

# 고속 프랙탈 영상압축을 위한 최적의 파이프라인 주기를 갖는 VLSI 어레이 구조 설계

정회원 성길영\*, 우종호\*\*

## Design of VLSI Array Architecture with Optimal Pipeline Period for Fast Fractal Image Compression

Kil-Young Sung\* , Chong-Ho Woo\*\* *Regular Members*

### 요 약

본 논문에서는 프랙탈 영상압축의 고속수행을 위한 최적의 파이프라인 주기를 갖는 일차원 VLSI 어레이를 설계했다. 고정분할 알고리즘을 변형하여 VLSI 어레이 설계에 적합하며 화질의 손상을 최소화하면서 압축율이 높은 알고리즘을 유도했다. 파이프라인의 각 세그먼트를 구성하는 PE의 연산시간을 가능한 균등하게 분포시켜 최적의 파이프라인의 주기를 얻었다. 이러한 결과로써 약 3배의 속도 향상을 얻을 수 있다. 정의역과 치역블럭의 임플렉과 연산장치를 공유하여 임플렉 편의 수를 줄였다.

### ABSTRACT

In this paper, we designed one-dimensional VLSI array with optimal pipeline period for high speed processing fractal image compression. The algorithm is derived which is suitable for VLSI array from fixed block partition algorithm. Also the algorithm satisfies high quality of image and high compression-ratio. The designed VLSI array has optimal pipeline period because the required processing time of PEs is distributed as same as possible. As this result, we can improve the processing speed up to about 3 times. The number of input/output pins can be reduced by sharing the input/output and arithmetic unit of the domain blocks and the range blocks.

### I. 서 론

영상데이터의 저장과 전송의 효율을 높이기 위한 압축방법 중 프랙탈 영상압축은 고압축비, 코드-북(code-book) 불필요, 고속도 복원, 해상도 독립성, 고품질 복호화 영상, 그리고 다른 기술과의 조합 등의 장점을 갖는다<sup>[1-4]</sup>. 그러나 프랙탈 영상압축은 코딩에 많은 시간이 필요한 단점을 가진다<sup>[5]</sup>.

프랙탈 영상압축 알고리즘은 병렬성이 대단히 높으므로 대량의 데이터를 동시에 처리하는 VLSI 어

레이를 구현하여 데이터 처리속도를 크게 향상시킬 수 있다. F. Ancarani는 PCI 버스를 이용하여 고정 분할분할 알고리즘을 수행하는 PC 플랫폼의 병렬 주문형 집적회로(ASIC) 구조를 제안하였다<sup>[6]</sup>. D. J. Jackson은 순환 계산 모델(circulating computation model)을 사용하는 128개의 처리요소로 구성된 nCube 다중처리 시스템에 워드-트리 분할 알고리즘을 시사하였다<sup>[7]</sup>. K. P. Acken은 워드-트리 분할을 위한 SIMD 모델의 ASIC 구조를 제안하였다<sup>[8]</sup>.

본 논문에서는 프랙탈 영상압축의 고속화를 위하여 참고문헌[9]에서 제안한 일차원 시스템릭 어레이

\* 경상대학교 정보통신공학과, 해양산업연구소,

\*\* 부경대학교 컴퓨터멀티미디어공학부

논문번호 : 00081-0306, 접수일자 : 2000년 3월 6일

에서 처리시간의 최적화를 위하여 파이프라인의 세그먼트를 구성하는 각 PE(Processing Element)의 수행시간을 가능한 균등하도록 분산시켜 파이프라인 주기를 최적화하였다. 압축시 영상의 화질을 유지하면서 높은 압축율을 얻기 위해 다양한 블록의 크기에 대해 고정분할 방식의 알고리즘을 반복수행해서 최적의 값을 선택했다. 선택된 s(scaling coefficient)와 o(offset coefficient)를 양자화시켜 화질의 손상을 최소화하면서 압축율을 높였다. 컴퓨터 시뮬레이션을 통하여 설계한 일차원 VLSI 어레이의 동작을 검증하였다.

