

**AESOP Embedded Forum**

한기태 ([Http: //www.aesop-embedded.org](http://www.aesop-embedded.org))

**개 요**

이 문서는 S3C4600 데이터시트에서 제 11장 DMA를 번역한 자료입니다.  
오역이 있을 수 있으니 이 점 양해 부탁드립니다.

**제 35장. DMA**

이 장에서는 S3C6400X RSIC 마이크로프로세서의 DMA 컨트롤러에 대해서 설명합니다.

**11.1 OVERVIEW**

S3C6400X는 4개의 DMA 컨트롤러를 가지고 있으며, 각각의 DMA 컨트롤러는 8개의 전송채널로 구성되어 있다.

DMA컨트롤러의 각 채널은 아무런 제한 없이 AHBtoAXI 브리지를 통해 AXI PERIPHERAL버스와 AXI SYSTEM버스 안의 장치 사이의 데이터전송을 할 수 있다.

다른 말로 하면, 각각의 채널은 다음과 같은 4가지 경우를 처리할 수 있다.

- 1) Source와 destination이 SYSTEM버스에 있는 경우
- 2) Destination이 PERIPHERAL버스를 사용 가능한 동안 source가 SYSTEM버스를 사용 가능한 경우
- 3) Destination이 SYSTEM버스를 사용 가능한 동안 source가 PERIPHERAL버스를 사용 가능한 경우
- 4) Source와 destination이 PERIPHERAL버스를 사용 할 경우

S3C6400X DMA컨트롤러는 ARM PrimeCell DMA controller PL080을 사용한다.

DMAC는 ARM사에 의해서 개발되고, 테스트되고, 라이선스 된 System-on-Chip (Soc) 주변장치의 Advanced Microcontroller Bus Architecture (AMBA) 이다.

DMAC는 하나의 AMBA AHB모듈이고 Advanced High performance Bus(AHB)에 접속한다.

DMA의 가장 큰 이점은 CPU의 조정 없이 데이터를 전송할 수 있는 것 이다.

DMA의 연산은 S/W에 의해서 초기화 될 수 있고 또는 내부 peripherals이나 외부 요청 핀에 의해서 초기화 될 수 있다.

## 11.2 FEATURES

DMA컨트롤러는 다음의 특징들을 제공한다.

- ? S3C6400X는 네 개의 DMA컨트롤러를 가지고 있으면 각각의 DMA컨트롤러는 8개의 전송채널로 구성되어 있고 각 채널은 단 방향 전송을 지원한다.
- ? 각 DMA컨트롤러는 16개의 peripheral DMA 요청 선을 제공한다.
- ? DMAC에 접속된 각 주변장치는 burst DMA요청 또는 single DMA요청을 요구할 수 있다. DAM burst크기는 DMAC을 프로그램 해서 설정한다.
- ? memory-to-Memory, memory-to-peripheral, peripheral-to-memory, peripheral-to-peripheral 전송을 지원한다.
- ? scatter 또는 gather DMA는 linked lists의 사용을 통해 지원된다.
- ? 각 DMA채널은 특정한 하드웨어 우선권을 갖는데 DMA채널 0은 가장 높은 우선권을 가지며 그 아래로 채널7이 가장 낮은 우선권을 가진다. 만약 동시에 두 채널에서 요청이 발생하면 높은 우선권의 채널이 먼저 서비스된다.
- ? DMAC는 AHB 슬레이브 인터페이스 위의 DAM 컨트롤 레지스터의 조작을 통해서 프로그램 된다.
- ? 두 AXI버스 마스터들은 전송데이터를 위한 AHB to AXI 브리지를 통해 전달한다. 이 인터페이스는 DMA전송이 활성화될 때 데이터 전송에 사용된다.
- ? source와 destination 위한 주소 증가와 또는 비 증가.
- ? 프로그램 가능한 DMA burst 크기. DMA burst 크기는 전송데이터의 능력 증가를 위해 프로그램 된다. 일반적으로 burst 크기는 주변장치 안 FIFO의 크기 반으로 설정한다.
- ? 각 채널에 4워드의 내부 FIFO가 있다.
- ? 8, 16, 32 비트 폭의 트랜잭션을 제공한다.
- ? Separate and combined DMA error and DMA 카운트 인터럽트 요청. 프로세서에게 전달된 인터럽트는 DMA 오류에 의해서나 DMA카운터가 0이 될 때 (이것은 일반적으로 전송이 끝난다는 것을 나타내다) 발생된다.

세가지 인터럽트신호는 다음과 같이 사용된다.

- DMACINTTC 신호: 전송이 완료됐을 때.
- DMACINTERR 신호: 에러가 발생했을 때.
- DMACINTR신호: DMACINTTC, DMACINTERR 인터럽트 신호가 동시에 발생할 때 DMACINTR 인터럽트 요청은 소수의 인터럽트 컨트롤러 요청 입력을 가지고 있는 시스템에서 사용될 수 있다.

? interrupt 마스킹. DMA에러와 DMA 종료 카운트 인터럽트 요청에 의해서 마스크 될 수 있다.

? Raw interrupt status. The DMA error 와 DMA count raw interrupt status 마스킹 이전에 읽을 수 있다.

### 11.3 BLOCK DIAGRAM

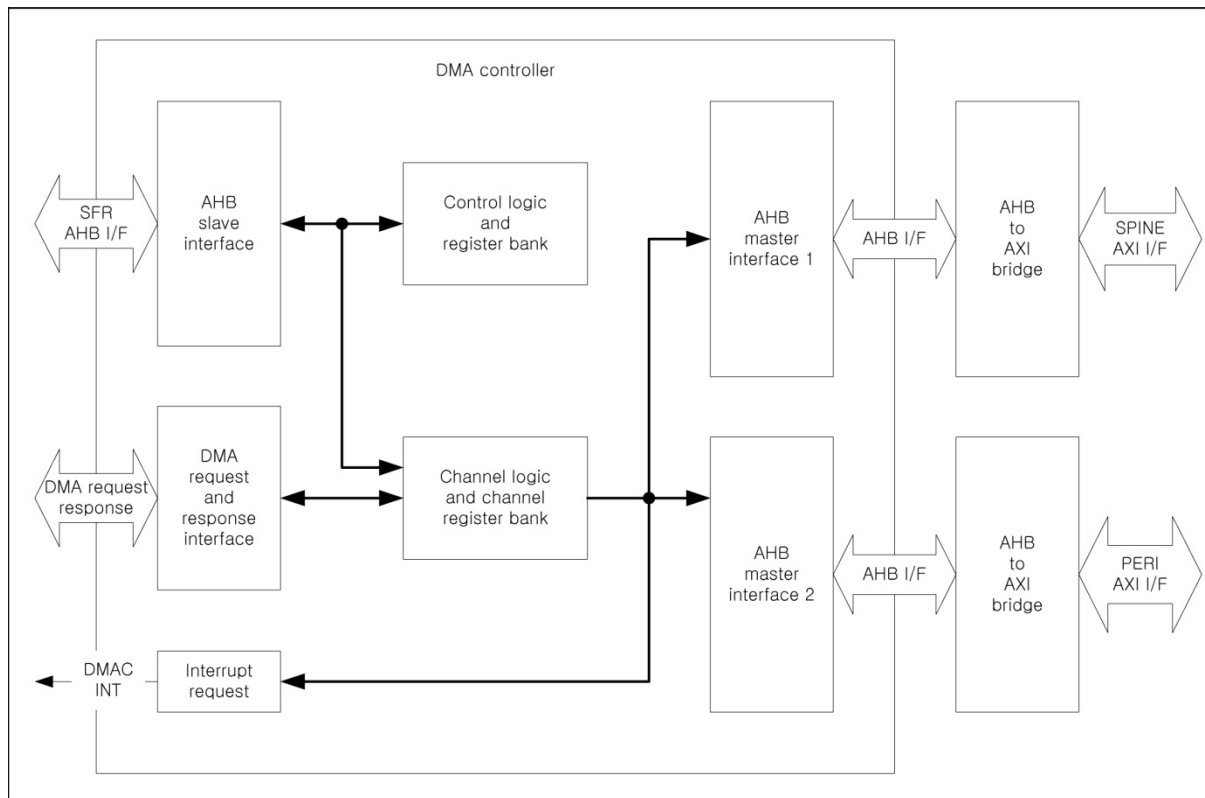


Figure 11-1. DMAC block diagram

## 11.4 DMA SOURCES

S3C6400x는 아래 표에 열거된 64개의 DMA source를 제공한다.

