Coherent Device Attribute Table (CDAT) Specification

Revision 1.04
November 2023

Revision	Change Description	Revision Date
1.00	First Revision	October 2020
1.02	Formatting Changes	October 2020
1.03	Incorporated https://github.com/pmem/acpi-spec-ecr/pull/1 Added new flags to describe sharable memory and dynamic capacity management. Miscellaneous clarifications.	July 2022
1.04	Incremented the revision field in CDAT Header. Added a new flag to describe Read-only memory mapping. Removed the assertion that missing DSEMTS automatically implies the memory type is EFI Conventional memory. Added EfiUnusableMemory Type encoding to DSEMTS.	November 2023

Contents

Background

Compute Express Link (CXL) and other industry standard coherent interconnects enable coherent switches, coherent memory devices or coherent accelerator devices to be attached to one or more processors. The systems containing such devices are heterogenous in nature and system software needs to understand the topology, device attributes, and affinity information in order to optimize resource assignment. In modern systems, the System Firmware communicates this information to the Operating System via static Advanced Configuration and Power Information (ACPI) Tables such as the System Resource Affinity Table (SRAT) and the Heterogeneous Memory Attribute Table (HMAT). SRAT ACPI table describes various Non-Uniform Memory Access (NUMA) domains in a system including host processors, accelerators, and memory. HMAT ACPI table describes bandwidth and latency from any initiator (a processor or an accelerator) to any memory target. SRAT and HMAT are constructed by system firmware or pre-boot environment.

Prior to emergence of these coherent interconnects, the processors were the sole coherent components in the system. System firmware could be expected to have apriori knowledge of performance attributes of the processors and by extension could construct SRAT and HMAT. That assumption is no longer a valid for systems with these coherent interconnects. Often, these systems are constructed using coherent devices and switches from multiple vendors. Having the system firmware keep track of performance properties of multiple components from various vendors can be impractical. A system firmware update would be needed to deal with a new type of device. Furthermore, some coherent interconnects allow dynamic addition or removal of devices and switches, thus making it even more challenging for system software to gather the performance information.

Coherent Device Attribute Table (CDAT) is introduced to address this challenge. CDAT is a data structure that is exposed by a coherent component and describes the performance characteristics of these components. The types of components include coherent memory devices, coherent accelerators, and coherent switches. For example, CXL accelerators may expose its performance characteristics via CDAT. CDAT describes the properties of the coherent component and is not a function of the system configuration.

Note: The data structures defined in this document are NOT ACPI tables. They are device attributes that may be utilized during the construction of ACPI tables.

CDAT structures discovery

CDAT structures can be discovered and extracted from devices or switches either during boot (by the system firmware), or after boot (by the OS).

During boot, system firmware may extract the CDAT from the component using either Bus specific mechanisms, such as the CXL Data Object Exchange (DOE) services as defined in the CXL 2.0 Specification

or the **EFI_ADAPTER_INFORMATION_PROTOCOL** instance **with EFI_ADAPTER_INFO_CDAT_TYPE_GUID** type (as defined in the UEFI Specification) installed on that device handle. System firmware may discover the performance characteristics of coherent components at boot time via CDAT and use that information during the construction of SRAT and HMAT ACPI tables. In addition, System firmware may use the performance characteristics of individual devices to make decisions such as which memory devices may be interleaved together. Figure 1 shows an example of this pre-boot discovery process for CXL devices.

