

Introduction

The LogiCORE™ IP AXI SYSMON ADC core is a 32-bit slave peripheral that connects to the AXI (Advanced eXtensible Interface) and provides the controller interface for the System Monitor (SYSMON) hard macro on Virtex®-6 family of FPGAs. This document describes the specifications for the AXI SYSMON ADC core. It is assumed that user is familiar with SYSMON hard macro. For information on SYSMON hard macro, see [UG370](#), *Virtex-6 FPGA System Monitor User Guide*, listed in the [Reference Documents](#) section.

Features

- AXI4-Lite interface is based on the AXI4 specification
- Connects as a 32-bit AXI4-Lite slave
- Uses dedicated System Monitor (SYSMON) hard macro on Virtex-6 devices
- Supports 10-bit, 200-kSPS (kilo-Samples Per Second) Analog-to-Digital Converter (ADC)
- Supports on-chip monitoring of supply voltages and temperature
- Supports one dedicated, high-bandwidth differential analog-input pair and 16 auxiliary low-bandwidth differential analog-input pairs
- Supports automatic alarms based on user-defined limits
- Supports optional interrupt request generation

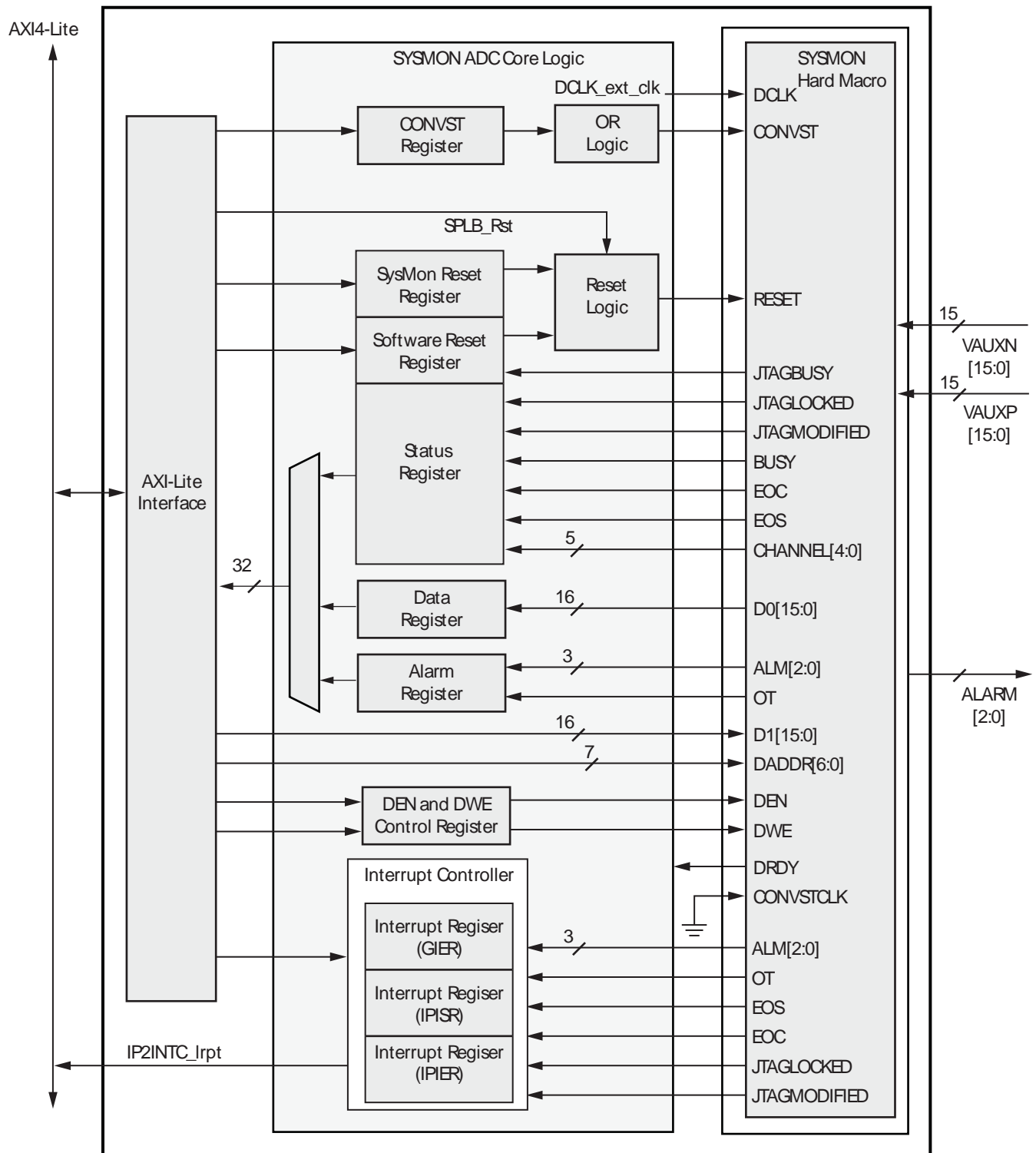
| LogiCORE IP Facts Table | | | | | |
|--|--------------------------------|-----|------------|------------|------------|
| Core Specifics | | | | | |
| Supported Device Family ⁽¹⁾ | Virtex-6 ⁽²⁾ | | | | |
| Supported User Interfaces | AXI4-Lite | | | | |
| | Resources | | | | Frequency |
| | LUTs | FFs | DSP Slices | Block RAMs | Max. Freq. |
| | See Table 13 . | | | | |
| Provided with Core | | | | | |
| Documentation | Product Specification | | | | |
| Design Files | VHDL | | | | |
| Example Design | Not Provided | | | | |
| Test Bench | Not Provided | | | | |
| Tested Design Tools | | | | | |
| Design Entry Tools | XPS 13.1 | | | | |
| Simulation | Mentor Graphics ModelSim 6.6d | | | | |
| Synthesis Tools | XST 13.1 | | | | |
| Support | | | | | |
| Provided by Xilinx, Inc. | | | | | |

Notes:

1. For a list of supported devices, see the [release notes](#) for this core.
2. For more information, see [DS150](#), *Virtex-6 Family Overview*.

Functional Description

The top-level block diagram for the AXI Sysmon ADC core is shown in [Figure 1](#).



DS790_01

Figure 1: Block Diagram of the AXI Sysmon ADC Core

The AXI Sysmon ADC core consists of following major blocks.

- AXI-Lite Interface Module
- SYSMON ADC Core Logic
- SYSMON Hard Macro

AXI-Lite Interface Module

The AXI-Lite Interface Module provides the AXI4-Lite decode service. Read and write transactions to and from the AXI4 are translated into equivalent SYSMON ADC core logic and SYSMON hard macro transactions. The register interfaces of the SYSMON ADC core logic connect to the AXI4 Interface Module.

Core Logic

The SYSMON ADC core logic provides necessary address decoding logic, control signal generation, and an interface between the AXI4-Lite and SYSMON hard macro. The read and write requests, along with the address and data (in case of a write transaction) from the AXI4 Interface Module, are transferred to either the Dynamic Reconfiguration Port (DRP) registers of SYSMON hard macro or local registers in the IP along with the necessary control signals, such as DEN and DWE.

The SYSMON ADC core logic supports the inclusion and exclusion of the Interrupt Controller based on the generic `C_INCLUDE_INTR`. The Interrupt Controller is included in the design if `C_INCLUDE_INTR = 1`.

There is new DRC limitation which has been imposed on the DCLK input clock of SYSMON hard macro on Virtex-6 devices. The DCLK for the hard macro must not exceed 80 MHz. To take care of this limitation, a new parameter `C_DCLK_RATIO` is added in the design .

Based on the core frequency (when used in the system), this parameter must be set to make the DCLK less than or equal to 80 MHz. These constraints are applicable only for Virtex-6 devices. The maximum clock at this port must be 80 MHz. If this clock increases beyond 80 MHz, a DRC violation related to the SYSMON hard macro will be raised, and the hard macro may not work properly.

The `C_DCLK_RATIO` supports range of values between 1 to 8. Internally, this value will be used to divide the AXI Clock. It is strongly recommended that, the value of `C_DCLK_RATIO` should be set in such a way that, the DCLK input frequency will be always equal to 80Mhz or close to 80Mhz. See [Assigning the C_DCLK_RATIO Parameter, page 21](#) before assigning the value to this parameter.

The SYSMON hard macro can be accessed via both the JTAG TAP (Test Access Port) and the AXI Sysmon ADC core. When simultaneous access of the SYSMON hard macro occurs, the JTAGLOCKED port can be asserted High by JTAG TAP, in which case the AXI Sysmon ADC core will not be allowed to do any read or write access from or to the DRP. When the JTAGLOCKED port is again de-asserted through JTAG TAP, the AXI Sysmon ADC core is allowed to perform read and write operation from or to the DRP.

This functionality is specially useful in applications where the user is configuring the DRP through JTAG TAP and does not want the fabric (AXI Sysmon ADC core) to alter the configuration. The user can make JTAGLOCKED = '1' through JTAG TAP which blocks any read or write transactions from or to the DRP through fabric, thus ensuring a non-destructive access through the JTAG TAP.SYSMON Hard Macro

The SYSMON hard macro is present in every Virtex-6 FPGA. The block diagram for the System Monitor ADC hard macro on a Virtex-6 FPGA is shown in Figure 2.

