FTF FREESCALE TECHNOLOGY FORUM 2015

# Addressing NCAP 2016 Vision Processing Requirements for **Pedestrian Detection**

FTF-ACC-F1270

Allan McAuslin | ADAS Product Manager
Leonardo Surico | ADAS Systems Engineer

J U N E . 2 0 1 5

freescale™

External Use

# Your Presenters:


**Leonardo Surico**

Systems Engineer

Advanced Driver Assistance

Leonardo.Surico@freescale.com



**Allan McAuslin**

Product Manager

Advanced Driver Assistance

Allan.mcauslin@freescale.com

# Agenda

- Automotive Safety Trends / Euro NCAP
- Introduction to S32V for ADAS
- Vision Processing for Pedestrian Detection
- Questions / Discussion

# Freescale AMPG for ADAS: Trends and Solutions Map

- Sensors & Maps & V2X
- Driverless
- Fail Operational

**Fully Automated**

- Sensor Fusion & Maps
- Co-Pilot
- Dependable & Reliable

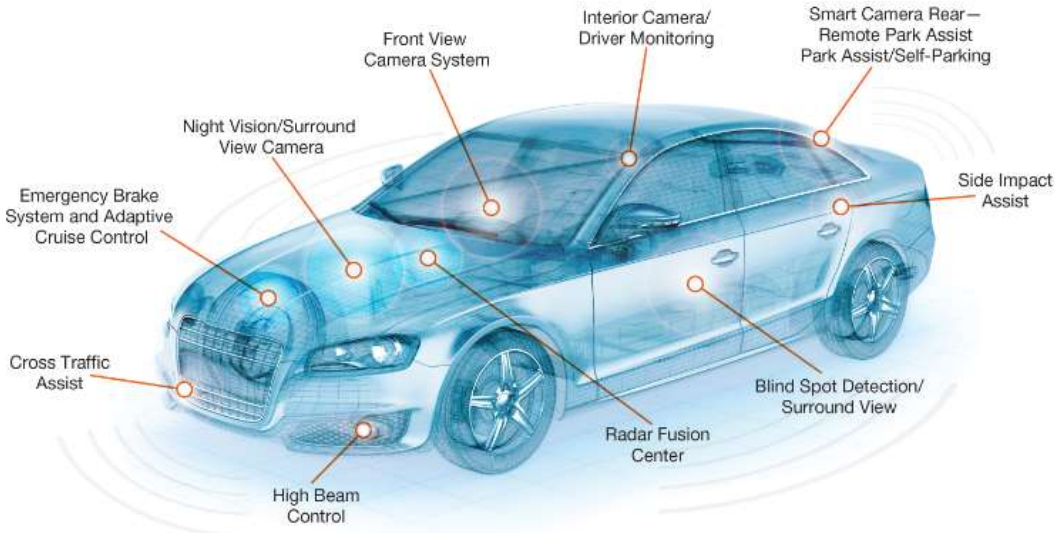
**Automated**

- Sensor
- Driver Active
- Fail Safe

**Assist**

freescale
S32 Automotive Processors

**Safe, Secure, Reliable solutions for Automotive ADAS**



# Euro NCAP Roadmap Driver / Pedestrian

## Assist

- *NCAP 2016: Lane detect, Pedestrian Detection, ACC*
- *Automotive Safety (ASIL B) as Driver is active*
- *Classic Machine Learning for Mono Front View Rear View*
- *2D/3D Surround View*

## Co-Pilot

- *Active **Steering**, Emergency **Braking**, HW platoon and self park*
- *Automotive Safety (ASIL B - C) with Security*
- *Optical Flow, Sensor Fusion & sophisticated classifiers*

## Automated

- *360° **Sensing**; 3D high accuracy environmental model*
- *Fully automated Vehicle & Fail operational System*
- *Deep Learning and advanced Machine Vision with integrated V2X*

**Driver protection**

**AEB City**

**Front + Side Update**

**Farside Occ protect**

**Pedestrian protection**

**Bonnet**

**Flex bumper**

**Upper Leg**

**AEB pedestrian**

**AEB cyclist**

2013 2014 2015 2016 2017 2018 2019

Source: Euro NCAP / Freescale

# S32V: Target Applications

freescale
S32 Automotive Processors

*The first automotive vision SoC with the requisite reliability, safety and security measures to automate and 'co-pilot' a self-aware car.*

## Applications

Front Camera

Rear Camera

360° Surround View

Sensor Data Fusion

**Freescale's S32V234:**

- *Designed for but not limited to ISO26262 ASIL B*
- *Hardware security encryption to protect against malicious hacking*
- *Designed to exacting automotive requirements*
- *Manufactured for robustness and long term automotive reliability*

# Freescale S32V Processors: Building the Foundation

- Simplify The Experience

"It can take up to 50-man-years to move my ADAS vision application from one HW platform to another...."

