

論文 2003-40SD-5-8

## 연속된 데이터를 위한 새로운 롬 압축 방식

(A New ROM Compression Method for Continuous Data)

梁炳燾\*, 金利燮\*

(Byung-Do Yang and Lee-Sup Kim)

## 요약

연속된 데이터를 위한 새로운 롬 압축 방식이 제안되었다. 이 롬 압축 방식은 새롭게 제안된 두 롬 압축 알고리즘들을 기반으로 한다. 하나는 영역 선택 롬 압축 알고리즘이다. 이 방식은 크기와 주소로 여러 영역들로 나눈 후 데이터를 포함하는 영역들만을 선택적으로 저장하는 방식이다. 다른 하나는 양자화 롬과 오차 롬 압축 알고리즘이다. 이 방식은 양자화된 데이터와 양자화에 의한 오차를 나누어 저장하는 방식이다. 이 두 알고리즘을 사용하면 다양한 연속된 데이터들에 대하여 40~60%의 롬 크기의 감소를 얻을 수 있다.

## Abstract

A new ROM compression method for continuous data is proposed. The ROM compression method is based on two proposed ROM compression algorithms. The first one is a region select ROM compression algorithm that stores only regions including data after dividing data into many small regions by magnitude and address. The second is a quantization ROM and error ROM compression algorithm that divides data into quantized data and their errors. Using these algorithms, 40~60% ROM size reductions are achieved for various continuous data.

**Keywords** : digital, low power, ROM, compression, continuous data

## I. 서론

집적도가 높아지면서 VLSI 칩들은 대용량의 데이터를 가진 롬(read only memory: ROM)들을 포함하게 되었다. 특히, 저전력 응용 제품들과 신호 처리 프로세서들에는 많은 롬들이 사용되고 있으며 그 크기 또한 지속적으로 커지고 있다. 큰 롬은 넓은 면적을 차지할 뿐만 아니라 전력 소모도 상당히 크다. 따라서, 롬의 크

기와 소모 전력을 줄이기 위한 많은 기법들이 제안되었다. 대부분의 기법들은 롬의 크기의 변화 없이 롬의 전력을 줄이기 위한 시도들이었다<sup>[1-4]</sup>. 다른 한편으로는 수학적 압축 기법을 이용하여 롬의 크기 자체를 줄이기 위한 방법들이 제안되었다<sup>[5,6]</sup>. 롬에 저장되는 데이터가 수학적으로 표현 가능한 함수 일 때, 수학적 방식으로 단순화함으로써 기존의 데이터를 압축한다. 롬에 저장되는 모든 새로운 데이터들은 수학적으로 표현되어야 하며 각 데이터를 위한 새로운 수학적 압축 방식을 만들어야 한다. 따라서, 이러한 기법들을 사용하여 수학적으로 표현하기 어려운 함수들을 압축하기 힘들다.

연속된 데이터들은 신호처리 프로세서와 같은 칩들

\* 正會員, 韓國科學技術院 電子電算學科

(Dept. of Electrical Engineering &amp; Computer Science, KAIST)

※ 본 연구는 한국과학재단과 미세정보센터의 지원을 받아 수행되었습니다.

接受日: 2002年4月4日, 수정완료일: 2003年4月30日

에서 널리 쓰인다. 신호처리를 위한 연속 데이터들은 보통 그 크기가 상당히 크다. 또한, 기존의 수학적 압축 방식들은 이러한 연속 데이터들에는 적당하지가 않다. 왜냐하면 대부분의 연속 데이터들을 단순한 수학 함수로 표현하기 힘들고 수학적으로 그 크기를 줄일 수 있는 방법을 찾기도 힘들다. 따라서, 이와 같이 수학적으로 그 크기를 줄이기 힘든 많은 연속 데이터들을 압축하여 저장할 수 있는 새로운 롬 압축 방식이 필요하다. 본 논문에서는 연속된 데이터를 압축하기 위한 새로운 롬 압축 방식이 제안되었다. 제안된 롬 압축 방식은 새롭게 제안된 두 롬 압축 알고리즘들을 기반으로 한다. 하나는 영역 선택 롬 압축 알고리즘이고 다른 하나는 양자화 롬과 오차 롬 압축 알고리즘이다. 이 두 알고리즘들을 사용한 롬 압축 방식은 임의의 연속된 데이터에 잘 적용되어 롬의 크기를 상당히 줄일 수 있다.

