

論文99-36C-12-2

효율적 프랙탈 영상 압축 복호기의 설계 및 구현

(Design and Implementation of Efficient Decoder for Fractal-based Compressed Image)

金 春 昊 * , 金 利 變 *

(Chun-Ho Kim and Lee-Sup Kim)

要 約

최근에 등장한 프랙탈 영상 압축 알고리즘은 소프트웨어적인 측면에서는 많이 연구되고 있으나, 하드웨어 구현을 위한 연구는 드물다. 그러나, 프랙탈 영상 압축 기법이 동영상 처리를 위해 사용될 경우 소프트웨어 적으로는 실시간 처리의 어려움이 있어 고속의 전용 하드웨어가 필요하다. 그러나, 아직 복호기의 구체적인 하드웨어의 설계 예는 드물다. 본 연구에서는 256×256 의 크기의 흑백 영상의 실시간 처리가 가능한 quadtree 방식의 프랙탈 영상 압축 복호기를 전용 하드웨어로 설계하였으며, 이를 위한 저전력 기법을 제안한다. 제안한 두 가지 방법 중 첫번째는 영상의 복원 후 발생하는 블록 현상을 제거하기 위한 post-processing 방법을 하드웨어 측면에서 최적화하는 것이다. 이 방식은 기존의 소프트웨어에서 사용하던 승산기가 필요한 가중 평균 방식보다 하드웨어를 적게 소모하여 비용을 줄이며, 속도는 69%정도의 향상이 있다. 두번째 방식은 데이터 패스 내부의 곱셈기를 입력 벡터의 통계적 특성을 이용하여 소비 전력이 적도록 설계하는 것이다. 이 방식으로 설계할 경우 8 bits 이하의 크기의 곱셈기에서 저전력에 유리하다고 알려진 어레이(array) 형태의 곱셈기에 비해 약 28%정도 소비 전력을 줄일 수 있었다. 위 두 가지 전력 절감 방식을 사용하여 동작 전압 3.3V, 1 poly 3 metal, $0.6\mu\text{m}$ CMOS 공정으로 복호기의 코어 부분을 칩으로 제작하였다.

Abstract

Fractal image compression algorithm has been studied mostly not in the view of hardware but software. However, a general processor by software can't decode fractal compressed images in real-time. Therefore, it is necessary that we develop a fast dedicated hardware. However, design examples of dedicated hardware are very rare. In this paper, we designed a quadtree fractal-based compressed image decoder which can decode 256×256 gray-scale images in real-time and used two power-down methods. The first is a hardware-optimized simple post-processing, whose role is to remove block effect appeared after reconstruction, and which is easier to be implemented in hardware than non- 2^k exponents weighted average method used in conventional software implementation, lessens costs, and accelerates post-processing speed by about 69%. Therefore, we can expect that the method dissipates low power and low energy. The second is to design a power-down multiplier used in data-path by using statistical characteristics of input vectors. The power dissipation in the multiplier can be reduced by about 28% with respect to a general array multiplier which is known efficient for low power design in the size of 8 bits or smaller. Using the above two power-down methods, we designed the decoder's core block in 3.3V, 1 poly 3 metal, $0.6\mu\text{m}$ CMOS technology.

* 正會員, 韓國科學技術院 電氣 및 電子工學科

Department of Electrical Engineering)

(Korea Advanced Institute of Science and Technology

接受日字: 1999年4月19日, 수정완료일: 1999年10月21日

I. 서 론

기본적으로 프랙탈 영상 압축은 영상의 자기 유사성에서 나타나는 중복성을 제거하는 방식이다^{[1][2]}. 지금까지 프랙탈 영상 압축에 대해 수학적인 접근 방식이 많이 연구되어 왔으며 소프트웨어적으로도 많이 구현되어왔다. 그러나, 동영상의 실시간 처리를 위해서는 고속의 선용 하드웨어가 필요하며 이를 위한 복호기의 효율적인 알고리즘(MRDA)이 제안된 바 있다^[3]. 그러나, 기존 설계^[3]에서는 소비 전력에 대한 고려가 전혀 없으며 post-processing을 위한 부분은 빠져있다. MRDA 방식에서는 지역(range)과 정의역(domain)을 위한 메모리를 공유하여 필요한 메모리 공간을 줄였으며 반복(iteration)에 따른 최종 영상으로의 접근 속도가 빨라져서 원하는 화질을 얻기 위한 반복 횟수를 3회로 줄일 수 있다. 그러나, 256×256 크기의 영상에 대해 이러한 방식을 사용하더라도 소프트웨어로는 여전히 초당 5~6 프레임 정도의 낮은 복원 속도 때문에 실시간 처리를 위한 복호화 시스템의 구현을 위해서는 더욱 빠른 복원 속도를 갖는 전용 하드웨어의 설계가 필요하다. 본 연구에서는 예전에 제안된 MRDA에 두 가지 저전력 설계 방식을 적용하여 칩으로 구현하였다. Quadtree 방식의 프랙탈 압축 영상의 복호화 과정은 quadtree 방식으로 분할된 각각의 치역(range)을 부호화 과정에서 유사성에 의해 찾아진 가로, 세로 각각 2배인 정의역(domain)으로부터의 축소 복사에 의해서 복원하는 것이다. 여기서 축소 복사는 정의역의 4 픽셀을 평균을 취하여 치역의 1 픽셀로 축소시키며 부호화 시에 찾은 계수(scale, offset)들을 사용해서 복원될 픽셀 값을 결정한다. 이렇게 모든 치역의 값을 복원하는 것이 1회 반복이다. 최종 영상의 해상도를 높이기 위해 그와 같은 과정을 몇 차례(MRDA의 경우 3회) 반복하게 된다. 이러한 과정에서 각각의 치역 영역은 다른 계수 값을 가지고 독립적으로 복원되기 때문에 복원된 전체 영상은 그림 6(a)에서 보는 것처럼 블록 현상을 갖게 된다. 프랙탈 영상의 경우 치역의 경계면을 따라 픽셀 값의 차이가 크기 때문에 이런 블록 현상은 MPEG에서 사용되는 블록 DCT의 경우보다 더욱 눈에 거슬린다. 따라서, 다소간 PSNR의 향상도 있지만 주관적인 영상의 화질 개선을 위해 경계면을 완화

