

1) Mallat tree 방법을 이용한 실시간 2-D DWT의 FPGA 구현을 위한 효율적인 메모리 사상

김 왕 현, 서 영 호, 김 종 현, 김 동 욱
광운대학교 반도체 및 신소재 공학과
전화 : 02-940-5167 / 핸드폰 : 018-272-3240

The Efficient Memory Mapping of FPGA Implementation for Real-Time 2-D Discrete Wavelet Transform using Mallat tree algorithm

Wang-Hyun Kim, Young-Ho Seo, Jong-Hyun Kim, Dong-Wook Kim
Dept. of Semiconductor and new materials, Kwangwoon University
E-mail : whkim@gwu.ac.kr

Abstract

This paper proposed an efficient memory scheduling method (E^2M^2) by which the real-time image compression using 2-dimensional discrete wavelet transform(2-D DWT) is possible in an FPGA chip. In this paper, we assumed that the 2-D DWT was performed as the Mallat-tree.

After the memory mapping method was proved in software, the memory controller was designed for an commercial SDRAM IC.

I. 서론

현재의 영상/비디오 처리를 위한 신호처리 방법은 DCT(Discrete Cosine Transform)를 기반으로 하는 방법들이 주로 사용되고 있으나, 압축률이 증가함에 따라 블록효과가 나타나는 문제가 발생한다.[1,2] 이러한 문제점들을 극복하기 위해서, 최근에 이산 웨이블릿(wavelet)을 이용한 변환방법이 대두되어 그 사용 영역을 넓혀가고 있다. 그러나, 2차원 DWT는 연산 과정

상에서 많은 데이터를 저장해야하는 문제가 발생한다. 또한, FPGA로 H/W를 구현할 경우, 막대한 양의 데이터를 저장할 만한 내부 메모리를 FPGA가 포함하고 있지 않기 때문에, 외부 메모리를 사용해야한다. 하지만, 외부 메모리는 제한된 접근 시간과 열 방향으로만 연속 쓰기와 읽기 동작이 되는 특성이 있다. 따라서, 외부 메모리의 제한된 특성을 최대한 이용하여 실시간 FPGA칩을 구현하기 위해서는 연산 모듈에서 출력되는 데이터를 외부 메모리에 효율적으로 사상하는 방법의 연구가 필수적이다.

본 논문은 2-D DWT를 mallat-tree 방식으로 수행하여 중간 결과를 외부 메모리에 저장할 때 항상 필터가 열 방향으로 놓이도록 데이터를 사상하는 방식을 제안한다. 제안된 사상 방법을 C-언어로 DWT 에뮬레이터를 구현하여 검증하고, 또한, 상용화된 SDRAM의 메모리 제어를 H/W로 구현한다.

II. Mallat Tree를 이용한 2-D DWT 알고리즘

Mallat-tree를 이용한 DWT 알고리즘은 주어진 영상을 주파수 성분에 따라 연차적으로 분할하여 영상의 에너지를 저주파 영역으로 집중시키는 작용을 한다. 그 수행 방법은 식 (1)과 (2)에 나타낸 것과 같이 영상의 고주파 성분과 저주파 성분을 각각 필터링 한

1) 본 연구는 2001년도 중소기업청 산학연 공동기술개발 컨소시엄 사업에 의해 지원되었음.

다.[3][4] 이 수식에 의해서 한 개의 화소가 DWT되는

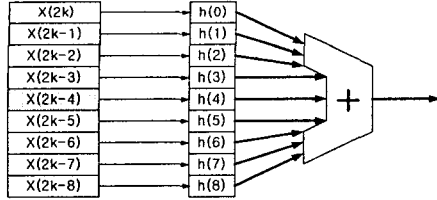


그림 1. (9,7) 필터에 의한 영상의 필터링

$$y_{low} = \sum_{n=0}^{N-1} x(2k-n)h(n) \quad \dots (1)$$

$$y_{high} = \sum_{n=0}^{N-1} x(2k-n)g(n) \quad \dots (2)$$

방법은 그림 1과 같다. 인간의 눈의 특성에 따라 저주파 영역에 시각적으로 중요한 정보가 집중되므로, Mallat-tree 알고리즘은 저주파영역에 대한 신호 분할 및 서브-샘플링을 계속 수행한다. 그림 2는 Mallat-tree를 이용하여 영상의 열을 기준으로 필터링을 수행하고 그 결과를 행 방향으로 필터링 할 때, 4레벨까지 2-D DWT를 수행하는 절차와 그 결과 분할된 영상을 보여주고 있다.

