



CXL ENGINEERING CHANGE NOTICE

TITLE:	CXL Hotness Monitoring Unit
DATE:	Introduced: 10/17/2023 Updated: 08/20/2024
AFFECTED DOCUMENT:	CXL 3.1
SPONSOR:	Micron, Microsoft, Intel

Part I

1. Summary of Functional Changes

This ECN defines an interface to manage hotness trackers on a CXL Type 3 Device. This initial version introduces the ability to discover capabilities, perform configurations, return outputs.

The above functionalities are manageable via a register interface enumerated through Register Locator DVSEC. A new Register Block type is introduced. If present in the Register Locator DVSEC, the feature is supported by the device.

Multiple CXL Hotness Monitoring Unit (CHMU) register interfaces can be supported by the device to control and manage different hotness trackers.

The device hotness trackers implementation is abstracted by the CHMU interface. The retrievable output is a list of addresses whose access counters values achieved a configurable threshold.

2. Benefits as a Result of the Changes

This interface enables software running on a CXL host to control and use hotness trackers implemented on CXL Type 3 devices.

Hotness trackers will be enabled to track accesses at DPA level related to various M2S Request types and can be utilized for different usage scenarios (memory workload analysis, memory tiering architectures, etc.).

3. Assessment of the Impact

The new feature introduced will be optional so there is no impact to backwards compatibility.

4. Analysis of the Hardware Implications

Implementation of CHMU on the CXL Device might entail hardware changes to support the logic to count accesses and resourced for the counters.

5. Analysis of the Software Implications

Host software can benefit from a standardized definition of the interface to manage hotness trackers.

6. Analysis of the Compliance and Test Implications

To be updated by CWG.

Update section 8.1.9.1 Register Offset Low (Offset: Varies) as follows

This register reports the BAR Indicator Register (BIR), the Register Block Identifier, and the lower address bits of the BAR offset associated with the Register Block.

Bits	Attributes	Description
2:0	HwInit	Register BIR: Indicates which one of a Function's BARs, located beginning at Offset 10h in Configuration Space, or entry in the Enhanced Allocation capability with a matching BAR Equivalent Indicator (BEI), is used to map the CXL registers into Memory Space. Defined encodings are: • 000b = Base Address Register 10h • 001b = Base Address Register 14h • 010b = Base Address Register 18h • 011b = Base Address Register 1Ch • 100b = Base Address Register 20h • 101b = Base Address Register 24h • All other encodings are reserved The Register block must be contained within the specified BAR. The specified BAR must be associated with the Function that implements the Register Locator DVSEC. For a 64-bit BAR, the Register BIR indicates the lower DWORD.
7:3	RsvdP	Reserved
15:8	HwInit	Register Block Identifier: Identifies the type of CXL registers. Defined encodings are: • 00h = Indicates the register block entry is empty and the Register BIR, Register Block Offset Low, and Register Block Offset High fields are invalid. • 01h = Component Registers. The format of the Component Register block is defined in Section 8.2.3. • 02h = BAR Virtualization ACL Registers. The format of the BAR Virtualization ACL Register Block is defined in Section 8.2.6. • 03h = CXL Device Registers. The format of the CXL Device Register block is defined in Section 8.2.9. • 04h = CPMU Registers. More than one instance per Register Locator DVSEC instance is permitted. The CPMU Register format is defined in Section 8.2.7. • 05h = CHMU Registers. More than one instance per Register Locator DVSEC instance is permitted. The CHMU Register format is defined in Section 8.2.8. • FFh = Designated Vendor Specific Registers. The format of the designated vendor specific register block starts with the header defined in Table 8-10. • All other encodings are reserved.
31:16	HwInit	Register Block Offset Low: A[31:16] byte offset from the starting address of the Function's BAR associated with the Register BIR field to point to the base of the Register Block. Register Block Offset is 64-KB aligned. Hence A[15:0] is 0000h.

