ALPHA250 User Guide

Getting started

Warnings

The ALPHA250 power supply must be turned off before connecting or disconnecting:

- the micro-SD card
- peripherals on the expansion connector

Power up the board

First, insert the micro-SD card into the micro-SD slot. Then connect the 12V jack of the power supply. The power good green LED (PWGD) and the FPGA done orange LED (DONE) indicate the system has correctly started.

Communicating with the board

LAN

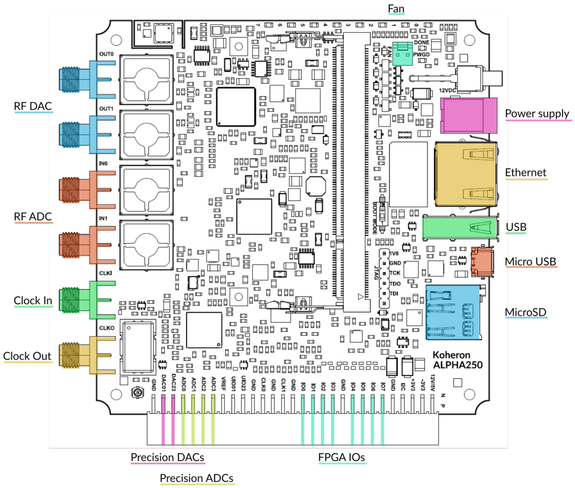
The ethernet port is the main communication interface with the ALPHA250. It can be connected to a local network via a router or directly to a computer. The last 8 bits of the IP address are displayed on the 8 user LEDs.

Serial interface

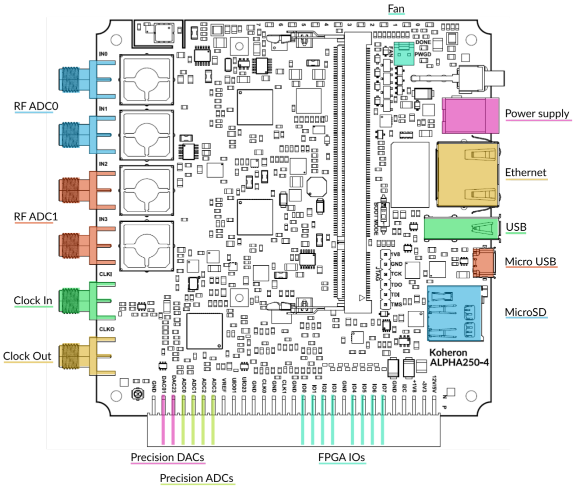
The serial UART debugging interface can be accessed by the micro USB connector. The required steps are described here.

Important note: Power up the board first and wait for the board to boot before connecting the serial cable.

Connectors



ALPHA250 connectors



ALPHA250-4 connectors

12 V external power supply

The external supply connector is a jack with 1.95 mm center pin and 6 mm outer diameter. Only 12 V must be supplied on this connector. Running the ALPHA250 requires at least 1 A. More current may be required depending on the load on the expansion connector. Maximum current is 3 A (protected by an electronic fuse).

USB 2.0

This is a USB 2.0 host connector. It provides up to 1 A current at 5 V (shared with the 5 V supply of the expansion connector). The power and data pins are ESD protected.

Micro USB 2.0

Connects to the `UART0` PS core via a FTDI device. It is used as a debugging serial interface. The power and data pins are ESD protected.

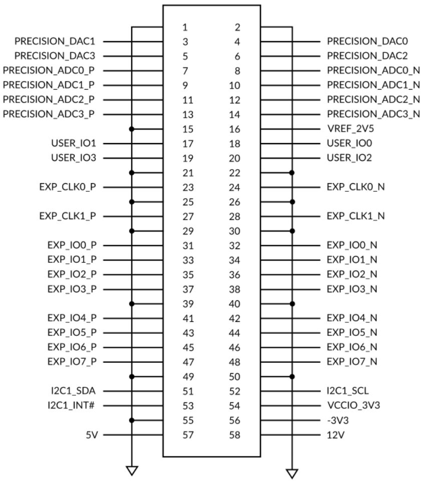
Gigabit ethernet

The ALPHA250 is capable of 10/100/1000 Mbit Ethernet.

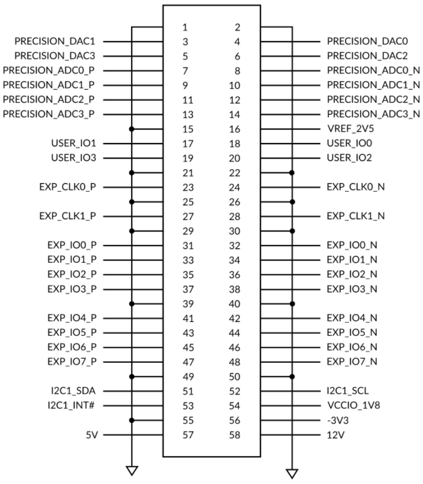
MicroSD card

The micro SD card is connected to the `SD0` PS core via a level-shifter. The SD card I/Os are ESD protected.

Expansion connector


1 2
PRECISION_DAC1 3 4 PRECISION_DAC0
PRECISION_DAC3 5 6 PRECISION_DAC2
PRECISION_ADC0_P 7 8 PRECISION_ADC0_N
PRECISION_ADC1_P 9 10 PRECISION_ADC1_N
PRECISION_ADC2_P 11 12 PRECISION_ADC2_N
PRECISION_ADC3_P 13 14 PRECISION_ADC3_N
15 16 VREF_2V5
USER_IO1 17 18 USER_IO0
USER_IO3 19 20 USER_IO2
21 22
EXP_CLK0_P 23 24 EXP_CLK0_N
25 26
EXP_CLK1_P 27 28 EXP_CLK1_N
29 30
EXP_IO0_P 31 32 EXP_IO0_N
EXP_IO1_P 33 34 EXP_IO1_N
EXP_IO2_P 35 36 EXP_IO2_N
EXP_IO3_P 37 38 EXP_IO3_N
39 40
EXP_IO4_P 41 42 EXP_IO4_N
EXP_IO5_P 43 44 EXP_IO5_N
EXP_IO6_P 45 46 EXP_IO6_N
EXP_IO7_P 47 48 EXP_IO7_N
49 50
I2C1_SDA 51 52 I2C1_SCL
I2C1_INT# 53 54 VCCIO_3V3
55 56 -3V3
5V 57 58 12V

