Features

- **Compatible with an Embedded ARM7TDMI™ Processor**
- **8- to 16-bit Programmable Data Length**
- **4 External Slave Chip Selects**
- **Provides Communication with External Devices in Master or Slave Mode**
- **Allows Communication Between Processors if an External Processor is Connected to the System**
- **Full Scan Testable (up to 98%)**
- **Can be Directly Connected to the Atmel Implementation of the AMBA™ Peripheral Bus (APB)**

32-Bit Embedded Core Peripheral

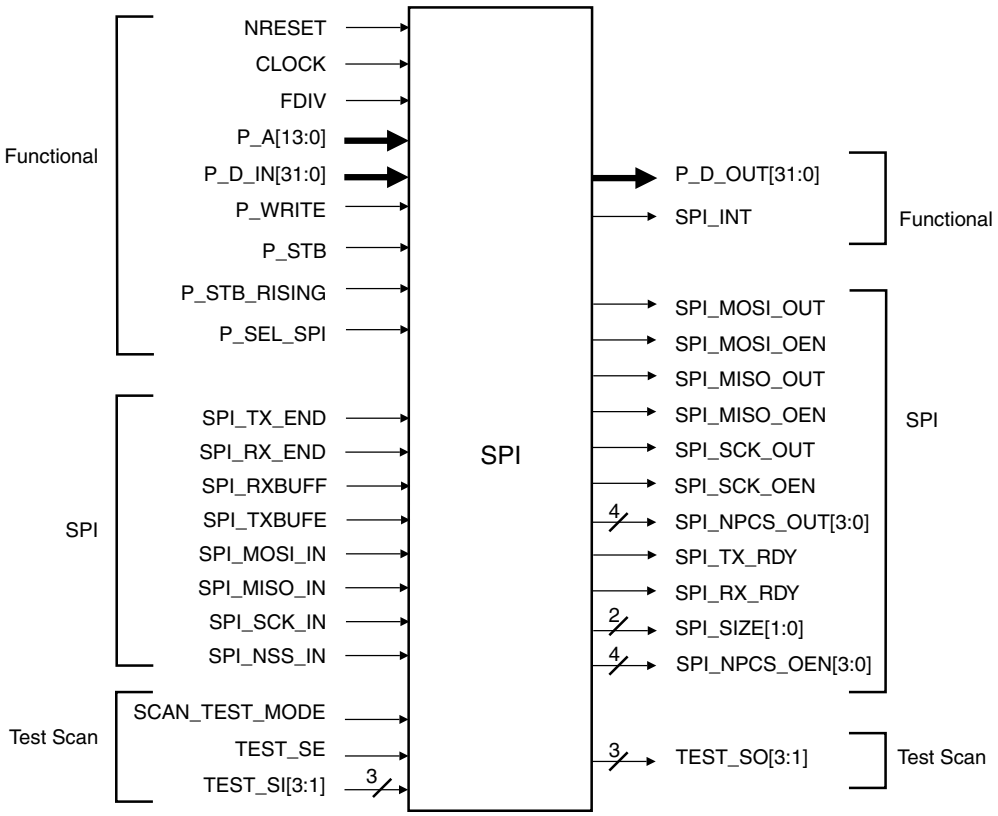
Serial Peripheral Interface (SPI)

Description

The Serial Peripheral Interface (SPI) provides communication with external devices in master or slave mode. It also allows communication between processors if an external processor is connected to the system.

The SPI can be used with any 32-bit microcontroller core if the timing diagram shown in Figure 3 on page 5 is respected. When using an ARM7TDMI as the core, the Atmel Bridge must be used to provide the correct bus interface to the peripheral.

Figure 1. SPI Symbol



Rev. 1244D-CASIC–01/03

Table 1. SPI Pin Description

Name	Function	Type	Active Level	Comments
Functional				
NRESET	Reset system	Input	Low	Resets all the counters and signals.
CLOCK	System clock	Input	---	System clock for the SPI output waveforms.
FDIV	SPI clock enable	Input	---	System clock (CLOCK) divided.
P_A[13:0]	Address bus	Input	---	The address takes into account the 2 LSBs [1:0], but the macrocell does not take these bits into account (left unconnected).
P_D_IN[31:0]	Input data bus	Input	---	From host (bridge).
P_D_OUT[31:0]	Output data bus	Output	---	To host (bridge).
P_WRITE	Write enable	Input	High	From host (bridge).
P_STB	Peripheral strobe	Input	High	From host (bridge).
P_STB_RISING	User interface clock signal	Input	---	From host (bridge). Clock for all DFFs controlling the configuration registers.
P_SEL_SPI	Selection of the block	Input	High	From host (bridge).
SPI_INT	Interrupt signal to AIC	Output	High	To Advanced Interrupt Controller (AIC).
SPI				
SPI_RX_END	End of SPI receive	Input	High	PDC/PDC2[1] generates this signal.
SPI_RXBUFF	Input signal from DMA Channel	Input	High	Generated by PDC2
SPI_TXBUFE	Input signal from DMA Channel	Input	High	Generated by PDC2
SPI_TX_END	End of SPI transfer	Input	High	PDC/PDC2[1] generates this signal.
SPI_MOSI_IN	Data slave input from MOSI pad	Input	---	Master out slave in: input dedicated to a bidir buffer.
SPI_MOSI_OUT	Data slave output to MOSI pad	Output	---	Master out slave in: output dedicated to a bidir buffer.
SPI_MOSI_OEN	Data MOSI output enable	Output	Low	Master out slave in: bidir enable
SPI_MISO_IN	Data master input from MISO pad	Input	---	Master in slave out: input dedicated to a bidir buffer.
SPI_MISO_OUT	Data slave output to MISO pad	Output	---	Master in slave out: output dedicated to a bidir buffer.
SPI_MISO_OEN	Data MISO output enable	Output	Low	Master in slave out: bidir enable
SPI_SCK_IN	Clock slave input from SCK pad	Input	---	Dedicated to a bidir buffer.
SPI_SCK_OUT	Clock master output to SCK pad	Output	---	Dedicated to a bidir buffer.
SPI_SCK_OEN	Clock output enable	Output	Low	Dedicated to a bidir buffer.
SPI_NSS_IN	Chip select input from NPCS[0] pad	Input	---	Dedicated to a bidir pad.
SPI_NPCS_OEN[3:0]	Chip select output enable	Output	Low	Dedicated to a bidir pad.
SPI_NPCS_OUT[3:0]	Chip select output to NPCS[3:0] pad	Output	---	Dedicated to a bidir pad.
SPI_TX_RDY	Transmitter ready to PDC	Output	---	PDC/PDC[1] uses this signal.

Table 1. SPI Pin Description (Continued)

Name	Function	Type	Active Level	Comments
SPI_RX_RDY	Receiver ready to PDC	Output	---	PDC/PDC2[1] uses this signal.
SPI_SIZE[1:0]	SPI transfer size to PDC (8, 16, 32)	Output	---	PDC/PDC2[1] uses this signal.
Test Scan				
SCAN_TEST_MODE	Scan test mode	Input	High	Must be set when running scan vectors.
TEST_SE	Scan test enable	Input	High/low	Scan shift /scan capture
TEST_SI[3:1]	Scan test input	Input	High	Entry of scan chain
TEST_SO[3:1]	Scan test output	Output	---	Ouput of scan chain

Note: 1. The Peripheral Data Controllers (PDC and PDC2) are separate blocks. Please refer to the corresponding datasheets.