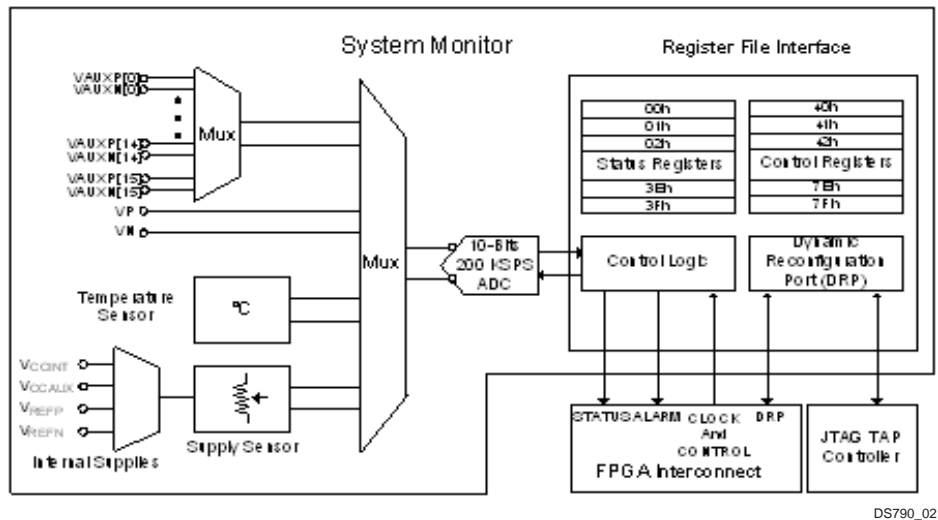


Figure 2: Block Diagram of the System Monitor ADC Hard Macro

The AXI Sysmon ADC core is built around the dedicated System Monitor hard macro of the Virtex-6 device family. The hard macro uses the 10-bit, 200-KSPS ADC internally for conversion of various analog data. The AXI Sysmon ADC core is used to measure die temperature and voltage. Additionally, the AXI Sysmon ADC core provides analog to digital conversion of up to 17 external channels. From a user point of view, the core is defined as a AXI Sysmon ADC which can monitor on-chip voltage and temperature, external analog voltages, or both.

The SYSMON hard macro consists of a Register File Interface (RFI) which in turn consists of status and control registers. Status registers are read-only and contain the results of analog-to-digital conversion of the on-chip sensors and external channels. The status registers also store the maximum and minimum temperature and VCCAUX/VCCINT voltages. The control registers are used to configure the SYSMON hard macro operation. SYSMON hard macro functionality, such as ADC operating modes, channel sequencer, and alarm limits, is controlled through these registers. The first three registers in the control register block, also known as configuration registers, are used to configure the SYSMON hard macro operating modes. In addition to the RFI of the hard macro, the AXI Sysmon ADC core consists of a set of local register and optional interrupt registers.

The SYSMON hard macro provides channel sequencing, averaging, and filtering functions. Many of the 16-bit registers are not defined in the SYSMON hard macro RFI. An undefined value is returned if accessing a location which is undefined.

In the SYSMON hard macro, a channel sequencer allows the user to specify the channels monitored (the sequence order is fixed). Users can specify an averaging filter to reduce noise. There are programmable alarm thresholds for the on-chip sensors, and if an on chip temperature or voltage is enabled and is outside the specified limit, an alarm is activated.

Structurally, the AXI Sysmon ADC core consists of the SYSMON hard macro, the AXI4 Interface Module, the Optional Interrupt Source Controller Module, the Soft Reset Module, the SYSMON Reset Register, and additional logic to interface to the core. The Soft Reset Module provides a way for resetting the entire IP without disturbing the entire system. The SYSMON Reset Register is provided to reset the SYSMON hard macro only.

All read and write operations to the configuration and limit registers are synchronized to the DCLK. The SYSMON hard macro has an internal clock divider which divides DCLK by any integer ranging from 2 to 255 to generate

ADCCLK. ADCCLK is an internal clock used by the ADC. Because an internal clock divider is provided, the DCLK frequency can be in the range of 2 MHz to 80 MHz. See the Virtex-6 FPGA data sheets for the maximum operating frequency of the SYSMON ADC core.

The SYSMON hard macro operates in either an event driven or continuous sampling mode. In event the driven sampling mode, the conversion process is initiated on the rising edge of CONVST. The AXI Sysmon ADC core supports this operation by providing a rising edge signal on the external CONVST port or by writing into the CONVST register. In the continuous sampling mode, the ADC continuous to carry out a conversion on the selected analog inputs as long as the ADCCLK (DCLK) is present. For more information on the SYSMON hard macro, see [Ref 1].

Design Parameters

To allow the user to obtain a AXI Sysmon ADC core that is uniquely tailored for their system, certain features can be parameterized in the AXI Sysmon ADC design. This allows the user to configure a design that utilizes the resources required by the system only and that operates with the best possible performance. The features that can be parameterized are as shown in Table 1.

Inferred Parameters

In addition to the parameters listed in Table 1, there are also parameters that are inferred for each AXI interface in the EDK tools. Through the design, these EDK-inferred parameters control the behavior of the AXI Interconnect. For a complete list of the interconnect settings related to the AXI interface, see the DS768, AXI Interconnect IP Data Sheet.

Table 1: Design Parameters

| Generic | Feature/Description | Parameter Name | Allowable Values | Default Value | VHDL Type |
|----------------------------------|------------------------------------|--------------------|--|---------------------|------------------|
| System Parameters | | | | | |
| G1 | Target FPGA family | C_FAMILY | virtex6 | virtex6 | string |
| AXI4 Parameters | | | | | |
| G2 | AXI Base Address | C_BASEADDR | Valid Address ⁽¹⁾ | None ⁽²⁾ | std_logic_vector |
| G3 | AXI High Address | C_HIGHADDR | Valid Address ⁽¹⁾ | None ⁽²⁾ | std_logic_vector |
| G4 | AXI Address Bus Width | C_S_AXI_ADDR_WIDTH | 32 | 32 | integer |
| G5 | AXI Data Bus Width | C_S_AXI_DATA_WIDTH | 32 | 32 | integer |
| AXI Sysmon ADC Parameters | | | | | |
| G6 | Include/Exclude interrupt support | C_INCLUDE_INTR | 0 = Exclude interrupt support 1 = Include interrupt support | 1 | integer |
| G7 | File name for Analog input stimuli | C_SIM_MONITOR_FILE | string | Design.txt | string |
| G8 | DCLK clock division ratio | C_DCLK_RATIO | 1,2,3,4,5,6,7,8 ⁽³⁾ | 2 ⁽³⁾ | string |

Table 1: Design Parameters (Cont'd)

| Generic | Feature/Description | Parameter Name | Allowable Values | Default Value | VHDL Type |
|---------|---------------------|----------------|------------------|---------------|-----------|
|---------|---------------------|----------------|------------------|---------------|-----------|

Notes:

1. The range C_BASEADDR to C_HIGHADDR is the address range for the AXI Sysmon ADC. This range is subject to restrictions to accommodate the simple address decoding scheme that is employed: The size, C_HIGHADDR - C_BASEADDR + 1, must be a power of two (2) and must be at least 0x400 to accommodate all AXI Sysmon ADC registers. However, a larger power than two (2) may be chosen to reduce decoding logic. C_BASEADDR must be aligned to a multiple of the range size.
2. An invalid default is used to ensure that an actual value appropriate to the system is set.
3. Based on the core frequency, this parameter should be set to generate the DCLK frequency less than or equal to 80 MHz. For more details, see [Core Logic, page 3](#) and [Assigning the C_DCLK_RATIO Parameter, page 21](#) before using this parameter.

I/O Signals

The AXI Sysmon ADC I/O signals are listed and described in [Table 2](#).

Table 2: I/O Signal Descriptions

| Port | Signal Name | Interface | I/O | Initial State | Description |
|---|--|-----------|-----|---------------|---|
| AXI Global System Signals | | | | | |
| P1 | S_AXI_ACLK | AXI | I | - | AXI Clock |
| P2 | S_AXI_ARESETN | AXI | I | - | AXI Reset, active LOW |
| AXI Write Address Channel Signals | | | | | |
| P3 | S_AXI_AWADDR[(C_S_AXI_ADDR_WIDTH - 1) : 0] | AXI | I | - | AXI Write address: The write address bus gives the address of the write transaction. |
| P4 | S_AXI_AWVALID | AXI | I | - | Write address valid: This signal indicates that a valid write address and control information are available. |
| P5 | S_AXI_AWREADY | AXI | O | 0 | Write address ready: This signal indicates that the slave is ready to accept an address and associated control signals. |
| AXI Write Channel Signals | | | | | |
| P6 | S_AXI_WDATA[(C_S_AXI_DATA_WIDTH - 1) : 0] | AXI | I | - | Write data |
| P7 | S_AXI_WSTB[((C_S_AXI_DATA_WIDTH/8) - 1) : 0] | AXI | I | - | Write strobes: This signal indicates which byte lanes to update in memory. |
| P8 | S_AXI_WVALID | AXI | I | - | Write valid: This signal indicates that valid write data and strobes are available. |
| P9 | S_AXI_WREADY | AXI | O | 0 | Write ready: This signal indicates that the slave can accept the write data. |
| AXI Write Response Channel Signals | | | | | |
| P10 | S_AXI_BRESP[1 : 0] | AXI | O | 0 | Write response: This signal indicates the status of the write transaction. "00" - OKAY (normal response) "10" - SLVERR (error response) "11" - DECERR (not issued by core) |
| P11 | S_AXI_BVALID | AXI | O | 0 | Write response valid: This signal indicates that a valid write response is available. |

