DBpia

논문 99-24-9A-21 한국통신학회논문지 '99-9 Vol.24 No.9A

MPEG-2 비디오 부호화기의 프레임 메모리 하드웨어 구현

정회원 고 영 기*, 강 의 성*, 이 경 훈**, 고 성 제*

Implementation of the Frame Memory Hardware for MPEG-2 Video Encoder

Young-Ki Ko*, Eui-Sung Kang*, Kyoung-Hoon Lee**, and Sung-Jea Ko* *Regular Members*

요 약

본 논문에서는 MPEG-2 비디오 부호화기에서의 프레임 메모리 하드웨어 구현을 위한 DRAM의 효율적인 메모리 맵과 이에 따른 하드웨어 구조를 제안한다. 논문에서 제시된 메모리 맵은 DRAM 사용 시간과 대역폭을 줄임으로써 시스템의 성능을 개선할 수 있도록 설정되었고, 개발된 하드웨어는 MPEG-2 비디오 부호화기에 구성된 타 모듈과 인터페이스를 위해 하드웨어 내부에 구성되는 내부 버퍼의 크기와 로직 크기를 줄일 수 있도록 설계하였다. 본 논문에서 설계된 하드웨어는 VHDL을 이용하여 구현하였으며, 0.5 μm, VTI, ASIC 라이브러리인 vcmn5a3을 이용하여 합성하였다. 개발된 하드웨어에 대한 RT (register transfer) 수준 및 게이트 수준의 검증을 위해 VHDL 시뮬레이터와 로직 합성 툴을 사용하였고, 추후 성능 개선과 기능 검증을 위해서 하드웨어 에뮬레이터를 개발하였다. 구현된 하드웨어는 MPEG-2의 MP@ML에서 요구하는 전송률로 데이터를 채널에 전송할 수 있도록 하였다.

ABSTRACT

In this paper, we present an efficient hardware architecture for the frame memory of the MPEG-2 video encoder. Both the total size of internal buffers and the number of logic gates are reduced by the proposed memory map which can provide an effective interface between MPEG-2 video encoder and the external DRAM. Furthermore, the proposed scheme can reduce the DRAM access time. To realize the frame memory hardware, 0.5 μm, VTI, vcmn5a3 standard cell library is used. VHDL simulator and logic synthesis tool are used for hardware design and RTL (register transfer level) function verification. The frame memory hardware emulator of the proposed architecture is designed for gate-level function verification. It is expected that the proposed frame memory hardware using VHDL can achieve suitable performance for MPEG-2 MP@ML.

* 고려대학교 전자공학과
** 한국전자통신연구원 정보보호기술연구본부
논문번호 : 99139-0413, 접수일자 : 1999년 4월 13일

www.dbpia.co.kr

Ⅰ. 서 론

MPEG-2 시스템은 고화질의 동영상, CD 음질의 디지털 오디오, 데이터 및 문자서비스를 제공할 수 있는 종합적인 멀티미디어 시스템이다. 방대한 정보량을 갖는 영상 데이터를 효과적으로 압축하여 저장하고 전송하는 것은 이러한 멀티미디어 시스템의 중요한 요소기술이라고 할 수 있다. 대용량의 하드디스크와 CD, DVD 등의 발달로 저장 문제는 어느 정도 해결이 되고 있지만, 동영상의 경우 제한된 대역폭 내에서 많은 데이터 량을 갖는 동영상을 전송하기 위해서는 효과적인 동영상 처리 방법들이 요구된다. 이를 위해 각종 정보의 표현이나 전송을 위한 국제 표준 규격인 H.261/263, MPEG-1/2들이 제정되었다[1]-[5].

MPEG-2 MP@ML (Main Profile at Main Level)에 해당하는 코딩 칩들이 마쯔시다, C-Cube, IBM, IIT 등의 여러 회사에서 개발되어 판매되고 있고, 이들을 이용하여 구현된 MPEG-2 시스템들이 여러 응용 서비스 등에 이용되고 있다[6]-[9]. 동영상 부호화기와 같이 방대한 양의 데이터를 처리하는 멀티미디어 시스템을 개발하는 하드웨어 개발 기술에서 가장 핵심적인 기술은 효과적으로 외부 메모리를 제어하고 데이터를 처리하는 기술이라 할 수 있다[8]. 효율적인 메모리 사용 방법은 데이터를 처리할 때에 발생하는 병목 현상을 제거할 수 있을 뿐만 아니라 시스템의 동작속도를 개선할 수 있기 때문에 중요하다.

MPEG-2 동영상 부호화기에서 프레임 메모리는 부호화기에 구성된 외부 DRAM을 제어하는 역할을 수행하는데, IP (input processing) 모듈, ME/MC (motion estimation and motion compensation) 모듈, coder 모듈, VLC (variable length coder) 모듈 등과 같은 MPEG-2 부호화기를 구성하는 각 모듈들이 발생시킨 데이터를 DRAM에 저장하고, 각 모듈들이 데이터를 요구할 경우에 DRAM으로부터 해당 데이터를 출력하는 역할을 한다[11]. 본 논문에서는 DRAM을 이용한 프레임 메모리 하드웨어 개발에 관한 내용을 다루고 있으며, 개발된 하드웨어의 규격은 MPEG-2의 MP@ML에 해당한다. 본 논문에서는 동영상 부호화기의 각 모듈들과 인터페이스를 담당하는 하드웨어 블록들의 효율적인 DRAM 사용 방법과 인터페이스 하드웨어에 사용되는 내부 버퍼와 제어 로직의 크기를 줄일 수 있는 하드웨어 구조를 제시하고 있다.

