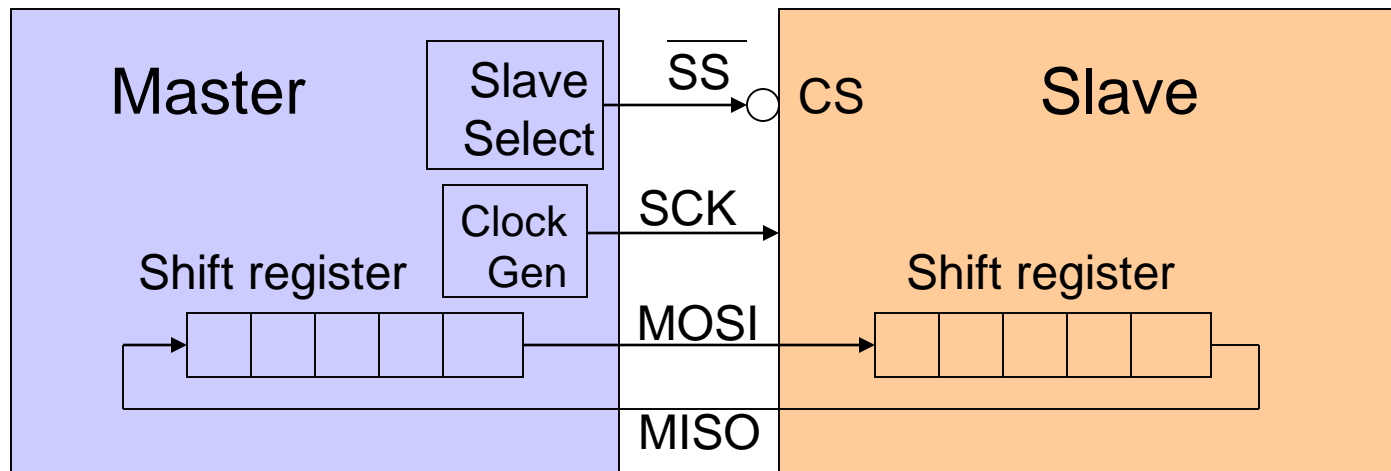


---

# Serial Peripheral Interface (SPI)

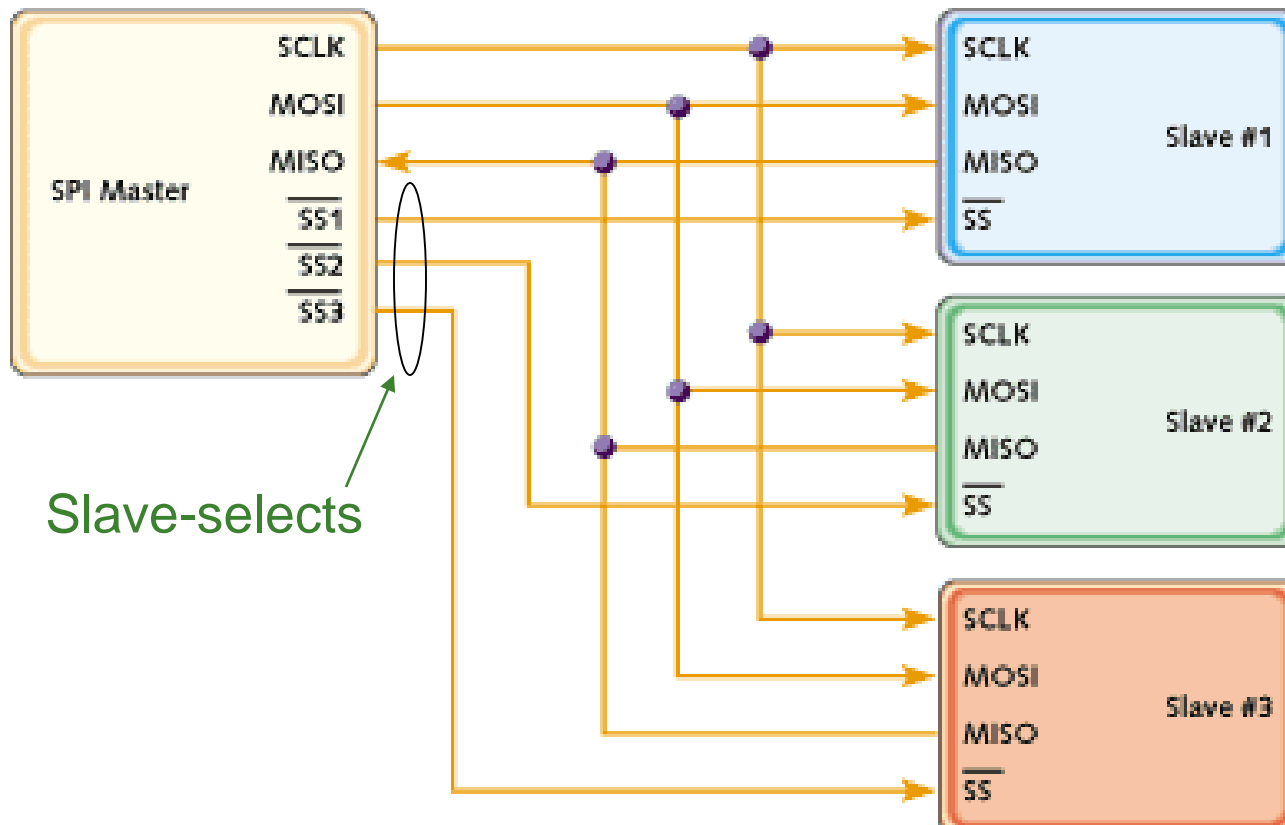
- Synchronous serial data transfers
    - Multipoint serial communication between a “**master**” and a “**slave**” device
    - Clock permits faster data rates than async communications (**framing unnecessary**)
    - Signals = clock, data in/out, “slave select”
    - **Master** controls data transfers:
      - transmit a synchronization clock
      - activate slave select signal
    - All device data registers effectively linked into a single “shift register”
-

# Single master, single-slave SPI connections



SCLK – serial clock, generated by the master  
MOSI – master output/slave input  
MISO – master input/slave output  
SS – slave select/enable signal

