AMBA Coherent Hub Interface (CHI)

Evolution from ACE

CHI is a successor to the AXI Coherency Extensions (ACE) protocol, enhancing it with additional features for better coherency and system performance.

AMBA Standard:

As part of the AMBA standard by Arm, CHI ensures compatibility and interoperability in SoC designs, making it easier to integrate various components seamlessly.

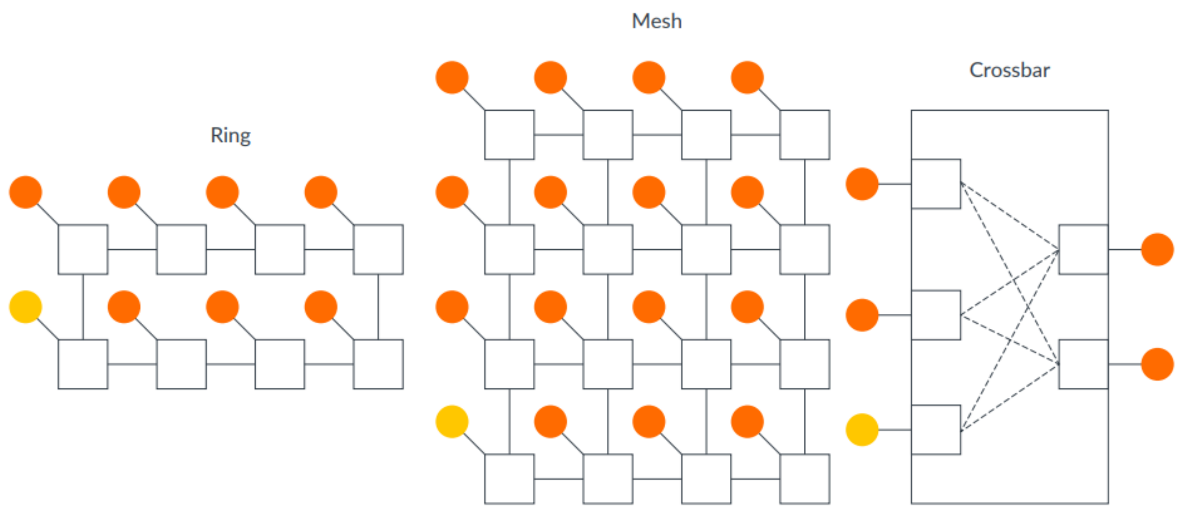
Applicability:

CHI is suitable for diverse applications including mobile, networking, automotive, and data centers, indicating its versatility and wide usability.

Topology Flexibility:

It offers flexibility in connecting components using different topologies like ring, mesh, or crossbar, allowing designers to optimize based on performance, power, and area requirements.

Figure 2-1: Possible topologies for CHI implementation



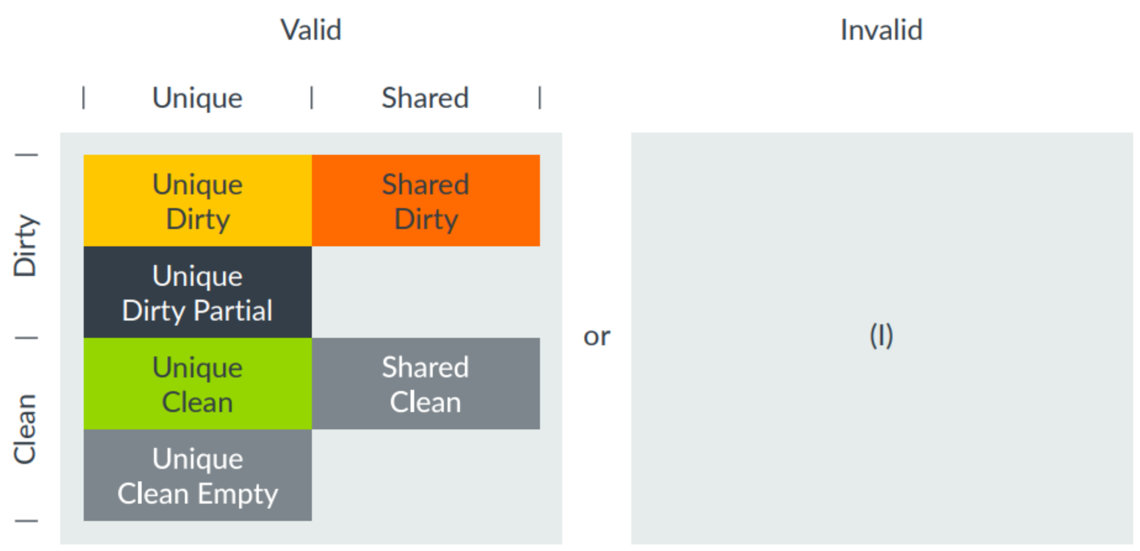
Protocol Versions:

