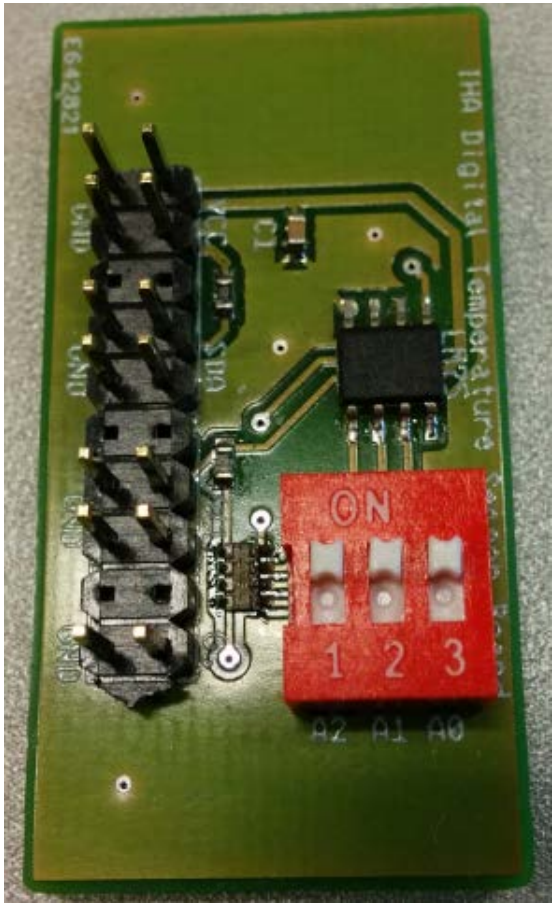
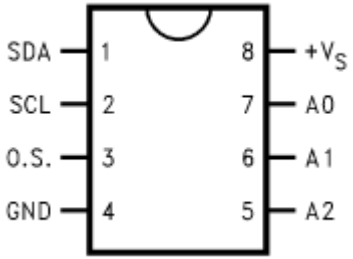
Communication buses lab experiment

I2C Communication



Read from LM75 temperature sensor



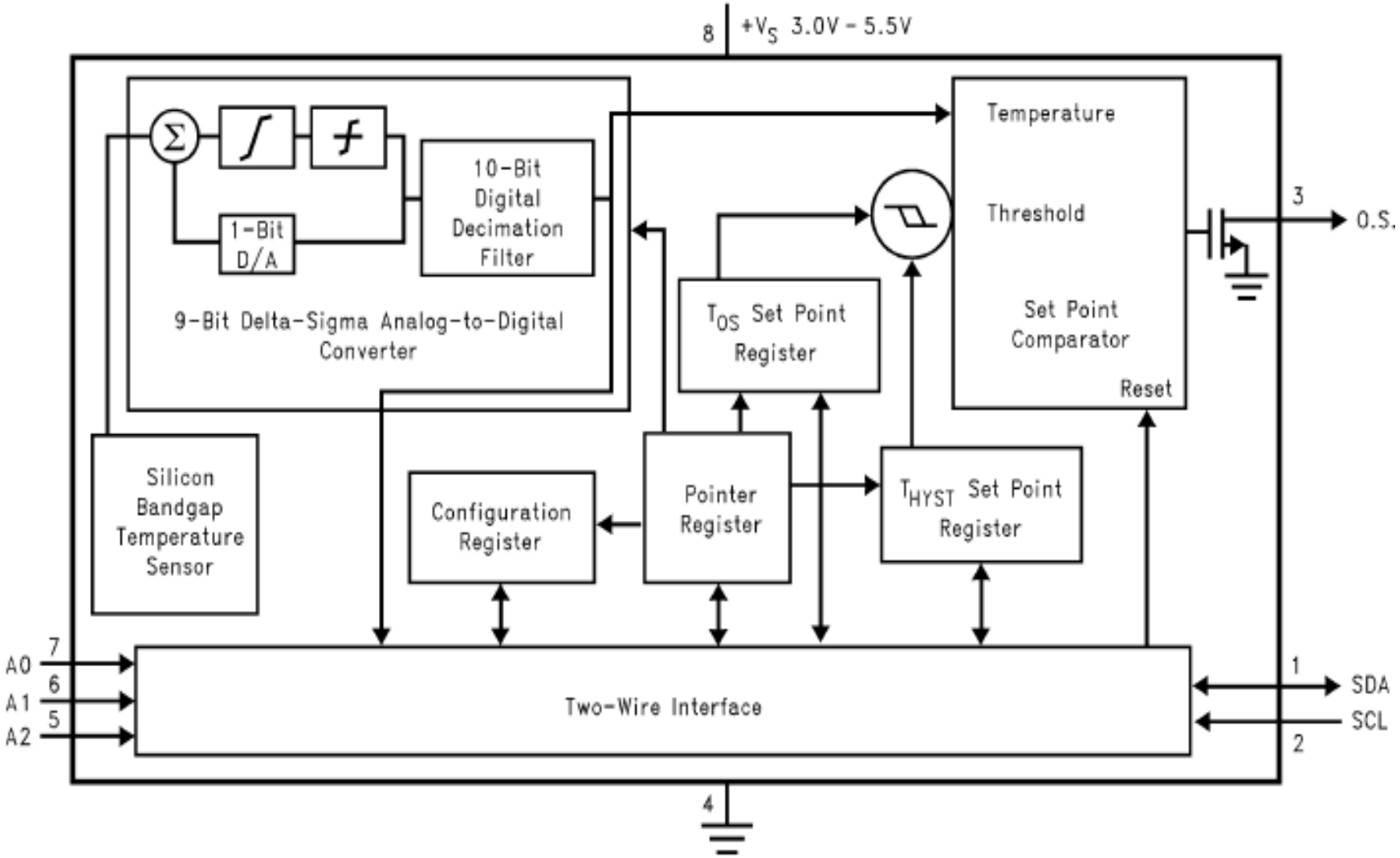
LM75
Digital Temperature Sensor and Thermal Watchdog with Two-Wire Interface

