LatticeMico PMBus Adapter

The LatticeMico PMBus Adapter provides major functions of PMBus protocol 1.1 for PMbus master to configure and monitor Analog Sense and Control (ASC) registers for Lattice Platform Manager 2 devices. The LatticeMico PMBus adapter also provide the SMBAltert for fault detection and the host can monitor the fault status through status commands. The LatticeMico PMBus master can control the operation of system such as system on or off, margin up or down and interleave on or off.

Version

This document describes the 1.0 version of the LatticeMico PMBus Adapter.

Features

The LatticeMico PMBus Adapter IP must be used together with the EFB (Embedded Functional Block) of the Platform Manager 2 device. Updating fault limit and measurement values are using the primary of I^2C port of the EFB. The LatticeMico PMBus Adapter has the following features to support the PMBus task:

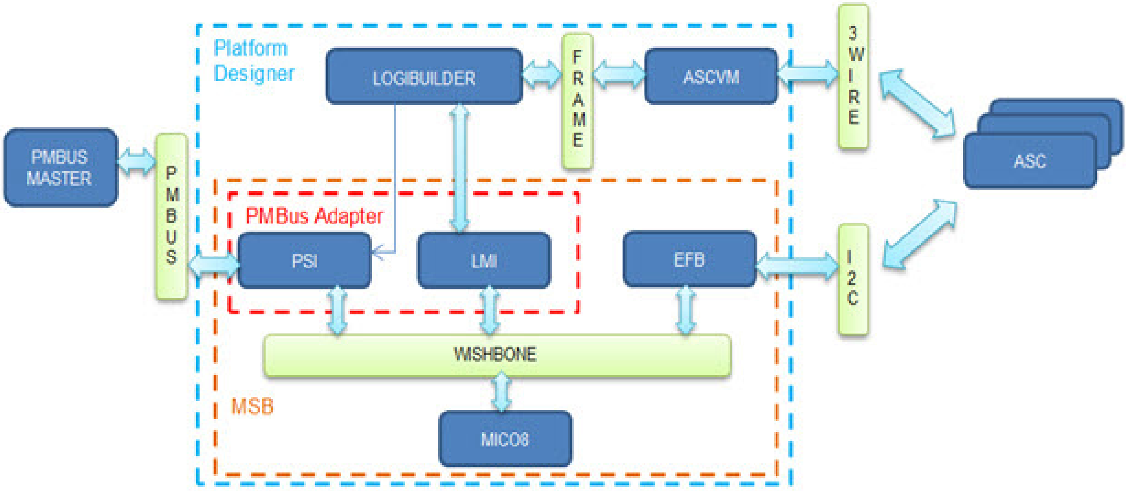
- WISHBONE interfaces with 8-bit data bus.
- Responds with a key capabilities byte depending on configuration.
- Turns the sequencing on or off; or margin up or down; or interleave one or off; to implement the functions needed.
- Prevents the external PMBus master from updating the threshold and margining by write protect.
- Allows the Platform Manager 2 device to manage the Analog Point of Loads (APOLs), the ASCs, and LogiBuilder (LgB).

- PMBus host can configure and monitor up to eight ASCs.
- PMBus host can configure and monitor ASC registers through LatticeMico PMBus Adapter dynamically.
- Indicate fault status driven by LogiBuilder.

Functional Description

The LatticeMico PMBus Adapter can be configured into several modes by selecting the configuration options. Enable SMBAlert, Package Error Checking (PEC), or both by selecting the appropriate options during configuration.

Figure 1: LatticeMico PMBus Adapter Block Diagram



For more information on the ASC functions, refer to *DS1042, L-ASC10 Data Sheet*.

Basic Operation

Table 1 shows the summary of LatticeMico PMBus Adapter supporting commands.

Table 1: LatticeMico PMBus Adapter Supporting Commands

Command		**Type**	**Byte**	**Page (Lower 4-bits)**		
Code	**Name**			**0x00-0x2F**	**0x30-0x3F**	**0x40-0x5F**
0x00	Page	Read/Write Byte	1	Read/Write	Read/Write	Read/Write
0x19	Capability	Read Byte	1	No Page Support		
0x01	Operation	Read/Write Byte	1	No Page Support		
0xD3	MFR_Coefficient	Read Word	2		Read Only	
0x78	Status_Byte	Read Byte	1	No Page Support		
0x79	Status_Word	Read Word	2	No Page Support		
0x7A	Status_VOUT	Read Byte	1	No Page Support		
0x7B	Status_IOUT	Read Byte	1	No Page Support		
0x7C	Status_Input	Read Byte	1	No Page Support		
0x7D	Status_Temperature	Read Byte	1	No Page Support		
0x7E	Status_CML	Read Byte	1	No Page Support		
0x7F	Status_Other	Read Byte	1	No Page Support		
0x80	Status_MFR_Specific	Read Byte	1	No Page Support		
0x81	Status_FANs_1_2	Read Byte	1	No Page Support		
0x82	Status_FANs_3_4	Read Byte	1	No Page Support		
0x03	Clear_Faults	Send Byte	1	No Page Support		
0xD0	MFR_Interleave_Off	Send Byte	1	No Page Support		
0xD1	MFR_Interleave_On	Send Byte	1	No Page Support		
0x40	MFR_VOUT_OV_Fault_Limit	Read/Write Word	2	Read/Write	-	-
0x44	MFR_VOUT_UV_Fault_Limit	Read/Write Word	2	Read/Write	-	-
0x46	MFR_IOUT_OC_Fault_Limit	Read/Write Word	2	-	Read/Write	-
0x4B	MFR_IOUT_UC_Fault_Limit	Read/Write Word	2	-	Read/Write	-
0x4F	MFR_TOUT_OT_Fault_Limit	Read/Write Word	2	-	-	Read/Write
0x53	MFR_TOUT_UT_Fault_Limit	Read/Write Word	2	-	-	Read/Write
0x8B	MFR_Read_VOUT	Read Word	2	Read Only	-	-
0x8C	MFR_Read_IOUT	Read Word	2	-	Read Only	-
0x8D	MFR_Read_Temperature	Read Word	2	-	-	Read Only

Table 1: LatticeMico PMBus Adapter Supporting Commands (Continued)

Command		Type	Byte	Page (Lower 4-bits)		
Code	**Name**			**0x00-0x2F**	**0x30-0x3F**	**0x40-0x5F**
0x98	PMBus_Revision	Read Byte	1	No Page Support		
0x10	Write_Protect	Read/Write Byte	1	No Page Support		

Page Command

The Page command uses the read/write protocol with one data byte. The PMBus host can either read or write a new page value into the LatticeMico8 microcontroller data memory. The valid range of page number is from 0x00 to 0x5F. If the PMBus host reads or writes a page number which is not in this range or not been configured any valid value, the CML MICO_PMBUS_INVALID_DATA bit will be flagged and/or trigger the SMBAlert signal, and PMBus Adapter will also ignore this page command. In this case, the page number will not change to new value and remain the least valid page number.

Write Page Command Behavior

1. The LatticeMico8 microcontroller translates the page value to the corresponding ASC register
2. The LatticeMico8 microcontroller saves this page value into the microprocessor memory.
3. The LatticeMico8 microcontroller sets the coefficient value based on the page number
4. The LatticeMico8 microcontroller reads the threshold value from the local cache memory

The LatticeMico8 microcontroller reads the voltage/current/temperature value from the ASC immediately based on the current PMBus page value. This Improves the performance if the subsequent command is a Read Value command. A subsequent PMBus Write/Read command is expected immediately after the Page command is issued.

Read Page Command Behavior

1. The LatticeMico8 microcontroller will respond the PMBus Host with the current Page value that has been set.

The Page command provides the ability to configure and monitor multiple ASC VMON/IMON/TMON register value using the same PMBUS address. The mapping should be dynamic according to the user’s selection in the Platform Designer software.

Table 2 shows the ASC VMON/IMON/TMON mapping.

Table 2: Page Number and corresponding Monitor

Page Numbers	Function
0x00-0x2F	VMON
0x30-0x3F	IMON
0x40-0x5F	TMON
Others	Reserved

Capability Command

The Capability command uses the Read Byte protocol with one data byte. This command provides a way for the PMBus Master to determine some key capabilities of the PMBus Slave device.

Table 3: Detailed Information of Capability Byte

Bits	Description	Value	Meaning
7	PEC	0	Not supported
		1	Support
6:5	Bus Speed	00	Max Bus Speed = 100kHz
		01	Max Bus Speed = 400kHz
		10/11	Reserved
4	SMBALERT	0	Not Support this pin
		1	Support this pin
3.0	Reserved	X	Reserved

Operation Command

The Operation command uses the Read/Write protocol with one data byte. The PMBus host can either read or write the operation value from the PMBus IP. This command is used to turn the unit on and off in conjunction with the input from the control pin. This command is also used to cause the unit to set the output voltage to the upper or lower margin voltages. The unit stays in the commanded operating mode until a subsequent Operation command instructs the device to change to another mode. The Operation command can use for any monitor in any ASC (IMON, VMON, and TMON).

“Margin Low (Act on Fault)” means that if a under-voltage/current/temperature is detected, the unit treats this as a fault and responds by a fault response command (or sends a PMBALERT to the PMBus master). Similarly, “Margin

High (Act on Fault)" means an over-voltage/current/temperature is treated as a fault.

"Ignore Fault" means that any voltage/current/temperature fault that is detected will be ignored, the unit will ignore the condition, and the unit will continue to operate.

Each Operation setting option maps to a particular LMI output port. Table 4 shows the Operation setting options and the mapping between the settings and LMI output ports.

Table 4: Summary of Operation Setting Options

Bit[7:6]	Bit[5:4]	Bit[3:2]	Bit[1:0]	Unit	Margin	LMI Port
00	XX	XX	XX	Immediate Off	N/A	PMBUS_IMMED_OFF
01	XX	XX	XX	Soft Off	N/A	PMBUS_SOFT_OFF
10	00	XX	XX	On	Off	PMBUS_MARGIN_OFF
10	01	01	XX	On	Margin Low (Ignore Fault)	PMBUS_MARGIN_LOW_IF
10	01	10	XX	On	Margin Low (Act On Fault)	PMBUS_MARGIN_LOW_AF
10	10	01	XX	On	Margin High (Ignore Fault)	PMBUS_MARGIN_HI_IF
10	10	10	XX	On	Margin High (Act On Fault)	PMBUS_MARGIN_HI_AF

Manufacturer Coefficient Command

The Manufacturer Coefficient command uses the Read Word command with 2 data bytes. This command is similar to the Coefficient command in the PMBus Specification, except it can only read out the "m" coefficient of IMON. The PMBus Host can use this command to retrieve the "m" coefficients needed by the data in Direct Format for encoding the actual threshold or measurement value of IMON.

If the user does not set this coefficient value, the default value of the coefficient will be 0. In this case, if the PMBus host requests to write IMON Fault Limit, the PMBus Adapter will not setup the IMON Fault Limit value, and it will flag CML MICO_PMBUS_STATUS_CML_INVALID_DATA bit and/or will trigger the SMBAlert.