Freescale customer

***Key Freescale Eco-System Partnerships:***

**IP** — cognivue CORPORATION

- Announced in 2012
- Partnership to deliver image processing IP and software
- Enabled by OpenCL

**RTOS Tools** — Green Hills SOFTWARE

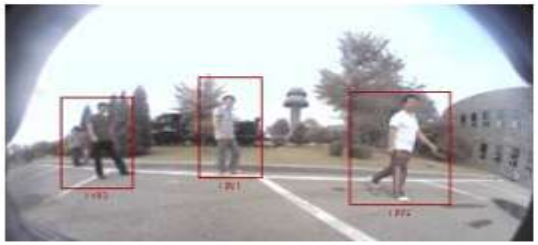
- Announced May 2014
- Partnership to deliver RTOS & Toolchain
- Dependable, Reliable, Predictable

**Soft-ware** — Neusoft Automotives

- Announced May 2014
- Partnership to deliver algorithms, demo's and full vision applications

# S32V ADAS Demo Apps

## ADAS Demo Apps



Pedestrian Detection


Traffic Sign Detection

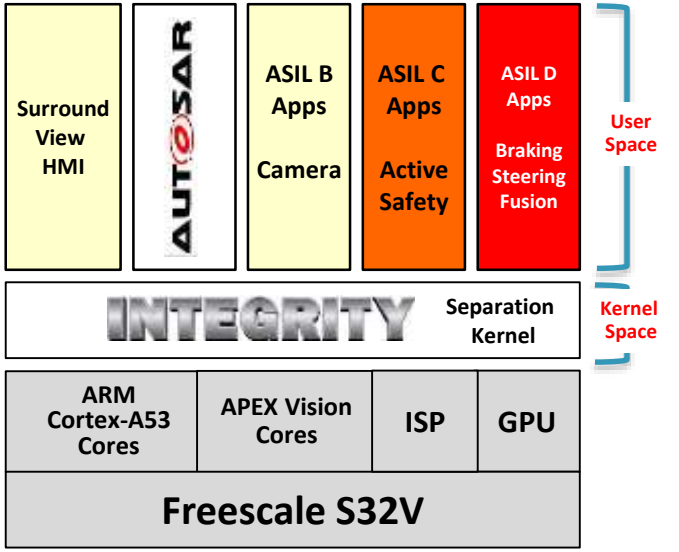
Lane Detection

**Demo App Selection:**

- Popular ADAS Apps based on camera sensor
  - Important for traffic environment sensing.
  - Based on camera sensor, cheap and efficient
  - Challenges: robustness & real-time

