SPI

SERIAL PERIPHERAL INTERFACE

Instituto Tecnológico de Costa Rica
Escuela de Ingeniería Electrónica
Diseño de Sistemas Digitales
Ing. Juan Scott Chaves M.S.E.E

SPI

CARACTERÍSTICAS

TRANSFERENCIA SERIAL Y SINCRÓNICA DE DATOS

Comunicación serial multipunto entre un dispositivo "master" y un dispositivo "slave".

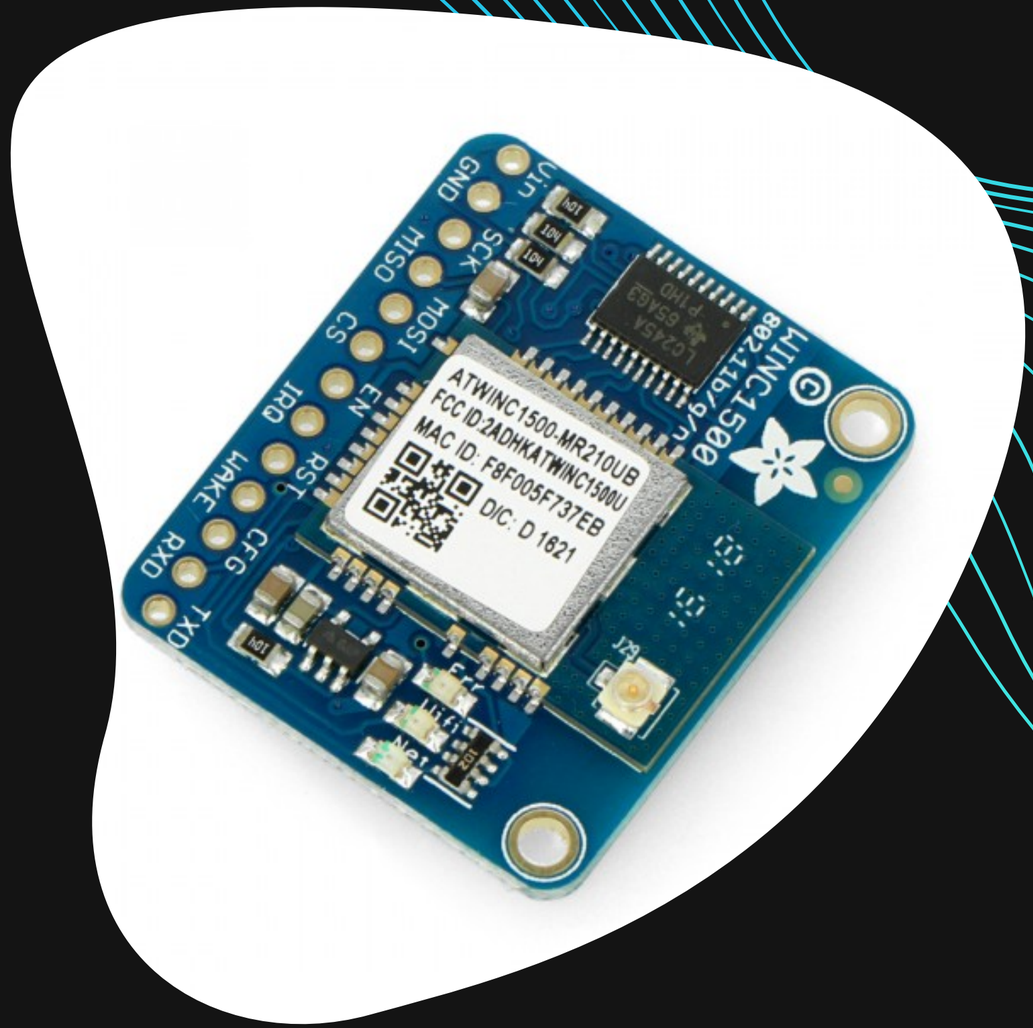
El clock permite velocidades de datos más rápidas que las comunicaciones asíncronas (no es necesario el framming).