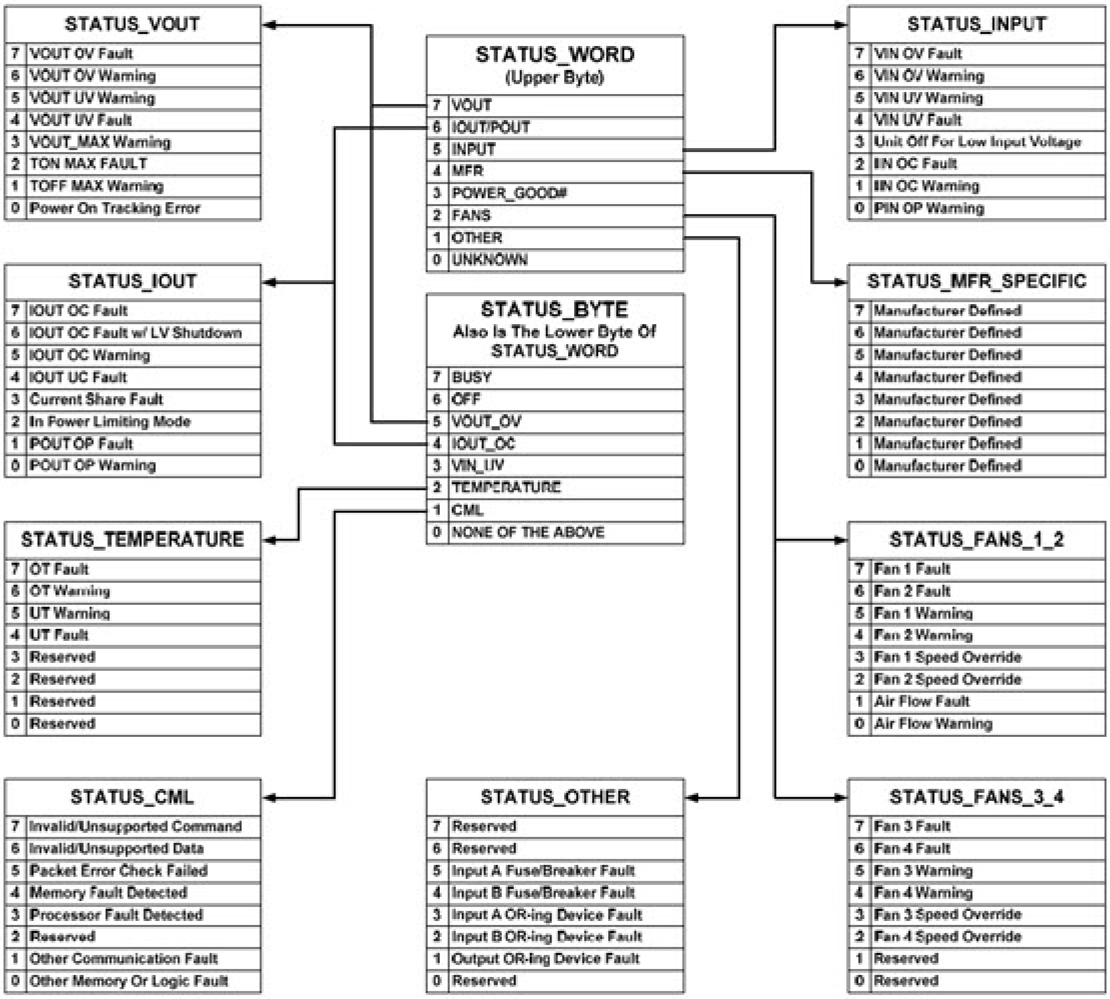
For more information about this data, refer to "Fault_limit Command" on page 14.

Status Command

The Status command uses the Read Byte/Word Protocol. All Status commands are binary and read only; these commands provide a way for the PMBus Master to retrieve the status information from the LatticeMico PMBus Adapter current status information.

The user is responsible for connecting these signals in the LogiBuilder in the Platform Designer software, and sending the corresponding status information back to the LatticeMico8 microcontroller.

Figure 2: Status Commands Mapping



STATUS_VOUT	
7	VOUT OV Fault
6	VOUT OV Warning
5	VOUT UV Warning
4	VOUT UV Fault
3	VOUT_MAX Warning
2	TON MAX FAULT
1	TOFF MAX Warning
0	Power On Tracking Error

STATUS_WORD (Upper Byte)	
7	VOUT
6	IOUT/POUT
5	INPUT
4	MFR
3	POWER_GOOD#
2	FANS
1	OTHER
0	UNKNOWN

STATUS_INPUT	
7	VIN OV Fault
6	VIN OV Warning
5	VIN UV Warning
4	VIN UV Fault
3	Unit Off For Low Input Voltage
2	IIN OC Fault
1	IIN OC Warning
0	PIN OP Warning

STATUS_IOUT	
7	IOUT OC Fault
6	IOUT OC Fault w/ LV Shutdown
5	IOUT OC Warning
4	IOUT UC Fault
3	Current Share Fault
2	In Power Limiting Mode
1	POUT OP Fault
0	POUT OP Warning

STATUS_BYTE Also Is The Lower Byte Of STATUS_WORD	
7	BUSY
6	OFF
5	VOUT_OV
4	IOUT_OC
3	VIN_UV
2	TEMPERATURE
1	CML
0	NONE OF THE ABOVE

STATUS_MFR_SPECIFIC	
7	Manufacturer Defined
6	Manufacturer Defined
5	Manufacturer Defined
4	Manufacturer Defined
3	Manufacturer Defined
2	Manufacturer Defined
1	Manufacturer Defined
0	Manufacturer Defined

STATUS_TEMPERATURE	
7	OT Fault
6	OT Warning
5	UT Warning
4	UT Fault
3	Reserved
2	Reserved
1	Reserved
0	Reserved

STATUS_FANS_1_2	
7	Fan 1 Fault
6	Fan 2 Fault
5	Fan 1 Warning
4	Fan 2 Warning
3	Fan 1 Speed Override
2	Fan 2 Speed Override
1	Air Flow Fault
0	Air Flow Warning

STATUS_CML	
7	Invalid/Unsupported Command
6	Invalid/Unsupported Data
5	Packet Error Check Failed
4	Memory Fault Detected
3	Processor Fault Detected
2	Reserved
1	Other Communication Fault
0	Other Memory Or Logic Fault

STATUS_OTHER	
7	Reserved
6	Reserved
5	Input A Fuse/Breaker Fault
4	Input B Fuse/Breaker Fault
3	Input A OR-ing Device Fault
2	Input B OR-ing Device Fault
1	Output OR-ing Device Fault
0	Reserved

STATUS_FANS_3_4	
7	Fan 3 Fault
6	Fan 4 Fault
5	Fan 3 Warning
4	Fan 4 Warning
3	Fan 3 Speed Override
2	Fan 4 Speed Override
1	Reserved
0	Reserved

Status_Byte Command

The Status_Byte command returns one byte of information with a summary of the most critical faults/warnings. LogiBuilder is responsible for setting these Ports/Nodes and LatticeMico PMBus Adapter will return these bits when the Status_Byte command is requested.

Table 5: Status_Byte Summary

Bit	Status
7	BUSY
6	OFF
5	VOUT_OV
4	IOUT_OC
3	VIN_UV
2	TEMPERATURE
1	CML
0	NONE OF THE ABOVE

Note: The CML bit is controlled by the PMBus Adapter only. The user cannot control this bit through LogiBuilder or any other input signal.

Status_Word Command

The Status_Word command returns two byte of information with a summary of the unit’s faults/warnings. Based on this information, the PMBus host can get more information by reading the appropriate Status Registers. LogiBuilder is responsible for setting these Ports/Nodes and LatticeMico PMBus Adapter will return these bits when the Status_Word command is requested.

Table 6: Status_Word Summary - Byte0

Bit	Status
7	BUSY
6	OFF
5	VOUT_OV
4	IOUT_OC
3	VIN_UV
2	TEMPERATURE
1	CML

Table 6: Status_Word Summary - Byte0 (Continued)

Bit	Status
0	NONE OF THE ABOVE

Note: The CML bit is controlled by the PMBus Adapter only. The user cannot control this bit through LogiBuilder or any other input signal.

Table 7: Status_Word Summary - Byte1

Bit	Status
7	VOUT
6	IOUT/POUT
5	MFR
4	POWER_GOOD
3	POWER_GOOD#
2	FANS
1	OTHERS
0	UNKNOWN

Status_VOUT

The Status_VOUT command returns one byte of information with content as follows. LogiBuilder is responsible for setting these ports/nodes and LatticeMico PMBus Adapter will return these bits when the Status_VOUT command is requested.

Table 8: Status_VOUT Summary

Bit	Status
7	VOUT_OV_Fault
6	VOUT_OV_Warning
5	VOUT_UV_Fault
4	VOUT_UV_Warning
3	VOUT_MAX_Warning
2	TON MAX Fault
1	TOFF MAX Warning
0	Power On Tracking Error

Status_IOUT

The Status_IOUT command returns one byte of information with content as follows. LogiBuilder is responsible for setting these Ports/Nodes and LatticeMico PMBus Adapter will return these bits when the Status_IOUT command is requested.

Table 9: Status_IOUT Summary

Bit	Status
7	IOUT_OC_Fault
6	IOUT_OC_Fault w/ LV Shutdown
5	IOUT_OC_Warning
4	IOUT_UC_Fault
3	Current Share Fault
2	In Power Limiting Mode
1	POUT OP Fault
0	POUT OP Warning

Status_Temperature

The Status Temperature command returns one byte of information with content as follows. LogiBuilder is responsible for setting these Ports/Nodes and PMBus will return these bits when the Status_Temperature command is requested.

Table 10: Status_Temperature Summary

Bit	Status
7	OT_Fault
6	OT_Warning
5	UT_Warning
4	UT_Fault
3	Reserved
2	Reserved
1	Reserved
0	Reserved

Status_CML

The Status_CML command returns one byte of information with content as follows. LatticeMico PMBus Adapter will return these bits when the Status_CML command is requested. The Status_CML byte is controlled by the PMBus Adapter only. The user cannot control this bit through LogiBuilder or any other input signal.

Table 11: Status_CML Summary

Bit	Status
7	Invalid/Unsupported Command
6	Invalid/Unsupported Data
5	Packet Error Check Failed
4	Memory Fault Detected
3	Processor Fault Detected
2	Reserved
1	Other Communication Fault
0	Other Memory or Logic Fault

Status_Other

The Status Other command returns one byte of information with content as follows. LogiBuilder is responsible for setting these Ports/Nodes and LatticeMico PMBus Adapter will return these bits when the Status_Other command is requested.

Table 12: Status_Other Summary

Bit	Status
7	Reserved
6	Reserved
5	Input A Fuse/Breaker Fault
4	Input B Fuse/Breaker Fault
3	Input A OR-ing DeviceFault
2	Input B OR-ing DeviceFault
1	Output OR-ing DeviceFault
0	Reserved

Status_Input

The Status Input command returns one byte of information with content as follows. LogiBuilder is responsible for setting these Ports/Nodes and LatticeMico PMBus Adapter will return these bits when the Status_Input command is requested.

Table 13: Status_Input Summary

Bit	Status
7	VIN OV Fault
6	VIN OV Warning
5	VIN UV Fault
4	VIN UV Warning
3	Unit Off for Low Input Voltage
2	IN OC Fault
1	IN OC Warning
0	PIN OP Warning

Status_MFR_Specific

The Status_MFR_Specific command returns one byte of information with content as follows. LogiBuilder is responsible for setting these Ports/Nodes and LatticeMico PMBus Adapter will return these bits when the Status_MFR_Specific command is requested.

Table 14: Status_MFR_Specific Summary

Bit	Status
7	Manufacturer Defined
6	Manufacturer Defined
5	Manufacturer Defined
4	Manufacturer Defined
3	Manufacturer Defined
2	Manufacturer Defined
1	Manufacturer Defined
0	Manufacturer Defined

Status_FANs_1_2

The Status Fans_1_2 command returns one byte of information with content as follows. LogiBuilder is responsible for setting these ports/nodes and LatticeMico PMBus Adapter will return these bits when the Status_FANs_1_2 command is requested.

