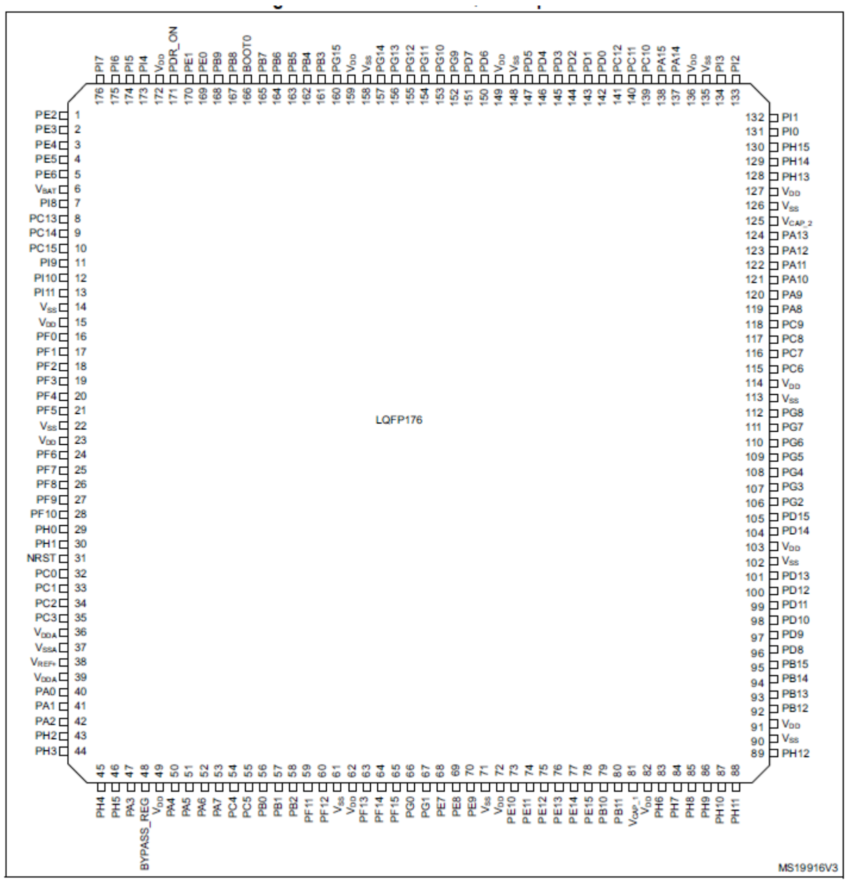
STM32F407 SPI



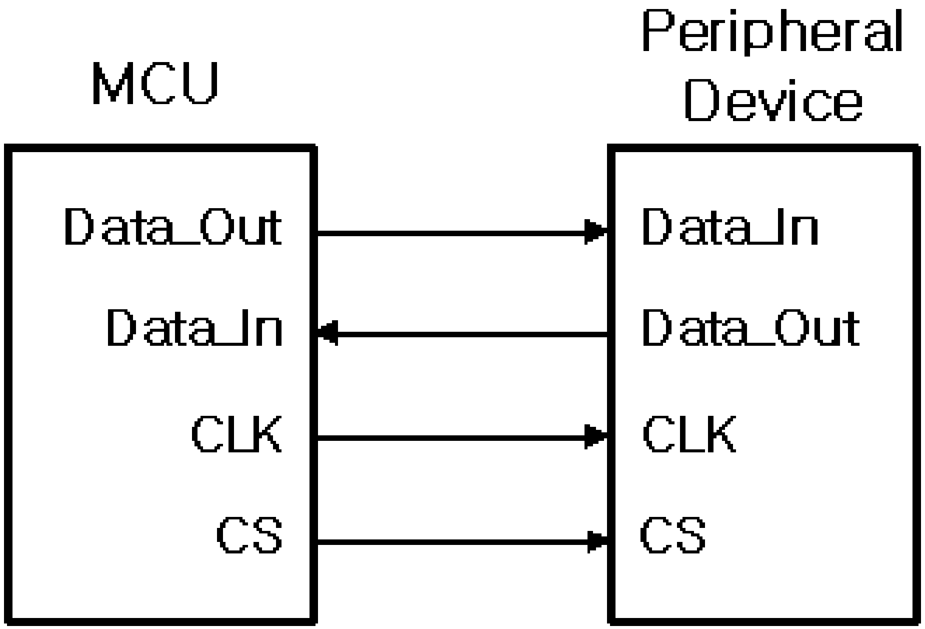
한국산업기술대학교 메카트로닉스공학과

마이크로컴퓨터응용

담당교수: 남윤석

1. SPI(Serial Peripheral Interface) 개요

- MCU(CPU)와 MCU(CPU), MCU(CPU)와 ADC 또는 EEPROM 같은 주변 소자와의 직렬 데이터 통신을 위해 개발한 근거리용 통신 규격
- CS(Chip Select), CLK, Data_In, Data_Out의 4개의 신호선을 이용하여 고속의 동기식 직렬 통신(20 Mbps 이상의 고속 전송)



< SPI 직렬 통신 개념도 >

2. STM32F407의 SPI 주요 사항

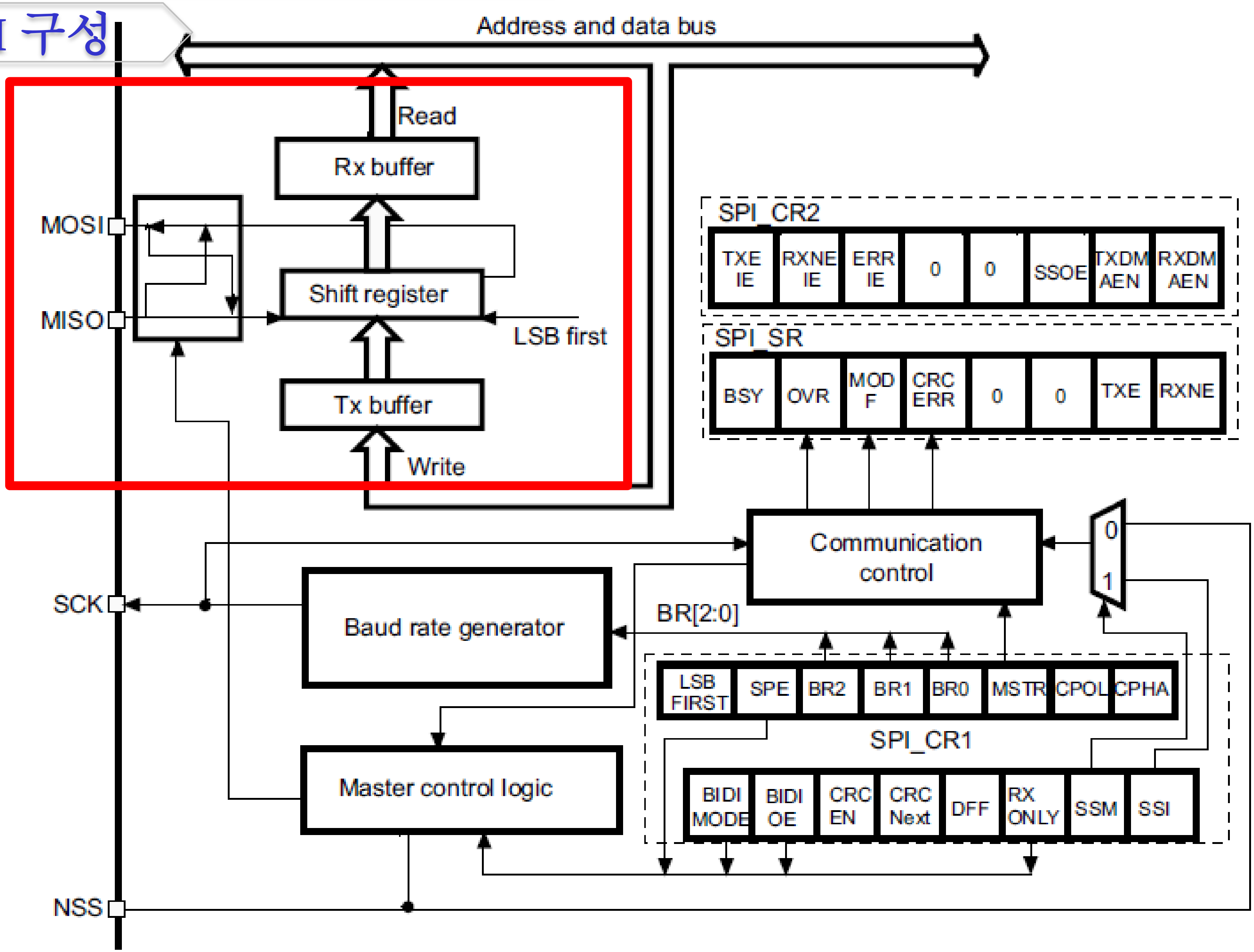
- 총 3개 모듈(SPI1, SPI2, SPI3) 내장
- 3라인(DO,DI,CLK)으로 Full-duplex 동기식 전송
- 전송프레임 형식 선택 : 8 or 16bit
- 마스터 또는 슬레이브로 동작 설정 가능
- 8가지 전송속도 선택 가능(마스터모드 보레이트 프리스케일러 (fPCLK/2 max))
- 슬레이브 모드 주파수 (fPCLK/2 max)
- 마스터가 데이터를 송신 또는 수신을 하더라도 클럭은 항상 마스터에서 발생
- 하나의 문자를 전송할 때, LSB 우선 또는 MSB 우선으로 선택적 전송이 가능
- Clock polarity and phase 선택 가능
- 전송 완료를 알리는 인터럽트 플래그와 쓰기 충돌 방지를 위한 플래그가 내장

Event : ‘TX empty’ and ‘RX full’ 등 flag (인터럽트 가능)
‘SPI bus busy’ status flag

- **Hardware CRC (에러 체크) 기능**
  - **CRC 값은 Tx mode시 마지막 byte로 전송**
  - **Rx 시 자동으로 CRC error checking**
- **Error/Fault: Master mode fault, overrun, CRC error flags (인터럽트 가능)**
- **NSS management by hardware or software for both master and slave: dynamic change of master/slave operations**

3. STM32F407의 SPI 구성과 동작

3.1 SPI 구성

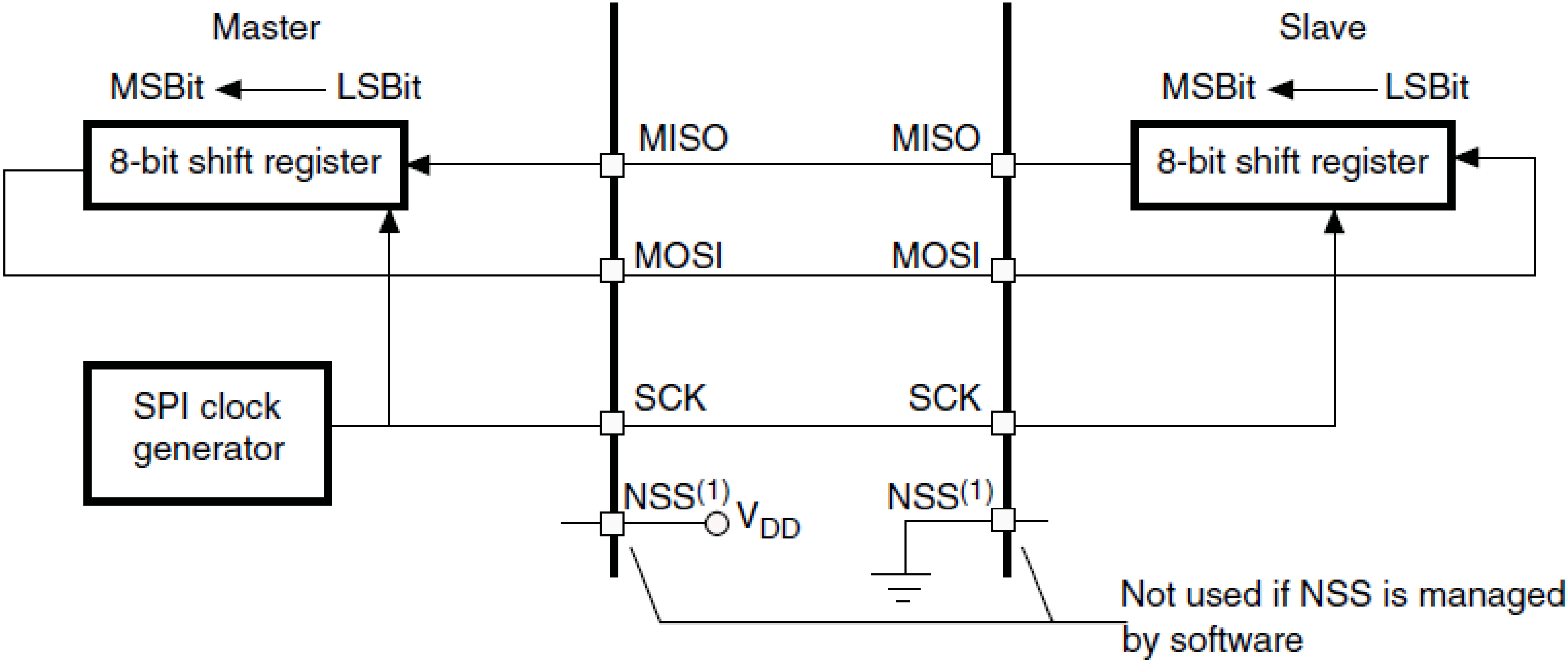


3.2 SPI 모듈의 핀 구성

- **MISO: Master In(RX) / Slave Out(TX) data**
- **MOSI: Master Out(TX) / Slave In(RX) data**
- **SCK: Serial Clock (SPI masters** 에서 출력, **SPI slaves**로 입력**)**
- **NSS: Slave select**
  - **Optional pin to select a slave device**
  - **SPI master**가 각 **slave**와 개별적으로 통신하기 위한 **'Chip Select'**
  - **NSS** 핀은 **software**모드로 사용 시 사용 안 함

