

논문 2011-48SD-3-7

저면적 제곱기 및 고정길이 제곱기의 설계

(Area-Efficient Squarer and Fixed-Width Squarer Design)

조 경 주*

(Kyung-Ju Cho)

요 약

제곱기의 부분곱 행렬은 대칭이므로 부분곱을 폴딩(folding), 쉬프트, 재배열하여 부분곱 행렬의 높이를 줄일 수 있다. 본 논문에서는 기존 제곱기와 비교하여 효율적인 제곱기의 설계 방법을 제안한다. 또한, 제안한 제곱기에 대해 고정길이 제곱기의 설계 방법을 제안한다. 시뮬레이션을 통해 제안한 제곱기는 기존 제곱기와 비교하여 면적은 약 17%, 지연시간(propagated delay time)은 약 10%, 전력소모는 약 10%까지 감소시킬 수 있음을 보인다. 제안한 고정길이 제곱기는 기존 고정길이 제곱기와 비교하여 절대오차와 평균오차의 성능비교에서 우수하면서, 일반 제곱기(full-width)와 비교하여 면적, 지연시간, 전력소모를 각각 30%, 16%, 28%까지 감소시킬 수 있음을 보인다.

Abstract

The partial product matrix (PPM) of a parallel squarer is symmetric. To reduce the depth of PPM, it can be folded, shifted and rearranged. In this paper, we present an area-efficient squarer design method using new partial product rearrangement. Also, a fixed-width squarer design method of the proposed squarer is presented. By simulations, it is shown that the proposed squarers lead to up to 17% reduction in area, 10% reduction in propagation delay and 10% reduction in power consumption compared with previous squarers. By using the proposed fixed-width squarers, the area, propagation delay and power consumption can be further reduced up to 30%, 16% and 28%, respectively.

Keywords : squarer, partial product reduction, fixed-width, error compensation

I. 서 론

제곱 연산은 벡터 양자화, 에러 정정, 이미지 압축, 패턴인식, OFDM 시스템에서 FEQ(Frequency Domain Equalizer)와 같은 많은 디지털신호처리 응용에 자주 사용된다. 따라서 제곱 연산을 효율적으로 수행하기 위한 많은 연구가 진행되었다^[1~5]. 승수와 피승수가 동일하다는 성질을 이용하여 제곱기의 부분곱 수와 부분곱 높이를 감소시키는 folding 기법^[1], 부분곱 비트의 재배열 기법^[2~3], Booth-folding 기법^[4] 등이 제안되었다. Booth-folding 기법은 modified Booth 알고리즘과 folding 기

법을 이용하여 제곱기의 부분곱 수와 부분곱 높이를 반으로 감소시켰으나, modified Booth 인코딩 회로와 부분곱 비트를 생성하는 회로에 대한 오버헤드가 존재한다. 또한, 수 표현 체계가 unsigned인 경우에는 MSB(Most significant bit)에 0을 1비트 추가시켜 계산해야 하는 단점이 존재한다.

제곱기 또는 곱셈기의 출력비트 수는 입력비트 수의 2배로 증가하게 된다. 그러나 대부분의 멀티미디어 및 디지털신호처리 응용에서는 내부 신호선의 수가 증가하는 것을 방지하기 위해 입력과 출력의 워드길이가 같은 고정길이 곱셈기 또는 제곱기가 요구된다. 예를 들어, 승수와 피승수가 각각 W 비트일 때 $2W$ 비트의 곱셈 출력을 얻는데 하위 LSB(Least Significant Bit)의 W 비트를 절사한 후 W 비트로 양자화 한다. 고정길이 곱

* 정회원, 항로표지기술협회

(Korea Association of Aids to Navigation)

접수일자: 2011년2월28일, 수정완료일: 2011년3월11일

셈기는 버려지는 W 비트의 계산에 필요한 adder cell을 생략하고, 확률적 추정에 의해 생략된 adder cell을 대신할 적절한 보상바이어스를 더해주는 곱셈기이다^[5~6].

본 논문에서는 제공기의 부분곱 행렬에서 반복적으로 나타나는 부분곱 패턴의 덧셈을 간소화시켜 부분곱의 비트 수와 부분곱의 높이를 감소시키는 새로운 부분곱 재배열 방법을 제안한다. 또한, 제안한 제공기의 구조를 바탕으로 고정길이 제공기 설계 방법을 제안한다.

