

공학석사 학위논문

디지털오디오 저장매체 Interface를  
위한 마이크로컨트롤러 설계 및  
FPGA 구현

2002년 2월

부산대학교 대학원

전자공학과

박종석

공학석사 학위논문

디지털오디오 저장매체 Interface를  
위한 마이크로컨트롤러 설계 및  
FPGA 구현

지도교수 박 주 성

2002년 2월

부산대학교 대학원

전 자 공 학 과

박 종 석

# 박종석의 공학석사 학위논문을 인준함

2001년 12월 일

주심 남 기 곤 (인)

위원 박 주 성 (인)

위원 이 장 명 (인)

# 목 차

I. 서 론 .....	1
II. 디지털 오디오를 위한 저장매체의 개요 .....	4
1. 디지털 오디오를 위한 저장매체의 종류 .....	4
2. 각 저장매체별 특징 및 기본 동작 .....	6
2.1 NAND FLASH memory / SMC(Smart Media Card) .....	6
2.2 MMC(Multi-Media Card) .....	10
2.3 IDE type HDD / CDROM .....	14
III. 디지털오디오 저장매체 interface용 $\mu$ Controller .....	17
1. Digital Audio 저장매체 interface용 $\mu$ Controller .....	17
2. i8302 core 기본 사양 및 특징 .....	18
3. 디지털오디오 interface를 위한 수정된 i8032 core .....	22
3.1 수정된 SFR 영역 .....	22
3.2 Interface용 기능블럭 .....	24
4. 설계될 마이크로컨트롤러를 이용한 MP3-Player system 개요 .....	25
IV. 각 저장매체별 interface 기능 블럭별 설계 .....	26
1. DMA 블럭의 설계 .....	26
2. MP3-stream output control 블럭의 설계 .....	31
3. NAND FLASH memory / Smart Media Card 블럭의 설계 .....	35
4. Multi-Media Card 블럭의 설계 .....	41
5. IDE(Integrated Drive Electronics) interface 블럭의 설계 .....	49

V. 시뮬레이션을 통한 검증 .....	53
1. Synplify / Synopsys tool을 이용한 simulation 환경 .....	53
2. 설계된 Interface용 마이크로프로세서 기능 블록의 simulation .....	54
2.1 DMA_block .....	54
2.2 MP3_CON block .....	56
2.3 FLASH_SMC_CON block .....	58
2.4 MMC_CON block .....	59
2.5 IDE_CON block .....	61
2.6 전체 controller block 검증 .....	61
VI. FPGA를 이용한 실시간 검증 .....	62
1. FPGA compile 과정 .....	62
2. i8032 및 디지털오디오 저장매체 interface FPGA system .....	64
3. 설계검증에 사용할 MP3 decoder board .....	66
4. USB Media-Data Writer board .....	66
5. FPGA 검증 결과 .....	68
6. System On Chip에 대한 생각 .....	69
VII. 결 론 .....	70
참고문헌 .....	71
Abstract .....	72

# I. 서론

반도체 및 초고속 인터넷 통신 기술의 급격한 발전으로 이제는 사회, 문화 전반에 디지털 오디오 데이터가 많이 확산되어 있다. 아날로그 테이프 시대에 CD(compact disk)가 처음 나왔을 때 이상한 물건으로 취급될 때도 있었던 것처럼 아직 MP3나 WMA, AAC등 새로운 압축된 디지털 오디오 포맷(이하 디지털 오디오 데이터)에 대한 친밀감이 덜한 것은 사실이다. 아직은 주변에서 MP3 Player같은 Portable Digital audio player를 사용하는 사람은 적다. 그러나 그 사용 추세는 점점 늘어가며 몇 년 후엔 디지털 오디오 데이터에 대한 친숙함은 점점 더해가서 음악을 듣기 위해 단순히 카세트나 CD player를 사용하는 것처럼 간편하게 MP3 같은 디지털 오디오 데이터를 Internet으로 간편하게 download해서 휴대하기 편한 저장장치에 저장해서 빠르게 저장하여 디지털 오디오를 들을 것이다. 디지털 오디오 데이터는 그 데이터의 보관이나 데이터 전송이 용이하기 때문에 그의 사용은 점점 늘어 갈 것이다. 물론 이러한 장점들이 때론 음악에 있어서 저작권 위반이라는 다른 단점으로 나타나기도 하지만 그 수요는 늘어갈 것은 불을 보듯 뻔하다.

