

Technical White Paper

TI's High-Performance Network Processing Accelerator (NPAC)



Abstract

New vehicle architectures and technologies such as software-defined vehicles and autonomous driving require a fundamental redesign of the electrical and electronic (E/E) architecture—shifting from highly distributed to domain-based, zonal-based, or hybrid approaches. These new architectures significantly increase the performance and security requirements of the vehicle network. The most efficient way to address these increased requirements is through dedicated hardware-based accelerators such as Texas Instruments (TI) Network Processing Accelerator (NPAC). TI's NPAC offloads networking and security tasks from the main compute cores to dedicated hardware accelerators and programmable RISC-V cores.

Table of Contents

1 Introduction	2
1.1 Zonal Architectures	2
2 Network Processing Accelerator (NPAC)	3
2.1 NPAC Architecture	4
2.2 NPAC Use Case Examples	4
2.3 NPAC Features and Benefits	5
3 Summary	6

Trademarks

Jacinto™ is a trademark of Texas Instruments.
All trademarks are the property of their respective owners.

1 Introduction

The automotive industry is undergoing the most significant architectural transformation since the invention of the automobile. Technologies such as software-defined vehicles, powertrain electrification, 48V systems, X-by-wire technology, AI-powered assistants, and autonomous driving are improving safety, performance, efficiency, and fundamentally changing how passengers interact with the vehicle.

To support these new technologies, the underlying vehicle electrical and electronic (E/E) architecture must be completely re-designed. Traditional distributed E/E architectures have evolved over time into a complex network of up to 150 electronic control units (ECUs) and thousands of meters of wiring. Each ECU operates with unique hardware and software, connected through a complex, costly wiring network. Managing software across these distributed ECUs is inherently difficult and expensive. The first attempt at simplification was the adoption of domain-based architectures, where electronics are grouped by function. This reduced complexity, but ECU count can still be as high as 100 in higher-end vehicles. To further simplify E/E architectures and enable new technologies like software-defined vehicles and autonomous driving, manufacturers are transitioning to zonal-based architectures.