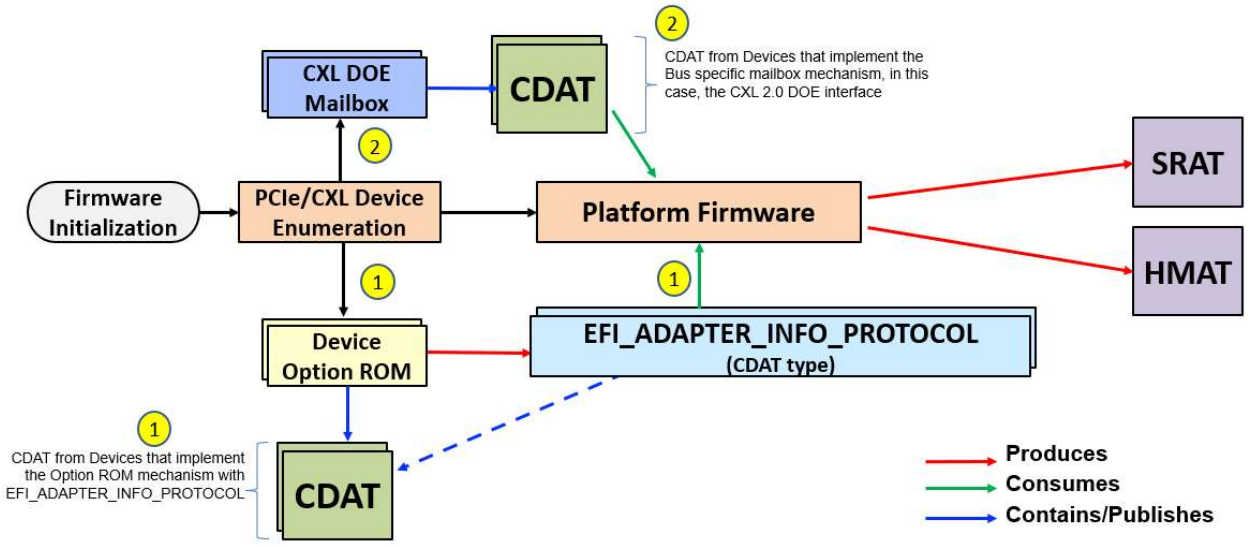

Figure 1 Pre-boot CDAT Extraction Method (for CXL devices)

After the OS has booted (at OS runtime), events such as addition of a component or a significant change to a component's performance attributes may trigger extraction of CDAT from the affected component. The OS may use bus specific mechanisms to extract the CDAT directly from the component (such as the CXL DOE mailbox). Using CDAT, the OS may directly update its internal SRAT and HMAT equivalent structures to account for onlined coherent devices. In addition, the OS may use the performance characteristics of individual devices to make decisions such as which memory devices may be interleaved together. Figure 2 shows an example of this CDAT OS discovery process for CXL devices.

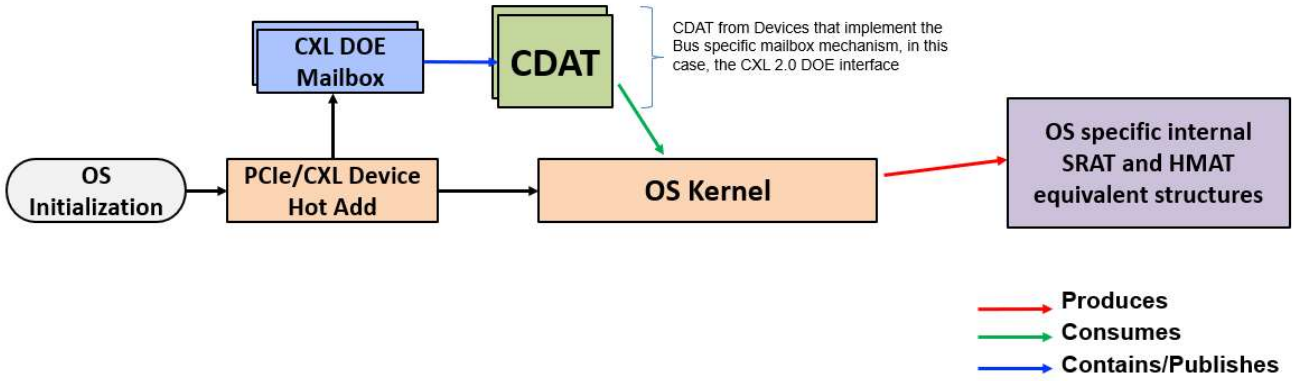

Figure 2 OS Runtime CDAT Extraction Method (for CXL devices)

Definitions

- **Device physical address (DPA**) is the relative address of a memory location within the device. System software is responsible for configuring the bus specific logic to map DPA into System Physical Address (SPA). DPA=0 represents the memory location that is mapped to the lowest system physical address. DPA = (Device Capacity-1) represents the memory location on the device that is mapped to the highest system physical address space.

