

# adc

## creating ip block

Project Summary x IP Catalog x

Cores | Interfaces

Q

≡

⬇

⚙

🔗

🔗

⚙

🔗

⚙

⚙

Search:  (2 matches)

Name	AXI4	Status	License	VLNV
▼ Vivado Repository				
▼ FPGA Features and Design				
▼ XADC				
🔦 XADC Wizard	AXI4, AXI4-Stream	Production	Included	xilinx.com:ip:xadc_wiz:3.3

Details

Path:

/tools/Xilinx/Vivado/2022.2/data

Number of IPs:

688

Number of interfaces:

434

click on "xadc wizard". this description only covers usage of the boolean board's potentiometer.

Customize IP

### XADC Wizard (3.3)

Documentation IP Location Switch to Defaults

☐ Show disabled ports

user\_temp\_alarm\_out

vccint\_alarm\_out

vccaux\_alarm\_out

ot\_out

channel\_out[4:0]

eoc\_out

alarm\_out

eos\_out

busy\_out

+ s\_drp

+ Vp\_Vn

- dclk\_in

- reset\_in

Component Name xadc\_wiz\_0

Basic
ADC Setup
Alarms
Single Channel
Summary

**Interface Options**

☐ AXI4Lite ☒ **DRP** ☐ None

**Startup Channel Selection**

☐ Simultaneous Selection

☐ Independent ADC

☒ **Single Channel**

☐ Channel Sequencer

**AXI4STREAM Options**

☐ Enable AXI4Stream

FIFO Depth 7 [7 - 1020]

**Control/Status Ports**

☒ reset\_in ☐ Temp Bus ☐ JTAG Arbiter

**Event Mode Trigger**

☒ convst\_in ☐ convstclk\_in

**Timing Mode**

☒ **Continuous Mode** ☐ Event Mode

**DRP Timing Options**

☒ Enable DCLK

DCLK Frequency(MHz) 100 [8.0 - 250.0]

ADC Conversion Rate(KSPS) 1000 [39.0 - 1000.0]

