임베디드 리눅스에서의 가상 SD 메모리 카드 시스템 설계

문지훈* · 오재철**

Design of the Virtual SD Memory Card System on the Embedded Linux

Ji-Hoon Moon* · Jae-Chul Oh**

요 약

SD 메모리 카드는 휴대용 디지털 장비에서 널리 사용하고 있으며, 저장 장치로 대부분 낸드 플래시 메모리를 사용하여, 저 비용으로 사용자의 중요한 데이터를 안전하게 저장하는 특징을 가지고 있다. 하지만 낸드 플래시 메모리를 저장 장치로 이용하는 경우에 대용량의 데이터를 전송시 메모리 용량이 부족한 경우에 사용자의 데이터를 저장할 수 있는 방법이 존재하지 않는다. 본 논문에서는 가상 SD 메모리 카드 시스템을 제안한다. SD 메모리 카드에서 데이터를 저장하기 위해 메모리 코어로 플래시 메모리를 이용하는 방식이 아닌 외부 저장 장치를 이용하여 호스트에서 요청된 데이터를 SD 메모리 카드 디바이스 드라이버를 이용하여 처리하도록 하였다. 실험을 위해서 S3C2450 ARM CPU의 SMC 컨트롤러에 FPGA 기반의 SD 카드 슬레이브 컨트롤러 IP를 이용하여 테스트 하였다.

ABSTRACT

SD memory cards are widely used in portable digital devices, and most of them exploit NAND flash memory as their storage, so that they have a feature of storing users' important data safely with low costs. In case of using NAND flash memory as storage, however, there is no method to store users' data if memory capacity is insufficient when transferring a large volume of data. This paper proposes a virtual SD memory card system. It used a SD memory card device driver to process data requested from a host by exploiting external storage rather than by exploiting flash memory as a memory core for storing data to the SD memory card. For experiment, it used the FPGA-based SD card slave controller IP on the SMC controller with a S3C2450 ARM CPU to test.

키워드

SD Memory Card, Device Driver, Embedded System
SD 메모리카드, 디바이스 드라이버, 임베디드 시스템

* 순천대학교 컴퓨터학과(p3161@naver.com)

** 교신저자(corresponding author) : 순천대학교 컴퓨터학과(ojc@sunchon.ac.kr)

접수일자 : 2013. 11. 04

심사(수정)일자 : 2013. 12. 23

게재확정일자 : 2014. 01. 13

I. 서 론

IT 기술의 급격한 발달로 휴대 단말 기술은 빠른 속도로 고도화 되고 있다[1]. 이러한 특징으로 인하여 디지털 전자기기와 같은 내장형 시스템의 사용이 증가되고 있다. 이와 같은 내장형 시스템은 미리 정해진 특정 기능을 수행하기 위해 컴퓨터의 하드웨어와 소프트웨어가 조합된 전자 제어 시스템을 말하며, 필요에 따라서는 일부 기계가 포함될 수도 있다[2]. 이러한 디지털 장비에 사용자의 데이터를 저장하기 위해서 SD 메모리 카드가 보편적으로 사용된다.

SD 메모리 카드는 저장 장치로 낸드 플래시 메모리를 대부분 이용하고 있다. 휴대용 기기에서는 대부분 대용량 저장장치로 플래시 메모리를 사용하고 있다[3]. 최근의 낸드 플래시 메모리는 전력소모가 적고, 이동성과 랜덤 액세스 성능이 우수하여 개인용 컴퓨터를 비롯한 다양한 가전제품의 보조 기억장치로 사용되고 있다[4]. 낸드 플래시 메모리는 비휘발성 기억장치로서 하드 디스크에 비해 탐색시간이 빠르기 때문에 데이터 처리속도가 우수하며, 외부의 충격에 강하고, 크기가 하드 디스크에 비해 매우 작고, 무게도 가벼워 최근 모바일 기기나 이동식 저장매체, MP3 플레이어 등의 디지털 미디어 기기에서 광범위하게 사용되고 있다[5].

SD 메모리 카드는 스마트폰, 디지털 카메라, 디지털 캠코더 등에서 대중적으로 이용되고 있다. SD 메모리 카드의 경우 저장 장치로 낸드 플래시를 사용하여 jycabcd@lycos.co.kr 며, 정해진 기억 용량을 이용하므로 대용량의 디지털 사진 및 영상을 저장할 경우에 문제가 발생한다.