## II. 제안된 롬의 개념

### 1. 영역 선택 롬

<그림 1>은 제안된 영역 선택 롬의 기본 개념을 보여준다. 임의의 연속된 데이터는 수학적 압축 방식을 사용하여 그 크기를 줄일 수 없기 때문에, 기존의 롬에서는 이런 임의의 연속된 데이터들을 압축 없이 저장한다. 제안된 롬 압축 방식에서는 임의의 연속된 데이터를 압축하기 위하여 수학적 접근 방식을 사용하지 않고 연속 데이터들이 가지고 있는 특성인 연속성만을 사용한다. 연속된 데이터는 주소(address)가 증가함에 따라 그 값이 연속적으로 변한다는 성질만을 이용한다.

<그림 1(a)>에서와 같이 기존 데이터를 저장하기 위해 필요한 영역을 데이터의 크기와 주소로 여러 작은 영역들로 나눈다. 이때, 데이터를 포함하는 영역은 매우 적고 대부분의 영역들이 비어 있게 된다. 영역 선택 롬은 데이터를 포함한 영역들만을 선택적으로 저장함으로써 영역 선택 롬의 크기를 기존의 롬에 비하여 상당히 줄일 수 있다. <그림 1(b)>는 영역 선택 롬 안에 저장될 영역들만을 보여준다. 영역 선택 롬 안에 포함되는 몇몇의 선택된 영역들은 비어 있게 되는데, 그 이유는 몇몇의 영역을 저장하기 위하여 사용되는 비트들이 데이터를 포함하지 않는 몇몇 다른 영역들도 포함하기 때문이다.

<그림 2>는 최소 영역 선택 과정의 한 예를 보여준다.

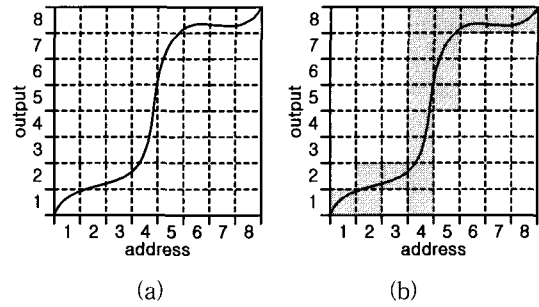


그림 1. (a) 연속 데이터 (b) 영역 선택 롬에 저장되는 영역들

Fig. 1. (a) A continuous data (b) Regions stored in a region select ROM (R-ROM).

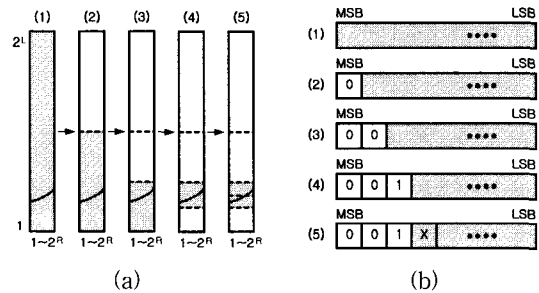


그림 2. (a) 최소 영역 선택 과정 (b) 선택된 영역의 데이터를 저장하기 위한 최소 비트

Fig. 2. (a) Minimum region select sequence (b) Minimum bits for storing the data in the selected region.

