout*과 *carry*는 다음 자리로 전파되기 때문에 2배의 가중치를 갖게 된다.

$$sum = Q1 \oplus Q2 \oplus Q3 \oplus Q4 \oplus cin \quad (1)$$

$$cout = (Q1 \oplus Q2)\,Q3 + \overline{(Q1 \oplus Q2)}\,Q1 \quad (2)$$

$$carry = (Q1 \oplus Q2 \oplus Q3 \oplus Q4)\,cin + \overline{(Q1 \oplus Q2 \oplus Q3 \oplus Q4)}\,Q4 \quad (3)$$

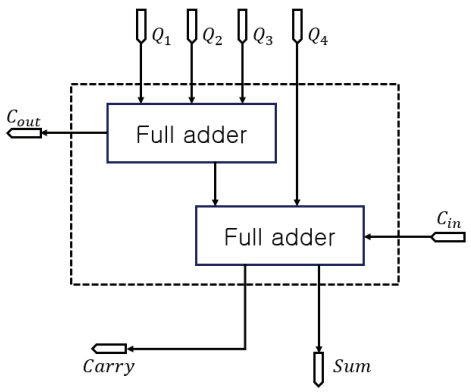


그림 1. 정확한 4-2 compressor
Fig. 1 Exact 4-2 compressor

2.2 근사 4-2 compressor와 오차 보상 모듈을 갖는 곱셈기

그림 2는 근사 compressor와 오차 보상 회로를 적용한 8×8 근사 곱셈기의 PPR 구조를 보여주고 있다[5].

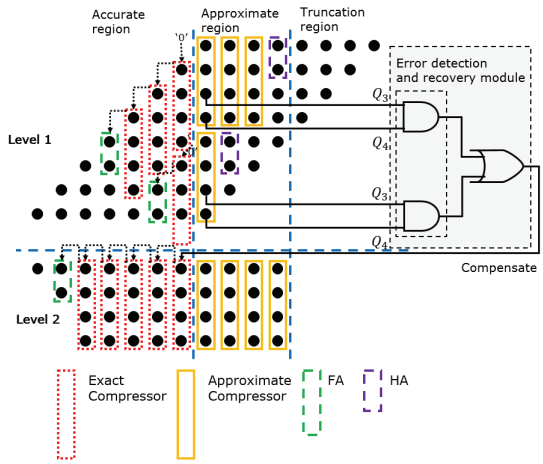


그림 2. 근사 compressor와 오차 보상 모듈 사용한 부분 곱 축소 구조
Fig. 2 Partial product reduction structure using approximate compressors and an error recovery module

근사 4-2 compressor는 두 개의 출력 *Carry*와 *Sum*으로 이루어져 있으며, 논리식은 식(4)와 식(5)와 같이 표현된다.

$$sum = (Q1 \oplus Q2) \oplus (Q3 + Q4) \quad (4)$$

$$carry = Q1 \cdot Q2 + Q1 \cdot Q3 + Q1 \cdot Q4 + Q2 \cdot Q3 + Q2 \cdot Q4 \quad (5)$$

8비트 근사 곱셈기의 PPR 구조는 그림 2와 같이 세 영역으로 나누고 각 영역은 아래와 같이 처리한다.

