



# **Intel® FPGA Programmable Acceleration Card D5005 Board Management Controller User Guide**



**Subscribe**

**Send Feedback**

**UG-20203 | 2019.08.05**

Latest document on the web: [PDF](#) | [HTML](#)



## Contents

---

<b>1. Intel FPGA Programmable Acceleration Card D5005 BMC Introduction.....</b>	<b>3</b>
1.1. Overview.....	4
<b>2. Remote System Update.....</b>	<b>6</b>
<b>3. PLDM over MCTP SMBus.....</b>	<b>7</b>
3.1. SMBus Slave Address.....	7
3.2. SMBus Interface Speed.....	7
3.3. MCTP Packetization Support.....	8
3.4. Supported Command Sets.....	8
3.5. PLDM Topology and Hierarchy.....	9
<b>4. EEPROM Data Format.....</b>	<b>12</b>
<b>5. Revision History.....</b>	<b>20</b>



## 1. Intel FPGA Programmable Acceleration Card D5005 BMC Introduction

---

The Intel FPGA Programmable Acceleration Card D5005 (Intel® FPGA PAC D5005) contains an Intel MAX® 10 Board Management Controller (BMC). The BMC provides power sequence management, board monitoring via sensors, JTAG management and Remote System Update (RSU).

The BMC supports Platform Level Data Model (PLDM) using Management Component Transport Protocol (MCTP) over SMBus . This feature enables the host to get real-time telemetry data for system monitoring through the SMBus that is a part of the PCIe\* connector .

The BMC also supports RSU for:

- Nios® firmware inside of the Intel MAX 10
- Intel MAX 10 image
- Intel Stratix® 10 SX FPGA image

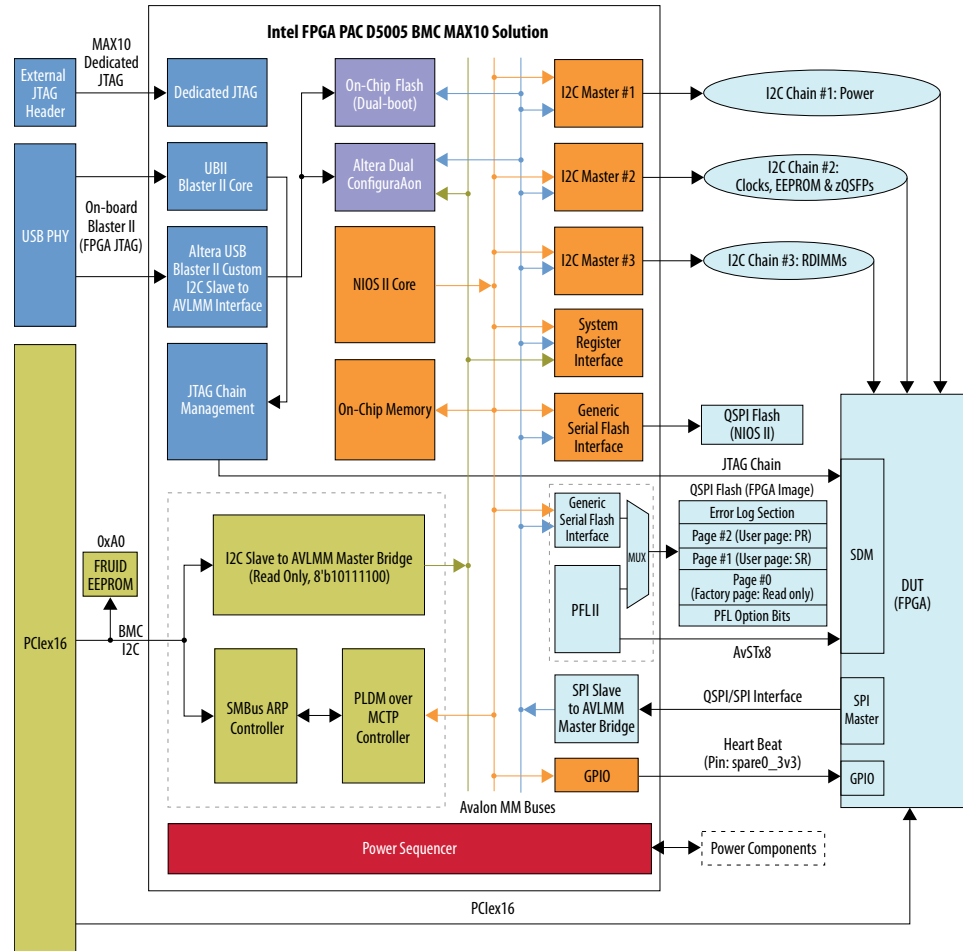
These updates are routed through an in-band path through the PCIe interface to the FPGA and then through the FPGA SPI master to the Intel MAX 10 SPI slave.

