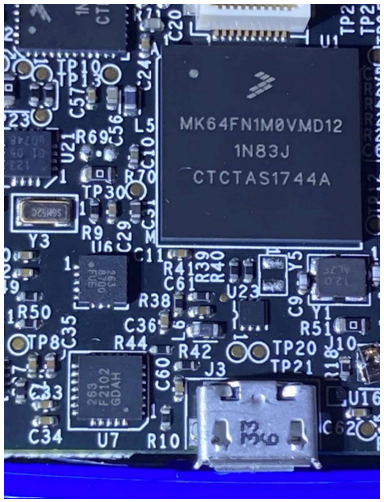
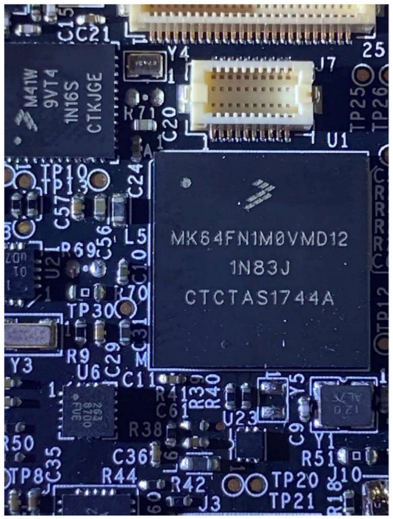
EXHIBIT 19

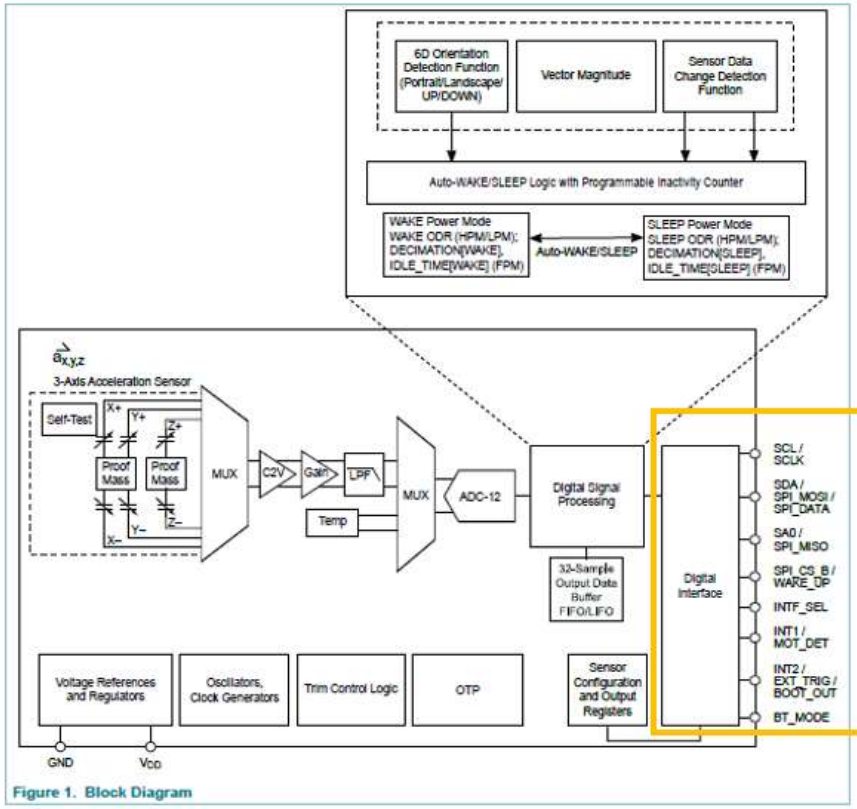
EXHIBIT 19

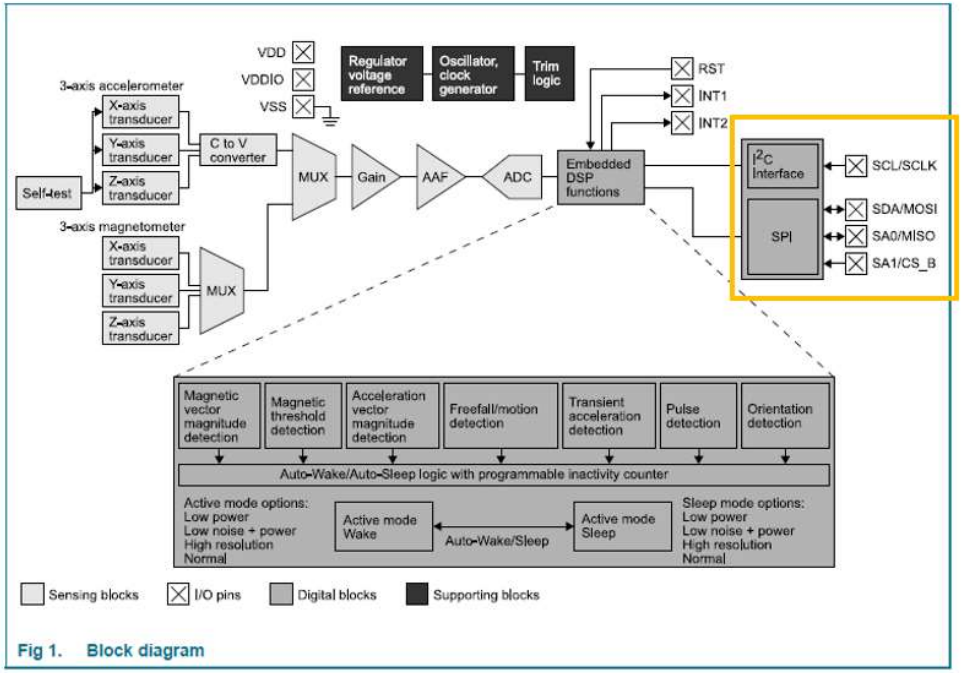
UNITED STATES PATENT NO. 7,231,474

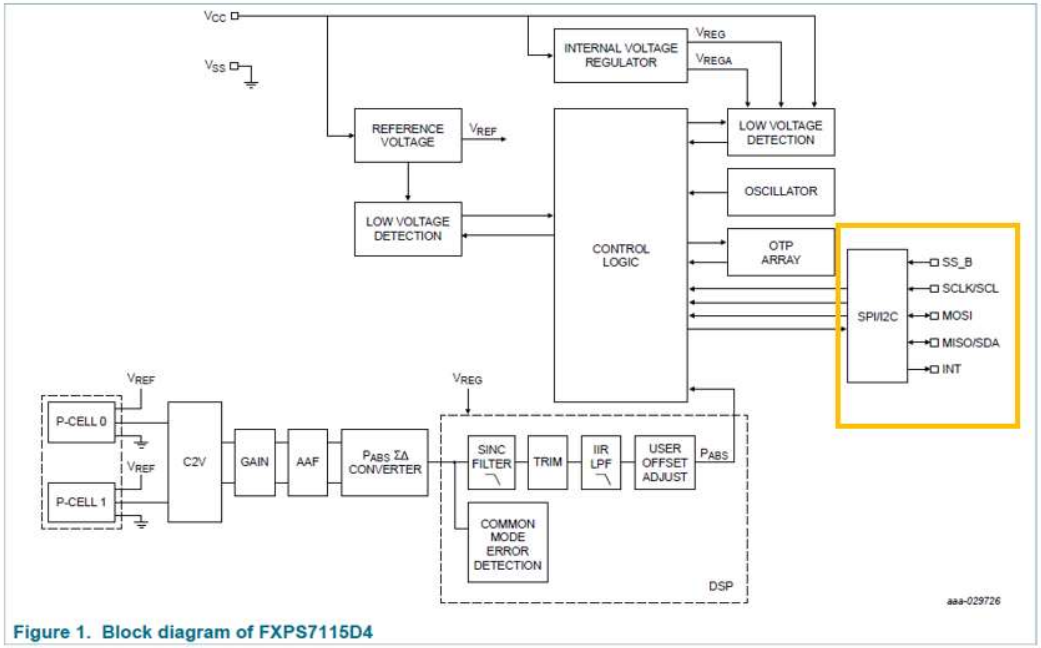
MediaTek hereby identifies evidence demonstrating the infringement of following NXP products: NXP Sensors specifically including but not limited to FXOS8700CQ, FXLS8962AF, FXPS7115D4 (the "Accused '474 Sensors"); NXP wireless microcontrollers specifically including but not limited to K32W061 and K32W041 (the "Accused '474 Wireless Microcontrollers"); and NXP i.MX Applications Processors specifically including but not limited to i.MX6 Dual and i.MX6 Quad (the "Accused '474 i.MX Processors") (collectively, the "Accused '474 products"). The chart below is based on evidence of representative products of the Accused '474 Products.

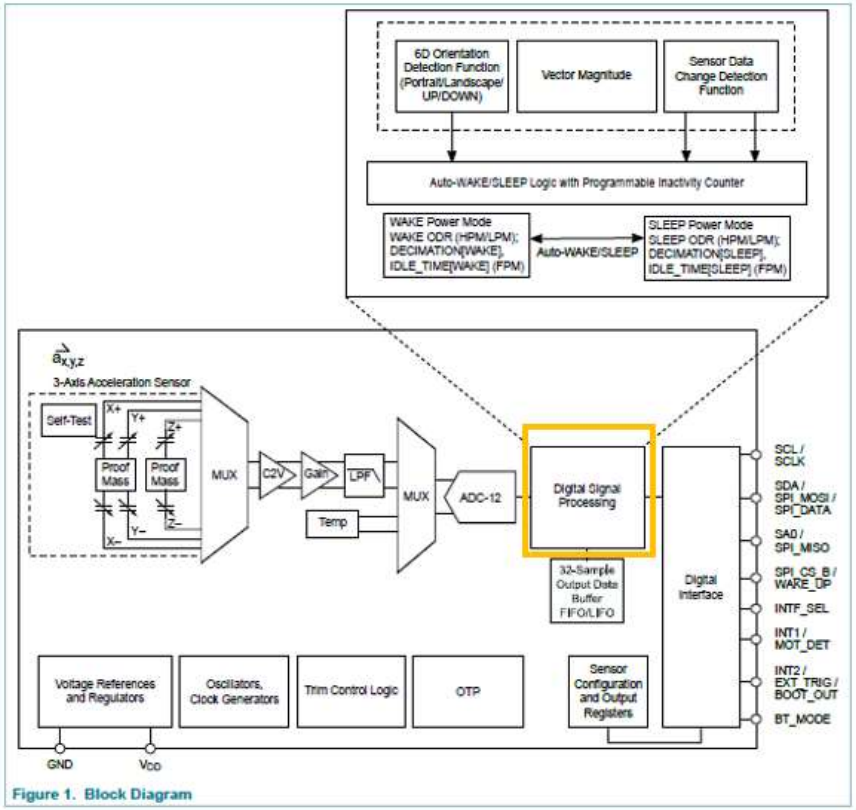
'474 Patent Claim	Representative NXP Product(s)
[1a.] A serial communication system comprising:	To the extent the preamble is limiting, the Accused '474 Sensors include a "serial communication system" as recited in the '474 patent. Exemplary systems and serial interfaces are identified in the block diagrams and rectangles below. *See, e.g.,* 

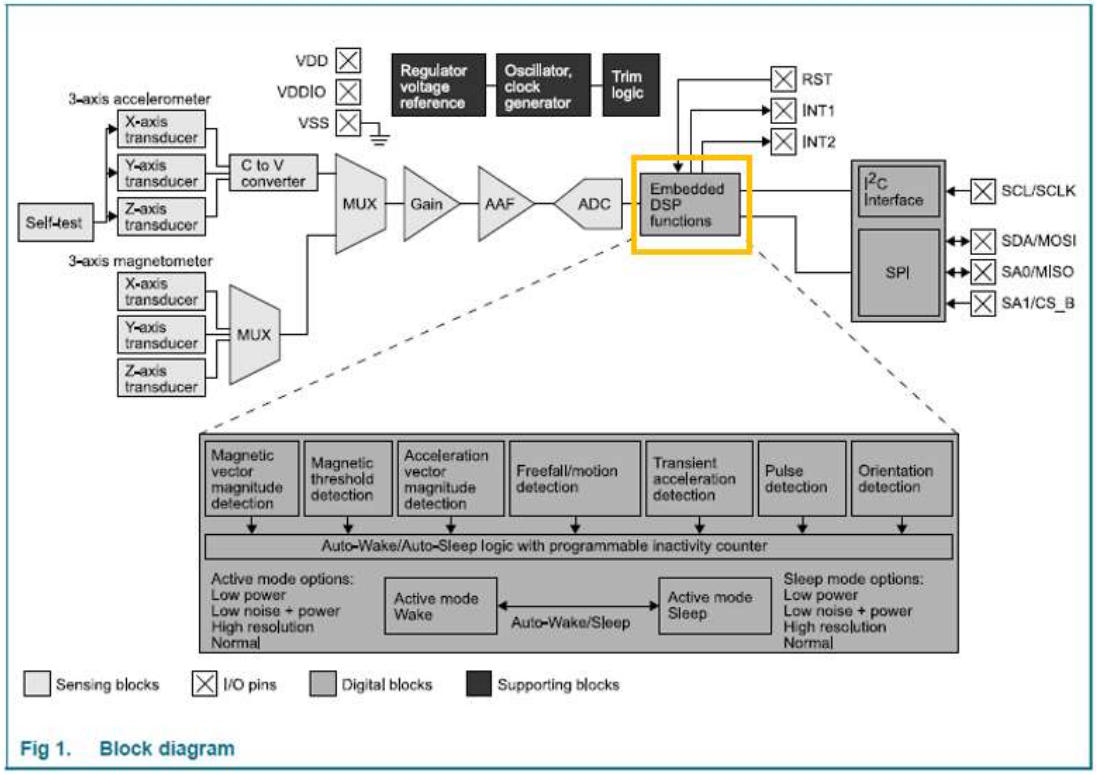
’474 Patent Claim	Representative NXP Product(s)
	https://www.avnet.com/shop/us/products/nxp/fxos8700cqr1-3074457345626313537?fromPage=autoSuggest&langId=-1&autoSuggestSearchTerm=FXOS  SLN-RPK-NODE (development board of FXOS8700CQ) (Arrow) 

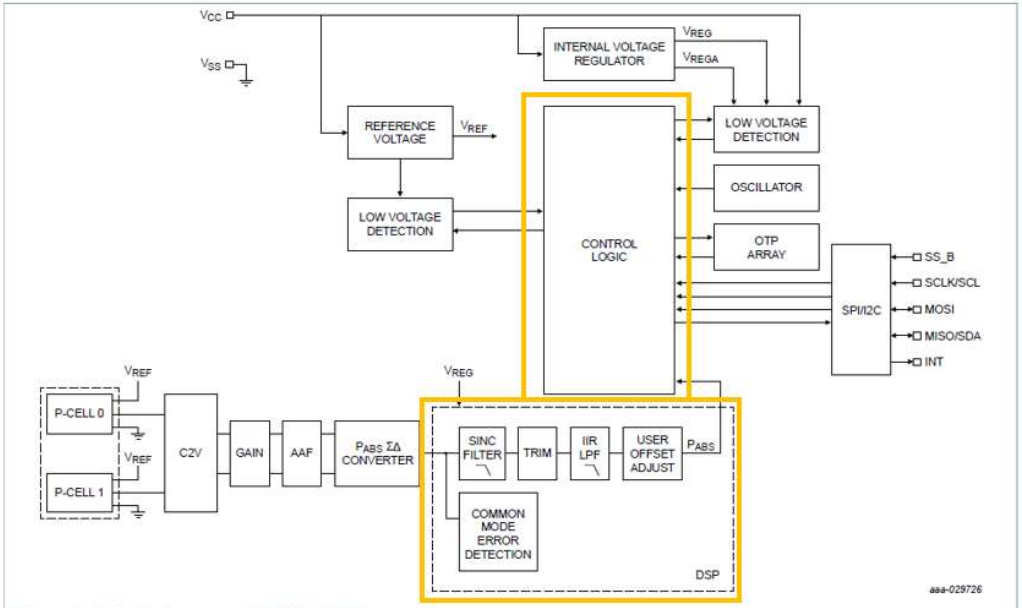
’474 Patent Claim	Representative NXP Product(s)
	SLN-RPK-NODE (development board of FXOS8700CQ) (Mouser) NXP Semiconductors **FXLS8962AF** 3-Axis Low-g Accelerometer **5 Block diagram**  Figure 1. Block Diagram FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 3.

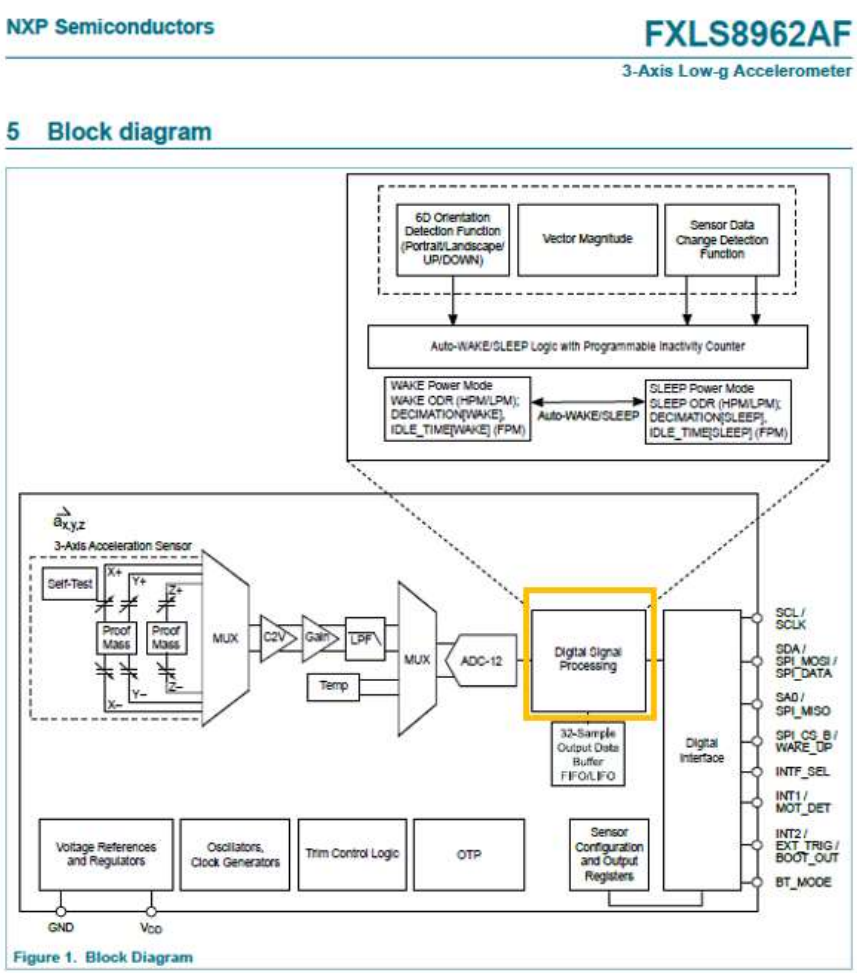
’474 Patent Claim	Representative NXP Product(s)
	NXP Semiconductors **FXOS8700CQ** 6-axis sensor with integrated linear accelerometer and magnetometer **5. Block diagram**  Fig 1. Block diagram FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 3.

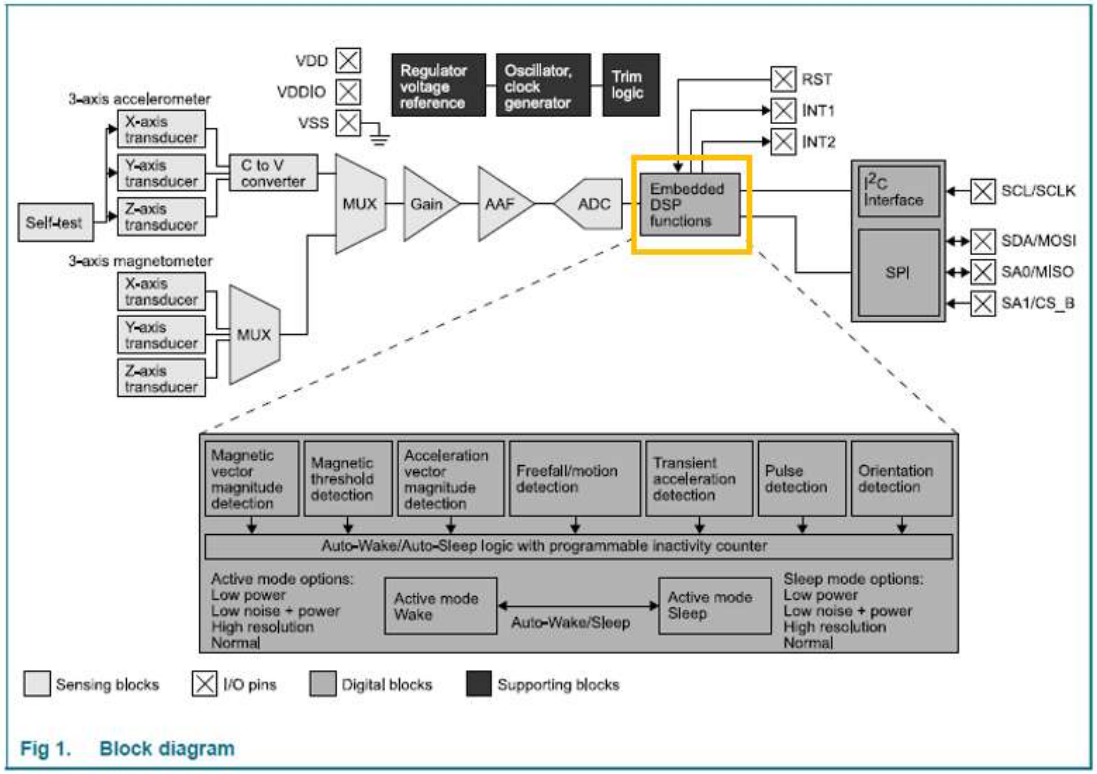
’474 Patent Claim	Representative NXP Product(s)
	NXP Semiconductors **FXPS7115D4** Digital absolute pressure sensor, 40 kPa to 115 kPa **5 Block diagram**  **Figure 1. Block diagram of FXPS7115D4** FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 3.

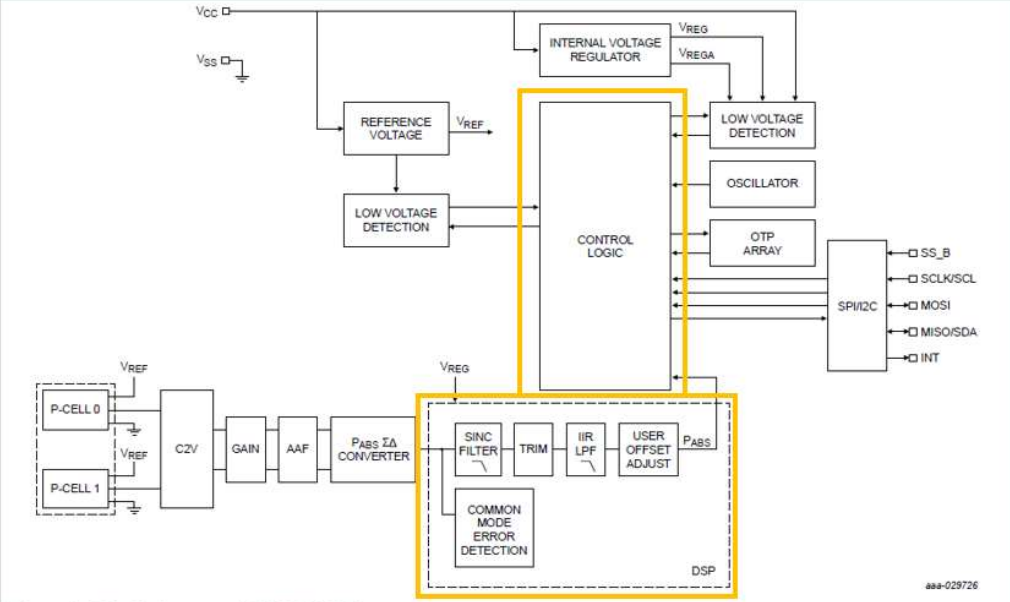
’474 Patent Claim	Representative NXP Product(s)
	NXP Semiconductors **FXLS8962AF** 3-Axis Low-g Accelerometer **5 Block diagram**  Figure 1. Block Diagram FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 3.

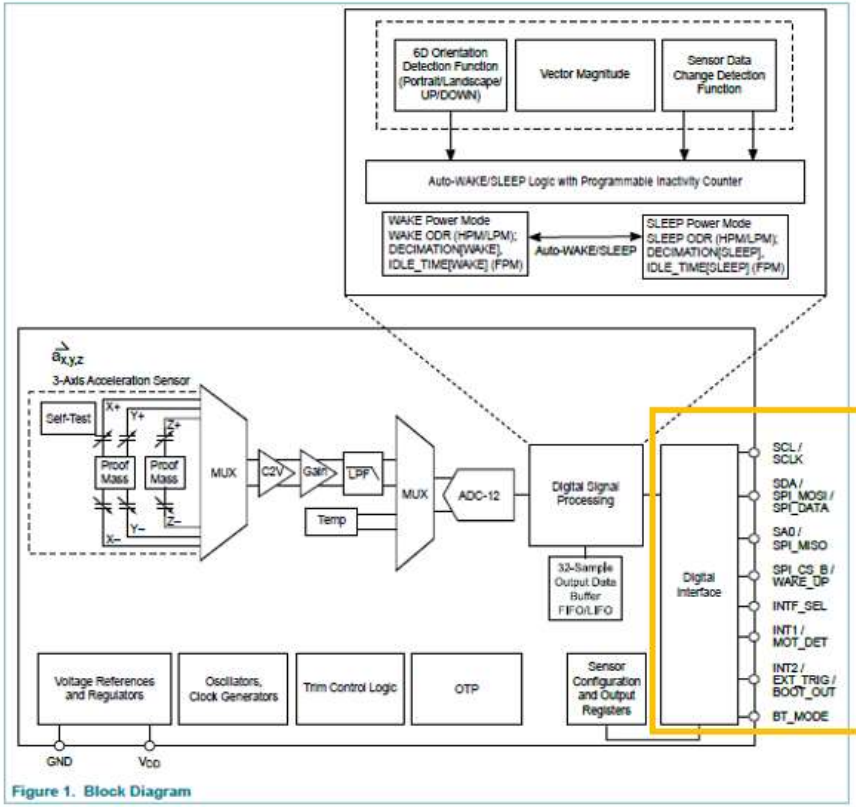
’474 Patent Claim	Representative NXP Product(s)
	NXP Semiconductors **FXOS8700CQ** 6-axis sensor with integrated linear accelerometer and magnetometer **5. Block diagram**  **Fig 1. Block diagram** FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 3.

’474 Patent Claim	Representative NXP Product(s)
	NXP Semiconductors **FXPS7115D4** Digital absolute pressure sensor, 40 kPa to 115 kPa **5 Block diagram**  Figure 1. Block diagram of FXPS7115D4 FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 3.
[1b.] an integrated circuit having a master serial interface; and	The Accused ’474 Sensors each includes an integrated circuit having a master serial interface. For example, the Accused ’474 Sensors each includes an integrated circuit having a master serial interface (*e.g.*, the SPI/I2C interface on the master/MCU and/or the host processor).

’474 Patent Claim	Representative NXP Product(s)
	See, e.g.:  FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 3.

’474 Patent Claim	Representative NXP Product(s)
	NXP Semiconductors **FXOS8700CQ** 6-axis sensor with integrated linear accelerometer and magnetometer **5. Block diagram**  **Fig 1. Block diagram** FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 3.

’474 Patent Claim	Representative NXP Product(s)
	NXP Semiconductors **FXPS7115D4** Digital absolute pressure sensor, 40 kPa to 115 kPa **5 Block diagram**  Figure 1. Block diagram of FXPS7115D4 FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 3.

’474 Patent Claim	Representative NXP Product(s)
	NXP Semiconductors **FXLS8962AF** 3-Axis Low-g Accelerometer **5 Block diagram**  Figure 1. Block Diagram FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 3.

’474 Patent Claim	Representative NXP Product(s)

10.1.2 I^2C read/write operations

10.1.2.1 Single byte read

The master transmits a start condition (ST) to FXLS8962AF, followed by the slave address, with the R/W bit set to '0' for a write, and the FXLS8962AF sends an acknowledgement. Then the master transmits the address of the register to read and the FXLS8962AF sends an acknowledgement. The master transmits a repeated start condition (SR), followed by the slave address with the R/W bit set to '1' for a read from the previously selected register. The FXLS8962AF then acknowledges and transmits the data from the requested register. The master does not acknowledge (NAK) the transmitted data and then transmits a stop condition to end the data transfer.

FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 18.

11 SPI interface

The SPI interface is a classical Master/Slave serial port. FXLS8962AF is always considered to be the slave device and thus never initiates communication with the host processor.

The SPI interface of FXLS8962AF is compatible with interface mode 00, corresponding to CPOL = 0 and CPHA = 0.

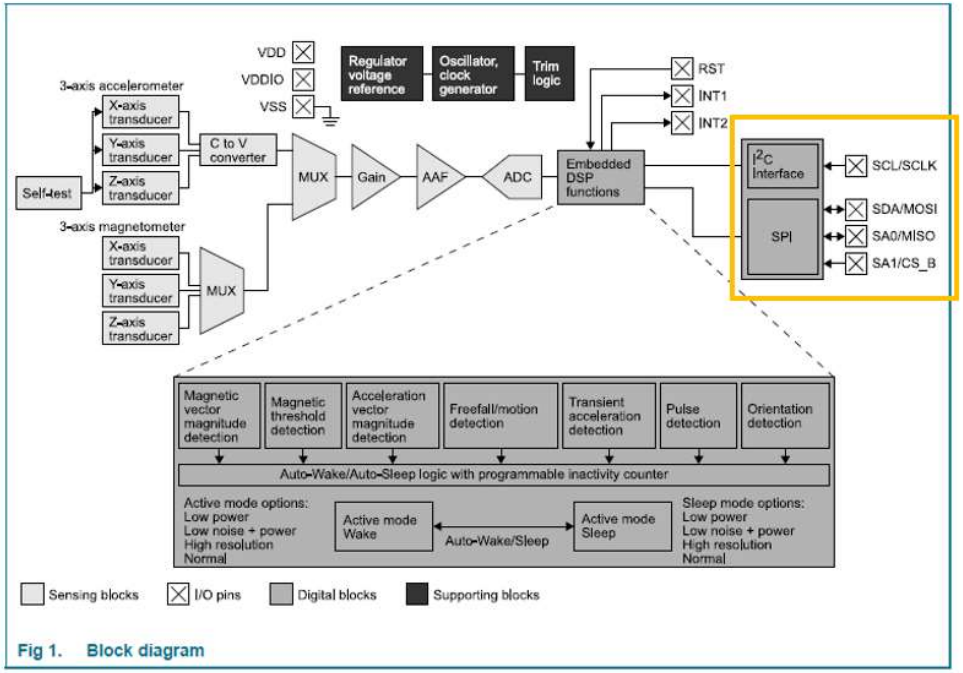
For CPOL = 0, the idle value of the clock is zero, and the active value of the clock is 1. For CPHA = 0, data is captured on the clock's rising edge (low to high transition) and data is propagated on the clock's falling edge (high to low transition).

FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 20.

10.1.1 General I^2C operation

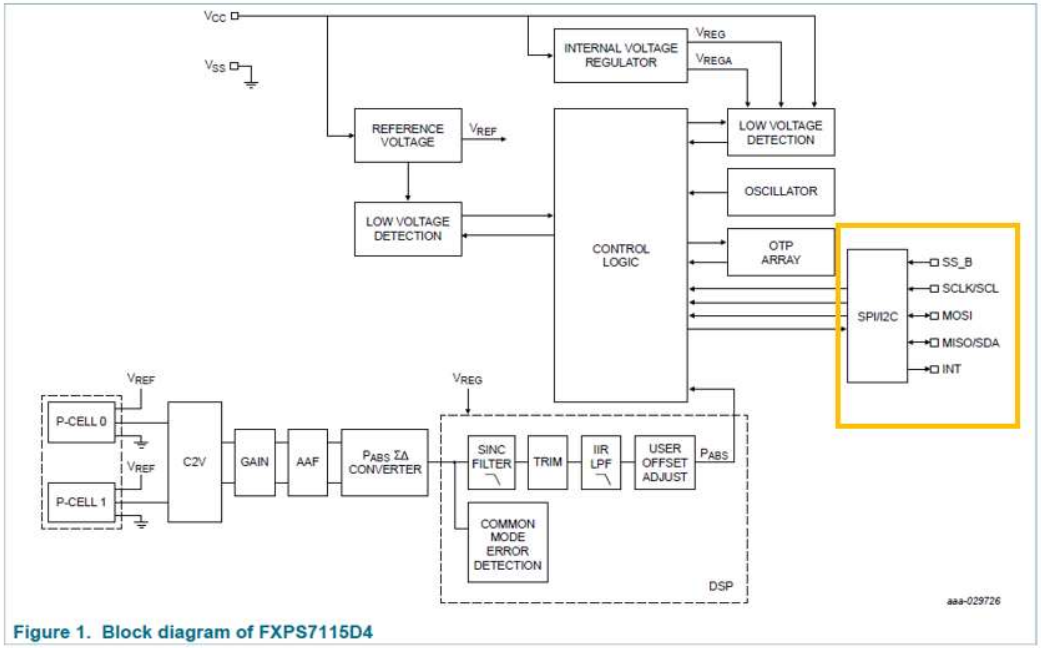
There are two signals associated with the I^2C-bus: the Serial Clock Line (SCL) and the Serial Data line (SDA). SDA is a bidirectional signal used for sending and receiving the data to/from the interface. External pull-up resistors connected to V_{DD} are required for SDA and SCL. When the I^2C-bus is free, SCL and SDA are high.

’474 Patent Claim	Representative NXP Product(s)
	FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 18. **11.1 General SPI operation** The SPI_CS_B pin is driven low at the start of a transaction, held low for the duration of the transfer, and then driven high again after the transaction is completed. During a transaction, the master toggles the clock (SCLK). The SCLK polarity is defined as having an idle value that is low, and an active phase that is high (CPOL = 0). Serial input and output data is captured on the clock's rising edge and propagated on the falling edge FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 20. **11.4 SPI read operations with 3-wire mode** FXLS8962AF can be configured to operate in 3-wire software enabled SPI mode. In this mode, the SPI_MISO pin is left unconnected and the SPI_MOSI pin becomes a bidirectional input/output pin (SPI_DATA). Read operations in 3-wire mode are the same as write operations in 3- and 4-wire modes, except that at the end of the address cycle (falling edge of clock pulse 16), the SPI_DATA pin automatically switches from an input to an output and with bit D7 as the current output state. FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 22.

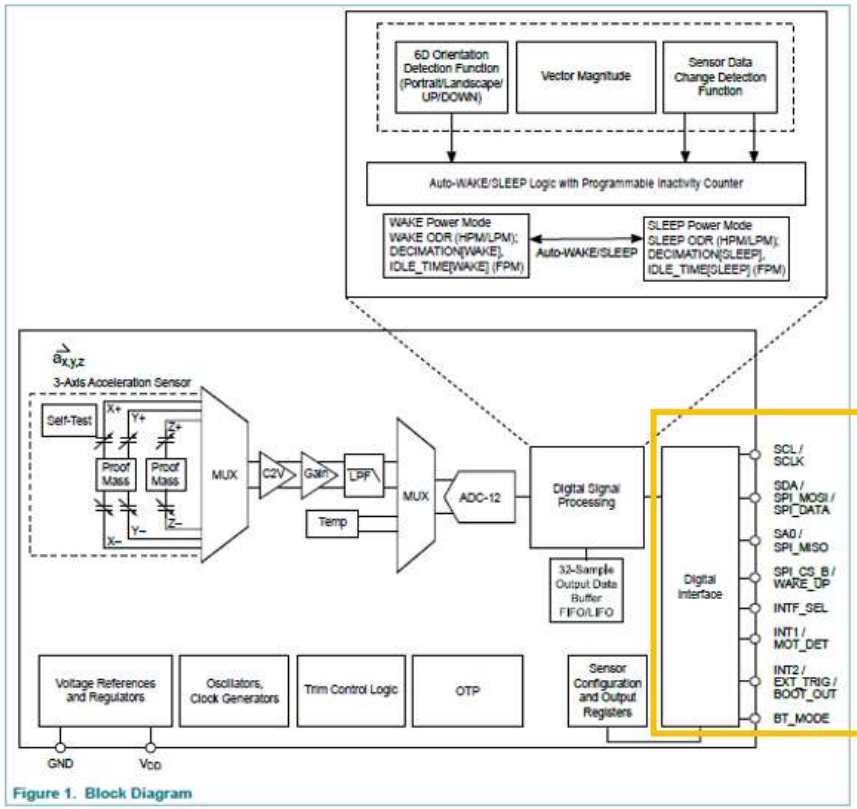
’474 Patent Claim	Representative NXP Product(s)
	NXP Semiconductors **FXOS8700CQ** 6-axis sensor with integrated linear accelerometer and magnetometer **5. Block diagram**  Fig 1. Block diagram FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 3.

’474 Patent Claim	Representative NXP Product(s)
	10.1.2 I²C read/write operations **Single-byte read** The master (or MCU) transmits a start condition (ST) to the FXOS8700CQ, followed by the slave address, with the R/W bit set to “0” for a write, and the FXOS8700CQ sends an acknowledgement. Then the master (or MCU) transmits the address of the register to read and the FXOS8700CQ sends an acknowledgement. The master (or MCU) transmits a repeated start condition (SR), followed by the slave address with the R/W bit set to “1” for a read from the previously selected register. The FXOS8700CQ then acknowledges and transmits the data from the requested register. The master does not acknowledge (NAK) the transmitted data, but transmits a stop condition to end the data transfer. FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 15. **10.1.1 General I²C operation** There are two signals associated with the I²C bus: the serial clock line (SCL) and the serial data line (SDA). The latter is a bidirectional line used for sending and receiving the data to/from the interface. External pullup resistors connected to VDDIO are required for SDA and SCL. When the bus is free both the lines are high. The I²C interface is compliant with fast mode (400 kHz), and normal mode (100 kHz) I²C standards. Operation at frequencies higher than 400 kHz is possible, but depends on several factors including the pullup resistor values, and total bus capacitance (trace + device capacitance). See Table 11 for more information. FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 14.

’474 Patent Claim	Representative NXP Product(s)
	10.2.1 General SPI operation **NOTE** FXOS8700CQ only supports a point-to-point SPI protocol, with only one master (MCU) and one slave device (FXOS8700CQ) present on the bus. FXOS8700CQ does not tri-state the MISO pin when the CS_B pin is deasserted (logic high), which can lead to a bus conflict if multiple slave devices are present on the bus. Do not connect more than one master and one slave device on the SPI bus. The CS_B pin is driven low at the start of a SPI transaction, held low for the duration of the transaction, and driven high after the transaction is complete. During a transaction the master toggles the SPI clock (SCLK) and transmits data on the MOSI pin. FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 17.

’474 Patent Claim	Representative NXP Product(s)
	NXP Semiconductors **FXPS7115D4** Digital absolute pressure sensor, 40 kPa to 115 kPa **5 Block diagram**  **Figure 1. Block diagram of FXPS7115D4** FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 3.

’474 Patent Claim	Representative NXP Product(s)
	7.4.1 I^2C bit transmissions The state of SDA when SCL is high determines the bit value being transmitted. SDA must be stable when SCL is high and change when SCL is low as shown in Figure 14. After the START signal has been transmitted by the master, the bus is considered busy. Timing for the start condition is specified in Table 105. FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 13. **7.5 Standard 32-bit SPI protocol** The device includes a standard SPI protocol requiring 32-bit data packets. The device is a slave device and requires that the base clock value be low (CPOL = 0) with data captured on the rising edge of the clock and data propagated on the falling edge of the clock (CPHA = 0). The most significant bit is transferred first (MSB first). SPI transfers are completed through a sequence of two phases. During the first phase, the command is transmitted from the SPI master to the device. During the second phase, response data is transmitted from the slave device. MOSI and SCLK transitions are ignored when SS_B is not asserted. FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 17.
[1c.] a processor having a slave serial interface coupled to the master serial interface through a clock signal line and a data signal line	The Accused ’474 Sensors each includes a processor having a slave serial interface coupled to the master serial interface through a clock signal line and a data signal line. For example, the Accused ’474 Sensors each includes a processor having a slave serial interface (*e.g.*, SPI/I2C) coupled to the master serial interface identified above through a clock signal line (*e.g.*, the serial clock line (SCL) and/or the SPI clock (SCLK)) and a data signal line (*e.g.*, the serial data line (SDA) and/or the SPI master serial data out slave serial data in (MOSI)). *See, e.g.*,

’474 Patent Claim	Representative NXP Product(s)
	NXP Semiconductors **FXLS8962AF** 3-Axis Low-g Accelerometer **5 Block diagram**  Figure 1. Block Diagram FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 3.

’474 Patent Claim	Representative NXP Product(s)

10.1.2 I^2C read/write operations

10.1.2.1 Single byte read

The master transmits a start condition (ST) to FXLS8962AF, followed by the slave address, with the R/W bit set to '0' for a write, and the FXLS8962AF sends an acknowledgement. Then the master transmits the address of the register to read and the FXLS8962AF sends an acknowledgement. The master transmits a repeated start condition (SR), followed by the slave address with the R/W bit set to '1' for a read from the previously selected register. The FXLS8962AF then acknowledges and transmits the data from the requested register. The master does not acknowledge (NAK) the transmitted data and then transmits a stop condition to end the data transfer.

FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 18.

11 SPI interface

The SPI interface is a classical Master/Slave serial port. FXLS8962AF is always considered to be the slave device and thus never initiates communication with the host processor.

The SPI interface of FXLS8962AF is compatible with interface mode 00, corresponding to CPOL = 0 and CPHA = 0.

For CPOL = 0, the idle value of the clock is zero, and the active value of the clock is 1. For CPHA = 0, data is captured on the clock's rising edge (low to high transition) and data is propagated on the clock's falling edge (high to low transition).

FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 20.

10.1.1 General I^2C operation

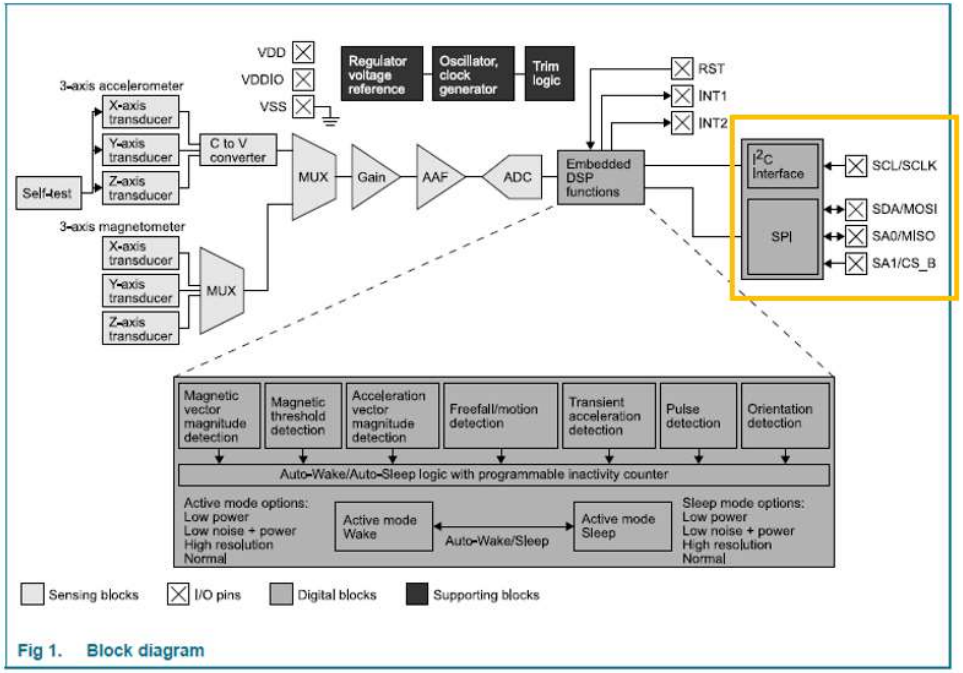
There are two signals associated with the I^2C-bus: the Serial Clock Line (SCL) and the Serial Data line (SDA). SDA is a bidirectional signal used for sending and receiving the data to/from the interface. External pull-up resistors connected to V_{DD} are required for SDA and SCL. When the I^2C-bus is free, SCL and SDA are high.

’474 Patent Claim	Representative NXP Product(s)
	FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 18. **11.1 General SPI operation** The SPI_CS_B pin is driven low at the start of a transaction, held low for the duration of the transfer, and then driven high again after the transaction is completed. During a transaction, the master toggles the clock (SCLK). The SCLK polarity is defined as having an idle value that is low, and an active phase that is high (CPOL = 0). Serial input and output data is captured on the clock's rising edge and propagated on the falling edge FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 20. **11.4 SPI read operations with 3-wire mode** FXLS8962AF can be configured to operate in 3-wire software enabled SPI mode. In this mode, the SPI_MISO pin is left unconnected and the SPI_MOSI pin becomes a bidirectional input/output pin (SPI_DATA). Read operations in 3-wire mode are the same as write operations in 3- and 4-wire modes, except that at the end of the address cycle (falling edge of clock pulse 16), the SPI_DATA pin automatically switches from an input to an output and with bit D7 as the current output state. FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 22.

’474 Patent Claim	Representative NXP Product(s)

SDA / SPI_MOSI / SPI_DATA	4	Mode dependent Multifunction serial interface pin.[2] INTF_SEL = V_{DD}: • SPI_MOSI: In 4-wire SPI mode this pin functions as the serial data input (Master Out Slave In). • SPI_DATA[3]: In 3-wire SPI mode this pin functions as the bidirectional serial data input/output. INTF_SEL = GND: • SDA: This pin functions as the I^2C Serial Data input/output.
SCL / SCLK	5	Mode dependent Multifunction serial interface pin.[2] INTF_SEL = V_{DD}: • SPI serial clock input (3- and 4-wire modes) INTF_SEL = GND: • I^2C serial clock input

FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 4.

’474 Patent Claim	Representative NXP Product(s)
	NXP Semiconductors **FXOS8700CQ** 6-axis sensor with integrated linear accelerometer and magnetometer **5. Block diagram**  Fig 1. Block diagram FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 3.

’474 Patent Claim	Representative NXP Product(s)
	10.1.2 I²C read/write operations **Single-byte read** The master (or MCU) transmits a start condition (ST) to the FXOS8700CQ, followed by the slave address, with the R/W bit set to “0” for a write, and the FXOS8700CQ sends an acknowledgement. Then the master (or MCU) transmits the address of the register to read and the FXOS8700CQ sends an acknowledgement. The master (or MCU) transmits a repeated start condition (SR), followed by the slave address with the R/W bit set to “1” for a read from the previously selected register. The FXOS8700CQ then acknowledges and transmits the data from the requested register. The master does not acknowledge (NAK) the transmitted data, but transmits a stop condition to end the data transfer. FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 15. **10.1.1 General I²C operation** There are two signals associated with the I²C bus: the serial clock line (SCL) and the serial data line (SDA). The latter is a bidirectional line used for sending and receiving the data to/from the interface. External pullup resistors connected to VDDIO are required for SDA and SCL. When the bus is free both the lines are high. The I²C interface is compliant with fast mode (400 kHz), and normal mode (100 kHz) I²C standards. Operation at frequencies higher than 400 kHz is possible, but depends on several factors including the pullup resistor values, and total bus capacitance (trace + device capacitance). See Table 11 for more information. FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 14.

’474 Patent Claim	Representative NXP Product(s)

10.2.1 General SPI operation

NOTE

FXOS8700CQ only supports a point-to-point SPI protocol, with only one master (MCU) and one slave device (FXOS8700CQ) present on the bus. FXOS8700CQ does not tri-state the MISO pin when the CS_B pin is deasserted (logic high), which can lead to a bus conflict if multiple slave devices are present on the bus.

Do not connect more than one master and one slave device on the SPI bus.

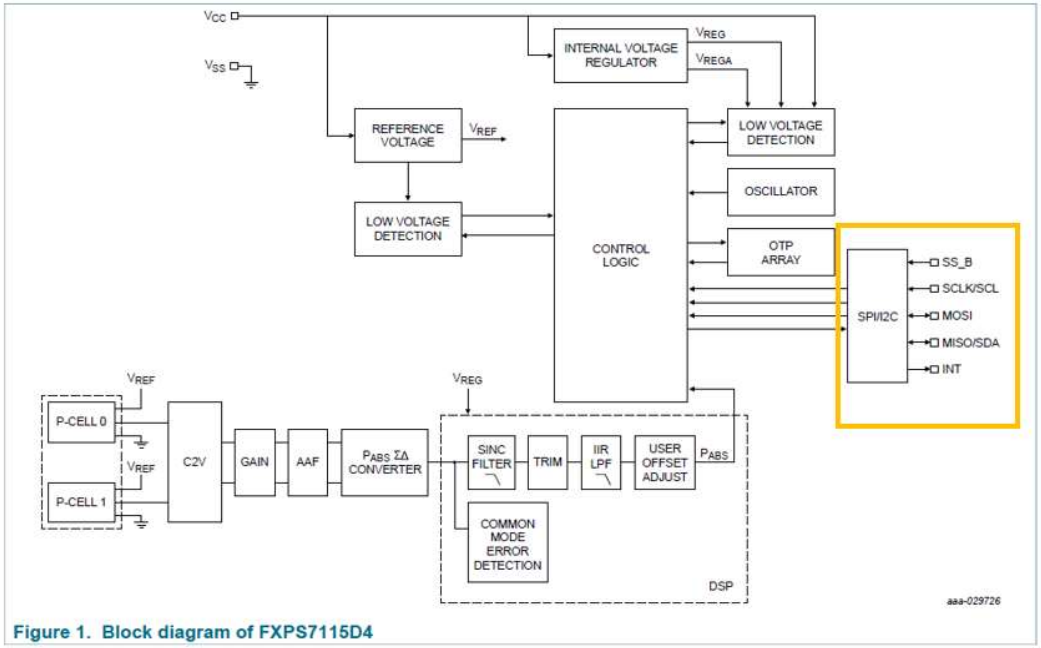
The CS_B pin is driven low at the start of a SPI transaction, held low for the duration of the transaction, and driven high after the transaction is complete. During a transaction the master toggles the SPI clock (SCLK) and transmits data on the MOSI pin.

FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 17.

Table 13. Serial interface pin descriptions

Pin name	Pin description
VDDIO	Digital interface power
SA1/CS_B	I^2C second least significant bit of device address/SPI chip select
SCL/SCLK	I^2C/SPI serial clock
SDA/MOSI	I^2C serial data/SPI master serial data out slave serial data in
SA0/MISO	I^2C least significant bit of the device address/SPI master serial data in slave out

FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 18.

’474 Patent Claim	Representative NXP Product(s)
	NXP Semiconductors **FXPS7115D4** Digital absolute pressure sensor, 40 kPa to 115 kPa **5 Block diagram**  Figure 1. Block diagram of FXPS7115D4 FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 3.

’474 Patent Claim	Representative NXP Product(s)

7.4.1 I^2C bit transmissions

The state of SDA when SCL is high determines the bit value being transmitted. SDA must be stable when SCL is high and change when SCL is low as shown in Figure 14. After the START signal has been transmitted by the master, the bus is considered busy. Timing for the start condition is specified in Table 105.

FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 13.

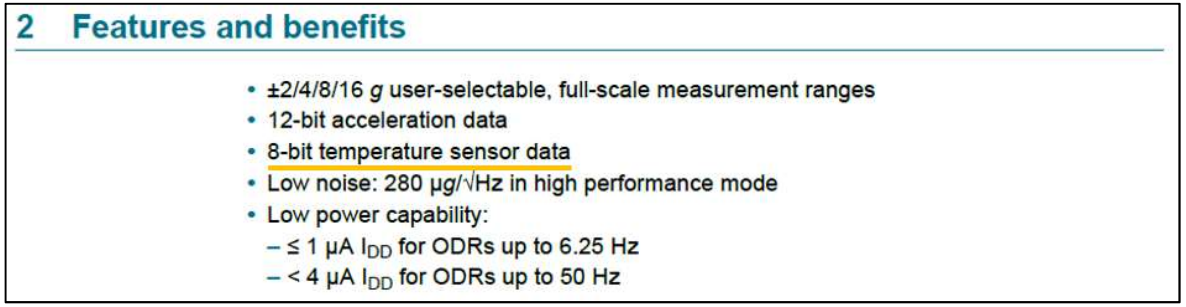
7.5 Standard 32-bit SPI protocol

The device includes a standard SPI protocol requiring 32-bit data packets. The device is a slave device and requires that the base clock value be low (CPOL = 0) with data captured on the rising edge of the clock and data propagated on the falling edge of the clock (CPHA = 0). The most significant bit is transferred first (MSB first). SPI transfers are completed through a sequence of two phases. During the first phase, the command is transmitted from the SPI master to the device. During the second phase, response data is transmitted from the slave device. MOSI and SCLK transitions are ignored when SS_B is not asserted.

FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 17.

9	SCLK/SCL	In I^2C mode, input pin 9 provides the serial clock. This pin must be connected to V_{CC} with an external pull-up resistor, as shown in the application diagram. In SPI mode, input pin 9 provides the serial clock. An internal pull-down device is connected to this pin.
10	MOSI	SPI data in In SPI mode, pin 10 functions as the serial data input to the SPI port. An internal pull-down device is connected to this pin.

FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 4.

’474 Patent Claim	Representative NXP Product(s)
[1d.] wherein the slave serial interface is responsive to a read temperature command issued by the master serial interface to return a temperature value associated with the processor.	In each of the Accused ’474 Sensors, the slave serial interface is responsive to a read temperature command issued by the master serial interface to return a temperature value associated with the processor. For example, in each of the Accused ’474 Sensors, the slave serial interface identified above is responsive to a read temperature command (*e.g.*, the read temperature command directed to the TEMP_OUT register) issued by the identified master serial interface to return a temperature value (*e.g.*, the temperature value in the TEMP_OUT register) associated with the processor. *See, e.g.*,  2 Features and benefits • ±2/4/8/16 *g* user-selectable, full-scale measurement ranges • 12-bit acceleration data • 8-bit temperature sensor data • Low noise: 280 µ*g*/√Hz in high performance mode • Low power capability: – ≤ 1 µA I_{DD} for ODRs up to 6.25 Hz – < 4 µA I_{DD} for ODRs up to 50 Hz FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 1.

