

Multi-Buffered Serial Peripheral Interface Module (MibSPI) with Parallel Pin Option (MibSPIP)

This chapter provides the specifications for a 16-bit configurable synchronous multi-buffered multi-pin serial peripheral interface (MibSPI). This chapter also provides the specifications for MibSPI with Parallel Pin Option (MibSPIP). The MibSPI is a programmable-length shift register used for high-speed communication between external peripherals or other microcontrollers.

이 장에서는 16 비트 구성 가능 동기식 멀티 버퍼 멀티 핀 직렬 주변 장치 인터페이스 (MibSPI)에 대한 사양을 제공합니다.

또한 이 장에서는 병렬 핀 옵션 (MibSPIP)이 있는 MibSPI의 사양을 제공합니다.

MibSPI는 외부 주변 장치 또는 다른 마이크로 컨트롤러 간의 고속 통신에 사용되는 프로그래머블 길이 시프트 레지스터입니다.

Throughout this chapter, all references to SPI also apply to MibSPI/MibSPIP, unless otherwise noted.

이 장에서 SPI에 대한 모든 언급은 별도의 언급이 없는 한 MibSPI / MibSPIP에도 적용됩니다.

NOTE: This chapter describes a superset implementation of the MibSPI/SPI modules that includes features and functionality that may not be available on some devices.

Device-specific content that should be determined by referencing the datasheet includes DMA functionality, MibSPI RAM size, number of transfer groups, number of chip selects, parallel mode support, and availability of 5-pin operation (SPInENA).

NOTE : 이 장에서는 일부 장치에서 사용할 수 없는 기능을 포함하는 MibSPI / SPI 모듈의 수퍼셋 구현에 대해 설명합니다.

데이터 시트를 참조하여 결정되어야 하는 장치 별 콘텐츠에는 DMA 기능, MibSPI RAM 크기, 전송 그룹 수, 칩 선택 수, 병렬 모드 지원 및 5 핀 작동 가용성 (SPINENA)이 포함됩니다.

Topic	Page
27.1 Overview.....	1224
27.2 Basic Operation.....	1226
27.3 Control Registers.....	1261
27.4 Multi-buffer RAM.....	1327
27.5 Parity/ECC Memory	1333
27.6 MibSPI Pin Timing Parameters	1338

27.1 Overview(개요)

27.1.1 Features(기능)

The MibSPI/SPI is a high-speed synchronous serial input/output port that allows a serial bit stream of programmed length (two to 16 bits) to be shifted into and out of the device at a programmed bit-transfer rate. The MibSPI/SPI is normally used for communication between the microcontroller and external peripherals or another microcontroller. Typical applications include interface to external I/O or peripheral expansion via devices such as shift registers, display drivers, and analog-to-digital converters. MibSPI is an Extension of SPI. MibSPI works in 2 Modes.

MibSPI / SPI는 프로그래밍 된 비트 전송률로 프로그램 된 길이 (2 ~ 16 비트)의 직렬 비트 스트림을 장치로 또는 장치 밖으로 이동시키는 고속 동기 직렬 입 / 출력 포트입니다. MibSPI / SPI는 일반적으로 마이크로 컨트롤러와 외부 주변 장치 또는 다른 마이크로 컨트롤러 간의 통신에 사용됩니다. 일반적인 애플리케이션으로는 외부 I / O에 대한 인터페이스 또는 시프트 레지스터, 디스플레이 드라이버 및 아날로그 - 디지털 변환기와 같은 장치를 통한 주변 장치 확장이 포함됩니다. MibSPI는 SPI의 확장입니다. MibSPI는 2 가지 모드로 작동합니다.

- Compatibility Mode (호환 모드)
- Multi-buffer Mode (다중 버퍼 모드)

The Compatibility mode of MibSPI makes it behave exactly like that of SPI and ensures full compatibility with the same. Everything described about compatibility mode of MibSPI , in this document, is directly applicable to SPI.

MibSPI의 호환성 모드는 SPI와 완전히 동일하게 동작하며 동일한 호환성을 보장합니다. 이 문서에서 MibSPI의 호환 모드에 대해 설명한 모든 것은 SPI에 직접 적용됩니다.

The MibSPI supports memory fault detection/correction via internal Parity/ECC circuit. MibSPI is configurable to include or not include Memory Parity/ECC logic during circuit synthesis.

