

AMBA AHB 기반 SDRAM 컨트롤러 설계†

(Design of an SDRAM Controller for AMBA AHB-Lite)

김 상 돈*, 이 승 은**

(Sang Don Kim and Seung Eun Lee)

요 약 본 논문은 FPGA를 이용한 SDRAM Controller 설계 방법에 관하여 기술한다. 임베디드 시스템의 성능 향상과 함께, 대용량의 메모리를 지원하기 위하여 SDRAM이 사용되고 있으며, 이를 위해서는 SDRAM 컨트롤러의 설계가 요구된다. 본 논문에서는 FPGA에서 SDRAM 제어기를 구현함으로써 SDRAM을 사용할 수 있도록 하며 ARM코어로부터 제어되는 AHB-Lite 버스에서 SDRAM이 동작하는 결과를 보여준다.

핵심주제어 : SDRAM 제어기, FPGA, AMBA AHB-Lite

Abstract In this paper, we introduce a SDRAM controller implemented on FPGA. Modern embedded system adopts SDRAM as a memory to meet the high capacity memory demands. Our SDRAM controller is written in Verilog and verified on an FPGA, demonstrating the functionality along with ARM Cortex-M0, supporting AMBA AHB.

Key Words : SDRAM Controller, FPGA, AMBA AHB-Lite

1. 서 론

최근 시스템이 빠르게 발전함에 따라 요구되는 하드웨어 성능이 다양해지고 있으며 FPGA를 채택하는 시스템이 증가되고 있는 추세이다. FPGA를 이용한 시스템은 하드웨어 리소스 변화에 신속하게 대처할 수 있고 효율적으로 디지털 하드웨어를 구현할 수 있으며 개발기간을 단축할 수 있다는 장점이 있으나 대용량 메모리를 포함하고 있지 않으므로 외부 메모리를 사용하여야 한다. SDRAM은 동작속도가 빠르며 용량대비 가격이 저렴하기 때문에 FPGA의 메인메모

리로서 이용할 수 있으며 이미지 처리, 통신 등 여러 분야에서 사용할 수 있다. 그러나 FPGA는 SDRAM을 제어할 수 있는 제어기를 포함하고 있지 않으므로 메모리 제어기를 구현해야 한다.

FPGA와 SDRAM은 활용범위가 넓은 디바이스이며 이를 이용한 SDRAM 제어기는 적용 분야에 따라 다양하게 연구되고 있다. Jiayi Zhu는 고화질 영상압축 제품에 최적화된 메모리 제어기 구조를 제시하였고[2], 국내에서도 비디오 처리를 위한 고성능 SDRAM 제어기를 FPGA에서 구현하였다[3].

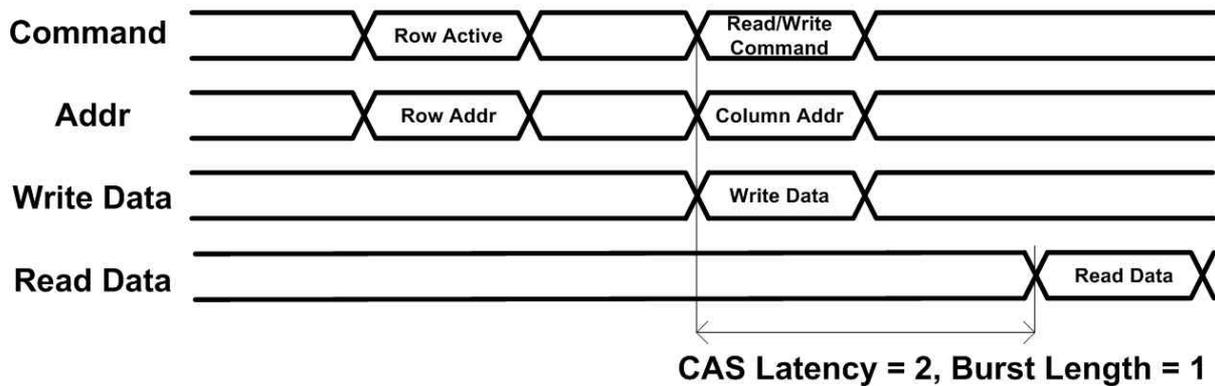
본 논문에서는 SDRAM 제어기에서 필수로 요구되는 읽기, 쓰기 및 Refresh기능을 포함한 SDRAM 제어기를 설계하는 방법에 대하여 기술한다. 실험결과를 설계된 메모리 제어기가 ARM 코어로부터 제어되는 AHB-Lite 버스 인터페이스를 포함하며, Refresh동작

† 본 연구는 서울과학기술대학교 교내 학술 연구비 지원으로 수행되었습니다.