본 논문에서는 SD 메모리 카드 저장 매체로 플래시 디바이스를 이용하지 않고, 임베디드 시스템 보드에 연결된 외부 디바이스를 이용하여 SD 호스트가 요청하는 데이터에 대한 처리를 수행한다. 호스트에서 메모리 카드에 메모리 코어 읽기, 쓰기 데이터 처리를 요청할 경우 이에 필요한 메모리 블록 및 데이터 처리를 위한 SD 슬레이브 디바이스 드라이버 및 응용 프로그램을 개발 하였다. 이 프로그램을 통하여 외부 저장장치 특정 메모리 영역에 데이터 처리를 수행하도록 하였다.

2장에서는 기본적인 SD 메모리 카드 구조에 대해

서 설명하고, 3장에서는 제안하는 가상 SD 메모리 카드 시스템 설계에 대해 설명한다. 실험에서는 S3C 2450 CPU가 탑재된 임베디드 리눅스 시스템 환경과 SD 슬레이브 IP가 탑재된 FPGA 보드 환경을 구성하여 가상 SD 메모리 카드 시스템을 실험한다. 결론 및 앞으로서의 과제에서는 본 논문 실험 결과와 향후 과제에 대해 기술한다.

II. SD 메모리 카드 구조

SD 메모리 카드는 크게 통신을 위한 인터페이스, SD 카드를 제어하기 위한 컨트롤러, 데이터 저장을 위해 사용되는 메모리 부분으로 나누어진다[6]. 그림 1은 SD 카드의 구조를 나타낸다.

그림 1. SD 카드의 구조
Fig. 1 SD card architecture

그림 1은 SD 카드 구조를 나타내며, 외부 인터페이스는 9개의 핀으로 구성된다[7]. SD는 호스트와 카드간의 동기 신호를 위한 클럭 라인(CLK), 양방향으로 명령 송수신 및 응답을 받기 위한 커맨드 라인(CMD), 4개의 양방향 데이터 라인(DAT0 - DAT3), 전원 및 그라운드(VDD, VSS1, VSS2)라인이 있다[8].

SD 메모리 카드에는 카드 동작에 관련된 각종 정보를 가지고 있는 OCR(Operation Condition Register

r), CID(Card Identification), CSD(Card Specific Data), RCA(Relative Card Address), DSR(Driver Stage Register), SCR(SD Configuration Register)의 6개 레지스터가 있다[9][10]. 이 레지스터들은 카드가 시스템(호스트)에 연결되었을 때 시스템이 카드와 통신을 하면서 초기화 과정에 필요한 각종 정보를 가지고 있는 레지스터이다. OCR 레지스터는 카드 동작에 관련된 동작 전압에 대한 프로필이 저장 되어 있다. CID 레지스터는 카드의 제조 정보를 가지고 있다. CSD 레지스터는 카드의 내용을 액세스하기 위한 각종 정보를 가지고 있다. RCA 레지스터는 쓰기 가능한 16비트 상대 주소 레지스터로 초기화 과정동안 카드에 의해 생성되며, 이 주소는 카드 초기화 과정이 끝난 다음 카드와 호스트간에 통신을 위해 사용된다. DSR 레지스터는 카드의 출력 드라이버에 대한 설정에 관련된 레지스터이다. 마지막으로 SCR 레지스터는 SD 카드의 특별한 정보를 가지고 있다. 예를 들어 시큐리티, 데이터버스의 크기, 카드의 버전 정보 등을 가지고 있다[8]. 메모리 코어는 일반적으로 플래시 디바이스가 사용되며, 호스트에서 데이터를 읽거나 기록할 때 사용된다. 예를 들어 데이터 읽기를 호스트에서 요청한다면 CMD 라인을 통하여 데이터 읽기 명령이 전달되며, 데이터 라인을 통하여 읽을 메모리 코어 내부의 위치 주소가 전달될 것이다. 메모리 카드 인터페이스 컨트롤러는 이를 분석하여 메모리 코어 인터페이스를 통하여 코어에 해당 메모리 주소로부터 데이터를 읽어들이어 카드 인터페이스 컨트롤러에 데이터를 전달하게 된다. 데이터를 전달 받은 후 카드 인터페이스의 데이터 라인을 통하여 호스트가 요청한 데이터를 전달하게 된다.

