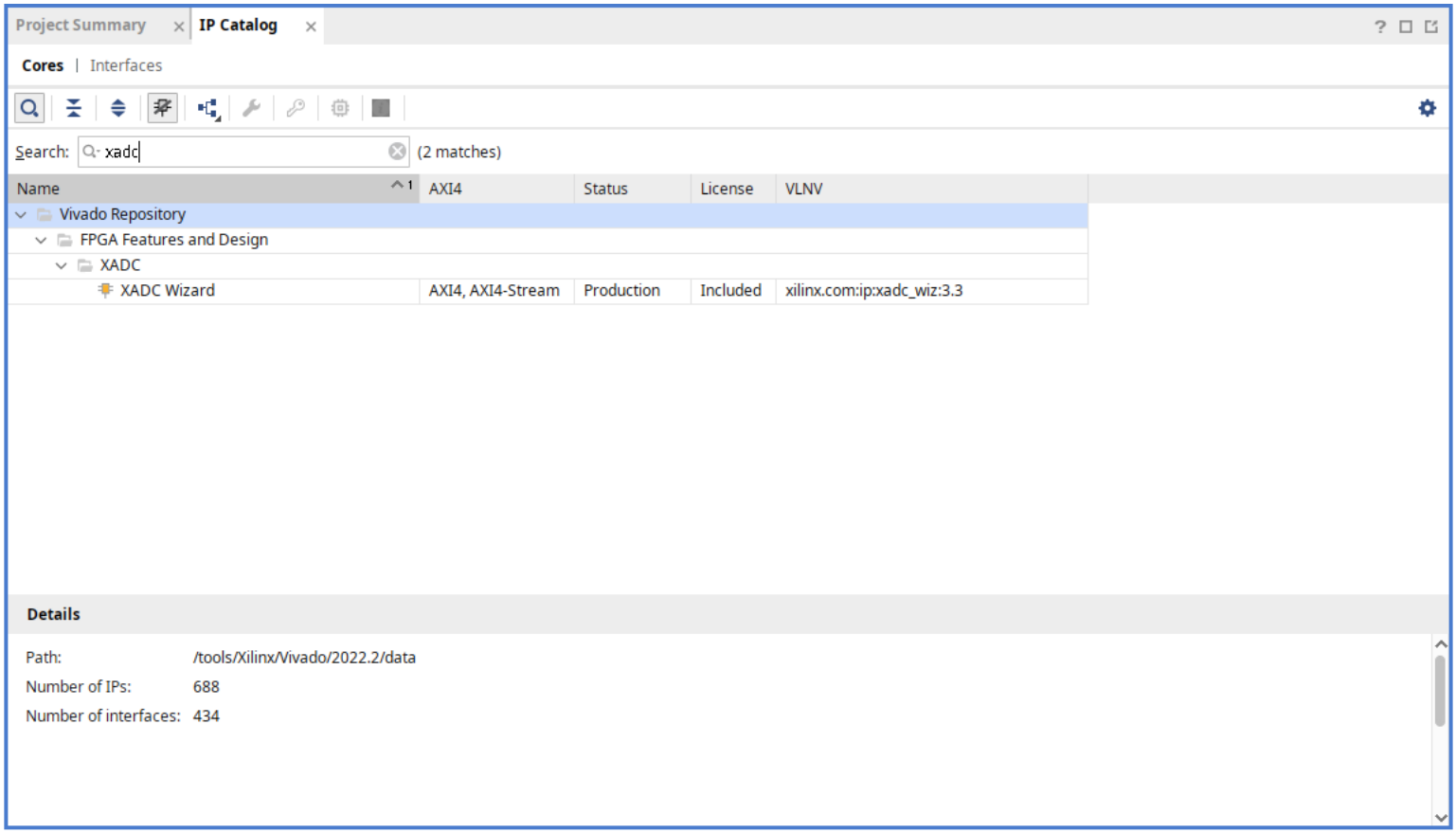