다. <그림 1>의 세 번째 컬럼(column)이 그 과정을 설명하기 위한 예로 사용되었다. (1) 컬럼은  $2^L$  주소 가지고 있고 롬 출력이 L비트 일 때 데이터의 범위는 1에서  $2^L$  이다. 우선, 영역 선택 롬 안에 저장되는 데이터는 L비트이다. (2) <그림 2(a)>에서와 같이 컬럼은 크기에 의하여 두 영역으로 나누어진다. 만약 모든 데이터가 한 영역 안에 있다면, 한 영역에 포함된 모든 데이터의 MSB(the most significant bit)는 같다. <그림 2(a)>에서의 모든 데이터가 나누어진 두 영역들 중 낮은 영역에 있기 때문에, <그림 2(b)>와 같이 모든 데이터의 MSB는 '0'이 된다. (3) 데이터를 포함하고 있는 영역은 다시 둘로 나누어진다. 두 번째로 나누어진 두 영역들 중 낮은 영역에 모든 데이터가 있기 때문에, 모든 데이터의 두 번째 MSB도 '0'이다. (4) 모든 데이터가 (3)에서 선택된 영역에서 다시 나누어진 영역들 중에서 높은 영역에 있기 때문에, 세 번째 MSB는 '1'

이다. (5) 다시 둘로 나누어진 영역들 중에서 몇몇의 데이터들은 '0'이고 다른 몇몇의 데이터들은 '1'이기 때문에, 네 번째 MSB의 값은 하나로 정해지지 않는다. 첫 번째 MSB부터 그 값을 정하기 시작하여 정해지지 않는 비트가 처음 나타날 때 최소 영역을 선택하는 과정은 끝난다. 최소 영역은 주소에 의하여 나누어진 영역 내에서 정해지지 않는 비트를 포함하는 영역이다. 주소에 의하여 나누어진 구간 내에서 항상 일정한 데이터를 가지는 비트들은 영역 선택 롬 안에 저장될 필요가 없다. 롬에서 데이터를 얻을 때, 영역 선택 롬에서 저장된 비트들에 고정된 비트들이 더해져서 기존의 L 비트 데이터가 만들어 진다.

나누어진 한 구간의 주소( $2^R$ )는 롬 레이아웃(layout) 조건에 의하여 결정된다. 주소의 수가 감소함에 따라, 영역 선택 롬의 압축 비율은 증가한다. 그러나, 롬의 워드라인의 수가 한 영역의 주소 수 보다 클 수 없기 때문이다. 한 영역의 주소 수의 감소는 워드라인(word line)의 최대 수를 제한한다. 또한, 워드라인 수의 감소는 비트라인(bit line)의 수를 증가시키기 때문에 워드라인 길이를 증가시킨다. 긴 워드라인은 롬의 속도를 감소시키기 때문에, 한 영역의 주소의 수는 롬의 속도, 면적, 그리고 레이아웃 요구에 의하여 조정된다.

## 2. 양자화 롬과 오차 롬

영역 선택 롬 알고리즘을 사용하여 압축된 데이터를 더욱 압축하기 위하여, <그림 3>에서와 같이 양자화 롬(quantization ROM)과 오차 롬(error ROM)으로 나누어 저장함으로써 롬의 크기를 줄인다.

양자화 롬은 저장할 데이터를  $q$ 비트로 양자화하여 데이터의 전체 비트들 중 MSB쪽  $q$ 비트만을 저장한다. <그림 3(b)>에서와 같이,  $2^R$  주소를  $2^S$  섹션(section)으로 균등하게 나누고 각 섹션마다 하나의  $q$ 비트 양자화된 데이터만을 양자화 롬에 저장된다. 따라서, 양자화 롬의 주소는  $2^R$ 에서  $2^S$ 로 줄어들어 양자화 롬의 크기는  $2^S \times q$ 비트가 된다.

오차 롬은 원본 데이터와 양자화 롬의 데이터 사이의 오차들을 저장한다. 오차의 크기가 원본 데이터의 크기보다 매우 작기 때문에 오차를 저장하기 위해 필요한 비트 수는 원본 데이터를 저장하기 위하여 필요한 비트 수보다 작게 된다. 모든 오차들을 저장하기 위해 필요한 최소 비트 수가  $e$ 비트라고 할 때, 오차 롬의 크기는 비트가  $2^R \times e$  된다.

