OCC Firmware Interface Specification for POWER9

Version 0.22

Maintained by: Martha Broyles
mbroyles@us.ibm.com

Change History

Date	Change Description
11/19/2015	*Version 0.1 – First document version*
12/16/2015	*Version 0.2 – Update sensor data format for 4 byte sensor IDs*
02/01/2016	*Version 0.3 – Updates for active state, OPAL interface and reset/code update*
04/13/2016	*Version 0.4 – Update to memory throttle packet and OPAL interface*
05/19/2016	*Version 0.5 – Updates to "TEMP" sensors in poll response, combine frequency data formats to just one, add per chip throttle for each memory throttle point, updates to OPAL interface from review comments.*
08/09/2016	*Version 0.6 – Added AVSBus configuration packet. Updates to OPAL interface to support in band AMESTER and main memory sensors*
09/19/2016	*Version 0.7 – Simics bit update in poll*
10/20/2016	*Version 0.8 – Misc OCC-OPAL updates*
11/11/2016	*Version 0.9 – Command updates*
1/13/2017	*Version 0.10 – Memory power control for IPS. New Observation state for characterization.*
01/25/2017	*Version 0.11 – Power and Performance modes. PGPE interface*
02/10/2017	*Version 0.12 – New mfg test cmd to set per quad Pstate. Updates to IPS memory power control.*
04/19/2017	*Version 0.13 – New mfg test cmd to read generated Pstate table.*
05/31/2017	*Version 0.14 – GPU handling. Support for no APSS*
06/07/2017	*Version 0.15 – In band sensor updates*
06/19/2017	*Version 0.16 – More sensor updates*
07/13/2017	*Version 0.17 – Update sensor times to reflect RTL change to 500us. Add GPU HBM Temperature support. Add WOF error bit to error actions.*
09/12/2017	*Version 0.18 – GPU config updates to support GPU detection without APSS GPIOs. VRM Vdd temperature support.*
10/17/2017	*Version 0.19 – Updates to PowerVM power management*

	modes
03/05/2018	*Version 0.20 – Support for VRM N mode*
04/11/2018	*Version 0.21 – Update AVSbus config for AVSBus current rollover workaround*
05/01/2018	*Version 0.22 – Add GPU presence to OPAL shared memory interface*

Copyright and Disclaimer

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®.

Table of Contents

1 OVERVIEW 9
1.1 TERMS 9
1.2 OCC FUNCTIONAL OVERVIEW 10
1.3 OCB CHANNELS 10
1.4 LINUX/OPAL REQUIREMENTS 11
1.5 ATTENTIONS/INTERRUPTS 11
1.5.1 To OCC 11
1.5.1.1 Command Write Attention Type = 0x01 11
1.5.2 OCC to Host (BMC system only) 12
1.5.3 OCC to FSP (TMGT) 12
1.5.4 BMC to Host 12
1.6 COMMANDS 13
1.6.1 Command and Response Buffer Locations 13
1.6.2 OCC Command/Response Sequence 14
1.6.3 Command Format 15
1.6.4 Response Format 16
1.6.5 Error Response Data Packet 17
1.6.6 Command Summary Table 18
2 COMMAND DEFINITIONS 21
2.1 POLL 22
2.1.1 Version 0x20 Poll Response Data Definition 24
2.2 CLEAR ERROR LOG 30
2.3 SET MODE AND STATE 32
2.4 CONFIGURATION DATA 34
2.4.1 Frequency Points (Format = 0x02) 36
2.4.2 Set OCC Role (Format = 0x03) 38
2.4.3 APSS Configuration (Format = 0x04) 39
2.4.4 Memory Configuration (Format = 0x05) 40
2.4.5 Power Cap Values Data Packet (Format = 0x07) 42
2.4.6 System Configuration (Format = 0x0F) 43
2.4.7 Idle Power Saver Settings (Format = 0x11) 47
2.4.8 Memory Throttling (Format = 0x12) 48
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 VRM Fault (Format = 0x21) 56
2.4.13 Setup Configuration Data Return Packet 57
2.5 SET USER POWER CAP 58
2.6 RESET PREP 60
2.7 SNAPSHOT SYNC 62
2.8 GET OCC SNAPSHOT BUFFER 64
2.9 DEBUG PASS THROUGH 66
2.10 AME PASS THROUGH 68
2.11 GET FIELD DEBUG DATA 70

2.12 MFG TEST COMMAND 72
2.12.1 Run/Stop Slew Between Two Modes (Sub Cmd = 0x02) *74*
2.12.2 List Sensors (Sub Cmd = 0x05) *75*
2.12.3 Get Sensor Information (Sub Cmd = 0x06) *77*
2.12.4 Enable/Disable Oversubscription Emulation (Sub Cmd = 0x07) *78*
2.12.5 Run/Stop Memory Slew Between 1-100 Percent (Sub Cmd = 0x09) *79*
2.12.6 Request Pstate per Quad (Sub Cmd = 0x0A) *80*
2.12.7 Read Generated Pstate Table (Sub Cmd = 0x0B) *81*
2.13 TUNABLE PARAMETERS COMMAND 82
2.13.1 Tunable Parameters List *84*
2.13.1.1 Tunable Parameters Unit Types *84*
2.13.1.2 Tunable Parameters Flags *85*
2.13.2 Sequence Diagram of Initial Exchange to Populate Parameters *85*
3 ERROR HANDLING 86
3.1 OCC ERRORS 86
3.1.1 Read OCC Error Log from SRAM *87*
3.2 ERRORS REQUIRING OCC RESET 90
3.3 OCC RESET PROCEDURE (SAFE MODE) 91
3.4 ERROR SCENARIOS 92
3.4.1 (H)TMGT-OCC Communication Failure *92*
3.4.2 BMC-OCC Communication Failure *92*
3.4.2.1 BMC-OCC Communication Failure Handling Flow *93*
3.4.3 OCC Fails to Load or Fails to go Active *94*
3.4.4 Checkstop *94*
3.4.5 OCC Detects an Error Requiring Reset *94*
4 OCC BOOT AND CODE UPDATE PROCESS 96
4.1 FSP SYSTEM BOOT PROCESS WITH PHYP 96
4.2 FSP OCC RESET WITH PHYP 96
4.3 FSP OCC RESET FAILS WITH PHYP 97
4.4 FSP CONCURRENT CODE UPDATE WITH PHYP AND OCCS ACTIVE 97
4.5 FSP CONCURRENT CODE UPDATE WITH PHYP AND PM IN RESET 98
4.6 OCC LOAD/START PROCESS ON FSP WITH OPAL 98
4.7 OCC LOAD/START PROCESS ON FSP WITH AVP 99
4.8 OCC LOAD/START PROCESS ON BMC WITH PHYP 99
4.9 OCC LOAD/START PROCESS ON BMC WITH OPAL 99
4.10 BMC ACTIVATE OCCS PROCEDURE 100
5 FREQUENCY POINTS 101
5.1 VPD 101
5.2 CONFIGURATION FILE 102
6 OCC POLL RESPONSE SENSOR DATA FORMAT DEFINITIONS 103
6.1 TEMPERATURE SENSORS (“TEMP”) 103
6.2 FREQUENCY SENSORS (“FREQ”) 104
6.3 POWER SENSORS (“POWR”) 105
6.3.1 System has APSS (Master only) *105*
6.3.2 No APSS (All OCCs Report) *106*
6.4 POWER CAPS (“CAPS”) 107
6.5 EXTENDED OCC DATA (“EXTN”) 108

6.5.1 Extended OCC Sensors List *108*
7 OCC TO PGPE COMMUNICATION **110**
7.1 IPC COMMANDS OCC TO PGPE 110
7.1.1 Start/Stop Pstate Protocol *111*
7.1.2 Pstate Clip Update *112*
7.1.3 Set PMCR *113*
7.1.4 WOF Control *114*
7.1.5 WOF VFRT *115*
8 POWER MANAGEMENT **116**
8.1 XML SETTINGS (BMC) 116
8.2 POWER MANAGEMENT SETTINGS (FSP) 118
8.3 POWERVM SYSTEM POWER AND PERFORMANCE MODES 118
8.4 USER POWER CAPPING 118
8.5 IDLE POWER SAVER 120
8.5.1 FSP Systems *120*
9 CHECKSTOP FIR COLLECTION **122**
10 MANUFACTURING IMPACTS **123**
10.1 MFG TEST COMMANDS 123
10.1.1 Processor Auto-Slew with OPAL *123*
10.2 PSTATE TABLE BIAS 123
10.3 ENABLE/DISABLE OCC CONTROL 124
10.3.1 Observation State (Disable OCC) Change Process *124*
10.3.2 Characterization State Change Process *124*
10.3.3 Active State (Enable OCC) Change Process *124*
10.4 EXTERNAL VOLTAGE AND FREQUENCY BIAS 124
10.4.1 Writing Voltage *125*
11 OCC-OPAL INTERFACE **126**
11.1 OCC-OPAL SHARED MEMORY INTERFACE 126
11.2 OPAL-OCC COMMAND/RESPONSE INTERFACE 130
11.2.1 OPAL-OCC Command Buffer *130*
11.2.2 OPAL-OCC Response Buffer *131*
11.2.3 OPAL-OCC Command/Response Sequence *132*
11.2.4 Error Handling *133*
11.2.4.1 Timeout for Command Processing *133*
11.2.4.2 Response Failures *133*
11.2.5 OPAL-OCC Commands *134*
11.2.5.1 AMESTER Pass Thru *135*
11.2.5.2 Clear Sensor Data *138*
11.2.5.3 Set Power Cap in Band *139*
11.2.5.4 Write Power Shifting Ratio *140*
11.2.5.5 Select Sensor Groups *141*
11.3 OCC MAIN MEMORY SENSOR DATA 142
11.3.1 OCC N Sensor Data Block Layout (150kB) *142*
11.3.1.1 Sensor Data Header Block (1kB) *143*
11.3.1.2 Sensor Names (50kB) *144*
11.3.1.3 Sensor Readings Ping and Pong Buffers (40kB each) *146*
11.3.1.3.1 sensor_structure_version = 0x01 (Full Reading) *147*

11.3.1.3.2 sensor_structure_version = 0x02 (Counter) *147*
11.3.2 Main Memory OCC Sensor List *148*
11.3.2.1 Performance Sensors *148*
11.3.2.2 Power Sensors *149*
11.3.2.3 Frequency Sensors *149*
11.3.2.4 Utilization Sensors *149*
11.3.2.5 Temperature Sensors *149*
11.3.2.6 Voltage Sensors *150*
11.3.2.7 Current Sensors *150*
11.3.3 Other OCC Sensors for AMESTER *151*
11.4 PIB I2C MASTER LOCK 153
11.4.1 OCC Flags Register *153*
11.4.2 OCC Miscellaneous Register – Interrupt to host *153*
11.4.2.1 External Interrupt Reason Defines *153*
11.4.3 I2C Lock Use Cases *154*
11.4.3.1 Host Wants Lock *154*
11.4.3.2 OCC Actions *154*
11.4.3.3 Host Hung Case *155*
11.4.3.4 OCC Hung Case *155*
11.5 GPU RESET HANDLING 155
11.5.1 GPU Numbering *156*
APPENDIX A. RETURN CODES 157
APPENDIX B. OCC STATES 159
APPENDIX C. SYSTEM POWER AND PERFORMANCE MODES 161
APPENDIX D. (H)TMGT-OCC COMPONENT IDS 163
APPENDIX E. OTHER BMC REQUIREMENTS 164
E.1 USER INTERFACE TO SEND AN OCC COMMAND 164

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 512K SRAM. Provide real time power and thermal monitoring. Monitoring times to allow for fast response, power read every 250us, processor and memory temperatures read every 2ms.

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 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.

1.4 Linux/OPAL Requirements

The following are required from Linux/OPAL to support OCC.

- Support for OCC-Host Interrupt. See OCC to Host Interrupt section for details.
- Support for OCC reset request. See BMC Request for OCC Reset section for details.
- Support to update “OCC Active Sensor”. See Activate OCCs Procedure and OCC Reset Procedure sections for details.
- Interface for manufacturing to disable and enable OCC. See Enable/Disable OCC Control section for details.
- Update PNOR ownership to indicate if OCC is allowed to write to PNOR when PNOR ownership is given/taken from BMC. See Checkstop FIR Collection section for details.
- 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.
- Full support for the OCC-OPAL interface.

1.5 Attentions/Interrupts

1.5.1 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:

Byte 1	***Byte 2***	***Bytes 3 thru 8***
Sender ID	Attn Type	Cmd Specific Data

1.5.1.1 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:

Byte 1	***Byte 2***	***Bytes 3 thru 8***
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

1.5.2 OCC to Host (BMC system only)

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 of OCC-HTMGT Service Required
2. The interrupt is controlled by the XIVR – OCC register that Linux/OPAL must have previously setup. NOTE: OCC is running before Linux, when setup is complete there may be an interrupt already pending that must be handled by Linux.
3. Linux/OPAL sees the interrupt and recognizes the reason of OCC-HTMGT Service Required and calls HTMGT interface to process with the OCC chip ID that generated the interrupt.
4. Linux/OPAL clears the SCOM bits so that OCC can generate an interrupt again as needed.

1.5.3 OCC to FSP (TMGT)

In P9 there will be 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.

1.5.4 BMC to Host

The P9 chip can be interrupted by the OEM defined SMS alert. 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.

1.6 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.

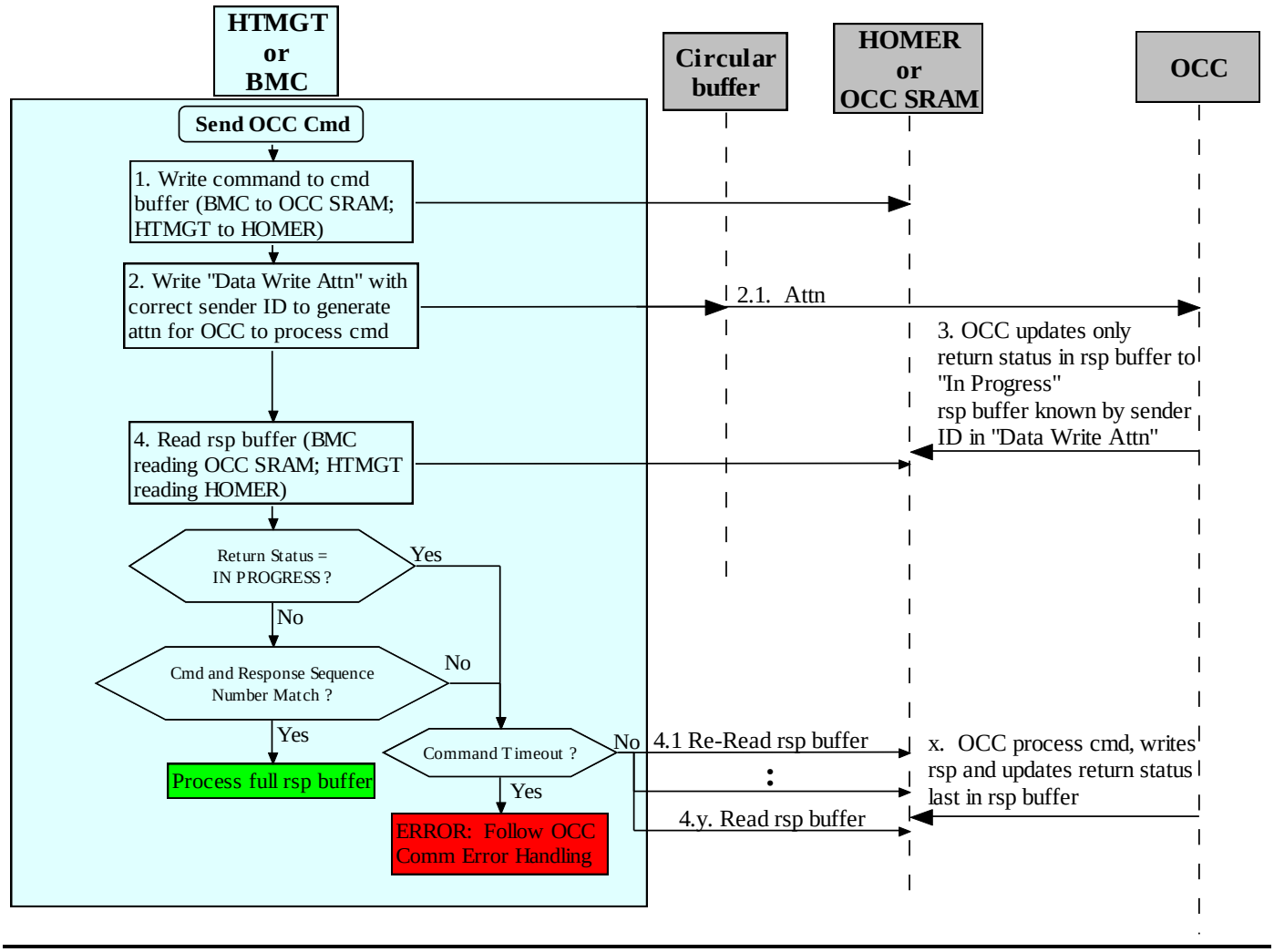
1.6.1 Command and Response Buffer Locations

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

	Command Buffer	***Response Buffer***
BMC / FSP (TMGT)	OCC SRAM 0xFFFBE000 – 0xFFFBEFFF	OCC SRAM 0xFFFBF000 – 0xFFFBFFFF
HTMGT	HOMER 0x000E0000 – 0x000E0FFF	HOMER 0x000E1000 – 0x000E1FFF

1.6.2 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.



1.6.3 Command Format

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

1.6.4 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

1.6.5 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:

Sequence Number	xx
Command Type	xx
Return Status	Non-Success See Appendix A for list of all non-successful return codes.
Data Length	0x0001
Data	There is 1 data byte returned: **Byte 1:** 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.
Checksum	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.

