# 제13장 USB 디바이스 컨트롤러

## 개요

USB(universal serial bus) 디바이스 컨트롤러는 DMA 인터페이스를 갖는 고성능의 컨트롤러 솔류션을 제공한다. USB 디바이스 컨트롤러는 DMA를 갖는 대량 전송, 인터럽트 전송, 컨트롤 전송을 허용한다.

USB 디바이스 컨트롤러 지원:
- USB 스펙 버전 1.1과 호환되는 full speed USB 디바이스 컨트롤러
- 대량 전송을 위한 DMA 인터페이스
- FIFO를 갖는 5개의 엔드포인트
  EP0 : 16바이트(레지스터)
  EP1 : 64바이트 입출력 FIFO(듀얼 포트 비동기 RAM): 인터럽트 혹은 DMA
  EP2 : 64바이트 입출력 FIFO(듀얼 포트 비동기 RAM): 인터럽트 혹은 DMA
  EP3 : 64바이트 입출력 FIFO(듀얼 포트 비동기 RAM): 인터럽트 혹은 DMA
  EP4 : 64바이트 입출력 FIFO(듀얼 포트 비동기 RAM): 인터럽트 혹은 DMA
- 통합 USB 송수신기

### 형태

- USB 스펙 버전 1.1과 완벽하게 호환
- full speed(12Mbps) 디바이tm
- 통합된 USB 트랜시버
- 컨트롤, 인터럽트, 대량 전송으 fwldnjs
- FIFO를 갖는 5개의 엔드포인트:
  16바이트의 FIFO(EP0)를 갖는 1개의 양방향 컨트롤 엔드포인트
  64바이트의 FIFO(EP1, EP2, EP3, EP4)를 갖는 4개의 양방향 대량 엔드포인트
- 대량 엔트포인트를 수신 혹은 송신을 위한 DMA 인터페이스 지원
- 처리량을 최대로 하기 위한 독립적인 64바이트의 수신과 송신 FIFO
- suspend와 remote wakeup 기능을 지원