이러한 디지털 오디오 데이터의 저장은 여러 저장매체에서 이루어진다. 비디오 데이터보다는 데이터 양이 적은 관계로 portable type의 조그만 메모리카드가 주로 많이 사용되어 진다. 현재 시중에서 사용되고 있는 Smart-Media Card나 Multi-Media Card가 대표적이며, Memory stick 및 NAND형 반도체 FLASH memory등도 많이 사용된다. 디지털 오디오 데이터의 대용량 저장이 필요한곳에는 하드디스크나 CDROM type의 저장매체도 사용된다. 어떤 압축된 디지털 오디오라 할지라도 저장할 매체가 필요하므로 이러한 저장매체와의 인터페이스를 담당할 전용 마이크로컨트롤러가 필요하다. 물론 이러한 마이크로컨트롤러는 embedded형태로 압축된 디지털 오디오 데이터를 압축 및 복원하

는 SOC type의 반도체에서도 사용될 수도 있다.

이러한 현실에 대응하기 위해 수많은 디지털 오디오 데이터의 포맷이 저장되어 있는 저장매체와의 Interface를 용이하게 하기 위한 인터페이스전용 마이크로컨트롤 설계가 필요하게 되었다. 디지털 오디오데이터의 처리는 비디오에 비해 빠르지 않으며 특히 압축된 MP3나 AAC, WMA등에 대한 처리는 8bit급 마이크로컨트롤러가 적당하다. 설계의 용이성 및 신뢰도를 높이고 개발 기간 단축을 고려하기 위해 기존에 개발된 8bit Micro-controller인 i8032 core를 사용했다. 이 core를 약간의 수정을 통해 기존의 core크기를 줄이고 디지털 오디오 데이터 처리를 위한 기능 블록을 추가 설계하였다. 이렇게 설계된 저장매체 인터페이스용 마이크로컨트롤러 검증을 위해 여러 가지 저장매체에 대응할 수 있는 MP3 player system을 FPGA로 구현하는 게 본 논문의 목적이다. 실제 꼭 디지털 오디오 데이터 인터페이스만을 위해서가 아닌 저장매체와 그 데이터를 사용하는 system간의 interface에 사용하여도 유용할 것이라 믿는다.

실제 설계에 있어서 i8032 core는 firmware적인 control 역할을 맡았으며 특정 기능 블록에 해당하는 control register의 setting만으로 편하게 데이터를 주고 받을 수 있는 구조로 설계하였다. 사실 i8032 core를 수정하여 사용하는 것보다 본 논문에서 논하고자 하는 여러 가지 저장매체의 인터페이스를 이해하고 구현하는데 더 많은 시간이 필요했다. 따라서 해당되는 기능 블록을 편리하게 사용할 수 있는 firmware 코드를 C-source로 작성하여 같이 논문에 기재하여 이 논문을 참고하는 이에게 도움을 주고자 한다.

같이 설계하려고 했던-저장매체에 빠르게 write할수 있는-USB interface에 대한 control 부분은 시간과 관련 지식의 부재로 구현하지 못했다. 대신 시중에 나와있는 USB control chip set을 사용하여 본 논문에서 설계할 controller와 별도로 USB media/read-writer를 제작하였다.

실제 system을 구현하는 부분인 FPGA 구현에 있어서 효율적인 VHDL source coding을 통하여 FPGA compile의 try-and-error를 줄일려고 노력하였

다. 이 과정은 Synplify라는 FPGA compiler 툴을 사용하였는데 이 툴을 통해 warning type의 code나 functional에 상관없는 정확한 don't care처리 등을 통해 VHDL source code의 신뢰성을 높이고자 했다. 그 결과 FPGA compile에 대하여 깊은 지식이 없어도 기본적인 setting의 FPGA compile만으로도 비교적 훌륭한 결과 얻을 수 있었다. FPGA는 Altera사의 FLEX10K250이며 Altera에서 제공하는 Synopsys FPGA Express compiler와 Altera사의 MAX-Plus2 v9.3을 사용하였다. 25만 게이트급을 사용하여 FPGA sell Area의 부족에서 오는 signal의 invalid여부의 검증을 많이 줄일 수 있었다.

II 장에서는 일반적으로 사용되는 디지털 오디오 저장매체의 종류와 그 기능을 살펴보고 그 매체에 대한 인터페이스 컨트롤 블록을 설계할 대략적인 구조를 설명하며, III 장에서는 i8032 core의 기본 사양 및 논문에 사용할 디지털 오디오 저장매체 인터페이스용 마이크로컨트롤러 설계를 위한 수정되고 추가된 부분에 서술이 있을 것이며, IV 장에서는 본 논문에서 구현한 NAND FLASH memory, SMC, MMC, HDD에 대한 interface구현의 기능 블록에 대응된 VHDL code설계, V 장에서는 설계할 컨트롤러의 기능 블록별 VHDL code simulation을 통한 검증 서술, VI 장에서는 Altera FPGA을 이용한 VHDL로 설계된 디지털오디오 저장매체 인터페이스용 마이크로컨트롤러를 MP3-player system에 적용시켜 보며, VII 장에서는 전체 논문의 결론을 서술하는 순서로 논 하겠다.



