ADC Core Module

User Guide

FPGA-IPUG-02168-1.5

January 2025

Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS, with all faults, and all associated risk is the responsibility entirely of the Buyer. The information provided herein is for informational purposes only and may contain technical inaccuracies or omissions, and may be otherwise rendered inaccurate for many reasons, and Lattice assumes no obligation to update or otherwise correct or revise this information. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. LATTICE PRODUCTS AND SERVICES ARE NOT DESIGNED, MANUFACTURED, OR TESTED FOR USE IN LIFE OR SAFETY CRITICAL SYSTEMS, HAZARDOUS ENVIRONMENTS, OR ANY OTHER ENVIRONMENTS REQUIRING FAIL-SAFE PERFORMANCE, INCLUDING ANY APPLICATION IN WHICH THE FAILURE OF THE PRODUCT OR SERVICE COULD LEAD TO DEATH, PERSONAL INJURY, SEVERE PROPERTY DAMAGE OR ENVIRONMENTAL HARM (COLLECTIVELY, "HIGH-RISK USES"). FURTHER, BUYER MUST TAKE PRUDENT STEPS TO PROTECT AGAINST PRODUCT AND SERVICE FAILURES, INCLUDING PROVIDING APPROPRIATE REDUNDANCIES, FAIL-SAFE FEATURES, AND/OR SHUT-DOWN MECHANISMS. LATTICE EXPRESSLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS OF THE PRODUCTS OR SERVICES FOR HIGH-RISK USES. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

Inclusive Language

This document was created consistent with Lattice Semiconductor’s inclusive language policy. In some cases, the language in underlying tools and other items may not yet have been updated. Please refer to Lattice’s inclusive language FAQ 6878 for a cross reference of terms. Note in some cases such as register names and state names it has been necessary to continue to utilize older terminology for compatibility.

Contents

Figures

Tables

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
ADC	Analog to Digital Converter
COG	Conversion Ongoing
DNL	Differential Non Linearity
DNU	Do Not Use
DTR	Digital Temperature Readout
EOC	End of Conversion
FPGA	Field Programmable Gate Array
GPLL	General Purpose PLL
GUI	Graphical User Interface
HIP	Hard Intellectual Property.
INL	Integral Non Linearity
IP	Intellectual Property
LSB	Least Significant Bit
MSPS	Mega Samples Per Second
MUX	Multiplexer
PLL	Phase-Locked Loop
PTAT	Proportional to Absolute Temperature
RTL	Register Transfer Logic
SAR	Successive Approximation Register
SIP	Soft Intellectual Property
SNDR	Signal to Noise Dynamic Range
SOC	Start-of-Conversion
THD	Total Harmonic Distortion

1. Introduction

1.1. Overview of the IP

The Lattice Semiconductor Nexus-series ADC feature comes with two different IP support which are ADC Core Module IP and ADC Sequencer IP. This user guide focuses on the ADC Core Module IP and covers the IP feature description, IP function description, IP configuration, and steps to create the ADC design in the Lattice Radiant software.

Table 1.1. High-level Overview of the ADC IPs

Feature	ADC Core Module IP	ADC Sequencer IP
Supported hardware feature	Comparator channels, dual function analog input channels, dedicated analog input channels, measure internal voltage rails, and digital temperature readout (DTR)	
Initiate PLL IP	Manual instantiation of the PLL IP	PLL IP instantiation covered by ADC sequencer IP
Built-in sequencer controller and threshold monitor	No	Yes
LMMI/APB support	No	Yes
Logic utilization	Low	High

1.2. Quick Facts

Table 1.2. Summary of the ADC Core Module IP

IP Requirements	Supported FPGA Family	Certus™-NX, CertusPro™-NX, CrossLink™-NX (excluding LIFCL-33), MachXO5™-NX, MachXO5T-NX
Resource Utilization	Resources	Refer to Table A.1
Design Tool Support	Synthesis	Lattice Synthesis Engine (LSE) Synopsys® Synplify Pro® for Lattice
	Lattice Implementation	V 1.3.1 – Lattice Radiant™ software 2022.1 and later
	Simulation	No simulation support
	Supported User Interface	Native

1.3. Features

The ADC Core Module IP supports the following features:

- Configurable clock divider to determine the ADC sampling rate
- User-define ADC conversion mode and ADC data output format
- Configurable reference voltage source and external reference voltage source level
- Enable for comparator channels and input source selection
- Enable for dual function ADC channel (dedicated ADC input channels are enabled by default)
- Able to control measure channel sequencing via channel select
- Able to measure the internal voltage rail such as VCC, VCCAUX, VCCIO0, VCCIO1, VCCM
- Able to measure the device junction temperature via digital temperature readout (DTR)

1.4. Licensing and Ordering Information

The ADC Core Module IP is provided at no additional cost with the Lattice Radiant software.

1.5. IP Validation Summary

The IP is timing validated with the following synthesis tools:

- Synopsys Synplify Pro Q-2023.1LR, Build 151R, February 7 2023
- Lattice Synthesis Engine (LSE)

1.6. Minimum Device Requirements

All devices with ADC block as mentioned in Table 1.1.

1.7. Naming Conventions

1.7.1. Nomenclature

The nomenclature used in this document is based on Verilog HDL.

1.7.2. Signal Names

- _*n* are active low (asserted when value is logic 0)
- _*i* are input signals
- _*o* are output signals

2. Functional Description

2.1. IP Architecture Overview

Figure 2.1 shows the connection between the ADC block and ADC Core Modular IP. The ADC Core Module IP controls and manages the ADC block. Use this ADC Core Module IP to configure the ADC settings and connect with the user logic design. The diagram below shows the entire HIP which is instantiated when you instantiate the ADC IP from the GUI provided by the IP Catalog.

