OCC Firmware Interface Specification for Open Power

Version 1.3

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

Change History

Date	*Change Description*
10/28/2014	*Version 1.1 – First document version*
02/10/2015	*Version 1.2 – Updates for PNOR ownership for FIR collection on check stop*
2/24/2016	*Version 1.3 – Updates for ultra turbo (WOF)*

Copyright and Disclaimer

All information contained in this document is subject to change without notice. The products described in this document are NOT intended for use in applications such as implantation, life support, or other hazardous uses where malfunction could result in death, bodily injury, or catastrophic property damage. The information contained in this document does not affect or change IBM product specifications or warranties. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of IBM or third parties. All information contained in this document was obtained in specific environments, and is presented as an illustration. The results obtained in other operating environments may vary. While the information contained herein is believed to be accurate, such information is preliminary, and should not be relied upon for accuracy or completeness, and no representations or warranties of accuracy or completeness are made.
Note: This document contains information on products in the design, sampling and/or initial production phases of development. This information is subject to change without notice. Verify with your IBM field applications engineer that you have the latest version of this document before finalizing a design.
THE INFORMATION CONTAINED IN THIS DOCUMENT IS PROVIDED ON AN "AS IS" BASIS. In no event will IBM be liable for damages arising directly or indirectly from any use of the information contained in this document.

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

The IBM home page can be found at ibm.com®.

Table of Contents

1 Overview

This document covers the firmware interfaces to the OCC for 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). Specifically, the thermal management code piece that runs on the processor and initializes and handles errors from the OCC.

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

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 for BMC fan control
    - Throttle memory if a memory temperature reaches a specified throttle temperature point
    - Lower Pmax register to have the chip lower the 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 Pmax register to have the chip lower the 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
- 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 lowering Pmax due to power or thermal reason.
- Provide power, thermal and frequency sensor data to the BMC 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. Host boot will configure all channels and some additional setup is done by the OCC to support a circular channel. 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	HTMGT use only. Used by HTMGT to read OCC error logs from SRAM.
1	Circular	Write only from BMC 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 OCC 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.

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 – Reserved (FSP)
0x10 – HTMGT
0x20 – BMC

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

1.5.2 OCC to Host

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

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

1. OCC sets bits 0 and 1 of OCB_OCI_OCCMISC SCOM register.
   - Bit 0 (OCB_OCI_OCCMISC_CORE_EXT_INTR) to generate the interrupt
   - Bit 1 to indicate source/reason 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 BMC to Host

The P8 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.

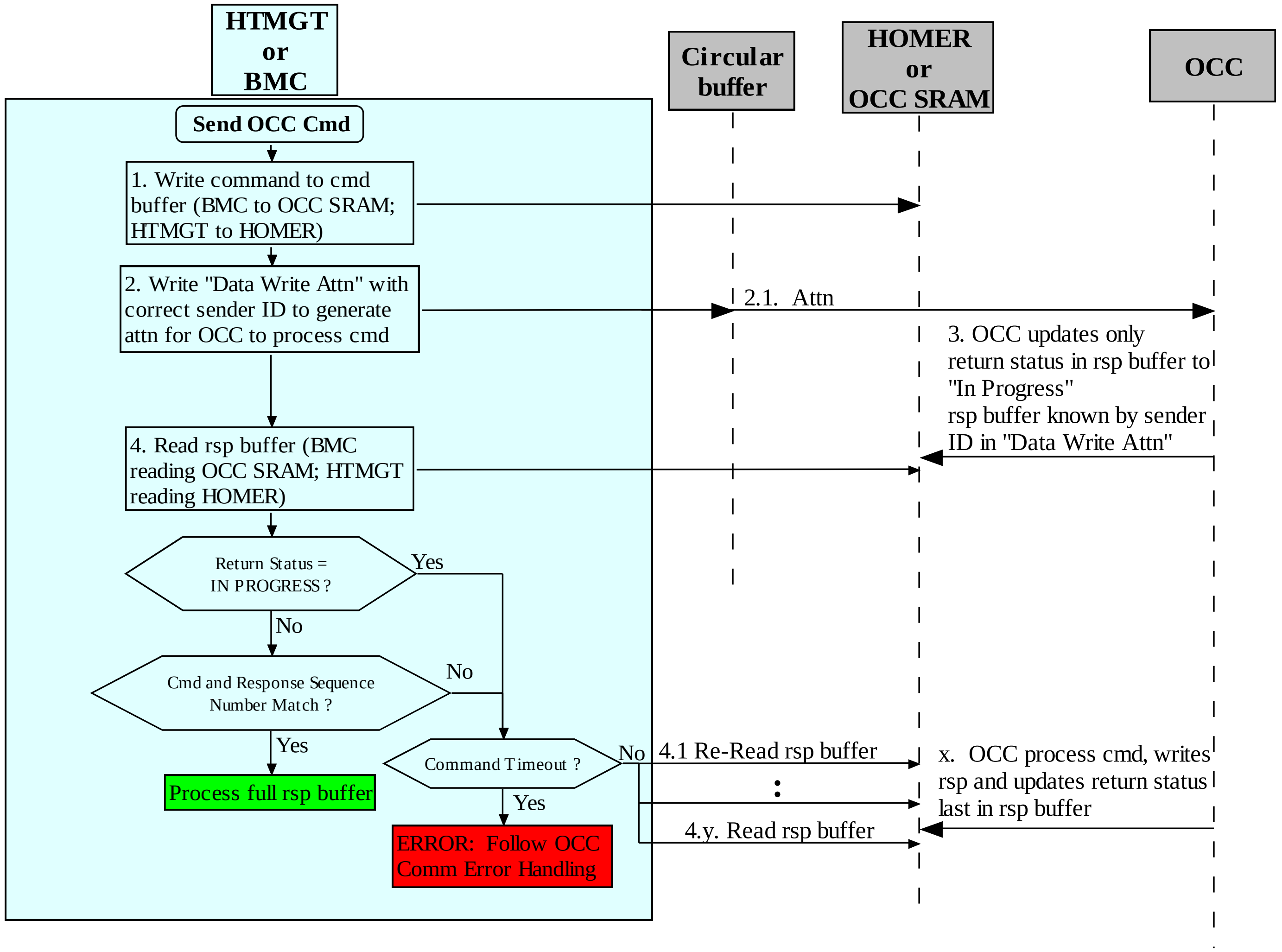
1.6.1 Command and Response Buffer Locations

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

	Command Buffer	***Response Buffer***
BMC	OCC SRAM 0xFFFF6000 – 0xFFFF6FFF	OCC SRAM 0xFFFF7000 – 0xFFFF7FFF
HTMGT	HOMER 0x001EE000 – 0x001EEFFF	HOMER 0x001EF000 – 0x001EFFFF

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.


HTMGT or BMC
Circular buffer
HOMER or OCC SRAM
OCC
Send OCC Cmd
1. Write command to cmd buffer (BMC to OCC SRAM; HTMGT to HOMER)
2. Write "Data Write Attn" with correct sender ID to generate attn for OCC to process cmd
2.1. Attn
3. OCC updates only return status in rsp buffer to "In Progress"
rsp buffer known by sender ID in "Data Write Attn"
4. Read rsp buffer (BMC reading OCC SRAM; HTMGT reading HOMER)
Return Status = IN PROGRESS ?
Yes
No
Cmd and Response Sequence Number Match ?
No
Yes
Process full rsp buffer
Command Timeout ?
No
Yes
ERROR: Follow OCC Comm Error Handling
4.1 Re-Read rsp buffer
:
4.y. Read rsp buffer
x. OCC process cmd, writes rsp and updates return status last in rsp buffer

1.6.3 Command Format

The command format is the same regardless of who the sender is (BMC or HTMGT).

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 or HTMGT).

Format:

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	Checksum MSB	Checksum 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. HTMGT will 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 HTMGT will 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

Recommended time to wait for the OCC to respond before taking error action. This is time from when the data write attention is sent to the OCC. The OCC can only process one command at a time across all senders. This timeout includes the worst case time for the longest processing command to handle when a command from a different sender is being processed first.

