

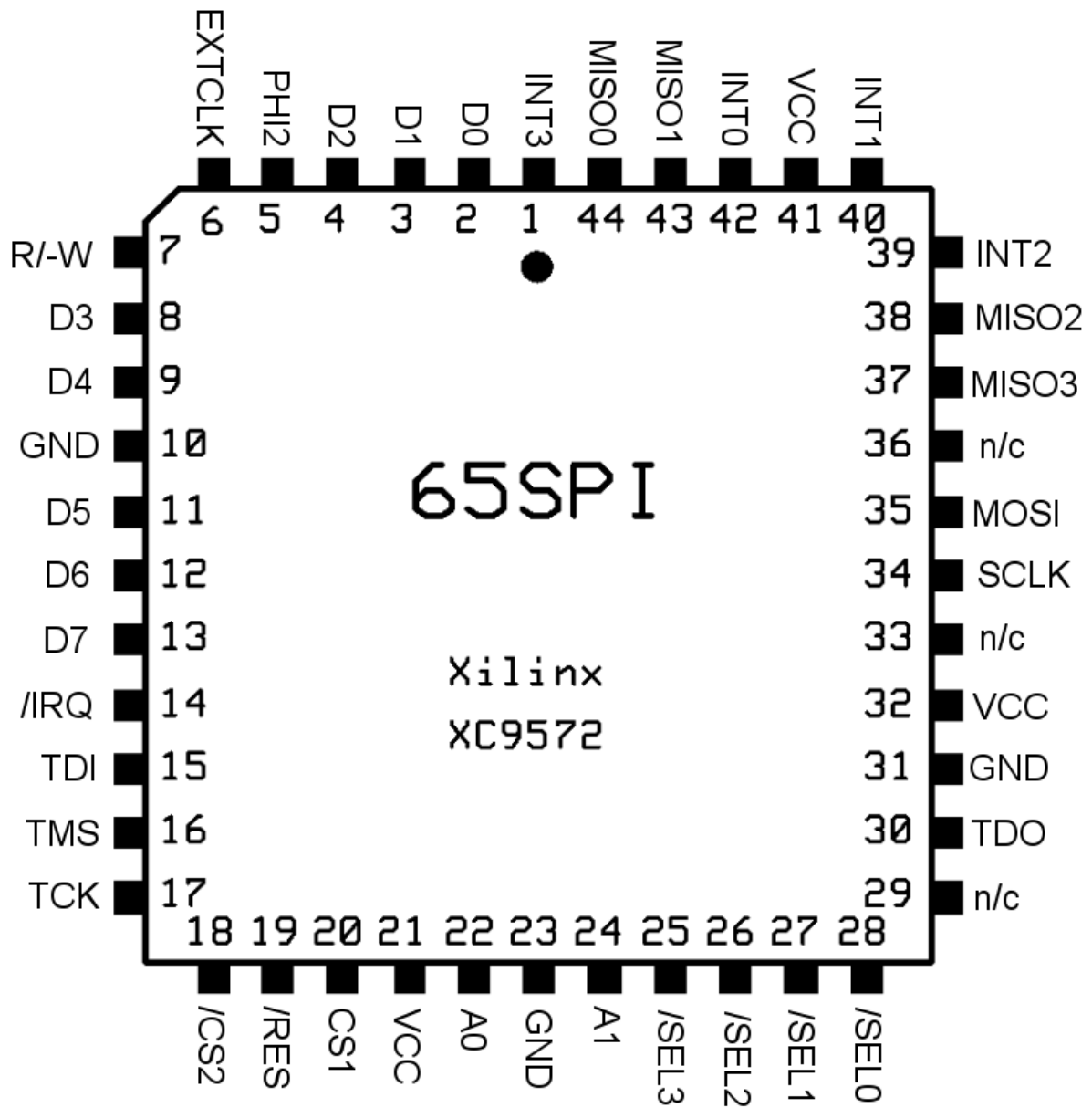
65SPI/B

An SPI interface for the 65(C)02 family of microprocessors

This device was created to provide a basic SPI interface for the 65xx family of microprocessors. Currently, the only way to provide SPI is to bit-bang it using a 6522 or equivalent device. That uses a lot of microprocessor time and program space. This device takes care of the data loading, shifting, clocking, and control - freeing the microprocessor for more important duties. There is interrupt support to allow an ISR to handle the SPI interface. The status register provides signals for polling if interrupts are not desired.

