XADC IMPLEMENTATION AND DEBUG VERIFICATION ON XILINX 7 SERIES FPGA

In digital signal processing, ADCs are an integral part of the digital system. The universe will always have signals and they are all analog in nature. They will always need processing and there will always be new applications, new mathematics and new implementation technologies. And all these evolving applications and technologies work in the digital domain. Hence, to convert those analog signals into digital and process them, ADCs will always be required.

1. Xilinx Analog to Digital Converter (XADC): Background Theory

The XADC is the basic building block that enables analog mixed signal (AMS) functionality which is new to 7 series FPGAs. By combining high quality analog blocks with the flexibility of programmable logic, it is possible to craft customized analog interfaces for a wide range of applications.

The XADC includes a dual 12-bit, 1 Mega sample per second (MSPS) ADC and on-chip sensors. The ADCs provide a general-purpose, high-precision analog interface for a range of applications.

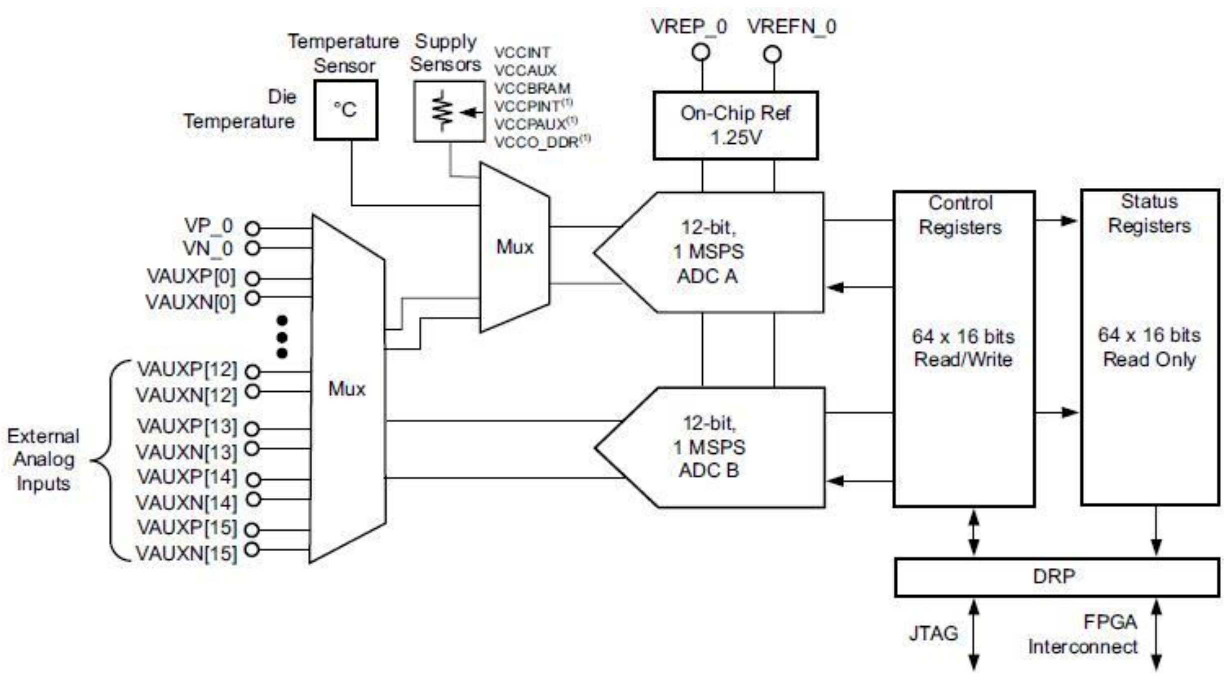


Figure 1: XADC Block Diagram

Figure 1 shows the block diagram of XADC. The dual ADCs support a range of operating modes, for example, externally triggered and simultaneous sampling on both ADCs and various analog input signal types, for example, unipolar and differential. The ADCs can access up to 17 external analog input channels.

The XADC also includes several on-chip sensors that support measurement of the on-chip power supply voltages and die temperature. The ADC conversion data is stored in dedicated registers called status registers. These registers are accessible through the FPGA interconnect using a 16-bit synchronous read and write port called the dynamic reconfiguration port (DRP). ADC conversion data is also accessible through the JTAG TAP, either before (pre-configuration) or after configuration. For JTAG TAP, users are not required to instantiate the XADC because it is a dedicated interface that uses the existing FPGA JTAG infrastructure. If the XADC is not instantiated in a design, the device operates in a predefined mode (called default mode) that monitors on-chip temperature and supply voltages.

XADC operation is user defined by writing to the control registers using either the DRP or JTAG interface. It is also possible to initialize these register contents when the XADC is instantiated in a design using the block attributes.

2. Operating Modes: Background Theory

The XADC can be configured to work on the following two modes by writing to the configuration registers.

