CSCE 317
Computer Systems Engineering

Serial Peripheral Interface

UNIVERSITY OF SOUTH CAROLINA

Bus Protocols

- Bus protocols are either:
  - Asynchronous
    - Not clocked
    - Requires handshaking protocol
    - For 328p: UART
  - Synchronous
    - Includes shared clock signal as a control signal
    - Devices communicate with a protocol that is relative to the clock
    - Example: send address in cycle 1, data is valid in cycle 4
    - For 328p: USART, SPI, I^2C

SPI (Serial Peripheral Interface)

- One master, one or more slaves

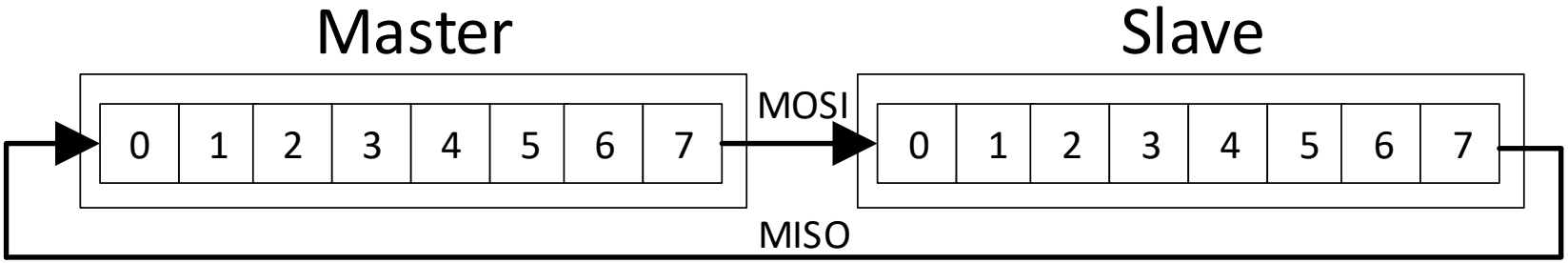
- Includes four signals:
  - SCLK: clock
  - MOSI: master out, slave in
  - MISO: master in, slave out
  - SS: slave select

- Basic idea:
  - clk is normally idle, active-low SS signal allows master to "activate" a slave
  - On each clock cycle, shift one bit out of master into slave and one bit from slave into master

SPI (Serial Peripheral Interface)

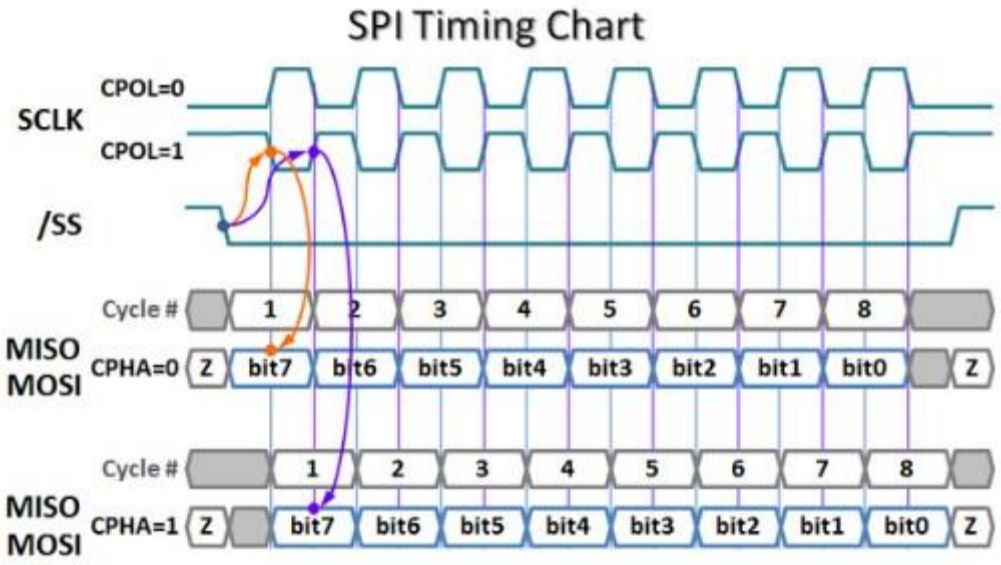
- One master, one or more slaves

- Master:
  - Drives clock, typically a few MHz
  - Selects slave by setting SS to logic low
  - During each cycle, master sends a bit to slave on MOSI and slave sends a bit to master on MISO



SPI (Serial Peripheral Interface)

- Clock polarity (CPOL) and clock phase (CPHA)
  - CPOL=0: clock idles at 0, leading edge is rising
  - CPOL=1: clock idles at 1, leading edge is falling
  - CPHA=0: data changes one-half cycle before first clock edge
  - CPHA=1: data changes on first clock edge
  - {CPOL,CPHA} determines the SPI mode