SPI 핀	마스터 모드	슬레이브 모드
MOSI	출력(사용자 설정)	입력
MISO	입력	출력(사용자 설정)
SCK	출력(사용자 설정)	입력
/SS	출력(사용자 설정)	입력

3.3 SPI 동작



- **MOSI(MASTER) ←→ MOSI(SLAVE)**
  MISO(MASTER) ←→ MISO(SLAVE)
- 통신은 항상 마스터가 시작
  :마스터가 **MOSI** 를 통해 슬레이브에 데이터 송신, 슬레이브는 **MISO**를 통해 마스터에 데이터를 송신(**SCK** 클럭을 가진 동기 **full-duplex** 통신)

4. Slave select (NSS) pin management

● **Management bit : SPI→CR1.SSM**

• **Software NSS management (SSM = 1)**

: SPI→CR1.SSI에 의해 slave chip select 신호 발생 (이때 NSS pin은 free)

Slave MCU에서는 SPI→CR1.SSI=0 로 설정

Master MCU에서는 SPI→CR1.SSI=1 로 설정

• **Hardware NSS management (SSM = 0)**

SPI→CR1.SSOE 에 따른 2가지 방법

(1) NSS output enabled (SSM = 0, SSOE = 1)

- 디바이스가 마스터 모드일 때 사용
- 마스터가 통신시작시 NSS은 'low', SPI가 disable될 때까지 'low' 유지 (즉, SPI 송신명령 실행시 자동으로 NSS 핀이 'low')

(2) NSS output disabled (SSM = 0, SSOE = 0)

- 마스터 모드에서 'multi-master 기능' 가능
- 슬레이브 모드에서 NSS 핀은 'chip select' 기능으로 사용:

NSS가 low이면 디바이스(슬레이브) 선택됨

NSS가 high이면 디바이스(슬레이브) 선택안됨

5. SPI Clock phase and Clock polarity

- SPI→CR1.CPOL

0:CK to 0 when idle

1:CK to 1 when idle

- SPI→CR1.CPHA

0:1st CK transition에서 data capture

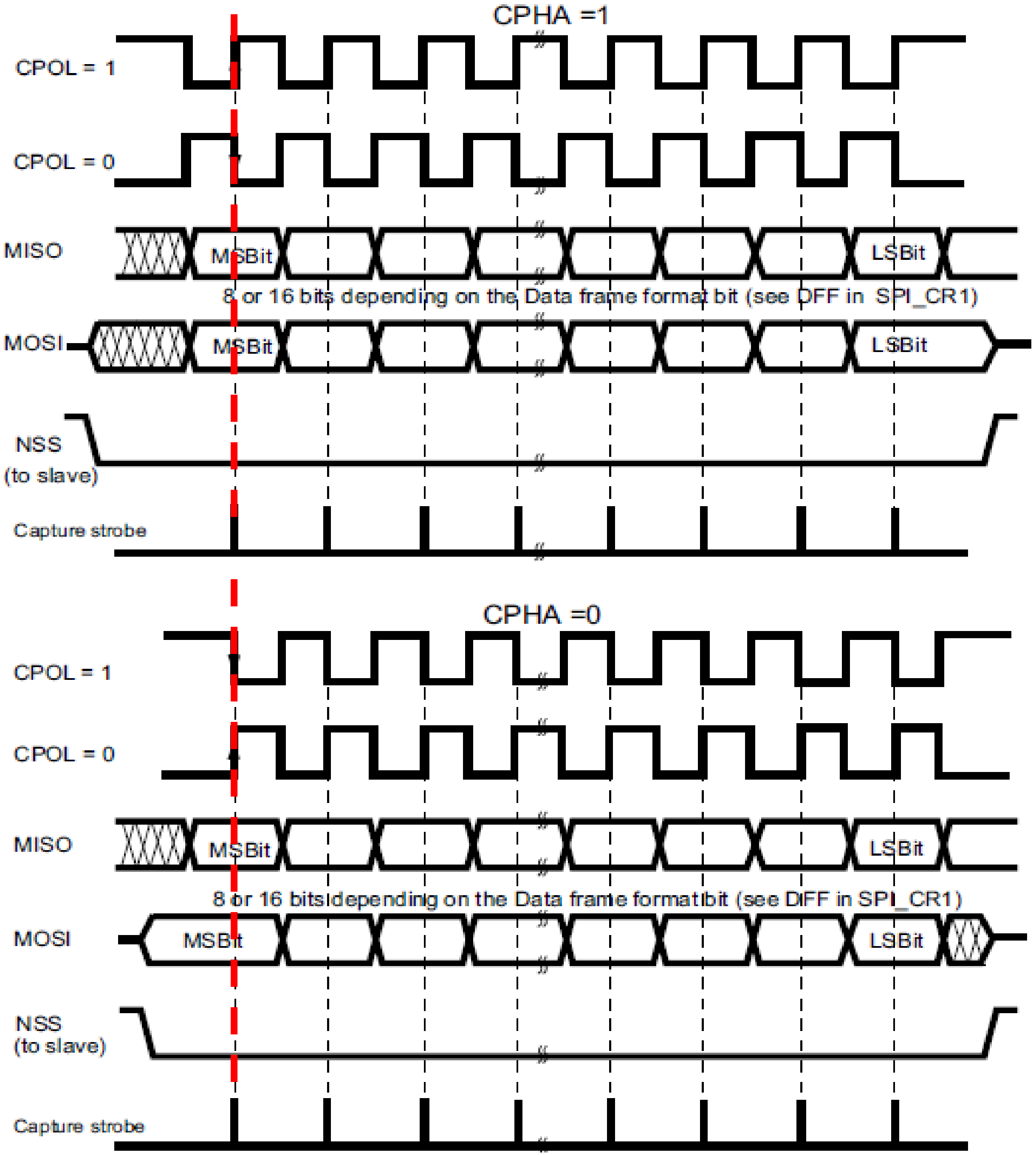
1: 2nd CK transition에서 data capture

- SPI→CR1.LSBFIRST

: MSB-first or LSB-first 선택

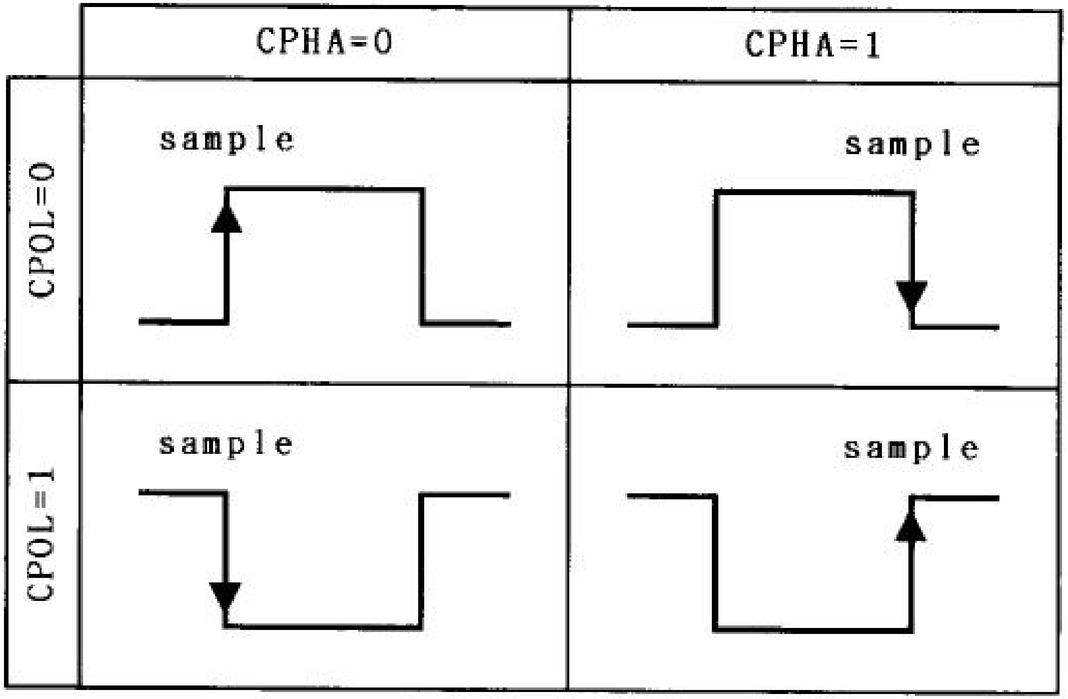
- SPI→CR1.DFF

: 8 or 16 bits 선택



● **CPOL (clock polarity)과 CPHA (clock phase) bit 결합으로 'data capture clock edge'를 선택 (4가지)**

- **CPOL 비트**
  - **0 : clock의 기본 상태가 0**
  - **1 : clock의 기본 상태는 1**
- **CPHA 비트**
  - **0 : 매 Clock 의 첫 번째 edge에서 데이터가 샘플링**
  - **1 : 두 번째 edge 에서 데이터가 샘플링**



6. SPI Master mode 설정

① **SPIx→CR1.BR [2 : 0]** 비트를 선택하여 클럭 속도 정의

② **CPOL** 및 **CPHA** 비트를 선택하여 데이터 전송과 클럭 간의 네 가지 관계 중 하나를 정의

③ **SPIx→CR1.DFF** 비트를 설정하여 **8** 비트 또는 **16** 비트 데이터 프레임 형식을 정의

④ 프레임 포맷을 정의하기 위해 **SPIx→CR1.LSBFIRST** 비트 설정

⑤ **NSS pin : Input mode** 인 경우

- **H/W mode**인 경우**: NSS** 핀을 **'H'**에 연결
- **S/W mode**인 경우**: SPIx→CR1.SSM** 및 **SSI** 비트 설정

NSS pin : Output mode인 경우

- **SPIx→CR1.SSOE** 설정

⑥ **SPIx→CR1.MSTR(Master/Slave selection)** 및 **SPIx→CR1. SPE (SPI enable/disable)** 비트 설정

* **MOSI** 핀은 데이터 출력**, MISO** 핀은 데이터 입력

●Transmit sequence

① 데이터(1 byte)가 Tx Buffer(SPIx→DR)에 write (coding)

② 데이터는 Tx Buffer에서 shift register로 이동

- SPIx→SR.TXE = 0
- Interrupt 발생 if SPI→CR2.TXEIE is set

③ 직렬로 MOSI pin을 통해 출력

●Receive sequence

① 상대방(slave)이 보낸 데이터가 MISO pin을 통해 shift register에 도착

② Shift register 에 있는 데이터는 RX Buffer(SPIx→DR)에 이동

- SPIx→SR.RXNE = 1
- Interrupt 발생 if SPIx→CR2.RXNEIE = 1

③ Reading SPIx→DR (coding)

- SPIx→SR.RXNE = 0

● 종합하면, Start sequence in master mode

: In full-duplex (BIDIMODE=0 and RXONLY=0)

- Tx_data(1 byte) → SPIx→DR(Tx buffer) → Shift register → MOSI pin

-동시에, Rx_data(1 byte)→ MISO pin → Shift register → SPIx→DR (Rx buffer)

● Event 발생

- TXE flag (Tx buffer empty) = 1 이 됨
  when SPIx→DR(Tx buffer) → Shift register (Interrupt 발생 if SPIx→CR2.TXEIE = 0)

- RXNE flag (Rx buffer not empty) = 1 이 됨
  when Shift register → SPIx→DR (Rx buffer) (Interrupt 발생 if SPIx→CR2.RXNEIE = 1)

- BSY flag can be used during the last data transfer to wait until the completion of the transfer

●Full-duplex transmit and receive procedure in master (BIDIMODE=0 and RXONLY=0)