# Green Hills Platform for ADAS – S32V200

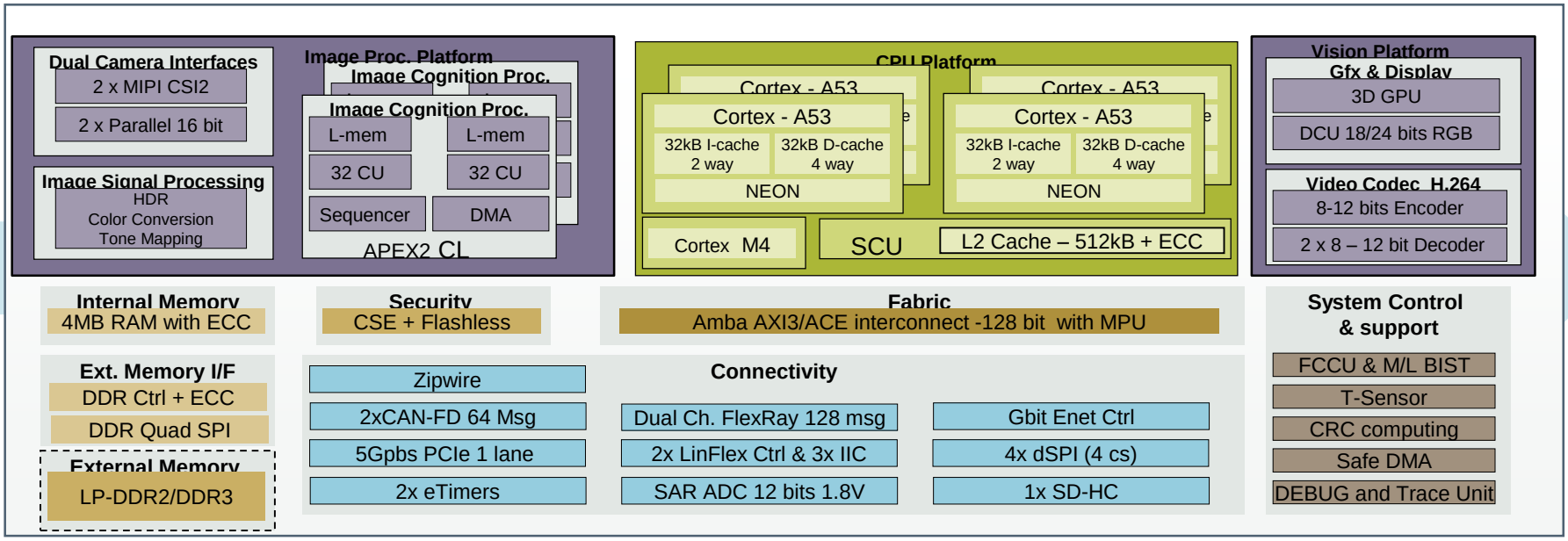
- **Proven safety and security – the world's highest safety & security certifications**
  - Experts in ISO 26262, IEC 61508, EN 50128
  - EAL6+ Common Criteria Separation Kernel Protection Profile
  - Safety OS & BSP, Certified Dev. Tools, Safety Consulting/Training
- **Trusted Execution Platform**
  - Safely isolate applications in secure partitions for guaranteed Freedom From Interference
  - Concurrently execute applications with mixed ASIL levels
  - Run AUTOSAR applications in secure partition
  - Securely run guest OSes Linux/Android on Multivisor hypervisor
- **Optimized for S32V Acceleration Units**
  - Dual APEX 2 vision processing cores
  - Image Signal Processing (ISP) core
  - 64-bit Quad ARM Cortex®-A53 + NEON SIMD unit
  - 3D GPU, OpenCV, GPGPU processing
- **Powerful 64-bit development tools**
  - High performance EEMBC® record-setting 64-bit C/C++ compilers
  - MULTI multicore debugger, TimeMachine Trace Suite
  - Code quality tools MISRA C/C++, Run-Time Error Checking and DoubleCheck™ static analyzer
- **Ecosystem**
  - Neusoft ADAS software – Pedestrian Detection, Traffic Sign Recognition, Lane Departure Warning, Surround View
  - Freescale Vision SDK integration
  - OpenGL graphics partners
  - Support for ARM® Fast Model simulator




*Mature and proven separation kernel assures Freedom from Interference*

# Vision Processing for Pedestrian Detection

# S32V234 – ADAS MCU




## Specifications:

- **CPU1-4:** ARM Cortex-A53 @1GHz, L1/L2 cache with ECC & Neon
- **CPU5:** Cortex -M4 for IO control with I/D Cache and ECC
- **ICP:** 2 x APEX2 CL (MIMD APU-64 CU each) at 400MHz
- **GPU:** GC3000 from Vivante
- **Package:** 17x17**FC-BGA**
- **Temp Range (Ta):** -40 to 105°C, 125 °C Tj, AEC-Q100 Grade 2
- **Main Supply:** 3.3V IO and 0.94V Core - external PMU + DDR rails

## Key Features:

- **F. Safety:** developed as per ISO26262 with target ASILB
- **SW Enablement:** OpenCL Tools for ICP, GPU, NEON.
- **Video Codec:** H.264 Encoder (8-12 bit) + Decoder (8-12 bit)
- **DRAM:** External LPDDR2 & DDR3 supported
- **Security:** SHE compliant Crypto Security Engine
- **Surround 3D**: 3D unified architecture. 19/38Gflops at 600MHz
- **Video dist. Network**: 2X Mipi CIS2 – 4 Virtual channels each
- **Connectivity**: Gbit Etehrnet, PCIe, FD-Can & Flexray

## Use Cases

- **Front-vision camera**
- **4 cameras Smart surround view**
- **Fusion Box**

# The Vision Pipeline

Each engine has a best efficiency on certain type of functions. To let the complete system working at highest efficiency each engine needs to work in parallel in pipeline mode.

Image sensor

Processing and Conditioning

Filtering

Edge Extraction

Transformation Feature Classification

Info Extraction

Other algos

GPU

Display

## Typical ISP functions

Black Level and Dead Pixel processing
Black level, Vignetting
Exposure control, color balance analysis
Geometric distortion corrections, chromatic aberration
HDR
De-mosaic Bayer pattern
RGB->YUV420, channel gain
Spatial de-noise
Gamma correction
Image Scaling

## Typical *APEX* functions

Whatever Kernel based filter:
Sobel X
Median
Histogram
Integral Image
FAST
BRIEF
ARB
FIR
Kirsh
Laplace filter
Gaussian
RGB to YUV
YUV 2 RGB
YUV 420 to RGB565
Harris Corner
Shi–Tomasi Corner
Lens Correction

ARM53 Neon

APEX2

ARM53 Neon

# The Vision Pipeline

Each engine has a best efficiency on certain type of functions. To let the complete system working at highest efficiency each engine needs to work in parallel in pipeline mode.