- **Device Scoped Memory Affinity Domain (DSMAD)** is a contiguous DPA range with unique memory attributes and should be treated as a distinct memory proximity domain by software

- A **coherent memory device** is modelled as one having
  - One or more DSMADs.

- A **coherent accelerator** is modelled as one having
  - One or more Generic Initiators, as defined in ACPI specification
  - Zero or more DSMADs

- A **coherent switch** has two or more bidirectional ports that carry traffic between host CPUs, coherent accelerators, and coherent memory devices.

CDAT Data Structures

Note: All numeric values in CDAT are always encoded in little endian format.

Several data elements in CDAT references various fields in ACPI HMAT table. Please refer to ACPI specification for definition of these fields.

Coherent Device Attribute Table (CDAT)

CDAT consists of a header followed by a variable number of entries.

Table 1- Coherent Device Attribute Table (CDAT) Format

Field	Byte Length	Byte Offset	Description
Header			The following 16 bytes comprise the CDAT header.
Length	4	0	Length, in bytes, of the entire CDAT (includes the CDAT header).
Revision	1	4	Must be set to 3 for the format defined by this specification. Any future revisions will maintain compatibility with prior revisions. Future CDAT revisions are permitted to introduce new structure types or assign meaning to Reserved fields in the CDAT header but are not permitted to redefine the meaning of previously defined fields.
Checksum	1	5	Entire table must sum to zero. Calculated by adding all the bytes in the table.
Reserved	6	6	Reserved

Field	Byte Length	Byte Offset	Description
Sequence	4	12	The contents of CDAT returned by a component may change during runtime. A component shall reset the sequence number to 0 upon reset. Sequence number field shall be incremented by 1 if the content of CDAT being returned is different from the content that was returned last. The sequence field shall roll over after reaching its maximum value. For Header Revision=1, the following changes are permitted during the runtime • Changes to the latency and bandwidth fields in DSLBIS • Changes to the latency and bandwidth fields in SSLBIS • Changes to the number of DSEMTS instances and their contents The changes to latency and bandwidth may represent events such as failover or degradation that are internal to a component. DSEMTS updates may represent memory being brought online or taken offline. For Header Revision=2 and higher, the following changes are permitted during runtime in addition to those changes permitted in Revision 1 • Changes to the DPA base and length fields in DSMAS The changes to DPA base and length may represent runtime configuration changes that affect the range of memory a DSMAS instance applies to (e.g., CXL Set Partition Info). No other changes are permitted. The list of not permitted changes include, but are not limited to • The Number of DSMAS instances • The Number of DSLBIS instances • The Number of DSIS instances and their content • The Number of DSMSCIS instances • The Number of SSLBIS instances If CDAT is being exposed via EFI_ADAPTER_INFORMATION_PROTOCOL, this field shall be set to 0.
CDAT Structure[n]	—	16	A list of CDAT structures for this implementation. This list contains all applicable DSMAS, DSIS, DSLBIS, DSMSCIS, DSEMTS, and SSLBIS entries. These structures are described in the following sections.

Table 2 CDAT Structure Types

Value	Description
0	Device Scoped Memory Affinity Structure (DSMAS)
1	Device Scoped Latency and Bandwidth Information Structure (DSLBIS)
2	Device Scoped Memory Side Cache Information Structure (DSMSCIS)
3	Device Scoped Initiator Structure (DSIS)
4	Device Scoped EFI Memory Type Structure (DSEMTS)
5	Switch Scoped Latency and Bandwidth Information Structure (SSLBIS)
6-FF	Reserved

Device Scoped Memory Affinity Structure (DSMAS)

DSMAS structure is used to return DPA range associated with each DSMAS and its attributes. The number of instances of DSMAS shall equal the number of DSMAD. For example, ACC1, ACC2 and ACC4 in Figure 3 each have one DSMAD. Therefore, each of these devices report one DSMAS entry via CDAT.