’474 Patent Claim	Representative NXP Product(s)

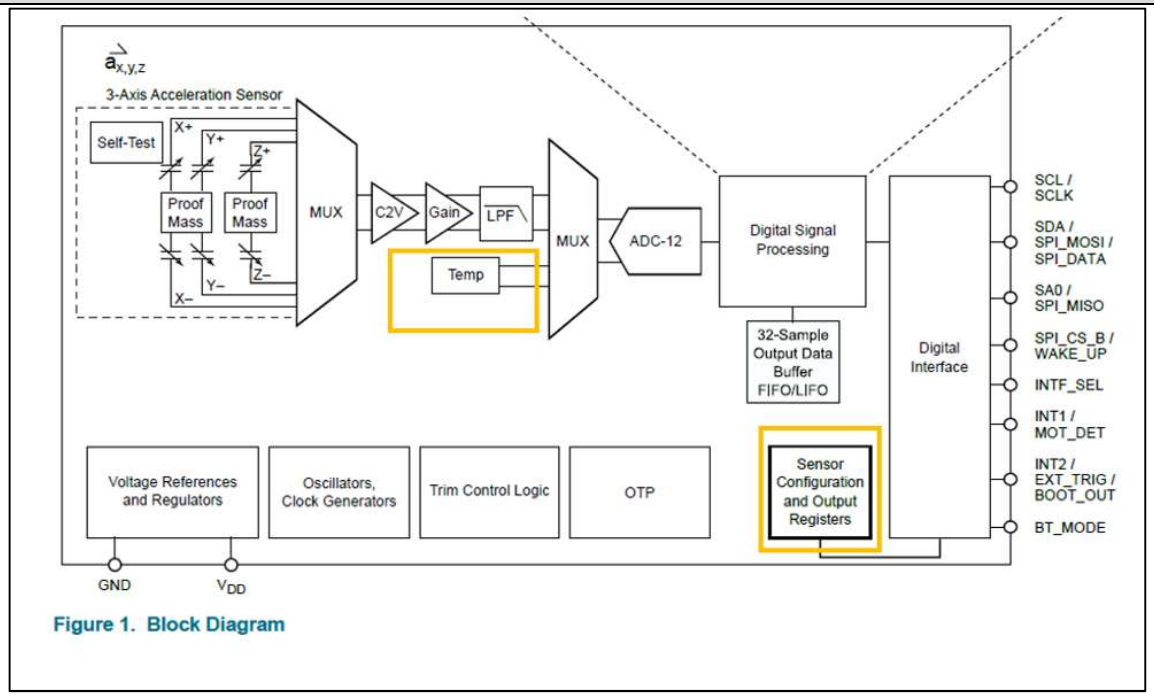


Figure 1. Block Diagram

FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 3.

13.2 TEMP_OUT register (address 01h)

Table 22. TEMP_OUT register (address 01h) bit allocation

Bit	7	6	5	4	3	2	1	0
Name	TEMP_OUT[7:0]							
Reset	—	—	—	—	—	—	—	—
Access	R	R	R	R	R	R	R	R

The **TEMP_OUT** register contains the 8-bit, 2's complement temperature value. When this register contains the value 00h, the measured temperature is 25 °C (typ). This register is updated on every ODR cycle.

FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 3.

’474 Patent Claim	Representative NXP Product(s)

10 I^2C digital interface

The registers embedded within FXLS8962AF may be accessed using an I^2C interface when the INTF_SEL pin is tied to GND. If the V_{DD} supply is not present, the device will be in shutdown mode and any communications on the interface are ignored. When the device is on a common I^2C-bus with other slave devices, the V_{DD} supply pin must be left unconnected (high-impedance) when the device supply is turned off to ensure that the internal ESD protection diodes do not become forward biased and prevent the bus from functioning normally (clamping).

FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 16.

10.1.2 I^2C read/write operations

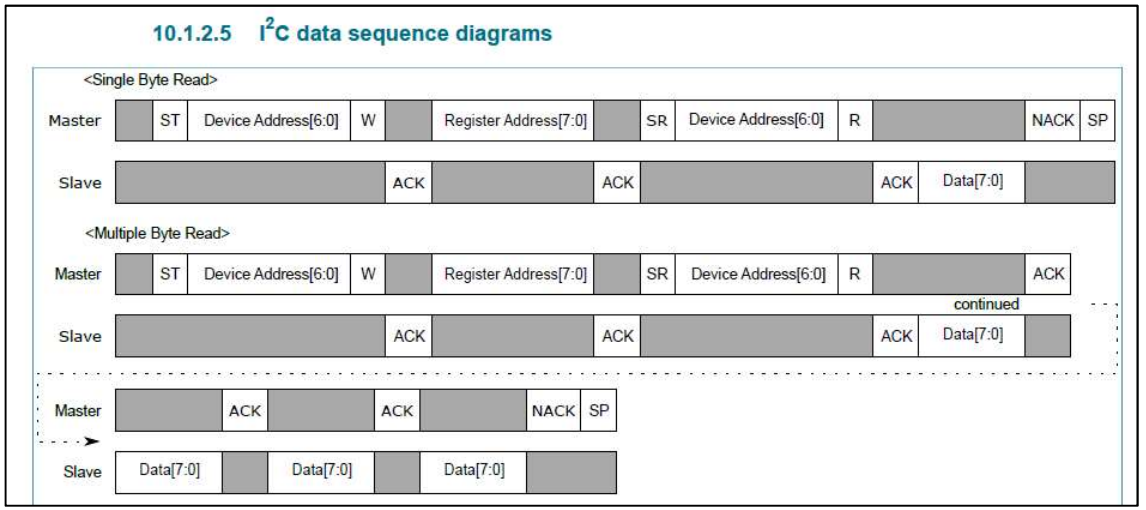
10.1.2.1 Single byte read

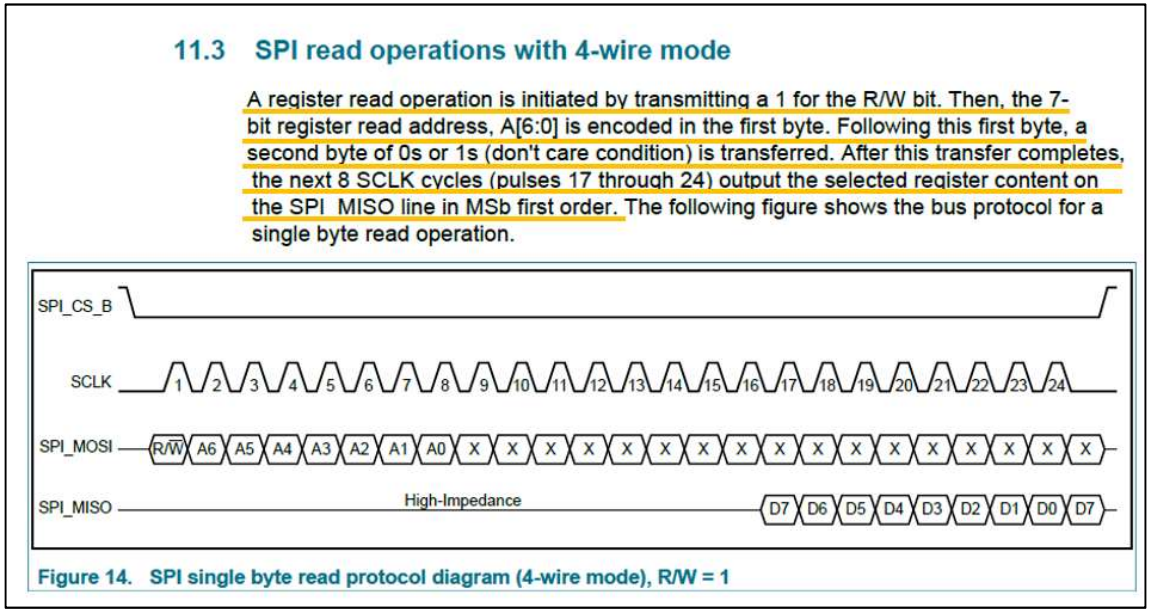
The master transmits a start condition (ST) to FXLS8962AF, followed by the slave address, with the R/W bit set to '0' for a write, and the FXLS8962AF sends an acknowledgement. Then the master transmits the address of the register to read and the FXLS8962AF sends an acknowledgement. The master transmits a repeated start condition (SR), followed by the slave address with the R/W bit set to '1' for a read from the previously selected register. The FXLS8962AF then acknowledges and transmits the data from the requested register. The master does not acknowledge (NAK) the transmitted data and then transmits a stop condition to end the data transfer.

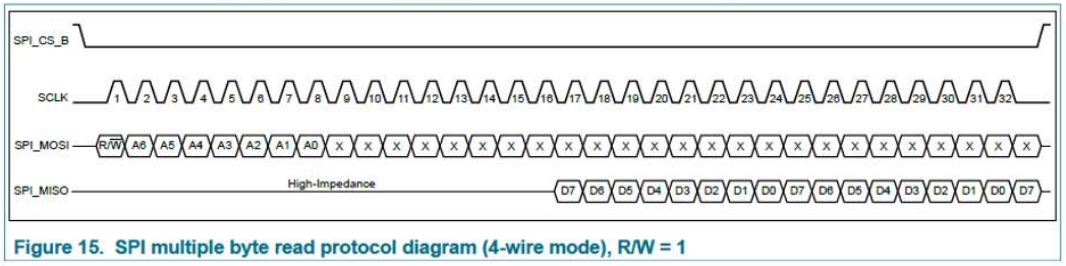
10.1.2.2 Multiple byte read

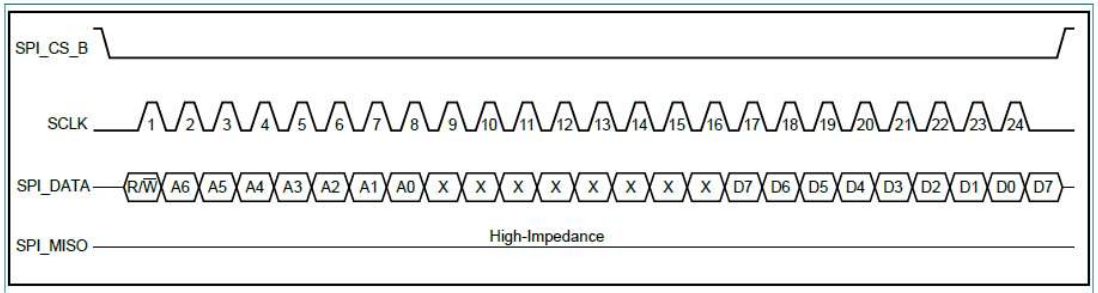
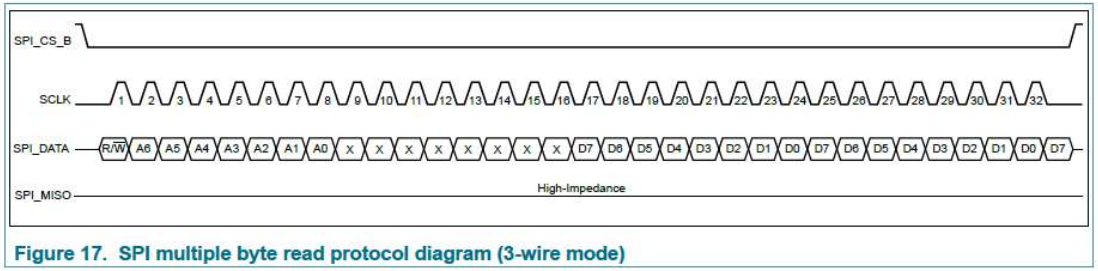
When performing a multi-byte or *burst* read, FXLS8962AF automatically increments the register read address pointer after a read command is received. Therefore, after following the steps of a single-byte read, multiple bytes of data can be read from sequential register addresses after each FXLS8962AF acknowledgment (ACK) is received until a no acknowledge (NAK) is issued by the master followed by a stop condition (SP) signaling the end of the transfer.

FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 18-19.

’474 Patent Claim	Representative NXP Product(s)
	 FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 18-20. **11.1 General SPI operation** The SPI_CS_B pin is driven low at the start of a transaction, held low for the duration of the transfer, and then driven high again after the transaction is completed. During a transaction, the master toggles the clock (SCLK). The SCLK polarity is defined as having an idle value that is low, and an active phase that is high (CPOL = 0). Serial input and output data is captured on the clock's rising edge and propagated on the falling edge (CPHA = 0). Single byte read and single byte write operations are completed in 24 SCLK cycles; multiple byte reads and writes are completed in additional multiples of 8 SCLK cycles per read or written byte. The first SCLK cycle latches the R/W (Read/Write) bit to select whether the desired operation is a read (R/W = 1) or a write (R/W = 0). The following seven SCLK cycles are used to latch the register read or write address. ***Note:*** *4-wire SPI interface mode is the default out of POR or after a soft reset. The 3-wire interface mode may also be selected by setting* ***SENS_CONFIG1[****SPI_M****]*** *= 1.*

’474 Patent Claim	Representative NXP Product(s)
	FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 18-20-21.  FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 18-21-22.

’474 Patent Claim	Representative NXP Product(s)
	Multiple-byte read operations are performed similarly to single-byte reads with additional bytes read out in multiples of eight SCLK cycles. The register read address is auto-incremented by FXLS8962AF so that every eighth clock edge will latch the address of the next sequential register read address. When the desired number of bytes has been read, a rising edge on SPI_CS_B terminates the transaction.  Figure 15. SPI multiple byte read protocol diagram (4-wire mode), R/W = 1 FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 18-22. 11.4 SPI read operations with 3-wire mode FXLS8962AF can be configured to operate in 3-wire software enabled SPI mode. In this mode, the SPI_MISO pin is left unconnected and the SPI_MOSI pin becomes a bidirectional input/output pin (SPI_DATA). Read operations in 3-wire mode are the same as write operations in 3- and 4-wire modes, except that at the end of the address cycle (falling edge of clock pulse 16), the SPI_DATA pin automatically switches from an input to an output and with bit D7 as the current output state. FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 18-22.

’474 Patent Claim	Representative NXP Product(s)
	 **Figure 16. SPI single byte read protocol diagram (3-wire mode)**  **Figure 17. SPI multiple byte read protocol diagram (3-wire mode)** FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 18-23.

’474 Patent Claim	Representative NXP Product(s)

14.3 Temperature register

14.3.1 TEMP register (address 0x51)

Eight-bit 2's complement sensor temperature value with 0.96 °C/LSB sensitivity. Temperature data is only valid between –40 °C and 125 °C. The temperature sensor output is only valid when M_CTRL_REG1[m_hms] > 0b00. Please note that the temperature sensor is uncalibrated and its output for a given temperature will vary from one device to the next.

Table 50. TEMP register (address 0x51) bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	die_temperature[7:0]							
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

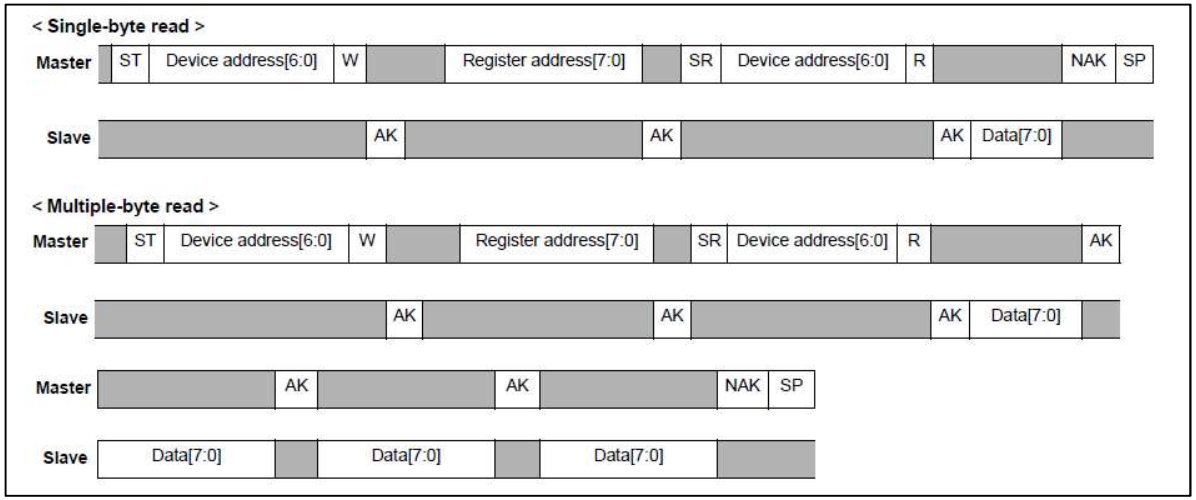
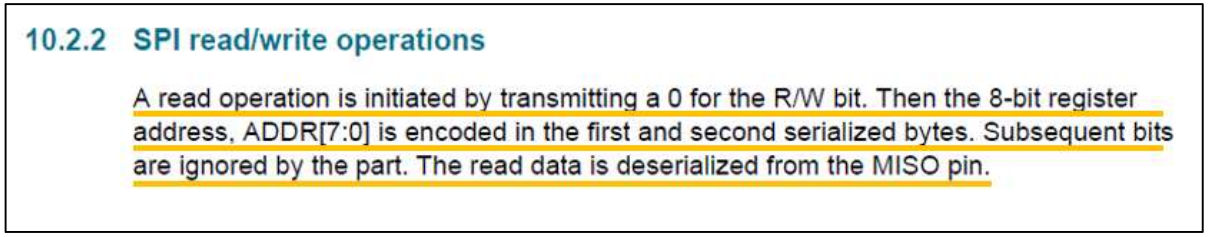
FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 52.

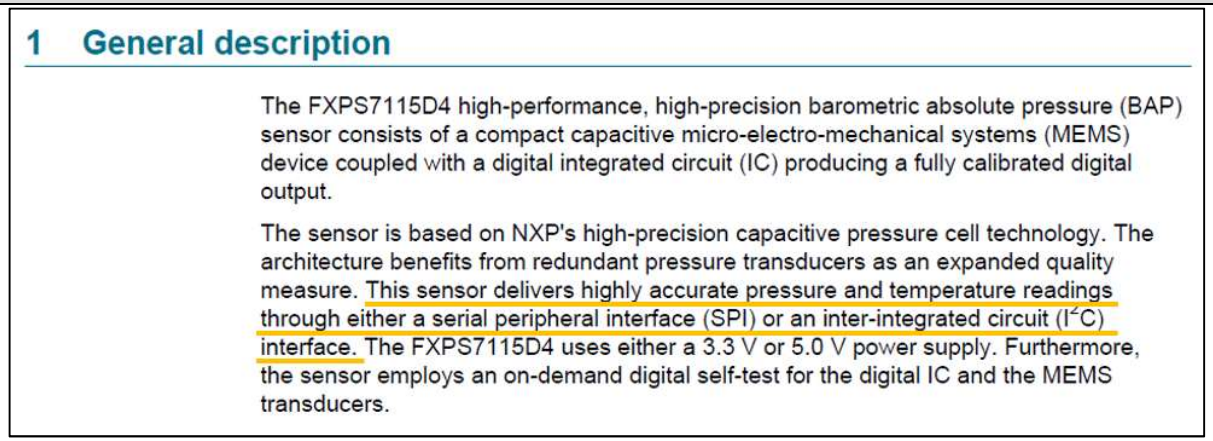
10.1.2 I²C read/write operations

Single-byte read

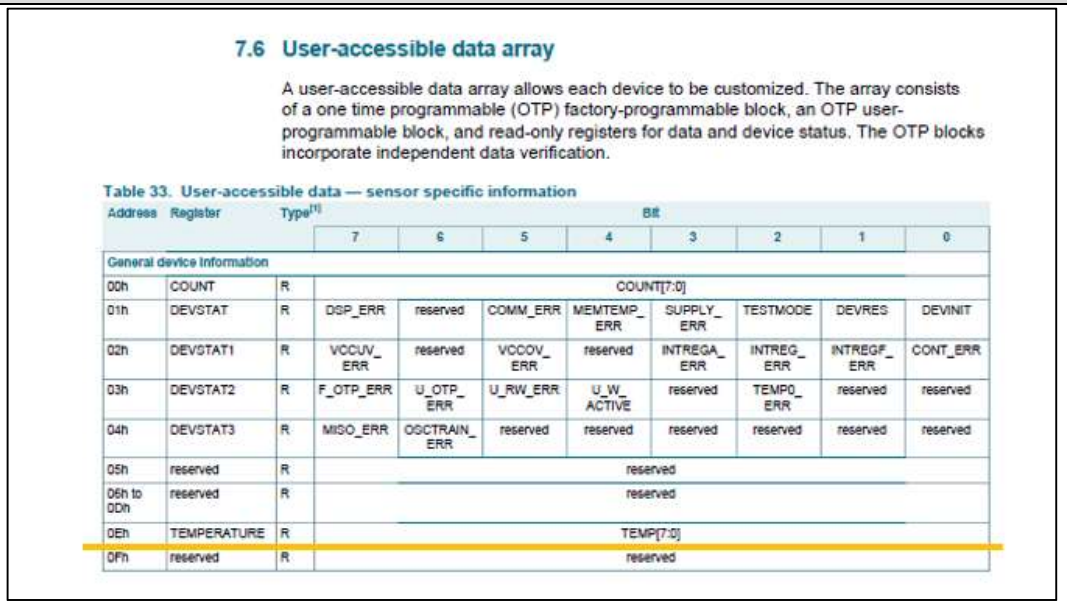
The master (or MCU) transmits a start condition (ST) to the FXOS8700CQ, followed by the slave address, with the R/W bit set to “0” for a write, and the FXOS8700CQ sends an acknowledgement. Then the master (or MCU) transmits the address of the register to read and the FXOS8700CQ sends an acknowledgement. The master (or MCU) transmits a repeated start condition (SR), followed by the slave address with the R/W bit set to “1” for a read from the previously selected register. The FXOS8700CQ then acknowledges and transmits the data from the requested register. The master does not acknowledge (NAK) the transmitted data, but transmits a stop condition to end the data transfer.

When performing a multi-byte or “burst” read, the FXOS8700CQ automatically increments the register address read pointer after a read command is received. Therefore, after following the steps of a single-byte read, multiple bytes of data can be read from sequential registers after each FXOS8700CQ acknowledgment (AK) is received until a no acknowledge (NAK) occurs from the master followed by a stop condition (SP) signaling an end of transmission.

’474 Patent Claim	Representative NXP Product(s)
	FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 15.  FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 16.  10.2.2 SPI read/write operations A read operation is initiated by transmitting a 0 for the R/W bit. Then the 8-bit register address, ADDR[7:0] is encoded in the first and second serialized bytes. Subsequent bits are ignored by the part. The read data is deserialized from the MISO pin. FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 18.

’474 Patent Claim	Representative NXP Product(s)
	 1 General description The FXPS7115D4 high-performance, high-precision barometric absolute pressure (BAP) sensor consists of a compact capacitive micro-electro-mechanical systems (MEMS) device coupled with a digital integrated circuit (IC) producing a fully calibrated digital output. The sensor is based on NXP's high-precision capacitive pressure cell technology. The architecture benefits from redundant pressure transducers as an expanded quality measure. This sensor delivers highly accurate pressure and temperature readings through either a serial peripheral interface (SPI) or an inter-integrated circuit (I^2C) interface. The FXPS7115D4 uses either a 3.3 V or 5.0 V power supply. Furthermore, the sensor employs an on-demand digital self-test for the digital IC and the MEMS transducers. FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 1.

’474 Patent Claim	Representative NXP Product(s)
	7.3.4 Temperature sensor 7.3.4.1 Temperature sensor signal chain The device includes a temperature sensor for signal compensation and user readability. Figure 12 shows a simplified block diagram. Temperature sensor parameters are specified in Table 104 and Table 105.  Figure 12. Temperature sensor signal chain block diagram 7.3.4.2 Temperature sensor output scaling equation Equation 5 is used to convert temperature readings with the variables specified in Table 8. $T_{DEGC} = \frac{T_{LSB} - T0_{LSB}}{T_{SENSE}}$ (5) where: T_{DEGC} = The temperature output in degrees C T_{LSB} = The temperature output in LSB $T0_{LSB}$ = The expected temperature output in LSB at 0 °C T_{SENSE} = The expected temperature sensitivity in LSB/°C Table 8. Temperature conversion variables Data reading | $T0_{LSB}$ (LSB) | T_{SENSE} LSB/C) 8-bit register read | 68 | 1 FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 12.

’474 Patent Claim	Representative NXP Product(s)
	

7.6 User-accessible data array

A user-accessible data array allows each device to be customized. The array consists of a one time programmable (OTP) factory-programmable block, an OTP user-programmable block, and read-only registers for data and device status. The OTP blocks incorporate independent data verification.

Table 33. User-accessible data — sensor specific information

Address	Register	Type[1]	Bit							
			7	6	5	4	3	2	1	0
General device information										
00h	COUNT	R	COUNT[7:0]							
01h	DEVSTAT	R	DSP_ERR	reserved	COMM_ERR	MEMTEMP_ERR	SUPPLY_ERR	TESTMODE	DEVRES	DEVINIT
02h	DEVSTAT1	R	VCCUV_ERR	reserved	VCCOV_ERR	reserved	INTREGA_ERR	INTREG_ERR	INTREGF_ERR	CONT_ERR
03h	DEVSTAT2	R	F_OTP_ERR	U_OTP_ERR	U_RW_ERR	U_W_ACTIVE	reserved	TEMP0_ERR	reserved	reserved
04h	DEVSTAT3	R	MISO_ERR	OSCTRAIN_ERR	reserved	reserved	reserved	reserved	reserved	reserved
05h	reserved	R	reserved							
06h to 0Dh	reserved	R	reserved							
0Eh	TEMPERATURE	R	TEMP[7:0]							
0Fh	reserved	R	reserved							

FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 27.

’474 Patent Claim	Representative NXP Product(s)
	7.4.6.2 Register read transfers The device supports I^2C register read data transfers. Register read data transfers are constructed as follows: 1. The master transmits a START condition. 2. The master transmits the 7-bit slave address. 3. The master transmits a '0' for the read/write bit to indicate a write operation. 4. The slave transmits an ACK. 5. The master transmits the register address to be read. 6. The slave transmits an ACK. 7. The master transmits a repeat START condition. 8. The master transmits the 7-bit slave address. 9. The master transmits a '1' for the read/write bit to indicate a read operation. 10. The slave transmits an ACK. 11. The slave transmits the data from the register addressed. 12. The master transmits a NACK. 13. The master transmits a STOP condition.  FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 16.

’474 Patent Claim	Representative NXP Product(s)
	7.5.3 Command summary **7.5.3.1 Register read command** The device supports a register read command. The register read command uses the upper 7 bits of the addresses defined in Section 7.6 "User-accessible data array" to address 8-bit registers in the register map. The response to a register read command is shown in Section 7.5.3.1.2 "Register read response message format". The response is transmitted on the next SPI message if and only if all of the following conditions are met: • No SPI error is detected (see Section 7.5.5.3 "SPI error") • No MISO error is detected (see Section 7.5.5.4 "SPI data output verification error") If these conditions are met, the device responds to the register read request as shown in Section 7.5.3.1.2 "Register read response message format". Otherwise, the device responds with the error response as defined in Section 7.5.5.2 "Detailed status field". The register read response includes the register contents at the rising edge of SS_B for the register read command. FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 20.

’474 Patent Claim	Representative NXP Product(s)

7.5.3.1.1 Register read command message format

Table 13. Register read command message format

MSB: bit 31; LSB: bit 0

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
Register access command																															
Command C[3:0]				Fixed bits: must = 0h				Register address								Register data								8-bit CRC							
1	1	0	0	0	0	0	0	RA[7:1]							RA[0]	0	0	0	0	0	0	0	0	CRC[7:0]							

Table 14. Register read command message bit field descriptions

Bit field	Definition
C[3:0]	Register read command = '1100'
RA[7:0]	RA[7:1] contains the word address of the register to be read.
CRC[7:0]	Read CRC Section

7.5.3.1.2 Register read response message format

Table 15. Register read response message format

MSB: bit 31; LSB: bit 0

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
Register access command																															
Command C[0], [3:1]				Basic Status		Unused Data 0h		Register data: contents of RA[7:1] high byte								Register data: contents of RA[7:1] low byte								8-bit CRC							
0	1	1	0	ST[1:0]		0	0	RD[15:8]								RD[7:0]								CRC[7:0]							

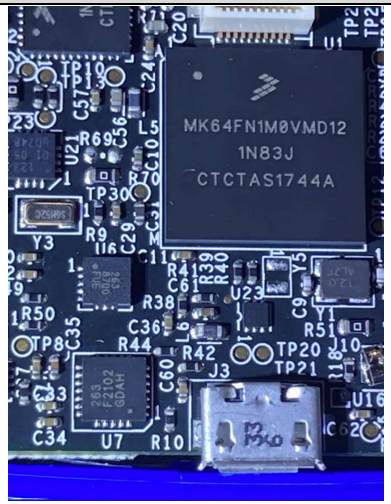
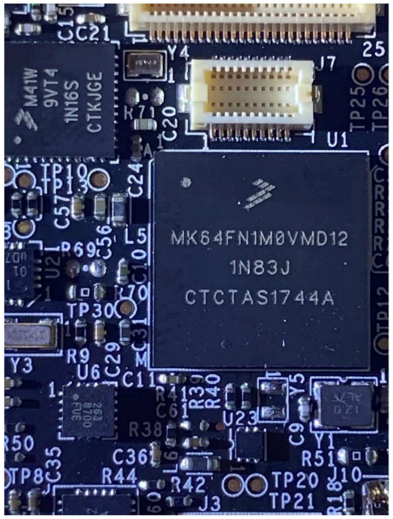
FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 20.

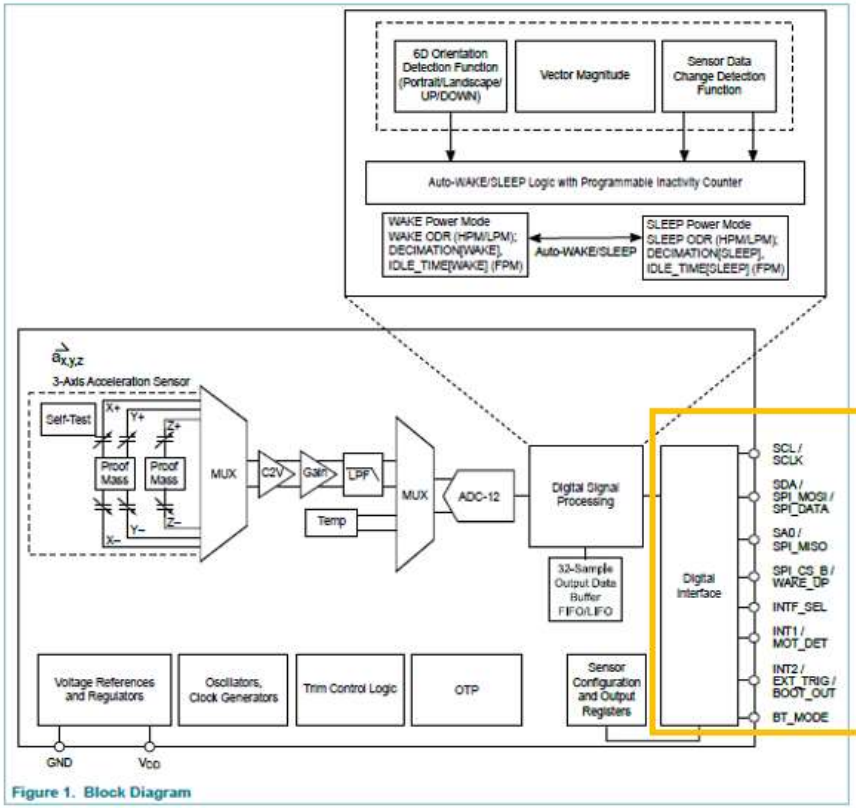
Table 16. Register read response message bit field descriptions

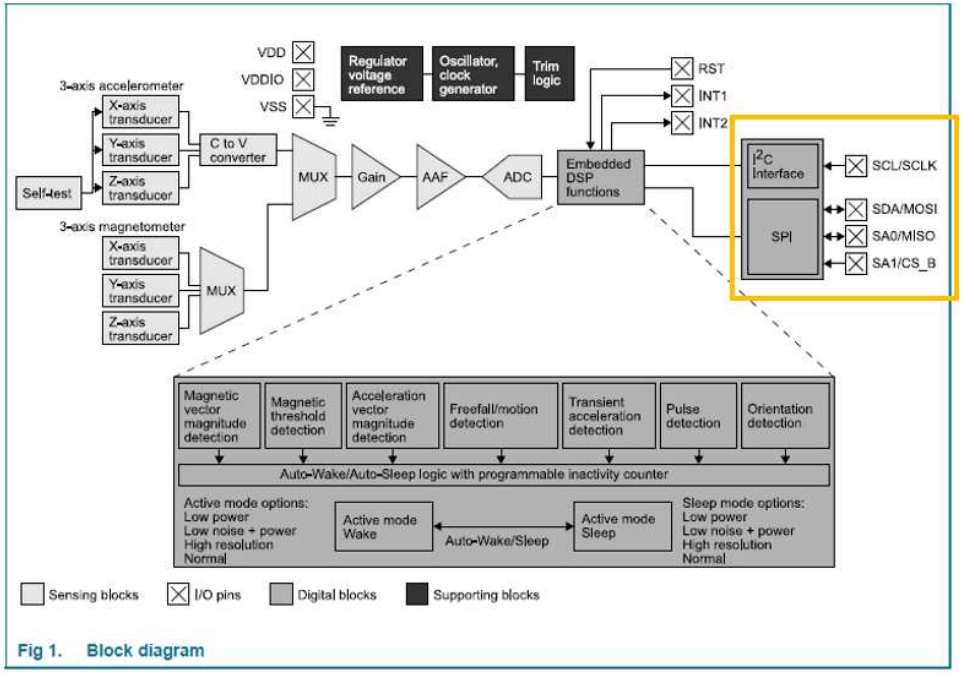
Bit field	Definition
C[0], [3:1]	Register Read Command = '0110'
ST[1:0]	Status
RD[15:8]	The contents of the register addressed by RA[7:1] high byte (RA[0] = 1)
RD[7:0]	The contents of the register addressed by RA[7:1] low byte (RA[0] = 0)
CRC[7:0]	8-bit CRC

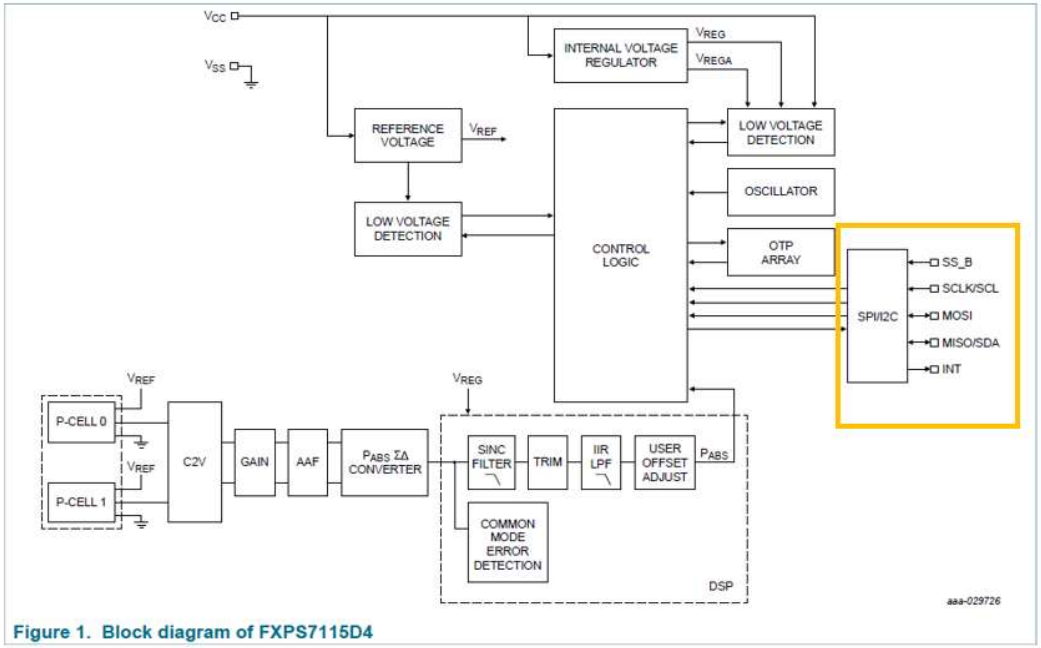
FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 21.

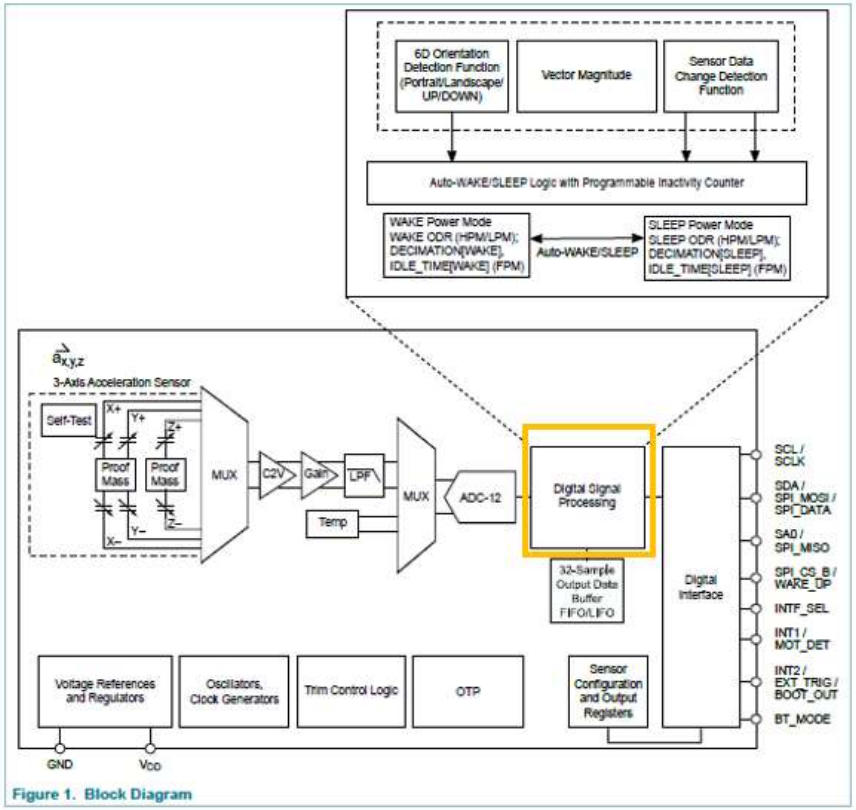
’474 Patent Claim	Representative NXP Product(s)
[8a.]. A method for communicating over a point to point serial communication system having a clock signal line and a data signal line coupling a serial interface master and a serial interface slave, the method comprising:	To the extent the preamble is limiting, the Accused ’474 Sensors perform a method for communicating over a point to point serial communication system having a clock signal line and a data signal line coupling a serial interface master and a serial interface slave. For example, each of the Accused ’474 Sensors perform a method for communicating over a point to point serial communication system (e.g., system identified below) having a clock signal line (*e.g.*, the serial clock line (SCL) and/or the SPI clock (SCLK)) and a data signal line (*e.g.*, the serial data line (SDA) and/or the SPI master serial data out slave serial data in (MOSI)) coupling a serial interface master (*e.g.*, the SPI/I2C interface on the master/MCU and/or the host processor) and a serial interface slave (*e.g.*, SPI/I2C). *See, e.g.*,  https://www.avnet.com/shop/us/products/nxp/fxos8700cqr1-3074457345626313537?fromPage=autoSuggest&langId=-1&autoSuggestSearchTerm=FXOS

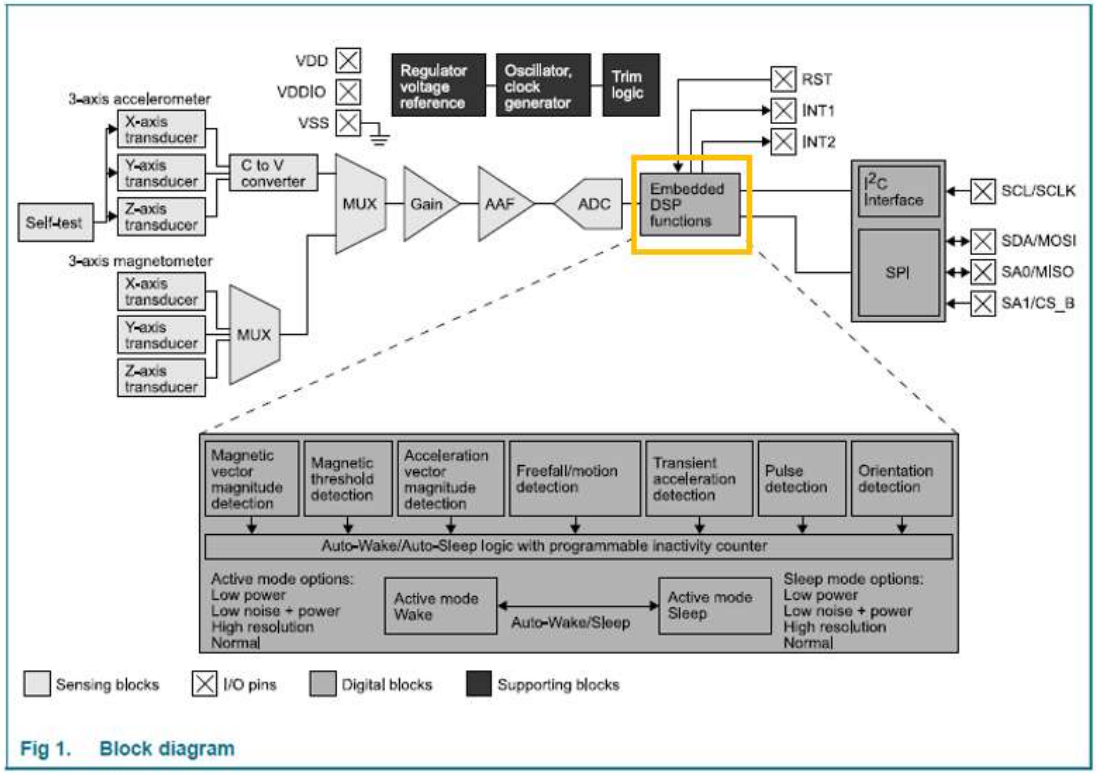
’474 Patent Claim	Representative NXP Product(s)
	 SLN-RPK-NODE (development board of FXOS8700CQ) (Arrow)  SLN-RPK-NODE (development board of FXOS8700CQ) (Mouser)

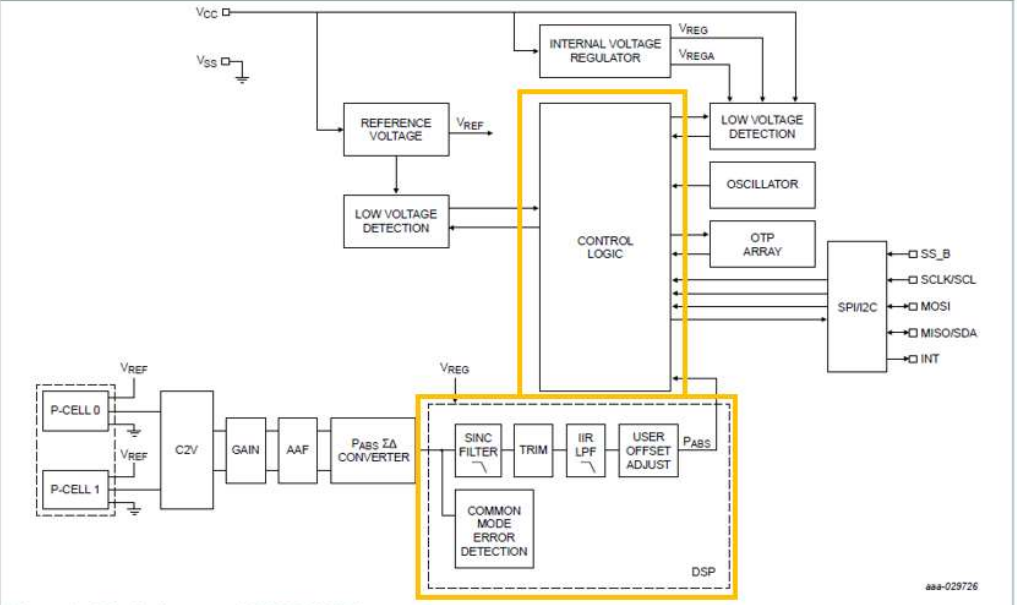
’474 Patent Claim	Representative NXP Product(s)
	NXP Semiconductors **FXLS8962AF** 3-Axis Low-g Accelerometer **5 Block diagram**  Figure 1. Block Diagram FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 3.

’474 Patent Claim	Representative NXP Product(s)
	NXP Semiconductors **FXOS8700CQ** 6-axis sensor with integrated linear accelerometer and magnetometer **5. Block diagram**  Fig 1. Block diagram FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 3.

’474 Patent Claim	Representative NXP Product(s)
	NXP Semiconductors **FXPS7115D4** Digital absolute pressure sensor, 40 kPa to 115 kPa **5 Block diagram**  **Figure 1. Block diagram of FXPS7115D4** FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 3.

’474 Patent Claim	Representative NXP Product(s)
	NXP Semiconductors **FXLS8962AF** 3-Axis Low-g Accelerometer **5 Block diagram**  Figure 1. Block Diagram FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 3.

’474 Patent Claim	Representative NXP Product(s)
	NXP Semiconductors **FXOS8700CQ** 6-axis sensor with integrated linear accelerometer and magnetometer **5. Block diagram**  **Fig 1. Block diagram** FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 3.

’474 Patent Claim	Representative NXP Product(s)
	NXP Semiconductors **FXPS7115D4** Digital absolute pressure sensor, 40 kPa to 115 kPa **5 Block diagram**  Figure 1. Block diagram of FXPS7115D4 FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 3.

’474 Patent Claim	Representative NXP Product(s)

10.1.2 I^2C read/write operations

10.1.2.1 Single byte read

The master transmits a start condition (ST) to FXLS8962AF, followed by the slave address, with the R/W bit set to '0' for a write, and the FXLS8962AF sends an acknowledgement. Then the master transmits the address of the register to read and the FXLS8962AF sends an acknowledgement. The master transmits a repeated start condition (SR), followed by the slave address with the R/W bit set to '1' for a read from the previously selected register. The FXLS8962AF then acknowledges and transmits the data from the requested register. The master does not acknowledge (NAK) the transmitted data and then transmits a stop condition to end the data transfer.

FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 18.

11 SPI interface

The SPI interface is a classical Master/Slave serial port. FXLS8962AF is always considered to be the slave device and thus never initiates communication with the host processor.

The SPI interface of FXLS8962AF is compatible with interface mode 00, corresponding to CPOL = 0 and CPHA = 0.

For CPOL = 0, the idle value of the clock is zero, and the active value of the clock is 1. For CPHA = 0, data is captured on the clock's rising edge (low to high transition) and data is propagated on the clock's falling edge (high to low transition).

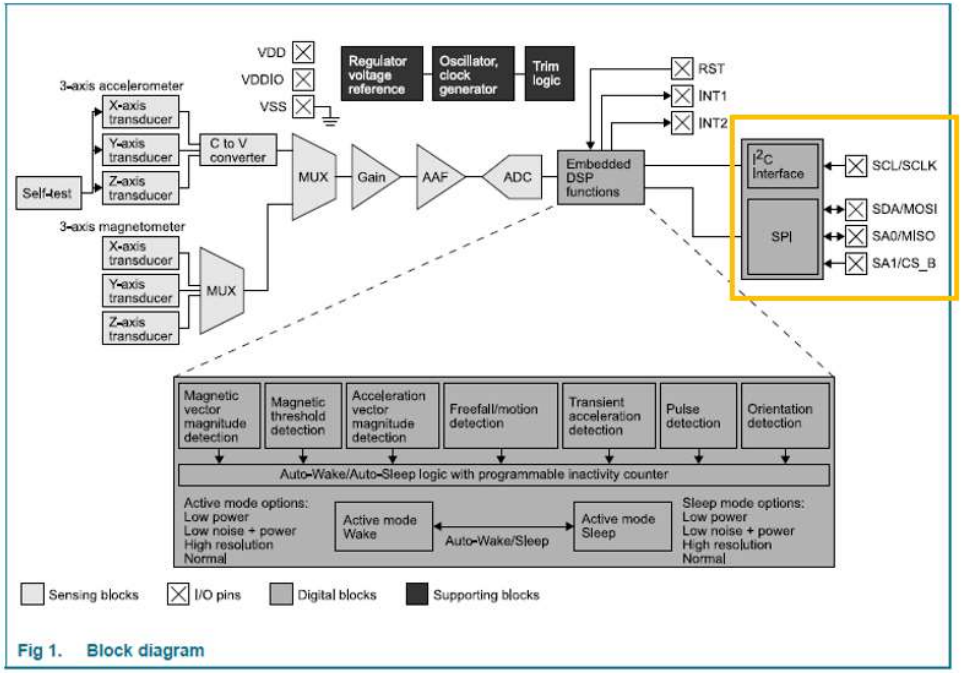
FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 20.

10.1.1 General I^2C operation

There are two signals associated with the I^2C-bus: the Serial Clock Line (SCL) and the Serial Data line (SDA). SDA is a bidirectional signal used for sending and receiving the data to/from the interface. External pull-up resistors connected to V_{DD} are required for SDA and SCL. When the I^2C-bus is free, SCL and SDA are high.

’474 Patent Claim	Representative NXP Product(s)
	FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 18. **11.1 General SPI operation** The SPI_CS_B pin is driven low at the start of a transaction, held low for the duration of the transfer, and then driven high again after the transaction is completed. During a transaction, the master toggles the clock (SCLK). The SCLK polarity is defined as having an idle value that is low, and an active phase that is high (CPOL = 0). Serial input and output data is captured on the clock's rising edge and propagated on the falling edge FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 20. **11.4 SPI read operations with 3-wire mode** FXLS8962AF can be configured to operate in 3-wire software enabled SPI mode. In this mode, the SPI_MISO pin is left unconnected and the SPI_MOSI pin becomes a bidirectional input/output pin (SPI_DATA). Read operations in 3-wire mode are the same as write operations in 3- and 4-wire modes, except that at the end of the address cycle (falling edge of clock pulse 16), the SPI_DATA pin automatically switches from an input to an output and with bit D7 as the current output state. FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 22.

’474 Patent Claim	Representative NXP Product(s)
	SDA / SPI_MOSI / SPI_DATA | 4 | Mode dependent Multifunction serial interface pin.[2] INTF_SEL = V_{DD}: • SPI_MOSI: In 4-wire SPI mode this pin functions as the serial data input (Master Out Slave In). • SPI_DATA[3]: In 3-wire SPI mode this pin functions as the bidirectional serial data input/output. INTF_SEL = GND: • SDA: This pin functions as the I^2C Serial Data input/output. SCL / SCLK | 5 | Mode dependent Multifunction serial interface pin.[2] INTF_SEL = V_{DD}: • SPI serial clock input (3- and 4-wire modes) INTF_SEL = GND: • I^2C serial clock input FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 4.

’474 Patent Claim	Representative NXP Product(s)
	NXP Semiconductors **FXOS8700CQ** 6-axis sensor with integrated linear accelerometer and magnetometer **5. Block diagram**  Fig 1. Block diagram FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 3.

’474 Patent Claim	Representative NXP Product(s)
	10.1.2 I²C read/write operations **Single-byte read** The master (or MCU) transmits a start condition (ST) to the FXOS8700CQ, followed by the slave address, with the R/W bit set to “0” for a write, and the FXOS8700CQ sends an acknowledgement. Then the master (or MCU) transmits the address of the register to read and the FXOS8700CQ sends an acknowledgement. The master (or MCU) transmits a repeated start condition (SR), followed by the slave address with the R/W bit set to “1” for a read from the previously selected register. The FXOS8700CQ then acknowledges and transmits the data from the requested register. The master does not acknowledge (NAK) the transmitted data, but transmits a stop condition to end the data transfer. FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 15. **10.1.1 General I²C operation** There are two signals associated with the I²C bus: the serial clock line (SCL) and the serial data line (SDA). The latter is a bidirectional line used for sending and receiving the data to/from the interface. External pullup resistors connected to VDDIO are required for SDA and SCL. When the bus is free both the lines are high. The I²C interface is compliant with fast mode (400 kHz), and normal mode (100 kHz) I²C standards. Operation at frequencies higher than 400 kHz is possible, but depends on several factors including the pullup resistor values, and total bus capacitance (trace + device capacitance). See Table 11 for more information. FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 14.

’474 Patent Claim	Representative NXP Product(s)

10.2.1 General SPI operation

NOTE

FXOS8700CQ only supports a point-to-point SPI protocol, with only one master (MCU) and one slave device (FXOS8700CQ) present on the bus. FXOS8700CQ does not tri-state the MISO pin when the CS_B pin is deasserted (logic high), which can lead to a bus conflict if multiple slave devices are present on the bus.

Do not connect more than one master and one slave device on the SPI bus.

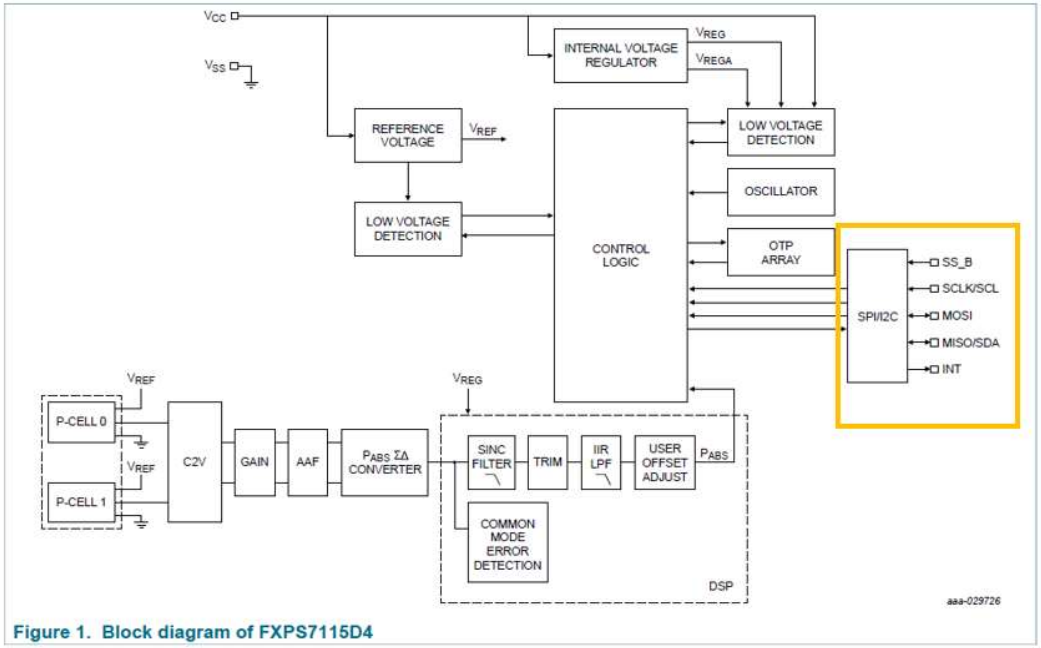
The CS_B pin is driven low at the start of a SPI transaction, held low for the duration of the transaction, and driven high after the transaction is complete. During a transaction the master toggles the SPI clock (SCLK) and transmits data on the MOSI pin.

FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 17.

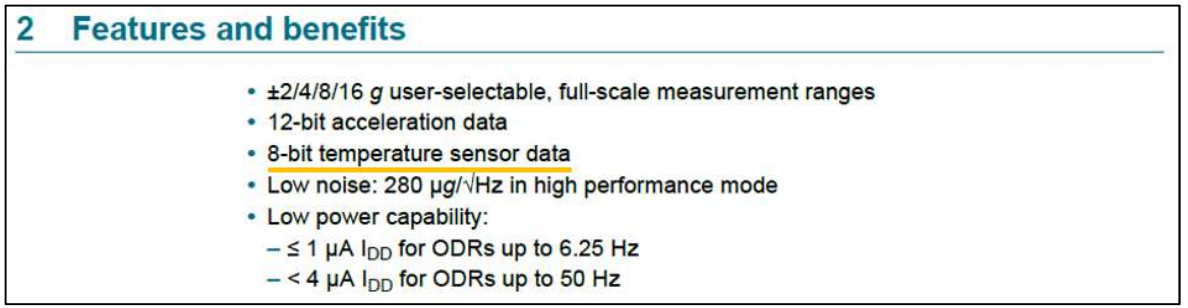
Table 13. Serial interface pin descriptions

Pin name	Pin description
VDDIO	Digital interface power
SA1/CS_B	I^2C second least significant bit of device address/SPI chip select
SCL/SCLK	I^2C/SPI serial clock
SDA/MOSI	I^2C serial data/SPI master serial data out slave serial data in
SA0/MISO	I^2C least significant bit of the device address/SPI master serial data in slave out

FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 18.

’474 Patent Claim	Representative NXP Product(s)
	NXP Semiconductors **FXPS7115D4** Digital absolute pressure sensor, 40 kPa to 115 kPa **5 Block diagram**  Figure 1. Block diagram of FXPS7115D4 FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 3.

’474 Patent Claim	Representative NXP Product(s)
	7.4.1 I^2C bit transmissions The state of SDA when SCL is high determines the bit value being transmitted. SDA must be stable when SCL is high and change when SCL is low as shown in Figure 14. After the START signal has been transmitted by the master, the bus is considered busy. Timing for the start condition is specified in Table 105. FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 13. **7.5 Standard 32-bit SPI protocol** The device includes a standard SPI protocol requiring 32-bit data packets. The device is a slave device and requires that the base clock value be low (CPOL = 0) with data captured on the rising edge of the clock and data propagated on the falling edge of the clock (CPHA = 0). The most significant bit is transferred first (MSB first). SPI transfers are completed through a sequence of two phases. During the first phase, the command is transmitted from the SPI master to the device. During the second phase, response data is transmitted from the slave device. MOSI and SCLK transitions are ignored when SS_B is not asserted. FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 17. 9 — SCLK/SCL — In I^2C mode, input pin 9 provides the serial clock. This pin must be connected to V_{CC} with an external pull-up resistor, as shown in the application diagram. In SPI mode, input pin 9 provides the serial clock. An internal pull-down device is connected to this pin. 10 — MOSI — SPI data in. In SPI mode, pin 10 functions as the serial data input to the SPI port. An internal pull-down device is connected to this pin. FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 4.
[8b.] sending a read	The Accused ’474 Sensors performs a step of sending a read temperature command to the serial

’474 Patent Claim	Representative NXP Product(s)
temperature command to the serial interface slave from the serial interface master using the clock signal line and the data signal line; and	interface slave from the serial interface master using the clock signal line and the data signal line. For example, each of the Accused ’474 Sensors perform a step of sending a read temperature command (*e.g.*, the read temperature command directed to the TEMP_OUT register) to the serial interface slave identified above from the serial interface master identified above using the clock signal line identified above and the data signal line identified above.  **2 Features and benefits** • ±2/4/8/16 *g* user-selectable, full-scale measurement ranges • 12-bit acceleration data • 8-bit temperature sensor data • Low noise: 280 µ*g*/√Hz in high performance mode • Low power capability: – ≤ 1 µA I_{DD} for ODRs up to 6.25 Hz – < 4 µA I_{DD} for ODRs up to 50 Hz FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 1.

’474 Patent Claim	Representative NXP Product(s)

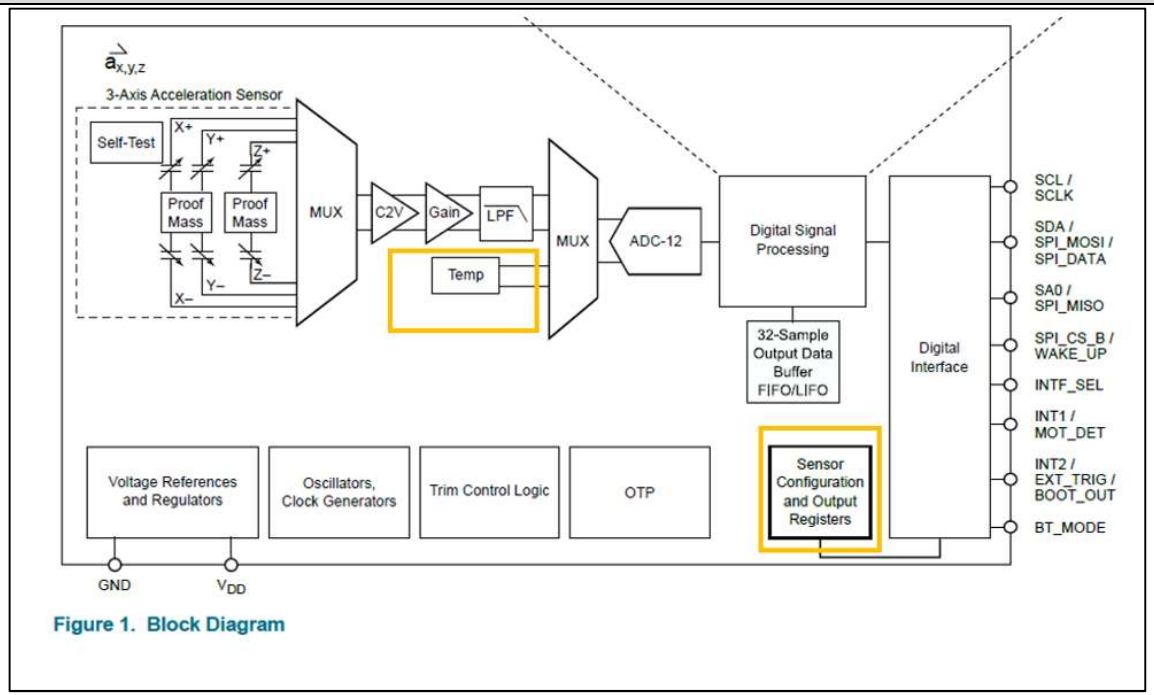


Figure 1. Block Diagram

FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 3.

13.2 TEMP_OUT register (address 01h)

Table 22. TEMP_OUT register (address 01h) bit allocation

Bit	7	6	5	4	3	2	1	0
Name	TEMP_OUT[7:0]							
Reset	—	—	—	—	—	—	—	—
Access	R	R	R	R	R	R	R	R

The **TEMP_OUT** register contains the 8-bit, 2's complement temperature value. When this register contains the value 00h, the measured temperature is 25 °C (typ). This register is updated on every ODR cycle.

FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 3.

’474 Patent Claim	Representative NXP Product(s)

10 I^2C digital interface

The registers embedded within FXLS8962AF may be accessed using an I^2C interface when the INTF_SEL pin is tied to GND. If the V_{DD} supply is not present, the device will be in shutdown mode and any communications on the interface are ignored. When the device is on a common I^2C-bus with other slave devices, the V_{DD} supply pin must be left unconnected (high-impedance) when the device supply is turned off to ensure that the internal ESD protection diodes do not become forward biased and prevent the bus from functioning normally (clamping).

FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 16.

10.1.2 I^2C read/write operations

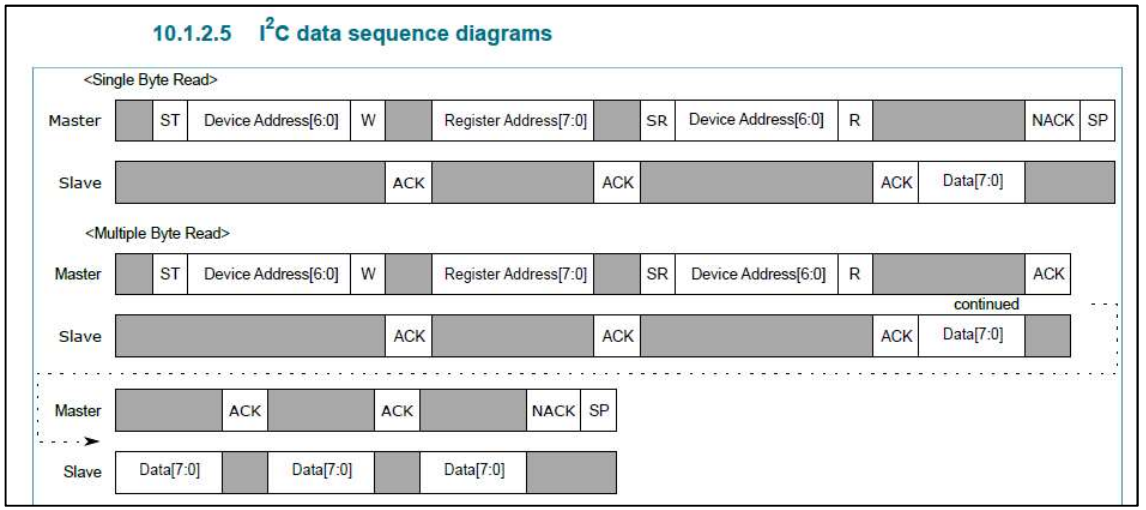
10.1.2.1 Single byte read

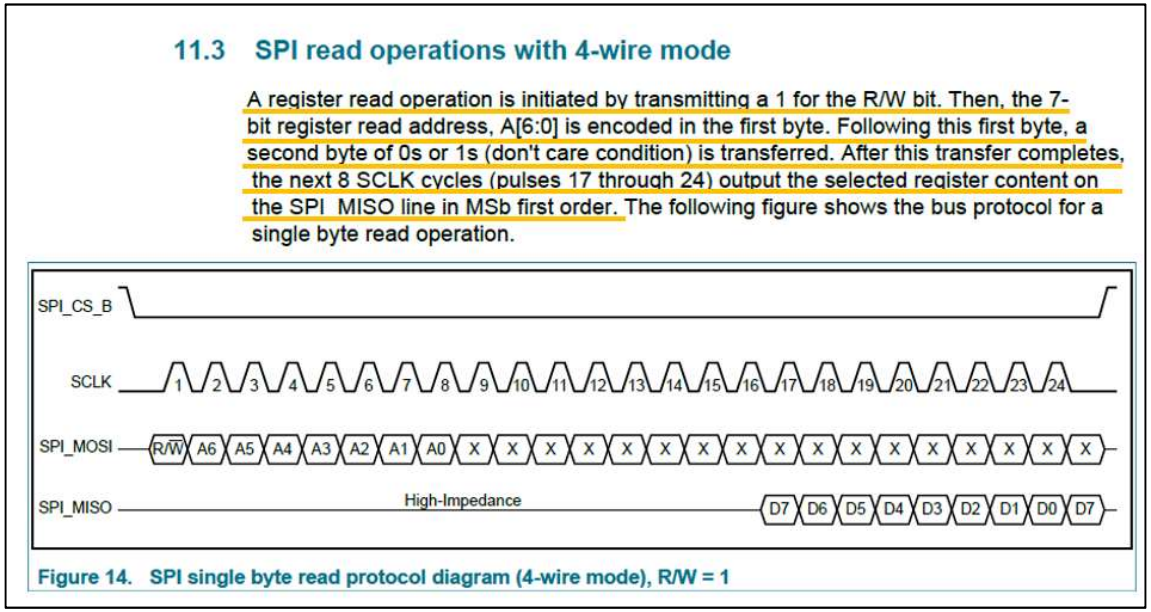
The master transmits a start condition (ST) to FXLS8962AF, followed by the slave address, with the R/W bit set to '0' for a write, and the FXLS8962AF sends an acknowledgement. Then the master transmits the address of the register to read and the FXLS8962AF sends an acknowledgement. The master transmits a repeated start condition (SR), followed by the slave address with the R/W bit set to '1' for a read from the previously selected register. The FXLS8962AF then acknowledges and transmits the data from the requested register. The master does not acknowledge (NAK) the transmitted data and then transmits a stop condition to end the data transfer.

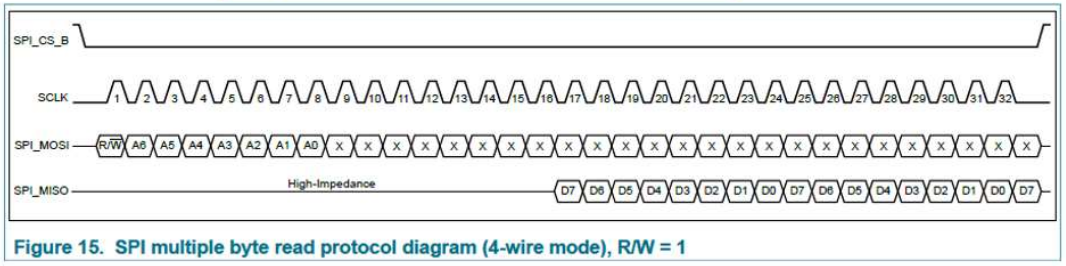
10.1.2.2 Multiple byte read

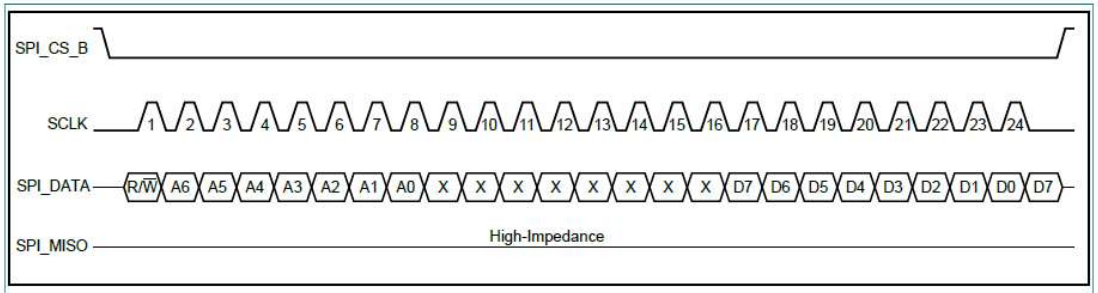
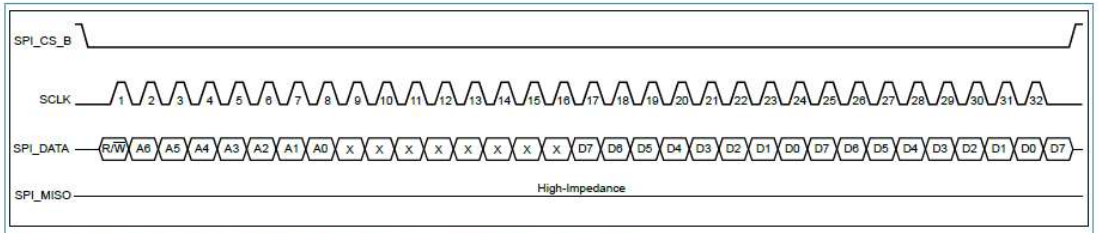
When performing a multi-byte or *burst* read, FXLS8962AF automatically increments the register read address pointer after a read command is received. Therefore, after following the steps of a single-byte read, multiple bytes of data can be read from sequential register addresses after each FXLS8962AF acknowledgment (ACK) is received until a no acknowledge (NAK) is issued by the master followed by a stop condition (SP) signaling the end of the transfer.

FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 18-19.

’474 Patent Claim	Representative NXP Product(s)
	 FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 18-20. **11.1 General SPI operation** The SPI_CS_B pin is driven low at the start of a transaction, held low for the duration of the transfer, and then driven high again after the transaction is completed. During a transaction, the master toggles the clock (SCLK). The SCLK polarity is defined as having an idle value that is low, and an active phase that is high (CPOL = 0). Serial input and output data is captured on the clock's rising edge and propagated on the falling edge (CPHA = 0). Single byte read and single byte write operations are completed in 24 SCLK cycles; multiple byte reads and writes are completed in additional multiples of 8 SCLK cycles per read or written byte. The first SCLK cycle latches the R/W (Read/Write) bit to select whether the desired operation is a read (R/W = 1) or a write (R/W = 0). The following seven SCLK cycles are used to latch the register read or write address. ***Note:*** *4-wire SPI interface mode is the default out of POR or after a soft reset. The 3-wire interface mode may also be selected by setting* ***SENS_CONFIG1[****SPI_M****]*** *= 1.*

’474 Patent Claim	Representative NXP Product(s)
	FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 18-20-21.  FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 18-21-22.

’474 Patent Claim	Representative NXP Product(s)
	Multiple-byte read operations are performed similarly to single-byte reads with additional bytes read out in multiples of eight SCLK cycles. The register read address is auto-incremented by FXLS8962AF so that every eighth clock edge will latch the address of the next sequential register read address. When the desired number of bytes has been read, a rising edge on SPI_CS_B terminates the transaction.  Figure 15. SPI multiple byte read protocol diagram (4-wire mode), R/W = 1 FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 18-22. 11.4 SPI read operations with 3-wire mode FXLS8962AF can be configured to operate in 3-wire software enabled SPI mode. In this mode, the SPI_MISO pin is left unconnected and the SPI_MOSI pin becomes a bidirectional input/output pin (SPI_DATA). Read operations in 3-wire mode are the same as write operations in 3- and 4-wire modes, except that at the end of the address cycle (falling edge of clock pulse 16), the SPI_DATA pin automatically switches from an input to an output and with bit D7 as the current output state. FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 18-22.

’474 Patent Claim	Representative NXP Product(s)
	 Figure 16. SPI single byte read protocol diagram (3-wire mode)  Figure 17. SPI multiple byte read protocol diagram (3-wire mode) FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 18-23.

’474 Patent Claim	Representative NXP Product(s)

14.3 Temperature register

14.3.1 TEMP register (address 0x51)

Eight-bit 2's complement sensor temperature value with 0.96 °C/LSB sensitivity. Temperature data is only valid between –40 °C and 125 °C. The temperature sensor output is only valid when M_CTRL_REG1[m_hms] > 0b00. Please note that the temperature sensor is uncalibrated and its output for a given temperature will vary from one device to the next.

Table 50. TEMP register (address 0x51) bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	die_temperature[7:0]							
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

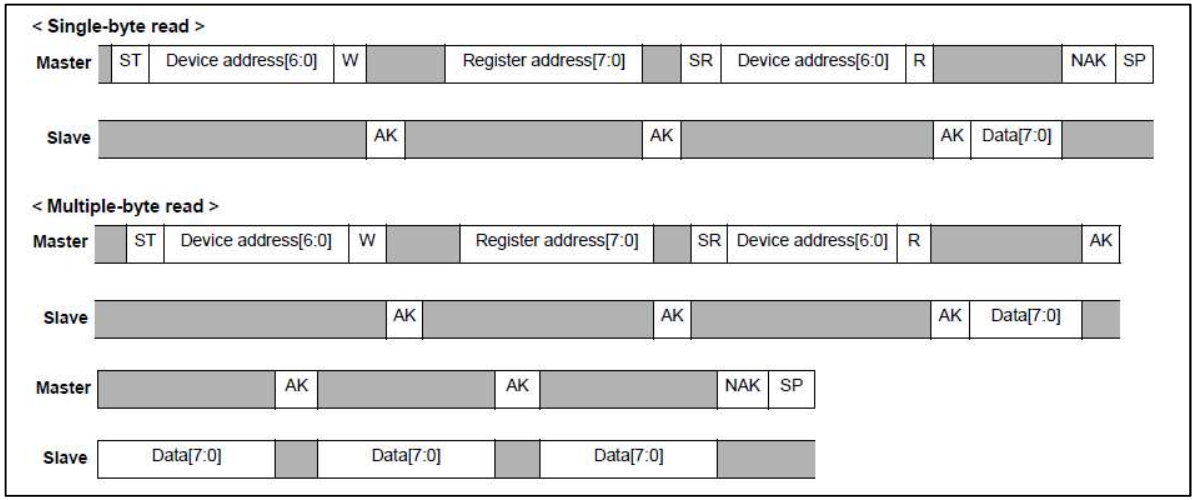
FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 52.

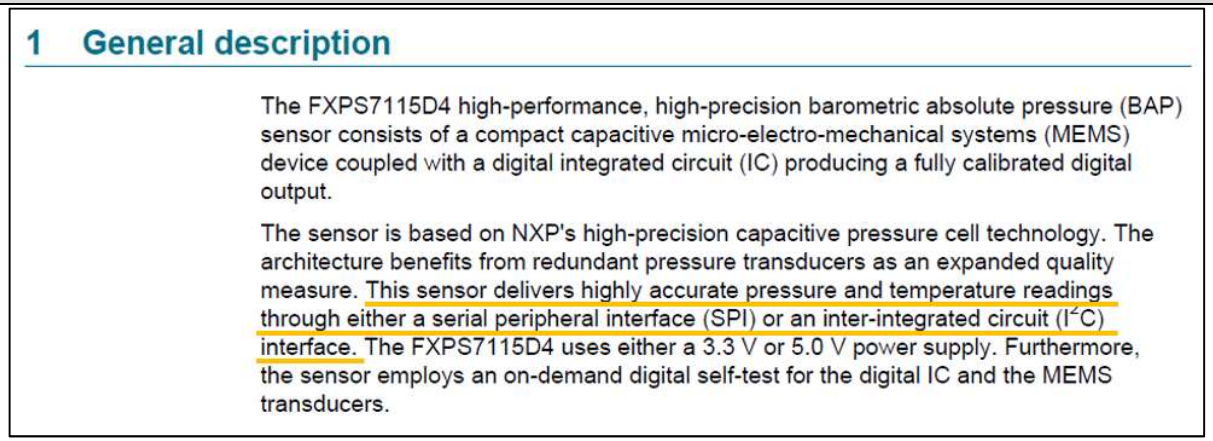
10.1.2 I²C read/write operations

Single-byte read

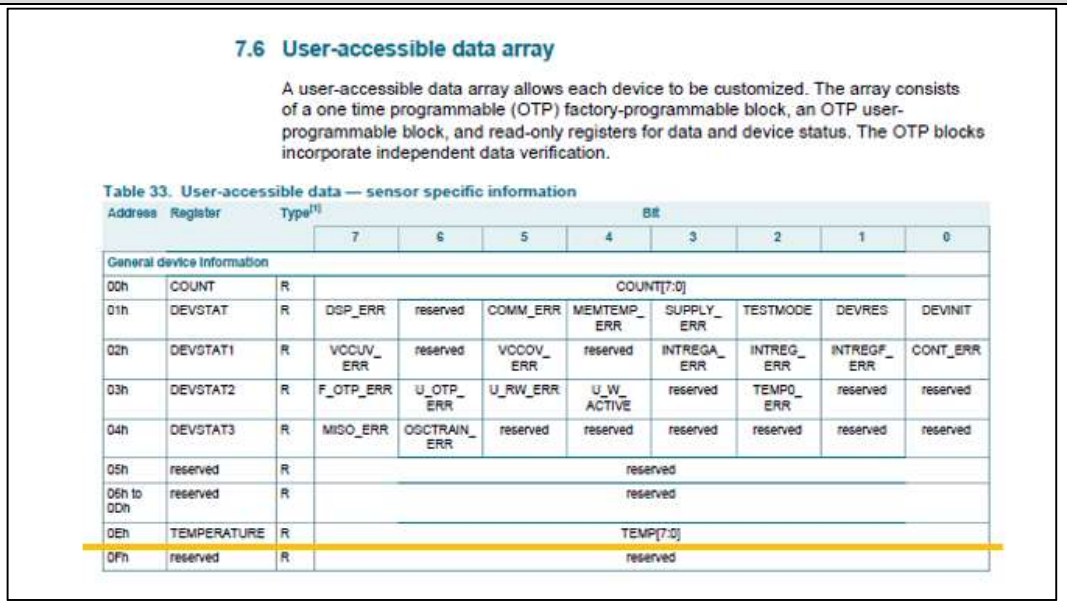
The master (or MCU) transmits a start condition (ST) to the FXOS8700CQ, followed by the slave address, with the R/W bit set to “0” for a write, and the FXOS8700CQ sends an acknowledgement. Then the master (or MCU) transmits the address of the register to read and the FXOS8700CQ sends an acknowledgement. The master (or MCU) transmits a repeated start condition (SR), followed by the slave address with the R/W bit set to “1” for a read from the previously selected register. The FXOS8700CQ then acknowledges and transmits the data from the requested register. The master does not acknowledge (NAK) the transmitted data, but transmits a stop condition to end the data transfer.

When performing a multi-byte or “burst” read, the FXOS8700CQ automatically increments the register address read pointer after a read command is received. Therefore, after following the steps of a single-byte read, multiple bytes of data can be read from sequential registers after each FXOS8700CQ acknowledgment (AK) is received until a no acknowledge (NAK) occurs from the master followed by a stop condition (SP) signaling an end of transmission.

’474 Patent Claim	Representative NXP Product(s)
	FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 15.  FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 16. **10.2.2 SPI read/write operations** A read operation is initiated by transmitting a 0 for the R/W bit. Then the 8-bit register address, ADDR[7:0] is encoded in the first and second serialized bytes. Subsequent bits are ignored by the part. The read data is deserialized from the MISO pin. FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 18.

’474 Patent Claim	Representative NXP Product(s)
	 1 General description The FXPS7115D4 high-performance, high-precision barometric absolute pressure (BAP) sensor consists of a compact capacitive micro-electro-mechanical systems (MEMS) device coupled with a digital integrated circuit (IC) producing a fully calibrated digital output. The sensor is based on NXP's high-precision capacitive pressure cell technology. The architecture benefits from redundant pressure transducers as an expanded quality measure. This sensor delivers highly accurate pressure and temperature readings through either a serial peripheral interface (SPI) or an inter-integrated circuit (I^2C) interface. The FXPS7115D4 uses either a 3.3 V or 5.0 V power supply. Furthermore, the sensor employs an on-demand digital self-test for the digital IC and the MEMS transducers. FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 1.

’474 Patent Claim	Representative NXP Product(s)
	7.3.4 Temperature sensor 7.3.4.1 Temperature sensor signal chain The device includes a temperature sensor for signal compensation and user readability. Figure 12 shows a simplified block diagram. Temperature sensor parameters are specified in Table 104 and Table 105.  Figure 12. Temperature sensor signal chain block diagram 7.3.4.2 Temperature sensor output scaling equation Equation 5 is used to convert temperature readings with the variables specified in Table 8. $$T_{DEGC} = \frac{T_{LSB} - T0_{LSB}}{T_{SENSE}} \quad (5)$$ where: T_{DEGC} = The temperature output in degrees C T_{LSB} = The temperature output in LSB $T0_{LSB}$ = The expected temperature output in LSB at 0 °C T_{SENSE} = The expected temperature sensitivity in LSB/°C Table 8. Temperature conversion variables Data reading | $T0_{LSB}$ (LSB) | T_{SENSE} LSB/C) 8-bit register read | 68 | 1 FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 12.

’474 Patent Claim	Representative NXP Product(s)
	

7.6 User-accessible data array

A user-accessible data array allows each device to be customized. The array consists of a one time programmable (OTP) factory-programmable block, an OTP user-programmable block, and read-only registers for data and device status. The OTP blocks incorporate independent data verification.

Table 33. User-accessible data — sensor specific information

Address	Register	Type[1]	Bit							
			7	6	5	4	3	2	1	0
General device information										
00h	COUNT	R	COUNT[7:0]							
01h	DEVSTAT	R	DSP_ERR	reserved	COMM_ERR	MEMTEMP_ERR	SUPPLY_ERR	TESTMODE	DEVRES	DEVINIT
02h	DEVSTAT1	R	VCCUV_ERR	reserved	VCCOV_ERR	reserved	INTREGA_ERR	INTREG_ERR	INTREGF_ERR	CONT_ERR
03h	DEVSTAT2	R	F_OTP_ERR	U_OTP_ERR	U_RW_ERR	U_W_ACTIVE	reserved	TEMP0_ERR	reserved	reserved
04h	DEVSTAT3	R	MISO_ERR	OSCTRAIN_ERR	reserved	reserved	reserved	reserved	reserved	reserved
05h	reserved	R	reserved							
06h to 0Dh	reserved	R	reserved							
0Eh	TEMPERATURE	R	TEMP[7:0]							
0Fh	reserved	R	reserved							

FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 27.

’474 Patent Claim	Representative NXP Product(s)
	7.4.6.2 Register read transfers The device supports I^2C register read data transfers. Register read data transfers are constructed as follows: 1. The master transmits a START condition. 2. The master transmits the 7-bit slave address. 3. The master transmits a '0' for the read/write bit to indicate a write operation. 4. The slave transmits an ACK. 5. The master transmits the register address to be read. 6. The slave transmits an ACK. 7. The master transmits a repeat START condition. 8. The master transmits the 7-bit slave address. 9. The master transmits a '1' for the read/write bit to indicate a read operation. 10. The slave transmits an ACK. 11. The slave transmits the data from the register addressed. 12. The master transmits a NACK. 13. The master transmits a STOP condition.  FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 16.

’474 Patent Claim	Representative NXP Product(s)
	7.5.3 Command summary **7.5.3.1 Register read command** The device supports a register read command. The register read command uses the upper 7 bits of the addresses defined in Section 7.6 "User-accessible data array" to address 8-bit registers in the register map. The response to a register read command is shown in Section 7.5.3.1.2 "Register read response message format". The response is transmitted on the next SPI message if and only if all of the following conditions are met: • No SPI error is detected (see Section 7.5.5.3 "SPI error") • No MISO error is detected (see Section 7.5.5.4 "SPI data output verification error") If these conditions are met, the device responds to the register read request as shown in Section 7.5.3.1.2 "Register read response message format". Otherwise, the device responds with the error response as defined in Section 7.5.5.2 "Detailed status field". The register read response includes the register contents at the rising edge of SS_B for the register read command. FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 20.

’474 Patent Claim	Representative NXP Product(s)

7.5.3.1.1 Register read command message format

Table 13. Register read command message format

MSB: bit 31; LSB: bit 0

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
Register access command																															
Command C[3:0]				Fixed bits: must = 0h				Register address								Register data								8-bit CRC							
1	1	0	0	0	0	0	0	RA[7:1]							RA[0]	0	0	0	0	0	0	0	0	CRC[7:0]							

Table 14. Register read command message bit field descriptions

Bit field	Definition
C[3:0]	Register read command = '1100'
RA[7:0]	RA[7:1] contains the word address of the register to be read.
CRC[7:0]	Read CRC Section

7.5.3.1.2 Register read response message format

Table 15. Register read response message format

MSB: bit 31; LSB: bit 0

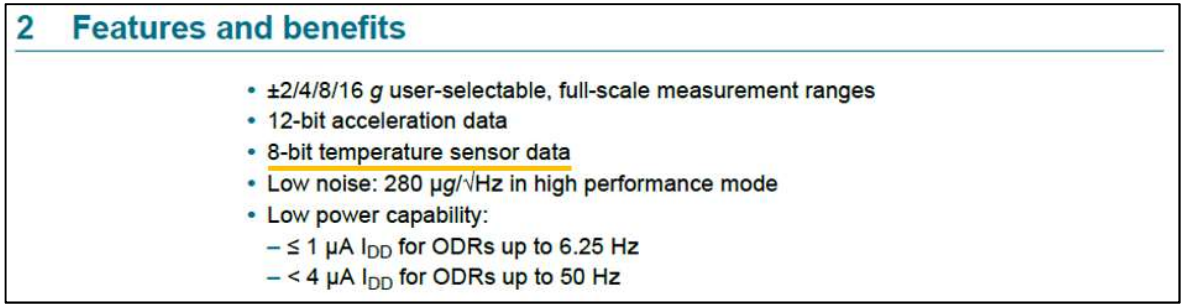
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Register access command																															
Command C[0], [3:1]				Basic Status		Unused Data 0h		Register data: contents of RA[7:1] high byte								Register data: contents of RA[7:1] low byte								8-bit CRC							
0	1	1	0	ST[1:0]		0	0	RD[15:8]								RD[7:0]								CRC[7:0]							

FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 20.

Table 16. Register read response message bit field descriptions

Bit field	Definition
C[0], [3:1]	Register Read Command = '0110'
ST[1:0]	Status
RD[15:8]	The contents of the register addressed by RA[7:1] high byte (RA[0] = 1)
RD[7:0]	The contents of the register addressed by RA[7:1] low byte (RA[0] = 0)
CRC[7:0]	8-bit CRC

FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 21.

’474 Patent Claim	Representative NXP Product(s)
[8c.] in response to the read temperature command, the serial interface slave supplying over the data signal line a temperature value associated with a processor on an integrated circuit containing the serial interface slave.	The Accused ’474 Sensors performs a step of in response to the read temperature command, the serial interface slave supplying over the data signal line a temperature value associated with a processor on an integrated circuit containing the serial interface slave. For example, each of the Accused ’474 Sensors perform a step of in response to the read temperature command, the serial interface slave identified above supplying over the data signal line identified above a temperature value (*e.g.*, the temperature value in the TEMP_OUT register) associated with a processor on an integrated circuit containing the serial interface slave identified above.  **2 Features and benefits** • ±2/4/8/16 *g* user-selectable, full-scale measurement ranges • 12-bit acceleration data • 8-bit temperature sensor data • Low noise: 280 µ*g*/√Hz in high performance mode • Low power capability: – ≤ 1 µA I_{DD} for ODRs up to 6.25 Hz – < 4 µA I_{DD} for ODRs up to 50 Hz FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 1.

’474 Patent Claim	Representative NXP Product(s)

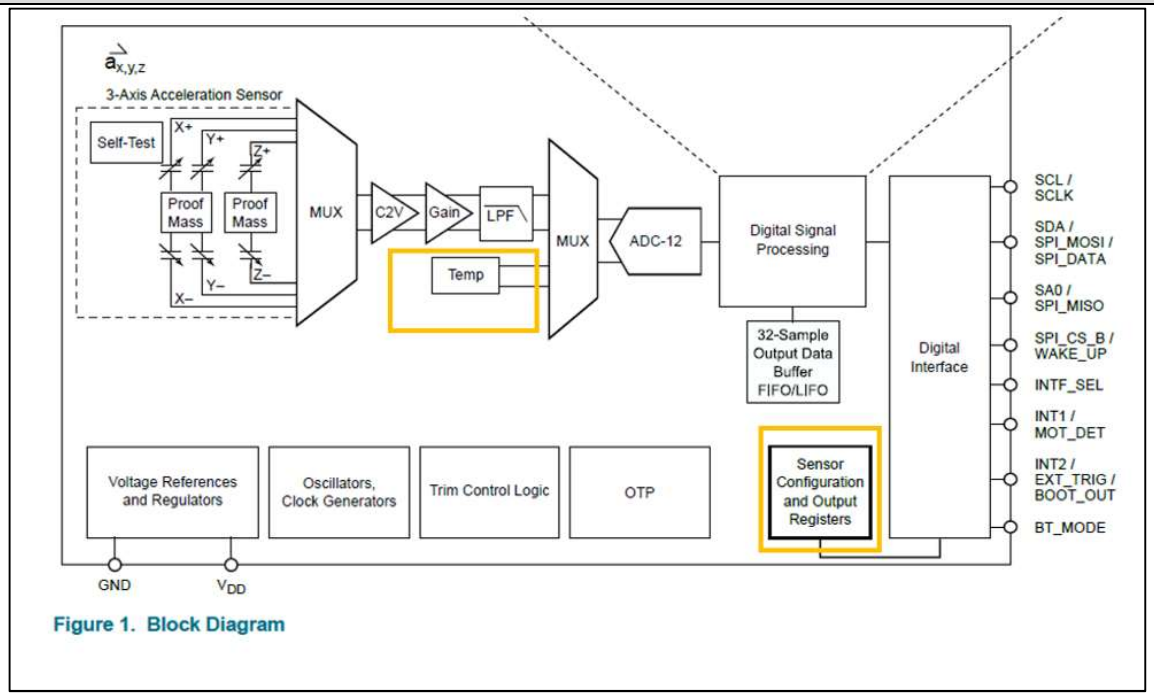


Figure 1. Block Diagram

FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 3.

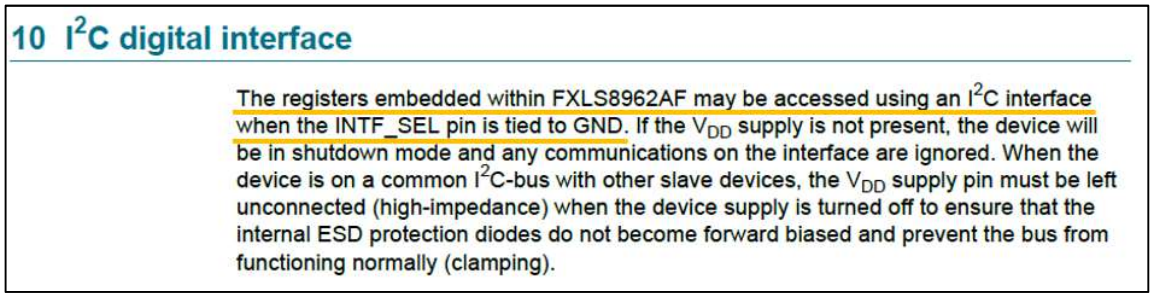
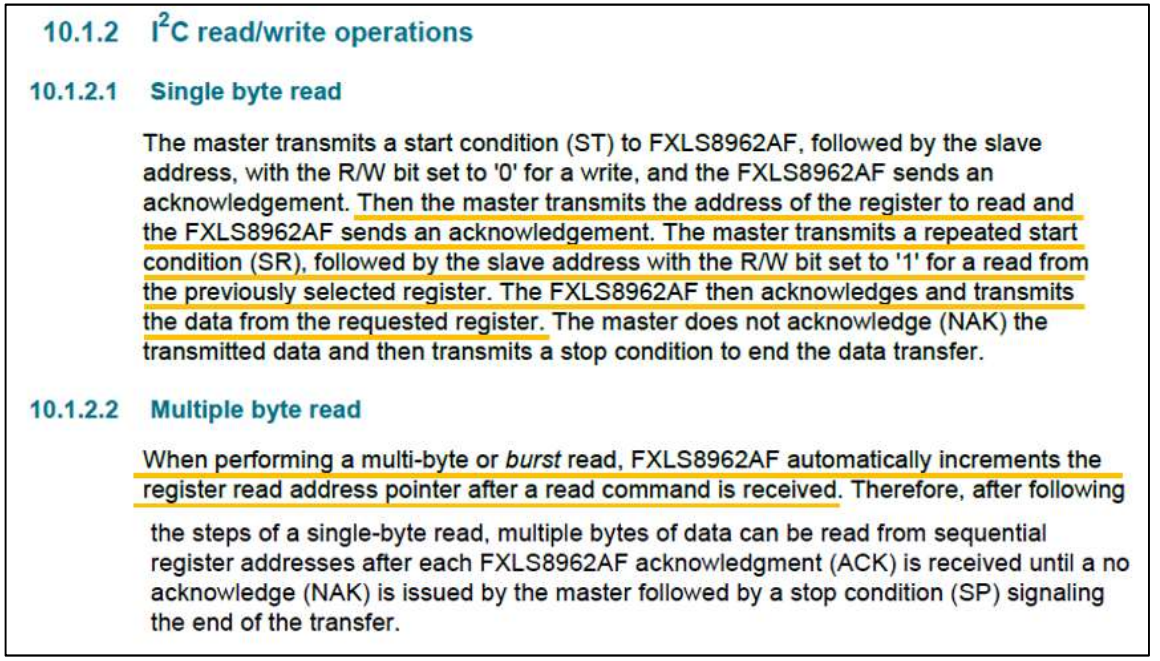
13.2 TEMP_OUT register (address 01h)

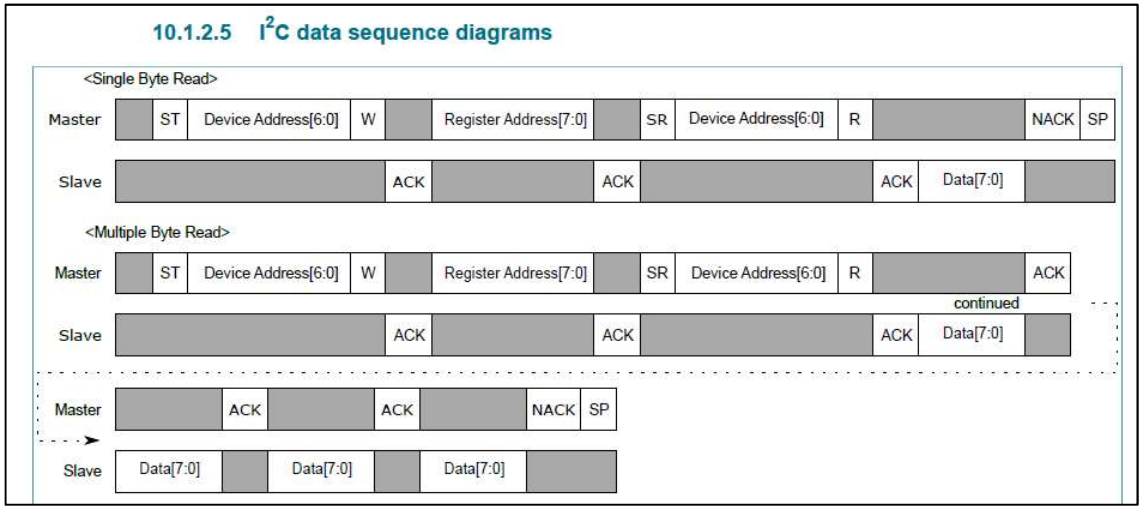
Table 22. TEMP_OUT register (address 01h) bit allocation

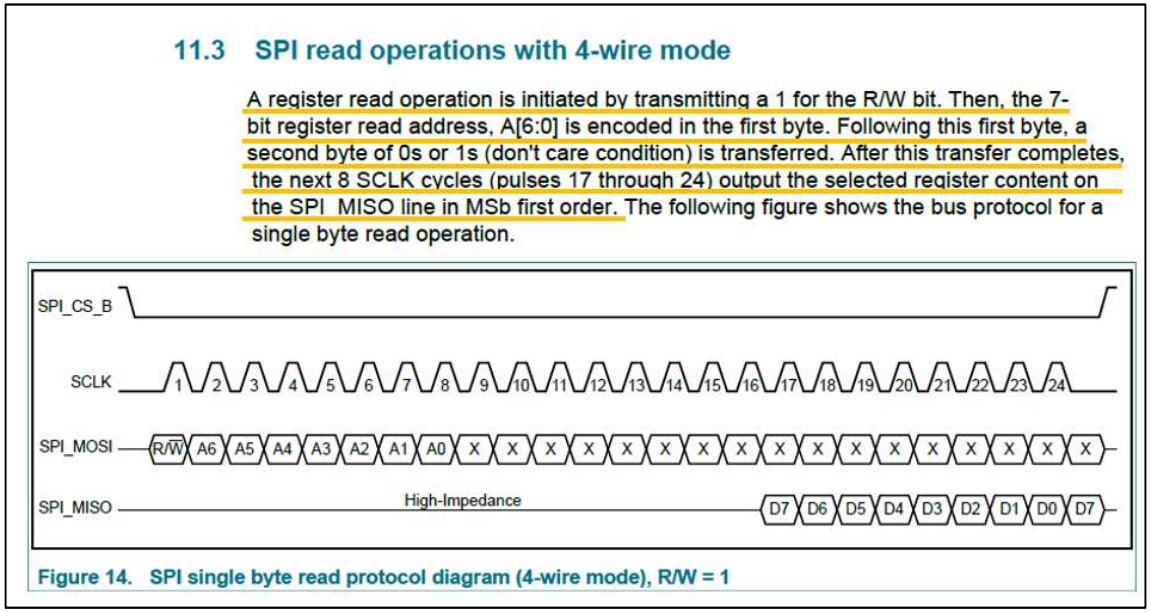
Bit	7	6	5	4	3	2	1	0
Name	TEMP_OUT[7:0]							
Reset	—	—	—	—	—	—	—	—
Access	R	R	R	R	R	R	R	R

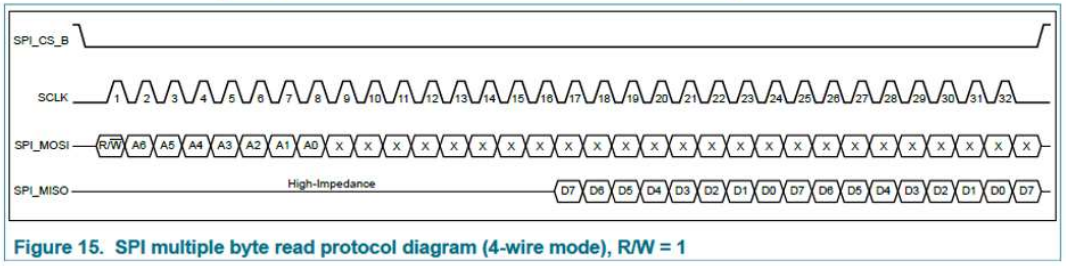
The **TEMP_OUT** register contains the 8-bit, 2's complement temperature value. When this register contains the value 00h, the measured temperature is 25 °C (typ). This register is updated on every ODR cycle.

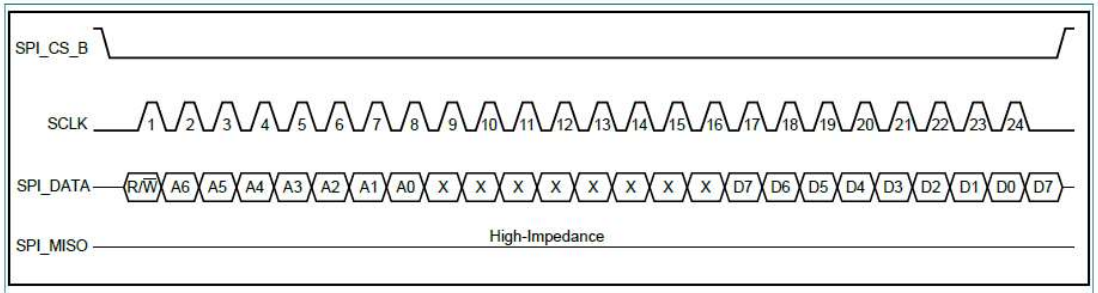
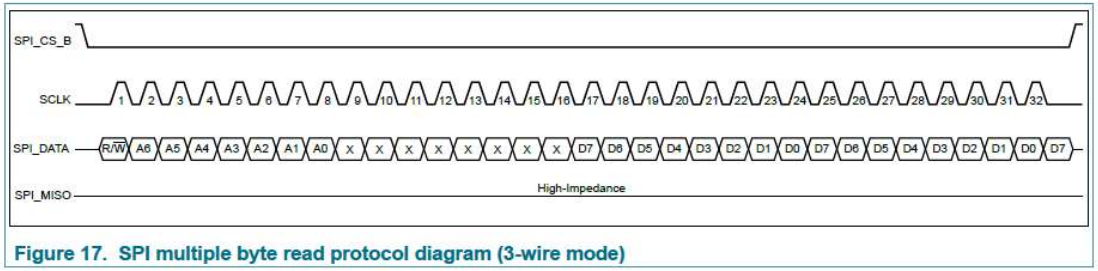
FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 3.

’474 Patent Claim	Representative NXP Product(s)
	 10 I²C digital interface The registers embedded within FXLS8962AF may be accessed using an I²C interface when the INTF_SEL pin is tied to GND. If the V_{DD} supply is not present, the device will be in shutdown mode and any communications on the interface are ignored. When the device is on a common I²C-bus with other slave devices, the V_{DD} supply pin must be left unconnected (high-impedance) when the device supply is turned off to ensure that the internal ESD protection diodes do not become forward biased and prevent the bus from functioning normally (clamping). FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 16.  10.1.2 I²C read/write operations 10.1.2.1 Single byte read The master transmits a start condition (ST) to FXLS8962AF, followed by the slave address, with the R/W bit set to '0' for a write, and the FXLS8962AF sends an acknowledgement. Then the master transmits the address of the register to read and the FXLS8962AF sends an acknowledgement. The master transmits a repeated start condition (SR), followed by the slave address with the R/W bit set to '1' for a read from the previously selected register. The FXLS8962AF then acknowledges and transmits the data from the requested register. The master does not acknowledge (NAK) the transmitted data and then transmits a stop condition to end the data transfer. 10.1.2.2 Multiple byte read When performing a multi-byte or *burst* read, FXLS8962AF automatically increments the register read address pointer after a read command is received. Therefore, after following the steps of a single-byte read, multiple bytes of data can be read from sequential register addresses after each FXLS8962AF acknowledgment (ACK) is received until a no acknowledge (NAK) is issued by the master followed by a stop condition (SP) signaling the end of the transfer. FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 18-19.

’474 Patent Claim	Representative NXP Product(s)
	 FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 18-20. **11.1 General SPI operation** The SPI_CS_B pin is driven low at the start of a transaction, held low for the duration of the transfer, and then driven high again after the transaction is completed. During a transaction, the master toggles the clock (SCLK). The SCLK polarity is defined as having an idle value that is low, and an active phase that is high (CPOL = 0). Serial input and output data is captured on the clock's rising edge and propagated on the falling edge (CPHA = 0). Single byte read and single byte write operations are completed in 24 SCLK cycles; multiple byte reads and writes are completed in additional multiples of 8 SCLK cycles per read or written byte. The first SCLK cycle latches the R/W (Read/Write) bit to select whether the desired operation is a read (R/W = 1) or a write (R/W = 0). The following seven SCLK cycles are used to latch the register read or write address. ***Note:*** *4-wire SPI interface mode is the default out of POR or after a soft reset. The 3-wire interface mode may also be selected by setting* ***SENS_CONFIG1[****SPI_M****]*** *= 1.*

’474 Patent Claim	Representative NXP Product(s)
	FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 18-20-21.  FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 18-21-22.

’474 Patent Claim	Representative NXP Product(s)
	Multiple-byte read operations are performed similarly to single-byte reads with additional bytes read out in multiples of eight SCLK cycles. The register read address is auto-incremented by FXLS8962AF so that every eighth clock edge will latch the address of the next sequential register read address. When the desired number of bytes has been read, a rising edge on SPI_CS_B terminates the transaction.  Figure 15. SPI multiple byte read protocol diagram (4-wire mode), R/W = 1 FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 18-22. 11.4 SPI read operations with 3-wire mode FXLS8962AF can be configured to operate in 3-wire software enabled SPI mode. In this mode, the SPI_MISO pin is left unconnected and the SPI_MOSI pin becomes a bidirectional input/output pin (SPI_DATA). Read operations in 3-wire mode are the same as write operations in 3- and 4-wire modes, except that at the end of the address cycle (falling edge of clock pulse 16), the SPI_DATA pin automatically switches from an input to an output and with bit D7 as the current output state. FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 18-22.

’474 Patent Claim	Representative NXP Product(s)
	 **Figure 16. SPI single byte read protocol diagram (3-wire mode)**  **Figure 17. SPI multiple byte read protocol diagram (3-wire mode)** FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 18-23.

’474 Patent Claim	Representative NXP Product(s)

14.3 Temperature register

14.3.1 TEMP register (address 0x51)

Eight-bit 2's complement sensor temperature value with 0.96 °C/LSB sensitivity. Temperature data is only valid between –40 °C and 125 °C. The temperature sensor output is only valid when M_CTRL_REG1[m_hms] > 0b00. Please note that the temperature sensor is uncalibrated and its output for a given temperature will vary from one device to the next.

Table 50. TEMP register (address 0x51) bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	die_temperature[7:0]							
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

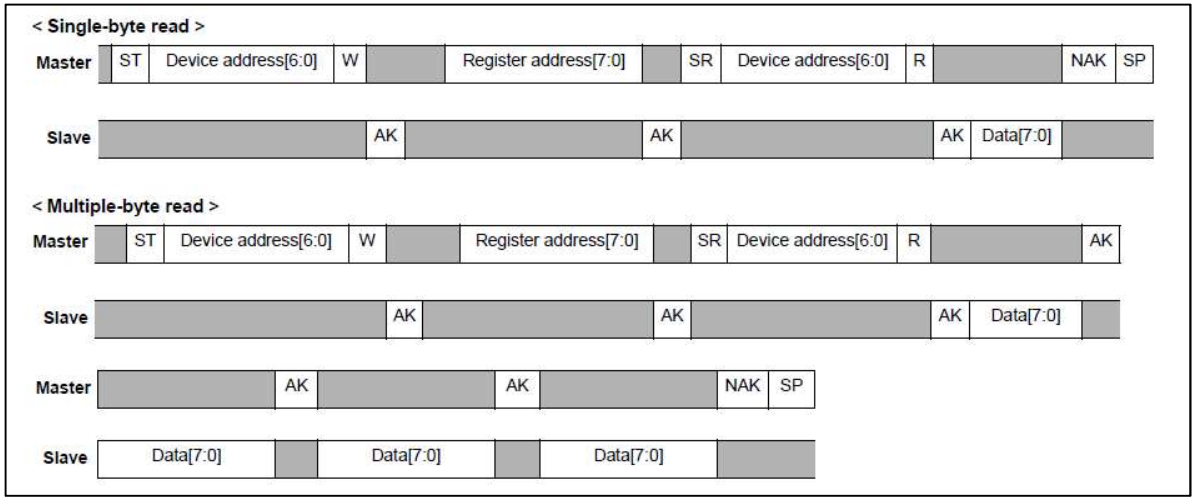
FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 52.