시스템 컨트롤러 안의 SDMA\_SEL 레지스터의 reset값은 SDMA선택을 의미하는 0x0이다.

그래서 환경설정은 general DMA 사용하기 위해서는 1로 set되어야 한다. 더 많은 정보는 System controller 부분을 참고해라

Group	DMA No.	Sources	Description
DMA0, SDMA0	0	DMA_UART0[0]	UART0 DMA source 0
DMA0, SDMA0	1	DMA_UART0[1]	UART0 DMA source 1
DMA0, SDMA0	2	DMA_UART1[0]	UART1 DMA source 0
DMA0, SDMA0	3	DMA_UART1[1]	UART1 DMA source 1
DMA0, SDMA0	4	DMA_UART2[0]	UART2 DMA source 0
DMA0, SDMA0	5	DMA_UART2[1]	UART2 DMA source 1
DMA0, SDMA0	6	DMA_UART3[0]	UART3 DMA source 0
DMA0, SDMA0	7	DMA_UART3[1]	UART3 DMA source 1
DMA0, SDMA0	8	DMA_PCM0_TX	PCM0 DMA TX source
DMA0, SDMA0	9	DMA_PCM0_RX	PCM0 DMA RX source
DMA0, SDMA0	10	DMA_I2S0_TX	I2S0 TX DMA source
DMA0, SDMA0	11	DMA_I2S0_RX	I2S0 RX DMA source
DMA0, SDMA0	12	DMA_SPI0_TX	SPI0 TX DMA source
DMA0, SDMA0	13	DMA_SPI0_RX	SPI0 RX DMA source
DMA0, SDMA0	14	DMA_HSI_TX	MIPI HSI DMA TX source
DMA0, SDMA0	15	DMA_HSI_RX	MIPI HSI DMA RX source
DMA1, SDMA1	0	DMA_PCM1_TX	PCM1 DMA TX source
DMA1, SDMA1	1	DMA_PCM1_RX	PCM1 DMA RX source
DMA1, SDMA1	2	DMA_I2S1_TX	I2S1 TX DMA source
DMA1, SDMA1	3	DMA_I2S1_RX	I2S1 RX DMA source
DMA1, SDMA1	4	DMA_SPI1_TX	SPI1 TX DMA source
DMA1, SDMA1	5	DMA_SPI1_RX	SPI1 RX DMA source
DMA1, SDMA1	6	DMA_AC_PCMout	AC97 PCMout DMA source
DMA1, SDMA1	7	DMA_AC_PCMin	AC97 PCMin DMA source
DMA1, SDMA1	8	DMA_AC_MICin	AC97 MICin DMA source
DMA1, SDMA1	9	DMA_PWM	PWM DMA source
DMA1, SDMA1	10	DMA_IrDA	IrDA DMA source
DMA1, SDMA1	11	DMA_EXTERNAL	External DMA source
DMA1, SDMA1	12	Reserved	
DMA1, SDMA1	13	Reserved	
SDMA1	14	DMA_SECU_RX	Security RX DMA source
SDMA1	15	DMA_SECU_TX	Security TX DMA source

## 11.5 DMA INTERFACE

### 11.5.1 DMA REQUEST SIGNALS

DMA 요청신호는 데이터 전송을 요청하는 주변장치에 의해서 사용된다.

DMA 요청신호는 다음을 나타낸다.

? 데이터의 burst(multi-word) 전송 또는 single word인지를 나타낸다.

? 전송이 데이터 패킷 안의 마지막인지 아닌지를 나타낸다.

각 주변장치를 위한 DMA컨트롤러에 대한 DMA 요청 신호는 다음과 같다.

**DMACxBREQ:** Burst request signal. 이것은 전송될 words의 프로그램 된 burst수를 실행한다

**DMACxSREQx:** Single transfer request signal. 이것은 전송을 위한 단일 word을 실행한다. DMA 컨트롤러는 주변장치와의 단일 word전송한다.

### 11.5.2 DMA RESPONSE SIGNALS

DMA 응답신호는 DMA요청신호에 의해서 시작된 전송이 완료되었는지를 나타낸다. 요청신호는 또한 완료packet이 전송되었는지를 나타내는데 사용될 수 있다.

각 주변장치를 위한 DMA컨트롤러로부터의 DMA 응답신호는 다음을 같다.

**DMACxCLR<sub>x</sub>:** DMA clear or acknowledge signal.

**DMACxTC:** DMA terminal count signal.

DMACxCLR<sub>x</sub> 신호는 주변장치로부터의 DMA 요청신호를 승인한 DMA컨트롤러에 의해서 사용된다.

DMACxTC신호는 DMA 전송이 완료된 주변장치를 나타내기 위해서 DMA 컨트롤러가 사용한다.

### 11.5.3 TRANSFER TYPES

DMA 컨트롤러는 4종류의 전송을 지원한다.

? memory-to-peripheral

? peripheral-to-memory

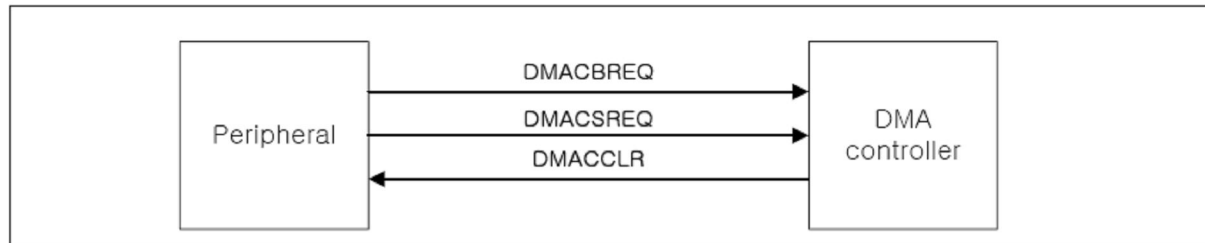
? memory-to-memory

? peripheral-to-peripheral.

각 전송은 주변장치 또는 DMA 컨트롤러에 의해서 흐름제어 된다. 그러므로 4가지의 제어 시나리오가 가능하다.

### 11.5.3.1 Peripheral-to-memory transaction under DMA controller flow control

Burst size의 배수가 아닌 트랜잭션을 위해서, 그림 11-2에서 같이 burst와 single 요청신호가 사용된다.

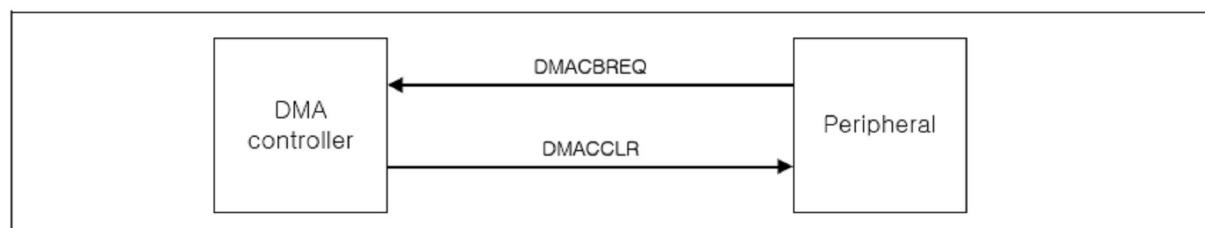


**Figure 11-2. Peripheral-to-memory transaction comprising bursts and single requests**

두 요청신호는 상호배타적이지 않다. DMA 컨트롤러는 전송 데이터의 남은 양이 burst size보다 큰 동안 DMACBREQ 신호를 모니터하고, 주변장치로부터 요청이 있을 때 burst 전송을 시작한다. 남은 데이터의 양이 burst size보다 작을 땐, DMA 컨트롤러는 DMACSREQ 신호를 모니터하고 요청 있을 때 Single 전송을 시작한다.

### 11.5.3.2 Memory-to- Peripheral transaction under DMA controller flow control

Burst size의 배수가 아닌 트랜잭션을 위해서, 그림 11-2에서와 같이 burst 요청신호만 사용한다. DMAC는 전송크기에 기초해서 전송하는 데이터의 양이 산정된다.