Table 15: Status_FANs_1_2 Summary

Bit	Status
7	Fan 1 Fault
6	Fan 2 Fault
5	Fan 1 Warning
4	Fan 2 Warning
3	Fan 1 Speed Override
2	Fan 2 Speed Override
1	Air Flow Fault
0	Air Flow Warning

Status_FANs_3_4

The Status_Fans_3_4 command returns one byte of information with content as follows. LogiBuilder is responsible for setting these Ports/Nodes and LatticeMico PMBus Adapter will return these bits when the Status_Fans_3_4 command is requested.

Table 16: Status_Fans_3_4 Summary

Bit	Status
7	Fan 3 Fault
6	Fan 4 Fault
5	Fan 3 Warning
4	Fan 4 Warning
3	Fan 3 Speed Override
2	Fan 4 Speed Override
1	Reserved
0	Reserved

Clear_Fault Command

The Clear_Fault command uses the Send Byte Protocol with one data byte. This PMBus command will be translated to the “PMBUS_CLEAR_FAULT” Nodes in the LogiBuilder.

This command is used to clear fault bits that have been set in both the LatticeMico8 microcontroller and LogiBuilder. At the same time, the LogiBuilder should negates the SMBALLERT signal if asserted.

If the fault is still present when the bit is clear, it is recommended to set the fault again immediately and notify the host.

MFR_Interleave_On/Off Command

The MFR_Interleave_Fault command uses the Send Byte Protocol with one data byte. This PMBus command will be translated to the “PMBUS_MFR_INTERLEAVE” in the LogiBuilder.

Fault_limit Command

The Fault command uses the Read/Write Word Protocol with two data bytes. The PMBus Host can only use the corresponding Fault Limit on specific page The user can modify the page information using the PMBus PAGE command.

If the VMON/TMON data does not match any valid values, the LatticeMico8 microcontroller will set the data to the closest value.

If the IMON data does not match to any valid value, the LatticeMico8 microcontroller will throw an error and set the CML status Bit 6 – Invalid Data and send the SMBAlert to the PMBus Host.

If the page does not support the command, the LatticeMico8 microcontroller will set the CML status Bit 7 – Invalid command, will be set and send the SMBAlert to the PMBus Host.

Table 17: Page number v.s. Fault limit type

Page Number	Function
0x00-0x2F	Voltage Fault Limit
0x30-0x3F	Current Fault Limit
0x40-0x5F	Temp Fault Limit

The host system uses the following equation to convert the value received from the PMBus device into a reading of volts, amperes, degree Celsius or other units as appropriate.

$$X = \frac{1}{m} \cdot (Y \cdot 10^{-R} - b)$$

- ***X*** is the calculated, real world value.
- ***Y*** is a 2 bytes two's complement integer received from PMBus device.
- ***m*** is the slope, a 2 bytes two's complement integer.
- **R** is the exponent, a 2 bytes two's complement integer always**.**
- **b** is the offset, a 2 bytes two's complement integer.

Sending a data to the LatticeMico PMBus Adapter

$$Y = (mX + b) \cdot 10^R$$

- ***X*** is real world value.
- ***Y*** is a 2 bytes two's complement integer to be sent to the LatticeMico PMBus Adapter.
- ***m*** is the slope, a 2 bytes two's complement integer.
- **R** is the exponent, a 2 bytes two's complement integer always**.**
- **b** is the offset, a 2 bytes two's complement integer

These values should reference to following Fault Limit configuration.

ASC VMON Fault_limit Configuration

The voltage monitor (VMON1-VMON9 and HVMON) are configurable over I^{2}C. 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 point settings should reference the ASC configuration data sheet. Any settings outside of the table range are prohibited.

When the LatticeMico8 microcontroller reads/writes the voltage threshold value, the LatticeMico8 microcontroller sets the fault limit values with a resolution of 2mV.

- ***m*** **= 500**, a 2 bytes two's complement integer.
- **R = 0,** a 2 bytes two's complement integer
- **b = 0,** a 2 bytes two's complement integer

Reading a VMON threshold/measurement value:

$$X = \frac{Y}{500}$$

- ***X*** is the calculated, real world value in volts;
- ***Y*** is a 2 bytes two's complement integer received from PMBus Adapter

Sending a VMON Threshold data to the PMBus Slave Adapter:

$$Y = (500X)$$

- ***X*** is real world value in volts;
- ***Y*** is a 2 bytes two's complement integer send to LatticeMico PMBus Adapter

ASC IMON Fault_limit Configuration

When the LatticeMico8 microcontroller reads/writes the voltage threshold value, the LatticeMico8 microcontroller sets the Coefficient values to R(sense) / IMON resolution.

- ***m* = R$_{\text{sense}}$ / IMON resolution**, a 2 bytes two's complement integer.
- **R = 0,** a 2 bytes two's complement integer
- **b = 0,** a 2 bytes two's complement integer

Reading an IMON threshold/measurement value:

$$X = \frac{Y * \text{IMON resolution}}{\text{Rsense}}$$

- ***X*** is the calculated, real world value in amps;
- ***Y*** is a 2 bytes two's complement integer received from the PMBus Adapter.

Sending an IMON threshold data:

$$Y = \frac{x * \text{Rsense}}{\text{IMON resolution}}$$

- ***X*** is the real world value in Amps;
- ***Y*** is a 2 bytes two's complement integer sent to the PMBus Adapter

Note

The user provides the voltage value. No conversion will be done in the LatticeMico8 microcontroller.

	Resolution	1/ Resolution
IMON	0.25 mA	4000

The current value can be calculated by this formula:

V = IR, where:

- V is the voltage threshold.
- R is the resistance value defined by the user.

For example, if trying to set 5 mA as the fault limit value and the resistance value is 2 ohms:

2 bytes two's complement integer data Y = $\frac{5m * 2}{0.25m}$ = 40 mV

Table 18 shows the valid voltage threshold values for IMON.

Table 18: Valid Voltage Threshold Values to Set IMON Fault Limit

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

ASC TMON Fault_limit Configuration

The TMON includes two individually programmable comparators, TMONA and TMONB with thresholds range for each of these monitors is -64 ºC to 155 ºC, with a resolution of 1 ºC. Values above 155 ºC or below -64 ºC are not valid threshold settings.

- ***m* = 4**, a 2 bytes two's complement integer.
- **R = 0,** a 2 bytes two's complement integer
- **b = 0,** a 2 bytes two's complement integer

Reading a TMON threshold/measurement value:

$$X = \frac{Y}{4}$$

- ***X*** is the calculated, real world value in degrees C;
- ***Y*** is a 2 bytes two's complement integer received from LatticeMico PMBus Adapter

Sending a TMON threshold value:

$$Y = (4X)$$

- ***X*** is real world value in degrees C;
- ***Y*** is a 2 bytes two's complement integer send to LatticeMico PMBus Adapter

Read_Value Command

The Read_Value command uses the Read_Word protocol with two data bytes. Similar to the Fault_Limit command, the PMBus Master can only use the corresponding Fault Limit on specific page. Beware that reading the measurement is read at the time when the Page command is issued. To achieve a better result, it is recommended to send a Page command to the LatticeMico PMBus Adapter before reading the value.

VMON read value:

VMON measurement value = 2 bytes two's complement integer with 2 mV resolution.

IMON read value:

IMON measurement value = 2 bytes two's complement integer *Coefficient in amps.

IMON read value:

IMON measurement value = 2 bytes two's complement integer with one degree C resolution.

If the page does not support the command, the CML status Bit 7 – Invalid command, will be set. The SMBAlert will be sent out to the Host if this signal is enabled.

Table 19: Page number v.s. Read measurement

Page Numbers	Function
0x00-0x2F	Read Voltage
0x30-0x3F	Read Current
0x40-0x5F	Read Temp

PMBus_Revision Command

The PMBUS_Revision command uses the Read Byte protocol with one data byte.

Write_Protect Command

The Write_Protect command uses the Read Byte protocol with one data byte. This command prevents the external PMBus master from updating the threshold and margining. Table 20 describes the Write_Protect operation.

Table 20: Write_Protect Summary

Value	Description
10000000	Disable All Write Command except Write_Protect
01000000	Disable All Write Command except Write Protect, Operation and Page
00000000	Enable All Write Command
Others	Reserved

Configuration

The following sections describe the graphical user interface (UI) parameters, the hardware description language (HDL) parameters, and the I/O ports that user can use to configure and operate the LatticeMico PMBus Adapter.

User Interface Parameters

Table 21 shows the User Interface (UI) parameters available for configuring the LatticeMico PMBus Adapter through the Mico System Builder (MSB) interface. For more information, refer to the Platform Manager 2 PMBus Adapter documentation in the Diamond software online help.

Table 21: PMBus Adapter UI Parameters

Dialog Box Option	Description	Allowable Values	Default Value
Instance Name	Specifies the name of the PMBus Adapter instance.	Alphanumeric and underscores	pmbus
PMBus Address Settings			
Adapter Base Address	Specifies the base address for configuring the PMBus Adapter. The minimum boundary alignment is 0x80.	0X80000000 – 0XFFFFFFFF	0X80000000
PMBus Slave Address	Specifies the PMBus Adapter slave address.	0x00 – 0x7F	0x60
PMBus Slave Interface Settings			
Enable support for SMBALERT Signal	When selected, PMBus will enable SMBALERT support.	selected|not selected	selected
Enable Package Error Checking (PEC)	When selected, PMBus will be enable Package Error Checking support.	selected|not selected	selected
Enable PMBus Slave Stretching	When selected, PMBus will enable PMBus Slave Clock Stretching support.	selected|not selected	selected
PMBus Bus Speed (kHz)	Specifies the PMBus Bus speed.	100kHz|400kHz	400kHz
LogiBuilder Interface Settings			
Enable LgB Interface	When selected, PMBus will enable SMBALERT support.	selected|not selected	selected
LgB Base Address	Specifies the base address for configuring the LogiBuilder (LMI).	0x00000000 – 0XFFFFFFFF	0x00000000

HDL Parameters

Table 22 lists the parameters that appear in the LatticeMico PMBus HDL.

Table 22: LatticeMico PMBus Adapter HDL Parameter

Parameter Name	Description	Allowable Value
PSMBALERT	Define enable/disable SMBALERT Signal support	0|1
PPEC_SUPPORT	Define enable/disable Package Error Checking support	0|1
PMAX_BUS_SPEED	Define MAX_BUS_SPEED. 1 is 400kHz; 0 is 100kHz	0|1
PSLAVE_ADDRESS	Define PMBus Adapter Slave Address	0x00 to 0x7F
PSMBUS_TIMEOUT	Define PMBus Timeout (for clock stretching)	0 to 2^{10}-1
PCLK_STRETCH	Define enable/disable PMBus Adapter Slave Clock Stretching support	0|1
ASC_S_ADDR	Define the Base ASC Address on I^2C Bus	0 to 0x70
Page_00_MAP – Page_2F_MAP	Store the ASC and Voltage Monitor (VMON) Mapping	0x00 to 0xFF
Page_30_MAP – Page_3F_MAP	Store the ASC and Current Monitor (IMON) Mapping	0x00 to 0xFF
Page_40_MAP – Page_5F_MAP	Store the ASC and Temperature Monitor (TMON) Mapping	0x00 to 0xFF

I/O Ports

Table 23 describes the input and output ports of the LatticeMico PMBus Adapter.