ALPHA250 expansion connector


1 2
PRECISION_DAC1 3 4 PRECISION_DAC0
PRECISION_DAC3 5 6 PRECISION_DAC2
PRECISION_ADC0_P 7 8 PRECISION_ADC0_N
PRECISION_ADC1_P 9 10 PRECISION_ADC1_N
PRECISION_ADC2_P 11 12 PRECISION_ADC2_N
PRECISION_ADC3_P 13 14 PRECISION_ADC3_N
15 16 VREF_2V5
USER_IO1 17 18 USER_IO0
USER_IO3 19 20 USER_IO2
21 22
EXP_CLK0_P 23 24 EXP_CLK0_N
25 26
EXP_CLK1_P 27 28 EXP_CLK1_N
29 30
EXP_IO0_P 31 32 EXP_IO0_N
EXP_IO1_P 33 34 EXP_IO1_N
EXP_IO2_P 35 36 EXP_IO2_N
EXP_IO3_P 37 38 EXP_IO3_N
39 40
EXP_IO4_P 41 42 EXP_IO4_N
EXP_IO5_P 43 44 EXP_IO5_N
EXP_IO6_P 45 46 EXP_IO6_N
EXP_IO7_P 47 48 EXP_IO7_N
49 50
I2C1_SDA 51 52 I2C1_SCL
I2C1_INT# 53 54 VCCIO_1V8
55 56 -3V3
5V 57 58 12V

ALPHA250-4 expansion connector

It contains:

- Power supplies. 12 V from external supply. 5 V up to 1 A (shared with USB 2.0 connector). +3.3 V up to 800 mA, sequenced with I/Os supply. -3.3 V up to 500 mA.
- A dedicated I2C bus `I2C1` with interrupt.
- 16 single ended or 8 differential I/Os `EXP_IOx`. They are connected to the FPGA Bank 35. ALPHA250 voltage level is 3.3 V, ALPHA250-4 voltage level is 1.8 V. All I/Os are ESD protected. **Warning**: These pins are connected directly to the Zynq SoC and must be driven from `VCCIO_3V3` for ALPHA250, `VCCIO_1V8` for ALPHA250-4, both for voltage compliance and power sequencing. Applying non compliant voltages on these pins may result in SoC failure.
- Two LVDS clocks from the clocking subsystem `EXP_CLK0` and `EXP_CLK1`.
- 4 user IOs from the GPIO expander `USER_IOx`. Voltage level is 3.3 V. They can be configured as inputs or outputs, open-drain or pull-up. These I/Os have 22 Ω series protection resistors. They are ESD protected.

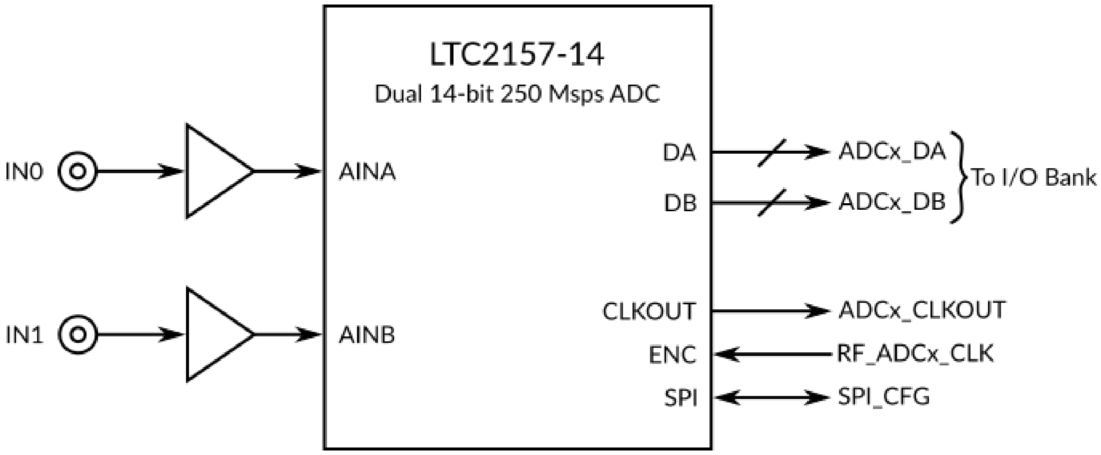
- A 2.5 V voltage reference. This reference acts as a ratiometric tracking with the reference used for all the data converters (RF ADC, RF DAC, precision ADC, precision DAC). It delivers up to 150 mA (Care must be taken with the voltage drop in track resistance at large currents).
- 4 precision ADC inputs. Differential inputs to the precision 24-bit ADC.
- 4 precision DAC outputs. Outputs from the 16-bit precision DAC.

Subsystems

RF front end

RF ADC

The RF Analog-to-Digital Converter has 2 acquisition channels with 14-bit resolution and 250 Msps maximum sampling rate (Linear Technologies LTC2157-14). It has two inputs labeled IN0 and IN1 on the SMA connectors. The inputs are DC coupled and 50 Ω terminated. The optimum DC offset is reached when the input is driven from a 50 Ω output impedance source. The peak-peak input range is 1 V_{pp} (between -500 and 500 mV). The inputs are protected by a transient voltage suppressor clamping over-voltages beyond ±8 V.



ALPHA250 RF ADC interface

The ALPHA250 encoding clock of the ADC is provided by RF_ADC_CLK. The ALPHA250-4 encoding clocks of the ADC0 and ADC1 are provided by RF_ADC0_CLK and RF_ADC1_CLK respectively.

The ADC is interfaced to the I/O Bank 34 of the FPGA for ALPHA250. ADC0 is interfaced to the Bank 35 and ADC1 to the Bank 34 of the FPGA for ALPHA250-4. It consists of 14 LVDS pairs operating in double data rate. The maximum transfer rate per LVDS pair is thus 500 Msps. The transfer protocol is described in the LTC2157-14 datasheet. A clock synchronous with data ADC_CLKOUT is connected to the Bank 34 for ALPHA250. Clocks synchronous with data ADC0_CLKOUT is connected to the Bank 35 and ADC1_CLKOUT is connected to the Bank 34 for ALPHA250-4.