**Note:**

Throughout this document, any mention of Intel FPGA PAC or Intel FPGA PAC D5005 shall specifically refer to the Intel FPGA Programmable Acceleration Card D5005.

## 1.1. Overview

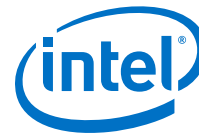
Figure 1. Intel FPGA PAC D5005 Intel MAX 10-Based BMC Diagram



- Platform Level Data Model (PLDM) over MCTP SMBus endpoint is supported. PLDM over MCTP via native PCIe is not supported.
- ACK-Poll is supported
  - Supported with SMBus default slave address 0xC0
  - Supported with a fixed or assigned slave address
- Standard I<sup>2</sup>C slave interface is supported, but read-only in the current release. It shares the same SMBus with the MCTP endpoint and the Field Replaceable Unit Identification (FRUID) EEPROM.
- Intelligent Platform Management Interface (IPMI) is not supported.



- **FRUID EEPROM:** Only the host can access this EEPROM for board related information.
- **MAC Address EEPROM:** It physically connects to both the Intel MAX 10 and the Intel Stratix 10 FPGA. The Intel Stratix 10 FPGA has a higher priority, but functionally it can only be accessed by the Intel MAX 10 device since FPGA default FPGA Interface Manager (FIM) image does not support MAC Address EEPROM access.



## 2. Remote System Update

---

The Intel FPGA PAC D5005 provides a mechanism to update Nios firmware, Intel MAX 10 image, or Intel Stratix 10 FPGA image over PCIe interface from the host called Remote System Update (RSU).

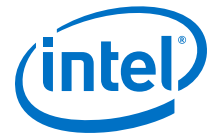
RSU can be used for the following three updates:

- Intel MAX 10 image update
- Nios firmware image update
- FPGA Interface Manager (FIM) image update

Refer to the *Intel FPGA Programmable Acceleration Card D5005 Quick Start Guide* for steps on how to remotely update the firmware and images.

### Related Information

[Intel FPGA Programmable Acceleration Card D5005 Quick Start Guide](#)



### 3. PLDM over MCTP SMBus

---

The BMC on the Intel FPGA PAC D5005 communicates with a server BMC over the PCIe SMBus.

The supported protocol is the Platform Level Data Model (PLDM) over Management Component Transport Protocol (MCTP) stack.

The Intel FPGA PAC D5005 BMC supports a subset of the PLDM and MCTP commands to enable a server (motherboard) BMC to obtain sensor data such as voltage, current and temperature.

The BMC supports version 1.3.0 of the MCTP Base Specification (DMTF specification DSP0236) and version 1.1.1 of the PLDM for Platform Monitoring and Control standard (DMTF specification DSP0248).

#### Related Information

[Distributed Management Task Force \(DMTF\) Specifications](#)

For link to specific DMTF specifications

#### 3.1. SMBus Slave Address

- Two SMBus slave addresses:
  - **MCTP endpoint slave address:** 0xCE by default. It can be reprogrammed into corresponding section of external FPGA Quad SPI flash via in-band way if necessary
  - **Standard I<sup>2</sup>C slave address:** 0xBC by default only for out-of-band accesses.
- SMBus device category:
  - "Fixed not Discoverable" device is supported by default, but all four device categories are supported and are field-reconfigurable

#### 3.2. SMBus Interface Speed

The Intel FPGA PAC D5005 implementation supports clock frequency self-adaptive from 10 KHz to 1 MHz.