This "/B" variant is a modified version of Daryl Rictor's original 65SPI version. I have reduced the number of selects to four, and instead integrated a MISO selector for four inputs, as well as an interrupt handler for four interrupt inputs. This makes it into a single-chip solution for four SPI devices.

Chip Layout:



44 pin PLCC

Pin Descriptions

PHI2	Microprocessor system clock (input)
A0-A1	Microprocessor address bus (input)
CS1	Chip select, active high (input)
/CS2	Chip select, active low (input)
R/-W	Microprocessor data read/write line (input)
D0-D7	Microprocessor data bus (bidirectional)
/IRQ	Microprocessor interrupt line, active low (output)
/RES	Microprocessor reset line, active low (input)
EXTCLK	External shift clock (optional input)
MISO0-3	SPI Master In, Slave Out line (4 inputs, one for each device)
MOSI	SPI Master Out, Slave In line (output)
SCLK	SPI Shift Clock output
/SEL0-3	Slave Select lines (4 outputs, one for each device)
INT0-3	Interrupt input (4 inputs, one for each device)
GND	System ground
VCC	System +5vdc
n/c	No connection

Features:

- CPU bus is compatible with 65C02 and 65C816 microprocessors
- Uses 4-byte memory map for host access to registers
- Operates as an SPI master
- SCLK derived from PHI2 or an External Clock source
- SCLK has an 4 bit programmable divider – CLK/2 through CLK/16
- SPI Mode 0, 1, 2, 3 supported
- Shifts MSB first
- 4-bit Slave select register with 4 Slave Select outputs
- Automatically selects MISO line for each of the four devices
- No external logic needed to attach 4 SPI devices
- Programmable interrupt
- Interrupt or polled transmit complete flags

Register Address Map

CS1	/CS2	PHI2	A1	A0	R/W=1 (Read)	R/W=0 (Write)
0	x	x	x	x	Hi-Z	Hi-Z
x	1	x	x	x	Hi-Z	Hi-Z
x	x	0	x	x	Hi-Z	Hi-Z
1	0	1	0	0	SPI Data In	SPI Data Out
1	0	1	0	1	SPI Status	SPI Control
1	0	1	1	0	SCLK Divisor / Interrupt Status	SCLK Divisor
1	0	1	1	1	Slave Select/ Interrupt Enable	Slave Select/ Interrupt Enable

x = don't care

Register Descriptions

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
D7	D6	D5	D4	D3	D2	D1	D0	<u>SPI Data</u> (read/write)

The SPI Data register contains data to be shifted in/out to the slave devices. Reading the register will also clear the internal Transmission Completed flag and clear the external IRQ line, if enabled. Writing this register will start a shifting sequence. The transmit and receive data registers are single buffered. The receive buffer must be read before writing another byte to prevent incoming data loss. If you write a byte while shifting is in progress, the previous byte will become corrupted and the new byte will not be shifted.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TC	IER	BSY	FRX	TMO	ECE	CPOL	CPHA	<u>SPI Status</u> (read only)

The SPI Status register provides status of the following internal flags:

- TC Transmission Complete – This flag is set when the last bit has been shifted and is cleared when the SPI Data register is read.
- IER Interrupt Enable – Interrupts are enabled when this is set to 1 and disabled when set to 0.
- BSY SPI Busy – This is 1 when data is written to the SPI data register and will stay high until the last bit is shifted.
- FRX Fast Receive mode – When set to 1, fast receive mode triggers shifting upon reading or writing the SPI Data register. When set to 0, shifting is only triggered by writing the SPI data register.
- TMO Tri-state MOSI - When set to 1, the MOSI pin will be tri-stated. When set to 0, the MOSI pin will have an active output. Tri-state will allow some three-wire interfaces to work properly.