OCC State

- Defines OCC states that the command is supported in
  - Sby = Standby
  - O = Observation
  - 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 Summary		***Sender***	***Timeout***	***OCC State***	***Supported By***	
					Master OCC	***Slave OCC***
Poll	Poll the OCC for status and sensor data.	BMC HTMGT	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	20s	Sby, O, A, Safe	Y	Y
Set State	Set the OCC state	HTMGT	20s	Sby, O, A	Y	N
Config Data	0x01: Pstate Table	HTMGT	20s	Sby, O	Y	Y
	0x02: Frequency Points	HTMGT	20s	Sby, O	Y	N
	0x03: Set OCC Role	HTMGT	20s	Sby	Y	Y
	0x04: APSS Config	HTMGT	20s	Sby, O, A	Y	Y
	0x05: Memory Config.	HTMGT	20s	Sby, O, A	Y	Y
	0x07: Power Cap Data	HTMGT	20s	Sby, O, A	Y	N
	0x0F: System Config.	HTMGT	20s	Sby, O, A	Y	Y
	0x12: Memory Throttles	HTMGT	20s	Sby, O, A	Y	Y
	0x13: Thermal Thresholds	HTMGT	20s	Sby, O, A	Y	Y

Command Summary		Sender	Timeout	OCC State	Supported By	
					Master OCC	Slave OCC
	0x30: WOF Core Frequency Data	HTMGT	20s	Sby, O	Y	Y
	0x31: WOF VRM Efficiency Data	HTMGT	20s	Sby, O	Y	Y
Set User Cap	Set a User Power Cap	BMC HTMGT	20s	Sby, O, A	Y	N
Reset Prep	Prepare OCC to be reset	HTMGT	20s	Sby, O, A, Safe	Y	Y
Get Field Debug Data	Used to collect additional data from OCC for hw errors detected by host.	HTMGT	20s	Sby, O, A, Safe	Y	Y

2 Command Definitions

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

2.1 Poll

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

BMC	Version 0x10 – Status and Sensor Poll. The BMC will send this periodically to read sensor data and verify that the OCC is functional.
HTMGT	Version 0x10 – 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. **0x10** = Status and Sensor Poll
Checksum	xxxx

Poll Return Packet:

<table>
<tr><td><b><i>Sequence Number</i></b></td><td>Xx</td></tr>
<tr><td><b><i>Command Type</i></b></td><td>0x00</td></tr>
<tr><td><b><i>Return Status</i></b></td><td>0x00 = Success
See Appendix A for list of all non-successful return codes.</td></tr>
<tr><td><b><i>Data Length</i></b></td><td>Variable. Minimum of 40 bytes to maximum of 4089.</td></tr>
<tr><td><b><i>Data</i></b></td><td><b>Version = 0x10</b> See Version 0x10 Poll Response Data Definition section for details.
<table>
<tr><th><i>Byte 1</i></th><th><i>Byte 2</i></th><th><i>Byte 3</i></th><th><i>Byte 4</i></th><th><i>Byte 5</i></th><th><i>Byte 6</i></th><th><i>Byte 7</i></th><th><i>Byte 8</i></th></tr>
<tr><td>Status</td><td>Ext Status</td><td>OCCs present</td><td>Config data</td><td>OCC State</td><td colspan="2">Reserved = 0x0000</td><td>Error Log ID</td></tr>
<tr><td colspan="4">Error Log Start Address</td><td colspan="2">Error Log Length</td><td colspan="2">Reserved = 0x0000</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><b><i>Checksum</i></b></td><td>xxxx</td></tr>
</table>

2.1.1 Version 0x10 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	FIR master					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.

FIR Master – 1 indicates that this OCC is the FIR master and will monitor for 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 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

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

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 OT	DVFS due to power	Mem Throttle OT	Quick Power Drop				

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

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

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 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 yet.
>
> *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 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 Handling: This byte will be checked on the first poll after sending a set state command. HTMGT 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.

Bytes 6-7: Reserved = 0x0000

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 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 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 Handling: Full support to collect and report OCC error log

BMC Handling: None

Byte 15-16: Reserved = 0x0000

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 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. Only master OCC will 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 HTMGT 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.
HTMGT	Sent after HTMGT 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 OCC State

BMC	Should never send.
HTMGT	HTMGT will send this as part of booting the OCCs or when a state change request is made thru Linux for manufacturing testing.

This command is used to set the OCC state.

- 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 HTMGT until all OCCs have finished the state change.
- A failure to change state by any OCC should result in a non-successful return code and an error log generated from the failing OCC to be collected by HTMGT. HTMGT 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 can either stay in the new state or fall back to previous state.

Set OCC 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:** Reserved = 0
Checksum	xxxx

Set OCC 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.
HTMGT	HTMGT will send this as part of booting the OCCs

Configuration Data Command Packet:

<table>
<tr><td><b><i>Sequence Number</i></b></td><td>xx</td></tr>
<tr><td><b><i>Command Type</i></b></td><td>0x21</td></tr>
<tr><td><b><i>Data Length</i></b></td><td>Dependent on Data Being Sent. See following sections for more details specific to each format.</td></tr>
<tr><td><b><i>Data</i></b></td><td>There are x bytes of data (not to exceed maximum):

<b>Byte 1:</b> Format – Indicates what format (i.e. Type of configuration data) the following command data is. See following sections for more details specific to each format. Some formats are only supported by the master OCC and the master OCC is responsible for broadcasting the information to all the slave OCCs:
<table>
<tr><th></th><th><i>Master OCC</i></th><th><i>Slave OCC</i></th></tr>
<tr><td><b>0x01: Pstate</b></td><td colspan="2">Required for observation sent from HTMGT</td></tr>
<tr><td><b>0x02: Frequency</b></td><td>Required for active sent from HTMGT</td><td>Required for active <b>from master OCC</b></td></tr>
<tr><td><b>0x03: OCC Role</b></td><td colspan="2">Required for observation sent from HTMGT</td></tr>
<tr><td><b>0x04: APSS Cfg</b></td><td colspan="2">Required for observation sent from HTMGT</td></tr>
<tr><td><b>0x05: Memory Cfg</b></td><td colspan="2">Required for observation sent from HTMGT</td></tr>
<tr><td><b>0x07: Pcap</b></td><td>Required for active sent from HTMGT</td><td>Required for active <b>from master OCC</b></td></tr>
<tr><td><b>0x0F: System Cfg</b></td><td colspan="2">Required for observation sent from HTMGT</td></tr>
<tr><td><b>0x12: Mem Throttle</b></td><td colspan="2">Optional for active state sent from HTMGT. Special handling to determine if required based on memory config data packet.</td></tr>
<tr><td><b>0x13: Thermal</b></td><td colspan="2">Required for observation sent from HTMGT</td></tr>
<tr><td><b>0x30: WOF Core Frequency Data</b></td><td colspan="2" rowspan="2">Required for active state sent from HTMGT.</td></tr>
<tr><td><b>0x31: WOF VRM Efficiency Data</b></td></tr>
</table>
</td></tr>
<tr><td><b><i>Checksum</i></b></td><td>xxxx</td></tr>
</table>

2.4.1 Pstate Super Structure (Format = 0x01)

Host boot reads #V and writes attributes needed for generating Pstate table procedure. HTMGT to call interface to run procedure to generate Pstate table. HTMGT does no data, version or length checking on the Pstate structure received; it is sent to the OCC exactly as it was received.

Data Length	x (not to exceed maximum)
Data	**Byte 1:** Format = 0x01 **Bytes 2-4:** Reserved = 0x000000 **Bytes 5-x:** Pstate Super Structure data

2.4.2 Frequency Points (Format = 0x02)

This gives the minimum and maximum frequency range and nominal operating point.