1.6.6 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/Characterizaion
  - 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

Command		***Sender***	***Timeout***		***OCC State***	***Supported By***	
			FSP	***BMC***		***Master OCC***	***Slave OCC***
Poll	Poll the OCC for status and sensor data.	BMC HTMGT TMGT	5s	20s	Sby, O, A, Safe	Y	Y
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
Set	Set the OCC state	HTMGT	15s	20s	Sby, O, A	Y	N

Command		Sender	Timeout		OCC State	Supported By	
			FSP	BMC		Master OCC	Slave OCC
Mode and State	and/or system power management mode	TMGT					
Config Data	0x02: Frequency Points	HTMGT TMGT	15s	20s	Sby, O	Y	N
	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	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
Set User Cap	Set a User Power Cap	BMC HTMGT TMGT	5s	20s	Sby, O, A	Y	N
Reset Prep	Prepare OCC to be reset	HTMGT TMGT	15s	20s	Sby, O, A, Safe	Y	Y
Snapshot Sync	Delete all snapshot buffers and start a new snapshot from 0. OCC collects output power from APSS FSP collects input power from power supplies as it does in P7 (TMGT collects from SPCN reading from DEVROS). Need to re-sync to keep AC and DC power in approximate same 30s time slice.	TMGT	5s	N/A	O, A	Y	N
Get	Get Snapshot buffer.	TMGT	5s	N/A	O, A	Y	N

Command		Sender	Timeout		OCC State	Supported By	
			FSP	BMC		Master OCC	Slave OCC
OCC Snapshot Buffer	Needed for CIM trending support.						
Debug Pass Through	Used for debug use only.	HTMGT TMGT	15s	20s	Sby, O, A, Safe	Y	Y
AME Pass Through	Used for debug use only by amester.	HTMGT TMGT	10s	20s	Sby, O, A, Safe	Y	Y
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
Mfg Test Cmd	Send a manufacturing test command	HTMGT TMGT	15s	20s	O : some sub cmds A : all	Y	Y (not all sub cmds)
Tunable Parameters	Query or set a tunable parameter	TMGT	5s	N/A	Sby, O, A	Y	N

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.

TMGT	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 getting a “service required” attention from the OCC.
BMC	The BMC will send this periodically to read sensor data and verify that the OCC is functional.
HTMGT	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:

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

Poll Return Packet:

<table>
<tr><td>Sequence Number</td><td>Xx</td></tr>
<tr><td>Command Type</td><td>0x00</td></tr>
<tr><td>Return Status</td><td>0x00 = Success
See Appendix A for list of all non-successful return codes.</td></tr>
<tr><td>Data Length</td><td>Variable. Minimum of 40 bytes to maximum of 4089.</td></tr>
<tr><td>Data</td><td><u>Version = 0x20</u> See Version 0x20 Poll Response Data Definition section for details.
<table>
<tr><th>Byte 1</th><th>Byte 2</th><th>Byte 3</th><th>Byte 4</th><th>Byte 5</th><th>Byte 6</th><th>Byte 7</th><th>Byte 8</th></tr>
<tr><td>Status</td><td>Ext Status</td><td>OCCs present</td><td>Config data</td><td>OCC State</td><td>Mode</td><td>IPS Status</td><td>Error Log ID</td></tr>
<tr><td colspan="4">Error Log Start Address</td><td colspan="2">Error Log Length</td><td>Reserved</td><td>GPU config</td></tr>
<tr><td colspan="8">OCC Code Level (16 bytes)</td></tr>
<tr><td colspan="6">“SENSOR”
(bytes 33-38)</td><td># of sensor data blocks</td><td>Sensor data block header version = 0x01</td></tr>
<tr><td colspan="8">Sensor Data Blocks
(Variable length. Byte 41 thru end of response data length)</td></tr>
</table></td></tr>
<tr><td>Checksum</td><td>Xxxx</td></tr>
</table>

2.1.1 Version 0x20 Poll Response Data Definition

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

Bit 7 (msb)	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0 (lsb)
Master OCC	Collect FIR Data		OCC PMCR Owner	SIMICS		OCC Observation 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 is the master to process master only data/commands from.

Collect FIR Data – BMC systems only. 1 indicates that this OCC will be monitoring for checkstops and collect additional FIR SCOM data on a checkstop see Checkstop FIR Collection section for more details.

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

DVFS due to proc OT – 1 indicates that the OCC has currently clipped max Pstate below turbo 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 turbo 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 an over temperature condition (DIMM or Centaur).

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 turbo 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

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: 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 – FSP system only, this byte will be 0x00 for BMC based systems. See Idle Power Saver Handling chapter for info on how these bits are used by TMGT.

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

Bits 7:2 Reserved '000000'

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

Bit 0: Idle Power Saver Enabled – This should reflect what TMGT 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 1) is ignored by TMGT.

Byte 8: Error Log Id – Any non-zero value indicates the log id associated with an OCC error log to be reported. 0x00 indicates no error log. There must also be an error log start address and error log length in the same poll response. The same error log id will be sent until a clear error log command has been sent for the error log id. HTMGT must handle this and not log the same error more than once.

HTMGT/TMGT Handling: Full support to collect and report OCC error log

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 OCC error log

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 OCC error log

BMC Handling: None

Byte 15: Reserved = 0x00

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

Bit 7 (msb)	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0 (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 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.

BMC	Should never send.
(H)TMGT	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:

Sequence Number	xx
Command Type	0x12
Data Length	0x0001
Data	There is 1 data byte: **Byte 1:** Error Log Id – The log id of the error log to be cleared.
Checksum	xxxx

Clear Error Log Return Packet:

Sequence Number	xx
Command Type	0x12
Return Status	0x00 = Success See Appendix A for list of all non-successful return codes.
Data Length	0x0000
Data	There is no data returned.
Checksum	xxxx

2.3 Set Mode and State

BMC	Should never send.
(H)TMGT	(H)TMGT will send this as part of booting the OCCs or when a state change request is made for manufacturing testing.

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:

Sequence Number	Xx
Command Type	0x20
Data Length	0x0003
Data	There are 3 bytes of data: **Byte 1:** Version = 0x00 **Byte 2:** OCC State – 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. **Byte 3:** Current System Power Management Mode – 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.
Checksum	Xxxx

Set Mode and State Return Packet:

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

2.4 Configuration Data

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

BMC	Should never send.
(H)TMGT	(H)TMGT will send this as part of booting the OCCs

Configuration Data Command Packet:

Sequence Number	Xx
Command Type	0x21
Data Length	Dependent on Data Being Sent. See following sections for more details specific to each format.
Data	There are x bytes of data (not to exceed maximum): **Byte 1:** 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:

	Master OCC	***Slave OCC***
0x02: Frequency	Required for active sent from (H)TMGT	Required for active **from master OCC**
0x03: OCC Role	Required for observation sent from (H)TMGT	
0x04: APSS Cfg	Required for observation sent from (H)TMGT	
0x05: Memory Cfg	Required for observation sent from (H)TMGT	
0x07: Pcap	Required for active sent from (H)TMGT	Required for active **from master OCC**
0x0F: System Cfg	Required for observation sent from (H)TMGT	
0x11: IPS	Not required, optional for active state from TMGT	Not supported (IPS data is not used by slaves)
0x12: Mem Throttle	Optional for active state sent from (H)TMGT. Special handling to determine if required based on memory config data packet.	
0x13: Thermal Control Thresholds	Required for observation sent from (H)TMGT	
0x14: AVSBus Cfg	Required for observation sent from (H)TMGT	
0x15: GPU	Not required. Sent from HTMGT	
0x21: VRM Fault	Optional for all states.	

Checksum	Xxxx

2.4.1 Frequency Points (Format = 0x02)

This gives the minimum and maximum frequency range and operating points for static modes.

Data Length	0x000E (version 0x20) or 0x0012 (version 0x21)
Data	**Byte 1:** Format = 0x02 **Byte 2:** Version = 0x20 or 0x21 **<u>Version 0x20 and 0x21</u>** **Bytes 3-4: Nominal Frequency Point**. In MHz; MSB first. (H)TMGT reads from ATTR_NOMINAL_FREQ_MHZ **Bytes 5-6: Turbo Frequency Point.** In MHz; MSB first. Highest frequency allowed in the absence of ultra turbo support. (H)TMGT determines if frequencies above nominal (aka Turbo) are supported or not: BMC systems: OPEN_POWER_TURBO_MODE_SUPPORTED from MRW FSP systems: Def file TURBO_MAX If Turbo is supported: BMC systems: HTMGT reads and sends ATTR_FREQ_CORE_MAX (VPD max frequency) to the OCC FSP systems: Calculated by TMGT as the smaller of [def file TURBO_MAX] and [ATTR_FREQ_CORE_MAX (VPD limit)] If Turbo is not supported then Nominal Frequency Point will be sent to limit the maximum frequency to nominal. **Bytes 7-8: Minimum Frequency Point.** In MHz; MSB first. Lowest frequency ever allowed. This is the lowest that the OCC may ever drop to for any reason. (H)TMGT reads from ATTR_MIN_FREQ_MHZ **Bytes 9-10: Ultra Turbo Frequency Point.** In MHz; MSB first. Highest frequency allowed for WOF (Workload Optimized Frequency) read from ATTR_ULTRA_TURBO_FREQ_MHZ. Values reserved to indicate WOF is NOT supported and reason why: • **0x0000 -** WOF is not supported due to missing ultra turbo VPD. HB writes ATTR_ULTRA_TURBO_FREQ_MHZ and will be 0 if any chip is missing the ultra turbo VPD point. • **0x0001 –** WOF is not supported due to system MRW having WOF disabled (ATTR_SYSTEM_WOF_DISABLE) • **0x0002 –** WOF is not supported due to OCC reset due to WOF error (WOF Retry count) being reached • **0x0003 –** WOF is not supported due to frequencies above nominal not supported (nominal was sent for turbo frequency point) **Bytes 11-12: Static Power Save Frequency Point.** In MHz; MSB first. Frequency to use when system is in Static Power Saver mode. Only valid

Data Length	0x000E (version 0x20) or 0x0012 (version 0x21)
	with PowerVM. (H)TMGT will send 0x0000 on systems with OPAL to indicate no support for static power saver mode. BMC Systems: HTMGT calculates from ATTR_PM_MODE_FREQ_PERCENT if ATTR_PM_MODE is set to Static Power Save. If ATTR_PM_MODE is not SPS HTMGT will send 0x0000 to indicate no support for SPS mode. FSP Systems: ATTR_POWERSAVE_FREQ_MHZ, written by TMGT as the maximum of def file STATIC_POWER_SAVE and ATTR_MIN_FREQ_MHZ **Bytes 13-14: FFO Frequency Point.** In MHz; MSB first. Frequency to use when system is in FFO mode. Only valid with PowerVM. (H)TMGT will send 0x0000 on systems with OPAL to indicate no support for FFO mode. BMC Systems: HTMGT calculates from ATTR_PM_MODE_FREQ_PERCENT if ATTR_PM_MODE is set to FFO. If ATTR_PM_MODE is not FFO HTMGT will send 0x0000 to indicate no support for FFO mode. FSP Systems: User frequency set via CIM. TMGT ensures frequency is within range and then writes to ATTR_FFO_FREQ_MHZ to send to OCC. **<u>Version 0x21 Only</u>** **Bytes 15-16: Oversubscription Maximum Frequency**. In MHz; MSB first. Maximum frequency allowed when system is in oversubscription. 0 indicates no frequency limit for oversubscription **Bytes 17-18: VRM N mode Maximum Frequency**. In MHz; MSB first. Maximum frequency allowed when OCC is notified from POWR of a VRM phase fault. 0 indicates no frequency limit for phase faults.

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.

Data Length	0x0004
Data	**Byte 1:** Format = 0x03 **Byte 2: OCC Role** **0x00** – Slave **0x01** – Master **0x02** – 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. **All other values reserved.** **Bytes 3-4:** 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.

<table>
<tr><td>Data Length</td><td>Variable based on Type:
Type 0x00 (APSS): 0x00F8
Type 0x02 (2 channel chip): 0x0020
Type 0xFF (none): 0x0004</td></tr>
<tr><td>Data</td><td>Byte 1: Format = 0x04
Byte 2: Version = 0x20
Byte 3: Type. Indicates type of SPI attached analog-to-digital converter is present for power readings
0x00 APSS – 16 channel and GPIO config follows
0x02 2 channel SPI attached chip (i.e. TI ADC122S021) – 2 channels (ADC channel 0 and 1) will follow. No GPIOs.
0xFF None – No SPI attached chip. Power capping will not be supported by the OCC.
Byte 4: Reserved = 0x00
Format of one ADC channel info data set:
<table>
<tr><td>Data byte x</td><td>ADC Channel Assignment. Enum. "Function ID" in MRW</td></tr>
<tr><td>Data byte x+1 thru x+4</td><td>Sensor ID. 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>Data byte x+5</td><td>Ground Select.</td></tr>
<tr><td>Data byte x+6 thru x+9</td><td>Gain.
Float.</td></tr>
<tr><td>Data byte x+10 thru x+13</td><td>Offset.
Float.</td></tr>
</table>
Bytes 5-18: ADC Channel 0 Info data set
Bytes 19-32: ADC Channel 1 Info data set
Bytes 33-46: ADC Channel 2 Info data set
:
Bytes 201-214: ADC Channel 14 Info data set
Bytes 215-228: ADC Channel 15 Info data set

Byte 229: GPIO Port 0 Mode In MRW unit-type= "gpio-global" id= GPIO_P0_MODE
Byte 230: GPIO Port 0 Reserved = 0x00
Bytes 231-238: GPIO Port 0 Pin[y] Assignment (enum) y=0-7 (8 pins)
Byte 239: GPIO Port 1 Mode In MRW unit-type= "gpio-global" id= GPIO_P1_MODE
Byte 240: GPIO Port 1 Reserved = 0x00
Bytes 241-248: GPIO Port 1 Pin[y] Assignment (enum) y=0-7 (8 pins)</td></tr>
</table>

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 Centaur 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.

<table>
<tr><td>Data Length</td><td>Variable</td></tr>
<tr><td>Data</td><td>
Byte 1: Format = 0x05
Byte 2: Version = 0x21
Byte 3: 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
Byte 4: 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)
0x03 = Power Down and STR Clock Stop
0xFF = Not supported
Byte 5: 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.

<u>Version 0x21</u>
The format of each set is:
<table>
<tr><td>Data bytes x thru x+3</td><td>Hardware Sensor ID. Sensor ID to use for calling out DIMM or Centaur error</td></tr>
<tr><td>Data bytes x+4 thru x+7</td><td>Temperature Sensor ID. Sensor ID to use for reporting DIMM or Centaur temperature in poll response. If not defined send Hardware sensor ID.</td></tr>
</table>
</td></tr>
</table>

Data Length	Variable	
	Data byte x+8	**Memory Type.** This will indicate what the remaining data represents. • ***0xFF = Nimbus:*** No centaurs. • ***0x00 – 0x07 = Cumulus:*** Centaur #. Value 0-7 that indicates physical location of Centaur.
	Data byte x+9	**DIMM Info Byte 1.** • *Nimbus:* PIB I2C Master Engine for the DIMM (“E” = 3) • *Cumulus:* DIMM # (0-7) that indicates physical DIMM location behind Centaur. 0xFF indicates the HW and Temperature Sensor IDs are for the Centaur itself.
	Data byte x+10	**DIMM Info Byte 2.** • *Nimbus:* DIMM I2C Port (0 or 1) • *Cumulus:* Reserved. 0x00
	Data byte x+11	**DIMM Info Byte 3.** • *Nimbus:* DIMM Temperature I2C Address • *Cumulus:* Reserved. 0x00

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

Data Length	0x000A
Data	**Byte 1:** Format = 0x07 **Version 0x20:** **Byte 2:** Version = 0x20 **Bytes 3 & 4: Minimum soft power cap.** 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. **Bytes 5 & 6: Minimum hard power cap**. In 1W units (MSB sent first), the lowest power cap that a user may set and is guaranteed to be held via processor DVFS. **Bytes 7 & 8: System Maximum power cap**. 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. **Bytes 9 & 10: Quick Power Drop power cap.** 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.

NOTES:

1. 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.
2. All power cap values are DC (output) power
3. 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.

<table>
<tr><td>Data Length</td><td>Version 0x20: 0x00CF Version 0x21: 0x00D7</td></tr>
<tr><td>Data</td><td>
Byte 1: Format = 0x0F
Byte 2: Version = 0x20, 0x21
Byte 3: General System Type (bit defined)
<table>
<tr><th>Bit 7 (msb)</th><th>Bit 6</th><th>Bit 5</th><th>Bit 4</th><th>Bit 3</th><th>Bit 2</th><th>Bit 1</th><th>Bit 0 (lsb)</th></tr>
<tr><td>KVM</td><td></td><td></td><td></td><td>Report DVFS below nominal</td><td></td><td>Non redundant ps</td><td>Single Node</td></tr>
</table>
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.