- I²C Bus interface
- Separate open-drain output pin operates as interrupt or comparator/thermostat output
- Register readback capability
- Power up defaults permit stand-alone operation as thermostat
- Shutdown mode to minimize power consumption
- Up to 8 LM75s can be connected to a single bus

Key Specifications

■ Supply Voltage		3.0V to 5.5V
■ Supply Current	operating	250 µA (typ)
		1 mA (max)
	shutdown	4 µA (typ)
■ Temperature Accuracy	–25˚C to 100˚C	±2˚C(max)
	–55˚C to 125˚C	±3˚C(max)

Block diagram



Address

1	0	0	1	A2	A1	A0
MSB						LSB

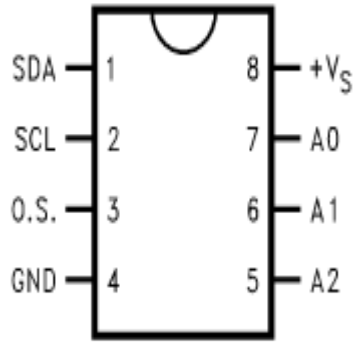
- LM75 is always an I2C slave.
- The 7 bit address consists of 4 hardcoded bits and 3 configurable bits.

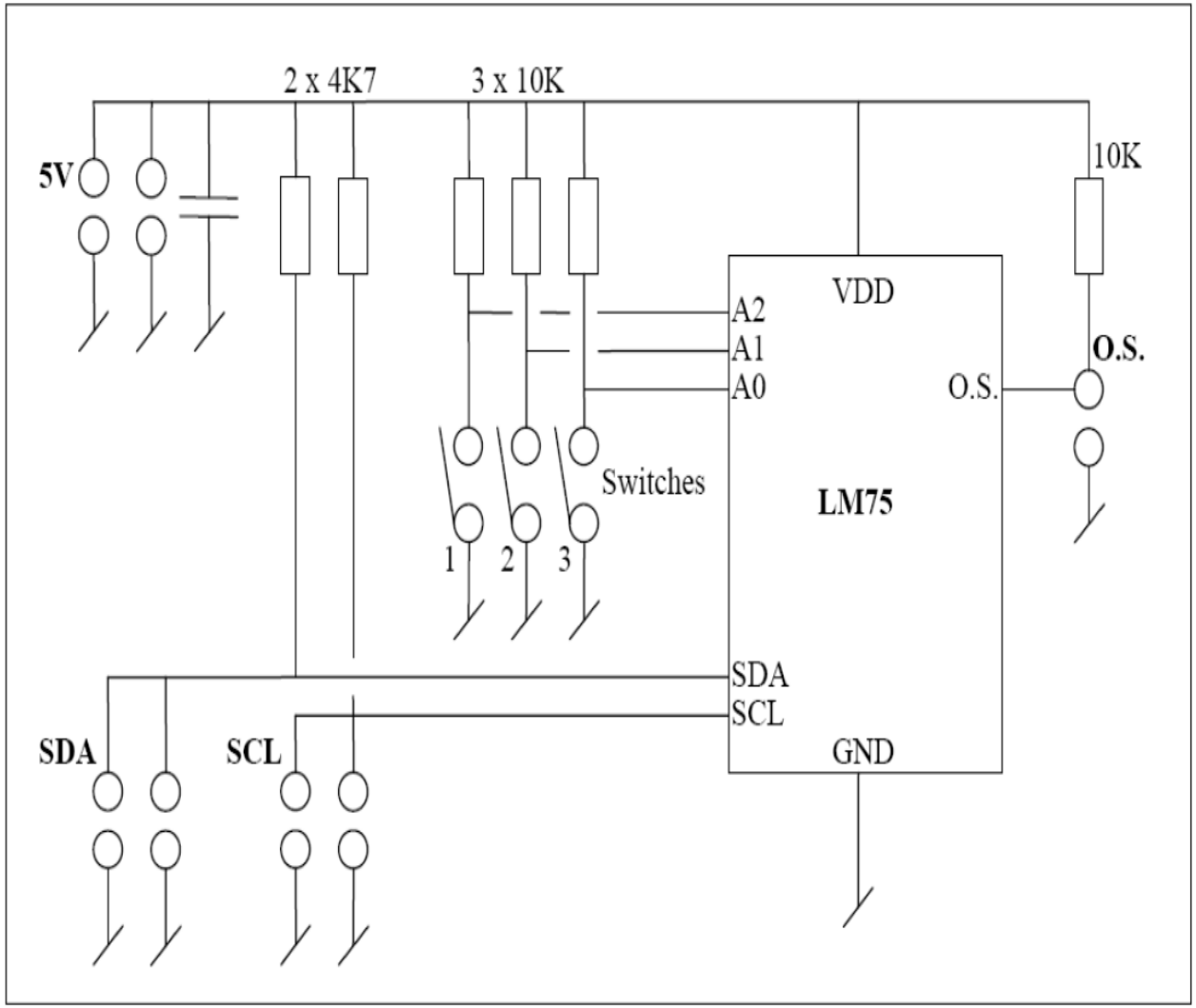
Temperature data

Temperature	Digital Output	
	Binary	Hex
+125°C	0 1111 1010	0FAh
+25°C	0 0011 0010	032h
+0.5°C	0 0000 0001	001h
0°C	0 0000 0000	000h
–0.5°C	1 1111 1111	1FFh
–25°C	1 1100 1110	1CEh
–55°C	1 1001 0010	192h

- 2's complement representation.
- 9 bits => a temperature is read as 2 bytes (endianess?).
- LSB = 0.5 deg. celcius.

