

논문 2004-41SD-8-14

# 연속된 데이터를 위한 에러패턴 ROM 압축 기법

(An Error Pattern ROM Compression Method for Continuous Data)

양 병 도\*, 김 이 섭\*

(Byung-Do Yang and Lee-Sup Kim)

## 요 약

본 논문에서는 연속적인 데이터를 위한 새로운 에러패턴 ROM(error pattern ROM: EP-ROM) 압축 기법이 제안되었다. EP-ROM은 연속적인 데이터를 근사값과 에러로 나눈 후 에러 대신 에러패턴의 인덱스를 저장함으로써 ROM의 크기를 줄인다. 이 기법은 연속된 데이터를 양자화할 때의 많은 경우 같은 에러패턴을 가진다는 특징을 이용한 것으로 ROM의 크기를 매우 작게 줄일 수 있다. 실험결과에서는 EP-ROM은 다양한 연속적인 데이터에 대하여 ROM 크기를 60~77% 줄였다.

## Abstract

This paper proposes a new error pattern ROM (EP-ROM) compression method for continuous data. The EP-ROM reduces the ROM size by dividing the continuous data into coarse values and their errors and by storing the indices of error patterns instead of the errors. This method significantly reduces the ROM size by exploiting the characteristic that the errors for continuous data possess the same patterns. The experiment results show that the EP-ROM achieves 60~77% ROM size reductions for various continuous data.

**Keywords:** VLSI, CMOS, ROM, and compression

## I. 서 론

ROM(Read Only Memory)은 VLSI칩에서 다양하게 사용되고 있다. 칩의 복잡도가 증가함에 따라, ROM의 크기도 지속적으로 커지고 있다. ROM의 크기가 커질수록 더 넓은 면적과 더 많은 전력을 소모하게 된다. 따라서, ROM의 크기를 줄이고 ROM의 전력 소모를 줄이기 위한 많은 기법들이 제안되어왔다.

대부분의 기법들은 ROM의 크기 변화 없이 ROM의 전력 소모를 줄이기 위한 기법들이었다<sup>[1]-[4]</sup>. 다른 몇가지 기법들은 ROM의 크기를 줄이는 것에 초점이 맞추어졌다. ROM의 크기를 줄이는 것은 ROM의 속도를 빠르게 할 뿐만 아니라 소모 전력과 면적을 동시에 줄여

주기 때문에 매우 효과적인 방법이라 할 수 있다. 제안된 ROM 압축 기법들은 대부분 수학에 기초하고 있다<sup>[5]-[6]</sup>. 그러나, 이런 기법들은 수학적으로 표현 가능한 함수들에만 개별적으로만 적용 가능하다.

DSP(digital signal processor)에서는 연속적인 데이터가 많이 사용되고 있다. 이런 데이터들은 보통 상당히 클 뿐만 아니라 수학적인 압축이 어려운 데이터들이 많다. 따라서, ROM의 크기를 줄이기 위한 압축 기법이 요구된다. 최근에 두 가지의 압축 방식이 제안되었다<sup>[7]</sup>. 하나는 영역 ROM(region ROM: R-ROM)이고 다른 하나는 양자화/에러 ROM(quantization and error ROM: QE-ROM)이다.

R-ROM은 연속적인 데이터를 크기와 어드레스에 의하여 작은 영역들로 나눈 후 데이터를 포함하고 있는 영역의 값들만을 저장함으로써 연속된 데이터를 압축한다. QE-ROM은 연속적인 데이터를 양자화된 값과 그 에러를 분리하여 저장하여 데이터를 압축한다.

본 논문에서 연속적인 데이터를 압축하기 위한 새로

\* 정희원, KAIST 전자전산학과  
(Dept. of Electrical Engineering & Computer Science, KAIST)

※ 본 연구는 한국과학재단과 미세정보센터의 지원을 받아 수행되었습니다.

접수일자: 2004년2월19일, 수정완료일: 2004년7월13일

은 에러패턴 ROM(error pattern ROM: EP-ROM)을 제안한다. EP-ROM은 연속적인 데이터를 어드레스로 양자화 시켜 근사값과 에러로 분리한 후 에러 대신 에러 패턴의 인덱스를 저장함으로써 ROM의 크기를 줄인다. 또한, R-ROM과 EP-ROM를 결합하여 ROM의 크기를 더욱 줄인 영역 에러패턴 ROM(region error pattern ROM: REP-ROM)도 제안한다.

