
실시간 데이터 저장을 위한 SD 메모리 카드 설계

문지훈*

*이노에이직

Design of SD Memory Card for Read-Time Data Storing

Ji-hoon Moon*

INNOASIC

E-mail : p3161@naver.com

요 약

휴대용 디지털 기기 보급의 확산되면서 휴대용 저장장치 수요가 급증하고 있으며, 디지털 카메라 및 캠코더에서는 대부분 SD 메모리 카드를 이용하고 있다. SD 메모리 카드는 일반적으로 플래시 메모리를 기반으로 사용자 데이터를 저장한 후 PC에 데이터를 복사하는 형태로 사용되고 있다. 본 논문에서는 플래시 메모리에 데이터를 저장하는 방식이 아닌, 네트워크를 통하여 사진 및 영상 데이터 저장을 할 수 있는 SD 메모리 카드를 제안한다. SD Slave IP를 통해서 들어오는 데이터 및 메모리 주소 값들을 플래시 메모리로 보내지 않고 네트워크 서버에 전달하여, 실시간으로 SD 메모리에 저장할 데이터를 안전하고 편리하게 저장할 수 있다.

ABSTRACT

As mobile digital devices have come into more widespread use, the demand for mobile storage devices have been increasing rapidly and most of digital cameras and camcorders are using SD memory cards. The SD memory card are generally employing a form of copying data into a personal computer after storing user data based on flash memory. The current paper proposes the SD memory card of being capable of storing photograph and image data through network rather than using a method of storing data in flash memory. By delivering data and memory address values obtained through SD Slave IP to network server without sending them to flash memory, one can store data necessary to be stored in a computer's SD memory in real time in a safe and convenient way.

키워드

SD memoy card, SD Slave IP , Real-Time, flash memory

1. 서 론

휴대 전자 기기의 사용이 늘어나면서 사용자 데이터를 저장하기 위해서 SD 메모리 카드 사용이 늘어나고 있다. SD 메모리 카드는 최근 들어 대용량 메모리를 지원하며, 빠른 속도로 데이터 처리가 가능하게 되었다.

본 논문은 SD 메모리 카드에 연결된 호스트에서 데이터 저장을 요청할 경우 이를 플래시 메모리에 저장하는 방식이 아닌 실시간으로 네트워크 서버를 통하여 데이터를 저장하는 방식을 제안한

다. 일반적인 SD 메모리 슬레이브 컨트롤러는 SD 호스트에서 요구하는 데이터를 전달하여 호스트와 통신을 수행한다. 일반적인 메모리 카드의 경우 호스트에서 요청한 정보를 플래시 메모리에 읽기 및 쓰기를 수행하게 된다. 본 논문에서 제안된 실시간 데이터 저장을 위해서 SD 슬레이브 IP에서 데이터 읽기 쓰기를 호스트에서 요청할 경우, 관련 정보를 네트워크 서버를 통하여 읽기 및 쓰기를 수행하여 호스트와의 통신을 수행하게 된다. 위와 같은 방법을 이용할 경우 메모리 용량의 제한으로 대용량의 메모리 카드로 교환하는 문제

점 및 카드 분실 시 사용자의 중요한 데이터 분실의 위험을 방지할 수 있다. 실시간으로 데이터 처리가 필요한 분야에서 효율적으로 이용할 수 있다.

제안된 SD 메모리 카드는 임베디드 리눅스 환경에서 실제 구현 되었으며, 400MHz S3C2450 CPU와 128 Mobile DDR, 128MB NAND 플래시 메모리 등이 장착된 임베디드 리눅스 장비와 SD Slave IP 사용을 위하여 XILINX VERTEX-4 XCAVLX60 FPGA 모듈을 이용 하였다. FPGA 모듈과 S3C2450 CPU와의 데이터 송수신을 위하여 SROM 인터페이스를 이용 하였다.