LM75 diagram



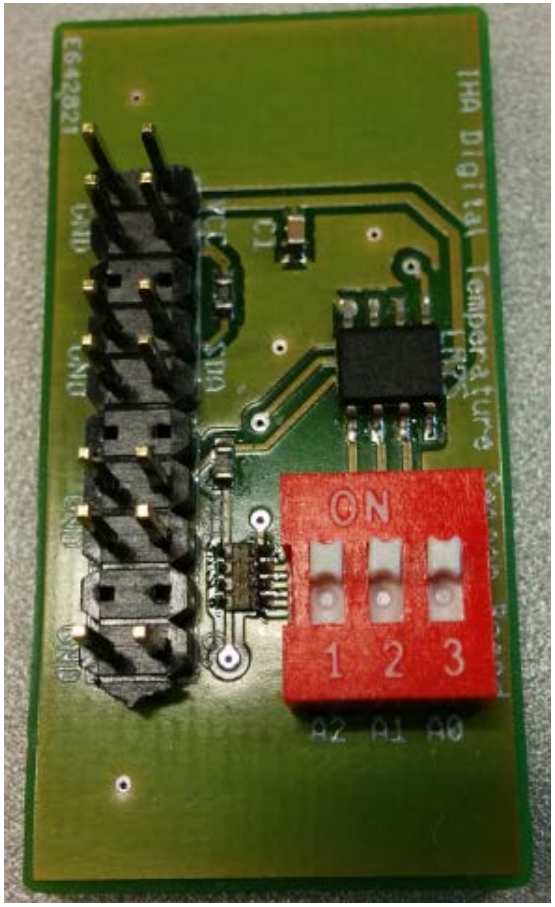


Address setting on the lab hardware

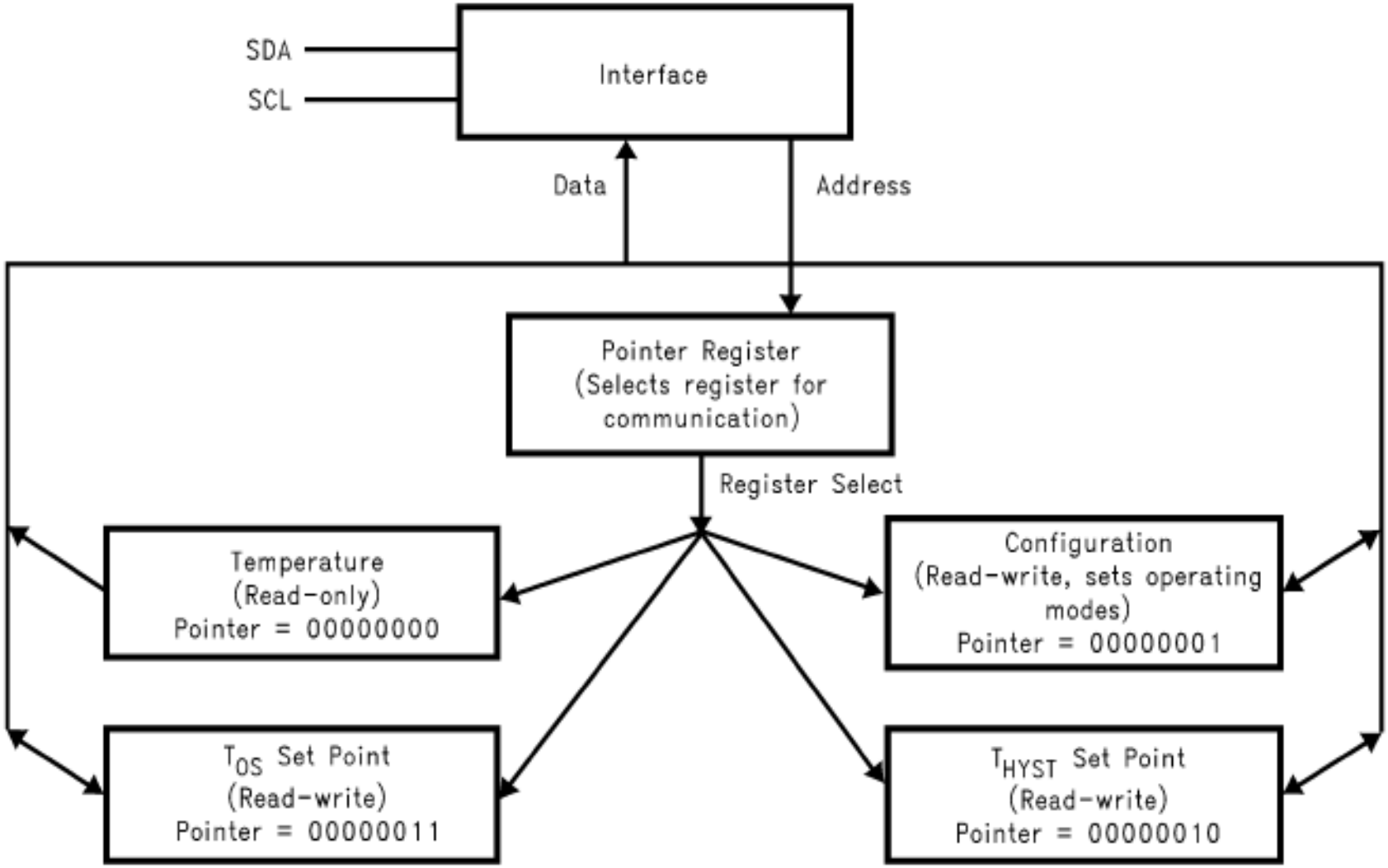
1	0	0	1	A2	A1	A0
MSB						LSB

Address for the setting in the picture:

1001000 (0x48)



Registers


SDA
SCL
Interface
Data
Address
Pointer Register
(Selects register for communication)
Register Select
Temperature
(Read-only)
Pointer = 00000000
Configuration
(Read-write, sets operating modes)
Pointer = 00000001
T_{OS} Set Point
(Read-write)
Pointer = 00000011
T_{HYST} Set Point
(Read-write)
Pointer = 00000010

SPI Communication

Write a program, where you can toggle the LED on the SPI Slave, by entering commands (i.e. pressing a key) in a console connected to the UART on the SPI Master.

Extend the program, so the SPI Master **continuously** reads the state of the push button from the SPI Slave and outputs the state to its UART. The push button is connected to P2.2.