Señales = reloj, entrada / salida de datos, "select slave".

El Master controla la transferencia de datos:

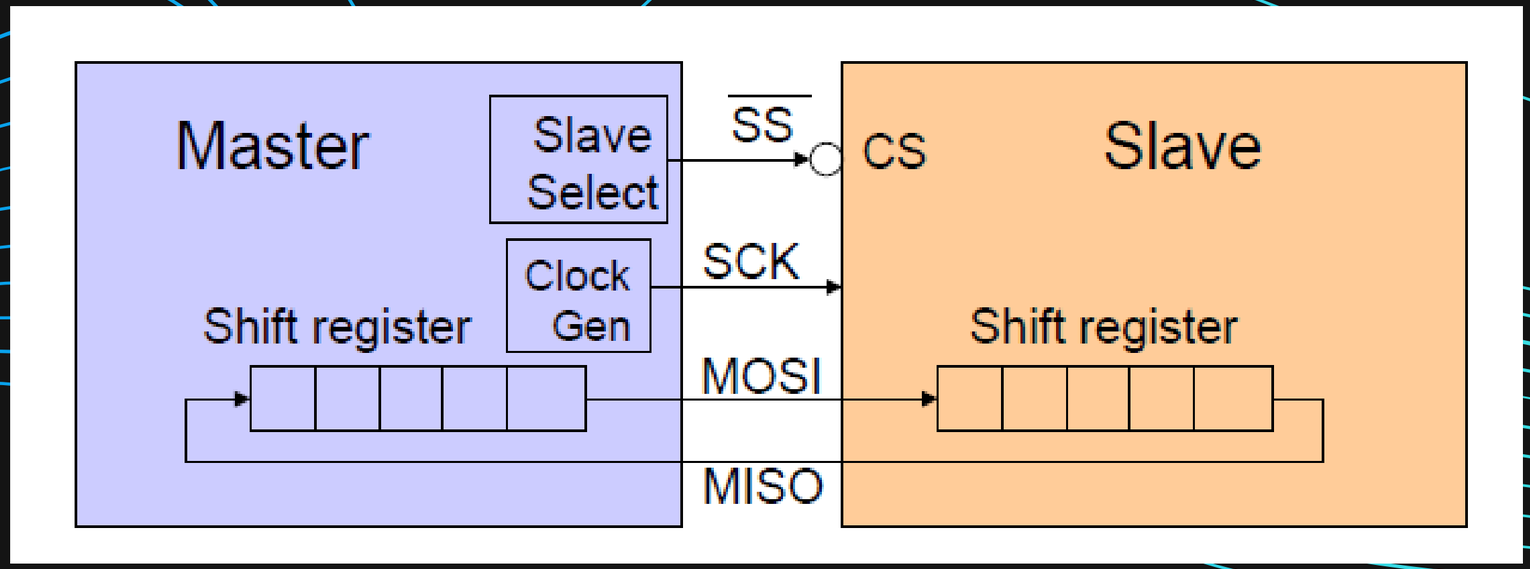
- transmite un reloj de sincronización
- activa la señal de selección de esclavo

Todos los registros de datos del dispositivo están efectivamente vinculados en un solo "shift register".