II. 기존 제공기의 부분곱 행렬

1. Unsigned 제공기

워드길이가 W 비트인 unsigned 정수 X 를 고려하자.

$$X = \sum_{i=0}^{W-1} x_i 2^i \quad (1)$$

그림 1은 워드길이가 7 비트인 경우에 unsigned 제곱을 구하는 다양한 방법들의 부분곱 행렬이다. 어레이(array) 곱셈을 이용하면 그림 1(a)와 같은 부분곱 행렬을 만들 수 있다. 그림 1(a)에서 $x_i x_j$ 와 $x_j x_i$ 는 동일한 값이므로 부분곱 행렬은 대각선을 경계로 대칭이며, $x_i x_j + x_j x_i = 2x_i x_j$ 이므로 그림 1(a)에서 대각선 위에 있는 부분곱 비트들을 왼쪽으로 1 비트 쉬프트 시키면 대각선 아래 부분곱 비트들을 제거할 수 있다. 따라서 그림 1(b)처럼 부분곱의 비트 수를 약 반으로 줄이고 부분곱 행렬의 높이를 $(\lfloor W/2 \rfloor + 1)$ 로 줄일 수 있다. 그림 1(b)와 같은 방법을 folding 기법이라 한다.

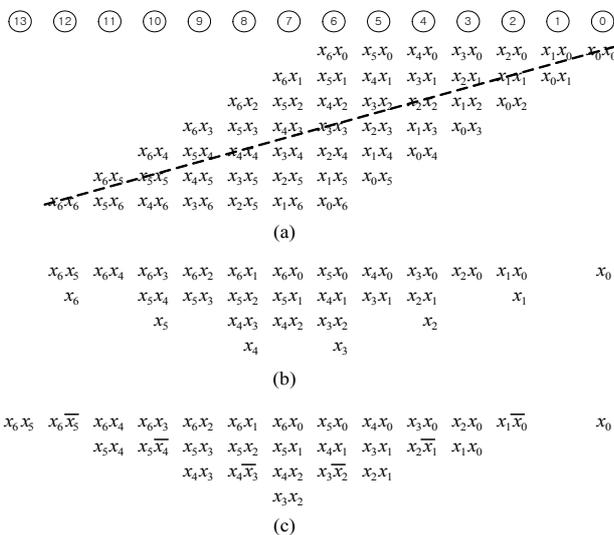


그림 1. Unsigned 제공기의 부분곱 행렬($W \neq 7$)
Fig. 1. Partial product matrices of unsigned squarer.

Folding을 적용한 부분곱 행렬에서 칼럼 $2i(i=1, \dots, W-1)$ 에서 나타나는 부분곱 비트 x_i 와 x_{W-1-i} 에 다음 식을 적용하여 그림 1(c)와 같은 부분곱 행렬을 제안하였다^[2].

$$(x_i + x_i x_{i-1}) 2^{2i} = x_i x_{i-1} 2^{2i+1} + x_i x_{i-1} 2^{2i} \quad (2)$$

이 방법의 부분곱 행렬의 높이는 $\lceil W/2 \rceil$ 이다.

2. 2의 보수 제공기

워드길이가 W 비트이고 2의 보수로 표현되는 정수 X 를 고려하자.

$$X = -x_{W-1} 2^{W-1} + \sum_{i=0}^{W-2} x_i 2^i \quad (3)$$

그림 2는 워드길이가 7인 경우, 2의 보수 제곱을 구하는 다양한 방법들의 부분곱 행렬이다. 그림 2(a)는 2의 보수 곱셈을 구하는 부분곱 행렬이고, 그림 2(b)는 folding을 이용한 부분곱 행렬이다. Folding 구조의 칼럼 $(2W-1)$ 과 $(2W-2)$ 에서 부분곱 비트들은 다음과 같이 간소화 시킬 수 있다.

$$\begin{aligned} & 2^{2W-1} + (x_{W-1} + \overline{x_{W-1} x_{W-2}}) 2^{2W-2} \\ &= \overline{x_{W-1} x_{W-2}} 2^{(2W-1)} + x_{W-1} x_{W-2} 2^{(2W-2)} \end{aligned} \quad (4)$$

워드길이를 홀수와 짝수인 경우로 구분하고 식 (2)를 적용한 부분곱 행렬에 다음 식을 적용하여 그림 2(c)와 같은 구조를 제안하였다^[3].