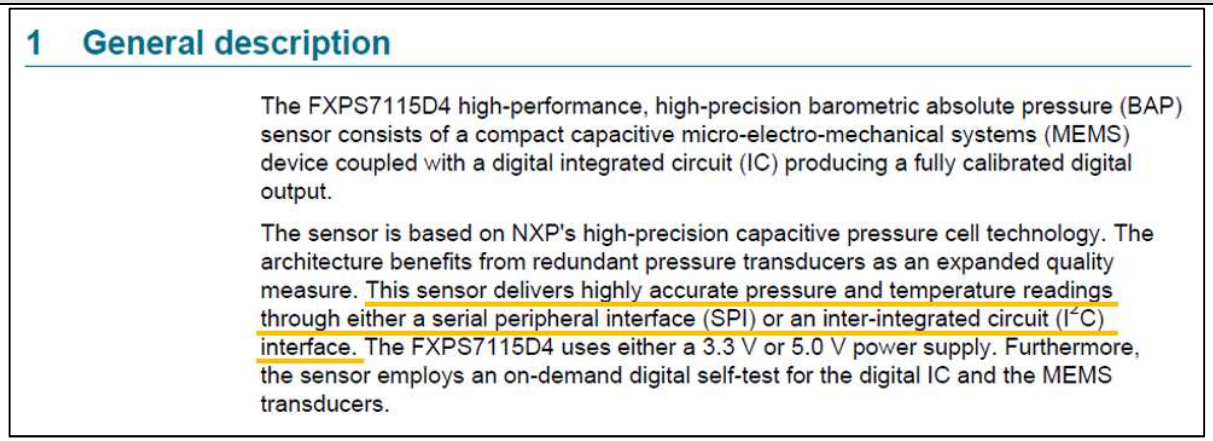
10.1.2 I²C read/write operations

Single-byte read

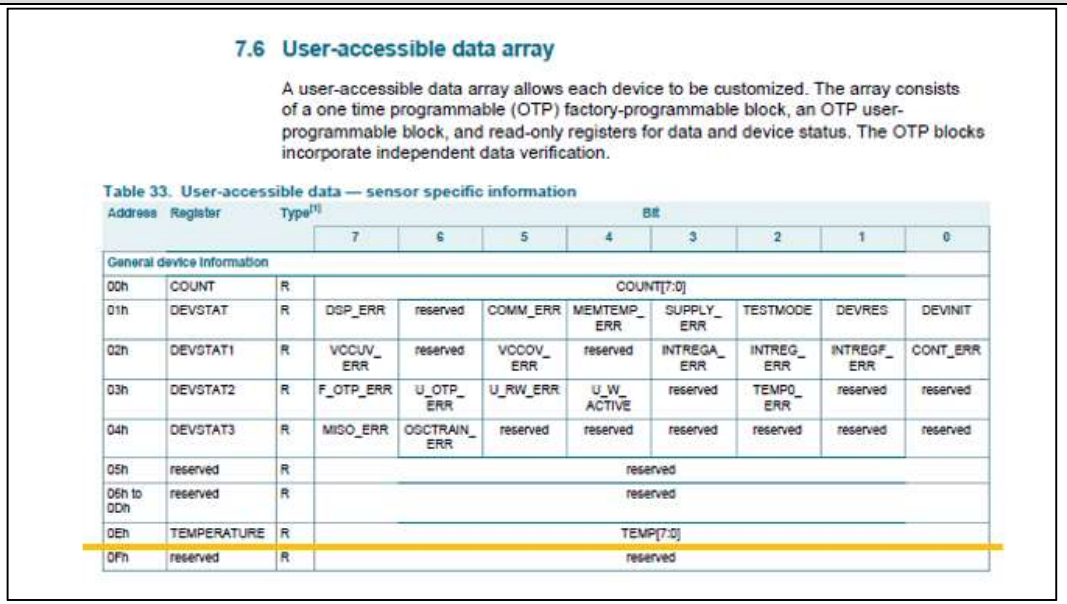
The master (or MCU) transmits a start condition (ST) to the FXOS8700CQ, followed by the slave address, with the R/W bit set to “0” for a write, and the FXOS8700CQ sends an acknowledgement. Then the master (or MCU) transmits the address of the register to read and the FXOS8700CQ sends an acknowledgement. The master (or MCU) transmits a repeated start condition (SR), followed by the slave address with the R/W bit set to “1” for a read from the previously selected register. The FXOS8700CQ then acknowledges and transmits the data from the requested register. The master does not acknowledge (NAK) the transmitted data, but transmits a stop condition to end the data transfer.

When performing a multi-byte or “burst” read, the FXOS8700CQ automatically increments the register address read pointer after a read command is received. Therefore, after following the steps of a single-byte read, multiple bytes of data can be read from sequential registers after each FXOS8700CQ acknowledgment (AK) is received until a no acknowledge (NAK) occurs from the master followed by a stop condition (SP) signaling an end of transmission.

’474 Patent Claim	Representative NXP Product(s)
	FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 15.  FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 16. **10.2.2 SPI read/write operations** A read operation is initiated by transmitting a 0 for the R/W bit. Then the 8-bit register address, ADDR[7:0] is encoded in the first and second serialized bytes. Subsequent bits are ignored by the part. The read data is deserialized from the MISO pin. FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 18.

’474 Patent Claim	Representative NXP Product(s)
	 1 General description The FXPS7115D4 high-performance, high-precision barometric absolute pressure (BAP) sensor consists of a compact capacitive micro-electro-mechanical systems (MEMS) device coupled with a digital integrated circuit (IC) producing a fully calibrated digital output. The sensor is based on NXP's high-precision capacitive pressure cell technology. The architecture benefits from redundant pressure transducers as an expanded quality measure. This sensor delivers highly accurate pressure and temperature readings through either a serial peripheral interface (SPI) or an inter-integrated circuit (I^2C) interface. The FXPS7115D4 uses either a 3.3 V or 5.0 V power supply. Furthermore, the sensor employs an on-demand digital self-test for the digital IC and the MEMS transducers. FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 1.

’474 Patent Claim	Representative NXP Product(s)
	7.3.4 Temperature sensor **7.3.4.1 Temperature sensor signal chain** The device includes a temperature sensor for signal compensation and user readability. Figure 12 shows a simplified block diagram. Temperature sensor parameters are specified in Table 104 and Table 105.  **Figure 12. Temperature sensor signal chain block diagram** **7.3.4.2 Temperature sensor output scaling equation** Equation 5 is used to convert temperature readings with the variables specified in Table 8. $$T_{DEGC} = \frac{T_{LSB} - T0_{LSB}}{T_{SENSE}} \quad (5)$$ where: T_{DEGC} = The temperature output in degrees C T_{LSB} = The temperature output in LSB $T0_{LSB}$ = The expected temperature output in LSB at 0 °C T_{SENSE} = The expected temperature sensitivity in LSB/°C **Table 8. Temperature conversion variables** | Data reading | $T0_{LSB}$ (LSB) | T_{SENSE} LSB/C) | |---|---|---| | 8-bit register read | 68 | 1 | FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 12.

’474 Patent Claim	Representative NXP Product(s)
	

7.6 User-accessible data array

A user-accessible data array allows each device to be customized. The array consists of a one time programmable (OTP) factory-programmable block, an OTP user-programmable block, and read-only registers for data and device status. The OTP blocks incorporate independent data verification.

Table 33. User-accessible data — sensor specific information

Address	Register	Type[1]	Bit							
			7	6	5	4	3	2	1	0
General device information										
00h	COUNT	R	COUNT[7:0]							
01h	DEVSTAT	R	DSP_ERR	reserved	COMM_ERR	MEMTEMP_ERR	SUPPLY_ERR	TESTMODE	DEVRES	DEVINIT
02h	DEVSTAT1	R	VCCUV_ERR	reserved	VCCOV_ERR	reserved	INTREGA_ERR	INTREG_ERR	INTREGF_ERR	CONT_ERR
03h	DEVSTAT2	R	F_OTP_ERR	U_OTP_ERR	U_RW_ERR	U_W_ACTIVE	reserved	TEMP0_ERR	reserved	reserved
04h	DEVSTAT3	R	MISO_ERR	OSCTRAIN_ERR	reserved	reserved	reserved	reserved	reserved	reserved
05h	reserved	R	reserved							
06h to 0Dh	reserved	R	reserved							
0Eh	TEMPERATURE	R	TEMP[7:0]							
0Fh	reserved	R	reserved							

FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 27.

’474 Patent Claim	Representative NXP Product(s)
	7.4.6.2 Register read transfers The device supports I^2C register read data transfers. Register read data transfers are constructed as follows: 1. The master transmits a START condition. 2. The master transmits the 7-bit slave address. 3. The master transmits a '0' for the read/write bit to indicate a write operation. 4. The slave transmits an ACK. 5. The master transmits the register address to be read. 6. The slave transmits an ACK. 7. The master transmits a repeat START condition. 8. The master transmits the 7-bit slave address. 9. The master transmits a '1' for the read/write bit to indicate a read operation. 10. The slave transmits an ACK. 11. The slave transmits the data from the register addressed. 12. The master transmits a NACK. 13. The master transmits a STOP condition.  FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 16.

’474 Patent Claim	Representative NXP Product(s)
	7.5.3 Command summary **7.5.3.1 Register read command** The device supports a register read command. The register read command uses the upper 7 bits of the addresses defined in Section 7.6 "User-accessible data array" to address 8-bit registers in the register map. The response to a register read command is shown in Section 7.5.3.1.2 "Register read response message format". The response is transmitted on the next SPI message if and only if all of the following conditions are met: • No SPI error is detected (see Section 7.5.5.3 "SPI error") • No MISO error is detected (see Section 7.5.5.4 "SPI data output verification error") If these conditions are met, the device responds to the register read request as shown in Section 7.5.3.1.2 "Register read response message format". Otherwise, the device responds with the error response as defined in Section 7.5.5.2 "Detailed status field". The register read response includes the register contents at the rising edge of SS_B for the register read command. FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 20.

’474 Patent Claim	Representative NXP Product(s)

7.5.3.1.1 Register read command message format

Table 13. Register read command message format

MSB: bit 31; LSB: bit 0

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
Register access command																															
Command C[3:0]				Fixed bits: must = 0h				Register address								Register data								8-bit CRC							
1	1	0	0	0	0	0	0	RA[7:1]							RA[0]	0	0	0	0	0	0	0	0	CRC[7:0]							

Table 14. Register read command message bit field descriptions

Bit field	Definition
C[3:0]	Register read command = '1100'
RA[7:0]	RA[7:1] contains the word address of the register to be read.
CRC[7:0]	Read CRC Section

7.5.3.1.2 Register read response message format

Table 15. Register read response message format

MSB: bit 31; LSB: bit 0

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
Register access command																															
Command C[0], [3:1]				Basic Status		Unused Data 0h		Register data: contents of RA[7:1] high byte								Register data: contents of RA[7:1] low byte								8-bit CRC							
0	1	1	0	ST[1:0]		0	0	RD[15:8]								RD[7:0]								CRC[7:0]							

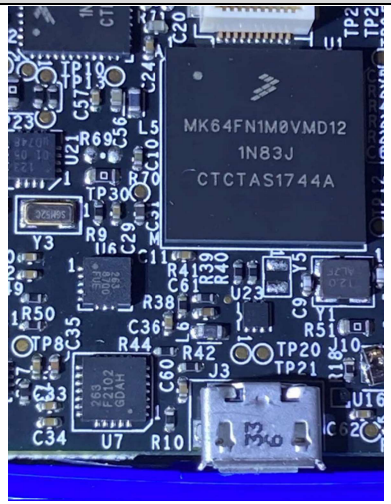
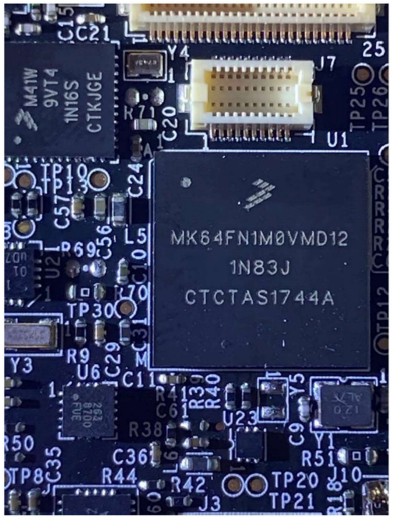
FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 20.

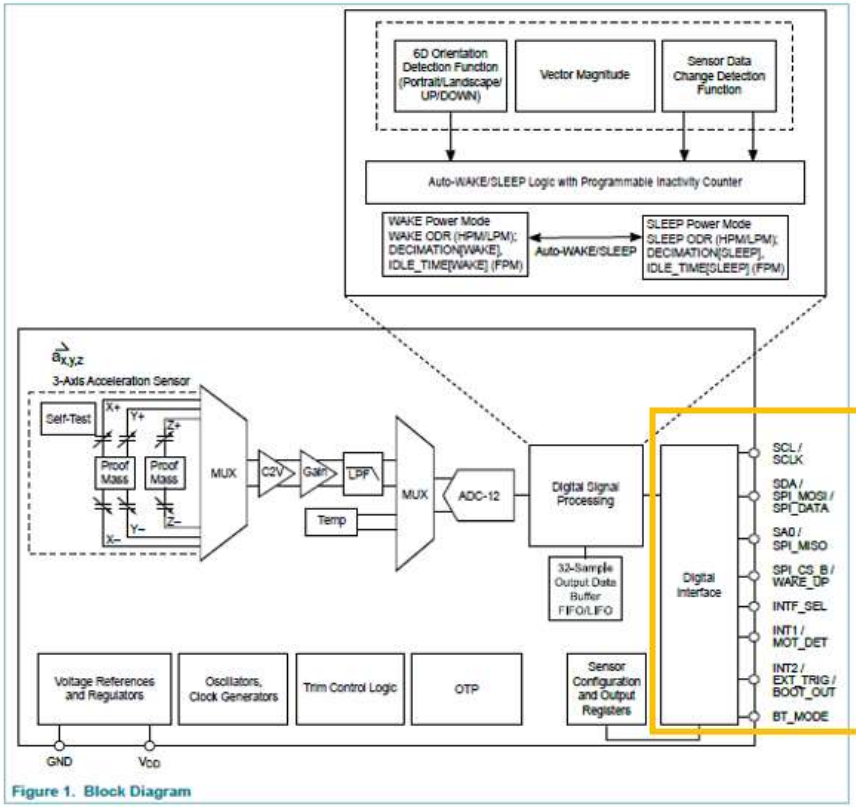
Table 16. Register read response message bit field descriptions

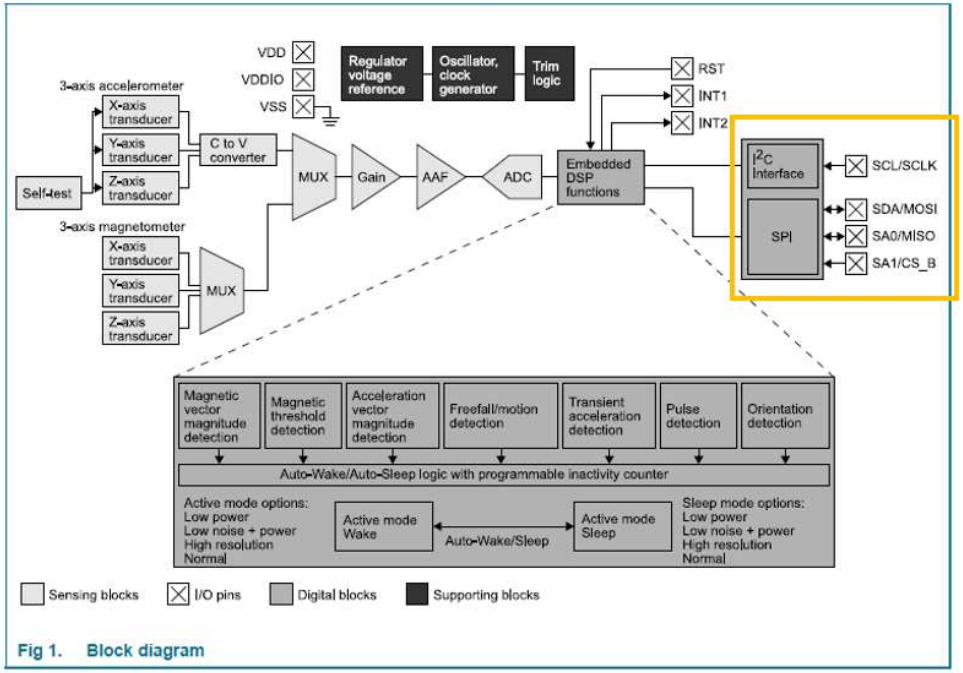
Bit field	Definition
C[0], [3:1]	Register Read Command = '0110'
ST[1:0]	Status
RD[15:8]	The contents of the register addressed by RA[7:1] high byte (RA[0] = 1)
RD[7:0]	The contents of the register addressed by RA[7:1] low byte (RA[0] = 0)
CRC[7:0]	8-bit CRC

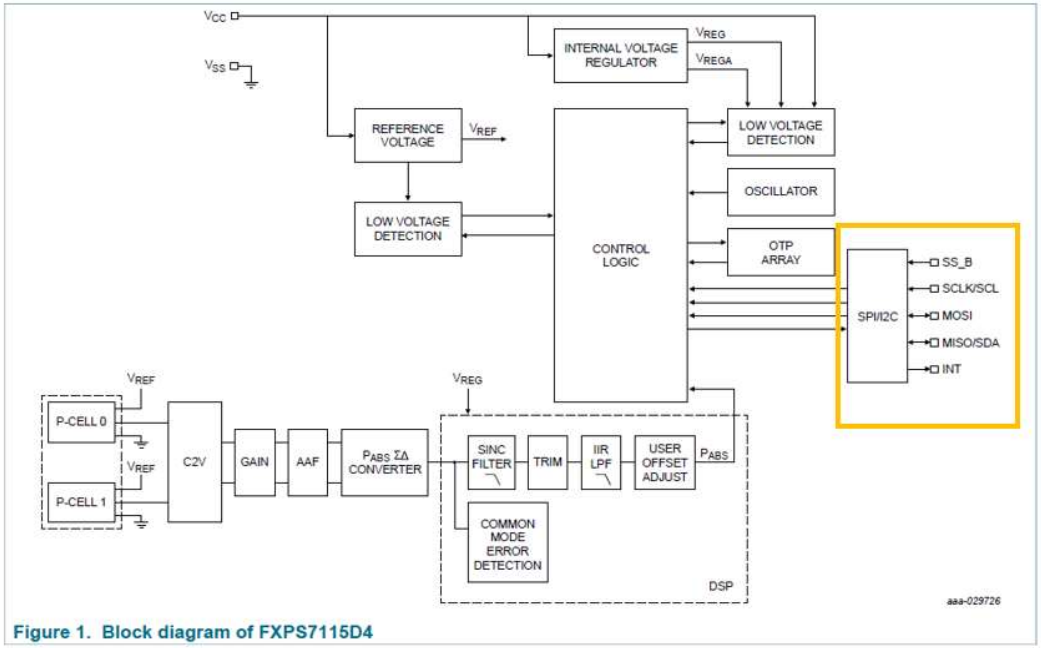
FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 21.

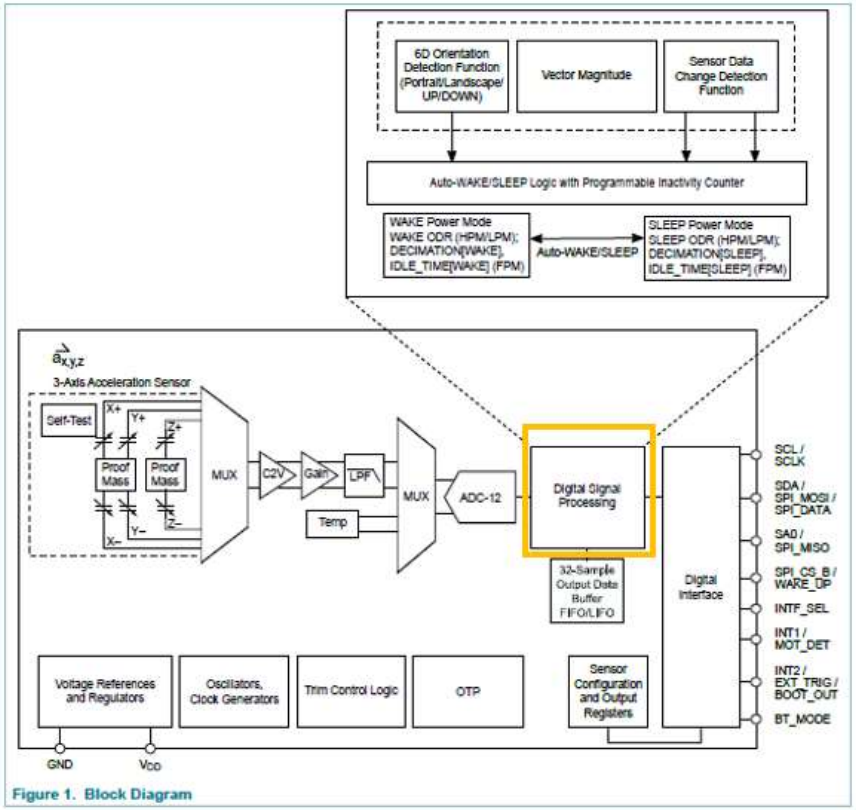
’474 Patent Claim	Representative NXP Product(s)
[14a.] A serial communication system comprising:	To the extent the preamble is limiting, the Accused ’474 Sensors include a “serial communication system” as recited in the ’474 patent. Exemplary systems and serial interfaces are identified in the block diagrams and rectangles below. *See, e.g.*:  https://www.avnet.com/shop/us/products/nxp/fxos8700cqr1-30744573456263135 37?fromPage=autoSuggest&langId=-1&autoSuggestSearchTerm=FXOS

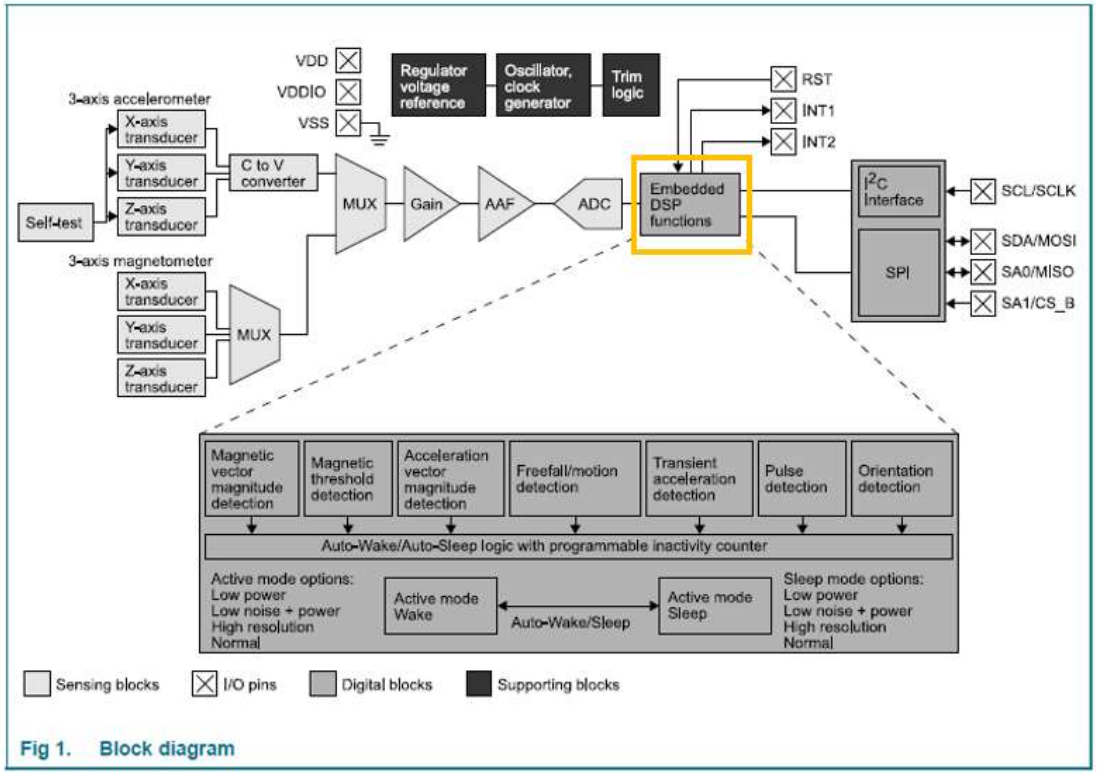
’474 Patent Claim	Representative NXP Product(s)
	 SLN-RPK-NODE (development board of FXOS8700CQ) (Arrow)  SLN-RPK-NODE (development board of FXOS8700CQ) (Mouser)

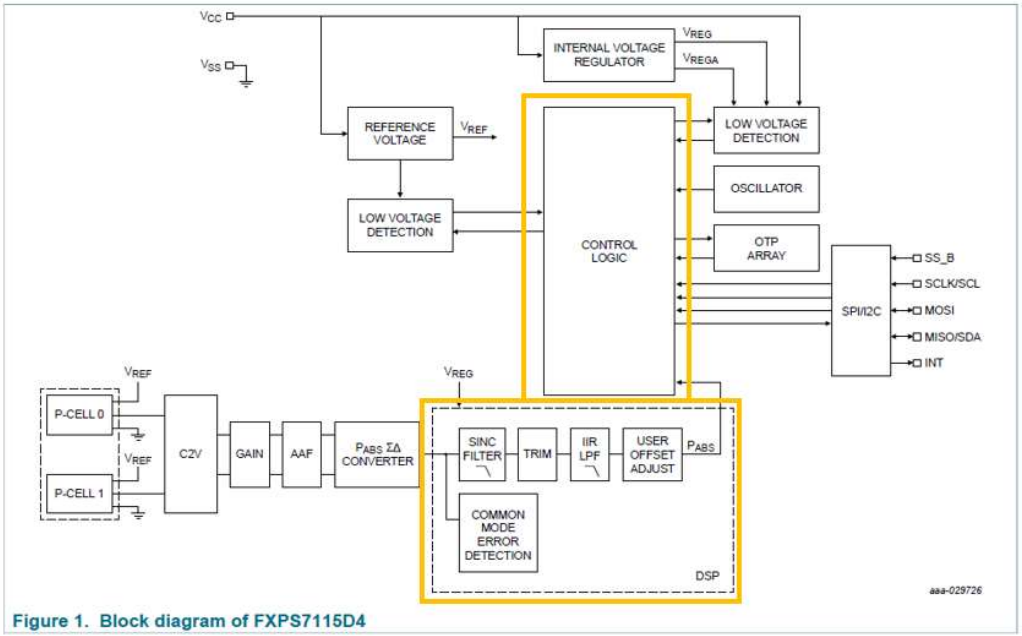
’474 Patent Claim	Representative NXP Product(s)
	NXP Semiconductors **FXLS8962AF** 3-Axis Low-g Accelerometer **5 Block diagram**  Figure 1. Block Diagram FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 3.

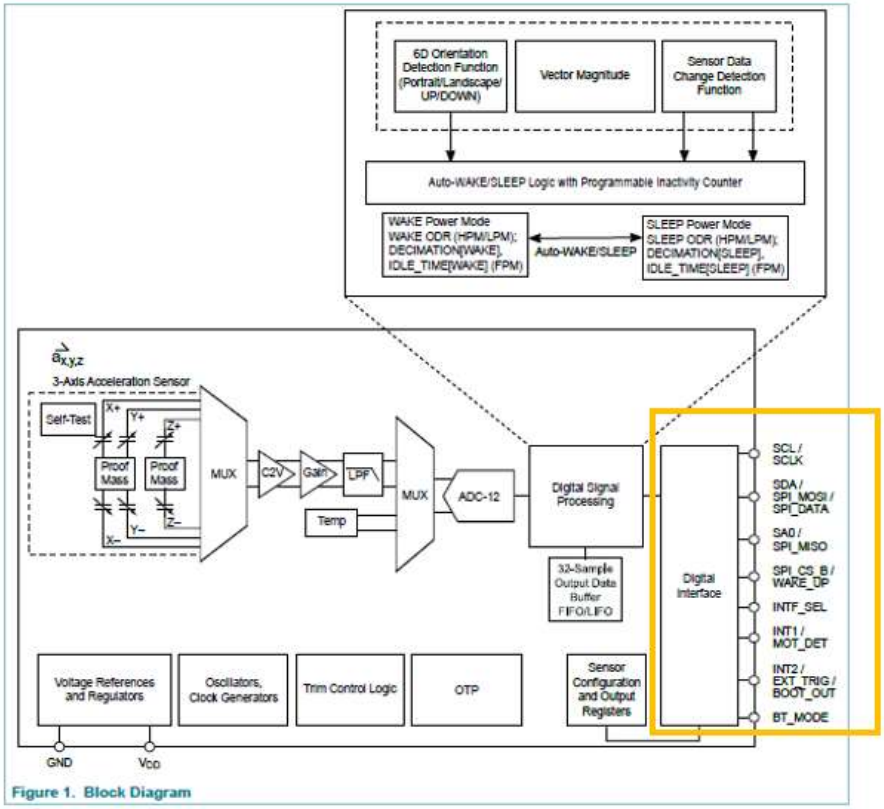
’474 Patent Claim	Representative NXP Product(s)
	NXP Semiconductors **FXOS8700CQ** 6-axis sensor with integrated linear accelerometer and magnetometer **5. Block diagram**  Fig 1. Block diagram FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 3.

’474 Patent Claim	Representative NXP Product(s)
	NXP Semiconductors **FXPS7115D4** Digital absolute pressure sensor, 40 kPa to 115 kPa **5 Block diagram**  Figure 1. Block diagram of FXPS7115D4 FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 3.
[14b.] a microprocessor having	The Accused ’474 Sensors each includes a microprocessor. For example, the Accused ’474 Sensors each constitutes a microprocessor because, among others, they

’474 Patent Claim	Representative NXP Product(s)
	include at least the processing logics identified below. *See, e.g.*: NXP Semiconductors **FXLS8962AF** 3-Axis Low-g Accelerometer **5 Block diagram**  Figure 1. Block Diagram FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 3.

’474 Patent Claim	Representative NXP Product(s)
	NXP Semiconductors **FXOS8700CQ** 6-axis sensor with integrated linear accelerometer and magnetometer **5. Block diagram**  **Fig 1. Block diagram** FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 3.

’474 Patent Claim	Representative NXP Product(s)
	NXP Semiconductors **FXPS7115D4** Digital absolute pressure sensor, 40 kPa to 115 kPa **5 Block diagram**  Figure 1. Block diagram of FXPS7115D4 FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 3.
[14c.] a slave serial interface for coupling to a master serial interface through a clock signal line output terminal and a data	In each of the Accused ’474 Sensors, the microprocessor has a slave serial interface for coupling to a master serial interface through a clock signal line output terminal and a data signal line output terminal. For example, each of the Accused ’474 Sensors has a slave serial interface (*e.g.*, SPI/I2C) for coupling to a master serial interface (*e.g.*, the SPI/I2C interface on the master/MCU and/or the host processor)

’474 Patent Claim	Representative NXP Product(s)
signal line output terminal	through a clock signal line output terminal (*e.g.*, the serial clock line (SCL) and/or the SPI clock (SCLK)) and a data signal line output terminal (*e.g.*, the serial data line (SDA) and/or the SPI master serial data out slave serial data in (MOSI)). NXP Semiconductors **FXLS8962AF** 3-Axis Low-g Accelerometer **5 Block diagram**  Figure 1. Block Diagram FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 3.

’474 Patent Claim	Representative NXP Product(s)

10.1.2 I^2C read/write operations

10.1.2.1 Single byte read

The master transmits a start condition (ST) to FXLS8962AF, followed by the slave address, with the R/W bit set to '0' for a write, and the FXLS8962AF sends an acknowledgement. Then the master transmits the address of the register to read and the FXLS8962AF sends an acknowledgement. The master transmits a repeated start condition (SR), followed by the slave address with the R/W bit set to '1' for a read from the previously selected register. The FXLS8962AF then acknowledges and transmits the data from the requested register. The master does not acknowledge (NAK) the transmitted data and then transmits a stop condition to end the data transfer.

FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 18.

11 SPI interface

The SPI interface is a classical Master/Slave serial port. FXLS8962AF is always considered to be the slave device and thus never initiates communication with the host processor.

The SPI interface of FXLS8962AF is compatible with interface mode 00, corresponding to CPOL = 0 and CPHA = 0.

For CPOL = 0, the idle value of the clock is zero, and the active value of the clock is 1. For CPHA = 0, data is captured on the clock's rising edge (low to high transition) and data is propagated on the clock's falling edge (high to low transition).

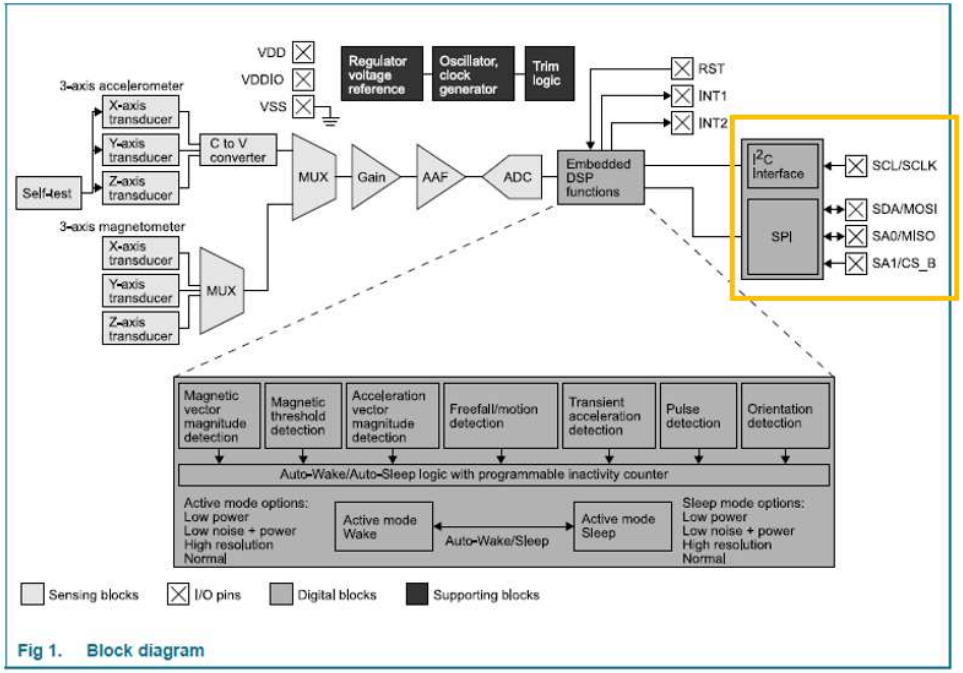
FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 20.

10.1.1 General I^2C operation

There are two signals associated with the I^2C-bus: the Serial Clock Line (SCL) and the Serial Data line (SDA). SDA is a bidirectional signal used for sending and receiving the data to/from the interface. External pull-up resistors connected to V_{DD} are required for SDA and SCL. When the I^2C-bus is free, SCL and SDA are high.

’474 Patent Claim	Representative NXP Product(s)
	FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 18. **11.1 General SPI operation** The SPI_CS_B pin is driven low at the start of a transaction, held low for the duration of the transfer, and then driven high again after the transaction is completed. During a transaction, the master toggles the clock (SCLK). The SCLK polarity is defined as having an idle value that is low, and an active phase that is high (CPOL = 0). Serial input and output data is captured on the clock's rising edge and propagated on the falling edge FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 20. **11.4 SPI read operations with 3-wire mode** FXLS8962AF can be configured to operate in 3-wire software enabled SPI mode. In this mode, the SPI_MISO pin is left unconnected and the SPI_MOSI pin becomes a bidirectional input/output pin (SPI_DATA). Read operations in 3-wire mode are the same as write operations in 3- and 4-wire modes, except that at the end of the address cycle (falling edge of clock pulse 16), the SPI_DATA pin automatically switches from an input to an output and with bit D7 as the current output state. FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 22.

’474 Patent Claim	Representative NXP Product(s)
	SDA / SPI_MOSI / SPI_DATA | 4 | Mode dependent Multifunction serial interface pin.[2] INTF_SEL = V_{DD}: • SPI_MOSI: In 4-wire SPI mode this pin functions as the serial data input (Master Out Slave In). • SPI_DATA[3]: In 3-wire SPI mode this pin functions as the bidirectional serial data input/output. INTF_SEL = GND: • SDA: This pin functions as the I^2C Serial Data input/output. SCL / SCLK | 5 | Mode dependent Multifunction serial interface pin.[2] INTF_SEL = V_{DD}: • SPI serial clock input (3- and 4-wire modes) INTF_SEL = GND: • I^2C serial clock input FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 4.

’474 Patent Claim	Representative NXP Product(s)
	NXP Semiconductors **FXOS8700CQ** 6-axis sensor with integrated linear accelerometer and magnetometer **5. Block diagram**  Fig 1. Block diagram FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 3.

’474 Patent Claim	Representative NXP Product(s)
	10.1.2 I²C read/write operations **Single-byte read** The master (or MCU) transmits a start condition (ST) to the FXOS8700CQ, followed by the slave address, with the R/W bit set to “0” for a write, and the FXOS8700CQ sends an acknowledgement. Then the master (or MCU) transmits the address of the register to read and the FXOS8700CQ sends an acknowledgement. The master (or MCU) transmits a repeated start condition (SR), followed by the slave address with the R/W bit set to “1” for a read from the previously selected register. The FXOS8700CQ then acknowledges and transmits the data from the requested register. The master does not acknowledge (NAK) the transmitted data, but transmits a stop condition to end the data transfer. FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 15. **10.1.1 General I²C operation** There are two signals associated with the I²C bus: the serial clock line (SCL) and the serial data line (SDA). The latter is a bidirectional line used for sending and receiving the data to/from the interface. External pullup resistors connected to VDDIO are required for SDA and SCL. When the bus is free both the lines are high. The I²C interface is compliant with fast mode (400 kHz), and normal mode (100 kHz) I²C standards. Operation at frequencies higher than 400 kHz is possible, but depends on several factors including the pullup resistor values, and total bus capacitance (trace + device capacitance). See Table 11 for more information. FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 14.

’474 Patent Claim	Representative NXP Product(s)

10.2.1 General SPI operation

NOTE

FXOS8700CQ only supports a point-to-point SPI protocol, with only one master (MCU) and one slave device (FXOS8700CQ) present on the bus. FXOS8700CQ does not tri-state the MISO pin when the CS_B pin is deasserted (logic high), which can lead to a bus conflict if multiple slave devices are present on the bus.

Do not connect more than one master and one slave device on the SPI bus.

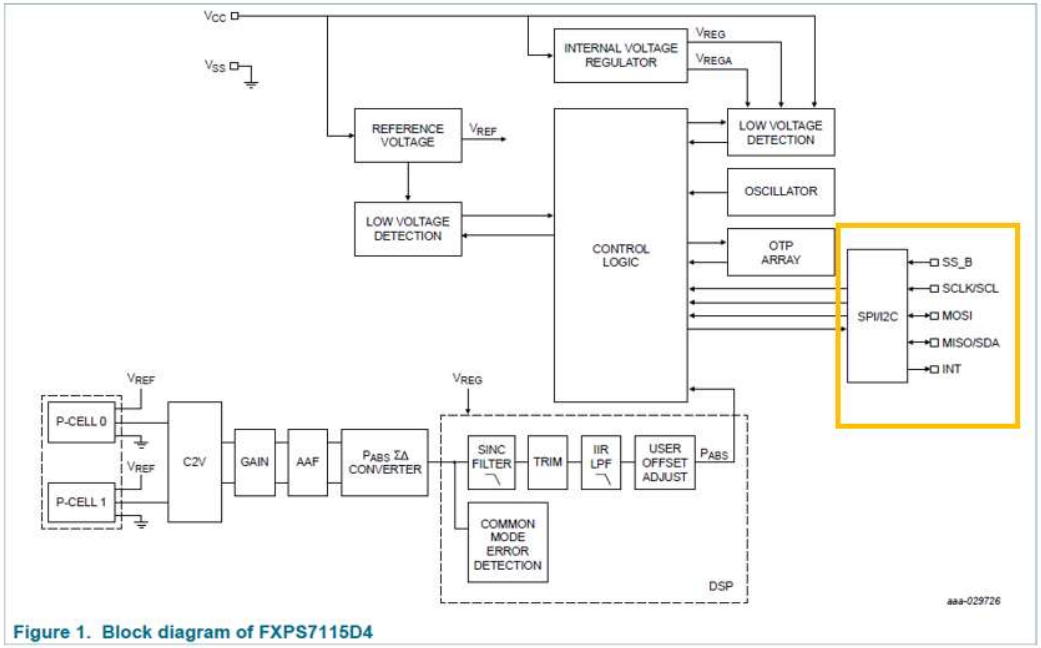
The CS_B pin is driven low at the start of a SPI transaction, held low for the duration of the transaction, and driven high after the transaction is complete. During a transaction the master toggles the SPI clock (SCLK) and transmits data on the MOSI pin.

FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 17.

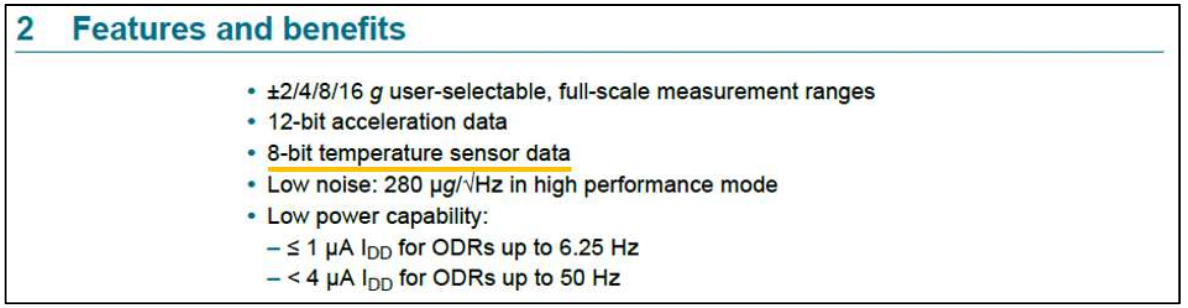
Table 13. Serial interface pin descriptions

Pin name	Pin description
VDDIO	Digital interface power
SA1/CS_B	I^2C second least significant bit of device address/SPI chip select
SCL/SCLK	I^2C/SPI serial clock
SDA/MOSI	I^2C serial data/SPI master serial data out slave serial data in
SA0/MISO	I^2C least significant bit of the device address/SPI master serial data in slave out

FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 18.

’474 Patent Claim	Representative NXP Product(s)
	NXP Semiconductors **FXPS7115D4** Digital absolute pressure sensor, 40 kPa to 115 kPa **5 Block diagram**  Figure 1. Block diagram of FXPS7115D4 FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 3.

’474 Patent Claim	Representative NXP Product(s)
	7.4.1 I^2C bit transmissions The state of SDA when SCL is high determines the bit value being transmitted. SDA must be stable when SCL is high and change when SCL is low as shown in Figure 14. After the START signal has been transmitted by the master, the bus is considered busy. Timing for the start condition is specified in Table 105. FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 13. **7.5 Standard 32-bit SPI protocol** The device includes a standard SPI protocol requiring 32-bit data packets. The device is a slave device and requires that the base clock value be low (CPOL = 0) with data captured on the rising edge of the clock and data propagated on the falling edge of the clock (CPHA = 0). The most significant bit is transferred first (MSB first). SPI transfers are completed through a sequence of two phases. During the first phase, the command is transmitted from the SPI master to the device. During the second phase, response data is transmitted from the slave device. MOSI and SCLK transitions are ignored when SS_B is not asserted. FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 17. 9 — SCLK/SCL — In I^2C mode, input pin 9 provides the serial clock. This pin must be connected to V_{CC} with an external pull-up resistor, as shown in the application diagram. In SPI mode, input pin 9 provides the serial clock. An internal pull-down device is connected to this pin. 10 — MOSI — SPI data in. In SPI mode, pin 10 functions as the serial data input to the SPI port. An internal pull-down device is connected to this pin. FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 4.
[14d.] wherein the slave serial interface is	In each of the Accused ’474 Sensors, the slave serial interface is responsive to a read temperature

’474 Patent Claim	Representative NXP Product(s)
responsive to a read temperature command issued by the master serial interface to return to the master serial interface a temperature value associated with the microprocessor.	command issued by the master serial interface to return to the master serial interface a temperature value associated with the microprocessor. For example, in each of the Accused ’474 Sensors, the slave serial interface (*e.g.*, SPI/I2C identified above) is responsive to a read temperature command (*e.g.*, the read temperature command directed to the TEMP_OUT register) issued by the master serial interface (*e.g.*, the SPI/I2C interface on the master/MCU and/or the host processor identified above) to return to the master serial interface a temperature value (*e.g.*, the temperature value in the TEMP_OUT register) associated with the microprocessor identified above.  **2 Features and benefits** • ±2/4/8/16 *g* user-selectable, full-scale measurement ranges • 12-bit acceleration data • 8-bit temperature sensor data • Low noise: 280 µ*g*/√Hz in high performance mode • Low power capability: – ≤ 1 µA I_{DD} for ODRs up to 6.25 Hz – < 4 µA I_{DD} for ODRs up to 50 Hz FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 1.

’474 Patent Claim	Representative NXP Product(s)

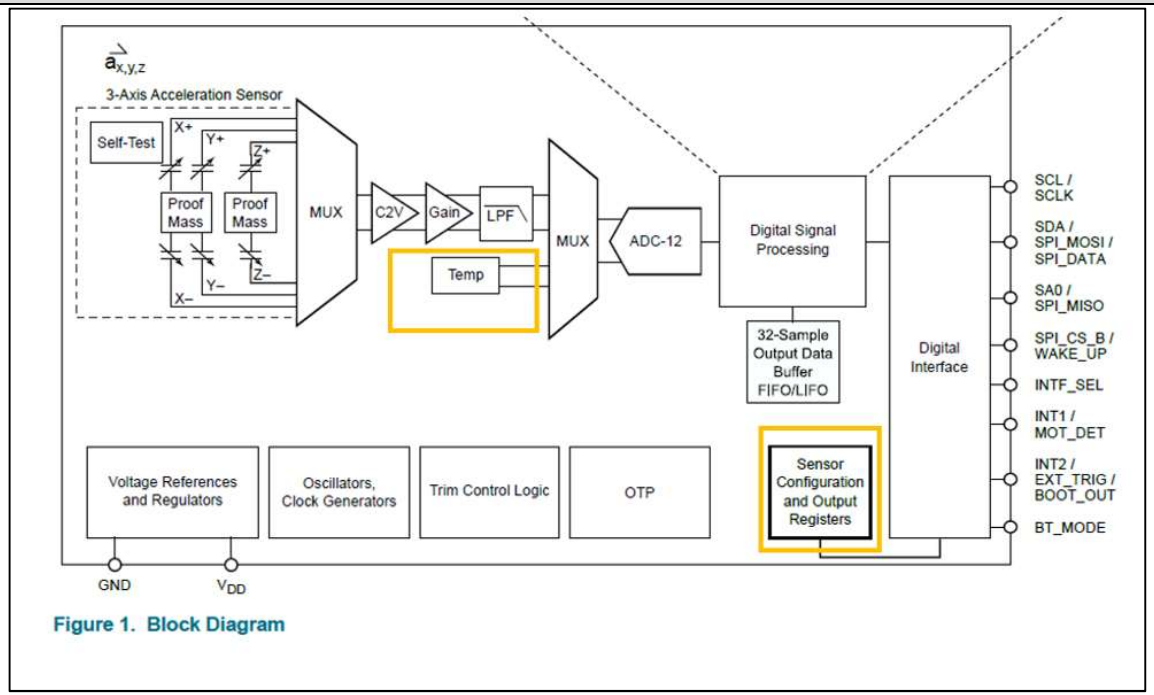


Figure 1. Block Diagram

FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 3.

13.2 TEMP_OUT register (address 01h)

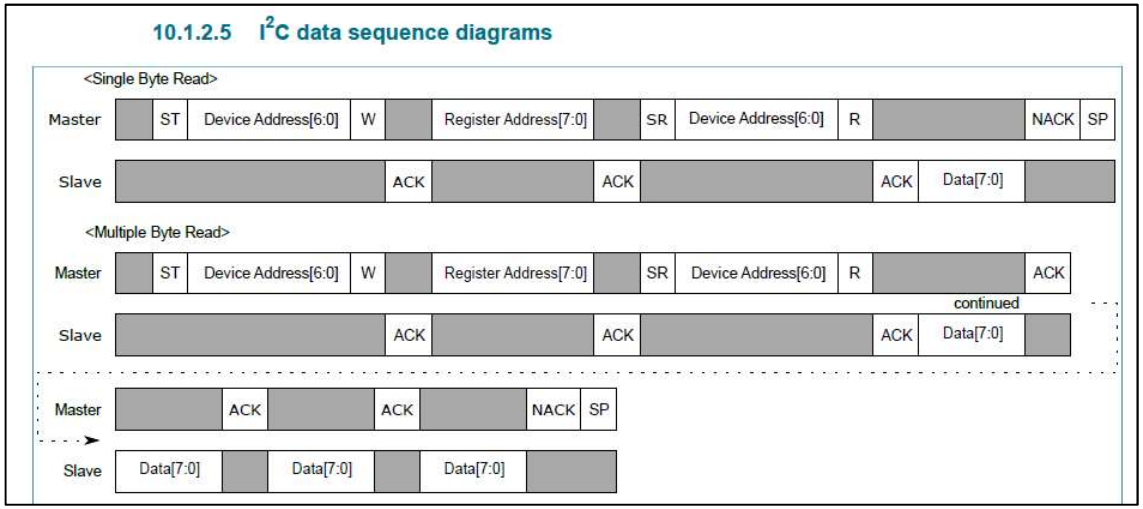
Table 22. TEMP_OUT register (address 01h) bit allocation

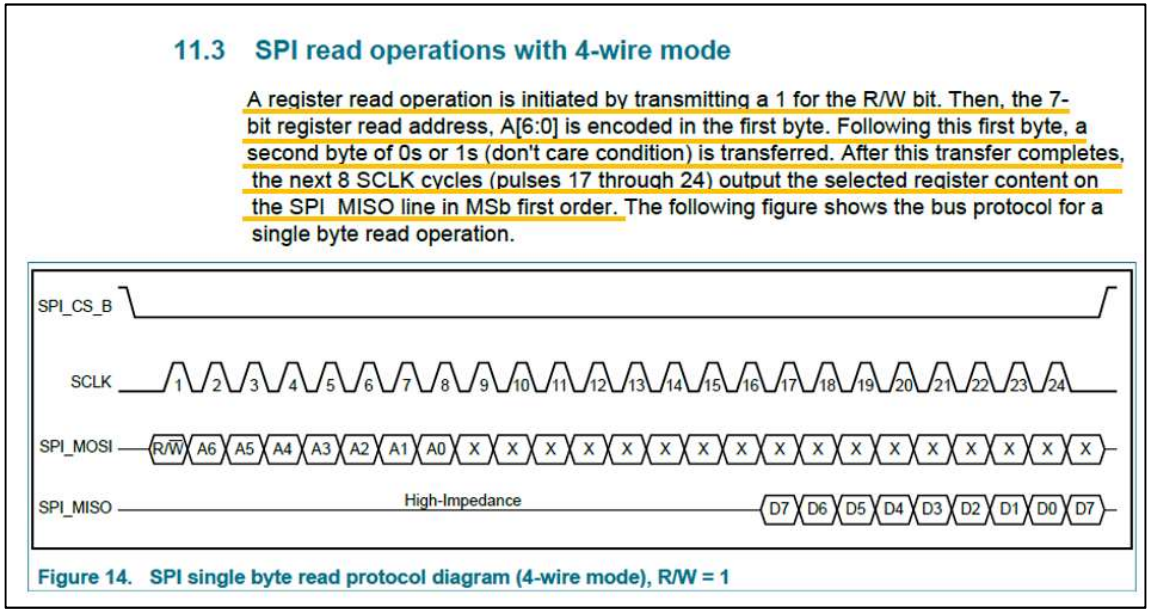
Bit	7	6	5	4	3	2	1	0
Name	TEMP_OUT[7:0]							
Reset	—	—	—	—	—	—	—	—
Access	R	R	R	R	R	R	R	R

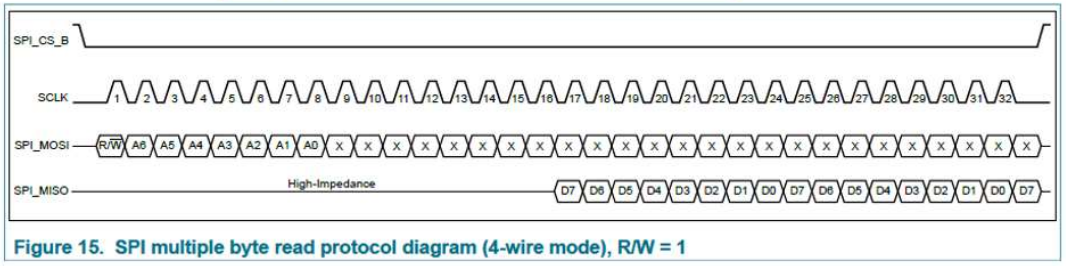
The **TEMP_OUT** register contains the 8-bit, 2's complement temperature value. When this register contains the value 00h, the measured temperature is 25 °C (typ). This register is updated on every ODR cycle.

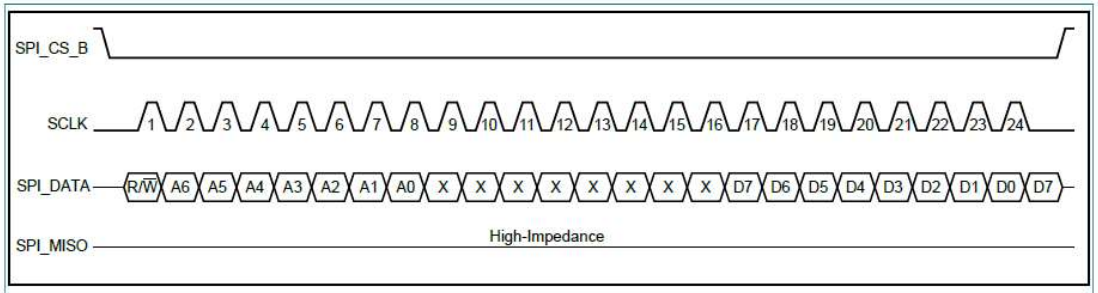
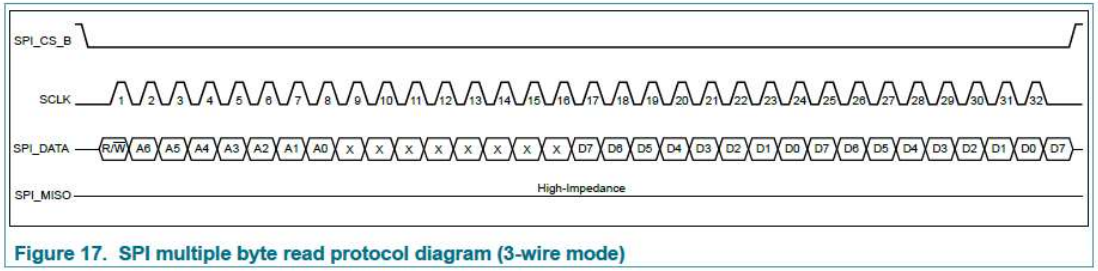
FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 3.