본 논문의 구성은 다음과 같다. Ⅱ장에서는 MPEG-2 비디오 부호화기의 구조에 대하여 설명하고, Ⅲ장에서는 본 논문에서 제안된 DRAM의 메모리 맵과 메모리 맵에 근거한 하드웨어의 구조에 대하여 설명한다. Ⅳ장에서는 VHDL을 이용한 하드웨어에 대한 회로 합성과 하드웨어 에뮬레이터를 이용한 시스템의 동작 및 기능 검증에 대하여 설명하고, Ⅴ장에서 결론을 맺는다.

Ⅱ. MPEG-2 비디오 부호화기

MPEG-2 비디오 부호화기는 그림 1과 같이, 카메라로부터 입력된 영상을 처리하는 IP 모듈, 움직임 보상과 추정을 위한 ME/MC 모듈, 이산여현변환(discrete cosine transform: DCT)과 양자화를 위한 coder 모듈, 가변길이 부호화를 위한 VLC 모듈, 그리고 각 모듈에서 발생하는 데이터를 저장하고 필요한 시기에 전달하는 프레임 메모리로 구성되어 있다[11].

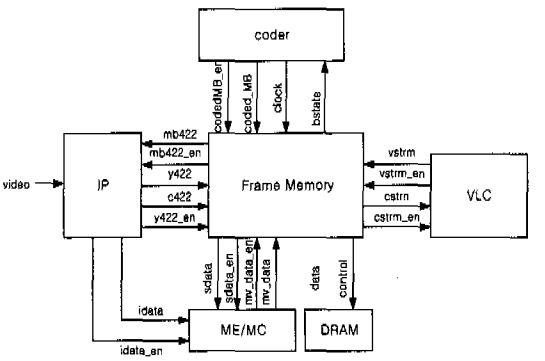


그림 1. MPEG-2 비디오 부호화기의 구조

부호화기에서 IP 모듈은 입력 장치로부터 입력된 다양한 형식의 데이터를 부호화기가 처리하기에 알맞은 형태로 변환하는 역할을 수행한다. IP 모듈이 처리하는 입력 영상은 4:4:4 형식의 RGB, 4:2:2 형식의 YUV, CCIR 601 형식의 데이터들이며, 출력은 4:2:0 형식의 YCbCr이다. 프레임 메모리는 IP 모듈이 변환한 4:2:2 형식의 데이터를 인터페이스 신호인 y422, c422, y422_en 신호를 이용하여 DRAM에 일시적으로 저장한 후, 일정 시간이 지나

www.dbpia.co.kr

다시 IP 모듈로 mb422, mb422_en 신호를 이용하여 전달함으로써 IP 모듈이 4:2:0 형식의 데이터로 변환할 수 있게 한다. IP 모듈에서 출력하는 데이터를 프레임 메모리에 저장하는 것은 4:2:0 형식으로 데이터를 변환하기 위한 필터링을 수행할 수 있도록 하기 위해서이다. IP 모듈에 의해 4:2:0 형식으로 변환된 데이터는 idata, idata_en 신호를 이용하여 ME/MC 모듈로 입력되어, 움직임 추정 및 보상에 이용된다. 움직임 추정과 보상을 수행한 결과로 출력된 데이터는 비디오 부호화기에서 DCT 변환과 양자화를 수행하기 위해 coder 모듈로 입력된다.

coder 모듈은 IP 모듈이 동작하는데 필요한 수평 기준신호 (horizontal reference signal), IP 모듈과 ME/MC 모듈이 필드 단위로 부호화를 하는데 필요한 필드 기준신호 (field reference signal) 뿐만 아니라, 프레임 단위로 부호화를 하는데 필요한 프레임 기준신호 (frame reference signal)들을 생성한다. 또한 coder 모듈에서는 버퍼상태, 부호화방법, 그리고 부호화기 내부의 각 모듈들에서 필요한 각종 타이밍 정보 (clock)들을 제공한다. ME/MC 모듈은 휘도 신호에 대한 움직임 추정과 색차신호에 대한 움직임 보상을 수행하는데, 프레임 메모리로부터 6개의 참조 매크로블럭을 sdata, sdata_en 인터페이스를 통해 입력받고, 움직임을 추정하여 움직임 벡터를 계산하고, 움직임 벡터에 해당하는 예측 데이터를 mvdata, mvdata_en을 이용하여 출력한다.

VLC 모듈은 coder 모듈로부터 입력받은 양자화된 데이터를 고정 길이 및 가변 길이 부호화를 수행하여 MPEG-2 부호화 방식에 맞게 비트스트림을 출력하고 그 결과를 고정 길이의 데이터로 변환하여 프레임 메모리로 전달하는 역할을 수행한다.

