MIPS P8700 Processors for Automotive Driver Assistance Systems and Autonomous Driving System-on-Chips

September 2025

MIPS.com
2870 Zanker Rd #210 San Jose CA 95134

Contents

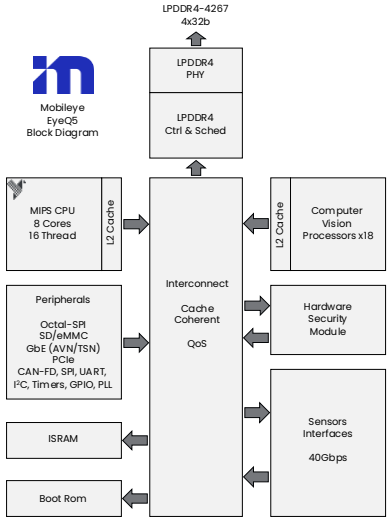
MIPS.com
2870 Zanker Rd #210 San Jose CA 95134

1. Introduction to ADAS/AD

Autonomous Driving (AD) is the ability of a vehicle to drive itself. ADAS (Advanced Driver Assistance Systems) are electrical/electronic systems designed to assist a driver of a vehicle using a suite of sensors like cameras, radar, lidars, ultrasonics and high-performance SoCs which are increasingly shifting towards Artificial Intelligence. Such systems not only enhance safety but also reduce driver fatigue and monitor alertness. According to a NHTSA report, in 2023 more than 40000 lives were lost in road accidents in US alone. Many of these were attributed to human errors which could have been significantly avoided with the help of ADAS systems.

ADAS SoCs are designed to handle very high computational and I/O workload. They also work under tight power/thermal budgets and space constraints. Therefore, most SoCs adopt a heterogeneous architecture where dedicated hardware blocks are used to accelerate specific functions. An example is shown below:

Figure 1 : Mobileye EyeQ5 Block Diagram



Source: *https://www.eetimes.com/mobileyes-new-eyeq5-how-open-is-open/*

The SoC also has 8 CPU cores. But if most of the computation is offloaded to HWAs, what is CPU's role in this SoC? To understand this, let's see what's going on in a typical ADAS SoC.

1. A suite of sensors provides environmental data streams over high-speed interfaces (like MIPI CSI-2, Ethernet etc.). The SoC receives sensor data and sends control

MIPS.com
2870 Zanker Rd #210 San Jose CA 95134

signals back to sensors over interfaces like I2C, SPI etc. In this data & control flow, the CPUs are responsible for:

   a. Configuring the communication interfaces
   b. Programming the HWAs
   c. Handling new frame events in a timely manner and passing the frame data to the right HWA
   d. Updating HWA configuration for optimal sensor processing. For example, Image Signal Processor (ISP) parameters must be updated with changing lighting conditions. CPU is constantly running image quality monitoring algorithms to compute new parameters and updates ISP registers accordingly.
   e. Closed loop feedback to the sensors – for example, Auto-Exposure control to update image sensor registers over I2C.

2. Non-vision sensors like radar, lidar, USS, etc. must be converted into a form where they can be fused with camera data for sensor fusion. One form of sensor fusion is object level fusion which takes object list detected in different modalities and fuses them together. Many of these steps involve processing like tracking (e.g. Kalman Filtering) and clustering (e.g. K-Means clustering) to group points in a point cloud into objects. These are computationally intensive tasks which do not map well to any HWA and fall back to the CPU.
3. HWAs process sensor data and generate an Environment Model (EM) using a combination of traditional CV and advanced AI based algorithms. CPUs are responsible for orchestrating the dataflow, ensuring that HWAs are always working on the latest sensor data and have the needed parameters loaded from external memory. In many cases, HWAs are chained such that the output of one block becomes the input of the next block. System software running on the CPU is responsible for ensuring the sanctity of buffers and correctness of the processing sequence.
4. Drive policy, a software algorithm, uses the EM as an input to take driving decisions. One example of an EM is an occupancy grid. Based on the grid, the vehicle decides to take the appropriate action. For e.g. it decides if the vehicle should go faster, or slow down, or apply a Minimum Risk Maneuver (MRM) to avoid a collision. Drive policy is another example of a task which does not map well to any HWA and usually falls back to the CPU.

MIPS.com
2870 Zanker Rd #210 San Jose CA 95134

5. In addition to the tasks described above, CPU is responsible for running and managing a large number of miscellaneous tasks. Some examples are
    a. High Level OS (Linux/QNX).
    b. File system management.
    c. Running device drivers.
    d. Running Self-Test Libraries to ensure system safety and reliability.
    e. Monitoring errors and activating recovery sequence.