CHI comes in three versions - CHI-A, CHI-B, and CHI-C, each building upon the previous version with added features for improved performance and compatibility with Armv8.1 and Armv8.2 extensions.

Coherency Model:

Similar to ACE, CHI supports snoop filters and directory-based systems for efficient coherency management. CHI also uses the same terms as ACE to define cache states and adds partial and empty cache line states. The cache line states the terms are:

Figure 2-2: A diagram showing cache line states



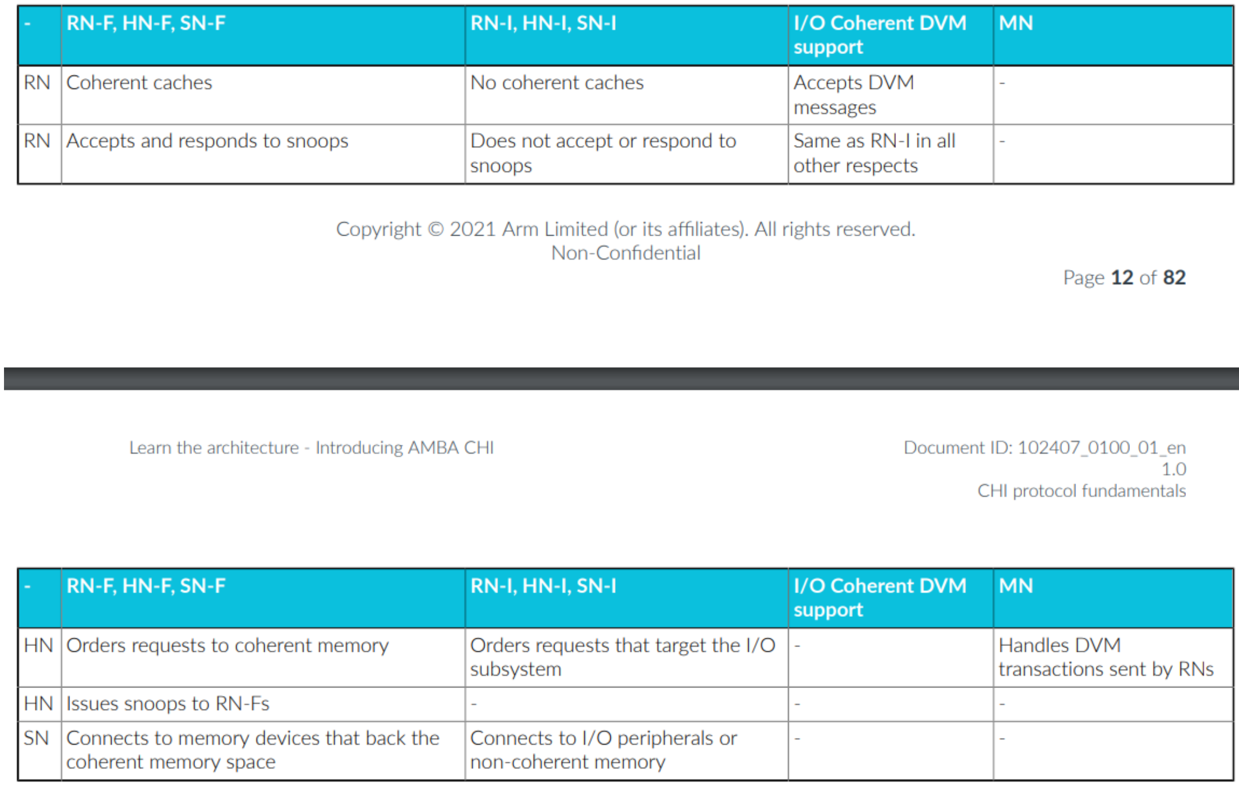
Node Types:

CHI classifies components into Request Nodes (RNs), Home Nodes (HNs), and Subordinate Nodes (SNs), facilitating communication and data transfer within the system.

