Intel® FPGA Voltage Sensor IP Core User Guide

Updated for Intel® Quartus® Prime Design Suite: **17.1**


Online Version


Send Feedback

UG-01167

ID: **683781**

Version: **2018.02.09**

Contents

1. Intel® FPGA Voltage Sensor IP Core Overview

The Intel® FPGA Voltage Sensor IP core monitors critical on-chip power supplies and external analog voltage.

Table 1. Release and IP Core Information

Item		
Release Information	Version	17.1
	Release	November 2017
IP Core Information	Core Features	Monitors the following: • Critical on-chip power supplies • External analog voltage Supports access from the following: • JTAG access • FPGA core access
	Device Family	Supports Intel Intel Arria® 10 and Intel Cyclone® 10 GX devices
	Device Tools	Intel Quartus® Prime software

2. Intel FPGA Voltage Sensor IP Core Getting Started

The Intel Quartus Prime software includes installation of the Intel FPGA Voltage Sensor IP core.

Related Information

Introduction to Intel FPGA IP Cores
Provides general information about all Intel FPGA IP cores, including parameterizing, generating, upgrading, and simulating IP cores.

2.1. Specifying Parameters and Options

Follow these steps to instantiate the Intel FPGA Voltage Sensor IP core parameters and options.

1. Create an Intel Quartus Prime project using the **New Project Wizard** available from the File menu.
2. On the **Tools** menu, click **IP Catalog**.
3. Under **Installed IP**, double-click **Library> Basic Functions> Configuration and Programming>Intel FPGA Voltage Sensor**.
4. Specify an entity name for the custom IP variation and select the targeted Intel FPGA device family. Click **OK**.
5. In the Intel FPGA Voltage Sensor parameter editor, specify the core variant and the memory type for your application.
6. Click **Generate HDL**.
7. Click **Generate** to generate the IP core and supporting files, including simulation models.
8. Click **Close** when file generation completes.
9. Click **Finish**.

2.2. Intel FPGA Voltage Sensor Parameters

You can use the GUI parameters to configure the Intel FPGA Voltage Sensor IP core.

Table 2. Intel FPGA Voltage Sensor Parameters

Parameters	Description
Core Variant	There are two configuration variants of the Intel FPGA Voltage Sensor IP core: • Voltage sensor controller with Avalon-MM sample storage • Voltage sensor controller with external sample storage

continued...

Parameters	Description
	Select the core variant that meets your requirement. For more information, refer to the related information.
Memory Type	Select the memory type that you will use to store the voltage samples—on-chip memory or register.

Related Information

- Voltage Sensor Controller Core on page 6
- Sample Storage Core on page 7

3. Intel FPGA Voltage Sensor IP Core Functional Description

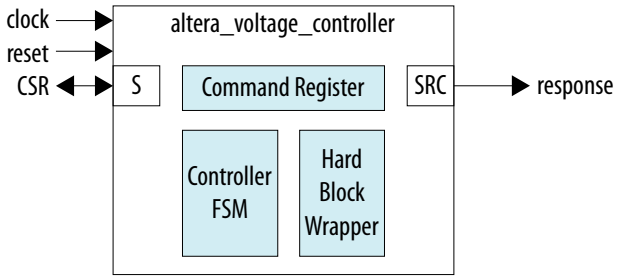
The Intel FPGA Voltage Sensor IP core consists of the following:

- Voltage Controller with Avalon-MM Sample Storage
- Voltage Controller with External Sample Storage

3.1. Voltage Sensor Controller Core

The voltage sensor controller core contains a command register and conversion sequence data. You can use the command register to configure your intended conversion mode. This core also contains control logic which communicates with the voltage sensor hard IP block. You can access the register through the Avalon Memory-Mapped (Avalon-MM) slave interface. This core uses the Avalon Streaming (Avalon-ST) interface to send responses.

Figure 1. Voltage Sensor Controller Block Diagram



This voltage sensor controller core receives commands through the Avalon-MM slave control and status register (CSR) interface. The command includes mode and sequences. This core decodes the command and drives the signals that are connected to the voltage sensor controller core accordingly.

The voltage sensor controller core supports the following sequences. You can select the channels by using the following mode (MD) bits setting.

Table 3. Channels Selection

MD1	MD0	Channel	Channel Mapping
0	0	Channel 2	V_{CC}
		Channel 3	V_{CCP}
		Channel 4	V_{CCPT}
		Channel 5	V_{CCERAM}

continued...

