

LIVE

LEARN

EMBEDDED SYSTEM ARM CORTEX M4

DAY 26/30

(30 DAYS CHALLENGE)

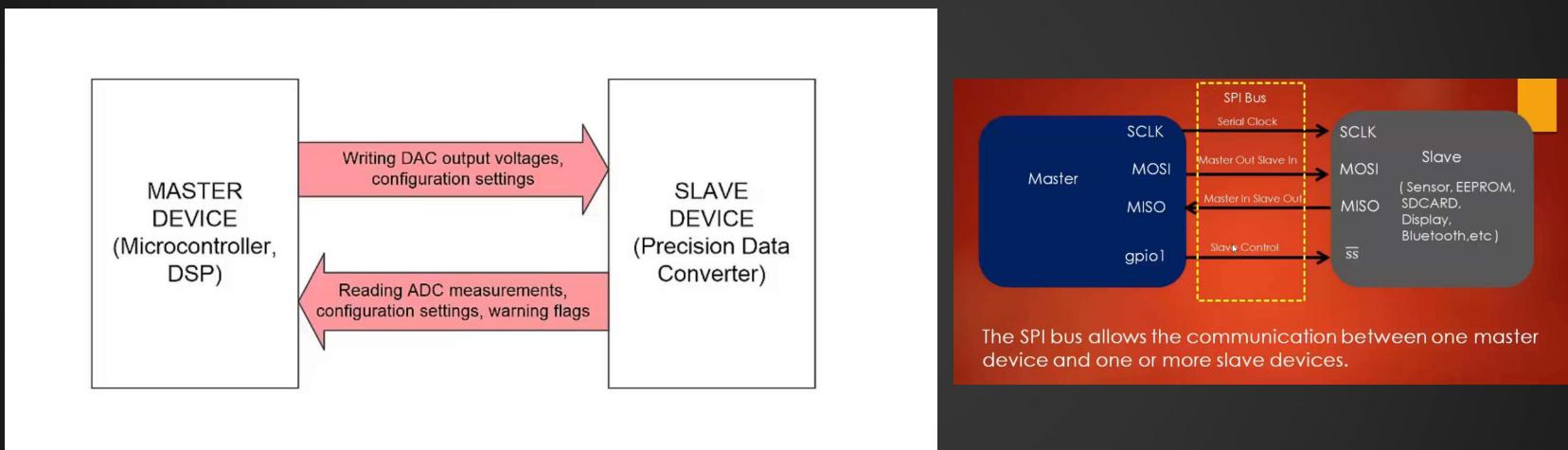
M K JEEVARAJAN



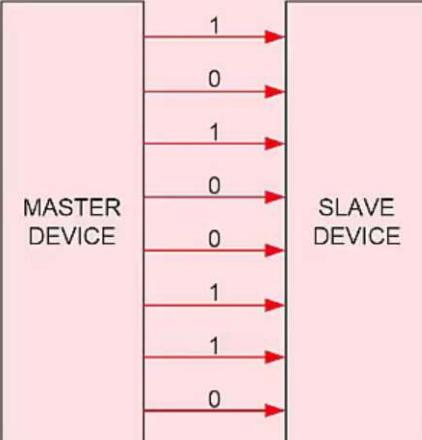
What you will Learn Today?

- ✓ What is SPI
- ✓ HOW SPI WORKS

Introduction to SPI



Parallel Vs Serial

Parallel	Serial
 <p>MASTER DEVICE</p> <p>SLAVE DEVICE</p> <p>Data is sent eight bits or one byte at a time, but it requires eight data lines</p>	 <p>MASTER DEVICE</p> <p>SLAVE DEVICE</p> <p>Data is sent one bit at a time and only requires one line, but it may be slower</p>

SPI Comparison

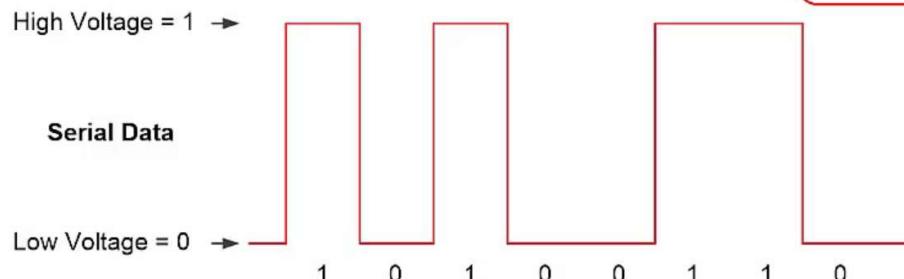
Protocol	Type	Max distance(ft.)	Max Speed (bps)	Typical usage
USB 3.0	dual simplex serial	9 (typical) (up to 49 with 5 hubs)	5 G	Mass storage, video
USB 2.0	half duplex serial	16 (98 ft. with 5 hubs)	1.5M, 12M, 480M	Keyboard, mouse, drive, speakers, printer, camera
Ethernet	serial	1600	10G	network communications
I2C	synchronous serial	18	3.4 M in High-speed mode.	Microcontroller communications

SPI,I2C these peripherals communicate over TTL signaling in the voltage range 0V to 5V where as protocols like CAN, RS485 communicate over differential signaling (> 10V). This is one of the reasons why SPI and I2C are short distance communication interfaces.

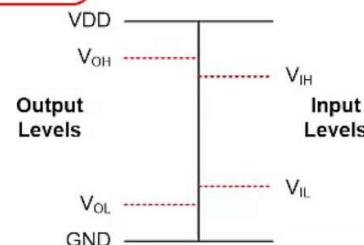
SPI	synchronous serial	10	fPCLK/2	and control systems
SPI	synchronous serial	10	fPCLK/2	SPI

Voltage Level

Voltage Levels

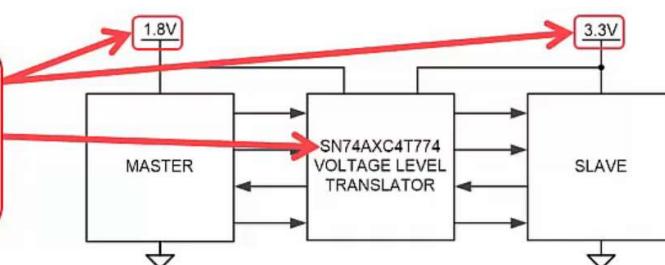


Datasheets will define the output driving range for digital outputs...



...And define the input range for high and low for digital inputs

If the master and slave operate on different supply voltages, a **voltage level translator** may be required



Serial Peripheral Interface(SPI)

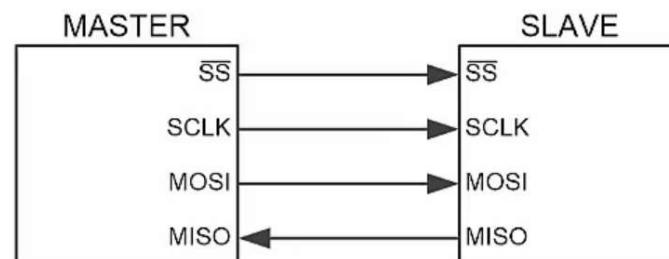
SPI

Master controls the slave select and the serial clock

An SPI bus can have only one master, but may control multiple slaves

Each slave has a slave select for independent control

Data can be transmitted from master to slave or slave to master that may be used as full duplex



SPI Connections (SS)

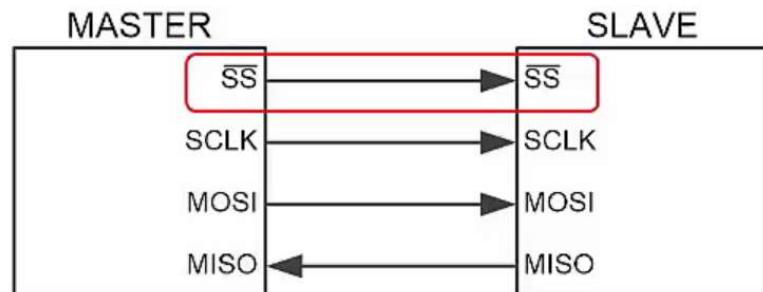
SS

Slave select: Selects the slave device for communication

Is often used as active low, which is often represented by an overbar

Also known as: \overline{SS} , SSEL, \overline{CS} , CS, \overline{SYNC} , nSS, SS#

Commonly labeled as \overline{CS} or \overline{SYNC} in TI data converters



SPI Connections (SCLK)

SCLK

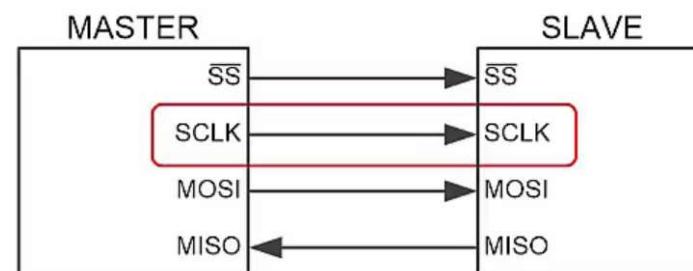
Serial Clock: Synchronizes data transmission between the master and slave

SCLK originates from master and shared with all slaves

Clock may idle high or low

Data is clocked in on either the rising or falling edge of the clock

Also known as: SCK



SPI Connections (MOSI)

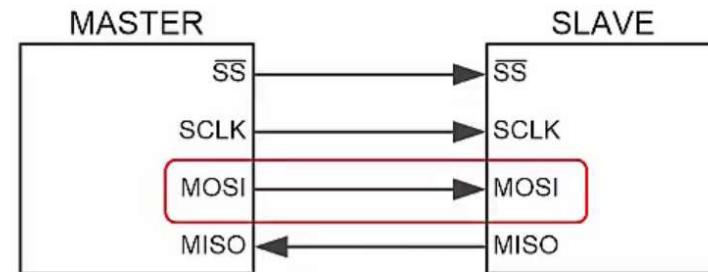
MOSI

Master Out, Slave In: Output from the master used to send data to the slave device

Can be shared between slave devices

Also known as: SIMO, MSTR; from the slave device: SDI, DI, DIN, SI; from the master device: SDO, DO, DOUT, SO

Commonly labeled as DIN in TI data converters



SPI Connections (MISO)

MISO

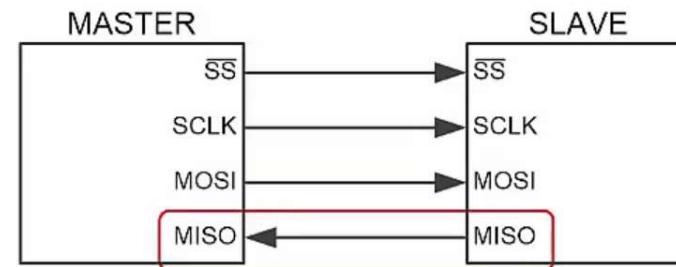
Master In, Slave Out: Output from the slave device used to send data to the master

