P8700 Multiprocessing System Datasheet   January 14, 2025

The MIPS™ P8700 Multiprocessing System (MPS) is a member of the MIPS 8th Generation Architecture family. In addition to full compatibility to RISC-V technology by supporting the RV64GCZba_Zbb Base ISA and Standard Extensions, the MIPS 8th generation architecture also brings MIPS-extended features already proven in MIPS ISA-based cores to enhance the performance and functions on running applications.

The MIPS™ P8700 Multiprocessing System (MPS) provides a highly scalable foundation for designing and building the types of cores required to handle the compute-intensive tasks in emerging safety-critical systems, such as:

- Automotive (ADAS, AV, IVI)
- Machine learning on device
- Networking (5G, WiFi-7, DPU, Sensors, Connections)
- Mid-tier data center
- High performance embedded applications

The P8700 MPS scales to 64 heterogeneous clusters of out-of-order, multi-threaded multi-core MIPS CPUs, and through the MIPS Coherence Manager with AMBA® ACE interface, enables integration with heterogeneous CPU clusters and other accelerators at the system-on-chip (SoC) level.

As such, the P8700 MPS provides best in class multi-core performance for use in sys- tem-on-chip (SoC) applications. The P8700 MPS combines a deep pipeline with multi- issue Out-Of Order-execution and multi-threading to deliver outstanding computational throughput.

The P8700 Coherence Manager maintains Level 2 (L2) cache and system level coherency between all cores, main memory, and I/O devices. The P8700 MPS is a configurable and a synthesizable solution. The collection of clusters of cores can be configured with a variable number of cores, I/O coherent interfaces, and L2 cache size. Each of the cores can be configured with Level 1 (L1) cache sizes, and number of harts.

Each P8700 core implements the RISC-V RV64GCZba_Zbb[1] Instruction Set Architecture (ISA) with full hardware multithreading.

Highlights of the P8700 MPS include:

- Multi-Cluster support
- PDtrace support
- Coherence Manager (CM) with integrated L2-cache:
    - Up to 8 I/O Coherence Units (total number of cores + IOCUs must not exceed 8)
   - Cluster Power Controller (CPC)

- G = IMAFD: Details provided in the Features section below. Zba and Zbb are bit manipulation extensions.

- Interrupt Controller
- Global Configuration Registers (GCR)
- Multiprocessor debug via in-system Debug Unit (DBU)
- Trace Funnel (TRF)

Figure 1 shows a block diagram of a single cluster P8700 Multiprocessing System (MPS).

Figure 1. Block Diagram of Single Cluster P8700 Multiprocessing System
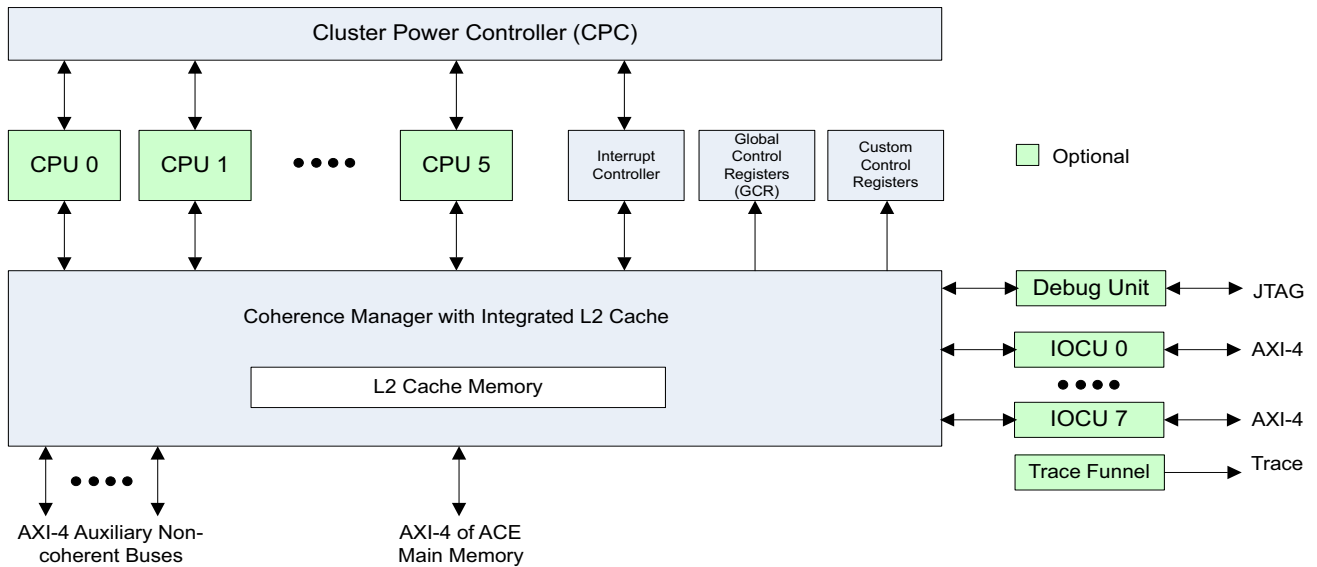
The P8700 can have a total of 8 CPU's and IOCU's in any combination. As such, in the above figure, note that all CPU's and all IOCU's are shown as optional. This is because a system can have 6 CPU's and 2 IOCU's, or 8 IOCU's and no CPU's. This is a build-time configuration option. For example, if a system is configured with 4 CPU's, then it can have up to 4 IOCU's for a total of eight, but NOT more than four IOCU's.

P8700 Features

The P8700 MPS implements the current RISC-V RV64GCZba_Zbb architecture, including new CPU and system-level features designed for performance, power, and area form factors. The MPS flexibility and features are well suited for a broad range of markets and applications, such as automotive, machine learning on device, networking, mid-tier data center, and high performance embedded applications.

P8700 Architecture

The P8700 Multiprocessing System has four key architectural features as described in the following subsections.

