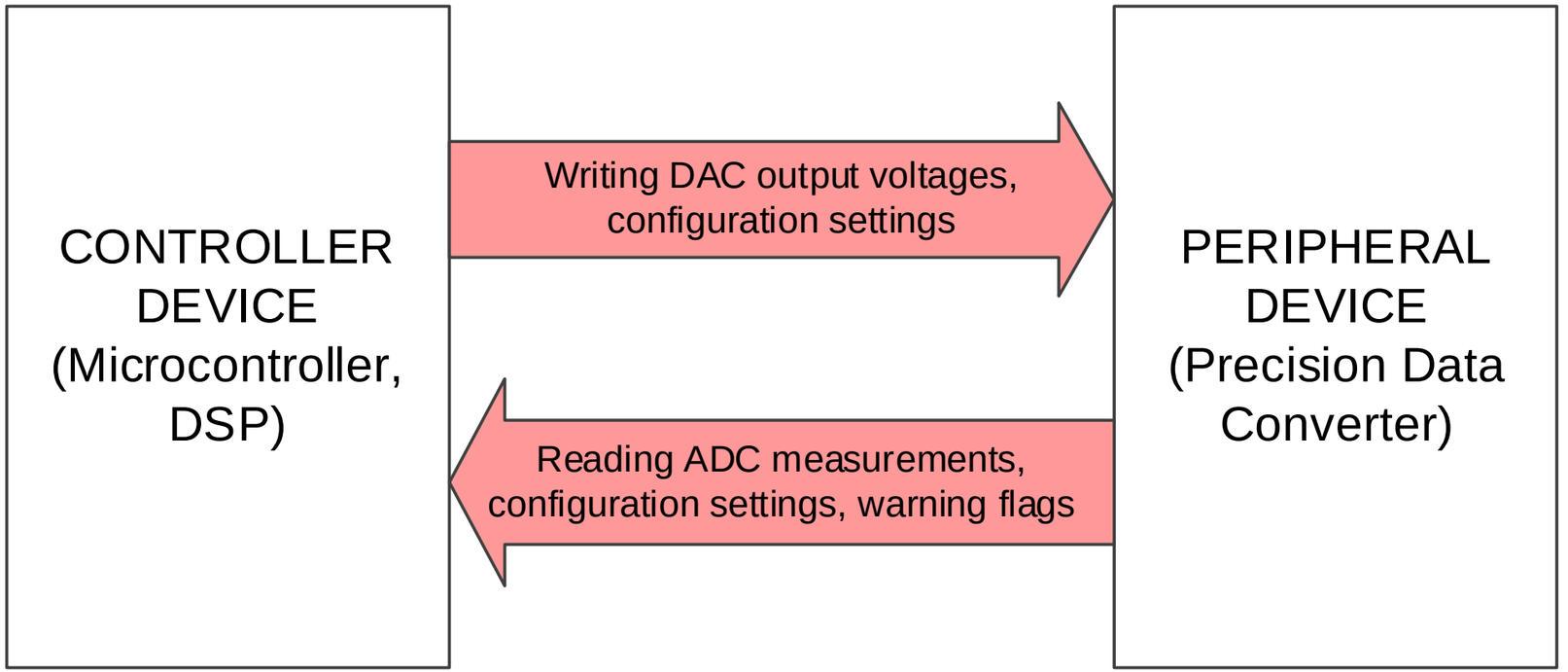
Basics of SPI: Serial Communications

TIPL 6001
TI Precision Labs – Digital Communications

Presented by Alex Smith

Prepared by Joseph Wu

Common Communication Standard



Communication Through Bits

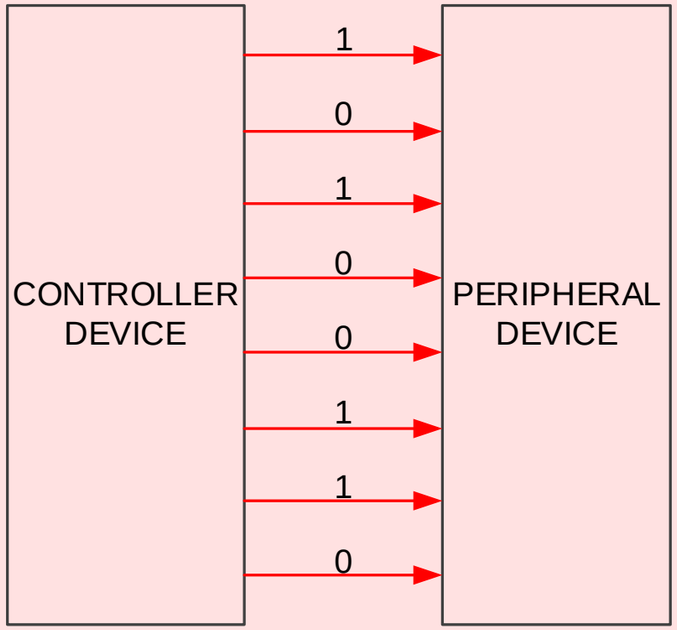
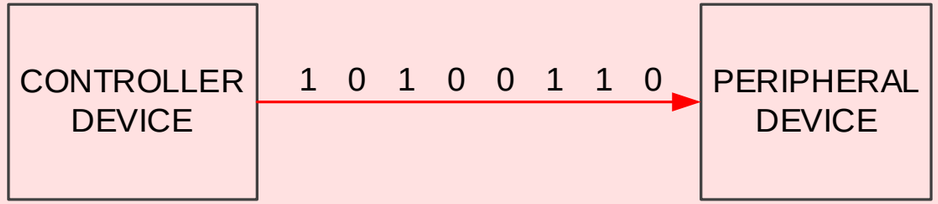
Communications are transmitted in binary, constructed from bits, and also be organized in hexadecimal

Eight bits is one byte → 1 0 1 0 0 1 1 0

Every four bits can be represented as hexadecimal → A 6

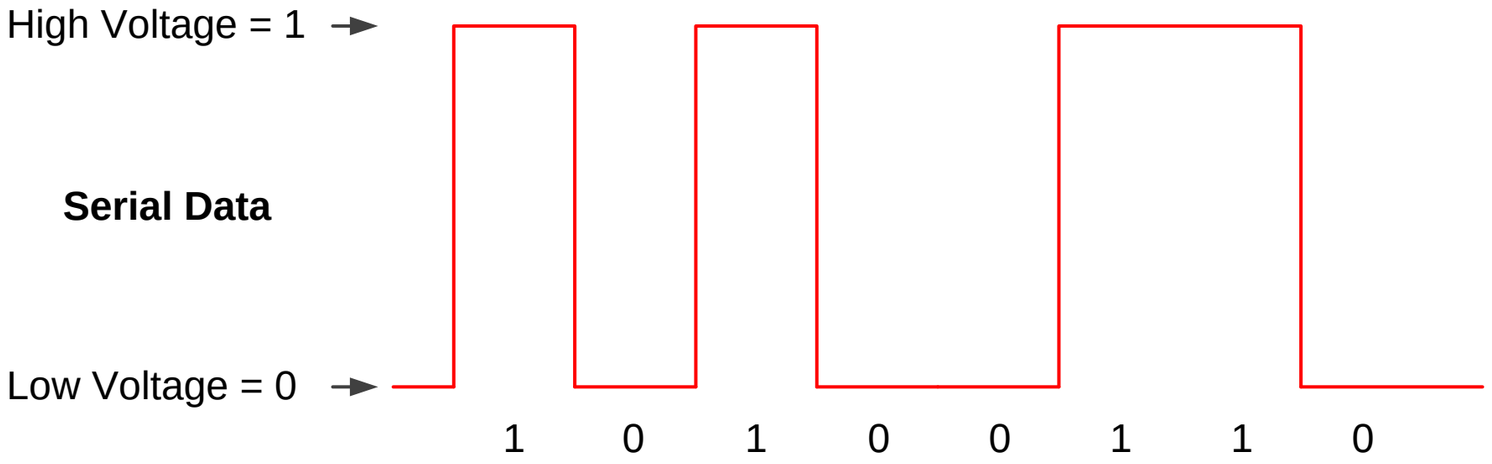
Bits	Hexadecimal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