Scan Test Configuration

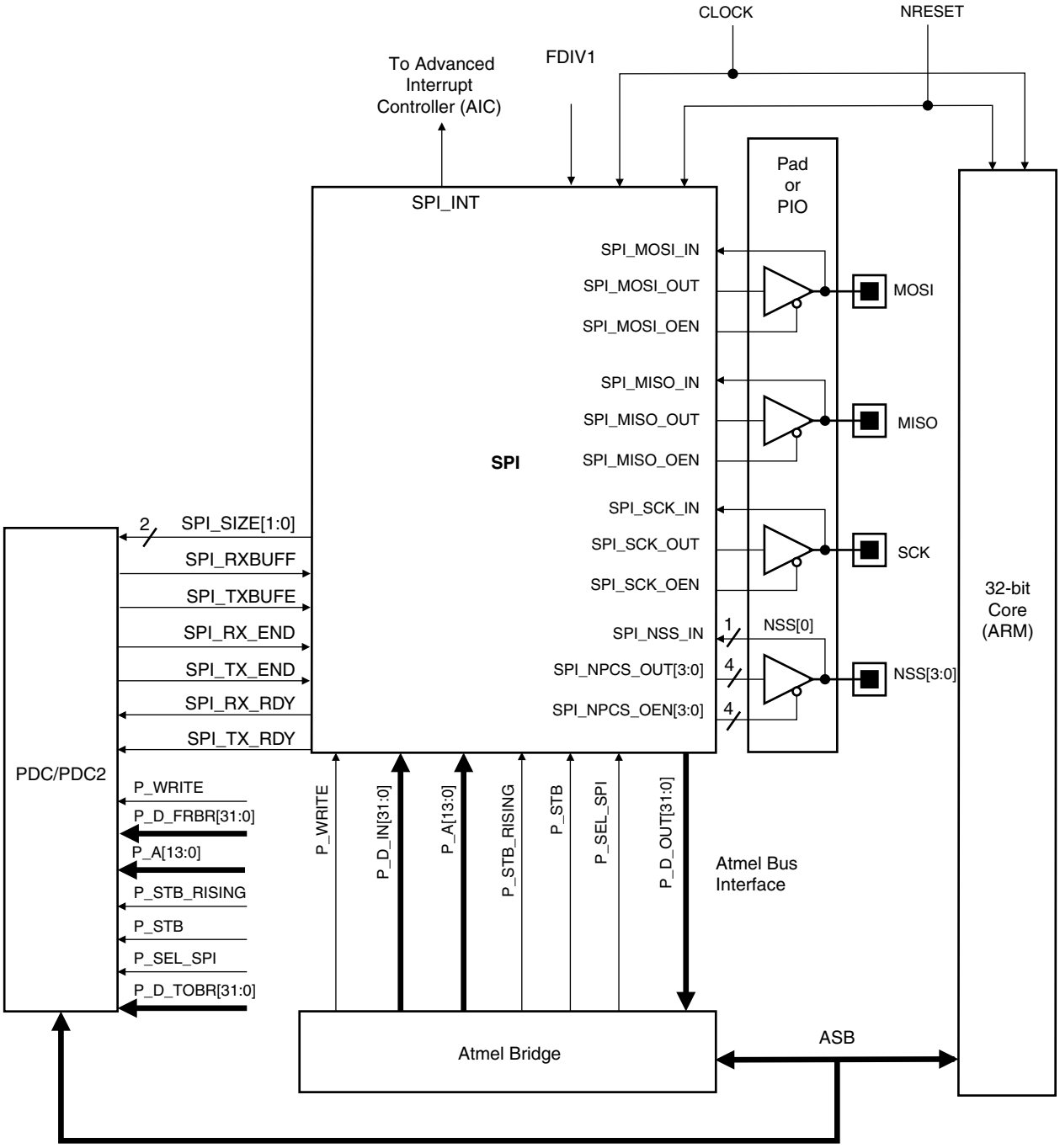
The fault coverage is maximum if all non-scan inputs can be controlled and all non-scan outputs can be observed. In order to achieve this, the ATPG vectors must be generated on the entire circuit (top-level) which includes the SPI or all SPI I/Os must have a top level access and ATPG vectors must be applied to these pins.

Peripheral Data Controller (PDC)

When the dedicated Atmel PDC is used, 4 additional registers are available in the SPI (see Table 3 on page 13). These registers are physically located in the PDC and accessed when selecting the SPI. For more details concerning these registers, please refer to the PDC datasheet.

The following pins are exclusively reserved for use with the PDC: SPI_SIZE[1:0], SPI_RX_END, SPI_TX_END, SPI_RX_RDY and SPI_TX_RDY. If the PDC is not used, SPI_RX_END and SPI_TX_END must be tied to zero.

Figure 2. Connecting the SPI to an ARM-Based Microcontroller



Seven pins are associated with the SPI Interface. Each can be connected to a bidirectional cell in a PIO or a pad.

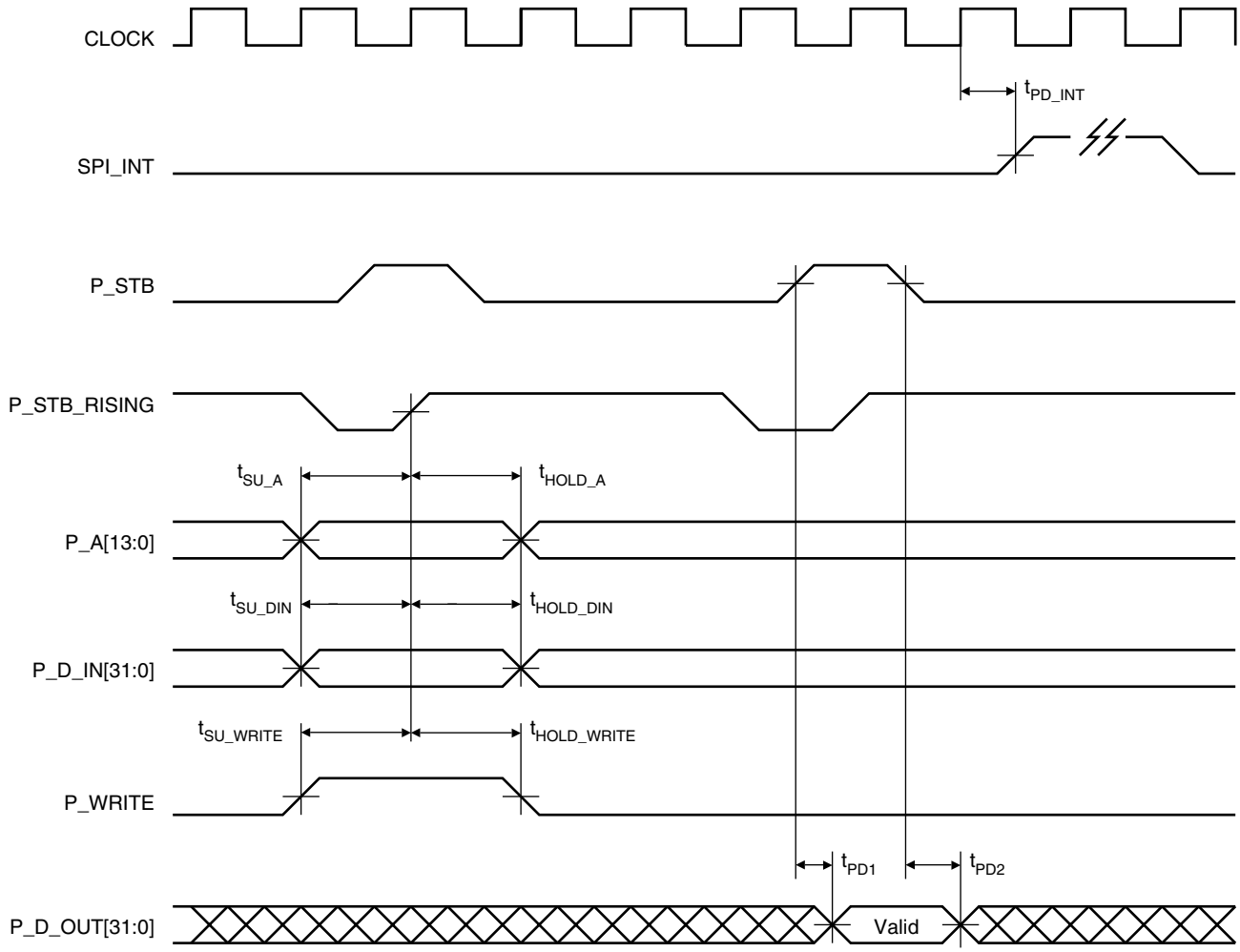
SPI_NSS_IN must be connected to NSS[0]. It can function as a peripheral chip select output or slave select input. Refer to Table 2 for a description of the SPI pins.

