X4 - Datasheet

Ultra Wideband (UWB) Impulse Radar Transceiver SoC

Rev. F - Preliminary - 03-March-2020

Key Features

- Single-chip Ultra Wideband (UWB) impulse radar transceiver.
- Accurate human presence detection up to a range of 10 m.
- Sees through any material except metal, operates at sub 10 GHz.
- Low peak power consumption, typically less than 120 mW.
- Advanced power management enabling low power duty cycle controlled operation.
- Industrial operating temperature range, -40°C to +85°C.
- Low power transmitter designed to operate in world-wide markets.
- Bi-phase coding of transmitted pulses for spectrum spreading.
- Master/Slave Serial Peripheral Interface (SPI).
- Quad SPI mode for higher data rates.
- Requires few external components enabling small form-factor systems with low BOM cost.
- Small footprint Chip Scale Packaging for high density integration.

Product Description

The X4 is an Ultra Wideband (UWB) short-range impulse radar transceiver System on Chip (SoC), combining a 7.29/8.748 GHz transmitter for unlicensed operation in world-wide markets, a direct RF-sampling receiver, a fully programmable system controller and advanced power management functions in a single chip.

Applications

- Presence detection for home- and building automation.
- Presence detection for computing devices such as laptops, monitors, etc.

Ordering Information

For orders, please contact Novelda sales through the contact form on our webpage:

https://www.novelda.com

Table of Contents

1. Electrical Characteristics

1.1. Absolute Maximum Ratings

Parameter	Min	Max	Unit
Supply voltage, all domains	-0.3	3.6	V
Input voltage, digital I/O	-0.3	3.6	V
Input voltage, RF and analog I/O	TBD	TBD	V
Ambient operating temperature	-40	85	°C

Table 1.1. Absolute maximum ratings

Parameter	Standard	Max	Unit
Storage temperature	JESD22-A103C	150	°C
Reflow soldering temperature	J-STD-020	260	°C
Moisture Sensitivity Level	JESD22-A113	TBD	
ESD, CDM	JESD22-C101E	TBD	V
ESD, HBM	JS-001-2012	TBD	V

Table 1.2. Environmental sensitivity

1.2. General Operating Conditions

Parameter		Min	Typ	Max	Unit
VDD	Supply voltage, all domains[1]	1.8		3.3	V
Digital I/O characteristics @ DVDD_IO = 3.3V					
V_{IL}	Logic '0' voltage input voltage			0.8	V
V_{IH}	Logic '1' voltage input voltage	2			V
V_{OL}	Logic '0' voltage output voltage			0.4	V
V_{OH}	Logic '1' voltage output voltage	2.4			V
R_{PU}	Internal pull-up resistor		54		kohm
I_{OL}	GPIO current at V_{OL} max		7.1		mA
I_{OH}	GPIO current at V_{OH} min		13.4		mA
Digital I/O characteristics @ DVDD_IO = 2.5V					
V_{IL}	Logic '0' input voltage			0.7	V
V_{IH}	Logic '1' input voltage	1.7			V
V_{OL}	Logic '0' output voltage			0.7	V
V_{OH}	Logic '1' output voltage	1.7			V
R_{PU}	Internal pull-up resistor		73		kohm
I_{OL}	GPIO current at V_{OL} max		9.0		mA
I_{OH}	GPIO current at V_{OH} min		9.4		mA
Digital I/O characteristics @ DVDD_IO = 1.8V					
V_{IL}	Logic '0' input voltage			0.63	V
V_{IH}	Logic '1' input voltage	1.17			V
V_{OL}	Logic '0' output voltage			0.45	V
V_{OH}	Logic '1' output voltage	1.35			V
R_{PU}	Internal pull-up resistor		111		kohm

Parameter		Min	Typ	Max	Unit
I_{OL}	GPIO current at V_{OL} max		4.4		mA
I_{OH}	GPIO current at V_{OH} min		4.1		mA

[1] ±10% variation allowed beyond nominal range.

Table 1.3. General operating conditions

1.3. RX Parameters

Parameter		Min	Typ	Max	Unit
Radar frame length			1536 9.87 65.8		bins meters ns
Sampling rate			23.328		GS/s
RX gain	ETSI		14.1		dB
	KCC		12.7		dB
RX noise figure	ETSI		6.7		dB
	KCC		8.3		dB
RX bandwidth (-3dB)	F_{low}		6.5		GHz
	F_{high}		10		GHz
RX input P1dB	ETSI		-13.9		dBm
	KCC		-11.4		dBm
RX S11			< -10		dB
Differential input impedance			100		ohm

Table 1.4. RX parameters summary

1.4. TX Parameters

Parameter			Min	Typ	Max	Unit
TX center frequency, ETSI/FCC compliant ($FBDIV_{Tx}$ = 5)				7.29		GHz
TX center frequency, KCC/FCC compliant ($FBDIV_{Tx}$= 6)				8.748		GHz
TX bandwidth (-10 dB)	ETSI			1.4		GHz
	KCC			1.5		GHz
Energy per pulse	ETSI	Low		0.45		pJ
		Medium		1.47		pJ
		High		2.65		pJ
	KCC	Low		0.30		pJ
		Medium		0.93		pJ
		High		1.68		pJ
Peak pulse output power, ETSI		Low		-0.7		dBm
		Medium		4.1		dBm
		High		6.3		dBm

Parameter	Min	Typ	Max	Unit
Maximum pulse repetition frequency (f_{PLL} / 6)			40.5	MHz
S_{22}, TX OFF		< -10		dB
Differential output impedance		100		ohm

Table 1.5. TX parameters summary

1.5. Current Consumption

Parameter	Min	Typ	Max	Unit
I_{AVDD}		6.4		mA
I_{AVDD_RX}		32.5		mA
I_{AVDD_TX}		0.4		mA
I_{DVDD}		14		mA
I_{DVDD_IO}		0.2		mA
I_{DVDD_RX}		10		mA
I_{DVDD_TX}		2.5		mA
Total		**66**		**mA**

Table 1.6. Current consumption when the system is active. Numbers are given as examples only, actual active mode current will depend on configuration

Parameter		Min	Typ	Max	Unit
I_{AVDD}			0.2		mA
I_{AVDD_RX}	AVDD_RX LDO enabled		32		mA
	AVDD_RX LDO disabled		0.01		µA
I_{AVDD_TX}			0.2		mA
I_{DVDD}	Clock dividers at default settings		1.8		mA
I_{DVDD_IO}			0.15		mA
I_{DVDD_RX}			0.2		mA
I_{DVDD_TX}			0.2		mA
Total	AVDD_RX LDO enabled		**35**		**mA**
	AVDD_RX LDO disabled		**2.75**		**mA**

Table 1.7. Current consumption when the system is idle. Numbers are given as examples only, actual idle mode current will depend on configuration

Parameter	Min	Typ	Max	Unit
I_{AVDD}		4.3		µA
I_{AVDD_RX}		0.01		µA
I_{AVDD_TX}		0.01		µA
I_{DVDD}		0.01		µA
I_{DVDD_IO}		30		µA
I_{DVDD_RX}		0.01		µA
I_{DVDD_TX}		0.01		µA
Total		**34.4**		**µA**

Table 1.8. Current consumption when the system is powered down (ENABLE is low).

1.6. Specification of Clock Sources

1.6.1. Internal Low Power Oscillator (LPOSC)

Parameter		Min	Typ	Max	Unit
F_{tol}	Absolute frequency accuracy	10		30	%
F_{LPOSC}	Output frequency		27		MHz

Table 1.9. Low power oscillator (LPOSC) specification

1.6.2. Crystal Oscillator (XOSC)

Parameter		Min	Typ	Max	Unit
$F_{fundamental}$	Crystal frequency		27		MHz
F_{tol}	Frequency accuracy			TBD	ppm
C_L	Crystal load capacitance	TBD		TBD	pF
ESR_{max}	Equivalent series resistance			40	ohm
$T_{startup}$			0.5	1.0	ms

Table 1.10. Crystal oscillator specification

1.6.3. External Clock Input/Output (LVDS)

Parameter		Min	Typ	Max	Unit
LVDS mode					
F_{max}	Receive and transmit mode		750		MHz
Output differential voltage peak, VCM = 1.2V	Rload = 100Ω±1%, Full LVDS mode	250	330	400	mV
Output differential voltage peak, VCM = 0.9V	Rload = 100Ω±1%, sub-LVDS mode	125	165	200	mV
LVCMOS mode					
F_{max}	Receive and transmit mode Cload less than 5pF		125		MHz
T_{rise}/T_{fall}	Output rise fall, 20% - 80% Cload = 5pF			1.0	ns

Table 1.11. External clock input and output specification

2. Pin Assignment

2.1. Pinout

Figure 2.1 shows the footprint of the X4 chip.

Figure 2.1. Top view footprint (coordinates in micrometers)

2.2. Terminal Functions

Ball	Name	I/O Type	Function
A1	AGND_RX	Ground (analog)	Analog RX RF return path
A2	AGND_RX	Ground (analog)	Analog RX RF return path
A3	AVDD_RX	Power (analog)	1.8 - 3.3 V analog RX RF supply
A4	Reserved	-	Tie to ground or leave unconnected
A5	Reserved	-	Leave unconnected
A6	Reserved	-	Leave unconnected
A7	AVDD_TX	Power (analog)	1.8 - 3.3 V analog TX RF supply
A8	GND_TX	Ground (analog)	TX RF return path
B1	RFIN_P	RF input	Connection to differential receive antenna
B2	Not populated	-	-
B3	AGND_RX	Ground (analog)	Analog RX RF return path
B4	Not populated	-	-
B5	AGND	Ground (analog)	XOSC, PLLs and voltage reference return path

Ball	Name	I/O Type	Function
B6	GND_TX	Ground (analog)	TX RF return path
B7	GND_TX	Ground (analog)	TX RF return path
B8	RFOUT_P	RF output	Connection to differential transmit antenna
C1	RFIN_ N	RF input	Connection to differential receive antenna
C2	Not populated	-	-
C3	AGND_RX	Ground (analog)	Analog RX RF return path
C4	Not populated	-	-
C5	VSS	Ground (digital)	Digital core return path and substrate bias
C6	AVDD	Power (analog)	1.8 - 3.3 V supply for XOSC and PLLs
C7	GND_TX	Ground (analog)	TX RF return path
C8	RFOUT_N	RF output	Connection to differential transmit antenna
D1	AGND_RX	Ground (analog)	Analog RX RF return path
D2	Reserved	-	Tie to ground or leave unconnected
D3	IO2	Digital I/O	Configurable I/O
D4	Reserved	-	Tie to ground, power or leave unconnected
D5	VSS	Ground (digital)	Digital core return path and substrate bias
D6	DVDD_RX	Power (digital)	1.8 - 3.3 V digital RX timing
D7	DVDD_TX	Power (digital)	1.8 - 3.3 V digital TX timing
D8	GND_TX	Ground (analog)	TX RF return path
E1	IO1	Digital I/O	Configurable I/O
E2	IO0	Digital I/O	Configurable I/O
E3	DGND_IO	Ground (digital)	Digital I/O return path
E4	DVDD_IO	Power (digital)	1.8 - 3.3 V, digital I/O power supply
E5	ENABLE	Digital input	Asynchronous enable / active low reset
E6	DGND_RX	Ground (digital)	Digital RX timing return path
E7	DVDD	Power (digital)	1.8 - 3.3 V digital core supply
E8	XO	Analog output	27 MHz crystal connection
F1	SPICLK	Digital input	SPI clock input
F2	SPIO0	Digital I/O	Configurable I/O
F3	SPIO1	Digital I/O	Configurable I/O
F4	IO3	Digital I/O	Configurable I/O
F5	IO4	Digital I/O	Configurable I/O
F6	IO5	Digital I/O	IO5 / Differential synchronization clock input/output
F7	IO6	Digital I/O	IO6 / Differential synchronization clock input/output
F8	XI	Analog input	27 MHz crystal connection / External clock input

Table 2.1. Terminal functions overview

2.3. Configurable I/O Functions

Ball	Name	QSPI Slave		QSPI Master		SPI Slave		SPI Master		GPIO		Alt. function 1		Alt. function 2		Reset state	
		Dir	Signal	Dir	Signal	Dir	Signal	Dir	Signal	Dir	Signal	Dir	Signal	Dir	Signal	Dir	Signal
D3	IO2	-	-	I/O	nSS2	-	-	I/O	nSS2	I/O	GPIO2	I	INT0	-	-	I	GPIO2
E1	IO1	-	-	I/O	nSS1	-	-	I/O	nSS1	I/O	GPIO1	I/O	TRX_SYNC	-	-	I	GPIO1
E2	IO0	I	nSS	O	nSS0	I	nSS	O	nSS0	I/O	GPIO0	-	-	-	-	I	nSS
F1	SPICLK	I	-	O	-	I	-	O	-	-	-	-	-	-	-	I	-
F2	SPIO0	I/O	SPIO0	I/O	SPIO0	I	MOSI	O	MOSI	-	-	-	-	-	-	I	MOSI
F3	SPIO1	I/O	SPIO1	I/O	SPIO1	O	MISO	I	MISO	-	-	-	-	-	-	O	MISO
F4	IO3	I/O	SPIO2	I/O	SPIO2	I/O	-	I/O	-	I/O	GPIO3	I/O	TRX_SYNC	-	-	I	SPIO2
F5	IO4	I/O	SPIO3	I/O	SPIO3	I/O	-	I/O	-	I/O	GPIO4	-	-	-	-	I	SPIO3
F6	IO5	-	-	-	-	-	-	-	-	I/O	GPIO5	I/O	CKIO_N	I	EXTCLK	I	GPIO5
F7	IO6	-	-	-	-	-	-	-	-	I/O	GPIO6	I/O	CKIO_P	I	EXTCLK	I	GPIO6

Table 2.2. Overview of configurable I/O functions

3. Circuit Overview

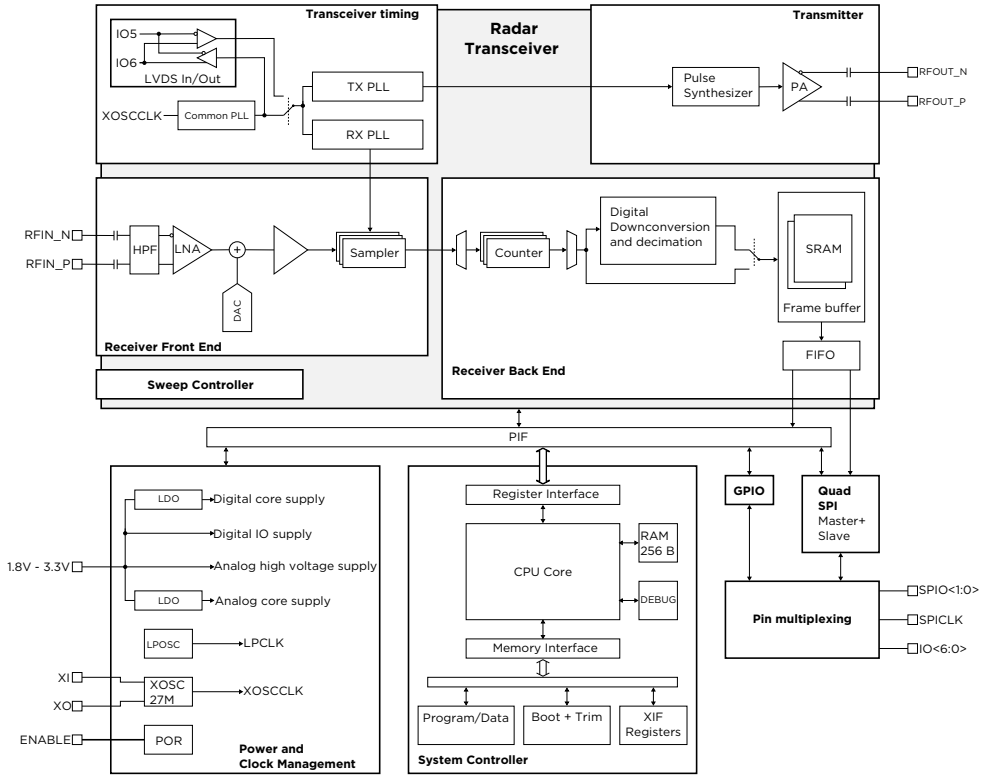


Figure 3.1. Circuit overview

X4 is a radar transceiver SoC for Ultra Wideband (UWB) impulse radar applications. The basic components are a transmitter, a receiver and related control circuits as shown in the figure above. The system is controlled by the System Controller and is configurable through a 4(6)-wire Serial Peripheral Interface.

