

# **OCC Firmware Interface Specification for POWER10**

Version 1.~~46~~17

**Maintained by: Martha Broyles**  
**[mbroyles@us.ibm.com](mailto:mbroyles@us.ibm.com)**

## Change History

<b>Date</b>	<b>Change Description</b>
10/10/2019	Version 1 – First document version
10/31/2019	Version 1.2 – Misc cleanup
12/05/2019	Version 1.3 – Host data area. Mfg test command to select WOF VRT
03/19/2020	Version 1.4 – Add altitude to ambient command. Define enum for bottom of throttle space. Added back “nominal” and power save modes.
06/23/2020	Version 1.5 – Add WOF reset limit reached to sys config. Add disabled freq to poll response. Update memory throttle config packet.
07/22/2020	Version 1.6 – Update Sensor list for processor IO ring and nest
08/14/2020	Version 1.7 – updated WOF sensors. Added Digital Droop sensors.
09/22/2020	Version 1.8 – New memory config version 0x30 with update time
10/21/2020	Version 1.9 – Updates to OPAL shared memory
01/26/2021	Version 1.10 – Add parameter table to shared memory
02/10/2021	Version 1.11 – Add folding parameter to table
03/12/2021	Version 1.12 – Added BMC Power and Thermal Management Settings section
03/23/2021	Version 1.13 – Updates to PHYP parameter table for freq points (text string changes, don't report WOF base). Update processor DVFS/error attributes for BMC.
04/06/2021	Version 1.14 – Updates to mode text strings in PHYP interface. Call home sensor updates.
04/23/2021	Version 1.15 – Added special FRU types to thermal control thresholds configuration packet to support using Processor and Processor IO ring VPD temperature limits.
06/24/2021	Version 1.16 – Add SFP mode in PHYP interface
<u>02/08/2022</u>	<u>Version 1.17 – New APSS function IDs, memory FRU type updates for 4U DDIMMs</u>

## **Copyright and Disclaimer**

© Copyright International Business Machines Corporation 2019

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and Service names might be trademarks of IBM or other companies. A current list of IBM Trademarks is available on the Web at “Copyright and trademark information” at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others. All information contained in this document is subject to change without notice. The products described in this document are NOT intended for use in applications such as implantation, life support, or other hazardous uses where malfunction could result in death, bodily injury, or catastrophic property damage. The information contained in this document does not affect or change IBM product specifications or warranties. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of IBM or third parties. All information contained in this document was obtained in specific environments, and is presented as an illustration. The results obtained in other operating environments may vary. While the information contained herein is believed to be accurate, such information is preliminary, and should not be relied upon for accuracy or completeness, and no representations or warranties of accuracy or completeness are made.

Note: This document contains information on products in the design, sampling and/or initial production phases of development. This information is subject to change without notice. Verify with your IBM field applications engineer that you have the latest version of this document before finalizing a design.

THE INFORMATION CONTAINED IN THIS DOCUMENT IS PROVIDED ON AN “AS IS” BASIS. In no event will IBM be liable for damages arising directly or indirectly from any use of the information contained in this document.

IBM Systems and Technology Group 2070 Route 52, Bldg. 330 Hopewell Junction, NY 12533-6351

*The IBM home page can be found at [ibm.com®](http://ibm.com®).*

# Table of Contents

<b>1</b>	<b>OVERVIEW .....</b>	<b>8</b>
1.1	TERMS.....	8
1.2	OCC FUNCTIONAL OVERVIEW .....	9
1.3	OCC405 SRAM LAYOUT .....	10
1.4	OCB CHANNELS .....	11
1.5	HOST REQUIREMENTS FOR BMC SYSTEMS .....	12
1.6	ATTENTIONS/INTERRUPTS .....	13
1.6.1	To OCC.....	13
1.6.1.1	Command Write Attention Type = 0x01 .....	13
1.6.2	OCC to Host.....	14
1.6.3	OCC to FSP (TMGT).....	14
1.6.4	BMC to Host.....	14
1.7	COMMANDS .....	15
1.7.1	Command and Response Buffer Locations .....	15
1.7.2	OCC Command/Response Sequence.....	16
1.7.3	Command Format .....	17
1.7.4	Response Format.....	18
1.7.5	Error Response Data Packet.....	19
1.7.6	Command Summary Table .....	20
<b>2</b>	<b>COMMAND DEFINITIONS.....</b>	<b>23</b>
2.1	POLL .....	24
2.1.1	Version 0x20 Poll Response Data Definition.....	26
2.2	CLEAR ERROR LOG .....	33
2.3	SET MODE AND STATE.....	35
2.4	CONFIGURATION DATA.....	38
2.4.1	Frequency Points (Format = 0x02) – <b>NOT SUPPORTED</b> .....	39
2.4.2	Set OCC Role (Format = 0x03).....	40
2.4.3	APSS Configuration (Format = 0x04).....	41
2.4.4	Memory Configuration (Format = 0x05).....	42
2.4.5	Power Cap Values Data Packet (Format = 0x07).....	44
2.4.6	System Configuration (Format = 0x0F) .....	45
2.4.7	Idle Power Saver Settings (Format = 0x11) .....	48
2.4.8	Memory Throttling (Format = 0x12).....	49
2.4.9	Thermal Control Thresholds (Format = 0x13).....	51
2.4.10	AVSBus Configuration (Format = 0x14).....	53
2.4.11	GPU (Format = 0x15).....	54
2.4.12	Setup Configuration Data Return Packet .....	56
2.5	SET USER POWER CAP .....	57
2.6	RESET PREP.....	59
2.7	SEND AMBIENT TEMPERATURE .....	61
2.8	DEBUG PASS THROUGH .....	63
2.9	AME PASS THROUGH.....	65
2.10	GET FIELD DEBUG DATA.....	67
2.11	MFG TEST COMMAND .....	69

2.11.1	<i>Run/Stop Slew Between Two Frequency Points (Sub Cmd = 0x02)</i>	71
2.11.2	<i>List Sensors (Sub Cmd = 0x05)</i>	72
2.11.3	<i>Get Sensor Information (Sub Cmd = 0x06)</i>	74
2.11.4	<i>Enable/Disable Oversubscription Emulation (Sub Cmd = 0x07)</i>	75
2.11.5	<i>Run/Stop Memory Slew Between 1-100 Percent (Sub Cmd = 0x09)</i>	76
2.11.6	<i>Read Generated Pstate Table (Sub Cmd = 0x0B)</i>	77
2.11.7	<i>Select WOF VRT (Sub Cmd = 0x0F)</i>	78
<b>3</b>	<b>ERROR HANDLING</b>	<b>80</b>
3.1	OCC ERRORS	80
3.2	PM COMPLEX ERRORS OUTSIDE OF OCC 405	80
3.3	READING ERROR LOG FROM SRAM – FORMAT IN SRAM	81
3.4	ERRORS REQUIRING OCC RESET	84
3.5	OCC RESET PROCEDURE (SAFE MODE)	85
3.6	ERROR SCENARIOS	86
3.6.1	<i>(H)TMGT-OCC Communication Failure</i>	86
3.6.2	<i>BMC-OCC Communication Failure</i>	86
3.6.2.1	<i>BMC-OCC Communication Failure Handling Flow</i>	87
3.6.3	<i>OCC Fails to Load or Fails to go Active</i>	88
3.6.4	<i>Checkstop</i>	88
3.6.5	<i>OCC Detects an Error Requiring Reset</i>	88
<b>4</b>	<b>OCC BOOT AND CODE UPDATE PROCESS</b>	<b>90</b>
4.1	OCC LOAD/START PROCESS ON BMC WITH PHYP	90
4.2	OCC LOAD/START PROCESS ON BMC WITH OPAL	90
4.3	ADDITIONAL BMC HANDLING AFTER OCCs ACTIVE WITH POWERVM	90
<b>5</b>	<b>FREQUENCY POINTS</b>	<b>91</b>
5.1	CONFIGURATION FILE	92
<b>6</b>	<b>OCC POLL RESPONSE SENSOR DATA FORMAT DEFINITIONS</b>	<b>93</b>
6.1	TEMPERATURE SENSORS (“TEMP”)	93
6.2	FREQUENCY SENSORS (“FREQ”)	95
6.3	POWER SENSORS (“POWER”)	96
6.3.1	<i>System has APSS (Master only)</i>	96
6.3.2	<i>No APSS (All OCCs Report)</i>	97
6.4	POWER CAPS (“CAPS”)	98
6.5	EXTENDED OCC DATA (“EXTN”)	99
6.5.1	<i>Extended OCC Sensors List</i>	99
<b>7</b>	<b>OCC INTER-CHIP COMMUNICATION</b>	<b>101</b>
<b>8</b>	<b>OCC TO PGPE COMMUNICATION</b>	<b>102</b>
8.1	IPC COMMANDS OCC TO PGPE	102
8.1.1	<i>Start/Stop Pstate Protocol</i>	103
8.1.2	<i>Pstate Clip Update</i>	104
8.1.3	<i>Set PMCR</i>	105
8.1.4	<i>WOF Control</i>	106
8.1.5	<i>WOF VRT</i>	107
<b>9</b>	<b>POWER MANAGEMENT</b>	<b>108</b>
9.1	BMC POWER AND THERMAL MANAGEMENT SETTINGS	108
9.1.1	<i>Fan Control</i>	108

9.1.2	<i>Entity Manager Consumed by BMC (Mode, IPS).....</i>	<i>108</i>
9.1.3	<i>MRW Consumed by HTMGT (Power cap, thermal limits...).....</i>	<i>108</i>
9.2	POWER MANAGEMENT SETTINGS (FSP) .....	110
9.3	POWERVM SYSTEM POWER AND PERFORMANCE MODES .....	110
9.4	USER POWER CAPPING.....	110
9.5	IDLE POWER SAVER.....	111
<b>10</b>	<b>MANUFACTURING IMPACTS .....</b>	<b>114</b>
10.1	MFG TEST COMMANDS .....	114
10.1.1	<i>Processor Auto-Slew with OPAL.....</i>	<i>114</i>
10.2	PSTATE TABLE BIAS .....	114
10.3	ENABLE/DISABLE OCC CONTROL .....	115
10.3.1	<i>Observation State (Disable OCC) Change Process.....</i>	<i>115</i>
10.3.2	<i>Characterization State Change Process .....</i>	<i>115</i>
10.3.3	<i>Active State (Enable OCC) Change Process .....</i>	<i>115</i>
10.4	EXTERNAL VOLTAGE AND FREQUENCY BIAS.....	115
10.4.1	<i>Writing Voltage.....</i>	<i>116</i>
<b>11</b>	<b>OCC MAIN MEMORY LAYOUT .....</b>	<b>117</b>
11.1	HOMER .....	118
11.2	OCC COMMON IMAGE .....	119
<b>12</b>	<b>OCC-OPAL/PHYYP INTERFACE.....</b>	<b>120</b>
12.1	OCC-OPAL/PHYYP SHARED MEMORY INTERFACE .....	120
12.1.1	<i>Parameter Table Definition.....</i>	<i>126</i>
12.2	OPAL-OCC COMMAND/RESPONSE INTERFACE .....	130
12.2.1	<i>OPAL-OCC Command Buffer .....</i>	<i>130</i>
12.2.2	<i>OPAL-OCC Response Buffer .....</i>	<i>131</i>
12.2.3	<i>OPAL-OCC Command/Response Sequence.....</i>	<i>132</i>
12.2.4	<i>Error Handling.....</i>	<i>133</i>
12.2.4.1	<i>Timeout for Command Processing .....</i>	<i>133</i>
12.2.4.2	<i>Response Failures .....</i>	<i>133</i>
12.2.5	<i>OPAL-OCC Commands.....</i>	<i>134</i>
12.2.5.1	<i>AMESTER Pass Thru – NOT SUPPORTED.....</i>	<i>135</i>
12.2.5.2	<i>Clear Sensor Data.....</i>	<i>138</i>
12.2.5.3	<i>Set Power Cap in Band.....</i>	<i>139</i>
12.2.5.4	<i>Write Power Shifting Ratio .....</i>	<i>140</i>
12.2.5.5	<i>Select Sensor Groups .....</i>	<i>141</i>
12.2.5.6	<i>WOF Control.....</i>	<i>142</i>
12.3	OCC MAIN MEMORY SENSOR DATA.....	143
12.3.1	<i>OCC N Sensor Data Block Layout (150kB).....</i>	<i>143</i>
12.3.1.1	<i>Sensor Data Header Block (1kB).....</i>	<i>144</i>
12.3.1.2	<i>Sensor Names (50kB).....</i>	<i>145</i>
12.3.1.3	<i>Sensor Readings Ping and Pong Buffers (40kB each).....</i>	<i>147</i>
12.3.1.3.1	<i>sensor_structure_version = 0x01 (Full Reading) .....</i>	<i>148</i>
12.3.1.3.2	<i>sensor_structure_version = 0x02 (Counter) .....</i>	<i>148</i>
12.3.2	<i>Main Memory OCC Sensor List.....</i>	<i>149</i>
12.3.2.1	<i>Performance Sensors .....</i>	<i>149</i>
12.3.2.2	<i>Power Sensors.....</i>	<i>150</i>

12.3.2.3	<i>Frequency Sensors</i>	150
12.3.2.4	<i>Utilization Sensors</i>	150
12.3.2.5	<i>Temperature Sensors</i>	150
12.3.2.6	<i>Voltage Sensors</i>	151
12.3.2.7	<i>Current Sensors</i>	151
12.3.3	<i>Other OCC Sensors for AMESTER</i>	152
12.4	PIB I2C MASTER LOCK	153
12.4.1	<i>OCC Flags Register</i>	153
12.4.2	<i>OCC Miscellaneous Register – Interrupt to host</i>	153
12.4.2.1	<i>External Interrupt Reason Defines</i>	153
12.4.3	<i>I2C Lock Use Cases</i>	154
12.4.3.1	<i>Host Wants Lock</i>	154
12.4.3.2	<i>OCC Actions</i>	154
12.4.3.3	<i>Host Hung Case</i>	155
12.4.3.4	<i>OCC Hung Case</i>	155
12.5	GPU RESET HANDLING	155
12.5.1	<i>GPU Numbering</i>	156
<b>APPENDIX A. RETURN CODES</b>		<b>157</b>
<b>APPENDIX B. OCC STATES</b>		<b>159</b>
<b>APPENDIX C. SYSTEM POWER AND PERFORMANCE MODES</b>		<b>161</b>
<b>APPENDIX D. (H)TMGT-OCC COMPONENT IDS</b>		<b>164</b>

---

# 1 Overview

This document covers the firmware interfaces to the OCC for FSP, BMC, Host, and OPAL.

---

## 1.1 Terms

**APSS** – Analog Power Subsystem Sweep, provide real time power measurements of voltage rails.

**BMC** – Baseboard Management Controller

**DCMI** – Data Center Manageability Interface

**Host Boot** – Code that runs on the processor that initializes it. Equivalent to BIOS

**HTMGT** – Host TMGT (Thermal Management). Exists on BMC systems only. Specifically, the thermal management code piece that runs on the processor and initializes and handles errors from the OCC(s).

**KVM** – Kernel-based Virtual Machine. Open source virtualization software for Linux.

**OCC** – On Chip Controller. Embedded 405 processor with 768K SRAM. Provide real time power and thermal monitoring. Monitoring times to allow for fast response.

**OPAL** – Open Power Abstraction Layer

**TMGT** – TMGT (Thermal Management). FSP systems only. The thermal management code piece that runs on the FSP that initializes, monitors and handles errors from the OCC(s)

**WOF** – Workload Optimized Frequency. OCC algorithm that monitors the current (amperage) being drawn by the present workload set on a given socket and the number of cores active within that socket to allow for the frequency of operational cores to be boosted up to a higher “ultra turbo” frequency point without exceeding the current limits of the regular subsystem.



---

## 1.2 OCC Functional Overview

OCC requirements:

- Keep the system thermally safe by monitoring memory and processor temperatures
  - Provide temperatures to the BMC or FSP for fan control
  - Throttle memory if a memory temperature reaches a specified throttle temperature point
  - Lower frequency and voltage if a processor reaches a specified throttle temperature point.
- Keep the system power safe by monitoring total system power and quick power drop line
  - Lower frequency and voltage to keep power below the current system power limit in effect
  - Take action when the quick power drop line is asserted by changing the memory throttles and current power limit to the quick power drop settings
- O/S Frequency Controlled Systems: The OCC will never directly set a Pstate (voltage/frequency). The o/s will have direct control for setting Pstates. The OCC will write the Pstate range and table per the defined [OCC-OPAL Shared Memory Interface](#). OCC will update the “throttle” status byte in this interface when limiting the hardware maximum Pstate due to power or thermal reason.
- Provide power, thermal and frequency sensor data for external display

### 1.3 OCC405 SRAM Layout

#### OCC405 SRAM Layout

0xFFFF40000 - 0xFFFF5FFF	OCC405 Code and Data	728KB (463KB in P9)	768KB
0xFFFF6000 - 0xFFFF60FF	GPE0/1 Shared Data	256B	
0xFFFF6100 - 0xFFFF61FF	WOF Ping Buffer	256B	
0xFFFF6200 - 0xFFFF62FF	WOF Pong Buffer	256B	
0xFFFF6300 - 0xFFFF63FF	Global Data Pointers	256B	
0xFFFF6400 - 0xFFFF87FF	Error Trace Buffer	9KB	
0xFFFF8800 - 0xFFFFABFF	Informational Trace Buffer	9KB	
0xFFFFAC00 - 0xFFFFCFFF	Important Trace Buffer	9KB	
0xFFFFD000 - 0xFFFFDFFF	FSP/BMC Command Buffer	4KB	
0xFFFFE000 - 0xFFFFEFFF	FSP/BMC Response Buffer	4KB	
0xFFFFF000 - 0xFFFFFBBF	Reserved (Expand response buffer?)	4,032B	
0xFFFFFFC0 - 0xFFFFFFF	Reserved for SRAM Control Registers *** To boot OCC HWP must write SRBV3 (0xFFFFFFFC) with branch instruction to 0xFFFF40000	64B	

---

## 1.4 OCB Channels

There are 4 channels. Three channels are used for communication to the OCC. The channels must be configured to linear mode for reading/writing SRAM or circular mode to generate attentions to the OCC. On BMC systems Host boot will configure all channels and on FSP systems the FSP HWSV code will configure all channels. Some additional setup is done by the OCC to support a circular channel. A linear channel can only handle one request at a time, to avoid collisions each user needing access to SRAM must have its own dedicated channel.

Channel	Mode	Usage
0	Linear	(H)TMGT use only. FSP Systems: TMGT to write command buffer in SRAM and read response buffer from SRAM. BMC Systems: HTMGT to read OCC error logs from SRAM.
1	Circular	Write only from BMC/FSP and HTMGT to generate attentions to the OCC.
2	Linear	BMC use only. Used by BMC to write command buffer in SRAM and read response buffer from SRAM.
3	Linear	Reserved for internal use.

---

## 1.5 Host Requirements for BMC systems

The following are required from Linux/OPAL and PHYP to support OCC on BMC systems.

- Support for OCC-Host Interrupt. See [OCC to Host Interrupt](#) section for details.
- PLDM message for OCC reset request. See [BMC Request for OCC Reset](#) section for details.
- PLDM message to update “OCC Active Sensor”. See [OCC Reset Procedure](#) section for use case.
- OPAL only: Interface for manufacturing to disable and enable OCC. See [Enable/Disable OCC Control](#) section for details.
- OPAL only: Provide a pass thru interface to send a generic command buffer to HTMGT and receive a response buffer from HTMGT dumping out the response to stdout in a hex dump format.
- OPAL only: Full support for the [OCC-OPAL](#) interface.

---

## 1.6 Attentions/Interrupts

### 1.6.1 To OCC

To generate an attention to the OCC a write to the OCB in circular mode will be used. There is no response to an attention. The data written to the OCB is limited to 8 bytes and will indicate who is sending the attention and what the attention is for. The general format for attentions to the OCC:

<b>Byte 1</b>	<b>Byte 2</b>	<b>Bytes 3 thru 8</b>
Sender ID	Attn Type	Cmd Specific Data

#### 1.6.1.1 Command Write Attention Type = 0x01

This attention type is used to inform the OCC that a command is ready to be processed. The OCC determines where to read the command buffer based on the sender ID.

##### Format:

<b>Byte 1</b>	<b>Byte 2</b>	<b>Bytes 3 thru 8</b>
Sender ID	Attn Type = 0x01	Reserved = 000000000000

**Byte 1: Sender Id.** One byte to identify the sender of the command  
    **0x01** – FSP (TMGT)  
    **0x10** – HTMGT  
    **0x20** – BMC

**Byte 2: Attention Type = 0x01.** Command Write Attention

## 1.6.2 OCC to Host

Each OCC has interrupt capability to the Host by using the PSIHB complex.

One “service required” interrupt is required for the OCC to inform HTMGT to check status. In response to this interrupt HTMGT will send a poll command to determine what service the OCC requires, this is how HTMGT is informed of an error log to collect.

1. OCC sets bits 0 and 1 of OCB\_OCI\_OCCMISC SCOM register.
  - Bit 0 (OCB\_OCI\_OCCMISC\_CORE\_EXT\_INTR) to generate the interrupt
  - Bit 1 to indicate source/reason for interrupt is OCC-HTMGT Service Required
2. The interrupt is controlled by the XIVR – OCC register that PHYP/OPAL must have previously setup. NOTE: OCC is running before OPAL, when setup is complete there may be an interrupt already pending that must be handled by OPAL.
3. PHYP/OPAL sees the interrupt and recognizes the reason of OCC-HTMGT Service Required and calls HTMGT to service the OCC by calling process\_occ\_error with the OCC chip ID that generated the interrupt. NOTE: This does NOT necessarily mean there is an error, the OCC will generate an informational log with sensor data every 24 hours.
4. PHYP/OPAL clears the SCOM bits so that OCC can generate an interrupt again as needed.

See [External Interrupt Reason Defines](#) section in this document for all reasons for OCC to generate this interrupt.

## 1.6.3 OCC to FSP (TMGT)

There is no attention from the OCC to the FSP. TMGT will be polling all OCCs at least every 15s to determine health of the OCC and to retrieve any OCC error logs. No need for a response ready, TMGT will act the same as BMC design and poll the OCC for response to all commands. This keeps the same design between BMC and FSP command handling.

## 1.6.4 BMC to Host

The P10 chip can be interrupted by sending a PLDM message. The BMC will alert host for the following:

- Request an OCC reset. Conditions requiring an OCC reset are defined in [BMC Detected Reasons for OCC Reset](#) section.

---

## 1.7 Commands

Each sender must be assigned a unique 4K pre-defined fixed memory location for a command buffer to send (write) commands to an OCC and a unique 4K pre-defined fixed location for a response buffer to read response data. After writing a command to the command buffer a data write attention must be written to the OCB to generate an attention to inform the OCC that there is a command to process. When the OCC receives a command, it will first write the response buffer return status byte to “In Process” to allow the sender to know that the command is in process, but the response is not ready. When the OCC is finished processing the command it will update the return status last.

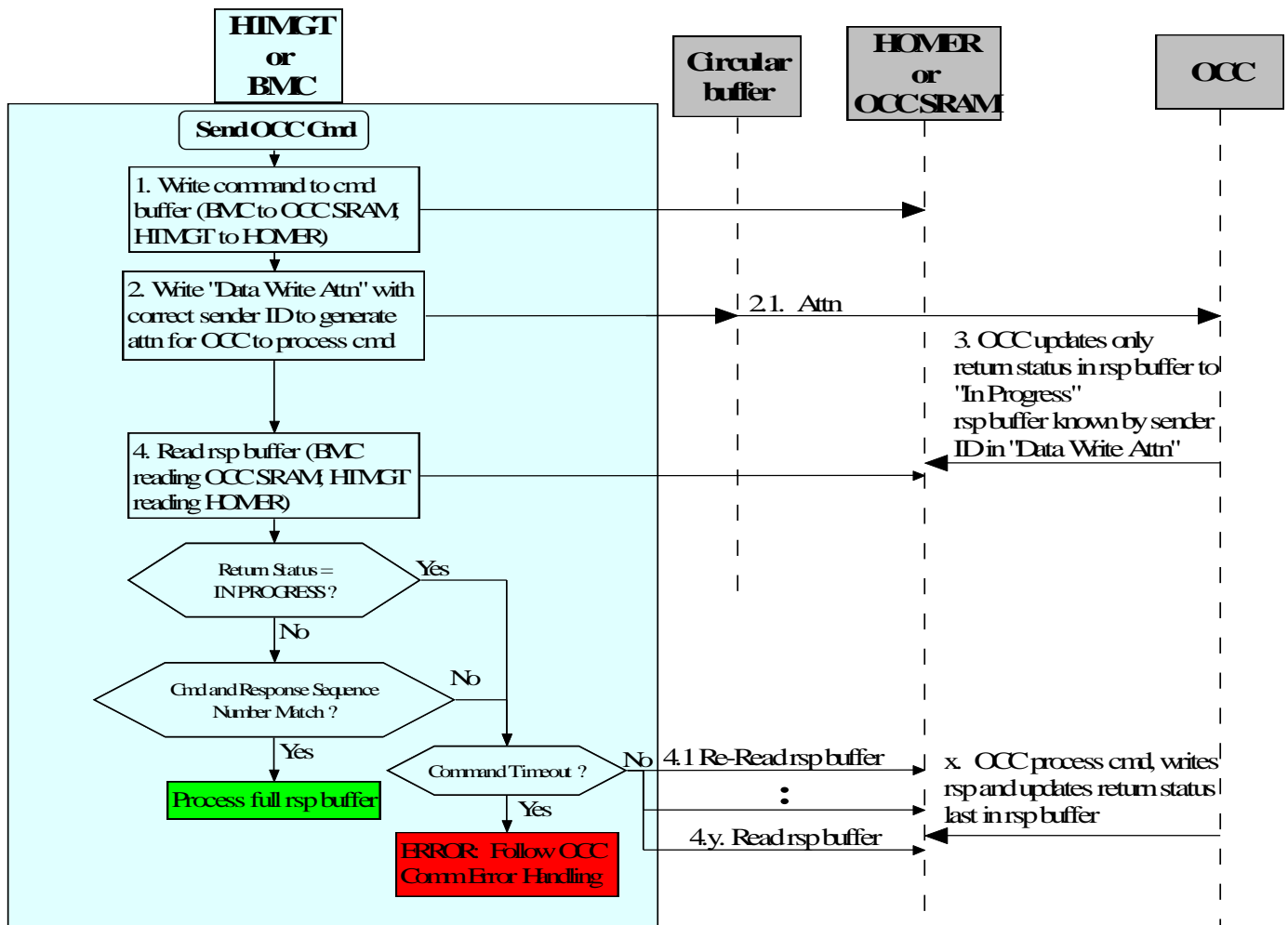
### 1.7.1 Command and Response Buffer Locations

These are the pre-defined 4K locations reserved for a command and response buffer for each sender:

	<b><i>Command Buffer</i></b>	<b><i>Response Buffer</i></b>
<b><i>BMC / FSP (TMGT)</i></b>	OCC SRAM 0xFFFFD000 – 0xFFFFDFFF	OCC SRAM 0xFFFFE000 – 0xFFFFEFFF
<b><i>HTMGT</i></b>	HOMER 0x000E0000 – 0x000E0FFF	HOMER 0x000E1000 – 0x000E1FFF

## 1.7.2 OCC Command/Response Sequence

NOTE: The OCC can only handle one command at a time across all senders, this can delay the time it takes for the OCC to update the response buffer to "In Progress" if the OCC is processing a command from a different sender. To handle a sender reading the return status before OCC updated it to "In Progress" the sender should also keep re-reading the response buffer if the response sequence number does not match. Re-reading the response buffer should continue until the command is no longer in progress and the sequence numbers match or until a command timeout is hit. See [Command Summary Table](#) for recommended timeouts.





### 1.7.3 Command Format

The command format is the same regardless of who the sender is (BMC/FSP, HTMG).T).

NOTE: SRAM space is 4K total the command header/checksum reduces the number of data bytes.

#### **Format:**

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	.....thru.....		Byte N-2	Byte N-1	Byte N
Seq. Number	Cmd Type	Data Length MSB	Data Length LSB	Data 1	Data 2	.....	Data M	Checksum MSB	Checksum LSB

**Byte 1: Sequence Number.** One byte unsigned (0x00 follows 0xFF) sequence number.

**Byte 2: Command Type.** The value of this byte indicates what type of command this is. See the Command chapter in this document for a list of valid values.

**Byte 3: Data Length MSB.** MSB of 2 byte data length, 0-M, maximum value of M is 4090 bytes.

**Byte 4: Data Length LSB.** LSB of 2 byte data length

**Byte 5 to N-2: Data Bytes.** 0-4090 data bytes, meaning depends on the command type byte. Definition of these bytes can be found in the Command chapter in this document under the definition of each command type.

**Byte N-1: Checksum MSB.** MSB of 2 byte checksum, checksum is the two byte sum (ignoring overflow) of all bytes starting with and including the sequence number.

**Byte N: Checksum LSB.** LSB of 2 byte checksum

### 1.7.4 Response Format

The response format is the same regardless of who the sender is (BMC/FSP or HTMGT).

NOTE: SRAM space is 4K total the response header/checksum reduces the number of data bytes.

#### **Format:**

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	.....thru.....		Byte N-2	Byte N-1	Byte N
Seq. #	Cmd Type	Return Status	Data Length MSB	Data Length LSB	Data 1	Data 2	.....	Data M	Check-sum MSB	Check-sum LSB

**Byte 1: Sequence Number.** Same sequence number value found in the command packet that this return packet is for.

**Byte 2: Command Type.** The value of this byte indicates what type of command this return packet is for. This will be the same value as the command type found in the command packet that this return packet is for.

**Byte 3: Return Status.** The value of this byte indicates the status of the command. Upon receiving a command, the OCC will first write this byte to 0xFF to indicate that the OCC is processing the command. Once the OCC finishes processing the command this byte will be updated last and represent the success or failure of the command. See Appendix A for a full list of values.

**Byte 4: Data Length MSB.** MSB of 2 byte data length 0-M, maximum value of M is 4089 bytes.

**Byte 5: Data Length LSB.** LSB of 2 byte data length

**Byte 6 to N-2: Data Bytes.** 0-4089 bytes of return data, meaning depends on the command type byte. Definition of these bytes can be found in the Command chapter in this document under the definition of each command type's return definition.

**Byte N-1: Checksum MSB.** MSB of 2 byte checksum, checksum is two byte sum (ignoring overflow) of all bytes starting with and including the sequence number.

**Byte N: Checksum LSB.** LSB of 2 byte checksum

### 1.7.5 Error Response Data Packet

When OCC returns any of the non-successful return codes listed in Appendix A the return packet will be the following:

<b>Sequence Number</b>	xx
<b>Command Type</b>	xx
<b>Return Status</b>	Non-Success See Appendix A for list of all non-successful return codes.
<b>Data Length</b>	0x0001
<b>Data</b>	There is 1 data byte returned: <b>Byte 1:</b> Error log id – Any non-zero value indicates the error log id corresponding to the OCC error log that was created for this command failure. The OCC may return 0 if no error log was generated.
<b>Checksum</b>	xxxx

The OCC returns the 1 byte error log id of the error log that it created for this failure. An error log id of 0x00 can be used to indicate no error log if the OCC did not generate an error log. The sender should retry the command once if the return code may indicate a transmission failure that a retry may help i.e. checksum failure or internal error. If the command is not to be retried or fails again on the retry then the sender should create an error log and put this error log id in it to allow correlation with the OCC command failure error log for debug. The HTMGT created error log is for the error on what was trying to be accomplished by the command i.e. a failure to change state with a “Set OCC State” command. The OCC error log for the command failure will be reported via the same path as all other OCC detected errors defined in “OCC Error Logging” section.

## 1.7.6 Command Summary Table

### **Sender**

Expected sender(s) for the command. The command will not be rejected if sent by a user not listed, but unexpected system behavior may result.

### **Timeout**

- The timeout is a recommended time to wait for the OCC response to be ready before taking error action.
- Timeout is from when the data write attention is sent to the OCC to when the OCC has completed processing the command by writing the response buffer.
- FSP Systems: Only one sender (TMGT) of OCC commands. The timeout is command specific to handle different processing times that an OCC may need to accomplish each command. If the response ready attn is not received from the OCC by the timeout TMGT will attempt to read the response buffer to see if a response is available to handle case that the response ready attn was missed. If there is a failure reading the response TMGT will retry the command
- BMC Systems: Possible of two senders (HTMGT, BMC) of OCC commands. The OCC can only process one command at a time across all senders. This timeout includes worst case time for the longest processing command to handle when a command from a different sender is being processed first. Both HTMGT and BMC should use this timeout value.