1) 7비트 정확한 영역: 상위 7비트에 해당하는 영역의 PPR은 정확한 4-2 compressor 사용

2) 4비트 근사 영역: 중간의 4비트에 해당하는 영역의 PPR은 근사 4-2 compressor 사용

3) 4비트 절사 영역: 하위 4비트에 해당하는 영역의 PPR은 모두 '0'으로 절사

오차를 검출하고 보상하기 위한 회로는 입력 Q4,

Q3, Q2, Q1에 따른 출력 carry와 sum에 대한 진리표를 작성하여 오차 보상 항을 도출하여 설계하였다. 16개의 입력 조합에서 Q4와 Q3의 입력이 “11”인 경우 정확한 출력값과 근사 출력값의 차이를 ‘-1’이 되도록 설정하여, 해당 입력 (Q4Q3Q2Q1: 11xx)이 인가되는 경우 AND 연산을 통해 오차를 보상하였다. 즉, 레벨 1의 근사 영역의 최상위 자리의 부분 곱을 처리하는 두 개의 근사 4-2 compressor 각각의 입력인 Q3와 Q4를 두 개의 AND 게이트와 한 개의 OR 게이트를 통해 오차 보상 값을 출력하여 레벨 2의 정확한 영역의 최하위 자리의 부분 곱을 처리하는 정확한 4-2 compressor의 입력으로 인가하였다.

2.3 오차 수정 모듈을 갖는 저전력 compressor 기반 근사 곱셈기

그림 3은 근사 compressor와 오차 보상 회로, 그리고 상수 수정 항을 적용한 8×8 근사 곱셈기의 PPR 구조를 보여주고 있다[12].

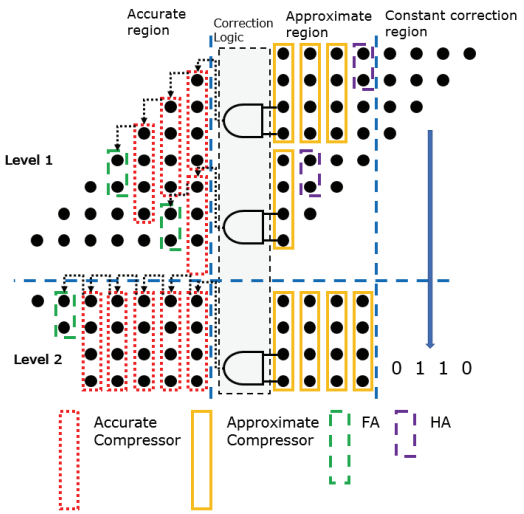


그림 3. 오차 수정 로직을 사용한 부분 곱 축소 구조
Fig. 3 Partial product reduction structure using error correction logic

근사 4-2 compressor는 두 개의 출력 *Carry*와 *Sum*으로 이루어져 있으며, 논리식은 식(6)과 식(7)과 같이 표현된다.

$$sum = G_1 \oplus G_3 \tag{6}$$

$$carry = G_2 \cdot \overline{G_1} + G_3 \cdot G_1 \tag{7}$$

여기에서, $G_1 = Q_1 \oplus Q_2$, $G_2 = Q_1 \cdot Q_2$, $G_3 = Q_3 + Q_4$ 이다.

8비트 근사 곱셈기의 PPR 구조는 그림 3과 같이 세 영역으로 나누고 각 영역은 아래와 같이 처리한다.

1) 7비트 정확한 영역: 상위 7비트에 해당하는 영역의 PPR은 정확한 4-2 compressor 사용

2) 4비트 근사 영역: 중간의 4비트에 해당하는 영역의 PPR은 근사 4-2 compressor 사용

3) 4비트 상수 수정 영역: 하위 4비트에 해당하는 영역의 PPR은 상수 “0110” 사용

그림 3의 구조에서도 그림 2의 구조와 같이 오차를 검출하고 보상하기 위한 회로는 입력 Q4, Q3, Q2, Q1에 따른 출력 carry와 sum에 대한 진리표를 작성하여 설계하였으며, 16개의 입력 조합에서 Q4와 Q3의 입력이 “11”인 경우 정확한 출력값과 근사 출력값의 차이를 ‘-1’이 되도록 설정하여, 해당 입력 (Q4Q3Q2Q1: 11xx)이 인가되는 경우 AND 연산을 통해 오차를 보상하였다. 그림 2와의 차이점은 레벨 1의 근사 영역의 최상위 자리의 부분 곱을 처리하는 두 개의 근사 4-2 compressor 각각의 입력인 Q3와 Q4를 두 개의 AND 게이트를 통해 오차 보상 값을 출력하여 같은 레벨의 정확한 영역의 최하위 자리의 부분 곱을 처리하는 정확한 4-2 compressor의 입력으로 인가한 것이다. 레벨 2에서도 레벨 1과 같은 방식으로 오차 보상 값을 처리하였다.