Table 2. SPI Pins after connection with Bidir Cells

Pin Name	Mnemonic	Mode	Function
Master In Slave Out	MISO	Master Slave	Serial data input to SPI Serial data output from SPI
Master Out Slave In	MOSI	Master Slave	Serial data output from SPI Serial data input to SPI
Serial Clock	SCK	Master Slave	Clock output from SPI Clock input to SPI
Peripheral Chip Selects	NPCS[3:1]	Master	Select peripherals
Peripheral Chip Select/ Slave Select	NPCS[0]/ NSS	Master Master Slave	Output: Selects peripheral Input: low causes mode fault Input: chip select for SPI

Timing Diagram

Figure 3. SPI Timing Diagram



Master Mode

In Master Mode, the SPI controls data transfers to and from the slave(s) connected to the SPI bus. The SPI drives the chip select(s) to the slave(s) and the serial clock (SCK). After enabling the SPI, a data transfer begins when the core writes to the SP_TDR (Transmit Data Register). See Table 3.

Transmit and Receive buffers maintain the data flow at a constant rate with a reduced requirement for high priority interrupt servicing. When new data is available in the SP_TDR (Transmit Data Register) the SPI continues to transfer data. If the SP_RDR (Receive Data Register) has not been read before new data is received, the Overrun Error (OVRES) flag is set.

The delay between the activation of the chip select and the start of the data transfer (DLYBS) as well as the delay between each data transfer (DLYBCT) can be programmed for each of the four external chip selects. All data transfer characteristics including the two timing values are programmed in registers SP_CSR0 to SP_CSR3 (Chip Select Registers). See Table 3.

In master mode the peripheral selection can be defined in two different ways:

- Fixed Peripheral Select: SPI exchanges data with only one peripheral
- Variable Peripheral Select: Data can be exchanged with more than one peripheral

Figures 4 and 5 show the operation of the SPI in Master Mode. For details concerning the flag and control bits in these diagrams, see the tables in the Programmer's Model, starting on page 13.

Fixed Peripheral Select

This mode is ideal for transferring memory blocks without the extra overhead in the transmit data register to determine the peripheral.

Fixed Peripheral Select is activated by setting bit PS to zero in SP_MR (Mode Register). The peripheral is defined by the PCS field, also in SP_MR.

This option is only available when the SPI is programmed in master mode.

Variable Peripheral Select

Variable Peripheral Select is activated by setting bit PS to one. The PCS field in SP_TDR (Transmit Data Register) is used to select the destination peripheral. The data transfer characteristics are changed when the selected peripheral changes, according to the associated chip select register.

The PCS field in the SP_MR has no effect.

This option is only available when the SPI is programmed in master mode.

Chip Selects

The Chip Select lines are driven by the SPI only if it is programmed in Master Mode. These lines are used to select the destination peripheral. The PCSDEC field in SP_MR (Mode Register) selects 1 to 4 peripherals (PCSDEC = 0) or up to 15 peripherals (PCSDEC = 1).

If Variable Peripheral Select is active, the chip select signals are defined for each transfer in the PCS field in SP_TDR. Chip select signals can thus be defined independently for each transfer.

If Fixed Peripheral Select is active, Chip Select signals are defined for all transfers by the field PCS in SP_MR. If a transfer with a new peripheral is necessary, the software must wait until the current transfer is completed, then change the value of PCS in SP_MR before writing new data in SP_TDR.

The value on the NPCS pins at the end of each transfer can be read in the SP_RDR (Receive Data Register).

By default, all NPCS signals are high (equal to one) before and after each transfer.

Mode Fault Detection

A mode fault is detected when the SPI is programmed in Master Mode and a low level is driven by an external master on the NPCS[0]/NSS signal.

When a mode fault is detected, the MODF bit in the SP_SR is set until the SP_SR is read and the SPI is disabled until re-enabled by bit SPIEN in the SP_CR (Control Register).

By default, Mode Fault Detection is enabled. It is disabled by setting the MODFDIS bit in the SPI Mode Register. See page 15.

