

論文2000-37SD-3-9

MPEG-2 비디오 인코더의 프레임 메모리 구조

(The Architecture of the Frame Memory in MPEG-2 Video Encoder)

徐 基 範 *, 鄭 正 和 *

(Ki-Bum Suh and Jong-Wha Chong)

요 약

본 논문에서는 MPEG-2 비디오 인코더의 프레임 인터페이스 모듈에 대한 효율적인 하드웨어 구조를 제안한다. 인코더 모듈과 SDRAM 사이의 메모리 버퍼 크기를 줄이기 위해, 한 매크로 블록에 필요한 메모리 액세스 시간을 dual-bank 동작과 버스트 길이 변화를 사용하여 필요 클럭 수를 최소화 한다. 이 최소화된 메모리 액세스 방법으로 인해 남은 클럭 사이클을 랜덤 액세스 횟수로 할당함으로써, 내부버퍼 크기, 데이터 버스의 폭과 제어논리회로의 크기를 줄일 수 있었다. 제안된 프레임 메모리 모듈은 54MHz의 주파수에서 동작하며 설계된 라이브러리는 VTI™ 0.5μm CMOS TLM 표준셀공정을 사용하였다. 제안된 구조를 C-code 하드웨어 모델에 의해 생성된 테스트 벡터와 합성된 회로의 모의실험 결과를 비교함으로써 검증하였다. 제안된 구조의 버퍼 면적은 기존 구조의 버퍼 면적의 40%로 줄일수 있었다.

Abstract

This paper presents an efficient hardware architecture of frame memory interface in MPEG-2 video encoder. To reduce the size of memory buffers between SDRAM and the frame memory module, the number of clocks needed for each memory access is minimized with dual bank operation and burst length change. By allocating the remaining cycles not used by SDRAM access, to the random access cycle, the internal buffer size, the data bus width, and the size of the control logic can be minimized.

The proposed architecture is operated with 54MHz clock and designed with the VTI™ 0.5μm CMOS TLM standard cell library. It is verified by comparing the test vectors generated by the c-code model with the simulation results of the synthesized circuit. The buffer area of the proposed architecture is reduced to 40 % of the existing architecture.

I. 서 론

멀티미디어 기술의 핵심은 방대한 정보량을 갖는 영상 데이터를 효과적으로 압축하여 전송하고 저장하는 것이다. 이것을 목적으로 MPEG(moving picture expert

group)에서는 동영상 압축방식의 표준 안으로 움직임 보상 기법과 변환 부호화 기법을 기반으로 하는 알고리즘을 채택하였다. MPEG-2의 표준화가 1994년 완성된 이후, 이 알고리즘을 적용한 하드웨어가 개발되어 왔으며, 많은 칩 제조회사에서 MPEG-2 복호화기 상용화에 성공하였다. 한편 부호화기는 몇몇 회사들이 상용화를 하였으나, 이들 부호화기에서는 외부 메모리 액세스 시간이 많이 소요되고, 제한된 시간 내에 액세스 사이클을 수용해야 하기 때문에, 데이터의 병목현상이 발생한다. 이것을 해결하기 위해 기존의 인코더^[1]에서

* 正會員, 漢陽大學校 電子工學科

(Dept. of Electronics Eng., Hanyang University)

接受日字:1998年12月2日, 수정완료일:2000年2月28日

는 내부 버퍼 메모리 사용량을 늘리게 되기 때문에, 최종 칩 면적의 크기가 커지는 단점이 있었다. 본 연구에서는 기존 구조의 과다 메모리 사용의 문제점을 해결할 수 있는 MPEG-2 Video 부호화기의 프레임 메모리 하드웨어 구조를 제안하고 그 구조 설계와 검증을 수행하였다. 과다하게 사용되는 내부 버퍼량을 줄이기 위해, 듀얼 뱅크 SDRAM을 사용하며, 매크로블록 단위 당 필요한 메모리 액세스량을 설정한 후, 이 메모리 액세스 방법을 dual bank 동작과 버스트 길이 변화를 사용하여 필요 클럭 수를 최소화 하였다. 버퍼 크기를 줄이기 위해, 이 최소화된 메모리 액세스 방법으로 인해 남은 클럭 사이클을 랜덤 액세스 횟수로 할당함으로써, 내부버퍼 크기, 데이터 버스 폭과 제어 논리회로의 크기를 줄였다. 제 II장에서는 본 구조가 적용되는 MPEG-2인코더의 구조를, III장에서는 SDRAM을 사용한 프레임 메모리부의 구조개선 방법에 대하여 설명하며, 마지막으로, IV장, V장에서는 실험과정과 결론을 제시한다.