1. SPIx→CR1.SPE = 1 to enable SPIx

2. SPIx→DR = 1st Tx data (this clears the TXE flag)

3. Wait until TXE=1 & SPIx→DR = 2nd Tx data

Wait until RXNE=1 & Reading SPIx→DR to get 1st Rx data (this clears the RXNE bit)

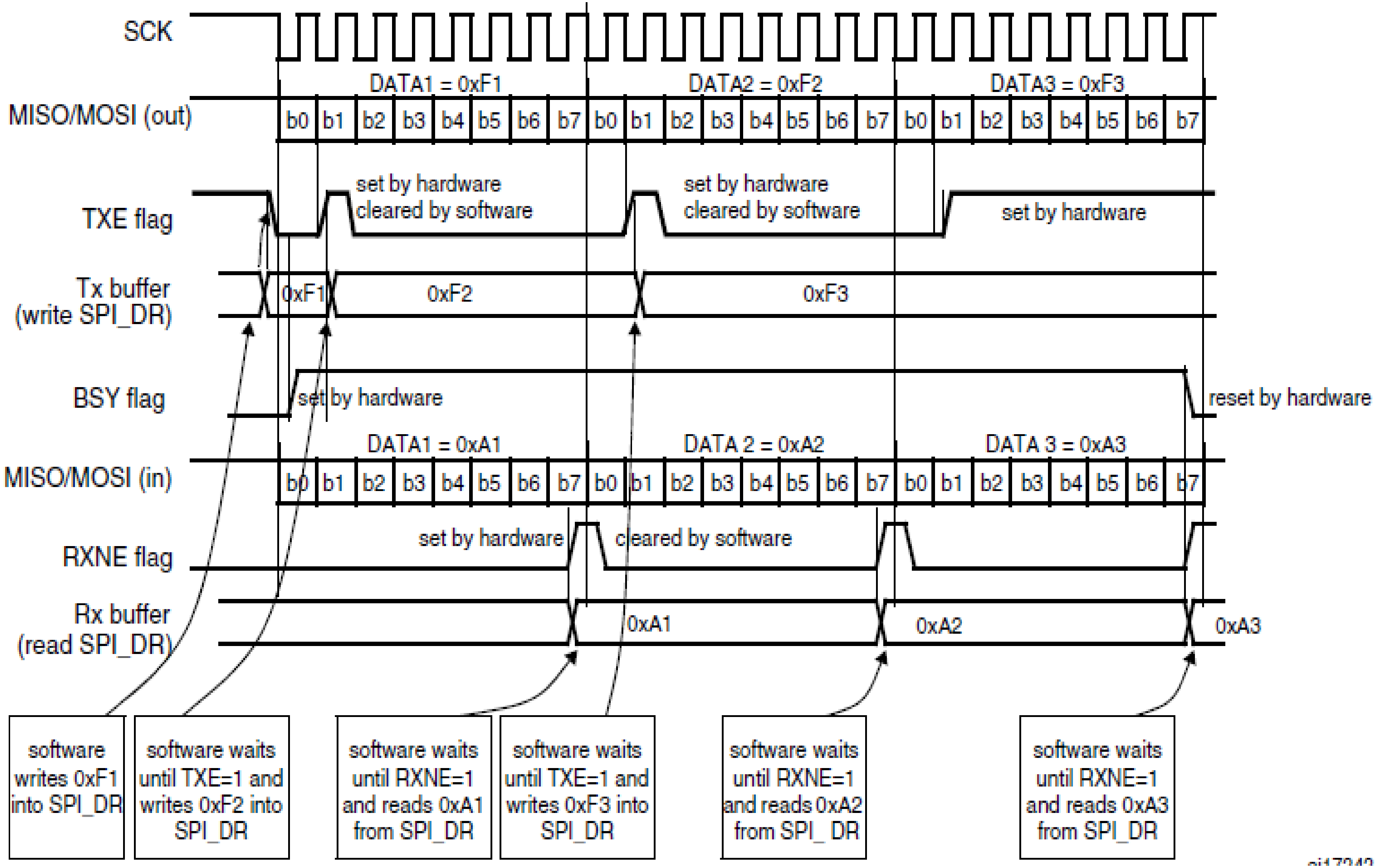
Repeat this operation for each data item to be transmitted /received until the n–1 received data

4. Wait until RXNE=1 and read the last Rx data

5. Wait until TXE=1 and then wait until BSY=0 before disabling the SPI (SPIx→CR1.SPE= 0)

*** 처음 보내는 Tx data(명령)는 그것에 대한 답변이 없고, 두번째 data를 보내면 그 때서야 첫번째 Tx data(명령)에 대한 Rx data(답변)가 수신됨**

Example in Master mode with CPOL=1, CPHA=1



software writes 0xF1 into SPI_DR

software waits until TXE=1 and writes 0xF2 into SPI_DR

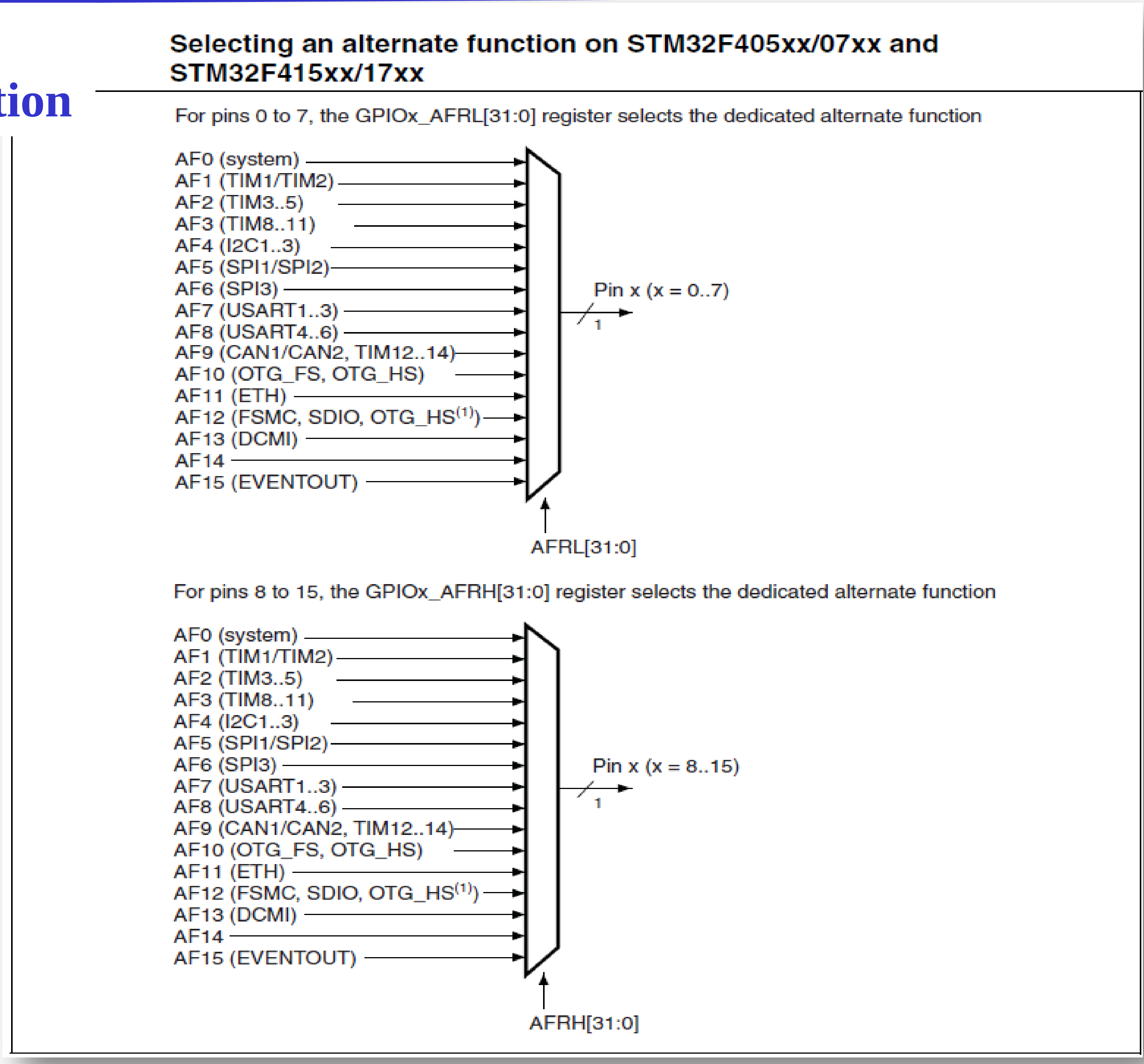
software waits until RXNE=1 and reads 0xA1 from SPI_DR

software waits until TXE=1 and writes 0xF3 into SPI_DR

software waits until RXNE=1 and reads 0xA2 from SPI_ DR

software waits until RXNE=1 and reads 0xA3 from SPI_DR

- **Selecting an Alternate function**



Selecting an alternate function on STM32F405xx/07xx and STM32F415xx/17xx

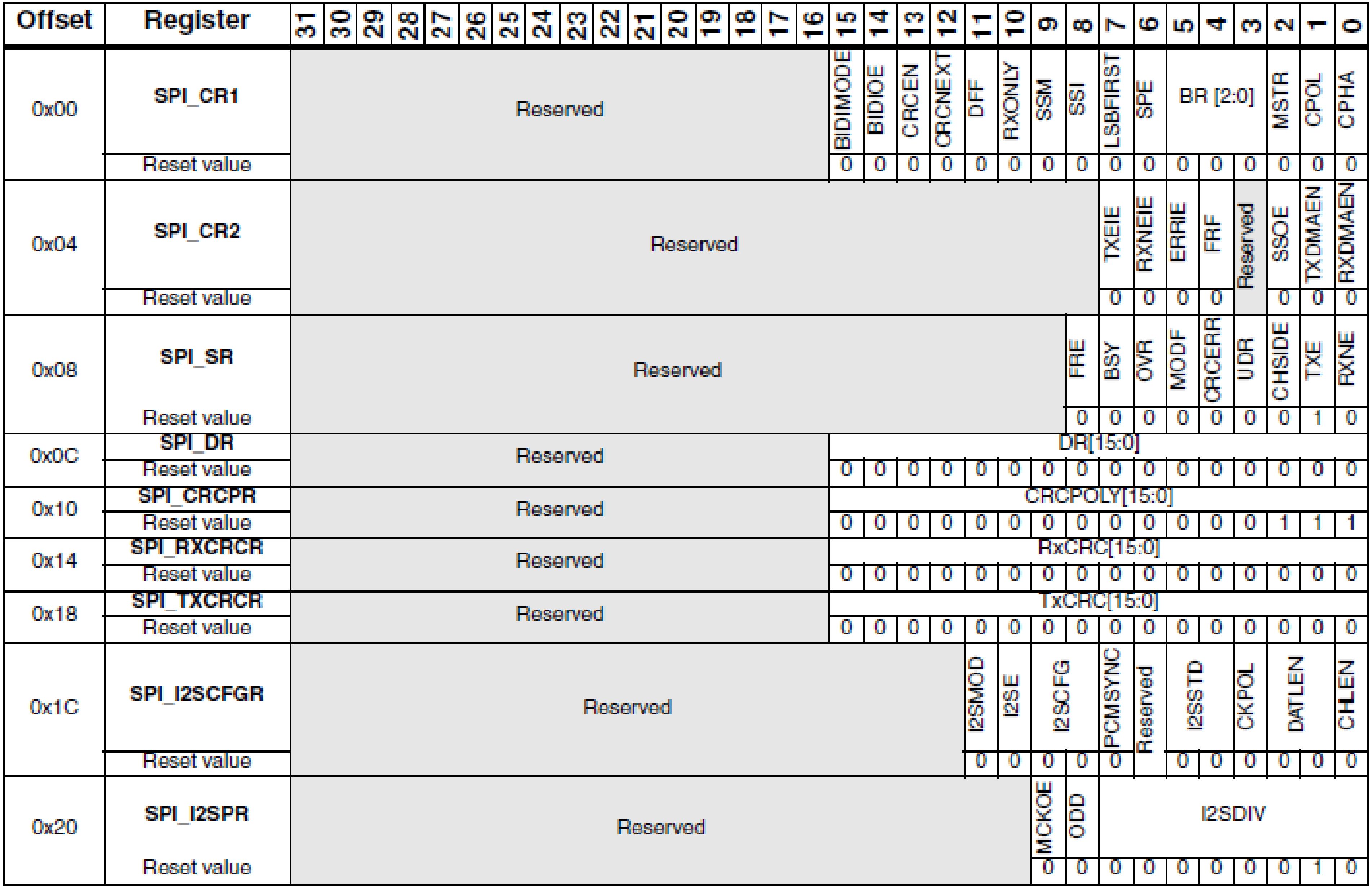
For pins 0 to 7, the GPIOx_AFRL[31:0] register selects the dedicated alternate function