MibSPI는 내부 패리티 / ECC 회로를 통해 메모리 오류 감지 / 수정을 지원합니다. MibSPI는 회로 합성 중 메모리 패리티 / ECC 로직을 포함하거나 포함하지 않도록 구성 할 수 있습니다.

The SPI / MibSPI can be configured in three pin, four pin or five pin mode of operation. The SPI / MibSPI allows multiple programmable chip-selects.

SPI / MibSPI는 3 핀, 4 핀 또는 5 핀 동작 모드로 구성 할 수 있습니다. SPI / MibSPI는 다중 프로그래머블 칩 선택을 허용한다.

The MibSPI has a programmable Multi-buffer array that enables programmed transmission to be completed without CPU intervention. The buffers are combined in different transfer groups that could be triggered by external events (Timers, I/O, and so on) or by the internal tick counter. The internal tick counter can support periodic trigger events. Each buffer of the MibSPI can be associated with different DMA channels in different transfer group, allowing the user to move data from/to internal memory to/from external slave with a minimal CPU interaction.

MibSPI에는 프로그래밍 된 다중 버퍼 어레이가있어 CPU 전송없이 프로그래밍 된 전송을 완료 할 수 있습니다. 버퍼는 외부 이벤트 (타이머, I / O 등) 또는 내부 틱 카운터에 의해 트리거 될 수있는 여러 전송 그룹으로 결합됩니다. 내부 틱 카운터는주기적인 트리거 이벤트를 지원할 수 있습니다. MibSPI의 각 버퍼는 서로 다른 전송 그룹의 서로 다른 DMA 채널과 연관 될 수 있으므로 사용자는 최소한의 CPU 상호 작용으로 내부 슬레이브에서 내부 슬레이브로 / 슬레이브로 데이터를 이동할 수 있습니다.

The pins SPICLK, SPISIMO and SPISOMI are used in all MibSPI pin modes. The pins SPIENA and SPISCS[7:0] are optional and may be used if the pin are present on a given device.

SPICLK, SPISIMO 및 SPISOMI 핀은 모든 MibSPI 핀 모드에서 사용됩니다. SPIENA와 SPISCS [7 : 0] 핀은 옵션이며 핀이 주어진 장치에 존재할 경우 사용될 수 있습니다.

The SPI has the following attributes:

SPI에는 다음과 같은 특성이 있습니다.

- 16-bit shift register
 - Receive buffer register
 - 8-bit baud clock generator
 - Serial Clock (SPICLK) I/O pin
 - Up to 8 Slave out, Master in (SPISOMI) I/O pins for faster data transfers
 - SPI enable (SPIENA) pin (4 or 5-pin mode only)
 - Up to 8 slave chip select (Slave Chip Select SPISCS[7:0]) pin (4 or 5-pin mode only).
 - pins (GPIOs)
-
- 16 비트 시프트 레지스터
 - 수신 버퍼 레지스터
 - 8 비트 보드 클럭 생성기
 - 직렬 클럭 (SPICLK) I / O 핀
 - 빠른 데이터 전송을 위해 최대 8 개의 슬레이브 출력, 마스터 인 (SPISOMI) I / O 핀
 - SPI 인 에이블 (SPIENA) 핀 (4 또는 5 핀 모드 만 해당)
 - 최대 8 개의 슬레이브 칩 선택 (슬레이브 칩 선택 SPISCS [7 : 0]) 핀 (4 또는 5 핀 모드 전용).
 - 핀 (GPIO)

The SPI / MibSPI allows software to program the following options:

SPI / MibSPI를 사용하면 소프트웨어가 다음 옵션을 프로그래밍 할 수 있습니다.