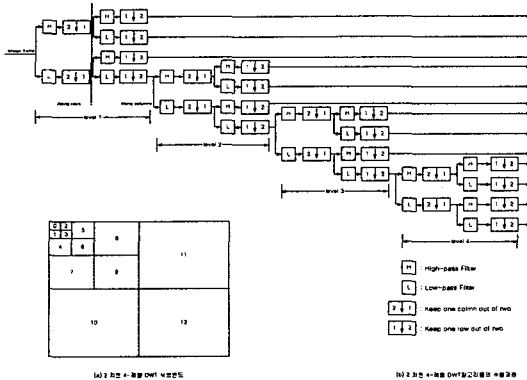


그림 2. 2차원 DWT의 4-레벨 수행절차와 영상분할 결과

4-레벨 수행 결과는 총 13개의 부대역으로 나뉘어지며, 각 부대역은 전체 영상에 대한 특별한 영상변화 정보들을 갖게 된다. 이 때, 행 연산이 수행되기 위해서는 열에 대해 변환이 수행된 웨이블릿 계수들을 메모리에 저장한 후, 이 값을 읽어서 행 방향 필터링을 수행하게 된다. 각 레벨의 열과 행 필터링을 수행한 결과는 총 4개의 영역(LL, LH, HL, HH)으로 분할되며, 이 중에서 LL영역이 그 다음 레벨의 변환에 사용된다. 따라서, 메모리를 읽고 쓰는 과정이 매우 많이

필요하며, H/W로 DWT를 구현하는 경우 연산을 수행하는 부분의 동작속도와 함께 메모리의 읽기/쓰기 시간은 H/W 복잡도 증가에 큰 영향을 미친다.

III. 메모리 읽기/쓰기 시간

본 논문에서는 DWT를 수행할 대상영상을 현재 가장 널리 사용되는 NTSC 방송영상의 크기를 가정하며, Y:Cb:Cr=4:2:2인 칼라정보로 구성되었다고 가정한다. 일반적으로 $N \times M$ (N: 행의 개수, M: 열의 개수) 칼라 영상에 대해 DWT를 수행할 때 한 개의 연산부만을 사용한다면, 연산부가 처리해야 되는 $Y(DY_{total})$, Cb(DCb_{total}), Cr(DCr_{total}) 각각과 전체 데이터 양($DAll_{total}$)은 식(3),(4),(5),(6)과 같다.

$$DY_{total} = \frac{(N \cdot M)}{2} \cdot 2 + \frac{(N \cdot M)}{8} \cdot 2 + \frac{(N \cdot M)}{32} \cdot 2 + \frac{(N \cdot M)}{128} \cdot 2 = \frac{85}{64}(N \cdot M) \quad (3)$$

$$DCb_{total} = \frac{(N \cdot M)}{4} \cdot 2 + \frac{(N \cdot M)}{16} \cdot 2 + \frac{(N \cdot M)}{64} \cdot 2 + \frac{(N \cdot M)}{256} \cdot 2 = \frac{85}{128}(N \cdot M) \quad (4)$$

$$DCr_{total} = \frac{(N \cdot M)}{4} \cdot 2 + \frac{(N \cdot M)}{16} \cdot 2 + \frac{(N \cdot M)}{64} \cdot 2 + \frac{(N \cdot M)}{256} \cdot 2 = \frac{85}{128}(N \cdot M) \quad (5)$$

$$DAll_{total} = DY_{total} + DCb_{total} + DCr_{total} = \frac{340}{128}(N \cdot M) \approx 2.66(N \cdot M) \quad (6)$$

예를 들어, NTSC방식의 640×480 영상에 대해 연산 모듈이 처리해야 되는 동작속도는 위의 일반식에 의해서 약 816,000이다. 따라서, 1초에 30프레임의 칼라영상을 처리하기 위해서는 1개의 연산 모듈이 한 화소를 처리해야하는 시간은 다음과 같다.

$$\frac{1}{30 \times 816000} \approx 41[\mu s]$$

이러한 동작속도를 내는 연산모듈을 실제로 구현하는 것은 불가능하고, 다음 연산을 위해서 메모리에 데이터를 저장하거나 읽어오는 연산 모듈과 메모리간의 접근 시간은 현재 상용화되어 있는 메모리로는 만족시키지 못하므로, 실제 H/W구현에서는 여러 개의 연산 모듈들을 사용하여 데이터 처리를 분산시켜, 위의 문제를 해결하게 된다.

IV. 메모리 사상 방법

이론상으로 열 방향의 변환을 수행할 때는 필터가 수평으로 대응되게 되고, 행 방향으로 처리할 때는 필터가 수직 방향으로 놓이게 된다. 따라서, 열(행) 방향 변환한 결과를 그대로 저장한다면 그 다음 단계의 행(열) 방향 변환을 수행할 때 필요한 데이터는 그림 3에서 보인 바와 같이 사용방향과 반대의 방향에 놓이게 된다. 그런데, 상용 SDRAM은 열에 대해서만 연속 동작(burst operation)을 수행할 수 있다. 따라서, 행 방향의 연산을 위해서 데이터를 메모리로부터 한 개씩 읽어 들인다면, 실시간 H/W 동작을 수행할 수 없다.