(smoothing)하는 것이 필요하다^[4]. 프랙탈 영상 분할 과정에서 치역은 깊이(depth)에 따라 크기가 달라지기 때문에 그림 1(a)에서 보는 것처럼 치역의 경계면은 불규칙적으로 복잡하게 흘러져 있다. 이러한 경계면을 모두 따라가면서 완화하는 것은 하드웨어적으로 복잡한 제어를 필요로 할뿐 아니라, 그로 인하여 하드웨어 구현 시 상당한 칩 면적을 요구하게 된다. 또한, 모든 치역의 경계를 인식하기 위해 각 치역의 정보를 저장하는 것이 필요하다. 따라서, 이를 복잡한 경계면을 모두 완화하는 것은 전력면에서도 상당히 불리하다. 한편, 곱셈기를 필요로 하는 역 프랙탈 변환은 전체 복호화 시간의 90% 이상을 소비하므로 영상 복원을 위한 대부분의 연산은 데이터 폐스 내부의 MAC 유닛에 집중되어 있다. 따라서, 전체 복호화 과정 동안의 소비 전력을 줄이기 위해서는 그 MAC 유닛 내의 곱셈기의 소비 전력을 줄이는 것이 효과적이다.

II. 저전력 Post-processing 방식

Post-processing의 중요한 역할은 복원 영상의 치역 간의 블록 현상을 제거하는 것이다. 사람의 눈은 공간 주파수의 고주파 성분보다 저주파 성분에 더욱 민감하다^[5]. 뿐만 아니라, 긴 경계면에서의 변화에 더욱 민감하다. 따라서, 블록 현상의 큰 부분을 차지하는 최소 깊이 경계의 블록 현상을 제거하는 것이 효과적일 수 있다. 따라서, 우리는 모든 치역의 경계를 완화하는 것이

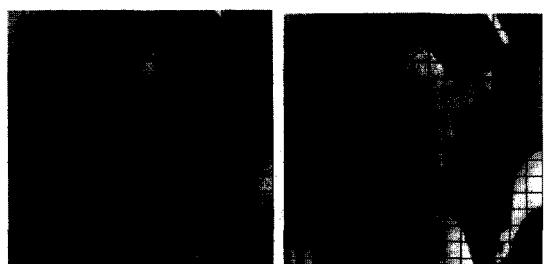


그림 1. 완화(smoothing)하는 치역의 경계:

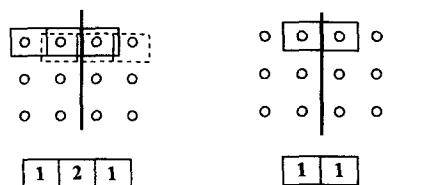
- (a) 모든 치역의 경계
- (b) 최소 깊이 치역의 경계

Fig. 1. Range's edges to smooth:

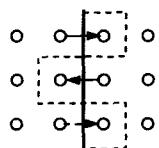
- (a) all ranges' edges
- (b) min-depth ranges' edges

아니라 그림 1(b)에서처럼 격자 모양을 갖는 최소 깊이 치역의 경계만 완화하는 것을 제안한다. 그러면, 그림 1(a)와 비교해서 완화할 부분이 규칙적인 모양이 되므로 제어 로직도 간단히 구현할 수 있고 클럭 사이클 수도 절약할 수 있다. 또한, 완화하는 경계를 줄임으로써 이를 위해 필요한 에너지를 더욱 줄일 수 있다.

