SYNOPSYS®

Software-Defined Everything

Processor IP for modern compute architectures

Pieter van der Wolf

MpSoC'25, June 15-20, 2025

Legal Disclosure

CONFIDENTIAL INFORMATION

The information contained in this presentation is the confidential and proprietary information of Synopsys. You are not permitted to disseminate or use any of the information provided to you in this presentation outside of Synopsys without prior written authorization.

IMPORTANT NOTICE

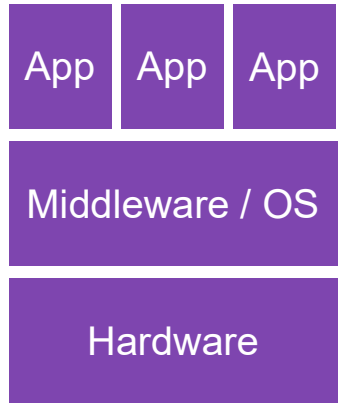
In the event information in this presentation reflects Synopsys' future plans, such plans are as of the date of this presentation and are subject to change. Synopsys is not obligated to update this presentation or develop the products with the features and functionality discussed in this presentation. Additionally, Synopsys' services and products may only be offered and purchased pursuant to an authorized quote and purchase order or a mutually agreed upon written contract with Synopsys.

Outline

- Introduction
  - Software-defined systems
- Architecture of software-defined vehicles
  - Technology shifts and changing business landscape
- SoCs for software-defined vehicles
  - And associated hardware/software development methodologies
- Processor IP for software-defined vehicles
- Conclusion

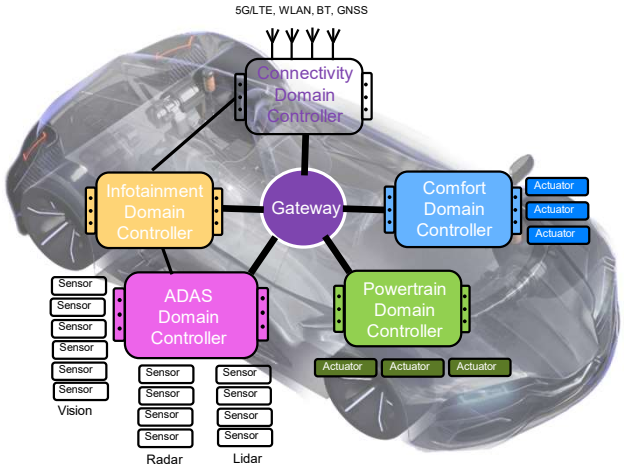
Software-Defined Systems

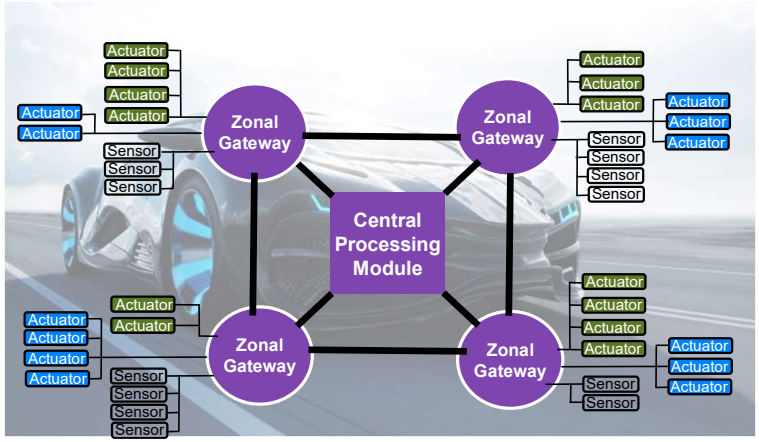
- Functionality of the system is software defined
- Using a hardware platform with standardized interfaces
  - With virtualization of hardware resources for ease of use by multiple applications
- Connected to the cloud
  - Continuous updates of features during the lifecycle of the product
- Personalization through software-based configuration of functions and features
- Enablement of new developers and new licensing & pricing models
- Examples
  - Mobile phones
  - Televisions
  - Cars



Increasingly the software needs dictate the hardware requirements

Zonal Architecture Reshaping Automotive SoCs





Yesterday
30-100+ ECUs in a car

Growing number of ECUs
Each with processing, power, wiring

Mainstream MCUs
Legacy processing technologies

Today
Domain Architecture

Consolidating ECUs into domain controllers
Reduced wiring

Integration of functions into SoCs
Typically 16nm .. 7nm

Tomorrow/Future
Zonal Architecture

Multi-Applications Central Processing
Zone controllers as local gateways
Connected by Ethernet backbone

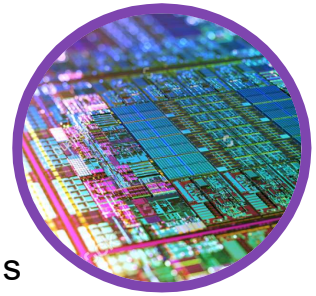
Advanced SoCs / multi-die systems
Advanced technologies, 5nm/3nm