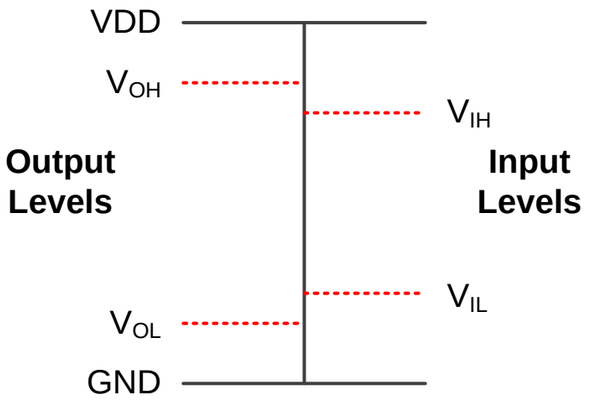
Parallel vs Serial

Parallel	Serial
	
Data is sent eight bits or one byte at a time, but it requires eight data lines	Data is sent one bit at a time and only requires one line, but it may be slower

Voltage Levels

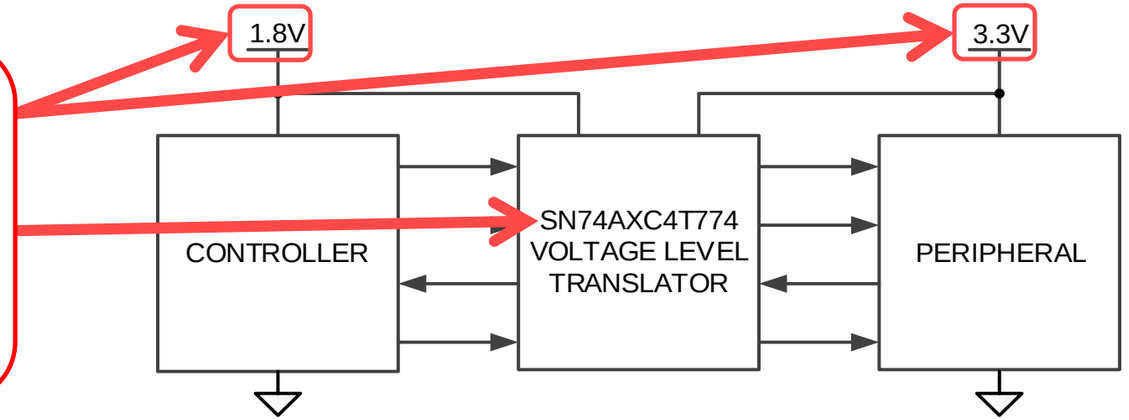
Datasheets will define the output driving range for digital outputs…





…And define the input range for high and low for digital inputs

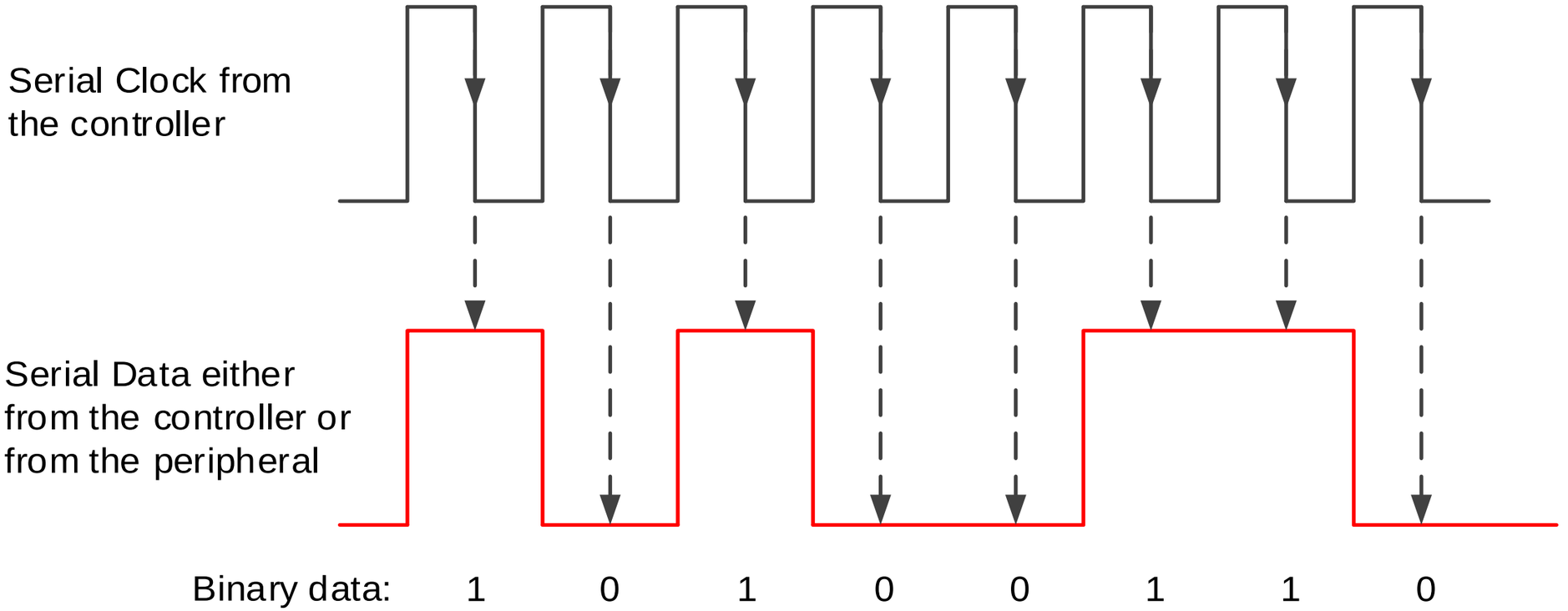
If the controller and peripheral operate on different supply voltages, a **voltage level translator** may be required



Common Timing

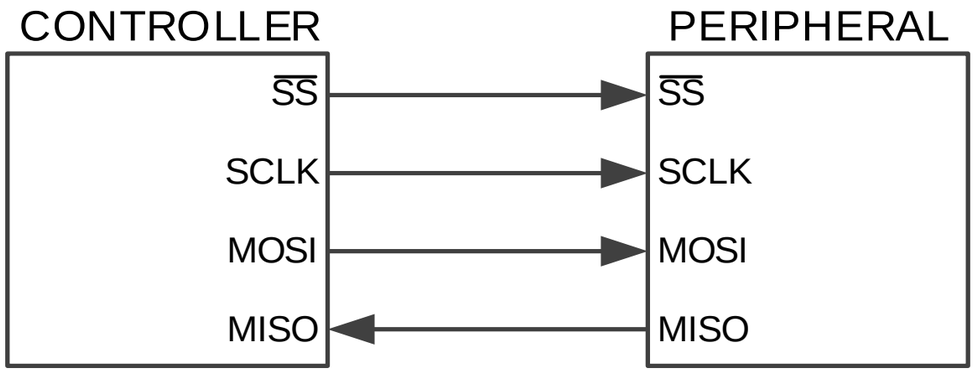
A serial clock is used to determine a common timing:

In this example, the data is read on the falling edge of the serial clock



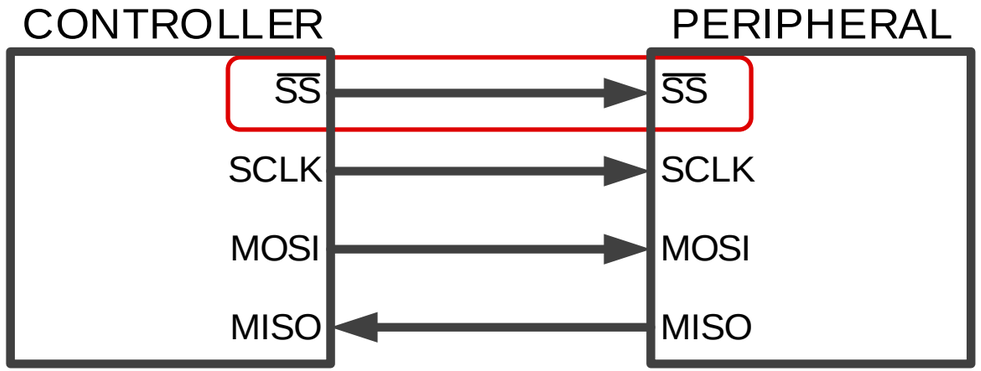
Serial Peripheral Interface (SPI)

SPI
Controller controls the peripheral select and the serial clock
An SPI bus can have only one controller, but may control multiple slaves
Each peripheral has a peripheral select for independent control
Data can be transmitted from controller to peripheral or peripheral to controller that may be used as full duplex



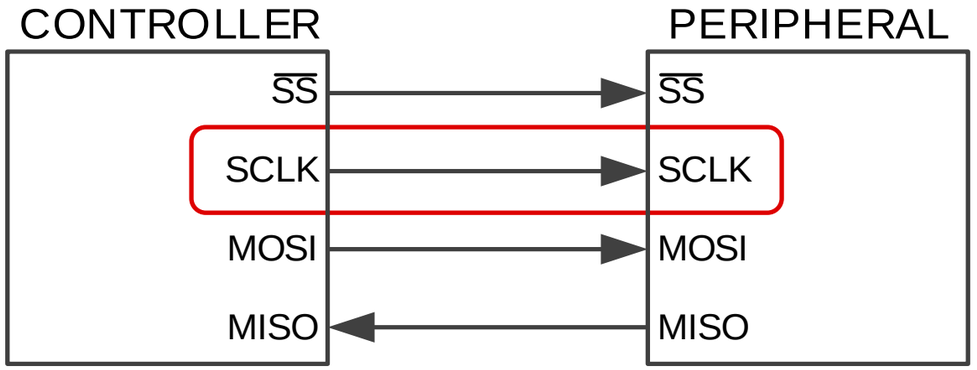
SPI connections: SS

SS
Peripheral select: Selects the peripheral device for communication
Is often used as active low, which is often represented by an overbar
Also known as: $\overline{SS}$, SSEL, $\overline{CS}$, CS, $\overline{SYNC}$, nSS, SS#
Commonly labeled as $\overline{CS}$ or $\overline{SYNC}$ in TI data converters



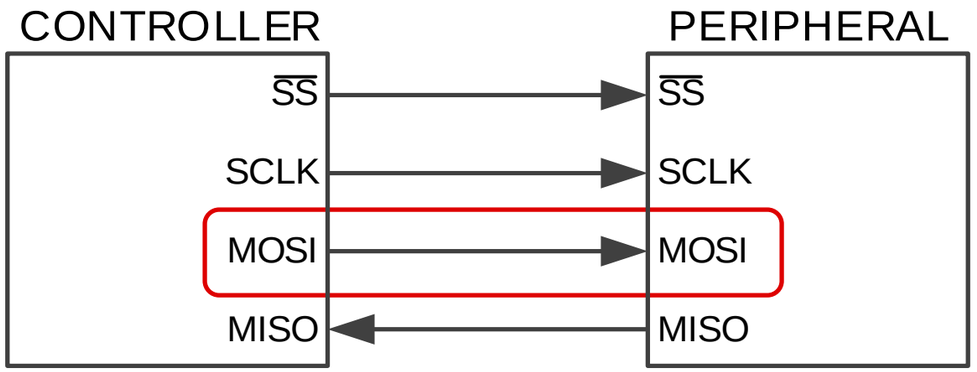
SPI connections: SCLK

SCLK
Serial Clock: Synchronizes data transmission between the controller and peripheral
SCLK originates from controller and shared with all slaves
Clock may idle high or low
Data is clocked in on either the rising or falling edge of the clock
Also known as: SCK



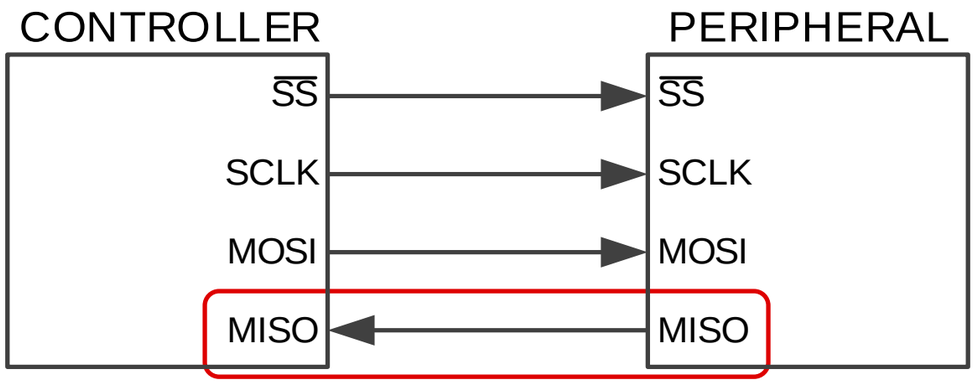
SPI connections: MOSI

MOSI
Controller Out, Peripheral In: Output from the Controller used to send data to the peripheral device
Can be shared between peripheral devices
Also known as: SIMO, MSTR; from the peripheral device: SDI, DI, DIN, SI; from the Controller device: SDO, DO, DOUT, SO
Commonly labeled as DIN in TI data converters