논문의 구성은 다음과 같다. II장에서는 기존의 ROM 압축 기법들을 소개한다. III장에서는 제안된 에러패턴 ROM의 설명한다. IV장에서는 실험 결과와 비교 결과들을 보여준다. 마지막으로 V장에서 결론을 맺는다.

## II. 기존의 ROM 압축 기법들

### 1. 영역 ROM (Region ROM: R-ROM)

그림 1은 영역 ROM(region ROM: R-ROM)의 개념을 보여준다<sup>[7]</sup>. 그림 1(a)의 연속적인 데이터는 크기와 어드레스에 의하여 많은 수의 작은 영역들로 나누어진다. 그림1(b)에서와 같이 데이터를 포함하고 있는 영역의 수는 적은 반면 많은 수의 다른 영역들은 비어있다. R-ROM은 데이터를 저장하고 있는 영역들의 데이터만을 저장함으로써 ROM의 크기를 줄인다.

그림 2(a)는 R-ROM에 저장된 데이터들을 보여주고 있다.  $2^A$ 의 어드레스를 가지는 연속적인 데이터를 N개의 컬럼으로 나눌 경우, R-ROM은 N개의 R-ROM 블록들을 가지게 된다. 각 R-ROM 블록은 똑같이  $2^R$ 의 어드레스를 가지고 R-ROM의 출력은 L비트이다. 그러나, 각 R-ROM블록들이 L비트의 데이터를 저장하고 있지는 않다. i번째 R-ROM블록은  $M_i$ 비트의 데이터를 저장한다. 그러나, R-ROM의 출력은 압축전의 데이터로 복원된다. 그 이유는 출력 데이터가 그림 2(a)에서와 같이  $L-M_i$ 비트의 영역 정보와  $M_i$ 비트의 R-ROM 데이터로 구성되기 때문이다. 그림 2(b)와 같이 MUX에서 R-ROM블록의 출력에 영역정보가 더해진다. 따라서, ROM의 크기는  $2^A \times L$ 비트에서  $2^R \times M_i$ 비트로 줄어들게 된다.

### 2. 양자화/에러 ROM

(quantization and error ROM: QE-ROM)

그림 3은 양자화/에러 ROM(quantization and error ROM: QE-ROM)의 개념을 보여주고 있다<sup>[7]</sup>. QE-ROM도 연속적인 데이터의 연속성을 이용한 ROM의 크기를 줄인다. 그림 3(b)의 연속적인 데이터가 크기와 어드레

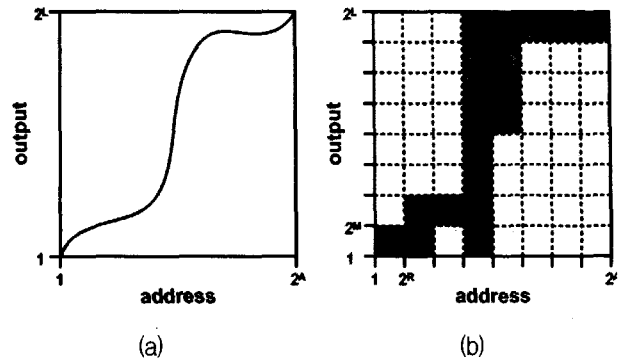


그림 1. (a) 연속적인 데이터 (b) R-ROM에 저장되는 영역들  
Fig. 1. (a) Continuous data (b) Regions stored in an R-ROM.

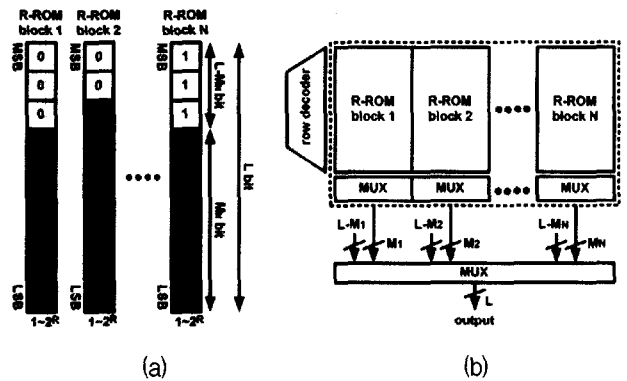


그림 2. (a) R-ROM에 저장된 데이터 (b) R-ROM 구조  
Fig. 2. (a) Data stored in an R-ROM (b) An R-ROM architecture.