8.2.8**CHMU Register Interface**

The CXL Hotness Monitoring Unit (CHMU) is an interface that allows software running on CXL hosts to identify the 'hot' memory ranges (i.e., memory ranges with higher access frequency relative to other memory ranges) in CXL memory devices in terms of memory access counts. The CHMU is not expected to be used for cold region tracking and shall not impact device performance.

The CHMU applies to HDM-H devices, HDM-DB devices, MLDs, and Multi-Headed devices. For MH-SLD devices, the CHMU is on a per-head basis whereas for LD-FAM devices, CHMU must be implemented on a per LD-ID basis. A CXL device may implement multiple CHMUs.

The CHMU interface will be used to manage the hardware units that are responsible for counting accesses, including discovering capabilities, setting configuration parameters, checking status, and collecting results.

To configure the hotness threshold, the CHMU management software shall first configure the unit size such that the device could arrange the internal counting structure resources to manage the configured unit size. Then, the device returns the actual counter width in the CHMU Status register(s). Finally, the hotness threshold can be configured depending on the counter width returned by the device.

CHMU counts CXL.mem M2S requests only and does not count UIO Direct Peer-to-peer accesses through CXL.io. CHMU does not count CXL.cache accesses.

CHMU counts accesses on specific DPA granularities called units; unit sizes may range from 256B to 2 GB. The unit size is set by the host software. The CHMU identifies DPA units that have an access frequency that is higher than a configurable threshold. These DPA units are called hot units.

The CHMUs contain a set of counters to count accesses to DPA units. The counter set is implementation specific and counters in the set are not directly accessible via the CHMU interface.

Access counting may be enabled on multiple address ranges with 256-MB granularity. Host software is responsible for setting properly the number of ranges considering the size of the units (e.g., if the unit size is 1 GB, a single 256-MB range should not be configured).

The device counts accesses to DPA units for programmed time intervals called epochs. The device reports the supported minimum and maximum epoch intervals that provide a high level of fidelity for the hotness tracking. Software is expected to program an epoch interval between the minimum and the maximum values. During an epoch, each accessed DPA unit is mapped to a counter within the set according to an implementation-specific mapping function. Examples of mapping function could be direct-mapped, set-associative, hash-function, etc. If the counter is free, it is allocated to the DPA unit and initialized to 1. If the counter is already allocated to the same DPA unit, the counter is incremented. If the counter is allocated to another DPA unit, the access might be not counted. All counters are freed at the end of their epochs.

If the counter related to a specific unit achieves the programmed hotness threshold during an epoch, the unit is considered hot, and is then reported to software. To report hot units to software, CHMU supports a circular structure called the CHMU Hotlist. The Hotlist is accessible through the CHMU's MMIO address space. There is a single Hotlist per CHMU instance for all the address ranges that can be configured on that CHMU. Each CHMU Hotlist entry is a 64-bit structure that contains a DPA unit address, called Unit ID,

and its counter value. After the unit size is configured by the software, the remaining bits in the 64-bit entry can be allocated for the counters. Therefore, the hotness threshold can be configured only after setting the unit size. The size of the counters is retrieved through the CHMU Status register.

The CHMU reports new hot units at the Hotlist's Tail pointer and increments the Tail. The software is expected to read the Hotlist entries at the Head pointer and increment the Head. When the Head and the Tail are equal, the Hotlist is considered empty. When the Tail is logically 1 behind the Head, the Hotlist is considered full. The Hotlist can be cleared by setting the Head register to be equal to the Tail register.

Two modes are defined for reporting hot units into the CHMU Hotlist: Epoch-based reporting mode and Always-on reporting mode. Software is responsible for enabling one of the two modes.

When Epoch-based reporting mode is enabled, hot units are reported in the CHMU Hotlist at the end of their counter's current epoch. Even if the counter reaches the hotness threshold before the epoch ends, the counter continues counting until the epoch ends. Consequently, the counter values in the CHMU Hotlist entries will contain the number of hot unit accesses within the epoch. The counters within the counting structure will be freed at the end of their respective epochs.