## II. 디지털 오디오를 위한 저장매체의 개요







### 1. 디지털 오디오를 위한 저장매체의 종류

본 논문에서 언급하는 디지털 데이터 저장매체들은 꼭 디지털 오디오 데이터 저장용으로만 대응되지는 않는다. 오디오뿐만 아니라 비디오 데이터 등 어떠한 디지털 형태의 데이터도 쉽게 저장하고 처리할 수 있다. 특히 요즘은 휴대형 디지털 장치, 디지털 카메라, 캠코더, MP3 player, PDA 등에 많이 사용되고 있다. 예를 들자면 디지털 카메라로 사진을 찍어서 카드타입의 휴대용 저장장치에 저장한 후 디지털 사진기 전용 프린트에 카드를 넣어서 출력하곤 한다. 캠코더로 필요한 부분을 MPEG 형태의 압축된 동영상화면을 메모리카드에 저장해서 PC로 가져와서 편집도 할 수 있다.

본 논문에서 논의하고자 하는 디지털 오디오를 저장하고 읽어낸다는 것은 이러한 디지털 저장매체에서 압축된 디지털 오디오 데이터, MP3, AAC, WMA 등의 디지털 압축 데이터 포맷을 저장하고 읽어내는 것에 기준을 맞추어 서술하고자 한다. 본 논문에서 설계하고자 하는 디지털 저장장치 인터페이스용 마이크로프로세서를 디지털 카메라 등과 같은 비디오 시스템에도 마찬가지로 적용될 수 있을 것이다.

이러한 저장매체의 대부분은 각종 디지털 제품의 데이터(음성, 영상, 문자 등) 저장 및 상호 교환을 위해 개발된 새로운 개념의 외장형 메모리 카드가 주류를 이루고 있으며, 휴대 및 사용이 간편할 뿐 아니라 여러 기기에 다양하게 응용할 수 있는 차세대 저장매체이다.

다음 표2-1는 일반적으로 폭넓게 사용되는 휴대형 타입의 디지털 멀티미디어 데이터 저장용 장치의 종류를 몇 가지 나타내어 보았다.

종 류	특 징 및 사 양
Smart Media CARD 	저장매체 미디어중에서 일보 앞선 존재로 도시바에서 개발함. 권리보호를 위한 ID 부착형도 있으며 두께에서 타 제품보다 앞섬. 문제는 대 용량화로의 발전 가능성으로 현재 삼성, OLYMPUS 등에서 생산중. * 무게 : 2g * 크기 : 45.0 × 37.0 × 0.76mm
MMC card 	콤팩트 플래시의 소형판으로 미국의 샌디스크사가 개발한 우표 크기의 메모리카드. 콤팩트 플래시와 비교해 부피가 5/1, 두께가 약 2/1 (1.4mm) 정도로 현재 MP3 플레이어에 가장 많이 사용된다. * 무게 : 1.5g * 크기 : 32.0 × 24.0 × 1.40mm
Compact Flash Memory 	꽤 등장이 오래된 제품으로 미국의 샌디스크에서 개발하여 디지털 카메라의 보급과 함께 성장한 반면 커다란 크기로 인하여 시대에 뒤처지는 경향. 용량은 꽤 큰 편이다. 몇 백Mbyte 제품도 나와있다.
Memory Stick 	SONY에서 개발하여 같은 회사 제품에 폭넓게 사용. 권리보호를 위한 MG메모리스틱도 있으며 오작동에 의한 지움 방지 기능으로 높은 신뢰성을 확보.
Micro-Driver 	IBM에서 개발한 세계 최소, 최경량의 하드 디스크 드라이버로 용량이 340MB. 드라이버 직경은 500원 주화 크기로 무게가 16g이며 콤팩트 TypeII 라는 전용 PC 카드 어댑터를 사용하여 고속 전송이 가능.
CLICK 	IOMEGA 사가 개발한 소형 자기기록 미디어. 사이즈는 명함크기의 약 2/1정도로 무게 약 10g, 용량은 40MB

[표2-1] 휴대형 디지털 멀티미디어 데이터 저장장치

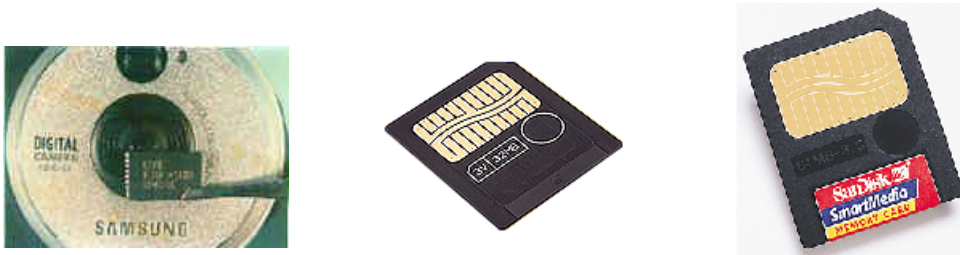
[표2-1]에 나타내어 있는 것들 뿐 아니라 그 외 SD카드, SMMC 카드 등 계속

개발되고 있는 중이다. 기타 카드 타입이 아닌 반도체 NAND형 FLASH memory도 시스템에 on-board 형태로 실장 되어서 시스템 기본 메모리로 많이 사용되고 있다. 그 외 대용량 저장장치로 HDD, CDROM을 들수 있으며 아직까지는 대용량 저장매체로는 위 2가지를 능가하는 제품은 없어 보인다.