- SPISOMI/SPISIMO pin direction configuration
 - SPICLK pin source (external/internal)
 - MibSPI pins as functional or digital I/O pins. For each Buffer, following features can be selected from 4 different combinations of Formats using the control fields in the buffer.
-
- SPISOMI / SPISIMO 핀 방향 구성
 - SPICLK 핀 소스 (외부 / 내부)
 - MibSPI 핀을 기능 또는 디지털 I / O 핀으로 사용합니다. 각 버퍼에 대해 버퍼의 제어 필드를 사용하여 4 가지 형식 조합 중에서 다음 기능을 선택할 수 있습니다.
-
- SPICLK frequency
 - Charecter length
 - Phase, Polarity
 - Enable/Disable parity for transmit and receive
 - Enable/Disable timers for ChipSelect Hold and Setup timers
 - Direction of shifting, MSBit first or LSBit first
 - Configurable Parallel modes to use multiple SIMO/SOMI pin
 - Configurable number of ChipSelects In Multi-buffer Mode, in addition to the above, many other features are configurable
 - Number of buffers for each peripheral(or data source/destination, up to 256 buffers supported) or group(up to 8 groupings)
 - Number of DMA controlled buffers & number of DMA request channels(up to 8 for each of transmit & receive)
 - Triggers for each groups, trigger types, trigger sources for individual groups(up to 14 external trigger sources & 1 internal trigger source)
 - Number of DMA transfers for each buffer(up to 65536 for up to 8 buffers)
 - Un-interrupted DMA buffer transfer(NOBREAK buffer).
-
- SPICLK 주파수
 - Charecter 길이
 - 위상, 극성
 - 전송 및 수신을 위한 패리티 사용 / 사용 안 함
 - ChipSelect Hold 및 Setup 타이머의 타이머 활성화 / 비활성화
 - 이동 방향, MSBit 먼저 또는 LSBit 먼저
 - 다중 SIMO / SOMI 핀을 사용하도록 구성 가능한 병렬 모드
 - ChipSelects의 구성 가능한 수 다중 버퍼 모드에서 위 이외에도 많은 다른 기능을 구성 할 수 있습니다
 - 각 주변 장치 (또는 데이터 소스 / 대상, 최대 256 개의 버퍼 지원) 또는 그룹 (최대 8 개의 그룹)에 대한 버퍼 수
 - DMA 제어 버퍼 수 및 DMA 요청 채널 수 (전송 및 수신마다 최대 8 개)
 - 각 그룹에 대한 트리거, 트리거 유형, 개별 그룹에 대한 트리거 소스 (최대 14 개의 외부 트리거 소스 및 1 개의 내부 트리거 소스)

- 각 버퍼에 대한 DMA 전송 수 (최대 8 버퍼의 경우 최대 65536)
- 중단되지 않은 DMA 버퍼 전송 (NOBREAK 버퍼).

NOTE: SIMO - Slave In Master Out Pin
 SOMI - Slave Out Master In Pin
 CS - SPI Chip Select Pin
 ENA - SPI Enable Pin

27.1.2 Pin Configurations(핀 구성)

The SPI supports data connections as shown in Table 27-1.

SPI는 표 27-1에 나와있는 것처럼 데이터 연결을 지원합니다.

Table 27-1. Pin Configurations

Pin	Master Mode		Slave Mode	
SPICLK	Drives the clock to external devices		Receives the clock from the external master	
SPISOMI	Receives data from the external slave		Sends data to the external master	
SPISIMO	Transmits data to the external slave		Receives data from the external master	
SPIENA	SPIENA disabled: GIO	SPIENA enabled: Receives ENA signal from the external slave	SPIENA disabled: GIO	SPIENA enabled: Receives ENA signal from the external master
SPICS[7:0]	SPICS disabled: GIO	SPICS enabled: Selects one or more slave devices	SPICS disabled: GIO	SPICS enabled: Receives the CS signal from the external master

- NOTE:** 1. When the SPICS[3:0] signals are disabled, the chip-select field in the transmit data is not used.
2. When the SPIENA signal is disabled, the SPIENA pin is ignored in master mode, and not driven as part of the SPI transaction in slave mode.

- 노트:** 1. SPICS [3 : 0] 신호가 디스 에이블되면, 전송 데이터의 칩 선택 필드는 사용되지 않는다.
2. SPIENA 신호가 비활성화되면 SPIENA 핀은 마스터 모드에서 무시되고 슬레이브 모드에서 SPI 트랜잭션의 일부로 구동되지 않습니다.