Image sensor

Other algos

- Performance
- Power
- Safety (ISO 26262)

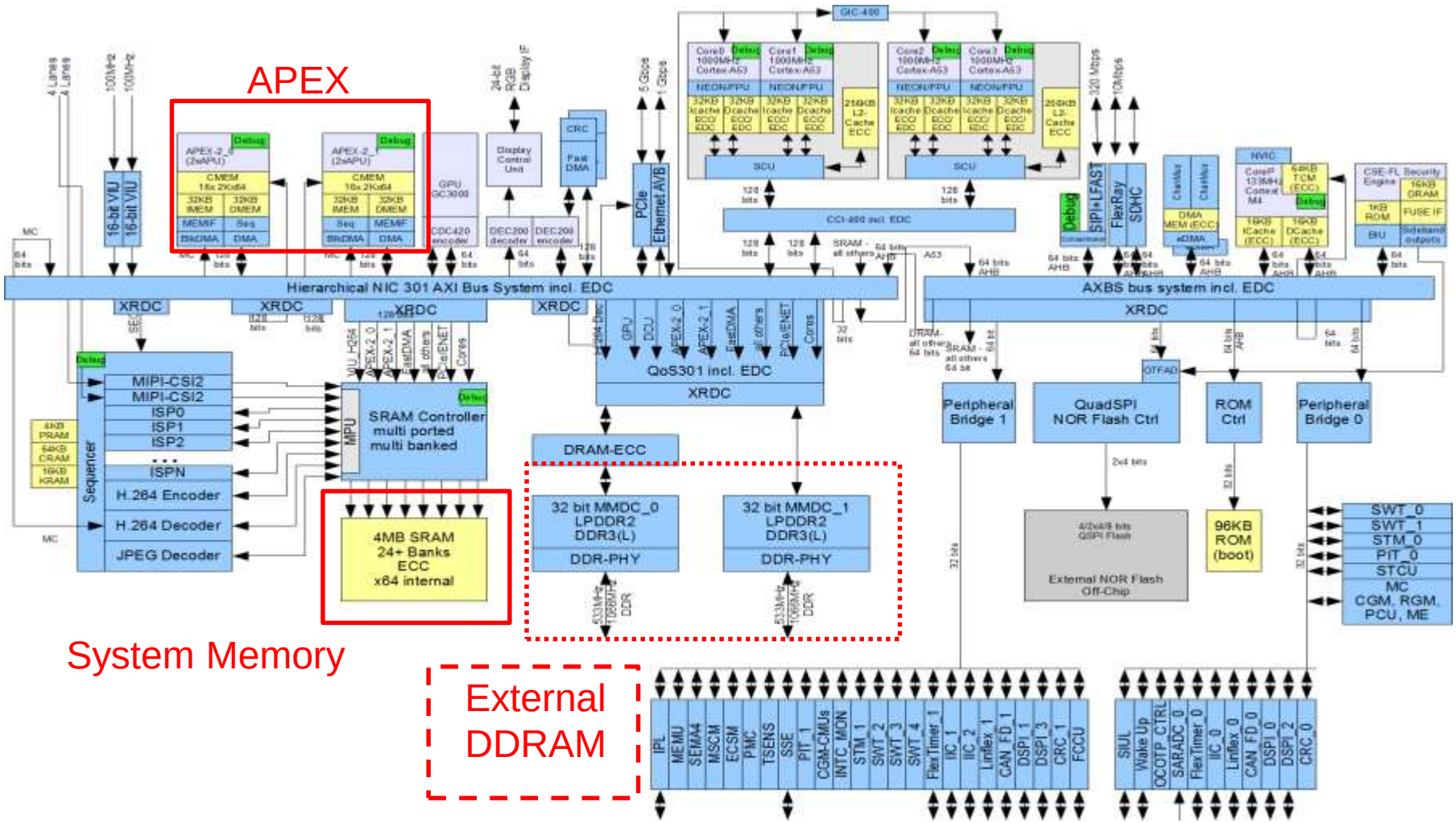
Typical ISP

Black Level and Dead Pixel
Black level, Vignetting
Exposure control, color bala
Geometric distortion correct
HDR
De-mosaic Bayer pattern
RGB->YUV420, channel ga
Spatial de-noise
Gamma correction
Image Scaling