### 3.3. MCTP Packetization Support

#### MCTP Definitions

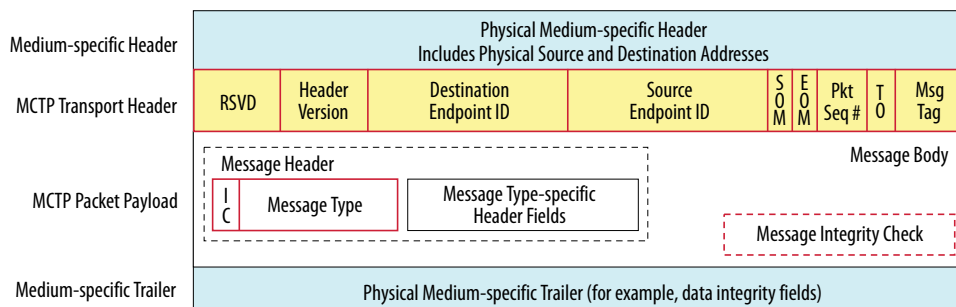
- The message body represents the payload of an MCTP message. The message body can span multiple MCTP packets.
- MCTP packet payload refers to the portion of the message body of an MCTP message that is carried in a single MCTP packet.
- Transmission Unit refers to the size of the portion of the MCTP packet payload.

#### Transmission Unit Size

- The baseline transmission unit (minimum transmission unit) size for MCTP is 64 bytes.
- All MCTP control messages are required to have a packet payload that is no larger than the baseline transmission unit without negotiation. (The negotiation mechanism for larger transmission units between endpoints is message type-specific and is not addressed in MCTP Base specification)
- Any MCTP message whose message body size is bigger than 64 bytes shall be split into multiple packets for a single message transmission.

#### MCTP Packet Fields

**Figure 2. Generic Packet/Message Fields**

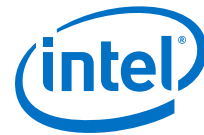


### 3.4. Supported Command Sets

#### Supported MCTP Commands

- Get MCTP Version Support
  - Base Spec Version Info
  - Control Protocol Version Info
  - PLDM over MCTP Version
- Set Endpoint ID
- Get Endpoint ID
- Get Endpoint UUID
- Get Message Type Support
- Get Vendor Defined Message Support





**Note:** For Get Vendor Defined Message Support command, the BMC responds with the completion code `ERROR_INVALID_DATA(0x02)`.

#### Supported PLDM Base Specification Commands

- SetTID
- GetTID
- GetPLDMVersion
- GetPLDMTypes
- GetPLDMCommands

#### Supported PLDM for Platform Monitoring and Control Specification Commands

- SetTID
- GetTID
- GetSensorReading
- GetSensorThresholds
- SetSensorThresholds
- GetPDRRepositoryInfo
- GetPDR

**Note:** The BMC Nios II core polls for different telemetry data every three seconds, hence the response message versus a corresponding request message of the command `GetSensorReading` or `GetSensorThresholds` accordingly updates every three seconds.

### 3.5. PLDM Topology and Hierarchy

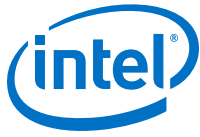
#### Defined Platform Descriptor Records

The Intel FPGA PAC D5005 uses 43 Platform Descriptor Records (PDRs). The PDRs are consolidated and do not support dynamic change even for events such as a QSFP module plug/unplug event. For QSFP module plug/unplug event, the BMC firmware uses another sensor that indicates if the QSFP is plugged or not, and then returns Normal for this sensor even if unplugged.

#### Sensor Names and Record Handle

All PDRs are assigned an opaque numeric value called the Record Handle. This value is used for accessing individual PDRs within the PDR Repository via `GetPDR` (DTMF specification DSP0248).

The following table is a consolidated list of sensors monitored on Intel FPGA PAC D5005.