- ECE External Clock Enable – This flag displays the selected shift clock source. 0 = PHI2 and 1 = external Shift clock pin (14).
- CPOL Clock Polarity – This flag displays the shift clock polarity.
0 = Rising edge; 1 = Falling edge
- CPHA Clock Phase – This flag displays the shift clock phase.
0 = Leading edge; 1 = Trailing edge

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	IER		FRX	TMO	ECE	CPOL	CPHA

SPI Control _____ (write only)

The SPI Control register is used to select operating conditions within the SPI controller. All bits are set to 0 on power up and when /RES is low.

- IER Interrupt Enable – This bit sets the interrupt enable flag
1= interrupts enabled and 0 = interrupts disabled.
- FRX Fast Receive mode – When set to 1, fast receive mode triggers shifting upon reading or writing the SPI Data register. When set to 0, shifting is only triggered by writing the SPI data register.
- TMO Tri-state MOSI - When set to 1, the MOSI pin will be tri-stated. When set to 0, the MOSI pin will have an active output. Tri-state will allow some three-wire interfaces to work properly.
- ECE External Clock Enable – This bit sets shift clock source.
0 = PHI2 and 1 = external Shift clock pin (14).
- CPOL Clock Polarity – This bit sets the shift clock polarity.
0 = Rising edge; 1 = Falling edge
- CPHA Clock Phase – This bit sets the shift clock phase.
0 = leading edge; 1 = falling edge

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
x	x	x	x	D3	D2	D1	D0

SCLK Divisor _____ (write)

The SCLK Divisor register contains the 4-bit value used to divide the Shift Clock source. \$00 = CLK / 2 through \$1F = CLK / 32.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INT3	INT2	INT1	INT0	D3	D2	D1	D0

SCLK Divisor / Interrupt Status _____ (read)

On Read the SCLK Divisor / Interrupt Status Register is used to read the value of 4 interrupt inputs (INT0-3).

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IEN3	IEN2	IEN1	IEN0	/SEL3	/SEL2	/SEL1	/SEL0

Slave Select / Interrupt Enable (read / write)

The Slave Select / Interrupt Enable Register is used to set the 4 slave select outputs (/SEL0 – /SEL3) and the 4 interrupt enable (IEN0-3, one for each interrupt input)

Slave Interrupt Handling

The 65SPI/B has four interrupt inputs, one for each device. These interrupts can be routed to the CPU interrupt line by enabling the interrupt enable in the Slave Select / Interrupt Enable Register.

For a CPU interrupt to be active, these conditions must be fulfilled:

CPU interrupt =

```

TC && IER
|| IEN0 && INT0
|| IEN1 && INT1
|| IEN2 && INT2
|| IEN3 && INT3

```

I.e. the interrupt is internally OR'd with the shift register interrupt as known by the original 65SPI, and each of the four interrupt inputs ANDed with their enable bits.

The slave interrupt inputs INT0-3 are thus high-active.

The CPU interrupt output is Open Collector and is either high-Z or actively pulling low.

Chip Initialization

The default power on state is all register bits set to 0. This includes the data, status, control, divisor, interrupt enable registers, and even the slave select registers. The slave select registers, as they are inverted logic, should be set to 1 on reset.

Therefore a low on the /RES pin has to be applied on power on. A low on /RES sets all registers to zero, but the slave select registers, which is sets to one (high = inactive)

SPI Operation

The following is a guideline for initiating SPI communications.

1. If interrupt-driven transmission is desired, ensure an IRQ handler is enabled for the SPI interface.
2. If the TC bit in the SPI Status register is 1, then read the SPI Data register to clear it.
3. Enable Interrupts if desired, and set the Clock mode bits by writing to the SPI Control register.
4. Write the clock divisor value to the SCLK Divisor register.
5. Enable the appropriate Slave Select line by writing the Slave Select register.
6. Write first data byte to SPI Data register.
7. The TC bit will be set in the SPI Status register when shifting is complete.
8. Read the SPI Data register to get the incoming byte. The TC flag will clear after byte is read.
9. Process the incoming byte as required.
10. If you have more data to send or receive, repeat steps 6-9
11. After the last byte is received, de-select the Slave device.