Table 3 Device Scoped Memory Affinity Structure

Field	Byte Length	Byte Offset	Description
Type	1	0	0 Device Scoped Memory Affinity Structure (DSMAS)
Reserved	1	1	Reserved
Length	2	2	Length of this record (24)
DSMADHandle	1	4	The handle used to refer to this DSMAD. Each instance of DSMAS shall be associated with a unique DSMADHandle value.

Flags	1	5	Bits 1:0 - Reserved Bit 2 NonVolatile - If set, the memory range represents Non-Volatile memory. Bit 3 Sharable – • 0 - The device maps this DPA range exclusively to the host that is accessing this CDAT and no other hosts. • 1 - The device may map this DPA range to the host that is accessing this CDAT and one or more other hosts. Bit 4 Hardware Managed Coherency – This bit is Reserved if Bit 3=0. If Bit 3 is 1, • 0 – When this range is shared, the software is responsible for managing cache coherency across hosts if necessary. • 1 – When this range is shared, the device manages the cache coherency across hosts if necessary and thereby ensures each host has a consistent view of the memory content. The mechanisms for managing cache coherency is outside the scope of this specification. Bit 5 Interconnect specific Dynamic Capacity Management – Dynamic Capacity Management is the ability of the device to dynamically change the memory capacity that is mapped to the host, without the need for resetting the device. If a device does not support Dynamic Capacity Management for this DPA range, this bit must be set to 0. • 0 –The currently mapped dynamic capacity is reported via DSMAS and DSEMTS structures. A DPA address range that is not currently mapped must be reported as EfiReservedMemoryType via DSEMTS. DSEMTS entries are updated to reflect dynamic capacity changes. • 1 - An interconnect specific mechanism is used to determine the currently mapped dynamic capacity and any changes. The DPA address ranges that are subject to Dynamic Capacity Management must be reported as EfiReservedMemoryType via DSEMTS. Dynamic capacity changes do not result in DSEMTS updates. Bit 6 Read-Only – This range is read-only. Device shall drop any writes to this range. Bits 7 – Reserved More than one bit may be set at a time.

Field	Byte Length	Byte Offset	Description
			Software must ignore the DSMAS entries if it is does not recognize one or more Flag bits that were added after Header Revision=2.
Reserved	2	6	Reserved
DPA Base	8	8	The lowest DPA address associated with this DSMAD
DPA Length	8	16	Length in bytes of this DSMAD. The DPA range of the DSMAD represents the maximum capacity that may be mapped, not necessarily the capacity that is currently mapped. The mechanism for discovering the currently mapped capacity depends upon the value returned in Flags[5] - Interconnect specific Dynamic Capacity Management.

The following mapping table may be used by software to construct internal data structures that are equivalent to SRAT Memory Affinity entries based on the DSMAS entries returned by a component.

If a component is hot-removed, software may utilize this mapping table to locate and eliminate internal data structures corresponding to the component.

Table 4 Mapping between DSMAS and SRAT Memory Affinity structure

SRAT Memory Affinity Structure Field	DSMAS field	Expected Mapping
Proximity Domain	DSMADHandle	In absence of interleaving across components, software may map one or more DSMAD Handles exposed by a single component to a single proximity domain in SRAT. If memory is interleaved across multiple components, each interleave set may be mapped to a single proximity domain. An interleave set is composed of memory contributed by multiple components that are part of the interleave.
Base Address, Low and High	DPA Base	The Base address field in SRAT is derived by mapping the DPA base in SPA space.
Length, Low and High	DPA Length	The Length field in SRAT is derived from the DPA length after accounting for interleaving.
Flags	Flags	Non-volatile attribute comes from DSMAS. Hot Pluggable flag in SRAT may be constructed by the system software using system specific knowledge.

Device scoped Latency and Bandwidth Information Structure (DSLBIS)

Table 5 Device Scoped Latency and Bandwidth Information Structure

Field	Byte Length	Byte Offset	Description
Type	1	0	1 Device scoped Latency and Bandwidth Information Structure (DSLBIS)
Reserved	1	1	Reserved
Length	2	2	Length of this record (24)
Handle	1	4	Either a DSMAS handle or an Initiator Handle with no memory attached.
Flags	1	5	Must be ignored if Handle represents an Initiator with no memory attached. If Handle matches a DSMAS handle, the definition of this field matches *Flags* field in HMAT System Locality Latency and Bandwidth Information Structure.
Data Type	1	6	Must be ignored if Handle represents an Initiator with no memory attached. If Handle matches a DSMAS handle, the definition of this field matches *Data Type* field in HMAT System Locality Latency and Bandwidth Information Structure.
Reserved	1	7	Reserved
Entry Base Unit	8	8	The definition of this field matches *Entry Base Unit* field in HMAT System Locality Latency and Bandwidth Information Structure.