2.1 Unipolar Mode

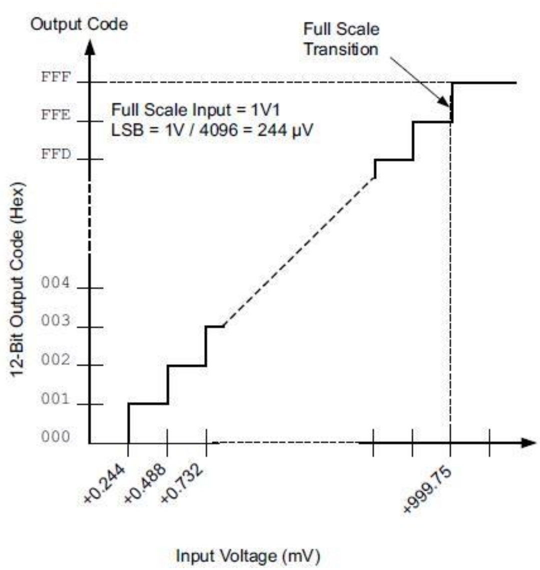


Figure 2: Unipolar Transfer Function

Figure 2 shows the 12-bit unipolar transfer function for the ADCs. The nominal analog input range to the ADCs is 0V to 1V in this mode. The ADC produces a zero code (`000h`) when 0V is present on the ADC input and a full scale code of all 1s (`FFFh`) when 1V is present on the input. The ADC output coding in unipolar mode is straight binary. The designed code transitions occur at successive integer LSB values such as one LSB, two LSBs, and three LSBs, etc. The LSB size in volts is equal to $1V/2^{12}$ or 1V/4096 = 244 µV. The analog input channels are differential in nature and require both the positive (V_P) and negative (V_N) inputs of the differential input to be driven.

2.2 Bipolar Mode

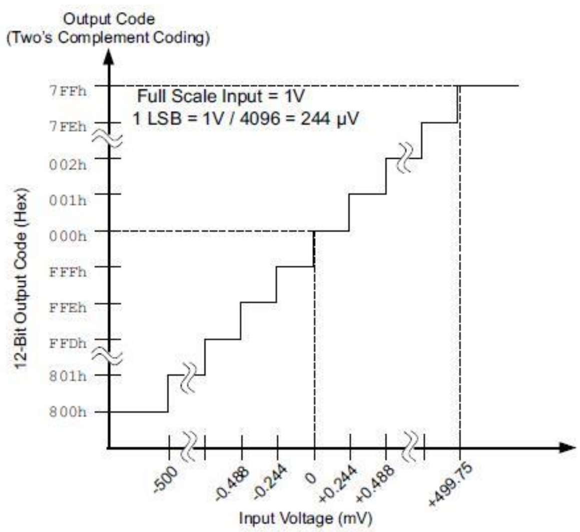


Figure 3: Bipolar Transfer Function

When the external analog input channels of the ADCs are configured as bipolar, they can accommodate true differential and bipolar analog signal. When dealing with differential signal types, it is useful to have both sign and magnitude information about the analog input signal. Figure 3 shows the ideal transfer function for bipolar mode operation. The output coding of the ADC in bipolar mode is two's complement and is intended to indicate the sign of the input signal on V_P relative to V_N. The designed code transitions occur at successive integer LSB values, that is, one LSB, two LSBs, three LSBs, etc. The LSB size in volts is equal to $1V/2_{12}$ or 1V/4096 = 244 µV.

3. Block Design: Top Level Block Design

The following design is generated and implemented on ZYBO using Vivado 2018.1 on a Windows machine. The process is similar for other versions of Vivado and different operating systems.