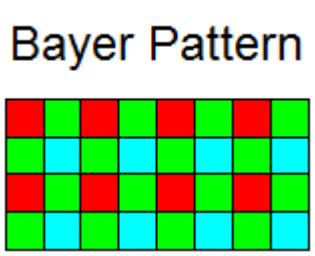
YUV 2 RGB
YUV 420 to RGB565
Harris Corner
Shi–Tomasi Corner
Lens Correction

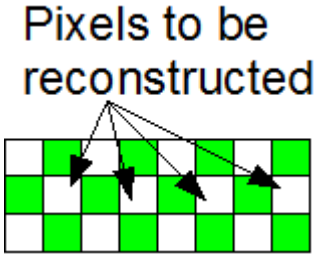
ARM53
Neon

# S32V234 Block diagram



# ISP – HDR, Tone Mapping, Color Balancing, ….

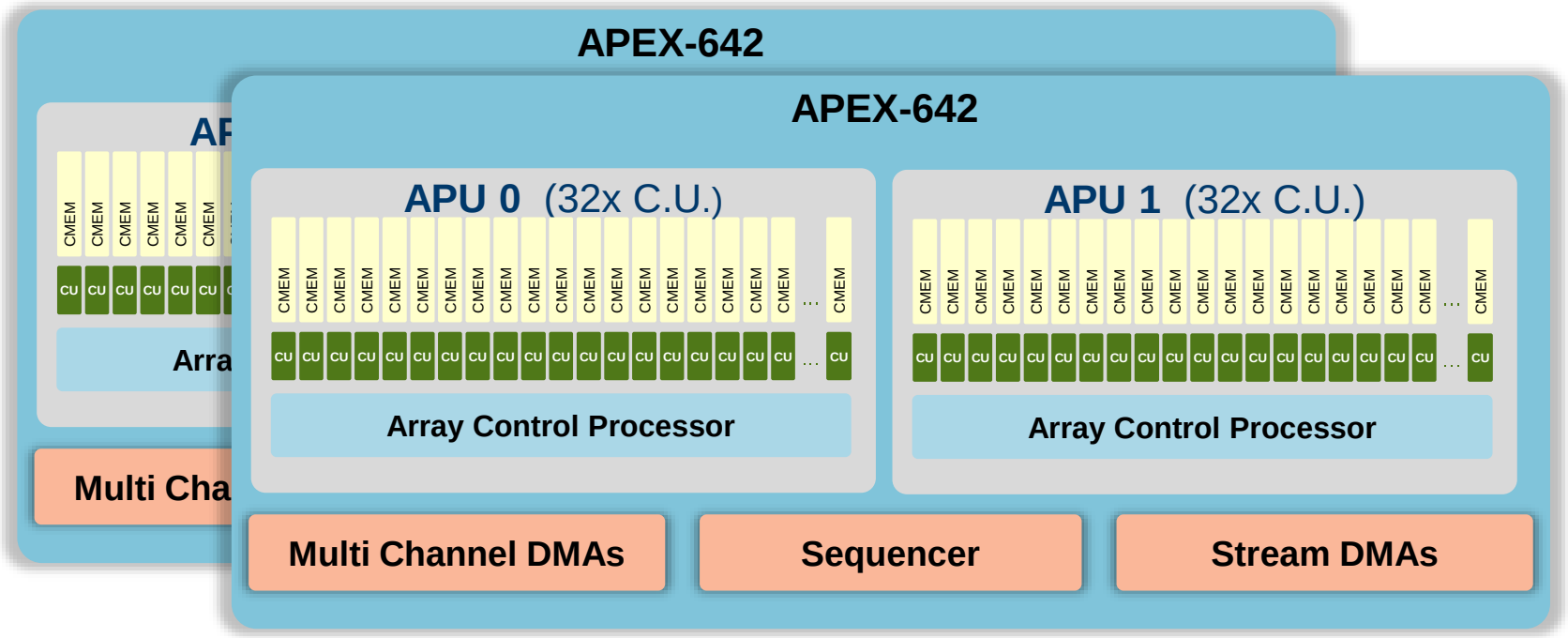




| Function | Type |
|---|---|
| Black Level and Dead Pixel processing | LUT, Linked List |
| Black level, Vignetting | 2D LUT (low res) |
| Exposure control, color balance analysis | Histogram/Stats |
| Geometric distortion corrections, chromatic aberration | calibrated per color, 2D LUT (low res), bi-linear interpolation |
| HDR | LUT, α-blending, conditional selection of exposure plane |
| De-mosaic Bayer pattern | Reconstruct missing Green values based on edge direction |
| RGB->YUV420, channel gain | Matrix multiplication,<br>factors based on Histogram |
| Spatial de-noise | Edge aware thresholding |
| Gamma correction | LUT |
| Image Scaling | Anti-alias (FIR), bi-linear interpolation |