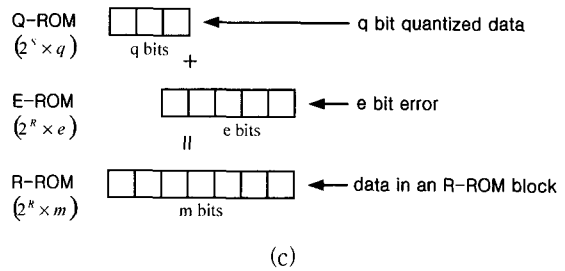
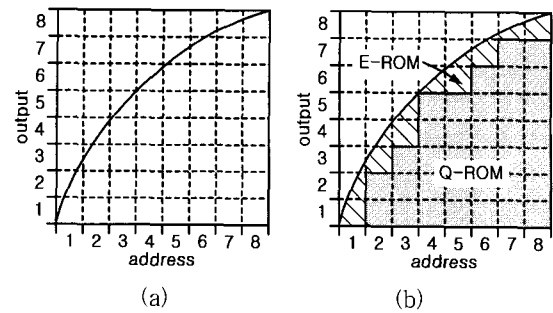


그림 3. (a) 한 영역 선택 롬 블록 안에 저장된 데이터 (b) 양자화 롬과 오차 롬에 저장된 데이터 (c) 롬에서 필요한 비트

Fig. 3. (a) Data in an R-ROM block (b) Data in Quantization ROM (Q-ROM) and Error ROM (E-ROM) (c) Required bits for ROMs.

따라서, 양자화 롬과 오차 롬 합은  $2^S \times q + 2^R \times e$  비트가 된다. 연속된 데이터들에 대하여, 이 오차들은 매우 작게 되므로, 양자화 롬과 오차 롬의 크기는 원본 데이터를 저장하기 위해 필요한 롬의 크기 보다 작게 된다.

<그림 3>은 섹션 수  $2^S$ 가 8이고, 양자화 비트  $q$ 가 3, 그리고 오차 비트  $e$ 가 5인 경우의 예이다.

## III. 제안된 롬의 구조

<그림 4(a)>는 한 영역 선택 롬 블록 안의 양자화 롬과 오차 롬에 저장된 데이터의 예를 보여준다. 양자화 롬과 오차 롬으로부터  $m$ 비트 출력을 만들기 위하여, 각 영역 선택 롬 블록마다  $m$ 비트 덧셈기가 필요할 것이다. 그 이유는 각 영역 선택 롬 블록의 출력인  $m$ 비트가 각 블록마다 다르기 때문이다. 그러나, 각 영역 선택 롬의 출력 비트는 원본 데이터 L비트를 넘지 않기 때문에, <그림 4(c)>와 같이 하나의 L비트 덧셈기만을 사용하여 각 영역 선택 롬 블록마다 필요한 덧셈기들을 대체할 수 있다. 하나의 L비트 덧셈기만을 사용하기 위해서, 각 영역 선택 롬 블록의 출력들을 두개

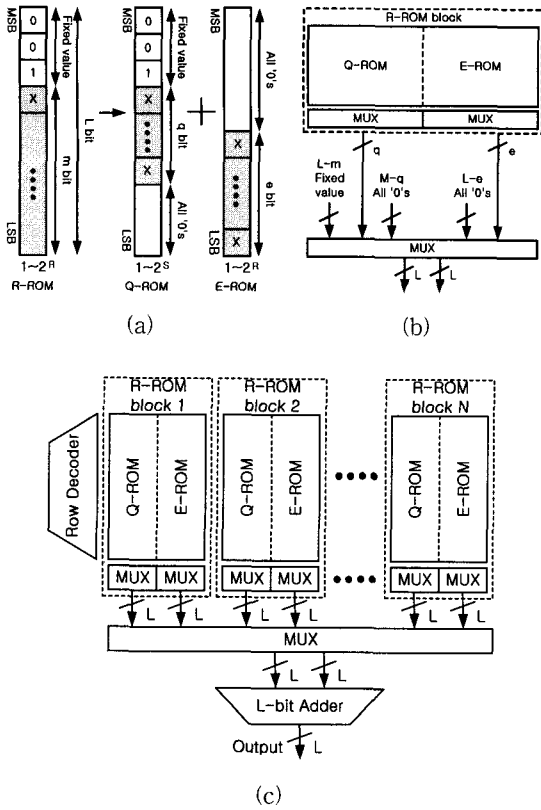


그림 4. (a) 한 영역 선택 롬 블록 안의 양자화 롬 데이터와 오차 롬 데이터 (b) 한 영역 선택 롬 블록의 구조 (c) 제안된 롬 구조

Fig. 4. (a) Q-ROM data and E-ROM data stored in an R-ROM block (b) An R-ROM block implementation (c) Proposed ROM architecture.