’474 Patent Claim	Representative NXP Product(s)
	10 I^2C digital interface The registers embedded within FXLS8962AF may be accessed using an I^2C interface when the INTF_SEL pin is tied to GND. If the V_{DD} supply is not present, the device will be in shutdown mode and any communications on the interface are ignored. When the device is on a common I^2C-bus with other slave devices, the V_{DD} supply pin must be left unconnected (high-impedance) when the device supply is turned off to ensure that the internal ESD protection diodes do not become forward biased and prevent the bus from functioning normally (clamping). FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 16. **10.1.2 I^2C read/write operations** **10.1.2.1 Single byte read** The master transmits a start condition (ST) to FXLS8962AF, followed by the slave address, with the R/W bit set to '0' for a write, and the FXLS8962AF sends an acknowledgement. Then the master transmits the address of the register to read and the FXLS8962AF sends an acknowledgement. The master transmits a repeated start condition (SR), followed by the slave address with the R/W bit set to '1' for a read from the previously selected register. The FXLS8962AF then acknowledges and transmits the data from the requested register. The master does not acknowledge (NAK) the transmitted data and then transmits a stop condition to end the data transfer. **10.1.2.2 Multiple byte read** When performing a multi-byte or *burst* read, FXLS8962AF automatically increments the register read address pointer after a read command is received. Therefore, after following the steps of a single-byte read, multiple bytes of data can be read from sequential register addresses after each FXLS8962AF acknowledgment (ACK) is received until a no acknowledge (NAK) is issued by the master followed by a stop condition (SP) signaling the end of the transfer. FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 18-19.

’474 Patent Claim	Representative NXP Product(s)
	 FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 18-20. **11.1 General SPI operation** The SPI_CS_B pin is driven low at the start of a transaction, held low for the duration of the transfer, and then driven high again after the transaction is completed. During a transaction, the master toggles the clock (SCLK). The SCLK polarity is defined as having an idle value that is low, and an active phase that is high (CPOL = 0). Serial input and output data is captured on the clock's rising edge and propagated on the falling edge (CPHA = 0). Single byte read and single byte write operations are completed in 24 SCLK cycles; multiple byte reads and writes are completed in additional multiples of 8 SCLK cycles per read or written byte. The first SCLK cycle latches the R/W (Read/Write) bit to select whether the desired operation is a read (R/W = 1) or a write (R/W = 0). The following seven SCLK cycles are used to latch the register read or write address. ***Note:*** *4-wire SPI interface mode is the default out of POR or after a soft reset. The 3-wire interface mode may also be selected by setting* ***SENS_CONFIG1[****SPI_M****]*** *= 1.*

’474 Patent Claim	Representative NXP Product(s)
	FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 18-20-21.  FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 18-21-22.

’474 Patent Claim	Representative NXP Product(s)
	Multiple-byte read operations are performed similarly to single-byte reads with additional bytes read out in multiples of eight SCLK cycles. The register read address is auto-incremented by FXLS8962AF so that every eighth clock edge will latch the address of the next sequential register read address. When the desired number of bytes has been read, a rising edge on SPI_CS_B terminates the transaction.  Figure 15. SPI multiple byte read protocol diagram (4-wire mode), R/W = 1 FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 18-22. 11.4 SPI read operations with 3-wire mode FXLS8962AF can be configured to operate in 3-wire software enabled SPI mode. In this mode, the SPI_MISO pin is left unconnected and the SPI_MOSI pin becomes a bidirectional input/output pin (SPI_DATA). Read operations in 3-wire mode are the same as write operations in 3- and 4-wire modes, except that at the end of the address cycle (falling edge of clock pulse 16), the SPI_DATA pin automatically switches from an input to an output and with bit D7 as the current output state. FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 18-22.

’474 Patent Claim	Representative NXP Product(s)
	 **Figure 16. SPI single byte read protocol diagram (3-wire mode)**  **Figure 17. SPI multiple byte read protocol diagram (3-wire mode)** FXLS8962AF, 3-Axis Low-g Accelerometer, Product data sheet (Rev. 5.3 — 10 December 2019) at 18-23.

’474 Patent Claim	Representative NXP Product(s)

14.3 Temperature register

14.3.1 TEMP register (address 0x51)

Eight-bit 2's complement sensor temperature value with 0.96 °C/LSB sensitivity. Temperature data is only valid between –40 °C and 125 °C. The temperature sensor output is only valid when M_CTRL_REG1[m_hms] > 0b00. Please note that the temperature sensor is uncalibrated and its output for a given temperature will vary from one device to the next.

Table 50. TEMP register (address 0x51) bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	die_temperature[7:0]							
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

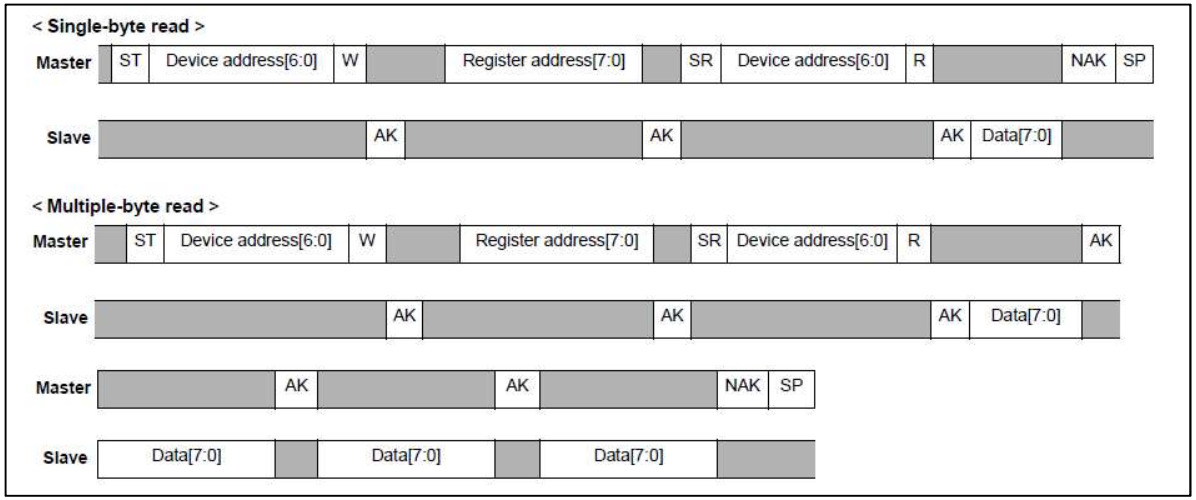
FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 52.

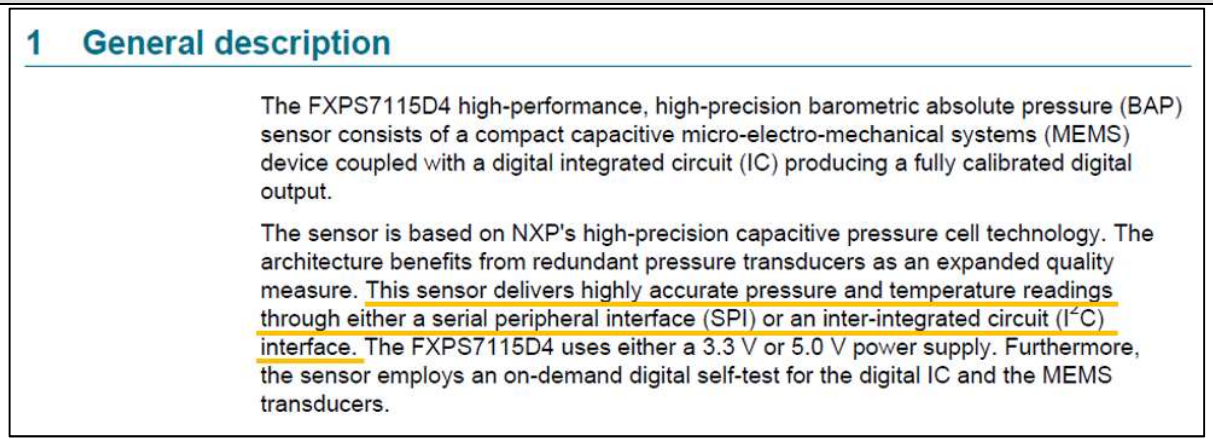
10.1.2 I²C read/write operations

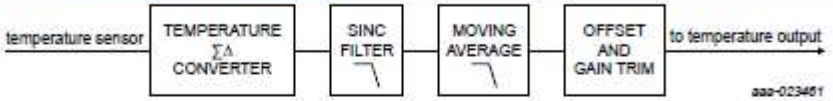
Single-byte read

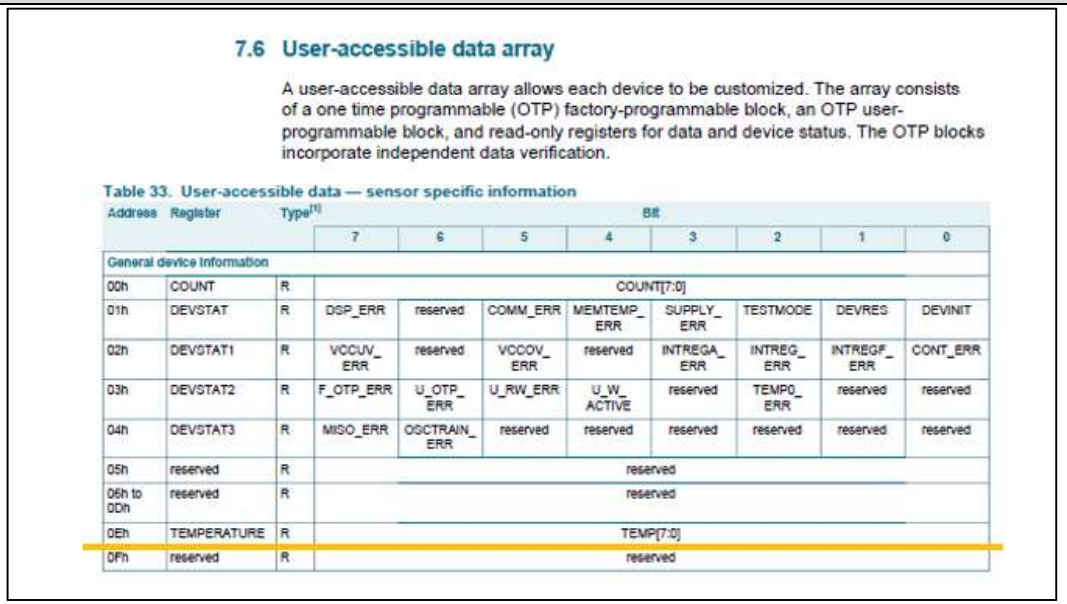
The master (or MCU) transmits a start condition (ST) to the FXOS8700CQ, followed by the slave address, with the R/W bit set to “0” for a write, and the FXOS8700CQ sends an acknowledgement. Then the master (or MCU) transmits the address of the register to read and the FXOS8700CQ sends an acknowledgement. The master (or MCU) transmits a repeated start condition (SR), followed by the slave address with the R/W bit set to “1” for a read from the previously selected register. The FXOS8700CQ then acknowledges and transmits the data from the requested register. The master does not acknowledge (NAK) the transmitted data, but transmits a stop condition to end the data transfer.

When performing a multi-byte or “burst” read, the FXOS8700CQ automatically increments the register address read pointer after a read command is received. Therefore, after following the steps of a single-byte read, multiple bytes of data can be read from sequential registers after each FXOS8700CQ acknowledgment (AK) is received until a no acknowledge (NAK) occurs from the master followed by a stop condition (SP) signaling an end of transmission.

’474 Patent Claim	Representative NXP Product(s)
	FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 15.  FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 16. **10.2.2 SPI read/write operations** A read operation is initiated by transmitting a 0 for the R/W bit. Then the 8-bit register address, ADDR[7:0] is encoded in the first and second serialized bytes. Subsequent bits are ignored by the part. The read data is deserialized from the MISO pin. FXOS8700CQ, 6-axis sensor with integrated linear accelerometer and magnetometer, Data sheet: Technical data (Rev. 8 — 25 April 2017) at 18.

’474 Patent Claim	Representative NXP Product(s)
	 1 General description The FXPS7115D4 high-performance, high-precision barometric absolute pressure (BAP) sensor consists of a compact capacitive micro-electro-mechanical systems (MEMS) device coupled with a digital integrated circuit (IC) producing a fully calibrated digital output. The sensor is based on NXP's high-precision capacitive pressure cell technology. The architecture benefits from redundant pressure transducers as an expanded quality measure. This sensor delivers highly accurate pressure and temperature readings through either a serial peripheral interface (SPI) or an inter-integrated circuit (I^2C) interface. The FXPS7115D4 uses either a 3.3 V or 5.0 V power supply. Furthermore, the sensor employs an on-demand digital self-test for the digital IC and the MEMS transducers. FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 1.

’474 Patent Claim	Representative NXP Product(s)
	7.3.4 Temperature sensor 7.3.4.1 Temperature sensor signal chain The device includes a temperature sensor for signal compensation and user readability. Figure 12 shows a simplified block diagram. Temperature sensor parameters are specified in Table 104 and Table 105.  Figure 12. Temperature sensor signal chain block diagram 7.3.4.2 Temperature sensor output scaling equation Equation 5 is used to convert temperature readings with the variables specified in Table 8. $$T_{DEGC} = \frac{T_{LSB} - T0_{LSB}}{T_{SENSE}} \quad (5)$$ where: T_{DEGC} = The temperature output in degrees C T_{LSB} = The temperature output in LSB $T0_{LSB}$ = The expected temperature output in LSB at 0 °C T_{SENSE} = The expected temperature sensitivity in LSB/°C Table 8. Temperature conversion variables Data reading | $T0_{LSB}$ (LSB) | T_{SENSE} LSB/C) 8-bit register read | 68 | 1 FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 12.

’474 Patent Claim	Representative NXP Product(s)
	

7.6 User-accessible data array

A user-accessible data array allows each device to be customized. The array consists of a one time programmable (OTP) factory-programmable block, an OTP user-programmable block, and read-only registers for data and device status. The OTP blocks incorporate independent data verification.

Table 33. User-accessible data — sensor specific information

Address	Register	Type[1]	Bit							
			7	6	5	4	3	2	1	0
General device information										
00h	COUNT	R	COUNT[7:0]							
01h	DEVSTAT	R	DSP_ERR	reserved	COMM_ERR	MEMTEMP_ERR	SUPPLY_ERR	TESTMODE	DEVRES	DEVINIT
02h	DEVSTAT1	R	VCCUV_ERR	reserved	VCCOV_ERR	reserved	INTREGA_ERR	INTREG_ERR	INTREGF_ERR	CONT_ERR
03h	DEVSTAT2	R	F_OTP_ERR	U_OTP_ERR	U_RW_ERR	U_W_ACTIVE	reserved	TEMP0_ERR	reserved	reserved
04h	DEVSTAT3	R	MISO_ERR	OSCTRAIN_ERR	reserved	reserved	reserved	reserved	reserved	reserved
05h	reserved	R	reserved							
06h to 0Dh	reserved	R	reserved							
0Eh	TEMPERATURE	R	TEMP[7:0]							
0Fh	reserved	R	reserved							

FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 27.

’474 Patent Claim	Representative NXP Product(s)
	7.4.6.2 Register read transfers The device supports I^2C register read data transfers. Register read data transfers are constructed as follows: 1. The master transmits a START condition. 2. The master transmits the 7-bit slave address. 3. The master transmits a '0' for the read/write bit to indicate a write operation. 4. The slave transmits an ACK. 5. The master transmits the register address to be read. 6. The slave transmits an ACK. 7. The master transmits a repeat START condition. 8. The master transmits the 7-bit slave address. 9. The master transmits a '1' for the read/write bit to indicate a read operation. 10. The slave transmits an ACK. 11. The slave transmits the data from the register addressed. 12. The master transmits a NACK. 13. The master transmits a STOP condition.  FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 16.

’474 Patent Claim	Representative NXP Product(s)
	7.5.3 Command summary **7.5.3.1 Register read command** The device supports a register read command. The register read command uses the upper 7 bits of the addresses defined in Section 7.6 "User-accessible data array" to address 8-bit registers in the register map. The response to a register read command is shown in Section 7.5.3.1.2 "Register read response message format". The response is transmitted on the next SPI message if and only if all of the following conditions are met: • No SPI error is detected (see Section 7.5.5.3 "SPI error") • No MISO error is detected (see Section 7.5.5.4 "SPI data output verification error") If these conditions are met, the device responds to the register read request as shown in Section 7.5.3.1.2 "Register read response message format". Otherwise, the device responds with the error response as defined in Section 7.5.5.2 "Detailed status field". The register read response includes the register contents at the rising edge of SS_B for the register read command. FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 20.

’474 Patent Claim	Representative NXP Product(s)

7.5.3.1.1 Register read command message format

Table 13. Register read command message format

MSB: bit 31; LSB: bit 0

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
Register access command																															
Command C[3:0]				Fixed bits: must = 0h				Register address								Register data								8-bit CRC							
1	1	0	0	0	0	0	0	RA[7:1]							RA[0]	0	0	0	0	0	0	0	0	CRC[7:0]							

Table 14. Register read command message bit field descriptions

Bit field	Definition
C[3:0]	Register read command = '1100'
RA[7:0]	RA[7:1] contains the word address of the register to be read.
CRC[7:0]	Read CRC Section

7.5.3.1.2 Register read response message format

Table 15. Register read response message format

MSB: bit 31; LSB: bit 0

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
Register access command																															
Command C[0], [3:1]				Basic Status		Unused Data 0h		Register data: contents of RA[7:1] high byte								Register data: contents of RA[7:1] low byte								8-bit CRC							
0	1	1	0	ST[1:0]		0	0	RD[15:8]								RD[7:0]								CRC[7:0]							

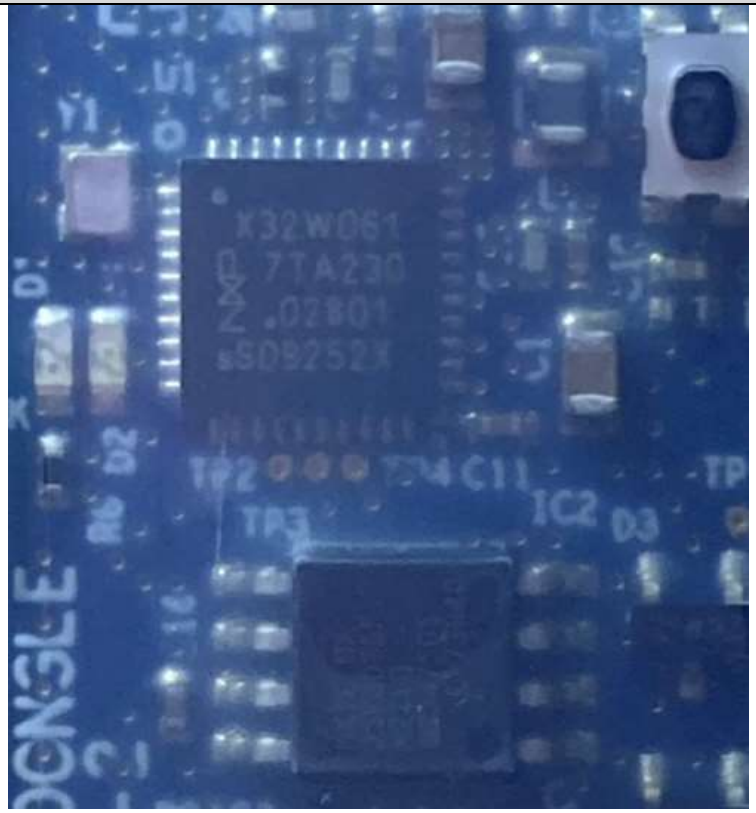
FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 20.

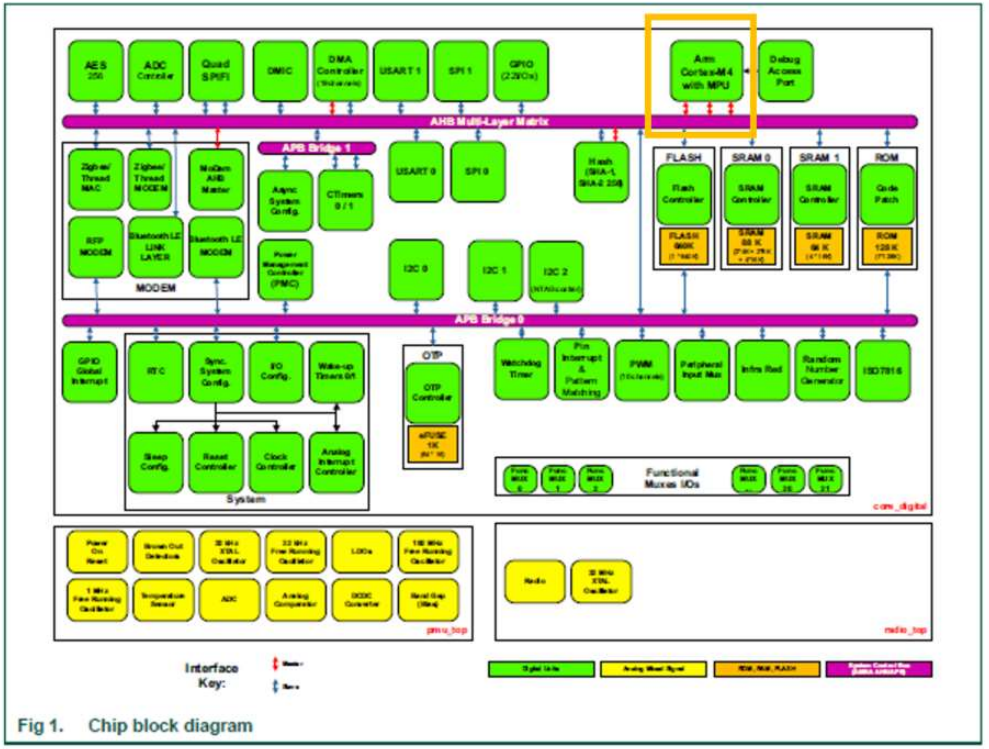
Table 16. Register read response message bit field descriptions

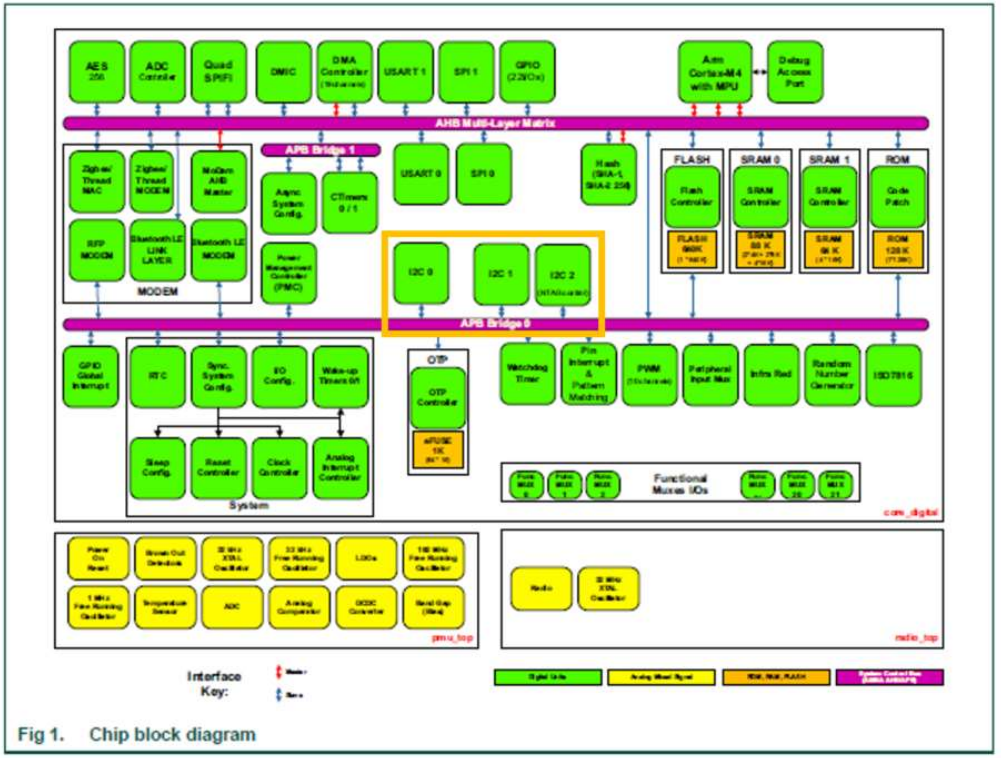
Bit field	Definition
C[0], [3:1]	Register Read Command = '0110'
ST[1:0]	Status
RD[15:8]	The contents of the register addressed by RA[7:1] high byte (RA[0] = 1)
RD[7:0]	The contents of the register addressed by RA[7:1] low byte (RA[0] = 0)
CRC[7:0]	8-bit CRC

FXPS7115D4, Digital absolute pressure sensor, 40 kPa to 115 kPa, Product data sheet (Rev. 4 — 17 July 2020) at 21.

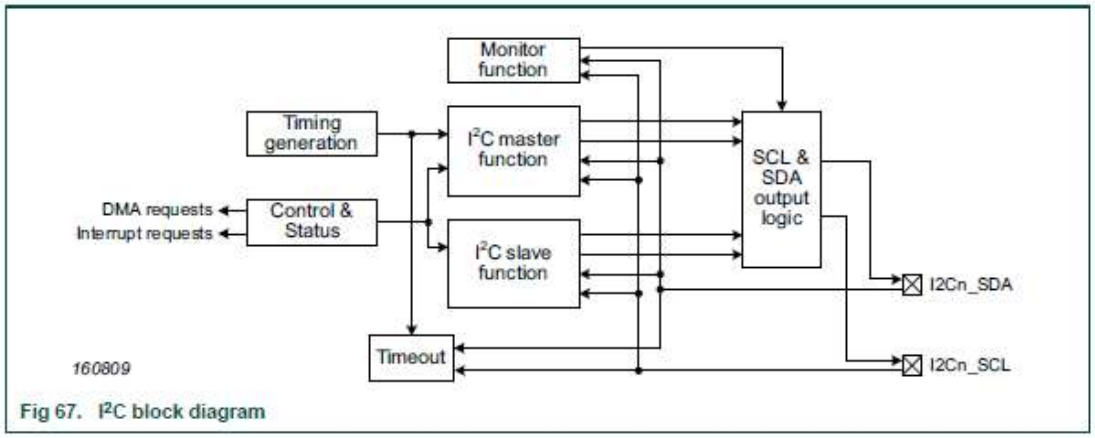
’474 Patent Claim	Representative NXP Product(s)
[1a.] A serial communication system comprising:	To the extent the preamble is limiting, the Accused ’474 Wireless Microcontrollers include a “serial communication system” as recited in the ’474 patent. Exemplary systems and serial interfaces are identified in the block diagrams below. *See, e.g.,*:  https://www.nxp.com/docs/en/data-sheet/K32W061.pdf *See also* https://www.avnet.com/shop/us/search/k32w041  OM15080-K32W (Development Board of K32W061/41) https://www.arrow.com/en/products/om15080-k32w/nxp-semiconductors?q=OM15080-K32W

’474 Patent Claim	Representative NXP Product(s)
	 OM15080-K32W (Development Board of K32W061/41) (Mouser)

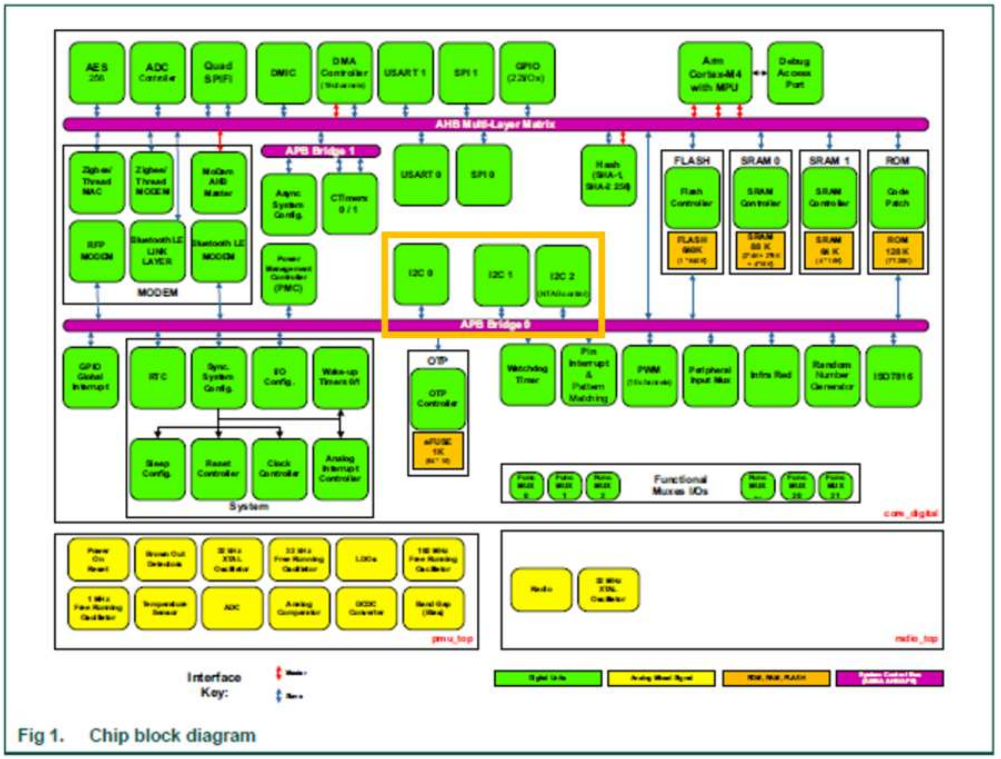
’474 Patent Claim	Representative NXP Product(s)
	1.3 Block diagram  Fig 1. Chip block diagram K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 6.
[1b.] an integrated circuit having a master serial interface; and	The Accused ’474 Wireless Microcontrollers each includes an integrated circuit having a master serial interface. For example, the Accused ’474 Wireless Microcontrollers each includes an integrated circuit having a master serial interface (*e.g.*, the I2C interface on the master/MCU and/or the host processor).

’474 Patent Claim	Representative NXP Product(s)
	See, e.g.: 1.3 Block diagram  Fig 1. Chip block diagram K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 6.

’474 Patent Claim	Representative NXP Product(s)
	25.2 Features • Independent Master, Slave, and Monitor functions. • Bus speeds supported: – Standard mode, up to 100 kbits/s. – Fast-mode, up to 400 kbits/s. – Fast-mode Plus, up to 1 Mbits/s (on pins PIO0_10 and PIO0_11 that include specific I^2C support). – High speed mode, 3.4 Mbits/s as a Slave only (on pins PIO0_10 and PIO0_11 that include specific I^2C support). • Supports both Multi-master and Multi-master with Slave functions. • Multiple I^2C slave addresses supported in hardware. • One slave address can be selectively qualified with a bit mask or an address range in order to respond to multiple I^2C bus addresses. • 10-bit addressing supported with software assist. • Supports System Management Bus (SMBus). • Separate DMA requests for Master, Slave, and Monitor functions. • No chip clocks are required in order to receive and compare an address as a Slave, so this event can wake up the device from deep-sleep mode. Additionally, I^2C0 can optionally generate a wake-up from power down. • Automatic modes optionally allow less software overhead for some use cases. K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 170.

’474 Patent Claim	Representative NXP Product(s)
	 Fig 67. I²C block diagram K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 176. **25.4.2 I²C receive/transmit in slave mode** In this example, I²C1 is used as an I²C slave. The slave receives 8 bits from the master and then sends 8 bits to the master. The SCL and SDA functions must be enabled on suitable pins, see Table 48. The pins should be configured as required for the I²C-bus mode. The transmission of the address and data bits is controlled by the STAT[SLVPENDING] status bit. Whenever the status is Slave pending, the slave can acknowledge (“ack”) or send or receive an address and data. The received data or the data to be sent to the master are available in the SLVDAT register. After sending and receiving data, continue to the next step of the transmission protocol by writing to the SLVCTL register. K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 173.

’474 Patent Claim	Representative NXP Product(s)
	25.4.2.1 Slave read from master This example uses polling to control the sequence and does not use interrupts. Configure the I²C as a slave with address x: 1. Write the slave address x to the address 0 match register. 2. Set the CFG[SLVEN] bit to 1. K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 173.
[1c.] a processor having a slave serial interface coupled to the master serial interface through a clock signal line and a data signal line	The Accused ’474 Wireless Microcontrollers each includes a processor having a slave serial interface coupled to the master serial interface through a clock signal line and a data signal line. For example, the Accused ’474 Wireless Microcontrollers each includes a processor having a slave serial interface (*e.g.*, I2C) coupled to the master serial interface identified above through a clock signal line (*e.g.*, SCL) and a data signal line (*e.g.*, SDA).

’474 Patent Claim	Representative NXP Product(s)
	1.3 Block diagram  Fig 1. Chip block diagram K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 6.

’474 Patent Claim	Representative NXP Product(s)
	25.2 Features • Independent Master, Slave, and Monitor functions. • Bus speeds supported: – Standard mode, up to 100 kbits/s. – Fast-mode, up to 400 kbits/s. – Fast-mode Plus, up to 1 Mbits/s (on pins PIO0_10 and PIO0_11 that include specific I^2C support). – High speed mode, 3.4 Mbits/s as a Slave only (on pins PIO0_10 and PIO0_11 that include specific I^2C support). • Supports both Multi-master and Multi-master with Slave functions. • Multiple I^2C slave addresses supported in hardware. • One slave address can be selectively qualified with a bit mask or an address range in order to respond to multiple I^2C bus addresses. • 10-bit addressing supported with software assist. • Supports System Management Bus (SMBus). • Separate DMA requests for Master, Slave, and Monitor functions. • No chip clocks are required in order to receive and compare an address as a Slave, so this event can wake up the device from deep-sleep mode. Additionally, I^2C0 can optionally generate a wake-up from power down. • Automatic modes optionally allow less software overhead for some use cases. K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 170.

’474 Patent Claim	Representative NXP Product(s)

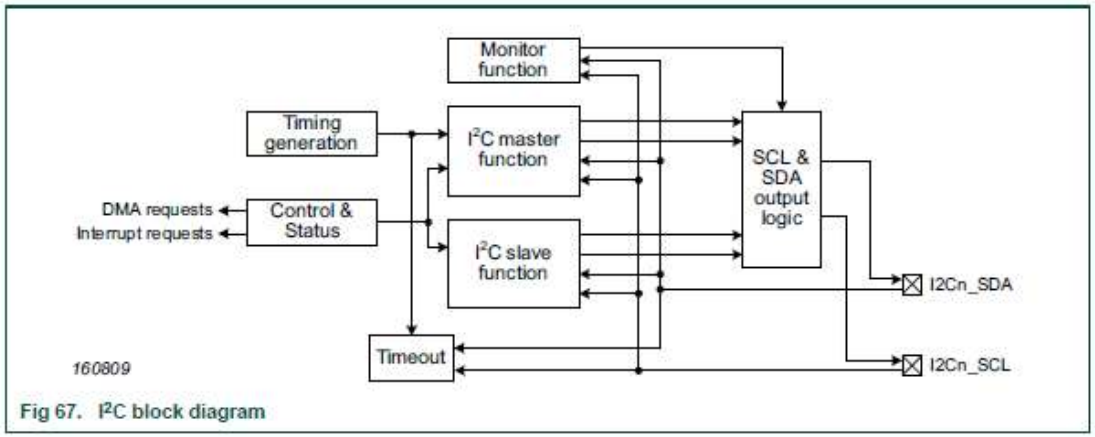


Fig 67. I²C block diagram

K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 176.

Table 46. I²C-bus pin description

Function	Type	Pin name used in data sheet Pin Description	Description
SCL	I/O	I2Cn_SCL	I²C serial clock
SDA	I/O	I2Cn_SDA	I²C serial data

K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 170.

Table 47. I²C bus pin assignments

Pin	Possible Pin Assignment	
I2C0_SCL	PIO0_10[1]	PIO0_15
I2C0_SDA	PIO0_11[1]	PIO0_16
I2C1_SCL	PIO0_6	PIO0_12
I2C1_SDA	PIO0_7	PIO0_13
I2C2_SCL	Internal connection	
I2C2_SDA	Internal connection	

K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 171.

’474 Patent Claim	Representative NXP Product(s)
	25.4.2 I²C receive/transmit in slave mode In this example, I²C1 is used as an I²C slave. The slave receives 8 bits from the master and then sends 8 bits to the master. The SCL and SDA functions must be enabled on suitable pins, see Table 48. The pins should be configured as required for the I²C-bus mode. The transmission of the address and data bits is controlled by the STAT[SLVPENDING] status bit. Whenever the status is Slave pending, the slave can acknowledge (“ack”) or send or receive an address and data. The received data or the data to be sent to the master are available in the SLVDAT register. After sending and receiving data, continue to the next step of the transmission protocol by writing to the SLVCTL register. K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 173. **25.4.2.1 Slave read from master** This example uses polling to control the sequence and does not use interrupts. Configure the I²C as a slave with address x: 1. Write the slave address x to the address 0 match register. 2. Set the CFG[SLVEN] bit to 1. K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 173.
[1d.] wherein the slave serial interface is responsive to a read temperature command issued by the master serial interface to return a temperature value associated with the processor.	In each of the Accused ’474 Wireless Microcontrollers, the slave serial interface is responsive to a read temperature command issued by the master serial interface to return a temperature value associated with the processor. For example, in each of the Accused ’474 Wireless Microcontrollers, the slave serial interface identified above is responsive to a read temperature command (*e.g.*, the read tempearture command requesting the result derived from the temperature sensor measurement) issued by the identified master serial interface to return a temperature value (*e.g.*, the result derived from the temperature sensor measurement) associated with the processor.

’474 Patent Claim	Representative NXP Product(s)
	25.4.2.2 Slave write to master This example uses polling to control the sequence and does not use interrupts. Configure the I^2C as a slave with address x: 1. Write the slave address x to the address 0 match register. 2. Set the CFG[SLVEN] bit to 1. Write data to the master: 1. Wait for the pending status to be set (STAT[SLVPENDING] = 1) by polling the STAT register. Check the status register STAT[SLVSTATE] indicating ADDR. If not then an error has occurred. 2. ACK the address by setting SLVCTL[SLVCONTINUE] = 1 in the slave control register. 3. Wait for the pending status to be set (STAT[SLVPENDING] = 1) by polling the STAT register. Check the status register STAT[SLVSTATE] is indicating TX. If not then an error has occurred. 4. Write 8 bits of data to SLVDAT register. 5. Continue the transaction by setting SLVCTL[SLVCONTINUE] = 1 in the slave control register. K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 174.

’474 Patent Claim	Representative NXP Product(s)
	27.2 Features • 12-bit successive approximation analog to digital converter. • Input multiplexing among up to 8 pins (6 external inputs, 1 temperature sensor and V_{BAT}). • A configurable conversion sequencer with configurable trigger • Optional automatic high/low threshold comparison and “zero crossing” detection. • 12-bit conversion rate of 190 kHz. Options for reduced resolution at higher conversion rates. • Burst conversion mode for single or multiple inputs. • Asynchronous operation. Asynchronous mode allows choosing ADC clock from FRO12M or XO32M. • A temperature sensor is connected to ADC channel 7, see Chapter 28 “Temperature Sensor” for further details. • Supply monitor is connected to ADC channel 6; this monitors V_{BAT}. K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 198. Configure the temperature sensor as follows: • Select the temperature sensor as source for channel 7 of the ADC by writing the SEQ_CTRL[CHANNELS] bits to 0x80. In order to return ADC channel 7 to measuring its related device pin, write the SEQ_CTRL[CHANNELS] bits to 0x80. • The digital temperature reading is available after an analog-to-digital conversion of ADC channel 7. **Remark:** To convert the ADC conversion result into a temperature reading, use the API provided. This uses device specific calibration data stored in the device to increase the accuracy of the temperature reading. K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 199.

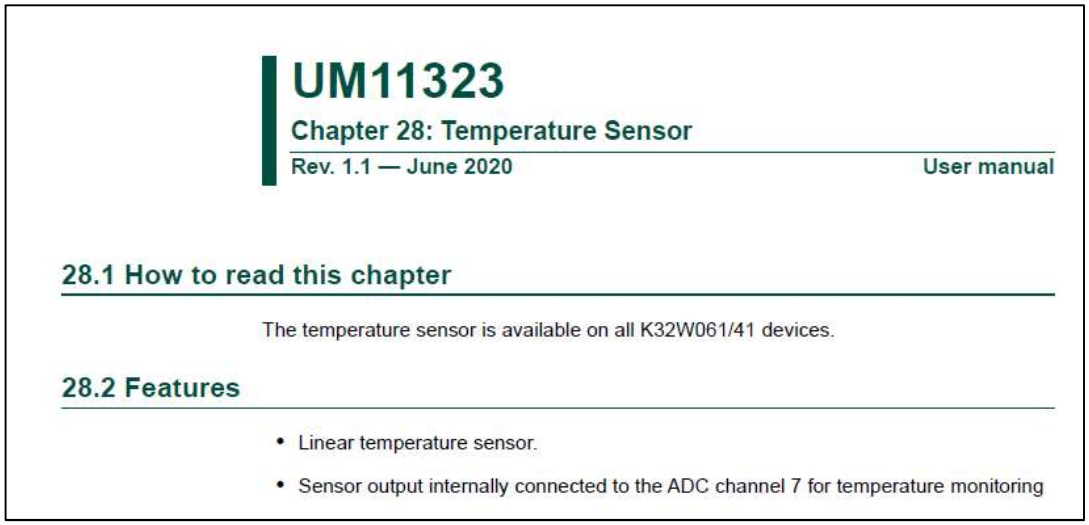
’474 Patent Claim	Representative NXP Product(s)

Table 58. ADC channels

ADC channel	Function	Device Pin
0	ADC0	PIO14
1	ADC1	PIO15
2	ADC2	PIO16
3	ADC3	PIO17
4	ADC4	PIO18
5	ADC5	PIO19
6	Supply monitor	Internal function
7	Temperature sensor	Internal function

K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 200.

See also:



UM11323

Chapter 28: Temperature Sensor

Rev. 1.1 — June 2020 User manual

28.1 How to read this chapter

The temperature sensor is available on all K32W061/41 devices.

28.2 Features

- Linear temperature sensor.
- Sensor output internally connected to the ADC channel 7 for temperature monitoring

K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 208.

’474 Patent Claim	Representative NXP Product(s)

28.3 Basic configuration

This section explains how the Temperature Sensor can be used. For a functional example see lpc_adc_basic.

- Enable the power to the temperature sensor by setting the ASYNC_SYSCON_TEMPSENSORCTRL[ENABLE] .
- Configure temperature sensor common mode output voltage setting ASYNC_SYSCON_TEMPSENSORCTRL[CM] = 0x2 for proper default operation
- To monitor the temperature continually, select the temperature sensor as source for channel 7 of ADC0. See Chapter 27. The digital temperature reading is available after an analog-to-digital conversion.
- The ADC reading must be converted into a temperature reading. To increase accuracy the sensor and ADC are calibrated during production, An API is provided to produce a temperature value; this performs the best configuration of the ADC for the purpose of the temperature sensor. The calibration data and other characteristics of the temperature and ADC are used to produce a high accuracy results.
- For highest accuracy, set ADCCLK mux source to be 32 MHz XTAL with a divider setting of 7, to give an ADCCLK of 4 MHz.
- The voltage range of operation of the ADC is set by ADC_GPADC_CTRL0[TEST]. In normal mode, the ADC can take an input voltage of 0 to 3.6 V, to V_{BAT} if this is lower. For the temperature sensor, the ADC must be configured in Unity Gain mode when the input voltage range is 0 to 0.9 V. Since the temperature sensor voltage output is within this range, the best accuracy is achieved. A consequence of this is that the temperature sensor can not be combined with the other ADC inputs as part of sequencer configuration. Also, safe practice is to set the mode back to normal mode after using the ADC with the temperature sensor.

K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 208.

28.3.1 Perform a single ADC conversion with the temperature sensor as ADC input

As mentioned in the previous chapter, the API should be used when performing temperature measurements. As a simple example of obtaining a temperature measurement, the following steps can be performed. In this case, the accuracy is not as high as that using the API.

K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 208.

’474 Patent Claim	Representative NXP Product(s)
	To perform a single ADC conversion for ADC0 channel 7 using the temperature sensor output: 1. Enable the temperature sensor output as input to ADC channel 7. 2. Configure the system clock and the ADC for operation. 3. Select the asynchronous mode in the ADC_CTRL register. 4. Select ADC channel 7 to perform the conversion by setting the ADC_SEQ_CTRL[CHANNELS] bits to 0x80. 5. Set the ADC_SEQ_CTRL[START] bit to 1. 6. Read the SEQ_GDAT[RESULT] bits for the conversion result. 7. The AHI software may be used to generate the temperature value. In fact, the example driver will perform this sequencing as well as making corrections due to the calibration data, and using averaging to give the best result. K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 209.

’474 Patent Claim	Representative NXP Product(s)

17.1.3 ADC Conversion Sequence Control Register (SEQ_CTRL)

Offset

Register	Offset
SEQ_CTRL	8h

Function

This register controls triggering and channel selection for conversion sequence. Also specifies interrupt mode for sequence. All ADC conversions are controlled through this sequencer which can be used for a single conversion or sets of conversions on one or more channels.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SEQ_ENA	MODE	Reserved	SINGLES...	BURST		STAR T_B...		Reserved				SYNC BYP...	TRIGP OL	TRIGGER	
W						START										
Reset	0	0	u	0	0	u	0	u	u	u	u	u	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TRIGGER				Reserved				CHANNELS							
W																
Reset	0	0	0	0	u	u	u	u	0	0	0	0	0	0	0	0

K32W061/K32W041 Register Manual (Rev. 1.1, 06/2020) at 440.

7-0 CHANNELS	ADC Channels Select which one or more of the ADC channels will be sampled and converted when this sequence is launched. A 1 in any bit of this field will cause the corresponding channel to be included in the conversion sequence, where bit 0 corresponds to channel 0, bit 1 to channel 1 and so forth. Bit 6 is channel 6; the supply monitor. Bit 7 is channel 7; the temperature sensor. When this conversion sequence is triggered, either by a hardware trigger or via software command, ADC conversions will be performed on each enabled channel, in sequence, beginning with the lowest-ordered channel. Remark: This field can ONLY be changed while SEQ_ENA (bit 31) is LOW. It is allowed to change this field and set bit 31 in the same write.

K32W061/K32W041 Register Manual (Rev. 1.1, 06/2020) at 443.

’474 Patent Claim	Representative NXP Product(s)

17.1.4 ADC Sequence Global Data Register (SEQ_GDAT)

Offset

Register	Offset
SEQ_GDAT	10h

Function

This register contains the result of the most recent ADC conversion completed under each conversion sequence. Results of ADC conversions can be read in one of two ways. One is to use this register to read data from the ADC at the end of each ADC conversion. The other is to read the individual ADC Channel Data (DATn) registers, typically after the entire sequence has completed. It is recommended to use one method consistently for a given conversion sequence.

This register is useful in conjunction with DMA operation - particularly when the channels selected for conversion are not sequential (hence the addresses of the individual result registers will not be sequential, making it difficult for the DMA engine to address them). For interrupt-driven code, it will more likely be advantageous to wait for an entire sequence to complete and then retrieve the results from the individual channel registers.

NOTE

The method to be employed for each sequence should be reflected in the SEQ_CTRL[MODE] bit since this will impact interrupt and overrun flag generation.

K32W061/K32W041 Register Manual (Rev. 1.1, 06/2020) at 443.

15-4 RESULT	ADC Conversion Result This field contains the 12-bit ADC conversion result from the most recent conversion performed under conversion sequence associated with this register. DATAVALID = 1 indicates that this result has not yet been read. If less than 12-bit resolultion is used the ADC result occupies the upper MSBs and unused LSBs should be ignored.

K32W061/K32W041 Register Manual (Rev. 1.1, 06/2020) at 445.

’474 Patent Claim	Representative NXP Product(s)

17.1.5 ADC Channel a Data Register (DAT0 - DAT7)

Function

These registers hold the result of the last conversion completed for each ADC channel. They also include status bits to indicate when a conversion has been completed, when a data overrun has occurred, and where the most recent conversion fits relative to the range dictated by the high and low threshold registers.

Results of ADC conversion can be read in one of two ways. One is to use the SEQ_GDAT register for each of the sequences to read data from the ADC at the end of each ADC conversion. The other is to use these individual ADC Channel Data registers, typically after the entire sequence has completed. It is recommended to use one method consistently for a given conversion sequence.

NOTE

The method to be employed for each sequence should be reflected in the MODE bit in the SEQ_CTRL register since this will impact interrupt and overrun flag generation.

The information presented in the DAT registers always pertains to the most recent conversion completed on that channel regardless of what sequence requested the conversion or which trigger caused it.

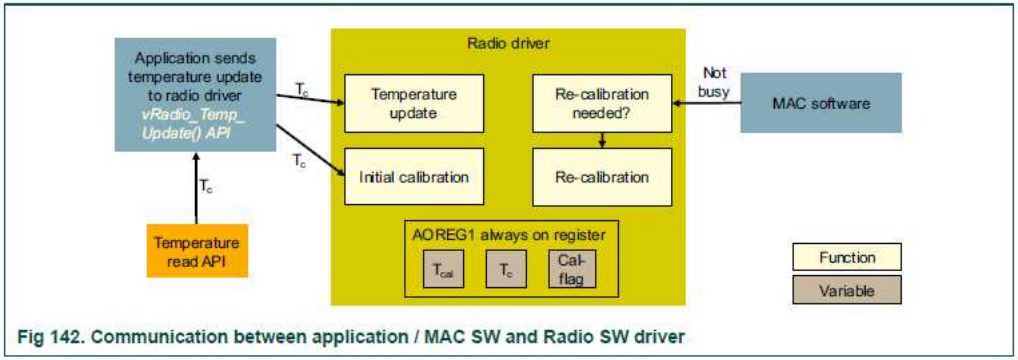
The OVERRUN fields for each channel are also replicated in the FLAGS register.

K32W061/K32W041 Register Manual (Rev. 1.1, 06/2020) at 445-446.

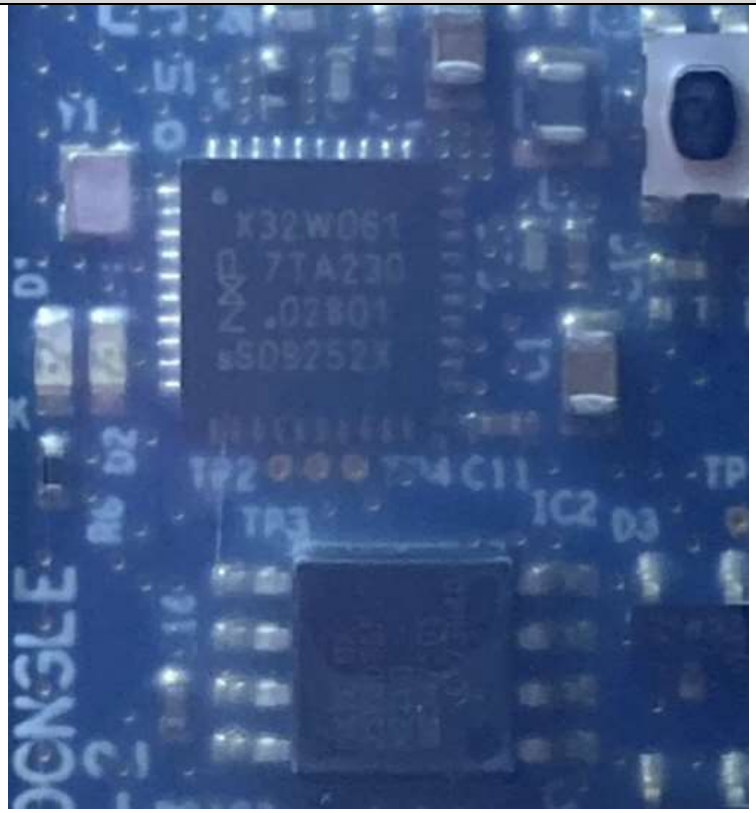
15-4 RESULT	ADC Conversion Result This field contains the 12-bit ADC conversion result from the most recent conversion performed under conversion sequence associated with this register. DATAVALID = 1 indicates that this result has not yet been read. If less than 12-bit resolultion is used, the ADC result occupies the upper MSBs and unused LSBs should be ignored.

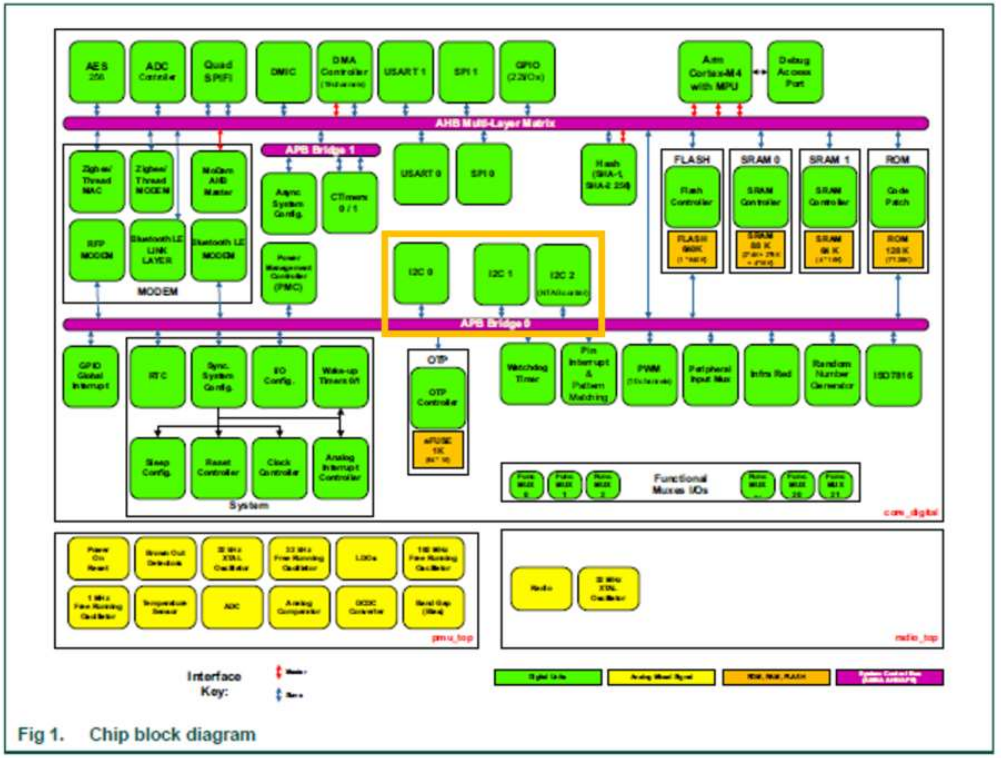
K32W061/K32W041 Register Manual (Rev. 1.1, 06/2020) at 447.