## II. VLSI 어레이의 설계

프랙탈 영상압축은 영상을 지역 및 정의역 블록으로 나누고 자기유사성을 찾아 데이터의 양을 줄인다. 일반적인 쿼드-트리 분할방식의 알고리즘은 데이터의 입출력이 불규칙적이므로 VLSI 어레이로 구현하기가 적합하지 않다. 본 논문에서는 데이터 입력이 규칙적인 고정분할 방식 알고리즘을 지역블록의 크기에 따라 반복수행하여 쿼드-트리 분할방식을 적용한 효과를 얻는다.

그림 1은  $n \times n$  픽셀의 크기를 갖는 지역블록에 대한 최적 정의역 블록을 선택하기 위한 알고리즘이다. 본 논문에서는  $n$ 이 16, 8, 4 인 경우에 대해 반복 수행하는 VLSI 어레이를 설계한다. 이 알고리즘의 데이터의존관계를 분석하여 데이터의존그래프를 유도하고, 시간 및 공간사상을 통하여 그림 2와

```

for m = 1 to M ;           M: the number of range blocks
  for l = 1 to L ;       L: the number of domain blocks
    s1 = s2 = s3 = s4 = s5 = 0
    for i = 1 to n      n: the number of pixel in one block
      s1 = s1 + dl-1,0
      s2 = s2 + dl
      s3 = s3 + n
      s4 = s4 + dl
      s5 = s5 + n
    next i
    s = (n.s1 - s2.s3) / (n.s4 - s2.s2)
    o = (s3 - s.s2) / n
    mse = (s2.s4 + n.o + s5 + 2s.o.s2 - 2o.s3 - 2s.s1) / n
    if (mse < smse)
      s, o, m, l, mse 선택
    end if
  next l
  
```

그림 1. 프랙탈 영상 압축 알고리즘

같은 일차원 VLSI 어레이의 구조를 얻는다<sup>[10]</sup>.

실제한 VLSI 어레이는 그림 1에서 지역과 정의역 블록의 각 픽셀을 누적하는 A PE, s를 연산하는 S PE, o를 연산하는 O PE와 오차(mse)를 연산하고 최소오차를 갖는 블록을 선택하는 M PE로 구성된다. 16개로 구성된 A PE는 지역과 정의역 블록에 대한 입력핀과 내부연산장치를 공유한다. 따라서 지역과 정의역블록의 픽셀이 번갈아 입력되어 누적된다.  $n$ 이 16 또는 8의 경우는 반복 수행을 통해 연산이 이루어진다. M PE에서 선택된 s와 o는 호스트 컴퓨터로 전송되어 블록의 크기가 큰 쌍을 최적 블록으로 선택한다.

그림 2의 M PE는 다른 PE들에 비해 많은 계산 시간이 요구된다. 파이프라인의 동작 주기는 계산시간이 가장 긴 PE를 기준으로 선택해야하므로 그림2의 VLSI 어레이는 최적의 성능을 발휘하지 못한다.

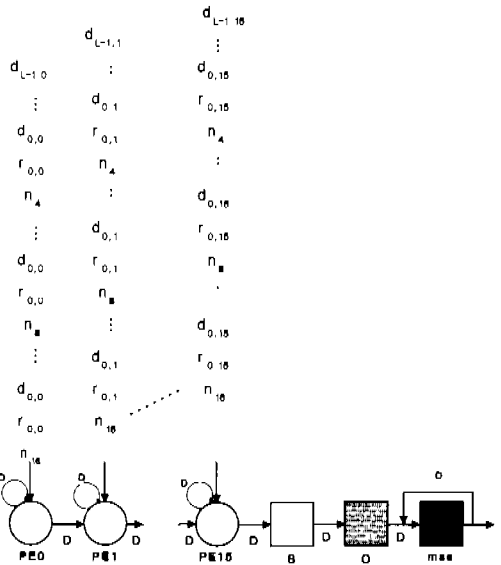


그림 2. 일차원 VLSI 어레이

최적의 성능을 갖는 VLSI 어레이를 유도하기 위해 그림1의 알고리즘을 한번 더 변형하며 데이터의 저장량을 줄이기 위해 s와 o를 각각 5-비트와 7-비트로 양자화시킨다. 그리고 파이프라인을 구성하는 세그먼트의 수행시간을 가능한 균등분배하기 위해 알고리즘을 그림 3과 같이 변형한다.

