

# SoC 설계용 고성능 SDRAM Controller 설계

권 오 현, 양 훈 모, 이 문 기  
연세대학교 전기전자공학과 VLSI&CAD 연구실  
전화 : 02-2123-4731 / 핸드폰 : 011-9160-7432

## A Design of high performance SDRAM Controller for SoC design

Oh-Hyun Kwon, Hoon-Mo Yang, Moon-Key Lee  
Dept. of Electrical and Electronic Engineering, Yonsei University  
E-mail : kwon0108@spark.yonsei.ac.kr

### Abstract

In this paper, we propose a SDRAM Controller. The SDRAM is often used a mainstream memory as embedded system memory due to its short latency, burst access and pipeline features. The proposed Controller provides essential functions for SDRAM initialization, read/write accesses, memory refresh and Burst access. Furthermore, the proposed controller is implemented in the form of SOFT IP. Therefore, it reduces the designer's effort greatly.

### I. 서론

ASIC 기술이 발달하고, 공정 기술이 발달하게 됨으로 해서, 모든 기능 블록을 하나의 칩으로 구현하려는 SoC(System on a Chip)에 많은 관심을 가지고, 이를 시스템 개발에 많이 이용하고 있다.

이러한 SoC시스템의 구현에서 고성능 메모리를 많이 요구하고 있다. 특히 네트워크프로세서의 라우팅 테이블이나, 3D 가속기 같은 빠른 처리 속도를 요구하는 곳에서 많이 사용된다.

그리고 그런 시스템에서 사용되는 Memory에는 SDRAM, SRAM, Flash등 여러 종류가 있으나 SRAM, Flash등은 SDRAM에 비해 처리 속도는 빠르나, 용량이 크지 않고, 비용이 비싼 단점이 있다. 반면, 본 논문에서 염두에 둔 SDRAM(Synchronous DRAM)은 그것의 속도와 Burst Access 그리고 파이프라인 특성 때문에 임베디드 시스템 메모리 디자인에서 많이 이용되고 있다.

Controller는 SDRAM의 특징인 Refresh 같은 기능들을 해결해 줄 수 있어야 한다. 또한 Burst Mode 지원뿐만 아니라, Master와 연결할 수 있는 BUS Interface 부분을 고려해야하고 초기 SDRAM의 초기화와 여러 Mode를 설정해 줄 수 있는 기능이 필요하다.

이 논문에서 제안하는 SDRAM Controller는 시스템 버스로 AMBA<sup>TM</sup>를 이용하는 시스템과 연결하여 사용할 수 있을 뿐 아니라, 위에서 언급한 SDRAM Controller에서 요구되는 기능들을 포함하여 동작할 수 있도록 설계되었다.

본 논문의 내용은 II 장에서 제안한 SDRAM Controller의 특징과 구조를 살펴보고, III 장에서 시뮬레이션 결과를 통해 기능을 검증하고, 마지막에서 최종 결론을 제시 한다

## II. SDRAM Controller의 특징과 구조

### 2.1 SDRAM Controller의 특징

SDRAM Controller의 기본적인 역할은 그림 1에서 보이는 것처럼 BUS MASTER에서 전해지는 명령어, 데이터 그리고 해당 데이터의 주소를 받아서 이를 연결된 SDRAM사항에 맞게 조절하여 전달해 주는 역할을 한다.

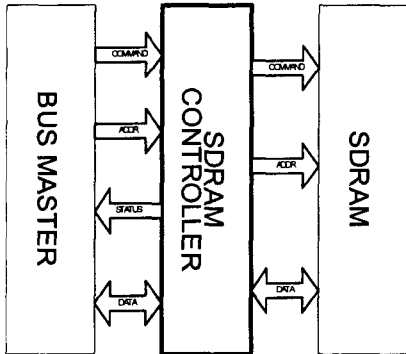


그림 1. SDRAM Controller System

본 논문에서 제안한 SDRAM Controller는 최근 가장 많이 이용되는 ARM Processor를 이용한 시스템에서 시스템 버스인 AMBA (Advanced Microcontroller Bus Architecture)와 인터페이스 할 수 있도록 설계되어 있다. 그러므로 Master에서 제공하는 HBURST 신호를 이용하여 BURST Mode 동작을 지원한다. 기본으로 4 Word BURST Mode로 설정되어 있으나, 설계자가 원할 경우, Single 혹은 2 Word 뿐만 아니라, 8 Word까지 지원이 가능하다. Dynamic 메모리에서 꼭 필요한 기능인 Refresh를 위해 Controller 자체적으로 수행할 수 있는 Self-Refresh 모드와 READ/WRITE 동작 후 Auto-Refresh 모드를 두어 Refresh 기능을 수행할 수 있게 하였다. 그리고 Power On 이후 이루어지는 Controller의 Mode 설정부분의 내용들을 Programmable 하게 할 수 있도록 하여, 설계자가 원하는 대로 바꿀 수 있게 하였다. 표 1에 위에서 언급한 사항들이 정리되어 있다.