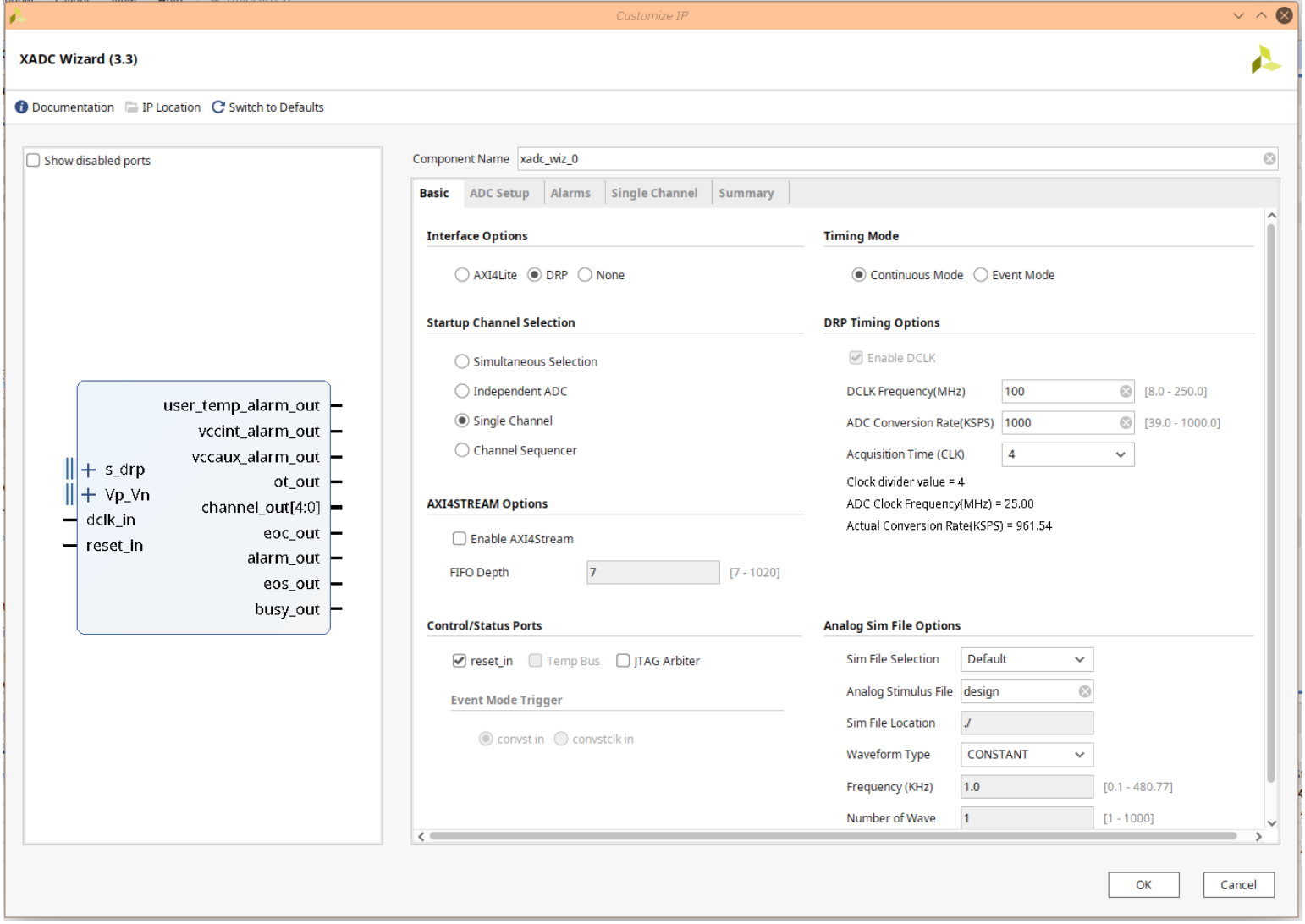
adc