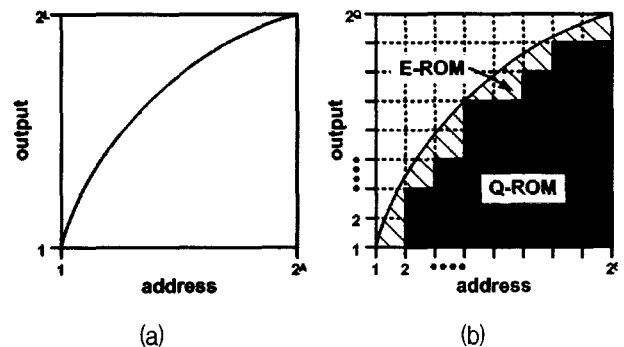


그림 3. (a) 연속적인 데이터 (b) Q-ROM과 E-ROM에 저장되는 데이터  
Fig. 3. (a) Continuous data (b) Quantization ROM and Error ROM.

스에 의하여 양자화 되었을 때, 양자화된 값들과 양자화 에러들 각각은 그림 3(b)에서 보이는 것처럼 원본 데이터보다 작게된다. 또한, 양자화된 값들과 양자화 에러들의 합도 원본 데이터 보다 작다. 어드레스가 변함에 따른 데이터의 값의 변화가 적을수록 양자화 에러는 더욱 줄어들게 된다. 즉, QE-ROM은 양자화를 이용하여 연속된 데이터를 압축한다.

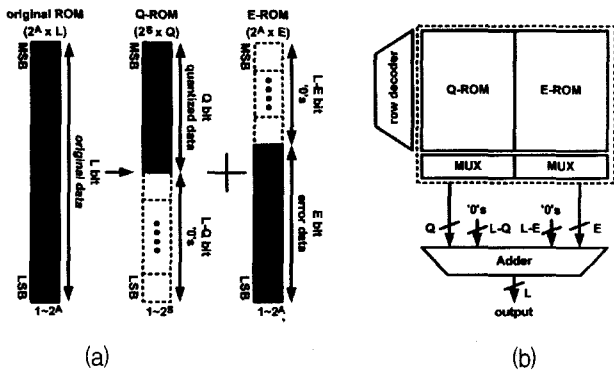


그림 4. (a) 원본 데이터와 Q-ROM과 E-ROM에 저장된 데이터 (b) Q-E-ROM 구조  
 Fig. 4. (a) Data stored in an original ROM, a Q-ROM, and an E-ROM (b) A QE-ROM architecture.

그림 4(a)는 원본 데이터와 양자화 ROM(quantization ROM: Q-ROM)과 에러 ROM(error ROM: E-ROM)에 저장된 데이터를 보여준다. QE-ROM은 Q-ROM과 E-ROM로 구성되어 있고, 각각은 양자화된 값들과 양자화 에러들을 저장하고 있다. 연속적인 데이터가 크기와 어드레스에 의하여 양자화 되었을 때, 데이터의  $2^A$ 의 어드레스는 똑같이  $2^S$ 의 섹션(section)으로 나누어지고 L비트의 데이터는 Q비트로 양자화 된다. Q-ROM은 각 섹션당 하나의 Q비트로 양자화된 값들을 저장한다. 따라서, Q-ROM의 크기는  $2^S \times Q$ 비트가 된다. 각 섹션의 양자화된 값은 양자화 에러의 크기를 양수로 만들기 위해서 그 섹션의 모든 값들보다 작은 값이 된다. E-ROM은 원본 데이터와 양자화된 값의 차이인 양자화 에러를 저장한다. 그 모든 에러들을 표현하기 위해 필요한 최소 비트가 E일 때, E-ROM의 크기는  $2^A \times E$ 비트가 된다. 따라서, ROM의 크기는  $2^A \times L$ 비트에서  $2^S \times Q + 2^A \times E$ 비트로 줄어든다. Q-ROM과 E-ROM으로부터 원본 데이터를 복원하기 위해서 그림 4(b)에서와 같이 한 개의 덧셈기가 사용된다.