27.1.3 MibSPI /SPI Configurations (구성)

Table 27-2. MibSPI/SPI Configurations

MibSPiX/SPIx	I/Os
MibSPI1	MIBSPI1SIMO[1:0], MIBSPI1SOMI[1:0], MIBSPI1CLK, MIBSPI1nCS[5:0], MIBSPI1nENA
MibSPI2	MIBSPI2SIMO[1:0], MIBSPI2SOMI[1:0], MIBSPI2CLK, MIBSPI2nCS[5:0], MIBSPI2nENA
MibSPI3	MIBSPI3SIMO[1:0], MIBSPI3SOMI[1:0], MIBSPI3CLK, MIBSPI3nCS[5:0], MIBSPI3nENA
MibSPI4	MIBSPI4SIMO[1:0], MIBSPI4SOMI[1:0], MIBSPI4CLK, MIBSPI4nCS[5:0], MIBSPI4nENA
MibSPI5	MIBSPI5SIMO[1:0], MIBSPI5SOMI[1:0], MIBSPI51CLK, MIBSPI5nCS[5:0], MIBSPI5nENA
SPI1	SPI1SIMO, ZSPI1SOMI, SPI1CLK, SPI2nCS[1:0], SPI1nENA
SPI2	SPI2SIMO, ZSPI2SOMI, SPI2CLK, SPI2nCS[1:0], SPI2nENA
SPI3	SPI3SIMO, ZSPI3SOMI, SPI3CLK, SPI3nCS[1:0], SPI3nENA

27.2 Basic Operation (기본 조작)

This section details the basic operation principle of the SPI mode and the MibSPI mode operation of the device.

이 섹션에서는 SPI 모드의 기본적인 작동 원리와 장치의 MibSPI 모드 작동에 대해 자세히 설명합니다.

27.2.1 SPI Mode

The SPI can be configured via software to operate as either a master or a slave. The MASTER bit (SPIGCR1[0]) selects the configuration of the SPISIMO and SPISOMI pins. CLKMOD bit (SPIGCR1[1]) determines whether an internal or external clock source will be used.

SPI는 소프트웨어를 통해 마스터 또는 슬레이브로 작동하도록 구성 할 수 있습니다. MASTER 비트 (SPIGCR1 [0])는 SPISIMO 및 SPISOMI 핀의 구성을 선택합니다. CLKMOD 비트 (SPIGCR1 [1])는 내부 또는 외부 클럭 소스 사용 여부를 결정합니다.

The slave chip select (SPISCS[7:0]) pins, are used when communicating with multiple slave devices. When the a write occurs to SPIDAT1 in master mode, the SPISCS pins are automatically driven to select the specified slave.

슬레이브 칩 선택 (SPISCS [7 : 0]) 핀은 여러 슬레이브 장치와 통신 할 때 사용됩니다. 마스터 모드에서 SPIDAT1에 쓰기가 발생하면 SPISCS 핀이 자동으로 구동되어 지정된 슬레이브를 선택합니다.

Handshaking mechanism, provided by the SPIENA pin, enables a slave SPI to delay the generation of the clock signal supplied by the master if it is not prepared for the next exchange of data.

SPIENA 핀에 의해 제공되는 핸드 셰이킹 메커니즘은 슬레이브 SPI가 마스터가 제공하는 클럭 신호의 생성을 지연시켜 다음 엑 쿼지 데이터를 준비하지 못하게한다.

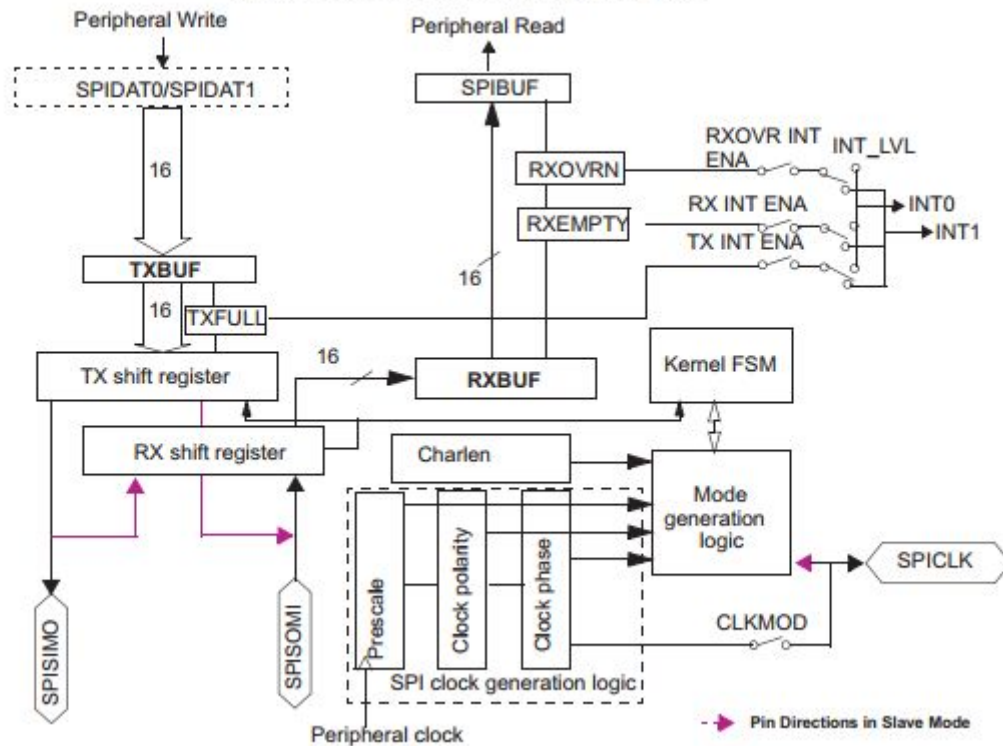
27.2.1.1 SPI Mode Operation Block Diagram