**Table 1. PDRs Sensor Names and Record Handle**

Function	Sensor Name	PDR Record Handle
Total Intel FPGA PAC input power	N/A	1
PCIe fingers 12V Current	12 V Backplane Current	2
PCIe fingers 12V Voltage	12 V Backplane Voltage	3
1.2 V Rail Voltage	N/A	4
1.2 V Rail Current	N/A	5
1.8 V Rail Voltage	1.8 V Voltage	6
1.8 V Rail Current	1.8 V Current	7
3.3 V Rail Voltage	3.3 V Voltage	8
3.3 V Rail Current	3.3 V Current	9
FPGA Core Voltage	FPGA Core Voltage	10
FPGA Core Current	FPGA Core Current	11
FPGA Core Temperature	FPGA Core Temperature	12
Board Temperature	N/A	13
QSFP 3.3V Rail	QSFP0 Voltage	14
QSFP Temperature	QSFP0 Temperature	15
Core Supply Temp	N/A	16
Core Intel FPGA PAC supply temp input	N/A	17
VCCR Rail Voltage	VCCR Voltage	18
VCCT Rail Voltage	VCCT Voltage	19
VCCR Rail Current	VCCR Current	20
VCCT Rail Current	VCCT Current	21
VPP Rail Voltage	N/A	22
VTT Rail Voltage	N/A	23
PCIe Auxillary 12 V Current	12 V AUX Current	24
PCIe Auxillary 12 V Voltage	12 V AUX Voltage	25
PCIe Auxillary 12 V Temperature	12 V AUX Temperature	26
PCIe fingers 12V Temperature	12 V Backplane Temperature	27
3.3 V Rail Temperature	3.3 V Temperature	28
1.8 V Rail Temperature	1.8 V Temperature	29
VCCERAM Rail Current	VCCERAM Current	30
VCCERAM Rail Voltage	VCCERAM Voltage	31
VCCERAM Rail Temperature	VCCERAM Temperature	32
VCCR_GXB Regulator Temperature	VCCR Temperature	33
FPGA Transceiver Die Temperature	FPGA Transceiver Temperature	34
Board Inlet Air Temperature	Board Inlet Air Temperature	35
continued...		



Function	Sensor Name	PDR Record Handle
Board Exhaust Air Temperature	Board Exhaust Air Temperature	36
QSFP 3.3 V Rail	QSFP1 Voltage	37
QSFP Temperature	QSFP1 Temperature	38
DDR4 Module 0 Temperature	RDIMM0 Temperature	39
DDR4 Module 1 Temperature	RDIMM1 Temperature	40
DDR4 Module 2 Temperature	RDIMM2 Temperature	41
DDR4 Module 3 Temperature	RDIMM3 Temperature	42
VCCT_GXB Regulator Temperature	VCCT Temperature	43



## 4. EEPROM Data Format

---

This section is used to define data format of both FRUID EEPROM and MAC Address EEPROM that can be accessed by host and FPGA respectively.

### MAC EEPROM Access

MAC EEPROM can only be read by Intel MAX 10 BMC or Intel Stratix 10 FPGA (but not from the host).

The MAC Address EEPROM only contains the starting 6-byte MAC address at address 0x00h followed by the MAC address count of 08. The starting MAC address is also printed on the label sticker on the back side of the Printed Circuit Board (PCB). From the starting MAC address, the remaining seven MAC addresses are obtained by sequentially incrementing the Least Significant Byte (LSB) of the starting MAC Address by a count of one for each subsequent MAC address.

Example:

Starting MAC: 644C360F4430

Count: 08

Subsequent MAC:

644C360F4431

644C360F4432

644C360F4433

644C360F4434

644C360F4435

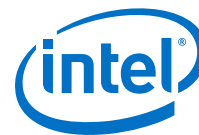
644C360F4436

644C360F4437

### FRUID EEPROM

- FRUID EEPROM (0xA0) can be read only from the host BMC via PCIe SMBus.

The structure in the FRUID EEPROM is based on the IPMI specification, *Platform Management FRU Information Storage Definition, v1.3, March 24, 2015*, from which a board information structure is derived. The FRUID EEPROM follows the common header format with Board Area and Product Info Area.



The common header bytes are placed from the first address of the EEPROM. The layout will look like the figure below. In the case of FRUID EEPROM, after factory programming, write protect will be enabled; no updates are possible to any of the EEPROM contents.