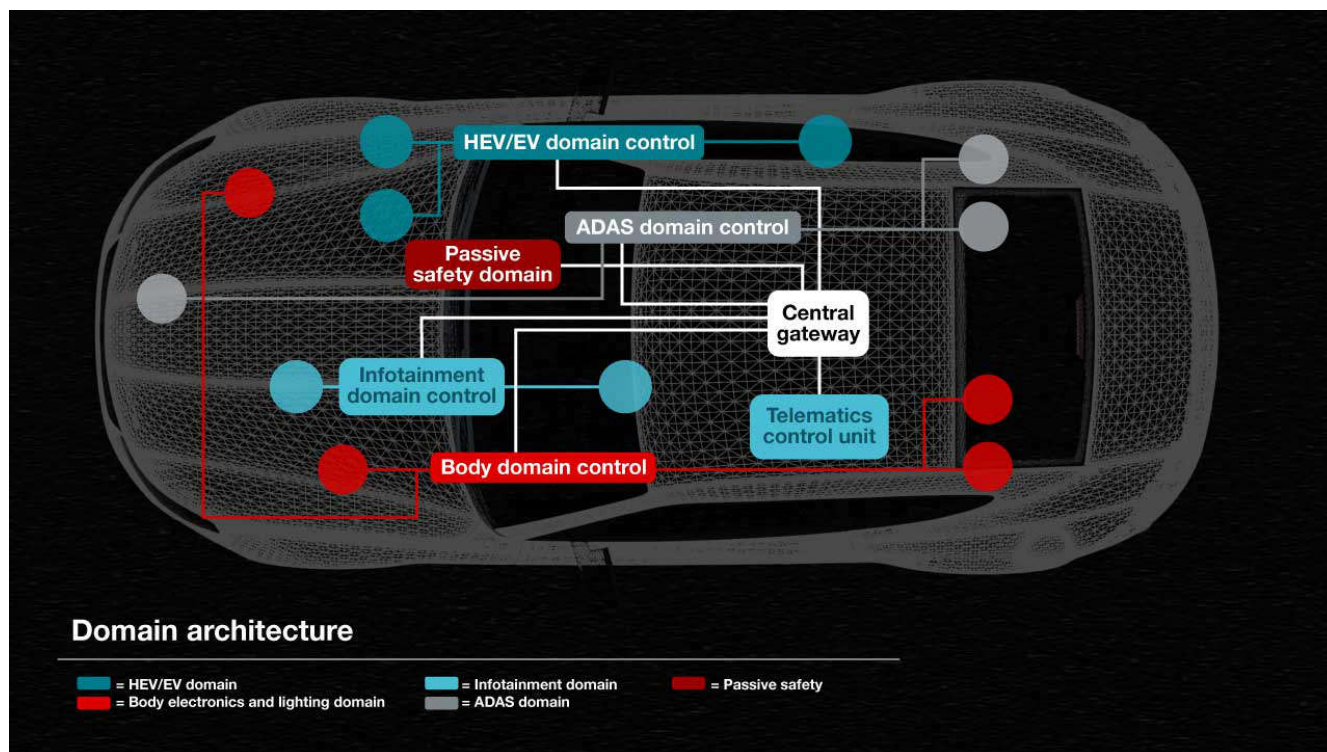


Figure 1-1. Domain Architecture With Distributed Processing and Costly, Complex Wiring Network

1.1 Zonal Architectures

Zonal architectures organize electronics by location, not function, dividing the vehicle into zones, such as front-left, front-right, back-left, and back-right. Each zone is connected to the high-performance compute (HPC) module through high-speed Ethernet, where much of the vehicle processing occurs. Each zone contains a zone controller that performs local processing, packetization and routing of sensor and control data between the zone and HPC module.

The amount of processing performed in the zone versus the HPC module is dependent on the architectural choices and the function being performed. For example, to minimize latency or lower costs, keeping processing in the zone for certain functions versus routing processing to the HPC module may make more sense.

Advantages of zonal architectures include:

- Reduced ECU count and complexity – fewer parts to manage, integrate, and secure
- Lower cost – less cabling, simpler wiring harness, fewer components

- Easier software updates – OTA updates deployed once, not across dozens of ECUs
- Service agility – new features added using software without hardware redesigns