Table 23: PMBus Adapter I/O Ports

I/O Port	Direction	Active	Description
System Clock and Reset			
CLK_I	I	—	WISHBONE System Clock
LGB_3WI_CLK_I	I	—	LogiBuilder 3Wire Clock (8 MHz)
LGB_INT_CLK_I	I	—	Logibuilder Clock (62.5 kHz)
RST_I	I	High	System Reset
IRQ_O	O	—	Interrupt Signal
PMBus			
SMBCLK	I/O	—	PMBus clock signal
SMBDAT	I/O	—	PMBus data signal
SMBALERT	O	Low	Indicated error happened in the system or user defined error from LogiBuilder

Table 23: PMBus Adapter I/O Ports (Continued)

I/O Port	Direction	Active	Description
PSI WISHBONE Slave Signal			
PSI_CYC_I	I	High	Indicates a valid bus cycle is present on the bus.
PSI_STB_I	I	High	Asserts an acknowledgment in response to the assertion of the WISHBONE Master strobe.
PSI_WE_I	I	—	Level sensitive Write/Read control signal. Low - Read operation, High - Write operation
PSI_ADR_I	I	—	32-bit wide address used to select a specific register
PSI_DAT_I	I	—	8-bit data used to read a byte of data from a specific register
PSI_CTI_I	I	—	Not used, always tied to 0
PSI_BTE_I	I	—	Not used, always tied to 0
PSI_LOCK_I	I	—	Not used, always tied to 0
PSI_SEL_I	I	—	Not used, always tied to 0
PSI_DAT_O	O	—	8-bit data used to read a byte of data from a specific register
PSI_ACK_O	O	High	Indicates the requested transfer is acknowledged.
PSI_ERR_O	O	—	Indicates the address is incorrect
PSI_RTY_O	O	—	Not used, always tied to 0
LMI WISHBONE Slave Signal			
LMI _CYC_I	I	High	Indicates a valid bus cycle is present on the bus.
LMI _STB_I	I	High	Asserts an acknowledgment in response to the assertion of the WISHBONE Master strobe.
LMI _WE_I	I	—	Level sensitive Write/Read control signal. Low - Read operation, High - Write operation
LMI _ADR_I	I	—	32-bit wide address used to select a specific register
LMI _DAT_I	I	—	8-bit data used to read a byte of data from a specific register

Table 23: PMBus Adapter I/O Ports (Continued)

I/O Port	Direction	Active	Description
LMI _CTI_I	I	—	Not used, always tied to 0
LMI _BTE_I	I	—	Not used, always tied to 0
LMI _LOCK_I	I	—	Not used, always tied to 0
LMI _SEL_I	I	—	Not used, always tied to 0
LMI _DAT_O	O	—	8-bit data used to read a byte of data from a specific register
LMI _ACK_O	O	High	Indicates the requested transfer is acknowledged.
LMI _ERR_O	O	—	Indicates the address is incorrect
LMI _RTY_O	O	—	Not used, always tied to 0
LogiBuilder Ports			
PMBUS_STATUS_BYTE_BUSY	I	High	This is the bit7 of Status Word – Byte0 and also bit7 of Status Byte from LogiBuilder.
PMBUS_STATUS_BYTE_OFF	I	High	This is the bit6 of Status Word – Byte0 and also bit6 of Status Byte from LogiBuilder.
PMBUS_STATUS_BYTE_VOUT_OV	I	High	This is the bit5 of Status Word – Byte0 and also bit5 of Status Byte from LogiBuilder.
PMBUS_STATUS_BYTE_IOUT_OC	I	High	This is the bit4 of Status Word – Byte0 and also bit4 of Status Byte from LogiBuilder.
PMBUS_STATUS_BYTE_VIN_UV	I	High	This is the bit3 of Status Word – Byte0 and also bit3 of Status Byte from LogiBuilder.
PMBUS_STATUS_BYTE_TEMPERATURE	I	High	This is the bit2 of Status Word – Byte0 and also bit2 of Status Byte from LogiBuilder.
PMBUS_STATUS_BYTE_CML	I	High	This is the bit1 of Status Word – Byte0 and also bit1 of Status Byte from LogiBuilder.
PMBUS_STATUS_BYTE_NONE_OF_ABOVE	I	High	This is the bit0 of Status Word – Byte0 and also bit0 of Status Byte from LogiBuilder.
PMBUS_STATUS_WORD_VOUT_OV	I	High	This is the bit7 of Status Word – Byte1 from LogiBuilder.
PMBUS_STATUS_WORD_IOUT_POUT	I	High	This is the bit6 of Status Word – Byte1 from LogiBuilder.
PMBUS_STATUS_WORD_INPUT	I	High	This is the bit5 of Status Word – Byte1 from LogiBuilder.

Table 23: PMBus Adapter I/O Ports (Continued)

I/O Port	Direction	Active	Description
PMBUS_STATUS_WORD_MFR	I	High	This is the bit4 of Status Word – Byte1 from LogiBuilder.
PMBUS_STATUS_WORD_POWER_GOOD	I	High	This is the bit3 of Status Word – Byte1 from LogiBuilder.
PMBUS_STATUS_WORD_FANS	I	High	This is the bit2 of Status Word – Byte1 from LogiBuilder.
PMBUS_STATUS_WORD_OTHER	I	High	This is the bit1 of Status Word – Byte1 from LogiBuilder.
PMBUS_STATUS_WORD_UNKNOWN	I	High	This is the bit0 of Status Word – Byte1 from LogiBuilder.
PMBUS_STATUS_VOUT_OV_FAULT	I	High	This is the bit7 of Status VOUT Byte from LogiBuilder.
PMBUS_STATUS_VOUT_OV_WARNING	I	High	This is the bit6 of Status VOUT Byte from LogiBuilder.
PMBUS_STATUS_VOUT_UV_WARNING	I	High	This is the bit5 of Status VOUT Byte from LogiBuilder.
PMBUS_STATUS_VOUT_UV_FAULT	I	High	This is the bit4 of Status VOUT Byte from LogiBuilder.
PMBUS_STATUS_VOUT_MAX_WARNING	I	High	This is the bit3 of Status VOUT Byte from LogiBuilder.
PMBUS_STATUS_VOUT_TON_MAX_FAULT	I	High	This is the bit2 of Status VOUT Byte from LogiBuilder.
PMBUS_STATUS_VOUT_TOFF_MAX_WARNING	I	High	This is the bit1 of Status VOUT Byte from LogiBuilder.
PMBUS_STATUS_VOUT_POWER_ON_TRACKING_ERROR	I	High	This is the bit0 of Status VOUT Byte from LogiBuilder.
PMBUS_STATUS_IOUT_OC_FAULT	I	High	This is the bit7 of Status IOUT Byte from LogiBuilder.
PMBUS_STATUS_IOUT_OC_FAULT_LV_SHUTDOWN	I	High	This is the bit6 of Status IOUT Byte from LogiBuilder.
PMBUS_STATUS_IOUT_OC_WARNING	I	High	This is the bit5 of Status IOUT Byte from LogiBuilder.
PMBUS_STATUS_IOUT_UC_FAULT	I	High	This is the bit4 of Status IOUT Byte from LogiBuilder.
PMBUS_STATUS_IOUT_CURRENT_SHARE_FAULT	I	High	This is the bit3 of Status IOUT Byte from LogiBuilder.
PMBUS_STATUS_IOUT_IN_POWER_LIMITING_MODE	I	High	This is the bit2 of Status IOUT Byte from LogiBuilder.
PMBUS_STATUS_IOUT_POUT_OP_FAULT	I	High	This is the bit1 of Status IOUT Byte from LogiBuilder.

Table 23: PMBus Adapter I/O Ports (Continued)

I/O Port	Direction	Active	Description
PMBUS_STATUS_IOUT_POUT_OP_WARNING	I	High	This is the bit0 of Status IOUT Byte from LogiBuilder.
PMBUS_STATUS_TEMP_OT_FAULT	I	High	This is the bit7 of Status Temperature Byte from LogiBuilder.
PMBUS_STATUS_TEMP_OT_WARNING	I	High	This is the bit6 of Status Temperature Byte from LogiBuilder.
PMBUS_STATUS_TEMP_UT_WARNING	I	High	This is the bit5 of Status Temperature Byte from LogiBuilder.
PMBUS_STATUS_TEMP_UT_FAULT	I	High	This is the bit4 of Status Temperature Byte from LogiBuilder.
PMBUS_STATUS_TEMP_RESERVED_3	I	High	This is the Reserved bit of Status Temperature Byte from LogiBuilder.
PMBUS_STATUS_TEMP_RESERVED_2	I	High	This is the Reserved bit of Status Temperature Byte from LogiBuilder.
PMBUS_STATUS_TEMP_RESERVED_1	I	High	This is the Reserved bit of Status Temperature Byte from LogiBuilder.
PMBUS_STATUS_TEMP_RESERVED_0	I	High	This is the Reserved bit of Status Temperature Byte from LogiBuilder.
PMBUS_STATUS_OTHER_RESERVED_7	I	High	This is the Reserved bit of Status Other Byte from LogiBuilder.
PMBUS_STATUS_OTHER_RESERVED_6	I	High	This is the Reserved bit of Status Other Byte from LogiBuilder.
PMBUS_STATUS_OTHER_INPUT_A_FUSEBREAKER_FAULT	I	High	This is the bit5 of Status Other Byte from LogiBuilder.
PMBUS_STATUS_OTHER_INPUT_B_FUSEBREAKER_FAULT	I	High	This is the bit4 of Status Other Byte from LogiBuilder.
PMBUS_STATUS_OTHER_INPUT_A_OR_ING_DEVICE_FAULT	I	High	This is the bit3 of Status Other Byte from LogiBuilder.
PMBUS_STATUS_OTHER_INPUT_B_OR_ING_DEVICE_FAULT	I	High	This is the bit2 of Status Other Byte from LogiBuilder.
PMBUS_STATUS_OTHER_OUTPUT_OR_ING_DEVICE_FAULT	I	High	This is the bit1 of Status Other Byte from LogiBuilder.
PMBUS_STATUS_OTHER_RESERVED_0	I	High	This is the Reserved bit of Status Other Byte from LogiBuilder.
PMBUS_STATUS_INPUT_VIN_OV_FAULT	I	High	This is the bit7 of Status Inputs Byte from LogiBuilder.
PMBUS_STATUS_INPUT_VIN_OV_WARNING	I	High	This is the bit6 of Status Inputs Byte from LogiBuilder.
PMBUS_STATUS_INPUT_VIN_UV_FAULT	I	High	This is the bit5 of Status Inputs Byte from LogiBuilder.