### III. 제안된 에러 패턴 ROM

#### 1. 에러 패턴 ROM

(error pattern ROM: EP-ROM)

그림 5는 제안된 에러패턴 ROM(error pattern ROM: EP-ROM)의 개념을 보여준다. EP-ROM은 연속적인 데이터를 근사의 값들과 그 에러들로 나누고 에러 대신 에러의 패턴의 인덱스(index)를 저장함으로써 ROM의 크기를 줄인다. EP-ROM은 그림 5(a)의  $2^A$ 어드레스의 연속적인 데이터를 그림 5(b)에서와 같이  $2^S$ 의 섹션으

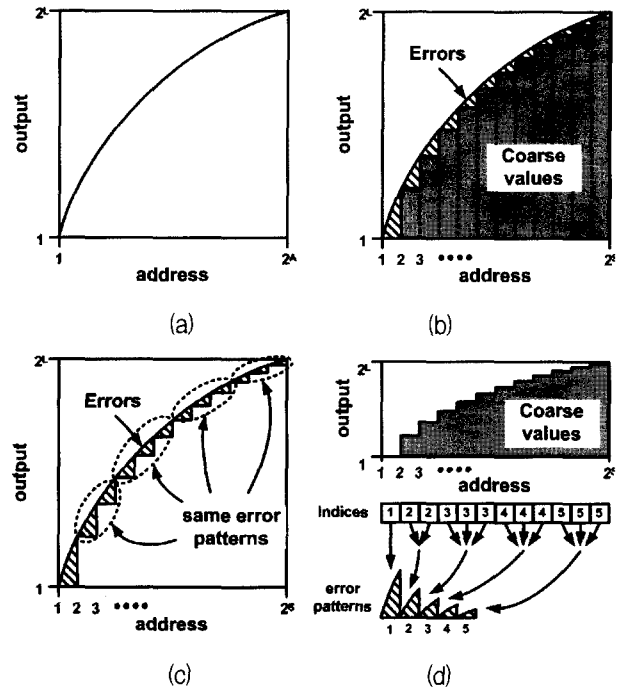


그림 5. (a) 연속적인 데이터 (b) 근사값과 에러 (c) 에러 패턴 (d) 근사값, 에러패턴의 인덱스, EP-ROM에 저장되는 에러패턴  
 Fig. 5. (a) Continuous data (b) Coarse values and their errors (c) errors patterns (d) Coarse values, indices of error patterns, error patterns stored in an EP-ROM.

로 균등하게 나눈다. EP-ROM은 섹션당 하나의 근사값을 저장한다. 따라서, 어드레스가  $2^A$ 에서  $2^S$ 로 줄어들면서 연속적인 데이터가 근사값으로 줄어든다. 그림 5(c)는 연속적인 데이터와 근사값 사이의 에러 패턴들을 보여준다. 같은 에러 패턴들이 많이 있지만, EP-ROM은 같은 에러 패턴들을 반복적으로 저장하지 않는다. 기존의 QE-ROM은 같은 에러 패턴의 존재와 상관없이 모든 에러를 저장하지만, EP-ROM은 그림 5(d)에서와 같이 각 에러 패턴들은 다른 ROM에 저장을 하고 에러 패턴을 이용하기 위한 인덱스를 근사값들과 함께 저장을 한다. EP-ROM은 에러가 아닌 에러 패턴의 인덱스만을 저장하기 때문에, ROM의 크기가 QE-ROM에 비하여 크게 줄어든다.

그림 6(a)는 원본 데이터와 EP-ROM에 저장되는 데이터를 보여준다.  $2^A \times L$ 비트의 데이터는  $2^S \times L$ 비트의 근사값과,  $2^S \times I$ 비트의 인덱스, 그리고  $2^{A-S+1} \times E$ 비트의 에러 패턴으로 나누어진다. 이때 근사값을 위한 어드레스가  $2^S$ 이고, 인덱스 수와 에러를 표현하기 위한 비트가 각각 I와 E이다. 원본 데이터의 A비트의 어드레스 중 MSB S비트는 근사값을 위한 어드레스로 사용되고, 나머지

LSB A-S비트는 에러 패턴을 위한 어드레스로 사용된다.