ATWINC1500 - WIFI SPI / UART + U.FL CONNECTOR

Single master, single-slave
Conexiones SPI



SCLK

CLK serial
Generado por el master

MISO

Master input
Slave output

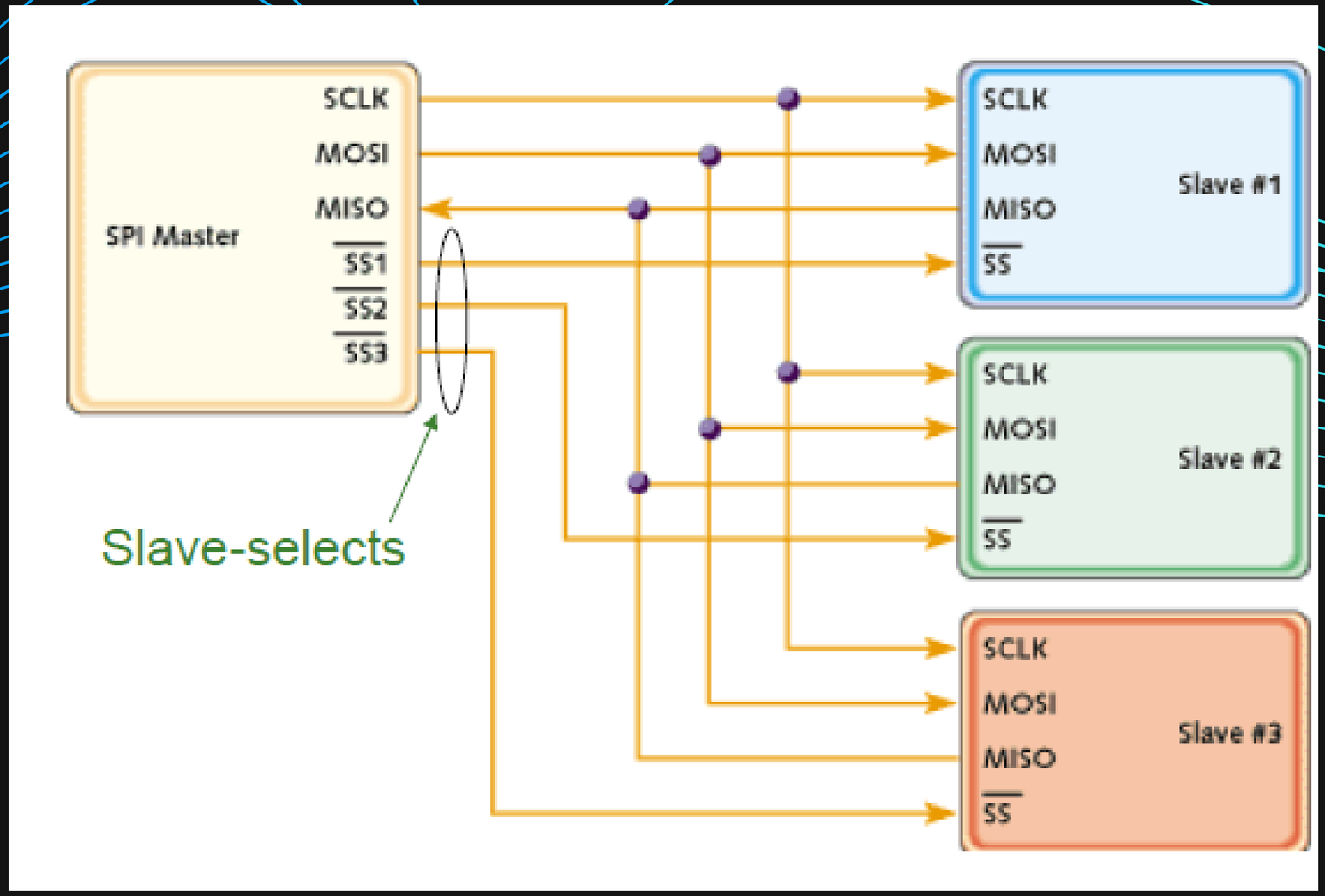
MOSI

Master output
Slave input

SS

Slave select
Enable signal

Single master, multiple slave Implementación SPI



Single master, multiple slave Daisy Chained

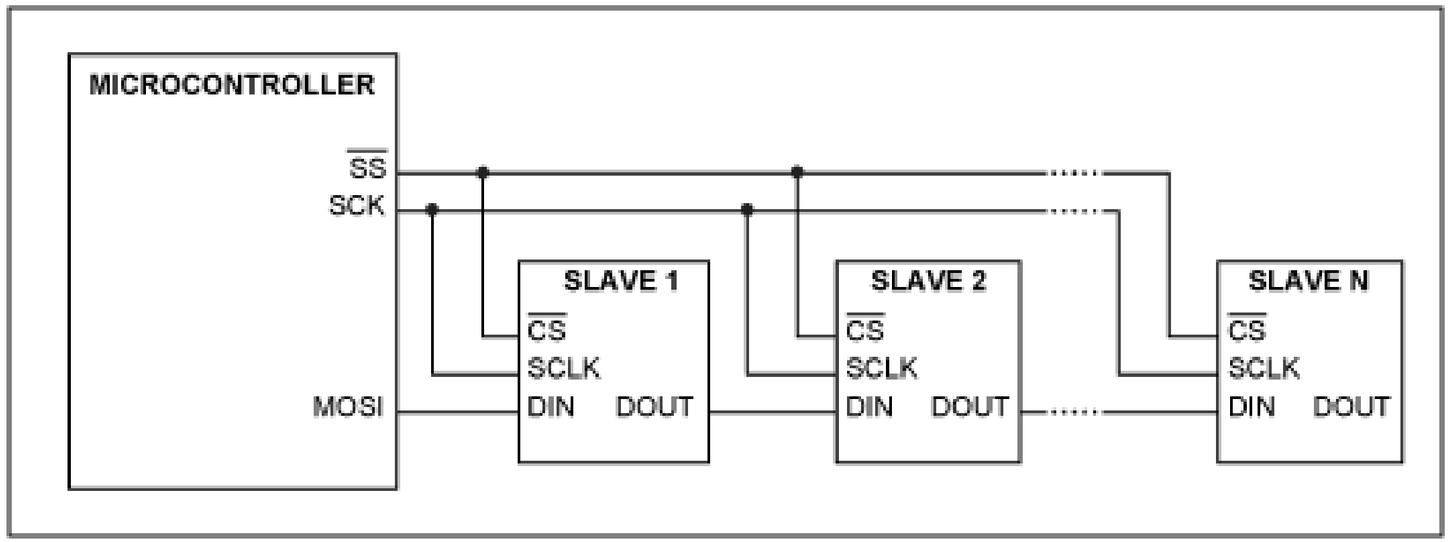


Figure 2. Microcontroller with multiple daisy-chained slave devices.

Una sola señal de active-low SS (or active-low CS) controla todas las entradas active-low CS de los esclavos.

Todos los esclavos reciben la misma señal de reloj.

Solo el primer esclavo de la cadena (SLAVE 1) recibe los datos del comando directamente desde el microcontrolador.

Todos los demás esclavos de la red reciben sus datos DIN de la salida DOUT del esclavo anterior en la cadena.

SPI Serial data timing

FRECUENCIA DE RELOJ PROGRAMABLE Y TIEMPO PARA FLEXIBILIDAD

- CPOL = polaridad del reloj (0 = activo-alto, 1 = activo-bajo)
- CPHA = fase de reloj (muestra en el borde de pulso inicial / final)