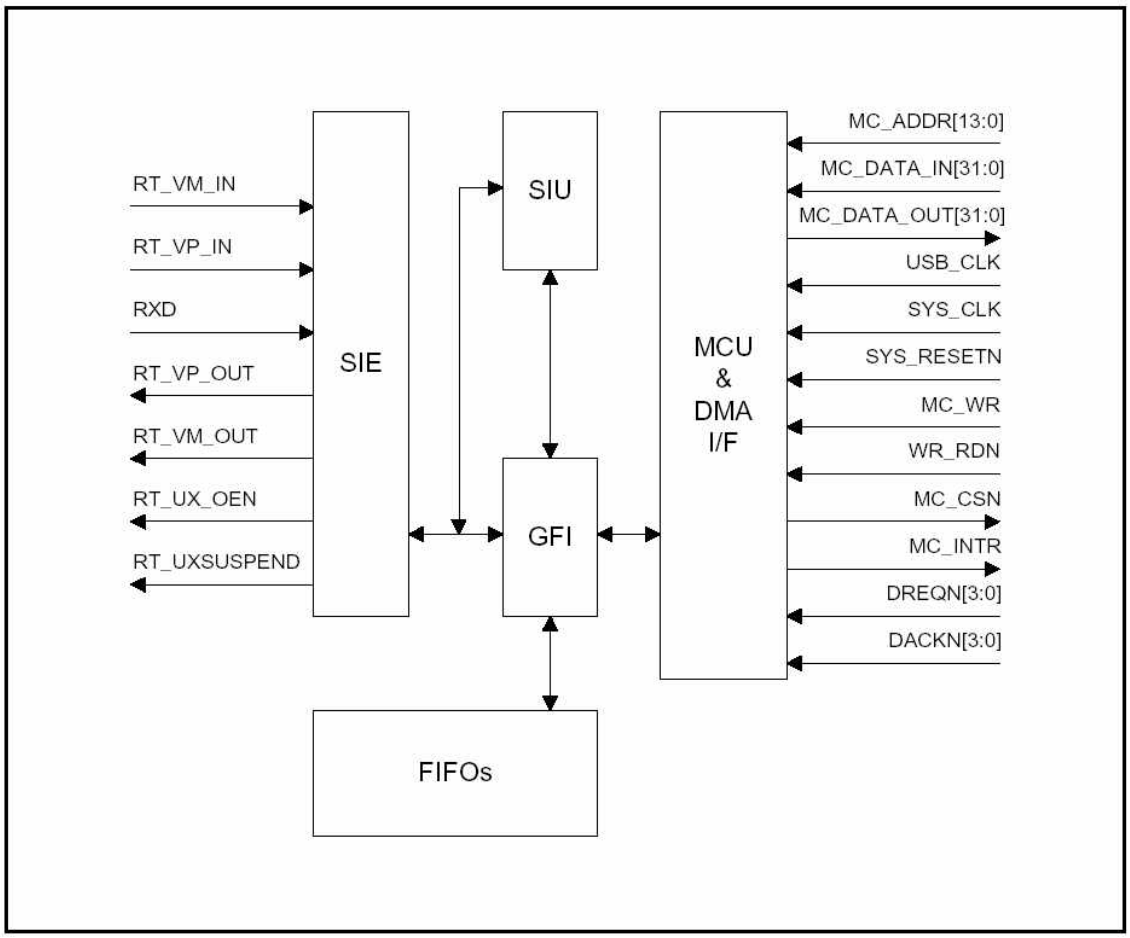


그림 13-1. USB 디바이스 컨트롤러 블록 다이어그램

## USB 디바이스 컨트롤러 특별 레지스터

여기서는 USB 디바이스 컨트롤러에 대한 세부 기능을 설명한다. 모든 특별 기능 레지스터는 바이트-접근과 워드-접근이 가능하다. 바이트 모드로 접근하면, 옵셋-어드레스는 리틀 엔디안과 빅 엔디안에서 차이가 있다. 모든 예약된 비트는 0이다. common indexed 레지스터는 INDEX 레지스터의 값에 달려있다. 예로, EP0 CSR 레지스터에 기록하려면, IN_CSR1 레지스터에 기록하기 전에 INDEX_REG에 '0x00'을 기록해야 한다.

NOTE

모든 레지스터는 호스트 리셋 신호가 동작한 후에 안정되어야 한다.

| Register Name | Description | Offset Address |
|---|---|---|
| **NON INDEXED REGISTERS** | | |
| FUNC_ADDR_REG | Function address register | 0x140(L) / 0x143(B) |
| PWR_REG | Power management register | 0x144(L) / 0x147(B) |
| EP_INT_REG (EP0–EP4) | Endpoint interrupt register | 0x148(L) / 0x14B(B) |
| USB_INT_REG | USB interrupt register | 0x158(L) / 0x15B(B) |
| EP_INT_EN_REG (EP0–EP4) | Endpoint interrupt enable register | 0x15C(L) / 0x15F(B) |
| USB_INT_EN_REG | USB Interrupt enable register | 0x16C(L) / 0x16F(B) |
| FRAME_NUM1_REG | Frame number 1 register | 0x170(L) / 0x173(B) |
| FRAME_NUM2_REG | Frame number 2 register | 0x174(L) / 0x177(B) |
| INDEX_REG | Index register | 0x178(L) / 0x17B(B) |
| EP0_FIFO_REG | Endpoint0 FIFO register | 0x1C0(L) / 0x1C3(B) |
| EP1_FIFO_REG | Endpoint1 FIFO register | 0x1C4(L) / 0x1C7(B) |
| EP2_FIFO_REG | Endpoint2 FIFO register | 0x1C8(L) / 0x1CB(B) |
| EP3_FIFO_REG | Endpoint3 FIFO register | 0x1CC(L) / 0x1CF(B) |
| EP4_FIFO_REG | Endpoint4 FIFO register | 0x1D0(L) / 0x1D3(B) |
| EP1_DMA_CON | Endpoint1 DMA control register | 0x200(L) / 0x203(B) |
| EP1_DMA_UNIT | Endpoint1 DMA unit counter register | 0x204(L) / 0x207(B) |
| EP1_DMA_FIFO | Endpoint1 DMA FIFO counter register | 0x208(L) / 0x20B(B) |
| EP1_DMA_TTC_L | Endpoint1 DMA transfer counter low-byte register | 0x20C(L) / 0x20F(B) |
| EP1_DMA_TTC_M | Endpoint1 DMA transfer counter middle-byte register | 0x210(L) / 0x213(B) |
| EP1_DMA_TTC_H | Endpoint1 DMA transfer counter high-byte register | 0x214(L) / 0x217(B) |
| EP2_DMA_CON | Endpoint2 DMA control register | 0x218(L) / 0x21B(B) |
| EP2_DMA_UNIT | Endpoint2 DMA unit counter register | 0x21C(L) / 0x21F(B) |
| EP2_DMA_FIFO | Endpoint2 DMA FIFO counter register | 0x220(L) / 0x223(B) |
| EP2_DMA_TTC_L | Endpoint2 DMA transfer counter low-byte register | 0x224(L) / 0x227(B) |