- RISC-V RV64GCZba_Zbb architecture (Base ISA and Standard Extensions)
- MIPS Multithreading
- Hybrid Debug
- RISC-V Privileged Architecture
- Functional Safety

The P8700 core is configured to support the RV64GCZba_Zbb (G = IMAFD) Standard ISA. It includes the RV64I base ISA, Multiply (M), Atomic (A), Single-Precision Floating Point (F), Double-Precision Floating Point (D), Compressed (C) RISC-V extensions, as well as the as well as the bit-manipulation extensions (Zba) and (Zbb).

The P8700 provides memory management through on-chip configuration registers and enables real-time operating systems and application code to be implemented once and then reused.

MIPS Out-of-Order Multithreading

CPU performance depends on minimizing the latency to the system memory. Even with a cache hierarchy, the CPU still stalls while waiting for data. To avoid this scenario, MIPS out-of-order multithreading provides significant performance improvements by running additional instructions concurrently.

This hardware out-of-order multithreading enables execution of multiple instructions from multiple threads (harts) every clock cycle, providing higher utilization and CPU efficiency. In this way, out-of-order multi-threading is a more area efficient alternative to the use of additional cores and offers a typical 60% performance boost for the execution of two harts simultaneously instead of sequentially.

Hybrid Debug

The P8700 offers proven MIPS EJTAG with RISC-V Trace and GDB support for Multi-Core/Cluster Debug.

P8700 Privileged Architecture

The RISC-V privileged architecture covers all aspects of RISC-V systems beyond the unprivileged ISA, including privileged instructions as well as additional functionality required for running operating systems and attaching external devices.

The P8700 implements the RISC-V compliant Privileged Architecture, as well as more Custom CSRs, for enhancement on features and performance. The P8700 Privileged Architecture includes:

- Privileged operating modes (Supervisor-mode, Machine-mode, Debug-mode)
- M-mode: All Machine-level CSRs and Privileged Instructions
- S-mode: All Supervisor-level CSRs and Supervisor Instructions
- A set of User-Defined Instructions and CSRs which have been proven in existing MIPS CPUs.
- D-mode: All Debug/Trace CSRs

Functional Safety

The P8700 IP is designed to also supports the ASIL-B(D) functional safety standard. In so doing, the P8700 cluster includes the following fault detection features:

- Fault bus to report detected faults to external fault handling logic
- End-to-end parity protection on address and data buses
- Parity protection of software visible registers in the GCR, Interrupt Controller, and CPC blocks
- Programmable transaction time out detection on memory requests originating from a CPU or IOCU
- SRAM error detection and correction
- Protocol error detection on IOCU and REGTC AXI slave interfaces
- AXI/ACE interface parity protection of address and data compatible with third-party interconnects

System-level Features

- Up to six coherent RISC-V RV64GCZba_Zbb CPU cores.

- Multi-Cluster support: Cluster composed of up to 0 - 6 CPUs and 0 - 8 IOCUs (sum being no more than 8 agents) and a Level 2 cache connection to a coherent interconnect. Support for up to 64 clusters.

- Integrated L2 cache controller supporting a 8-way and 16-way set-associativity:

  - Inclusive of the L1 data caches

  - 256 KB to 8 MB cache sizes

  - Single bit correction and double bit detection

- CPC to shut down idle cores for power efficiency.

- Cache-to-cache data transfers.

- Out-of-order data return.

- Hardware L2 cache prefetch controller significantly improves performance of workloads such as memory to memory data transfer/copy (memcpy).

- Independent clock ratios on core, memory, and IOCU ports.

- SoC system interface supports AXI-4 (Advanced eXtensible Interface rev. 4, also known as AMBA 4 AXI) or ACE (AXI Coherency Extensions) protocol with 48-bit address and 256-bit data paths. This interface can be configured to support up to 96 outstanding requests.

- High bandwidth 128-bit data paths between each core and the Coherence Manager.

- Software controlled core level and cluster level power management.

- Debug port supporting multi-core debug (JTAG/APB).

- Program and Data trace (PDtrace) mechanism to debug software.

CPU Core-Level Features

- Full 64-bit Instruction Set Architecture with Compressed Instructions via RISC-V RV64GCZba_Zbb

- 48-bit virtual and physical addresses

- 8-wide instruction fetch, 4-wide decode, rename and graduation, 7-wide issue

- Hardware out-of-order multithreading

- L1 caches with Error Correction Code (ECC) protection

- L2 cache support — Implemented as shared L2 in the Coherence Manager

- Programmable Memory Management Unit with large first-level ITLB/DTLB backed by fast on-core second-level variable page size TLB (VTLB) and fixed page size TLB (FTLB)

  - Shared FTLB across all hardware threads (harts) in a CPU

  - MIPS DVM support through Global Instruction cache and TLB invalidation

- Load and store bonding support

- Unaligned load / store support in hardware
- Program and Data Trace (PDtrace) support for Instructions and Data (Virtual Addresses and Data Values)

P8700 CPU Core Features

Figure 2 shows a block diagram of a single P8700 core. The logic blocks in this diagram are described in the following sections.