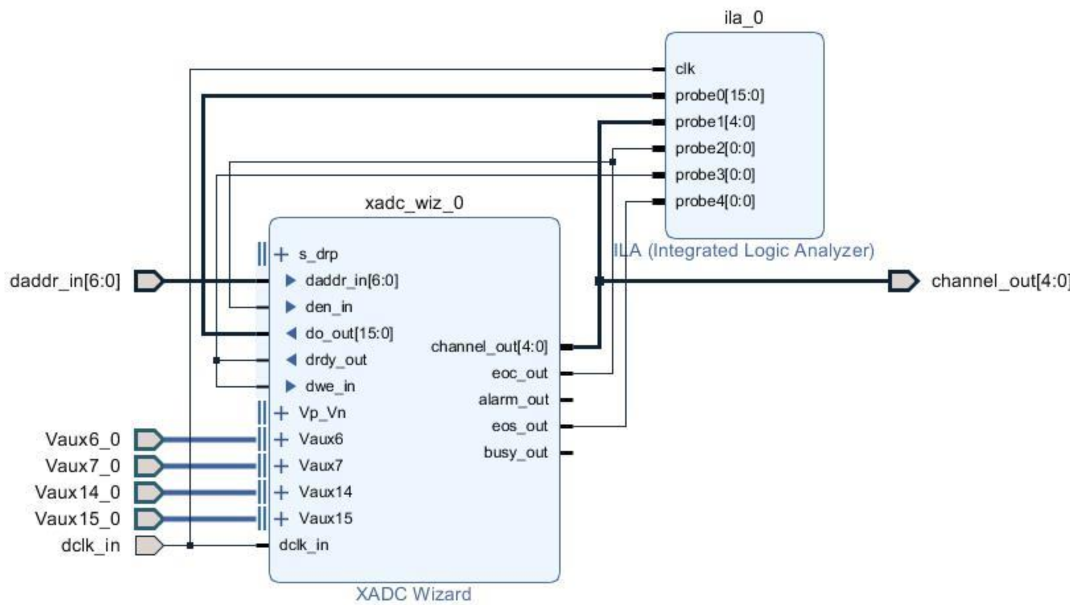


Figure 4: Block Design

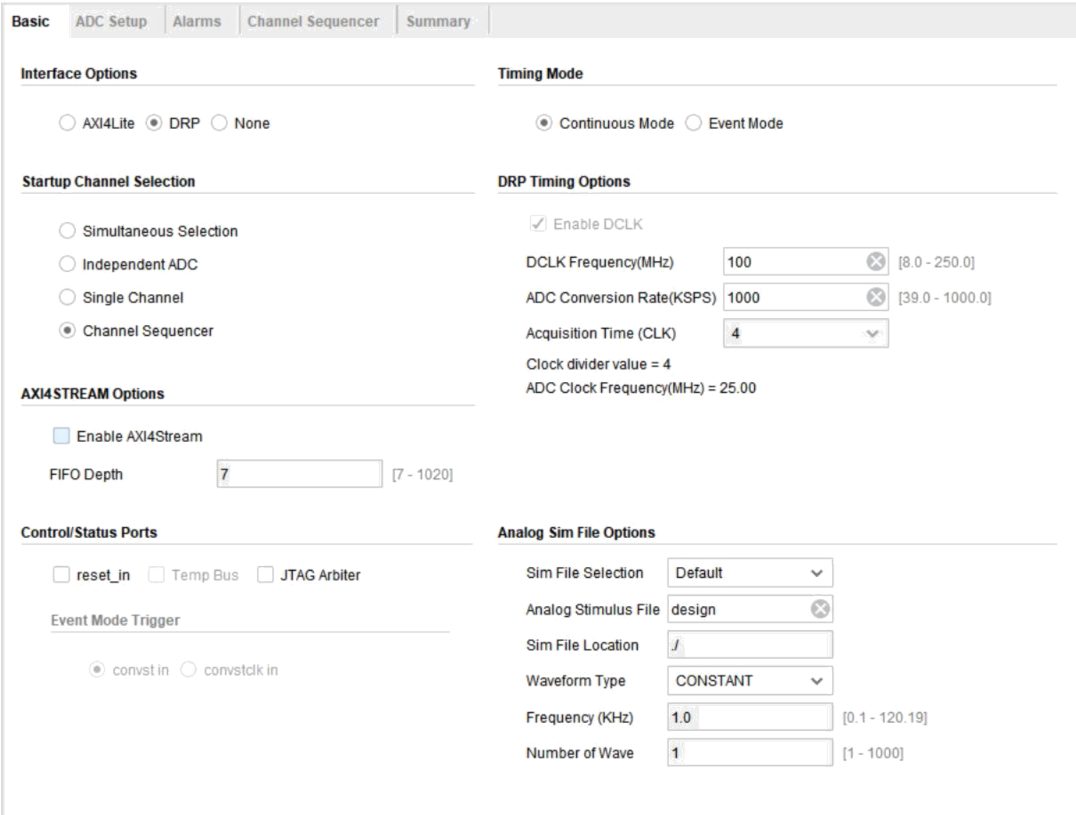
The design uses the four auxiliary analog input pairs available on the board. All the internal sensors have been excluded in the design to make it minimal and extremely simple. Out of the four input pairs, input pairs 6 and 15 are configured in unipolar mode and the other two pairs, 7 and 14, are configured in bipolar mode. The design uses the onboard 100 MHz clock.