Report DVFS Below Nominal – Indicates if OCC should wait to report DVFS due to power/thermal until max frequency has been lowered below nominal. Default OCC will report DVFS when max frequency is lowered below turbo. DVFS is reported in poll response extended status bits and in OPAL shared memory throttle status byte. Comes from xml attribute REPORT_THROTTLE_BELOW_NOMINAL
‘1’ = OCC will report throttling only when max frequency has been lowered below nominal
‘0’ = OCC will report throttling when max frequency has been lowered below turbo

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: Core 0 Temperature Sensor ID – Sensor ID for physical core 0, used by OCC to report core 0 temperature
Bytes 12-15: Core 0 Frequency Sensor ID – Sensor ID for physical core 0,
</td></tr>
</table>

	used by OCC to report core 0 frequency **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 1 Frequency Sensor ID – Sensor ID for physical core 1, used by OCC to report core 1 frequency **Bytes 24-27:** Core 2 Temperature Sensor ID – Sensor ID for physical core 2, used by OCC to report core 2 temperature **Bytes 28-31:** Core 2 Frequency Sensor ID – Sensor ID for physical core 2, used by OCC to report core 2 frequency **Bytes 32-35:** Core 3 Temperature Sensor ID – Sensor ID for physical core 3, used by OCC to report core 3 temperature **Bytes 36-39:** Core 3 Frequency Sensor ID – Sensor ID for physical core 3, used by OCC to report core 3 frequency **Bytes 40-43:** Core 4 Temperature Sensor ID – Sensor ID for physical core 4, used by OCC to report core 4 temperature **Bytes 44-47:** Core 4 Frequency Sensor ID – Sensor ID for physical core 4, used by OCC to report core 4 frequency **Bytes 48-51:** Core 5 Temperature Sensor ID – Sensor ID for physical core 5, used by OCC to report core 5 temperature **Bytes 52-55:** Core 5 Frequency Sensor ID – Sensor ID for physical core 5, used by OCC to report core 5 frequency **Bytes 56-59:** Core 6 Temperature Sensor ID – Sensor ID for physical core 6, used by OCC to report core 6 temperature **Bytes 60-63:** Core 6 Frequency Sensor ID – Sensor ID for physical core 6, used by OCC to report core 6 frequency **Bytes 64-67:** Core 7 Temperature Sensor ID – Sensor ID for physical core 7, used by OCC to report core 7 temperature **Bytes 68-71:** Core 7 Frequency Sensor ID – Sensor ID for physical core 7, used by OCC to report core 7 frequency **Bytes 72-75:** Core 8 Temperature Sensor ID – Sensor ID for physical core 8, used by OCC to report core 8 temperature **Bytes 76-79:** Core 8 Frequency Sensor ID – Sensor ID for physical core 8, used by OCC to report core 8 frequency **Bytes 80-83:** Core 9 Temperature Sensor ID – Sensor ID for physical core 9, used by OCC to report core 9 temperature **Bytes 84-87:** Core 9 Frequency Sensor ID – Sensor ID for physical core 9, used by OCC to report core 9 frequency **Bytes 88-91:** Core 10 Temperature Sensor ID – Sensor ID for physical core 10, used by OCC to report core 10 temperature **Bytes 92-95:** Core 10 Frequency Sensor ID – Sensor ID for physical core 10, used by OCC to report core 10 frequency **Bytes 96-99:** Core 11 Temperature Sensor ID – Sensor ID for physical core 11, used by OCC to report core 11 temperature **Bytes 100-103:** Core 11 Frequency Sensor ID – Sensor ID for physical core 11, used by OCC to report core 11 frequency **Bytes 104-107:** Core 12 Temperature Sensor ID – Sensor ID for physical

core 12, used by OCC to report core 12 temperature
Bytes 108-111: Core 12 Frequency Sensor ID – Sensor ID for physical core 12, used by OCC to report core 12 frequency
Bytes 112-115: Core 13 Temperature Sensor ID – Sensor ID for physical core 13, used by OCC to report core 13 temperature
Bytes 116-119: Core 13 Frequency Sensor ID – Sensor ID for physical core 13, used by OCC to report core 13 frequency
Bytes 120-123: Core 14 Temperature Sensor ID – Sensor ID for physical core 14, used by OCC to report core 14 temperature
Bytes 124-127: Core 14 Frequency Sensor ID – Sensor ID for physical core 14, used by OCC to report core 14 frequency
Bytes 128-131: Core 15 Temperature Sensor ID – Sensor ID for physical core 15, used by OCC to report core 15 temperature
Bytes 132-135: Core 15 Frequency Sensor ID – Sensor ID for physical core 15, used by OCC to report core 15 frequency
Bytes 136-139: Core 16 Temperature Sensor ID – Sensor ID for physical core 16, used by OCC to report core 16 temperature
Bytes 140-143: Core 16 Frequency Sensor ID – Sensor ID for physical core 16, used by OCC to report core 16 frequency
Bytes 144-147: Core 17 Temperature Sensor ID – Sensor ID for physical core 17, used by OCC to report core 17 temperature
Bytes 148-151: Core 17 Frequency Sensor ID – Sensor ID for physical core 17, used by OCC to report core 17 frequency
Bytes 152-155: Core 18 Temperature Sensor ID – Sensor ID for physical core 18, used by OCC to report core 18 temperature
Bytes 156-159: Core 18 Frequency Sensor ID – Sensor ID for physical core 18, used by OCC to report core 18 frequency
Bytes 160-163: Core 19 Temperature Sensor ID – Sensor ID for physical core 19, used by OCC to report core 19 temperature
Bytes 164-167: Core 19 Frequency Sensor ID – Sensor ID for physical core 19, used by OCC to report core 19 frequency
Bytes 168-171: Core 20 Temperature Sensor ID – Sensor ID for physical core 20, used by OCC to report core 20 temperature
Bytes 172-175: Core 20 Frequency Sensor ID – Sensor ID for physical core 20, used by OCC to report core 20 frequency
Bytes 176-179: Core 21 Temperature Sensor ID – Sensor ID for physical core 21, used by OCC to report core 21 temperature
Bytes 180-183: Core 21 Frequency Sensor ID – Sensor ID for physical core 21, used by OCC to report core 21 frequency
Bytes 184-187: Core 22 Temperature Sensor ID – Sensor ID for physical core 22, used by OCC to report core 22 temperature
Bytes 188-191: Core 22 Frequency Sensor ID – Sensor ID for physical core 22, used by OCC to report core 22 frequency
Bytes 192-195: Core 23 Temperature Sensor ID – Sensor ID for physical core 23, used by OCC to report core 23 temperature
Bytes 196-199: Core 23 Frequency Sensor ID – Sensor ID for physical core

	23, used by OCC to report core 23 frequency **Bytes 200-203:** Backplane Sensor ID – Used by OCC for system backplane error call out **Bytes 204-207:** APSS/SPI attached ADC Sensor ID – Used by OCC for APSS/SPI attached ADC error call out **Version 0x21 (8 additional bytes):** **Bytes 208-211**: VRM Vdd Sensor ID – Sensor ID for VRM Vdd, used by OCC for VRM Vdd error call out **Bytes 212-215**: VRM Vdd Temperature Sensor ID – Sensor ID for VRM Vdd, used by OCC to report VRM Vdd temperature

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 will send all 0's (Idle Power Saver disabled).

Data Length	9
Data	**Byte 1:** Format = 0x11 **Byte 2:** Version **<u>Version 0x00:</u>** **Byte 2:** Version = 0x00 **Byte 3:** Idle Power Saver Enable: 0 = disabled; 1 = enabled **Bytes 4-5:** Delay Time (in seconds) to enter Idle Power **Byte 6:** Utilization threshold percentage to enter Idle Power **Bytes 7-8:** Delay Time (in seconds) to exit Idle Power **Byte 9:** 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's are 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.
- Both Nimbus and Cumulus have the same N/M Throttling concept. The register to throttle memory is different for Nimbus and Cumulus reflected in the throttle information bytes. Nimbus vs Cumulus is sent in the "Memory Configuration" config data packet.
- 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 nominal and a PM mode (PowerVM only) or to the power capping throttles the OCC must write the appropriate N_PER_MBA and N_PER_CHIP. NOTE: In P9 the OCC does not switch to different memory throttles when in oversubscription. The OCC just starts actuating to the oversubscription power cap and will only throttle memory to the power capping throttle settings if necessary to maintain the oversubscription power cap.
- When throttling due to OT the OCC will only change N_PER_MBA, the N_PER_CHIP will remain unchanged.
- No Sensor ID for MBA will be sent to OCC. Any error reading/writing memory throttle will call out:
  - Nimbus: Processor.
  - Cumulus: Centaur. The Sensor ID for Centaur by Centaur # is sent in the "Memory Configuration" config data packet. OCC to lookup Centaur # to get Sensor ID from the "Memory Configuration" config data packet.
- Thermal reason to change throttles will be calling out the component that was OT

<table>
<tr><td>Data Length</td><td>Variable</td></tr>
<tr><td>Data</td><td>Byte 1: Format = 0x12
Byte 2: Version = 0x20
Byte 3: Number of memory throttling data sets to follow.

<u>Version 0x20:</u>
The format of each set is:
<table>
<tr><td>Data byte x</td><td>Throttle Info Byte 1:
<u>Nimbus:</u> MC. Indicates physical MC that the throttles are for. Valid values: 0=MC01; 1=MC23</td></tr>
</table>
</td></tr>
</table>

	Cumulus: Centaur #. Value 0-7 that indicates physical Centaur that the throttles are for
Data byte x+1	**Throttle Info Byte 2:** Nimbus: Port#. Value 0-3 that indicates physical port for the MC that the throttles are for. NOTE: Each MCU memory port supports up to two DIMM slots. Cumulus: MBA#. Value 0-1 that indicates MBA within the previous Centaur # that the throttles are for
Data bytes x+2 & x+3	**Minimum N_PER_MBA.** Lowest per MBA numerator ever allowed when OCC is throttling memory. This is calculated based on the MRW/def file minimum memory utilization.
Data bytes x+4 & x+5	**Memory Power @Minimum Throttle.** Maximum memory power in centiWatts with throttle set @Minimum
Data bytes x+6 & x+7	**Turbo N_PER_MBA.** Static per MBA numerator setting for turbo (i.e. DPS modes) on PowerVM systems. Calculated based on power available @turbo frequency. If power does not allow for this to meet MRW/def file minimum turbo memory utilization (H)TMGT will send Nominal N values and clip the maximum frequency sent in frequency data packet to nominal (no turbo support).
Data bytes x+8 & x+9	**Turbo N_PER_CHIP.** Static per chip numerator setting for turbo (i.e. DPS modes) on PowerVM systems.
Data bytes x+10 & x+11	**Memory Power @Turbo Throttle.** Maximum memory power in centiWatts with throttle set @Turbo
Data bytes x+12 & x+13	**Power Capping N_PER_MBA.** Static per MBA numerator setting when OCC needs to shed power due to a power cap. Calculated as the lower of throttles needed for oversubscription memory power and the power cap memory utilization value in MRW/def file. (H)TMGT to guarantee that this is lower or equal to turbo throttles to allow OCC power capping alg to use the settings regardless of mode.
Data bytes x+14 & x+15	**Power Capping N_PER_CHIP.** Static per chip numerator setting when OCC needs to shed power due to a power cap.
Data bytes x+16 & x+17	**Memory Power @PCap Throttle.** Maximum memory power in centiWatts with throttle set @Power Capping
Data bytes x+18 & x+19	**Nominal Power N_PER_MBA.** Static per MBA numerator setting for OPAL systems and when not in turbo on PowerVM systems. Calculated based on MRW memory power with redundant power. (H)TMGT to guarantee that this is not lower than MRW/def file minimum memory utilization for redundant power
Data bytes x+20 & x+21	**Nominal Power N_PER_CHIP.** Static per chip numerator setting for OPAL systems and when not in turbo on PowerVM systems.
Data bytes x+22 & x+23	**Memory Power @Nominal Throttle.** Maximum memory power in centiWatts with throttle set @Nominal

Data bytes x+24 & x+25	**Reserved**
Data bytes x+26 & x+27	**Reserved**
Data bytes x+28 & x+29	**Reserved**

2.4.9 Thermal Control Thresholds (Format = 0x13)

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

<table>
<tr><td>Data Length</td><td>Variable</td></tr>
<tr><td>Data</td><td>
Byte 1: Format = 0x13
Byte 2: Version= 0x20

<u>Version 0x20:</u>
Byte 3: Processor Core Weight. In 1/10ths. Weight factor for each core DTS to calculate a core temperature. 0 = core DTS not used to calculate core temperature (and eventual processor temperature)
Byte 4: Processor Quad Weight. In 1/10ths. Weight factor for each quad (cache) DTS to calculate a core temperature. 0 = quad DTS not used to calculate core temperature (and eventual processor temperature)
Byte 5: Number of data sets that follows. Format of each data set is:
<table>
<tr><td>Data byte x</td><td>FRU type. Indicates FRU type that thermal info is for
0x00: Processor (hottest core temperature)
0x01: Centaur
0x02: DIMM
0x03: VRM OT Status
0x04: GPU core
0x05: GPU memory
0x06: VRM Vdd</td></tr>
<tr><td>Data byte x+1</td><td>DVFS - Temperature above which DVFS/throttling will be invoked for OS controlled systems or in nominal mode for PowerVM systems. 0xFF indicates not defined.</td></tr>
<tr><td>Data byte x+2</td><td>ERROR - Temperature to generate error and callout FRU over temperature for OS controlled systems or in nominal mode for PowerVM systems. 0xFF indicates not defined.</td></tr>
<tr><td>Data byte x+3</td><td>PM_DVFS – powerVM systems only. Temperature above which DVFS/throttling will be invoked in any power management mode. 0xFF indicates not defined.</td></tr>
<tr><td>Data byte x+4</td><td>PM_ERROR – powerVM systems only. Temperature to generate error and callout FRU over temperature when in any power management mode. 0xFF indicates not defined.</td></tr>
<tr><td>Data byte x+5</td><td>MAX_READ_TIMEOUT – Maximum time (in seconds) allowed without having new temperature readings. Throttling/dvfs will occur if this timeout is hit. 0xFF indicates not defined.</td></tr>
</table>
Bytes 6-11: Data set #1
Bytes 12-17: Data set #2
</td></tr>
</table>

	:

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 read VDD and VDN on the AVSBus via OCI to SPIPMBus (O2S) Bridge B (NOTE: bridge A is PGPE owned and must not be used by the OCC). Version 2 is only sent if Vdd current overflow detection is enabled in the MRW (OP systems) or def file (FSP systems) (VDD_CURRENT_OVERFLOW_WORKAROUND_ENABLE is 1) AND MAX_VDD_CURRENT_READING is defined, else version 1 must be sent.

Data Length	Version dependent. 0x000A (version 1) or 0x000E (version 2)
Data	**Byte 1:** Format = 0x14 **Byte 2:** Version= 0x01 or 0x02 **<u>Versions 0x01 and 0x02:</u>** **Byte 3: Vdd Bus Num**. Defines the AVSBus (0 or 1) which has the core VDD rail VRM. 0xFF=Not defined, Vdd not monitored. **Byte 4: Vdd Rail Select**. Defines the AVSBus rail selector (0 – 15) for the VDD VRM on Vdd Bus Num **Bytes 5-6: Reserved**. **Byte 7: Vdn Bus Num.** Defines the AVSBus (0 or 1) which has the chip VDN (nest) rail VRM. 0xFF=Not defined, Vdn not monitored. **Byte 8: Vdn Rail Select**. Defines the AVSBus rail selector (0 – 15) for the VDN VRM on Vdn Bus Num **Bytes 9-10: Processor power adder**. A constant power (in watts) that the OCC is to add onto Vdd and Vdn power to represent any processor part that is not measured (i.e. Vcs, Vio…) This is used by the OCC to calculate a total processor socket power. Comes from ATTR_NO_APSS_PROC_POWER_VCS_VIO_WATTS, 0x0000 if not defined. **<u>Version 0x02 Only (Vdd Current overflow workaround enabled) :</u>** **Bytes 11-12: Vdd Current Rollover Point**. Current reading in 10mA at which the current reading from AVSBus rolls over to 0. This is used by the OCC to calculate Vdd current when a roll over is detected. Hard coded to 0x7FFF (327.67A). NOTE: This is the value that the chip physically rolls over and may differ from the roll over point defined to have the OCW status bit set to indicate a roll over occurred (set by FSP/BMC at boot) which may be set lower to give a little buffer. **Bytes 13-14: Maximum Vdd Current**. Maximum theoretical Vdd current reading in 10mA. This is used by the OCC as a sanity check when roll over is detected. Comes from mrw or def file MAX_VDD_CURRENT_READING

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.

Data Length	Version dependent. 0x002C (version 1) 0x000A minimum (version 2)
Data	**Byte 1:** Format = 0x15 **Byte 2:** Version = 0x01 or 0x02 **Version = 1** **Bytes 3-4:** 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. **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) **Bytes 7-8:** Reserved = 0x0000 **Bytes 9-12:** 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. **Bytes 13-16:** 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. **Bytes 17-20:** GPU 0 Sensor ID – Sensor ID for physical GPU 0, used by OCC for GPU 0 error callout. 0 if not defined in xml. **Bytes 21-24:** 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. **Bytes 25-28:** 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. **Bytes 29-32:** GPU 1 Sensor ID – Sensor ID for physical GPU 1, used by OCC for GPU 1 error callout. 0 if not defined in xml. **Bytes 33-36:** 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. **Bytes 37-40:** 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. **Bytes 41-44:** GPU 2 Sensor ID – Sensor ID for physical GPU 2, used by OCC for GPU 2 error callout. 0 if not defined in xml. **Version = 2** **Bytes 3-4: 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.

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:

Data byte x	**GPU ID.** Number (starting from 0 for first GPU) to indicate GPU that this data set is for
Data byte x+1	**GPU I2C Port.** 0xFF if not defined in xml and GPU will not be monitored.
Data byte x+2	**GPU I2C Slave Address.** 0xFF if not defined in xml and GPU will not be monitored.
Data byte x+3	**Reserved = 0x00**
Data bytes x+4 thru x+7	**GPU Core Temperature Sensor ID.** 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.
Data bytes x+8 thru x+11	**GPU HBM Temperature Sensor ID.** 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.
Data bytes x+12 thru x+15	**GPU Sensor ID.** Sensor ID for physical GPU. Used by OCC for GPU error callout. 0 if not defined in xml.

2.4.12 VRM Fault (Format = 0x21)

This is sent when TMGT is informed by POWR of a phase fault and gives the OCC notification of a VRM phase fault. In response to the phase fault the OCC will clip maximum frequency to the “VRM N mode maximum frequency” that was sent in the frequency configuration packet.