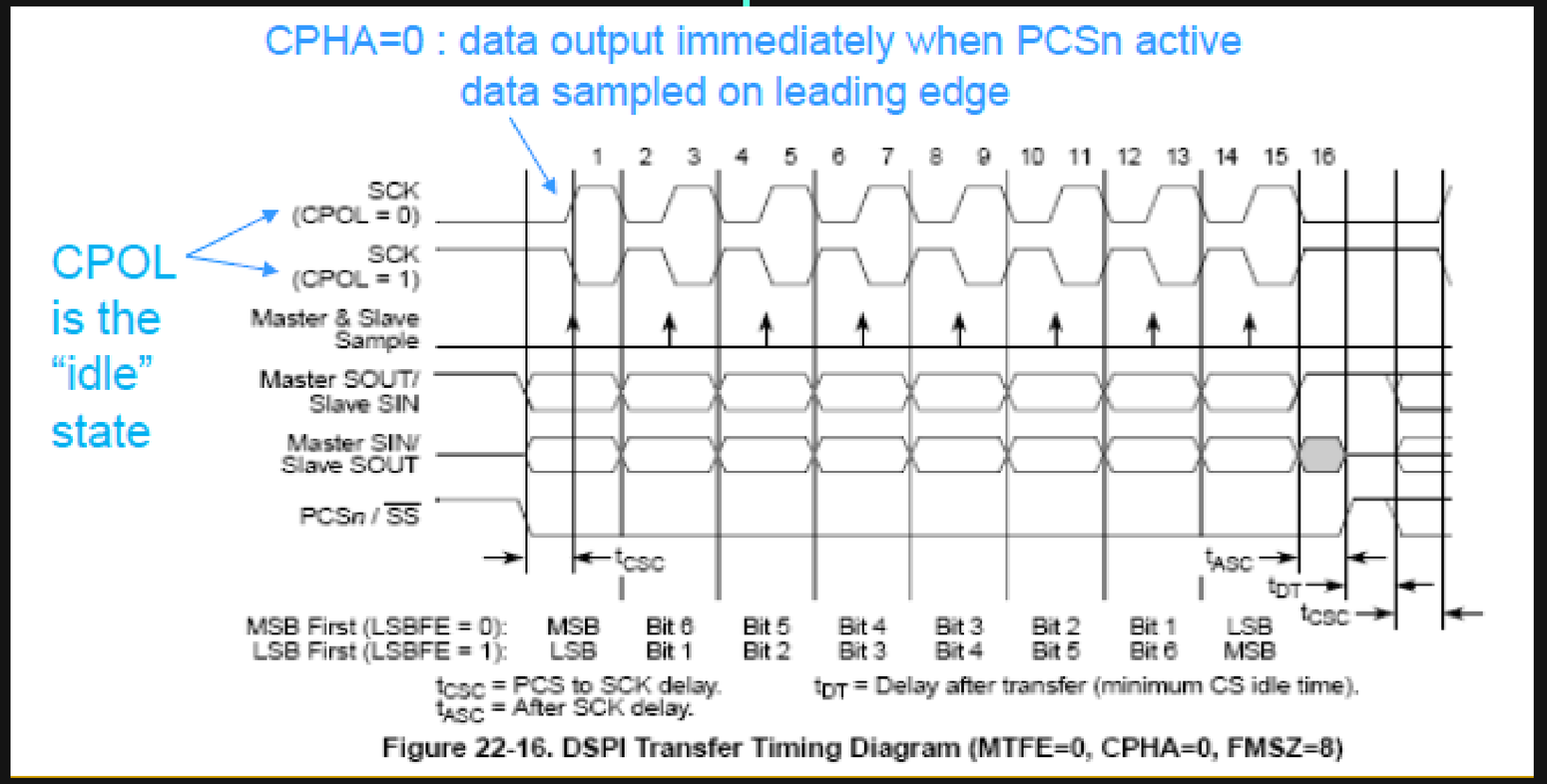


Figure 22-16. DSPI Transfer Timing Diagram (MTFE=0, CPHA=0, FMSZ=8)

SPI Serial data timing

FRECUENCIA DE RELOJ PROGRAMABLE Y TIEMPO PARA FLEXIBILIDAD

- CPHA = 1
- salida de datos en el borde del primer reloj después de PCSn activo
- datos muestreados en trailing edge CPHA = fase de reloj (muestra en el borde de pulso inicial / final)