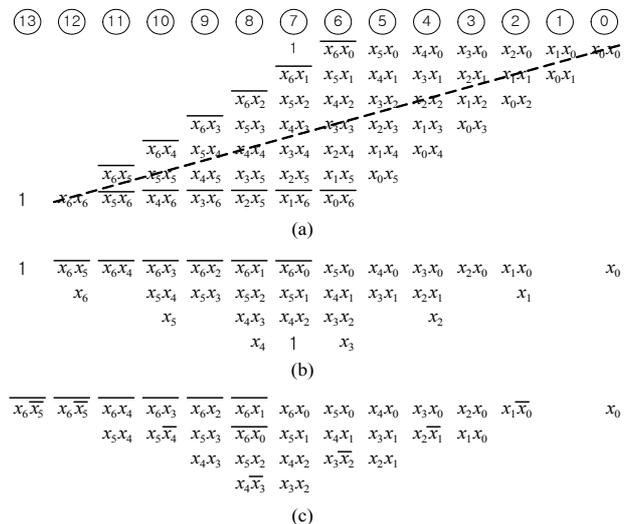


그림 2. 2의 보수 제공기의 부분곱 행렬($W \neq 7$)
Fig. 2. Partial product matrices of two's complement squarer.

$$\begin{aligned}
 & (1 + \overline{x_{W-1}x_0})2^W \\
 & = x_{W-1}x_0 2^{W+1} + x_{W-1}x_0 2^W \quad W = \text{odd} \\
 & (1 + x_{W/2} + \overline{x_{W/2}x_{W/2-1}})2^W \\
 & = x_{W/2}2^{W+1} + x_{W/2} + x_{W/2-1}2^W \quad W = \text{even}
 \end{aligned}
 \tag{5}$$

이 방법의 부분곱 행렬의 높이는 $\lceil W/2 \rceil$ 이다.

III 제안한 제곱기 및 고정길이 제곱기

1. 제안한 unsigned 제곱기

그림 1(c)의 칼럼 $(2i-1)(i=2, \dots, W-1)$ 에서 부분곱 비트 $x_i x_{i-2}, x_{i-1} x_{i-2}$ 와 칼럼 $2i$ 에서 $\overline{x_i x_{i-1}}$ 의 덧셈은 다음과 같이 간소화 시킬 수 있다.

$$\begin{aligned}
 & (x_i \overline{x_{i-1}})2^{2i} + (x_i x_{i-2} + x_{i-1} x_{i-2})2^{(2i-1)} \\
 & = x_i (\overline{x_{i-1}} \vee x_{i-1} x_{i-2}) 2^{2i} + (x_i \oplus x_{i-1}) x_{i-2} 2^{(2i-1)} \\
 & = c_{i(i-1)(i-2)} 2^{2i} + s_{i(i-1)(i-2)} 2^{(2i-1)}
 \end{aligned}
 \tag{6}$$

여기서 연산자 ‘ \vee ’와 ‘ \oplus ’는 각각 OR과 XOR 연산자이다. 그림 3은 그림 1(c)의 부분곱 행렬에 식 (6)을 차례로 적용한 부분곱 행렬이다.

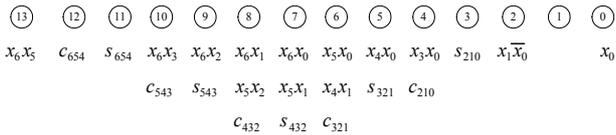


그림 3. 제안한 unsigned 제곱기의 부분곱 행렬($W \neq 7$)
Fig. 3. Proposed partial product matrix of unsigned squarer.