Ⅲ. 하드웨어 아키텍처

프레임 메모리 하드웨어는 그림 1의 부호화기 내부에 구성된 IP, coder, VLC 모듈로부터 입력된 데이터를 SRAM으로 구성된 내부 버퍼에 저장한 후, 데이터를 DRAM의 I/O 버스 크기로 묶어서 저장하고, 내부 모듈의 요구에 따라 DRAM에서 읽어내어 하드웨어 내부에 구성된 출력 버퍼에 저장하고 데이터를 요구하는 타 모듈로 전달한다. 그림 2는 프레임 메모리의 하드웨어 구성을 보인 것으로서, 그림 1에서의 IP, coder, ME/MC, VLC 등의 각 모듈들과 인터페이스를 전담하기 위한 인터페이스 서브모듈과 각 인터페이스 서브모듈들의 동작 타이밍을 제어하기 위한 시퀀스 디코더로 구성되어 있다. 각 인터페이스 서브 모듈들은 시퀀스 디코더로부터 전달받은 기준신호들을 이용하여 DRAM을 사용하기 위한 읽기 (OE), 쓰기 (WR), DRAM 어드레스 (DRAMADDR), RAS (row address strobe), CAS (column address strobe) 신호들을 생성한다.

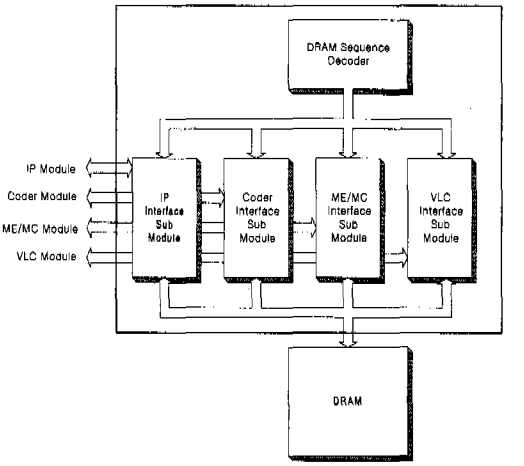


그림 2. 프레임 메모리 하드웨어 구조

MPEG-2 비디오 부호화기는 외부에 DRAM을 설정하여 시스템의 주 메모리로 사용하는데, DRAM의 데이터 처리속도와 저장할 수 있는 용량 등에 의해 부호화기의 성능이 결정될 뿐만 아니라, DRAM에서 데이터를 처리하는 유형에 따라서도 성능이 결정될 수 있다. 따라서, 인터페이스를 통한 데이터 전송이 효율적으로 이루어지도록 메모리 맵이 설계되어야 데이터 대역폭에 따른 병목현상을 제거할 수 있게 되며, 부호화기의 성능 또한 개선될 수 있다. 본 논문에서는 프레임 메모리가 제어하는 DRAM에 대한 메모리 맵을 패스트 페이지 모드 (fast page mode)를 최대로 활용할 수 있도록 설정하고 있다. 그림 3은 DRAM의 랜덤억세스 모드 (random access mode)와 패스트 페이지 모드에 대

www.dbpia.co.kr

한 타이밍도를 나타내고 있다. 그림에서 볼 수 있듯이 랜덤 억세스 모드를 이용한 DRAM 접근 방법은 행에 대한 어드레스와 열에 어드레스를 모두 지정해야 하므로 많은 시스템 클럭을 필요로 하게 되어 시스템의 처리 속도가 느려지게 된다. 패스트 페이지 모드를 이용하면 열에 대한 어드레스가 지정된 후 CAS 신호만을 변화시켜 DRAM을 사용할 수 있다. 따라서 이러한 패스트 페이지 모드를 최대로 이용하게 되면 데이터의 처리 속도를 빠르게 하여 시스템의 성능을 개선할 수 있다. 랜덤 액세스 모드로 데이터를 처리하는 경우에는 최소 3 사이클의 시스템 클럭이 요구되지만, 페스트 페이지 모드의 경우에는 1 사이클의 시스템 클럭으로 DRAM을 접근할 수 있다.

