1: Next transfer is CRC (CRC phase). See STM32F4xx Ref. Manual
Bit 11 **DFF:** Data frame format
0: 8-bit data frame format is selected for transmission/reception
1: 16-bit data frame format is selected for transmission/reception
Bit 6 **SPE:** SPI enable
0: Peripheral disabled
1: Peripheral enabled
Bits 5:3 **BR[2:0]:** Baud rate control
000: fPCLK/2 100: fPCLK/32
001: fPCLK/4 101: fPCLK/64
010: fPCLK/8 110: fPCLK/128
011: fPCLK/16 111: fPCLK/256 Bit 2 **MSTR:** Master selection
0: Slave configuration
1: Master configuration
Bitl **CPOL:** Clock polarity
0: CK to 0 when idle
1: CK to 1 when idle
Bit 0 **CPHA:** Clock phase
0: The first clock transition is the first data capture edge
1: The second clock transition is the first data capture edge

**Figure 8-9: SPI Control 1 (SPI_CR1) Register in STM32F4xx**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | TXEIE | RXNEIE | ERRIE | FRF | [illegible] | SSOE | TXDMAEN | RXDMAEN |
| | | | | | | | | rw | rw | rw | rw | | rw | rw | rw |

Bit 7 **TXEIE:** Tx buffer empty interrupt enable
0: TXE interrupt masked
1: TXE interrupt not masked. Used to generate an interrupt request when the TXE flag is set. Bit 6 **RXNEIE:** RX buffer not empty interrupt enable
0: RXNE interrupt masked
1: RXNE interrupt not masked. Used to generate an interrupt request when the RXNE flag is set. Bit 5 **ERRIE:** Error interrupt enable
This bit controls the generation of an interrupt when an error condition occurs (OVR, CRCERR, MODF, FRE in SPI mode, and UDR, OVR, FRE in $I^2S$ mode).
0: Error interrupt is masked

1: Error interrupt is enabled
Bit 4 **FRF**: Frame format
0: SPI Motorola mode
1: SPI TI mode

**Figure 8-10: SPI_CR2 (Contolr 2) Registers in STM32F4xx**

When SPE is set (SPE=1), the SPI module is enabled.

***Clock Polarity and Clock Phase***

As discussed in Section 8.1, we need to choose clock polarity and clock phase that agree with the devices connected to the SPI. In the STM32F4xx, the clock polarity is determined by CPOL bit and the clock phase is determined by CPHA bit both in the SPI_CR1 register.

***Master or Slave***

***Master or Slave*** 11B. A master is a device that generates the clock and drives the SCLK pin. A slave is a device that uses the SCK pin as the SPI clock. We use MSTR (Master Selection) bit in SPI Control register 1 (SPI_CR1) to designate the SPI Module in the STM32F4xx chip as an SPI master or an SPI slave. Each module is independent of others. Some of them may be configured as masters and the others as slaves.