III. 가상 SD 메모리 카드 시스템 설계

기존 SD 메모리 카드의 대부분 경우는 플래시 디바이스로 낸드 플래시를 이용한다. 고정된 플래시 디바이스를 이용할 경우 대용량의 메모리 용량인 경우는 큰 문제가 없지만, 그렇지 않는 경우는 용량의 부족함을 느낀다. 만일 디지털 카메라나 캠코더를 통해서 녹화를 하는 경우에는 아무리 큰 용량을 가진 플래시 디바이스라도 용량이 충분하지 않는 문제가 발

생하게 된다. 본 논문에서는 기존 SD 메모리 카드에 내장된 낸드 플래시 디바이스를 이용하는 방식이 아니라 외부 저장 또는 스토리지 저장 장치를 이용하는 가상 SD 메모리 카드 방식을 제안한다. 즉, SD 메모리 카드에 낸드 플래시 메모리가 존재하지 않고, 외부 저장 장치를 이용하여 SD 호스트가 요청하는 데이터나 커맨드를 전달해 주는 방법이다. SD 호스트 입장에서 통신시 호스트가 요구하는 커맨드 및 리스폰스를 SD 호스트가 요구하는 응답 시간내에 처리 된다면 아무런 문제가 되지 않는다. 그림 2는 가상 SD 메모리 카드 시스템 구성도를 나타낸다. SD 호스트와의 인터페이스 역할은 SD 슬레이브 IP와 연결되어 있다. SD 슬레이브 IP는 호스트에서 데이터 읽기, 쓰기를 요청 커맨드인 경우는 인터럽트를 타겟 보드 시스템에 요청한다. 타겟 보드의 SMC 컨트롤러에 연결된 가상 SD 메모리 카드 디바이스 드라이버는 인터럽트 발생을 확인한 후, 이를 데몬 프로세스에게 알린다. 응용 프로그램은 가상 SD 메모리 카드 디바이스 드라이버를 호출하여 데이터 읽기, 쓰기 명령인지를 확인한다. 만일 데이터 쓰기 요청인 경우라면 디바이스 드라이버를 통해서 FPGA의 SRAM 메모리 블록으로 데이터를 읽어 들인 후 이를 연결된 저장 장치의 메모리 블록에 저장한다. 데이터 읽기 요청인 경우는 SD 호스트로 데이터를 전달해야 하므로, 저장 장치의 메모리 블록에서 원하는 위치와 크기에 해당하는 데이터를 읽은 후 디바이스 드라이버에게 전달한다.

그림 2. 가상 SD 메모리 카드 시스템 구성도
Fig. 2 Block diagram of the virtual SD memory card system

3.1 가상 SD 메모리 카드 디바이스 드라이버

가상 SD 메모리 카드 디바이스 드라이버는 SMC와 연결 되어 있는 SD 슬레이브 IP 모듈의 레지스터에 OCR, CSD 레지스터를 설정 하도록 하여, 메모리 카드 용량 및 SD 메모리 카드 사용 전원 등을 호스트에 전달하는 역할을 한다. 또한 호스트에서 데이터 읽기, 쓰기를 요청할 경우 발생하는 인터럽트를 응용 프로그램에 전달하여, 외부 저장 장치로부터 데이터를 읽어오거나, 데이터를 기록할 수 있도록 하는 목적을 가지고 있다. 그림 3은 가상 SD 메모리 카드 시스템에서 사용되는 SROM 인터페이스 메모리 구조를 나타낸다. 현재 SD 슬레이브 IP의 내부 SRAM 메모리 블록과의 데이터 송수신을 위한 메모리 영역과 레지스터 파일을 읽고, 쓰기를 위한 영역으로 나누어져 있다. 데이터 송수신용 메모리 영역에 데이터를 기록하거나 읽어 들이는 호스트의 요청에 대한 응답을 수행한다. 레지스터 메모리 영역을 이용하여 레지스터 설정 및 호스트에서 요청하는 메모리 위치, 크기, 읽기인지 쓰기 요청인지를 알 수 있다. S3C2450 CPU의 SMC에서 사용 가능한 메모리 영역인 0x10000000과 0x18000000 메모리 주소를 이용한다[11]. 즉, 호스트에서 요청한 데이터를 읽거나 기록할 경우라면 0x18000000 메모리 위치에 이용하여 데이터 처리를 하며, 레지스터 영역을 읽을 경우라면 0x10000000 메모리 영역을 이용한다.