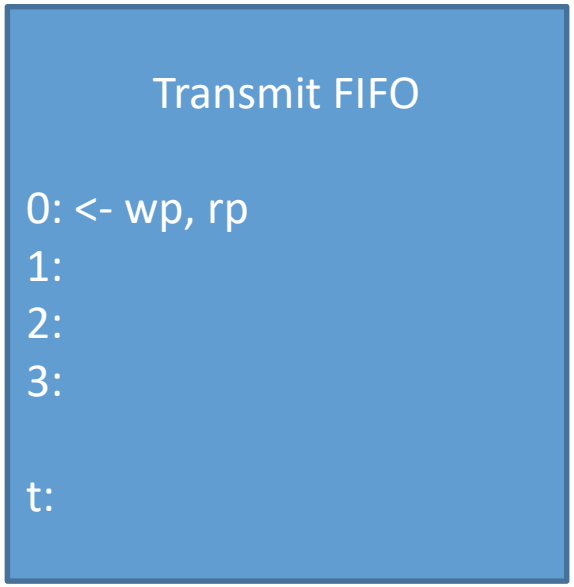
SPI Slave implementation in the PSoC5 LP

- The PSoC 5LP has a 4 byte transmit-buffer (FIFO) implemented in hardware.
- It also has a register, from which the actual transmission takes place.

- This is how it works...

Disclaimer:
This information is
found by experimenting
with the PSoC

SPI Slave implementation in the PSoC5 LP



Rules:

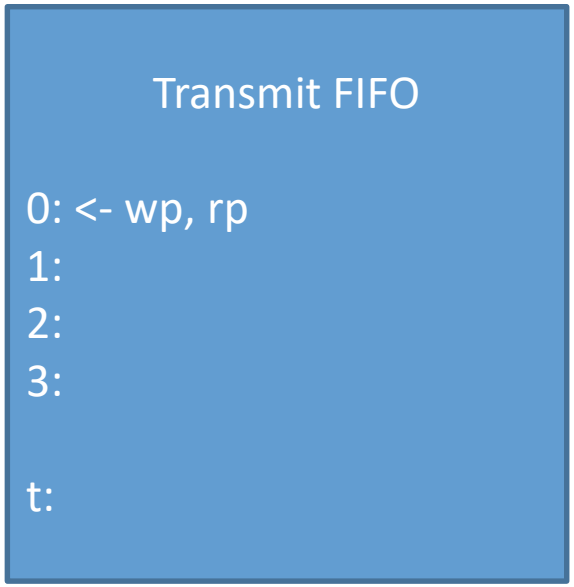
The buffer is circular.

Write pointer (wp) blocks writing if FIFO is full.

Read pointer (rp) will not go further than the write pointer.

Data is transfered from 't' register.

SPI Slave implementation in the PSoC5 LP



Rules:

The buffer is circular.

Write pointer (wp) blocks writing if FIFO is full.

Read pointer (rp) will not go further than the write pointer.

Data is transfered from 't' register.

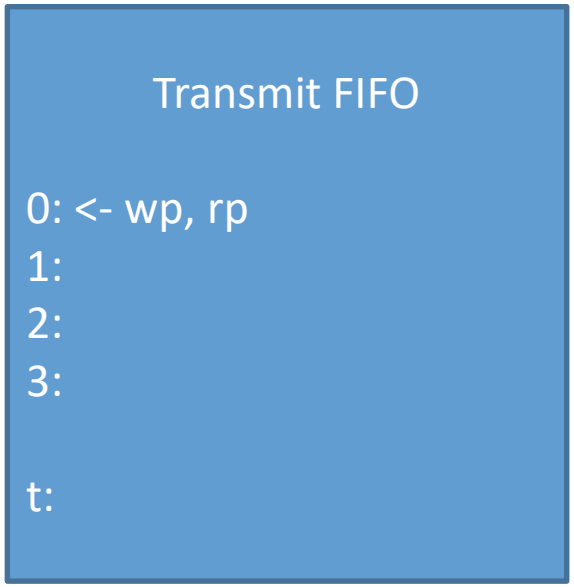
CPHA 0:

Transfer from **t** register is on leading clock edge.

Transfer from **rp** to **t** register is when a byte transfer is complete.

You can use the SPIS_WriteTxDataZero(byte) method to set the **t** register directly.

SPI Slave implementation in the PSoC5 LP



Rules:

The buffer is circular.

Write pointer (wp) blocks writing if FIFO is full.

Read pointer (rp) will not go further than the write pointer.

Data is transfered from 't' register.

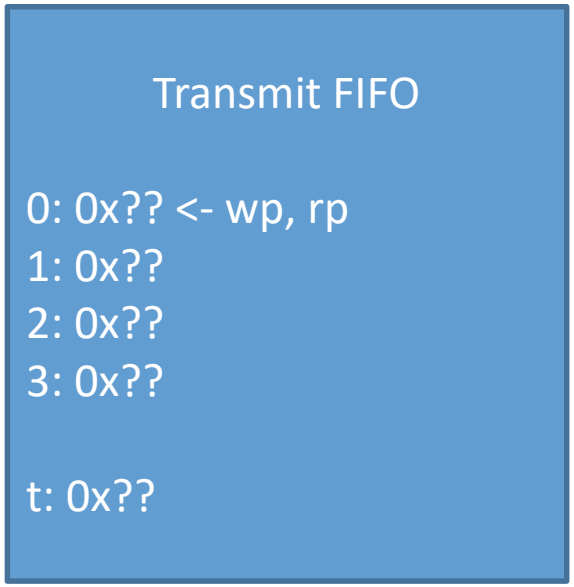
CPHA 1:

Transfer from **t** register is on trailing clock edge.

Transfer from **rp** to **t** register is before starting a byte transfer.

CPHA 0

CPHA0 example



Rules:

The buffer is circular.
Write pointer (wp) blocks writing if FIFO is full.
Read pointer (rp) will not go further than the write pointer.
Data is transfered from 't' register.

CPHA 0:
Transfer from **t** register is on leading clock edge.
Transfer from **rp** to **t** register is when a byte transfer is complete.
You can use the SPIS_WriteTxDataZero(byte) method to set the **t** register directly.

Write	0x01
Transfer	

CPHA0 example

Transmit FIFO

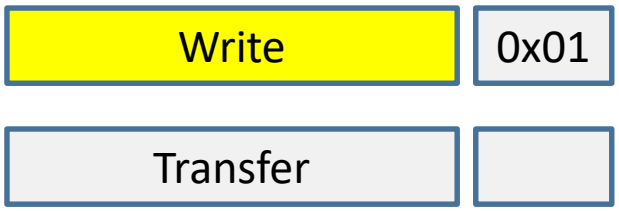
0: 0x01 <- rp
1: 0x?? <- wp
2: 0x??
3: 0x??

t: 0x??