Can be shared between all slave devices,

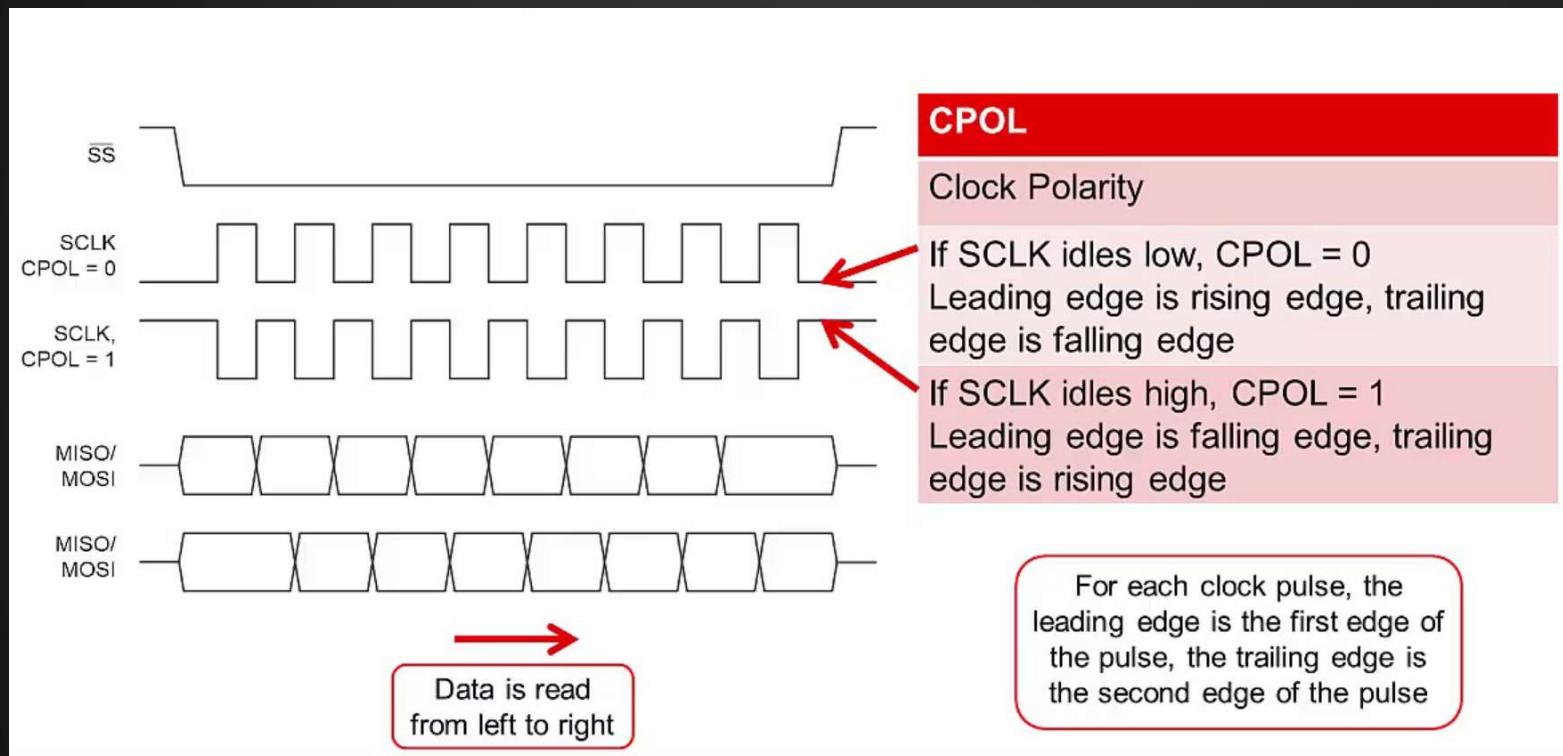
Slave output becomes high impedance when SS is not selected

Also known as: SOMI; from the slave device: SDO, DO, DOUT, SO; to the master device: SDI, DI, DIN, SI

Commonly labeled as DOUT in TI data converters



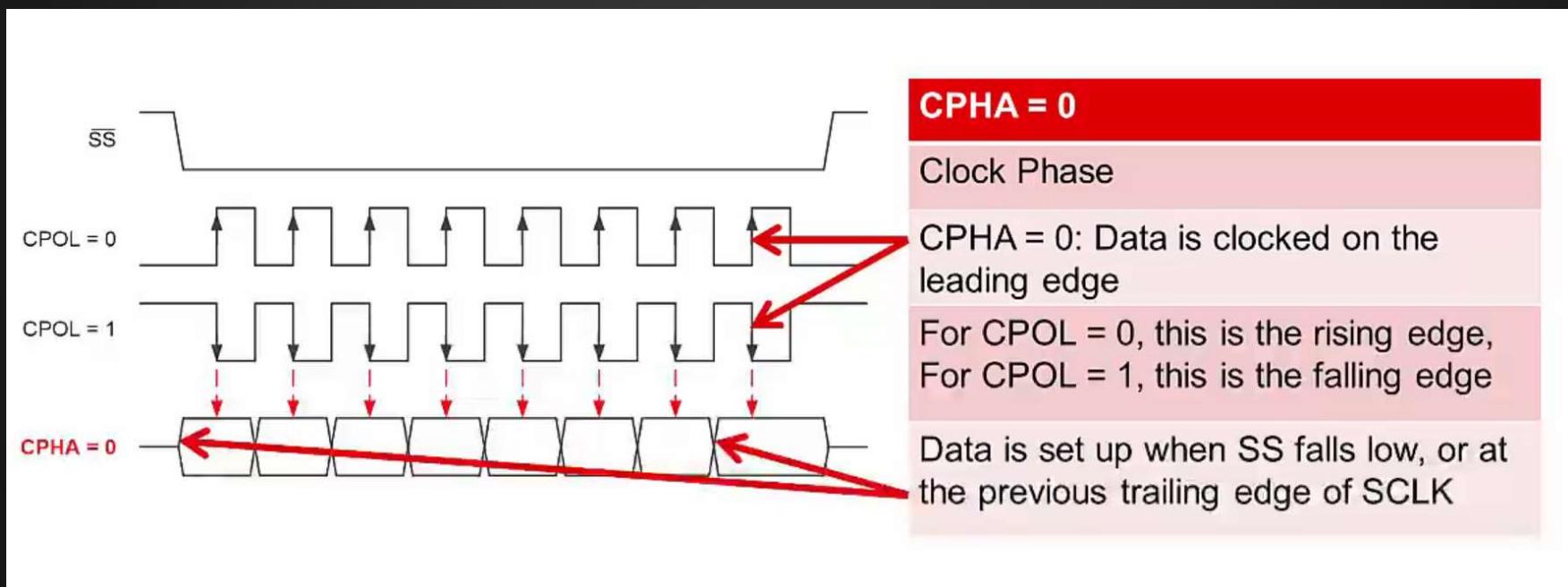
SPI CLOCL POLARITY(CPOL)



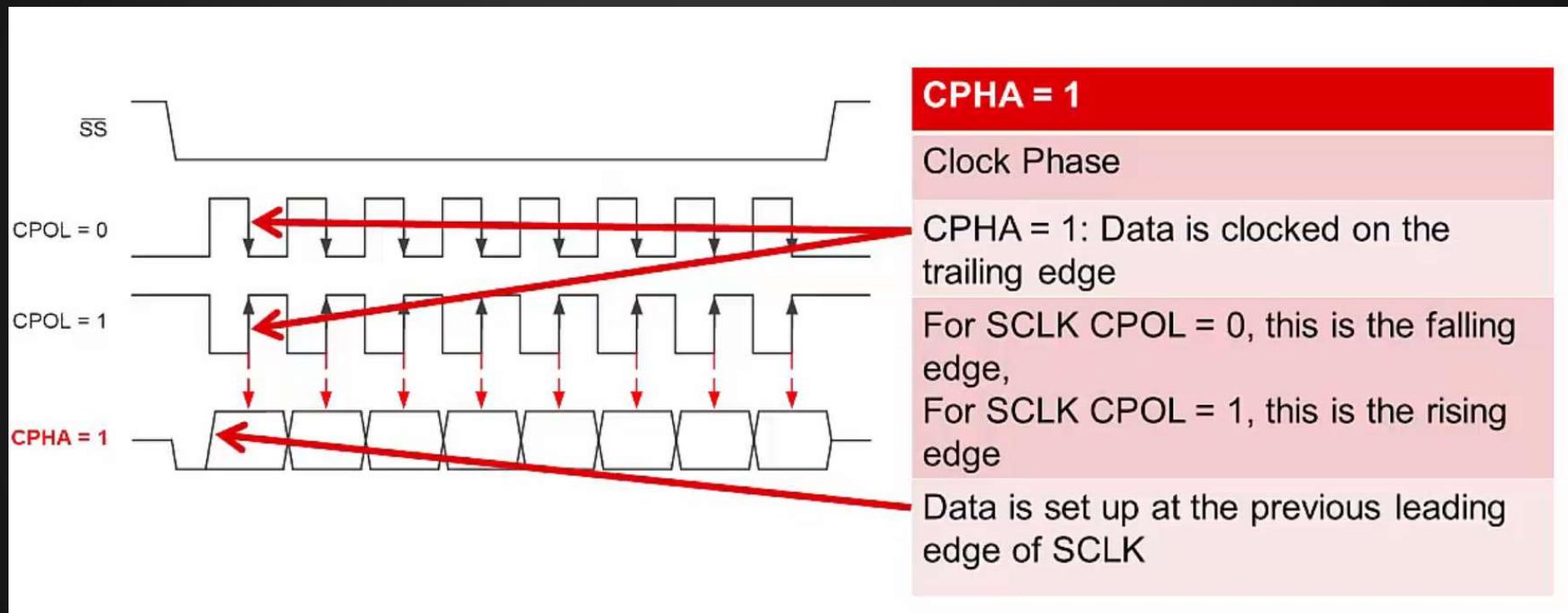
The SPI Protocol -BUS Modes

- **CPOL = 0** Means sampling on the first edge
- **CPOL = 1** Means sampling on the second edge

SPI CLOCK PHASE (CPHA=0)



SPI CLOCK PHASE (CPHA=1)

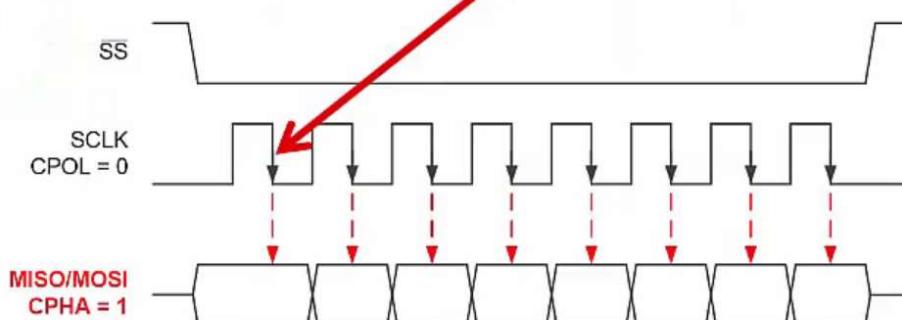


SPI MODE NUMBERS

SPI Mode	CPOL	CPHA
0	0	0
1	0	1
2	1	0
3	1	1

Example:

SPI Mode 1 – SCLK idles low, data is read on the trailing edge of the clock



SPI PROTOCOL BUS MODES

	CPOL	CPHA
Mode0	0	0
Mode1	0	1
Mode2	1	0
Mode3	1	1

- **CPOL = 0**

Active state of clock = 1

Idle state of clock = 0

CPHA = 0

Data is captured on the *rising edge*

Data is output on the *falling edge*

CPHA = 1

 Data is captured on the *falling edge*

Data is output on the *rising edge*

- **CPOL = 1**

Active state of clock = 0

Idle state of clock = 1

CPHA = 0

Data is captured on the *falling edge*

Data is output on the *rising edge*

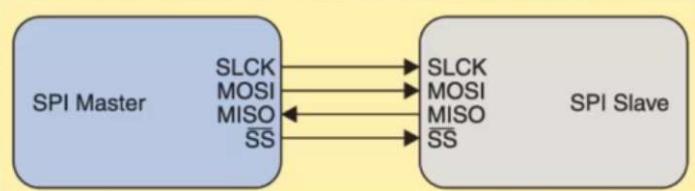
CPHA = 1

Data is captured on the *rising edge*

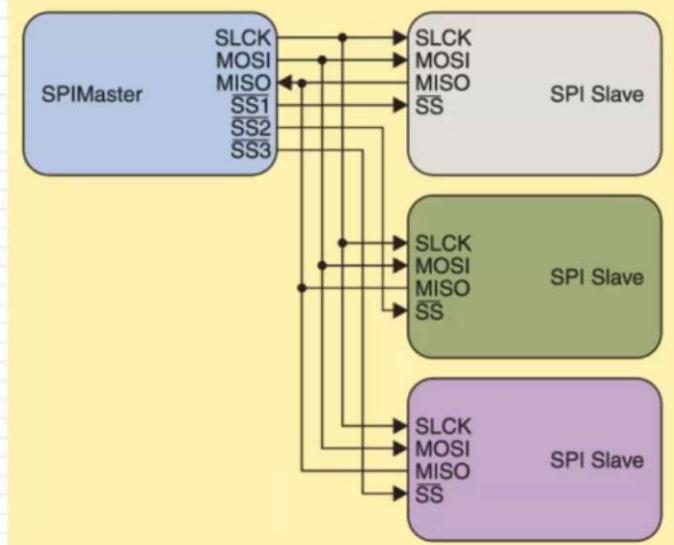
Data is output on the *falling edge*

Single Vs Multiple Slaves

Single Slave



Multiple Slaves



- **SDI** = **MOSI**
- **SDO** = **MISO**
- **SCLK** = **SCK**
- **CE** = **SS**

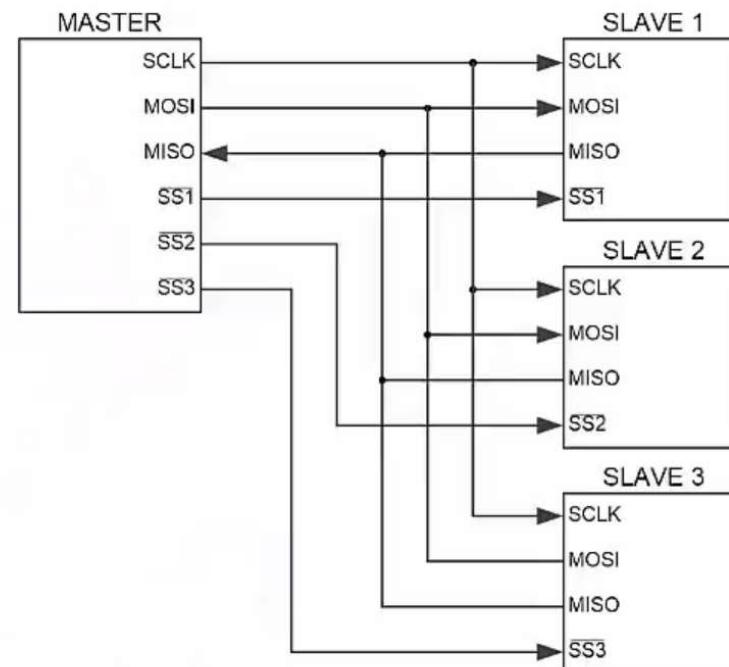
Controlling Multiple Slaves

Controlling Multiple Slaves Method 1 – Multiple SS

Each device has an independent slave select

SCLK, MISO, MOSI are each shared between devices

If device is not selected MISO is becomes high impedance



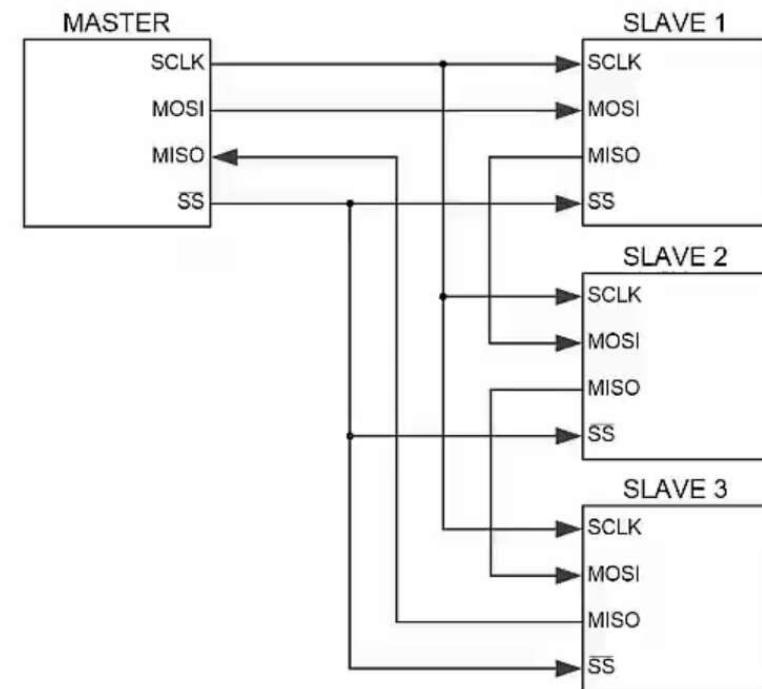
Controlling Multiple Slaves :DAISY CHAINING

Controlling Multiple Slaves Method 2 – Daisy Chain

Single SS controls all slave devices

Data is sent from one device to the next as in a chain; MISO from one device connects to the MOSI of the next device

Not all devices support daisy chaining of communications



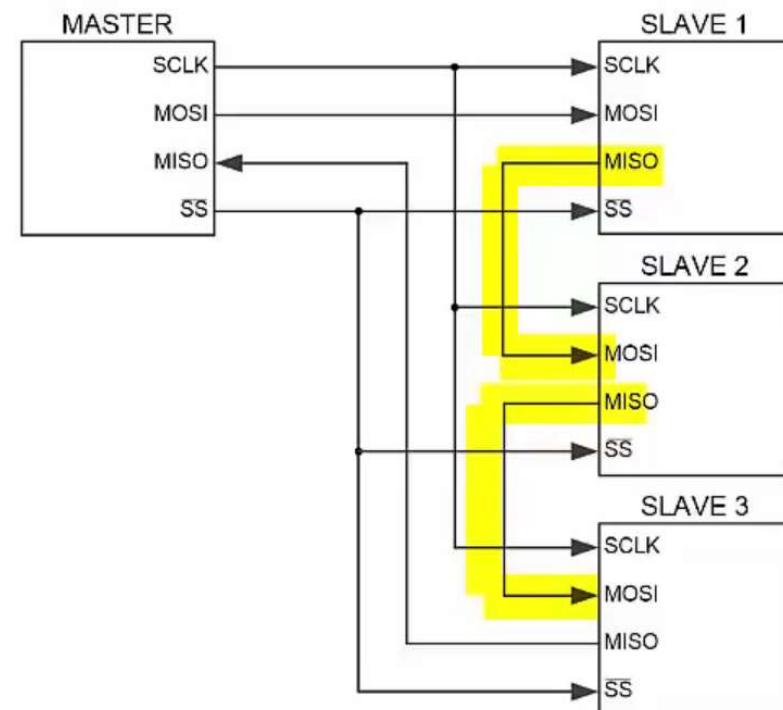
Controlling Multiple Slaves :DAISY CHAINING

Controlling Multiple Slaves Method 2 – Daisy Chain

Single SS controls all slave devices

Data is sent from one device to the next as in a chain; MISO from one device connects to the MOSI of the next device

Not all devices support daisy chaining of communications



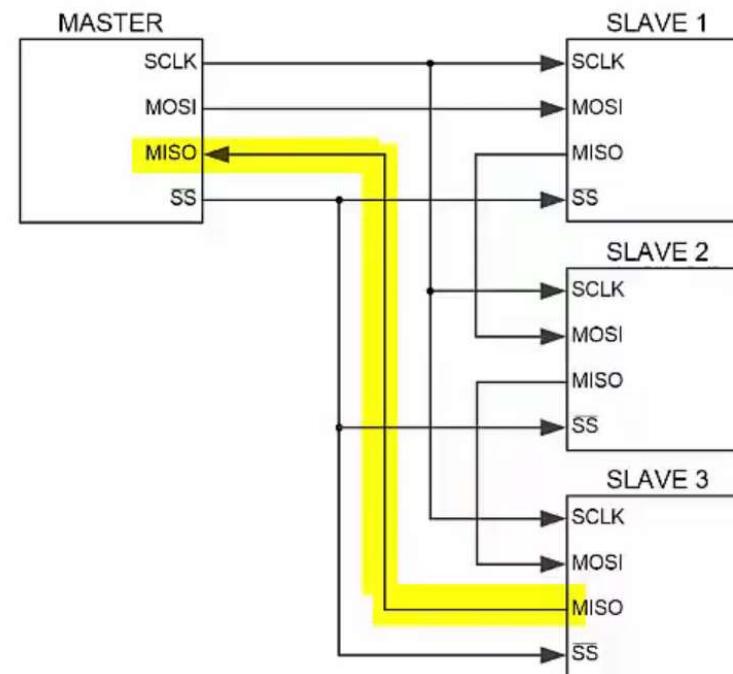
Controlling Multiple Slaves :DAISY CHAINING

Controlling Multiple Slaves Method 2 – Daisy Chain

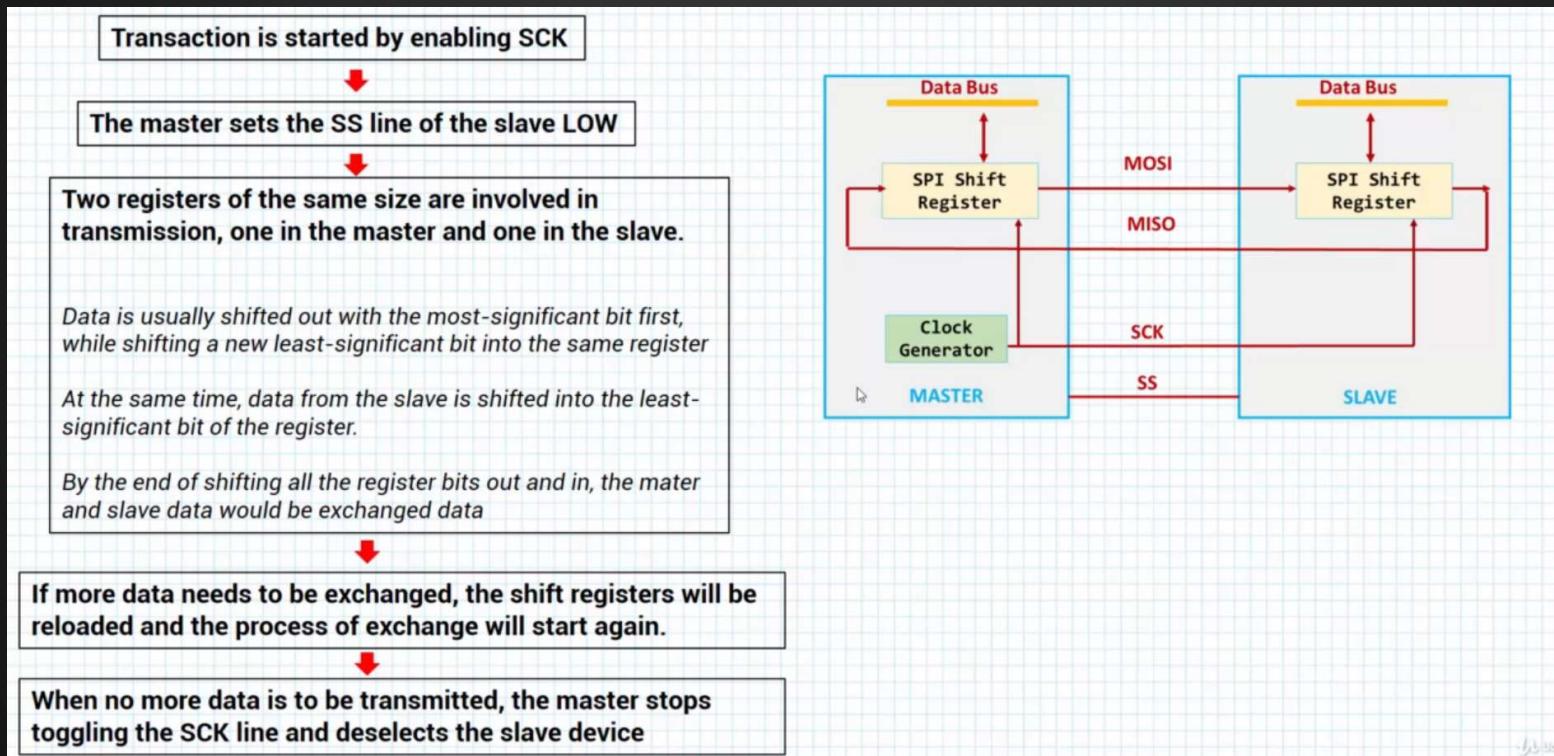
Single SS controls all slave devices

Data is sent from one device to the next as in a chain; MISO from one device connects to the MOSI of the next device