고속 연산을 위하여 2의 멱수와 곱셈과 나눗셈은 쉬프트연산으로 대체한다. 하나의 곱셈과 덧셈연산

```

for m = 1 to M
  for l = 1 to L
    for i = 1 to n
      PE0      s1 = s1 + r[i]*d[i]
              s2 = s2 + d[i]
              s3 = s3 + r[i]
      PE15     s4 = s4 + d[i]^2
              s5 = s5 + r[i]^2
    next i
    PE18     det = s4 □ ns - s2*s2
              s23 = s2*s3
      PE17     ts = (s1 □ ns - s23)/det
      PE18     s = 0.5 + (ts + 1) □ 5
      PE19     ts = s/16 - 1
              to = (s3 - ts*s2) □ ns
      PE20     fts = (abs(ts)+1) □ 8
              if (ts>0)
                to = to + ts □ 8
      PE21     tmp = to/fts
              o = tmp □ 7 + 0.5 - tmp
              tmp = o*fts
              if (ts>0)
                to = tmp □ 8 - tmp
              else
                to = tmp □ 8 - tmp - ts □ 8
              ts2 = ts ^ 2
              ts4 = s4*ts2
              tss1 = ts*s1 □ 1
              tos2 = ts*to □ 1
              to2 = to ^ 2
              tos3 = to*s3 □ 1
              tmp1 = to2 □ ns
              tmp2 = ts4 + s5
              tmp3 = tos2 * s2
              tmp4 = tss1 + tos3
              tmp5 = tmp1 + tmp2
              tmp6 = tmp3 - tmp4
              mse = tmp5 + tmp6
              if (mse < smse)
                smse = mse
                ss = s
                so = o
              endif
    next i
  next m
  
```

그림 3. 양자화를 삼입한 프랙탈 영상압축 알고리즘

에 필요한 시간을 단위시간으로 파이프라인 주기를 결정한다.

그림 3의 알고리즘으로부터 유도된 VLSI 어레이는 그림 4의 구조를 갖는다.

설계한 VLSI 어레이는 정의역과 치역블럭의 픽셀값을 누적하기 위한 PE0에서 PE15까지 16개의 PE, s를 연산하고 양자화를 위한 PE16, 17, 18과 o를 연산하고 양자화를 위한 PE19, 20, 21, 그리고 mse를 연산하고 최적의 블럭을 선택하기 위한 PE22, 23, 24로 구성된다.

### II. PE 내부의 데이터 흐름

어레이를 구성하는 PE는 호스트 컴퓨터 또는 이전 세그먼트의 PE로부터 데이터를 수신하여 연산하

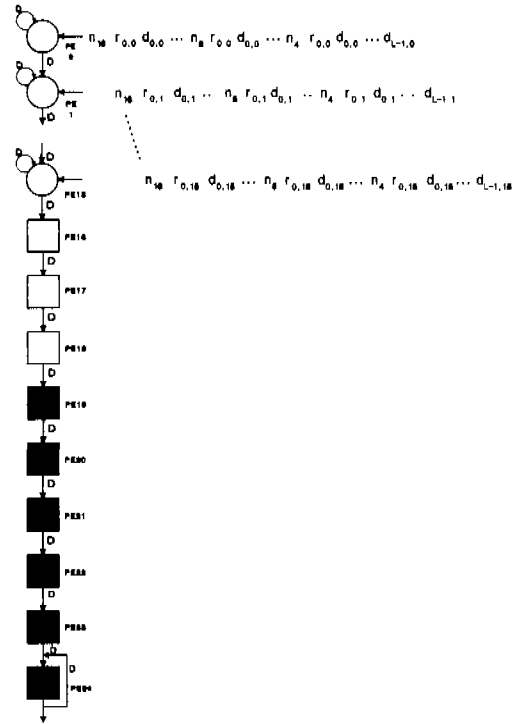


그림 4. 최적의 파이프라인 주기를 갖는 일차원 VLSI 어레이

고, 그 결과를 보관 및 다음 세그먼트로 전달하기 위해 레지스터에 저장한다. 결과의 VLSI 어레이는 시간 및 공간적 지역성을 만족한다.