Figure 4. Functional Flow Diagram in Master Mode

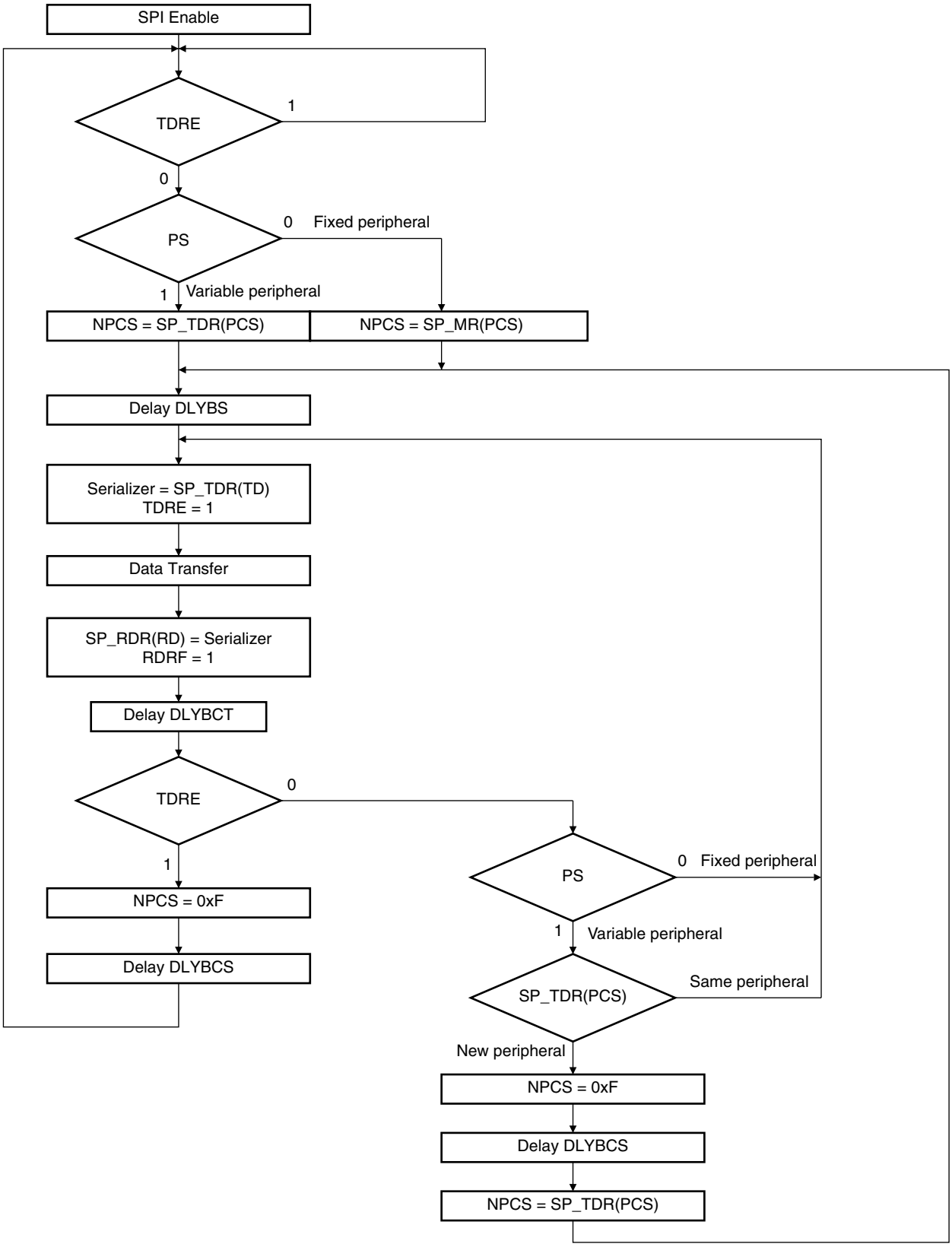
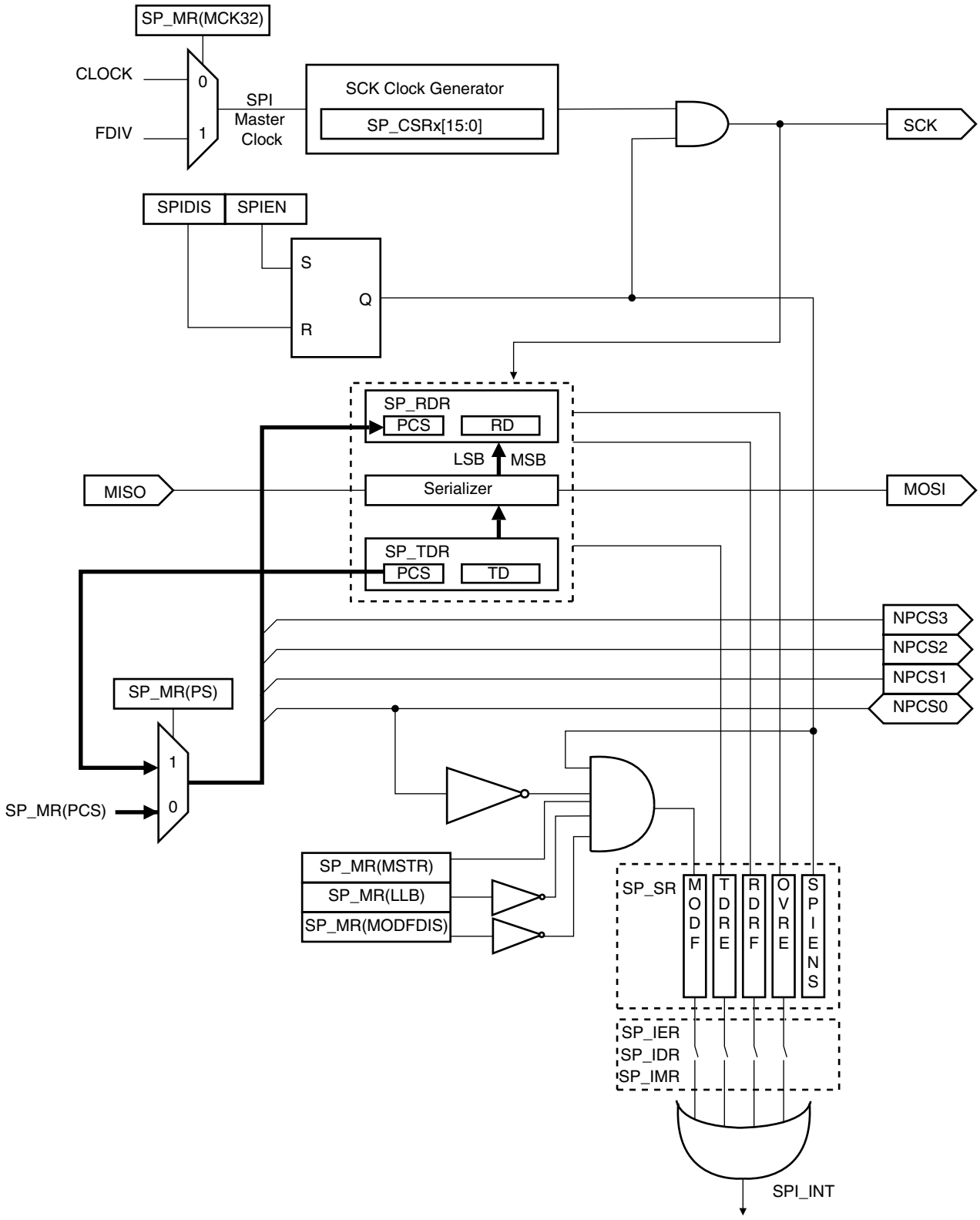