| Register Name | Description | Offset Address |
|---|---|---|
| EP2_DMA_TTC_M | Endpoint2 DMA transfer counter middle-byte register | 0x228(L) / 0x22B(B) |
| EP2_DMA_TTC_H | Endpoint2 DMA transfer counter high-byte register | 0x22C(L) / 0x22F(B) |
| EP3_DMA_CON | Endpoint3 DMA control register | 0x240(L) / 0x243(B) |
| EP3_DMA_UNIT | Endpoint3 DMA unit counter register | 0x244(L) / 0x247(B) |
| EP3_DMA_FIFO | Endpoint3 DMA FIFO counter register | 0x248(L) / 0x24B(B) |
| EP3_DMA_TTC_L | Endpoint3 DMA transfer counter low-byte register | 0x24C(L) / 0x24F(B) |
| EP3_DMA_TTC_M | Endpoint3 DMA transfer counter middle-byte register | 0x250(L) / 0x253(B) |
| EP3_DMA_TTC_H | Endpoint3 DMA transfer counter high-byte register | 0x254(L) / 0x247(B) |
| EP4_DMA_CON | Endpoint4 DMA control register | 0x258(L) / 0x25B(B) |
| EP4_DMA_UNIT | Endpoint4 DMA unit counter register | 0x25C(L) / 0x25F(B) |
| EP4_DMA_FIFO | Endpoint4 DMA FIFO counter register | 0x260(L) / 0x263(B) |
| EP4_DMA_TTC_L | Endpoint4 DMA transfer counter low-byte register | 0x264(L) / 0x267(B) |
| EP4_DMA_TTC_M | Endpoint4 DMA transfer counter middle-byte register | 0x268(L) / 0x26B(B) |
| EP4_DMA_TTC_H | Endpoint4 DMA transfer counter high-byte register | 0x26C(L) / 0x26F(B) |
| **COMMON INDEXED REGISTERS** | | |
| MAXP_REG | Endpoint MAX packet register | 0x18C(L) / 0x18F(B) |
| **IN INDEXED REGISTERS** | | |
| IN_CSR1_REG/EP0_CSR | EP In control status register 1/EP0 control status register | 0x184(L) / 0x187(B) |
| IN_CSR2_REG | EP In control status register 2 | 0x188(L) / 0x18B(B) |
| **OUT INDEXED REGISTERS** | | |
| OUT_CSR1_REG | EP out control status register 1 | 0x190(L) / 0x193(B) |
| OUT_CSR2_REG | EP out control status register 2 | 0x194(L) / 0x197(B) |
| OUT_FIFO_CNT1_REG | EP out write count register 1 | 0x198(L) / 0x19B(B) |
| OUT_FIFO_CNT2_REG | EP out write count register 2 | 0x19C(L) / 0x19F(B) |

## function 어드레스 레지스터(FUNC_ADDR_REG)

이 레지스터는 호스트에 의해서 할당된 USB 디바이스 컨트롤러를 관리한다. MCU(Micro controller unit)는 이 레지스터에 SET_ADDRESS descriptor를 통해서 수신된 값을 기록한다. 이 어드레스는 다음 발생시에 사용된다.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| FUNC_ADDR_REG | 0x52000140(L)<br>0x52000143(B) | R/W<br>(byte) | Function address register | 0x00 |

| FUNC_ADDR_REG | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| ADDR_UPDATE | [7] | R<br>/SET | R<br>/CLEAR | Set by the MCU whenever it updates the FUNCTION_ADDR field in this register. This bit will be cleared by USB when DATA_END bit in EP0_CSR register. | 0 |
| FUNCTION_ADDR | [6:0] | R/W | R | The MCU write the unique address, assigned by host, to this field. | 00 |

## 파워 관리 레지스터(PWR_REG)

이 레지스터는 USB 블록에서 파워 컨트롤 레지스터로 동작한다.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| PWR_REG | 0x52000144(L)<br>0x52000147(B) | R/W<br>(byte) | Power management register | 0x00 |

| PWR_ADDR | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| ISO_UPDATE | [7] | R/W | R | Used for ISO mode only.<br>If set, GFI waits for a SOF token to set IN_PKT_RDY even though a packet to send is already loaded by MCU. If an IN token is received before a SOF token, then a zero length data packet will be sent. | 0 |
| Reserved | [6:4] | – | – | – | – |
| USB_RESET | [3] | R | SET | Set by the USB if reset signaling is received from the host. This bit remains set as long as reset signaling persists on the bus | 0 |
| MCU_RESUME | [2] | R/W | R<br>/CLEAR | Set by the MCU for MCU Resume.<br>The USB generates the resume signaling during 10ms, if this bit is set in suspend mode. | |
| SUSPEND_MODE | [1] | R | SET<br>/CLEAR | Set by USB automatically when the device enter into suspend mode.<br>It is cleared under the following conditions:<br>1) The MCU clears the MCU_RESUME bit by writing '0', in order to end remote resume signaling.<br>2) The resume signal form host is received. | 0 |
| SUSPEND_EN | [0] | R/W | R | Suspend mode enable control bit<br>0 = Disable (default).<br>The device will not enter suspend mode.<br>1 = Enable suspend mode. | 0 |

## 인터럽트 레지스터(EP_INT_REG/USB_INT_REG)

USB 코어는 2개의 인터럽트 레지스터를 갖는다.
이러한 레지스터는 인터럽트가 걸릴 때 MCU에 대한 상태 레지스터로 동작한다. 설정된 각 비트에 1을 기록해서 클리어한다.