The RF ADC is configured by the configuration SPI bus. The source code of the matching C++ drivers are on Github (ALPHA250, ALPHA250-4).

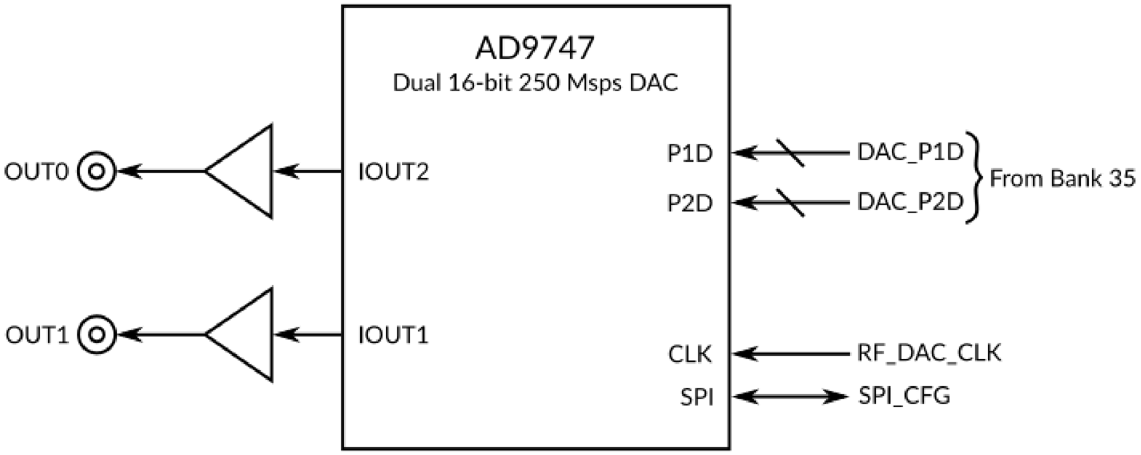
RF DAC

The section below is for ALPHA250 only.

The RF Digital-to-Analog Converter has 2 outputs with 16-bit resolution and 250 Msps maximum sampling rate (Analog Devices AD9747). The outputs are labeled OUT0 and OUT1 on the SMA connectors. Output impedance is 50 Ω. The outputs are protected by a transient voltage suppressor clamping over-voltages beyond ±8 V.

The output range is 1.5 V_{pp} maximum in a 50 Ω load. It can be adjusted using the DAC gain on the configuration

SPI bus. In the default configuration, the DAC outputs 1 V_{pp} in a 50 Ω load.



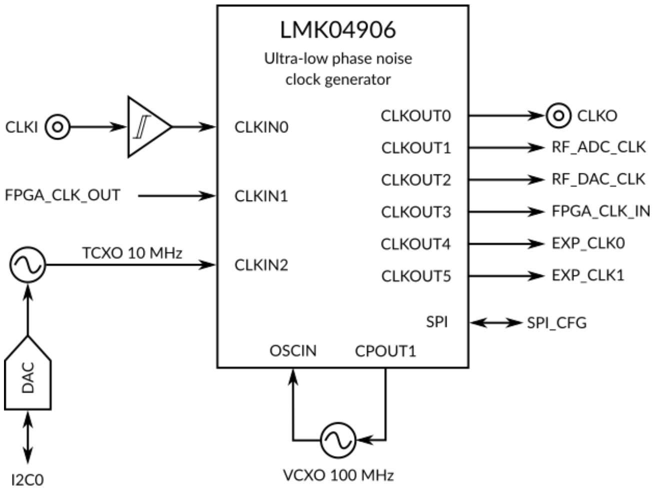
ALPHA250 RF DAC interface

The sampling clock of the DAC is provided by the RF_DAC_CLK of the clocking system. The input data lines are interfaced to the I/O Bank 35 of the FPGA. It consists of 32 single-ended lines at 3.3 V.

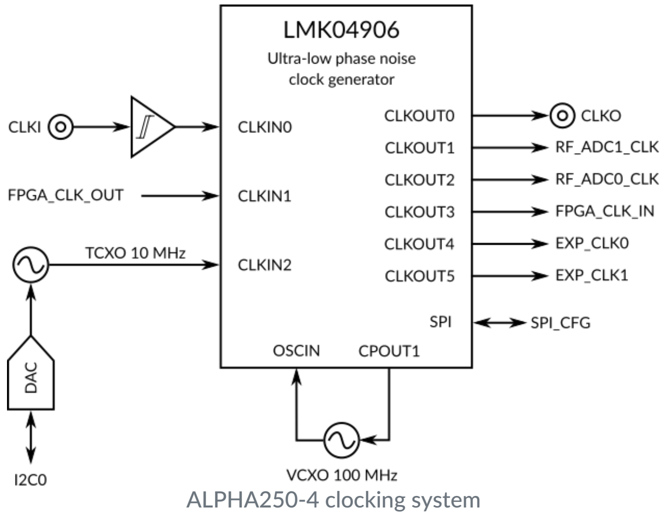
The RF DAC is configured by the configuration SPI bus. The source code of the corresponding C++ driver is on Github.

Clocking

The clocking system is organized around the ultra-low phase noise clock generator (Texas Instruments LMK04906). A dual PLL setup is used. The first loop locks an ultra-low phase noise VCXO (ABLNO-V-100.000MHz) onto a reference clock. It serves as a phase-noise cleaner for the reference clock. The second loop locks the LMK04906 internal VCO (2.37 to 2.6 GHz) onto the VCXO. A set of clock dividers allows to produce up to six clocks at desired frequencies.



ALPHA250 clocking system



ALPHA250-4 clocking system

The clock generator is designed to accept 10 MHz reference clocks.

The reference clock sources are:

- The CLKI SMA input on the board. It is a 50 Ω impedance input that accepts an AC voltage of up to 2.5 V_{pp} (10 dBm). It also supports up to 5 V_{DC}. An onboard precision high-speed comparator (890 Mbps) provides effective clock recovery. It is ESD protected.
- The FPGA_CLK_OUT signal to discipline the system on a clock provided by the FPGA.
- The onboard 10 MHz temperature compensated voltage controlled crystal oscillator (TCVCXO). It has a tight stability (±280 ppb over the industrial temperature range -40 to +85 °C). Its aging is ±1 ppm per year maximum. A DAC controlled from the I2C0 bus can be used to precisely tune its frequency.

