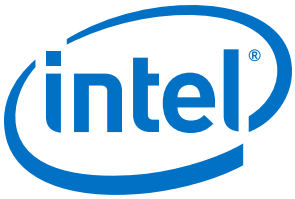


Intel® FPGA Programmable Acceleration Card D5005 Board Management Controller User Guide

Subscribe
Send Feedback

UG-20203 | 2019.11.04
Latest document on the web: PDF | HTML

Contents

1. Intel FPGA Programmable Acceleration Card D5005 BMC Introduction

1.1. About this document

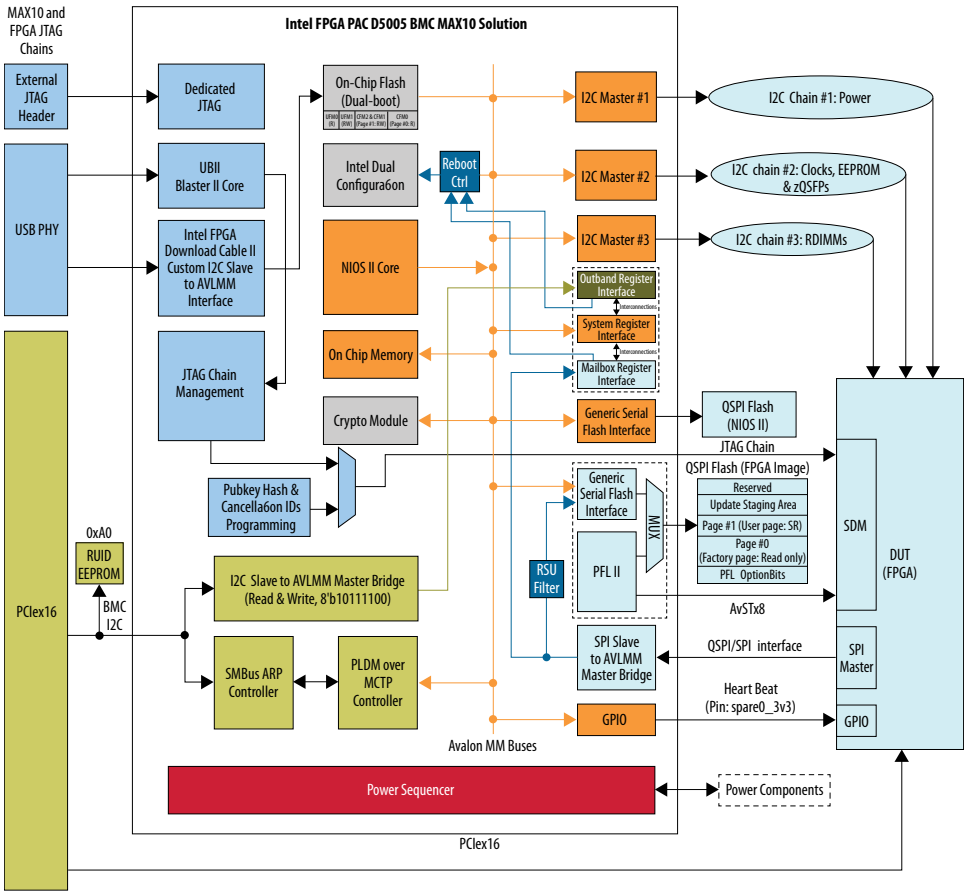
Reference the Intel FPGA Programmable Acceleration Card D5005 Board Management User Guide to learn more about the functions and features of the Intel® MAX® 10 BMC and to understand how to read telemetry data on the Intel FPGA Programmable Acceleration Card D5005 using I^2C and SMBus. An introduction to Intel MAX 10 root of trust (RoT) and secure remote system update is included.

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.

1.2. Overview

An Intel MAX 10 FPGA contains the Board Management Controller (BMC) for the Intel FPGA Programmable Acceleration Card D5005. The BMC acts as Root of Trust (RoT) on the Intel FPGA PAC D5005. The Intel FPGA PAC D5005 BMC supports features such as power sequence management and board monitoring through sensors. It supports PLDM over MCTP through SMBus protocol so that the host can get real-time telemetry data for system monitoring via PCIe SMBus. BMC also supports secure remote system update for Nios firmware, Intel MAX 10 image and FPGA Interface Manager (FIM) image updates.