Table 23: PMBus Adapter I/O Ports (Continued)

I/O Port	Direction	Active	Description
PMBUS_STATUS_INPUT_VIN_UV_WARNING	I	High	This is the bit4 of Status Inputs Byte from LogiBuilder.
PMBUS_STATUS_INPUT_UNIT_OFF_FOR_LOW _INPUT_VOLTAGE	I	High	This is the bit3 of Status Inputs Byte from LogiBuilder.
PMBUS_STATUS_INPUT_IN_OC_FAULT	I	High	This is the bit2 of Status Inputs Byte from LogiBuilder.
PMBUS_STATUS_INPUT_IN_OC_WARNING	I	High	This is the bit1 of Status Inputs Byte from LogiBuilder.
PMBUS_STATUS_INPUT_PIN_OP_WARNING	I	High	This is the bit0 of Status Inputs Byte from LogiBuilder.
PMBUS_STATUS_MFR_SPECIFIC_7	I	High	This is the bit7 of Status Manufacturer Specifics Byte from LogiBuilder.
PMBUS_STATUS_MFR_SPECIFIC_6	I	High	This is the bit6 of Status Manufacturer Specifics Byte from LogiBuilder.
PMBUS_STATUS_MFR_SPECIFIC_5	I	High	This is the bit5 of Status Manufacturer Specifics Byte from LogiBuilder.
PMBUS_STATUS_MFR_SPECIFIC_4	I	High	This is the bit4 of Status Manufacturer Specifics Byte from LogiBuilder.
PMBUS_STATUS_MFR_SPECIFIC_3	I	High	This is the bit3 of Status Manufacturer Specifics Byte from LogiBuilder.
PMBUS_STATUS_MFR_SPECIFIC_2	I	High	This is the bit2 of Status Manufacturer Specifics Byte from LogiBuilder.
PMBUS_STATUS_MFR_SPECIFIC_1	I	High	This is the bit1 of Status Manufacturer Specifics Byte from LogiBuilder.
PMBUS_STATUS_MFR_SPECIFIC_0	I	High	This is the bit0 of Status Manufacturer Specifics Byte from LogiBuilder.
PMBUS_STATUS_FANS_1_2_FAN_1_FAULT	I	High	This is the bit7 of Status Fans 1-2 Byte from LogiBuilder.
PMBUS_STATUS_FANS_1_2_FAN_2_FAULT	I	High	This is the bit6 of Status Fans 1-2 Byte from LogiBuilder.
PMBUS_STATUS_FANS_1_2_FAN_1_WARNING	I	High	This is the bit5 of Status Fans 1-2 Byte from LogiBuilder.
PMBUS_STATUS_FANS_1_2_FAN_2_WARNING	I	High	This is the bit4 of Status Fans 1-2 Byte from LogiBuilder.
PMBUS_STATUS_FANS_1_2_FAN_1_SPEED_OVERRIDE	I	High	This is the bit3 of Status Fans 1-2 Byte from LogiBuilder.
PMBUS_STATUS_FANS_1_2_FAN_2_SPEED_OVERRIDE	I	High	This is the bit2 of Status Fans 1-2 Byte from LogiBuilder.
PMBUS_STATUS_FANS_1_2_AIR_FLOW_FAULT	I	High	This is the bit1 of Status Fans 1-2 Byte from LogiBuilder.

Table 23: PMBus Adapter I/O Ports (Continued)

I/O Port	Direction	Active	Description
PMBUS_STATUS_FANS_1_2_AIR_FLOW_WARNING	I	High	This is the bit0 of Status Fans 1-2 Byte from LogiBuilder.
PMBUS_STATUS_FANS_1_2_FAN_3_FAULT	I	High	This is the bit7 of Status Fans 3-4 Byte from LogiBuilder.
PMBUS_STATUS_FANS_1_2_FAN_4_FAULT	I	High	This is the bit6 of Status Fans 3-4 Byte from LogiBuilder.
PMBUS_STATUS_FANS_1_2_FAN_3_WARNING	I	High	This is the bit5 of Status Fans 3-4 Byte from LogiBuilder.
PMBUS_STATUS_FANS_1_2_FAN_4_WARNING	I	High	This is the bit4 of Status Fans 3-4 Byte from LogiBuilder.
PMBUS_STATUS_FANS_1_2_FAN_3_SPEED_OVERRIDE	I	High	This is the bit3 of Status Fans 3-4 Byte from LogiBuilder.
PMBUS_STATUS_FANS_1_2_FAN_4_SPEED_OVERRIDE	I	High	This is the bit2 of Status Fans 3-4 Byte from LogiBuilder.
PMBUS_STATUS_FANS_3_4_RESERVED_1	I	High	This is the Reserved bit of Status Fans 3-4 Byte from LogiBuilder.
PMBUS_STATUS_FANS_3_4_RESERVED_0	I	High	This is the Reserved bit of Status Fans 3-4 Byte from LogiBuilder.
PMBUS_IMMED_OFF	O	High	Active when PMBus adapter receives OPERATION IMMED_OFF command
PMBUS_SOFT_OFF	O	High	Active when PMBus adapter receives OPERATION SOFT_OFF command
PMBUS_MARGIN_OFF	O	High	Active when PMBus adapter receives OPERATION MARGIN_OFF command
PMBUS_MARGIN_LOW_IF	O	High	Active when PMBus adapter receives OPERATION MARGIN_LOW_IF command
PMBUS_MARGIN_LOW_IF	O	High	Active when PMBus adapter receives OPERATION MARGIN_LOW_IF command
PMBUS_MARGIN_LOW_IF	O	High	Active when PMBus adapter receives OPERATION MARGIN_LOW_IF command
PMBUS_MARGIN_LOW_IF	O	High	Active when PMBus adapter receives OPERATION MARGIN_LOW_IF command
PMBUS_CLEAR_FAULTS	O	High	Active when PMBus adapter receives CLEAR FAULTS command
PMBUS_INTERLEAVE_ON_OFF	O	—	Identify the MFR_INTERLEAVE_ON or MFR_INTERLEAVE_OFF command. When this bit is High, the interleave is on; when it is Low, the interleave is off

Table 23: PMBus Adapter I/O Ports (Continued)

I/O Port	Direction	Active	Description
PMBUS_PAGE_BYTE_7	O	High	The bit7 of Page Byte showing the current page number.
PMBUS_PAGE_BYTE_6	O	High	The bit6 of Page Byte showing the current page number.
PMBUS_PAGE_BYTE_5	O	High	The bit5 of Page Byte showing the current page number.
PMBUS_PAGE_BYTE_4	O	High	The bit4 of Page Byte showing the current page number.
PMBUS_PAGE_BYTE_3	O	High	The bit3 of Page Byte showing the current page number.
PMBUS_PAGE_BYTE_2	O	High	The bit2 of Page Byte showing the current page number.
PMBUS_PAGE_BYTE_1	O	High	The bit1 of Page Byte showing the current page number.
PMBUS_PAGE_BYTE_0	O	High	The bit0 of Page Byte showing the current page number.
Other signals			
PMBUS_SMBALERT	i	High	This edge-sensitive signal is from LogiBuilder to the LatticeMico8 microcontroller. Users can trigger this signal to pull low the SMBALERT on PMBus.

Register Descriptions

The LatticeMico PMBus Adapter WISHBONE module has a register map to allow the service of the hardened functions through the WISHBONE bus interface read/write operations. Table 24 describes the register map of the PMBus module.

Table 24: WISHBONE Addressable Registers for PMBus Adapter Module

Register Name	Register Function	Address	Access
PSI			
RDAT	3-byte-deep FIFO contains data bytes received over the SMBus to be delivered to Mico	0x00	Read
TDAT	2-byte-deep FIFO contains data to be transmitted to the PMBus Master	0x01	Write
CTL0	Configure or control resigster0 of PSI logic	0x02	Read/Write
CTL1	Configure or control resigster1 of PSI logic	0x03	Read/Write
STA0	Status resigster0 of PSI	0x04	Read

Table 24: WISHBONE Addressable Registers for PMBus Adapter Module (Continued)

Register Name	Register Function	Address	Access
STA1	Status resigster1 of PSI	0x05	Read
CAP	capability information register	0x06	Read
ERRS	detailed error status information register	0x07	Read
LMI			
PAGE_BYTE	Current PAGE number	0x000	Write
OPERATION	Operation command control register	0x001	Write
CLR_FAULTS	Clean fault command control register	0x003	Write
STATUS_WORD_0	Information of Status_Byte/Status_Word_Byte0	0x078	Read
STATUS_WORD_1	Information of Status_Word_Byte1	0x079	Read
STATUS_VOUT	Information of Status_VOUT	0x07A	Read
STATUS_IOUT	Information of Status_IOUT	0x07B	Read
STATUS_INPUT	Information of Status_INPUT	0x07C	Read
STATUS_TEMP	Information of Status_TEMP	0x07D	Read
STATUS_OTHER	Information of Status_OTHER	0x07F	Read
STATUS_MFR	Information of Status_MFR	0x080	Read
STATUS_FAN_12	Information of Status_FAN_12	0x081	Read
STATUS_FAN_34	Information of Status_FAN34	0x082	Read
INTLV_ON_OFF	MFR_Interleave_ON/OFF control	0x0D0	Write
ASC_ADDR	Base address of ASC0	0x0D4	Read
RDAT	3-byte-deep FIFO contains data bytes received over the SMBus to be delivered to Mico	0x00	Read
TDAT	2-byte-deep FIFO contains data to be transmitted to the PMBus Master	0x01	Write
Page_xx_MAP	Page Mapping information registers	0x100 – 0x15F	Read

Note: xx = 0x00 – 0x5F

Following section provides details about each register in the LatticeMico PMBus Adapter.

RX Data FIFO – RDAT

This 3-byte-deep FIFO contains data bytes received over the SMBus to be delivered to Mico. The first data byte of a transfer is a command code. The PSI logic checks if the command is supported by the IP before storing it into the FIFO. The PSI logic NACKs the transaction if the command is not supported and discards all data bytes associated with the transfer. The

PMBus IP MRD specifies the commands the IP is required to support. The FIFO's live empty and full flags can be read from the Status Register; the LatticeMico8 microcontroller needs to monitor these bits to determine when to read the FIFO. If interrupt is enabled then the PSI logic generates the interrupt output corresponding to each status bit. If the FIFO is not empty when a new transfer is started then the main FSM NACKs the transaction and waits for STOP.

The LatticeMico8 microcontroller can reset this FIFO by setting the bit "RDAT reset" in the Control Register.

TX Data FIFO – TDAT

This 2-byte-deep FIFO contains data to be transmitted to the SMBus Master. The FIFO's live empty and full flags can be read from the Status Register. The main FSM monitors these bits to determine when data is available for transmit. If the FIFO is empty when data needs to be put on the SMBDAT line then the FSM sends all ones until STOP is detected.

The LatticeMico8 microcontroller can reset this FIFO by setting the bit "TDAT reset" in the Control Register. The main FSM resets this FIFO when a START condition is detected.

Control Register 0 – CTL0

This register contains the bits that the LatticeMico8 microcontroller can program to configure or control the PSI logic.

Table 25: CTL0 Register Bit Definition

Bits	Field	Description	Access
[7:3]	Reserved	Reserved	N/A
[2]	RDAT Reset (RRST)	0 = no-op 1 = reset RX Data FIFO This bit is set for one WB clock, i.e. not sticky	Read/Write
[1]	TDAT Reset (TRST)	0 = no-op 1 = reset TX Data FIFO This bit is set for one WB clock, i.e. not sticky	Read/Write
[0]	Sync Reset (SRST)	1 = reset the entire PSI block This bit is set for one WB clock, i.e. not sticky	Read/Write

Control Register 1 – CTL1

This register contains the bits that the LatticeMico8 microcontroller can program to configure or control the PSI logic.

Table 26: CTL1 Register Bit Definition

Bits	Field	Description	Access
[7:6]	Reserved	Reserved	N/A
[5]	Clock Low Extend (CKEX)	The LatticeMico8 microcontroller sets this bit to 1 when it wants the PSI to hold the SMBCLK line low. The PSI stops holding the clock low when the LatticeMico8 microcontroller resets this bit or if an SMBus Timeout condition has been detected.	Read/Write
[4]	Send Alert (ALRT)	The LatticeMico8 microcontroller sets this bit to 1 when it wants to notify the host of a fault. Refer to the section SMBALERT# and Alert Response Address for more details.	Read/Write
[3:1]	Reserved	Reserved	N/A
[0]	Interrupt Enable (IEN)	0 = interrupt is disabled 1 = interrupt is enabled	Read/Write

Status Register 0 – STA0

This register contains the status of the PSI logic. All bits are read-only. The bits in this register can be cleared by the SRST. Some bits have additional specific reset conditions.