그리고, 다음으로 생각할 문제는 어떤 방법으로 완화하는 효과를 얻는가 하는 것이다. 우선 그림 2에서 보는 것과 같이 기존의 소프트웨어 구현의 경우와는 달리 제산기(divider)를 필요로 하지 않는 하드웨어 구현에 적합한 세가지 방법을 들 수 있다. 그림 2의 세가지 방식을 각각 속도, 영상의 화질, 하드웨어 비용 측면에서 비교해봄으로써 그 중 최적의 방식을 결정할 것이다. 그림 2(a)방식에서는 경계면을 따라가면서 좌우 각각의 픽셀 값을 구하기 위해 그림에서처럼 3개의 픽셀을 1-2-1의 가중치를 가지고 평균을 취한다. 따라서, 다른 두 가지 방식들에 비해 계산량이 많고 덧셈기를 많이 필요로 한다. (b)방식에서는 경계면을 따라 좌우 두 픽셀 값을 읽어 평균값을 다시 그 두 픽셀 값으로 쓴다. 따라서, 이 경우는 원래의 밝기 레벨의 차이가 심한 경계면을 흐릿하게 만들 가능성이 있지만 필요한 하드웨어와 계산량이 적다. 그리고 (c)방식에서는 경계 면을 따라 이동하면서 한번은 좌에서 우로 다음은 우에서 좌로 지퍼(zipper) 모양으로 번갈아 가면서 읽은 값을 곧바로 쓴다. 따라서, 이 경우 블록 현상이 심한 경계면에 대해 지퍼 모양의 잡음(noise)이 나타날 수 있다. 그러나, 한 개의 레지스터를 제외하고 추가적인 하드웨어가 필요 없으므로 하드웨어 측면에서 가장 유리하다.



(a) 1-2-1 weight smoothing (b) 1-1 weight smoothing



(c) zipper smoothing

그림 2. 하드웨어 구현이 쉬운 세가지 완화 방식
Fig. 2. Three smoothing type for easy hardware implementation.

그림 3은 위 세가지 완화 방식의 하드웨어 구현시의 비용(costs)에 대한 비교를 보여준다. 그리고, 그림 4는 여러 가지 완화 방식에 의한 영상의 화질 비교에 사용된 세가지 테스트 영상을 보여준다. 여러 가지 테스트 영상에 대해 완화하지 않은 경우와 모든 경계를 완화한 경우 그리고 앞에서 제안한 세가지 완화 방식을 적용한 경우 각각의 완화 속도와 복원 영상의 화질을 비교해 보면 그림 5 (1),(2)와 같다. 참고로, 완화하지 않는 경우에 있어 전체 복호화 과정의 클럭 사이클 수는 약 590,000 정도이다.

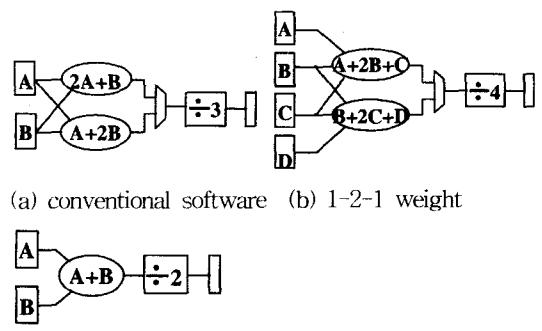


그림 3. 세가지 완화 방식의 하드웨어 비교

Fig. 3. Comparison of hardware for three smoothing types.

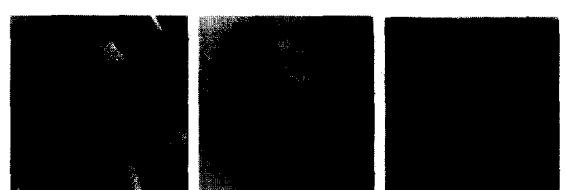
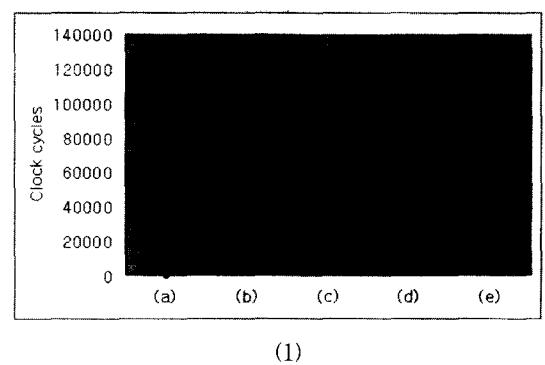
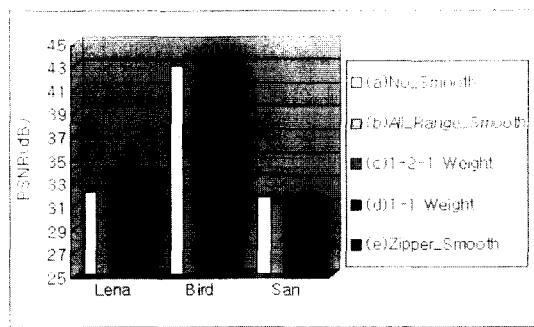


그림 4. 완화를 위한 세가지 테스트 영상
Fig. 4. Three test images for smoothing.





(2)

그림 5. 여러 가지 완화 방식의 클럭 사이클과 화질(PSNR)의 비교: (1) 필요한 클럭 사이클 비교 (2) 화질(PSNR)의 비교 (a) 완화하지 않은 경우 (b) 모든 치역의 경계를 1-2-1 가중치로 완화한 경우 (c) 1-2-1 가중치로 완화한 경우 (d) 1-1 가중치로 완화한 경우 (e) 지퍼 방식으로 완화한 경우

Fig. 5. Clock cycle and PSNR comparison of several smoothing types: (1) comparison of needful clock cycles (2) comparison of PSNR (a) no smoothing (b) All Range smoothing (c) 1-2-1 weight smoothing (d) 1-1 weight smoothing (e) zipper smoothing.