CPOL/CPHA

Figure 16-3. SPI Transfer Format with CPHA = 0

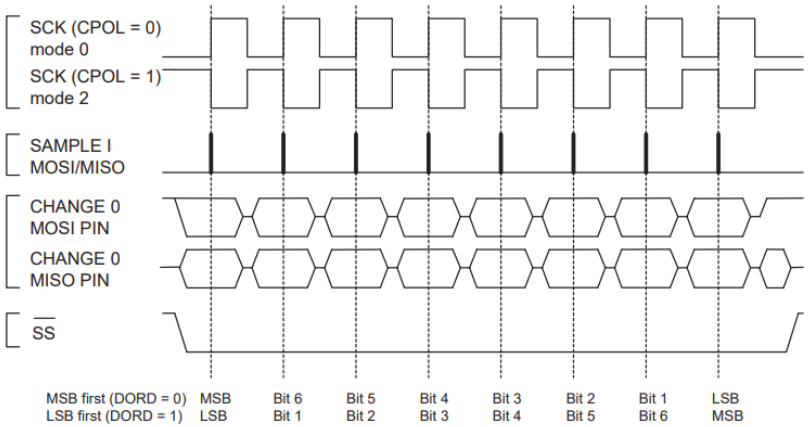
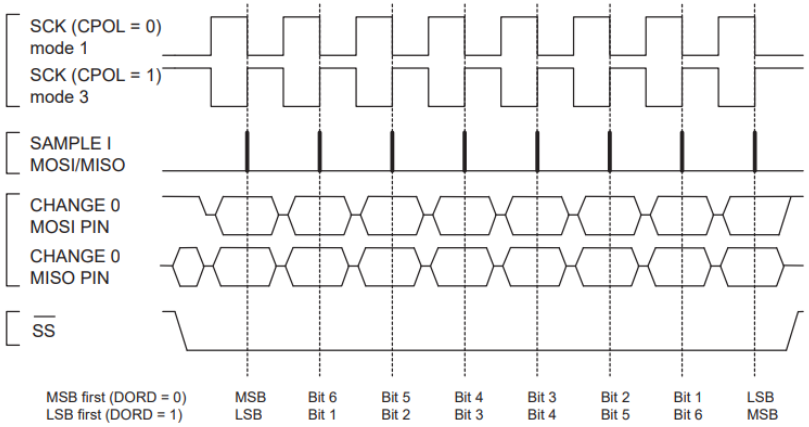
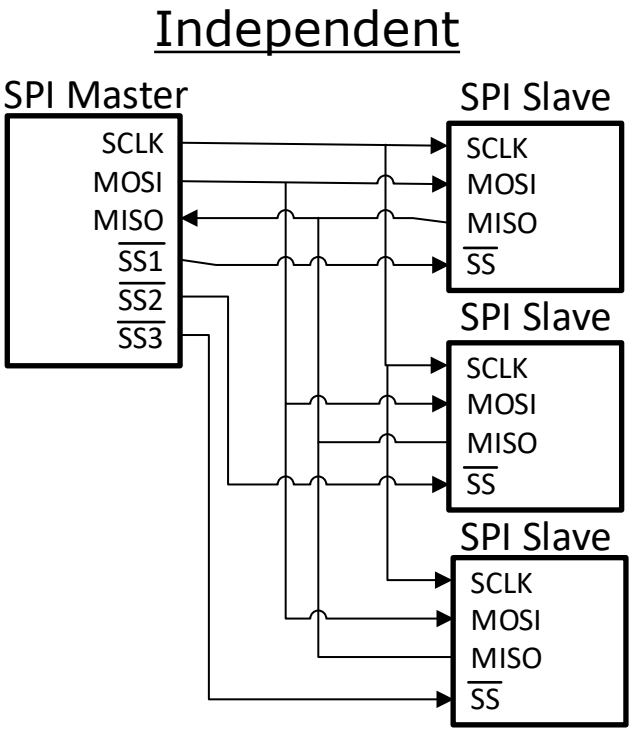


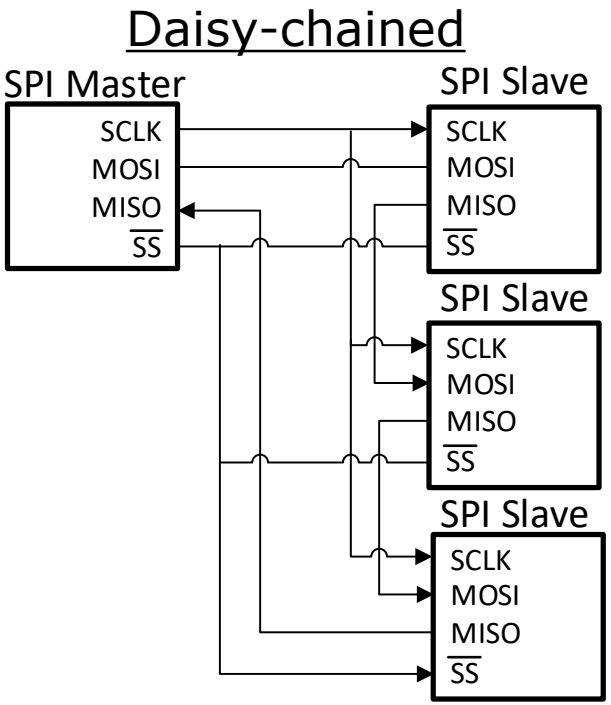
Figure 16-4. SPI Transfer Format with CPHA = 1