Acquisition Time (CLK) 4

Clock divider value = 4

ADC Clock Frequency(MHz) = 25.00

Actual Conversion Rate(KSPS) = 961.54

**Analog Sim File Options**

Sim File Selection Default

Analog Stimulus File design

Sim File Location /

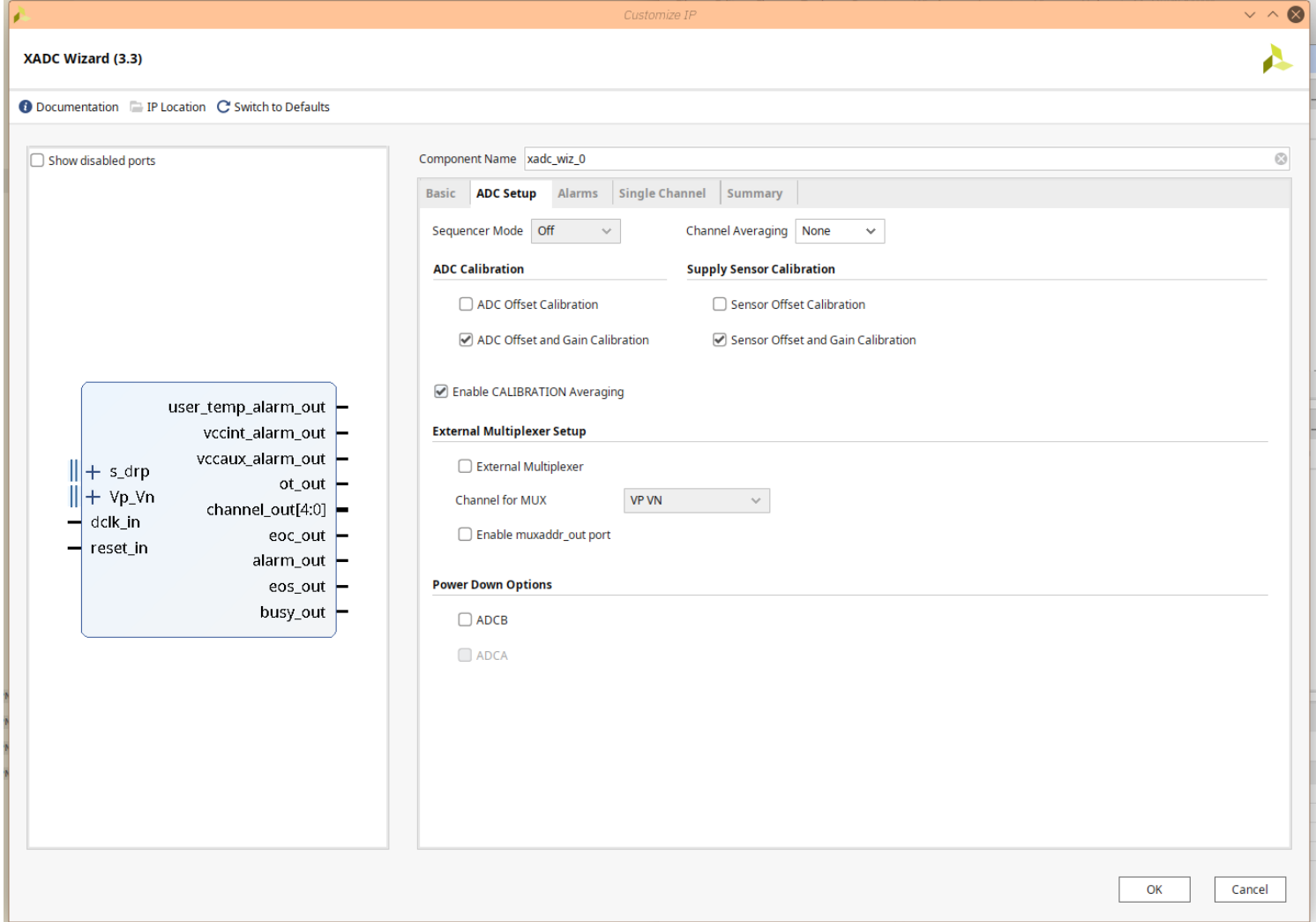
Waveform Type CONSTANT

Frequency (KHz) 1.0 [0.1 - 480.77]

Number of Wave 1 [1 - 1000]

OK
Cancel

- select "DRP" as the interface: necessary to communicate with the adc and read the relevant registers
- only single channel is necessary
- ignore AXI4Stream
- we only care about reset\_in
- i only tried continuous timing mode, event-based mode appears to work much the same, except that eoc\_out and eos\_out may also need to be monitored.
- DCLK can be set to 100MHz --- the XADC has a built-in clock divider anyways, so we can use the boolean board's onboard clock.

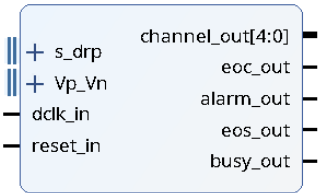


- adc calibration does what it says on the tin, supply sensor calibration makes sure the reference voltages are ok
- the adc also includes a multiplexer for switching between the 16 extra channels, we don't care about this. set channel to VP\_VN to make sure we're really reading the right inputs
- don't care about power down options: these turn off one of two ADCs internal to the XADC, we can ignore these for now

XADC Wizard (3.3)

Documentation IP Location Switch to Defaults

Show disabled ports



Component Name xadc\_pot

BasicADC SetupAlarmsSingle ChannelSummary

☐ Over Temperature Alarm (°C)

Trigger125.0[-40.0 - 125.0]

Reset70.0[-40.0 - 125.0]

☐ User Temperature Alarm (°C)

Trigger85.0[-40.0 - 125.0]

Reset60.0[-40.0 - 125.0]

☐ VCCINT Alarm (Volts)

Lower0.97[0.0 - 1.05]

Upper1.03[0.0 - 1.05]

☐ VCCAUX Alarm (Volts)

Lower1.75[0.0 - 1.89]

Upper1.89[0.0 - 1.89]