Hardware-defined → Software-defined

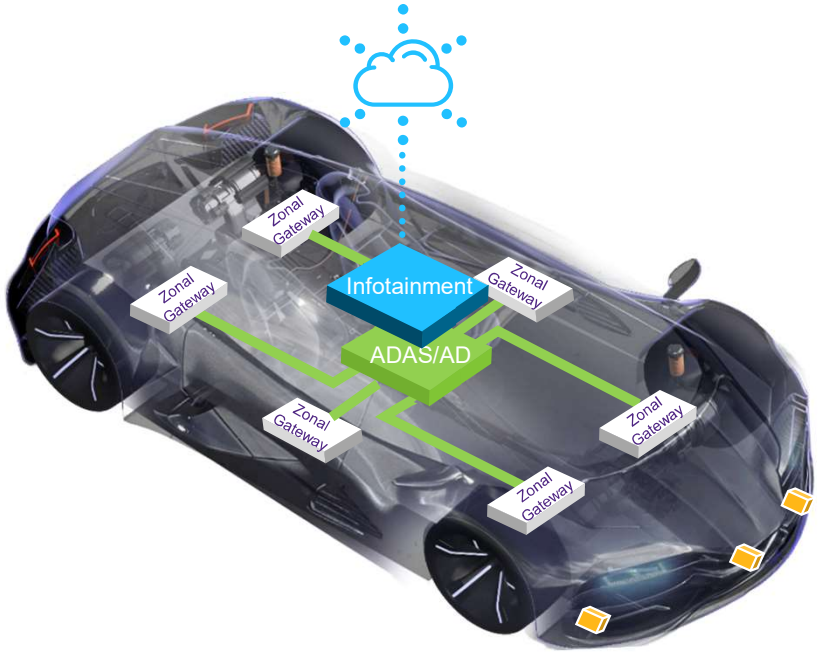
Software-Defined Vehicles

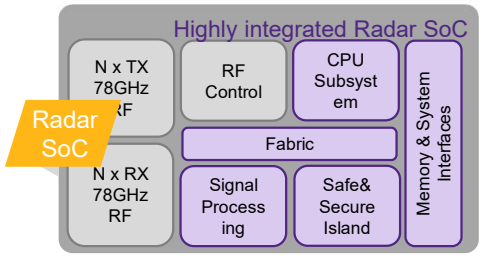
Technology shifts and a changing business landscape

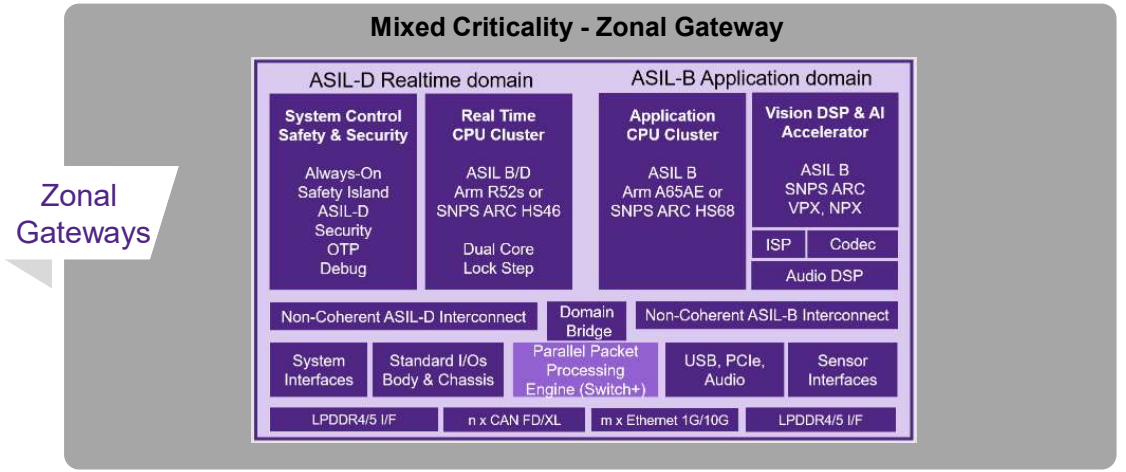
- Software-defined functions implemented on standard hardware abstractions
- Vehicle integrates into the Internet-of-Things through cloud connectivity
  - Over-the-air (OTA) updates over the vehicle's lifespan
  - Continuously deliver software updates and new functionality
- More complex software development and deployment
  - >100M lines of code, written by different companies
- More complex SoCs in advanced technologies
- Increasing safety and security challenges
  - Compliance with safety and security standards such as ISO 26262 and ISO 21434
- Accelerating time to market
- Changing business environment with new players
  - Affects roles of IP suppliers, semiconductor companies, Tier-1's, and OEMs
  - For example, OEMs doing custom SoCs optimized for their requirements and software workloads

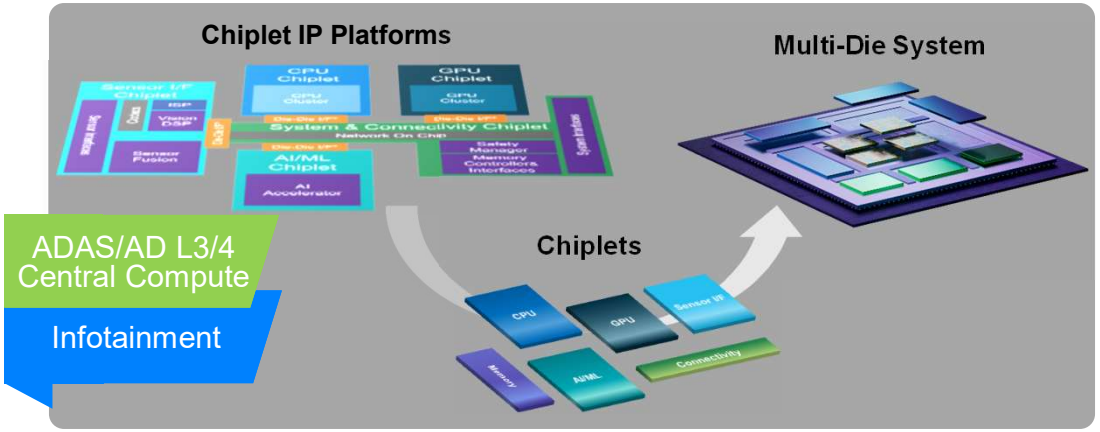


Three Key Areas Where OEMs/Tier1s Develop Custom SoCs









Automotive ADAS Heavily Reliant on Sensor Fusion

Radar

- Detect objects based on reflected microwaves
- Robust operation in different light and weather conditions
- Key component of Level 3+ and autonomous vehicles