# APEX-642 – programmable Image Cognition Processor



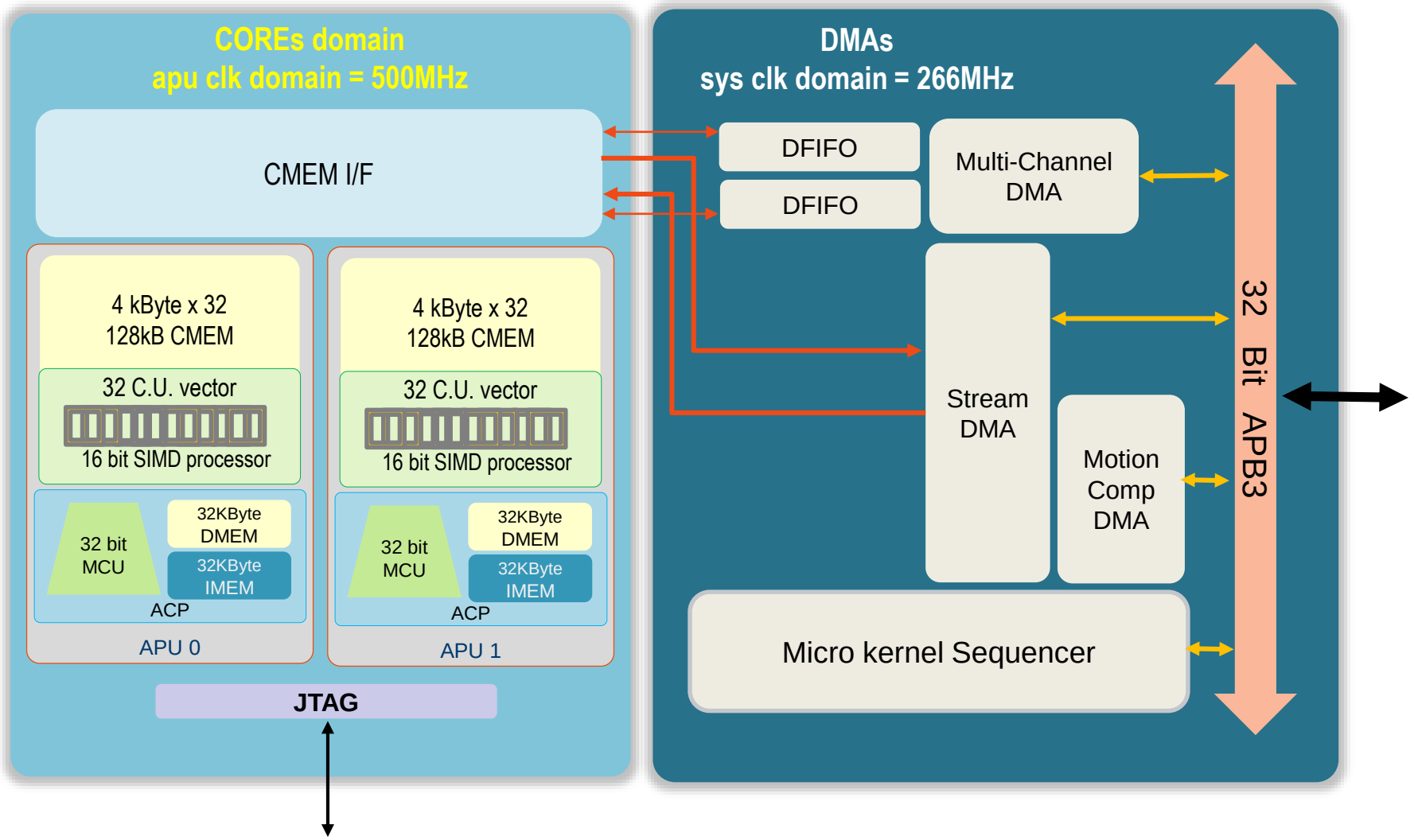

**S32V234 has 2 