III. 제안한 근사 곱셈기

3.1 제안한 근사 4-2 compressor

본 연구에서는 회로의 면적, 지연시간, 전력을 줄이기 위해 *Cout*과 *Cin*을 제거한 근사 compressor를 제안하였다. 표 1은 제안한 근사 4-2 compressor에 대한 진리표를 보여준다. 기존의 연구에서는 네 가지 입력 “1100”, “1101”, “1110”, “1111”에 대해 정확한 결과와 근사 결과의 차이가 -1이 되게 하여 회로를 설

계하였으나[5, 12], 본 연구에서는 표 1과 같이 Q4가 '1'인 경우에 나머지 입력 값과 관계없이 compressor의 정확한 결과 값과 근사 결과 값의 차이를 -1이 되게 하였다. 제안한 근사 compressor의 논리식은 식(8)과 식(9)와 같고 회로는 그림 4에 보여주고 있다.

표 1. 제안한 근사 4-2 compressor의 진리표
Table 1. Truth table of the proposed approximate 4-2 compressor

Compressor input				Compressor output		Exact value (a)	Appr. value (b)	diff. (b-a)
Q4	Q3	Q2	Q1	Carry	Sum			
0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1	0
0	0	1	0	0	1	1	1	0
0	0	1	1	1	0	2	2	0
0	1	0	0	0	1	1	1	0
0	1	0	1	1	0	2	2	0
0	1	1	0	1	0	2	2	0
0	1	1	1	1	1	3	3	0
1	0	0	0	0	0	1	0	-1
1	0	0	1	0	1	2	1	-1
1	0	1	0	0	1	2	1	-1
1	0	1	1	1	0	3	2	-1
1	1	0	0	0	1	2	1	-1
1	1	0	1	1	0	3	2	-1
1	1	1	0	1	0	3	2	-1
1	1	1	1	1	1	4	3	-1

$$sum = \overline{Q3}(Q1 \oplus Q2) + Q3(\overline{Q1 \oplus Q2}) \quad (8)$$

$$carry = Q1Q2 + Q2Q3 + Q1Q3 \quad (9)$$

3.2 제안한 근사 곱셈기 구조

본 연구에서 제안한 근사 8×8 곱셈기는 그림 5와 같다. 제안한 근사 곱셈기의 PPR 구조는 그림 5와 같이 세 영역으로 나누고 각 영역은 아래와 같이 처리한다.