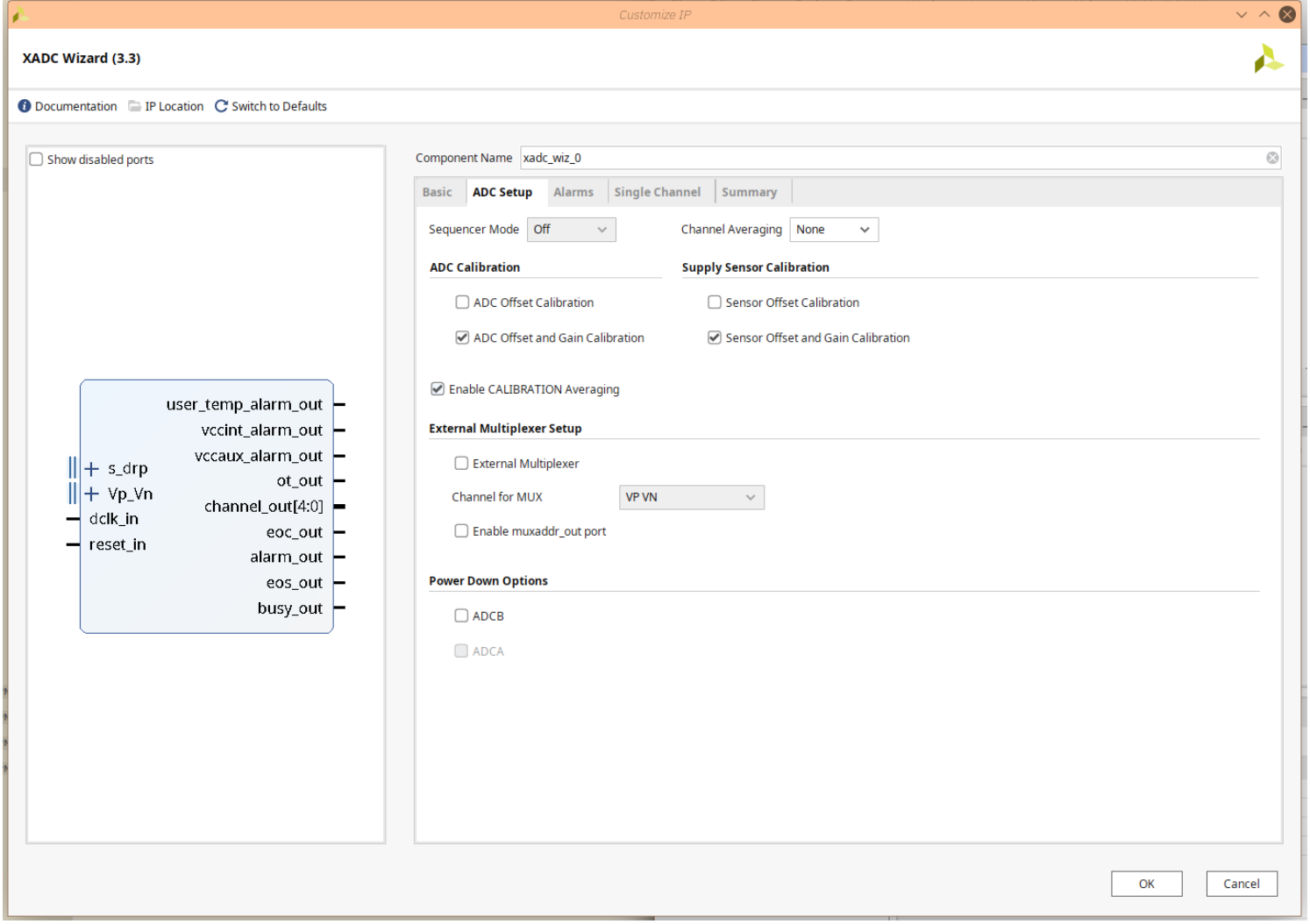
creating ip block



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



- 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.



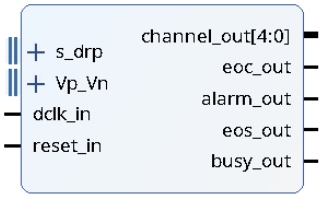
- adc calibration does what it says on the tin, supply sensor calibration makes sure the reference voltages are ok
- the adc also includes a multiplexer for switching between the 16 extra channels, we don't care about this. set channel to VP VN to make sure we're really reading the right inputs
- don't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

Re-customize IP

XADC Wizard (3.3)

Documentation | IP Location | Switch to Defaults

Show disabled ports



Component Name: xadc_pot

Basic | ADC Setup | **Alarms** | Single Channel | Summary

Over Temperature Alarm (°C)

Trigger 125.0 [-40.0 - 125.0]

Reset 70.0 [-40.0 - 125.0]

User Temperature Alarm (°C)

Trigger 85.0 [-40.0 - 125.0]

Reset 60.0 [-40.0 - 125.0]

VCCINT Alarm (Volts)

Lower 0.97 [0.0 - 1.05]

Upper 1.03 [0.0 - 1.05]

VCCAUX Alarm (Volts)