Figure 27-1 shows the SPI transaction hardware.

TXBUF and RXBUF are internal buffers that are intended to improve the overall throughput of data transfer. TXBUF is a transmit buffer, while RXBUF is a receive buffer.

그림 27-1은 SPI 트랜잭션 하드웨어를 보여줍니다. TXBUF 및 RXBUF는 데이터 전송의 전반적인 처리량을 향상시키기위한 내부 버퍼입니다. TXBUF는 송신 버퍼이고, RXBUF는 수신 버퍼이다

Figure 27-1. SPI Functional Logic Diagram



- 1 This is a representative diagram, which shows three-pin mode hardware.
- 2 TXBUF, RXBUF, and SHIFT_REGISTER are user-invisible registers.
- 3 SPIDAT0 and SPIDAT1 are user-visible, and are physically mapped to the contents of TXBUF.
- 4 SPISIMO, SPISOMI, SPICLK pin directions depend on the Master or Slave Mode.

- 1 이것은 3 핀 모드 하드웨어를 나타내는 대표적인 다이어그램입니다.
- 2 TXBUF, RXBUF 및 SHIFT_REGISTER는 사용자가 볼 수 없는 레지스터입니다.
- 3 SPIDAT0 및 SPIDAT1은 사용자에게 표시되며 물리적으로 TXBUF의 내용에 매핑됩니다.
- 4 SPISIMO, SPISOMI, SPICLK 핀 방향은 마스터 또는 슬레이브 모드에 따라 다릅니다.

27.2.1.2 Data Flow and Handling for TX and RX.(데이터 흐름 및 처리)

27.2.1.2.1 Data Sequencing when SPIDAT0 or SPIDAT1 is Written(기록될 때 데이터 시퀀싱)

- If both the TX shift register and TXBUF are empty, then the data is directly copied to the TX shift register.

For devices with DMA, if DMA is enabled, a transmit DMA request (TX_DMA_REQ) is generated to cause the next word to be fetched. If transmit interrupts are enabled, a transmitter-empty interrupt is generated.

- If the TX shift register is already full or is in the process of shifting and if TXBUF is empty then the data written to SPIDAT0 / SPIDAT1 is copied to TXBUF and TXFULL flag is set to 1 at the same time.
- When a shift operation is complete, data from the TXBUF (if it is full) is copied into TX shift register and the TXFULL flag is cleared to 0 to indicate that next data can be fetched. A transmit DMA request (if enabled) or a transmitter-empty interrupt (if enabled) is generated at the same time.

27.2.2.1 Data Handling for TX and RX Transfer Groups (전송 그룹에 대한 데이터 처리)

27.2.2.1.1 Data Sequencing of a Transmit Data (송신 데이터의 데이터 시퀀싱)

In multi-buffer mode, any buffer that needs to be transmitted over by the SPI, should be associated with a Transfer Group. Each TG (Transfer Group) will have a Trigger Source based on which it'll be triggered. Once a TG is triggered, the buffers belonging to it will be transmitted.

