L-ASC10

In-System Programmable
Hardware Management Expander

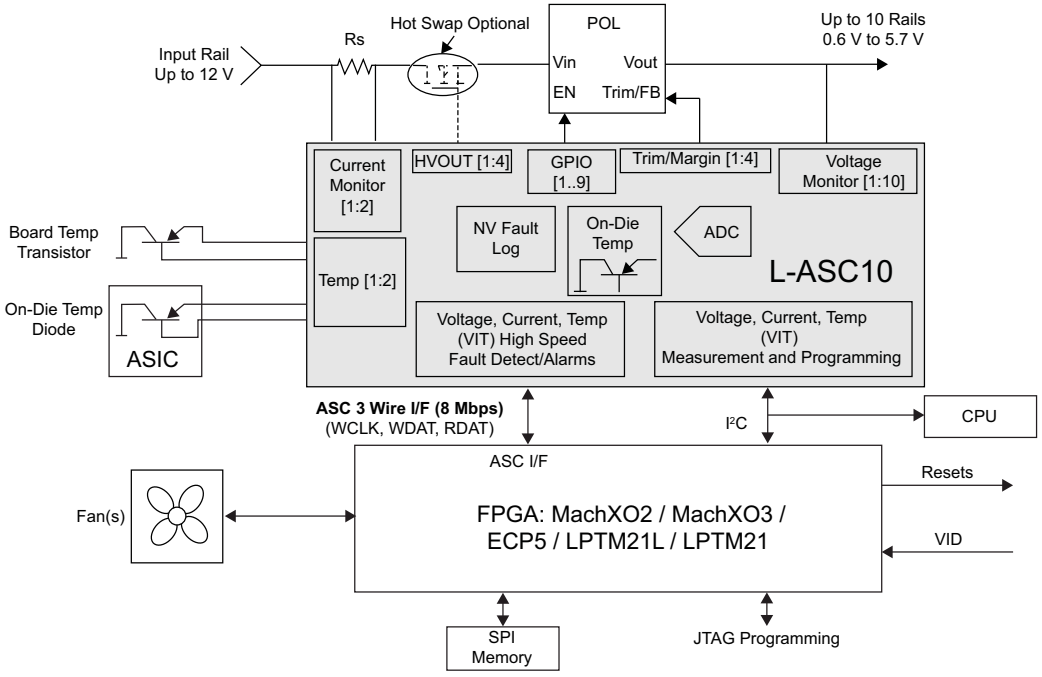
September 2019 **Data Sheet FPGA-DS-02038**

Features

- **Ten Rail Voltage Monitoring and Measurement**
  - UV/OV Fault Detection Accuracy - 0.2% Typ.
  - Fault Detection Speed <100 µs
  - High Voltage, Single Ended and Differential Sensing
- **Two Channel Wide-Range Current Monitoring and Measurement**
  - High-side current Measurement up to 12 V
  - Programmable OC/UC Fault Detect
  - Detects Current faults in < 1 µs
- **Three Temperature Monitoring and Measurement Channels**
  - Programmable OT/UT Faults Threshold
  - Two channels of Temperature Monitoring using external diodes
  - One On-Chip Temperature Monitor
- **Four High-Side MOSFET Drivers**
  - Programmable Charge Pump
- **Four Precision Trim and Margin Channels**
  - Closed Loop Operation
  - Voltage Scaling and VID Support
- **Nine General Purpose Input / Output**
  - 5 V tolerant I/O
- **Non-Volatile Fault Logging**
- **In-system Programmable Through I²C**
  - Non-Volatile Configuration
  - Background Programming Support
- **System Level Support**
  - 3.3 V Operation, wide input supply range 2.8 V to 3.6 V
  - Industrial temperature range
  - 48-pin QFN
  - RoHS compliant and halogen-free
- **Applications**
  - Telecommunication and Networking
  - Industrial, Test & Measurement
  - Medical Systems
  - Servers and Storage Systems
  - High Reliability Systems

Application Diagram

Figure 1. Hardware Management Application Block Diagram



Description

The L-ASC10 (Analog Sense and Control - 10 rail) is a Hardware Management (Power, Thermal, and Control Plane Management) Expander designed to be used with Platform Manager 2, MachXO2, MachXO3, or ECP5 FPGAs to implement the Hardware Management Control function in a circuit board. The L-ASC10 (referred to as ASC) enables seamless scaling of power supply voltage and current monitoring, temperature monitoring, sequence and margin control channels. The ASC includes dedicated interfaces supporting the exchange of monitor signal status and output control signals with these centralized hardware management controllers. Up to eight ASC devices can be used to implement a hardware management system.

The ASC provides three types of analog sense channels: voltage (nine standard channels and one high voltage channel), current (one standard voltage and one high voltage), and temperature (two external and one internal) as shown in Figure 2.

Each of the analog sense channels is monitored through two independently programmable comparators to support both high/low and in-bounds/out-of-bounds (window-compare) monitor functions. The current sense channels feature a programmable gain amplifier and a fast fault detect (<1 µs response time) for detecting short circuit events. The temperature sense channels can be configured to work with different external transistor or diode configurations.

Nine general purpose 5 V tolerant open-drain digital input/output pins are provided that can be used in a system for controlling DC-DC converters, low-drop-out regulators (LDOs) and optocouplers, as well as for supervisory and general purpose logic interface functions. Four high-voltage charge pumped outputs (HVOUT1-HVOUT4) may be configured as high-voltage MOSFET drivers to control high-side MOSFET switches. These HVOUT outputs can also be programmed as static output signals or as switched outputs (to support external charge pump implementation) operating at a dedicated duty cycle and frequency.

The ASC device incorporates four TRIM outputs for controlling the output voltages of DC-DC converters. Each power supply output voltage can be maintained typically within 0.5% tolerance across various load conditions using the Digital Closed Loop Control mode.

The internal 10-bit A/D converter can be used to monitor the voltage and current through the I^2C bus. The ADC is also used in the digital closed loop control mode of the trimming block.

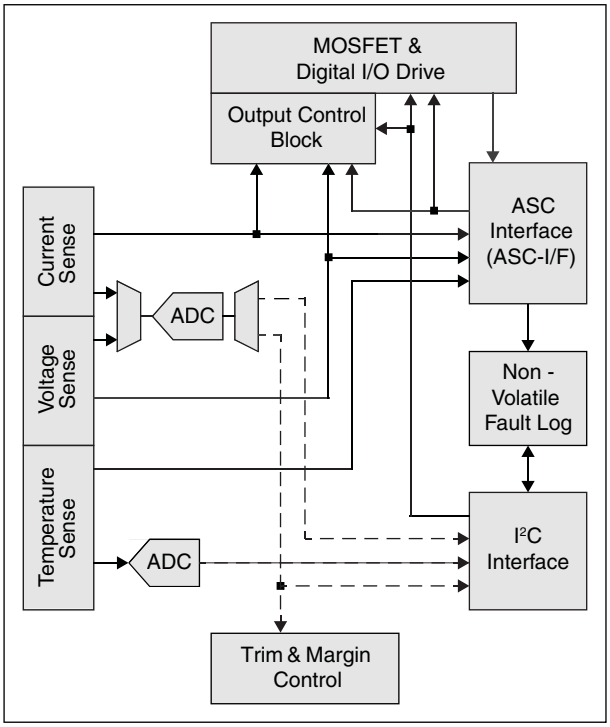
The ASC also provides the capability of logging up to 16 status records into the on-chip nonvolatile EEPROM memory. Each record includes voltage, current and temperature monitor signals along with digital input and output levels.

The dedicated ASC Interface (ASC-I/F) is a reliable serial channel used to communicate with a Platform Manager 2, MachXO2, MachXO3, or ECP5 FPGA in a scalable star topology. The centralized control algorithm in the FPGA monitors signal status and controls output behavior via this ASC-I/F. The ASC I^2C interface is used by the FPGA or an external microcontroller for ASC background programming, interface configuration, and additional data transfer such as parameter measurement or I/O control or status. For example, voltage trim targets can be set over the I^2C bus and measured voltage, current, or temperature values can be read over the I^2C bus.

The ASC also includes an on-chip output control block (OCB) which allows certain alarms and control signals a direct connection to the GPIOs or HVOUTs, bypassing the ASC-I/F for a faster response. The OCB is used to connect the fast current fault detect signal to an FPGA input directly. It also supports functions like Hot Swap with a programmable hysteretic controller.

ASC Block Diagram

Figure 2. ASC Block Diagram



DC and Switching Characteristics

Absolute Maximum Ratings

Symbol	Parameter	Conditions	Min	Max.	Units
V_{CCA}	Main Power Supply		–0.5	4.5	V
V_{IN_VMON}	VMON input voltage		–0.5	6	V
V_{IN_VMONGS}	VMON input voltage ground sense		–0.5	6	V
V_{IN_HIMONP}	High voltage IMON input voltage		–0.5	13.3	V
V_{IN_HIMONN}	High voltage IMON return/ VMON input voltage		–0.5	13.3	V
V_{DIFF_HIMON}	High voltage IMON differential voltage		–2.0	2.0	V
V_{IN_IMONP}	Low voltage IMON1 input voltage		–0.5	6.0	V
V_{IN_IMONN}	Low voltage IMON1 return voltage		–0.5	6.0	V
V_{DIFF_IMON}	Low voltage IMON1 differential voltage		–2.0	2.0	V
V_{IN_TMONP}	TMON input voltage		–0.5	V_{CCA}	V
V_{IN_TMONN}	TMON return voltage		–0.5	V_{CCA}	V
V_{IN_GPIO}	Digital input voltage		–0.5	6	V
V_{OUT}	Open-drain output voltage	HVOUT [1:4]	–0.5	13.3	V
		GPIO[1:6], GPIO[8:10]	–0.5	6	V
V_{TRIM}	TRIM output voltage		–0.5	V_{CCA}	V
$I_{SINKMAX}$	Maximum Sink Current on any output			23	mA
T_S	Device Storage Temperature		–65	+125	°C
T_A	Ambient Temperature		–40	+125	°C

Recommended Operating Conditions

Symbol	Parameter	Conditions	Min	Max.	Units
V_{CCA}	Main Power Supply[1]		2.8	3.6	V
V_{IN_VMON}	VMON input voltage		–0.3	5.9	V
V_{IN_VMONGS}	VMON input voltage ground sense		–0.2	0.3	V
V_{IN_HIMONP}	High voltage IMON input voltage[2]		4.5	13.2	V
V_{IN_HIMONN}	High voltage IMON return/ VMON input voltage[2]		4.5	13.2	V
V_{DIFF_HIMON}	High voltage IMON differential voltage		0	500	mV
V_{IN_IMONP}	Low voltage IMON1 input voltage	Low Side Sense Disabled	0.6	5.9	V
		Low Side Sense Enabled	–0.3	1.0	V
V_{IN_IMONN}	Low voltage IMON1 return voltage	Low Side Sense Disabled	0.6	5.9	V
		Low Side Sense Enabled	–0.3	1.0	V
V_{DIFF_IMON}	Low voltage IMON1 differential voltage		0	500	mV
V_{IN_GPIO}	Digital input voltage		–0.3	5.5	V
V_{OUT}	Open-drain output voltage	HVOUT [1:4]	–0.3	13.2	V
		GPIO[1:6], GPIO[8:10]	–0.3	5.5	V
T_A	Ambient Temperature		–40	+85	°C

1. The VCC of the I/O bank of the MachXO2, MachXO3, ECP5, LPTM21L, or LPTM21 that is used for the ASC-I/F needs to be connected to the VCCA of the respective ASC device. See System Connections section for more details
2. HIMON circuits are operational down to 3 V. Accuracy is guaranteed within Recommended Operating Conditions

Analog Specifications

Symbol	Parameter	Conditions	Min.	Typ.	Max	Units
I_{CCA}	Supply Current	V_{CCA} = 3.3 V, Ta 25 °C		25	35	mA
$I_{CC\text{-}HVOUT}$	Supply Current Adder per HVOUT	V_{HVOUT} = 12 V, I_{SRC} = 100 uA, V_{CCA} = 3.3 V, Ta 25 °C			2	mA
I_{CCPROG}	Supply Current during Programming	V_{CCA} = 3.3 V, Ta 25 °C			40	mA

ESD Performance

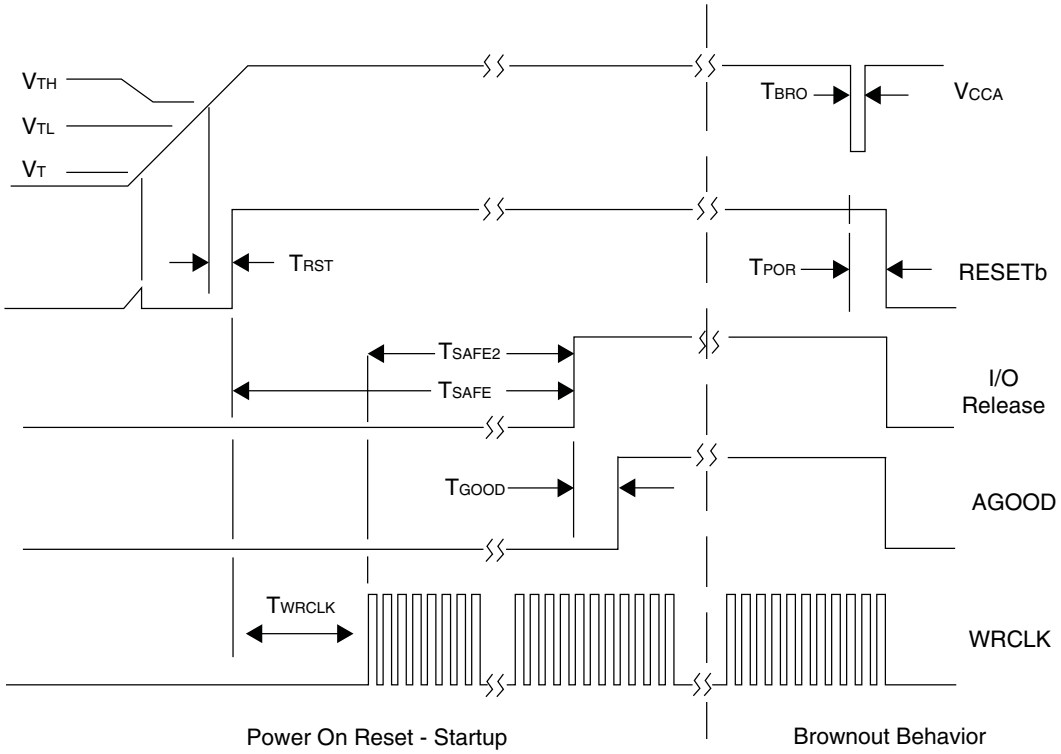
Please refer to the Platform Manager 2 Product Family Qualification Summary for complete qualification data, including ESD performance.

Power-On Reset

Symbol	Parameter	Conditions	Min	Typ.	Max.	Units
T_{RST}	Delay from VTH to start-up state				100	µs
T_{SAFE}	Delay from RESETb release to ASC Safe State Exit and I/O Release[1, 2]			1.8		ms
T_{SAFE2}	Delay from WRCLK start to ASC Safe State Exit and I/O Release[1, 2, 3]		56			µs
T_{GOOD}	Delay from I/O release to AGOOD asserted high in FPGA section[4]			16		µs
T_{WRCLK}	Delay from RESETb release to WRCLK start[5]			1.4		ms
T_{BRO}	Minimum duration brown out required to trigger RESETb		1		5	µs
T_{POR}	Delay from Brown out to reset state				13	µs
V_{TL}	Threshold below which RESETb is LOW				2.3	V
V_{TH}	Threshold above which RESETb is Hi-Z		2.7			V
V_T	Threshold above which RESETb is valid		0.8			V
C_L	Capacitive load on RESETb				200	pF

1. Both T_{SAFE} and T_{SAFE2} must complete before I/O are released from Safe State.
2. During the calibration period before T_{SAFE} and T_{SAFE2}, the ASC may ignore RESETb being driven low. After T_{SAFE} and T_{SAFE2}, the ASC can be reset by another device by driving RESETb low.
3. Safe State is released at ASC after a fixed number (64) of WRCLK cycles (typ. 8 MHz frequency) and three ASC-I/F data packets are properly detected.
4. AGOOD asserted in the FPGA on the next ASC-I/F packet after I/O exits Safe State as ASC.
5. Parameter is dependent on the FPGA configuration refresh time during POR. See Platform Manager 2, MachXO2, MachXO3, or ECP5 data sheet for details.

Figure 3. ASC Power-On Reset



Voltage Monitors[1]

Symbol	Parameter	Conditions	Min	Typ	Max	Units
R_{VMON_in}	Input Resistance		55	65	75	kΩ
C_{VMON_in}	Input Capacitance			8		pF
V_{MON} Range	Programmable trip-point Range		0.075		5.734	Volts
V_{MON} Accuracy	Absolute accuracy of any trip-point – Differential VMON pins	VMON voltage > 0.650 V		0.2	0.7	%
	Single-ended VMON pins	VMON voltage > 0.650 V		0.3	0.9	%
V_{MON} HYST	Hysteresis of any trip-point (relative to setting)			1		%
V_{MON} CMR	Differential VMON Common mode rejection ratio			60		dB
V_Z Sense	Low Voltage Sense Trip Point Error – Differential VMON1-4	Trip Point = 0.075 V	–5		+5	mV
		Trip Point = 0.150 V	–5		+5	mV
		Trip Point = 0.300 V	–10		+10	mV
		Trip Point = 0.545 V	–15		+15	mV
	Low Voltage Sense Trip Point Error – Single-Ended VMON5-9	Trip Point = 0.080 V	–10		+10	mV
		Trip Point = 0.155 V	–15		+15	mV
		Trip Point = 0.310 V	–25		+25	mV
		Trip Point = 0.565 V	–55		+55	mV
High Voltage Monitor						
HV_{MON} Range	High Voltage VMON programmable trip-point range		0.3		13.2	Volts
HV_{MON} Accuracy	HVMON Absolute accuracy of any trip-point	HVMON voltage > 1.8 V		0.4	1.0	%
V_Z Sense	Low Voltage Sense Trip Point Error - HVMON pin	Trip Point = 0.220 V	–20		+20	mV
		Trip Point = 0.425 V	–35		+35	mV
		Trip Point = 0.810 V	–75		+75	mV
		Trip Point = 1.280 V	–130		+130	mV

1. VMON accuracy may degrade based on SSO conditions of hardware management controller ASC-I/F. See the System Connections section for more details.

Current Monitors

Symbol	Parameter	Conditions	Min	Typ	Max	Units
$I_{IMONPleak}$	IMON1P input leakage	Low Side Sense Disabled Fast Trip Point V_{sns} = 500 mV	–2		250	µA
		Low Side Sense Enabled Fast Trip Point V_{sns} = 500 mV	–2		40	µA
$I_{IMONNleak}$	IMON1N input leakage	Low Side Sense Disabled Fast Trip Point V_{sns} = 500 mV	–2		2	µA
		Low Side Sense Enabled Fast Trip Point V_{sns} = 500 mV	–200		2	µA
$I_{HIMONPleak}$	HIMONP input leakage	Fast Trip Point V_{sns} = 500 mV			550	µA
$I_{HIMONNleak}$	HIMONN_HVMON input leakage				350	µA
$I_{MONA/B}$ Accuracy[2]	HIMON, IMON1A/B Comparator Trip Point accuracy	Gain = 100x		8		%
		Gain = 50x		5		%
		Gain = 25x		3		%
		Gain = 10x		2		%
$I_{MONA/B}$ Gain	Programmable Gain Setting	Four settings in software		10		V/V
				25		V/V
				50		V/V
				100		V/V
I_{MONF} Accuracy[2]	Fast comparator trip-point accuracy	V_{sns}[1] = 50 mV, 100 mV, or 150 mV		8		%
		V_{sns} = 200 mV, 250 mV, or 300 mV		5		%
		V_{sns} = 400 mV or 500 mV		3		%
t_{IMONF}	Fast comparator response time				1	µs

1. V_{sns} is the differential voltage between IMON1P and IMON1N (or HIMONP and HIMONN).
2. IMON accuracy may degrade based on SSO conditions of hardware management controller ASC-I/F. See the System Connections section for more details.

ADC Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Units
	Resolution			10		Bits
$t_{CONVERT}$	Conversion Time from I^2C Request				200	µs
Voltage Monitors						
$V_{VMON-IN}$	Input Range Full scale	Programmable Attenuator = 1	0		2.048	V
		Programmable Attenuator = 3	0		5.91	
LSB	ADC Step Size	Programmable Attenuator = 1		2		mV
		Programmable Attenuator = 3		6		
$E_{VMON-attenuator}$	Error due to attenuator	Programmable Attenuator = 3		+/– 0.1		%
High Voltage Monitor						
$V_{HVMON-IN}$	Input Range Full scale	Programmable Attenuator = 4	0		8.192	V
		Programmable Attenuator = 8	0		13.21	
LSB	ADC Step Size	Programmable Attenuator = 4		8		mV
		Programmable Attenuator = 8		16		
$E_{HVMON-attenuator}$	Error due to attenuator	Programmable Attenuator = 4		+/–0.2		%
		Programmable Attenuator = 8		+/–0.4		%
Current Monitors						
$t_{IMON-sample}$	Sample period of HIMON and IMON1 conversions for averaged value	4 Settings via I^2C command		1		ms
				2		
				4		
				8		
$V_{IMON-IN}$	Input Range Full scale[1]	Programmable Gain 10x	0		200	mV
		Programmable Gain 25x	0		80	
		Programmable Gain 50x	0		40	
		Programmable Gain 100x	0		20	
LSB	ADC Step Size	Programmable Gain 10x		0.2		mV
		Programmable Gain 25x		0.08		
		Programmable Gain 50x		0.04		
		Programmable Gain 100x		0.02		

1. Differential voltage applied across HIMONP/IMON1P and HIMONN/IMON1N before programmable gain amplification.

ADC Error Budget Across Entire Operating Temperature Range

Symbol	Parameter	Conditions	Min	Typ	Max	Units
TADC Error	Total ADC Measurement Error at Any Voltage (Differential Analog Inputs)[1, 3]	Measurement Range 600 mV - 2.048 V, VMONxGS > –100 mV, Attenuator =1	8	+/– 4	8	mV
		Measurement Range 600 mV - 2.048 V, VMONxGS > –200 mV, Attenuator =1		+/– 6		mV
		Measurement Range 0 - 2.048 V, VMONxGS > –200 mV, Attenuator =1		+/– 10		mV
	Total Measurement Error at Any Voltage (Single-Ended Analog Inputs including IMON)[1, 2, 3]	Measurement Range 600 mV - 2.048 V, Attenuator =1	–8	+/– 4	8	mV

1. Total error, guaranteed by characterization, includes INL, DNL, Gain, Offset, and PSR specs of the ADC.
2. Programmable gain error on IMON not included.
3. ADC accuracy may degrade based on SSO conditions of hardware management controller ASC-I/F. See the System Connections section for more details

Temperature Monitors

Symbol	Parameter	Conditions	Min	Typ	Max	Units
T_{MON_REMOTE} Accuracy[1, 7]	Temp Error – Remote Sensor	Ta=–40 to +85 °C Td=–64 to 150 °C		1		°C
T_{MON_INT} Accuracy[7]	Internal Sensor – Relative to ambient[6]	Ta=–40 to +85 °C		1		°C
T_{MON} Resolution	Measurement Resolution			0.25		°C
T_{MON} Range	Programmable threshold range		–64		155	°C
T_{MON} Offset	Temperature offset	Programmable in software	–64		63.75	°C
T_{MON} Hysteresis	Hysteresis of trip points	Programmable in software	0		63	°C
t_{TMON_settle}[2]	Temperature measurement settling time[3]	Measurement Averaging Coefficient = 1		15		ms
		Measurement Averaging Coefficient = 8		120		ms
		Measurement Averaging Coefficient = 16		240		ms
T_n	Ideality Factor ***n***	Programmable in software	0.9		2	
T_{limit}	Temperature measurement limit[4]				160	°C
C_{TMON}	Maximum capacitance between T_{MONP} and T_{MONN} pins				200	pF
$R_{TMONSeries}$	Equivalent external resistance to sensor[5]				200	ohms

1. Accuracy number is valid for the use of a grounded collector pnp configuration, programmed with proper ideality factor, and 16x measurement filter enabled. Any other device or configuration can have additional errors, including beta, series resistance and ideality factor accuracy. See the Temperature Monitors section for more details.
2. Settling time based on one TMON enabled. For multiple TMONs, settling time can be multiplied by the number of enabled TMON channels.
3. Settling time is defined as the time it takes a step change to settle to 1% of the measured value.
4. All values above T_{limit} read as 0x3FF over I^2C. There is no cold temperature limiting reading, although performance is not specified below –64 °C.
5. This is the maximum series resistance which the TMON circuit can compensate out. Equivalent series resistance includes all board trace wiring (TMONP and TMONN) as well as parasitic base and emitter resistances. Re=1/gm should not be included as part of series resistance.
6. Internal sensor is subject to self-heating, dependent on PCB design and device configuration. Self-heating not included in published accuracy.
7. TMON accuracy may degrade based on SSO conditions of hardware management controller ASC-I/F. See the System Connections section for more details.

Digital Specifications

Symbol	Parameter	Conditions	Min	Typ	Max	Units
I_{IL},I_{IH}	Input Leakage, no pull-up, pull-down[2]				+/– 10	µA
I_{PD}	Active Pull-Down Current[2]	GPIO[1:10] configured as Inputs, Internal Pull-Down enabled		200		µA
$I_{PD-ASCIF}$	Input Leakage (WDAT and WRCLK)[3]	Internal Pull-Down		175		µA
$I_{OH-HVOUT}$	Output Leakage Current	HVOUT[1:4] in open drain mode and pulled up to 12 V		35	100	µA
$I_{PU-RESETb}$	Input Pull-Up Current (RESETb)			–50		µA
V_{IL}	Voltage input, logic low	GPIO[1:10]			0.8	V
		SCL, SDA			30% VCCA	
V_{IH}	Voltage input, logic high	GPIO[1:10]	2.0			V
		SCL, SDA	70% VCCA			
V_{OL}	HVOUT[1:4] (open drain mode)	I_{SINK} = 10 mA			0.8	V
	GPIO[1:6], GPIO[8:10]	I_{SINK} = 20 mA				
$I_{SINKTOTAL}$[1]	All digital outputs				130	mA

1. Sum of maximum current sink from all digital outputs combined. Reliable operation is not guaranteed if this value is exceeded.
2. During safe-state, all GPIO default to output, see the Safe State of Digital Outputs section for more details. GPIO[1:6] and GPIO[10] default to active low output. This will result in a leakage current dependent on the input voltage which can exceed the specified input leakage
3. WRCLK and WDAT pins may see transients above 1 mA in hot socket conditions. DC levels will remain below 1 mA.

High Voltage FET Drivers

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{PP}	Gate driver output voltage	Four settings in software		12		Volts
				10		
				8		
				6		
I_{OUTSRC}	Gate driver source current (HIGH state)	Four settings in software		12.5		µA
				25		
				50		
				100		
$I_{OUTSINK}$	Gate driver sink current (LOW state)	Four settings in software		100		µA
				250		
				500		
				3000		
Frequency	Switched Mode Frequency	Two settings in software		15.625		kHz
				31.25		
Duty Cycle	Switched Mode Programmable Duty Cycle Range	Programmable in software	6.25		93.75	%
	Duty Cycle step size			6.25		%

Margin/Trim DAC Output Characteristics

Symbol	Parameter	Conditions	Min	Typ.	Max.	Units
	Resolution			8 (7 + sign)		Bits
FSR	Full scale range			+/– 320		mV
LSB	LSB step size			2.5		mV
I_{OUT}	Output source/sink current		–200		200	µA
I_{TRIM_Hi-Z}	Tri-state mode leakage			0.1		µA
BPZ	Bipolar zero output voltage (code=80h)	Four settings in software		0.6		V
				0.8		
				1.0		
				1.25		
t_S	TrimCell output voltage settling time[1]	DAC code changed from 80H to FFH or 80H to 00H			2.5	ms
		Single DAC code change		256		µs
C_LOAD	Maximum load capacitance				50	pF
TOSE	Total open loop supply voltage error[2]	Full scale DAC corresponds to +/– 5% supply voltage variation	–1%		+1%	V/V

1. To 1% of set value with 50 pF load connected to trim pins.
2. Total resultant error in the trimmed power supply output voltage referred to any DAC code due to DAC's INL, DNL, gain, output impedance, offset error and bipolar offset error across the temperature, and V_{CCA} ranges of the device.

Fault Log

Symbol	Parameter	Conditions	Min	Typ.	Max.	Units
Records	Number of available fault log records in EEPROM			16		Records
$t_{faultTrigger}$	Minimum active time of trigger signal to start fault recording		64			µs
$t_{faultRecord}$	Time to copy fault record to EEPROM				5	ms
$t_{faultWrite}$	Time to complete writing fault record in EEPROM				10	ms

Oscillator

Symbol	Parameter	Conditions	Min	Typ.	Max.	Units
CLK_{ASC}	Internal ASC0 Clock		7.6	8	8.4	MHz
CLK_{ext}	Externally Applied Clock		7.6	8	8.4	MHz

Propagation Delays

Symbol	Parameter	Conditions	Min	Typ.	Max.	Units
Voltage Monitors						
$t_{VMONtoFPGA}$	Propagation delay VMON input to signal update at FPGA	Glitch Filter Off		48		µs
		Glitch Filter ON		96		µs
$t_{VMONtoOCB}$[2]	Propagation delay VMON input to output update at OCB	Glitch Filter Off			16	µs
		Glitch Filter ON			64	µs
Current Monitors						
$t_{IMONtoFPGA}$	Propagation delay IMON input to signal update at FPGA	Glitch Filter Off		48		µs
		Glitch Filter ON		96		µs
$t_{IMONtoOCB}$[2]	Propagation delay IMON input to output update at OCB	Glitch Filter Off			16	µs
		Glitch Filter ON			64	µs
$t_{IMONFtoOCB}$[2]	Propagation delay IMONF input to output update at OCB				1	µs
Temperature Monitors						
$t_{TMONtoFPGA}$	Propagation delay TMON input to signal update at FPGA[1]	Monitor Alarm Filter Depth = 1		15		ms
		Monitor Alarm Filter Depth = 16		240		ms
GPIO – Inputs						
$t_{GPIOtoFPGA}$	Propagation delay GPIO input to signal update at FPGA			32		µs
$t_{GPIOtoOCB}$[2]	Propagation delay GPIO input to output update at OCB				50	ns
GPIO – Outputs						
$t_{FPGAtoGPIO}$	Propagation delay FPGA signal update to GPIO output			32		µs
$t_{OCBtoGPIO}$[3]	Propagation delay OCB input to output update at GPIO				50	ns
HVOUT						
$t_{FPGAtoHVOUT}$	Propagation delay FPGA signal update to HVOUT output			32		µs
$t_{OCBtoHVOUT}$[3, 4]	Propagation delay OCB input to output update at HVOUT				110	ns
TRIM DAC						
$t_{FPGAtoTrimOE}$	Propagation delay FPGA signal update to TRIM_OE update			32		µs

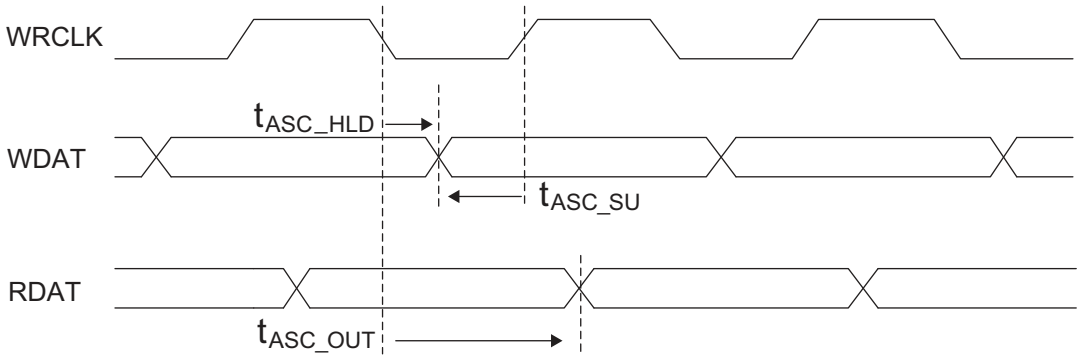
1. Propagation delay based on one TMON enabled. For multiple TMONs, propagation delay can be multiplied by the number of enabled TMON channels.
2. OCB output propagation delays measured using time delay to GPIO output from OCB. Propagation delay is measured on falling GPIO outputs. Rising output propagation time will be dependent on external pull-up resistor.
3. OCB input propagation delays measured using time delay from GPIO input to OCB. Propagation delay is measured on falling GPIO outputs. Rising output propagation time will be dependent on external pull-up resistor.
4. HVOUT propagation delay measured with HVOUT in open-drain mode, with switched mode disabled. Propagation delay in charge pump mode is dependent on external load and HVOUT settings.

ASC Interface (ASC-I/F) Timing Specifications[1]

Symbol	Parameter	Conditions	Min	Typ	Max	Units
f_{wrclk}	WRCLK frequency			8		MHz
t_{ASC_HLD}	Hold time between WRCLK falling edge and WDAT transition		0			ns
t_{ASC_SU}	Setup time between WDAT transition and WRCLK rising edge		25			ns
t_{ASC_OUT}	Delay from WRCLK falling edge to RDAT transition				50	ns

1. All timing conditions valid for VCCIO = 3.3 V at FPGA ASC-I/F and ASC VCCA range of 2.8 V to 3.6 V.

Figure 4. ASC Interface (ASC-I/F) Timing Diagram



I²C Port Timing Specifications

Symbol	Parameter	Min.	Max.	Units
f_{MAX}	Maximum SCL clock frequency		400	kHz

1. ASC supports the following modes:
   a. Standard-mode (Sm), with a bit rate up to 100 kbit/s (user and configuration mode)
   b. Fast-mode (Fm), with a bit rate up to 400 kbit/s (user and configuration mode)
2. Refer to the I²C specification for timing requirements.

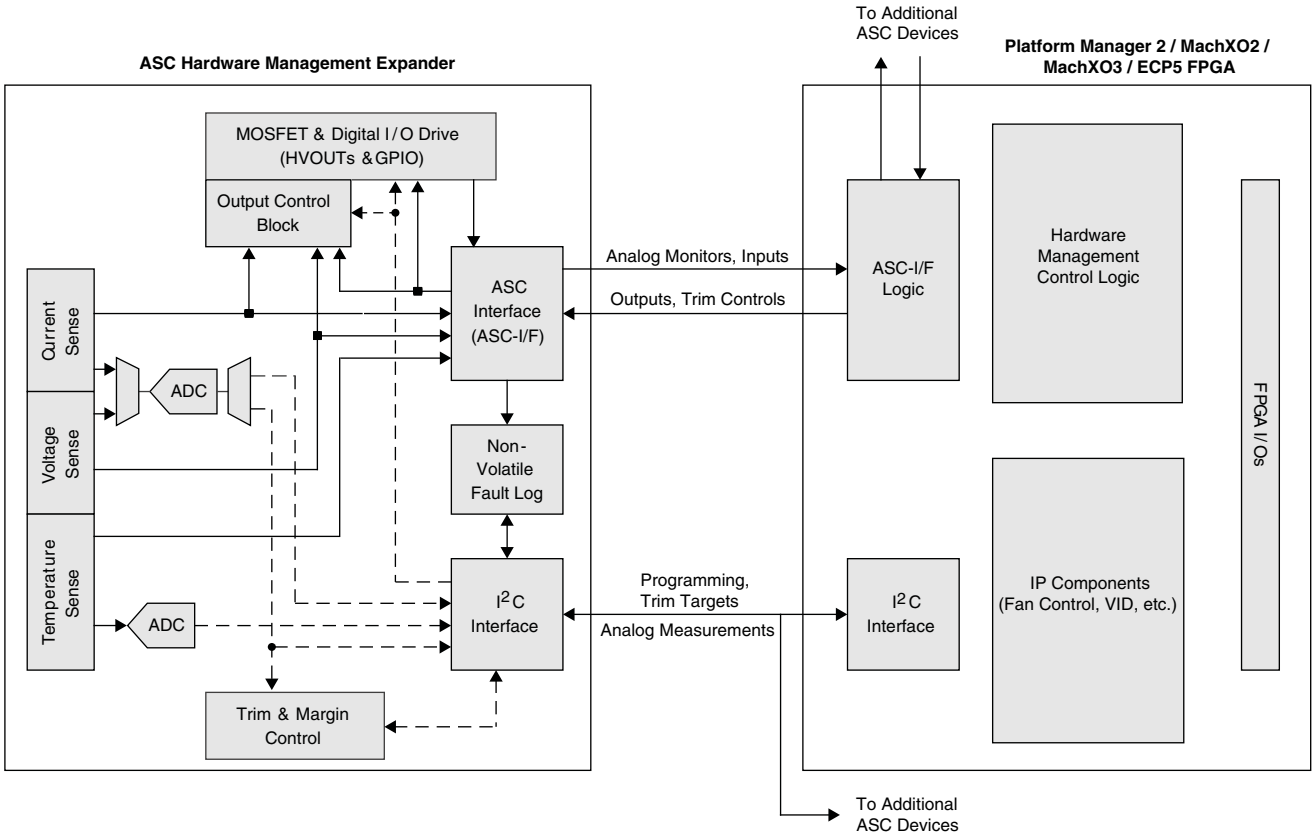
Theory of Operation

Hardware Management System

The ASC Hardware Management Expander is designed to seamlessly increase the number of analog sense and control channels in the hardware management section of a circuit board. The device functions as a hardware management expander in systems with the Lattice Platform Manager 2, MachXO2, MachXO3, or ECP5 FPGAs. The functional blocks for analog voltage, current and temperature monitoring, measurement, and control are built into the ASC. The ASC depends on the Platform Manager 2, MachXO2, MachXO3, or ECP5 FPGA to interpret the analog monitor status signals and provide control commands.

The Platform Manager 2, MachXO2, MachXO3, or ECP5 FPGA includes the hardware management control logic and other plug-in IP components to support functions like Fan Control, or Voltage by Identification (VID). ASC devices can be added to the hardware management system to scale with application requirements and are connected to the same Platform Manager 2, MachXO2, MachXO3, or ECP5 FPGA. This architecture supports a single centralized hardware management logic design, with up to eight distributed ASC devices. The basic system concept is shown in Figure 5. The connections are described in detail in the System Connections section.

Figure 5. Hardware Management System with ASC Hardware Management Expander

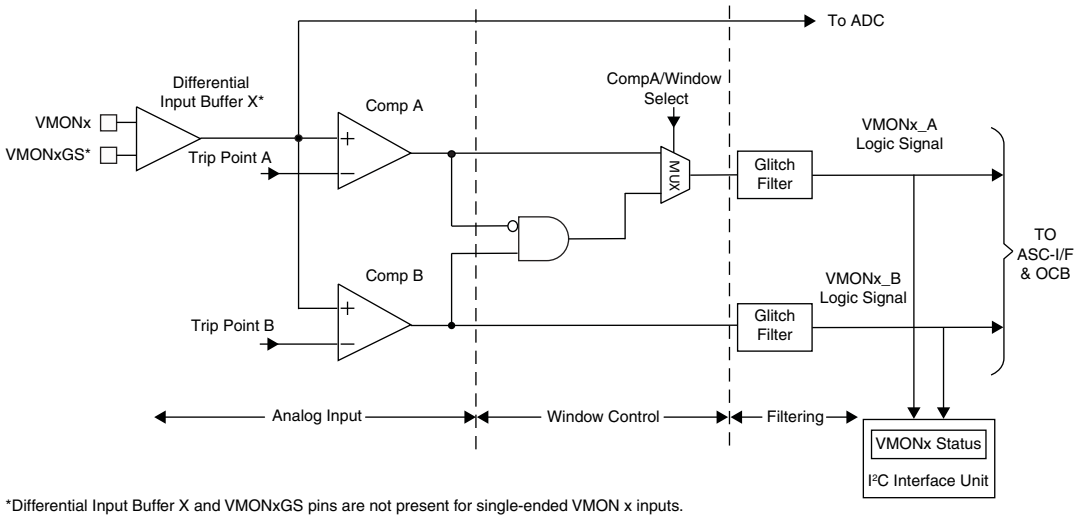


The Hardware Management System is configured using Platform Designer, a part of Lattice Diamond software. Platform Designer provides an easy to use graphical and spreadsheet based interface. Platform Designer automatically generates the device memory configuration based on the options selected in the software. See the For Further Information section for more details.