하드디스크 및 CDROM은 기본적으로 IDE(Integrated Drive Electronics, ATA) 및 ATAPI interface방식으로 데이터를 주고받으며 위의 휴대형 타입의 저장장치보다 속도가 월등히 빠르고 데이터 저장용량도 우위에 있다.

## 2. 각 저장매체별 특징 및 기본동작

### 2.1 NAND FLASH memory / SMC(Smart Media Card)

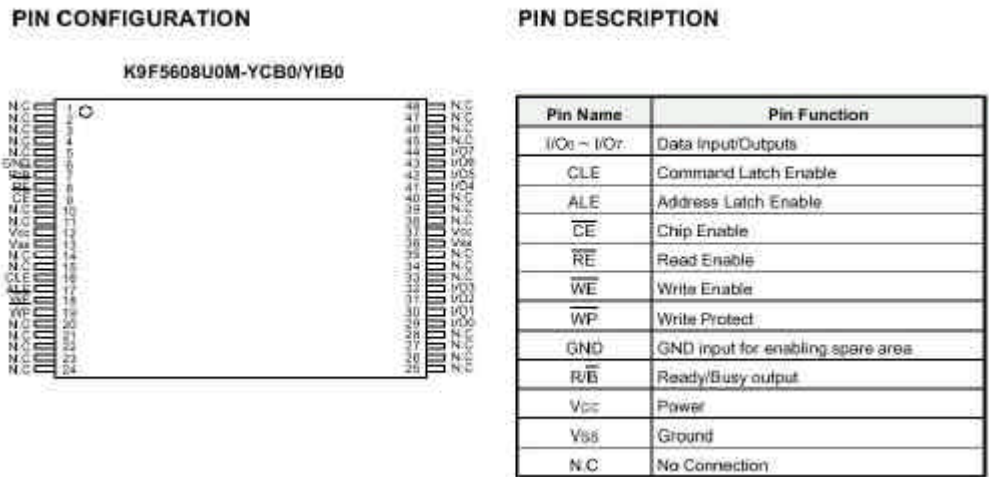


[그림2-1] NAND FLASH memory와 SMC

FLASH 메모리는 소용량의 NOR형과 비교적 대용량의 저장용 NAND형 메모리로 구분된다. 특히 NAND형 플래시 메모리는 MP3플레이어, 인터넷 휴대폰, 디지털 카메라 등 최신 디지털 전자제품에서 그 수요가 증가하고 있다. 반면 NOR형은 반응속도가 매우 빠르기 때문에 embedded system의 ROM용으로 주로 사용되고 있으며 NAND형에 비해 고가이다. 이러한 FLASH type의 메모리들은 48PIN TSOP I type의 반도체 메모리와 카드형태의 스마트미디어카드로

서 많이 사용되고 있다

- NAND FLASH memory



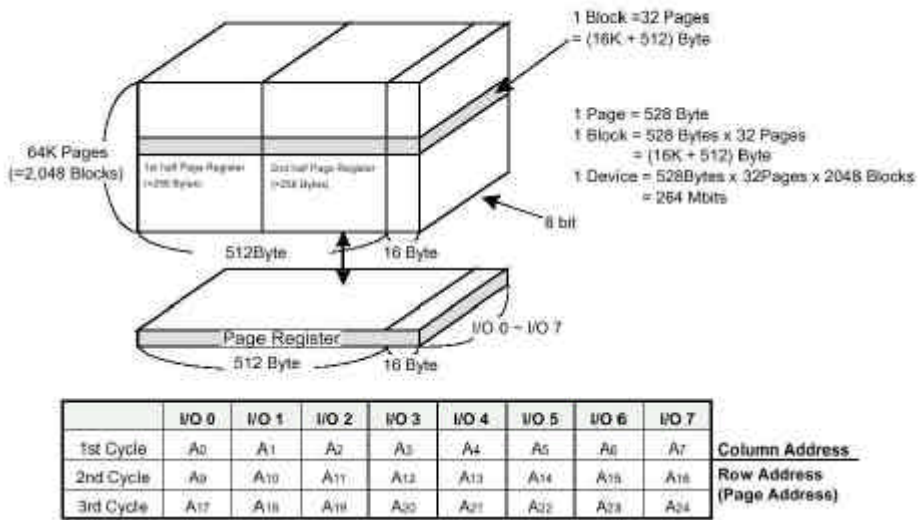
[그림2-2] NAND FLASH memory PIN

위에 나타난 NAND형 FLASH memory는 256Mbit, 즉 32Mbytes 용량을 사용하였으며 필요에 따라 2,3 개를 같이 버스에 물려 사용할 수도 있다.