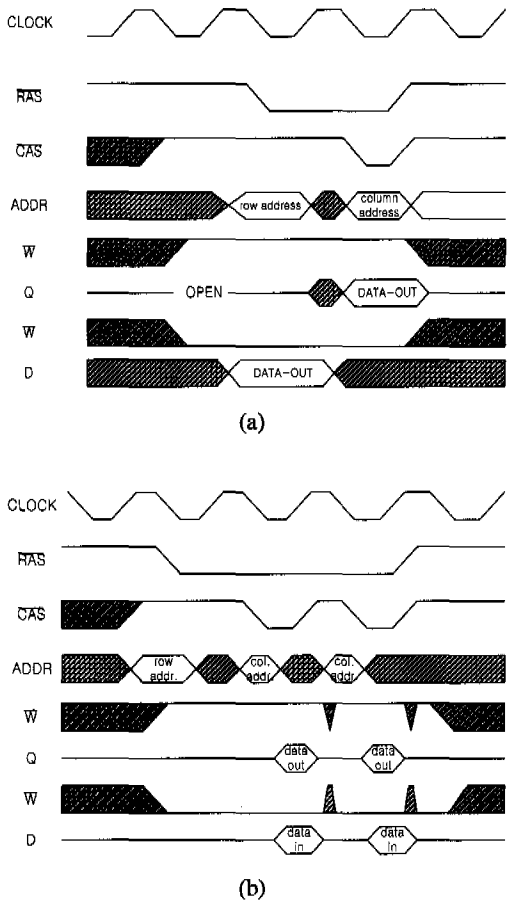


그림 3. DRAM 억세스 타이밍도: (a) 랜덤 억세스 모드, (b) 패스트 페이지 모드

DRAM에 대한 메모리 맵과 메모리 맵에 따른 데이터 처리 방법은 부호화기 내의 각 모듈에서 발생시킨 데이터를 DRAM의 한 줄에 최대로 저장할 수 있도록 하였다. 프레임 메모리가 제어하는 메모리의 사용 내역은 다음과 같다. IP 모듈이 발생시킨 입력 데이터인 4:2:2 형식의 Y (original Y), CbCr (original CbCr) 영상에 대해서는 CCIR 601 형식의 한 프레임 크기를 저장할 수 있는 영역을 할당하고, coder 모듈에 의해 생성된 데이터인 부호화된 Y (coded Y) 영상과 CbCr (coded CbCr) 영상에 대해서는 3 개의 필드를 저장할 수 있는 영역을 확보하였다. 또한 가변장 부호화기 모듈과의 인터페이스를 위한 별도의 채널 버퍼를 구성하여 고정 길이의 데이터로 출력할 수 있는 영역을 할당하였다. 그림 4는 본 논문에서 제안하는 16 Mbits의 DRAM에 대한 메모리 맵을 보인 것이며, 그림에서 18F~1FF의 예비영역은 향후 사용할 오디오 부호화에 대한 결과를 저장하기 위한 공간이다. 그림에서 표시된 수치들은 16진수로 표현된 DRAM의 어드레스들을 의미한다.

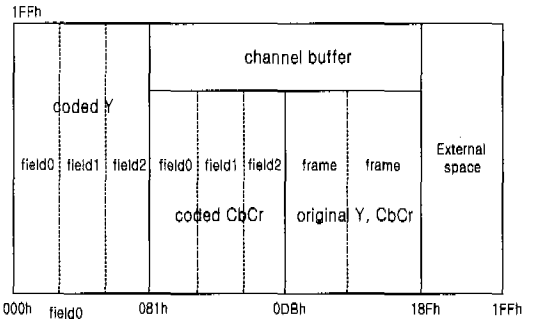


그림 4. 제안된 DRAM 메모리 맵

1. 부호화된 휘도신호의 메모리 맵

그림 4와 같이 지정된 DRAM 영역에 해당 데이터를 저장할 때, 데이터를 저장하는 방법에 따라서 시스템의 성능이 달라질 수 있다. coded Y 영역은 DCT와 양자화된 결과를 저장하기 위한 곳이며 움

www.dbpia.co.kr

직임 추정 시에 매크로블록 단위로 ME/MC 모듈로 전달한다. DRAM에서 해당 매크로블럭을 찾아내어 빠른 시간 내에 처리하기 위해서는 매크로블록에 해당하는 데이터들을 DRAM의 한 줄에 연속적으로 저장하여 패스트 페이지 모드를 최대로 활용할 수 있도록 한다.

그림 5는 데이터의 처리 단위를 매크로블록 또는 영상의 라인단위로 가정하고 데이터를 처리하는 방법을 보인 것이다. 그림 5(a)는 부호화된 휘도영상에서 매크로블럭의 위치를 보인 것이며, 그림 5(b)는 패스트 페이지 모드를 최대로 활용하기 위해 메모리의 한 줄에 연속적으로 저장되는 방법을, 그림 5(c)는 raster 스캔 순서대로 DRAM 영역에 매크로블럭이 저장되는 방법을 보인 것이다. 그림 5(b)와 같이 데이터를 메모리에 연속적으로 저장하면 패스트 페이지 모드를 최대한 활용하게 되어 DRAM의 사용 효율을 증대시킬 수 있다.

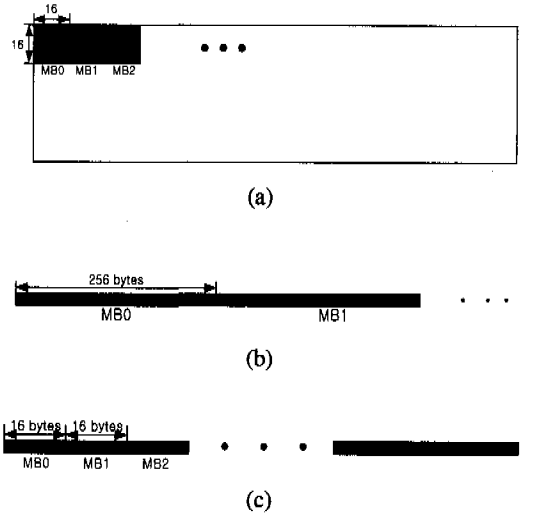


그림 5. coded Y에 대한 메모리 맵: (a) 영상에서의 매크로블럭의 위치, (b) 매크로블럭 단위의 coded Y 처리 방법, (c) 라인단위의 coded Y 처리 방법

만약 그림 5(c)와 같은 방법으로 데이터를 저장하게 되면 랜덤 억세스 모드가 발생할 가능성이 커져 DRAM 사용 효율이 저하된다. 그림 5(b)와 같은 방법으로 패스트 페이지 모드를 최대로 활용하면 1번의 랜덤 억세스 모드와 31번의 패스트 페이지 모드로 하나의 매크로블럭을 처리할 수 있으며, 이 때 필요한 시스템 클럭의 개수는 34이다. 그림 5(c)와 같은 방법으로 데이터를 저장한다고 가정하면 최악의 경우 랜덤 억세스 모드는 3번 그리고, 패스트 페이지 모드가 29번 발생하게 되어 38개의 시스템 클럭이 필요하게 된다. 따라서 본 논문에서는 그림 5(b)의 방법을 이용하여 coded Y 데이터를 처리하고 있다.

