Serial Peripheral Interface, SPI

The SPI is a synchronous serial interface in which data in an 8-bit byte can be shifted in and/or out one bit at a time. It can be used to communicate with a serial peripheral device or with another microcontroller with an SPI interface. The SPI system in the 68HC12 contains the four signals as shown in Fig. 1.

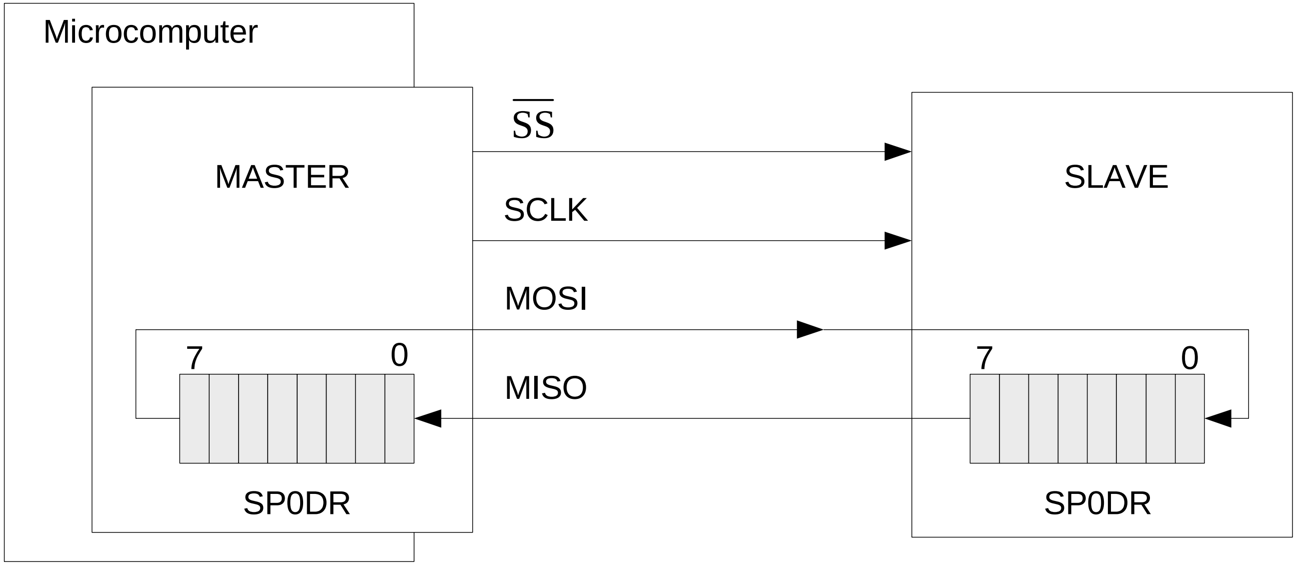

Fig. 1: Two SPI Modules Connected in a Master-Slave Configuration with 68HC12

Table 1: 68HC12 SPI Signals

Pin	SPI Signal	Name
PS4	MISO	Master-In-Slave-Out
PS5	MOSI	Master-Out-Slave-In
PS6	SCLK	Serial Clock
PS7	SS	Slave Select

Operation of the SPI

In the master SPI, the bits are sent out of the MOSI pin and received in the MISO pin. The bits to be shifted out are stored in the SPI data register, SP0DR, and are sent out most significant bit (bit 7) first. When bit 7 of the master is shifted out through MOSI pin, a bit from bit 7 of the slave is being shifted into bit 0 of the master via the MISO pin. After 8 clock pulses or shifts, this bit will eventually end up in bit 7 of the master.

Operation of the SPI

In the 68HC12 the least significant bit can be sent out first by setting the LSBF bit to 1 in the SPI Control Register. The clock, which controls how fast the bits are shifted out and into SP0DR, is the signal SCLK at PS6. The frequency of this clock can be controlled by the SPI baud rate register, SP0BR. The SS pin must be low to select a slave. This signal can come from any pin on the master, including its SS pin when it is configured as an output.

The SPI Registers

The SPI registers in the 68HC12 are shown in Table 2. The SPI registers in the 68HC11 are shown in Table 4. Note that the 68HC12 has added a second control register and a separate baud rate register. An SPI transmission is always initiated by the master, and the peripheral device is called the slave. The master initiates a transfer by storing a byte in the SPI data register (SP0DR for 6812, SPDR for 6811). The bits are shifted out as shown in Fig. 1.

Name	Register Addr.	Description
SP0CR1	00D0	SPI Control Register 1
SP0CR2	00D1	SPI Control Register 2
SP0BR	00D2	SPI Baud Rate Register
SP0SR	00D3	SPI Status Register
SP0DR	00D5	SPI Date Register

Table 2: SPI Registers for 68HC12

	7	6	5	4	3	2	1	0
$00D2	0	0	0	0	0	SPR2	SPR1	SPR0

Fig. 2: SPI Baud Rate Register, SP0BR

Table 3： 68HC12 SPI Clock Rate Selection

SPR[2:0]	Divisor	Frequency (*E*=8 MHz)
000	2	4.0 MHz
001	4	2.0 MHz
010	8	1.0 MHz
011	16	500 KHz
100	32	250 KHz
101	64	125 KHz
110	128	62.5 KHz
111	256	31.25 KHz

Table 4: SPI Registers for 68HC11

Name	Register Address	Description
SPCR	1028	SPI Control Register
SPSR	1029	SPI Status Register
SPDR	102A	SPI Data Register

Table 5： 68HC11 SPI Clock Rate Selection

SPR[1:0]	Divisor	Frequency (*E* = 2 MHz)
00	2	1.0 MHz
01	4	500 KHz
10	16	125 KHz
11	32	62.5 KHz

The SPI Registers

The clock which controls how fast the bits are shifted in or out of SP0DR is the SCLK. The frequency of this clock can be controlled by the SPI baud rate register, SP0BR, as shown in Fig. 2. Eight different frequencies can be selected with bits SPR2:SPR0, as indicates in Table 3. The 68HC11 does not have an SPI baud rate register, it uses two bits, SPR1:SPR0, in its SPI control register to select four different clock frequencies as shown in Table 5.