1) 7비트 정확한 영역: 상위 7비트에 해당하는 영역의 PPR은 정확한 4-2 compressor 사용

2) 4비트 근사 영역: 중간의 4비트에 해당하는 영역의 PPR은 제안한 근사 4-2 compressor 사용

3) 4비트 상수 수정 영역: 하위 4비트에 해당하는 영역의 PPR은 상수 "0100" 사용

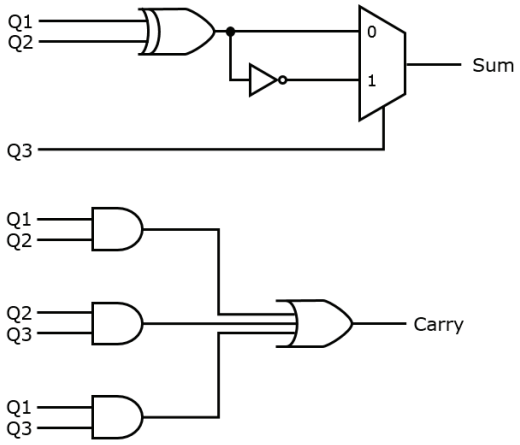


그림 4. 제안한 근사 4-2 compressor
Fig. 4 Proposed approximate 4-2 compressor

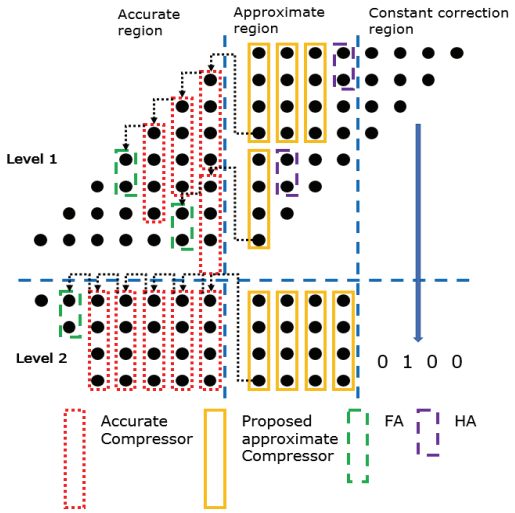


그림 5. 제안한 근사 compressor, 오차 보상 방안, 그리고 상수 수정 값을 사용한 부분 곱 축소 구조
Fig. 5 Partial product reduction structure using proposed approximate compressors, error recovery scheme and constant correction value

본 논문에서는 곱셈기의 오차를 줄이기 위해, 기존의 연구와 같이 하위 부분 곱 영역을 모두 '0'으로 절사하거나[5], 상수 "0110"을 사용하는 대신[12], 발생할 확률이 가장 높은 상수를 사용하여 상수 수정 영

역에 사용하였다. 적용할 상수는 기댓값으로 얻을 수 있으며, 기댓값이란 각 사건이 벌어졌을 때의 이득과 그 사건이 벌어질 확률을 곱한 것을 전체 사건에 대해 합한 값이다. 식(10)은 기댓값을 얻기 위해 사용한 식이다.

$$E[n] = \sum_{i=0}^{2^n-1} (p^k \times (1-p)^{n-k}) \times i \quad (10)$$

여기에서, n은 전체 비트의 수, p는 1이 나올 확률, k는 1의 비트 수, i는 상수 수정 영역의 n비트에서 나타날 수 있는 모든 수이다. 또한 피승수와 승수가 생성하는 부분 곱에 '1'이 생성될 확률은 1/4이고, '0'이 생성될 확률은 3/4이 된다. 따라서 상수 수정 영역이 4비트인 경우에는 기댓값이 $3.75_{(10)}$가 되어 이에 가장 가까운 정수 "0100"을 사용하였다.

또한, 근사 compressor의 오차를 검출하고 보상하기 위한 구조는 표 1의 진리표를 사용하여 제안하였다. Q4가 '1'인 경우에 나머지 입력 값과 관계없이 compressor의 정확한 결과 값과 근사 결과 값의 차이가 -1이 되므로, 해당 입력 (Q4Q3Q2Q1: 1xxx)이 인가되는 경우 별도의 게이트를 추가적으로 사용하지 않고 입력 Q4만을 사용하여 오차를 보상하였다. 즉, 그림 5와 같이 레벨 1의 근사 영역의 최상위 자리의 PPR을 수행하는 두 개의 근사 4-2 compressor의 각 입력 값 Q4를 같은 레벨의 정확한 영역의 최하위 자리의 PPR을 위한 두 개의 정확한 4-2 compressor의 입력으로 인가한다. 레벨 2에서도 레벨 1과 같은 방식으로 오차 보상 값을 보상하였다.

Ⅳ. 시뮬레이션 결과 및 분석

제안한 근사 곱셈기는 [5]와 [12]의 근사 곱셈기와 정확도 (오차), 면적, 지연시간, 전력에 대해 비교 분석하였다.