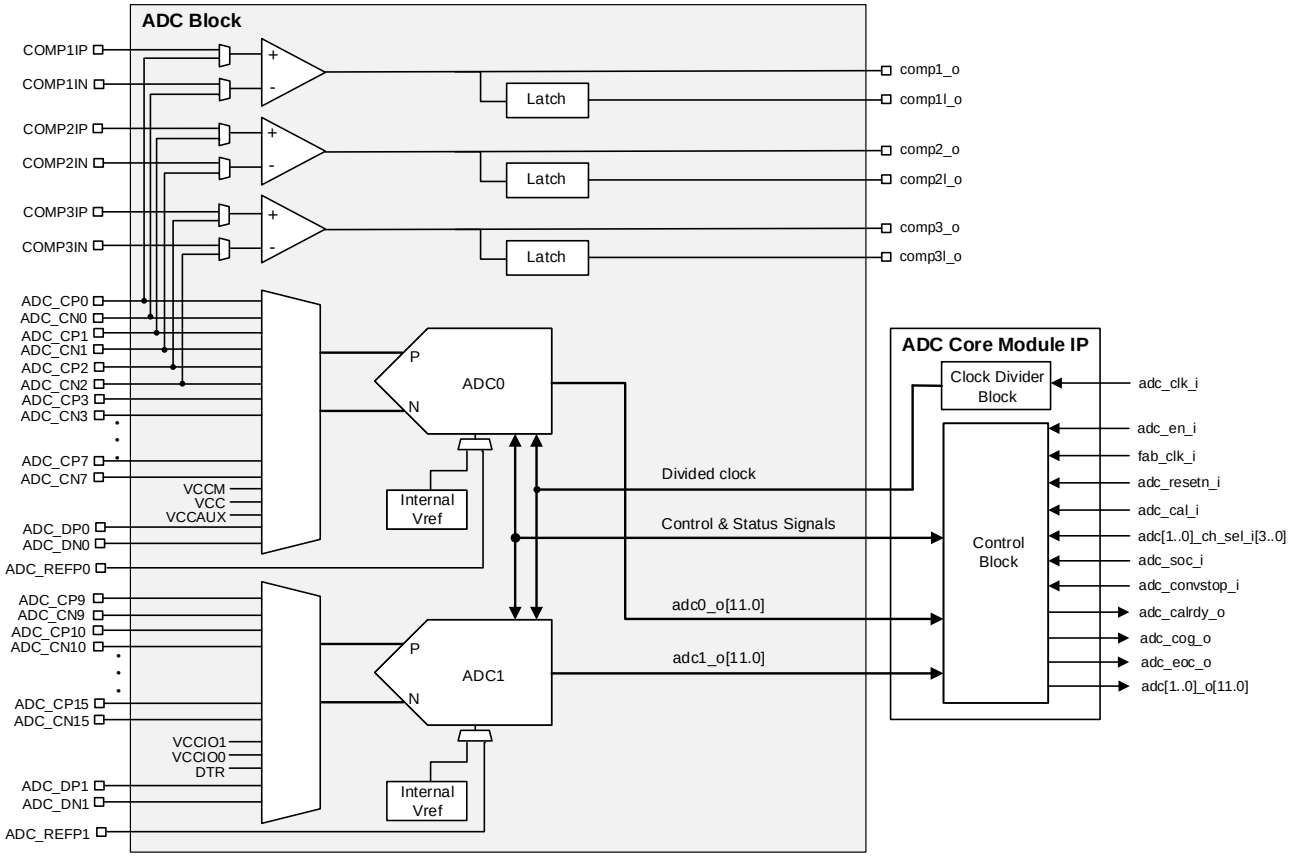


Figure 2.1. Lattice ADC Core Module HIP

2.2. Clocking

When implementing ADC Core Module IP in your design, provide the following two clock sources to the ADC Core Module IP:

- adc_clk_i

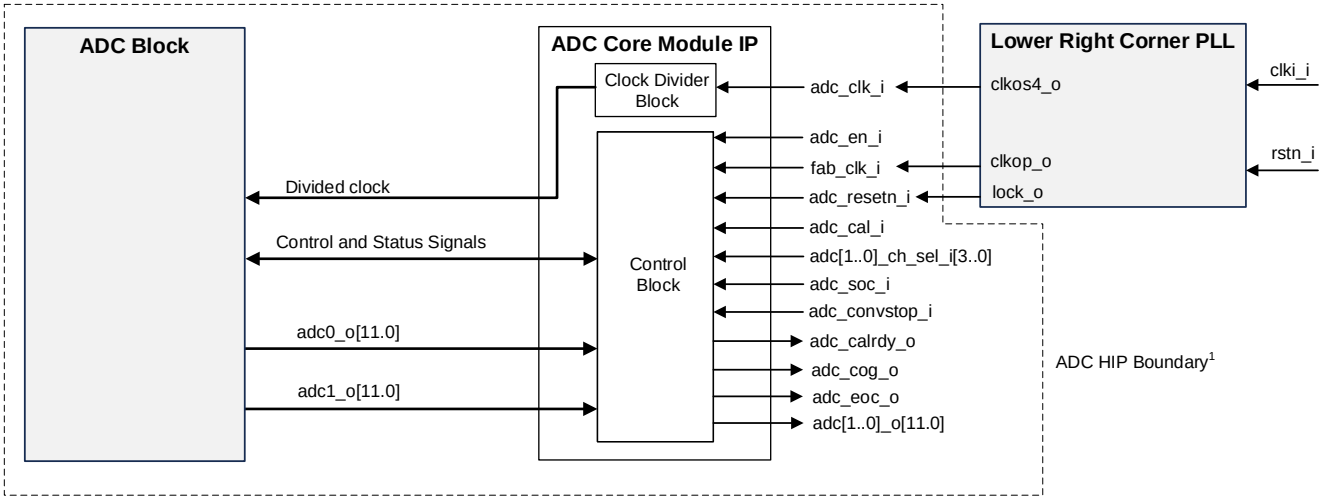
  This clock is the ADC sampling clock that passes through the ADC clock divider to slow down the ADC clock rate to meet the sampling requirement.

- fab_clk_i

  This clock is used as the clock for the ADC control block. All the signals/control functions in this block are reference to the fab_clk_i source. The comparator latched output also uses this clock.

2.2.1. Clocking Overview

The ADC implementation requires two clock sources. It is recommended to generate the synchronized clock source to drive the adc_clk_i and fab_clk_i from PLL, as shown in Figure 2.2. Use only the PLL located at the lower right corner to drive the clock source to ADC block. The clock source for adc_clk_i must come from PLL clkos4. The clock, fab_clk_i can receive signal from any PLL output port.