Figure 2. P8700 Core Block Diagram
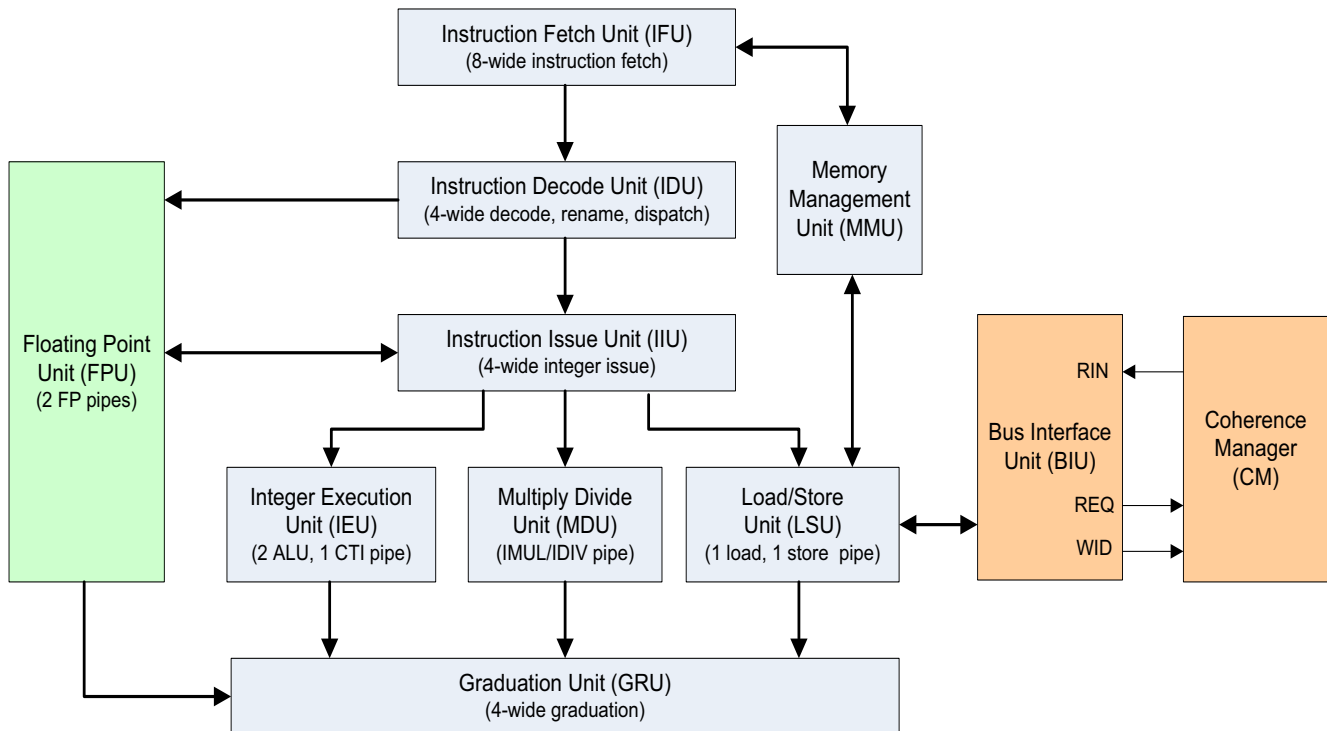
For more information on the P8700 core in a multiprocessing environment, refer to "Multiprocessing System Features" on page 13.

Instruction Fetch Unit (IFU)

The Instruction Fetch Unit (IFU) fetches instructions from the L1 instruction cache and supplies them to the Instruction Decode Unit (IDU). The IFU can fetch up to eight instructions at a time from the L1 cache and fill the instruction buffers, which decouple the instruction fetch unit from the issue and execution of the instructions.

The Fetch Director Logic is implemented to choose which hart to fetch next. There is an Instruction Buffer and one Return Prediction Stack (RPS) per hart.

Branch Prediction

The IFU employs sophisticated branch prediction that anticipates the branch direction to improve performance and efficiency. The prediction is based on a TAgged GEometric length (TAGE) predictor. The predictor adapts to the program by self learning.

The TAGE predictor is shared by all harts, and the global and local history are stored per hart.

Jump Prediction

The IFU has a hardware-based Jump Register Cache (JRC) and Return Prediction Stack (RPS) to predict jump target addresses. This results in faster throughput during subroutine calls and returns.

Level 1 Instruction Cache

The P8700 L1 instruction cache is configurable as 32 KB or 64 KB in size and is organized as 4-way set-associative. The instruction cache is virtually indexed and physically tagged to allow data accesses and virtual-to-physical address translation to occur in parallel. This cache is shared by all harts.

This cache is used to fetch eight instructions per cycle. To conserve power, a way-prediction mechanism enables only the expected way. The cache is protected by single- and double-bit error detection logic.

Each cache line holds 64 bytes of instructions and the coherency of the cache is maintained by software with hardware assistance.

Instruction Decode and Issue

The fetch unit delivers up to four instructions per cycle to the Instruction Decode Unit (IDU). The Instruction Issue Unit (IIU) keeps these instructions in a deep instruction buffer.

Up to seven instructions may be issued for execution during a clock cycle. The instructions can be issued from the same hart or from different harts. A round-robin priority scheme is used to arbitrate among harts.

Instructions can be concurrently issued to any three of the following EXU functional units:

- 2 Integer Arithmetic and Logical Units (ALUs)
- 1 Multiply / Divide Unit
- 1 Branch Unit
- 1 Load Store Unit
- 1 Short Floating Point Pipe
- 1 Long Floating Point Pipe

Source Operand Read and Bypass

The execution units can simultaneously read source operands from the Architectural Register File (ARF) or Working Register File (WRF) for each of the instructions (regardless of hart context). In addition, the execution unit implements a fully symmetric operand bypass network to bypass a result from a preceding execution stage.

![MIPS logo](a GlobalFoundries company)

Integer Execution Unit

The Integer Execution Unit (IEU) has two complete ALUs that perform single-cycle operations including add, subtract, shifts, rotates, bit-wise logical, and several other operations. The IEU had one branch resolution pipe.

The IEU also contains a dedicated 64x64 integer multiplier and radix 4 SRT divider to speed up compute intensive applications.

Floating Point Pipelines

The Floating Point Unit (FPU) implements two separate pipelines (1 short, 1 long) to execute floating point instructions. These two pipelines allow the execution of simple floating point instructions to bypass and execute in parallel with less frequently used complex and iterative instructions. One pipeline executes logical ops, integer adds, and FP compares and stores. The other pipeline executes FP adds, FP multiplies, and FP divides.

The FPU contains thirty-two 64-bit vector registers. Single-precision floating-point instructions use the lower 32 bits of the 64-bit register. Double-precision floating point instructions use all 64 bits of the register.

All floating point denormalized input operands and results are fully supported in hardware.

Result Collection and Graduation

The Graduation Unit (GRU) collects all results from single-cycle, fixed-latency, and variable-latency instructions and pairs them up with associated completion status (such as exceptions and interrupts), and commits the results into the Architectural Register File (ARF). This committing of final results is called the *graduation* of the instruction. Up to four instructions are graduated per clock per hart.