의 L비트 데이터로 재 구성해야만 한다. 첫 번째 L비트 데이터는 l-m비트 영역 정보, q비트 양자화 롬 데이터, 그리고 LSB(the least significant bit) m-q비트의 '0'으로 구성된다. 두 번째 L비트 데이터는 MSB(the most significant bit) l-e비트의 '0'과 e비트 오차 롬 데이터로 구성된다. <그림 4(b)>와 같이, 양자화 롬과 오차 롬은 하드웨어 구현에서 두개의 롬으로 나누어지지 않는다. 양자화 롬과 오차 롬은 약간의 주소 비트를 공유하기 때문에, 양자화 롬 데이터와 오차 롬 데이터는 하나의 row decoder를 공유하는 하나의 롬에 저장된다. 그러나, 양자화 롬과 오차 롬의 전체 주소 비트는 다르기 때문에, 양자화 롬과 오차 롬에 두개의 다른 column MUX를 사용한다. 따라서, 양자화 롬과 오차 롬으로 나눔으로 인한 면적의 증가는 매우 적다.

<그림 4(c)>는 영역 선택 롬, 양자화 롬, 그리고 오차 롬을 사용한 제안된 롬 구조이다. 압축되지 않은 롬이  $2^A$ 의 주소와 L비트 데이터를 가지고 있을 때, i번째 영역 선택 롬 블록은  $2^R$ 의 주소와  $m_i$ 비트 데이터를 가지고, i번째 영역 선택 롬 블록 안의 양자화 롬과 오차 롬은  $2^{S_i}$ 와  $2^R$ 의 주소와  $q_i$ 와  $e_i$ 의 비트 데이터를 가진다. 가장 작은 롬을 위한 파라미터인 R,  $S_i$ ,  $m_i$ ,  $q_i$ ,  $e_i$ 는 영역 선택 롬 압축 알고리즘과 양자화 롬과 오차 롬 압축 알고리즘에 의하여 결정된다. 모든 영역 선택 롬 블록들의 두 L비트 출력들은 두 번째 MUX에 들어가고, 두 번째 MUX는 많은 영역 선택 롬 블록들 중에서 한 블록을 선택한다. 두 번째 MUX의 두 L비트 출력은 하나의 L비트 덧셈기에 의하여 더해진다. 덧셈기에 의한 시간 지연을 최소화 하기 위해서 고속의 덧셈기가 요구된다. 만약 덧셈기의 연산이 롬 비트라인의 precharge 시간 안에 수행 된다면, 덧셈기에 의한 시간 지연은 전체 속도에 영향을 주지 않을 것이다.

하드웨어 구현에서, 영역 선택 롬 블록들은 하나의 롬에서 구현된다. 모든 영역 선택 롬 블록들은 하나의 row decoder를 공유하며, 워드라인의 수는 속도와 면적의 요구에 의하여 결정된다. 양자화 롬의 섹션 수는 워드라인의 수를 넘을 수 없기 때문에, 워드라인 수는 양자화 롬의 섹션 수를 제한하게 된다.

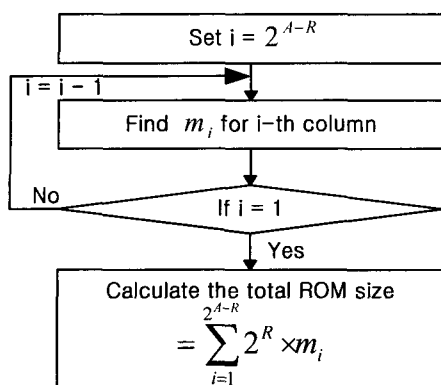
#### IV. 알고리즘

##### 1. 영역 선택 롬의 압축 알고리즘

영역 선택 롬의 전체 크기( $\sum 2^R \times m_i$ )를 최소화 하기 위하여, 우리는 각 i번째 영역 선택 롬 블록에 요구되는 최소 비트( $m_i$ )를 찾아야만 한다. 영역 선택 롬의 주소( $2^R$ )는 요구되는 워드라인의 최소 수( $2^W$ )와 같다. 파라미터  $m_i$ 는 <그림 5>의 알고리즘을 사용하여 찾을 수 있다.