그림 6(b)는 EP-ROM 구조를 보여준다. EP-ROM은 두 개의 ROM과 하나의 덧셈기로 구성된다. 근사값과 인덱스는 같은  $2^S$ 의 어드레스를 가지기 때문에, 하나의 ROM에 같이 저장된다. 따라서, ROM의 크기는  $2^S \times (L+I)$ 비트가 된다. 하드웨어 구현에서 섹션 수  $2^S$ 는 ROM의 워드라인 수  $2^W$ 를 초과할 수 없다. 에러패턴은 다른 ROM에 저장되는데 각 에러 패턴은  $2^{A-S}$ 의 어드레스와 E비트의 값을 가지기 때문에, 각 에러패턴의 크기는  $2^{A-S} \times E$ 비트이다. 에러패턴을 찾는 인덱스는 I비트이므로, ROM은  $2^I$ 개의 에러패턴을 저장하고 있다. 실제로 저장되어야 하는 에러패턴의 수가  $2^I$ 보다 작은 경우에도 A-S+I비트 어드레스에 의하여 ROM의 크기가 결정된다. 근사값과 에러를 더하기 위하여 사용되는 덧셈기는 원본 데이터를 출력한다.

아래의 알고리즘으로 최적의 S, I, 그리고 E의 값을 찾으면 EP-ROM의 크기  $2^S \times L + 2^{A-S+I} \times E$ 를 최소화 할 수 있다.

- (i)  $S = A$ .
- (ii)  $i = 0$ .
- (iii)  $j = 1$ .
- (iv) j 번째 섹션을 위한 L비트의 최소 근사값을 찾는다.
- (v) j 번째 섹션을 위한 에러패턴을 찾는다.
- (vi) 만약 에러패턴이 기존의 에러패턴 리스트에 존재하면, j 번째 섹션의 인덱스는 기존 에러패턴 리스트에서 찾는다. 새로운 에러패턴인 경우에는 에러패턴 리스트에 추가하고 i를 1만큼 증가시킨다.
- (vii) 만약  $j = 2^S$ 이면 (viii)로 가고, 아니면 j를 1만큼 증가시키고 (iv)를 반복한다.
- (viii)  $I = \lceil \log_2 i \rceil$ .
- (ix) EP-ROM의 크기  $2^S \times L + 2^{A-S+I} \times E$ 를 계산한다.
- (x) 만약  $S = W$ 이면 (xi)로 가고, 아니면 S를 1만큼 감소시킨다. (ii) (ix)를 반복한다.
- (xi) (ix)의 결과에서 최적의 S, I, E를 찾는다.

2. 영역 에러 패턴 ROM

(region error pattern ROM:: REP-ROM)

그림 7은 R-ROM과 EP-ROM를 동시에 이용한 제안된 영역 에러패턴 ROM(region error pattern ROM (REP-ROM)의 개념이다. 연속적인 데이터가 R-ROM에 의하여 압축되어 N 개의 R-ROM 블록으로 나누어진다. 각 R-ROM 블록은 EP-ROM에 의하여 다시 압축된다.

그림 8(a)는 R-ROM과 REP-ROM에 저장된 데이터를 보여준다. R-ROM은 ROM의 크기를  $2^A \times L$ 비트에서  $2^R \times M_i$ 비트로 압축한다. R-ROM 압축 후, EP-ROM은

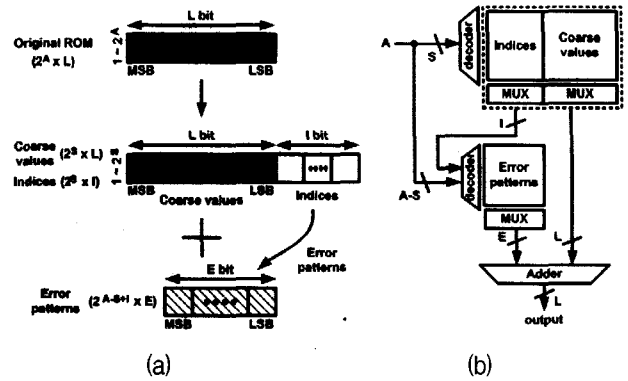


그림 6. (a) 원본 데이터와 EP-ROM에 저장되는 데이터 (b) EP-ROM 구조  
Fig. 6. (a) Original data and data stored in an EP-ROM (b) An EP-ROM architecture.