그림 3. SROM 인터페이스 메모리 구조
Fig. 3 SROM interface memory structure

3.2 응용 프로그램 구조

외부 저장장치를 이용하여 SD 슬레이브 IP가 요구하는 데이터 전달 및 저장을 쉽게 하기 위해서 응용 프로그램 설계가 필요하다. 외부 저장 장치를 타겟 보

드에 연결 후, 응용 프로그램을 이용하여 손쉽게 데이터를 저장하거나 저장 장치에서 데이터를 읽은 후 디바이스 드라이버로 데이터를 전송할 수 있다. 응용 프로그램은 사용할 외부 디바이스에 대한 마운트 완료 후, 가상 SD 메모리 카드 디바이스 드라이버의 인터럽트 요청을 대기한다. 인터럽트가 디바이스 드라이버로부터 발생한 경우, ioctl 시스템 콜을 이용하여 동작 종류를 파악한 후 데이터 읽기인지 쓰기 요청인지를 파악한다. 데이터 읽기 요청인 경우 디바이스 드라이버에 호스트가 요청한 데이터 크기 및 메모리 위치를 읽어들인 후, 마운트된 디바이스의 해당 메모리 위치에서 데이터를 읽은 다음 이를 디바이스 드라이버에 전달한다. 모든 데이터를 드라이버에 전달 완료한 후, SD 슬레이브 IP가 요청하는 처리를 모두 완료 되었음을 알리기 위해서 ARM_SET 레지스터에 '1'을 설정한다. 데이터 쓰기의 경우도 데이터 읽기와 유사한 방식으로 처리한다. 인터럽트 발생 후 operation 및 요청 크기 및 메모리 위치 등의 정보를 얻은 후, 디바이스 드라이버에서 데이터를 읽어들인 후 이를 마운트된 디바이스에 전달한다. 호스트가 요청한 데이터를 모두 전달한 후 ARM_SET 레지스터에 '1'을 설정하여 데이터 쓰기 요청이 마무리 됨을 SD 슬레이브 IP에 알린다. 그림 4는 데이터 읽기 및 쓰기를 위한 순서도를 나타낸다.

그림 4. 데이터 읽기/쓰기 함수 구조
Fig. 4 Data read/write function structure

IV. 실험

본 논문에서는 기존 SD 메모리 카드 내부의 플래시 메모리를 이용하여 사용자의 데이터를 저장하는 방식이 아닌, 외부 저장 장치를 이용하는 방식을 제안하였다. 그림 5는 가상 SD 메모리 카드 시스템 구조를 나타낸다. 실험 보드와 SD 호스트 사이에 SD 인터페이스 라인으로 연결되어 있다. 이 라인을 이용하여 커맨드, 리스폰스, 데이터 값들이 슬레이브와 통신하게 된다.

그림 5. 가상 SD 메모리 카드 시스템
Fig. 5 Virtual SD memory card system

그림 6은 보드 연결 후 SD 인터페이스를 SD 호스트에 연결한 후 디스크 포맷 화면을 나타낸다. 테스트를 위해서 SD 슬레이브 컨트롤러의 CSD 레지스터를 이용하여 메모리 용량은 32MByte로 설정하였다.