ADAS workloads are very diverse, with hundreds of processes and software threads simultaneously running a combination of compute-intensive and low latency control tasks. Computationally intensive tasks like tracking, clustering, driver policy, ISP algorithms require more ALUs, FPUs and high memory bandwidth. Control-centric tasks like device drivers, interrupt handling, HWA control and sensor feedback loops require fast context switches, accurate branch prediction and low latency peripheral control. This complexity is not accurately reflected by synthetic benchmarks (DMIPS, SpecInt, CoreMark) and using such benchmarks for ADAS CPUs can be quite misleading.

MIPS.com
2870 Zanker Rd #210 San Jose CA 95134

In the next section we look at MIPS P8700 processor and see why it is the right choice for SoCs running ADAS and AD use cases.

2. MIPS® P8700 Multi-Processing System (MPS)

2.1. Overview

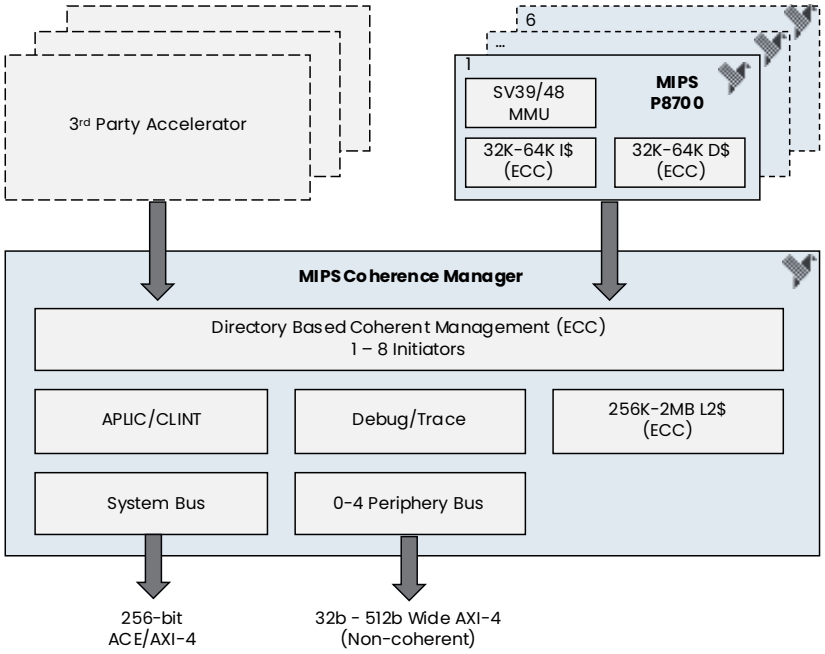


Figure 2 : MIPS® P8700 Multi-Processing System

MIPS® P8700 MPS is the first RV64 compliant CPU IP focused on high performance, safety-critical Automotive applications and certified against random failures to the ISO26262 ASIL-B standard, and systematic failures to the ISO 26262 ASIL-D standard. MIPS has a proven track record with the compute IP deployed in more than 28 car models across global OEMs. As such, the P8700 MPS provides best-in-class multi-core performance for use in Automotive system-on-chip (SoC) applications. The P8700 combines a deep pipeline with multi-issue Out-Of-Order execution and multi-threading to deliver outstanding computational throughput and low latency sensor control.

MIPS.com
2870 Zanker Rd #210 San Jose CA 95134

2.2. Feature Set

- RISC-V RV64 ISA compliant
  - RVA20 standard profile
  - Zba + Zbb extensions (bit manipulation)
  - SV48 – virtual memory supporting up to 48-bit addressing.
- 48-bit Physical Addressing
- 4-issue, out-of-order, 16-stage pipeline
- 7 execution units
  - 2 Integer ALUs
  - 1 Load
  - 1 Store
  - 2 Floating Point Units (FPUs)
  - 1 Integer Multiply and Divide Unit (MDU)
    - 64x64 bit multiplier
    - radix 4 SRT divider
- Single/Double precision multi-threaded FPU
- Simultaneous Multi-Threading (SMT), up to 2 Hardware Threads per core
- L1 I and D Caches. 32/64KB. 4-way set-associative
- Coherence Manager (CM) for multi-core coherent clusters
  - Integrated L2 Cache up to 2MB, 8/16-way set-associative.
  - Up to 6 CPUs per cluster
  - Scalable to multi-cluster via ACE interface to system/NoC
- I/O coherence units (IOCUs)
  - Provide one-way (I/O) coherency between CPUs, I/O devices and hardware accelerators
  - Directory-based coherency scheme – improves power consumption, performance and scalability