# Single master, multiple slave SPI implementation



# Single master, multiple slave SPI implementation – daisy chained

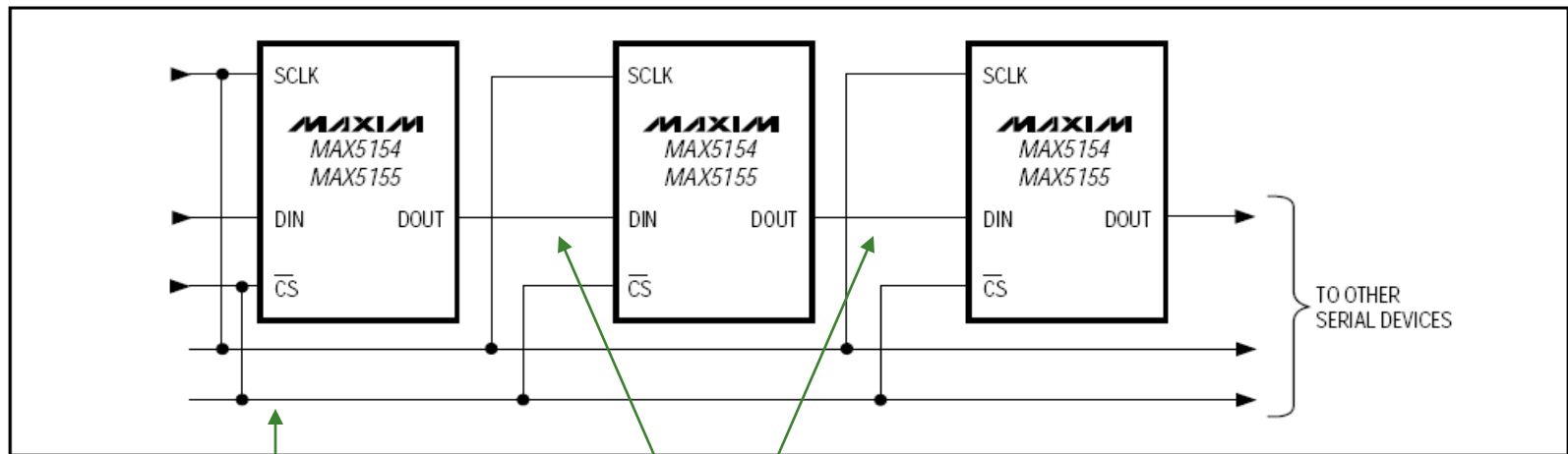


Figure 7. Daisy Chaining MAX5154/MAX5155s

- Dout of one device connected to Din of next (creates a single shift register)
- All devices selected concurrently by the Master

# SPI serial data timing

- Programmable clock rate and timing for flexibility
  - CPOL = clock polarity (0=active-high, 1= active-low)
  - CPHA = clock phase (sample on leading/trailing pulse edge)

CPHA=0 : data output immediately when PCSn active  
data sampled on leading edge

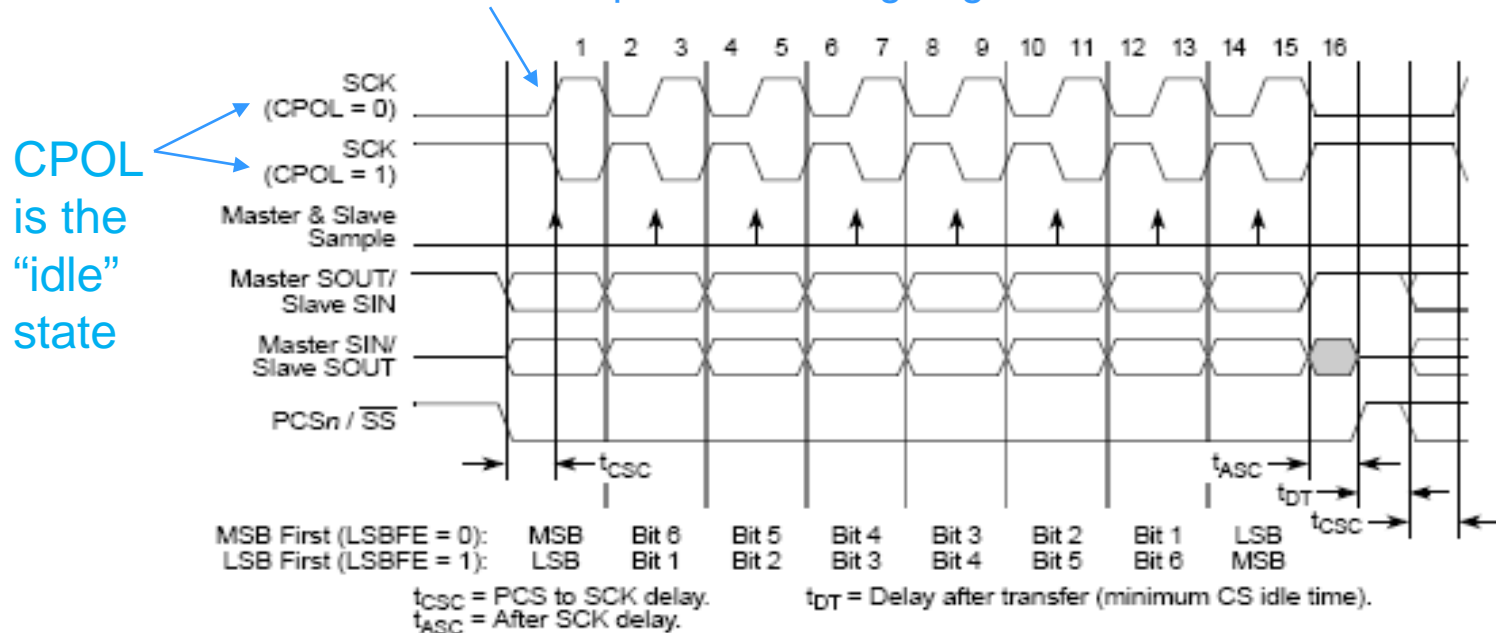


Figure 22-16. DSPI Transfer Timing Diagram (MTFE=0, CPHA=0, FMSZ=8)