| 내용            | 특징                        |
|---------------|---------------------------|
| BUS Interface | AMBA-AHB                  |
| DATA Width    | 32 bit                    |
| BURST Mode    | Programmable (1, 2, 4, 8) |
| CAS Latency   | Programmable (2, 3)       |
| Row address   | 2K, 4K, 8K                |

표 1. SDRAM Controller의 특징

### 2.2 SDRAM Controller의 구조

본 절에서는 앞 절에서 살펴본 특징을 가진 Controller의 구조를 살펴보고, 이를 이루고 있는 각 Block의 동작 원리와 특성에 대해 설명하겠다.

그림 2는 SDRAM Controller의 전체 구조를 보여준다. 그림에서 보이는 것과 같이 모두 6개의 Block으로 이루어져 있다. 그림 세부적으로 각 Block의 동작과 특징을 살펴보자.

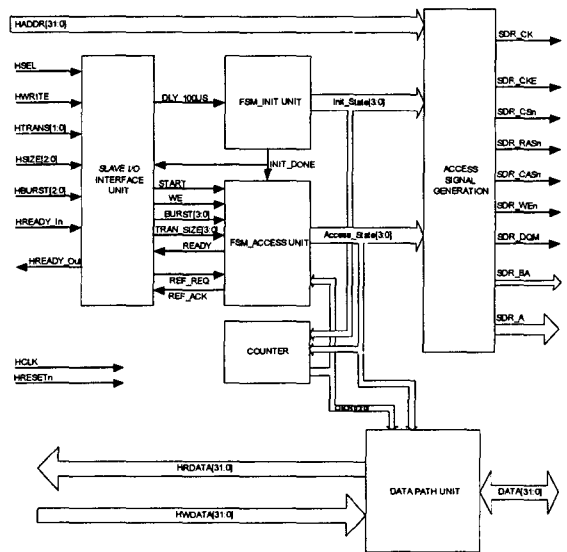


그림 2. SDRAM Controller의 전체 구조

#### 2.2.1 SLAVE I/O INTERFACE UNIT (SIU)

이 Block은 Bus Master와 인터페이스 하는 부분으로 Master에서 전해지는 신호를 받아서 Controller의 동작을 수행하는 역할과 SDRAM의 Refresh를 위한 카운터를 가지고 있어서 REF\_REQ(Refresh Request) 신호를 발생시키는 역할을 하고, 내부로부터 REF\_ACK 신호를 받아서 카운터를 RESET 하는 기능을 수행한다. 그리고 시스템이 Power On 되었을 때 초기 안정화를 위한 Delay 시간인 100us를 카운트하여 안정화가 완료되었음을 알리는 DLY\_100US 신호를 만들어 초기화를 시작할 수 있도록 한다.

또한 Controller 동작의 완료여부를 나타내는 신호를 내부로부터 받아서, Master에 Controller의 현재 상태를 알려주는 역할을 한다. 이는 HREADY\_Out 신호를 통해 이루어진다.

그리고 HBURST 신호와 HSIZE 신호를 가지고 앞에서 설명한 BURST MODE 전송과 BYTE MODE 전송이 가능 하도록 하는 BURST, TRAN\_SIZE 신호를 만들어 위의 전송을 수행하게 된다.

### 2.2.2 FSM\_INIT UNIT (FIU)

이 부분은 시스템이 처음 Power On 되거나 RESET 되었을 때, Memory의 초기화를 위해 이에 해당하는 State를 생성하여 이를 다음 Block으로 넘겨주어서 각 State에 해당하는 신호들을 만들어 Memory에 전달 할 수 있도록 한다. 하지만 이Block은 SIU(Slave I/O Interface Unit)에서 전해지는 DLY\_100US의 신호가 HIGH일 경우, 동작을 시작하게 된다. 초기화가 끝나면 INIT\_DONE 신호를 발생하여, READ/WRITE 동작을 시작할 수 있도록 한다.

그림 3은 FIU 내부에서 발생하는 각각의 동작을 State Diagram으로 나타낸 것이다.