설계한 VLSI 어레이는 크게 4종류의 PE로 구성된다. 그림 5는 호스트 컴퓨터로부터 수신한 치역 및 정의역의 픽셀을 누적하는 기능을 갖는 PE들이다. 이 PE들은 블럭의 크기와 치역 및 정의역의 픽셀을 입력하기 위한 핀을 공유한다. 호스트 컴퓨터로부터 입력된 데이터는 디멀티플렉서를 통해 해당 레지스터에 저장된다. PE내의 사각형은 레지스터와 연산기를 나타낸다. PE0은 VLSI 어레이의 최상단에 해당하는 PE로서 입력된 데이터를 연산하여 저장한다. PE1부터 PE14까지는 호스트 컴퓨터로부터 입력되는 데이터의 연산결과와 앞단의 PE에서 전달되는 값을 누적한다. 설계한 어레이는 블럭의 크기 n이 4의 경우에 해당한다. 따라서 8과 16의 경우, 반복연산을 통해 그 결과를 얻는다. 그러므로 PE15에 기존의 연산결과를 현재의 연산결과와 누적하는 기능이 추가된다.

그림 6은 s를 연산하기 위한 PE들인데, PE들은 호스트 컴퓨터로부터 직접 데이터를 입력받지는 않

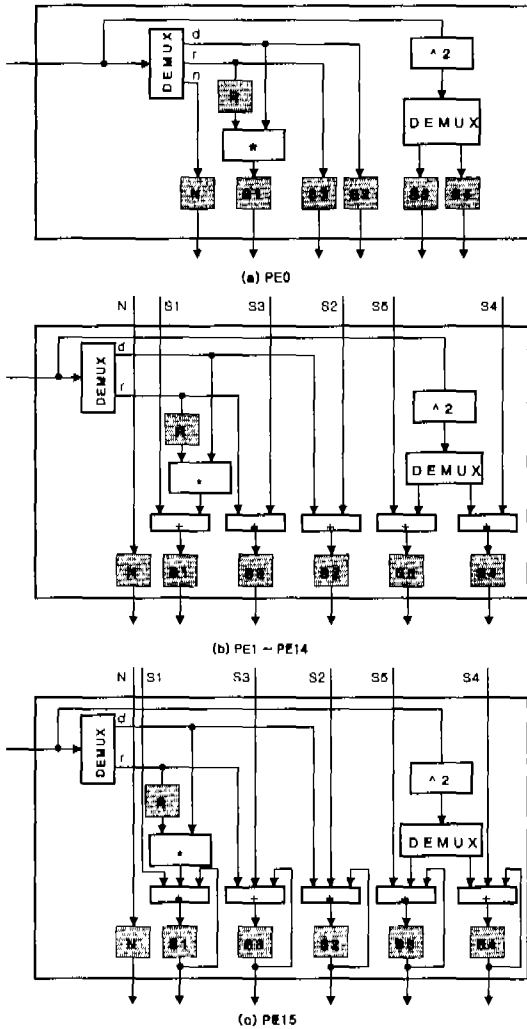


그림 5. 각 픽셀의 누적을 위한 PE들

는다. PE15에서 연산된 결과를 입력받아  $s$ 를 구하고 양자화시킨다. PE17의 TS 레지스터에 계산된 값이  $s$ 이며, PE18의  $s$  레지스터는 양자화된 결과이다. 계산 속도의 향상을 위해 2의 멱수에 대한 곱셈과 나눗셈은 쉬프트연산으로 처리한다.

그림 7은  $o$ 를 계산하고 양자화시키기 위한 PE들이다. PE19는 PE18로부터 입력받은 양자화된  $s$ 를 실수형태로 변환한다. PE20은 이 결과를 이용하여  $o$ 를 계산하고 PE21에서 양자화시킨다.