When Always-on reporting mode is enabled, after the counters achieve the hotness threshold, their corresponding Unit ID is immediately reported in the CHMU Hotlist and becomes visible to the software through the CHMU Hotlist-related registers. The hot unit-related counters are freed when the counters are reported in the Hotlist. The counters within the counting structure will be freed at the end of their respective epochs.

CHMU supports a down-sampling factor for the incoming M2S requests that allows sampling at a configurable rate, or at a rate that is controlled by the device. The down-sampling factor can be either selectable by software (if supported by the device) or can be selected by the device.

The CHMU interface supports MSI/MSI-X interrupts to alert when the:

- CHMU Hotlist overflows
- CHMU Hotlist crosses a level of fullness

If the CHMU Hotlist overflows, the new hot units may be discarded or retained as 'outstanding' hot units that will be reported when space is available in the CHMU Hotlist. The ability to report these outstanding hot units is indicated in the device capabilities. If the CHMU is disabled, any outstanding hot units will be dropped and never added to the CHMU Hotlist.

Each CHMU implements a set of CHMU-scoped registers within a register block. Multiple CHMU register block identifiers are permitted. In this version of the specification, within a CHMU register block, the device can support up to eight CHMU instances, and a single instance is composed of CHMU Capability, CHMU Configuration, CHMU Status, CHMU Hotlist Head, CHMU Hotlist Tail, CHMU Range Configuration Bitmap, and CHMU Hotlist Entry registers. The CHMU Configuration register may be configured only when the CHMU is disabled. When the CHMU is enabled, all the Configuration registers are locked (i.e., Read Only) except the Control field in the CHMU Configuration register. The CHMU Configuration register remains Read Only until the CHMU is disabled again.

Table 8-AA. CHMU Register Layout (Version=1)

Byte Offset	Length in Bytes	Register Name
00h	16	CHMU Common Capabilities
10h	64	CHMU Capability [0]
50h	32	CHMU Configuration [0]
70h	8	CHMU Status [0]
78h	2	CHMU Hotlist Head [0]
7Ah	2	CHMU Hotlist Tail [0]
7Ch	4	Reserved
10h+R ₀ ¹ h	H ₀ ² - R ₀	CHMU Range Configuration Bitmap [0]
10h+H ₀	M ₀ ³ *8	CHMU Hotlist [0]
10h+L ⁴	64	CHMU Capability [1]
50h+L	32	CHMU Configuration [1]
70h+L	8	CHMU Status [1]
78h+L	2	CHMU Hotlist Head [1]
7Ah+L	2	CHMU Hotlist Tail [1]
7Ch+L	4	Reserved
10h+L+R ₁	H ₁ - R ₁	CHMU Range Configuration Bitmap [1]
10h+L+H ₁	M ₁ *8	CHMU Hotlist [1]
...	...	...
10h+L*7d	64	CHMU Capability [7]
50h+L*7d	32	CHMU Configuration [7]
70h+L*7d	8	CHMU Status [7]
78h+L*7d	2	CHMU Hotlist Head [7]
7Ah+L*7d	2	CHMU Hotlist Tail [7]
7Ch+L*7d	4	Reserved
10h+L*7d+R ₇	H ₇ -R ₇	CHMU Range Configuration Bitmap [7]
10h+L*7d+H ₇	M ₇ *8	CHMU Hotlist [7]

¹Value R_i is reported in the CHMU Range Configuration Bitmap register offset of the CHMU Capability register.
²Value H_i is reported in the CHMU Hotlist Register Offset of the CHMU Capability register.
³Value M_i depends on the size of the hotlist in the CHMU instance.
⁴Value L corresponds to the value of the CHMU Instance Length field in the CHMU Common Capabilities register.

8.2.8.1 CHMU Common Capability Register (Offset 00h)

The CHMU Common Capability register must be accessed as an 8-byte quantity.

Table 8-XX. CHMU Common Capability Register (Offset 00h)

Bit	Attributes	Description
3:0	HwInit	Version: Set to 1. The version is incremented whenever the CHMU register structure is extended to add more functionality. Backward compatibility shall be maintained during this process. For all values of n, version n+1 may extend version n by replacing fields that are marked as reserved in version n or appending new registers but must not redefine the meaning of existing fields. Software that was written for a lower version may continue to operate on CHMU registers with a higher version but will not be able to take advantage of new functionality. Each field in the CHMU register structure is assumed to be introduced in version 1 of that structure unless specified otherwise in the field's definition within this specification.