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



1.3. Root of Trust (RoT)

The Intel MAX 10 BMC acts as a Root of Trust (RoT) and enables the secure remote system update feature of the Intel FPGA PAC D5005. Please see section Secure Remote System Update on page 5 for more information. The RoT includes features that may help prevent the following:

- Loading or executing of unauthorized code or designs.
- Disruptive operations attempted by unprivileged software, privileged software, or the host BMC.
- Unintended execution of older code or designs with known bugs or vulnerabilities by enabling the BMC to revoke authorization.

The Intel FPGA PAC D5005 BMC also enforces several other security policies relating to access through various interfaces,as well as protecting the on-board flash through write rate limitation.

Please refer to the *Security User Guide For Intel FPGA Programmable Acceleration Card D5005* for information on RoT and security features of the Intel FPGA PAC D5005.

Related Information

Security User Guide For Intel FPGA Programmable Acceleration Card D5005

1.4. Power Sequence Management

The BMC Power sequencer state machine is used to manage programmable acceleration card power-on and power-off sequences, to handle different corner cases during power-on process or normal operation. Intel MAX 10 power-up flow covers the entire process including Intel MAX 10 boot-up, Nios® II boot-up, and power sequence management for FPGA configuration. The host needs to check the build versions of both Intel MAX 10 and FPGA, and the Nios II status every time after a power-cycle, and then takes corresponding actions in case the Intel FPGA PAC D5005 runs into corner cases such as a Intel MAX 10/FPGA factory build load failure or Nios II boot up failure.

1.5. Board Monitoring Through Sensors

The Intel MAX 10 BMC monitors voltage, current and temperature of various components on the Intel FPGA PAC D5005. Host BMC can access the telemetry data through PCIe SMBus. The PCIe SMBus between the host BMC and Intel FPGA PAC D5005 Intel MAX 10 BMC is shared by both: PLDM over MCTP SMBus endpoint and standard I^2C slave to Avalon-MM interface (read-only).

1.6. Secure Remote System Update

The Intel FPGA PAC D5005 provides a mechanism to securely update Nios II 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 II firmware image update
- FPGA Interface Manager (FIM) image update

The Nios II firmware is in charge of authenticating the image during the update process. The updates are pushed over the PCIe interface to the Intel Stratix 10 SX FPGA, which in turn writes to the Intel Stratix 10 FPGA SPI master and finally to the Intel MAX 10 FPGA SPI slave. A temporary flash area called staging area stores any type of authentication bitstream through SPI interface.

BMC ROT contains the cryptographic module which implements SHA2-256 bit Hash verification function to authenticate the keys and ECDSA-256P-256 signature verification to authenticate your AFU. Nios II Firmware uses the cryptographic module to authenticate the user signed image in the staging area, if authentication passes, Nios II Firmware copies the user image to user flash area. If the authentication fails Nios II Firmware reports an error. Please refer to the *Security User Guide for the Intel FPGA Programmable Acceleration Card D5005* for information on RoT and security features of the Intel FPGA PAC D5005.

1.7. FRUID EEPROM

Only the host can access this EEPROM for board related information.

1.8. 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.

Related Information

Security User Guide For Intel FPGA Programmable Acceleration Card D5005

2. 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

2.1. 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^2C slave address**: 0xBC by default only for out-of-band accesses.
- SMBus device category:

Table 1. SMBus device category

Device Category	Description
2'b00	Non-ARP capable
2'b01	Fixed, not discoverable (by default)
2'b10	Fixed and Discoverable
2'b11	ARP-capable, but only "Dynamic and volatile address device" is supported

2.2. SMBus Interface Speed

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

2.3. MCTP Packetization Support

MCTP Definitions

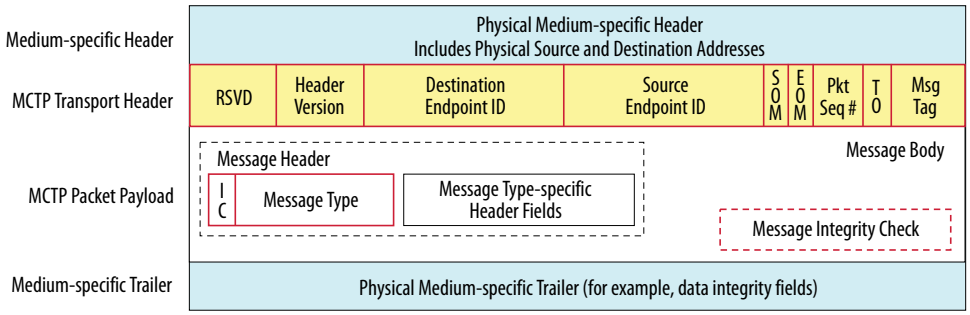
- The message body represents the payload of an MCTP message. The message body can span multiple MCTP packets.
- MCTP packet payload refers to the portion of the message body of an MCTP message that is carried in a single MCTP packet.
- Transmission Unit refers to the size of the portion of the MCTP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



2.4. 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 second, hence the response message versus a corresponding request message of the command `GetSensorReading` or `GetSensorThresholds` accordingly updates every three seconds.