2. 부호화된 색차신호의 메모리 맵

coded CbCr 영역에 데이터를 처리하기 위한 방법을 그림 6에 보였다. 그림 6(a)는 색차신호인 CbCr 영상을 보인 것이며, 그림 6(b)는 앞절에서 설명한 coded Y의 경우와 같이 매크로블럭 단위로 데이터를 처리하는 것을 보인 것이다. 그림 6(c)는 영상이 입력되는 순서대로 처리하는 방법을 보인 것이다. coded CbCr은 coder 모듈로부터 매크로블럭 단위로 DRAM에 전달되지만, ME/MC 모듈로 전달될 때는 움직임 벡터의 값에 따라 매크로블록 단위로 전달되지 않는 경우가 발생하게 된다. 움직임 추정 결과로 매크로블럭의 경계를 벗어난 위치에 해당하는 데이터들을 전달할 경우가 발생하기 때문에 매크로블록 단위로 저장하기보다는 데이터를 영상의 라인 단위로 저장하는 것이 메모리의 사용 효율을 증가시킬 수 있다. 만약 색차신호를 매크로블럭 단위로 정렬한 후 저장하고, 데이터를 매크로블럭 단위로 처리한다고 가정하면, 움직임 보상에 사용될 데이터가 매크로블럭의 경계를 벗어나게 되는 값들은 DRAM에서 서로 다른 메모리의 열에 대한 어드레스를 갖게 되어, 빈번한 랜덤 억세스 모드를 초래하게된다. 빈번한 랜덤 억세스 모드의 발생은 시스템의 성능을 저하시키며, 데이터들을 처리하기 위한 하드웨어 크기 또한 증가한다.

그림 6(b) 방법으로 DRAM의 데이터를 처리하기 위해 필요한 시스템 클럭 수를 계산하면 8번의 랜덤 억세스 모드와 8번의 패스트 페이지 모드로 32개의 시스템 클럭이 필요하다. 그림 6(c)의 경우에

www.dbpia.co.kr

는 2번의 랜덤 억세스 모드와 14번의 패스트 페이지 모드로 20개의 시스템 클럭으로 데이터를 처리할 수 있다. 그림 6(b)와 (c)의 처리 방법에서 요구되는 시스템 클럭 수가 같다고 가정해도, 그림 6(b) 처리 방법을 제어하기 위한 하드웨어는 그림 6(c) 처리 방법을 제어하기 위한 하드웨어에 비해 상대적으로 크고 복잡한 논리회로들이 필요하다. 따라서, 본 논문에서는 그림 6(c)에 나타낸 방법을 이용하여 데이터를 처리하였다.

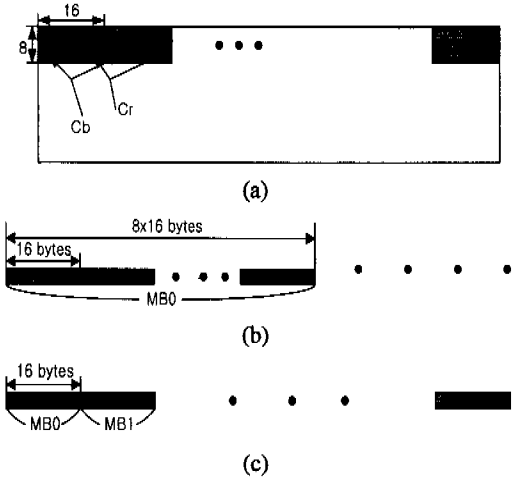


그림 6. coded CbCr에 대한 메모리 맵: (a) CbCr 영상, (b) 매크로블럭 단위의 coded CbCr 처리 방법, (c) 라인 단위의 coded CbCr 처리 방법

3. 4:2:2 입력 영상의 메모리 맵

IP 모듈이 처리하는 입력 영상은 4:4:4 형식의 RGB, 4:2:2 형식의 YUV, CCIR 601 형식의 데이터들이며, 출력은 4:2:0 형식의 YCbCr이다. IP 모듈은 입력 영상에 대해 크기 변환만을 수행하기 때문에 데이터의 처리 단위는 라인 혹은 매크로블럭 단위 모두 가능하다. 그러나 IP 모듈에서 입력 영상을 매크로블럭 단위로 처리한다고 가정하면 매크로블럭 단위로 데이터를 정렬하기 위한 부가적인 메모리가 필요하게 되어 전체적인 시스템의 크기가 증가하기 때문에 영상이 입력되는 순서대로 데이터를 처리하는 것이 바람직하다. 뿐만 아니라, DRAM의 메모리 맵에서 original Y의 경계를 CbCr 영상의 4 라인으로 제안하여, 영상의 한 라인에 해당하는 데이터가 메모리의 새로운 열에 저장되는 것을 방지함으로써 DRAM의 사용 효율을 높일 수 있도록 하였다. Original Y, CbCr 메모리 영역에 대한 데이터 처리 방법은 coded CbCr 메모리 영역에 적용한 방법과 동일하다.

4. 채널 버퍼

채널 버퍼는 VLC 모듈이 출력한 데이터를 처리하기 위한 곳으로 데이터의 처리 방법이 위에서 기술한 방법과 동일하지 않다. 채널 버퍼 영역에 대한 데이터 처리 방법은 VLC 모듈이 발생시킨 가변 길이의 데이터를 채널 버퍼 영역에 순차적으로 저장하고 채널이 요구하는 전송률에 맞추어 저장된 순서대로 출력하는 FIFO (first-in first-out)의 형태이다. 가변 길이의 데이터는 동시다발적으로 급격히 많은 데이터를 발생시키는 특성을 갖고 있는데, 본 논문에서는 많은 양의 데이터가 갑자기 발생하더라도 처리할 수 있게 채널버퍼 영역을 설정하였다.