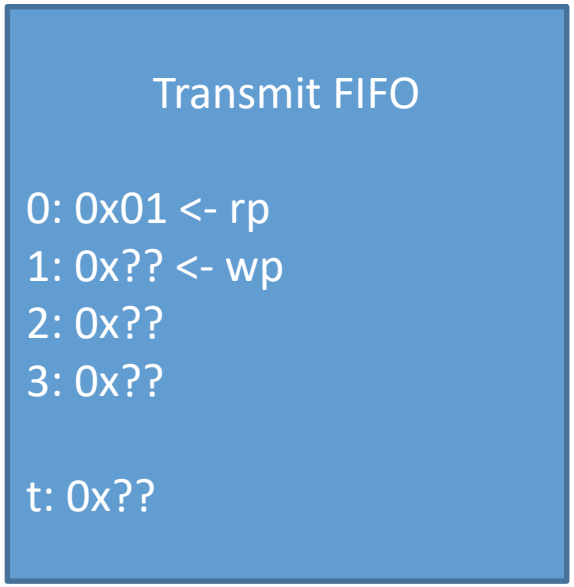
Rules:

The buffer is circular.
Write pointer (wp) blocks writing if FIFO is full.
Read pointer (rp) will not go further than the write pointer.
Data is transfered from 't' register.

CPHA 0:
Transfer from **t** register is on leading clock edge.
Transfer from **rp** to **t** register is when a byte transfer is complete.
You can use the SPIS_WriteTxDataZero(byte) method to set the **t** register directly.



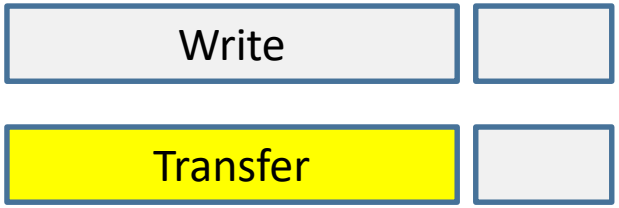
CPHA0 example



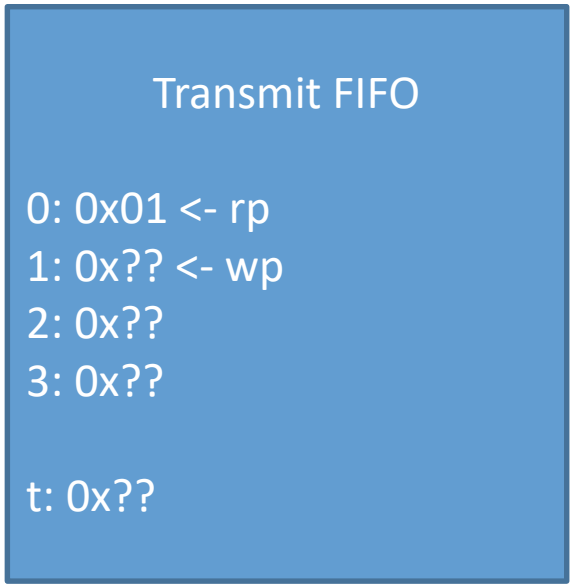
Rules:

The buffer is circular.
Write pointer (wp) blocks writing if FIFO is full.
Read pointer (rp) will not go further than the write pointer.
Data is transfered from 't' register.

CPHA 0:
Transfer from **t** register is on leading clock edge.
Transfer from **rp** to **t** register is when a byte transfer is complete.
You can use the SPIS_WriteTxDataZero(byte) method to set the **t** register directly.



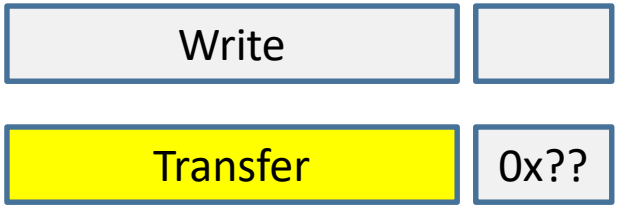
CPHA0 example



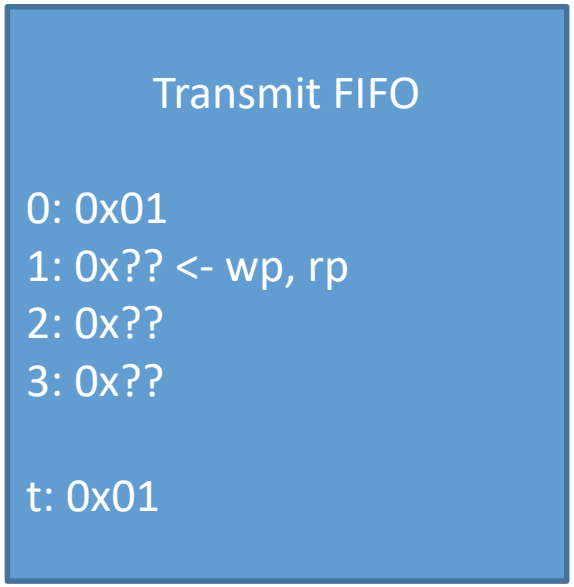
Rules:

The buffer is circular.
Write pointer (wp) blocks writing if FIFO is full.
Read pointer (rp) will not go further than the write pointer.
Data is transfered from 't' register.

CPHA 0:
Transfer from **t** register is on leading clock edge.
Transfer from **rp** to **t** register is when a byte transfer is complete.
You can use the SPIS_WriteTxDataZero(byte) method to set the **t** register directly.



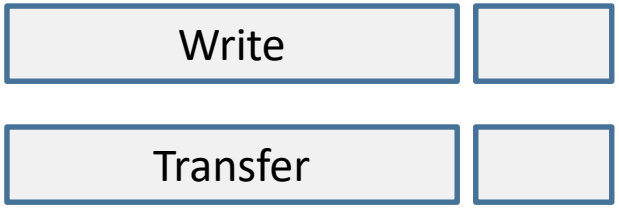
CPHA0 example



Rules:

The buffer is circular.
Write pointer (wp) blocks writing if FIFO is full.
Read pointer (rp) will not go further than the write pointer.
Data is transfered from 't' register.