MD1	MD0	Channel	Channel Mapping
		Channel 6	V_{CCL_HPS}
		Channel 7	ADCGND
0	1	Channel 0	VSIG[P,N]_0
		Channel 1	VSIG[P,N]_1
		Channel 2	V_{CC}
		Channel 3	V_{CCP}
		Channel 4	V_{CCPT}
		Channel 5	V_{CCERAM}
		Channel 6	V_{CCL_HPS}
		Channel 7	ADCGND
1	0	Channel 0	VSIG[P,N]_0
		Channel 1	VSIG[P,N]_1
1	1	User select mode	Defined by user

This core allows you to monitor separate channels. You can configure the sequences during run time.

3.1.1. Conversion Modes

The voltage sensor controller core supports two modes of conversion that you can control using the command register.

3.1.1.1. One Round Conversion

When you set the RUN bit, conversion starts with the defined sequence (based on the setting of the MD bits) and stops when a conversion is complete. The hardware automatically clears the RUN bit.

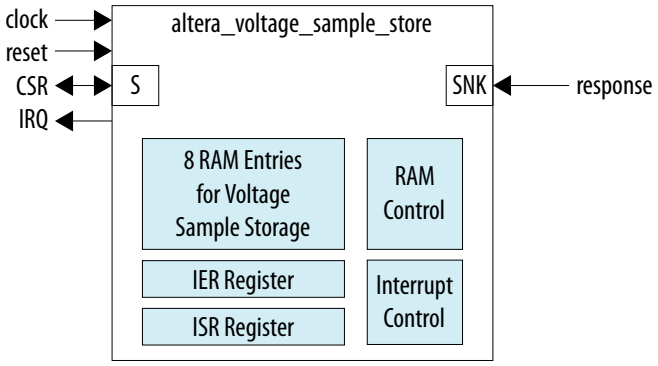
3.1.1.2. Continuous Conversion

When you set the RUN bit, conversion starts with the defined sequence (based on the setting of the MD bits) and when a conversion is complete, the conversion repeats the same set of conversion again. For example, if you choose the sequence for Channel 0 to Channel 7, the IP block restarts the whole sequence when the Channel 7 sample is complete. For a single channel read (MD = 2'b11), the IP block reads the value from that channel until the RUN bit is cleared. In this continuous conversion mode, the software clears the RUN bit.

3.2. Sample Storage Core

The sample storage core stores voltage samples. The control core passes the voltage samples to this core through the Avalon-ST interface. The on-chip RAM stores the voltage samples and you can retrieve them through the Avalon-MM slave. This core provides an option to generate an interrupt when it retrieves a block of voltage samples using one full round of the conversion sequence.

Figure 2. Sample Storage Core Block Diagram



You can parameterize the internal RAM as an on-chip memory or a register. For the memory type selection, it depends on your design resource utilization. For example, if your design uses many registers but lesser usage of the on-chip memory, you can write the voltage sensor data into the on-chip memory.

The core stores the sample on per slot basis instead of per channel basis. The sample storage core asserts interrupt request (IRQ) when it receives a complete block of samples. You can disable the IRQ assertion at run time. If you disable the IRQ assertion, the software must poll to know when the core receives a complete block of samples.

3.3. Configuring the Voltage Sensor Controller

You can configure the voltage sensor controller to support high or low performance:

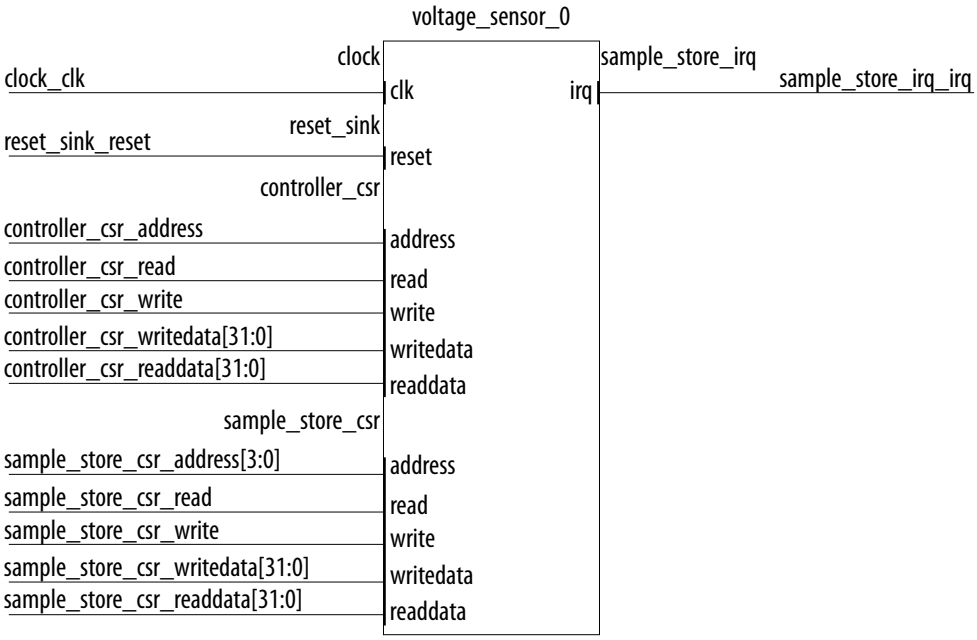
- Configure the voltage sensor controller with Avalon-MM sample storage—to support low performance systems and to enable quicker hook up in the Platform Designer system to allow easy reading of the sample storage data.
- Configure the voltage sensor controller with external sample storage—for high performance data streaming.