본 논문의 구성은 다음과 같다. 2장에서 관련 연구 및 연구 목적을 설명하며, 3장에서는 실시간 데이터 저장을 위한 SD 메모리 카드에 대한 설계를 설명하며, 4장에서 실험 및 결론을 보인다.

II. 관련 연구 및 연구 목적

본 장에서는 실시간 데이터 저장을 위한 SD 메모리 카드 관련 연구로 SD 메모리 카드 구조에 대해 살펴본 후, 연구 목적에 대해서 설명한다.

2.1 SD 메모리 카드 구조

SD 메모리 카드 구조는 크게 SD 호스트 인터페이스, 카드 인터페이스 컨트롤러, 메모리 코어 인터페이스로 나누어진다. 그림 1은 SD 메모리 구조를 나타낸다.

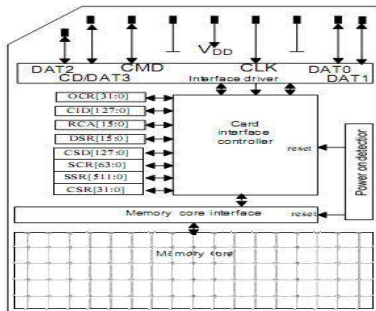


그림 1 SD 메모리 카드 구조 [1]

SD 호스트 인터페이스 부분은 SD 호스트와 연결되는 부분으로, SD 프로토콜 통신을 이용하여 호스트와 통신을 수행한다. SD 메모리 카드는 데이터 읽기/쓰기를 할 경우 단일 블록과 다중 블록을 지원하며, 다중 블록인 경우 4개의 데이터 라인을, 싱글인 경우는 하나의 데이터 라인을 이용하여 데이터를 전송한다[1]. CMD를 호스트에서 SD 메모리 카드로 전송하여 필요한 데이터를 받

아 처리하게 된다.

카드 인터페이스 컨트롤러 부분은 SD 호스트 인터페이스 부분에서 전달 받은 프로토콜을 분석하여, 메모리 코어 인터페이스에 데이터를 저장하거나 읽어 SD 호스트 인터페이스에 저장하는 역할을 수행한다. SD 메모리 레지스터는 카드 식별자, 제조사, 메모리 코어 크기, SD Status, Card Status 등의 메모리 카드 제어 정보 읽기 및 설정할 수 있는 부분이다[1]. 메모리 컨트롤러 레지스터를 통하여 현재 SD 호스트에서 Write시 사용되는 메모리 주소, 크기 등의 정보들이 메모리 코어에 전달되는 중요한 부분이다.

메모리 코어 인터페이스는 메모리 코어 인터페이스를 통하여 전달되는 정보를 이용하여 메모리 코어의 특정 데이터를 전달하거나 저장하게 된다. 일반적인 SD 메모리 코어는 NAND 플래시 메모리가 대부분 사용되지만, 다른 메모리 또한 사용될 수 있다. 근래에는 SSD 메모리 카드로 NAND 플래시 메모리가 아닌 DRAM을 이용하여 구현된 제품이 출시되고 있다[3].

2.2 연구 목적

본 논문에서는 SD 메모리 코어로 플래시 메모리 타입이 아닌 DDR 메모리의 일부 영역을 이용하며, 호스트에서 요구하는 정보를 네트워크 서버를 통하여 데이터를 처리하는 구조로 되어 있다. 이와 같은 방법을 이용하여 구현할 경우의 장점은 아래와 같다.