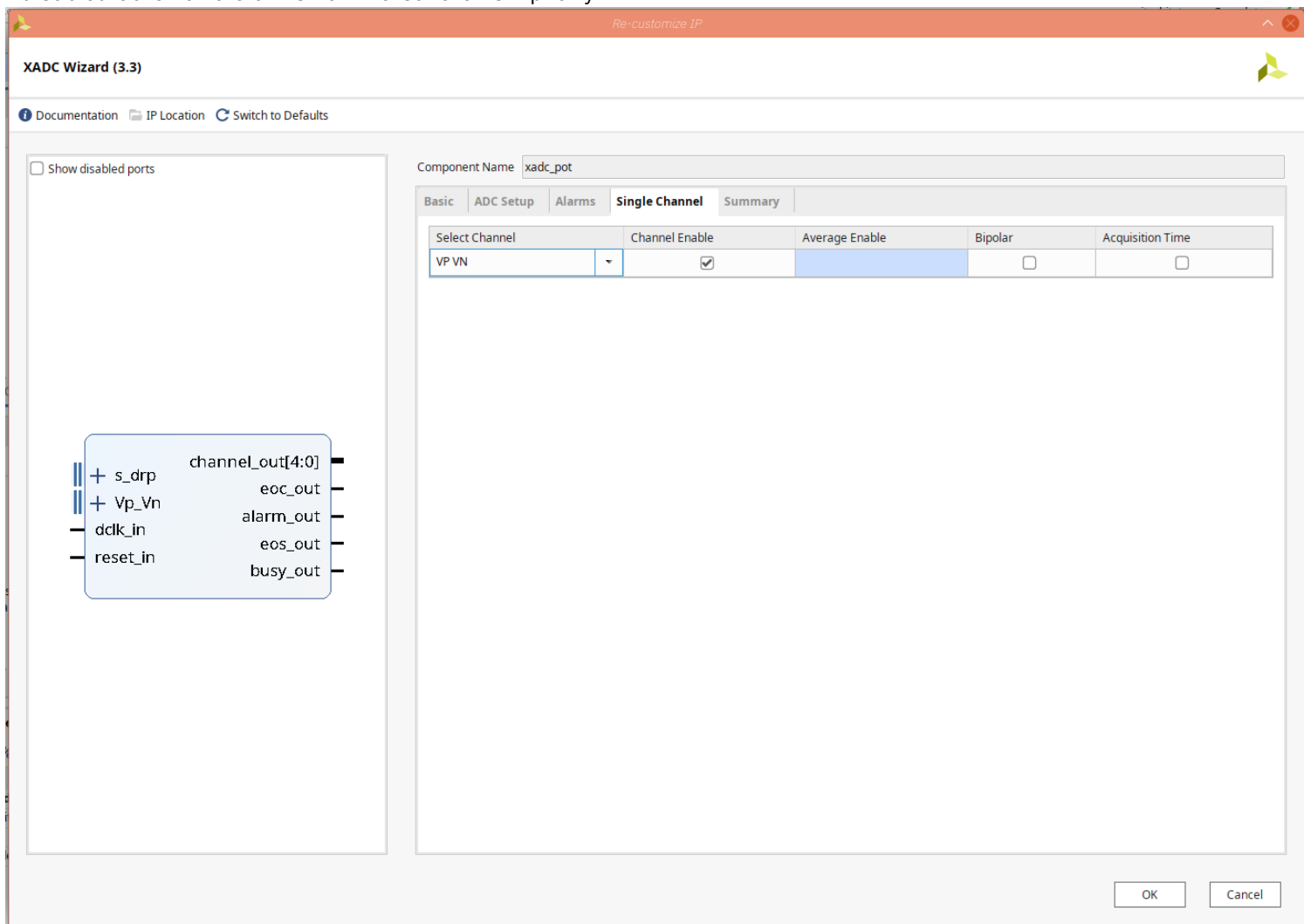
☐ VCCBRAM Alarm (Volts)

Lower0.95[0.0 - 1.05]

Upper1.05[0.0 - 1.05]

OKCancel

- i disabled automatic alarms for the sake of simplicity



- select the correct channel, non-bipolar, and we don't care about acquisition time

this generates an `.xci` "file", which you can then directly use in another verilog module.

## verilog module

the FPGA interfaces with the ADC with the DRP (dynamic reconfiguration port). broadly, the interface works like this (roughly cribbed from page 63 in the XADC user guide):

- set the address of the register to read from/write to, in `daddr_in`
  - if writing, then `di` must be set as well
- set whether to read or write via `dwe` (set to high to write, set to low to read)
- send enable signal via `den`, which should be high only for one DCLK period
- wait for `drdy` signal, which indicates that data has been read / written
- if reading data, read data from `do_out`.

in this case, the addresses containing VP / VN data are at `0x03` (see page 29 in the XADC user guide) for the locations of other possible parameters), so `daddr_in` should be set to `'h03` to read from the relevant registers.

example code is given below.

```
module pot_reader(input clk, reset, output reg [15:0] r);
    wire drdy;
    wire [15:0] val;
    reg den;
    // the following block was copied and modified from the verilog file embedded in the .xci generated
    in the previous step
    xadc_pot x0(
        .daddr_in('h03), // Address bus for the dynamic reconfiguration port
```

```

.dclk_in(clk), // Clock input for the dynamic reconfiguration port
.den_in(den), // Enable Signal for the dynamic reconfiguration port
.dwe_in(1'b0), // Write Enable for the dynamic reconfiguration port
.reset_in(reset), // Reset signal for the System Monitor control logic
.do_out(val), // Output data bus for dynamic reconfiguration port
.drdy_out(drdy) // Data ready signal for the dynamic reconfiguration port
); // replace xadc_pot with the name of your .xci module

always @(posedge clk) begin
    den <= ~den; // clock enable at half the rate as the adc's clock
end
always @(posedge drdy) begin // if data is ready, read to output
    r <= val;
end

endmodule

// simple example usage
module main(input mclk, [3:0]btn, output [15:0]led);
    pot_reader p(mclk, btn[0],led); // outputs to LEDs
endmodule

```

## **sources:**

<https://docs.xilinx.com/viewer/book-attachment/q0eib0vlzXa1isUAFuFz0Q/2mNWBzeqk05~0C4MjwWAKw> (detailed XADC user guide)

<https://docs.xilinx.com/v/u/en-US/pg091-xadc-wiz> (description of XADC IP block)

[https://aydos.de/digital-logic\\_/example-adc-reading.html](https://aydos.de/digital-logic_/example-adc-reading.html) (ended up not really being too helpful, but may be a curiosity)