Data Length	0x0004
Data	**Byte 1:** Format = 0x21 **Byte 2:** Version= 0x01 **Version 0x01:** **Byte 3: VRM Fault Status**. Indicates if there is a phase fault (could be VDD or VCS, there is no differentiation) **0x00** = No fault present **0x01** = Fault present. OCC will clip max frequency to “VRM N mode” **Bytes 4: Reserved**.

2.4.13 Setup Configuration Data Return Packet

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

2.5 Set User Power Cap

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

BMC	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.
(H)TMGT	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:

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

Set User Power Cap Return Packet:

Sequence Number	Xx
Command Type	0x22
Return Status	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.
Data Length	0x0000
Data	There is no data returned.
Checksum	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.

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

Reset Prep Command Packet:

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

Reset Prep Return Packet:

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

2.7 Snapshot Sync

This command is used to tell the OCC to delete all previous snapshot buffers and start a new snapshot buffer from time 0.
Possible reasons this would be needed:

- SPCN requested a sync due to a power supply reset
- TMGT has detected clock drift between the SPCN AC readings and the OCC DC readings
- An OCC reset
- OCC requested sync in OCC poll response

TMGT	Sent when re-sync is needed
BMC	Never send
HTMGT	Never send

Snapshot Sync Command Packet:

Sequence Number	xx
Command Type	0x26
Data Length	0x0001
Data	There is 1 byte of data: **Byte 1:** Version = 0x00. Delete all snapshots. Start new snapshot from time 0. Currently, 0x00 is the only valid version.
Checksum	xxxx

Snapshot Sync Return Packet:

Sequence Number	xx
Command Type	0x26
Return Status	0x00 = Success See Appendix A for list of all non-successful return codes.
Data Length	0x0000
Data	There is no data returned.
Checksum	xxxx

2.8 Get OCC Snapshot Buffer

This command is used to get the OCC snapshot buffer and info about what snapshot buffers are available. TMGT will use information from the OCC snapshot buffer for reporting to CIMP. This command is only sent to the master OCC and the master OCC is reporting back total node values.

TMGT	Sent on a periodic time interval
BMC	Never send
HTMGT	Never send

Get OCC Snapshot Buffer Command Packet:

Sequence Number	xx
Command Type	0x35
Data Length	0x0004
Data	**Byte 1: Version** – Indicates what Get OCC Snapshot Buffer response version is being requested. **0x00** – Support for CIMP **Version 0x00:** **Byte 2: Reserved = 0x00** **Byte 3: Snapshot ID Number** – Indicates ID number for which snapshot buffer to return. The snapshot ID number should start at 0 when OCC is started and re-started at 0 on a sync command. Every 30s the master OCC is creating a new snapshot buffer and the ID number is incremented to correspond to the new snapshot buffer. The master OCC must store 6 snapshot buffers. **Byte 4: Return newest snapshot buffer** – **0x00** = Return the snapshot buffer for the snapshot ID number indicated in byte 3 **0x01** = OCC should ignore the snapshot ID number (byte 3) and just return the most recent snapshot buffer that OCC has available
Checksum	Xxxx

Get OCC Snapshot Buffer Return Packet:

<table>
<tr><td>Sequence Number</td><td>Xx</td></tr>
<tr><td>Command Type</td><td>0x35</td></tr>
<tr><td>Return Status</td><td>0x00 = Success
See Appendix A for list of all non-successful return codes.</td></tr>
<tr><td>Data Length</td><td>Version dependent</td></tr>
<tr><td>Data</td><td><u>Version = 0x00:</u>
323 bytes of data are returned:
Byte 1: Oldest available snapshot buffer ID number
Byte 2: Newest available snapshot buffer ID number
Byte 3: Snapshot buffer ID number of buffer data to follow, typically this should match the snapshot ID number in the command data if the snapshot buffer for the requested ID number is not available or if TMGT asked for the newest snapshot buffer then OCC should return the newest possible snapshot buffer.
Bytes 4-323: This is buffer data that TMGT sends to CIMP to be reported via CIM profiles. The data is in the following format (16 bytes/seq #) and repeated times with the most recent sequence number/data first
i.e. Bytes 4-19 newest data .... bytes 307-323 oldest data saved:
<table>
<tr><td>Byte x
(1 byte)</td><td>Sequence number. This sequence number is a one byte value that is incremented every 30 seconds when the snapshot buffer data is created and represents the past 30 seconds worth of data being added to the snapshot buffer. NOTE: this sequence number does NOT need to correlate in any way to the snapshot buffer ID numbers</td></tr>
<tr><td>Bytes x+1 thru x+4
(4 bytes)</td><td>Average Total Node DC power. In watts over 30 second time slice sequence number is for.</td></tr>
<tr><td>Bytes x+5 thru x+8
(4 bytes)</td><td>Maximum Total Node DC power. In watts over 30 second time slice sequence number is for.</td></tr>
<tr><td>Bytes x+9 thru x+12
(4 bytes)</td><td>Minimum Total Node DC power. In watts over 30 second time slice sequence number is for. (Currently TMGT will not send min to CIMP since profile doesn’t support, but will define having just in case min is added to the CIM profile)</td></tr>
<tr><td>Bytes x+13 thru x+14
(2 bytes)</td><td>Average Node CPU Frequency. In MHz over 30 second time slice sequence number is for. This is an average of all configured cores in the node.</td></tr>
<tr><td>Byte x+15
(1 byte)</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>Checksum</td><td>xxxx</td></tr>
</table>

2.9 Debug Pass Through

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

Debug Pass Through Command Packet:

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

Debug Pass Through Return Packet:

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

2.10 AME Pass Through

This command is for AMESTER use only.

AME Pass Through Command Packet:

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

AME Pass Through Return Packet:

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

2.11 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.

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

Get Field Debug Data Command Packet:

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

Get Field Debug Data Return Packet:

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

2.12 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:

Sequence Number	Xx
Command Type	0x53
Data Length	Sub command dependent
Data	There are x bytes of data (not to exceed maximum): **Byte 1:** Sub Command – Indicates type of mfg test cmd being sent. See following sections for command data details specific to each sub cmd. **Sub Cmd = 0x02: Run/Stop Slew Between Two Modes** **Sub Cmd = 0x05: List Sensors** **Sub Cmd = 0x06: Get Sensor Information** **Sub Cmd = 0x07: Enable/Disable Oversubscription Emulation** **Sub Cmd = 0x09: Run/Stop Memory Slew Between 1-100 Percent** **Sub Cmd = 0x0A: Request Pstate Per Quad** **Sub Cmd = 0x0B: Read Generated Pstate Table**
Checksum	Xxxx

Mfg Test Return Packet:

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

2.12.1 Run/Stop Slew Between Two Modes (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 Modes Mfg Test Command Packet:

Data Length	0x0009
Data	**Byte 1:** Sub Cmd = 0x02 **Byte 2:** Version – see below for defined values **Version = 0x00:** **Byte 3:** Action: 0x00 = Start 0x01 = Stop **Byte 4:** Bottom mode to slew from **Byte 5:** Bottom mode offset (percentage) **Byte 6:** High mode to slew to **Byte 7:** High mode offset (percentage) **Byte 8:** Slew step mode: 0x00 = Single step 0x01 = Full slew **Byte 9:** Additional delay between steps in ms (applicable for single step only)

Run/Stop Slew Between Two Modes Mfg Test Return Packet:

Data Length	0x0006
Return Data	**Bytes 1-2:** Count of slew number on a stop action (0 on start action) **Bytes 3-4:** fstart used **Bytes 5-6:** fstop used

2.12.2 List Sensors (Sub Cmd = 0x05)

List Sensors Command Packet:

Data Length	0x0009
Data	**Byte 1:** Sub Cmd = 0x05 **Byte 2:** Version = 0x00 **Bytes 3-4:** The GUID of sensor to start listing **Byte 5:** Present Option: 0x00 = List all sensors from start GUID 0x01 = List non-zero sensors from start GUID **Bytes 6-7:** 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 **Bytes 8-9:** 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 0x0100 = CPM 0x0200 = Performance 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.

List Sensors Response Packet:

Data Length	Variable
Return Data	**Byte 1:** Sensor list truncated. **0x01** = requested list of sensors do not fit within maximum of 50 sensors allowed to get remaining sensors send command with starting

<table>
<tr><td>Data Length</td><td>Variable</td></tr>
<tr><td></td><td>GUID of the last GUID in response +1.
0x00 = all requested sensors are returned
Byte 2: Number of sensor data sets that follows.
The format of a sensor data set is:
<table><tr><td>Data bytes x & x+1</td><td>Sensor GUID</td></tr><tr><td>Data bytes x+2 thru x+17</td><td>Sensor Name</td></tr><tr><td>Data bytes x+18 thru x+19</td><td>Sensor Sample</td></tr></table>
Bytes 3-22: Sensor data set #1.
Bytes 23-42: Sensor data set #2.
...</td></tr>
</table>

2.12.3 Get Sensor Information (Sub Cmd = 0x06)

Get Sensor Information Command Packet:

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

Get Sensor Information Response Packet:

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

2.12.4 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:

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

Enable/Disable Oversubscription Emulation Return Packet:

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

2.12.5 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:

Data Length	0x0003
Data	**Byte 1:** Sub Cmd = 0x09 **Byte 2:** Version = 0x00 **Byte 3:** Action: **0x00** = Start **0x01** = Stop

Run/Stop Memory Slew Return Packet:

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

2.12.6 Request Pstate per Quad (Sub Cmd = 0x0A)

This command is to be sent to each OCC to request Pstates per quad for only the OCC the command is sent to. This command is only valid when OCC is in active state with PowerVM. This is not supported in characterization or observation states.

NOTES:

- Pmin is the Pstate associated with the current lowest possible frequency (highest Pstate value). Pmax is the Pstate associated with the current highest possible frequency (lowest Pstate value).
- If the Pstate request is not within the current OCC Pmin-Pmax limits based on the current mode (i.e. in nominal Pmax is the Pstate associated with the frequency for nominal) the OCC will clip the Pstate request to Pmin or Pmax. Any clipping will be reflected in the return data.
- This is only a request for a Pstate, the actual Pstate set may be for a lower frequency if throttling is needed due to a power or thermal reason or due to WOF clipping.
- Any state change from active will clear these settings. i.e. the next time OCC goes active it will be in full control of all quads Pstate requests.
- To put the OCC back in full control while in the active state a 0xFF Pstate request must be sent for all quads

Request Pstate per Quad Command Packet:

Data Length	0x0008
Data	**Byte 1:** Sub Cmd = 0x0A **Byte 2:** Version = 0x00 **Bytes 3-8:** Pstate request per quad for quad 0 (byte 3) … quad 5 (byte 8) **0xFF** = No Pstate request. OCC in control based on mode

Request Pstate per Quad Return Packet:

Data Length	0x0006
Return Data	**Bytes 1-6:** Pstate request per quad for quad 0 … quad 5. This will be Pmin or Pmax if the request had to be clipped.

2.12.7 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 chips/p9/procedures/hwp/lib/p9_pstates_table.h the caller is responsible for understanding the format of the data and will not be documented here.

Read Generated Pstate Table Command Packet:

Data Length	0x0002
Data	**Byte 1:** Sub Cmd = 0x0B **Byte 2:** Request: **0x00…0x10 = Read Generated Pstate Table Block Offset.** 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. **0xFF = Query.** Returns the HOMER offset of the beginning of the generated Pstate table and the size of the table.

Read Generated Pstate Table Return Packet:

Data Length	Variable. 0x0008 (Query) or max of 0x0400 (Read Generated Pstate Table Block Offset)
Return Data	**Request Block Offset = 0x00…0x10** **Bytes 1… max of 1024:** Generated Pstate Table starting address + (block offset * 1024) **Query = 0xFF** **Bytes 1-4:** Generated Pstate Table HOMER Offset as an OCI PBA memory address (i.e. 0x8xxxxxx) **Bytes 5-8:** Generated Pstate Table Length

2.13 Tunable Parameters Command

This command is used to query or set tunable parameters. See Tunable Parameters List section below for definition of parameters supported.

Tunable Parameters Command Packet:

<table>
<tr><td>Sequence #</td><td>xx</td></tr>
<tr><td>Command</td><td>0x60</td></tr>
<tr><td>Data Length</td><td>Variable</td></tr>
<tr><td>Data</td><td>
Byte 1: Sub Cmd. Indicates operation:
0x00 = Query
0x01 = Write
0x02 = Restore Default Values

<u>Sub Cmd = 0x00 (Query)</u>
This returns a list of all supported tunable parameters defined in Tunable Parameters List section below..
Byte 2: Version. Defines format version to return. Currently only 0x00 supported.

<u>Sub Cmd = 0x01 (Write)</u>
This is used to set the value for tunable parameter(s)
Byte 2: Version. Defines format version. Currently only 0x00 supported.
Byte 3: Number of Parameters. Number of tunable parameters to follow.

Format for one parameter (repeated for # of parameters)
<table>
<tr><td>Data byte x</td><td>Tunable Parameter ID. Numerical value of tunable parameter to write. Values defined in Tunable Parameters List section below.</td></tr>
<tr><td>Data bytes x+1 and x+2</td><td>Value. Value to write to the given tunable parameter.</td></tr>
</table>
<u>Sub Cmd = 0x02 (Restore Default Values)</u>
No additional data. This is used to tell OCC to use default value for all supported tunable parameters defined in Tunable Parameters List section below. The OCC should use the default values until a Write is sent to change a parameter.
</td></tr>
<tr><td>Checksum</td><td>xxxx</td></tr>
</table>

Tunable Parameters Return Packet:

<table>
<tr><td>Sequence #</td><td>xx</td></tr>
<tr><td>Command</td><td>0x60</td></tr>
<tr><td>Return Status</td><td>0x00 = Success

See Appendix A for list of all non-successful return codes.</td></tr>
<tr><td>Data Length</td><td>Variable</td></tr>
<tr><td>Data</td><td><u>Sub Cmd = 0x00 (Query)</u>
This returns all supported tunable parameters from the Tunable Parameters List that this system type supports.
Byte 1: Version. Defines format version. Currently only 0x00 supported.
Byte 2: Number of Parameters. Number of tunable parameters to follow. Maximum of 29 allowed to stay within MTU.

Format of one parameter (repeated for # of parameters)
<table>
<tr><td>Data byte x</td><td>Tunable Parameter ID. Numerical value to represent the parameter</td></tr>
<tr><td>Data bytes x+1 thru x+60</td><td>Tunable Parameter Name. ASCII string. This will be the text displayed to the user. Must be padded with 0’s for the fixed length.</td></tr>
<tr><td>Data byte x+61</td><td>Unit Type. Defines the unit for the tunable parameter.</td></tr>
<tr><td>Data bytes x+62 and x+63</td><td>Flags. Bit defined for any special behavior/restrictions this parameter has. Currently there are no flags defined and this will be 0x0000 for all parameters.</td></tr>
<tr><td>Data bytes x+64 and x+65</td><td>Value. Current value of this tunable parameter. If none has been set this will be the default value.</td></tr>
<tr><td>Data bytes x+66 and x+67</td><td>Minimum. Minimum value that can be set for this tunable parameter.</td></tr>
<tr><td>Data bytes x+68 and x+69</td><td>Maximum. Maximum value that can be set for this tunable parameter.</td></tr>
</table>
<u>Sub Cmd = 0x01 (Write)</u>
No return data. A non-successful return code will be sent if one or more parameter sent in the command data to write was invalid. OCC will process as many parameters as possible, TMGT can query to see which values were changed.

<u>Sub Cmd = 0x02 (Restore Default Values)</u>
No return data</td></tr>
<tr><td>Checksum</td><td>xxxx</td></tr>
</table>

2.13.1 Tunable Parameters List

This is the list of tunable parameters that may be used with the Tunable Parameters command.

Parm ID	*Name*	*Unit Type*	*Flags*	*Default*	*Min*	*Max*	*Notes*
1	Upper Threshold	3	0	99.9%	0%	100%	dpsalg.alpha_up
2	Lower Threshold	3	0	99.9%	0%	100%	dpsalg.alpha_down
3	Sample Count	4	0	4	1	1024	dpsalg.sample_count_util
4	Step-up Size	3	0	0.8%	0.1%	100%	dpsalg.step_up
5	Step-down Size	3	0	0.8%	0.1%	100%	dpsalg.step_down
6	Activity Threshold	2	0	18%	0%	100%	dpsalg.epsilon_perc
7	Slack Threshold	3	0	98.0%	0%	100%	dpsalg.tlutil
8	f delta enable	0	0	0	0	1	Used by master only G_mst_tunable_parameter_table_ext[7].adj_value
9	Max f delta	2	0	10%	10%	100%	Used by master only G_mst_tunable_parameter_table_ext[8].adj_value
10	WOF Enable	0	0	1	0	1	dpsalg.wof_enable This will only enable WOF in power management modes that support WOF and assuming all other requirements (VPD, system WOF enable…) are met to allow for WOF to run.