**Figure 3. FRUID EEPROM Memory Layout Block Diagram**



**Table 2. Board Area of FRUID EEPROM**

Field Length in Bytes	Field Description	Field Values	Field Encoding
1	Board Area Format Version 7:4 - reserved, write as 0000b 3:0 - format version number	0x01	Set to 1h (0000 0001b)
1	Board Area Length (in multiples of 8 bytes)	0x0B	88 bytes (includes 2 pad 00 bytes)
1	Language Code	0x00	Set to 0 for English <i>Note:</i> No other languages supported at this time
3	<b>Mfg. Date / Time</b> Number of minutes from 0:00 hrs 1/1/96. Least Significant byte first (little endian) 00_00_00h = unspecified	0x10 0x65 0xB7	Time difference between 12:00 AM 1/1/96 to 12 PM 11/07/2018 is 12018960 minutes = b76510h – stored in little endian format
1	<b>Board Manufacturer</b> type/length byte	0xD2	8-bit ASCII + LATIN1 coded 7:6 – 11b 5:0 – 010010b (18 bytes of data)
P	Board Manufacturer bytes	0x49	8-bit ASCII + LATIN1 coded

*continued...*



Field Length in Bytes	Field Description	Field Values	Field Encoding
		0x6E 0x74 0x65 0x6C 0xAE 0x20 0x43 0x6F 0x72 0x70 0x6F 0x72 0x61 0x74 0x69 0x6F 0x6E	Intel Corporation
1	<b>Board Product Name</b> type/length byte	0xD5	8-bit ASCII + LATIN1 coded 7:6 – 11b 5:0 – 010101b (21 bytes of data)
Q	Board Product Name bytes	0x49 0x6E 0x74 0x65 0x6C 0xAE 0x20 0x46 0x50 0x47 0x41 0x20 0x50 0x41 0x43 0x20 0x44 0x35 0x30 0x30 0x35	8-bit ASCII + LATIN1 coded Intel FPGA PAC D5005
1	<b>Board Serial Number</b> type/length byte*	0xCC	8-bit ASCII + LATIN1 coded 7:6 – 11b 5:0 – 001100b (12 bytes of data)
N	Board Serial Number bytes*	0x46 0x38 0x42 0x43 0x31 0x32	8-bit ASCII + LATIN1 coded 1st 6 hex digits are OUI: <b>F8BC12</b> 2nd 6 hex digits are MAC address: <b>00AB2E</b>
continued...			



Field Length in Bytes	Field Description	Field Values	Field Encoding
		0x30 0x30 0x41 0x42 0x32 0x45	<i>Note:</i> This is coded as an example and will need to be modified in an actual device
1	<b>Board Part Number</b> type/length byte	0xCE	8-bit ASCII + LATIN1 coded 7:6 – 11b 5:0 – 001110b (14 bytes of data)
M	Board Part Number bytes	0x4B 0x31 0x39 0x30 0x36 0x32 0x2D 0x30 0x30 0x33 0x52 0x30 0x2E 0x33	8-bit ASCII + LATIN1 coded with BOM ID. Specific example in cell alongside <b>K19062-003R0.3</b> <i>Note:</i> this is coded as an example and will need to be modified in an actual device.
1	<b>FRU File ID</b> type/length byte	0x00	8-bit ASCII + LATIN1 coded 7:6 – 00b 5:0 – 000000b (0 bytes of data) The FRU File ID bytes field that should follow this is not included as the field would be 'null'. <i>Note:</i> FRU File ID bytes. The FRU File version field is a pre-defined field provided as a manufacturing aid for verifying the file that was used during manufacture or field update to load the FRU information. The content is manufacturer-specific. This field is also provided in the Board Info area. Either or both fields may be 'null'.
1	<b>MMID</b> type/length byte	0xC6	8-bit ASCII + LATIN1 coded 7:6 – 11b 5:0 – 000110b (6 bytes of data)

continued...



Field Length in Bytes	Field Description	Field Values	Field Encoding
			<i>Note:</i> This is coded as an example and will need to be modified in an actual device
M	MMID bytes	0x41 0x42 0x31 0x32 0x33 0x34	Formatted as 6 hex digits. Specific example in cell alongside <b>AB1234</b>
1	C1h (type/length byte encoded to indicate no more info fields).	0xC1	
Y	00h - any remaining unused space	0x00	
1	Board Area Checksum (zero checksum)	0x9F	<i>Note:</i> The checksum in this table is a zero checksum computed for the values used in the table. It must be recomputed for the actual values of a Intel FPGA PAC D5005