첫째 NAND 플래시 컨트롤러 IP가 필요 없다. 컨트롤러가 고가이며, 특성상 최신 NAND 디바이스를 이용할 경우 NAND 플래시 컨트롤러에서 해당 메모리 사이즈를 지원하지 않은 경우라면 사용이 불가능한 문제점을 가지고 있다. 둘째 대용량의 데이터를 SD 메모리에 저장할 경우 공간 부족으로 인한 현상으로 인하여 저장할 수 없는 경우가 발생한다. 예를 들어 디지털 캠코더를 통하여 대용량의 동영상을 SD 메모리 카드 저장시 Class를 지원하지 않거나, 용량 부족 현상으로 촬영을 정지해야 하는 경우가 발생할 수 있다. 메모리 코어 인터페이스에서 요청하는 데이터를 네트워크 서버로 전송하여 이러한 문제점을 해결할 수 있다. 셋째 메모리 카드의 분실의 문제가 없다. SD 메모리 카드를 분실할 경우 중요한 데이터가 유출되는 문제가 발생하지만, 실시간으로 데이터를 네트워크 서버에 전송할 경우 이러한 문제점을 해결할 수 있는 장점을 가진다.

III. SD 메모리 카드 설계

3.1 IA650 SD 메모리 컨트롤러 IP

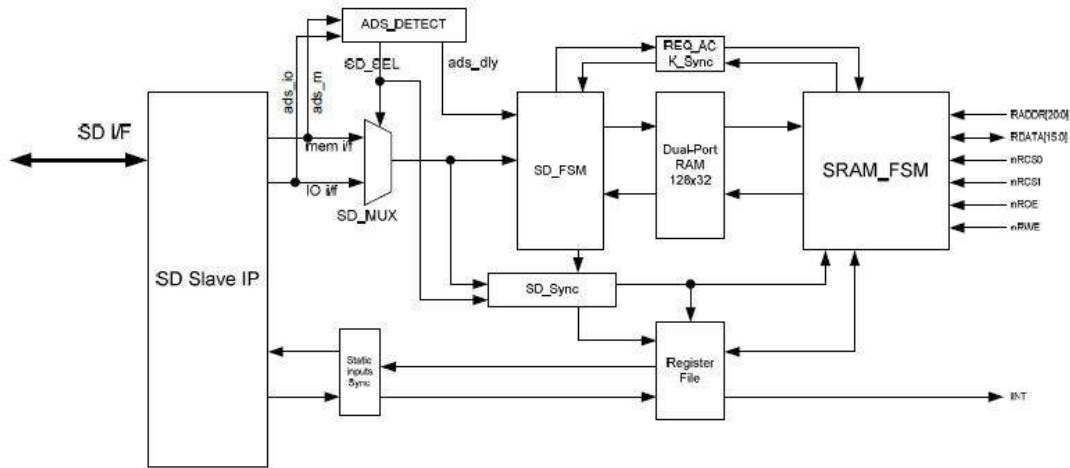


그림 2 650 SD Slave Top 블록 다이어그램

그림 2는 실시간 데이터 저장을 위한 SD 메모리 카드 Top 블록 다이어그램을 나타낸다. SD Slave IP 블록은 SD 인터페이스를 통하여 SD 호스트와 데이터 송수신을 담당한다. 호스트에서 들어오는 프로토콜을 분석하여 해당 블록에 전달하는 역할을 한다.

Register File 블록은 SD Slave IP가 동작하기 위한 기본 설정을 한다. 초기 값은 대부분 SD Slave IP에서 CMD/Response를 수행하기 위한 값이며, SD 호스트에 SD Slave IP가 Ready 되었음을 알리는 값을 설정하여야만 정상적인 동작이 진행된다.

SD 호스트에서 데이터를 쓰려는 동작이 감지되면 SD_FSM에서 SD Slave IP로부터 데이터를 받아 Dual-Port RAM에 512Byte를 쓴다. S3C2450에 내부 RAM에 데이터가 쌓여 있음을 알리는 INT를 '1'로 설정된다. S3C2450은 Register File 블록에서 필요한 레지스터를 읽어 데이터를 쓸 주소와 크기 정보를 알아낸다. S3C2450에서 모든 데이터 처리가 끝났음을 알리는 ARM_FINISH를 Register File 블록에 설정하면 Write Operation이 끝난다.