# SPI serial data timing

- CPHA=1
  - data output on 1<sup>st</sup> clock edge after PCSn active
  - data sampled on trailing edge

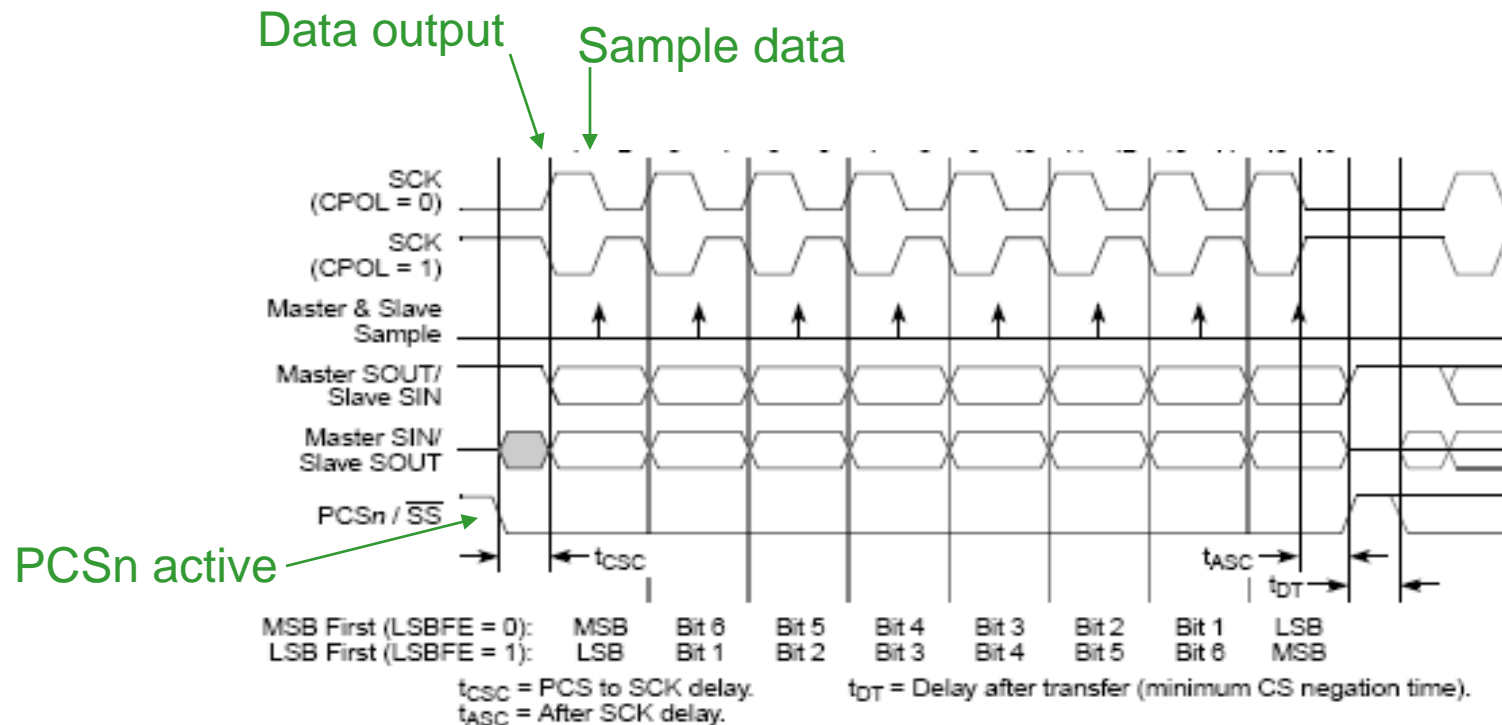
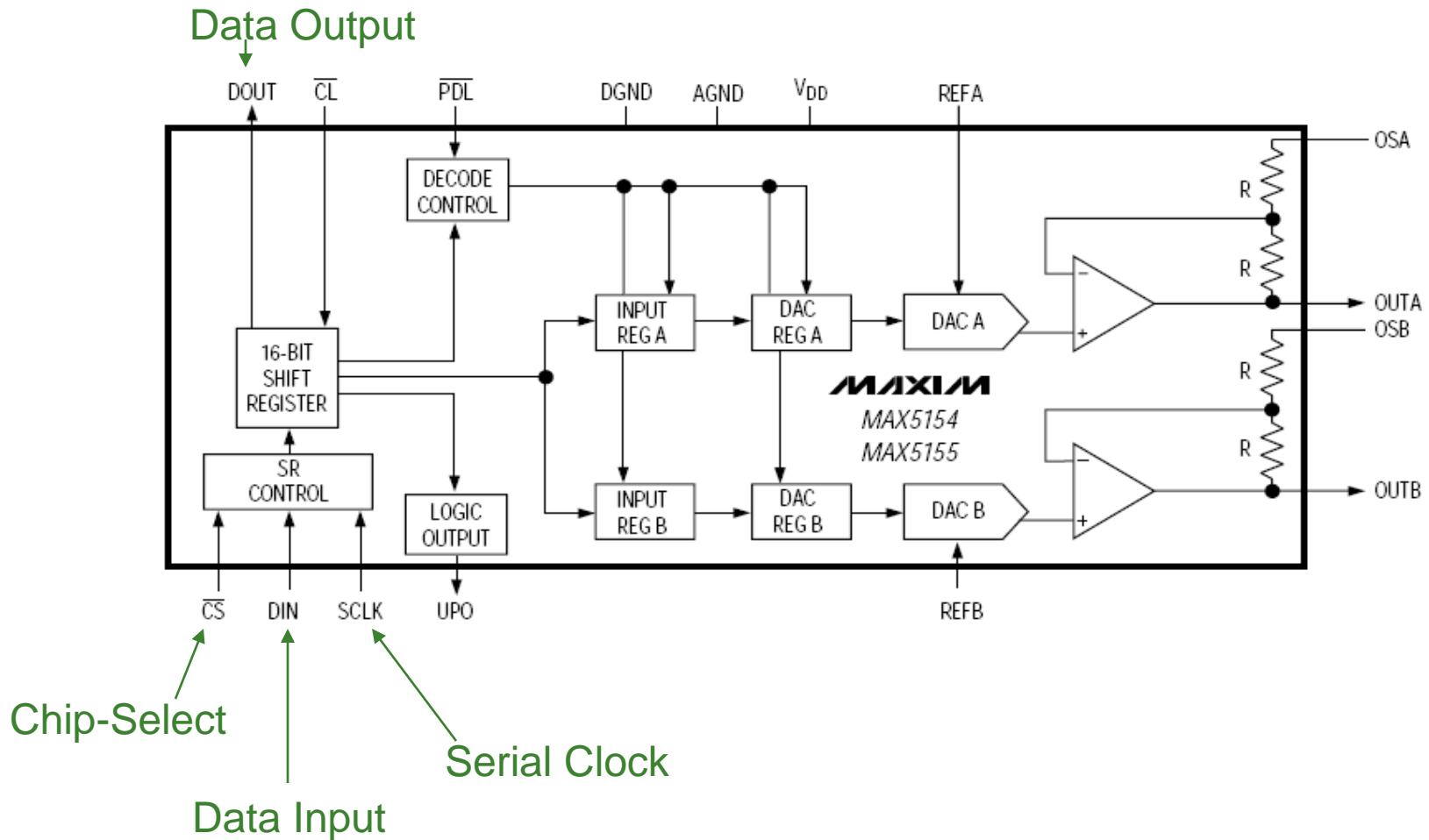