1. **Request Nodes (RNs):**
    - Role: Request Nodes are components within the system that generate transactions, such as read and write requests, to access data from memory or other components.
    - Responsibilities: RNs initiate transactions and send them to Home Nodes (HNs) for further processing. They are responsible for generating requests for data access and initiating coherent transactions within the system.
2. **Home Nodes (HNs):**
    - Role: Home Nodes play a central role in the CHI protocol. They are responsible for coordinating and managing transactions between Request Nodes (RNs) and Subordinate Nodes (SNs).

- Responsibilities: HNs receive transactions from RNs, order these requests, generate transactions to SNs for data access or modification, and handle snoops or Distributed Virtual Memory (DVM) operations. They ensure coherence and consistency in data access within the system.

3. **Subordinate Nodes (SNs):**
   - Role: Subordinate Nodes are components within the system that store data and respond to transactions initiated by Request Nodes (RNs) or Home Nodes (HNs).
   - Responsibilities: SNs receive transactions from HNs, process these requests, and respond accordingly by providing data or performing requested operations. They store data and ensure coherence by responding to requests in a timely and consistent manner.
4. **Miscellaneous Node (MN):**
   - Role: Miscellaneous Nodes encompass components that do not fit into the categories of RNs, HNs, or SNs but still play a role in the coherency protocol.
   - Responsibilities: MNs may have various functions within the system, such as providing additional processing capabilities, managing specific peripherals, or handling specialized tasks related to system operation.

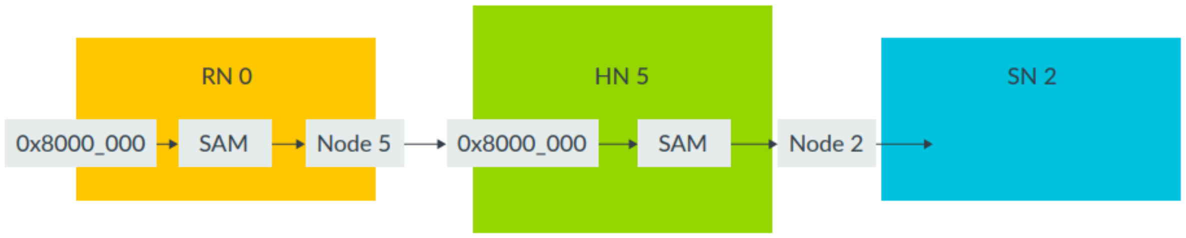


-	RN-F, HN-F, SN-F	RN-I, HN-I, SN-I	I/O Coherent DVM support	MN
RN	Coherent caches	No coherent caches	Accepts DVM messages	-
RN	Accepts and responds to snoops	Does not accept or respond to snoops	Same as RN-I in all other respects	-
HN	Orders requests to coherent memory	Orders requests that target the I/O subsystem	-	Handles DVM transactions sent by RNs
HN	Issues snoops to RN-Fs	-	-	-
SN	Connects to memory devices that back the coherent memory space	Connects to I/O peripherals or non-coherent memory	-	-

Unique Node ID:

Each component in the system is assigned a Unique Node ID, enabling efficient routing of transactions using the System Address Map (SAM).

Figure 3-1: The RN SAM maps physical addresses to HN Node IDs, and the HN SAM maps physical addresses to SN Node IDs



Flits and Handshake Mechanism:

CHI uses Flits as packetized messages and employs a handshake mechanism for communication, ensuring reliable data transfer.

Ordering Mechanism:

Transactions in CHI are ordered based on Endpoint Order and Request Order, maintaining consistency and reliability.