Related Information

Avalon Memory-Mapped Interfaces
Provides more information about the voltage sensor in the Intel Arria 10 devices.

3.3.1. Configuring the Voltage Sensor Controller with Avalon-MM Sample Storage

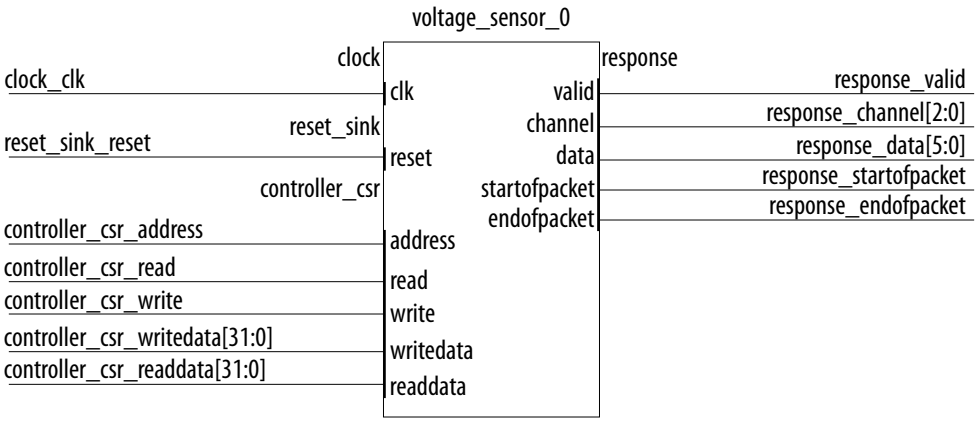
Figure 3. Voltage Sensor Controller with Avalon-MM Sample Storage



This configuration mode allows you to use the voltage sensor controller with internal on-chip RAM to store samples. The host sends one command and waits until the storage sends interrupt to read the data. In this configuration mode, the controller captures a block of voltage sample data and stores them inside an on-chip RAM. The host processor retrieves the data before triggering another request.

3.3.2. Configuring the Voltage Sensor Controller with External Sample Storage

Figure 4. Voltage Sensor Controller with External Sample Storage



This configuration mode allows you to use the voltage sensor controller while processing or storing the voltage samples through Avalon-ST samples.

4. Intel FPGA Voltage Sensor IP Core Interface Signals

4.1. CSR Interface

The interface type is Avalon-MM slave.

Table 4. CSR Interface

Signal	Width	Description
sample_store_csr_address	1 or 4	Avalon-MM address bus. The address bus is Word addressing. • The address width for altera_voltage_sensor_sample_store is 4. • The address width for altera_voltage_sensor_controller is 1.
sample_store_csr_read	1	Avalon-MM read request.
sample_store_csr_write	1	Avalon-MM write request.
sample_store_csr_writedata[31:0]	32	Avalon-MM write data bus.
sample_store_csr_readdata[31:0]	32	Avalon-MM read data bus.

4.2. Response Interface

The interface type is Avalon-ST.

Table 5. Response Interface

Signal	Width	Description
response_valid	1	Indicates from the source port that current data is valid.
response_channel[2:0]	3	Indicates which channel the voltage sample data corresponds to. • Bits 16:8 — not used. • Bits 7:0 — Channel 7 to Channel 0.
response_data[5:0]	6	Voltage sample data.
response_startofpacket	1	When asserted, indicates the start of a packet.
response_endofpacket	1	When asserted, indicates the last cycle of a packet.