Table 27: STA0 Register Bit Definition

Bits	Field	Description	Access
[7]	SMBus Timeout (TOUT)	This bit is set when the SCL line has been low for greater than the SMbus timeout duration and reset when a START condition is detected. The PSI logic resets the FIFOs and all status information associated with the data transfer when a timeout is detected.	Read
[6:5]	Data Transmission Error (ERR)	This bit is set when a fault has been detected on a data transfer. The LatticeMico8 microcontroller needs to monitor this bit to validate the data in RDAT before processing. The bit is reset when a START condition is detected. Refer to the section "Data Transmission Fault" for more details.	Read
[4]	Master's NACK (MNAK)	The PSI logic updates this bit every time it receives a NACK from the Master and clears it if the next response is an ACK or when a START condition is detected.	Read
[3]	Master's ACK (MAK)	The PSI logic updates this bit every time it receives an ACK from the Master and clears it if the next response is a NACK or when a START condition is detected.	Read

Table 27: STA0 Register Bit Definition (Continued)

Bits	Field	Description	Access
[2]	Read Command Valid (RC_VLD)	This is the Read command Valid status bit. It is set when the read command bit in STA1 register (STA1_RD_CMD) is set by the slave. The slave sets the read command as well as read command valid as soon as it knows the received command is a read or write command. This bit is cleared when a STOP condition is detected. 0 = Read command bit is not set. 1 = Read command bit has been set and valid.	Read
[1]	Group Command (GRP)	This bit is set when START is received instead of a STOP at the end of a write command cycle signifying a GROUP command. This bit is reset when a STOP condition is detected. The LatticeMico8 microcontroller needs to monitor this bit so that it won't execute the received command until STOP is detected.	Read
[0]	Bus Busy (BBS)	This bit indicates the status of the SMBus; it's set when a START condition is detected and reset when a STOP condition is detected. 0 = bus is idle 1 = bus is busy	Read

Status Register 1 – STA1

This register contains the status of the PSI logic. All bits are read-only. The bits in this register can be cleared by the SRST. Some bits have additional specific reset conditions.

Table 28: STA1 Register Bit Definition

Bits	Field	Description	Access
[7]	Read Command (RD_CMD)	This bit is set to 1 if the currently serviced command is a read command This is set when a START is received after the command code and reset at the following STOP condition.	Read
[6]	Read/ Write Word Command (RW_WORD)	This bit is set if the currently serviced command is a write word or read word command This bit is assigned after the PMBus slave receives a command code and reset when a STOP is received or when the main state machine goes to IDLE.	Read
[5]	Read/ Write Byte Command (RW_BYTE)	This bit is set if the currently serviced command is a write byte or read byte command This bit is assigned after the PMBus slave receives a command code and reset when a STOP is received or when the main state machine goes to IDLE.	Read

Table 28: STA1 Register Bit Definition (Continued)

Bits	Field	Description	Access
[4]	Receive / Send Byte Command (RS_BYTE)	This bit is set if the currently serviced command is a send byte or receive byte command This bit is assigned after the PMBus slave receives a command code and reset when a STOP is received or when the main state machine goes to IDLE.	Read
[3]	TFIFO Full (TFUL)	This is the TXData FIFO's live full flag. 0 = not full 1 = full	Read
[2]	RFIFO Full (RFUL)	This is the RXData FIFO's live full flag. 0 = not full 1 = full	Read
[1]	TFIFO Empty (TMTY)	This is the TXData FIFO's live empty flag. 0 = not empty 1 = empty	Read
[0]	RFIFO Empty (RMTY)	This is the RXData FIFO's live empty flag. 0 = not empty 1 = empty	Read

Capability Register – CAP

This register holds the capability information of the PMBus device. This register value can be returned as is when PMBus host issues a Capability Read command.

Table 29: Capability Register – CAP Bit Definition

Bits	Field	Description	Access
[7]	Packet Error Checking	Packet Error Checking support. 0 = Packet Error Checking not supported. 1 = Packet Error Checking is supported.	Read
[6:5]	Maximum Bus Speed	Maximum bus speed supported by the device. 00 = Maximum supported bus speed is 100 kHz. 01 = Maximum supported bus speed is 400 kHz. 10 = Reserved. 11 = Reserved.	Read

Table 29: Capability Register – CAP Bit Definition (Continued)

Bits	Field	Description	Access
[4]	SMBALERT#	SMBALERT support b y the device. 0 = The device does not have a SMBALERT# pin and does not support the SMBus Alert Response protocol. 1 = The device does have a SMBALERT# pin and does support the SMBus Alert Response protocol.	Read
3:0]	Reserved	Reserved	N/A

Error Status – ERRS

This register holds the detailed error status information from the PSI logic. All bits are read-only.

Table 30: Error Status Register Bit Definition

Bits	Field	Description	Access
[7]	UCC	**Unsupported command code.** This bit is set when the device receives a command code not supported by it. This bit is set as soon as an unsupported command is received and is reset at the following START condition.	Read
[6]	IRB	**Improperly set read bit in the address byte.** The slave address sent at the start of a transaction must have its Read/Write bit set to 0 for writing. This bit is set when the device receives its own slave address at the start of the transaction with the Read/Write bit set to 1 (for reading). This bit is reset at the following START condition.	Read
[5]	DEVBC	**Device busy.** This bit is set when PSI logic is busy and cannot successfully receive or transmit data. This condition arises when (RFUL or TMTY) and clock stretching is not enabled or when PSI is stretching the clock waiting for the FIFO and meets a timeout. The bit is reset at the following START condition.	Read
[4]	RTMBY	**Host reading too many bytes.** This bit is set when the PSI logic determines this condition and is reset at the following START condition.	Read
[3]	STMBY	**Host sending too many bytes.** This bit is set when the PSI logic determines this condition and is reset at the following START condition.	Read
[2]	SRTFBY	**Host sending or reading too few bytes.** This bit is set when the PSI logic determines these conditions and is reset at the following START condition.	Read

Table 30: Error Status Register Bit Definition (Continued)

Bits	Field	Description	Access
[1]	RTFBI	**Host reading too few bits.** This bit is set when the PSI logic determines this condition and is reset at the following START condition.	Read
[0]	STFBI	**Host sending too few bits.** This bit is set when the PSI logic determines this condition and is reset at the following START condition.	Read

PAGE_BYTE

PAGE_BYTE is an 8-bit register that specifics the current page number.

OPERATION

OPERATION is an 8-bit register that specifics current operation information. Table 31 showing the corresponding operation and its value range. “Margin Low (Act on Fault)” means that if a under voltage/current/temperature is detected, the unit treats this as a fault and responds by a fault response command (or send a PMBALERT to the PMBus Master). Similarly, “Margin High (Act on Fault)” means if an over voltage/current/temperature is treated as a fault.

“Ignore Fault” means that any voltage/current/temperature fault is detected will be ignored and the unit ignores the condition and continues to operate.

Table 31: Operation Register Definition

Bit[7:6]	Bit[5:4]	Bit[3:2]	Bit[1:0]	Unit	Margin
00	XX	XX	XX	Immediate Off	N/A
01	XX	XX	XX	Soft Off	N/A
10	00	XX	XX	On	Off
10	01	01	XX	On	Margin Low (Ignore Fault)
10	01	10	XX	On	Margin Low (Act On Fault)
10	10	01	XX	On	Margin High (Ignore Fault)
10	10	10	XX	On	Margin High (Act On Fault)

CLR_FAULTS

CLR_FAULTS is an 8-bit register that specifics and controls the clean fault status and CML byte. When the LatticeMico PMBus Adapter acknowledges the LatticeMico8 microcontroller the clean fault command were delivered from PMBus host, the LatticeMico8 microcontroller will write 1 to CLR_FAULTS register, then follow by 0 to trigger clean fault activity. Once the clear fault be trigger, the fault bits have been set in the LatticeMico8 microcontroller will be cleared and SMBAlert will be reset.

STATUS_WORD_0

STATUS_WORD_0 is an 8-bit register that specifics the current information with a summary of the most critical faults/warnings. This register also represents the information of STATUS_BYTE.

Table 32: Status_WORD_0 Register Definition

Bit	Status
7	Busy
6	OFF
5	VOUT_OV
4	IOUT_OC
3	VIN_UV
2	TEMPERATURE
1	CML (Handled by LatticeMico8 microcontroller)
0	NONE OF THE ABOVE

STATUS_WORD_1

STATUS_WORD_1 is an 8-bit register that specifics the other current information with a summary of the most critical faults/warnings.

Table 33: Status_WORD_1 Register Definition

Bit	Status
7	VOUT
6	IOUT/POUT
5	MFR
4	POWER_GOOD
3	POWER_GOOD#
2	FANS

Table 33: Status_WORD_1 Register Definition

Bit	Status
1	OTHERS
0	UNKNOWN

STATUS_VOUT

STATUS_VOUT is an 8-bit register that specifics the currently VOUT-associated information with the content as follows.

Table 34: Status_VOUT Register Definition

Bit	Status
7	VOUT_OV_Fault
6	VOUT_OV_Warning
5	VOUT_UV_Fault
4	VOUT_UV_Warning
3	VOUT_MAX_Warning
2	TON MAX Fault
1	TOff Max Warning
0	Power On Tracking Error

STATUS_IOUT

STATUS_IOUT is an 8-bit register that specifics the currently IOUT-associated information with the content as follows.

Table 35: Status_IOUT Register Definition

Bit	Status
7	IOUT_OC_Fault
6	IOUT_OC_Fault w/ LV Shutdown
5	IOUT_OC_Warning
4	IOUT_UC_Fault
3	Current Share Fault
2	In Power Limiting Mode
1	POUT OP Fault
0	POUT OP Warning

STATUS_TEMPERATURE

STATUS_TEMPERATURE is an 8-bit register that specifics the currently Temperature-associated information with the content as follows.

Table 36: Status_Temperature Register Definition

Bit	Status
7	OT_Fault
6	OT_Warning
5	UT_Warning
4	UT_Fault
3	Reserved
2	Reserved
1	Reserved
0	Reserved

STATUS_CML

STATUS_CML is an 8-bit register that specifics the currently CML-associated information with the content as follows.

Table 37: Status_CML Register Definition

Bit	Status
7	Invalid/Unsupported Command
6	Invalid/Unsupported Data
5	Packet Error Check Failed
4	Memory Fault Detected
3	Processor Fault Detected
2	Reserved
1	Other Communication Fault
0	Other Memory or Logic Fault

STATUS_OTHER

STATUS_OTHER is an 8-bit register that specifics the currently other fault information with the content as follows.

Table 38: Status_OTHER Register Definition

Bit	Status
7	Reserved
6	Reserved
5	Input A Fuse/Breaker Fault
4	Input B Fuse/Breaker Fault
3	Input A OR-ing DeviceFault
2	Input B OR-ing DeviceFault
1	Output OR-ing DeviceFault
0	Reserved

STATUS_INPUT

STATUS_INPUT is an 8-bit register that specifics the currently INPUT-associated information with the content as follows.

Table 39: Status_INPUT Register Definition

Bit	Status
7	VIN OV Fault
6	VIN OV Warning
5	VIN UV Fault
4	VIN UV Warning
3	Unit Off for Low Input Voltage
2	IN OC Fault
1	IN OC Warning
0	PIN OP Warning

STATUS_MFR

STATUS_MFR is an 8-bit register that specifics the currently manufacture-associated information with the content as follows.