2.5. PLDM Topology and Hierarchy

Defined Platform Descriptor Records

The Intel FPGA PAC D5005 uses 43 Platform Descriptor Records (PDRs). 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 2. 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

continued...

Function	Sensor Name	PDR Record Handle
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
Board Exhaust Air Temperature	Board Exhaust Air Temperature	36
QSFP 3.3 V Rail	QSFP1 Voltage	37
QSFP Temperature	QSFP1 Temperature	38

continued...

Function	Sensor Name	PDR Record Handle
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

2.6. Board Monitoring through I²C SMBus

The standard I²C slave to Avalon-MM interface (read-only) shares the PCIe SMBus between the host BMC and the Intel MAX 10 RoT. The Intel FPGA PAC D5005 supports standard I²C slave interface and the slave address is 0xBC by default only for out-of-band access. Byte addressing mode is 2-byte offset address mode.

This section covers the telemetry data register memory map that you can use to access information through the I²C commands. The description column describes how the returned register values may be further processed to get the actual values. The units can be Celsius (°C), mA, mV, mW depending on what sensor you read.

Table 3. Telemetry Data Register Memory Map

Register	Offset	Width	Description
Board Inlet Air Temperature	0x100	32	Register value is signed integer register value * 0.5
Board Inlet Air Temperature High Warn	0x104	32	Register value is signed integer register value * 0.5
Board Inlet Air Temperature High Fatal	0x108	32	Register value is signed integer register value * 0.5
Board Inlet Air / Core Die Hysteresis	0x10C	32	Register value is signed integer register value * 0.5
FPGA Core Temperature	0x110	32	Register value is signed integer register value * 0.5
FPGA Core Temperature High Warn	0x114	32	Register value is signed integer register value * 0.5
FPGA Core Temperature High Fatal	0x118	32	Register value is signed integer register value * 0.5
Board Exhaust Air Temperature	0x11C	32	Register value is signed integer register value * 0.5
Board Exhaust Air Temperature High Warn	0x120	32	Register value is signed integer register value * 0.5

continued...

Register	Offset	Width	Description
Board Exhaust Air Temperature High Fatal	0x124	32	Register value is signed integer register value * 0.5
Board Exhaust Air / Transceiver Die Hysteresis	0x128	32	Register value is signed integer register value * 0.5
FPGA Core Temperature	0x110	32	Register value is signed integer register value * 0.5
FPGA Core Temperature High Warn	0x114	32	Register value is signed integer register value * 0.5
FPGA Core Temperature High Fatal	0x118	32	Register value is signed integer register value * 0.5
Board Exhaust Air Temperature	0x11C	32	Register value is signed integer register value * 0.5
Board Exhaust Air Temperature High Warn	0x120	32	Register value is signed integer register value * 0.5
Board Exhaust Air Temperature High Fatal	0x124	32	Register value is signed integer register value * 0.5
Board Exhaust Air / Transceiver Die Hysteresis	0x128	32	Register value is signed integer register value * 0.5
FPGA Transceiver Temperature	0x12C	32	Register value is signed integer register value * 0.5
FPGA Transceiver Temperature High Warn	0x130	32	Register value is signed integer register value * 0.5
FPGA Transceiver Temperature High Fatal	0x134	32	Register value is signed integer register value * 0.5
RDIMM0 Temperature	0x138	32	Register value is signed integer register value * 0.5
RDIMM0 Temperature High Warn	0x13C	32	Register value is signed integer register value * 0.5
RDIMM0 Temperature High Fatal	0x140	32	Register value is signed integer register value * 0.5
RDIMM0 Temperature Hysteresis	0x144	32	Register value is signed integer register value * 0.5

continued...