Figure 22-17. DSPI Transfer Timing Diagram (MTFE=0, CPHA=1, FMSZ=8)

# Maxim MAX5154 12-Bit Serial DAC



# MAX5154 serial data format

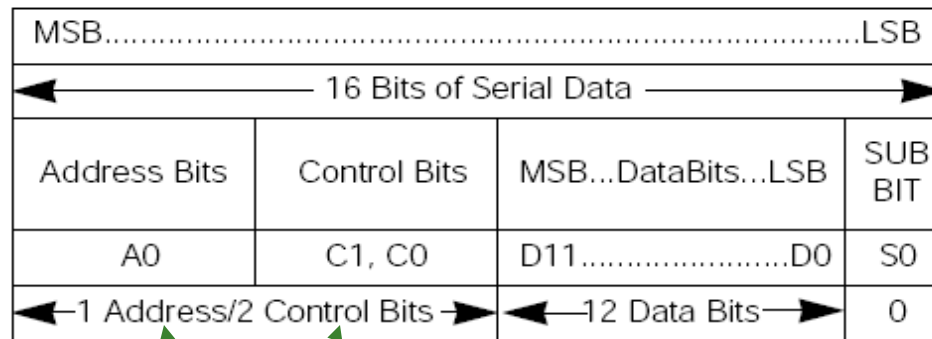
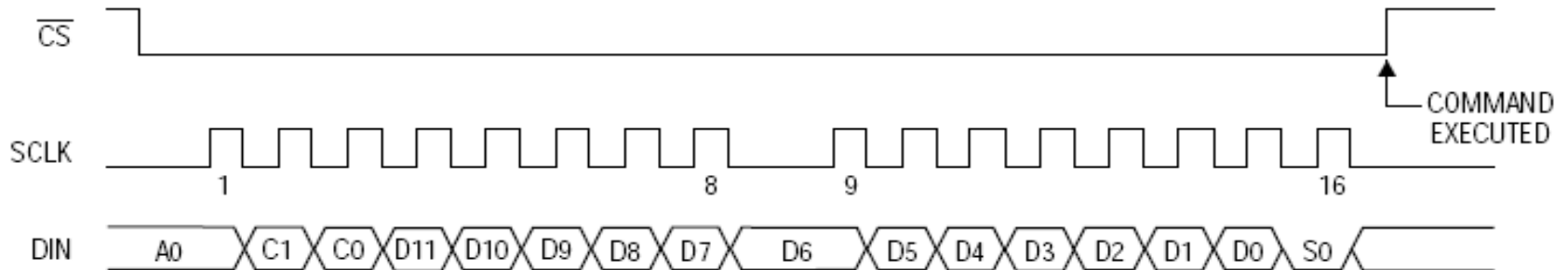


Figure 4. Serial-Data Format

Command and address bits select channel and conversion properties.