4.1 오차 분석

제안한 근사 곱셈기의 정확도 (오차)를 분석하기 위해 식(11)과 식(12)를 이용하여 MAE(Maximum Absolute Error)와 RMSE(Root Mean Square Error)를 계산하였다.

$$\mathrm{MAE} = | a*b - a \times b | \quad (11)$$

$$\mathrm{RMSE} = \sqrt{\frac{\sum(\mathrm{a*b} - \mathrm{a} \times \mathrm{b})^2}{2^{2\mathrm{N}}}} \quad (12)$$

여기에서 $a*b$: 근사 곱셈 (approximate multiplication), $a \times b$: 정확한 곱셈 (exact multiplication)이다.

표 2는 제안한 근사 곱셈기와 [5], [12]의 곱셈기에 대한 오차 결과를 보여주고 있다. MAE는 제안한 근사 곱셈기가 [5]에 비해 작은 수치를 보이지만 [12]에 비해 큰 수치를 보여주고 있다. 또한 RMSE는 기존의 곱셈기에 비해 2배 이상의 수치를 나타내고 있다. 기존의 연구에서는 입력 Q4와 Q3가 "11"인 경우에만 근사 compressor의 결과 값에 오차가 발생하기 때문에 확률적 오차율이 16/256인 반면에 제안한 근사 compressor의 경우에는 입력 Q4가 '1'인 경우에 결과 값에 오차가 발생하여 확률적 오차율이 64/256이 되기 때문에 RMSE가 증가한 것으로 분석된다.

표 2. 8×8 근사 곱셈기의 오차 분석
Table 2. Error analysis of 8×8 approximate multipliers

Multiplier	MAE	RMSE
[5]	321	51.95
[12]	254	47.43
Proposed	318	111.86

4.2 하드웨어 분석

곱셈기의 면적, 전력, 지연시간을 분석하기 위해 Verilog HDL을 사용하여 설계하였고, 회로합성은 25 ㎚ CMOS 공정을 Synopsys Design Complier (DC)에 적용하여 수행하였다. 표 3은 제안한 근사 곱셈기와 [5], [12]의 곱셈기에 대한 면적, 전력, 지연시간을 보여주고 있다. 제안한 곱셈기는 지연시간을 제외하고 면적과 전력에서 개선된 수치를 보여주고 있다. 제안한 근사 compressor는 기존의 compressor에 비해 논리게이트를 적게 사용하였고, 오차를 보상하는 부분에서 게이트를 사용하지 않았기 때문에 개선된 결과를 얻은 것으로 분석된다.

표 3. 8×8 근사 곱셈기의 합성 결과
Table 3. Synthesis results of 8×8 approximate multipliers

Multiplier	Area (μm^2)	Power (nW)	Delay (ns)
[5]	7096.32	67.96	17.58
[12]	7211.52	64.23	17.77
Proposed	6958.08	63.34	17.93

Ⅴ. 결 론

본 논문에서는 하드웨어 성능을 높일 수 있는 새로운 근사 곱셈기를 제안하였다. 제안한 근사 곱셈기는 비트의 자리수에 따라 정확한 영역, 근사 영역, 상수 수정 영역의 세 부분으로 나누어지며, 근사 영역에서는 sum과 carry만을 갖는 근사 compressor를 사용하고, 상수 수정 영역에서는 확률적인 분석에 통해 계산한 기댓값을 적용하였다. 또한 추가적인 게이트를 사용하지 않고 근사 compressor에서 발생하는 오차를 줄이는 구조를 적용하였다. 하드웨어 합성 결과 새로운 곱셈기는 면적과 전력에서 기존의 곱셈기에 비해 향상된 성능을 확인할 수 있었다.

제안한 곱셈기는 image sharpening과 JPEG compression 등의 응용 분야에 적용하여 실질적인 효과를 평가하는 것이 필요하다. 또한 하드웨어 성능을 높이기 위해 오차에 대한 손실이 발생하여 이에 대한 향후 연구가 요구된다.

References

[1] Q. Xu, T. Mytkowicz, and N. Kim, "Approximate computing: A survey," *IEEE Design & Test,* vol. 33, no. 1, 2016, pp. 8-22.