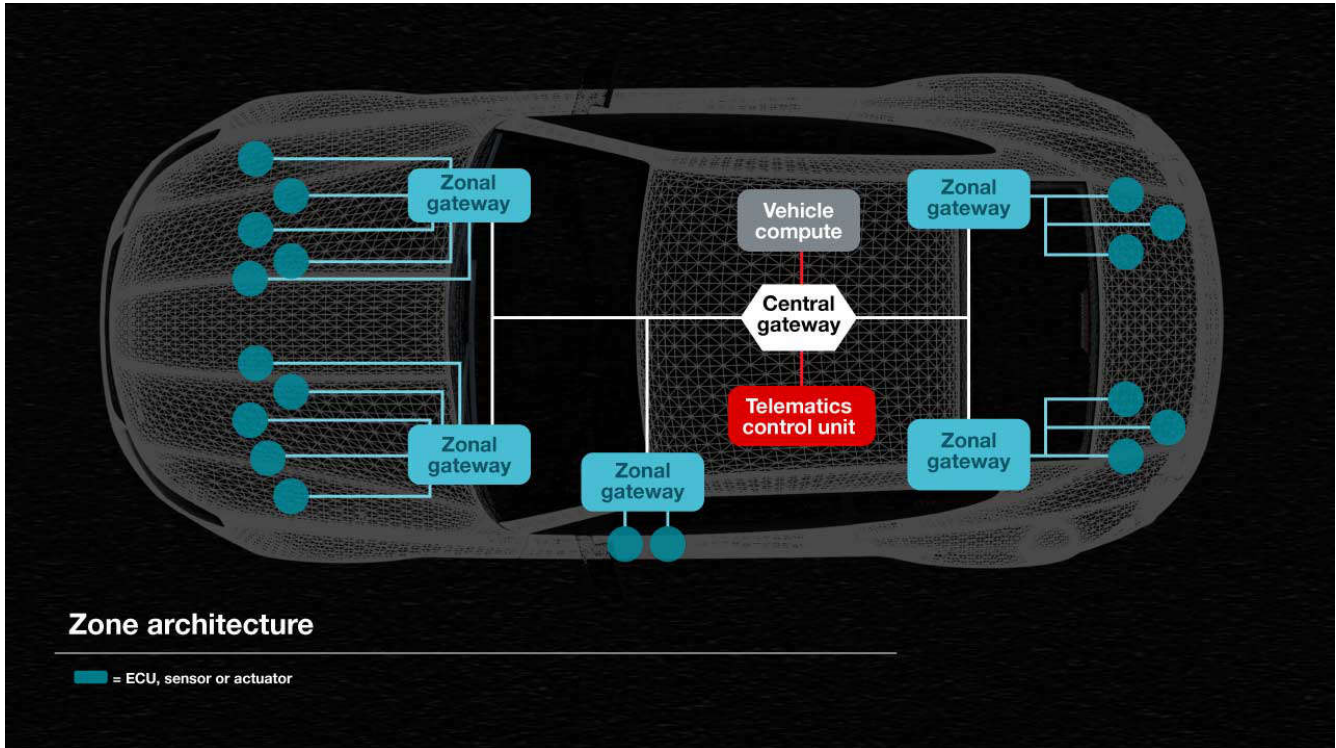


Figure 1-2. Zone Architecture With Centralized Processing and Simplified, High-Speed Ethernet Network

Zonal architectures significantly increase performance demands on the vehicle network. Sensor data must be aggregated and transmitted to the HPC module, processed, and commands returned in real-time. The vehicle network must simultaneously support diverse applications, from safety-critical braking and steering to audio and interior lighting convenience features, each with differing bandwidth and latency requirements. Managing quality-of-service and priority arbitration across this heterogeneous traffic becomes a critical design challenge.

The most efficient way to meet rising vehicle networking performance demands is through the use of hardware-based networking and security accelerators. Hardware-based accelerators provide low power deterministic offloading for networking, cryptographic, and firewall functions, freeing vehicle compute resources for other applications.

2 Network Processing Accelerator (NPAC)

Texas Instruments' (TI) network processing accelerator (NPAC) is a high-performance networking processing engine that offloads networking and security tasks from the main processing cores. The NPAC consists of dedicated hardware accelerators, memory, look-up tables, and programmable RISC-V cores. This combination of deterministic hardware IP blocks and programmable cores enables high-throughput, low-power, low-latency packet processing that can support a wide range of networking applications.

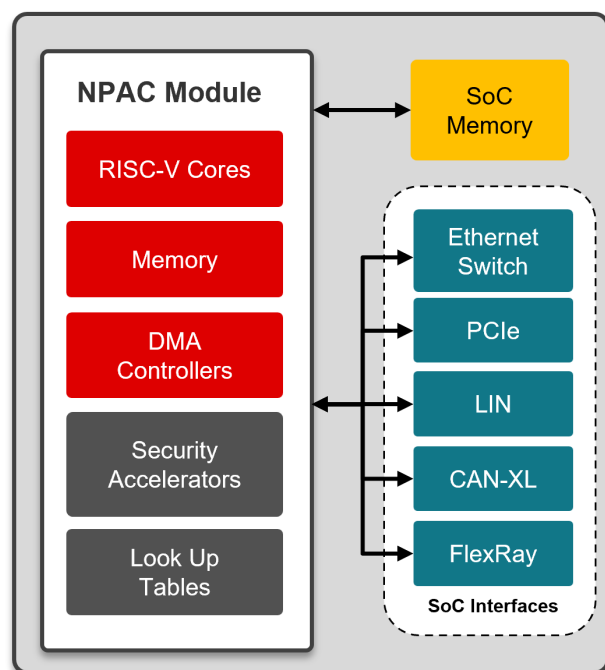


Figure 2-1. TDA54x NPAC Block Diagram

NPAC features or applications:

- Process millions of packets per second
- L3 (IP), L4, and VLAN routing and gateway support
- Security offload (IPSec, SSL, TLS, SecOC, and so forth)
- Firewall, access control list (ACL), and packet inspection
- Packetization and depacketization offload
- Pre-defined, customizable code examples
- Graph-based programming framework

2.1 NPAC Architecture

The NPAC executes a run-to-completion vector packet processing (VPP) model that breaks networking tasks into a series of functions or steps called nodes. Each node is assigned to a hardware accelerator or RISC-V core for processing by the scheduler. When a node completes its assigned task, it signals back to the scheduler, which, along with the RISC-V cores, orchestrates the next series of tasks.