Bit	Attributes	Description
7:4	RsvdP	Reserved
15:8	HwInit	Number of Supported CHMU Instances: Number of CHMU instances supported within the CHMU register block. In this version of the specification, the value of this field shall be less than or equal to 8.
63:16	RsvdP	Reserved
79:64	HwInit	CHMU Instance Length: Length in bytes of a single CHMU instance.
127:80	RsvdP	Reserved

8.2.8.2 CHMU Capability Register (Offset 10h + CHMU Instance Length * i¹)

The CHMU Capability register must be accessed as an 8-byte quantity.

Table 8-XX. CHMU Capability Register

Bit	Attributes	Description
3:0	HwInit	Interrupt Message Number: If Interrupt on Hotlist Overflow Support=1, this field indicates which MSI/MSI-X vector is used for the interrupt message that is generated in association with this CHMU instance. For MSI, the value in this field indicates the offset between the base Message Data and the interrupt message that is generated. Hardware is required to update this field so that it is correct if the number of MSI Messages assigned to the Function changes when software writes to the Multiple Message Enable field in the Message Control register for MSI. For MSI-X, the value in this field indicates which MSI-X Table entry is used to generate the interrupt message. The entry shall be one of the first 16 entries, regardless of whether the Function implements more than 16 entries. The value in this field shall be within the range configured by system software to the device. For a given MSI-X implementation, the entry shall remain constant. If both MSI and MSI-X are implemented, they are permitted to use different vectors, though software is permitted to enable only one mechanism at a time. If MSI-X is enabled, the value in this field shall indicate the vector for MSI-X. If MSI is enabled or neither is enabled, the value in this field indicates the vector for MSI. If software enables both MSI and MSI-X at the same time, the value in this field is undefined. It is recommended that the component allocate a distinct Interrupt Message Number to each CHMU instance.
4	HwInit	Interrupt on Hotlist Overflow Support: 0 = The CHMU does not support generation of interrupts upon hotlist overflow. 1 = The CHMU supports generation of interrupts upon hotlist overflow. Interrupt generation is controlled by the Interrupt on Hotlist Overflow bit in the CHMU Configuration register. The Interrupt Message Number is reported in the Interrupt Message Number field of this register.
5	HwInit	Interrupt on Hotlist Levels Crossing Support: 0 = CHMU does not support generation of interrupts upon hotlist levels crossing. 1 = CHMU supports generation of interrupts upon hotlist levels crossing. Interrupt generation is controlled by the Interrupt on Hotlist Levels Crossing bit in the CHMU Configuration register. The Interrupt Message Number is reported in the Interrupt Message Number field of this register.
7:6	HwInit	Epoch Type: 00b = Global. All counters start and end their epoch simultaneously. 01b = Per counter. Each counter starts and ends their epoch independently. The counter starts when it is allocated. - All other encodings are reserved.

¹ i is the index of the CHMU instance. In this version of the specification, the device can support up to 8 CHMU instances.