II. 인코더 시스템 구조

제안된 MPEG-2 프레임 메모리 모듈이 사용되는 인코더 시스템은 그림1과 같이, 영상을 입력으로 받아들이는 입력처리(IP) 모듈, 움직임 보상과 추정을 위한 MEMC(Motion Estimation and Motion Compensation) 모듈, 부호화하는 영상을 전체적으로 제어하기 위한 제어모듈, 그리고 각 모듈에서 발생하는 데이터를 저장하고 필요한 시기에 전달하는 프레임 메모리(FM) 모듈로 구성되어 있다.

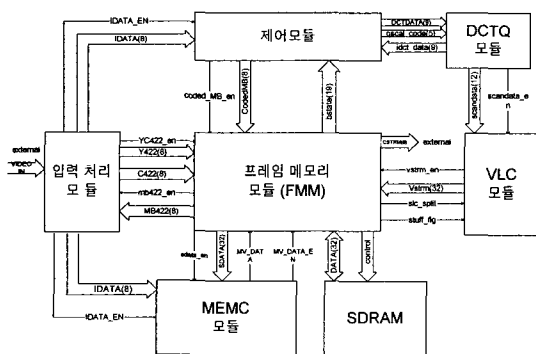


그림 1. MPEG-2 인코더 블록도

Fig. 1. The block diagram of MPEG-2 encoder.

IP 모듈은 외부로부터 들어오는 4:4:4 또는 4:2:2 형

식의 NTSC신호 표본 영상을 4:2:0형식의 매크로블록 영상으로 생성시키기 위한 데시메이션 필터 역할을 한다. FM모듈은 IP모듈의 Y422, C422 신호선을 통해 래스터 스캔 형식의 영상을 받아들여, 이 데이터를 SDRAM의 원 영상 입력 버퍼 영역(Original Y, CbCr)에 저장한다. 이 영역의 데이터는 FM 모듈에서 매크로 블록 단위의 영상 데이터로 추출되며, MB422 신호를 통해 IP 모듈로 전송된다. IP 모듈은 전달된 4:2:2 영상을 4:2:0영상으로 처리하여 그 결과를 제어모듈과 MEMC 모듈에 IDATA신호선으로 전달한다. MEMC 모듈은 휘도 신호에 대한 움직임 추정과 색도신호에 대한 움직임 보상의 역할을 한다. FM 모듈은 32bit의 SDATA신호선을 통해, 한 개 매크로 블록 당 6개 참조 매크로 블록을 전송하여 MEMC모듈이 움직임 추정을 가능하게 해주며, 움직임 추정 모듈로부터 움직임 벡터의 계산이 끝났을 때, 그 매크로 블록의 색도 신호를 움직임 추정 모듈로 전송한다. MEMC 모듈로부터 출력되는 PDATA는 현재 매크로 블록의 예측 블록 데이터이다. 제어 모듈은 이 데이터 흐름을 제어하는 역할을 수행한다. IDATA는 현재 매크로 블록의 픽셀 값을 전달하는 신호선이며, PDATA는 예측 블록의 신호선이다. 제어 모듈은 매크로 블록이 움직임 보상 형식인 경우는 IDATA와 PDATA의 차 값을, 화면 내 부호화인 경우는 IDATA 값을 DCTDATA(DCT의 입력)신호선을 통해 변환기 모듈에 전달한다. 이때 제어모듈은 전송률 제어를 위하여, 양자화 값을 함께 전송하여, 변환기 모듈이 2-D DCT 와 양자화, 역 양자화, 2-D IDCT를 수행할 수 있게 한다. 변환기 모듈은 IDCT된 데이터를 IDCTDATA신호선을 통해 제어 모듈로 돌려주며, 제어 모듈은 움직임 보상처리를 하여 화면을 재구성한다. 즉, 제어모듈은 움직임 보상이 있는 경우에는 PDATA와 IDCTDATA의 더한 결과를, 화면 내 부호화가 된 경우에는 IDCTDATA를 codedMB(부호화된 매크로 블록) 신호로 전달하여 움직임 보상을 수행한다. 움직임 보상된 codedMB 신호는 FM모듈에 의해 다음 픽처의 예측 블록으로 사용되기 위해 SDRAM의 프레임 버퍼영역에 저장된다. 또한 변환기 모듈은 양자화된 DCT값을 스캔 데이터 형식으로 scandata신호선을 통해 VLC모듈로 전달한다. VLC 모듈은 영상에 대한 비트스트림을 생성하는 역할을 하며, 변환기 모듈을 통해 들어온 zigzag 또는 교번(alternate) 스캔 형식의 스캔 데이터를 런 길이 코딩과 가변길이 코드 코딩을 이용해 비트