Table 2: I/O Signal Descriptions (Cont'd)

| Port | Signal Name | Interface | I/O | Initial State | Description |
|--|--|-----------|-----|---------------|---|
| P12 | S_AXI_BREADY | AXI | I | - | Response ready: This signal indicates that the master can accept the response information. |
| AXI Read Address Channel Signals | | | | | |
| P13 | S_AXI_ARADDR[(C_S_AXI_ADDR_WIDTH - 1) : 0] | AXI | I | - | Read address: The read address bus gives the address of a read transaction. |
| P14 | S_AXI_ARVALID | AXI | I | - | Read address valid: This signal indicates, that when HIGH, the read address and control information is valid and will remain stable until the address acknowledgement signal, S_AXI_ARREADY, is High. |
| P15 | S_AXI_ARREADY | AXI | O | 1 | Read address ready: This signal indicates that the slave is ready to accept an address and associated control signals. |
| AXI Read Data Channel Signals | | | | | |
| P16 | S_AXI_RDATA[(C_S_AXI_DATA_WIDTH - 1) : 0] | AXI | O | 0 | Read data |
| P17 | S_AXI_RRESP[1 : 0] | AXI | O | 0 | Read response: This signal indicates the status of the read transfer. "00" - OKAY (normal response) "10" - SLVERR (error condition) "11" - DECERR (not issued by core) |
| P18 | S_AXI_RVALID | AXI | O | 0 | Read valid: This signal indicates that the required read data is available and the read transfer can complete. |
| P19 | S_AXI_RREADY | AXI | I | - | Read ready: This signal indicates that the master can accept the read data and response information. |
| AXI Sysmon ADC Core Interface Signals | | | | | |
| P20 | VAUXP[15 : 0] | SYSMON | I | - | Positive auxiliary differential analog inputs. |
| P21 | VAUXN[15 : 0] | SYSMON | I | - | Negative auxiliary differential analog inputs/ |
| P22 | CONVST | SYSMON | I | - | Convert Start input port is used to control the sampling instant on the ADC input and only in event-driven sampling mode. This port will be auto-connected to ground internally, if not in use. |
| P23 | ALARM[2:0] | SYSMON | O | 0 | SYSMON hard macro alarm output signals. |

Parameter - Port Dependencies

The dependencies between the AXI Sysmon ADC core design parameters and I/O signals are described in [Table 3](#).

Table 3: Parameter-Port Dependencies

| Generic or Port | Name | Affects | Depends | Relationship Description |
|--------------------------|--|-------------|---------|--|
| Design Parameters | | | | |
| G4 | C_S_AXI_ADDR_WIDTH | P3, P13 | - | Affects the number of bits in the address bus |
| G5 | C_S_AXI_DATA_WIDTH | P6, P7, P16 | | Affects the number of bits in the data bus |
| I/O Signals | | | | |
| P3 | S_AXI_AWADDR[(C_S_AXI_ADDR_WIDTH - 1) : 0] | - | G4 | Width of the S_AXI_AWADDR varies with C_S_AXI_ADDR_WIDTH. |
| P6 | S_AXI_WDATA[(C_S_AXI_DATA_WIDTH - 1) : 0] | - | G5 | Width of the S_AXI_WDATA varies according to C_S_AXI_DATA_WIDTH. |
| P7 | S_AXI_WSTB[((C_S_AXI_DATA_WIDTH/8) - 1) : 0] | - | G5 | Width of the S_AXI_WSTB varies according to C_S_AXI_DATA_WIDTH. |
| P13 | S_AXI_ARADDR[(C_S_AXI_ADDR_WIDTH - 1) : 0] | - | G4 | Width of the S_AXI_ARADDR varies with C_S_AXI_ADDR_WIDTH. |
| P16 | S_AXI_RDATA[(C_S_AXI_DATA_WIDTH - 1) : 0] | - | G5 | Width of the S_AXI_RDATA varies according to C_S_AXI_DATA_WIDTH. |

Register Descriptions

[Table 4](#) shows the AXI Sysmon ADC core registers and their corresponding addresses.

Table 4: Core Registers

| Base Address + Offset (hex) | Register Name | Access Type | Default Value (hex) | Description |
|--|---|----------------------|---------------------|--|
| AXI Sysmon ADC Local Register Grouping | | | | |
| C_BASEADDR + 0x00 | Software Reset Register (SRR) | Write ⁽¹⁾ | N/A | Software reset register. |
| C_BASEADDR + 0x04 | Status Register (SR) | Read ⁽²⁾ | N/A | Status register. |
| C_BASEADDR + 0x08 | Alarm Output Status Register (AOSR) | Read ⁽²⁾ | 0x0 | Alarm output status register. |
| C_BASEADDR + 0x0C | CONVST Register (CONVSTR) | Write ⁽¹⁾ | N/A | ADC convert start register. ⁽³⁾ |
| C_BASEADDR + 0x10 | SYSMON Reset Register (SYSMONRR) | Write ⁽¹⁾ | N/A | SYSMON hard macro reset register. |
| AXI Sysmon ADC Interrupt Controller Register Grouping | | | | |
| C_BASEADDR + 0x5C | Global Interrupt Enable Register (GIER) | R/W | 0x0 | Global interrupt enable register. |
| C_BASEADDR + 0x60 | IP Interrupt Status Register (IPIISR) | R/TOW ⁽⁴⁾ | N/A | IP interrupt status register. |

Table 4: Core Registers (Cont'd)

| Base Address + Offset (hex) | Register Name | Access Type | Default Value (hex) | Description |
|--|---|---------------------|---------------------|--|
| C_BASEADDR + 0x68 | IP Interrupt Enable Register (IPIER) | R/W | 0x0 | IP interrupt enable register. |
| SYSMON Hard Macro Register Grouping⁽⁵⁾ | | | | |
| C_BASEADDR + 0x200 | Temperature | Read ⁽⁶⁾ | N/A | The 10-bit MSB justified result of on-chip temperature measurement is stored in this register. |
| C_BASEADDR + 0x204 | V _{CCINT} | Read ⁽⁶⁾ | N/a | The 10-bit MSB justified result of on-chip V _{CCINT} supply monitor measurement is stored in this register. |
| C_BASEADDR + 0x208 | V _{CCAUX} | Read ⁽⁶⁾ | N/A | The 10-bit MSB justified result of on-chip V _{CCAUX} Data supply monitor measurement is stored in this register. |
| C_BASEADDR + 0x20C | V _P /V _N | R/W ⁽⁷⁾ | 0x0 | When read: The 10-bit MSB justified result of A/D conversion on the dedicated analog input channel (V _p /V _n) is stored in this register. When written: A write to this register will reset the SYSMON hard macro. No specific data is required. Applicable only when the Virtex-6 device is targeted. |
| C_BASEADDR + 0x210 | V _{REFP} | Read ⁽⁶⁾ | 0x0 | The 10-bit MSB justified result of an A/D conversion on the reference input V _{REFP} is stored in this register |
| C_BASEADDR + 0x214 | V _{REFN} | Read ⁽⁶⁾ | 0x0 | The 10-bit MSB justified result of an A/D conversion on the reference input V _{REFN} is stored in this register. |
| C_BASEADDR + 0x218 to C_BASEADDR + 0x21C | Undefined | N/A | Undefined | These locations are unused and contain invalid data. Do not Read or Write to these registers. |
| C_BASEADDR + 0x220 | Supply Offset | Read ⁽⁶⁾ | N/A | The calibration coefficient for the supply sensor offset is stored in this register. |
| C_BASEADDR + 0x224 | ADC Offset | Read ⁽⁶⁾ | N/A | The calibration coefficient for the ADC offset calibration is stored in this register. |
| C_BASEADDR + 0x228 | Gain Error | Read ⁽⁶⁾ | N/A | The calibration coefficient for the gain error is stored in this register. |
| C_BASEADDR + 0x22C to C_BASEADDR + 0x23C | Undefined | N/A | Undefined | These locations are unused and contain invalid data. Do not Read or Write to these registers. |
| C_BASEADDR + 0x240 | V _{AUXP} [0]/V _{AUXN} [0] | Read ⁽⁶⁾ | 0x0 | The 10-bit MSB justified result of an A/D conversion on the auxiliary analog input 0 is stored in this register. |
| C_BASEADDR + 0x244 | V _{AUXP} [1]/V _{AUXN} [1] | Read ⁽⁶⁾ | 0x0 | The 10-bit MSB justified result of an A/D conversion on the auxiliary analog input 1 is stored in this register. |

Table 4: Core Registers (Cont'd)