Bit	Attributes	Description
15:8	HwInit	Tracked M2S Requests: CXL.mem requests that can be tracked by the device. The following values are defined: • Bit[8]: Non-TEE read only (MemRd, MemRdData, MemSpecRd, MemInv) • Bit[9]: Non-TEE write only (MemWr, MemWrPtl) • Bit[10]: Non-TEE read and non-TEE write (MemRd, MemRdData, MemSpecRd, MemInv, MemWr, MemWrPtl) • Bit[11]: Non-TEE and TEE read (MemRd, MemRdData, MemSpecRd, MemInv, MemRdTEE, MemRdDataTEE, MemSpecRdTEE) • Bit[12]: Non-TEE and TEE write (MemWr, MemWrPtl, MemWrTEE, MemWrPtlTEE) • Bit[13]: Non-TEE read, TEE read, non-TEE write, TEE write • Bits[15:14]: Reserved
31:16	HwInit	Max Epoch Length: Maximum epoch length that the device supports. • Bits[19:16]: These bits specify the time scale. – 1h = 100 us – 2h = 1 ms – 3h = 10 ms – 4h = 100 ms – 5h = 1 s – All other encodings are reserved • Bits[31:20]: These bits specify the maximum epoch length, using the time scale indicated in bits[19:16]. Max value is 4095. A value of 0 is not permitted.
47:32	HwInit	Min Epoch Length: Minimum epoch length that the device supports. • Bits[35:32]: These bits specify the time scale. – 1h = 100 us – 2h = 1 ms – 3h = 10 ms – 4h = 100 ms – 5h = 1 s – All other encodings are reserved • Bits[47:36]: These bits specify the minimum epoch length, using the time scale indicated in bits[35:32]. Max value is 4095. A value of 0 is not permitted.
63:48	HwInit	Hotlist Size: Maximum number of hotlist entries. The minimum value to be supported is 64. Max value is 65,535.
95:64	HwInit	Supported Unit Sizes: Each bit corresponds to the granularity of memory for which accesses are counted separately. Supported values are expressed as a power of 2 and range from 256B to 2 GB. It is recommended to support at least one unit size of 2 MB or smaller granularity. The parameter is represented by a bitmap in which the unit size doubles with each bit increment. • Bit[64]: 256 B • Bit[65]: 512 B • Bit[66]: 1 KB • Bit[67]: 2 KB • Bit[68]: 4 KB • ... • Bit[86]: 1 GB • Bit[87]: 2 GB • Bits[95:88]: Reserved
111:96	HwInit	Supported Down-sampling Factor: The device can be configured to track one M2S request over one of the values encoded in this field. • Bit[96]: 1 (the device can sustain the full request rate) • Bit[97]: 2 • Bit[98]: 4 • Bit[99]: 8 • Bit[100]: 16 • Bit[101]: 32 • ... • Bit[111]: 32,768

Bit	Attributes	Description
127:112	HwInit	Capability Flags: Features supported for hotness tracking. • Bit[112]: Epoch-based reporting mode: if set, the device updates the list of hot units in the hotlist at the end of an epoch. • Bit[113]: Always-on Reporting Mode: When set to 1, the device pushes entries into the hotlist as soon as the related unit achieves or exceeds the hotness threshold. • Bit[114]: Randomized Down-sampling: When set to 1, the device supports implementation-specific randomization for selecting accesses to perform down-sampling. • Bit[115]: Address Overlap: When set to 1, the CHMU can overlap address ranges that are covered by other CHMU instances within the same register block. • Bit[116]: Postponed Outstanding Hot Units Insertion in the CHMU Hotlist on Overflow: When set to 1, if the CHMU hotlist overflows, outstanding hot units identified when the hotlist is in Overflow state become visible when one or more CHMU Hotlist entries are freed. The hot units detected in Overflow state will be visible within 10 ms if the number of entries that are freed is sufficient. • Bits[127:117]: Reserved
191:128	HwInit	CHMU Range Configuration Bitmap Register Offset: Offset from the first byte of the CHMU Capability register in which the CHMU Range Configuration Bitmap starts.
255:192	HwInit	CHMU Hotlist Register Offset. Offset from the first byte of the CHMU Capability register in which the CHMU Hotlist starts.
511:256	RsvdP	Reserved

8.2.8.3 CHMU Configuration [i] (Offset 50h + CHMU Instance Length * i)

The CHMU Configuration register must be accessed as an 8-byte quantity.