스트림으로 생성한다. 생성된 비트스트림은 Vstream 신호선을 통하여 FM 모듈에 전달되며, FM은 고정 비트율로 비트스트림을 출력하기 위하여, 비트스트림을 SDRAM의 채널 버퍼영역에 저장하였다가, 적절한 시간에 SDRAM 으로부터 다시 읽어와 고정 비트율의 비트스트림을 cstream 신호선으로 출력한다. 이 논문은 기존의 인코더에서 상당한 면적을 차지하였던 FM 모듈의 구조 제안과 설계를 통하여 인코더의 성능을 개선하는 것을 목적으로 한다.

III. 제안된 프레임 메모리의 구조

프레임 메모리 모듈은 외부의 SDRAM 을 제어하여 부호화 과정에서 발생하는 데이터의 요구에 적시에 공급해주는 역할을 한다. 프레임 메모리 모듈은 입력되는 데이터를 내부 버퍼에 담아두었다가 SDRAM의 접근 스케줄링에 따라 각 데이터를 32 비트씩 묶어 SDRAM 에 저장해 두고, 외부 모듈의 데이터의 공급요구에 따라 SDRAM 으로부터 읽어와 내부 버퍼에 저장하였다가 데이터를 요구하는 모듈에 공급해 준다. 그러므로, 그림 2와 같이 프레임 메모리 모듈은 인코더 모듈과의 입출력을 제어하는 각 서브 모듈들과 서브 모듈에 동작시간을 관리하는 시퀀스 디코더(Sequence Decoder)로 이루어진다. 각 서브 모듈은 SDRAM 과의 데이터를 주고 받는 제어기와 내부 버퍼로 구성되며, 프레임 메모리부의 설계는 내부 버퍼의 크기와 제어기의 제어 논리를 최소화 하는 것을 목적으로 하였다.

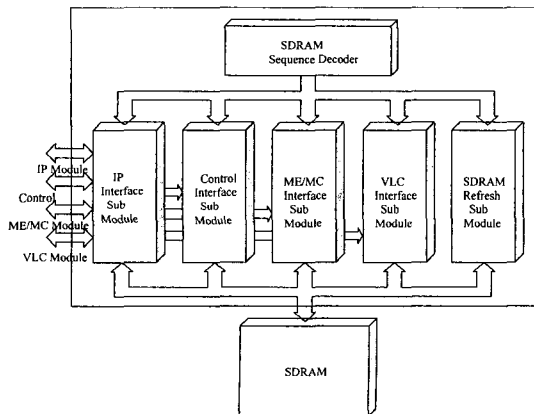
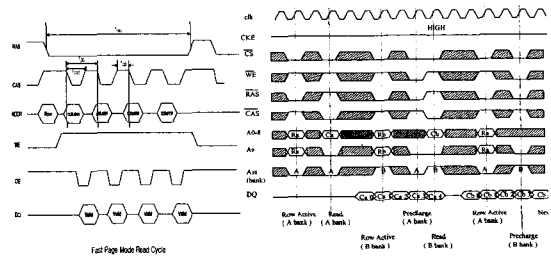


그림 2. FM 모듈의 블록도
Fig. 2. The block diagram of Frame memory Module.

