Xilinx Zynq FPGA, TI DSP, MCU 프로그래밍 및 회로 설계 전문가 과정

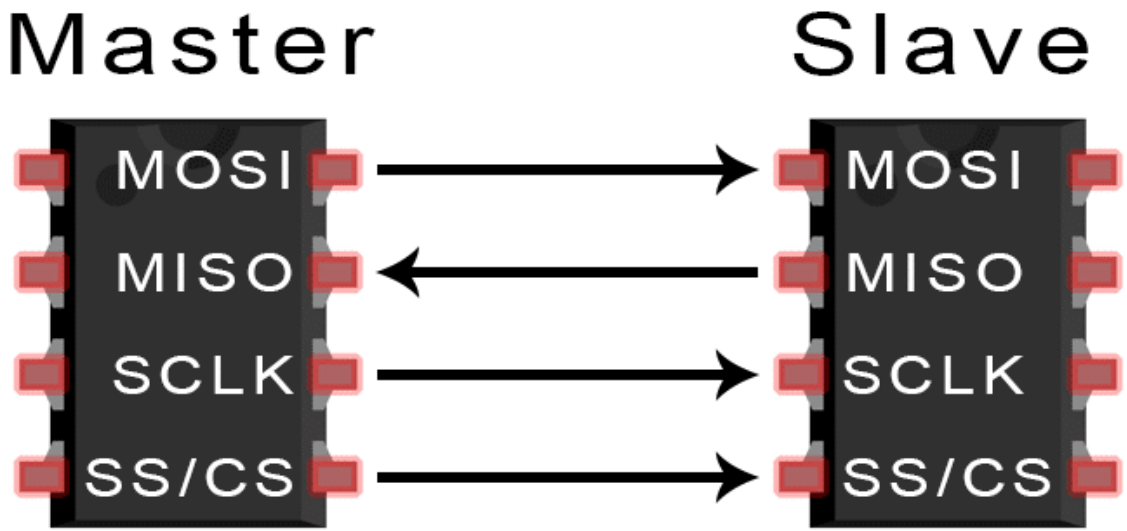
강사 – Innova Lee(이상훈)
gcccompil3r@gmail.com

SPI(Serial Peripheral Interface)

SPI(Serial Peripheral Interface)

SPI 는 주변 장치 연결을 위한 Serial Interface 중 하나다.
UART 의 경우 Full-Duplex(전 이중) 방식의 Serial 통신이지만 일대일 통신만 가능하다는 한계가 있다.
반면 SPI 는 전이중 방식의 Serial 통신이라는 부분은 동일하지만
Master Device 하나에 Slave Device 여러 개가 연결된 일대다 통신이 가능하다.