2. 제안한 2의 보수 제곱기

그림 2(b)의 보수 제곱기에 식 (6)을 적용하면 부분곱의 높이는 $\lceil W/2 \rceil$ 이다. 최고높이를 줄이기 위해 식 (5)와 달리 칼럼 W 에서 1, $x_{W-2}x_1$ 와 칼럼 $(W+1)$ 에서 $x_{W-2}x_0$ 의 덧셈을 다음과 같이 계산하면 부분곱의 높이를 줄일 수 있다.

$$\begin{aligned}
 & x_{W-2}x_2 2^{W+1} + (1 + x_{W-2}x_1)2^W \\
 & = x_1 x_2 x_{(W-2)} 2^{W+2} + s_{12(W-2)} 2^{W+1} + \overline{x_{W-2}x_1} 2^W
 \end{aligned}
 \tag{7}$$

식 (6)과 (7)을 그림 2(b)에 적용하면 그림 4와 같은 부분곱 행렬을 구할 수 있다. 따라서 부분곱 행렬의 높이는 $\lceil W/2 \rceil$ 에서 $\lfloor W/2 \rfloor$ 로 감소된다.

표 1에 기존 제곱기와 제안한 제곱기의 부분곱 비트 수와 부분곱의 높이를 정리하였다.

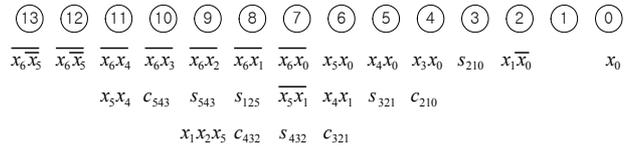


그림 4. 제안한 2의 보수 제곱기의 부분곱 행렬($W \neq 7$)
Fig. 4 Proposed partial product matrix of two's complement squarer.

표 1. 제곱기의 부분곱 비트 수와 부분곱 높이의 비교
Table 1. Comparison of the number of partial product bits and depth of partial product for squarers.

		부분곱 비트 수	부분곱 높이
Unsigned	[2]	$(W^2 + W)/2$	$\lceil W/2 \rceil$
	제안한 방법	$(W^2 - W)/2 + 2$	$\lfloor W/2 \rfloor$
2의 보수	[3]	$(W^2 + W)/2 + 1$	$\lceil W/2 \rceil$
	제안한 방법	$(W^2 - W)/2 + 4$	$\lfloor W/2 \rfloor$

3. 고정길이 제곱기 설계

워드길이가 7인 제곱기의 부분곱 행렬을 그림 5와 같이 MP(Most Significant Part)와 LP(Least Significant Part) 부분으로 나누고, LP를 다시 LP_{major} 와 LP_{minor} 로 나눈다. $2W$ 비트의 곱(product) P 는 다음과 같이 표현할 수 있다

$$P = S_{MP} + S_{LP} \tag{8}$$

여기서, S_{MP} 와 S_{LP} 는 각각 MP와 LP에 포함된 부분곱 비트들의 합이다.

일반적인 고정길이 제곱기는 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.

$$P_Q = S_{MP} + \sigma \times 2^W \tag{9}$$

여기서 σ 는 오차보상 바이어스로서 LP에서 MP로 전파되는 근사 캐리신호 값이다.

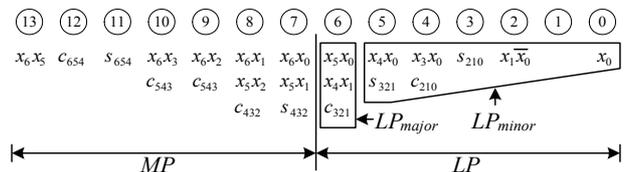


그림 5. Unsigned 제곱기의 MP와 LP($W \neq 7$)
Fig. 5. MP and LP of unsigned squarer.

LP에서 LP_{major} 는 가장 큰 weight를 가지므로 LP에서 MP로 전파되는 캐리를 계산하는데 중요한 영향을 준다. 따라서 본 논문에서는 LP_{major} 의 원소를 이용하여 S_{LP} 의 근사값을 계산하는 방법을 제안한다.