  - Enables "accelerator-only" clusters without CPU cores, with up to 8 IOCU ports per cluster, targeting use cases like AI/ML acceleration and safety-critical systems.
- RISC-V compliant APLIC (Advanced Platform Level Interrupt Controller) implementation, compatible with RISC-V AIA specification
- Cluster Power Controller (CPC) for fine-grained clock and power management
  - Individual CPUs within the cluster can have their clock, power, or both gated off
- Load/Store bonding - Two consecutive loads/stores of same type which access contiguous memory locations are combined into a single operation. Results in 2x improvement in memory intensive code.
- Unaligned load/store

3. How MIPS P8700 accelerates ADAS/AD workloads

3.1. Scalability

As the level of automation increases from L1-L5, the system complexity also increases, sometimes exponentially. Higher levels of autonomy require more sensors, higher resolutions, more advanced AI, redundancy and failsafe operation. This increases demand for I/O, computational and memory resources in SoC, with the CPU being at the center of everything. This calls for a scalable architecture from a dual-core CPU for L1 to 20+ CPUs for L3 and above. Coherence Manager (CM) in P8700 MPS supports up to 6 cores/cluster**,** and paves way for SoC architectures scalable from L1- L5.

MIPS.com
2870 Zanker Rd #210 San Jose CA 95134

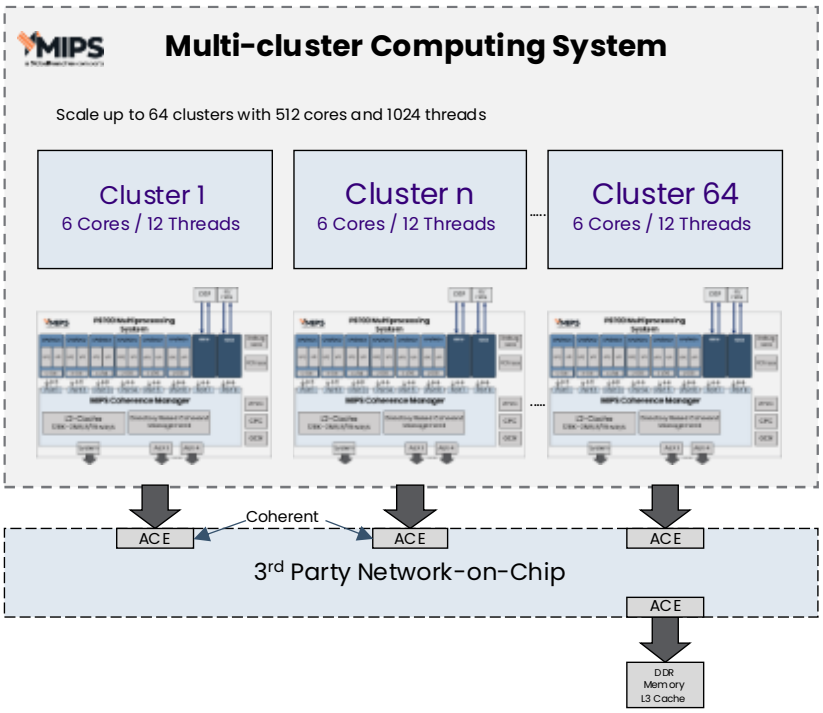


Figure 3 : Multi-Threaded Multi-Core Multi-Cluster Computing System

MIPS Coherent Manager acts as a high-performance interconnect that controls traffic between up to 8 ports for CPUs and IOCUs (I/O Coherent Units). It maintains L2 Cache (up to 2MB) and system level coherency between all cores. It uses a directory-based cache coherency protocol, which is more scalable and offers better performance than a snooping-based protocol, making it the right choice for large scale ADAS/AD SoCs.