LiDAR

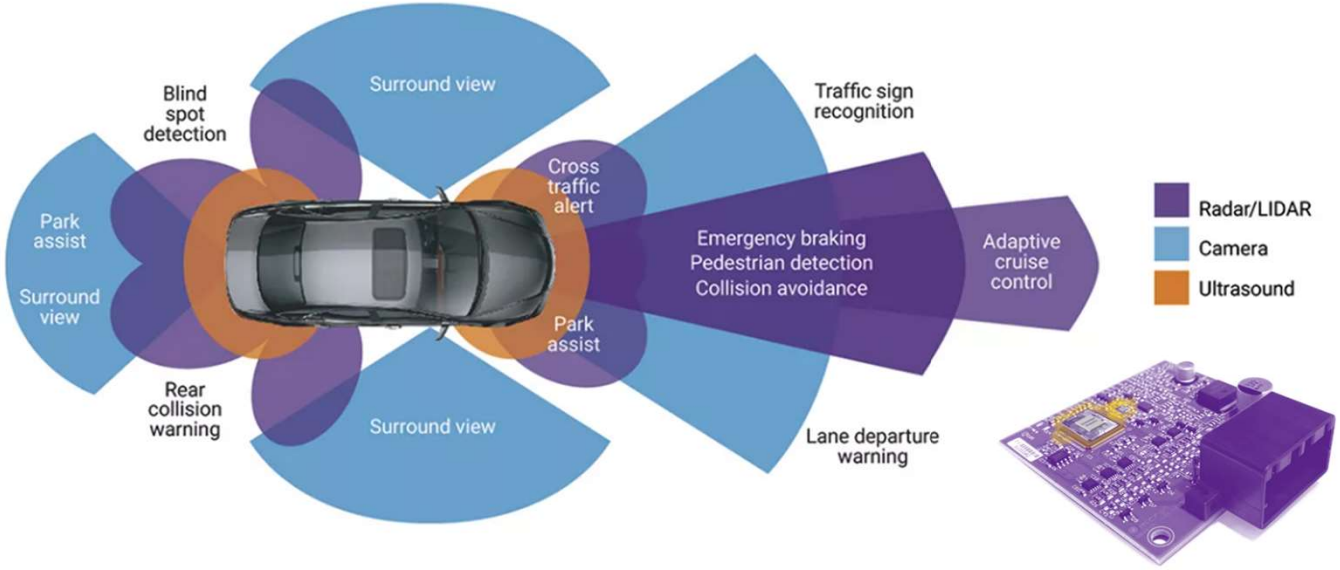
- Measures time-of-flight of reflected (laser) light
- Wide field of view with good angular resolution
- Some sensitivity to weather conditions (e.g. heavy rain)

Vision

- Analysis of camera data
- Fast and accurate object classification using AI/ML
- Good angular resolution
- Sensitive to light and weather conditions

Different technologies have complementary strengths

Automotive ADAS Heavily Reliant on Sensor Fusion



SAE Autonomy Level	Estimated Vehicle Sensors
L2+	8 - 16
L3	13 - 26+
L4	17 – 29+

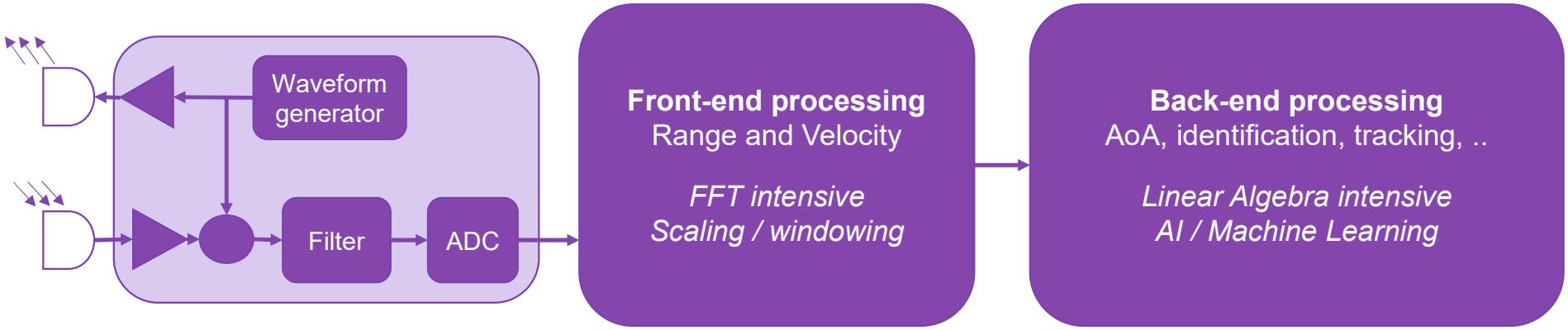
Based on survey of automobile and Robotaxi vendors including radar, camera, lidar and domain controllers

S&P Global Mobility, Autonomy Forecast, March 2024

- Sensor fusion is key to advanced ADAS
  - Combines sensors to make more accurate representations of the environment
  - Using multiple sensors together offsets weaknesses of individual sensors
- Integration challenge
  - How to validate early in the development process that the intended use cases will be correctly supported?