CPHA 0:
Transfer from **t** register is on leading clock edge.
Transfer from **rp** to **t** register is when a byte transfer is complete.
You can use the SPIS_WriteTxDataZero(byte) method to set the **t** register directly.



CPHA0 example

Transmit FIFO

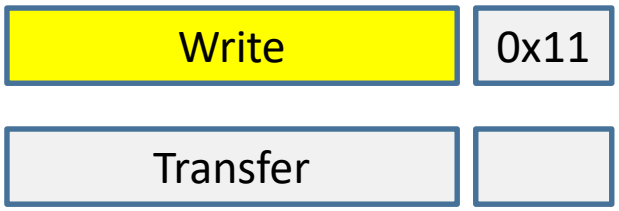
0: 0x01
1: 0x11 <- rp
2: 0x?? <- wp
3: 0x??

t: 0x01

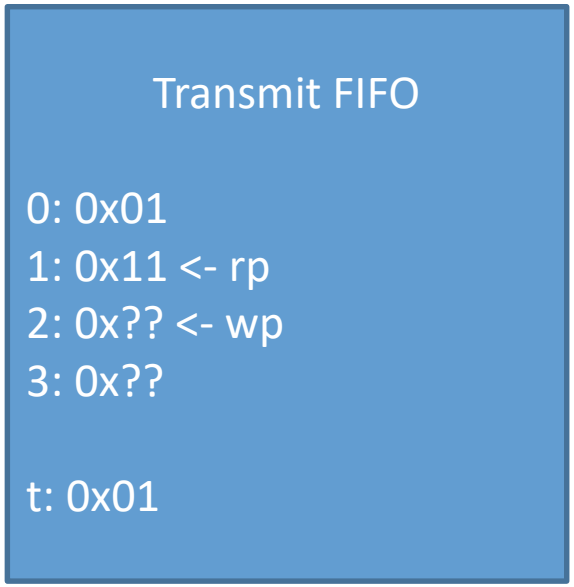
Rules:

The buffer is circular.
Write pointer (wp) blocks writing if FIFO is full.
Read pointer (rp) will not go further than the write pointer.
Data is transfered from 't' register.

CPHA 0:
Transfer from **t** register is on leading clock edge.
Transfer from **rp** to **t** register is when a byte transfer is complete.
You can use the SPIS_WriteTxDataZero(byte) method to set the **t** register directly.



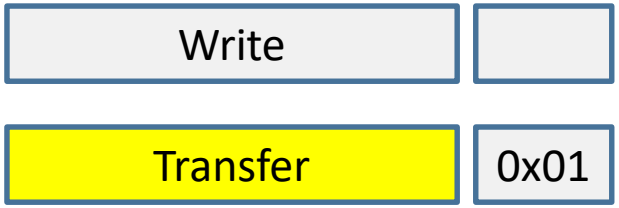
CPHA0 example



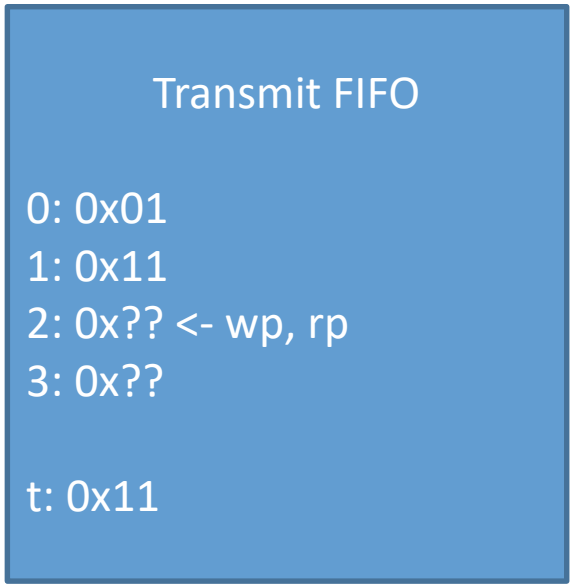
Rules:

The buffer is circular.
Write pointer (wp) blocks writing if FIFO is full.
Read pointer (rp) will not go further than the write pointer.
Data is transfered from 't' register.

CPHA 0:
Transfer from **t** register is on leading clock edge.
Transfer from **rp** to **t** register is when a byte transfer is complete.
You can use the SPIS_WriteTxDataZero(byte) method to set the **t** register directly.



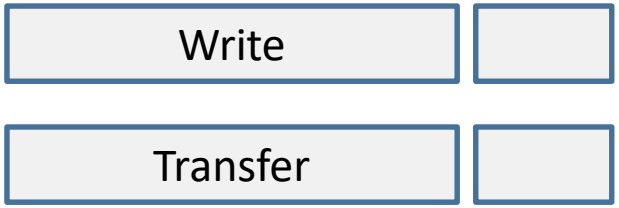
CPHA0 example



Rules:

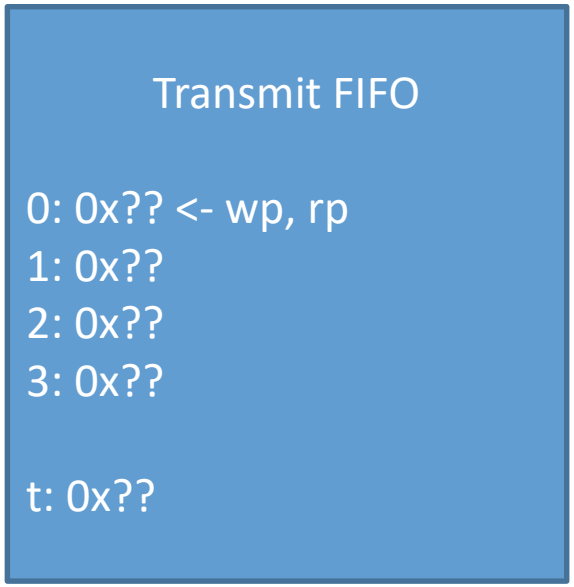
The buffer is circular.
Write pointer (wp) blocks writing if FIFO is full.
Read pointer (rp) will not go further than the write pointer.
Data is transfered from 't' register.