SPI (Serial Peripheral Interface)

- Multiple slaves:
  - Two options:

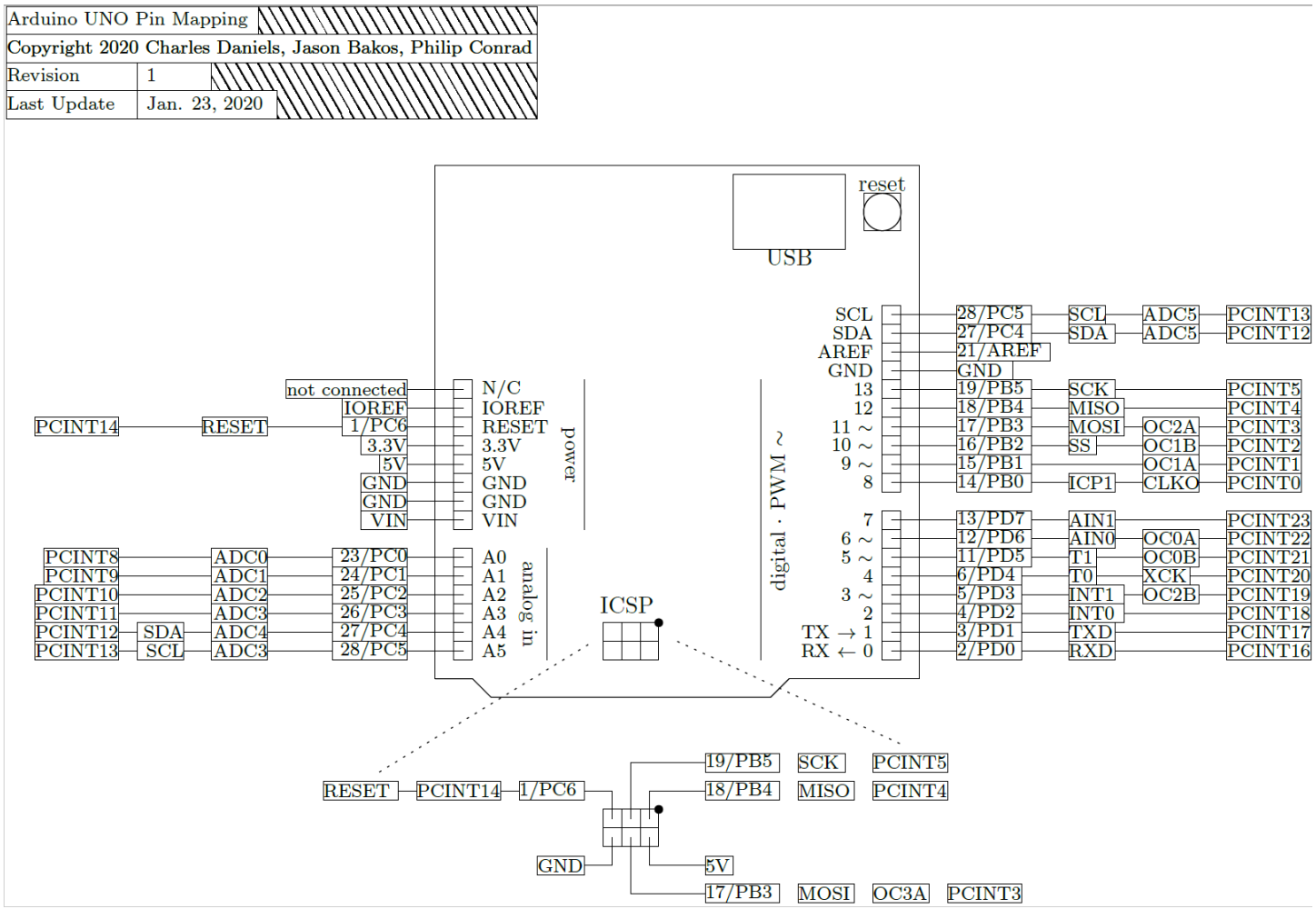




ATmega328 SPI Interface

- When configured as a Master
  - SS line must be handled by user software before communication
  - Writing a byte to the SPI Data Register starts the SPI clock generator, and the hardware shifts the eight bits into the Slave
  - After shifting one byte, the SPI clock generator stops, setting the end of Transmission Flag (SPIF)
  - If the SPI Interrupt Enable bit (SPIE) in the SPCR Register is set, an interrupt is requested
  - The Master may continue to shift the next byte by writing it into SPDR, or signal the end of packet by pulling high the Slave Select, SS line
  - The last incoming byte will be kept in the Buffer Register for later use

