



OpenCAPI AFU Address Space Usage

OpenCAPI Work Group Notes

Version 1.0
28 August 2020

Approved

Approved

OpenCAPI AFU Address Space Usage

OpenCAPI Enablement Work Group
OpenCAPI Consortium

Version 1.0 (28 August 2020)

Copyright © OpenCAPI Consortium 2020

All capitalized terms in the following text have the meanings assigned to them in the OpenCAPI Intellectual Property Rights Policy (the "OpenCAPI IPR Policy"). The full Policy may be found at the OpenCAPI Consortium website.

This document and the information contained herein is provided on an "AS IS" basis AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, THE OPENCAPI CONSORTIUM AS WELL AS THE AUTHORS AND DEVELOPERS OF THIS DRAFT STANDARD OR OTHER DOCUMENT HEREBY DISCLAIM ALL OTHER WARRANTIES AND CONDITIONS, EITHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES, DUTIES OR CONDITIONS OF MERCHANTABILITY, OF FITNESS FOR A PARTICULAR PURPOSE, OF ACCURACY OR COMPLETENESS OF RESPONSES, OF RESULTS, OF WORKMANLIKE EFFORT, OF LACK OF VIRUSES, OF LACK OF NEGLIGENCE OR NON-INFRINGEMENT.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OpenCAPI, except as needed for the purpose of developing any document or deliverable produced by an OpenCAPI Work Group (in which case the rules applicable to copyrights, as set forth in the OpenCAPI IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OpenCAPI or its successors or assigns while the specification in question is the current version. Upon release by the OpenCAPI Consortium of a new version, all the above rights shall automatically cease.

OpenCAPI and the OpenCAPI logo design are trademarks of the OpenCAPI Consortium.

Other company, product, and service names may be trademarks or service marks of others.

Abstract

This document contains Work Group Notes on the OpenCAPI AFU Address Space usage and definition. This Work Group note is intended to go along with the OpenCAPI Discovery and Configuration Architecture Specification. It is the work product of the OpenCAPI Consortium Enablement Work Group.

This document is handled in compliance with the requirements outlined in the OpenCAPI Consortium Work Group (WG) process document. Comments, questions, etc. can be submitted to membership@opencapi.org.

Approved

Participants

Sanjay Goyal, Microchip Technology, *Chair*

Brian Bakke, IBM, *Technical Editor*

Brian Allison, IBM

Myron Slota, IBM

Contents

List of figures	5
List of tables	5
Revision log	6
About this document	7
Notes	7
Engineering notes	7
Developer notes	7
Terms	8
1. CFG and AFU Memory	9
1.1 AFU special purpose memory	14
1.2 Potential use cases	14

List of figures

Figure 1-1. Memory, MMIO, and PASIDs (for 1 AFU)	9
Figure 1-2. Memory, MMIO, and PASIDs (for 1 AFU and Expansion ROM)	10
Figure 1-3. Memory, MMIO, and PASIDs (for 1 AFU and AFU Special Purpose Memory)	11
Figure 1-4. Memory, MMIO, and PASIDs (for Multiple AFUs)	12
Figure 1-5. Memory, MMIO, and PASIDs (for multiple AFUs including AFU Special Purpose Memory).....	13

List of tables

Table 1-1. Example of AFU MEM space contents	13
--	----

Approved

Revision log

Each release of this document supersedes all previously released versions. The revision log lists all significant changes made to the document since its initial release.

Revision date	Summary of changes
28 August 2020	Version 1.0. Initial version.

About this document

This section contains some general format information (for example, notes) and document conventions.

Notes

This section describes Engineering and Developer.

Engineering notes

Engineering notes provide additional implementation details and recommendations not found elsewhere. The notes might include architectural compliance requirements. That is, the text might include Architecture compliance terminology. These notes should be read by all implementation and verification teams to ensure architectural compliance.

Engineering note:

This is an example of an Engineering note. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin cursus hendrerit enim, vel tempus nibh ornare ut. Quisque ac augue eu augue convallis hendrerit. Mauris iaculis viverra ipsum nec dapibus. Nunc at porta libero. Curabitur luctus ultrices augue non pulvinar. Vestibulum mattis non ipsum at venenatis. Suspendisse euismod, neque et suscipit luctus, odio metus semper lectus, quis volutpat est libero quis nunc. Vivamus rutrum mauris sed tristique malesuada.