Figure 5. SPI in Master Mode

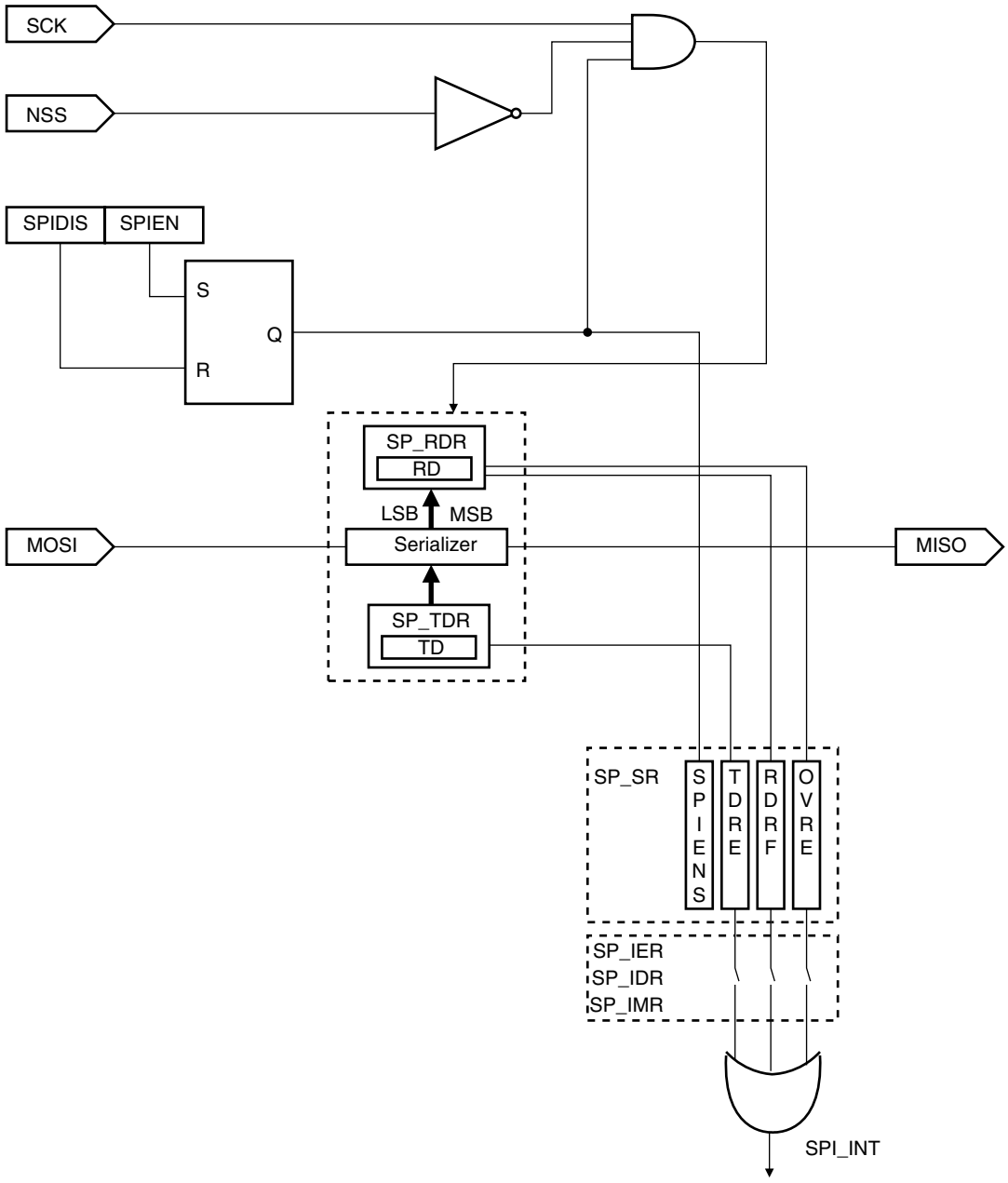


Slave Mode

In Slave Mode, the SPI waits for NSS to go active low before receiving the serial clock from an external master.

In slave mode CPOL, NCPHA and BITS fields of SP_CSR0 are used to define the transfer characteristics. The other Chip Select Registers are not used in slave mode.

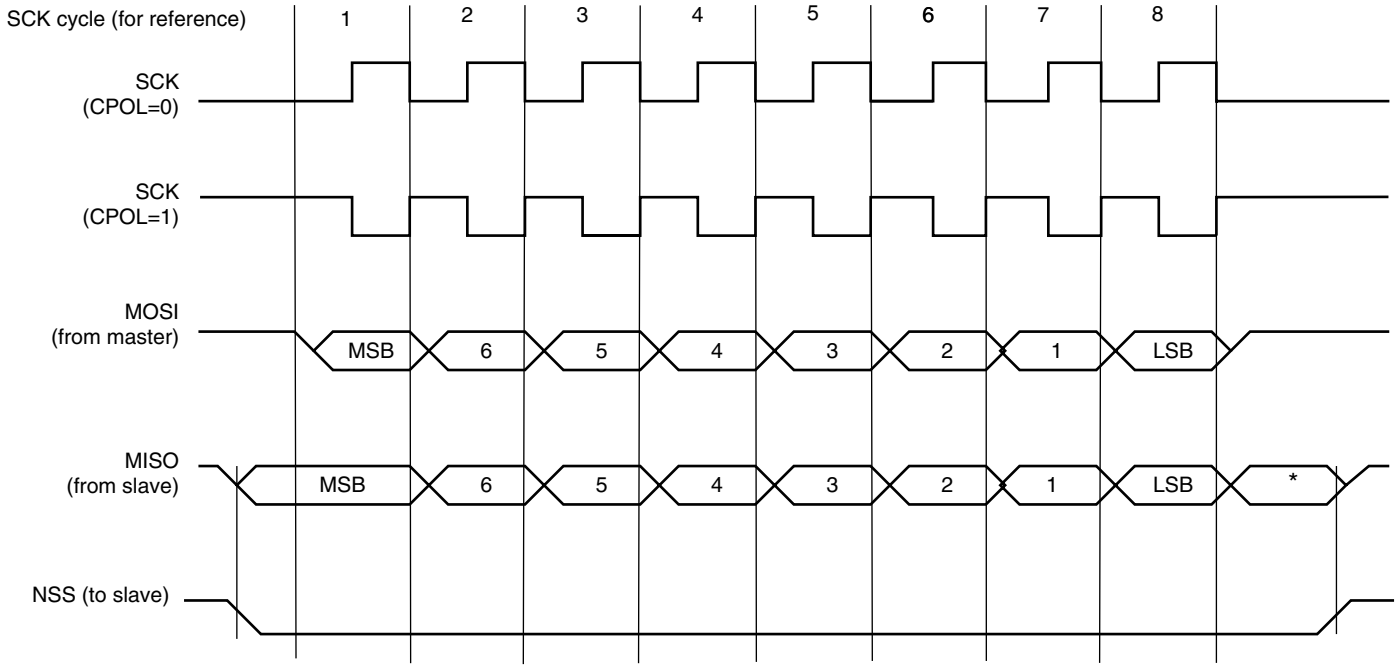
Figure 6. SPI in Slave Mode



Data Transfer

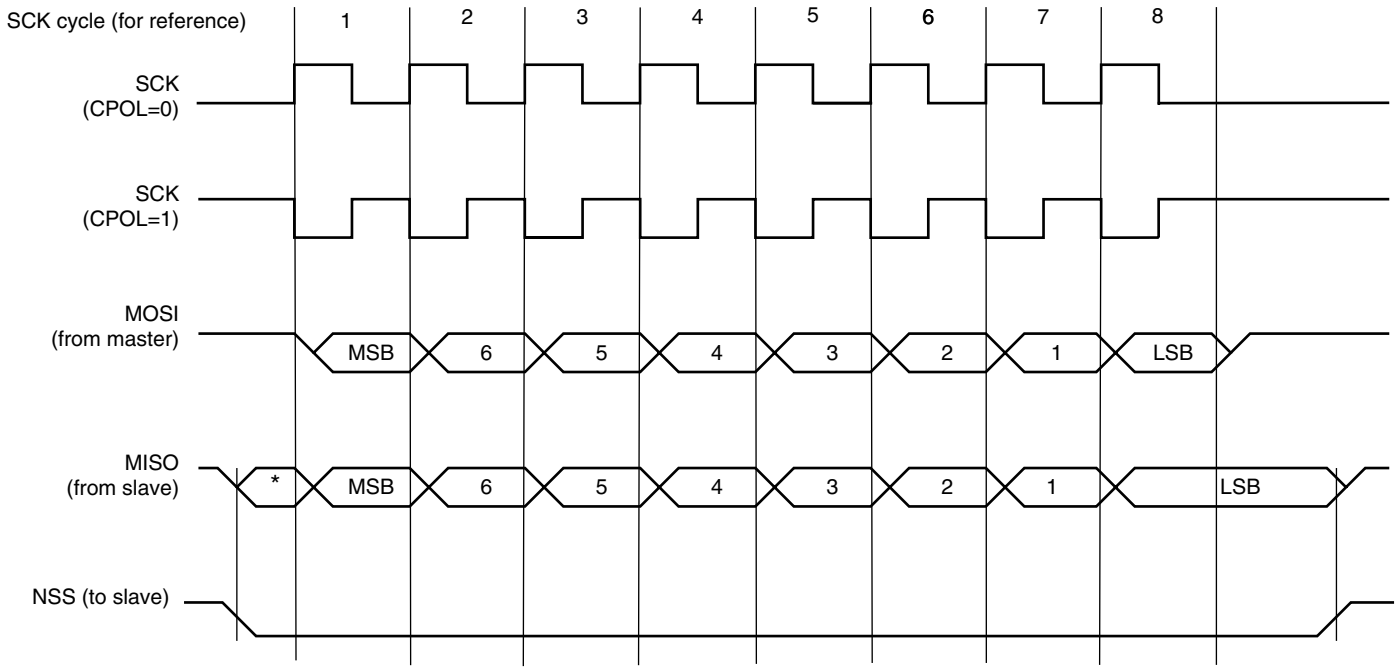
The following waveforms show examples of data transfers.