### **OCC State**

- Defines OCC states that the command is supported in
  - Sby = Standby
  - O = Observation/Characterization
  - A = Active
  - Safe
- A command sent to an OCC in a state that does not support the command will be rejected by the OCC with PRESENT STATE PROHIBITS

### **Supported By**

- A master OCC is also considered a slave and will not reject any command
- A slave OCC must reject a command that is only to be supported by a master OCC

<b>Command</b>		<b>Sender</b>	<b>Timeout</b>		<b>OCC State</b>	<b>Supported By</b>	
			<b>FSP</b>	<b>BMC</b>		<b>Master OCC</b>	<b>Slave OCC</b>
0x00 Poll	Poll the OCC for status and sensor data.	BMC HTMGT TMGT	5s	20s	Sby, O, A, Safe	Y	Y
0x12 Clear Error Log	Tell OCC to clear an error log, this is an ack that error log was read	HTMGT TMGT	5s	20s	Sby, O, A, Safe	Y	Y

<b>Command</b>		<b>Sender</b>	<b>Timeout</b>		<b>OCC State</b>	<b>Supported By</b>	
			<b>FSP</b>	<b>BMC</b>		<b>Master OCC</b>	<b>Slave OCC</b>
0x20 Set Mode and State	Set the OCC state and/or system power management mode	BMC HTMGT TMGT	15s	20s	Sby, O, A	Y	N
0x21 Config Data	0x03: Set OCC Role	HTMGT TMGT	5s	20s	Sby	Y	Y
	0x04: APSS Config	HTMGT TMGT	5s	20s	Sby, O, A	Y	Y
	0x05: Memory Config.	HTMGT TMGT	5s	20s	Sby, O, A	Y	Y
	0x07: Power Cap Data	HTMGT TMGT	5s	20s	Sby, O, A	Y	N
	0x0F: System Config.	HTMGT TMGT	5s	20s	Sby, O, A	Y	Y
	0x11: Idle Power Saver	BMC TMGT	5s	N/A	Sby, O, A	Y	N
	0x12: Memory Throttles	HTMGT TMGT	5s	20s	Sby, O, A	Y	Y
	0x13: Thermal Control Thresholds	HTMGT TMGT	5s	20s	Sby, O, A	Y	Y
	0x14: AVS Bus Config	HTMGT TMGT	5s	20s	Sby, O, A	Y	Y
	0x15: GPU	HTMGT TMGT	5s	20s	Sby, O, A	Y	Y
0x22 Set User Cap	Set a User Power Cap	BMC HTMGT TMGT	5s	20s	Sby, O, A	Y	N
0x25 Reset Prep	Prepare OCC to be reset	HTMGT TMGT	15s	20s	Sby, O, A, Safe	Y	Y
0x30 Send Ambient	Send Ambient Temperature to OCC	BMC TMGT	5s	20s	Sby, O, A	Y	Y
0x40 Debug Pass Through	Used for debug use only.	HTMGT TMGT	15s	20s	Sby, O, A, Safe	Y	Y
0x41 AME Pass Through	Used for debug use only by amester.	HTMGT TMGT	10s	20s	Sby, O, A, Safe	Y	Y

<b>Command</b>		<b>Sender</b>	<b>Timeout</b>		<b>OCC State</b>	<b>Supported By</b>	
			<b>FSP</b>	<b>BMC</b>		<b>Master OCC</b>	<b>Slave OCC</b>
0x42 Get Field Debug Data	Used to collect additional data from OCC for hw errors detected by host or FSP.	HTMGT TMGT	10s	20s	Sby, O, A, Safe	Y	Y
0x53 Mfg Test Cmd	0x02: Run/Stop Slew Between Two Frequency Points	HTMGT TMGT	15s	20s	A	Y	N
	0x05: List Sensors	HTMGT TMGT	15s	20s	O, A	Y	Y
	0x06: Get Sensor Information	HTMGT TMGT	15s	20s	O, A	Y	Y
	0x07: Enable/Disable Oversubscription Emulation	HTMGT TMGT	15s	20s	O, A	Y	N
	0x09: Run/Stop Memory Slew	HTMGT TMGT	15s	20s	A	Y	Y
	0x0B: Read Generated Pstate Table	HTMGT TMGT	15s	20s	O, A	Y	Y
	0x0F: Select WOF VRT	HTMGT TMGT	15s	20s	A	Y	Y

---

## 2 **Command Definitions**

NOTE: For all command responses the return packet is for a successful response. If the command fails i.e. Returning a non-successful return code as listed in Appendix A, the return packet will be the error return packet that is described in the “Error Handling” chapter of this document.

---

## 2.1 Poll

This command is used to read status and sensor data from the OCC.

<b><i>TMGT</i></b>	This command is used by TMGT to periodically poll the OCC for status and as a heartbeat to make sure the OCC is functional. The poll is also used in response to a “service required” attention from the OCC.
<b><i>BMC</i></b>	The BMC will send this periodically to read sensor data for fan control and verify that the OCC is functional.
<b><i>HTMGT</i></b>	HTMGT will send this in response to getting a “service required” interrupt from an OCC or an OCC error handling indication from BMC.

### **Poll Command Packet:**

<b><i>Sequence Number</i></b>	Xx
<b><i>Command Type</i></b>	0x00
<b><i>Data Length</i></b>	0x0001
<b><i>Data</i></b>	There is 1 data byte: <b>Byte 1:</b> Version – Indicates what poll response version is being requested. <b>0x20</b> = Status and Sensor Poll
<b><i>Checksum</i></b>	Xxxx



**Poll Return Packet:**

<b>Sequence Number</b>	Xx						
<b>Command Type</b>	0x00						
<b>Return Status</b>	0x00 = Success See Appendix A for list of all non-successful return codes.						
<b>Data Length</b>	Variable. Minimum of 40 bytes to maximum of 4089.						
<b>Data</b>	<b>Version = 0x20</b> See <a href="#">Version 0x20 Poll Response Data Definition</a> section for details.						
	<b>Byte 1</b>	<b>Byte 2</b>	<b>Byte 3</b>	<b>Byte 4</b>	<b>Byte 5</b>	<b>Byte 6</b>	<b>Byte 7</b>
	Status	Ext Status	OCCs present	Config data	OCC State	Mode	IPS Status
	Error Log Start Address				Error Log Length	Elog Source	GPU config
	OCC Code Level (16 bytes)						
	"SENSOR" (bytes 33-38)					# of sensor data blocks	Sensor data block header version = 0x01
	Sensor Data Blocks (Variable length. Byte 41 thru end of response data length)						
<b>Checksum</b>	Xxxx						

## 2.1.1 Version 0x20 Poll Response Data Definition

**Byte 1: Status** – Indicates current general status of the OCC. Bit defined:

Bit 0 (msb)	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7 (lsb)
Master OCC			OCC PMCR Owner	SIMICS		OCC Observ ation Ready	OCC Active Ready

**Master OCC** – 1 indicates that the OCC is running as a master OCC. 0 indicates the OCC is running as a slave only.

HTMGT Handling: Verify that each OCC has this bit set correctly based on what was sent in “Set OCC Role” config data command. HTMGT will log an error and reset the OCCs if this is not reported correctly from any OCC.

BMC Handling: Used to know which OCC the master is to process master only data/commands from.

**TBD** – TBD

HTMGT Handling: None, FYI only.

BMC Handling: None.

**OCC PMCR Owner** – 1 indicates OCC owns setting PMCR (Pstates) this means the system is PowerVM and OCC is in active state. 0 indicates that the OCC does not own setting the PMCR i.e. OPAL is present and owns PMCR (OCC will still clip Pmax due to power/thermal if in active state) or the OCC is in characterization state (OCC will not do any Pmax clipping).

(H)TMGT Handling: None, FYI only.

BMC Handling: None.

**SIMICS** – 1 indicates running in SIMICS environment, known by bit 63 of OCC Flags register. Some OCC function is not supported in SIMICS and timings will not be accurate.

(H)TMGT Handling: None, FYI only.

BMC Handling: None.

**OCC Observation Ready** – 1 indicates that the OCC has received all needed data to support observation state

HTMGT Handling: Used during initialization to know that the OCC has all needed config data to make a state change to Observation.

BMC Handling: None.

**OCC Active Ready** – 1 indicates that the OCC has received all needed data to support the full actuation “active” state. Characterization state requires active state data (i.e. frequency points) and will only change to characterization state if active ready.

HTMGT Handling: Used during initialization to know that the OCC has all needed config data to make a state change to Active or Characterization state.

BMC Handling: None.

**Byte 2: Extended Status** – Continuation of the current general status of the OCC. Bit defined:

Bit 0 (msb)	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7 (lsb)
DVFS due to proc OT	DVFS due to power	Mem Throttle OT	Quick Power Drop	DVFS due to Vdd OT	GPU2 Throttle	GPU1 Throttle	GPU0 Throttle

**DVFS due to proc OT** – 1 indicates that the OCC has currently clipped max Pstate below WOF base due to a processor over temperature condition

HTMGT Handling: None.

BMC Handling: If “Processor Frequency Limited due to Over Temperature” sensor exists then the BMC should assert the “Processor Frequency Limited due to over temperature” sensor for this processor when this bit goes from ‘0’ to ‘1’ between two polls and de-assert when this bit goes from ‘1’ to ‘0’ between two polls

**DVFS due to power** – 1 indicates that the OCC has currently clipped max Pstate below WOF base due to reaching the current power cap limit.

HTMGT Handling: None.

BMC Handling: If “Processor Frequency Limited due to Power” sensor exists then the BMC should assert the “Processor Frequency Limited due to Power” sensor for this processor when this bit goes from ‘0’ to ‘1’ between two polls and de-assert when this bit goes from ‘1’ to ‘0’ between two polls

**Mem Throttle OT** – 1 indicates that the OCC has currently throttled memory due to a memory over temperature condition.

HTMGT Handling: None.

BMC Handling: If “Memory Throttled due to Over Temperature” sensor exists then the BMC should assert the “Memory Throttled due to Over Temperature” sensor for this processor when this bit goes from ‘0’ to ‘1’ between two polls and de-assert when this bit goes from ‘1’ to ‘0’ between two polls

**Quick Power Drop** – 1 indicates that the quick power drop line is asserted. 0 indicates

quick power drop line is not asserted.

HTMGT Handling: None.

BMC Handling: If “Quick Power Drop” sensor exists then the BMC should assert the “Quick Power Drop” sensor for this processor when this bit goes from ‘0’ to ‘1’ between two polls and de-assert when this bit goes from ‘1’ to ‘0’ between two polls

**DVFS due to Vdd OT** – 1 indicates that the OCC has currently clipped max Pstate below WOF base due to a VRM Vdd over temperature condition

HTMGT Handling: None.

BMC Handling: If “Processor Frequency Limited due to VRM Over Temperature” sensor exists then the BMC should assert the “Processor Frequency Limited due to VRM over temperature” sensor for this processor when this bit goes from ‘0’ to ‘1’ between two polls and de-assert when this bit goes from ‘1’ to ‘0’ between two polls

**GPU2 Throttle** – ‘1’ indicates that GPU2 monitored by this OCC is currently throttled

HTMGT Handling: None.

BMC Handling: BMC should assert the “Proc x GPU2 Throttled” sensor for GPU2 behind this processor x when this bit goes from ‘0’ to ‘1’ between two polls and de-assert when this bit goes from ‘1’ to ‘0’ between two polls

**GPU1 Throttle** – ‘1’ indicates that GPU1 monitored by this OCC is currently throttled

HTMGT Handling: None.

BMC Handling: BMC should assert the “Proc x GPU1 Throttled” sensor for GPU1 behind this processor x when this bit goes from ‘0’ to ‘1’ between two polls and de-assert when this bit goes from ‘1’ to ‘0’ between two polls

**GPU0 Throttle** – ‘1’ indicates that GPU0 monitored by this OCC is currently throttled

HTMGT Handling: None.

BMC Handling: BMC should assert the “Proc x GPU0 Throttled” sensor for GPU0 behind this processor x when this bit goes from ‘0’ to ‘1’ between two polls and de-assert when this bit goes from ‘1’ to ‘0’ between two polls

**Byte 3: OCCs Present** – Bit defined lsb is OCC “0” which is the chip id.

Master OCC Response: May have multiple bits set; has bit set for every OCC it sees (including itself).

Slave OCC Response: Sets only one bit for the chip id it is.

HTMGT/TMGT Handling: HTMGT will verify that the master OCC is reporting the same OCCs that HTMGT is communicating with and that no more than one

slave OCC is reporting the same chip id. An error and reset will occur if either of these conditions is not met.

BMC Handling: Verify number of OCCs and request OCC reset if there is a mismatch.

**Byte 4: Configuration Data needed** – This byte indicates the format value of the “Configuration Data” command that OCC is requesting to be sent. 0X00 indicates no request.  
HTMGT/TMGT Handling: When non-zero, HTMGT will send a “Configuration Data” command with this format value.

BMC Handling: None

**Byte 5: Current OCC State** – Indicates the current OCC state that the OCC is in. See Appendix B for valid OCC states.

HTMGT/TMGT Handling: This byte will be checked on the first poll after sending a set state command. (H)TMGT will verify that the OCC is reporting the new state and will log an error and reset the OCCs if it is not.

BMC Handling: Always send even with OPAL? Support systems booting in either OPAL or PHYP? PowerVM: Must check this byte for a change to “Active” state. When this byte changes to “Active” the BMC must send the following commands:

- Set Mode and State to set the customer mode
- Idle Power Saver Settings

Must check this byte for “Safe” state. When in safe state the BMC should do the following:

- Ignore the sensor data starting at byte 33. The OCC is not updating sensors while in safe state.
- Request for OCC reset if OCC is in safe state for one minute and the “OCC Active” sensor is still TRUE.

**Byte 6: Current System Power Management Mode** – Indicates the system power management mode, 0x00 if the system is under O/S Frequency control. See Appendix C for valid system power management modes.

**Byte 7: Current Idle Power Saver Status** – PowerVM system only, this byte will be 0x00 for OPAL based systems. See [Idle Power Saver Handling chapter](#) for info on how these bits are used by TMGT/BMC.

Bit 0 (msb)	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7 (lsb)
						Idle Power Saver Active	Idle Power Saver Enabled

**Bits 0:5 Reserved '000000'**

**Bit 6: Idle Power Saver Active** – This indicates when OCC is actively in Idle Power Saver. 1 indicates Idle Power Saver is active.

**Bit 7: Idle Power Saver Enabled** – This should reflect what TMGT/BMC has last sent for “Idle Power Saver” via the Idle Power Saver Settings config data packet. 1 indicates Idle Power Saver is enabled, 0 indicates disabled. When this is a 0, the Idle Power Saver Active (bit 6) is ignored by TMGT/BMC.

**Byte 8: Error Log Id** – Any non-zero value indicates the log id associated with a PM complex error log to be reported. 0x00 indicates no error log. There must also be an error log start address, error log length and error log source in the same poll response. The same error log id and source will be sent until a clear error log command has been sent for the error log id and source. Error log IDs can be repeated between sources (i.e. there is no coordination of elog ID between OCC and PGPE) the elog id AND source makes the error log unique. (H)TMGT must handle this and not log the same error more than once or miss logging the same elog ID from two different sources.

HTMGT/TMGT Handling: Full support to collect and report an error log from the PM complex

BMC Handling: None

**Bytes 9-12: Error Log Start Address** – Only valid when error log id in previous byte is not 0.

HTMGT/TMGT Handling: Full support to collect and report an error log from the PM complex

BMC Handling: None

**Bytes 13-14: Error Log Length** – Length of total error log starting at the error log start address thru the last byte of error log user data.

HTMGT/TMGT Handling: Full support to collect and report an error log from the PM complex

BMC Handling: None

**Byte 15: Error Log Source** – Indicates who is reporting the error log.

0x00 = OCC 405 (component ID 0x2A)

0x10 = PGPE (component ID 0x2E)

0x20 = XGPE (component ID 0x2F)

0x40 = QME (component ID 0x29)

HTMGT/TMGT Handling: Full support to collect and report an error log from the PM complex. Source is used to determine component ID for the MSB of the SRC (i.e. OCC 405 is 0x2A)

BMC Handling: None

**Byte 16: GPU Configuration** – Bit defined representing GPU(s) present monitored by this OCC.

Bit 0 (msb)	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7 (lsb)
Reserved = '00000'					GPU2 presence '1' = present '0' = not present	GPU1 presence '1' = present '0' = not present	GPU0 presence '1' = present '0' = not present

HTMGT Handling: After OCC reaches active state, for each GPU present HTMGT to set GPU present/functional sensor

BMC/TMGT Handling: None

**Bytes 17-32: OCC Code Level** – ASCII String of OCC build level currently running. i.e. "occ830\_082214a"

HTMGT Handling: None

BMC Handling: For future code compatibility checking

---

START SENSOR DATA – Remaining data is for BMC/TMGT use; HTMGT will ignore

---

**Bytes 33-38: "SENSOR"** – 6 byte ASCII eye catcher for start of sensor data

**Byte 39: Number of Sensor Data Blocks** – Indicates number of sensor data blocks in the sensor data blocks section of response data.

**Byte 40: Sensor Data Block Header Version** – Indicates format version of the sensor data block. Currently, only 0x01 is supported.

**Bytes 41-End of Response Data: Sensor Data Blocks** – 1 or more sensor data blocks, indicated by "Number of Sensor Data Blocks" (byte 39). If there is more than 1 sensor data block the next sensor data block immediately follows the previous one. One sensor data block consists of an 8 byte header followed by the sensor data, see [Sensor Data Format Definitions chapter](#) for details on sensor format that follows the 8 byte sensor data block header for each type. NOTE: Some sensor types are only available from the master OCC.

Format of 8 byte Sensor Data Block Header Version 0x01:

Bytes 0x00 thru 0x03	0x04	0x05	0x06	0x07
----------------------	------	------	------	------

Sensor Eye Catcher	Reserved = 0x00	Sensor Format	Sensor Length	Number sensors
--------------------	--------------------	------------------	------------------	-------------------

**Sensor Eye Catcher** – 4 byte ASCII indicating type of sensor data that follow.

Supported values:

“**TEMP**” – Following sensors are for temperature readings. All OCCs (master and slave) will report.

“**FREQ**” – Following sensors are for current frequency. All OCCs (master and slave) will report.

“**POWR**” – Following sensors are for power readings. All OCCs will report if the system does not have an APSS. Only master OCC will report if APSS is present.

“**EXTN**” – Following if for reporting extended OCC data. All OCCs may report.

“**CAPS**” – Following is for reporting power caps. Only master OCC will report.

**Reserve** – 1 byte reserve = 0x00 for future use.

**Sensor Format** – 1 byte indicating format level for the sensor data that follows.

**Sensor Length** – 1 byte indicating length for one sensor in the sensor data that follows.

**Number of Sensors** – 1 byte indicating number of sensors that follows



---

## 2.2 Clear Error Log

This command is used by (H)TMGT as an ack to OCC that the given error log id from error log source from the OCC poll response has successfully been collected. When received the OCC no longer needs to keep this error log and can reuse the SRAM address that this error log id was at. When received with a source outside the 405 the OCC will clear the shared memory area for the source so the PM Hcode knows it can reuse SRAM and report another error.

<b>BMC</b>	Should never send.
<b>(H)TMGT</b>	Sent after (H)TMGT has successfully collected the error log it created from reading SRAM for this error log id.

### Clear Error Log Command Packet:

<b>Sequence Number</b>	Xx
<b>Command Type</b>	0x12
<b>Data Length</b>	0x0004
<b>Data</b>	There are 4 data bytes:  <b>Byte 1:</b> Version – 0x01 <b>Byte 2:</b> Error Log Id – The log id of the error log to be cleared. <b>Byte 3:</b> Error Log Source – The source of the error log to be cleared. <b>Byte 4:</b> Reserved = 0x00
<b>Checksum</b>	Xxxx

**Clear Error Log Return Packet:**

<b><i>Sequence Number</i></b>	Xx
<b><i>Command Type</i></b>	0x12
<b><i>Return Status</i></b>	0x00 = Success See Appendix A for list of all non-successful return codes.
<b><i>Data Length</i></b>	0x0000
<b><i>Data</i></b>	There is no data returned.
<b><i>Checksum</i></b>	Xxxx

## 2.3 Set Mode and State

<b>BMC</b>	PowerVM: Must send on every transition state to active to set mode only (state is sent with no change value). Send on all customer mode changes. OPAL: Never sends
<b>(H)TMGT</b>	(H)TMGT will send this as part of booting the OCCs or when a state change request is made for manufacturing testing. BMC: Set state only (mode sent with no change value) FSP: Set mode and state

This command is used to set the OCC state and/or system power management mode.

- Command is only sent to the master OCC, the master OCC will then broadcast to all slaves
- The master OCC must NOT return a response to (H)TMGT until all OCCs have finished the state and/or mode changes required. This includes on a static mode change that the v/f must have finished being set for the new mode.
- Both the OCC state and system power management mode are sent in this command, the OCC will compare to determine which one (or both) is being changed.
- The OCC must support both the OCC state and system power management mode being changed with one “Set Mode and State” command.
- A failure to change state or mode by any OCC should result in a non-successful return code and an error log generated from the failing OCC to be collected by (H)TMGT. (H)TMGT will process the error log and determine what action should happen next i.e. reset OCCs or retry command. Any OCCs that had already successfully changed state/mode can either stay in the new state/mode or fall back to previous state/mode.

### Set Mode and State Command Packet:

<b>Sequence Number</b>	Xx
<b>Command Type</b>	0x20
<b>Data Length</b>	0x0006
<b>Data</b>	<b><u>Version 0x30</u></b> <b>Byte 1: Version</b> = 0x30 <b>Byte 2: OCC State</b> – Indicates the OCC state that the OCC should be in, if not OCC should change to this state. See Appendix B for valid OCC states. <b>Byte 3: System Power Management Mode</b> – Indicates the current system power management mode the OCC should be in, if not OCC should change to this mode. See Appendix C for valid system power management modes. <b>Bytes 4-5: Additional Mode Parameter</b> – Required when the System Power Management Mode in the previous byte is FFO or Static Frequency Point mode. This field is ignored by the OCC for all other modes. <b>FFO Mode – Frequency.</b> In MHz; MSB first. This is the frequency to

	<p>set for FFO and must be within the system minimum to Fmax frequency range. Command will be rejected if the frequency is not within the valid range. NOTE: WOF is disabled while in FFO, frequencies above WOF base are supported in FFO but not guaranteed that all chips will reach the desired frequency point.</p> <p><b>Static Frequency Point Mode – Point.</b> This is the point to pin the voltage and frequency to for Static Frequency Point mode. See enum defined in <a href="#">Frequency Points</a> chapter in this document for valid points</p> <p><b>Byte 6: Reserved = 0x00</b></p>
<b>Checksum</b>	Xxxx

**Set Mode and State Return Packet:**

<b><i>Sequence Number</i></b>	Xx
<b><i>Command Type</i></b>	0x20
<b><i>Return Status</i></b>	0x00 = Success See Appendix A for list of all non-successful return codes.
<b><i>Data Length</i></b>	0x0000
<b><i>Data</i></b>	There is no data returned.
<b><i>Checksum</i></b>	Xxxx

## 2.4 Configuration Data

This command is used to send configuration data that is needed by the OCC.

<b>BMC</b>	PowerVM only: Must send IPS config on every active state change
<b>(H)TMGT</b>	(H)TMGT will send this as part of booting the OCCs

### Configuration Data Command Packet:

<b>Sequence Number</b>	Xx																																		
<b>Command Type</b>	0x21																																		
<b>Data Length</b>	Dependent on Data Being Sent. See following sections for more details specific to each format.																																		
<b>Data</b>	<p>There are x bytes of data (not to exceed maximum):</p> <p><b>Byte 1:</b> Format – Indicates what format (i.e. Type of configuration data) the following command data is. See following sections for more details specific to each format. Some formats are only supported by the master OCC and the master OCC is responsible for broadcasting the information to all the slave OCCs:</p> <table><tr><th></th><th><b>Master OCC</b></th><th><b>Slave OCC</b></th></tr><tr><td><b>0x03: OCC Role</b></td><td colspan="2">Required for observation sent from (H)TMGT</td></tr><tr><td><b>0x04: APSS Cfg</b></td><td colspan="2">Required for observation sent from (H)TMGT</td></tr><tr><td><b>0x05: Memory Cfg</b></td><td colspan="2">Required for observation sent from (H)TMGT</td></tr><tr><td><b>0x07: Pcap</b></td><td>Required for active sent from (H)TMGT</td><td>Required for active from master OCC</td></tr><tr><td><b>0x0F: System Cfg</b></td><td colspan="2">Required for observation sent from (H)TMGT</td></tr><tr><td><b>0x11: IPS</b></td><td>Not required, optional for active state from TMGT/BMC</td><td>Not supported (IPS data is not used by slaves)</td></tr><tr><td><b>0x12: Mem Throttle</b></td><td colspan="2">Optional for active state sent from (H)TMGT. Special handling to determine if required based on memory config data packet.</td></tr><tr><td><b>0x13: Thermal Control Thresholds</b></td><td colspan="2">Required for observation sent from (H)TMGT</td></tr><tr><td><b>0x14: AVSBus Cfg</b></td><td colspan="2">Required for observation sent from (H)TMGT</td></tr><tr><td><b>0x15: GPU</b></td><td colspan="2">Not required. Sent from HTMGT</td></tr></table>			<b>Master OCC</b>	<b>Slave OCC</b>	<b>0x03: OCC Role</b>	Required for observation sent from (H)TMGT		<b>0x04: APSS Cfg</b>	Required for observation sent from (H)TMGT		<b>0x05: Memory Cfg</b>	Required for observation sent from (H)TMGT		<b>0x07: Pcap</b>	Required for active sent from (H)TMGT	Required for active from master OCC	<b>0x0F: System Cfg</b>	Required for observation sent from (H)TMGT		<b>0x11: IPS</b>	Not required, optional for active state from TMGT/BMC	Not supported (IPS data is not used by slaves)	<b>0x12: Mem Throttle</b>	Optional for active state sent from (H)TMGT. Special handling to determine if required based on memory config data packet.		<b>0x13: Thermal Control Thresholds</b>	Required for observation sent from (H)TMGT		<b>0x14: AVSBus Cfg</b>	Required for observation sent from (H)TMGT		<b>0x15: GPU</b>	Not required. Sent from HTMGT	
	<b>Master OCC</b>	<b>Slave OCC</b>																																	
<b>0x03: OCC Role</b>	Required for observation sent from (H)TMGT																																		
<b>0x04: APSS Cfg</b>	Required for observation sent from (H)TMGT																																		
<b>0x05: Memory Cfg</b>	Required for observation sent from (H)TMGT																																		
<b>0x07: Pcap</b>	Required for active sent from (H)TMGT	Required for active from master OCC																																	
<b>0x0F: System Cfg</b>	Required for observation sent from (H)TMGT																																		
<b>0x11: IPS</b>	Not required, optional for active state from TMGT/BMC	Not supported (IPS data is not used by slaves)																																	
<b>0x12: Mem Throttle</b>	Optional for active state sent from (H)TMGT. Special handling to determine if required based on memory config data packet.																																		
<b>0x13: Thermal Control Thresholds</b>	Required for observation sent from (H)TMGT																																		
<b>0x14: AVSBus Cfg</b>	Required for observation sent from (H)TMGT																																		
<b>0x15: GPU</b>	Not required. Sent from HTMGT																																		
<b>Checksum</b>	Xxxx																																		

### 2.4.1 Frequency Points (Format = 0x02) – **NOT SUPPORTED**

In P10 the OCC will be reading min/max frequency and operating points from the OCC Pstate Parameter Block (OPPB). Currently, there are no defined additional frequency points that need to be sent from (H)TMGT.

<b>Data Length</b>	0x0006 (version 0x30)
<b>Data</b>	<b>Byte 1:</b> Format = 0x02 <b>Byte 2:</b> Version = 0x30  <u><b>Version 0x30</b></u> <b>Currently not supported.</b>

## 2.4.2 Set OCC Role (Format = 0x03)

Tell the OCC if it should run as a master or slave.

- (H)TMGT knows which OCC is the master from the MRW. To be the master OCC requires a connection to the APSS.
- Until an OCC is told a role it should default to running as a slave
- Redundant APSS systems only. TMGT will determine when a master OCC failover is needed and is in charge of switching the OCC roles. Switching OCC roles is a failure scenario and requires the OCCs to be reset, after the reset the new master OCC will be told “Master OCC” role and sent all commands that are for master only.

<b>Data Length</b>	0x0004
<b>Data</b>	<b>Byte 1:</b> Format = 0x03 <b>Byte 2: OCC Role</b> <b>0x00</b> – Slave <b>0x01</b> – Master <b>0x02</b> – Backup Master. Redundant APSS systems only. OCC will act as a slave but will periodically do a health check on the redundant APSS. The OCC will not switch to the backup master on its own.  <b>All other values reserved.</b>  <b>Bytes 3-4:</b> Reserved = 0x0000



### 2.4.3 APSS Configuration (Format = 0x04)

Send APSS configuration data. This data comes from the MRW. “Function ID” value of 0 for ADC Channel Assignment or GPIO Pin Assignment indicates not assigned.

<b>Data Length</b>	Variable based on Type: Type 0x00 (APSS): 0x00F8 Type 0x02 (2 channel chip): 0x0020 Type 0xFF (none): 0x0004										
<b>Data</b>	<p><b>Byte 1:</b> Format = 0x04  <b>Byte 2:</b> Version = 0x20  <b>Byte 3: Type.</b> Indicates type of SPI attached analog-to-digital converter is present for power readings              <b>0x00 APSS</b> – 16 channel and GPIO config follows              <b>0x02 2 channel SPI attached chip (i.e. TI ADC122S021)</b> – 2 channels (ADC channel 0 and 1) will follow. No GPIOs.              <b>0xFF None</b> – No SPI attached chip. Power capping will not be supported by the OCC.  <b>Byte 4: Reserved = 0x00</b>          Format of one ADC channel info data set:</p> <table border="1"> <tr> <td><b>Data byte x</b></td><td><b>ADC Channel Assignment.</b> Enum. “Function ID” in MRW</td></tr> <tr> <td><b>Data byte x+1 thru x+4</b></td><td><b>Sensor ID.</b> Sensor ID for channel defined in MRW. Used by OCC for reporting channel power in poll response. 0 indicates no sensor ID.</td></tr> <tr> <td><b>Data byte x+5</b></td><td><b>Ground Select.</b></td></tr> <tr> <td><b>Data byte x+6 thru x+9</b></td><td><b>Gain.</b> Float.</td></tr> <tr> <td><b>Data byte x+10 thru x+13</b></td><td><b>Offset.</b> Float.</td></tr> </table> <p><b>Bytes 5-18:</b> ADC Channel 0 Info data set  <b>Bytes 19-32:</b> ADC Channel 1 Info data set  <b>Bytes 33-46:</b> ADC Channel 2 Info data set          :  <b>Bytes 201-214:</b> ADC Channel 14 Info data set  <b>Bytes 215-228:</b> ADC Channel 15 Info data set</p> <p><b>Byte 229:</b> GPIO Port 0 Mode in MRW unit-type= “gpio-global” id= GPIO_P0_MODE  <b>Byte 230:</b> GPIO Port 0 Reserved = 0x00  <b>Bytes 231-238:</b> GPIO Port 0 Pin[y] Assignment (enum) y=0-7 (8 pins)  <b>Byte 239:</b> GPIO Port 1 Mode in MRW unit-type= “gpio-global” id= GPIO_P1_MODE  <b>Byte 240:</b> GPIO Port 1 Reserved = 0x00  <b>Bytes 241-248:</b> GPIO Port 1 Pin[y] Assignment (enum) y=0-7 (8 pins)</p>	<b>Data byte x</b>	<b>ADC Channel Assignment.</b> Enum. “Function ID” in MRW	<b>Data byte x+1 thru x+4</b>	<b>Sensor ID.</b> Sensor ID for channel defined in MRW. Used by OCC for reporting channel power in poll response. 0 indicates no sensor ID.	<b>Data byte x+5</b>	<b>Ground Select.</b>	<b>Data byte x+6 thru x+9</b>	<b>Gain.</b> Float.	<b>Data byte x+10 thru x+13</b>	<b>Offset.</b> Float.
<b>Data byte x</b>	<b>ADC Channel Assignment.</b> Enum. “Function ID” in MRW										
<b>Data byte x+1 thru x+4</b>	<b>Sensor ID.</b> Sensor ID for channel defined in MRW. Used by OCC for reporting channel power in poll response. 0 indicates no sensor ID.										
<b>Data byte x+5</b>	<b>Ground Select.</b>										
<b>Data byte x+6 thru x+9</b>	<b>Gain.</b> Float.										
<b>Data byte x+10 thru x+13</b>	<b>Offset.</b> Float.										

#### 2.4.4 Memory Configuration (Format = 0x05)

Send present memory for memory associated with this OCC. OCC will require this packet for observation state but (H)TMGT may re-send in any state. If this is resent while in observation or active state, the OCC will only use it to enable if previously disabled or update the Sensor IDs if already enabled. If memory monitoring is already enabled and 0 data sets (disable memory) is sent the OCC will not disable in order to disable will require an OCC reset. Each OCC will only know about memory behind its processor, (H)TMGT must separate out memory to be sent to each specific OCC. There are two sensor IDs for each Memory Controller and DIMM, one to be used by the OCC for error call out and one for reporting memory temperatures. Memory power control is not supported with OPAL (no IPS) (H)TMGT will send 0xFF for memory power control settings to indicate no support.