APEX2-642 blocks**

2 APEX-642 =

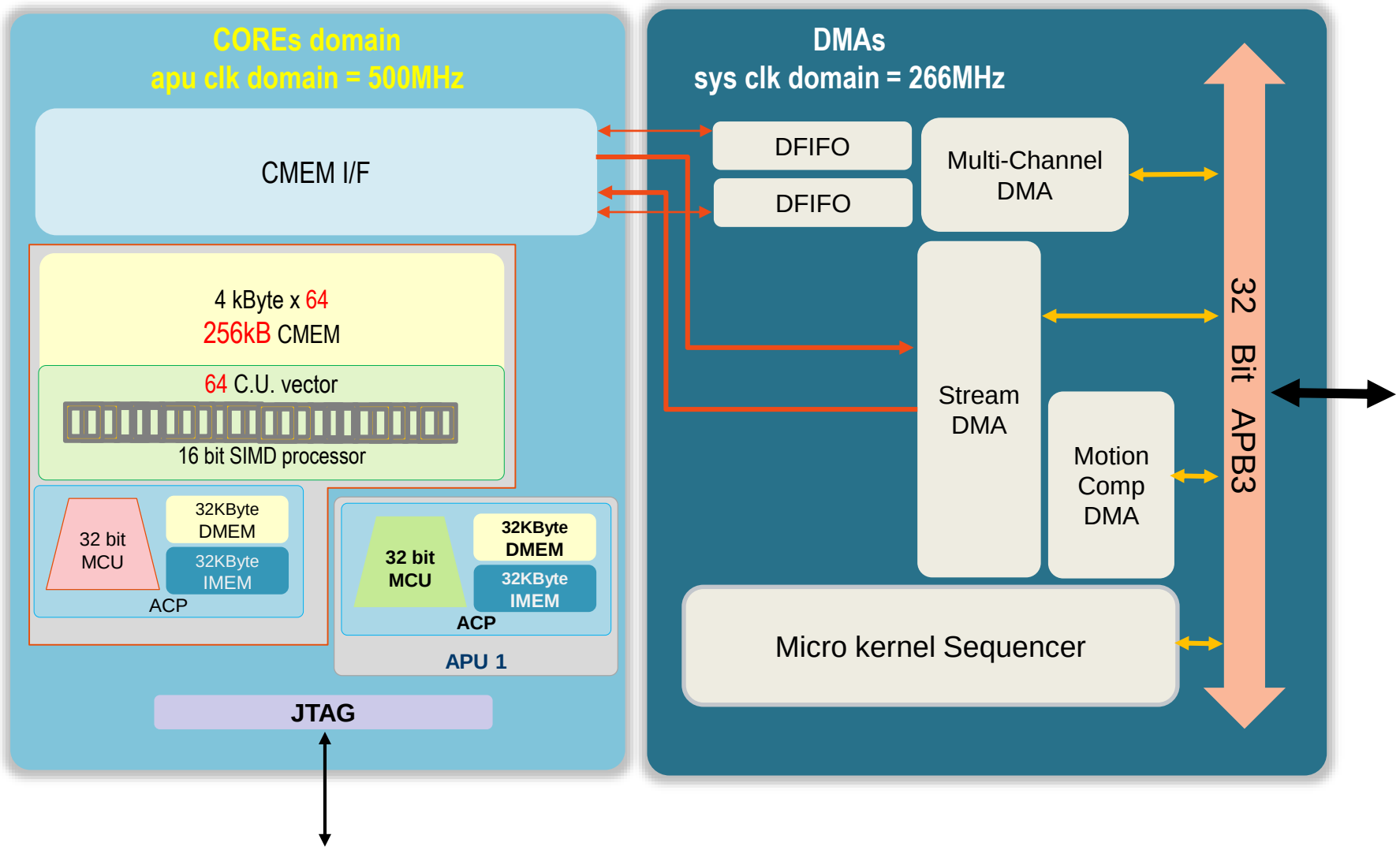
2 x (32 x 2 C. U. , 2 ACP, 256KByte ram) =