Field	Byte Length	Byte Offset	Description
Entry[3]	6	16	If the handle matches a DSIS handle with no memory attached, a single latency or bandwidth number is provided that represent the pathway between the device ingress and the initiator within the device. This information is provided in the first WORD. The 2nd and the 3rd WORDs are set to 0. The System firmware or software may combine this with rest of the system information (e.g. system topology, host characteristics, information from other devices) to construct bandwidth and latency characteristics as seen by the initiator within the device. If the handle matches a DSMAS handle and this structure represents a memory side cache, a single latency or bandwidth number is provided based on device ingress as the reference point. This information is provided in the first WORD. The 2nd and the 3rd WORDs are set to 0. System firmware/software may combine this with rest of the system information (e.g. system topology, host characteristics, information from other devices) to construct memory side cache performance characteristics as seen by the initiators external to the device. If the handle indicates a memory only device (DSMAS handle, not referenced by any DSIS) and this structure represents memory, a single latency or bandwidth number are provided. That number represents the pathway between the device ingress and the device memory. This information is provided in the first WORD. The 2nd and the 3rd WORDs are set to 0. System firmware/software may combine this with rest of the system information (e.g. system topology, host characteristics, information from other devices) to construct memory performance characteristics as seen by the initiators external to the device. If the handle represents an Initiator with memory attached (DSMAS handle, also referenced by a DSIS) and this structure represents memory, three latency or bandwidth numbers are provided. The first WORD represents the pathway between the device ingress and the device memory. The second WORD represents the pathway between the device egress and the initiator inside the device. The third WORD represents the pathway between the initiator inside the device and the device memory.
Reserved	2	22	Reserved

Device Scoped Memory Side Cache Information Structure (DSMSCIS)

Table 6 Device Scoped Memory Side Cache Information Structure

Field	Byte Length	Byte Offset	Description
Type	1	0	2 Device Scoped Memory Side Cache Information Structure (DSMSCIS)
Reserved	1	1	Reserved
Length	2	2	Length of this record (20)
DSMASHandle	1	4	Handle of the DSMAS entry that represents the memory being cached by this instance of memory side cache
Reserved	3	5	Reserved
Memory Side Cache Size	8	8	The definition of this field matches *Memory Side Cache Size* field in HMAT Memory Side Cache Information Structure
Cache Attributes	4	16	The definition of this field matches *Cache Attributes* field in HMAT Memory Side Cache Information Structure

This structure describes memory side caches that are internal to the coherent device. HMAT memory cache structure also includes the SMBIOS Type 17 handles that represents the memory side cache physical devices. When the system firmware constructs HMAT based on CDAT, it shall set the "Number of SMBIOS handles" field in these HMAT structure to be 0. This is because the memory side cache on a coherent device is not a FRU, and thus will not have a corresponding SMBIOS Type 17 records.

Device Scoped Initiator Structure (DSIS)

DSIS structure is used to return the ACPI Initiators that are part of the device. The number of instances of DSIS shall equal the number of Initiators. The Initiator may be part of the same proximity domain as memory (Flags[0]=1) or it may be an initiator with no memory attached(Flags[0]=0).

Table 7 Device Scoped Initiator Structure

Field	Byte Length	Byte Offset	Description
Type	1	0	3 Device Scoped Initiator Structure (DSIS)
Reserved	1	1	Reserved
Length	2	2	Length of this record (8)

Field	Byte Length	Byte Offset	Description
Flags	1	4	Bit 0 – 1 indicates that the Initiator has memory attached. In this case, this Initiator may be represented as memory proximity node by system firmware. Bits 7:1 : Reserved
Handle	1	5	If Bit 0 of Flags field is 1, this field represents the DSMAS handle associated with this initiator. Otherwise, this field represents a handle that can be used to reference the Initiator in DSLBIS.
Reserved	2	6	Reserved

If Flags[0]=0, the Device Handle field of SRAT Generic Initiator (or equivalent) structure may be used to corelate with the component containing the Initiator.