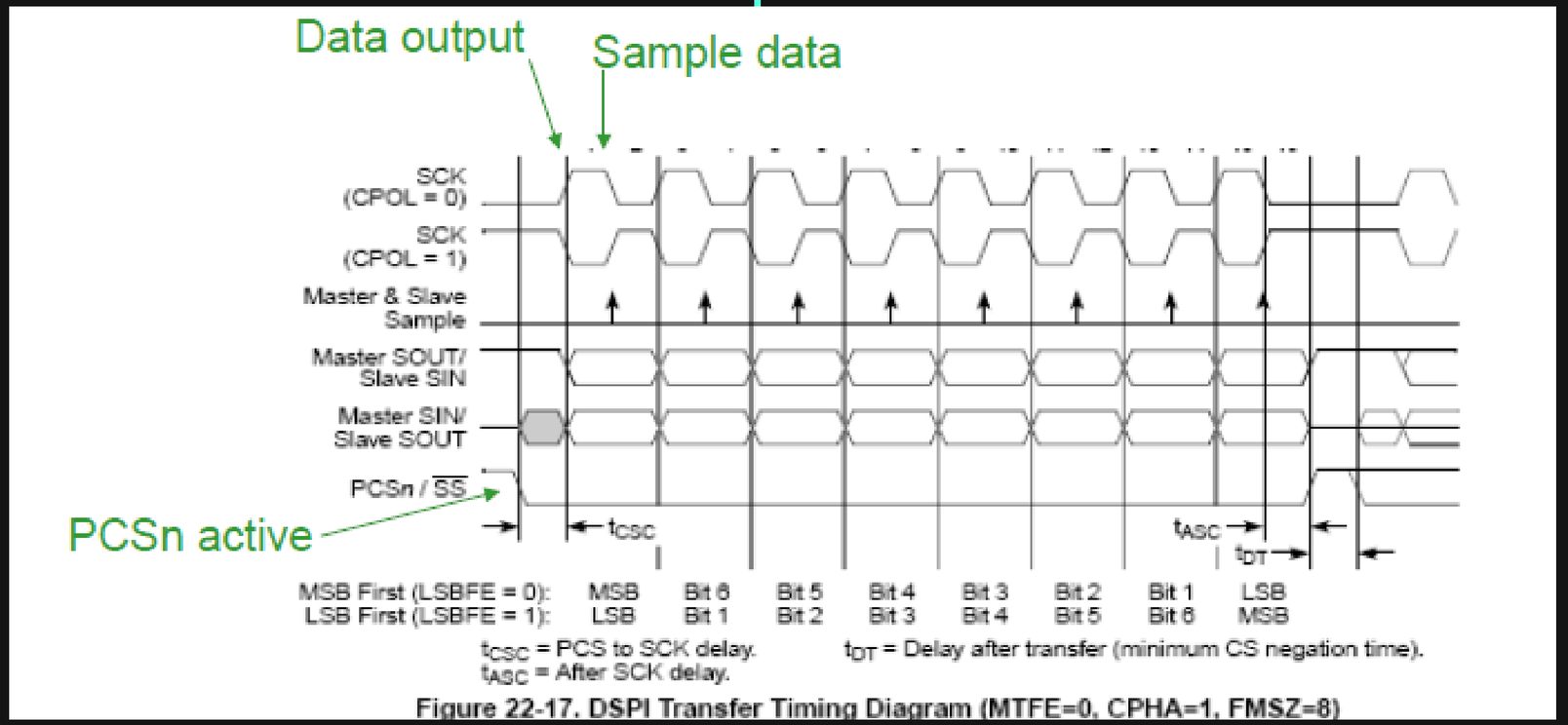


Figure 22-17. DSPI Transfer Timing Diagram (MTFE=0, CPHA=1, FMSZ=8)

STM32 Serial Peripheral Interface (SPI)

Es una familia de circuitos integrados de microcontroladores basados en el ARM RISC de 32 bits