Developer notes

Developer notes are used to document the reasoning and discussions that led to the current version of the architecture. These notes might also include recommended changes for future versions of the architecture, or warnings of approaches that have failed in the past. These notes should be read by verification teams and contributors to the architecture.

Developer note:

This is an example of a Developer note. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin cursus hendrerit enim, vel tempus nibh ornare ut. Quisque ac augue eu augue convallis hendrerit. Mauris iaculis viverra ipsum nec dapibus. Nunc at porta libero. Curabitur luctus ultrices augue non pulvinar. Vestibulum mattis non ipsum at venenatis. Suspendisse euismod, neque et suscipit luctus, odio metus semper lectus, quis volutpat est libero quis nunc. Vivamus rutrum mauris sed tristique malesuada.

Approved

Terms

The following terms are used in this document.

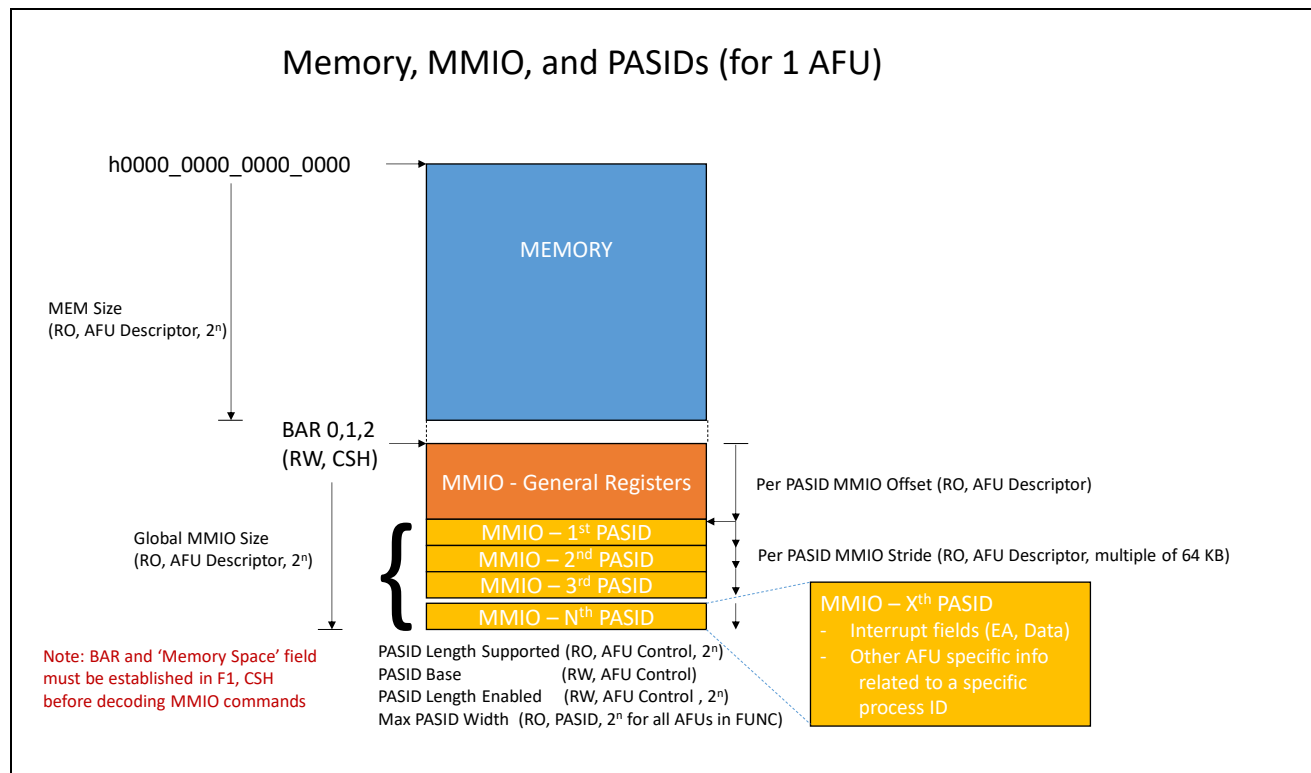
AFU	Accelerator functional unit
BAR	Base Address Register
CFG	Configuration
EA	Effective address
MMIO	Memory mapped input/output
PASID	Process address space ID
ROM	Read only memory

Approved

1. CFG and AFU Memory

Figure 1-1 illustrates the memory, MMIO, and PASIDs for a single accelerator functional unit (AFU).