See also:

’474 Patent Claim	Representative NXP Product(s)
	 Fig 142. Communication between application / MAC SW and Radio SW driver K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 324.
[8a.]. A method for communicating over a point to point serial communication system having a clock signal line and a data signal line coupling a serial interface master and a serial interface slave, the method comprising:	To the extent the preamble is limiting, the Accused ’474 Wireless Microcontrollers perform a method for communicating over a point to point serial communication system having a clock signal line and a data signal line coupling a serial interface master and a serial interface slave. For example, each of the Accused ’474 Wireless Microcontrollers perform a method for communicating over a point to point serial communication system identified below having a clock signal line (*e.g.*, SCL) and a data signal line (*e.g.*, SDA) coupling a serial interface master (*e.g.*, the I2C interface on the master/MCU and/or the host processor) and a serial interface slave (*e.g.*, I2C). *See, e.g.*:  https://www.nxp.com/docs/en/data-sheet/K32W061.pdf *See also* https://www.avnet.com/shop/us/search/k32w041

’474 Patent Claim	Representative NXP Product(s)
	 OM15080-K32W (Development Board of K32W061/41) https://www.arrow.com/en/products/om15080-k32w/nxp-semiconductors?q=OM15080-K32W

’474 Patent Claim	Representative NXP Product(s)
	 OM15080-K32W (Development Board of K32W061/41) (Mouser)

’474 Patent Claim	Representative NXP Product(s)
	1.3 Block diagram  Fig 1. Chip block diagram K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 6.

’474 Patent Claim	Representative NXP Product(s)
	25.2 Features • Independent Master, Slave, and Monitor functions. • Bus speeds supported: – Standard mode, up to 100 kbits/s. – Fast-mode, up to 400 kbits/s. – Fast-mode Plus, up to 1 Mbits/s (on pins PIO0_10 and PIO0_11 that include specific I^2C support). – High speed mode, 3.4 Mbits/s as a Slave only (on pins PIO0_10 and PIO0_11 that include specific I^2C support). • Supports both Multi-master and Multi-master with Slave functions. • Multiple I^2C slave addresses supported in hardware. • One slave address can be selectively qualified with a bit mask or an address range in order to respond to multiple I^2C bus addresses. • 10-bit addressing supported with software assist. • Supports System Management Bus (SMBus). • Separate DMA requests for Master, Slave, and Monitor functions. • No chip clocks are required in order to receive and compare an address as a Slave, so this event can wake up the device from deep-sleep mode. Additionally, I^2C0 can optionally generate a wake-up from power down. • Automatic modes optionally allow less software overhead for some use cases. K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 170.

’474 Patent Claim	Representative NXP Product(s)

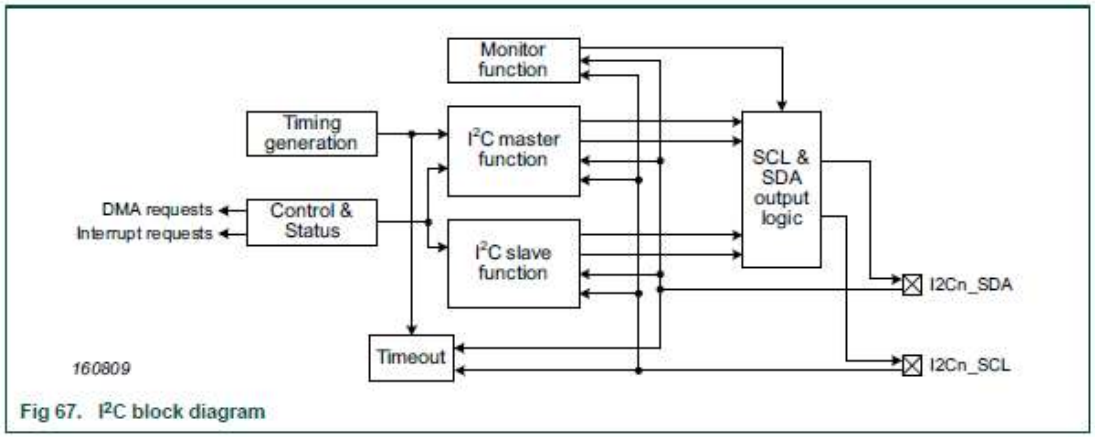


Fig 67. I²C block diagram

K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 176.

Table 46. I²C-bus pin description

Function	Type	Pin name used in data sheet Pin Description	Description
SCL	I/O	I2Cn_SCL	I²C serial clock
SDA	I/O	I2Cn_SDA	I²C serial data

K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 170.

Table 47. I²C bus pin assignments

Pin	Possible Pin Assignment	
I2C0_SCL	PIO0_10[1]	PIO0_15
I2C0_SDA	PIO0_11[1]	PIO0_16
I2C1_SCL	PIO0_6	PIO0_12
I2C1_SDA	PIO0_7	PIO0_13
I2C2_SCL	Internal connection	
I2C2_SDA	Internal connection	

K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 171.

’474 Patent Claim	Representative NXP Product(s)
	25.4.2 I²C receive/transmit in slave mode In this example, I²C1 is used as an I²C slave. The slave receives 8 bits from the master and then sends 8 bits to the master. The SCL and SDA functions must be enabled on suitable pins, see Table 48. The pins should be configured as required for the I²C-bus mode. The transmission of the address and data bits is controlled by the STAT[SLVPENDING] status bit. Whenever the status is Slave pending, the slave can acknowledge (“ack”) or send or receive an address and data. The received data or the data to be sent to the master are available in the SLVDAT register. After sending and receiving data, continue to the next step of the transmission protocol by writing to the SLVCTL register. K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 173. **25.4.2.1 Slave read from master** This example uses polling to control the sequence and does not use interrupts. Configure the I²C as a slave with address x: 1. Write the slave address x to the address 0 match register. 2. Set the CFG[SLVEN] bit to 1. K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 173.
[8b.] sending a read temperature command to the serial interface slave from the serial interface master using the clock signal line and the data signal line; and	The Accused ’474 Wireless Microcontrollers performs a step of sending a read temperature command to the serial interface slave from the serial interface master using the clock signal line and the data signal line. For example, each of the Accused ’474 Wireless Microcontrollers perform a step of sending a read temperature command (*e.g.*, the read tempearture command requesting the result derived from the temperature sensor measurement) to the serial interface slave identified above from the serial interface master identified above using the clock signal line identified above and the data signal line identified above.

’474 Patent Claim	Representative NXP Product(s)
	25.4.2.2 Slave write to master This example uses polling to control the sequence and does not use interrupts. Configure the I^2C as a slave with address x: 1. Write the slave address x to the address 0 match register. 2. Set the CFG[SLVEN] bit to 1. Write data to the master: 1. Wait for the pending status to be set (STAT[SLVPENDING] = 1) by polling the STAT register. Check the status register STAT[SLVSTATE] indicating ADDR. If not then an error has occurred. 2. ACK the address by setting SLVCTL[SLVCONTINUE] = 1 in the slave control register. 3. Wait for the pending status to be set (STAT[SLVPENDING] = 1) by polling the STAT register. Check the status register STAT[SLVSTATE] is indicating TX. If not then an error has occurred. 4. Write 8 bits of data to SLVDAT register. 5. Continue the transaction by setting SLVCTL[SLVCONTINUE] = 1 in the slave control register. K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 174.

’474 Patent Claim	Representative NXP Product(s)
	27.2 Features • 12-bit successive approximation analog to digital converter. • Input multiplexing among up to 8 pins (6 external inputs, 1 temperature sensor and V_{BAT}). • A configurable conversion sequencer with configurable trigger • Optional automatic high/low threshold comparison and “zero crossing” detection. • 12-bit conversion rate of 190 kHz. Options for reduced resolution at higher conversion rates. • Burst conversion mode for single or multiple inputs. • Asynchronous operation. Asynchronous mode allows choosing ADC clock from FRO12M or XO32M. • A temperature sensor is connected to ADC channel 7, see Chapter 28 “Temperature Sensor” for further details. • Supply monitor is connected to ADC channel 6; this monitors V_{BAT}. K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 198. Configure the temperature sensor as follows: • Select the temperature sensor as source for channel 7 of the ADC by writing the SEQ_CTRL[CHANNELS] bits to 0x80. In order to return ADC channel 7 to measuring its related device pin, write the SEQ_CTRL[CHANNELS] bits to 0x80. • The digital temperature reading is available after an analog-to-digital conversion of ADC channel 7. **Remark:** To convert the ADC conversion result into a temperature reading, use the API provided. This uses device specific calibration data stored in the device to increase the accuracy of the temperature reading. K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 199.

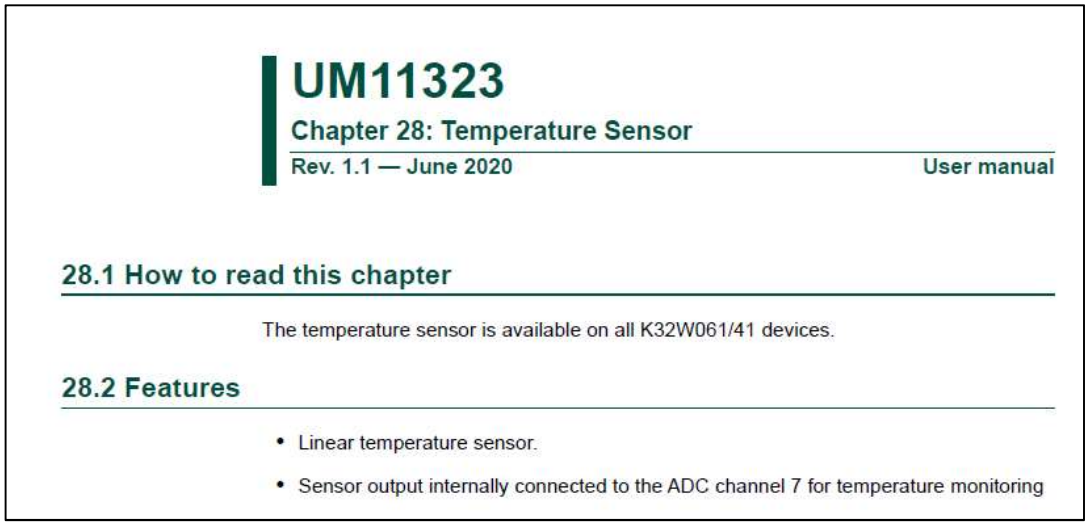
’474 Patent Claim	Representative NXP Product(s)

Table 58. ADC channels

ADC channel	Function	Device Pin
0	ADC0	PIO14
1	ADC1	PIO15
2	ADC2	PIO16
3	ADC3	PIO17
4	ADC4	PIO18
5	ADC5	PIO19
6	Supply monitor	Internal function
7	Temperature sensor	Internal function

K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 200.

See also:



UM11323

Chapter 28: Temperature Sensor

Rev. 1.1 — June 2020 User manual

28.1 How to read this chapter

The temperature sensor is available on all K32W061/41 devices.

28.2 Features

- Linear temperature sensor.
- Sensor output internally connected to the ADC channel 7 for temperature monitoring

K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 208.

’474 Patent Claim	Representative NXP Product(s)

28.3 Basic configuration

This section explains how the Temperature Sensor can be used. For a functional example see lpc_adc_basic.

- Enable the power to the temperature sensor by setting the ASYNC_SYSCON_TEMPSENSORCTRL[ENABLE] .
- Configure temperature sensor common mode output voltage setting ASYNC_SYSCON_TEMPSENSORCTRL[CM] = 0x2 for proper default operation
- To monitor the temperature continually, select the temperature sensor as source for channel 7 of ADC0. See Chapter 27. The digital temperature reading is available after an analog-to-digital conversion.
- The ADC reading must be converted into a temperature reading. To increase accuracy the sensor and ADC are calibrated during production, An API is provided to produce a temperature value; this performs the best configuration of the ADC for the purpose of the temperature sensor. The calibration data and other characteristics of the temperature and ADC are used to produce a high accuracy results.
- For highest accuracy, set ADCCLK mux source to be 32 MHz XTAL with a divider setting of 7, to give an ADCCLK of 4 MHz.
- The voltage range of operation of the ADC is set by ADC_GPADC_CTRL0[TEST]. In normal mode, the ADC can take an input voltage of 0 to 3.6 V, to V_{BAT} if this is lower. For the temperature sensor, the ADC must be configured in Unity Gain mode when the input voltage range is 0 to 0.9 V. Since the temperature sensor voltage output is within this range, the best accuracy is achieved. A consequence of this is that the temperature sensor can not be combined with the other ADC inputs as part of sequencer configuration. Also, safe practice is to set the mode back to normal mode after using the ADC with the temperature sensor.

K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 208.

28.3.1 Perform a single ADC conversion with the temperature sensor as ADC input

As mentioned in the previous chapter, the API should be used when performing temperature measurements. As a simple example of obtaining a temperature measurement, the following steps can be performed. In this case, the accuracy is not as high as that using the API.

K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 208.

’474 Patent Claim	Representative NXP Product(s)
	To perform a single ADC conversion for ADC0 channel 7 using the temperature sensor output: 1. Enable the temperature sensor output as input to ADC channel 7. 2. Configure the system clock and the ADC for operation. 3. Select the asynchronous mode in the ADC_CTRL register. 4. Select ADC channel 7 to perform the conversion by setting the ADC_SEQ_CTRL[CHANNELS] bits to 0x80. 5. Set the ADC_SEQ_CTRL[START] bit to 1. 6. Read the SEQ_GDAT[RESULT] bits for the conversion result. 7. The AHI software may be used to generate the temperature value. In fact, the example driver will perform this sequencing as well as making corrections due to the calibration data, and using averaging to give the best result. K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 209.

’474 Patent Claim	Representative NXP Product(s)

17.1.3 ADC Conversion Sequence Control Register (SEQ_CTRL)

Offset

Register	Offset
SEQ_CTRL	8h

Function

This register controls triggering and channel selection for conversion sequence. Also specifies interrupt mode for sequence. All ADC conversions are controlled through this sequencer which can be used for a single conversion or sets of conversions on one or more channels.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SEQ_ENA	MODE	Reserved	SINGLES...	BURST		STAR T_B...		Reserved				SYNC BYP...	TRIGP OL	TRIGGER	
W						START										
Reset	0	0	u	0	0	u	0	u	u	u	u	u	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TRIGGER				Reserved				CHANNELS							
W																
Reset	0	0	0	0	u	u	u	u	0	0	0	0	0	0	0	0

K32W061/K32W041 Register Manual (Rev. 1.1, 06/2020) at 440.

7-0 CHANNELS	ADC Channels Select which one or more of the ADC channels will be sampled and converted when this sequence is launched. A 1 in any bit of this field will cause the corresponding channel to be included in the conversion sequence, where bit 0 corresponds to channel 0, bit 1 to channel 1 and so forth. Bit 6 is channel 6; the supply monitor. Bit 7 is channel 7; the temperature sensor. When this conversion sequence is triggered, either by a hardware trigger or via software command, ADC conversions will be performed on each enabled channel, in sequence, beginning with the lowest-ordered channel. Remark: This field can ONLY be changed while SEQ_ENA (bit 31) is LOW. It is allowed to change this field and set bit 31 in the same write.

K32W061/K32W041 Register Manual (Rev. 1.1, 06/2020) at 443.

’474 Patent Claim	Representative NXP Product(s)

17.1.4 ADC Sequence Global Data Register (SEQ_GDAT)

Offset

Register	Offset
SEQ_GDAT	10h

Function

This register contains the result of the most recent ADC conversion completed under each conversion sequence. Results of ADC conversions can be read in one of two ways. One is to use this register to read data from the ADC at the end of each ADC conversion. The other is to read the individual ADC Channel Data (DATn) registers, typically after the entire sequence has completed. It is recommended to use one method consistently for a given conversion sequence.

This register is useful in conjunction with DMA operation - particularly when the channels selected for conversion are not sequential (hence the addresses of the individual result registers will not be sequential, making it difficult for the DMA engine to address them). For interrupt-driven code, it will more likely be advantageous to wait for an entire sequence to complete and then retrieve the results from the individual channel registers.

NOTE

The method to be employed for each sequence should be reflected in the SEQ_CTRL[MODE] bit since this will impact interrupt and overrun flag generation.

K32W061/K32W041 Register Manual (Rev. 1.1, 06/2020) at 443.

15-4 RESULT	ADC Conversion Result This field contains the 12-bit ADC conversion result from the most recent conversion performed under conversion sequence associated with this register. DATAVALID = 1 indicates that this result has not yet been read. If less than 12-bit resolultion is used the ADC result occupies the upper MSBs and unused LSBs should be ignored.

K32W061/K32W041 Register Manual (Rev. 1.1, 06/2020) at 445.

’474 Patent Claim	Representative NXP Product(s)

17.1.5 ADC Channel a Data Register (DAT0 - DAT7)

Function

These registers hold the result of the last conversion completed for each ADC channel. They also include status bits to indicate when a conversion has been completed, when a data overrun has occurred, and where the most recent conversion fits relative to the range dictated by the high and low threshold registers.

Results of ADC conversion can be read in one of two ways. One is to use the SEQ_GDAT register for each of the sequences to read data from the ADC at the end of each ADC conversion. The other is to use these individual ADC Channel Data registers, typically after the entire sequence has completed. It is recommended to use one method consistently for a given conversion sequence.

NOTE

The method to be employed for each sequence should be reflected in the MODE bit in the SEQ_CTRL register since this will impact interrupt and overrun flag generation.

The information presented in the DAT registers always pertains to the most recent conversion completed on that channel regardless of what sequence requested the conversion or which trigger caused it.

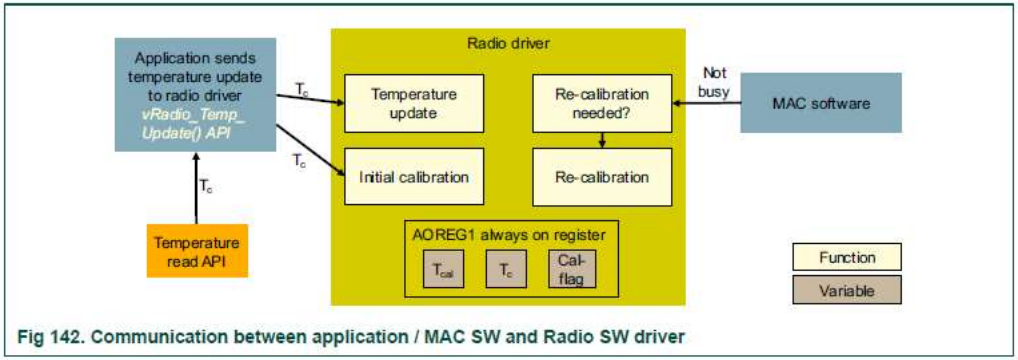
The OVERRUN fields for each channel are also replicated in the FLAGS register.

K32W061/K32W041 Register Manual (Rev. 1.1, 06/2020) at 445-446.

15-4 RESULT	ADC Conversion Result This field contains the 12-bit ADC conversion result from the most recent conversion performed under conversion sequence associated with this register. DATAVALID = 1 indicates that this result has not yet been read. If less than 12-bit resolultion is used, the ADC result occupies the upper MSBs and unused LSBs should be ignored.

K32W061/K32W041 Register Manual (Rev. 1.1, 06/2020) at 447.

See also:

’474 Patent Claim	Representative NXP Product(s)
	 Fig 142. Communication between application / MAC SW and Radio SW driver K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 324.
[8c.] in response to the read temperature command, the serial interface slave supplying over the data signal line a temperature value associated with a processor on an integrated circuit containing the serial interface slave.	The Accused ’474 Wireless Microcontrollers performs a step of in response to the read temperature command, the serial interface slave supplying over the data signal line a temperature value associated with a processor on an integrated circuit containing the serial interface slave. For example, each of the Accused ’474 Wireless Microcontrollers perform a step of in response to the read temperature command, the serial interface slave (e.g., the serial interface slave identified above in [8b.]) supplying over the data signal line (e.g., the data signal identified above in [8b.]) a temperature value associated with a processor on an integrated circuit containing the serial interface slave (*e.g.*, the result derived from the temperature sensor measurement).

’474 Patent Claim	Representative NXP Product(s)
	25.4.2.2 Slave write to master This example uses polling to control the sequence and does not use interrupts. Configure the I^2C as a slave with address x: 1. Write the slave address x to the address 0 match register. 2. Set the CFG[SLVEN] bit to 1. Write data to the master: 1. Wait for the pending status to be set (STAT[SLVPENDING] = 1) by polling the STAT register. Check the status register STAT[SLVSTATE] indicating ADDR. If not then an error has occurred. 2. ACK the address by setting SLVCTL[SLVCONTINUE] = 1 in the slave control register. 3. Wait for the pending status to be set (STAT[SLVPENDING] = 1) by polling the STAT register. Check the status register STAT[SLVSTATE] is indicating TX. If not then an error has occurred. 4. Write 8 bits of data to SLVDAT register. 5. Continue the transaction by setting SLVCTL[SLVCONTINUE] = 1 in the slave control register. K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 174.

’474 Patent Claim	Representative NXP Product(s)
	27.2 Features • 12-bit successive approximation analog to digital converter. • Input multiplexing among up to 8 pins (6 external inputs, 1 temperature sensor and V_{BAT}). • A configurable conversion sequencer with configurable trigger • Optional automatic high/low threshold comparison and “zero crossing” detection. • 12-bit conversion rate of 190 kHz. Options for reduced resolution at higher conversion rates. • Burst conversion mode for single or multiple inputs. • Asynchronous operation. Asynchronous mode allows choosing ADC clock from FRO12M or XO32M. • A temperature sensor is connected to ADC channel 7, see Chapter 28 “Temperature Sensor” for further details. • Supply monitor is connected to ADC channel 6; this monitors V_{BAT}. K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 198. Configure the temperature sensor as follows: • Select the temperature sensor as source for channel 7 of the ADC by writing the SEQ_CTRL[CHANNELS] bits to 0x80. In order to return ADC channel 7 to measuring its related device pin, write the SEQ_CTRL[CHANNELS] bits to 0x80. • The digital temperature reading is available after an analog-to-digital conversion of ADC channel 7. **Remark:** To convert the ADC conversion result into a temperature reading, use the API provided. This uses device specific calibration data stored in the device to increase the accuracy of the temperature reading. K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 199.

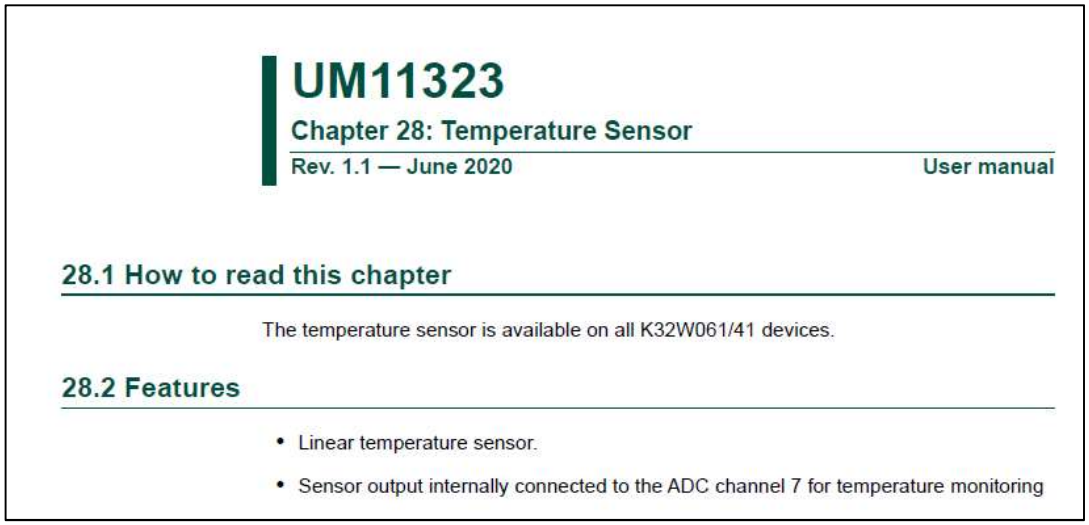
’474 Patent Claim	Representative NXP Product(s)

Table 58. ADC channels

ADC channel	Function	Device Pin
0	ADC0	PIO14
1	ADC1	PIO15
2	ADC2	PIO16
3	ADC3	PIO17
4	ADC4	PIO18
5	ADC5	PIO19
6	Supply monitor	Internal function
7	Temperature sensor	Internal function

K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 200.

See also:



UM11323

Chapter 28: Temperature Sensor

Rev. 1.1 — June 2020 User manual

28.1 How to read this chapter

The temperature sensor is available on all K32W061/41 devices.

28.2 Features

- Linear temperature sensor.
- Sensor output internally connected to the ADC channel 7 for temperature monitoring

K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 208.

’474 Patent Claim	Representative NXP Product(s)

28.3 Basic configuration

This section explains how the Temperature Sensor can be used. For a functional example see lpc_adc_basic.

- Enable the power to the temperature sensor by setting the ASYNC_SYSCON_TEMPSENSORCTRL[ENABLE] .
- Configure temperature sensor common mode output voltage setting ASYNC_SYSCON_TEMPSENSORCTRL[CM] = 0x2 for proper default operation
- To monitor the temperature continually, select the temperature sensor as source for channel 7 of ADC0. See Chapter 27. The digital temperature reading is available after an analog-to-digital conversion.
- The ADC reading must be converted into a temperature reading. To increase accuracy the sensor and ADC are calibrated during production, An API is provided to produce a temperature value; this performs the best configuration of the ADC for the purpose of the temperature sensor. The calibration data and other characteristics of the temperature and ADC are used to produce a high accuracy results.
- For highest accuracy, set ADCCLK mux source to be 32 MHz XTAL with a divider setting of 7, to give an ADCCLK of 4 MHz.
- The voltage range of operation of the ADC is set by ADC_GPADC_CTRL0[TEST]. In normal mode, the ADC can take an input voltage of 0 to 3.6 V, to V_{BAT} if this is lower. For the temperature sensor, the ADC must be configured in Unity Gain mode when the input voltage range is 0 to 0.9 V. Since the temperature sensor voltage output is within this range, the best accuracy is achieved. A consequence of this is that the temperature sensor can not be combined with the other ADC inputs as part of sequencer configuration. Also, safe practice is to set the mode back to normal mode after using the ADC with the temperature sensor.

K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 208.

28.3.1 Perform a single ADC conversion with the temperature sensor as ADC input

As mentioned in the previous chapter, the API should be used when performing temperature measurements. As a simple example of obtaining a temperature measurement, the following steps can be performed. In this case, the accuracy is not as high as that using the API.

K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 208.

’474 Patent Claim	Representative NXP Product(s)
	To perform a single ADC conversion for ADC0 channel 7 using the temperature sensor output: 1. Enable the temperature sensor output as input to ADC channel 7. 2. Configure the system clock and the ADC for operation. 3. Select the asynchronous mode in the ADC_CTRL register. 4. Select ADC channel 7 to perform the conversion by setting the ADC_SEQ_CTRL[CHANNELS] bits to 0x80. 5. Set the ADC_SEQ_CTRL[START] bit to 1. 6. Read the SEQ_GDAT[RESULT] bits for the conversion result. 7. The AHI software may be used to generate the temperature value. In fact, the example driver will perform this sequencing as well as making corrections due to the calibration data, and using averaging to give the best result. K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 209.

’474 Patent Claim	Representative NXP Product(s)

17.1.3 ADC Conversion Sequence Control Register (SEQ_CTRL)

Offset

Register	Offset
SEQ_CTRL	8h

Function

This register controls triggering and channel selection for conversion sequence. Also specifies interrupt mode for sequence. All ADC conversions are controlled through this sequencer which can be used for a single conversion or sets of conversions on one or more channels.

Diagram

K32W061/K32W041 Register Manual (Rev. 1.1, 06/2020) at 440.

7-0 CHANNELS	ADC Channels Select which one or more of the ADC channels will be sampled and converted when this sequence is launched. A 1 in any bit of this field will cause the corresponding channel to be included in the conversion sequence, where bit 0 corresponds to channel 0, bit 1 to channel 1 and so forth. Bit 6 is channel 6; the supply monitor. Bit 7 is channel 7; the temperature sensor. When this conversion sequence is triggered, either by a hardware trigger or via software command, ADC conversions will be performed on each enabled channel, in sequence, beginning with the lowest-ordered channel. Remark: This field can ONLY be changed while SEQ_ENA (bit 31) is LOW. It is allowed to change this field and set bit 31 in the same write.

K32W061/K32W041 Register Manual (Rev. 1.1, 06/2020) at 443.

’474 Patent Claim	Representative NXP Product(s)

17.1.4 ADC Sequence Global Data Register (SEQ_GDAT)

Offset

Register	Offset
SEQ_GDAT	10h

Function

This register contains the result of the most recent ADC conversion completed under each conversion sequence. Results of ADC conversions can be read in one of two ways. One is to use this register to read data from the ADC at the end of each ADC conversion. The other is to read the individual ADC Channel Data (DATn) registers, typically after the entire sequence has completed. It is recommended to use one method consistently for a given conversion sequence.

This register is useful in conjunction with DMA operation - particularly when the channels selected for conversion are not sequential (hence the addresses of the individual result registers will not be sequential, making it difficult for the DMA engine to address them). For interrupt-driven code, it will more likely be advantageous to wait for an entire sequence to complete and then retrieve the results from the individual channel registers.

NOTE

The method to be employed for each sequence should be reflected in the SEQ_CTRL[MODE] bit since this will impact interrupt and overrun flag generation.

K32W061/K32W041 Register Manual (Rev. 1.1, 06/2020) at 443.

15-4 RESULT	ADC Conversion Result This field contains the 12-bit ADC conversion result from the most recent conversion performed under conversion sequence associated with this register. DATAVALID = 1 indicates that this result has not yet been read. If less than 12-bit resolution is used the ADC result occupies the upper MSBs and unused LSBs should be ignored.

K32W061/K32W041 Register Manual (Rev. 1.1, 06/2020) at 445.

’474 Patent Claim	Representative NXP Product(s)

17.1.5 ADC Channel a Data Register (DAT0 - DAT7)

Function

These registers hold the result of the last conversion completed for each ADC channel. They also include status bits to indicate when a conversion has been completed, when a data overrun has occurred, and where the most recent conversion fits relative to the range dictated by the high and low threshold registers.

Results of ADC conversion can be read in one of two ways. One is to use the SEQ_GDAT register for each of the sequences to read data from the ADC at the end of each ADC conversion. The other is to use these individual ADC Channel Data registers, typically after the entire sequence has completed. It is recommended to use one method consistently for a given conversion sequence.

NOTE

The method to be employed for each sequence should be reflected in the MODE bit in the SEQ_CTRL register since this will impact interrupt and overrun flag generation.

The information presented in the DAT registers always pertains to the most recent conversion completed on that channel regardless of what sequence requested the conversion or which trigger caused it.

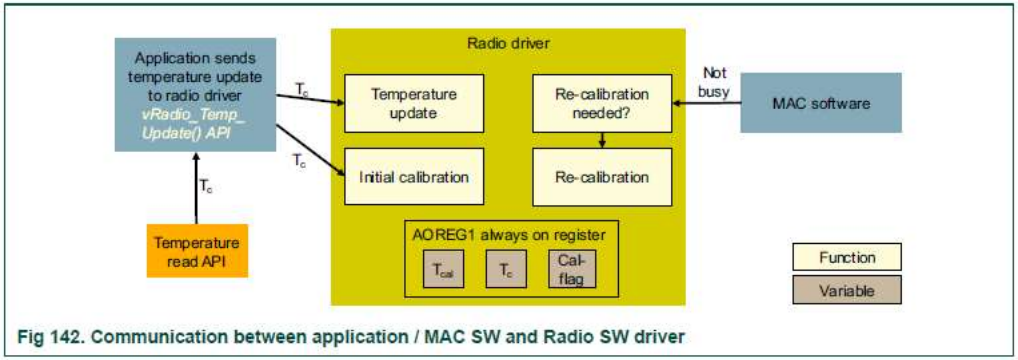
The OVERRUN fields for each channel are also replicated in the FLAGS register.

K32W061/K32W041 Register Manual (Rev. 1.1, 06/2020) at 445-446.

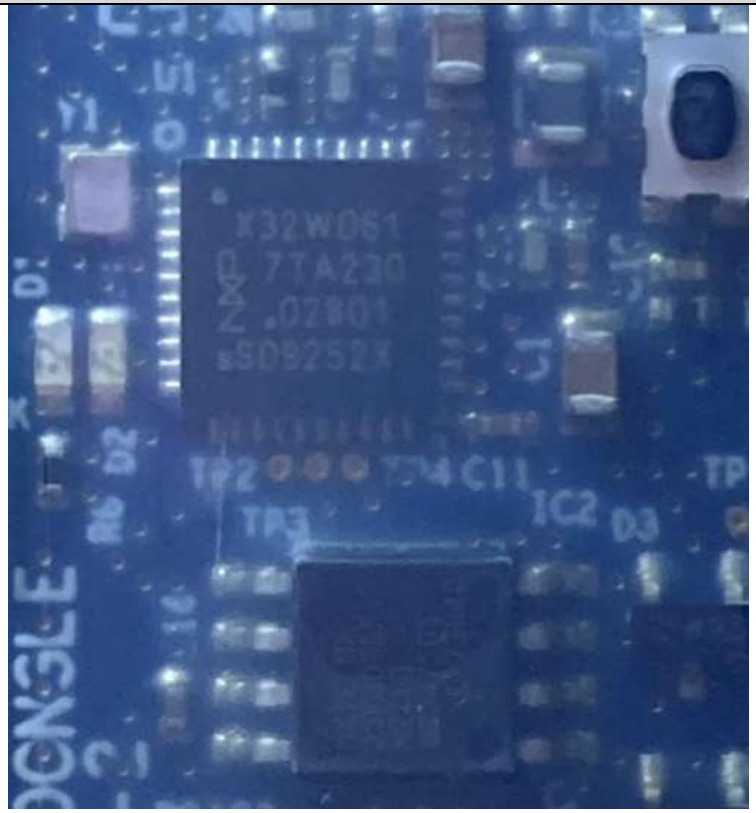
15-4 RESULT	ADC Conversion Result This field contains the 12-bit ADC conversion result from the most recent conversion performed under conversion sequence associated with this register. DATAVALID = 1 indicates that this result has not yet been read. If less than 12-bit resolultion is used, the ADC result occupies the upper MSBs and unused LSBs should be ignored.

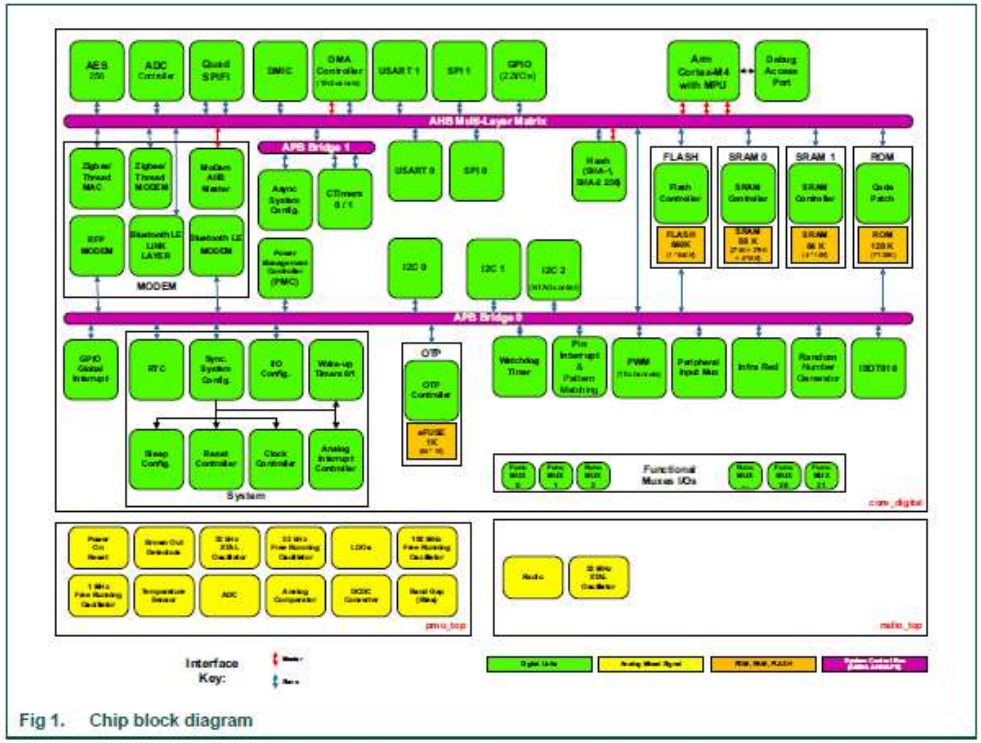
K32W061/K32W041 Register Manual (Rev. 1.1, 06/2020) at 447.

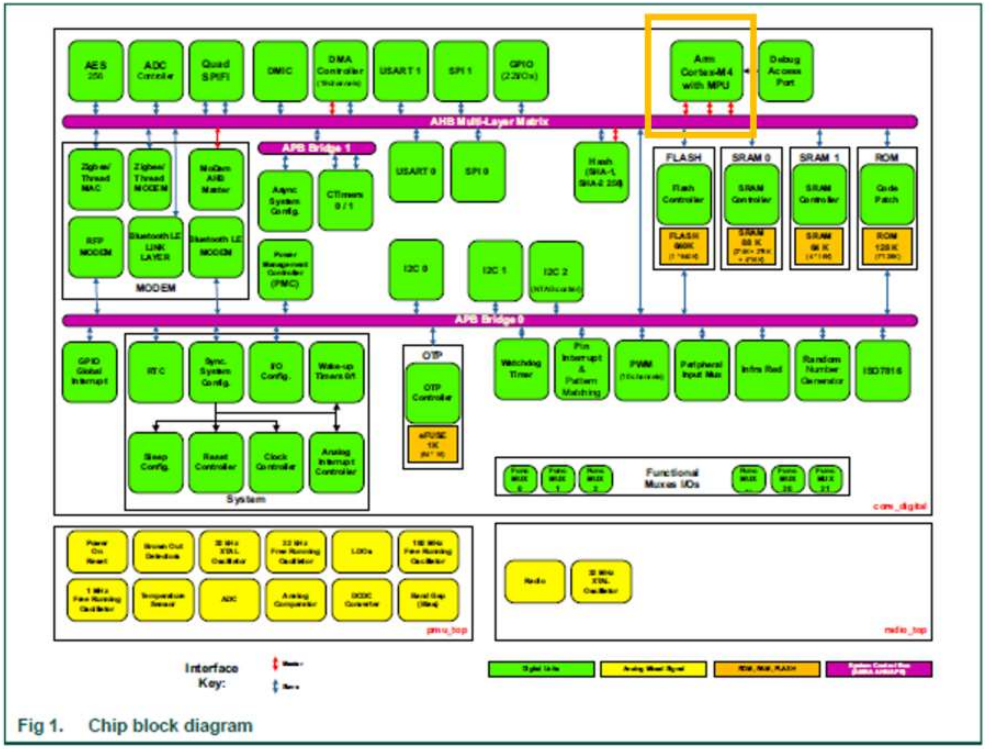
See also:

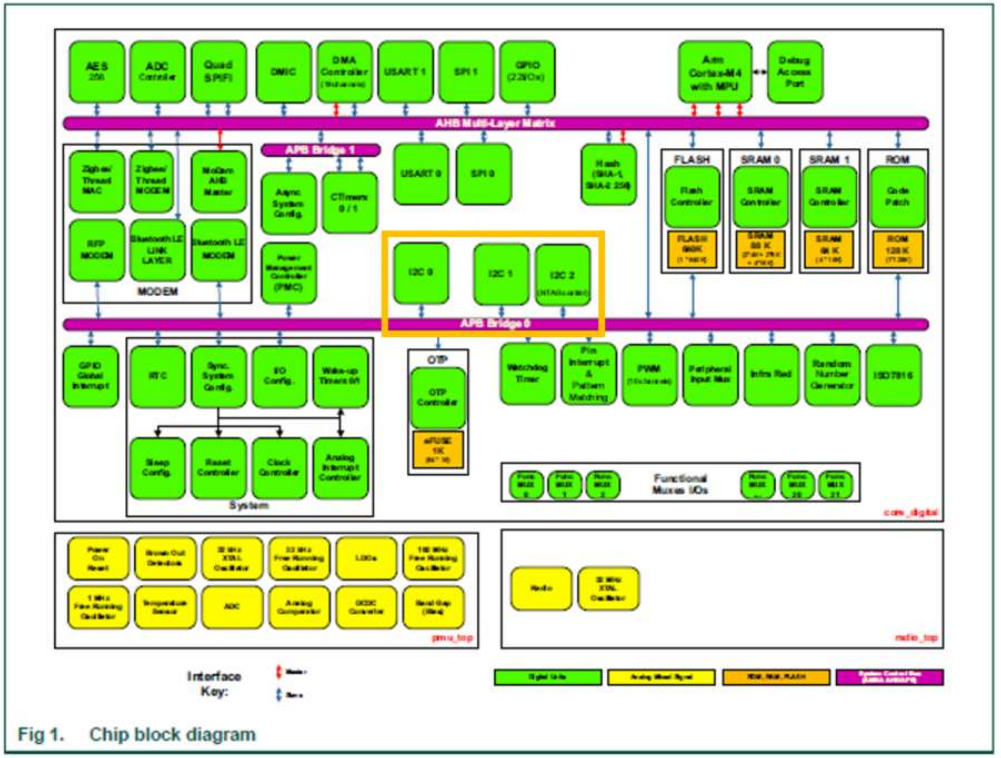
’474 Patent Claim	Representative NXP Product(s)
	 Fig 142. Communication between application / MAC SW and Radio SW driver K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 324.
[14a.] A serial communication system comprising:	To the extent the preamble is limiting, the Accused ’474 Wireless Microcontrollers include a “serial communication system” as recited in the ’474 patent. Exemplary systems and serial interfaces are identified in the block diagrams below. *See, e.g.*:  https://www.nxp.com/docs/en/data-sheet/K32W061.pdf *See also* https://www.avnet.com/shop/us/search/k32w041

’474 Patent Claim	Representative NXP Product(s)
	 OM15080-K32W (Development Board of K32W061/41) https://www.arrow.com/en/products/om15080-k32w/nxp-semiconductors?q=OM15080-K32W

’474 Patent Claim	Representative NXP Product(s)
	 OM15080-K32W (Development Board of K32W061/41) (Mouser)

’474 Patent Claim	Representative NXP Product(s)
	1.3 Block diagram  Fig 1. Chip block diagram K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 6.
[14b.] a microprocessor having	The Accused ’474 Wireless Microcontrollers each includes a microprocessor. For example, the Accused ’474 Wireless Microcontrollers each constitutes a microprocessor because, among others, they include at the processing logics identified below. *See, e.g.*:

’474 Patent Claim	Representative NXP Product(s)
	1.3 Block diagram  Fig 1. Chip block diagram K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 6.
[14c.] a slave serial interface for coupling to a master serial interface through a clock signal line output terminal and a data signal line output terminal	In each of the Accused ’474 Wireless Microcontrollers, the microprocessor has a slave serial interface for coupling to a master serial interface through a clock signal line output terminal and a data signal line output terminal. For example, each of the Accused ’474 Wireless Microcontrollers has a slave serial interface (*e.g.*, I2C) for coupling to a master serial interface (*e.g.*, the I2C interface on the master/MCU and/or the host processor) through a clock signal line output terminal (*e.g.*, SCL) and a data signal line output terminal

’474 Patent Claim	Representative NXP Product(s)
	(*e.g.*, SDA)). **1.3 Block diagram**  Fig 1. Chip block diagram K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 6.

’474 Patent Claim	Representative NXP Product(s)
	25.2 Features • Independent Master, Slave, and Monitor functions. • Bus speeds supported: – Standard mode, up to 100 kbits/s. – Fast-mode, up to 400 kbits/s. – Fast-mode Plus, up to 1 Mbits/s (on pins PIO0_10 and PIO0_11 that include specific I^2C support). – High speed mode, 3.4 Mbits/s as a Slave only (on pins PIO0_10 and PIO0_11 that include specific I^2C support). • Supports both Multi-master and Multi-master with Slave functions. • Multiple I^2C slave addresses supported in hardware. • One slave address can be selectively qualified with a bit mask or an address range in order to respond to multiple I^2C bus addresses. • 10-bit addressing supported with software assist. • Supports System Management Bus (SMBus). • Separate DMA requests for Master, Slave, and Monitor functions. • No chip clocks are required in order to receive and compare an address as a Slave, so this event can wake up the device from deep-sleep mode. Additionally, I^2C0 can optionally generate a wake-up from power down. • Automatic modes optionally allow less software overhead for some use cases. K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 170.

’474 Patent Claim	Representative NXP Product(s)

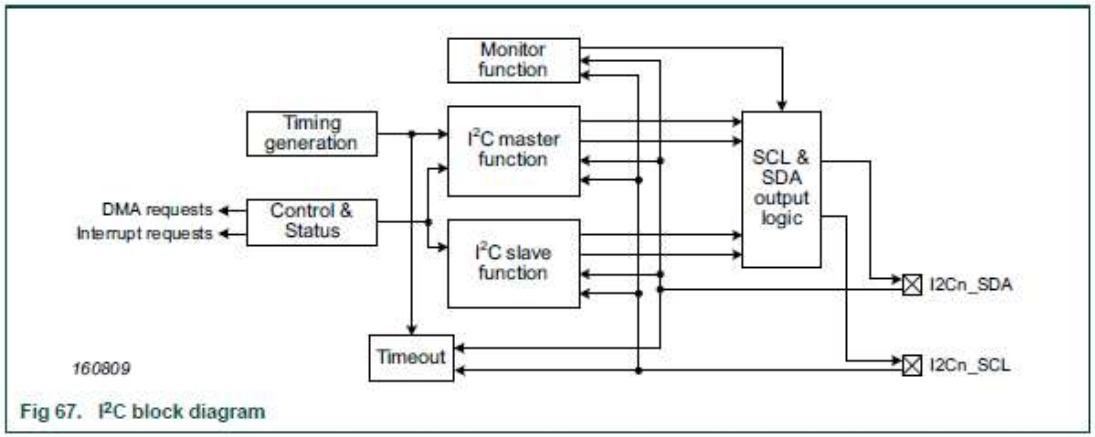


Fig 67. I²C block diagram

K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 176.

Table 46. I²C-bus pin description

Function	Type	Pin name used in data sheet Pin Description	Description
SCL	I/O	I2Cn_SCL	I²C serial clock
SDA	I/O	I2Cn_SDA	I²C serial data

K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 170.

Table 47. I²C bus pin assignments

Pin	Possible Pin Assignment	
I2C0_SCL	PIO0_10[1]	PIO0_15
I2C0_SDA	PIO0_11[1]	PIO0_16
I2C1_SCL	PIO0_6	PIO0_12
I2C1_SDA	PIO0_7	PIO0_13
I2C2_SCL	Internal connection	
I2C2_SDA	Internal connection	

K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 171.

’474 Patent Claim	Representative NXP Product(s)
	25.4.2 I²C receive/transmit in slave mode In this example, I²C1 is used as an I²C slave. The slave receives 8 bits from the master and then sends 8 bits to the master. The SCL and SDA functions must be enabled on suitable pins, see Table 48. The pins should be configured as required for the I²C-bus mode. The transmission of the address and data bits is controlled by the STAT[SLVPENDING] status bit. Whenever the status is Slave pending, the slave can acknowledge (“ack”) or send or receive an address and data. The received data or the data to be sent to the master are available in the SLVDAT register. After sending and receiving data, continue to the next step of the transmission protocol by writing to the SLVCTL register. K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 173. **25.4.2.1 Slave read from master** This example uses polling to control the sequence and does not use interrupts. Configure the I²C as a slave with address x: 1. Write the slave address x to the address 0 match register. 2. Set the CFG[SLVEN] bit to 1. K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 173.
[14d.] wherein the slave serial interface is responsive to a read temperature command issued by the master serial interface to return to the master serial interface a temperature value associated with the microprocessor.	In each of the Accused ’474 Wireless Microcontrollers, the slave serial interface is responsive to a read temperature command issued by the master serial interface to return to the master serial interface a temperature value associated with the microprocessor. For example, in each of the Accused ’474 Wireless Microcontrollers, the slave serial interface (*e.g.*, I2C identified above) is responsive to a read temperature command (*e.g.*, the read tempearture command requesting the result derived from the temperature sensor measurement) issued by the master serial interface (*e.g.*, the I2C interface on the master/MCU and/or the host processor identified above) to return to the master serial interface a temperature value (*e.g.*, the result derived from the temperature

’474 Patent Claim	Representative NXP Product(s)
	sensor measurement) associated with the microprocessor identified above. **25.4.2.2 Slave write to master** This example uses polling to control the sequence and does not use interrupts. Configure the I^2C as a slave with address x: 1. Write the slave address x to the address 0 match register. 2. Set the CFG[SLVEN] bit to 1. Write data to the master: 1. Wait for the pending status to be set (STAT[SLVPENDING] = 1) by polling the STAT register. Check the status register STAT[SLVSTATE] indicating ADDR. If not then an error has occurred. 2. ACK the address by setting SLVCTL[SLVCONTINUE] = 1 in the slave control register. 3. Wait for the pending status to be set (STAT[SLVPENDING] = 1) by polling the STAT register. Check the status register STAT[SLVSTATE] is indicating TX. If not then an error has occurred. 4. Write 8 bits of data to SLVDAT register. 5. Continue the transaction by setting SLVCTL[SLVCONTINUE] = 1 in the slave control register. K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 174.

’474 Patent Claim	Representative NXP Product(s)
	27.2 Features • 12-bit successive approximation analog to digital converter. • Input multiplexing among up to 8 pins (6 external inputs, 1 temperature sensor and V_{BAT}). • A configurable conversion sequencer with configurable trigger • Optional automatic high/low threshold comparison and “zero crossing” detection. • 12-bit conversion rate of 190 kHz. Options for reduced resolution at higher conversion rates. • Burst conversion mode for single or multiple inputs. • Asynchronous operation. Asynchronous mode allows choosing ADC clock from FRO12M or XO32M. • A temperature sensor is connected to ADC channel 7, see Chapter 28 “Temperature Sensor” for further details. • Supply monitor is connected to ADC channel 6; this monitors V_{BAT}. K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 198. Configure the temperature sensor as follows: • Select the temperature sensor as source for channel 7 of the ADC by writing the SEQ_CTRL[CHANNELS] bits to 0x80. In order to return ADC channel 7 to measuring its related device pin, write the SEQ_CTRL[CHANNELS] bits to 0x80. • The digital temperature reading is available after an analog-to-digital conversion of ADC channel 7. **Remark:** To convert the ADC conversion result into a temperature reading, use the API provided. This uses device specific calibration data stored in the device to increase the accuracy of the temperature reading. K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 199.

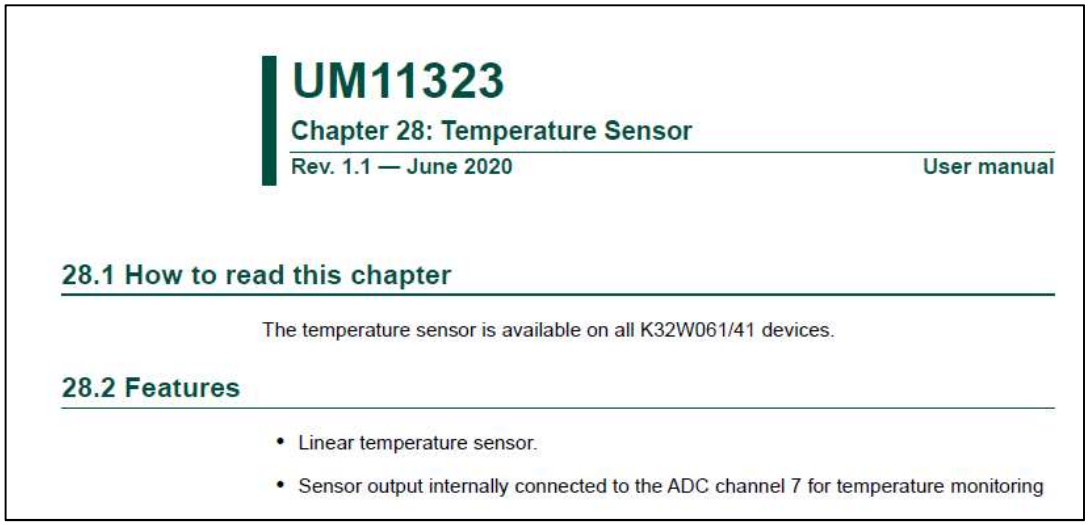
’474 Patent Claim	Representative NXP Product(s)

Table 58. ADC channels

ADC channel	Function	Device Pin
0	ADC0	PIO14
1	ADC1	PIO15
2	ADC2	PIO16
3	ADC3	PIO17
4	ADC4	PIO18
5	ADC5	PIO19
6	Supply monitor	Internal function
7	Temperature sensor	Internal function

K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 200.

See also:



UM11323

Chapter 28: Temperature Sensor

Rev. 1.1 — June 2020 User manual

28.1 How to read this chapter

The temperature sensor is available on all K32W061/41 devices.

28.2 Features

- Linear temperature sensor.
- Sensor output internally connected to the ADC channel 7 for temperature monitoring

K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 208.

’474 Patent Claim	Representative NXP Product(s)
	28.3 Basic configuration This section explains how the Temperature Sensor can be used. For a functional example see lpc_adc_basic. • Enable the power to the temperature sensor by setting the ASYNC_SYSCON_TEMPSENSORCTRL[ENABLE] . • Configure temperature sensor common mode output voltage setting ASYNC_SYSCON_TEMPSENSORCTRL[CM] = 0x2 for proper default operation • To monitor the temperature continually, select the temperature sensor as source for channel 7 of ADC0. See Chapter 27. The digital temperature reading is available after an analog-to-digital conversion. • The ADC reading must be converted into a temperature reading. To increase accuracy the sensor and ADC are calibrated during production, An API is provided to produce a temperature value; this performs the best configuration of the ADC for the purpose of the temperature sensor. The calibration data and other characteristics of the temperature and ADC are used to produce a high accuracy results. • For highest accuracy, set ADCCLK mux source to be 32 MHz XTAL with a divider setting of 7, to give an ADCCLK of 4 MHz. • The voltage range of operation of the ADC is set by ADC_GPADC_CTRL0[TEST]. In normal mode, the ADC can take an input voltage of 0 to 3.6 V, to V_{BAT} if this is lower. For the temperature sensor, the ADC must be configured in Unity Gain mode when the input voltage range is 0 to 0.9 V. Since the temperature sensor voltage output is within this range, the best accuracy is achieved. A consequence of this is that the temperature sensor can not be combined with the other ADC inputs as part of sequencer configuration. Also, safe practice is to set the mode back to normal mode after using the ADC with the temperature sensor. K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 208. **28.3.1 Perform a single ADC conversion with the temperature sensor as ADC input** As mentioned in the previous chapter, the API should be used when performing temperature measurements. As a simple example of obtaining a temperature measurement, the following steps can be performed. In this case, the accuracy is not as high as that using the API. K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 208.