일단 MCU에 인터럽트가 들어오면, MCU는 인터럽트-관련 레지스터의 내용을 읽고 필요하면 내용을 클리어 하기 위해서 back으로 쓰기를 한다.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| EP_INT_REG | 0x52000148(L)<br>0x5200014B(B) | R/W<br>(byte) | EP interrupt pending/clear register | 0x00 |

| EP_INT_REG | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| EP1~EP4 Interrupt | [4:1] | R /CLEAR | SET | **For BULK/INTERRUPT IN endpoints:**<br>Set by the USB under the following conditions:<br>1. IN_PKT_RDY bit is cleared.<br>2. FIFO is flushed<br>3. SENT_STALL set.<br><br>**For BULK/INTERRUPT OUT endpoints:**<br>Set by the USB under the following conditions:<br>1. Sets OUT_PKT_RDY bit<br>2. Sets SENT_STALL bit<br><br>**For ISO IN endpoints:**<br>Set by the USB under the following conditions:<br>1. UNDER_RUN bit is set<br>2. IN_PKT_RDY bit is cleared.<br>3. FIFO is flushed<br>**NOTE:** Conditions 1 and 2 are mutually exclusive<br><br>**For ISO OUT endpoints:**<br>Set by the USB under the following conditions:<br>1. OUT_PKT_RDY bit is set<br>2. OVER RUN bit is set.<br>**NOTE:** Conditions 1 and 2 are mutually exclusive. | 0 |
| EP0 Interrupt | [0] | R /CLEAR | SET | Correspond to endpoint 0 interrupt.<br>Set by the USB under the following conditions:<br>1. OUT_PKT_RDY bit is set.<br>2. IN_PKT_RDY bit is cleared.<br>3. SENT_STALL bit is set<br>4. SETUP_END bit is set<br>5. DATA_END bit is cleared<br>(it indicates the end of control transfer). | 0 |

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| USB_INT_REG | 0x52000158(L)<br>0x5200015B(B) | R/W (byte) | USB interrupt pending/clear register | 0x00 |

| USB_INT_REG | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| RESET Interrupt | [2] | R /CLEAR | SET | Set by the USB when it receives reset signaling. | 0 |
| RESUME Interrupt | [1] | R /CLEAR | SET | Set by the USB when it receives resume signaling, *while in Suspend mode*.<br>If the resume occurs due to a USB reset, then the MCU is first interrupted with a RESUME interrupt. Once the clocks resume and the SE0 condition persists for 3ms, USB RESET interrupt will be asserted. | 0 |
| SUSPEND Interrupt | [0] | R /CLEAR | SET | Set by the USB when it receives suspend signalizing.<br>This bit is set whenever there is no activity for 3ms on the bus. Thus, if the MCU does not stop the clock after the first suspend interrupt, it will continue to be interrupted every 3ms as long as there is no activity on the USB bus.<br>By default, this interrupt is disabled. | 0 |

NOTE : 리셋 인터럽트가 발생되면, 모든 USB 디바이스 레지스터는 재-설정되어야 한다.

## 인터럽트 인에이블 레지스터(EP_INT_EN_REG/USB_INT_REG)

각 인터럽트 레지스터에 대응해서, USB 디바이스 컨트롤러는 2개의 인터럽트 인에이블 레지스터(resume 인터럽트 인에이블을 제외)를 갖는다. 디폴트로, USB 리셋 인터럽트가 인에이블 된다.

비트=0이면, 인터럽트가 디스에이블 된다.
비트=1이면, 인터럽트가 인에이블 된다.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| EP_INT_EN_REG | 0x5200015C(L)<br>0x5200015F(B) | R/W (byte) | Determine which interrupt is enabled | 0xFF |

| EP_INT_EN_REG | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| EP4_INT_EN | [4] | R/W | R | EP4 Interrupt Enable bit<br>0 = Interrupt disable 1 = Enable | 1 |
| EP3_INT_EN | [3] | R/W | R | EP3 Interrupt Enable bit<br>0 = Interrupt disable 1 = Enable | 1 |
| EP2_INT_EN | [2] | R/W | R | EP2 Interrupt Enable bit<br>0 = Interrupt disable 1 = Enable | 1 |
| EP1_INT_EN | [1] | R/W | R | EP1 Interrupt Enable bit<br>0 = Interrupt disable 1 = Enable | 1 |
| EP0_INT_EN | [0] | R/W | R | EP0 Interrupt Enable bit<br>0 = Interrupt disable 1 = Enable | 1 |

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| USB_INT_EN_REG | 0x5200016C(L)<br>0x5200016F(B) | R/W (byte) | Determine which interrupt is enabled | 0x04 |

| INT_MASK_REG | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| RESET_INT_EN | [2] | R/W | R | Reset interrupt enable bit<br>0 = Interrupt disable 1 = Enable | 1 |
| Reserved | [1] | – | – | – | 0 |
| SUSPEND_INT_EN | [0] | R/W | R | Suspend interrupt enable bit<br>0 = Interrupt disable 1 = Enable | 0 |

## 프레임 number 레지스터(FRAME_NUM1_REG/FRAME_NUM2_REG)

호스트가 USB 패킷을 전송하면, 각 프레임의 시작(SOF) 패킷은 프레임 number를 포함한다. USB 디바이스 컨트롤러는 이러한 프레임 number를 캐치하며 이 레지스터에 자동적으로 호출한다.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| FRAME_NUM1_REG | 0x52000170(L)<br>0x52000173(B) | R<br>(byte) | Frame number lower byte register | 0x00 |

| FRAME_NUM_REG | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| FRAME_NUM1 | [7:0] | R | W | Frame number lower byte value | 00 |