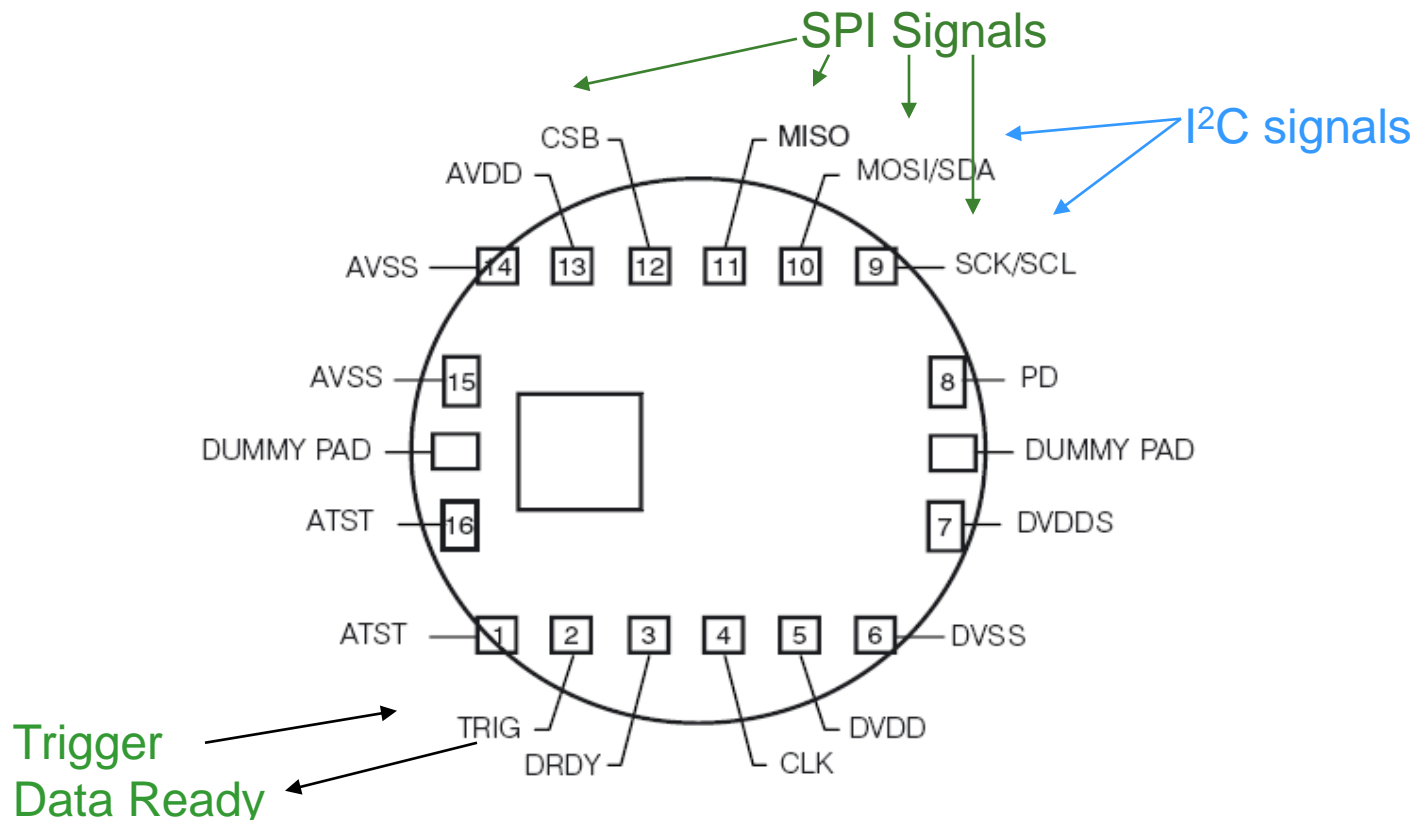


# VTI Technologies

## SCP1000 Pressure Sensor

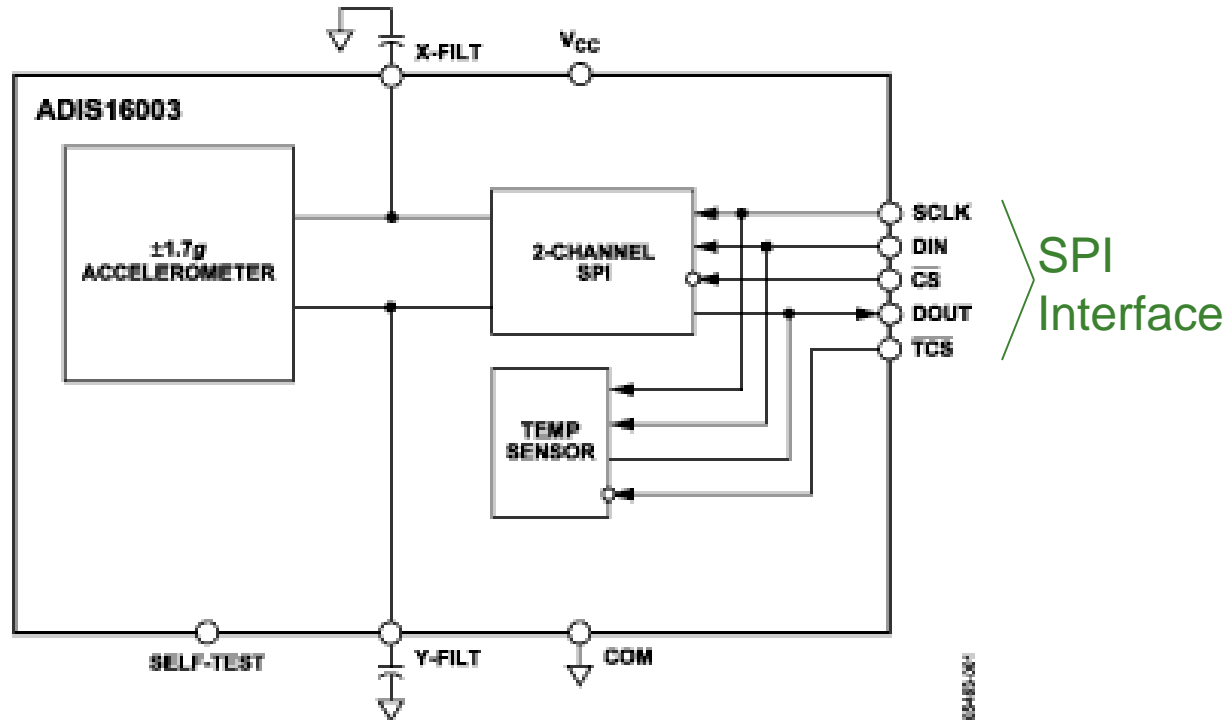
Supports SPI or I2C (factory programmed)