’474 Patent Claim	Representative NXP Product(s)
	To perform a single ADC conversion for ADC0 channel 7 using the temperature sensor output: 1. Enable the temperature sensor output as input to ADC channel 7. 2. Configure the system clock and the ADC for operation. 3. Select the asynchronous mode in the ADC_CTRL register. 4. Select ADC channel 7 to perform the conversion by setting the ADC_SEQ_CTRL[CHANNELS] bits to 0x80. 5. Set the ADC_SEQ_CTRL[START] bit to 1. 6. Read the SEQ_GDAT[RESULT] bits for the conversion result. 7. The AHI software may be used to generate the temperature value. In fact, the example driver will perform this sequencing as well as making corrections due to the calibration data, and using averaging to give the best result. K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 209.

’474 Patent Claim	Representative NXP Product(s)

17.1.3 ADC Conversion Sequence Control Register (SEQ_CTRL)

Offset

Register	Offset
SEQ_CTRL	8h

Function

This register controls triggering and channel selection for conversion sequence. Also specifies interrupt mode for sequence. All ADC conversions are controlled through this sequencer which can be used for a single conversion or sets of conversions on one or more channels.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SEQ_ENA	MODE	Reserved	SINGLES...	BURST		STAR T_B...		Reserved				SYNC BYP...	TRIGPOL	TRIGGER	
W						START										
Reset	0	0	u	0	0	u	0	u	u	u	u	u	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TRIGGER				Reserved				CHANNELS							
W																
Reset	0	0	0	0	u	u	u	u	0	0	0	0	0	0	0	0

K32W061/K32W041 Register Manual (Rev. 1.1, 06/2020) at 440.

7-0 CHANNELS	ADC Channels Select which one or more of the ADC channels will be sampled and converted when this sequence is launched. A 1 in any bit of this field will cause the corresponding channel to be included in the conversion sequence, where bit 0 corresponds to channel 0, bit 1 to channel 1 and so forth. Bit 6 is channel 6; the supply monitor. Bit 7 is channel 7; the temperature sensor. When this conversion sequence is triggered, either by a hardware trigger or via software command, ADC conversions will be performed on each enabled channel, in sequence, beginning with the lowest-ordered channel. Remark: This field can ONLY be changed while SEQ_ENA (bit 31) is LOW. It is allowed to change this field and set bit 31 in the same write.

K32W061/K32W041 Register Manual (Rev. 1.1, 06/2020) at 443.

’474 Patent Claim	Representative NXP Product(s)

17.1.4 ADC Sequence Global Data Register (SEQ_GDAT)

Offset

Register	Offset
SEQ_GDAT	10h

Function

This register contains the result of the most recent ADC conversion completed under each conversion sequence. Results of ADC conversions can be read in one of two ways. One is to use this register to read data from the ADC at the end of each ADC conversion. The other is to read the individual ADC Channel Data (DATn) registers, typically after the entire sequence has completed. It is recommended to use one method consistently for a given conversion sequence.

This register is useful in conjunction with DMA operation - particularly when the channels selected for conversion are not sequential (hence the addresses of the individual result registers will not be sequential, making it difficult for the DMA engine to address them). For interrupt-driven code, it will more likely be advantageous to wait for an entire sequence to complete and then retrieve the results from the individual channel registers.

NOTE

The method to be employed for each sequence should be reflected in the SEQ_CTRL[MODE] bit since this will impact interrupt and overrun flag generation.

K32W061/K32W041 Register Manual (Rev. 1.1, 06/2020) at 443.

15-4 RESULT	ADC Conversion Result This field contains the 12-bit ADC conversion result from the most recent conversion performed under conversion sequence associated with this register. DATAVALID = 1 indicates that this result has not yet been read. If less than 12-bit resolultion is used the ADC result occupies the upper MSBs and unused LSBs should be ignored.

K32W061/K32W041 Register Manual (Rev. 1.1, 06/2020) at 445.

’474 Patent Claim	Representative NXP Product(s)

17.1.5 ADC Channel a Data Register (DAT0 - DAT7)

Function

These registers hold the result of the last conversion completed for each ADC channel. They also include status bits to indicate when a conversion has been completed, when a data overrun has occurred, and where the most recent conversion fits relative to the range dictated by the high and low threshold registers.

Results of ADC conversion can be read in one of two ways. One is to use the SEQ_GDAT register for each of the sequences to read data from the ADC at the end of each ADC conversion. The other is to use these individual ADC Channel Data registers, typically after the entire sequence has completed. It is recommended to use one method consistently for a given conversion sequence.

NOTE

The method to be employed for each sequence should be reflected in the MODE bit in the SEQ_CTRL register since this will impact interrupt and overrun flag generation.

The information presented in the DAT registers always pertains to the most recent conversion completed on that channel regardless of what sequence requested the conversion or which trigger caused it.

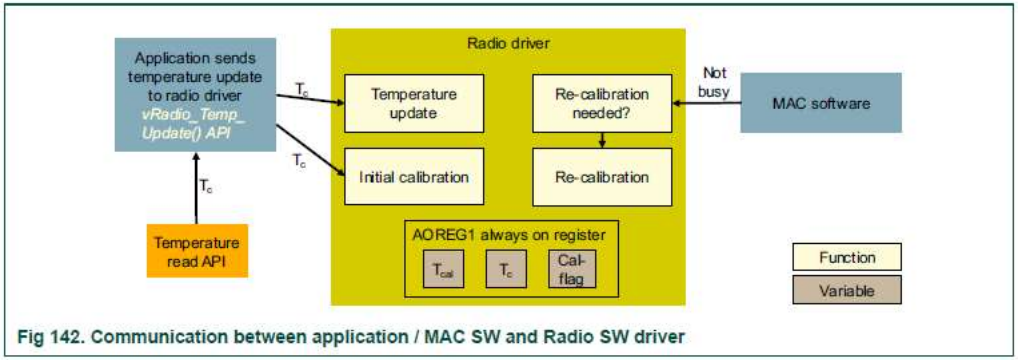
The OVERRUN fields for each channel are also replicated in the FLAGS register.

K32W061/K32W041 Register Manual (Rev. 1.1, 06/2020) at 445-446.

15-4 RESULT	ADC Conversion Result This field contains the 12-bit ADC conversion result from the most recent conversion performed under conversion sequence associated with this register. DATAVALID = 1 indicates that this result has not yet been read. If less than 12-bit resolultion is used, the ADC result occupies the upper MSBs and unused LSBs should be ignored.

K32W061/K32W041 Register Manual (Rev. 1.1, 06/2020) at 447.

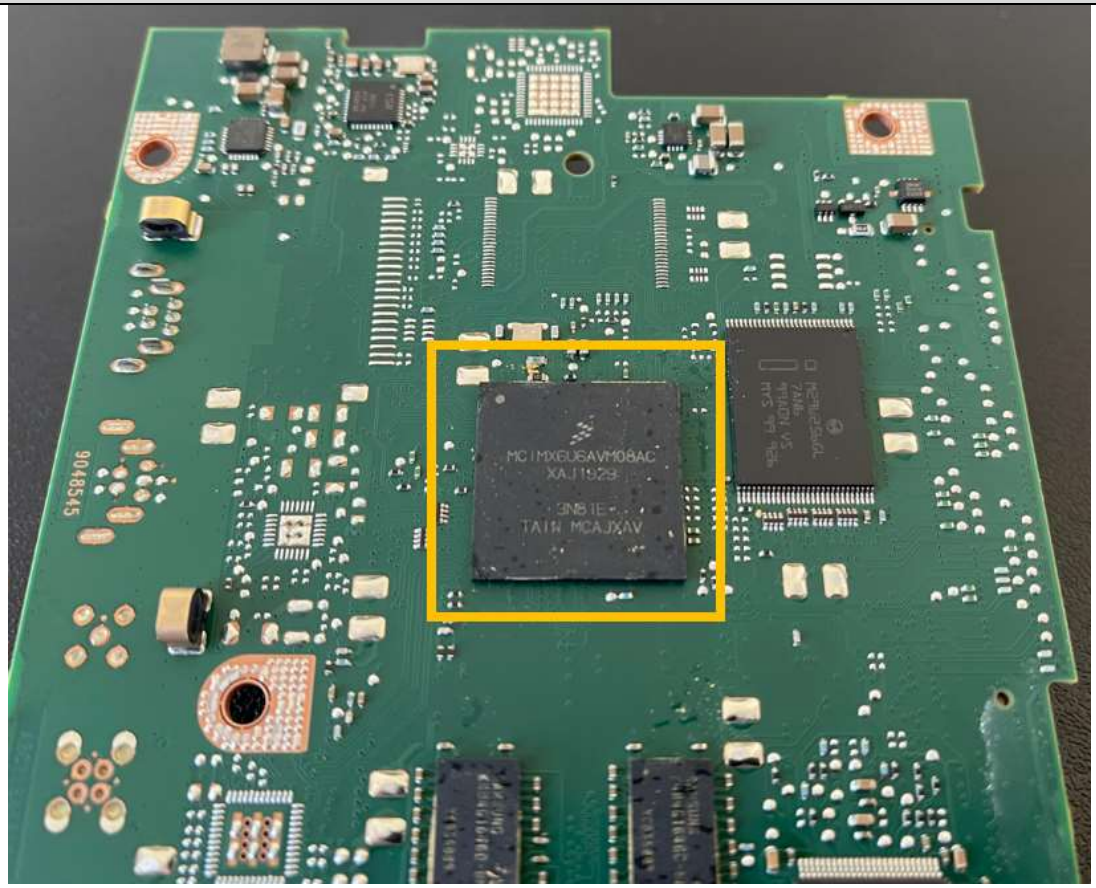
See also:

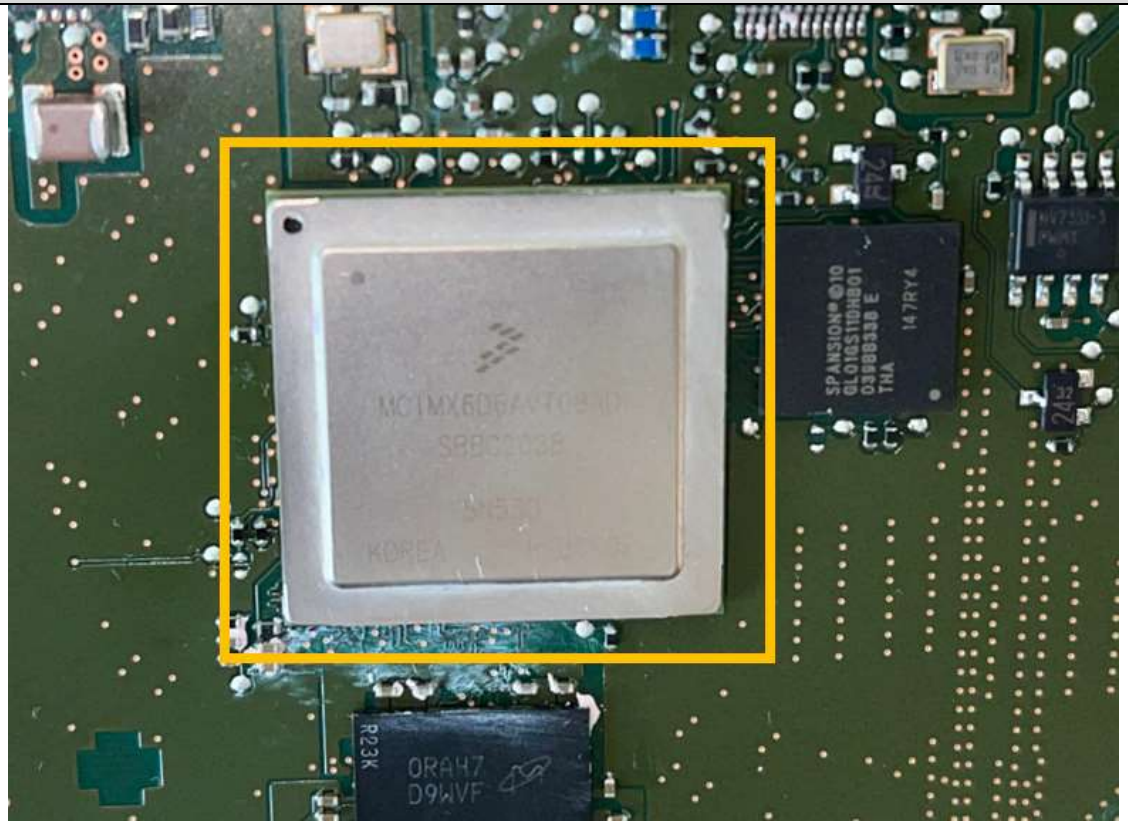
’474 Patent Claim	Representative NXP Product(s)
	 Fig 142. Communication between application / MAC SW and Radio SW driver K32W061/K32W041 User Manual, UM11323 (Rev. 1.1 — June 2020) at 324.

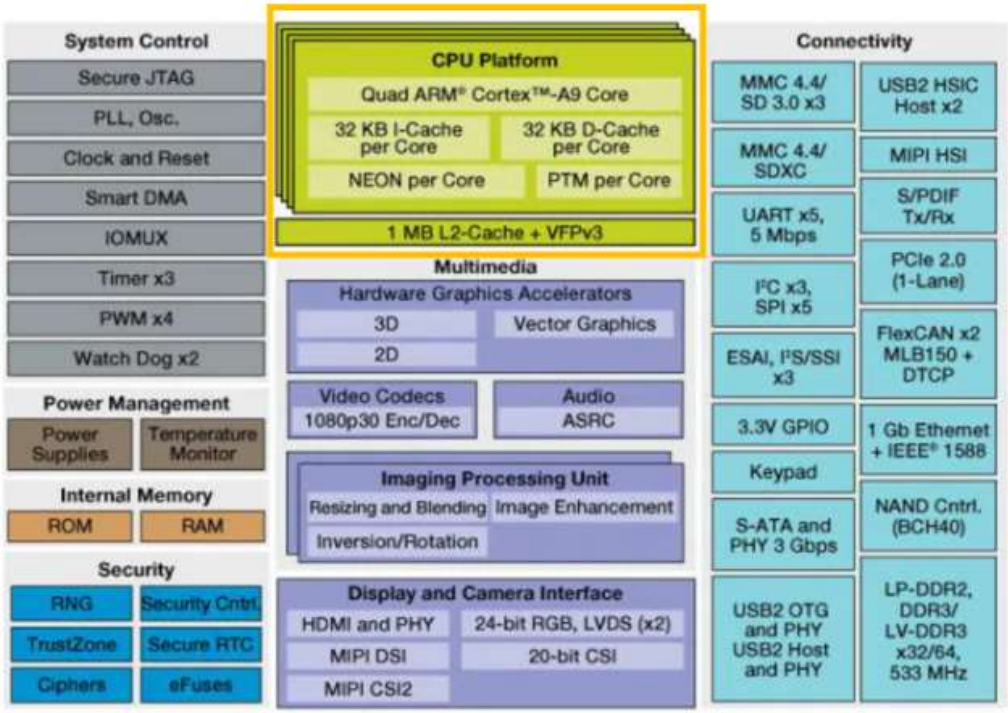
’474 Patent Claim	Representative NXP Product(s)
[1a.] A serial communication system comprising:	To the extent the preamble is limiting, the Accused ’474 i.MX Processors include a “serial communication system” as recited in the ’474 patent. Exemplary systems and serial interfaces are identified in the block diagrams below. *See, e.g.,* :  https://www.avnet.com/shop/us/products/nxp/mcimx6d6avt08ac-30744573456396267 03/

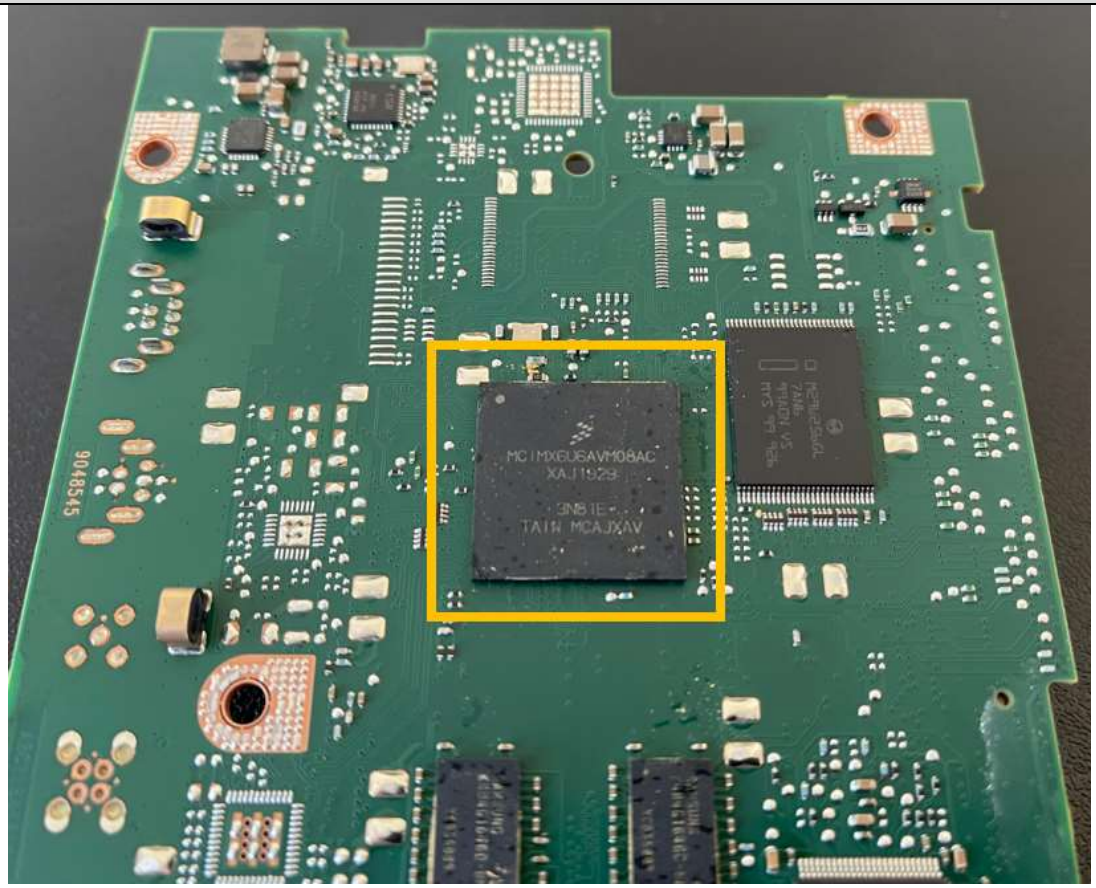
’474 Patent Claim	Representative NXP Product(s)
	 MCIMX6Q6AVT10AD Development Board (Arrow)

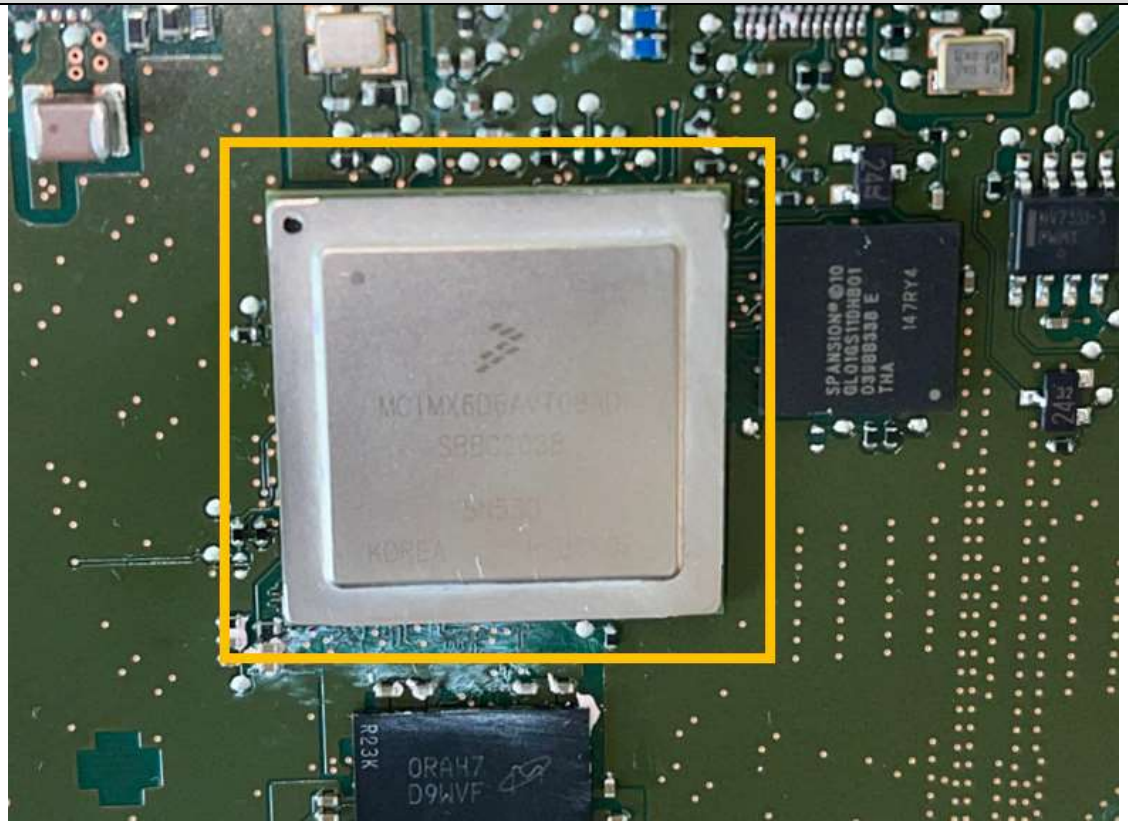
’474 Patent Claim	Representative NXP Product(s)
	 MCIMX6Q6AVT10AD Development Board (Mouser)  MCIMX6DP6AVT8AA (Mouser)

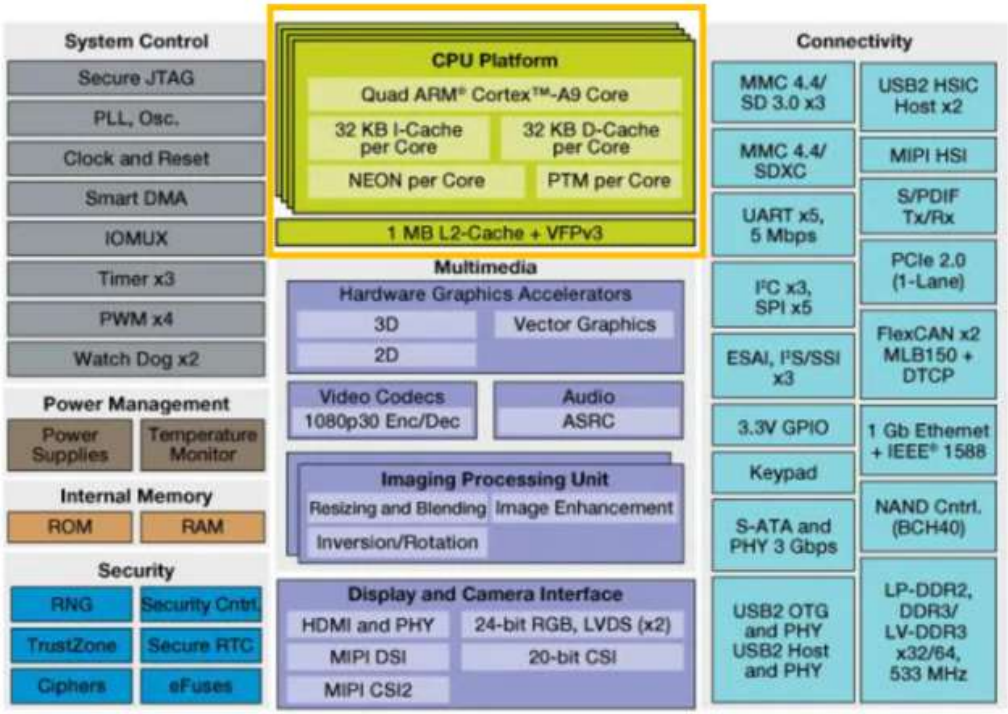
’474 Patent Claim	Representative NXP Product(s)
	 Continental Model VP2RFP (MCIMX6U6AVM08AC)

’474 Patent Claim	Representative NXP Product(s)
	 Bosch Model ICAM2-ECU V2 (MCIMX6D6AVT08AD)

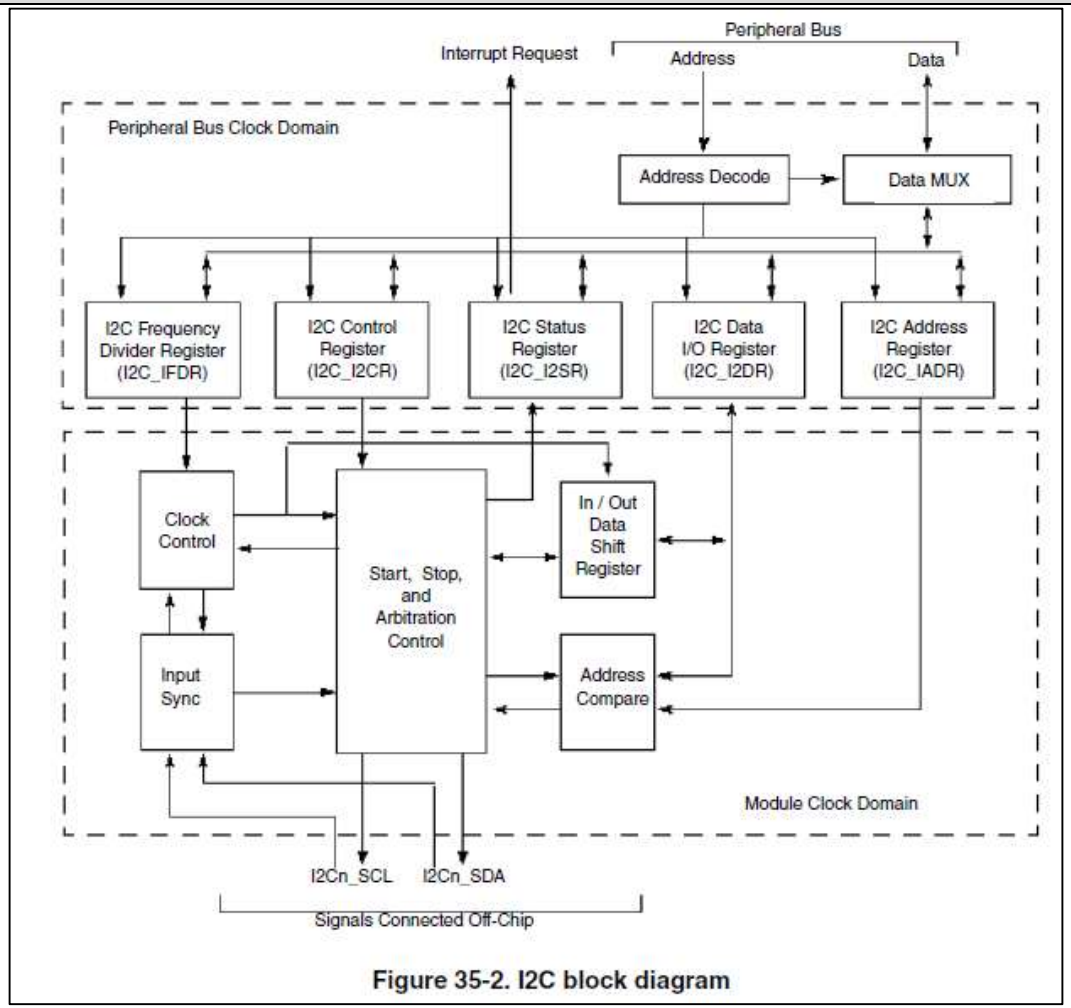
’474 Patent Claim	Representative NXP Product(s)
	 https://www.nxp.com/products/processors-and-microcontrollers/arm-processors/i-mx-applications-processors/i-mx-6-processors/i-mx-6quad-processors-high-performance-3d-graphics-hd-video-arm-cortex-a9-core:i.MX6Q
[1b.] an integrated circuit having a master serial interface; and	The Accused ’474 i.MX Processors each includes an integrated circuit having a master serial interface. For example, the Accused ’474 i.MX Processors each includes an integrated circuit having a master serial interface (*e.g.*, the I2C interface on the master/MCU and/or the host processor). *See, e.g.*:

’474 Patent Claim	Representative NXP Product(s)
	

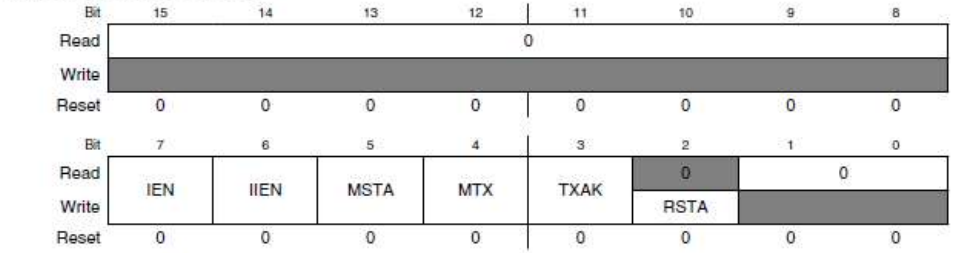
’474 Patent Claim	Representative NXP Product(s)
	

’474 Patent Claim	Representative NXP Product(s)
	 https://www.nxp.com/products/processors-and-microcontrollers/arm-processors/i-mx-applications-processors/i-mx-6-processors/i-mx-6quad-processors-high-performance-3d-graphics-hd-video-arm-cortex-a9-core:i.MX6Q

’474 Patent Claim	Representative NXP Product(s)
	Chapter 35 **I2C Controller (I2C)** **35.1 Overview** This chapter describes block-level operation and programming of I2C. The chapter is intended for a block-driver software developer. To understand how the block is integrated at the SoC level, a system software developer should see discussions of the block in the appropriate SoC-level chapter(s). **References:** This document assumes an understanding of the following document: • *The I2C Bus Specification*, Version 2.1, by Philips Semiconductor The Inter IC (I2C) provides functionality of a standard I2C slave and master. The I2C is designed to be compatible with the standard NXP I2C bus protocol. **NOTE** Three independent I2C channels are available. I2C is a two-wire, bidirectional serial bus that provides a simple, efficient method of data exchange, minimizing the interconnection between devices. This bus is suitable for applications requiring occasional communications over a short distance between many devices. The flexible I2C standard allows additional devices to be connected to the bus for expansion and system development. See the connection diagram in the figure below. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1893.

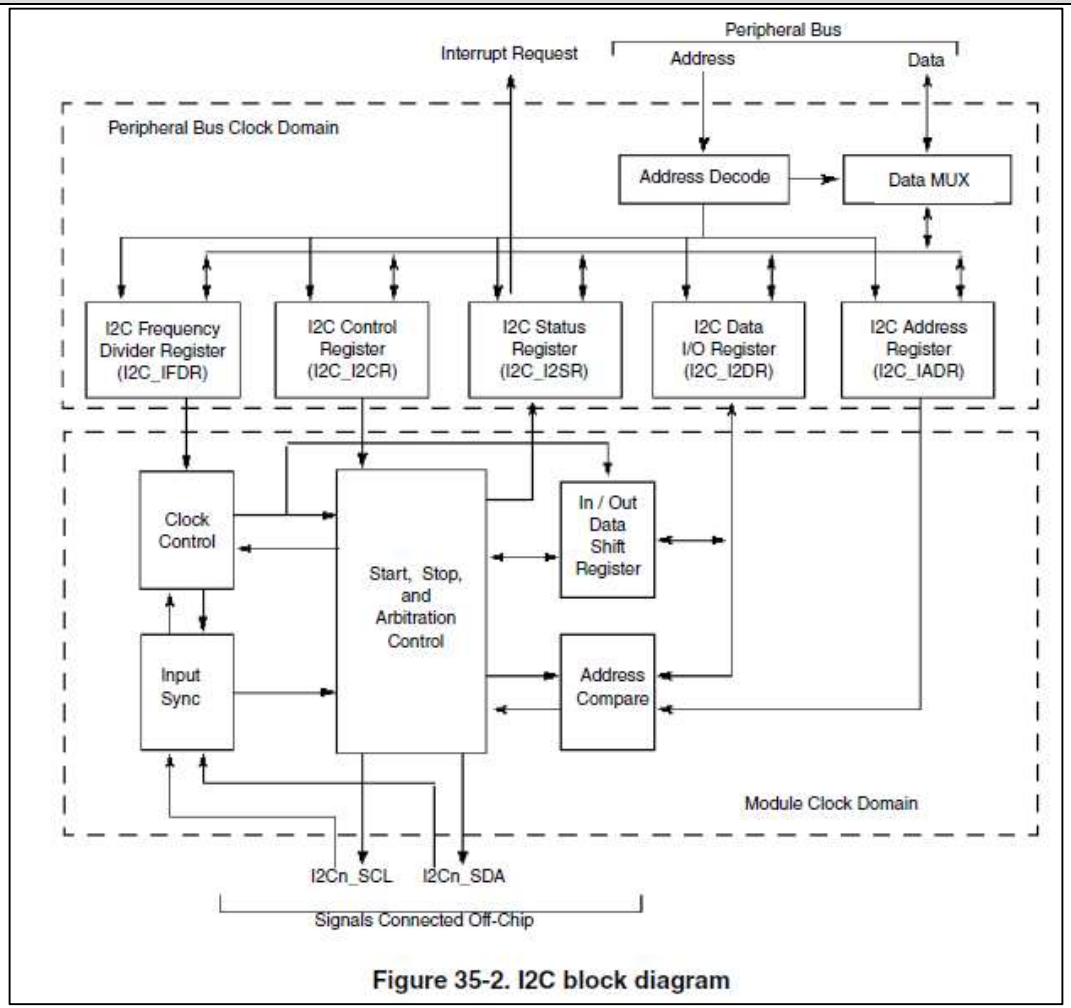
’474 Patent Claim	Representative NXP Product(s)
	 **Figure 35-2. I2C block diagram** i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1895.

’474 Patent Claim	Representative NXP Product(s)
	35.4.1 I2C system configuration After a reset, the I2C defaults to Slave Receive operations. Thus, when not operating as a master or responding to a slave transmit address, the I2C defaults to the Slave Receive state. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1897. **NOTE** The I2C is designed to be compatible with the Philips™ I2C bus protocol. For information on system configuration, protocol, and restrictions, see the *I2C Bus Specification*, version 2.1, by Philips Semiconductors. The I2C supports Standard and Fast modes only. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1898.

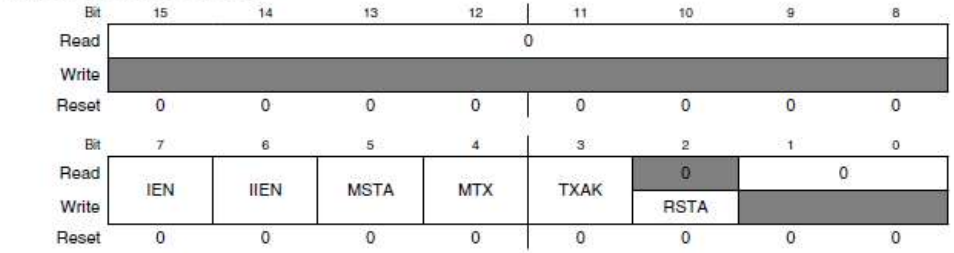
’474 Patent Claim	Representative NXP Product(s)
	35.7.3 I2C Control Register (I2Cx_I2CR) The I2C_I2CR is used to enable the I2C and the I2C interrupt. It also contains bits that govern operation as a slave or a master. Address: Base address + 8h offset  i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1913. **35.7.4 I2C Status Register (I2Cx_I2SR)** The I2C_I2SR contains bits that indicate transaction direction and status. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1914.

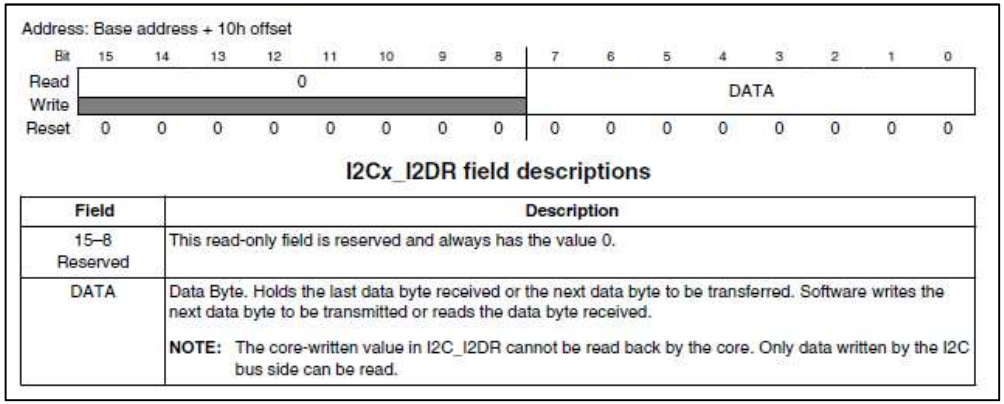
’474 Patent Claim	Representative NXP Product(s)
	35.7.5 I2C Data I/O Register (I2Cx_I2DR) In Master Receive mode, reading the data register allows a read to occur and initiates the next byte to be received. In Slave mode, the same function is available after it is addressed. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1916.
[1c.] a processor having a slave serial interface coupled to the master serial interface through a clock signal line and a data signal line	The Accused ’474 i.MX Processors each includes a processor having a slave serial interface coupled to the master serial interface through a clock signal line and a data signal line. For example, the Accused ’474 i.MX Processors each includes a processor having a slave serial interface (*e.g.*, the I2C interface on the slave processor) coupled to the master serial interface identified above through a clock signal line (*e.g.*, SCL) and a data signal line (*e.g.*, SDA).

’474 Patent Claim	Representative NXP Product(s)
	Chapter 35 **I2C Controller (I2C)** **35.1 Overview** This chapter describes block-level operation and programming of I2C. The chapter is intended for a block-driver software developer. To understand how the block is integrated at the SoC level, a system software developer should see discussions of the block in the appropriate SoC-level chapter(s). **References:** This document assumes an understanding of the following document: • *The I2C Bus Specification*, Version 2.1, by Philips Semiconductor The Inter IC (I2C) provides functionality of a standard I2C slave and master. The I2C is designed to be compatible with the standard NXP I2C bus protocol. **NOTE** Three independent I2C channels are available. I2C is a two-wire, bidirectional serial bus that provides a simple, efficient method of data exchange, minimizing the interconnection between devices. This bus is suitable for applications requiring occasional communications over a short distance between many devices. The flexible I2C standard allows additional devices to be connected to the bus for expansion and system development. See the connection diagram in the figure below. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1893.

’474 Patent Claim	Representative NXP Product(s)
	 **Figure 35-2. I2C block diagram** i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1895.

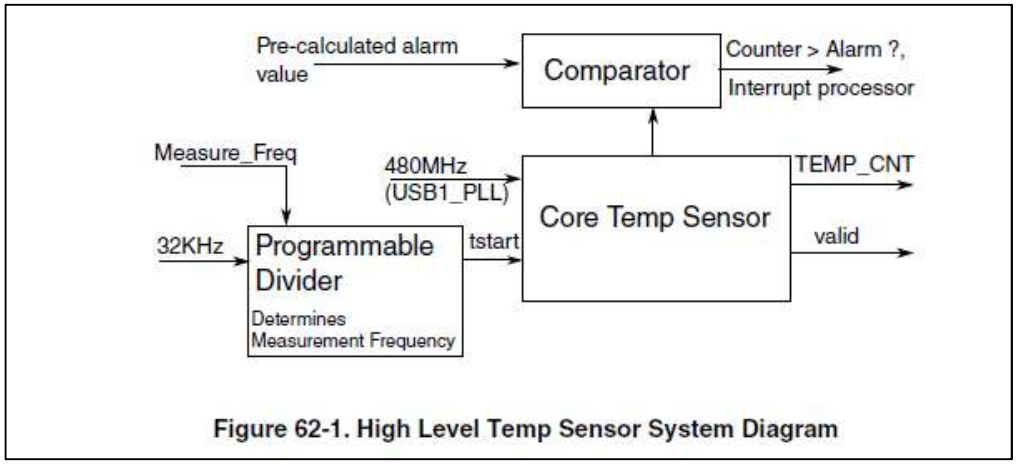
’474 Patent Claim	Representative NXP Product(s)
	35.4.1 I2C system configuration After a reset, the I2C defaults to Slave Receive operations. Thus, when not operating as a master or responding to a slave transmit address, the I2C defaults to the Slave Receive state. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1897. **NOTE** The I2C is designed to be compatible with the Philips™ I2C bus protocol. For information on system configuration, protocol, and restrictions, see the *I2C Bus Specification*, version 2.1, by Philips Semiconductors. The I2C supports Standard and Fast modes only. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1898.

’474 Patent Claim	Representative NXP Product(s)
	35.7.3 I2C Control Register (I2Cx_I2CR) The I2C_I2CR is used to enable the I2C and the I2C interrupt. It also contains bits that govern operation as a slave or a master. Address: Base address + 8h offset  i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1913. **35.7.4 I2C Status Register (I2Cx_I2SR)** The I2C_I2SR contains bits that indicate transaction direction and status. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1914.

’474 Patent Claim	Representative NXP Product(s)
	35.7.5 I2C Data I/O Register (I2Cx_I2DR) In Master Receive mode, reading the data register allows a read to occur and initiates the next byte to be received. In Slave mode, the same function is available after it is addressed. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1916.  **I2Cx_I2DR field descriptions** | Field | Description | | --- | --- | | 15–8 Reserved | This read-only field is reserved and always has the value 0. | | DATA | Data Byte. Holds the last data byte received or the next data byte to be transferred. Software writes the next data byte to be transmitted or reads the data byte received. **NOTE:** The core-written value in I2C_I2DR cannot be read back by the core. Only data written by the I2C bus side can be read. | i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1916.
[1d.] wherein the slave serial interface is responsive to a read temperature command issued by the master serial interface to return a temperature value associated with the	In each of the Accused ’474 i.MX Processors, the slave serial interface is responsive to a read temperature command issued by the master serial interface to return a temperature value associated with the processor. For example, in each of the Accused ’474 i.MX Processors, the slave serial interface identified above is responsive to a read temperature command (*e.g.*, the read tempearture command requesting the result derived from the temperature monitor “TEMPMON” measurement) issued by the identified master

’474 Patent Claim	Representative NXP Product(s)
processor.	serial interface to return a temperature value (*e.g.*, the result derived from the temperature monitor “TEMPMON” measurement) associated with the processor. **35.7.5 I2C Data I/O Register (I2Cx_I2DR)** In Master Receive mode, reading the data register allows a read to occur and initiates the next byte to be received. In Slave mode, the same function is available after it is addressed. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1916. Address: Base address + 10h offset Bit 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 Read 0 DATA Write Reset 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 **I2Cx_I2DR field descriptions** **Field** — **Description** 15–8 Reserved — This read-only field is reserved and always has the value 0. DATA — Data Byte. Holds the last data byte received or the next data byte to be transferred. Software writes the next data byte to be transmitted or reads the data byte received. **NOTE:** The core-written value in I2C_I2DR cannot be read back by the core. Only data written by the I2C bus side can be read. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1916.

’474 Patent Claim	Representative NXP Product(s)
	10.4.2.2 Thermal-aware power management The temperature sensor block (TEMPMON) implements a temperature sensor/conversion function. The block features an alarm function that can raise an interrupt signal if the temperature is above a specified threshold. Software may implement temperature aware DVFS for the ARM domain and the GPU domain, as well as temperature aware frequency scaling for other system components to ensure that both the frequency and voltage is lowered when the die temperature is above the specified limit. Software may also implement temperature aware task scheduling to ensure that non-critical tasks are suspended when the die temperature is above the specified limit. See Temperature Monitor (TEMPMON) for further details on temperature monitor functions and programmability options. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 539. **Chapter 62** **Temperature Monitor (TEMPMON)** **62.1 Overview** The temperature sensor module implements a temperature sensor/conversion function based on a temperature-dependent voltage to time conversion. The module features an alarm function that can raise an interrupt signal if the temperature is above a specified threshold. A self-repeating mode can also be programmed which executes a temperature sensing operation based on a programmed delay. Software can use this module to monitor the on-die temperature and take appropriate actions such as throttling back the core frequency when a temperature interrupt is set. The high-level implementation of the temperature sensor is shown in the figure below. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev

’474 Patent Claim	Representative NXP Product(s)
	2, 06/2014) at 5159.  **Figure 62-1. High Level Temp Sensor System Diagram** i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 5159. **62.2 Software Usage Guidelines** During normal system operation software can use the temperature sensor counter output (TEMP_CNT) in conjunction with the fused temperature calibration data to determine the on-die operational temperature or to set an over-temperature interrupt alarm to within a couple of °C. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 5160.

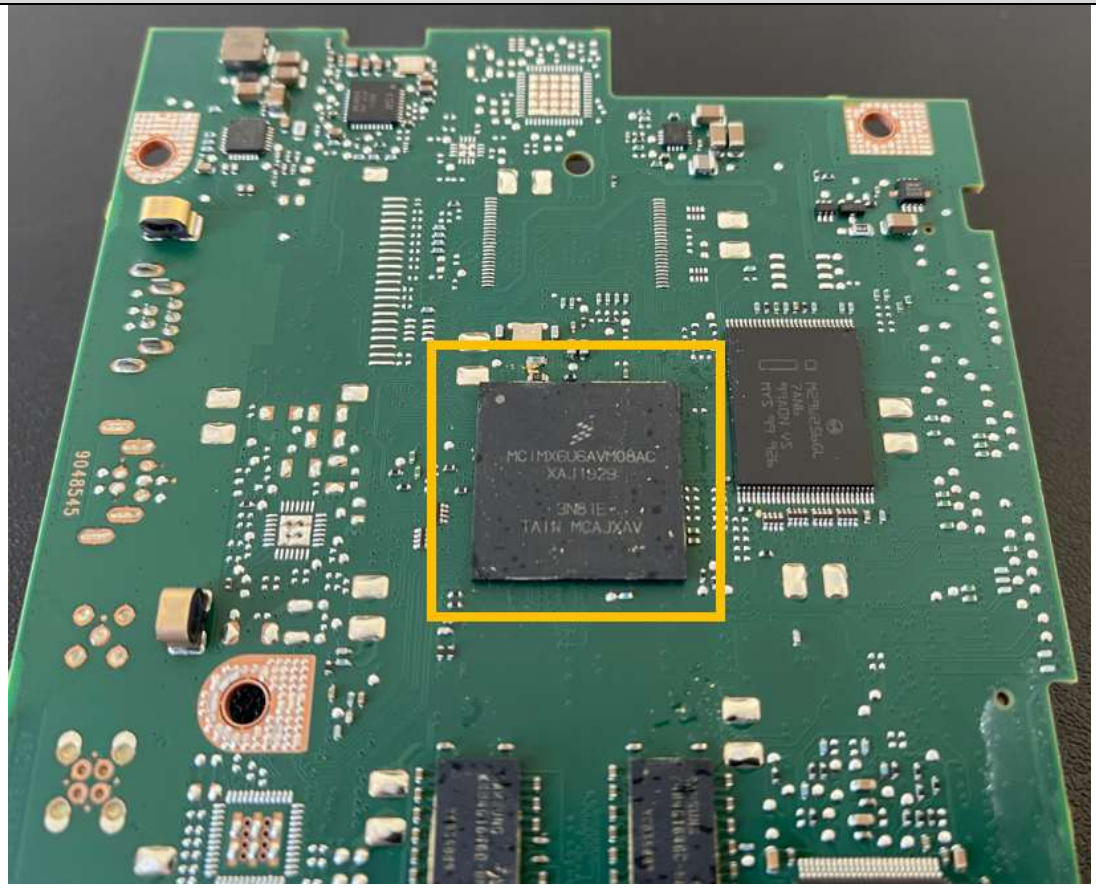
’474 Patent Claim	Representative NXP Product(s)
	TEMPMON_TEMPSENSE0*n* field descriptions **Field** / **Description** 31–20 ALARM_VALUE — This bit field contains the temperature count (raw sensor output) that will generate an alarm interrupt. 19–8 TEMP_CNT — This bit field contains the last measured temperature count. 7 - — This field is reserved. Reserved. 6 - — This field is reserved. Reserved. 5–3 - — This field is reserved. Reserved 2 FINISHED — Indicates that the latest temp is valid. This bit should be cleared by the sensor after the start of each measurement. 0 **INVALID** — Last measurement is not ready yet. 1 **VALID** — Last measurement is valid. 1 MEASURE_TEMP — Starts the measurement process. If the measurement frequency is zero in the TEMPSENSE1 register, this results in a single conversion. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 5163. The Accused ’474 i.MX Processors reads and returns the “TEMPMON” value in response to a read temperature command issued by an external I2C master. *See, e.g.*:

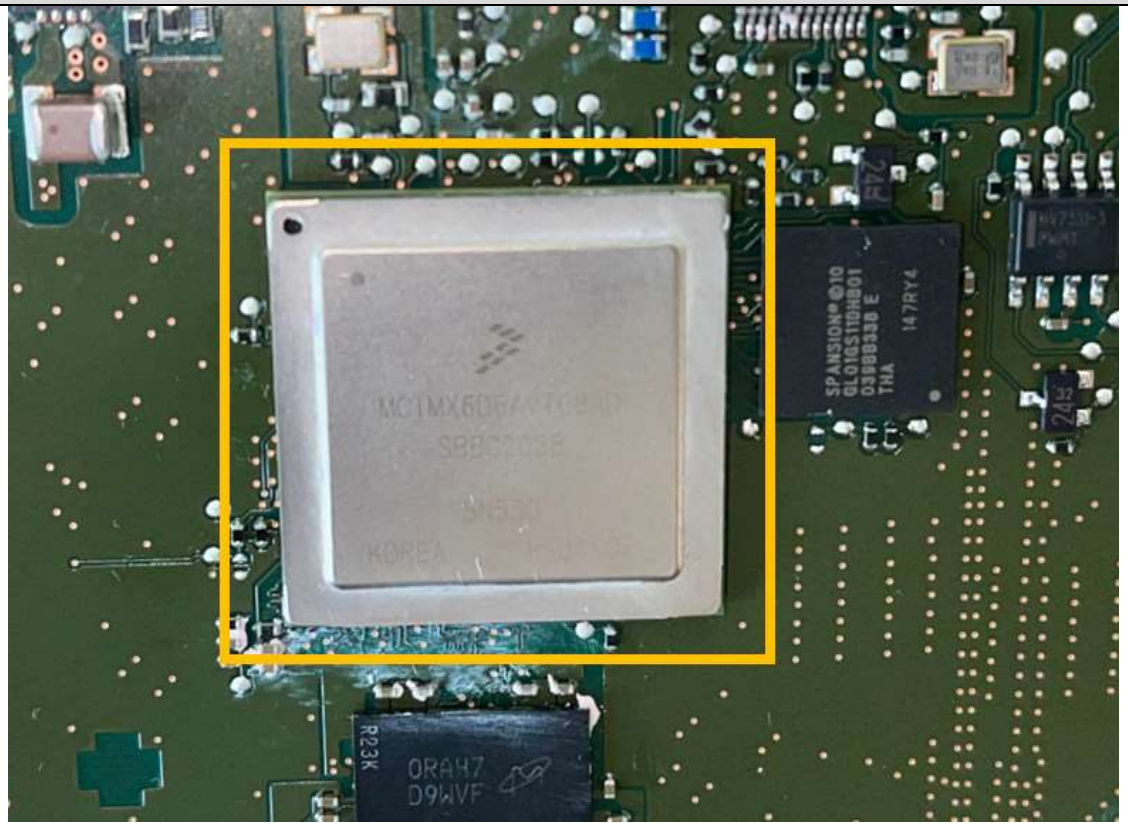
’474 Patent Claim	Representative NXP Product(s)
	`` if (status & I2SR_SRW) { /* Master wants to read from us*/ dev_dbg(&i2c_imx->adapter.dev, "read requested"); i2c_slave_event(i2c_imx->slave, I2C_SLAVE_READ_REQUESTED, &value); /* Slave transmit */ ctl |= I2CR_MTX; imx_i2c_write_reg(ctl, i2c_imx, IMX_I2C_I2CR); /* Send data */ imx_i2c_write_reg(value, i2c_imx, IMX_I2C_I2DR); } else { /* Master wants to write to us */ dev_dbg(&i2c_imx->adapter.dev, "write requested"); i2c_slave_event(i2c_imx->slave, I2C_SLAVE_WRITE_REQUESTED, &value); ``` https://patchwork.kernel.org/project/linux-arm-kernel/patch/20191009101802.19309-1-biwen.li@nxp.com/
[8a.]. A method for communicating over a point to point serial communication system having a clock signal line and a data signal line coupling a serial interface master and a serial interface slave, the method comprising:	To the extent the preamble is limiting, the Accused ’474 i.MX Processors perform a method for communicating over a point to point serial communication system having a clock signal line and a data signal line coupling a serial interface master and a serial interface slave. For example, each of the Accused ’474 i.MX Processors perform a method for communicating over a point to point serial communication system identified below having a clock signal line (*e.g.*, SCL) and a data signal line (*e.g.*, SDA) coupling a serial interface master (*e.g.*, the I2C interface on the master/MCU and/or the host processor) and a serial interface slave (*e.g.*, the I2C interface on the slave processor). *See, e.g.*:

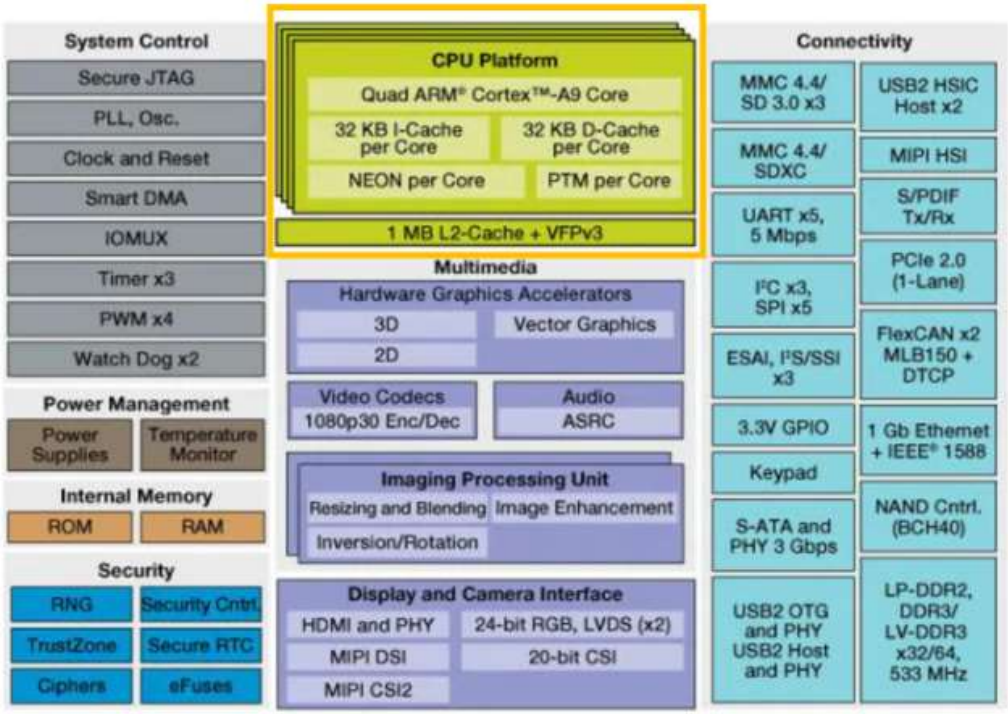
’474 Patent Claim	Representative NXP Product(s)
	 https://www.avnet.com/shop/us/products/nxp/mcimx6d6avt08ac-3074457345639626703/

’474 Patent Claim	Representative NXP Product(s)
	 MCIMX6Q6AVT10AD Development Board (Arrow)

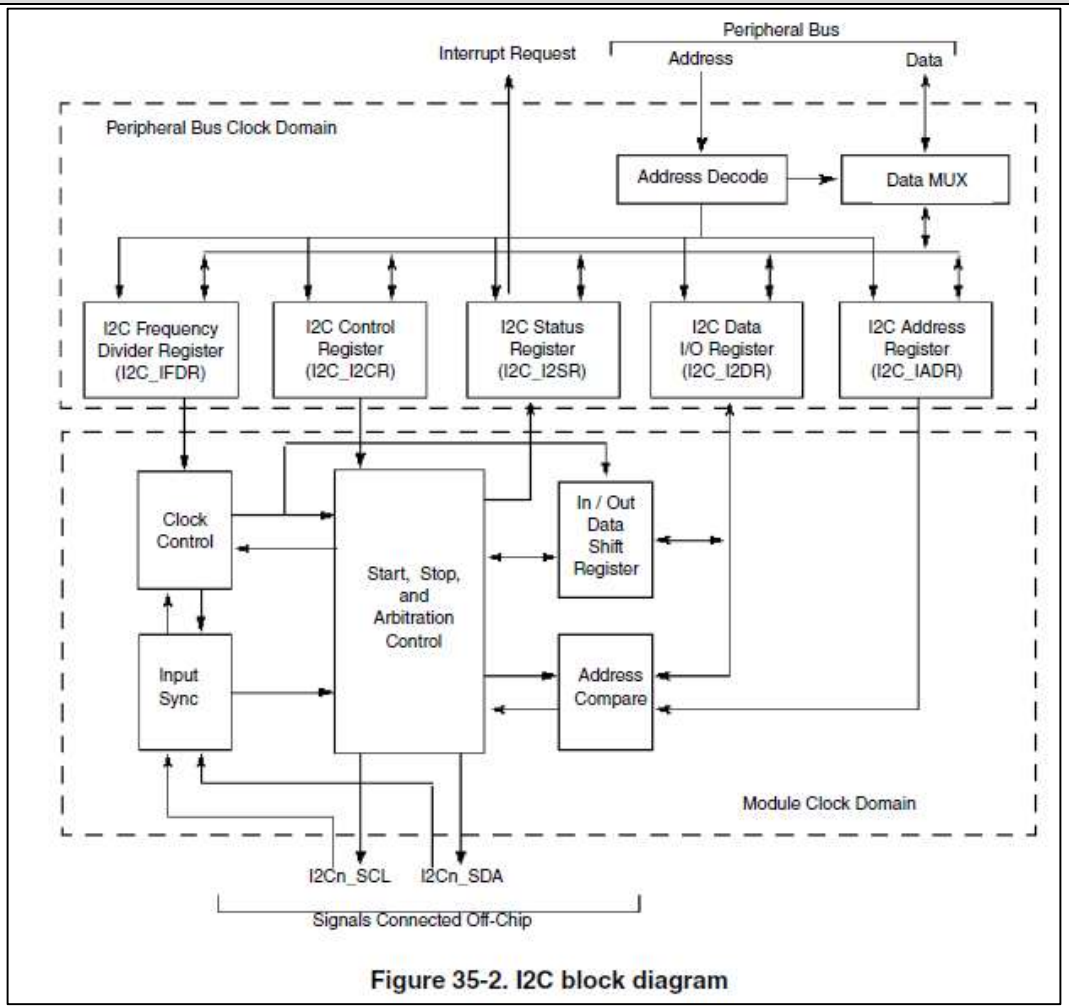
’474 Patent Claim	Representative NXP Product(s)
	 MCIMX6Q6AVT10AD Development Board (Mouser)  MCIMX6DP6AVT8AA (Mouser)

’474 Patent Claim	Representative NXP Product(s)
	 Continental Model VP2RFP (MCIMX6U6AVM08AC)

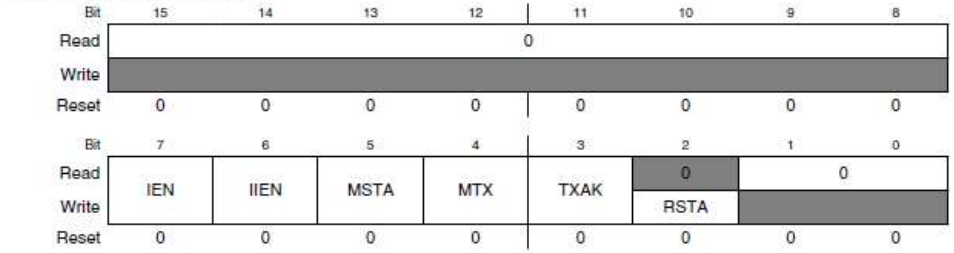
’474 Patent Claim	Representative NXP Product(s)
	 Bosch Model ICAM2-ECU V2 (MCIMX6D6AVT08AD)

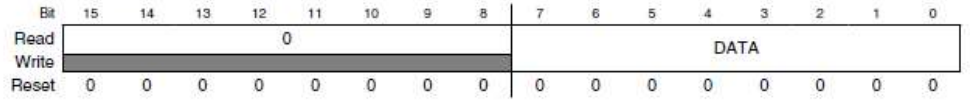
’474 Patent Claim	Representative NXP Product(s)
	 https://www.nxp.com/products/processors-and-microcontrollers/arm-processors/i-mx-applications-processors/i-mx-6-processors/i-mx-6quad-processors-high-performance-3d-graphics-hd-video-arm-cortex-a9-core:i.MX6Q

’474 Patent Claim	Representative NXP Product(s)
	Chapter 35 **I2C Controller (I2C)** **35.1 Overview** This chapter describes block-level operation and programming of I2C. The chapter is intended for a block-driver software developer. To understand how the block is integrated at the SoC level, a system software developer should see discussions of the block in the appropriate SoC-level chapter(s). **References:** This document assumes an understanding of the following document: • *The I2C Bus Specification*, Version 2.1, by Philips Semiconductor The Inter IC (I2C) provides functionality of a standard I2C slave and master. The I2C is designed to be compatible with the standard NXP I2C bus protocol. **NOTE** Three independent I2C channels are available. I2C is a two-wire, bidirectional serial bus that provides a simple, efficient method of data exchange, minimizing the interconnection between devices. This bus is suitable for applications requiring occasional communications over a short distance between many devices. The flexible I2C standard allows additional devices to be connected to the bus for expansion and system development. See the connection diagram in the figure below. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1893.