2.13.1.1 Tunable Parameters Unit Types

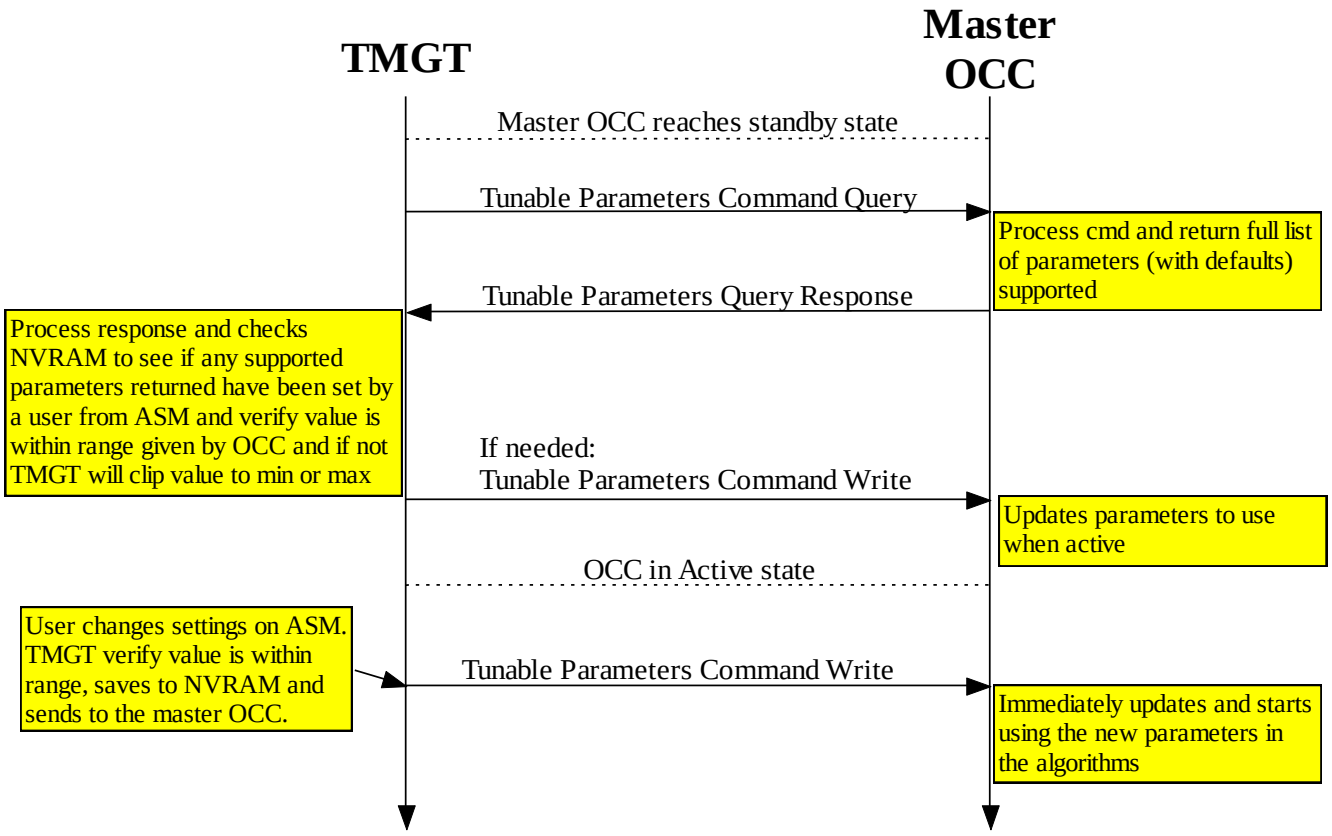
Type	*Description*
0	**Flag.** Used for parameters that can only be either 1 (on/enabled) or 0 (off/disabled)
1	**Time in seconds.** Values are in seconds. i.e. 10 = 10 seconds
2	**Percent.** Values are full percents. i.e. 10 = 10%
3	**Tenth of a Percent.** Values are in tenth of a percent, i.e. 100 = 10.0%
4	**Number.** Values are unsigned integers

2.13.1.2 Tunable Parameters Flags

Bit	*Description*
0:15	**Reserved for future use**

2.13.2 Sequence Diagram of Initial Exchange to Populate Parameters

This shows the exchange that will happen after an OCC reset. This reset may be due to an IPL or an OCC code update and is how TMGT will populate the list of sensors supported with ranges a user is allowed to set. Parameters may only be set from ASM, these parameters are not supported from AEM. Once OCC is in active state additional tunable write commands may be received as user changes settings from ASM, these changes should take effect immediately.



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 and length 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.

3.1.1 Read OCC Error Log from 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 = 0x01. Indicate format version of error log to parse data.

Version = 0x01:
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 0x2A.
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	Recoverable 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***
0:2	*Reserved*
3	**Manufacturing error**. 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)
4	*Reserved*
5	**Reset due to WOF Error**. 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.
6	**Safe Mode Required**. 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). One case for this is a checkstop.
7 (msb)	**Reset Required**. 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.

Bytes 8-9: Reserved 0x00 0x00
Bytes 10-11: Extended Reason Code. (H)TMGT will use this as the lower 2 bytes for the error log user data word 4. NOTE: Upper 2 bytes of user data word 4 is OCC module ID.
Byte 12: Max Number of callouts = 6
Maximum number of callouts that follow, this is to know size of callout section to know where user data begins. Prior to reaching the end of max number of callouts a callout of all 0's will indicate the end of actual callouts. Each callout is 12 bytes.
Bytes 13-84: Callouts – Each of the callouts contains 12 bytes in order:

Callout Byte x	**Type.** Type of callout: **0x01** – Sensor ID (following 8 Callout bytes is either sensor ID (IPMI) or HUID (no IPMI) **0x02** – TMGT-OCC Component ID (following 8 Callout bytes are 7 0x00's followed by the HTMGT-OCC ID defined in Appendix D) **0x03** – GPU ID (following 8 Callout bytes is a sensor ID for a GPU)
Callout Bytes x+1 thru x+8	**Callout.** Callout value format type defined in previous "Type" byte.
Callout Byte x+9	**Priority.** Priority for this callout: **0x01** - Low priority. **0x02** - Medium priority. **0x03** - High priority.
Callout Bytes x+10 & x+11	**Reserved.**

Bytes 85 thru Error Log Length from OCC poll response that contained the error log start address this error log is for: Additional OCC Data. Order of data:

Name	***Size***	***Description***
Version	1 byte	User Details version
Reserved	1 byte	
modId	2 bytes	OCC 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
fwLevel	4 bytes	OCC FW Level
timestamp	8 bytes	OCC Time stamp
occId	1 byte	OCC ID error log is from
occRole	1 byte	OCC role
operatingState	1 byte	OCC state
committed	1 byte	Indicates if log is committed
userData1	4 bytes	(H)TMGT: building error log appended as user data word 1
userData2	4 bytes	(H)TMGT: building error log appended as user data word 2
userData3	4 bytes	(H)TMGT: building error log appended as user data word 3
entrySize	2 bytes	Size of complete OCC error log
userDetailEntrySize	2 bytes	User Details Size
OCC Trace	variable	To end of error log: OCC Trace data

3.2 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, SGPEs). 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.

3.2.1 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.

3.2.1.1 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 SMS Attention for OCC reset to OPAL with sensor ID of failing OCC
4. OPAL reads IPMI command buffer and translates sensor ID to a chip ID and calls HBRT Process OCC Error interface with the chip ID
5. HBRT calls HTMGT OCC error handling function with the chip ID
6. HTMGT runs OCC Reset Procedure.

3.2.2 (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

3.2.3 OCC Detected Reasons for OCC Reset

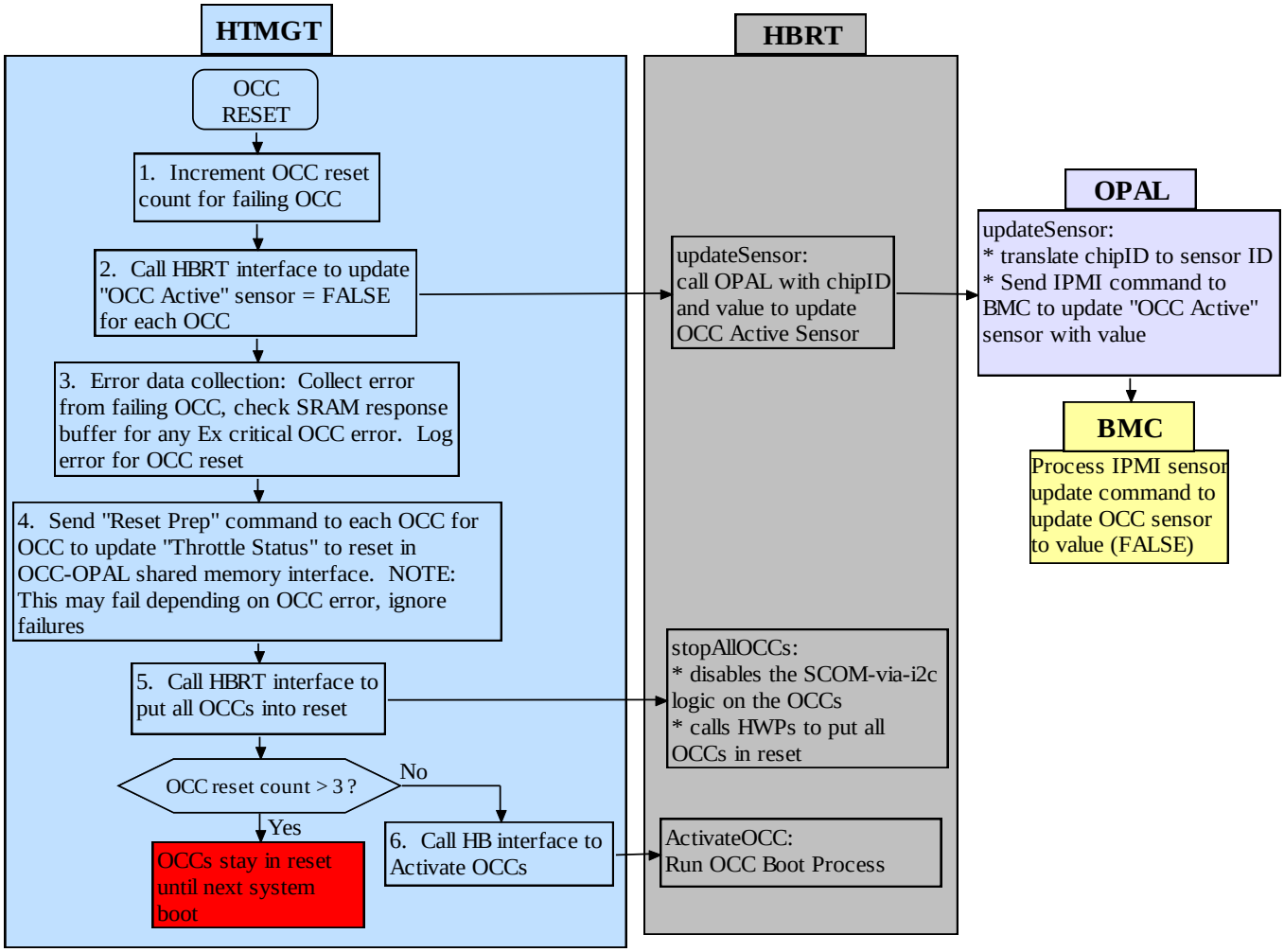
OCC will create an error log and request a reset for the following:

- 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.

3.3 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.



3.4 Error Scenarios

3.4.1 (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.

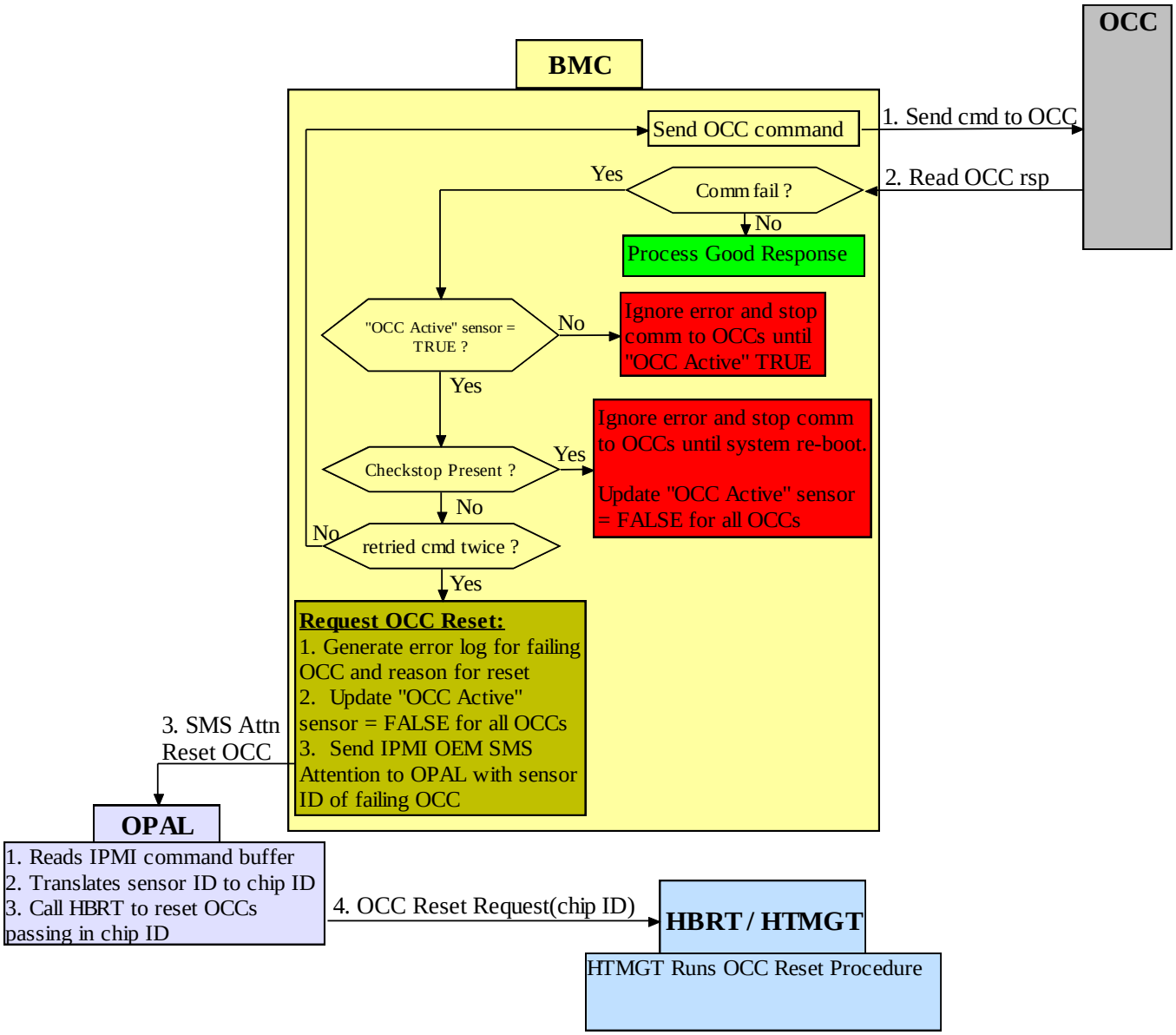
3.4.2 BMC-OCC Communication Failure

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
- I2C 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 retired 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.

3.4.2.1 BMC-OCC Communication Failure Handling Flow



3.4.3 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 FLASE 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.

3.4.4 Checkstop

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

OCC Actions: OCC FIR Master will detect the checkstop and collect FIR SCOM data and the 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.

3.4.5 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.

3.4.6 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.

3.4.7 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.

3.4.8 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.

3.4.9 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 FSP System Boot Process with PHYP

1. HB will start/use SGPE to activate all cores

2. HB shuts down SGPE

3. HB starts PHYP

4. Reach PHYP standby

5. TMGT is informed from transition file to activate OCCs. TMGT waits to make sure PHYP reach standby and then sends MBOX message “Load/Start PM Complex” with scope of system

6. PHYP loops around every functional proc chip and calls HBRT:load_pm_complex(procchip, homer_addr, occ_common_addr, LOAD) ←PHYP knows “LOAD”

7. HBRT loads images from lids via PHYP

    7.1 load hcode reference image lid, run xip_customize against reference image and copy result into mainstore HOMER

    7.2 load occ lid, fill in occ config data

8. PHYP loops around every functional proc chip calls HBRT:start_pm_complex(procchip)

9. HBRT executes HWP p9_pm_init(INIT)

10. PHYP sends one “Load/Start PM Complex Status” message to TMGT

11. Success: TMGT starts communicating with OCCs. Failure: TMGT starts OCC reset process

4.2 FSP OCC Reset with PHYP

0. System at runtime, OCCs enabled

1. TMGT detects a need to reset the OCCs and sends MBOX “Reset PM Complex” message with scope of system and master OCC chip ID for each node

2. PHYP would start queueing up requests at this point

3. PHYP loops through every functional proc chip and puts PM complex into reset last calling HBRT:reset_pm_complex(procchip)

4. HBRT executes p9_pm_init(reset)

5. PHYP sends a single MBOX message “Reset PM Complex Status” with status success if all chips successfully put in reset

6. Success: TMGT sends MBOX “Load/Start PM Complex” with scope of system. (If failure TMGT goes back to step 1 for a maximum of 3 retries, after which everything is kept in reset)

7. PHYP loops around every functional proc chip and calls HBRT:load_pm_complex(procchip, RELOAD) ←PHYP knows “RELOAD”

8. HBRT reloads OCC lid and rewrites OCC config data into HOMER. NOTE: HBRT doesn’t touch any of the rest of HOMER (24x7, hcode…)

9. PHYP loops around every functional proc chip calls HBRT:start_pm_complex(procchip)

10. HBRT executes HWP p9_pm_init(INIT)

11. PHYP sends one MBOX “Load/Start PM Complex Status” message to TMGT

12. Success: TMGT starts communicating with OCCs. Failure: TMGT goes back to step 1 for a maximum of 3 retries, after which everything is kept in reset

4.3 FSP OCC Reset Fails with PHYP

If any of the steps in the FSP OCC Reset with PHYP fails for any chip PHYP will send back failure in the Reset PM Complex Status message or Load/Start PM Complex Status message to TMGT. TMGT will then start the whole reset process over with step 1 for a maximum of 3 retries. After 3 retires TMGT will log an unrecoverable error and the whole power management complex is kept in reset. While in reset 24x7 messages, Concurrent Core Init (CCI), cache repair (column repair) are lost. Only STOP 1 (nap equivalent) will work.