Load Store Unit (LSU)

The Load Store Unit (LSU) moves data between the core and the system memory. It also maintains an L1 data cache to accelerate access to commonly used data by the core. The LSU accepts a single operation per cycle and maintains several buffers to keep the data moving between the Integer Execution Unit (IEU) and L1 cache, and between the L1 cache and the Bus Interface Unit (BIU) at an optimal rate.

The LSU supports non-blocking loads in multi-threaded configurations. It offers a 128-bit data path to the caches to support load/store bonding.

Level 1 Data Cache

The P8700 L1 data cache is configurable as 32 KB or 64 KB in size and is organized as 4-way set-associative. The L1 Data cache is virtually indexed and physically tagged, but contains logic to correct virtual aliasing.

The L1 data cache is capable of fetching data on both aligned and unaligned memory accesses. In addition, it can combine multiple loads and stores into a single operation using a feature called "instruction bonding" to maximize memory bandwidth.

To conserve power, a way-prediction mechanism enables only the expected way. The cache is protected by single-bit error correction and double-bit error detection logic.

Each cache line holds 64 bytes of data as well as the associated tag and replacement information.

Store and Write Buffer

The LSU contains store buffers that decouple the main pipeline from the memory sub-system, allowing the LSU to efficiently schedule cache writes and coherence operations while the main pipeline continues to execute subsequent instructions. After a store instruction graduates in the main pipeline, the LSU takes control and forwards the store data from the store buffer to subsequent load instructions until the data is committed to the cache or main memory.

The store buffers can merge multiple cacheable stores into a single larger write operation, which can take advantage of the 512-bit cache write data-path. This store buffer improves performance by avoiding contention at the cache RAM ports and saves power by reducing the number of RAM accesses.

The store buffer also merges multiple uncached-accelerated stores into a single burst-write transaction, to increase the efficiency of the bus and avoid stalling the main pipeline. Gathering of uncached accelerated stores can start on any arbitrary address and can be combined in any order within a 64-byte aligned block of memory.

Memory Management Unit (MMU)

The Memory Management Unit (MMU) translates virtual addresses to physical addresses and provides attribute information for different segments of memory. The P8700 MMU contains the following Translation Lookaside Buffer (TLB) structures:

- Instruction TLB (ITLB)
- Data TLB (DTLB)
- Variable Page Size Translation Lookaside Buffer (VTLB) per hart
- Fixed Page Size Translation Lookaside Buffer (FTLB) per core

Instruction and Data TLB (ITLB and DTLB)

The ITLB and DTLB (micro TLBs) are fully associative. The micro TLBs are used by the IFU and LSU to perform high speed virtual to physical memory address translation for instruction fetch and data movements respectively.

The ITLB is implemented in the IFU with support for 4 KB or 64 KB page sizes per entry. The DTLB is implemented in the LSU with support for 4 KB or 64 KB page sizes per entry. The micro TLB arrays are shared between harts.

MIPS
a GlobalFoundries company

The micro TLBs are managed completely by hardware and are transparent to the software. The micro TLBs are backed up by larger VTLB and FTLB structures. If a virtual address cannot be translated by the micro TLB, the VTLB / FTLB attempts to translate the address in the following clock cycle or when available. If successful, the translation information is copied into the appropriate micro TLB for future use.

Variable Page Size TLB (VTLB)

The VTLB is a fully associative translation lookaside buffer with 32 dual-entries per hart that can map variable page sizes such as 4 KB, 64 KB, 2 MB, 1 GB, and 512 GB.

Fixed Page Size TLB (FTLB)

The FTLB contains 512 dual entries organized as 128 sets and 4-way set-associative. The FTLB page size can be configured at 4 KB or 64 KB.

Bus Interface (BIU)

The BIU interfaces the instruction and data caches with the CM. This interface implements MIPS Coherence Protocol (MCP) and has three channels that support 128-bit data transfers. The transaction size can vary from 1 byte to 16 bytes for single uncached access or the full 64 bytes for a cache line. The BIU supports full memory coherency, including interventions.

Interrupt Handling

Each P8700 core supports six hardware interrupts including a timer interrupt and a performance counter interrupt. In addition, it support two software interrupts. These interrupts can be used in any of three interrupt modes, as defined by the RISC-V Architecture:

- *Interrupt compatibility* mode.
- *Vectored Interrupt (VI)* mode adds the ability to prioritize and vector interrupts to a handler dedicated to that interrupt.
- *External Interrupt Controller (EIC)* mode provides support for an external interrupt controller that handles prioritization and vectoring of interrupts.

Operating Modes

The P8700 core supports four modes of operation:

- User mode - is used for running application programs.
- Supervisor mode – is used for executing the OS.
- Machine mode – has the highest privileges and is the only mandatory privilege level for a RISC-V hardware platform.
- Debug mode is used during system bring-up and software development. Refer to Section ""Core Debug Support" on page 12" for more information on debug mode.

P8700 Core Power Management

The P8700 core offers several power-management features. It supports low-power design, such as active power management and power-down modes of operation. The P8700 core is a static design that supports slowing or halting the clocks to reduce system power consumption during idle periods.

Instruction-Controlled Power Management

The Instruction Controlled power-down mode is invoked through execution of the WFI instruction.

The WFI instruction puts the processor in a quiescent mode where no instructions are running. When the WFI instruction is seen by the Instruction Fetch Unit (IFU), subsequent instruction fetches are stopped. However, the internal timer and some of the input pins continue to run. Any interrupt, NMI, or reset condition causes the CPU to exit this mode and resume normal operation.

Core Debug Support

The P8700 core includes a debug block available for use in software debugging. For this purpose, in addition to standard user, supervisor, and machine modes of operation, the P8700 core provides a Debug mode.

The P8700 provides hybrid debug capability of MIPS Debug and the RISC-V architecture by offering proven MIPS EJTAG with RISC-V trace and GDB support for multi-core / cluster debug.