영상의 블록 현상을 제거하기 위한 post-processing 후의 PSNR은 완화 방식에 크게 영향을 받지 않는다. 다만, 중요한 것은 블록 현상을 충분히 제거하는데 있다. 따라서, 여러 완화 방식 중 블록 현상을 충분히 제거하면서 하드웨어, 전력 및 에너지면에서 유리한 방식을 선택하는 것이 필요하다. 그림 6에서 여러 가지 완화 방식에 의해 얻어진 영상과 그 영상들의 원 Lena 영상과의 차이 영상을 비교해 보면 블록 현상이 제거된 정도를 알 수 있고 완화의 필요성과 적절한 완화 방식을 결정할 수 있다. (a)그림에 비해 (b),(c),(d)에서는 치역의 경계에서 블록 현상이 많이 사라진 것을 볼 수 있다. 또한, (e)그림은 지퍼 잡음이 많이 나타나는 것을 볼 수 있으며, (5)의 차이 영상을 보더라도 (3),(4)보다 큼을 알 수 있다. 모든 치역의 경계를 완화하는 것은 복잡한 제어 로직과 많은 하드웨어를 필요로 하므로 (c),(d)의 방법이 적은 하드웨어로 원하는 효과를 충분히 얻을 수 있어 하드웨어 구현에 더욱 적합하다. 또한, (c)와 (d)를 비교해보면 블록 현상은 둘 다 충분히 제거되며 PSNR에 있어서도 그림 5(2)에서 보듯이 미미한 차이가 있을 뿐이다. 그러나, 하드웨어와 속도면

에서 (d) 방식이 유리하다. 따라서, 제안된 완화 방식 중 최적의 방식은 1-1 가중치 완화 방식이다. 그림 5(a)에서 1-1 가중치 완화 방식이 기존의 모든 경계를 완화하는 방식보다 완화 시간을 약 69% 정도 절약할 수 있음을 알 수 있다. 따라서, 1-1 가중치 완화 방식은 간단한 하드웨어(적은 연산)를 사용하고 완화 시간을 절약하므로 저전력 및 저에너지 설계이다.

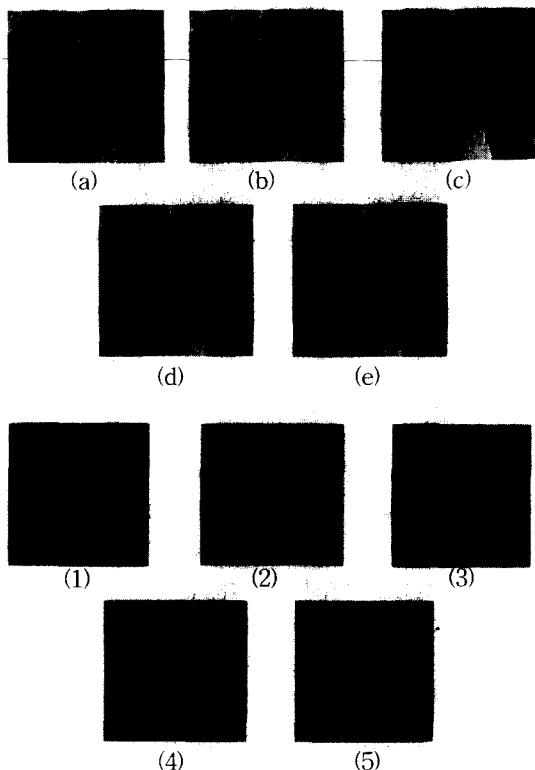


그림 6. 여러 가지 완화 방식에 의한 결과 Lena 영상 및 원 영상과의 차이 영상

(a),(1) 완화하지 않은 경우 (b),(2) 모든 치역의 경계를 완화한 경우 (c),(3) 1-2-1 가중치로 완화한 경우 (d),(4) 1-1 가중치로 완화한 경우 (e),(5) 지퍼 방식으로 완화한 경우

Fig. 6. Various smoothed results of Lena image and difference images.

(a),(1) No smoothing (b),(2) All range smoothing (c),(3) 1-2-1 weight smoothing (d),(4) 1-1 weight smoothing (e),(5) zipper smoothing.

III. 저전력 곱셈기의 설계

프래탈 역변환은 주로 MAC 연산에 의해 얻어지는

축소 복사(contractive mapping)의 반복에 의해 이루어 진다. 여기서 MAC 연산은 두 픽셀의 합(9 bits)과 scale(5 bits)값의 곱셈 그리고 그 결과값과 offset(15 bits)값의 덧셈으로 구성된다. 여기서, 픽셀 값은 매 클럭마다 값이 바뀌지만 scale값과 offset값은 하나의 차역을 복원하는 동안 값이 고정되어 있다. 따라서, 수십 사이클동안 고정된 값을 갖는 이들 계수를 FIR 필터의 고정된 계수로 간주할 수 있다. 일반적으로, 8 bits 이하의 작은 크기의 곱셈기에 대해 어레이(array) 형태의 곱셈기가 저전력에 유리하다고 알려져 있다^[6]. 그러나, 한쪽 입력이 고정된 FIR 필터를 위한 곱셈기의 경우 고정된 입력을 Booth 인코딩시켜 Booth 곱셈기를 구성하면 어레이 형태의 곱셈기보다 더 전력 소모를 절감할 수 있다^[7]. 우리가 설계한 곱셈기도 이처럼 한쪽 입력이 상당기간 고정된 값이므로 동일한 방식을 적용해서 소비 전력을 절감할 수 있을 것이다.