NOTE: Info bytes 1 and 2 are used to give the OCC the mapping of what the 3 thermal sensors read from each memory buffer cache line is used for (defines what thermal control thresholds to use for the sensor). (H)TMGT gets this usage from attributes that are setup during IPL. For each Memory Controller:

<b>DIMM Info byte 1</b>	<b>DIMM Info byte 2</b> Translation IS REQUIRED from the value read from the attribute to the value to be sent to the OCC. OCC supported values defined below.
0x00	ATTR_MEM_EFF_THERM_SENSOR_0_USAGE
0x01	ATTR_MEM_EFF_THERM_SENSOR_1_USAGE
0xFF	ATTR_MEM_EFF_THERM_SENSOR_DIFF_USAGE




<b>Data Length</b>	Variable
<b>Data</b>	<p><b>Byte 1:</b> Format = 0x05 <b>Byte 2:</b> Version = 0x30 <b>Bytes 3-4:</b> Update time. Time in ms that the memory cache line is being updated. The OCC should not try to read memory faster than it is being updated. Comes from attribute ATTR_MSS_MRW_THERMAL_SENSOR_POLLING_PERIOD . Default to 1000ms <b>Byte 5:</b> Memory Power Control Default. ATTR_MSS_MRW_POWER_CONTROL_REQUESTED defines memory power control when system is NOT actively in idle power save: 0x00 = Off 0x01 = Power Down 0x02 = Power Down and Self Time Refresh (STR) 0x03 = Power Down and STR Clock Stop 0xFF = Not supported <b>Byte 6:</b> Idle Power Memory Power Control. ATTR_MSS_MRW_IDLE_POWER_CONTROL_REQUESTED defines memory power control when system is actively in idle power save: 0x00 = Off 0x01 = Power Down 0x02 = Power Down and Self Time Refresh (STR)</p>

<b>Data Length</b>	Variable												
	<p>0x03 = Power Down and STR Clock Stop 0xFF = Not supported</p> <p><b>Byte 7:</b> Number of data sets to follow. NOTE: A 0x00 indicates that memory monitoring is to be disabled (if not already enabled). When disabled there is no memory monitoring and the OCC will not require the Memory Throttling data packet.</p> <p><b>Version 0x30</b> The format of each set is:</p> <table> <tr> <td><b>Data bytes x thru x+3</b></td><td><b>Hardware Sensor ID.</b> Sensor ID to use for calling out over temperature error due to this sensor</td></tr> <tr> <td><b>Data bytes x+4 thru x+7</b></td><td><b>Temperature Sensor ID.</b> Sensor ID to use for reporting this sensor's temperature in the poll response. If not defined send Hardware sensor ID.</td></tr> <tr> <td><b>Data byte x+8</b></td><td> <b>Memory Type.</b> This will indicate what the remaining data represents. <ul style="list-style-type: none"> <li>• <b>0xAy :</b> Where y = Memory Buffer # 0-F that indicates physical location of Memory Buffer.</li> </ul> </td></tr> <tr> <td><b>Data byte x+9</b></td><td> <b>DIMM Info Byte 1.</b> <ul style="list-style-type: none"> <li>• <b>Memory DTS # 0 or 1 :</b> indicates which external DIMM thermal sensor field as OCC reads from the sensor cache this data set is for. 0xFF indicates that the HW and Temperature Sensor IDs are for the "on chip" thermal sensor field in the sensor cache. NOTE: the next byte indicates what the actual type is for the thermal sensor to know what thermal control thresholds to use for this thermal sensor.</li> </ul> </td></tr> <tr> <td><b>Data byte x+10</b></td><td> <b>DIMM Info Byte 2.</b> <ul style="list-style-type: none"> <li>• <b>Temperature type</b> <ul style="list-style-type: none"> <li>○ 0xFF: Not used (OCC will ignore sensor reading)</li> <li>○ 0x01: Internal memory controller</li> <li>○ 0x02: DIMM</li> <li>○ 0x03: Memory Controller + DIMM</li> <li>○ 0x07: PMIC</li> <li>○ 0x08: External Memory controller</li> </ul> </li> </ul> </td></tr> <tr> <td><b>Data byte x+11</b></td><td> <b>DIMM Info Byte 3.</b> <ul style="list-style-type: none"> <li>• Reserved. 0x00</li> </ul> </td></tr> </table>	<b>Data bytes x thru x+3</b>	<b>Hardware Sensor ID.</b> Sensor ID to use for calling out over temperature error due to this sensor	<b>Data bytes x+4 thru x+7</b>	<b>Temperature Sensor ID.</b> Sensor ID to use for reporting this sensor's temperature in the poll response. If not defined send Hardware sensor ID.	<b>Data byte x+8</b>	<b>Memory Type.</b> This will indicate what the remaining data represents. <ul style="list-style-type: none"> <li>• <b>0xAy :</b> Where y = Memory Buffer # 0-F that indicates physical location of Memory Buffer.</li> </ul>	<b>Data byte x+9</b>	<b>DIMM Info Byte 1.</b> <ul style="list-style-type: none"> <li>• <b>Memory DTS # 0 or 1 :</b> indicates which external DIMM thermal sensor field as OCC reads from the sensor cache this data set is for. 0xFF indicates that the HW and Temperature Sensor IDs are for the "on chip" thermal sensor field in the sensor cache. NOTE: the next byte indicates what the actual type is for the thermal sensor to know what thermal control thresholds to use for this thermal sensor.</li> </ul>	<b>Data byte x+10</b>	<b>DIMM Info Byte 2.</b> <ul style="list-style-type: none"> <li>• <b>Temperature type</b> <ul style="list-style-type: none"> <li>○ 0xFF: Not used (OCC will ignore sensor reading)</li> <li>○ 0x01: Internal memory controller</li> <li>○ 0x02: DIMM</li> <li>○ 0x03: Memory Controller + DIMM</li> <li>○ 0x07: PMIC</li> <li>○ 0x08: External Memory controller</li> </ul> </li> </ul>	<b>Data byte x+11</b>	<b>DIMM Info Byte 3.</b> <ul style="list-style-type: none"> <li>• Reserved. 0x00</li> </ul>
<b>Data bytes x thru x+3</b>	<b>Hardware Sensor ID.</b> Sensor ID to use for calling out over temperature error due to this sensor												
<b>Data bytes x+4 thru x+7</b>	<b>Temperature Sensor ID.</b> Sensor ID to use for reporting this sensor's temperature in the poll response. If not defined send Hardware sensor ID.												
<b>Data byte x+8</b>	<b>Memory Type.</b> This will indicate what the remaining data represents. <ul style="list-style-type: none"> <li>• <b>0xAy :</b> Where y = Memory Buffer # 0-F that indicates physical location of Memory Buffer.</li> </ul>												
<b>Data byte x+9</b>	<b>DIMM Info Byte 1.</b> <ul style="list-style-type: none"> <li>• <b>Memory DTS # 0 or 1 :</b> indicates which external DIMM thermal sensor field as OCC reads from the sensor cache this data set is for. 0xFF indicates that the HW and Temperature Sensor IDs are for the "on chip" thermal sensor field in the sensor cache. NOTE: the next byte indicates what the actual type is for the thermal sensor to know what thermal control thresholds to use for this thermal sensor.</li> </ul>												
<b>Data byte x+10</b>	<b>DIMM Info Byte 2.</b> <ul style="list-style-type: none"> <li>• <b>Temperature type</b> <ul style="list-style-type: none"> <li>○ 0xFF: Not used (OCC will ignore sensor reading)</li> <li>○ 0x01: Internal memory controller</li> <li>○ 0x02: DIMM</li> <li>○ 0x03: Memory Controller + DIMM</li> <li>○ 0x07: PMIC</li> <li>○ 0x08: External Memory controller</li> </ul> </li> </ul>												
<b>Data byte x+11</b>	<b>DIMM Info Byte 3.</b> <ul style="list-style-type: none"> <li>• Reserved. 0x00</li> </ul>												

## 2.4.5 Power Cap Values Data Packet (Format = 0x07)

<b>Data Length</b>	0x000A
<b>Data</b>	<p><b>Byte 1:</b> Format = 0x07</p> <p><b>Version 0x20:</b></p> <p><b>Byte 2:</b> Version = 0x20</p> <p><b>Bytes 3 &amp; 4: Minimum soft power cap.</b> In 1W units (MSB sent first), not supported on all systems, if soft power capping is not supported (H)TMGT will send the minimum hard power cap for soft. This is the lowest power cap ever allowed to be set. A power cap set between the minimum soft power cap and the minimum hard power cap is called a soft power cap. Setting a soft power cap is not guaranteed to be maintained at all times.</p> <p><b>Bytes 5 &amp; 6: Minimum hard power cap.</b> In 1W units (MSB sent first), the lowest power cap that a user may set and is guaranteed to be held via processor DVFS.</p> <p><b>Bytes 7 &amp; 8: System Maximum power cap.</b> In 1W units (MSB sent first). This is a permanent power cap that is required by the system. A user can not set a power cap higher than this value. OCC will actuate to this power cap or, if set, the user power cap set with the “Set User Power Cap” command. The current power cap value that the OCC is actuating to will be sent in the sensor poll response.</p> <p><b>Bytes 9 &amp; 10: Quick Power Drop power cap.</b> In 1W units (MSB sent first). If there is no support for QPD this will be 0x0000 and the OCC will not be monitoring for QPD. This is also known as Oversubscription.</p>

### NOTES:

-  Power cap data is only supported by the master OCC. The master OCC will then broadcast this data to all slave OCCs. This is to ensure all OCCs have the same power cap data within a tick.
-  All power cap values are DC (output) power
-  On multi-node systems TMGT will send the power cap data to the master OCC in each node and the data must be node level power cap data. TMGT is responsible for taking a system power cap and breaking it down to node level data prior to sending to the OCC.

## 2.4.6 System Configuration (Format = 0x0F)

This packet gives additional information and sensor IDs for the system.

Data Length	0xA3																
Data	Byte 1: Format = 0x0F																
	Byte 2: Version = 0x30																
	Byte 3: General System Type (bit defined)																
	<table><tr><td>Bit 0 (msb)</td><td>Bit 1</td><td>Bit 2</td><td>Bit 3</td><td>Bit 4</td><td>Bit 5</td><td>Bit 6</td><td>Bit 7 (lsb)</td></tr><tr><td>KVM</td><td></td><td></td><td></td><td></td><td>WOF Reset Limit</td><td></td><td>Single Node</td></tr></table>	Bit 0 (msb)	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7 (lsb)	KVM					WOF Reset Limit		Single Node
	Bit 0 (msb)	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7 (lsb)									
	KVM					WOF Reset Limit		Single Node									
	KVM – ‘1’ = System boot mode is OPAL. OCC does not set pStates directly, all power and thermal management is done by clipping Pmax register instead and the OCC-OPAL shared memory interface is updated.																
	‘0’ = PowerVM. OCC is in full control of frequency and power management modes are supported.																
	WOF Reset Limit – Indicates that max number of resets due to WOF has been reached and WOF should be disabled.																
	‘1’ = WOF reset limit reached, disable WOF																
‘0’ = WOF reset limit not reached, WOF may be enabled.																	
Single Node – Indicates if system is single or multi-node.																	
‘1’ = system is a single node																	
‘0’ = multi-node (NOTE: This doesn’t necessarily mean more than 1 node is present, this just means that more than 1 node may be present)																	
Bytes 4-7: Processor Sensor ID – Sensor ID for this OCC processor, used by OCC for processor error call out																	
Bytes 8-11: Processor Frequency Sensor ID – Sensor ID for this OCC processor frequency, used by OCC to report processor frequency																	
Bytes 12-15: Core 0 Temperature Sensor ID – Sensor ID for physical core 0, used by OCC to report core 0 temperature																	
Bytes 16-19: Core 1 Temperature Sensor ID – Sensor ID for physical core 1, used by OCC to report core 1 temperature																	
Bytes 20-23: Core 2 Temperature Sensor ID – Sensor ID for physical core 2, used by OCC to report core 2 temperature																	

**Bytes 24-27:** Core 3 Temperature Sensor ID – Sensor ID for physical core 3, used by OCC to report core 3 temperature

**Bytes 28-31:** Core 4 Temperature Sensor ID – Sensor ID for physical core 4, used by OCC to report core 4 temperature

**Bytes 32-35:** Core 5 Temperature Sensor ID – Sensor ID for physical core 5, used by OCC to report core 5 temperature

**Bytes 36-39:** Core 6 Temperature Sensor ID – Sensor ID for physical core 6, used by OCC to report core 6 temperature

**Bytes 40-43:** Core 7 Temperature Sensor ID – Sensor ID for physical core 7, used by OCC to report core 7 temperature

**Bytes 44-47:** Core 8 Temperature Sensor ID – Sensor ID for physical core 8, used by OCC to report core 8 temperature

**Bytes 48-51:** Core 9 Temperature Sensor ID – Sensor ID for physical core 9, used by OCC to report core 9 temperature

**Bytes 52-55:** Core 10 Temperature Sensor ID – Sensor ID for physical core 10, used by OCC to report core 10 temperature

**Bytes 56-59:** Core 11 Temperature Sensor ID – Sensor ID for physical core 11, used by OCC to report core 11 temperature

**Bytes 60-63:** Core 12 Temperature Sensor ID – Sensor ID for physical core 12, used by OCC to report core 12 temperature

**Bytes 64-67:** Core 13 Temperature Sensor ID – Sensor ID for physical core 13, used by OCC to report core 13 temperature

**Bytes 68-71:** Core 14 Temperature Sensor ID – Sensor ID for physical core 14, used by OCC to report core 14 temperature

**Bytes 72-75:** Core 15 Temperature Sensor ID – Sensor ID for physical core 15, used by OCC to report core 15 temperature

**Bytes 76-79:** Core 16 Temperature Sensor ID – Sensor ID for physical core 16, used by OCC to report core 16 temperature

**Bytes 80-83:** Core 17 Temperature Sensor ID – Sensor ID for physical core 17, used by OCC to report core 17 temperature

**Bytes 84-87:** Core 18 Temperature Sensor ID – Sensor ID for physical core 18, used by OCC to report core 18 temperature

**Bytes 88-91:** Core 19 Temperature Sensor ID – Sensor ID for physical core 19, used by OCC to report core 19 temperature

**Bytes 92-95:** Core 20 Temperature Sensor ID – Sensor ID for physical core 20, used by OCC to report core 20 temperature

**Bytes 96-99:** Core 21 Temperature Sensor ID – Sensor ID for physical core 21, used by OCC to report core 21 temperature

**Bytes 100-103:** Core 22 Temperature Sensor ID – Sensor ID for physical core 22, used by OCC to report core 22 temperature

**Bytes 104-107:** Core 23 Temperature Sensor ID – Sensor ID for physical core 23, used by OCC to report core 23 temperature

**Bytes 108-111:** Core 24 Temperature Sensor ID – Sensor ID for physical core 24, used by OCC to report core 24 temperature

**Bytes 112-115:** Core 25 Temperature Sensor ID – Sensor ID for physical core 25, used by OCC to report core 25 temperature

**Bytes 116-119:** Core 26 Temperature Sensor ID – Sensor ID for physical core 26, used by OCC to report core 26 temperature  
**Bytes 120-123:** Core 27 Temperature Sensor ID – Sensor ID for physical core 27, used by OCC to report core 27 temperature  
**Bytes 124-127:** Core 28 Temperature Sensor ID – Sensor ID for physical core 28, used by OCC to report core 28 temperature  
**Bytes 128-131:** Core 29 Temperature Sensor ID – Sensor ID for physical core 29, used by OCC to report core 29 temperature  
**Bytes 132-135:** Core 30 Temperature Sensor ID – Sensor ID for physical core 30, used by OCC to report core 30 temperature  
**Bytes 136-139:** Core 31 Temperature Sensor ID – Sensor ID for physical core 31, used by OCC to report core 31 temperature  
**Bytes 140-143:** Backplane Sensor ID – Used by OCC for system backplane error call out  
**Bytes 144-147:** APSS/SPI attached ADC Sensor ID – Used by OCC for APSS/SPI attached ADC error call out  
**Bytes 148-151:** VRM Vdd Sensor ID – Sensor ID for VRM Vdd, used by OCC for VRM Vdd error call out  
**Bytes 152-155:** VRM Vdd Temperature Sensor ID – Sensor ID for VRM Vdd, used by OCC to report VRM Vdd temperature  
**Bytes 156-163:** Reserved

### 2.4.7 Idle Power Saver Settings (Format = 0x11)

This packet sends the state of idle power saver and the parameters for idle power saver. If a system doesn't support Idle Power Saver TMGT/BMC will send all 0's (Idle Power Saver disabled).

Data Length	9
Data	<b>Byte 1:</b> Format = 0x11 <b>Byte 2:</b> Version  <b><u>Version 0x00:</u></b> <b>Byte 2:</b> Version = 0x00 <b>Byte 3:</b> Idle Power Saver Enable: 0 = disabled; 1 = enabled <b>Bytes 4-5:</b> Delay Time (in seconds) to enter Idle Power <b>Byte 6:</b> Utilization threshold percentage to enter Idle Power <b>Bytes 7-8:</b> Delay Time (in seconds) to exit Idle Power <b>Byte 9:</b> Utilization threshold percentage to exit Idle Power



## 2.4.8 Memory Throttling (Format = 0x12)

This packet sends the throttle settings that are calculated by hardware procedures based on power allocated for memory and memory utilization defined in the MRW. This packet is required for active state only if memory configuration packet indicated there is memory monitoring support.

### **NOTES:**

- Can only have one denominator (time based) which is set by Host Boot and will not be sent to OCC since this will not be changed.
- OCC will reject this data packet if any N value is 0.
- (H)TMGT must ensure that 0 is never sent for any numerator values. If there is a failure to calculate throttle values the safe mode memory throttle defined in the MRW will be sent to the OCC.
- This spec will only refer to N\_PER\_MBA and N\_PER\_CHIP in chip specs:
  - “N\_PER\_SLOT” is the same as “N\_PER\_MBA”
  - “N\_PER\_PORT” is the same as “N\_PER\_CHIP”
- When OCC switches between modes and oversubscription the OCC must write the appropriate N\_PER\_MBA and N\_PER\_CHIP.
- When throttling due to OT the OCC will only change N\_PER\_MBA, the N\_PER\_CHIP will remain unchanged.
- Any error reading/writing memory throttle will call out the processor
- Thermal reason to change throttles will be calling out the component that was OT

Data Length	Variable								
Data	<p><b>Byte 1:</b> Format = 0x12 <b>Byte 2:</b> Version = 0x40 <b>Byte 3:</b> Number of memory throttling data sets to follow.</p> <p><b><u>Version 0x40:</u></b> The format of each set is:</p> <table><tr><td><b>Data byte x</b></td><td><b>Throttle Info Byte 1:</b> Membuf #. Value 0-15 that indicates physical Membuf that the throttles are for</td></tr><tr><td><b>Data byte x+1</b></td><td><b>Throttle Info Byte 2:</b> Reserved = 0x00</td></tr><tr><td><b>Data bytes x+2 &amp; x+3</b></td><td><b>Minimum N_PER_MBA.</b> Lowest per MBA numerator ever allowed when OCC is throttling memory due to OT. This is calculated based on the MRW/def file minimum memory utilization.</td></tr><tr><td><b>Data bytes x+4 &amp; x+5</b></td><td><b>Modes disabled N_PER_MBA.</b> Static per MBA numerator setting when all power management modes are disabled. BMC systems this is calculated using xml N_PLUS_ONE_MAX_MEM_POWER_WATTS. FSP only: Calculated based on power available @Disabled frequency</td></tr></table>	<b>Data byte x</b>	<b>Throttle Info Byte 1:</b> Membuf #. Value 0-15 that indicates physical Membuf that the throttles are for	<b>Data byte x+1</b>	<b>Throttle Info Byte 2:</b> Reserved = 0x00	<b>Data bytes x+2 &amp; x+3</b>	<b>Minimum N_PER_MBA.</b> Lowest per MBA numerator ever allowed when OCC is throttling memory due to OT. This is calculated based on the MRW/def file minimum memory utilization.	<b>Data bytes x+4 &amp; x+5</b>	<b>Modes disabled N_PER_MBA.</b> Static per MBA numerator setting when all power management modes are disabled. BMC systems this is calculated using xml N_PLUS_ONE_MAX_MEM_POWER_WATTS. FSP only: Calculated based on power available @Disabled frequency
<b>Data byte x</b>	<b>Throttle Info Byte 1:</b> Membuf #. Value 0-15 that indicates physical Membuf that the throttles are for								
<b>Data byte x+1</b>	<b>Throttle Info Byte 2:</b> Reserved = 0x00								
<b>Data bytes x+2 &amp; x+3</b>	<b>Minimum N_PER_MBA.</b> Lowest per MBA numerator ever allowed when OCC is throttling memory due to OT. This is calculated based on the MRW/def file minimum memory utilization.								
<b>Data bytes x+4 &amp; x+5</b>	<b>Modes disabled N_PER_MBA.</b> Static per MBA numerator setting when all power management modes are disabled. BMC systems this is calculated using xml N_PLUS_ONE_MAX_MEM_POWER_WATTS. FSP only: Calculated based on power available @Disabled frequency								

	with redundant power. (H)TMGT to guarantee that this is not lower than MRW/def file minimum memory utilization for redundant power
<b>Data bytes x+6 &amp; x+7</b>	<b>Modes disabled N_PER_CHIP.</b> Static per chip numerator setting when all power management modes are disabled, and system is NOT in oversubscription.
<b>Data bytes x+8 &amp; x+9</b>	<b>Ultra Turbo N_PER_MBA.</b> Static per MBA numerator setting for ultra turbo modes (i.e. max performance, dynamic performance) BMC systems this will be the same as Modes disabled. FSP only: Calculated based on power available @UT frequency. If power does not allow for this to meet MRW/def file minimum UT memory utilization (H)TMGT will send the minimum utilization value and log an unrecoverable error.
<b>Data bytes x+10 &amp; x+11</b>	<b>Ultra Turbo N_PER_CHIP.</b> Static per chip numerator setting for ultra turbo.
<b>Data bytes x+12 &amp; x+13</b>	<b>Fmax N_PER_MBA.</b> Static per MBA numerator setting when in Fmax mode. BMC systems this will be the same as Modes disabled. FSP only: Calculated based on power available @Fmax frequency. If power does not allow for this to meet MRW/def file minimum Fmax memory utilization (H)TMGT will send the minimum utilization value and log an informational error. TBD Disallow Fmax mode?.
<b>Data bytes x+14 &amp; x+15</b>	<b>Fmax N_PER_CHIP.</b> Static per chip numerator setting when in Fmax mode.
<b>Data bytes x+16 &amp; x+17</b>	<b>Oversubscription N_PER_MBA.</b> Static per MBA numerator setting when in oversubscription. (H)TMGT to guarantee that this is not lower than MRW/def file minimum utilization for oversubscription
<b>Data bytes x+18 &amp; x+19</b>	<b>Oversubscription N_PER_CHIP.</b> Static per chip numerator setting when in oversubscription.
<b>Data bytes x+20 &amp; x+21</b>	<b>Reserved</b>

## 2.4.9 Thermal Control Thresholds (Format = 0x13)

This command is used to send the temperature thresholds. All temperatures are in Celsius.

Data Length	Variable		
Data	<p><b>Byte 1:</b> Format = 0x13  <b>Byte 2:</b> Version= 0x30</p> <p><b>Version 0x30:</b>  <b>Byte 3: Processor Core Weight.</b> In 1/10ths. Weight factor for the 2 core DTS to calculate a core temperature. 0 = core DTS not used to calculate core temperature (and eventual processor temperature)  <b>Byte 4: Processor Quad Weight.</b> In 1/10ths. Weight factor for the 1 quad (racetrack) DTS to calculate a core temperature. 0 = quad DTS not used to calculate core temperature (and eventual processor temperature)  <b>Byte 5: Processor L3 Weight.</b> In 1/10ths. Weight factor for the 1 L3 DTS to calculate a core temperature. 0 = L3 DTS not used to calculate core temperature (and eventual processor temperature)  <b>Byte 6:</b> Number of data sets that follows. Format of each data set is:</p> <table> <tr> <td><b>Data byte x</b></td><td> <p><b>FRU type.</b> Indicates FRU type that thermal info is for  0x00: Processor (hottest core temperature)  0x01: Internal Memory controller sensor  0x02: DIMM (on board sensor for DRAM)  0x03: Memory Controller+DIMM <u>NOTE: For 4U DDIMMs this is used as "DIMM" type</u>  0x04: GPU core  0x05: GPU memory  0x06: VRM Vdd  0x07: PMIC (on board sensor for PMIC)  0x08: External Memory controller sensor <u>NOTE: For 4U DDIMMs this is used as "PMIC" type</u>  0x09: Processor I/O Ring</p> <p><b>Special FRU Types for DVFS/ERROR Deltas</b>  DVFS and ERROR fields are signed deltas that the OCC will apply to calculate the ERROR and DVFS temperatures. NOTE: If using deltas for a specific FRU type then that FRU type defined above is either not sent or must be sent before so the deltas will overwrite the temperatures if there is a valid VPD temperature to apply deltas to if there is no valid VPD temperature then OCC will use the values sent for the FRU type if sent and not result in an error.</p> <p><b>0xF0: Processor Deltas</b>  DVFS and ERROR fields are signed deltas from</p> </td></tr> </table>	<b>Data byte x</b>	<p><b>FRU type.</b> Indicates FRU type that thermal info is for  0x00: Processor (hottest core temperature)  0x01: Internal Memory controller sensor  0x02: DIMM (on board sensor for DRAM)  0x03: Memory Controller+DIMM <u>NOTE: For 4U DDIMMs this is used as "DIMM" type</u>  0x04: GPU core  0x05: GPU memory  0x06: VRM Vdd  0x07: PMIC (on board sensor for PMIC)  0x08: External Memory controller sensor <u>NOTE: For 4U DDIMMs this is used as "PMIC" type</u>  0x09: Processor I/O Ring</p> <p><b>Special FRU Types for DVFS/ERROR Deltas</b>  DVFS and ERROR fields are signed deltas that the OCC will apply to calculate the ERROR and DVFS temperatures. NOTE: If using deltas for a specific FRU type then that FRU type defined above is either not sent or must be sent before so the deltas will overwrite the temperatures if there is a valid VPD temperature to apply deltas to if there is no valid VPD temperature then OCC will use the values sent for the FRU type if sent and not result in an error.</p> <p><b>0xF0: Processor Deltas</b>  DVFS and ERROR fields are signed deltas from</p>
<b>Data byte x</b>	<p><b>FRU type.</b> Indicates FRU type that thermal info is for  0x00: Processor (hottest core temperature)  0x01: Internal Memory controller sensor  0x02: DIMM (on board sensor for DRAM)  0x03: Memory Controller+DIMM <u>NOTE: For 4U DDIMMs this is used as "DIMM" type</u>  0x04: GPU core  0x05: GPU memory  0x06: VRM Vdd  0x07: PMIC (on board sensor for PMIC)  0x08: External Memory controller sensor <u>NOTE: For 4U DDIMMs this is used as "PMIC" type</u>  0x09: Processor I/O Ring</p> <p><b>Special FRU Types for DVFS/ERROR Deltas</b>  DVFS and ERROR fields are signed deltas that the OCC will apply to calculate the ERROR and DVFS temperatures. NOTE: If using deltas for a specific FRU type then that FRU type defined above is either not sent or must be sent before so the deltas will overwrite the temperatures if there is a valid VPD temperature to apply deltas to if there is no valid VPD temperature then OCC will use the values sent for the FRU type if sent and not result in an error.</p> <p><b>0xF0: Processor Deltas</b>  DVFS and ERROR fields are signed deltas from</p>		

	<p>#V TDP sort temperature sent to OCC via OPPB.</p> <p><b>0xF9: Processor I/O Ring Deltas</b></p> <p>DVFS and ERROR fields are signed deltas from #V I/O throttle temperature sent to OCC via OPPB</p>
<b>Data byte x+1</b>	<p><b>DVFS</b> - Temperature above which DVFS/throttling will be invoked. 0xFF indicates not defined i.e. no throttle/DVFS action for this FRU type.</p> <p><b>For FRU TYPES 0xFy:</b> this is a signed delta (-128 to 127) that the OCC will apply to the VPD value to determine DVFS temperature.</p>
<b>Data byte x+2</b>	<p><b>ERROR</b> - Temperature to generate error and callout FRU over temperature. 0xFF indicates not defined and no error is logged due to OT.</p> <p><b>For FRU TYPES 0xFy:</b> this is a signed delta (-128 to 127) that the OCC will apply to the VPD value to determine ERROR temperature.</p>
<b>Data byte x+3</b>	<p><b>MAX_READ_TIMEOUT</b> – Maximum time (in seconds) allowed without having new temperature readings. Throttling/dvfs will occur if this timeout is hit. 0xFF indicates not defined and will never timeout.</p>
<b>Data bytes x+4 &amp; x+5</b>	Reserved.
<p><b>Bytes 7-12:</b> Data set #1</p> <p><b>Bytes 13-18:</b> Data set #2</p> <p>:</p>	

## 2.4.10 AVSBus Configuration (Format = 0x14)

This command is used to send the AVSbus configuration. All values come from the xml file.  
OCC Usage: OCC will only be reading VDD temperature on the AVSBus via OCI to SPIPMBus (O2S) Bridge B. The PGPE reads currents and voltages, the OCC will use the PGPE readings. NOTE: bridge A is PGPE owned and must not be used by the OCC.

Data Length	0x08
Data	<b>Byte 1:</b> Format = 0x14 <b>Byte 2:</b> Version= 0x30  <b><u>Version 0x30:</u></b> <b>Byte 3: Vdd Bus Num.</b> Defines the AVSBus (0 – 2) which has the core VDD rail VRM. 0xFF=Not defined, Vdd not monitored. Comes from ATTR_AVSBUS_BUSNUM[0] <b>Byte 4: Vdd Rail Select.</b> Defines the AVSBus rail selector (0 – 15) for the VDD VRM on Vdd Bus Num. Comes from ATTR_AVSBUS_RAIL[0] <b>Bytes 5-8: Reserved.</b>

## 2.4.11 GPU (Format = 0x15)

This command is used to send information needed by the OCC for GPU handling. The OCC will determine which GPUs are present from the APSS GPIOs. No GPU support on FSP systems, TMGT will send all 0's for power and sensor IDs.

<b>Data Length</b>	Version dependent. 0x002C (version 1) 0x000A minimum (version 2)
<b>Data</b>	<p><b>Byte 1:</b> Format = 0x15</p> <p><b>Byte 2:</b> Version = 0x01 or 0x02</p> <p><b>Version = 1</b></p> <p><b>Bytes 3-4:</b> Total non-GPU maximum power in watts. Maximum system power excluding GPUs when CPUs are at maximum frequency (ultra turbo) and memory at maximum power (least throttled) plus everything else (fans...) excluding GPUs. HTMGT calculates max CPU and memory power and adds xml ATTR_MISC_SYSTEM_COMPONENTS_MAX_POWER_WATTS to calculate this total.</p> <p><b>Bytes 5-6:</b> Total Processor/Memory Power Drop in watts. Amount the total non-GPU maximum power can be reduced by. HTMGT calculates this as the CPU power at minimum frequency plus memory at minimum power (most throttled)</p> <p><b>Bytes 7-8:</b> Reserved = 0x0000</p> <p><b>Bytes 9-12:</b> GPU 0 core Temperature Sensor ID – Sensor ID for GPU 0 core, used by OCC to report GPU 0 core temperature. 0 if not defined in xml.</p> <p><b>Bytes 13-16:</b> GPU 0 HBM Temperature Sensor ID – Sensor ID for GPU 0 HBM, used by OCC to report GPU 0 memory temperature. 0 if not defined in xml.</p> <p><b>Bytes 17-20:</b> GPU 0 Sensor ID – Sensor ID for physical GPU 0, used by OCC for GPU 0 error callout. 0 if not defined in xml.</p> <p><b>Bytes 21-24:</b> GPU 1 core Temperature Sensor ID – Sensor ID for GPU 1 core, used by OCC to report GPU 1 core temperature. 0 if not defined in xml.</p> <p><b>Bytes 25-28:</b> GPU 1 HBM Temperature Sensor ID – Sensor ID for GPU 1 HBM, used by OCC to report GPU 1 memory temperature. 0 if not defined in xml.</p> <p><b>Bytes 29-32:</b> GPU 1 Sensor ID – Sensor ID for physical GPU 1, used by OCC for GPU 1 error callout. 0 if not defined in xml.</p> <p><b>Bytes 33-36:</b> GPU 2 core Temperature Sensor ID – Sensor ID for GPU 2 core, used by OCC to report GPU 2 core temperature. 0 if not defined in xml.</p> <p><b>Bytes 37-40:</b> GPU 2 HBM Temperature Sensor ID – Sensor ID for GPU 2 HBM, used by OCC to report GPU 2 memory temperature. 0 if not defined in xml.</p> <p><b>Bytes 41-44:</b> GPU 2 Sensor ID – Sensor ID for physical GPU 2, used by OCC for GPU 2 error callout. 0 if not defined in xml.</p> <p><b>Version = 2</b></p> <p><b>Bytes 3-4:</b> Total non-GPU maximum power in watts. Maximum system power excluding GPUs when CPUs are at maximum frequency (ultra turbo) and memory at maximum power (least throttled) plus everything else (fans...)</p>

excluding GPUs. HTMGT calculates max CPU and memory power and adds xml ATTR\_MISC\_SYSTEM\_COMPONENTS\_MAX\_POWER\_WATTS to calculate this total.

**Bytes 5-6: Total Processor/Memory Power Drop in watts.** Amount the total non-GPU maximum power can be reduced by. HTMGT calculates this as the CPU power at minimum frequency plus memory at minimum power (most throttled)

**Byte 7: Total number of GPUs that may be present in the system** (across all processors). On systems with an APSS the OCC will determine GPU presence from the APSS GPIOs, and this is the number of GPUs possible but not necessarily present. On systems without an APSS, HB is determining GPU presence, and this is the number of GPUs that are actually present in the system. OCC will only use this number if system has a non-APSS SPI attached chip for power capping without an APSS.

**Byte 8: PIB I2C Master Engine** for the GPUs (0x01 = "C") Any other engine requires additional OCC changes.

**Byte 9: GPU I2C Bus Voltage** in tenths. Supported values:

0x00 = Leave at default, OCC will not set

0x12 = 1.8V OCC will set

**Byte 10: Number of data sets.** This is the number of GPUs that may be present behind this processor that this OCC may monitor. On systems with an APSS the OCC will determine GPU presence from the APSS GPIOs, and this is the number of GPUs possible for this OCC but not necessarily present. On systems without an APSS, HB is determining GPU presence, and this is the number of GPUs that are actually present.

**The format of each set is:**

<b>Data byte x</b>	<b>GPU ID.</b> Number (starting from 0 for first GPU) to indicate GPU that this data set is for
<b>Data byte x+1</b>	<b>GPU I2C Port.</b> 0xFF if not defined in xml and GPU will not be monitored.
<b>Data byte x+2</b>	<b>GPU I2C Slave Address.</b> 0xFF if not defined in xml and GPU will not be monitored.
<b>Data byte x+3</b>	<b>Reserved = 0x00</b>
<b>Data bytes x+4 thru x+7</b>	<b>GPU Core Temperature Sensor ID.</b> Sensor ID for reporting GPU core temperature in poll response. 0 if not defined in xml, temperature will still be read but not used for fan control.
<b>Data bytes x+8 thru x+11</b>	<b>GPU HBM Temperature Sensor ID.</b> Sensor ID for reporting GPU memory temperature in poll response. 0 if not defined in xml, temperature will still be read but not used for fan control.
<b>Data bytes x+12 thru x+15</b>	<b>GPU Sensor ID.</b> Sensor ID for physical GPU. Used by OCC for GPU error callout. 0 if not defined in xml.