The receive path (RX) of the X4 consists of a Low Noise Amplifier (LNA), a Digital-to-Analog Converter (DAC) for threshold setting, 1536 parallel comparators/samplers and digital integrators as well as an output memory buffer, accessible through the SPI. The RX is tightly integrated with the transmitter (TX) and is designed for coherent integration of the received energy.

The transmit path (TX) of the X4 consists of a pulse generator capable of generating pulses at a rate of up to 60.75 MHz. The output frequency and bandwidth is designed to fit worldwide regulatory requirements.

The radar transceiver is able to operate completely autonomously and can be programmed to capture data at predefined intervals and then alert or wake up a host MCU or DSP through dedicated interrupt pins. The Power Management Unit controls the on-chip voltage regulators and enables low-power applications to use efficient duty cycling by powering down parts of the circuit when they are not needed. The system can be configured to consume less than 1mW in idle mode when all analog front end components are turned off.

4. System Controller

The main component of the System Controller is the CPU core. There are two bus interfaces through which the CPU can communicate with its surroundings:

- The Processor Interface (PIF) bus connects to registers controlling core radar functionality, power and clock management functions and off-chip communications.
- The Memory Bus connects to program/data memory (SRAM), boot loader/trim data memory (ROM) and some additional control registers on the Extended Processor Interface (XIF).

4.1. Processor Interface (PIF) Registers

The PIF bus is a single cycle bus connecting the CPU to a range of 8-bit registers mapped in a 7-bit address space. On the CPU the registers on this bus are accessible both as Special Function Registers (SFR) and through the External Memory interface.

This means that the PIF registers can be accessed both directly through register access instructions and indirectly through external memory access instructions. Note that registers in the PIF address space that are native to the CPU core can only be accessed as SFR registers through direct access.

An overview of registers on the PIF bus is available in Chapter 11.

4.2. Memory Bus

The following features are connected to the memory bus:

- ROM for boot loader and trimming data.
- SRAM for program code and data storage.
- XIF and PIF registers for control of radar functionality.

The CPU communicates to the memory bus through its external memory interface. In addition, the memory bus is accessible through the SPI.

4.2.1. Memory Control Logic

The memory control logic controls the access to the on-chip memories. Both the ROM and SRAM is accessible from the SPI:

- When the memory program mode is enabled the memory control logic holds the CPU in reset until the program mode bit is cleared and takes control of the memory bus.
- When the memory SPI readback mode is enabled the memory control logic holds the CPU by forcing the memack low. The memory bus can then be accessed from the SPI.
- When neither program or readback mode is enabled, the control logic passes through the memory bus signals from the CPU.

Program and readback mode is controlled by the mem_programming_mode and mem_readback_mode SPI registers, respectively.

4.2.2. Reset Vector ROM

The boot_from_otp register controls the reset vector of the CPU. When it is set, the lower three addresses of the bus are redirected to a ROM with a jump instruction to the bootloader ROM. If the boot_from_otp register is not set, the reset vector is read from the SRAM.

After a power-on-reset, the ROM access logic will check the programming status of the bootloader ROM. If it is unprogrammed, the reset vector will point to 0, making the CPU loop at that address as long as boot_from_otp is set.

The boot_from_otp register is available both as a PIF register and as a SPI register.

4.2.3. PIF Bus Mux

PIF registers can be accessed on the memory bus at address range 0x8000-0x807F.

4.2.4. Extended Interface (XIF) Registers

XIF registers are control registers that are only available on the memory interface at address range 0x8080-0x80FF.

An overview of XIF registers is available in Chapter 12.

5. Serial Peripheral Interface (SPI)

The SPI enables communication between the X4 and an external host or slave device for transfer of radar and configuration data. It features:

- Two different modes of operation:
  - **SPI**: Single bit SPI using 4 wires and transfers data in full duplex with input and output from the device on dedicated pins.
  - **QSPI**: Quad SPI using 6 wires and transfers data in half duplex with input and output on bidirectional pins.
- Direct access to radar backend allowing external host to access sampled data while System Controller is idle.
- Selectable slave and master mode operation.
- Up to 54 MHz operation.

The system starts in single bit SPI slave mode by default and returns to this mode after a hardware reset or power toggle. This means that the SPI can only be set in master mode by the embedded System Controller. Single bit SPI/QSPI operating mode can be programmed through a PIF register from the System Controller, or additionally through an SPI register in slave mode.

5.1. Selecting QSPI Mode

The spi_mode_pif bit in the spi_config_pif PIF register selects whether QSPI or SPI mode is enabled.

In slave mode, the same bit can be accessed through the spi_mode bit in the spi_config SPI register which can be written with a write-only instruction over the SPI interface. This enables external SPI hosts to change SPI mode on the fly. Changes to this register takes effect after nSS has been toggled high and then low.

5.2. SPI Master Mode

In master mode the SPI module owns the SPI bus and defines communication protocol. All communication is initiated by the System Controller. There are two types of transfers that can be performed master mode:

- Mailbox transfers. The mailbox consists of two 8-byte deep FIFOs, one for outgoing transfers and one for incoming transfers. It can be used to transfer any kind of data from the System Controller to the external host. In QSPI master mode, X4 only supports writes. No read data will be returned to the SPI Master.
- Radar data transfers. The SPI master transfers the sampled data directly from the radar backend.

Radar data and mailbox data can be freely mixed. The SPI Master follows the following rules:

- When the SPI Master is idle, mailbox data is prioritized of radar data.
- When a radar data transfer has been started and there is no mailbox data, the SPI Master sends the full radar data burst, ignoring any mailbox activity until the burst is complete.

This means it is easy to mix different scenarios. For example, the System Controller can set up a transaction in the mailbox, then trigger a radar burst. Because the mailbox takes priority, the SPI Master will send the content of the mailbox first, then start the radar burst.

Note that the SPI Master does not distinguish between a pure mailbox access and a mixed access. It cannot know if mailbox activity will be followed by radar data or not. Therefore, all mailbox activity will (in single-bit SPI) be considered possible reads, and the incoming mailbox FIFO will be filled accordingly.

It is possible to mix single-bit and QSPI behavior in a single transaction (without toggling NSS). Typical use would be flash components that require single-bit SPI setup before accepting QSPI data in the same transaction.

5.3. SPI Slave mode

All communication between the SPI master (the external host) and the SPI slave take place as units called commands. A command starts with an 8-bit instruction and may be followed by a payload.

During single bit SPI commands data is transferred from the master to the slave on the SPIO0/MOSI pin and from the slave to the master on the SPIO1/MISO pin.

QSPI commands use four bidirectional pins for half-duplex data transfer between master and slave. Instructions and data are transferred as nibbles (four bit at a time) in both directions.

The SPI slave starts driving the IO signals on the falling edge of the end of the last instruction cycle in QSPI read commands. The host is expected to stop driving the I/O signals at this point.

The SPI can be driven by an SPI master in either of the two following clocking modes:

- **Mode 0** with Clock Polarity (CPOL) = 0 and, Clock Phase (CPHA) = 0.
- **Mode 3** with CPOL = 1 and, CPHA = 1.

In both of these modes the SPI slave samples the input on the rising edge of SPICLK and changes the output following a falling edge. The difference is in the polarity of the clock between transfers:

- SPICLK will stay at logic low state with CPOL = 0, CPHA = 0.
- SPICLK will stay at logic high state with CPOL = 1, CPHA = 1.

All transfers begin with the most significant bit and ends with the least significant bit. In Quad SPI transfers the following applies:

- The most significant nibble is transferred on the first clock cycle followed by the least significant nibble on the second clock cycle.
- The least significant bit of each nibble is transferred on SPIO0 and the most significant bit of each nibble is transferred on SPIO3.

When nSS is inactive (high) the SPIO1/MISO pin must be in high impedance mode to allow multiple slave to operate against a single master. The SPI slave module therefore disables the output enable signal on the SPIO1/MISO pin when nSS is high.

5.3.1. SPI Commands

The commands are structured as follows:

- A command begins when nSS is pulled low by the master.
- All commands start with an eight bit instruction transferred from the master to the slave.
- The length of all transfers following the instruction are determined by the instruction.
- The commands end when nSS is pulled high by the master.

The instructions are structured as follows:

- The most significant bit is always high for write instructions and low for read instructions.
- The 7 least significant bits contain an address to an SPI register.
- The addressable locations (the SPI registers) can be write-only, read-only or both read and writeable.

A complete list over all the SPI registers can be found in Chapter 13. Note that the SPI registers are only used in slave mode.

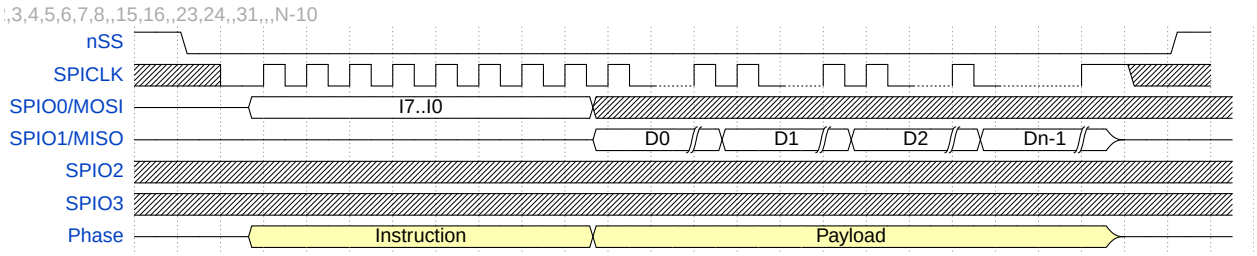


Figure 5.1. Single bit wide SPI read sequence example timing diagram.

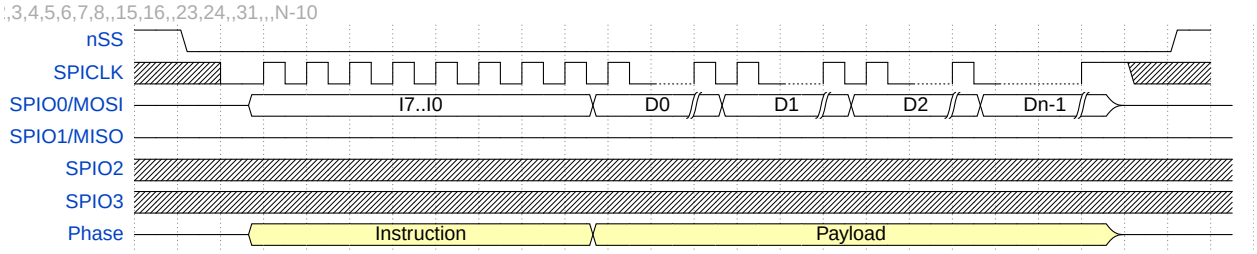


Figure 5.2. Single bit wide SPI write sequence example timing diagram.

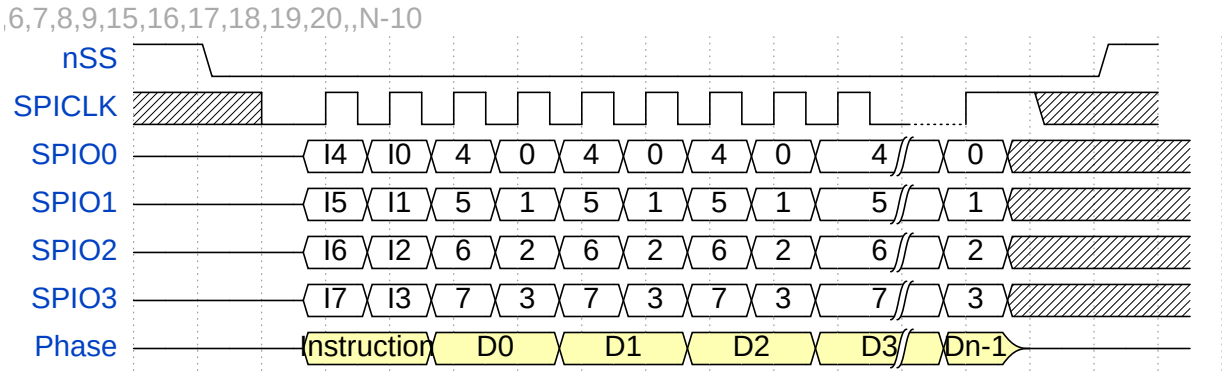


Figure 5.3. QSPI read/write sequence example timing diagram.

6. Input/Output Pins

There are 7 pins that can be configured as General Purpose I/O pins, each providing individually configurable input/output direction. In addition IO0-4 feature configurable pull-up resistor while IO5 and IO6 feature programmable Schmitt trigger and input buffer.

IO5 and IO6 can also be configured to be used as an LVDS input or output clock buffer.

SPIO0, SPIO1 and SPICLK are dedicated to the SPI/QSPI.

See Section 2.3 for a complete overview of IO functions available at each pin. To use a pin as GPIO, set the corresponding bit in the io_gpio_sel segment of the io_ctrl_3 register.

6.1. Block Diagram

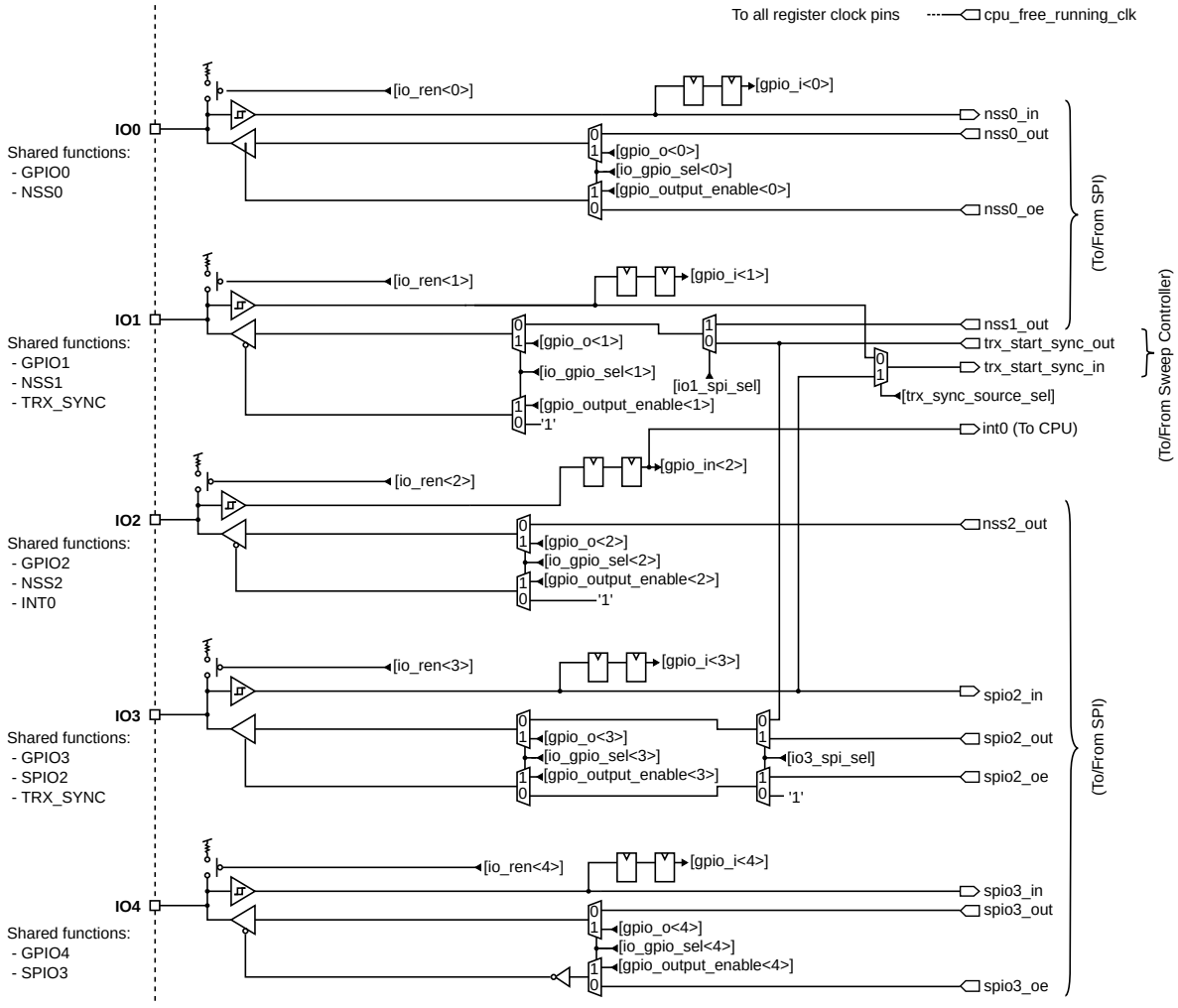


Figure 6.1. Overview of multifunctional digital IO pins.