Data Length	0x0008 (version 0x10) or 0x000A (version 0x11)
Data	**Byte 1:** Format = 0x02 **Byte 2:** Version **<u>Version 0x10</u>** **Bytes 3-4: Nominal Frequency Point**. In MHz; MSB first. HTMGT reads from ATTR_NOMINAL_FREQ_MHZ **Bytes 5-6: Maximum Frequency Point.** In MHz; MSB first. Highest frequency ever allowed. HTMGT read OPEN_POWER_TURBO_MODE_SUPPORTED from xml file to indicate if frequencies above nominal (aka Turbo) are supported or not. If supported HTMGT reads and sends ATTR_FREQ_CORE_MAX (VPD max frequency) to the OCC, if 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 due to thermal or power reasons. HTMGT reads from ATTR_MIN_FREQ_MHZ **<u>Version 0x11</u>** **Bytes 3-4: Nominal Frequency Point**. In MHz; MSB first. HTMGT 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. HTMGT read OPEN_POWER_TURBO_MODE_SUPPORTED from xml file to indicate if frequencies above nominal (aka Turbo) are supported or not. If supported HTMGT reads and sends ATTR_FREQ_CORE_MAX (VPD max frequency) to the OCC, if 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 due to thermal or power reasons. HTMGT 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). 0x0000 indicates that WOF is not supported. HTMGT will first read xml file ATTR_WOF_ENABLED to determine if WOF is enabled. If WOF is not enabled HTMGT will send 0x0000 for ultra turbo frequency. Else HTMGT sends ATTR_ULTRA_TURBO_FREQ_MHZ to the OCC, which is written by HB and will be 0x0000 if any chip is missing the ultra turbo VPD point. WOF is not supported if any processor is missing ultra turbo VPD. This will also be 0x0000 if xml file indicates that frequencies above nominal are not supported

2.4.3 Set OCC Role (Format = 0x03)

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

- HTMGT knows which OCC is the master from the xml file OCC_MASTER_CAPABLE. 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

<table>
<tr><td>***Data Length***</td><td>0x0004</td></tr>
<tr><td>***Data***</td><td>**Byte 1:** Format = 0x03
Byte 2: OCC Role – 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></td><td></td><td></td><td></td><td></td><td></td><td></td><td>Master OCC</td></tr>
</table>

Master OCC – ‘1’ indicates that the OCC should run as the master OCC. ‘0’ indicates the OCC should run as a slave only.

Bytes 3-4: Reserved = 0x0000</td></tr>
</table>

2.4.4 APSS Configuration (Format = 0x04)

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

<table>
<tr><td><b><i>Data Length</i></b></td><td>0x00D8</td></tr>
<tr><td><b><i>Data</i></b></td><td>
<b>Byte 1:</b> Format = 0x04
<b>Byte 2:</b> Version = 0x10
<b>Bytes 3-4:</b> Reserved. 0x0000 will be sent.

Format of one ADC channel info data set:
<table>
<tr><td><b><i>Data byte x</i></b></td><td><b>ADC Channel Assignment.</b> Enum to identify what the power reading is for. ADC_CHANNEL_ID in xml file defined values 0 indicates not used:
<b>1</b>=Memory Proc0
<b>:</b>
<b>4</b>=Memory Proc3
<b>5</b>=Processor0
<b>:</b>
<b>8</b>=Processor3
<b>9</b>=Processor0 pcie
<b>:</b>
<b>12</b>=Processor3 pcie
<b>13-15</b>=IO A, B, C
<b>16-17</b>=Fans A, B
<b>18-19</b>=Storage A, B
<b>20</b>=12V Sense
<b>21</b>=Ground
<b>22</b>=Total system power
<b>23</b>=Memory Cache (Centaur)
<b>24</b>=GPU
<b>25-27</b>=Additional channels for Memory Proc0</td></tr>
<tr><td><b><i>Data byte x+1 and x+2</i></b></td><td><b>Sensor ID.</b> If IPMI supported IPMI_SENSOR_ID for channel defined in xml. Used by OCC for reporting channel power.</td></tr>
<tr><td><b><i>Data byte x+3</i></b></td><td><b>Ground Select.</b> ADC_CHANNEL_GROUND in xml file.</td></tr>
<tr><td><b><i>Data byte x+4 thru x+7</i></b></td><td><b>Gain.</b> ADC_CHANNEL_GAIN in xml file.</td></tr>
<tr><td><b><i>Data byte x+8 thru x+11</i></b></td><td><b>Offset.</b> ADC_CHANNEL_OFFSET in xml file.
Float.</td></tr>
</table>
<b>Bytes 5-16:</b> ADC Channel 0 (ADC_CHANNEL_ASSIGNMENT in xml) Info data set
<b>Bytes 17-28:</b> ADC Channel 1 Info data set
<b>Bytes 29-40:</b> ADC Channel 2 Info data set
<b>:</b>
<b>Bytes 173-184:</b> ADC Channel 14 Info data set
<b>Bytes 185-196:</b> ADC Channel 15 Info data set

<b>Byte 197:</b> GPIO Port 0 Mode In xml file unit-type= “gpio-global” id= GPIO_P0_MODE
<b>Byte 198:</b> GPIO Port 0 Reserved = 0x00
<b>Bytes 199-206:</b> GPIO Port 0 Pin[y] Assignment (enum) y=0-7 (8 pins)
<b>Byte 207:</b> GPIO Port 1 Mode In xml file unit-type= “gpio-global” id= GPIO_P1_MODE
</td></tr>
</table>

	Byte 208: GPIO Port 1 Reserved = 0x00 **Bytes 209-216:** GPIO Port 1 Pin[y] Assignment (enum) y=0-7 (8 pins)

2.4.5 Memory Configuration (Format = 0x05)

Send present memory for memory associated with this OCC. OCC will require this packet for observation state but HTMGT 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, HTMGT 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.

<table>
<tr><td><b><i>Data Length</i></b></td><td>Variable</td></tr>
<tr><td><b><i>Data</i></b></td><td><b>Byte 1:</b> Format = 0x05
<b>Byte 2:</b> Version = 0x10
<b>Byte 3:</b> Number of data sets to follow. NOTE: A 0x00 indicates that memory monitoring is to be disabled (if not already enabled). When disabled there is no communication to Centaur and the OCC will not require the Memory Throttling data packet.

<b><u>Version 0x10:</u></b>
The format of each set is:
<table>
<tr><td><b>Data bytes x thru x+3</b></td><td>Reserved. 0x00000000</td></tr>
<tr><td><b>Data bytes x+4 and x+5</b></td><td><b>Hardware Sensor ID.</b> Sensor ID to use for calling out DIMM or Centaur error</td></tr>
<tr><td><b>Data bytes x+6 and x+7</b></td><td><b>Temperature Sensor ID.</b> Sensor ID to use for reporting DIMM or Centaur temperature in poll response</td></tr>
<tr><td><b>Data byte x + 8</b></td><td><b>Centaur #.</b> Value 0-7 that indicates physical location of Centaur</td></tr>
<tr><td><b>Data byte x + 9</b></td><td><b>DIMM #.</b> Value 0-7 that indicates physical DIMM location behind Centaur. 0xFF indicates the Sensor ID is for the Centaur itself.</td></tr>
<tr><td><b>Data bytes x+10 and x+11</b></td><td>Reserved. 0x0000</td></tr>
</table>
</td></tr>
</table>

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

Data Length	0x0008
Data	**Byte 1:** Format = 0x07 **Version 0x10:** **Byte 2:** Version = 0x10 **Bytes 3 & 4: Minimum power cap**. In 1W units, the lowest power cap that a user may set and is guaranteed to be held via processor DVFS. **Bytes 5 & 6: System Maximum power cap**. In 1W units. 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 7 & 8: Oversubscription power cap.** In 1W units. If there is no oversubscription support this will be 0x0000 and the OCC will not be monitoring for 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. All data comes from xml file

2.4.7 System Configuration (Format = 0x0F)

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

<table>
<tr><td><b><i>Data Length</i></b></td><td>0x0039</td></tr>
<tr><td><b><i>Data</i></b></td><td><b>Byte 1:</b> Format = 0x0F
<b>Byte 2:</b> Version = 0x10
<b>Byte 3:</b> 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 = ‘1’</td><td></td><td></td><td></td><td></td><td></td><td></td><td>Single Node = ‘1’</td></tr>
</table>
<b>KVM</b> – HTMGT hard code to ‘1’ = System boot mode selection. 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 (not supported)

<b>Single Node</b> – Indicates if system is single or multi-node. HTMGT to hard code to ‘1’ for single node

‘0’ = Reserved for multi-node (not supported)