**Figure 11-3. Memory-to-Peripheral transaction comprising bursts that are not multiples of the burst size**

오직 DMACBREQ신호만 요청된다. DMA 컨트롤러는 전송할 남은 데이터의 양이 burst size보다 클 때 데이터의 full burst를 전송한다. 남은 데이터의 양이 burst size보다 작을 땐, DMAC는 다시 DMACBREQ를 모니터하고 요청이 있을 때 데이터의 나머지를 전송한다.

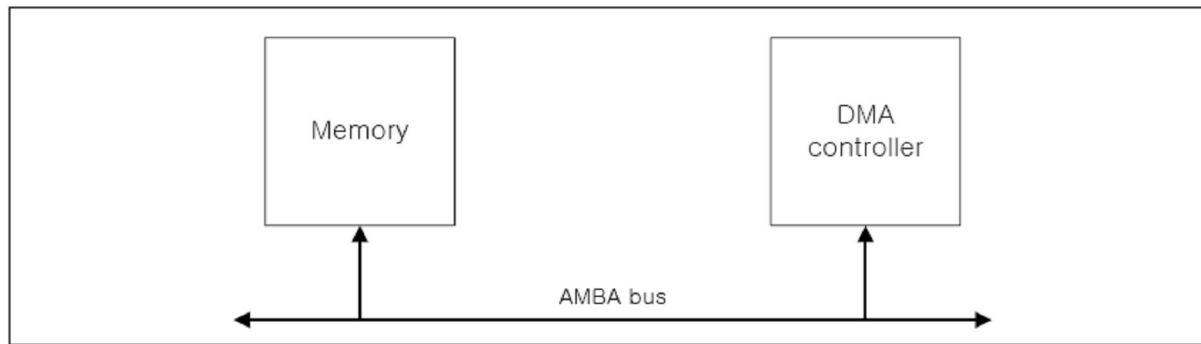
### 11.5.3.3 Memory-to-memory transaction under DMA controller flow control

소프트웨어는 DMA 채널 memory-to-memory 전송을 프로그램 한다. 전송이 활성화될 때, DMA 채널은 DMA 요청 없이 전송을 시작한다. 전송은 다음의 중의 한가지가 발생할 때까지 계속된다.

- ? 모든 데이터 전송이 완료 됐을 때
- ? 채널이 소프트웨어에 의해서 비 활성화 됐을 때

#### NOTE:

낮은 채널 우선권을 가지 채널로 memory-to-memory 전송을 프로그램 해야 한다. 만약 그렇지 않으면 다른 DMA 채널은 memory-to-memory 전송이 끝날 때까지 bus에 접근을 할 수 없거나, AHB 마스터는 어떤 트랜잭션도 수행할 수 없다.



### 11.5.3.4 Peripheral-to-peripheral transaction under DMA controller flow control

트랜잭션이 burst 크기의 배수가 아닐 때, 다음의 신호를 사용:

- ? source 주변장치의 single과 burst 요청 신호 (**DMACBREQ** and **DMACSREQ**)
- ? destination 주변장치의 burst 요청 신호 (**DMACBREQ**) .

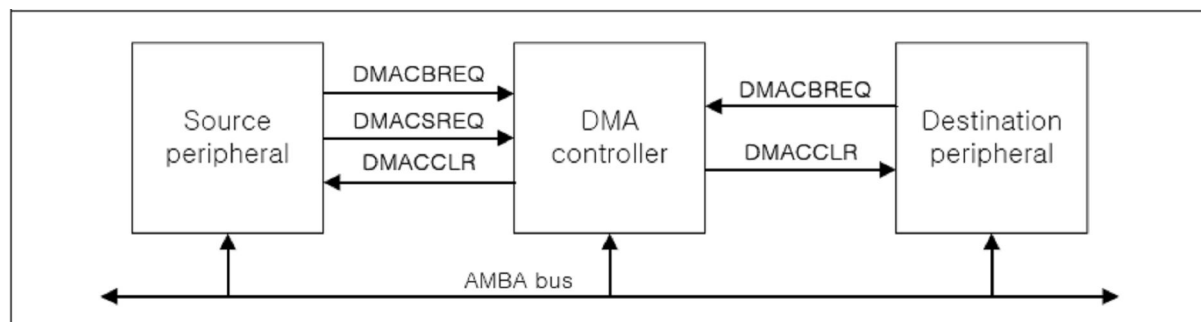


Figure 11-5. Peripheral-to-peripheral transaction comprising bursts and single requests

다음 LLI는 모든 읽기, 쓰기 전송이 완료 됐을 때 로드 된다. 주변장치로 마지막 데이터 전송을 나타내는데 DMACTC 신호를 사용할 수 있다.

Transfer Direction	Request Generator	Request Signals Used
Peripheral-to-Memory	Peripheral	DMACBREQ, DMACSREQ
Memory-to-Peripheral	Peripheral	DMACBREQ
Memory-to-Memory	DMA Controller	None
Peripheral-to-Peripheral	Peripheral	Src: DMACBREQ, DMACSREQ Des: DMACBREQ

**NOTE:** DMACSREQ 신호는 SW요청에 의해서만 생성될 수 있다.

#### 11.5.4 SIGNAL TIMING

DMA 신호의 타이밍 작용은 다음에 설명된다.

##### DMA request signal DMAC{L}(B/S)REQx

주변장치가 표시된 크기의 DMA전송을 진행할 준비가 되어 있는 것에 대해 DMA컨트롤러에게 통지한다.

Active HIGH. HCLK의 상승 에지에 DMA 컨트롤러에 의해서 추출된다. DMA요청신호는 DMACCLR 신호의 신호변경 수행에 관련하여 사용된다.

##### DMA Acknowledge or Clear DMACCLR<sub>x</sub>

DMA 전송이 완료된 슬레이브를 타나 낸다

Active HIGH

##### DMA Terminal Count DMACTC<sub>x</sub>

패킷의 끝에 도달된 슬레이브를 타나 낸다

Active HIGH

#### NOTE:

DMA 요청 소스가 DMA 컨트롤러와 같은 클럭을 사용하지 않는다면, 요청은 DMZCSync 레지스터 안의 적절한 비트를 셋 해서 동기화 되어 저야 한다.



## 11.6 FUNCTIONAL TIMING DIAGRAM

주변장치는 DMA 요구를 주장하고 활성화를 보전한다. DMACCLR 신호는 마지막 데이터 아이템이 전송되었을 때 DMA 컨트롤러에 의해서 발생된다.

주변장치가 DMAcCLR 신호가 활성화 되었다는 것을 인지했을 때 DMA 요청신호는 비 활성화 된다.

DMAC 컨트롤은 DMA 요청신호가 비 활성화 됐을 땐 DMACCLR신호를 발생하지 않는다.

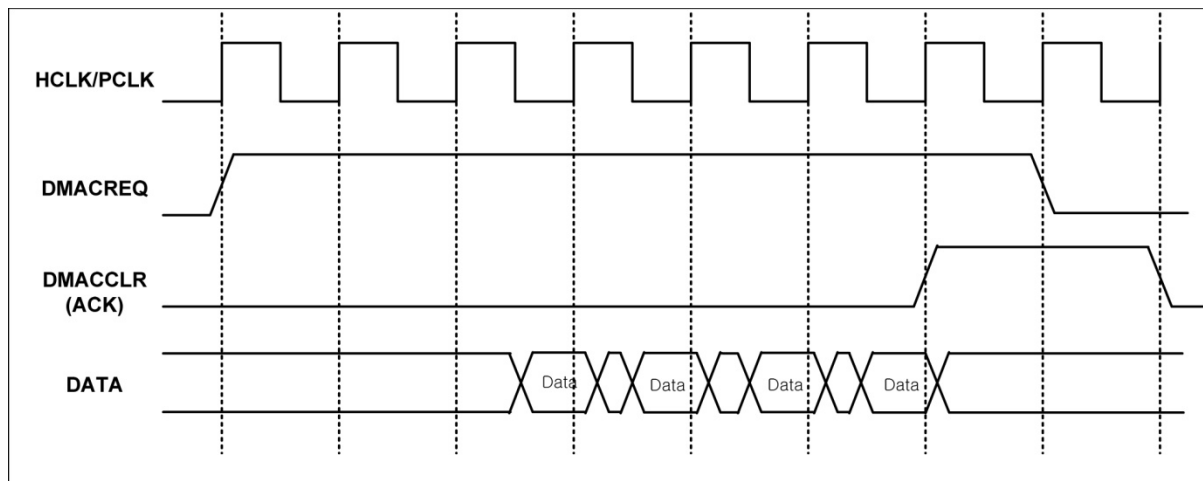


Figure 11-6. DMA interface timing

## 11.7 PROGRAMMER'S MODEL

### 11.7.1 PROGRAMMING THE DMA CONTROLLER

모든 AHB 슬레이브 프로그래밍 버스의 트랜잭션은 반드시 32 비트 폭이어야 한다.