FMCW Radar Processing

Frequency-Modulated Continuous-Wave Radar Processing Requirements



Example RADAR case

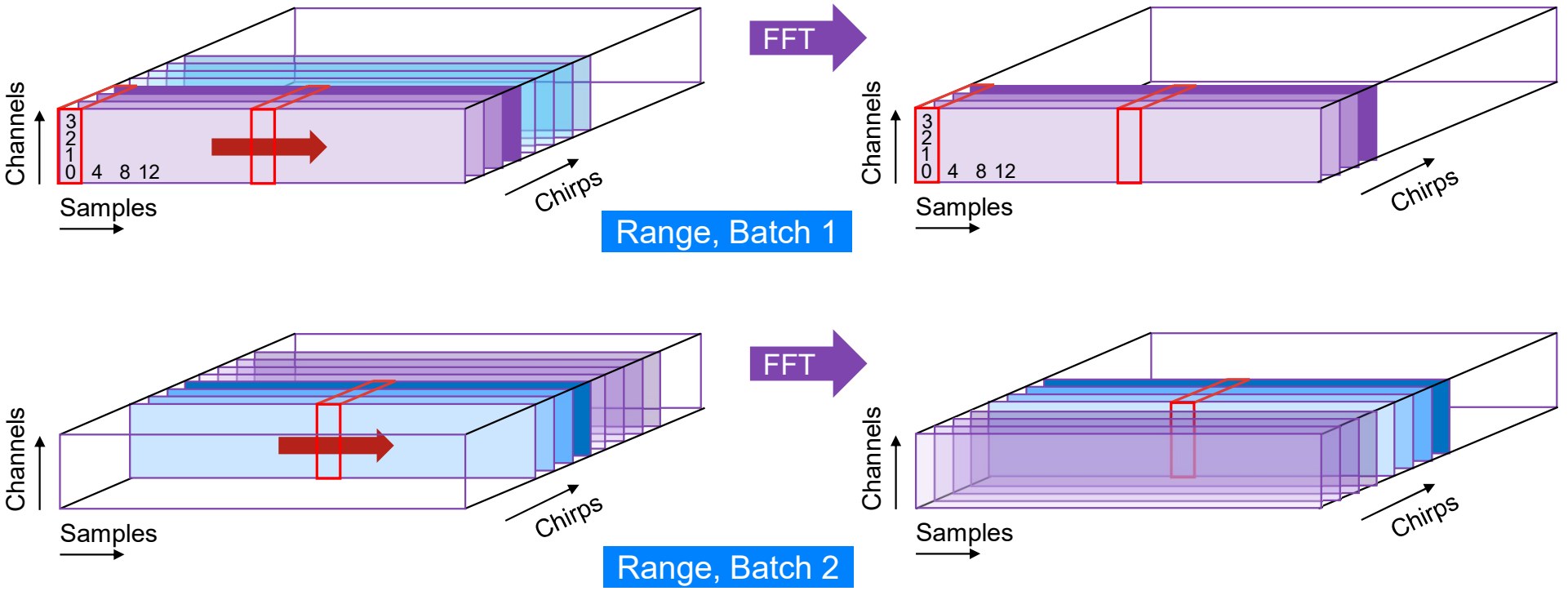
- 4 channels, 50 frames/s, 256 chirps/frame, 512 samples/chirp
- Range FFTs: 4 x 50 x 256 = 51,200 complex FFTs of length 512 per second
- Velocity FFTs: 4 x 50 x 512 = 102,400 complex FFTs of length 256 per second
- Size of radar cube with 32b+32b complex data is 4.2MB
- Bandwidth requirement for a single data stream is 50 x 4.2 = 210MB/s