Register	Offset	Width	Description
RDIMM1 Temperature	0x148	32	Register value is signed integer register value * 0.5
RDIMM1 Temperature High Warn	0x14C	32	Register value is signed integer register value * 0.5
RDIMM1 Temperature High Fatal	0x150	32	Register value is signed integer register value * 0.5
RDIMM1 Temperature Hysteresis	0x154	32	Register value is signed integer register value * 0.5
RDIMM2 Temperature	0x158	32	Register value is signed integer register value * 0.5
RDIMM2 Temperature High Warn	0x15C	32	Register value is signed integer register value * 0.5
RDIMM2 Temperature High Fatal	0x160	32	Register value is signed integer register value * 0.5
RDIMM2 Temperature Hysteresis	0x164	32	Register value is signed integer register value * 0.5
RDIMM3 Temperature	0x168	32	Register value is signed integer register value * 0.5
RDIMM3 Temperature High Warn	0x16C	32	Register value is signed integer register value * 0.5
RDIMM3 Temperature High Fatal	0x170	32	Register value is signed integer register value * 0.5
RDIMM3 Temperature Hysteresis	0x174	32	Register value is signed integer register value * 0.5
QSFP0 Temperature	0x178	32	Register value is signed integer register value * 0.5
QSFP0 Temperature High Fatal	0x17C	32	Register value is signed integer register value * 0.5
QSFP0 Temperature High Warn	0x180	32	Register value is signed integer register value * 0.5
QSFP0 Supply Voltage	0x184	32	Voltage(mV) = register value
QSFP1 Temperature	0x188	32	Register value is signed integer

continued...

Register	Offset	Width	Description
			register value * 0.5
QSFP1 Temperature High Fatal	0x18C	32	Register value is signed integer register value * 0.5
QSFP1 Temperature High Warn	0x190	32	Register value is signed integer register value * 0.5
QSFP1 Voltage	0x194	32	register value
FPGA Core Voltage	0x198	32	register value
FPGA Core Current	0x19C	32	register value
3.3 V Temperature	0x1A0	32	Register value is signed integer register value * 0.5
3.3 V Temperature High Warn	0x1A4	32	Register value is signed integer register value * 0.5
3.3 V Temperature High Fatal	0x1A8	32	Register value is signed integer register value * 0.5
3.3 V Voltage	0x1AC	32	Voltage(mV) register value
3.3 V Voltage High Warn	0x1B0	32	Voltage(mV) register value
3.3 V Voltage High Fatal	0x1B4	32	Voltage(mV) register value
3.3 V Current	0x1B8	32	Voltage(mA) register value * 10
VCCERAM Temperature	0x1BC	32	Register value is signed integer register value * 0.5
VCCERAM Temperature High Warn	0x1C0	32	Register value is signed integer register value * 0.5
VCCERAM Temperature High Fatal	0x1C4	32	Register value is signed integer register value * 0.5
VCCERAM Voltage	0x1C8	32	Voltage(mV) register value
VCCERAM Voltage High Warn	0x1CC	32	Voltage(mV) register value
VCCERAM Voltage High Fatal	0x1D0	32	Voltage(mV) register value
VCCERAM Current	0x1D4	32	Voltage(mA) register value * 10
VCCR Temperature	0x1D8	32	Register value is signed integer

continued...

Register	Offset	Width	Description
			register value * 0.5
VCCR Temperature High Warn	0x1DC	32	Register value is signed integer register value * 0.5
VCCR Temperature High Fatal	0x1E0	32	Register value is signed integer register value * 0.5
VCCR Voltage	0x1E4	32	Voltage(mV) register value
VCCR Voltage High Warn	0x1E8	32	Voltage(mV) register value
VCCR Voltage High Fatal	0x1EC	32	Voltage(mV) register value
VCCR Current	0x1F0	32	Voltage(mA) register value * 10
VCCT Temperature	0x1F4	32	Register value is signed integer register value * 0.5
VCCT Temperature High Warn	0x1F8	32	Register value is signed integer register value * 0.5
VCCT Temperature High Fatal	0x1FC	32	Register value is signed integer register value * 0.5
VCCT Voltage	0x200	32	Voltage(mV) register value
VCCT Voltage High Warn	0x204	32	Voltage(mV) register value
VCCT Voltage High Fatal	0x208	32	Voltage(mV) register value
VCCT Current	0x20C	32	Voltage(mA) register value * 10
1.8 V Temperature	0x210	32	Register value is signed integer register value * 0.5
1.8 V Temperature High Warn	0x214	32	Register value is signed integer register value * 0.5
1.8 V Temperature High Fatal	0x218	32	Register value is signed integer register value * 0.5
1.8 V Voltage	0x21C	32	Voltage(mV) register value
1.8 V Voltage High Warn	0x220	32	Voltage(mV) register value
1.8 V Voltage High Fatal	0x224	32	Voltage(mV)

continued...