Debug mode is entered when a debug exception occurs and continues until a debug exception return instruction is executed or the CPU is reset. In debug mode, the P8700 operates as if it were in M-mode. Each hart in the P8700 may enter debug mode independently of the other. The P8700 implements the MIPS PDTrace specification (for enabling the tracing of function calls/returns) with only minor changes.

Multiprocessing System Features

The P8700 Multiprocessing System (MPS) provides multi-cluster support where each cluster consists of up to six P8700 cores, a Coherence Manager (CM) with integrated L2 cache, up to eight IOCUs, a cluster power controller (CPC), Interrupt Controller that conforms to the Advanced Interrupt Architecture (AIA) standard, debug unit (DBU), and global configuration registers (GCR). The CM maintains coherence with the cores' L1 caches by implementing a directory-based coherence protocol that enables both low power and high performance.

The P8700 extends capability from a single coherent six-core cluster with support I/O coherency to a new set of capabilities that enable more complex systems, such as:

- Multiple coherent clusters of CPUs
- Heterogeneous Multi-processing (CPU + GPU or other coherently designed processing elements)
- Groups of coherent I/O or co-processing functions or clusters

A cluster is composed of up to 0 - 6 CPUs and 0 - 8 IOCUs (sum being no more than 8 agents) and a Level 2 cache connection to a coherent interconnect. An agent is either a CPU, which is included in the cluster, or an external I/O device.

The P8700 implementation support is scalable up to 64 clusters, 512 cores, and 2048 harts (threads).

The P8700 cluster can be configured in one of two modes:

1. It can be configured as a single non-coherent cluster. In this case the main memory bus interface from the cluster is AXI-4.

2. It can be configured to support multiple coherent clusters. In this case, the main memory bus interface is ACE.

Figure 3 shows a reference design of a cluster integrated with a network.

Figure 3.   P8700 Integrated Cluster with Network
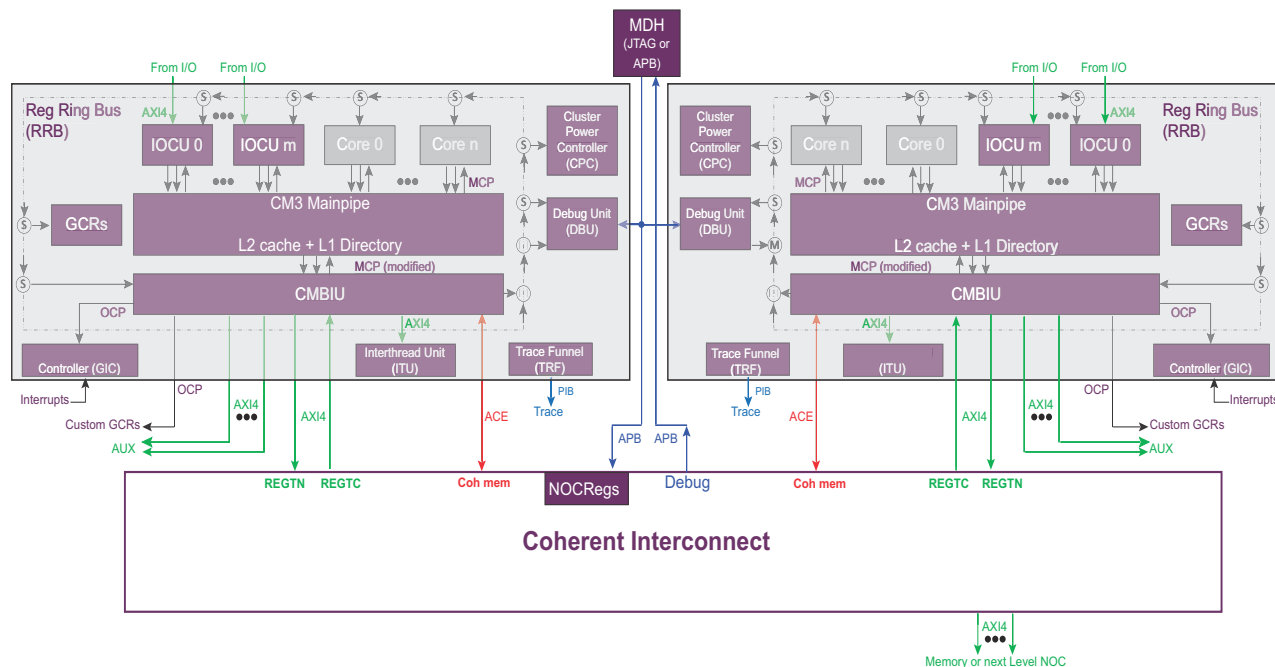
Directory Based Level 1 Cache Coherence

The Coherence Manager (CM) keeps all the L1 data caches coherent with each other by maintaining a directory that tracks the state of each L1 data cache line for each core. The directory uses the same address tags as the Level 2 cache, reducing the power and area required to maintain coherence. All Level 1 data and instruction cache misses are looked up in the directory to determine the state of the line in the L1 data caches as well as the L2 cache. Depending on the request attributes and directory state, the CM sends intervention requests to cores that have the line in their L1 data cache, reads the data from the L2 Data RAMs, or issues a request to the memory subsystem. The CM immediately updates the directory state and routes the corresponding data to the requesting core.

With a directory-based coherence architecture, each of the cores do not need to maintain a second copy of the L1 cache tags to "watch" the memory transactions and compare them against its internal cache contents. Instead, that information is maintained by the directory, which shares the L2 address tags.

L1 Instruction Cache Coherence

The Level 1 instruction caches are not coherent, in that the CM directory does not track their contents. However, L1 instruction cache misses will be looked-up in the CM directory, and depending on the state, may receive its data from a core's L1 data cache. This feature reduces the overhead of the software required to maintain L1 instruction cache coherence.

CM Main Pipeline

The CM Main Pipeline manages all the data and control flows throughout the CM and the P8700 Multiprocessing System.