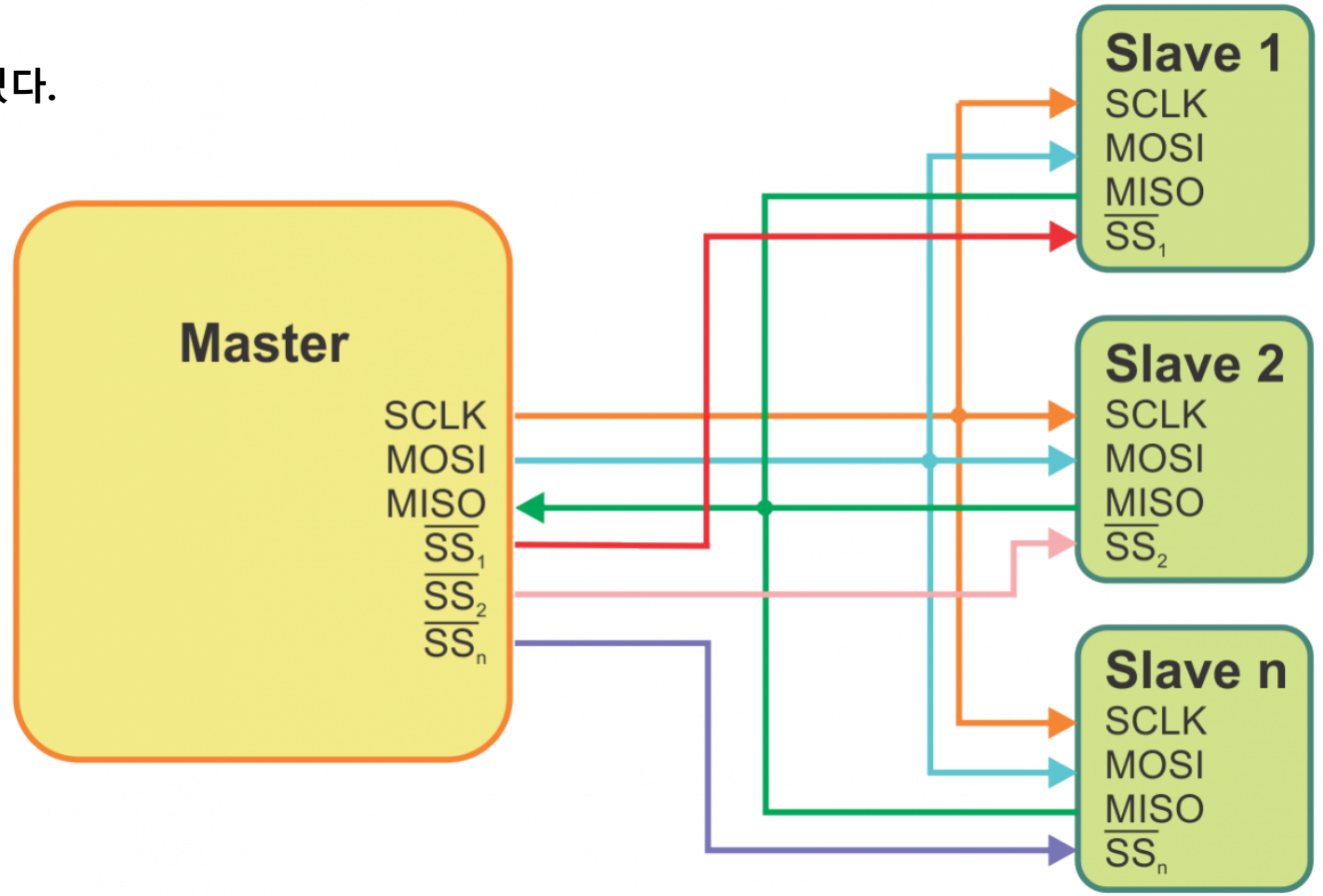
Master Device 는 연결을 시작하고 통신을 제어하는 책임을 지는 Device 이며
연결이 성립된 이후에는 2 개의 Data Line 을 통해 전이중 방식으로 Data 를 동시에 주고 받을 수 있다.
Master Device 에 하나의 Slave Device 가 연결된 경우는 아래와 같다.



SPI 연결을 4 개의 선을 연결해야 한다.
SCK(Serial Clock) 혹은 SCLK 로 표현하는 Clock 을 사용한다.
이 Clock 으로 SPI 가 동기 전송 방식을 사용할 수 있다.

MOSI 는 Master Out Slave In 의 약어로 Master Device 에서 Slave Device 로 데이터를 전송하기 위해 사용되며
MISO 는 Master In Slave Out 의 약어로 Slave Device 에서 Master Device 로 데이터 전송을 위해 사용된다.
SS 는 Slave Select 의 약어로 여러 개의 Slave 장치 중
Master Device 가 Data 를 주고받을 Slave Device 를 선택하기 위해 사용한다.

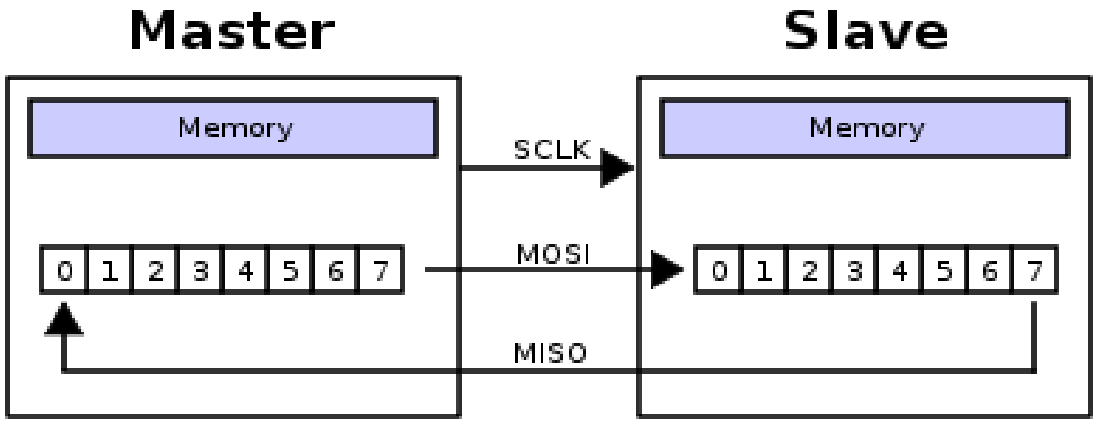
선택되지 않은 Slave Device 의 SS 는 LOW 상태에 있다.
여러 개의 Slave Device 가 연결된 경우 명확하게 확인할 수 있다.