Pin Mapping


Arduino UNO Pin Mapping
Copyright 2020 Charles Daniels, Jason Bakos, Philip Conrad
Revision 1
Last Update Jan. 23, 2020
USB
reset
ICSP
power
analog in
digital · PWM ~
not connected N/C
IOREF IOREF
PCINT14 RESET 1/PC6 RESET
3.3V 3.3V
5V 5V
GND GND
GND GND
VIN VIN
PCINT8 ADC0 23/PC0 A0
PCINT9 ADC1 24/PC1 A1
PCINT10 ADC2 25/PC2 A2
PCINT11 ADC3 26/PC3 A3
PCINT12 SDA ADC4 27/PC4 A4
PCINT13 SCL ADC3 28/PC5 A5
SCL 28/PC5 SCL ADC5 PCINT13
SDA 27/PC4 SDA ADC5 PCINT12
AREF 21/AREF
GND GND
13 19/PB5 SCK PCINT5
12 18/PB4 MISO PCINT4
11 ~ 17/PB3 MOSI OC2A PCINT3
10 ~ 16/PB2 SS OC1B PCINT2
9 ~ 15/PB1 OC1A PCINT1
8 14/PB0 ICP1 CLKO PCINT0
7 13/PD7 AIN1 PCINT23
6 ~ 12/PD6 AIN0 OC0A PCINT22
5 ~ 11/PD5 T1 OC0B PCINT21
4 6/PD4 T0 XCK PCINT20
3 ~ 5/PD3 INT1 OC2B PCINT19
2 4/PD2 INT0 PCINT18
TX → 1 3/PD1 TXD PCINT17
RX ← 0 2/PD0 RXD PCINT16
RESET PCINT14 1/PC6
19/PB5 SCK PCINT5
18/PB4 MISO PCINT4
GND
5V
17/PB3 MOSI OC3A PCINT3

ATmega328 SPI Interface

- When configured as a Slave
  - SPI interface will remain sleeping with MISO tri-stated as long as the SS pin is driven high
  - Software may update the contents of the SPI Data Register, SPDR, but the data will not be shifted out by incoming clock pulses on the SCK pin until the SS pin is driven low
  - As one byte has been completely shifted, the end of Transmission Flag, SPIF is set
  - If the SPI Interrupt Enable bit, SPIE, in the SPCR Register is set, an interrupt is requested
  - The Slave may continue to place new data to be sent into SPDR before reading the incoming data
  - The last incoming byte will be kept in the Buffer Register for later use

- The system is single buffered in the transmit direction and double buffered in the receive direction
  - Bytes to be transmitted cannot be written to the SPI Data Register before the entire shift cycle is completed
  - When receiving data, a received character must be read from the SPI Data Register before the next character has been completely shifted in, otherwise, the first byte is lost

ATmega328 SPI Interface

- To ensure correct sampling of the clock signal, the minimum low and high periods should be:
  - Low periods: Longer than 2 CPU clock cycles
  - High periods: Longer than 2 CPU clock cycles
  - This gives a clock speed of 4 times slower than CPU clock
    - 4 Mbps

SPI Control Register

Bit	7	6	5	4	3	2	1	0	
0x2C (0x4C)	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- SPIE: SPI interrupt enable (vector: SPI_STC_vect)
- SPE: SPI enable (important!)
- DORD: data order (1=LSB first, 0=MSB first)
- MSTR: master/slave select
- CPOL: clock polarity
- CPHA: clock phase
- SPR[1:0]: speed

SPI2X	SPR1	SPR0	SCK Frequency
0	0	0	$f_{osc}/4$
0	0	1	$f_{osc}/16$
0	1	0	$f_{osc}/64$
0	1	1	$f_{osc}/128$
1	0	0	$f_{osc}/2$
1	0	1	$f_{osc}/8$
1	1	0	$f_{osc}/32$
1	1	1	$f_{osc}/64$

SPI Status Register

Bit	7	6	5	4	3	2	1	0	
0x2D (0x4D)	**SPIF**	**WCOL**	**–**	**–**	**–**	**–**	**–**	**SPI2X**	**SPSR**
Read/Write	R	R	R	R	R	R	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- SPIF: SPI interrupt flag
- WCOL: write collision flag
- SPI2X: double SPI speed