For pins 8 to 15, the GPIOx_AFRH[31:0] register selects the dedicated alternate function

7. SPI 레지스터

7.1 Register Map

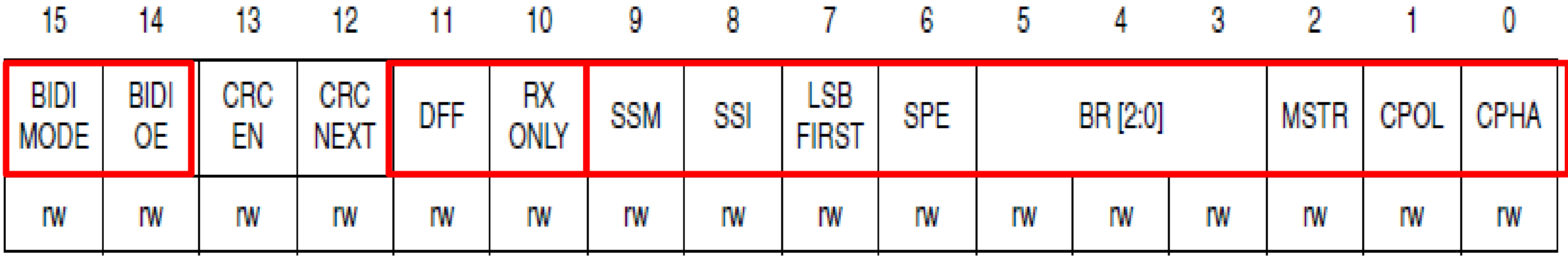
Table 127. SPI register map and reset values



Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	SPI_CR1	Reserved																BIDIMODE	BIDIOE	CRCEN	CRCNEXT	DFF	RXONLY	SSM	SSI	LSBFIRST	SPE	BR [2:0]			MSTR	CPOL	CPHA
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	SPI_CR2	Reserved																								TXEIE	RXNEIE	ERRIE	FRF	Reserved	SSOE	TXDMAEN	RXDMAEN
	Reset value																									0	0	0	0		0	0	0
0x08	SPI_SR	Reserved																							FRE	BSY	OVR	MODF	CRCERR	UDR	CHSIDE	TXE	RXNE
	Reset value																								0	0	0	0	0	0	0	1	0
0x0C	SPI_DR	Reserved																DR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	SPI_CRCPR	Reserved																CRCPOLY[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0x14	SPI_RXCRCR	Reserved																RxCRC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	SPI_TXCRCR	Reserved																TxCRC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	SPI_I2SCFGR	Reserved																				I2SMOD	I2SE	I2SCFG		PCMSYNC	Reserved	I2SSTD		CKPOL	DATLEN		CHLEN
	Reset value																					0	0	0	0	0		0	0	0	0	0	0
0x20	SPI_I2SPR	Reserved																						MCKOE	ODD	I2SDIV							
	Reset value																							0	0	0	0	0	0	0	0	1	0

7.2 SPI 주요 레지스터

- SPI Control register 1 (SPIx_CR1) : **Reset value: 0x0000 0000**



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIDI MODE	BIDI OE	CRC EN	CRC NEXT	DFF	RX ONLY	SSM	SSI	LSB FIRST	SPE	BR [2:0]			MSTR	CPOL	CPHA
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

<u>Bit 15 BIDIMODE: Bidirectional data mode enable</u>

0: 2-line unidirectional data mode selected (full-duflex)

1: 1-line bidirectional data mode selected (half-duflex)

Bit 14 BIDIOE: Output enable in bidirectional mode

This bit combined with the BIDImode bit selects the direction of transfer in bidirectional mode

0: Output disabled (receive-only mode)

1: Output enabled (transmit-only mode)

Note: In master mode, the MOSI pin is used and in slave mode, the MISO pin is used.

<u>Bit 11 DFF: Data frame format</u>

0: 8-bit data frame format is selected for transmission/reception

1: 16-bit data frame format is selected for transmission/reception

Note: This bit should be written only when SPI is disabled (SPE = '0') for correct operation

Bit 10 RXONLY: Receive only

This bit combined with the BIDImode bit selects the direction of transfer in 2-line unidirectional mode. This bit is also useful in a multislave system in which this particular slave is not accessed, the output from the accessed slave is not corrupted.

0: Full duplex (Transmit and receive)

1: Output disabled (Receive-only mode)

Bit 9 SSM: Software slave management

When the SSM bit is set, the NSS pin input is replaced with the value from SSI bit.

0: Software slave management disabled

1: Software slave management enabled

Bit 8 SSI: Internal slave select

This bit has an effect only when the SSM bit is set. The value of this bit is forced onto the NSS pin and the IO value of the NSS pin is ignored.

Bit 7 LSBFIRST: Frame format

0: MSB transmitted first   1: LSB transmitted first

Bit 6 SPE: SPI enable

0: Peripheral disabled   1: Peripheral enabled

Bits 5:3 BR[2:0]: Baud rate control

000: fPCLK/2   100: fPCLK/32   001: fPCLK/4   101: fPCLK/64
010: fPCLK/8   110: fPCLK/128   011: fPCLK/16   111: fPCLK/256

Bit 2 MSTR: Master selection

0: Slave configuration   1: Master configuration

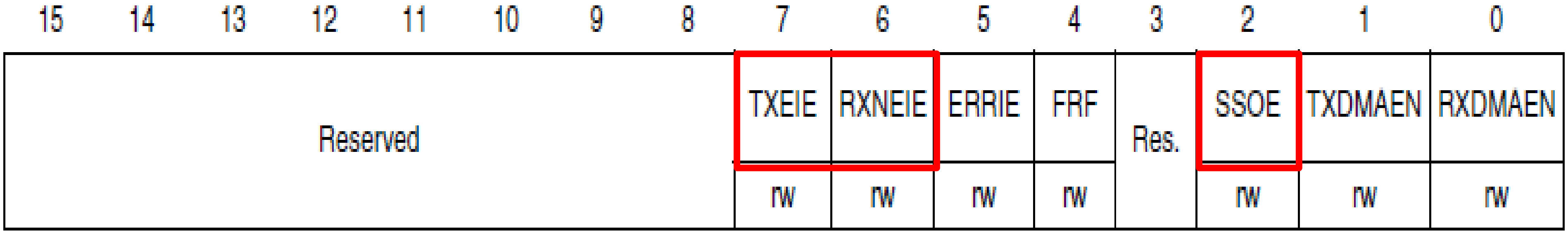
Bit1 CPOL: Clock polarity

0: CK to 0 when idle   1: CK to 1 when idle

Bit 0 CPHA: Clock phase

0: The first clock transition is the first data capture edge
1: The second clock transition is the first data capture edge

- **SPI Control register 2 (SPIx_CR2) : Reset value: 0x0000 0000**



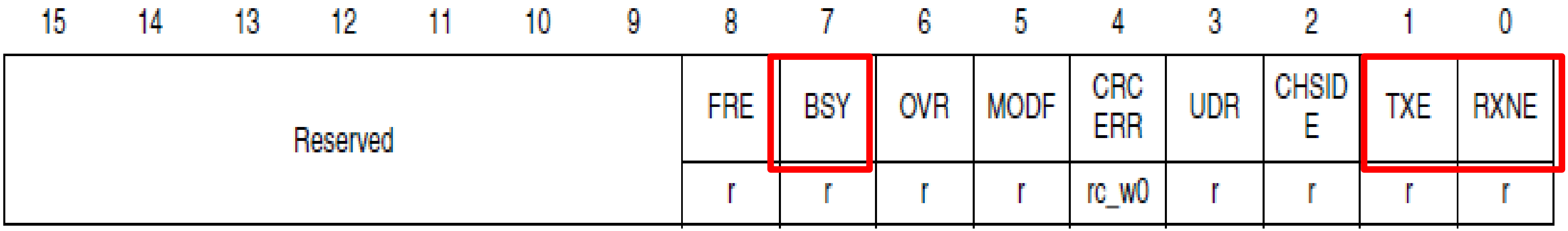
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TXEIE	RXNEIE	ERRIE	FRF	Res.	SSOE	TXDMAEN	RXDMAEN
								rw	rw	rw	rw		rw	rw	rw

<u>Bit 7 TXEIE: Tx buffer empty interrupt enable</u>
0: TXE interrupt masked
1: TXE interrupt not masked. Used to generate an interrupt request when the TXE flag is set.

<u>Bit 6 RXNEIE: RX buffer not empty interrupt enable</u>
0: RXNE interrupt masked
1: RXNE interrupt not masked. Used to generate an interrupt request when the RXNE flag is set.

<u>Bit 2 SSOE: SS output enable</u>
0: SS output is disabled in master mode and the cell can work in multimaster configuration
1: SS output is enabled in master mode and when the cell is enabled. The cell cannot work in a multimaster environment.

- SPI Status register (SPIx_SR) : Reset value: 0x0002



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							FRE	BSY	OVR	MODF	CRC ERR	UDR	CHSIDE	TXE	RXNE
							r	r	r	r	rc_w0	r	r	r	r

<u>Bit 7 BSY: Busy flag</u>

0: SPI (or I2S) not busy

1: SPI (or I2S) is busy in communication or Tx buffer is not empty

This flag is set and cleared by hardware.

<u>Bit 1 TXE: Transmit buffer empty</u>

0: Tx buffer not empty

1: Tx buffer empty

<u>Bit 0 RXNE: Receive buffer not empty</u>

0: Rx buffer empty

1: Rx buffer not empty

- **SPI Data register (SPIx_DR) : Reset value: 0x0000**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 15:0 DR[15:0]: Data register

Data received or to be transmitted.

The data register is split into 2 buffers - one for writing (Transmit Buffer) and another one for reading (Receive buffer). A write to the data register will write into the Tx buffer and a read from the data register will return the value held in the Rx buffer.

Notes for the SPI mode:

Depending on the data frame format selection bit (DFF in SPI_CR1 register), the data sent or received is either 8-bit or 16-bit. This selection has to be made before enabling the SPI to ensure correct operation.

For an 8-bit data frame, the buffers are 8-bit and only the LSB of the register (SPI_DR[7:0]) is used for transmission /reception. When in reception mode, the MSB of the register (SPI_DR[15:8]) is forced to 0.

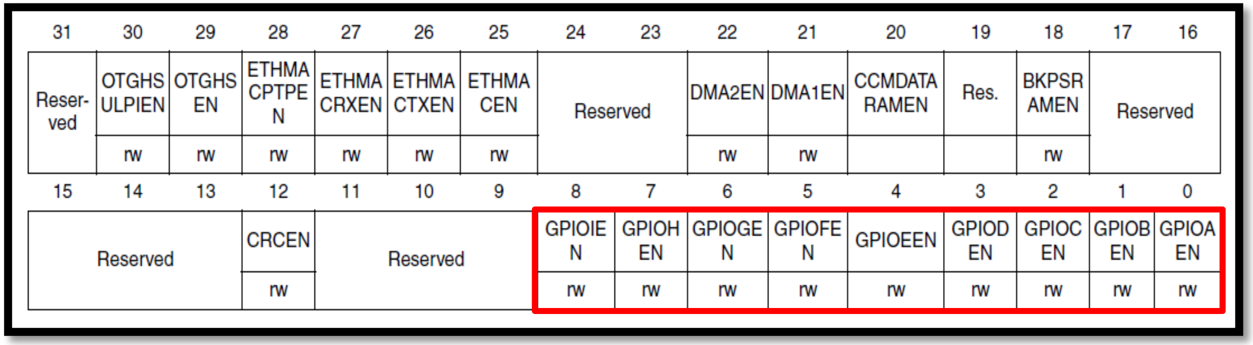
For a 16-bit data frame, the buffers are 16-bit and the entire register, SPI_DR[15:0] is used for transmission/reception.