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| FRAME_NUM2_REG | 0x52000174(L)<br>0x52000177(B) | R<br>(byte) | Frame number higher byte register | 0x00 |

| FRAME_NUM_REG | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| FRAME_NUM2 | [7:0] | R | W | Frame number higher byte value | 00 |

## 인덱스 레지스터(INDEX_REG)

인덱스 레지스터는 어떤 엔드포인트 레지스터를 효율적으로 지시하는데 사용된다. MCU는 인덱스 레지스터를 이용해서 코어 안의 엔드포인드에 대해서 엔드포인트 레지스터에 접근할 수 있다.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| INDEX_REG | 0x52000178(L)<br>0x5200017B(B) | R/W<br>(byte) | Register index register | 0x00 |

| INDEX_REG | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| INDEX | [7:0] | R/W | R | Indicate a certain endpoint | 00 |

## 엔드 포인트0 컨트롤 상태 레지스터(EP0_CSR)

이 레지스터는 엔드포인트0에 대한 컨트롤과 상태 비트를 가지고 있다. 컨트롤 트랜잭션

이 IN 토큰과 OUT 토큰에 포함되기 때문에, IN CSR1 레지스터에 맵핑되는 1개의 CSR 레지스터가 있다.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| EP0_CSR | 0x52000184(L)<br>0x52000187(B) | R/W<br>(byte) | Endpoint 0 status register | 0x00 |

| EP0_CSR | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| SERVICED_SETUP_END | [7] | W | CLEAR | The MCU should write a "1" to this bit to clear SETUP_END. | 0 |
| SERVICED_OUT_PKT_RDY | [6] | W | CLEAR | The MCU should write a "1" to this bit to clear OUT_PKT_RDY. | 0 |
| SEND_STALL | [5] | R/W | CLEAR | MCU should write a "1" to this bit at the same time it clears OUT_PKT_RDY, if it decodes an invalid token.<br>0 = Finish the STALL condition<br>1 = The USB issues a STALL and shake to the current control transfer. | 0 |
| SETUP_END | [4] | R | SET | Set by the USB when a control transfer ends before DATA_END is set.<br>When the USB sets this bit, an interrupt is generated to the MCU.<br>When such a condition occurs, the USB flushes the FIFO and invalidates MCU access to the FIFO. | 0 |
| DATA_END | [3] | SET | CLEAR | Set by the MCU on the conditions below:<br>1. After loading the last packet of data into the FIFO, at the same time IN_PKT_RDY is set.<br>2. While it clears OUT_PKT_RDY after unloading the last packet of data.<br>3. For a zero length data phase. | 0 |
| SENT_STALL | [2] | CLEAR | SET | Set by the USB if a control transaction is stopped due to a protocol violation. An interrupt is generated when this bit is set. The MCU should write "0" to clear this bit. | 0 |

| EP0_CSR | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| IN_PKT_RDY | [1] | SET | CLEAR | Set by the MCU after writing a packet of data into EP0 FIFO. The USB clears this bit once the packet has been successfully sent to the host. An interrupt is generated when the USB clears this bit, so as the MCU to load the next packet. For a zero length data phase, the MCU sets DATA_END at the same time. | 0 |
| OUT_PKT_RDY | [0] | R | SET | Set by the USB once a valid token is written to the FIFO. An interrupt is generated when the USB sets this bit. The MCU clears this bit by writing a "1" to the SERVICED_OUT_PKT_RDY bit. | 0 |

## 엔드 포인트 IN 컨트롤 상태 레지스터(IN_CSR1_REG/IN_CSR2_REG)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| IN_CSR1_REG | 0x52000184(L)<br>0x52000187(B) | R/W<br>(byte) | IN END POINT control status register1 | 0x00 |

| IN_CSR1_REG | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| Reserved | [7] | – | – | – | 0 |
| CLR_DATA_<br>TOGGLE | [6] | R/W | R/<br>CLEAR | Used in Set-up procedure.<br>0: There are alternation of DATA0 and DATA1<br>1: The data toggle bit is cleared and PID in packet will maintain DATA0 | 0 |
| SENT_STALL | [5] | R/<br>CLEAR | SET | Set by the USB when an IN token issues a STALL handshake, after the MCU sets SEND_STALL bit to start STALL handshaking. When the USB issues a STALL handshake, IN_PKT_RDY is cleared | 0 |
| SEND_STALL | [4] | W/R | R | 0: The MCU clears this bit to finish the STALL condition.<br>1: The MCU issues a STALL handshake to the USB. | 0 |
| FIFO_FLUSH | [3] | R/W | CLEAR | Set by the MCU if it intends to flush the packet in Input-related FIFO. This bit is cleared by the USB when the FIFO is flushed. The MCU is interrupted when this happens. If a token is in process, the USB waits until the transmission is complete before FIFO flushing. If two packets are loaded into the FIFO, only first packet (The packet is intended to be sent to the host) is flushed, and the corresponding IN_PKT_RDY bit is cleared | 0 |
| UNDER_RUN | [2] | R/<br>CLEAR | Set | *Valid only For Iso mode.*<br>Set by the USB when in ISO mode, an IN token is received and the IN_PKT_RDY bit is not set.<br>The USB sends a zero length data packet for such conditions, and the next packet that is loaded into the FIFO is flushed.<br>This bit is cleared by writing 0. | 0 |
| Reserved | [1] | – | – | – | 0 |