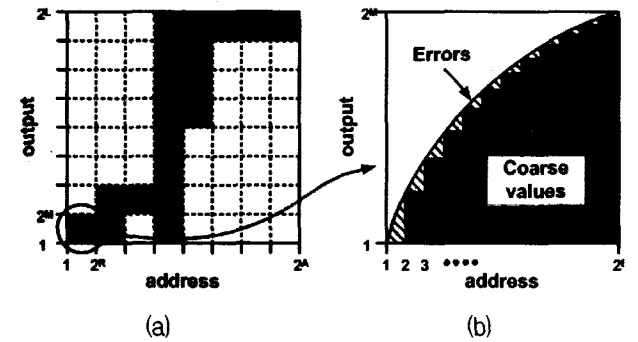


그림 7. (a) R-ROM에 저장되는 영역들 (b) 한 R-ROM 블록에서의 근사값들과 에러들  
Fig. 7. (a) Regions stored in an R-ROM (b) Coarse values and their errors in an R-ROM block.

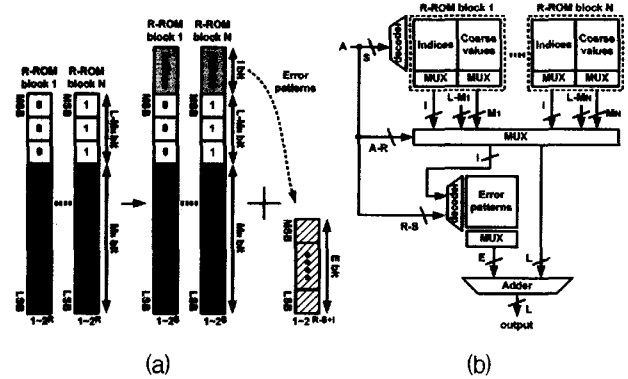


그림 8. a) R-ROM과 REP-ROM에 저장되는 데이터 (b) REP-ROM 구조  
Fig. 8. (a) Data stored in an R-ROM and an REP-ROM (b) An REP-ROM architecture.

R-ROM 블록들의 어드레스를  $2^R$ 에서  $2^S$ 로 줄인다. 그러나 i번째 R-ROM 블록의 출력은 인덱스가 추가되기 때문에  $M_i$ 비트에서  $M_i+I$ 비트로 증가한다. 따라서, i번째 R-ROM 블록의 크기는  $2^R \times M_i$ 비트에서  $2^S \times (M_i+I)$ 비트로 줄어든다. 모든 R-ROM 블록들은 에러패턴을 공유한다. 각 에러패턴이  $2^{R-S}$ 어드레스에 E비트 값을 가

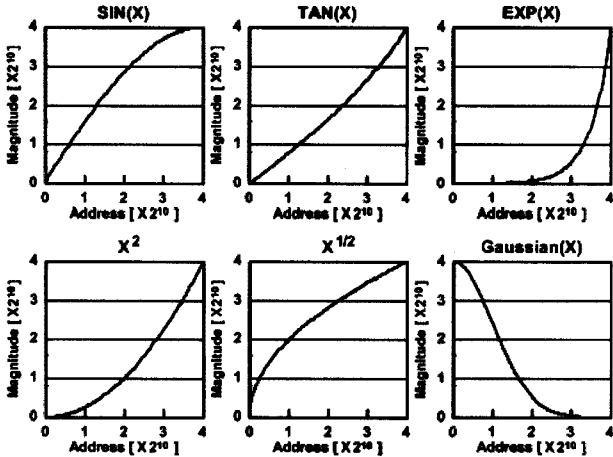


그림 9. 시뮬레이션에 사용된 연속적인 데이터  
Fig. 9. Simulated continuous data.