[2] D. Esposito, A. Strollo, E. Napoli, D. Caro, and N. Petra, "Approximate Multipliers Based on New Approximate Compressors," *IEEE Transactions on Circuits and Systems I,* vol. 65, no. 12, 2018, pp. 4169-4182.

[3] M. Horowitz, "Computing's energy problem (and what we can do about it)," *In IEEE Int. Solid-State Circuits Conf. (ISSCC),* San Francisco, CA, USA, Feb. 2014, pp. 10-14.

[4] H. Jiang, C. Liu, N. Maheshwari, F. Lombardi, and J. Han, "A comparative evaluation of approximate multipliers," *In 2016 IEEE/ACM Int. Symp. on Nanoscale Architectures (NANOARCH),* July 2016, pp. 191-196.

[5] M. Ha and S. Lee, "Multipliers With Approximate 4-2 Compressors and Error Recovery Modules," *IEEE Embedded System Letters,* vol. 10, no. 1, 2018, pp. 6-9.

[6] C.-H. Chang, J. Gu, and M. Zhang, "Ultra low-voltage low-power CMOS 4-2 and 5-2 compressors for fast arithmetic circuits," *IEEE Trans. Circuits Syst. I, Reg. Papers,* vol. 51, no. 10, 2004, pp. 1985-1997.

[7] M. S. Ansari, H. Jiang, B. F. Cockburn, and J. Han, "Low-power approximate multipliers using encoded partial products and approximate compressors," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems,* vol. 8, no. 3, 2018, pp. 404-416.

[8] H. Waris, C. Wang, W. Liu, J. Han, and F. Lombardi, "Hybrid Partial Product-based High-Performance Approximate Recursive Multipliers," *IEEE Transactions on Emerging Topics in Computing (Early Access),* 2020, pp. 1-6.

[9] S. Nambi, U. Kumar, K. Radhakrishnan, M. Venkatesan, and S. Ahmed, "DeBAM: Decoder Based Approximate Multiplier for Low Power Applications," *IEEE Embedded System Letters,* vol. 13, no. 4, 2020, pp. 174-177.

[10] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "Design and analysis of approximate compressors for multiplication," *IEEE Transactions on Computers*, vol. 64, no. 4, 2015, pp. 984-994.

[11] Z. Yang, J. Han, and F. Lombardi,

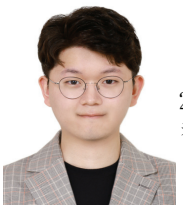
"Approximate compressors for error resilient multiplier design," *In 2015 IEEE Int. Symp. on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFTS)*, Oct. 2015, pp. 183-186.

[12] U. Kumar, S. Chatterjee, and S. Ahmed, "Low-Power Compressor-based Approximate Multipliers with Error Correcting Module," *IEEE Embedded Systems Letters (Early Access)*, 2021, pp. 1-4.

저자 소개

김석(Seok Kim)

1988년 전북대학교 전기공학과 졸업(공학사)
1990년 전북대학교 대학원 전기공학과 졸업(공학석사)
1996년 전북대학교 대학원 전기공학과 졸업(공학박사)
1996년~현재 건양대학교 의공학부 교수
※ 관심분야 : 아날로그 회로설계, 생체계측시스템



서호성(Ho-Sung Seo)

2021년 전남대학교 전기전자통신컴퓨터공학부 졸업(공학사)
2021년~현재 전남대학교 전자통신공학과 석사과정
※ 관심분야 : 저전력 회로설계, 이미지 프로세싱

김수(Su Kim)

2018년~현재 전남대학교 전기전자통신컴퓨터공학부 학사과정
※ 관심분야 : 인공지능

김대익(Dae-Ik Kim)

1991년 전북대학교 전자공학과 졸업(공학사)
1993년 전북대학교 대학원 전자공학과 졸업(공학석사)
1996년 전북대학교 대학원 전자공학과 졸업(공학박사)
2002년~현재 전남대학교 전기전자통신컴퓨터공학부 교수
※ 관심분야 : VLSI 설계, 저전력 회로설계