Voltage Monitor Inputs

The ASC provides ten independently programmable voltage monitor input circuits. There are nine standard voltage channels and one high voltage channel. The standard voltage channels are shown in Figure 6, while the high voltage channel is described in the High Voltage Monitor section. Two individually programmable trip-point comparators are connected to each voltage monitoring input. Each comparator reference has programmable trip points over the range of 0.075 V to 5.734 V. The 75 mV 'zero-detect' threshold allows the voltage monitors to determine if a monitored signal has dropped to ground level. This feature is especially useful for determining if a power supply's output has decayed to a substantially inactive condition after it has been switched off.

Figure 6. ASC Voltage Monitors



*Differential Input Buffer X and VMONxGS pins are not present for single-ended VMON x inputs.

Figure 6 shows the functional block diagram of one of the nine voltage monitor inputs - 'x' (where x = 1...9). Each voltage monitor can be divided into three sections: Analog Input, Window Control, and Filtering. The first section provides a differential input buffer to monitor the power supply voltage through VMONx (to sense the positive terminal of the supply) and VMONxGS (to sense the power supply ground). Differential voltage sensing minimizes inaccuracies in voltage measurement with ADC and monitor thresholds due to the potential difference between the ASC device ground and the ground potential at the sensed node on the circuit board.

The voltage output of the differential input buffer is monitored by two individually programmable trip-point comparators, shown as Comp A and Comp B. The differential input buffer shown above is not present for any of the single-ended VMON inputs. VMON1-4 are differential inputs, while VMON5-10 are single-ended.

Each comparator outputs a HIGH signal to the ASC-I/F if the voltage at its positive terminal is greater than its programmed trip point setting; otherwise it outputs a LOW signal. The VMON4A and VMON9A comparators also output their status signals to the OCB.

Hysteresis is provided by the comparators to reduce false triggering as a result of input noise. The hysteresis provided by the voltage monitor is a function of the input divider setting. Table 1 lists the typical hysteresis versus voltage monitor trip-point.

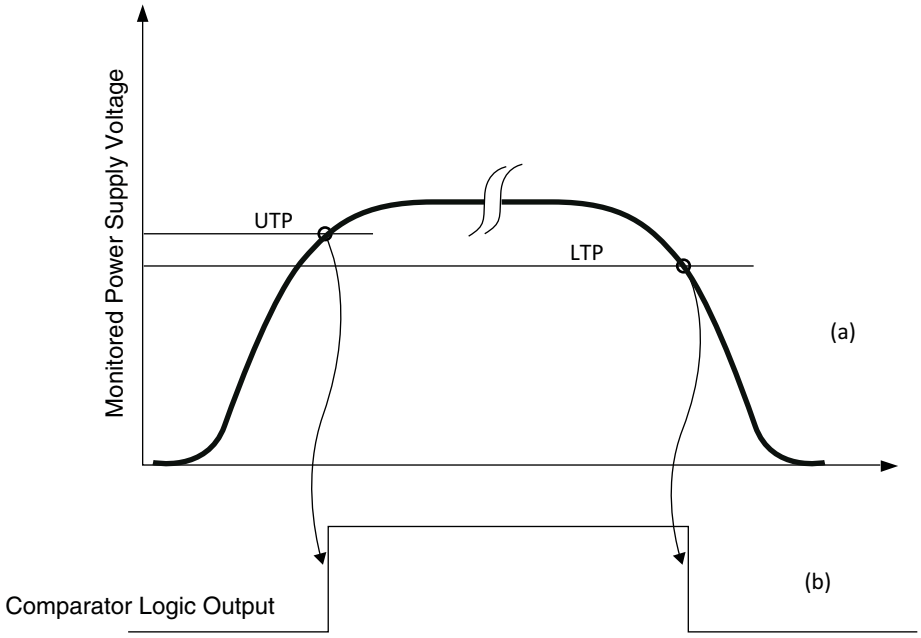
AGOOD Logic Signal

All the VMON, IMON and TMON comparators auto-calibrate following a power-on reset event. During this time, the digital glitch filters are also initialized. This process completion is signaled by an internally generated logic signal: AGOOD. The ASC-I/F will not begin communicating valid VMON status bits or receiving GPIO control signals until the AGOOD signal is initialized.

Programmable Over-Voltage and Under-Voltage Thresholds

Figure 7 shows the power supply ramp-up and ramp-down voltage waveforms. Because of hysteresis, the comparator outputs change state at different thresholds depending on the direction of excursion of the monitored power supply.

Figure 7. Power Supply Voltage Ramp-up and Ramp-down Waveform and the Resulting Comparator Output (a) and Corresponding to Upper and Lower Trip Points (b)



During power supply ramp-up the comparator output changes from logic zero to one when the power supply voltage crosses the upper trip point (UTP). During ramp down the comparator output changes from logic state one to zero when the power supply voltage crosses the lower trip point (LTP). To monitor for over voltage fault conditions, the UTP should be used. To monitor under-voltage fault conditions, the LTP should be used. The upper and lower trip points are automatically selected in software depending on whether the user is monitoring for an over-voltage condition or an under-voltage condition. Table 1 shows the comparator hysteresis versus the trip-point range.

Table 1. Voltage Monitor Comparator Hysteresis vs. Trip-Point

Trip-point Range (V)		Hysteresis (mV)
Low Limit	**High Limit**	
0.66	0.79	8
0.79	0.9	10
0.94	1.12	12
1.12	1.33	14
1.33	1.58	17
1.58	1.88	20
1.88	2.24	24
2.24	2.66	28
2.66	3.16	34
3.16	3.76	40
4.05	4.82	51
4.82	5.73	61
0.075	0.57	0 (Disabled)

The window control section of the voltage monitor circuit is an AND gate (with inputs: an inverted COMPA “ANDed” with COMPB signal) and a multiplexer that supports the ability to develop a ‘window’ function in hardware. Through the use of the multiplexer, voltage monitor’s ‘A’ output may be set to report either the status of the ‘A’ comparator, or the window function of both comparator outputs. The voltage monitor’s ‘A’ output indicates whether the input signal is between or outside the two comparator thresholds. **Important:** This windowing function is only valid in cases where the threshold of the ‘A’ comparator is set to a value higher than that of the ‘B’ comparator. Table 2 shows the operation of window function logic.

Table 2. Voltage Monitoring Window Logic

Input Voltage	Comp A	Comp B	Window (B and Not A)	Comment
V_{IN} < Trip-Point B < Trip-Point A	0	0	0	Outside window, low
Trip-Point B <V_{IN} < Trip-Point A	0	1	1	Inside window
Trip-Point B < Trip-Point A < V_{IN}	1	1	0	Outside window, high

Note that when the ‘A’ output of the voltage monitor circuit is set to windowing mode, the ‘B’ output continues to monitor the output of the ‘B’ comparator. This can be useful in that the ‘B’ output can be used to augment the windowing function by determining if the input is above or below the windowing range.

The third section in the voltage monitor circuit is a glitch filter. When enabled, glitches of less than 64 µs will not result in the comparator output changing. This results in a comparator output delay of 64 µs (typical) for all comparator transitions. This is especially useful for reducing the possibility of false triggering from noise that may be present on the voltages being monitored. When the filter is disabled, the comparator output will be delayed by 16 µs (typical). See the Propagation Delays section for more details.

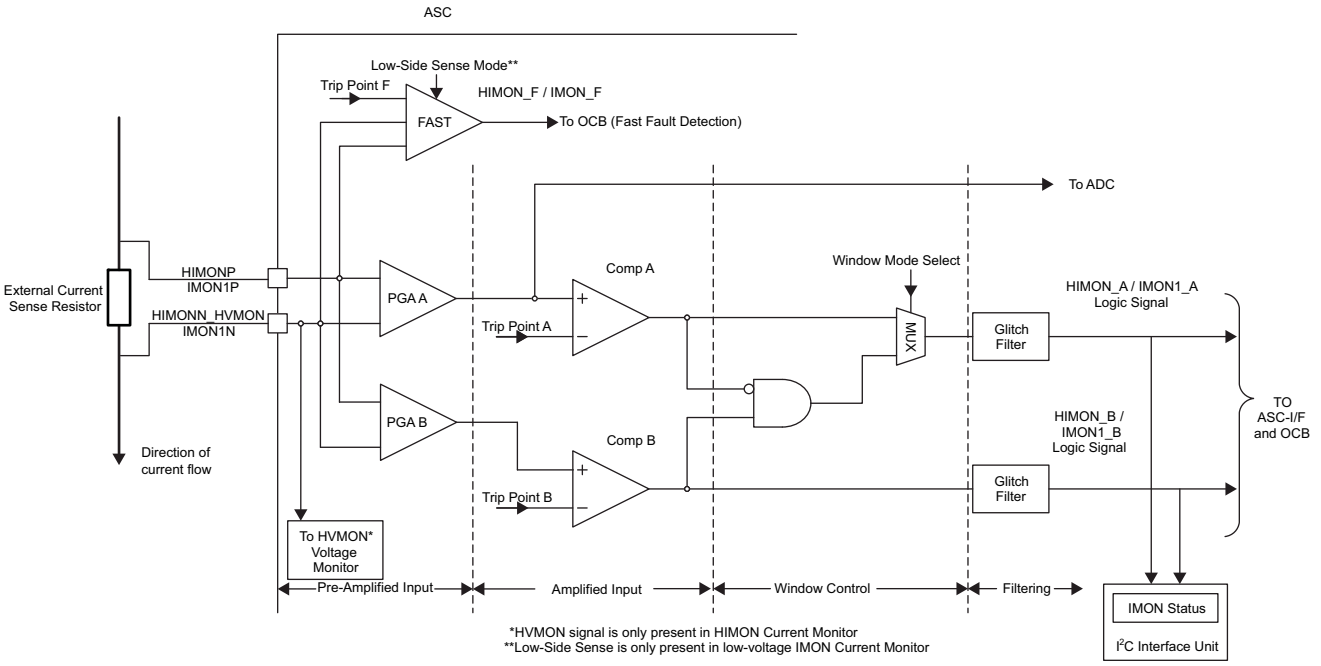
The comparator status can be read from the I^2C interface. For details on the I^2C interface, please refer to the I^2C Interface section of this data sheet.

Current Monitor Inputs

The ASC provides two current monitor circuits as shown in Figure 8. This includes a low-voltage current monitor (with a common mode voltage up to around 6V, see V_{IN_IMONP} in Recommended Operating Conditions section) and a high-voltage current monitor (with a common mode voltage range of up to around 13 V, see V_{IN_HIMONP} in Recommended Operating Conditions section). The low-voltage and high-voltage current monitors share the same basic functional blocks, which are described in this section. Only the low-voltage current monitor supports the low-side sensing mode (shown in Figure 8). The high-voltage current monitor shares input pins with the high-voltage monitor described in the next section.

The current monitor circuits have a differential input that is connected to an external shunt resistor. The differential input goes to a pair of programmable gain amplifiers (PGA) and a fast comparator. The output of PGA A is connected to the ADC and the programmable trip point comparator A. The output of PGA B is connected to the programmable trip point comparator B. The output of the fast comparator is routed to the on-chip Output Control Block (OCB). This signal is useful for fast overcurrent or short circuit shutdown scenarios.

Figure 8. ASC Current Monitor



The Current Monitors can be divided into four sections: Pre-Amplified Input, Amplified Input, Window Control, and Filtering. The first section includes the differential input pins IMON1P and IMON1N (low-voltage current monitor) or HIMONP and HIMONN_HVMON (high-voltage current monitor). These pins are connected to the PGA circuits as well as the direct differential connection to the Fast Fault Detector.

The differential input is monitored by the fast fault detector. The fast fault detector has coarse accuracy and eight programmable trip points. The key feature of the fast fault detector is its response time. The fast fault detector outputs a HIGH signal to the OCB if the differential voltage across the current sensing shunt exceeds the programmed trip point setting. The current shunt is normally connected on the high-side of the input voltage. However, the low-voltage current monitor also supports low-side sensing. The low-side sensing mode should be enabled when sensing negative voltage supplies (such as –48 V) or if the current sense resistor is placed in the return line between the load and ground. This insures proper operation of the fast comparator in a low-side sensing circuit.

Table 3 shows the available trip points for the fast fault detector vs. three frequently used sense resistor values.

Table 3. Fast Fault Detector Current Trip Points vs. Frequently Used Sense Resistor Values

Trip Point Setting	Frequently Used Sense Resistor Value		
	1 Milliohm	5 Milliohm	10 Milliohm
50 mV	50 A	10 A	5 A
100 mV	100 A	20 A	10 A
150 mV	150 A	30 A	15 A
200 mV	200 A	40 A	20 A
250 mV	250 A	50 A	25 A
300 mV	300 A	60 A	30 A
400 mV	400 A	80 A	40 A
500 mV	500 A	100 A	50 A

The Programmable Gain Amplifiers have gain settings of 10x, 25x, 50x, and 100x. The PGA circuits amplify the voltage differential across the current shunt and pass the results to the amplified input section of the current monitor.

The Amplified Input section provides two individually programmable trip-point comparators, shown as Comp A and Comp B above. Each comparator supports four different trip points. Combining these trip points with the respective PGA settings, 16 unique threshold levels are selected for each current monitor.

Table 4 shows the available voltage differential trip points.

Table 4. Comparator Trip Points

Trip Point Setting	Programmable Gain Amplifier Setting (V/V)			
	10 x	**25 x**	**50 x**	**100 x**
1	75 mV	30.5 mV	15.5 mV	8 mV
2	100 mV	40.5 mV	20.5 mV	10.5 mV
3	140 mV	56.5 mV	28.5 mV	14.5 mV
4	190 mV	77 mV	39 mV	20 mV

The output of PGA A is also passed to the on-chip ADC. The current is measured and averaged by the ADC at regular intervals, as described in the Current Measurement with ADC section of the datasheet.

The window control section of the current monitor circuit is an AND gate (with inputs: an inverted COMPA “ANDed” with COMPB signal) and a multiplexer that supports the ability to develop a ‘window’ function in hardware, similar to the voltage monitor window function. Through the use of the multiplexer, the current monitor’s ‘A’ output may be set to report either the status of the ‘A’ comparator, or the window function of both comparator outputs. The current monitor’s ‘A’ output indicates whether the input signal is between or outside the two comparator thresholds. Important: This windowing function is only valid in cases where the threshold of the ‘A’ comparator is set to a value higher than that of the ‘B’ comparator. Table 5 shows the operation of window function logic.

Table 5. IMON Window Mode Behavior

Input Voltage	Comp A	Comp B	Window (B and Not A)	Comment
IIN < Trip-Point B < Trip-Point A	0	0	0	Outside window, low
Trip-Point B <IIN < Trip-Point A	0	1	1	Inside window
Trip-Point B < Trip-Point A < IIN	1	1	0	Outside window, high

Note that when the ‘A’ output of the current monitor circuit is set to windowing mode, the ‘B’ output continues to monitor the output of the ‘B’ comparator. This can be useful in that the ‘B’ output can be used to augment the windowing function by determining if the input is above or below the windowing range.

The fourth section in the current monitor circuit is a glitch filter. When enabled, glitches of less than 64 µs will not result in the comparator output changing. This results in a comparator output delay of 64 µs (typical) for all comparator transitions. This is especially useful for reducing the possibility of false triggering from noise that may be present on the currents being monitored. When the filter is disabled, the comparator output will be delayed by 16 µs (typical). See the Propagation Delays section for more details.

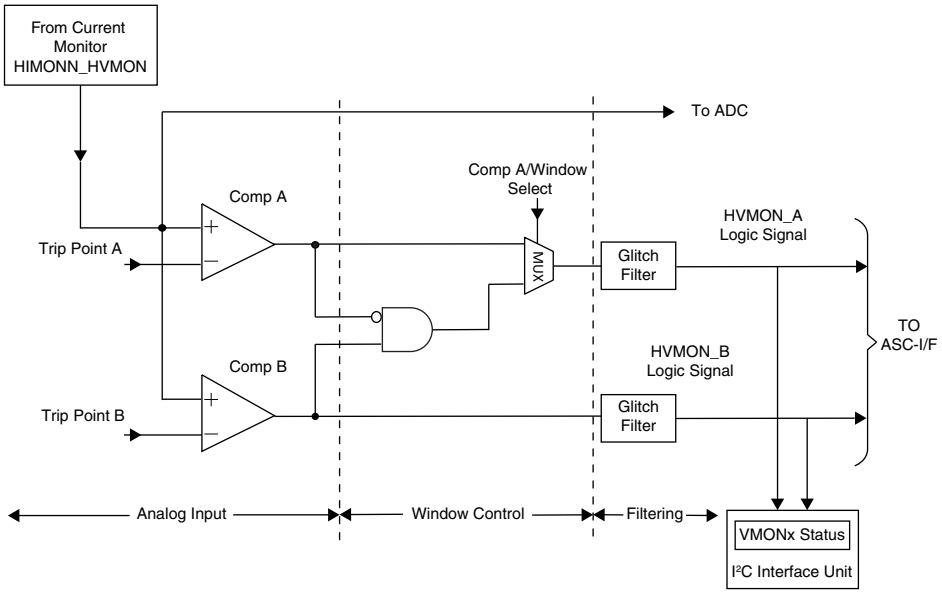
The comparator status can be read from the I^2C interface. For details on the I^2C interface, please refer to the I^2C Interface section of this data sheet.

High Voltage Monitor

The High Voltage Monitor circuit is a single-ended high voltage monitor (HVMON) which is connected to the same input pin as the High Voltage Current Monitor (HIMONN_HVMON). Figure 9 shows the single-ended monitor circuit, which monitors the voltage on the HIMON pin. Extending the input voltage range with an external voltage

divider as described in AN6041, Extending the VMON Input Range of Power/Platform Management Devices is not recommended for the HVMON inputs.

Figure 9. HVMON Monitor Circuit



The HVMON follows the same structure as the Voltage Monitor circuits. Two individually programmable trip-point comparators are connected to the HIMONN_HVMON pin voltage. Each of the comparator references has 408 programmable trip points, over a range of 0.227 V to 13.226 V.

The functional block diagram, shown in Figure 9, is a similar structure to the other single-ended Voltage Monitor circuits. Each comparator outputs a HIGH signal to the ASC-I/F if the voltage at its positive terminal is greater than its programmable trip point setting. A hysteresis of approximately 1% of the setpoint is provided by the comparators to reduce false triggering. Table 6 shows a typical hysteresis versus voltage monitor trip point.

Table 6. HVMON Hysteresis vs Trip Point Range

Trip-point Range (V)		Hysteresis (mV)
Low Limit	**High Limit**	
1.91	2.27	22
2.27	2.7	28
2.69	3.2	30
3.16	3.76	38
3.72	4.43	44
4.40	5.24	52
5.18	6.17	61
6.04	7.20	72
7.08	8.43	84
8.29	9.87	99
9.68	11.52	115
11.17	13.2	133
0.23	1.28	0 (Disabled)

The Over-Voltage and Under-Voltage thresholds, along with the window mode and glitch filter, are identical to the features described in the voltage monitor section.

VMON and IMON Measurement with the On-Chip Analog to Digital Converter (ADC)

The ASC has an on-chip analog to digital converter that can be used for measuring the voltages at the VMON inputs or the currents at the IMON inputs. This ADC is also used in closed loop trimming of DC-DC converters. Closed loop trimming is covered later in this document.

Figure 10. ADC Monitoring VMON and IMON

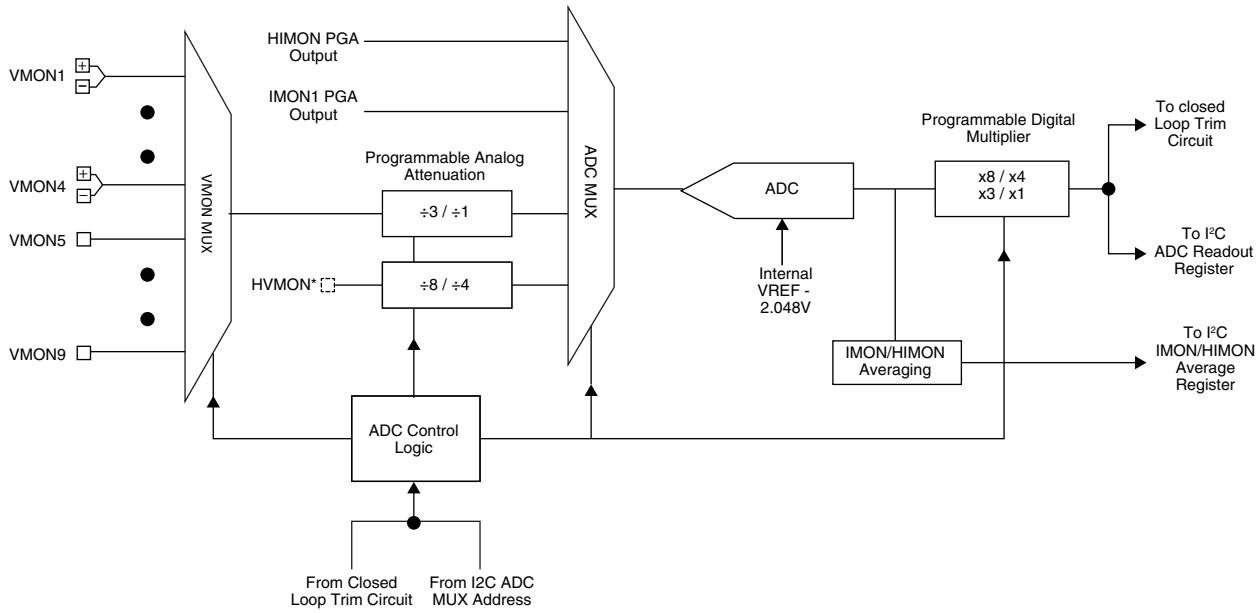


Figure 10 shows the ADC circuit arrangement within the ASC device. The ADC can measure all analog input voltages up to 2.048V through the multiplexer, ADC MUX. The ADC MUX receives inputs from the High Voltage IMON Programmable Amplifier (PGA), the IMON1 PGA, the High Voltage Monitor (HVMON) at the HIMONN_HVMON pin, and the VMON MUX. The VMON voltages can be attenuated (divided by three) or unattenuated (divided by one). The divided-by-three setting is used to measure voltages from 0 V to 6 V range and divided-by-one setting is used to measure the voltages from 0 V to 2 V range. The HVMON voltage requires attenuation, with settings for divided by eight (voltages between 8 V and 13.2 V) or divided by four (voltages between 8 V and 0 V). The HIMON and IMON1 PGA output voltages must be kept below 2.0 V for proper ADC operation since they are not attenuated.

The ADC control logic manages the MUX and attenuation settings. The control logic manages conversion requests from I^2C and the Closed Loop Trim Circuit. The control logic also schedules regular IMON1 and HIMON conversions, which are subsequently averaged and stored for user access. These IMON conversions are configured through the I^2C bus and filtered using an eight sample, weighted averaging scheme.

The control logic also sets the digital multiplication factor. This results in VMON and HVMON voltages, regardless of attenuation setting, maintaining a 2 mV per LSB scale. (See Calculation section for more details). The IMON1/HIMON voltages are not multiplied.

A microcontroller or FPGA IP can place a request for any VMON or IMON voltage measurement at any time through the I^2C bus. After the receipt of an I^2C command, the control logic will connect the ADC to the I^2C selected VMON or IMON through the ADC MUX. The ADC output is then latched into the I^2C readout registers.

Calculation

The algorithm to convert the ADC code to the corresponding VMON / HVMON voltage takes into consideration the relevant attenuation setting. In other words, if the attenuation is set to divide-by-eight, then the 10-bit ADC result is automatically multiplied by eight to calculate the actual voltage at that VMON input. Thus, the I^2C readout register is 13 bits instead of 10 bits. The other attenuator settings are also automatically compensated using the digital multiplier. The following formula can always be used to calculate the actual voltage from the ADC code.

Voltage at the VMONx Pins

VMON= ADC code (13 bits, converted to decimal) * 2 mV

The ADC code includes the ADC_VALUE_HIGH (8 bits) and ADC_VALUE_LOW (5 bits) read from I^2C interface

Calculating the HIMON or IMON1 current is slightly more complex, and requires knowledge about the current PGA setting and the resistance value of the current sense shunt resistor. The PGA has four settings (x10, x25, x50, and x100) which are automatically set in the software based on the trip point selection, while the current sensing resistance is chosen by the customer.

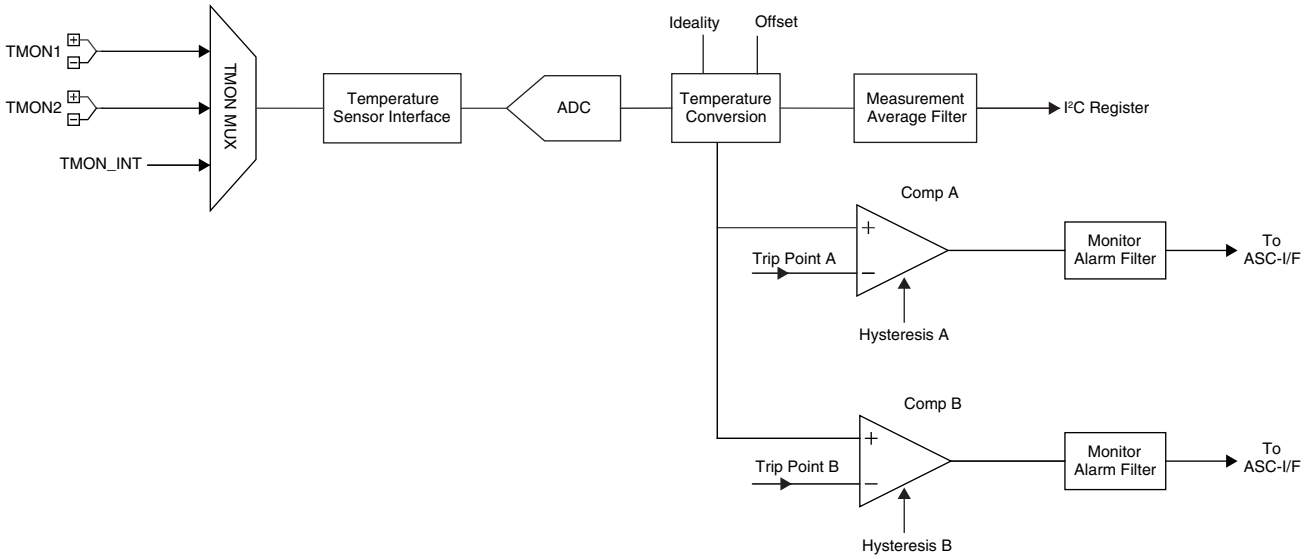
Current at the HIMON / IMON1 Pins

IMON current = ADC code (13 bits, converted to decimal) * 2 mV / (PGAsetting x Rsense)

Temperature Monitor Inputs

The ASC provides two external temperature monitor inputs and one internal temperature monitor as shown in Figure 11.

Figure 11. Temperature Monitor



The independently programmable temperature monitor inputs can be used with internal substrate diodes on microprocessors, FPGAs, ASICs, or with low cost external NPN or PNP transistors. The temperature sensor interface block includes programmable support for a variety of sensor configurations as shown in Figure 12. The sensor configuration settings available in the design software are described in Table 7.

Figure 12. Remote TMON Diode Configurations

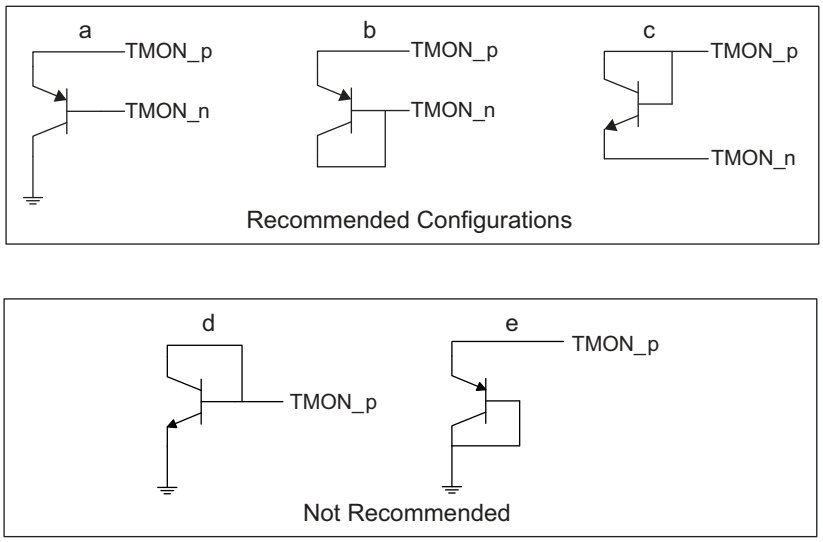


Table 7. Remote TMON Diode Configurations

Sensor Configuration	Figure Number	Auto-β Compensation	Series Resistance Compensation	Accuracy
Beta Compensated PNP	12-a	Effective	Effective	Specified in recommended operating conditions
Differential PNP or NPN or Diode	12-b / 12-c	Not effective	Effective	Dependent on β variance
Single Ended	12-d / 12-e	Not effective	Not Effective	Not specified

The temperature sensor interface block also has built-in circuits to automatically compensate for the series resistance of the PCB traces to the sensor as well as the intrinsic device resistance. In addition, the interface block has circuits to compensate for the variable Beta (β) of the transistor sensor when it is connected in the configuration shown in Figure 12-a. (In order for the variable β compensation circuit to be effective it must be able to measure the base current separately from the collector current.) For a discrete PNP or NPN transistor with high β (approximately 100 or greater) the effect of variable beta is typically negligible. However, most substrate diode temperature sensors will have a low β value which can vary considerably over temperature and current density making this a very useful feature.

The temperature signal information is converted to digital data by the dedicated TMON ADC. The digital data is scaled and converted to a two's complement, 11-bit temperature reading by the Temperature Conversion block. The measurement resolution is 0.25 ºC per bit. The temperature conversion block takes into account the user entered ideality factor and offset value.

The ideality factor (also known as the emission coefficient or the N-factor) is a measure of how closely a real diode follows the ideal diode equation. In a real diode imperfections allow some recombination to occur in the junctions or by other methods which are not accounted for in the ideal equation. The ASC temperature conversion block is optimized for an ideality factor of 1 so any errors in the actual ideality factor of the sensor will produce a proportional error in the temperature value (in Kelvins). The diode ideality factor can be programmed in the range from 0.9 to 2.0 to match the actual ideality factor of the sensor.

An approximate value for the ideality factor for a 2N3904 NPN transistor is 1.004 and for a 2N3906 PNP transistor is 1.008. A substrate diode temperature sensor will typically have an ideality factor published in its data sheet.

Uncertainty can be introduced in temperature measurement by using an approximate value rather than the actual value for a 2N3904 or a 2N3906 transistor. This can lead to an error of around 0.4 °C. If the ideality factor for the transistor being used is not published it can be determined by the ASC using the following procedure.

1. Force the system temperature to a known value (Tref).
2. With the ASC ideality factor set to 1.000, record the temperature value calculated (Tnocal).
3. Convert the Tref and Tnocal to Kelvin.
4. Divide the Tnocal (K) by Tref (K).

The result will be the actual ideality factor to be entered for the given TMON channel in the design software.

Note: The calibration is only as accurate as the Tref value. Any errors in the test equipment used will be transferred to the ASC readings.

The temperature conversion block also provides user programmable temperature offset from –64 °C to 63.75 °C for each channel's digital data to mitigate errors due to self-heating of the sensor, systematic offset and other unforeseen errors.

The conversion block also includes programmable output values for detected short or open conditions at the monitor input. The output levels are shown in Table 8.

Table 8. Temperature Measurement Fault Readings

Fault	Short	Open
0	–255.75 °C	255.75 °C
1	255.75 °C	–255.75 °C

The converted temperature data for each channel is stored in two registers and can be read out via I^2C, after the programmable Measurement Averaging filter. For more details about the data register format, please refer to Measurement and Control Register Access in the I^2C Interface section of this data sheet.

The programmable measurement filter performs exponential averaging. Data is available immediately after one update cycle and is continually averaged using the programmable filter coefficients of 1, 8, or 16 per channel. The filtering equation is shown below:

$$TempAve[x] = \frac{TempMeas[x]}{FiltCo} + TempAve[x-1] \times \frac{FiltCo-1}{FiltCo}$$

When the temperature input changes it will require some settling time for the new value to be fully reflected in the results register due to the averaging filter. The settling time will vary depending upon how many channels are enabled and the programmed averaging coefficient. The settling time for various averaging coefficients and number of channels is shown in Table 9.

Table 9. Temperature Measurement Settling Time[1]

Measurement Averaging Coefficient	Number of Channels Enabled	Average Settling Time (ms)
1	1	14.2
8	1	114
16	1	228
1	2	28.4
8	2	227
16	2	454
1	3	42.6
8	3	341
16	3	682

1. Values are approximate and are not guaranteed by characterization.

In addition to the direct temperature measurement the ASC has a temperature comparison function. The digital data of each channel is monitored by two trip-point comparators, shown as Comp A and Comp B in Figure 11. The digital temperature data monitored at the comparators is not processed by the measurement averaging filters. Each comparator reference has programmable trip points over the range of –64 ºC to 155 ºC with resolution of 1ºC. Whenever the monitored temperature is above the trip point, the comparator output is set to one. The comparator outputs are transmitted over the ASC-I/F to the FPGA, depending on the setting of the Alarm Filter.

The two comparators each support programmable Hysteresis of 0 ºC to 63 ºC. When a comparator is used for over-temperature monitoring the programmed hysteresis value is subtracted from the trip point and when the comparator is used for under-temperature monitoring the programmed hysteresis value is added to the trip point. The hysteresis behavior is displayed in Figure 13 and Figure 14.

Figure 13. Monitor Alarm Signal Behavior - Overtemperature (OT) Setting

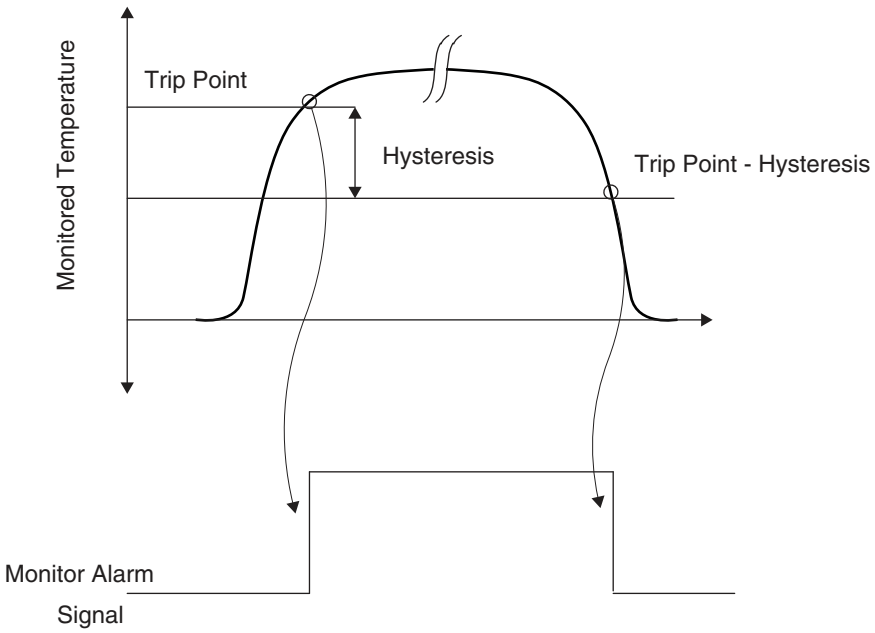
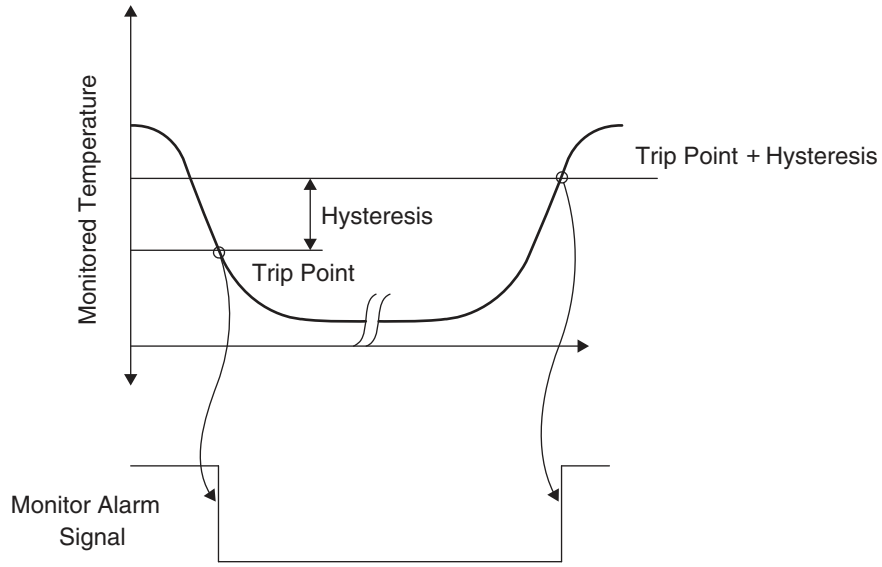


Figure 14. Monitor Alarm Signal Behavior - Undertemperature (UT) Setting



Each comparator can be individually selected as either over-temperature or under-temperature operation.

A programmable Alarm Filter (separate from the measurement averaging filter) is available at the output of the comparators. The depth of this filter is programmable from 1 to 16. The filter monitors the comparator alarm output each time the temperature measurement is refreshed. The filter counts up each time the comparator alarm value is 1, and down each time the comparator alarm value is 0. When the filter counter reaches the programmed filter depth, the TMONx_A or TMONx_B signal are set to one.

Digital Inputs/Outputs

The ASC has four dedicated digital outputs (HVOUTs) and nine General Purpose Input/Output (GPIO) pins. The four HVOUT pins can be configured as high-voltage FET drivers or Open Drain outputs. This provides a high degree of flexibility when interfacing to power supply control inputs or other external logic signals.

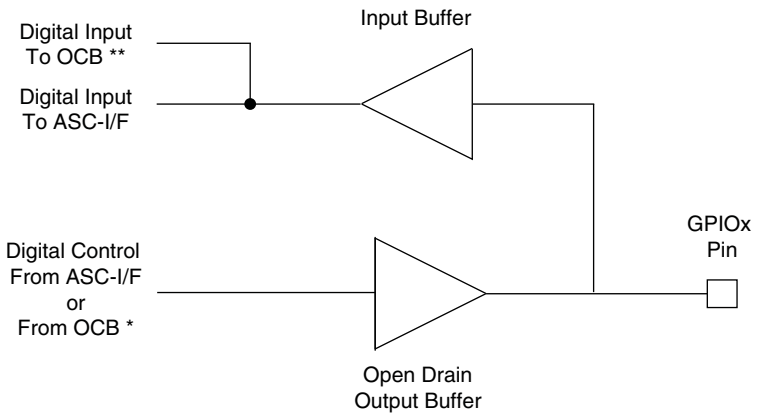
The nine GPIO pins can be configured as inputs or Open Drain outputs. Figure 15 shows a block diagram of the GPIO circuitry. When configured as inputs, GPIO1 through GPIO10 inputs are registered and made available to the FPGA using the ASC-I/F. GPIO5 through GPIO10 are also made available to the Output Control Block (OCB) directly without being registered. When configured as outputs, GPIO1 through GPIO10 are controlled by the ASC-I/F. Table 10 shows a summary of the input and output sources for each GPIO pin.