Related Information

- Voltage Sensor
  Provides more information about the voltage sensor in the Intel Arria 10 devices.

- Voltage Sensor
  Provides more information about the voltage sensor in the Intel Cyclone 10 GX devices.

4.3. Interrupt Interface

The interface type is interrupt.

Table 6. Interrupt Interface

Signal	Width	Description
sample_store_irq_irq	1	Interrupt request.

5. Intel FPGA Voltage Sensor IP Core Registers

5.1. Voltage Controller Core Registers

Table 7. Controller Core Registers

Offset	Register Name	Bits	Bit Name	RO/RW	Description	Reset Value
0x0	Command	13..31	Reserved	—	Reserved.	0x0
		12	CAL	RW	Determines whether the output data is using a calibrated or a non-calibrated value. • 0 specifies non-calibrated value • 1 specifies calibrated value You must not use this bit to enable calibration.	0x0
		11	BU1	RW	Unipolar selection for Channel 0 or 1.(1) • 0 specifies unipolar selection	0x0
		10	BU0	RW		0x0
		9	MD1	RW	Mode select for channel sequencer. • MD[1:0] = 2'b00 specifies channel sequencer cycles from Channel 2 to Channel 7 • MD[1:0] = 2'b01 specifies channel sequencer cycles from Channel 0 to Channel 7 • MD[1:0] = 2'b10 specifies channel sequencer cycles from Channel 0 to Channel 1 • MD[1:0] = 2'b11 specifies channel in CHSEL to be converted	0x0
		8	MD0	RW		0x0
		7	Reserved	—	Reserved.	0x0
		4:6	CHSEL	RW	Specifies the channel to be converted and used when MD[1:0] = 2'b11.	0x0

continued...

(1) Only Unipolar mode is supported.

Offset	Register Name	Bits	Bit Name	RO/RW	Description	Reset Value
		3	Reserved	—	Reserved.	0x0
		1:2	MODE	RW	Specifies the controller core's mode of operation. • MD[1:0] = 2'b11 to 2'b10 are reserved • MD[1:0] = 2'b01 specifies continuous voltage sensor conversion • MD[1:0] = 2'b00 specifies one round of voltage sensor conversion Do not write to this address when the RUN bit is set. You must wait for the hardware to clear these bits before updating this address.	0x0
		0	RUN	RW	Control bit to trigger the sequencer core operation. • 1 specifies run • 0 specifies stop When the Intel Quartus Prime software writes a 0 to this address, the controller core completes its current operation and clears this field.	0x0

5.2. Sample Store Core Registers

Table 8. Sample Store Core Registers

Offset	Register Name	Bits		RO/RW	Description	Reset Value
		6:31	0:5			
0x0	Voltage Sample	Reserved	SAMPLE	RO	Values correspond to Channel 0 for Offset 0x0 and Channel 7 for Offset 0x7 (based on the MD[0:1] setting). For more information, refer to the Controller Core Registers table.	0x0
0x1						
0x2						
0x3						
0x4						
0x5						
0x6						
0x7						

Offset	Register Name	Bits	Field	RO/RW	Description	Reset Value
0x8	Interrupt Enable	1:31	Reserved	RO	Reserved	0x0
		0	M_EOP	RW	Enable bit for end-of-packet interrupt. • 1 specifies the corresponding interrupt is enabled • 0 specifies the corresponding interrupt is disabled	0x1
0x9	Interrupt Status	1:31	Reserved	RO	Reserved	0x0
		0	EOP	RW	End-of-packet interrupt. • 1 specifies a complete block of samples is received • Writing a 1 to this address clears this bit	0x1

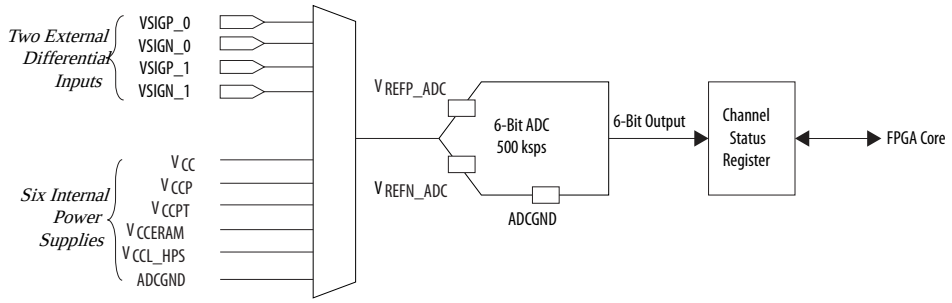
Related Information

Voltage Controller Core Registers on page 13

6. Intel FPGA Voltage Sensor IP Core Implementation Guide

Intel Arria 10 and Intel Cyclone 10 GX support an on-chip voltage sensor. The voltage sensor provides a 6-bit digital representation of the analog signal being observed. The voltage sensor monitors two external differential inputs and six internal power supplies as shown in the following figure. To get the analog-to-digital converter (ADC) input, the V_{CCPT} voltage value is divided by two. To get the actual V_{CCPT} voltage value, multiply the ADC output by two.