그럼 5로부터 S_{LP} 는 다음과 같이 표현할 수 있다.

$$S_{LP} = S_{LP_{major}} + S_{LP_{minor}} \quad (10)$$

$S_{LP_{major}}$ 와 $S_{LP_{minor}}$ 를 다음과 같다.

$$\begin{aligned} S_{LP_{major}} &= 2^{-1}(x_5x_0 + x_4x_1 + c_{321}) \\ S_{LP_{minor}} &= 2^{-2}(x_4x_0 + a_{321}) + 2^{-3}(x_3x_0 + c_{210}) \\ &\quad + 2^{-4}(s_{210}) + 2^{-5}(x_1x_0) + 2^{-7}(x_0) \end{aligned} \quad (11)$$

LP_{major} 값의 변화에 따른 S_{LP} 의 근사값을 계산하기 위해 LP_{minor} 를 LP_{major} 에 영향을 받는 정도에 따라 MD(More Dependent)와 LD(Less Dependent)로 구분하면 $S_{LP_{minor}}$ 다음과 같이 표현 할 수 있다.

$$S_{LP_{minor}} = S_{MD} + S_{LD} \quad (12)$$

여기서, S_{MD} 와 S_{LD} 는 다음과 같다.

$$\begin{aligned} S_{MD} &= 2^{-2}(x_4x_0) + 2^{-3}(x_3x_0) + 2^{-5}(x_1x_0) + 2^{-7}(x_0) \\ S_{LD} &= 2^{-2}(s_{321}) + 2^{-3}(c_{210}) + 2^{-4}(s_{210}) \end{aligned} \quad (13)$$

파라미터 K 를 다음과 같이 정의하자.

$$K = S_{LP_{major}} + S_{MD} \quad (14)$$

$S_{LP_{minor}}$ 의 근사값을 계산하기 위해 S_{MD} 는 LP_{major} 의 원소 값에 의존하는 $E[S_{MD}]$ 를 구하고, $E[S_{LD}]$ 는 X 의 각 비트가 균일한 확률 분포를 가진다고 가정할 때의 $E[S_{ijk}]$, $E[C_{ijk}]$ 을 적용하여 계산한다. 따라서 S_{LP} 의 근사값은 다음과 같이 구할 수 있다.

$$S_{LP} \approx K + E[S_{LD}] \quad (15)$$

여기서,

$$\begin{aligned} K &\approx S_{LP_{major}} + E[S_{MD}] \\ &\approx 2^{-1}(x_5x_0 + x_4x_1 + c_{321}) + 2^{-2}E[x_4x_0] \\ &\quad + 2^{-3}E[x_3x_0] + 2^{-5}E[x_1x_0] + 2^{-7}E[x_0] \end{aligned} \quad (16)$$

이다. 입력 X 의 각 비트가 균일한 확률 분포를 가진다고 가정하면 $E[x_i]=0.5$, $E[S_{ijk}=1]=1/4$, $E[C_{ijk}=3/8]$ 이다. 따라서 식 (15)는 다음과 같다.

$$S_{LP} \approx K + 0.125 \quad (17)$$

LP_{major} 원소 값의 변화에 따른 K 를 계산하기 위해 다음과 같은 인덱스를 도입한다.

$$I(q_2, q_1, q_0) = \langle x_5x_0 \rangle^{q_2} + \langle x_4x_0 \rangle^{q_1} + \langle c_{321} \rangle^{q_0} \quad (18)$$

여기서, $q_2, q_1, q_0 \in \{0, 1\}$ 이며

$$\langle T \rangle^{q_i} = \begin{cases} T=0, & \text{if } q_i=0 \\ T=1, & \text{if } q_i=1 \end{cases} \quad (19)$$

이다. $I(q_2, q_1, q_0)$ 는 다음과 같이 Q 번째 인덱스로 정의한다.

$$Q = q_2 \times 2^2 + q_1 \times 2^1 + q_0 \times 2^0 \quad (20)$$

만약 $Q=4$ 이면, $x_5x_0=1$, $x_4x_1=0$, $c_{321}=0$ 이므로 식 (18)을 이용하면 K 는 다음과 같이 구할 수 있다.

$$K_{Q=4} = 2^{-1} + 2^{-2}E[x_4] + 2^{-2}E[x_3] + 2^{-7} = 0.6953 \quad (21)$$

동일한 방법으로 $Q=2$, $Q=1$ 인 경우 K 는 다음과 같다.

$$\begin{aligned} K_{Q=2} &= 2^{-1} + 2^{-2}E[x_0] + 2^{-5}E[x_0] = 0.6406 \\ K_{Q=1} &= 2^{-1} + 2^{-3}E[x_0] = 0.5625 \end{aligned} \quad (22)$$

식 (21)과 (22)의 $K_{Q=4}$, $K_{Q=2}$, $K_{Q=1}$ 로부터 다른 인덱스에 대한 $K_{Q=i}$ 를 쉽게 계산할 수 있다. 예를 들면, $K_{Q=7}$ 은 $K_{Q=4} + K_{Q=2} + K_{Q=1}$ 이다.