Table 10. GPIO Input and Output Sources

GPIO	ASC-I/F Input	OCB Input	ASC-I/F output	OCB output	Hysteretic control
GPIO1	Y	N	Y	N	N
GPIO2	Y	N	Y	Y	Y
GPIO3	Y	N	Y	Y	Y
GPIO4	Y	N	Y	N	N
GPIO5	Y	Y	Y	N	N
GPIO6	Y	Y	Y	N	N
GPIO7[1]	N/A	N/A	N/A	N/A	N/A
GPIO8	Y	Y	Y	N	N
GPIO9	Y	Y	Y	N	N
GPIO10	Y	Y	Y	N	N

1. GPIO7 is not bonded out.

Figure 15. GPIO Block Diagram



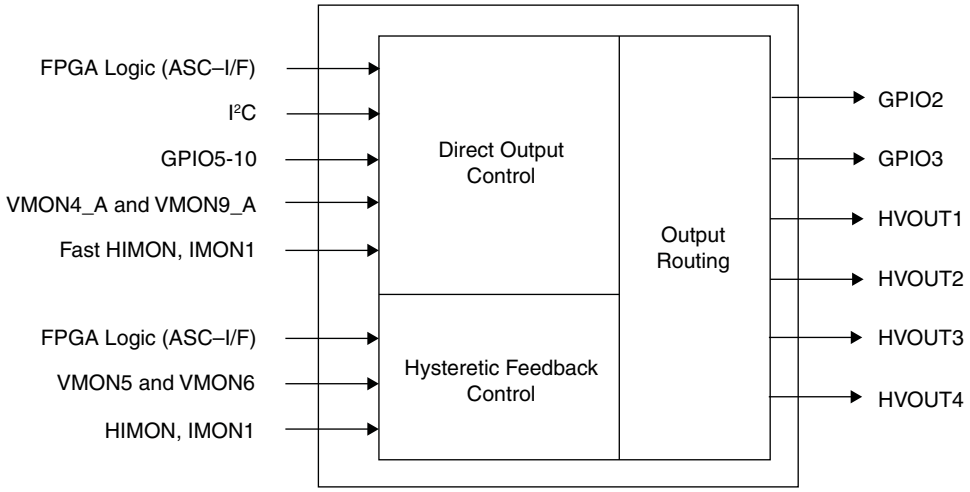
* Digital Control comes from OCB for GPIO 2 and 3.
Digital Control comes from ASC-I/F for remaining GPIO.
** Only available for GPIO 5, 6, 7, 8, 9 and 10.

Output Control Block

The ASC Output Control Block (OCB) is used to control GPIO2, GPIO3, and the four HVOUTs. The Output Control Block has two modes of operation available; Direct Output control and Hysteretic Feedback control as shown in Figure 16.

Direct Output control is supported by various inputs which include the I^2C registers, GPIO pins 5-10 (when configured as inputs), the VMON4A and VMON9A comparator output signals, and the Fast IMON1 and Fast HIMON comparator output signals. These inputs are individually selectable for each of the outputs. When these inputs are used with the Direct Output control mode they provide a fast path for control which has very low propagation delay. The outputs in the OCB can also be controlled from the FPGA Logic over the ASC Interface (ASC-I/F) with the normal propagation delays. See the Propagation Delays section for more details.

Figure 16. Output Control Block – Simplified Diagram



The OCB outputs can also be configured for Hysteretic Feedback control if desired. In the Hysteretic Feedback control mode the output will be switched on and off based upon the feedback signal chosen. The available feedback signals are the HIMON, IMON1, VMON5, or VMON6 trip points. As the feedback signal changes it will turn the output on or off depending upon whether it is above or below the chosen set-point and depending upon the output polarity. The user logic can switch between the high and low trip points of a signal to provide additional flexibility. In addition, the FPGA Logic can be dynamically selected to provide the feedback signal over the ASC-I/F which allows the user logic to change from a conditional output to a static value.

One example of how the Hysteretic feedback control feature can be used is to modulate a high-voltage FET driver using the FPGA logic and the trip points to control the rate of modulation over different voltage ranges - such as in a Hot Swap application. The design software provides an easy to use interface for configuring the device as a hot swap controller. The software will generate the required device settings and control algorithm automatically.

Figure 17. HVOUT Output Routing MUX Block Diagram

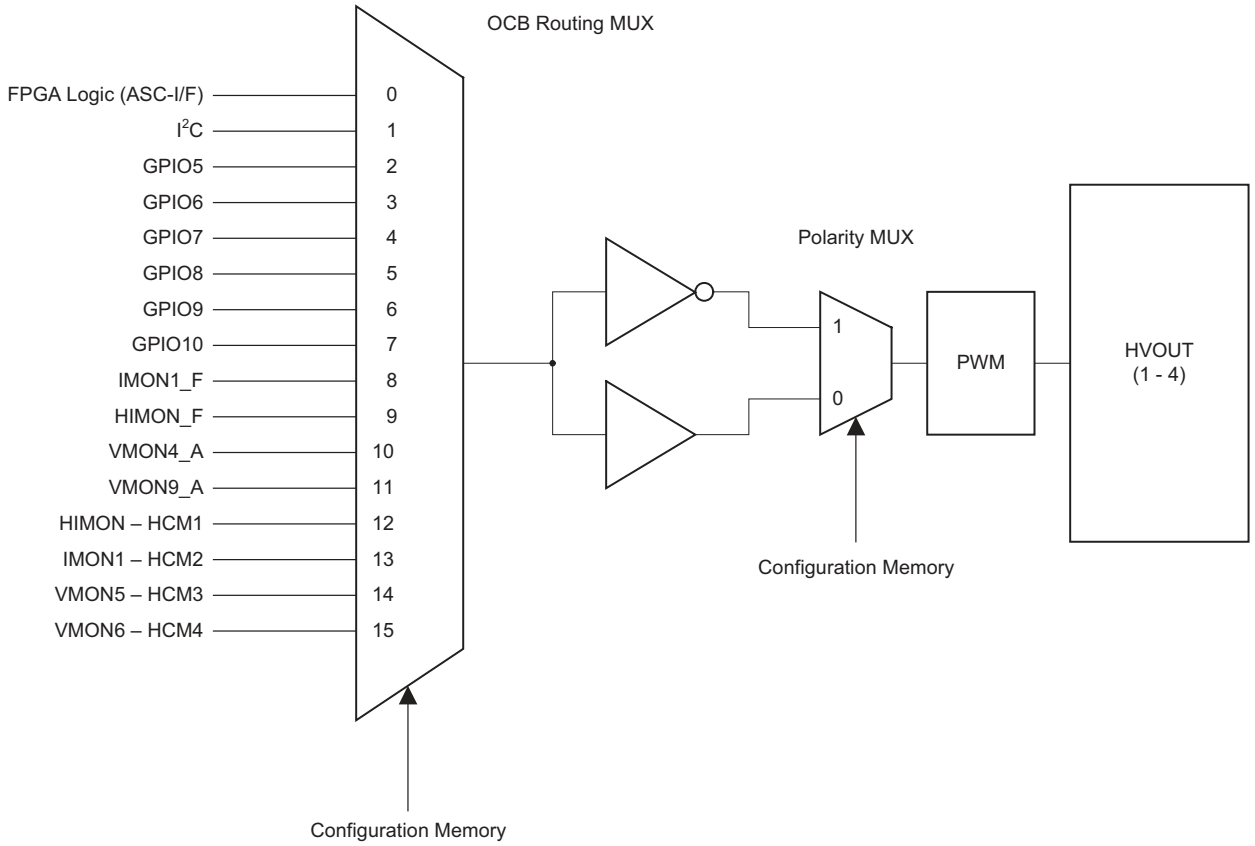


Figure 17 is a generic HVOUT routing diagram that applies to all the HVOUTs and provides a bit more detail than the simplified diagram in Figure 16. The MUX on the left is configured by Platform Designer software and selects from either the 12 direct control signals at the top or the four Hysteretic Control Module (HCM) signals at the bottom. The software is also used to select normal or inverted control. The features and configuration of the PWM and HVOUT blocks are covered in the High Voltage Outputs section. If PWM is enabled then the output of the Polarity MUX will enable or disable the PWM; otherwise the HVOUT will be on or off based on the Polarity Mux output signal.

Figure 18. GPIO Output Routing MUX Block Diagram

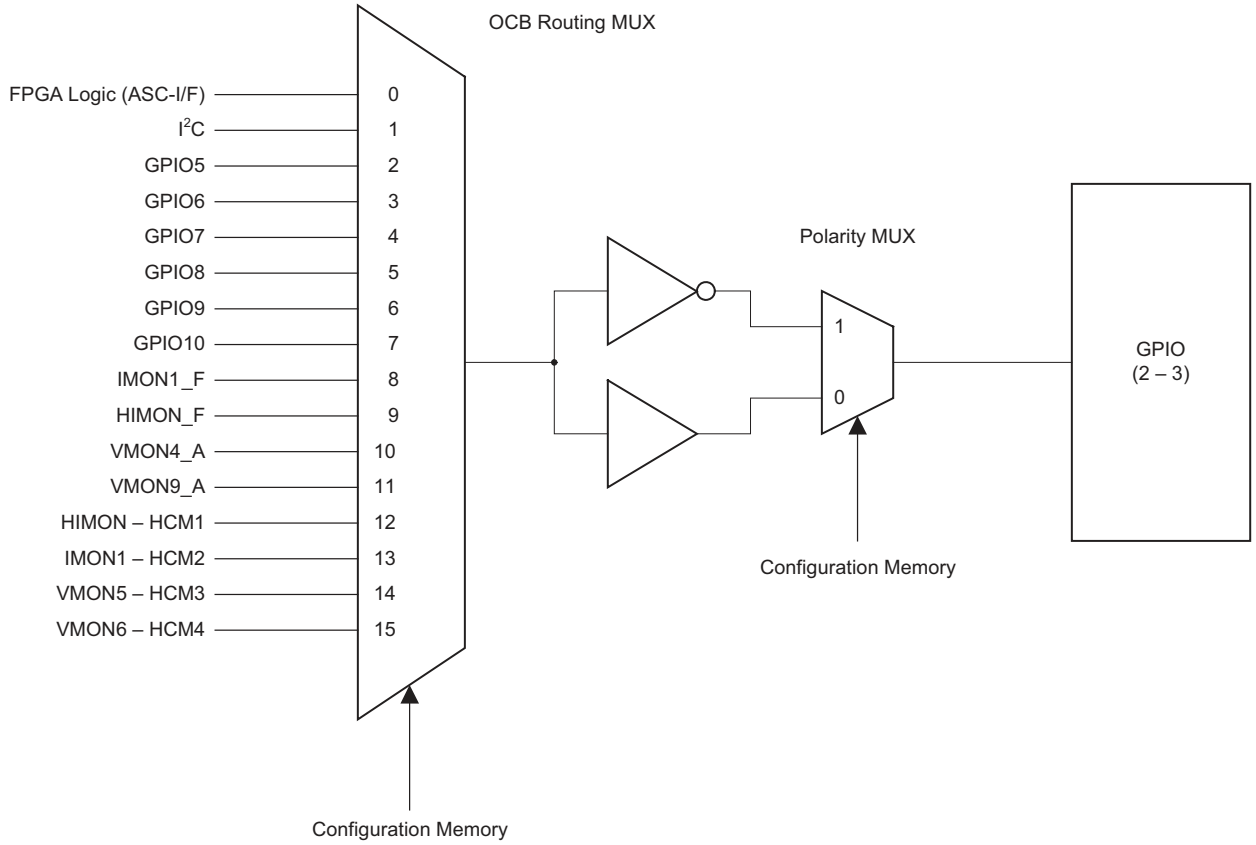
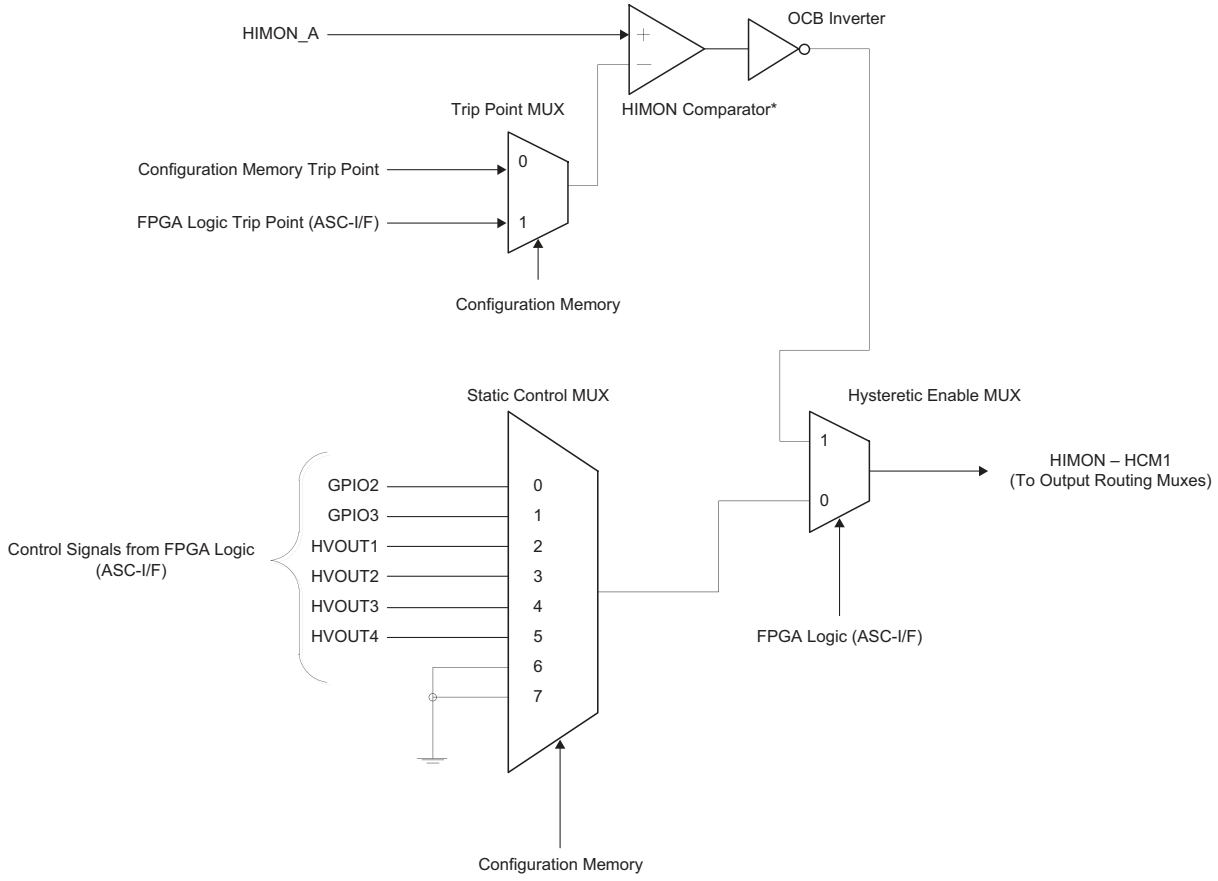


Figure 18 is a generic GPIO routing diagram that applies to GPIO2 & GPIO3 and it provides a bit more detail than the simplified diagram in Figure 16. The MUX on the left is configured by Platform Designer software and selects from either the 12 direct control signals at the top or the four Hysteretic Control Module (HCM) signals at the bottom. The software is also used to select normal or inverted control. The GPIO pin will be on or off based on the Polarity Mux output signal.

Figure 19 is a diagram of Hysteretic Control Module #1 (HCM1) which is a little more complex than the other HCMs in the device. It is unique in that it is the only HCM that supports a dynamic selection of the Trip Point from the table; while the other HCMs can only dynamically switch between two comparator outputs (for example a low and high setting). The software configures the Trip Point MUX to either use a fixed configuration HIMON trip point or the dynamic HIMON trip point which is set by the FPGA Logic based upon the operating conditions. The output of the HIMON comparator is inverted before sending it to the Hysteretic Enable MUX.

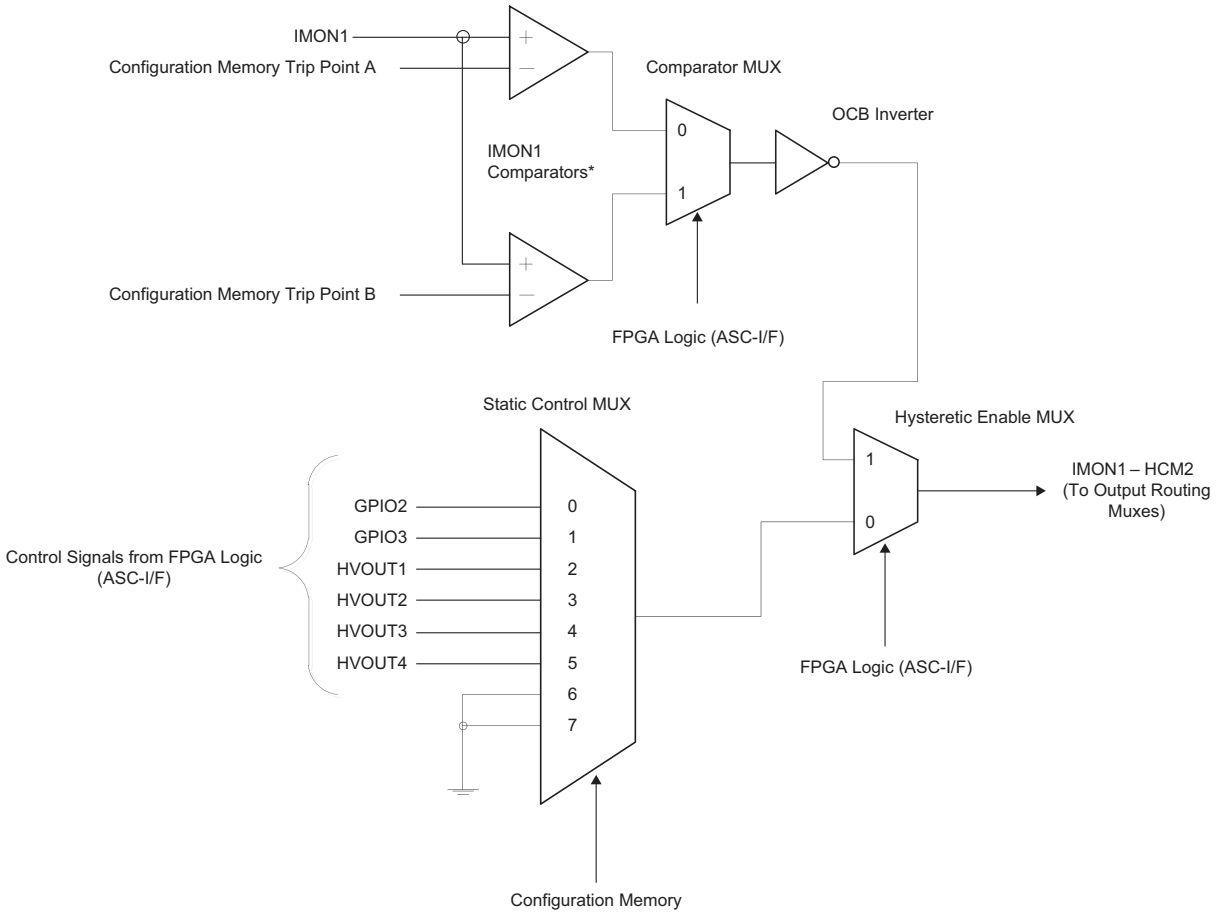
Figure 19. OCB HIMON HCM1 Block Diagram



* HIMON Windowing and Glitch Filters are not shown in this diagram for clarity.
The input to the OCB inverter is from the glitch filter.

The Hysteretic Enable MUX is controlled by the FPGA Logic over the ASC-I/F. The HIMON comparator signal is selected when the Hysteretic mode is enabled and the HVOUT or GPIO is controlled based on the voltage sensed at the HIMON input pins. When the Hysteretic mode is disabled the HVOUT or GPIO is controlled by the output of the Static Control MUX. All the input signals to the Static Control MUX come from the FPGA Logic over the ASC-I/F. Typically the Static Control MUX is configured by the software to connect to the corresponding output being controlled by the HCM. For example, if HCM1 is selected for HVOUT2, then the Static Control MUX would also be set to HVOUT2. In this manner when Hysteretic Mode is disabled it is just like setting the OCB Routing MUX to zero where the FPGA Logic controls the HVOUT or GPIO over the ASC-I/F.

Figure 20. OCB IMON1 HCM2 Block Diagram



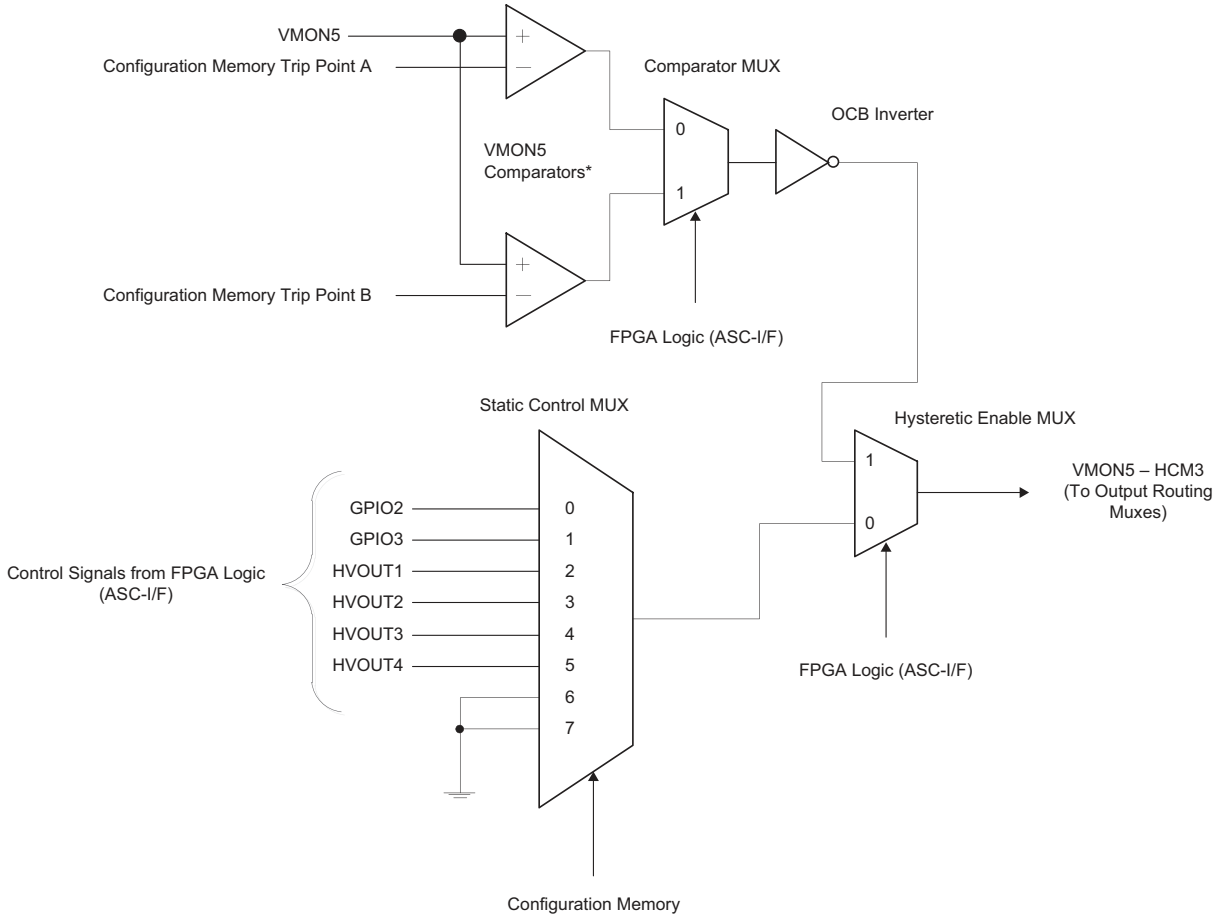
* IMON Windowing and Glitch Filters are not shown in this diagram for clarity.
The input to the comparator MUX is from the glitch filters.

Figure 20 is a diagram of Hysteretic Control Module #2 (HCM2) which is also an IMON based HCM. The software is used to select both the A and B trip points, while the FPGA Logic is used to dynamically switch between the two comparator outputs. The output of the Comparator MUX is inverted before sending it to the Hysteretic Enable MUX.

The Hysteretic Enable MUX is controlled by the FPGA Logic over the ASC-I/F. The Comparator MUX output signal is selected when the Hysteretic mode is enabled and the HVOUT or GPIO is controlled based on the voltage sensed at the IMON1 input pins. When the Hysteretic mode is disabled the HVOUT or GPIO is controlled by the output of the Static Control MUX. All the input signals to the Static Control MUX come from the FPGA Logic over the ASC-I/F. Typically the Static Control MUX is configured by the software to connect to the corresponding output being controlled by the HCM.

When using the hysteretic mode in hot swap applications, the design software will automatically configure the muxes and generate the control algorithm. See the For Further Information section for more details.

Figure 21. OCB VMON5 HCM3 Block Diagram

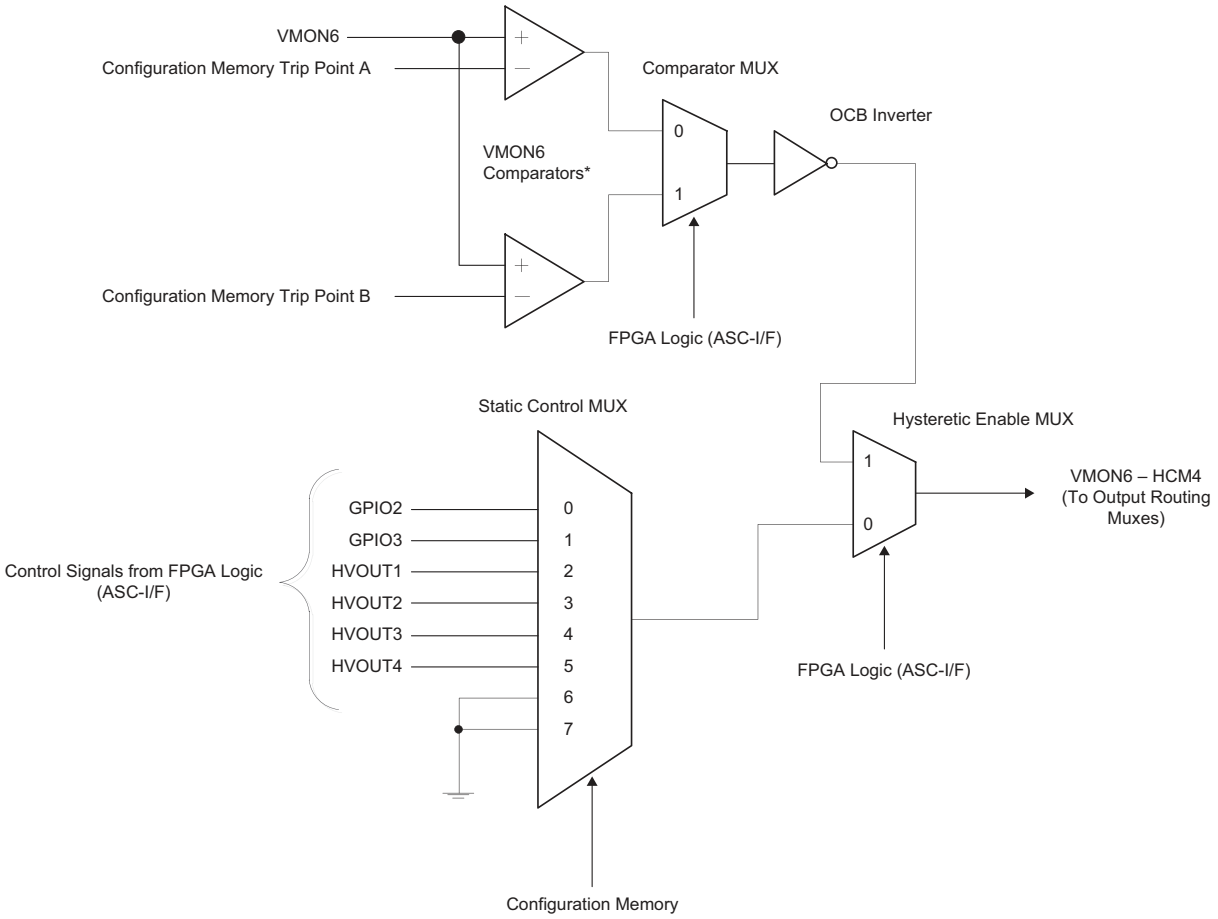


* VMON5 Windowing and Glitch Filters are not shown in this diagram for clarity. The input to the comparator MUX is from the glitch filters.

Figure 21 is a diagram of Hysteretic Control Module #3 (HCM3) which is a VMON based HCM. The software is used to select both the A and B trip points, while the FPGA Logic is used to dynamically switch between the two comparator outputs. The output of the Comparator MUX is inverted before sending it to the Hysteretic Enable MUX.

The Hysteretic Enable MUX is controlled by the FPGA Logic over the ASC-I/F. The Comparator MUX output signal is selected when the Hysteretic mode is enabled and the HVOUT or GPIO is controlled based on the voltage sensed at the VMON5 input pins. When the Hysteretic mode is disabled the HVOUT or GPIO is controlled by the output of the Static Control MUX. All the input signals to the Static Control MUX come from the FPGA Logic over the ASC-I/F. Typically the Static Control MUX is configured by the software to connect to the corresponding output being controlled by the HCM.

Figure 22. OCB VMON6 HCM4 Block Diagram



* VMON6 Windowing and Glitch Filters are not shown in this diagram for clarity. The input to the comparator MUX is from the glitch filters.

Figure 22 is a diagram of Hysteretic Control Module #4 (HCM4) which is a second VMON based HCM. The software is used to select both the A and B trip points but, the FPGA Logic is used to dynamically switch between the two comparator outputs. The output of the Comparator MUX is inverted before sending it to the Hysteretic Enable MUX.

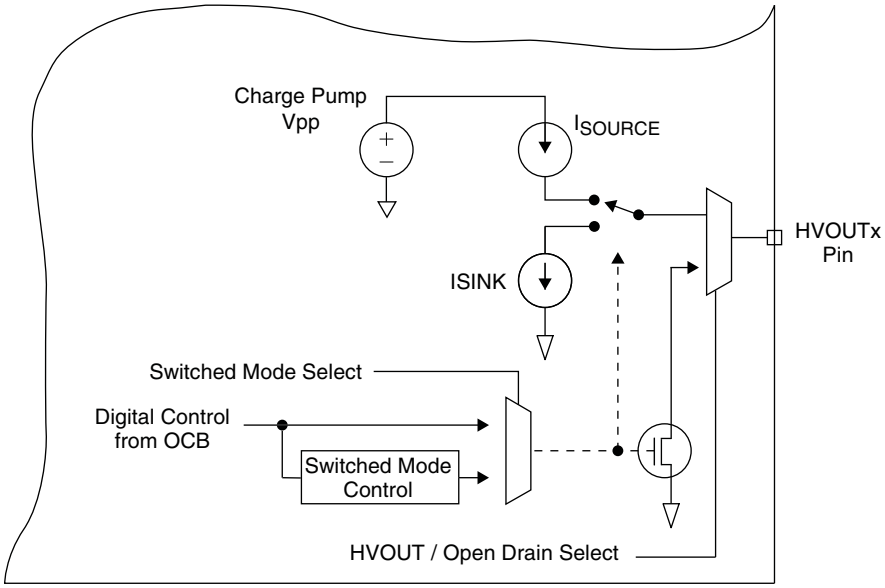
The Hysteretic Enable MUX is controlled by the FPGA Logic over the ASC-I/F. When the Comparator MUX output signal is selected the Hysteretic mode is enabled and the HVOUT or GPIO is controlled based on the voltage sensed at the VMON6 input pins. When the Hysteretic mode is disabled the HVOUT or GPIO is controlled by the output of the Static Control MUX. All the input signals to the Static Control MUX come from the FPGA Logic over the ASC-I/F. Typically the Static Control MUX is configured by the software to connect to the corresponding output being controlled by the HCM.

The Platform Designer software has component interfaces that are used to simplify the task of configuring the OCB blocks discussed in this section. For example, the Hot Swap component provides a functional interface for the designer while setting the trip points for VMONs and IMONs, and routing them to HVOUTs using OCB paths.

High Voltage Outputs

In addition to being usable as digital Open Drain outputs the four HVOUT pins can be configured as high-voltage FET drivers. Figure 23 shows the details of the HVOUT circuitry.

Figure 23. HVOUT Block Diagram



When the HVOUT is configured as a high-voltage FET driver the output either sources current from a charge pump or sinks current. The output level at the pin can rise to a configurable maximum voltage. The maximum voltage levels that are required depend on the gate-to-source threshold of the FET being driven and the power supply voltage being switched. The maximum voltage level needs to be sufficient to bias the gate-to-source threshold on and also accommodate the load voltage at the FET's source with the source pin of the FET tied to the supply of the target board. Using this arrangement allows the system to provide a wide range of ramp rates for the FET driver.

The HVOUT FET driver outputs a configurable source current (I_{SOURCE}) in order to charge the FET gate. When the driver is turned off, it outputs a configurable sink current (I_{SINK}) to discharge the FET gate. The Isink setting also includes a fast turn off setting. See the High Voltage Outputs section in DC and Switching Characteristics for more details.

The four HVOUT pins can also be configured as switched mode outputs in either the high-voltage FET driver or Open Drain mode. This is useful when the HVOUT is driving a High side MOSFET controlling a supply greater than 6 V. This feature is also useful for driving a MOSFET in a charge pump circuit to generate voltages above 12 volts. The switched output duty cycle can be configured from 6.25% up to 93.75% in step sizes of 6.25% and the frequency can be configured as either 15.625 kHz or 31.25 kHz. This flexibility allows the output to be configured to drive a wide variety of circuit components for a design. The rise and fall of the switched mode outputs may not complete with certain combinations of the charge pump settings (V_{PP}, I_{SOURCE}, I_{SINK}) and the switched mode settings (Duty Cycle and Frequency). The configuration should be chosen with the output circuit in mind.

Safe State of Digital Outputs

During power-up the GPIO will be configured as outputs and will be in the “Safe-State” as defined in Table 11. The HVOUTs will be configured in the FET driver mode during power-up. When the ASC completes the power up sequence then the HVOUT and GPIO control is transferred to the ASC-I/F or OCB depending upon the configuration. The ASC will indicate that it has completed the power up sequence by asserting the AGOOD signal to the FPGA using the ASC-I/F.

Table 11. GPIO and HVOUT Safe-State definitions

I/O	SAFE-STATE
HVOUT1	Low
HVOUT2	Low
HVOUT3	Low
HVOUT4	Low
GPIO1	Low
GPIO2	Low
GPIO3	Low
GPIO4	Low
GPIO5	Low
GPIO6	Low
GPIO8	Hi-Z
GPIO9	Hi-Z
GPIO10	Low

Controlling Power Supply Output Voltage by Trim and Margin Block

One of the key features of the ASC is its ability to make adjustments to the power supplies that it may also be monitoring and/or sequencing. This is accomplished through the Trim and Margin Block of the device.

As shown in Figure 24 the Trim and Margin Block can adjust voltages of up to four different power supplies through the DACs built-in the Trim Cells. The DC-DC blocks in the figure represent virtually any type of DC power supply that has a trim or feedback input. This can be an off-the-shelf unit or custom circuit designed around a switching regulator IC. The interface between ASC and the power supply shown in diagram by a resistor actually represents a resistor network.

The individual ASC-I/F control signals for each Trimcell are:

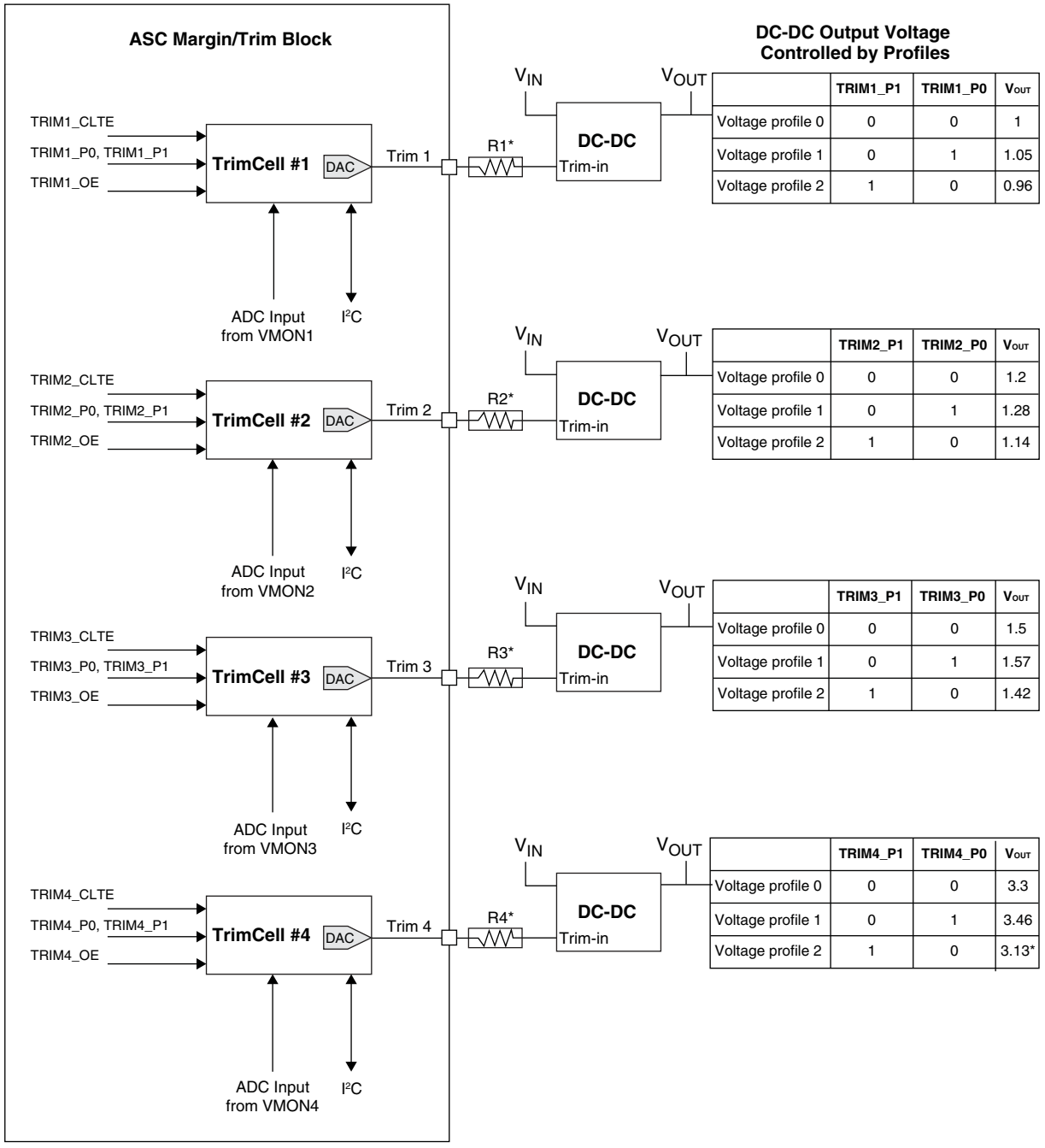
- ASCx_TRIMx_CLTE — This is a closed loop trim enable signal of a TrimCell. When ASCx_TRIMx_CLTE =1 the closed loop trimming for the DC-DC power supply connected to the TrimCell is enabled.
- ASCx_TRIMx_P0 and ASCx_TRIMx_P1 — These are two closed loop Trim Profile select signals used to select the active voltage profile of a TRIM cell.
- ASCx_TRIMx_OE — This control signal enables the DAC output of a TrimCell. When ASCx_TRIMx_OE=1 the DAC output of the Trim cell is active.