최소 데이터 R/W 단위는 page 이며, 1 page는 528bytes의 크기를 갖는다. 1 page는 512bytes의 데이터 영역 이외에 별도로 16bytes의 Spare영역을 가지고 있다. Spare영역의 용도는 사용자 임의로 정할 수 있으며, 대개의 경우 데이터 신뢰성을 높이기 위해서 데이터 영역의 에러 감지 및 교정을 위한 ECC 등의 에러 교정 코드를 저장한다. FLASH 메모리의 데이터 삭제 단위는 block이며, 1 block은 32 page이다. 그리고 전체 FLASH 메모리는 1024 block을 가지고 있다. 따라서 FLASH 메모리 전체 용량은 다음과 같이 계산할 수 있다.

$$512\text{bytes(데이터 영역)} * 32\text{page} * 2048\text{block} = 264\text{Mbits}$$

Figure 2. ARRAY ORGANIZATION



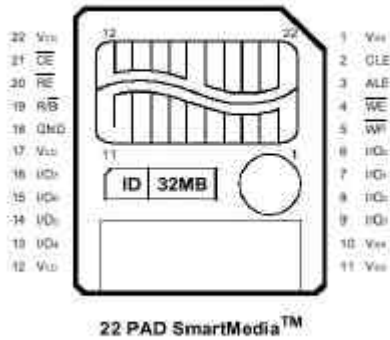
[그림2-3] NAND FLASH memory array 구조

플래쉬 메모리는 물리적인 특징으로 인하여 생산 시부터 손상된 block(Invalid Block)을 가지고 있으며, 또한 사용 중에도 잘못된 동작 등의 이유로 인하여 손상된 block이 발생할 수 있다. 메모리 생산 시에 발생한 Invalid block은 해당 block의 0, 1 page의 데이터가 다 0x00으로 채워져 다른 block과 구별된다. 그러나, 0번 block은 칩 메이커 측에서 Valid block으로 보장하고 있다. Invalid block은 FLASH 메모리 사용 중에도 발생할 수 있다. 현재 칩 메이커에서 보장하는 최대 Invalid block의 개수는 20개이며, 이런 Invalid block들은 정상적인 동작을 보장하지 않으며, 향후 사용을 막아야한다. 그러기 위해서는 S/W적으로 block 관리 알고리즘이 필요하다.

그 외 자세한 내용은 나중에 뒤에 설계할 마이크로프로세서 기능블럭의 설명 때 자세히 하도록 하겠다.

○ SMC ( Smart Media Card)

SmartMedia™ CARD(SSFDC)



PIN DESCRIPTION

Pin Name	Pin Function
I/O0 - I/O7	Data Input/Outputs
CLE	Command Latch Enable
ALE	Address Latch Enable
CE	Chip Enable
RE	Read Enable
WE	Write Enable
WP	Write Protect
GND	Ground
R/B	Ready/Busy output
Vcc	Power
Vss	Ground
N.C	No Connection

[그림2-4] Smart Media Card PIN

도시바 등에 의해 개발된 스마트 미디어는 애초에 SSFDC(Solid State Floppy Disk Card)로 불렸으나 1996년부터 스마트 미디어로 호칭이 통일됐다. 경우에 따라 SMC (Smart Media Card)라고도 부른다. 스마트 미디어는 다른 메모리에 비해 얇고(0.76mm) 가벼워(2g) 휴대기기용으로 뛰어난 특징을 가지고 있다.

이는 컨트롤러를 탑재하지 않고 플래시 메모리 반도체 칩을 카드 내에 삽입한 구조이기 때문이다. 그러나 칩 내장이라는 구조 때문에 대용량 미디어를 사용할 수 없다는 단점이 있다. 컨트롤러를 내장하지 않아 미디어 가격이 저렴한 편이다. 워낙 얇기 때문에 보기도 파손의 위험성이 따르고, 전극이 외부에 노출되어 있어 정전기 등에 의해 데이터가 지워지는 사례가 보고되고 있지만 우려할 만한 수준은 아니다. compact하기 때문에 휴대용 제품에 널리 사용되고 있으며, 컴퓨터 사용자들이 많이 사용하는 디지털 카메라와 MP3 플레이어에 모두 채택되어 범용성 또한 큰 편이다. 현재로는 주로 유통되는 것이 32,64M이

며 최대 용량은 256MB가 나와있다. 반도체 FLASH memory와 기본동작은 동일하며 package모양만 다르다. FLASH 메모리의 기본적인 동작 시퀀스를 간략화하면 다음과 같다.