MIPS.com
2870 Zanker Rd #210 San Jose CA 95134

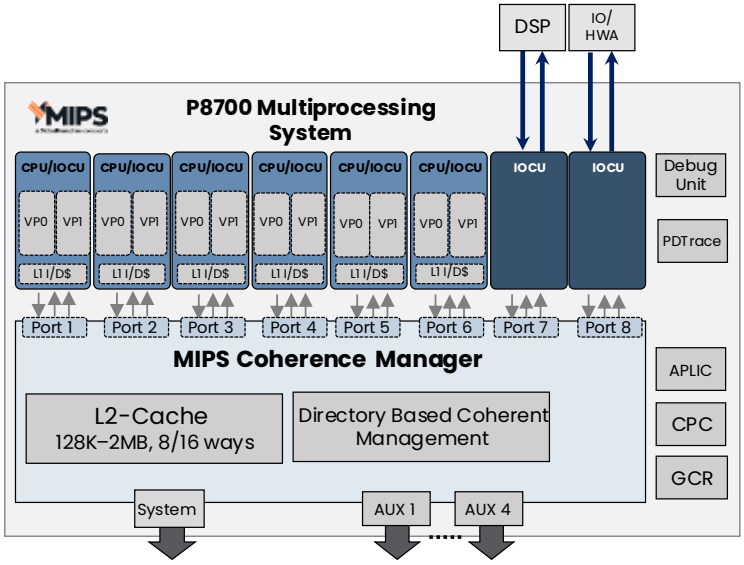


Figure 4 : Coherence Manager (CM)

As we discussed earlier, ADAS workload is a blend of computational tasks and control tasks, each requiring a different kind of CPU resource utilization. Simply scaling up SoCs with homogeneous cores results in underutilization of CPU resources. For example, imagine a 20+ SpecInt rated core spending most of its time handling interrupts and moving pointers between HWAs. It is not unusual to see high power CPUs in ADAS SoCs doing low value tasks. So, what is the solution? Simultaneous Multithreading (SMT) to the rescue!

3.2. **Simultaneous Multithreading** (SMT)

SMT is a technology which allows every CPU core to have multiple Hardware Threads (HARTs). From a software perspective, each HART appears as a unique CPU. A CPU subsystem with 8 cores x 2 HARTs appears as a 16-core subsystem to the software. An SMP operating system (like Linux or QNX or RTOS) will be able to seamlessly distribute workload across multiple HARTs, without any software changes. Under the hood, P8700 executes multiple threads simultaneously so that two execution contexts are running at the same time, sharing CPU resources.

MIPS.com
2870 Zanker Rd #210 San Jose CA 95134

Resource utilization is a challenge in single-threaded CPUs because of frequent stalls and dependencies between pipeline stages. If that happens in P8700, SMT hardware scheduler issues instructions from the other thread and keeps the resources utilized. This results in higher IPC (Instructions Retired Per Cycle), and increased throughput.

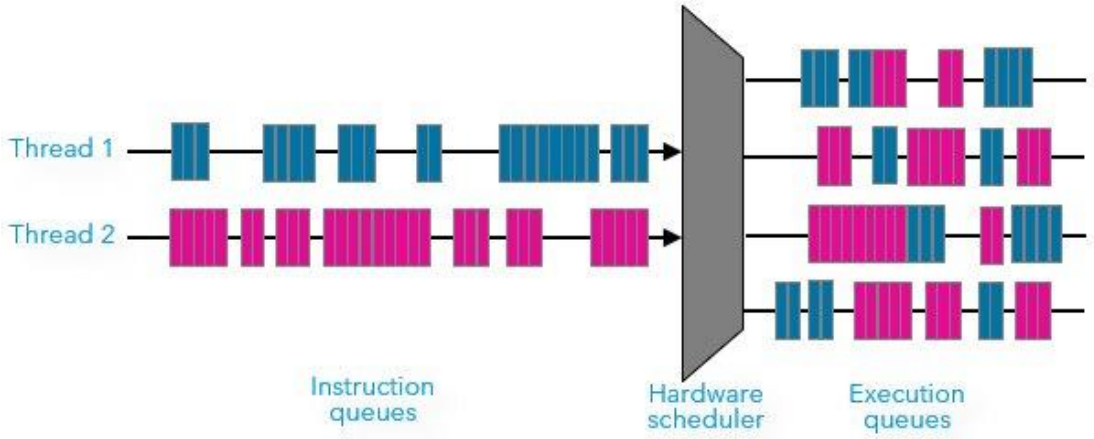


Figure 5 : Simultaneous Multithreading (SMT)

An additional benefit of multi-threading is fast context switches. A single-threaded CPU core has one hardware context which includes general-purpose registers (GPRs), a program counter (PC), and some multiplier and coprocessor state. On a task switch, the context must be saved to memory and must be restored when task execution resumes.