본 논문에서는 S_{LP} 근사값의 반올림 값을 오차보상 바이어스로 정의한다. 표 2에 $K_{Q=i}$ 대한 S_{LP} 의 근사값과 오차보상 바이어스를 나타내었다. 오차보상 바이어스의 최대값이 2이므로 두 개의 바이어스 신호(bias0, bias1)가 필요하며 바이어스 신호는 다음과 같다.

$$\begin{aligned} bias0 &= x_5x_0 \vee x_4x_1 \vee c_{321} \\ bias1 &= x_5x_0 \wedge x_4x_1 \wedge c_{321} \end{aligned} \quad (23)$$

여기서, 연산자 ‘ \wedge ’는 AND 연산자이다.

2의 보수 고정길이 제공기의 오차보상 바이어스는 LP 부분이 unsigned 제공기와 같으므로 식 (23)과 동일하다.

표 2. 오차보상 바이어스 표
Table 2. Error compensation bias table.

Q	x_5x_0	x_4x_1	c_{321}	$K_Q + E[S_{LD}]$	σ_{prop}	bias0	bias1
0	0	0	0	0.125	0	0	0
1	0	0	1	0.6875	1	1	0
2	0	1	0	0.7656	1	1	0
3	0	1	1	1.3281	1	1	0
4	1	0	0	0.8203	1	1	0
5	1	0	1	1.3823	1	1	0
6	1	1	0	1.4609	1	1	0
7	1	1	1	2.0234	2	1	1

IV. 실험결과

기존 제곱기와 제안한 제곱기의 효율성을 비교하기 위해 unsigned 제곱기와 2의 보수 제곱기를 Verilog를 이용하여 구현하고 Synopsys CAD 툴을 사용하여 삼성 0.18 μ m CMOS 공정의 cell library로 합성한 결과를 표 3~5에 나타내었다. 제안한 제곱기는 기존 제곱기보다 면적에서 약 17%, 지연시간에서 약 10%, 전력소모에서 약 10%까지 우수 하였다.

제안한 고정길이 제곱기는 2W 비트를 출력하는 일반 제곱기와 비교하여 면적에서 30%, 지연시간에서 16%, 전력소모에서 28%까지 감소시킬 수 있다.

표 3. 제곱기의 면적 비교

Table 3. Comparison of area for squarers.

		W=7	W=8	W=9	W=10	W=11	W=12
Unsigned	[2]의 방법 (full-width)	204	251	360	418	515	599
	제안한 방법 (full-width)	167	225	294	379	446	558
	제안한 방법 (fixed-width)	120	164	203	245	308	386
2의 보수	[3]의 방법 (full-width)	201	248	346	414	510	594
	제안한 방법 (full-width)	175	230	301	384	450	566
	제안한 방법 (fixed-width)	131	166	210	249	310	390

표 4. 제곱기의 지연시간 비교

Table 4. Comparison of propagated delay time for squarers.

		W=7	W=8	W=9	W=10	W=11	W=12
Unsigned	[2]의 방법 (full-width)	6.25	6.29	6.85	6.69	7.59	7.70
	제안한 방법 (full-width)	5.63	6.26	6.40	6.92	7.02	7.67
	제안한 방법 (fixed-width)	4.28	5.31	5.48	5.87	6.05	6.51
2의 보수	[3]의 방법 (full-width)	6.24	6.28	7.01	7.14	7.52	7.72
	제안한 방법 (full-width)	5.61	6.24	6.39	7.05	7.12	7.69
	제안한 방법 (fixed-width)	4.30	5.34	5.48	5.89	6.10	6.61

표 5. 제곱기의 전력소모 비교

Table 5. Comparison of power consumption for squarers.