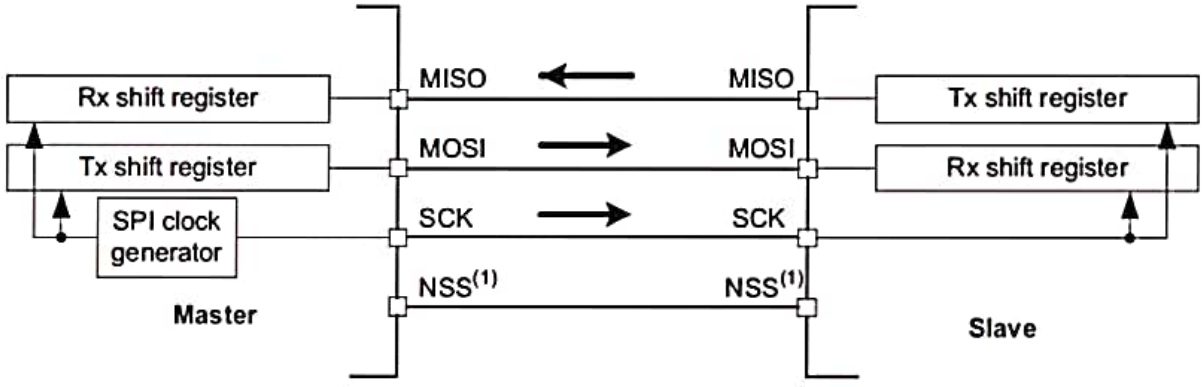


**Figure 8-11A: Using STM32F4xx SPI module as Master**

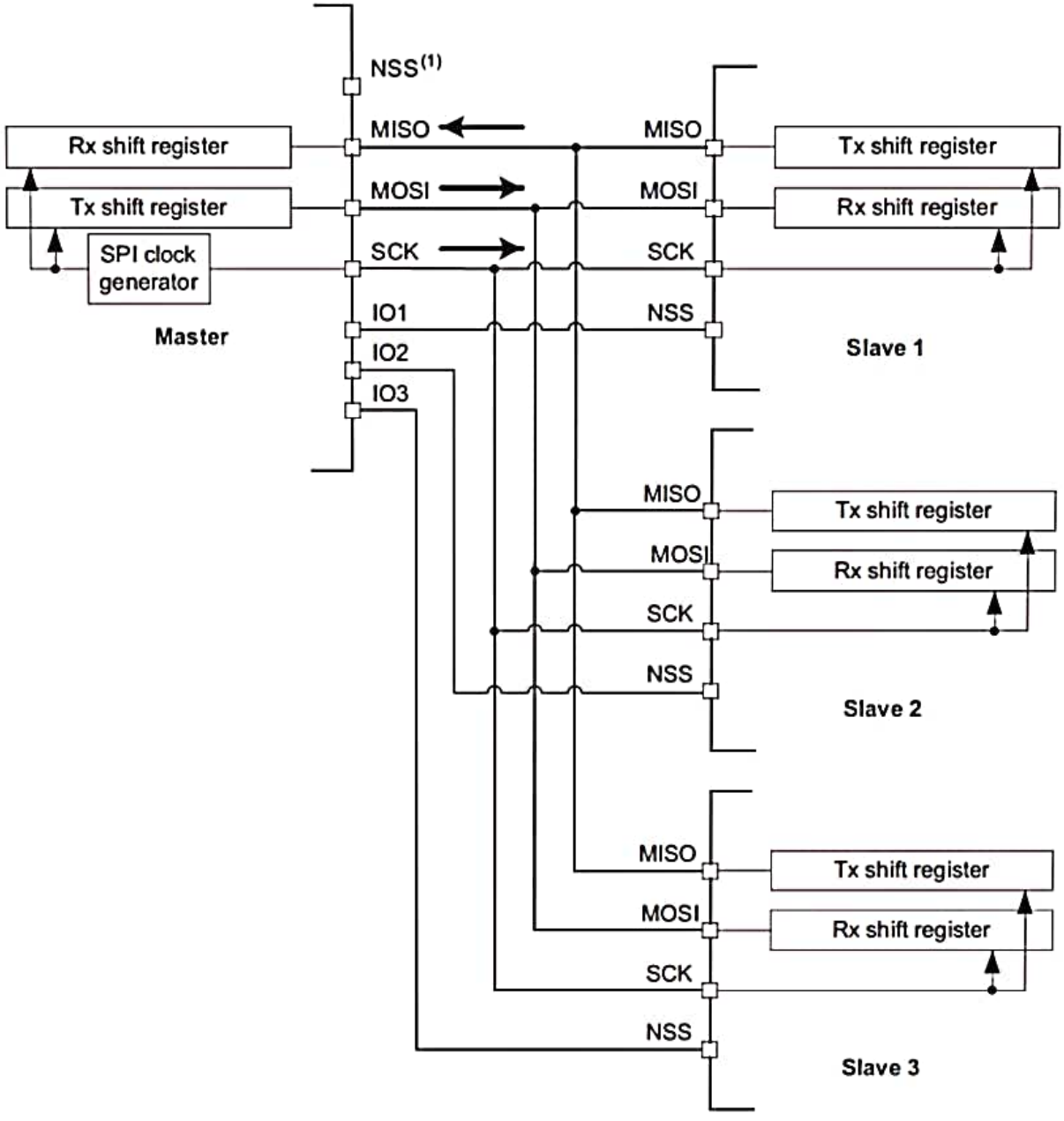


**Figure 8-11B: Using STM32F4xx SPI module as Master**

### *Setting Bit Rate*

Like some other peripherals, the SPI clock is derived from the APBx clock which comes from SYSCLK through AHB and APBx prescalers. In SPI module, the BR bits in the SPI CR1 register are used to set the clock rate. See