19 bit pressure, 14 bit temperature



# Analog Devices

## ADIS16003 Dual-axis accelerometer



12-bit acceleration/10-bit temperature

# ADIS16003 Dual-axis accelerometer

## SPI interface timing

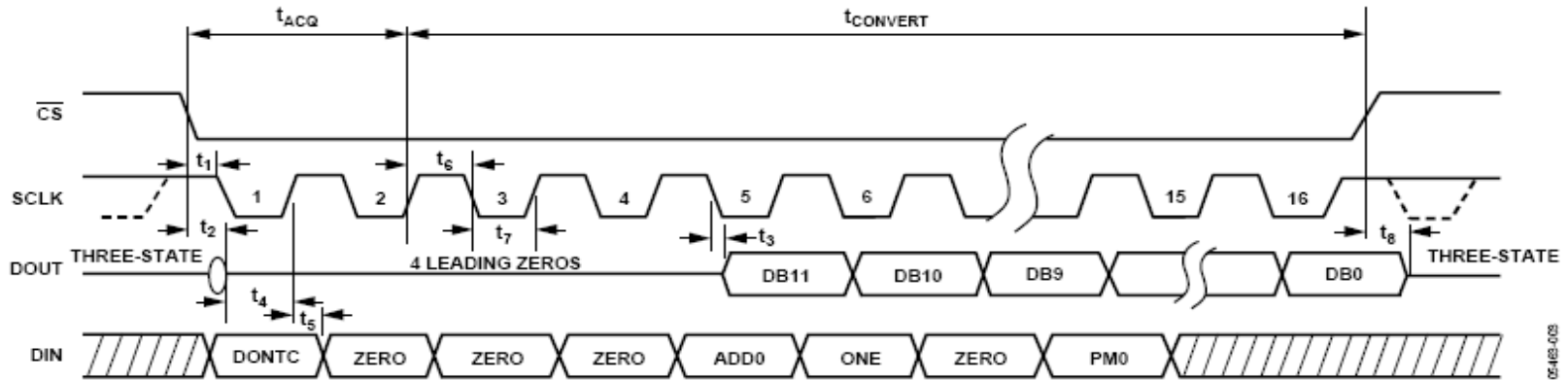


Figure 3. Accelerometer Serial Interface Timing Diagram

054103-003

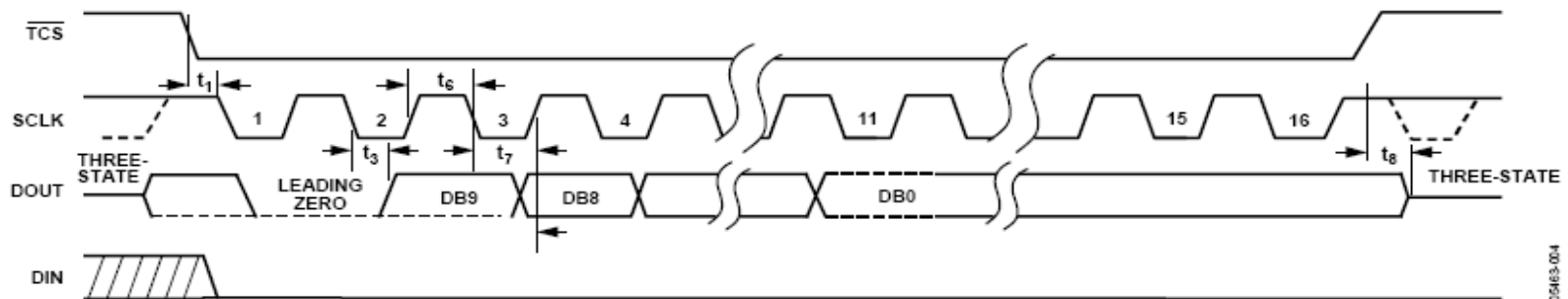
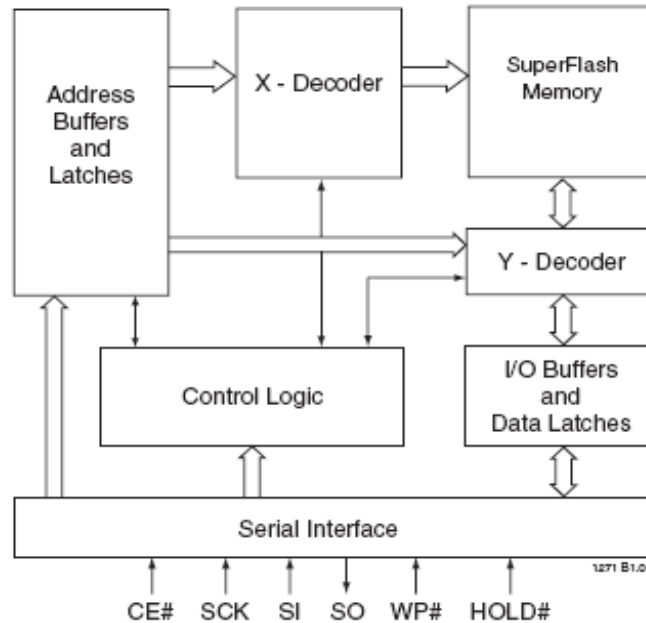


Figure 4. Temperature Serial Interface Timing Diagram

054103-004

# SST: SST25VF016B

## 16 Mbit SPI serial flash memory



Use in DVDs, hard drives, PCs, WLANs, LCD monitors, MP3 players, FPGAs, etc.

- 50 MHz clock
- 8-lead SOIC package
- 7 us byte program
- 18 ms block erase

4-wire SPI interface

suspend (hold) serial transfer

write protection

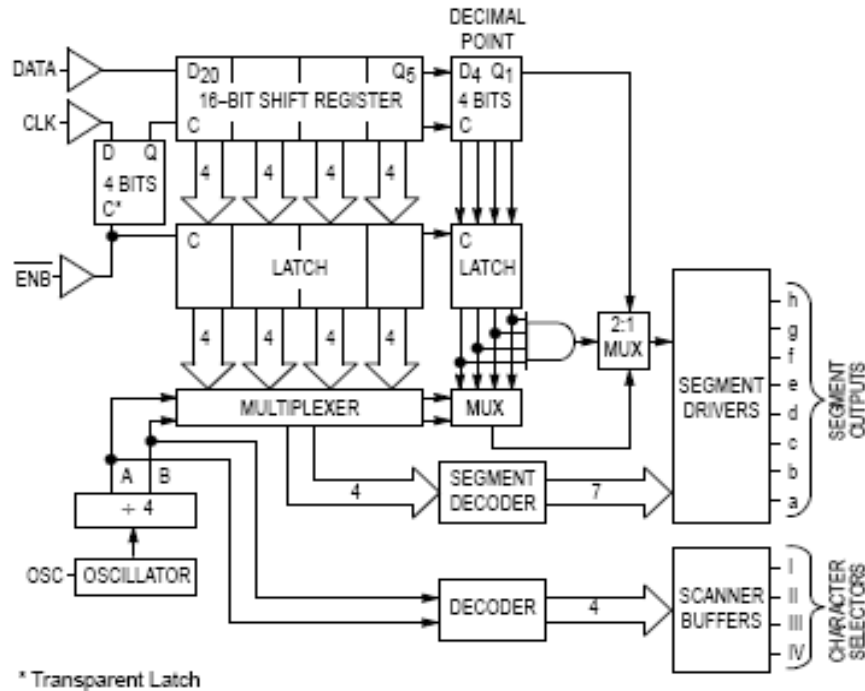
# SST25VF016B serial flash : device operation instructions