##### 2. 양자화 롬과 오차 롬의 압축 알고리즘

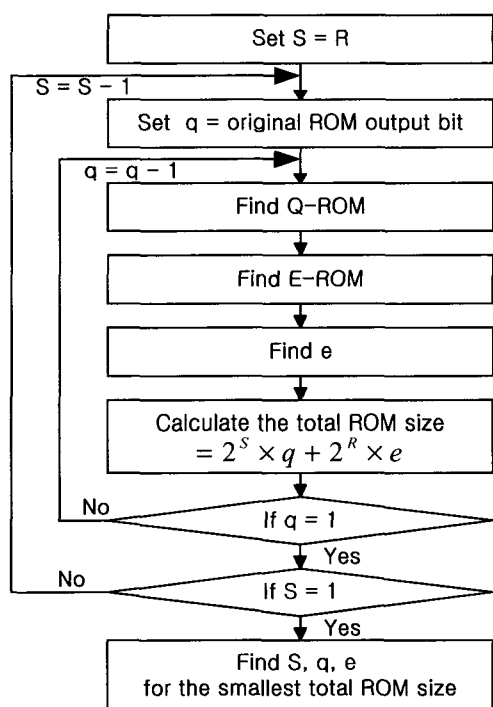
양자화 롬과 오차 롬의 전체 크기 ( $2^S \times q + 2^R \times e$  bits)를 최소화 하기 위하여, 우리는 최적의 섹션 수( $2^S$ ), 양자화 롬의 양자화 비트 수(q), 그리고 오차 롬의 모든 오차 값들을 저장하기 위해 필요한 최소 비트 수(e)를 찾아야 한다. 파라미터 S, q, e는 <그림 6>의 알고리즘을 사용하여 찾을 수 있다.



- (i) Set  $i = 2^{A-R}$  when the address of the original ROM is  $2^A$ .
- (ii) Find  $m_i$  the unfixed bit for all data of  $i$ th R-ROM block.
- (iii) If  $i = 1$ , then go to (iv) else decrease  $i$  and repeat (ii).
- (iv) Calculate the total ROM size ( $\sum 2^R \times m_i$ ).

그림 5. 영역 선택 롬 압축 알고리즘

Fig. 5. R-ROM compression algorithm.



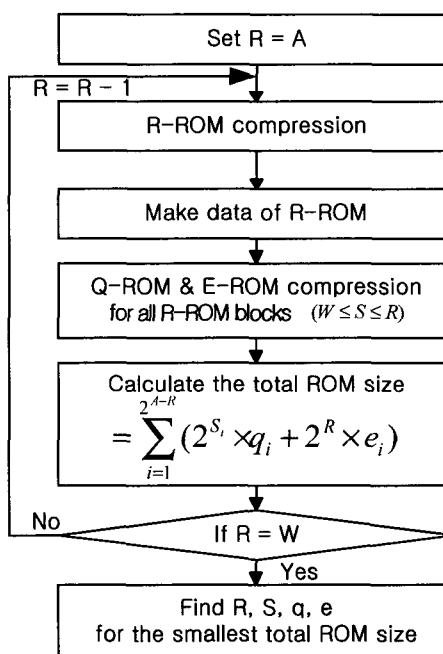
- (i) Set  $S = R$  when the section number is  $2^S$  and the original ROM address is  $2^R$ .
- (ii) Set  $q$  = the output bit of the original ROM.
- (iii) Find the minimum  $q$  bit quantized data for each section to be stored in Q-ROM.
- (iv) Find the errors between the original ROM and

the Q-ROM to be stored in E-ROM.