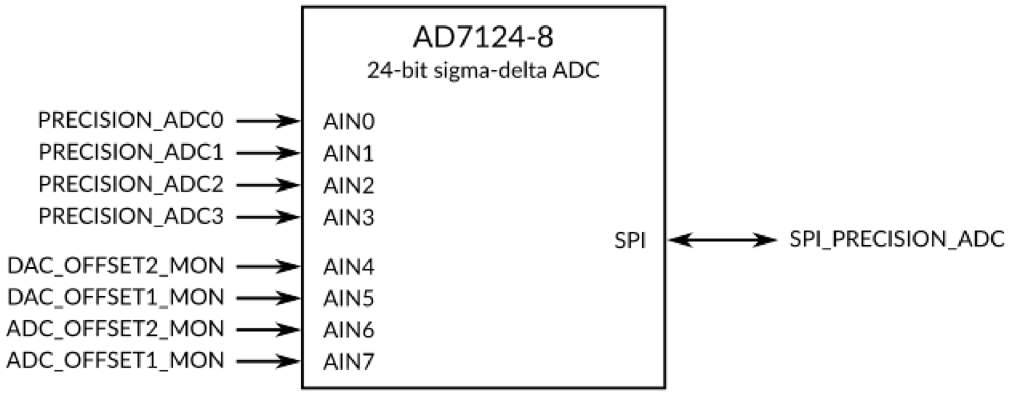
The clock generator produces the following clocks:

- An LVCMOS clock available on the board CLKO SMA output. This output is ESD protected.
- RF ADC sampling clock(s) RF_ADCx_CLK distributed as an LVDS signal.
- For ALPHA250, RF DAC sampling clock RF_DAC_CLK distributed as an LVDS signal.
- FPGA_CLK_IN is an LVDS input clock on the FPGA.
- EXP_CLK0 and EXP_CLK1 are LVDS clocks available on the expansion connector.

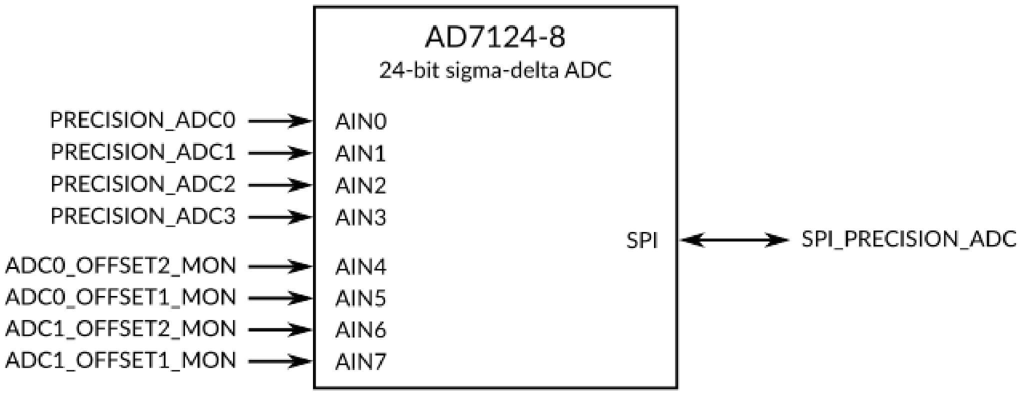
The clock generator is configured by the configuration SPI bus via the ClockGenerator driver available on Github (ALPHA250, ALPHA250-4).

Precision ADC

The precision ADC is a 8-channel, 24-bit sigma-delta ADC with programmable gain amplifier (Analog Devices AD7124-8). The inputs differential pairs can be used to sense either floating or ground referenced signal. They also facilitate Kelvin sense connections. Differential input voltage range is ±1.25 V. The voltage on any precision ADC input pin must be between -1.8 V and +1.8 V.



ALPHA250 precision ADC interface



ALPHA250-4 precision ADC interface

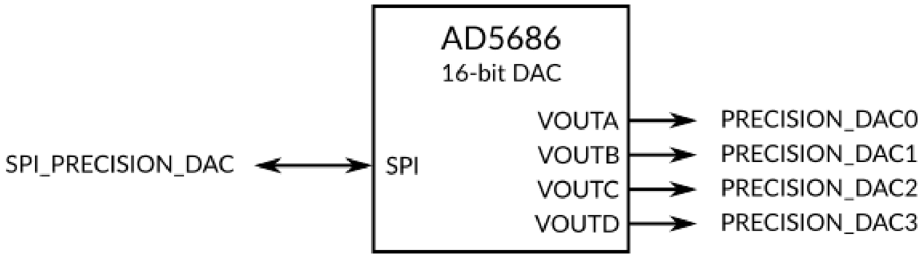
The first four channels are available on the expansion connector. The next four channels monitor the offsets of the RF front end.

Communication with the precision ADC is done through a dedicated SPI bus. The precision ADC data can be retrieved using the PrecisionAdc driver.

Precision DAC

The precision DAC is a 4-channel, 16-bit DAC (Analog Devices AD5686).

The output voltage ranges from 0 to 2.5 V. It includes an output buffer that can deliver up to 20 mA per channel. The outputs are ESD protected.

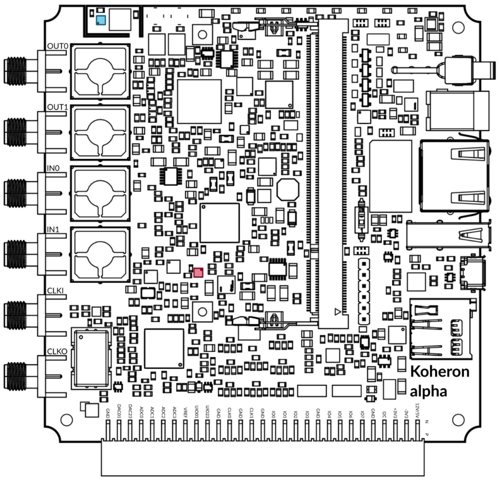


ALPHA250 precision DAC interface

The four output channels are available on the expansion connector. Communication with the precision DAC is done through a dedicated SPI bus via the PrecisionDac driver.