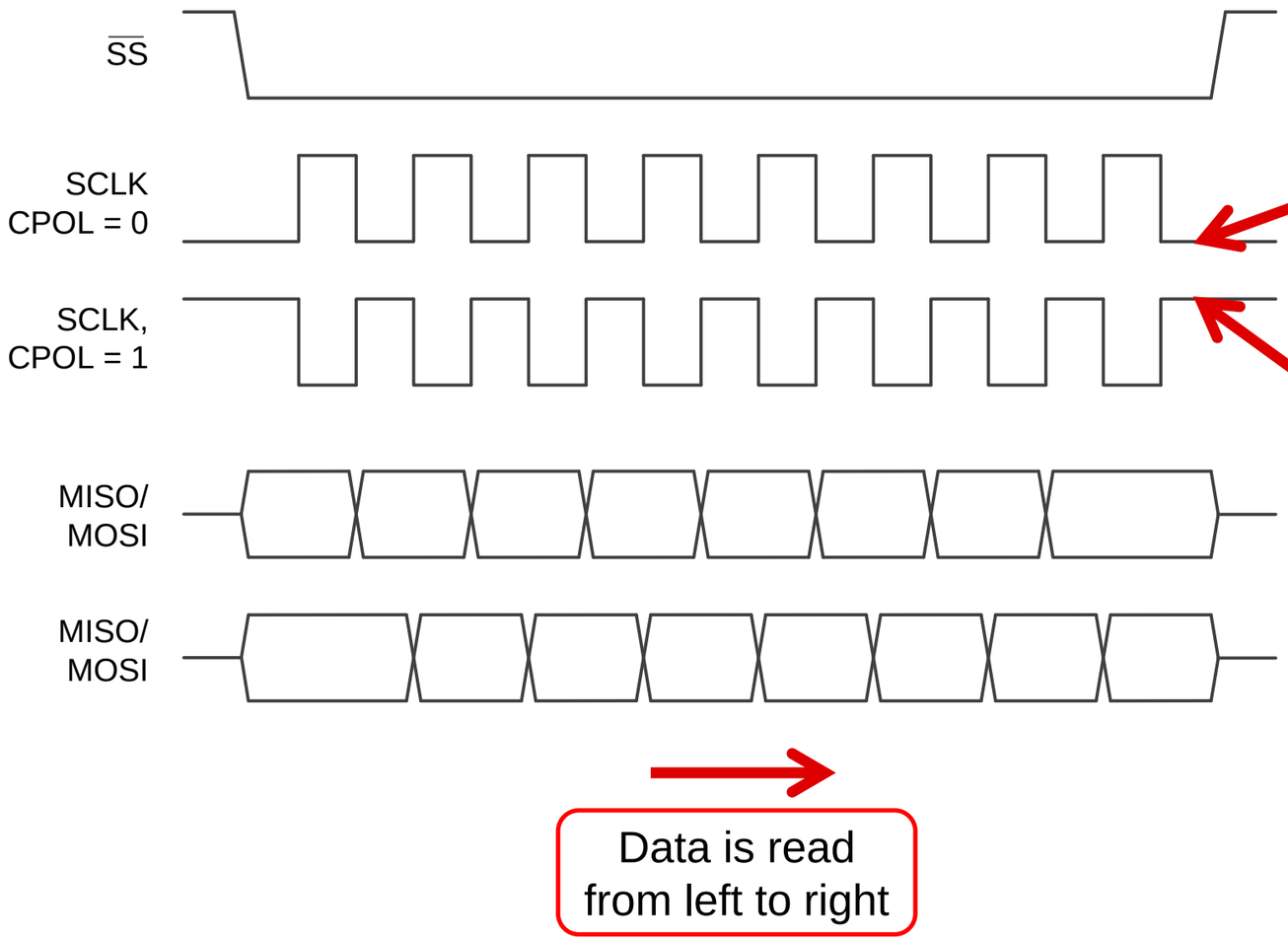


SPI connections: MISO

MISO
Controller In, Peripheral Out: Output from the peripheral device used to send data to the Controller
Can be shared between all peripheral devices,
Peripheral output becomes high impedance when SS is not selected
Also known as: SOMI; from the peripheral device: SDO, DO, DOUT, SO; to the controller device: SDI, DI, DIN, SI
Commonly labeled as DOUT in TI data converters



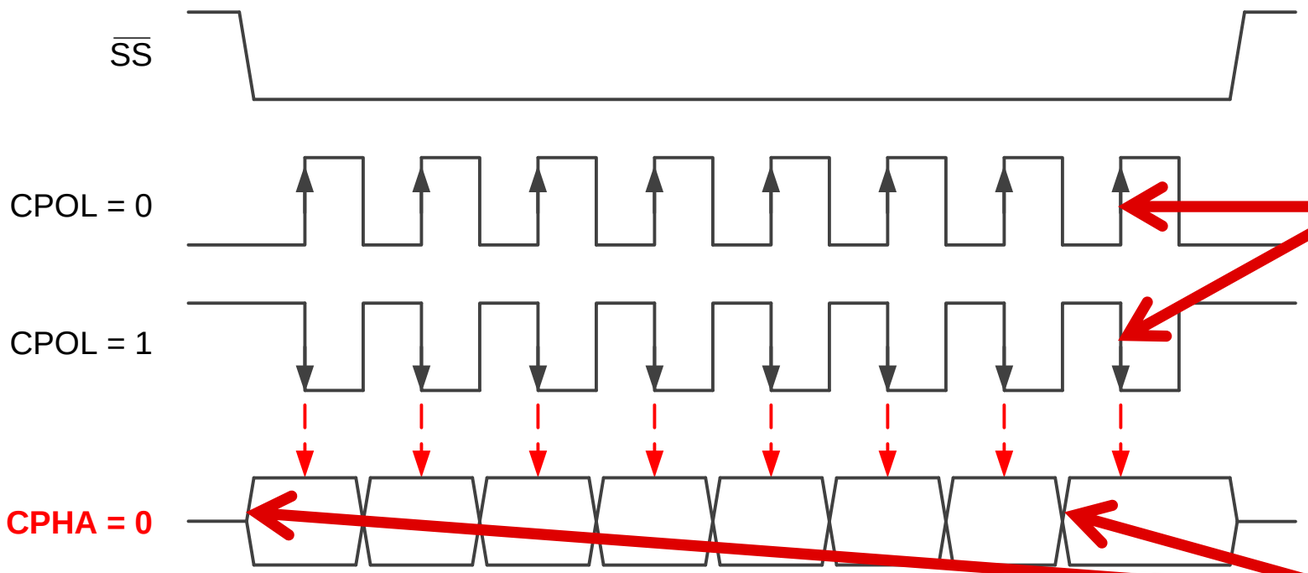
SPI Clock Polarity: CPOL



CPOL
Clock Polarity
If SCLK idles low, CPOL = 0 Leading edge is rising edge, trailing edge is falling edge
If SCLK idles high, CPOL = 1 Leading edge is falling edge, trailing edge is rising edge

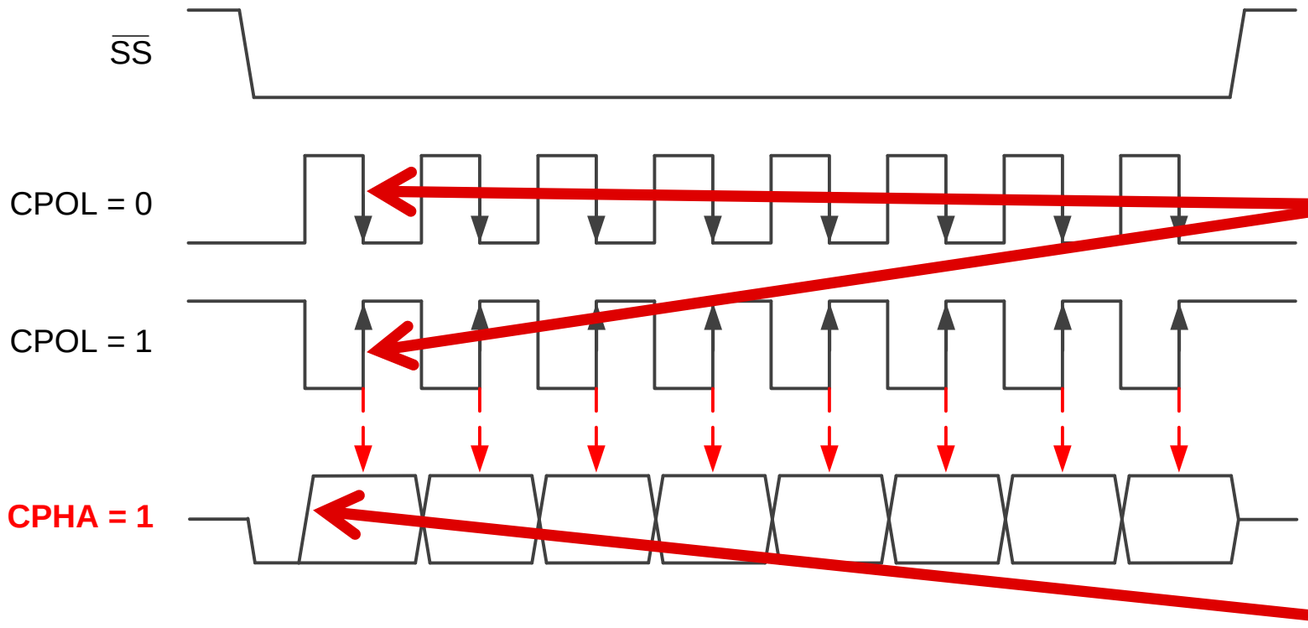
For each clock pulse, the leading edge is the first edge of the pulse, the trailing edge is the second edge of the pulse

SPI Clock Phase, CPHA = 0



CPHA = 0
Clock Phase
CPHA = 0: Data is clocked on the leading edge
For CPOL = 0, this is the rising edge, For CPOL = 1, this is the falling edge
Data is set up when SS falls low, or at the previous trailing edge of SCLK