제안된 MPEG-2 인코더는 최대 입력 영상의 크기가 720x480 이고, 한 화면의 매크로 블록의 수는1550, 초당 프레임 수는 30 이다. 따라서, 한 매크로 블록 당 할당된 처리시간은 27MHz일 때, 660 clock이며, 54MHz 일 때, 1320 클럭이다. 따라서 제안된 구조에서는 1320 클럭 분의 메모리 액세스 시간을 스케줄링 하는데 있어, 외부 메모리의 버스 폭을 줄이고, 그 인터페이스를 위한 내부 SRAM의 크기를 줄이며, 제어기의 논리를 간편하는 것을 목적으로 설계하였다. 기존의 FM 의 구조^{1), 3)}는 27MHz의 주파수에서 동작하며, 64bit 폭의 DRAM을 사용하는데, 제안된 구조에서는 동기 클럭 54MHz를 이용하여, 32bit폭 SDRAM을 사용한다. 사용된 SDRAM은 KM4132G512 CMOS SGRAM(256k X 32bits X 2bank)이며, 듀얼 뱅크동작과 버스트 길이 변화를 사용하였으며, 연속적인 데이터를 액세스 할 때 제어부가 간편해지는 SDRAM의 특징을 최대한으로 이용할 수 있도록 설계하였다. 그림 3(a)와 같은 DRAM의 읽기 동작의 경우 DRAM은 비동기적으로 동작하므로 데이터를 RAS 신호의 하강에지에서 행 주소를 받아 들이고 CAS 신호의 하강에지에서 열 주소를 받아 들인다. 따라서 연속된 데이터를 읽더라도 읽을 때 마다 CAS(column address strobe)신호와 열 주소를 변경하여야 한다. 하지만, SDRAM 에서는 CAS신호와 주소의 변경을 한번만 수행하더라도 한 행의 모든 데이터를 접근할 수 있다. 따라서 FM의 구조는 매우 간단하고 신호 변화도가 낮기 때문에 자연적으로 저전력으로 구현된다.



(a) DRAM (b) SDRAM
그림 3. DRAM과 SDRAM의 read cycle 비교
Fig. 3. The comparison of DRAM and SDRAM read cycle.

또한 dual bank 의 특징을 잘 활용함으로써, 뱅크간에 연속적인 액세스 동작의 랜덤 액세스 지연시간을

숨기게 하였다. 54MHz의 시스템 클럭을 사용하는 경우, 읽기 랜덤 액세스 시간은 4 사이클이며, 쓰는 것은 2 사이클이다. 이러한 임의의 데이터를 접근하는데 사용되는 랜덤 액세스 시간(Random Access Time)은 메모리 사용시간을 길게 하여 메모리 병목 현상을 일으키게 한다. 그림 3(b)에서는 SDRAM의 읽기 동작을 보여주고 있다. 여기서 Bank A와 Bank B의 데이터를 읽어내고 있다. A10의 신호는 0일때 Bank A의 값을 가리키고 1일때는 Bank B를 가리킨다. 그림에서 Bank A를 읽어내기 위하여 행 주소(Ra)의 신호를 주소신호로 주고(Row active) 두 cycle 후에 Bank A의 열주소(Ca)를 주면(Read) 두 Cycle후에 출력(DQ)를 통해 데이터값(Ca0,Ca1,Ca2)들을 차례로 읽어 내게 된다. 따라서 처음에 임의의 데이터를 접근하는 랜덤 액세스 시간은 4 cycle이 된다. 그런데 다른 Bank의 데이터를 읽어내는 경우 그림 3(b)에서 보이는 것과 같이 bank A를 읽어내는 기간중 Bank B의 Row Active와 Read를 먼저 해줄수 있기 때문에 랜덤 액세스시간이 필요하지 않고 다음 Bank로 전환하기 위한 프리차지(Precharge) 한 사이클만을 사용함으로써 액세스가 가능하다. 이러한 동작을 듀얼 Bank 동작이라 정의하며, 메모리 맵 설계는 듀얼 Bank 동작의 사용을 최대한 이용할 수 있도록 설계하여, 각 동작에 할당된 시간을 최소화 하였다.

1. SDRAM의 메모리 맵

그림 4는 SDRAM를 사용한 MPEG-2인코더의 제안한 메모리 맵이다. 사용한 SDRAM은 삼성 KM4132 G512 CMOS SGRAM(256k X 32bits X 2bank)으로 1024 개의 행과 256 개의 열을 갖는 맵을 2 개 가지고 있다.