하나의 Master Device 에 3 개의 Slave Device 가 연결된 것을 볼 수 있다.
모든 Slave Device 는 Master 로부터 SCLK, MOSI, MISO 라인을 공유하고 있다.
즉 Master Device 의 Data 가 모든 Slave Device 로 전달된다.
누가 데이터를 받을 것인가는 SS 에 의해 결정된다.

Master 로부터 Data 를 받을 Slave Device 의 SS 만 LOW 상태에 있고
나머지 Slave Device 의 SS 는 HIGH 상태에 있게 된다.
즉 일대다 연결이 가능하지만 특정 순간에는 일대일 통신을 수행한다는 것이다.
또한 Slave Device 의 수가 늘어날수록 전용의 SS Line 은 Slave Device 의 수와 동일하게 증가한다.

SPI 의 데이터 전송은 송신과 수신이 항상 동시에 일어난다.
UART 에서도 송신과 수신이 동시에 일어날 수 있지만 항상 그렇진 않다.



SPI 는 동기 방식으로 동작하므로 동기화를 위해 클록이 필요하고
이 클록은 Master Device 에 의해 공급된다.
Slave Device 에서 Master Device 로 데이터를 보내고자 하는 경우에도
Slave Device 는 Master Device 의 클록을 기준으로
데이터를 전송해야 하지만 Master Device 의 클록을 알기 어렵다.

SPI 에서 이러한 동기화의 문제점을
송신과 수신이 항상 동시에 진행되도록 함으로써 해결하고 있다.
SPI 의 Master 와 Slave Device 의 Data Buffer 는 Circular Queue 를 이루고 있다.
Master Device 에서 Slave Device 로 보낼 Data 가 Master Device Queue 에 저장되어 있고
Slave Device 에서 Master Device 로 보낼 Data 가 Slave Device 의 Queue 에 저장되어 있다.

Data 전송은 전적으로 Master Device 가 책임지므로 Master Device 가 제공하는 Clock 에 의해 Data 전송이 수행된다.
데어탁 준비된 상태에서 Clock 이 발생하여 Master Device 의 1 bit 데이터가 MOSI 를 통해 Slave Device 에 전달되며
동시에 Slave Device 의 1 bit 데이터가 MISO 를 통해 Master Device 에 전달된다.
전달된 이후엔 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 형태로 되어있던것이 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 형태가 된다.

자료 구조 시간에 Circular Queue 를 다루는 것과 동일하지만
차이가 있다면 Local 방식이 아닌 Network 방식의 원격이라는 것이다.

8 개의 Clock 이 발생하면 Master Device 의 1 byte 데이터가 Slave Device 로 전달되고
Slave Device 의 1 byte Data 는 Master Device 로 전달되어 1 byte 의 Data 가 교환되는 결과를 얻는다.