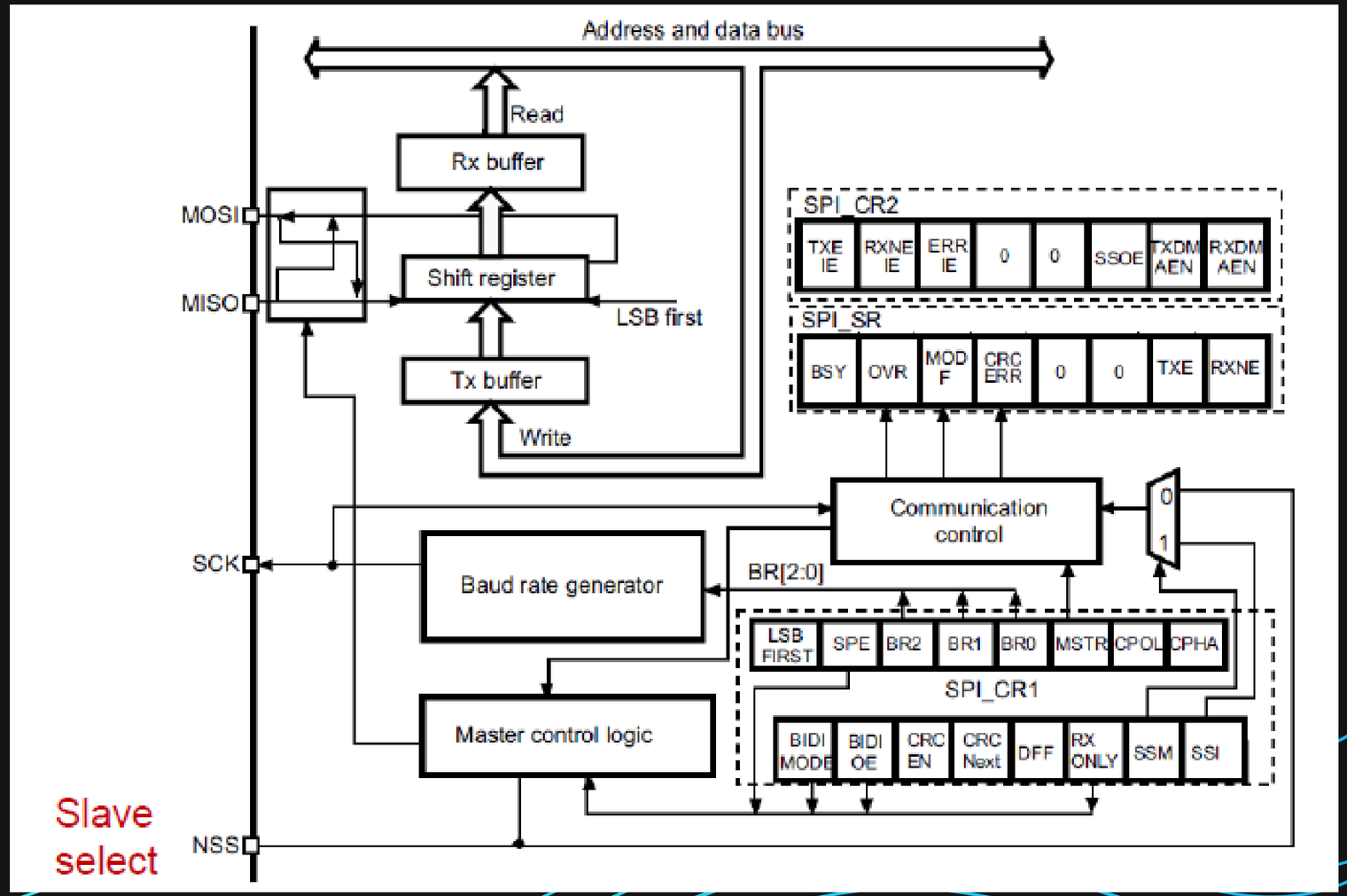
2

Función dual: SPI (default) o I2C
Comunicación sincrónica, serial, full-duplex.

Configurable como SPI master o slave
CLK programable polaridad / fase

Baud rate programable
Suporta busy-wait, interrupt, and DMA I/O

STM32 SPI diagrama de bloques



SPI data register (SPI_DR)