## 2.4.12 Setup Configuration Data Return Packet

<b>Sequence Number</b>	Xx
<b>Command Type</b>	0x21
<b>Return Status</b>	0x00 = Success See Appendix A for list of all non-successful return codes.
<b>Data Length</b>	0x0000
<b>Data</b>	There is no data returned.
<b>Checksum</b>	Xxxx



---

## 2.5 Set User Power Cap

This command is used to set a user specified power cap.

<b>BMC</b>	Will send to master OCC only when “OCC Active” sensor is TRUE and there is a change to the user power cap. NOTE: If user is setting the power limit as input power, the BMC must do conversion to output power using the power supply efficiency factor from the Configuration file.
<b>(H)TMGT</b>	Will send as part of the OCC boot process to ensure OCC has current user power cap prior to going active

### Set User Power Cap Command Packet:

<b>Sequence Number</b>	xx
<b>Command Type</b>	0x22
<b>Data Length</b>	0x0002
<b>Data</b>	<b>Bytes 1-2: Activate Power Cap</b> – Output Power cap to activate in 1W units (MSB first). 0x0000 = Disable user power cap (user power cap is not active)
<b>Checksum</b>	Xxxx

**Set User Power Cap Return Packet:**

<b>Sequence Number</b>	Xx
<b>Command Type</b>	0x22
<b>Return Status</b>	0x00 = Success See Appendix A for list of all non-successful return codes.  NOTE: The OCC will return an error if the activate power cap sent is not within the min/max power cap range.
<b>Data Length</b>	0x0000
<b>Data</b>	There is no data returned.
<b>Checksum</b>	Xxxx

---

## 2.6 Reset Prep

This command is used to tell the OCC it will be reset. The OCC should update the [OCC-OPAL shared memory](#) “throttle status” to indicate OCC reset and move to standby state. The OCC may also generate FFDC error log prior to returning to this command. After this command HTMGT will send a poll to get the error log id to collect all error logs before the reset. If there is no error log id in the poll the OCC will be reset with no additional error logs collected.

<b>BMC</b>	Should never send.
<b>(H)TMGT</b>	(H)TMGT will send this before putting the OCC into reset or on an FSP system power off/re-IPL

### Reset Prep Command Packet:

<b>Sequence Number</b>	xx
<b>Command Type</b>	0x25
<b>Data Length</b>	0x0002
<b>Data</b>	<b>Byte 1:</b> Version = 0x00 <b>Byte 2:</b> Reason for reset (Except for power off reason all OCCs must update OCC-OPAL shared memory throttle status to reset) <b>0x00 = Non-failure.</b> Code update, external user request. No FFDC error logs should be generated. <b>0x01 = Failure detected on this OCC.</b> FFDC error log should be generated. <b>0x02 = Failure detected on a different OCC.</b> FFDC error log should be generated if this OCC is master. <b>0x03 = Failure detected on a different OCC in different node.</b> No FFDC error log should be generated. An OCC failure in a different node should never be reason for an OCC failure. <b>0xFF = Power off / re-IPL.</b> OCC should stop DCOM and other RTL tasks that still run in OCC standby state.
<b>Checksum</b>	xxxx

**Reset Prep Return Packet:**

<b><i>Sequence Number</i></b>	xx
<b><i>Command Type</i></b>	0x25
<b><i>Return Status</i></b>	0x00 = Success See Appendix A for list of all non-successful return codes.
<b><i>Data Length</i></b>	0x0000
<b><i>Data</i></b>	There is no data returned.
<b><i>Checksum</i></b>	xxxx

---

## 2.7 Send Ambient Temperature

This command is used to send the current ambient temperature and altitude to the OCC.

<b><i>TMGT</i></b>	Send on all OCC state changes to active and when ambient temperature changes.
<b><i>BMC</i></b>	Send when OCC active sensor is set to true and when ambient temperature changes.
<b><i>HTMGT</i></b>	Never send

### Send Ambient Temperature Command Packet:

<b><i>Sequence Number</i></b>	xx
<b><i>Command Type</i></b>	0x30
<b><i>Data Length</i></b>	0x0008
<b><i>Data</i></b>	<p>There are 8 bytes of data:</p> <p><b>Byte 1: Version</b> = 0x00. Currently, 0x00 is the only valid version.</p> <p><b><u>Version 0x00:</u></b></p> <p><b>Byte 2: Ambient Status.</b></p> <p>0x00 = Ambient temperature successfully read 0xFF = Failure reading ambient temperature, next byte ignored</p> <p><b>Byte 3: Current ambient temperature</b> in degrees C. Ignored on failure status.</p> <p><b>Bytes 4-5: Altitude</b> in meters. Set to 0xFFFF if altitude is not available.</p> <p><b>Bytes 6-8: Reserved.</b> 0x000000</p>
<b><i>Checksum</i></b>	xxxx

**Send Ambient Temperature Return Packet:**

<b><i>Sequence Number</i></b>	xx
<b><i>Command Type</i></b>	0x30
<b><i>Return Status</i></b>	0x00 = Success See Appendix A for list of all non-successful return codes.
<b><i>Data Length</i></b>	0x0000
<b><i>Data</i></b>	There is no data returned.
<b><i>Checksum</i></b>	xxxx

---

## 2.8 Debug Pass Through

This command is for debug use only. `tmgtclient -X 0x40 --data .....`

### **Debug Pass Through Command Packet:**

<b><i>Sequence Number</i></b>	xx
<b><i>Command Type</i></b>	0x40
<b><i>Data Length</i></b>	Variable
<b><i>Data</i></b>	There are N bytes of data: <b>Bytes 1-N:</b> User defined.
<b><i>Checksum</i></b>	xxxx

**Debug Pass Through Return Packet:**

<b><i>Sequence Number</i></b>	xx
<b><i>Command Type</i></b>	0x40
<b><i>Return Status</i></b>	0x00 = Success See Appendix A for list of all non-successful return codes.
<b><i>Data Length</i></b>	Variable
<b><i>Data</i></b>	N bytes of data are returned: <b>Bytes 1-N:</b> User defined.
<b><i>Checksum</i></b>	xxxx



---

## 2.9 AME Pass Through

This command is for AMESTER use only.

### **AME Pass Through Command Packet:**

<b><i>Sequence Number</i></b>	xx
<b><i>Command Type</i></b>	0x41
<b><i>Data Length</i></b>	Variable
<b><i>Data</i></b>	There are N bytes of data: <b>Bytes 1-N:</b> User defined.
<b><i>Checksum</i></b>	xxxx

**AME Pass Through Return Packet:**

<b>Sequence Number</b>	Xx
<b>Command Type</b>	0x41
<b>Return Status</b>	0x00 = Success See Appendix A for list of all non-successful return codes.
<b>Data Length</b>	Variable
<b>Data</b>	N bytes of data are returned: <b>Bytes 1-N:</b> User defined.
<b>Checksum</b>	xxxx

---

## 2.10 Get Field Debug Data

This command is used to get data from OCC to be added to an OCC user details section of an error log. HTMGT is called by HBRT to add a user details section for all errors that calls out hardware. HTMGT will generate two user details sections, one with HTMGT specific data and another with the OCC data returned from this command. Only the OCC team has knowledge of what the data returned is and the OCC team is responsible for writing the plug in to format the OCC data user details section created.

<b>BMC</b>	Should never send
<b>(H)TMGT</b>	(H)TMGT will send when requested by HBRT/HWSV

### Get Field Debug Data Command Packet:

<b>Sequence Number</b>	xx
<b>Command Type</b>	0x42
<b>Data Length</b>	0x0001
<b>Data</b>	There is 1 byte of data: <b>Byte 1:</b> Version = 0x00.
<b>Checksum</b>	xxxx

**Get Field Debug Data Return Packet:**

<b><i>Sequence Number</i></b>	xx
<b><i>Command Type</i></b>	0x42
<b><i>Return Status</i></b>	0x00 = Success  See Appendix A for list of all non-successful return codes.
<b><i>Data Length</i></b>	Variable. Not to exceed max (currently 4089)
<b><i>Data</i></b>	1 to M bytes of data are returned:  <b>Bytes 1-M:</b> User Data – OCC defined debug data.
<b><i>Checksum</i></b>	xxxx

---

## 2.11 Mfg Test Command

This command is only intended to be used by an external user (i.e. mnfgvapi and mnfgvfstool) to accomplish various mfg tests. All data and length validation will be done by OCC. Internally TMGT will never initiate a send of any sub command or version of this command.

### Mfg Test Command Packet:

<b>Sequence Number</b>	Xx
<b>Command Type</b>	0x53
<b>Data Length</b>	Sub command dependent
<b>Data</b>	<p>There are x bytes of data (not to exceed maximum):</p> <p><b>Byte 1:</b> Sub Command – Indicates type of mfg test cmd being sent. See following sections for command data details specific to each sub cmd.</p> <p><u><b>Sub Cmd = 0x02:</b></u> Run/Stop Slew Between Two Frequency Points</p> <p><u><b>Sub Cmd = 0x05:</b></u> List Sensors</p> <p><u><b>Sub Cmd = 0x06:</b></u> Get Sensor Information</p> <p><u><b>Sub Cmd = 0x07:</b></u> Enable/Disable Oversubscription Emulation</p> <p><u><b>Sub Cmd = 0x09:</b></u> Run/Stop Memory Slew Between 1-100 Percent</p> <p><u><b>Sub Cmd = 0x0B:</b></u> Read Generated Pstate Table</p> <p><u><b>Sub Cmd = 0x0F:</b></u> Select WOF VRT</p>
<b>Checksum</b>	Xxxx

**Mfg Test Return Packet:**

<b><i>Sequence Number</i></b>	xx
<b><i>Command Type</i></b>	0x53
<b><i>Return Status</i></b>	0x00 = Success See Appendix A for list of all non-successful return codes.
<b><i>Data Length</i></b>	Sub command dependent
<b><i>Return Data</i></b>	0-N bytes of data returned (not to exceed maximum). See following sections for return data details specific to each sub cmd.
<b><i>Checksum</i></b>	xxxx

### 2.11.1 Run/Stop Slew Between Two Frequency Points (Sub Cmd = 0x02)

This is a master OCC only command, the master OCC will broadcast this to all slaves. Slave OCC will reject the command.

#### Run/Stop Slew Between Two Frequency Points Mfg Test Command Packet:

<b>Data Length</b>	0x0009
<b>Data</b>	<b>Byte 1:</b> Sub Cmd = 0x02 <b>Byte 2:</b> Version – see below for defined values  <b><u>Version = 0x30:</u></b> <b>Byte 3:</b> Action: 0x00 = Start 0x01 = Stop  <b>For bytes 4-7 See <a href="#">Frequency Points</a> chapter for valid values</b> <b>Bytes 4-5:</b> Bottom frequency point to slew from <b>Byte 6-7:</b> Top frequency point to slew to <b>Byte 8:</b> Slew step mode: 0x00 = Single step 0x01 = Full slew  <b>Byte 9:</b> Additional delay between steps in ms (applicable for single step only)

#### Run/Stop Slew Between Two Frequency Points Mfg Test Return Packet:

<b>Data Length</b>	0x0006
<b>Return Data</b>	<b>Bytes 1-2:</b> Count of slew number on a stop action (0 on start action) <b>Bytes 3-4:</b> master OCC fstart used in terms of Pstate <b>Bytes 5-6:</b> master OCC fstop used in terms of Pstate

### 2.11.2 List Sensors (Sub Cmd = 0x05)

#### List Sensors Command Packet:

<b>Data Length</b>	0x0009
<b>Data</b>	<p><b>Byte 1:</b> Sub Cmd = 0x05</p> <p><b>Byte 2:</b> Version = 0x00</p> <p><b>Bytes 3-4:</b> The GUID of sensor to start listing</p> <p><b>Byte 5:</b> Present Option:              0x00 = List all sensors from start GUID              0x01 = List non-zero sensors from start GUID</p> <p><b>Bytes 6-7:</b> Location of sensors to be listed this is a bit mask and multiple locations may be selected:              0xFFFF = All locations              0x0001 = System              0x0002 = Processor              0x0004 = Partition              0x0008 = Memory              0x0010 = VRM              0x0020 = OCC              0x0040 = Core              0x0080 = GPU              0x0100 = Quad</p> <p><b>Bytes 8-9:</b> Type of sensors to be listed this is a bit mask and multiple types may be selected:              0xFFFF = All types              0x0001 = Generic              0x0002 = Current              0x0004 = Voltage              0x0008 = Temperature              0x0010 = Utilization              0x0020 = Time              0x0040 = Frequency              0x0080 = Power              0x0200 = Performance              0x0400 = WOF</p> <p>i.e. to display frequency and temperature for processor and cores the “Location” would be 0x0042 and “Type” would be 0x0048. Every combination of location and type bits are processed, sensor must match both location and type to be selected.</p>

#### List Sensors Response Packet:

<b>Data Length</b>	Variable
<b>Return Data</b>	<b>Byte 1:</b> Sensor list truncated.



<b>Data Length</b>	Variable						
	<p><b>0x01</b> = requested list of sensors does not fit within maximum that can be returned. To get remaining sensors send command with starting GUID of the last GUID in response +1.</p> <p><b>0x00</b> = all requested sensors are returned</p> <p><b>Byte 2:</b> Number of sensor data sets that follows.</p> <p>The format of a sensor data set is:</p> <table> <tr> <td><b>Data bytes <math>x</math> &amp; <math>x+1</math></b></td><td><b>Sensor GUID</b></td></tr> <tr> <td><b>Data bytes <math>x+2</math> thru <math>x+17</math></b></td><td><b>Sensor Name</b></td></tr> <tr> <td><b>Data bytes <math>x+18</math> thru <math>x+19</math></b></td><td><b>Sensor Sample</b></td></tr> </table> <p><b>Bytes 3-22:</b> Sensor data set #1.</p> <p><b>Bytes 23-42:</b> Sensor data set #2.</p> <p>...</p>	<b>Data bytes <math>x</math> &amp; <math>x+1</math></b>	<b>Sensor GUID</b>	<b>Data bytes <math>x+2</math> thru <math>x+17</math></b>	<b>Sensor Name</b>	<b>Data bytes <math>x+18</math> thru <math>x+19</math></b>	<b>Sensor Sample</b>
<b>Data bytes <math>x</math> &amp; <math>x+1</math></b>	<b>Sensor GUID</b>						
<b>Data bytes <math>x+2</math> thru <math>x+17</math></b>	<b>Sensor Name</b>						
<b>Data bytes <math>x+18</math> thru <math>x+19</math></b>	<b>Sensor Sample</b>						

### 2.11.3 Get Sensor Information (Sub Cmd = 0x06)

#### Get Sensor Information Command Packet:

<b>Data Length</b>	0x0004
<b>Data</b>	<b>Byte 1:</b> Sub Cmd = 0x06 <b>Byte 2:</b> Version = 0x00 <b>Bytes 3-4:</b> Sensor GUID

#### Get Sensor Information Response Packet:

<b>Data Length</b>	0x002D
<b>Return Data</b>	<b>Bytes 1-2:</b> Sensor GUID <b>Bytes 3-4:</b> Latest sensor sample <b>Bytes 5:</b> Status <b>Bytes 6-9:</b> Accumulator <b>Bytes 10-11:</b> Minimum sample value since last reset <b>Bytes 12-13:</b> Maximum sample value since last reset <b>Bytes 14-29:</b> Sensor name <b>Bytes 30-33:</b> Sensor sample value units <b>Bytes 34-37:</b> Updated frequency <b>Bytes 38-41:</b> Scaling factor <b>Bytes 42-43:</b> Sensor location (bit defined in sub cmd 0x05) <b>Bytes 44-45:</b> Sensor type (bit defined in sub cmd 0x05)

#### 2.11.4 Enable/Disable Oversubscription Emulation (Sub Cmd = 0x07)

This is a master OCC only command, the master OCC will broadcast this to all slaves to enter/exit oversubscription. Slave OCC will reject the command.

##### **Enable/Disable Oversubscription Emulation Command Packet:**

<b>Data Length</b>	0x0004
<b>Data</b>	<b>Byte 1:</b> Sub Cmd = 0x07 <b>Byte 2:</b> Version = 0x00 <b>Byte 3:</b> Action: <b>0x00</b> = Disable oversubscription emulation <b>0x01</b> = Enable oversubscription emulation <b>0xFF</b> = Query. No change to current setting. <b>Byte 4:</b> Reserved

##### **Enable/Disable Oversubscription Emulation Return Packet:**

<b>Data Length</b>	0x0001
<b>Return Data</b>	<b>Byte 1:</b> State of Oversubscription Emulation set by this mfg test command <b>0x00</b> = Oversubscription emulation is disabled <b>0x01</b> = Oversubscription emulation is enabled

### 2.11.5 Run/Stop Memory Slew Between 1-100 Percent (Sub Cmd = 0x09)

This command may be sent to a slave OCC, master will not broadcast.

#### **Run/Stop Memory Slew Command Packet:**

<b>Data Length</b>	0x0003
<b>Data</b>	<b>Byte 1:</b> Sub Cmd = 0x09 <b>Byte 2:</b> Version = 0x00 <b>Byte 3:</b> Action: <b>0x00</b> = Start <b>0x01</b> = Stop

#### **Run/Stop Memory Slew Return Packet:**

<b>Data Length</b>	0x0002
<b>Return Data</b>	<b>Bytes 1-2:</b> Count of slew number on a stop action (0 on start action)

### 2.11.6 Read Generated Pstate Table (Sub Cmd = 0x0B)

This command is to be sent to each OCC to read the Pstate table generated by the PGPE from main memory.

#### NOTES:

- Only 1K can be returned at a time, the caller is responsible to query the size of the generated Pstate table and then read 1K blocks at a time to read the full generated Pstate table.
- This command is returning a raw dump of main memory. The structure of the generated Pstate table is defined in EKB. The caller is responsible for understanding the format of the data and will not be documented here.

#### Read Generated Pstate Table Command Packet:

<b>Data Length</b>	0x0002
<b>Data</b>	<b>Byte 1:</b> Sub Cmd = 0x0B <b>Byte 2:</b> Request: <b>0x00...0x10 = Read Generated Pstate Table Block Offset.</b> Defines 1K offset of generated Pstate table to return. 0x00 gives first 1K of Generated Pstate table. Caller must call with sequential block offsets until full generated Pstate table has been read. If the last block is less than 1K the return data length will be only for the remaining pstate table. Request for blocks beyond the Generated Pstate Table length will result in an "Invalid Data Field" error return code.  <b>0xFF = Query.</b> Returns the HOMER offset of the beginning of the generated Pstate table and the size of the table.

#### Read Generated Pstate Table Return Packet:

<b>Data Length</b>	Variable. 0x0008 (Query) or max of 0x0400 (Read Generated Pstate Table Block Offset)
<b>Return Data</b>	<b><u>Request Block Offset = 0x00...0x10</u></b> <b>Bytes 1... max of 1024:</b> Generated Pstate Table starting address + (block offset * 1024)  <b><u>Query = 0xFF</u></b> <b>Bytes 1-4:</b> Generated Pstate Table HOMER Offset as an OCI PBA memory address (i.e. 0x8xxxxxxx) <b>Bytes 5-8:</b> Generated Pstate Table Length

### 2.11.7 Select WOF VRT (Sub Cmd = 0x0F)

This command is to select a specific WOF Voltage Ratio Table to use regardless of the one calculated by the WOF algorithm. The WOF algorithm will continue to run but the VRT will not change from the one selected via this command until another Select WOF VRT command is sent.

#### Select WOF VRT Command Packet:

<b>Data Length</b>	Variable dependent on action.
<b>Data</b>	<p><b>Byte 1:</b> Sub Cmd = 0x0F</p> <p><b>Byte 2:</b> Action.</p> <p style="padding-left: 40px;"><b>0x00 – Query.</b> Used to query size of each dimension and current overwrite state.</p> <p style="padding-left: 40px;"><b>0x01 – Overwrite VRT dimension(s).</b> Used to overwrite index for one or more of the VRT dimensions.</p> <p><b><u>Action = 0x00 (Query)</u></b> No additional data bytes.</p> <p><b><u>Action = 0x01 (Overwrite VRT dimension(s))</u></b></p> <p><b>Byte 3: Vcs Index.</b> Indicates Vcs index to use, starting from 0 for the first Vcs. An index greater than the last entry in the VRT will result in the last entry being used. 0xFF indicates not to override Vcs and OCC will use calculated Vcs for selecting VRT.</p> <p><b>Bytes 4: Vdd Index.</b> Indicates Vdd index to use, starting from 0 for the first Vdd. An index greater than the last entry in the VRT will result in the last entry being used. 0xFF indicates not to override Vdd and OCC will use calculated Vdd for selecting VRT.</p> <p><b>Bytes 5: IO Power Index.</b> Indicates IO Power index to use, starting from 0 for the first IO Power. An index greater than the last entry in the VRT will result in the last entry being used. 0xFF indicates not to override IO Power and OCC will use calculated IO Power for selecting VRT.</p> <p><b>Bytes 6: Ambient Index.</b> Indicates ambient index to use, starting from 0 for the first ambient. An index greater than the last entry in the VRT will result in the last entry being used. 0xFF indicates not to override ambient and OCC will use actual ambient for selecting VRT.</p> <p><b>Bytes 7: V Ratio Index.</b> Vratio index to use within the selected VRT, starting with 0 for the first entry in the VRT. The Vratio is used by the PGPE and is sent by the OCC to the PGPE for usage. An index greater than the last entry in the VRT will result in the last entry being used. 0xFF indicates not to override Vratio and PGPE will use the calculated Vratio.</p>

#### Select WOF VRT Return Packet:

<b>Data Length</b>	0x0028
--------------------	--------

<b>Data Length</b>	0x0028
<b>Return Data</b>	<p><b>Bytes 1-2:</b> Vcs 0. Percentage corresponding to the first Vcs index 0.</p> <p><b>Bytes 3-4:</b> Vcs step. Percentage for each Vcs step to next index.</p> <p><b>Byte 5:</b> Last Vcs Index. Highest possible index for Vcs</p> <p><b>Byte 6:</b> Current Vcs override index. Current index set for Vcs. 0xFF if Vcs is not overwritten</p> <p><b>Bytes 7-8:</b> Reserved = 0x0000</p> <p><b>Bytes 9-10:</b> Vdd 0. Percentage corresponding to the first Vdd index 0.</p> <p><b>Bytes 11-12:</b> Vdd step. Percentage for each Vdd step to next index.</p> <p><b>Byte 13:</b> Last Vdd Index. Highest possible index for Vdd</p> <p><b>Byte 14:</b> Current Vdd override index. Current index set for Vdd. 0xFF if Vdd is not overwritten</p> <p><b>Bytes 15-16:</b> Reserved = 0x0000</p> <p><b>Bytes 17-18:</b> IO Power 0. Percentage corresponding to the first IO power index 0.</p> <p><b>Bytes 19-20:</b> IO Power step. Percentage for each IO Power step to next index.</p> <p><b>Byte 21:</b> Last IO Power Index. Highest possible index for IO Power</p> <p><b>Byte 22:</b> Current IO Power override index. Current index set for IO Power. 0xFF if IO Power is not overwritten</p> <p><b>Bytes 23-24:</b> Reserved = 0x0000</p> <p><b>Bytes 25-26:</b> Ambient 0. Percentage corresponding to the first Ambient index 0.</p> <p><b>Bytes 27-28:</b> Ambient step. Percentage for each Ambient step to next index.</p> <p><b>Byte 29:</b> Last Ambient Index. Highest possible index for Ambient</p> <p><b>Byte 30:</b> Current Ambient override index. Current index set for Ambient. 0xFF if Ambient is not overwritten</p> <p><b>Bytes 31-32:</b> Reserved = 0x0000</p> <p><b>Bytes 33-34:</b> Vratio 0. Percentage corresponding to the first Vratio index 0.</p> <p><b>Bytes 35-36:</b> Vratio step. Percentage for each Vratio step to next index.</p> <p><b>Byte 37:</b> Last Vratio Index. Highest possible index for Vratio</p> <p><b>Byte 38:</b> Current Vratio override index. Current index set for Vratio. 0xFF if Vratio is not overwritten</p> <p><b>Bytes 39-40:</b> Reserved = 0x0000</p>

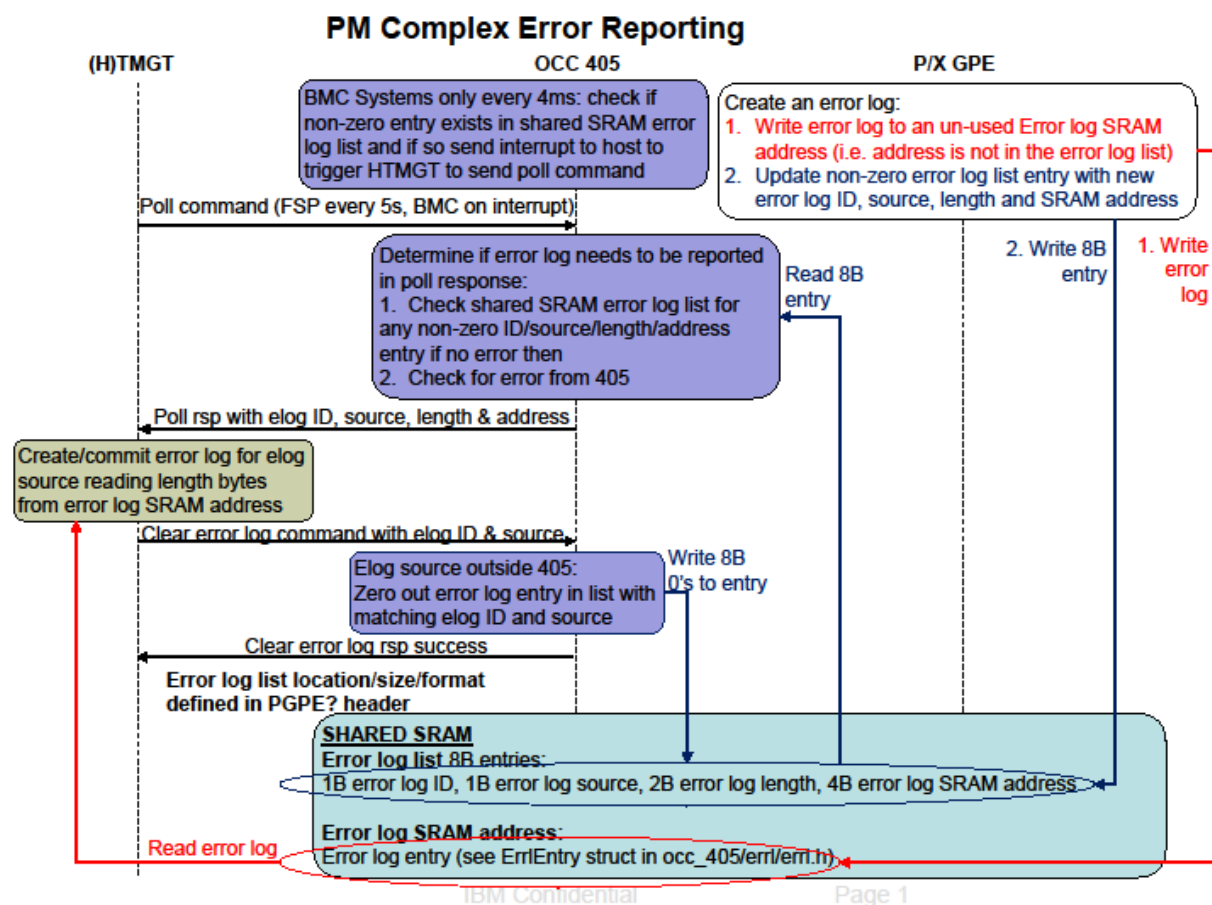
### 3 Error Handling

### 3.1 OCC Errors

When an OCC detects an error, it writes the error to some location in SRAM and sends a “service required” attention to host. In response to the attn HTMGT sends a poll, the poll response includes the error log ID, starting SRAM address, length and source of the error to be collected. HTMGT reads and process the error log from SRAM per defined format in [Read OCC Error Log from SRAM](#) section.

### 3.2 PM Complex Errors Outside of OCC 405

Errors generated by Hcode will be reported and collected the same way as an error being reported by the OCC. Hcode must follow the same error log format for (H)TMGT processing to be common with reading an OCC Error Log from SRAM.





---

### 3.3 Reading Error Log from SRAM – Format in SRAM

To read an error log, (H)TMGT will read error log length bytes from the OCC poll response starting at the error log start address from the same OCC poll response.

NOTE: (H)TMGT will append everything from Byte 1 (checksum) to the end of the error log as a user data section of the error log. In addition (H)TMGT is processing data to build other fields of the error log i.e. mod id, user data words, callouts...

Order in SRAM starting from Error log start address in OCC poll response:

**Bytes 1-2:** Checksum. Checksum is two byte sum (ignoring overflow) of all bytes starting with and including the version byte thru the last byte of the error log defined by the error log length from OCC poll response.

**Byte 3:** Version. Indicate format version of error log to parse data.

**Byte 4:** Error Log ID – Due to limited memory and re-use of same memory for future error logs the ID is used to know that the correct error log at the SRAM address is being read.

**Byte 5:** Reason Code – (H)TMGT will use this as the LSB for the reason code that this error will be committed with, the MSB for the SRC will be determined based on the error log source from the poll response. i.e. OCC 405 will be 0x2A, PGPE will be 0x2E....

**Byte 6:** Severity – Indicates the severity of the error. Depending on (H)TMGT processing (H)TMGT may change this severity when committing the error log.

Severity	Description
0x00	Informational Only.
0x01	Predictive Error.
0x02	Un-recoverable Error.

**Byte 7:** Actions – Bit defined and indicates special processing that HTMGT may need to do in order to process the error. Multiple bits may be set and HTMGT will process these bits in order from lsb to msb.

Bit(s)	Description
(lsb) 7:5	Reserved
4	<b>Manufacturing error.</b> FSP only. If in mfg mode change severity to unrecoverable to guarantee it is seen and will terminate mfg, else the severity from OCC (for this case should be informational) will remain unchanged for the field. Example where this may be used: Oversubscription asserted (should not happen in mfg)
3	<b>Force error to be posted.</b> Currently only used on BMC systems (HTMGT) to force the error to be sent to the BMC. This is to ensure informational errors will be seen i.e. PGPE errors requiring a reset to recover want to see the additional PGPE data.
2	<b>Reset due to WOF Error.</b> An error running WOF was encountered requiring an OCC reset. (H)TMGT will change the error severity to informational if the WOF retry count (3) has not been reached and not running in manufacturing mode.

	NOTE: Resetting due to WOF should NOT count towards a permanent safe mode reset count. If WOF retry count has been met (H)TMGT will still do an OCC reset but will tell the OCC WOF is NOT supported in the frequency config data to disable WOF until the next IPL/reset-reload i.e. same conditions that exit safe mode.
1	<b>Safe Mode Required.</b> Error is critical with no hope of recovery from an OCC reset; system will be put in safe mode (i.e. OCCs held in reset). Cases for this is a system checkstop and EPOW detection. (H)TMGT should make this error informational since this error is a side effect of something outside the PM complex.
0 (msb)	<b>Reset Required.</b> Error is critical but may recover by resetting the OCC. (H)TMGT will change the error severity to informational if the reset retry count was not reached and not running in manufacturing mode. NOTE: If reset retry count has been met the OCCs will remain in reset and the severity of the error is NOT changed except if this was also a manufacturing error.

**Byte 8:** Num callouts

**Bytes 9-10:** Extended Reason Code. (H)TMGT will use this as the lower 2 bytes for the error log user data word 4. 405 NOTE: Upper 2 bytes of user data word 4 is OCC module ID.

**Bytes 11-12:** Maximum error log size including all user details

**Bytes 13-16:** Reserved.

**Bytes 17-112:** Callouts – Hard code space for maximum of 6. Actual number of callouts is defined in “Num callouts” byte. Each of the callouts contains 16 bytes in order:

<b>Callout Bytes x thru x+7</b>	<b>Callout.</b> Callout value format type defined in “Type” byte.
<b>Callout Byte x+8</b>	<b>Type.</b> Type of callout: <b>0x01</b> – Sensor ID (following 8 Callout bytes is either sensor ID (IPMI) or HUID (no IPMI)) <b>0x02</b> – TMGT-OCC Component ID (following 8 Callout bytes are 7 0x00’s followed by the HTMGT-OCC ID defined in Appendix D) <b>0x03</b> – GPU ID (following 8 Callout bytes is a sensor ID for a GPU)
<b>Callout Byte x+9</b>	<b>Priority.</b> Priority for this callout: <b>0x01</b> - Low priority. <b>0x02</b> - Medium priority. <b>0x03</b> - High priority.
<b>Callout Bytes x+10 thru x+15</b>	<b>Reserved.</b>

**Bytes 113 thru Error Log Length from OCC poll response that contained the error log start address this error log is for:** Additional User Details defined by the error log source owner.

<b>Size</b>	<b>Name</b>	<b>Description</b>
1 byte	Version	User Details version
1 byte	Reserved	
2 bytes	modId	Module ID. (H)TMGT: The lower byte will be used as the error log module ID. The full 2 bytes will be available as the upper 2 bytes of user data word 4. NOTE: lower 2 bytes of user data word 4 is the extended RC
4 bytes	405: fclipHistory	Frequency clip history
	PPE: procVersion	PPE Processor Version
8 bytes	timestamp	OCC/PPE Time stamp
2 bytes	405: occId / occRole	1 byte OCC ID and 1 byte OCC role that error log is from
	PPE: ppId	PPE Instance in Chip
1 byte	405: operatingState	OCC state
	PPE: Reserved	
1 byte	committed	Indicates if log is committed
4 bytes	userData1	(H)TMGT: building error log appended as user data word 1
4 bytes	userData2	(H)TMGT: building error log appended as user data word 2
4 bytes	userData3	(H)TMGT: building error log appended as user data word 3
2 bytes	entrySize	Size of complete error log
2 bytes	userDetailEntrySize	User Details Size
variable	User Defined data	To end of error log: OCC/PPE Defined data

---

## 3.4 Errors Requiring OCC Reset

Any OCC requiring a reset will result in running the OCC Reset (safe mode) procedure which resets the whole power management complex (all OCCs, PGPEs, XGPEs...). The power management complex will be held in reset (i.e. system in safe mode) after reaching 3 reset attempts due to the same OCC failing.