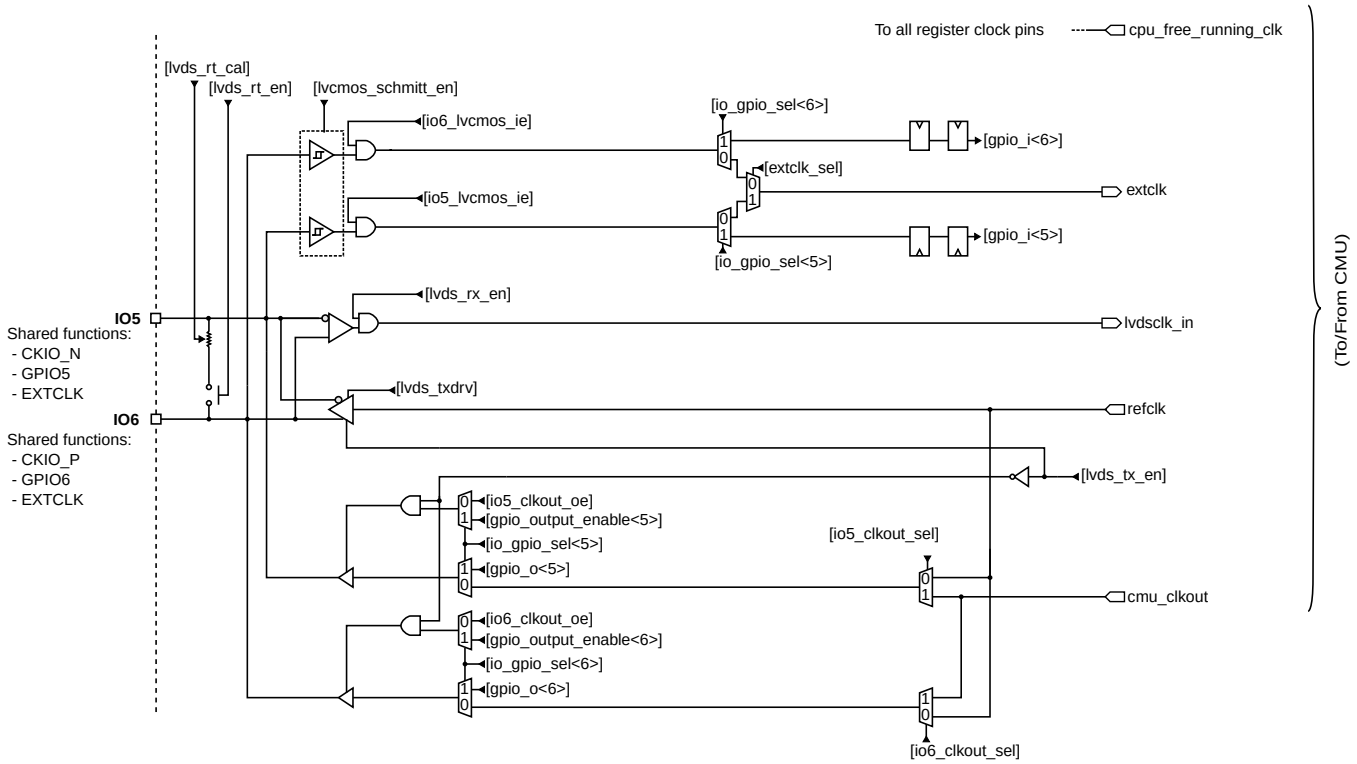


Figure 6.2. Overview of multifunctional IO pins with LVDS option.

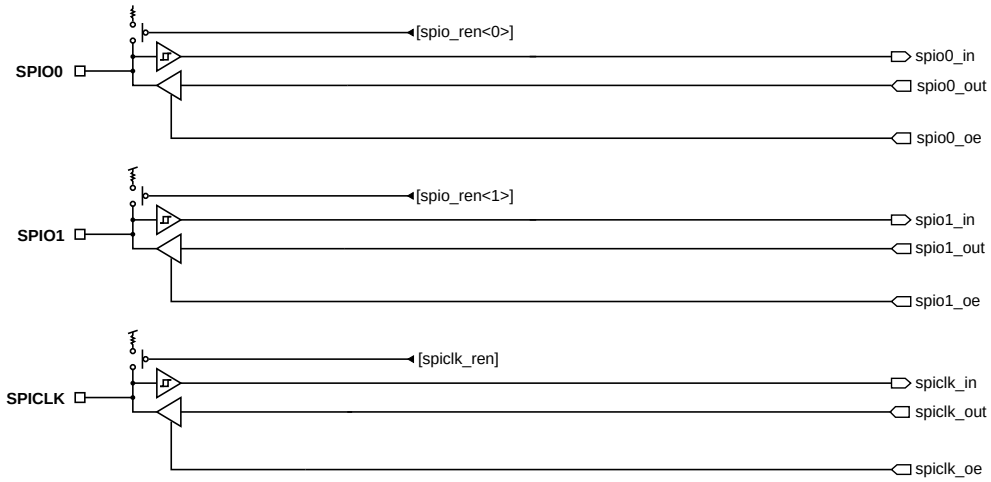


Figure 6.3. Overview of pins dedicated to SPI.

7. Transceiver

The radar transceiver consists of a transmitter, a receiver and timing circuits for synchronized sampling and pulse transmission.

7.1. Transmitter

The transmitter features programmable output power and is capable of generating bi-phase pulses (180 degrees phase shift) with high accuracy for spectrum smoothing. Figure 7.1 shows a block diagram of the transmitter.

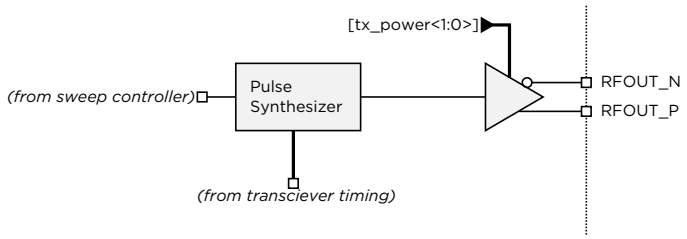


Figure 7.1. Transmitter block diagram.

7.2. Receiver

The receiver front end features an LNA, a DAC, comparator and 12 physical samplers that are re-used for a total frame size of up to 1536 range bins. The analog to digital conversion is based on the Swept Threshold principle. There are two parallel frame buffers available which allows one frame to be read out while the next is being sampled.

An optional downconversion, filtering and decimation feature is available for converting the sampled RF data to complex baseband data, reducing the amount of data required for each radar frame.

7.3. Timing

The receiver and transmitter are clocked by two separate PLLs, the RX PLL and the TX PLL. Figure 7.2 shows a block diagram of the transceiver timing circuits.

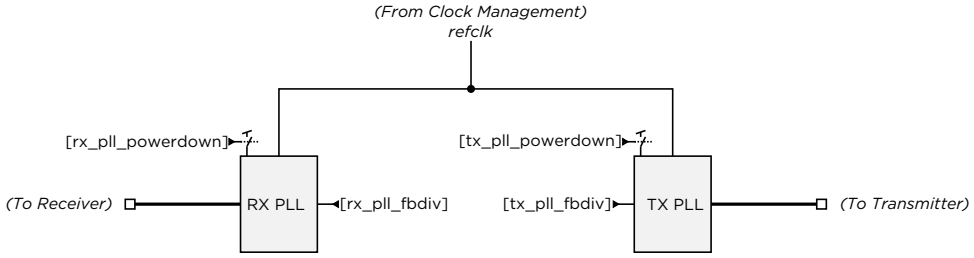


Figure 7.2. Transceiver timing block diagram.

7.3.1. Receiver Timing

In the receiver, the rx_phase signal is used to clock the latches in the samplers. The effective sampling rate F_S is given by the RX PLL frequency:

$$F_S = F_{RX\_PLL} \cdot 12 \tag{7.1}$$

The RX PLL frequency is a product of the feedback divider settings of the RX PLL and the Common PLL clock frequency F_{refclk}:

$$F_{RX\_PLL} = F_{refclk} \cdot \mathrm{N}_{\mathrm{fbdiv\_rx}} \tag{7.2}$$

The RX PLL feedback divider value N_{fbdiv_rx} is configured through the rx_pll_ctrl_1 register.

Using the register default values the RX PLL frequency is 27 x 8 x 9 = 1944 MHz and the sampling rate is 1944 x 12 = 23.328 GS/s.

Important

The receiver is designed to operate with a maximum clock frequency of 1944 MHz. It is not recommended to use the receiver with non-default feedback divider settings.

7.3.2. Transmitter Timing

In the transmitter, the tx_phase signal is used to control the internal timing and set the center frequency of the transmitter pulse. The center frequency of the pulse F_{TX} is:

$$F_{TX} = F_{TX\_PLL} \cdot 6 \tag{7.3}$$

The TX PLL frequency is a product of the feedback divider setting of the TX PLL N_{fbdiv_tx} and the Common PLL clock frequency F_{refclk}:

$$F_{TX\_PLL} = F_{refclk} \cdot \mathrm{N}_{\mathrm{fbdiv\_tx}} \tag{7.4}$$

The TX PLL feedback divider value N_{fbdiv_tx} is configured through the tx_pll_ctrl_1 register. Table 7.1 lists the recommended default values for the transmitter.

Transmitter frequency	N_{fbdiv_tx}	tx_pll_fbdiv register setting
7.29 GHz	5	3
8.748 GHz	6	4

Table 7.1. Transmitter configurations

7.4. Configuring the Transceiver

The transceiver is configured through the PIF registers. The most important configuration registers that affect the transceiver are described in this section. See Chapter 11 for a complete overview.

7.4.1. Setting PRF

The Pulse Repetition Frequency (PRF) is derived from the refclk clock signal and is set by the trx_clocks_per_pulse (CPP) register:

$$F_{PRF} = \frac{F_{refclk}}{\mathrm{CPP}} \tag{7.5}$$

7.4.2. Setting DAC Sweep Parameters

The DAC sweep range must be set large enough to cover the input signal of interest in a given application. Narrower DAC sweep boundaries will decrease the sweep time or allow more integration.

The upper DAC sweep boundary is configured by the trx_dac_max_h and trx_dac_max_l registers, while the lower DAC sweep boundary is configured by the trx_dac_min_h and trx_dac_min_l registers.

The Sweep Controller can be configured to perform sweeps in six different modes. Figure 7.3 illustrates the time-voltage relationship of the output of the DAC for each of the modes (numbered A through F in the figure).

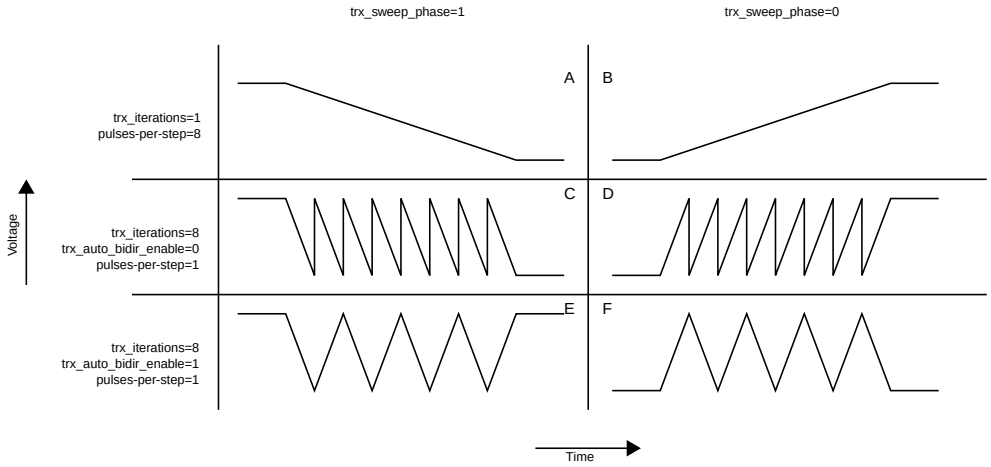


Figure 7.3. Time-voltage relationship of the sweep modes, together with their corresponding register settings.

There are several settings that control the behavior of the sweep. Plots A and B in Figure 7.3 illustrates the DAC output voltage for a sweep with trx_iterations = 1. The sweep sequence can be split over several iterations without increasing the total level of integration. This spreads out the samples taken for a given DAC value. The resulting DAC output voltage is illustrated in plots C and D. Here, trx_iterations has been increased to 8. To maintain the same total level of integration, the number of pulses per step (PPS) has to be decreased accordingly. This is configured through trx_pulses_per_step_msb and trx_pulses_per_step_lsb.

The sweep can be performed in either bi-directional mode or non bi-directional mode, controlled by trx_auto_bidir_enable. Finally, the initial value of the DAC can be set with trx_sweep_phase.

Note that the DAC step size and DAC settle time can be configured by the trx_dac_step register. However it is highly recommended to leave these at their respective default values.

7.4.3. Processing Gain and Sweep Time

X4 uses coherent integration to achieve processing gain and the level of processing gain increase with higher integration. Increased integration can be achieved by increasing the number of pulses per step or by increasing the number of iterations. The total integration is the product of these two values. The amount of processing gain is doubled with twice the integration, resulting in a Signal-to-Noise Ratio (SNR) enhancement of 3 dB. The Pulse Repetition Frequency (PRF), the DAC sweep range and the total amount of integration determine the frame rate. Depending on the requirements of a given application the radar can be configured with more processing gain at the cost of lower frame rate, or higher frame rate at the cost of a lower SNR. The sweep time with default DAC setup and hold times is

$$T_{frame} = \frac{\text{rx\_mframes} \cdot 3 + 1}{F_{trx\_backend\_clk}} + \frac{(\text{PPS} \cdot \text{CPP} + 1) \cdot (\text{DAC}_{\max} - \text{DAC}_{\min} + 1) \cdot \text{trx\_iterations}}{F_{refclk}} \quad (7.6)$$

7.5. Starting a Sweep

A sweep sequence is started by the trx_start action register. The trx_ctrl_done and trx_backend_done bits are cleared when a new sweep is started. Once the Sweep Controller has completed the threshold sweep, trx_ctrl_done is set. The receiver backend now transfers the sampled data to one of the two SRAM blocks, selectable with the ram_select register. Once this process has completed, trx_backend_done is set and the sampled radar frame is ready to be read.

7.6. Master/Slave Mode

Transceiver slave mode is enabled by the trx_ctrl_slave_mode bit. When this bit is set, the trx_start signal from the System Controller is ignored, and in its stead the trx_start_sync_in is used to trigger the start of a frame capture. trx_start_sync_in can be sourced from either IO1 or IO3, selectable by the trx_sync_source_sel bit in the io_ctrl_5 register.

The master generates trx_start_sync_out from its internally synchronized trx_start signal. A negative edge clocked flip flop is inserted for hold time fixing when trx_start_sync_negedge is set. trx_start_sync_out can be output on IO1 and/or IO3 by clearing the io1_spi_sel and/or io3_spi_sel bits in the io_ctrl_5 register as long as the corresponding bit in the io_gpio_sel segment of the io_ctrl_3 register is cleared.

The master also shares its internal refclk signal with the slave through the CKIO LVDS interface. On the slave, the trx_start_sync_in signal is synchronized to its local version of the refclk signal.

8. Clock Management Unit

The Clock Management Unit (CMU) supplies the radar transceiver core as well as the system controller and peripheral units with clock signals.

See Section 1.6 for electrical specifications of the different clock sources.

