

SDRAM을 이용한 이차원 웨이블릿 변환기의 설계

이 선 영*, 홍 석 일**, 조 경 순*
 **㈜에이직프라자, *한국의국어대학교 전자공학과
 경기도 용인시 모현면 왕산리 산 89번지 449-791
 kscho@maincc.hufs.ac.kr

A Design of Two-Dimensional Wavelet Transformer Using SDRAM

Seonyoung Lee*, Seokil Hong**, Kyeongsoon Cho*
 **ASIC Plaza, Inc.,

*Department of Electronics Engineering, Hankuk University of Foreign Studies
 kscho@maincc.hufs.ac.kr

Abstract – The amount of data stored, processed and transmitted in the multi-media systems has been growing very fast, especially for the image data. For example, it takes 0.75 Mbytes to store 512 x 512 pixels of 24-bit color image. A video signal with 30 frames per second will require 22.5 Mbytes of storage space. To solve this problem, we need a good image compression technique. Recently, many researches on the image compression technique based on the wavelet transform are being pursued to overcome the problems of traditional JPEG. This paper describes the architecture and design of two-dimensional wavelet transform circuit. To keep the size of the circuit small, we tried to minimize the internal storage space by using external SDRAM. This circuit was designed in Verilog-HDL, synthesized using Design Compiler and verified using Verilog-XL.

1. 서론

멀티미디어 시스템이 저장, 처리 및 전송해야 하는 자료의 양은 매우 빠른 속도로 증가하고 있다. 특히 영상 처리를 하는 경우 자료의 양은 방대하다. 예를 들어서 512 x 512 개의 화소를 갖는 24 비트 칼라 영상의 경우 0.75M 바이트를 필요로 한다. 1초당 30 프레임으로 이루어지는 동영상의 경우 22.5M 바이트가 필요하다. 이러한 문제점의 해결책은 우수한 영상 압축 기술이다. 영상 압축은 DCT에 기반을 둔 JPEG, MPEG 등의 방법이 대표적이다. 최근 웨이블릿 변환[1]에 기반을 두는 영상 압축 방법이 많이 연구되고 있다. JPEG에 비해 압축률이 높을 때 화질이 더 우수하고, 변화가 크고 다양한 즉 잡음이 많은 통신 채널에 적응성이 강한 장점이 있다.

본 논문은 이차원 웨이블릿 변환기의 구조 및 설계를 기술하고 있다. 이 회로는 향후 영상 압축기의 한 모듈로 사용될 것이다. 실시간 변환을 위해서 시스템의 이들을 이용한 구조[2,3] 등이 제안된 바 있으나 회로의 규모가 크다. 본 논문은 SDRAM을 사용함으로써 칩 내부의 자료 저장을 최소화하는 칩단 방식을 택하였다. Verilog-HDL[4]을 사용하여 회로 설계를 하였으며, 삼성 전자[5]사의 0.5 μ m Standard Cell을 바탕으로 Synopsys 사

의 Design Compiler 를 사용하여 논리 수준 회로를 합성하였고, Cadence 사의 Verilog-XL을 사용하여 검증하였다.

2. 웨이블릿 변환 이론

JPEG, MPEG 등의 영상 자료 압축 알고리즘은 DCT (Discrete Cosine Transform)과 같은 푸리에 변환(Fourier transform)을 이용한다. 웨이블릿 변환은 푸리에 변환의 기본적인 특성에 시간과 주파수에 대한 국부성(locality)을 가진 신호를 표현할 수 있고, 인간의 시각 특성과 비슷한 다중 해상도(multiresolution)로 신호를 표현한다는 장점을 갖는다. 이를 위해 저주파 성분 분해에는 폭이 넓은 윈도우를 사용하고 고주파 성분 분해를 위해 폭이 좁은 윈도우를 사용한다. 웨이블릿 변환은 아래의 식과 같이 웨이블릿 기저 함수(mother wavelet basis function)의 확대, 축소 및 평행 이동과 데이터와의 컨볼루션을 통해 이루어진다.

$$c_j(n) = \sum_{k=0}^{N-1} h(k-2n)c_{j-1}(k)$$

$$d_j(n) = \sum_{k=0}^{N-1} g(k-2n)c_{j-1}(k)$$

여기서 $h(k)$ 는 저역 통과 필터, $g(k)$ 는 고역 통과 필터를 나타내며 다음 식과 같은 관계를 갖는다. $\tilde{h}(k)$ 와 $\tilde{g}(k)$ 는 변환된 신호를 역변환하여 원래의 신호를 복원하기 위하여 사용하는 필터를 의미한다.