The main pipeline implements the directory-based coherence architecture and manages a unified and shared L2 cache. Some key features of the L2 cache are:

- 64-byte cache line size
- 8- or 16-way set-associative
- 256 KB, 512 KB, 1 MB, 2 MB, 4 MB, and 8 MB cache-size options
- 1 or 2 cycle tag RAM access
- 2 or 4 cycle data RAM access
- 1 or 2 L2 cache pipelines, each with two memory banks
- Pseudo LRU line-replacement algorithm
- Writeback architecture
- L2 is inclusive of the L1 data caches, that is, it is always a superset of all L1 Data Caches
- Physically Indexed and Physically Tagged
- Non-Blocking architecture (Fully Pipelined)
- 48-Bit Physical Address
- L2 Hardware Prefetcher automatically recognizes workloads, such as memcopy, and efficiently prefetches data into the L2 cache
- Hardware can automatically initialize L2 cache upon reset. Hardware can also be programmed to initialize/flush all or part of the L2 cache.
- Cache line locking support
- ECC support (single-bit error correction and double-bit error detection) for Tag and Data arrays
- Parity support on data buses

The CM main pipeline arbitrates among the requests received from the cores, IOCUs, and L2 hardware prefetcher. It accesses and updates the directory and L2 cache tags, performs reads or writes to the L2 data RAMs as necessary, and issues interventions to manage each core's L1 data caches.

Uncached requests are also handled by the CM main pipeline, but neither the directory nor L2 cache is accessed. Uncached accesses are decoded based on a programmable address map and routed to the CM's Bus Interface Unit (CMBIU). The programmable address map determines the final target of the request, such as uncached memory or a configuration register in the interrupt controller, power controller, etc.

The CM main pipeline identifies and resolves conflicting accesses as required.

The CM includes high performance features for data movement:

- 512-bit wide internal data paths throughout the CM

- Three channel (two of 128-bit wide) system MCP interface to each of the CPU cores and IOCUs
- When configured as multi-cluster, ACE interface to inter-cluster network; AXI4 interface when configured as single cluster
- Support for up to 4 non-coherent Auxiliary AXI4 ports.

Cluster Power Controller (CPC)

Individual CPUs within the cluster can have their clock, power, or both gated off when they are not in use. This gating is managed by the Cluster Power Controller (CPC). The CPC handles the power shutdown and ramp-up of all cores in the cluster. The CPC can be controlled via software by accessing and changing values in the registers and by hardware through a signal interface.

The CPC also organizes power-cycling of the CM, dependent on the individual core status and shutdown policy. Reset and root-level clock gating of individual CPUs are considered part of this sequencing.

The CPC also controls the clock ratios of the cores, CM, I/O buses, and main memory bus. The CPC allows for the clock ratio of each component to be controlled independently, programmed by means of software commands or hardware signals. The clock ratio can be changed dynamically while the system is fully operating.

Reset Control

The reset input of the system resets the Cluster Power Controller (CPC). Reset sideband signals are required to qualify a reset as system cold, or warm start. Signal settings determine the course of action at deassertion of reset:

- Remain powered down
- Go into clock-off mode
- Power-up and start execution

In case of a system cold start and after reset is released, the CPC powers up the P8700 CPUs as directed in the CPC cold start configuration. If at least one CPU has been chosen to be powered up on system cold start, the CM is also powered up.

At a warm start reset, the CPC brings all power domains into their cold start configuration. To ensure power integrity for all domains, the CPC ensures that domain isolation is raised before power is gated off. Domains that were previously powered and are configured to power up at cold start remain powered and go through a reset sequence.

The CM includes memory-mapped registers that can override the default exception vector location. This allows different boot vectors to be specified for each of the harts, so they can execute unique code if required. Furthermore, signals by the system also determine which harts on each core start execution up. The CPC implements the capability to bring a core out of reset with no harts running, letting the system hardware start one or more harts at a later time.

I/O Coherence Unit (IOCU)

Hardware I/O coherence is provided by the I/O Coherence Unit (IOCU), which maintains I/O coherence of the caches in all coherent CPUs in the cluster.

The IOCU acts as an interface block between the Coherence Manager (CM) and I/O devices. Reads and writes from I/O devices may access the L1 and L2 caches by passing through the IOCU and the CM. Each request from an I/O device may be marked as coherent, or uncached. Coherent requests access the L1 and L2 caches. Uncached requests bypass both the L1 and L2 caches and are routed to main memory.

The IOCU provides an AXI slave interface to the I/O interconnect for I/O devices to read and write system memory.

The IOCU provides several features for easier integration:

- Supports incremental bursts up to 256 beats (128 bits per beat) on I/O side. These requests are split into cache-line sized requests on the CM side
- Coherent writes are issued to the CM in the order they were received

Interrupt Controller

The I8500 Interrupt Controller is a set of functional blocks that handle the distribution of interrupts among the CPUs and harts in an I8500 cluster. It is compatible with the RISC-V Advanced Interrupt Architecture (AIA) specification, and compliant with the RISC-V architecture, and consists of the following components:

- Advanced Platform-Level Interrupt Controller (APLIC)
- Advanced Core Local Interrupt (ACLINT) Machine-level Timer
- ACLINT Machine-level Software Interrupt (MSWI)
- ACLINT Supervisor-level Software Interrupt (SSWI)
- Watchdog Timer (WDT)

In addition to the standard components, the Interrupt Controller implements custom extensions to support Non-Maskable Interrupt (NMI) routing, timer synchronization, and Watchdog Timer (WDT) configuration. The Interrupt Controller does not implement the AIA IMSIC component.

The Interrupt Controller has the following features:

- Software interface through relocatable memory-mapped address range
- Configurable number of system interrupts from 8 to 512 in powers of two
- Support for different interrupt types:
  - Level-sensitive: active-high or active-low
  - Edge-triggered: positive-edge or negative-edge triggered
- Ability to mask and control routing of interrupts to a particular hart
- Interrupt prioritization
- Software/Inter-processor interrupts

![MIPS logo — a GlobalFoundries company]

- Timer interrupts
- Watchdog Timer

Global Configuration Registers (GCR)

The Global Configuration Registers (GCR) are a set of memory-mapped registers that are used to configure and control various aspects of the Coherence Manager and the coherence scheme.

