Coherent vs Non-Coherent Fabrics: Choosing the Right NoC for Your SoC

A SignatureIP technical white paper on partitioning on-die transport for modern multi-core SoCs

SignatureIP Engineering Team
Center of Excellence for Compute and I/O Subsystem IP

***Abstract*, The on-die transport choice in a modern SoC is no longer a single decision. It is a partitioning problem: which traffic needs hardware-enforced cache coherency, and which does not. Coherent fabrics built around AMBA CHI Rev E.b (Issue E) with a directory-based home node carry the traffic between CPU clusters, GPU compute, and AI/ML accelerators that must see the same view of shared memory. Non-coherent fabrics built around AXI4 (and AXI3, AXI4-Lite, AXI4-Stream, AHB, APB) carry I/O, peripheral, and legacy traffic that has no need for coherency. Picking a single fabric for everything wastes silicon, bandwidth, and power on traffic that does not need coherency, or forces software cache maintenance that does not scale on traffic that does. The pattern that current production SoCs converge on is a coherent fabric, a non-coherent fabric, a bridging cache between the two for I/O coherency, and bidirectional protocol bridges for migration and interop. SignatureIP delivers this as one integrated system: C-NOC as the CHI coherent fabric, NC-NOC as the non-coherent fabric, Proxy Cache as the bridging cache, AXI2CHI and CHI2AXI as the protocol bridges, ATC as the IOMMU-style translation cache, PCIe Controller and CXL Controller (with the pre-integrated CXL Subsystem) as the standardized off-die attach paths, AXI4-LLL as the link-layer IP for photonic and copper serial interconnect, Ethernet IP as a representative non-coherent master, and Inoculator as the cloud-based topology generator. Functional-safety support across the portfolio is on the SignatureIP roadmap.**

***Index Terms*, AMBA CHI, AXI4, network-on-chip, directory-based snoop filter, I/O coherency, proxy cache, IOMMU, PCIe 6.x, CXL 3.x, Flex Bus, UCIe, mixed-fabric SoC, heterogeneous SoC.**

I. EXECUTIVE SUMMARY

The on-die transport choice in a modern SoC is no longer a single decision. It is a partitioning problem: which traffic needs hardware-enforced cache coherency, and which does not. Coherent fabrics built around AMBA CHI Rev E.b (Issue E) with a directory-based home node carry the traffic between CPU clusters, GPU compute, and AI/ML accelerators that must see the same view of shared memory. Non-coherent fabrics built around AXI4 (and its AXI3, AXI4-Lite, AXI4-Stream, AHB, and APB variants) carry I/O, peripheral, and legacy traffic that has no need for coherency and would waste coherent bandwidth if forced through it.

The architectural lever is partitioning, with a defined bridging path between the two domains. Picking a single coherent fabric for everything wastes silicon, bandwidth, and power on traffic that does not need coherency. Picking a single non-coherent fabric forces software cache maintenance that does not scale and causes silent data corruption on missed maintenance calls. The pattern that current production SoCs converge on is a coherent fabric, a non-coherent fabric, a bridging cache between the two for I/O coherency, and bidirectional protocol bridges for migration and interop. SignatureIP delivers this as one integrated system: C-NOC as the CHI coherent fabric, NC-NOC as the non-coherent fabric, Proxy Cache as the bridging cache, AXI2CHI and CHI2AXI as the protocol bridges, ATC as the IOMMU-style translation cache, PCIe Controller and CXL Controller (with the pre-integrated CXL Subsystem) as the standardized off-die attach paths, AXI4-LLL as the link-layer IP for photonic and copper serial links, Ethernet IP as a representative non-coherent master, and Inoculator as the cloud-based topology generator that ties the two fabrics together at design time. Functional-safety support across the portfolio is on the SignatureIP roadmap.

II. WHY THE CHOICE MATTERS

Every multi-core SoC has caches, and every cache that holds writable data shared with another agent creates a coherency problem. The fabric that enforces coherency carries snoops, responses, and directory traffic in addition to the data payload. The fabric that does not enforce coherency carries only the data payload, plus the metadata each protocol needs to route it.