**TABLE 5: DEVICE OPERATION INSTRUCTIONS**

Instruction	Description	Op Code Cycle <sup>1</sup>	Address Cycle(s) <sup>2</sup>	Dummy Cycle(s)	Data Cycle(s)	Maximum Frequency
Read	Read Memory at 25 MHz	0000 0011b (03H)	3	0	1 to ∞	25 MHz
High-Speed Read	Read Memory at 50 MHz	0000 1011b (0BH)	3	1	1 to ∞	50 MHz
4 KByte Sector-Erase <sup>3</sup>	Erase 4 KByte of memory array	0010 0000b (20H)	3	0	0	50 MHz
32 KByte Block-Erase <sup>4</sup>	Erase 32 KByte block of memory array	0101 0010b (52H)	3	0	0	50 MHz
64 KByte Block-Erase <sup>5</sup>	Erase 64 KByte block of memory array	1101 1000b (D8H)	3	0	0	50 MHz
Chip-Erase	Erase Full Memory Array	0110 0000b (60H) or 1100 0111b (C7H)	0	0	0	50 MHz
Byte-Program	To Program One Data Byte	0000 0010b (02H)	3	0	1	50 MHz
AAI-Word-Program <sup>6</sup>	Auto Address Increment Programming	1010 1101b (ADH)	3	0	2 to ∞	50 MHz
RDSR <sup>7</sup>	Read-Status-Register	0000 0101b (05H)	0	0	1 to ∞	50 MHz
EWSR	Enable-Write-Status-Register	0101b 0000b (50H)	0	0	0	50 MHz
WRSR	Write-Status-Register	0000 0001b (01H)	0	0	1	50 MHz
WREN	Write-Enable	0000 0110b (06H)	0	0	0	50 MHz
WRDI	Write-Disable	0000 0100b (04H)	0	0	0	50 MHz
RDID <sup>8</sup>	Read-ID	1001 0000b (90H) or 1010 1011b (ABH)	3	0	1 to ∞	50 MHz
JEDEC-ID	JEDEC ID read	1001 1111b (9FH)	0	0	3 to ∞	50 MHz
EBSY	Enable SO to output RY/BY# status during AAI programming	0111 0000b (70H)	0	0	0	50 MHz
DBSY	Disable SO to output RY/BY# status during AAI programming	1000 0000b (80H)	0	0	0	50 MHz

# Motorola MC14499 7-segment LED display decoder/driver with SPI

- 7-segment alphanumeric LED decoder/driver
- drives 4 characters with decimal points
- NPN output drivers for common-cathode LEDs