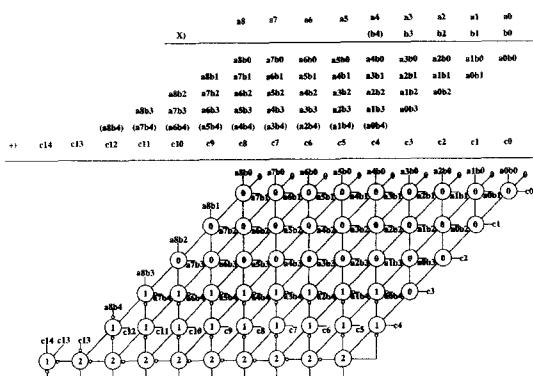


그림 7. 일반적인 어레이(array) 형태의 곱셈기
Fig. 7. GAM (General Array Multiplier).

그림 7은 일반적인 어레이 형태의 곱셈기의 9x5 크기의 어레이 구성을 보여준다. 그림 8과 9는 Booth 곱셈기에서 고정된 입력을 각각 피승수, 승수중 어느쪽에 인가하는가에 따른 곱셈기의 소비 전력 비교를 위해 구성한 두가지 형태의 곱셈기를 보여준다. 실제 설계 시 하드웨어 절약을 위해 아래의 두가지 기법이 사용되었다. 두 곱셈기의 설계 시 2의 보수의 부호 확장을 위해 필요한 1 bit 덧셈기의 개수를 줄이기 위해 부호 생성 알고리즘(sign generation algorithm)을 사용하였다^[8]. 또한, 그림 8, 9에서 한 열에 있는 bit 수를 줄이기 위해 각각 위쪽 상자를 아래쪽 상자로 변환하였다. 왜냐하면, 어떤 한 열에서의 bit의 개수가 증가하게 되

면 그 옆에서 1 bit 덧셈기가 하나 더 필요하게 되며 carry 전파를 위해서 그 상위 열에서 모두 1 bit 덧셈기가 하나씩 더 늘어나게 된다. 이것은 하드웨어 면에서나 전력면에서나 모두 불리하기 때문이다.

그림 8, 9에서 부분 합 생성(partial product generator) 로직과 Booth 인코딩 단은 모두 생략되었고, 프라임 표시된 것은 Booth 인코딩에 의해 만들어진 부분 항을 의미한다. 참고로, b5, a9, a10은 2의 보수 표기를 위한 부호 확장이며 a9, a10은 양의 값인 픽셀 값의 부호 확장이므로 항상 0의 값이다. 그리고, 그림에서 (1)은 Booth 인코딩에서 음수일 때만 1이다.

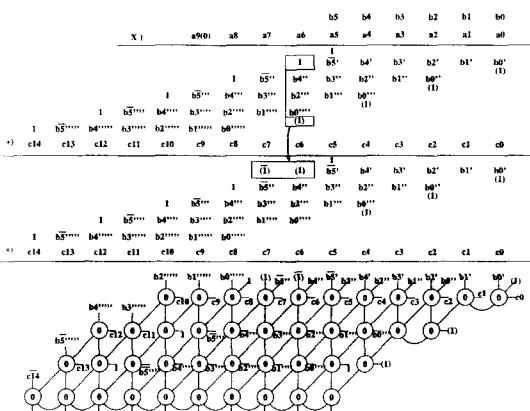


그림 8. 픽셀 값 Booth 인코딩 곱셈기
Fig. 8. PBEM (Pixel Booth encoding multiplier).

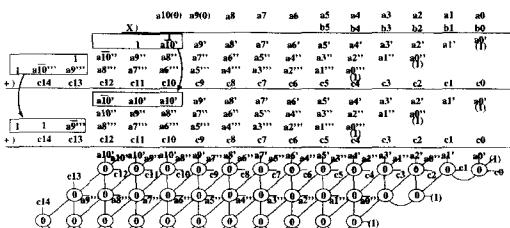


그림 9. Scale 계수 Booth 인코딩 곱셈기
Fig. 9. SBEM (Scale Booth encoding multiplier).

일반적으로, Booth 곱셈기는 승수와 피승수 사이의 경주 조건(race condition)에 의한 곱셈기 내의 덧셈기 어레이(array)에서의 glitch 증가 때문에 전력 소모가 많다. 그러나, Booth 인코더의 입력 값이 오랫동안 고정된 값이면 경주 조건과 glitch가 많이 줄어 전력을 절감할 수 있다. 그림 10은 위의 세가지 곱셈기에 대해 앞에서 보인 3가지 테스트 영상의 중앙에 표시된 작은 영역(1024 픽셀)에 대한 HSPICE 모의 실험으로부터

연속 신력 소모의 비교를 보여준다.