8.1. Block diagram

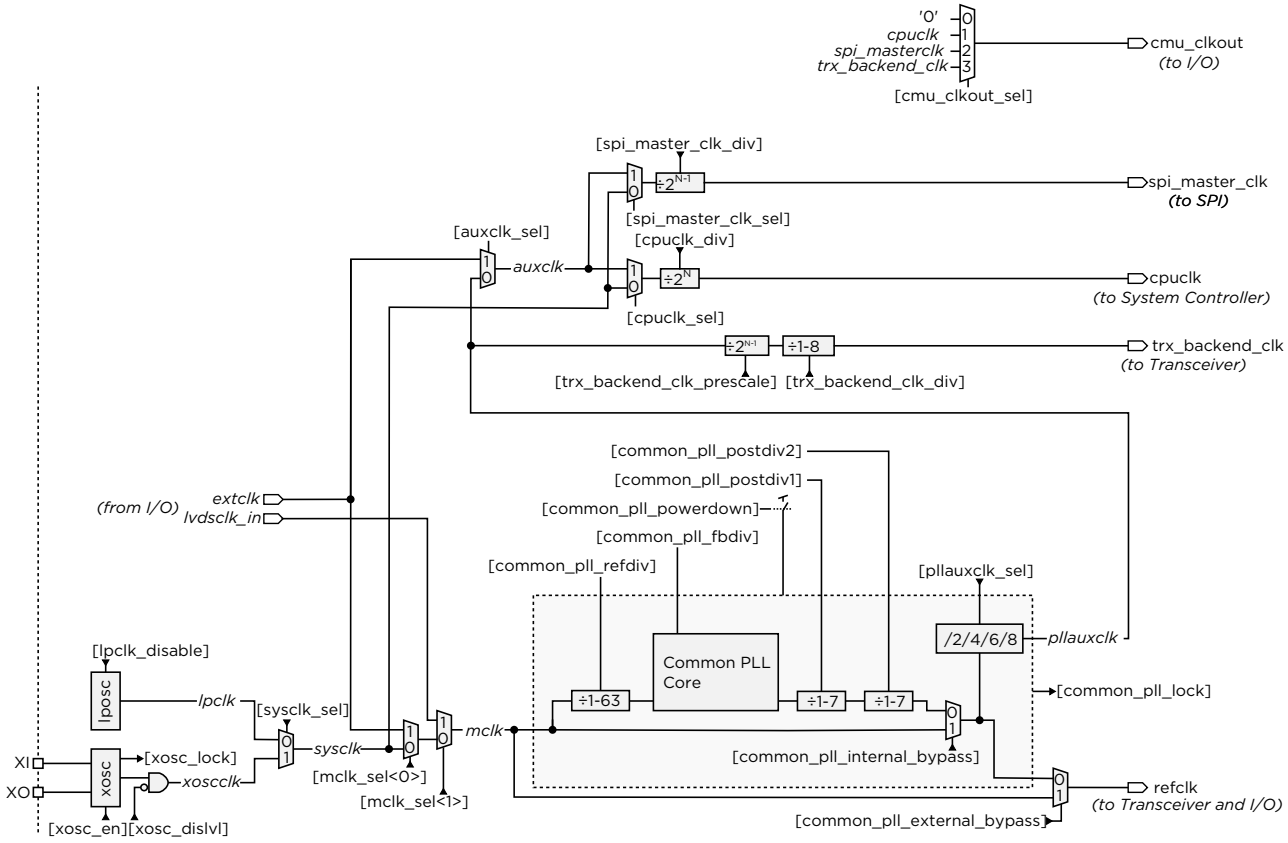


Figure 8.1. Detailed block diagram of the Clock Management Unit.

8.2. Low Power Oscillator (LPOSC)

The low power oscillator (LPOSC) is designed primarily as a source of the system controller during start up.

Because of the reduced frequency accuracy and phase noise performance in the LPOSC compared with the XOSC, it is not recommended to clock the transceiver from the LPOSC as this will directly impact both the frequency accuracy of the transmitter and the performance of the radar system as a whole.

8.3. Internal Crystal Oscillator (XOSC)

The crystal oscillator (XOSC) is designed for providing the radar transceiver core with a stable and accurate reference clock signal.

Figure 8.2 shows a typical block schematic when the X4 is clocked using the XOSC.

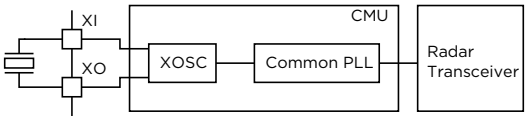


Figure 8.2. Crystal oscillator block diagram

8.3.1. XOSC bypass

In XOSC bypass mode the radar core is clocked by an external low frequency clock source. This mode can be useful if there is already a clock signal available on the application PCB. The external clock input pin supports single-ended inputs.

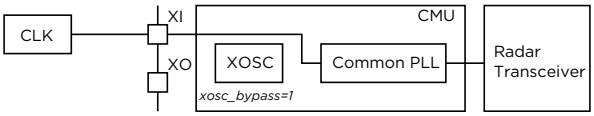


Figure 8.3. Single-ended clock input block diagram

8.4. Common PLL

The Common PLL generates the refclk signal, which is the common clock signal for the RX and TX PLLs. In normal operation, the Common PLL should be configured to output a 243 MHz clock signal. There are three recommended operating modes for the Common PLL, listed in Table 8.1.

common_pll_fbdiv	common_pll_postdiv1	Mode	Comment
18	2	Lower power	Default
36	4	Lower phase noise	
54	6	Lower phase noise	Requires AVDD=2.5V or higher

Table 8.1. Recommended Common PLL configurations. Registers not listed should be left at default values.

Note that in Common PLL bypass the PLL must be configured differently, see Section 8.4.1.

8.4.1. Common PLL bypass

In Common PLL bypass mode the radar transceiver core is clocked by an external high frequency clock from the LVDS interface on pins IO5 and IO6 and the Common PLL is bypassed. This allows multiple radars to stay synchronized which is a requirement for digital beamforming applications. Figure 8.4 shows a block diagram of the Common PLL in bypass mode.

The Common PLL can be configured in internal or external bypass mode:

- In internal bypass mode (common_pll_internal_bypass=1) the pllauxclk signal is still generated which is required for the receiver to function. This mode must therefore be used if the receiver is to be used.
- In external bypass mode (common_pll_external_bypass=1) the entire Common PLL is bypassed, and can be powered down to save energy. However, since the pllauxclk signal is no longer generated, only the transmitter can be used in this mode.

See Section 7.6 for more details on Transceiver master/slave mode.

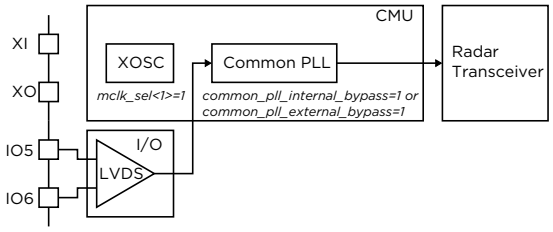


Figure 8.4. Common PLL bypass mode

9. Power Management

9.1. Power Domains

X4 supports a wide supply range from 1.8 to 3.3V.

All core voltage domains are supplied by internal voltage regulators, and each regulator connects to a dedicated set of power pins. The different power domains are described below.

- DVDD_IO powers the digital input and output drivers
- DVDD_TX and DVDD_RX powers the clock network and digital part of the transmitter and receiver
- DVDD powers the digital core
- AVDD powers the on-chip oscillators and PLLs
- AVDD_TX and AVDD_RX powers the pure analog parts of the transmitter and receiver

On the PCB all power domains can be connected to a single available power source such as a switch-mode DC-DC converter. Other than decoupling capacitors no external components are required for the power network.

Because a majority of the current is drawn by the core domains it may be beneficial to supply these with a different voltage than the IO domain if a larger IO voltage is desired. Each of the power domains can therefore optionally be supplied with different voltages.

9.2. Overview of Power Modes

The X4 embeds a set of voltage regulators that are managed by the Power Management Unit (PMU) and powers the core of the chip. These internal regulators are designed to supply the internal power domains of the X4. The X4 feature different operating modes, as shown in the table below.

Power mode	Wake-up source	Description
Power down	ENABLE pin	Controlled by external Enable pin (system cannot enter power down mode on its own). All internal LDOs are in power-down mode. Wake-up causes a full system restart, all registers and memories are reset to their default (startup) values.
Stop	TRX done GPIO pin	When the system controller is in stop mode both the system controller core functionality and peripherals are stopped. Otherwise, this mode is similar to Idle mode.
Idle	16-bit timer TRX done GPIO pin	Register values are retained, system can resume operation from where it left off. When the system controller is in idle mode the system controller core functionality is stopped while some peripherals are active. All parts of the radar transceiver and PLLs can be individually powered down and disabled if necessary.
Active	N/A	All required modules are active. If the transceiver is used in transmitter or receiver only mode, the unused parts can be powered down.

Table 9.1. Overview of power modes

Different peripheral modules can be powered down when the system is in idle or stop mode. The current consumption and wake-up time will depend on the status of the individual modules.

Typical wake up times are listed below:

- XOSC enable: < 1ms
- LDO enable: < 20µs
- Common PLL enable: < 50µs
- RX/TX PLL enable: < 10µs

9.2.1. Clock Management Power Down

All parts of the CMU can be individually powered down when they are not required by the application.

The LPOSC can be powered down by setting the lpclk_disable bit in the osc_ctrl register when the system is clocked from either the XOSC or an external clock source.

The XOSC can be powered down by clearing the xosc_en bit in the osc_ctrl register. In idle mode this can be used when the system is clocked from the lower power LPCLK while the transceiver is not in use. In active mode this can be used to reduce power consumption of the system when the Common PLL is in bypass mode, or when the system is clocked from an external source through IO5 and/or IO6.

The Common PLL can be powered down by setting the common_pll_powerdown bit in the common_pll_ctrl_1 register. This can be done when the radar transceiver is not in use, or when the Common PLL is in external bypass mode (see Section 8.4.1).

9.2.2. Receiver Power Down

If the receiver part of the radar transceiver is not needed, it can be partially or completely powered down to save energy.

The RX PLL can be powered down by setting the rx_pll_powerdown bit in the rx_pll_ctrl_2 register. This will eliminate all dynamic power consumption on the DVDD_RX power domain.

The powerdown_sampler bit in the smpl_mode register can also be used to reduce the dynamic power consumption on the DVDD_RX domain without requiring the RX PLL to be completely disabled.

In order to eliminate the leakage induced static power consumption, the DVDD_RX LDO can be disabled by setting the dvdd_rx_disable bit in the dvdd_rx_ctrl register.

The analog receiver front end can be powered down by disabling the AVDD_RX LDO by setting avdd_rx_disable bit in the avdd_rx_ctrl.

9.2.3. Transmitter Power Down

If the transmitter part of the radar transceiver is not needed, it can be partially or completely powered down to save energy.

The TX PLL can be powered down by setting the tx_pll_powerdown bit in the tx_pll_ctrl_2 register. This will eliminate all dynamic power consumption on the DVDD_TX power domain. In order to eliminate the leakage induced static power consumption as well, the DVDD_TX LDO can be disabled by setting the dvdd_tx_disable bit in the dvdd_tx_ctrl register.

The analog transmitter front end can be powered down by disabling the AVDD_TX LDO which is done by setting avdd_tx_disable bit in the avdd_tx_ctrl.

9.3. Recommended Power Up Sequence

The recommended power up sequence is described below. Note that this is the default sequence for starting up the entire system under normal conditions. Deviations to this sequence may be necessary depending on the requirements of the application. The System Controller controls the registers used to power up the system and must be enabled and executing an appropriate firmware. This firmware is supplied by Novelda.

The following assumes that all power pins are supplied by an appropriate voltage. Note that if DVDD_IO is supplied from a different source than the core domain pins, the DVDD_IO supply should be enabled before any of the core domains.

1. Enable the AVDD_RX, DVDD_RX, AVDD_TX and DVDD_TX LDOs.
   - Each LDO is enabled by clearing the corresponding disable bit in the avdd_tx_ctrl, dvdd_tx_ctrl, avdd_rx_ctrl and dvdd_rx_ctrl registers.
   - After enabling each LDO, ensure that the LDO was powered up successfully by checking the corresponding *_power_good status bit in the ldo_status_2 register.
2. Enable the XOSC by setting the xosc_en bit and clearing the xosc_dislvl bit in the osc_ctrl register. Before continuing, ensure that the XOSC has started successfully by checking the xosc_lock bit in the lock_status register.
3. Enable the XOSC as source for sysclk by setting the sysclk_sel bit in the osc_ctrl register.
4. Enable Common PLL by clearing the common_pll_powerdown bit in the common_pll_ctrl_1 register. Before continuing, ensure that the PLL has locked by checking the common_pll_lock bit in the lock_status register.
5. Enable the RX and TX PLLs by clearing the tx_pll_powerdown and rx_pll_powerdown bits in the tx_pll_ctrl_2 and rx_pll_ctrl_2 registers, respectively. Before continuing, ensure that the PLLs have locked by checking the tx_pll_lock and rx_pll_lock bits in the tx_pll_status and rx_pll_status registers, respectively.
6. Enable clocks to the receiver digital backend by setting the trx_backend_clk_prescale bits in the mclk_trx_backend_clk_ctrl register to 1 or higher.

10. Implementation and Layout

The X4 requires a minimum of external components enabling small form-factor implementations with low BOM cost. It can operate with a single crystal as clock reference and using a single power supply. The X4 communicates with a host MCU or DSP through a serial interface, either SPI or QSPI. More advanced applications requiring digital beamforming can be acheived by using multiple X4 devices.

10.1. Typical Application Circuit

10.1.1. Single Radar

The following figure shows the X4 in a typical application circuit with a single radar.

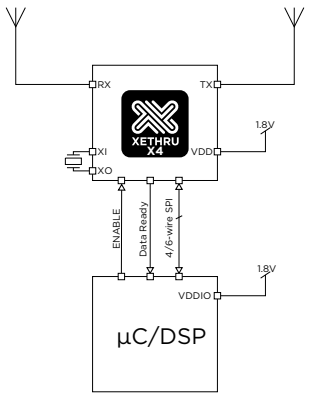


Figure 10.1. Typical application circuit diagram

10.1.2. Multi Radar Applications

Multiple instances of the X4 radar IC is required in applications applying digital beamforming techniques to perform 2D or 3D imaging. In a typical setup, the ICs use one or more transmitter and two or more receivers connected to antennas spaced by approximately half the wavelength of the transmitted signal (about 2 cm for a 7.29 GHz signal).

The ICs must be carefully synchronized to maintain coherency. This is achieved by sharing the common PLL between the systems; they are configured in a master/slave style setup where one chip acts a master providing the slaves with a clock signal through the CKIO LVDS interface on IO5 and IO6 and the TRX_SYNC signal through either IO1 or IO3.

The following figure shows a setup with three radar ICs sharing a single TX. The three radars are connected to the same SPI bus; the host must select between the slaves by independently selecting them with the slave select signals (not shown in the figure).

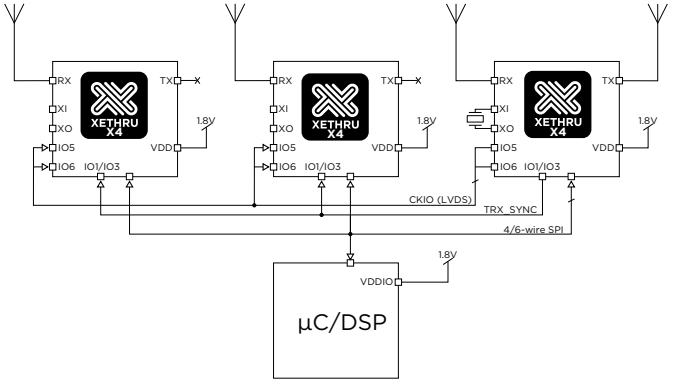


Figure 10.2. Typical application circuit of a setup with multiple radars for digital beamforming

10.2. Reference Schematic and PCB Layout

10.2.1. Recommended Chip Connections

Figure 10.3 shows the reference circuit with required external components for the X4 IC. For the simplest configuration, all the supply pins must be connected to the same power supply net and decoupled with a five different decoupling capacitors in parallel, with the values of 4.7µF, 100nF, 10nF, 1nF and 100pF. If VDD_IO is supplied from a separate voltage source, it must be decoupled with a 100 nF capacitor. The capacitors must be placed as close as possible to the pins that they decouple, also keeping in mind the ground return path from the capacitor to the X4.