| Base Address + Offset (hex) | Register Name | Access Type | Default Value (hex) | Description |
|-----------------------------|---|---------------------|---------------------|---|
| C_BASEADDR + 0x248 | V _{AUXP} [2]/V _{AUXN} [2] | Read ⁽⁶⁾ | 0x0 | The 10-bit MSB justified result of an A/D conversion on the auxiliary analog input 2 is stored in this register. |
| C_BASEADDR + 0x24C | V _{AUXP} [3]/V _{AUXN} [3] | Read ⁽⁶⁾ | 0x0 | The 10-bit MSB justified result of an A/D conversion on the auxiliary analog input 3 is stored in this register. |
| C_BASEADDR + 0x250 | V _{AUXP} [4]/V _{AUXN} [4] | Read ⁽⁶⁾ | 0x0 | The 10-bit MSB justified result of an A/D conversion on the auxiliary analog input 4 is stored in this register. |
| C_BASEADDR + 0x254 | V _{AUXP} [5]/V _{AUXN} [5] | Read ⁽⁶⁾ | 0x0 | The 10-bit MSB justified result of an A/D conversion on the auxiliary analog input 5 is stored in this register. |
| C_BASEADDR + 0x258 | V _{AUXP} [6]/V _{AUXN} [6] | Read ⁽⁶⁾ | 0x0 | The 10-bit MSB justified result of an A/D conversion on the auxiliary analog input 6 is stored in this register. |
| C_BASEADDR + 0x25C | V _{AUXP} [7]/V _{AUXN} [7] | Read ⁽⁶⁾ | 0x0 | The 10-bit MSB justified result of an A/D conversion on the auxiliary analog input 7 is stored in this register. |
| C_BASEADDR + 0x260 | V _{AUXP} [8]/V _{AUXN} [8] | Read ⁽⁶⁾ | 0x0 | The 10-bit MSB justified result of A/D conversion on the auxiliary analog input 8 is stored in this register. |
| C_BASEADDR + 0x264 | V _{AUXP} [9]/V _{AUXN} [9] | Read ⁽⁶⁾ | 0x0 | The 10-bit MSB justified result of an A/D conversion on the auxiliary analog input 9 is stored in this register. |
| C_BASEADDR + 0x268 | V _{AUXP} [10]/V _{AUXN} [10] | Read ⁽⁶⁾ | 0x0 | The 10-bit MSB justified result of an A/D conversion on the auxiliary analog input 10 is stored in this register. |
| C_BASEADDR + 0x26C | V _{AUXP} /V _{AUXN} | Read ⁽⁶⁾ | 0x0 | The 10-bit MSB justified result of an A/D conversion on the auxiliary analog input 11 is stored in this register. |
| C_BASEADDR + 0x270 | V _{AUXP} /V _{AUXN} | Read ⁽⁶⁾ | 0x0 | The 10-bit MSB justified result of an A/D conversion on the auxiliary analog input 12 is stored in this register. |
| C_BASEADDR + 0x274 | V _{AUXP} /V _{AUXN} | Read ⁽⁶⁾ | 0x0 | The 10-bit MSB justified result of an A/D conversion on the auxiliary analog input 13 is stored in this register. |
| C_BASEADDR + 0x278 | V _{AUXP} /V _{AUXN} | Read ⁽⁶⁾ | 0x0 | The 10-bit MSB justified result of an A/D conversion on the auxiliary analog input 14 is stored in this register. |
| C_BASEADDR + 0x27C | V _{AUXP} /V _{AUXN} | Read ⁽⁶⁾ | 0x0 | The 10-bit MSB justified result of an A/D conversion on the auxiliary analog input 15 is stored in this register. |
| C_BASEADDR + 0x280 | Max Temp | Read ⁽⁶⁾ | N/A | The 10-bit MSB justified maximum temperature measurement. |
| C_BASEADDR + 0x284 | Max V _{CCINT} | Read ⁽⁶⁾ | N/A | The 10-bit MSB justified maximum V _{CCINT} measurement. |
| C_BASEADDR + 0x288 | Max V _{CCAUX} | Read ⁽⁶⁾ | N/A | The 10-bit MSB justified maximum V _{CCAUX} measurement. |

Table 4: Core Registers (Cont'd)

| Base Address + Offset (hex) | Register Name | Access Type | Default Value (hex) | Description |
|--|----------------------------|---------------------|---------------------|---|
| C_BASEADDR + 0x28C | Undefined | N/A | Undefined | This location is unused and contains invalid data. Do not Read or Write to this register. |
| C_BASEADDR + 0x290 | Min Temp | Read ⁽⁶⁾ | N/A | The 10-bit MSB justified minimum temperature measurement. |
| C_BASEADDR + 0x294 | Min V _{CCINT} | Read ⁽⁶⁾ | N/A | The 10-bit MSB justified minimum V _{CCINT} measurement. |
| C_BASEADDR + 0x298 | Min V _{CCAUX} | Read ⁽⁶⁾ | N/A | The 10-bit MSB justified minimum V _{CCAUX} measurement. |
| C_BASEADDR + 0x29C to C_BASEADDR + 0x2F8 | Undefined | N/A | Undefined | These locations are unused and contain invalid data. Do not Read or Write to these registers. |
| C_BASEADDR + 0x2FC | Flag Register | Read ⁽⁶⁾ | N/A | The 16-bit register gives general status information of ALARM, OT, Disable information of SYSMON, and information about whether the SYSMON is using internal reference voltage or external reference voltage. |
| C_BASEADDR + 0x300 | Configuration Register 0 | R/W ⁽⁹⁾ | 0x0 | SYSMON Configuration register 0. |
| C_BASEADDR + 0x304 | Configuration Register 1 | R/W ⁽⁹⁾ | 0x0 | SYSMON Configuration register 1. |
| C_BASEADDR + 0x308 | Configuration Register 2 | R/W ⁽⁹⁾ | 0x1E00 | SYSMON Configuration register 2. |
| C_BASEADDR + 0x30C to C_BASEADDR + 0x31C | Test register 0 to 4 | N/A | N/A | SYSMON Test register 0 to 4 (for factory test only) |
| C_BASEADDR + 0x320 | Sequence Register 0 | R/W | 0x0 | SYSMON Sequence register 0 (ADC channel selection). |
| C_BASEADDR + 0x324 | Sequence Register 1 | R/W | 0x0 | SYSMON Sequence register 1 (ADC channel selection). |
| C_BASEADDR + 0x328 | Sequence Register 2 | R/W | 0x0 | SYSMON Sequence register 2 (ADC channel averaging enable). |
| C_BASEADDR + 0x32C | Sequence Register 3 | R/W | 0x0 | SYSMON Sequence register 3 (ADC channel averaging enable). |
| C_BASEADDR + 0x330 | Sequence Register 4 | R/W | 0x0 | SYSMON Sequence register 4 (ADC channel analog-input mode). |
| C_BASEADDR + 0x334 | Sequence Register 5 | R/W | 0x0 | SYSMON Sequence register 5 (ADC channel analog-input mode). |
| C_BASEADDR + 0x338 | Sequence Register 6 | R/W | 0x0 | SYSMON Sequence register 6 (ADC channel acquisition time). |
| C_BASEADDR + 0x33C | Sequence Register 7 | R/W | 0x0 | SYSMON Sequence register 7 (ADC channel acquisition time). |
| C_BASEADDR + 0x340 | Alarm Threshold Register 0 | R/W | 0x0 | The 10-bit MSB justified alarm threshold register 0 (Temperature Upper). |

Table 4: Core Registers (Cont'd)

| Base Address + Offset (hex) | Register Name | Access Type | Default Value (hex) | Description |
|--|----------------------------|------------------------|---------------------|---|
| C_BASEADDR + 0x344 | Alarm Threshold Register 1 | R/W | 0x0 | The 10-bit MSB justified alarm threshold register 1 (V _{CCINT} Upper). |
| C_BASEADDR + 0x348 | Alarm Threshold Register 2 | R/W | 0x0 | The 10-bit MSB justified alarm threshold register 2 (V _{CCAUX} Upper). |
| C_BASEADDR + 0x34C | Alarm Threshold Register 3 | R/W ⁽⁹⁾⁽¹⁰⁾ | 0x0 | The 12-bit MSB justified alarm threshold register 3 (OT Upper). |
| C_BASEADDR + 0x350 | Alarm Threshold Register 4 | R/W | 0x0 | The 10-bit MSB justified alarm threshold register 4 (Temperature Lower). |
| C_BASEADDR + 0x354 | Alarm Threshold Register 5 | R/W | 0x0 | The 10-bit MSB justified alarm threshold register 5 (V _{CCINT} Lower). |
| C_BASEADDR + 0x358 | Alarm Threshold Register 6 | R/W | 0x0 | The 10-bit MSB justified alarm threshold register 6 (V _{CCAUX} Lower). |
| C_BASEADDR + 0x35C | Alarm Threshold Register 7 | R/W | 0x0 | The 10-bit MSB justified alarm threshold register 7 (OT Lower). |
| C_BASEADDR + 0x360 to C_BASEADDR + 0x3FC | Undefined | N/A | Undefined | Do not Read or Write to these registers. |

Notes:

1. Reading of this register returns undefined value.
2. Writing into this register has no effect.
3. Used in event-driven sampling mode only.
4. TOW = Toggle On Write. Writing a '1' to a bit position within the register causes the corresponding bit position in the register to toggle.
5. These are 16-bit registers internal to SYMON hard macro. These are mapped to lower half word boundary on 32-bit AXI Sysmon ADC core registers. For more details refer System Monitor Register File Interface section in [Reference Documents](#).
6. Writing to this SYSMON hard macro register is not allowed. The SYSMON ADC hard macro data registers are 16-bits in width. The SYSMON hard macro specification guarantees the first 10-MSB bits accuracy, therefore only these bits are used for reference.
7. Writing to this register will reset the SYSMON hard macro. No specific data pattern is required to reset the SYSMON hard macro. Reading of this register will give the details of Vp/Vn port.
8. See [Reference Documents](#) for setting the different bits available in configuration registers for Virtex-6 devices.
9. The OT Upper register is available only in Virtex-6 FPGA devices. This register location is N/A when Virtex-5 devices are targeted.
10. The OT Upper register is a user configurable register for upper threshold level of temperature. If this register is left un-configured, then the SYSMON will consider 125°C as the upper threshold value for OT. Note that while configuring this register, the last 4-bits must be set to 0011, for example, Alarm Threshold Register 3[3:0] = 0011. In addition, the upper 12 bits of this register are user configurable.
11. The timeout counter is not included in the core logic. Targeting the register space where there is no register may cause no response from the core, and the system may wait indefinitely. This is an important factor in creating designs.

Local Register Grouping

It is expected that the AXI Sysmon ADC core registers will be accessed in their preferred-access mode only. If the write attempt is made to read-only registers, there will not be any effect on the register contents. If the write-only registers are read, it will result in undefined data. All the internal registers of the core must be accessed in 32-bit format. If any other kind of access, such as half word or byte access, is done for the local 32-bit registers of the AXI Sysmon ADC core, the transaction will be completed with the generation of error for the corresponding transaction.