그림 6. 가상 SD 메모리 카드 포맷
Fig. 6 Format the virtual SD memory card

그림 7은 가상 SD 메모리 카드 로그 데이터를 나타낸다. [R],[W]은 SD 호스트 입장에서 데이터 읽기 쓰기를 나타낸다. 로그 데이터에서 Addr은 읽거나

쓰고자 하는 메모리 영역 위치를 나타내며, Sec는 Addr의 값을 512로 나눈 값을 나타낸다.

```
[R]:Addr[512],Sec[1]
[R]:Addr[1024],Sec[2]
[R]:Addr[1536],Sec[3]
[AR Spe]:Finish !!!
[R]:Addr[0],Sec[0]
[R]:Addr[512],Sec[1]
[W]:Addr[33553920],Sec[65535]
```

그림 7. 가상 SD 메모리 카드 로그 데이터
Fig. 7 Virtual SD memory card log data

IV. 결론 및 향후 과제

본 논문에서는 SD 메모리 코어에 플래시 메모리가 아닌 외장형 저장 장치를 타겟 보드에 연결하여 디바이스 드라이버에 의해서 SD 메모리 카드 용량을 설정 및 데이터를 저장하도록 하였다. 소프트웨어 방법으로 CSD 레지스터를 설정하여 대용량의 외장형 저장 장치나 네트워크 서버를 마운트 하여 사용한다면 대용량의 데이터 처리가 가능함을 실험을 통해서 확인하였다.

향후 과제로 일반적으로 SD 메모리 카드는 SD 호스트 연결 시 1초 안에 응답을 하지 않은 경우 대부분의 호스트는 SD 메모리 카드를 인식하지 못하는 특성을 가지고 있다. 논문에서는 개발의 편의성을 위해서 임베디드 리눅스 시스템을 이용하여, 외장형 저장 장치를 마운트 하였으므로 리눅스가 부팅이 완료된 후 SD 메모리 카드를 이용해야 하는 문제점을 해결해야 한다.

참고 문헌

- [1] J. Jun, J. Kim, Y. Kang, and S. Han, "Design of graphic middleware for unique UI style on mobile device," *J. of The Korea Institute of Electronic Communication Sciences*, vol. 4, no. 4, 2009, pp. 281-286.
- [2] K. Kim, G. Kim and J. Lee, "Embedded Linux System for Self-Control System of Car", *J. of T*

he Korea Institute of Electronic Communication Sciences, vol. 2, no.1, 2007, pp. 62-66, 2007.

- [3] C. Baik and Y. Lee, "A NAND Flash Controller for Mobile Devices," *Proc. of 5th KITT(Korean Institute of Information Technology) Summer Conference*, pp. 667-670, Gumi, South Korea, Jun 2007, pp. 667-670.
- [4] J. Kim, "Efficient Prefetching Scheme for Hybrid Hard Disk," *J. of The Korea Institute of Electronic Communication Sciences*, vol. 6, no. 5, 201, pp. 665-671.
- [5] M. Park, and S. Kim, "A Policy of Page Management Using Double Cache for NAND Flash Memory File System," *J. of Computing Science and Engineering*, vol. 36, no. 5, 2009, pp. 412-421.
- [6] S. Chung, "Design of Secure USB Memory based on Hardware Fingerprints," *M. S. Theses, Hanyang University*, 2011.
- [7] A. Edvardsson, "A wishbone compatible SD card mass storage controller for embedded usage," *Master's Thesis at Malardalen University*, 2009.
- [8] S. Kim, "A Study on Firmware Design for SD Memory Card Interface of CalmRISC," *M. S. Theses, KyungHee University*, Feb. 2002.
- [9] EP563 SD Card Slave Controller, <http://www.eurekatech.com/products/peripheral/ep563.htm>
- [10] SD Specification Part 1 Physical Layer Specification Ver 3.0, 2009.
- [11] AP app part, "S3C2450 16/32-Bit RISC Microprocessor User's manual," *Samsung Electronics, Inc.*, 2009.

저자 소개

문지훈(Ji-Hoon Moon)

2002년 동의대학교 컴퓨터공학과 졸업(공학사)

2004년 동의대학교 대학원 컴퓨터공학과 졸업(공학석사)

2009년 순천대학교 대학원 컴퓨터공학과 이학박사
과정 수료

※ 관심분야 : 정보보안, 임베디드시스템

오재철(Jae-Chui Oh)

1978년 전북대학교 컴퓨터공학과 졸업(공학사)

1982년 전북대학교 대학원 컴퓨터공학과 졸업(공학석사)

1988년 전북대학교 대학원 컴퓨터공학과 졸업(공학박사)

1984년~1986년 기전대학교 전자계산학과 전임강사

1986년~현재 순천대학교 컴퓨터공학과 교수

※ 관심분야 : 임베디드시스템, USN, 네트워크 설계 및 분석