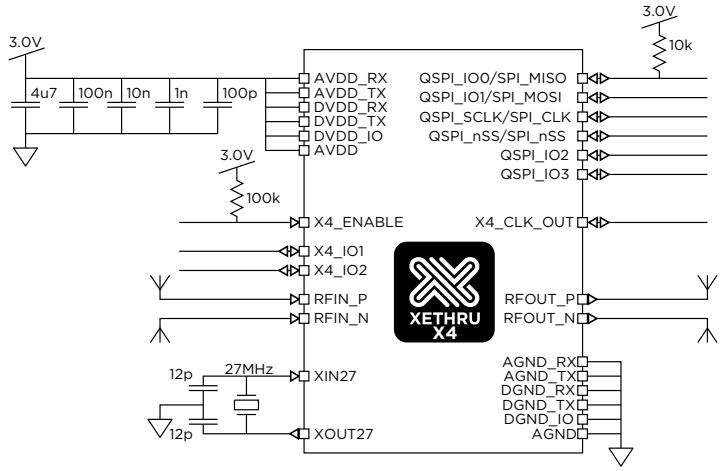


Figure 10.3. Recommended Chip Operation

To drive the internal crystal oscillator a 27 MHz crystal is required. The oscillator has internal load capacitance of TBD pF so external capacitors are not required provided a crystal with the appropriate C_L requirement is used. It is possible to disable the internal load capacitance through register TBD if a crystal with other C_L value is used. In the latter case, the crystal must be properly loaded with external capacitors.

Novelda also recommend connecting a 100 kOhm pull-up resistor on the ENABLE pin, and a 10 kOhm pull-up resistor on the QSPI_IO0/SPI_MISO pin.

X4 features one separate ground pin for each power domain. All ground pins must be firmly connected to a single common ground plane.

Novelda reference layout must be followed closely for proper performance.

10.2.2. PCB Layout Recommendations

Figure 10.4 shows one example for the layout of the X4. The most critical part is the RF lines. The figure shows the TX and RX lines routed on inner layer 1, with ground planes on both sides of the traces, and with the bottom ground plane on inner layer 2. This is not required, but the RF lines must be designed to be 100 Ohm differential.

In general, Novelda recommends

- Proper grounding and short traces (if possible) on all external componens to reduce parsitic capacitance.
- Proper bypass capacitors on all power pins.
- Avoid routing any analog or digital signals close to the RF lines.

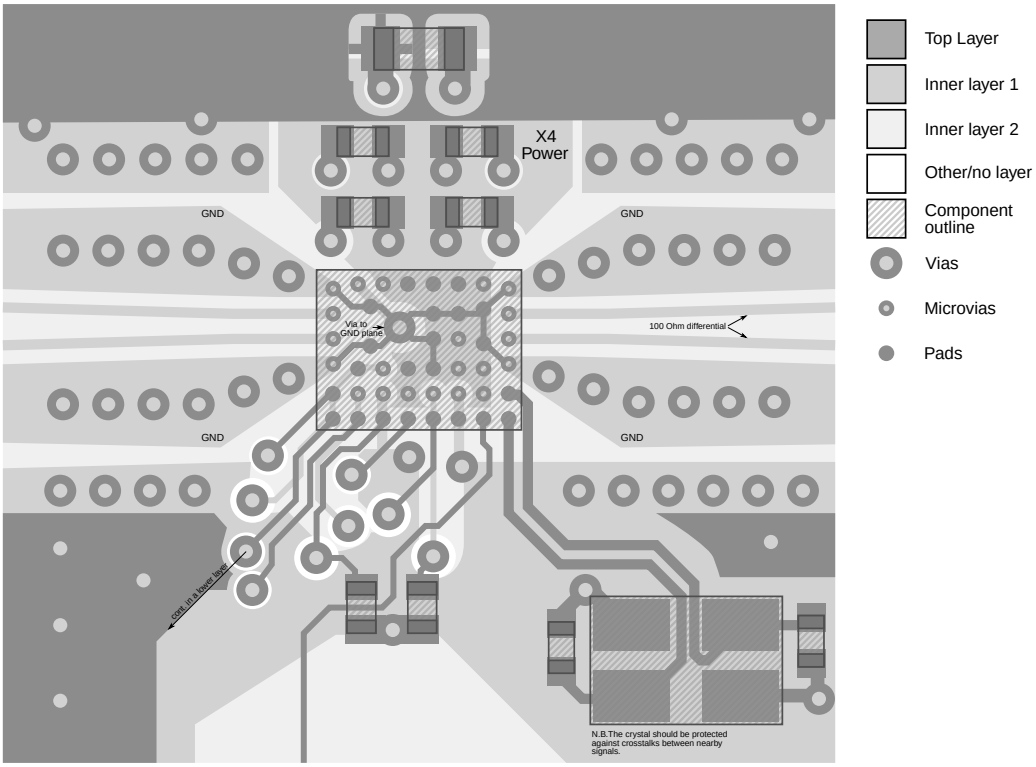
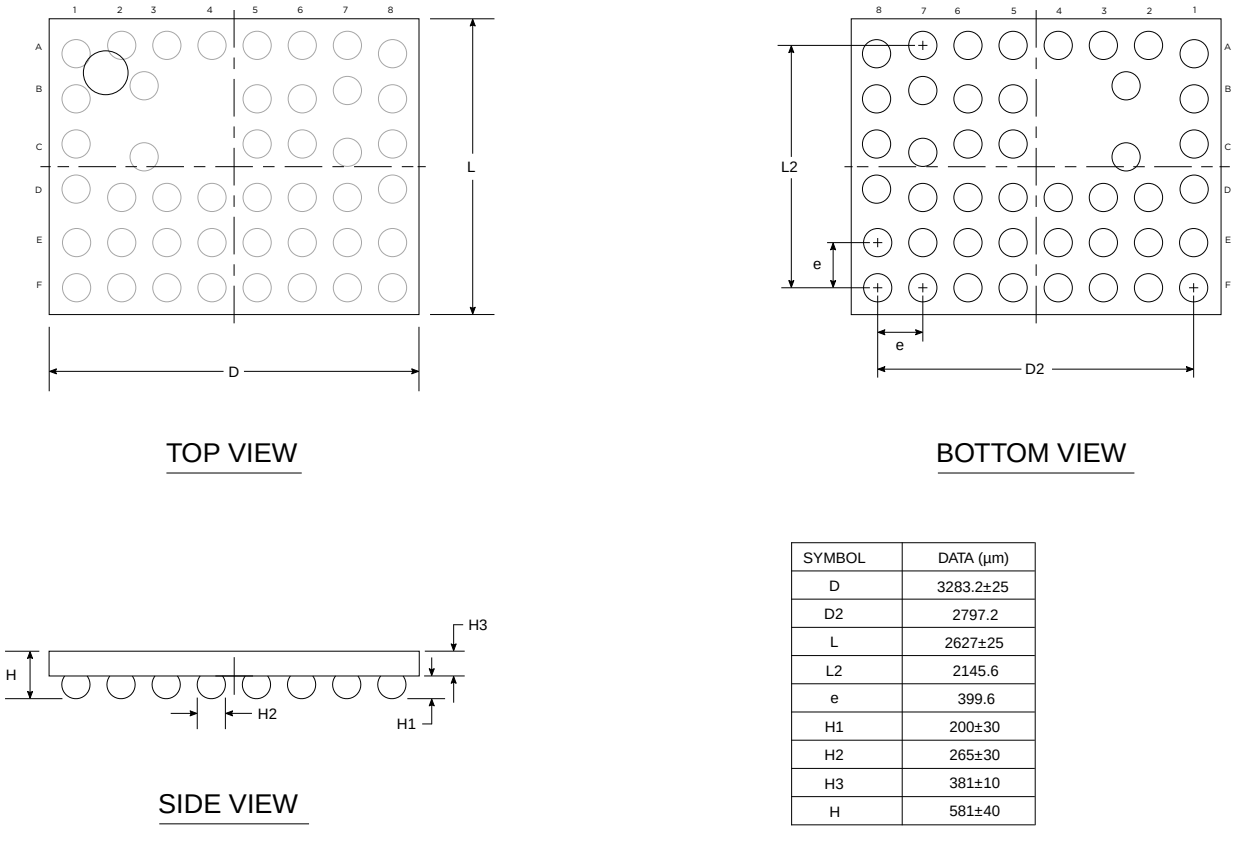


Figure 10.4. PCB layout example.

10.3. Mechanical Specifications

10.3.1. WLCSP Package



SYMBOL	DATA (µm)
D	3283.2±25
D2	2797.2
L	2627±25
L2	2145.6
e	399.6
H1	200±30
H2	265±30
H3	381±10
H	581±40

Figure 10.5. X4 WLCSP package.

See Section 2.1 for pin coordinates.

10.3.2. Recommended PCB Footprint

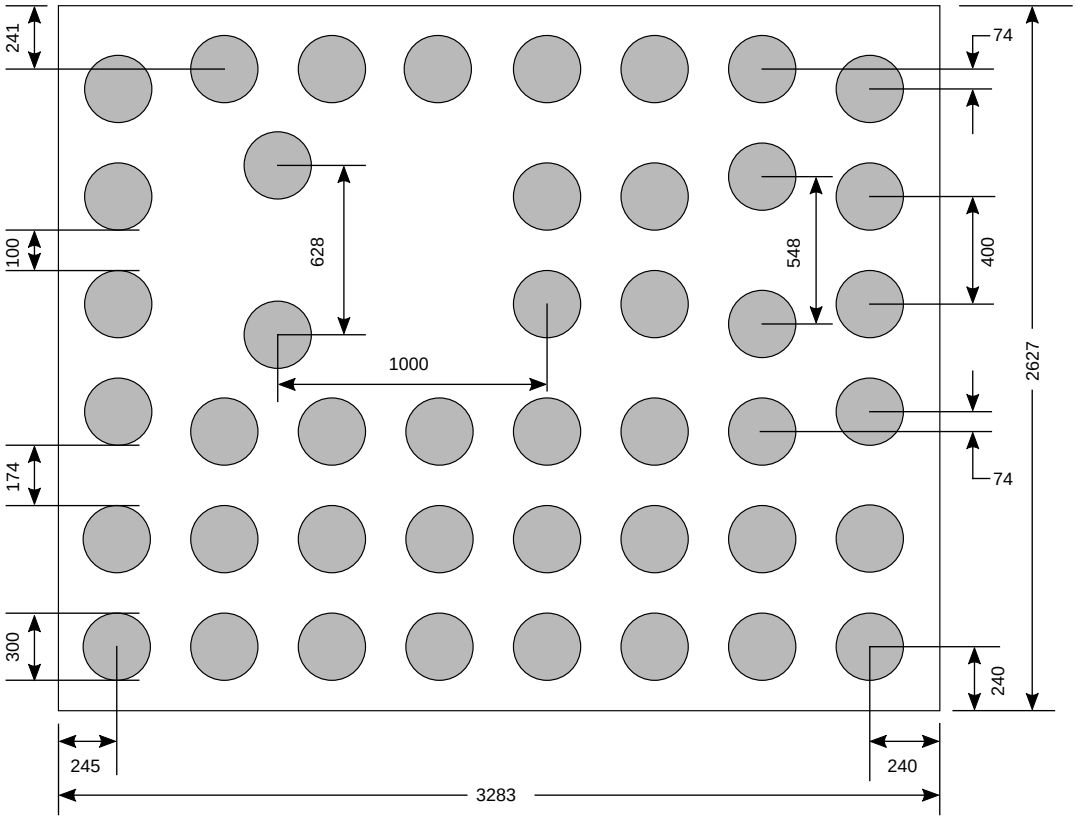


Figure 10.6. Recommended PCB footprint for X4 WLCSP. All dimensions in micrometers.

11. PIF register specification

Address	Name	Type
0x00	firmware_version	Register
0x06	gpio_out	Register
0x0B	gpio_in	Register
0x0D	gpio_oe	Register
0x0E	rx_mframes	Register
0x0F	smpl_mode	Register
0x10	rx_downconversion_coeff_i1	Register
0x11	rx_downconversion_coeff_i2	Register
0x14	rx_downconversion_coeff_q1	Register
0x15	rx_downconversion_coeff_q2	Register
0x16	rx_ram_write_offset_msb	Register
0x17	rx_ram_line_first_msb	Register
0x18	rx_ram_line_last_msb	Register
0x19	rx_ram_lsbs	Register
0x1B	rx_counter_num_bytes	Register
0x1C	rx_counter_lsb	Register
0x1D	fetch_radar_data_spi	Action
0x1E	fetch_radar_data_pif	Action
0x1F	radar_data_pif	Register
0x21	radar_data_pif_status	Register
0x27	ram_select	Register
0x2A	radar_readout_idle	Register
0x2B	rx_reset_counters	Action
0x2C	trx_clocks_per_pulse	Register
0x2D	rx_mframes_coarse	Register
0x2E	trx_pulses_per_step_msb	Register
0x2F	trx_pulses_per_step_lsb	Register
0x30	trx_dac_max_h	Register
0x31	trx_dac_max_l	Register
0x32	trx_dac_min_h	Register
0x33	trx_dac_min_l	Register
0x34	trx_dac_step	Register
0x35	trx_iterations	Register
0x36	trx_start	Action
0x37	trx_ctrl_done	Register
0x3A	trx_backend_done	Register
0x3B	trx_ctrl_mode	Register
0x3C	trx_lfsr_taps_0	Register
0x3D	trx_lfsr_taps_1	Register
0x3E	trx_lfsr_taps_2	Register
0x41	trx_lfsr_reset	Action

Address	Name	Type
0x42	rx_wait	Register
0x43	tx_wait	Register
0x44	trx_dac_override_h	Register
0x45	trx_dac_override_l	Register
0x46	trx_dac_override_load	Action
0x47	cpu_spi_master_clk_ctrl	Register
0x49	mclk_trx_backend_clk_ctrl	Register
0x4A	osc_ctrl	Register
0x4B	io_ctrl_1	Register
0x4C	io_ctrl_2	Register
0x4D	io_ctrl_3	Register
0x4E	io_ctrl_4	Register
0x4F	io_ctrl_5	Register
0x51	io_ctrl_6	Register
0x52	pif_mb_fifo_status	Register
0x53	to_cpu_read_data	Register
0x54	from_cpu_write_data	Register
0x55	pif_mb_clear_status	Register
0x56	pif_mem_fifo_status	Register
0x57	pif_mem_clear_status	Register
0x58	spi_master_send	Register
0x59	spi_master_radar_burst_kick	Action
0x5A	spi_master_idle	Register
0x5B	spi_master_mode	Register
0x5C	spi_master_radar_burst_size_lsb	Register
0x5E	boot_from_otp_pif	Register
0x5F	rx_pll_ctrl_1	Register
0x61	rx_pll_ctrl_2	Register
0x64	rx_pll_status	Register
0x65	tx_pll_ctrl_1	Register
0x66	tx_pll_ctrl_2	Register
0x69	tx_pll_status	Register
0x6A	common_pll_ctrl_1	Register
0x6B	common_pll_ctrl_2	Register
0x6C	common_pll_ctrl_3	Register
0x6D	common_pll_ctrl_4	Register
0x6E	common_pll_frac_2	Register
0x6F	common_pll_frac_1	Register
0x71	common_pll_frac_0	Register
0x72	lock_status	Register
0x73	clkout_sel	Register
0x74	apc_dvdd_testmode	Register

Address	Name	Type
0x75	misc_ctrl	Register
0x76	dvdd_rx_ctrl	Register
0x78	dvdd_tx_ctrl	Register
0x79	dvdd_testmode	Register
0x7A	avdd_rx_ctrl	Register
0x7B	avdd_tx_ctrl	Register
0x7C	avdd_testmode	Register
0x7D	ldo_status_1	Register
0x7E	ldo_status_2	Register
0x7F	spi_config_pif	Register

Table 11.1. Register overview

0x00 firmware_version

Read-writable 8-bit register.

Register writable by firmware to define its version. Readable on SPI as firmware_version_spi.

Bit	Segment name	Default	Description
[7:0]	firmware_version	0x00	

0x06 gpio_out

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7]	-	NA	
[6:0]	gpio_o	0x00	Each of these register bits drive the output pin of the corresponding IO pad when they are configured in GPIO mode.