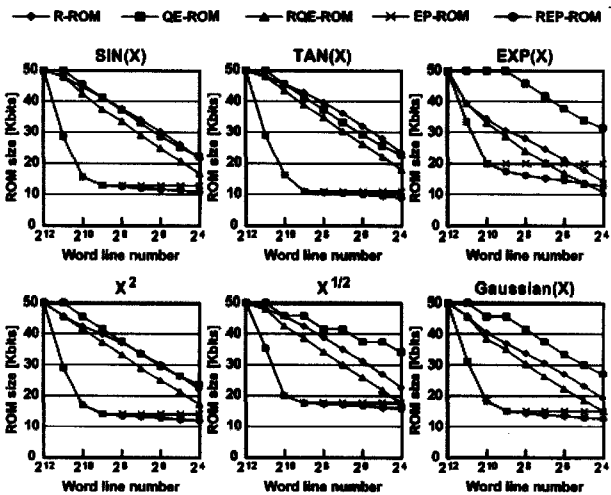


그림 10. 연속적인 데이터의 ROM 크기  
Fig. 10. ROM sizes of continuous data.

지고 인덱스는 I비트일 때, 에러패턴을 저장하고 있는 ROM의 크기는  $2^{R-S+I} \times E$ 비트가 된다. 따라서, REP-ROM은  $2^A \times L$ 비트에서  $2^S \times (M_i + I) + 2^{R-S+I} \times E$ 비트로 줄어든다. 또한, REP-ROM은 근사값과 에러로부터 원본 데이터를 재구성하기 위하여 한 개의 덧셈기를 사용한다.

아래의 알고리즘으로 최적의 S, I,  $M_i$ , I, 그리고 E의 값을 찾아 REP-ROM의 크기  $2^S \times (M_i + I) + 2^{R-S+I} \times E$ 를 최소화 할 수 있다.

- (i)  $R = A$ .
- (ii)  $i = 1$ .
- (iii) i 번째 R-ROM 블록의 데이터를 저장하기 위한 최소의  $M_i$  를 찾는다.
- (iv) 만약  $i = 2^{A-R}$ 이면 (v)로 가고, 아니면 i를 1만큼 증가시키고 (iii)를 반복한다.
- (v) 모든 R-ROM 블록의 비트 데이터에 대하여 EP-ROM압축을 수행한다. 이때, 에러패턴 리스트는 모든 R-ROM블록에서 서로 공유한다.

표 1. 연속적인 데이터의 ROM 크기 [비트]  
Table 1. ROM sizes for various continuous data [bits].

Function	Original ROM	R-ROM	QE-ROM	RQE-ROM	EP-ROM	REP-ROM
SIN(X)	49152	34176	32960	29370	13824	13184
TAN(X)	49152	35712	32960	30266	11264	10752
EXP(x)	49152	24320	41088	20139	19712	14976
X2	49152	33408	33056	28722	13824	12928
X1/2	49152	34944	41056	30135	17920	17408
Gaussian(X)	49152	30080	37024	25907	14848	13696

표 2. 연속적인 데이터의 ROM의 상대적 크기  
Table 2. Normalized ROM sizes for continuous data.

Function	Original ROM	R-ROM	QE-ROM	RQE-ROM	EP-ROM	REP-ROM
SIN(X)	1	0.70	0.67	0.60	0.28	0.27
TAN(X)	1	0.73	0.67	0.62	0.23	0.22
EXP(x)	1	0.49	0.84	0.41	0.40	0.30
X2	1	0.68	0.67	0.58	0.28	0.26
X1/2	1	0.71	0.84	0.61	0.36	0.35
Gaussian(X)	1	0.61	0.75	0.53	0.30	0.28

- (vi) REP-ROM의 크기  $2^S \times (M_i + I) + 2^{R-S+I} \times E$ 를 계산한다.
- (vii) 만약  $R = W$ 이면 (ix)로 가고, 아니면 R을 1만큼 감소시키고 (ii)를 반복한다.
- (viii) (vi)의 결과를 이용하여 최적의 S, R,  $M_i$ , I, E를 찾는다.

#### IV. 실험 및 비교 결과

기존의 ROM 압축 기법들과 제안된 ROM 압축 기법을 그림 9의 다양한 연속적인 데이터에 대하여 실험해 보았다. 실험에 사용된 데이터는 모두  $2^{12} \times 12$ 비트(49152 비트)이다. 그림 10은 R-ROM, QE-ROM, RQE-ROM, EP-ROM, 그리고 REP-ROM에 의하여 압축된 데이터들의 ROM 크기이다. RQE-ROM은 R-ROM과 QE-ROM을 동시에 사용한 압축기법이다<sup>[7]</sup>. EP-ROM과 REP-ROM의 압축률은 기존의 ROM 압축 기법들에 비하여 상당히 높다.