SPI Data Register

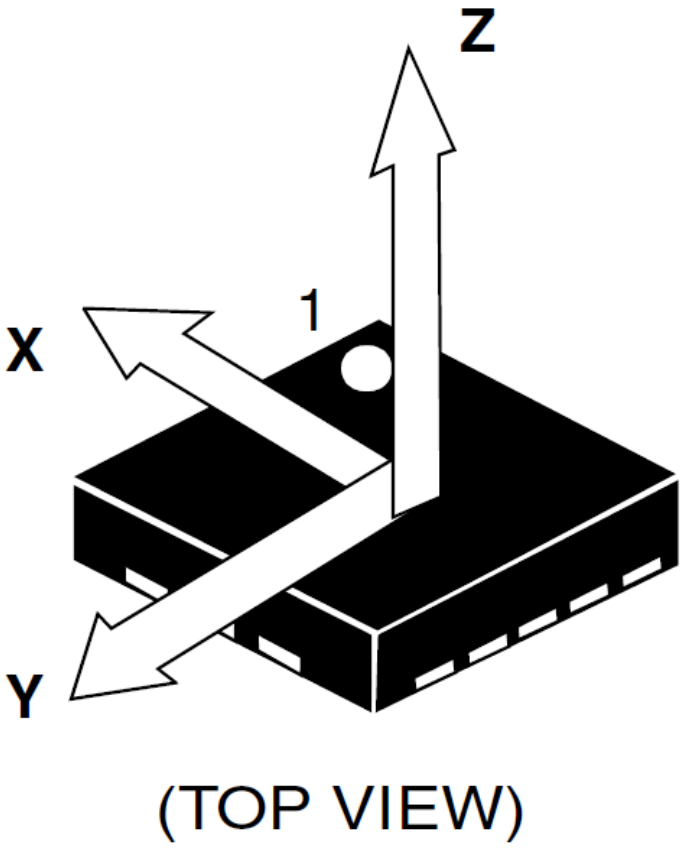
Bit	7	6	5	4	3	2	1	0	
0x2E (0x4E)	**MSB**							**LSB**	**SPDR**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	X	X	X	X	X	X	X	X	Undefined

- Writing to the register initiates transmission
- Reading the register causes the shift register receive buffer to be read

LIS3DH 3D Accelerometer

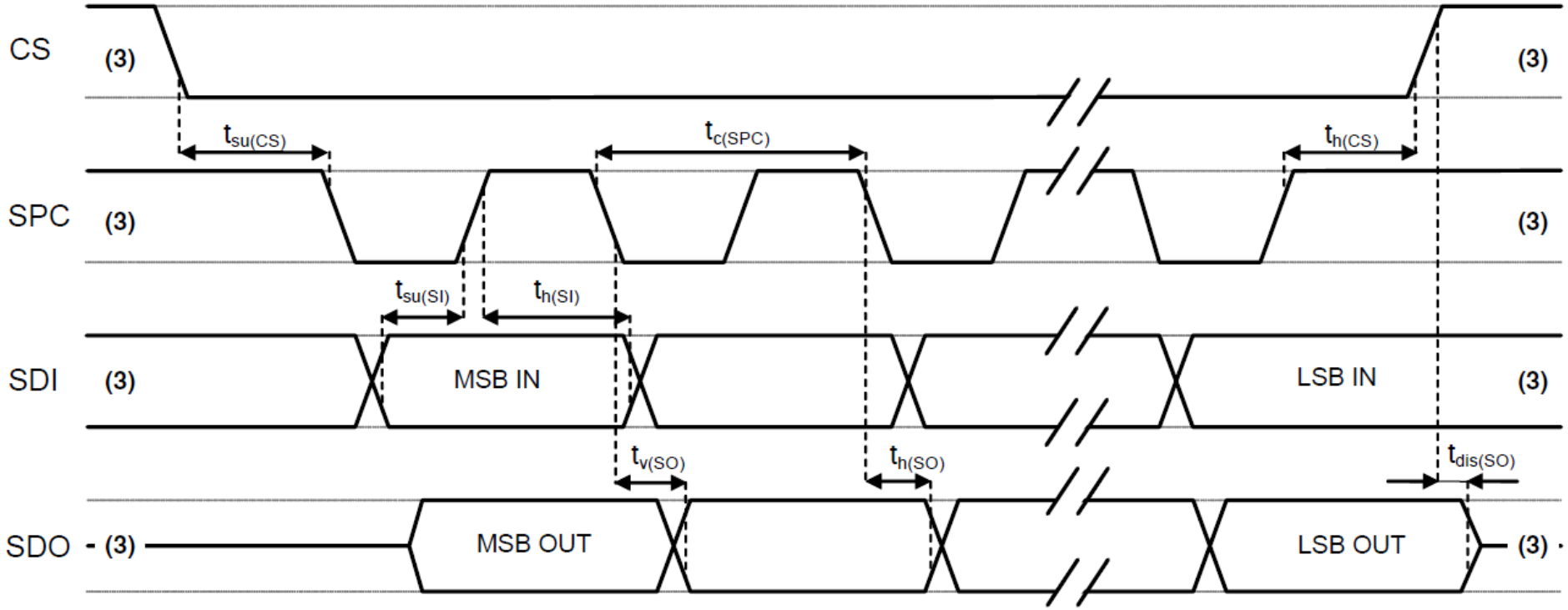
- 4 range modes:
  - +/-2g, +/-4g, +/-8g, +/-16g
- Sampling rates: 1 Hz to 5 KHz
- SPI clock rate <= 10 MHz