| IN_CSR1_REG | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| IN_PKT_RDY | [0] | R/SET | CLEAR | Set by the MCU after writing a packet of data into the FIFO.<br>The USB clears this bit once the packet has been successfully sent to the host.<br>An interrupt is generated when the USB clears this bit, so the MCU can load the next packet. While this bit is set, the MCU will not be able to write to the FIFO.<br>If the MCU sets SEND STALL bit, this bit cannot be set. | 0 |

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| IN_CSR2_REG | 0x52000188(L)<br>0x5200018B(B) | R/W<br>(byte) | IN END POINT control status register2 | 0x20 |

| IN_CSR2_REG | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| AUTO_SET | [7] | R/W | R | If set, whenever the MCU writes MAXP data, IN_PKT_RDY will automatically be set by the core without any intervention from MCU.<br>If the MCU writes less than MAXP data, IN_PKT_RDY bit has to be set by the MCU. | 0 |
| ISO | [6] | R/W | R | *Used only for endpoints whose transfer type is programmable.*<br>1: Reserved<br>0: Configures endpoint to Bulk mode | 0 |
| MODE_IN | [5] | R/W | R | *Used only for endpoints whose direction is programmable.*<br>1: Configures Endpoint Direction as IN<br>0: Configures Endpoint Direction as OUT | 1 |
| IN_DMA_INT_EN | [4] | R/W | R | Determine whether the interrupt should be issued or not, when the EP1 IN_PKT_RDY condition happens. This is only useful for DMA mode.<br>0 = Interrupt enable, 1 = Interrupt Disable | 0 |
| Reserved | [3:0] | – | – | – | – |

## 엔드 포인트 OUT 컨트롤 상태 레지스터(OUT_CSR1_REG/OUT_CSR2_REG)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| OUT_CSR1_REG | 0x52000190(L)<br>0x52000193(B) | R/W<br>(byte) | End Point out control status register1 | 0x00 |

| OUT_CSR1_REG | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| CLR_DATA_ TOGGLE | [7] | R/W | CLEAR | When the MCU writes a 1 to this bit, the data toggle sequence bit is reset to DATA0. | 0 |
| SENT_STALL | [6] | R/ CLEAR | SET | Set by the USB when an OUT token is ended with a STALL handshake. The USB issues a stall handshake to the host if it sends more than MAXP data for the OUT TOKEN. | 0 |
| SEND_STALL | [5] | R/W | R | 0: The MCU clears this bit to end the STALL condition handshake, IN PKT RDY is cleared.<br>1: The MCU issues a STALL handshake to the USB. The MCU clears this bit to end the STALL condition handshake, IN PKT RDY is cleared. | 0 |
| FIFO_FLUSH | [4] | R/W | CLEAR | The MCU writes a 1 to flush the FIFO.<br>This bit can be set only when OUT_PKT_RDY (D0) is set. The packet due to be unloaded by the MCU will be flushed. | 0 |
| DATA_ERROR | [3] | R | R/W | *Valid only in ISO mode.*<br>This bit should be sampled with OUT_PKT_RDY.<br>When set, it indicates the data packet due to be unloaded by the MCU has an error (either bit stuffing or CRC). If two packets are loaded into the FIFO, and the second packet has an error, then this bit gets set only after the first packet is unloaded. This bit is automatically cleared when OUT_PKT_RDY gets cleared. | 0 |
| OVER_RUN | [2] | R/ CLEAR | R/W | *Valid only in ISO mode.*<br>This bit is set if the core is not able to load an OUT ISO token into the FIFO.<br>The MCU clears this bit by writing 0. | 0 |
| Reserved | [1] | – | – | – | 0 |
| OUT_PKT_RDY | [0] | R/ CLEAR | SET | Set by the USB after it has loaded a packet of data into the FIFO. Once the MCU reads the packet from FIFO, this bit should be cleared by MCU (write a "0"). | 0 |

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| OUT_CSR2_REG | 0x52000194(L)<br>0x52000197(B) | R/W (byte) | End Point out control status register2 | 0x00 |

| OUT_CSR2_REG | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| AUTO_CLR | [7] | R/W | R | If the MCU is set, whenever the MCU reads data from the OUT FIFO, OUT_PKT_RDY will automatically be cleared by the logic without any intervention from the MCU. | 0 |
| ISO | [6] | R/W | R | Determine endpoint transfer type.<br>0: Configures endpoint to Bulk mode.<br>1: Reserved | 0 |
| OUT_DMA_INT_ MASK | [5] | R/W | R | Determine whether the interrupt should be issued or not.<br>OUT_PKT_RDY condition happens. This is only useful for DMA mode<br>0 = Interrupt Enable 1 = Interrupt Disable | 0 |