* 서울과학기술대학교 전자공학과

** 서울과학기술대학교 전자공학과,

교신저자(seung.lee@seoultech.ac.kr)



<그림 1> SDRAM 읽기 및 쓰기 타이밍도

을 통하여 데이터를 정상적으로 유지하고 읽기 및 쓰기 동작을 수행할 수 있음을 보여준다. 본 논문에서 SDRAM 제어기는 Verilog HDL로 기술되었고 FPGA를 이용하여 구현되었다.

2. 본 론

2.1 SDRAM 제어기

SDRAM은 Transistor와 Capacitor를 이용하여 데이터를 저장하는 방식을 사용하므로 저가격으로 대용량 메모리를 구현할 수 있는 장점이 있다. 대용량 메모리는 주소를 입력하기 위하여 요구되는 주소 핀이 많아질 수 있으므로 SDRAM은 Matrix구조를 채택함으로써 주소를 2Cycle에 걸쳐 입력받는다. 따라서 효율적으로 외부 입력을 받아들일 수 있으며 Cell을 배치하기 용이한 구조를 채택하였다. 또한 복수의 Matrix를 Bank구조로써 사용하여 면적대비 용량을 극대화하였다. SDRAM Cell에 저장되는 디지털 데이터는 Capacitor에 충전된 전하로써 저장되며 이를 디지털 데이터로 변환할 수 있는 회로가 메모리 내부에 포함되어야 한다. Capacitor에 충전된 전하는 트랜지스터의 누설전류 문제로 인해 동작 중 데이터가 손실될 수 있는 문제점이 있다. 데이터가 손실되기까지의 시간은 SDRAM의 Retention Time으로 정의되며 Refresh를 수행함으로써 데이터를 보존하여야 한다. 따라서 메모리의 읽기 및 쓰기동작 뿐만 아니라 추가적인 작업이 요구되며

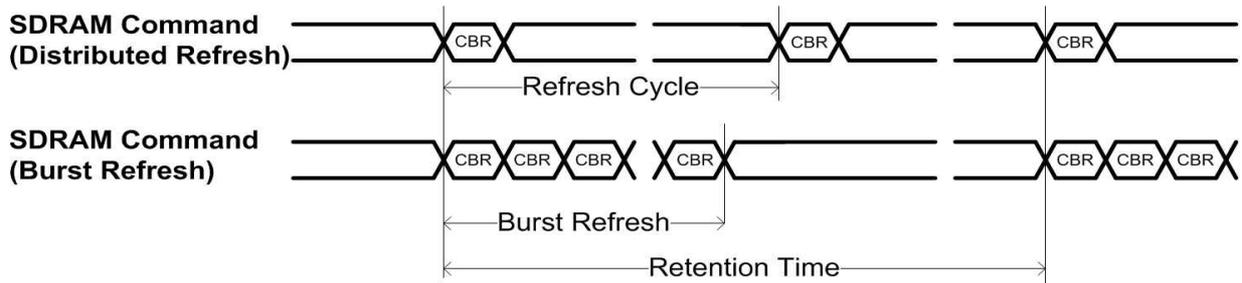
SDRAM을 사용하기 위한 별도의 제어가 필요하다.

2.2 SDRAM 초기화

SDRAM은 휘발성 메모리이므로 최초 전원 인가 후 초기화 과정을 수행해야 한다. SDRAM의 초기화 동작은 SDRAM 내부에 포함된 회로를 동작시키고 메모리 동작모드를 설정하기 위하여 반드시 필요하다. 메모리 동작 모드는 SDRAM의 Burst Length와 CAS Latency에 대한 설정을 수행하며 CAS Latency의 설정에 따라 데이터를 읽어올 수 있는 시간 및 메모리의 최대동작 주파수가 달라지므로 적절한 값이 선택되어야 한다.

2.3 SDRAM 읽기 및 쓰기동작

Matrix 구조는 SDRAM이 효율적으로 리소스를 사용하는데 도움이 되지만 제어기의 동작이 복잡해지는 결과를 가져온다. SDRAM의 읽기 및 쓰기동작을 제어하기 위한 타이밍도는 그림 1과 같다. 먼저 데이터를 읽거나 쓰기 위한 Row에 접근하기 위하여 Row Active 명령이 실행되어야 하며 동시에 Row Address를 입력한다. 다음 Read 또는 Write 명령과 Column Address를 입력함으로써 Cell을 동작시키고 데이터를 읽거나 쓸 수 있다. Data Write는 Column Address와 동시에 데이터를 인가하여 쓸 수 있으며 Read동작은 메모리 셀에 저장된 데이터를 디지털 레벨로 증폭하는 시간이 필요하므로 Column Address를 인가하고



<그림 2> SDRAM Refresh 타이밍도

CAS Latency 이후에 데이터를 읽어올 수 있다. SDRAM에서 CAS Latency는 2Cycle 또는 3Cycle이다. Burst Mode를 사용하여 미리 설정한 Burst Length만큼 연속적으로 데이터를 접근할 수 있으며 CAS Latency와 Burst Length 설정은 메모리 초기화 과정에서 설정된다. 따라서 CAS Latency에 의해 발생하는 데이터 지연을 보완하고 전체적인 메모리 동작 효율을 향상시킬 수 있다.