- Objective:
  - Read acceleration data at native sample rate
  - Minimize load on ATmega328p
    - Use interrupts

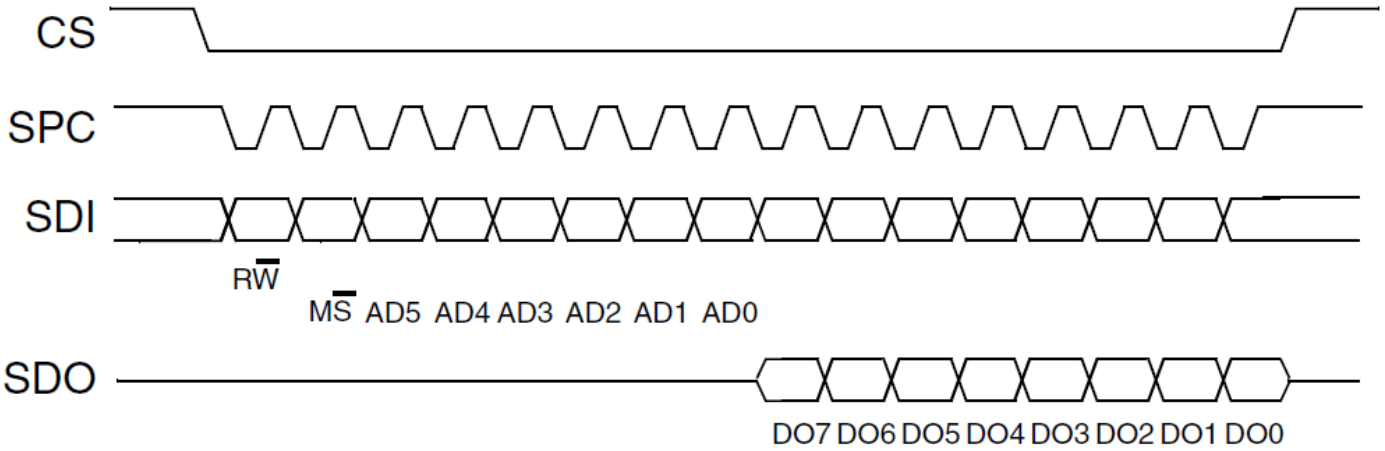


(TOP VIEW)

SPI Timing Diagram



SPI Read



bit 0: READ bit. The value is 1.

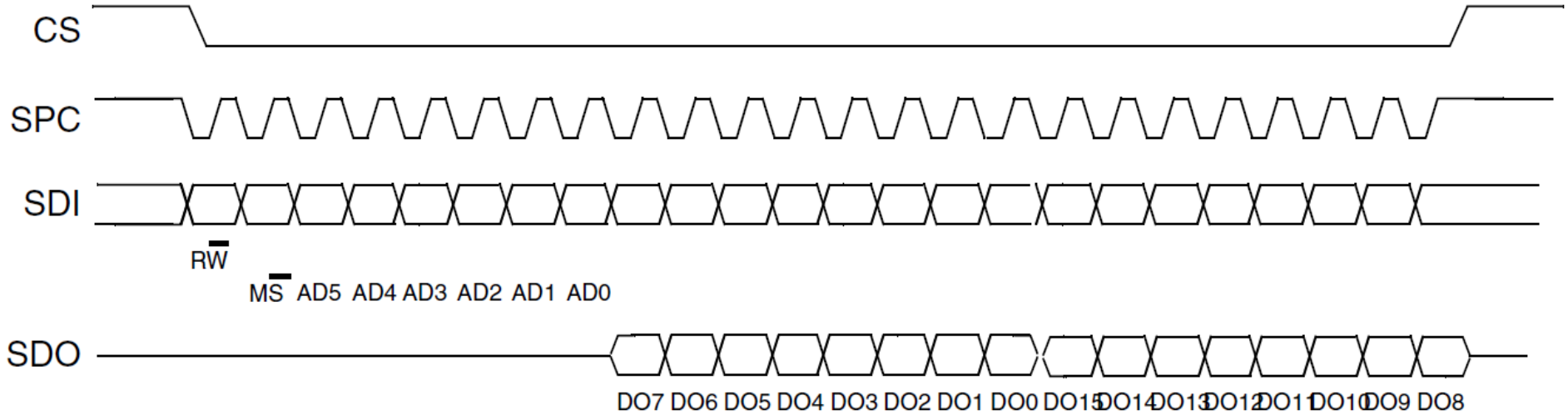
bit 1: $\overline{MS}$ bit. When 0 do not increment address, when 1 increment address in multiple reading.

bit 2-7: address AD(5:0). This is the address field of the indexed register.