52.4M FFT points per second

Few 10s MHz on DSP that does >1 FFT point per cycle

FFT Processing Example for Radar

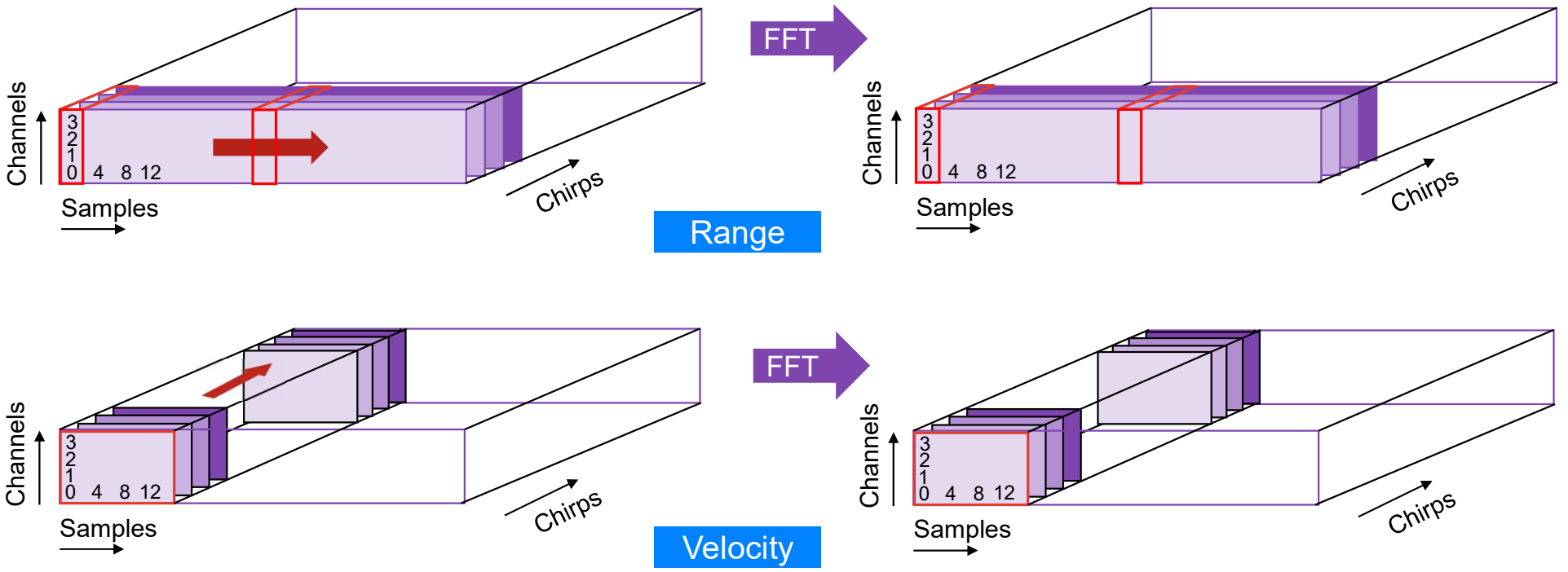
Parallel Processing and DMA Transfers for Batch-of-16 FFTs with Double Buffering



While FFT processes data using one set of buffers, DMA transfers fill/empty another set of buffers

FFT Processing Example for Radar

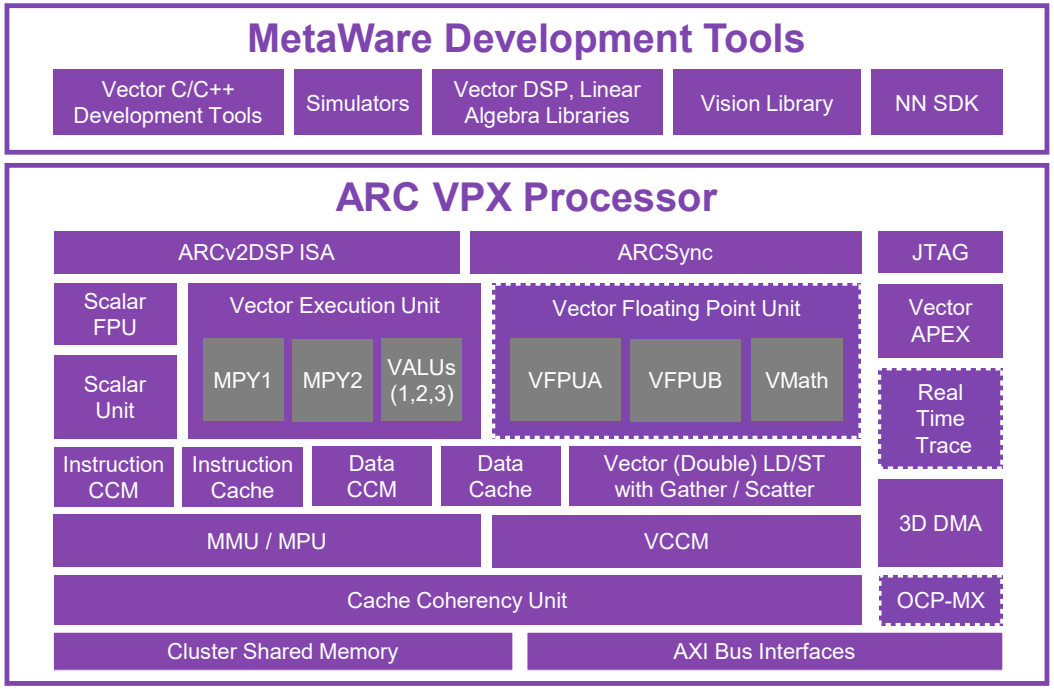
Organization of Input & Output Data in Vector Memory for Batch of 16 FFTs



Demands advanced DMA capabilities and a vector unit with flexible LD/ST access patterns

ARC VPX DSP Processor IP

Next-Generation DSP Architecture for a Data Centric World



VPX6	1024-bit vector SIMD/VLIW DSP
VPX6x2	Dual Core 1024-bit vector SIMD/VLIW DSP
VPX6x4	Quad Core 1024-bit vector SIMD/VLIW DSP
VPX5	512-bit vector SIMD/VLIW DSP
VPX5x2	Dual Core 512-bit vector SIMD/VLIW DSP
VPX5x4	Quad Core 512-bit vector SIMD/VLIW DSP