7.3 NVIC Parameter 및 INT Handler Function

(NVIC Parameter 및 INT Handler Function)
SPIx_IRQn
SPIx_IRQHandler (x=1~3)

RCC clock enable 레지스터 구조

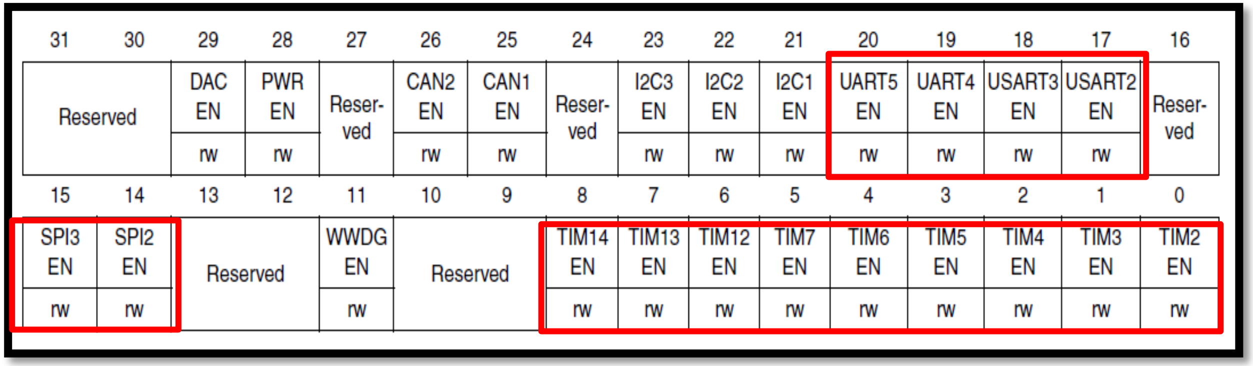
- RCC_AHB1ENR



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reser- ved	OTGHS ULPIEN	OTGHS EN	ETHMA CPTPE N	ETHMA CRXEN	ETHMA CTXEN	ETHMA CEN	Reserved		DMA2EN	DMA1EN	CCMDATA RAMEN	Res.	BKPSR AMEN	Reserved	
	rw	rw	rw	rw	rw	rw			rw	rw			rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CRCEN	Reserved			GPIOIE N	GPIOH EN	GPIOGE N	GPIOFE N	GPIOEEN	GPIOD EN	GPIOC EN	GPIOB EN	GPIOA EN
			rw				rw	rw	rw	rw	rw	rw	rw	rw	rw

GPIO

- RCC_APB1ENR

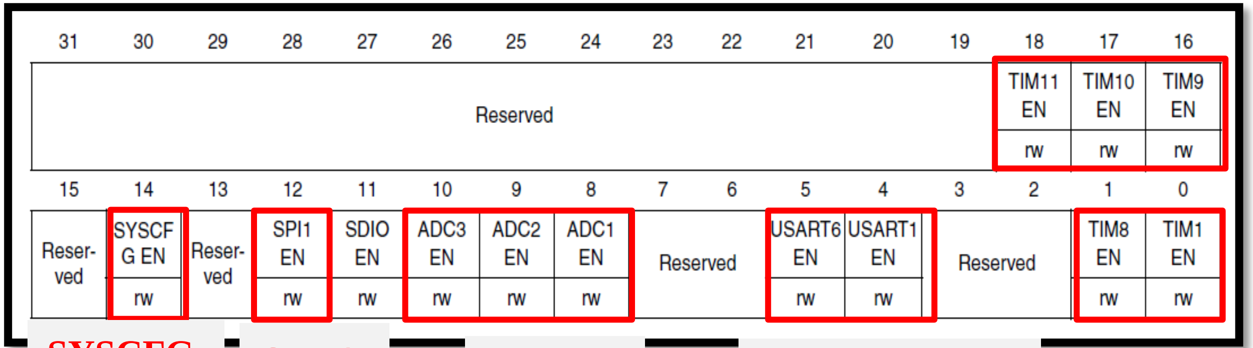


31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		DAC EN	PWR EN	Reser- ved	CAN2 EN	CAN1 EN	Reser- ved	I2C3 EN	I2C2 EN	I2C1 EN	UART5 EN	UART4 EN	USART3 EN	USART2 EN	Reser- ved
		rw	rw		rw	rw		rw	rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3 EN	SPI2 EN	Reserved		WWDG EN	Reserved		TIM14 EN	TIM13 EN	TIM12 EN	TIM7 EN	TIM6 EN	TIM5 EN	TIM4 EN	TIM3 EN	TIM2 EN
rw	rw			rw			rw	rw	rw	rw	rw	rw	rw	rw	rw

USART:2~5

Timer:2~7,
12~14

- RCC_APB2ENR



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													TIM11 EN	TIM10 EN	TIM9 EN
													rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reser- ved	SYSCF G EN	Reser- ved	SPI1 EN	SDIO EN	ADC3 EN	ADC2 EN	ADC1 EN	Reserved		USART6 EN	USART1 EN	Reserved		TIM8 EN	TIM1 EN
	rw		rw	rw	rw	rw	rw			rw	rw			rw	rw

Timer:9~11

Timer:1,8

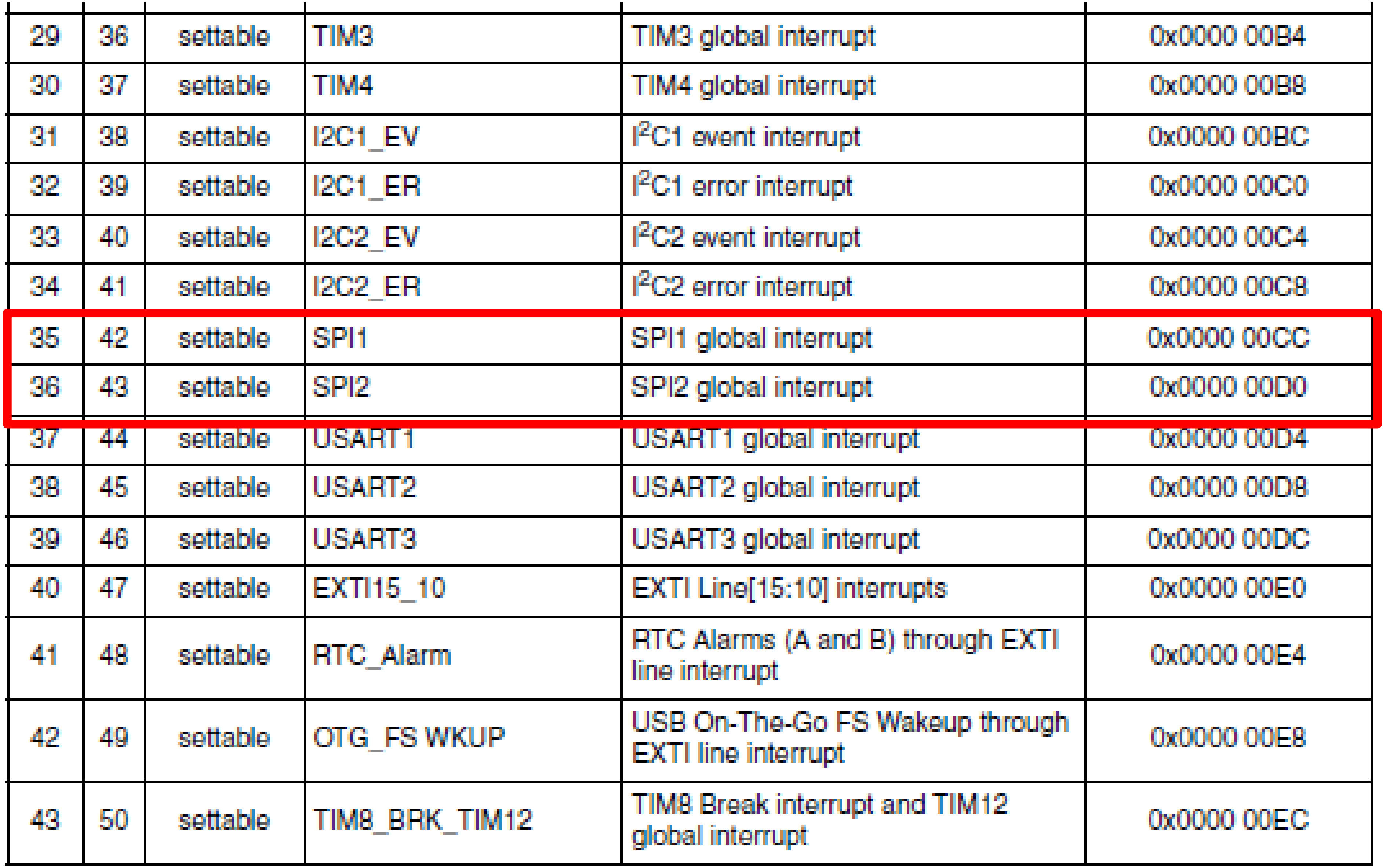
SYSCFG
_EN

SPI1

ADC

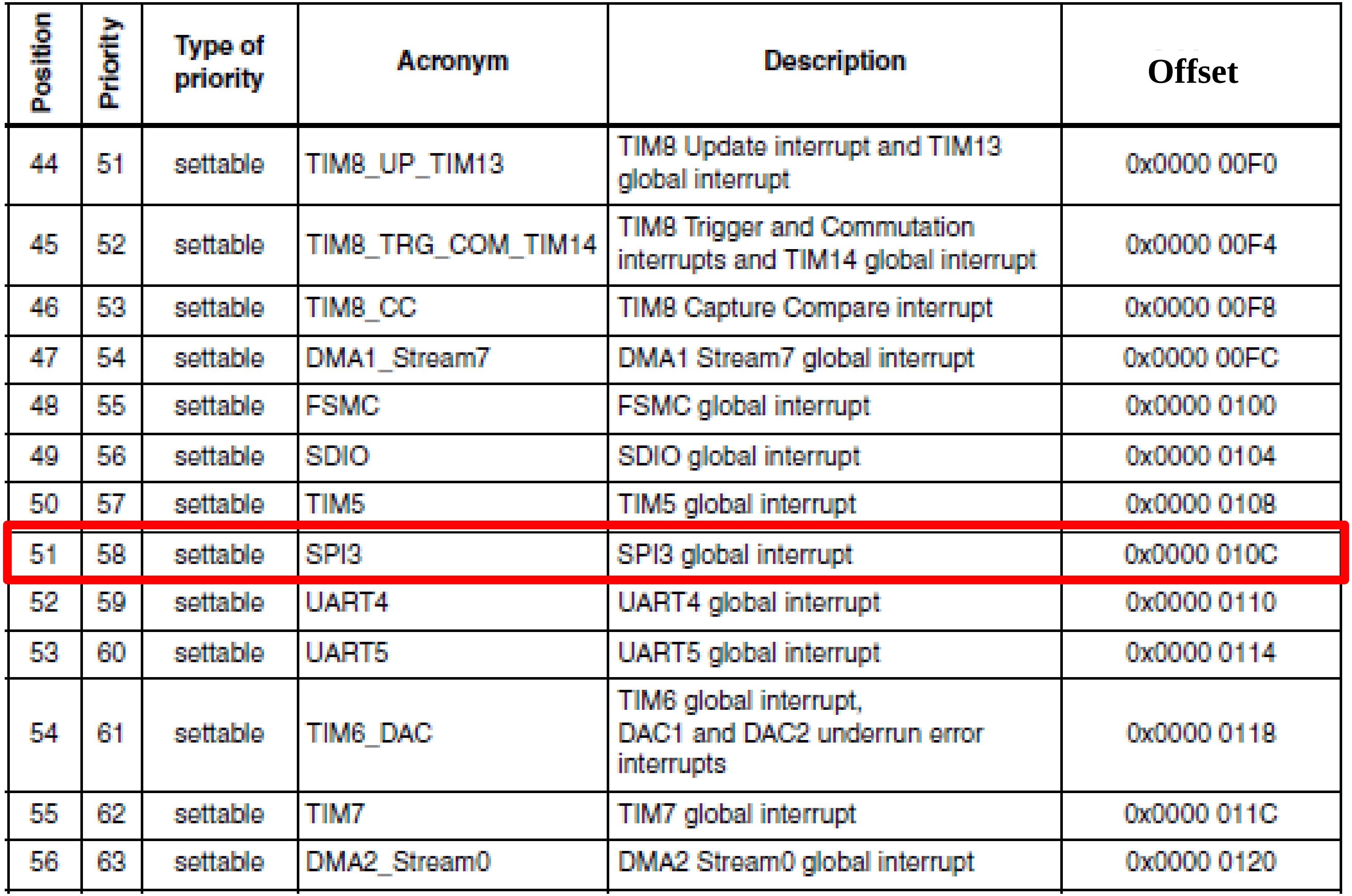
USART:1,6

●인터럽트 벡터 테이블- Interrupts (29~43)



29	36	settable	TIM3	TIM3 global interrupt	0x0000 00B4
30	37	settable	TIM4	TIM4 global interrupt	0x0000 00B8
31	38	settable	I2C1_EV	I^{2}C1 event interrupt	0x0000 00BC
32	39	settable	I2C1_ER	I^{2}C1 error interrupt	0x0000 00C0
33	40	settable	I2C2_EV	I^{2}C2 event interrupt	0x0000 00C4
34	41	settable	I2C2_ER	I^{2}C2 error interrupt	0x0000 00C8
35	42	settable	SPI1	SPI1 global interrupt	0x0000 00CC
36	43	settable	SPI2	SPI2 global interrupt	0x0000 00D0
37	44	settable	USART1	USART1 global interrupt	0x0000 00D4
38	45	settable	USART2	USART2 global interrupt	0x0000 00D8
39	46	settable	USART3	USART3 global interrupt	0x0000 00DC
40	47	settable	EXTI15_10	EXTI Line[15:10] interrupts	0x0000 00E0
41	48	settable	RTC_Alarm	RTC Alarms (A and B) through EXTI line interrupt	0x0000 00E4
42	49	settable	OTG_FS WKUP	USB On-The-Go FS Wakeup through EXTI line interrupt	0x0000 00E8
43	50	settable	TIM8_BRK_TIM12	TIM8 Break interrupt and TIM12 global interrupt	0x0000 00EC