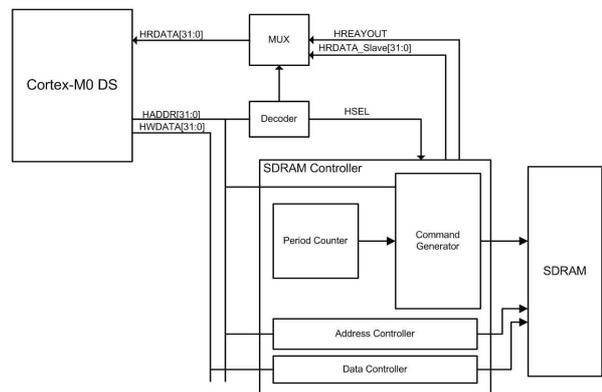
2.4 SDRAM Refresh

SDRAM에 데이터를 기록한 후 Retention Time을 경과하면 데이터가 손실될 수 있으므로 이미 기록된 데이터를 재기록 하는 과정이 필요하다. Refresh 동작은 제어기로부터 명령을 입력받음으로써 하나의 Row 단위로 수행된다. 모든 Row는 데이터를 손실하지 않기 위하여 Retention Time안에 Refresh되어야 한다. CBR Refresh는 SDRAM 내부의 카운터를 사용하여 동작을 수행할 Row를 자동적으로 선택하므로 효율적인 Refresh를 수행할 수 있다. Refresh는 정책에 따라 동일한 시간 간격으로 수행되는 Distributed Refresh와 메모리의 전체 Row를 연속적으로 Refresh하는 Burst Refresh를 사용할 수 있다. 그림 2는 Refresh 정책에 따른 타이밍도를 나타낸다. Burst Refresh 동작에서 *Burst Refresh* 구간에서 수행되는 CBR Refresh 명령의 개수는 SDRAM의 전체 Row 개수와 같으며 이 시간동안 다른 명령이 수행될 수 없다. 따라서 읽기 및 쓰기 명령은 Refresh가 끝날 때 까지 지연되는 단점이 있으며 Refresh 수행중이 아닌 경우 동일한 읽기 및 쓰기 Cycle을 보장한다. *Distributed Refresh*에서는 Refresh를 수행하는 간격으로 *Refresh Cycle*을 사용한다. Refresh Cycle은 Retention Time을 전체 Row

의 개수로 나눈 값과 같은 시간이며 읽기 및 쓰기 명령을 수행할 때 Refresh구간의 영향을 적게 받음으로써 Real-Time에 가까운 응답속도를 얻을 수 있다.

2.5 AHB-Lite 버스 인터페이스

설계된 SDRAM 제어기는 FPGA에 구현된 ARM 코어 IP의 메모리 영역으로써 이용될 수 있도록 AHB-Lite 인터페이스를 포함하여 설계되었다. 검증에 사용된 ARM 코어는 Cortex-M0 DS이며 하나의 Master를 이용할 수 있는 AHB-Lite 버스를 지원한다. SDRAM의 AMBA 인터페이스는 Slave로써 동작하며 Address Phase에서 접근할 메모리 주소를 선택하고 Data Phase에서 데이터를 접근한다. Refresh 정책은 메모리 제어기에서 결정되므로 버스 인터페이스는 Refresh 정책을 고려하지 않고 데이터를 접근하기 위한 명령을 전달하며 HREADY 신호를 포함하여 AHB-Lite 버스인터페이스에 적절한 응답을 생성함으로써 AMBA 버스를 사용할 수 있다.



<그림 3> AHB-Lite 버스 인터페이스를 포함한 SDRAM 제어기의 블록다이어그램



김 상 돈 (Sang Don Kim)

- 정회원
- 서울과학기술대학교 전자공학과 공학사
- 서울과학기술대학교 전자공학과 석사과정

• 관심분야 : 메모리 시스템, 멀티프로세서 플랫폼, 컴퓨팅 시스템, 메디컬 시스템

이 승 은 (Seung Eun Lee)

- 정회원
- KAIST 전기 및 전자공학과 공학사
- KAIST 전기 및 전자공학과 공학석사
- 캘리포니아대학교 어바인 전자컴퓨터공학과 공학박사
- 서울과학기술대학교 전자공학과 조교수
- 관심분야 : SoC Platform, Low Power Resilient System Design, Computing System, Medical IT Systems

논문 접수일 : 2013년 09월 13일

1차수정완료일 : 2013년 10월 08일

2차수정완료일 : 2013년 10월 15일

게재확정일 : 2013년 10월 16일