이 SDRAM은 32비트의 데이터 폭을 지니므로, 한 주소 당 4 픽셀의 값을 저장한다. 래스터 스캔 형식으로 들어오는 원영상을 저장하는 Original Y, Original CbCr영역에는 IP 모듈에서 들어오는 4:2:2 영상을 저장한다. 이 영역은 화면에서의 한 라인을 메모리에서의 한 행에 저장하게 구성하여, 메모리 주소를 쉽게 구현하게 하였으며, 짝수 라인은 B Bank에, 홀수 라인은 A Bank에 저장함으로써, dual Bank 동작이 최대한 사용될 수 있도록 하였다. 이 영역의 데이터는 프레임의 내용을 래스터 스캔형식으로 저장하며, 코딩을 위하여 매크로 블록 단위로 IP 모듈로 전달된다. Coded Y와 Coded

CbCr 영역은 코딩 된 영상의 저장 공간으로 인코더 loop에서 복원된 영상을 저장한다. Coded Y영역은 180 열부터 243 열까지 위치하며, 메모리 맵에서 한 라인은 한 매크로 블록을 나타낸다. Coded CbCr 영역은 0 열부터 180 열까지 위치하며, Original CbCr영역과 같이 메모리맵 상의 배치가 화면위치와 동일하다. 각각의 Coded Y와 Coded CbCr영역은 3 개 필드 분을 사용하며, 듀얼 Bank 동작을 최대한 이용할 수 있다.

이 영역에 대한 데이터는 제어부 모듈에서 쓰고, MEMC모듈에서 읽어간다. VLC모듈이 발생시킨 비트스트림을 고정률로 외부로 출력하기 위해, 채널 버퍼를 필요로 하게 된다. 따라서 채널 버퍼는 저장한 순서대로 읽어내는 FIFO로 쓰이며, 그림에서 Channel buffer로 표시되어 있다.

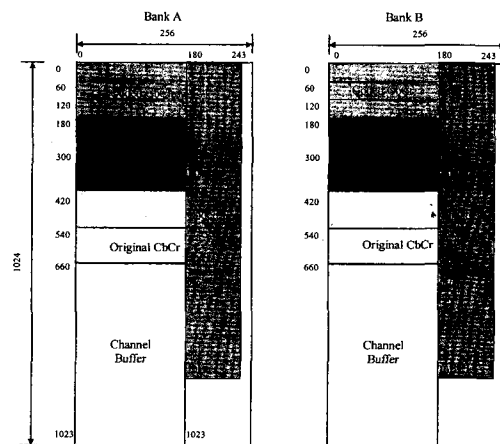


그림 4. SDRAM의 메모리맵
Fig. 4. The memory map of SDRAM.

2. 타이밍 스케줄링

FM모듈에 대한 SDRAM에 액세스 타임의 스케줄링 결과를 표 1에 요약하였다. 스케줄링 목적은 내부 버퍼 크기와 내부 논리를 최소화하는 것이다. 내부 버퍼를 최소화 시키기 위해 FM모듈 내에 있는 각 부 모듈들의 접근 시간을 나누는 방법을 사용하였다. 액세스 시간의 나누는 것은 내부 버퍼 크기를 줄이지만, 전체 액세스 시간을 증가 시킨다. 이것은 여러 번에 나눈 만큼의 랜덤 액세스 숫자만큼 늘어 나기 때문이다. 따라서 내부 버퍼의 크기를 줄이기 위해서는, 메모리 액세스 사이클 수를 줄이는 것이 중요한 문제가 된다.

표 1. SDRAM의 접근시간의 스케줄링
Table 1. The access time scheduling of SDRAM.

Start	End	# of clock	DATA & DATA NUMBER	IN/OUT	Burst Length
1	68	68	CodedY 출력	64 FM→MEMC	256
69	114	46	OriginC 입력	42 IP→FM	256
115	167	53	Vstrm 입력	49 VLC→FM	256
168	303	136	CodedY 출력	128 FM→MEMC	256
304	349	46	OriginY 입력	42 IP→FM	256
350	402	53	Vstrm 입력	49 VLC→FM	256
403	538	136	CodedY 출력	128 FM→MEMC	256
539	591	53	Vstrm 입력	49 VLC→FM	256
592	685	94	OriginC 출력	88 FM→IP	4
686	695	10	Cstrm 출력	6 FM→EXT	256
696	705	10	Refresh		
706	751	46	OriginC 입력	42 IP→FM	256
752	804	53	Vstrm 입력	49 VLC→FM	256
805	850	46	OriginY 입력	42 IP→FM	256
851	940	90	CodedC 출력	50 FM→MEMC	256
941	957	17	No Operation		
958	1010	53	vstrm 입력	49 VLC→FM	256
1011	1080	70	OriginY 출력	64 FM→IP	4
1081	1189	109	CodedYC 입력	96 CTL→FM	256
1190	1199	10	cstrm 출력	6 FM→EXT	256
1200	1252	53	vstrm 입력	49 VLC→FM	256
1253	1320	68	codedY 출력	64 FM→MEMC	256