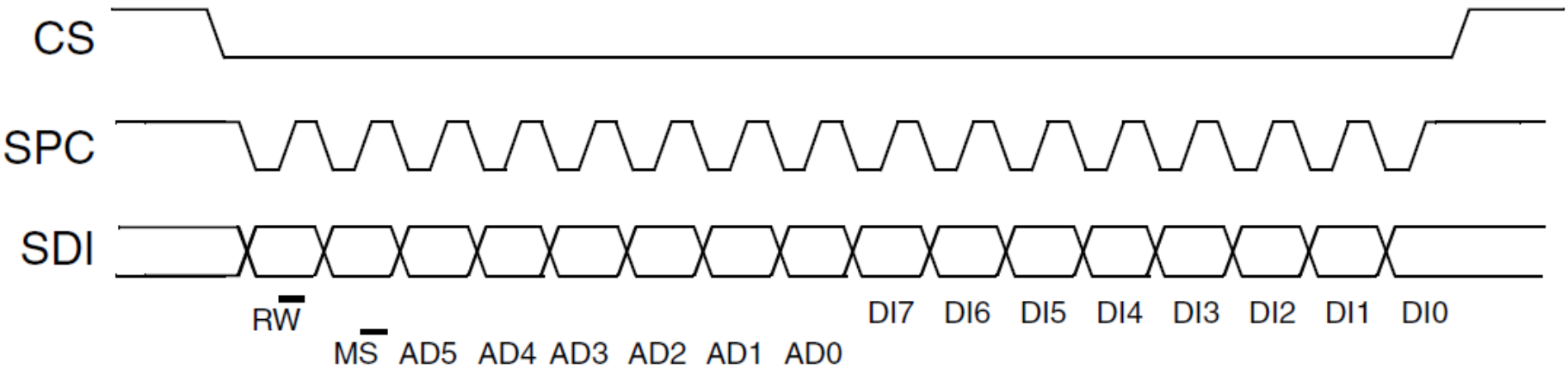
bit 8-15: data DO(7:0) (read mode). This is the data that is read from the device (MSb first).

bit 16-... : data DO(...-8). Further data in multiple byte reading.

SPI Multiple Read



SPI Write



bit 0: WRITE bit. The value is 0.

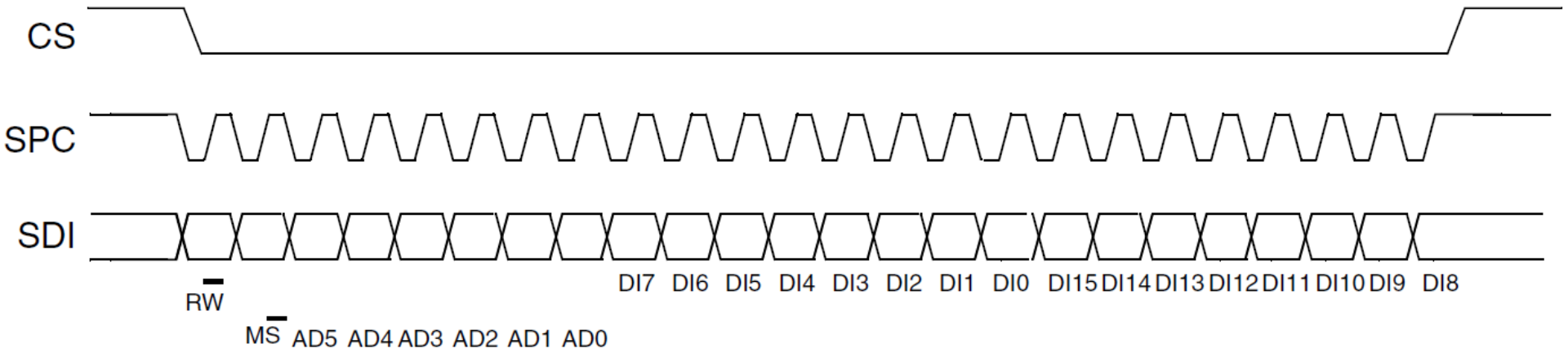
bit 1: $\overline{\text{MS}}$ bit. When 0 do not increment address, when 1 increment address in multiple writing.

bit 2 -7: address AD(5:0). This is the address field of the indexed register.

bit 8-15: data DI(7:0) (write mode). This is the data that is written inside the device (MSb first).

bit 16-... : data DI(...-8). Further data in multiple byte writing.