### 11.7.2 ENABLING THE DMA CONTROLLER

DMA 컨트롤러의 작동시키기 위해선 DMAConfiguration 레지스터 안의 DMA Enable bit를 셋 해야 한다.

### 11.7.3 DISABLING THE DMA CONTROLLER

DMA컨트롤러의 작동을 중지하기 위해선 다음의 단계를 거쳐야 한다.

1. DMACEnbldChs 레지스터를 읽고 모든 DMA채널이 사용 중이 아니라는 것을 확인한다.  
만약 어떤 채널이 사용 중이라면 DMA채널이 사용되지 않을 때를 확인한다.
2. DMAConfiguration 레지스터 안의 DMA Enable bit를 조작해서 DMA 컨트롤의 사용을 중단할 수 있다.

### 11.7.4 ENABLING A DMA CHANNEL

DMA 채널은 DMA채널 설정 레지스터 안의 Channel Enable bit를 셋 해서 작동시킬 수 있다.

**NOTE:**

채널은 사용하기 전에 완전하게 초기화 되어 있어야 한다. 추가로, DMA컨트롤러의 Enable 비트는 어떤 채널이 작동되기 전에 셋 되 있어야 한다.

**11.7.5 DISABLING A DMA CHANNEL**

DMA채널은 다음의 세가지 방법으로 중지 할 수 있다.

? Channel Enable bit를 직접 조작한다. 만약 이 방법을 사용하면 FIFO안의 outstanding data는 손실된다.

? Channel Enable bit와 관련된 Active와 Halt bits를 사용한다.

? 전송이 완료될 때까지 기다린다. 채널은 자동으로 중단된다.

**11.7.5.1 Disabling a DMA channel and losing data in the FIFO:**

관련된 채널 설정 레지스터 안의 관련된 Channel enable bit를 클리어 해라.

현재의 AHB전송은(전송이 진행중인 경우)완료되고 채널은 작동 중단된다. FIFO안의 어떤 데이터는 손실된다.

**11.7.5.2 Disabling a DMA channel without losing data in the FIFO:**

FIFO안의 데이터 손실 없이 DMA채널을 중단하는 절차

1. 관련된 채널 설정 레지스터의 Halt bit를 셋 하면 앞으로 발생할 DMA 요청은 무시된다.
2. 관련된 채널 설정 레지스터의 Active bit가 셋 될 때까지 폴링한다.  
이 비트는 채널 안 의 어떤 데이터가 전송됐는지 아닌지를 나타낸다.
3. 채널 설정 레지스터의 관련 Channel bit를 클리어 한다.

**11.7.6 SET UP A NEW DMA TRANSFER**

새로운 DMA 전송 설정 절차

1. 채널이 DMA전송을 위한 설정이 안되 있다면
  - a. DMACEnbldChns 컨트롤 레지스터를 읽어서 사용되지 않는 채널을 찾아라
  - b. 요청된 우선순위의 사용되지 않는 채널을 선택한다.
2. DMA 컨트롤러를 프로그램 한다.

**11.7.7 HALTING A DMA CHANNEL**

채널 설정 레지스터에 관련 Halt bit를 셋 한다. 현재의 source 요청이 서비스되고 앞으로 발생하는 source DMA 요청은 Halt bit가 클리어 될 때까지 무시된다.

### 11.7.8 PROGRAMMING A DMA CHANNEL

#### DMA 프로그램 절차

1. Secure DMAC(SDMAC)또는 general DMAC(DMAC)를 사용할 건지 결정해라.  
General DMAC를 사용함에 따라, 시스템 컨트롤러의 secure DMA 컨트롤 레지스터(SDMA\_SEL)을 비 활성화 한다. (Reset value is SDMAC)
2. 요구된 우선권의 사용 가능한 DMA채널을 선택한다.  
DMA채널 0는 가장 높은 우선순위이고 DMA채널 7은 가장 낮은 우선순위이다.
3. DMACIntTCClr와 DMACIntErrClr 레지스터를 조작해서 채널에 사용된 pending 인터럽트를 클리어 한다. 이전 채널 연산은 인터럽트를 활성화 상태로 돌지도 모른다.
4. DMACCxSrcAddr 레지스터에 source주소를 쓴다.
5. DMACCxDestAddr 레지스터에 destination 주소를 쓴다.
6. DMACCxLLI 레지스터에 다음 LLI주소를 쓴다. 전송은 데이터의 단일 패킷으로 이루어지므로 다음의 레지스터는 설정해야만 한다.

Offset	Contents
Next LLI address	Source Address for next transfer
Next LLI address + 0x04	Destination Address for next transfer
Next LLI address + 0x08	Next LLI address for next transfer
Next LLI address + 0x0C	DMACCxControl0 data for next transfer
Next LLI address + 0x10	DMACCxControl1 data for next transfer

7. DMACCxControl 레지스터에 컨트롤 정보를 쓴다.
8. DMACCxConfiguration 레지스터에 채널 설정정보를 쓴다. Enable bit가 셋 되어 있으면 DMA 채널은 자동으로 활성화 된다.

## 11.8 REGISTER DESCRIPTION

4개의 DMA컨트롤러의 이름은 DMAC0, DMAC1, SDMAC0 그리고 SDMAC1이다.

DMAC0, DMAC1, SDMAC0 그리고 SDMAC1의 base 주소는 각 0x7500\_0000, 0x7510\_0000, 0x7DB0\_0000 그리고 0x7DC0\_0000이다.

OneNAND 컨트롤러를 위한 Page-access 기능은 DMAC0와 SDMAC0의 채널 3에 추가됐다.

### 11.8.1 DMA REGISTER LOCATION

Table 11-1. DMA register summary

Name	Type	Width	Description	Offset	Reset Value
DMACIntStatus	R	8	이 레지스터는 DMA 컨트롤러의 인터럽트상태를 제공한다. HIGH 비트는 특정 DMZ 컨트롤러 채널이 활성화라는 것을 나타낸다.	0x000	0x00
DMACIntTCStatus	R	8	이 레지스터는 인터럽트가 전송완료(종료 카운트)때문에 생성됐는지를 결정하기 위해서 사용된다. HIGH 비트는 전송이 완료된 것을 나타낸다.	0x004	0x00
DMACIntTCClear	W	8	이 레지스터를 쓸 때, 각 데이터 비트가 HIGH이면 DMACIntTCStatus와 DMACRawIntTCStatus 레지스터 안의 대응되는 비트를 클리어 되는 원인이 된다. 데이터 비트가 LOW이면 레지스터의 대응되는 비트에는 영향을 주지 않는다.	0x008	-
DMACIntErrorStatus	R	8	이 레지스터는 인터럽트가 에러 때문에 생성된 것인지를 결정하는데 사용된다.	0x00C	0x00
DMACIntErrClr	W	8	이 레지스터를 쓸 때, 각 데이터 비트가 HIGH이면 DMACIntErrorStatus와 DMACRawIntErrorStatus 레지스터 안의 대응되는 비트를 클리어 되는 원인이 된다. 데이터 비트가 LOW이면 레지스터의 대응되는 비트에는 영향을 주지 않는다.	0x010	-
DMACRawIntTCStatus	R	8	이 레지스터는 마스킹 전에 DMA 종료 카운트 인터럽트의 raw 상태를 제공한다. HIGH 비트는 인터럽트요정이 마스킹 전에 활성화된 것을 나타낸다.	0x014	-
DMACRawIntErrorStatus	R	8	이 레지스터는 마스킹 전에 DMA 에러 인터럽트의 raw 상태를 제공한다. HIGH 비트는 인터럽트요정이 마스킹 전에 활성화된 것을 나타낸다.	0x018	-
DMACEnbldChns	R	8	이 레지스터는 활성화된 DMA채널을 보여준다. HIGH 비트는 활성화된 DMA 채널을 나타낸다.	0x01C	0x00