Note 1. For fast transmit without polling and without receiving data, you can perform step 6 repeatedly without doing steps 7-9 as long as the SCLK rate is fast enough to keep up. When finished, do step 8 and 11 to clear the controller for the next operation.

Note 2. For fast receive without polling or sending data, Set the FRX bit in the control register and do steps 1 – 5. Now, just read the SPI data port, store the value, and repeat until you are done. The controller will automatically send the last byte written after each read. When done, set the FRX bit to 0 and do steps 8 and 11 to clear the controller for the next operation.

Note 3. For three wire interfaces, connect the MOSI and MISO pins together. Use the TMO bit in the control register to tri-state the MOSI pin during data reads, using the fast receive procedures in note 2.

Note 4. In case you want to discard the read data, for example when sending multiple bytes, you can leave out step 8 if you go back to step 6. With the /B variant both reads and writes clear the tc bit and the interrupt.

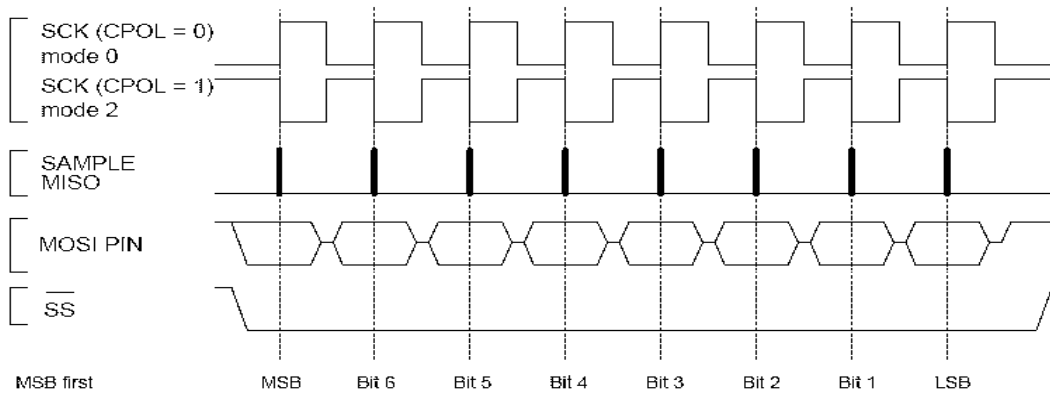
CPOL & CPHA Timing Reference

The following table describes how to configure the SPI modes:

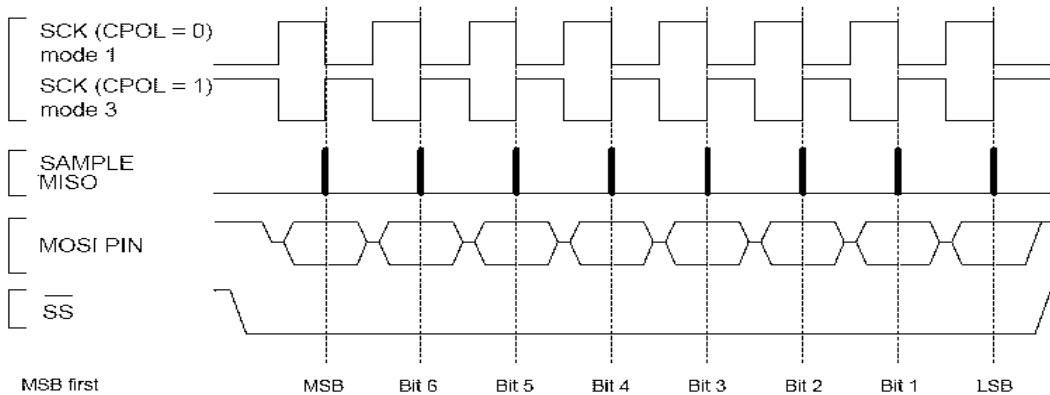
SPI Mode	CPOL	CPHA
0	0	0
1	0	1
2	1	0
3	1	1