-Read : Read Command => Address => Status Check => Read Data

-Write : Write Command => Address => Write Data => Write\_Start Command

=> Status Check (success ?)

-Erase : Erase Command => Address => Erase\_Start Command

=> Status Check (success ?)

그 외 자세한 내용은 나중에 뒤에 설계할 마이크로프로세서 기능 블록의 설명 때 자세히 하도록 하겠다.

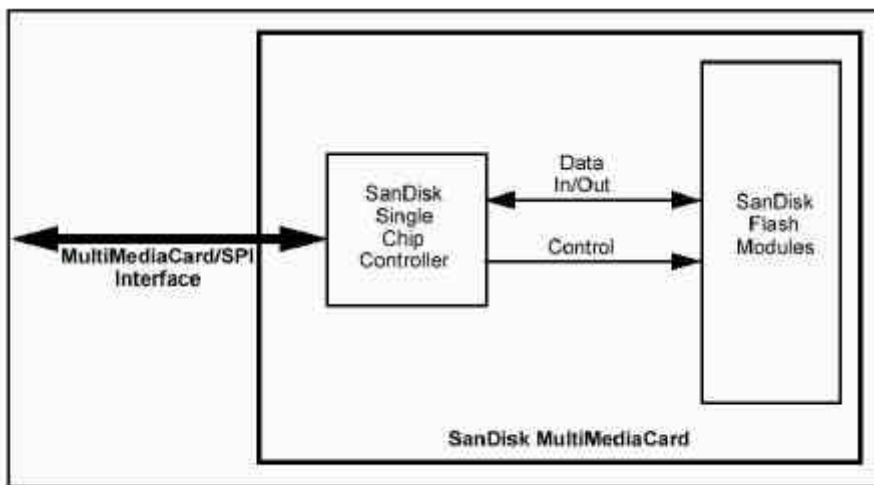
## 2.2 MMC - MultiMedia Card



[그림2-5] MMC 카드 / MMC 카드를 사용하는 JVC의 비디오 카메라

크기  $32 \times 24 \times 1.44\text{mm}$  로 플래시 메모리 규격의 최소의 크기를 가진다. 작은 크기로 인해 초소형 휴대 기기에 사용되지만 작은 용량이 단점이다

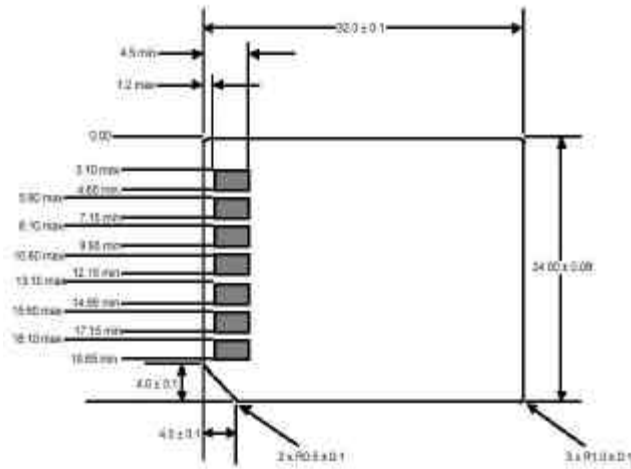
미국의 샌디스크(SanDisk)와 독일의 지멘스(Siemens)가 공동으로 개발한 제품으로 두께는 1.44mm, 크기는 CF의 1/4 배로 매우 작다. MMC는 크기가 작아 소형 기기를 만드는 데에 유리하고 내부에 컨트롤러가 내장되어 제어하기가 쉬운 반면, 시리얼 데이터 버스를 사용하므로 전송 속도가 스마트 미디어에 비해 현저하게 느리다. 그러나 하나의 시리얼 버스를 통해 최대 30개의 카드가 연결될 수 있다. 용량을 늘리기가 쉽다. 메모리 카드로는 처음으로 플래시 메모리와 데이터 변환용 컨트롤러를 하나의 칩에 집약해 두 가지 기능을 일체화했기 때문에 정보처리 속도를 향상시킬 수 있다. 현재까지 개발된 최대 용량은 128MB이다. Compact FLASH나 SMART-media에 비해 뒤늦게 개발됐기 때문에 지원하는 제품이 폭넓지 않다. 하지만 로열티가 싼 편이고 크기 면에서 워낙 유리하기 때문에 많은 업체에서 지원할 의사를 밝히고 있어 전망이 밝다. 국내에서는 60g 이하의 초소형 MP3 플레이어와 MP3 겸용 휴대폰 등에서 주로 사용된다