**Table 11-1 DMA register summary (continued)**

Name	Type	Width	Description	Offset	Reset Value
DMACSoftBReq	R/W	16	이 레지스터는 DMA burst 요청이 소프트웨어에 의해서 생성되는 것을 허가한다.	0x020	0x0000
DMACSoftSReq	R/W	16	이 레지스터는 DMA single 요청이 소프트웨어에 의해서 생성되는 것을 허가한다	0x024	0x0000
Reserved	-	16	-	0x028	0x0000
Reserved	-	16	-	0x02C	0x0000
DMACConfiguration	R/W	3	이 레지스터는 DMA 컨트롤러를 구성하는데 사용된다.	0x030	0x000
DMACSync	R/W	16	이 레지스터는 DMA 요청 신호를 위한 동기화 로직을 활성화 또는 비활성화 한다.	0x034	0x0000
DMACC0SrcAddr	R/W	32	DMA channel 0 source address.	0x100	0x00000000
DMACC0DestAddr	R/W	32	DMA channel 0 destination address.	0x104	0x00000000
DMACC0LLI	R/W	32	DMA channel 0 linked list address.	0x108	0x00000000
DMACC0Control0	R/W	32	DMA channel 0 control0.	0x10C	0x00000000
DMACC0Control1	R/W	32	DMA channel 0 control1.	0x110	0x00000000
DMACC0Configuration	R/W	19	DMA channel 0 configuration register.	0x114	0x000000
DMACC1SrcAddr	R/W	32	DMA channel 1 source address.	0x120	0x00000000
DMACC1DestAddr	R/W	32	DMA channel 1 destination address.	0x124	0x00000000
DMACC1LLI	R/W	32	DMA channel 1 linked list address.	0x128	0x00000000
DMACC1Control0	R/W	32	DMA channel 1 control0.	0x12C	0x00000000
DMACC1Control1	R/W	32	DMA channel 1 control1.	0x130	0x00000000
DMACC1Configuration	R/W	19	DMA channel 1 configuration register.	0x134	0x000000
DMACC2SrcAddr	R/W	32	DMA channel 2 source address.	0x140	0x00000000
DMACC2DestAddr	R/W	32	DMA channel 2 destination address.	0x144	0x00000000
DMACC2LLI	R/W	32	DMA channel 2 linked list address.	0x148	0x00000000
DMACC2Control0	R/W	32	DMA channel 2 control.	0x14C	0x00000000
DMACC2Control1	R/W	32	DMA channel 2 control.	0x150	0x00000000
DMACC2Configuration	R/W	19	DMA channel 2 configuration register.	0x154	0x000000
DMACC3SrcAddr	R/W	32	DMA channel 3 source address.	0x160	0x00000000
DMACC3DestAddr	R/W	32	DMA channel 3 destination address.	0x164	0x00000000
DMACC3LLI	R/W	32	DMA channel 3 linked list address.	0x168	0x00000000
DMACC3Control0	R/W	32	DMA channel 3 control0.	0x16C	0x00000000
DMACC3Control1	R/W	32	DMA channel 3 control1.	0x170	0x00000000
DMACC3Configuration	R/W	19	DMA channel 3 configuration register.	0x174	0x000000

**Table 11-1 DMA register summary (continued)**

Name	Type	Width	Description	Offset	Reset Value
DMACC4SrcAddr	R/W	32	DMA channel 4 source address.	0x180	0x00000000
DMACC4DestAddr	R/W	32	DMA channel 4 destination address.	0x184	0x00000000
DMACC4LLI	R/W	32	DMA channel 4 linked list address.	0x188	0x00000000
DMACC4Control0	R/W	32	DMA channel 4 control0.	0x18C	0x00000000
DMACC4Control1	R/W	32	DMA channel 4 control1.	0x190	0x00000000
DMACC4Configuration	R/W	19	DMA channel 4 configuration register.	0x194	0x000000
DMACC5SrcAddr	R/W	32	DMA channel 5 source address.	0x1A0	0x00000000
DMACC5DestAddr	R/W	32	DMA channel 5 destination address.	0x1A4	0x00000000
DMACC5LLI	R/W	32	DMA channel 5 linked list address.	0x1A8	0x00000000
DMACC5Control0	R/W	32	DMA channel 5 control0.	0x1AC	0x00000000
DMACC5Control1	R/W	32	DMA channel 5 control1.	0x1B0	0x00000000
DMACC5Configuration	R/W	19	DMA channel 5 configuration register.	0x1B4	0x000000
DMACC6SrcAddr	R/W	32	DMA channel 6 source address.	0x1C0	0x00000000
DMACC6DestAddr	R/W	32	DMA channel 6 destination address.	0x1C4	0x00000000
DMACC6LLI	R/W	32	DMA channel 6 linked list address.	0x1C8	0x00000000
DMACC6Control0	R/W	32	DMA channel 6 control0.	0x1CC	0x00000000
DMACC6Control1	R/W	32	DMA channel 6 control1.	0x1D0	0x00000000
DMACC6Configuration	R/W	19	DMA channel 6 configuration register.	0x1D4	0x000000
DMACC7SrcAddr	R/W	32	DMA channel 7 source address.	0x1E0	0x00000000
DMACC7DestAddr	R/W	32	DMA channel 7 destination address.	0x1E4	0x00000000
DMACC7LLI	R/W	32	DMA channel 7 linked list address.	0x1E8	0x00000000
DMACC7Control0	R/W	32	DMA channel 7 control0.	0x1EC	0x00000000
DMACC7Control1	R/W	32	DMA channel 7 control1.	0x1F0	0x00000000
DMACC7Configuration	R/W	19	DMA channel 7 configuration register.	0x1F4	0x000000

### 11.8.2 INTERRUPT STATUS REGISTER, DMACINTSTATUS

DMACIntStatus 레지스터는 read-only이며 마스크 후의 인터럽트의 상태를 나타낸다. HIGH bit는 특정한 DMA 채널 인터럽트 요청이 active된 것을 나타낸다. 요청은 에러나 종료카운트 인터럽트에 의해서 발생될 수 있다.

표 11-2는 DMACIntStatus 레지스터의 비트할당을 보여준다.

**Table 11-2. Bit Assignment of DMACIntStatus register**

DMACIntStatus	Bits	Type	Function
IntStatus	[7:0]	R	Status of the DMA interrupts after masking

### 11.8.3 INTERRUPT TERMINAL COUNT STATUS REGISTER, DMACINTTCSTATUS

DMACIntTCStatus 레지스터는 read-only며 마스크 후 종료 카운트의 상태를 나타낸다.

이 레지스터는 만약 combined 인터럽트 요청(DMACINTCOMBINE)이 인터럽트 요청에 사용된다면 DMACIntStatus 레지스터와 관련해서 사용되어야만 한다.

DMACINTTC 인터럽트 요청이 사용된다면 DMACIntTCStatus 레지스터를 읽어서만이 인터럽트 요청의 source를 판명할 수 있다.

표 11-3는 DMACIntTCStatus 레지스터의 비트할당을 보여준다

**Table 11-3. Bit Assignment of DMACIntTCStatus register**

DMACIntTCStatus	Bits	Type	Function
IntTCStatus	[7:0]	R	Interrupt terminal count request status

### 11.8.4 INTERRUPT TERMINAL COUNT CLEAR REGISTER, DMACINTTCCLEAR

DMACIntTCClear 레지스터는 read-only며 종료 카운트 인터럽트요청을 클리어 하는데 사용된다.

이 레지스터에 쓸 때, 각 데이터비트가 HIGH로 셋 되면 상태레지스터에 상응하는 비트가 클리어 된다. 데이터 비트가 LOW이면 레지스터 안의 상응하는 비트는 아무런 영향이 없다.

표 11-4는 DMACIntTCClear 레지스터의 비트할당을 보여준다

**Table 11-4. Bit Assignment of DMACIntTCClear register**

DMACIntTCClear	Bits	Type	Function
IntTCClear	[7:0]	W	Terminal count request clear

### 11.8.5 INTERRUPT ERROR STATUS REGISTER, DMACINTERRORSTATUS

DMACIntErrorStatus 레지스터는 read-only며 마스크 후의 에러요청의 상태를 나타낸다.