다중 버퍼 모드에서, SPI에 의해 전송 될 필요가있는 버퍼는 전송 그룹과 연관되어야한다. 각 TG (Transfer Group)에는 트리거 될 트리거 소스가 있습니다. TG가 트리거되면 해당 TG에 속한 버퍼가 전송됩니다.

Sequencer(FSM) controls the data flow from the multi- buffer RAM to the Shift Register. The Multi-buffer Control Logic has arbitration logic between VBUS and the Sequencer accessing the multi buffer RAM. Sequencer picks up a highest priority Transfer Group from among the active TGs to be serviced. For the selected TG the starting buffer to be transferred is obtained from the PSTART of the respective TGxCTRL register.

시퀀서 (FSM)는 다중 버퍼 RAM에서 시프트 레지스터로의 데이터 흐름을 제어합니다. 다중 버퍼 제어 논리는 VBUS와 다중 버퍼 RAM에 액세스하는 시퀀서 사이의 조정 논리를가집니다. 시퀀서는 서비스 할 활성 TG 중에서 우선 순위가 가장 높은 전송 그룹을 선택합니다. 선택된 TG에 대해, 전송 될 시작 버퍼는 각각의 TGxCTRL 레지스터의 PSTART로부터 얻어진다.

Sequencer requests for the selected buffer through the Multi-buffer Control Logic, and once it receives the data, it reads the control fields to determine the subsequent action. Once the buffer is determined to be ready for transfer, the data is written to the TX SHIFT REGISTER by the Sequencer. This triggers the Kernel FSM to initiate the SPI transfer.

Sequencer는 Multi-Buffer Control Logic을 통해 선택한 버퍼를 요청하고 데이터를 수신하면 제어 필드를 읽어 후속 작업을 결정합니다. 일단 버퍼가 전송을 위해 준비되면, 데이터는 시퀀서에 의해 TX SHIFT REGISTER에 기록됩니다. 그러면 커널 FSM이 SPI 전송을 시작합니다.

Once the Sequencer is finished writing to the TX SHIFT REGISTER, it prefetches the next buffer to be transferred from the multi-buffer RAM and stores the Data.

일단 시퀀서가 TX SHIFT REGISTER에 쓰기를 끝내면 다중 버퍼 RAM에서 전송 될 다음 버퍼를 선 인출하고 데이터를 저장합니다.

Once the Sequencer is finished writing to the TX SHIFT REGISTER, it prefetches the next buffer to be transferred from the multi-buffer RAM and stores the Data.

일단 시퀀서가 TX SHIFT REGISTER에 쓰기를 끝내면 다중 버퍼 RAM에서 전송 될 다음 버퍼를 선 인출하고 데이터를 저장합니다.

Sequencer writes the prefetched Transmit Data to the Shift Register immediately upon request by the Kernel. This way, the throughput of the SPI transfer is increased in Master mode of operation. In case of Slave mode, after the Receive data is copied to the RX RAM, Sequencer waits for the next active Chip Select trigger to fetch the next data.

시퀀서는 커널이 요청하면 프리 페치 된 전송 데이터를 즉시 시프트 레지스터에 씁니다. 이렇게하면 마스터 작동 모드에서 SPI 전송 처리량이 증가합니다. 슬레이브 모드의 경우 수신 데이터가 RX RAM으로 복사 된 후 Sequencer는 다음 데이터를 가져 오기 위해 다음 활성 칩 선택 트리거를 기다립니다.

27.2.2.1.2 Data Sequencing of the Received Data (수신 된 데이터의 데이터 시퀀싱)

At the end of a SPI transfer, the received Data is copied to SPIBUF register and then forwarded to the Sequencer. The Sequencer then, requests the Multi-buffer Control Logic to write the received data to the respective RXRAM location. Along with Received Data, the Status fields like Transmission Error Flags and the Last Chip Select Number (LCSNR) are forwarded to be updated in the Status Field of the RXRAM.

SPI 전송이 끝나면 수신 된 데이터는 SPIBUF 레지스터에 복사 된 다음 시퀀서로 전달됩니다. 그런 다음 Sequencer는 수신 된 데이터를 각 RXRAM 위치에 기록하도록 다중 버퍼 제어 로직에 요청합니다. Received Data와 함께 전송 오류 플래그 및 LCSNR (Last Chip Select Number)과 같은 상태 필드는 RXRAM의 상태 필드에서 업데이트되도록 전달됩니다.

