

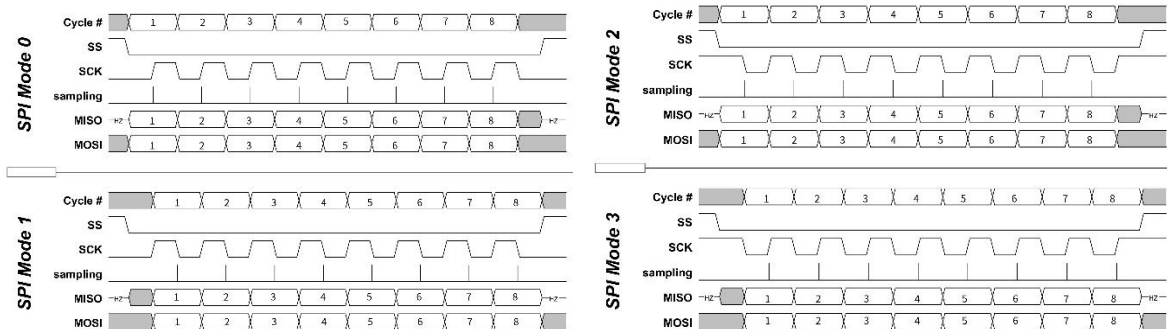
SPI(Serial Peripheral Interface)

1. SPI Overview

SPI는 네 개의 선을 사용하는 직렬 통신 프로토콜로 마이크로컨트롤러와 주변기기(센서, 메모리, 다른 MCU) 간 Full-Duplex 통신을 가능하게 한다. 고속 데이터 전송 인터페이스로, 클럭 동기 방식으로 전송 속도가 빠르고 신뢰성이 높다.

MOSI(Master Out Slave In), MISO(Master In Slave Out), SCLK(Serial Clock), SS/CS(Slave Select/Chip Select) 라인이 기본이다. 마스터가 클럭을 생성하며, Full-Duplex 데이터 전송을 지원하여, 데이터 송수신이 가능하다. 하나의 마스터에 여러 슬레이브를 연결할 수 있지만, 각 슬레이브는 개별 SS/CS 라인으로 선택된다. 설계 시 다중 슬레이브를 연결할 경우 SS 라인 관리가 필요하다.

2. SPI Data Transfer Modes



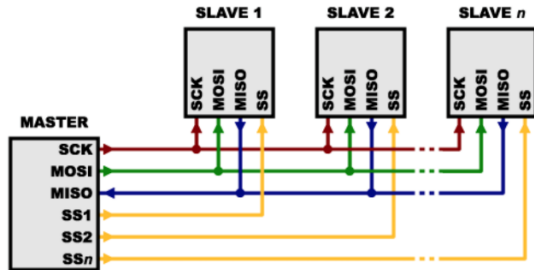
SPI 는 마스터의 CPOL(Clock Polarity)과 CPHA(Clock Phase)에 따라 4 개의 동작 모드로 구분되며, 각 모드는 데이터 샘플링 시점 및 클럭 신호의 기본 상태에 영향을 준다.

구분	값 0	값 1
CPOL (Clock Polarity)	클럭 IDLE 상태 LOW	클럭 IDLE 상태 HIGH
CPHA (Clock Phase)	첫 클럭 edge에서 샘플링	두 번째 클럭 edge에서 샘플링

모드	CPOL	CPHA	클럭 유휴 상태	샘플링 엣지	데이터 전환 엣지
Mode 0	0	0	SCK = Low	첫 번째 rising edge	첫 번째 falling edge
Mode 1	0	1	SCK = Low	첫 번째 falling edge	첫 번째 rising edge
Mode 2	1	0	SCK = High	첫 번째 falling edge	첫 번째 rising edge
Mode 3	1	1	SCK = High	첫 번째 rising edge	첫 번째 falling edge

3. SPI Protocol

3.1 데이터 전송 시퀀스



SPI에서 데이터 전송은 다음과 같은 순서로 진행된다.

- ① 마스터는 특정 슬레이브를 선택하기 위해 SS/CS 신호를 Low로 내린다. 이때 선택되지 않은 슬레이브들은 Tri-state 상태로 버스에서 제외된다.
- ② 마스터는 SCLK 클럭을 발생시키며, MOSI 라인에 송신할 데이터를 출력한다.
- ③ 슬레이브는 들어오는 SCLK 신호의 엣지에 맞춰 MOSI 데이터를 샘플링하고, 동시에 MISO 라인에 응답 데이터를 출력한다.
- ④ 데이터 프레임 전송이 끝나면, 마스터는 SS/CS 신호를 High로 올려 슬레이브를 비활성화한다.

3.2 다중 슬레이브 제어

하나의 마스터가 여러 슬레이브를 제어하는 경우, 각 슬레이브는 고유한 SS/CS 라인을 가진다.

마스터는 원하는 슬레이브의 SS/CS 신호만 Low로 설정해 해당 슬레이브를 활성화하고, 나머지 슬레이브들은 High 상태로 유지해 통신에 참여하지 않도록 한다.

다중 슬레이브 환경에서는 SS/CS 라인의 관리가 중요하며, 동시에 여러 슬레이브를 활성화할 경우 버스 충돌이 발생할 수 있으므로 반드시 주의해야 한다.