SPI Multiple Write



Relevant Registers

- Acceleration data:
  - XH and XL: 0x29 and 0x28
  - YH and YL: 0x2B and 0x2A
  - ZH and ZL: 0x2D and 0x2C

- WHO_AM_I: 0x0F (always reads as 0x33)

Relevant Registers

- CTRL_REG1: 0x20

Table 24. CTRL_REG1 register

ODR3	ODR2	ODR1	ODR0	LPen	Zen	Yen	Xen

Table 25. CTRL_REG1 description

ODR3-0	Data rate selection. Default value: 00 (0000:50 Hz; Others: Refer to *Table 25*, "Data rate configuration")
LPen	Low power mode enable. Default value: 0 (0: normal mode, 1: low power mode)
Zen	Z axis enable. Default value: 1 (0: Z axis disabled; 1: Z axis enabled)
Yen	Y axis enable. Default value: 1 (0: Y axis disabled; 1: Y axis enabled)
Xen	X axis enable. Default value: 1 (0: X axis disabled; 1: X axis enabled)

Table 26. Data rate configuration

ODR3	ODR2	ODR1	ODR0	Power mode selection
0	0	0	0	Power down mode
0	0	0	1	Normal / low power mode (1 Hz)
0	0	1	0	Normal / low power mode (10 Hz)
0	0	1	1	Normal / low power mode (25 Hz)
0	1	0	0	Normal / low power mode (50 Hz)
0	1	0	1	Normal / low power mode (100 Hz)
0	1	1	0	Normal / low power mode (200 Hz)
0	1	1	1	Normal / low power mode (400 Hz)
1	0	0	0	Low power mode (1.6 KHz)
1	0	0	1	Normal (1.25 kHz) / low power mode (5 KHz)

Relevant Registers

- CTRL_REG3: 0x22

Table 30. CTRL_REG3 register

I1_CLICK	I1_AOI1	I1_AOI2	I1_DRDY1	I1_DRDY2	I1_WTM	I1_OVERRUN	--

Table 31. CTRL_REG3 description

I1_CLICK	CLICK interrupt on INT1. Default value 0. (0: Disable; 1: Enable)
I1_AOI1	AOI1 interrupt on INT1. Default value 0. (0: Disable; 1: Enable)
I1_AOI2	AOI2 interrupt on INT1. Default value 0. (0: Disable; 1: Enable)
I1_DRDY1	DRDY1 interrupt on INT1. Default value 0. (0: Disable; 1: Enable)
I1_DRDY2	DRDY2 interrupt on INT1. Default value 0. (0: Disable; 1: Enable)
I1_WTM	FIFO Watermark interrupt on INT1. Default value 0. (0: Disable; 1: Enable)
I1_OVERRUN	FIFO Overrun interrupt on INT1. Default value 0. (0: Disable; 1: Enable)

Relevant Registers

- CTRL_REG5: 0x24

Table 35. CTRL_REG5 register

BOOT	FIFO_EN	--	--	LIR_INT1	D4D_INT1	0	0

Table 36. CTRL_REG5 description

BOOT	Reboot memory content. Default value: 0 (0: normal mode; 1: reboot memory content)
FIFO_EN	FIFO enable. Default value: 0 (0: FIFO disable; 1: FIFO Enable)
LIR_INT1	Latch interrupt request on INT1_SRC register, with INT1_SRC register cleared by reading INT1_SRC itself. Default value: 0. (0: interrupt request not latched; 1: interrupt request latched)
D4D_INT1	4D enable: 4D detection is enabled on INT1 when 6D bit on INT1_CFG is set to 1.

Relevant Registers

- CTRL_REG6: 0x25

Table 37. CTRL_REG6 register

I2_CLICKen	I2_INT1	0	BOOT_I1	0	- -	H_LACTIVE	-

- H_LACTIVE: set INT1 to be active low

Relevant Registers

- INT1_SRC: 0x31

Table 49. INT1_SRC register

0	IA	ZH	ZL	YH	YL	XH	XL

Table 50. INT1_SRC description

IA	Interrupt active. Default value: 0 (0: no interrupt has been generated; 1: one or more interrupts have been generated)
ZH	Z high. Default value: 0 (0: no interrupt, 1: Z High event has occurred)
ZL	Z low. Default value: 0 (0: no interrupt; 1: Z Low event has occurred)
YH	Y high. Default value: 0 (0: no interrupt, 1: Y High event has occurred)
YL	Y low. Default value: 0 (0: no interrupt, 1: Y Low event has occurred)
XH	X high. Default value: 0 (0: no interrupt, 1: X High event has occurred)
XL	X low. Default value: 0 (0: no interrupt, 1: X Low event has occurred)

Pulse Width Modulation

- Way to implement an analog value with a digital pin
- Regular periodic signal whose duty cycle determines average voltage over time