Register	Offset	Width	Description
			register value
1.8 V Current	0x228	32	Voltage(mA) register value * 10
12 V Backplane Temperature	0x22C	32	Register value is signed integer register value * 0.5
12 V Backplane Temperature High Warn	0x230	32	Register value is signed integer register value * 0.5
12 V Backplane Temperature High Fatal	0x234	32	Register value is signed integer register value * 0.5
12 V Backplane Voltage	0x238	32	Voltage(mV) register value
12 V Backplane Voltage Low Warn	0x23C	32	Voltage(mV) register value
12 V Backplane Current	0x240	32	Voltage(mA) register value
12 V Backplane Current High Warn	0x244	32	Voltage(mA) register value
12 V AUX Temperature	0x248	32	Register value is signed integer register value * 0.5
12 V AUX Temperature High Warn	0x24C	32	Register value is signed integer register value * 0.5
12 V AUX Temperature High Fatal	0x250	32	Register value is signed integer register value * 0.5
12 V AUX Voltage	0x254	32	Voltage(mV) register value
12 V AUX Voltage Low Warn	0x258	32	Voltage(mV) register value
12 V AUX Current	0x25C	32	Voltage(mA) register value
12 V AUX Current High Warn	0x260	32	Voltage(mA) register value

Use the Intelligent Platform Management Interface (IPMI) tool to read the telemetry data through the I^2C bus.

3. 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

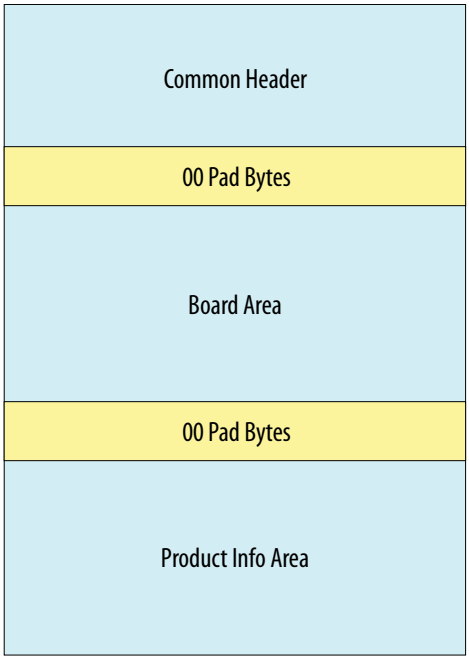


Table 4. Board Area of FRUID EEPROM

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 *Note:* No other languages supported at this time
3	**Mfg. Date / Time**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	**Board Manufacturer** 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	**Board Product Name** 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	**Board Serial Number** 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: **F8BC12** 2nd 6 hex digits are MAC address: **00AB2E**

continued...

Field Length in Bytes	Field Description	Field Values	Field Encoding
		0x30 0x30 0x41 0x42 0x32 0x45	*Note:* This is coded as an example and will need to be modified in an actual device
1	**Board Part Number** 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 **K19062-003R0.3** *Note:* this is coded as an example and will need to be modified in an actual device.
1	**FRU File ID** 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'. *Note:* 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	**MMID** 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
			Note: 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 **AB1234**
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	*Note:* 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 5. 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 *Note:* No other languages supported at this time
1	**Manufacturer Name** 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 **Intel Corporation**

continued...

Field Length in Bytes	Field Description	Field Values	Field Encoding
		0x72 0x61 0x74 0x69 0x6F 0x6E	
1	**Product Name** 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	**Product Part/Model Number** 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 **BD-ACD-D5005-1**
1	**Product Version** 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	**Product Serial Number** 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: **F8BC12** 2nd 6 hex digits are MAC address: **00AB2E** *Note:* This is coded as an example and will need to be modified for a specific Intel FPGA PAC instance
1	**Asset Tag** type/length byte	0x01	8-bit binary 7:6 – 00b 5:0 – 000001b (1 byte of data)
Q	Asset Tag	0x00	Not supported
1	**FRU File ID** 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'. *Note:* 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...

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	*Note:* 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

4. Revision History

Table 6. Revision History for the Intel FPGA Programmable Acceleration Card D5005 Board Management Controller (BMC) User Guide

Document Version	Changes
2019.11.04	Added new sections: • Root of Trust (RoT) on page 4 • Power Sequence Management on page 5 • Board Monitoring Through Sensors on page 5 • Secure Remote System Update on page 5 • FRUID EEPROM on page 5 • MAC Address EEPROM on page 6
2019.08.05	Initial Release