ROM 크기와 상대적인 ROM의 크기는 표 1과 표 2에 정리되었다. 실험결과에서는 EP-ROM과 REP-ROM 의하여 각각 60~77%와 65~78%의 ROM 크기가 감소한 것을 알 수 있다. EP-ROM과 REP-ROM의 압축률은 기존의 ROM 압축률 보다 2배 이상이다.

#### V. 결 론

본 논문에서는 연속적인 데이터를 저장하고 있는 ROM의 크기를 줄이기 위한 새로운 EP-ROM이 제안되었다. EP-ROM은 연속적인 데이터를 근사값과 에러로 나눈 후 에러 대신 에러패턴의 인덱스만을 저장함으

로써 ROM 크기를 줄인다. 이 기법은 두 개의 ROM과 한 개의 덧셈기를 필요로 하지만, ROM의 크기를 상당히 줄여줄 수 있다. 또한 R-ROM과 EP-ROM을 함께 사용한 REP-ROM은 EP-ROM또한 효과적으로 ROM 크기를 줄여준다. 실험결과에서는 EP-ROM과 REP-ROM의하여 각각 60~77%와 65~78%의 ROM 크기가 감소한 것을 알 수 있었다.

### 참 고 문 헌

[1] Edwin de Angel, Earl E. Swartzlander, Jr. "Survey of Low Power Techniques for ROMs," International Symposium on Low Power Electronics and Design, 1997, pp. 7-11.

[2] M. M. Khellah and M. I. Elmasry, "Low-Power Design of High-Capacitive CMOS Circuits Using a New Charge Sharing Scheme," ISSCC Digest of Technical Papers, 1999, pp. 286-287.

[3] Byung-Do Yang and Lee-Sup Kim, "A Low Power ROM using Charge Recycling and Charge Sharing Techniques," IEEE Journal of Solid-State Circuits, vol. 38, no.4, pp. 641-653, Apr. 2003.

[4] Byung-Do Yang and Lee-Sup Kim, "A Low Power Charge Recycling ROM Architecture," IEEE Transactions on Very Large Scale Integration Systems, vol. 11, no. 4, pp. 590-600, Aug. 2003.

[5] J. Vankka, "Methods of Mapping from Phase to Sine Amplitude in Direct Digital Synthesis," IEEE Tr. on ultrasonics, ferroelectrics, and frequency control, pp. 526-534, Mar. 1997.

[6] J. Cao, B. W. Y. Wei, J. Cheng, "High-performance architectures for elementary function generation," IEEE Symposium on Computer Arithmetic, 2001, pp. 136-144.

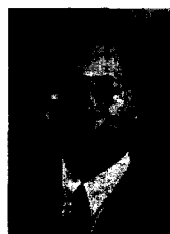
[7] Byung-Do Yang and Lee-Sup Kim, "A ROM compression method for continuous data," Custom Integrated Circuits Conference, 2002, pp. 119-122.

### 저 자 소 개



양 병 도(정회원)  
 1999년 한국과학기술원 전자전산학과 전기및전자공학 전공 졸업 (공학사).  
 2001년 한국과학기술원 전자전산학과 전기및전자공학 전공 졸업 (공학석사).

2001년 ~ 현재 한국과학기술원 전자전산학과 전기및전자공학 전공 박사과정.  
 <주관심분야: 저전력/고성능 디지털 회로 설계, 저전력 메모리 설계>



김 이 섭(정회원)  
 1982년 서울대학교 전자공학과 학사.  
 1986년 Stanford University 전자공학과 석사.  
 1990년 Stanford University 전자공학과 박사.

1990년 ~ 1993년 Toshiba Corporation 연구원.  
 1993년 ~ 현재 한국과학기술원 전자전산학과 전기및전자공학 전공 교수.  
 <주관심분야: 3D Graphics 프로세서 및 하드웨어 설계, 고성능/저전력 디지털 IC 설계>