●인터럽트 벡터 테이블- Interrupts (44~56)



Position	Priority	Type of priority	Acronym	Description	Offset
44	51	settable	TIM8_UP_TIM13	TIM8 Update interrupt and TIM13 global interrupt	0x0000 00F0
45	52	settable	TIM8_TRG_COM_TIM14	TIM8 Trigger and Commutation interrupts and TIM14 global interrupt	0x0000 00F4
46	53	settable	TIM8_CC	TIM8 Capture Compare interrupt	0x0000 00F8
47	54	settable	DMA1_Stream7	DMA1 Stream7 global interrupt	0x0000 00FC
48	55	settable	FSMC	FSMC global interrupt	0x0000 0100
49	56	settable	SDIO	SDIO global interrupt	0x0000 0104
50	57	settable	TIM5	TIM5 global interrupt	0x0000 0108
51	58	settable	SPI3	SPI3 global interrupt	0x0000 010C
52	59	settable	UART4	UART4 global interrupt	0x0000 0110
53	60	settable	UART5	UART5 global interrupt	0x0000 0114
54	61	settable	TIM6_DAC	TIM6 global interrupt, DAC1 and DAC2 underrun error interrupts	0x0000 0118
55	62	settable	TIM7	TIM7 global interrupt	0x0000 011C
56	63	settable	DMA2_Stream0	DMA2 Stream0 global interrupt	0x0000 0120

8. STM32F407의 SPI 프로그래밍 실습

7.1 프로그램에서의 SPIx set-up 과정 및 레지스터 설정

RCC 설정

- RCC→AHB1ENR(GPIOy Clock Enable)
- RCC→APBzENR(SPIx Clock Enable)

INT Enable

- NVIC→ISER[ ]

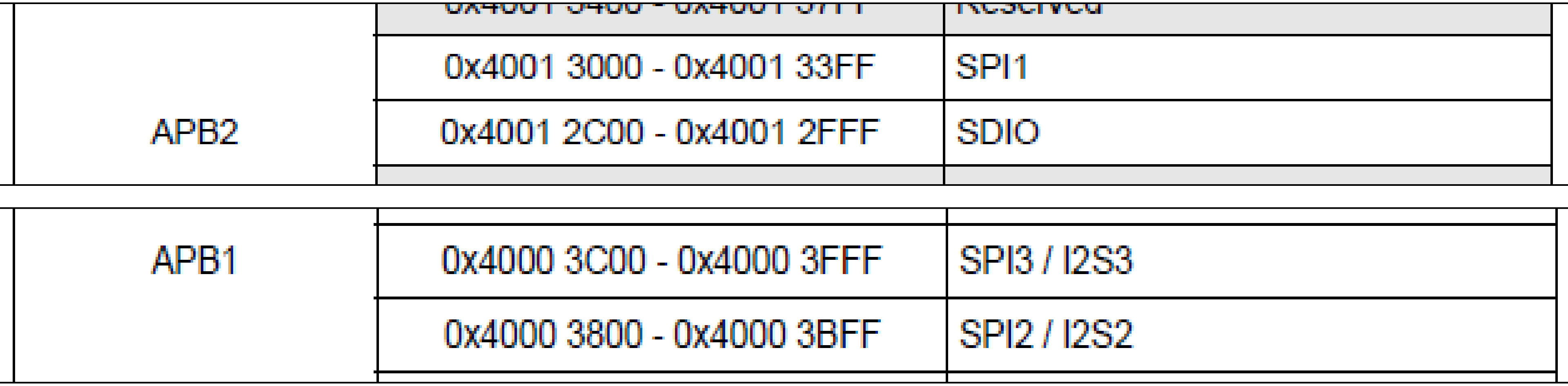
SPIx 초기 설정

- SPIx→CR1, CR2

Int Handler 설정

- SPIx_IRQHandler()

7.2 STM32F407의 SPI의 Address(Memory map)



APB2	0x4001 3400 - 0x4001 37FF	Reserved
	0x4001 3000 - 0x4001 33FF	SPI1
	0x4001 2C00 - 0x4001 2FFF	SDIO

APB1		
	0x4000 3C00 - 0x4000 3FFF	SPI3 / I2S3
	0x4000 3800 - 0x4000 3BFF	SPI2 / I2S2

7.3 STM32F407의 SPI관련 header file(stm32f4xx.h)주요 부분

```
#define PERIPH_BASE    ((uint32_t)0x40000000) /* Peripheral base address */
#define APB1PERIPH_BASE       PERIPH_BASE
#define APB2PERIPH_BASE       (PERIPH_BASE + 0x00010000)

#define SPI1_BASE            (APB2PERIPH_BASE + 0x3000)
#define SPI2_BASE            (APB1PERIPH_BASE + 0x3800)
#define SPI3_BASE            (APB1PERIPH_BASE + 0x3C00)

#define SPI1              ((SPI_TypeDef *) SPI1_BASE)
#define SPI2              ((SPI_TypeDef *) SPI2_BASE)
#define SPI3              ((SPI_TypeDef *) SPI3_BASE)

typedef struct
{ __IO uint16_t CR1;  //SPI control register 1, offset: 0x00
  __IO uint16_t CR2;  // SPI control register 2, offset: 0x04
  __IO uint16_t SR;    // SPI status register, offset: 0x08
  __IO uint16_t DR;   // SPI data register,  offset: 0x0C
  __IO uint16_t CRCPR;     // SPI CRC polynomial register, offset: 0x10
  __IO uint16_t RXCRCR;  // SPI RX CRC register, offset: 0x14
  __IO uint16_t TXCRCR;   // SPI TX CRC register, offset: 0x18
  __IO uint16_t I2SCFGR;   // SPI_I2S configuration register,  offset: 0x1C
  __IO uint16_t I2SPR;     // SPI_I2S prescaler register,  offset: 0x20
} SPI_TypeDef;
```

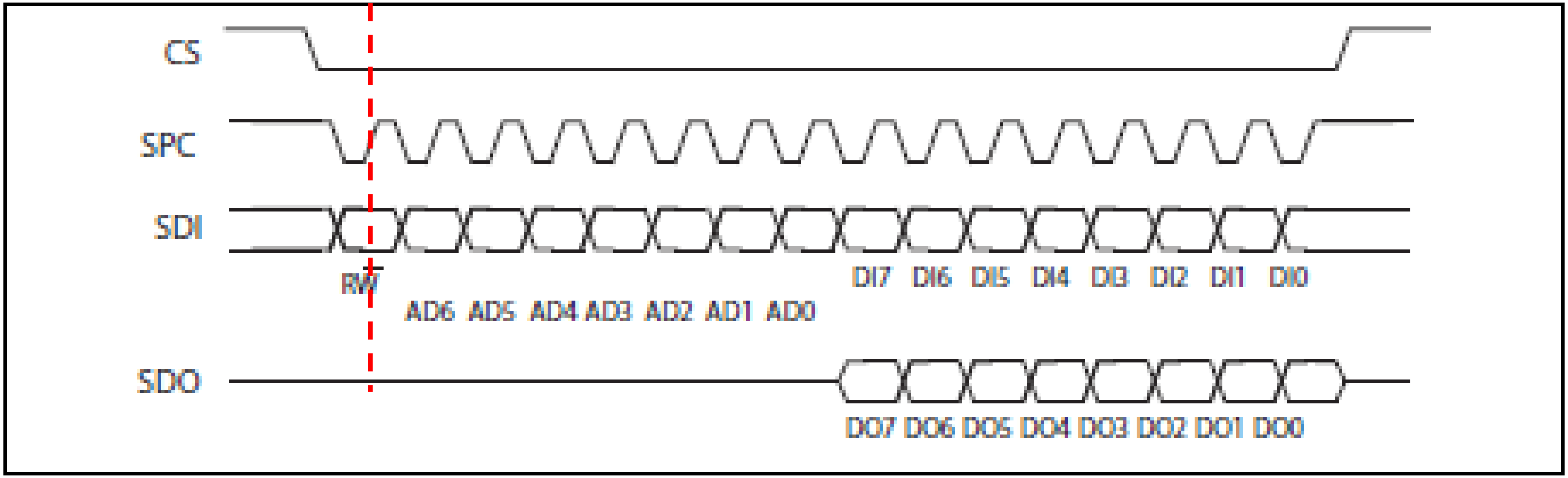
9. 부록 : 3-AXIS accelerometer (LIS2HH12)

9.1 3-AXIS accelerometer SPI

9.1.1. Read and Write protocol

CPOL = 1 (High)
CPHA = 1 (2edge)

- SPI read (MSB first)


CS
SPC
SDI
RW
AD6 AD5 AD4 AD3 AD2 AD1 AD0
DI7 DI6 DI5 DI4 DI3 DI2 DI1 DI0
SDO
DO7 DO6 DO5 DO4 DO3 DO2 DO1 DO0

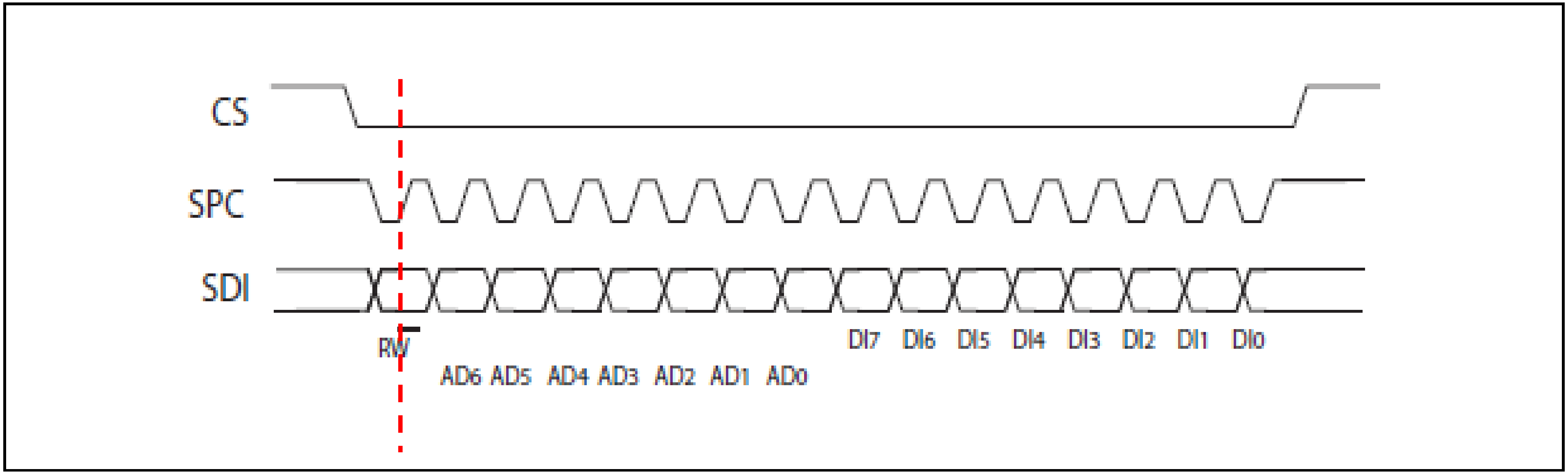
- SPI 읽기 명령(Master에서 보낸 read 명령)은 16 clock pulse로 수행

-bit 0 : R/W bit = 1 (read)

-bit 1-7 : AD(6:0) (Address)

-bit 8-15 : DO(7:0) (Date Out)