Temperature sensors

The ALPHA250 has two high-accuracy temperature sensors (TMP116) with an accuracy of ±0.2 °C over -10 °C to +85 °C. One sensor is placed near the voltage reference (**T0** highlighted in blue) to allow temperature compensation in high precision measurements. The other one is placed between the clock generator and the RF ADC which is close to the hottest point on the board (**T1** highlighted in red).



ALPHA250 temperature sensors. Voltage reference in blue, board in red

The Zynq temperature **T2** is also monitored using the XADC on the FPGA. The TemperatureSensor driver allows to retrieve the 3 above temperatures. For reliable operation, make sure that **T1** < 70 °C and **T2** < 85 °C.

Power monitors

The ALPHA250 includes two power monitors (Texas Instruments INA230). Two rails are monitored: the external 12 V power supply and the clocking subsystem supply. In both cases, the current shunt resistor is 10 mΩ. Both power monitors are accessible on the `I2C0` bus via the PowerMonitor driver.

EEPROM

The ALPHA250 has a 64-kbit EEPROM (Microchip 24AA64T-I/MC). It is accessible on the `I2C0` bus via the Eeprom driver.

The EEPROM is divided into two parts. The lower addresses are used by Koheron to store identification and calibration data. The higher addresses (above 0x1000) are for user applications. The EEPROM map addressing is given in the table below.

Description	Offset	Range
Identifications	0x000	0x100
Precision DAC	0x100	0x100
RF ADC channel 0	0x200	0x100
RF ADC channel 1	0x300	0x100
Clock generator	0x400	0x100
RF DAC channel 0	0x500	0x100
RF DAC channel 1	0x600	0x100
Precision ADC	0x700	0x100
	0x800	0x100
	0x900	0x100
User application	0x1000	0x100

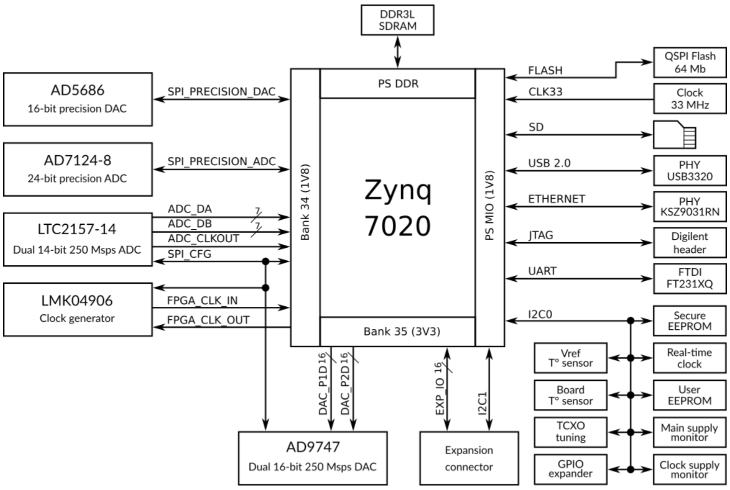
ALPHA250 EEPROM map addressing

Description	Offset	Range
Identifications	0x000	0x100
Precision DAC	0x100	0x100
RF ADC0 channel 0	0x200	0x100
RF ADC0 channel 1	0x300	0x100
Clock generator	0x400	0x100
	0x500	0x100
	0x600	0x100
Precision ADC	0x700	0x100
RF ADC1 Channel 0	0x800	0x100
RF ADC1 Channel 1	0x900	0x100
User application	0x1000	0x100

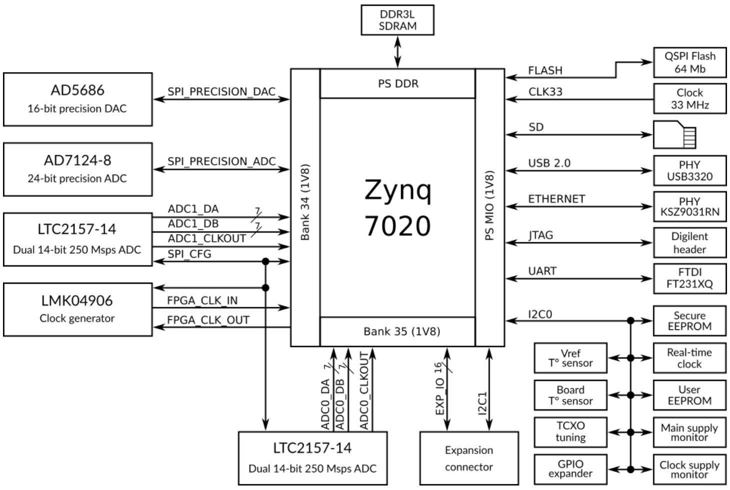
ALPHA250-4 EEPROM map addressing

Zynq I/Os

The Zynq XC7Z020-2CLG400I has 2 I/O banks for the programmable logic (Banks 34 and 35) with 48 IOs each. One bank (Bank 0) is dedicated to the processing system with a multiplexed I/O (MIO) interface. The set of peripherals and interface buses is depicted below.



ALPHA250 Zynq peripherals and communication buses



ALPHA250-4 Zynq peripherals and communication buses

I/O constraints are defined in the ports.xdc file on Github (ALPHA250, ALPHA250-4).