Figure 5. Voltage Sensor

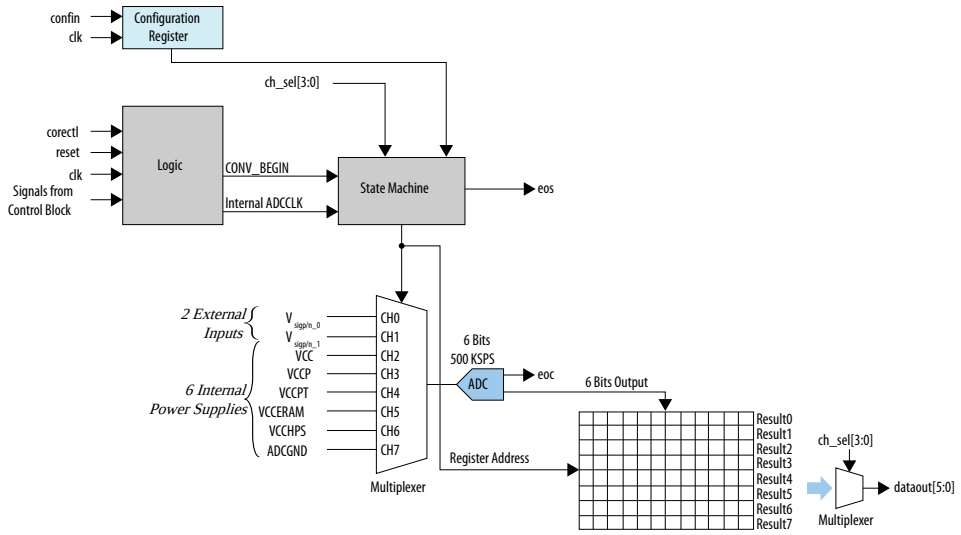


The conversion speed of the ADC is 500 kilosymbols per second (ksps) cumulative. When you specify multiple channels, the speed per channel is reduced accordingly.

6.1. Using Voltage Sensor in Intel Arria 10 and Intel Cyclone 10 GX Devices

You can use the voltage sensor feature to monitor critical on-chip power supplies and external analog voltages. The voltage sensor block for the Intel Arria 10 and Intel Cyclone 10 GX devices supports access from the FPGA core. The following sections describe the flow in using the voltage sensor for the Intel Arria 10 and Intel Cyclone 10 GX devices.

Figure 6. Voltage Sensor Components



The conversion speed of the ADC is 500 ksps cumulative. When multiple channels are used, the speed per channel is reduced accordingly.

Note: VREFP_ADC pins consume very little current, leakage current accounts for most of the current drawn, which is less than 10 µA. For VREFN_ADC pins, the current is less than 0.1 mA.

For better ADC performance, connect VREFP_ADC and VREFN_ADC pins to an external 1.25 V accurate reference source (±0.2%). An on-chip reference source (±10%) is activated by connecting the VREFP_ADC pin to GND. Treating VREFN_ADC as an analog signal together with the VREFP_ADC signal provides a differential 1.25 V voltage.

Connect both VREFP_ADC and VREFN_ADC pins to GND if no external reference is supplied.

6.1.1. Accessing the Voltage Sensor Using FPGA Core Access

In user mode, you can implement a soft IP to access the voltage sensor block. To access the voltage sensor block from the core fabric, include the following WYSIWYG atom in your Intel Quartus Prime project:

Example 1. WYSIWYG Atom to Access the Voltage Sensor Block

```
twentynm_vsblock<name>
(
    .clk (<input>, clock signal from core),
    .reset(<input>, reset signal from core),
    .corectl(<input>, core enable signal from core),
    .coreconfig(<input>, config signal from core),
    .confin(<input>, config data signal from core),
    .chsel(<input>, 4 bits channel selection signal from core),
    .eoc(<output>, end of conversion signal from vsblock),
    .eos(<output>, end of sequence signal from vsblock),
    .dataout(<output>, 12 bits data out of vsblock)
);
```