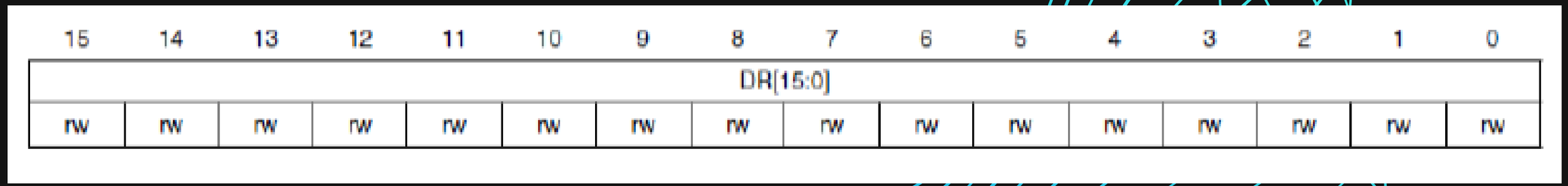
Transmit buffer for writing / Receive buffer for reading

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



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 control register 1 (SPI_CR1)

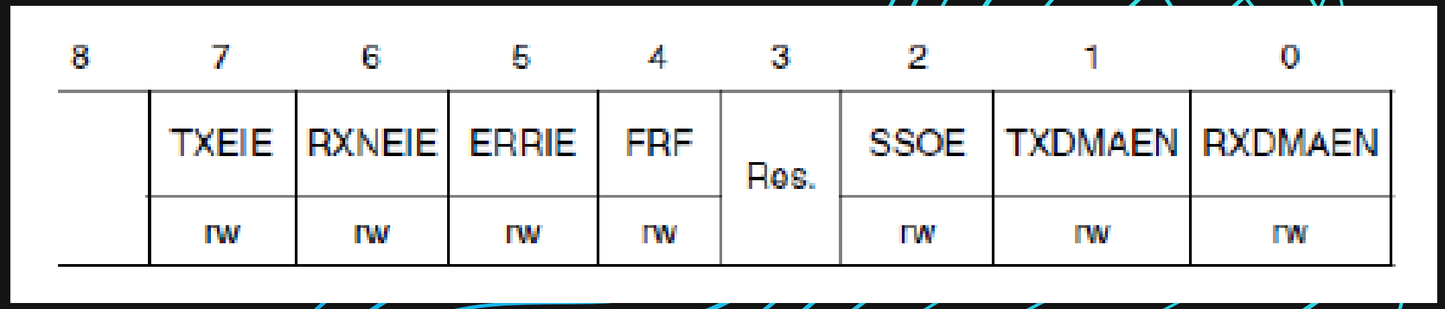
BIDIMODE: 0 = 2-line/unidirectional, 1 = 1-line/bidirectional
BIDIOE: bidirectional mode output enable (0 = receive, 1 = xmit)
CRCEN: hardware CRC calculation enable
CRCNEXT: 1 = next xferis data (no CRC), 0 = next xferis CRC
DFF: data frame format (0 = 8-bit, 1 = 16-bit)
RXONLY: receive only (0 = full 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) Fbaud= Fpclk/ (2^(BR+1))
MSTR: master selection (0 = slave, 1 = master)
CPOL: clock polarity (idle value)
CPHA: clock phase (0 = 1stclktransition to capture data, 1 = 2nd)

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

SPI control register 2 (SPI_CR2)

TXEIE: Txbuffer 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: Txbuffer DMA enable (DMA request when TXE flag set)
RXDMAEN: Rx buffer DMA enable (DMA request when RXNE flag set)

- DMA automatically xfersdata between memory and SPI_DR



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

SPI status register (SPI_SR)

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: underrunerror (I2S only) 1stclock before data in DR
CHSIDE: channel side to xmit/has been received (0 = left/1 = right) (I2S only)
TXE: 1 = Txbuffer 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

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

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 CPOLand CPHAbits to define one of the four relationships between the data transfer and the serial clock.

3.Select DFFbit to define 8-or 16-bit data frame format

4.Select LSBFIRSTbit to define the frame format (MSB or LSB first).

5.Set MSTRand SPEbits.

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 SSMand SSIbits in the SPI_CR1 register. If the NSS pin is required in output mode, the SSOEbit only should be set.

7.Select FRF bit in SPI_CR2 to select the Motorola or TI SPI protocol.