Table 8-YY. CHMU Configuration Register

Bit	Attributes	Description
7:0	RW	M2S Requests to Track: CXL.mem requests to be tracked by the device. The following values are defined: • 01h = Non-TEE read only (MemRd, MemRdData, MemSpecRd, MemInv) • 02h = Non-TEE write only (MemWr, MemWrPtl) • 03h = Non-TEE read and non-TEE write (MemRd, MemRdData, MemSpecRd, MemInv, MemWr, MemWrPtl) • 04h = Non-TEE and TEE read (MemRd, MemRdData, MemSpecRd, MemInv, MemRdTEE, MemRdDataTEE, MemSpecRdTEE) • 05h = Non-TEE and TEE write (MemWr, MemWrPtl, MemWrTEE, MemWrPtlTEE) • 06h = Non-TEE read, TEE read, non-TEE write, TEE write • All other encodings are reserved

Bit	Attributes	Description
15:8	RW	Flags: Features supported for hotness tracking. • Bit[8]: Randomized Down-sampling: When set to 1, the device uses implementation-specific randomization for selecting accesses to perform down-sampling. • Bit[9]: Interrupt on Hotlist Overflow: When set to 1, the device generates an interrupt when the hotlist overflows. The Interrupt Message Number is reported in the Interrupt Message Number field in the CHMU Capability registers. This bit can be set to 1 if the Interrupt on Hotlist Overflow Support bit in the CHMU Capability register is set to 1; otherwise, the bit is permitted to be hardwired to 0. • Bit[10]: Interrupt on Hotlist Levels Crossing: When set to 1, the device generates an interrupt when the hotlist achieves a number of host-configured elements through the Hotness Notification Threshold parameter. The Interrupt Message Number is reported in the Interrupt Message Number field in the CHMU Capability registers. This bit can be set to 1 if the Interrupt on Hotlist Levels Crossing Support bit in the CHMU Capability register is set; otherwise, the bit is permitted to be hardwired to 0. • Bits[15:11]: Reserved.
31:16	RW	Control: This field is used to require performing operations on the Hotness Monitoring Unit • Bit[16]: Enable Hotness Monitoring Unit: This bit is set to 1 to enable the Hotness Monitoring Unit, and is cleared to 0 to disable the CHMU. CHMU enablement and/or disablement completion status can be retrieved from the CHMU Status register. • Bit[17]: Reset Counters: This bit is set to 1 to clear the counters in the Hotness Monitoring Unit. The completion of reset counter operation can be established by polling the Operation in Progress field in the CHMU Status register. When the device completes clearing of the counters, this bit returns to a value of 0. The configuration is not affected. • Bits[31:18]: Reserved.
63:32	RW	Hotness Threshold: Host-configured hotness threshold value. This value cannot exceed the maximum counter width value returned in the Counter Width field in the CHMU Status registers.
95:64	RW	Unit Size: Host-configured counting granularity. Configurable values are expressed as a power of 2 and range from 256B to 2 GB. The unit sizes that the device supports are in the Supported Unit Sizes field in the CHMU Capability register.
103:96	RW	Down-sampling Factor: If down-sampling is enabled, the device tracks one M2S request over the value encoded in this field. • Bits[99:96]: Down-sampling factor: One of the 16 possible values reported in the Supported Down-sampling Factor field in the CHMU Capability register. • Bits[103:100]: Reserved.
111:104	RW	Reporting Mode: This field is used to enable one of the reporting modes that the device supports. • 00h = Enable Epoch-based reporting mode. The device ignores this field if the host programs a value that corresponds to an unsupported capability. • 01h = Enable Always-on Reporting Mode. The device ignores this field if the host programs a value that corresponds to an unsupported capability. • All other encodings are reserved.
127:112	RW	Epoch Length: Host-configured epoch length. • Bits[115:112]: These bits specify the time scale. - 1h = 100 us - 2h = 1 ms - 3h = 10 ms - 4h = 100 ms - 5h = 1 s - All other encodings are reserved. • Bits[127:116]: These bits specify the epoch length, using the time scale indicated in bits[115:112].