4.4 FSP Concurrent Code Update with PHYP and OCCs Active

0. Steady state runtime, everything enabled
1. FSP updates code and does a reset/reload
2. tmgtserver is started. TMGT reads registry tmgt/CodeUpdate that CUPD wrote prior to the reset/reload to indicate concurrent code update was done
3. TMGT talks to OCC and sends Prep for Reset command to each OCC
4. After PHYP communication established: TMGT sends MBOX 'Reset PM Complex' message with scope of system and master OCC chip ID for each node to PHYP
5. PHYP would start queueing up requests at this point
6. PHYP loops through every functional proc chip and puts PM complex into reset last calling HBRT:reset_pm_complex(procchip)
7. HBRT executes p9_pm_init(reset)
8. PHYP sends a single MBOX message “Reset PM Complex Status” with status success if all chips successfully put in reset
9. FSP finishes code update

10. TMGT is called from transition file to runtime prior to updating to runtime state to ensure everything is in reset. HMC is polling the FSP to reach “runtime” state, once it reaches runtime state HMC tells PHYP to update code. To prevent PHYP from updating TMGT blocks transition file from updating to runtime state until PM complex reset is complete.
11. HMC tells PHYP to update code
12. PHYP loops through every functional proc chip and calls HBRT:load_pm_complex(procchip,RELOAD)
13. HBRT loads lids and customizes as needed
- all lids must be new and uncached
- load hcode, rerun customize, copy out to mainstore, preserve runtime portions like P8
    - same as IPL path except runtime portions are preserved
- load occ lid, reconfig homer (same as IPL and OCC reset paths)

14. PHYP loops through every functional proc chip and calls HBRT:start_pm_complex(procchip)
- exactly the same as IPL path

15. PHYP sends single “Load/Start PM Complex Status” message to TMGT indicating success or failure using reserved sequence id of 0xFFFFFFFE to indicate PHYP started the PM complex without a Load/Start message from TMGT.
16. Success: PHYP does STOP11 (winkle equivalent) cycles on every functional core to apply CCI. Failure: TMGT will start reset process over for max of 3 retries after TMGT sees that PHYP code complete is done. In addition on failure CCI are not applied and no failure message (same as P8). CCI will not be applied on the reset retry. To apply CCI will require another concurrent code update.
17. PHYP code update complete
18. FSP:TMGT resumes active communication and monitoring of the OCC
19. Steady state

4.5 FSP Concurrent Code Update with PHYP and PM in Reset

0. Steady state runtime, PM Complex in reset
1. FSP updates code and does a reset/reload
2. tmgtserver is started. TMGT reads registry that CUPD wrote prior to the reset/reload to indicate concurrent code update was done
3. TMGT talks to OCC and sees communication failure due to already being in reset
4. After PHYP communication established: TMGT sends MBOX 'Reset PM Complex' message with scope of system and master OCC chip ID for each node to PHYP. PHYP sees that all OCCs are in reset and rejects the request with 0xC5. TMGT will ignore this return code, no error generated.
5. Concurrent code update continues the same as OCC active scenario.

4.6 OCC Load/Start Process on FSP with OPAL

1. HOMER is placed at the top of memory
2. HB loads images from lids via HWSV
3. HB starts complex

4. HB adds HOMER to reserved memory space in HDAT
5. OPAL is started
6. TMGT is informed from transition file to activate OCCs. TMGT waits to make sure OPAL is ready and then TMGT sends MBOX message “Load/Start PM Complex” with scope of system
7. OPAL immediately sends “Load/Start PM Complex Status” message to TMGT
8. Success: TMGT starts communicating with OCCs. Failure: TMGT starts OCC reset process

4.7 OCC Load/Start Process on FSP with AVP

1. HOMER is placed at the top of memory
2. HB loads images from lids via HWSV
3. HB starts complex
4. AVP is started
5. TMGT does nothing

4.8 OCC Load/Start Process on BMC with PHYP

1. PHYP is started
2. PHYP loops around every functional proc chip and calls HBRT:load_pm_complex(procchip, homer_addr, occ_common_addr, LOAD) ←PHYP knows “LOAD”
3. HBRT loads images from PNOR via PHYP
4. PHYP loops around every functional proc chip calls HBRT:start_pm_complex(procchip)
5. HBRT executes HWP p9_pm_init(INIT)
6. HBRT calls HTMGT to setup OCCs
7. HTMGT communicates with OCCs and make them active
8. HBRT returns to PHYP

4.9 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 complex
4. HBRT calls HTMGT to setup OCCs
5. HTMGT communicates with OCCs and make them active
6. HB adds HOMER to reserved memory space in HDAT
7. OPAL is started

4.10 BMC Activate OCCs Procedure

5 Frequency Points

- Frequency points are across whole system

5.1 VPD

These are frequency points defined in the module VPD #V keyword.
#V processing will be done at istep 6.12, TMGT must not try to read the frequency attributes until after this istep.

Frequency Point	Notes
Power Save	➢ This is the lowest frequency/voltage point ➢ May have different frequencies, Host Boot to take the max of all chips in the system. No error generated due to differences. ➢ FSP only: Host Boot determines if the system owner has limited the lowest frequency by getting "frequency_points"{DPO_MIN_FREQ} from the power management def file ➢ Host Boot writes the max of VPD and def file limit to ATTR_MIN_FREQ_MHZ attribute that (H)TMGT reads to know minimum frequency and sends to the OCC ➢ Epsilon is set based on ATTR_MIN_FREQ_MHZ
Nominal	➢ Must be the same across all chips, Host Boot will generate an error if it is not ➢ Host Boot writes this value to ATTR_NOMINAL_FREQ_MHZ attribute that (H)TMGT will read to get nominal to send to the OCC
Turbo	➢ Host Boot to take the min of all chips in the system. Not expected to be different but if it is no error is generated ➢ Host Boot writes this value (the min of all chips) to ATTR_FREQ_CORE_MAX attribute that (H)TMGT will read to get the legacy "turbo" chip max frequency.
Ultra Turbo	➢ Host Boot to take the min of all chips in the system. Not expected to be different but if it is no error is generated. ➢ Host Boot writes this value (the min of all chips) to ATTR_ULTRA_TURBO_FREQ_MHZ attribute that (H)TMGT will read to send to the OCC. ➢ If any chip is missing this VPD point, ultra turbo is not supported and the attribute should be written to 0 by Host Boot.

5.2 Configuration File

Defined in the configuration file.

Frequency	Notes
Boot Frequency	➢ Frequency that the system will be booted at must be low enough to keep system power and thermal safe until OCCs are active ➢ Must be >= ATTR_MIN_FREQ_MHZ (cannot be below epsilon value) o Host Boot to make this check and if it isn't log error and raise boot to ATTR_MIN_FREQ_MHZ
Nest Frequency	➢ Written to HOMER by Host Boot. OCC reads directly from HOMER.

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 = 0x02
Sensor Length = 0x06

Format for one sensor and repeated for Number of Sensors:

Offset	
0x00	**Sensor ID –** 4 bytes. To identify what the temperature represents
0x04	**FRU Type –** 1 byte. Indicates what type of FRU the temperature is for: 0x00: Processor (core temperature) 0x01: Centaur 0x02: DIMM 0x03: VRM OT Status: Not a temperature “Current Reading” will be OT status bit read via AVSBUS 0 = Not OT 1 = OT 0xFF = Error reading 0x04: GPU core 0x05: GPU memory 0x06: VRM Vdd
0x05	**Current Reading –** 1 byte. Current temperature reading in degrees C. **0x00** = Not present / Unavailable **0xFF** = 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.

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:

Offset	
0x00	**Sensor ID –** 4 bytes. To identify what the frequency represents
0x04	**Current Reading –** 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:

<table>
<tr><th>Offset</th><th></th></tr>
<tr><td>0x00</td><td>Sensor ID – 4 bytes. Used to report power.</td></tr>
<tr><td>0x04</td><td>Function ID – 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 (Centaur)</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 (2nd channel of memory power for Proc 0)</td></tr>
<tr><td>6 = Processor 1</td><td>26 = Memory Proc 0-1 (3rd channel of memory power for Proc 0)</td></tr>
<tr><td>7 = Processor 2</td><td>27 = Memory Proc 0-2 (4th 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></td></tr>
<tr><td>15 = IO C</td><td></td></tr>
<tr><td>16 = Fans A</td><td></td></tr>
<tr><td>17 = Fans B</td><td></td></tr>
<tr><td>18 = Storage A</td><td></td></tr>
<tr><td>19 = Storage B</td><td></td></tr>
</table></td></tr>
<tr><td>0x05</td><td>APSS Channel – 1 byte 0–15. Indicates the APSS channel that the power was read from. This is the ADC_CHANNEL_ASSIGNMENT in xml file.</td></tr>
<tr><td>0x06</td><td>Reserved – 2 bytes reserved = 0x0000</td></tr>
<tr><td>0x08</td><td>Update Tag – 4 bytes. Count of number of 500us samples represented in Accumulator. Used for time derived sensor.</td></tr>
</table>

0x0C	**Accumulator –** 8 bytes. Accumulation of 500us power readings
0x14	**Current Reading –** 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.

Offset	
0x00	**System Power Sensor ID –** 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)
0x04	**System Power Update Time –** 2 bytes. The time in microseconds that the system power is read by the OCC
0x06	**Current System Power –** 2 bytes. Most recent system power reading in watts
0x08	**System Power Update Tag –** 4 bytes. Count of number of update time samples represented in the System Power Accumulator.
0x0C	**System Power Accumulator –** 8 bytes. Accumulation of system power readings
0x14	**Reserved –** 4 bytes. 0x00000000
0x18	**Processor Power Update Time –** 2 bytes. The time in microseconds that processor power is read by the OCC
0x1A	**Current Total Processor Power –** 2 bytes. Most recent total processor power reading in watts. This includes Vdd + Vdn + fixed adder for proc power not read
0x1C	**Total Processor Power Update Tag –** 4 bytes. Count of number of processor update time samples represented in the Total Processor Power Accumulator.
0x20	**Total Processor Power Accumulator –** 8 bytes. Accumulation of total processor power readings
0x28	**Current Processor Vdd Power –** 2 bytes. Most recent processor Vdd power reading in watts
0x2A	**Processor Vdd Power Update Tag –** 4 bytes. Count of number of processor update time samples represented in the Processor Vdd Power Accumulator.
0x2E	**Processor Vdd Power Accumulator –** 8 bytes. Accumulation of processor Vdd power readings
0x36	**Current Processor Vdn Power –** 2 bytes. Most recent processor Vdn power reading in watts
0x38	**Processor Vdn Power Update Tag –** 4 bytes. Count of number of processor update time samples represented in the Processor Vdn Power Accumulator.
0x3C	**Processor Vdn Power Accumulator –** 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	**Current Power Cap** – 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: • N Power Cap • Maximum System Power Cap • User Power Limit
0x02	**Current System Power Reading** – 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	**N Power Cap** – 2 bytes. In 1W units the (output) power cap limit when there is not redundant power.
0x06	**Maximum System Power Cap** – 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	**Hard Minimum Power Cap** – 2 bytes. In 1W units the minimum (output) power cap that may be set and held by the OCC.
0x0A	**Soft Minimum Power Cap** – 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	**User Power Limit** – 2 bytes. In 1W units the (output) power cap specified by a user. NOTES: • This will be 0x0000 if no user set power limit or the user set power limit is not active • 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.
0x0E	**User Power Limit Source** – 1 byte. Indicates how the power limit was set **0x01** = User power limit was set out of band (cmd from (H)TMGT or BMC) **0x02** = 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	**Name** – 4 bytes to identify the data. This can be ASCII or sensor ID
0x04	**Flags** – 1 byte bit defined flags for any special processing to be done by the BMC: **Bit 7 (msb) – Sensor ID.** ’1’ **=** Name is a sensor ID to be exported via normal sensor processing else Name is an ASCII string to be exported TBD. **Bits 6:0 – Reserved**
0x05	**Reserved** – 1 byte reserved = 0x00
0x06	**Data** – 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:

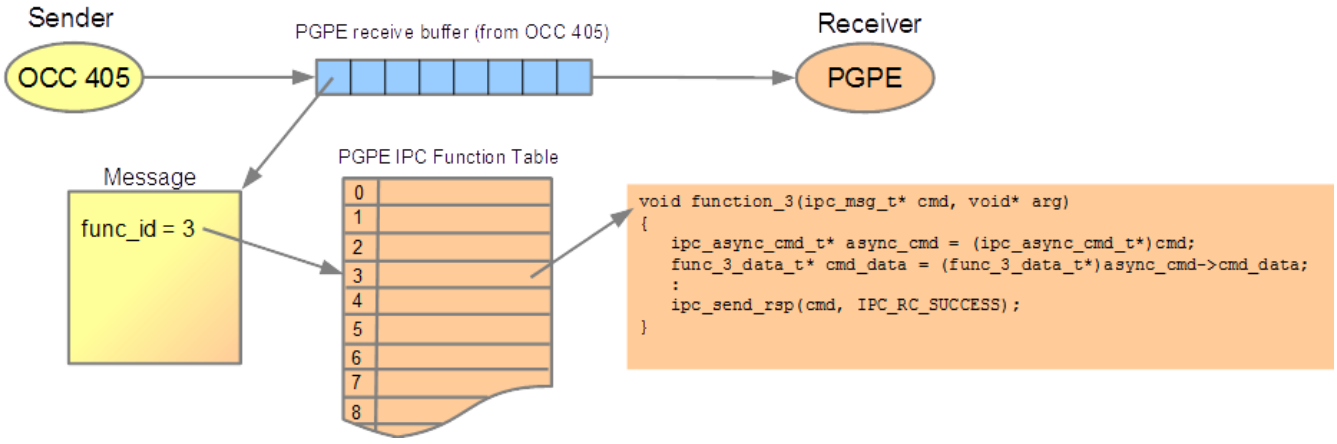
<table>
<tr><th>Name</th><th>Flags</th><th>Data</th></tr>
<tr><td>“FMIN”</td><td>0x00</td><td>Minimum Frequency – Minimum frequency system is allowed to run at:
<table>
<tr><th>Byte 1</th><th>Byte 2</th><th>Byte 3</th><th>Byte 4</th><th>Byte 5</th><th>Byte 6</th></tr>
<tr><td>Pstate</td><td colspan="2">Frequency in MHz</td><td colspan="3">Reserved = 0</td></tr>
</table></td></tr>
<tr><td>“FNOM”</td><td>0x00</td><td>Nominal Frequency– Nominal frequency for the system:
<table>
<tr><th>Byte 1</th><th>Byte 2</th><th>Byte 3</th><th>Byte 4</th><th>Byte 5</th><th>Byte 6</th></tr>
<tr><td>Pstate</td><td colspan="2">Frequency in MHz</td><td colspan="3">Reserved = 0</td></tr>
</table></td></tr>
<tr><td>“FT”</td><td>0x00</td><td>Turbo Frequency – Turbo frequency. Will be all 0’s if turbo is not supported:
<table>
<tr><th>Byte 1</th><th>Byte 2</th><th>Byte 3</th><th>Byte 4</th><th>Byte 5</th><th>Byte 6</th></tr>
<tr><td>Pstate</td><td colspan="2">Frequency in MHz</td><td colspan="3">Reserved = 0</td></tr>
</table></td></tr>
<tr><td>“FUT”</td><td>0x00</td><td>Ultra Turbo Frequency – Ultra Turbo frequency. Will be all 0’s if ultra turbo (WOF) is not supported:
<table>
<tr><th>Byte 1</th><th>Byte 2</th><th>Byte 3</th><th>Byte 4</th><th>Byte 5</th><th>Byte 6</th></tr>
<tr><td>Pstate</td><td colspan="2">Frequency in MHz</td><td colspan="3">Reserved = 0</td></tr>
<tr><td></td><td colspan="2"></td><td colspan="3"></td></tr>
</table></td></tr>
</table>

<table>
<tr><td>“CLIP”</td><td>0x00</td><td>Clip – 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.
<table>
<tr><th>Byte 1</th><th>Byte 2</th><th>Byte 3</th><th>Byte 4</th><th>Byte 5</th><th>Byte 6</th></tr>
<tr><td>Current max Pstate</td><td>Count</td><td colspan="4">Clip Reason</td></tr>
</table></td></tr>
<tr><td>“WOFC”</td><td>0x00</td><td>WOF Clip – Gives WOF information from the PGPE
<table>
<tr><th></th><th>Byte 1</th><th>Byte 2</th><th>Byte 3</th><th>Byte 4</th><th>Byte 5</th><th>Byte 6</th></tr>
<tr><td>WOF Enabled</td><td>F Clip (Pstate)</td><td colspan="2">Vclip</td><td colspan="2">Vratio</td><td>0x00</td></tr>
<tr><td>WOF Disabled</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></td></tr>
<tr><td>“ERRH”</td><td>0x00</td><td>Error History – 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.
<table>
<tr><th>Byte 1</th><th>Byte 2</th><th>Byte 3</th><th>Byte 4</th><th>Byte 5</th><th>Byte 6</th></tr>
<tr><td>Enum</td><td>Count</td><td>Enum</td><td>Count</td><td>Enum</td><td>Count</td></tr>
</table></td></tr>
</table>

7 OCC to PGPE Communication

7.1 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.



IPC Function ID	*Operation*
1	Start/Stop Pstate Protocol
2	Pstate Clip Update
3	Set PMCR
4	WOF Control
5	WOF VFRT (Voltage/Frequency Ratio Table)

7.1.1 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.

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

7.1.2 Pstate Clip Update

This command is used to set the minimum and maximum Pstate per quad.

- 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)

Input Data	**Pmin** – 6 bytes minimum Pstate for quad 0…5 **Pmax** – 6 bytes maximum Pstate for quad 0…5
Output Data	**Return Code** Any non-successful return code will result in a PM complex reset

7.1.3 Set PMCR

This command is used to send the PGPE a PMCR write request (requested Pstate to set per quad). 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 this is to off load the OCCs need to do 24 SCOMs every 250us writing the PMCR for all cores.