0x0B gpio_in

Read-only 8-bit register.

Bit	Segment name	Default	Description
[7]	-	NA	
[6:0]	gpio_i	0x00	Input signal on each IO pin is synchronized to the CPU clock and can be read from this register.

0x0D gpio_oe

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7]	-	NA	
[6:0]	gpio_output_enable	0x00	Output enable for each IO pad in GPIO mode. 0: Output driver disabled. 1: Output driver enabled.

0x0E rx_mframes

Read-writable 8-bit register.

Defines the frame length in steps of 12 sampling bins. Number of counters enabled is min(1536, (rx_mframes + 1)*12).

This register controls how many sampled radar bins are written to RAM. The number of physically sampled radar bins is controlled by rx_mframes_coarse.

Bit	Segment name	Default	Description
[7:0]	rx_mframes	0x80	

0x0F smpl_mode

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7:3]	-	NA	
[2]	Reserved	0x00	Always read and write as 0
[1]	powerdown_sampler	0x01	Powerdown of samplers. Also disables distribution of the RX PLL clock to the receiver backend. 1: Samplers (and receiver back end) are powered down (only static leakage) 0: Samplers are active
[0]	rx_downconversion_enable	0x00	Enable downconversion of sampled signal. This signal affects how the internal data is stored. It must therefore not change between frame sampling (trx_start to trx_backend_done) and frame readout (started by fetch_radar_data_pif or fetch_radar_data_spi).

0x10 rx_downconversion_coeff_i1

Write-only 8-bit register.

Bit	Segment name	Default	Description
[7:6]	-	NA	
[5:0]	rx_downconversion_coeff_i1	0x00	Set of coefficients for downconversion. Weight vector 1 for I-channel. The coefficients cannot be read back. The register works like a 32-deep FIFO - each register has 32 coefficients behind it. They should be written most significant coefficient first.

0x11 rx_downconversion_coeff_i2

Write-only 8-bit register.

Bit	Segment name	Default	Description
[7:6]	-	NA	
[5:0]	rx_downconversion_coeff_i2	0x00	Set of coefficients for downconversion. Weight vector 2 for I-channel. See also rx_downconversion_coeff_i1.

0x14 rx_downconversion_coeff_q1

Write-only 8-bit register.

Bit	Segment name	Default	Description
[7:6]	-	NA	
[5:0]	rx_downconversion_coeff_q1	0x00	Set of coefficients for downconversion. Weight vector 1 for Q-channel. See also rx_downconversion_coeff_i1.

0x15 rx_downconversion_coeff_q2

Write-only 8-bit register.

Bit	Segment name	Default	Description
[7:6]	-	NA	
[5:0]	rx_downconversion_coeff_q2	0x00	Set of coefficients for downconversion. Weight vector 2 for Q-channel. See also rx_downconversion_coeff_i1.

0x16 rx_ram_write_offset_msb

Read-writable 8-bit register.

MSB of rx_ram_write_offset. The first RAM line to be written when generating a new frame. Can be used to stitch together sub-frames.

Bit	Segment name	Default	Description
[7:0]	rx_ram_write_offset_msb	0x00	

0x17 rx_ram_line_first_msb

Read-writable 8-bit register.

MSB of rx_ram_line_first. The first RAM line to read when reading radar data.

Bit	Segment name	Default	Description
[7:0]	rx_ram_line_first_msb	0x00	

0x18 rx_ram_line_last_msb

Read-writable 8-bit register.

MSB of rx_ram_line_last. The last RAM line to read when reading radar data.

Bit	Segment name	Default	Description
[7:0]	rx_ram_line_last_msb	0xBF	

0x19 rx_ram_lsbs

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7:3]	-	NA	
[2]	rx_ram_write_offset_lsb	0x00	LSB of rx_ram_write_offset.
[1]	rx_ram_line_first_lsb	0x00	LSB of rx_ram_line_first.
[0]	rx_ram_line_last_lsb	0x01	LSB of rx_ram_line_last.

0x1B rx_counter_num_bytes

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7:3]	-	NA	
[2:0]	rx_counter_num_bytes	0x03	Number of bytes to read from each counter. 0 : Read 8 bytes 1-7: Read 1-7 bytes, respectively In downconversion, negative values are sign extended to the selected byte width.

0x1C rx_counter_lsb

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7:6]	-	NA	
[5:0]	rx_counter_lsb	0x00	Selects the least significant bit to read. Can be any value from 0 to 45. Note that there is no real data above 24 bits in normal mode and 46 bits in downconversion mode, only sign-extended padding.

0x1D fetch_radar_data_spi

Executable Action.

When this action is triggered, the SPI radar_data FIFO is emptied, and new radar data is streamed into it. Important note: This register causes an asynchronous reset of flip-flops in the SPI domain. This is only safe as long as the SPI clock is not running. A toggle of SPI clock around the time of the reset recovery will cause a timing problem. Therefore, the system handshaking between the external SPI master and the CPU must ensure that the SPI is silent when fetch_radar_data_spi is triggered. (I.e., after the external master has asked for a frame, it should NOT poll a register until data is ready, but rather wait for a handshake signal from the CPU.)

0x1E fetch_radar_data_pif

Executable Action.

When this action is triggered, the PIF radar_data FIFO is emptied, and new radar data is streamed into it.

0x1F radar_data_pif

Read-only 8-bit register.

Radar data, made available by fetch_radar_data_pif.

Bit	Segment name	Default	Description
[7:0]	radar_data_pif	0x00	

0x21 radar_data_pif_status

Read-only 8-bit register.

Bit	Segment name	Default	Description
[7]	Reserved	0x00	
[6]	Reserved	0x00	
[5]	Reserved	0x00	
[4]	Reserved	0x00	
[3:1]	-	NA	
[0]	radar_data_pif_fifo_empty	0x01	Status flag indicating whether radar data FIFO is empty

0x27 ram_select

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7:1]	-	NA	
[0]	ram_select	0x00	Selects which RAM to write the sampled counter values into, and selects the opposite RAM for reading out data from last frame. This bit should never be changed while sampling is ongoing (between trx_start and trx_backend_done), or while a radar readout is ongoing. If the frames are pipelined (reading the last frame while a new frame is written), toggle ram_select at a safe time after trx_backend_done and completed readout of the last frame. In the non-pipelined case, ram_select must be toggled between every write and read.

0x2A radar_readout_idle

Read-only 8-bit register.

Bit	Segment name	Default	Description
[7:1]	-	NA	
[0]	radar_readout_idle	0x01	Indicates that there is no ongoing radar readout

0x2B rx_reset_counters

Executable Action.

Clears the result of the last sweep from the samplers. Always call this before initiating a new sweep.

0x2C trx_clocks_per_pulse

Read-writable 8-bit register.

Controls the divider that determines the PRF (PRF will be Common PLL output frequency divided by the value of this register).

Legal values are between 4 and 255. Values outside this range will result in undefined behaviour.

Bit	Segment name	Default	Description
[7:0]	trx_clocks_per_pulse	0x10	

0x2D rx_mframes_coarse

Read-writable 8-bit register.

Defines the frame length in steps of 96 sampling bins.

This register controls the size of the frame that is actually captured by the front end. See also rx_mframes.

It is important that this register is always set less than or equal to trx_clocks_per_pulse.

Bit	Segment name	Default	Description
[7:0]	rx_mframes_coarse	0x10	

0x2E trx_pulses_per_step_msb

Read-writable 8-bit register.

MSB of pulses-per-step.

Bit	Segment name	Default	Description
[7:0]	trx_pulses_per_step_msb	0x00	

0x2F trx_pulses_per_step_lsb

Read-writable 8-bit register.

LSB of pulses-per-step. Number of pulses integrated per DAC step.

Always set pulses-per-step to 1 or higher. Setting pulses-per-step to 0 may lead to undefined behaviour.

Bit	Segment name	Default	Description
[7:0]	trx_pulses_per_step_lsb	0x14	

0x30 trx_dac_max_h

Read-writable 8-bit register.

MSB of trx_dac_max, the maximum value of the DAC sweep. Always set max value higher than or equal to trx_dac_min.

Bit	Segment name	Default	Description
[7:0]	trx_dac_max_h	0xFF	

0x31 trx_dac_max_l

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7:3]	-	NA	
[2:0]	trx_dac_max_l	0x07	3 least significant bits of trx_dac_max.

0x32 trx_dac_min_h

Read-writable 8-bit register.

MSB of trx_dac_min, the minimum value of the DAC sweep. Always set this field lower than or equal to trx_dac_max.

Bit	Segment name	Default	Description
[7:0]	trx_dac_min_h	0x00	

0x33 trx_dac_min_l

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7:3]	-	NA	
[2:0]	trx_dac_min_l	0x00	3 least significant bits of trx_dac_min.

0x34 trx_dac_step

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7:6]	-	NA	
[5]	trx_auto_bidir_enable	0x01	Enable/disable bidirectional sweeps
[4]	trx_sweep_phase	0x00	Set the initial sweep direction of bi-directional sweeps, and the direction of single direction sweeps. 0: Sweep starts at trx_dac_min and sweeps upwards 1: Sweep starts at trx_dac_max and sweeps downwards
[3:2]	trx_dac_settle_clog2	0x00	Sets clocks for DAC to settle: 0: 1 clock settling time 1: 2 clocks settling time 2: 4 clocks settling time 3: 8 clocks settling time
[1:0]	trx_dac_step_clog2	0x00	Sets step size of DAC sweep: 0: DAC step of 1 1: DAC step of 2 2: DAC step of 4 3: DAC step of 8

0x35 trx_iterations

Read-writable 8-bit register.

Number of sweep iterations to perform.

It is highly recommended to set trx_iterations to a value that is integer divisible by (2^noiseless_ghost_order) * (2^trx_auto_bidir_enable).

Bit	Segment name	Default	Description
[7:0]	trx_iterations	0x0A	

0x36 trx_start

Executable Action.

Signal the sweep controller to initiate sampling of a new radar frame. Note that this action is ignored when the transceiver is in send-every-pulse mode, controlled by the trx_send_every_pulse bit.

0x37 trx_ctrl_done

Read-only 8-bit register.

Bit	Segment name	Default	Description
[7:1]	-	NA	
[0]	trx_ctrl_done	0x01	Set high by the sweep controller when radar frame sampling is complete. Note that the receiver backend performs some additional post-processing before the frame can be read. It is recommended to check trx_backend_done before attempting to read the new radar frame.

0x3A trx_backend_done

Read-only 8-bit register.

Bit	Segment name	Default	Description
[7:1]	-	NA	
[0]	trx_backend_done	0x01	Set high by the receiver backend when a new radar frame is ready.

0x3B trx_ctrl_mode

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7]	tx_strobe_enable	0x01	Mask out tx strobe when disabled (force tx_strobe to 0 when tx_strobe_enable = 0)
[6]	rx_strobe_enable	0x01	Mask out rx strobe when disabled (force rx_strobe to 0 when rx_strobe_enable = 0)
[5]	trx_phase_override	0x00	Set this bit to override the trx_phase signal, setting it to trx_phase_override_val instead of LFSR output
[4]	trx_phase_override_val	0x00	Value of trx_phase when trx_phase_override is enabled
[3]	trx_send_every_pulse	0x00	When this bit is set, the transceiver is in send-every-pulse mode where pulses are transmitted with a constant rate determined by trx_clocks_per_pulse. When disabled, pulse transmission is only enabled while a sweep is in progress. Note that this bit masks out the trx_start action and is only intended to be used to test the transmitter.
[2]	trx_constant_clocks_per_pulse	0x00	Keep the number of clock cycles between consecutive pulses constant within one sweep.
[1]	trx_ctrl_slave_mode	0x00	Set transceiver control in slave mode. In this mode, trx_start is driven by the trx_start_sync_in input signal instead of the trx_start PIF register.
[0]	trx_start_sync_negedge	0x00	Synchronize trx_start_sync_out to negative clock edge instead of positive. Can be used as a workaround to fix timing issues.

0x3C trx_lfsr_taps_0

Read-writable 8-bit register.

LFSR feedback taps, least significant 8 bits. Setting bit n results in tapping bit position n of the LFSR register, effectively configuring the polynomial of the pseudo random pulse bi-phasing function.

Bit	Segment name	Default	Description
[7:0]	trx_lfsr_taps_0	0x00	

0x3D trx_lfsr_taps_1

Read-writable 8-bit register.

LFSR feedback taps, middle 8 bits.

Bit	Segment name	Default	Description
[7:0]	trx_lfsr_taps_1	0x00	

0x3E trx_lfsr_taps_2

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7]	-	NA	
[6:4]	noiseless_ghost_order	0x01	Set the noiseless ghost order. Legal values: 0-6.
[3:0]	trx_lfsr_taps_2	0x09	LFSR feedback taps, most significant 4 bits.

0x41 trx_lfsr_reset

Executable Action.

Write to this action to reset the LFSR to the trx_lfsr_taps value. By default, the LFSR will continue from wherever it was last radar frame.

0x42 rx_wait

Read-writable 8-bit register.

Controls the frame offset by delaying the receiver with rx_wait Common-PLL clock cycles.

Bit	Segment name	Default	Description
[7:0]	rx_wait	0x00	

0x43 tx_wait

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7:4]	-	NA	
[3:0]	tx_wait	0x00	Controls the frame offset by delaying the transmitter with tx_wait Common-PLL clock cycles.

0x44 trx_dac_override_h

Read-writable 8-bit register.

The trx_dac_override_load action must be triggered for changes to this register to take effect.

Bit	Segment name	Default	Description
[7]	trx_dac_override_enable	0x00	When this bit is set, the DAC is controlled by trx_dac_override_val_h/l. When this bit is cleared, the DAC is controlled by the Sweep Controller.
[6:3]	-	NA	
[2:0]	trx_dac_override_val_h	0x00	MSB of trx_dac_override value. Can be used to force the DAC to a specific value. Enabled by trx_dac_override_enable.

0x45 trx_dac_override_l

Read-writable 8-bit register.

The trx_dac_override_load action must be triggered for changes to this register to take effect.

Bit	Segment name	Default	Description
[7:0]	trx_dac_override_val_l	0x00	8 least significant bits of of trx_dac_override value.

0x46 trx_dac_override_load

Executable Action.

Write to this action for trx_dac_override settings to take place. Both trx_dac_override_enable and trx_override_val_h/l have shadow registers that are loaded by trx_dac_override_load.

0x47 cpu_spi_master_clk_ctrl

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7:5]	spi_master_clk_div	0x00	Select division of spi_master_clk from source. 0: disabled 1: divide by 1 2: divide by 2 3: divide by 4 4: divide by 8 5: divide by 16 6: divide by 32 7: divide by 64
[4]	spi_master_clk_sel	0x00	0: spi_master_clk is sourced from sysclk 1: spi_master_clk is sourced from auxclk
[3:1]	cpuclk_div	0x00	Select division of cpuclk from source. Divide by 2^N.
[0]	cpuclk_sel	0x00	0: cpuclk is sourced from sysclk 1: cpuclk is sourced from auxclk

0x49 mclk_trx_backend_clk_ctrl

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7:6]	mclk_sel	0x00	2'b00: mclk is sourced from sysclk 2'b01: mclk is sourced from auxlkc 2'b1x: mclk is sourced from lvdsclk_in
[5:3]	trx_backend_clk_prescale	0x00	Select division of prescaled trx_backend_clk from refclk 0: Disabled 1: Divide by 1 2: Divide by 2 3: Divide by 4 4: Divide by 8 5: Divide by 16 6: Divide by 32 7: Divide by 64
[2:0]	trx_backend_clk_div	0x00	Select division of trx_backend_clk from prescaled signal. 0: Divide by 1 1: Divide by 2 2: Divide by 3 3: Divide by 4 4: Divide by 5 5: Divide by 6 6: Divide by 7 7: Divide by 8

0x4A osc_ctrl

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7]	-	NA	
[6]	auxclk_sel	0x01	Select auxclk source signal: 0: pllauxclk 1: extclk
[5]	sysclk_sel	0x00	Select sysclk source signal: 0: lpclk 1: xoscclk
[4]	xosc_bypass	0x00	Set crystal oscillator in bypass mode.
[3]	xosc_dislvl	0x01	Disable level shifter in crystal oscillator.
[2]	xosc_discap	0x00	Disable built in decoupling capacitors in crystal oscillator.
[1]	xosc_en	0x00	Enable crystal oscillator.
[0]	lpclk_disable	0x00	Disable low-power oscillator.