Figure 1-1. Memory, MMIO, and PASIDs (for 1 AFU)

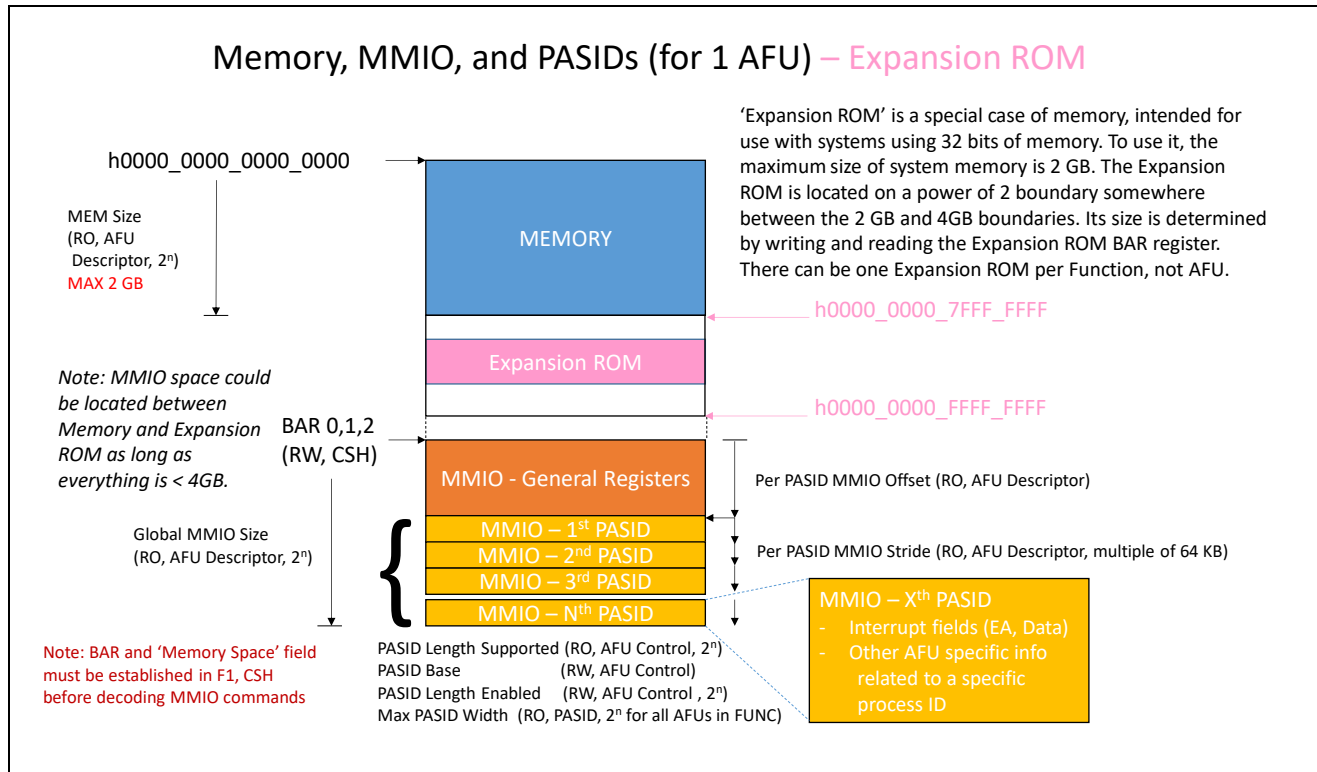


Approved

Figure 1-2 illustrates the memory, MMIO, and PASIDs for a single AFU for the case of where expansion ROM exists. 'Expansion ROM' is a special case of memory, intended for use with systems using 32 bits of memory. To use it, the maximum size of system memory is 2 GB. The expansion ROM is located on a power of 2 boundary somewhere between the 2 GB and 4GB boundaries. Its size is determined by writing and reading the expansion ROM BAR register.

Note: There can be one expansion ROM per function, not per AFU.