Table 9. Description for the Voltage Sensor Block WYSIWYG

Port Name	Type	Description
clk	Input	Clock signal from the core. The voltage sensor supports up to an 11-MHz clock.
reset	Input	Active high reset signal. An asynchronous high-to-low transition on the reset signal starts voltage sensor conversion. All registers are cleared and the internal voltage sensor clock is gated off when the reset signal is high.
corectl	Input	Active high signal. "1" indicates the voltage sensor is enabled for core access. "0" indicates the voltage sensor is disabled for core access.
coreconfig	Input	Serial configuration signal. Active high.
confin	Input	Serial input data from the core to configure the configuration register. The configuration register for the core access mode is 8 bits wide. The LSB is the first bit shifted in.
chsel[3:0]	Input	4-bit channel address. Specifies the channel to be converted.
eoc	Output	Indicates the end of the conversion. This signal is asserted after the conversion of each channel data packet.
eos	Output	Indicates the end of sequence. This signal is asserted for one cycle after the completion of the conversion of the selected sequence.
dataout[11:0]	Output	• dataout[11:6]—6-bit output data. • dataout[5:0]—Reserved.

6.1.1.1. Configuration Registers for the Core Access Mode

The core access configuration register is an 8-bit register.

Figure 7. Core Access Configuration Register

D7	D6	D5	D4	D3	D2	D1	D0
NA	CAL	NA	NA	BU1	BU0	MD1	MD0

Table 10. Description for the Core Access Configuration Register

Bit Number	Bit Name	Description
D0	MD0	Mode select for channel sequencer: • MD[1:0]=2'b00—channel sequencer cycles from channel 2 to channel 7 • MD[1:0]=2'b01—channel sequencer cycles from channel 0 to channel 7 • MD[1:0]=2'b10—channel sequencer cycles from channel 0 to channel 1 • MD[1:0]=2'b11—controlled by IP core. Specify the channel to be converted on chsel[3:0].
D1	MD1	
D2	BU0	Channel 0—Register bit that indicates channel 0. Set to "0".
D3	BU1	Channel 1—Register bit that indicates channel 1. Set to "0".
D4	NA	Reserved. Set to "0".

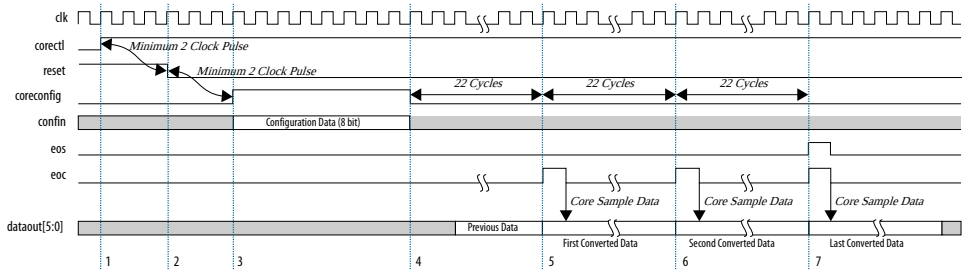
continued...

Bit Number	Bit Name	Description
D5	NA	Reserved. Set to "0".
D6	CAL	Calibration enable bit. "0" indicates calibration is off. "1" indicates calibration is on. The calibration result does not include the final 12-bit converted data when calibration is off.
D7	NA	Reserved. Set to "0".

6.1.1.1.1. Accessing the Voltage Sensor in the Core Access Mode when MD[1:0] is not Equal to 2'b11

The following timing diagram shows the IP core timing to access the voltage sensor when MD[1:0] is not equal to 2'b11.

Figure 8. Timing Diagram when MD[1:0] is not Equal to 2'b11



1. A low-to-high transition on the `corectl` signal enables the core access mode.

   Wait for a minimum of two clock cycles before proceeding to step 2.
2. De-asserting the `reset` signal releases the voltage sensor from the reset state.