Some of the control options include:

- *Address map* — The base address for the various peripheral blocks, such as the CPC, GCR, User GCRs, and Interrupt Controller address ranges can be specified

- *Error reporting and control* — Logs information about errors detected by the CM and controls how errors are handled (ignored, interrupt, etc.)

- *Control Options* — Various features of the CM can be disabled or configured

- *L2 Cache operations* — Registers used during L2 cache maintenance instructions

- *Mapping registers* — Route requests to one of the non-coherent Auxiliary (AUX) AXI-4 ports

- *Multi-cluster register access* — Allows a CPU of one cluster to access a register on a remote cluster via the REGTC/REGTN AXI4 buses

Custom GCRs

The CM provides the ability to implement a 64 KB block of custom registers that can be used to control system level functions. These registers are user defined and then instantiated into the design. Two global registers are provided by the CM to implementation custom registers: the Global Custom Base register, and the Global Custom Status register.

Clocking Options

The P8700 Multiprocessing System has the following clock domains:

- *Reference Clock* — This clock is created by the SOC and used by the P8700 Multiprocessing System. The reference clock is controlled and scaled by the input clock. This clock drives the CPC.

- *Prescaled clock* — The reference clock can be prescaled by a programmable value of 1:1 (no prescale) to 1:255. This prescaled clock is used a base clock for all cluster components, except the CPC.

- *Cluster clock domain* — This clock drives the CM (including Coherence Manager, Global Interrupt Controller, IOCU, and L2 cache). This clock can be configured to be the same as Prescale Clock or Prescale / 2.

- *Core-N clock domain* — Each core in the cluster can operate at independent frequency. This clock can be controlled at run time (via CPC).

- When the CM is operating at 1:1, the cores can run at 1:1, 1:2, 1:3, 1:4, 1:5, 1:6, 1:7 or 1:8 of the prescale clock.

- When the CM is operating at 1:2, the cores can run at 1:1, 1:2, 1:4, 1:6, or 1:8 of the prescale clock.

- *System clock domain* — The AXI-4 or ACE port connecting to the SOC and the rest of the memory subsystem may operate at a ratio of the cluster clock domain. The system clock domain can be configured to use an internal clock or an external clock. When configured to use an internal clock, the rate is a ratio of the prescale clock. Supported ratios are 1:1, 1:2, 1:3, 1:4, 1:5, 1:6, 1:7, 1:8.

- AUX AXI clock domains — The optional non-coherent AXI-4 port connecting to the SOC may operate at a ratio of the cluster clock domain. Each auxiliary AXI clock domain can be independently configured to use an internal clock or an external clock. When configured to use an internal clock, the rate is a ratio of the pre-scale clock. Supported ratios are 1:1, 1:2, 1:3, 1:4, 1:5, 1:6, 1:7, 1:8.

- When configured to use an external asynchronous clock, the AXI interface captures/drives on that clock and there is an asynchronous boundary crossing implemented internal to the cluster.

- *TAP clock domain* — This is a low-speed clock domain for the JTAG TAP controller, controlled by the EJ_TCK pin. It is asynchronous to the Reference Clock.

- *I/O clock domains* — Each port connects the IOCU to the I/O Subsystem. Each IOCU clock may operate at a ratio of the prescale clock domain. Supported ratios are the same as the system clock domain. Similar to the System clock domain, each I/O clock domain can be configured to operate at a ratio of the prescale clock or an asynchronous external clock.

Debug Unit

The Debug Unit (DBU) is an optional component that is compliant with the RISC-V debug specification. The DBU enables debug using a probe connected through a JTAG scan chain. Alternatively, the DBU can be connected to the system through an APB transactor port. The DBU contains the single TAP controller in the cluster, which can access registers through the cluster. The DBU also contains a RAM to hold instructions and data accessed by the cores while in debug mode.

Features of the debug unit include RISC-V triggers and a Fast Debug Channel.

RISC-V triggers stop the normal operation of the CPU and force the system into debug mode. There are two types of RISC-V triggers implemented in the P8700 CPU: Instruction triggers and Data triggers.

Instruction triggers occur on virtual instruction execution addresses and may be qualified by an ASID, hart, GuestID, and context. Addresses may be single, masked, or ranges.

Data triggers occur on load and store operations based on virtual address, ASID, hart, GuestID, context, and data value. Addresses may be single, masked, or ranges. Loads and stores may be aligned or misaligned.

The Fast Debug Channel is a mechanism for efficient bidirectional transfer between a CPU and the debug probe. Data is transferred serially via the TAP interface. Memory-mapped FIFOs buffer the data, isolating software running on the CPU from the actual data transfer. Software can configure the FDC block to generate an interrupt based on the FIFO occupancy level or can operate in a polled mode. Up to 16 virtual channels can travel in each direction.

Program Trace

The P8700 core includes trace support for real-time tracing of instructions, data addresses, data values, and performance counters. The trace block is compliant with the RISC-V Trace Specification. A Trace funnel muxes the trace stream from all cores and the CM, and either stores the trace information into an on-chip trace RAM or off-chip memory for post-capture processing by trace regeneration software. Software-only control of trace is possible in addition to probe-based control. The on-chip trace memory may be accessed either through load instructions or the existing JTAG TAP interface, which requires no additional chip pins.

The off-chip trace is managed with the PIB3 (3rd-generation Probe Interface Block) hardware that ships with the product. It provides a selectable trace port width of 8 or 16 pins plus DDR clock. The trace data is streamed on these pins and captured using a compatible probe from popular third party tool vendors.

Initial and Possible Configurations

The P8700 Multiprocessing System can support a variety of configurations. Initially, the following configurations listed in Table 1 will be supported. If a different configuration is required, contact your sales account manager for current configurations and to request a new configuration.