Note:
1. The ADC HIP is what is instantiated from the IP catalog GUI.

Figure 2.2. ADC Implementation Clock Domain Block Diagram

2.3. Reset

This section describes the reset and initialization sequence for ADC Core Module IP implementation.

2.3.1. Reset Overview

During operation, the ADC Core Module IP requires clock supply from PLL. As shown in Figure 2.2, PLL provides adc_clk_in and fab_clk_in to the ADC Core Module IP. For the ADC Core Module IP to function properly, lock the PLL to a stable clock source. ADC reset and initialization sequence includes PLL reset.

Lattice recommends following these steps to reset and initialize the ADC design:

1. Connect the PLL IP input clock (clki_i) to the reference clock source. The adc_clk_i signal must come from the lower right PLL. After division inside the ADC HIP, the frequency should not exceed 25 MHz. The clock speed to fab_clk_i should not exceed 40 MHz.
2. Assert the PLL reset (make rstn_i = 1) and wait for PLL locked (lock_o) to assert. Before the PLL locked is stable, adc_resetn_i and adc_cal_i must be in reset mode.
3. De-assert the ADC Core Module IP reset (adc_resetn_i).
4. Assert adc_en_i to enable the ADC block. You can switch the order of Steps 3 and 4.
5. Wait for at least 8 clock cycles of adc_clk_i or 12 cycles of fab_clk_i, then assert adc_cal_i.
6. Wait for adc_calrdy_o to assert. It takes about 5500 adc_clk_i cycles.

When adc_calrdy_o is asserted high, the ADC is in operation mode. You can start the ADC read out accordingly. Figure 2.3 shows the ADC reset and calibration sequence.

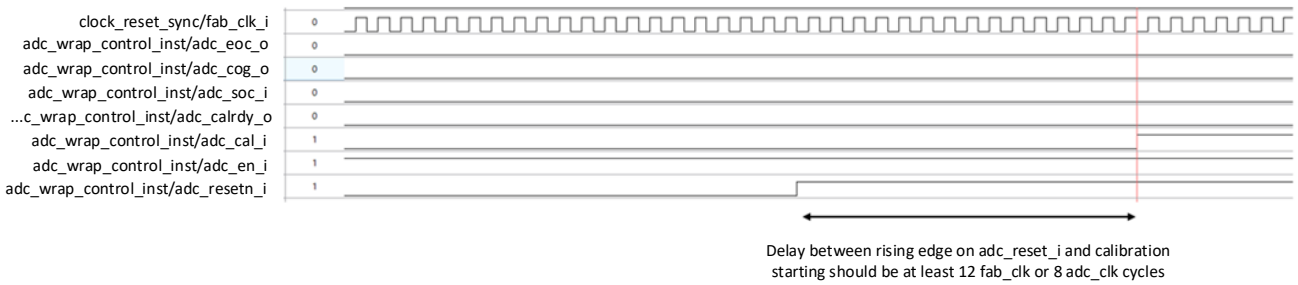


Figure 2.3. ADC Reset and Calibration Sequence

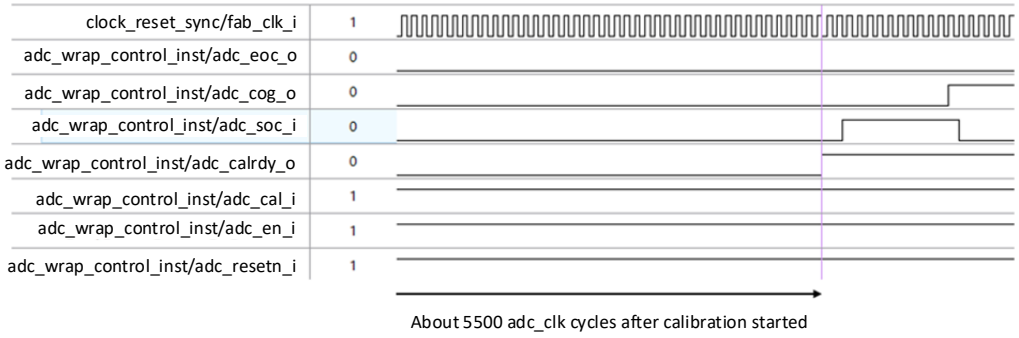


Figure 2.4. ADC Calibration Completion and Conversion Starting

If you use only the comparator function, skip steps 4–6. Each comparator has two outputs: a continuous output and a latch output. The latched output is synchronized with fab_clk_i and cleared by adc_resetn_i. The adc_resetn_i is shared with the ADC block. If you clear the latched comparator output, the ADC falls to reset mode. Follow steps 4–6 to reinitialize and calibrate the ADC.

Notes:

1. If PLL losses its lock, the ADC sampled data may be inaccurate. Reinitialize and recalibrate the ADC block after PLL locked is restored.
2. If PLL is locked but you have reset the ADC using adc_resetn_i, reinitialize and recalibrate the ADC design from steps 4–6.
3. Calibrate both ADCs at same time. You cannot calibrate only one ADC at a time.
4. Calibration is only required before ADC enters operation mode. Do not need to recalibrate the ADC over time.
5. Interrupting the ADC conversion with recalibration causes the ADC data to be invalid.

2.4. User Interfaces

ADC Core Module IP supports only native interface. AHB-LITE and LMMI interfaces are not supported.

2.5. ADC Core Module IP Implementation

The output clock to drive the ADC Core Module IP, adc_clk_i, must come from PLL clkos4_o of the lower right PLL. The clock to drive fab_clk_i can be any output clock port from same PLL. Refer to Figure 2.2 for more details. You can use dedicated reference clock pin (GPLL) or non-dedicated reference clock pin (PCLK) to drive the PLL IP. Connecting the clkos4_o output from the PLL to the ADC IP causes the Radiant software to designate the PLL as the lower right PLL. If you write manual constraint, ensure that you do not assign some other PLL to lower right.

3. IP Parameter Description

The configurable attributes of the ADC Core Module IP are shown in the following tables. You can configure the IP by setting the attributes accordingly in the IP Catalog's Module/IP wizard of the Lattice Radiant software.