Lower 1.75 [0.0 - 1.89]

Upper 1.89 [0.0 - 1.89]

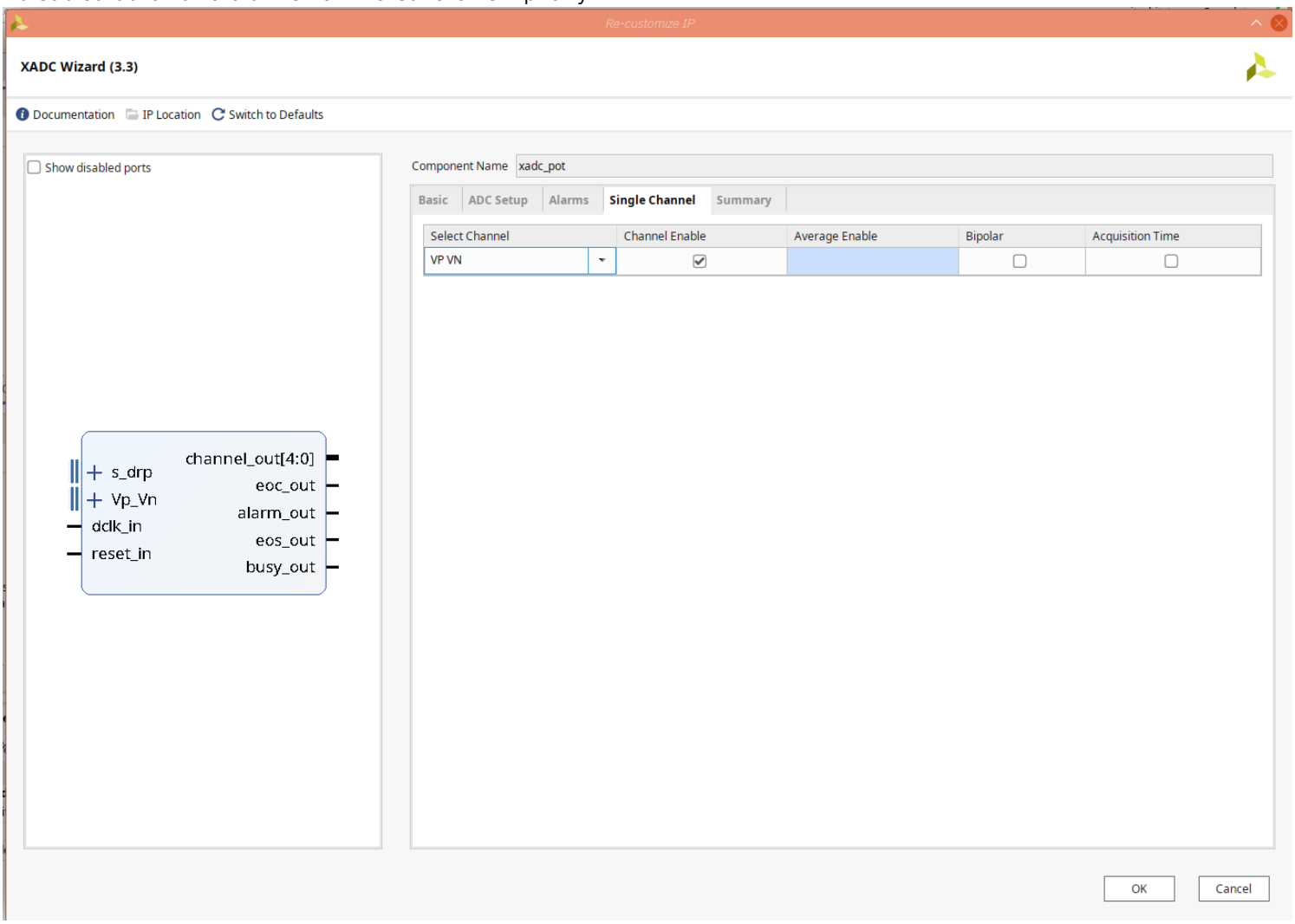
VCCBRAM Alarm (Volts)

Lower 0.95 [0.0 - 1.05]

Upper 1.05 [0.0 - 1.05]

OK | Cancel

- 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/qOeib0vlzXa1isUAfuFzOQ/2mNWBzeqkO5~OC4MjwWAKw (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 (ended up not really being too helpful, but may be a curiousity)