- **SPI write (MSB first)**



- **SPI 쓰기 명령 (Master에서 보낸 write 명령) 은 16 clock pulse로 수행**

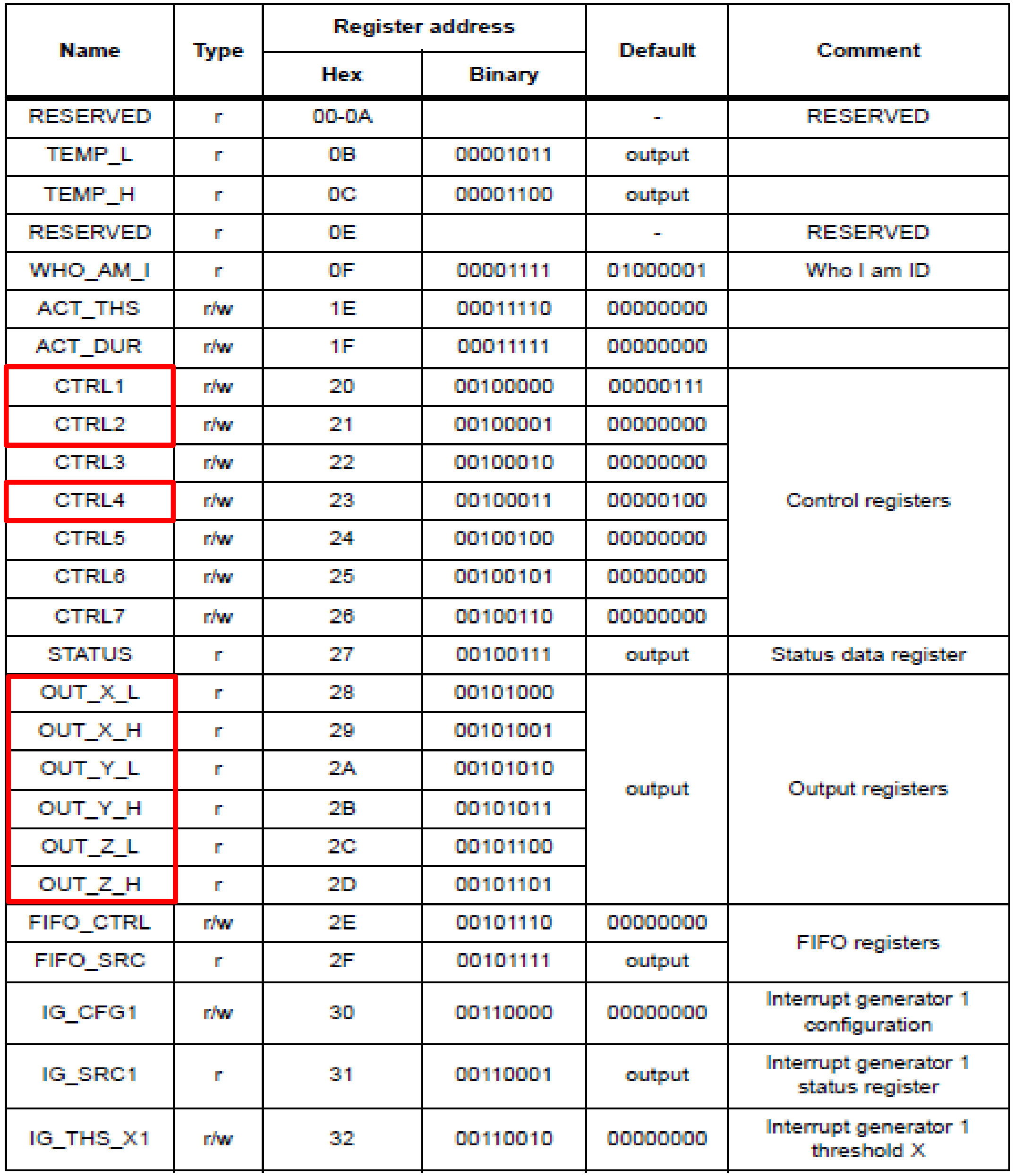
-bit 0 : R/W bit = 0 (write)

-bit 1-7 : AD(6:0) (Address)

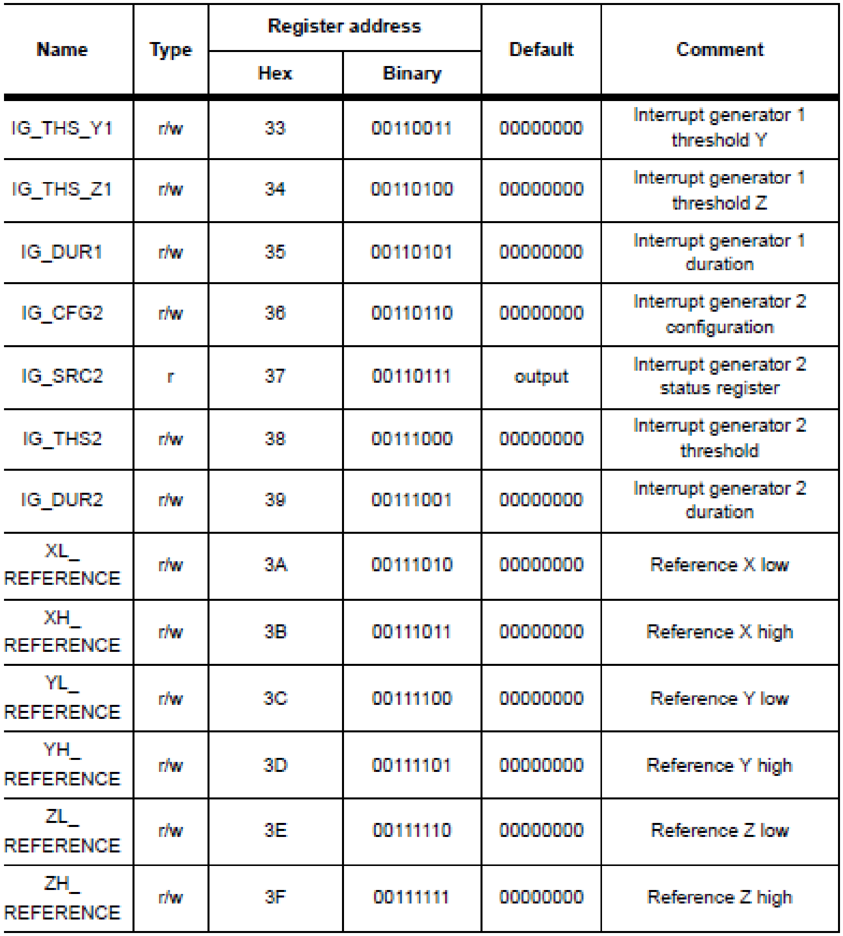
-bit 8-15 : DI(7:0) (Date In)

9.2 레지스터

9.2.1 레지스터 맵

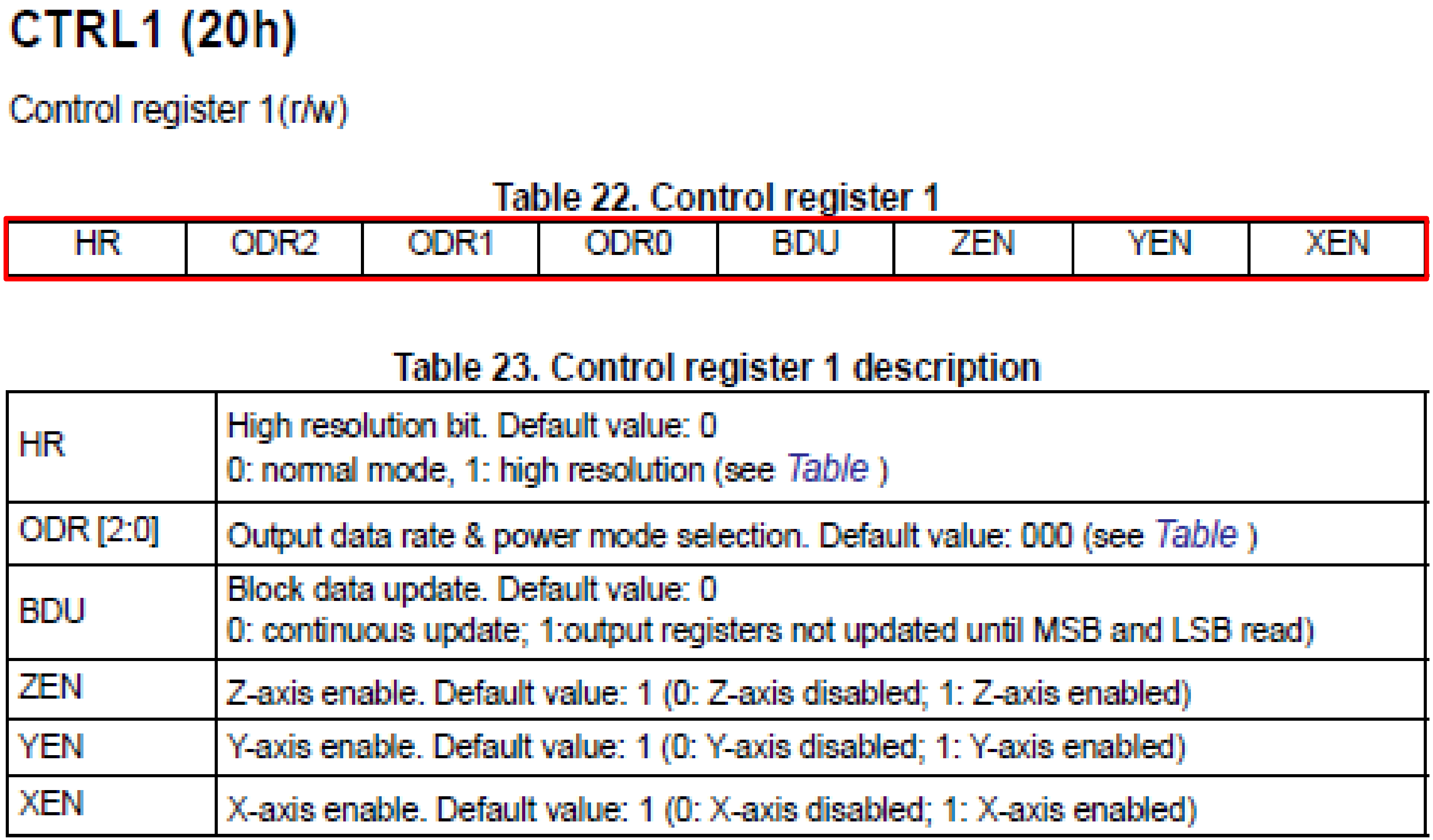


Name	Type	Register address		Default	Comment
		Hex	Binary		
RESERVED	r	00-0A		-	RESERVED
TEMP_L	r	0B	00001011	output	
TEMP_H	r	0C	00001100	output	
RESERVED	r	0E		-	RESERVED
WHO_AM_I	r	0F	00001111	01000001	Who I am ID
ACT_THS	r/w	1E	00011110	00000000	
ACT_DUR	r/w	1F	00011111	00000000	
CTRL1	r/w	20	00100000	00000111	Control registers
CTRL2	r/w	21	00100001	00000000	
CTRL3	r/w	22	00100010	00000000	
CTRL4	r/w	23	00100011	00000100	
CTRL5	r/w	24	00100100	00000000	
CTRL6	r/w	25	00100101	00000000	
CTRL7	r/w	26	00100110	00000000	
STATUS	r	27	00100111	output	Status data register
OUT_X_L	r	28	00101000	output	Output registers
OUT_X_H	r	29	00101001		
OUT_Y_L	r	2A	00101010		
OUT_Y_H	r	2B	00101011		
OUT_Z_L	r	2C	00101100		
OUT_Z_H	r	2D	00101101		
FIFO_CTRL	r/w	2E	00101110	00000000	FIFO registers
FIFO_SRC	r	2F	00101111	output	
IG_CFG1	r/w	30	00110000	00000000	Interrupt generator 1 configuration
IG_SRC1	r	31	00110001	output	Interrupt generator 1 status register
IG_THS_X1	r/w	32	00110010	00000000	Interrupt generator 1 threshold X



Name	Type	Register address		Default	Comment
		Hex	Binary		
IG_THS_Y1	r/w	33	00110011	00000000	Interrupt generator 1 threshold Y
IG_THS_Z1	r/w	34	00110100	00000000	Interrupt generator 1 threshold Z
IG_DUR1	r/w	35	00110101	00000000	Interrupt generator 1 duration
IG_CFG2	r/w	36	00110110	00000000	Interrupt generator 2 configuration
IG_SRC2	r	37	00110111	output	Interrupt generator 2 status register
IG_THS2	r/w	38	00111000	00000000	Interrupt generator 2 threshold
IG_DUR2	r/w	39	00111001	00000000	Interrupt generator 2 duration
XL_ REFERENCE	r/w	3A	00111010	00000000	Reference X low
XH_ REFERENCE	r/w	3B	00111011	00000000	Reference X high
YL_ REFERENCE	r/w	3C	00111100	00000000	Reference Y low
YH_ REFERENCE	r/w	3D	00111101	00000000	Reference Y high
ZL_ REFERENCE	r/w	3E	00111110	00000000	Reference Z low
ZH_ REFERENCE	r/w	3F	00111111	00000000	Reference Z high

9.2.2 레지스터 설정



CTRL1 (20h)

Control register 1(r/w)