이 레지스터는 만약 combined 인터럽트 요청(DMACINTCOMBINE)이 인터럽트요청에 사용된다면 DMACIntStatus 레지스터와 관련해서 사용되어야만 한다.

DMCINTERROR 인터럽트 요청이 사용된다면 DMACIntErrorStatus 레지스터는 읽어야만 한다.

표 11-5는 DMACIntErrorStatus 레지스터의 비트할당을 보여준다.

**Table 11-5. Bit Assignment of DMACIntErrorStatus register**

DMACIntErrorStatus	Bits	Type	Function
IntErrorStatus	[7:0]	R	Interrupt error status

#### 11.8.6 INTERRUPT ERROR CLEAR REGISTER, DMACINTERRCLR

DMACIntErrClr 레지스터는 write-only며 에러 인터럽트 요청을 클리어 하는데 사용된다.

이 레지스터를 조작할 경우, HIGH인 각 데이터 비트는 상태레지스터에 부합하는 비트를 클리어 하는 이유가된다. LOW인 데이터 비트는 레지스터 안의 부합하는 비트에 아무런 영향 없다.

표 11-6는 DMACIntErrorStatus 레지스터의 비트할당을 보여준다

**Table 11-6. Bit Assignment of DMACIntErrClr register**

DMACIntErrClr	Bits	Type	Function
IntErrClr	[7:0]	W	Interrupt error clear

#### 11.8.7 RAW INTERRUPT TERMINAL COUNTER STATUS REGISTER, DMACRAWINTTCSTATUS

DMACRawIntTCStatus 레지스터는 read-only이다.

이 것은 DMA 채널은 마스킹 이전에 완료된 전송(종료 카운트 인터럽트)을 요구하는 것을 나타낸다. HIGH 비트는 종료 카운트 인터럽트 요청이 마스킹 이전에 활성화를 나타낸다.

표 11-7는 DMACRawIntTCStatus 레지스터의 비트할당을 보여준다

**Table 11-7. Bit Assignment of DMACRawIntTCStatus register**

DMACRawIntTCStatus	Bits	Type	Function
RawIntTCStatus	[7:0]	R	Status of the terminal count interrupt prior to masking

#### 11.8.8 RAW ERROR INTERRUPT STATUS REGISTER, DMACRAWINTERRORSTATUS

DMACRawIntErrorStatus 레지스터는 read-only이다. 이것은 DMA채널이 마스킹 이전에 에러 인터럽트가 요청된 것을 나타낸다. HIGH비트는 에러 인터럽트 요청이 마스킹 이전에 활성화 된 것을 나타낸다.

표 11-8는 DMACRawIntErrorStatus 레지스터의 비트할당을 보여준다

**Table 11-8. Bit Assignment of DMACRawIntErrorStatus register**

DMACRawIntErrorStatus	Bits	Type	Function
RawIntErrorStatus	[7:0]	R	Status of the error interrupt prior to masking



### 11.8.9 ENABLE CHANNEL REGISTER, DMACENBLDCHNS

DMACEnbldChns 레지스터는 read-only이며 DMACxConfiguraton 레지스터 안 Enable 비트에 의해서 타나 내진 DMA 채널이 사용 가능하다는 것을 나타낸다. HIGH 비트는 DMA 채널이 사용 가능하다는 것을 나타낸다. 비트는 DMA 전송이 완료되면 클리어 된다.

표 11-9는 DMACEnbldChns 레지스터의 비트할당을 보여준다

**Table 11-9. Bit Assignment of DMACEnbldChns register**

DMACEnbldChns	Bits	Type	Function
EnabledChannels	[7:0]	R	Channel enable status

### 11.8.10 SOFTWARE BURST REQUEST REGISTER, DMACSOFTBREQ

DMACSoftBReq 레지스터는 read/write며 그것은 DMA burst 요청이 소프트웨어에 의해서 생성되는 것을 허가한다. DMA 요청은 각 source를 위해서 일치하는 레지스터 비트를 1로 써서 생성될 수 있다. 레지스터 비트는 작업이 완료되면 클리어 된다. 이 레지스터에 0을 쓰는 것은 무의미하다.

레지스터를 읽어서 어떤 source가 DMA burst 전송을 요청했는지 알 수 있다. 요청은 주변장치 또는 소프트웨어 요청 레지스터에로부터 발생될 수 있다

표 11-10는 DMACSoftBReq 레지스터의 비트할당을 보여준다

**Table 11-10. Bit Assignment of DMACSoftBReq register**

DMACSoftBReq	Bits	Type	Function
SoftBReq	[15:0]	R/W	Software burst request

**NOTE:** 소프트웨어와 하드웨어 주변장치 요청을 동시에 사용하지 않는 것을 권고 한다.

### 11.8.11 SOFTWARE SINGLE REQUEST REGISTER, DMACSOFTSREQ

DMACSoftSReq 레지스터는 read/write며 소프트웨어에 의해 DMA single 요청이 발생하는 것을 허가한다. DMA 요청은 각 source를 위해서 일치하는 레지스터 비트를 1로 쓰면 생성될 수 있다. 레지스터 비트는 작업이 완료되면 클리어 된다. 이 레지스터에 0을 쓰는 것은 무의미하다.

레지스터를 읽어서 소스가 DMA burst 전송을 요청했는지 알 수 있다. 요청은 주변장치 또는 소프트웨어 요청 레지스터에로부터 발생될 수 있다

표 11-11는 DMACSoftSReq 레지스터의 비트할당을 보여준다

**Table 11-11. Bit Assignment of DMACSoftSReq register**

DMACSoftSReq	Bits	Type	Function
SoftSReq	[15:0]	R/W	Software single request

**NOTE:** 소프트웨어와 하드웨어 주변장치 요청을 동시에 사용하지 않는 것을 권고 한다.

#### 11.8.14 CONFIGURATION REGISTER, DMACCONFIGURATION

DMACConfiguration 레지스터는 read/write며 DMA컨트롤러의 연산을 설정하는데 사용된다. AHB 마스터 인터페이스는 리셋되면 little-endian 모드로 셋 한다.

표 11-12는 DMACConfiguration 레지스터의 비트할당을 보여준다

**Table 11-12. Bit Assignment of DMACConfiguration register**

DMACConfiguration	Bits	Type	Function
Reserved	[2:1]	R/W	Should be 0
E	[0]	R/W	DMA controller enable: 0 =disabled 1 =enabled. This bit is reset to 0.Disabling the DMA controller reduces power consumption.

#### 11.8.15 SYNCHRONIZATION REGISTER, DMACSYNC

DMACSync 레지스터는 read/write며 DMA요청 신호를 위한 동기화 로직을 활성화 또는 비 활성화 하는데 사용된다.

비트가 0으로 셋 되면 DMA요청의 특정 그룹을 위한 동기화 로직이 활성화 된다.

비트가 1로 셋 되면 DMA요청의 특정 그룹을 위한 동기화 로직이 비 활성화 된다.

이 레지스터는 리셋될 때 동기화 로직이 활성화되게 0으로 셋 된다.

표 11-13는 DMACSync 레지스터의 비트할당을 보여준다

**Table 11-13. Bit Assignment of DMACSync register**

DMACSync	Bits	Type	Function
DMACSync	[15:0]	R/W	DMA 요청 신호 활성화 또는 비활성화를 위한 DMA 동기화 로직이다. LOW비트는 DMACBREQ[15:0], DMACSREQ[15:0], DMACLBREQ[15:0], DMACLSREQ[15:0] 요청신호를 위한 동기화 로직이 활성화 된 것을 나타낸다. HIGH 비트는 동기화 로직의 비활성화를 나타낸다.

**NOTE:** 동기화 로직은 DMA 컨트롤러와 다른 클럭으로 돌아가는 주변장치에서 DMA요청이 발생될 때는 반드시 사용되어야 한다. DMA 컨트롤러와 같은 클럭으로 동작하는 주변장치를 위해서 동기화 로직을 비 활성화 하는 것은 DMA 요청 응답시간을 개선한다. 필요하다면, DMACCLR과 DMACTC같은 DMA 응답신호는 주변장치에 동기화 돼야만 한다.

#### 11.8.16 CHANNEL SOURCE ADDRESS REGISTER, DMACCXSRCAADDR

8개 DMACCxSrcAddr는 read/write며 전송된 데이터의 현재 source 주소를 포함한다.