RF ADC parallel bus

The RF ADC is interfaced to the Bank 34 by a LVDS parallel bus for ALPHA250. RF ADC0 is interfaced to Bank 35 and RF ADC1 to Bank 34) by a LVDS parallel bus for ALPHA250-4. The data for each ADC channel are transferred in double data rate on a 7 line sub-bus. The RF ADC also provides a clock synchronous with the output data ADC_CLKOUT for ALPHA250. The RF ADCs also provide clocks synchronous with the output data ADC0_CLKOUT and ADC1_CLKOUT for ALPHA250-4. The configuration is performed via the configuration SPI bus.

```
# RF ADC
set_property IOSTANDARD DIFF_HSTL_I_18 [get_ports adc_*]

set_property PACKAGE_PIN P19 [get_ports adc_clk_in_clk_n]
set_property PACKAGE_PIN N18 [get_ports adc_clk_in_clk_p]

# Channel 0
set_property PACKAGE_PIN U17 [get_ports {adc_0_n[0]}]
set_property PACKAGE_PIN T16 [get_ports {adc_0_p[0]}]
set_property PACKAGE_PIN Y19 [get_ports {adc_0_n[1]}]
set_property PACKAGE_PIN Y18 [get_ports {adc_0_p[1]}]
set_property PACKAGE_PIN P16 [get_ports {adc_0_n[2]}]
set_property PACKAGE_PIN P15 [get_ports {adc_0_p[2]}]
set_property PACKAGE_PIN W19 [get_ports {adc_0_n[3]}]
set_property PACKAGE_PIN W18 [get_ports {adc_0_p[3]}]
set_property PACKAGE_PIN P18 [get_ports {adc_0_n[4]}]
set_property PACKAGE_PIN N17 [get_ports {adc_0_p[4]}]
set_property PACKAGE_PIN W20 [get_ports {adc_0_n[5]}]
set_property PACKAGE_PIN V20 [get_ports {adc_0_p[5]}]
set_property PACKAGE_PIN U20 [get_ports {adc_0_n[6]}]
set_property PACKAGE_PIN T20 [get_ports {adc_0_p[6]}]

# Channel 1
set_property PACKAGE_PIN W13 [get_ports {adc_1_n[0]}]
set_property PACKAGE_PIN V12 [get_ports {adc_1_p[0]}]
set_property PACKAGE_PIN Y14 [get_ports {adc_1_n[1]}]
set_property PACKAGE_PIN W14 [get_ports {adc_1_p[1]}]
set_property PACKAGE_PIN P20 [get_ports {adc_1_n[2]}]
set_property PACKAGE_PIN N20 [get_ports {adc_1_p[2]}]
set_property PACKAGE_PIN R14 [get_ports {adc_1_n[3]}]
set_property PACKAGE_PIN P14 [get_ports {adc_1_p[3]}]
set_property PACKAGE_PIN W15 [get_ports {adc_1_n[4]}]
set_property PACKAGE_PIN V15 [get_ports {adc_1_p[4]}]
set_property PACKAGE_PIN T15 [get_ports {adc_1_n[5]}]
set_property PACKAGE_PIN T14 [get_ports {adc_1_p[5]}]
set_property PACKAGE_PIN Y17 [get_ports {adc_1_n[6]}]
set_property PACKAGE_PIN Y16 [get_ports {adc_1_p[6]}]
```

RF DAC parallel bus

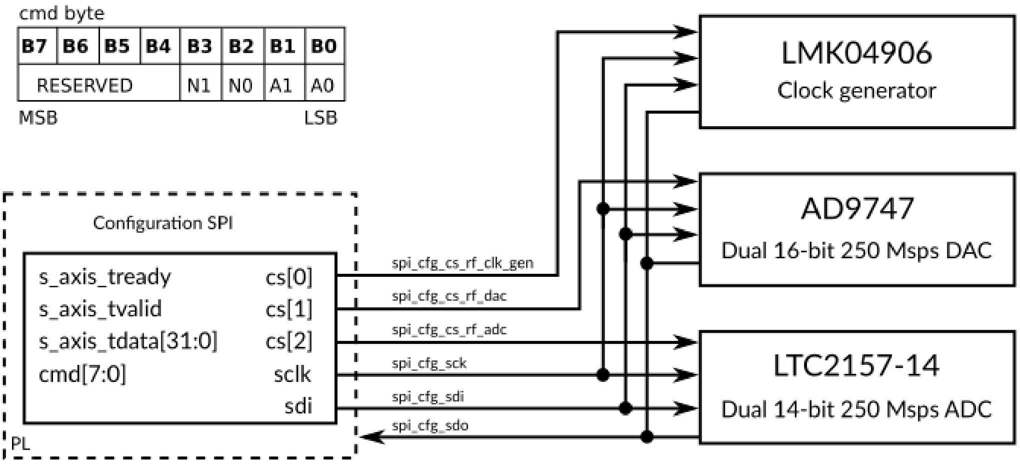
The section below is for ALPHA250 only.

The RF DAC is interfaced to the Bank 35 by a LVCMOS 3V3 parallel bus. The data for each channel are transferred on a 16 line sub-bus. The configuration is performed via the configuration SPI bus.

```
# RF DAC
set_property IOSTANDARD LVCMOS33 [get_ports dac_*]
set_property DRIVE 8 [get_ports dac_*]

set_property IOSTANDARD LVCMOS33 [get_ports dac_*]
set_property DRIVE 8 [get_ports dac_*]

# Channel 0
set_property PACKAGE_PIN D18 [get_ports {dac_0[0]}]
set_property PACKAGE_PIN E17 [get_ports {dac_0[1]}]
set_property PACKAGE_PIN E19 [get_ports {dac_0[2]}]
set_property PACKAGE_PIN E18 [get_ports {dac_0[3]}]
set_property PACKAGE_PIN A20 [get_ports {dac_0[4]}]
set_property PACKAGE_PIN B19 [get_ports {dac_0[5]}]
set_property PACKAGE_PIN F17 [get_ports {dac_0[6]}]
set_property PACKAGE_PIN F16 [get_ports {dac_0[7]}]
set_property PACKAGE_PIN B20 [get_ports {dac_0[8]}]
set_property PACKAGE_PIN C20 [get_ports {dac_0[9]}]
set_property PACKAGE_PIN L17 [get_ports {dac_0[10]}]
set_property PACKAGE_PIN L16 [get_ports {dac_0[11]}]
set_property PACKAGE_PIN D20 [get_ports {dac_0[12]}]
set_property PACKAGE_PIN D19 [get_ports {dac_0[13]}]
set_property PACKAGE_PIN G18 [get_ports {dac_0[14]}]
set_property PACKAGE_PIN G17 [get_ports {dac_0[15]}]

# Channel 1
set_property PACKAGE_PIN F20 [get_ports {dac_1[0]}]
set_property PACKAGE_PIN F19 [get_ports {dac_1[1]}]
set_property PACKAGE_PIN J16 [get_ports {dac_1[2]}]
set_property PACKAGE_PIN K16 [get_ports {dac_1[3]}]
set_property PACKAGE_PIN G20 [get_ports {dac_1[4]}]
set_property PACKAGE_PIN G19 [get_ports {dac_1[5]}]
set_property PACKAGE_PIN K18 [get_ports {dac_1[6]}]
set_property PACKAGE_PIN K17 [get_ports {dac_1[7]}]
set_property PACKAGE_PIN H20 [get_ports {dac_1[8]}]
set_property PACKAGE_PIN J20 [get_ports {dac_1[9]}]
set_property PACKAGE_PIN M18 [get_ports {dac_1[10]}]
set_property PACKAGE_PIN M17 [get_ports {dac_1[11]}]
set_property PACKAGE_PIN H18 [get_ports {dac_1[12]}]
set_property PACKAGE_PIN J18 [get_ports {dac_1[13]}]
set_property PACKAGE_PIN G15 [get_ports {dac_1[14]}]
set_property PACKAGE_PIN H15 [get_ports {dac_1[15]}]
```

Configuration SPI bus

A shared SPI bus is dedicated to the configuration of the RF ADC, the RF DAC and the clock generator. In the reference design, a HDL core is used for the communication on this bus. The interface is described below.