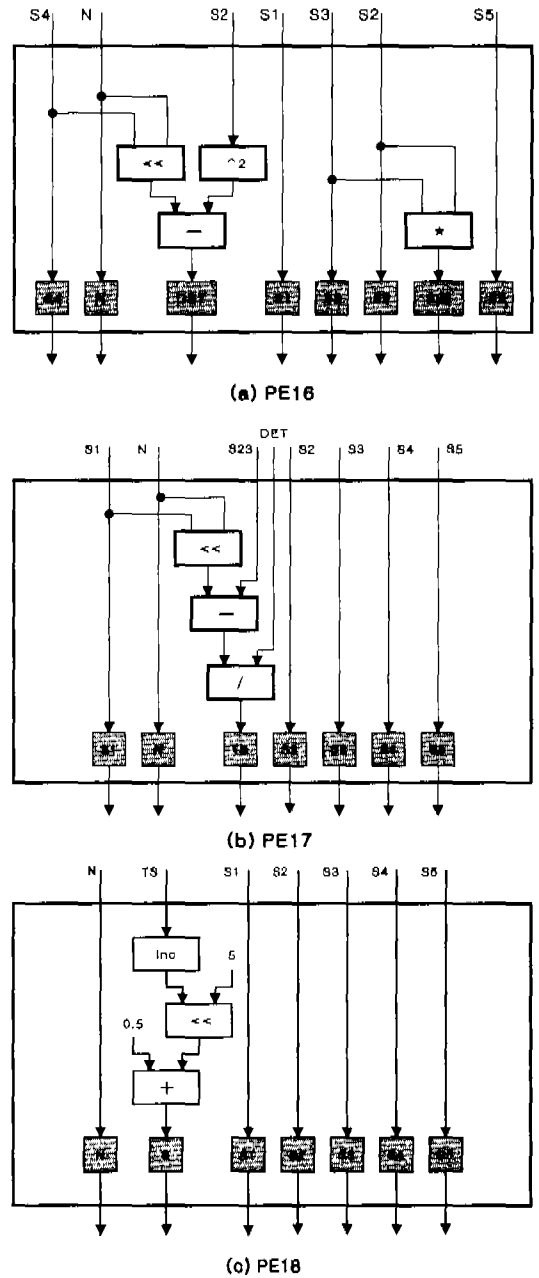


그림 6.  $s$ 를 구하기 위한 PE들

그림 8은  $s$ 와  $o$ 를 이용하여 지역과 정의역블럭의 유사정도인 mse를 계산하기 위한 PE들이다. 또한 각 지역블럭에 대해 최적의 정의역블럭을 선택하고 호스트 컴퓨터로 전송한다.