각 레지스터는 적당한 채널이 활성화 되기 전에 소프트웨어에 의해서 직접 프로그램 된다. DMA 채널이 활성화될 때 이 레지스터는 업데이트된다.

? source 주소가 증가되면 업데이트 된다.

? 데이터의 완료 패킷이 전송될 때 다음의 linked list에 의해서 업데이트 된다.

채널이 동작할 때 레지스터를 읽으면 쓸모 있는 정보가 제공되지 않는다. 왜냐하면 소프트웨어가 변수 값을 읽는 중에도 채널은 전송을 진행하고 있기 때문이다. 이 레지스터는 채널이 정지되어야 읽을 수 있다. 이 경우 이 레지스터는 마지막으로 읽은 아이템의 source 주소를 보여준다

**NOTE:** source와 destination 주소는 source와 destination의 폭으로 정렬돼야만 한다

표 11-14는 DMACCxSrcAddr 레지스터의 비트할당을 보여준다

**Table 11-14. Bit Assignment of DMACCxSrcAddr register**

DMACCxSrcAddr	Bits	Type	Function
SrcAddr	[31:0]	R/W	DMA Source address

#### 11.8.17 Channel destination address register, DMACCxDestAddr

8개 DMACCxDestAddr는 read/write며 전송되는 데이터의 현재 목적지 주소(byte정렬된)를 나타낸다.

각 레지스터는 채널이 활성화되기 전에 소프트웨어에 의해서 직접 프로그램 해야 한다. DMA채널이 활성화 될 때, 레지스터는 destination 주소의 증가와 데이터의 완료 패킷이 전송될 때 다음에 따라오는 linked list에 의해서 업데이트 된다.

채널이 동작할 때 레지스터를 읽으면 쓸모 있는 정보가 제공되지 않는다. 왜냐하면 소프트웨어가 변수 값을 읽는 중에도 채널은 전송을 진행하고 있기 때문이다. 이 레지스터는 채널이 정지되어야 읽을 수 있다. 이 경우 이 레지스터는 마지막으로 읽은 아이템의 destination 주소를 보여준다

표 11-15는 DMACCxDestAddr 레지스터의 비트할당을 보여준다

**Table 11-15. Bit Assignment of DMACCxDestAddr register**

DMACCxDestAddr	Bits	Type	Function
DestAddr	[31:0]	R/W	DMA destination address

#### 11.8.18 CHANNEL LINKED LIST ITEM REGISTER, DMACCxLLI

8개 DMACCxLLI는 read/write며 다음 Linked List Item(LLI)의 워드로 정렬된 주소를 포함한다.

현재 LLI가 chain의 마지막이고, 일단 그것과 관련된 모든 DMA전송이 완료되면 DMA 채널이 비활성화 된다.

#### NOTE:

DMA 채널이 활성화 될 때 이 레지스터가 프로그램 하면 예측할 수 없는 부작용을 가진다. 몇몇 시스템을 위해서 더욱 효과적인 LLI들의 로딩을 만들기 위해선, LLI 데이터 구조는 4워드 정렬일 수 있다.

표 11-16는 DMACCxLLI 레지스터의 비트할당을 보여준다

**Table 11-16. Bit Assignment of DMACCxLLI register**

DMACCxLLI	Bits	Type	Function
LLI	[31:2]	R/W	Linked list item. Bits [31:2] of the address for the next LLI. Address bits [1:0] are.
R	[1]	R/W	Reserved, and must be written as 0, masked on read.
LM	[0]	R/W	AHB master select for loading the next LLI: LM = 0 = AHB master 1 LM = 1 = AHB master 2.

#### 11.8.19 CHANNEL CONTROL REGISTER, DMACCXCONTROL0

8개 DMACCxControl0는 read/write며 burst size나 전송 폭 같은 DMA 채널 조작 정보를 포함한다.

각 레지스터는 DMA 채널이 활성화 전에 소프트웨어에 의해서 직접 프로그램 된다.

채널이 활성화되면 레지스터는 데이터의 완료 패킷이 전송될 때 다음 linked list에 의해서 업데이트 된다.

채널이 동작하는 동안 레지스터를 읽으면 쓸모 있는 정보가 제공되지 않는다. 왜냐하면 소프트웨어가 변수 값을 읽는 중에도 채널은 전송을 진행하고 있기 때문이다. 이 레지스터는 채널이 정지되어야 읽을 수 있다.

표 11-17는 DMACCxControl0 레지스터의 비트할당을 보여준다

**Table 11-17. Bit Assignment of DMACCxControl0 register**

DMACCxControl	Bits	Type	Function
I	[31]	R/W	종료 카운트 인터럽트 enable bit, 이것은 현재 LLI가 종료 카운트 인터럽트를 유발할 것을 예상할지 안 할지에 관해 조작한다.
Prot	[30:28]	R/W	Protection. Refer to Table 11-20
DI	[27]	R/W	Destination increment. destination 주소는 각 전송 후에 증가된다.
SI	[26]	R/W	Source increment. 소스 주소는 각 전송 후에 증가된다.
D	[25]	R/W	Destination AHB master select: 0 = AHB master 1 (AXI_SYSTEM) selected for the destination transfer. 1 = AHB master 2 (AXI_PERI) selected for the destination transfer.

**Table 11-17. Bit Assignment of DMACCxControl register (continued)**

DMACCxControl	Bits	Type	Function
S	[24]	R/W	Source AHB master select: 0 = AHB master 1 (AXI_SYSTEM) selected for the source transfer. 1 = AHB master 2 (AXI_PERI) selected for the source transfer.
Dwidth	[23:21]	R/W	Destination transfer width. AHB 마스터 버스 폭보다 전송 폭이 넓으면 안 된다. source 와 destination 의 전송 폭은 서로 다를 수 있다. 하드웨어는 요구된 데이터에 따라 자동으로 packs 과 unpacks 된다.
SWidth	[20:18]	R/W	Source transfer width. AHB 마스터 버스 폭보다 전송 폭이 넓으면 안 된다. source 와 destination 의 전송 폭은 서로 다를 수 있다. 하드웨어는 요구된 데이터에 따라 자동으로 packs 과 unpacks 된다.
DBSize	[17:15]	R/W	Destination burst size. destination burst 전송 요구를 구성하는 전송 수를 나타낸다. 이 값은 destination 주변장치의 burst 크기에 또는 destination이 메모리인 경우, 메모리 바운더리로 셋 돼야만 한다. Burst 크기는 destination 주변장치의 DMACxBREQ 신호가 활성화 될 때 전송된 데이터의 양이다. Burst 크기는 AHB HBURST 신호와는 관련 없다.
SBSIZE	[14:12]	R/W	Source burst size. Source burst로 구성하는 전송의 수를 나타낸다. 이 값은 source 주변장치의 burst 크기에 또는 source가 메모리인 경우, 메모리 바운더리로 셋 돼야만 한다. Burst 크기는 source 주변장치의 DMACxBREQ 신호가 활성화될 때 전송된 데이터의 양이다. Burst 크기는 AHB HBURST 신호와는 관련 없다.
Reserved	[11:0]	R	Reserved

**Table 11-18. Source or destination burst size**

Bit value of SBSize or DBSize	Source or destination burst transfer request size
0b000	1
0b001	4
0b010	8
0b011	16
0b100	Reserved
0b101	Reserved
0b110	Reserved
0b111	Reserved

**Table 11-19. Source or destination transfer width**

Bit value of SWidth or DWidth	Source or destination width
0b000	Byte (8-bit)
0b001	Half word (16-bit)
0b010	Word (32-bit)
0b011	Reserved
0b100	Reserved
0b101	Reserved
0b110	Reserved
0b111	Reserved

**NOTE:** DMAC는 각 채널당 4 word의 내부 FIFO가 있다. 그래서 burst 크기와 전송 폭은 FIFO의 크기에 제한된다. 예를 들어, data 폭이 word라면 유효한 burst 크기는 4 이하이다. Data의 폭이 byte면 유효한 burst 크기는 16이하이다.