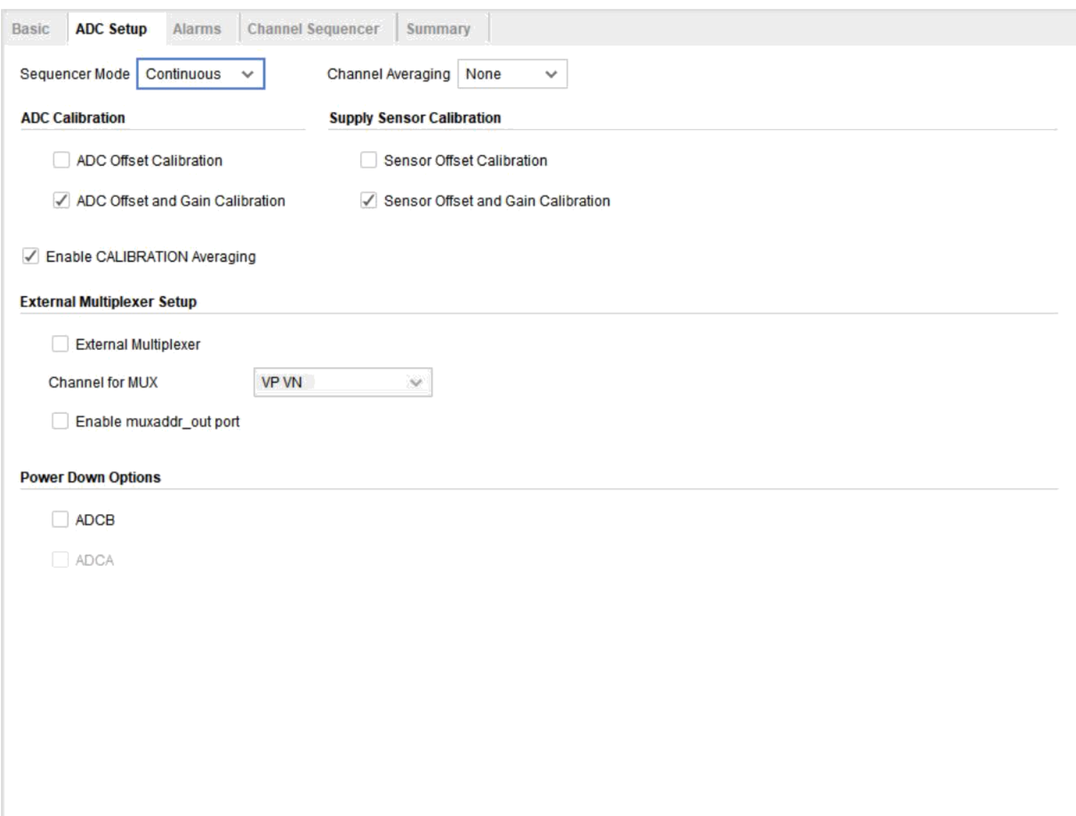
4. Configuration Options: Step by Step Tutorial

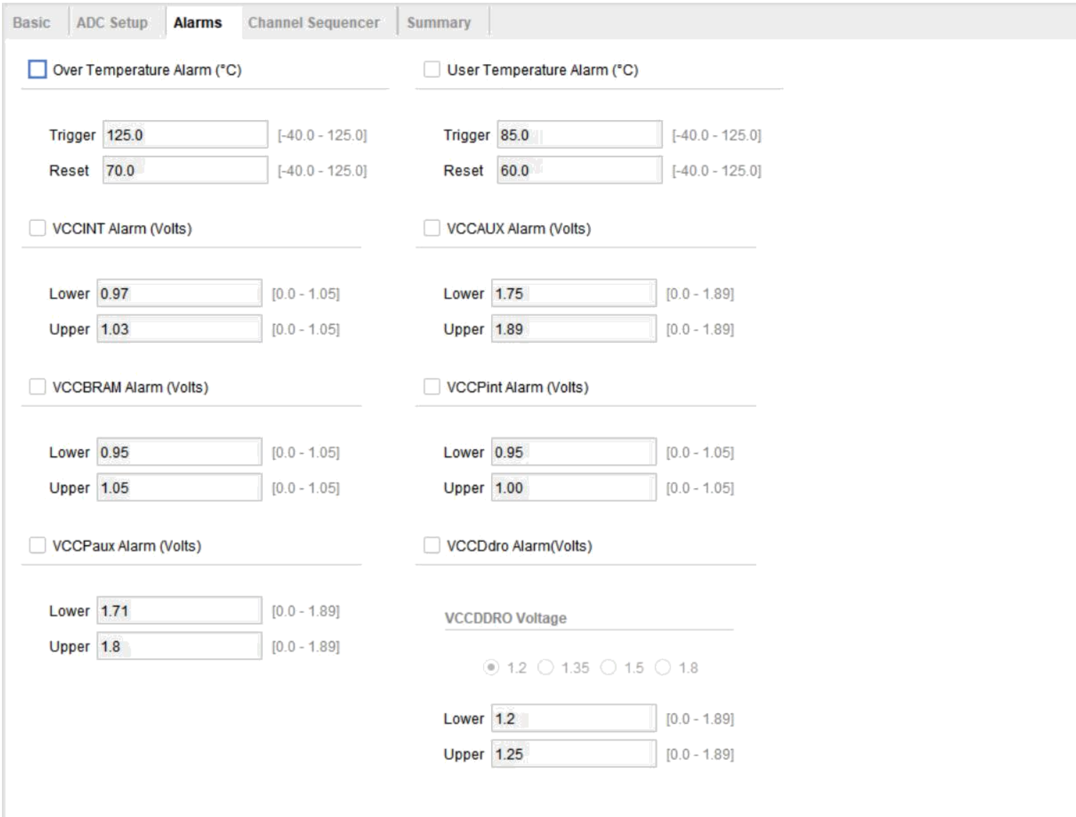
This section describes the configuration options used in the block design for the XADC and ILA.