Endpoint Order maintains the order of transactions from a single requester to a single subordinate address range. For example in Endpoint Order, multiple device accesses are issued to the programmable register bank of a subordinate.

Request Order maintains the order of transactions from a single requester to the same address. For example, ordering is required when multiple requests are issued to an overlapping noncacheable address such as Normal NC, Device-GRE and Device-nGRE. CHI does not require an exact granularity for address matching when Request Order is set, and the granularity is defined by the implementation.

Request Retry:

CHI provides a mechanism for retrying requests to prevent blocking when resources are unavailable, enhancing system robustness.

Cache Stashing:

Introduced in CHI-B, cache stashing improves system performance by allocating cache lines near their future usage points, reducing memory access latency.

Cache stashing is a mechanism to install data within specific caches in a system. CHI-B introduced this feature to improve system performance. The Cache Stashing mechanism improves system performance by allocating a cache line near its future point of use. This results in lower memory access latency when the data is used.
Cache Stashing support was added to the ACE5-Lite protocol. The CHI protocol is very flexible regarding Cache Stashing and allows a Stash Request to take multiple forms.

I/O Deallocation Transactions:

Purpose: CHI-B allows an I/O Requester to deallocate cache lines in a Fully Coherent Node.
Functionality: These transactions provide hints to invalidate cache lines and manage Dirty data by either writing it back to memory or discarding it.
Flexibility: While these requests are hints and can be ignored by Fully Coherent Nodes, they help in avoiding cache pollution with data not likely to be used in the near future.
Types of Requests: CHI defines two types of requests for I/O deallocation - ReadOnceCleanInvalid and ReadOnceMakeInvalid, with the latter not requiring the writing of Dirty data to the next level of memory.

Direct Memory Transfer (DMT) and Direct Cache Transfer (DCT) Mechanisms:

Purpose: Introduced in CHI-B to reduce access latency by bypassing the Home Node for read and snoop data transfer.

Functionality: DMT and DCT operations allow for the direct transfer of data between components without involving the Home Node.

Additional Identifiers: To support DMT and DCT operations, additional identifiers are added to the Request, Snoop, and Data flits, specifying the necessary routing information.

Atomic Transactions:

Purpose: Support for atomic instructions added in the Armv8.1 architecture.
Functionality: Atomic Transactions transport atomic operations and their operands between devices, reducing the time data is inaccessible to other agents.
Benefit: Using atomics instead of exclusive access enhances system efficiency and concurrency.

Reliability, Availability, and Serviceability (RAS) Features:

Purpose: Added to support the Armv8 RAS specification, enhancing error detection and system debug capabilities.

Features: Includes Data Poisoning and DataCheck for indicating corrupted data, and the Trace Tag feature for profiling and debug purposes.

Advantages:

Enhanced Performance: CHI optimizes data transfer and access latency, improving overall system performance, crucial for applications demanding high throughput and responsiveness.
Scalability: Its support for various topologies and scalable designs allows for seamless integration of multiple components, ensuring efficient resource utilization and system expansion as needed.
Versatility: CHI's adaptability across different applications such as mobile, networking, automotive, and data centers showcases its broad utility, offering flexibility for diverse system requirements.
Reliability and Data Integrity: The protocol's ordering mechanisms, error detection features, and support for coherent transactions ensure data integrity and system reliability, reducing the risk of errors and inconsistencies.

Disadvantages:

Complexity: Implementing CHI and managing coherency introduces complexity to system design and verification processes, demanding additional time, expertise, and resources, which could potentially increase development costs.
Cost: The integration of CHI may incur additional expenses in terms of licensing fees, development tools, and hardware resources, potentially impacting the overall project budget, especially for smaller-scale designs.
Power Consumption: Maintaining cache coherency and supporting coherent transactions may lead to increased power consumption, particularly in larger-scale designs with numerous components, potentially impacting energy efficiency and battery life in mobile applications.
Compatibility Challenges: Ensuring interoperability between different components and peripherals may pose challenges, especially when integrating third-party IP cores or legacy components, potentially leading to compatibility issues and additional development efforts.