Other inputs to the TrimCell are:

- ADC — This input to the Trim cell is from the ADC which converts each VMON voltage into digital. The ADC input is used by the Trim Cell for controlling the closed loop trim operation.
- I^2C interface — Internal registers of the TrimCell can be accessed via I^2C interface. The Platform Designer software provides control signals which can be programmed to restrict I^2C access to the ASC.

Next to each DC-DC converter, three example voltages are shown. These example voltages correspond to the operating voltage profile of the corresponding TrimCell. As shown in Figure 24, the active operating profile for each TrimCell is selected independently (of other TrimCells) using TRIMx_P0 and TRIMx_P1 signals.

Figure 24. ASC Margin/Trim Block



DC-DC Output Voltage Controlled by Profiles

	TRIM1_P1	TRIM1_P0	V_{OUT}
Voltage profile 0	0	0	1
Voltage profile 1	0	1	1.05
Voltage profile 2	1	0	0.96

	TRIM2_P1	TRIM2_P0	V_{OUT}
Voltage profile 0	0	0	1.2
Voltage profile 1	0	1	1.28
Voltage profile 2	1	0	1.14

	TRIM3_P1	TRIM3_P0	V_{OUT}
Voltage profile 0	0	0	1.5
Voltage profile 1	0	1	1.57
Voltage profile 2	1	0	1.42

	TRIM4_P1	TRIM4_P0	V_{OUT}
Voltage profile 0	0	0	3.3
Voltage profile 1	0	1	3.46
Voltage profile 2	1	0	3.13*

*Indicates resistor network, see Figure 25.

There are four independently enabled TrimCells in the ASC device for controlling up-to four individual power supplies. Each Trimcell can generate up-to three trimming voltages to control the output voltage of the DC-DC converter.

Figure 25. TrimCell Driving a Typical DC-DC Converter

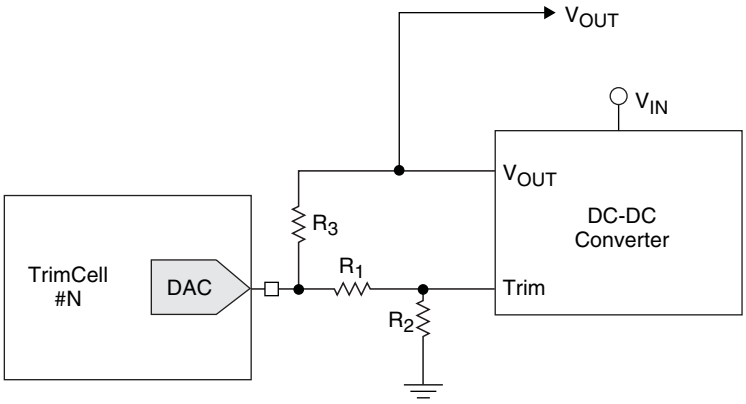


Figure 25 shows an example resistor network between the TrimCell #N in the ASC and the DC-DC converter. The values of these resistors depend on the type of DC-DC converter used and its operating voltage range. The calculation to determine the values of the resistors R1, R2, and R3 is performed automatically in the Platform Designer software.

TrimCell Architecture

The TrimCell block diagram is shown in Figure 26. Each TrimCell can be used in either of two modes to control an 8-bit DAC. The output of the DAC can be used to apply a voltage to trim an external power supply or DC-DC converter. The **Trim Configuration Mode** is selected in the Platform Designer software; the default mode is **Trim Calculator**, and the optional mode is **Manual**. Manual mode applies the user specified DAC Codes directly to the DAC Register. The Trim Calculator mode is the Closed Loop Trim Mode where feedback from the corresponding VMON is compared to a programmable Voltage Setpoint Register and the result is used to update the DAC register.

Manual Mode

Shown in the upper portion of Figure 26, PROFILE 0, PROFILE 1, PROFILE 2 are 8-bit DAC Codes that are written in the EEPROM memory during programming. The active DAC Code for each TrimCell is independently chosen based on ASC-I/F signals TRIMx_P0 / TRIMx_P1. The active DAC Code is written to the DAC Register whenever the TRIMx_P0, TRIMx_P1 signals change. As shown, the PROFILE 0 DAC Code written in configuration memory can be overwritten by I^2C commands during run-time. The I^2C access to the PROFILE 0 DAC Code can be restricted based on the ASC I^2C write protect feature that can be enabled during configuration (see the I^2C Interface section for more details).

Closed Loop Trim Mode

Shown in the lower portion of Figure 26, PROFILE 0, PROFILE 1 and PROFILE 2 are 12-bit Setpoints, which are written in the EEPROM memory during programming. The active Setpoint for each TrimCell is independently chosen based on ASC-I/F signals TRIMx_P0 / TRIMx_P1. This Setpoint is copied to the Voltage Setpoint Register whenever the TRIMx_P0, TRIMx_P1 signals change. As shown, the PROFILE 0 Setpoint written in configuration memory can be overwritten by I^2C commands during run-time. The I^2C access to the PROFILE 0 Setpoint can be restricted based on the ASC I^2C write protect feature that can be enabled during configuration (see the I^2C Interface section for more details). The Digital Closed Loop Trim Logic (near the center of Figure 26) compares the Voltage Setpoint Register with the corresponding VMON voltage (digitized by the ADC) to make active adjustments to the 8-bit DAC Register. The Closed Loop Trim is enabled on a per channel basis, depending on the ASC-I/F signal TRIMx_CLTE. See the Digital Closed Loop Trim Mode section for additional details on this mode of operation.

DAC Output Control

The DAC output of the TrimCell is enabled using ASC-I/F signal TRIMx_OE. When enabled, the DAC register value is converted to an analog voltage and output on the TRIMx pin. When disabled, the DAC output is high impedance. The Trim Configuration Mode sets how the DAC Register is controlled (either Manual or Closed Loop Trim) and is programmed into the EEPROM memory. The Trim Configuration Mode is set by the user in the *Trim/Margin* view of Platform Designer software based on the selection of either **Manual** or **Trim Calculator**. The DAC output values versus configuration settings are shown in Table 12.

Figure 26. TrimCell Architecture

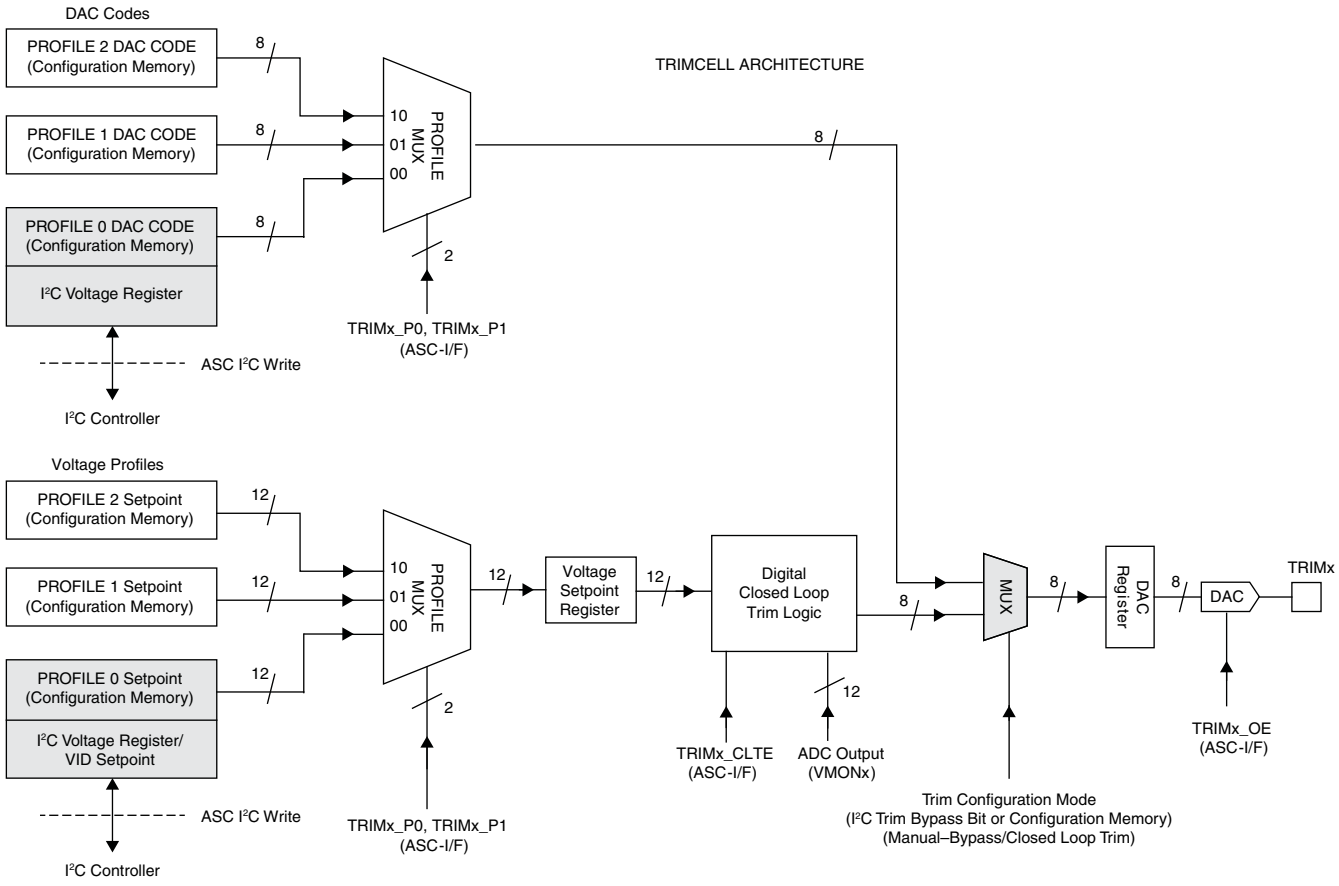


Table 12. DAC Output Value vs. Configuration Settings

Trim Configuration Mode (Platform Designer Software)	TRIMx_CLTE (ASC –I/F)	TRIMx_OE (ASC- I/F)	DAC Output Value
x	x	0	Hi-Z
Manual (Bypass)	x	1	DAC Code
Trim Calculator (Closed Loop Trim Mode)	0	1	Held at last updated value by Closed Loop Trim Logic. Reset value is 80h.
Trim Calculator (Closed Loop Trim Mode)	1	1	Dynamically updated based on measured VMONx voltage and Digital Closed Loop Trim Logic.

VID Selection

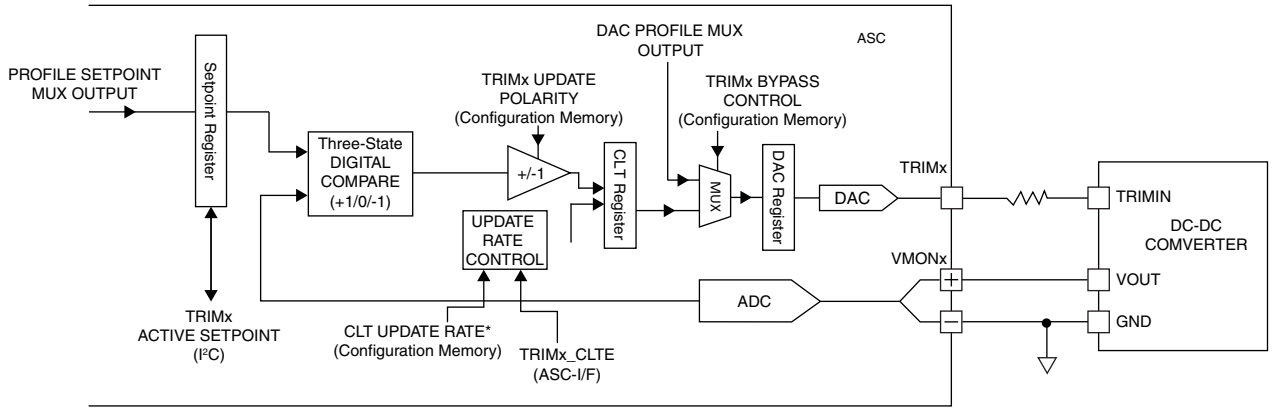
The ASC can be configured to support VID (Voltage Identification) control using the TRIM block. The control signals and VID tables are created using the Platform Designer software. As shown in Figure 26, the VID mechanism uses the I^2C interface to control the VID Setpoint (duplicated as PROFILE 0 Setpoint). The I^2C access to the VID Setpoint can be restricted based on the ASC I^2C write protect feature that can be set during configuration.

Digital Closed Loop Trim Mode

Closed loop trim mode operation can be used when tight control over the DC-DC converter output voltage at a desired value is required. The closed loop trim mechanism operates by comparing the measured output voltage of the DC-DC converter with the internally stored Voltage Setpoint. The difference between the Voltage Setpoint and the actual DC-DC converter voltage generates an error voltage. This error voltage adjusts the DC-DC converter output voltage toward the Voltage Setpoint. This operation iterates until the Voltage Setpoint and the DC-DC converter voltage are equal. The closed loop trim hardware then continues monitoring the converter voltage and adjusts the converter output voltage as necessary. Figure 27 shows the closed loop trim operation of a TrimCell. At regular intervals (as determined by the Update Rate Control register) the ASC device initiates the closed loop power supply voltage correction cycle through the following blocks

- Volatile Voltage Setpoint Register stores the desired output voltage (set by the TRIMx_P0 and TRIMx_P1 ASC-I/F signals)
- On-chip ADC is used to measure the voltage of the DC-DC converter
- Three-state comparator is used to compare the measured voltage from the ADC with the Voltage Setpoint Register contents. The output of the three state comparator can be one of the following:
  - +1 if the setpoint voltage is greater than the DC-DC converter voltage
  - –1 if the setpoint voltage is less than the DC-DC converter voltage
  - 0 if the setpoint voltage is equal to the DC-DC converter voltage
- Channel polarity control determines the polarity of the error signal (Polarity is set on a per channel basis in configuration memory)
- Closed loop trim register is used to compute and store the DAC code corresponding to the error voltage. The contents of the Closed Loop Trim will be incremented or decremented depending on the channel polarity and the three-state comparator output. If the three-state comparator output is 0, the closed loop trim register contents are left unchanged.
- The DAC in the TrimCell is used to generate the analog error voltage that adjusts the attached DC-DC converter output voltage.

Figure 27. Digital Closed Loop Trim Operation



* CLT UPDATE RATE parameter is shared between all four TRIM Cells

The closed loop trim cycle interval is programmable and is set by the update rate control register. Table 13 lists the programmable update interval that can be selected by the update rate register. The update rate register is set in configuration memory and is shared between all TRIM cells.

Table 13. Closed Loop Trim Update Rates

CLT Update Rate Settings
860 µs
1.72 ms
13.8 ms
27.6 ms

There is a one-to-one relationship between the selected TrimCell and the corresponding VMON input for the closed loop operation. For example, if TrimCell 3 is used to control the power supply in the closed loop trim mode, VMON3 must be used to monitor its output power supply voltage. The closed loop operation can only be started by asserting the TRIMx_CLTE ASC-I/F signal.

TrimCell at Start-up
The status of registers and the TrimCell output during start-up or POR of the ASC is as follows.

1. The TRIM DAC output is High-Z.

2. DAC register is based on Trim configuration Mode.

Trim Configuration Mode for TRIMx channel (Platform Designer Software)	TRIMx DAC register
Manual	Profile 0 DAC code is copied to the DAC register.
Trim Calculator	Value of 80h (Bipolar-zero) is copied to the DAC register.

3. The Closed Loop Trim Logic is disabled.

4. Profile 0 Setpoint is copied to the Voltage Setpoint Register.

The DAC output mode can be enabled (TRIMx_OE) at any time by the user logic, depending on the application requirements. Normally the chosen profile (TRIMx_P0, TRIMx_P1) setpoint should be loaded and the DAC output enabled when the application is ready for trimming. If closed loop trimming is to be used, the user logic should enable the closed loop trim (TRIMx_CLTE) after the DAC output and trim profile have already been configured.

Details of the Digital to Analog Converter (DAC)

Each trim cell has an 8-bit bipolar DAC to set the trimming voltage as shown in Figure 28. The full-scale output voltage of the DAC is +/– 320 mV. A code of 80H results in the DAC output set at its bi-polar zero value.

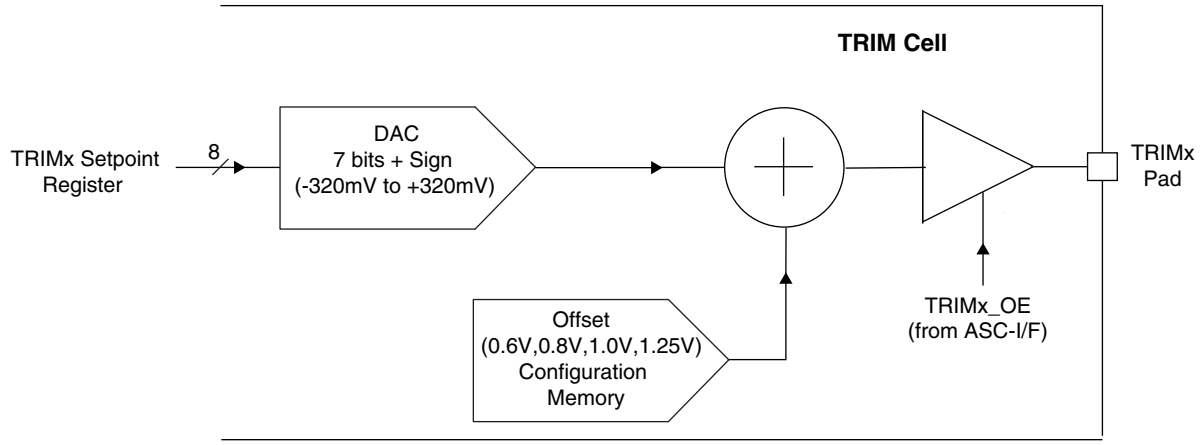
The voltage output from the DAC is added to a programmable offset value and the resultant voltage is then applied to the trim output buffer. The offset voltage is typically selected to be approximately equal to the DC-DC converter open circuit trim node voltage. This results in maximizing the DC-DC converter output voltage range.

The programmed offset value can be set to 0.6 V, 0.8 V, 1.0 V or 1.25 V. This value selection is stored in configuration memory. The configuration memory is loaded with the value set in EEPROM memory at power-on. It can be updated during runtime via I^2C commands.

The combined offset and DAC output is applied to the TRIM cell output buffer. Each output buffer is controlled by a unique TRIMx_OE signal via the ASC-I/F. When TRIMx_OE = 0, the corresponding TRIMx Pad will be placed in a high impedance state. Setting TRIMx_OE = 1 will enable the output buffer, resulting in the combined offset and DAC output being applied to the TRIM output pin.

The default state at power-on reset is TRIMx_OE = 0. The TRIM cell will maintain this setting until the ASC-I/F communication is successfully established. This ensures that the TRIM function will remain in a passive, high impedance state, until it is enabled the user control logic.

Figure 28. Offset Voltage is Added to DAC Output Voltage to Derive Trim Pad Voltage



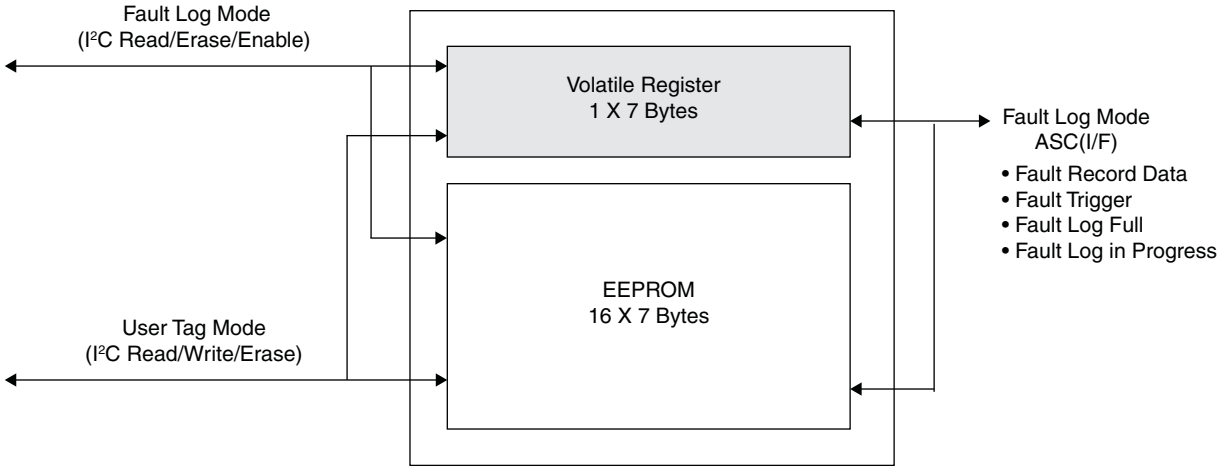
Fault Logging and User Tag Memory

The ASC contains the following storage space used with Fault Logging or User Tag operation:

- Non-volatile EEPROM memory array which has 16 rows where each row stores 7 bytes of data.
- A volatile memory register which stores 7 bytes of data.

The ASC can be configured to choose this memory, either for User Tag Operation or Fault Log Operation through Platform Designer Software. Figure 29 shows the interface to the EEPROM memory and Volatile register for data access.

Figure 29. Access to EEPROM and Volatile Memory for Fault Logging/User Tag Operation



User Tag Memory

When the ASC is configured for User Tag Mode, the memory block can be used as a scratch pad memory for critical data, board serialization, board revision logs, programmed pattern identification or as general data storage in EEPROM.

As shown in Figure 29, in the User Tag Mode, data can be read, written or erased from the EEPROM or Volatile Register via the I²C interface of the ASC. For more details, please refer to User Tag Memory Access in the I²C Interface section of this data sheet.

Fault Log Memory

When the ASC is configured for Fault Log Mode, the memory block is used to record the status of the ASC GPIOs, VMON, IMON, TMON and other significant logic signals on the occurrence of the user defined fault trigger condition. The ASC can also be used with Platform Manager 2, MachXO2, MachXO3, or ECP5 devices to log faults to User Flash Memory (UFM) or external SPI flash. See TN1277, Fault Logging Using Platform Manager 2 for more details.

Each fault record has seven bytes, six bytes of ASC specific data and one byte of user specified FPGA signals. The ASC Fault Log Record Memory Map is shown in Table 14. Erased fault records and fault records which have not been written yet will read all zeros.

Table 14. Fault Log Record Memory Map

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	User bit7	User bit6	User bit5	User bit4	User bit3	User bit2	User bit1	User bit0
1	AGOOD	GPIO10	GPIO9	GPIO8	X	GPIO6	GPIO5	GPIO4
2	GPIO3	GPIO2	GPIO1	HVOUT4	HVOUT3	HVOUT2	HVOUT1	IMON1B
3	IMON1A	HIMONB	HIMONA	HVMONB	HVMONA	VMON9B	VMON9A	VMON8B
4	VMON8A	VMON7B	VMON7A	VMON6B	VMON6A	VMON5B	VMON5A	VMON4B
5	VMON4A	VMON3B	VMON3A	VMON2B	VMON2A	VMON1B	VMON1A	TMON2B
6	TMON2A	TMON1B	TMON1A	TMONINB	TMONINA	1	0	1

The ASC can be configured to store fault log data either in the EEPROM array or the Volatile register.

The EEPROM memory array can store up to 16 fault log records. When the fault log memory is full no further fault log records can be stored in the EEPROM and any future trigger signals will be ignored.

The volatile register can also be used to store faults. The volatile fault log contains only one record of 7 bytes and each time the trigger signal is asserted the current data will be stored in the register overwriting any previous data. In order to preserve the volatile register fault log data it must be read back prior to the next assertion of the trigger signal.

The following control signals for ASC based Fault Logging are defined in the Platform Designer software for use in the FPGA logic:

- Fault_Log_Trigger: This user defined signal is used to initiate fault log recording. Recording is initiated by toggling the fault log trigger signal high based on the FPGA logic. The Fault log trigger signal should be set high for a minimum period (see Recommended Operating Conditions section). The fault log trigger signal initiates fault log recording for all ASCs in the system. Readback must be disabled for the fault log recording to begin.
- ASCx_Fault_Log_Full: This ASC-I/F signal reports to the user logic when the EEPROM for the given ASC is full.
- ASCx_Fault_Log_In_Progress: This ASC-I/F signal reports to the user logic when a fault log operation for the given ASC is in progress.

When the ASC is configured for Fault Log Operation, the Fault Record Data frame, as shown in Table 14, is captured every 16 us. When the fault log trigger signal is asserted, the captured data is stored in the selected memory. This includes the user bits in the fault record. These user bits are not used for any other ASC functions.

The read-back function of the Fault Log must be enabled in order to read or erase the Fault Log. The read-back is enabled using the I^2C interface. As shown in Figure 29, the Fault Log contents can be read or erased from the EEPROM or the volatile register via the I^2C interface of the ASC.

When the user enables the read-back of the fault log contents, the fault log recording is disabled and must be re-enabled by the user after the read-back is completed in order to store future fault log events. For more information about reading, erasing and enabling the fault log recording refer to Fault Log Memory Access in the I^2C Interface section of this data sheet.

System Connections

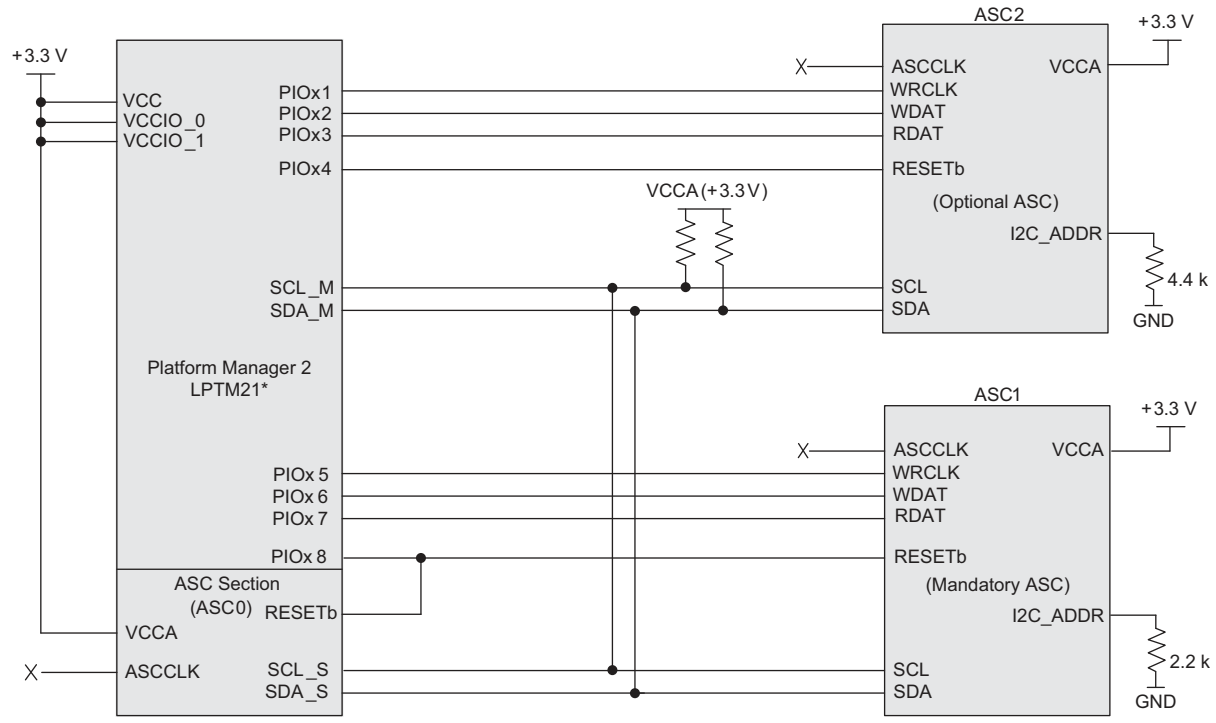
The ASC device is a hardware management expander, designed for use in systems which use either the Platform Manager 2, MachXO2, MachXO3, or ECP5 FPGA as the hardware management controller. Platform Manager system designs can be built using several combinations of Lattice devices as listed in Table 15. In order for the ASC to function properly as a hardware management expander, there are a number of mandatory connections to the hardware management controller. The overall set of required connections between the ASC and the hardware management controller are shown in Figure 30 thru Figure 32 below. The required connections include Clock, Reset, ASC Interface (ASC-I/F) and I^2C. These connections are assigned and managed using Diamond software and the Platform Designer tool. Each of the connection requirements is described below.

Table 15. Platform Manager 2 Design Options

Central Hardware Manager	Hardware Management Expander	Number of Expanders Supported[1, 2]
LPTM21	LPTM21L or L-ASC10	0 – 3
LPTM21L	LPTM21L or L-ASC10	0 – 3
MachXO2	LPTM21L or L-ASC10	1 – 8
MachXO3	LPTM21L or L-ASC10	1 – 8
ECP5	LPTM21L or L-ASC10	1 – 8

1. Platform Manager 2 designs with 6 hardware expanders are best supported with MachXO2 and MachXO3 devices of 2k LUTs or larger.
2. Platform Manager 2 designs with 8 hardware expanders are best supported with MachXO2 and MachXO3 devices of 4k LUTs or larger.

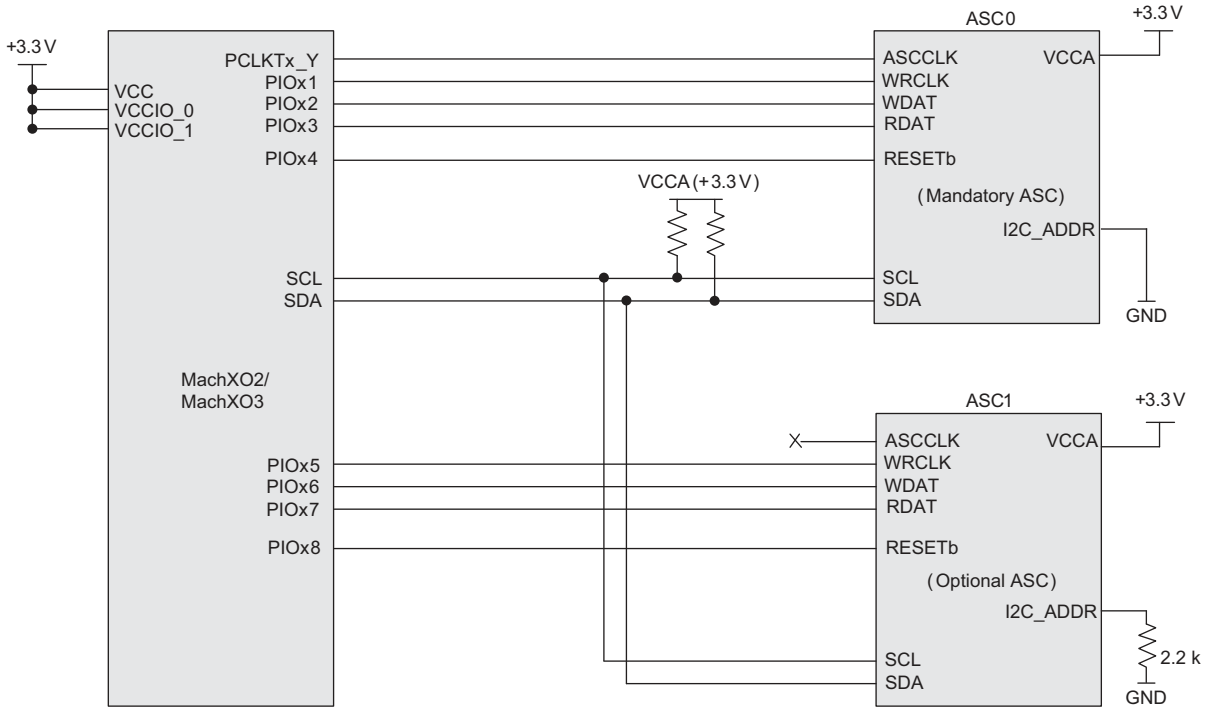
Figure 30. System Connections - ASC and Platform Manager 2



* LPTM21L (100-ball package) has I^2C master and slave connected internally

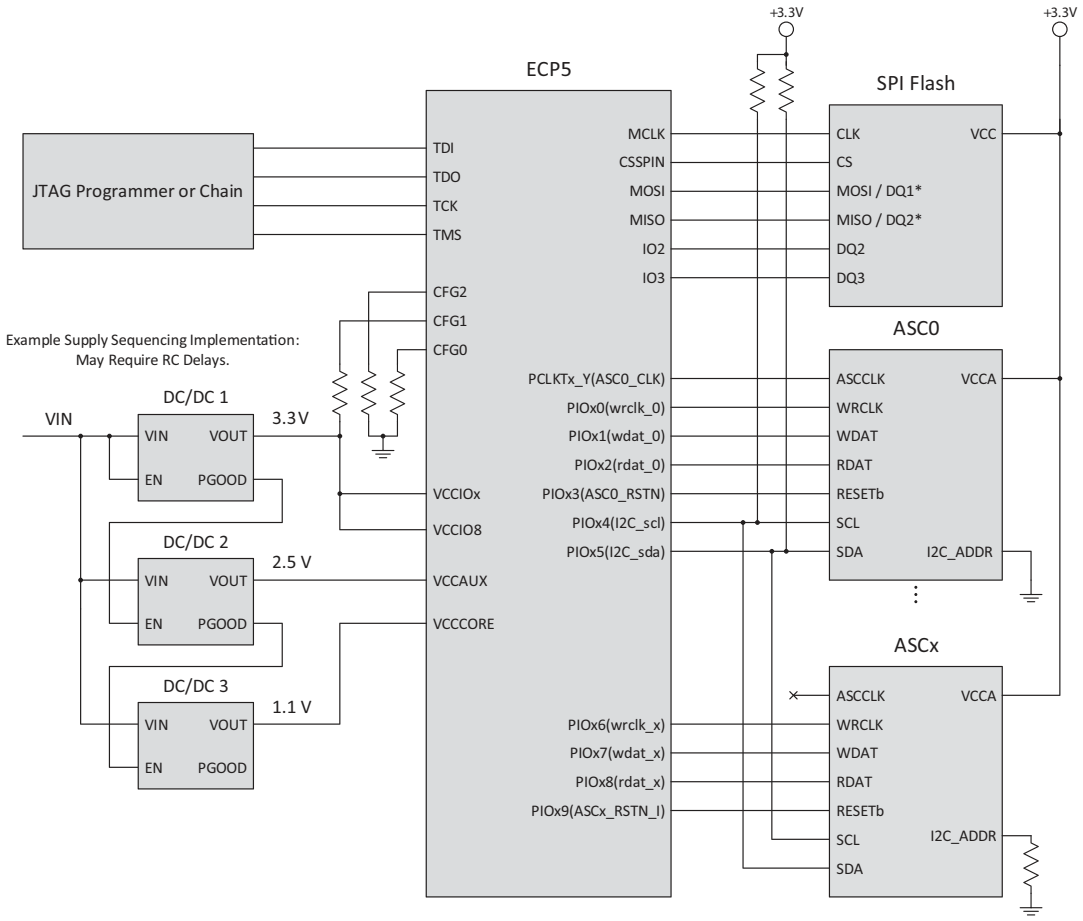
Note: Hardware connections may require additional passive components not shown, see TN1225, Platform Manager 2 Hardware Checklist for more details.

Figure 31. System Connections - ASC and MachXO2, or MachXO3



Note: Hardware connections may require additional passive components not shown, see TN1225, Platform Manager 2 Hardware Checklist for more details.

Figure 32. System Connections - ASC and ECP5



Note: Hardware connections may require additional passive components not shown, see TN1225, Platform Manager 2 Hardware Checklist for more details.

Clock Requirements

The ASC has an internal 8 MHz clock source which is used by the device during startup. Once startup has successfully completed, the ASC will switch to the ASC-I/F system clock signal (WRCLK) for operation. The hardware management controller provides the WRCLK signal for each ASC in the system. This ensures that the system is fully synchronized to a common clock source to minimize any differences in timing.

The ASC has a built-in detection circuit for WRCLK loss. If a loss of WRCLK is detected, the ASC will reset itself and pull RESETb low. The device I/O will return to safe state, as described in the Safe State of Digital Outputs section.

The ASC internal clock signal is made available on the ASCCLK pin of the ASC0 device for use as the system source clock. This signal is connected internally in the Platform Manager 2 device (see Figure 30), making the ASCCLK pin a no-connect in Platform Manager 2 systems. In systems using the MachXO2, MachXO3, or ECP5 and external ASC devices, the ASC0 ASCCLK will be enabled and must be connected to a MachXO2, MachXO3, or ECP5 primary clock input, as shown in Figure 31 and Figure 32. The hardware connection and MachXO2, MachXO3, or ECP5 pin assignment must be made by the user in the design software. All other ASC devices (both optional and mandatory) in the system will disable their ASCCLK output signal and this pin should be treated as a no connect.

An external 8 MHz clock source can be used as the system clock instead of the ASC0 ASCCLK. In this case, the ASCCLK output will be disabled, and the external clock should be connected to a primary clock pin on MachXO2, MachXO3, or ECP5 FPGA or to the ASCCLK pin on Platform Manager 2. The user must specify that an external clock source is being used in software.

Reset Requirements

The ASC RESETb pin is used for synchronizing the ASCs with the Platform Manager 2, MachXO2, MachXO3, or ECP5 FPGA device. The RESETb pin should not be driven by any external device as this will adversely affect the system operation. A software reset signal for the internal logic can be created using a PIO pin on the Platform Manager 2, MachXO2, MachXO3, or ECP5 device.

The external ASCs for a project can be designated as either Mandatory or Optional. The Mandatory or Optional designation determines how the RESETb pins must be connected and how the system will treat the Reset signal from each ASC. The Mandatory or Optional designation must be specified in the design software.

A Mandatory ASC is required to be present at system start-up. The RESETb pins for all mandatory ASCs must be connected to the RESETb pin on the Platform Manager 2 (as shown in Figure 30). The ASC0 device is always considered mandatory (this includes the internal ASC section of Platform Manager 2, which is always designated as ASC0). If any one of the mandatory ASCs cannot be detected by the hardware management controller, the system will be held in reset. Any of the mandatory ASCs which experience a critical issue (such as loss of WRCLK signal) will hold the Reset signal low, keeping the system in reset.

In MachXO2, MachXO3, ECP5, and Platform Manager 2 systems, the ASC0_RESETb and other mandatory ASC reset signals must be connected to a PIO externally, as shown in Figure 30 and Figure 31. This PIO should be assigned to the ASC0_RSTN signal in the design software.

An Optional ASC is not required to be present at system start-up. This designation can be used for ASCs placed on plug in modules or optional boards in a system. The RESETb pin of each optional ASC should be connected to a unique PIO pin on the Platform Manager 2, MachXO2, MachXO3, or ECP5 FPGA. Each reset signal is treated individually, so that only the registers associated with a particular Optional ASC will reset when the reset input is driven low. The rest of the system, both Mandatory and other Optional ASCs, will continue to operate normally without interruption.