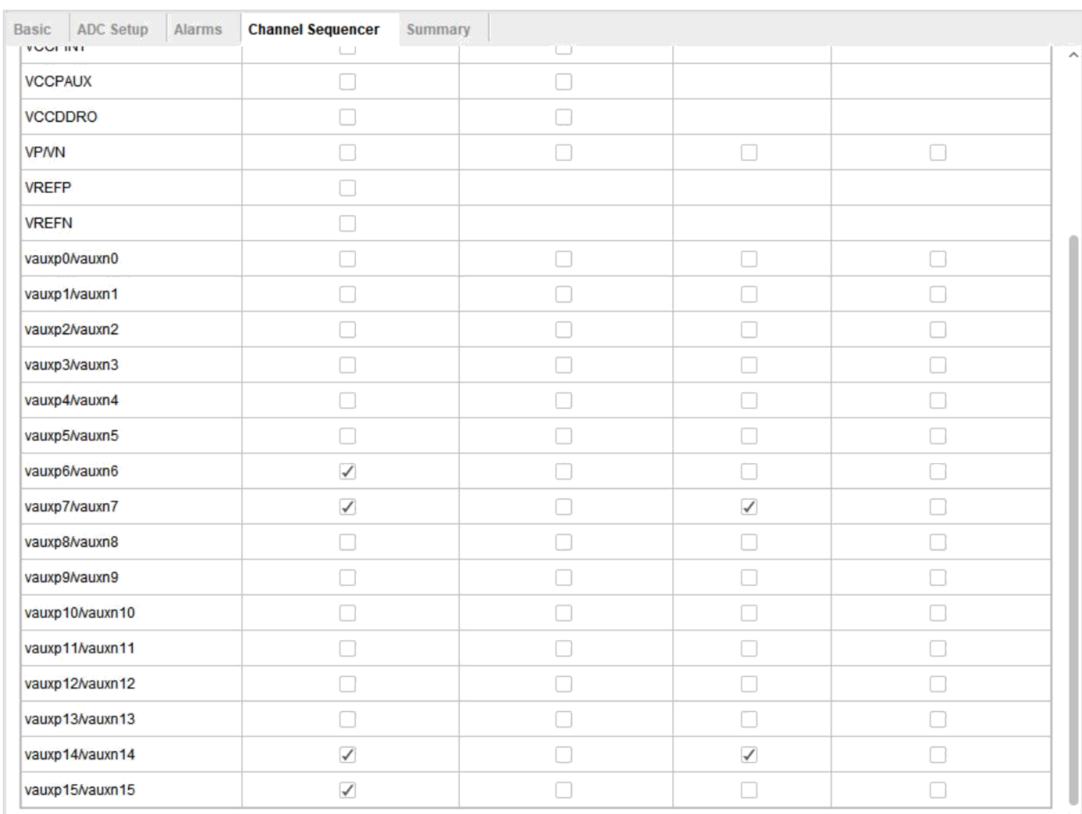
4.1 XADC

The XADC can be configured with following options.


Basic
ADC Setup
Alarms
Channel Sequencer
Summary
Interface Options
AXI4Lite
DRP
None
Timing Mode
Continuous Mode
Event Mode
Startup Channel Selection
Simultaneous Selection
Independent ADC
Single Channel
Channel Sequencer
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
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
Analog Sim File Options
Sim File Selection
Default
Analog Stimulus File
design
Sim File Location
./
Waveform Type
CONSTANT
Frequency (KHz)
1.0
[0.1 - 120.19]
Number of Wave
1
[1 - 1000]