**Table 3. Product Info Area of FRUID EEPROM**

Field Length in Bytes	Field Description	Field Values	Field Encoding
1	Product Area Format Version 7:4 - reserved, write as 0000b 3:0 - format version number = 1h for this specification	0x01	Set to 1h (0000 0001b)
1	Product Area Length (in multiples of 8 bytes)	0x0A	Total of 80 bytes
1	Language Code	0x00	Set to 0 for English <i>Note:</i> No other languages supported at this time
1	<b>Manufacturer Name</b> type/length byte	0xD2	8-bit ASCII + LATIN1 coded 7:6 - 11b 5:0 - 010010b (18 bytes of data)
N	Manufacturer Name bytes	0x49 0x6E 0x74 0x65 0x6C 0xAE 0x20 0x43 0x6F 0x72 0x70 0x6F	8-bit ASCII + LATIN1 coded <b>Intel Corporation</b>

*continued...*



#### 4. EEPROM Data Format

UG-20203 | 2019.08.05



Field Length in Bytes	Field Description	Field Values	Field Encoding
		0x72 0x61 0x74 0x69 0x6F 0x6E	
1	<b>Product Name</b> type/length byte	0xD5	8-bit ASCII + LATIN1 coded 7:6 – 11b 5:0 – 010101b (21 bytes of data)
M	Product Name bytes	0x49 0x6E 0x74 0x65 0x6C 0xAE 0x20 0x46 0x50 0x47 0x41 0x20 0x50 0x41 0x43 0x20 0x44 0x35 0x30 0x30 0x35	8-bit ASCII + LATIN1 coded Intel FPGA PAC D5005
1	<b>Product Part/Model Number</b> type/length byte	0xCE	8-bit ASCII + LATIN1 coded 7:6 – 11b 5:0 – 001110b (14 bytes of data)
O	Product Part/Model Number bytes	0x42 0x44 0x2D 0x41 0x43 0x44 0x2D 0x44 0x35 0x30 0x30 0x35 0x2D 0x31	8-bit ASCII + LATIN1 coded OPN for the board <b>BD-ACD-D5005-1</b>
1	<b>Product Version</b> type/length byte	0x01	8-bit binary 7:6 – 00b

*continued...*



Field Length in Bytes	Field Description	Field Values	Field Encoding
			5:0 – 000001b (1 byte of data)
R	Product Version bytes	0x00	This field is encoded as family member
1	<b>Product Serial Number</b> type/length byte*	0xCC	8-bit ASCII + LATIN1 coded 7:6 – 11b 5:0 – 001100b (12 bytes of data)
P	Product Serial Number bytes*	0x46 0x38 0x42 0x43 0x31 0x32 0x30 0x30 0x41 0x42 0x32 0x45	8-bit ASCII + LATIN1 coded 1st 6 hex digits are OUI: <b>F8BC12</b> 2nd 6 hex digits are MAC address: <b>00AB2E</b> <i>Note:</i> This is coded as an example and will need to be modified for a specific Intel FPGA PAC instance
1	<b>Asset Tag</b> type/length byte	0x01	8-bit binary 7:6 – 00b 5:0 – 000001b (1 byte of data)
Q	Asset Tag	0x00	Not supported
1	<b>FRU File ID</b> type/length byte	0x00	8-bit ASCII + LATIN1 coded 7:6 – 00b 5:0 – 000000b (0 bytes of data) The FRU File ID bytes field that should follow this is not included as the field would be 'null'. <i>Note:</i> FRU The FRU File version field is a pre-defined field provided as a manufacturing aid for verifying the file that was used during manufacture or field update to load the FRU information. The content is manufacturer-specific. This field is also provided in the Board Info area. Either or both fields may be 'null'.

*continued...*

#### 4. EEPROM Data Format

UG-20203 | 2019.08.05



Field Length in Bytes	Field Description	Field Values	Field Encoding
1	C1h (type/length byte encoded to indicate no more info fields).	0xC1	
Y	00h - any remaining unused space	0x00	
1	Product Info Area Checksum (zero checksum)	0xB9	<i>Note:</i> the checksum in this table is a zero checksum computed for the values used in the table. It must be recomputed for the actual values of a Intel FPGA PAC



## 5. Revision History

---

**Table 4. Revision History for the Intel FPGA Programmable Acceleration Card D5005 Board Management Controller (BMC) User Guide**

Document Version	Changes
2019.08.05	Initial Release

Intel Corporation. All rights reserved. Agilix, Altera, Arria, Cyclone, Enpirion, Intel, the Intel logo, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

\*Other names and brands may be claimed as the property of others.

**ISO  
9001:2015  
Registered**