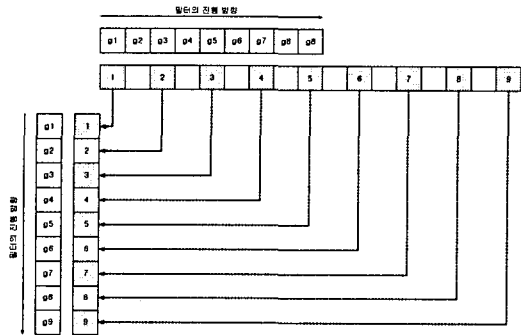


그림 4. 이론적인 웨이블릿 계수의 메모리 사상 방법

본 논문에서는 상용 SDRAM의 열 방향 연속 동작을 이용하여, 행 방향의 동작을 할 때 필터가 수평으로 놓이도록 하기 위해서 열 방향의 연산을 수행한 필터링 값을 그림 4와 같이 몇 개씩 그룹을 만들어서 연속으로 쓰는 방법과 행 방향의 연산을 위해서 데이터를 읽어들이는 때는 그림 5와 같이 외부 메모리의 특성인 열 방향 연속 동작을 이용하여 읽어 들임으로서 연산 모듈이 실시간으로 동작할 수 있도록 한 방법이다.

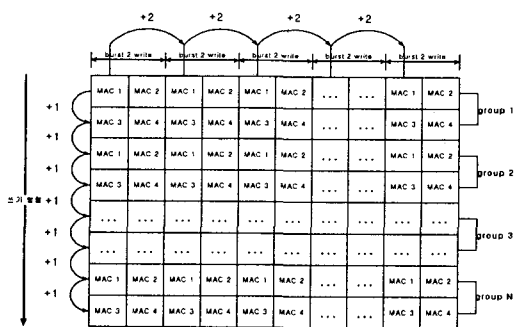


그림 5. 본 논문에서 제안한 쓰기 방법

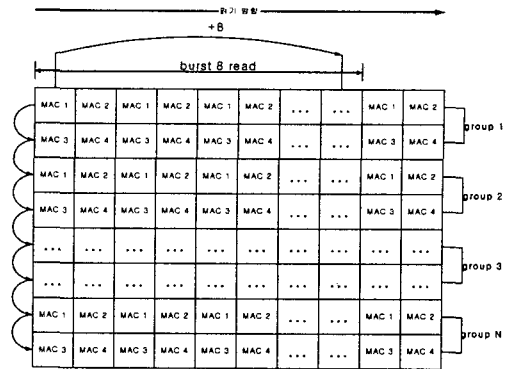


그림 6. 본 논문에서 제안한 메모리 읽기 방법 위의 방법에 의하여 데이터를 읽어 들일 경우 그림 6과 같이 1차원으로 데이터를 배열시킬 수 있게 되어 행 방향의 필터가 수평으로 놓이게 되는 효과를 볼 수 있다.

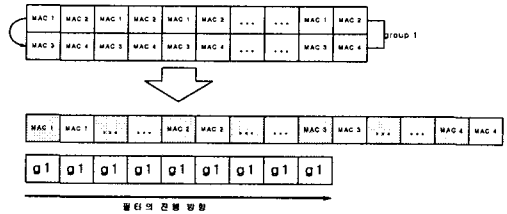


그림 7. 1차원 데이터를 배열한 경우

V. S/W 및 H/W 구현 및 결과

이상에서 설명한 본 논문의 메모리 사상 방법을 구현함에 있어서 본 논문의 내용이 영상압축과 밀접한 관계를 가지므로 영상 압축률에 직접적인 영향을 미치는 양자화 과정을 포함하는 처리 과정을 구현하였다. 양자화기는 H/W 구현의 용이성을 고려하여 그림 7에 나타난 고정 스칼라 양자화기를 설계하여 구현하였다.