The following diagram details the relationship between CPOL, CPHA, SCLK, and the MISO/MOSI sampling:

SPI Transfer with CPHA = 0

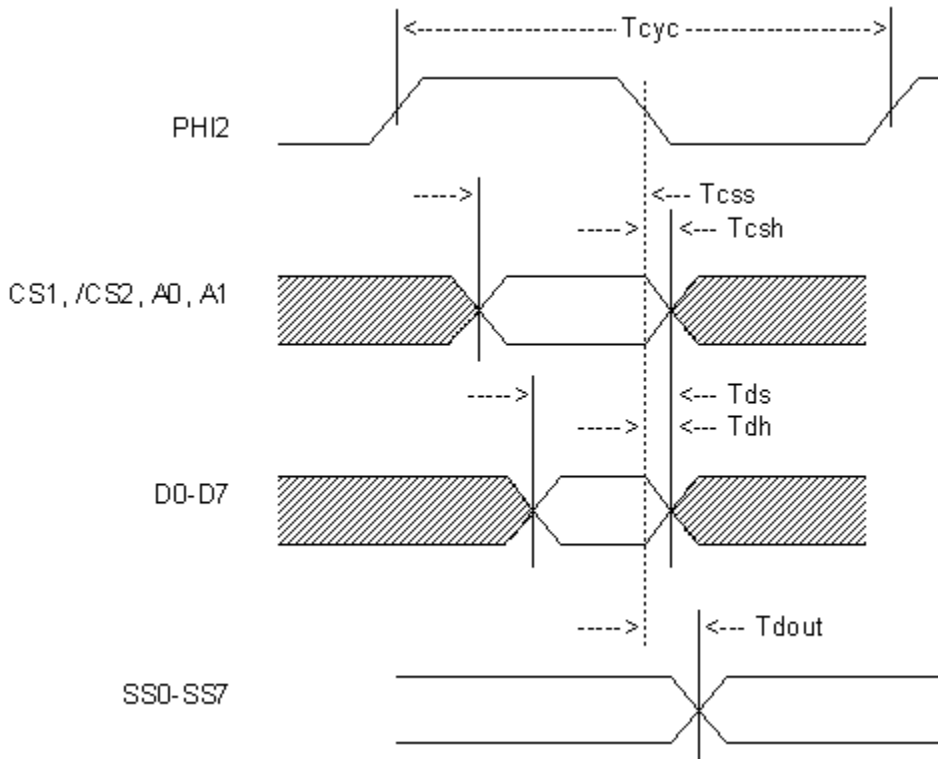


SPI Transfer with CPHA = 1



Microprocessor Bus Timing

The following diagram details the timing relationships between PHI2, CS1, /CS2, A1, A0, & R/W, and SSx:



Symbol	Description	Min	Max	Unit
	PHI2 Frequency		14	MHz
T_{cyc}	PHI2 Cycle	71		ns
T_{css}	Chip Select Setup	6		ns
T_{csh}	Chip Select Hold	0.5		ns
T_{ds}	Data setup	5		ns
T_{dh}	Data hold	0.5		ns
T_{dout}	Data out delay		15	ns
EXTC	External Clock		45	MHz

Note: these timings are from Daryl's original implementation, they need to be verified for the 65SPI/B!

Revision History

First Revisions by D. Rictor:

Rev 1 – First release	June 8, 2008
Rev 2 – Removed ENA and replaced with TDR, Transmit Data Repeat.	June 13, 2008
Rev 3 – Removed TDR and replaced with TMO, Tri-state MOSI pin. Defined EXTC max frequency	June 29, 2008
Rev 4 – Introduced the “/B” variant by A. Fachat	Dec 30, 2011