SPI Clock Phase, CPHA = 1



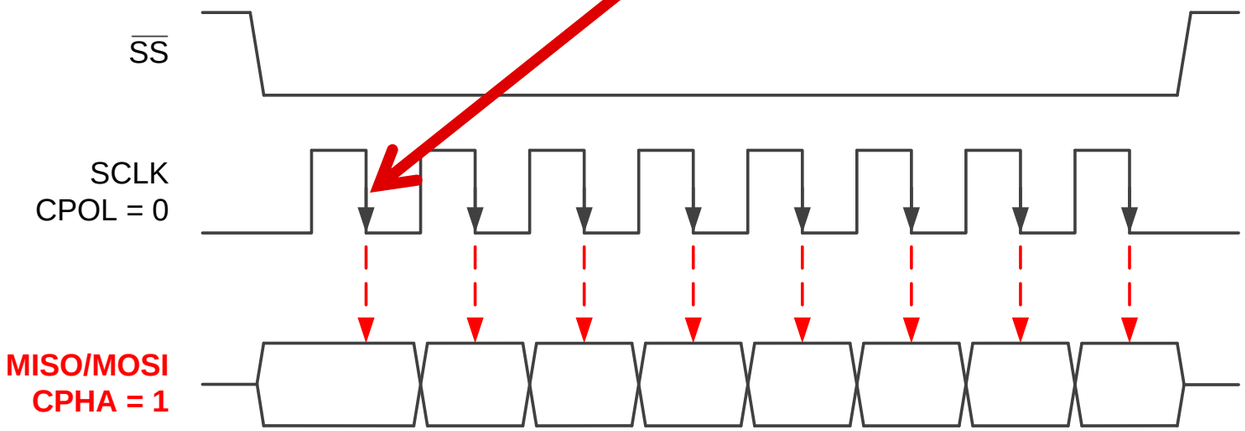
CPHA = 1
Clock Phase
CPHA = 1: Data is clocked on the trailing edge
For SCLK CPOL = 0, this is the falling edge, For SCLK CPOL = 1, this is the rising edge
Data is set up at the previous leading edge of SCLK

SPI Mode Numbers

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

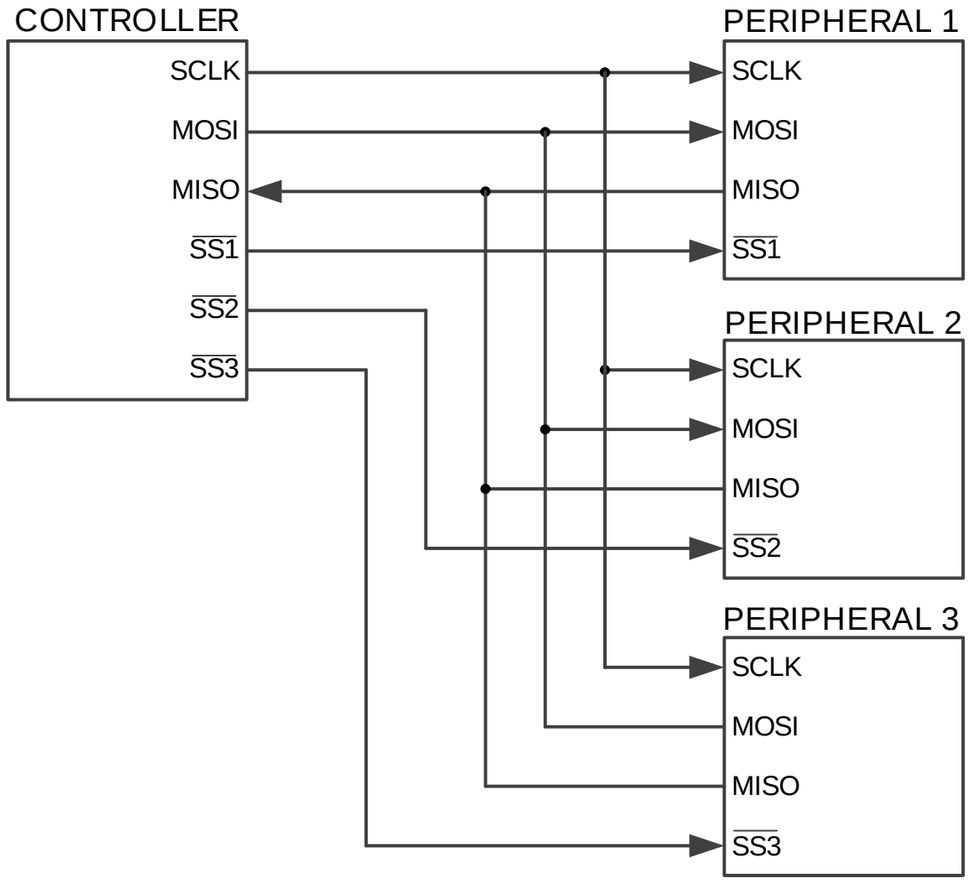
Example:

SPI Mode 1 – SCLK idles low, data is read on the trailing edge of the clock



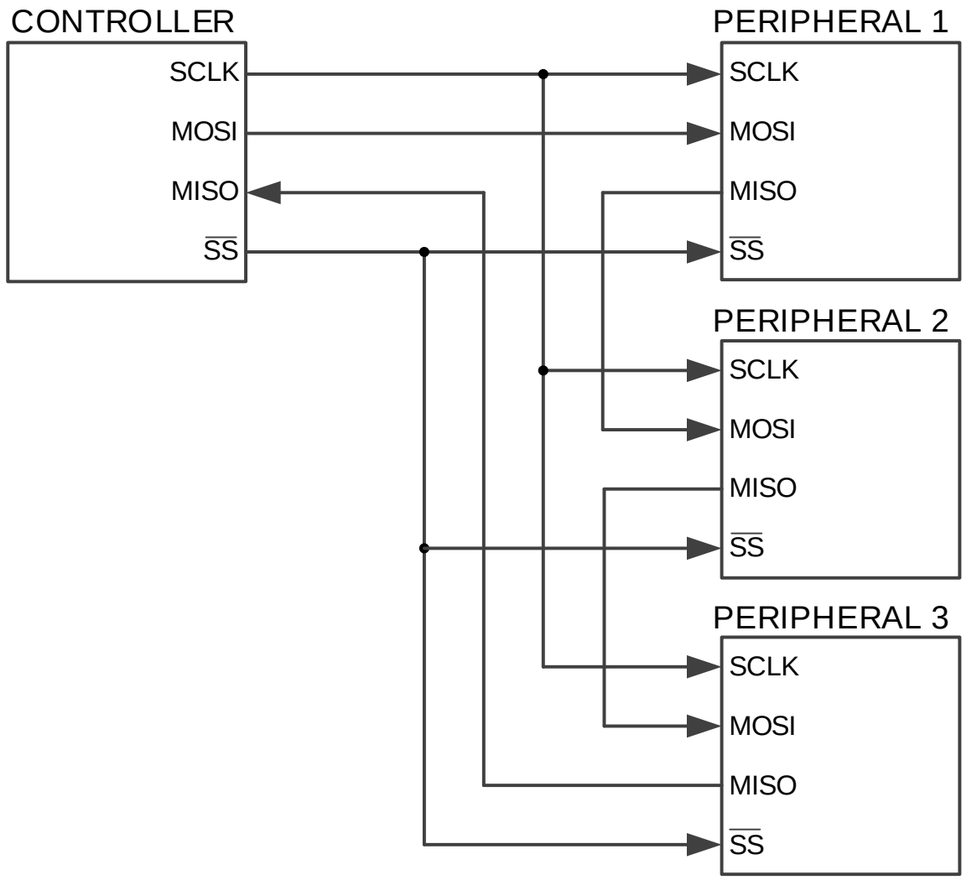
Controlling Multiple Peripherals: Multiple Peripheral Selects

Controlling Multiple Peripherals Method 1 – Multiple SS
Each device has an independent peripheral select
SCLK, MISO, MOSI are each shared between devices
If device is not selected MISO is becomes high impedance