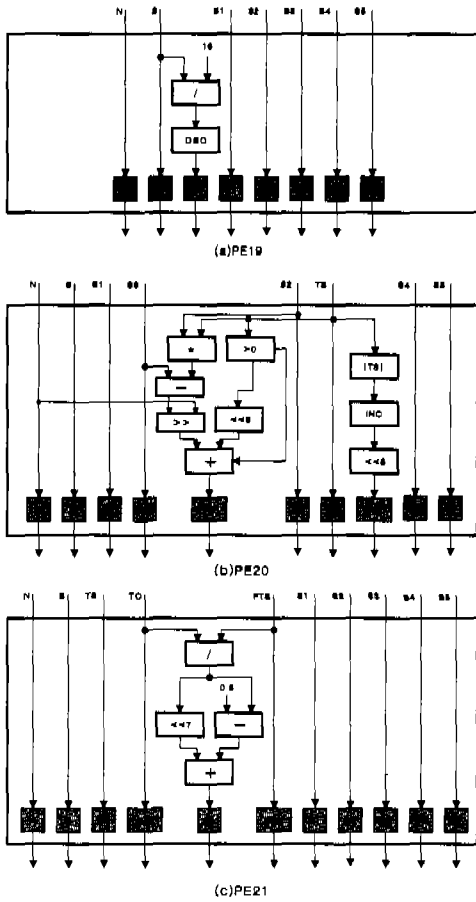


그림 7. o를 구하기 위한 PE들

#### IV. 결과 및 고찰

실제한 일차원 VLSI 어레이의 동작을 검증하기 위하여 64×64 크기의 그레이스케일의 Lena 영상을 이용하여 컴퓨터시뮬레이션을 수행하였다. 그림 9는 C++ 언어로 작성된 시뮬레이터의 4646 단위시간 후의 스냅샷이다. 큰 사각형은 각 PE를 나타내고, PE 내의 작은 사각형은 값을 저장하기 위한 레지스터와 임시 저장되는 데이터를 나타낸다.

최적화되지 않은 그림 3의 구조와 본 논문에서 설계한 최적화된 파이프라인 주기를 갖는 그림 4의 구조를 갖는 VLSI 어레이의 성능을 비교하면 표 1과 같다.  $t_1$ 과  $t_2$ 는 각각 그림 3과 4의 파이프라인 주기이다.  $t_2$ 는  $t_1$ 에 비해 두배 이상 짧다. 두 구조

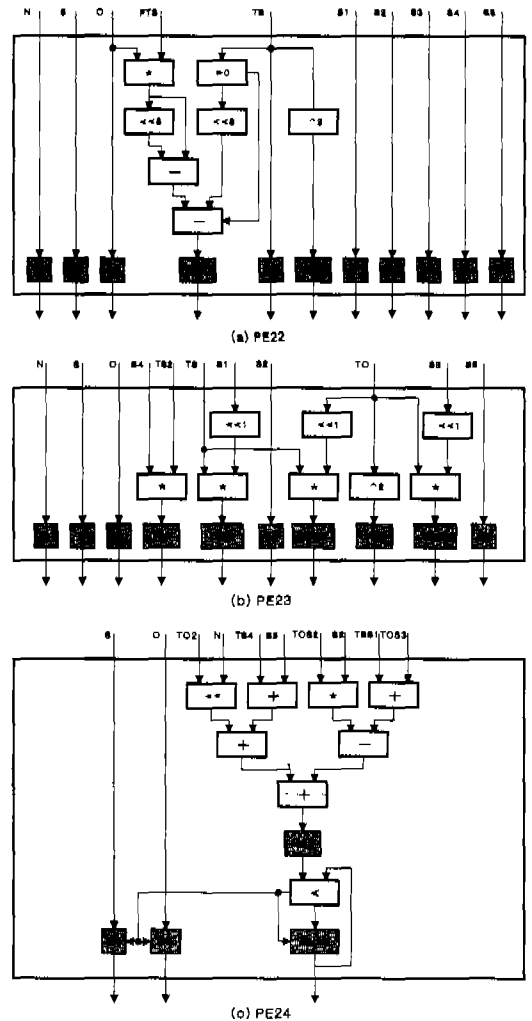


그림 8. msec를 구하기 위한 PE들

의 상대적 처리시간인  $T_R$ 을  $AT^2$ 의 척도로 비교하면 식 (1)과 같다. 처리해야하는 작업의 수(N)가 많다면 그림 4의 구조의 성능이 그림 3의 구조에 비해 약 3배 향상된다.

$$\begin{aligned}
 T_R &= AT^2 \\
 &= \frac{19}{25} \left( \frac{19+(N-1) \times 2}{25+(N-1)} \right)^2 \\
 &= \frac{19}{25} \left( \frac{2N+17}{N+24} \right)^2 \\
 &= 0.78 \times \frac{4N^2 + 68N + 289}{N^2 + 48N + 576} \\
 &\approx 3.04 \quad (\text{when } N \rightarrow \infty)
 \end{aligned}
 \tag{1}$$