Wherever applicable, default values are in **bold**.

3.1. General Tab

Table 3.1 shows the general attributes of the ADC Core Module IP.

Table 3.1. General Attributes

Attribute	Selectable Values	Description
ADC Clock Divide	**2**, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40	Configures the clock divider factor for adc_clk_i. The divided clock is the ADC sampling clock. Refer to the Controlling Sampling Frequency and Maximum Sampling Frequency sections for details.
ADC Conversion Data Format	**Straight Binary**, Twos Complement	Configures the ADC conversion output data format. Refer to the ADC Conversion Format section for details.
ADC0 Reference Voltage Select	**Internal**, External	Selects the internal or external reference voltage for ADC0. When using internal reference mode, the voltage is fixed to 1.2 V. For accurate measurements, external reference is recommended.
ADC0 External Reference Voltage (V)	1.0, 1.1, **1.2**, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8	Configures the external reference voltage value connected to ADC_REFP[0] for ADC0.
ADC0 Conversion Mode	**Uni-Polar**, Bi-Polar	Configures the ADC conversion mode for ADC0.
ADC1 Reference Voltage Select	**Internal**, External	Selects the internal or external reference voltage for ADC1. When using internal reference mode, the voltage is fixed to 1.2 V. For accurate measurements, external reference is recommended.
ADC1 External Reference Voltage (V)	1.0, 1.1, **1.2**, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8	Configures the external reference voltage value connected to ADC_REFP[1] for ADC1.
ADC1 Conversion Mode	**Uni-Polar**, Bi-Polar	Configures the ADC conversion mode for ADC1.

3.2. Comparator Tab

Table 3.2 shows the attributes in the Comparator tab of the ADC Core Module IP.

Table 3.2. Comparator Tab

Attribute	Selectable Values	Description
COMP1 Enable	Checked, **Unchecked**	Checked to enable comparator 1 function.
COMP1 Input Source Select	**COMP1IP and COMP1IN inputs**, ADC_CP0 and ADC_CN0 inputs	Grayed out if COMP1 is not enabled. Select the comparator input source.
COMP2 Enable	Checked, **Unchecked**	Checked to enable comparator 2 function.
COMP2 Input Source Select	**COMP2IP and COMP2IN inputs**, ADC_CP1 and ADC_CN1 inputs	Grayed out if COMP2 is not enabled. Select the comparator input source.
COMP3 Enable	Checked, **Unchecked**	Checked to enable comparator 3 function.
COMP3 Input Source Select	**COMP3IP and COMP3IN inputs**, ADC_CP2 and ADC_CN2 inputs	Grayed out if COMP3 is not enabled. Select the comparator input source.

3.3. ADC Channels Tab

Table 3.3 shows the attributes in the ADC Channels tab of the ADC Core Module IP.

Table 3.3. ADC Channels Tab

Attribute	Selectable Values	Description
ADC0 Channels		
Channel 1 Enable	Checked, **Unchecked**	Checked to enable ADC0 Channel 1
Channel 2 Enable	Checked, **Unchecked**	Checked to enable ADC0 Channel 2
Channel 3 Enable	Checked, **Unchecked**	Checked to enable ADC0 Channel 3
Channel 4 Enable	Checked, **Unchecked**	Checked to enable ADC0 Channel 4
Channel 5 Enable	Checked, **Unchecked**	Checked to enable ADC0 Channel 5
Channel 6 Enable	Checked, **Unchecked**	Checked to enable ADC0 Channel 6
Channel 7 Enable	Checked, **Unchecked**	Checked to enable ADC0 Channel 7
Channel 8 Enable	Checked, **Unchecked**	Checked to enable ADC0 Channel 8
ADC1 Channels		
Channel 9 Enable	Checked, **Unchecked**	Checked to enable ADC1 Channel 9
Channel 10 Enable	Checked, **Unchecked**	Checked to enable ADC1 Channel 10
Channel 11 Enable	Checked, **Unchecked**	Checked to enable ADC1 Channel 11
Channel 12 Enable	Checked, **Unchecked**	Checked to enable ADC1 Channel 12
Channel 13 Enable	Checked, **Unchecked**	Checked to enable ADC1 Channel 13
Channel 14 Enable	Checked, **Unchecked**	Checked to enable ADC1 Channel 14
Channel 15 Enable	Checked, **Unchecked**	Checked to enable ADC1 Channel 15
Channel 16 Enable	Checked, **Unchecked**	Checked to enable ADC1 Channel 16

3.4. IP Configuration Settings for Example Use Cases

This section provides the IP configuration settings for your use cases.

3.4.1. Controlling Sampling Frequency

The adc_clk_i and fab_clk_i can be asynchronous. If they are asynchronous, then signals going between domains have a variable latency. However, you can derive formulae that give the maximum latency or the minimum sampling frequency based on adc_clk_i and fab_clk_i frequency. In the formulae, SOC time T_{SOC} is the time from the SOC going high to the time adc_soc_i goes 0 in response to adc_cog_o becoming 1. The EOC time T_{EOC} is the time from adc_cog_o becoming 1 and adc_eoc_o becoming 1. These are shown in Figure 3.1.

$$T_{SOC} = 10T_{ADCCLK} \times CLKDIV + 6T_{FABCLK} \qquad \textbf{Eq. 1.}$$

$$T_{EOC} = 34T_{ADCCLK} \times CLKDIV + 16T_{FABCLK} \qquad \textbf{Eq. 2.}$$

Where:

- T_{ADCCLK} is the period of adc_clk_i.
- T_{FABCLK} is the period of fab_clk_i.
- The sampling period of the ADC is the sum of T_{SOC} and T_{EOC}.
- CLKDIV is the value by which the adc_clk_i is internally divided, as shown in Figure 3.2.

The values listed above are a floor on the ADC sampling frequency, and the ADC sampling frequency can exceed the value predicted by the equations above.