If Flags[0]=1, the Initiator is represented as SRAT Memory Affinity (or equivalent) structure.

Device Scoped EFI Memory Type Structure (DSEMTS)

DSEMTS structure is used to communicate the expected memory usage and any associated hints associated with different subranges of device memory. The number of DSEMTS ranges can exceed the number of DSMAS entries since DSMAS entries often represent hardware constructs and DSEMTS represent software usage. DPA ranges covered by DSEMTS entries must not overlap and must fit within the DPA range associated with the associated DSMAS Handle.

Table 8 Device Scoped EFI Memory Type Structure

Field	Byte Length	Byte Offset	Description
Type	1	0	4 Device Scoped EFI Memory Type Structure (DSEMTS)
Reserved	1	1	Reserved
Length	2	2	Length of this record (24)
DSMAS Handle	1	4	Device Scoped Memory Affinity Structure (DSMAS) handle this record references

Field	Byte Length	Byte Offset	Description
EFI Memory Type and Attribute	1	5	0 – EfiConventionalMemory 1 - EfiConventionalMemory Type with EFI_MEMORY_SP Attribute 2 – EfiReservedMemoryType 3 – EfiUnusableMemory Type 4-255 – Reserved encoding The memory attribute EFI_MEMORY_NV may be inferred from NonVolatile flag in DSMAS.
Reserved	2	6	Reserved
DPA Offset	8	8	Offset from base of the DSMAS referenced by the Handle
DPA Length	8	16	Length of this range in bytes.

Switch Scoped Latency and Bandwidth Information Structure (SSLBIS)

This structure represents the latency and bandwidth characteristics of various data paths within a coherent switch.

Table 9 Switch Scoped Latency and Bandwidth Information Structure

Field	Byte Length	Byte Offset	Description
Type	1	0	5 Switch Scoped Latency and Bandwidth Information Structure (SSLBIS)
Reserved	1	1	Reserved
Length	2	2	Length of this record
Data Type	1	4	See *Data Type* field in HMAT System Locality Latency and Bandwidth Information Structure for Memory Hierarchy=0.
Reserved	3	5	Reserved
Entry Base Unit	8	8	See *Entry Base Unit* field in HMAT System Locality Latency and Bandwidth Information Structure.

Field	Byte Length	Byte Offset	Description
SSLBE Entry[]	8 * n	16	Variable number of SSLBE entries describing the latency and bandwidth between pairs of ports. The format of each entry is defined in Table 10 Switch Scoped Latency and Bandwidth Entry (SSLBE). These entries should describe all legal data paths within the switch. If any port to any port latency is the same, shorthand Port ID of 0FFFFh allows latency characteristics of such a switch to be described using a single SSLBE entry. If any port to any port bandwidth is the same, shorthand Port ID of 0FFFFh allows bandwidth characteristics of such a switch to be described using a single SSLBE entry.

Table 10 Switch Scoped Latency and Bandwidth Entry (SSLBE)

Field	Byte Length	Byte Offset	Description
Port X ID	2	0	Identifies the first Port. 0FFFFh – shorthand that represents any port For the remaining values, interpretation of Port identifier is specific to the bus. For a CXL bus, Port ID value of 100h represents an upstream port. For downstream ports, this field matches the *Port Number* field in the Link Capabilities Register of the port being described.
Port Y ID	2	2	Identifies the second Port. 0FFFFh – shorthand that represents any port For the remaining values, interpretation of Port identifier is bus specific. For a CXL bus, Port ID value of 100h represents an upstream port. For downstream ports, this field matches the *Port Number* field in the Link Capabilities Register of the port being described.
Latency or Bandwidth	2	4	Latency or bandwidth information for path between Port X and Port Y, as defined in System Locality Latency and Bandwidth Information Structure. It is assumed that the latency and BW is identical in both directions. The switch does not provide an entry with Port X and Y swapped.
Reserved	2	6	Reserved