CPHA 0:
Transfer from **t** register is on leading clock edge.
Transfer from **rp** to **t** register is when a byte transfer is complete.
You can use the SPIS_WriteTxDataZero(byte) method to set the **t** register directly.



CPHA 1

CPHA1 example



Rules:

The buffer is circular.
Write pointer (wp) blocks writing if FIFO is full.
Read pointer (rp) will not go further than the write pointer.
Data is transfered from 't' register.

CPHA 1:
Transfer from **t** register is on trailing clock edge.
Transfer from **rp** to **t** register is before starting a byte transfer.

Write	0x01
Transfer	

CPHA1 example

Transmit FIFO

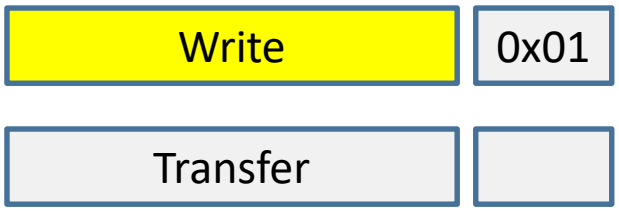
0: 0x01 <- rp
1: 0x?? <- wp
2: 0x??
3: 0x??

t: 0x??

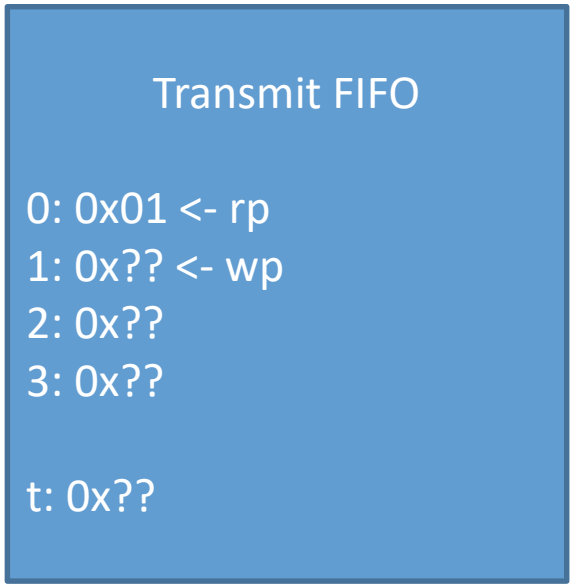
Rules:

The buffer is circular.
Write pointer (wp) blocks writing if FIFO is full.
Read pointer (rp) will not go further than the write pointer.
Data is transfered from 't' register.

CPHA 1:
Transfer from **t** register is on trailing clock edge.
Transfer from **rp** to **t** register is before starting a byte transfer.



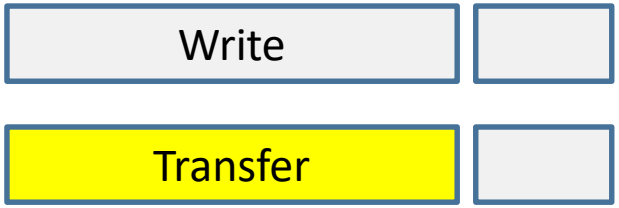
CPHA1 example



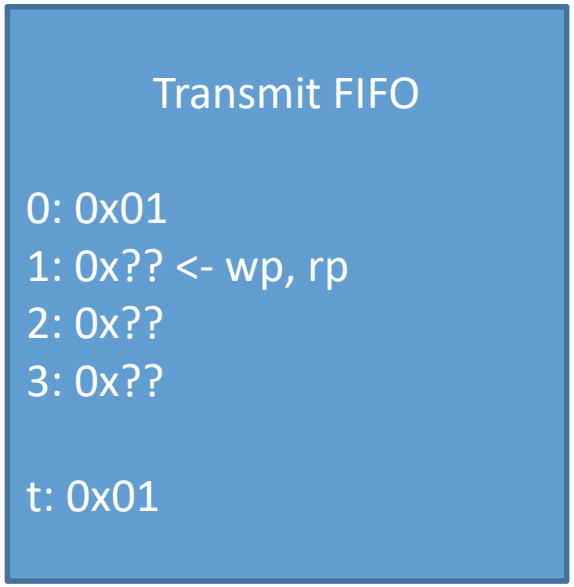
Rules:

The buffer is circular.
Write pointer (wp) blocks writing if FIFO is full.
Read pointer (rp) will not go further than the write pointer.
Data is transfered from 't' register.

CPHA 1:
Transfer from **t** register is on trailing clock edge.
Transfer from **rp** to **t** register is before starting a byte transfer.



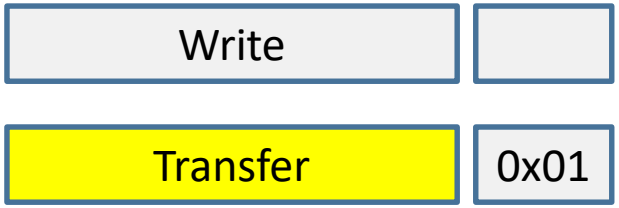
CPHA1 example



Rules:

The buffer is circular.
Write pointer (wp) blocks writing if FIFO is full.
Read pointer (rp) will not go further than the write pointer.
Data is transfered from 't' register.

CPHA 1:
Transfer from **t** register is on trailing clock edge.
Transfer from **rp** to **t** register is before starting a byte transfer.