AHB 접근 정보는 source와 목적지 주변 장치에 전송이 일어날 때 제공된다. 전송 정보는 DMA 채널(DMACCxControl 레지스터의 Prot 비트, DMACCxConfiguratio 레지스터의 Lock bit)를 프로그램 해서 제공된다. 이 비트는 소프트웨어에 의해 프로그램 되고 주변장치는 이 정보를 필요하다면 사용할 수 있다. 정보의 3개 비트는 제공되고 표 11-20은 3개의 보호 비트의 목적을 보여준다

**Table 11-20. Protection bits**

Bits	Description	Purpose
0	Privileged or User	Indicates that the access is in User, or privileged mode: 0 = User mode. 1 = privileged mode. This bit controls the AHB HPROT[1] signal.
1	Bufferable or not bufferable	Indicates that the access is bufferable, or not bufferable: 0 = not bufferable. 1 = bufferable. 이 비트는 접근이 bufferable라는 것을 타나 낸다. 예를 들면, 이 비트는 AMBA 브릿지가 읽기에서 source 버스의 제로 대기 상태에서 자료를 받아들이는 슬레이브와 destination 버스를 중재하기 위하여 기다림 없이 읽기 완료할 수 있다는 것을 나타내기 위해서 이용될 수 있다. 이 비트는 AHB HPROT[2] 신호를 통제한다.
2	Cacheable or not cacheable	Indicates that the access is cacheable or not cacheable: 0 = not cacheable. 1 = cacheable. 이것은 접근이 cacheable인지를 나타낸다. 예를 들어, 이 비트는 8개의 burst의 첫 번째 read를 보고 처리가 한번에 하나씩 끝까지 전달되기 보다는 destination 버스에서 8개의 burst전부를 읽어서 전송할 수 있는 AMBA 브릿지를 나타내는데 사용된다. 이 비트는 AHB HPROT[3] 신호를 조정한다.

#### 11.8.20 CHANNEL CONTROL REGISTER, DMACCXCONTROL1

8개의 read/write DMACCxControl1 레지스터는 전송크기와 같은 DMA채널 컨트롤 정보를 포함한다.

각 레지스터는 소프트웨어에 의해 DMA채널이 활성화되기 전에 프로그램 된다. 채널이 활성화되면 레지스터는 데이터의 완료 패킷이 전송될 때 다음 linked list를 의해서 업데이트 된다.

채널이 동작하는 동안 레지스터를 읽으면 쓸모 있는 정보가 제공되지 않는다. 왜냐하면 소프트웨어가 변수 값을 읽는 중에도 채널은 전송을 진행하고 있기 때문이다. 이 레지스터는 채널이 정지되어야 읽을 수 있다.

표 11-21는 DMACCxControl1 레지스터의 비트할당을 보여준다

**Table 11-21. Bit Assignment of DMACCxControl1 register**

DMACCxControl	Bits	Type	Function
TransferSize	[24:0]	R/W	Transfer size. 이 필드에 데이터 전송 크기를 적는다. 이 필드를 읽은 값은 destination 버스에 완료된 전송 수를 나타낸다. 채널이 동작하는 동안 레지스터를 읽는 것은 소프트웨어가 값을 읽어서 처리하는 동안에도 채널이 전송을 계속할지도 모르기 때문에 유용한 정보를 제공하지 않는다. 이것은 채널이 활성화 된 후 비활성화 될 때만 사용 될 수 있다.

### 11.8.21 CHANNEL CONFIGURATION REGISTER, DMACCXCONFIGURATION

8개의 DMACCxConfiguration 레지스터는 read/write하며 DMA채널을 설정하기 위해서 사용한다. 레지스터는 새로운 LLI가 요구될 때 업데이트 되지 않는다.

표 11-22는 DMACCxConfiguration 레지스터의 비트할당을 보여준다

**Table 11-22. Bit Assignment of DMACCxConfiguration register**

DMACCxConfiguration	Bits	Type	Function
H	[18]	R/W	Halt: 0 = allow DMA requests. 1 = ignore further source DMA requests. 채널 FIFO의 내용은 비워진다. 이 값은 active와 channel enable bits가 DMA채널을 간단하게 비활성화하는데 사용된다.
A	[17]	R	Active: 0 = there is no data in the FIFO of the channel. 1 = the FIFO of the channel has data. 이 값은 Halt와 Channel Enable bits와 같이 DMA채널을 간단하게 비활성화하는데 사용된다.
L	[16]	R/W	Lock. 이 비트가 셋 되면 잠금 전송이 가능하다
ITC	[15]	R/W	Terminal count interrupt mask. 이 비트가 클리어 되면 관련된 채널의 종료 카운트 인터럽트는 마스크가 되지 않는다.
IE	[14]	R/W	Interrupt error mask. 이 비트가 클리어면 관련된 채널의 오류인터럽트는 마스크 되지 않는다.
FlowCntrl	[13:11]	R/W	Flow control and transfer type. 이 값은 흐름제어와 전송종류를 나타내는데 사용된다. 흐름 제어기는 DMA 컨트롤러만 제공된다. 전송 종류는 memory-to-memory, memory-to-peripheral, peripheral-to-memory, peripheral-to-peripheral이다.
OneNandModeDst	[10]	R/W	SDMAC0의 채널 3을 위한 이 비트는 OneNAND 컨트롤러를 위한 pagewrite 기능을 지원하기 위해서 사용된다. 이 비트가 1이고 destination 주소가 OneNAND 컨트롤러의 주소 필드를 가리키면, destination 주소 증가 셋팅은 OneNAND 컨트롤러의 01명령을 지원할 수 있다. 물론, 이 비트가 1일 때, D는 $iAHB$ 마스터 1, DI는 $iincrement_i$ , DWidth는 $iWord_i$ 이어야 하고 DBSize는 4의 배수이어야 한다. DMAC0 사용은 정확한 연산을 보장하지 않는 것을 주의 하십시오.
DestPeripheral	[9:6]	R/W	Destination peripheral. 이 값은 DMA destination 요청 주변장치를 선택한다. 전송 destination이 memory이면 이 필드는 무시된다.



DMACCxConfiguration	Bits	Type	Function
OneNandModeSrc	[5]	R/W	SDMAC0의 채널 3만을 위한 이 비트는 OneNAND 컨트롤러를 위한 pageread기능 지원에 사용된다. 이 비트가 1이고 source 주소가 OneNAND 컨트롤러의 주소 필드를 가리키면, source 주소 증가는 OneNAND 컨트롤러의 01명령을 지원할 수 있다. 물론, 이 비트가 1일 때, S는 $i$ AHB 마스터 1, SI는 $i$ increment $i$ , SWidth는 $i$ Word $i$ 이어야 하고 SBSIZE는 4의 배수이어야 한다. DMAC0 사용은 정확한 연산을 보장하지 않는 것을 주의 하십시오.
SrcPeripheral	[4:1]	R/W	Source peripheral. 이 값은 DMA source 요청 주변장치를 선택한다. 전송 source가 memory이면 이 필드는 무시된다.
E	[0]	R/W	<b>Channel enables.</b> 이 비트는 채널이 지금 활성화인지 아닌지를 나타낸다: 0 = channel disabled 1 = channel enabled. Channel Enable 상태 비트는 DMACEnbldChns 레지스터를 읽어서도 확인 할 수 있다. 채널은 이 비트를 셋팅해서 사용 가능하게 한다. 채널은 Enable bit를 클리어 해서 사용 가능하지 않게 만들 수 있다. 이것은 AHB전송(전송인 진행중인 경우)을 완료하게 하는 원인이 되고 채널은 그때 사용할 수 없게 된다. 채널의 FIFO 안에 있는 데이터는 손실된다. Channel Enable 비트를 단순히 셋팅해서 채널은 재 시작하면 예측할 수 없는 결과를 가지므로 채널은 완전하게 다시 초기화를 해야만 한다. 마지막 LLI가 도착되거나 또는 채널 에러를 만나면 채널은 비활성화되고 Channel Enable비트는 클리어 된다. 채널이 채널 FIFO안의 데이터손실 없이 비활성화돼야 한다면 Halt 비트는 다음의 DMA 요청을 무시할 수 있게 셋 되어야만 한다. Active 비트는 채널 FIFO안에 데이터가 있지 않다는 것을 나타내는 값인 0이 될 때까지 폴링되어야만 한다. 마지막으로 Channel Enable 비트는 클리어 될 수 있다.

표 11-23는 세가지 흐름제어의 비트 값과 전송종류 비트를 설명한다.

**Table 11-23. Flow control and transfer type bits**

Bits value	Transfer type
000	Memory to memory
001	Memory to peripheral
010	Peripheral to memory
011	Source peripheral to destination peripheral
100~111	Reserved