The two cost models look different in silicon. A coherent fabric pays directory storage at the home node, snoop bandwidth on the fabric, response bandwidth to the requester, and serialization points where the protocol enforces ordering. A non-coherent fabric pays none of these. Both fabrics pay for QoS, ordering within their own protocols, address interleaving, and power-domain control. The non-coherent fabric pays for these without the coherency overhead on top.

The choice is not academic. Snoop traffic grows as the square of agent count on broadcast-snooping fabrics, and even directory-based fabrics carry response and forward traffic that consumes link bandwidth. Memory bandwidth utilization on a representative server-class system saturates near 53 percent of theoretical peak under heavy coherency traffic, against roughly 82 percent without it [1]. Inter-core communication latency on the same system ranges from approximately 36 ns when cores share an L3 segment to approximately 137 ns when they sit in different sockets, a 3.8x topology penalty [1]. These costs are unavoidable on the coherent path. They are entirely avoidable on the non-coherent path.

The question for the SoC architect is not which fabric to use. It is how to partition traffic across both, and how to bridge between them where partial coherency is needed.

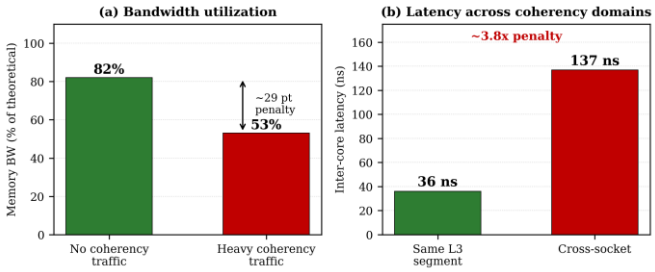

Fig. 1. Measured cost of coherency. (a) Memory bandwidth utilization drops by approximately 29 percentage points under heavy coherency traffic. (b) Inter-core latency rises by 3.8x when communication crosses a coherency domain.

III. TECHNICAL FOUNDATIONS

A. What a Coherent Fabric Does

A coherent fabric is responsible for ensuring that all initiators with cacheable views of memory see writes to a shared location in a consistent order, and that no stale copy survives past the point where the architecture says it should be gone. AMBA CHI Rev E.b is the production specification for SoC-scale coherency. It is a layered protocol, separating protocol, transport, and link layers, with four channels (REQ for requests, RSP for responses, DAT for data, SNP for snoops) carrying their own flow control. Coherency is enforced at the home node, designated HN-F for fully coherent and HN-I for the non-coherent equivalent. A directory at the home node tracks which agents currently hold each line. Snoops are sent only to the agents that hold a copy, which keeps the cost tractable even at hundreds of cores [2].

CHI adds three direct-transfer features that move the home node off the data path on the common case: Direct Memory Transfer (DMT) on a directory miss, Direct Cache Transfer (DCT) on a directory hit, and Direct Write Transfer (DWT) on the write-back path. The control path stays through the home node so the directory stays consistent. The data path is shortened to the minimum number of hops the topology allows [3].

B. What a Non-Coherent Fabric Does

A non-coherent fabric carries transactions that do not need cross-cache consistency. The dominant protocol is AXI4, with AXI3, AXI4-Lite, AXI4-Stream, AHB, and APB covering the slower and narrower variants used for peripherals, control, and legacy IP. A non-coherent fabric does not maintain a directory, does not issue snoops, and does not pay the area or power cost of either. It does carry first-order features that I/O traffic needs: configurable QoS classes, address interleaving across memory channels, traffic regulation to bound latency under burst load, performance counters for visibility, and Q-Channel and P-Channel handshakes for power-domain control.

The architectural value of separating the non-coherent fabric from the coherent fabric is that each runs on its own network with its own flow control and its own QoS budget. I/O traffic does not compete with coherent traffic for snoop or response bandwidth. Coherent traffic does not compete with I/O traffic for fabric link bandwidth on the coherent path.