<b>Bytes 4-5:</b> Processor Sensor ID – Sensor ID for this OCC processor, used by OCC for processor error call out
<b>Bytes 6-7:</b> Core 0 Temperature Sensor ID – Sensor ID for physical core 0, used by OCC to report core 0 temperature
<b>Bytes 8-9:</b> Core 0 Frequency Sensor ID – Sensor ID for physical core 0, used by OCC to report core 0 frequency
<b>Bytes 10-11:</b> Core 1 Temperature Sensor ID – Sensor ID for physical core 1, used by OCC to report core 1 temperature
<b>Bytes 12-13:</b> Core 1 Frequency Sensor ID – Sensor ID for physical core 1, used by OCC to report core 1 frequency
<b>Bytes 14-15:</b> Core 2 Temperature Sensor ID – Sensor ID for physical core 2, used by OCC to report core 2 temperature
<b>Bytes 16-17:</b> Core 2 Frequency Sensor ID – Sensor ID for physical core 2, used by OCC to report core 2 frequency
<b>Bytes 18-19:</b> Core 3 Temperature Sensor ID – Sensor ID for physical core 3, used by OCC to report core 3 temperature
<b>Bytes 20-21:</b> Core 3 Frequency Sensor ID – Sensor ID for physical core 3, used by OCC to report core 3 frequency
<b>Bytes 22-23:</b> Core 4 Temperature Sensor ID – Sensor ID for physical core 4, used by OCC to report core 4 temperature</td></tr>
</table>

	Bytes 24-25: Core 4 Frequency Sensor ID – Sensor ID for physical core 4, used by OCC to report core 4 frequency **Bytes 26-27:** Core 5 Temperature Sensor ID – Sensor ID for physical core 5, used by OCC to report core 5 temperature **Bytes 28-29:** Core 5 Frequency Sensor ID – Sensor ID for physical core 5, used by OCC to report core 5 frequency **Bytes 30-31:** Core 6 Temperature Sensor ID – Sensor ID for physical core 6, used by OCC to report core 6 temperature **Bytes 32-33:** Core 6 Frequency Sensor ID – Sensor ID for physical core 6, used by OCC to report core 6 frequency **Bytes 34-35:** Core 7 Temperature Sensor ID – Sensor ID for physical core 7, used by OCC to report core 7 temperature **Bytes 36-37:** Core 7 Frequency Sensor ID – Sensor ID for physical core 7, used by OCC to report core 7 frequency **Bytes 38-39:** Core 8 Temperature Sensor ID – Sensor ID for physical core 8, used by OCC to report core 8 temperature **Bytes 40-41:** Core 8 Frequency Sensor ID – Sensor ID for physical core 8, used by OCC to report core 8 frequency **Bytes 42-43:** Core 9 Temperature Sensor ID – Sensor ID for physical core 9, used by OCC to report core 9 temperature **Bytes 44-45:** Core 9 Frequency Sensor ID – Sensor ID for physical core 9, used by OCC to report core 9 frequency **Bytes 46-47:** Core 10 Temperature Sensor ID – Sensor ID for physical core 10, used by OCC to report core 10 temperature **Bytes 48-49:** Core 10 Frequency Sensor ID – Sensor ID for physical core 10, used by OCC to report core 10 frequency **Bytes 50-51:** Core 11 Temperature Sensor ID – Sensor ID for physical core 11, used by OCC to report core 11 temperature **Bytes 52-53:** Core 11 Frequency Sensor ID – Sensor ID for physical core 11, used by OCC to report core 11 frequency **Bytes 54-55:** Backplane Sensor ID – Used by OCC for system backplane error call out **Bytes 56-57:** APSS Sensor ID – Used by OCC for APSS error call out

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 with and without power supply redundancy defined in the xml file. 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.
- HTMGT must ensure that 0's are never sent for any numerator values. If oversubscription is not supported HTMGT will send redundant power supply throttle values for them.
- If there is a failure to calculate throttle values the safe mode memory throttle defined in the xml file will be sent to the OCC.
- When OCC switches between redundant and oversubscription the OCC must write the appropriate N_PER_MBA and N_PER_CHIP.
- When throttling due to OT 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 the 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><b>Byte 1:</b> Format = 0x12
<b>Byte 2:</b> Version = 0x10
<b>Byte 3:</b> Number of memory throttling data sets to follow.

<b><u>Version 0x10:</u></b>
The format of each set is:
<table>
<tr><td><b>Data byte x</b></td><td><b>Centaur #.</b> Value 0-7 that indicates physical Centaur that the throttles are for</td></tr>
<tr><td><b>Data byte x+1</b></td><td><b>MBA #.</b> Value 0-1 that indicates MBA within the previous Centaur # that the throttles are for</td></tr>
<tr><td><b>Data byte x+2 &amp; x+3</b></td><td><b>Minimum OT N_PER_MBA.</b> Lowest per MBA numerator ever allowed when OCC is throttling due to OT. This is calculated based on the xml file minimum memory utilization for over temperature value.</td></tr>
<tr><td><b>Data byte x+4 &amp; x+5</b></td><td><b>Redundant Power N_PER_MBA.</b> Static per MBA numerator setting when not in oversubscription. Calculated based on xml file memory power with redundant power. HTMGT to guarantee that this is not lower than xml file minimum memory utilization for redundant power</td></tr>
</table>
</td></tr>
</table>

	Data byte x+6 & x+7	**Redundant Power N_PER_CHIP.** Static per chip numerator setting when not in oversubscription.
	Data byte x+8 & x+9	**Oversubscription N_PER_MBA.** Static per MBA numerator setting when in oversubscription. Calculated based on xml file oversubscription memory power. HTMGT to guarantee that this is not lower than xml file minimum memory utilization for oversubscription.
	Data byte x+10 & x+11	**Oversubscription N_PER_CHIP.** Static per chip numerator setting when in oversubscription.

2.4.9 Thermal Control Thresholds (Format = 0x13)

This command is used to send the temperature thresholds. The thresholds come from the xml file. All temperatures are in Celsius.

<table>
<tr><td>Data Length</td><td>Variable</td></tr>
<tr><td>Data</td><td><b>Byte 1:</b> Format = 0x13
<b>Byte 2:</b> Version= 0x10

<b><u>Version 0x10:</u></b>
<b>Byte 3:</b> Number of data sets that follows

Data set size is 4 bytes. Format of each data set is:
<table>
<tr><td><b>Data byte x</b></td><td><b>FRU type.</b> Indicates FRU type that thermal info is for
0x00: Processor
0x01: Centaur
0x02: DIMM</td></tr>
<tr><td><b>Data byte x+1</b></td><td><b>DVFS -</b> Temperature above which DVFS/throttling will be invoked. 0xFF indicates not defined.</td></tr>
<tr><td><b>Data byte x+2</b></td><td><b>ERROR -</b> Temperature to generate error and callout FRU over temperature. 0xFF indicates not defined.</td></tr>
<tr><td><b>Data byte x+3</b></td><td><b>MAX_READ_TIMEOUT –</b> Maximum time (in seconds) allowed without having new temperature readings. Throttling/dvfs will occur if this timeout is hit. 0xFF indicates not defined.</td></tr>
</table>
<b>Bytes 4-7:</b> Data set #1
<b>Bytes 8- 11:</b> Data set #2
<b>:</b></td></tr>
</table>

2.4.10 WOF Core Frequency Data (Format = 0x30)

This command is used to send the active core frequency data that is needed by the OCC to support WOF. This data packet is required by the OCC for active state and cannot change while OCC is active. If the OCC has received a zero ultra turbo frequency point in the frequency point data packet indicating WOF is not supported this data packet will still be sent. The frequency uplift tables are defined in the xml file, one table per chip sort. HB will determine the chip sort and pick the correct frequency uplift table to be used and write that one table to ATTR_WOF_FREQUENCY_UPLIFT_SELECTED for HTMGT to send to the OCC.