0x4B io_ctrl_1

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7]	spiclk_ren	0x00	Active low enable of pullup resistor for spiclk pin.
[6:5]	spio_ren	0x00	Active low enable of pullup resistor for SPIO pins.
[4:0]	io_ren	0x00	Active low enable of pullup resistor for IO pins.

0x4C io_ctrl_2

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7:6]	lvds_rt_cal	0x00	LVDS termination resistor calibration bits. Nominal values: 00 = 80 Ohm (-20%) 01 = 90 Ohm (-10%) 10 = 103 Ohm (+3%) 11 = 120 Ohm (+20%)
[5]	lvds_rt_en	0x00	Enable LVDS internal termination resistor.
[4]	lvcmos_schmitt_en	0x00	Enable hysteresis in GPIO5/GPIO6 input CMOS buffers.
[3]	io6_lvcmos_ie	0x01	Active high enable of CMOS input buffer for IO6 pin.
[2]	io5_lvcmos_ie	0x01	Active high enable of CMOS input buffer for IO5 pin.
[1]	lvds_rx_en	0x00	Enable LVDS receiver.
[0]	sub_lvds_en	0x00	Enable sub-LVDS mode. 0: Nominal common mode voltage of LVDS is 1.2V 1: Nominal common mode voltage of LVDS is 0.9V

0x4D io_ctrl_3

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7]	lvds_tx_en	0x00	Enable LVDS transmitter: 0 = LVDS outputs High-Z and LVCMOS buffers are controlled by GPIO5/6 control signals 1 = LVDS Driver enabled and LVCMOS buffers are disabled
[6:0]	io_gpio_sel	0x66	Select whether pin is used as GPIO. This setting takes precedence over other pin functionality. Each bit represents the corresponding IO pin (e.g. bit 0 represents IO0). For each bit: 0: Pin is used for other functionality 1: Pin is used for GPIO

0x4E io_ctrl_4

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7]	Reserved	0x00	Always read and write as 0
[6]	io6_clkout_sel	0x00	Select which clock signal to transmit on the IO6 pin 0 = trx_clkout_2 1 = cmu_clkout
[5]	io5_clkout_sel	0x00	Select which clock signal to transmit on the IO5 pin 0 = trx_clkout_1 1 = cmu_clkout
[4]	extclk_sel	0x00	Select which of IO6 and IO5 the extclk signal is sourced from. 0 = IO6 1 = IO5
[3:2]	lvds_txdrv	0x00	LVDS driver output current control. 00 = Output driver current 0mA. If lvds_rt_en==1'b0: 2'b01 = Nominal output driver current 1.2mA. 2'b10 = Nominal output driver current 2.4mA. 2'b11 = Nominal output driver current 3.6mA. If lvds_rt_en==1'b1: 2'b01 = Nominal output driver current 2.4mA. 2'b10 = Nominal output driver current 4.8mA. 2'b11 = Nominal output driver current 7.2mA.
[1:0]	lvds_biastrim	0x00	Output current bias resistor trim: 00 = +26% 01 = +13% 10 = 0% 11 = -18%

0x4F io_ctrl_5

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7:3]	lpclk_trim	0x00	Trim lpclk, encoded as a 5-bit signed value. 0 -> 0% trim (nominal).
[2]	io3_spi_sel	0x01	Select whether pin is used as SPI. io_gpio_sel[3] must be set to 0 to enable this setting. 0: IO3 is used for trx_sync output. Output driver is automatically enabled. 1: IO3 is used for SPI (SPIO2). Output enable is controlled by SPI.
[1]	trx_sync_source_sel	0x00	Select source for trx_start_sync_in. 0 = IO1 1 = IO3
[0]	io1_spi_sel	0x01	Select whether pin is used as SPI. io_gpio_sel[1] must be set to 0 to enable this setting. 0: IO1 is used for trx_sync output. Output driver is automatically enabled. 1: IO1 is used for SPI (NSS1_OUT)

0x51 io_ctrl_6

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7:5]	-	NA	
[4:2]	Reserved	0x00	Always read and write as 0
[1]	io6_clkout_oe	0x00	Enable output driver on IO6 pin when clkout function is enabled
[0]	io5_clkout_oe	0x00	Enable output driver on IO5 pin when clkout function is enabled

0x52 pif_mb_fifo_status

Read-only 8-bit register.

Bit	Segment name	Default	Description
[7]	pif_to_cpu_fifo_overflow	0x00	Indicates an overflow in to_cpu_fifo (mailbox).
[6]	pif_to_cpu_fifo_underflow	0x00	Indicates an underflow in to_cpu_fifo (mailbox).
[5]	pif_from_cpu_fifo_overflow	0x00	Indicates an overflow in from_cpu_fifo (mailbox).
[4]	pif_from_cpu_fifo_underflow	0x00	Indicates an underflow in from_cpu_fifo (mailbox).
[3]	-	NA	
[2]	from_cpu_fifo_empty	0x01	Goes high to indicate to the CPU that from_cpu_fifo is empty, i.e. data has been fully read. Note that data transmission has not necessarily been completed, but the last word has been read from FIFO by the SPI.
[1]	to_cpu_data_valid	0x00	Goes high if there is valid data in the incoming mailbox FIFO (see to_cpu_read_data).
[0]	from_cpu_fifo_full	0x00	Goes high if there is no more space in the read_data FIFO. Must be checked before writing to from_cpu_write_data.

0x53 to_cpu_read_data

Read-only 8-bit register.

When reading this, data is popped from mailbox FIFO. Valid only if to_cpu_data_valid is 1.

Bit	Segment name	Default	Description
[7:0]	to_cpu_read_data	0x00	

0x54 from_cpu_write_data

Write-only 8-bit register.

A write to this register goes into the 8 byte deep outgoing mailbox FIFO data.

Bit	Segment name	Default	Description
[7:0]	from_cpu_write_data	0x00	

0x55 pif_mb_clear_status

Write-only 8-bit register.

Bit	Segment name	Default	Description
[7]	pif_clear_to_cpu_fifo_overflow	0x00	Write 1 to clear to_cpu_fifo_overflow
[6]	pif_clear_to_cpu_fifo_underflow	0x00	Write 1 to clear to_cpu_fifo_underflow
[5]	pif_clear_from_cpu_fifo_overflow	0x00	Write 1 to clear from_cpu_fifo_overflow
[4]	pif_clear_from_cpu_fifo_underflow	0x00	Write 1 to clear from_cpu_fifo_underflow
[3:0]	-	NA	

0x56 pif_mem_fifo_status

Read-only 8-bit register.

Bit	Segment name	Default	Description
[7]	pif_to_mem_fifo_overflow	0x00	Indicates an overflow in to_mem_fifo (memory programming).
[6]	pif_to_mem_fifo_underflow	0x00	Indicates an underflow in to_mem_fifo (memory programming). Should not happen by design.
[5]	pif_from_mem_fifo_overflow	0x00	Indicates an overflow in from_mem_fifo (memory readback). Should not happen by design.
[4]	pif_from_mem_fifo_underflow	0x00	Indicates an underflow in from_mem_fifo (memory readback).
[3:0]	-	NA	

0x57 pif_mem_clear_status

Write-only 8-bit register.

Bit	Segment name	Default	Description
[7]	pif_clear_to_mem_fifo_overflow	0x00	Write 1 to clear to_mem_fifo_overflow
[6]	pif_clear_to_mem_fifo_underflow	0x00	Write 1 to clear to_mem_fifo_underflow
[5]	pif_clear_from_mem_fifo_overflow	0x00	Write 1 to clear from_mem_fifo_overflow
[4]	pif_clear_from_mem_fifo_underflow	0x00	Write 1 to clear from_mem_fifo_underflow
[3:0]	-	NA	

0x58 spi_master_send

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7:1]	-	NA	
[0]	spi_master_send	0x00	Set to 1 to send data from SPI master. As long as this bit is high, NSS will be active out of the chip. Data already in the mailbox FIFO will be sent immediately, and the CPU can add more data to the same SPI transaction by writing more data to the FIFO. To send radar data, the CPU should write to spi_master_radar_burst_kick. The SPI Master will then send spi_master_radar_burst_size number of bytes when they are available in the radar_data_pif FIFO. NSS is directly controlled by this register, so spi_slave_sel should be set before spi_master_send, and should not be changed while spi_master_send is high. When a transaction is complete, the CPU must write 0 in spi_master_send to release NSS. To see whether the SPI Master is done sending the current data, please see spi_master_idle.

0x59 spi_master_radar_burst_kick

Executable Action.

Trigger this action to send radar data in SPI master mode. See also spi_master_send.

0x5A spi_master_idle

Read-only 8-bit register.

Bit	Segment name	Default	Description
[7:2]	-	NA	
[1]	spi_master_nss	0x01	Indicates the state of NSS from spi_master. Should be used actively by firmware when spi_master_clk is much slower than cpuclk. When firmware wants to leave spi_master_send mode and re-enter it soon after, it is important that the slower SPI Master has seen the transition back and forth. In such a case, firmware should leave spi_master_send mode, then poll until spi_master_nss is high, before entering spi_master_send mode again.
[0]	spi_master_idle	0x01	0: SPI Master is busy 1: SPI Master is in IDLE (no activity), or in MAILBOX_WAIT, i.e. in the middle of a mailbox transaction, waiting for data. This flag should be used actively during SPI Master mailbox send, and should be checked before writing 0 to spi_master_send. (Especially in QSPI - in single-wire SPI, the CPU will typically wait for return data in to_cpu_read_data. When the expected amount of return data has been received, it is not necessary to check spi_master_idle.) Due to the elastic nature of the asynchronous FIFO used in the mailbox, it is possible for the FIFO to look empty (and therefore for this bit to be 1) right after the CPU has written to from_cpu_write_data. One should therefore always have a gap after the last write before checking this bit. (How large a gap is necessary is dependent on the relationship between the SPI Master clock and the CPU clock.) In practice, this could mean to read the flag several times.

0x5B spi_master_mode

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7:3]	spi_master_radar_burst_size_msb	0x12	MSB of spi_master_radar_burst_size. See spi_master_send.
[2:1]	spi_slave_sel	0x00	Select which SPI slave to address when in SPI Master mode. 0: Use NSS0 1: Use NSS1 2: Use NSS2 3: Reserved. Behaviour is undefined.
[0]	spi_master_mode	0x00	0: SPI interface works in slave mode 1: SPI interface works in master mode

0x5C spi_master_radar_burst_size_lsb

Read-writable 8-bit register.

LSB of spi_master_radar_burst_size. See spi_master_send.

Bit	Segment name	Default	Description
[7:0]	spi_master_radar_burst_size_lsb	0x00	

0x5E boot_from_otp_pif

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7:1]	-	NA	
[0]	boot_from_otp_pif	0x01	By default, the hardware enables a small boot ROM that jumps to the OTP. Write 0 to this register to boot directly from program memory (SRAM).

0x5F rx_pll_ctrl_1

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7]	-	NA	
[6:4]	rx_pll_fbdiv	0x06	RX PLL feedback divider setting 3'b000 : 2 3'b001 : 3 3'b010 : 4 3'b011 : 5 3'b100 : 6 3'b101 : 7 3'b110 : 8 3'b111 : 9
[3:0]	rx_pll_foutt_sel	0x00	Select phase for rx_pll test output signal

0x61 rx_pll_ctrl_2

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7:6]	-	NA	
[5:3]	rx_pll_postdiv	0x00	Divider setting for divided output. 3'b000 : 5 3'b001 : 2 3'b010 : 3 3'b011 : 4 3'b100 : 5 3'b101 : 6 3'b110 : 7 3'b111 : 8
[2]	rx_pll_powerdown	0x01	Control power down of RX PLL
[1]	rx_pll_powerdown_foutdiv	0x01	Power down test output divider
[0]	rx_pll_powerdown_foutt	0x01	Power down test output driver

0x64 rx_pll_status

Read-only 8-bit register.

Bit	Segment name	Default	Description
[7]	rx_pll_lock	0x01	Lock signal from RX PLL.
[6:0]	Reserved	0x00	

0x65 tx_pll_ctrl_1

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7]	-	NA	
[6:4]	tx_pll_fbdiv	0x03	TX PLL feedback divider setting 3'b000 : 2 3'b001 : 3 3'b010 : 4 3'b011 : 5 3'b100 : 6 3'b101 : 7 3'b110 : 8 3'b111 : 9
[3:0]	tx_pll_foutt_sel	0x00	Select phase for tx_pll test output signal

0x66 tx_pll_ctrl_2

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7:6]	-	NA	
[5:3]	tx_pll_postdiv	0x00	Divider setting for divided output. 3'b000 : 5 3'b001 : 2 3'b010 : 3 3'b011 : 4 3'b100 : 5 3'b101 : 6 3'b110 : 7 3'b111 : 8
[2]	tx_pll_powerdown	0x01	Control power down of TX PLL
[1]	tx_pll_powerdown_foutdiv	0x01	Power down test output divider
[0]	tx_pll_powerdown_foutt	0x01	Power down test output driver

0x69 tx_pll_status

Read-only 8-bit register.

Bit	Segment name	Default	Description
[7]	tx_pll_lock	0x01	Lock signal from TX PLL.
[6:0]	Reserved	0x00	

0x6A common_pll_ctrl_1

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7]	common_pll_powerdown	0x01	Power down entire common PLL
[6]	common_pll_dsmpd	0x01	Power down delta-sigma modulator in common PLL
[5]	common_pll_dacpd	0x01	Power down noise cancelling DAC in common PLL
[4]	common_pll_internal_bypass	0x00	Bypass common PLL. Connected to frac pll bypass pin. (Output disabled when common PLL is in power down)
[3:0]	common_pll_fbdiv_msb	0x00	Upper bits of common_pll_fbdiv. Feedback divider setting in common PLL. Valid range 16 - 1600.

0x6B common_pll_ctrl_2

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7:0]	common_pll_fbdiv_lsb	0x12	Lower bits of common_pll_fbdiv. Feedback divider setting in common PLL. Valid range 16 - 1600.

0x6C common_pll_ctrl_3

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7]	common_pll_foutpostdivpd	0x00	Power down post divider in Common PLL
[6]	common_pll_foutvcopd	0x01	Power down VCO direct output in Common PLL
[5:0]	common_pll_refdiv	0x01	Reference divider setting in Common PLL

0x6D common_pll_ctrl_4

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7]	common_pll_fout4phasepd	0x00	Power down 4 phase generator in Common PLL.
[6]	common_pll_external_bypass	0x00	Bypass Common PLL when Common PLL is in power down.
[5:3]	common_pll_postdiv1	0x02	Post divider 1 setting in Common PLL. The value of common_pll_postdiv1 should always be greater than or equal to common_pll_postdiv2.
[2:0]	common_pll_postdiv2	0x01	Post divider 2 setting in Common PLL.

0x6E common_pll_frac_2

Read-writable 8-bit register.

Byte 2 (MSB) of fractional divider setting in Common PLL.

Bit	Segment name	Default	Description
[7:0]	common_pll_frac_2	0x00	

0x6F common_pll_frac_1

Read-writable 8-bit register.

Byte 1 of fractional divider setting in Common PLL.

Bit	Segment name	Default	Description
[7:0]	common_pll_frac_1	0x00	

0x71 common_pll_frac_0

Read-writable 8-bit register.

Byte 0 (LSB) of fractional divider setting in Common PLL.

Bit	Segment name	Default	Description
[7:0]	common_pll_frac_0	0x00	

0x72 lock_status

Read-only 8-bit register.

Bit	Segment name	Default	Description
[7]	common_pll_lock	0x01	Indicates that the common PLL has locked.
[6]	xosc_lock	0x00	Indicates that the XOSC has locked.
[5:0]	-	NA	

0x73 clkout_sel

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7:6]	cmu_clkout_sel	0x00	Select which clock signal to output as cmu_clkout. 0: Off 1: cpuclk 2: spi_masterclk 3: trx_backend_clk
[5:4]	pllauxclk_sel	0x03	Select source of pllauxclk 0: refclk divided by 2 1: refclk divided by 4 2: refclk divided by 6 3: refclk divided by 8
[3:2]	Reserved	0x00	Always read and write as 0
[1:0]	Reserved	0x00	Always read and write as 0

0x74 apc_dvdd_testmode

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7:4]	apc_testmode	0x00	Enable APC testmode.
[3:0]	dvdd_testmode	0x00	Enable DVDD testmode.