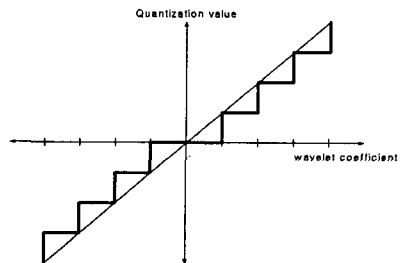


그림 8. 고정 스칼라 양자화기

H/W 구현에 있어서 대상 플랫폼은 Altera FPGA를

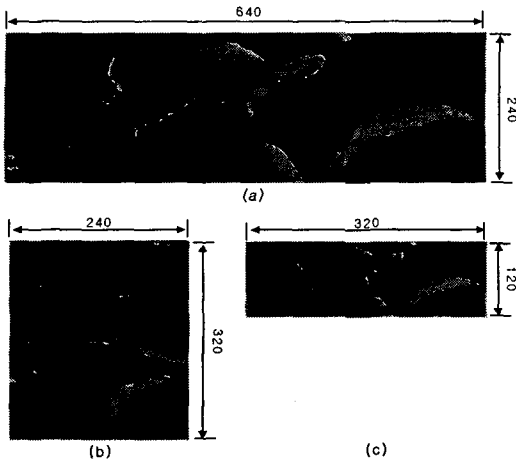


그림 9. 본 논문에서 제안한 사상 법을 적용시킨 C-언어를 이용한 에플레이터 결과 선택하였으며, 설계 툴로는 MAX Plus II를 사용하였고 외부 메모리로는 삼성 512K × 16bit × 2banks SDRAM을 사용하였다. 구현에 사용된 필터로는 Daubechies의 (9,7) bi-orthogonal 필터를 사용하여

NTSC 형식의 640×480 영상을 4-레벨까지 DWT를 수행하는 것으로 하였다. 먼저, C-언어를 이용하여 구현한 MAC, 버퍼 및 다중화기와 메모리 에플레이터를 사용하여 본 논문에서 제시한 메모리 사상방법을 적용, 레벨 1의 열과 행의 DWT를 수행한 결과를 그림 8에 나타내었다.

그림 9에는 메모리 제어기의 H/W 결과를 나타내었다. H/W는 크게 각 레벨에서 DWT에 필요한 주소를 생성해 주는 address generator와 메모리의 동작상태에 따라 메모리에 직접적인 제어신호를 만들어 주는 control signal generator의 두 부분으로 나뉜다. 그림 10에는 시뮬레이션 결과를 나타내었다.

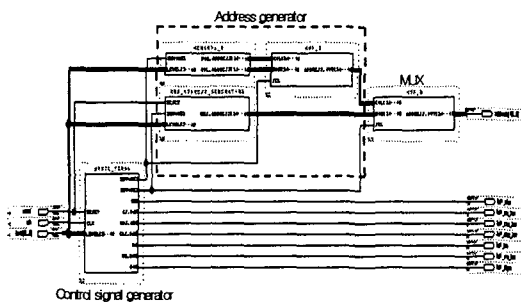


그림 10. Altera MAC Plus+II합성결과

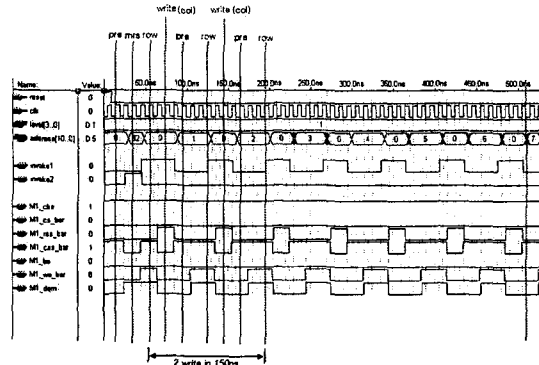


그림 11. 레벨 1의 열 방향 DWT의 쓰기 동작 시뮬레이션 결과

VI. 결론

Mallat-tree 원래의 이론대로 데이터를 저장시키려면 데이터의 실시간 처리를 위해서 외부 메모리의 행 방향이 연속동작으로 읽기를 수행하여야한다. 그러나, 사용 SDRAM은 이러한 기능을 가지고 있지 않기 때문에 칩의 실시간 동작이 불가능하다. 이 경우 본 논문에서 제안한 방법을 적용하여, C-언어를 사용한 에플레이터로 실험한 결과 원래의 Mallat tree이론을 그대로 적용했을 때와 마찬가지로의 결과를 얻을 수 있었으며, 실제 100MHz로 동작하는 상용 메모리를 선정하여, 메모리 제어기를 구현해봄으로서 칩의 실시간 동작을 검증하였다.

따라서, 영상을 압축하는 응용분야에서 FPGA로 H/W를 구현할 경우 본 논문이 제안한 메모리 사상방법이 유용하게 사용될 수 있을 것으로 사료된다.

참고문헌(또는 Reference)

- [1] Gilbert Strang and Truong Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, pp. 337-401, 1997
- [2] Khalid sayood, *Introduction to data compression*, Morgan Kaufmann, pp. 373-404, 2000
- [3] S.Mallat, "A theory for multiresolution signal decomposition: The wavelet representation", *IEEE Trans. on Pattern Anal. Machine Intell.*, vol11, pp. 674-693, July 1989
- [4] 유지상, 김대경외, "웨이블릿 기반 신호처리 시스템 설계", 반도체 설계교육센터, 2000. 3. 22-2000. 3. 24.