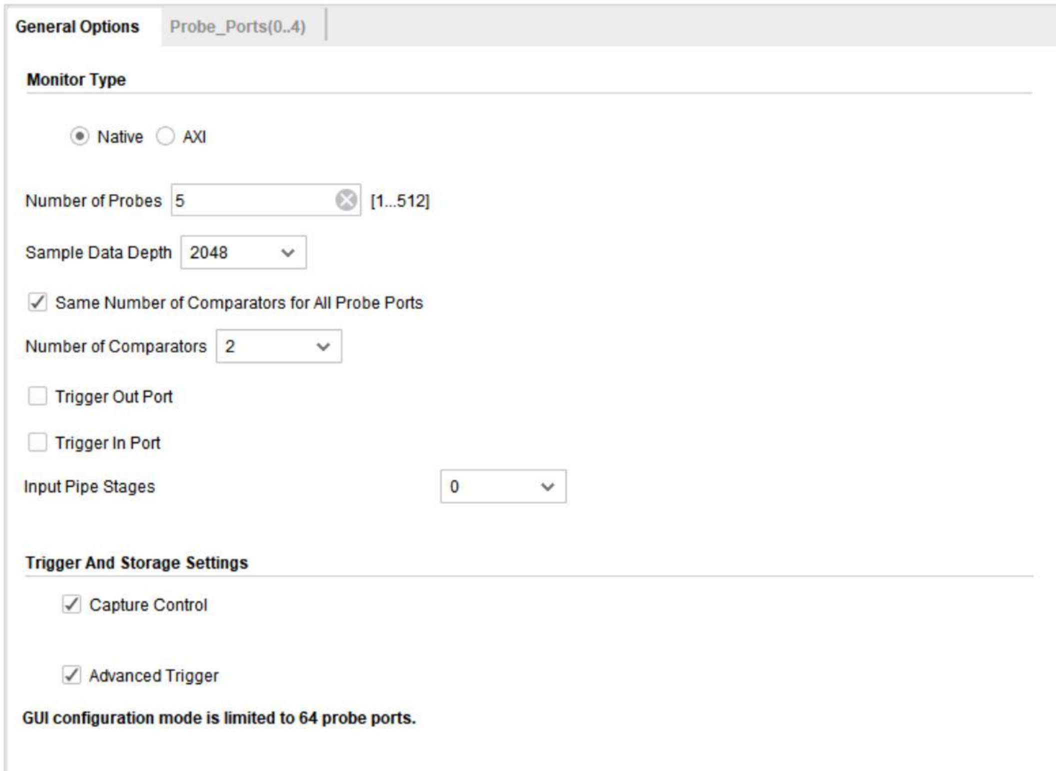

Basic
ADC Setup
Alarms
Channel Sequencer
Summary
Sequencer Mode
Continuous
Channel Averaging
None
ADC Calibration
ADC Offset Calibration
ADC Offset and Gain Calibration
Supply Sensor Calibration
Sensor Offset Calibration
Sensor Offset and Gain Calibration
Enable CALIBRATION Averaging
External Multiplexer Setup
External Multiplexer
Channel for MUX
VP VN
Enable muxaddr_out port
Power Down Options
ADCB
ADCA

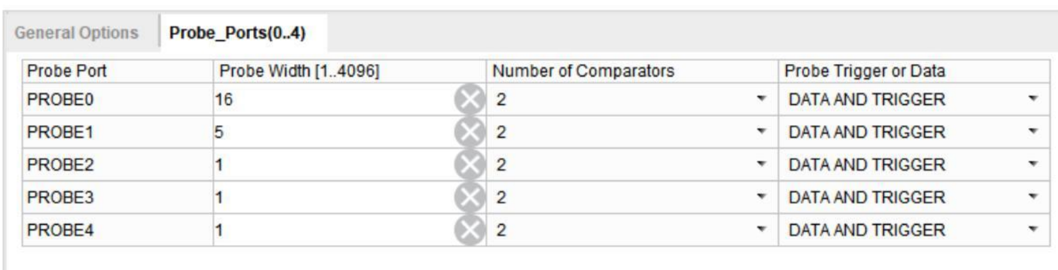

Basic
ADC Setup
Alarms
Channel Sequencer
Summary
Over Temperature Alarm (°C)
Trigger 125.0 [-40.0 - 125.0]
Reset 70.0 [-40.0 - 125.0]
User Temperature Alarm (°C)
Trigger 85.0 [-40.0 - 125.0]
Reset 60.0 [-40.0 - 125.0]
VCCINT Alarm (Volts)
Lower 0.97 [0.0 - 1.05]
Upper 1.03 [0.0 - 1.05]
VCCAUX Alarm (Volts)
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]
VCCPint Alarm (Volts)
Lower 0.95 [0.0 - 1.05]
Upper 1.00 [0.0 - 1.05]
VCCPaux Alarm (Volts)
Lower 1.71 [0.0 - 1.89]
Upper 1.8 [0.0 - 1.89]
VCCDdro Alarm(Volts)
VCCDDRO Voltage
1.2 1.35 1.5 1.8
Lower 1.2 [0.0 - 1.89]
Upper 1.25 [0.0 - 1.89]


Basic
ADC Setup
Alarms
Channel Sequencer
Summary
VCCPAUX
VCCDDRO
VP/VN
VREFP
VREFN
vauxp0/vauxn0
vauxp1/vauxn1
vauxp2/vauxn2
vauxp3/vauxn3
vauxp4/vauxn4
vauxp5/vauxn5
vauxp6/vauxn6
vauxp7/vauxn7
vauxp8/vauxn8
vauxp9/vauxn9
vauxp10/vauxn10
vauxp11/vauxn11
vauxp12/vauxn12
vauxp13/vauxn13
vauxp14/vauxn14
vauxp15/vauxn15

4.2 ILA

To verify the working of the design, the ILA is included in the design. It can be configured with following options.





4.3 Integration of XADC and ILA

Integrate the XADC and ILA core as shown on the following block design:

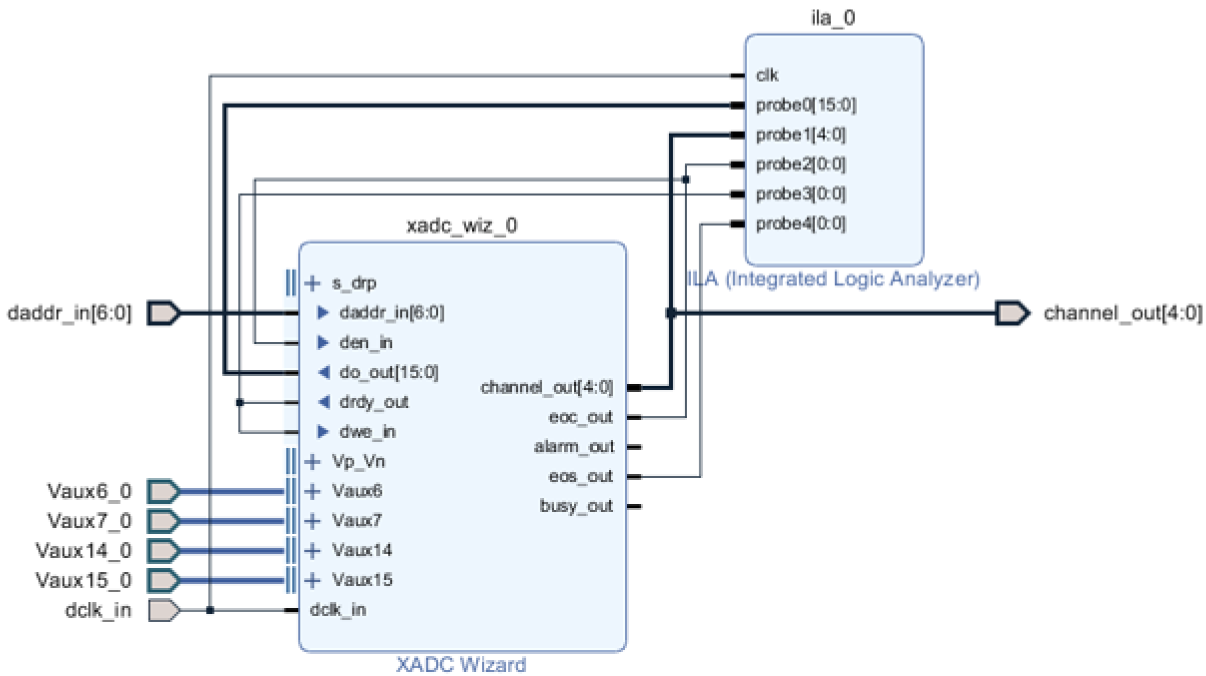


Figure 7: XADC and ILA Integration, Block Design

4.4 Including the Constraint on the Design

After step 4.3 completes, add the constraint source, name the constraint file as "Zybo_XADC" or any other. And open the constraint file from "Sources" tab of VIVADO. Then copy following constraint and paste on the Vivado constraint "Zybo_XADC".

```
##Clock signal
set_property -dict {PACKAGE_PIN L16 IOSTANDARD LVCMOS33}
[get_ports dclk_in]
create_clock -period 10.000 -name sys_clk_pin -waveform {0.000
5.000} -add [get_ports dclk_in]

##Pmod Header JA (XADC)
set_property -dict {PACKAGE_PIN N16 IOSTANDARD LVCMOS33}
[get_ports Vaux14_0_v_n]
set_property -dict {PACKAGE_PIN N15 IOSTANDARD LVCMOS33}
[get_ports Vaux14_0_v_p]
set_property -dict {PACKAGE_PIN L15 IOSTANDARD LVCMOS33}
[get_ports Vaux7_0_v_n]
set_property -dict {PACKAGE_PIN L14 IOSTANDARD LVCMOS33}
[get_ports Vaux7_0_v_p]
set_property -dict {PACKAGE_PIN J16 IOSTANDARD LVCMOS33}
[get_ports Vaux15_0_v_n]
set_property -dict {PACKAGE_PIN K16 IOSTANDARD LVCMOS33}
[get_ports Vaux15_0_v_p]
set_property -dict {PACKAGE_PIN J14 IOSTANDARD LVCMOS33}
[get_ports Vaux6_0_v_n]
```

```
set_property -dict {PACKAGE_PIN K14 IOSTANDARD LVCMOS33}
[get_ports Vaux6_0_v_p]
```

4.5 Synthesize, Implement and Generate the Design

After the step 4.4 completes, validate the design. After the validation, run the synthesis, implement and generate the design.

4.6 Hardware Programming

When the bitstream is generated, its time to connect your FPGA Board [Zybo FPGA] with PC with USB JTAG Cable and setting the Board on JTAG Configuration mode.
Now program the FPGA Board.
After programming, the hardware ILA manger will open as follows:

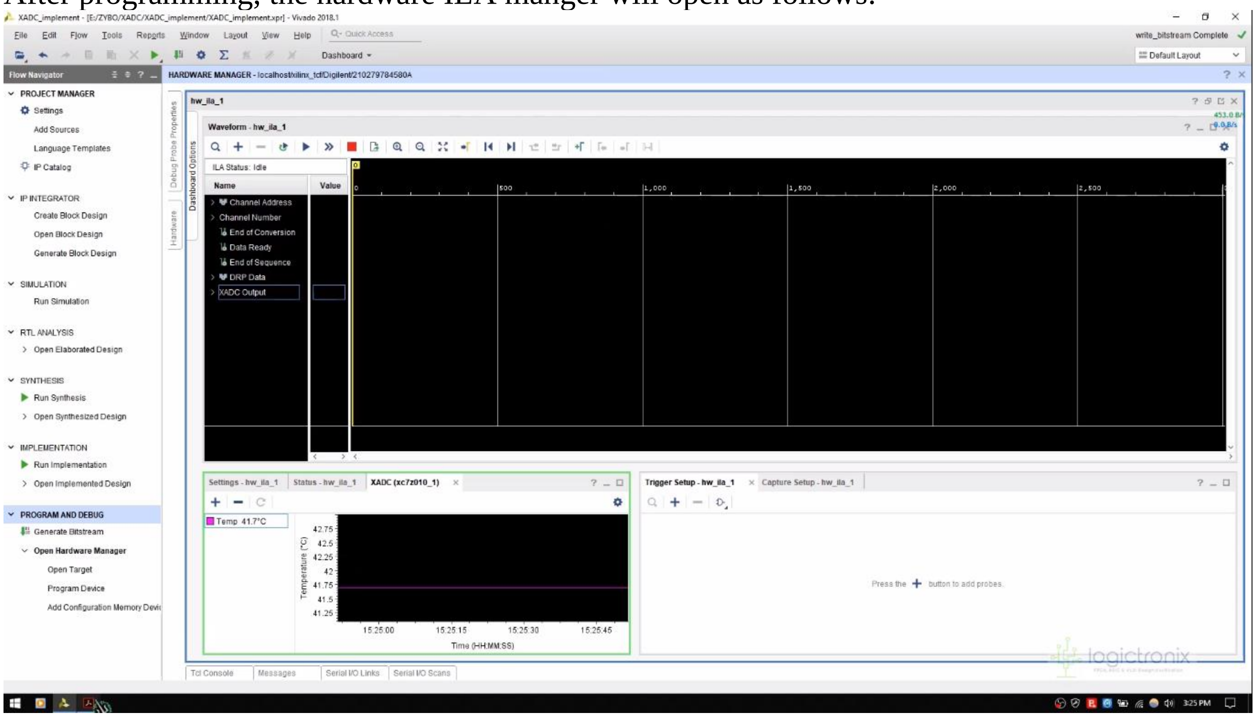


Figure 8: ILA Manager after programming FPGA Board

Now re-trigger the ILA from the option on GUI, click on “>>” icon. Now you will get the data on the ILA waveform window. More details of the ILA waveform window is presented on the Video.

5. Verification

The output of the XADC is given by the following equation.

$$D[0:11] = \frac{Vin}{1} x4095 \qquad \text{equation 1}$$

The input channels 6 and 15 were connected to 0.511 V and 0.995 V respectively. Hence, from the equation 1,

$$D[0:11],6 = \frac{0.511}{1} x4095 \cong 834h$$

$$D[0:11],15 = \frac{0.995}{1} x4095 \cong fe8h$$

The input channels 7 and 14, being configured in bipolar mode uses 2's complement for representation. No input was connected to channel 14. The channel 7 was first connected to 0.753 V and then to 0.340 V. The channel reference was 0.5 V. Then from equation 1,

$$D[0:11],7 = \frac{(0.753 - 0.5)}{1} x4095 \cong 414h$$

$$D[0:11],7 = \frac{(0.340 - 0.5)}{1} x4095 \cong d74h$$

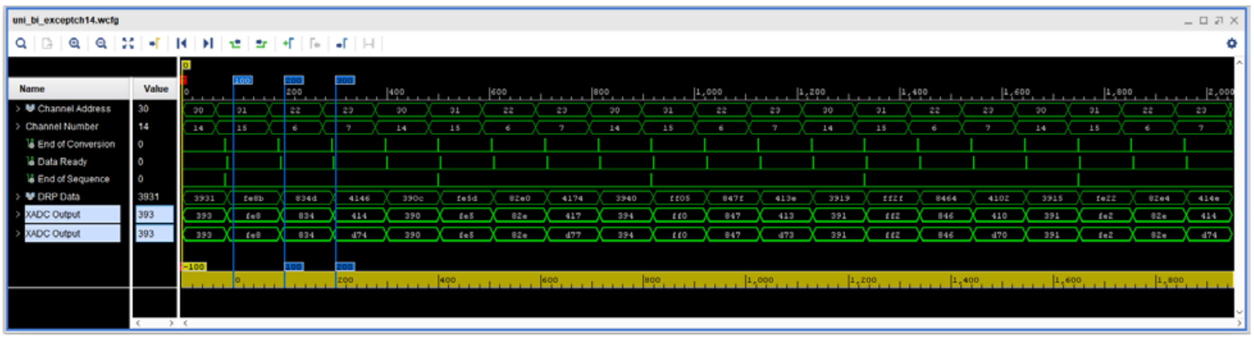


Figure 5: Waveform

The outputs of the channels are marked on the waveform by markers.

6. References

[1] I. Xilinx, "7 Series FPGAs and Zynq-7000 All Programmable SoC XADC Dual 12-Bit 1 MSPS Analog-to-Digital Converter User Guide (UG480)," Xilinx, Inc., December 23, 2017.