If a lower sampling frequency is required, then adc_soc_i can be kept 1 even after adc_cog_o becomes 1 to increase the sampling period. If adc_clk_i and fab_clk_i are of different frequencies or have a variable phase shift, the sampling period changes slightly due to the latency of signals going between the clock domains. If a constant sampling frequency is required (one such use case is the spectral analysis of an input signal), adc_clk_i and fab_clk_i can be made synchronous and come from the lower right PLL.

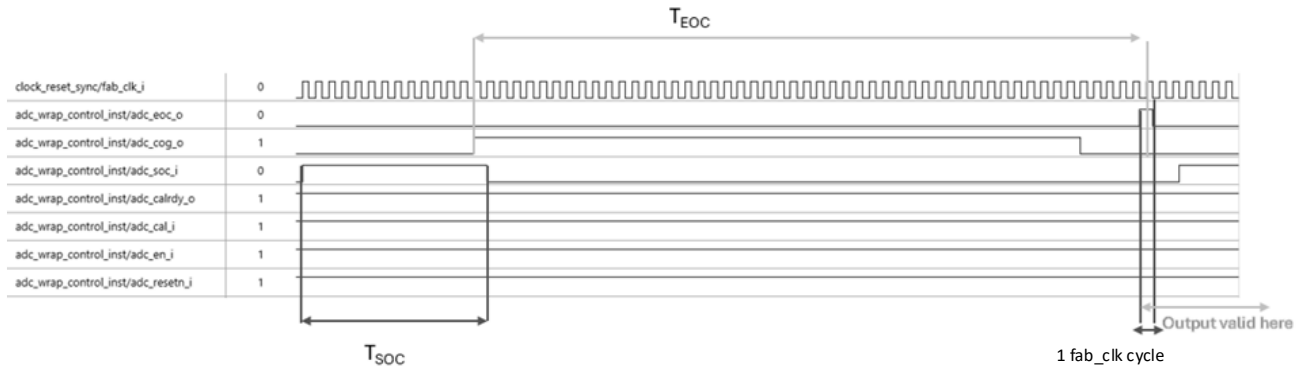


Figure 3.1. ADC Timing Information

3.4.2. Maximum Sampling Frequency

To achieve the maximum sampling frequency, set fab_clk_i to 40 MHz and the internal ADC clock to 25 MHz, both configured in the PLL. Then, set the clock divider in the ADC block to 2. This configuration will result in a sampling frequency of 623 ksps. The internal ADC clock is the adc_clk_i divided by the internal ADC clock divider. This can be achieved by setting adc_clk_i as 50 MHz and the clock divider as 2, or adc_clk_i as 100 MHz and the clock divider as 4, and so on. The ADC has less jitter if adc_clk_i operates at a higher frequency and is divided by a larger number.

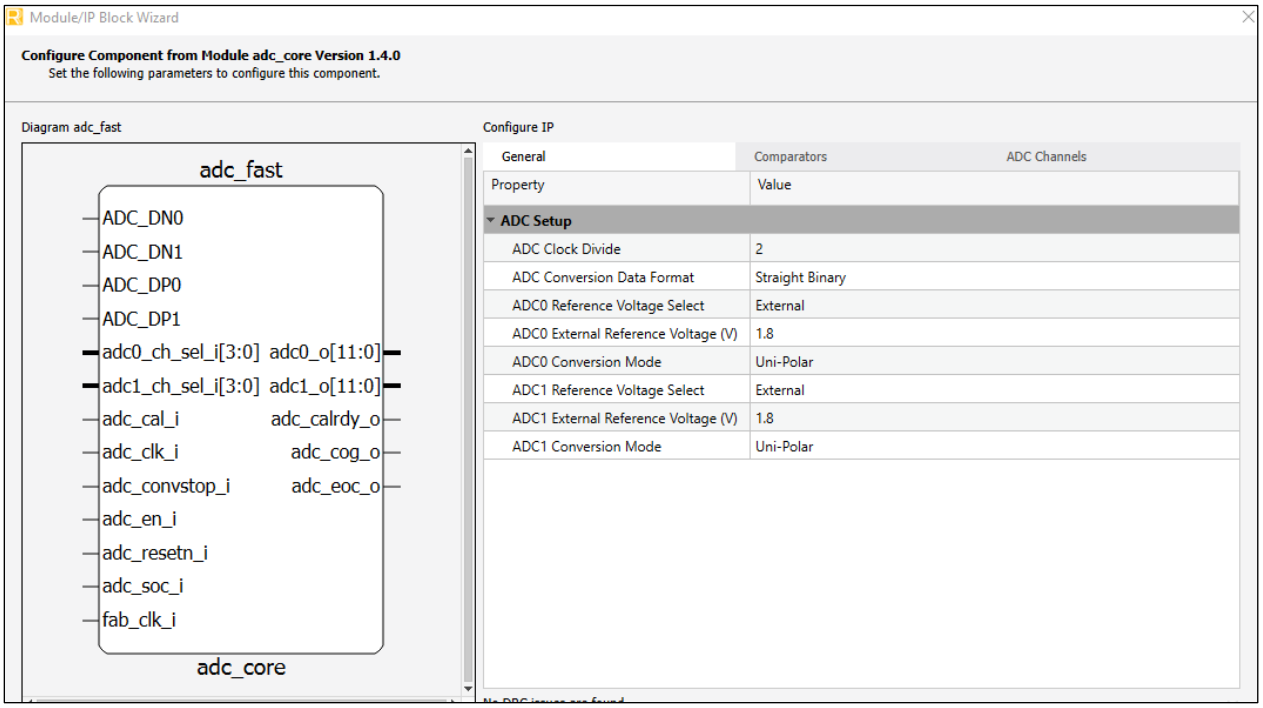


Figure 3.2. ADC Configuration Options

3.4.3. Calculating the Sampling Rate for Each ADC Channel

The sampling rate calculated in the Controlling Sampling Frequency and Maximum Sampling Frequency sections refers to the entire sampling rate for all ADC0 and ADC1 channels. The ADC0 sampling rate is shared among the channels within ADC0 and the same applies for ADC1. See Figure 2.1 for more details. When your design enables more ADC channels and the sampling rates are shared equally, take the total sampling divided by the enabled ADC channels to calculate the maximum sampling rate for each channel.