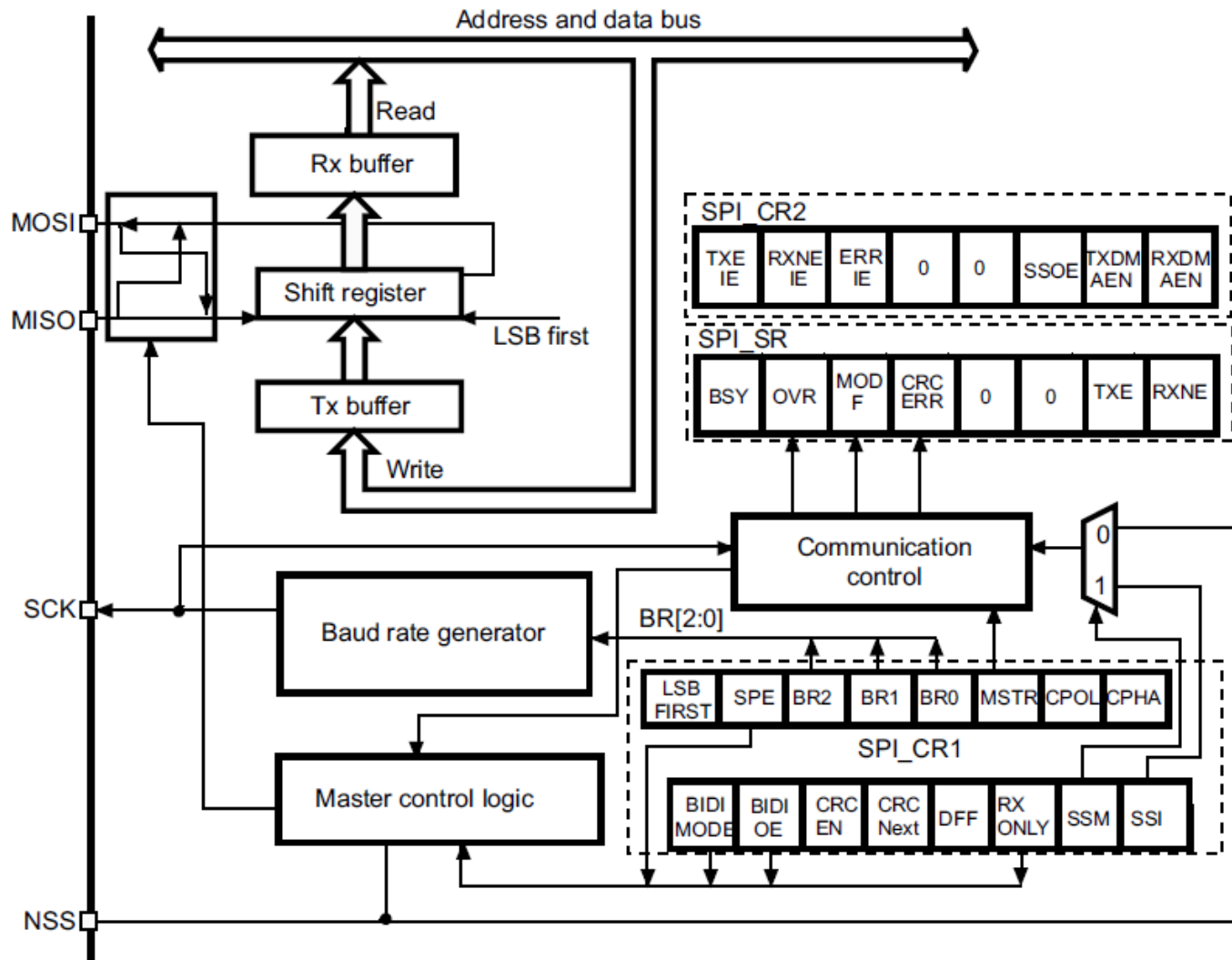


---

# STM32 Serial Peripheral Interface (SPI)

- Dual function: **SPI** (default) or **I<sup>2</sup>S**
  - Synchronous, serial, full-duplex communication
  - Configurable as SPI master or slave
  - Programmable clock polarity/phase
  - Programmable baud rate
  - Supports busy-wait, interrupt, and DMA I/O
-

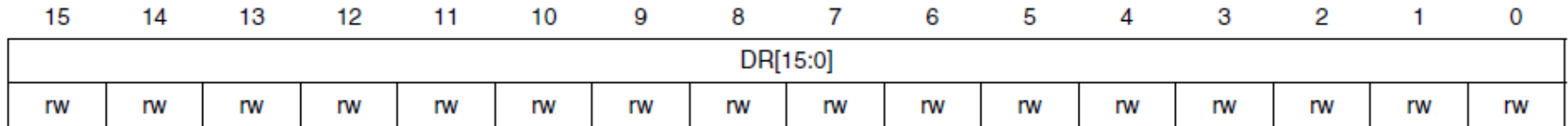
# STM32 SPI block diagram



Slave select



# SPI data register (SPI\_DR)



Transmit buffer for writing / Receive buffer for reading

SPI 8-bit data frame format (DFF = 0):

DR[7:0] = data; DR[15:8] = 00000000

SPI 16-bit data frame format (DFF = 1):

DR[15:0] = data

# SPI control register 1 (SPI\_CR1)

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:** 0 = 2-line/unidirectional, 1 = 1-line/bidirectional

**BIDIOE:** bidirectional mode output enable (0 = receive, 1 = xmit)

**CRCEN:** hardware CRC calculation enable

**CRCNEXT:** 1 = next xfer is data (no CRC), 0 = next xfer is CRC

**DFF:** data frame format (0 = 8-bit, 1 = 16-bit)

**RXONLY:** receive only (0 = full duplex, 1 = output disabled/receive-only)

**SSM:** software slave management – NSS pin ignored (1 = enable)

**SSI:** internal slave select (this bit forced onto NSS pin if output enabled: SSOE)

**LSBFIRST:** frame format (0 = shift out MSB first, 1 = shift out LSB first)

**SPE:** SPI enable

**BR[2:0]** – baud rate control (master)  $F_{\text{baud}} = F_{\text{pclk}} / (2^{(\text{BR}+1)})$

**MSTR:** master selection (0 = slave, 1 = master)

**CPOL:** clock polarity (idle value)

**CPHA:** clock phase (0 = 1<sup>st</sup> clk transition to capture data, 1 = 2<sup>nd</sup>)

# SPI control register 2 (SPI\_CR2)

8	7	6	5	4	3	2	1	0
	TXEIE	RXNEIE	ERRIE	FRF	Res.	SSOE	TXDMAEN	RXDMAEN
	rw	rw	rw	rw		rw	rw	rw

**TXEIE:** Tx buffer empty interrupt enable (on TXE flag set)

**RXNEIE:** Rx buffer not empty interrupt enable (on RXNE flag set)

**ERRIE:** error interrupt enable (CRCERR, OVR, MODF in SPI mode)

**FRF:** frame format (0 = Motorola mode, 1 = TI mode)

**SSOE:** SS output enable (if in Master mode)

**TXDMAEN:** Tx buffer DMA enable (DMA request when TXE flag set)

**RXDMAEN:** Rx buffer DMA enable (DMA request when RXNE flag set)

DMA automatically xfers data between memory and SPI\_DR

# SPI status register (SPI\_SR)

9	8	7	6	5	4	3	2	1	0
	FRE	BSY	OVR	MODF	CRC ERR	UDR	CHSIDE	TXE	RXNE
	r	r	r	r	rc_w0	r	r	r	r

**FRE:** frame format error (for SPI TI slave mode or I2S slave mode)

**BSY:** SPI/I2S busy communicating (set/cleared by hardware)

**OVR:** overrun error – master sends before RXNE cleared by slave

**MODF:** master mode fault – master NSS pin lulled low(SPI only)

**CRCERR:** CRC error in received value (SPI only)

**UDR:** underrun error (I2S only) 1<sup>st</sup> clock before data in DR

**CHSIDE:** channel side to xmit/has been received (0 = left/1 = right) (I2S only)

**TXE:** 1 = Tx buffer empty: can load next data to buffer;  
clears on DR write

**RXNE:** 1 = Rx buffer not empty: valid received data in buffer;  
clears on DR read

- Use TXE/RXNE rather than BSY for each transmission.
- Trigger SPI interrupts with TXE, RXNE, MODF, OVR, CRCERR, FRE

# Master Operation Setup

MOSI pin = data output; MISO pin = data input.

## SPI\_CR1:

1. Select **BR[2:0]** bits to define the serial clock baud rate
2. Select **CPOL** and **CPHA** bits to define one of the four relationships between the data transfer and the serial clock.
3. Select **DFF** bit to define 8- or 16-bit data frame format
4. Select **LSBFIRST** bit to define the frame format (MSB or LSB first).
5. Set **MSTR** and **SPE** bits.
6. If the NSS pin is required in input mode, in hardware mode, connect the NSS pin to a high-level signal during the complete byte transmit sequence. In NSS software mode, set the **SSM** and **SSI** bits in the SPI\_CR1 register. If the NSS pin is required in output mode, the **SSOE** bit only should be set.
7. Select FRF bit in SPI\_CR2 to select the Motorola or TI SPI protocol.