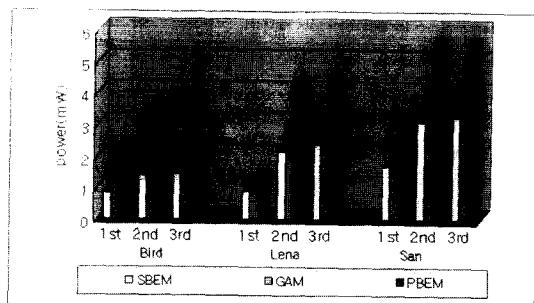


그림 10. 위 세가지 곱셈기의 세가지 테스트 영상에 대한 클럭당 평균 전력 소모

Fig. 10. Average power dissipation per one clock cycle in the above three multipliers to three test images.

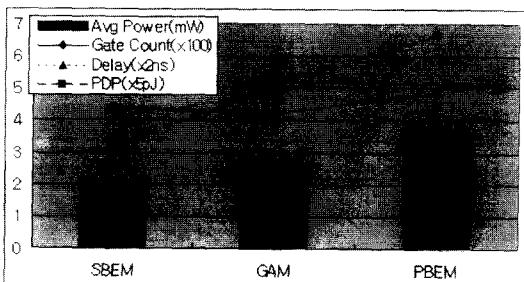


그림 11. 세가지 곱셈기의 PDP 비교

Fig. 11. Comparison of PDP to the above three multipliers.

Lena 영상은 픽셀 값의 변화가 중간 정도이고, Bird 영상은 픽셀 값의 변화가 적고, San Francisco 영상은 픽셀 값의 변화가 많다. 이를 다양한 형태의 영상에 대해 SBEM 방식은 GAM이나 PBEM 방식보다 적은 전력을 소모하는 것을 볼 수 있다. 픽셀 값의 변화가 심한 San Francisco 영상의 경우가 다른 영상의 경우보다 전력 소모가 많다. 또한, 그림에서 알 수 있듯이 반복 횟수가 증가할수록 전력 소모도 따라서 증가한다. 이것은 반복 횟수가 증가할수록 영상의 화질이 좋아지면서 픽셀 값의 변화가 증가하기 때문이다. SBEM 방식이 GAM 방식보다 평균적으로 약 28%의 전력을 절감할 수 있다. 따라서, 설계하고자 하는 복호기의 곱셈기처럼 작은 크기이며 입력 값이 고정인 특성을 가지면 어레이 형태의 곱셈기 대신 SBEM 방식의 곱셈기를 사용함으로써 전력 소모를 줄일 수 있다. 그림 11은 세가지 방식의 곱셈기에 대한 평균 전력, 지연 시간 및

PDP(Power Delay product)의 비교를 보여준다. PDP의 비교로부터 SBEM 방식의 우수성을 다시 확인할 수 있다.

IV. VLSI 구현

앞에서 언급한 두 가지 전력 절감 방법과 MRIA 알고리즘을 사용하여 $0.6\mu m$ CMOS 공정으로 $4mm \times 4mm$ 의 면적 내에 프래탈 영상 압축 복호기의 코어 블럭을 설계하였다. 사용한 공정에서 전체 편수는 80편이며, Verilog-XL과 HSPICE를 사용하여 제안한 두가지 방법을 검증하였다. 칩의 레이아웃은 작성한 Verilog 코드를 SYNOPSYS 툴로 합성하여 MENTOR 툴로 자동 P&R하여 얻었다.

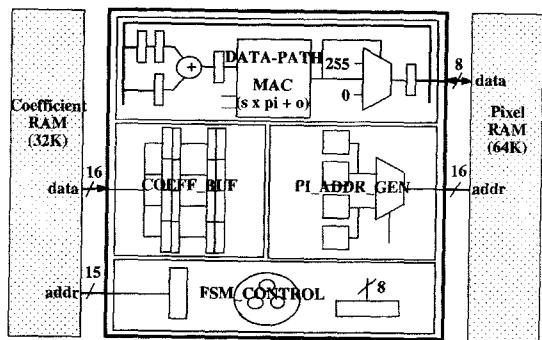


그림 12. 복호기 칩의 동작도

Fig. 12. Block diagram of the decoder.

그림 12는 제작된 복호기의 블록도를 보여준다. 전체 칩은 그림에서 굵은 선으로 둘러싸인 부분이며, 코어 부분에서 사용된 전체 게이트의 수는 3,692 개이다. 코어 부분은 다시 DATA-PATH(849개), COEFF-BUF(1024개), PI-ADDR-GEN(1026개) 및 FSM-CONTROL(793개)의 4부분으로 구성되어 있으며 3.3V에서 동작하도록 설계되었다. 그림 12에서 보듯이 DATA-PATH는 하나의 덧셈기, MAC 유닛과 픽셀 값의 바운딩을 위한 MUX로 구성되어 있다. COEFF-BUF는 계수 값의 prefetch 및 유지를 위한 버퍼 메모리이고, prefetch 구조를 통해서 그림 13에서 보는 것처럼 하나의 치역의 축소 복사를 하는 동안 4개의 클럭을 절약 할 수 있게 하였다. PI-ADDR-GEN은 치역과 정의역의 X, Y 좌표 값을 위한 4개의 8 bits 카운터로 구성하였다. 마지막으로, FSM-CONTROL 부분은 복호기의 내외부의 제어 신호를 만들기 위한 FSM이며, 다양한 크기(4x4, 8x8, 16x16)의 치역의 경계면을 인식하기 위해

전용 하드웨어 구현의 장점을 살려서 하드웨어 면적과 소비 전력을 줄이기 위해 많은 수의 덧셈기와 비교기를 사용하는 대신 그림 14의 간단한 조합 회로를 사용하였다.