$$\tilde{g}_n = (-1)^n \tilde{h}_{-n+1}$$

$$\tilde{g}_n = (-1)^n \tilde{h}_{-n+1}$$

영상 압축을 위해서는 이차원 이산 웨이블릿 변환을 그림 1과 같이 반복적으로 적용하여 주파수 대역별로 분해한다. 본 논문에서는 계산 시간과 압축을 고려하여 옥타브 1의 행 방향의 이산 웨이블릿 변환 결과인 LI

과 H1의 모든 계수에 대해서 열 방향 변환을 적용하지 않고 그림 2와 같이 L1에 대해서만 적용하였다. 나머지 옥타브에 대해서도 같은 개념을 적용하였다. 이 방법으로 회로를 설계할 경우 3개의 옥타브까지 웨이블릿 계산을 한다고 했을 때, 27.08%의 계산 시간을 절감할 수 있고, 압축율이나 PSNR은 원래 웨이블릿 변환에 비해 약 1% 내외의 손실만 있는 것으로 판명되었다.

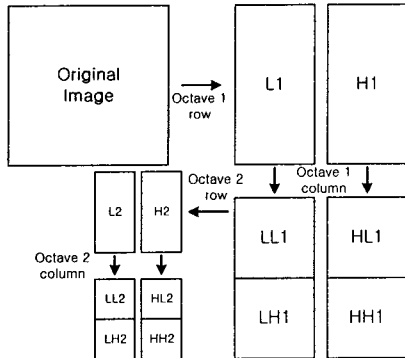


그림 1. 일반적인 이차원 웨이블릿 변환

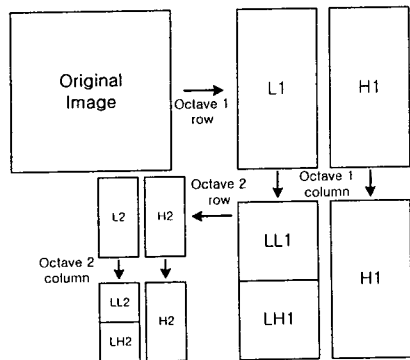


그림 2. 계산량을 줄이기 위한 이차원 웨이블릿 변환

3. SDRAM의 동작

웨이블릿 변환을 하려면 웨이블릿 변환 도중의 자료 및 최종 결과를 저장하기 위한 자료 저장 장소가 필요하다. 칩 내에 SRAM과 같은 블록들을 사용하는 것이 가능하지만 칩의 크기가 너무 커지게 된다. 이 문제를 해결하기 위해 본 논문에서는 외부의 SDRAM을 사용하는 방법을 채택하였다. 대부분의 영상 자료 처리 시스템에서는 자료의 저장을 위해서 DRAM을 장착하고 있으므로 이를 이용하자는 것이다. 사용된 모델은 KM416S1120D라는 512K x 16bit x 2 Banks Synchronous DRAM이다.[5]

SDRAM은 표 1과 같이 Burst Length와 CAS Latency 등에 따라 동작이 다르다. Burst Length는 SDRAM에 자료를 입력하거나 읽어올 때 한 명령에 의해 입출력되는 16비트 데이터의 개수이다. RAS와 CAS 신호는 SDRAM에 입출력 데이터의 주소를 주기 위해 사용한다. CAS Latency는 데이터를 SDRAM에서 읽을 때 RAS와 CAS 신호 후 데이터가 출력되는 때를 나타낸다. 표 2는 사용하고자 하는 모드를 설정할 때 입력되는 MRS (Mode Register Set)에 대한 값을 나타낸 것이다. 본 논문에서는 Burst Length=4, CAS Latency=2와 Burst Length=1, CAS Latency=2를 사용하였다. 이에 해당하는 MRS의 11비트 주소는 각각 02200₍₁₆₎과 02000₍₁₆₎이다.