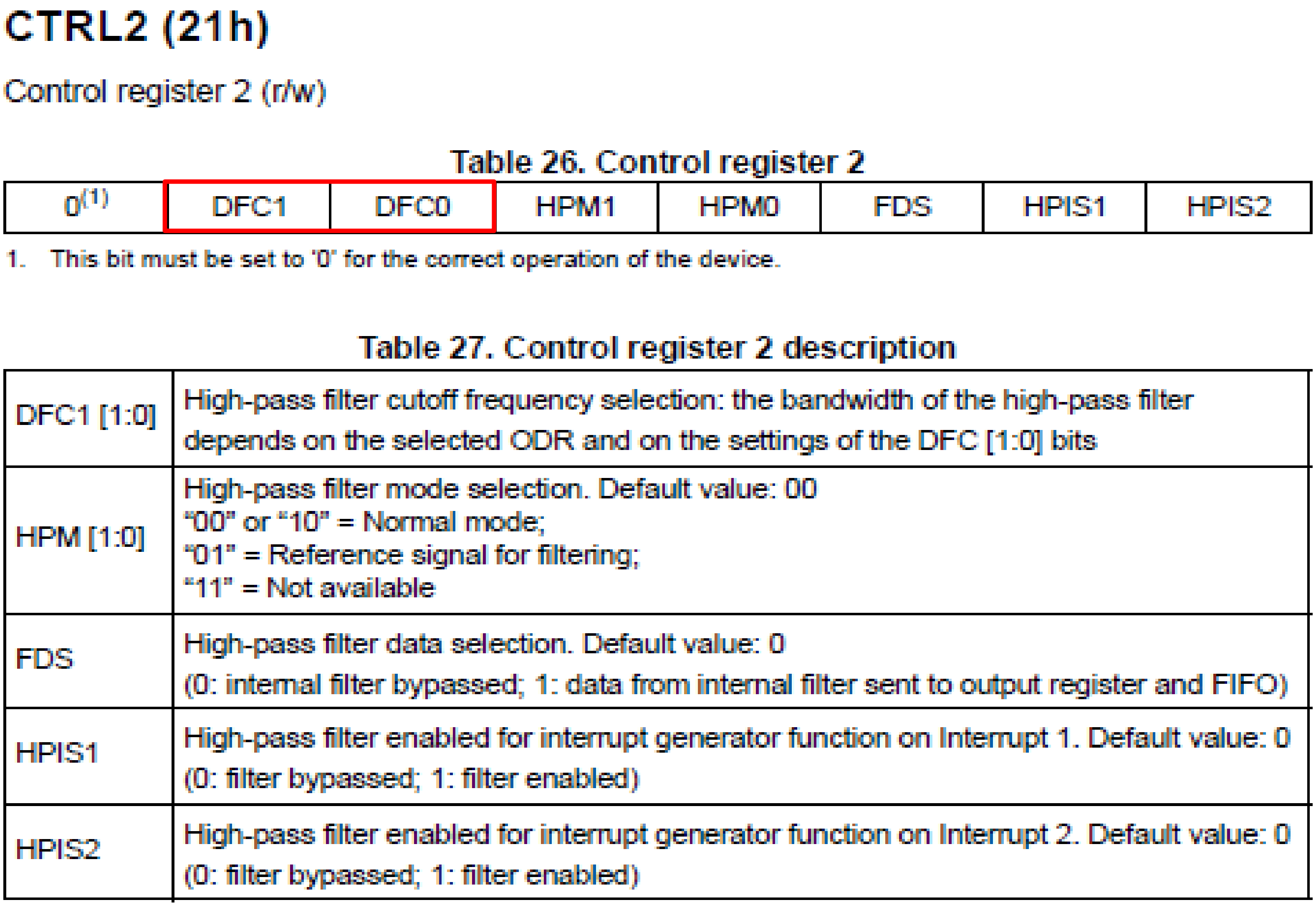
Table 22. Control register 1

HR	ODR2	ODR1	ODR0	BDU	ZEN	YEN	XEN

Table 23. Control register 1 description

HR	High resolution bit. Default value: 0 0: normal mode, 1: high resolution (see *Table*)
ODR [2:0]	Output data rate & power mode selection. Default value: 000 (see *Table*)
BDU	Block data update. Default value: 0 0: continuous update; 1:output registers not updated until MSB and LSB read)
ZEN	Z-axis enable. Default value: 1 (0: Z-axis disabled; 1: Z-axis enabled)
YEN	Y-axis enable. Default value: 1 (0: Y-axis disabled; 1: Y-axis enabled)
XEN	X-axis enable. Default value: 1 (0: X-axis disabled; 1: X-axis enabled)

- **HR : high resolution (Value:1)**
- **ODR[2:0] : 800Hz (Value:110)**
- **ZEN : Z-axis enabled (Value:1)**
- **YEN : Y-axis enabled (Value:1)**
- **XEN : X-axis enabled (Value:1)**



CTRL2 (21h)

Control register 2 (r/w)

Table 26. Control register 2

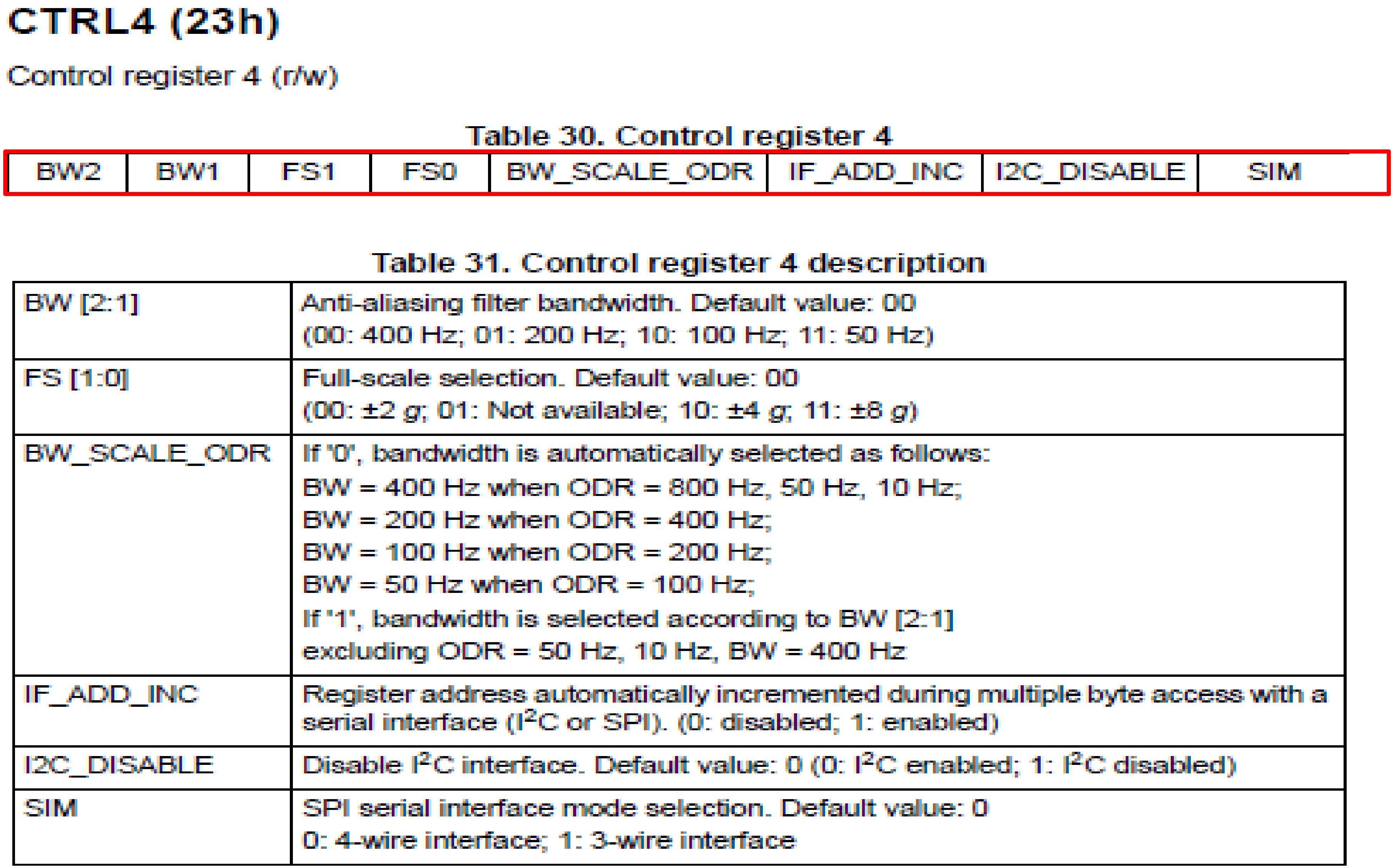
0[1]	DFC1	DFC0	HPM1	HPM0	FDS	HPIS1	HPIS2

[1] This bit must be set to '0' for the correct operation of the device.

Table 27. Control register 2 description

DFC1 [1:0]	High-pass filter cutoff frequency selection: the bandwidth of the high-pass filter depends on the selected ODR and on the settings of the DFC [1:0] bits
HPM [1:0]	High-pass filter mode selection. Default value: 00 "00" or "10" = Normal mode; "01" = Reference signal for filtering; "11" = Not available
FDS	High-pass filter data selection. Default value: 0 (0: internal filter bypassed; 1: data from internal filter sent to output register and FIFO)
HPIS1	High-pass filter enabled for interrupt generator function on Interrupt 1. Default value: 0 (0: filter bypassed; 1: filter enabled)
HPIS2	High-pass filter enabled for interrupt generator function on Interrupt 2. Default value: 0 (0: filter bypassed; 1: filter enabled)

- **DFC[1:0] : ODR 9 (Value:10)**



CTRL4 (23h)

Control register 4 (r/w)

Table 30. Control register 4

BW2	BW1	FS1	FS0	BW_SCALE_ODR	IF_ADD_INC	I2C_DISABLE	SIM

Table 31. Control register 4 description

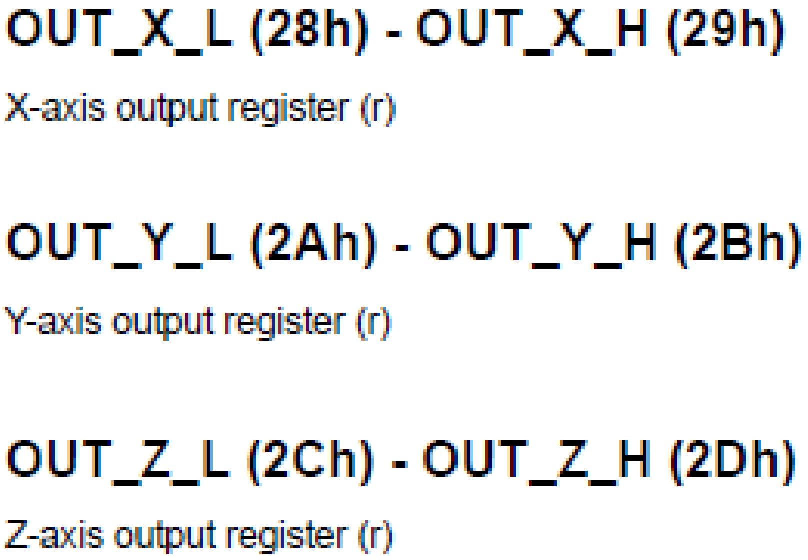
BW [2:1]	Anti-aliasing filter bandwidth. Default value: 00 (00: 400 Hz; 01: 200 Hz; 10: 100 Hz; 11: 50 Hz)
FS [1:0]	Full-scale selection. Default value: 00 (00: ±2 *g*; 01: Not available; 10: ±4 *g*; 11: ±8 *g*)

BW_SCALE_ODR	If '0', bandwidth is automatically selected as follows:
	BW = 400 Hz when ODR = 800 Hz, 50 Hz, 10 Hz;
	BW = 200 Hz when ODR = 400 Hz;
	BW = 100 Hz when ODR = 200 Hz;
	BW = 50 Hz when ODR = 100 Hz;
	If '1', bandwidth is selected according to BW [2:1]
	excluding ODR = 50 Hz, 10 Hz, BW = 400 Hz

IF_ADD_INC	Register address automatically incremented during multiple byte access with a serial interface (I^2C or SPI). (0: disabled; 1: enabled)
I2C_DISABLE	Disable I^2C interface. Default value: 0 (0: I^2C enabled; 1: I^2C disabled)
SIM	SPI serial interface mode selection. Default value: 0 0: 4-wire interface; 1: 3-wire interface

- **BW[2:1] : 400Hz (Value:00)**
- **FS : +/-2g (Value:00)**
- **BW_SCALE_ODR : automatically(Value:0)**
- **IF_ADD_INC : enabled (Value:1)**
- **I2C_DISABLE : Disabled (Value:1)**
- **SIM : 4-wire interface (Value:0)**

- **OUTPUT Resister**



OUT_X_L (28h) - OUT_X_H (29h)

X-axis output register (r)

OUT_Y_L (2Ah) - OUT_Y_H (2Bh)

Y-axis output register (r)

OUT_Z_L (2Ch) - OUT_Z_H (2Dh)

Z-axis output register (r)