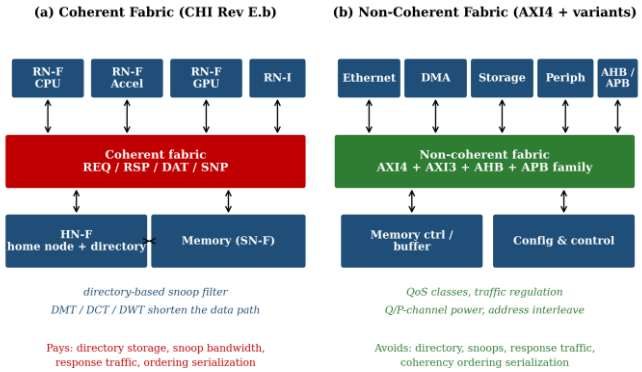

Fig. 2. Coherent versus non-coherent fabric structure. The coherent fabric pays directory, snoop, response, and serialization overhead the non-coherent fabric avoids entirely.

C. The Spectrum, Not the Dichotomy

Coherency is not binary. Fully coherent agents (RN-F in CHI terms) hold cacheable copies of memory, are snooped, and respond to snoops. I/O coherent agents (RN-I) can read coherent memory and may snoop into other caches, but they are not themselves snooped, on the assumption that they do not retain modified data over long horizons. Non-coherent agents bypass the coherent fabric entirely and rely on software to flush caches at the right boundaries.

The right level for a given block depends on how it uses memory. CPU clusters and tightly coupled accelerators need full coherency. DMA engines and network controllers benefit from I/O coherency: they touch coherent data occasionally but do not hold it. Boot code, configuration registers, and trace buffers are non-coherent and stay that way. Picking the wrong level wastes bandwidth, area, or correctness, depending on the direction of the error.

IV. THE COST OF EACH CHOICE

A. Cost of Running Coherency on Traffic That Does Not Need It

Routing I/O traffic through a coherent fabric pays the coherency tax on every transaction. Each I/O burst takes directory storage at the home node, snoop bandwidth on the fabric, and serialization at the home-node ordering point. None of that overhead delivers value for traffic that does not have a long-lived cached presence anywhere on the chip.

At sustained AI/ML or networking bandwidth, this overhead is not a rounding error. Tensor staging, network packet handling, and storage controllers all run at tens to hundreds of gigabytes per second of fabric throughput. Pushing that through a coherent fabric means scaling the coherent fabric to carry the sum of coherent and non-coherent demand, which scales coherency overhead proportionally [4].

B. Cost of Running Without Coherency on Traffic That Needs It

The opposite mistake is more dangerous. Software-managed coherency requires the application or driver to issue explicit cache maintenance operations (clean, invalidate, flush) and memory barriers at the right points in the program. Every shared-data interaction becomes a manual operation. Driver code accumulates defensive flushes, which cost performance, and missed flushes cause data corruption, which costs reputation. False sharing, where two agents write to different bytes within the same cache line, multiplies the maintenance traffic with no visible cause in source code [5].

At modern bandwidth and agent counts, software-managed coherency does not scale. The per-transaction software overhead is incompatible with the throughput required by AI accelerators, network controllers, and storage controllers. Missed maintenance is a silent failure mode.

C. The Cost of Getting the Partition Wrong

The partition between coherent and non-coherent fabrics is the architectural lever. Get it right, and the coherent fabric is sized for traffic that benefits from coherency, the non-coherent fabric is sized for traffic that does not, and the bridging cache absorbs the boundary cases. Get it wrong, and either the coherent fabric saturates under I/O load it should not be carrying, or the non-coherent agents are forced into software coherency they cannot meet.

SignatureIP Solution, NC-NOC and C-NOC. SignatureIP ships two separate fabrics so the partition is real, not nominal. C-NOC carries the CHI coherent traffic and is sized for the agent count and bandwidth that actually need coherency. NC-NOC carries AXI4, AXI3, AXI4-Lite, AXI4-Stream, AHB, and APB traffic on its own network, with its own QoS, address interleaving, traffic regulation, and Q-Channel/P-Channel power control. Coherent traffic never contends with I/O traffic for fabric link bandwidth, and I/O traffic never pays the coherency tax.