3.3 전송 타이밍 개요

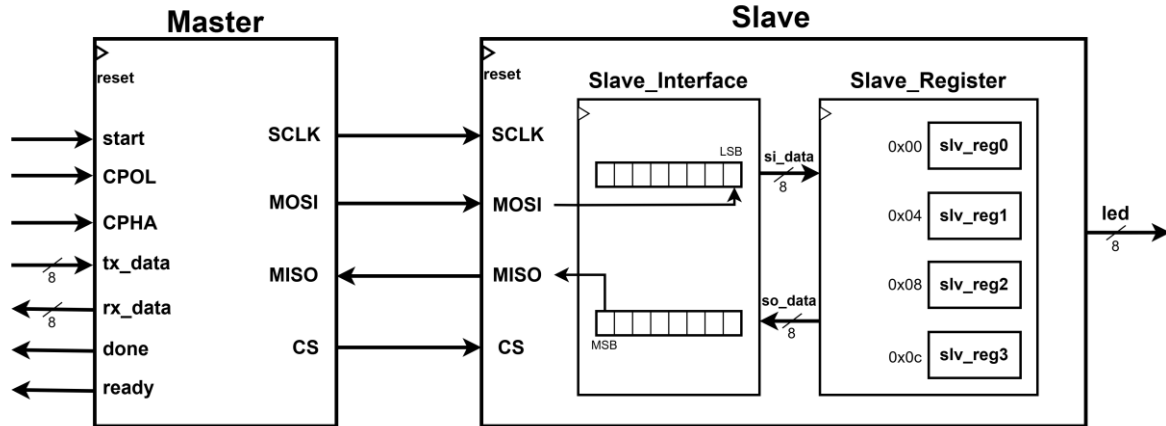
SPI 전송은 클럭 신호의 특정 엣지에서 데이터가 유효화되고 샘플링되는 구조로 이루어진다.

마스터는 출력 데이터를 설정한 후 클럭 엣지에서 슬레이브가 해당 데이터를 안정적으로 읽을 수 있도록 타이밍을 보장해야 한다.

슬레이브는 클럭 엣지에서 데이터를 읽어들이고, 응답 데이터는 클럭 주기 내 MISO 라인을 통해 출력된다.

4. SPI Interface Hardware Design

4.1 Signal Description



Signal	Source	Width	Description
clk	System bus clock	1	Clock
reset	System bus reset	1	Reset.
CPOL	AXI4- Lite SPI Intf CR Register	1	Control Sig CPOL, CPHA 조합에 따라 모드가 결정된다. start 신호에 따라 SPI Master의 데이터 전송을 시작한다. {CPOL, CPHA} - 00: mode 0 - 01: mode 1 - 10: mode 2 - 11: mode 3
CPHA	AXI4- Lite SPI Intf CR Register	1	
start	AXI4- Lite SPI Intf CR Register	1	
tx_data	AXI4- Lite SPI Intf SOD Register	8	Tx Data. SPI Master로 전송할 8bit data
rx_data	Master	8	Rx Data. SPI Master가 수신한 8bit data
ready	Master	1	Ready - 통신 중이 아닐 때: SPI가 통신 가능할 때 high
done	Master	1	done SPI Master의 8bit 송, 수신이 끝나면 high
SCLK	Master	1	Serial Clock SPI Master에서 만든 Serial Clock
MOSI	Master	1	Master Out Slave in Tx data를 1bit 씩 left shift 하여 출력된 1bit data
MISO	Slave	1	Slave Out Master in

			SPI Slave Register에서 출력된 so data를 1bit 씩 left shift 하여 출력된 1bit data
CS	Master	1	Chip Select SPI Master에서 1개의 Slave를 선택하기 위한 1bit data(Active Low)

4.2 Register Map

offset	Register	31...8	7	6	5	4	3	2	1	0
0x00	CR	Reserved						start	cpha	cpol
0x04	SOD	Reserved	master_out_data[7:0]							
0x08	SID	Reserved	master_in_data[7:0]							
0x0C	SR	Reserved							done	ready

4.2.1 Control Register (CR)

Address offset: 0x00

Reset value: 0x0000 0000

31...3	2	1	0
Res	start	CPHA	CPOL
	rw	rw	rw

Bit 2 **start**: start bit

SPI 통신 시작 시 설정

Bit 1 **CPHA**: mode bit

데이터의 sampling, transfer 위치 설정

Bit 0 **CPOL**: mode bit

SCLK의 시작 데이터 설정

Start에 의해 동작이 시작되면 CPOL과 CPHA에 의해 동작 방식이 결정된다.

4.2.2 Serial Out Data (SOD)

Address offset: 0x04

Reset value: 0x0000 0000

31...3	7...0
Res	master out data[7:0]
	rw

Bits [7:0] **mo_data**: master out data

슬레이브로부터의 데이터 수신이 완료되었음을 나타내는 플래그

수신 완료 시 1 SCL 클럭 주기 동안 High(1)로 설정되며,

읽기 동작 수행 시 소프트웨어에서 이를 감지하여 적절한 처리를 수행해야 한다.

4.3 Top-Level Architecture

4.4 Internal Clock Generator 설계 및 최적화 효과

4.5 SPI 동작 제어 구조