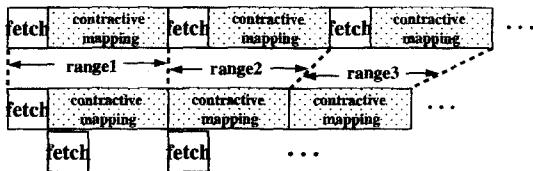


그림 13. 클럭 사이클 절약을 위한 계수 값 prefetch
Fig. 13. Coefficients prefetch for clock saving.

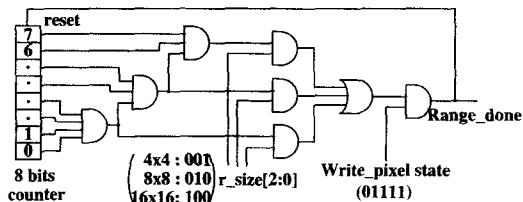


그림 14. 치역의 경계 인식을 위한 간단한 조합 회로
Fig. 14. Simple combinational logic for range boundary control.

자세한 상태 천이도는 그림 15에 나타나 있다. 그림 15에서 축소 복사(contractive mapping) 부분이 전체 복호화 시간의 90% 이상을 차지하며 이 부분의 전력 소모를 줄이기 위해 앞에서 저전력 곱셈기를 제안하였다. 그리고 완화를 수행하는 부분이 5% 정도를 차지하며 이 부분의 소비 전력을 줄이기 위해 간단한 post-processing 방법을 제안하였다. 전체 동작은 계수 메모리(Coefficients RAM)로부터 필요한 치역 정보들을 읽어 들인 후 데이터 패스 연산과 영상 메모리

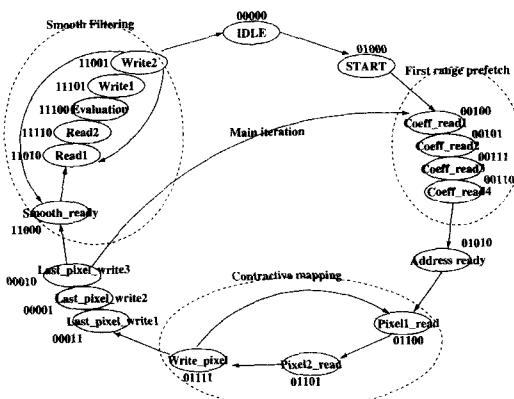


그림 15. FSM의 설계
Fig. 15. Design of FSM.

(Pixel RAM)로부터 읽기, 쓰기를 반복하는 것이며, 이를 통해서 영상을 복원하게 된다. 이 복호기는 256×256 흑백 영상을 초당 63 프레임의 속도로 복원할 수 있다. 그림 16은 제작된 전체 칩의 사진을 보여준다. 또한, 칩의 주요 사양은 표 1에 나타나 있다. 설계된 칩의 제작이 완료되었으며, 테스트로부터 얻어진 Shmoo 플롯은 그림 17과 같다.

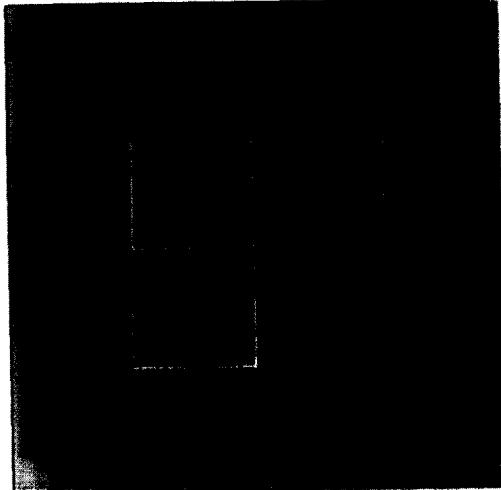


그림 16. 설계 칩의 사진
Fig. 16. Chip Photo.

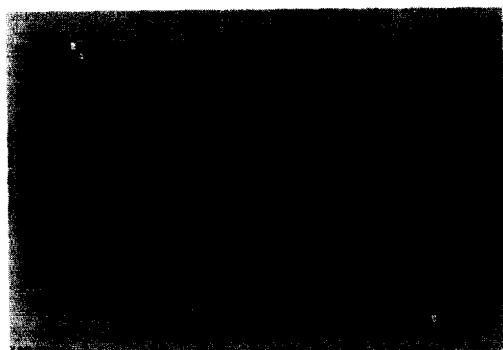


그림 17. 슈무(Shmoo) 테스트 결과
Fig. 17. Shmoo test results.