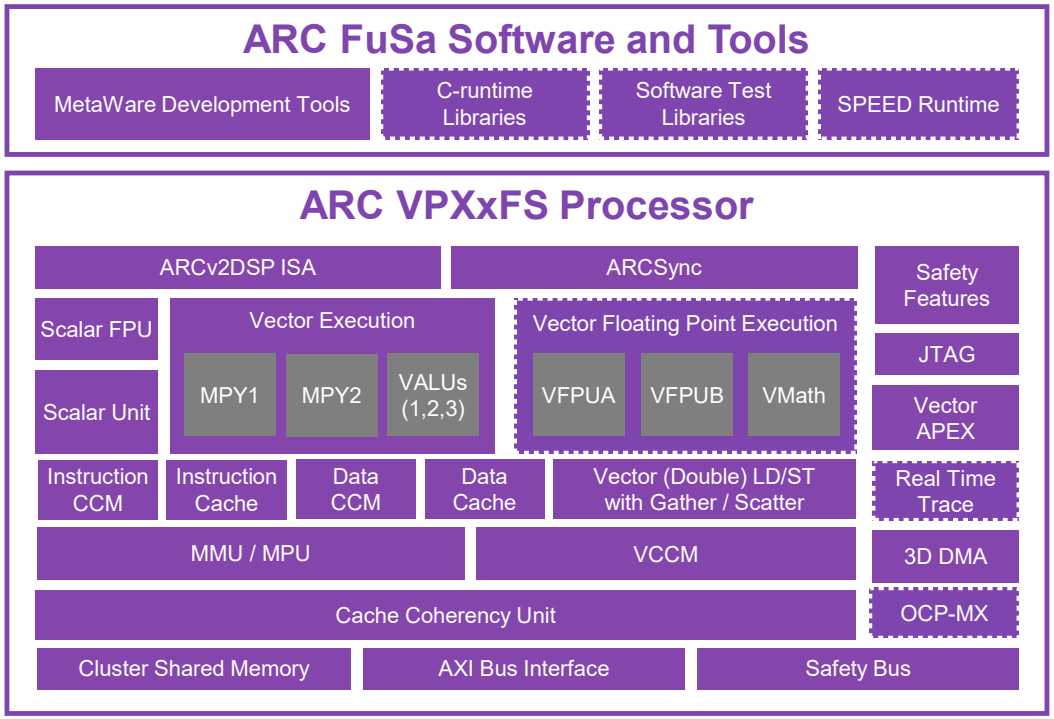
VPX3	256-bit vector SIMD/VLIW DSP
VPX3x2	Dual Core 256-bit vector SIMD/VLIW DSP
VPX2	128-bit vector SIMD/VLIW DSP
VPX2x2	Dual Core 128-bit vector SIMD/VLIW DSP

- Advanced **SIMD / VLIW DSP IP** addresses broad range of DSP workloads including RADAR/LiDAR, vision and sensor fusion
  - Vector lengths of **128-bit, 256-bit, 512-bit** and **1024-bit** enable users to select optimum PPA for required workload
  - **Scalable, configurable** and **extensible** to tune area performance and power for specific SoC requirements
- **Vector-length agnostic** programming model allows for smooth migration of S/W among members of VPX family
  - **Fully C-programmable** with optimizing compiler
- Special features for precision results on latest algorithms
  - **Ultra high-performance floating-point** processing
  - **Hardware acceleration** for DSP and linear algebra
  - Architecture, data types and software libraries optimized for **efficient machine learning**
- ISO 26262 **ASIL B/C safety** and ISO 21434 **CyberSecurity** compliant versions

ARC VPXxFS DSP Processor IP

SGS TÜV SAAR
ASIL C COMPLIANT
Functional Safety
ISO 26262
www.sgs-tuev-saar.com

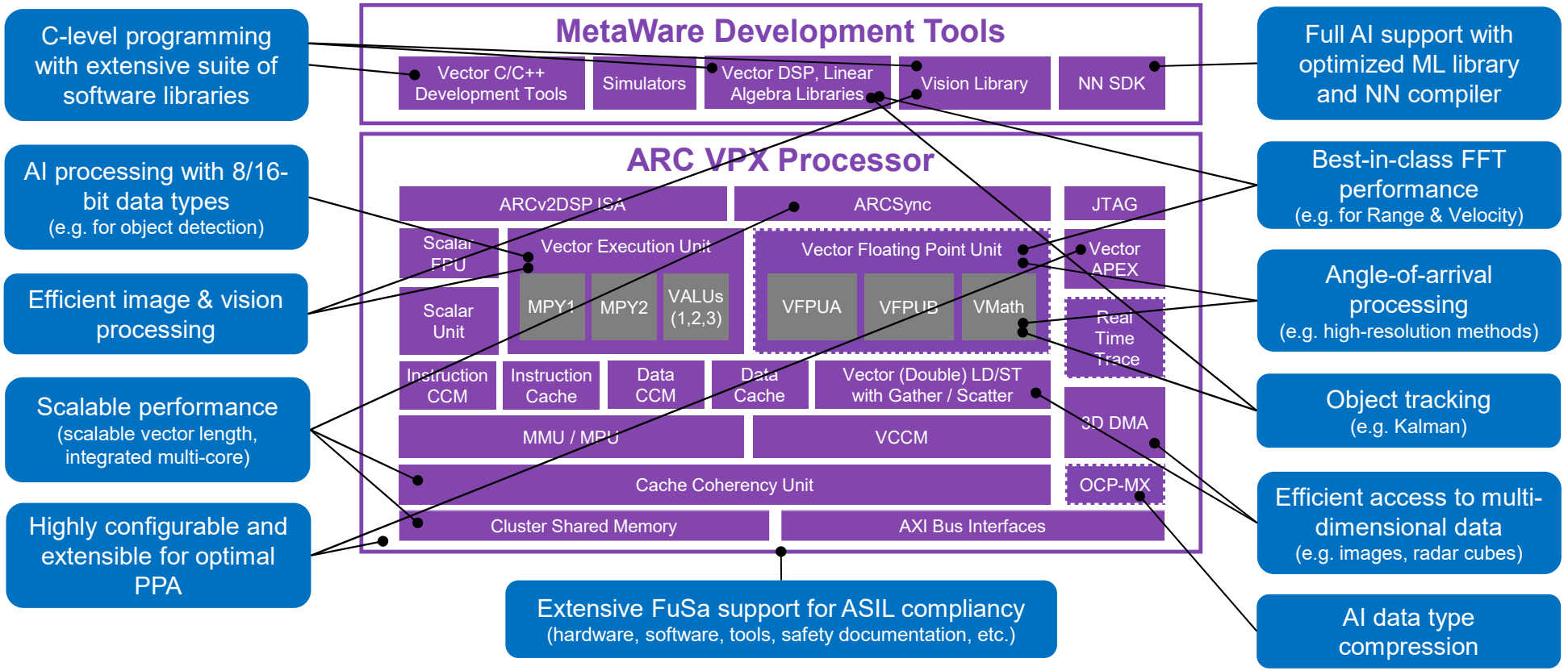
ISO 26262 Safety / ISO 21434 CyberSecurity Compliant DSP for Functional Safety Applications