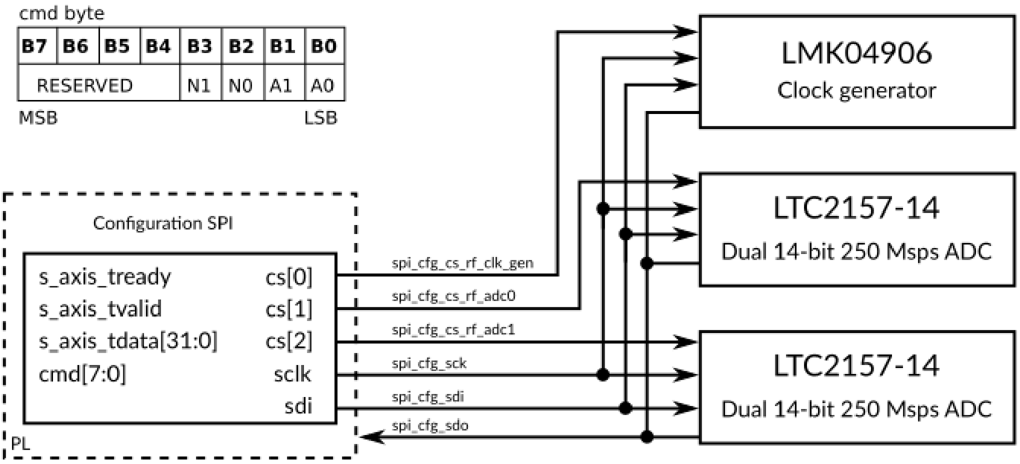


cmd byte

B7	B6	B5	B4	B3	B2	B1	B0
RESERVED				N1	N0	A1	A0

MSB LSB

ALPHA250 configuration SPI bus



cmd byte

B7	B6	B5	B4	B3	B2	B1	B0
RESERVED				N1	N0	A1	A0

MSB LSB

ALPHA250-4 configuration SPI bus

Constraint file

The configuration SPI bus pins are connected to Bank 34 with 1.8 V LVCMOS signals.

```
# Configuration SPI (Bank 34)
set_property IOSTANDARD LVCMOS18 [get_ports spi_cfg_*]

set_property PACKAGE_PIN R17 [get_ports spi_cfg_sdo]
set_property PACKAGE_PIN R16 [get_ports spi_cfg_sdi]
set_property PACKAGE_PIN W16 [get_ports spi_cfg_sck]

set_property PACKAGE_PIN V16 [get_ports spi_cfg_cs_rf_adc]
set_property PACKAGE_PIN U12 [get_ports spi_cfg_cs_rf_dac]
set_property PACKAGE_PIN T12 [get_ports spi_cfg_cs_clk_gen]
```

Transfer core

The core is write only. It does not read back configurations from the chips. The `s_axis_tready` signal can be used to determine when the core finishes sending a message and is ready to send a new one.

The core can send 1, 2, 3 or 4 bytes of data. The number of bytes to transfer is specified using the bits `B2` and `B3` of the `cmd` byte according to the table below.

N1	N0	Description
0	0	Transfer one byte
0	1	Transfer two bytes
1	0	Transfer three bytes
1	1	Transfer four bytes

Configuration SPI byte transfer count

The chip select address is specified using the bits `B0` and `B1` of the `cmd` byte according to the table below.

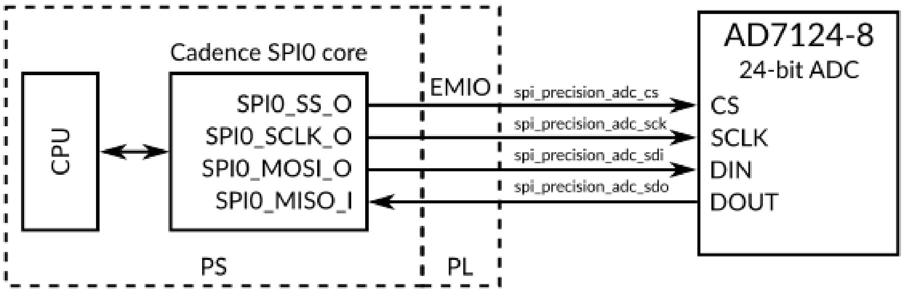
A1	A0	Description
0	0	`CS` = 0
0	1	`CS` = 1
1	0	`CS` = 2

Configuration SPI chip select address

The transferred data must be wired to the `s_axis_tdata` pin. The transfer is triggered on the `s_axis_tvalid` pin falling edge. The core is controlled via the SpiConfig driver.

Precision ADC SPI bus

A dedicated SPI is used for the communication with the precision ADC. The bus is connected to PL I/Os on bank 34.



ALPHA250 precision ADC SPI bus

Constraint file

The precision ADC SPI pins are connected to the Bank 34 with 1.8 V LVCMOS signals.