표 1. 칩의 동작 특성

Table 1. Chip statistics.

공정	1 poly 3 metal, 0.6(m CMOS
칩 크기	$4\text{mm} \times 4\text{mm}$
I/O 핀수	80 pins
공급 전압	3.3 V
동작 주파수	40 MHz
소비 전력	9.5mW @ 3.3V, 25°C
복원 속도	63 프레임/초

V. 결 론

본 연구에서는 256×256 크기의 프랙탈 압축 영상의 실시간 복원을 위한 복호기의 구조를 제안하고 실제 하드웨어로 구현하였다. 또한, 휴대용 장비로의 응용을 위해 두 가지 저전력 기법을 제안하였다. 첫째로, 주관적인 영상의 화질 향상을 고려해 quadtree 방식의 프랙탈 영상을 복원한 후 발생하는 블록 현상을 줄이기 위한 간단하고 효과적인 post-processing 방법을 제안하였다. 그 방법은 기존의 소프트웨어에서 사용되는 방식과 비교해 하드웨어 및 수행시간을 적게 요구한다. 특히, 맨하탄 스타일의 San Francisco 영상에서는 0.3dB 정도의 PSNR의 손실이 있었지만, Lena 영상이나 Bird 영상 같은 일반적인 영상에 대해 치역의 경계에서의 블록 현상의 감소는 충분히 얻을 수 있었다. 제안된 방법은 모든 치역의 경계를 완화하는 경우에 비해 약 69%의 사이클을 절약할 수 있으며, 적은 하드웨어를 사용하고 필요한 사이클 수를 줄였으므로 저전력 및 저 에너지 설계이다. 축소 복사를 수행하는 구간과 완화를 수행하는 구간의 평균 소비 전력이 같다는 가정을 하면 1-2-1 가중치로 모든 경계를 완화하는 경우에 비해 제안한 방식에 의해 절약되는 에너지는 전체 코어 블록에서 약 3.7% 정도이다. 두 번째로 제안한 저전력 곱셈기의 설계에서는 8 bits 이하의 적은 bit 수의 곱셈기에서 SBEM 방식을 사용하여 곱셈기 내에서의 평균적인 소비 전력을 기존의 일반적인 어레이(array) 형태의 곱셈기에 비해 약 28%, PBEM 방식에 비해서는 약 48%를 절약할 수 있었다. 이 저전력 곱셈기의 사용에 의해 전체 코어에서 5.6%의 소비 전력을 줄여서 두 가지 방법에 의해 전체적으로는 약 9.3%의 전력(에너지) 절감 효과를 얻었다. 제작 칩의 테스트 결과 최고 동작 주파수는 40MHz였으며, 3.3V 공급 전압에서 코어 블록의 평균 소비 전력은 9.5mW였다. 그리고 초당 30 프레임의 실시간 처리가 가능한 최소 동작 주파수 및 전압은 각각 20MHz, 1.9V 였으며, 이때 평균 소비 전력은 1.57mW까지 낮아질 수 있었다.

감사의 글

이 논문에서 제작된 칩은 Integrated circuits Design Education Center(IDE)에서 개최한 제 5차 MPW에 공모한 작품입니다.

참 고 문 헌

- [1] Y. Fisher, E. W. Jacobs, and R. D. Boss, "Fractal Image Compression Using Iterated Transform", NOSC Technical Report, Naval Ocean Systems Center, San Diego 1990.
- [2] Jacquin, A. E. "A fractal theory iterated Markov operators with applications to digital image coding", Ph. D Thesis. Georgia Tech, 1989.
- [3] Kyung-Hoon Kim, "VLSI implementation of decoder for decompressing fractal-based compressed image", Proc. IEEE ISCAS'98, Vol.4, pp. 221-224.
- [4] Yuval Fisher, "Fractal Image Compression", Springer-Verlag New York Inc., pp 59-61, 1995.
- [5] Anil K. Jain, "Fundamentals of digital image processing", Prentice-Hall Inc., pp 54-55, 1989.
- [6] Jan M. Rabaey, "Low power design methodologies", Kluwer Academic Publisher, pp 54-55, 1989.
- [7] Chris J. Nicol, "Low Power Multiplication for FIR Filters", ACM ISLPED Proc., pp 76-79, 1997.
- [8] M. Annaratone, "Digital CMOS circuit Design", Kluwer Academic Publisher, Boston 1986.

저자소개

**金 春 昊(正會員)**

1996년 경북대학교 전자공학과 학사.
1999년 한국과학기술원 전기 및 전자공학과 석사. 주관심분야 : 프랙탈
영상 압축, 저전력 VLSI 설계, 3D Graphics

**金 利 煥(正會員)**

1982년 서울대학교 전자공학과 학사. 1986년 Stanford University 전자공학과 석사. 1990년 Stanford University 전자공학과 박사. 1990~1993년 Toshiba Corporation 연구원. 1993년~현재 한국과학기술원 전기 및 전자공학과 부교수. 주관심 분야 : 멀티미디어 VLSI 설계, 저전력 설계, EML 설계, 3D Graphics