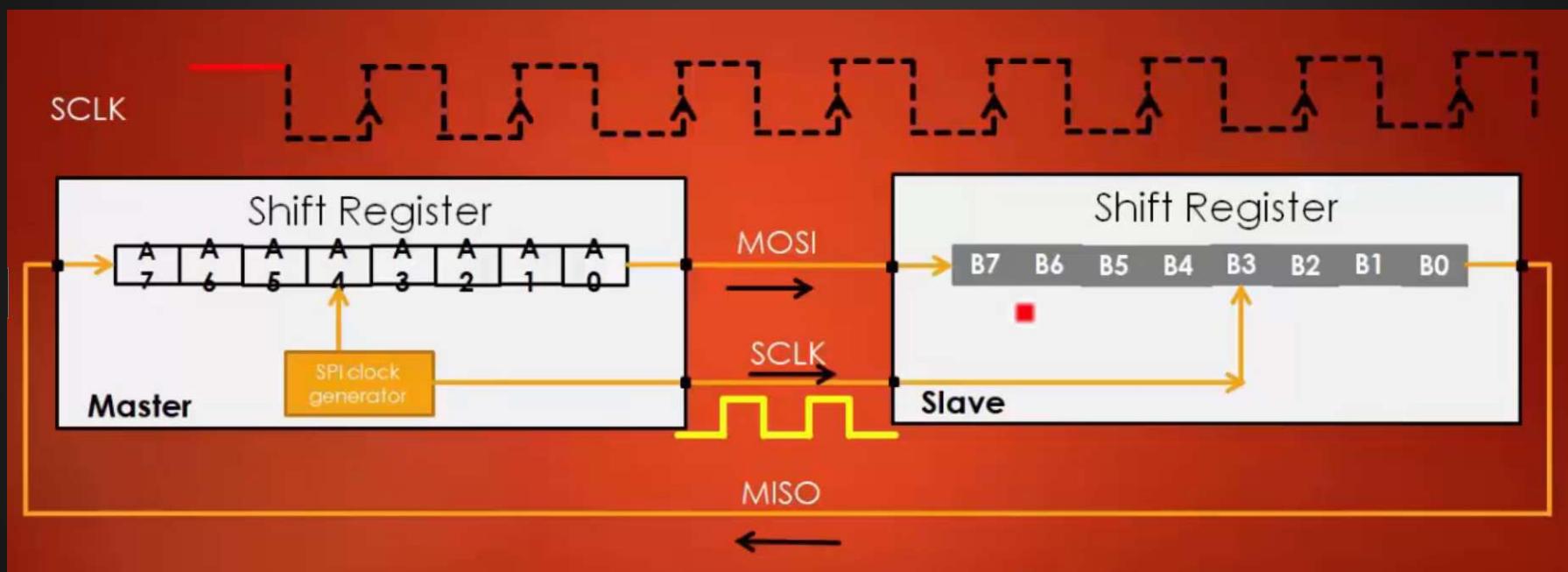
Not all devices support daisy chaining of communications