SD 호스트에서 SD 메모리의 데이터를 요청할 경우의 처리 방법은 다음과 같다. SD Slave IP에서 데이터 읽기 요청이 발생하며, CPU에 인터럽트를 설정한다. CPU는 Register File 블록에서 필요한 값을 참조한다. CPU에서 요구하는 데이터를 모두 SRAM_FSM 블록에 전달한 후 Register File 블록에 ARM_FINISH를 설정한다.

3.2 SD 메모리 카드 설계

기존의 SD 메모리 카드는 메모리 용량 한계로 인하여 디지털 카메라 및 캠코더 데이터 저장 시

문제점을 가지고 있다. 또한 디지털 장비 및 메모리 카드를 분실한 경우 중요한 사용자 데이터 정보까지 잃게 된다. 이러한 문제점을 해결하기 위해서 SDIO 기능을 가진 메모리 카드가 존재하지만 고가격등의 문제로 인하여 널리 사용되고 있지 않는 실정이다. 이러한 문제점을 해결하기 위해서 본 논문에서는 SD 메모리 카드에 데이터를 저장할 경우 메모리 카드에 데이터를 저장하는 방식이 아닌 네트워크 서버를 통해 실시간적으로 데이터를 저장하는 방식을 제안한다. 기존의 메모리 카드의 경우 메모리 카드에 플래시 메모리를 이용하지만, 논문에서 사용된 방식은 SD 메모리 카드를 포맷한 경우는 DDR 메모리를 이용하며, 사용자 데이터는 네트워크 서버에 저장하는 방식이다. 그림 3은 SD 메모리 시스템 블록도를 나타낸다.

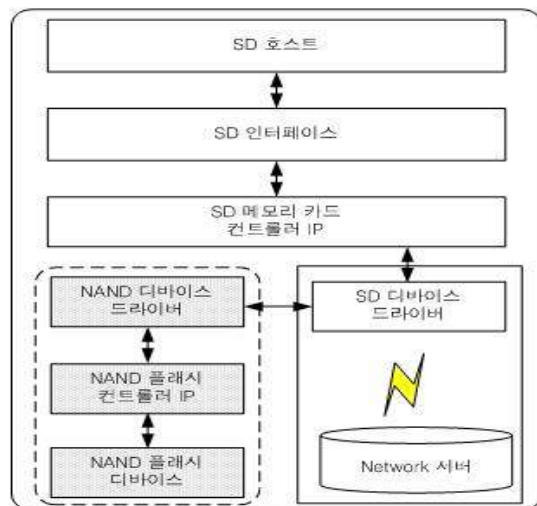


그림 3 SD 메모리 시스템 블록도

그림 3에서와 같이 SD 데이터 저장 장치를 네트워크 서버를 이용할 경우 NAND 플래시 관련 부분이 사용되지 않음을 알 수 있다. SD 디바이스 드라이버는 네트워크 서버의 특정 디스크 영역에 데이터를 저장하는 방식으로 SD 호스트에서 요청하는 데이터를 서버에 저장하거나 읽기를 수행한다. SD 디바이스 드라이버에 의해서 SD 메모리 컨트롤러 IP에서 디스크 I/O가 발생한 경우 요청되는 operation mode, 메모리 위치 및 사이즈 정보를 SD 메모리 카드 컨트롤러 IP의 레지스터에서 읽어 들인다. operation mode가 read인지 write인지를 판단하여, read인 경우라면 네트워크 서버로부터 특정 데이터를 읽어 컨트롤러에 전달하도록 하며, 데이터 write인 경우는 컨트롤러에서 전달해주는 데이터를 읽어 네트워크 서버에 전달한다.