- (v) Find  $e$  which is the minimum bit to represent all errors.
- (vi) Calculate the total ROM size ( $2^S \times q + 2^R \times e$  bits).
- (vii) If  $q = 1$ , then go to (viii) else decrease  $q$  by 1 and repeat (iii) - (vi).
- (viii) If  $S = 1$ , then go to (ix) else decrease  $S$  by 1 and repeat (ii) - (vii).
- (ix) Find the optimal  $S, q, e$  by using the results at (vi).

그림 6. 양자화 롬과 오차 롬의 압축 알고리즘

Fig. 6. Q-ROM and E-ROM compression algorithm.



- (i) Set  $R = A$  when the address of the original ROM is  $2^A$ .
- (ii) Do the R-ROM compression.
- (iii) Make data of R-ROM using parameters from (ii).
- (iv) Do the Q-ROM and E-ROM compression for all R-ROM blocks when  $W \leq S \leq R$ .
- (v) Calculate the total ROM size ( $\sum 2^{S_i} \times q_i + 2^R \times e_i$  bits).
- (vi) If  $R = W$ , then go to (vii) else decrease  $R$  by 1 and repeat (ii) - (v).
- (vii) Find the optimal  $R, S_i, q_i$ , and  $e_i$  by using the results at (v).

그림 7. 연속된 데이터를 위한 롬 압축 알고리즘

Fig. 7. ROM compression algorithm for continuous data.

3. 연속된 데이터를 위한 제안된 롬 압축 알고리즘  
 연속된 데이터를 압축 저장하기 위하여 제안된 롬 압축 알고리즘은 영역 선택 롬 압축 알고리즘과 양자화 롬과 오차 롬 압축 알고리즘을 기반으로 한다.  $i$  번째 영역 선택 롬 블록의 데이터 비트가  $m_i$ , 그 블록 안의 양자화 롬과 오차 롬의 주소가  $2^S$ 와  $2^R$ 이고 데이터 비트가  $q_i$ 와  $e_i$ , 그리고 워드라인 수는  $2^W$ 일 때, 전체 롬의 크기를 줄이기 위한 파라미터  $R, S, m_i, q_i, e_i$ 의 최적 값을 찾아야 한다. 이 파라미터들은 <그림 7>의 알고리즘을 사용하여 찾을 수 있다.

V. 실험 결과와 성능 비교

<표 1>은 다양한 연속 데이터에 대한 롬의 크기 비교 결과를 보여준다. 압축되지 않은 롬, 영역 선택 롬 압축 알고리즘을 사용한 롬, 양자화 롬과 오차 롬 압축 알고리즘을 사용한 롬, 그리고 두 압축 알고리즘을 모두 사용한 롬의 크기가 표로 정리 되었다.

모의 실험에 사용된 연속 데이터들의 크기는 모두  $2^{12} \times 12$  비트(49152 비트)이다. <그림 8>은 실험에 사용된 연속 데이터들이다. <그림 9>는 워드라인 수( $2^W$ )에 따른 롬 크기를 보여준다. 워드라인의 수가 감소함에 따라, 롬의 크기는 감소한다.

<표 1>은 워드라인의 수가 128일 때, 롬의 크기를 보여 준다. 워드라인 수는 롬이 최고 속도를 가질 경우를 기준으로 선택 되었다. 같은 크기의 롬에서 워드라인의 수를 줄이면 워드라인에 연결되는 비트라인의 수가 증가하여 롬의 속도가 감소한다. 그러나 워드라인의 수가 감소함에 따라, 제안된 롬 압축 알고리즘에 의한 압축 비율이 증가하기 때문에 롬의 면적은 감소한다. 따라서, 워드라인 수는 요구되는 속도와 면적에 의하여 결정 된다. 실험 결과는 영역 선택 롬 (R-ROM), 양자화 롬과 오차 롬 (QE-ROM), 그리고 이 두 압축 알고리즘을 모두 사용한 롬 (C-ROM)이 각각 27~50%, 16~32%, 그리고 40~60% 압축 됨을 보여준다.