V. THE PARTIAL-COHERENCY CASE

Most non-CPU IP on a real SoC is not CHI native. DMA engines, network controllers, storage controllers, video and image engines, and most accelerator blocks expose AXI4 or AXI4-Stream. Forcing every such block to implement a full RN-F interface to participate in coherency is rarely justified, both because of the design effort and because the access patterns do not warrant full coherency.

The architectural answer is I/O coherency through a proxy cache. The proxy cache sits at the boundary between the non-coherent fabric and the coherent fabric. On the coherent side it is a CHI agent with directory state tracked by the home node. On the non-coherent side it presents AXI4. A non-coherent block reads coherent memory through the proxy cache. The block sees a coherent memory view without software cache maintenance. The fabric sees a CHI-compliant participant.

The proxy cache also absorbs repeated reads of shared data on behalf of the non-coherent block. A network controller that re-reads the same descriptor ring each pass, or a DMA engine that re-reads the same buffer header, pays the snoop cost only on the proxy cache's misses, not on every read.

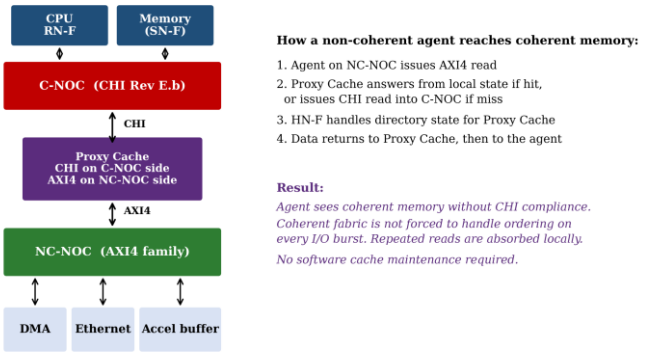

Fig. 3. I/O coherency through a Proxy Cache. The non-coherent agent reaches coherent memory through Proxy Cache without redesign, without CHI compliance, and without software cache maintenance.

SignatureIP Solution, Proxy Cache. Proxy Cache is the bridging cache between NC-NOC and C-NOC. It is a CHI agent on the C-NOC side, an AXI4 endpoint on the NC-NOC side, and a small local cache that absorbs repeated reads of shared lines. Non-coherent blocks reach coherent memory without redesign, and the coherent fabric is not forced to handle ordering for every I/O burst.

A second piece sits on the same boundary: address translation. DMA-capable masters on the non-coherent fabric need IOMMU-style translation in any system that runs under virtualization, isolates guests, or supports a process-level memory model on the I/O side. Routing every DMA access through a centralized IOMMU adds latency that the rest of the coherent path was designed to remove.

SignatureIP Solution, ATC. ATC is the IOMMU-style address translation cache that holds recently used translations close to the requester. Translation misses still go to the IOMMU page tables; hits are answered locally. DMA latency stays tolerable in a virtualized system without giving up the isolation guarantees that justify virtualization in the first place.

VI. HETEROGENEOUS SOCS IN PRACTICE

A. The Mixed-Fabric Pattern

Heterogeneous SoCs are now the dominant pattern in compute, networking, automotive, and AI. CPU clusters, GPU compute, AI/ML accelerators, video and image engines, DMA engines, storage controllers, network controllers, and a long tail of peripherals all coexist on the same die. Industry coverage in 2024 to 2025 treats the mixed-fabric pattern as the default, not the exception: complex chips need coherent and non-coherent sub-NoCs to ensure efficient data paths, with the correct hierarchy between them as a first-order architectural concern [6].

B. AI Workloads as the Stress Test

AI inference and training stress every mechanism in the coherent fabric harder than the original CPU-cluster workloads it was designed for. Tensor data crosses cluster boundaries between CPU pre-processing, accelerator compute, and DMA staging. Weights are loaded once and read many times by multiple compute units. Activations are written by one engine and consumed by another within microseconds. These access patterns produce sustained coherent traffic at bandwidth levels that older fabrics meet by saturating [4].