### 1.1.1 BMC Detected Reasons for OCC Reset

The BMC must send a request to reset the OCCs when it detects one of the following and the “OCC Active” sensor is TRUE with no checkstop present:

- Communication failure to an OCC defined in [BMC-OCC Communication Failure](#) section.
- Number of bits set in “OCCs Present” from master poll response does not match the number of OCCs the BMC is communicating with.
- Current OCC State byte in poll response is “Safe” for one minute and the “OCC Active” sensor is TRUE.

#### 1.1.1.1 BMC Request for OCC Reset

Any request for an OCC reset will be resetting the whole power management complex in the system. When BMC determines that it needs to request a reset the following must be done:

1. BMC generates an error log with the reason for reset to aid in debug.
2. BMC updates “OCC Active” sensor to FALSE for all OCCs
3. BMC sends PLDM message to host with chip ID of failing OCC
4. Host passes PLDM message along to HBRT process\_pldm\_message
5. HBRT calls HTMGT OCC error handling function with the chip ID
6. HTMGT runs [OCC Reset Procedure](#).

### 1.1.2 (H)TMGT Detected Reasons for OCC Reset

(H)TMGT will reset the OCCs when it sees one of the following:

- (H)TMGT can’t communicate with an OCC
- OCC fails to make requested OCC state change
- “OCCs Present” byte in poll response does not match (H)TMGT view of OCCs present
- OCC poll response not reporting correct OCC role that (H)TMGT set

### 1.1.3 OCC Detected Reasons for OCC Reset

OCC will create an error log and request a reset for the following:

- Processor SCOM failure
- Failure to maintain a hard power cap
- Timeout reading processor temperatures
- Failure from SSX operating system
- Failure within the power management complex i.e. PGPE error/halted
- GPE halted due to checkstop

HTMGT will process the reset request from the OCC as part of collecting the error log from the OCC and run the [OCC Reset Procedure](#).

---

### 3.5 OCC Reset Procedure (Safe Mode)

This procedure will run when any OCC needs a reset. This will be resetting the whole power management complex. When bringing the OCCs active again the same process is followed as a system boot documented in the [OCC Boot Process chapter](#).

1. HTMGT: Increment OCC reset count for failing OCC
2. HTMGT: Send PLDM message to host to set every "OCC Active" sensor to FALSE
3. HOST: Passes PLDM message to BMC
4. BMC: Process PLDM message and updates OCC Active sensor to FALSE
5. HTMGT: Collect error from failing OCC, check SRAM response buffer for an Ex critical OCC error and log error for OCC reset
6. HTMGT: Send "reset prep" command to each OCC. This will cause OCC to update "Throttle Status" and "OCC State" in OCC-OPAL shared memory interface. NOTE: This may fail depending on OCC error, HTMGT will ignore failures.
7. HTMGT: Call HBRT to run HWP to put all OCCs into reset.
8. HTMGT: OCC reset count > 3?
  - YES: OCCs stay in reset, system is in safe mode until next boot
  - NO:
    1. Call HBRT to load and start all OCCs.
    2. HTMGT communicates with OCCs and make them active and informs the BMC the OCCs are now active

---

## 3.6 Error Scenarios

### 3.6.1 (H)TMGT-OCC Communication Failure

If any of the steps to send a command or read response from an OCC fails or there is a checksum failure then the whole command will be retired once. If the retry fails then all OCCs will be reset. If the max OCC reset count has been reached for the failing OCC then all OCCs will be held in reset (i.e. safe mode) else the OCCs will be taken out of reset and brought active again. (H)TMGT must be able to communicate with all OCCs.

### 3.6.2 BMC-OCC Communication Failure

A communication failure is defined as one of the following:

- Response Checksum failure
- Sequence number mismatch after command timeout has been reached.
- OCC Return Status still “In Progress” after command timeout has been reached. See [“Command Summary Table”](#) section for recommended timeout by command.
- Any non-successful Return Code
- Any failure sending command or reading response

When any of the above communication failures occur, the BMC should first verify that the “OCC Active” sensor is TRUE. If the OCCs are not active the error should be ignored and communication with the OCC should not be retried until the “OCC Active” sensor is TRUE. If the “OCC Active” sensor is TRUE the command should be retried twice. If the command still fails after two retries and the “OCC Active” sensor is still “TRUE” and there is no checkstop the error is valid and a request to reset the OCCs should be sent.

### 3.6.2.1 BMC-OCC Communication Failure Handling Flow

### 3.6.3 OCC Fails to Load or Fails to go Active

**HTMGT Actions:** Prior to booting the OCCs the “OCC Active” sensor will be FALSE. In the case that there is any failure loading or configuring the OCC to an active state HTMGT will not make the call to update the “OCC Active” sensor and it will remain FALSE as shown in [OCC Boot Process](#) flow. HTMGT will put the OCCs in reset if there is a failure going active which will cause any BMC-OCC communication to fail.

**BMC Actions:** BMC should not be trying to communicate with the OCCs when the “OCC Active” sensor is FALSE. If the BMC does try to communicate, the communication will fail and BMC should be following [BMC-OCC Communication Failure Handling Flow](#) and see the “OCC Active” sensor is FALSE and stop all communication to the OCCs.

### 3.6.4 Checkstop

Main memory cannot be used. HTMGT is not running. BMC cannot talk to OCC.

**OCC Actions:** OCCs move themselves to safe state.

**BMC Actions:** On any OCC communication failure the BMC must be checking for a checkstop and stop communication to the OCCs as shown in the [BMC-OCC Communication Failure Handling Flow](#).

### 3.6.5 OCC Detects an Error Requiring Reset

**OCC Actions:** Creates the error log, move to safe state and send attention to HTMGT to collect the error and reset OCCs. Safe state will be reflected in the OCC poll response “Current OCC State” byte.

**HTMGT Actions:** Process error log and follow [OCC Reset Procedure](#) which will update the OCC Active sensor to FALSE.

**BMC Actions:**

- Any poll before HTMGT makes the call to update the “OCC Active” sensor may be successful; however, the BMC should be checking the “Current OCC State” byte in the poll response which will be safe, and BMC should not use the sensor data in the response.
- Any communication once OCCs are put in reset will fail and the BMC should follow the [BMC-OCC Communication Failure Handling Flow](#) to recognize that the OCCs are no longer active and stop communication.

### 1.1.4 Attention Line to Host is Broken

This will not be detected until the OCC has an error that requires a reset.

**OCC Actions:** Creates the error log and move to safe state, safe state will be reflected in the OCC poll response “Current OCC State” byte. The attention to HTMGT to collect the error



and reset OCCs will not be processed due to broken attention line.

**HTMGT Actions:** Process OCC reset request from BMC, at this point the errors from OCC will be collected.

**BMC Actions:** Check “Current OCC State” byte in poll response for safe and send OCC reset request to HTMGT after defined time of being in safe state. Time defined in [BMC Detected Reasons for OCC Reset](#) section.

### 1.1.5 OCC Takes a Kernel Exception and goes to Halt

**OCC Actions:** As part of halt OCC collects and writes data for debug to the SRAM response buffer with an Ex (Critical OCC Error) return code in the return status. OCC is no longer running, watchdogs will expire moving the system into a safe state.

**HTMGT Actions:** Process OCC reset request from BMC. Part of the reset request HTMGT will read the SRAM response buffer and see the Ex return status to collect the data into an error log for debug.

**BMC Actions:** All communication to the OCC will fail with non-successful return code (Critical OCC Error Ex return code). BMC will follow the [BMC-OCC Communication Failure Handling Flow](#) and will send request for OCC reset to HTMGT.

### 1.1.6 OCC-BMC Interface is Broken

**OCC Actions:** Nothing. OCC is unaware.

**BMC Actions:** Log an error (there will be no error from the OCC) and after following retries in [BMC-OCC Communication Failure Handling Flow](#) request OCC reset. If the error is a hard failure after going thru three OCC resets the OCCs will be held in reset.

**HTMGT Actions:** Process OCC reset request from BMC. In this case there will be no errors from OCC since the OCC was unaware of the failed BMC communication. HTMGT will log a generic OCC reset error but the BMC error log will have the data as to why the reset was needed.

### 1.1.7 OCCs Held in Reset

After three resets (per system boot) due to the same OCC failing the OCCs will be held in reset. The “OCC Active” sensor will stay FALSE and the BMC should not be communicating.

---

## 4 OCC Boot and Code Update Process

On all system types, after the OCC is loaded and taken out of reset it will default to “standby” state and wait for configuration data from (H)TMGT and for (H)TMGT to send Set State command to Active. There is no thermal or power monitoring while the OCC is in standby state. When OCC is told to go active it will populate [OCC-OPAL shared memory interface](#) with ‘valid’ and all Pstate data.

---

### 4.1 OCC Load/Start Process on BMC with PHYP

1. PHYP determines location for OCC common area and HOMERs
2. HBRT loads and starts the PM complexes calling PHYP for OCC Common Area and HOMER locations as part of the startup
3. HBRT calls HTMGT to setup OCCs
4. HTMGT communicates with OCCs and make them active and informs the BMC the OCCs are now active
5. PHYP receives host attention from each OCC and reads OCC state from HOMER to know each OCC is active

---

### 4.2 OCC Load/Start Process on BMC with OPAL

1. HOMER is placed at the top of memory
2. HB loads images from PNOR via OPAL
3. HB starts PM complexes
4. HBRT calls HTMGT to setup OCCs
5. HTMGT communicates with OCCs and make them active and informs the BMC the OCCs are now active
6. HB adds HOMER to reserved memory space in HDAT
7. OPAL is started

---

### 4.3 Additional BMC Handling after OCCs Active with PowerVM

In PowerVM the OCC is in control of frequency and modes must be settable via BMC. This additional customer settable mode information must be sent to the OCCs after they have become active. On every OCC state change to active the BMC must send the following commands to the OCC

1. Set mode and state (set mode only to set user set mode, state is un-changed)
2. Idle Power Save Parameters

---

## 5 Frequency Points

The OCC will be reading the following frequency points from the OCC Pstate Parameter Block.

\* Enum is used to define the frequency point when setting Static Frequency Point mode and for processor auto slew mfg test command.

Frequency Point	Enum*	Notes
VPD Curve Fit Point 0, 1 ... 7	0x1000 0x1001 : 0x1007	<ul style="list-style-type: none"><li>➤ Unique per chip</li><li>➤ Points in VPD</li><li>➤ Used for interpolation</li></ul>
Power Save	0x2000	<ul style="list-style-type: none"><li>➤ System wide</li><li>➤ Maximum of all chips "Power Save" VPD point aka minimum frequency the chip can run</li></ul>
WOF Base	0x2001	<ul style="list-style-type: none"><li>➤ System wide</li><li>➤ Roughly equivalent to legacy turbo</li><li>➤ Point in VPD should be same for all chips of a given sort</li></ul>
Ultra Turbo	0x2002	<ul style="list-style-type: none"><li>➤ System wide</li><li>➤ Point in VPD should be same for all chips of a given sort</li></ul>
Fmax	0x2003	<ul style="list-style-type: none"><li>➤ Unique per chip</li><li>➤ Point in VPD</li><li>➤ Maximum frequency the chip can run at</li></ul>
Mode Disabled	0x2004	<ul style="list-style-type: none"><li>➤ System wide</li><li>➤ Roughly equivalent to P9 nominal</li><li>➤ Point in VPD ("fixed frequency") should be same for all chips of a given sort</li></ul>
Bottom Throttle Space	0x4000	<ul style="list-style-type: none"><li>➤ Pstate is set to lowest possible (since Pstates increase going down from Fmax this is the highest Pstate number)</li><li>➤ Lowest frequency and most severe throttle setting</li></ul>
Pstate	0xFFpp	<ul style="list-style-type: none"><li>➤ pp is Pstate to set</li><li>➤ if sent a larger Pstate than supported OCC will clip to largest Pstate (aka Bottom Throttle Space)</li></ul>

---

## 5.1 Configuration File

Defined in the configuration file.

Frequency	Notes
Boot Frequency	<ul style="list-style-type: none"><li>➤ Frequency that the system will be booted at must be low enough to keep system power and thermal safe until OCCs are active</li><li>➤ Must be <math>\geq</math> ATTR_MIN_FREQ_MHZ (cannot be below epsilon value)<ul style="list-style-type: none"><li>○ Host Boot to make this check and if it isn't log error and raise boot to ATTR_MIN_FREQ_MHZ</li></ul></li></ul>
OCC Timebase Frequency	<ul style="list-style-type: none"><li>➤ Written to HOMER by Host Boot. OCC reads directly from HOMER.</li></ul>

---

## 6 OCC Poll Response Sensor Data Format Definitions

This chapter defines the formats for each sensor type that may be returned in the Status and Sensor Poll response.

### NOTES:

- Sensor ID field is always 4 bytes (MSB first) and is used to give a correlation for reporting data.
- When IPMI is supported this will be the IPMI sensor ID with 3 bytes of 0x00 followed by the 1 byte IPMI sensor ID.
- On BMC systems when IPMI is not supported this is a 4 byte HUID.
- FSP systems will always use the HUID.

---

### 6.1 Temperature Sensors (“TEMP”)

This is available in master and slave OCC poll responses.

Sensor Eye Catcher = “TEMP”

Sensor Version = 0x10

Sensor Length = 0x08

Format for one sensor and repeated for Number of Sensors:

Offset	
0x00	<b>Sensor ID</b> – 4 bytes. To identify what the temperature represents
0x04	<b>FRU Type</b> – 1 byte. Indicates what type of FRU the temperature is for: 0x00: Processor (core temperature) 0x01: Internal Memory Controller 0x02: DIMM (on board sensor for DRAM) 0x03: Memory Controller + DIMM. Thermal sensor is located to cover both DRAM an MC. <u>(some EUH DDIMMs used) / For 4U DDIMMs this is “DIMM” type to allow different thresholds for 2U DDIMMs (using “DIMM” fru type 2) and 4U DDIMMs</u> 0x04: GPU core 0x05: GPU memory 0x06: VRM Vdd 0x07: PMIC (on board sensor for PMIC) 0x08: Memory Controller external sensor <u>(not used) / For 4U DDIMMs this is “PMIC” type to allow different thresholds for 2U DDIMMs (using “PMIC” fru type 7) and 4U DDIMMs</u> 0x09: Processor I/O Ring
0x05	<b>Current Reading</b> – 1 byte. Current temperature reading in degrees C. <b>0x00</b> = Not present / Unavailable <b>0xFF</b> = Error reading the temperature sensor after MAX_READ_TIMEOUT. This is indication that fans should be increased. Until TIMEOUT is reached

	OCC will continue to return the last successful read or 0 if sensor was never successfully read.
<b>0x06</b>	<b>Throttle Temperature</b> – 1 byte. Temperature in degrees C in which the OCC will reduce performance i.e. throttling (memory), DVFS (processor)... in order to lower the temperature. The BMC/FSP should ensure that fan increases start at a temperature lower than the throttle temperature to try to keep the temperature from reaching the point of impacting performance. <b>0xFF</b> = No throttle temperature. The OCC does not take any action based on this temperature. Temperature is used for fan control only.
<b>0x07</b>	<b>Reserved = 0x00</b> – 1 byte reserved for future use.

---

## 6.2 Frequency Sensors (“FREQ”)

This is available in master and slave OCC poll responses.

Sensor Eye Catcher = “FREQ”

Sensor Version = 0x02

Sensor Length = 0x06

Format for one sensor and repeated for Number of Sensors:

<b>Offset</b>	
<b>0x00</b>	<b>Sensor ID</b> – 4 bytes. To identify what the frequency represents
<b>0x04</b>	<b>Current Reading</b> – 2 bytes. Current frequency in MHz

## 6.3 Power Sensors (“POWR”)

This is only available from the master OCC poll response if an APSS is present. If the system does not have an APSS all OCCs will return the sensor version for no APSS. All values are output power, if input power is required the output power must be converted to input by using a ps efficiency factor.

### 6.3.1 System has APSS (Master only)

Sensor Eye Catcher = “POWR”

Sensor Version = 0x02 (APSS is present)

Sensor Length = 0x16

Format for one sensor and repeated for Number of Sensors:

Offset																																									
0x00	<b>Sensor ID</b> – 4 bytes. Used to report power.																																								
0x04	<b>Function ID</b> – 1 byte. Identifies what the power reading is for. This is the ADC_CHANNEL_ID in xml file: <table><tr><td>0 = Not Used.</td><td>20 Reserve. (12V Voltage sense)</td></tr><tr><td>1 = Memory Proc 0</td><td>21 Reserve. (ground remote sense)</td></tr><tr><td>2 = Memory Proc 1</td><td>22 = Total System Power</td></tr><tr><td>3 = Memory Proc 2</td><td>23 = Memory Cache</td></tr><tr><td>4 = Memory Proc 3</td><td>24 = Proc 0 GPU 0</td></tr><tr><td>5 = Processor 0</td><td>25 = Memory Proc 0-0 (2<sup>nd</sup> channel of memory power for Proc 0)</td></tr><tr><td>6 = Processor 1</td><td>26 = Memory Proc 0-1 (3<sup>rd</sup> channel of memory power for Proc 0)</td></tr><tr><td>7 = Processor 2</td><td>27 = Memory Proc 0-2 (4<sup>th</sup> channel of memory power for Proc 0)</td></tr><tr><td>8 = Processor 3</td><td>28 = Reserve (12V standby current)</td></tr><tr><td>9 = Processor 0 cache/io/pcie</td><td>29 = Proc 0 GPU 1</td></tr><tr><td>10 = Processor 1 cache/io/pcie</td><td>30 = Proc 0 GPU 2</td></tr><tr><td>11 = Processor 2 cache/io/pcie</td><td>31 = Proc 1 GPU 0</td></tr><tr><td>12 = Processor 3 cache/io/pcie</td><td>32 = Proc 1 GPU 1</td></tr><tr><td>13 = IO A</td><td>33 = Proc 1 GPU 2</td></tr><tr><td>14 = IO B</td><td>34 = PCIe</td></tr><tr><td>15 = IO C</td><td>35 ... 38 = Vpcie DCM0 ... DCM 3</td></tr><tr><td>16 = Fans A</td><td>39 ... 42 = Vio DCM0 ... DCM 3</td></tr><tr><td>17 = Fans B</td><td>43 = AVdd Total</td></tr><tr><td>18 = Storage A</td><td></td></tr><tr><td>19 = Storage B</td><td></td></tr></table>	0 = Not Used.	20 Reserve. (12V Voltage sense)	1 = Memory Proc 0	21 Reserve. (ground remote sense)	2 = Memory Proc 1	22 = Total System Power	3 = Memory Proc 2	23 = Memory Cache	4 = Memory Proc 3	24 = Proc 0 GPU 0	5 = Processor 0	25 = Memory Proc 0-0 (2 <sup>nd</sup> channel of memory power for Proc 0)	6 = Processor 1	26 = Memory Proc 0-1 (3 <sup>rd</sup> channel of memory power for Proc 0)	7 = Processor 2	27 = Memory Proc 0-2 (4 <sup>th</sup> channel of memory power for Proc 0)	8 = Processor 3	28 = Reserve (12V standby current)	9 = Processor 0 cache/io/pcie	29 = Proc 0 GPU 1	10 = Processor 1 cache/io/pcie	30 = Proc 0 GPU 2	11 = Processor 2 cache/io/pcie	31 = Proc 1 GPU 0	12 = Processor 3 cache/io/pcie	32 = Proc 1 GPU 1	13 = IO A	33 = Proc 1 GPU 2	14 = IO B	34 = PCIe	15 = IO C	35 ... 38 = Vpcie DCM0 ... DCM 3	16 = Fans A	39 ... 42 = Vio DCM0 ... DCM 3	17 = Fans B	43 = AVdd Total	18 = Storage A		19 = Storage B	
0 = Not Used.	20 Reserve. (12V Voltage sense)																																								
1 = Memory Proc 0	21 Reserve. (ground remote sense)																																								
2 = Memory Proc 1	22 = Total System Power																																								
3 = Memory Proc 2	23 = Memory Cache																																								
4 = Memory Proc 3	24 = Proc 0 GPU 0																																								
5 = Processor 0	25 = Memory Proc 0-0 (2 <sup>nd</sup> channel of memory power for Proc 0)																																								
6 = Processor 1	26 = Memory Proc 0-1 (3 <sup>rd</sup> channel of memory power for Proc 0)																																								
7 = Processor 2	27 = Memory Proc 0-2 (4 <sup>th</sup> channel of memory power for Proc 0)																																								
8 = Processor 3	28 = Reserve (12V standby current)																																								
9 = Processor 0 cache/io/pcie	29 = Proc 0 GPU 1																																								
10 = Processor 1 cache/io/pcie	30 = Proc 0 GPU 2																																								
11 = Processor 2 cache/io/pcie	31 = Proc 1 GPU 0																																								
12 = Processor 3 cache/io/pcie	32 = Proc 1 GPU 1																																								
13 = IO A	33 = Proc 1 GPU 2																																								
14 = IO B	34 = PCIe																																								
15 = IO C	35 ... 38 = Vpcie DCM0 ... DCM 3																																								
16 = Fans A	39 ... 42 = Vio DCM0 ... DCM 3																																								
17 = Fans B	43 = AVdd Total																																								
18 = Storage A																																									
19 = Storage B																																									
0x05	<b>APSS Channel</b> – 1 byte 0–15. Indicates the APSS channel that the power was read from. This is the ADC_CHANNEL_ASSIGNMENT in xml file.																																								
0x06	<b>Reserved</b> – 2 bytes reserved = 0x0000																																								
0x08	<b>Update Tag</b> – 4 bytes. Count of number of 500us samples represented in Accumulator. Used for time derived sensor.																																								



<b>0x0C</b>	<b>Accumulator</b> – 8 bytes. Accumulation of 500us power readings
<b>0x14</b>	<b>Current Reading</b> – 2 bytes. Most recent 500us reading in watts

### 6.3.2 No APSS (All OCCs Report)

Sensor Eye Catcher = “POWR”

Sensor Version = 0xA0 (no APSS)

Sensor Length = 0x44

This is not repeated, number of sensors in poll response will be 1 for this POWR version. Processor power is calculated from AVS bus readings and is for only the processor that the OCC poll response is from. Total system power is read from a SPI attached chip (if present) by the master OCC and sent to the slave OCCs for all OCCs to report.

<b>Offset</b>	
<b>0x00</b>	<b>System Power Sensor ID</b> – 4 bytes. Used to report system power. If no SPI attached chip to read system power this and the following fields for system power will be 0x00000000 (NOTE: sensor ID may also be 0 if not defined in xml)
<b>0x04</b>	<b>System Power Update Time</b> – 2 bytes. The time in microseconds that the system power is read by the OCC
<b>0x06</b>	<b>Current System Power</b> – 2 bytes. Most recent system power reading in watts
<b>0x08</b>	<b>System Power Update Tag</b> – 4 bytes. Count of number of update time samples represented in the System Power Accumulator.
<b>0x0C</b>	<b>System Power Accumulator</b> – 8 bytes. Accumulation of system power readings
<b>0x14</b>	<b>Reserved</b> – 4 bytes. 0x00000000
<b>0x18</b>	<b>Processor Power Update Time</b> – 2 bytes. The time in microseconds that processor power is read by the OCC
<b>0x1A</b>	<b>Current Total Processor Power</b> – 2 bytes. Most recent total processor power reading in watts. This includes Vdd + Vdn
<b>0x1C</b>	<b>Total Processor Power Update Tag</b> – 4 bytes. Count of number of processor update time samples represented in the Total Processor Power Accumulator.
<b>0x20</b>	<b>Total Processor Power Accumulator</b> – 8 bytes. Accumulation of total processor power readings
<b>0x28</b>	<b>Current Processor Vdd Power</b> – 2 bytes. Most recent processor Vdd power reading in watts
<b>0x2A</b>	<b>Processor Vdd Power Update Tag</b> – 4 bytes. Count of number of processor update time samples represented in the Processor Vdd Power Accumulator.
<b>0x2E</b>	<b>Processor Vdd Power Accumulator</b> – 8 bytes. Accumulation of processor Vdd power readings
<b>0x36</b>	<b>Current Processor Vdn Power</b> – 2 bytes. Most recent processor Vdn power reading in watts
<b>0x38</b>	<b>Processor Vdn Power Update Tag</b> – 4 bytes. Count of number of processor update time samples represented in the Processor Vdn Power Accumulator.
<b>0x3C</b>	<b>Processor Vdn Power Accumulator</b> – 8 bytes. Accumulation of processor Vdn power readings

---

## 6.4 Power Caps (“CAPS”)

This is only available from the master OCC poll response. If there is no SPI attached chip for system power reading, power capping is not supported and all 0's will be returned.

Sensor Eye Catcher = “CAPS”

Sensor Version = 0x03

Sensor Length = 0x0F

Format for power caps, this is system based and not repeated. Number of sensors in poll response will always be 1 for power caps:

Offset	
0x00	<b>Current Power Cap</b> – 2 bytes. In 1W units the current (output) power cap value that is in effect that the OCC is monitoring power to. This will be equal to one of the following: <ul style="list-style-type: none"><li>• N Power Cap</li><li>• Maximum System Power Cap</li><li>• User Power Limit</li></ul>
0x02	<b>Current System Power Reading</b> – 2 bytes. In 1W units the current system (output) power. This is the value being compared to the current power cap to decide if any actions are needed to maintain the current power cap.
0x04	<b>N Power Cap</b> – 2 bytes. In 1W units the (output) power cap limit when there is not redundant power.
0x06	<b>Maximum System Power Cap</b> – 2 bytes. In 1W units the maximum (output) power cap that may be set. This is the system maximum power limit with redundant power.
0x08	<b>Hard Minimum Power Cap</b> – 2 bytes. In 1W units the minimum (output) power cap that may be set and held by the OCC.
0x0A	<b>Soft Minimum Power Cap</b> – 2 bytes. In 1W units the minimum (output) power cap that may be set that. A power cap limit set below hard minimum is called a soft power cap and is not guaranteed to be held under all conditions by the OCC.
0x0C	<b>User Power Limit</b> – 2 bytes. In 1W units the (output) power cap specified by a user. NOTES: <ul style="list-style-type: none"><li>• This will be 0x0000 if no user set power limit or the user set power limit is not active</li><li>• If user is setting the power limit as input power, the BMC must do conversion between input/output power using the power supply efficiency factor from the Configuration file.</li></ul>
0x0E	<b>User Power Limit Source</b> – 1 byte. Indicates how the power limit was set <b>0x01</b> = User power limit was set out of band (cmd from (H)TMGT or BMC) <b>0x02</b> = User power limit was set in band (cmd from OPAL) NOTE: an OCC reset or system power on will maintain a power limit set in band, however the command to resend to OCC after a reset will come from (H)TMGT and the source will reflect that the power limit was now set out of band

## 6.5 Extended OCC Data (“EXTN”)

This is available in master and slave OCC poll responses. This is to give a way to return OCC data/sensors that do not fit under any previously defined sensor data section.

Sensor Eye Catcher = “EXTN”

Sensor Version = 0x01

Sensor Length = 0x0C

Format for one sensor and repeated for Number of Sensors:

Offset	
0x00	<b>Name</b> – 4 bytes to identify the data. This can be ASCII or sensor ID
0x04	<b>Flags</b> – 1 byte bit defined flags for any special processing to be done by the BMC: <b>Bit 0 (msb) – Sensor ID.</b> ‘1’ = Name is a sensor ID to be exported via normal sensor processing else Name is an ASCII string to be exported TBD. <b>Bits 1:7 – Reserved</b>
0x05	<b>Reserved</b> – 1 byte reserved = 0x00
0x06	<b>Data</b> – 6 bytes data. The format of the data is dependent on what this data represents defined by the Name.

### 6.5.1 Extended OCC Sensors List

OCC data returned in the “EXTN” sensor data section:

Name	Flags	Data												
“FMIN”	0x00	<b>Minimum Frequency</b> – Minimum frequency system can run at:												
		<table><tr><td>Byte 1</td><td>Byte 2</td><td>Byte 3</td><td>Byte 4</td><td>Byte 5</td><td>Byte 6</td></tr><tr><td>Pstate</td><td colspan="2">Frequency in MHz</td><td colspan="3">Reserved = 0</td></tr></table>	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Pstate	Frequency in MHz		Reserved = 0		
		Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6							
Pstate	Frequency in MHz		Reserved = 0											
“FDIS”	0x00	<b>Modes Disabled Frequency</b> – frequency for the system when all power management modes are disabled:												
		<table><tr><td>Byte 1</td><td>Byte 2</td><td>Byte 3</td><td>Byte 4</td><td>Byte 5</td><td>Byte 6</td></tr><tr><td>Pstate</td><td colspan="2">Frequency in MHz</td><td colspan="3">Reserved = 0</td></tr></table>	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Pstate	Frequency in MHz		Reserved = 0		
		Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6							
Pstate	Frequency in MHz		Reserved = 0											
“FBAS”	0x00	<b>WOF Base Frequency</b> – WOF Base frequency for the system:												
		<table><tr><td>Byte 1</td><td>Byte 2</td><td>Byte 3</td><td>Byte 4</td><td>Byte 5</td><td>Byte 6</td></tr><tr><td>Pstate</td><td colspan="2">Frequency in MHz</td><td colspan="3">Reserved = 0</td></tr></table>	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Pstate	Frequency in MHz		Reserved = 0		
		Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6							
Pstate	Frequency in MHz		Reserved = 0											
“FUT”	0x00	<b>UltraTurbo Frequency</b> – Ultra Turbo frequency for the system. Will be all 0’s if ultra turbo (WOF) is not supported:												
		<table><tr><td>Byte 1</td><td>Byte 2</td><td>Byte 3</td><td>Byte 4</td><td>Byte 5</td><td>Byte 6</td></tr><tr><td>Pstate</td><td colspan="2">Frequency in MHz</td><td colspan="3">Reserved = 0</td></tr></table>	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Pstate	Frequency in MHz		Reserved = 0		
		Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6							
Pstate	Frequency in MHz		Reserved = 0											
”FMAX”	0x00	<b>Chip Maximum Frequency</b> – Maximum frequency this chip can achieve. Chips in a system can have a different Fmax. Will be all 0’s if not												

		supported:																					
		<table><tr><td>Byte 1</td><td>Byte 2</td><td>Byte 3</td><td>Byte 4</td><td>Byte 5</td><td>Byte 6</td></tr><tr><td>Pstate</td><td colspan="2">Frequency in MHz</td><td colspan="3">Reserved = 0</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Pstate	Frequency in MHz		Reserved = 0											
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6																		
Pstate	Frequency in MHz		Reserved = 0																				
“CLIP”	0x00	<p><b>Clip</b> – Gives Pstate clip information. Current max Pstate is the current allowable maximum Pstate. Count is a counter of # of ticks the maximum Pstate has been reduced (0 if not currently reduced), no rollover supported. Clip Reason will be 0x00000000 if the OCC never had to limit the maximum Pstate else it will be a history of reason(s) max Pstate was ever clipped. If multiple reasons are set it is not possible to know from the count how many times each reason caused clipping.</p> <table><tr><td>Byte 1</td><td>Byte 2</td><td>Byte 3</td><td>Byte 4</td><td>Byte 5</td><td>Byte 6</td></tr><tr><td>Current max Pstate</td><td>Count</td><td colspan="4">Clip Reason</td></tr></table>	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Current max Pstate	Count	Clip Reason												
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6																		
Current max Pstate	Count	Clip Reason																					
“WOFC”	0x00	<p><b>WOF Clip</b> – Gives WOF information from the PGPE</p> <table><tr><td></td><td>Byte 1</td><td>Byte 2</td><td>Byte 3</td><td>Byte 4</td><td>Byte 5</td><td>Byte 6</td></tr><tr><td><b>WOF Enabled</b></td><td>F Clip (Pstate)</td><td>0x00</td><td></td><td colspan="2">Vratio</td><td>0x00</td></tr><tr><td><b>WOF Disabled</b></td><td>0xFF (not a valid Pstate)</td><td>0x00</td><td colspan="4">WOF Disabled Reason (each bit represents a reason WOF is disabled)</td></tr></table>		Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	<b>WOF Enabled</b>	F Clip (Pstate)	0x00		Vratio		0x00	<b>WOF Disabled</b>	0xFF (not a valid Pstate)	0x00	WOF Disabled Reason (each bit represents a reason WOF is disabled)			
	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6																	
<b>WOF Enabled</b>	F Clip (Pstate)	0x00		Vratio		0x00																	
<b>WOF Disabled</b>	0xFF (not a valid Pstate)	0x00	WOF Disabled Reason (each bit represents a reason WOF is disabled)																				
“ERRH”	0x00	<p><b>Error History</b> – Optional field that will only exist if there is at least 1 non-zero history count, each non-zero counter will be added as 1 byte enum, 1 byte count for that enum. There is no support for rollover.</p> <table><tr><td>Byte 1</td><td>Byte 2</td><td>Byte 3</td><td>Byte 4</td><td>Byte 5</td><td>Byte 6</td></tr><tr><td>Enum</td><td>Count</td><td>Enum</td><td>Count</td><td>Enum</td><td>Count</td></tr></table>	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Enum	Count	Enum	Count	Enum	Count									
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6																		
Enum	Count	Enum	Count	Enum	Count																		

---

## 7 OCC Inter-Chip Communication

This data is not used for anything and is just put in place to test inter-chip communication for possible future needs.

One 8 byte frame sent every tick:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Fmax Pstate	Curr Reques ted Pstate	Pstate Reason		Chip Temp			

**Byte 1: Fmax Pstate** – This is the highest frequency Pstate this chip may ever reach

**Byte 2: Curr Requested Pstate** – This is the currently requested Pstate for this chip from the OCC voting box. This Pstate request does not include the WOF clip done by PGPE.