A ₁₀	A ₉	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀
0	W. B. L.	T.M.	CAS Latency			B. T.	Burst Length			

표 1. 모드 레지스터의 내용

Ar	Ar	Type	CAS Latency				AC	Type	Burst Length				
			Ar	Ar	Ar	Ar			Ar	Ar	Ar	Ar	
0	0	Mode Register Set	0	0	0	Reserved	0	Sequential	0	0	0	1	1
0	1	Reserved	0	0	1	-	1	Interleave	0	0	1	2	2
1	0	Reserved	0	1	0	2	-	-	0	1	0	4	4
1	1	Reserved	0	1	1	3	-	-	0	1	1	8	8
Write Burst Length			1	0	0	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Ar Length			1	0	1	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Burst			1	1	0	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Single Bit			1	1	1	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved

표 2. 각 모드에 대한 MRS

그림 3은 Burst Length=4, CAS Latency=2이고 READ하기 위해 사용되는 신호들을 나타낸 타이밍 다이어그램이다. 1번 클럭에서 RAS='0', 5번 클럭에서 CAS='0'이고 1번에서 5번 클럭까지 WE='1'이면 READ모드로서 동작한다. CAS Latency=2이기 때문에 CAS 신호가 '0'으로 활성화 후 2번째 클럭 즉 7번 클럭에서부터 데이터를 SDRAM에서 읽을 수 있다. SDRAM에 데이터를 입출력하기 위한 주소는 총 19비트로 구성된다. SDRAM이 주소 입력에 사용되는 핀은 11비트이다. 그래서 MSB에서부터 11비트를 RAS='0'에 입력하고 나머지 8비트를 CAS='0'일때 입력한다.

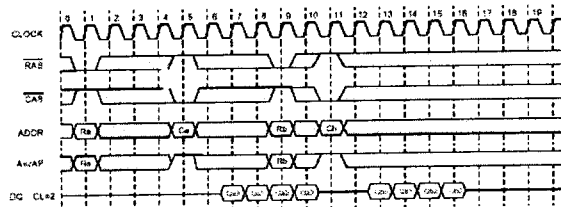


그림 3. SDRAM의 타이밍 다이어그램 예 (READ 경우)

이 때문에 실제 데이터를 SDRAM에 액세스하기 위해 사용되는 클럭 수는 Burst Length=4, CAS Latency=2 일때 읽기/쓰기에 각각 8 클럭이 사용된다. Burst Length=1, CAS Latency=2 일때 읽기/쓰기에 각각 5 클럭이 필요하다.

4. 웨이블릿 변환기의 구조

웨이블릿 변환기는 콘볼루션 계산부, 자료 저장부, 제어부로 구성되어 있으며 전체 블록 다이어그램은 그림 4와 같다. 이 절에서는 각 부분의 동작에 대해서 설명하고자 한다.

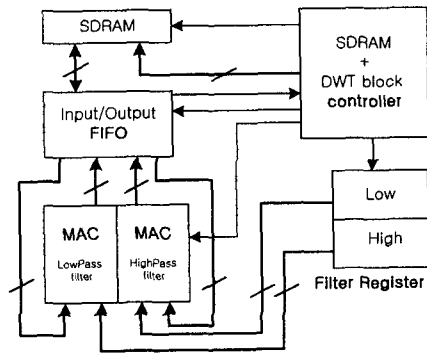


그림 4. 전체 블록 다이어그램

4.1. 콘볼루션 계산부

웨이블릿 변환은 2절에서 설명하였듯이 저대역 및 고대역 필터링이므로 기본적으로 콘볼루션 계산의 연속이다. 알고리즘에서 콘볼루션 계산은 부동소수점(floating point)이지만 회로로 설계할 때는 고정소수점(fixed point)으로 변환하여 수행했다. 콘볼루션 연산에는 저주파 및 고주파 필터 계수 레지스터와 각 영역 연산을 위해 2개의 MAC. 그리고 계산된 결과를 저장하기 위해 24 비트의 레지스터를 사용했다.

Daubechies의 9 탭 필터를 사용하였으며, 필터의 계수를 저장하는 레지스터는 부호 비트를 포함하여 8비트로 구성하였다. MAC은 곱셈기, 덧셈기 및 레지스터로 구성되며, 내부는 24 비트 데이터 폭으로 연산이 이루어진다. 이것은 입력 데이터가 16 비트, 필터 계수가 8 비트 데이

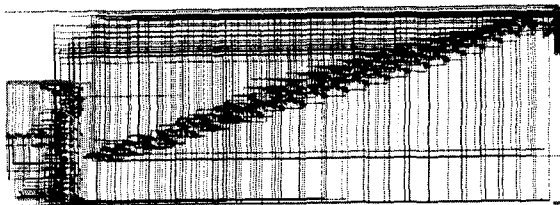


그림 5. MAC 합성 결과

터 폭을 갖기 때문이다. 그리고 SDRAM에 데이터를 저장하기 위해 연산 후 데이터를 $2^7=128$ 로 나누어 16비트의 폭을 갖도록 한다. 그림 5는 MAC의 합성 결과를 보이고 있다. 24비트로 구성된 결과 저장 레지스터에는 필터의 탭 수 만큼 필터 계수와 입력 데이터를 곱한 결과가 저장되며, 최종 결과는 자료저장부인 FIFO에 전달된다.