		W=7	W=8	W=9	W=10	W=11	W=12
Unsigned	[2]의 방법 (full-width)	62.9	72.5	99.1	114.2	129.3	153.1
	제안한 방법 (full-width)	56.7	67.3	91.5	108.9	122.8	147.3
	제안한 방법 (fixed-width)	37.7	48.3	60.1	73.1	85.5	99.7
2의 보수	[3]의 방법 (full-width)	63.5	73.8	99.7	116.8	130.0	154.5
	제안한 방법 (full-width)	57.2	69.5	92.1	109.2	123.6	148.1
	제안한 방법 (fixed-width)	40.7	52.1	67.5	80.9	92.5	108.4

표 6. 최대절대오차의 비교

Table 6. Comparison of maximum absolute error.

	W=7	W=8	W=9	W=10	W=11	W=12
Post	121	249	505	1017	2041	4089
[5]constant	153	513	1105	3073	6433	168385
[5]variable	111	311	535	1383	2471	6279
제안한 방법	89	217	465	1017	2359	4809

표 7. 평균오차의 비교

Table 7. Comparison of average absolute error.

	W=7	W=8	W=9	W=10	W=11	W=12
Post	121	249	505	1017	2041	4089
[5]constant	153	513	1105	3073	6433	168385
[5]variable	111	311	535	1383	2471	6279
제안한 방법	89	217	465	1017	2359	4809

제안한 고정길이 제곱기의 오차성능을 평가하기 위해 다양한 방법의 최대절대오차와 절대오차평균을 표 6~7에 비교하였다. 'Post'는 2W의 곱 P로부터 W 비트로 절사하여 양자화한 방법을 의미하고, '[5]constant'와 '[5]variable'은 기존 고정길이 제곱기의 설계^[5]에서 상수보상과 적응보상을 적용한 방법이다. 상수보상은 오차보상바이어스가 간단하지만 오차가 크고, 적응보상은 입력에 따라 변하기 때문에 오차가 작으나 상수보상 바이어스보다 복잡하다. 표 6~7로부터 제안한 방법이 기존 고정길이 제곱기 보다 오차가 적음을 알 수 있다.

V. 결론

제곱기의 부분곱 행렬의 덧셈에서 존재하는 불필요한 연산을 간소화시켜 부분곱의 비트 수를 줄이고 부분곱 높이를 감소시키는 새로운 방법을 제안하였다. 또한, 제안한 제곱기의 구조에 오차가 적은 고정길이 제곱기 설계 방법을 제안하였다. 다양한 시뮬레이션을 통해 제안한 제곱기가 기존 제곱기보다 우수함을 보였다.

참고 문헌

[1] J. Pihl and E. Aas, "A multiplier and squarer generator for high performance DSP applications," in *Proc. IEEE 39th Midwest Symp. on Circuits and Systems*, pp. 109-112, 1996

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

- [3] K. E. Wires, M. J. Schulte, L. P. Marquette, and P. I. Balzola, "Combined unsigned and two's complement squarers," in Proc. *33rd Asilomar Conference on Signals, Systems, and Computers*, pp. 1215-1219, 1999.
- [4] A. G. M. Strollo and D. D. Caro, "Booth folding encoding for high performance squarer circuits," *IEEE Trans. Circuits and Systems II*, vol. 50, pp. 250-254, May 2003.
- [5] E. G. Walters, M. J. Schulte, and M. G. Arnold, "Truncated squarers with constant and variable correction," in Proc. *SPIE: Advanced Signal Processing Algorithms, Architectures, and Implementations XIII*, Denver, Co, Aug. 2004.
- [6] L. D. Van and C. C. Yang, "Generalized low-error area-efficient fixed-width multipliers," *IEEE Trans. Circuits and Systems I*, vol. 52, pp. 1068-1619, Aug. 2005.

— 저 자 소 개 —



조 경 주(정회원)

2000년 원광대학교 전자공학과
학사 졸업

2002년 전북대학교 정보통신학과
석사 졸업

2006년 전북대학교 정보통신공학과
박사 졸업

2006년~2009년 2월 : 전북대학교 Post-Doc.

2009년~현재 향로표지기술협회 연구개발실 과장
< 주관심 분야 > VLSI 신호처리, 저전력 회로설계,
SoC 설계, 임베디드시스템 설계