ASC Interface and I²C Connections

The ASC uses two communication links to transfer information between the ASC and the Platform Manager 2, MachXO2, MachXO3, or ECP5 FPGA. These are the ASC Interface (ASC-I/F) and I²C bus. These two links are used for different types of information and both must be connected properly for the system to operate correctly.

The ASC-I/F bus uses three signals: WRCLK, WDAT, and RDAT. The ASC-I/F bus operates at 8 MHz and includes error checking and reporting capabilities. The ASC-I/F pins on external ASC devices must be connected to three PIO pins on the Platform Manager 2, MachXO2, MachXO3, or ECP5 device. These three PIO pins are assigned using the design software. The design software will automatically instantiate the interface for communicating with the ASC devices. Each ASC device requires its own unique ASC-I/F link, as shown in Figure 30 and Figure 31.

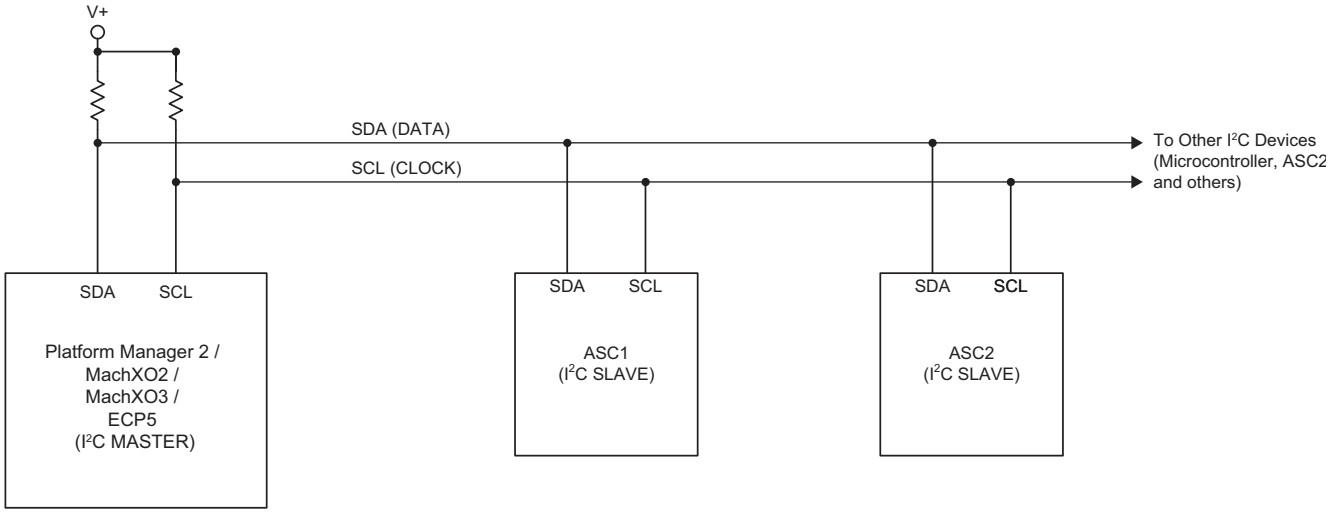
The VCCA pin for an external ASC0 device must be connected to the VCCIO of the I/O bank used for the PIO assignment of WRCLK, WDAT, RDAT. Care should be taken that the I/O bank used for the ASC-I/F link is not exposed to significant SSO noise, as this can degrade the performance of the analog monitors. See TN1225, Platform Manager 2 Hardware Checklist for more details.

The I²C bus uses the SDA and SCL pins and operates at 100 to 400 kHz. The user must connect the SDA and SCL pins on the ASC to the SDA_M/SCL_M pins on the Platform Manager 2 (Figure 30), or the SDA/SCL pins on the MachXO2 or MachXO3 (Figure 31). The ECP5 does not have dedicated I2C pins, so any I/O pins can be assigned to I2C_sda and I2C_scl, as long as the supply for the I/O bank is connected to the VCCA of the ASC (Figure 32). The Platform Manager 2 also requires connections between the SDA_M/SCL_M and the SDA_S/SCL_S pins. External pull-up resistors to VCCA are required in all configurations. See the I²C Interface section for full details.

I²C Interface

I²C is a low-speed serial interface protocol designed to enable communications among a number of devices on a circuit board. The ASC supports the I²C communications protocol 7-bit addressing. The I²C interface of the ASC is used for programming by the master FPGA or other system processor. The interface is also used for accessing measurement and control functions and fault log memory on the device. Figure 33 shows a typical I²C configuration, in which one or more ASC devices are slaved to a Platform Manager 2, MachXO2, MachXO3, or ECP5 FPGA. SDA is used to carry data signals, while SCL provides a synchronous clock signal. The 7-bit address of the ASC is determined by EEPROM programming and the resistor setting on the I2C_ADDR pin.

Figure 33. ASC Devices in an I²C System



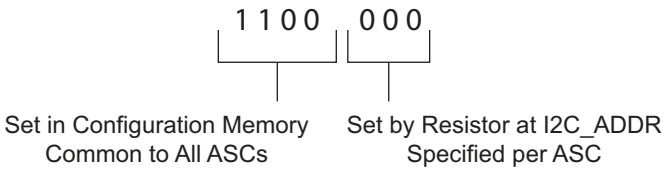
In the I²C protocol, the bus is controlled by a single MASTER device at any given time. This master device generates the SCL clock signal and coordinates all data transfers to and from a number of slave devices. The ASC is designed as an I²C slave. In a multiple ASC system configuration, all ASCs share the same I²C bus. This shared I²C bus is used by the Platform Manager 2, MachXO2, MachXO3, or ECP5 master to program the ASC devices. Each slave device is assigned a unique address. Any 7-bit address can be assigned to the ASC, however one should note that several addresses are reserved by the I²C standard and should not be assigned to the ASC to ensure bus compatibility. These are shown in Table 16.

Table 16. I²C Reserved Slave Device Addresses

Address	R/W bit	I²C Function Description
0000 000	0	General Call Address
0000 000	1	Start Byte
0000 001	X	CBUS Address
0000 010	X	Reserved
0000 011	X	Reserved
0000 1xx	X	HS-mode master code
1111 0xx	1	10-bit addressing
1111 1xx	X	Device ID

The 7-bit address of the ASC device is set based on both a configuration memory parameter and the pin state of I2C_ADDR (see Figure 33). The 4 MSB of the slave address are programmable and stored in configuration memory. The 4 MSB are common between all ASC devices used in a platform management configuration. The 4 MSB in all blank ASC devices are set to 1100.

Figure 34. I²C Slave Address Construction



The 3 LSB of the slave address are set by connecting the I2C_ADDR pin to ground via a given resistor value. The seven states of the 3 LSB have a one to one correspondence with the ASC number designation in the platform management configuration. Table 17 shows the relationship between the resistor values and the 3 LSB of the I²C Address / ASC device number. Resistors with 1% accuracy or better should be used to ensure proper address resolution. Two ASC device numbers do not require a resistor: ASC0 (I2C_ADDR connected directly to ground) and ASC7 (I2C_ADDR connected directly to V_{CCA}).

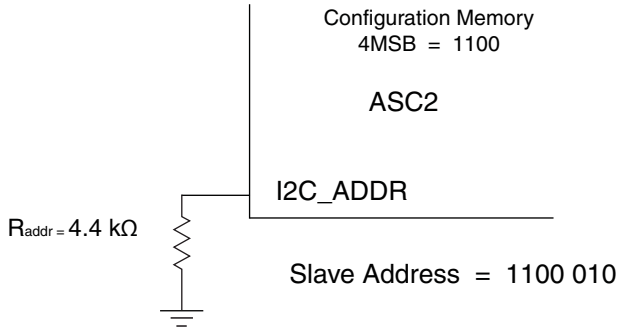
Table 17. R_{addr} Value vs. ASC Device Number

R_{addr} Value[1]	3 LSB of I²C Slave Address	ASC Device Number
None (Tie to GND)	000	0
2.2 kΩ	001	1
4.4 kΩ[2]	010	2
7 kΩ[3]	011	3
10 kΩ	100	4
13.7 kΩ	101	5
17.8 kΩ	110	6
None (Tie to V_{CCA})	111	7

1. All resistor values should be +/- 1% tolerance or better.
2. For designs that utilize E-96 resistors a value of 4.42 kΩ can also be used.
3. For designs that utilize E-96 resistors a value of 7.15 kΩ can also be used.

Figure 35 shows an example configuration of the I²C Slave Address. In this example, the ASC Device is ASC2. The I2C_ADDR pin is tied to ground via a 4.4 kΩ resistor as specified by the value for ASC2 in Table 17. The configuration memory in this example is programmed with the common 4 MSB for all ASC devices in the system, 1100. The constructed I²C slave address is shown at the bottom of the diagram: 1100 010 (0x62).

Figure 35. I²C Address Resolution Example



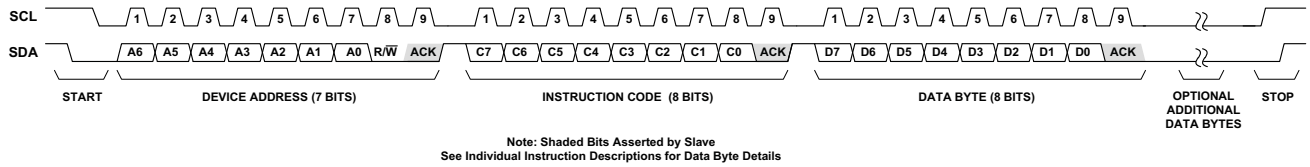
The ASC supports a dedicated 8-bit instruction set. These instructions are divided as follows among device programming instructions, measurement and control access, and fault log/user tag memory access. The ASC also supports configuration memory protection.

The ASC's I^2C interface allows data to be both written to and read from the device. A data write transaction, as shown in Figure 36, consists of the following operations:

1. Start the bus transaction
2. Transmit the slave address (7 bits) along with a low write bit
3. Transmit the instruction code as described in Table 18 (8 bits)
4. Transmit the first data byte to be written (8 bits). Note some instructions do not include data bytes, while others support multiple data bytes. For information on which instructions support multiple data bytes, see individual instruction details
5. Stop the bus transaction

To start the transaction, the master device holds the SCL line high while pulling SDA low. Address, instruction code and data bits are then transferred on each successive SCL pulse, in consecutive byte frames of 9 SCL pulses. Data is transferred on the first 8 SCL clocks in each frame, while an acknowledge signal is asserted by the slave device on the 9th clock in each frame. The first frame contains the 7-bit slave address, with bit 8 held low to indicate a write operation. The second frame contains the instruction code indicating the type of data to be written. The remaining frames contain the actual data to be written. The number of allowed or required data frames is determined by the instruction code used and is described in the Instruction Codes section.

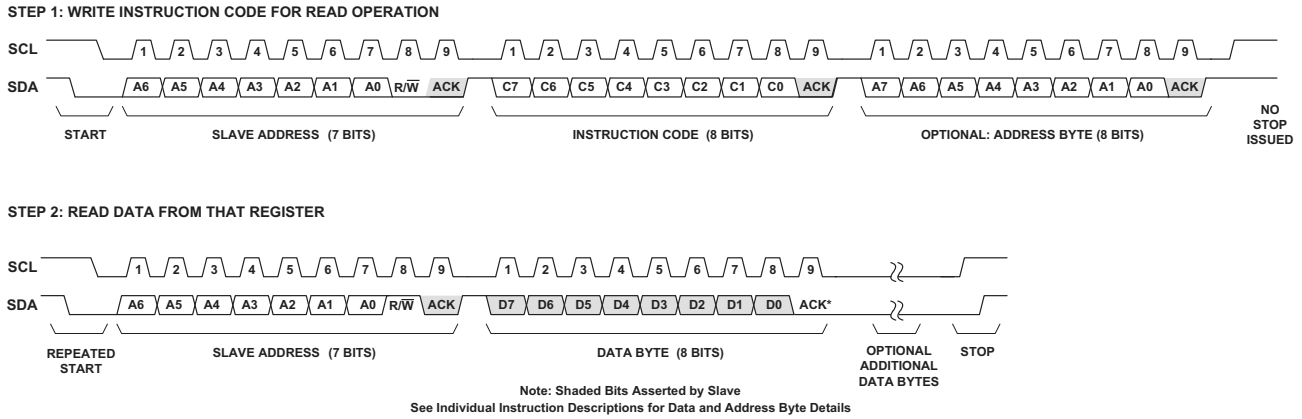
Figure 36. I^2C Write Operation



Reading a data byte from the ASC requires two separate bus transactions, as shown in Figure 37. The first transaction writes the device address with write bit, and then the instruction code indicating the type of data to be read. This transaction typically ends after the second frame since no data is being written to the slave. However, some instruction codes include additional frames, such as address information for the type of data to be read. See the Instruction Codes section for more information about the number of allowed or required data frames. No stop condition is issued at the end of the first step, to ensure that the full read operation is completed properly.

The second transaction performs the actual read, beginning with the issuing of a repeated start condition. A repeated start is a start condition issued by the master which does not follow a stop condition. This prevents the bus from being released by the master. The first frame contains the 7-bit slave address with the R/W bit held high. In the second frame, the ASC asserts data out on the bus in response to the SCL signal. Note that the acknowledge signal in the second frame is asserted by the master device and not the ASC. Depending on the instruction code, the ASC may assert additional data bytes in response to additional SCL frames depending on the instruction as detailed in the Instruction Codes section. The master completes the transaction by issuing a stop condition.

Figure 37. I²C Read Operation



* After final data byte read, master should NACK before issuing the STOP command

Instruction Codes

The ASC device supports a set of 8-bit instruction codes. These instructions are used to access EEPROM programming functions, shadow register programming functions, measurement and control functions, and User Tag or Fault Log memories. The instruction space is shown in Table 18. Each set of instructions is described in more detail in the following sections. Do not read or write to instruction codes marked reserved.

Table 18. I²C Instruction Summary

Instruction Code	Instruction Name	Instruction Group
0x01	RESERVED	N/A
0x02	READ_ID	Device Status and Mode Management
0x03	READ_STATUS	
0x04	ENABLE_PROG	
0x05	ENABLE_USER	
0x06-0x24	RESERVED	N/A
0x25	READ_CFG_EEPROM	ASC Configuration Memory Access
0x26-0x30	RESERVED	N/A
0x31	WRITE_CFG_REG	ASC Configuration Memory Access
0x32	WRITE_CFG_REG_wMASK	
0x33	READ_CFG_REG	
0x34	READ_ALL_CFG_REG	
0x35	LOAD_CFG_REG	
0x36-0x40	RESERVED	N/A
0x41	TRIM1_CLT_P0_SET	Closed Loop Trim Setpoint Access
0x42	TRIM2_CLT_P0_SET	
0x43	TRIM3_CLT_P0_SET	
0x44	TRIM4_CLT_P0_SET	
0x45-0x50	RESERVED	N/A
0x51	WRITE_MEAS_CTRL	Measurement and Control Register Access
0x52	READ_MEAS_CTRL	
0x53-0x60	RESERVED	N/A
0x61	ERASE_USER_TAG_EEPROM	User Tag Memory Access
0x62	WRITE_USER_TAG_REG	
0x63	READ_USER_TAG_REG	
0x64	PROG_USER_TAG_EEPROM	
0x65	READ_USER_TAG_EEPROM	
0x66-0x70	RESERVED	N/A
0x71	ERASE_FAULT_EEPROM	Fault Log Memory Access
0x72	RESERVED	N/A
0x73	READ_FAULT_VOLATILE_REG	Fault Log Memory Access
0x74	READ_FAULT_ENABLE	
0x75	READ_FAULT_RECORD_EEPROM	
0x76	READ_ALL_FAULT_EEPROM	
0x77-0xFF	RESERVED	N/A

Each instruction is described in detail in the following sections. The description includes information about the individual instruction code, the instruction format and any associated write or read addresses or data. The instruction format uses the following notation:

I²C Instruction Format Key (See Figure 37 for details of each condition or bit):

- S – Start Condition
- A[6:0] – Slave Address
- W – Write Bit (Logic 0)
- A – Acknowledge Bit
- NA – Not Acknowledge Bit
- Sr – Repeated Start Condition
- R – Read Bit (Logic 1)
- P – Stop Bit
- Shaded Bits (A) – Bits asserted by the slave

Device Status and Mode Management

There are several miscellaneous registers from the programming flow which are useful or required for completing separate operations (such as entering the programming mode to enable the User Tag memory access).

Table 19. Device Status and Mode Management Instruction Codes

Instruction Code	Instruction Name	Read/Write	Description
0x01	RESERVED		N/A
0x02	READ_ID	R	Read the device ID Code
0x03	READ_STATUS	R	Read the ASC Status Register
0x04	ENABLE_PROG	W	Enable the programming mode (correct two byte key required)
0x05	ENABLE_USER	W	Enable the device user mode

The READ_ID instruction is used to verify that the slave device is an ASC or the ASC section of Platform Manager 2. The device IDCODES are shown in Table 20. The format for the READ_ID instruction is shown in Figure 38.

Figure 38. READ_ID Instruction Format

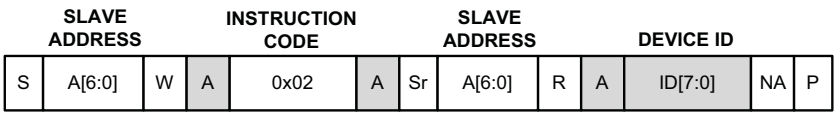


Table 20. ASC ID Codes

Device	ID Code
ASC Hardware Management Expander	0x88
ASC Section of LPTM21	0x8A
ASC Section of LPTM21L	0x8A

The READ_STATUS instruction provides readout access to the two byte status register of the ASC. The READ_STATUS instruction provides information about the status of the ASC fault log memory, the current chip mode (Programming Mode or User Mode), and the status of the DONE bit of the I²C address resolution and the configuration memory. The READ_STATUS instruction format is shown in Figure 39. The ASC_Status_Register bit mapping is shown in Figure 40.

Figure 39. READ_STATUS - I2C Instruction Format

SLAVE ADDRESS				INSTRUCTION CODE			SLAVE ADDRESS			ASC_STATUS REGISTER_LO		ASC_STATUS REGISTER_HI		
S	A[6:0]	W	A	0x03	A	Sr	A[6:0]	R	A	R[7:0]	A	R[15:8]	NA	P

Figure 40. ASC_Status Register

ASC_STATUS_REGISTER_LO (Read)

DUALBOOT_CRC _ERROR	CFGARRAY_ DONE	I2CSA_DONE	PROG_MODE	RESERVED	CFG_SHADOW_ REG_REFRESH	RESERVED	ERASE
b7	b6	b5	b4	b3	b2	b1	b0

ASC_STATUS_REGISTER_HI (Read)

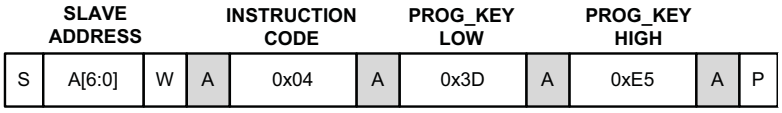
PROGRAM	FAULT_UT_ ERASE	FAULT_PROG	FAULT_LOG_ FULL	FAULT_CNT[3]	FAULT_CNT[2]	FAULT_CNT[1]	FAULT_CNT[0]
b15	b14	b13	b12	b11	b10	b9	b8

The individual status bits are described below:

- DUALBOOT_CRC_ERROR – Reset to logic 0 at power up and at the beginning of a dual-boot configuration write I²C instruction. Logic 1 when a CRC error is encountered during dual-boot configuration.
- CFGARRAY_DONE – Logic 1 if the configuration memory done bit has been programmed (set to 1 at the proper completion of an EEPROM programming operation)
- I2CSA_DONE – Logic 1 if the chip I²C slave address I2CSADone has been programmed (set to 1 at the proper completion of an EEPROM programming operation)
- PROG_MODE – Logic 1 if the chip is in programming mode, Logic 0 if the chip is in user mode
- RESERVED
- CFG_SHADOW_REG_REFRESH –Set to Logic 1 if Configuration EEPROM data was copied into corresponding shadow registers just after a Reset or after the shadow register refresh I²C instruction is given. This bit is cleared just after the status register is read out.
- ERASE – Logic 1 if any EEPROM Erase operation is in progress
- PROGRAM – Logic 1 if any EEPROM Program operation is in progress
- FAULT_UT_ERASE – Logic 1 if the ASC Fault Log or User Tag memory is currently being erased
- FAULT_PROG – Logic 1 if the ASC Fault Log data is being programmed into Fault Log EEPROM array
- FAULT_LOG_FULL – Logic 1 if all rows of the ASC Fault Log EEPROM have been programmed
- FAULT_CNT [3:0] – 4-bit value that is equal to the last row of ASC Fault Log EEPROM that has been programmed with fault log data. Row 0 up to Row FAULT_CNT have been programmed with Fault Log Data.

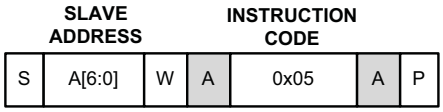
The ENABLE_PROG instruction places the ASC into the programming mode. The instruction requires that a specific key code is written along with it in order to ensure that the programming mode is not entered unintentionally. The ENABLE_PROG instruction should only be used by the Lattice delivered programming algorithms or to write or erase the User Tag memory. The ASC_PROG_KEY is a two byte value of 0xE53D. The ENABLE_PROG instruction format is shown in Figure 41.

Figure 41. ENABLE_PROG - I2C Instruction Format



After completing a user tag operation, it is important to exit the programming mode and return to user mode. This will prevent unintentional programming operations. The ENABLE_USER instruction will return the ASC to the user mode. The ENABLE_USER instruction format is shown in Figure 42.

Figure 42. ENABLE_USER - I2C Instruction Format



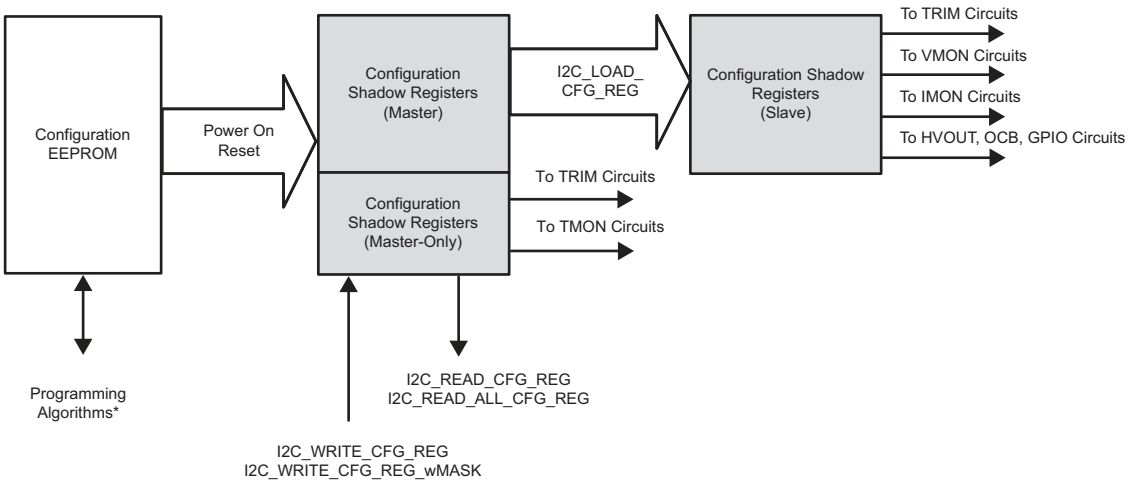
ASC Configuration Memory Access

The I^2C interface is used for programming the ASC device. The ASC device includes an EEPROM configuration memory which stores the device configuration in non-volatile memory. The ASC device also includes a set of shadow registers, which are used during runtime by the device to determine operational thresholds, output controls, etc. At power-on reset, the device automatically copies the EEPROM configuration memory to the shadow registers, provided the EEPROM done bit is set in the ASC Status Register. The EEPROM configuration settings are automatically generated by the Platform Designer software tool.

The I^2C interface unit provides access to both the non-volatile EEPROM memory and the configuration shadow registers for erase, programming, and verify operations. The EEPROM memory is background programmed. It can be copied to the configuration shadow registers at the end of programming by an additional I^2C instruction. The EEPROM configuration memory map is automatically generated by the Platform Designer software. The flow and usage of the EEPROM instructions is handled by the Lattice Diamond Programmer software (for PC-based programming) or the Lattice deployment tool (for programming the device via a tester or an on-board microcontroller using the I^2C embedded solution). Lattice recommends using these software tools to access the EEPROM configuration programming instruction space.

The I^2C interface can also be used to re-configure the shadow registers directly. These instructions provide access to individual voltage monitor thresholds, temperature measurement settings, and other device configuration parameters. Some configuration shadow registers are implemented as master/slave pairs. These shadow registers do not update operational parameters immediately after I^2C configuration writes to the master shadow register. They support an additional load instruction which updates all slave shadow registers from the master shadow registers at the same time. Other shadow registers are implemented as a single master-only register. These registers update their operation (or reset the associated circuit) immediately after an I^2C configuration write. The configuration memory architecture is shown in Figure 43. The ASC Configuration Registers section details which registers support the additional load instruction. The configuration registers can be accessed in user mode, although overwriting the registers can be protected through additional device settings. The configuration register access instructions are shown in Table 21.

Figure 43. Configuration Memory Architecture



* - EEPROM access algorithms generated by Lattice Design Software

Table 21. Configuration Register Instruction Codes

Instruction Code	Instruction Name	Read/Write	Description
0x25	READ_CFG_EEPROM	R	Read out the selected configuration EEPROM byte or bytes
0x31	WRITE_CFG_REG	W	Write configuration data byte to addressed register
0x32	WRITE_CFG_REG_wMASK	W	Write masked configuration data bits to addressed register
0x33	READ_CFG_REG	R	Read addressed configuration register
0x34	READ_ALL_CFG_REG	R	Read all configuration registers, starting at address 0x00
0x35	LOAD_CFG_REG	W	Load the slave shadow configuration registers from the I^2C master shadow configuration registers (not all registers supported, see Table 22)

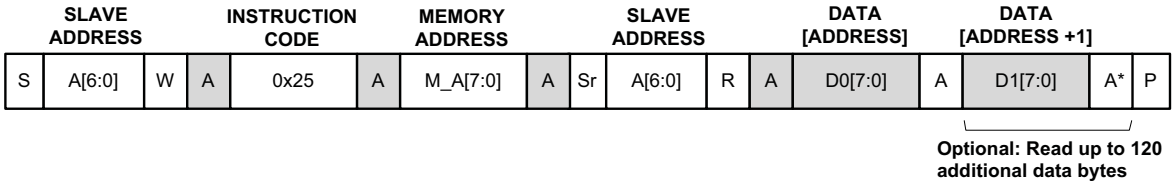
The configuration registers and address map are shown in the tables in the ASC Configuration Registers section. The tables in this section also describe which registers support the LOAD_CFG_REG instruction.

Special configuration memory parameters (such as the Write Protect setting, User Tag / Fault Log mode, and UES bits) can only be modified in EEPROM. They cannot be modified using configuration register instructions. This increases the reliability of the device operation.

The READ_CFG_EEPROM instruction is used to readout the contents of an addressed byte or bytes of configuration EEPROM. This instruction will readout the configuration data stored in the EEPROM memory – this is not necessarily the current device configuration. The current device configuration can be readout using the READ_CFG_REG or READ_ALL_CFG_REG commands. The address map for the configuration EEPROM is the same as the configuration register map. The READ_CFG_EEPROM instruction is the only mechanism for reading out the User Electronic Signature (described in Table 70). The READ_CFG_EEPROM is a two-step transaction operation, as shown in Figure 44. In the first step, a write transaction is performed with the 0x25 instruction, and an

8-bit address code corresponding to a specific memory address (defined in the ASC Configuration Registers section). In the second step, a read transaction is used to read the EEPROM memory contents. The memory address will auto-increment to support reading multiple bytes in a single transaction. This means a single transaction can support reading the entire configuration address map (120 bytes), if the starting address of 0x00 is used. A stop condition will complete the read transaction, this can be issued after any number of bytes have been read.

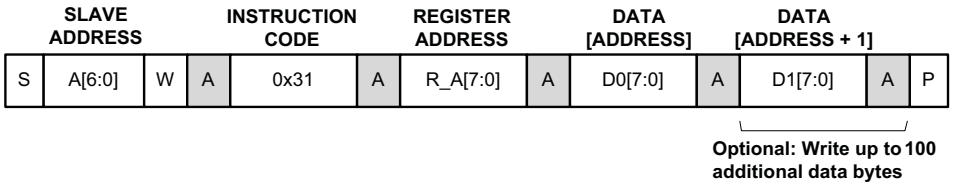
Figure 44. READ_EEPROM - I2C Instruction Format



* After final data byte read, master should NACK before issuing the STOP command

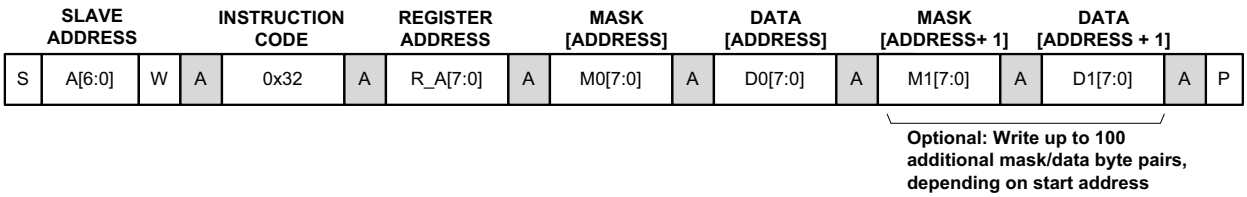
The WRITE_CFG_REG instruction is used to write configuration data to an addressed register. The instruction format includes an address byte and at least one data byte, as shown in Figure 45. Additional data bytes can be written in a single transaction as the configuration register address will increment automatically. A stop condition will complete the write transaction, this can be issued after any number of bytes have been written. The WRITE_CFG_REG instruction should be used with caution, as many of the configuration registers are used to define multiple device options. In many cases, the WRITE_CFG_REG_wMASK instruction is a more reliable method for updating a single configuration parameter. For configuration registers which support the LOAD_CFG_REG instruction, the slave shadow registers will not be updated until the LOAD_CFG_REG instruction is executed. Master-only shadow registers will be updated immediately, and in some cases will reset their circuitry (see the ASC Configuration Registers section).

Figure 45. WRITE_CFG_REG - I2C Instruction Format



The WRITE_CFG_REG_wMASK instruction is used to write the masked configuration data bits to an addressed master register. The instruction format includes an address byte and at least one mask byte / data byte pair, as shown in Figure 46. Additional mask and data byte pairs can be written in a single transaction as the configuration register address will increment automatically. A stop condition will complete the write transaction, this can be issued after any number of data and mask pairs have been written. This instruction will not modify the configuration bits set to 1 in the mask byte. Those configuration bits will keep their current value. Bit locations set to 0 in the mask byte will be modified by the data byte. For configuration registers which support the LOAD_CFG_REG instruction, the slave shadow registers will not be updated until the LOAD_CFG_REG instruction is executed. Master-only shadow registers will be updated immediately, and in some cases will reset their circuitry.

Figure 46. WRITE_CFG__REG_wMASK - I2C Instruction Format



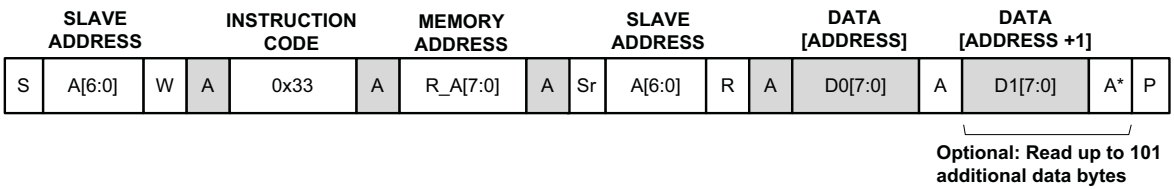
Using the WRITE_CFG_REG_wMASK Instruction Format

The WRITE_CFG_REG_wMASK instruction is the preferred instruction for updating a single programmable device parameter. As an example, the following I^2C write transaction can be used to update only the VMON4_A threshold. The example will update the A_TRIP_FINE to a value of hex 0x0A (binary 001010) See Table 28 for more details.

1. Start the bus transaction.
2. Transmit the device address (7 bits) along with a low write bit.
3. Transmit the 0x32 instruction code (WRITE_CFG_REG_wMASK.)
4. Transmit the 0x1F address byte (VMON4_CFG0 register as defined by Table 28).
5. Transmit 0x3F as the MASK0 byte (only A_TRIP_FINE[1:0] will be modified, B_TRIP_SELECT[5:0] will maintain its current configuration).
6. Transmit the data to be written to the two highest bits of VMON4_CFG0 (0x80 corresponds to A_TRIP_FINE[1:0] = 10).
7. Transmit 0xF0 as the next mask byte (address will auto-increment to 0x20, the VMON4_CFG1 register). Only A_TRIP_FINE[5:2] will be modified. Other VMON4_CFG1 parameters will be unchanged.
8. Transmit the data to be written to the four lowest bits of VMON4_CFG1 (0x02 corresponds to A_TRIP_FINE[5:2] = 0010).
9. Stop the bus transaction.
10. Start an additional bus transaction using the LOAD_CFG_REG instruction (see Figure 48).

The configuration register settings can also be readout over I^2C. This is accomplished using the READ_CFG_REG instruction and the READ_ALL_CFG_REG instruction. The READ_CFG_REG is a two-step transaction operation, as shown in Figure 47. In the first step, a write transaction is performed with the 0x33 instruction, and an 8-bit address code corresponding to a specific register address (defined in the ASC Configuration Registers section). In the second step, a read transaction is used to read the register contents. The register address will auto-increment to support reading multiple registers in a single transaction. This means a single transaction can support reading the entire configuration address map (102 bytes), if the starting address of 0x00 is used. A stop condition will complete the read transaction, this can be issued after any number of bytes have been read.

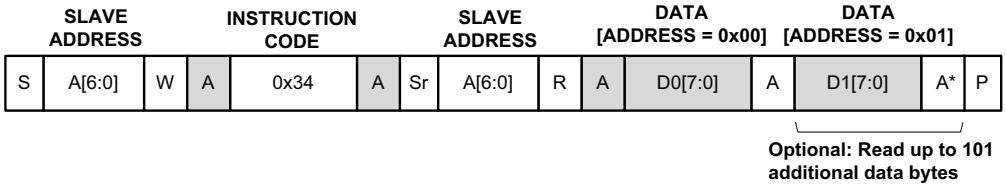
Figure 47. READ_CFG_REG - I2C Instruction Format



* After final data byte read, master should NACK before issuing the STOP command

The READ_ALL_CFG instruction works in a similar way to the READ_CFG_REG. The difference is that the READ_ALL_CFG instruction always starts at register address 0x00. Multiple data bytes can be read out in a single transaction, with the register address auto-incrementing after each byte is read. The entire configuration register memory space can be read out with a single transaction (102 data bytes). A stop condition will complete the read transaction, this can be issued after any number of bytes have been read. The format for the READ_ALL_CFG_REG instruction is shown in Figure 48.

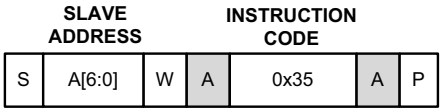
Figure 48. READ_ALL_CFG_REG - I2C Instruction Format



* After final data byte read, master should NACK before issuing the STOP command

The LOAD_CFG_REG instruction is used to load the data from the I^2C master shadow registers to the slave shadow registers. All the slave shadow registers are loaded at once when the instruction is received. The LOAD_CFG_REG instruction should be used after WRITE_CFG_REG and WRITE_CFG_REG_wMask updates to the I^2C configuration registers are completed. This instruction is useful for updating multiple parameters which affect the operation of a single circuit (such as a VMON or IMON), as these parameters are often spread across multiple configuration addresses. Note that certain configuration registers (such as temperature monitor or trim profiles) do not support this instruction. These master-only shadow registers are updated immediately by a WRITE_CFG_REG instruction, they do not require a LOAD_CFG_REG instruction. The format for the LOAD_CFG_REG instruction is shown in Figure 49.

Figure 49. LOAD_CFG_REG - I2C Instruction Format



ASC Configuration Registers

The ASC Configuration registers are grouped below by function and shown in the following tables:

- Table 22, Trim Configuration Register Summary
- Table 28, Voltage Monitor Configuration Register Summary
- Table 37, Current Monitor Configuration Register Summary
- Table 43, Temperature Monitor Configuration Register Summary
- Table 51, High Voltage Output Configuration Register Summary
- Table 60, Output Control Block Configuration Register Summary
- Table 65, GPIO Input Configuration Register Summary
- Table 67, Write Protect and User Tag Configuration Register
- Table 70, UES Memory Summary
- Table 71, Reserved Configuration Addresses

The configuration register address space is 8-bits (0x00-0xFF). The registers contain the configuration information for all the analog blocks in the ASC. These registers are automatically populated with their configuration information at power on reset, either from the ASC EEPROM memory or external memory through Dual Boot algorithm. These registers should not be confused with the *Measurement and Control* registers described in a later section. The measurement and control registers are used to read voltage, current, and temperature measurements and are accessed with a different set of instructions.

Table 22. Trim Configuration Register Summary

Register Address	Register Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Reconfiguration Details
0x00	Trim1_P1_Lo[1]	Trim1_P1_Set [7:0]								Master-Only (Immediate Update)
0x01	Trim1_Trim2_P1_Hi	Trim1_P1_Set[11:8]				Trim2_P1_Set[11:8]				
0x02	Trim2_P1_Lo[1]	Trim2_P1_Set [7:0]								
0x03	Trim3_P1_Lo[1]	Trim3_P1_Set [7:0]								
0x04	Trim3_Trim4_P1_Hi	Trim3_P1_Set [11:8]				Trim3_P1_Set [11:8]				
0x05	Trim4_P1_Lo[1]	Trim4_P1_Set [7:0]								
0x06	Trim1_P2_Lo[1]	Trim1_P2_Set [7:0]								
0x07	Trim1_Trim2_P2_Hi	Trim1_P2_Set[11:8]				Trim1_P2_Set[11:8]				
0x08	Trim2_P1_Lo[1]	Trim2_P2_Set [7:0]								
0x09	Trim3_P2_Lo[1]	Trim3_P2_Set [7:0]								
0x0A	Trim3_Trim4_P2_Hi	Trim3_P2_Set [11:8]				Trim3_P2_Set [11:8]				
0x0B	Trim4_P1_Lo[1]	Trim4_P1_Set [7:0]								
0x0C	Trim1_P0_Lo[1]	Trim1_P0_Set [7:0]								Master/Slave (LOAD_CFG_REG supported)
0x0D	Trim1_P0_Hi_Cfg	POL	BYP	ATT	x	Trim1_P0_Set[11:8]				
0x0E	Trim2_P0_Lo[1]	Trim2_P0_Set [7:0]								
0x0F	Trim2_P0_Hi_Cfg	POL	BYP	ATT	x	Trim2_P0_Set[11:8]				
0x10	Trim3_P0_Lo[1]	Trim3_P0_Set [7:0]								
0x11	Trim3_P0_Hi_Cfg	POL	BYP	ATT	x	Trim3_P0_Set[11:8]				
0x12	Trim4_P0_Lo[1]	Trim4_P0_Set [7:0]								
0x13	Trim4_P0_Hi_Cfg	POL	BYP	ATT	x	Trim4_P0_Set[11:8]				
0x14	Trim_CLT_Rate							RATE[1:0]		
0x15	Trim_DAC_BPZ	D4_BPZ[1:0]		D3_BPZ[1:0]		D2_BPZ[1:0]		D1_BPZ[1:0]		

1.When the bypass bit (manual mode) is set, the lower 8-bits of the profile set-point are the profile DAC registers shown in Figure 26.