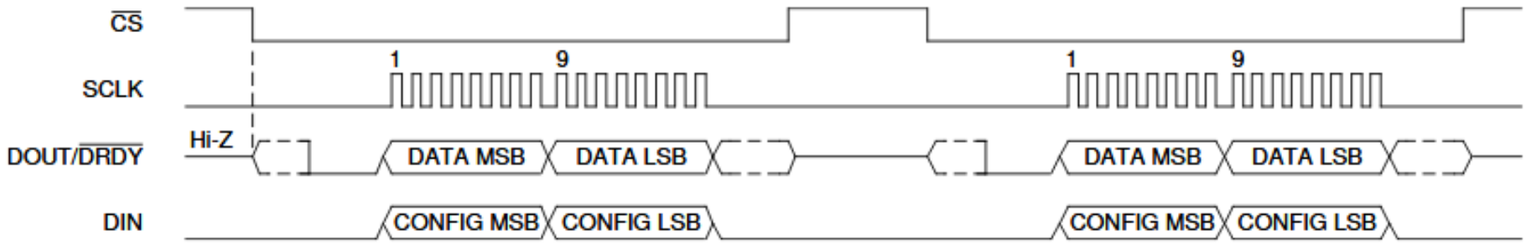


Controlling Multiple Peripherals: Daisy Chaining

Controlling Multiple Peripherals Method 2 – Daisy Chain
Single SS controls all peripheral devices
Data is sent from one device to the nex as in a chain; MISO from one device connects to the MOSI of the next device
Not all devices support daisy chaining of communications



SPI Communication Example: ADS1118



BIT	FIELD	SETTING	DESCRIPTION
15	OS	1	Start conversion
14:12	MUX[2:0]	000	Differential input measurement, AINP = AIN0, AINN = AIN1, selection of the first channel
11:9	PGA[2:0]	000	FSR = ±6.144V, sets the ADC to be able to measure the full supply range of 0V to VDD
9	MODE	1	Operation in single-shot conversion mode
7:5	DR[2:0]	100	Data rate = 128SPS
4	TS_MODE	0	Measures analog inputs instead of the temperature sensor
3	PULL_UP_EN	0	Disables the internal pullup resistor on the DOUT/DRDY
2:1	NOP[1:0]	01	Write 01 to these bits to ensure that the device writes to the configuration register
0	RESERVED	1	Always write 1 to this bit

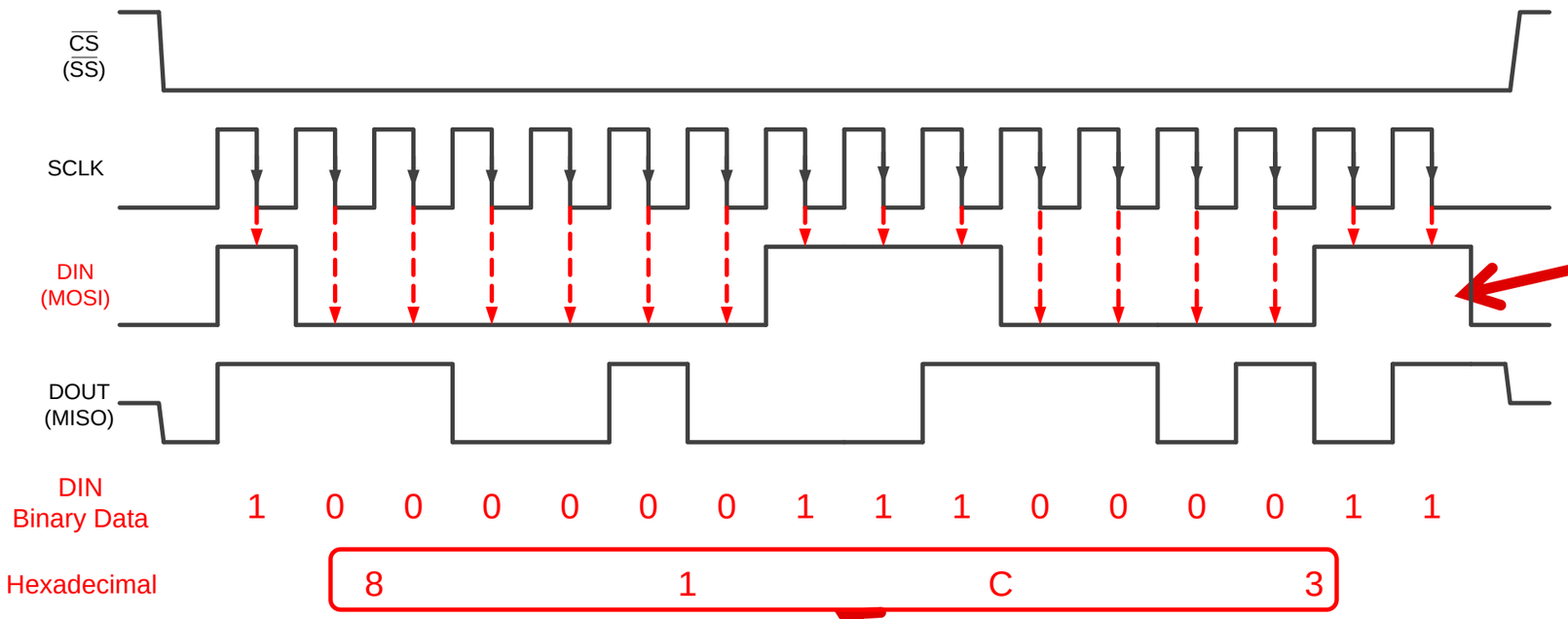
ADS1118

- 16-bit ADC
- Uses SPI Mode 1
- Data is clocked in and out of the device at the same time
- Configuration register is 16 bits
- Diagram shows two transfers of data

In this example, the data sent to the device shown in the Setting column

Here the configuration register is set to:
1000 0001 1100 0011 (81C3h)

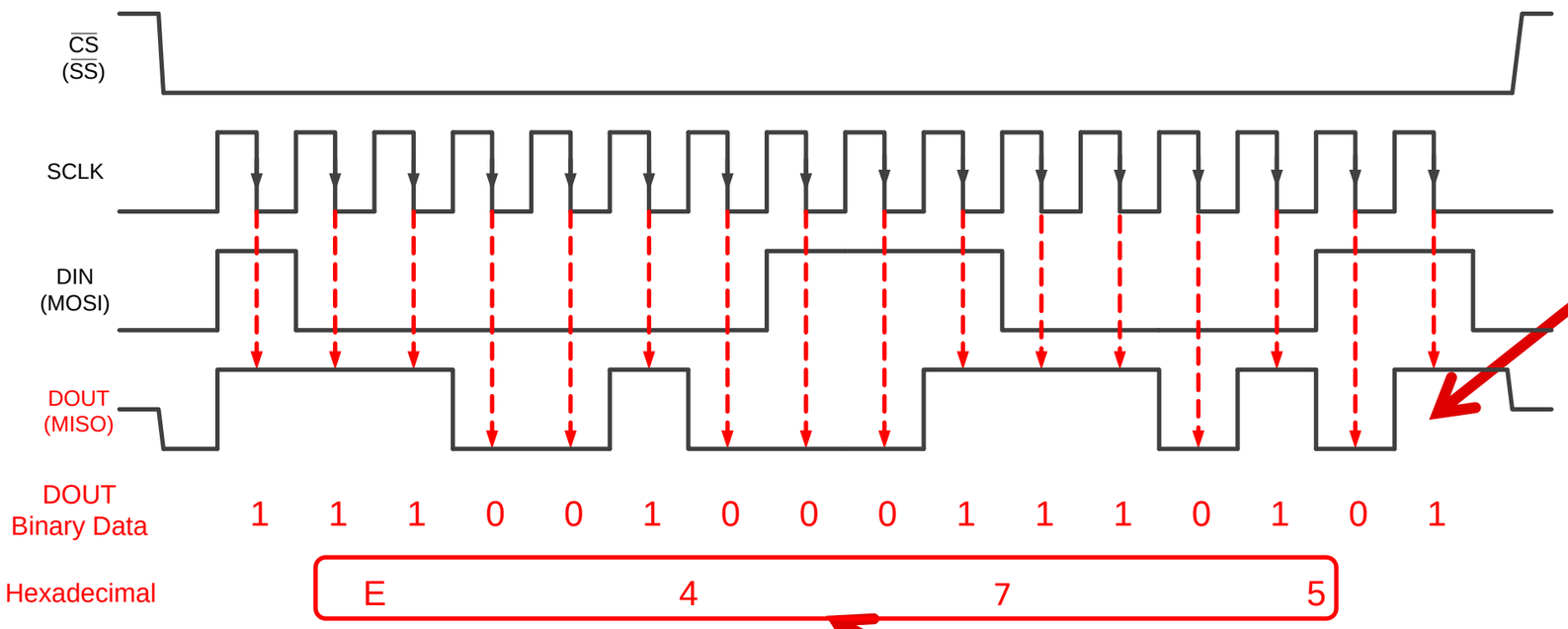
SPI Communication Example: ADS1118 Write