Breaking up networking processing into a series of independent nodes has several advantages over processing a single packet from start to finish. Advantages include lower latency, higher throughput, parallel packet processing, separation of control and data planes, and support for modular programming through a library of predefined networking functions.

Typical NPAC applications include IP, layer 4, and VLAN routing, gateway (for example, CAN to Ethernet network bridging), packetization and de-packetization, packet encryption, decryption, authentication, and firewall services. Many of these networking applications can be executed completely independent of the SoC processing cores.

The NPAC connects to the Ethernet switch, PCIe controller, CAN controllers, and SoC memory through a dedicated low-latency packet bus. Dedicated hardware IP blocks handle all traffic flow independent of the SoC cores, eliminating interrupts and minimizing packet latency.

The NPAC uses an intuitive graph-based programming GUI to configure operation. To program the NPAC, the user selects from a list of pre-defined networking functions (nodes), linking them together in a linear fashion. Each node can be easily configured, and new, custom nodes can be added to the library, allowing the NPAC to support custom networking applications. For advanced applications requiring more precise programming, the NPAC can be programmed directly in C.

2.2 NPAC Use Case Examples

Figure 2-2 shows an access control list (ACL) filter implemented on the NPAC. An ACL filter permits or denies traffic entering or leaving a network interface based on a set of predefined security rules. In this example, Ethernet traffic from the on-chip Ethernet switch is sent to the NPAC through the dedicated, low-latency packet bus. The traffic is then depacketized into IP frames, authenticated and decrypted, and run through an ACL filter

to allow only authorized packets through. A routing lookup is then performed, and the packet is routed to the SoC processing cores for additional processing or sent out through the Ethernet network.

Figure 2-3 shows a CAN packet protocol data unit (PDU) routing function commonly used in zone controller gateways. Incoming CAN packets are decrypted, authenticated and then routed based on an AUTOSAR PDU routing table.

In both the ACL filter and PDU routing examples, all packet processing is done without the support from the SoC processing cores.