For example, ADC operates at a maximum sampling rate of 623 ksps and you have enabled 5 channels in ADC0 and 2 channels in ADC1. If the sampling points are shared equally by these channels, the maximum sampling rate for each channel is 623 ksps ÷ 5 = 124 ksps for ADC0 and 623 ksps ÷ 2 = 321 ksps for ADC1.

Enabling the comparator channel does not affect the sampling rate. The comparator latched output is directly clocked by fab_clk_i, and the clock rate is not affected by the number of comparator channel.

3.4.4. ADC Conversion Format

The ADC Core Module IP supports analog to digital conversion to bipolar mode and unipolar mode.

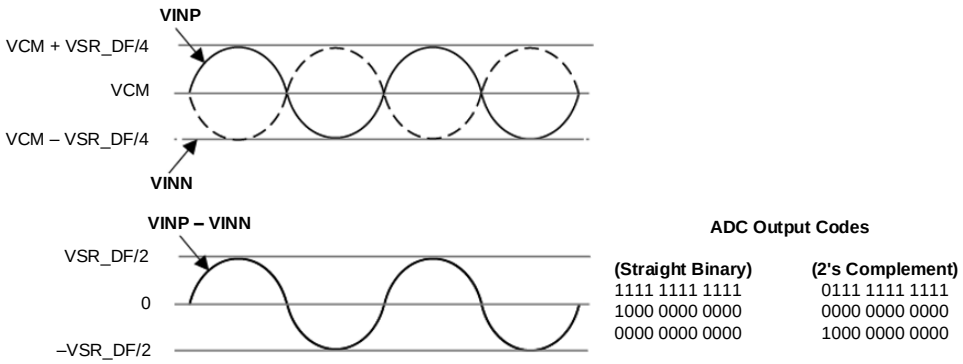


Figure 3.3. Bipolar Input Signal and Output Code Example

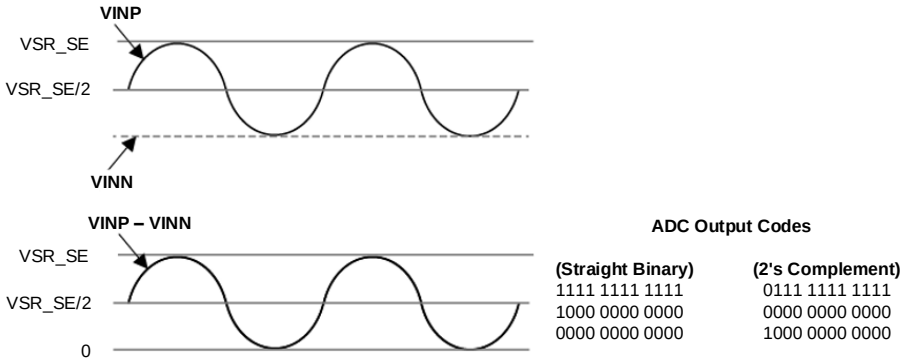


Figure 3.4. Unipolar Input Signal and Output Code Example

3.4.5. Reference Voltage Selection

You can configure the reference voltage model and the voltage level in the IP parameter for both ADC. The ADC reference voltage works independently, where you can configure different mode or voltage level per design implementation. However, Lattice recommends using external reference voltage mode in the design to achieve high precision analog to digital conversion. If using the internal reference voltage, the voltage level is fixed at 1.2 V and the accuracy is ±15%.

In the design implementation, the reference voltage applied must cover the measurement range. For example, if the analog input signal is 1.35 V, the recommended reference voltage is 1.4 V. The reference voltage should not be lower than the input signal. When using internal reference voltage, V_{CCIO0}, V_{CCIO1}, and V_{CCAUX} must operate at 2.5 V or lesser.

4. Signal Description

This section provides the information about the ADC Core Module IP ports function and the supported settings.

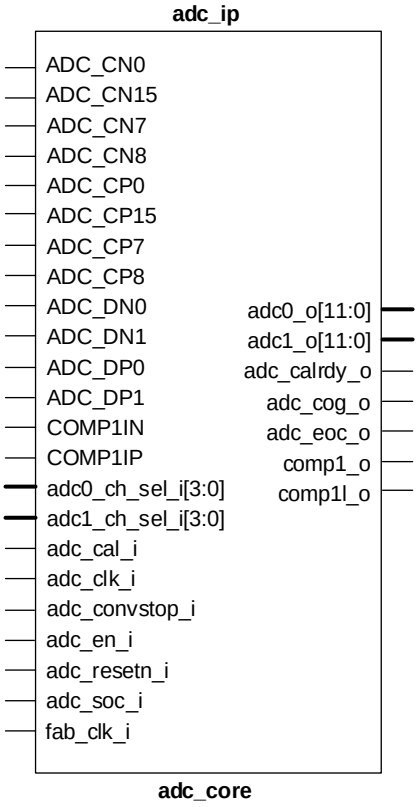


Figure 4.1. ADC Core Module IP Port

Note: Figure 4.1 illustrates only the enabled feature based on IP configuration. For more details about the ports and functions supported, refer to the subsequent subsections.

4.1. ADC Core Module IP Interface

Table 4.1 shows the ports in the ADC Core Module IP. lists the top-level input and output signals for ADC Core Module. The port names in upper-case represent signals that connect to physical pins of the device and port names in lower-case represent internal signals.