ADS1118 Example Write
CS selects the device
SCLK idles low and data is clocked on the falling edge
DIN sends 81C3h to the device
DOUT is clocked in with the same SCLK pulses

The configuration register is set to:
1000 0001 1100 0011 (81C3h)

SPI Communication Example: ADS1118 Read



ADS1118 Example Read
ADC data comes out on DOUT
DOUT is clocked out with the same SCLK pulses as DIN
The output data clocked out is E375h

The output data clocked out of the device is:
1110 0100 0111 0101 (E475h)

Thanks for your time!
Please try the quiz.

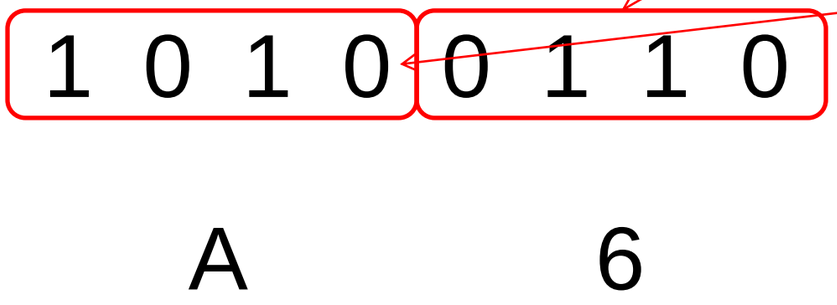
Quiz: Basics of SPI: Serial Communication

1. The following binary output is what value in hexadecimal?
   a. 59
   b. 6A
   c. 95
   d. A6

1 0 1 0 0 1 1 0

Quiz: Basics of SPI: Serial Communication

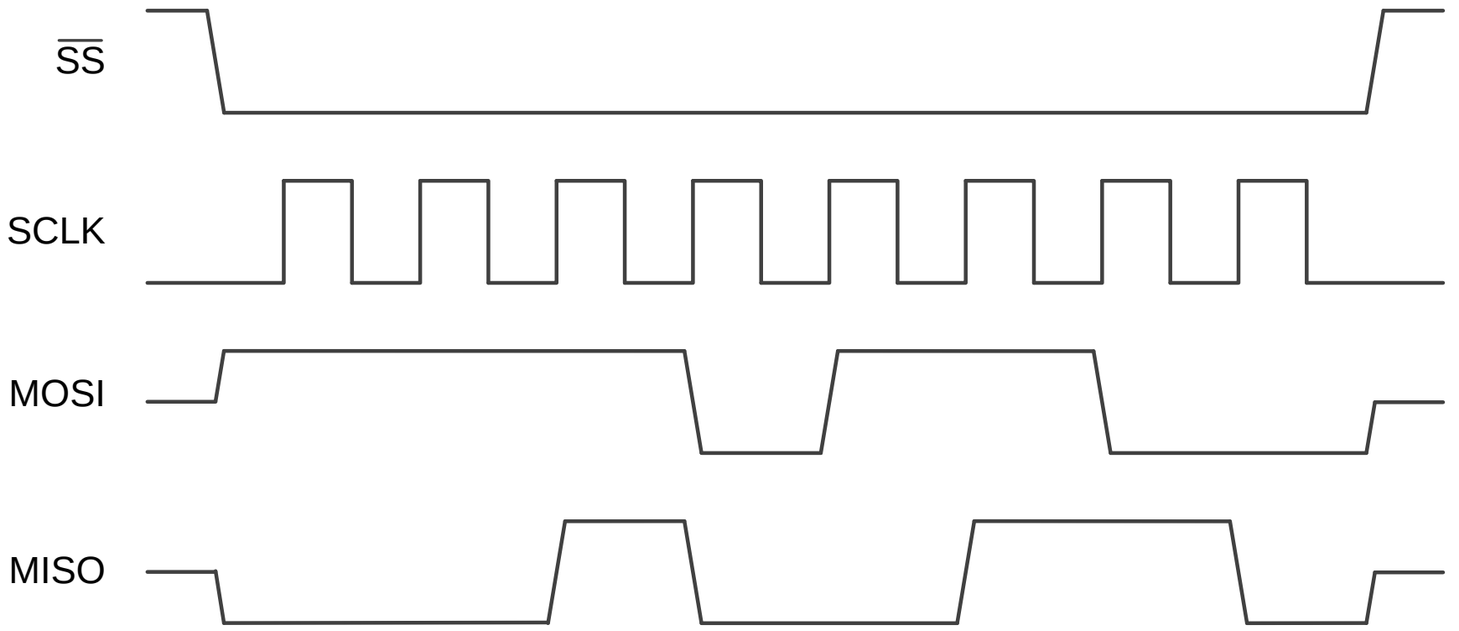
1. The following binary output is what value in hexadecimal?
   a. 59
   b. 6A
   c. 95
   d. A6



Bits	Hexadecimal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Quiz: Basics of SPI: Serial Communication

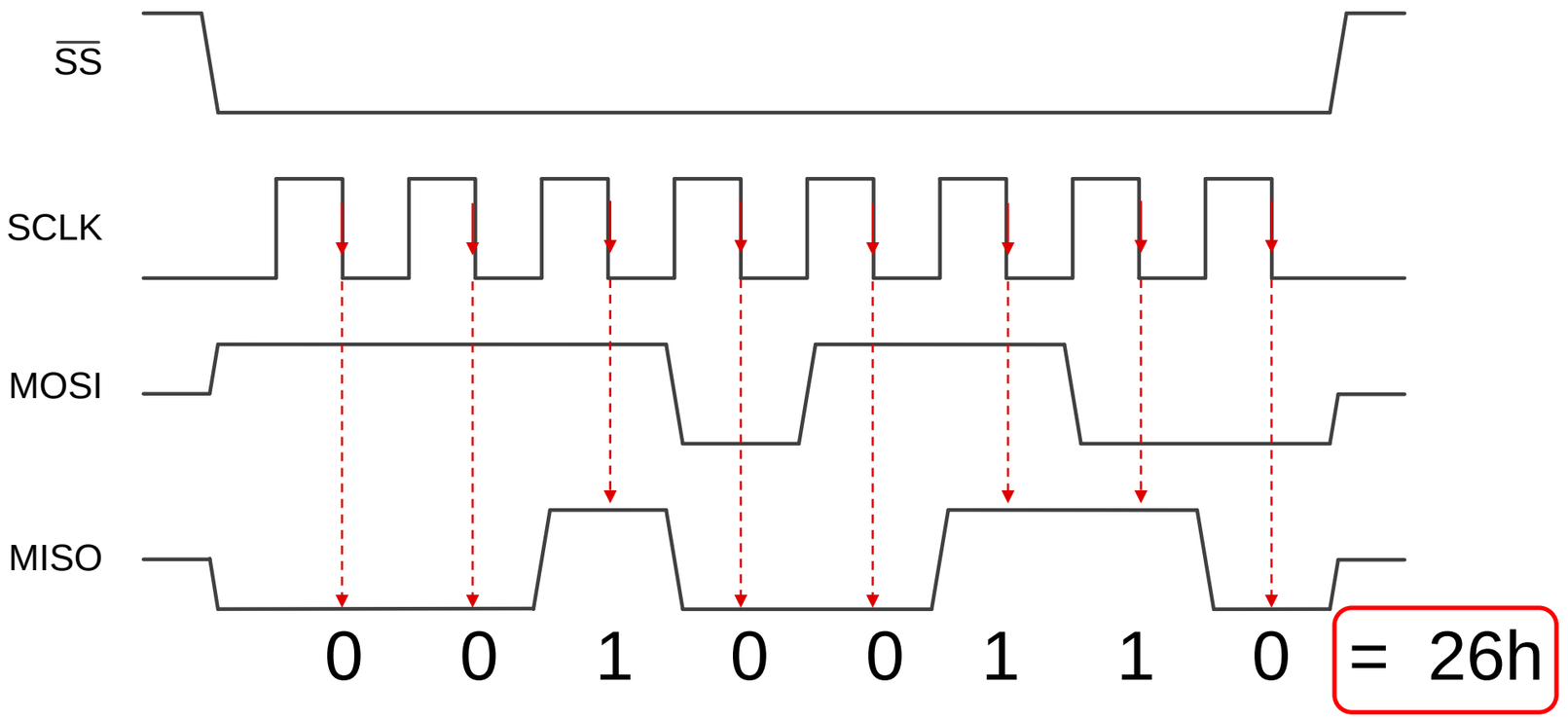
2. The following diagram is CPOL = 0, CPHA = 1. What hexadecimal byte is the sent from the peripheral to the controller in this transaction?
   a. 24
   b. 26
   c. E8
   d. EC