ADAS CPUs receive 1000s of interrupts and task switches every second and therefore context switching is a major source of efficiency loss.

MIPS.com
2870 Zanker Rd #210 San Jose CA 95134

3.3. I/O Coherence

I/O Coherence is a big challenge in complex heterogeneous systems directly affecting system reliability, performance, and correctness when multiple masters access shared memory. This is of special concern in ADAS SoCs which may have several input/output ports, multi-core CPU clusters and a large number of DSPs and hardware accelerators. Take the sensor fusion example as shown below

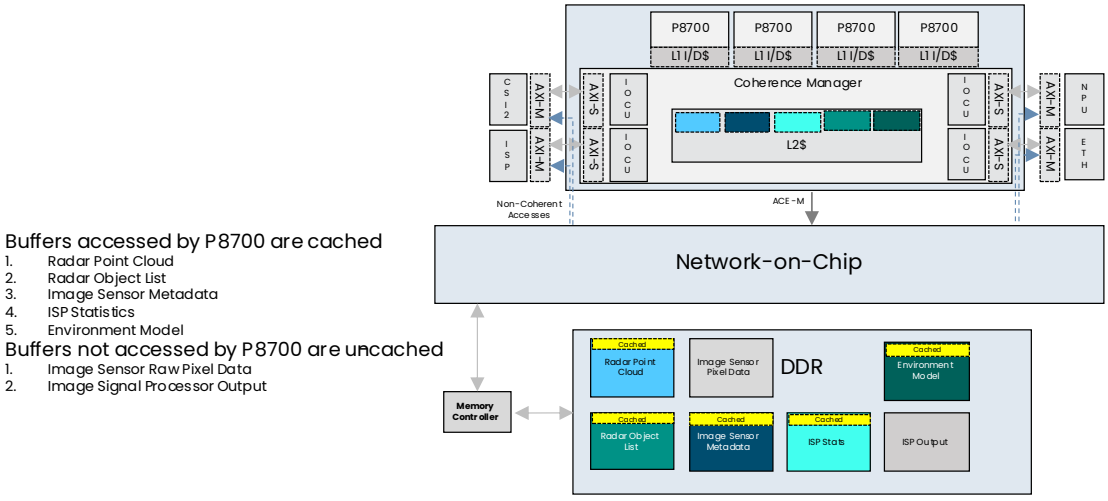


Figure 6 : Sensor Fusion Example

1. Radar sends point cloud over Ethernet.
2. P8700 converts point cloud to object list.
3. In parallel, camera sensor sends raw pixel data & metadata over CSI-2
4. ISP (Image Signal Processor) processes raw pixels to YUV/RGB and generates 3A statistics.
5. P8700 reads sensor metadata and ISP generated statistics for running Auto-Exposure, Auto-White Balance and other sensor control algorithms.
6. NPU reads radar object list and ISP output. Runs an AI based sensor fusion algorithm to generate Environment Model.

MIPS.com
2870 Zanker Rd #210 San Jose CA 95134

7. P8700 reads Environment Model to execute drive policy.

For best performance, P8700 would cache the memory it is addressing. However, the same memory is accessed by other masters in the system. For correct, reliable and safe system behavior, IO Coherence Unit (IOCU) ensures one-way coherency between CPU and other masters in the SoC. This is extensible to multi-cluster topologies, as shown below:

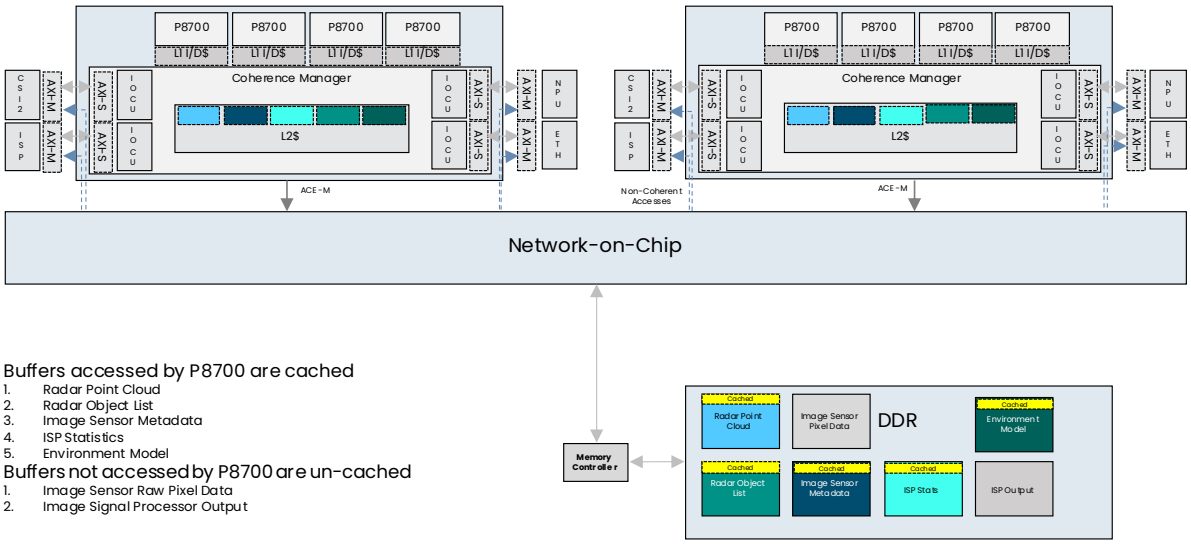


Figure 7 : Multi-Cluster I/O Coherency Example

MIPS also offers the coherence manager in an "accelerator-only" configuration without CPU cores. This makes it unique for ADAS SoCs which have a very large number of non-CPU masters including multiple CSI-2 ports, Ethernet ports, ISPs, DSP, NPUs, GPUs and other accelerators.

3.4. Functional Safety

No discussion on ADAS would be complete without Functional Safety (FuSa). MIPS P8700 is developed as a SEooC (Safety Element Out of Context) in compliance with

MIPS.com
2870 Zanker Rd #210 San Jose CA 95134

ISO26262:2018. It meets ASIL-B for random failures and ASIL-D for systematic failures which makes it usable in systems up to ASIL-D with the right decomposition strategy.

Highlighted blocks (in yellow color) in the diagram below show the safety features of P8700.

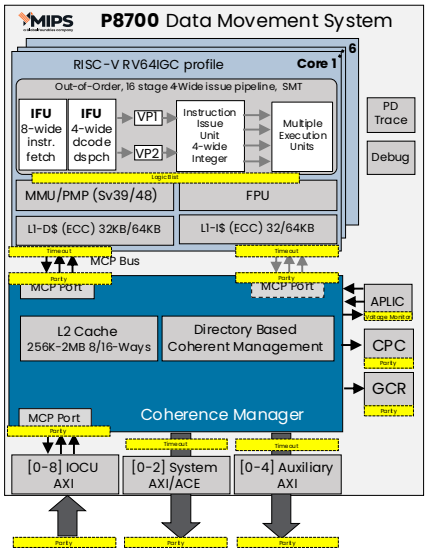


Figure 8 : P8700 Functional Safety Features

Functional Safety (FuSa) Feature Set Summary

- End-to-end parity protection on address and data buses.
- Parity protection of software visible registers in the configuration register, interrupt controller, and Cluster Power Controller.
- ECC on all L1 instruction, L1 data, and L2 caches.
- Fault bus to report detected faults to external fault handling logic.
- Programmable transaction timeout detection on memory requests originating from a CPU or IOCU.

MIPS.com
2870 Zanker Rd #210 San Jose CA 95134

- Protocol error detection on IOCU and AXI slave interfaces.
- AXI/ACE interface parity protection of address and data compatible with third-party interconnects.
- Redundancy for Safety on critical registers and on-chip memories.
- LBIST-enabled.
- Events and status to monitor voltage spikes and permanent damage.
- Self-Test Libraries

MIPS has partnered with Resiltech for safety analysis Product certification up to ASIL-D. MIPS delivers a safety package to its licensees. Safety package includes

- Functional Safety Assessment Report Certificate
- Safety User's Manual
- Safety Application Note
- FMEDA worksheet

4. Conclusion

The MIPS P8700 processor is a powerful, scalable and customizable IP designed specially to handle the diverse workloads of ADAS SoCs, from Level 1 to Level 5 applications. It combines the benefits of superscalar architecture, simultaneous multithreading, memory and I/O coherency to achieve high throughput in ADAS applications with low power and area footprint.

The MIPS P8700 comes with a full suite of functional safety readiness, up to ASIL-B for random faults and ASIL-D for systematic fault coverage. The processor is a fully synthesizable and configurable IP, giving SoC designers freedom to match the CPU subsystem to the target application performance requirements.

For more information, please visit https://mips.com/products/hardware/p8700.

MIPS.com
2870 Zanker Rd #210 San Jose CA 95134