Software Reset Register (SRR)

The Software Reset Register permits the programmer to reset the AXI Sysmon ADC core including the SYSMON hard macro output ports (except JTAG related outputs), independently of other IP cores in the systems. To activate the software reset, the value 0x0000_000A must be written to the register. Any other access, read or write, has undefined results. The bit assignment in the software reset register is shown in Figure 3 and described in Table 5.

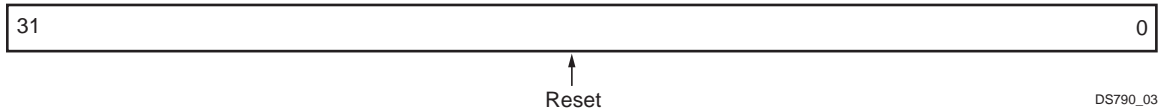


Figure 3: Software Reset Register

Table 5: Software Reset Register Description (C_BASEADDR + 0x00)

| Bit(s) | Name | Core Access | Reset Value | Description |
|--------|-------|-------------|-------------|--|
| 0-31 | Reset | Write only | N/A | The only allowed operation on this register is a write of 0x0000_000A, which resets the AXI Sysmon ADC core. The reset is active only for 16 clock cycles. |

Status Register (SR)

The Status Register (SR) contains the AXI Sysmon ADC core channel status, EOC, EOS, and JTAG access signals. This register is read only. Any attempt to write the bits of the register will not change the bits. The Status Register bit definitions are shown in Figure 4 and explained in Table 6.

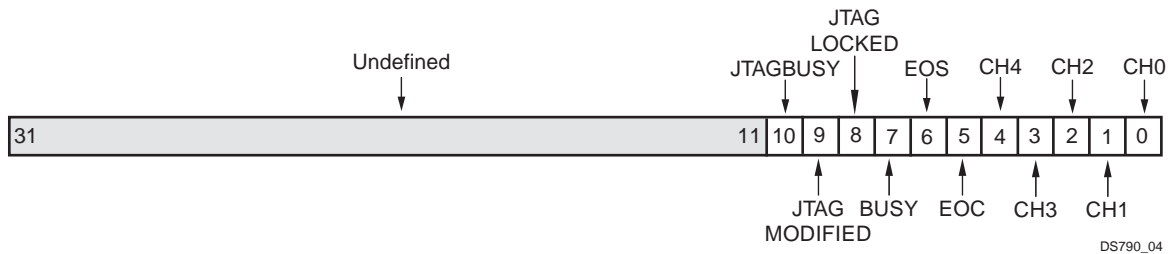


Figure 4: Status Register

Table 6: Status Register (C_BASEADDR + 0x04)

| Bit(s) | Name | Core Access | Reset Value | Description |
|---------|---------------|-------------|-------------|---|
| 31 - 11 | Undefined | N/A | N/A | Undefined. |
| 10 | JTAGBUSY | Read | '0' | Used to indicate that a JTAG DRP transaction is in progress. |
| 9 | JTAG MODIFIED | Read | '0' | Used to indicate that a write to DRP through JTAG interface has occurred. This bit is cleared when a successful DRP read/write operation through fabric is performed. The DRP read/write through fabric fails, if JTAGLOCKED = '1'. |
| 8 | JTAG LOCKED | Read | '0' | Used to indicate that a DRP port lock request has been made by the Joint Test Action Group (JTAG) interface. |
| 7 | BUSY | Read | N/A | ADC busy signal: This signal transitions high during an ADC conversion. |

Table 6: Status Register (C_BASEADDR + 0x04) (Cont'd)

| Bit(s) | Name | Core Access | Reset Value | Description |
|--------|-----------------|-------------|-------------|---|
| 6 | EOS | Read | N/A | End of Sequence: This signal transitions to an active High when the measurement data from the last channel in the auto sequence is written to the status registers. This bit is cleared when a read operation is performed on status register. |
| 5 | EOC | Read | N/A | End of Conversion signal: This signal transitions to an active High at the end of an ADC conversion when the measurement is written to the status register of the SYSMON hard macro. This bit is cleared when a read operation is performed on the status register. |
| 4 - 0 | CHANNEL [4 : 0] | Read | N/A | Channel selection outputs: The ADC input MUX channel selection for the current ADC conversion is placed on these outputs at the end of an ADC conversion. |

Alarm Output Status Register (AOSR)

The Alarm Output Status Register (AOSR) contains the alarm outputs for the AXI Sysmon ADC core. This register is read only. Any attempt to write the bits of the register will not change the bits. The Alarm Output Status Register bit definitions are shown in Figure 5 and explained in Table 7.

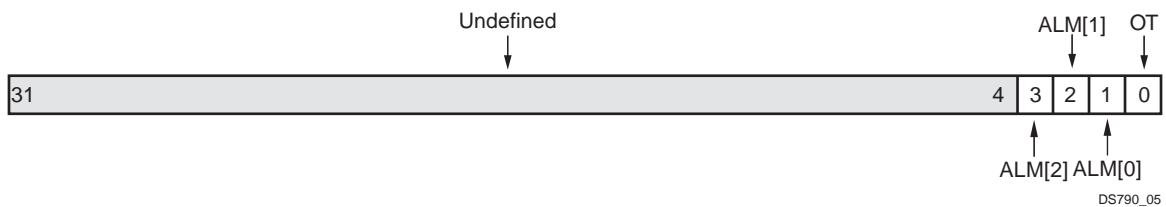


Figure 5: Alarm Output Status Register

Table 7: Alarm Output Status Register (C_BASEADDR + 0x08)

| Bit(s) | Name | Core Access | Reset Value | Description |
|--------|-----------|-------------|-------------|--|
| 31 - 4 | Undefined | N/A | N/A | Undefined. |
| 3 | ALM[2] | Read | '0' | System Monitor VCCAUX-sensor Interrupt: System Monitor V _{CCAUX} -sensor alarm output interrupt occurs when V _{CCAUX} exceeds user defined threshold. |
| 2 | ALM[1] | Read | '0' | System Monitor VCCINT-sensor Interrupt: System Monitor V _{CCINT} -sensor alarm output interrupt occurs when V _{CCINT} exceeds user defined threshold. |
| 1 | ALM[0] | Read | '0' | System Monitor temperature-sensor Interrupt: The System Monitor temperature-sensor alarm output interrupt occurs when the device temperature exceeds the user defined threshold. |
| 0 | OT | Read | '0' | Over-Temperature alarm Interrupt.: The Over-Temperature alarm output interrupt occurs when the die temperature exceeds a factory set limit of 125 degree celsius. |

CONVST Register (CONVSTR)

The CONVST Register (CONVSTR) is used for initiating a new conversion in the event-driven sampling mode. The output of this register is logically OR'ed with external the CONVST input signal. The attempt to read this register will result un-defined data. The CONVST Register bit definitions are shown in Figure 6 and explained in Table 8.

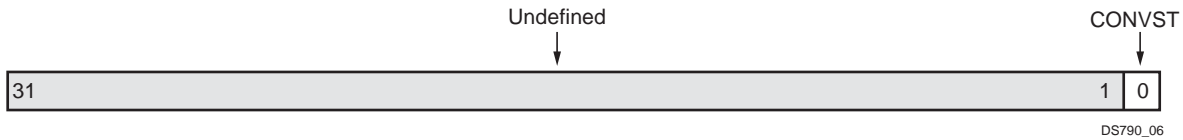


Figure 6: CONVST Register

Table 8: CONVST Register (C_BASEADDR + 0x0C)

| Bit(s) | Name | Core Access | Reset Value | Description |
|--------|-----------|-------------|-------------|---|
| 31 - 1 | Undefined | N/A | N/A | Undefined. |
| 0 | CONVST | Write | '0' | A rising edge on the CONVST input initiates the start of the ADC conversion in the event-driven sampling mode. For a selected channel, the CONVST bit in the register must be set to '1' and again reset to '0' to start a new conversion cycle. The conversion cycle ends with EOC bit going High. |

SYSMON Reset Register (SYSMONRR)

The SYSMON Reset Register (SYSMONRR) is used to reset the SYSMON hard macro only. As soon as the reset is released, the ADC begins with a new conversion. If sequencing is enabled, this conversion is the first in the sequence. This register resets the OT and ALM[n] output from the SYSMON hard macro. This register does not reset the interrupt registers if they are included in the design. Also note that any reset from the fabric does not affect the RFI (Register File Interface) contents of the SYSMON hard macro. The attempt to read this register will result in un-defined data. The SYSMON Reset Register bit definitions are shown in Figure 7 and explained in Table 9.

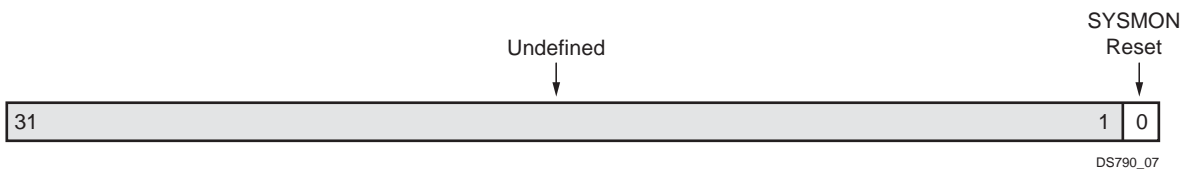


Figure 7: SYSMON Reset Register

Table 9: SYSMON Reset Register (C_BASEADDR + 0x10)

| Bit(s) | Name | Core Access | Reset Value | Description |
|--------|--------------|-------------|-------------|--|
| 31 - 1 | Undefined | N/A | N/A | Undefined. |
| 0 | SYSMON Reset | Write | '0' | Writing '1' to this bit position resets the SYSMON hard macro. The reset is released only after '0' is written to this register. |

Interrupt Controller Register Grouping

When C_INCLUDE_INTR = '1', the Interrupt Controller Module is included in the AXI Sysmon ADC core design. The AXI Sysmon ADC has a number of distinct interrupts that are sent to the Interrupt Controller Module which is one of the sub-modules of the AXI Sysmon ADC core. The Interrupt Controller Module allows each interrupt to be

enabled independently (via the IP interrupt enable register (IPIER)). All the interrupt signals are rising edge sensitive.