<table>
<tr><td><b><i>Data Length</i></b></td><td>0x0244 if WOF is supported else 0x0008 if WOF not supported</td></tr>
<tr><td><b><i>Data</i></b></td><td><b>Byte 1:</b> Format = 0x30
<b>Byte 2:</b> Version = 0x10
<b>Byte 3:</b> Max Cores Per Chip. Max number of cores that will ever be considered good for this OCC chip. De-configured cores will count towards this max. HTMGT determines this value by using the targeting model to count present cores. This value will be sent even if WOF is not supported.
<b>Bytes 4-6:</b> Reserved = 0x000000
<b>Byte 7:</b> Number of rows (r) of 2 byte values, including the column header row, in the following frequency uplift table. 0x16 if WOF is supported. 0x00 if WOF is not supported.
<b>Byte 8:</b> Number of columns (c) of 2 byte values, including the row header column, in the following frequency uplift table. 0x0D if WOF is supported. 0x00 if WOF is not supported.

<b>Bytes 9 thru 580:</b> Frequency Uplift Table. Gives percentage (in hundredths of a percent i.e. 100 = 1.00%) frequency may be uplifted from turbo based on current and active cores. Columns are # of cores active and the rows are AC reduction (Current efficiency) in hundredths of a percent i.e. 100 = 1.00%. The OCC is responsible for sorting the data in the order required by the OCC. Every table value, including headers, are 2 bytes, MSB is sent first for each value and the table is sent in order by row:
<table>
<tr><th><i>0x0000</i></th><th><i>Column 1: # cores active</i></th><th><i>Column 2: # cores active</i></th><th><i>..</i></th><th><i>Column c-1: # cores active</i></th></tr>
<tr><td><b><i>Row 1: Current eff in 1/100%</i></b></td><td>Freq uplift from turbo in 1/100%</td><td>Freq uplift from turbo in 1/100%</td><td>Freq uplift from turbo in 1/100%</td><td>Freq uplift from turbo in 1/100%</td></tr>
<tr><td><b><i>Row 2: Current eff in 1/100%</i></b></td><td>Freq uplift from turbo in 1/100%</td><td>..</td><td>..</td><td>Freq uplift from turbo in 1/100%</td></tr>
<tr><td><b><i>:</i></b></td><td>:</td><td>:</td><td>:</td><td>:</td></tr>
<tr><td><b><i>Row r-1: Current eff in 1/100%</i></b></td><td>Freq uplift from turbo in 1/100%</td><td>Freq uplift from turbo in 1/100%</td><td>Freq uplift from turbo in 1/100%</td><td>Freq uplift from turbo in 1/100%</td></tr>
</table></td></tr>
</table>

2.4.11 WOF VRM Efficiency Data (Format = 0x31)

This command is used to send the VRM efficiency data that is needed by the OCC to support WOF. This data packet is required by the OCC for active state and cannot change while OCC is active. If the OCC has received a zero ultra turbo frequency point in the frequency point data packet indicating WOF is not supported this data packet will still be sent. The VRM efficiency table is defined in the xml file WOF_REGULATOR_EFFICIENICIES, there is only one for the system.

<table>
<tr><td><b><i>Data Length</i></b></td><td>0x005C if WOF is supported else 0x0008 if WOF not supported</td></tr>
<tr><td><b><i>Data</i></b></td><td><b>Byte 1:</b> Format = 0x31
<b>Byte 2:</b> Version = 0x10
<b>Bytes 3-6:</b> Reserved = 0x00000000
<b>Byte 7:</b> Number of rows (r) of 2 byte values, including the column header row, in the following VRM Efficiency table. 0x03 if WOF is supported. 0x00 if WOF is not supported.
<b>Byte 8:</b> Number of columns (c) of 2 byte values, including the row header column, in the following VRM Efficiency table. 0x0E if WOF is supported. 0x00 if WOF is not supported.

<b>Bytes 9 thru 92:</b> VRM Efficiency Table. Gives VRM efficiency in hundredths of a percent (i.e. 100 = 1.00%) based on output voltage and output current. Columns are output current in AMPs and the rows are output voltage in hundredths of a volt (i.e. 100 = 1V). The OCC is responsible for sorting the data in the order required by the OCC. Every table value, including headers, are 2 bytes, MSB is sent first for each value and the table is sent in order by row:
<table>
<tr><th><i>0x0000</i></th><th><i>Column 1: Output Current in AMPs</i></th><th><i>Column 2: Output Current in AMPs</i></th><th>..</th><th><i>Column c-1: Output Current in AMPs</i></th></tr>
<tr><td><b><i>Row 1: Output Voltage in 1/100 V</i></b></td><td>Efficiency in 1/100%</td><td>Efficiency in 1/100%</td><td>Efficiency in 1/100%</td><td>Efficiency in 1/100%</td></tr>
<tr><td><b><i>:</i></b></td><td>:</td><td>:</td><td>:</td><td>:</td></tr>
<tr><td><b><i>Row r-1: Output Voltage in 1/100 V</i></b></td><td>Efficiency in 1/100%</td><td>Efficiency in 1/100%</td><td>Efficiency in 1/100%</td><td>Efficiency in 1/100%</td></tr>
</table>
</td></tr>
</table>

2.4.12 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.
HTMGT	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. 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.
HTMGT	HTMGT will send this before putting the OCC into reset

Reset Prep Command Packet:

Sequence Number	xx
Command Type	0x25
Data Length	0x0002
Data	**Byte 1:** Version = 0x00 **Byte 2:** Reason for reset (Regardless of 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 log is optional, if this OCC is master OCC it may want to generate FFDC log.
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 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
HTMGT	HTMGT will send this when requested by HBRT

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

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, HTMGT will read error log length bytes from the OCC poll response starting at the error log start address from the same OCC poll response.

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.

<u>Version = 0x01:</u>
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 – HTMGT 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 HTMGT processing HTMGT 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***
0:5	*Reserved*
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 known by GPE being frozen.
7 (msb)	**Reset Required**. Error is critical but may recover by resetting the OCC. NOTE: If reset retry count has been met the OCCs will remain in reset.

Bytes 8-11: 4 bytes reserved for future use
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 are 6 0x00's followed by the 2 byte Sensor ID) **0x02** – HTMGT-OCC Component ID (following 8 Callout bytes are 7 0x00's followed by the HTMGT-

	OCC ID defined in Appendix C)
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: User Data – OCC defined data that OCC wants appended to the error log. This may include things like OCC trace, OCC fw level, OCC role, OCC ID, OCC module ID.....

3.2 HTMGT-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. HTMGT must be able to communicate with all OCCs.

3.3 BMC-OCC Communication Failure

A communication failure is defined as one of the following:

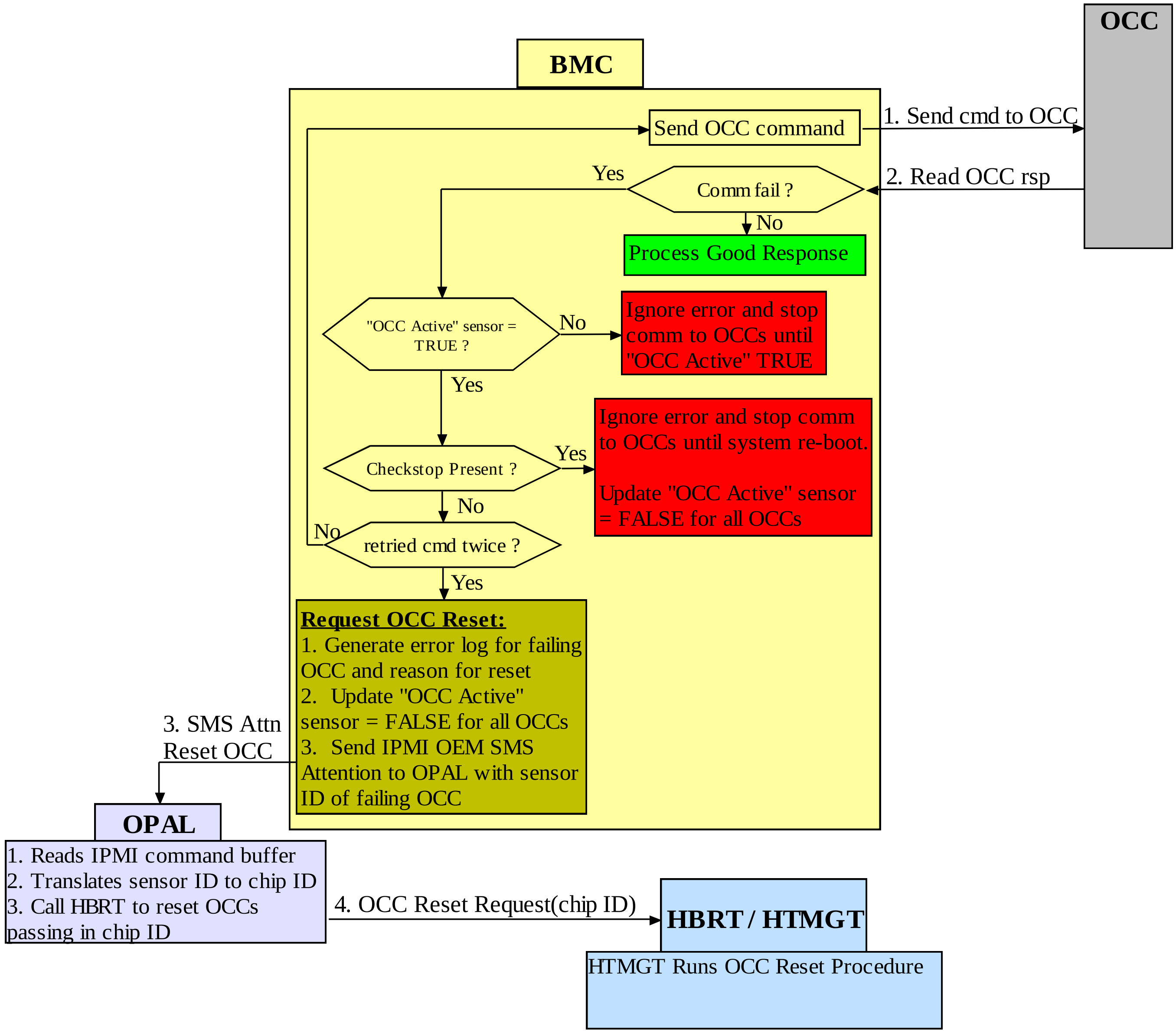
- Response Checksum failure
- Sequence number mismatch after command timeout has been reached.
- OCC Return Status still “In Progress” after command timeout has been reached. See “Command Summary Table” section for recommended timeout by command.
- Any non-successful Return Code
- I2C Failure sending command or reading response

When any of the above communication failures occur the BMC should follow the BMC-OCC Communication Failure Handling process defined in the next section.

3.3.1 BMC-OCC Communication Failure Handling

On failures communicating with an OCC 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.

3.3.1.1 BMC-OCC Communication Failure Handling Flow


OCC
BMC
Send OCC command
1. Send cmd to OCC
2. Read OCC rsp
Yes
Comm fail ?
No
Process Good Response
"OCC Active" sensor = TRUE ?
No
Ignore error and stop comm to OCCs until "OCC Active" TRUE
Yes
Checkstop Present ?
Yes
Ignore error and stop comm to OCCs until system re-boot.
Update "OCC Active" sensor = FALSE for all OCCs
No
No
retried cmd twice ?
Yes
Request OCC Reset:
1. Generate error log for failing OCC and reason for reset
2. Update "OCC Active" sensor = FALSE for all OCCs
3. Send IPMI OEM SMS Attention to OPAL with sensor ID of failing OCC
3. SMS Attn Reset OCC
OPAL
1. Reads IPMI command buffer
2. Translates sensor ID to chip ID
3. Call HBRT to reset OCCs passing in chip ID
4. OCC Reset Request(chip ID)
HBRT / HTMGT
HTMGT Runs OCC Reset Procedure

3.4 Errors Requiring OCC Reset

Any OCC requiring a reset will result in running the OCC Reset (safe mode) procedure which resets all OCCs. All OCCs will be held in reset (i.e. system in safe mode) after reaching 3 reset attempts due to the same OCC failing.

3.4.1 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.

3.4.1.1 BMC Request for OCC Reset

Any request for an OCC reset will be resetting all OCCs in the system. When BMC determines that it needs to request the OCCs to be 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.

3.4.2 HTMGT Detected Reasons for OCC Reset

HTMGT will reset the OCCs when it sees one of the following:

- HTMGT can't communicate with an OCC
- OCC fails to make requested OCC state change
- "OCCs Present" byte in poll response does not match HTMGT view of OCCs present
- OCC poll response not reporting correct OCC role that HTMGT set

3.4.3 OCC Detected Reasons for OCC Reset

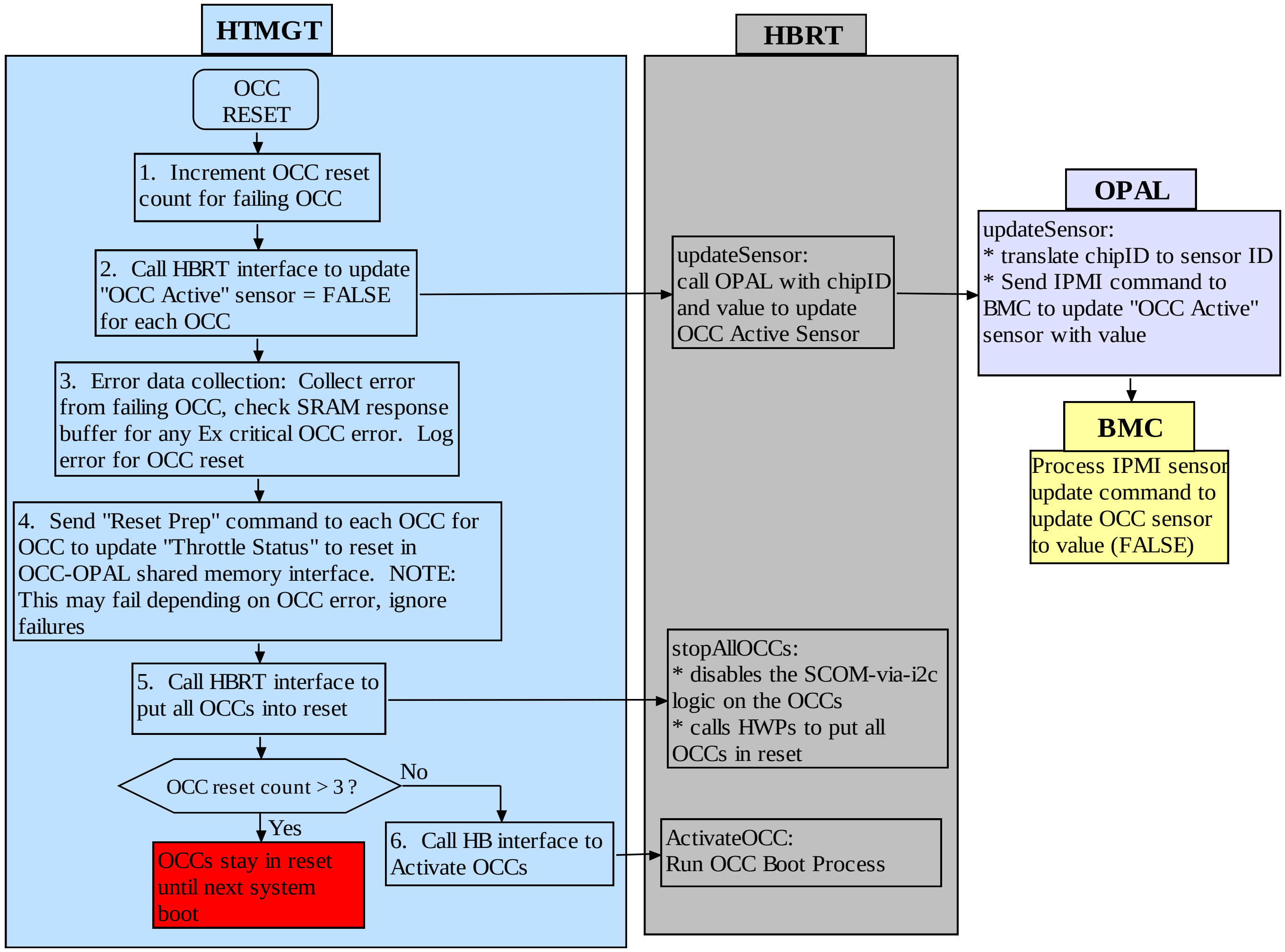
OCC will create an error log and request a reset for the following:

- Processor SCOM failure
- Failure to maintain power cap
- Timeout reading processor temperatures
- Failure from SSX operating system
- Failure within the OCC complex of the processor
- GPE halted due to checkstop
- PMC failure

HTMGT will process the reset request from the OCC as part of collecting the error log from the OCC and run the OCC Reset Procedure.