Bit	Attributes	Description
143:128	RW	Hotlist Notification Threshold: The device generates an interrupt when the number of entries in the Hotlist is greater than or equal to the value in this field and bit[1] of the Overflow Interrupt Status field, "Hotlist Level Crossed", in the CHMU Status register is equal to 0. Upon this condition, bit[1] of the Overflow Interrupt Status field in the CHMU Status register is set to 1. The value in this field cannot exceed the Hotlist size programmed in the Hotlist Size field in the CHMU Capability register. If the value of this field exceeds the programmed Hotlist size, this field is ignored, and no interrupts are generated. When bit[10] of the Flags field, "Interrupt on Hotlist Levels Crossing", is cleared to 0, this field is ignored.
255:144	RsvdP	Reserved

8.2.8.4 CHMU Status [i] (Offset 70h + CHMU Instance Length * i)

The CHMU Status register must be accessed as an 8-byte quantity.

Table 8-WW. CHMU Status Register

Bit	Attributes	Description
15:0	RO	Status: This field is used to retrieve CHMU-related status and error information. - Bit[0]: Tracking enabled: When set to 1, the CHMU is enabled. When cleared to 0, the CHMU is disabled. - Bits[15:1]: Reserved.
31:16	RO	Operation In progress: This status information returns the operation currently in progress. 0000h = No operation in progress. Indicates that no operations that can be started through the Control field in the CHMU Configuration register are in progress. 0001h = Enablement in progress. Indicates that enabling of the CHMU is not yet complete. 0002h = Disablement in progress. Indicates that disabling of the CHMU is not yet complete. 0003h = Reset counters in progress. Indicates that resetting of the CHMU counters is not yet complete. - All other encodings are reserved.
39:32	RO	Counter Width: Current number of bits in each counter used for counting the accesses. This field also indicates the counter width in each Hotlist entry.
47:40	RW1C	Overflow Interrupt Status: - Bit[40]: Hotlist Overflow. When set to 1, the Hotlist has encountered an overflow condition. Additional Hotlist Overflow interrupts are not generated until software clears this bit. - Bit[41]: Hotlist Level Crossed. When set to 1, the Hotlist has achieved the Hotlist Notification Threshold set by the software. Additional Hotlist Level Crossed interrupts are not generated until software clears this bit to 0. - Bits[47:42]: Reserved.
63:48	RsvdP	Reserved

8.2.8.5 CHMU Hotlist Head Register (Offset 78h + CHMU Instance Length * i)

The CHMU Hotlist Head register must be accessed as a 2-byte quantity.

Table 8-VV. CHMU Hotlist Head Register

Bit	Attributes	Description
15:0	RW	Head: Index that points to the oldest hot Unit ID in the hotlist that has not yet been read by software. Software reads hot unit entries at this index. This field can be written, by way the host, to the first unread entry. The Hotlist is considered empty when Head == Tail. The Hotlist is considered full when Head == (Tail + 1) mod (Hotlist Size). In this condition, the number of units in the CHMU Hotlist is equal to the Hotlist Size – 1.

8.2.8.6 CHMU Hotlist Tail Register (Offset 7Ah + CHMU Instance Length * i)

The CHMU Hotlist Tail register must be accessed as a 2-byte quantity. The value of this register must not be changed while the CHMU is enabled. If the value is changed while the CHMU is enabled, the device behavior is undefined.

Table 8-ZZ. CHMU Hotlist Tail Register

Bit	Attributes	Description
15:0	RW	Tail: Index that points to the Hotlist entry in which the device will report the next hot unit. The device increments this field after writing a hot unit entry at this index. The Hotlist is considered empty when Head == Tail. The Hotlist is considered full when Head == (Tail + 1) mod (Hotlist Size). In this condition, the number of units in the CHMU Hotlist is equal to the Hotlist Size – 1.

8.2.8.7 CHMU Range Configuration Bitmap Register (Offset 10h + CHMU Instance Length * i + CHMU Range Configuration Bitmap Offset[i])

This register is used to enable tracking with a 256-MB DPA range granularity. The valid number of Range Configuration Bitmap fields will cover the total device memory capacity. The register may be padded with reserved bits (with values of 0) until the offset indicated by the device through the CHMU Hotlist Register Offset of the CHMU Capability register. The length of the CHMU Range Configuration Bitmap register is equal to CHMU Hotlist Register Offset - CHMU Range Configuration Bitmap Register Offset. These offset parameters are reported in the CHMU Capability register. The CHMU Range Configuration Bitmap register must be accessed as an 8-byte quantity.