Interrupt registers are strictly 32-bit accessible. If byte/half-word or without byte-enables type of access is made, the core behavior is not guaranteed.

The interrupt registers are in the Interrupt Controller Module. The AXI Sysmon ADC core permits multiple conditions for an interrupt or an interrupt strobe which occurs only after the completion of a transfer.

Global Interrupt Enable Register (GIER)

The Global Interrupt Enable Register (GIER) is used to enable globally the final interrupt output from the Interrupt Controller as shown in Figure 8 and described in Table 10. This bit is a read or write bit and is cleared upon reset.

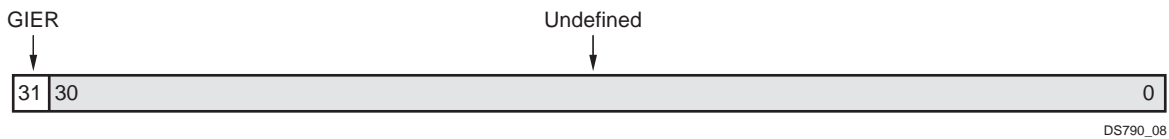


Figure 8: Global Interrupt Enable Register (GIER)

Table 10: Global Interrupt Enable Register (GIER) Description (C_BASEADDR + 0x5C)

| Bit(s) | Name | Access | Reset Value | Description |
|--------|-----------|--------|-------------|---|
| 31 | GIER | R/W | '0' | Global Interrupt Enable Register: It enables all individually enabled interrupts to be passed to the interrupt controller. '0' = Disabled '1' = Enabled |
| 30 - 0 | Undefined | N/A | N/A | Undefined. |

IP Interrupt Status Register (IPISR)

Six unique interrupt conditions are possible in the AXI Sysmon ADC core.

The Interrupt Controller has a register that enables each interrupt independently. The bit assignment in the interrupt register for a 32-bit data bus is shown in Figure 9 and described in Table 11. The interrupt register is a read/toggle on write register. Writing a '1' to a bit position within the register causes the corresponding bit position in the register to *toggle*. All register bits are cleared upon reset.



Figure 9: IP Interrupt Status Register (IPISR)

Table 11: IP Interrupt Status Register (IPISR) Description (C_BASEADDR + 0x60)

| Bit(s) | Name | Access | Reset Value | Description |
|---------|-----------------|-------------------------|-------------|--|
| 31 - 10 | Undefined | N/A | N/A | Undefined. |
| 9 | ALM[0] Deactive | R/TOW ⁽¹⁾ | '0' | ALM[0] Deactive Interrupt: This signal indicates that the falling edge of the ALM[0] (ALM[0] indicates that the user temperature range violation) signal is detected. It is cleared by writing '1' to this bit position. The ALM[0] Deactive signal is generated locally from the core. This signal indicates that the SYSMON macro has deactivated the user temperature violation signal output. |
| 8 | OT Deactive | R/TOW ⁽¹⁾ | '0' | OT Deactive Interrupt: This signal indicates that the falling edge of the Over Temperature signal is detected. It is cleared by writing '1' to this bit position. The OT Deactive signal is generated locally from the core. This signal indicates that the SYSMON macro has deactivated the Over Temperature signal output. |
| 7 | JTAG MODIFIED | R/TOW ⁽¹⁾⁽²⁾ | '0' | JTAGMODIFIED Interrupt: This signal indicates that a write to DRP through JTAG interface has occurred. It is cleared by writing '1' to this bit position. |
| 6 | JTAG LOCKED | R/TOW ⁽¹⁾⁽²⁾ | '0' | JTAGLOCKED Interrupt: This signal is used to indicate that a DRP port lock request has been made by the Joint Test Action Group (JTAG) interface. |
| 5 | EOC | R/TOW ⁽¹⁾⁽²⁾ | N/A | End of Conversion signal Interrupt: This signal transitions to an active High at the end of an ADC conversion when the measurement is written to the SYSMON hard macro's status register. |
| 4 | EOS | R/TOW ⁽²⁾ | N/A | End of Sequence Interrupt: This signal transitions to an active High when the measurement data from the last channel in the auto sequence is written to the status registers. |
| 3 | ALM[2] | R/TOW ⁽²⁾ | '0' | System Monitor VCCAUX-sensor Interrupt: The System Monitor VCCAUX-sensor alarm output interrupt occurs when the VCCAUX exceeds the user defined threshold. |
| 2 | ALM[1] | R/TOW ⁽²⁾ | '0' | System Monitor VCCINT-sensor Interrupt: The System Monitor VCCINT-sensor alarm output interrupt occurs when the VCCINT exceeds the user defined threshold. |
| 1 | ALM[0] | R/TOW ⁽²⁾ | '0' | System Monitor temperature-sensor Interrupt: The System Monitor temperature-sensor alarm output interrupt occurs when the device temperature exceeds the user defined threshold. |
| 0 | OT | R/TOW ⁽²⁾ | '0' | Over-Temperature alarm Interrupt: The Over-Temperature alarm output interrupt occurs when the die temperature exceeds the factory set limit of 125 degree celsius. |

Notes:

1. TOW = Toggle On Write: Writing a '1' to a bit position within the register causes the corresponding bit position in the register to toggle.
2. This interrupt signal is directly generated from SYSMON hard macro.

IP Interrupt Enable Register (IPIER)

The IPIER register has an enable bit for each defined bit of the IPISR as shown in Figure 10 and described in Table 12. All bits are cleared upon reset.

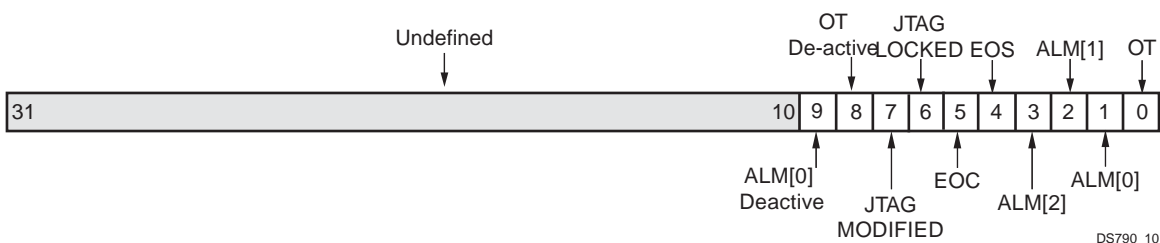


Figure 10: IP Interrupt Enable Register (IPIER)

Table 12: IP Interrupt Enable Register (IPIER) Description (C_BASEADDR + 0x68)

| Bit(s) | Name | Access | Reset Value | Description |
|---------|-----------------|--------|-------------|---|
| 31 - 10 | Undefined | N/A | N/A | Undefined. |
| 9 | ALM[0] Deactive | R/W | '0' | ALM[0] Deactive Interrupt: '0' = Disabled '1' = Enabled |
| 8 | OT Deactive | R/W | '0' | OT Deactive Interrupt: '0' = Disabled '1' = Enabled |
| 7 | JTAG MODIFIED | R/W | '0' | JTAGMODIFIED Interrupt: '0' = Disabled '1' = Enabled |
| 6 | JTAG LOCKED | R/W | '0' | JTAGLOCKED Interrupt: '0' = Disabled '1' = Enabled |
| 5 | EOC | R/W | '0' | End of Conversion signal Interrupt: '0' = Disabled '1' = Enabled |
| 4 | EOS | R/W | '0' | End of Sequence Interrupt: '0' = Disabled '1' = Enabled |
| 3 | ALM[2] | R/W | '0' | System Monitor V _{CCAUX} -sensor Interrupt: '0' = Disabled '1' = Enabled |
| 2 | ALM[1] | R/W | '0' | System Monitor V _{CCINT} -sensor Interrupt: '0' = Disabled '1' = Enabled |
| 1 | ALM[0] | R/W | '0' | System Monitor temperature-sensor Interrupt: '0' = Disabled '1' = Enabled |
| 0 | OT | R/W | '0' | Over-Temperature alarm Interrupt: '0' = Disabled '1' = Enabled |

More about Locally Generated Interrupt Bits in IPIER and IPISR

The interrupt bits ranging from the bit-9 to bit-0 in IPISR, as well as IPIER, are direct output signals of the SYSMON hard macro. Signals such as OT Deactive (bit-8) and ALM[0] Deactive (bit-9) are locally generated in the core. These two interrupts will be generated on the falling edge of the Over Temperature and AML[0] signals. The falling edge of these two signals may be used to controlling external functions such as the fan or the air-conditioning of the system. See [Reference Documents](#) for details about the significance of these interrupts.