Ⅳ. VHDL 회로 합성 및 시뮬레이션 결과

본 논문에서 설계된 하드웨어는 0.5 μm VTI, vcmn5a3 셀 라이브러리와 COMPASS 툴을 이용하여 합성하였다. RT (register transfer) 수준의 기능 검증은 Mentor graphics 사의 Quick VHDL을 이용하여 수행하였으며, 게이트 수준의 기능 검증은 COMPASS 사의 qsim을 사용하였다. 회로 합성 및 시뮬레이션 과정은 다음과 같다.

RT 수준의 기능 검증을 위해 고수준의 C 언어를 이용하여 하드웨어를 모델링하고, 모델링된 프로그램의 결과와 VHDL로 기술된 하드웨어의 시뮬레이션 결과를 비교하여 RT 수준 기능 검증을 수행한다. RT 수준의 기능 검증이 완료되면, 기능 검증에 사용된 RT 수준의 기능 검증용 테스트벤치의 입력 값을 그대로 이용하여 게이트 수준의 검증에 필요한 입력으로 가해준다. 입력된 데이터에 따라 출력된 값들은 파일 형태로 저장되고 RT 수준에서 출력된 데이터와 게이트 수준의 검증에서 출력된 데이터를 C 프로그램으로 상호 비교하여 검증을 수행한다. 하드웨어의 기능 검증은 하드웨어에 구성된 네 개의 인터페이스 서브 모듈 별로 수행한 후, 각

www.dbpia.co.kr

하드웨어들을 통합하여, 통합된 하드웨어에 대해서 검증을 수행하는 탑다운 방식으로 수행한다.

하드웨어 기능 검증은 DRAM을 사용하는 타이밍에서 DRAM의 어드레스, DRAM 인터페이스와 관련된 여러 제어신호 (RAS, CAS, OE, WE), 그리고 입출력되는 DRAM의 데이터들이 각 타이밍에서 유효한지를 검증함으로써 전체 하드웨어에 대한 동작을 확인하였다. 그림 7은 비디오 부호화기가 프레임 단위로 영상을 처리하는 경우에 대한 테스트 벤치의 실행 결과로써 각 인터페이스 서브 모듈별로 검증된 결과를 보여준다. 그림 7은 VHDL 시뮬레이터인 Mentor graphics 사의 Quick VHDL 상에서 검증된 결과이며, 입·출력된 신호들인 시스템클럭 SCK2,인터페이스를 위한 여러 인에이블 신호들, DRAM 제어신호인 RAS, CAS, OE, WE, 그리고 DRAM 데이터들을 보여주고 있다.

DRAM 억세스 방법에 따른 DRAM 사용 기간과 처리할 데이터의 양은 하드웨어 내부에 구성되는 입출력 버퍼의 크기를 결정하며, 이러한 내부 버퍼의 크기는 칩의 면적을 결정하는 중요한 인자로 작용한다. 입출력 버퍼의 크기가 커지면 상대적으로 칩의 면적이 커져 칩의 비용이 증가되기 때문에 비용을 감소할 수 있도록 적절한 크기로 동작하는 입출력 버퍼가 설정되어야 한다. 본 논문에서 구현된 하드웨어 내부의 입출력 버퍼들은 SRAM으로 구성되어 있다. 표 1은 제시된 메모리 맵과 인터페이스 방법에 따라 데이터를 처리할 때 필요한 내부 버퍼의 크기를 정리한 것이다. 표에서 살펴볼 때, coder 모듈에서 DRAM으로 전달하는 coded Y와 coded CbCr을 처리하기 위한 입력 버퍼의 크기는 각각 1개의 매크로블럭을 저장할 수 있는 크기인 128 바이트와 256 바이트이며, ME/MC 모듈로 전달할 coded Y와 coded CbCr를 출력하기 위한 버퍼의 크기는 3개의 coded Y 매크로블럭을 저장할 수 있는 크기인 768 바이트이다. 3개의 매크로블럭을 저장하기 위한 내부버퍼의 크기는 프레임 기반 부호화와 필드 기반 부호화 방법 모두에 적용할 수 있도록 되어 있고, DRAM 억세스에 따른 병목 현상을 제거하기 위해 설정된 크기이다. 이 내부 버퍼는 또한 coded CbCr를 처리할 때에도 그대로 이용한다. IP 모듈로부터 들어오는 데이터의 크기는 최대 330 바이트이나, 필터링을 위한 여분의 공간을 위해 384 바이트 크기로 설정한다. Original Y, CbCr를 모두 처리할 수 있도록 768 바이트 크기의 입력 버퍼를 설정하였다. 프레임 메모리에서 IP 모듈로 데이터를 출력하기 위한 출력버퍼는 original Y에 대해 256 바이트 크기를 original CbCr 데이터들을 위해 384 바이트 크기로 설정하였다. VLC 모듈과의 인터페이스를 위해 설정된 버퍼는 최대 발생할 수 있는 가변율의 데이터를 모두 수용할 수 있는 크기인 1448 바이트로 설정하였다. 항등율의 데이터를 출력하기 위해서는 128 바이트 크기로 설정하였다. 이 출력버퍼의 크기는 MPEG-2 MP@ML에서 요구되는 채널 전송률로 데이터를 전송할 수 있는 크기이다.