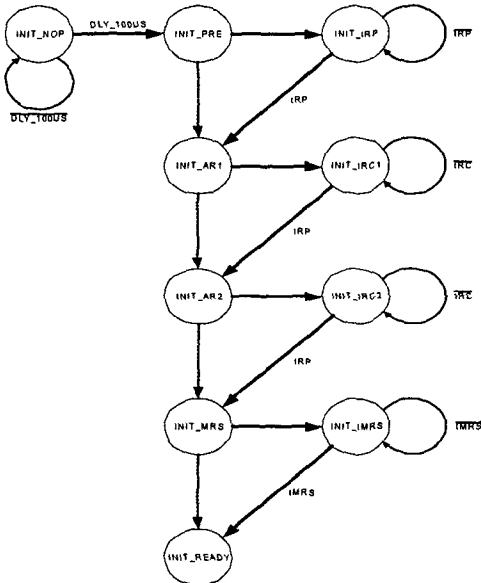


그림 3. FIU State Diagram

초기화가 완료되면, FIU에서 발생되는 State인 Init State는 INIT\_READY 상태가 되어 Memory를 READ/WIRTE 할 수 있는 상태로 있게된다.

### 2.2.3 FSM\_ACCESS UNIT (FAU)

이 Block은 Controller에서 주된 역할을 수행하는 부분이다. SIU로부터 들어오는 신호를 받아서 실질적인 동작을 수행할 수 있고, Master에서 요구하는 조건을 충족시키는 State를 발생하여 ACCESS SIGNAL GENERATION과 DATA PATH UNIT으로 전달하는 역할을 한다.

먼저 FAU는 SIU에서 발생된 STARTn 신호와 FIU에서 발생된 INIT\_DONE 신호를 받아 동작을 시작하게 되고, SIU의 WE 신호에 의해 READ와 WRITE를

구분하게 된다. 또한 BURST 신호와 TRAN\_SIZE 신호를 가지고 BURST Mode 전송과BYTE Mode 전송에 관한 State를 만들고, REF\_REQ 신호를 받아서 Memory의 Refresh에 해당하는 State를 만들게 된다. 그리고 COUNTER로부터 각 동작에 필요한 Parameter 값을 입력 받아 동작하게 된다.

### 2.2.4 ACCESS SIGNAL GENERATION (ASU)

ASU는 FIU와 FAU에서 받은 입력 State를 이용하여 직접 Memory에 Access 할 수 있는 신호를 만드는 역할을 한다. FIU에서 받은 State에는 Pre-Charge, Refresh, 그리고 Mode Register Setting을 위한 명령이 있는데, 이에 해당하는 SDRAM 신호 조합을 만들어 낸다. 그리고 Master로부터 받은 32-bit 주소를 Row, Column, 그리고 Bank 주소로 분리하여 SDR\_A, SDR\_BA 포트를 이용해 Memory에 전달하는 역할을 한다.

여기서, SDR\_A 신호의 사이즈에 의해 Memory의 Row 주소의 크기가 정해지는데, 본 논문에서 설계한 Controller는 SDR\_A의 사이즈가 12bit 이므로, 4K의 Row 주소까지 사용될 수 있다.

### 2.2.5 DATA PATH UNIT (DPA)

이 Block은 Master로부터 오는 Write 데이터를 받아 Memory에 전달하거나, Master에서 요구하는 데이터를 Memory로부터 읽어 Master에 전해주는 역할을 한다. Master와 DATA를 주고 받을 경우는, READ와 WRITE를 구분하여 동작하도록 하였고, Memory와의 전송은 하나의 포트를 이용하여 Bi-Direction으로 동작하도록 설계하였다. 그리고 Memory와의 데이터 전송에서 32-bit의 사이즈만을 지원하기 때문에, Memory의 데이터 입출력이 32-bit 사이즈인 Memory만 사용이 가능하다. 하지만, Master에서 사용하는 데이터가 32 bit이므로 4, 8, 16 bit의 데이터 입출력을 갖는 Memory를 사용할 경우, 32 bit보다 데이터 전송에 있어서 Clock 손실을 가질 수 있다.

### 2.2.6 COUNTER

Counter는 Clock의 수를 카운트 하는 역할을 수행한다. 각 동작을 수행하면서 요구되는 Parameter 값을 계산할 수 있도록Clock 수를 필요한 Block으로 전달하는 역할을 하고, 각 동작이 완료되는 시점을 FIU와 FAU로부터 State 상태로 입력 받아 Clock 수를 Reset 하게 된다.