Table 40: Status_MFR Register Definition

Bit	Status
7	Manufacturer Defined
6	Manufacturer Defined
5	Manufacturer Defined
4	Manufacturer Defined
3	Manufacturer Defined
2	Manufacturer Defined
1	Manufacturer Defined
0	Manufacturer Defined

STATUS_FAN_12

STATUS_FAN_12 is an 8-bit register that specifics the currently fan12-associated information with the content as follows.

Table 41: Status_FAN_12 Register Definition

Bit	Status
7	Fan 1 Fault
6	Fan 2 Fault
5	Fan 1 Warning
4	Fan 2 Warning
3	Fan 1 Speed Override
2	Fan 2 Speed Override
1	Air Flow Fault
0	Air Flow Warning

STATUS_FAN_34

STATUS_FAN_34 is an 8-bit register that specifics the currently fan34-associated information with the content as follows.

Table 42: Status_FAN_34 Register Definition

Bit	Status
7	Fan 3 Fault
6	Fan 4 Fault
5	Fan 3 Warning
4	Fan 4 Warning
3	Fan 3 Speed Override
2	Fan 4 Speed Override
1	Reserved
0	Reserved

INTLV_ON_OFF

INTLV_ON_OFF is an 8-bit register that specifics the current MFR_Interleave_ON/OFF information. The LatticeMico8 microcontroller can control interleave ON/OFF by write 1/0 to this register.

ASC_ADDR

ASC_ADDR is an 8-bit register that specifics the current ASC0 slave address.

Page_xx_Map

Page_xx_Map is a set of registers for Page 0x00 to Page 0x5F storing the page mapping for each VMON/IMON/TMON.

Table 43: Page_xx_Map Register Definition

Bit	Allowable Value	Description
[7]	-	Reserved
[6:4]	0-7	Specific which ASC is pointing to
[3:0]	0-15	Specific which the register is pointing to. (See table below.)

Table 44: PAGE_MAP Bit [t3:0] Definition

Bit [3:0]	ASC Register
0x0	VMON1
0x1	VMON2
0x2	VMON3
0x3	VMON4
0x4	VMON5
0x5	VMON6
0x6	VMON7
0x7	VMON8
0x8	VMON9
0x9	HVMON
0xA	IMON
0xB	HIMON
0xC	TMON1
0xD	TMON2
0xE	TMONint

- For Example:

  User set the ASC2 VMON3 to Page 0x00

  Then the PAGE_00_MAP = 0x13

LatticeMico8 Microcontroller Software Support

This section describes the LatticeMico8 microcontroller software support provided for the LatticeMico PMBus Adapter component.

Device Driver

The PMBus Adapter device driver interacts directly with the PMBus Adapter instance. This section describes the limitations, type definitions, structure, and functions of the PMBus Adapter device driver.

Type Definitions

This section describes the type definitions for the PMBus Adapter device context structure. This structure, shown in Figure 3, contains the PMBus Adapter component instance-specific information and is dynamically generated in the DDStructs.h header file. This information is largely filled in by the managed build process by extracting the PMBus Adapter component-specific information from the platform specification file. As part of the managed build process, designers can choose to control the size of the generated structure, and hence the software executable, by selectively enabling some of the elements in this structure via C preprocessor macro definitions. These C preprocessor macro definitions are explained later in this document. You should not manipulate the members directly, because this structure is for exclusive use by the device driver. Table 42 describes the parameters of the PMBus Adapter device context structure shown in Figure 3.

Device Context Structure

Figure 3 shows the PMBus Adapter device context structure.

Figure 3: PMBus Adapter Device Context Structure

```
struct st_MicoPMBUSCtx_t {
    const char *   name;
    size_t   psi_base;
    size_t   lmi_base;
    unsigned char   current_page;
    unsigned char   current_mon;
    unsigned char   current_asc_addr;
    unsigned char   write_protect;
    unsigned char   operation;
    unsigned char   cml_status;
    void *   p_efb;
    void *   p_mutex;
    unsigned char   i2c_mutex;
} MicoPMBUSCtx_t;
```

Table 45 describes the PMBus Adapter device context parameters.

Table 45: PMBus Adapter Device Context Parameters

Parameter	Data Type	Description
name	const char*	PMBus Adapter instance name (entered in MSB)
psi_base	size_t	MSB-assigned PSI base address for this instance
lmi_base	size_t	MSB-assigned LMI base address for this instance
current_page	unsigned char	This value specific the current PAGE number
current_mon	unsigned char	This value specific which VMON/IMON/TMON is pointed currently
current_asc_addr	unsigned char	This value specific current ASC slave address
write_protect	unsigned char	This value specific current write protect setting
operation	unsigned char	This value specific current operation setting

Table 45: PMBus Adapter Device Context Parameters

Parameter	Data Type	Description
cml_status	unsigned char	This value specific current CML_status information
p_efb	void *	This value points to the EFB instance used by PMBus Adapter
p_mutex	void *	This value points to the Mutex instance used by PMBus Adapter
i2c_mutex	unsigned char	This value specific to the Mutex owner ID for I^{2}C communication protocol

C Preprocessor Macro Definitions

This section describes the C preprocessor macro definitions that are available to the software developer. There are two types of macro definitions: 'object-like' and 'function-like'.

The 'object-like' macro definitions do not take any arguments and are used to control the size of the generated application executable. There are three ways an 'object-like' macro definition can be used by the software developer.

1. Manually adding the -D<macro name> option to the compiler's command line in the application's 'Build Properties'. Refer to the *LatticeMico8 Developer User Guide* for more information on how to manually add the macro definition in the application's 'Build Properties' GUI.

2. Automatically adding the -D<macro name> option to the compiler's command-line in the application's 'Build Properties' by enabling the 'check-box' associated with the macro definition. Refer to the *LatticeMico8 Developer User Guide* for more information on how to set up the check/uncheck the macro definitions in the application's 'Build Properties' GUI.

3. Manually adding the macro definition to the C code using the following syntax:

   ```
   #define <macro name>
   ```

It is recommended that the developer use option 1 or 2.

- __MICOFL_NO_SPI_INIT_VALIDATION__

  This preprocessor macro definition disables code and data structures within the device driver that disable the LatticeMico8 EFB SPI module in the software driver and application. In order words, LatticeMico8 assumes the connected SPI Flash is NOT shared with other SPI Master, and does not check whether the SPI is occupied by other SPI Master or not during the power cycle. It is not defined by default.

- __MICOFL_USER_IRQ_HANDLER__

This preprocessor macro definition disables code and data structures within the device driver that allow the user to define the custom interrupt routine, the default routine will be disabled. It is not defined by default.

Table 46: C Preprocessor Function-like Macros For PMBus Adapter

Macro Name	Second Argument to Macro / Third Argument to Macro (if exist.	Description
PSI		
MICO_PMBUS_PSI_READ_RDAT	The 8-bit value reads from the PSI RX Data FIFO register	This macro reads a character from the PSI RX Data FIFO register
MICO_PMBUS_PSI_READ_CTL0	The 8-bit value reads from the PSI Control Register 0.	This macro reads a character from the PSI Control register 0
MICO_PMBUS_PSI_READ_CTL1	The 8-bit value reads from the PSI Control Register 1.	This macro reads a character from the PSI Control register 1
MICO_PMBUS_PSI_READ_STA0	The 8-bit value reads from the PSI Status Register 0.	This macro reads a character from the PSI Status register 0
MICO_PMBUS_PSI_READ_STA1	The 8-bit value reads from the PSI Status Register 1.	This macro reads a character from the PSI Status register 1
MICO_PMBUS_PSI_READ_CAP	The 8-bit value reads from the PSI Capability register.	This macro reads a character from the PSI Capability register
MICO_PMBUS_PSI_READ_ERRS	The 8-bit value reads from the PSI Error Status register.	This macro reads a character from the PSI Error Status register
MICO_PMBUS_PSI_WRITE_TDAT	The 8-bit value writes to the PSI TX Data FIFO.	This macro writes a character to the PSI TX Data FIFO register.
MICO_PMBUS_PSI_WRITE_CTL0	The 8-bit value writes to the PSI Control Register 0.	This macro writes a character to the PSI Control Register 0.
MICO_PMBUS_PSI_WRITE_CTL1	The 8-bit value writes to the PSI Control Register 1.	This macro writes a character to the PSI Control Register 1.
LMI		
MICO_LMI_READ_REGISTER	The 8-bit value reads from the LMI register/address.	This macro reads a character from the LMI register.
MICO_LMI_READ_STATUS_BYTE	The 8-bit value reads from the LMI Status_Byte / Status_Word_0 register.	This macro reads a character from the LMI STATUS BYTE/ STATUS_WORDS-_world register.
MICO_LMI_READ_STATUS_WORD	The 8-bit value reads from the LMI STATUS_WORD_1 register.	This macro reads a character from the LMI STATUS _WORD_1 register.
MICO_LMI_READ_ASC_BASE_ADDR	The 8-bit value reads from the LMI ASC_ADDR register.	This macro reads a character from the LMI ASC_ADDR register.
MICO_LMI_WRITE_REGISTER	The 8-bit value writes to LMI register / address offset.	This macro writes a character to the LMI register with a specific address offset.

Table 46: C Preprocessor Function-like Macros For PMBus Adapter (Continued)

Macro Name	Second Argument to Macro / Third Argument to Macro (if exist.	Description
MICO_LMI_WRITE_PAGE	The 8-bit value writes to LMI PAGE register.	This macro writes a character to the LMI PAGE register.
MICO_LMI_WRITE_OPERATION	The 8-bit value writes to LMI OPERATION register.	This macro writes a character to LMI OPERATION register.
MICO_LMI_WRITE_CLEAR_FAULTS	None	This macro writes a character value 1 to the LMI CLEAR FAULT register, then write a character value 0 to this register again.
MICO_LMI_READ_PAGE_MAP	The 8-bit value read from LMI PAGE_MAP register / address offset	This macro read from LMI PAGE_MAP register with a specific address offset.

Note: The first argument to the macro is the PMBus Adapter PSI or LMI address.

Functions

This section describes the implemented device-driver-specific functions.

MicoPMBUSInit Function

```
void MicoPMBUSInit (MicoPMBUSCtx_t *ctx);
```

This is the PMBus Adapter initialization function.

Table 47 describes the parameter in the MicoPMBUSInit function syntax.

Table 47: MicoPMBUSInit Function Parameter

Parameter	Description
MicoPMBUSCtx_t	Pointer to a valid MicoPMBUSCtx _t structure representing a valid PMBUS Adapter instance.

MicoPMBUSRegisterEFBnMutex Function

```
void MicoPMBUSRegisterEFBnMutex (MicoPMBUSCtx_t *ctx,
   MicoEFBCtx_t *p_efb,
   MicoMutexCtx_t *p_mutex,
   unsigned char i2c_mutex_id);
```

This function registers an EFB and Mutex instance into the PMBus Adapter instance. This EFB and Mutex will be used for the communication between PMBus Adapter control and ASCs.

Table 48 describes the parameter in the MicoPMBUSRegisterEFB function syntax.

Table 48: MicoPMBUSRegisterEFBnMutex Function Parameter

Parameter	Description
MicoPMBUSCtx_t	Pointer to a valid MicoPMBUSCtx_t structure representing a valid PMBUS Adapter instance.
MicoEFBCtx_t	Pointer to a valid MicoEFBCtx_t structure representing a valid EFB instance.
MicoMutexCtx_t	Pointer to a valid MicoMutexCtx_t structure representing a valid Mutex instance.
unsigned char	Mutex owner ID for I^{2}C communication protocol.