Multi-core offerings are symmetric (same cache size, harts, and FPU for each core) but different clock ratios per core are supported.

Table 1. P8700 Configuration Options

Feature	Description	Allowed values
num_clusters	Number of coherent clusters. NOTE: This refers to the number of clusters in the example mips_soc. It does NOT refer to the number of clusters that a customer may instantiate in their design.	Generally set to 2 for non-inte-grated NOC because a dual-cluster mips_soc example is provided.
ace	MEM port includes AXI Coherency Extensions	0 (absent) or 1 (present)
num_cores	Number of cores per cluster; Total number of cores plus number of IOCUS must be less than or equal to 8	0, 1, 2, 3, 4, 5, 6
num_vps	Number of Virtual Processors per core	1 or 2
l1_icache_size	L1 Instruction Cache Size	32, 64 KB
l1_dcache_size	L1 Data cache Size	32, 64 KB
l2_cache_size	L2 Cache Size	256, 512, 1024, 2048KB
num_pipes	Number of pipes or scheduler paths in the Coher-ence Manager.	1 or 2
num_irqs	Number of interrupts.	8, 16, 32, 64, 128, 256, 512
num_iocus	Number of I/O Coherence Units (IOCUs); Total number of cores plus number of IOCUS must be less than or equal to 8	0 - 8
iocu_size	IOCU size information. Chooses the size of the IOCU implementation for reads/writes and SDB IDs.	Small = 4 RDs, 4 WRs, 4 SDB IDs Medium = 8 RDs, 8 WRs, 8 SDB IDs Large = 16 RDs, 12 WRs, 16 SDB IDs Xlarge = 32 RDs, 24 WRs, 32 SDB IDs
iocu_sdb_ids	Number of IOCU Store Data Buffer IDs. N.B: overrides iocu_size.	8 - 32
iocu_user_width	IO AxUSER width, routed to MEM port	0 - 9 (default 8)
l2_missq_override	L2 Cache Misses in Flight to memory (to override default)	8 - 96
l2_mem_sdb_override	L2 Store Data Buffer size (to override default)	8 - 64

Table 1. P8700 Configuration Options (Continued)

Feature	Description	Allowed values
l2_pfu_sdb_override	L2 PFU SDB ID allocation override	4 - 48
l2_num_intv_override	Number of incoming ACE snoop requests	4 - 16
ugcr	User Global Configuration Registers (GCRs).	0 (absent) or 1 (present)
num_relay_stages_cm_mcp_core	Number of relay stages between cores and CM	0, 1, 2 (0 default)
c_mem_ext_clk_en	External Clock on ACE memory port (only used if ace=1 and not integrated NOC)	0 (absent) or 1 (present)
pdtrace	PDTrace unit	0 (absent) or 1 (present)
tru_mem	PDTrace internal memory size	6 (16KB), 7 (32KB), 8 (64KB)
tru_ext_bus_type	Type of external bus for Pdtrace. PIB: output of Probe Interface Block. TC: 256b wide output of Trace Funnel	PIB or TC
tru_sys	System Trace Port Implemented	0 (absent) or 1 (present). Default = 0.
tru_PIB	PDTrace PIB size (width). Only used if tru_ext_bus is PIB	8, 16 bits
axi_addr_parity	AXI address parity supported.	0 (absent) or 1 (present)
axi_data_parity	AXI data parity supported.	0 (absent) or 1 (present)
reg_parity	Control and status register parity protection	1: Parity on status/control registers 0: No parity on status/control registers
timeout_fault	Transaction timeout protection	1: Transaction timeout protection 0: No transaction timeout
regtc_impl	Enable cluster register input port	0 (REGTC absent) or 1 (REGTC present)
regtn_impl	Enable cluster register output port	0 (REGTN absent) or 1 (REGTN present)
num_aux	Number of auxillary ports	0 - 4
aux0_data_width	Auxillary Port 0 data width	32, 64, 128, 256, 512
aux1_data_width	Auxillary Port 1 data width	32, 64, 128, 256, 512
aux2_data_width	Auxillary Port 2 data width	32, 64, 128, 256, 512
aux3_data_width	Auxillary Port 3 data width	32, 64, 128, 256, 512

Revision History

Table 2. Revision History

Revision	Date	Description
1.00	January 31, 2022	Initial release of P8700 Data Sheet.
1.01	June 22, 2022	Added configuration to Table 2, DBU TAP daisy-chain connection.
1.10	July 24, 2023	Removed all references to SIMD and MSA. Updated Figure 1, Block Diagram of Single Cluster P8700. Removed references to compact branches. Removed references to GIC and replaced with AIA. Updated CPU Core-Level Features section. Replaced P8700 Core Block Diagram in Figure 2. Updated Instruction Fetch Unit (IFU) section. Updated Branch Prediction section. Removed section on Execution Unit (EXU). Updated Instruction Buffer Management and Issue section. Updated Integer Execution Units section. Updated Floating Point Pipelines section. Updated Store and Write Buffer section. Updated Variable Size TLB (VTLB) section. Updated Fixed Size TLB (FTLB) section. Updated Debug Unit section. Removed Inter-CPU Debug Breaks section. Removed PC Sampling section. Updated Table 1, P8700 Configuration Options.
1.20	April 8, 2024	Updated document to new template.
1.21	June 7, 2024	Updated supported page sizes for ITLB and DTLB.
1.22	November 6, 2024	Removed references to Hypervisor. Removed references to lock-step. Updated Figure 1, Block Diagram. Standardized document for 6 CPU's maximum. Updated Table 1, Configuration Options. Updated AIA references to Interrupt Controller. Standardize naming as RV64GCZba_Zbb.
1.23	January 14, 2025	Miscellaneous updates from internal review, Removed references to virtualization.

Document Number: MD01500

Legal Statement

This publication contains proprietary information which is subject to change without notice and is supplied 'as is' without warranty of any kind. MIPS, the MIPS logo, Meta, and Codescape are trademarks or registered trademarks of MIPS LLC. All other logos, products, trademarks and registered trademarks are the property of their respective owners.