## 엔드 포인트 FIFO 레지스터(EPN_FIFO_REG)

EPn_FIFO_REG는 EPn FIFO에 접근하기 위해서 MCU를 인에이블 한다.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| EP0_FIFO | 0x520001C0(L)<br>0x520001C3 (B) | R/W<br>(byte) | End Point0 FIFO register | 0xXX |
| EP1_FIFO | 0x520001C4(L)<br>0x520001C7(B) | R/W<br>(byte) | End Point1 FIFO register | 0xXX |
| EP2_FIFO | 0x520001C8(L)<br>0x520001CB(B) | R/W<br>(byte) | End Point2 FIFO register | 0xXX |
| EP3_FIFO | 0x520001CC(L)<br>0x520001CF(B) | R/W<br>(byte) | End Point3 FIFO register | 0xXX |
| EP4_FIFO | 0x520001D0(L)<br>0x520001D3(B) | R/W<br>(byte) | End Point4 FIFO register | 0xXX |

| EPn_FIFO | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| FIFO_DATA | [7:0] | R/W | R/W | FIFO data value | 0xXX |

## Max 패킷 레지스터(MAXP_REG)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| MAXP_REG | 0x5200018C(L)<br>0x5200018F(B) | R/W<br>(byte) | End Point MAX packet register | 0x01 |

| MAXP_REG | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| MAXP | [3:0] | R/W | R | 0000: Reserved<br>0001: MAXP = 8 Byte0010: MAXP = 16 Byte<br>0100: MAXP = 32 Byte1000: MAXP = 64 Byte<br>For EP0, MAXP=8 is recommended.<br>For EP1~4, MAXP=32 or MAXP=64 is recommended. And, if MAXP=32, the dual packet mode will be enabled automatically. | 0001 |

## 엔드 포인트 OUT 쓰기 카운트 레지스터

## (OUT_FIFO_CNT1_REG/OUT_FIFO_CNT2_REG)

이 레지스터는 number가 MCU에 의해서 호출되지 않을 때 패킷의 바이트 수를 관리한다.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| OUT_FIFO_CNT1_ REG | 0x52000198(L) 0x5200019B(B) | R (byte) | End Point out write count register1 | 0x00 |

| OUT_FIFO_CNT1_REG | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| OUT_CNT_LOW | [7:0] | R | W | Lower byte of write count | 0x00 |

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| OUT_FIFO_CNT2_REG | 0x5200019C(L) 0x5200019F(B) | R (byte) | End Point out write count register2 | 0x00 |

| OUT_FIFO_CNT2_REG | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| OUT_CNT_HIGH | [7:0] | R | W | Higher byte of write count. The OUT_CNT_HIGH may be always 0 normally. | 0x00 |

## DMA 인터페이스 컨트롤 레지스터(EPN_DMA_CON)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| EP1_DMA_CON | 0x52000200(L) 0x52000203(B) | R/W (byte) | EP1 DMA interface control register | 0x00 |
| EP2_DMA_CON | 0x52000218(L) 0x5200021B(B) | R/W (byte) | EP2 DMA interface control register | 0x00 |
| EP3_DMA_CON | 0x52000240(L) 0x52000243(B) | R/W (byte) | EP3 DMA interface control register | 0x00 |
| EP4_DMA_CON | 0x52000258(L) 0x5200025B(B) | R/W (byte) | EP4 DMA interface control register | 0x00 |

| EPn_DMA_CON | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| IN_RUN_OB | [7] | R/W | W | Read) IN_DMA_Run Observation<br>0: DMA is stopped 1:DMA is running<br>Write) Ignore EPn_DMA_TTC_n register<br>0: DMA requests will be stopped if EPn_DMA_TTC_n reaches 0.<br>1: DMA requests will be continued although EPn_DMA_TTC_n reaches 0. | 0 |
| STATE | [6:4] | R | W | DMA State Monitoring | 0 |
| DEMAND_MODE | [3] | R/W | R | DMA Demand mode enable bit<br>0: Demand mode disable<br>1: Demand mode enable | 0 |
| OUT_RUN_OB/ OUT_DMA_RUN | [2] | R/W | R/W | Functionally separated into write and read operation.<br>Write operation: '0' = Stop '1' = Run<br>Read operation: OUT DMA Run Observation | 0 |
| IN_DMA_RUN | [1] | R/W | R | Start DMA operation.<br>0 = Stop 1 = Run | 0 |
| DMA_MODE_EN | [0] | R/W | R/ CLEAR | Set DMA mode.If the IN_RUN_OB has been wrtten as 0 and EPn_DMA_TTC_n reaches 0, DMA_MODE_EN bit will be cleared by USB.<br>0 = Interrupt mode 1 = DMA mode | 0 |