Table 8-TT. CHMU Range Configuration Bitmap Register

Bit	Attributes	Description
63:0	RW	Range Configuration Bitmap 0: Bitmap indicating 256-MB DPA ranges that are enabled to track accesses starting from DPA = 0000 0000 0000 0000h.
127:64	RW	Range Configuration Bitmap 1: Bitmap indicating 256-MB DPA ranges that are enabled to track accesses starting from DPA = 0000 0004 0000 0000h.
...	RW	...

8.2.8.8 CHMU Hotlist Entry Register (Offset 10h + CHMU Instance Length * i + CHMU Hotlist Register Offset[i])

The CHMU Hotlist entries provide the Unit ID that exceeds the programmed hotness threshold and the corresponding counter value. The number of bits for the counter values is the same as the one returned by the device in the CHMU Status register after the unit

size is programmed by the Unit Size field in the CHMU Configuration register. The number of bits associated with the Unit IDs depend on the programmed unit size. The counter bits will be placed in the least significant bits of the entry.

Each CHMU Hotlist Entry register must be accessed as an 8-byte quantity.

Table 8-UU. CHMU Hotlist Register

Bit	Attributes	Description
63:0	RO	Entry 0: CHMU Hotlist Entry 0, which reports the Unit ID and related counter value that exceeds the programmed Hotness Threshold: • Bits[63:N]: Unit ID • Bits[N-1:0]: Counter value N is the Counter width value returned by the device in the CHMU Status register. The counter value is invalid if Always-on Reporting mode is enabled.
127:64	RO	Entry 1. CHMU Hotlist Entry 1, which reports the Unit ID and related counter value exceeding the programmed Hotness Threshold: • Bits[63:N] Unit ID • Bits[N-1:0] Counter value N is the Counter width value returned by the device in the CHMU Status register. The counter value is invalid if Always-on Reporting mode is enabled.
...	RO	...
(64*M-1):(64*(M-1))	RO	Entry M-1¹. CHMU Hotlist Entry M-1, which reports the Unit ID and related value counter exceeding the programmed Hotness Threshold: • Bits[63:N] Unit ID • Bits[N-1:0] Counter value N is the Counter width value returned by the device in the CHMU Status register. The counter value is Invalid if Always-on Reporting mode is enabled.

¹ M is the Hotlist Size in the CHMU Capability register.

14.8.xx

CHMU Register Capability

Prerequisites

- Device supports CHMU register block

Test steps

1. Locate CHMU Register Block within Register Locator DVSEC. Record CMHU_REGISTER_BLOCK_OFFSET
 2. Read CXL_REGISTER_DVSEC_BASE + CMHU_REGISTER_BLOCK_OFFSET
Decode this information:

Bits	Description
2:0	Register BIR
31:16	Register Block Offset

Record CHMU_REGISTER_BASE = Base address indicated by 'Register BIR' + 'Register Block Offset'
 3. Read memory mapped CHMU Common Capability register at address CHMU_REGISTER_BASE length 16 bytes
Decode this information:

Bits	Description
2:0	Version
15:8	Number of supported CHMU Instances
79:64	CHMU Instance Length
 4. Verify

Variable	Value	Condition
Version	1h	Always
Number of supported CHMU Instances	≤ 8h	Always
- Repeat Step 5 – 6 for CHMU Instance number 0 to 'Number of supported CHMU Instances' – 1.
5. Read memory mapped 'CHMU Capability register' at address = "CHMU_REGISTER_BASE + 10h + 'CHMU Instance Length' * CHMU instance number" and length 64 bytes
Decode this information:

Bits	Description
31:16	Max epoch length
47:32	Min epoch length
 6. Verify

Variable	Value	Condition
Max epoch length	!= 0	Always
Min epoch length	!= 0	Always

Pass Criteria:

- Test 14.8.19 Passed
- Verify Conditions are met

Fail Conditions:

- Verify Conditions Failed