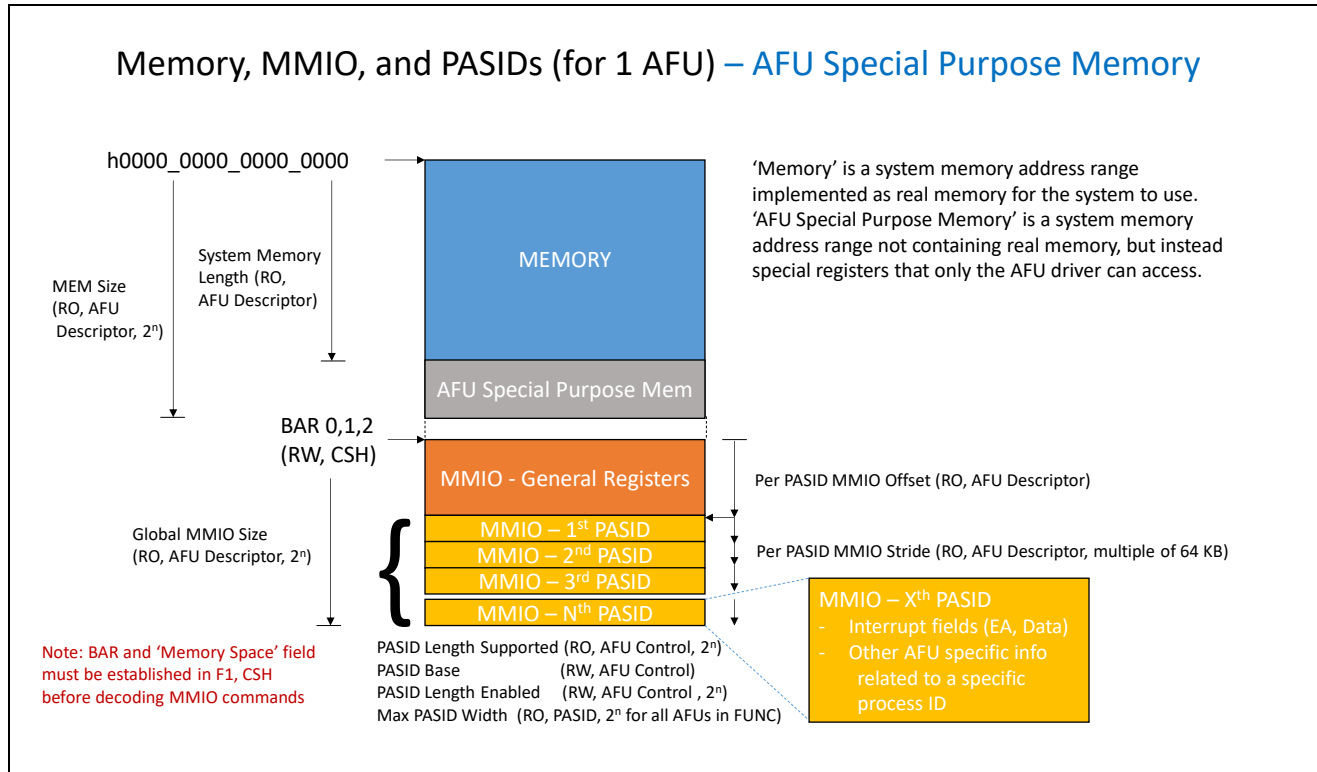
Figure 1-2. Memory, MMIO, and PASIDs (for 1 AFU and Expansion ROM)



Approved

Figure 1-3 illustrates the memory, MMIO, and PASIDs for a single AFU using AFU special purpose memory. 'Memory' is a system memory address range implemented as real memory for the system to use. AFU special purpose memory' is a system memory address range not containing real memory, but instead special registers that only the AFU driver can access.