C. Automotive, ADAS, and Autonomous Driving

ADAS and autonomous driving systems sit in the same category. Sensor fusion, perception, planning, and actuation pipelines run across CPU clusters, AI accelerators, and high-bandwidth I/O on the same SoC, with deterministic worst-case latency budgets measured in microseconds and a safety classification that does not tolerate silent data corruption. Cache coherency is not a performance optimization in these systems; it is the mechanism that lets the perception stack and the control stack agree on what the vehicle just saw, with bounded latency, every cycle.

D. Migration and Interop

A green-field SoC can design every initiator and subordinate around the right fabric from the start. A practical product line cannot. There is always an existing portfolio of AXI subsystems, memory controllers, debug logic, and legacy IP that has to coexist with the new architecture. A clean migration needs bidirectional protocol bridges that handle the channel-to-packet semantic gap, not a forklift redesign.

SignatureIP Solution, AXI2CHI and CHI2AXI. AXI2CHI translates AXI4 transactions into CHI Rev E.b transactions, so AXI-native initiators can reach C-NOC without redesign. CHI2AXI translates CHI transactions to AXI4, so a CHI-native fabric can reach existing AXI subordinates (memory controllers, debug logic, legacy IP) without forcing them to migrate.

SignatureIP Solution, Ethernet IP. Ethernet IP is the worked example of the mixed-fabric architecture in production. The Ethernet block runs at packet-rate throughput. It needs to read coherent memory for transmit data and write coherent memory for received data. Software cache maintenance at that rate is not viable. Ethernet IP sits on NC-NOC, reaches coherent memory through Proxy Cache, and demonstrates the NC-NOC plus Proxy Cache plus C-NOC pattern end to end.

VII. THE OFF-DIE EXTENSION

Fabric choice extends off-die as soon as the SoC commits to a chiplet, multi-die, or off-package attach topology. The two fabrics on-die need a defined extension path to memory and accelerators that sit on a different die, package, or socket.

The non-coherent off-die transport is PCIe. PCIe 6.x runs at 64 GT/s per lane on PAM4 with link widths from x1 to x16, uses 256-byte FLIT framing with FEC at Gen6, and 128b/130b encoding on Gen3 through Gen5. Reliability comes from FEC, CRC, ACK/NAK replay, and Advanced Error Reporting. PCIe is the production substrate for high-bandwidth I/O, storage, and accelerator attach, and it is the physical and link layer that CXL builds on.

The coherent off-die transport is CXL. CXL 3.x layered on the Flex Bus carries CXL.io (PCIe-equivalent), CXL.cache (coherent device caches), and CXL.mem (memory expansion), with device Types 1, 2, and 3. The CHI fabric on-die extends onto the Flex Bus through a bridge that appears to the coherent fabric as both an RN-F (when the off-die device caches host memory) and an SN-F (when host transactions address device-attached memory).

UCIe 2.0, ratified in August 2024, is the die-to-die transport standard on which chiplet-style coherent extensions ride [8]. C-NOC today supports a UCIe interface for die-to-die communication, so the physical and link path for chiplet integration is in the current product, not a future capability. The remaining piece, coherency extensions across the die boundary (the directory distribution and snoop routing that turn UCIe links into a coherent multi-die fabric), is on the SignatureIP roadmap as a future release. The architectural foundation in the current monolithic C-NOC, including directory placement and home-node distribution, is being designed against that path. Customers committing to C-NOC today are positioned to extend into chiplet topologies without reworking the coherency protocol, and without adding the die-to-die transport, because the transport is already there.

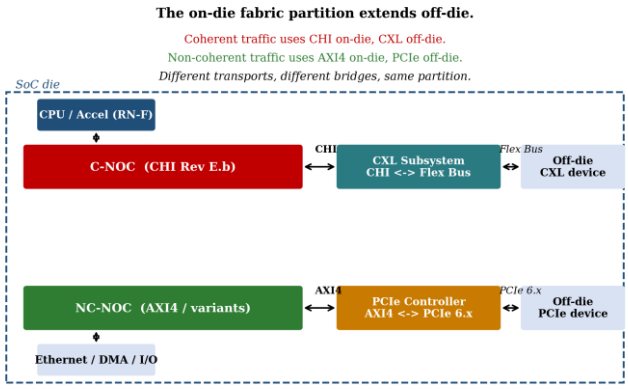

Fig. 4. The fabric partition extends off-die. Coherent traffic uses CHI on-die and CXL off-die. Non-coherent traffic uses AXI4 on-die and PCIe off-die.