’474 Patent Claim	Representative NXP Product(s)
	 **Figure 35-2. I2C block diagram** i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1895.

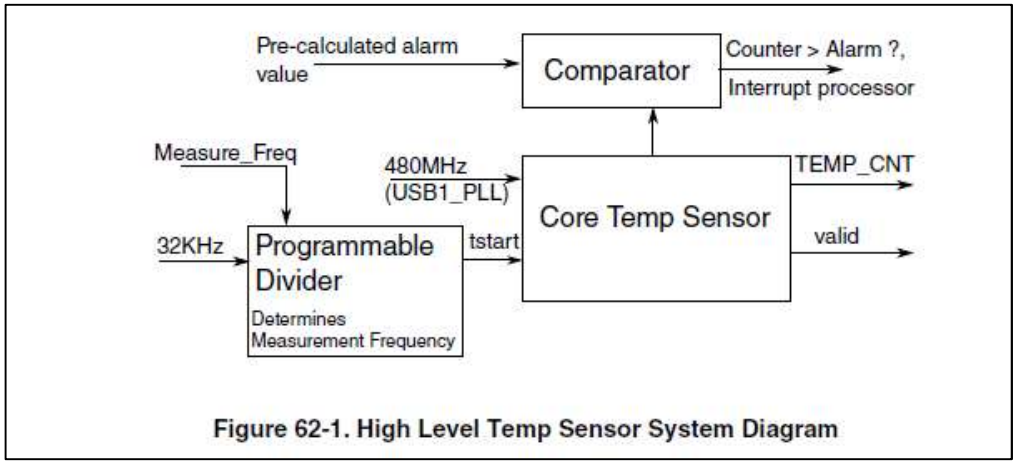
’474 Patent Claim	Representative NXP Product(s)
	35.4.1 I2C system configuration After a reset, the I2C defaults to Slave Receive operations. Thus, when not operating as a master or responding to a slave transmit address, the I2C defaults to the Slave Receive state. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1897. **NOTE** The I2C is designed to be compatible with the Philips™ I2C bus protocol. For information on system configuration, protocol, and restrictions, see the *I2C Bus Specification*, version 2.1, by Philips Semiconductors. The I2C supports Standard and Fast modes only. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1898.

’474 Patent Claim	Representative NXP Product(s)
	35.7.3 I2C Control Register (I2Cx_I2CR) The I2C_I2CR is used to enable the I2C and the I2C interrupt. It also contains bits that govern operation as a slave or a master. Address: Base address + 8h offset  i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1913. **35.7.4 I2C Status Register (I2Cx_I2SR)** The I2C_I2SR contains bits that indicate transaction direction and status. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1914.

’474 Patent Claim	Representative NXP Product(s)
	35.7.5 I2C Data I/O Register (I2Cx_I2DR) In Master Receive mode, reading the data register allows a read to occur and initiates the next byte to be received. In Slave mode, the same function is available after it is addressed. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1916. Address: Base address + 10h offset  **I2Cx_I2DR field descriptions** **Field** — **Description** 15–8 Reserved — This read-only field is reserved and always has the value 0. DATA — Data Byte. Holds the last data byte received or the next data byte to be transferred. Software writes the next data byte to be transmitted or reads the data byte received. **NOTE:** The core-written value in I2C_I2DR cannot be read back by the core. Only data written by the I2C bus side can be read. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1916.
[8b.] sending a read temperature command to the serial interface slave from the serial interface master using the clock signal line and the data signal line; and	The Accused ’474 i.MX Processors performs a step of sending a read temperature command to the serial interface slave from the serial interface master using the clock signal line and the data signal line. For example, each of the Accused ’474 i.MX Processors perform a step of sending a read temperature command (*e.g.*, the read tempearture command requesting the result derived from the temperature monitor “TEMPMON” measurement) to the serial interface slave identified above from the serial interface master identified above using the clock signal line identified above and the data signal line identified above.

’474 Patent Claim	Representative NXP Product(s)
	See, e.g., **35.7.5 I2C Data I/O Register (I2Cx_I2DR)** In Master Receive mode, reading the data register allows a read to occur and initiates the next byte to be received. In Slave mode, the same function is available after it is addressed. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1916. Address: Base address + 10h offset Bit 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 Read 0 DATA Write Reset 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 **I2Cx_I2DR field descriptions** **Field** — **Description** 15–8 Reserved — This read-only field is reserved and always has the value 0. DATA — Data Byte. Holds the last data byte received or the next data byte to be transferred. Software writes the next data byte to be transmitted or reads the data byte received. **NOTE:** The core-written value in I2C_I2DR cannot be read back by the core. Only data written by the I2C bus side can be read. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1916.

’474 Patent Claim	Representative NXP Product(s)
	10.4.2.2 Thermal-aware power management The temperature sensor block (TEMPMON) implements a temperature sensor/conversion function. The block features an alarm function that can raise an interrupt signal if the temperature is above a specified threshold. Software may implement temperature aware DVFS for the ARM domain and the GPU domain, as well as temperature aware frequency scaling for other system components to ensure that both the frequency and voltage is lowered when the die temperature is above the specified limit. Software may also implement temperature aware task scheduling to ensure that non-critical tasks are suspended when the die temperature is above the specified limit. See Temperature Monitor (TEMPMON) for further details on temperature monitor functions and programmability options. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 539. **Chapter 62** **Temperature Monitor (TEMPMON)** **62.1 Overview** The temperature sensor module implements a temperature sensor/conversion function based on a temperature-dependent voltage to time conversion. The module features an alarm function that can raise an interrupt signal if the temperature is above a specified threshold. A self-repeating mode can also be programmed which executes a temperature sensing operation based on a programmed delay. Software can use this module to monitor the on-die temperature and take appropriate actions such as throttling back the core frequency when a temperature interrupt is set. The high-level implementation of the temperature sensor is shown in the figure below. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev

’474 Patent Claim	Representative NXP Product(s)
	2, 06/2014) at 5159.  **Figure 62-1. High Level Temp Sensor System Diagram** i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 5159. **62.2 Software Usage Guidelines** During normal system operation software can use the temperature sensor counter output (TEMP_CNT) in conjunction with the fused temperature calibration data to determine the on-die operational temperature or to set an over-temperature interrupt alarm to within a couple of °C. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 5160.

’474 Patent Claim	Representative NXP Product(s)
	TEMPMON_TEMPSENSE0*n* field descriptions **Field** / **Description** 31–20 ALARM_VALUE — This bit field contains the temperature count (raw sensor output) that will generate an alarm interrupt. 19–8 TEMP_CNT — This bit field contains the last measured temperature count. 7 - — This field is reserved. Reserved. 6 - — This field is reserved. Reserved. 5–3 - — This field is reserved. Reserved 2 FINISHED — Indicates that the latest temp is valid. This bit should be cleared by the sensor after the start of each measurement. 0 **INVALID** — Last measurement is not ready yet. 1 **VALID** — Last measurement is valid. 1 MEASURE_TEMP — Starts the measurement process. If the measurement frequency is zero in the TEMPSENSE1 register, this results in a single conversion. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 5163. The Accused ’474 i.MX Processors reads and returns the “TEMPMON” value in response to a read temperature command issued by an external I2C master. *See, e.g.*:

’474 Patent Claim	Representative NXP Product(s)
	`` if (status & I2SR_SRW) { /* Master wants to read from us*/ dev_dbg(&i2c_imx->adapter.dev, "read requested"); i2c_slave_event(i2c_imx->slave, I2C_SLAVE_READ_REQUESTED, &value); /* Slave transmit */ ctl |= I2CR_MTX; imx_i2c_write_reg(ctl, i2c_imx, IMX_I2C_I2CR); /* Send data */ imx_i2c_write_reg(value, i2c_imx, IMX_I2C_I2DR); } else { /* Master wants to write to us */ dev_dbg(&i2c_imx->adapter.dev, "write requested"); i2c_slave_event(i2c_imx->slave, I2C_SLAVE_WRITE_REQUESTED, &value); ``` https://patchwork.kernel.org/project/linux-arm-kernel/patch/20191009101802.19309-1-biwen.li@nxp.com/
[8c.] in response to the read temperature command, the serial interface slave supplying over the data signal line a temperature value associated with a processor on an integrated circuit containing the serial interface slave.	The Accused ’474 i.MX Processors performs a step of in response to the read temperature command, the serial interface slave supplying over the data signal line a temperature value associated with a processor on an integrated circuit containing the serial interface slave. For example, each of the Accused ’474 i.MX Processors perform a step of in response to the read temperature command, the serial interface slave identified above supplying over the data signal line identified above a temperature value (*e.g.*, the result derived from the temperature monitor “TEMPMON” measurement) associated with a processor on an integrated circuit containing the serial interface slave identified above. *See, e.g.*,

’474 Patent Claim	Representative NXP Product(s)

35.7.5 I2C Data I/O Register (I2Cx_I2DR)

In Master Receive mode, reading the data register allows a read to occur and initiates the next byte to be received. In Slave mode, the same function is available after it is addressed.

i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1916.

Address: Base address + 10h offset

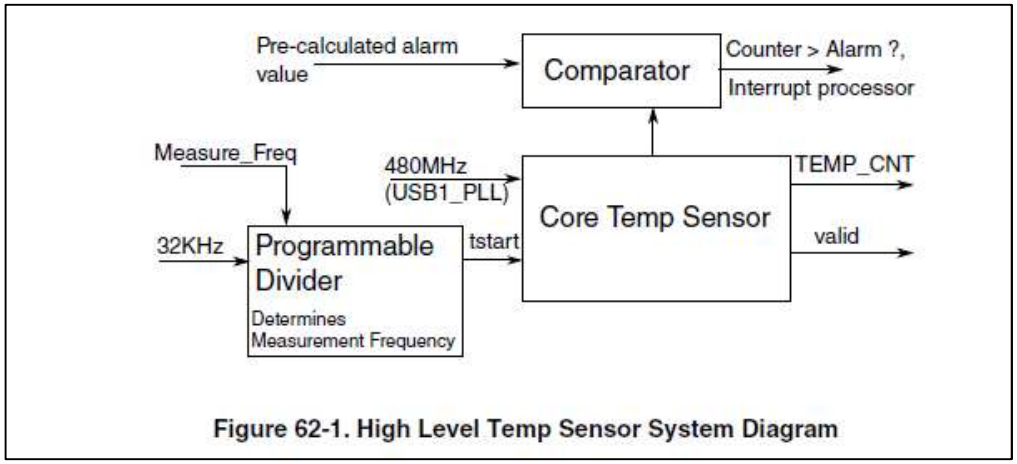
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0								DATA							
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Cx_I2DR field descriptions

Field	Description
15–8 Reserved	This read-only field is reserved and always has the value 0.
DATA	Data Byte. Holds the last data byte received or the next data byte to be transferred. Software writes the next data byte to be transmitted or reads the data byte received. **NOTE:** The core-written value in I2C_I2DR cannot be read back by the core. Only data written by the I2C bus side can be read.

i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1916.

’474 Patent Claim	Representative NXP Product(s)
	10.4.2.2 Thermal-aware power management The temperature sensor block (TEMPMON) implements a temperature sensor/conversion function. The block features an alarm function that can raise an interrupt signal if the temperature is above a specified threshold. Software may implement temperature aware DVFS for the ARM domain and the GPU domain, as well as temperature aware frequency scaling for other system components to ensure that both the frequency and voltage is lowered when the die temperature is above the specified limit. Software may also implement temperature aware task scheduling to ensure that non-critical tasks are suspended when the die temperature is above the specified limit. See Temperature Monitor (TEMPMON) for further details on temperature monitor functions and programmability options. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 539. **Chapter 62** **Temperature Monitor (TEMPMON)** **62.1 Overview** The temperature sensor module implements a temperature sensor/conversion function based on a temperature-dependent voltage to time conversion. The module features an alarm function that can raise an interrupt signal if the temperature is above a specified threshold. A self-repeating mode can also be programmed which executes a temperature sensing operation based on a programmed delay. Software can use this module to monitor the on-die temperature and take appropriate actions such as throttling back the core frequency when a temperature interrupt is set. The high-level implementation of the temperature sensor is shown in the figure below. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev

’474 Patent Claim	Representative NXP Product(s)
	2, 06/2014) at 5159.  **Figure 62-1. High Level Temp Sensor System Diagram** i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 5159. **62.2 Software Usage Guidelines** During normal system operation software can use the temperature sensor counter output (TEMP_CNT) in conjunction with the fused temperature calibration data to determine the on-die operational temperature or to set an over-temperature interrupt alarm to within a couple of °C. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 5160.

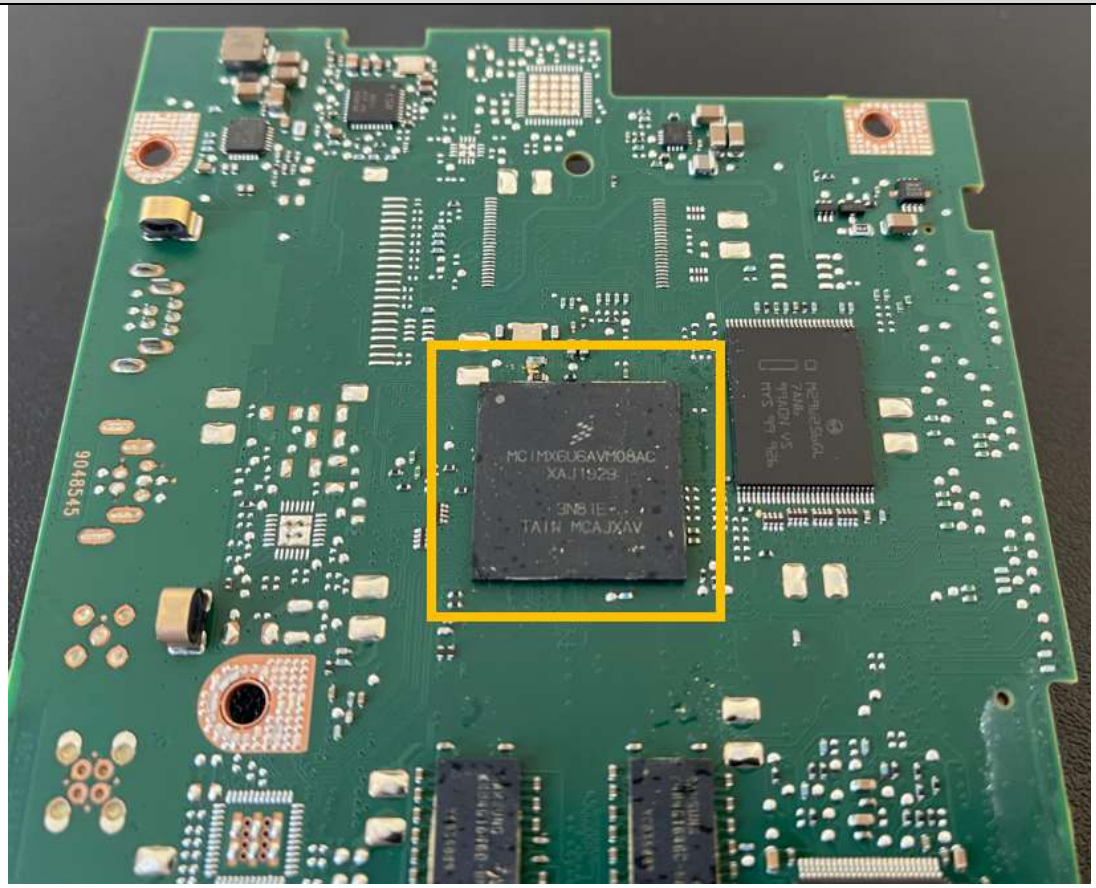
’474 Patent Claim	Representative NXP Product(s)
	TEMPMON_TEMPSENSE0*n* field descriptions **Field** / **Description** 31–20 ALARM_VALUE — This bit field contains the temperature count (raw sensor output) that will generate an alarm interrupt. 19–8 TEMP_CNT — This bit field contains the last measured temperature count. 7 - — This field is reserved. Reserved. 6 - — This field is reserved. Reserved. 5–3 - — This field is reserved. Reserved 2 FINISHED — Indicates that the latest temp is valid. This bit should be cleared by the sensor after the start of each measurement. 0 **INVALID** — Last measurement is not ready yet. 1 **VALID** — Last measurement is valid. 1 MEASURE_TEMP — Starts the measurement process. If the measurement frequency is zero in the TEMPSENSE1 register, this results in a single conversion. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 5163. The Accused ’474 i.MX Processors reads and returns the “TEMPMON” value in response to a read temperature command issued by an external I2C master. *See, e.g.*:

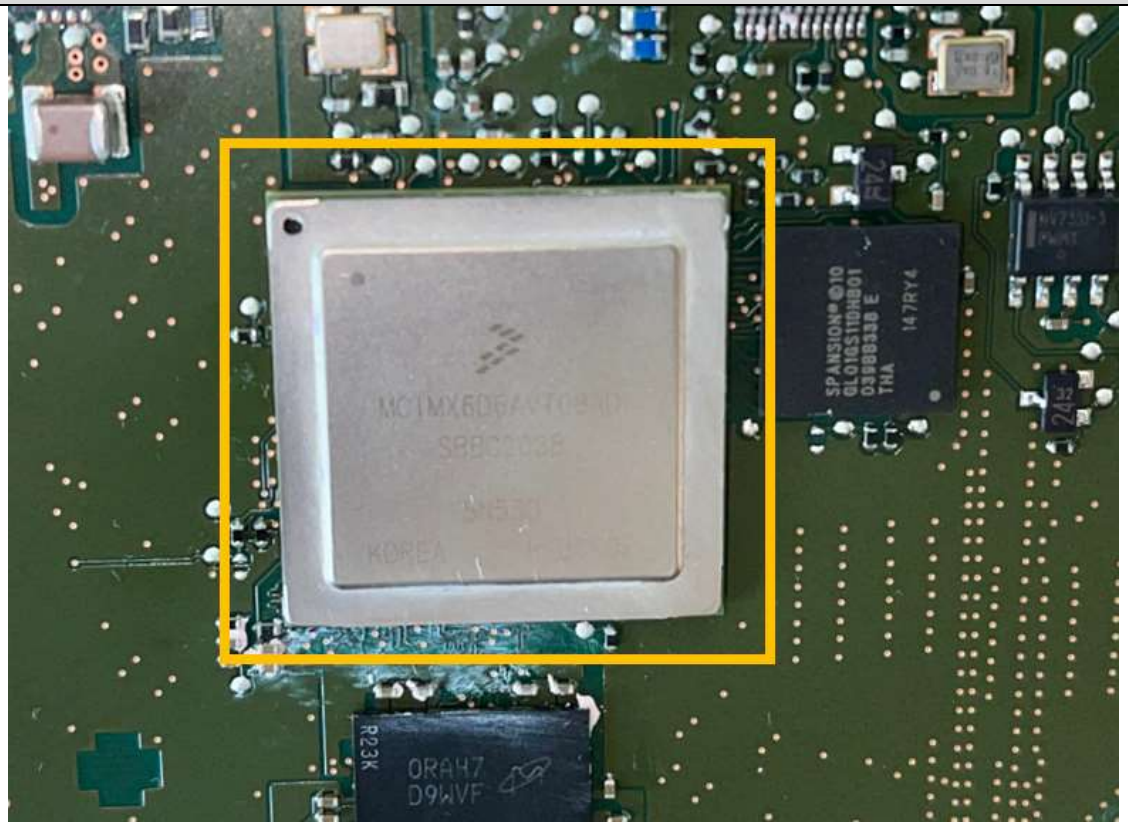
’474 Patent Claim	Representative NXP Product(s)
	`` if (status & I2SR_SRW) { /* Master wants to read from us*/ dev_dbg(&i2c_imx->adapter.dev, "read requested"); i2c_slave_event(i2c_imx->slave, I2C_SLAVE_READ_REQUESTED, &value); /* Slave transmit */ ctl |= I2CR_MTX; imx_i2c_write_reg(ctl, i2c_imx, IMX_I2C_I2CR); /* Send data */ imx_i2c_write_reg(value, i2c_imx, IMX_I2C_I2DR); } else { /* Master wants to write to us */ dev_dbg(&i2c_imx->adapter.dev, "write requested"); i2c_slave_event(i2c_imx->slave, I2C_SLAVE_WRITE_REQUESTED, &value); ``` https://patchwork.kernel.org/project/linux-arm-kernel/patch/20191009101802.19309-1-biwen.li@nxp.com/
[14a.] A serial communication system comprising:	To the extent the preamble is limiting, the Accused ’474 i.MX Processors include a “serial communication system” as recited in the ’474 patent. Exemplary systems and serial interfaces are identified in the block diagrams below. *See, e.g.*:

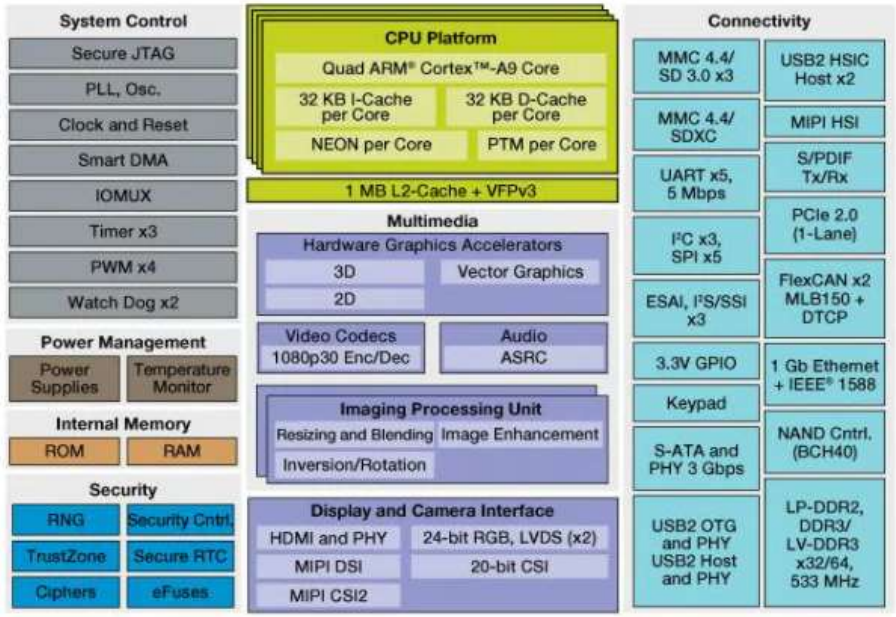
’474 Patent Claim	Representative NXP Product(s)
	 https://www.avnet.com/shop/us/products/nxp/mcimx6d6avt08ac-3074457345639626703/

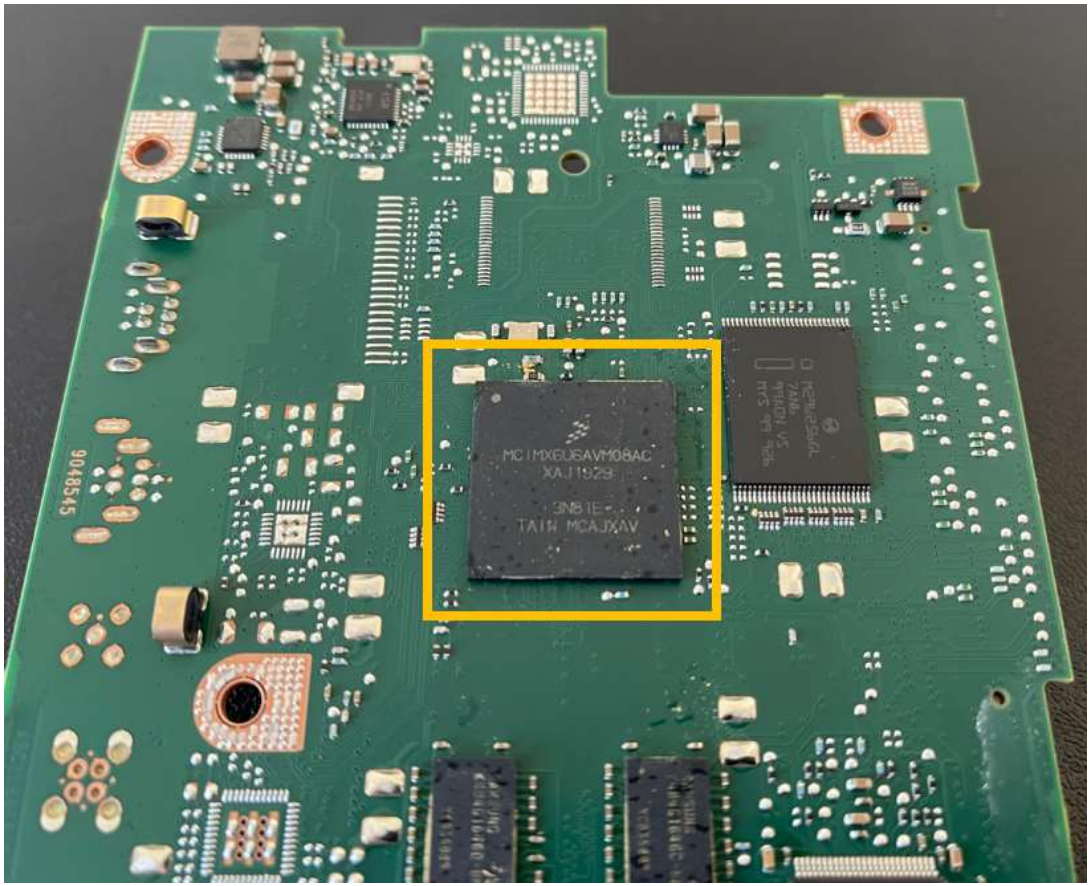
’474 Patent Claim	Representative NXP Product(s)
	 MCIMX6Q6AVT10AD Development Board (Arrow)

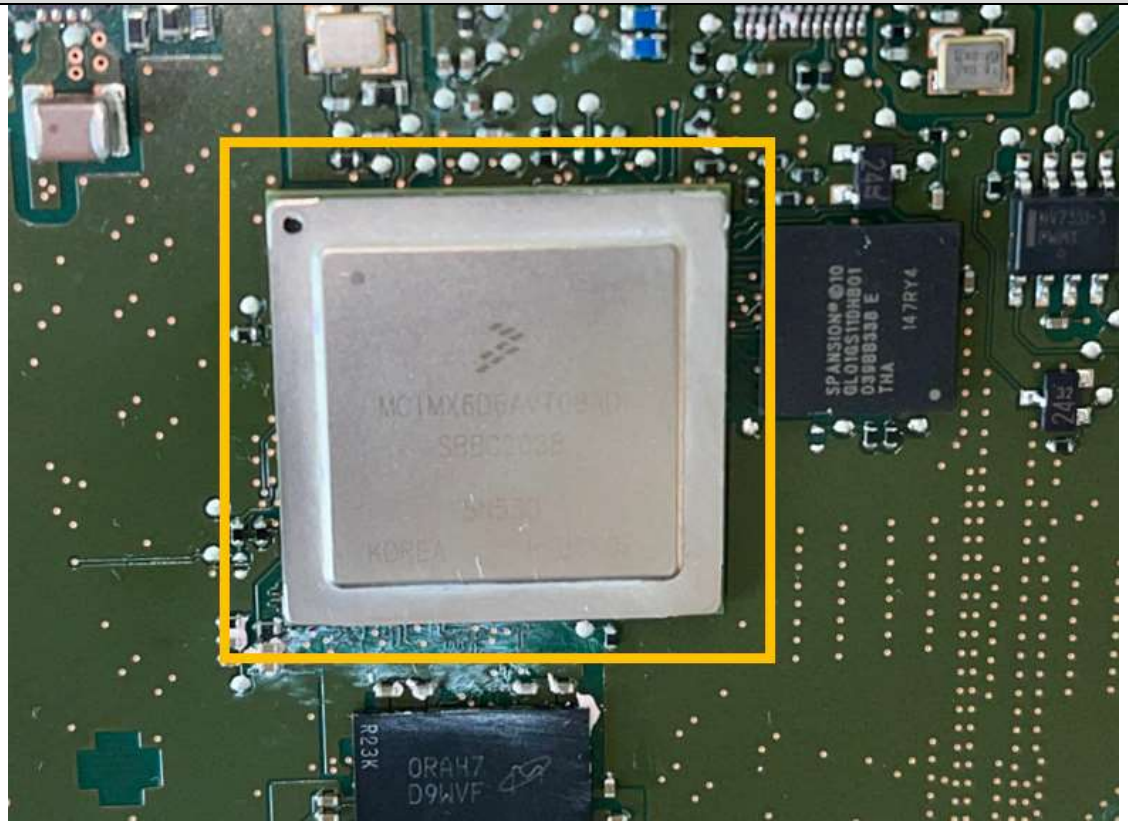
’474 Patent Claim	Representative NXP Product(s)
	 MCIMX6Q6AVT10AD Development Board (Mouser)  MCIMX6DP6AVT8AA (Mouser)

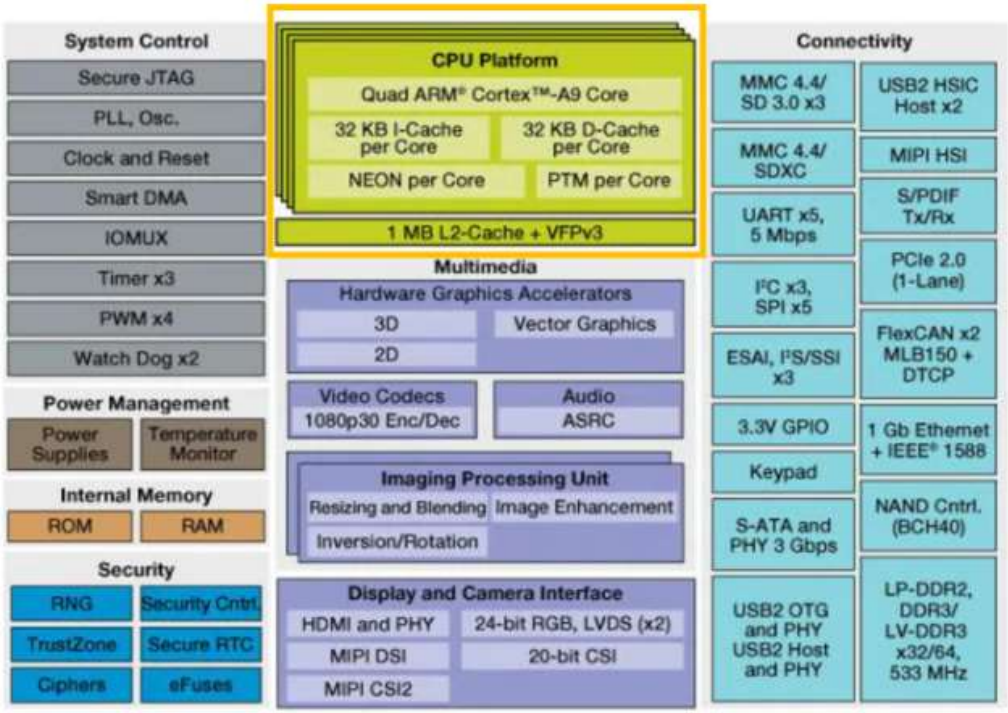
’474 Patent Claim	Representative NXP Product(s)
	 Continental Model VP2RFP (MCIMX6U6AVM08AC)

’474 Patent Claim	Representative NXP Product(s)
	 Bosch Model ICAM2-ECU V2 (MCIMX6D6AVT08AD)

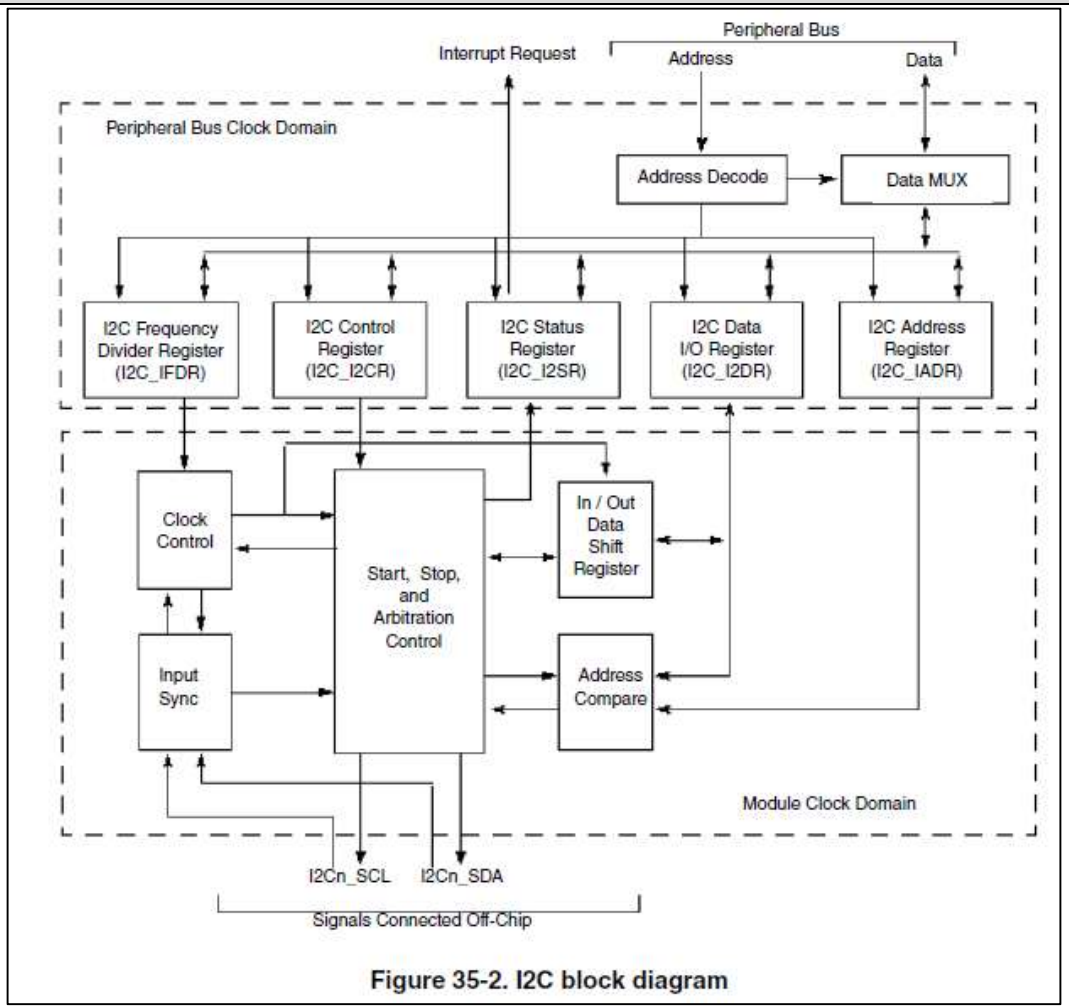
’474 Patent Claim	Representative NXP Product(s)
	 https://www.nxp.com/products/processors-and-microcontrollers/arm-processors/i-mx-applications-processors/i-mx-6-processors/i-mx-6quad-processors-high-performance-3d-graphics-hd-video-arm-cortex-a9-core:i.MX6Q
[14b.] a microprocessor having	The Accused ’474 i.MX Processors each includes a microprocessor. For example, the Accused ’474 i.MX Processors each constitutes a microprocessor because, among others, they include at the processing logics identified below. *See, e.g.*:

’474 Patent Claim	Representative NXP Product(s)
	

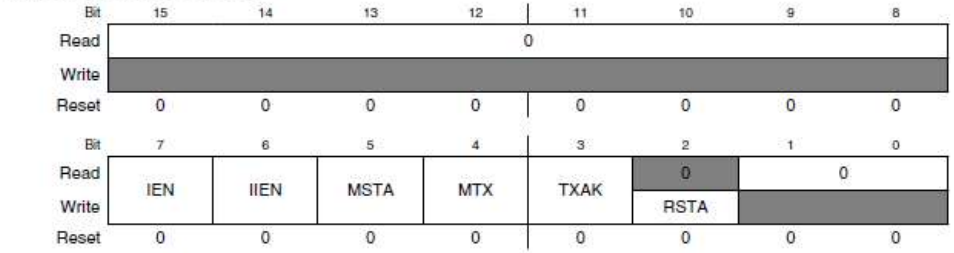
’474 Patent Claim	Representative NXP Product(s)
	

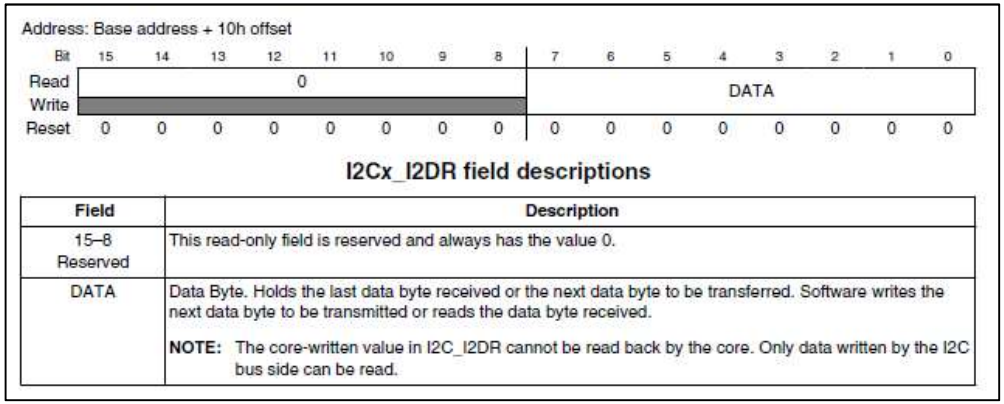
’474 Patent Claim	Representative NXP Product(s)
	 https://www.nxp.com/products/processors-and-microcontrollers/arm-processors/i-mx-applications-processors/i-mx-6-processors/i-mx-6quad-processors-high-performance-3d-graphics-hd-video-arm-cortex-a9-core:i.MX6Q
[14c.] a slave serial interface for coupling to a master serial interface through a clock signal line output terminal and a data signal line output terminal	In each of the Accused ’474 i.MX Processors, the microprocessor has a slave serial interface for coupling to a master serial interface through a clock signal line output terminal and a data signal line output terminal. For example, each of the Accused ’474 i.MX Processors has a slave serial interface (*e.g.*, I2C) for coupling to a master serial interface (*e.g.*, the I2C interface on the master/MCU and/or the host processor) through a clock signal line output terminal (*e.g.*, SCL) and a data signal line output terminal

’474 Patent Claim	Representative NXP Product(s)
	(*e.g.*, SDA)). **Chapter 35** **I2C Controller (I2C)** **35.1 Overview** This chapter describes block-level operation and programming of I2C. The chapter is intended for a block-driver software developer. To understand how the block is integrated at the SoC level, a system software developer should see discussions of the block in the appropriate SoC-level chapter(s). **References:** This document assumes an understanding of the following document: • *The I2C Bus Specification*, Version 2.1, by Philips Semiconductor The Inter IC (I2C) provides functionality of a standard I2C slave and master. The I2C is designed to be compatible with the standard NXP I2C bus protocol. **NOTE** Three independent I2C channels are available. I2C is a two-wire, bidirectional serial bus that provides a simple, efficient method of data exchange, minimizing the interconnection between devices. This bus is suitable for applications requiring occasional communications over a short distance between many devices. The flexible I2C standard allows additional devices to be connected to the bus for expansion and system development. See the connection diagram in the figure below. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1893.

’474 Patent Claim	Representative NXP Product(s)
	 **Figure 35-2. I2C block diagram** i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1895.

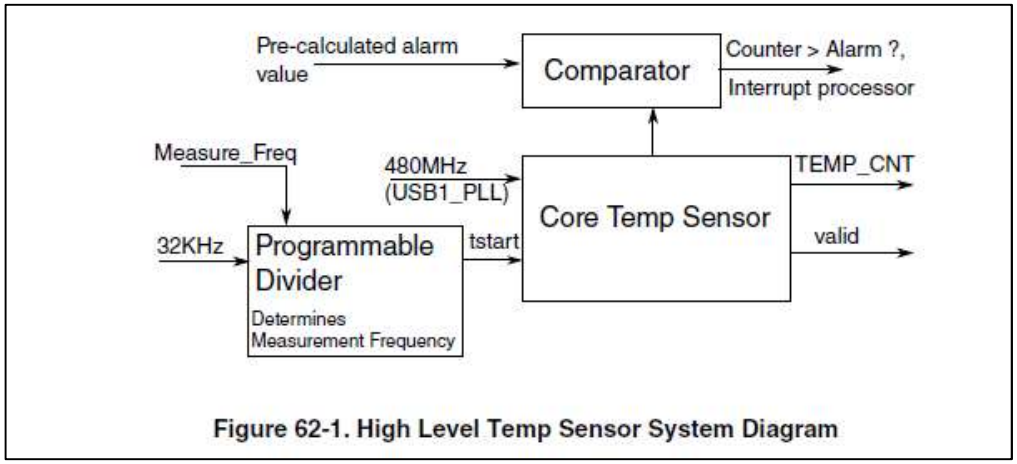
’474 Patent Claim	Representative NXP Product(s)
	35.4.1 I2C system configuration After a reset, the I2C defaults to Slave Receive operations. Thus, when not operating as a master or responding to a slave transmit address, the I2C defaults to the Slave Receive state. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1897. **NOTE** The I2C is designed to be compatible with the Philips™ I2C bus protocol. For information on system configuration, protocol, and restrictions, see the *I2C Bus Specification*, version 2.1, by Philips Semiconductors. The I2C supports Standard and Fast modes only. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1898.

’474 Patent Claim	Representative NXP Product(s)
	35.7.3 I2C Control Register (I2Cx_I2CR) The I2C_I2CR is used to enable the I2C and the I2C interrupt. It also contains bits that govern operation as a slave or a master. Address: Base address + 8h offset  i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1913. **35.7.4 I2C Status Register (I2Cx_I2SR)** The I2C_I2SR contains bits that indicate transaction direction and status. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1914.

’474 Patent Claim	Representative NXP Product(s)
	35.7.5 I2C Data I/O Register (I2Cx_I2DR) In Master Receive mode, reading the data register allows a read to occur and initiates the next byte to be received. In Slave mode, the same function is available after it is addressed. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1916.  Address: Base address + 10h offset Bit 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 Read 0 DATA Write Reset 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 **I2Cx_I2DR field descriptions** **Field** — **Description** 15–8 Reserved — This read-only field is reserved and always has the value 0. DATA — Data Byte. Holds the last data byte received or the next data byte to be transferred. Software writes the next data byte to be transmitted or reads the data byte received. **NOTE:** The core-written value in I2C_I2DR cannot be read back by the core. Only data written by the I2C bus side can be read. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1916.
[14d.] wherein the slave serial interface is responsive to a read temperature command issued by the master serial interface to return to the master serial interface a temperature value	In each of the Accused ’474 i.MX Processors, the slave serial interface is responsive to a read temperature command issued by the master serial interface to return to the master serial interface a temperature value associated with the microprocessor. For example, in each of the Accused ’474 i.MX Processors, the slave serial interface (*e.g.*, I2C identified above) is responsive to a read temperature command (*e.g.*, the read tempearture command requesting the result derived from the temperature monitor “TEMPMON” measurement) issued by the

’474 Patent Claim	Representative NXP Product(s)
associated with the microprocessor.	master serial interface (*e.g.*, the I2C interface on the master/MCU and/or the host processor identified above) to return to the master serial interface a temperature value (*e.g.*, the result derived from the temperature monitor “TEMPMON” measurement) associated with the microprocessor identified above. **35.7.5 I2C Data I/O Register (I2Cx_I2DR)** In Master Receive mode, reading the data register allows a read to occur and initiates the next byte to be received. In Slave mode, the same function is available after it is addressed. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1916. Address: Base address + 10h offset Bit 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 Read 0 DATA Write Reset 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 **I2Cx_I2DR field descriptions** **Field** — **Description** 15–8 Reserved — This read-only field is reserved and always has the value 0. DATA — Data Byte. Holds the last data byte received or the next data byte to be transferred. Software writes the next data byte to be transmitted or reads the data byte received. **NOTE:** The core-written value in I2C_I2DR cannot be read back by the core. Only data written by the I2C bus side can be read. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 1916.

’474 Patent Claim	Representative NXP Product(s)
	10.4.2.2 Thermal-aware power management The temperature sensor block (TEMPMON) implements a temperature sensor/conversion function. The block features an alarm function that can raise an interrupt signal if the temperature is above a specified threshold. Software may implement temperature aware DVFS for the ARM domain and the GPU domain, as well as temperature aware frequency scaling for other system components to ensure that both the frequency and voltage is lowered when the die temperature is above the specified limit. Software may also implement temperature aware task scheduling to ensure that non-critical tasks are suspended when the die temperature is above the specified limit. See Temperature Monitor (TEMPMON) for further details on temperature monitor functions and programmability options. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 539. **Chapter 62** **Temperature Monitor (TEMPMON)** **62.1 Overview** The temperature sensor module implements a temperature sensor/conversion function based on a temperature-dependent voltage to time conversion. The module features an alarm function that can raise an interrupt signal if the temperature is above a specified threshold. A self-repeating mode can also be programmed which executes a temperature sensing operation based on a programmed delay. Software can use this module to monitor the on-die temperature and take appropriate actions such as throttling back the core frequency when a temperature interrupt is set. The high-level implementation of the temperature sensor is shown in the figure below. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev

’474 Patent Claim	Representative NXP Product(s)
	2, 06/2014) at 5159.  **Figure 62-1. High Level Temp Sensor System Diagram** i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 5159. **62.2 Software Usage Guidelines** During normal system operation software can use the temperature sensor counter output (TEMP_CNT) in conjunction with the fused temperature calibration data to determine the on-die operational temperature or to set an over-temperature interrupt alarm to within a couple of °C. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 5160.

’474 Patent Claim	Representative NXP Product(s)
	TEMPMON_TEMPSENSE0*n* field descriptions **Field** / **Description** 31–20 ALARM_VALUE — This bit field contains the temperature count (raw sensor output) that will generate an alarm interrupt. 19–8 TEMP_CNT — This bit field contains the last measured temperature count. 7 - — This field is reserved. Reserved. 6 - — This field is reserved. Reserved. 5–3 - — This field is reserved. Reserved 2 FINISHED — Indicates that the latest temp is valid. This bit should be cleared by the sensor after the start of each measurement. 0 **INVALID** — Last measurement is not ready yet. 1 **VALID** — Last measurement is valid. 1 MEASURE_TEMP — Starts the measurement process. If the measurement frequency is zero in the TEMPSENSE1 register, this results in a single conversion. i.MX 6Dual/6Quad Applications Processor Reference Manual, Document Number: IMX6DQRM (Rev 2, 06/2014) at 5163. The Accused ’474 i.MX Processors reads and returns the “TEMPMON” value in response to a read temperature command issued by an external I2C master. *See, e.g.*:

’474 Patent Claim	Representative NXP Product(s)
	`` if (status & I2SR_SRW) { /* Master wants to read from us*/ dev_dbg(&i2c_imx->adapter.dev, "read requested"); i2c_slave_event(i2c_imx->slave, I2C_SLAVE_READ_REQUESTED, &value); /* Slave transmit */ ctl |= I2CR_MTX; imx_i2c_write_reg(ctl, i2c_imx, IMX_I2C_I2CR); /* Send data */ imx_i2c_write_reg(value, i2c_imx, IMX_I2C_I2DR); } else { /* Master wants to write to us */ dev_dbg(&i2c_imx->adapter.dev, "write requested"); i2c_slave_event(i2c_imx->slave, I2C_SLAVE_WRITE_REQUESTED, &value); ``` https://patchwork.kernel.org/project/linux-arm-kernel/patch/20191009101802.19309-1-biwen.li@nxp.com/