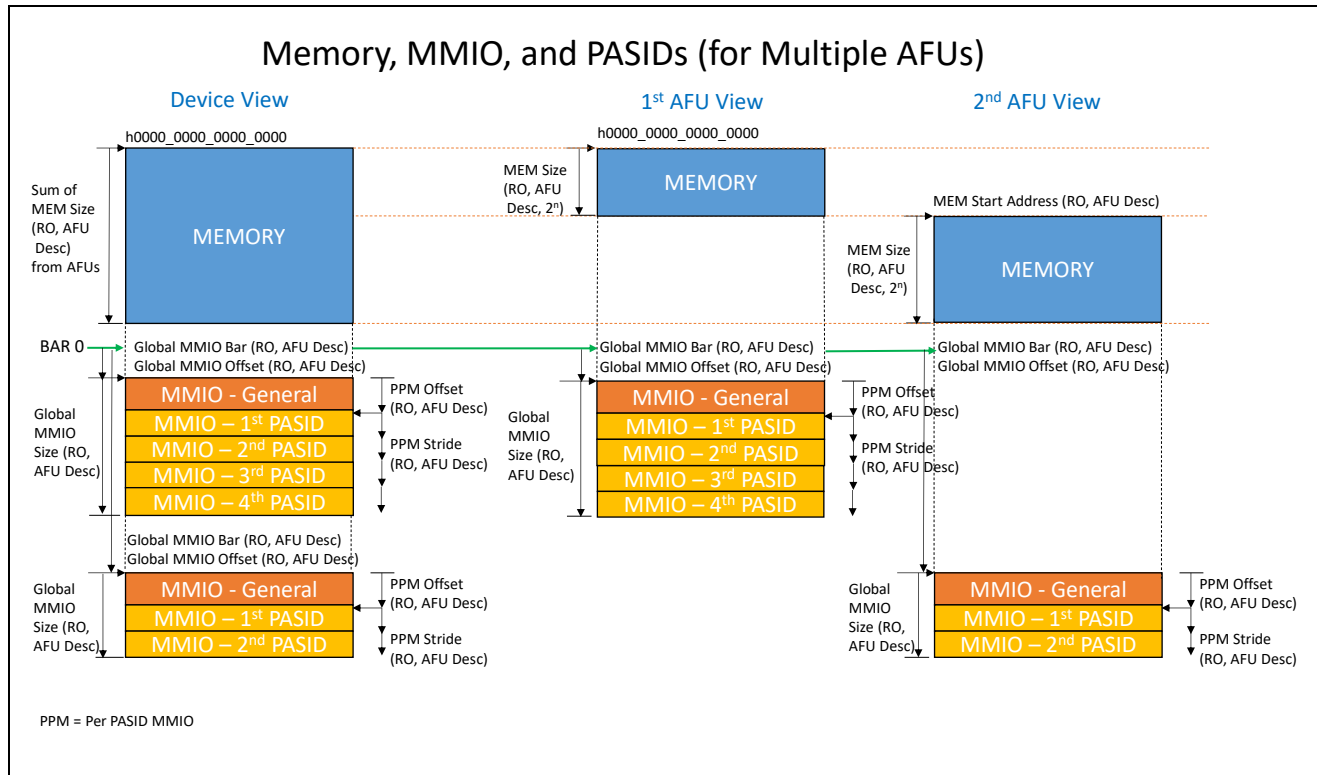
Figure 1-3. Memory, MMIO, and PASIDs (for 1 AFU and AFU Special Purpose Memory)



Approved

Figure 1-4 illustrates memory, MMIO, and PASIDs (for multiple AFUs).

Figure 1-4. Memory, MMIO, and PASIDs (for Multiple AFUs)



Approved

Figure 1-5 illustrates Memory, MMIO, and PASIDs for multiple AFUs including AFU special purpose memory.

Figure 1-5. Memory, MMIO, and PASIDs (for multiple AFUs including AFU Special Purpose Memory)