Closed Loop Trim Configuration Registers

The ASC configuration memory specifies the operation of the closed loop trim circuitry, described in the Controlling Power Supply Output Voltage by Trim and Margin Block section. Each of the configurable parameters, shown in Table 22, are described in the following section.

Trimx_Py_Set [11:0] (Trim1_P0 … Trim4_P2) – Trim Channel Profile Setpoints 0, 1 and 2

The Trim profile setpoints are configured as 12 bit numbers, where each bit corresponds to 2 mV. The equation below (which is a reversal of the calculation equation found in the ADC section) describes how to calculate the trim target.

TRIM_SETPOINT_CODE (12_bits, converted to binary) = ROUND (Target Voltage / 2mV)

Each of the 4 Trim channels supports three separate programmable setpoints, as shown in Table 22. The P1 and P2 setpoints for each channel do not support the LOAD_CFG_REG instruction and are updated immediately after being written by I^2C instructions. It is not recommended to update these registers during operation. Updating the trim setpoint is best accomplished using the Closed Loop Trim Register Access instructions.

POL – Polarity

The Polarity setting for each trim channel determines the closed loop trim behavior of trim voltage control versus output voltage feedback, as shown in Figure 27. The polarity settings are described in Table 23.

Table 23. POL Setting vs Closed Loop Trim Polarity

POL	Closed Loop Trim Polarity
0	Positive
1	Negative

BYP – Bypass

The Bypass setting for each trim channel determines whether the trim output voltage is controlled by the closed loop trim circuitry or by the stored profile DAC codes, as shown in Figure 26. When the Trim-DAC circuitry is in bypass mode, the lower 8-bits of the profile set-point are the profile DAC registers shown in Figure 26. The bypass settings are described in Table 24.

Table 24. BYP Setting vs Trim Voltage Source

BYP	Trim Voltage Source
0	Closed Loop Trim Logic (Trim Calculator)
1	Profile DAC Code (Manual)

ATT – Attenuator Enable

The Attenuator Enable setting for each trim channel determines whether the monitored DC-DC output voltage needs to be attenuated before ADC measurement, as shown in Figure 10. DC-DC output voltages above 2 V need to be attenuated. The attenuator settings are described in Table 25.

Table 25. ATT Setting vs Attenuation Value

ATT	Attenuation Value
0	÷ 1 (no attenuation)
1	÷ 3

RATE[1:0] – Closed Loop Trim Update Rate

The Closed Loop Trim update rate is a common setting for all four trim channels. The available settings are shown in Table 26.

Table 26. RATE[1:0] Setting vs Closed Loop Trim Update Rate

RATE[1:0]	Update Rate
00	860 µs
01	1.72 ms
10	13.8 ms
11	27.6 ms

Dx_BPZ (DAC1_BPZ … DAC4_BPZ) – DAC Bi-Polar Zero Output Voltage

The DAC Bi-Polar Zero Output Voltage for each channel determines the Trim outputs Bi-Polar Zero voltage as shown in Figure 28. There are four available settings shown in Table 27.

Table 27. Dx_BPZ[1:0] Setting vs DAC Bi-Polar Zero Output Voltage

Dx_BPZ[1:0]	DAC BPZ Voltage
00	0.6V
01	0.8V
10	1.0V
11	1.25V

Voltage Monitor Configuration Registers

Table 28. Voltage Monitor Configuration Register Summary

Register Address	Register Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Reconfiguration Details
0x16	VMON1_Config0	V1_ATF[1:0]		V1_BTF[5:0]						Master/Slave (LOAD_CFG_REG supported)
0x17	VMON1_Config1	1	1	GBP	WM	V1_ATF[5:2]				
0x18	VMON1_Config2	V1_ATC[3:0]				V1_BTC[3:0]				
0x19	VMON2_Config0	V2_ATF[1:0]		V2_BTF[5:0]						
0x1A	VMON2_Config1	1	1	GBP	WM	V2_ATF[5:2]				
0x1B	VMON2_Config2	V2_ATC[3:0]				V2_BTC[3:0]				
0x1C	VMON3_Config0	V3_ATF[1:0]		V3_BTF[5:0]						
0x1D	VMON3_Config1	1	1	GBP	WM	V3_ATF[5:2]				
0x1E	VMON3_Config2	V3_ATC[3:0]				V3_BTC[3:0]				
0x1F	VMON4_Config0	V4_ATF[1:0]		V4_BTF[5:0]						
0x20	VMON4_Config1	1	1	GBP	WM	V4_ATF[5:2]				
0x21	VMON4_Config2	V4_ATC[3:0]				V4_BTC[3:0]				
0x22	VMON5_Config0	V5_ATF[1:0]		V5_BTF[5:0]						
0x23	VMON5_Config1	1	1	GBP	WM	V5_ATF[5:2]				
0x24	VMON5_Config2	V5_ATC[3:0]				V5_BTC[3:0]				
0x25	VMON6_Config0	V6_ATF[1:0]		V6_BTF[5:0]						
0x26	VMON6_Config1	1	1	GBP	WM	V6_ATF[5:2]				
0x27	VMON6_Config2	V6_ATC[3:0]				V6_BTC[3:0]				
0x28	VMON7_Config0	V7_ATF[1:0]		V7_BTF[5:0]						
0x29	VMON7_Config1	1	1	GBP	WM	V7_ATF[5:2]				
0x2A	VMON7_Config2	V7_ATC[3:0]				V7_BTC[3:0]				
0x2B	VMON8_Config0	V8_ATF[1:0]		V8_BTF[5:0]						
0x2C	VMON8_Config1	1	1	GBP	WM	V8_ATF[5:2]				
0x2D	VMON8_Config2	V8_ATC[3:0]				V8_BTC[3:0]				
0x2E	VMON9_Config0	V9_ATF[1:0]		V9_BTF[5:0]						
0x2F	VMON9_Config1	1	1	GBP	WM	V9_ATF[5:2]				
0x30	VMON9_Config2	V9_ATC[3:0]				V9_BTC[3:0]				
0x31	HVMON_Config0	HV_ATF[1:0]		HV_BTF[5:0]						
0x32	HVMON_Config1	1	1	GBP	WM	HV_ATF[5:2]				
0x33	HVMON_Config2	HV_ATC[3:0]				HV_BTC[3:0]				

The ASC configuration memory specifies the operation of the voltage monitor (VMON), described in the Voltage Monitor Inputs section. The voltage monitor (VMON1-VMON9 and HVMON) trip points, glitch filter setting, and window mode are configurable over I^2C. The configuration registers are summarized in Table 28.

Vx_ATF[5:0], Vx_ATC[3:0], Vx_BTF[5:0], Vx_BTC[3:0] (V1_ATF … V9_BTC) – Voltage Monitor Fine and Coarse, A and B Trip Points

Each voltage monitor includes programmable trip points A and B, corresponding to the two comparators for each voltage monitor input pin. The A and B trip points of the Differential Voltage Monitors (VMON1 - VMON4) are defined based on the fine and coarse settings shown in Table 29 (for over-voltage monitoring) and Table 30 (for under-voltage monitoring). The A and B trip points of the Single-Ended Voltage Monitors (VMON5-VMON9) are defined based on the fine and coarse settings shown in Table 31 (for over-voltage monitoring) and Table 32 (for under-voltage monitoring). Fine and Coarse settings outside of the table range are prohibited. There is no program-

mable setting for over or under voltage. Based on the type of voltage monitoring, choose the applicable table. For more details on over and under voltage monitoring, see the Programmable Over-Voltage and Under-Voltage Thresholds discussion in the voltage monitor inputs section. Setting the trip point to the Low-Voltage sense row (Fine Range 0x21) disables hysteresis for that voltage monitor input for both under and over voltage detection.

Table 29. Trip Point for Over-Voltage Detection (Differential VMON1-VMON4)

Fine Range Setting	Coarse Range Setting											
	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xA	0xB
0x00	0.795	0.947	1.127	1.341	1.589	1.897	2.259	2.677	3.172	3.779	4.848	5.775
0x01	0.790	0.942	1.121	1.334	1.581	1.887	2.247	2.663	3.156	3.759	4.822	5.744
0x02	0.786	0.937	1.115	1.327	1.572	1.876	2.235	2.648	3.139	3.739	4.797	5.713
0x03	0.782	0.931	1.109	1.320	1.564	1.866	2.223	2.634	3.122	3.719	4.771	5.683
0x04	0.778	0.926	1.103	1.313	1.555	1.856	2.211	2.620	3.105	3.699	4.746	5.652
0x05	0.773	0.921	1.097	1.306	1.547	1.846	2.199	2.605	3.088	3.679	4.720	5.621
0x06	0.769	0.916	1.091	1.299	1.538	1.836	2.187	2.591	3.071	3.658	4.694	5.590
0x07	0.765	0.911	1.085	1.291	1.530	1.826	2.175	2.577	3.055	3.638	4.668	5.559
0x08	0.761	0.906	1.079	1.284	1.521	1.816	2.163	2.563	3.038	3.618	4.642	5.529
0x09	0.756	0.901	1.073	1.277	1.513	1.806	2.151	2.548	3.021	3.598	4.616	5.498
0x0A	0.752	0.896	1.067	1.270	1.504	1.796	2.139	2.534	3.004	3.578	4.590	5.468
0x0B	0.748	0.891	1.061	1.263	1.497	1.786	2.127	2.520	2.987	3.558	4.565	5.437
0x0C	0.744	0.886	1.055	1.256	1.488	1.775	2.115	2.505	2.970	3.537	4.539	5.406
0x0D	0.739	0.881	1.049	1.249	1.480	1.765	2.103	2.492	2.953	3.517	4.513	5.375
0x0E	0.735	0.876	1.043	1.241	1.472	1.755	2.091	2.478	2.936	3.497	4.487	5.345
0x0F	0.731	0.871	1.037	1.234	1.463	1.745	2.079	2.464	2.919	3.478	4.462	5.314
0x10	0.727	0.866	1.031	1.227	1.455	1.735	2.066	2.449	2.902	3.458	4.436	5.283
0x11	0.723	0.861	1.025	1.220	1.446	1.725	2.054	2.435	2.885	3.438	4.410	5.252
0x12	0.718	0.856	1.019	1.213	1.438	1.715	2.042	2.421	2.868	3.417	4.384	5.221
0x13	0.714	0.851	1.013	1.206	1.429	1.705	2.030	2.406	2.851	3.397	4.359	5.191
0x14	0.710	0.846	1.007	1.199	1.421	1.695	2.018	2.392	2.835	3.377	4.333	5.160
0x15	0.706	0.841	1.001	1.191	1.412	1.685	2.006	2.378	2.819	3.357	4.307	5.130
0x16	0.701	0.836	0.995	1.184	1.404	1.674	1.994	2.364	2.802	3.337	4.281	5.099
0x17	0.697	0.831	0.989	1.177	1.395	1.664	1.982	2.349	2.785	3.317	4.255	5.068
0x18	0.693	0.826	0.983	1.170	1.387	1.654	1.970	2.335	2.768	3.296	4.229	5.038
0x19	0.689	0.821	0.977	1.163	1.378	1.644	1.958	2.321	2.751	3.276	4.203	5.007
0x1A	0.684	0.816	0.971	1.156	1.370	1.634	1.946	2.307	2.734	3.256	4.178	4.976
0x1B	0.680	0.810	0.965	1.149	1.362	1.624	1.934	2.292	2.717	3.236	4.153	4.945
0x1c	0.676	0.805	0.959	1.141	1.353	1.614	1.922	2.278	2.700	3.216	4.127	4.914
0x1d	0.672	0.800	0.953	1.134	1.345	1.604	1.910	2.264	2.683	3.196	4.101	4.884
0x1e	0.668	0.795	0.947	1.127	1.336	1.594	1.898	2.249	2.666	3.176	4.075	4.853
Low-Voltage Sense												
0x21	0.075	0.089	0.106	0.126	0.150	0.178	0.212	0.252	0.300	0.356	0.457	0.545

Table 30. Trip Point for Under-Voltage Detection (Differential VMON1-VMON4)

Fine Range Setting	Coarse Range Setting											
	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xA	0xB
0x00	0.786	0.937	1.115	1.327	1.572	1.876	2.235	2.648	3.139	3.739	4.797	5.713
0x01	0.782	0.931	1.109	1.320	1.564	1.866	2.223	2.634	3.122	3.719	4.771	5.683
0x02	0.778	0.926	1.103	1.313	1.555	1.856	2.211	2.620	3.105	3.699	4.746	5.652
0x03	0.773	0.921	1.097	1.306	1.547	1.846	2.199	2.605	3.088	3.679	4.720	5.621
0x04	0.769	0.916	1.091	1.299	1.538	1.836	2.187	2.591	3.071	3.658	4.694	5.590
0x05	0.765	0.911	1.085	1.291	1.530	1.826	2.175	2.577	3.055	3.638	4.668	5.559
0x06	0.761	0.906	1.079	1.284	1.521	1.816	2.163	2.563	3.038	3.618	4.642	5.529
0x07	0.756	0.901	1.073	1.277	1.513	1.806	2.151	2.548	3.021	3.598	4.616	5.498
0x08	0.752	0.896	1.067	1.270	1.504	1.796	2.139	2.534	3.004	3.578	4.590	5.468
0x09	0.748	0.891	1.061	1.263	1.497	1.786	2.127	2.520	2.987	3.558	4.565	5.437
0x0A	0.744	0.886	1.055	1.256	1.488	1.775	2.115	2.505	2.970	3.537	4.539	5.406
0x0B	0.739	0.881	1.049	1.249	1.480	1.765	2.103	2.492	2.953	3.517	4.513	5.375
0x0C	0.735	0.876	1.043	1.241	1.472	1.755	2.091	2.478	2.936	3.497	4.487	5.345
0x0D	0.731	0.871	1.037	1.234	1.463	1.745	2.079	2.464	2.919	3.478	4.462	5.314
0x0E	0.727	0.866	1.031	1.227	1.455	1.735	2.066	2.449	2.902	3.458	4.436	5.283
0x0F	0.723	0.861	1.025	1.220	1.446	1.725	2.054	2.435	2.885	3.438	4.410	5.252
0x10	0.718	0.856	1.019	1.213	1.438	1.715	2.042	2.421	2.868	3.417	4.384	5.221
0x11	0.714	0.851	1.013	1.206	1.429	1.705	2.030	2.406	2.851	3.397	4.359	5.191
0x12	0.710	0.846	1.007	1.199	1.421	1.695	2.018	2.392	2.835	3.377	4.333	5.160
0x13	0.706	0.841	1.001	1.191	1.412	1.685	2.006	2.378	2.819	3.357	4.307	5.130
0x14	0.701	0.836	0.995	1.184	1.404	1.674	1.994	2.364	2.802	3.337	4.281	5.099
0x15	0.697	0.831	0.989	1.177	1.395	1.664	1.982	2.349	2.785	3.317	4.255	5.068
0x16	0.693	0.826	0.983	1.170	1.387	1.654	1.970	2.335	2.768	3.296	4.229	5.038
0x17	0.689	0.821	0.977	1.163	1.378	1.644	1.958	2.321	2.751	3.276	4.203	5.007
0x18	0.684	0.816	0.971	1.156	1.370	1.634	1.946	2.307	2.734	3.256	4.178	4.976
0x19	0.680	0.810	0.965	1.149	1.362	1.624	1.934	2.292	2.717	3.236	4.153	4.945
0x1A	0.676	0.805	0.959	1.141	1.353	1.614	1.922	2.278	2.700	3.216	4.127	4.914
0x1B	0.672	0.800	0.953	1.134	1.345	1.604	1.910	2.264	2.683	3.196	4.101	4.884
0x1c	0.668	0.795	0.947	1.127	1.336	1.594	1.898	2.249	2.666	3.176	4.075	4.853
0x1d	0.663	0.790	0.941	1.120	1.328	1.584	1.886	2.235	2.649	3.156	4.049	4.822
0x1e	0.659	0.785	0.935	1.113	1.319	1.573	1.874	2.221	2.632	3.136	4.023	4.792
Low-Voltage Sense												
0x21	0.075	0.089	0.106	0.126	0.150	0.178	0.212	0.252	0.300	0.356	0.457	0.545

Table 31. Trip Point for Over-Voltage Detection (Single-Ended VMON5-VMON9)

Fine Range Setting	Coarse Range Setting											
	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb
0x0	0.799	0.952	1.133	1.347	1.597	1.907	2.270	2.688	3.185	3.794	4.868	5.798
0x1	0.794	0.947	1.126	1.340	1.589	1.897	2.258	2.674	3.168	3.774	4.842	5.767
0x2	0.790	0.942	1.120	1.333	1.580	1.886	2.246	2.659	3.151	3.754	4.816	5.736
0x3	0.786	0.936	1.114	1.326	1.572	1.875	2.234	2.645	3.134	3.734	4.790	5.706
0x4	0.782	0.931	1.108	1.319	1.563	1.865	2.222	2.631	3.117	3.714	4.765	5.675
0x5	0.777	0.926	1.102	1.312	1.555	1.855	2.210	2.616	3.100	3.694	4.739	5.644
0x6	0.773	0.921	1.096	1.305	1.546	1.845	2.198	2.602	3.083	3.673	4.713	5.613
0x7	0.769	0.916	1.090	1.297	1.538	1.835	2.186	2.588	3.067	3.653	4.687	5.582
0x8	0.765	0.911	1.084	1.290	1.529	1.825	2.174	2.574	3.050	3.633	4.661	5.552
0x9	0.760	0.906	1.078	1.283	1.521	1.815	2.162	2.559	3.033	3.613	4.635	5.521
0xa	0.756	0.901	1.072	1.276	1.512	1.805	2.150	2.545	3.016	3.593	4.609	5.489
0xb	0.752	0.896	1.066	1.269	1.504	1.795	2.138	2.531	2.999	3.573	4.584	5.458
0xc	0.748	0.891	1.060	1.262	1.495	1.784	2.125	2.516	2.982	3.552	4.558	5.427
0xd	0.743	0.886	1.054	1.255	1.487	1.774	2.113	2.501	2.965	3.532	4.532	5.396
0xe	0.739	0.881	1.048	1.247	1.479	1.764	2.101	2.487	2.948	3.512	4.506	5.366
0xf	0.735	0.875	1.042	1.240	1.470	1.754	2.089	2.473	2.931	3.491	4.479	5.335
0x10	0.731	0.870	1.036	1.233	1.462	1.744	2.076	2.458	2.914	3.471	4.453	5.304
0x11	0.727	0.865	1.030	1.226	1.453	1.734	2.064	2.444	2.897	3.451	4.427	5.273
0x12	0.722	0.860	1.024	1.219	1.445	1.724	2.052	2.430	2.880	3.430	4.401	5.242
0x13	0.718	0.855	1.018	1.212	1.436	1.714	2.040	2.415	2.863	3.410	4.376	5.212
0x14	0.714	0.850	1.012	1.205	1.428	1.704	2.028	2.401	2.847	3.390	4.350	5.181
0x15	0.710	0.845	1.006	1.197	1.419	1.694	2.016	2.387	2.830	3.370	4.324	5.150
0x16	0.705	0.840	1.000	1.190	1.411	1.683	2.004	2.373	2.813	3.350	4.298	5.119
0x17	0.701	0.835	0.994	1.183	1.402	1.673	1.992	2.358	2.796	3.330	4.272	5.088
0x18	0.697	0.830	0.988	1.176	1.394	1.663	1.980	2.344	2.779	3.309	4.246	5.058
0x19	0.693	0.825	0.982	1.169	1.385	1.653	1.968	2.330	2.762	3.289	4.220	5.027
0x1a	0.688	0.820	0.976	1.162	1.376	1.643	1.956	2.316	2.745	3.269	4.195	4.996
0x1b	0.684	0.814	0.970	1.155	1.368	1.633	1.944	2.301	2.728	3.249	4.169	4.965
0x1c	0.680	0.809	0.964	1.147	1.359	1.622	1.932	2.287	2.711	3.229	4.143	4.934
0x1d	0.676	0.804	0.958	1.140	1.351	1.612	1.920	2.273	2.694	3.209	4.117	4.904
0x1e	0.672	0.799	0.952	1.133	1.342	1.602	1.908	2.258	2.677	3.189	4.091	4.873
Low-Voltage Sense												
0x21	0.080	0.093	0.110	0.132	0.155	0.186	0.220	0.262	0.310	0.370	0.475	0.565

Table 32. Trip Point for Under-Voltage Detection (Single-Ended VMON5-VMON9)

Fine Range Setting	Coarse Range Setting											
	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb
0x0	0.790	0.942	1.120	1.333	1.580	1.886	2.246	2.659	3.151	3.754	4.816	5.736
0x1	0.786	0.936	1.114	1.326	1.572	1.875	2.234	2.645	3.134	3.734	4.790	5.706
0x2	0.782	0.931	1.108	1.319	1.563	1.865	2.222	2.631	3.117	3.714	4.765	5.675
0x3	0.777	0.926	1.102	1.312	1.555	1.855	2.210	2.616	3.100	3.694	4.739	5.644
0x4	0.773	0.921	1.096	1.305	1.546	1.845	2.198	2.602	3.083	3.673	4.713	5.613
0x5	0.769	0.916	1.090	1.297	1.538	1.835	2.186	2.588	3.067	3.653	4.687	5.582
0x6	0.765	0.911	1.084	1.290	1.529	1.825	2.174	2.574	3.050	3.633	4.661	5.552
0x7	0.760	0.906	1.078	1.283	1.521	1.815	2.162	2.559	3.033	3.613	4.635	5.521
0x8	0.756	0.901	1.072	1.276	1.512	1.805	2.150	2.545	3.016	3.593	4.609	5.489
0x9	0.752	0.896	1.066	1.269	1.504	1.795	2.138	2.531	2.999	3.573	4.584	5.458
0xa	0.748	0.891	1.060	1.262	1.495	1.784	2.125	2.516	2.982	3.552	4.558	5.427
0xb	0.743	0.886	1.054	1.255	1.487	1.774	2.113	2.501	2.965	3.532	4.532	5.396
0xc	0.739	0.881	1.048	1.247	1.479	1.764	2.101	2.487	2.948	3.512	4.506	5.366
0xd	0.735	0.875	1.042	1.240	1.470	1.754	2.089	2.473	2.931	3.491	4.479	5.335
0xe	0.731	0.870	1.036	1.233	1.462	1.744	2.076	2.458	2.914	3.471	4.453	5.304
0xf	0.727	0.865	1.030	1.226	1.453	1.734	2.064	2.444	2.897	3.451	4.427	5.273
0x10	0.722	0.860	1.024	1.219	1.445	1.724	2.052	2.430	2.880	3.430	4.401	5.242
0x11	0.718	0.855	1.018	1.212	1.436	1.714	2.040	2.415	2.863	3.410	4.376	5.212
0x12	0.714	0.850	1.012	1.205	1.428	1.704	2.028	2.401	2.847	3.390	4.350	5.181
0x13	0.710	0.845	1.006	1.197	1.419	1.694	2.016	2.387	2.830	3.370	4.324	5.150
0x14	0.705	0.840	1.000	1.190	1.411	1.683	2.004	2.373	2.813	3.350	4.298	5.119
0x15	0.701	0.835	0.994	1.183	1.402	1.673	1.992	2.358	2.796	3.330	4.272	5.088
0x16	0.697	0.830	0.988	1.176	1.394	1.663	1.980	2.344	2.779	3.309	4.246	5.058
0x17	0.693	0.825	0.982	1.169	1.385	1.653	1.968	2.330	2.762	3.289	4.220	5.027
0x18	0.688	0.820	0.976	1.162	1.376	1.643	1.956	2.316	2.745	3.269	4.195	4.996
0x19	0.684	0.814	0.970	1.155	1.368	1.633	1.944	2.301	2.728	3.249	4.169	4.965
0x1a	0.680	0.809	0.964	1.147	1.359	1.622	1.932	2.287	2.711	3.229	4.143	4.934
0x1b	0.676	0.804	0.958	1.140	1.351	1.612	1.920	2.273	2.694	3.209	4.117	4.904
0x1c	0.672	0.799	0.952	1.133	1.342	1.602	1.908	2.258	2.677	3.189	4.091	4.873
0x1d	0.667	0.794	0.946	1.125	1.334	1.592	1.896	2.244	2.660	3.168	4.065	4.842
0x1e	0.663	0.789	0.940	1.118	1.325	1.581	1.884	2.230	2.643	3.148	4.039	4.811
Low-Voltage Sense												
0x21	0.080	0.093	0.110	0.132	0.155	0.186	0.220	0.262	0.310	0.370	0.475	0.565

GBP – Glitch Filter Bypass

Each of the voltage monitors include a glitch filter at each of the trip point comparator outputs as shown in Figure 6. This glitch filter can be bypassed dependent on the GBP setting shown in Table 33.

Table 33. GBP Setting vs Glitch Bypass Behavior

GBP	Glitch Filter Setting
0	Glitch Filter On
1	Glitch Filter Bypassed

WM – Window Mode

Each of the voltage monitors include a selectable window mode, as described in Table 2. The window mode setting is shown in Table 34.

Table 34. WM Setting vs Window Mode Value

WM	Window Mode
0	Off
1	On

HV_ATF[5:0], HV_ATC[3:0], HV_BTF[5:0], HV_BTC[3:0] – High Voltage Monitor Fine and Coarse, A and B Trip Points

The High Voltage Monitor (HVMON) is configured in a similar fashion to the low voltage monitor inputs. The key difference from the low voltage monitor inputs is the trip point table. The HVMON range is up to 13.2 V, as reflected in Table 35 (Over-Voltage Trip Points) and Table 36 (Under-Voltage Trip Points). Setting the trip point to the Low-Voltage sense row (Fine Range 0x21) disables hysteresis for that voltage monitor input for both under and over voltage detection.

Table 35. Trip-Point for Over-Voltage Detection (HVMON)

Fine Range Setting	Coarse Range Setting											
	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb
0x0	2.269	2.694	3.193	3.748	4.421	5.207	6.137	7.160	8.382	9.819	11.455	13.218
0x1	2.257	2.680	3.176	3.729	4.398	5.179	6.104	7.121	8.337	9.767	11.394	13.147
0x2	2.245	2.666	3.159	3.709	4.374	5.152	6.071	7.083	8.293	9.714	11.333	13.077
0x3	2.233	2.651	3.142	3.689	4.351	5.124	6.039	7.045	8.248	9.662	11.272	13.007
0x4	2.221	2.637	3.125	3.669	4.327	5.096	6.006	7.007	8.204	9.610	11.212	12.936
0x5	2.208	2.623	3.108	3.649	4.304	5.068	5.974	6.969	8.159	9.558	11.151	12.866
0x6	2.196	2.608	3.091	3.629	4.280	5.041	5.941	6.931	8.114	9.505	11.090	12.796
0x7	2.184	2.594	3.074	3.609	4.257	5.013	5.908	6.893	8.070	9.453	11.029	12.725
0x8	2.172	2.580	3.057	3.589	4.233	4.985	5.876	6.855	8.025	9.401	10.968	12.655
0x9	2.160	2.565	3.040	3.569	4.210	4.958	5.843	6.817	7.981	9.349	10.907	12.585
0xa	2.148	2.551	3.023	3.549	4.186	4.930	5.810	6.779	7.936	9.297	10.846	12.515
0xb	2.136	2.537	3.006	3.529	4.163	4.902	5.778	6.741	7.891	9.244	10.785	12.444
0xc	2.124	2.522	2.989	3.509	4.139	4.875	5.745	6.703	7.847	9.192	10.724	12.374
0xd	2.112	2.508	2.972	3.489	4.116	4.847	5.712	6.664	7.802	9.140	10.663	12.304
0xe	2.100	2.494	2.955	3.469	4.092	4.819	5.680	6.626	7.758	9.088	10.602	12.233
0xf	2.088	2.479	2.938	3.449	4.069	4.791	5.647	6.588	7.713	9.035	10.541	12.163
0x10	2.076	2.465	2.921	3.429	4.045	4.764	5.614	6.550	7.669	8.983	10.480	12.093
0x11	2.064	2.451	2.904	3.410	4.021	4.736	5.582	6.512	7.624	8.931	10.419	12.022
0x12	2.052	2.436	2.887	3.390	3.998	4.708	5.549	6.474	7.579	8.879	10.358	11.952
0x13	2.040	2.422	2.870	3.370	3.974	4.681	5.517	6.436	7.535	8.826	10.298	11.882
0x14	2.027	2.408	2.853	3.350	3.951	4.653	5.484	6.398	7.490	8.774	10.237	11.811
0x15	2.015	2.393	2.836	3.330	3.927	4.625	5.451	6.360	7.446	8.722	10.176	11.741
0x16	2.003	2.379	2.819	3.310	3.904	4.598	5.419	6.322	7.401	8.670	10.115	11.671
0x17	1.991	2.365	2.803	3.290	3.880	4.570	5.386	6.284	7.356	8.618	10.054	11.601
0x18	1.979	2.350	2.786	3.270	3.857	4.542	5.353	6.246	7.312	8.565	9.993	11.530
0x19	1.967	2.336	2.769	3.250	3.833	4.515	5.321	6.207	7.267	8.513	9.932	11.460
0x1a	1.955	2.322	2.752	3.230	3.810	4.487	5.288	6.169	7.223	8.461	9.871	11.390
0x1b	1.943	2.307	2.735	3.210	3.786	4.459	5.255	6.131	7.178	8.409	9.810	11.319
0x1c	1.931	2.293	2.718	3.190	3.763	4.431	5.223	6.093	7.134	8.356	9.749	11.249
0x1d	1.919	2.279	2.701	3.170	3.739	4.404	5.190	6.055	7.089	8.304	9.688	11.179
0x1e	1.907	2.264	2.684	3.150	3.716	4.376	5.157	6.017	7.044	8.252	9.627	11.108
Low-Voltage Sense												
0x21	0.220	0.260	0.308	0.361	0.425	0.504	0.593	0.692	0.810	0.949	1.108	1.280

Table 36. Trip-Point for Under-Voltage Detection (HVMON)

Fine Range Setting	Coarse Range Setting											
	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb
0x0	2.245	2.666	3.159	3.709	4.374	5.152	6.071	7.083	8.293	9.714	11.333	13.077
0x1	2.233	2.651	3.142	3.689	4.351	5.124	6.039	7.045	8.248	9.662	11.272	13.007
0x2	2.221	2.637	3.125	3.669	4.327	5.096	6.006	7.007	8.204	9.610	11.212	12.936
0x3	2.208	2.623	3.108	3.649	4.304	5.068	5.974	6.969	8.159	9.558	11.151	12.866
0x4	2.196	2.608	3.091	3.629	4.280	5.041	5.941	6.931	8.114	9.505	11.090	12.796
0x5	2.184	2.594	3.074	3.609	4.257	5.013	5.908	6.893	8.070	9.453	11.029	12.725
0x6	2.172	2.580	3.057	3.589	4.233	4.985	5.876	6.855	8.025	9.401	10.968	12.655
0x7	2.160	2.565	3.040	3.569	4.210	4.958	5.843	6.817	7.981	9.349	10.907	12.585
0x8	2.148	2.551	3.023	3.549	4.186	4.930	5.810	6.779	7.936	9.297	10.846	12.515
0x9	2.136	2.537	3.006	3.529	4.163	4.902	5.778	6.741	7.891	9.244	10.785	12.444
0xa	2.124	2.522	2.989	3.509	4.139	4.875	5.745	6.703	7.847	9.192	10.724	12.374
0xb	2.112	2.508	2.972	3.489	4.116	4.847	5.712	6.664	7.802	9.140	10.663	12.304
0xc	2.100	2.494	2.955	3.469	4.092	4.819	5.680	6.626	7.758	9.088	10.602	12.233
0xd	2.088	2.479	2.938	3.449	4.069	4.791	5.647	6.588	7.713	9.035	10.541	12.163
0xe	2.076	2.465	2.921	3.429	4.045	4.764	5.614	6.550	7.669	8.983	10.480	12.093
0xf	2.064	2.451	2.904	3.410	4.021	4.736	5.582	6.512	7.624	8.931	10.419	12.022
0x10	2.052	2.436	2.887	3.390	3.998	4.708	5.549	6.474	7.579	8.879	10.358	11.952
0x11	2.040	2.422	2.870	3.370	3.974	4.681	5.517	6.436	7.535	8.826	10.298	11.882
0x12	2.027	2.408	2.853	3.350	3.951	4.653	5.484	6.398	7.490	8.774	10.237	11.811
0x13	2.015	2.393	2.836	3.330	3.927	4.625	5.451	6.360	7.446	8.722	10.176	11.741
0x14	2.003	2.379	2.819	3.310	3.904	4.598	5.419	6.322	7.401	8.670	10.115	11.671
0x15	1.991	2.365	2.803	3.290	3.880	4.570	5.386	6.284	7.356	8.618	10.054	11.601
0x16	1.979	2.350	2.786	3.270	3.857	4.542	5.353	6.246	7.312	8.565	9.993	11.530
0x17	1.967	2.336	2.769	3.250	3.833	4.515	5.321	6.207	7.267	8.513	9.932	11.460
0x18	1.955	2.322	2.752	3.230	3.810	4.487	5.288	6.169	7.223	8.461	9.871	11.390
0x19	1.943	2.307	2.735	3.210	3.786	4.459	5.255	6.131	7.178	8.409	9.810	11.319
0x1a	1.931	2.293	2.718	3.190	3.763	4.431	5.223	6.093	7.134	8.356	9.749	11.249
0x1b	1.919	2.279	2.701	3.170	3.739	4.404	5.190	6.055	7.089	8.304	9.688	11.179
0x1c	1.907	2.264	2.684	3.150	3.716	4.376	5.157	6.017	7.044	8.252	9.627	11.108
0x1d	1.895	2.250	2.667	3.130	3.692	4.348	5.125	5.979	7.000	8.200	9.566	11.038
0x1e	1.883	2.236	2.650	3.110	3.669	4.321	5.092	5.941	6.955	8.148	9.505	10.968
Low-Voltage Sense												
0x21	0.220	0.260	0.308	0.361	0.425	0.504	0.593	0.692	0.810	0.949	1.108	1.280

Current Monitor Configuration Registers

The ASC configuration memory defines the operation of the current monitor (IMON1/HIMON) circuitry, described in the Theory of Operation section. The low and high voltage current monitor trip points, glitch filter setting, and window mode are configurable over I^2C. The IMON1 (low voltage) also includes a Low-Side bit, which configures the low-side sense setting on IMON1.

The configuration registers are described in Table 37.

Table 37. Current Monitor Configuration Register Summary

Register Address	Register Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Reconfiguration Details
0x34	IMON1_Config0	FAST_TH[2:0]			GBP	WM	LSS	0	0	Master/Slave (LOAD_CFG_REG supported)
0x35	IMON1_Config1	A_TH[1:0]		B_TH[1:0]		A_GAIN[1:0]		B_GAIN[1:0]		
0x36	HIMON_Config0	FAST_TH[2:0]			GBP	WM	X	0	0	
0x37	HIMON_Config1	A_TH[1:0]		B_TH[1:0]		A_GAIN[1:0]		B_GAIN[1:0]		

A_TH[1:0], B_TH[1:0], A_GAIN[1:0], B_GAIN[1:0] – Threshold and Gain Setting for A and B Comparators

The A and B current monitor trip points are defined by the combination of the threshold and gain settings. Table 38 shows the trip point settings for both the IMON1 and HIMON current monitor circuits.

Table 38. Current Monitor Trip Points (Differential Voltage)

A_TH/B_TH[1:0]	GAIN[1:0]			
	00 (GAIN = 100V/V)	01 (GAIN = 50V/V)	10 (GAIN = 25V/V)	11 (GAIN = 10V/V)
00	8 mV	15.5 mV	30.5 mV	75 mV
01	10.5 mV	20.5 mV	40.5 mV	100 mV
10	14.5 mV	28.5 mV	56.5 mV	140 mV
11	20 mV	39 mV	77 mV	190 mV

GBP – Glitch Filter Bypass

Each of the current monitors include a glitch filter at each of the trip point comparator outputs as shown in Figure 8. This glitch filter can be bypassed dependent on the GBP setting shown in Table 39.

Table 39. GBP Setting vs Glitch Bypass Behavior

GBP	Glitch Filter Setting
0	Glitch Filter On
1	Glitch Filter Bypassed

WM – Window Mode

Each of the current monitors include a selectable window mode, as described in Table 5. The window mode setting is shown in Table 40.

Table 40. WM Setting vs Window Mode Value

WM	Window Mode
0	Off
1	On

LSS – Low Side Sense Mode

The IMON1 current monitor includes a low side sense mode, as shown in Figure 8. The low side sense settings are shown in Table 41.

Table 41. LSS Setting vs Low Side Sensing Mode

LSS	Low Side Sense Mode
0	Disabled
1	Enabled

FAST_TH[2:0] - Fast Comparator Threshold

The fast trip point for both IMON1 and HIMON is set according to the FAST_TH[2:0] code. Table 42 shows the fast trip point settings vs the FAST_THRESH code for both IMON1 and HIMON current monitor circuits.

Table 42. Fast Current Monitor Trip Points (Differential Voltage)

FAST_TH[2:0]	Trip Point
000	50 mV
001	100 mV
010	150 mV
011	200 mV
100	250 mV
101	300 mV
110	400 mV
111	500 mV

Temperature Monitor Configuration Registers

Table 43. Temperature Monitor Configuration Register Summary

Register Address	Register Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Reconfiguration Details
0x38	TMON1_Config0	Ideality_Code[13:6]								Master-Only (TMON circuit resets after each time configuration update)
0x39	TMON1_Config1	Ideality_Code[5:0]						Cfg[1:0]		
0x3A	TMON1_Config2	Off[8:1]								
0x3B	TMON1_Config3	Th_A[8:2]							Off[0]	
0x3C	TMON1_Config4	Th_B[8:3]						Th_A[1:0]		
0x3D	TMON1_Config5	X	X	FLT	AVE[1:0]		Th_B[2:0]			
0x3E	TMON1_Config6	FilterA[3:0]				FilterB[3:0]				
0x3F	TMON1_Config7	X	HystA[6:0]							
0x40	TMON1_Config8	X	HystB[6:0]							
0x41	TMON2_Config0	Ideality_Code[13:6]								
0x42	TMON2_Config1	Ideality_Code[5:0]						Cfg[1:0]		
0x43	TMON2_Config2	Off[8:1]								
0x44	TMON2_Config3	Th_A[8:2]							Off[0]	
0x45	TMON2_Config4	Th_B[8:3]						Th_A[1:0]		
0x46	TMON2_Config5	X	X	FLT	AVE[1:0]		Th_B[2:0]			
0x47	TMON2_Config6	FilterA[3:0]				FilterB[3:0]				
0x48	TMON2_Config7	X	HystA[6:0]							
0x49	TMON2_Config8	X	HystB[6:0]							
0x4A	TMONint_Config0	Ideality_Code[13:6]								
0x4B	TMONint_Config1	Ideality_Code[5:0]						Cfg[1:0]		
0x4C	TMONint_Config2	Off[8:1]								
0x4D	TMONint_Config3	Th_A[8:2]							Off[0]	
0x4E	TMONint_Config4	Th_B[8:3]						Th_A[1:0]		
0x4F	TMONint_Config5	X	X	FLT	AVE[1:0]		Th_B[2:0]			
0x50	TMONint_Config6	FilterA[3:0]				FilterB[3:0]				
0x51	TMONint_Config7	X	HystA[6:0]							
0x52	TMONint_Config8	X	HystB[6:0]							

The temperature monitor circuit (TMON) configuration registers can be updated over I^2C. The definition and function of these parameters is described in the Temperature Monitor Inputs section. There are nine configuration registers per TMON channel (TMON1, TMON2, and TMON_int). The diode ideality factor, transducer configuration, temperature offset, A and B monitor characteristics (threshold, filter, hysteresis) and measurement averaging and fault behavior are all configurable according to the format in Table 43. A description of how to calculate each parameter follows the register format. The temperature monitor circuit will reset each time a configuration parameter is updated over I^2C.