Figure 7. SPI Transfer Format (NCPHA equals One, 8 bits per transfer)



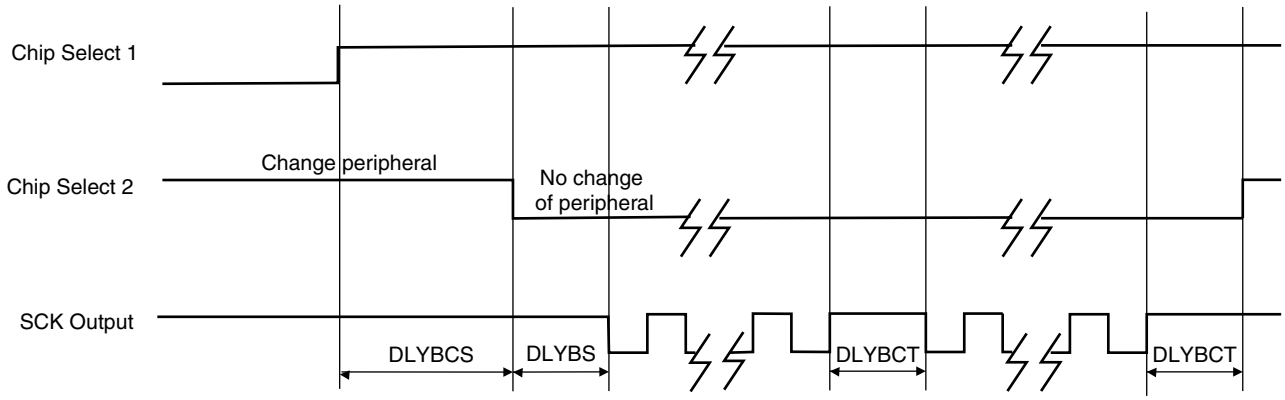
* Not defined, but normally MSB of previous character received.

Figure 8. SPI Transfer Format (NCPHA equals Zero, 8 bits per transfer)



* Not defined but normally LSB of previous character transmitted.

Figure 9. Programmable Delays (DLYBCS, DLYBS and DLYBCT)



Clock Generation

In master mode the SPI Master Clock is either CLOCK or FDIV, as defined by the MCK32 field of SP_MR. The SPI baud rate clock is generated by dividing the SPI Master Clock by a value between 4 and 510. The divisor is defined in the SCBR field in each Chip Select Register. The transfer speed can thus be defined independently for each chip select signal.

CPOL and NCPHA in the Chip Select Registers define the clock/data relationship between master and slave devices. CPOL defines the inactive value of the SCK. NCPHA defines which edge causes data to change and which edge causes data to be captured.

In Slave Mode, the input clock low and high pulse duration must strictly be longer than two system clock (CLOCK) periods.

SPI User Interface

Table 3. SPI Memory Map

Offset	Register	Name	Access	Reset State
0x0000	Control Register	SP_CR	Write only	---
0x0004	Mode Register	SP_MR	Read/Write	0
0x0008	Receive Data Register	SP_RDR	Read only	0
0x000C	Transmit Data Register	SP_TDR	Write only	---
0x0010	Status Register	SP_SR	Read only	0
0x0014	Interrupt Enable Register	SP_IER	Write only	---
0x0018	Interrupt Disable Register	SP_IDR	Write only	---
0x001C	Interrupt Mask Register	SP_IMR	Read only	0
0x0020	Reserved for PDC connection	---	---	----
0x0024				
0x0028				
0x002C				
0x0030	Chip Select Register 0	SP_CSR0	Read/Write	0
0x0034	Chip Select Register 1	SP_CSR1	Read/Write	0
0x0038	Chip Select Register 2	SP_CSR2	Read/Write	0
0x003C	Chip Select Register 3	SP_CSR3	Read/Write	0

Notes:
1. The address takes into account the 2 LSBs [1:0], but the macrocell does not take these bits into account (left unconnected). Therefore loading 0x0001, 0x0002 or 0x0003 on P_A[13:0] addresses the Control Register.
2. In the following register description, all undefined bits (“---”) read “0”.
3. If the user selects an address which is not defined in the above table, the value of P_D_OUT[31:0] is 0x00000000.

SPI Control Register

Register Name: SP_CR

Access Type: Write only

31	30	29	28	27	26	25	24
---	---	---	---	---	---	---	---
23	22	21	20	19	18	17	16
---	---	---	---	---	---	---	---
15	14	13	12	11	10	9	8
---	---	---	---	---	---	---	---
7	6	5	4	3	2	1	0
SWRST	---	---	---	---	---	SPIDIS	SPIEN

- **SPIEN: SPI Enable**

0 = No effect.

1 = Enables the SPI to transfer and receive data.

- **SPIDIS: SPI Disable**

0 = No effect.

1 = Disables the SPI.

All pins are set in input mode and no data is received or transmitted.

If a transfer is in progress, the transfer is finished before the SPI is disabled.

If both SPIEN and SPIDIS are equal to one when the control register is written, the SPI is disabled.

- **SWRST: SPI Software reset**

0 = No effect.

1 = Resets the SPI.

A software triggered hardware reset of the SPI interface is performed.

SPI Mode Register

Register Name: SP_MR

Access Type: Read/Write

31	30	29	28	27	26	25	24
DLYBCS							
23	22	21	20	19	18	17	16
---	---	---	---	PCS			
15	14	13	12	11	10	9	8
---	---	---	---	---	---	---	---
7	6	5	4	3	2	1	0
LLB	---	---	MODFDIS	MCK32	PCSDEC	PS	MSTR

- **MSTR: Master/Slave Mode**

0 = SPI is in Slave mode.

1 = SPI is in Master mode.

MSTR configures the SPI Interface for either master or slave mode operation.

- **PS: Peripheral Select**

0 = Fixed Peripheral Select.

1 = Variable Peripheral Select.

- **PCSDEC: Chip Select Decode**

0 = The chip selects are directly connected to a peripheral device.

1 = The four chip select lines are connected to a 4- to 16-bit decoder.