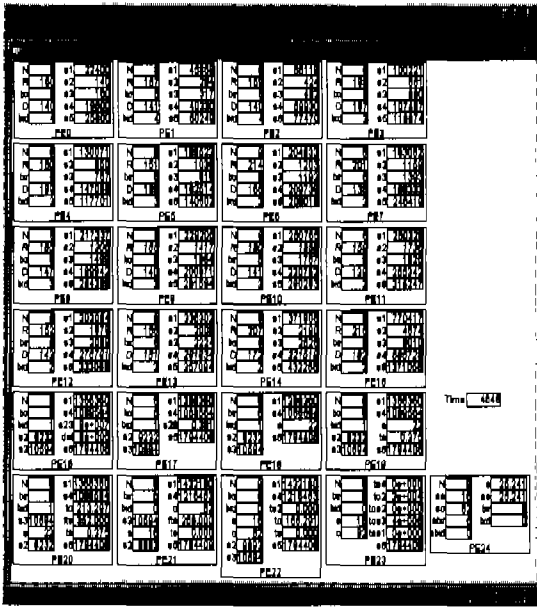


그림 9. 컴퓨터 시뮬레이션 화면  
: 4646 단위시간 후의 스냅샷

표 1. 그림 2와 4의 구조의 성능비교

	그림 2의 구조	그림 4의 구조
PE 수 (개)	19	25
입력핀수 (개)	16×8	16×8
출력핀수 (개)	13	13
처리시간	$\{19 + (N-1)\} \times t_1$ $t_1 \approx 2 \times t_2$	$\{25 + (N-1)\} \times t_2$
상대적인 처리시간 $T_R$	3.04	1

### V. 결론

본 논문에서는 최적의 파이프라인 주기를 갖는 프랙탈 영상압축의 고속수행을 위한 일차원 VLSI 어레이를 설계했다. 고정분할 알고리즘을 변형하여 VLSI 어레이 설계에 적합하며 화질의 손상을 최소화하면서 압축율이 높은 알고리즘을 유도했다. 이 알고리즘을 시간 및 공간사상을 통해 일차원 VLSI 어레이를 설계하였다. 파이프라인의 각 세그먼트를 구성하는 PE의 연산시간을 최대한 균등하게 분포시

켜 최적의 파이프라인의 주기를 얻었다. 설계한 일차원 VLSI 어레이는 25개의 PE로 구성된다. 또한 PE내부의 연산장치와 입력핀을 공유하여 ASIC의 구현에 유리하다.

본 논문의 연구 결과는 프랙탈압축의 고속화를 위한 ASIC을 구현하기 위한 기초 연구로 이용될 수 있다.

### 참고 문헌

- [1] E. W. Jacobs, Y. Fisher, and R. D. Boss, "Image compression: A study of the iterated transform method," Elsevier Science Publishers B. V, Signal Processing, Vol. 29, pp. 251-263, 1992.
- [2] A. E. Jacquin, "Fractal image coding: A review," Proceedings of the IEEE, Vol. 81, pp. 1451-1465, Oct., 1993.
- [3] K. P. Aiken, M. J. Irwin, and R. M. Owens, "A parallel ASIC architecture for efficient fractal image coding," Journal of VLSI Signal Processing 19, pp. 97-113, 1998.
- [4] A. E. Jacquin, "Fractal Image Coding: A Review," Proceedings of the IEEE, Vol. 81, pp. 1451-1465, Oct., 1993.
- [5] R. F. Uys, "Parallel implementation of fractal image compression," IEEE, Proceedings of the 1998 South African Symp. Comm. & Sig. Proc., pp. 143-148, 1998.
- [6] F. Ancarani, A. De Gloria, M. Olivieri, and C. Stazzone, "Design of an ASIC architecture for high speed fractal image compression," Proc. 9th annual IEEE International ASIC Conference and Exhibit, pp. 223-226, 1996.
- [7] D. J. Jackson and T. Blom, Fractal image compression using a circulating pipeline computation model, Proceedings of the ISCA 10th International Conference on Parallel and Distributed Comp. Syst., pp. 141-144, 1997.
- [8] K. P. Acken, "Low Power Architectural Optimizations for 3D Graphics Subsystems," Ph. D. thesis, Pennsylvania State University, Penn., 1997.
- [9] 성길영, 이수진, 우종호, "고속 프랙탈 영상압축을 위한 VLSI 어레이 구조" 한국통신학회논문