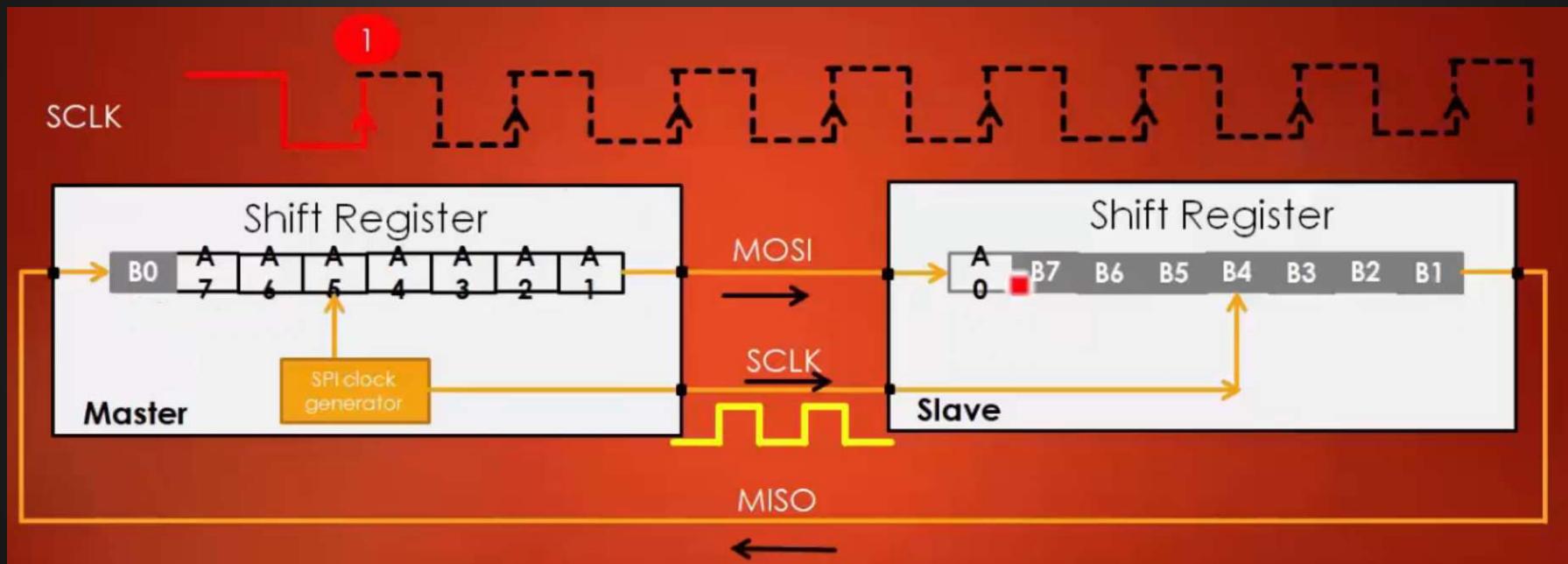
SPI Protocol -How It Works



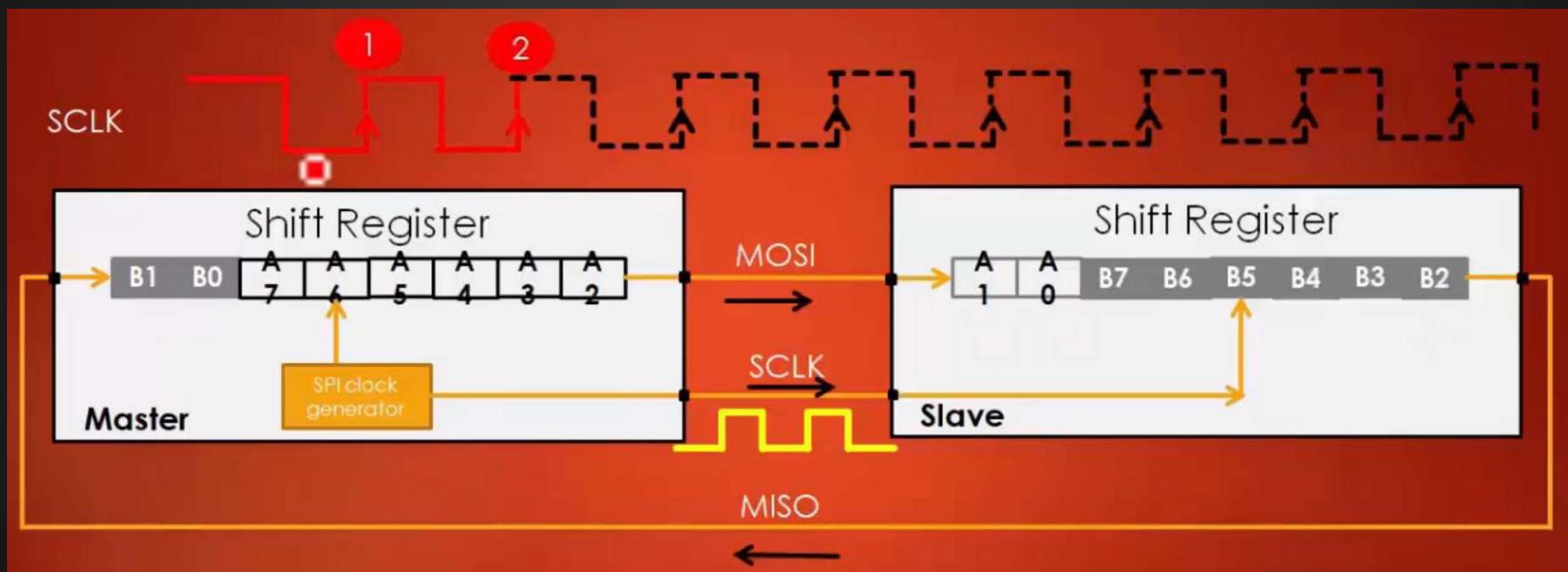
SPI Protocol - How It Works



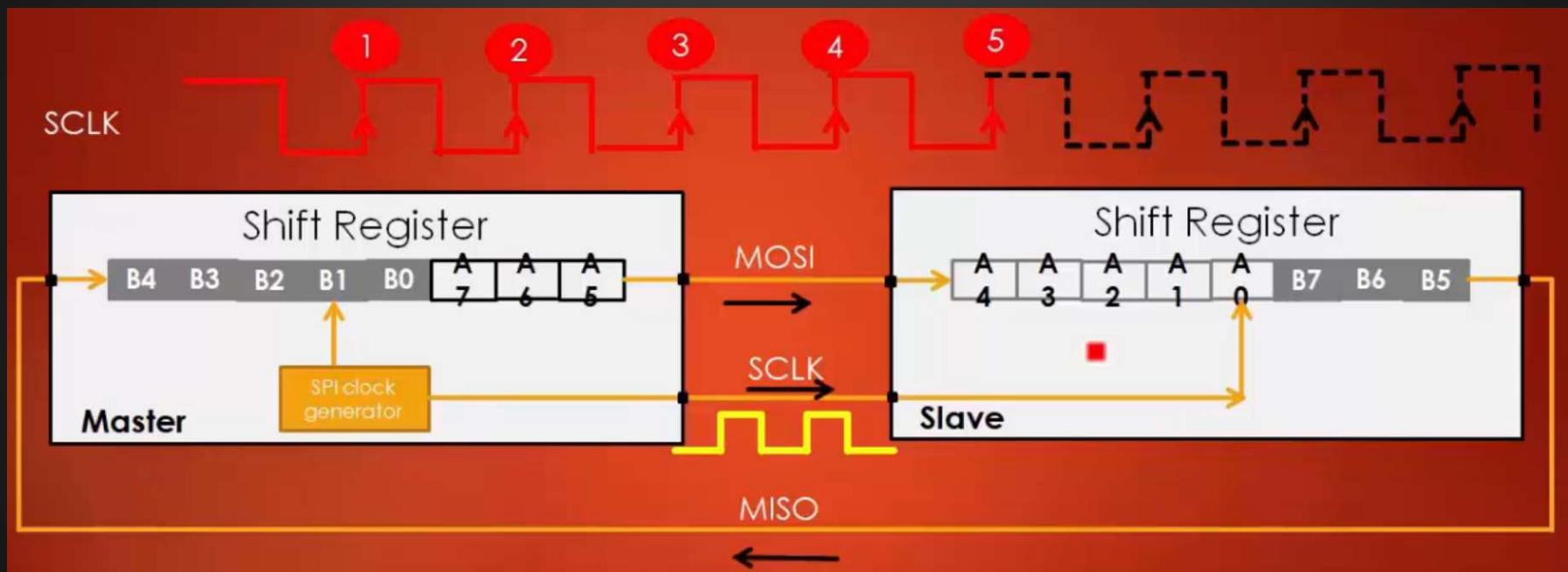
SPI Protocol -How It Works



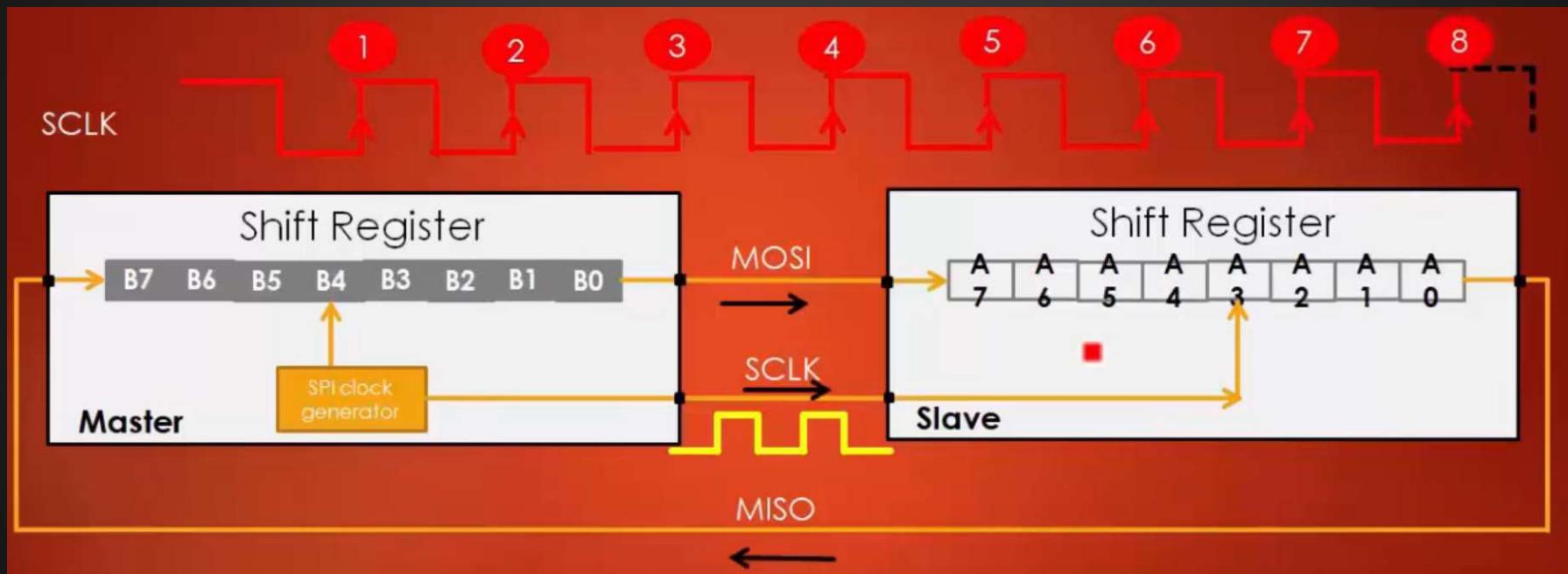
SPI Protocol – How It Works



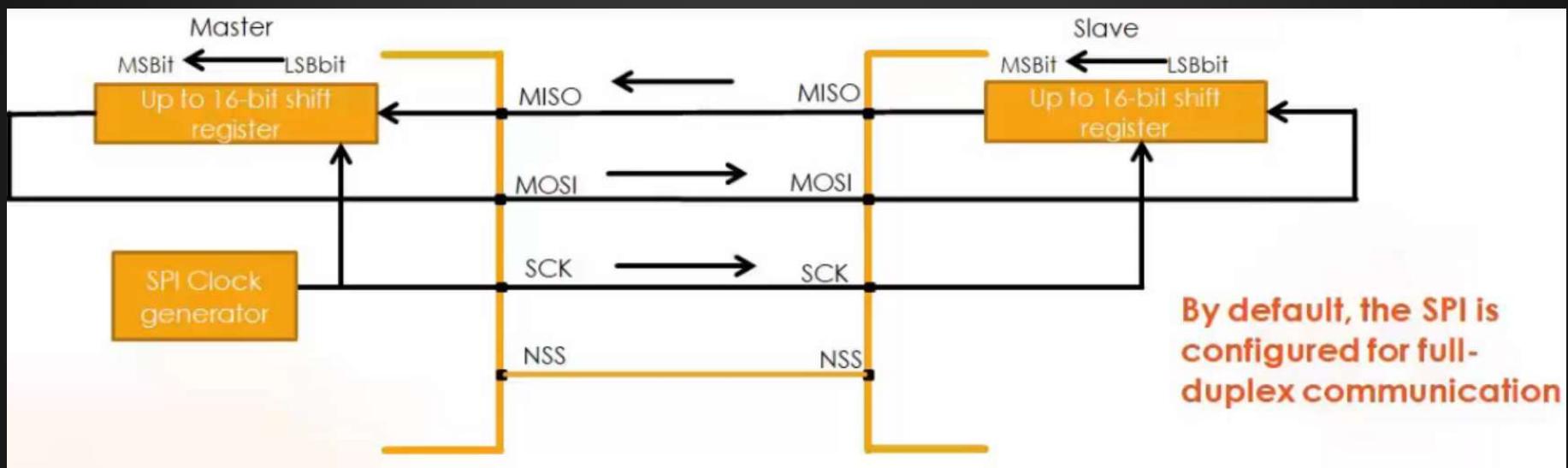
SPI Protocol - How It Works



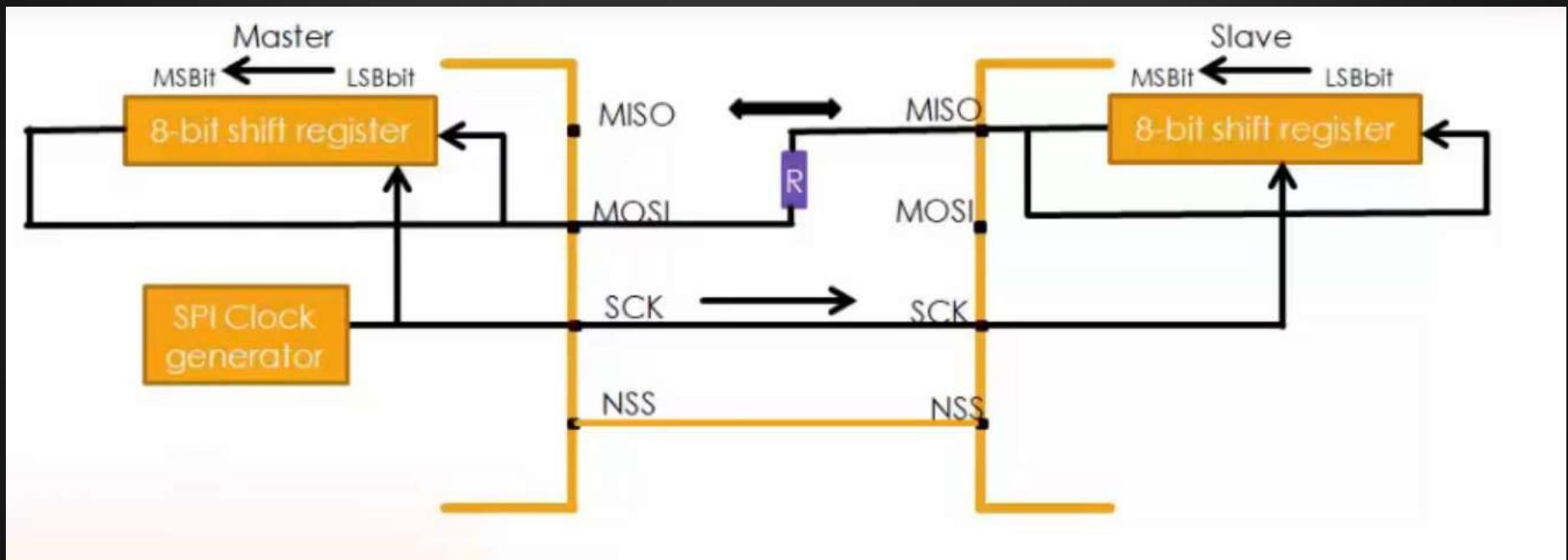
SPI Protocol - How It Works



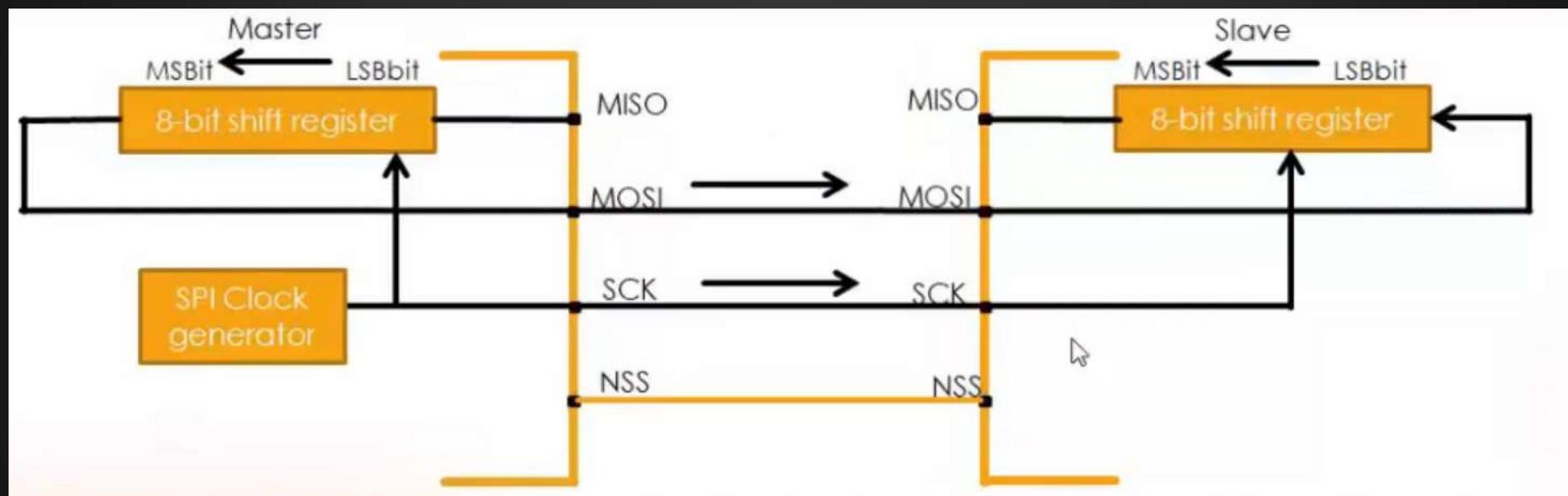
FULL DUPLEX COMMUNICATION



HALF DUPLEX COMMUNICATION



SIMPLEX COMMUNICATION



Transmit Only, Receive only mode

SPI COMMUNICATION

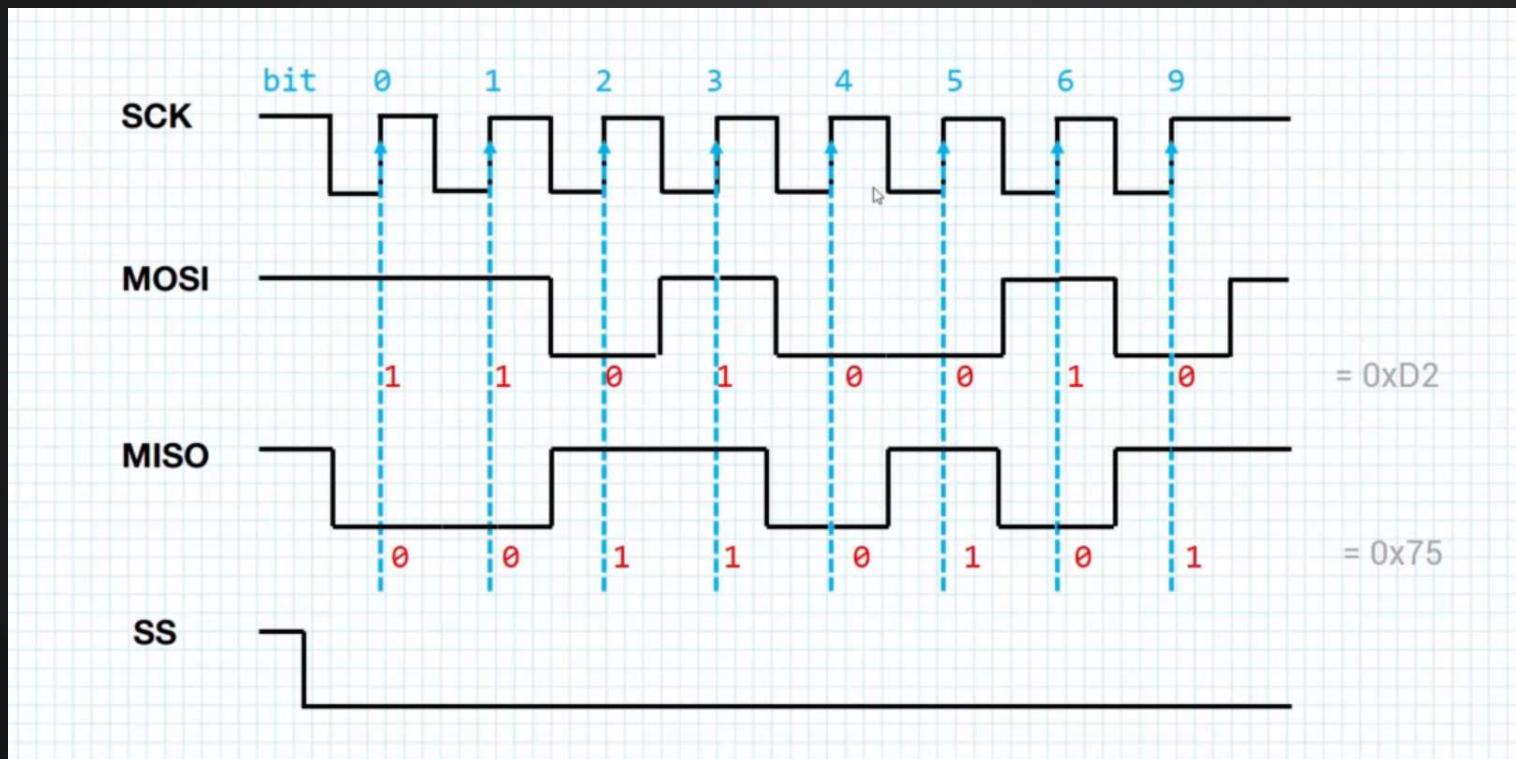
A timing diagram shows the specifications and the timing relationship between the SPI digital lines