- Multi-core vector DSP addresses broad range of automotive applications such as ADAS sensors (LiDAR, RADAR), powertrain, sensor fusion, etc.
- Integrated safety-critical hardware features
  - ECC protection for all memories and interfaces
  - Safety Monitor and Safety Bus
  - Lockstep capabilities
- ISO 262626 Safety Certified, ASIL-D Systematic, up to ASIL-C Random
- SAE/ISO 21434 CyberSecurity Compliant
- Safety-certified ARC MetaWare Development Tools speed ISO 26262-compliant software development
- Licensable safety-certified software libraries
- Safety documentation: FMEDA reports & safety manuals speeds functional safety assessments

ARC VPX Optimized for High-Performance Sensor Fusion

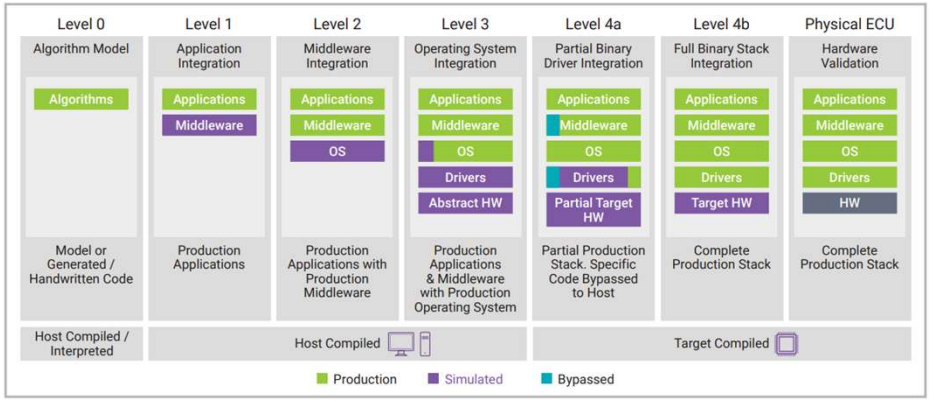
Massively Parallel Execution for Efficient Processing of Multiple Sensory Data Streams



Addressing the Hardware/Software Integration Challenge

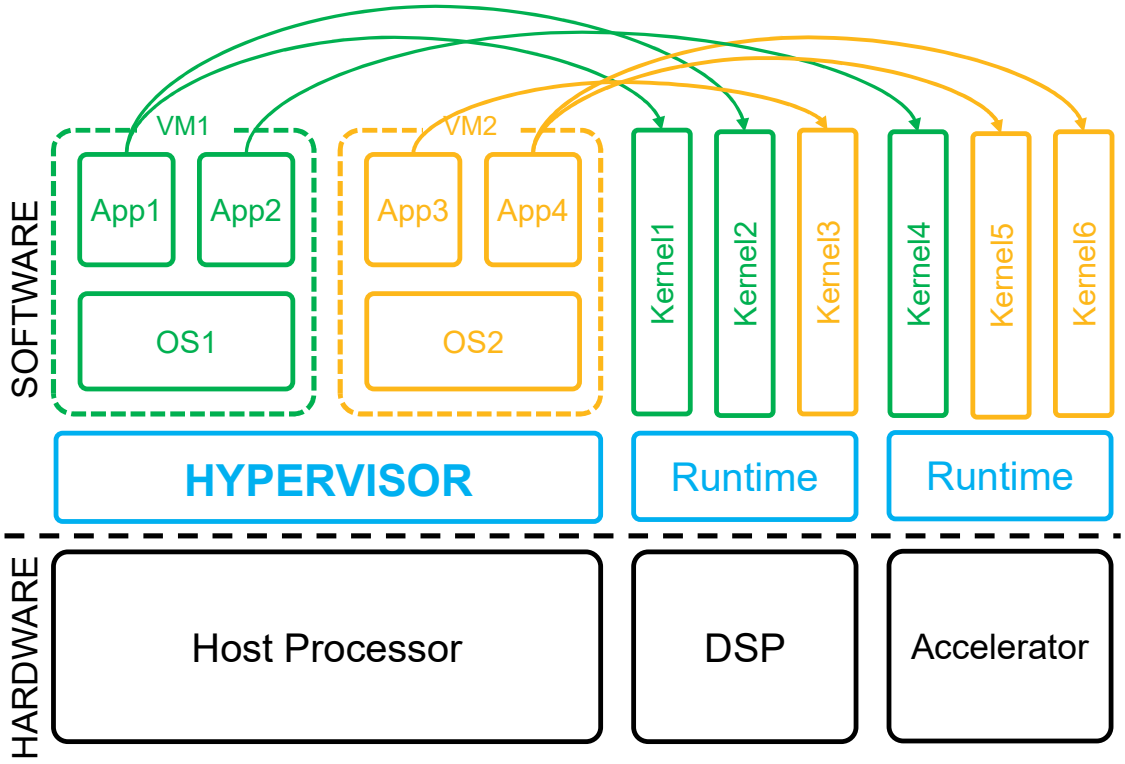
Development and Validation of Complex Software Stacks