표 1에 각 동작을 수행하는 클럭 수를 최소화 하기 위해, 듀얼 뱅크 동작과 버스트 (burst) 길이 변화를 적용하였다. 256 버스트 모드에서, 그림 3(b)에 보이는 것과 같이 한 뱅크에서 다른 뱅크로 주소 천이가 일어날 때 프리차지(Precharge)가 필요하다. 그러나, 프리차지는 뱅크 천이시의 한 클럭을 필요로 한다. 따라서 뱅크 천이가 일어나는 동작에서는 많은 사이클이 소비된다. 메모리 뱅크의 조사에 의해 버스트 길이 4의 오토 프리차지(auto precharge) 방법을 IP 모듈의 경우에 적용할 수 있었다. 오토 프리차지는 한 번 정의해준 버스트 길이만큼 읽기 작업이나 쓰기 작업을 수행하면 자동적으로 뱅크를 바꾸어 주는 것으로 따로 프리차지 사이클이 필요로 하지 않는다. 따라서 뱅크의 변화가 많은 작업에는 기본 사이클 수를 줄일 수가 있다. 그림 5는 버스트 길이 4로 설정과 버스트 길이 4로 읽어내는 과정, 다시 256 버스트로 설정하는 과정을 보여주고 있다. 그림 3.(b)에는 프리차지 사이클이 존재 하지만, 그림 5에서는 존재하지 않는다. 따라서 프리차지 사이클대신에, 오토 프리차지 방법이 적용함으로써, 기본 사이클 수를 줄여 들게 하였다. 따라서 남은 사이클을 액세스 시간

의 분할에 사용하게 함으로써 내부 버퍼의 크기를 줄일 수 있었다.

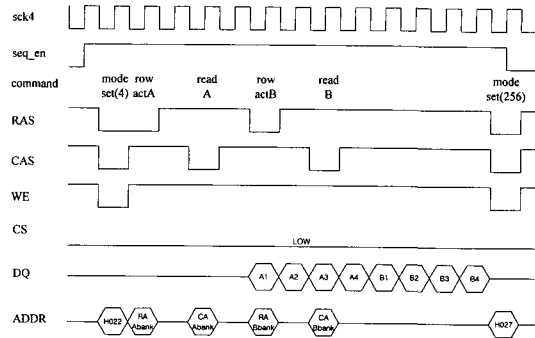


그림 5. 버스트 길이 변화 및 오토 프리 차지
Fig. 5. The burst length change and auto-precharge.

FM 내의 시퀀스 디코더(sequence decoder)는 각 서브 모듈의 SDRAM 액세스 시간을 제어한다. 이것의 스케줄링은 외부 모듈로 나가는 데이터의 시간을 고정하고 SDRAM의 액세스 시간을 배치함으로써 수행하였다. 그림 6은 한 매크로 블록 시간에 대한 스케줄링 결과이다. 여기서 굵은 실 선은 다른 모듈과의 액세스 시간이고 가는 실 선은 스케줄링된 서브 모듈의 SDRAM 액세스 시간이다.

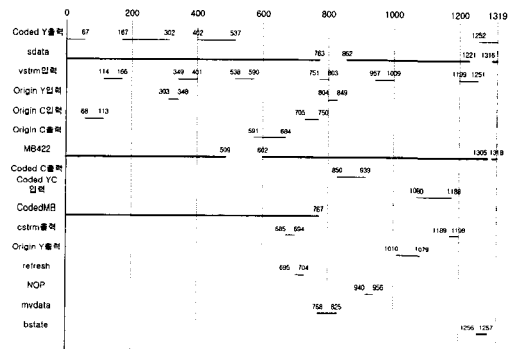


그림 6. 매크로 블록내의 메모리 액세스 시간
Fig. 6. Memory access time in one macroblock time

DRAM의 시간 제한 조건 때문에, 기존의 DRAM을 사용한 구조^[1]는 DRAM의 64 비트의 데이터 폭을 가지며, 할당된 메모리 660 클럭 내에 각 모듈에 필요한 동작을 수행하기 위하여 할당되는 액세스 타임의 한계 때문에 동작의 분할이 불가능하였다. 따라서 사용되는 내부 메모리는 많은 양을 필요로 하였다. 그러나 본 논

문에서는 SDRAM의 특성을 이용하여, 듀얼 뱅크 동작과 버스트 모드의 적절한 선택으로 각 동작 당 필요한 클럭 수를 줄임으로써 그 여유 분의 클럭 수를 여러 번의 랜덤 액세스 시간에 할당하여, 버퍼 크기를 줄일 수 있었다. 또한 데이터 버스 폭을 32 비트로 줄여 스위칭 활동 도를 줄였다.