3.5 OCC Reset Procedure (Safe Mode)

This procedure will run when any OCC needs a reset. This will be resetting all OCCs. When bringing the OCCs active again the same process is followed as a system boot documented in the OCC Boot Process chapter.


HTMGT
OCC RESET
1. Increment OCC reset count for failing OCC
2. Call HBRT interface to update "OCC Active" sensor = FALSE for each OCC
3. Error data collection: Collect error from failing OCC, check SRAM response buffer for any Ex critical OCC error. Log error for OCC reset
4. Send "Reset Prep" command to each OCC for OCC to update "Throttle Status" to reset in OCC-OPAL shared memory interface. NOTE: This may fail depending on OCC error, ignore failures
5. Call HBRT interface to put all OCCs into reset
OCC reset count > 3 ?
No
Yes
OCCs stay in reset until next system boot
6. Call HB interface to Activate OCCs
HBRT
updateSensor: call OPAL with chipID and value to update OCC Active Sensor
stopAllOCCs: * disables the SCOM-via-i2c logic on the OCCs * calls HWPs to put all OCCs in reset
ActivateOCC: Run OCC Boot Process
OPAL
updateSensor: * translate chipID to sensor ID * Send IPMI command to BMC to update "OCC Active" sensor with value
BMC
Process IPMI sensor update command to update OCC sensor to value (FALSE)

3.6 Error Scenarios

3.6.1 OCC Fails to Load or Fails to go Active after Load

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.

3.6.2 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.

3.6.3 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.

3.6.4 Attention Line to Host is Broken

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

OCC Actions: Creates the error log and move to safe state, safe state will be reflected in the OCC poll response "Current OCC State" byte. The attention to HTMGT to collect the error and reset OCCs will not be processed due to broken attention line.

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

BMC Actions: Check "Current OCC State" byte in poll response for safe and send OCC reset request to HTMGT after defined time of being in safe state. Time defined in BMC Detected Reasons for OCC Reset section.

3.6.5 OCC Takes a Kernel Exception and goes to Halt

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

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

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

3.6.6 OCC-BMC Interface is Broken

OCC Actions: Nothing. OCC is unaware.

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

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

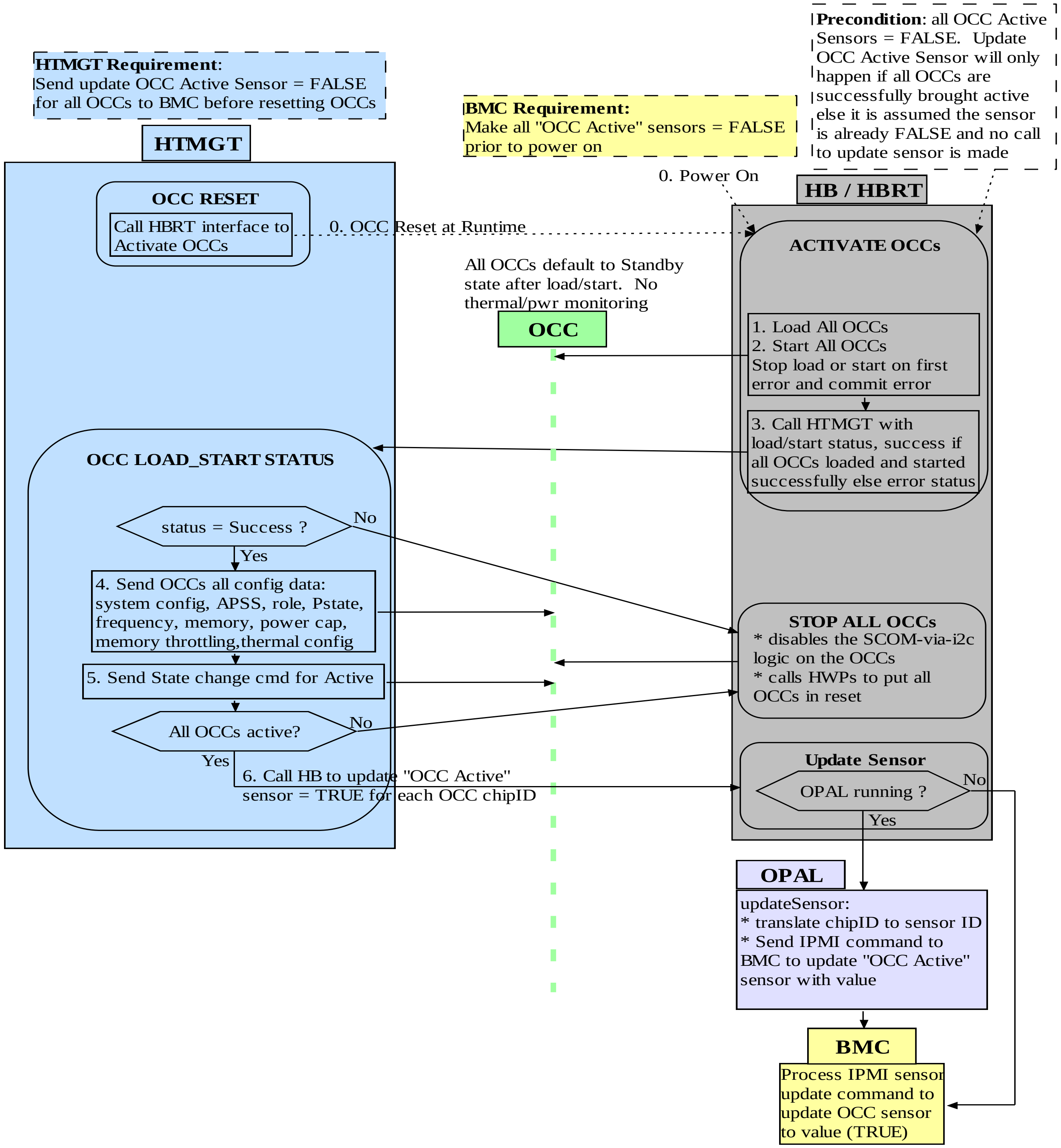
3.6.7 OCCs Held in Reset

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

4 OCC Boot Process

After the OCC is loaded and taken out of reset it will default to “standby” state and wait for configuration data from HTMGT and for HTMGT to send Set State command to Active. There is no thermal or power monitoring until the OCC is in active state. When OCC is told to go active it will populate OCC-OPAL shared memory interface with ‘valid’ and all Pstate data.

4.1 Activate OCCs Procedure


HTMGT Requirement: Send update OCC Active Sensor = FALSE for all OCCs to BMC before resetting OCCs
HTMGT
OCC RESET
Call HBRT interface to Activate OCCs
0. OCC Reset at Runtime
BMC Requirement: Make all "OCC Active" sensors = FALSE prior to power on
Precondition: all OCC Active Sensors = FALSE. Update OCC Active Sensor will only happen if all OCCs are successfully brought active else it is assumed the sensor is already FALSE and no call to update sensor is made
0. Power On
HB / HBRT
ACTIVATE OCCs
All OCCs default to Standby state after load/start. No thermal/pwr monitoring
OCC
1. Load All OCCs
2. Start All OCCs
Stop load or start on first error and commit error
3. Call HTMGT with load/start status, success if all OCCs loaded and started successfully else error status
OCC LOAD_START STATUS
status = Success ?
No
Yes
4. Send OCCs all config data: system config, APSS, role, Pstate, frequency, memory, power cap, memory throttling,thermal config
5. Send State change cmd for Active
All OCCs active?
No
Yes
6. Call HB to update "OCC Active" sensor = TRUE for each OCC chipID
STOP ALL OCCs
* disables the SCOM-via-i2c logic on the OCCs
* calls HWPs to put all OCCs in reset
Update Sensor
OPAL running ?
No
Yes
OPAL
updateSensor:
* translate chipID to sensor ID
* Send IPMI command to BMC to update "OCC Active" sensor with value
BMC
Process IPMI sensor update command to update OCC sensor to value (TRUE)

5 Frequency Points

- Frequency points are across whole system

5.1 VPD

These are frequency points defined in the module VPD #V keyword.