Table 4.1. IP Ports

Port	Type	Description
Clocks and Reset		
adc_clk_i	Input	The clock source is used for ADC sampling. This clock source must come from the Lower Right Corner (LRC) PLL secondary output clock 4 (clkos4_o). Maximum frequency allowed = 50 MHz.
fab_clk_i	Input	This clock source synchronizes the ADC output and Comparator latches at fabric. Maximum frequency allowed is subject to core fabric Fmax.
adc_resetn_i	Input	Asynchronous reset for ADC block and comparator latches. Assert to release the reset. When this reset signal deasserted, the ADC block and comparator latches register are in reset mode until this signal is released. This signal also returns the ADC block to uncalibrated state. Recalibrate the ADC block once the reset is released.
Control and Status		
adc_en_i	Input	Asynchronous control to enable or disable the ADC block. Assert this port to enable the ADC block. When this signal is deasserted, the ADC block is in shutdown mode without impacting the comparator channels.
adc_cal_i	Input	Asserts to enable ADC internal calibration. Hold for at least one adc_clk_i (divided clk) clock cycle to initiate the ADC internal calibration.
adc0_ch_sel_i[3:0]	Input	ADC0 input channel selection. Refer to Table 4.2 for channel mapping.
adc1_ch_sel_i[3:0]	Input	ADC1 input channel selection. Refer to Table 4.2 for channel mapping.
adc_calrdy_o	Output	Indicates ADC internal calibration status. The output is asserted when the calibration is complete.
adc_soc_i	Input	Start of conversion (SOC). This signal is acquired on the failing edge of the adc_clk (divided clk).
adc_cog_o	Output	Indicates conversion on-going (COG).
adc_convstop_i	Input	Stop on-going conversion control. This signal is used to abort an ongoing conversion.
adc_eoc_o	Output	Indicates end of conversion (EOC).
Analog input		
ADC_DP/N[1:0]	Input	Dedicated ADC input pin for ADC0 and ADC1
ADC_CP/N[15:0]	Input	Dual purpose ADC input pin. If the pin is not used for ADC, this pin can be used for GPIO.
COMP[3:1]IP/N	Input	Dedicated comparators input pin.
Data output		
adc0_o[11:0]	Output	12-bits ADC0 data output.
adc1_o[11:0]	Output	12-bits ADC1 data output.
comp[3..1]_o	Output	Continuous data output from comparator 1, 2, and 3.
comp[3..1]l_o	Output	Latched data output from comparator 1, 2, and 3.

Table 4.2 shows the channel select mapped with the actual ADC channel.

Table 4.2. Channel Select Mapping

ADC0_CH_SEL_i[3:0] (Hex)	ADC0	ADC1_CH_SEL_i[3:0] (Hex)	ADC1
0	ADC_CP0/ADC_CN0	0	ADC_CP8/ADC_CN8
1	ADC_CP1/ADC_CN1	1	ADC_CP9/ADC_CN9
2	ADC_CP2/ADC_CN2	2	ADC_CP10/ADC_CN10
3	ADC_CP3/ADC_CN3	3	ADC_CP11/ADC_CN11
4	ADC_CP4/ADC_CN4	4	ADC_CP12/ADC_CN12
5	ADC_CP5/ADC_CN5	5	ADC_CP13/ADC_CN13

ADC0_CH_SEL_i[3:0] (Hex)	ADC0	ADC1_CH_SEL_i[3:0] (Hex)	ADC1
6	ADC_CP6/ADC_CN6	6	ADC_CP14/ADC_CN14
7	ADC_CP7/ADC_CN7	7	ADC_CP15/ADC_CN15
8	VCCM	8	VCCIO1
9	VCC	9	VCCIO0
A	VCCAUX	A	DTR
B	ADC_DP0/ADC_DN0	B	ADC_DP1/ADC_DP1
C	Reserved	C	Reserved
D		D	
E		E	
F		F	

4.2. ADC Core Module IP Port Name Update

Port names in the ADC core module IP are updated starting from the Lattice Radiant software version 2022.1 onwards. When you migrate your design from older Lattice Radiant software, regenerate the ADC Core Module IP and update the port name at the top-level design file.

Table 4.3. Port Name Update

Lattice Radiant Software Version 3.2	Lattice Radiant Software Version 2022.1 Onwards
ADC_CP1/ADC_CN1	ADC_CP0/ADC_CN0
ADC_CP2/ADC_CN2	ADC_CP1/ADC_CN1
ADC_CP3/ADC_CN3	ADC_CP2/ADC_CN2
ADC_CP4/ADC_CN4	ADC_CP3/ADC_CN3
ADC_CP5/ADC_CN5	ADC_CP4/ADC_CN4
ADC_CP6/ADC_CN6	ADC_CP5/ADC_CN5
ADC_CP7/ADC_CN7	ADC_CP6/ADC_CN6
ADC_CP8/ADC_CN8	ADC_CP7/ADC_CN7
ADC_CP9/ADC_CN9	ADC_CP8/ADC_CN8
ADC_CP10/ADC_CN10	ADC_CP9/ADC_CN9
ADC_CP11/ADC_CN11	ADC_CP10/ADC_CN10
ADC_CP12/ADC_CN12	ADC_CP11/ADC_CN11
ADC_CP13/ADC_CN13	ADC_CP12/ADC_CN12
ADC_CP14/ADC_CN14	ADC_CP13/ADC_CN13
ADC_CP15/ADC_CN15	ADC_CP14/ADC_CN14
ADC_CP16/ADC_CN16	ADC_CP15/ADC_CN15

5. Example Design

For the example design, refer to the ADC Demo Design on CertusPro-NX Versa Board web page.

6. Designing with the IP

For instructions on generating and using the ADC IP, as well as debugging tips, refer to the ADC Demo Design on CertusPro-NX Versa Board web page.

7. Design Considerations

Before implementing the ADC design, go through the following areas in these documents:

- ADC User Guide for Nexus Platform (FPGA-TN-02129)
  - Architecture of the Nexus ADC
  - Design guidelines in the user guide
  - Supported feature per package in Appendix B
- Respective device Hardware Checklist
  - Power rail design requirements including the filtering circuitry
  - Analog input pin board design including how to handle unused analog input pins
  - External reference voltage pin board design requirements

These documents provide more information about the Nexus ADC architecture, feature supported, and board design considerations.

Appendix A. Resource Utilization

Table A.1 shows a sample resource utilization of the ADC Core Module IP in LFCPNX-100-9LFG672C device.