Sequencer clears the RXEMPTY bit while writing a new Received Data in the RXRAM. If the RXEMPTY bit is already 0, then the Sequencer sets the RCVR_OVRN bit to 1 to indicate that this particular location has been overwritten in the RXRAM.

시퀀서는 새로운 수신 데이터를 RXRAM에 쓰는 동안 RXEMPTY 비트를 지웁니다. RXEMPTY 비트가 이미 0이면 시퀀서는 RCVR_OVRN 비트를 1로 설정하여이 특정 위치가 RXRAM에 겹쳐 사용하였음을 나타냅니다.

27.2.3 DMA Requests (DMA 요청)

In order to reduce CPU overhead in handling SPI message traffic on a character-by-character basis, SPI can use the DMA controller to transfer the data

문자 단위로 SPI 메시지 트래픽을 처리 할 때 CPU 오버 헤드 줄이기 위해 SPI는 DMA 컨트롤러를 사용하여 데이터를 전송할 수 있습니다

27.2.3.1 SPI/MibSPI Compatibility Mode DMA Requests (호환모드 DMA요청)

The DMA request enable bit (DMA REQ EN) controls the assertion of requests to the DMA controller module.

When a character is being transmitted or received, the SPI will signal the DMA via the DMA request signals, TX_DMA_REQ and RX_DMA_REQ. The DMA controller will then perform the required data transfer.

DMA 요청 인 에이블 비트 (DMA REQ EN)는 DMA 컨트롤러 모듈에 대한 요청의 표명을 제어합니다.

캐릭터가 송신 또는 수신 될 때, SPI는 DMA 요구 신호 TX_DMA_REQ 및 RX_DMA_REQ를 통해 DMA를 신호 할 것이다. DMA 컨트롤러는 필요한 데이터 전송을 수행합니다.

For efficient behavior during DMA operations, the transmitter empty and receive-buffer full interrupts can be disabled. For specific DMA features, see the DMA controller specification.

DMA 동작 중 효율적인 동작을 위해 송신기 비어 있고 수신 버퍼 가득 참 인터럽트가 비활성화 될 수 있습니다. 특정 DMA 기능에 대해서는 DMA 컨트롤러 사양을 참조하십시오.

The SPI generates a request on the TX_DMA_REQ line each time the TX data is copied to the TX shift register either from the TXBUF or from peripheral data bus (when TXBUF is empty).

SPI는 TX 데이터가 TXBUF 또는 주변 데이터 버스 (TXBUF가 비어있는 경우)에서 TX 시프트 레지스터로 복사 될 때마다 TX_DMA_REQ 라인에 대한 요청을 생성합니다.

The first TX_DMA_REQ pulse is generated when either of the following is true:

- DMA REQ EN (SPIINT0[16]) is set to 1 while SPIEN (SPIGCR1[24]) is already 1.
- SPIEN (SPIGCR1[24]) is set to 1 while DMA REQ EN (SPIINT0[16]) is already 1.

첫 번째 TX_DMA_REQ 펄스는 다음 중 하나가 참일 때 생성됩니다 :

- SPIEN (SPIGCR1 [24])이 이미 1 인 동안 DMA REQ EN (SPIINT0 [16])은 1로 설정됩니다.
- DMA REQ EN (SPIINT0 [16])이 이미 1 인 동안 SPIEN (SPIGCR1 [24])이 1로 설정됩니다.

The SPI generates a request on the RX_DMA_REQ line each time the received data is copied to the SPIBUF.

SPI는 수신 된 데이터가 SPIBUF에 복사 될 때마다 RX_DMA_REQ 라인에서 요청을 생성합니다.

27.2.3.2 DMA in Multi-Buffer Mode

The MibSPI provides sophisticated programmable DMA control logic that completely eliminates the necessity of CPU intervention for data transfers, once programmed. When the multi-buffer mode is used, the DMA enable bit in the SPIINT0 register is ignored. DMA source or destination should be only the multibuffer RAM and not SPIDAT0 / SPIDAT1 or SPIBUF register as in case of compatibility mode DMA

MibSPI는 프로그래밍 된 데이터 전송을 위한 CPU 개입의 필요성을 완전히 제거하는 정교한 프로그래밍 가능 DMA 제어 로직을 제공합니다.