The diagram illustrates the SPI communication setup. On the left, a 'CONTROLLER' box is connected to a 'PERIPHERAL' box via four lines: SS, SCLK, MOSI, and MISO. The SS line connects to the CS pin of the peripheral, and the MISO line connects to the DOUT pin of the peripheral. A red callout box labeled 'Names commonly used in TI devices' points to the CS and DOUT pins. A large red arrow points from this section to a timing diagram on the right. The timing diagram shows four signals over time: CS (Chip Select), SCLK (Serial Clock), DIN (Data In), and DOUT (Data Out). The CS signal is asserted (low) during data transfer. The SCLK signal is a continuous square wave. The DIN signal shows data being clocked into the peripheral. The DOUT signal shows data being clocked out of the peripheral.

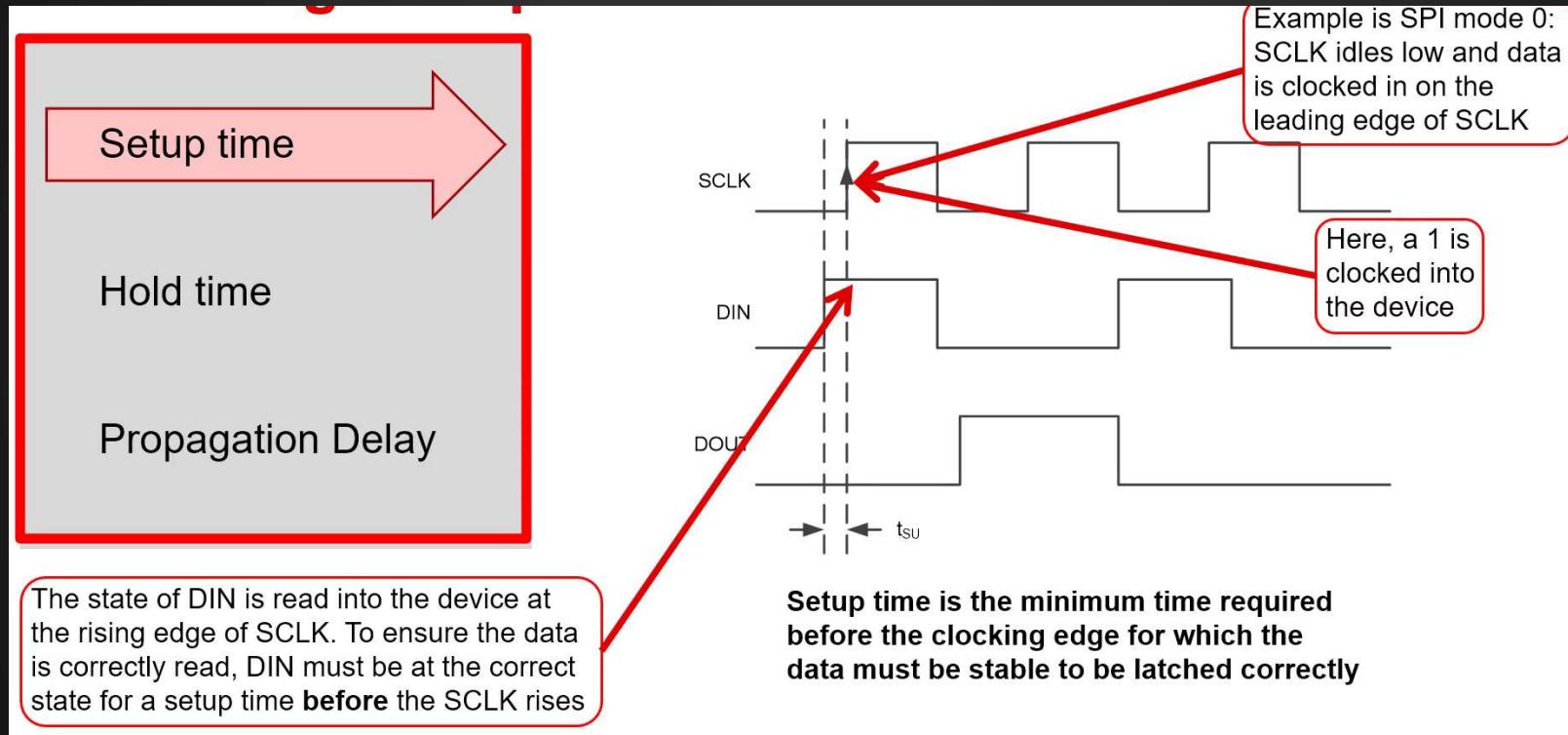
Names commonly used in TI devices

Violating a timing specification can cause a failure to read the data and may cause unexpected results.