When PCSDEC equals one, up to 16 Chip Select signals can be generated with the four lines using an external 4- to 16-bit decoder.

The Chip Select Registers define the characteristics of the 16 chip selects according to the following rules:

SP_CSR0 defines peripheral chip select signals 0 to 3.

SP_CSR1 defines peripheral chip select signals 4 to 7.

SP_CSR2 defines peripheral chip select signals 8 to 11.

SP_CSR3 defines peripheral chip select signals 12 to 15*.

***Note**: The 16th state corresponds to a state in which all chip selects are inactive. This allows a different clock configuration to be defined by each chip select register.*

- **MCK32: Clock Selection**

0 = SPI Master Clock equals CLOCK.

1 = SPI Master Clock equals FDIV.

- **MODFDIS: Mode Fault Detection Disable**

0 = Mode fault detection is enabled.

1 = Mode fault detection is disabled.

- **LLB: Local Loopback Enable**

0 = Local loopback path disabled

1 = Local loopback path enabled

LLB controls the local loopback on the data serializer for testing in master mode only.

- **PCS: Peripheral Chip Select**

This field is only used if Fixed Peripheral Select is active (PS=0).

If PCSDEC=0:

PCS = xxx0 NPCS[3:0] = 1110

PCS = xx01 NPCS[3:0] = 1101

PCS = x011 NPCS[3:0] = 1011

PCS = 0111 NPCS[3:0] = 0111

PCS = 1111 forbidden (no peripheral is selected)

(x = don't care)

If PCSDEC=1:

NPCS[3:0] output signals = PCS

- **DLYBCS: Delay Between Chip Selects**

This field defines the delay from NPCS inactive to the activation of another NPCS. The DLYBCS time guarantees non-overlapping chip selects and solves bus contentions in case of peripherals having long data float times.

If DLYBCS is less than or equal to six, six SPI Master Clock periods will be inserted by default.

Otherwise, the following equation determines the delay:

NPCS_to_SCK_Delay = DLYBCS * SPI_Master_Clock_period

SPI Receive Data Register

Register Name: SP_RDR

Access Type: Read Only

31	30	29	28	27	26	25	24
---	---	---	---	---	---	---	---
23	22	21	20	19	18	17	16
---	---	---	---	PCS			
15	14	13	12	11	10	9	8
RD							
7	6	5	4	3	2	1	0
RD							

- **RD: Receive Data**

Data received by the SPI Interface is stored in this register right-justified. Unused bits read zero.

- **PCS: Peripheral Chip Select Status**

In Master Mode only, these bits indicate the value on the NPCS pins at the end of a transfer. Otherwise, these bits read zero.

SPI Transmit Data Register

Register Name: SP_TDR

Access Type: Write Only

31	30	29	28	27	26	25	24
---	---	---	---	---	---	---	---
23	22	21	20	19	18	17	16
---	---	---	---	PCS			
15	14	13	12	11	10	9	8
TD							
7	6	5	4	3	2	1	0
TD							

- **TD: Transmit Data**

Data which is to be transmitted by the SPI Interface is stored in this register. Information to be transmitted must be written to the transmit data register in a right-justified format.

- **PCS: Peripheral Chip Select**

This field is only used if Variable Peripheral Select is active (PS = 1).

If PCSDEC = 0:

PCS = xxx0 NPCS[3:0] = 1110

PCS = xx01 NPCS[3:0] = 1101

PCS = x011 NPCS[3:0] = 1011

PCS = 0111 NPCS[3:0] = 0111

PCS = 1111 forbidden (no peripheral is selected)

(x = don’t care)

If PCSDEC = 1:

NPCS[3:0] output signals = PCS

SPI Status Register

Register Name: SP_SR

Access Type: Read only

31	30	29	28	27	26	25	24
---	---	---	---	---	---	---	---
23	22	21	20	19	18	17	16
---	---	---	---	---	---	---	SPIENS
15	14	13	12	11	10	9	8
---	---	---	---	---	---	---	---
7	6	5	4	3	2	1	0
TXBUFE	RXBUFF	ENDTX	ENDRX	OVRES	MODF	TDRE	RDRF

- **RDRF: Receive Data Register Full**

0 = No data has been received since the last read of SP_RDR

1 = Data has been received and the received data has been transferred from the serializer to SP_RDR since the last read of SP_RDR.

- **TDRE: Transmit Data Register Empty**

0 = Data has been written to SP_TDR and not yet transferred to the serializer.

1 = The last data written in the Transmit Data Register has been transferred in the serializer.

TDRE equals zero when the SPI is disabled or at reset. The SPI enable command sets this bit to one.

- **MODF: Mode Fault Error**

0 = No Mode Fault has been detected since the last read of SP_SR.

1 = A Mode Fault occurred since the last read of the SP_SR.

- **OVRES: Overrun Error Status**

0 = No overrun has been detected since the last read of SP_SR.

1 = An overrun has occurred since the last read of SP_SR.

An overrun occurs when SP_RDR is loaded at least twice from the serializer since the last read of the SP_RDR.

- **ENDRX: End of Receiver Transfer**

0 = The End of Transfer signal from the Peripheral Data Controller channel dedicated to the receiver is inactive.

1 = The End of Transfer signal from the Peripheral Data Controller channel dedicated to the receiver is active.

- **ENDTX: End of Transmitter Transfer**

0 = The End of Transfer signal from the Peripheral Data Controller channel dedicated to the receiver is inactive.

1 = The End of Transfer signal from the Peripheral Data Controller channel dedicated to the receiver is active.

- **RXBUFF: Reception Buffer Full**

0 = PDC2 Reception Buffer is not full.

1 = PDC2 Reception Buffer is full.

- **TXBUFE: End of Receiver Transfer**

0 = PDC2 Transmission Buffer is not empty.

1 = PDC2 Transmission Buffer is empty.

- **SPIENS: SPI Enable Status**

0 = SPI is disabled.

1 = SPI is enabled.

SPI Interrupt Enable Register

Register Name: SP_IER

Access Type: Write only

31	30	29	28	27	26	25	24
---	---	---	---	---	---	---	---
23	22	21	20	19	18	17	16
---	---	---	---	---	---	---	---
15	14	13	12	11	10	9	8
---	---	---	---	---	---	---	---
7	6	5	4	3	2	1	0
TXBUFE	RXBUFF	ENDTX	ENDRX	OVRES	MODF	TDRE	RDRF

- **RDRF: Receive Data Register Full Interrupt Enable**

0 = No effect.

1 = Enables the Receiver Data Register Full Interrupt.

- **TDRE: SPI Transmit Data Register Empty Interrupt Enable**

0 = No effect.

1 = Enables the Transmit Data Register Empty Interrupt.

- **MODF: Mode Fault Error Interrupt Enable**

0 = No effect.

1 = Enables the Mode Fault Interrupt.

- **OVRES: Overrun Error Interrupt Enable**

0 = No effect.

1 = Enables the Overrun Error Interrupt.

- **ENDRX: Enable End of Receive Interrupt**

0 = No effect.

1 = Enables End of Receive Interrupt.

- **ENDTX: Enable End of Transmit Interrupt**

0 = No effect.

1 = Enables End of Transmit Interrupt.

- **RXBUFF: Enable Buffer Full Interrupt**

0 = No effect.

1 = Enables Buffer Full Interrupt.

- **TXBUFE: Enable Buffer Empty Interrupt**

0 = No effect.