4.2. 자료 저장부

SDRAM에 저장된 자료를 가지고 와서 콘볼루션 계산을 하거나, 계산된 결과를 SDRAM에 저장하려면 내부에 자료 저장부가 필요하다. SDRAM은 3절에서 설명했듯이 자료의 입출력에 제한이 있다. 자료를 읽고 쓰기 위해 Burst Length에 따라 여러 개의 자료를 저장할 수 있어야 한다. 그러나 칩 내에 많은 저장부를 사용할 경우 칩의 크기가 커지게 된다는 부담이 있다. 이에 본 논문에서는 콘볼루션 계산이 수행될 때까지 16개의 16비트 FIFO에 SDRAM에서 읽은 자료를 저장하도록 하였다. 계산된 결과는 8개의 16비트 FIFO에 임시로 저장되었다가 SDRAM으로 출력하도록 하였다. 콘볼루션 계산에 2개의 MAC을 사용하기 때문에 FIFO도 각각 8개씩 총 16개가 필요하다. 따라서 자료 저장부는 모두 32개의 16비트 FIFO로 구성된다. 그림 6은 이 회로에 대한 합성 결과를 나타내고 있다.

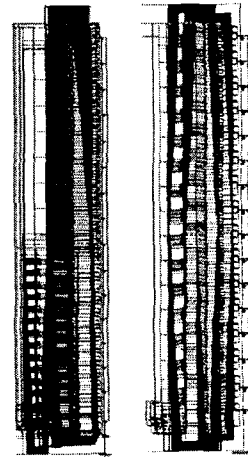


그림 6. 자료저장부 합성 결과 (왼쪽:입력, 오른쪽:출력)

4.3. 제어부

4.1절과 4.2절에서 언급한 모듈들을 총괄적으로 제어해 주는 부분이다. 자료의 입출력과 계산은 각각 독립적으로 이루어질 수 있다. 예를 들어 자료를 SDRAM에서 읽거나 쓰면서 콘볼루션 연산을 할 수 있다. 콘볼루션 연산을 위해서는 일정한 개수의 자료가 자료 저장부에 있어야 한다. 이 때 메모리에 자료를 저장하면서 동시에 읽을 수 없기 때문에 이를 고려하여야 한다.

영상 자료의 콘볼루션과 메모리 액세스는 연산 방향에 따라 제어하는 방식이 다르다. 행 방향 연산에서는 메모리에 자료를 액세스하는 것 보다 콘볼루션 계산을 수행할 때 필요한 클럭의 수가 더 많다. 반면 열 방향 연산에서는 콘볼루션 연산보다는 메모리 입출력에 더 많은 클럭을 필요로 한다. 행 방향 연산의 경우 SDRAM을 Burst Length=4, CAS Latency=2로 4개의 데이터를 액세스 하는데 8개의 클럭이 사용된다. 열 방향 연산의 경우는 Burst Length=1, CAS Latency=2이므로 4개의 데이