   Wait for a minimum two clock cycles before proceeding to step 3.
3. Configure the voltage sensor by writing the configuration registers and asserting the `coreconfig` signal for eight clock cycles. The configuration register access mode is 8 bits and configuration data is shifted in serially.
4. The `coreconfig` signal going low indicates the start of the conversion based on the configuration defined in the configuration register.
5. Poll the `eoc` and `eos` status signals to check if conversion for the first channel defined by MD[1:0] is complete. Latch the output data on the `dataout[5:0]` signal at the falling edge of the `eoc` signal.
6. Poll the `eoc` and `eos` status signals to check if conversion for the subsequent channels defined by MD[1:0] are complete. Latch the output data on the `dataout[5:0]` signal at the falling edge of the `eoc` signal.
7. Repeat step 6 until the `eos` signal is asserted, indicating the completion of the conversion of one cycle on the channels specified by MD[1:0].
   a. Both the `eoc` and `eos` signals are asserted on the same clock cycle when the voltage sensor completes the conversion for the last channel.

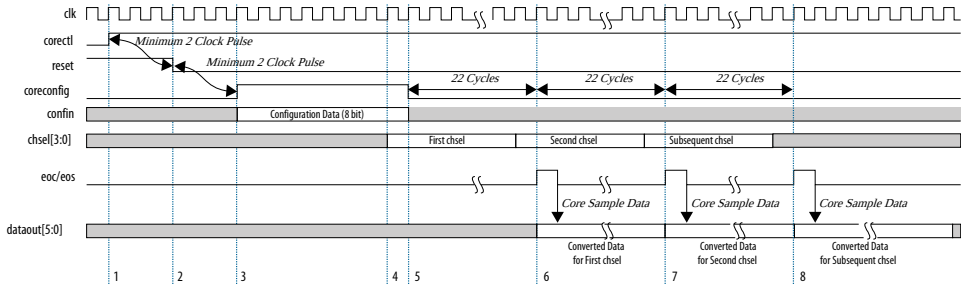
b. You can only interrupt the operation of the voltage sensor by writing into the configuration register after one cycle of the eos signal completes.

8. When the sequence completes, and if the corectl and reset signals remain unchanged, the conversion repeats the same sequence again until corectl is 0 and reset is 1. If you want to measure other sequences, repeat step 2 to step 7.

6.1.1.1.2. Accessing the Voltage Sensor in the Core Access Mode when MD[1:0] is Equal to 2'b11

The following timing diagram shows the IP core timing to access the voltage sensor when MD[1:0] is equal to 2'b11.

Figure 9. Timing Diagram when MD[1:0] is Equal to 2'b11



1. A low-to-high transition on the corectl signal enables the core access mode.

   Wait for a minimum of two clock cycles before proceeding to step 2.

2. De-asserting the reset signal releases the voltage sensor from the reset state.

   Wait for a minimum two clock cycles before proceeding to step 3.

3. Configure the voltage sensor by writing the configuration registers and asserting the coreconfig signal for eight clock cycles. The configuration register access mode is 8 bits and configuration data is shifted in serially.

4. Specify the channel for conversion on the chsel[3:0] signal. Data on the chsel[3:0] signal must be stable before the coreconfig signal is de-asserted.

5. The coreconfig signal going low indicates the start of the conversion based on the configuration defined in the configuration register and the chsel[3:0] signal.

6. Specify the next channel for conversion on the chsel[3:0] signal. Data on the chsel[3:0] signal must be stable one cycle before the eoc signal asserts. Poll the eoc and eos status signals to check if conversion for the first channel defined by the chsel[3:0] signal in step 4 is complete. Latch the output data on the dataout[5:0] signal at the falling edge of the eoc signal.

7. Repeat step 6 for all the subsequent channels.

6.1.2. Voltage Sensor Transfer Function

The following figure shows the voltage sensor transfer function for the unipolar mode.