- Digital twins
  - Enable parallel hardware and software development
  - Virtual prototypes for different levels of abstraction
  - Early software bring-up with high degree of visibility
  - Easy fault injection when validating safety and security
  - Host compiled → target compiled



- Virtualization
  - Multiple mixed-criticality software stacks running on same SoC
    - For example, advanced driver assistance systems (ADAS) and in-vehicle infotainment (IVI) functions
  - Hardware must provide separation mechanisms to support modular development
    - Allow functions to be developed, verified and validated largely separately
    - Offering mechanisms for spatial and temporal isolation
  - Virtualization is key feature of compute platform to enable freedom-from-interference
    - Supporting modular development and validation of complex mixed-criticality software stacks
  - → Software-friendly hardware

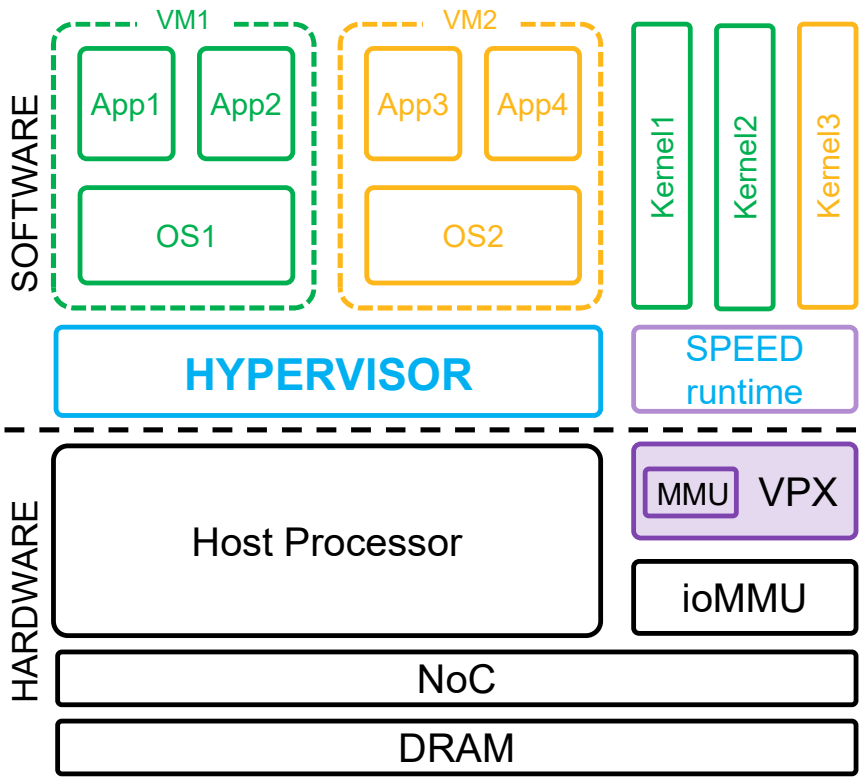
High-Level HW/SW View of SoC



- Host may run multiple operating systems
- Every operating system may run multiple applications
- Applications invoke kernels on DSPs and accelerators
- Hypervisor controls switching between virtual machines, with associated allocation of hardware resources
- Need controlled sharing of DSPs, accelerators and peripherals to achieve isolation between virtual machines

High-Level HW/SW View of SoC

Virtualization to enable freedom-from-interference



- Host sets up address maps for VMs/applications
  - Host controls ioMMUs
- Host allocates shared buffers for applications
  - Passing pointers to shared buffers to VPX
- Host triggers VM/application switches on VPX
  - Passing StreamID/SubStreamID and QoS parameters to VPX
- Device-side runtime schedules VM/application on VPX
  - Setting registers for (Sub)StreamID and QoS parameters
- Device-side runtime controls local MMU of VPX
  - For address translation supporting large physical address space
- VPX passes sideband signals on AXI interfaces
  - Sideband signals for (Sub)StreamID and QoS parameters
- VPX DMA accesses are sandboxed by ioMMU
  - With zero-copy data sharing
- Virtualization enables spatial and temporal isolation
  - While allowing hardware resources to be shared
  - Supporting modular software development
  - Enhancing safety and security

Conclusion

Trend towards software-defined systems

- And software-defined vehicles in particular

Increased dynamics in automotive

- Evolving system architectures
- New functionalities
- New development methodologies
- Changing business landscape

New requirements for processor IP for advanced automotive SoCs

- Advanced capabilities for processing and data movement
- Support for Safety and Security (hardware, software, tools, documentation, ..)
- Support for modular software development (virtualization)

Thank you