## DMA 유닛 카운터 레지스터(EPN_DMA_UNIT)

이 레지스터는 Demand 모드에서 유효한다. 다른 모드에서, 레지스터 값은 '0x01'로 설정되어야 한다.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| EP1_DMA_UNIT | 0x52000204(L)<br>0x52000207(B) | R/W (byte) | EP1 DMA transfer unit counter base register | 0x00 |
| EP2_DMA_UNIT | 0x5200021C(L)<br>0x5200021F(B) | R/W (byte) | EP2 DMA transfer unit counter base register | 0x00 |
| EP3_DMA_UNIT | 0x52000244(L)<br>0x52000247(B) | R/W (byte) | EP3 DMA transfer unit counter base register | 0x00 |
| EP4_DMA_UNIT | 0x5200025C(L)<br>0x5200025F(B) | R/W (byte) | EP4 DMA transfer unit counter base register | 0x00 |

| DMA_UNIT | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| EPn_UNIT_CNT | [7:0] | R/W | R | EP DMA transfer unit counter value | 0x00 |

## DMA FIFO 카운터 레지스터(EPN_DMA_FIFO)

이 레지스터는 DMA에 의해서 전송되는 FIFO에 바이트 크기의 값을 갖는다. OUT_DMA_RUN이 인에이블 되는 경우에, OUT FIFO 쓰기 카운트 레지스터1의 값은 이 레지스터에 자동적으로 로딩된다. IN DMA 모드의 경우에, MCU는 소프트웨어에 의해서 적정한 값으로 설정되어야 한다.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| EP1_DMA_FIFO | 0x52000208(L)<br>0x5200020B(B) | R/W<br>(byte) | EP1 DMA transfer FIFO counter base register | 0x00 |
| EP2_DMA_FIFO | 0x52000220(L)<br>0x52000223(B) | R/W<br>(byte) | EP2 DMA transfer FIFO counter base register | 0x00 |
| EP3_DMA_FIFO | 0x52000248(L)<br>0x5200024B(B) | R/W<br>(byte) | EP3 DMA transfer FIFO counter base register | 0x00 |
| EP4_DMA_FIFO | 0x52000260(L)<br>0x52000263(B) | R/W<br>(byte) | EP4 DMA transfer FIFO counter base register | 0x00 |

| DMA_FIFO | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| EPn_FIFO_CNT | [7:0] | R/W | R | EP DMA transfer FIFO counter value | 0x00 |

## DMA 전체 전송 카운터 레지스터(EPN_DMA_TTC_L,M,H)

이 레지스터는 DMA를 이용해서 전송할 전체 number를 가져야 한다.(총 20비트 카운터)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| EP1_DMA_TTC_L | 0x5200020C(L)<br>0x5200020F(B) | R/W<br>(byte) | EP1 DMA total transfer counter(lower byte) | 0x00 |
| EP1_DMA_TTC_M | 0x52000210(L)<br>0x52000213(B) | R/W<br>(byte) | EP1 DMA total transfer counter(middle byte) | 0x00 |
| EP1_DMA_TTC_H | 0x52000214(L)<br>0x52000217(B) | R/W<br>(byte) | EP1 DMA total transfer counter(higher byte) | 0x00 |
| EP2_DMA_TTC_L | 0x52000224(L)<br>0x52000227(B) | R/W<br>(byte) | EP2 DMA total transfer counter(lower byte) | 0x00 |
| EP2_DMA_TTC_M | 0x52000228(L)<br>0x5200022B(B) | R/W<br>(byte) | EP2 DMA total transfer counter(middle byte) | 0x00 |
| EP2_DMA_TTC_H | 0x5200022C(L)<br>0x5200022F(B) | R/W<br>(byte) | EP2 DMA total transfer counter(higher byte) | 0x00 |
| EP3_DMA_TTC_L | 0x5200024C(L)<br>0x5200024F(B) | R/W<br>(byte) | EP3 DMA total transfer counter(lower byte) | 0x00 |
| EP3_DMA_TTC_M | 0x52000250(L)<br>0x52000253(B) | R/W<br>(byte) | EP3 DMA total transfer counter(middle byte) | 0x00 |
| EP3_DMA_TTC_H | 0x52000254(L)<br>0x52000257(B) | R/W<br>(byte) | EP3 DMA total transfer counter(higher byte) | 0x00 |
| EP4_DMA_TTC_L | 0x52000264(L)<br>0x52000267(B) | R/W<br>(byte) | EP4 DMA total transfer counter(lower byte) | 0x00 |
| EP4_DMA_TTC_M | 0x52000268(L)<br>0x5200026B(B) | R/W<br>(byte) | EP4 DMA total transfer counter(middle byte) | 0x00 |
| EP4_DMA_TTC_H | 0x5200026C(L)<br>0x5200026F(B) | R/W<br>(byte) | EP4 DMA total transfer counter(higher byte) | 0x00 |

| DMA_TX | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| EPn_TTC_L | [7:0] | R/W | R | DMA total transfer count value (lower byte) | 0x00 |
| EPn_TTC_M | [7:0] | R/W | R | DMA total transfer count value (middle byte) | 0x00 |
| EPn_TTC_H | [3:0] | R/W | R | DMA total transfer count value (higher byte) | 0x00 |