SYSMON Hard Macro Register (DRP Register) Grouping

The SYSMON hard macro register set consists of all the registers present in the SYSMON hard macro on the Virtex-6 FPGAs. The addresses of these registers are mentioned in [Table 4](#). Because these registers are 16-bit wide but the processor data bus is 32-bit wide, the hard macro register data resides on the lower 16 bits of the 32-bit data bus as shown in [Figure 11](#). The 10-bit MSB aligned A/D converted value of different channels from the SYSMON hard macro are left shifted and reside from bit position 15 to 6 of the processor data bus. The remaining bit positions from 5 to 0 should be ignored while considering the ADC data for different channels. Along with 16-bit data, JTAGMODIFIED and JTAGLOCKED bits are passed which can be used by the software driver application to determine the validity of the DRP read data. The JTAGMODIFIED bit is cleared when a DRP read/write operation through the fabric is successful. A DRP read/write through the fabric fails if JTAGLOCKED = '1'. The JTAGLOCKED signal is independently controlled through JTAG TAP. It is expected that these SYSMON hard macro registers should be accessed in their preferred access-mode only. The AXI Sysmon ADC core will not be able to differentiate any non-preferred access to the SYSMON hard macro registers. For more information on these registers, see [Reference Documents](#) section.



Figure 11: SYSMON Hard Macro Register

DRP registers are accessed as part of cores local registers — these registers must be accessed through the core local registers. Any attempt to access these registers in byte or half-word manner will return the error response from the core.

Design Implementation

Target Technology

The intended target technology is the Virtex-6 family FPGAs.

Device Utilization and Performance Benchmarks

Core Performance

Because the AXI Sysmon ADC core will be used with other design modules in the FPGA, the utilization and timing numbers reported in this section are estimates only. When the AXI Sysmon ADC core is combined with other

designs in the system, the utilization of FPGA resources and timing of the AXI Sysmon ADC core design will vary from the results reported here.

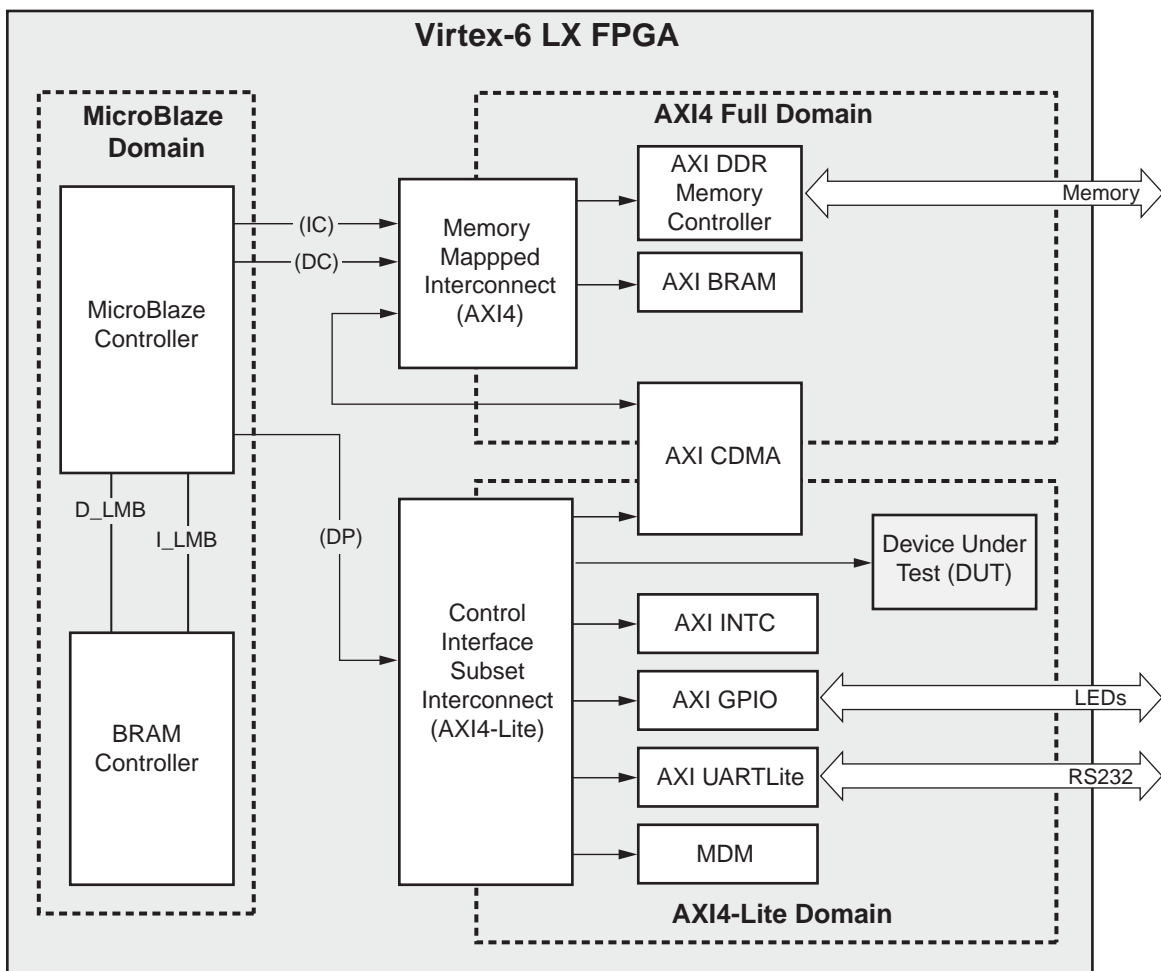
The AXI Sysmon ADC core resource utilization for various parameter combinations measured with the Virtex-6 FPGA as the target device are detailed in [Table 13](#).

Table 13: Performance and Resource Utilization Benchmarks on the Virtex-6 FPGA (xc6vlx130t-ff1156-1)

| Parameter Values (Other parameters at default values) | Device Resources | | Performance |
|---|------------------|------|------------------------|
| C_INCLUDE_INTR | Slice Flip-Flops | LUTs | F _{Max} (MHz) |
| 0 | 113 | 107 | 200 |
| 1 | 181 | 183 | 200 |

Note:

- For above utilization calculation, the C_DCLK_RATIO = 3 is used, while the AXI clock was targeted at 200 MHz.



DS790_13

Figure 12: Virtex-6 LX FPGA System with the AXI Sysmon ADC Core as the DUT

The target FPGA was then filled with logic to drive the LUT and BRAM utilization to approximately 70% and the I/O utilization to approximately 80%. Using the default tool options and the slowest speed grade for the target FPGA, the resulting target F_{MAX} numbers are shown in [Table 14](#).

Table 14: System Performance

| Target FPGA | Target F_{MAX} (MHz) |
|-------------|------------------------|
| V6LX130t-1 | 180 |

The target F_{MAX} is influenced by the exact system and is provided for guidance. It is not a guaranteed value across all systems.

Assigning the C_DCLK_RATIO Parameter

The parameter C_DCLK_RATIO has a range from 1 to 8. Any value in this range divides the AXI clock with the value. The divided clock output of the BUFR primitive will be provided to the DCLK input of SYSMON hard macro.

The maximum frequency limitation of 80 MHz is applicable only for Virtex-6 devices. See the SYSMON user guide for Virtex-6 devices for maximum operating DCLK clock frequencies.

Along with the DCLK clock, Configuration Register 2 can be set for internal clock division of DCLK. The DCLK clock input is further divided by this configuration register which is used as the clock for internal operation of the SYSMON macro.

The internal operation speed of the SYSMON macro will now be decided by the Configuration Register 2 settings. When a Virtex-6 device is targeted, the DCLK clock input to the macro can be a maximum of 80 MHz. It is recommended to have the core frequency and DCLK clock be an even integer ratio (for example, core frequency = 200 MHz, C_CLK_RATIO = 4 making the DCLK = 50 MHz). If these are not an even integer ratio, then the macro will generate a DRC error. The Configuration Register 2 of the SYSMON hard macro decides the internal operating frequency of the macro.

Setting DCLK and Configuration Register 2 decides the internal clock of the SYSMON macro. It is recommended to read the SYSMON user guide before setting any of the above mentioned values.

User Application Examples

This section provides examples on configuring AXI Sysmon ADC core in either continuous cycling of sequence or single channel (continuous or event driven) mode. It is assumed that the user is aware with the AXI Sysmon ADC core register descriptions given in [Table 4 on page 8](#).

Continuous Cycling Of Sequence Mode Example

To configure the AXI Sysmon ADC core in Continuous Cycling of Sequence Mode, set the SEQ1 and SEQ0 bits in Configuration Register 1 to '1' and '0' respectively. The specific value written to registers may vary depending upon the need of application. Below is the configuration example for monitoring the On-Chip Temperature, V_{CCINT} , and V_{CCAUX} channel in the continuous cycling of sequence mode with the clock ratio set to 32.

1. Issue a software reset by writing the data word 0x0000_000A to the SRR. This asserts the reset of AXI Sysmon ADC core for 16 AXI clock cycles.
2. Write 0x0000_0000 to Configuration Register 0. This configures the SYSMON hard macro in continuous sampling mode.