MicoPMBUS_ASCStart Function

```
char MicoPMBUS_ASCStart (MicoEFBCtx_t *ctx,
                         unsigned char read,
                         unsigned char address,
                         unsigned char restart);
```

This function starts the initial communication connection between the LatticeMico8 microcontroller and ASC through EFB I^{2}C. Error code will return when the write process fails.

Table 49 describes the parameter in the MicoPMBUS_ASCStart function syntax.

Table 49: MicoPMBUS_ASCStart Function Parameter

Parameter	Description
MicoEFBCtx_t	Pointer to a valid MicoEFBCtx _t structure representing a valid EFB instance.
unsigned char	This value identifies the communication is write or read I^{2}C protocol.
unsigned char	This value identifies the slaver address of ASC.
unsigned char	This value identifies if the communication need to insert restart/ start bit.

Table 50 describes the values returned by the MicoPMBUS_ASCStart function.

Table 50: Values Returned by the MicoPMBUS_ASCStart Function

Return Value	Description
0	successful writes
-1	failed to receive ACK during addressing

Table 50: Values Returned by the MicoPMBUS_ASCStart Function

Return Value	Description
-2	failed to receive ACK when writing data
-3	arbitration lost during the operation

MicoPMBUS_WriteASC Function

```
char MicoPMBUS_WriteASC (MicoPMBUSCtx_t *ctx,
                         unsigned char buffersize,
                         unsigned char *data,
                         unsigned char insert_stop);
```

This function initiates a write I^2C transaction protocol through EFB primary I^2C channel. Error code will return when the write process is failed.

Table 51 describes the parameter in the MicoPMBUS_WriteASC function syntax.

Table 51: MicoPMBUS_WriteASC Function Parameter

Parameter	Description
MicoPMBUSCtx_t	Pointer to a valid MicoPMBUSCtx _t structure representing a valid PMBus Adapter instance.
unsigned char	This value identifies the Number of bytes to be transferred. (min 1 and max 256)
unsigned char	This value identifies the Buffer containing the data to be transferred.
unsigned char	This Value identifies if Insert Stop bit at end of data transfer.

Table 52 describes the values returned by the MicoPMBUS_WriteASC function. Error code will return when the write process is failed.

Table 52: Values Returned by the MicoPMBUS_WriteASC Function

Return Value	Description
0	Successful writes
-1	Failed to write

MicoPMBUS_ReadASC Function

```
char MicoPMBUS_ReadASC (MicoPMBUSCtx_t *ctx,
                        unsigned char buffersize,
                        unsigned char *data);
```

This function initiates a read I^2C transaction protocol through EFB primary I^2C channel. Error code will return when the write process is failed.

Table 53 describes the parameter in the MicoPMBUS_ReadASC function syntax.

Table 53: MicoPMBUS_ReadASC Function Parameter

Parameter	Description
MicoPMBUSCtx_t	Pointer to a valid MicoPMBUSCtx _t structure representing a valid PMBus Adapter instance.
unsigned char	This value identifies the Number of bytes to be transferred. (min 1 and max 256)
unsigned char	This value identifies the Buffer containing the data to be transferred.

Table 54 describes the values returned by the MicoPMBUS_ReadASC function.

Table 54: Values Returned by the MicoPMBUS_ReadASC Function

Return Value	Description
0	Successful reads
-1	Failed to read

MicoPMBUS_TxData Function

```
void MicoPMBUS_TxData (size_t psi_address,
                       unsigned char *data
                       unsigned char buffersize);
```

This function provide the communication channel for the LatticeMico8 microcontroller to write data to Tx FIFO of PMBus Adapter

Table 55 describes the parameter in the MicoPMBUS_TxData function syntax.

Table 55: MicoPMBUS_TxData Function Parameter

Parameter	Description
Size_t	PSI port base address of PMBus Adapter.
unsigned char*	This value identifies the Buffer containing the data to be transferred.
unsigned char	This value identifies the Number of bytes to be transferred. (min 0 and max 255)

MicoPMBUS_RxData Function

```
void MicoPMBUS_RxData (size_t psi_address,
                       unsigned char *data
                       unsigned char buffersize);
```

This function provide the communication channel for the LatticeMico8 microcontroller to read data from Rx FIFO of PMBus Adapter

Table 56 describes the parameter in the MicoPMBUS_TxData function syntax.

Table 56: MicoPMBUS_RxData Function Parameter

Parameter	Description
Size_t	PSI port base address of PMBus Adapter.
unsigned char*	This value identifies the Buffer containing the data to be transferred.
unsigned char	This value identifies the Number of bytes to be received. (min 0 and max 255)

MicoPMBUS_PageChecking Function

```
unsigned char MicoPMBUS_PageChecking(MicoPMBUSCtx_t *ctx,
 unsigned char current_cmd);
```

This function checks weather Read/Write fault limit and Read measurement command match to the current PAGE number. Current command has to match the associated type of monitor of current page. Error code will return when the checking result is failed.

Table 57 describes the parameter in the MicoPMBUS_PageChecking function syntax.

Table 57: MicoPMBUS_PageChecking Parameter

Parameter	Description
MicoPMBUSCtx_t	PSI port base address of PMBus Adapter.
unsigned char	This value identifies the current PMBus command handling by the LatticeMico8 microcontroller

Table 58 describes the values returned by the MicoPMBUS_PageChecking function.

Table 58: Values Returned by the MicoPMBUS_PageChecking Function

Return Value	Description
0x00	Page checking is Successful
0x10	Current PMBus command is not matching to the current page number

MicoPMBUS_ReadMeasurement Function

```
unsigned char MicoPMBUS_ReadMeasurement(MicoPMBUSCtx_t *ctx,
    unsigned int * p_read_data);
```

This function read the VMON/TMON/IMON measurement value from the ASC base on current page number. Error code will return when the read process is failed. It is user responsibility to modify the following variable before calling this function:

1) ctx->current_page: The Current PMBus Page

2) ctx->current_asc: The ASC to be read

3) ctx->current_mon: The Monitor (ASC register) to be read

Table 59 describes the parameter in the MicoPMBUS_ReadMeasurement function syntax.

Table 59: MicoPMBUS_ReadMeasurement Parameter

Parameter	Description
MicoPMBUSCtx_t	Pointer to a valid MicoPMBUSCtx _t structure representing a valid PMBus Adapter instance.
unsigned int *	This value identifies the buffer containing read data from ASC.

Table 60 describes the values returned by the MicoPMBUS_ReadMeasurement function.

Table 60: Values Returned by the MicoPMBUS_ReadMeasurement Function

Return Value	Description
0x00	Successful reading measurement value from ASC

MicoPMBUS_WriteThreshold Function

```
unsigned char MicoPMBUS_WriteThreshold (MicoPMBUSCtx_t *ctx,
unsigned int current_data,
unsigned char setting);
```

This function writes the Threshold Value into corresponding VMON/IMON/TMON of ASC base on current page number. Error code will return when the write process is failed. It is user responsibility to modify the following variable before calling this function:

1) ctx->current_page: The Current PMBus Page

2) ctx->current_asc: The ASC to be written

3) ctx->current_mon: The Monitor (ASC register) to be written

Table 61 describes the parameter in the MicoPMBUS_WriteThreshold function syntax.

Table 61: MicoPMBUS_WriteThreshold Parameter

Parameter	Description
MicoPMBUSCtx_t	Pointer to a valid MicoPMBUSCtx _t structure representing a valid PMBus Adapter instance.
unsigned int	This value identifies the Threshold Data to be written into ASC Register.
unsigned char	This value identifies the Voltage/Current/Temperature Threshold Setting. 0: Over Voltage/Current/Temperature Threshold Setting 1: Under Voltage/Current/Temperature Threshold Setting

Table 62 describes the values returned by the MicoPMBUS_WriteThreshold function.

Table 62: Values Returned by the MicoPMBUS_WriteThreshold Function

Return Value	Description
0x00	Successful writing threshold value to ASC
0x40	Invalid input data

MicoPMBUS_WPCheking Function

```
unsigned char MicoPMBUS_WPChecking (MicoPMBUSCtx_t *ctx,
                   unsigned char current_cmd);
```

This function checks the write protecting setting for Write PMBus protocol type command. Error code will return when the result is failed. It is user responsibility to modify the following variable before calling this function:

1) ctx->write_protect: Write Protect setting

Disable All (0x80)

Disable Most (0x40)

Enable All   (0x00)

Table 63 describes the parameter in the MicoPMBUS_WPChecking function syntax.

Table 63: MicoPMBUS_WPCheking Parameter

Parameter	Description
MicoPMBUSCtx_t	Pointer to a valid MicoPMBUSCtx _t structure representing a valid PMBus Adapter instance.
unsigned char	Current PMBus command handling by the LatticeMico8 microcontroller.

Table 64 describes the values returned by the MicoPMBUS_WPCheckingfunction.

Table 64: Values Returned by the MicoPMBUS_WriteThreshold Function

Return Value	Description
0x00	Successful writing threshold value to ASC
0x80	Invalid PMBus Command

MicoPMBUS_Execute Function

```
unsigned char PMBUS_Execute(MicoPMBUSCtx_t *ctx);
```

This function provide PMBus command decoding and executing. Error code will return when the process is failed. It is user responsibility to modify the following variable before calling this function:

1) ctx->write_protect: Write Protect setting

2) ctx->current_page: The Current PMBus Page

3) ctx->current_asc: The ASC to be read

4) ctx->current_mon: The Monitor (ASC register) to be read

Table 65 describes the parameter in the MicoPMBUS_PMBUS_Execute function syntax..

Table 65: MicoPMBUS_ Execute Parameter

Parameter	Description
MicoPMBUSCtx_t	Pointer to a valid MicoPMBUSCtx _t structure representing a valid PMBus Adapter instance.

Table 66 describes the values returned by the PMBUS_Execute function.

Table 66: Values Returned by the MicoPMBUS_ Execute Function

Return Value	Description
0x00	Successful writing threshold value to ASC
0x01	Other memory fault
0x02	Other communication fault
0x08	Process fault
0x10	Memory fault
0x40	Invalid PMBus Data
0x80	Invalid PMBus Command

Software Usage Example

This section provides an example of using the PMBus Adapter. The example is shown in Figure 4 and assumes the presence of a PMBus Adapter component named “pmb”, an EFB component named “efb” and a Mutex component named “mutex”.

Figure 4: PMBus Adapter Software Example

```
#include "DDStructs.h"
#include "MicoPMBUS.h"
#include "MicoEFB.h"
#include "MicoMUTEX.h"

int main(void){

      MicoEFBCtx_t * efb = &efb_machxo2_efb;
      MicoPMBUSCtx_t * pmb = &pmbus_pmbus;
      MicoMutexCtx_t *mutex = &mutex_mutex;

      unsigned char i2c_mutex_id = 0x00;
      MicoPMBUSInit(pmbus);
      MicoPMBUSRegisterEFBnMutex (pmb, efb, mutex,
i2c_mutex_id);

      unsigned char status;
      unsigned char cml_status = pmb->cml_status;
      size_t psi_address = pmb->psi_base;

      do{
        status = MicoPMBUS_Execute(pmb);
        if (status != MICO_PMBUS_RETURN_SUCESSFULLY){
              pmb->cml_status = status | cml_status;
              MICO_PMBUS_PSI_WRITE_CTL1(psi_address,
MICO_PMBUS_PSI_CTL1_ALRT);
        }
      }while(1);

      return 0;
`
```

Revision History

Component Version	Description
1.0	Initial release.

Trademarks