멀티 버퍼 모드가 사용되면, SPIINT0 레지스터의 DMA 인 에이블 비트는 무시된다.

DMA 소스 또는 대상은 호환 모드 DMA의 경우와 같이 SPIDAT0 / SPIDAT1 또는 SPIBUF 레지스터가 아닌 멀티 버퍼 RAM이어야 합니다.

The MibSPI offers up to eight DMA channels (for SEND and RECEIVE). All of the DMA channels are programmable individually and can be hooked to any buffer. The MibSPI provides up to 16 DMA request lines, and DMA requests from any channel can be programmed to be routed through any of these 16 lines. A DMA transfer can trigger both transmit and receive.

MibSPI는 최대 8 개의 DMA 채널을 제공합니다 (SEND 및 RECEIVE 용).

모든 DMA 채널은 개별적으로 프로그래밍 가능하며 모든 버퍼에 연결할 수 있습니다. MibSPI는 최대 16 개의 DMA 요청 라인을 제공하며 모든 채널의 DMA 요청은 이 16 개의 라인 중 하나를 통해 라우팅되도록 프로그래밍 할 수 있습니다.

DMA 전송은 전송과 수신을 모두 트리거 할 수 있습니다.

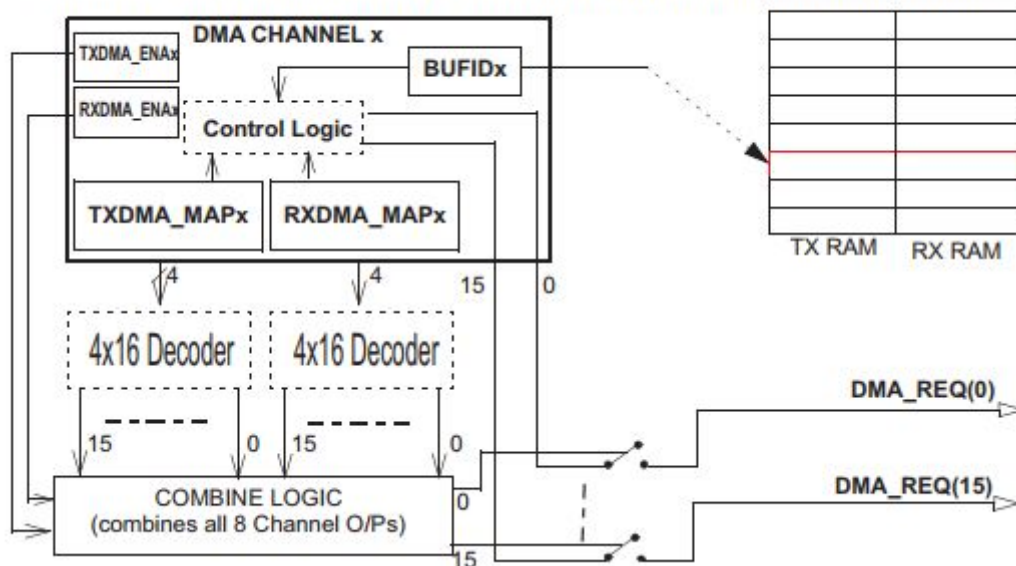
Each DMA channel has the capability to transfer a block of up to 32 data words without interruption using only one buffer of the array by configuring the DMAxCTRL register.

Using the DMAxCOUNT and DMAxTLEN register, up to 65535 (64K) words of data can be transferred without any interruption using just one buffer of the array.

This enables the transfer of memory blocks from or into an external SPI memory.

각 DMA 채널은 DMAxCTRL 레지스터를 구성하여 어레이의 하나의 버퍼만을 사용하여 중단없이 최대 32 개의 데이터 워드 블록을 전송할 수 있습니다. DMAxCOUNT 및 DMAxTLEN 레지스터를 사용하면 어레이의 단 하나의 버퍼를 사용하여 중단없이 최대 65535 (64K) 워드의 데이터를 전송할 수 있습니다. 이를 통해 메모리 블록을 외부 SPI 메모리에서 또는 외부 SPI 메모리로 전송할 수 있습니다.

Figure 27-3. DMA Channel and Request Line (Logical) Structure in Multi-buffer Mode



P. 1231 이후로는 필요한거 찾아서 보세용