Quiz: Basics of SPI: Serial Communication

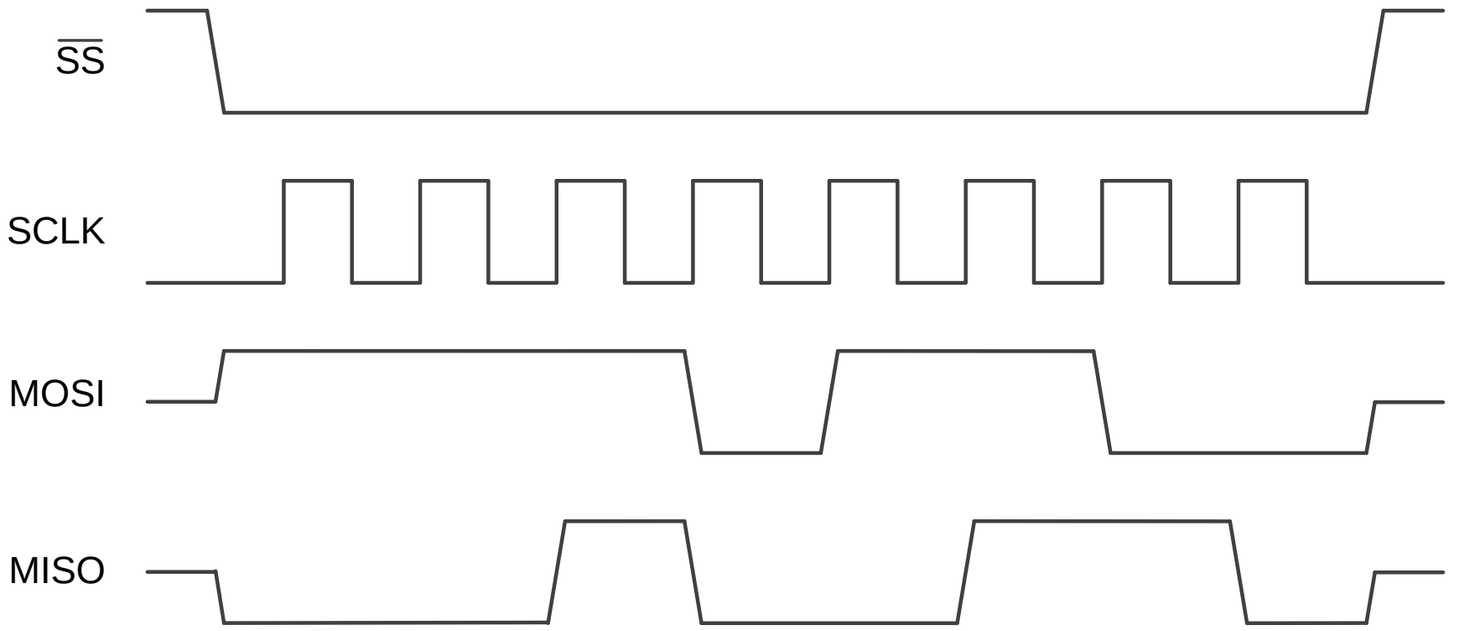
2. The following diagram is CPOL = 0, CPHA = 1. What hexadecimal byte is the sent from the peripheral to the controller in this transaction?

   a. 24
   b. 26
   c. E8
   d. EC



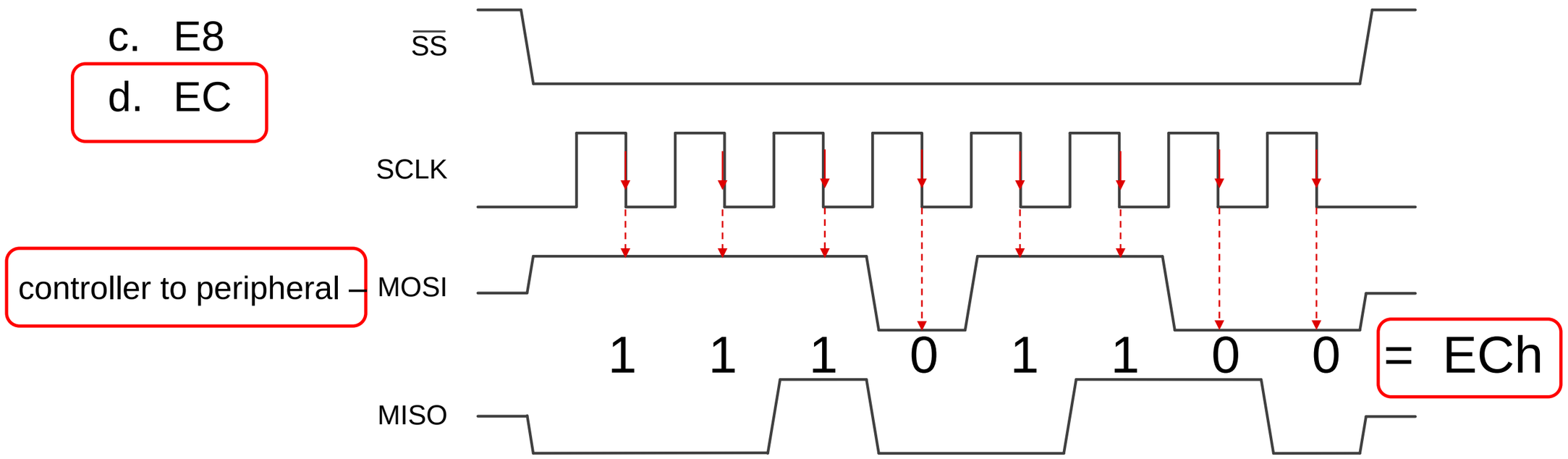
Quiz: Basics of SPI: Serial Communication

3. The following diagram is CPOL = 0, CPHA = 1. What hexadecimal byte is the sent from the controller to the peripheral in this transaction?
   a. 24
   b. 26
   c. E8
   d. EC



Quiz: Basics of SPI: Serial Communication

3. The following diagram is CPOL = 0, CPHA = 1. What hexadecimal byte is the sent from the controller to the peripheral in this transaction?
   a. 24
   b. 26
   c. E8
   d. EC



Thanks for your time!