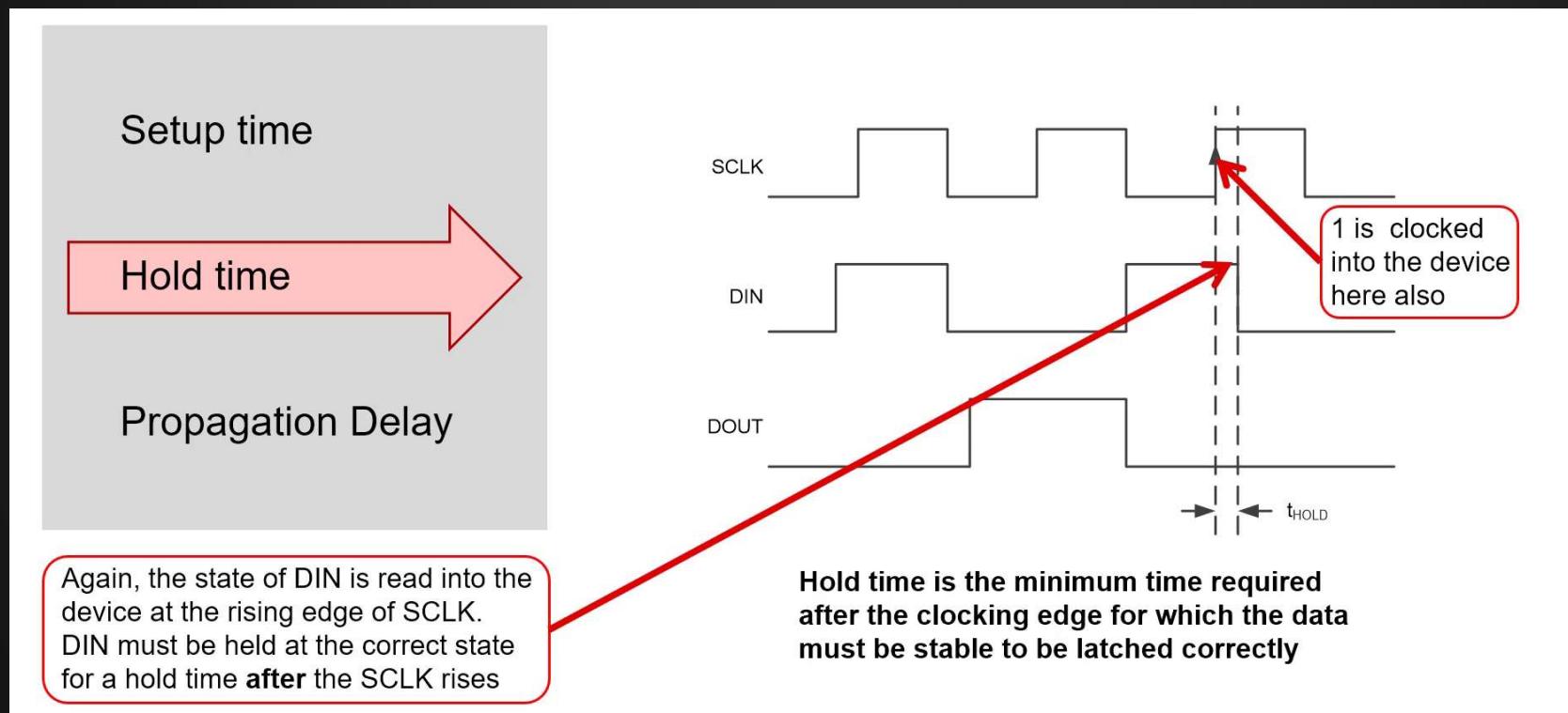
SPI PROTOCOL TRANSMISSION



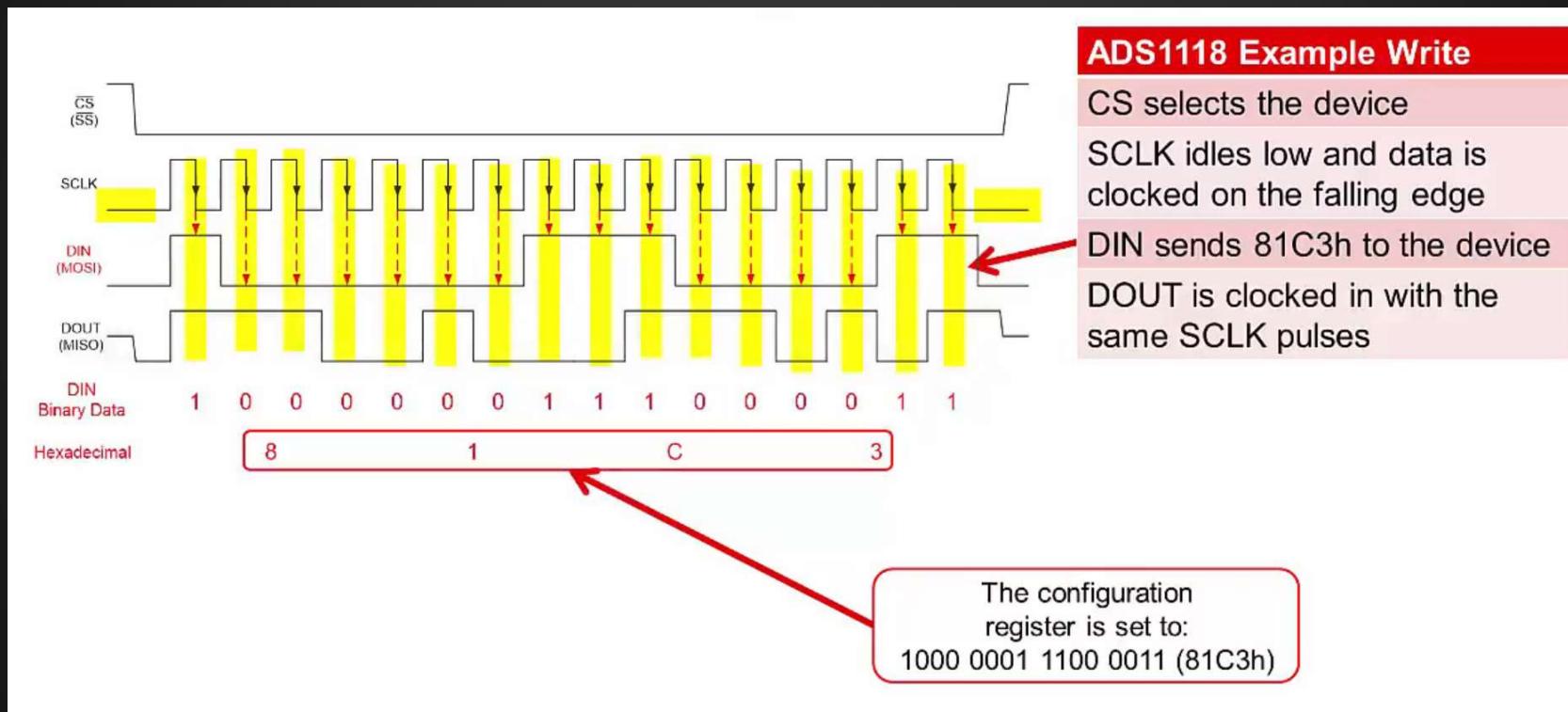
SPI TIMING -SET UP TIME



SPI SWITCHING-PROPAGATION DELAY



SPI Communication Example



Advantages and Disadvantages of SPI

Advantages of SPI:

- It's faster than asynchronous serial
- The receive hardware can be a simple shift register
- It supports multiple peripherals

Disadvantages of SPI:

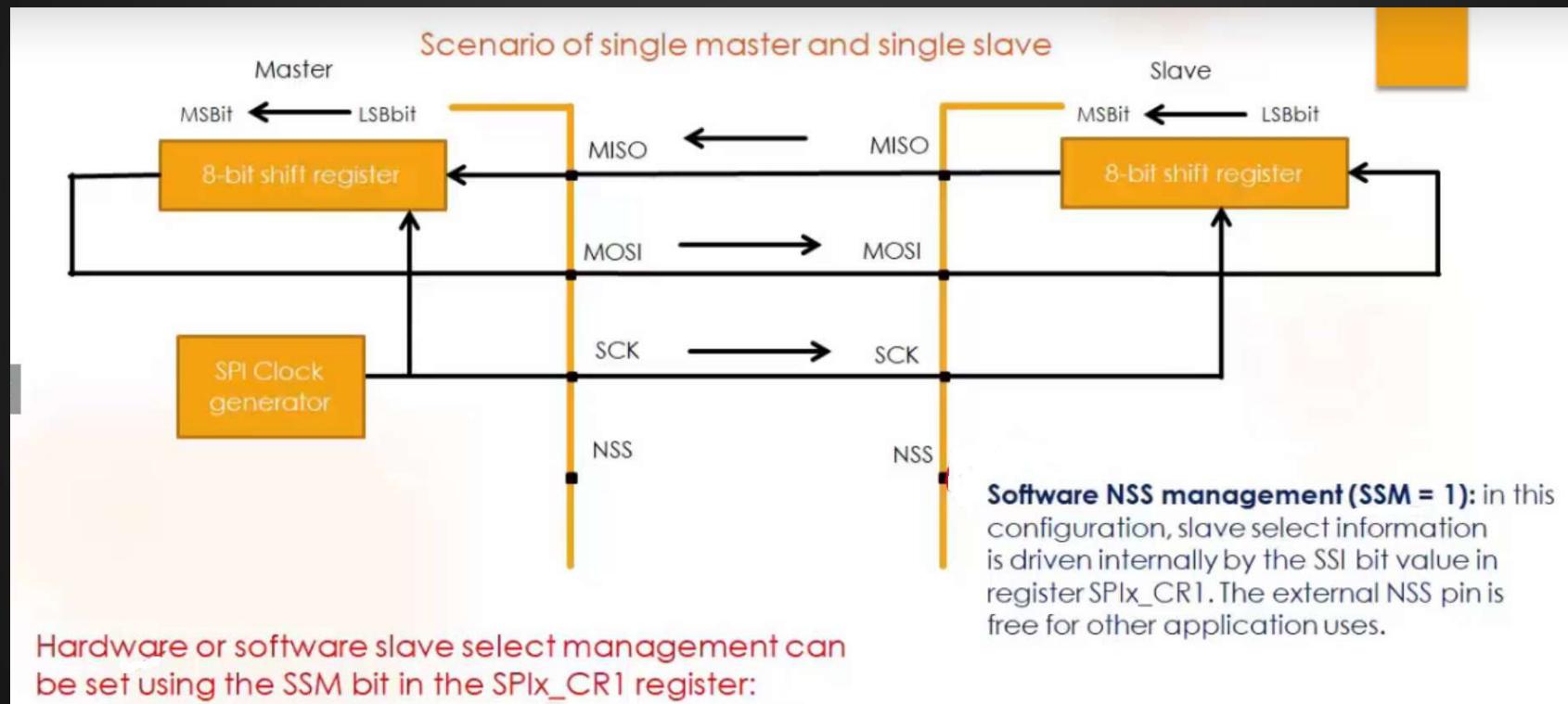
- It requires more signal lines (wires) than other communications methods
- The communications must be well-defined in advance (you can't send random amounts of data whenever you want)
- The controller must control all communications (peripherals can't talk directly to each other)
- It usually requires separate CS lines to each peripheral, which can be problematic if numerous peripherals are needed.

SPI IN STM32F411

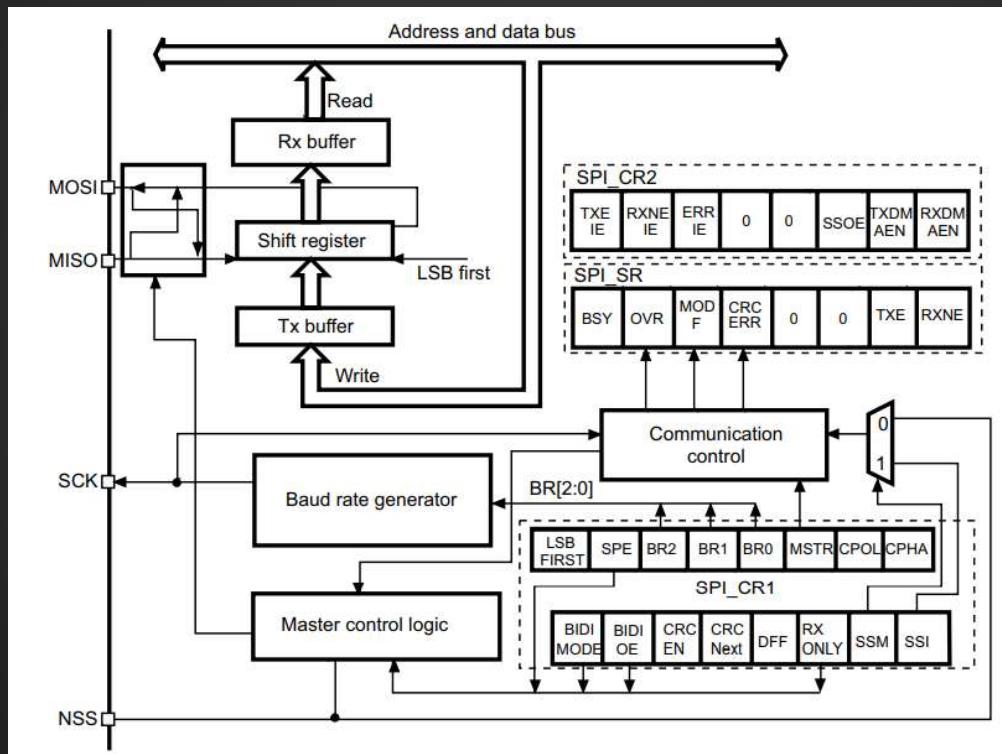
SLAV SELECT(NSS) PIN MANAGEMENT

- **When a Device is Slave Mode**
 - In slave mode , the NSS Work as a standard chip select input and lets the slave communicate with the master.
- **When a Device is Master**
 - In Master mode ,NSS Can be used either as output or input. As an input can prevent multimaster bus collision and as an output it can drive a slave select signal of a single slave.