1 = Disables Buffer Empty Interrupt.

SPI Interrupt Disable Register

Register Name: SP_IDR

Access Type: Write only

31	30	29	28	27	26	25	24
---	---	---	---	---	---	---	---
23	22	21	20	19	18	17	16
---	---	---	---	---	---	---	---
15	14	13	12	11	10	9	8
---	---	---	---	---	---	---	---
7	6	5	4	3	2	1	0
TXBUFE	RXBUFF	ENDTX	ENDRX	OVRES	MODF	TDRE	RDRF

- **RDRF: Receive Data Register Full Interrupt Disable**

0 = No effect.

1 = Disables the Receiver Data Register Full Interrupt.

- **TDRE: Transmit Data Register Empty Interrupt Disable**

0 = No effect.

1 = Disables the Transmit Data Register Empty Interrupt.

- **MODF: Mode Fault Interrupt Disable**

0 = No effect.

1 = Disables the Mode Fault Interrupt.

- **OVRES: Overrun Error Interrupt Disable**

0 = No effect.

1 = Disables the Overrun Error Interrupt.

- **ENDRX: Disable End of Receive Interrupt**

0 = No effect.

1 = Disables End of Receive Interrupt.

- **ENDTX: Disable End of Transmit Interrupt**

0 = No effect.

1 = Disables End of Transmit Interrupt.

- **RXBUFF: Disable Buffer Full Interrupt**

0 = No effect.

1 = Disables Buffer Full Interrupt.

- **TXBUFE: Disable Buffer Empty Interrupt**

0 = No effect.

1 = Disables Buffer Empty Interrupt.

SPI Interrupt Mask Register

Register Name: SP_IMR

Access Type: Read only

31	30	29	28	27	26	25	24
---	---	---	---	---	---	---	---
23	22	21	20	19	18	17	16
---	---	---	---	---	---	---	---
15	14	13	12	11	10	9	8
---	---	---	---	---	---	---	---
7	6	5	4	3	2	1	0
TXBUFE	RXBUFF	ENDTX	ENDRX	OVRES	MODF	TDRE	RDRF

- **RDRF: Receive Data Register Full Interrupt Mask**

0 = Receive Data Register Full Interrupt is disabled.

1 = Receive Data Register Full Interrupt is enabled.

- **TDRE: Transmit Data Register Empty Interrupt Mask**

0 = Transmit Data Register Empty Interrupt is disabled.

1 = Transmit Data Register Empty Interrupt is enabled.

- **MODF: Mode Fault Interrupt Mask**

0 = Mode Fault Interrupt is disabled.

1 = Mode Fault Interrupt is enabled.

- **OVRES: Overrun Error Interrupt Mask**

0 = Overrun Error Interrupt is disabled.

1 = Overrun Error Interrupt is enabled.

- **ENDRX: Mask End of Receive Interrupt**

0 = End of Receive Transmit Interrupt is disabled.

1 = End of Receive Transmit Interrupt is enabled.

- **ENDTX: Masl End of Transmit Interrupt**

0 = End of Transfer Interrupt is disabled.

1 = End of Transfer Interrupt is enabled.

- **RXBUFF: Enable Buffer Full Interrupt**

0 = RXBUFF Interrupt is disabled.

1 = RXBUFF Interrupt is endabled.

- **TXBUFE: Enable Buffer Empty Interrupt**

0 = TXBUFE Interrupt is disabled.

1 = TXBUFE Interrupt is enabled.

SPI Chip Select Register

Register Name: SP_CSR0.. SP_CSR3

Access Type: Read/Write

31	30	29	28	27	26	25	24
DLYBCT							
23	22	21	20	19	18	17	16
DLYBS							
15	14	13	12	11	10	9	8
SCBR							
7	6	5	4	3	2	1	0
BITS				---	---	NCPHA	CPOL

- **CPOL: Clock Polarity**

0 = The inactive state value of SCK is logic level zero.

1 = The inactive state value of SCK is logic level one.

CPOL is used to determine the inactive state value of the serial clock (SCK). It is used with NCPHA to produce a desired clock/data relationship between master and slave devices.

- **NCPHA: Clock Phase**

0 = Data is changed on the leading edge of SCK and captured on the following edge of SCK.

1 = Data is captured on the leading edge of SCK and changed on the following edge of SCK.

NCPHA determines which edge of SCK causes data to change and which edge causes data to be captured. NCPHA is used with CPOL to produce a desired clock/data relationship between master and slave devices.

- **BITS: Bits Per Transfer**

The BITS field determines the number of data bits transferred. Reserved values should not be used.

BITS[3:0]	Bits Per Transfer
0000	8
0001	9
0010	10
0011	11
0100	12
0101	13
0110	14
0111	15
1000	16
1001	Reserved
1010	Reserved
1011	Reserved
1100	Reserved
1101	Reserved
1110	Reserved
1111	Reserved

- **SCBR: Serial Clock Baud Rate**

In Master Mode, the SPI Interface uses a modulus counter to derive the SCK baud rate from the SPI Master Clock (selected between CLOCK and FDIV). The Baud rate is selected by writing a value from 2 to 255 in the field SCBR. The following equation determines the SCK baud rate:

$$SCK\_Baud\_Rate = \frac{SPI\_Master\_Clock\_frequency}{2 \times SCBR}$$

Giving SCBR a value of zero or one disables the baud rate generator. SCK is disabled and assumes its inactive state value. No serial transfers may occur. At reset, baud rate is disabled.

- **DLYBS: Delay Before SCK**

This field defines the delay from NPCS valid to the first valid SCK transition.

When DLYBS equals zero, the NPCS valid to SCK transition is 1/2 the SCK clock period.

Otherwise, the following equation determines the delay:

NPCS_to_SCK_Delay = DLYBS * SPI_Master_Clock_period

- **DLYBCT: Delay Between Consecutive Transfers**

This field defines the delay between two consecutive transfers with the same peripheral without removing the chip select. The delay is always inserted after each transfer and before removing the chip select if needed.

When DLYBCT equals zero, a delay of four SPI Master Clock periods are inserted.

Otherwise, the following equation determines the delay:

Delay_After_Transfer = 32 * DLYBCT * SPI_Master_Clock_period

Atmel Headquarters

Corporate Headquarters
2325 Orchard Parkway
San Jose, CA 95131
TEL 1(408) 441-0311
FAX 1(408) 487-2600

Europe
Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
TEL (41) 26-426-5555
FAX (41) 26-426-5500

Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Operations

Memory
2325 Orchard Parkway
San Jose, CA 95131
TEL 1(408) 441-0311
FAX 1(408) 436-4314

Microcontrollers
2325 Orchard Parkway
San Jose, CA 95131
TEL 1(408) 441-0311
FAX 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
TEL (33) 2-40-18-18-18
FAX (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards
Zone Industrielle
13106 Rousset Cedex, France
TEL (33) 4-42-53-60-00
FAX (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL 1(719) 576-3300
FAX 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
TEL (44) 1355-803-000
FAX (44) 1355-242-743

RF/Automotive
Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
TEL (49) 71-31-67-0
FAX (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL 1(719) 576-3300
FAX 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/
High Speed Converters/RF Datacom
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
TEL (33) 4-76-58-30-00
FAX (33) 4-76-58-34-80

e-mail
literature@atmel.com

Web Site
http://www.atmel.com

1244D–CASIC–01/03 0M