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. - Host Boot writes the max to ATTR_MIN_FREQ_MHZ attribute that HTMGT reads to know minimum frequency and sends to the OCC
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 HTMGT 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 HTMGT will read to get the legacy “turbo” chip max frequency.
Ultra Turbo (NOTE: this was previously called Fvmin in VPD doc which was never used)	- 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 HTMGT 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-OPAL Shared Memory Interface

Offset is from HOMER starting location 0x001F8000.

6.1 Version 0x01

Offset	*Byte 0*	*Byte 1*	*Byte 2*	*Byte 3*	*Byte 4*	*Byte 5*	*Byte 6*	*Byte 7*
0x00	Valid (0x01)	Version (0x01)	Throttle Status	Min Pstate	Nominal Pstate	Max Pstate	Reserved	
0x08	Reserved							
0x10	Pstate number	Flag	VDD (vid)	VCS (vid)	Freq in kHz			
0x18	Pstate number	Flag	VDD (vid)	VCS (vid)	Freq in kHz			
:	:	:	:	:	:			
	Up to 256 Pstates							

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

Version – Indicates format version = 0x01.

Throttle Status – Indicates reason that OCC may have limited Max Pstate.
0x00 = No throttle
0x01 = Power Cap
0x02 = Processor Over Temperature
0x03 = Power Supply Failure (currently not used)
0x04 = Overcurrent (currently not used)
0x05 = OCC reset. Some OCC failures will not allow for OCC to update throttle status.

Min Pstate – Minimum Pstate ever allowed.

Nominal Pstate – Pstate for nominal.

Max Pstate – Maximum Pstate ever allowed with no power, thermal or error condition present. NOTE: Current state of system may not allow for maximum Pstate to be reached, see Throttle Status. Max Pstate field is not updated when throttling occurs.

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

6.2 Version 0x02

New version to support P8+ ultra turbo.

Offset	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x0000	Valid (0x01)	Version (0x02)	Throttle Status	Min Pstate	Nominal Pstate	Turbo Pstate	Ultra Turbo Pstate	Reserve
0x0008	Reserved							
0x0010	Pstate number	Flag	VDD (vid)	VCS (vid)	Freq in kHz			
0x0018	Pstate number	Flag	VDD (vid)	VCS (vid)	Freq in kHz			
:	:	:	:	:	:			
0x0808	256 Pstates							
0x0810	Max Pstate with 1 active core	Max Pstate with 2 active cores	Max Pstate with 3 active cores	Max Pstate with 4 active cores	Max Pstate with 5 active cores	Max Pstate with 6 active cores	Max Pstate with 7 active cores	Max Pstate with 8 active cores
0x0818	Max Pstate with 9 active cores	Max Pstate with 10 active cores	Max Pstate with 11 active cores	Max Pstate with 12 active cores	Reserve = 0xFF	Reserve = 0xFF	Reserve = 0xFF	Reserve = 0xFF

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

Version – Indicates format version = 0x02.

Throttle Status – Indicates reason that OCC may have limited Max Pstate. NOTE: 0x06/0x07 are for debug purposes only and should not result in errors.

0x00 = No throttle
0x01 = Power Cap
0x02 = Processor Over Temperature
0x03 = Power Supply Failure (currently not used)
0x04 = Overcurrent (currently not used)
0x05 = OCC reset. Some OCC failures will not allow for OCC to update throttle status.
0x06 = Exceeded Power Cap above Turbo Pstate
0x07 = Processor Over Temperature above Turbo Pstate

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

Max Pstate With x Active Cores – Used for WOF. Linux will not request a Pstate above that allowed per core. NOTE: If WOF is not enabled the max Pstate for active cores will all be set to turbo Pstate. NOTE2: Parts with less than 12 cores will have 0xFF for extra cores.

7 Sensor Data Format Definitions

This chapter defines the formats for each sensor type that may be returned in the Status and Sensor Poll response.
NOTE: Sensor ID field is always 2 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 1 byte of 0x00 followed by the 1 byte IPMI sensor ID.

7.1 Temperature Sensors ("TEMP")

This is available in master and slave OCC poll responses.

Sensor Eye Catcher = "TEMP"
Sensor Version = 0x01
Sensor Length = 0x04

Format for one sensor and repeated for Number of Sensors:

Offset	
0x00	**Sensor ID –** 2 bytes. To identify what the temperature represents
0x02	**Current Reading –** 2 bytes. Current temperature reading in degrees C. 0xFFFF = Error reading the temperature sensor

7.2 Frequency Sensors ("FREQ")

This is available in master and slave OCC poll responses.

Sensor Eye Catcher = "FREQ"
Sensor Version = 0x01
Sensor Length = 0x04

Format for one sensor and repeated for Number of Sensors:

Offset	
0x00	**Sensor ID –** 2 bytes. To identify what the frequency represents
0x02	**Current Reading –** 2 bytes. Current frequency in MHz

7.3 Power Sensors ("POWR")

This is only available from the master OCC poll response. All power values are output power, if input power is required the output power must be converted to input by using the power supply efficiency factor from the Configuration file.

Sensor Eye Catcher = "POWR"
Sensor Version = 0x01
Sensor Length = 0x0C

Format for one sensor and repeated for Number of Sensors:

Offset	
0x00	**Sensor ID –** 2 bytes. To identify what the power represents
0x02	**Update Tag –** 4 bytes. Count of number of 250us samples represented in Accumulator. Used for time derived sensor.
0x06	**Accumulator –** 4 bytes. Accumulation of 250us power readings
0x0A	**Current Reading –** 2 bytes. Most recent 250us reading in watts

7.4 Power Caps (“CAPS”)

This is only available from the master OCC poll response.

Sensor Eye Catcher = “CAPS”
Sensor Version = 0x01
Sensor Length = 0x0C

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	**Minimum Power Cap** – 2 bytes. In 1W units the minimum (output) power cap that may be set and held by the OCC.
0x0A	**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.

8 Power Management

8.1 XML Settings

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
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.
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_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 over-temp 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_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
WOF_ENABLED	Indicates if WOF is supported. 1 = WOF is supported, max frequency will be ultra turbo 0 = WOF is not supported, max frequency will be turbo (if turbo is supported) or nominal (turbo not supported)
WOF_PROC_SORT	Processor sort with frequency and WOF_FREQUENCY_UPLIFT table number to use
WOF_FREQUENCY_UPLIFT	Multiple tables, one for each processor sort to give percentage (in hundredths of a percent) frequency may be uplifted from turbo based on current and active cores. Columns are # of cores active and the rows are AC reduction (Current efficiency in hundredths of a percent)
WOF_REGULATOR_EFFICIENCIES	1 table to give VRM efficiency in hundredths of a percent based on output voltage and output current. Columns are output current in AMPs and the rows are output voltage in hundredths of a volt

8.2 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.

8.2.1 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.

8.2.2 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.

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

All manufacturing voltage and frequency testing will go thru Linux. The pState table can be biased for testing but this will require an OCC reset to load the new pState table.

10.2 Enable/Disable OCC Control

For manufacturing testing to prevent the OCC from 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 communication is allowed to send the poll command to read the sensors from OCC. When finished with testing the OCC must be put back into Active state (full actuation). Setting of OCC state is done via a command to Linux->HTMGT->OCC.

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 **0xE2 → Watchdog Timeout.** Halt due to OCC watchdog expiring.

Return Code	***Description***
	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
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 logging purposes ▪ HTMGT will never tell OCC to move to this state

Appendix C. HTMGT-OCC Component Ids

Following table is a list of component IDs internal to HTMGT-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 D. Other BMC Requirements

D.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	Checksum MSB	Checksum 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. 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>
```