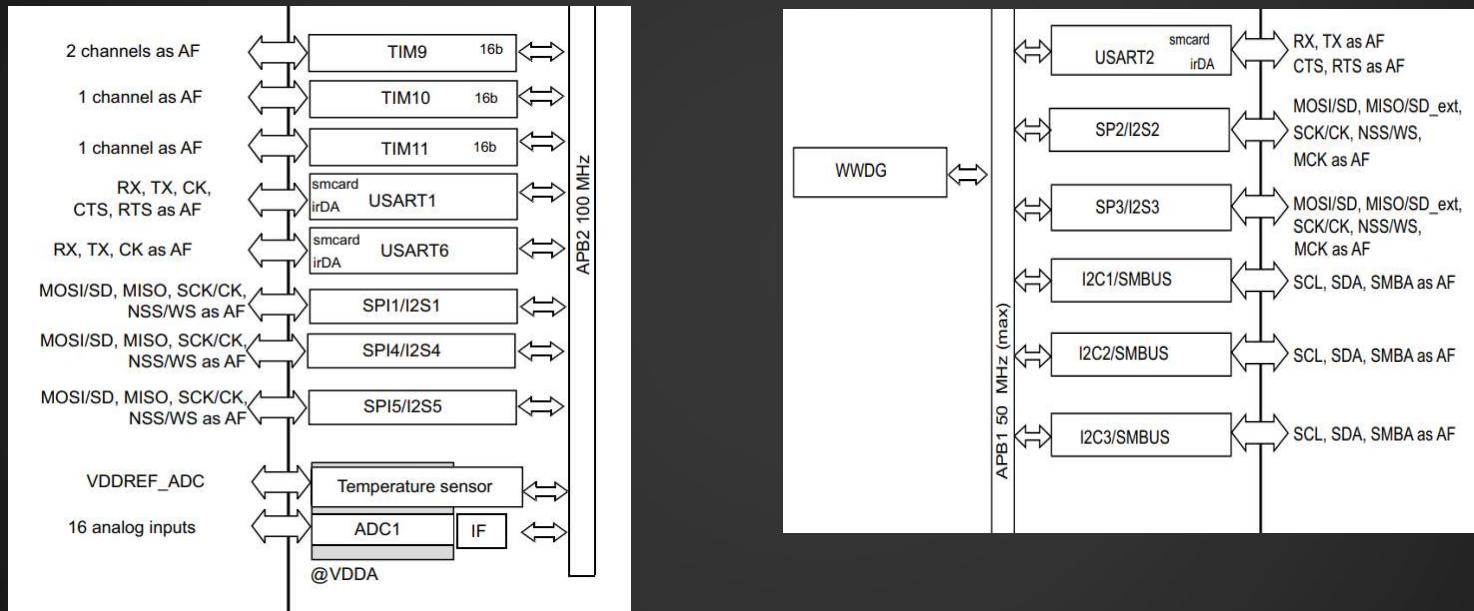
TYPES OF SLAVE MANAGEMENT



SPI FUNCTIONAL BLOCK DIAGRAM

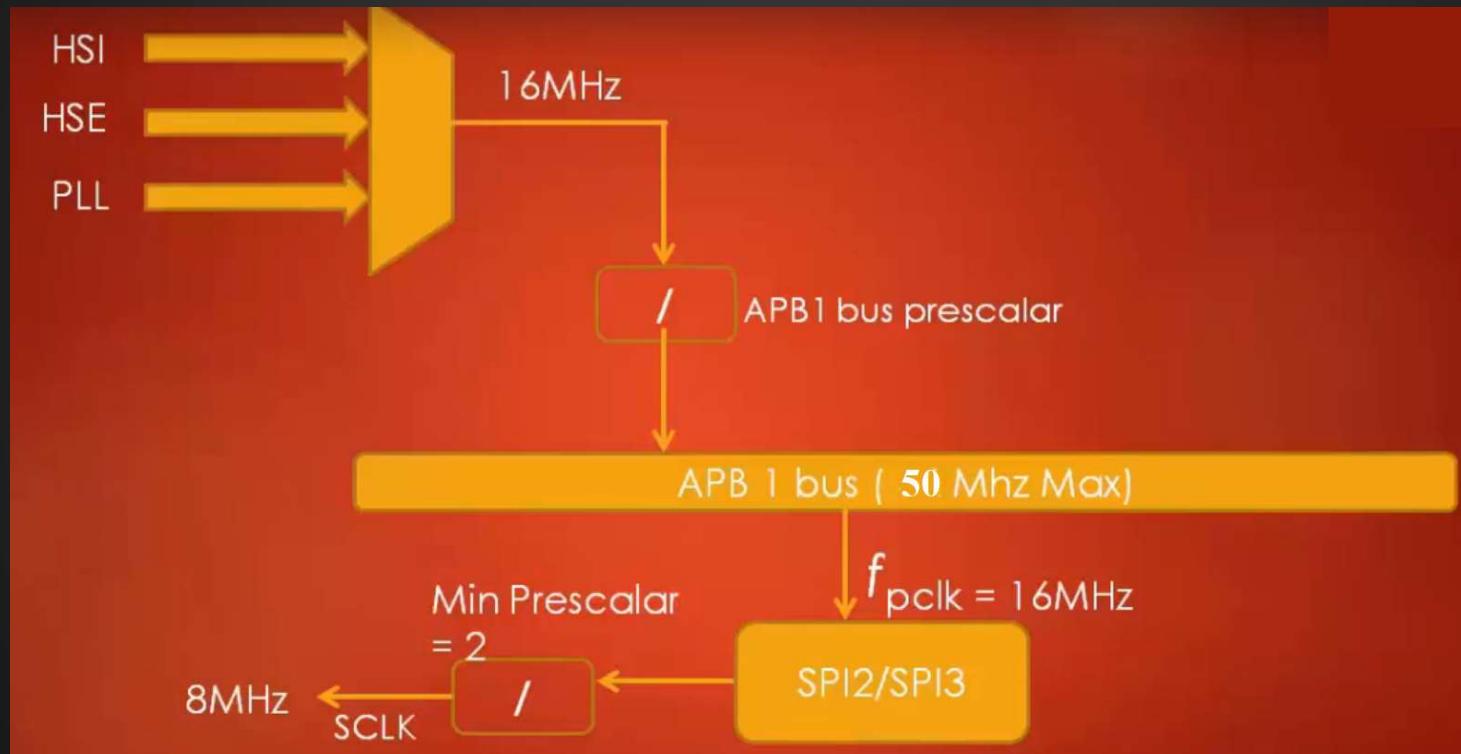


SPI PERIPHERAL BUS CONNECTION



```
#define GPIOAEN (1U<<0)
#define SPI1EN (1U<<12)
```

SPI SERIAL CLK FREQUENCY



SPI Control Register(SPI_CR1)

27.5.1 SPI control register 1 (SPI_CR1) (not used in I²S mode)

Address offset: 0x00

Reset value: 0x0000

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

BIDIMODE: Bidirectional data mode enable

BIDIOE: Output enable in bidirectional mode

CRCEN: Hardware CRC calculation enable

CRCNEXT: CRC transfer next

DFF: Data frame format

RXONLY: Receive only

SSM: Software slave management

SSI: Internal slave select

LSBFIRST: Frame format

SPE: SPI enable

BR[2:0]: Baud rate control

MSTR: Master selection

CPOL: Clock polarity

CPHA: Clock phase

SPI Control Register(SPI_CR2)

27.5.2 SPI control register 2 (SPI_CR2)

Address offset: 0x04

Reset value: 0x0000

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

SPI Status Register(SPI_SR)

27.5.3 SPI status register (SPI_SR)

Address offset: 0x08

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	CHSID E	TXE	RXNE
r	r	r	r	r	rc_w0	r	r	r	r	r	r	r	r	r	r

```
#define SR_TXE (1U<<1)
#define SR_RXNE (1U<<0)
#define SR_BUSY (1U<<7)
```

SPI Data Register(SPI_DR)

27.5.4 SPI data register (SPI_DR)

Address offset: 0x0C

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

SPI RX CRC Register(SPI_RXCRCR)

27.5.6 SPI RX CRC register (SPI_RXCRCR) (not used in I²S mode)

Address offset: 0x14

Reset value: 0x0000

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

SPI TX CRC Register(SPI_TXCRCR)

27.5.6 SPI RX CRC register (SPI_RXCRCR) (not used in I²S mode)

Address offset: 0x14

Reset value: 0x0000

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

SPI TX CRC Register(SPI_TXCRCR)

27.5.6 SPI RX CRC register (SPI_RXCRCR) (not used in I²S mode)

Address offset: 0x14

Reset value: 0x0000

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

SPI I²S Configuration Register(SPI_I2SCFGR)

27.5.8 SPI_I²S configuration register (SPI_I2SCFGR)

Address offset: 0x1C

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				I2SMOD	I2SE	I2SCFG		PCMSY NC	Reserved	I2SSTD		CKPOL	DATLEN		CHLEN
				rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw

SPI I²S PRESCALER Register(SPI_I2SPR)

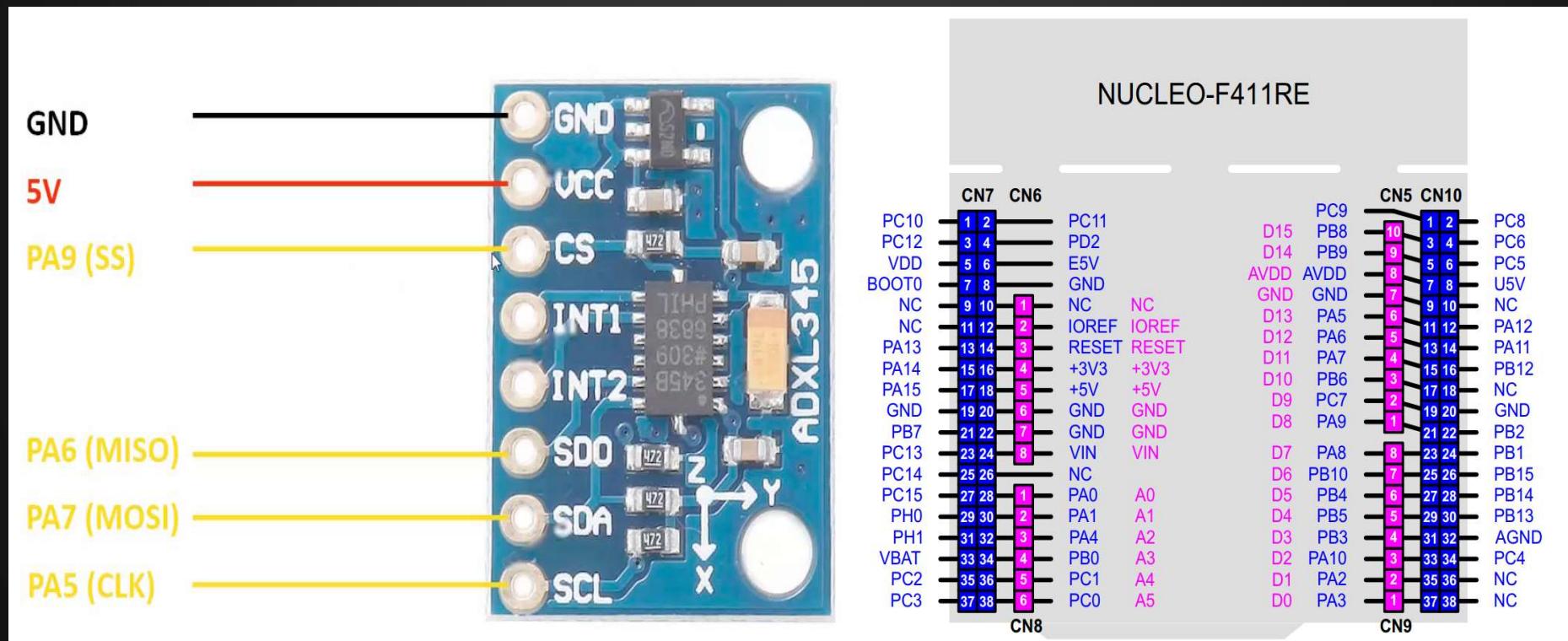
27.5.9 SPI_I²S prescaler register (SPI_I2SPR)

Address offset: 0x20

Reset value: 0000 0010 (0x0002)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						MCKOE	ODD	I2SDIV							
						rw	rw	rw							

SCHEMATIC DESIGN



Mindset Activity

- ✓ Write Down Your Top 10 Goals.(1 Mark)
- ✓ Write Down Your Top 10 Ideas to Achieve Your Goal.(1 Mark)
- ✓ 30 Minutes for Workout (5000-7000 Steps a Day)(2 Mark).
- ✓ 15 Minutes to Meditate (2 Mark)
- ✓ 10 Minutes to Visualize of Achieving Your Goals(1)
- ✓ 10 Minutes to Focus on the Day Plan (1)
- ✓ 2 Hr's for Learning and Take Notes. (2 Mark)



10/10

<https://www.facebook.com/groups/embeddedsystemsandiot/>

THANK YOU