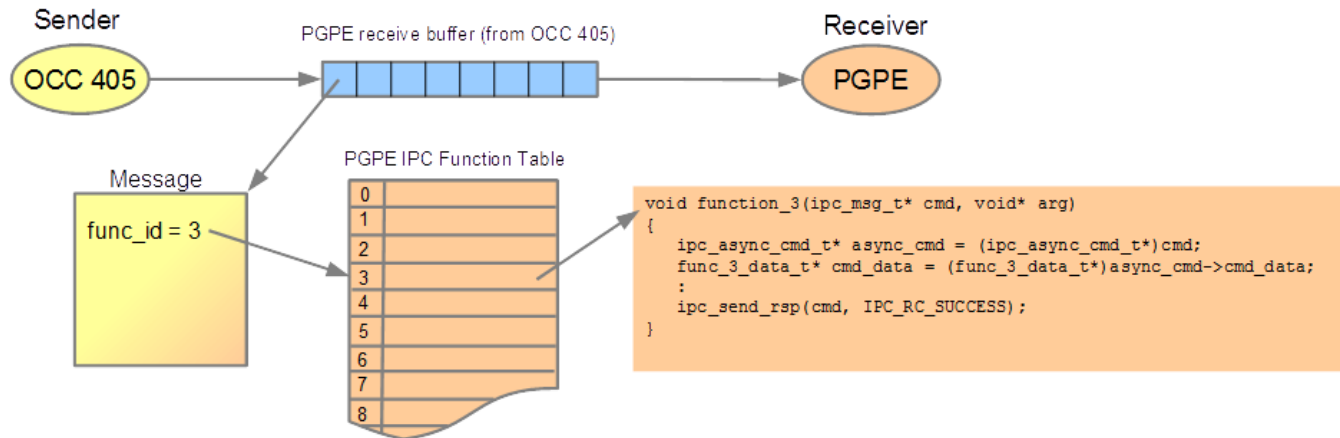
**Byte 3: Pstate Reason** – This is a bit mask reason for the currently requested Pstate

**Byte 5: Chip Temp** – This is the hottest of the weighted DTS core temperatures. i.e. sensor THRM this is the same sensor used for determining if DVFS is needed due to thermal

## 8 OCC to PGPE Communication

### 8.1 IPC Commands OCC to PGPE

Inter-Processor Communication (IPC) is used to send Pstate operations from the OCC 405 to the PGPE. IPC uses circular buffers in the SRAM tank. A function ID is used as a look up in the PGPE IPC function table.



<i>IPC Function ID</i>	<i>Operation</i>
1	Start/Stop Pstate Protocol
2	Pstate Clip Update
3	Set PMCR
4	WOF Control
5	WOF VRT (Voltage Ratio Table)

### 8.1.1 Start/Stop Pstate Protocol

This command is used to start or suspend Pstate protocol.

- Default is Pstate protocol stopped. The OCC must send a start after every PM complex reset to enable Pstates
- Stop is used when OCC is about to be reset. Stop will NOT be sent prior to changing PMCR owner the PGPE will handle anything that is needed prior to switching the owner bit
- On Stop action the PGPE completes outstanding Pstate change then sends IPC response. While stopped: PGPE ignores any new Pstate requests. WOF is also stopped. PGPE still does beacon to OCC and heartbeat to PPM to prevent safe mode.
- On Start action the PGPE writes the LMCR based on the owner passed in, enables Pstate protocol and then sends IPC response.

<b><i>Input Data</i></b>	<b>Action</b> – Defines what action to take on the Pstate Protocol: <b>Start</b> <b>Suspend</b> <b>PMCR Owner</b> – Used on start action only. Defines who owns setting Pstates: <b>HOST</b> <b>OCC</b> <b>CHAR</b> – special mode for test to write PMCR externally
<b><i>Output Data</i></b>	<b>Return Code</b> Any non-successful return code will result in a PM complex reset

### 8.1.2 Pstate Clip Update

This command is used to set the minimum and maximum Pstate (frequency range).

- Pmax default is Psafe, Pmin default is Pstate of Fmin
- Pmax in this command does not include WOF. The PGPE calculates the max clip for WOF and picks the lower frequency Pstate between WOF and what is sent in this command to set as the actual Pmax
- Pmax with OPAL: vote based on thermal and power capping needs
- Pmax with PHYP: Set to be wide open Fmin to Fmax (OCC sends Set PMCR to get desired Pstate)

<b><i>Input Data</i></b>	<b>Pmin</b> – minimum frequency → highest Pstate <b>Pmax</b> – maximum frequency → lowest Pstate
<b><i>Output Data</i></b>	<b>Return Code</b> Any non-successful return code will result in a PM complex reset



### 8.1.3 Set PMCR

This command is used to send the PGPE a PMCR write request. PGPE to determine actual Pstate from clips.

- OCC may only send this command if the last “Start/Stop Pstate Protocol” command was a start with OCC owner.
- The OCC will never write the PMCR directly. This command will always be used to set the requested Pstate.

<b><i>Input Data</i></b>	<b>PMCR</b>
<b><i>Output Data</i></b>	<b>Return Code:</b> <b>NOT_PMCR_OWNER</b> – PGPE ignored command due to Pstate Protocol not set for OCC owner. OCC to send a Start with OCC owner  All other non-successful return codes will result in a PM complex reset

#### 8.1.4 WOF Control

This command is used to control WOF enablement.

- Default is WOF off. The OCC must send a command to turn WOF on
- Pstate Protocol MUST be started prior to turning WOF on

<b><i>Input Data</i></b>	<b>Action</b> – Defines if WOF should be on or off: WOF_ON WOF_OFF
<b><i>Output Data</i></b>	<b>Return Code</b> <ul style="list-style-type: none"><li>• Any non-successful RC will result in a PM complex reset</li></ul>

### 8.1.5 WOF VRT

This command is used to send the PGPE a new Voltage Ratio Table for WOF.

- Pstate Protocol MUST be started in order to send WOF VRT
- One WOF VRT will be sent when WOF is OFF to allow for WOF to be turned ON
- While WOF is ON the OCC will be sending the VRT that was calculated on the previous tick at the beginning of each tick even if the VRT did not change. This is a PGPE requirement to get a notification from the OCC on a consistent time base.
- OCC will have 2 SRAM buffers for a VRT, one that the PGPE is actively using as pointed to from the last successful WOF VRT response and one that the OCC can actively update with a new VRT and will be sent in the next WOF VRT command. The OCC cannot switch to the SRAM buffer that the PGPE is using until the PGPE has responded success.

<b><i>Input Data</i></b>	<b>VRT Pointer</b> – SRAM address of VRT to use <b>Vdd ceff ratio</b> – Vdd ceff ratio used to determine VRT <b>Vcs ceff ratio</b> – Vcs ceff ratio used to determine VRT
<b><i>Output Data</i></b>	<b>Return Code</b> <ul style="list-style-type: none"><li>• Success. New VRT in place OCC can take back memory from previous VRT</li><li>• Any other non-successful return code should result in a PM complex reset</li></ul>

---

## 9 Power Management

---

### 9.1 BMC Power and Thermal Management Settings

#### 9.1.1 Fan Control

Pointers to fan control settings used by the BMC can be found at <https://github.ibm.com/openbmc/openbmc/wiki/System-Engineers-Reference>

#### 9.1.2 Entity Manager Consumed by BMC (Mode, IPS)

The following are defaults for user settable settings. These are defined in entity manager and used by the BMC

	Description
IPS_ENABLE	Default value for Idle Power Saver enablement
ENTER_IPS_TIME_S	Default Idle Power Saver delay time between 10 and 600s to enter IPS when IPS is enabled
ENTER_IPS_UTIL_PERCENT	Default Idle Power Saver utilization threshold between 1% and 95% to enter IPS when IPS is enabled
EXIT_IPS_TIME_S	Default Idle Power Saver delay time between 10 and 600s to exit IPS
EXIT_IPS_UTIL_PERCENT	Default Idle Power Saver utilization threshold between 5% and 95% to exit IPS.
DEFAULT_PWR_PERF_MODE	Default Power and Performance mode 1 = Modes Disabled (aka Nominal) 5 = Static Power Saver <i>10 = Dynamic Performance (not supported GA1/GA2)</i> 12 = Maximum Performance

#### 9.1.3 MRW Consumed by HTMGT (Power cap, thermal limits...)

The following power and thermal configuration settings are defined in the system MRW and used by HTMGT

XML Attribute	Description
N_PLUS_ONE_BULK_POWER_LIMIT_WATTS	System maximum power cap in output watts when QPD line is not asserted. This value must guarantee WOF base.
N_PLUS_ONE_MAX_MEM_POWER_WATTS	The amount of N+1 Bulk Power to allocate to memory, this value will be used to calculate memory throttles to cap memory to this power. This value must be the left over power from N+1 Bulk Power after allocating power for fixed resources and processor power to guarantee WOF base. NOTE: This value is first reduced by Regulator Efficiency Factor to account for regulator loss before running the hw procedure.
N_BULK_POWER_LIMIT_WATTS	System maximum power cap in output watts when QPD line is asserted. This value must guarantee WOF base.
N_MAX_MEM_POWER_WATTS	The amount of N Bulk Power to allocate to memory, this value will be used to calculate memory throttles to cap memory to this power. This value must be the left over power from N Bulk Power after allocating power for fixed resources and processor power to guarantee WOF base. NOTE: This value is first reduced by Regulator Efficiency Factor before running the procedure to account for regulator loss.
REGULATOR_EFFICIENCY_FACTOR	Percentage to lower N+1 Maximum Memory Power and N Maximum

	Memory Power to account for regulator loss prior to calling procedure to calculate memory throttles. NOTE: The procedure calculating memory throttles do not account for regulator loss.
MIN_POWER_CAP_WATTS	Lowest output power in watts that a user may set, and the OCC can guarantee to hold via processor DVFS under all conditions. Aka Hard minimum power cap.
SOFT_MIN_POWER_CAP_WATTS	Minimum soft power cap, this is the lowest output power in watts that a user may set. A power cap set below the hard minimum (MIN_POWER_CAP_WATTS) is called a soft power cap and is not guaranteed under all conditions.
MIN_MEM_UTILIZATION_THROTTLING	The lowest utilization allowed that the OCC can throttle memory due to a memory over temp condition.
CORE_WEIGHT_TENTHS	Weight factor (1 = 0.1) for each core DTS to calculate a core temperature and eventual processor temperature that is used for DVFS and fan control.
QUAD_WEIGHT_TENTHS	Weight factor (1 = 0.1) for each quad (racetrack) DTS to calculate a core temperature. A value of 0 means not to include racetrack DTS in core temperature.
L3_WEIGHT_TENTHS	Weight factor (1 = 0.1) for L3 DTS to calculate a core temperature. A value of 0 means not to include L3 DTS in core temperature.
PROC_DVFS_TEMP_DELTA_C	Per Chip. Degrees C to change (+/-) the processor chip VPD DVFS temperature to invoke DVFS (clip max Pstate). Default is 0 (no change to VPD limit) NOTE: If we fail to read DVFS limit from VPD system will go to safe mode.
PROC_ERROR_TEMP_DELTA_C	Per Chip. Degrees C to add to the processor chip VPD DVFS temperature that a processor chip over temperature error will be logged calling out the processor.
PROC_READ_TIMEOUT_SEC	Maximum time in seconds allowed without having a new processor temperature before DVFS will occur
PROC_IO_DVFS_TEMP_DELTA_C	Per Chip. Degrees C to change (+/-) the processor IO chip VPD DVFS temperature to invoke DVFS (clip max Pstate). Default is 0 (no change to VPD limit) NOTE: Current plan is no support to DVFS based on IO temp
PROC_IO_ERROR_TEMP_DELTA_C	Per Chip. Degrees C to add to the processor IO chip VPD DVFS temperature that a over temperature error will be logged calling out the processor.
PROC_IO_READ_TIMEOUT_SEC	Maximum time in seconds allowed without having a new Proc IO temperature before DVFS (if supported) will occur
VRM_VDD_DVFS_TEMP_DEG_C	VRM Vdd Temperature in degrees C to invoke DVFS (clip max Pstate)
VRM_VDD_ERROR_TEMP_DEG_C	VRM Vdd Temperature in degrees C that an overtemp error will be logged
VRM_VDD_READ_TIMEOUT_SEC	Maximum time in seconds allowed without having a new VRM Vdd temperature before DVFS will occur
MEMCTRL_THROTTLE_TEMP_DEG_C	MC Temperature to invoke memory throttling
MEMCTRL_ERROR_TEMP_DEG_C	MC Temperature in degrees C that a MC overtemp error will be logged calling out the MC
MEMCTRL_READ_TIMEOUT_SEC	Maximum time in seconds allowed without having a new MC temperature before memory throttling will occur
DIMM_THROTTLE_TEMP_DEG_C	DIMM Temperature to invoke memory throttling
DIMM_ERROR_TEMP_DEG_C	DIMM Temperature in degrees C that a DIMM overtemp error will be logged calling out the DIMM
DIMM_READ_TIMEOUT_SEC	Maximum time in seconds allowed without having a new DIMM temperature before memory throttling will occur
MC_EXT_THROTTLE_TEMP_DEG_C	External MC Temperature to invoke memory throttling
MC_EXT_ERROR_TEMP_DEG_C	External MC Temperature in degrees C that a overtemp error will be logged calling out the MC
MC_EXT_READ_TIMEOUT_SEC	Maximum time in seconds allowed without having a new external MC temperature before memory throttling will occur
PMIC_THROTTLE_TEMP_DEG_C	PMIC Temperature to invoke memory throttling
PMIC_ERROR_TEMP_DEG_C	PMIC Temperature in degrees C that an overtemp error will be logged calling out the MC

PMIC_READ_TIMEOUT_SEC	Maximum time in seconds allowed without having a new PMIC temperature before memory throttling will occur
MC_DRAM_THROTTLE_TEMP_DEG_C	Sensor covering both External MC and DRAM Temperature to invoke memory throttling
MC_DRAM_ERROR_TEMP_DEG_C	Sensor covering both External MC and DRAM Temperature in degrees C that a overtemp error will be logged calling out the MC
MC_DRAM_READ_TIMEOUT_SEC	Maximum time in seconds allowed without having a new temperature for the sensor covering both external MC and DRAM before memory throttling will occur
PBAX_CHIPID and PBAX_GROUPID	For PBAX (OCC-OCC communication) group ID should be 0 for all processors and chipID 0 for 1 <sup>st</sup> processor and 1 for 2 <sup>nd</sup> processor.
APSS Configuration	All attributes assigning the 16 ADC Channels and GPIOs must be defined.

---

## 9.2 Power Management Settings (FSP)

FSP systems support a power management def file to give settings that are software control i.e. thermal thresholds, fan floor tables, and power cap information. Requirements for the power management def file can be found here:

<https://mcdoc.boeblingen.de.ibm.com/out/out.ViewDocument.php?documentid=3139>

The following settings that are hardware control will come from the MRW:

- APSS Channel and GPIO assignment
- Memory configuration
- PBAX Chip IDs

---

## 9.3 PowerVM System Power and Performance Modes

System Power and Performance modes are only supported with PowerVM. See appendix C for a list of Power and Performance modes.

---

## 9.4 User Power Capping

All power cap values sent to the OCC must be output power. All power readings and power cap values from the OCC are output power. DCML is using input power the BMC must do all conversions between output and input using the power supply efficiency from the configuration file. The BMC must support the capability for HTMGT to read the “power limit” and “power limit activation” that was last set by the user for OCC initialization at boot and OCC reset.

### 1.1.8 Reading Current User Power Limit

The power limit set and if active should be persistent across AC cycles and will be stored by the BMC. The OCC poll response “CAPS” sensor data section will contain the current active set user power limit.

### 1.1.9 Setting Power Limit or Activate/Deactivate Power Limit

When setting an input power limit, the BMC must first convert the power limit to output using the power supply efficiency from the Configuration file.

#### **BMC Requirements to Determining if Power Limit is within bounds**

- Prior to any communication with the OCC, the BMC will have a default min/max power limit from the configuration file that must cover all power configuration settings.
- On the first poll with the OCC the BMC must update the min/max power limit that the master OCC provides in the “CAPS” sensor section of poll response. In addition, in the case that the current power limit set is now out of bounds from the new min/max power limit being reported from the OCC the BMC must clip the current power limit to be min or max.

#### **BMC Receives Command to Set or Active/Deactivate Power Limit**

1. BMC receives set power limit or Activate/Deactivate power limit command; BMC will decide if the power limit is within bounds and reject if it is not.
2. The BMC stores the power limit or activate/deactivate into persistent memory.
3. If the “OCC Active” sensor is TRUE then the BMC sends the master OCC the “Set User Power Cap” command with the appropriate data else no command is sent and HTMGT will send as part of bringing the OCCs active.

#### **Sending OCC Power Limit after System Boot or OCC Reset**

1. Whenever HTMGT is bringing the OCCs active (i.e. system boot, after an OCC reset...) HTMGT will call HB interface to read the “power limit” and “power limit activation” from the BMC
2. If there is an active power limit HTMGT will verify that it is within the min/max for the power/thermal configuration setting. If the active power limit falls out of bounds HTMGT will lower it to the max or raise it to the minimum.
3. HTMGT sends the master OCC a “Set User Power Cap” command with the appropriate data prior to sending state change to active.
4. On first poll of the master OCC the BMC must update the minimum/maximum range it uses for determining a power limit is within bounds from the power limit data the master OCC provides in the “CAPS” sensor section of poll response. In addition, in the case that the current power limit set is now out of bounds from the new min/max power limit being reported from the OCC the BMC must clip the current power limit to be min or max.

---

## 9.5 Idle Power Saver

Idle Power Saver (IPS) is only supported on single node PowerVM systems. Multi-node systems will not support IPS due to these reasons:

- It is not required for energy star
- It would not be enabled by default, so not many customers will enable it
- Entering IPS on such a large system would be very unlikely

Default IPS enter and exit criteria are defined in the power management def file (FSP) or in

entity manager (BMC). The master OCC gets a 3 second average of all cores utilizations from all slaves every 500usec. When IPS is enabled the master OCC will determine when the IPS enter or exit criteria is met and inform the slave OCCs, each OCC will then change frequency and memory power control settings.

Enter and exit criteria must be made available for a user to override in the field. Whenever there is an IPS parameter change the FSP/BMC must verify the parameter is in the allowed range and if it is update NVRAM with the new user set values and send an “Idle Power Saver Settings” config data command to the master OCC only. If any IPS parameter is changed from its default value the IPS fan floor will not be used. This is needed to avoid thermal issues. The default parameters were assumed when determining the IPS fan floor.

The master OCC will also tell the FSP/BMC if IPS is “active” or not. When IPS is enabled the master OCC should have the IPS enabled bit in the Current Idle Power Save status byte of the poll response ‘1’ and remain ‘1’ until IPS is disabled. While enabled the master OCC will be setting the IPS active bit in that byte to reflect the actual state of IPS. FSP/BMC will be processing these bits (from master OCC poll response only) to determine what to do based on this table:

<b><i>Idle Power Save Enable from User</i></b>	<b><i>Master OCC Poll response Current Idle Power Saver Status Byte</i></b>		<b><i>Valid?</i></b>	<b><i>FSP/BMC Actions</i></b>
	<b><i>bit 6: Idle Power Saver Active</i></b>	<b><i>bit 7: Idle Power Saver Enabled</i></b>		
0	0	0	Y	No action
0	0	1	N	Re-send OCC command to disable IPS, log error if this persists.
0	1	0	N	Log error if this persists across multiple polls
0	1	1	N	Re-send OCC command to disable IPS, log error if this persists.
1	0	0	N	Re-send OCC command to enable IPS, log error if this persists.
1	0	1	Y	On 1->0 transition of Active bit: <ul style="list-style-type: none"> <li>➤ FSP/BMC will check if fan floor needs to change due to IPS no longer active</li> <li>➤ No message to PHYP, OCC will inform PHYP of the change in the OCC shared memory</li> </ul>



				interface
1	1	0	N	Re-send OCC command to enable IPS, log error if this persists.
1	1	1	Y	<p>On 0-&gt;1 transition of Active bit:</p> <ul style="list-style-type: none"> <li>➤ FSP/BMC will check if fan floor needs to be updated due to IPS active. This can only be done if all IPS parameters are set to their default.</li> <li>➤ No message to PHYP, OCC will inform PHYP of the change in the OCC shared memory interface</li> </ul>

While IPS is enabled a user may at any time change the enter and/or exit parameters, these will be sent to the master OCC and take effect immediately while IPS is enabled.

---

## 10 Manufacturing Impacts

---

### 10.1 MFG Test Commands

See the [Manufacturing Test command](#) for details for direct OCC manufacturing test commands.

#### 10.1.1 Processor Auto-Slew with OPAL

On start of auto slew, first need to set the PMCR mode register for OCC owner to lock out OPAL from changing Pstates and the OCC informs OPAL via shared memory interface that they are no longer in control. Need to revisit previous decision that changing PMCR owner requires a reset, any issues with the following:







##### **Start Auto-Slew Process with OPAL**

(H)TMGT receives command and tells the OCC to start auto slew:

1. OCC updates OCC-OPAL shared memory to inform OPAL they are not in control of Pstates
2. OCC sends PGPE IPC command WOF Control to turn WOF off
3. OCC sends PGPE IPC command Set Pstate Clips for auto slew min/max (need to make sure clips are open enough to cover full auto slew range)
4. OCC sends PGPE IPC command Control Pstate Protocol to suspend Pstate protocol
5. OCC sends PGPE IPC command Control Pstate Protocol to start Pstate protocol for OCC owner
6. OCC starts auto slew process by sending PGPE IPC Command Write PMCR with Requested Pstate and then sends response to (H)TMGT

##### **Stop Auto-Slew Process with OPAL**

(H)TMGT receives command and tells the OCC to stop auto slew:

-  OCC sends PGPE IPC command Set Pstate Clips to set min/max clips back
-  OCC sends PGPE IPC command Write PMCR to set requested Pstate to Pmin (min clip)
-  OCC sends PGPE IPC command Control Pstate Protocol to suspend Pstate protocol
-  OCC sends PGPE IPC command Control Pstate Protocol to start Pstate protocol for OPAL owner
-  If WOF is enabled OCC sends PGPE IPC command WOF Control to turn WOF on
-  OCC updates OCC-OPAL shared memory to inform OPAL they control Pstates and then sends response to (H)TMGT

---

### 10.2 Pstate Table Bias

Requires a power management complex reset after updating the attributes to rebuild the

---

## 10.3 Enable/Disable OCC Control

For manufacturing testing to prevent the OCC from changing Pstate or clipping the max p State due to thermal or power the OCC should be disabled by putting the OCC into Observation state. While in observation state all OCC, sensors are still updated with no actuation due to thermal or power. BMC and FSP communication is allowed to send the poll command to read the sensors from OCC and should still be setting fans based on the temperatures. The PGPE is also in an observation state defined as WOF off and Pstate protocol suspended. When finished with testing the OCC must be put back into Active state (full actuation).

### 10.3.1 Observation State (Disable OCC) Change Process

(H)TMGT receives command and tells the OCC to go to observation state:

1. OCC tells PGPE to set Pstate clip to WOF base
2. OCC tells PGPE to turn WOF off – PGPE only uses clip value, stops calculating WOF clip, unblock stop requests, then respond to WOF off request when at/below clip.
3. OCC sends PGPE IPC command Control Pstate Protocol to suspend Pstate protocol
4. OCC continues to monitor power and thermals but takes no action

### 10.3.2 Characterization State Change Process

(H)TMGT receives command and tells the OCC to go to characterization state:

1. OCC tells PGPE to set Pstate clips to be wide open to allow full frequency range
2. OCC tells PGPE to turn WOF off
3. OCC sends PGPE IPC command Control Pstate Protocol to enable Pstate protocol with characterization as PMCR owner
4. OCC continues to monitor power and thermals but takes no action (same as observation state)

### 10.3.3 Active State (Enable OCC) Change Process

After bias return the OCC to full function active state.

(H)TMGT receives command and tells the OCC to go to active state:

1. OCC tells PGPE to start Pstate protocol
2. If WOF is enabled the OCC tells PGPE to turn WOF on
3. OCC tells PGPE to set Pstate clip based on running control loops

---

## 10.4 External Voltage and Frequency Bias

All frequency and voltage biasing require the OCC complex to first be put in observation state to prevent the OCC from changing the voltage/frequency during bias. There is an OP tool command to change the OCC state. Frequency bias uses OP tool setclockspeed after the

OCC is in observation state.





### 10.4.1 Writing Voltage

Process for writing voltages:

1. Put the OCC complex in observation state
2. Set voltages using one of the following:
  - A. OP Tool commands. This uses I2C interface.
  - B. I2C commands directly
  - C. AVSbus Bridge A. NOTE: In observation state Pstate protocol is turned off freeing up PGPE owned bridge A, bridge B is still actively used by the OCC for monitoring.
3. Put the OCC complex back in active state

### 1.1.10 Reading Voltage

Read voltages using one of the following:

-  OP Tool commands. This uses I2C interface.
-  I2C commands directly
-  If OCC in observation state or in reset: AVSbus bridge A. In observation state PGPE is not using bridge A
-  If OCC is in active state: Read Vdd and Vdn sensors from OCC directly. Not allowed to use AVSbus, both AVSbus bridges are actively used by the PGPE and OCC.

---

## 11 OCC Main Memory Layout

Four sets of PowerBus Real Base Address Registers (PBABAR0..3) mapping the OCI address to the PowerBus space. These registers are setup by HB and define the PowerBus range of system memory that can be accessed by PBA. OCC must set up slaves for BAR1 and BAR2. The CME sets up slaves for BAR0. OCC will never change these.

BAR0: HOMER Image – how OCC access the memory that contains images, config data and interfaces to HTMGT and OPAL.

BAR1: Memory – this is how GPE1 will access the memory buffers to read memory temperatures

BAR2: OCC Common Image – Includes OCC-OCC communication and sensor data

BAR3: SBE – never used by OCC

---

## 11.1 HOMER

The OCC firmware is allocated the first 1MB of HOMER. The remaining HOMER image is defined by the chip team for PGPE, XGPE....

### OCC HOMER Region (per chip) 1MB

Start (offset from base HOMER address)	Size	Description
0x00000000	896kB	OCC Image: OCC Bootloader (4kB) 405 (768kB) GPE0 (60kB) GPE1 (64kB)
0x000E0000	4kB	HTMGT to OCC Command Buffer
0x000E1000	4kB	HTMGT to OCC Response Buffer
0x000E2000	32kB	OCC-OPAL Shared Memory Interface
0x000EA000	40kB	Reserved
0x000F4000	8kB	OCC Host Data Area. In order: <ul style="list-style-type: none"><li>• 4B Version = 0x000000A0</li><li>• 4B OCC Timebase Frequency MHz. In P9 this was equal to nest in P10 this is equal to PAU. (The OCC will /4 for SSX. GPEs use OCB Timebase Register (OTBR) this value is /64 for GPEs)</li><li>• 4B OCC Interrupt Type<ul style="list-style-type: none"><li>• 0x00000000 = FSP. No Interrupt from OCC to host</li><li>• 0x00000001 = BMC. Interrupt via PSIHCB complex</li></ul></li><li>• 4B Secure Memory Facility (SMF)<ul style="list-style-type: none"><li>• 0x00000000 = Disabled</li><li>• 0x00000001 = Enabled</li></ul></li><li>• 8,176B Reserved</li></ul>
0x000F6000	40kB	Reserved

---

## 11.2 OCC Common Image

OCC Common image is 8MB per physical drawer. PHYP/HB will zero out memory in OCC Common Area when reloading/resetting all OCCs.

### OCC Common Image (per drawer) 8MB

Start (offset from base HOMER address)	Size	Description
0x00000000	256kB	OCC-OCC Messaging: Master->Slave (256 bytes * 8 chips max = 0x0800) <ul style="list-style-type: none"><li>• Ping: 0x00000000 - 0x000007FF</li><li>• Pong: 0x00000800 - 0x00000FFF</li></ul> Slave->Master (1024 bytes * 8 chips max = 0x2000) <ul style="list-style-type: none"><li>• Ping: 0x00001000 - 0x00002FFF</li><li>• Pong: 0x00003000 - 0x00004FFF</li></ul> 0x00005000 - 0x0003FFFF Reserved
0x00040000	256kB	Reserved
0x00080000	5MB	AMESTER Traces
0x00580000	1.5MB	Sensor Data. NOTE: Block copy has 4K limit, try to keep size of sensor readings being updated to under 4K to have just 1 block copy every 100ms. The static data associated with sensors is written once and can be much larger.
0x00800000	1MB	Reserved

## 12 OCC-OPAL/PHYP Interface

### 12.1 OCC-OPAL/PHYP Shared Memory Interface

Starting address is 0x000E2000 from the base HOMER address, the offset below is from the starting address. Maximum size is 32kB.

The data is separated into first “static” typically cannot change at runtime (however lab tools may allow for a change i.e. Pstate table bias) and then “dynamic” data that can change at runtime.

Offset	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x0000	Valid (0x01)	Static Major Version (0xA0)	OCC Role	Min Freq Pstate	Fixed Freq Pstate	WOF base Pstate	Ultra Turbo Pstate	Fmax Pstate
0x0008	Static Minor Version (0x01)	Bottom Throttle Space Pstate						
0x0010	Reserved							
0x0018	Pstate# = 0	Valid (0 or 1)	Reserved		Freq in kHz			
0x0020	Pstate # = 1	Valid (0 or 1)	Reserved		Freq in kHz			
:	:	:	:	:				
0x0810	Up to 255 Pstate							
0x0818	Param table version (0x10)	# Param table entries	Reserved for future Parameter table header info					
0x0820	Parameter ID		Param Flags	Parameter Data				
0x0828	27 byte Parameter text description							
0x0830								
0x0838								
0x0840	Parameter ID		Param Flags	Parameter Data		27 byte Parameter text description		
:	:							
0x0B40	Repeat for up to 25 Parameter table entries. See “Parameter Table Definition” section							



Offset	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x0B48 :	56B Reserved for future “static” data							
0x0B80	OCC State	Dynamic Major Version	Dynami c Minor Version	GPUs Present	WOF State	Proc Throttle Status	Memory Throttle Status	Quick Power Drop
0x0B88	Power Shifting Ratio	Power Cap Type	Hard Min Power Cap		Max Power Cap		Current Power Cap	
0x0B90	Soft Min Power Cap		Pwr Perf mode	IPS Status	Proc Folding Status	107B Reserved		
0x0C00	OPAL Cmd Flags	Cmd Request ID	OPAL-OCC Cmd	Reserve	Cmd Data Length (MSB)	Cmd Data Length (LSB)	OPAL Command Data.....	
:	.....OPAL Command Data up to max of Cmd Data Length 4090 bytes							
0x1C00	OCC Rsp Flags	Cmd Request ID	OPAL-OCC Cmd	Rsp Status	Rsp Data Length (MSB)	Rsp Data Length (LSB)	OCC Response Data.....	
:	.....OCC Response Data up to max of Rsp Data Length 8698 bytes							
0x3E00	1K Reserved for future “dynamic” data							
:								
0x4200	End of data							

**Valid** – Indicates if data is valid. Pstate data should only be used if valid = 0x01.

**Static Major Version** – Indicates format version for the static data area = 0xA0.

**OCC Role** – Indicates the role of the OCC.

**0x00** = Slave.

**0x01** = Master.

**Min Frequency Pstate** – aka “Power Save” VPD point. Pstate for minimum frequency.

NOTE: There may be additional Pstates beyond this up to “Bottom Throttle Space Pstate” where frequency remains at minimum and processor throttling is invoked.

**Fixed Frequency Pstate** – Pstate for the “fixed frequency” VPD point previously known as “nominal”. With PowerVM this is the frequency used when all power management modes are disabled.

**WOF base Pstate** – Pstate for the base frequency. NOTE: Current state of system may not allow for WOF base Pstate to be reached, see Throttle Status. WOF base Pstate field is not updated when throttling occurs.

**Ultra Turbo Pstate** – Ultra Turbo Pstate, if ultra turbo is not supported this will be equal to WOF base Pstate. NOTE: Current state of system may not allow for ultra turbo Pstate to be reached either due to WOF clipping (not reflected in Throttle Status) or something else reflected in Throttle Status. Ultra Turbo Pstate field is not updated when throttling occurs.

**Fmax Pstate** – Pstate for Fmax. This is the Pstate for the highest frequency this chip may achieve. This is chip specific and may not be Pstate 0 for all chips in the system. If Fmax is not supported this will be equal to the Pstate for the highest supported frequency (i.e. UT or WOF base). NOTE: By design the WOF tables limit frequency to a maximum of UT in order for a chip to achieve frequencies higher than ultra turbo (to the chip's Fmax) WOF must be disabled. Fmax Pstate field is not updated when throttling occurs nor on WOF enable/disable changes.

**Static Minor Version** – Indicates minor format version for the static data area. A change in the minor version should not impact how previous versions are parsed. If a parsing change to an existing field is required the static major version needs to be updated.

**0x01** – Original static minor version

**Bottom of Throttle Space Pstate** – This is the last possible (highest number) Pstate supported. Pstates between “Min Frequency” and this Pstate is changing processor throttling only, the frequency will remain at the minimum frequency.

**Pstate Number / Valid / Frequency** – Continuously numbered from 0 to min frequency Pstate. NOTE: Pstates down to bottom of throttle space will not be reflected since they do not change frequency.

**Valid:**

**0 = Pstate not supported.** Due to Fmax being unique per chip not all chips may support the top frequency Pstates (i.e. Pstate 0)

**1 = Pstate supported.**

**OCC State** – Indicates the current state of the OCC. Used to know what OPAL-OCC cmd/rsp communication is allowed. Used by PHYP to know when OCC is in active state.

**0x00 = OCC not running.** No communication allowed.

**0x01 = Standby.** No communication allowed.

**0x02 = Observation.** Pstates and WOF are disabled, OCC monitoring only no actuation. Communication allowed and is command dependent.