0x75 misc_ctrl

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7]	Reserved	0x00	Always read and write as 0
[6:5]	tx_power	0x00	Output power selection. 0: TX off 1: Low 2: Medium 3: High
[4:0]	dvdd_trim	0x10	Control output voltage of DVDD LDO.

0x76 dvdd_rx_ctrl

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7]	-	NA	
[6]	dvdd_rx_disable	0x01	Disable DVDD_RX LDO.
[5]	dvdd_rx_disable_pulldown	0x00	Enable/Disable pulldown resistor when DVDD_RX is disabled.
[4:0]	dvdd_rx_trim	0x13	Control output voltage of DVDD_RX LDO.

0x78 dvdd_tx_ctrl

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7]	-	NA	
[6]	dvdd_tx_disable	0x01	Disable DVDD_TX LDO.
[5]	dvdd_tx_disable_pulldown	0x00	Enable/Disable pulldown resistor when DVDD_TX is disabled.
[4:0]	dvdd_tx_trim	0x10	Control output voltage of DVDD_TX LDO.

0x79 dvdd_testmode

Read-writable 8-bit register.

Enable DVDD testmode.

Bit	Segment name	Default	Description
[7:4]	dvdd_rx_testmode	0x00	Enable DVDD_RX LDO testmode.
[3:0]	dvdd_tx_testmode	0x00	Enable DVDD_TX LDO testmode.

0x7A avdd_rx_ctrl

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7]	-	NA	
[6]	avdd_rx_disable	0x01	Disable AVDD_RX LDO.
[5]	avdd_rx_disable_pulldown	0x00	Enable/Disable pulldown resistor when AVDD_RX is disabled.
[4:0]	avdd_rx_trim	0x13	Control output voltage of AVDD_RX LDO.

0x7B avdd_tx_ctrl

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7]	-	NA	
[6]	avdd_tx_disable	0x01	Disable AVDD_TX LDO.
[5]	avdd_tx_disable_pulldown	0x00	Enable/Disable pulldown resistor when AVDD_TX is disabled.
[4:0]	avdd_tx_trim	0x08	Control output voltage of AVDD_TX LDO.

0x7C avdd_testmode

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7:4]	avdd_rx_testmode	0x00	Enable AVDD_RX LDO testmode.
[3:0]	avdd_tx_testmode	0x00	Enable AVDD_TX LDO testmode.

0x7D ldo_status_1

Read-only 8-bit register.

Bit	Segment name	Default	Description
[7]	avdd_rx_anatestreq	0x00	anatestreq from avdd_rx LDO
[6]	avdd_rx_digtestbus	0x00	digtestbus from avdd_rx LDO
[5]	avdd_tx_anatestreq	0x00	anatestreq from avdd_tx LDO
[4]	avdd_tx_digtestbus	0x00	digtestbus from avdd_tx LDO
[3]	dvdd_rx_anatestreq	0x00	anatestreq from dvdd_rx LDO
[2]	dvdd_rx_digtestbus	0x00	digtestbus from dvdd_rx LDO
[1]	dvdd_tx_anatestreq	0x00	anatestreq from dvdd_tx LDO
[0]	dvdd_tx_digtestbus	0x00	digtestbus from dvdd_tx LDO

0x7E ldo_status_2

Read-only 8-bit register.

Bit	Segment name	Default	Description
[7]	dvdd_anatestreq	0x00	anatestreq from dvdd LDO
[6]	dvdd_digtestbus	0x00	digtestbus from dvdd LDO
[5]	apc_anatestreq	0x00	anatestreq from APC
[4]	apc_digtestbus	0x00	digtestbus from APC
[3]	dvdd_rx_power_good	0x00	power_good from dvdd_rx LDO
[2]	dvdd_tx_power_good	0x00	power_good from dvdd_tx LDO
[1]	avdd_rx_power_good	0x00	power_good from avdd_rx LDO
[0]	avdd_tx_power_good	0x00	power_good from avdd_tx LDO

0x7F spi_config_pif

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7:1]	-	NA	
[0]	spi_mode_pif	0x00	Set SPI mode: 0: Single-bit, 4-wire, SPI (clk, nss, miso, mosi) 1: QSPI (clk, nss, spi_io[3:0]) Note that this register is mirrored on the SPI bus as spi_mode. It can be written from both PIF and SPI, but can only be read from PIF side.

12. XIF register specification

Address	Name	Type
0x01	sampler_preset_msb	Register
0x02	sampler_preset_lsb	Register

Table 12.1. Register overview

0x01 sampler_preset_msb

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7]	sampler_preset_en	0x00	Enable override of sampler values
[6:4]	-	NA	
[3:0]	sampler_preset_val_msb	0x00	MSB of the sampler override value sampler_preset_val

0x02 sampler_preset_lsb

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7:0]	sampler_preset_val_lsb	0x00	LSB of the sampler override value sampler_preset_val

13. SPI register specification

Address	Name	Type
0x00	force_zero	Register
0x01	force_one	Register
0x02	chip_id_dig	Register
0x03	chip_id_sys	Register
0x04	debug	Register
0x05	radar_data_spi	Register
0x06	radar_data_spi_status	Register
0x0E	firmware_version_spi	Register
0x0F	to_cpu_write_data	Register
0x10	spi_mb_fifo_status	Register
0x11	from_cpu_read_data	Register
0x12	spi_mb_clear_status	Register
0x13	to_mem_write_data	Register
0x14	spi_mem_fifo_status	Register
0x15	from_mem_read_data	Register
0x16	spi_mem_clear_status	Register
0x17	mem_mode	Register
0x18	mem_first_addr_msb	Register
0x19	mem_first_addr_lsb	Register
0x1A	boot_from_otp_spi	Register
0x1D	spi_config	Register
0x7F	cpu_reset	Register

Table 13.1. Register overview

0x00 force_zero

Read-only 8-bit register.

Always returns 0x00.

Bit	Segment name	Default	Description
[7:0]	force_zero	0x00	

0x01 force_one

Read-only 8-bit register.

Always returns 0xFF.

Bit	Segment name	Default	Description
[7:0]	force_one	0xFF	

0x02 chip_id_dig

Read-only 8-bit register.

Chip ID byte 1.

Bit	Segment name	Default	Description
[7:0]	chip_id_dig	0x01	

0x03 chip_id_sys

Read-only 8-bit register.

Chip ID byte 2.

Bit	Segment name	Default	Description
[7:0]	chip_id_sys	0x02	

0x04 debug

Read-writable 8-bit register.

This debug register has no functional effect, but can be written and read to test communication between host and SPI slave.

Bit	Segment name	Default	Description
[7:0]	debug	0xAA	

0x05 radar_data_spi

Read-only 8-bit register.

Radar data, made available by fetch_radar_data_spi.

Bit	Segment name	Default	Description
[7:0]	radar_data_spi	0x00	

0x06 radar_data_spi_status

Read-only 8-bit register.

Bit	Segment name	Default	Description
[7]	Reserved	0x00	
[6]	spi_radar_data_spi_fifo_underflow	0x00	Indicates an underflow in radar_data_spi_fifo.
[5]	Reserved	0x00	
[4]	spi_radar_data_pif_fifo_underflow	0x00	Indicates an underflow in radar_data_pif_fifo.
[3:1]	-	NA	
[0]	radar_data_spi_fifo_empty	0x01	Status flag indicating whether radar data FIFO is empty

0x0E firmware_version_spi

Read-only 8-bit register.

A value set by firmware in the register firmware_version. Readable by SPI.

Bit	Segment name	Default	Description
[7:0]	firmware_version_spi	0x00	

0x0F to_cpu_write_data

Write-only 8-bit register.

A write to this register goes into the 8 byte deep mailbox FIFO, and an interrupt is raised for the CPU to handle the access.

Bit	Segment name	Default	Description
[7:0]	to_cpu_write_data	0x00	

0x10 spi_mb_fifo_status

Read-only 8-bit register.

Bit	Segment name	Default	Description
[7]	spi_to_cpu_fifo_overflow	0x00	Indicates an overflow in to_cpu_fifo (mailbox).
[6]	spi_to_cpu_fifo_underflow	0x00	Indicates an underflow in to_cpu_fifo (mailbox).
[5]	spi_from_cpu_fifo_overflow	0x00	Indicates an overflow in from_cpu_fifo (mailbox).
[4]	spi_from_cpu_fifo_underflow	0x00	Indicates an underflow in from_cpu_fifo (mailbox).
[3]	-	NA	
[2]	to_cpu_fifo_empty	0x01	Indicates that the to_cpu_fifo is empty.
[1]	from_cpu_data_valid	0x00	Set high when read data is available to the SPI from the CPU in the 8-byte deep mailbox FIFO. If the flag is 1, from_cpu_read_data is valid If the flag is 0, from_cpu_read_data is not valid.
[0]	to_cpu_fifo_full	0x00	Goes high when the to_cpu_write_data FIFO is full. To be completely safe, this bit should be checked before each write.

0x11 from_cpu_read_data

Read-only 8-bit register.

Read data from CPU via the mailbox FIFO. Valid only if from_cpu_data_valid is 1.

Bit	Segment name	Default	Description
[7:0]	from_cpu_read_data	0x00	

0x12 spi_mb_clear_status

Write-only 8-bit register.

Bit	Segment name	Default	Description
[7]	spi_clear_to_cpu_fifo_overflow	0x00	Write 1 to clear to_cpu_fifo_overflow
[6]	spi_clear_to_cpu_fifo_underflow	0x00	Write 1 to clear to_cpu_fifo_underflow
[5]	spi_clear_from_cpu_fifo_overflow	0x00	Write 1 to clear from_cpu_fifo_overflow
[4]	spi_clear_from_cpu_fifo_underflow	0x00	Write 1 to clear from_cpu_fifo_underflow
[3:0]	-	NA	

0x13 to_mem_write_data

Write-only 8-bit register.

Write to this FIFO when mem_programming_mode is 1 to program memory or OTP.

Bit	Segment name	Default	Description
[7:0]	to_mem_write_data	0x00	

0x14 spi_mem_fifo_status

Read-only 8-bit register.

Bit	Segment name	Default	Description
[7]	spi_to_mem_fifo_overflow	0x00	Indicates an overflow in to_mem_fifo (memory programming).
[6]	spi_to_mem_fifo_underflow	0x00	Indicates an underflow in to_mem_fifo (memory programming). Should not happen by design.
[5]	spi_from_mem_fifo_overflow	0x00	Indicates an overflow in from_mem_fifo (memory readback). Should not happen by design.
[4]	spi_from_mem_fifo_underflow	0x00	Indicates an underflow in from_mem_fifo (memory readback).
[3]	-	NA	
[2]	from_mem_data_valid	0x00	After enabling mem_readback_mode, this bit goes high when data is available to read.
[1]	to_mem_fifo_empty	0x01	This bit indicates whether the FIFO to_mem_fifo is empty.
[0]	to_mem_fifo_full	0x00	Indicates that the FIFO to_mem_fifo is full. It is illegal to write to to_mem_write_data when this bit is high. (FIFO will overflow, i.e. ignore write.)

0x15 from_mem_read_data

Read-only 8-bit register.

After enabling mem_readback_mode, data from memory is available to read from this register.

Bit	Segment name	Default	Description
[7:0]	from_mem_read_data	0x00	

0x16 spi_mem_clear_status

Write-only 8-bit register.

Bit	Segment name	Default	Description
[7]	spi_clear_to_mem_fifo_overflow	0x00	Write 1 to clear to_mem_fifo_overflow
[6]	spi_clear_to_mem_fifo_underflow	0x00	Write 1 to clear to_mem_fifo_underflow
[5]	spi_clear_from_mem_fifo_overflow	0x00	Write 1 to clear from_mem_fifo_overflow
[4]	spi_clear_from_mem_fifo_underflow	0x00	Write 1 to clear from_mem_fifo_underflow
[3:0]	-	NA	

0x17 mem_mode

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7:2]	-	NA	
[1]	mem_readback_mode	0x00	Set to 1 to enable readback of OTP or program memory. mem_first_addr should be programmed before setting this bit to 1. As long as this bit is 1, memory data can be read from from_mem_read_data when available. When reset to 0, there will be some words left in the from_mem_read_data FIFO that should be drained before next readout. Note that this FIFO is not cleared automatically. mem_programming_mode and mem_readback_mode is not allowed to be 1 simultaneously. The CPU is NOT held in reset when mem_readback_mode is high, but can be held in reset by setting cpu_reset if this is desired.
[0]	mem_programming_mode	0x00	Set to 1 to enable programming of OTP or program memory. mem_first_addr should be programmed before setting this bit to 1. Make sure to_mem_fifo_empty is 1 before clearing this bit. To target OTP, set mem_first_addr to 0xfe00. Otherwise, program memory is target. mem_programming_mode and mem_readback_mode is not allowed to be 1 simultaneously. The CPU is held in reset as long as mem_programming_mode is high.

0x18 mem_first_addr_msb

Read-writable 8-bit register.

MSB of mem_first_addr, the first address for programming or readback.

Programming will continue from this address as long as mem_programming_mode is high and there is data in the FIFO.

Readback will continue from this address as long as mem_readback_mode is high.

Bit	Segment name	Default	Description
[7:0]	mem_first_addr_msb	0x00	

0x19 mem_first_addr_lsb

Read-writable 8-bit register.

LSB of mem_first_addr, the first address for programming or readback.

Bit	Segment name	Default	Description
[7:0]	mem_first_addr_lsb	0x00	

0x1A boot_from_otp_spi

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7:1]	-	NA	
[0]	boot_from_otp_spi	0x01	By default, the hardware enables a small boot ROM that jumps to the OTP. Write 0 to this register to boot directly from program memory (SRAM).

0x1D spi_config

Write-only 8-bit register.

Bit	Segment name	Default	Description
[7:1]	-	NA	
[0]	spi_mode	0x00	Set SPI mode: 0: Single-bit, 4-wire, SPI (clk, nss, miso, mosi) 1: QSPI (clk, nss, spi_io[3:0]) Note that this register is mirrored on the PIF bus as spi_mode_pif. It can be written from both PIF and SPI, but can only be read from PIF side.

0x7F cpu_reset

Read-writable 8-bit register.

Bit	Segment name	Default	Description
[7:1]	-	NA	
[0]	cpu_reset	0x00	Write a 1 to this bit to hold the CPU in reset. Note that the CPU is also held in reset if mem_programming_mode is enabled.

14. Disclaimer

The information provided in this document represents Novelda's knowledge and beliefs as of the time of writing. Novelda AS reserves the right to make corrections, modifications, enhancements, improvements and other changes to its products and services at any time, and to discontinue any product or service without prior notice. Customers are encouraged to obtain the latest information before placing orders, and should verify that the information is up-to-date and complete. Information is supplied upon the condition that the persons receiving same will make their own determination as to its suitability for their purposes prior to use. In no event will Novelda be responsible for damages of any nature whatsoever resulting from the use of or reliance upon information.

All products are sold subject to Novelda's terms and conditions of sale supplied at the time of order acknowledgement. No representations or warranties, either express or implied, of merchantability, fitness for a particular purpose, that the products to which the information refers may be used without infringing the intellectual property rights of others, or of any other nature are made hereunder with respect to the information or the product to which the information refers. In no case shall the information be considered a part of our terms and conditions of sale.

Document History

Rev.	Release date	Change description
F - Preliminary	03-March-2020	Editorial updates throughout the document. Added recommended chip operation, including decoupling capacitors and other recommended passive components. Added X4 layout example.
E - Preliminary	23-Nov-2017	Added technical drawings for X4 WLCSP package. Added recommended PCB layout for X4 WLCSP package.
D - Preliminary	21-Aug-2017	Update description of Transceiver, Clock Management and Power Management. Update Electrical Characteristics.
C - Preliminary	29-Dec-2016	Add register maps. Update description of System Controller, SPI and Transceiver. Update front page. Small adjustment to Table 2.2
B - Preliminary	25-Oct-2016	Add specification for 1.8V IO to Chapter 1. Clock source specifications moved to Chapter 1. Cosmetic updates.
A - Preliminary	03-Jun-2016	Initial release of X4 datasheet based on early sampling prototype. Specifications are subject to change.