이와 같이 시스템을 구성할 경우 디지털 장비에서 저장장치로 SD 메모리를 사용하는 경우 사용자가 메모리의 데이터를 다른 기억장치에 저장하는 불편, 메모리 분실 문제점 및 데이터를 네트워크 서버에 실시간으로 저장하게 되므로 신속하게 데이터가 저장되어야 하는 분야에서 효율적으로 사용될 수 있다.

SD 메모리 컨트롤러 IP의 CSD 레지스터를 변경하여 소프트웨어적인 방법으로 대용량 메모리 용량으로 변경할 수 있다.

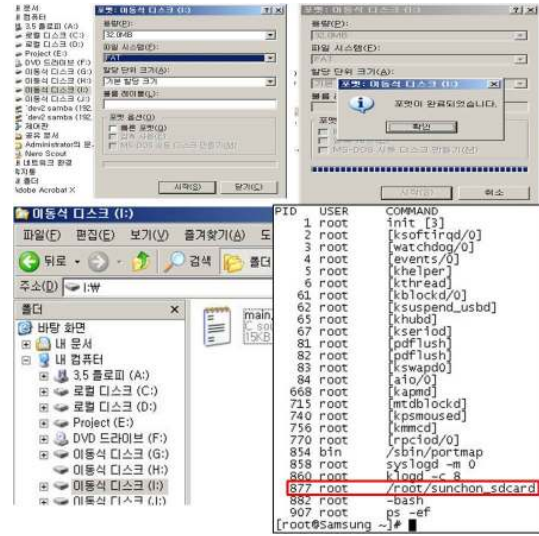


그림 5 SD 시스템 테스트 결과

IV. 실험 및 결론

실시간 데이터 저장을 위한 SD 메모리 카드 시스템은 그림 4와 같다.



그림 4 SD 메모리 카드 시스템

호스트에 SD 카드 인터페이스가 연결되어 있다. 인터페이스를 통하여 CMD,CLK,DATA 라인의 값이 FPGA의 IA650 SD 컨트롤러 IP를 통하여 memory core와 데이터 읽기,쓰기가 필요한 경우 S3C2450의 SRAM 인터페이스를 통하여 SD 디바이스 드라이버와 통신을 수행하여 Network 서버에 데이터 읽기 및 쓰기 동작을 수행하게 된다[2]. 그림 5는 SD 테스트 결과를 나타낸다. 실험의 편의성을 위해서 메모리 용량은 32MB로 하였으며, SD 디바이스 드라이버를 통하여 IA650

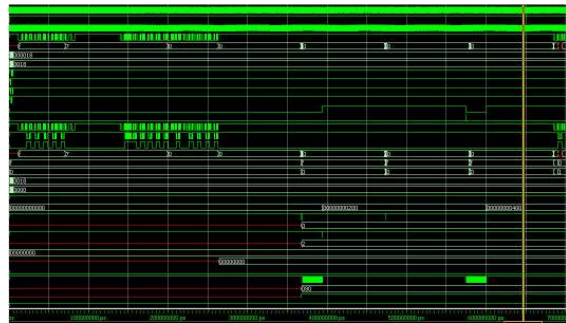


그림 6. SD Slave IP Wave Form

그림 6은 SD Slave IP Wave Form을 나타낸다. SD 메모리 카드로 사용자가 데이터를 복사하는 경우의 Wave Form이다.

결론으로 기존 플래시 메모리 방식이 아닌 Network 서버에 SD 메모리 데이터를 저장하는 방식을 통하여 실시간으로 데이터가 서버에 저장되며, 사용자의 카드 분실시의 문제점도 해결 될 수 있었다.

참고문헌

- [1] SD Specifications Physical Layer Simplified Specification Version 2.00 ,September 25, 2006
- [2] S3C2450 USER'S MANUAL 16/32-Bit RISC Microprocessor REV1.11 ,August 2008
- [3] Jet-Speed Storage, <http://www.taejin.co.kr>
- [4] IA650 SD/SDIO2.0 Combo Controller IC , <http://www.innoasic.com>