VI. 결론

제안된 롬 압축 방식은 연속된 데이터를 저장하는 롬의 크기를 줄여준다. 제안된 영역 선택 롬 압축 알고리즘은 주소와 크기로 기존의 데이터를 저장하는 영역을 많은 수의 작은 영역들로 나눈 후, 데이터를 가진

표 1. 다양한 연속 데이터들에 대한 롬 크기 비교(워드라인 수가 128일 때) [비트]

Table 1. Size comparison of ROMs for various continuous data (when word line number is 128) [bits]

Function	Original	R-ROM	QE-ROM	C-ROM
SIN(X)	49152	34176	32960	29370
TAN(X)	49152	35712	32960	30266
EXP(x)	49152	24320	41088	20139
$X^2$	49152	33408	33056	28722
$X^{1/2}$	49152	34944	41056	30135
Gaussian(X)	49152	30080	37024	25907

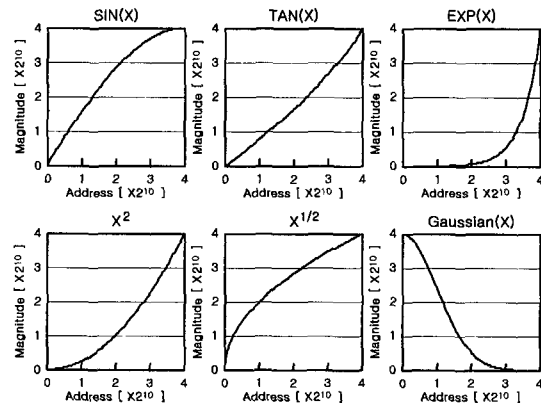


그림 8. 다양한 연속 데이터들  
 Fig. 8. Various continuous data.

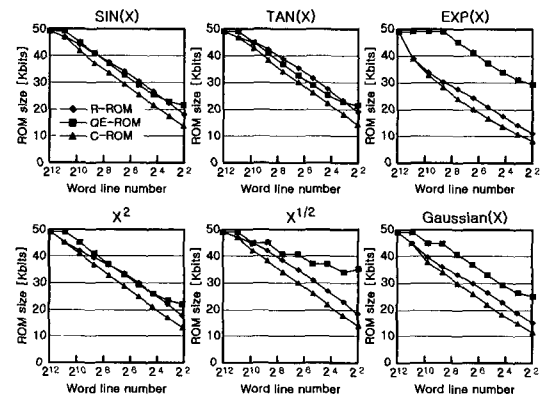


그림 9. 워드라인 수에 따른 다양한 연속 데이터들의 롬 크기  
 Fig. 9. ROM sizes of various continuous data for the word line number.

영역들 만을 저장함으로써 롬을 압축하는 알고리즘이다. 제안된 양자화 롬과 오차 롬 압축 알고리즘은 기존 데이터를 양자화 시켜 양자화된 데이터와 양자화에 의한 오차를 나누어서 저장함으로써 롬을 압축하는 알고리즘이다. 이 두 알고리즘을 사용하여 다양한 연속 데이터를 저장하기 위한 롬 크기를 40%~60% 줄일 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, pp. 286~287, Feb. 1999.
- [3] Byung-Do Yang and Lee-Sup Kim, "A Low Power Charge-Recycling ROM Architecture", IEEE International Symposium on Circuits and Systems, 2001, pp. 510~513.
- [4] Byung-Do Yang and Lee-Sup Kim, "A Low-Power ROM using Charge Recycling and Charge Sharing", ISSCC Digest of Technical Papers, Feb. 2002, in press.
- [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.

### 저 자 소 개



梁炳燾(正會員)

1999년 2월 : 한국과학기술원 전자전산학과 전기및전자공학 전공 졸업(공학사). 2001년 2월 : 한국과학기술원 전자전산학과 전기 및 전자공학 전공 졸업(공학석사). 2001년 3월~현재 : 한국과학기술원 전자전산학과 전기 및 전자공학 전공 박사과정. <주관심분야 : 저전력/고성능 디지털 회로 설계, 저전력 메모리 설계>



金利燮(正會員)

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>