그 결과 표 2에 보이는 것과 같이, 기존의 구조에 40% 정도의 면적으로 구현이 가능하였다.

표 2. 개선전과 후의 SRAM의 크기와 면적 비교

Table 2. The comparison of SRAM size and area,

Module	I/O	기존 구조[1, 11]		기존 구조[1, 11]	
		buffer size	Buffer area	buffer size	Buffer area
IP module	I N	512*512=10 24bytes	Y 2559.152mil ² C 2559.152mil ²	512*512=10 24bytes	Y 2559.152mil ² C 2559.152mil ²
		O U T	256*384=76 8bytes	Y 2895.81mil ² C 3108.07mil ²	256*384=76 8bytes
MEMC module	O U T	256*3=768 bytes	20.13*10.92*24= 5725.67mil ²	256*3=768 bytes	20.13*10.92*24= 5725.67mil ²
CNTL module	I N	256*256=51 2bytes	20.13*13.46*8=2 167.6mil ²	256*256=51 2bytes	20.13*13.46*8=2 167.6mil ²
VLC module	I N	32*256*2 =2048bytes	60.74*35.10*2=4 263.95mil ²	32*256*2 =2048bytes	60.74*35.10*2=4 263.95mil ²
		O U T	8*16*8=1 28bytes	20.13*8.37*8=13 47.90mil ²	8*16*8=1 28bytes

조에 대한 검증 및 테스트 벡터파일의 생성을 하였다. 이 테스트 벡터는 VHDL로 기술된 하드웨어를 검증 하는데 사용되었고, 전체 시뮬레이션을 수행 중 중간 데이터의 확인을 쉽게 하였다.

설계된 library는 VTI 0.5 um CMOS TLM [13] 공정을 이용하여 설계되었으며 Compass tool을 사용하여 layout 되었다.

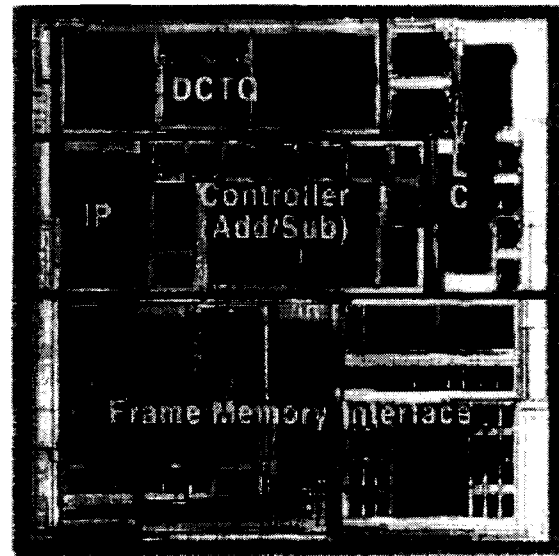


그림 8. 설계된 프레임 메모리의 layout
Fig. 8. Layout of the designed frame memory.

IV. 시뮬레이션과 검증

제안된 구조를 검증하기 위하여, MPEG-2 비디오 부호화기의 하드웨어 구조에 대한 C-코드 모델링을 수행하였다. 이 모델링을 가지고, 인코더모듈의 기능 및 구

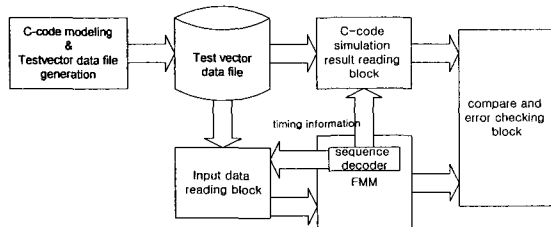


그림 7. 하드웨어 검증을 위한 테스트 벡터의 구성
Fig. 7. The configuration of test bench for verification of hardware.

V. 결론

본 논문에서는, MPEG-2 비디오 인코더 프로세서에서 사용되는 프레임메모리모듈의 효율적인 구조를 제안하였다.