표 1. 프레임 메모리 하드웨어에 구성된 내부 버퍼의 크기

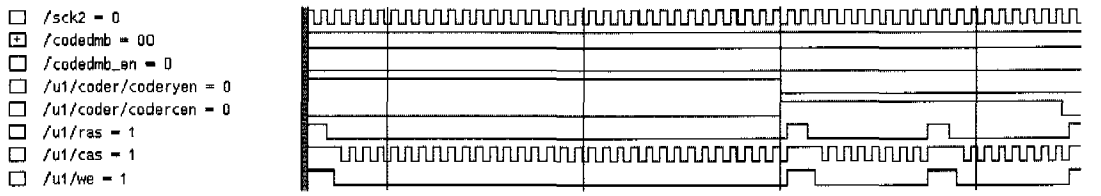
입력	출력	버퍼크기 (bytes)	입력 버스 (bits)	출력 버스 (bits)
IP	DRAM	768	8	64
DRAM	IP	640	64	8
DRAM	ME/MC	768	64	32
coder	DRAM	384	8	64
VLC	DRAM	1448	32	64
DRAM	VLC	128	64	8

V. 결 론

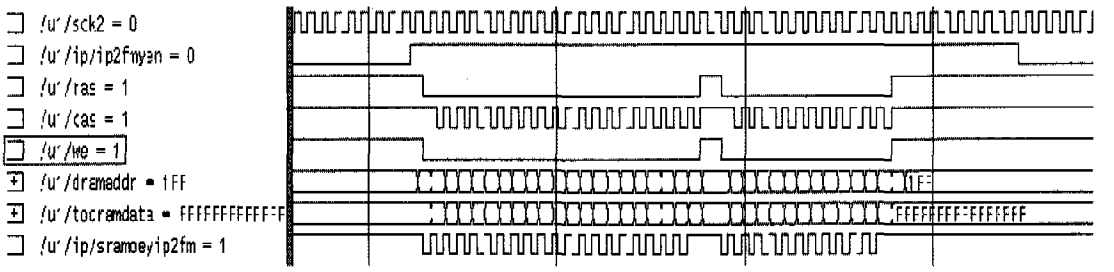
본 논문에서는 부호화기의 성능을 개선할 수 있는 방안으로 효율적인 프레임 메모리 하드웨어의 구조를 제안하였다. 제안된 하드웨어는 효율적인 DRAM의 메모리 맵과 이를 기반으로 설계된 하드웨어 구조를 갖도록 하였으며, 설계된 하드웨어는 제시된 메모리 맵을 기반으로 내부 버퍼와 로직의 크기를 줄일 수 있도록 구현되었다. 설계된 하드웨어는 VHDL을 이용하여 기본 블럭들을 설계한 후, 각각의 기본 블럭들을 이용하여 탑다운 방식으로 구현되었고, 구현된 하드웨어에 대한 기능 검증은 인터페이스를 담당하는 기능별 서브 인터페이스 하드웨어에 대해 단계별 검증을 수행하였다. 또한 개발된 하드웨어에 대한 기능 검증과 추후 성능 개선을 위해서 C 언어로 구현된 하드웨어 에뮬레이터를 개발하였다.

검증을 위한 환경은 하드웨어 검증용 테스트 벡터 및 테스트 벤치와 C 프로그램을 사용하여 Mentor의 Quick VHDL 상에서 RT 수준의 기능을

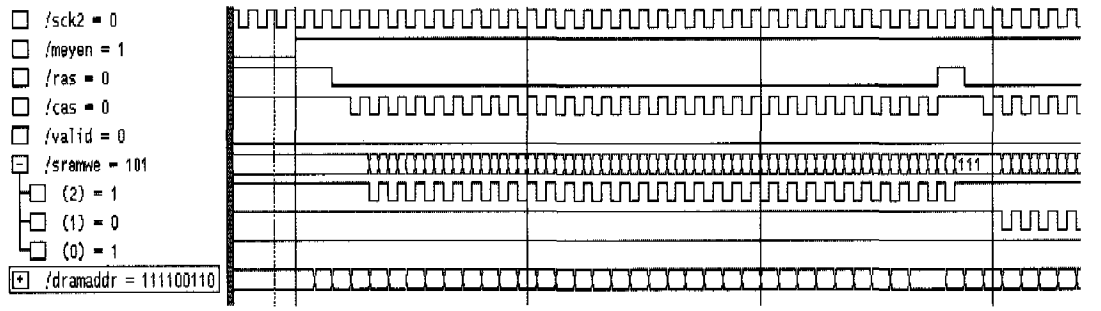
www.dbpia.co.kr



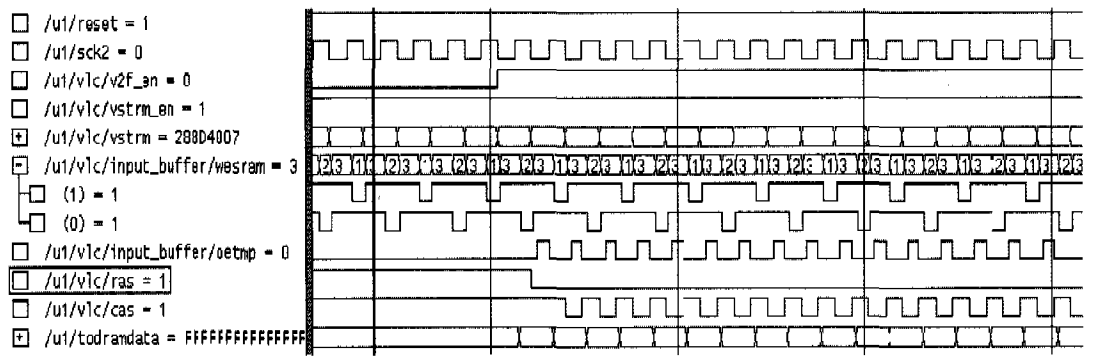
(a)