Example

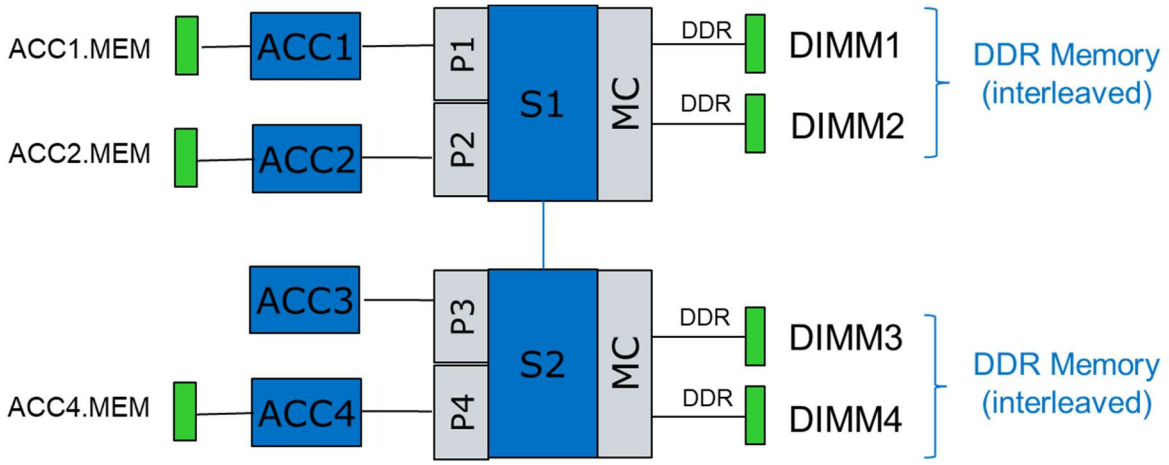

Figure 3 Configuration 1

Figure 3 represents a system configuration where two coherent accelerators, namely ACC1 and ACC2 are attached to CPU S1 via a coherent interconnect such as CXL. Two accelerators, ACC3 and ACC4 are connected to CPU S2 via the same coherent interconnect. Each CPU also has a local memory controller with two DDR channels and one DIMM attached to each channel.

The system firmware may combine the information it has about the CPU and various CPU links with CDAT extracted from each of the coherent accelerators.

In this example, it is assumed that read latency is always equal to the write latency for every data path and read bandwidth is always equal to the write bandwidth for every data path.

Information known to system firmware (apriori knowledge):

- DIMM1, DIMM2, DIMM3, DIMM4 size = 128GB
- DDR Read/Write Latency = 50ns
- DDR Bandwidth = 20GB/s/DDR channel
- S1 – S2 access latency = 50ns
- S1 – S2 bandwidth = 30GB/s
- Coherent Interconnect Latency = 40 ns
- Coherent Interconnect Bandwidth = 30 GB/s

System firmware is aware that ACC1 memory is mapped starting at System Physical Address (SPA) of 256 GB. ACC2 memory base SPA is at 272 GB and ACC4 memory base SPA is at 536 GB.

ACC1 returns the following CDAT entries

- One DSMAS Entry, DPA Base = 0, DPA Length = 16 GB, handle = 0
- One DSIS entry, associated DSMAS Handle =0
- DSLBIS entries which state latency for all 3 data paths is 60 ns and bandwidth for all 3 data paths is 80 GB/s

ACC2 returns the following CDAT entries

- One DSMAS Entry, DPA Base = 0, DPA Length = 8 GB, handle = 0
- One DSIS entry, associated DSMAS Handle =0
- DSLBIS entries which state latency for all 3 data paths is 60 ns and bandwidth for all 3 data paths is 80 GB/s

ACC3 returns the following CDAT entries

- One DSIS entry which is not associated with any DSMAS
- DSLBIS entries which state latency for the ingress to the initiator data path is 60 ns and bandwidth for the ingress to the initiator data path is 80 GB/s

ACC4 returns the following CDAT entries

- One DSMAS Entry, DPA Base = 0, DPA Length = 32 GB, handle = 0
- One DSIS entry, associated DSMAS Handle =0
- DSLBIS entries which state latency for all 3 data paths is 60 ns and bandwidth for all 3 data paths is 80 GB/s

Using the above information, the system firmware concludes that each accelerator must be described as a separate proximity domain in SRAT. ACC1, ACC2 and ACC4 each have a Generic Initiator as well as memory associated with them, whereas ACC3 appears as a Generic Initiator only proximity domain. The system firmware is also able to construct Memory Proximity Domain Attributes Structure in HMAT. This is shown in Figure 4.