Input Data	**Pstate** – 6 bytes requested Pstate for quad 0…5
Output Data	**Return Code:** **NOT_PMCR_OWNER** – 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

7.1.4 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

Input Data	**Action** – Defines if WOF should be on or off: WOF_ON WOF_OFF
Output Data	**Return Code** • NOT_ENABLED – PGPE unable to enable WOF (i.e. missing a piece of data). PGPE is still running without WOF. OCC will continue to run (no PM reset) without WOF and the max frequency will be limited to legacy turbo. OCC will log an error (FW callout only) so user knows WOF function is disabled. • Any other non-successful RC will result in a PM complex reset

7.1.5 WOF VFRT

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

- Pstate Protocol MUST be started in order to send WOF VFRT
- One WOF VFRT will be sent when WOF is OFF to allow for WOF to be turned ON
- While WOF is ON there is no timing requirement for how often this command will be sent. The PGPE should continue to use the last VFRT until another VFRT is received.
- OCC will have 2 SRAM buffers for a VFRT, one that the PGPE is actively using as pointed to from the last successful WOF VFRT response and one that the OCC can actively update with a new VFRT and will be sent in the next WOF VFRT command. The OCC cannot switch to the SRAM buffer that the PGPE is using until the PGPE has responded success.
- Handling of quad wanting to turn on:
  1. Every 2ms OCC will check for a change to requested active quads from PGPE shared memory
  2. OCC determines new VFRT based on new number of active quads
  3. OCC sends WOF VFRT with new active quads. PGPE updates snapshot with new quad status and then sends back IPC response to OCC. (100's of microseconds)

Input Data	**VFRT Pointer** – SRAM address of VFRT to use **Active quads** – Bit vector of active quads that was used to determine VFRT
Output Data	**Return Code** • VFRT_QUAD_MISMATCH i.e. active quad mismatch with requested active quads. OCC cannot take back memory from previous VFRT. • Success. New VFRT in place OCC can take back memory from previous VFRT • Any other non-successful return code should result in a PM complex reset

8 Power Management

8.1 XML Settings (BMC)

The power and thermal configuration settings are data driven defined in the system xml file.

XML Attribute	Description
OPEN_POWER_TURBO_MODE_SUPPORTED	Indicates if setting a frequency above nominal is supported. 1=supported 0=not supported and the max frequency allowed will be nominal and WOF is disabled.
OPEN_POWER_N_PLUS_ONE_BULK_POWER_LIMIT_WATTS	System maximum power cap in output watts when QPD line is not asserted. If Turbo Support is supported this power must assume chip power at the maximum turbo frequency and make maximum memory power lower to allow turbo frequency. If Turbo is not supported then this value is to guarantee nominal processor power.
OPEN_POWER_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 turbo or nominal (based on Turbo Support). NOTE: This value is first reduced by Regulator Efficiency Factor before running the procedure to account for regulator loss.
OPEN_POWER_N_BULK_POWER_LIMIT_WATTS	System maximum power cap in output watts when QPD line is asserted. If Turbo is supported this power must assume chip power at the maximum turbo frequency and make maximum memory power lower to allow turbo frequency. If Turbo is not supported then this value is to guarantee nominal processor power.
OPEN_POWER_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 turbo or nominal (based on Turbo Support). NOTE: This value is first reduced by Regulator Efficiency Factor before running the procedure to account for regulator loss.
OPEN_POWER_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.
OPEN_POWER_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.
OPEN_POWER_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 (OPEN_POWER_MIN_POWER_CAP_WATTS) is called a soft power cap and is not guaranteed under all conditions.
OPEN_POWER_MIN_MEM_UTILIZATION_THROTTLING	The lowest utilization allowed that the OCC can throttle memory due to a memory over temp condition.
OPEN_POWER_MIN_MEM_UTILIZATION_POWER_CAP	The lowest utilization allowed that the OCC can throttle

	memory due to power capping.
OPEN_POWER_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.
OPEN_POWER_QUAD_WEIGHT_TENTHS	Weight factor (1 = 0.1) for each quad (cache) DTS to calculate a core temperature. A value of 0 means not to use quad DTS and only the core DTSs are used to calculate a core temperature.
OPEN_POWER_PROC_DVFS_TEMP_DEG_C	Processor Temperature in degrees C to invoke DVFS (clip max Pstate)
OPEN_POWER_PROC_ERROR_TEMP_DEG_C	Processor Temperature in degrees C that a processor overtemp error will be logged calling out the processor
OPEN_POWER_PROC_READ_TIMEOUT_SEC	Maximum time in seconds allowed without having a new processor temperature before DVFS will occur
OPEN_POWER_VRM_VDD_DVFS_TEMP_DEG_C	VRM Vdd Temperature in degrees C to invoke DVFS (clip max Pstate)
OPEN_POWER_VRM_VDD_ERROR_TEMP_DEG_C	VRM Vdd Temperature in degrees C that an overtemp error will be logged
OPEN_POWER_VRM_VDD_READ_TIMEOUT_SEC	Maximum time in seconds allowed without having a new VRM Vdd temperature before DVFS will occur
OPEN_POWER_MEMCTRL_THROTTLE_TEMP_DEG_C	Centaur Temperature to invoke memory throttling
OPEN_POWER_MEMCTRL_ERROR_TEMP_DEG_C	Centaur Temperature in degrees C that a centaur overtemp error will be logged calling out the centaur
OPEN_POWER_MEMCTRL_READ_TIMEOUT_SEC	Maximum time in seconds allowed without having a new centaur temperature before memory throttling will occur
OPEN_POWER_DIMM_THROTTLE_TEMP_DEG_C	DIMM Temperature to invoke memory throttling
OPEN_POWER_DIMM_ERROR_TEMP_DEG_C	DIMM Temperature in degrees C that a DIMM overtemp error will be logged calling out the DIMM
OPEN_POWER_DIMM_READ_TIMEOUT_SEC	Maximum time in seconds allowed without having a new DIMM temperature before memory throttling will occur
OPEN_POWER_GPU_ERROR_TEMP_DEG_C	GPU core Temperature in degrees C that a GPU overtemp error will be logged calling out the GPU
OPEN_POWER_GPU_READ_TIMEOUT_SEC	Maximum time in seconds allowed without having a new GPU core temperature before logging a communication error
OPEN_POWER_GPU_MEM_ERROR_TEMP_DEG_C	GPU memory Temperature in degrees C that a GPU overtemp error will be logged calling out the GPU
OPEN_POWER_GPU_MEM_READ_TIMEOUT_SEC	Maximum time in seconds allowed without having a new GPU memory temperature before logging a communication error
REPORT_THROTTLE_BELOW_NOMINAL	Indicates when OCC should report processor throttling to the BMC (poll response bits) and OPAL (shared memory throttle reason) 0 = OCC will report throttling when max frequency has been clipped below turbo 1 = OCC will report throttling only when max frequency has been clipped below nominal
SYSTEM_WOF_DISABLE	Indicates if WOF is supported. 0 = WOF is supported, max frequency will be ultra turbo 1 = WOF is disabled, max frequency will be turbo (if turbo is supported) or nominal (turbo not supported)
PBAX_CHIPID and PBAX_GROUPID	For PBAX (OCC-OCC communication) group ID should be 0 for all processors and chipID 0 for 1st processor and 1 for 2nd processor.
APSS Configuration	All attributes assigning the 16 ADC Channels and GPIOs must be defined.
NO_APSS_PROC_POWER_VCS_VIO_WATTS	Only needed for systems that do NOT have an APSS. This gives a constant processor power for processor components not measured by the OCC over AVSBus to give a more accurate total processor power.
FREQ_PROC_REFCLOCK_KHZ	Required by the PGPE

VDD_CURRENT_OVERFLOW_WORKAROUND_ENABLE	Used to enable Vdd current overflow worakaround required for Infineon chips Vdd current rolling over at a lower point than what is possible. 0 = disabled 1 = enabled
MAX_VDD_CURRENT_READING	Maximum theoretical Vdd current reading in 10mA units. Only used when Vdd current overflow workaround is enabled in order to detect valid Vdd rollover.

8.2 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 seen as hardware control will come from the MRW:

- APSS Channel and GPIO assignment
- WOF_ENABLED
- Memory configuration (Nimbus I2C DIMM info)
- PBAX Chip IDs

8.3 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. Changing between modes that is changing to a different WOF power table will require a reset of the PM complex. TMGT will handle the mode change as follows:

1. If WOF is supported in the new mode, determine the WOF power limit table (Nominal vs Turbo) required for the new mode defined in appendix C.

2. If WOF is supported in the new mode AND the WOF power limit table has changed:

    2.1 Update the WOF_POWER_LIMIT attribute with NOMINAL (0) or TURBO (1)

    2.2 Reset the PM complex

3. If the new or current mode is FMF: TMGT sends IPS disable (if new mode is FMF) or IPS enable setting (if current mode is FMF) to OCC and then TMGT sends new system power and performance mode in the set mode and state command to the OCC.

8.4 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. DCMI 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.

8.4.1 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.

8.4.2 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.

8.5 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

IPS enter and exit criteria are defined in the power management def file (FSP) or the xml file (BMC). The master OCC gets a 3 second average of all cores utilizations from all slaves every 250usec. 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.

8.5.1 FSP Systems

On FSP systems that support IPS a user may override the defaults via ASM. Whenever TMGT receives an IPS parm change from ASM TMGT will 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 TMGT 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. TMGT will be processing these bits (from master OCC poll response only) to determine what to do based on this table:

Idle Power Save Enable from TMGT (i.e. user)	***Master OCC Poll response Current Idle Power Saver Status Byte***		***Valid?***	***TMGT Actions***
	bit 1: Idle Power Saver Active	***bit 0: Idle Power Saver Enabled***		
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: ➢ TMGT will check if fan floor needs to change due to IPS no longer active ➢ TMGT sends “Proc Folding Data” to PHYP with mode modifier updated to indicate IPS is not active (turn off winkle)
1	1	0	N	Re-send OCC command to enable IPS, log error if this persists.
1	1	1	Y	On 0->1 transition of Active bit: ➢ TMGT 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. ➢ TMGT sends “IPS Active” for mode modifier in the Proc Folding Data command to PHYP. PHYP puts ceded cores into deep winkle and tells the o/s to enable folding (PHYP hack is to tell o/s “SPS” for mode as o/s does not understand IPS.

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.

9 Checkstop FIR Collection

When there is a checkstop the OCC will collect Fault Isolation Registers (FIR) and write the data to PNOR to aid in debug of the checkstop. The OCC that is monitoring for a checkstop and collecting the FIR data must be the OCC that has a connection to PNOR, this OCC is called the “FIR Master”. On power on Hostboot writes an area of HOMER to indicate if the OCC is FIR Master and the configuration data needed for the FIR collection.

During a BMC update the BMC owns PNOR and the OCC cannot write to PNOR. OCC capability to write to PNOR will be communicated by bit 18 of GP Register 0. This bit defaults to ‘0’ on a power on and ‘0’ will indicate that OCC is allowed to write to PNOR. After power on OPAL must update this bit to ‘1’ when giving BMC ownership of PNOR and back to ‘0’ when BMC no longer owns PNOR. No FIR data will be collected on a checkstop if the OCC fails to read this bit or this bit is a ‘1’.

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:

1. OCC sends PGPE IPC command Set Pstate Clips to set min/max clips back
2. OCC sends PGPE IPC command Write PMCR to set requested Pstate to Pmin (min clip)
3. OCC sends PGPE IPC command Control Pstate Protocol to suspend Pstate protocol
4. OCC sends PGPE IPC command Control Pstate Protocol to start Pstate protocol for OPAL owner
5. If WOF is enabled OCC sends PGPE IPC command WOF Control to turn WOF on
6. 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 Pstate Parameter Block.

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 legacy turbo
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 requires 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

10.4.2 Reading Voltage

Read voltages using one of the following:

A. OP Tool commands. This uses I2C interface.
B. I2C commands directly
C. If OCC in observation state or in reset: AVSbus bridge A. In observation state PGPE is not using bridge A
D. 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-OPAL Interface

11.1 OCC-OPAL 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)	Version (0x90)	OCC Role	Min Pstate	Nominal Pstate	Turbo Pstate	Ultra Turbo Pstate	Reserved
0x0008	Reserved							
0x0010	Reserved							
0x0018	Pstate number = 0	Flag	Reserved		Freq in kHz			
0x0020	Pstate number = 1	Flag	Reserved		Freq in kHz			
:	:	:	:		:			
0x0810	Up to 255 Pstates							
0x0818	TBD. Data to tell Pstate opportunity for WOF							
0x0820								
0x0828								
0x0830 :	848B Reserved for future “static” data							
0x0B80	OCC State	Dynamic Major Version	Dynamic Minor Version	GPUs Present	Reserve	Proc Throttle Status	Memory Throttle Status	Quick Power Drop
0x0B88	Power Shifting Ratio	Power Cap Type	Min Power Cap		Max Power Cap		Current Power Cap	
0x0B90	Soft Min Power Cap		110B Reserved					

Offset	*Byte 0*	*Byte 1*	*Byte 2*	*Byte 3*	*Byte 4*	*Byte 5*	*Byte 6*	*Byte 7*
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.

Version – Indicates format version = 0x90.

OCC Role – Indicates the role of the OCC.
0x00 = Slave.
0x01 = Master.

Min Pstate – Minimum Pstate ever allowed.

Nominal Pstate – Pstate for nominal.

Turbo Pstate – Turbo Pstate. NOTE: Current state of system may not allow for turbo Pstate to be reached, see Throttle Status. Turbo 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 turbo Pstate. NOTE: Current state of system may not allow for ultra turbo Pstate to be reached, see Throttle Status. Ultra Turbo Pstate field is not updated when throttling occurs.

Pstate Number / Flag / Frequency – Continuously numbered from max to min Pstate. Flag is reserved for future use 0x00 for now.

OCC State – Indicates the current state of the OCC. Used to know what OPAL-OCC cmd/rsp communication is allowed
0x00 = OCC not running. No communication allowed.
0x01 = Standby. No communication allowed.
0x02 = Observation State. Communication allowed and is command dependent.
0x03 = Active State. Communication allowed and is command dependent.
0x04 = Safe State. No communication allowed. Just like CPU throttle status, some failures will not allow for OCC to update state to safe.

0x05 = Characterization State. 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 existing fields is required the major version needs to be updated.

0x00 – Original minor version
0x01 – Support for GPUs Present field added

GPUs Present – Bit mask indicating GPUs present behind this OCC (processor) only.

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

Processor Throttle Status – Indicates reason that OCC may have limited Max Pstate. NOTE: 0x06/0x07/0x09/0xAA are for debug purposes only and should not result in errors. The report Pstate is either nominal or turbo based on xml attribute REPORT_THROTTLE_BELOW_NOMINAL

0x00 = No throttle
0x01 = Power Cap (max freq clipped below report Pstate)
0x02 = Processor Over Temperature (max freq clipped below report Pstate)
0x03 = Power Supply Failure (currently not used)
0x04 = Overcurrent (currently not used)
0x05 = OCC reset. Some failures will not allow for OCC to update throttle status.
0x06 = Exceeded Power Cap above throttle report Pstate (throttling is happening but not below the report Pstate)
0x07 = Processor Over Temperature above throttle report Pstate (throttling is happening but not below the report Pstate)
0x08 = VRM Vdd Over Temperature (max freq clipped below report Pstate)
0x09 = VRM Vdd Over Temperature above throttle report Pstate (throttling is happening but not below the report Pstate)
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

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

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)

11.2 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.

11.2.1 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 7 (msb)	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0 (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

11.2.2 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 7 (msb)	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0 (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

11.2.3 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.

11.2.4 Error Handling

11.2.4.1 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

11.2.4.2 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.

11.2.5 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

OPAL-OCC Command		***OCC State***	***Timeout***	***Supported By***	
				Master OCC	***Slave OCC***
0x41	AMESTER Pass Thru	O, A	1s	Y	Y
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

11.2.5.1 AMESTER Pass Thru

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:

OPAL-OCC Command Value	0x41
Data Length	Variable
Command Data	Command data from AMESTER:

Byte Offset	***Description***
0	AMESTER API command = 0x3C
1	Reserved = 0x00

2 thru end of cmd data	**Byte offset 2: AMESTER Sub-Command**	**Byte offset 3 thru end of buffer sub-command specific data**
	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 Buffer Recording	No data

		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:

OPAL-OCC Command Value	0x41
Response Status	0x00 = Success 0x11 = Invalid Command – in secure mode 0x15 = Internal OCC error
Data Length	Variable.
Response Data	Response specific data for AMESTER command:

Byte Offset	***Description***	
0	AMESTER API command = 0x3C	
1	Reserved = 0x00	
2 thru end of rsp data	**AMESTER Sub-Command**	**Sub-command specific response data**
	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 Buffer	No response data

		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

11.2.5.2 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:

OPAL-OCC Command Value	0xD0	
Data Length	0x0004	
Command Data	**Byte 1 (first)**	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
	Byte 2	Reserved = 0x00
	Byte 3	Reserved = 0x00
	Byte 4 (last)	Reserved = 0x00

Clear Sensor Data Response:

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

11.2.5.3 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:

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

Set Power Cap in Band Response:

OPAL-OCC Command Value	0xD1	
Response Status	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	
Data Length	0x0002	
Response Data	**Bytes 1-2**	Current power cap. (MSB first) On a successful set this should match the power cap sent in the command data

11.2.5.4 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:

OPAL-OCC Command Value	0xD2	
Data Length	0x0001	
Command Data	**Byte 1**	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:

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

11.2.5.5 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:

OPAL-OCC Command Value	0xD3	
Data Length	0x0002	
Command Data	**Bytes 1-2**	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:

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

11.3 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

Start (Offset from BAR2 base address)	***End***	***Size***	***Description***
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	0x006FFFFF	336kB	Reserved

*OCC number is the PBAX chip ID (rcv_chipid field in the PBAX Configuration Register)

11.3.1 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.