[그림2-6] MMC 카드의 Block diagram



○ MMC의 기본동작 Protocol



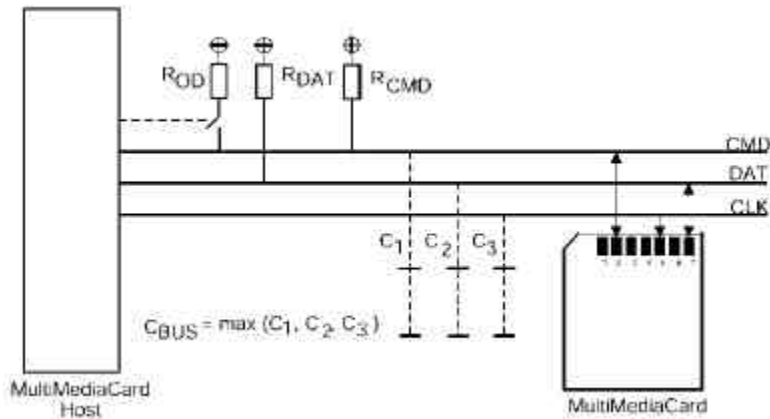
[그림2-7] MMC 카드의 Dimension

MMC card는 2가지의 communication protocol을 가지고 있다. 하나는 Multi-media 모드이고 하나는 SPI 모드이다. 두 모드의 차이점은 입출력 핀의 사용으로 구분지어 질 수 있다. 아래의 [표 2-2]와 같이 두 모드 간에 사용되는 핀은

다르게 주어진다.

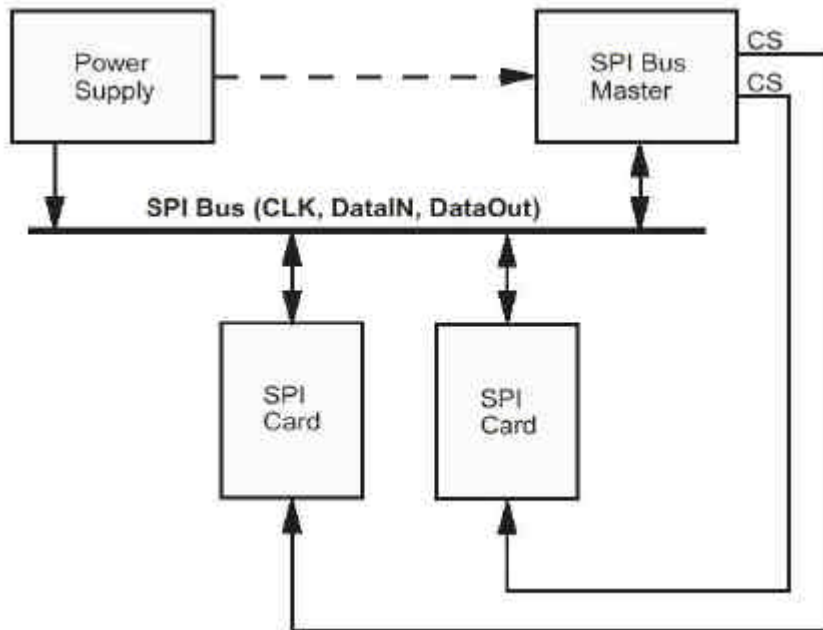
PIN	Multimedia mode		SPI mode	
	Name	Type	Name	Type
#1	RSV	N.C.	CS	Input
#2	CMD	I/O,PP,OD	DataIn	Input
#3	VSS1	power	VSS1	power
#4	VDD	power	VDD	power
#5	CLK	Input	CLK	Input
#6	VSS2	power	VSS2	power
#7	DAT0	I/O/PP	DataOut	Output

[표2-2] 2가지 모드에 따른 MMC pin



[그림2-8] MMC 카드의 Multimedia BUS diagram

2가지 모드의 가장 큰 차이점은 SPI모드가 Multimedia mode에 비해 단순하게 구성되어 있다. Multimedia모드의 bi-directional pin인 CMD, DAT가 data input, data output으로 고정되어 있어서 데이터의 reading과 writing이 이루어질 때 sequential한 multiple block의 데이터 read/write가 불가능하게 되어 있다. 따라서 SPI mode에선 단지 한 블록 단위의 read/write가 이루어진다. 반면에 프로토콜 구조가 단순하여 구현하기가 쉽다. 반대로 Multimedia mode에서는 SPI 프로토콜에서 가지지 못하는 다양한 블록 read/write와 여러 추가적인 기능이 포함되어 있다. 또한 Card Insertion/removal기능을 제공하여 SPI mode와 구분 지어진다. MMC 카드 자체는 Card Insertion/removal에 따라 Power protection 기능이 내장되어 있다. 참고로 데이터 input은 clock의 rising에 데이터의 output은 clock의 falling에 valid 하다. 자세한 MMC의 spec은 sandisk사 ([www.sandisk.com/tech/product\\_man-main.asp](http://www.sandisk.com/tech/product_man-main.asp)) 데이터시트를 참고하자. 본 논문에서는 설계구현의 편의성을 고려하여 기본적인 SPI mode로 MMC를 동작시키고자 한다. 다음 [그림 2-9]는 SPI BUS의 블록다이어그램을 나타내어 보았다.