SRAT				
Proximity Domain	Type	SPA Base	Length	Note
0	Processor	S1 APIC IDs		S1
0	Memory	0	256GB	DIMM1,DIMM2
1	GI			ACC1
1	Memory	256 GB	16 GB	ACC1.MEM
2	GI			ACC2
2	Memory	272 GB	8 GB	ACC2.MEM
3	Processor	S2 APIC IDs		S2
3	Memory	280 GB	256 GB	DIMM3,DIMM4
4	GI			ACC4
4	Memory	536 GB	32GB	ACC4.MEM
5	GI			ACC3

Memory Proximity Domain Attributes		
Flags	Initiator Proximity Domain	Memory Proximity Domain
IPD Valid	0	0
IPD Valid	1	1
IPD Valid	2	2
IPD Valid	3	3
IPD Valid	4	4

Figure 4 SRAT Summary and HMAT MPD Attribute Structure

The system firmware is also able to calculate the latency from any initiator to any target by simply adding the latency contribution of every hop in the data path. Similarly, the system firmware is also able to calculate the bandwidth from any initiator to any target by selecting the smallest value among the bandwidth associated with various hops in the data path. It is assumed that 2 way interleaving across DDR channels doubles the effective bandwidth. The results are shown in Figure 5.

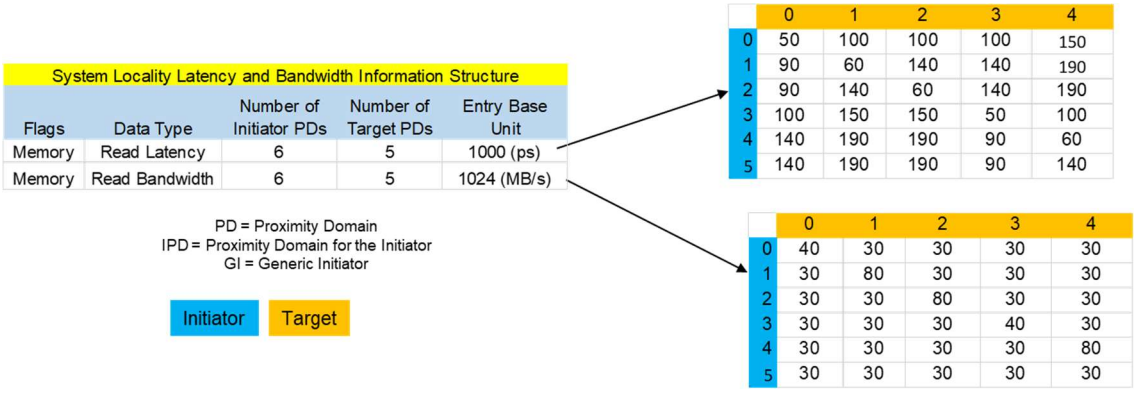

Figure 5 HMAT System Locality and Bandwidth Information Structure Summary

If ACC1 is removed from the system, software may wish to remove ACC1 related entries from these structures. Software may use bus specific mechanisms to determine that ACC1 memory base is 256 GB and size is 16 GB. By matching these addresses against the SRAT entries, software can unambiguously determine that proximity domain 1 represents ACC1. Software may map domain 1 entries in SRAT as invalid and purge the corresponding entries from HMAT.

If another ACC3 like device is dynamically added to the system, Operating System may extract CDAT from that device and insert new entries in the OS internal structure that is equivalent to SRAT and a new row in the OS internal structure that is equivalent to HMAT using an algorithm like the one used by the system firmware.