Ideality_Code[13:0] - Ideality Factor Setting

The Temperature Monitor inputs support a programmable ideality factor (emission coefficient) for interfacing to different remote transistor diodes. The programmable ideality factor is calculated based on a 14-bit code. The allowed range of ideality factors is 0.9 to 2.0, values outside this range are not allowed. Calculating the code for a given ideality factor is done using the following calculation:

Ideality_Code (14 bits, converted to binary) = ROUND (4572 / ideality factor)

Table 44 shows some common ideality factors and their corresponding codes.

Table 44. Ideality Factor vs Ideality_Code Setting

Ideality_Code[13:0]	Ideality Factor
0x08EE	2.0000
---	---
0x11B6	1.0083
0x11B7	1.0082
0x11B8	1.0079
---	---
0x11C8	1.0044
0x11C9	1.0042
0x11CA	1.0039
---	---
0x11D9	1.0007
0x11DA	1.0004
0x11DB	1.0002
0x11DC	1.0000
0x11DD	0.9998
0x11DE	0.9996
0x11DF	0.9993
---	---
0x13D8	0.9000

Cfg[1:0] - Temperature Monitor Diode Configuration

As described in the Temperature Monitor Inputs section, the TMON supports different transistor-based diode configurations for connection to the ASC Temperature Monitors. The two bit value Tran_cfg[1:0], corresponds to the supported configurations as shown in Table 45.

Table 45. Temperature Monitor Diode Configuration Settings

Cfg[1:0]	Diode Configuration
00	TMON disabled
01	Beta Compensated PNP
10	Differential PNP or NPN
11	Single-Ended (Not recommended)

Off[8:0] - Temperature Monitor Offset

The TMON supports a 9-bit programmable temperature offset, which is applied to the temperature measurement for both readout and the A and B monitor comparison. The programmable offset range is from –64 °C to 63.75 °C, with a resolution of 0.25 °C. The offset is stored as a 2's complement number, with the 9th bit as the signed bit. Table 46 shows the settings associated with several different offset temperatures.

Table 46. Temperature Monitor Offset Settings

Off[8:0]	Offset Temperature °C
0x0FF	63.75
0x0FE	63.50

0x002	0.50
0x001	0.25
0x000	0.00
0x1FF	–0.25
0x1FE	–0.50

0x101	–63.75
0x100	–64.00

Th_A[8:0], Th_B[8:0] - Comparator Thresholds for A and B alarms

The TMON includes two individually programmable comparators, TMONA and TMONB. The 9-bit alarm thresholds range for each of these monitors is –64 °C to 155 °C, with a resolution of 1 °C. The thresholds are stored as 2's complement numbers, with the 9th bit as the signed bit. Values above 155 °C or below –64 °C are not valid threshold settings. Table 47 shows the settings associated with several different threshold temperatures.

Table 47. Temperature Monitor Thresholds Settings

Th_A / Th_B[8:0]	Threshold Temperature °C
0x09B	155
0x09A	154

0x002	2
0x001	1
0x000	0
0x1FF	–1
0x1FE	–2

0x1C1	–63
0x1C0	–64

FLT - Fault Reading Setting

The TMON circuit includes open and short fault detection circuitry for the remote diode channels. (The fault detect is not applicable for the internal temperature monitor, TMON_INT). The 1-bit programmable fault setting determines the measurement readout behavior of both open and short faults. The readout values compared to the fault setting is shown in Table 48.

Table 48. Temperature Monitor Fault Setting

FLT	Short Condition Reading		Open Condition Reading	
	°C	Code	°C	Code
0	255.75	0x3FF	–255.75	0x401
1	–255.75	0x401	255.75	0x3FF

AVE[1:0] - Average Filter Coefficient

The TMON temperature measurement can be read out over I^2C. The TMON circuit includes a programmable exponential averaging filter that is applied before the measurement readout. The Average parameter can be programmed to three different averaging coefficients, as shown in Table 49.

Table 49. Temperature Monitor Measurement Average Settings

AVE[1:0]	Coefficient
00	1
01	8
10	16
11	N/A

FilterA[3:0] / FilterB[3:0] - Monitor Alarm Filter

The TMONA and TMONB comparators each support programmable monitor alarm filters. The depth of the alarm filter can be programmed between 1 and 16, based on the Filter[3:0] setting. The relationship between the filter code and the filter depth is given by the following equation:

DEPTH = Filter[3:0] + 1

HystA[3:0] / HystB[3:0] - Temperature Monitor Hysteresis

The TMONA and TMONB comparators each support programmable temperature hysteresis. The 7-bit hysteresis range for each of these monitors is –64 ºC to 63 ºC, with a resolution of 1 ºC. (The negative hysteresis range should be applied to over-temperature comparisons, while the positive hysteresis should be applied to under-temperature comparisons.) The hysteresis settings are stored as 2's complement numbers, with the 7th bit as the signed bit. Table 50 shows the settings associated with several different hysteresis temperatures.

Table 50. Temperature Monitor Hysteresis Settings

Hyst[6:0]	Temperature Hysteresis ºC
0x3F	63
0x3E	62

0x02	2
0x01	1
0x00	0
0x7F	–1
0x7E	–2

0x41	–63
0x40	–64

High Voltage Output (HVOUT) Configuration Registers

Table 51. High Voltage Output Configuration Register Summary

Register Address	Register Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Reconfiguration Details
0x53	HVOUT1_Config0	OCB	1	I_SRC[1:0]		I_SNK[1:0]		VPP[1:0]		Master/Slave (LOAD_CFG_REG supported)
0x54	HVOUT1_Config1	SW	FR	OD	X	DUTY[3:0]				
0x55	HVOUT2_Config0	OCB	1	I_SRC[1:0]		I_SNK[1:0]		VPP[1:0]		
0x56	HVOUT2_Config1	SW	FR	OD	X	DUTY[3:0]				
0x57	HVOUT3_Config0	OCB	1	I_SRC[1:0]		I_SNK[1:0]		VPP[1:0]		
0x58	HVOUT3_Config1	SW	FR	OD	X	DUTY[3:0]				
0x59	HVOUT4_Config0	OCB	1	I_SRC[1:0]		I_SNK[1:0]		VPP[1:0]		
0x5A	HVOUT4_Config1	SW	FR	OD	X	DUTY[3:0]				

The High Voltage Output pins (HVOUT) configuration registers can be updated over I^2C. The definition and function of these parameters is described in the High Voltage Outputs section. There are two configuration registers per HVOUT (HVOUT1, HVOUT2, HVOUT3, HVOUT4). The open-drain/charge pump setting, charge pump voltage, source and sink current, switched/static mode, and switched mode duty cycle and frequency are all configurable according to the format in Table 51. A description of how to calculate each parameter follows the register format.

OCB - Output Control Block Source

The OCB parameter is used to select the control signal source for the HVOUT pin from either the Output Control Block (OCB) or the ASC-I/F. Table 52 shows the available settings.

Table 52. OCB Setting vs HVOUT Source Selection

OCB	HVOUT Source
0	ASC-I/F Signal
1	OCB

I_SRC[1:0] - HVOUT Source Current

The I_SRC[1:0] setting is used to choose between the four supported source currents for the HVOUT in charge pump mode. The available choices are shown in Table 53.

Table 53. HVOUT Source Current Settings

I_SRC[1:0]	Output Source Current
00	12.5 uA
01	25 uA
10	50 uA
11	100 uA

I_SNK[1:0] - HVOUT Sink Current

The I_SNK[1:0] setting is used to choose between the four supported sink currents for the HVOUT in charge pump mode. The available choices are shown in Table 54.

Table 54. HVOUT Sink Current Settings

I_SNK[1:0]	Output Sink Current
00	100 uA
01	250 uA
10	500 uA
11	3000 uA

VPP[1:0] - Charge Pump Output Voltage Settings
The VPP[1:0] setting is used to choose between the four programmable output voltage levels for the HVOUT in charge pump mode. The available choices are shown in Table 55.

Table 55. HVOUT Output Voltage Settings

VPP[1:0]	Output Voltage
00	6V
01	8V
10	10V
11	12V

SW - Switched Output Setting
The SW parameter configures the HVOUT pin for either static drive mode (on or off) or switched mode (either switched at the programmed frequency and duty, or off). Table 56 shows the available settings.

Table 56. SW Setting vs HVOUT Mode

SW	HVOUT Mode
0	On/Off
1	Switched

FR - Output Frequency Select (Switched Mode Only)
The FR parameter configures the output frequency of the HVOUT when the device is placed in switched mode. Table 57 shows the available settings.

Table 57. FR Setting vs HVOUT Output Frequency (Switched Mode only)

FR	Frequency
0	31.25 kHz
1	15.625 kHz

OD - Open Drain Output Mode Setting
The OD parameter is used to configure the output mode of the device. Table 58 shows the available settings.

Table 58. OD Setting vs HVOUT Output Mode

OD	HVOUT Output Mode
0	Charge-Pump Mode
1	Open-Drain Mode

DUTY[3:0] - Duty Cycle Selection (Switched Mode Only)
The Duty_Cycle[3:0] setting is used to choose between the sixteen programmable duty cycles for the HVOUT in switched mode. The available choices are shown in Table 59.

Table 59. HVOUT Switched Output Duty Cycle Settings

Duty_Cycle[3:0]	Duty Cycle%
0x0	6.25%
0x1	12.5%
0x2	18.75%
0x3	25%
0x4	31.25%
0x5	37.55%
0x6	43.75%
0x7	50.00%
0x8	56.25%
0x9	62.50%
0xA	68.75%
0xB	75.00%
0xC	81.25%
0xD	87.50%
0xE	93.75%
0xF	50.00%

Output Control Block Configuration Registers

Table 60. Output Control Block Configuration Register Summary

Register Address	Register Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Reconfiguration Details
0x5B	OCB_Config0	GPIO3_src[3:0]				GPIO2_src[2:0]				Master/Slave (LOAD_CFG_REG supported)
0x5C	OCB_Config1	HVOUT2_src[3:0]				HVOUT1_src[3:0]				
0x5D	OCB_Config2	HVOUT4_src[3:0]				HVOUT3_src[3:0]				
0x5E	OCB_Config3	X	X	IM_HCM_CTRL[2:0]		HI_HCM_CTRL[2:0]				
0x5F	OCB_Config4	X	X	V6_HCM_CTRL[2:0]		V5_HCM_CTRL[2:0]				
0x60	OCB_Config5	X	HI_T	H4i	H3i	H2i	H1i	G3i	G2i	

The Output Control Block (OCB) configuration registers can be updated over I^2C. The definition and function of these parameters is described in the Output Control Block section. There are 6 total configuration registers for the output control block. The settings in the six registers define the operation of the output control block output muxes, the hysteretic control muxes, and the dynamic threshold management. Registers 0x5B-0x5D define the out control muxes according to the format shown in Table 60.

OUTPUT_src[3:0] - Output Channel Source Signal Select

The output control source for each of the six OCB based outputs is defined by the four bit _src[3:0] code. The outputs selected by each code are shown in Table 61.

Table 61. Output Control Block – Output Source Signals

OUTPUT_src[3:0]	Source Signal
0x0	ASC-I/F
0x1	I^2C
0x2	GPIO5
0x3	GPIO6
0x4	RESERVED
0x5	GPIO8
0x6	GPIO9
0x7	GPIO10
0x8	HIMON_F
0x9	IMON1_F
0xA	VMON_4A
0xB	VMON_9A
0xC	HIMON_HCM
0xD	IMON1_HCM
0xE	VMON5_HCM
0xF	VMON6_HCM

HCM_CTRL - Hysteretic Mux Configuration for IMON1, HIMON, VMON6 and VMON5

Registers 0x5E and 0x5F define the control signal inputs for the 4 Hysteretic Control Muxes (HCM). The register format is shown in Table 60. The control signals selected by the _HCM_CTRL[2:0] code are shown in Table 62.

Table 62. Output Control Block – Hysteretic Control Mux Settings

HCM_CTRL[2:0]	ASC-I/F Source Signal
000	GPIO2
001	GPIO3
010	HVOUT1
011	HVOUT2
100	HVOUT3
101	HVOUT4
110	N/A
111	N/A

H4i / H3i / H2i / H1i/ G3i / G2i - Output Invert Control

The OCB outputs can be inverted with respect to their control signals. As shown in Table 60, the register 0x60 defines the programmable invert option for each of the outputs.The invert setting parameter is shown in Table 63.

Table 63. H4i … G2i Setting vs OCB Output Behavior

H4i / H3i / H2i / H1i/ G3i / G2i	OCB Output
0	Normal
1	Inverted

HI_T - HIMON_A Threshold Source

Register 0x60 also defines the programmable threshold source select for the HIMON circuit. The threshold source can be selected as either the configuration memory or the ASC-I/F, as shown in Table 64.

Table 64. HI_T Setting vs HIMONA Threshold Source

HI_T	HIMONA Threshold Source
0	Configuration Memory
1	ASC-I/F

GPIO Input Configuration Registers

Table 65. GPIO Input Configuration Register Summary

Register Address	Register Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Reconfiguration Details
0x62	GPIO_Config0	X	G4in	X	G3in	X	G2in	X	G1in	Master/Slave (LOAD_CFG_REG supported)
0x63	GPIO_Config1	X	G8in	X	X	X	G6in	X	G5in	
0x64	GPIO_Config2	X	X	X	X	X	G10in	X	G9in	

The GPIO pins can be configured as input or output. The GPIO configuration registers can be updated over I^2C. The registers at addresses 0x62, 0x63, and 0x64 include single configuration bit for each of the GPIO. When the device is configured as an input, the GPIO pin is put into a Hi-Z state and a weak pulldown is enabled. The input setting is described in Table 66. The input status can still be read at the pin, regardless of the Gxin setting. The format for registers 0x62, 0x63, and 0x64 is shown in Table 65.

Table 66. Gxin Setting vs GPIO Input Setting

Gxin	GPIO Setting
0	Output – Weak Pulldown Disabled
1	Input – Weak Pulldown Enabled

Write Protect and User Tag Configuration Register

Table 67. Write Protect and User Tag Configuration Register

Register Address	Register Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Reconfiguration Details
0x66	WRITEPROTECT_USERTAG	X	X	X	X	X	UT_EN	WP[1:0]		Read Only

The Write Protect and User Tag modes are defined by the bit settings in register 0x66 (shown in Table 67). This register cannot be written using the configuration register commands. It can only be overwritten in the EEPROM memory.

UT_EN - User Tag Enable

The UT_EN bit configures the device for either User Tag Memory mode or Fault Logging mode, as described in Table 68. The User Tag and Fault Log features are described in more details in the Fault Logging and User Tag Memory section

Table 68. UT_EN vs Fault Log / User Tag Mode

UT_EN	Fault Log / User Tag Mode
0	Fault Log Enabled / User Tag Disabled
1	User Tag Enabled / Fault Log Disabled

WP[1:0] - Write Protect Setting

The write protect setting bits are defined in Table 69. The write protect function is described in detail later in the I2C Write Protection section.

Table 69. Write Protect Settings

WP[1:0]	Write Protect Settings
00	No protection
01	No protection
10	Protection based on GPIO1 level
11	I^2C write disabled

User Electronic Signature (UES) Registers

Table 70. UES Memory Summary

Register Address	Register Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Reconfiguration Details
0x70	UES0	UES[7:0]								EEPROM Read Only
0x71	UES1	UES[15:8]								
0x72	UES2	UES[23:16]								
0x73	UES3	UES[31:24]								
0x74	UES4	UES[39:32]								
0x75	UES5	UES[47:40]								
0x76	UES6	UES[55:48]								
0x77	UES7	UES[63:56]								

The ASC includes a User Electronic Signature feature in the EEPROM memory of the device. This consists of 64 bits that can be configured by the user to store unique data such as ID codes, revision numbers, or inventory control data. The UES code can only be written and readout using the EEPROM memory access commands. The UES storage format is shown in Table 70.

Reserved Configuration Addresses

The configuration memory map includes several reserved addresses, which should not be read or written to. The reserved addresses are shown in Table 71 below.

Table 71. Reserved Configuration Addresses

Register Address	Register Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Reconfiguration Details
0x61	RESERVED	RESERVED								Do not read or write to these addresses
0x65	RESERVED	RESERVED								
0x67-0x6F	RESERVED	RESERVED								
0x78-0xFF	RESERVED	RESERVED								

Closed Loop Trim Register Access

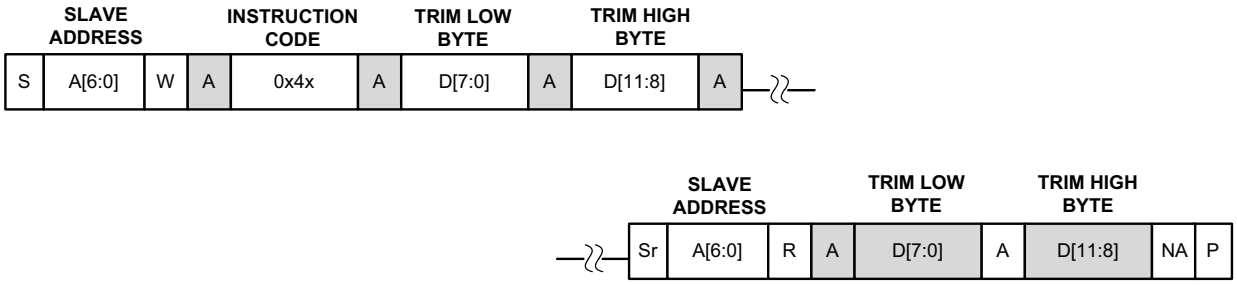
The Trim and Margin block provides for I²C access to write closed loop trim profile 0 target values for each Trim channel on-chip as shown in Table 72. The 12 bits of each closed trim setpoint register can be updated and read back atomically using the dedicated instructions below.

Table 72. Closed Loop Trim Access Instructions

Instruction Code	Instruction Name	Read/Write	Description
0x41	TRIM1_CLT_P0_SET	R/W	Update and readback of TRIM1 closed loop trim profile 0 setpoint register[11:0]
0x42	TRIM2_CLT_P0_SET	R/W	Update and readback of TRIM2 closed loop trim profile 0 setpoint register[11:0]
0x43	TRIM3_CLT_P0_SET	R/W	Update and readback of TRIM3 closed loop trim profile 0 setpoint register[11:0]
0x44	TRIM4_CLT_P0_SET	R/W	Update and readback of TRIM4 closed loop trim profile 0 setpoint register[11:0]

The format for these instructions is shown in Figure 50 below.

Figure 50. TRIMx_CLT_P0_SET - I²C Instruction Format



The new trim target is latched in the hardware at the completion of the write sequence, the LOAD_CFG_REG command is not needed.

In Closed Loop Trim mode, the Trim targets are 12-bit positive numbers where each bit corresponds to 2 mV. See Closed Loop Trim Configuration Registers for details on calculating the trim target voltage. In Manual mode, the DAC targets are 8-bit signed numbers (low byte) where each bit corresponds to 2.5 mV (see the Margin/Trim DAC Output Characteristics table) and the high byte is ignored. Both bytes have to be written for the new value to take effect.

Measurement and Control Register Access

The measurement and control section of the ASC is accessed via two different I²C instructions. The WRITE_MEAS_CTRL instruction is used to write the measurement selection for voltage and current measurements, the selection for reading the monitor status, and the control selection for the output control block. The READ_MEAS_CTRL is used to read the measurement result selected by the WRITE_MEAS_CTRL instruction. The instructions are used to access the register set shown in Table 73. The instructions use the register addresses in Table 73 and follow the format shown in Figure 51 for WRITE_MEAS_CTRL and Figure 52 for READ_MEAS_CTRL.

Figure 51. WRITE_MEAS_CTRL - I²C Instruction Format

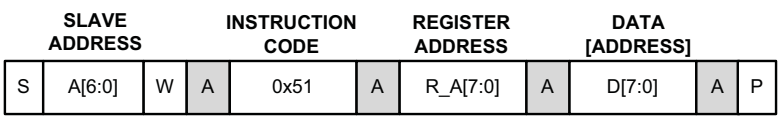
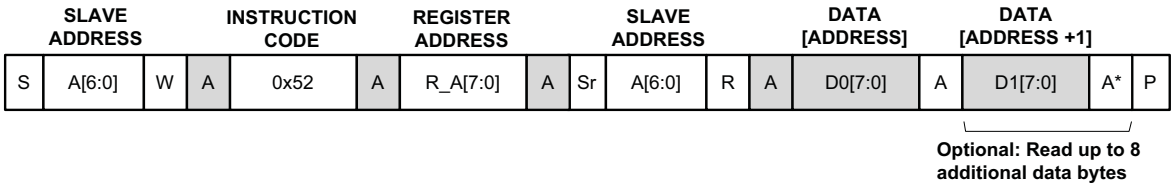


Figure 52. READ_MEAS_CTRL - I²C Instruction Format



* After final data byte read, master should NACK before issuing the STOP command

The measurement and control register address map is shown in Table 73. These register are used to read voltage and current measurements from the ADC, read temperature measurements from the TMON circuit, and control I/Os configured for I²C control. These registers should not be confused with the configuration registers, which are accessed with a different set of instructions and comprise a completely separate 8-bit address space.

Table 73. Measurement and Control Register Overview

Register Address	Register Name	Read/Write	Description	Value after POR
0x00	adc_mux	R/W	ADC Attenuator and SEL[4:0]	0000 0000
0x01	adc_value_low	R	ADC Result [4:0] and status	0000 0000
0x02	adc_value_high	R	ADC Result [12:5]	0000 0000
0x03	imon_average_ctrl	R/W	Average Control [3:0]	0000 0000
0x04	imon_average_select	R/W	IMON MUX[1:0]	0000 0000
0x05	imon_average_result_low	R	IMON moving average [7:0]	0000 0000
0x06	imon_average_result_high	R	IMON moving average [9:8]	0000 0000
0x07	monitor_select	R/W	Monitor Select[3:0]	0000 0000
0x08	monitor_ record	R	Monitor Record[7:0]	0000 0000
0x09-0x6F	RESERVED			
0x70	output_control_block	R/W	HVOUT1-4 and GPIO2-3 control	0000 0000
0x71-0x7F	RESERVED			
0x80	tmon_meas_1_high	R	TMON_1 Measurement [15:8]	1110 0000
0x81	tmon_meas_1_low	R	TMON_1 Measurement [7:5]	0000 0000
0x82	tmon_meas_2_high	R	TMON_2 Measurement [15:8]	1110 0000
0x83	tmon_meas_2_low	R	TMON_2 Measurement [7:5]	0000 0000
0x84	tmon_meas_int_high	R	TMON_int Measurement [15:8]	1110 0000
0x85	tmon_meas_int_low[1]	R	TMON_int Measurement [7:5]	0000 0000
0x86	tmon_stat_a[1]	R	TMON_A Status [2:0]	0000 0000
0x87	tmon_stat_b	R	TMON_B Status [2:0]	0000 0000
0x88-0xFF	RESERVED			

1. Any READ_MEAS_CTRL command which reads addresses 0x85 or 0x86 as the final data byte must be followed by an additional READ_MEAS_CTRL command of a different address (such as 0x70) before issuing any other I²C read command.

The registers shown in Figure 53 are provided for interfacing to the ADC.

Figure 53. ADC Registers

0x00 – ADC_MUX (Read/Write)

ATTEN	0	0	SEL4	SEL3	SEL2	SEL1	SEL0
b7	b6	b5	b4	b3	b2	b1	b0

0x01 – ADC_VALUE_LOW (Read)

D4	D3	D2	D1	D0	Pending	Active	Done
b7	b6	b5	b4	b3	b2	b1	b0

0x02 – ADC_VALUE_HIGH (Read)

D12	D11	D10	D9	D8	D7	D6	D5
b7	b6	b5	b4	b3	b2	b1	b0

To perform an A/D conversion, one must set the input attenuator and channel selector. For VMON input voltage conversions, two input ranges may be set using the attenuator, 0-2.048 V and 0-5.9 V. For conversion of the HVMON input voltage, the available attenuator ranges are 0-8.192 V and 0-13.2 V. These settings are shown in Table 74.

Table 74. ADC Input Attenuator Control

ATTEN(ADC_MUX.b7)	VMON1-VMON9		HVMON	
	Resolution	Full Scale Range	Resolution	Full Scale Range
0	2 mV	0-2.048 V	8mV	0-8.192
1	6 mV	0-5.9 V	16mV	0-13.2

The input selector may be set to monitor any of the VMON input voltages, the VCCA supply voltage, or the IMON differential voltages. The selectable input channels are shown in Table 75. Do not read or write to ADC_MUX selections not shown in the table.

Table 75. ADC Input Selection

SEL[4:0] (ADC_MUX Selection)	Input Channel
0x00	VMON1
0x01	VMON2
0x02	VMON3
0x03	VMON4
0x04	VMON5
0x05	VMON6
0x06	VMON7
0x07	VMON8
0x08	VMON9
0x09	HVMON
0x0C	VCCA
0x10	IMON1
0x13	HIMON

Writing a value to the ADC_MUX register using the WRITE_MEAS_CTRL instruction to set the input attenuator and selector will automatically initiate a conversion. The PENDING bit will be set to 1 when a conversion is requested but not yet active. The ACTIVE bit will be set to 1 when the requested conversion is the active conversion. When the conversion is in process, the DONE bit (ADC_VALUE_LOW.b0) will be reset to 0. When the conversion is complete, this bit will be set to 1. When the result from the ADC is copied to the I^2C registers, the ACTIVE bit is reset and the result may be read out by performing two I^2C read operations using the READ_MEAS_CTRL instruction; one for ADC_VALUE_LOW, and one for ADC_VALUE_HIGH. It is recommended that the I^2C master load a second conversion instruction only after the completion of the current conversion operation (Waiting for the DONE bit to be set to 1). The example flow below shows the necessary instructions to complete an ADC read.

Voltage Monitor ADC Readout Over I^2C

1. Perform an I^2C Write Operation with Instruction 0x51, Register Address 0x00, and the ADC Channel and Attenuator setting from Table 75. This will initiate the ADC conversion.

2. Perform an I^2C Read Operation with Instruction 0x52, Register Address 0x01 to check that the DONE bit is set and the ACTIVE bit is reset. Repeat if Register 0x01, bit 0 is not set to 1 and bit 1 is not reset to 0. Read ADC Data[4:0] for ADC low byte.

3. Perform an I^2C Read Operation with Instruction 0x052, Register Address 0x02 to read ADC Data [12:5] for ADC high byte

The Current Monitor (IMON1 and HIMON) averaging measurement hardware is also accessed through the I^2C interface. The IMON1/HIMON averaging is controlled by dedicated hardware and is managed separately from the single conversion IMON1/HIMON access through the ADC registers. The IMON1/HIMON averaging registers are shown in Figure 54.

Figure 54. IMON Average Control Registers

0x03 – IMON_AVG_CTRL (Read/Write)

0	**0**	**HIMON AVG_EN**	**0**	**0**	**IMON1 AVG_EN**	**SMPL_ INT_1**	**SMPL_ INT_0**
b7	b6	b5	b4	b3	b2	b1	b0

0x04 – IMON_AVG_SELECT (Read/Write)

0	**0**	**0**	**0**	**0**	**0**	**SEL1**	**SEL0**
b7	b6	b5	b4	b3	b2	b1	b0

0x05 – IMON_AVG_RESULT_LOW (Read)

D7	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
b7	b6	b5	b4	b3	b2	b1	b0

0x06 – IMON_AVG_RESULT_HIGH (Read)

0	**0**	**0**	**0**	**0**	**0**	**D9**	**D8**
b7	b6	b5	b4	b3	b2	b1	b0

To perform a moving average read, first enable the moving average for the given channel in the IMON_AVG_CTRL register. When the averaging is enabled the averaging hardware will begin to compute a binary exponential weighted moving average, as shown below:

$$CurrentAve[x] = \frac{CurrentMeas[x]}{8} + CurrentAve[x-1] \times \frac{7}{8}$$

The IMON_AVG_CTRL register is used to enable the averaging operation and set the sample period of the averaging. The HIMON and IMON1 averaging operations are enabled independently by setting the HIMON_AVG_EN and IMON1_AVG_EN respectively. Setting these bits to 1 enables the 1/8 moving averaging for that input channel. The SMPL_PER[1:0] is used to select the sampling interval from the values shown in Table 76.

Table 76. IMON Average Sample Interval Values

SMPL_INT[1:0]	Sample Interval
00	1 ms
01	2 ms
10	4 ms
11	8 ms

Once the averaging is enabled, the IMON_AVG_SELECT register is used to select between the HIMON and IMON1 average values for readout (see Table 77).

Table 77. Selected IMON Average Readout Channel

SELECT[1:0]	Channel
00	IMON1
01	Not Used
10	Not Used
11	HIMON

After writing to the IMON_AVG_SELECT register, the selected channels 10-bit moving average can be read out of the IMON_AVG_RESULT_LOW and IMON_AVG_RESULT_HIGH registers. The value in the IMON_AVG_SELECT register will remain active until overwritten by another I²C instruction (or power on reset). The updated 10-bit moving average for the previously selected channel can be readout from the result registers without re-writing the SELECT register.

Current Monitor Moving Average Readout Over I²C

1. Perform an I²C Write Operation with Instruction 0x51, Register Address 0x03, and enable the HIMON averaging and configure the sampling period. This will enable the HIMON averaging. See Table 76 and Table 77.
2. Moving average results are available immediately based on limited samples. Waiting additional sample periods ensures that the averaging filter has time to settle. After enough time has elapsed, perform an I²C Write Operation with Instruction 0x51, Register Address 0x04, and select the HIMON channel for readout.
3. Perform an I²C Read Operation with Instruction 0x052, Register Address 0x05 to read IMON_AVG_RESULT_LOW[7:0] for HIMON low byte.
4. Perform an I²C Read Operation with Instruction 0x052, Register Address 0x06 to read IMON_AVG_RESULT_HIGH[9:8] for HIMON high byte.

The status of the voltage, current, and temperature monitor alarms, as well as the GPIO input status, can be read out over I²C. Access to the alarm signals is controlled by the MONITOR_SELECT register. To read out the alarm or GPIO status over I²C, write the applicable selection to the MONITOR_SELECT register as shown in Table 78. After

writing the corresponding value to the MONITOR_SELECT, you can read the monitor signal status from the MONITOR_RECORD register. These registers are shown in Figure 55.

The MONITOR_SELECT also includes a valid bit. This bit is set to 1 once the MONITOR_RECORD register includes the monitor signals selected in the MONITOR_SELECT register. The monitor signals are refreshed much faster than the I²C access time, so normally the valid bit will read as 1. If the device is in safe state and ASC-I/F communication has not started properly, the valid bit will read as 0. The MONITOR_RECORD register will continue to include the latest status of the specified alarm signals, as specified in the MONITOR_SELECT register.

Figure 55. Monitor Signal Access Registers

0x07 – MONITOR_SELECT (Read/Write)

Valid	0	0	0	SEL3	SEL2	SEL1	SEL0
b7	b6	b5	b4	b3	b2	b1	b0

0x08 – MONITOR_RECORD (Read)

D7	D6	D5	D4	D3	D2	D1	D0
b7	b6	b5	b4	b3	b2	b1	b0

The register map of the monitor data is shown in Table 78 below. An example for how to read out selected data is shown after the table.

Table 78. MONITOR_RECORD Byte Selection

MONITOR_SELECT [3:0]	MONITOR_RECORD[7:0]							
	D7	D6	D5	D4	D3	D2	D1	D0
0x0	0	Fault_Log_ Full	Fault_Log_Busy	0	0	1	X	1
0x1	AGOOD	GPIO10	GPIO9	GPIO8	X	GPIO6	GPIO5	GPIO4
0x2	GPIO3	GPIO2	GPIO1	HVOUT4	HVOUT3	HVOUT2	HVOUT1	HIMON_b
0x3	HIMON_a	IMON1_b	IMON1_a	HVMON_b	HVMON_a	VMON9_b	VMON9_a	VMON8_b
0x4	VMON8_a	VMON7_b	VMON7_a	VMON6_b	VMON6_a	VMON5_b	VMON5_a	VMON4_b
0x5	VMON4_a	VMON3_b	VMON3_a	VMON2_b	VMON2_a	VMON1_b	VMON1_a	TMON2_b
0x6	TMON2_a	TMON1_b	TMON1_a	TMONint_b	TMONint_a	1	0	1

Monitor Record Readout Over I²C

1. Perform an I²C Write Operation with Instruction 0x51, Register Address 0x07, and write the selected data byte for readout. This will populate the MONITOR_RECORD register, Address 0x08, with the latest sampled data of the chosen bits as described in Table 78.

2. Perform an I²C Read Operation with Instruction 0x52, Register Address 0x08 to read the selected Monitor Record.

The output signals for HVOUT1-4 and GPIO2-3 can be controlled via I²C instruction, dependent on their Output Control Block configuration. The OUTPUT_CONTROL_BLOCK register (shown in Figure 56) is used to set the I²C control input to the Output Control Block. Outputs which are not configured for I²C control will ignore the setting in the I²C register. See the Output Control Block section for more details.

Figure 56. Output Control Block Register

0x70 – OUTPUT_CONTROL_BLOCK (Read/Write)

0	0	HVOUT4	HVOUT3	HVOUT2	HVOUT1	GPIO3	GPIO2
b7	b6	b5	b4	b3	b2	b1	b0

Several registers are provided for accessing the temperature monitor measurements. Each temperature monitor channel has a TMON_MEAS_CHx_LO and TMON_MEAS_CHx_HI register for accessing the 11-bit temperature reading (shown in Figure 57). Provided the temperature monitor is enabled, these registers are updated automatically with the latest temperature reading. The update rate of the temperature reading is dependent on the number of channels enabled (see the Temperature Monitors section for more details).

The temperature measurement hardware will latch its latest reading after the TMON_MEAS_CHx_HIGH byte is read over I^2C. This will ensure that the corresponding TMON_MEAS_CHx_LOW byte is from the same measurement as the TMON_MEAS_CHx_HIGH byte. The HIGH byte should be read first in the I^2C transaction, followed by the LOW byte. This will ensure the two bytes are from the same measurement reading.

Figure 57. Temperature Monitor Measurement Registers

0x80 – TMON_MEAS_CH1_HIGH (Read)

D10	D9	D8	D7	D6	D5	D4	D3
b7	b6	b5	b4	b3	b2	b1	b0

0x81 – TMON_MEAS_CH1_LOW (Read)

D2	D1	D0	0	0	0	0	0
b7	b6	b5	b4	b3	b2	b1	b0

0x82 – TMON_MEAS_CH2_HIGH (Read)

D10	D9	D8	D7	D6	D5	D4	D3
b7	b6	b5	b4	b3	b2	b1	b0

0x83 – TMON_MEAS_CH2_LOW (Read)

D2	D1	D0	0	0	0	0	0
b7	b6	b5	b4	b3	b2	b1	b0

0x84 – TMON_MEAS_INT_HIGH (Read)

D10	D9	D8	D7	D6	D5	D4	D3
b7	b6	b5	b4	b3	b2	b1	b0

0x85 – TMON_MEAS_INT_LOW (Read)

D2	D1	D0	0	0	0	0	0
b7	b6	b5	b4	b3	b2	b1	b0

The format of the 11-bit temperature measurement is shown in Table 79 along with some example values. The measurement format is 2-complement, with bit D10 used as the sign bit. The measurement resolution is 0.25 °C per bit. ***Temperature monitor circuits which are reset or disabled will always readout –64°C.***

Any READ_MEAS_CTRL command which reads addresses 0x85 or 0x86 as the final data byte must be followed by an additional READ_MEAS_CTRL command of a different address (such as 0x70) before issuing any other I2C read command.

Table 79. Temperature Measurement Data Format

D[10:0]	Measured Temperature (°C)
0x3FF	160.00*
0x26C	155.00
0x26B	154.75

0x002	0.50
0x001	0.25
0x000	0.00
0x7FF	–0.25
0x7FE	–0.50

0x701	–63.75
0x700	–64.00*

Note: Measurements above 160 °C will limit to 0x3FF. There is no lower limit, although the TMON accuracy is unguaranteed below –64 °C.

The temperature monitor alarm signals are also available to be read out over I^2C directly. The A comparator alarm signals for each temperature monitor can be read out from the TMON_STAT_A register while the B comparator alarm signals can be read out from the TMON_STAT_B. The register format is shown in Figure 58.

Figure 58. Temperature Monitor Status Registers

0x86 – TMON_STAT_A (Read)

0	0	0	0	0	TMONINT_A	TMON2_A	TMON1_A
b7	b6	b5	b4	b3	b2	b1	b0

0x87 – TMON_STAT_B (Read)

0	0	0	0	0	TMONINT_B	TMON2_B	TMON1_B
b7	b6	b5	b4	b3	b2	b1	b0

Any READ_MEAS_CTRL command which reads addresses 0x85 or 0x86 as the final data byte must be followed by an additional READ_MEAS_CTRL command of a different address (such as 0x70) before issuing any other I^2C read command.

User Tag Memory Access

The I²C interface is used to access the User Tag memory feature of the ASC. The User Tag memory block consists of a 7 byte User Tag register, a programming hardware block, and a 16 row x 7 byte EEPROM memory. The User Tag memory block architecture is shown in Figure 59, along with the I²C access instructions.

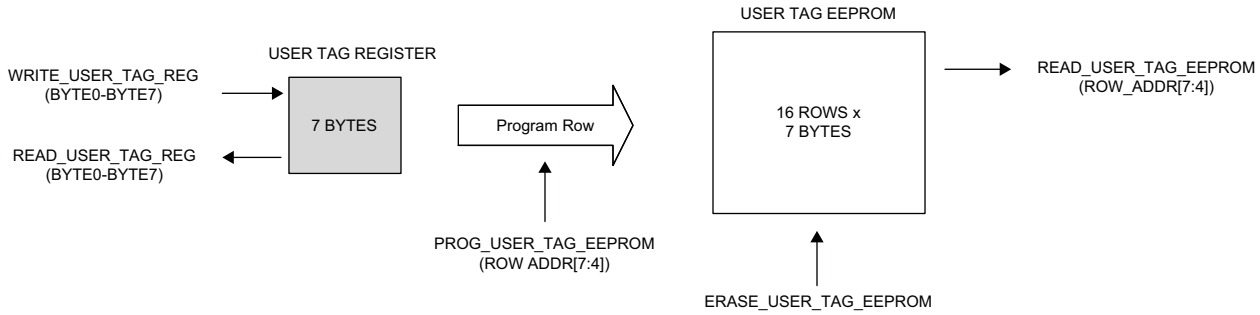
The User Tag feature cannot be used when the ASC Fault Log is enabled. These features use the same memory array and only one of the two features can be enabled at a given time. The User Tag instructions shown below are only applicable for accessing the memory in User Tag mode. Access to the memory in Fault Log Mode should follow the Fault Log instructions detailed in the next section.

Table 80 shows the User Tag access instructions. The instructions are used to access either the User Tag register or the User Tag EEPROM memory block. Accessing the User Tag memory requires that the ASC is placed into programming mode (see the ENABLE_PROG instruction). The format for each instruction is in the section that follows.