3. Write 0x0000_2000 to Configuration Register 1. This configures the SYSMON hard macro in continuous cycling of sequence mode which results in all calibration disabled and all alarm outputs enabled.
4. Write 0x0000_2000 to Configuration Register 2. This configures the SYSMON hard macro to have $ADCCLK = DCLK/32$.
5. Read the Status Register (SR) to reset EOC/EOS signal set by any previous conversions.
6. If an interrupt controller is present, read the IPISR to learn the value set by any previous conversions. Assume that for this application, the value read is 0x0000_003E.
7. Write 0x0000_003E to IPISR to toggle the bits which are '1' so that the new value of IPISR becomes 0x0000_0000.
8. If an interrupt controller is present, for example, $C_INCLUDE_INTR = 1$, perform a global enabling of interrupts by writing 0x8000_0000 to the GIER.
9. Write 0x0000_00FF to the IPIER to enable the operational interrupts.
10. Write 0x0000_0700 to Sequence Register 0 and 0x0000_0000 to Sequence Register 1. This configures SYSMON hard macro for monitoring On-Chip Temperature, V_{CCINT} and V_{CCAUX} channel.
11. Write 0x0000_0000 to Sequence Register 2 and 3. This disables ADC channel averaging.
12. Write 0x0000_0000 to Sequence Register 4 and 5. This configures the ADC channel in unipolar input mode.
13. Write 0x0000_0000 to Sequence Register 6 and 7. This configures the ADC channel acquisition time to four $ADCCLK$ cycles.
14. Write 0x0000_A900 to Alarm Register 0. This configures the upper limit for the temperature alarm, which for this application is set to 60° C.
15. Write 0x0000_9980 to Alarm Register 1. This configures the upper limit for the V_{CCINT} alarm, which for this application is set to 1.8 V.
16. Write 0x0000_EE80 to Alarm Register 2. This configures the upper limit for the V_{CCAUX} alarm, which for this application is set to 2.8 V.
17. Write 0x0000_A000 to Alarm Register 4. This configures the lower limit for the temperature alarm, which for this application is set to 42° C.
18. Write 0x0000_4400 to Alarm Register 5. This configures the lower limit for the V_{CCINT} alarm, which for this application is set to 0.8 V.
19. Write 0x0000_9980 to Alarm Register 6. This configures the lower limit for the V_{CCAUX} alarm, which for this application is set to 1.8 V.
20. Write 0x0000_A180 to Alarm Register 7. This configures the lower limit for the OT alarm, which for this application is set to 45° C.
21. The Alarm Register 3 will be active only in case of the Virtex-6 device. When the Virtex-6 FPGA is targeted, this register is used to set the upper limit of OT. The OT upper is a 12 bit register. Set the lower 4 bits to "0011". If this register is left un-initialized, 125° C will be considered as the default upper temperature for OT.
22. Write 0x0000_2000 to Configuration Register 1. This configures the SYSMON hard macro in continuous cycling of sequence mode, all calibration disabled, and all alarm outputs enabled. Perform a write operation on this register to enable the sequence written to sequence registers. [Refer System Monitor User Guide for bits of Configuration Register 0, when targeted for Virtex-6 devices]
23. Read SR. If the present conversion cycle is completed, EOS bit in SR is set to '1'. If the interrupt controller is present, the EOS bit in IPISR is also set to '1'.
24. Read converted value of the On-Chip Temperature, V_{CCINT} and V_{CCAUX} channel, from address $C_BASEADDR + 0x200$, $C_BASEADDR + 0x204$, and $C_BASEADDR + 0x208$, respectively.

Single Channel Mode Examples

To configure the AXI Sysmon ADC core in single channel mode, set both SEQ1 and SEQ0 bits in configuration register 1 to '1'. The single channel operation can be programmed to operate either in the event-driven sampling mode or continuous sampling mode by setting the \overline{EC} bit in the configuration register to 0 to '1' or '0'.

Should this be register 0 instead of register to 0?

Single Channel Event-Driven Sampling Mode Example

To configure the AXI Sysmon ADC core in the single channel event-driven sampling mode, set the \overline{EC} bit in the configuration register 0 to '1'.

The specific value written to the registers may vary depending upon the need of the application. In addition, if the On-Chip temperature or voltages are monitored, configure the alarm registers with the appropriate values before writing to the configuration registers. The subsequent numbered instructions outline the configuration example for monitoring the V_P/V_N channel with the clock ratio set to 32.

1. Issue a software reset by writing the data word 0x0000_000A to the SRR. This asserts the reset of the AXI Sysmon ADC core for 16 clock cycles.
2. If an interrupt controller is present, such as C_INCLUDE_INTR = 1, write 0x8000_0000 to the GIER to enable globally the interrupts.
3. Writing 0x0000_00FF to the IPIER to enable the operational interrupts.
4. Write 0x0000_0203 to configuration register 0. This configures the SYSMON hard macro with no averaging, unipolar mode, event driven sampling, and selects channel 3 (V_P/V_N), for conversion.
5. Write 0x0000_3000 to Configuration Register 1. This configures the SYSMON hard macro in single channel mode resulting in calibration disabled and alarm outputs enabled.
6. Write 0x0000_2000 to Configuration Register 2. This configures the SYSMON hard macro to have ADCCLK = DCLK/32.
7. Read Status Register (SR) to reset EOC/EOS signal set by any previous conversions.
8. If interrupt controller is present, read IPISR to learn the value set by any previous conversions. For this application, assume that the value read is 0x0000_003E.
9. Write 0x0000_003E to IPISR to toggle the bits which are '1', so that the new value of IPISR becomes 0x0000_0000.
10. Conversion Start can be signalled by writing 0x0000_0001 to the CONVSTR or by making the external CONVST port = '1'.
11. Reset the CONVSTR by writing 0x0000_0000 to it or by making CONVST port = '0' depending upon which type of trigger (either CONVSTR register or CONVST port) is used for conversion start.
12. Read SR. If conversion is completed, the EOC bit in SR will be set to '1'. If the interrupt controller is present, the EOC bit in the IPISR is also set to '1'.
13. Read the converted value of channel 3 (V_P/V_N) from address C_BASEADDR + 0x20C.

Single Channel Continuous Sampling Mode Example

To configure AXI Sysmon ADC core in Single Channel Continuous Sampling Mode, \overline{EC} bit in Configuration Register 0 should be set to '0'. The specific value written to registers may vary depending upon the need of application. Also if On-Chip temperature or voltages are monitored then Alarm registers should be configured with appropriate values before writing to Configuration Registers. Below is the configuration example for monitoring V_P/V_N channel with clock ratio set to 32.

1. Issue a software reset by writing the data word 0x0000_000A to the SRR. This asserts the reset of AXI Sysmon ADC core for 16 clock cycles.

2. If interrupt controller is present, i.e. C_INCLUDE_INTR = 1, do global enabling of interrupts by writing 0x8000_0000 to GIER.
3. Enable the operational interrupts by writing 0x0000_00FF to the IPIER.
4. Write 0x0000_0003 to Configuration Register 0. This configures the SYSMON hard macro with no averaging, unipolar mode, event driven sampling, and selects channel 3 (V_P/V_N) for conversion.
5. Write 0x0000_3000 to Configuration Register 1. This configures the SYSMON hard macro in single channel mode, all calibration disabled and all alarm outputs enabled.
6. Write 0x0000_2000 to Configuration Register 2. This configures the SYSMON hard macro to have $ADCCLK = DCLK/32$.
7. Write 0x0000_0001 to the SYSMON Reset Register to reset the SYSMON hard macro. This step is required to put the SYSMON hard macro in the reset state.
8. Read Status Register (SR) to reset EOC/EOS signal set by any previous conversions. After reading the Status Register the EOC, EOS from IP core will be in reset state.
9. If interrupt controller is present, read IPISR to know the value set by any previous conversions. Assume that for this application, the value read is 0x0000_003E.
10. Write 0x0000_003E to IPISR to toggle the bits which are '1' so that the new value of IPISR becomes 0x0000_0000.
11. Write 0x0000_0000 to the SYSMON Reset Register to bring the SYSMON hard macro out of reset. Once the SYSMON hard macro comes out of reset, it will start its normal operation of data acquisition of the configured channels.
12. Read SR, if conversion is completed then the EOC bit in SR will be set to '1'. If the interrupt controller is present, the EOC bit in the IPISR is also set to '1'.
13. Read the converted value of channel 3 (V_P/V_N) from address C_BASEADDR + 0x20C.

Reference Documents

1. [UG370](#), *Virtex-6 FPGA System Monitor User Guide*
2. AXI4 AMBA® AXI Protocol Version: 2.0 Specification
3. [DS765](#), *LogiCORE IP AXI Lite IPIF Data Sheet*

Support

Xilinx provides technical support for this LogiCORE product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support of product if implemented in devices that are not defined in the documentation, if customized beyond that allowed in the product documentation, or if changes are made to any section of the design labeled *DO NOT MODIFY*.

Ordering Information

This Xilinx LogiCORE IP module is provided at no additional cost with the Xilinx ISE® Design Suite Embedded Edition software under the terms of the [Xilinx End User License](#). The core is generated using the Xilinx ISE Embedded Edition software (EDK).

Information about this and other Xilinx LogiCORE IP modules is available at the [Xilinx Intellectual Property](#) page. For information on pricing and availability of other Xilinx LogiCORE modules and software, please contact your [local Xilinx sales representative](#).

Revision History

| Date | Version | Revision |
|----------|---------|---|
| 9/21/10 | 1.0 | Xilinx Initial Release |
| 12/14/10 | 1.1 | Updated to v1.01a for 12.4 release; updated design tools and Performance and Resource Utilization data in Table 13 on page 20 . |
| 12/22/10 | 1.2 | Closed CR 580235. Updated the figure 1. Added description for C_DCLK_RATIO and added the same parameter in Table 1. |
| 3/1/11 | 3.0 | Updated core to v2.00a. Updated Xilinx tools to v13.1. |

Notice of Disclaimer

Xilinx is providing this product documentation, hereinafter "Information," to you "AS IS" with no warranty of any kind, express or implied. Xilinx makes no representation that the Information, or any particular implementation thereof, is free from any claims of infringement. You are responsible for obtaining any rights you may require for any implementation based on the Information. All specifications are subject to change without notice. XILINX EXPRESSLY DISCLAIMS ANY WARRANTY WHATSOEVER WITH RESPECT TO THE ADEQUACY OF THE INFORMATION OR ANY IMPLEMENTATION BASED THEREON, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OR REPRESENTATIONS THAT THIS IMPLEMENTATION IS FREE FROM CLAIMS OF INFRINGEMENT AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Except as stated herein, none of the Information may be copied, reproduced, distributed, republished, downloaded, displayed, posted, or transmitted in any form or by any means including, but not limited to, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of Xilinx.