(b)



(c)



(d)

그림 7. 프레임 기반 부호화에 대한 기능 검증 결과: (a) coder, (b) IP, (c) ME/MC, (d) VLC 인터페이스 하드웨어

www.dbpia.co.kr

검증하였고, 기능 검증이 완료된 하드웨어를 0.5 μ m, VTI의 vcmn5a3을 이용하여 회로 합성 툴인 COMPASS 환경에서 회로를 합성하고, 합성된 회로를 바탕으로 게이트 수준의 시뮬레이션을 수행하여 동작을 확인하였다. 개발된 프레임 메모리 하드웨어는 효율적인 DRAM 사용 시간과 내부 버퍼의 크기로 동작함을 개발된 하드웨어 에뮬레이터를 통해 확인할 수 있었다. 구현된 하드웨어는 MPEG-2의 MP@ML에서 요구하는 전송률로 채널에 데이터를 전송할 수 있도록 하였다.

참 고 문 헌

[1] ITU-T Recommendation H.261, Video codec for audiovisual services at p × 64 kbit/s, 1993.

[2] ITU-T Recommendation H.263, Video coding for low bitrate communication, May. 1996.

[3] ISO-IEC 11172-1: 1993 Information technology, *Coding of Moving Pictures and Associated Audio for digital storage media at up to about 1.5 Mbit/s*, Switzerland, 1993.

[4] ISO-IEC JTC/SC2/WG11, *Generic Coding of Moving Pictures and Associated Audio:ISO/IEC 13818-2 MPEG-2 Video International Standard*, Mar. 1995.

[5] K. R. Rao and J. J. Hwang, *Techniques and Standard for Image, Video, and Audio Coding*, Prentice Hall, New Jersey, 1996.

[6] S. C. Purcell and D. Galbi, "C-Cube MPEG video processor," *SPIE, Image processing and Interchange*, vol. 1659, 1992.

[7] S. Bode, "A single-chip multistandard video codec," *Proc. IEEE Hot Chips V, Stanford*, CA, Aug. 1993.

[8] P. Pirsch, N. Demassieux, and W. Gehrke, "VLSI Architecture for Video Compression," *Proc. IEEE ICASSP*, vol. 83, pp. 220-246, Feb. 1995.

[9] J. Berge, A. Fonkura, and S. Maginot, *VHDL Designer's Reference*, Kluwer Academic Publishers, Dordrecht, 1992.

[10] R. Airiau, J. Berge, and V. Olive, *Circuit Synthesis with VHDL*, Kluwer Academic Publishers, Boston, 1993.

[11] Y. -K. Ko, K. -H. Lee, E. -S. Kang, S. -H. Lee, S. -H. Jang, S. -J. Ko, "The Design of the Frame memory module for MPEG-2 Video Encoder," *Summer Conference of KICS*, vol. 15, no. 1, pp. 452-458, Jul. 1996.

고 영 기(Young-Ki Ko) 정회원

1995년 2월 : 고려대학교 정보공학과(공학사)
1997년 2월 : 고려대학교 전자공학과(공학석사)
1997년 3월~현재 : 고려대학교 전자공학과 박사과정

<주관심 분야> 영상압축, 멀티미디어통신

강 의 성(Eui-Sung Kang) 정회원

1991년 2월 : 고려대학교 전자공학과(공학사)
1995년 2월 : 고려대학교 전자공학과(공학석사)
1999년 8월 : 고려대학교 전자공학과(공학박사)

<주관심 분야> 영상압축, 멀티미디어통신

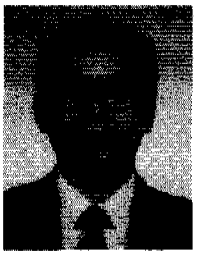
이 경 훈(Kyoung-Hoon Lee) 정회원

1992년 2월 : 고려대학교 전자공학과(공학사)
1994년 8월 : 고려대학교 전자공학과(공학석사)
1998년 8월 : 고려대학교 전자공학과(공학박사)
1998년 12월~현재 : 한국전자통신연구원 선임연구원

<주관심 분야> 비선형 신호처리, 영상처리, 정보보호, 암호학

www.dbpia.co.kr

고 성 제(Sung-Jea Ko) 정회원



1980년 2월 : 고려대학교 전자공학과(공학사)
1986년 5월 : State Univ. of New York at Buffalo, 전기 및 컴퓨터공학과 (공학석사)
1988년 8월 : State Univ. of New York at Buffalo, 전기 및 컴퓨터공학과(공학박사)
1981년 8월~1983년 12월 : 대한전선 중앙연구소 연구원
1988년 8월~1992년 5월: The Univ. of Michigan Dearborn, 전기 및 컴퓨터공학과 조교수
1992년 3월~현재: 고려대학교 전자공학과 교수
1996년 11월: IEEE APCCAS best paper award
1997년 12월: 대한전자공학회 해동논문상 수상
IEEE Senior member, IEE member
<주관심 분야> 신호 및 영상처리, 영상 압축 및 통신, 멀티미디어 통신
E-mail: sjko@dali.korea.ac.kr

www.dbpia.co.kr