Table 80. User Tag Memory Access Instructions

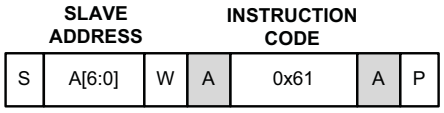
Instruction Code	Instruction Name	Read/Write	Description
0x61	ERASE_USER_TAG_EEPROM	W	Erase the User Tag EEPROM array
0x62	WRITE_USER_TAG_REG	W	Write to the User Tag data register
0x63	READ_USER_TAG_REG	R	Read out the contents of the User Tag data register
0x64	PROG_USER_TAG_EEPROM	W	Program the selected row of EEPROM bits with the tag data register
0x65	READ_USER_TAG_EEPROM	R	Read out the selected row of User Tag EEPROM bits

Figure 59. User Tag Memory Architecture with I²C Instruction Access



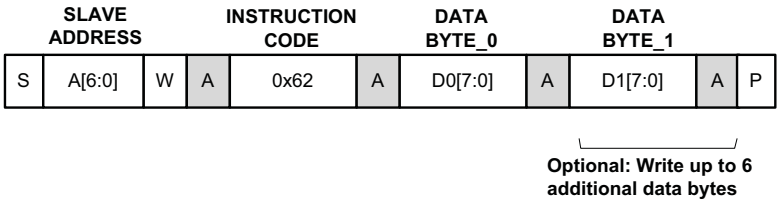
The ERASE_USER_TAG_EEPROM instruction is used to erase the entire User Tag EEPROM array. The User Tag can only be erased as a full block. A row must be erased prior to programming it. The instruction format for ERASE_USER_TAG_EEPROM is shown in Figure 60.

Figure 60. ERASE_USER_TAG_EEPROM - I²C Instruction Format



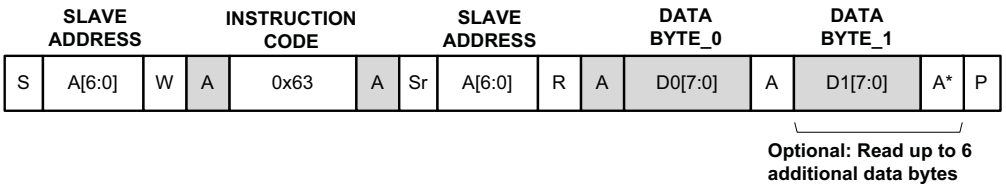
The User Tag EEPROM is programmed in a two-step process. First the data which is to be programmed is written into the User Tag register space, as shown in Figure 61. Up to 7 bytes (known as a data “row”) can be written into the User Tag register space in a single write transaction. The WRITE_USER_TAG_REG instruction is used for writing the User Tag register space.

Figure 61. WRITE_USER_TAG_REG - I²C Instruction Format



The READ_USER_TAG_REG instruction is used to read out the User Tag register bytes as shown in Figure 62. Up to 7 bytes can be read out from the User Tag register space in a single read transaction. The bytes are read back in order from byte 0 to byte 6.

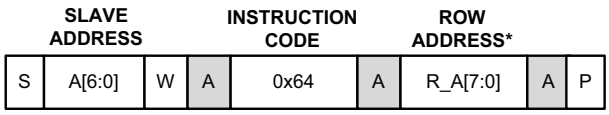
Figure 62. READ_USER_TAG_REG - I²C Instruction Format



* After final data byte read, master should NACK before issuing the STOP command

The second step of the User Tag programming is writing to the EEPROM from the User Tag registers. This is accomplished using the PROGRAM_USER_TAG_EEPROM instruction. The data in the User Tag registers is copied by the programming block to the EEPROM row specified by the R_A[7:4] bits in the data byte as shown in Figure 63. The 4-bit code corresponds to Row 0 to Row 15 in the User Tag EEPROM memory block, as shown in the User Tag block diagram in Figure 59.

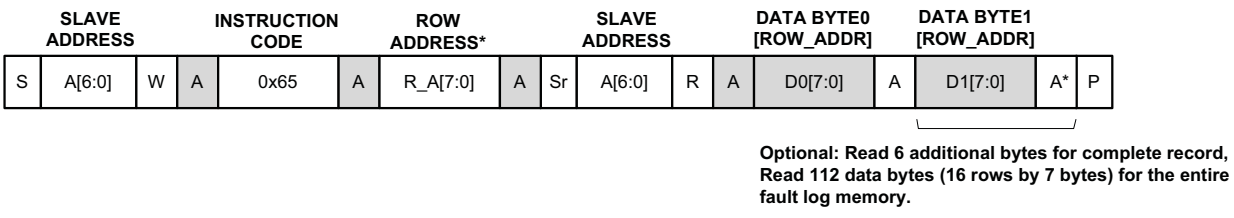
Figure 63. PROG_USER_TAG_EEPROM - I²C Instruction Format



Note: The Row Address R_A[7:0] contains the 4-bit address code in bits [7:4]. Bits [3:0] are always zero.

The READ_USER_TAG_EEPROM I²C instruction provides the mechanism to readback data stored in the User Tag memory. The READ_USER_TAG_EEPROM is a two step read transaction operation shown in Figure 64. In the first step, a write transaction is performed with the 0x65 instruction, and a 4-bit address code [7:4]. The address code corresponds to Row 0 to Row 15 as shown in the User Tag block diagram in Figure 59. In the second step, the row can be read out in 7 bytes, from byte 0 to byte 6, using a read transaction. The row address will auto-increment to support reading multiple rows in a single transaction. This means a single transaction can support reading the entire user tag array, if the starting address of Row 0 is used. A stop condition will complete the read transaction, this can be issued after any number of rows and bytes have been read.

Figure 64. READ_USER_TAG_EEPROM - I²C Instruction Format



* After final data byte read, master should NACK before issuing the STOP command

User Tag Access Example

This example describes the steps necessary to program 7 Data Bytes to Row 4 of the EEPROM array:

1. Perform an I^2C Write with the ENABLE_PROG instruction (0x04), with the 2-byte key code 0xE53D. This will place the chip in programming mode, a required step for User Tag access.

2. (Optional) Perform an I^2C Write with the ERASE_USER_TAG_EEPROM instruction (0x61). This is only required if data has already been written to Row 4.

3. Perform an I^2C Write with the WRITE_USER_TAG_REG instruction (0x62), with the 7 data bytes to be written.

4. Optional) Perform an I^2C Read with the READ_USER_TAG_REG instruction (0x63) to confirm that the 7 Data Bytes were properly written the USER_TAG_REGISTER.

5. Perform an I^2C Write with the PROGRAM_USER_TAG_EEPROM instruction (0x64), with an R_A[0x40], row address 4. This copies the data from the USER_TAG_REG into Row 4 of the USER_TAG_EEPROM array.

6. (Optional) Perform an I^2C Read with the READ_USER_TAG_EEPROM instruction (0x65), with an address of 0x04. You can use this operation to verify the EEPROM programming.

7. Perform an I^2C Write with the ENABLE_USER instruction (0x05). This operation will place the ASC back in User Mode, in order to prevent accidental programming access.

Fault Log Memory Access

The ASC includes a fault logging block. The fault logging block consists of fault recording hardware, a volatile memory register which holds one 7 byte fault log record, a programming block, and a 16 row by 7 byte EEPROM memory for fault record storage. The fault logging block, including I^2C access instructions is shown in Figure 65. The fault logging block is described in detail in the fault log section.

The ASC Fault Log cannot be used when the user tag feature is enabled. These features use the same memory array and only one of the two features can be enabled at a given time. The Fault Log instructions shown below are only applicable for accessing the memory in Fault Log mode. Access to the memory in User Tag Mode should follow the User Tag instructions detailed in the previous section.

Table 81 shows the Fault Log access instructions. The instructions are used to read out or erase either the fault log register or the fault log EEPROM memory block. The format for each instruction is in the section that follows.

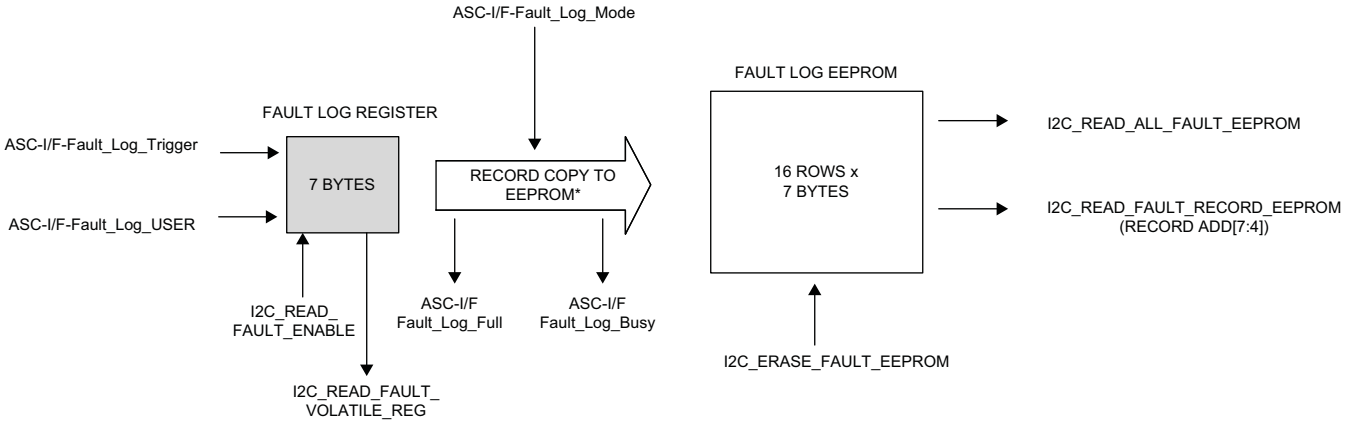
Writing fault logs to the EEPROM memory is triggered by the ASC-I/F. Faults can be recorded in either the fault logging register or in the EEPROM (for more details see the Fault Log section). The fault log recording hardware has priority access to the fault log EEPROM memory, and needs to be disabled via the READ_FAULT_ENABLE instruction prior to accessing the fault log via I^2C. The fault log recording process also takes precedence over the reconfiguration/programming of the device. If a fault log record process is active, the device will reject reconfiguration requests. You can avoid that scenario by executing the READ_FAULT_ENABLE instruction prior to starting the reconfiguration process.

The fault log instructions are summarized in Table 81 below, with individual instruction details in the following section.

Table 81. Fault Log Access Instructions

Instruction Code	Instruction Name	Read/Write	Description
0x71	ERASE_FAULT_EEPROM	W	Erase the entire fault log memory
0x73	READ_FAULT_VOLATILE_REG	R	Read the fault log volatile register contents
0x74	READ_FAULT_ENABLE	R/W	Disables the fault log recording hardware and enables the ERASE and READ instructions
0x75	READ_FAULT_RECORD_EEPROM	R	Read out the selected record of Fault Log Array EEPROM bits
0x76	READ_ALL_FAULT_EEPROM	R	Reads out all records of the Fault Log EEPROM Array

Figure 65. Fault Log Memory Block with I²C Access Instructions



* - Fault Log record programming number is automatically incremented each time a fault log is recorded in the memory. The next programming row is recorded in the status register, accessed by the READ_STATUS I2C command.

The ERASE_FAULT_EEPROM instruction is used to erase the entire fault log EEPROM record storage. The ASC must be in programming mode in order to execute this instruction (See the PROGRAM_MODE instruction). The READ_FAULT_ENABLE instruction must also have been sent to the ASC in order to disable recording of new faults. The format for the ERASE_FAULT_EEPROM is shown in Figure 66 below.

Figure 66. ERASE_FAULT_EEPROM - I²C Instruction Format

	SLAVE ADDRESS			INSTRUCTION CODE		
S	A[6:0]	W	A	0x71	A	P

The READ_FAULT_VOLATILE_REG instruction is used to read back the contents of the fault logging register. The 7 bytes are read back in order from Byte 0 to Byte 7. The READ_FAULT_ENABLE instruction must have been sent to the ASC in order to disable recording of new faults, prior to executing a READ_FAULT_VOLATILE_REG instruction. The format is shown in Figure 67.

Figure 67. READ_FAULT_VOLATILE_REG - I²C Instruction Format

	SLAVE ADDRESS			INSTRUCTION CODE		SLAVE ADDRESS			DATA BYTE_0		DATA BYTE_1		
S	A[6:0]	W	A	0x73	A	Sr	A[6:0]	R	A	D0[7:0]	A	D1[7:0]	A*	P

Optional: Read up to 6 additional data bytes

* After final data byte read, master should NACK before issuing the STOP command

The READ_FAULT_ENABLE instruction is used to disable the fault log recording hardware, and to enable the readback and erase of the fault logging memory block and fault log register. The READ_FAULT_ENABLE instruction is a write instruction with readback. The first transaction is a write transaction, as shown in Figure 68 below. The Address Byte with W=0 is sent, followed by the 0x74 instruction byte, followed by a keycode value of 0xAC. This key code is required to enable reading out or erasing of faults and disable fault recording. Sending any other keycode will disable reading and enable fault recording. Sending an incorrect keycode is the mechanism used to re-enable fault log recording without resetting the device.

The second transaction is a read transaction, with the Address Byte with R=1 sent, followed by a readback of the Fault Log Status Register. The Fault Log status register is described in Figure 69.

Figure 68. READ_FAULT_ENABLE - I²C Instruction Format

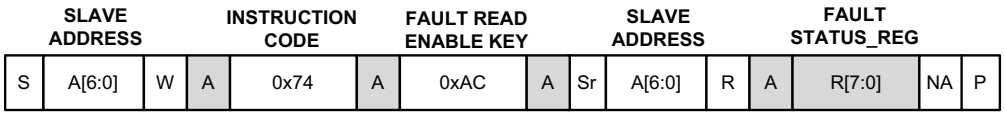


Figure 69. Fault Log Status Register

Fault Log Status Register (Read Only)

REQ[1]	REQ[0]	EN[1]	EN[0]	0	0	0	0
b7	b6	b5	b4	b3	b2	b1	b0

The Fault Log status register is a read-only register which indicates the status of the fault log hardware. The register bits indicate whether reading of fault logs is enabled or disabled. The possible readout combinations of the register are shown in Table 82.

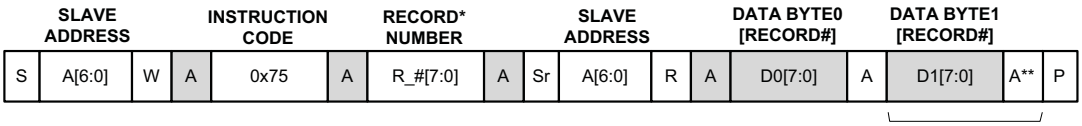
Table 82. Fault Log Status Details

REQ[1]	REQ[0]	EN[1]	EN[0]	Fault Log Status
0	0	0	0	ASC device is in safe state, or ERASEFAULT operation is active
0	0	1	1	Fault Logging is active, reading faults is disabled
1	1	0	0	Future fault logging is disabled and fault log read enable is requested. Reading fault logs will be enabled pending completion of in progress fault log recording or other EEPROM erase/program operation on-chip
1	1	1	1	Reading of fault logs via I²C is now enabled. Fault log recording is disabled
All other values				Invalid reading

The READ_FAULT_RECORD_EEPROM instruction provides the mechanism to readback fault log records stored in the EEPROM memory. The READ_FAULT_RECORD_EEPROM is a two-step read transaction instruction, as shown in Figure 70. In the first step, a write transaction is performed with the 0x75 instruction, and a 4-bit address code [7:4]. The address code corresponds to fault log record 0 to 15, as shown in the fault log block diagram in Figure 69. In the second step, the fault log record can be read out in 7 bytes, from byte 0 to byte 6, using a read transaction. The record address will auto-increment to support reading multiple records in a single transaction. A stop condition will complete the read transaction, this can be issued after any number of rows and bytes have been read.The fault record is organized according to Table 14, in the Fault Logging and User Tag Memory section. This means a single transaction can support reading the all 15 fault log memory records, if the starting address of Record 0 is used.

The READ_FAULT_RECORD_EEPROM instruction will be ignored if the READ_FAULT_ENABLE instruction has not been used to disable active fault recording.

Figure 70. READ_FAULT_RECORD_EEPROM - I²C Instruction Format

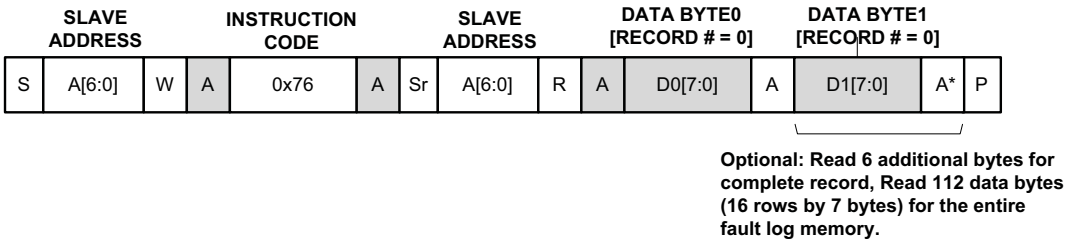


Optional: Read 6 additional bytes for complete record, Read 112 data bytes (16 rows by 7 bytes) for the entire fault log memory, depending on starting row

*The Record Number R_#[7:0] contains the 4-bit record number code in bits [7:4]. Bits [3:0] are always zero.
** After final data byte read, master should NACK before issuing the STOP command

The READ_ALL_FAULT_EEPROM is similar to the READ_FAULT_RECORD_EEPROM instruction and provides an alternative mechanism for reading back fault log records stored in EEPROM memory. The READ_ALL_FAULT_EEPROM instruction always starts the readback at Record 0 and does not support requesting an individual fault log record request. During the read transaction, shown in step 2 of Figure 71, fault log row 0 can be read out in 7 bytes. The row address will auto-increment to allow reading out the entire fault log array in a single transaction. A stop condition will complete the read transaction, this can be issued after any number of rows and bytes have been read. The READ_ALL_FAULT_EEPROM instruction will be ignored if the READ_FAULT_ENABLE instruction has not been used to disable active fault recording.

Figure 71. READ_ALL_FAULT_EEPROM - I²C Instruction Format



* After final data byte read, master should NACK before issuing the STOP command

Fault Log Memory Readout Example

This example describes the steps necessary to read the fault log EEPROM array over I²C. Note that the device must be in fault logging mode or it will not respond to these instructions.

1. Perform an I²C Write with the READ_FAULT_ENABLE instruction (0x74), with the 1-byte key code 0xAC. Complete the READ_FAULT_ENABLE operation by reading back the Fault Log Status Register. Repeat this operation until the Fault Log Status register reads as 0xF0 (Fault Log Reading is Enabled). When reading is enabled, fault log recording will be disabled.

2. Perform an I²C Read with the READ_STATUS instruction (0x03). The ASC_STATUS_REGISTER_LO[7:4] bits (FAULT_CNT[3:0]) are equal to the number of fault log records which have been written to the EEPROM memory array by the recording hardware. ASC_STATUS_REGISTER_LO[3] (FAULT_LOG_FULL) is set to 1 when all sixteen records have been used for fault logging.

3. Perform an I²C READ with the READ_ALL_FAULT_EEPROM instruction (0x76). Read out the fault logs starting at Record 0, Byte 0. Continue reading data until you reach the last populated fault record (given by FAULT_CNT[3:0] in step 3), byte 6.

4. Perform an I²C Write with the READ_FAULT_ENABLE instruction (0x74), with any keycode besides 0xAC. This disables fault log I²C access and returns the device to fault log recording mode. Repeat this operation until the Fault Log Status register reads as 0x00 (Fault Log Recording is Enabled).

Fault Log Memory Erase Example

This example describes the steps necessary to erase the fault log EEPROM array. Note that the device must be in fault logging mode or it will not respond to these instructions.

1. Perform an I²C Write with the ENABLE_PROGRAM instruction (0x04), with the 2-byte key code 0xE53D. This will place the chip in programming mode, a required step to erase the fault log EEPROM array.

2. Perform an I²C Write with the READ_FAULT_ENABLE instruction (0x74), with the 1-byte key code 0xAC. Complete the READ_FAULT_ENABLE operation by reading back the Fault Log Status Register. Repeat this operation until the Fault Log Status register reads as 0xF0 (Fault Log Reading is Enabled).

3. Perform an I²C Write with the ERASE_FAULT_EEPROM instruction (0x71). This will erase the fault log EEPROM memory.

4. Perform an I²C Write with the READ_FAULT_ENABLE instruction (0x74), with any keycode besides 0xAC. This disables fault log I²C access and return the device to fault log recording mode. Repeat this operation until the Fault Log Status register reads as 0x00 (Fault Log Recording is Enabled).

5. Perform an I²C Write with the ENABLE_USER instruction (0x05). This operation places the ASC back in User Mode in order to prevent accidental programming access.

I²C Write Protection

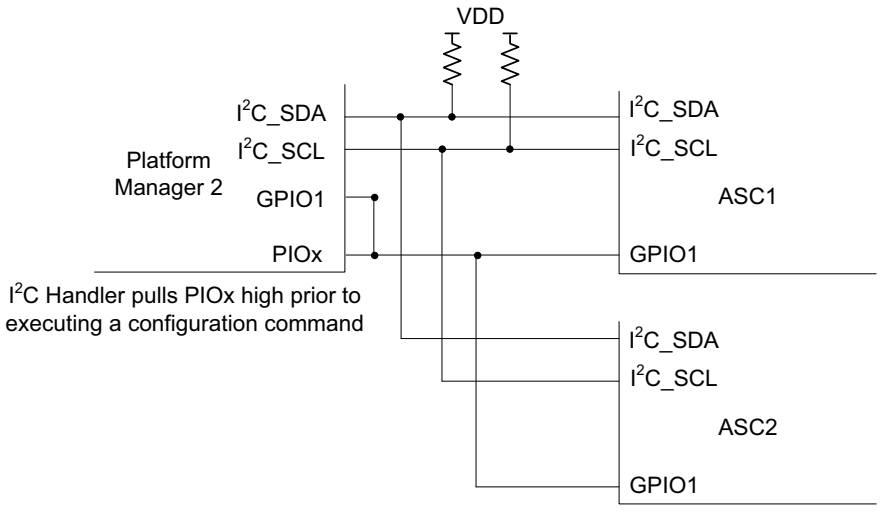
The ASC includes multiple protection mechanisms to prohibit accidental or incorrect access to the active device configuration. The active device configuration access protections are set during the initial programming of the device (also described in Table 69). There are three possible protection modes:

- I²C Configuration Write Enabled – The ASC configuration parameters can be freely overwritten by I²C commands
- I²C Configuration Write Disabled – The ASC configuration parameters cannot be overwritten by I²C instructions
- I²C Configuration Write Controlled by GPIO1 Pin State – The ASC configuration parameters can only be overwritten by I²C instructions when GPIO1 is pulled high by an external device (FPGA or Microcontroller)

These protection modes control the configuration access by the following I²C instructions: WRITE_CFG_REG, WRITE_CFG_REG_wMASK, and TRIMx_CLT_P0_SET. This protection does not prevent EEPROM access instructions. EEPROM access is protected by the ENABLE_PROG_MODE instruction and instruction key.

Figure 72 shows the typical configuration for working in the “Configuration Write Controlled by GPIO1 pin State” protection mode.

Figure 72. I²C Write Protect by GPIO1



The ASC device will still provide ACK bits in I²C write transmissions, even when the configuration write is disabled (by either the GPIO1 state or configuration setting). Even though the device presents ACK bits, the configuration memory will not be overwritten.

When using Write Protect by GPIO1 with WRITE_CFG_REG or WRITE_CFG_REG_wMASK, the GPIO1 signal should be asserted before transmission of the 7-bit slave address. It should be de-asserted after the STOP or RESTART signaling that marks the end of the write access. For TRIMx_CLT_P0_SET access, the GPIO1 signal

should be asserted before transmission of the 7-bit slave address of the write phase. It should be de-asserted during or after the transmission of the slave address at the start of the readback phase (See the Closed Loop Trim Register Access section for more details).

Pin Descriptions

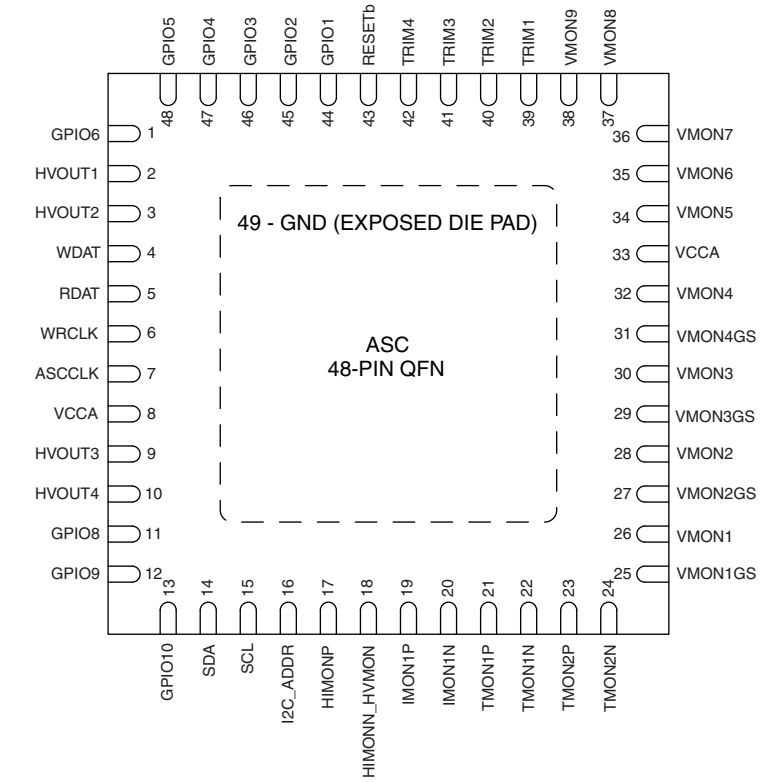
Pin Function	48-Pin QFN	Pin Type	Description
VMON1	26	Analog Input	Voltage Monitor Input
VMON1GS	25	Analog Input	Voltage Monitor Input Ground Sense
VMON2	28	Analog Input	Voltage Monitor Input
VMON2GS	27	Analog Input	Voltage Monitor Input Ground Sense
VMON3	30	Analog Input	Voltage Monitor Input
VMON3GS	29	Analog Input	Voltage Monitor Input Ground Sense
VMON4	32	Analog Input	Voltage Monitor Input
VMON4GS	31	Analog Input	Voltage Monitor Input Ground Sense
VMON5	34	Analog Input	Voltage Monitor Input
VMON6	35	Analog Input	Voltage Monitor Input
VMON7	36	Analog Input	Voltage Monitor Input
VMON8	37	Analog Input	Voltage Monitor Input
VMON9	38	Analog Input	Voltage Monitor Input
HIMONP[3]	17	Analog Input	12V Current Monitor Input Source
HIMONN_HVMON	18	Analog Input	12V Current Monitor Input Return / Voltage Monitor Input
IMON1P	19	Analog Input	Low Voltage Current Monitor Input source
IMON1N	20	Analog Input	Low Voltage Current Monitor Input return
TMON1P	21	Analog Input	Temperature Monitor Input source
TMON1N	22	Analog Input	Temperature Monitor Input return
TMON2P	23	Analog Input	Temperature Monitor Input source
TMON2N	24	Analog Input	Temperature Monitor Input return
GPIO1	44	Digital I/O	Digital Input/ Open Drain Output, reset Low
GPIO2	45	Digital I/O	Digital Input/ Open Drain Output, reset Low
GPIO3	46	Digital I/O	Digital Input/ Open Drain Output, reset Low
GPIO4	47	Digital I/O	Digital Input/ Open Drain Output, reset Low
GPIO5	48	Digital I/O	Digital Input/ Open Drain Output, reset Low
GPIO6[2]	1	Digital I/O	Digital Input/ Open Drain Output, reset Low
GPIO8[2]	11	Digital I/O	Digital Input/ Open Drain Output, reset Hi-Z
GPIO9	12	Digital I/O	Digital Input/ Open Drain Output, reset Hi-Z
GPIO10	13	Digital I/O	Digital Input/ Open Drain Output, reset Low
HVOUT1	2	Analog/Digital Out	Current Source/ Open Drain Output, reset Low
HVOUT2	3	Analog/Digital Out	Current Source/ Open Drain Output, reset Low
HVOUT3	9	Analog/Digital Out	Current Source/ Open Drain Output, reset Low
HVOUT4	10	Analog/Digital Out	Current Source/ Open Drain Output, reset Low
TRIM1	39	Analog Output	Trim DAC Output, reset Hi-Z
TRIM2	40	Analog Output	Trim DAC Output, reset Hi-Z
TRIM3	41	Analog Output	Trim DAC Output, reset Hi-Z
TRIM4	42	Analog Output	Trim DAC Output, reset Hi-Z
RESETb[1]	43	Digital I/O	Device reset (Active Low)
SCL	15	Digital Input	Slave I^2C Serial Clock input
SDA	14	Digital I/O	Slave I^2C Serial Data, Bi-directional pin
I2C_ADDR	16	Analog Input	Resistor Input to set I^2C address low bits
ASCCLK	7	Digital Output	8 MHz ASC Clock Output (Tristate) CMOS
RDAT	5	Digital Output	ASC Interface Data signal
WDAT	4	Digital Input	ASC Interface Data Signal

Pin Function	48-Pin QFN	Pin Type	Description
WRCLK	6	Digital Input	ASC Interface Clock signal
VCCA	33	Power	Main Power Supply
	8	Power	
GND	49	Power	Exposed die pad is the device ground

1. Do not connect any external drivers (push buttons, switches, or other devices) to the RESETb pins. See the Reset Requirements section for more information.
2. GPIO7 is not bonded out.
3. When only using HIMONN_HVMON to measure voltage, connect HIMONP to the same source to prevent differential over-voltage.

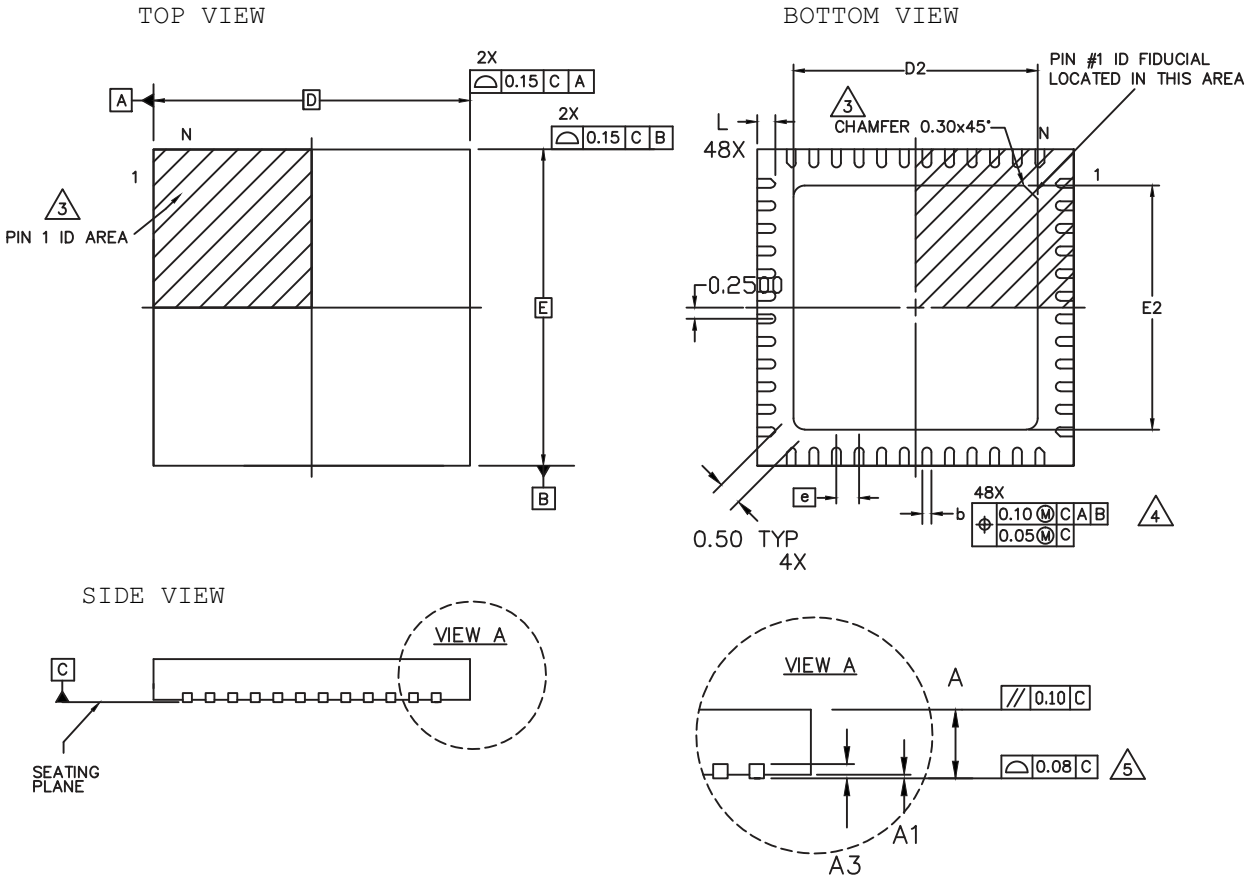
Device Pinout

Top View



Package Diagram

48-Pin QFN (Dimensions in mm)

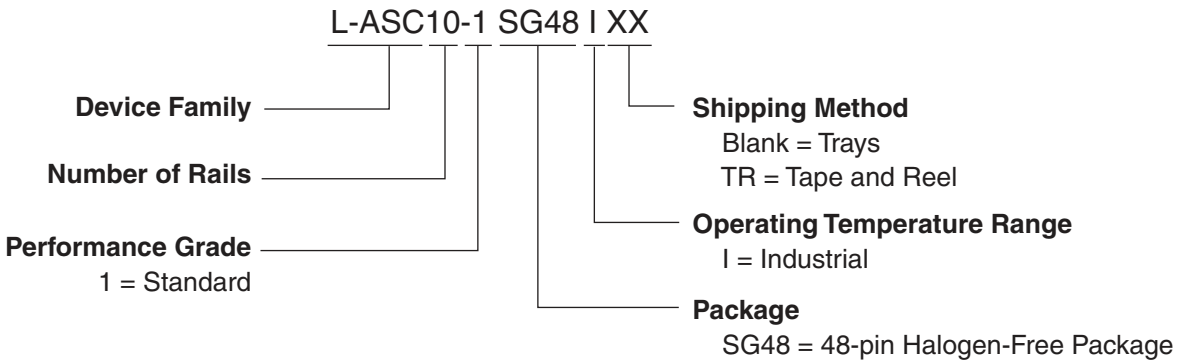


NOTES: UNLESS OTHERWISE SPECIFIED

1. DIMENSIONS AND TOLERANCES PER ANSI Y14.5M.
2. ALL DIMENSIONS ARE IN MILLIMETERS.
3. EXACT SHAPE AND SIZE OF THIS FEATURE IS OPTIONAL.
4. DIMENSION b APPLIES TO PLATED TERMINAL AND IS MEASURED BETWEEN 0.15 AND 0.30 mm FROM TERMINAL TIP.
5. APPLIES TO EXPOSED PORTION OF TERMINALS.

SYMBOL	MIN.	NOM.	MAX.
A	0.80	0.90	1.00
A1	0.00	0.02	0.05
A3		0.2 REF	
D		7.0 BSC	
D2	5.30	5.40	5.50
E		7.0 BSC	
E2	5.30	5.40	5.50
b	0.15	0.20	0.25
e		0.50 BSC	
L	0.35	0.40	0.45

Part Number Description



Ordering Information

Halogen-Free Packaging

Part Number	Package	Pins
L-ASC10-1SG48I	Halogen-Free QFN	48

For Further Information

For more information on the Platform Manager 2 family of devices, consult the Platform Manager 2, MachXO2, MachXO3, and ECP5 family data sheets along with related application and technical notes on the Lattice website.

- FPGA-DS-02012 (previously DS1044), ECP5 and ECP5-5G Family Data Sheet
- DS1035, MachXO2 Family Data Sheet
- FPGA-DS-02032 (previously DS1047), MachXO3 Family Data Sheet
- FPGA-DS-02036 (previously DS1043), Platform Manager 2 Family Data Sheet
- AN6094, Adding Scalable Power and Thermal Management to MachXO2 and MachXO3 Using L-ASC10
- AN6095, Adding Scalable Power and Thermal Management to ECP5 Using L-ASC10
- TN1225, Platform Manager 2 Hardware Checklist
- Platform Designer User Guide

Technical Support Assistance

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

Revision History

Date	Version	Section	Change Summary
September 2019	2.1	Temperature Monitor Inputs	Updated Figure Number in Table 7.
		I²C Interface	— Updated Raddr Values in Table 17. — Added footnotes to Table 17. — Added clarification to Voltage Monitor ADC Readout Over I²C.
June 2019	2.0	—	Clarified Closed Loop Trim mode and Bypass mode in several sections.
		Part Number Description	Added Tape and Reel.
May 2019	1.9	Device Pinout	Added Top View label.
		Package Diagrams	Correctly labeled Top and Bottom views.
		—	Added Disclaimers section.
September 2018	1.8	—	Changed document number from DS1042 to FPGA-DS-02038.
		Application Diagram	Added LPTM21L to Figure 1.
		DC and Switching Characteristics	Clarification added to Fault Log table.
		Theory of Operation	Updated High Voltage Monitor section. Added information on extending input voltage range.
			Updated Calculation section. — Added information on PGA settings under the Voltage at the VMONx Pins subsection.
			Updated System Connections section. — Added reference to Figure 32 in the section introduction. — Added Table 15. — Indicated Platform Manager 2 LPTM21 in Figure 30 and added note.
		I²C Interface	Added LPTM21L row to Table 20.
		Pin Descriptions	Added note 3.
		For Further Information	Added product names and updated document numbers.
June 2017	1.7	Multiple	Added references to MachXO3 and ECP5.
		Theory of Operation	Added clarification to TrimCell Architecture section. Updated Figure 26, TrimCell Architecture.
			— Modified caption of Figure 31, System Connections - ASC and MachXO2, or MachXO3. — Added Figure 32, System Connections - ASC and ECP5.
May 2016	1.6	I²C Interface	Updated Measurement and Control Register Access section to address restrictions on READ_MEAS_CTRL command with addresses 0x85 and 0x86.
April 2015	1.5	Multiple	Deleted all references to LPTM20.
		Theory of Operation	Updated System Connections section.
			Modified the following figures to clarify I2C_ADDR pin usage: — Figure 30, System Connections - ASC and Platform Manager 2 — Figure 31, System Connections - ASC and MachXO2, MachXO3, or ECP5
		I²C Interface	Updated ASC Configuration Registers section. Updated Polarity bit setting in Table 23, POL Setting vs Closed Loop Trim Polarity.
October 2014	1.4	I²C Interface	Updated Device Status and Mode Management section. Revised Figure 39, READ_STATUS - I2C Instruction Format.
		Package Diagram	Updated 48-Pin QFN (Dimensions in mm) diagram.

Date	Version	Section	Change Summary
May 2014	01.3	—	Data sheet status changed from preliminary to final.
		DC and Switching Characteristics	Specifications populated with characterization results.
			Added ASC-I/F Timing section.
		Multiple	Renamed IMON to IMON1.
			Updated ASC-IF TRIM control signal names.
		Theory of Operation	Removed IMON Hysteresis feature.
			Expanded Output Control Block section.
			Updated System Connections section.
		I^2C Interface	Corrected error in ADC Input Selection table for IMON1 and HIMON SEL bits.
			Updated VMON and IMON tables with final device trip points.
March 2014	01.1	DS and Switching Characteristics	Added preliminary ESD Performance section.
	01.2		Corrected formatting error on Page 27. Moved footnote after Figure 24, ASC Margin/Trim Block.
December 2013	01.0		Preliminary release.