SignatureIP Solution, PCIe Controller, CXL Controller, and CXL Subsystem. PCIe Controller is SignatureIP's PCIe 6.x controller covering the full Transaction Layer, Data Link Layer, and MAC, with FEC, CRC, ACK/NAK replay, AER, 512-bit AXI4 application interface, Root Complex or Endpoint configuration, and up to four PFs with 64 VFs per PF. CXL Controller covers CXL 3.x across CXL.io, CXL.cache, and CXL.mem; it is the bridge between a CHI-coherent fabric and the Flex Bus, appearing to C-NOC as both RN-F and SN-F. CXL Subsystem is the pre-integrated drop-in subsystem (sig_cxl_system_top).

PCIe and CXL cover the standardized off-die paths. A fourth, narrower path is photonic and copper serial interconnect for disaggregated memory and chiplet attach, where the latency budget is tight enough that the retransmission overhead of a packetized retry protocol is unacceptable. The link layer for these paths is its own IP class: complete PHY and PCS datapath behind an AXI4-Stream host interface, with forward error correction integrated directly on the data path so that single-byte errors are corrected without retransmission and the latency budget never absorbs a retry.

SignatureIP Solution, AXI4-LLL. AXI4-LLL is the SignatureIP link-layer IP for photonic and copper serial interconnect. It exposes a 256-bit AXI4-Stream host interface on the application side and drives PIPE/PMA SerDes (Gen1 through Gen5, 2.5 to 64 GT/s) on the link side, with the same RTL covering optical or copper media. An integrated Reed-Solomon GF(2^8) FEC engine corrects one byte and detects two bytes per 256-byte flit, which removes retransmission from the critical path and is claimed to cut effective end-to-end latency by up to 37 percent in the validated configuration. The full LTSSM, 128b/130b encoding, lane deskew, elasticity FIFO, and EIEOS detection ride on a PCIe-compatible PIPE 5.x interface; lane count is parameterized for x1 through x16. P0, P0s, P1, and P2 power states are PIPE-compliant. AXI4-LLL has been validated on AMD VP1902 FPGA hardware through the Siemens proFPGA CS infrastructure. Target use cases are XPU and AI accelerator chiplet interconnect, disaggregated HBM and DRAM expansion over optical links, photonic interconnect SoCs, and HPC fabric data planes. AXI4-LLL sits on the non-coherent side of the fabric partition alongside PCIe Controller, and is the path for cases where PCIe is not the right answer because the media is photonic, the link is custom, or the latency budget cannot tolerate retransmission.

VIII. FUNCTIONAL SAFETY AND COMPLIANCE

Both fabrics carry safety-critical traffic in automotive, industrial, medical, and aerospace SoCs. The coherent path carries every snoop, response, and forward, and a fault anywhere on that path can propagate as wrong-data into the application. For ADAS and autonomous driving, the same fault can propagate into perception or actuation output with safety consequences. The non-coherent path carries I/O streams that often touch safety-critical sensor data on ingress and actuator commands on egress.

A. Protecting Data at Rest and Data in Motion

Functional safety has to cover both data at rest and data in motion. Data at rest lives in on-chip SRAMs (cache lines, directory entries, buffers, register files) and in off-chip DRAM. These structures are the primary target of soft errors and need ECC at the storage boundary. Data in motion lives on the buses, bridges, and pipeline stages of the interconnect itself, where a transient flip can corrupt a transaction in transit. Protecting one without the other leaves a hole in the safety case.

B. ECC, Parity, Lockstep, and the SECDED to DECTED Shift

The protection mechanisms in a safety-grade fabric are well understood. SECDED ECC on data buses catches single-bit errors and detects double-bit errors. Parity on address and control buses catches single-bit cases on narrower fields. End-to-end protection across pipeline stages keeps detection coverage continuous from initiator to target. Lockstep on critical state machines provides diagnostic coverage where ECC and parity cannot reach.