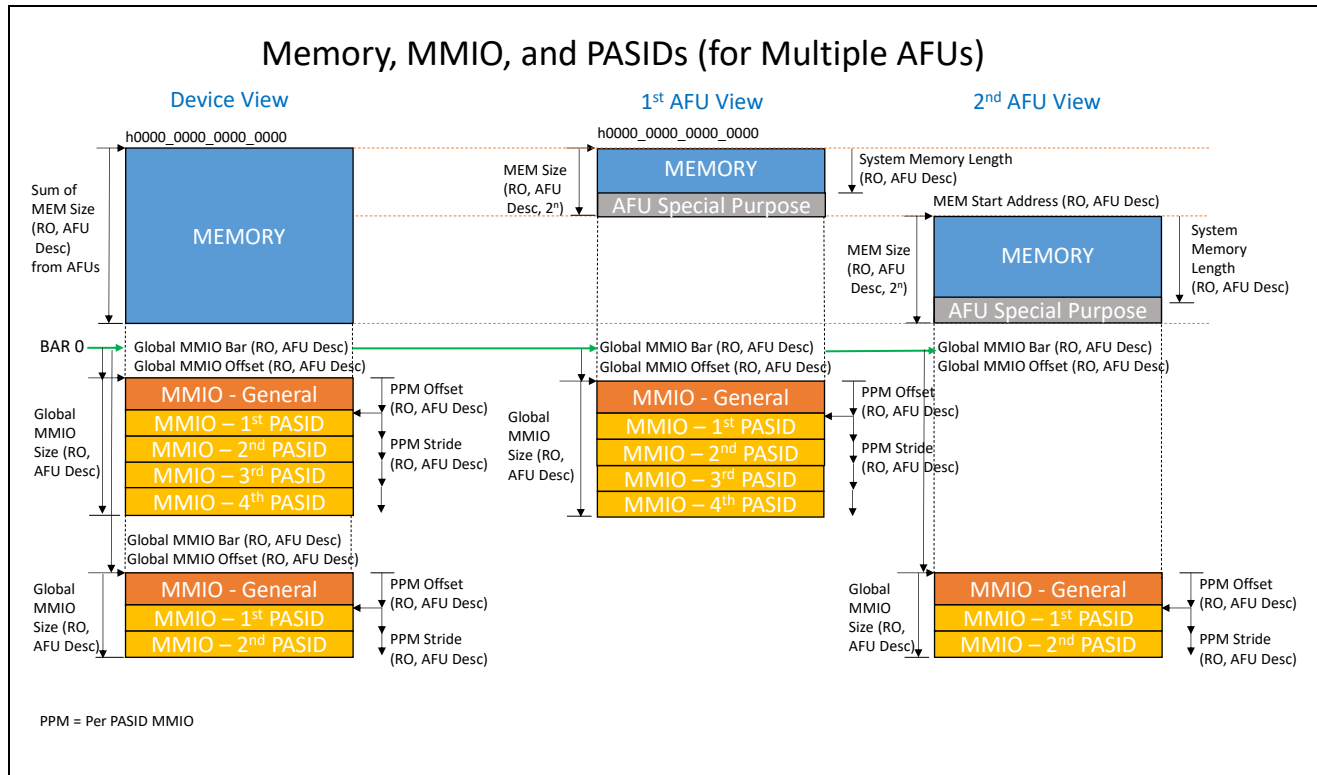


Table 1-1 shows an example of the AFU memory space contents

Table 1-1. Example of AFU MEM space contents

Contents			
Example 1	Example 2	Example 3	Example 4
Power of 2 System Memory only (System Memory Length = MEM Space Size)	Non-Power of 2 System Memory only (0 < System Memory Length < MEM Space Size)	System Memory + AFU Special Purpose Memory (0 < System Memory Length < MEM Space Size)	AFU Special Purpose Memory only (System Memory Length = 0)
Address: MEM Start Address High MEM Start Address Low			
General Purpose System Memory	General Purpose System Memory Address: (MEM Start Address High MEM Start Address Low) + (System Memory Length High System Memory Length Low)		Address: (MEM Start Address High MEM Start Address Low) + (System Memory Length High System Memory Length Low)
	(empty address space)		
Address: (MEM Start Address High MEM Start Address Low) + MEM Space size			

1.1 AFU special purpose memory

- Address space contained within power of 2 MEM window.
 - Not part of the kernel's general purpose System Memory footprint (for example, Heap, malloc space).
- AFU design is unique.
 - AFU design determines the contents/usage of this address space.
 - AFU-specific software (driver or user-space application) requests the address mapping of this range from the Kernel.
 - AFU-specific software directly accesses via this address mapping.
- Access characteristics.
 - Space is cacheable, accessed via load or store semantics.
 - Cache-line load or cache-line store operations (for example, 128 bytes) versus MMIO (for example 4 or 8 bytes)

1.2 Potential use cases

- Physical memory not a power of 2
 - Allow space / cost tradeoffs (for example, storage class memory)
 - Trims MEM address range to backing physical memory size
- Use memory commands to access special AFU registers
 - Higher bandwidth / lower latency for greater than 8-byte data transfers (rather than MMIO)