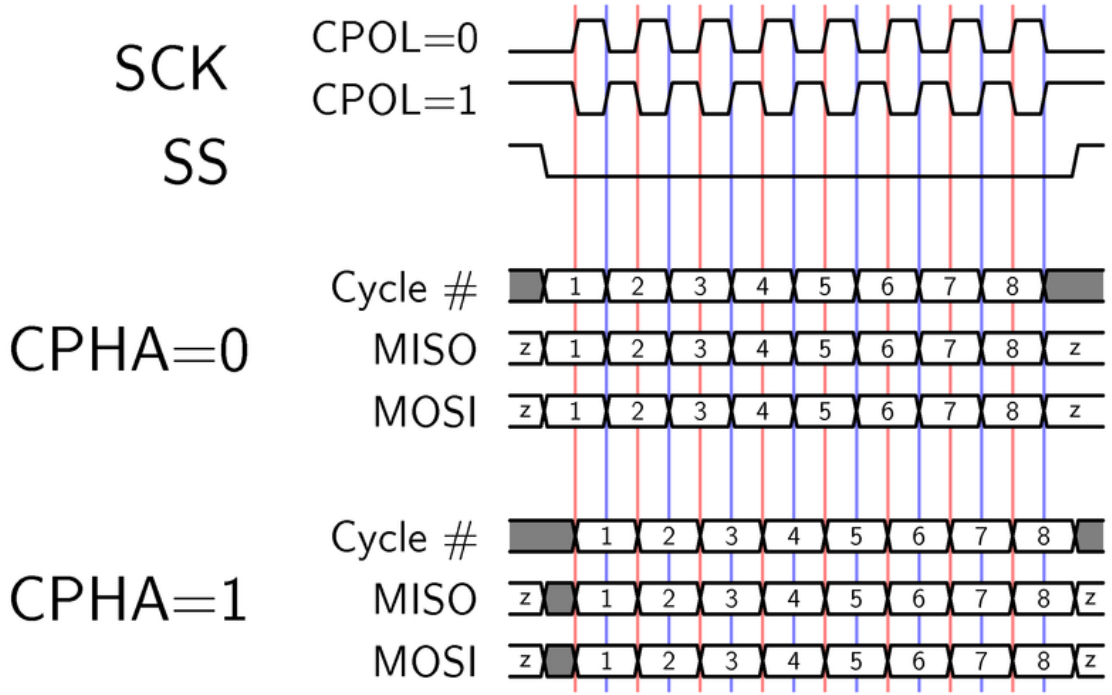
Master Device 와 Slave Device 는 항상 데이터를 동시에 주고받음으로써 동기화 문제를 해결하고 있다.
이러한 상황에서 유의할 점은 Clock 의 역할이다.

기본적으로 Clock 은 Data 가 전달되는 시점을 알려주고 전달된 Data 가 안정적인 상태를 유지할 때
수신된 Device 쪽에서 Data 를 Sampling 하게 만들어주는데 사용된다.
하지만 Clock 의 어느 부분에서 Data 전송이 완료되었는지에 따라 수신 장치쪽에서 Data 를 읽는 시점이 달라진다.
이를 위해 Clock 의 극성(Polarity, CPOL) 과 Clock 의 위상(Clock Phase, CPHA) 가 사용된다.

CPOL: SPI Bus 가 쉬고 있을 때의 Clock 값을 결정한다.
CPOL = 0 이면 비활성 상태일 때 SCLK 는 LOW 값을 가지고 CPOL = 1 이면 비활성 상태일 때 SCLK 는 HIGH 값을 가진다.

CPHA: Data Sampling 시점을 결정한다.
CPHA = 0 이면 Data 는 비활성 상태에서 활성 상태로 바뀌는 Edge 에서 Sampling 되고
CPHA = 1 이면 Data 는 활성 상태에서 비활성 상태로 바뀌는 Edge 에서 Sampling 된다.

CPHA = 0 인 경우 CPOL = 0 이면 Rising Edge, CPOL = 1 이면 Falling Edge
CPHA = 1 인 경우 CPOL = 0 이면 Falling Edge, CPOL = 1 이면 Rising Edge



수신 장치가 Data Sampling 하는 시점은 Data 가 안정화된 이후, 즉 데이터 비트의 중앙에서 한 번 뿐이다.
UART 의 경우 비동기적으로 데이터가 전송되므로 한 비트의 데이터 확인을 위해 여러 번의 Sampling 을 통해 실제 전송된 데이터를 파악하지만
SPI 의 경우에는 동기화를 위한 클록이 별도로 존재하므로 한 번의 Sampling 만으로 충분하다.

역시 앞서서 설명하였듯이 Data Sampling 동안 SS Line 은 LOW 상태를 유지하고 있다.
즉 선택된 Slave 만 Data 를 수신할 수 있다.

이제 Cortex-R5F 에서 SPI 예제를 수행하며 위 내용들을 확인해보도록 하자!