첫 번째로, 외부 메모리 사용을 제어하는 프레임 메모리 모듈의 효율적인 구조를 제안하였다. 이 구조는 64 비트를 사용하던 기존 DRAM구조에서 32 비트의 SDRAM 구조를 사용하였고 효과적인 메모리 액세스 스케줄링 방법을 적용하여 설계되었다. 이 결과, 데이터 버스 폭과 제어 논리가 작아짐으로써, 전력소모를 줄일 수 있었고, 기존의 구조에 비해 내부 버퍼의 크기를 60%줄일 수 있었다.

제안된 구조는 VHDL을 이용하여 설계되었으며, Mentor tool을 사용하여 행위 레벨로 검증하였고, Layout 은 Compass tool을 사용하였다.

본 논문이 제안한 하드웨어 구조개선은, 기존에 설계되었던 인코더 프로세서^[12] 칩 면적에서 상당한 부분을 차지하던 프레임 메모리 모듈의 면적을 50%로 줄임으로써 전체 칩 면적을 줄일 수 있음을 확인하였다. 개발된 인코더 프로세서는 향 후 광대역 멀티미디어 통신, 디지털 CATV, 디지털 캠코더 등에 활용가능하며, 유무선 멀티미디어 서비스에 대한 휴대용 멀티미디어 단말기등에 적용 가능할 것이다.

참 고 문 헌

[1] Y. K. Ko, K. H. Lee, E. S. Kang, S. H. Lee, S. H. Jang, S. J. Ko, "The design of Frame memory module for MPEG-2 video encoder", summer conference of Korean institute of communication sciences vol 15. No1. pp.452-458, July 1996.

[2] ISO/IEC JTC1/SC29/WG11 13818-2: Moving Picture Experts Group, IS, May 1996.

[3] International Telecommunication Union, ITU-T Draft H.263, Video Coding for Low bitrate Communication, July 1995.

[4] Grand Alliance HDTV System Specification, version 2.0. Princeton, New Jersey, December 7, 1994.

[5] Samsung Electronics co.,LTD. "Graphic Memory", pp.231-278, May, 1997.

[6] "IEEE standard Specifications for the Implementation of 8x8 inverse discrete cosine transforms," *IEEE standard*, pp.1180-1190, March 1991.

[7] S.A. White, "Applications of Distributed Arithmetic to Digital Signal Processing: A Tutorial Review," *IEEE ASSP Magazine*, July 1989.

[8] M. T. Sun, T. C. Chen, and A. M. Gottlieb, "VLSI Implementation of a 16x16 Discrete Cosine Transform Chip", *IEEE Trans. Circuits and Syst.*, pp.610-617, April 1989.

[9] I. Uramoto et al., "A 100-MHz 2-D Discrete Cosine Transform Core Processor", *IEEE J. of Solid-State Circuits*, Vol.27, No.4, pp.492-499, APRIL 1992.

[10] A. Madisetti et al., "A 100 MHz 2-D DCT/IDCT Processor for HDTV Applications", *IEEE Trans. CAS for Video Tech.*, Vol.5, No.2, pp. 158-165, APRIL 1995.

[11] Kyeonsoo Kim, Jong-Sung Yoon, Seong-Hwan Jang, Jae-Ho Hwang, Jai-Sop Hyun, Soon-Hwa Jang, and Soon-Hong Kwon ; "A. VLSI Implementation of MPEG-2 Video Encoder", *5th International Conference on VLSI and CAD*, October 13-15, 1997, Seoul, Korea.

[12] 김건수, 윤종성, 장성환, 황재호, 현재섭, 장순화, 권순홍, "ATM통신용 MPEG-2 비디오 인코더 ASIC 구현", '97 정보통신의 날 기념 정책전략포럼 및 학술발표회 논문집., April 23-25. pp.426-430, Seoul

[13] VLSI Technology Inc., 0.5-micron HDI 3V core cell-based libraries, Sept. 1995.

저 자 소 개



徐 基 範(正會員)
1989년 한양대학교 전자공학과(학사). 1991년 한양대학교 전자공학과(석사). 1995년 한양대학교 전자공학과 박사과정 수료. 관심분야는 영상 압축 알고리즘 및 아키텍처 설계, VLSI 설계

鄭 正 和(正會員) 第 36 號 C編 第 10號 參照