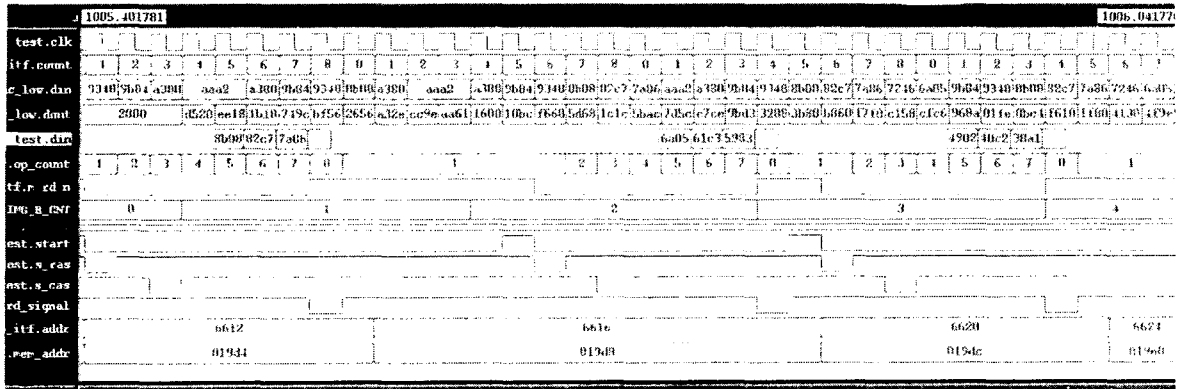


그림 8. Lena256의 시뮬레이션 결과



그림 7. 제어부 합성 결과

터블을 읽거나 쓰는데 4*5=20 클럭이 필요하다. 메모리에서 저장되어 있는 자료가 8비트의 그레이 스케일이기 때문에, 옥타브 1의 행 방향 연산의 경우 8개 클럭에 8개의 데이터를 읽어올 수 있다. 콘볼루션 연산은 한 사이클 연산하는데 9클럭이 필요하다. 행 방향 콘볼루션에서는 8개의 데이터를 묶어서 데이터 입출력하고 열 방향에서는 데이터를 2개씩 묶어서 자료를 액세스하였다.

5. 실험 결과

본 논문에서 제안한 이차원 이산 웨이블릿 변환기의 구조를 Verilog-HDL로 구현하였다. HDL로 기술한 회로는 삼성전자의 0.5μm Standard Cell을 바탕으로 Synopsys사의 Design Compiler를 사용하여 논리 수준 회로로 합성하였고, Cadence사의 Verilog-XL을 사용하여 검증하였다. 합성된 회로는 약 14,000개의 게이트로 구성되었다. 논문 [2],[3]에서 언급된 회로의 게이트 수가 75,000개이므로 이와 비교했을 때 집 크기가 매우 작다는 것을 알 수 있다. 표 3은 회로를 구성하는 각 모듈에 대한 게이트 수를 나타내고 있다.

본 논문에서 사용한 알고리즘을 C언어로 구현하여 Lena256 정지 영상에 대해서 얻은 결과를 이 회로에 대한 시뮬레이션 결과와 비교함으로써 회로의 동작을 검증하였다. 전체 회로에 대한 시뮬레이션 결과는 그림 8에 나타내었다. 제안된 회로에 의해 256 x 256 크기의 Lena256을 처리하는데 849,560개의 클럭이 사용되었다.

따라서 54Mhz 클럭을 사용한다면 15.72ms의 시간이 필요하다. 256 x 256의 영상의 경우 1초에 약 63 프레임의 변환이 가능하므로 1초에 30 프레임의 변환을 필요로 하는 동영상의 처리에도 적용될 수 있다.

6. 결론

본 논문에서는 외부의 SDRAM을 이용한 이차원 이산 웨이블릿 변환기의 구조를 제안하였으며, 이를 Verilog-HDL로 구현하고 합성한 결과를 기술하였다. 256 x 256의 영상을 변환하는데 15.72ms의 시간이 소요되었으며, 총 14,000개의 게이트만으로 회로를 합성할 수 있었다. 이 구조는 기존에 제안된 방법에 비해서 회로의 크기가 현저하게 작을 뿐만 아니라 동영상에 대해서도 실시간 처리가 가능하다는 장점을 갖는다.

이 회로는 향후 영상 압축기의 한 모듈로 사용될 것이며, FPGA를 사용하여 검증할 계획이다. 확보된 영상 압축기는 무인 감시 시스템, 디지털 카메라 등에 내장될 점의 코어로 활용될 예정이다.

블록 이름	게이트 수
콘볼루션 계산부 (MAC)	4113
자료 저장부 (Input/Output FIFO)	4722
제어부 (Controller)	3620.45
그 외	1383.5
합 계	13838.95

표 3. 모듈별 게이트 수

참고 문헌

[1] S. G. Mallat. "A Theory for Multiresolution Signal

SDRAM을 이용한 이차원 웨이블릿 변환기의 설계

- Decomposition: The Wavelet Representation." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674-693, July 1989.
- [2] M. Vishwanath, R. M. Owens and M. J. Irwin, "VLSI Architectures for the Discrete Wavelet Transform." *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, 42(5):305-316, May 1995.
- [3] A. Grzeszczak, M. K. Mandal, S. Panchanathan and T. Yeap, "VLSI Implementation of Discrete Wavelet Transform," *IEEE Transactions on VLSI Systems*, 4(4):421-433, December 1996.
- [4] D. E. Thomas and P. R. Moorby, *The Verilog Hardware Description Language*, Kluwer Academic Publishers, 1997.
- [5] Samsung Electronics, *KM416S1120D: 1M x 16 SDRAM*, Revision 1.4, June 1999.