**0x03 = Active.** Pstates enabled, this is full function. Communication allowed and is command dependent.

**0x04 = Safe.** OCC in a failure state waiting for a reset. NOTE: PGPE may still be running until the reset actually happens. No communication allowed. Just like CPU throttle status, some failures will not allow for OCC to update state to safe.

**0x05 = Characterization.** Pstates enabled with characterization as PMCR owner. WOF is disabled. OCC monitoring only, no actuation. Communication allowed and is command dependent.

**Dynamic Major Version** – Indicates major format version for the dynamic data area = 0x00

**Dynamic Minor Version** – Indicates minor format version for the dynamic data area. A change in the minor version should not impact how previous versions are parsed. If a parsing change to an existing field is required the major version needs to be updated.

**0x00** – Original minor version

**0x01** – Support for GPUs Present field added

**0x02** – Added mode and IPS fields

**GPUs Present** – ONLY SUPPORTED ON SYSTEMS THAT OCC HAS GPU MANAGEMENT.  
Bit mask indicating GPUs present behind this OCC (processor) only.

Bit 0 (msb)	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7 (lsb)
Reserved = '00000'					GPU2 presence '1' = present '0' = not present	GPU1 presence '1' = present '0' = not present	GPU0 presence '1' = present '0' = not present

**WOF State** – Indicates state of WOF. Defaults to WOF enabled on OCC boot/reset and then can be set with OPAL-OCC WOF Control Command.

**0x00** = WOF Disabled (highest frequency WOF base)

**0x01** = WOF Enabled (highest frequency ultra turbo)

**0x02** = Fmax mode. By definition WOF is Disabled (highest frequency chip Fmax)

**0xE0** = WOF Disabled by OCC. WOF control command wants WOF enabled but OCC is unable to enable WOF (highest frequency WOF base)

**Processor Throttle Status** – Indicates reason that OCC may have limited Max Pstate.

NOTE: 0x06/0x07/0x09/0x0A/0xAA are for debug purposes only and should not result in errors.

**0x00** = No throttle

**0x01** = Power Cap (max freq clipped below WOF base)

**0x02** = Processor Over Temperature (max freq clipped below WOF base)

**0x03** = Power Supply Failure (currently not used)

**0x04** = Overcurrent Protection Error. OCP is not running, max freq WOF base

**0x05** = OCC reset. Some failures will not allow for OCC to update throttle status.

**0x06** = Exceeded Power Cap above WOF base (throttling is happening but not below WOF base)

**0x07** = Processor Over Temperature above WOF base (throttling is happening but not below WOF base)

**0x08** = VRM Vdd Over Temperature (max freq clipped below WOF base)

**0x09** = VRM Vdd Over Temperature above WOF base (throttling is happening but not below WOF base)

**0x0A** = Chip Fmax limited. Since not all chips will support pState 0 which is defined as the maximum frequency of each individual chip Fmax this reason indicates that this chip has a lower Fmax than at least one other chip in the system, this reason should not result in any error messages. (NOTE: if the OCC must clip frequency less than the chip's Fmax

a different throttle status to indicate why i.e. power, thermal...)

**0xAA** = Manufacturing override (OCC in control of Pstates set by PMCR mode register (SCOM) to lock out OPAL from changing pstates)

**Memory Throttle Status** – Indicates reason that OCC has throttled memory

**0x00** = No throttle

**0x01** = Power Cap

**0x02** = Memory Over Temperature

**Quick Power Drop** – Indicates if QPD is asserted

**0x00** = QPD not asserted

**0x01** = QPD asserted

**Power Shifting Ratio** – Indicates percentage (0-100) of power to take away from the CPU vs GPU when shifting power to maintain a power cap. 100=take all pwr from CPU

**Power Cap Type** – Indicates type of power cap in effect

**0x00** = System Default (no set user cap)

**0x01** = User Set Power Cap set out of band

**0x02** = User Set Power Cap set in band

**Hard Minimum Power Cap** – Hard Minimum system (node) power cap in 1W units (output power) that the OCC can guarantee to maintain

**Maximum Power Cap** – Maximum system (node) power cap in 1W units (output power) that is allowed to be set

**Current Power Cap** – Current system (node) power cap in 1W units (output power)

**Soft Minimum Power Cap** – Minimum system (node) power cap in 1W units (output power) that is allowed to be set. A power cap set below the minimum power cap is not guaranteed.

**Power and Performance mode** – Current system power and performance mode the system is running in. Valid from master OCC only.

**0x01** = Modes Disabled aka Nominal

**0x04** = **Safe**. OCC failed and is in reset (or about to be). NOTE: Some rare failure cases may not allow for OCC to update mode to safe prior to being put into reset.

**0x05** = Static Power Save

**0x09** = Fmax

**0x0A** = Dynamic Performance

**0x0B** = FFO

**0x0C** = Max Performance

**Idle Power Saver Status** – Current Idle Power Saver status. Valid from master OCC only, this will be 0 on non-master OCCs. OCC will update IPS status to 0 when updating mode to safe.

**0x00** – Idle Power Saver disabled

**0x01** – Idle Power Saver enabled

**0x03** – Idle Power Saver active

**Processor Folding Status** – Indicates if processor folding should currently be enabled or disabled. Valid from master OCC only.

**0x00** – Folding should be disabled

**0x01** – Folding should be enabled

### 12.1.1 Parameter Table Definition

Within the static portion of the shared memory interface there is a “Parameter Table” containing an 8 byte header and then a maximum of 25 entries. Each entry is 32 bytes containing 2 byte identifier, 1 byte flags, 2 byte data field and 27 byte null terminated text description.

NOTE: Order of table matters for PHYP. Table must be ordered first by Parameter ID and then under each parameter ID the 0xFF value ID must be first after the 0xFF value ID remaining value IDs can be in any order.

#### ID:

<b>Byte 0 (MSB)</b>	<b>Parameter Identifier</b> Defines the parameter.  <b>0xD0-0xFF</b> = Reserved for PHYP usage
<b>Byte 1</b>	<b>Parameter Value ID</b> Defines the value for the parameter defined in byte 1.  <b>0xFF</b> = Not a value, the ID is the parameter

#### Parameter Flags:

Bit	
<b>0 (msb)</b>	<b>Report</b>  ‘1’ = the parameter has an actual value associated with it that should be reported to the OS. If bit 1 “Data Field Valid” is set then that is the value for this parameter else it is up to PHYP to know where to get the value.  ‘0’ = the parameter is just for translation purposes and there is no value
<b>1</b>	<b>Data Field Valid</b> Indicates if the following 2 byte data field is valid. Only parameters that have a value that cannot change at runtime may use the data field. ‘1’ = The following 2 byte data field is the value for the parameter ‘0’ = Data field unused
<b>2</b>	<b>Master Only</b> ‘1’ = This parameter is only available from the master OCC ‘0’ = This parameter is chip specific
<b>3:7</b>	<b>Reserved</b>

The following table gives a list of parameters that will be written out to shared memory when

the OCC is started. This is a hard coded table that requires an OCC firmware update for any changes. All OCCs will write out this table, but only the master OCC will write out entries that are marked as “master only”.

ID		Flags (1 byte)	Data (2 bytes)	Text Description (27 bytes)	Notes
Param ID (1 byte)	Param Value ID (1 byte)				
0x01	0xFF	0xA0 * Report * Master	N/A	Power and Performance Mode	PHYP to read from OCC shared memory (MBOX proc folding message no longer sent). When there is a mode change the master OCC will send an interrupt to PHYP to indicate shared memory update for PHYP to re-read.  Description for mode value xx will be Parameter ID 0x01 with Parameter Value ID xx
0x01	0x01	0x20 * Master	N/A	Static	Used for translating the “Power and Performance mode” value into useful text
0x01	0x03	0x20 * Master	N/A	SFP lab only	
0x01	0x04	0x20 * Master	N/A	Safe	
0x01	0x05	0x20 * Master	N/A	Power Saving	
0x01	0x09	0x20 * Master	N/A	Fmax	
0x01	0x0A	0x20 * Master	N/A	Balanced Performance	
0x01	0x0B	0x20 * Master	N/A	Fixed Frequency Override	
0x01	0x0C	0x20 * Master	N/A	Maximum Performance	
0x02	0xFF	0xA0	N/A	Idle Power Saver	PHYP to read from OCC shared memory (MBOX proc

		* Report * Master		Status	folding message no longer sent) When there is an IPS change the master OCC will send an interrupt to PHYP to indicate shared memory update for PHYP to re-read.  Text descriptions for Idle Power Save status xx will be Parameter ID 0x02 with Parameter Value ID xx
0x02	0x00	0x20 * Master	N/A	Idle Power Saver Disabled	Used for translating the “Idle Power Save Status” value into useful text
0x02	0x01	0x20 * Master	N/A	Idle Power Saver Enabled	
0x02	0x03	0x20 * Master	N/A	Idle Power Saver Active	
0x02	0xE0	0x20 * Master	N/A	Not Supported	
0x03	0xFF	0xE0 * Report * Data * Master	xxxx	Minimum Frequency (MHz)	Minimum system frequency in MHz
0x04	0xFF	0xE0 * Report * Data * Master	xxxx	Static Frequency (MHz)	Aka Nominal. In MHz PHYP will now read from here and no longer from device tree.
0x05	0xFF	0x60 * Data * Master	xxxx	Base Frequency (MHz)	WOF Base frequency in MHz
0x06	0xFF	0xE0 * Report * Data * Master	xxxx	Maximum Frequency (MHz)	Max frequency in MHz for the system
0x07	0xFF	0x40	xxxx	Chip Max Frequency (MHz)	Chip specific frequency in MHz Chip may only reach



		* Data			this when in Fmax mode
0x08	0xFF	0xA0 * Report * Master	N/A	Processor Folding Status	<p>PHYP to read folding status from OCC shared memory. When there is a folding change the master OCC will send an interrupt to PHYP to indicate shared memory update for PHYP to re-read.</p> <p>There are no parameter value IDs for translation. The folding value in OCC shared memory is either enable (1) or disable (0).</p>

---

## 12.2 OPAL-OCC Command/Response Interface

Within the shared memory interface there is a command and a response buffer for an OPAL-OCC command/response interface.

### 12.2.1 OPAL-OCC Command Buffer

OPAL Cmd Flags	Cmd Request ID	OPAL- OCC Cmd	Reserved	Cmd Data Length (MSB)	Cmd Data Length (LSB)	OPAL Command Data.....
.....OPAL Command Data up to max of Cmd Data Length 4090 bytes						

**OPAL Cmd Flags** – Provides general status of command. Bit defined:

Bit 0 (msb)	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7 (lsb)
Cmd Ready							

**Command Ready** – ‘1’ indicates that OPAL has a command ready for OCC to process.

**Command Request ID** – OPAL handle to identify request, repeated in response from OCC

**OPAL-OCC Cmd** – Command sent by OPAL repeated in response from OCC. See “OPAL-OCC Commands” section for defined values.

**Command Data Length** – 2 byte data length. Maximum value of 4090

**Command Data** – 0 to 4090 bytes of command specific data

### 12.2.2 OPAL-OCC Response Buffer

OCC Rsp Flags	Cmd Request ID	OPAL-OCC Cmd	Rsp Status	Rsp Data Length (MSB)	Rsp Data Length (LSB)	OCC Response Data.....
.....OCC Response Data up to max of Rsp Data Length 8698 bytes						

**OCC Rsp Flags** – Provides general status of response. Bit defined:

Bit 0 (msb)	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7 (lsb)
						OCC in progress	Rsp Ready

**OCC in Progress** – ‘1’ indicates that OCC is currently processing a command. (not implemented)

**Response Ready** – ‘1’ indicates that a response from OCC is ready for OPAL

**Command Request ID** – Same ID found in the command that this response is for

**OPAL-OCC Cmd** – Same Command found in the command that this response is for

**Response Status** – Indicates success/failure status of the command

**Response Data Length** – 2 byte data length. Maximum value of 8698

**Response Data** – 0 to 8698 bytes of command specific response data

### 12.2.3 OPAL-OCC Command/Response Sequence

1. OPAL must ensure that another command is not sent until the response from any previous command has been processed or the command has timed out. If 'Rsp Ready' is 1 it is up to OPAL to ensure that they have processed the response prior to sending a new command. Remaining steps are only done if OPAL has determined that the response from the last command has been processed or the previous command has timed out.
2. OPAL writes 0x00s to the OCC Rsp flags to ensure 'Rsp Ready' is 0 (if not being done when previous command response is processed)
3. OPAL writes OPAL-OCC Command value, and command specific data to the OPAL-OCC command buffer.
4. OPAL informs OCC there is a command to process by writing 0x80 to OPAL Cmd flags and writing a different command request ID than the previous command. NOTE: Command request ID must be different than the previous command for OCC to know that this is a new command request.
5. OCC will be polling the OPAL Cmd flags and Command Request ID every 4ms and see 'Cmd Ready' is set in command flags and the Command Request ID is different than the last command processed that OCC knows this is a new command to process.
6. OCC processes the command and writes the response area including writing 0x01 to OCC Rsp flags to indicate the response is ready. OCC will not clear the 'Cmd Ready' bit, OCC will look for a different Command Request ID to prevent processing the command again.
7. OCC writes OCC Miscellaneous Register 'core\_ext\_intr' bit to '1' and 'ext\_intr\_reason' bit 3 to '1' to send interrupt to OPAL to indicate shared memory interface has been updated
8. OPAL receives interrupt and reads 0x01 (Rsp Ready AND OCC not in progress) from OCC Rsp flags and processes the response. OPAL must clear 'core\_ext\_intr' bit in OCC Misc register to allow OCC to send additional interrupts.

## 12.2.4 Error Handling

### 12.2.4.1 Timeout for Command Processing

Command timeout is defined per command in the “OPAL-OCC Commands” section. This time is from when OPAL sets the “command ready” bit in the OPAL Command Flags to OPAL seeing “response ready” bit set in the OCC Response Flags. If the timeout is reached OPAL should first verify that the OCC is in a state that allows for communication (read OCC State from OPAL-OCC interface) if so then OPAL will retry the command once. If not OPAL should zero out the command buffer and wait for indication from OCC that shared memory interface has been updated (will happen when OCC updates state) before sending any additional commands

### 12.2.4.2 Response Failures

This includes command value or request ID mismatch in OCC response. OPAL should retry the command once. NOTE: If OPAL does not clear the ‘Rsp Ready’ in the OCC Rsp flags after processing a response or before sending the next command the request ID mismatch may be due to OCC not processing the command yet i.e. Rsp Ready still set from last command. OPAL should wait the timeout time and re-check the response before considering request ID mismatch an error and retrying a command.

### 12.2.5 OPAL-OCC Commands

This section defines OPAL-OCC commands that are sent via the OPAL-OCC command/response interface inside the shared memory interface.

#### **OCC State**

- Defines OCC states that the command is supported in
  - O = Observation/Characterization
  - A = Active
- A command sent to an OCC in a state that does not support the command will be rejected by the OCC with 0x16 (PRESENT STATE PROHIBITS)

#### **Timeout**

- Defines how long OPAL should wait for OCC to have a response ready before retrying the command. Time is from when “command ready” bit in the OPAL Command Flags is set to the “response ready” bit set in the OCC Response Flags

#### **Supported By**

- A master OCC is also considered a slave and will not reject any command
- A slave OCC will reject a command with 0x11 (INVALID COMMAND) for any command that is only to be supported by a master OCC

<b><i>OPAL-OCC Command</i></b>		<b><i>OCC State</i></b>	<b><i>Timeout</i></b>	<b><i>Supported By</i></b>	
				<b><i>Master OCC</i></b>	<b><i>Slave OCC</i></b>
0xD0	Clear Sensor Data	O, A	1s	Y	Y
0xD1	Set Power Cap in Band	O, A	1s	Y	N
0xD2	Write Power Shifting Ratio	O, A	1s	Y	Y
0xD3	Select Sensor Groups	O, A	1s	Y	Y
0xD4	WOF Control	O, A	1s	Y	N

### 12.2.5.1 AMESTER Pass Thru – NOT SUPPORTED

This command is for sending commands from AMESTER to the OCC. The full command data from AMESTER should be sent to the OCC and the command data length must represent the number of bytes being sent. The full response data specified by the response data length should be sent back to AMESTER. NOTE: AMESTER is not supported while in secure memory (SMF) mode and all AMESTER commands will be rejected with response status 0x11 (Invalid command).

#### **AMESTER Pass Thru Command:**

<b>OPAL-OCC Command Value</b>	0x41		
<b>Data Length</b>	Variable		
<b>Command Data</b>	Command data from AMESTER:		
	<b>Byte Offset</b>	<b>Description</b>	
	0	AMESTER API command = 0x3C	
	1	Reserved = 0x00	
	2 thru end of cmd data	<b>Byte offset 2: AMESTER Sub-Command</b>	<b>Byte offset 3 thru end of buffer sub-command specific data</b>
		0x07 – Get Multiple Sensor Data	List of 16 bit sensor IDs to get sensor data for
		0x0A – Get AMESTER API Version	No data
		0x1C – Get AMESTER Component Level Constants	No data
		0x21 – Clear min/max fields of all sensors	No data
		0x25 – Get sensor info	16 bit sensor ID followed by 8 bit field type of fields to return
		0x30 – Get Trace Buffer Configuration	8 bit trace buffer ID to read configuration
		0x31 – Configure Trace Buffer	8 bit trace buffer ID to configure. 16 bit number of sensors, 16 bit number of parameters, list of 16 bit sensor IDs and 16 bit parameter IDs to trace
		0x32 – Read Trace Buffer	8 bit trace buffer ID to read, 32 bit trace buffer offset to start read from
		0x33 – Start Trace Buffer Recording	No data
		0x34 – Stop Trace	No data

		Buffer Recording	
		0x3F – Return all configurable parameters for a trace	8 bit trace buffer ID
		0x40 – Get number of parameters	No data
		0x41 – Return configuration parameters	16 bit parameter ID
		0x42 – Read Parameter	32 bit byte offset, 16 bit parameter ID

### **AMESTER Pass Thru Response:**

<b>OPAL-OCC Command Value</b>	0x41		
<b>Response Status</b>	0x00 = Success 0x11 = Invalid Command – in secure mode 0x15 = Internal OCC error		
<b>Data Length</b>	Variable.		
<b>Response Data</b>	Response specific data for AMESTER command:		
	<b>Byte Offset</b>	<b>Description</b>	
	0	AMESTER API command = 0x3C	
	1	Reserved = 0x00	
	2 thru end of rsp data	<b>AMESTER Sub-Command</b>	<b>Sub-command specific response data</b>
		0x07 – Get Multiple Sensor Data	Sensor data for the given IDs
		0x0A – Get AMESTER API Version	Major and Minor version amester fw supported
		0x1C – Get AMESTER Component Level Constants	Major and Minor version fw supported, number of sensors supported, number of trace buffers supported
		0x21 – Clear min/max fields of all sensors	No response data
		0x25 – Get sensor info	Sensor fields for give sensor ID and field(s) requested
		0x30 – Get Trace Buffer Configuration	Trace buffer configuration
		0x31 – Configure Trace	No response data



		Buffer	
		0x32 – Read Trace Buffer	Trace buffer starting from given offset
		0x33 – Start Trace Buffer Recording	No response data
		0x34 – Stop Trace Buffer Recording	No response data
		0x3F – Return all configurable parameters for a trace	Trace buffer parameters
		0x40 – Get number of parameters	Number of parameters supported
		0x41 – Return configuration parameters	Parameter configuration for given parameter ID
		0x42 – Read Parameter	Parameter starting from given offset
		0xFD – Configure AMESTER data length (input 2 byte data length)	No response data

### 12.2.5.2 Clear Sensor Data

This command is used to tell the OCC to clear the minimum and maximum for every sensor for the given owner. NOTE: The OCC owned min/max cannot be cleared via this command. The OCC fw owns clearing of its own data.

#### Clear Sensor Data Command:

<b>OPAL-OCC Command Value</b>	0xD0	
<b>Data Length</b>	0x0004	
<b>Command Data</b>	<b>Byte 1 (first)</b>	Sensor Data Owner ID – Indicates owner to clear all sensors: Bit mask where ‘1’ indicates to clear sensors min/max for the owner: Bit 0 (msb) = reserved Bit 1 = Job Scheduler Bit 2 = Profiler Bit 3 = CSM Bits 4:7 = reserved
	<b>Byte 2</b>	Reserved = 0x00
	<b>Byte 3</b>	Reserved = 0x00
	<b>Byte 4 (last)</b>	Reserved = 0x00

#### Clear Sensor Data Response:

<b>OPAL-OCC Command Value</b>	0xD0	
<b>Response Status</b>	0x00 = Success 0x12 = Invalid Command Data Length 0x13 = Invalid Sensor Data Owner ID 0x15 = Internal OCC error	
<b>Data Length</b>	0x0004 on success	
<b>Response Data</b>	<b>Byte 1 (first)</b>	Sensor Data Owner ID – Indicates owner of the data that was cleared, should match sensor data owner ID in command data
	<b>Byte 2</b>	Reserved = 0x00
	<b>Byte 3</b>	Reserved = 0x00
	<b>Byte 4 (last)</b>	Reserved = 0x00

### 12.2.5.3 Set Power Cap in Band

This command is used to set the system (node) power cap in band. This is an output power cap and can only be sent to the master OCC. Setting the power cap in band will over write any power cap that was set out of band, the BMC will see the current power cap change in the OCC poll response and update the power cap being reported out of band to reflect what was set in band. Likewise, any out of band power cap set after this command will over write the power cap set in band and will be reflected in the shared memory interface power cap. I.e. There is only one power cap and the OCC will use the last set power cap regardless of how it was set.

#### **Set Power Cap in Band Command:**

<b>OPAL-OCC Command Value</b>	0xD1	
<b>Data Length</b>	0x0002	
<b>Command Data</b>	<b>Bytes 1-2</b>	Output power cap to set in 1W units (MSB first) 0x0000 = Clear Power Cap

#### **Set Power Cap in Band Response:**

<b>OPAL-OCC Command Value</b>	0xD1	
<b>Response Status</b>	0x00 = Success 0x11 = Invalid Command – command sent to slave only OCC 0x12 = Invalid Command Data Length 0x13 = Invalid Data – the non-zero power cap to set is not within the power cap min and max range defined in the shared memory interface	
<b>Data Length</b>	0x0002	
<b>Response Data</b>	<b>Bytes 1-2</b>	Current power cap. (MSB first) On a successful set this should match the power cap sent in the command data

#### 12.2.5.4 Write Power Shifting Ratio

This command is used to write the CPU-GPU power shifting ratio used by the OCC power capping algorithm. Until received the OCC will be using the power shifting ratio defined in the xml file.

##### **Write Power Shifting Ratio Command:**

<b>OPAL-OCC Command Value</b>	0xD2	
<b>Data Length</b>	0x0001	
<b>Command Data</b>	<b>Byte 1</b>	0-100 Power Shifting Ratio in 1% units. 0=Take zero power away from CPU (cap GPU first) : 100=Take all power away from CPU first

##### **Write Power Shifting Ratio Response:**

<b>OPAL-OCC Command Value</b>	0xD2	
<b>Response Status</b>	0x00 = Success 0x12 = Invalid Command Data Length 0x13 = Invalid Data – the power shifting ratio is not valid	
<b>Data Length</b>	0x0001	
<b>Response Data</b>	<b>Byte 1</b>	Power Shifting Ratio. On a successful write this should match the power shifting ratio sent in the command data

### 12.2.5.5 Select Sensor Groups

This command is used to tell the OCC group(s) of sensors to actively copy to main memory. NOTE: Having fewer groups being copied to main memory allows the selected group(s) to be copied to main memory more frequently. By default, the OCC will copy all sensors to main memory unless this command is received. This command does not persist across OCC resets, after an OCC reset the OCC will default back to copying all sensors to main memory. Sending this command with no groups selected will turn off all copies of sensors to memory.

#### **Select Sensor Groups Command:**

<b>OPAL-OCC Command Value</b>	0xD3	
<b>Data Length</b>	0x0002	
<b>Command Data</b>	<b>Bytes 1-2</b>	Sensor Group Mask – Bit mask where ‘1’ indicates to update sensors of sensor type to main memory ‘0’ indicates sensor type will not be updated in main memory: Bits 0:5 = reserved Bit 6 = Performance Bit 7 = reserved Bit 8 = Power Bit 9 = Frequency Bit 10 = Time Bit 11 = Utilization Bit 12 = Temperature Bit 13 = Voltage Bit 14 = Current Bit 15 (lsb) = Generic

#### **Select Sensor Groups Response:**

<b>OPAL-OCC Command Value</b>	0xD3	
<b>Response Status</b>	0x00 = Success 0x12 = Invalid Command Data Length 0x13 = Invalid Sensor Group mask 0x15 = Internal OCC error	
<b>Data Length</b>	0x0002 on success	
<b>Response Data</b>	<b>Bytes 1-2</b>	Sensor Group Mask – Echo back group mask selected, should match sensor group mask in command data

#### 12.2.5.6 WOF Control

This command is used to enable or disable WOF for the system (node), ultimately controlling the maximum frequency the chips may achieve. This command can only be sent to the master OCC and the master OCC will broadcast the new WOF state to all slave OCCs. Status of WOF is reflected in “WOF State” field in the dynamic shared memory interface.

##### **WOF Control Command:**

<b>OPAL-OCC Command Value</b>	0xD4	
<b>Data Length</b>	0x0001	
<b>Command Data</b>	<b>Byte 1</b>	<b>WOF Control</b> 0x00 = Disable WOF (max freq WOF base) 0x01 = Enable WOF (max freq UT) 0x02 = Enable Fmax mode. By definition this disables WOF (max freq per chip Fmax)

##### **WOF Control Response:**

<b>OPAL-OCC Command Value</b>	0xD4	
<b>Response Status</b>	0x00 = Success 0x11 = Not supported. System owner has disabled user WOF control or command sent to non-master OCC 0x12 = Invalid Command Data Length 0x13 = Invalid WOF Control 0x15 = Internal OCC error	
<b>Data Length</b>	0x0001	
<b>Response Data</b>	<b>Byte 1</b>	WOF Control. On success this should match the WOF Control sent in the command data

---

## 12.3 OCC Main Memory Sensor Data

In addition to the OCC poll response providing some sensors out of band a larger set of sensors will be provided in main memory to be collected in band. The main memory OCC sensor data will use BAR2 (OCC Common is per physical drawer). Starting address is at offset 0x00580000 from BAR2 base address. Maximum size is 1.5MB

<b><i>Start (Offset from BAR2 base address)</i></b>	<b><i>End</i></b>	<b><i>Size</i></b>	<b><i>Description</i></b>
0x00580000	0x005A57FF	150kB	OCC 0* Sensor Data Block
0x005A5800	0x005CAFFF	150kB	OCC 1* Sensor Data Block
:	:	:	:
0x00686800	0x006ABFFF	150kB	OCC 7* Sensor Data Block
0x006AC000	0x006FFFFFFF	336kB	Reserved

\*OCC number is the PBAX chip ID (rcv\_chipid field in the PBAX Configuration Register)

### 12.3.1 OCC N Sensor Data Block Layout (150kB)

The sensor data block layout is the same for each OCC N.

NOTE: Block copy has 4K limit. The ping and pong buffers of sensor readings are limited to 40kB to allow 4K of sensor readings updated every 8ms to have all sensors updated every 80ms. The sensor names and static data associated with sensors is written once and may be larger than 40kB.

<b><i>Start (Offset from OCC N Sensor Data Block)</i></b>	<b><i>End</i></b>	<b><i>Size</i></b>	<b><i>Description</i></b>
0x00000000	0x000003FF	1kB	Sensor Data Header Block
0x00000400	0x0000CBFF	50kB	Sensor Names
0x0000CC00	0x0000DBFF	4kB	Reserved
0x0000DC00	0x00017BFF	40kB	Sensor Readings ping buffer
0x00017C00	0x00018BFF	4kB	Reserved
0x00018C00	0x00022BFF	40kB	Sensor Readings pong buffer
0x00022C00	0x000257FF	11kB	Reserved

### 12.3.1.1 Sensor Data Header Block (1kB)

The Sensor Data Header Block is written once by the OCC during initialization after a load or reset.

Offset	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x0000	Valid (0x01)	Header Version (0x01)	Number of sensors		Sensor Readings Version	Reserved		
0x0008	Sensor Names offset				Sensor Names Version	Bytes / Sensor Name	Reserved	
0x0010	Sensor Readings ping buffer offset				Sensor Readings pong buffer offset			
0x0018	Reserved							
0x0400	End of data							

**Valid** – Written to 0x01 after the “Sensor Names” buffer has been written with all sensor static data. When 0x01 this header and the “Sensor Names” buffer are ready.

**Header Version** – Indicates format version for the sensor data header. Currently only 0x01 supported.

**Number of Sensors** – Indicates number of sensors in “Sensor Names”, “Sensor Readings Ping Buffer” and “Sensor Readings Pong Buffer”. The exact same number of sensors must exist in each of these buffers.

**Sensor Readings Version** – Indicates format version for the Sensor Readings ping and pong buffers. Both ping and pong buffers must be using the same version.

**Sensor Names Offset** – Location of the “Sensor Names” buffer as an offset from the start of OCC N Sensor Data block. Currently defined as 0x00000400

**Sensor Names Version** – Indicates format version for the Sensor Names buffer.

**Bytes/Sensor Name** – Indicates length of each sensor in the “Sensor Names” buffer. All sensors in “Sensor Names” will be the same length and format.

**Sensor Readings Ping Buffer Offset** – Location of the “Sensor Readings Ping Buffer” as an offset from the start of OCC N Sensor Data block. Currently defined as 0x0000DC00

**Sensor Readings Pong Buffer Offset** – Location of the “Sensor Readings Pong Buffer” as an offset from the start of OCC N Sensor Data block. Currently defined as 0x00018C00



### 12.3.1.2 Sensor Names (50kB)

The Sensor Names block is written once by the OCC during initialization after a load or reset. It contains static information for each sensor. The number of sensors, format version and length of each sensor is defined in the “Sensor Data Header Block”. “Sensor Names” is valid if the “Valid” byte in the “Sensor Data Header Block” is 0x01. The first sensor starts at offset 0 followed immediately by the next sensor.

Sensor Names Version = 0x01

Each sensor in “Sensor Names” block will be 48 bytes with the following format:

FIELD	SIZE (BYTES)	DESCRIPTION									
Name	16	Sensor name									
Units	4	Sensor units of measurement									
Gsid	2	Global sensor ID – assigned by its constructor									
Freq	4	Update frequency									
scale_factor	4	Scaling factor									
Type	2	Sensor type: <table border="1"> <tr> <td>0x0001 = Generic</td><td>0x0008 = Temp</td><td>0x0040 = Freq</td></tr> <tr> <td>0x0002 = Current</td><td>0x0010 = Util</td><td>0x0080 = Power</td></tr> <tr> <td>0x0004 = Voltage</td><td>0x0020 = Time</td><td>0x0200 = Perform</td></tr> </table>	0x0001 = Generic	0x0008 = Temp	0x0040 = Freq	0x0002 = Current	0x0010 = Util	0x0080 = Power	0x0004 = Voltage	0x0020 = Time	0x0200 = Perform
0x0001 = Generic	0x0008 = Temp	0x0040 = Freq									
0x0002 = Current	0x0010 = Util	0x0080 = Power									
0x0004 = Voltage	0x0020 = Time	0x0200 = Perform									
Location	2	Sensor location: <table border="1"> <tr> <td>0x0001 = System</td><td>0x0008 = Memory</td><td>0x0040 = Core</td></tr> <tr> <td>0x0002 = Proc</td><td>0x0010 = VRM</td><td>0x0080 = GPU</td></tr> <tr> <td>0x0004 = Partition</td><td>0x0020 = OCC</td><td>0x0100 = Quad</td></tr> </table>	0x0001 = System	0x0008 = Memory	0x0040 = Core	0x0002 = Proc	0x0010 = VRM	0x0080 = GPU	0x0004 = Partition	0x0020 = OCC	0x0100 = Quad
0x0001 = System	0x0008 = Memory	0x0040 = Core									
0x0002 = Proc	0x0010 = VRM	0x0080 = GPU									
0x0004 = Partition	0x0020 = OCC	0x0100 = Quad									
sensor_structure_version	1	Indicates type of data structure used for the sensor readings in the ping and pong buffers for this sensor 0x01 = Full reading structure (min/max fields supported) 0x02 = Counter structure (this sensor is a counter no min/max/current)									
reading_offset	4	offset from the start of the ping and pong buffers to the readings for this sensor									
sensor_specific_info1	1	Additional sensor information specific to sensor. PWRAPSSCHx --> ADC func ID: 1=Mem Proc0 : 4=Mem Proc3 5=Proc0 : 8=Proc3 9=Proc0 cache/io/pcie : 12=Proc3 cache/io/pcie 13=IO A 14=IO B 15=IO C 16=Fans A 17=Fans B 18=Storage A 19=Storage B 22=Total System Power									

		23=Memory Cache 24=GPU Proc0-0 25=Mem Proc0-0 26=Mem Proc0-1 27=Mem Proc0-2 29=GPU Proc0-1 30=GPU Proc0-2 31=GPU Proc1-0 32=GPU Proc1-1 33=GPU Proc1-2
Reserved	8	Reserved = 0

### 12.3.1.3 Sensor Readings Ping and Pong Buffers (40kB each)

There will be two 40kB buffers to store the sensor readings. One buffer that is currently being updated by the OCC and one that is available to be read. Each of these buffers will be the same format. The number of sensors and the format version of the ping and pong buffers is defined in the “Sensor Data Header Block”. NOTE: Each sensor within the ping and pong buffers may be of a different format and length. For each sensor the length and format are determined by its “sensor\_structure\_version” in the Sensor Names buffer.

Sensor Readings Version = 0x01

Offset	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7							
0x0000	Valid (0x01)	Reserved													
0x0008	Sensor Readings														
:	:														
0xA000	End of data														

**Valid** – Written to 0x01 after the buffer is completely written and available to be read. Written to 0x00 before the start of being updated.