SPI Control Register 1

	7	6	5	4	3	2	1	0
$00D0	SPIE	SPE	SWOM	MSTR	CPOL	CPHA	SSOE	LSBF

Fig.3: SPI Control Register 1, SP0CR1

SPIE: Serial Peripheral Interrupt Enable

SPE: Serial Peripheral System Enable

SWOM: Port S Wired-OR Mode (affects PS[4:7] pins)
0–Normal CMOS outputs 1–Open-drain outputs

MSTR: Master/Slave Mode Select 0–Slave mode 1–Master mode

CPOL: Clock Polarity

CPHA: Clock Phase

SSOE: Slave Select Output Enable (master mode only)

LSBF: Least-Significant Bit Enable

The SPI Control Register 1

The SPI Control Register 1, SP0CR1, is shown in Fig. 3. The two bits CPOL and CPHA control the polarity and phase of the clock.

If CPOL=0(1), the clock idles low (high) and data are shifted in and out on the rising (falling) edge of the clock if CPHA=0 and on the falling (rising) edge of the clock if CPHA=1(0).

If CPHA=1, the SS slave select line can remain low during successive transfers. If CPHA=0, the SS line must be de-asserted and reasserted between each successive byte of data transferred.

The SPI Control Register 1

To use the SPI, the SPE bit is in the SP0CR1 must be set to 1 and to use the SPI as the master, the MSTR bit must be set to 1. Setting the SPIE bit will enable interrupts which will cause a hardware interrupt to occur when a byte of data transfer has been completed.

When 8 bits have been completely shifted out of the SP0DR, the SPIF flag (bit 7) in the SPI status register, SP0SR, shown in Fig. 4 is set to 1. This bit is cleared by reading the status register, SP0SR, followed by accessing the data register, SP0DR.

The SPI Control Register 1

The SPI control register in the 68HC11, SP0CR, is the same as the 68HC12 register, SP0CR1, shown in Fig. 3, except that bits 0 and 1 contain the two baud rate select bits, SPR1 and SPR0, given in Table 5. The bits SSOE and LSBF in Fig. 3 are available only in the 68HC12. The SSOE bit can enable an SS output mode in a master in which the SS output automatically goes low during each SPI transmission and then goes high during each idling state so that external devices are deselected. If LSBF bit is set, then data are transferred least significant bit first rather than the most significant bit first.

SPI Status Register

	7	6	5	4	3	2	1	0
$00D3	SPIF	WCOL	0	MODF	0	0	0	0

Fig.4: SPI Status Register, SP0SR

SPIF: SPI Transfer Complete Flag

0 – Cleared by SP0SR read with SPIF set, followed by SP0DR access

1 – Set upon completion of data transfer between processor and external device

WCOL: Write Collision

0 – No write collision

1 – Write collision

MODF: Mode Fault

0 – No mode fault

1 – Mode fault

The SPI Status Register

The SPI register in the 68HC11, SPSR, is the same as the 68HC12 SPI status register, SP0CR1, shown in Fig. 4. The 68HC12 SPI Control Register 2, for which there is no counterpart in 68HC11, is shown in Fig. 5. There are three bits in this control register that enable internal pull-ups on port S inputs, reduce the drive capability on port S outputs, and enable a bi-directional mode. In the bi-directional mode a single pin (MOSI for a master and MISO for a slave) can be used for both input and output.

SPI Control Register 2

	7	6	5	4	3	2	1	0
$00D1	0	0	0	0	PUPS	RDS	0	SPC0

Fig.5: SPI Control Register 2, SP0CR2

PUPS: Pull-Up Port S Enable
0 – No internal pull-ups on port S
1 – All port S input pins have an active pull-up device

RDS: Reduce Drive of Port S
0 – Normal port S output drivers
1 – Reduced drive capability on all port S outputs for lower power and less noise

SPC0: Serial Pin Control 0
0 – Normal operation
1 – Bidirectional mode

Port S

Eight, bi-directional Port S pins are shared with the serial communication interfaces (`SCI`s) (two on the 68HC812A4 and one on the 68HC912B32) and one serial peripheral interface (`SPI`). If any of these devices are enabled, the corresponding Port S bits are used for serial I/O and may not used for parallel I/O. On reset, Port S is available for use as a general purpose, 8-bit I/O register. If you wish to use any of the serial I/O features of Port S, you must enable them by setting bits in control registers.

Pull-up Control

It is a good electronic design practice to tie unused input pins to either a high or low logic level. In CMOS devices this reduces the chance for a potentially destructive condition called *latch-up* to occur. PUPS is to enable pull-up resistors on Port S which are used as inputs.

Reduced Drive

High drive current is an advantage when the output must drive a capacitive load. High drive current also results in higher speed switching between logic levels. This creates higher power consumption increases radio frequency interference.