For applications that demand very high reliability or run in harsh environments, such as space systems and avionics, DECTED (double-error-correct, triple-error-detect) is replacing SECDED as the data-protection scheme of choice. DECTED tolerates the higher rate of multi-bit upsets caused by radiation and aging.

The harder problem above the protection bits is freedom from interference (FFI): demonstrating that a fault in one initiator's traffic cannot propagate into another initiator's domain. FMEDA analysis, fault-injection campaigns, and the safety case documentation are required deliverables for ISO 26262 [9] ASIL-B through ASIL-D, and adding them after the fact is rarely viable.

C. SignatureIP Roadmap

Functional safety is on the SignatureIP roadmap, not in the current product release. C-NOC, NC-NOC, Proxy Cache, ATC, AXI2CHI, CHI2AXI, Ethernet IP, PCIe Controller, CXL Controller, CXL Subsystem, and AXI4-LLL today are production-grade for compute, networking, and storage SoCs, but they do not ship with an ISO 26262 certification package, FMEDA, or formal safety case. ASIL-B through ASIL-D readiness across the portfolio is targeted as a future release.

IX. THE SIGNATUREIP ARCHITECTURE

The architecture described in the preceding sections is not abstract. It is the architecture SignatureIP ships, summarized in Fig. 5, and each block in the portfolio answers a specific piece of the fabric-partition problem.

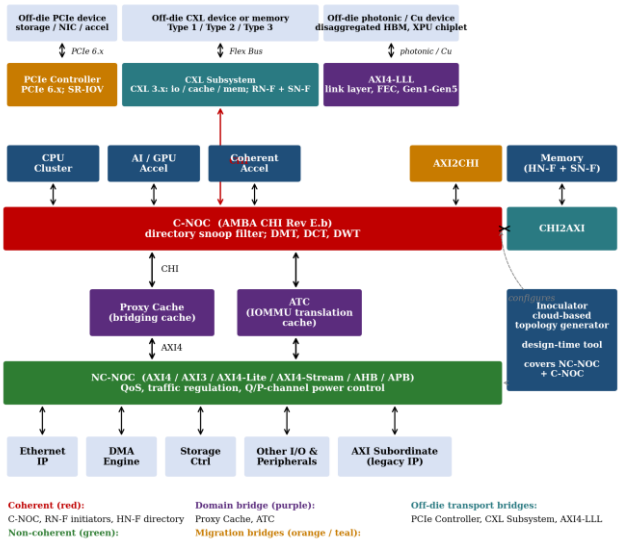

Fig. 5. SignatureIP portfolio in one SoC context. C-NOC carries the CHI coherent traffic; NC-NOC carries AXI4 non-coherent traffic; Proxy Cache and ATC bridge the two domains; AXI2CHI and CHI2AXI are the bidirectional protocol bridges; PCIe Controller and CXL Subsystem extend the architecture off-die; Inoculator generates the topology at design time.

C-NOC is the coherent fabric. It implements AMBA CHI Rev E.b end to end, with directory-based snoop filtering at the home node and DMT, DCT, and DWT enabled on the data path. NC-NOC is the non-coherent fabric, carrying AXI4 and the AXI3, AXI4-Lite, AXI4-Stream, AHB, and APB variants at high throughput.

Proxy Cache bridges the two domains. ATC sits alongside, providing IOMMU-style address translation for DMA-capable masters on the non-coherent fabric. AXI2CHI and CHI2AXI are the bidirectional protocol bridges. PCIe Controller, CXL Controller, CXL Subsystem, and AXI4-LLL are the off-die extension. PCIe Controller is the standardized non-coherent off-die transport. CXL Controller (with the pre-integrated CXL Subsystem) is the coherent off-die attach. AXI4-LLL is the link-layer IP for photonic and copper serial links where retransmission overhead would break the latency budget. Ethernet IP is the representative non-coherent master.

Inoculator is the cloud-based topology generation tool. Current scope covers NC-NOC and C-NOC; coverage for the rest of the portfolio is on the Inoculator roadmap and will be added incrementally. Inoculator is a productivity and correctness tool. It is not part of the ECC, parity, or lockstep path, and it is not a functional safety, fault injection, or error-protection block.