Start (Offset from OCC N Sensor Data Block)	***End***	***Size***	***Description***
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

11.3.1.1 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

11.3.1.2 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: 0x0001 = Generic, 0x0008 = Temp, 0x0040 = Freq 0x0002 = Current, 0x0010 = Util, 0x0080 = Power 0x0004 = Voltage, 0x0020 = Time, 0x0200 = Perform
Location	2	Sensor location: 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

11.3.1.3 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 is 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.

11.3.1.3.1 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
Reserved	8	

11.3.1.3.2 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	

11.3.2 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..23 and quad number 0..5 in a sensor name represents physical core and quad (cache) number from CORE or CACHE SCOM 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
- Rows in red are currently NOT supported by the OCC additional code needed to support the sensor before it may be included in main memory

11.3.2.1 Performance Sensors

Sensor Name	SMF Mode?	Master only?	#	Unit	Type	Loc	Sample time	Scale Factor	Description
IPS	N	N	1	MIP	PERF	PROC	8MS	1	Vector sensor that takes the average of all the cores in Processor x
STOPDEEPACTCy	Y	N	24	ss	PERF	CORE	8MS	1	Deepest actual stop state that was fully entered during sample time for core y
STOPDEEPREQCy	Y	N	24	ss	PERF	CORE	8MS	1	Deepest stop state that has been requested during sample time for core y
IPSCy	N	N	24	MIP	PERF	CORE	8MS	1	Instructions per second for core y on this Processor
NOTBZECy	N	N	24	cyc	PERF	CORE	8MS	1	Not Busy (stall) cycles counter for core y on this Processor
NOTFINCy	N	N	24	cyc	PERF	CORE	8MS	1	Not Finished (stall) cycles counter for core y on this Processor
MRDMx	N	N	8	GBs	PERF	MEM	4MS	0.00064	Memory read requests per sec for MCA x (0..7)
MWRMx	N	N	8	GBs	PERF	MEM	4MS	0.00064	Memory write requests per sec for MCA x (0..7)
MEMSPSTATMx	Y	N	8	%	UTIL	MEM	4MS	0.1	Static Memory throttle level setting for MCA x (0..7) when not in a memory throttle condition
MEMSPMx	N	N	8	%	UTIL	MEM	4MS	0.1	Current Memory throttle level setting for MCA x (0..7)
PROCPWRTHROT	N	N	1	#	PERF	PROC	500US	1	Count of processor throttled due to power
PROCOTTHROT	N	N	1	#	PERF	PROC	32MS	1	Count of processor throttled for temperature
MEMPWRTHROT	N	N	1	#	PERF	MEM	500US	1	Count of memory throttled due to power

MEMOTTHROT	N	N	1	#	PERF	MEM	32MS	1	Count of memory throttled due to memory Over temperature

11.3.2.2 Power Sensors

Sensor Name	SMF Mode?	Master only?	#	Unit	Type	Loc	Sample time	Scale Factor	Description
PWRSYS	Y	Y	1	W	POWER	SYS	500US	1	Bulk power of the system
PWRGPU	Y	N	1	W	POWER	GPU	500US	1	Power consumption for GPUs per socket (OCC) read from APSS
PWRAPSSCHx	Y	Y	16	W	POWER	SYS	500US	1	Power Provided by APSS channel x (where x=0…15)
PWRPROC	Y	N	1	W	POWER	PROC	500US	1	Power consumption for this Processor
PWRVDD	Y	N	1	W	POWER	PROC	1MS	1	Power consumption for this Processor's Vdd (calculated from AVSBus readings)
PWRVDN	Y	N	1	W	POWER	PROC	1MS	1	Power consumption for this Processor's Vdn (nest) (calculated from AVSBus readings)
PWRMEM	Y	N	1	W	POWER	MEM	500US	1	Power consumption for Memory for this Processor read from APSS

11.3.2.3 Frequency Sensors

Sensor Name	SMF Mode?	Master only?	#	Unit	Type	Loc	Sample time	Scale Factor	Description
FREQA	Y	N	1	MHz	FREQ	PROC	8MS	1	Average of all core frequencies for Processor
FREQACy	Y	N	24	MHz	FREQ	CORE	8MS	1	Average/actual frequency for this processor, Core y based on OCA data

11.3.2.4 Utilization Sensors

Sensor Name	SMF Mode?	Master only?	#	Unit	Type	Loc	Sample time	Scale Factor	Description
UTILCy	N	N	24	%	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
UTIL	N	N	1	%	UTIL	PROC	8MS	0.01	Average of all Cores UTILCy sensor
NUTILCy	N	N	24	%	UTIL	CORE	3S	0.01	Normalized average utilization, rolling average of this Processor's Core y

11.3.2.5 Temperature Sensors

Sensor Name	SMF Mode?	Master only?	#	Unit	Type	Loc	Sample time	Scale Factor	Description
TEMPNEST	N	N	1	C	TEMP	PROC	8MS	1	Average temperature of nest DTS sensors

TEMPPROCTHRMCy	N	N	24	C	TEMP	CORE	8MS	1	The combined weighted core/quad temperature for processor core y
TEMPVDD	N	N	1	C	TEMP	VRM	3MS	1	VRM Vdd temperature
TEMPDIMMx	N	N	16	C	TEMP	MEM	64MS	1	DIMM temperature for DIMM x
TEMPGPUx	?	N	3	C	TEMP	GPU	1S	1	GPU x (0..2) board temperature
TEMPGPUxMEM	?	N	3	C	TEMP	GPU	1S	1	GPU x hottest HBM temperature (individual memory temperatures are not available)

11.3.2.6 Voltage Sensors

Sensor Name	SMF Mode?	Master only?	#	Unit	Type	Loc	Sample time	Scale Factor	Description
VOLTVDD	Y	N	1	mV	VOLTAGE	VRM	1.5MS	0.1	Processor Vdd Voltage (read from AVSBus)
VOLTVDDSENSE	Y	N	1	mV	VOLTAGE	PROC	1.5MS	0.1	Vdd Voltage at the remote sense. (AVS reading adjusted for loadline)
VOLTVDN	Y	N	1	mV	VOLTAGE	VRM	1.5MS	0.1	Processor Vdn Voltage (read from AVSBus)
VOLTVDNSENSE	Y	N	1	mV	VOLTAGE	PROC	1.5MS	0.1	Vdn Voltage at the remote sense. (AVS reading adjusted for loadline)
VOLTDROOPCNTCx	Y	N	24	#	VOLTAGE	CORE	8MS	1	Small voltage droop count for core x
VOLTDROOPCNTQx	Y	N	6	#	VOLTAGE	QUAD	8MS	1	Large voltage droop count for quad x

11.3.2.7 Current Sensors

Sensor Name	SMF Mode?	Master only?	#	Unit	Type	Loc	Sample time	Scale Factor	Description
CURVDD	Y	N	1	A	CURRENT	VRM	1.5MS	0.01	Processor Vdd Current (read from AVSBus)
CURVDN	Y	N	1	A	CURRENT	VRM	1.5MS	0.01	Processor Vdn Current (read from AVSBus)

11.3.3 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
PWRFAN	1	W	POWER	SYS	500US	1	Power consumption of the system fans
PWRIO	1	W	POWER	SYS	500US	1	Power consumption of the IO subsystem (including storage digital 5Volt or less rail)
PWRSTORE	1	W	POWER	SYS	500US	1	Power consumption of the storage subsystem (storage 12Volt rail)
FREQREQCy	24	MHz	FREQ	CORE	8MS	1	Requested frequency from OCC voting box for this processor, Core y
TEMPCy	24	C	TEMP	CORE	8MS	1	Average temperature of core DTS sensors for Processor's Core y
TEMPQy	6	C	TEMP	QUAD	8MS	1	Average temperature of quad (in cache) DTS sensors for Processor's Quad y
TEMPPROCTHRM	1	C	TEMP	PROC	8MS	1	Maximum of all TEMPPROCTHRMCy core temperatures
TEMPPROCAVG	1	C	TEMP	PROC	8MS	1	Average of all TEMPPROCTHRMCy core temperatures
TODclock0	1	us	TIME	SYS	8MS	16	TOD clock Low 16 bits in 16us resolution
TODclock1	1	s	TIME	SYS	8MS	1.048576	TOD clock mid 16 bits in 1.05s resolution
TODclock2	1	day	TIME	SYS	8MS	0.795364	TOD clock high 3 bits in 0.796 day resolution
CUR12VSTBY	1	A	CURRENT	SYS	500US	.01	12V standby current read from APSS channel (if channel assigned for 12V standby current)
VRHOTMEMPRCCNT	1	#	GENERIC	SYS	500US	1	Memory VR HOT read from APSS GPIO(s). This is info only not used in any control loops. The accumulator is a count of number of times this was asserted (updated every sample time) '1' = one of the procs memory VR_HOT is asserted '0' = no Memory VR_HOT asserted
MRDMy	8	GBs	PERF	MEM	4MS	0.00128	Memory read requests per second for memory controller y (0..7)
MWRMy	8	GBs	PERF	MEM	4MS	0.00128	Memory write requests per second for memory controller y (0..7)
MIRCMy	8	eps	PERF	MEM	4MS	1	0.01 Events/second. Memory Inter-request arrival idle interval longer than programmed threshold for memory controller y (0..7)
MLP2My	8	eps	PERF	MEM	4MS	1	(events/s) Number of LP2 exits for memory controller y (0..7)
TEMPDIMMAXMy	8	C	TEMP	MEM	4MS	1	CENTAUR ONLY! Max DIMM temperature for Centaur y

LOCDIMMAXMy	8	loc	TEMP	MEM	4MS	1	CENTAUR ONLY! Location of max DIMM temperature for Centaur y
MACMxCyPz	16	rps	PERF	MEM	4MS	1	Memory activation requests per sec for Centaur x, Channel y, port z
MPUMxCyPz	16	rps	PERF	MEM	4MS	1	Memory power-up requests per sec for Centaur z, channel y, port z
MIRBMxCyPz	16	eps	PERF	MEM	4MS	1	Memory Inter-request arrival idle intervals for Centaur z, channel y, port z
MIRLMxCyPz	16	eps	PERF	MEM	4MS	1	Memory Inter-request arrival idle intervals longer than low threshold for Centaur z, channel y, port z
MIRMMxCyPz	16	eps	PERF	MEM	4MS	1	Memory Inter-request arrival idle intervals longer than medium threshold for Centaur z, channel y, port z
MIRHMxCyPz	16	eps	PERF	MEM	4MS	1	Memory Inter-request arrival idle intervals longer than high threshold for Centaur z, channel y, port z
MTSMxCyPz	16	cnt	PERF	MEM	4MS	1	Last received Timestamp (frame count) from Centaur z, channel y, port z
M4RDMxCyPz	16	GBs	PERF	MEM	4MS	1	Memory cached (L4) read requests per sec for Centaur z, channel y, port z
M4WRMxCyPz	16	GBs	PERF	MEM	4MS	1	Memory cached (L4) write requests per sec for Centaur z, channel y, port z
TEMPCENT	1	C	TEMP	MEM	4MS	1	CENTAUR ONLY! Hottest Centaur temperature.
TEMPDIMMTHRM	1	C	TEMP	MEM	64MS	1	Hottest DIMM temperature

11.4 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.

11.4.1 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

11.4.2 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.

11.4.2.1 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	**OCC-HTMGT Service Required Interrupt** (defined in P8 same usage in P9) – host to call HTMGT interface with OCC chip ID
2	**I2C Master Ownership Change** – host to read PIB I2CM engine locks in OCC Flags register
3	**OCC-OPAL Shared Memory Interface Change** – Re-read and process changes in dynamic shared memory interface i.e. OCC command response, throttle status

11.4.3 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.

11.4.3.1 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.

11.4.3.2 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.

11.4.3.3 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.

11.4.3.4 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.

11.5 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.

11.5.1 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 2nd OCC these are physical GPUs 3, 4, 5 respectively.

APSS GPIO	***OCC FLAGS REGISTER***
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

Return Code	*Description*
0xFF	**Command In Progress.** Command is being processed and the response buffer is not valid.
0x00	**Success.** Command completed normally
0x11	**Invalid Command.** The command type is invalid or unsupported. • i.e. Slave OCC receiving a command that is supported by master only
0x12	**Invalid Command Length.** The command data length is invalid for the particular command.
0x13	**Invalid Data Field.** The command data has an invalid value for a field. • i.e. Poll version not supported
0x14	**Checksum Failure.** The command packet checksum is not correct.
0x15	**Internal OCC error.** An error occurred within OCC to prevent the command from being processed but the OCC is still running and the command may be retried.
0x16	**Present State Prohibits.** The OCC cannot execute the command in its present state. • OCC is not in a state that the command requires
0xE0 thru 0xEF	**Critical OCC error.** 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: • Generate an error log including the full Rsp Data buffer to capture info for debug. • 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. **0xE0 → OCC Exception.** An Unrecoverable OCC exception. i.e. SSX panic. **0xE1 → OCC Initialization Checkpoint.** Indicates how far into initialization OCC got before it died, typically this will never be seen as the reason to fail initialization should result in a different reason code

Return Code	*Description*
	0xE2 → Watchdog Timeout. Halt due to OCC watchdog expiring. **0xE3 → OCB Timeout.** Halt due to OCB timer expiring. **0xE4 → OCC HW Error.** Halt due to an OCC hardware error. OCCLFIR bit being set (OISR0 bit 2)

Appendix B. OCC States

OCC State	*Description*
0x00	**Reserved.** This value is reserved for command data to indicate no change to current OCC state.
0x01	**Standby** ▪ The OCC is ready to handle commands from HTMGT ▪ If OCC is FIR master it will monitor for and collect FIRs due to a checkstop ▪ No communication allowed from BMC ▪ No monitoring or actuation done by OCC ▪ 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 ▪ HTMGT will never tell OCC to move to this state
0x02	**Observation** ▪ Full communication with HTMGT and BMC; some commands may be rejected if only supported in Active state. ▪ OCC is monitoring only, no DVFS/throttling actuation is done due to power or thermal ▪ Maximum Pstate clip is set to legacy turbo ▪ WOF is disabled ▪ Pstate protocol is disabled
0x03	**Active** ▪ This is the full function state ▪ Full communication with HTMGT and BMC ▪ OCC will monitor all sensors and actuate to maintain power and thermal limits
0x04	**Safe** ▪ This is NOT safe mode ▪ 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 ▪ 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 ▪ This is a state while OCC is waiting for a reset (safe mode) ▪ Sensor data is not updated while in this state ▪ OCC will stop poking watchdogs to allow system to drop v/f and memory throttles ▪ The OCC will continue to communicate with HTMGT and BMC for error

OCC State	*Description*
	logging purposes ▪ HTMGT will never tell OCC to move to this state
0x05	**Characterization** ▪ OCC treats this the same as observation state – full communication with FSP/BMC, monitoring only ▪ OCC sets min/max Pstate clips wide open to allow full frequency range ▪ WOF is disabled ▪ Pstate protocol is enabled with characterization as PMCR owner ▪ Characterization will be writing the PMCR directly to set pStates

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	**Reserved.** This value is reserved for command data to indicate no change to current system power and performance mode.
0x01	**Nominal.** ▪ User Settable ▪ Frequency and voltages are fixed at nominal ▪ WOF is off ▪ 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 in nominal
0x02	**Reserved.**
0x03	**Turbo. aka "Static Turbo"** ▪ Lab only mode to test the legacy turbo v/f point and is NOT user settable ▪ Frequency and voltages is pinned to a defined point calculated as a percentage above nominal. See turbo frequency in the frequency data point config packet ▪ WOF is off ▪ 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 turbo point
0x04	**Safe.** ▪ NOT user settable ▪ OCC is non-functional and held in reset. The whole power management complex (all OCCs, PGPEs and SGPEs) is held in reset ▪ 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 ▪ An exit from safe mode will be attempted after a re-IPL, FSP reset or power cycle
0x05	**Static Power Save** ▪ User settable ▪ Frequency and voltages are fixed to a defined point calculated as a percentage below nominal defined in the def file/MRW ▪ WOF is off

Mode Value	Description
0x06	**Reserved. (Was previously Dynamic Power Save – Favor Energy)**
0x07	**Reserved.**
0x08	**Reserved.**
0x09	**Reserved.**
0x0A	**Dynamic Performance (aka DPS – Favor Performance)** ▪ User settable ▪ OCC lowers frequency if idle for a period of time otherwise maximum frequency ▪ WOF is on using the nominal power limit tables
0x0B	**Fixed Frequency Override (FFO)** ▪ User settable. Limited to legacy turbo. ▪ OCC pins frequency to the user specified FFO frequency ▪ WOF is off ▪ If (FFO frequency <= nominal frequency) ====> treat FFO the same as nominal mode: Use non power management thermal thresholds. 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. ▪ If (FFO frequency > nominal frequency) ====> treat FFO same as a power management mode: Use power management thermal thresholds. 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 nominal.
0x0C	**Maximum Performance (aka Fixed Maximum Frequency)** ▪ User settable ▪ Frequency is at max as long as power and thermal limits allow. ▪ Frequency is NOT lowered due to utilization ▪ WOF is on using the turbo power limit tables ▪ P9: IPS may be enabled while in this mode

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

Appendix E. Other BMC Requirements

E.1 User Interface to Send an OCC Command

The BMC must provide a user interface for a user to be able to send any command to a specified OCC. The user will send in which OCC the command is for and the command buffer from the command type thru the specific command data.

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

When received:

1. The BMC must fill in the sequence number and checksum
2. Send the command to the specified OCC following the full OCC Command/Response Sequence for a command coming from the BMC
3. Send back the response buffer from the OCC to the user or error code if command failed

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

NOTES:

1. The OCC can only process one command at a time; the BMC must ensure that this is the only command the BMC has in process to the OCC.
2. The BMC implementation does not require data length to be sent, the BMC calculates the data length, the occ numbering starts from 1 for the first physical OCC:

ipmitool –H <bmc ip> –U <usr> –P <pwd> raw 0x3a 0x0d <occ number> <cmd type> <data>