CPHA1 example

Transmit FIFO

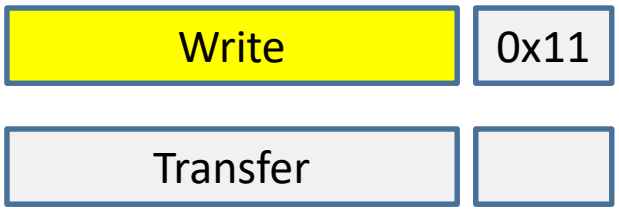
0: 0x01
1: 0x11 < rp
2: 0x?? <- wp
3: 0x??

t: 0x01

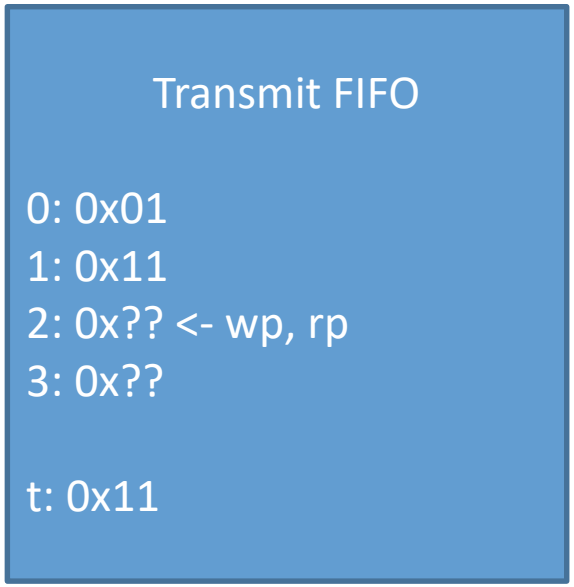
Rules:

The buffer is circular.
Write pointer (wp) blocks writing if FIFO is full.
Read pointer (rp) will not go further than the write pointer.
Data is transfered from 't' register.

CPHA 1:
Transfer from **t** register is on trailing clock edge.
Transfer from **rp** to **t** register is before starting a byte transfer.



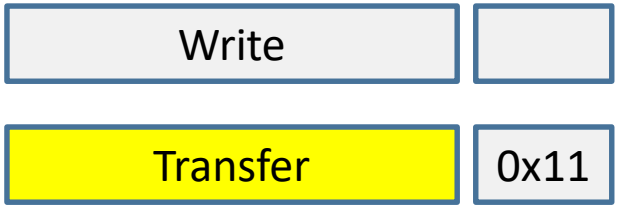
CPHA1 example



Rules:

The buffer is circular.
Write pointer (wp) blocks writing if FIFO is full.
Read pointer (rp) will not go further than the write pointer.
Data is transfered from 't' register.

CPHA 1:
Transfer from **t** register is on trailing clock edge.
Transfer from **rp** to **t** register is before starting a byte transfer.



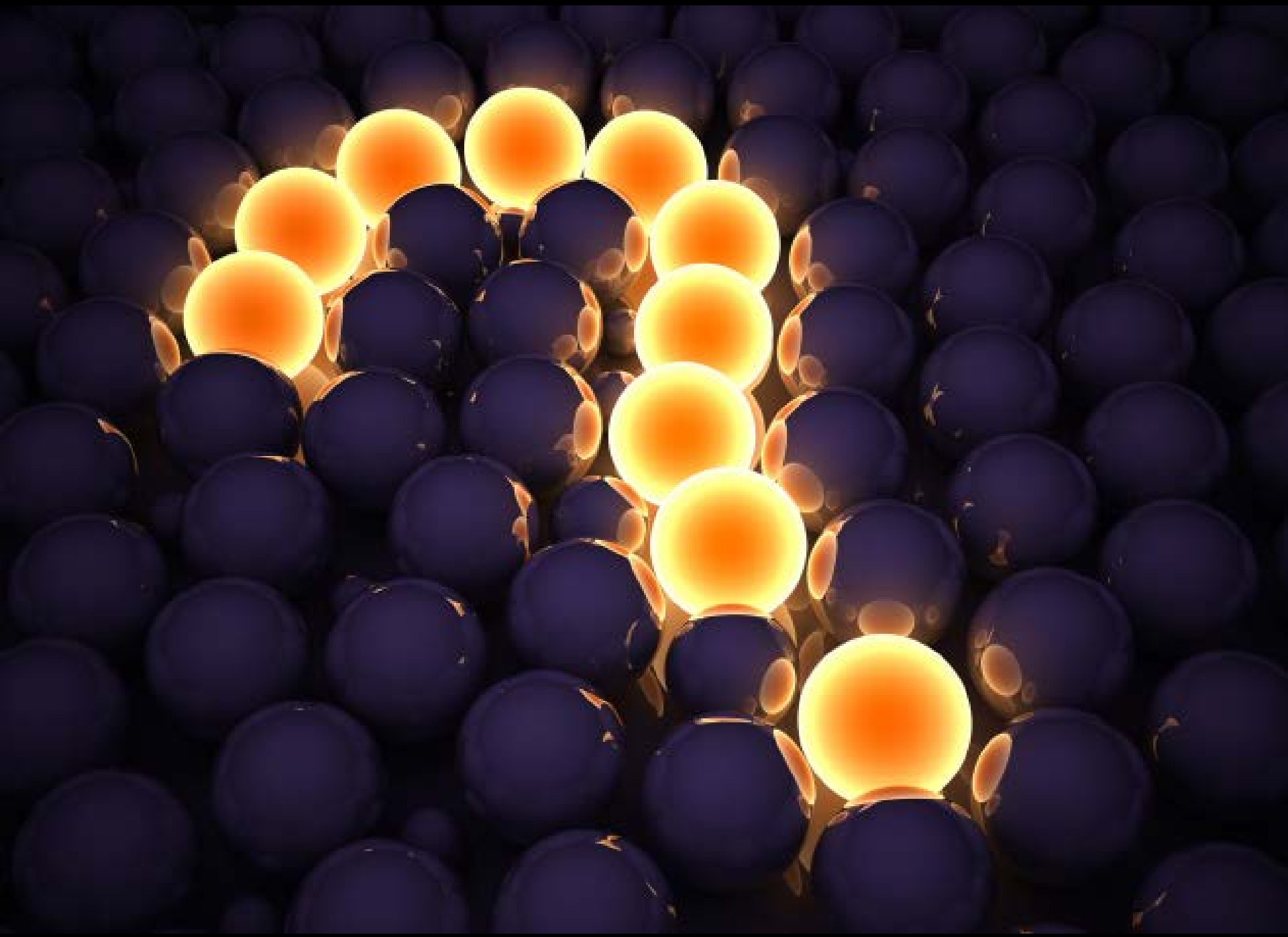


Image resources

- Question mark: https://wall.alphacoders.com/big.php?i=437563