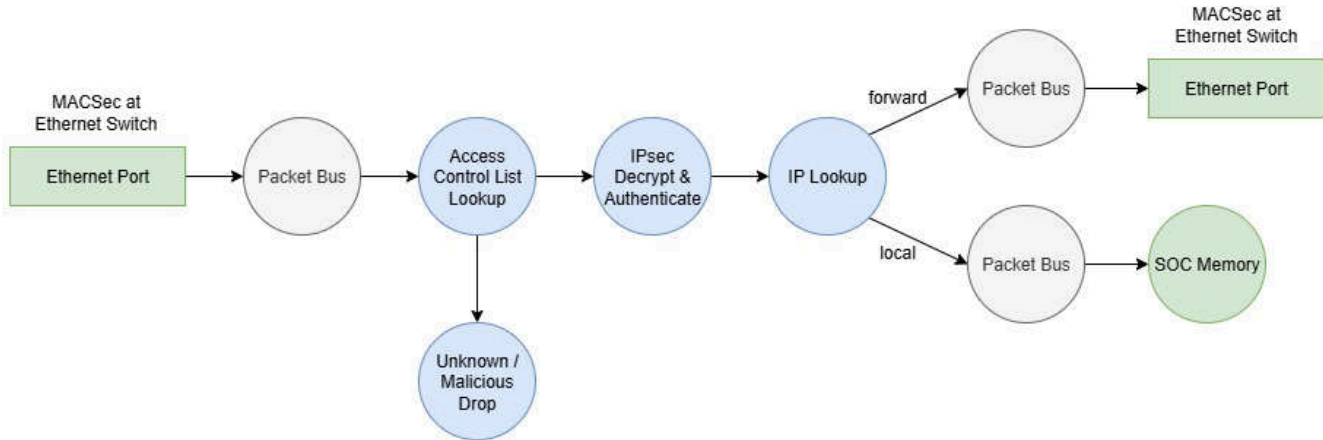


Figure 2-2. NPAC ACL Filter Implementation

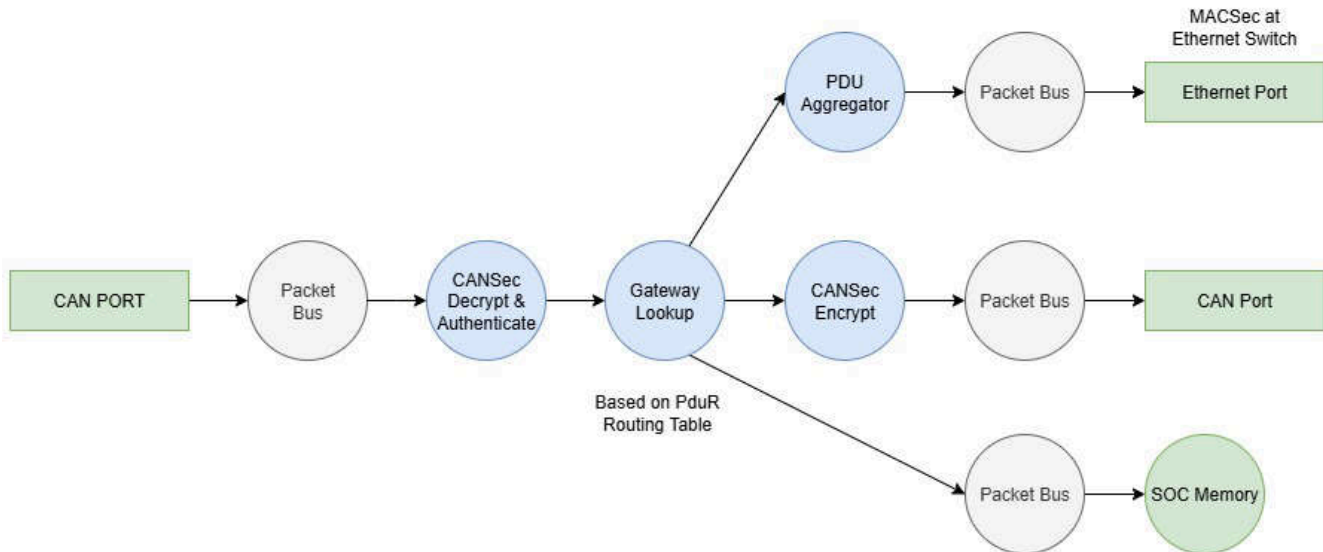


Figure 2-3. NPAC CAN PDU Routing Implementation

2.3 NPAC Features and Benefits

Table 2-1. Features and Benefits

NPAC Feature	Comment
Processing offload	Networking functions are offloaded to the NPAC, freeing up SoC cores to perform other tasks.
Multi-function	Support multiple functions, including L3 (IP), L4, and VLAN routing, gateway, packetization, depacketization, encryption, decryption, authentication, and firewall filtering.
Programmable	Combination of dedicated hardware accelerators and programmable RISC-V cores allows customers to easily configure and customize networking functions.
Low power low latency	Fixed function hardware and high-speed packet bus reduce power and latency.

Table 2-1. Features and Benefits (continued)

NPAC Feature	Comment
Packet routing	L3 (IP), L4, and VLAN routing offload.
Packetization and de-packetization	Packetization and depacketization offload; a direct packet bus connection to system memory reduces latency and increases throughput when SoC processing is required.
Firewall and packet filtering	Supports access control list (ACL) filtering and packet inspections, routing traffic to authorized areas only.
Encryption, decryption, and authentication	Dedicated, low-latency cryptographic accelerators support CANsec, IPsec, TLS, SecOC, and other cryptographic standards.
Graphic based programming GUI	An intuitive, easy-to-program graphical model enables customization of NPAC functionality. A library of pre-defined and custom nodes maximizes supported use cases and decreases NPAC programming time.
Scalable architecture	Scale networking performance from zonal to central compute gateways on a common hardware architecture maintaining code compatibility.

3 Summary

In summary, the adoption of new E/E architectures, such as zonal networks, to support next-generation vehicle technologies is increasing network performance and security requirements. Programmable hardware-based networking accelerators, such as TI's NPAC, are the most efficient method of meeting the performance requirements while freeing up compute resources for other tasks. TI's NPAC architecture scales based on networking requirements and can be found in the TDA5x Jacinto™ family of high-performance ADAS processors as well as TI's MCU zonal controllers.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you fully indemnify TI and its representatives against any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to [TI's Terms of Sale](#), [TI's General Quality Guidelines](#), or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products. Unless TI explicitly designates a product as custom or customer-specified, TI products are standard, catalog, general purpose devices.

TI objects to and rejects any additional or different terms you may propose.

Copyright © 2026, Texas Instruments Incorporated

Last updated 10/2025