Table A.1. Resource Utilization

IP Configuration	Slices	LUTs	Registers	EBR	PLL
All ADC channels are enabled, PLL is connected, ADC conversion format is straight binary, external reference voltage mode, without ADC reset and initialization control design	678 / 79872	895 / 79872	678 / 80769	13 / 208	1 / 4

References

- ADC User Guide for Nexus Platform (FPGA-TN-02129)
- ADC specifications:
  - CertusPro-NX Family Data Sheet (FPGA-DS-02086)
  - Certus-NX Family Data Sheet (FPGA-DS-02078)
  - CrossLink-NX Family Data Sheet (FPGA-DS-02049)
  - MachXO5-NX Family Data Sheet (FPGA-DS-02102)
- Device Hardware Checklist:
  - CertusPro-NX Hardware Checklist (FPGA-TN-02255)
  - Certus-NX Hardware Checklist (FPGA-TN-02151)
  - CrossLink-NX Hardware Checklist (FPGA-TN-02149)
  - MachXO5-NX Hardware Checklist (FPGA-TN-02274)
- ADC Demo Design on CertusPro-NX Versa Board User Guide (FPGA-EB-02061)
- ADC Sequencer Module User Guide (FPGA-IPUG-02062)
- Certus-NX web page
- CertusPro-NX web page
- CrossLink-NX web page
- MachXO5-NX web page
- ADC Demo Design on CertusPro-NX Versa Board web page
- Lattice Radiant Software web page
- Lattice Insights for Lattice Semiconductor training courses and learning plans

Technical Support Assistance

Submit a technical support case through www.latticesemi.com/techsupport.

For frequently asked questions, refer to the Lattice Answer Database at www.latticesemi.com/Support/AnswerDatabase.

Revision History

Revision 1.5, January 2025

Section	Change Summary
All	Removed the *Debugging* section.
Introduction	Updated *VCCAux* to *VCCAUX* in the Features section.

Functional Description	• Updated the descriptions in the IP Architecture Overview section.
	• Updated the caption to Figure 2.1. Lattice ADC Core Module HIP.
	• Update Figure 2.2. ADC Implementation Clock Domain Block Diagram.
	• In the Reset Overview section:
	  • Updated the steps to reset and initialize the ADC design.
	  • Updated the *comparator function* description.
	  • Updated the *Notes* section.
	• Updated Figure 2.3. ADC Reset and Calibration Sequence.
	• Added Figure 2.4. ADC Calibration Completion and Conversion Starting.
	• Updated the ADC Core Module IP Implementation section.

IP Parameter Description	• Updated the descriptions for the *ADC0 Reference Voltage Select* and *ADC1 Reference Voltage Select* attributes in Table 3.1. General Attributes. • Replaced *Parameter* with *Configuration* in the IP Configuration Settings for Example Use Cases section title and content. • Removed the *Calculating the Maximum Sampling Rate* section. • Added the 3.4.1 and 3.4.2 sections. • Updated the Calculating the Sampling Rate for Each ADC Channel section. • Updated the internal reference voltage accuracy from *±5%* to *±15%* in the Reference Voltage Selection section.
Signal Description	• Updated *ADC0* and *ADC1* for the *C*, *D*, *E*, and *F* channels in Table 4.2. Channel Select Mapping.
Example Design	Updated this section.
Designing with the IP	Updated this section.
References	Added the *ADC Demo Design on CertusPro-NX Versa Board* web page.

Revision 1.4, February 2024

Section	Change Summary
All	Renamed document from *ADC Core Module - Lattice Radiant Software* to *ADC Core Module*.
Disclaimers	Updated disclaimers.
Inclusive Language	Added inclusive language boilerplate.
Introduction	• Reworked section contents. • Reworked subsection 2.5 *Licensing the IP* and renamed to subsection 1.4 Licensing and Ordering Information. • Reworked subsection 2.7 *ADC Core Validation* and subsection 2.8 *Hardware Evaluation* and renamed to subsection 1.5 IP Validation Summary. • Added subsection 1.6 Minimum Device Requirements. • Reworked subsection 1.3 *Conventions* and renamed to subsection 1.7 Naming Conventions.

Functional Description	• Reworked subsection 2.1 *Overview* and renamed to subsection 2.1 IP Architecture Overview.
	• Added the following subsections:
	  • 2.2 Clocking
	  • 2.3 Reset
	  • 2.4 User Interfaces
	  • 2.5 ADC Core Module IP Implementation

Section	Change Summary

IP Parameter Description	• Reworked subsection 2.3 *Attribute Summary* and renamed to the following subsections:
	  • 3.1 General Tab
	  • 3.2 Comparator Tab
	  • 3.3 ADC Channels Tab
	• Added subsection 3.4 IP Parameter Settings for Example Use Cases.

Signal Description	• Reworked subsection 2.2 *Signal Description* and renamed to subsection 4.1 ADC Core Module IP Interface. • Reworked Appendix B *Migration* and renamed to subsection 4.2 ADC Core Module IP.
Example Design	Added this section.
Designing with the IP	Added this section.
Debugging	Added this section.
Design Considerations	Added this section.
Resource Utilization	Reworked Appendix A *Limitations* and renamed to Appendix A Resource Utilization.
References	Updated references.

Revision 1.3, March 2023

Section	Change Summary
Functional Description	• Updated Figure 2.3. ADC Core Module ADC1 Block. • Added contents to the ADC0 Reference Voltage Select/ADC1 Reference Voltage Select section. • Updated the Internal Reference Voltage Generation Block section to add that VCCIO0, VCCIO1, and VCCAUX need to operate at 2.5 V or less when the internal voltage reference is used.
All	Minor adjustments in formatting and style.

Revision 1.2, November 2022

Section	Change Summary
Functional Description	• Updated Figure 2.2. ADC Core Module ADC0 Block and Figure 2.3. ADC Core Module ADC1 Block. • Updated Table 2.1. ADC Core Module Ports. • Updated the Generating and Synthesizing the IP section from *Open IP Catalog (as shown in Figure 2.12) then double-click on the ADC in the Architectural_Modules* to *Create a new Lattice Radiant software project or open an existing project*.
Appendix A. Limitations	Added LFMXO5-25 MachXO5-NX device information in Table A.1. ADC Core Module Features versus Device and Package.
Appendix B. Migration	Added Migration section.
Technical Support Assistance	Added reference to the Lattice Answer Database on the Lattice website.

Revision 1.1, August 2022

Section	Change Summary
Functional Description	Updated Figure 2.2. ADC Core Module ADC0 Block exchanging VCC and VCCM positions.

Revision 1.0, December 2021

Section	Change Summary
All	Initial release

www.latticesemi.com