[그림2-9] MMC 카드의 SPI BUS system

그 외 자세한 내용은 나중에 뒤에 설계할 마이크로프로세서 기능블럭의 설명때 자세히 하도록 하겠다.

### 2.3 IDE - HDD / ATAPI - CDROM

예전부터 대용량 저장매체로 사용되어져 온 HDD와 CDROM은 각각 IDE(ATA)나 SCSI 및 ATAPI(ATA packet interface) 방식으로 데이터를 주고 받는다. 일반적으로 IDE 규격을 ATA라고도 부르기도 하는데 PC/AT 호환기에 사용되고 있는 하드디스크 인터페이스의 하나인 IDE(Integrated Device Electronics)를 미국 표준 협회(ANSI)가 규격화해서 ATA(AT attachment)라

한 것이다. 맨 처음 제정된 ATA 규격은 528MB까지의 용량을 규정한 IDE를 대상으로 하고 있으나 그 후 528MB를 초과하는 E-IDE(Enhanced-IDE)에 대해서는 ATA-2로 규격화하였다. 유사 명칭으로는 ATAPI(AT attachment packaged interface)가 있는데 CD-ROM 드라이브나 테이프 백업 장치 등을 IDE 인터페이스에 접속하기 위한 규격이다. 내장형의 CD-ROM이나 테이프 백업 장치의 대부분은 ATAPI 규격에 적합하게 해 두고, 하드디스크와 같은 열의 동일 인터페이스에 접속할 수 있다. ATAPI 기기를 이용하려면 ATA-2 이상의 인터페이스를 필요로 하나 보통 모기판에 실장되어 있다. 울트라 ATA는 미국 퀀텀(Quantum) 등이 규정한 IDE의 확장 규격이며, E-IDE(Enhanced IDE)와 마찬가지로 ANSI에서 규격화된 ATA2의 marketing 규격이다. ATA에 비해 최대 데이터 전송 속도가 33Mbps로 향상된 것이 가장 큰 특징이다.

본 논문에서 구현하고자 하는 것은 ATA-1 spec인 일반 IDE interface를 구현하며 DMA모드가 아닌 PIO mode로 동작할 것을 원칙으로 하겠다. ATAPI를 구현함에 있어서도 기본 IDE입출력을 원칙으로 하므로 아래와 같은 pin 입출력 정보를 가지고 컨트롤러 설계에 적용했다.

#PIN	Signal Name		#PIN	Signal Name	
1	/RESET	Driver H/W reset	2	GND	Power Ground
3	D7	inout data7	4	D8	inout data8
5	D6	inout data6	6	D9	inout data9
7	D5	inout data5	8	D10	inout data10
9	D4	inout data4	10	D11	inout data11
11	D3	inout data3	12	D12	inout data12
13	D2	inout data2	14	D13	inout data13
15	D1	inout data1	16	D14	inout data14
17	D0	inout data0	18	D15	inout data15

#PIN	Signal Name		#PIN	Signal Name	
19	GND	Power Ground	20	KEY	N.C.
21	Reserved		22	GND	Power Ground
23	/IOW	data write strob	24	GND	Power Ground
25	/IOR	data read strob	26	GND	Power Ground
27	/IORDY	r/w access flag	28	SPSYNC/AL E	**
29	Reserved		30	GND	
31	INTRQ	Int request	32	/IOCS	16bit data CS
33	A0	reg addr 0	34	/PIAG	**
35	A1	reg addr 1	36	A2	reg addr 2
37	/CS1FX	reg select CS	38	/CS3FX	reg select CS
39	/DASP	**	40	GND	Power Ground

[표2-3] 표준 IDE(ATA) 규격의 interface pin ( \*\* : ATA 규격 참조)

ADDR	nCS1FX	nCS3FX	A2	A1	A0	READ(nIOR)	Write(nIOW)	
	0	0	x	x	x	illegal		control block register
	1	1	x	x	x	Z	not used	
3FX	1	0	0	x	x	Z	not used	
3FX	1	0	1	0	x	Z	not used	
3F6	1	0	1	1	0	Altern status	Device Cntr	
3F7	1	0	1	1	1	Drive addr	not used	
1F0	0	1	0	0	0	Data port		command block register
1F1	0	1	0	0	1	Error reg	Precomp	
1F2	0	1	0	1	0	Sector count		
1F3	0	1	0	1	1	Sector Number		
1F4	0	1	1	0	0	Cylinder Low		
1F5	0	1	1	0	1	Cylinder High		
1F6	0	1	1	1	0	Drive/Head		
1F7	0	1	1	1	1	Status	Command	

[표2-4] IDE interface I/O function table