Figure 10. Voltage Sensor Transfer Function for the Unipolar Mode

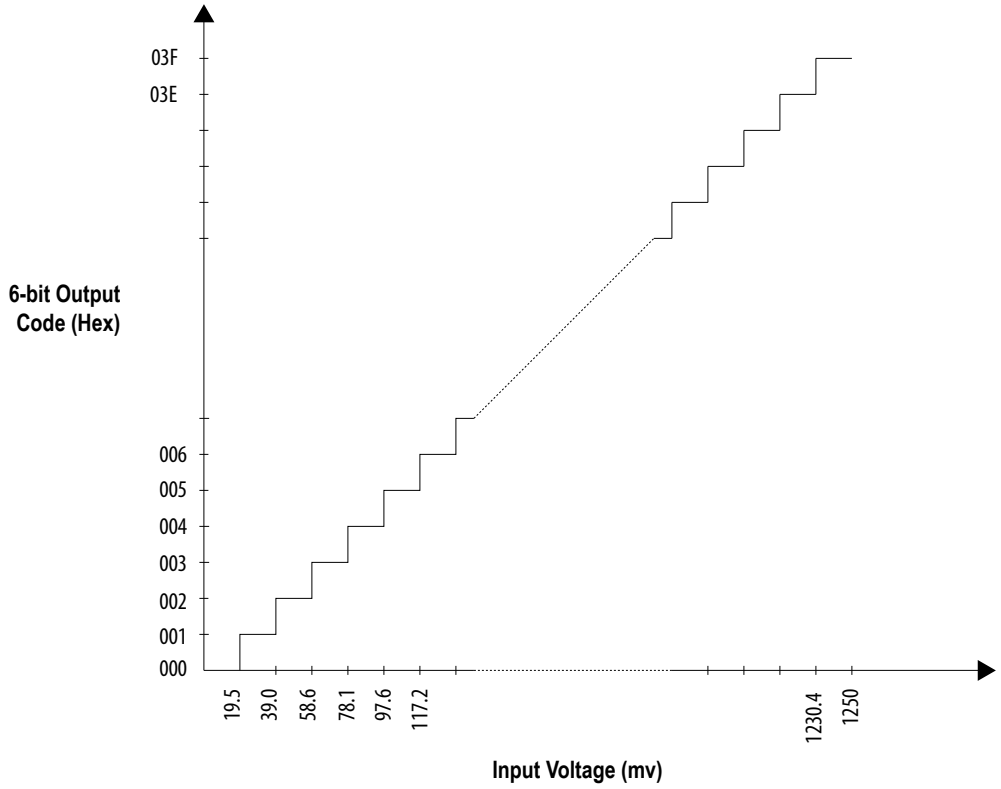


Table 11. Analog Input Conversion

6-Bit Output Code (Hex)	Input Voltage (mV)
000	0.000
001	19.531
002	39.063
003	58.594
004	78.125
005	97.656
006	117.188
007	136.719
008	156.250
009	175.781
00A	195.313
00B	214.844
00C	234.375
00D	253.906
00E	273.438

continued...

6-Bit Output Code (Hex)	Input Voltage (mV)
00F	292.969
010	312.500
011	332.031
012	351.563
013	371.094
014	390.625
015	410.156
016	429.688
017	449.219
018	468.750
019	488.281
01A	507.813
01B	527.344
01C	546.875
01D	566.406
01E	585.938
01F	605.469
020	625.000
021	644.531
022	664.063
023	683.594
024	703.125
025	722.656
026	742.188
027	761.719
028	781.250
029	800.781
02A	820.313
02B	839.844
02C	859.375
02D	878.906
02E	898.438
02F	917.969
030	937.500
031	957.031
032	976.563

continued...

6-Bit Output Code (Hex)	Input Voltage (mV)
033	996.094
034	1015.625
035	1035.156
036	1054.688
037	1074.219
038	1093.750
039	1113.281
03A	1132.813
03B	1152.344
03C	1171.875
03D	1191.406
03E	1210.938
03F	1230.469

7. Document Revision History for Intel FPGA Voltage Sensor IP Core User Guide

Date	Version	Changes
February 2018	2018.02.09	• Rebranded as Intel. • Beginning from the Intel Quartus Prime software version 17.1, the name of this IP core has been changed from Altera Voltage Sensor IP Core to Intel FPGA Voltage Sensor IP Core. • Added Intel Cyclone 10 GX device information. • Added the Intel FPGA Voltage Sensor IP Core Implementation Guide chapter. • Added the supported configuration variants in the Intel FPGA Voltage Sensor Parameters table. • Added the Channel Selection table. • Added a note to indicate only unipolar mode is supported in the unipolar selection bits of the Controller Core Registers table. • Updated the signal names in the CSR Interface table. • Updated the signal names in the Response Interface table. • Updated the signal name in the Interrupt Interface table. • Updated the Bit Name header column in the Controller Core Registers table. • Updated the voltage sample register in the Sample Store Core Registers table. • Updated the Sample Storage Core Block Diagram. • Updated the Voltage Sensor Controller with Avalon-MM Sample Storage figure. • Updated the Voltage Sensor Controller with External Sample Storage figure. • Minor text edits to the One Round Conversion and Continuous Conversion sections. • Minor text edits to the Configuring the Voltage Sensor Controller section.
September 2016	2016.09.01	• Changed instances of Quartus II to Quartus Prime. • Updated the `data` width in the Response Interface table. • Updated the Voltage Sample bits in the Sample Store Core Registers table. • Removed the bipolar mode support.
May 2015	2015.05.04	Initial release.