#### 12.3.1.3.1 sensor\_structure\_version = 0x01 (Full Reading)

A sensor with sensor\_structure\_version of 1 will be 48 bytes and have the following format in the Sensor Readings Ping and Pong buffers:

FIELD	SIZE (BYTES)	DESCRIPTION
Gsid	2	Global sensor ID – assigned by its constructor
timestamp	8	64bit time base counter in the core. Resolution is 512MHz
Sample	2	Latest sample of this sensor
sample_min	2	Minimum value since last OCC reset
sample_max	2	Maximum value since last OCC reset
CSM_sample_min	2	Minimum value since last reset request by CSM
CSM_sample_max	2	Maximum value since last reset request by CSM
profiler_sample_min	2	Minimum value since last reset request by profiler
profiler_sample_max	2	Maximum value since last reset request by profiler
job_s_sample_min	2	Minimum value since last reset by job scheduler
job_s_sample_max	2	Maximum value since last reset by job scheduler
Accumulator	8	Accumulator register for this sensor
update_tag	4	Count of the number of 'ticks' that have passed between updates to this sensor – used for time-derived sensors
Sample_info	2	Information regarding Sample used only for following sensors: <ul style="list-style-type: none"><li>• DDSMIN gives core (0..31) that sample is from</li></ul>
Reserved	6	

#### 12.3.1.3.2 sensor\_structure\_version = 0x02 (Counter)

A sensor with sensor\_structure\_version of 2 will be 24 bytes and have the following format in the Sensor Readings Ping and Pong buffers:

FIELD	SIZE (BYTES)	DESCRIPTION
Gsid	2	Global sensor ID – assigned by its constructor
Timestamp	8	64bit time base counter in the core. Resolution is 512MHz
Accumulator	8	Accumulator register for this sensor (count of number of times sensor is “on” each time the sensor is read)
Sample	1	Latest sample of this sensor (0 or 1)
Reserved	5	

## 12.3.2 Main Memory OCC Sensor List

This is a list of sensors that are currently supported being written to main memory.

- Master only sensors are only available from the master OCC
- A master OCC is also a slave and will have all sensors
- Core number 0..31 in a sensor name represents physical core number from CORE base address
- Core sensors will exist for the max number of physical cores possible. There is no support to dynamically add/delete sensors at run time. De-configured core sensors will exist but will not be updated.
- There should be no hard coding of sensor name, sample time, existence of a sensor... the OCC may change the sensor name, sample time, add or delete sensors from this list at any time without interlock with Linux/OPAL. All sensors that the OCC does provide should be available to the user
- Secure Memory (SMF) mode – when in secure mode the OCC will only copy a subset of sensors to main memory
- Sensors are stored in main memory grouped together by type in order to allow fewer BCE requests when only certain type(s) are being monitored
- Call Home – name of the sensor in the OCC Bxxx2A01 call home log

### 12.3.2.1 Performance Sensors

Sensor Name	SMF Mode?	Master only?	#	Unit	Type	Loc	Sample time	Scale Factor	Description	Call Home
IPS	N	N	1	MIP	PERF	PROC	8MS	1	Vector sensor that takes the average of all the cores in Processor x	<b>CHOMIPS</b> Total across all processors
STOPDEEPACTCy	Y	N	32	ss	PERF	CORE	8MS	1	Deepest actual stop state that was fully entered during sample time for core y	No
STOPDEEPPREQCy	Y	N	32	ss	PERF	CORE	8MS	1	Deepest stop state that has been requested during sample time for core y	No
IPSCy	N	N	32	MIP	PERF	CORE	8MS	1	Instructions per second for core y on this Processor	No
NOTBZECy	N	N	32	cyc	PERF	CORE	8MS	1	Not Busy (stall) cycles counter for core y on this Processor	No
NOTFINCY	N	N	32	cyc	PERF	CORE	8MS	1	Not Finished (stall) cycles counter for core y on this Processor	No
MRDMx	N	N	16	GBs	PERF	MEM	256MS	0.00064	Memory read requests per sec for MC x (0..15)	<b>CHOMBWPxMy</b> Sum of read and write requests. (where x is processor 0..3 and y is MC 0..15)
MWRMx	N	N	16	GBs	PERF	MEM	256MS	0.00064	Memory write requests per sec for MC x (0..15)	
PROCPWRTHROT	N	N	1	#	PERF	PROC	500US	1	Count of processor throttled due to power	No
PROCOTTHROT	N	N	1	#	PERF	PROC	32MS	1	Count of processor throttled for temperature	No
DDSAVG	N	N	1		PERF	PROC	8MS	1	Average DDS_DATA field in the Secondary Droop Sensor Register (SDSR) across all good cores	<b>CHOMDDSAVGPx</b> (x is processor 0..3)
DDSMIN	N	N	1		PERF	PROC	8MS	1	Minimum of DDS_MIN field in the SDRS across all good cores. NOTE: sample_info field (sensor_status) gives core (0..31) that is currently setting the minimum.	<b>CHOMDDSMINPx</b> (x is processor 0..3)
MEMOTTHROT	N	N	1	#	PERF	MEM	32MS	1	Count of memory throttled due to memory Over temperature	No
GPUxHWTTHROT	N	N	3	ns	PERF	GPU	5s	1	Total time GPU x has been throttled by	No

									hardware (thermal or power brake)	
GPUxSWTHROT	N	N	3	ns	PERF	GPU	5s	1	Total time GPU x has been throttled by software for any reason	No
GPUxSWOTTHROT	N	N	3	ns	PERF	GPU	5s	1	Total time GPU x has been throttled by software due to thermal	No
GPUxSWPWRTHROT	N	N	3	ns	PERF	GPU	5s	1	Total time GPU x has been throttled by software due to power	No

### 12.3.2.2 Power Sensors

Sensor Name	SMF Mode?	Master only?	#	Unit	Type	Loc	Sample time	Scale Factor	Description	Call Home
PWRSYS	Y	Y	1	W	POWER	SYS	500US	1	Bulk power of the system/node	CHOMPWR
PWRGPU	Y	N	1	W	POWER	GPU	500US	1	Power consumption for GPUs per socket (OCC) read from APSS	No
PWRAPSSCHx	Y	Y	16	W	POWER	SYS	500US	1	Power Provided by APSS channel x (where x=0...15) NOTE: sensor_specific_info1 field gives enum for what the channel represents	CHOMPWRAPSSCHx (where x is APSS channel 0..15)
PWRPROC	Y	N	1	W	POWER	PROC	500US	1	Power consumption for this Processor. NOTE: this is not accurate for a DCM due to not all power separated by chip	No
PWRVDD	Y	N	1	W	POWER	PROC	1MS	1	Power consumption for this Processor's Vdd (calculated from AVSBus readings)	CHOMPWRVDDPx (where x is processor 0..7)
PWRVDN - NOT SUPPORTED	Y	N	1	W	POWER	PROC	1MS	1	Power consumption for this Processor's Vdn (nest) (calculated from AVSBus readings)	No
PWRVCS	Y	N	1	W	POWER	PROC	1MS	1	Power consumption for this Processor's Vcs (calculated from AVSBus readings)	CHOMPWRVCSPx (where x is processor 0..7)

### 12.3.2.3 Frequency Sensors

Sensor Name	SMF Mode?	Master only?	#	Unit	Type	Loc	Sample time	Scale Factor	Description	Call Home
FREQA	Y	N	1	MHz	FREQ	PROC	500us	1	Processor chip level frequency requested (comes from PGPE "avg freq pstate")	CHOMFREQPx (where x is processor 0..7)

### 12.3.2.4 Utilization Sensors

Sensor Name	SMF Mode?	Master only?	#	Unit	Type	Loc	Sample time	Scale Factor	Description	Call Home
UTILCy	N	N	32	%	UTIL	CORE	8MS	0.01	Utilization of this Processor's Core y (where 100% = fully utilized): NOTE: per thread HW counters are combined as appropriate to give this core level utilization sensor	No
UTIL	N	N	1	%	UTIL	PROC	8MS	0.01	Average of all Cores UTILCy sensor	CHOMUTILPx (where x is processor 0..7)
NUTILCy	N	N	32	%	UTIL	CORE	3S	0.01	Normalized average utilization, rolling average of this Processor's Core y	No
MEMSPSTATMx	Y	N	16	%	UTIL	MEM	4MS	0.1	Static Memory throttle level setting for MCA x (0..15) when not in a memory throttle condition	No
MEMSPMx	N	N	16	%	UTIL	MEM	4MS	0.1	Current Memory throttle level setting for MCA x (0..15)	No

### 12.3.2.5 Temperature Sensors

Sensor Name	SMF Mode?	Master only?	#	Unit	Type	Loc	Sample time	Scale Factor	Description	Call Home
TEMPNESTx	N	N	2	C	TEMP	PROC	8MS	1	Temperature of the 1 nest DTS in	No

									each endcap x (0 (aka north), 1 (aka south))	
TEMPPROCIOThrm	N	N	1	C	TEMP	PROC	8MS	1	Maximum of TEMPPROCIOxy this sensor is used for fan control.	<b>CHOMTEMPIOPx</b> (where x is processor 0..7)
TEMPPROCTHRMCy	N	N	32	C	TEMP	CORE	8MS	1	The combined weighted core/L3/race track DTS temperature for processor core y.	No
TEMPVDD	N	N	1	C	TEMP	VRM	2MS	1	VRM Vdd temperature read via AVSbus by OCC	<b>CHOMTEMPVDDPx</b> (where x is processor 0..7)
TEMPMEMBUFxx	N	N	16	C	TEMP	MEM	256MS	1	Memory controller (internal sensor) temperature for Memory buffer xx (00..15) 1 OCMB read every 16ms	No
TEMPDIMMThrm	N	N	1	C	TEMP	MEM	8MS	1	Hottest DIMM temperature across all DIMMs monitored by this OCC	<b>CHOMTEMPDIMMPx</b> (where x is processor 0..7)
TEMPMCDIMMThrm	N	N	1	C	TEMP	MEM	8MS	1	Hottest temperature sensor covering both Mem controller and DIMM across all MCs monitored by this OCC	<b>CHOMTEMPMCDIMMPx</b> (where x is processor 0..7)
TEMPPMICThrm	N	N	1	C	TEMP	MEM	8MS	1	Hottest PMIC temperature across all MCs monitored by this OCC	<b>CHOMTEMPPMICPx</b> (where x is processor 0..7)
TEMPMCEXTThrm	N	N	1	C	TEMP	MEM	8MS	1	Hottest external memory controller temperature sensor across all MCs monitored by this OCC	<b>CHOMTEMPMCEXTPx</b> (where x is processor 0..7)
TEMPGPUx	?	N	3	C	TEMP	GPU	1S	1	GPU x (0..2) board temperature	No
TEMPGPUxMEM	?	N	3	C	TEMP	GPU	1S	1	GPU x hottest HBM temperature (individual memory temperatures are not available)	No

### 12.3.2.6 Voltage Sensors

Sensor Name	SMF Mode?	Master only?	#	Unit	Type	Loc	Sample time	Scale Factor	Description	Call Home
VOLTVD	Y	N	1	mV	VOLTAGE	VRM	500uS	0.1	Processor Vdd Voltage (read from AVSBus by PGPE)	No
VOLTVDSENSE	Y	N	1	mV	VOLTAGE	PROC	500uS	0.1	Vdd Voltage at the remote sense. (AVS reading adjusted for loadline)	No
VOLTVDN - NOT SUPPORTED	Y	N	1	mV	VOLTAGE	VRM	500uS	0.1	Processor Vdn Voltage (read from AVSBus by PGPE)	No
VOLTVC	Y	N	1	mV	VOLTAGE	VRM	500uS	0.1	Processor Vcs Voltage (read from AVSBus by PGPE)	No
VOLTVCSENSE	Y	N	1	mV	VOLTAGE	PROC	500uS	0.1	Vcs Voltage at the remote sense. (AVS reading adjusted for loadline)	No
VOLTDRPOPCNTCx	Y	N	32	#	VOLTAGE	CORE	8MS	1	Small voltage droop count for core x	No

### 12.3.2.7 Current Sensors

Sensor Name	SMF Mode?	Master only?	#	Unit	Type	Loc	Sample time	Scale Factor	Description	Call Home
CURVDD	Y	N	1	A	CURRENT	VRM	500uS	0.01	Processor Vdd Current (read from AVSBus by PGPE)	<b>CHOMCURVDDPx</b> (where x is processor 0..7)
CURVDN - NOT SUPPORTED	Y	N	1	A	CURRENT	VRM	500uS	0.01	Processor Vdn Current (read from AVSBus by PGPE)	No
CURVCS	Y	N	1	A	CURRENT	VRM	500uS	0.01	Processor Vcs Current (read from AVSBus by PGPE)	No

### 12.3.3 Other OCC Sensors for AMESTER

This is a list of sensors that are NOT written to main memory but are available via AMESTER.

Sensor Name	#	Unit	Type	Loc	Sample time	Scale Factor	Description	Call Home
TEMPCy	32	C	TEMP	CORE	8MS	1	Average (non-weighted) temperature of the 2 core DTS and 1 L3 DTS for Processor Core y	No
TEMPQy	8	C	TEMP	QUAD		1	1 racetrack DTS for "quad" y	No
TEMPPROCTHRM	1	C	TEMP	PROC	8MS	1	Maximum of all TEMPPROCTHRMCy core temperatures	<b>CHOMTEMPPROCx</b> (where x is processor 0..7)
TEMPPROCAVG	1	C	TEMP	PROC	8MS	1	Average of all TEMPPROCTHRMCy core temperatures	No
TEMPPROCIOxy	4	C	TEMP	PROC	8MS	1	Processor IO ring DTS for location xy 00 = SE PAU 01 = NE PAU 10 = SW PAU 11 = NW PAU	No
TEMPRTAVG	1	C	TEMP	PROC		1	Average of all TEMPQy racetrack temperatures and TEMPNESTx	No
TEMPMEMBUFTHRM	1	C	TEMP	MEM	8MS	1	Hottest Internal memory controller temperature sensor (TEMPMEMBUFx) across all MCs monitored by this OCC	<b>CHOMTEMPMEMBUF Px</b> (where x is processor 0..7)
PROCOTIME	1	ms	TIME	PROC	32MS	1	Consecutive time processor temperature is above ERROR temperature	No
TODclock0	1	us	TIME	SYS	8MS	16	TOD clock Low 16 bits in 16us resolution	No
TODclock1	1	s	TIME	SYS	8MS	1.048576	TOD clock mid 16 bits in 1.05s resolution	No
TODclock2	1	day	TIME	SYS	8MS	0.795364	TOD clock high 3 bits in 0.796 day resolution	No
CUR12VSTBY	1	A	CURRENT	SYS	500uS	.01	12V standby current read from APSS channel (if channel assigned for 12V standby current)	No
VOLTVDNSENSE	1	mV	VOLT	PROC	500us	0.1	Vdn Voltage at the remote sense. (AVS reading adjusted for loadline)	No
CEFFVDDRATIO	1	%	WOF	PROC	500uS	0.01	Raw un-clipped Ceff ratio Vdd	<b>CHOMCEFFRATIOVDD Px</b> (where x is processor 0..7)
CEFFVDNRATIO	1	%	WOF	PROC	500uS	0.01	Raw un-clipped Ceff ratio Vdn	No
VRATIO	1		WOF	PROC	500uS	1	Vratio	No
OCS_ADDR	1	%	WOF	PROC	500uS	0.01	Additional amount added to Ceff ratio Vdd for over current protection	No
CEFFVDDRATIOADJ	1	%	WOF	PROC	500uS	0.01	Final adjusted Ceff ratio Vdd used (includes OCS_ADDR)	No
IO_PWR_PROXY	1	W	WOF	PROC	500uS	0.01	IO Power Proxy read from XGPE produced WOF values	No
UV_AVG	1	%	WOF	PROC	500uS	0.1	Average undervolting read from PGPE produced WOF values	<b>CHOMUVAVGPx</b> (where x is processor 0..7)
OV_AVG	1	%	WOF	PROC	500uS	0.1	Average overvolting read from PGPE produced WOF values	<b>CHOMOVAVGPx</b> (where x is processor 0..7)



---

## 12.4 PIB I2C Master Lock

The OCC will be reading DIMM temperatures (Nimbus) and communicating with GPUs via PIB I2C master engine. The I2C engine will need to be shared with host (OPAL and PHYP). A firmware lock mechanism will be used to handle ownership of each I2C master engine that the OCC uses.

### 12.4.1 OCC Flags Register

The OCC Flags register will be used to indicate OCC ownership for PIB I2C master engines 1, 2 and 3. Two bits are defined per engine: msb is for host ownership, lsb for OCC ownership.

#### NOTES:

- Engines 0...3 are also referred to as engines B...E in documentation
- No bits will be defined for engine 0 since that is used by SBE.
- Bits will be defined for engine 2 even though currently there is no known OCC usage for engine 2, OCC will never take ownership of an engine it does not use.

Bit(s)	Name	Description
16:17	PIB I2C Master Engine 1 Lock	'00' = Nobody has engine 1 lock (default) '01' = OCC has engine 1 lock '10' = Host has engine 1 lock '11' = Host wants engine 1 lock, OCC has it
18:19	PIB I2C Master Engine 2 Lock	'00' = Nobody has engine 2 lock (default) '01' = OCC has engine 2 lock '10' = Host has engine 2 lock '11' = Host wants engine 2 lock, OCC has it
20:21	PIB I2C Master Engine 3 Lock	'00' = Nobody has engine 3 lock (default) '01' = OCC has engine 3 lock '10' = Host has engine 3 lock '11' = Host wants engine 3 lock, OCC has it

### 12.4.2 OCC Miscellaneous Register – Interrupt to host

The OCC miscellaneous register will be used to send an interrupt to host (core\_ext\_intr bit 0) to inform host when OCC has given up an I2C engine. The reason for the interrupt is encoded in bits 1:3 and will be I2C master ownership change. In response to a reason of I2C master ownership change host should read the OCC Flags register to verify they now own.

#### 12.4.2.1 External Interrupt Reason Defines

NOTE: Host must clear core\_ext\_intr bit to allow OCC to send another interrupt. OCC will not write out a new reason and send another interrupt unless the core\_ext\_intr bit is cleared indicating host has processed the previous interrupt. Under current implementation only 1 bit

(i.e. 1 reason) will be set per interrupt.

Bit	Description
1	<b>OCC-HTMGT Service Required Interrupt</b> – host to call HTMGT process_occ_error interface to collect error log with OCC chip ID
2	<b>I2C Master Ownership Change</b> – host to read PIB I2CM engine locks in OCC Flags register
3	<b>OCC Shared Memory Interface Change</b> – Re-read and process changes in dynamic shared memory interface i.e. OCC State, OCC command response, throttle status

### 12.4.3 I2C Lock Use Cases

Typical time for OCC to give up the lock is 4ms which is the rate the OCC will check for lock ownership changes. Host must handle a worst case time to get the lock of 15 seconds for an OCC error condition requiring a reset where the OCC reset procedure is handing over the lock. The OCC will not enforce a time limit for host ownership or for how soon/often they may request a lock again. However, if the OCC times out getting new data requiring an I2C engine the OCC will take appropriate timeout actions specific to the data missing see “Host Hung Case” section for more details.

NOTE: The OCC will never use or take back an i2c engine that host owns, host must clear its ownership bit for the engine when it is done.

#### 12.4.3.1 Host Wants Lock

Note: Initial Condition bits default to b00. Host can use lock whenever by setting b1X and reading back b10 for the engine lock required.

1. Host sets msb of the 2 bits for required engine in the OCC Flags register
2. Host reads 2 bits for required engine from OCC Flags register:
  - 2.1 **b10** → Host uses the bus. When done using the bus, host clears msb of the 2 bits for engine in OCC Flags register.
  - 2.2 **b11** → Host cannot use the bus and must wait for an interrupt from the OCC with reason “I2C Master Ownership Change” defined in OCC Miscellaneous register. When get interrupt go to #2 above.

#### 12.4.3.2 OCC Actions

Approximately every 4ms after OCC confirms that an I2C operation is complete:

1. OCC reads lock bits for each engine it needs from OCC Flags register
  - 1.1. **b00** → OCC takes ownership by setting lsb in OCC Flags register and reads back lock bits following actions defined below
  - 1.2. **b01** → OCC owns and allows jobs requiring the engine to be scheduled
  - 1.3. **b10** → Host owns. OCC will not use.

- 1.4. **b11** → Host wants bus. OCC stops scheduling jobs requiring the engine. OCC clears OCC ownership bit for engine in OCC Flags register and sends “I2C Master Ownership Change” interrupt to host.

#### 12.4.3.3 Host Hung Case

The OCC will not time host ownership and will instead rely on system timeouts to take action due to not being able to use a bus. For example, the OCC will eventually timeout due to not being able to read DIMM temperatures, this timeout is system specific set by the system owner in the def file/MRW. When this timeout occurs, the OCC will log an informational error and will stop resetting the deadman timer to allow memory to throttle to safe mode. This will not cause an OCC reset. NOTE: It is expected that normal operation should never hit this timeout and this error will be made unrecoverable in manufacturing.

#### 12.4.3.4 OCC Hung Case

The FSP or BMC is periodically polling the OCC and will reset the OCC if the communication fails. The OCC reset procedure will clear OCC ownership bit for every engine and send “I2C Master Ownership Change” interrupt. This scenario gives a worst case time of 15 seconds (FSP polling period) to see an OCC is dead and a reset is needed for lock ownership to change.

---

### 12.5 GPU Reset Handling

The OS may assert PERST (via OPAL call) to a GPU at any time for various reasons. To prevent the OCC from logging GPU communication errors due to PERST being asserted OPAL will give indication that a GPU is in reset via OCC flags register bits 22:24. The OCC flags register is per OCC and each OCC can be monitoring a maximum of 3 GPUs.

#### **OCC FLAGS REGISTER**

Bit	Name	Description
22	GPU 0 Reset Status	'0' = GPU0 is in reset (default) '1' = GPU0 is NOT in reset
23	GPU 1 Reset Status	'0' = GPU1 is in reset (default) '1' = GPU1 is NOT in reset
24	GPU 2 Reset Status	'0' = GPU2 is in reset (default) '1' = GPU2 is NOT in reset

#### **NOTES:**

- The OCC will continually attempt communication with all present GPUs in order to detect reset changes
- The default is that the GPU is in reset if OPAL never gives indication that a GPU is NOT in reset the OCC will still be continually attempting to communicate with the GPU, but no errors will ever be logged due to OCC-GPU communication failures.

- On communication failures the OCC will check the GPU reset status bit in the OCC Flags register. If the GPU is indicated to be in reset the OCC will return 0 (not available) for the GPU temperature in the poll response, this is not considered an error case. If the GPU is NOT in reset and the OCC hasn't been able to get a new temperature reading from the GPU for the temperature time out defined in the xml then the OCC will log an error and return 0xFF for temperature in the poll response.

### 12.5.1 GPU Numbering

The GPU sensor IDs are defined in the system xml and sensor to slca index that will be added to the HDAT information to OPAL. The xml must guarantee that the GPU sensors match the GPU presence GPIOs from the APSS. The GPU information must separate out the GPUs (maximum of 3) monitored by each OCC. The OCC Flags register refers to GPU numbers 0, 1, 2 for the 2<sup>nd</sup> OCC these are physical GPUs 3, 4, 5 respectively.

<b>APSS GPIO</b>	<b>OCC FLAGS REGISTER</b>
GPU0	OCC 0 GPU 0 (bit 22)
GPU1	OCC 0 GPU 1 (bit 23)
GPU2*	OCC 0 GPU 2 (bit 24)
GPU3	OCC 1 GPU 0 (bit 22)
GPU4	OCC 1 GPU 1 (bit 23)
GPU5*	OCC 1 GPU 2 (bit 24)

\*On systems that supports a maximum of 4 total GPUs GPU2 and GPU5 (OCC 1 GPU 2) are never present

---

## Appendix A. Return Codes

<b><i>Return Code</i></b>	<b><i>Description</i></b>
<b>0xFF</b>	<b>Command in Progress.</b> Command is being processed and the response buffer is not valid.
<b>0x00</b>	<b>Success.</b> Command completed normally
<b>0x11</b>	<b>Invalid Command.</b> The command type is invalid or unsupported. <ul style="list-style-type: none"><li>• i.e. Slave OCC receiving a command that is supported by master only</li></ul>
<b>0x12</b>	<b>Invalid Command Length.</b> The command data length is invalid for the particular command.
<b>0x13</b>	<b>Invalid Data Field.</b> The command data has an invalid value for a field. <ul style="list-style-type: none"><li>• i.e. Poll version not supported</li></ul>
<b>0x14</b>	<b>Checksum Failure.</b> The command packet checksum is not correct.
<b>0x15</b>	<b>Internal OCC error.</b> An error occurred within OCC to prevent the command from being processed but the OCC is still running, and the command may be retried.
<b>0x16</b>	<b>Present State Prohibits.</b> The OCC cannot execute the command in its present state. <ul style="list-style-type: none"><li>• OCC is not in a state that the command requires</li></ul>
<b>0x17</b>	<b>No Support in SMF.</b> The OCC cannot execute the command when system is in Secure Memory Facility.
<b>0xE0 thru 0xEF</b>	<b>Critical OCC error.</b> The OCC has hit a critical error and cannot run. When possible along with this return status the OCC will include special register info to aid in OCC debug to the response data buffer. Special handling to be done by the sender for all Ex return codes: <ul style="list-style-type: none"><li>• Generate an error log including the full Rsp Data buffer to capture info for debug.</li><li>• Reset all OCCs. NOTE: the OCC is not running, sending any additional commands to this OCC will not be processed and should not be sent until after it is reset.</li></ul> <b>0xE0 → OCC Exception.</b> An Unrecoverable OCC exception. i.e. SSX panic.

<b><i>Return Code</i></b>	<b><i>Description</i></b>
	<p><b>0xE1 → OCC Initialization Checkpoint.</b> Indicates how far into initialization OCC got before it died, typically this will never be seen. Detecting an error during initialization will result in an 0xE5 reason code.</p> <p><b>0xE2 → Watchdog Timeout.</b> Halt due to OCC watchdog expiring.</p> <p><b>0xE3 → OCB Timeout.</b> Halt due to OCB timer expiring.</p> <p><b>0xE4 → Reserved.</b></p> <p><b>0xE5 → OCC Initialization Failure.</b> Halt due to failure during initialization.</p>

---

## Appendix B. OCC States

<b>OCC State</b>	<b>Description</b>
<b>0x00</b>	<b>Reserved.</b> This value is reserved for command data to indicate no change to current OCC state.
<b>0x01</b>	<b>Standby</b> <ul style="list-style-type: none"><li>▪ The OCC is ready to handle commands from HTMGT</li><li>▪ No communication allowed from BMC</li><li>▪ No monitoring or actuation done by OCC</li><li>▪ OCC will default to this state after being loaded and wait for communication from HTMGT to get the needed configuration data to move to observation or active state</li><li>▪ HTMGT will never tell OCC to move to this state</li></ul>
<b>0x02</b>	<b>Observation</b> <ul style="list-style-type: none"><li>▪ Full communication with HTMGT and BMC; some commands may be rejected if only supported in Active state.</li><li>▪ OCC is monitoring only, no DVFS/throttling actuation is done due to power or thermal</li><li>▪ Maximum Pstate clip is set to WOF base</li><li>▪ WOF is disabled</li><li>▪ Pstate protocol is disabled</li></ul>
<b>0x03</b>	<b>Active</b> <ul style="list-style-type: none"><li>▪ This is the full function state</li><li>▪ Full communication with HTMGT and BMC</li><li>▪ OCC will monitor all sensors and actuate to maintain power and thermal limits</li></ul>
<b>0x04</b>	<b>Safe</b> <ul style="list-style-type: none"><li>▪ This is NOT safe mode</li><li>▪ Internally OCC will move to this state when it detects an error and needs to be reset this state will be reflected in the OCC poll response "Current State" byte</li><li>▪ Used for internal OCC usage, HTMGT will not reset based on this, the full safe mode (i.e. OCC reset) will happen via error log processing requesting reset</li><li>▪ This is a state while OCC is waiting for a reset (safe mode)</li><li>▪ Sensor data is not updated while in this state</li><li>▪ OCC will stop poking watchdogs to allow system to drop v/f and memory throttles</li><li>▪ The OCC will continue to communicate with HTMGT and BMC for error logging purposes</li><li>▪ HTMGT will never tell OCC to move to this state</li></ul>

<b><i>OCC State</i></b>	<b><i>Description</i></b>
<b>0x05</b>	<b>Characterization</b> <ul style="list-style-type: none"> <li>▪ OCC treats this the same as observation state – full communication with FSP/BMC, monitoring only</li> <li>▪ OCC sets min/max Pstate clips wide open to allow full frequency range</li> <li>▪ WOF is disabled</li> <li>▪ Pstate protocol is enabled with characterization as PMCR owner</li> <li>▪ Characterization will be writing the PMCR directly to set pStates</li> </ul>



---

## Appendix C. System Power and Performance Modes

User settable System Power and Performance modes are only supported with PowerVM. For all modes the OCC must be in active state.

Mode Value	Description
0x00	<b>Reserved.</b> This value is reserved for command data to indicate no change to current system power and performance mode.
0x01	<b>Power Management Modes Disabled. Previously known as “Nominal”</b> <ul style="list-style-type: none"><li>▪ User Settable</li><li>▪ Frequency and voltages are fixed at a defined point defined in processor VPD</li><li>▪ WOF is off</li><li>▪ IPS may be enabled while in this mode</li><li>▪ OCC may change v/f to maintain a set power cap (system or user) or for thermal reasons. If this happens the OCC will log an error to indicate performance loss.</li></ul>
0x02	<b>Reserved.</b>
0x03	<b>Static Frequency Point. Previously known as “Turbo”</b> <ul style="list-style-type: none"><li>▪ Lab only mode to test a v/f point either VPD or an operating point.</li><li>▪ NOT field settable.</li><li>▪ Frequency and voltages are pinned for the given point. See <a href="#">Frequency Points</a> chapter for defined points that can be set with this mode.</li><li>▪ WOF is off</li><li>▪ OCC may drop from this point due to a power cap or thermal reason. If this happens the OCC will log an error so user (mfg/characterization) knows that system is no longer at the given frequency point. NOTE: OCC will use thermal thresholds that were in place prior to setting this mode. i.e. NO change to thermal thresholds is made when this mode is entered.</li></ul>
0x04	<b>Safe.</b> <ul style="list-style-type: none"><li>▪ NOT user settable</li><li>▪ OCC is non-functional and held in reset. The whole power management complex (all OCCs, PGPEs and XGPEs) is held in reset</li><li>▪ Safe mode is entered after any 3 errors from the same OCC within one hour is detected that causes the OCC to be unable to perform all required functions</li><li>▪ An exit from safe mode will be attempted after a re-IPL, FSP reset or power cycle</li></ul>

Mode Value	Description
0x05	<b>Static Power Save</b> <ul style="list-style-type: none"> <li>▪ User settable</li> <li>▪ Frequency and voltages are fixed to a defined point calculated as a percentage below WOF Base defined in the def file/MRW</li> <li>▪ WOF is off</li> <li>▪ IPS may be enabled while in this mode</li> </ul>
0x06	<b>Reserved.</b>
0x07	<b>Reserved.</b>
0x08	<b>Reserved.</b>
0x09	<b>Fmax</b> <ul style="list-style-type: none"> <li>▪ User settable interface TBD will not be on the main mode GUI.</li> <li>▪ Fmax mode must be supported at the system level enabled by the system owner for this mode to be set</li> <li>▪ OCC pins frequency for each chip to its VPD Fmax frequency point</li> <li>▪ WOF is off</li> <li>▪ IPS may be enabled while in this mode</li> </ul>
0x0A	<b>Dynamic Performance</b> <ul style="list-style-type: none"> <li>▪ User settable</li> <li>▪ OCC lowers frequency if idle for a period of time otherwise ultra turbo frequency</li> <li>▪ WOF is on</li> <li>▪ IPS may be enabled while in this mode</li> </ul>
0x0B	<b>Fixed Frequency Override (FFO)</b> <ul style="list-style-type: none"> <li>▪ User settable interface TBD, will not be on the main mode GUI</li> <li>▪ OCC pins frequency to the user specified FFO frequency</li> <li>▪ Valid FFO frequency range is minimum to ultra turbo. Setting a frequency between UT and Fmax is not supported.</li> <li>▪ WOF is off</li> <li>▪ If (FFO frequency &lt;= WOF base frequency) ==&gt; Log error if throttle due to power/thermal limit so there is indication that system is no longer running at the fixed frequency that was set.</li> <li>▪ If (FFO frequency &gt; WOF base frequency) ==&gt; No error will be logged due to power/thermal throttling i.e. There will be no error log if dropped from a fixed frequency above WOF base.</li> <li>▪ IPS may be enabled while in this mode</li> </ul>
0x0C	<b>Maximum Performance</b> <ul style="list-style-type: none"> <li>▪ User settable</li> <li>▪ Frequency is at ultra turbo as long as power and thermal limits allow.</li> <li>▪ Frequency is NOT lowered due to utilization</li> <li>▪ WOF is on</li> <li>▪ IPS may be enabled while in this mode</li> </ul>



---

## Appendix D. (H)TMGT-OCC Component Ids

Following table is a list of component IDs internal to (H)TMGT-OCC communication. These will be used for error log callouts to cover anything that a Sensor ID does not exist for. All hardware callouts should have a Sensor ID associated with it and use the Sensor ID for a callout, this list should only have things like procedure callouts.

Component ID	Description
0x01	Firmware
0x04	Over temperature – Only used as an error log callout and will result in TMGT adding “OVERTMP” procedure (tells CE to look for airflow blockage, ambient and FRU cooling errors) to the OCC error log.
0x05	Oversubscription Throttling – Error log callout when OCC throttles due to enforcing an oversubscription power cap. TMGT translates this to “TPMD_OV” symbolic FRU (tells CE to look for POWR SRCs first, replace power supply...)
0xFF	None