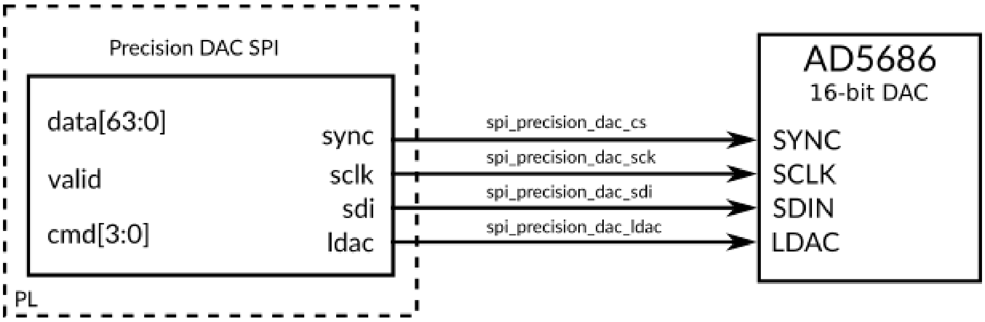
```
# Precision ADC (Bank 34)
set_property IOSTANDARD LVCMOS18 [get_ports spi_precision_adc_*]

set_property PACKAGE_PIN U13 [get_ports spi_precision_adc_cs]
set_property PACKAGE_PIN V13 [get_ports spi_precision_adc_sck]
set_property PACKAGE_PIN T11 [get_ports spi_precision_adc_sdi]
set_property PACKAGE_PIN T10 [get_ports spi_precision_adc_sdo]
```

Data transfer

In the reference design, the precision ADC SPI bus is connected to the SPI0 PS core. This is done using the EMIO interface which allows to connect PL signals to the MIO interface of the PS. It can be controlled using the PrecisionAdc driver.

Precision DAC SPI bus

Data is transferred to the precision DAC using a SPI bus that can be clocked up to 50 MHz. In addition, a latched pin LDAC pin is used to update the 4 channel outputs synchronously. In the reference design, a dedicated HDL core is used. The interface is shown below.



ALPHA250 precision DAC SPI bus

Constraint file

The precision DAC SPI pins are connected to the Bank 34 with 1.8 V LVCMOS signals.

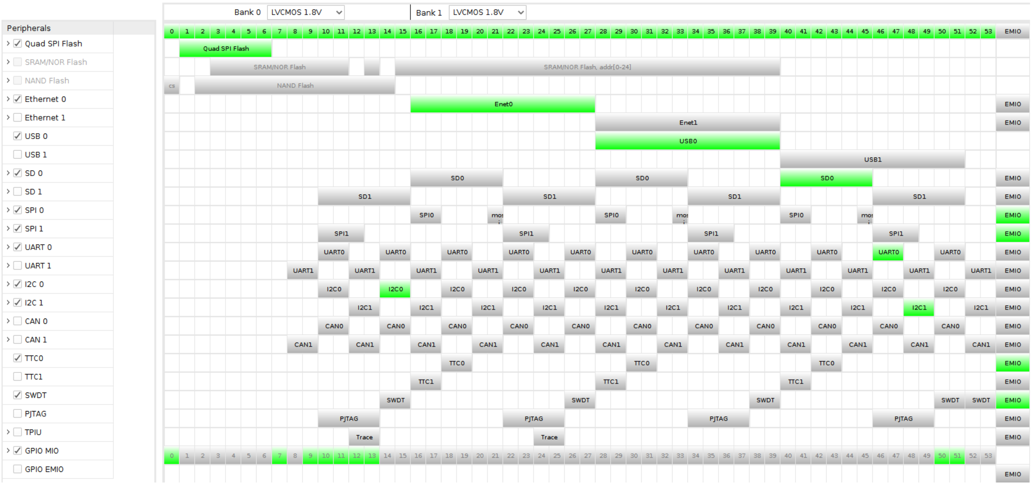
```
# Precision DAC (Bank 34)
set_property IOSTANDARD LVCMOS18 [get_ports spi_precision_dac_*]

set_property PACKAGE_PIN V17 [get_ports spi_precision_dac_cs]
set_property PACKAGE_PIN V18 [get_ports spi_precision_dac_sck]
set_property PACKAGE_PIN T17 [get_ports spi_precision_dac_sdi]
set_property PACKAGE_PIN R18 [get_ports spi_precision_dac_ldac]
```

Transfer core

While the valid is high, the core updates the DAC channels with the values on pin data. The 64 bits of the data pin contain the concatenation of the 4 x 16 bits values to be set on the 4 channels. Channel 0 being on the 16 least significant bits, followed by channels 1, 2 and 3. After sending the data for the 4 channels, the core latches ldac. The data will be synchronously updated if cmd = 1. If cmd = 3, the output is updated as new values arrive. The core is written in Verilog. It is controlled with the PrecisionDac driver.

PS cores

The processing system contains hard cores (by opposition with the soft cores that can be deployed on the PL). The PS cores are interfaced with MIO pins on Bank 0. The logic level is LVCMOS 1.8 V. MIO configuration can be found in the board configuration file.



Bank 0 LVCMOS 1.8V

Bank 1 LVCMOS 1.8V

Peripherals

- ☑ Quad SPI Flash
- □ SRAM/NOR Flash
- □ NAND Flash
- ☑ Ethernet 0
- □ Ethernet 1
- ☑ USB 0
- □ USB 1
- ☑ SD 0
- □ SD 1
- ☑ SPI 0
- ☑ SPI 1
- ☑ UART 0
- □ UART 1
- ☑ I2C 0
- ☑ I2C 1
- □ CAN 0
- □ CAN 1
- ☑ TTC0
- □ TTC1
- ☑ SWDT
- □ PJTAG
- □ TPIU
- ☑ GPIO MIO
- □ GPIO EMIO

ALPHA250 Zynq peripheral I/O pins in Xilinx Vivado

The constraints file for the PS is:

```
set_property CFGBVS GND [current_design]
set_property CONFIG_VOLTAGE 1.8 [current_design]
```

I2C0

This bus is used on the ALPHA250 internally and is not accessible from the expansion connector. The I2C0 bus addressing is:

- `1100100` / `1011100`: Secure EEPROM
- `1010001`: RTC registers
- `1010100`: User EEPROM
- `0100000`: GPIO expander
- `1000001`: Main power supply monitor
- `1000101`: Clocking subsystem supply monitor
- `1001000`: Voltage reference temperature sensor
- `1001001`: Board temperature sensor
- `0101111`: TCXO control voltage

I2C1

The I2C1 bus is for the expansion connector. A 3.3 V level shifter with 2.2 kΩ pull-ups provides the interface with the Bank 0. The `I2C1` core can be replaced with a CAN core `CAN0` with proper PS configuration.

ENET0

The ethernet peripheral is interfaced with the `ENET0` MAC core.

USB0

The USB 2 connector is interfaced with the `USB0` core.

SD0

The SD card is interfaced with the `SD0` core.

UART0

The serial port debugging USB interface connects to the `UART0` core.