X. CONCLUSION AND DECISION FRAMEWORK

The on-die transport choice for a modern SoC is a partition problem, not a single decision. The architect's job is to route the right traffic through the right fabric, and to bridge between the two where partial coherency is needed.

The criteria for the coherent fabric are concrete: traffic from initiators that hold cacheable copies of shared memory, where the cost of software-managed coherency would be unacceptable. CPU clusters, tightly coupled accelerators, GPU compute, and AI/ML accelerators that share working sets all belong here. The implementation criteria are AMBA CHI Rev E.b conformance with DMT, DCT, and DWT enabled, a directory-based home node sized for the agent count, and channel separation.

The criteria for the non-coherent fabric are equally concrete: traffic from initiators that do not need a cached, coherent view of memory. DMA engines, network controllers, storage controllers, video and image engines, and peripherals all belong here. The implementation criteria are AXI4 (with AXI3, AXI4-Lite, AXI4-Stream, AHB, APB support for the variants), configurable QoS, address interleaving, traffic regulation, and Q-Channel and P-Channel power-domain control.

The off-die extension follows the same partition: PCIe for standardized non-coherent off-die transport, CXL for coherent off-die attach, a UCIe die-to-die interface on C-NOC for chiplet integration, and AXI4-LLL for photonic and copper serial links where the latency budget cannot tolerate retransmission. The die-to-die transport is in C-NOC today; coherency extensions that turn UCIe links into a multi-die coherent fabric are on the SignatureIP roadmap. Functional safety across the portfolio is also on the SignatureIP roadmap.

SignatureIP's portfolio is designed to meet these criteria today: C-NOC for the coherent traffic, NC-NOC for the non-coherent traffic, Proxy Cache and ATC at the boundary, AXI2CHI and CHI2AXI for migration and interop, PCIe Controller and CXL Controller (with the pre-integrated CXL Subsystem) for standardized off-die attach, AXI4-LLL for photonic and copper serial links, Ethernet IP as the worked-example non-coherent master, and Inoculator as the cloud-based topology generator covering NC-NOC and C-NOC today, with the rest of the portfolio on its roadmap.

. REFERENCES

[1] R. K. Somarouthu, "Demystifying Cache Coherency in Modern Multiprocessor Systems," *European Journal of Computer Science and Information Technology*, vol. 13, no. 37, 2025.

[2] Arm Limited, *AMBA CHI Architecture Specification* (IHI0050), latest revision. [Online]. Available: https://developer.arm.com/documentation/ihi0050/latest/

[3] Arm Limited, *Data Memory Transfer, Direct Cache Transfer, PrefetchTgt, and Direct Write Transfer*, Arm Developer Documentation (102407). [Online]. Available: https://developer.arm.com/documentation/102407/latest/

[4] Semiconductor Engineering, "NoC Coherency Challenges Balloon With AI SoCs And Chiplets." [Online]. Available: https://semiengineering.com/noc-coherency-challenges-balloon-with-ai-socs-and-chiplets/

[5] T. Liu et al., "Leveraging Cache Coherence to Detect and Repair False Sharing On-the-fly," in *Proc. IEEE/ACM MICRO*, 2024. [Online]. Available: https://www.cse.iitk.ac.in/users/swarnendu/files/papers/micro24.pdf

[6] Semiconductor Engineering, "Confusion Grows With More Interconnect Options And Tradeoffs." [Online]. Available: https://semiengineering.com/confusion-grows-with-more-interconnect-options-and-tradeoffs/

[7] Arteris, *NoC Interconnect Fundamentals: Coherent and Non-Coherent Fabrics for Modern SoCs*. [Online]. Available: https://www.arteris.com/learn/network-on-chip-technology/

[8] UCIe Consortium, *Universal Chiplet Interconnect Express (UCIe) 2.0 Specification*, August 2024. [Online]. Available: https://www.uciexpress.org/2-0-spec-download

[9] ISO, *ISO 26262-1:2018 Road vehicles: Functional safety*. [Online]. Available: https://www.iso.org/standard/68383.html