**128 C.U. 4 ACP 512KByte ram - 2 complex DMA**

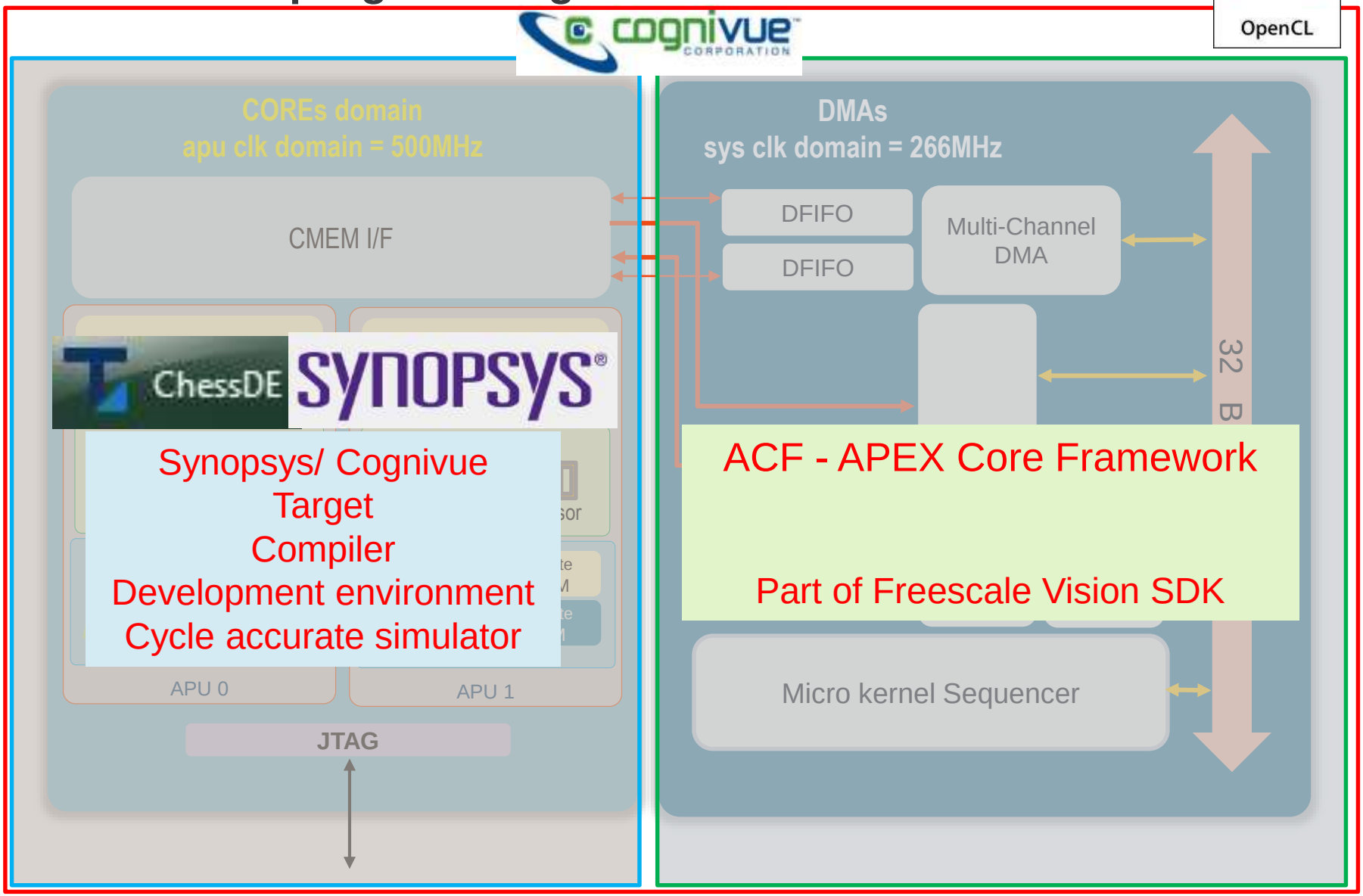
# APEX642-CL Hardware Architecture



# APU Vector Unit Sharing



# APEX642-CL programming tools



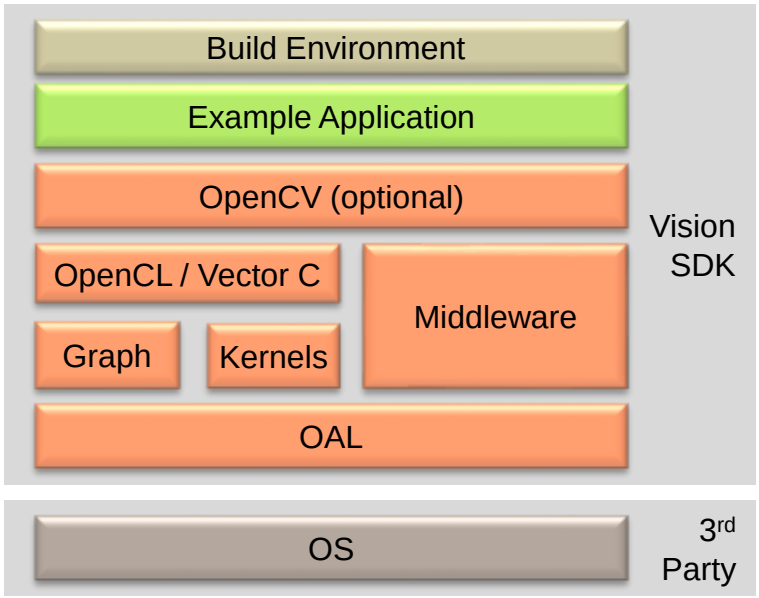
# Programming APEX is as easy as any other MCU

- C/C++ unified compiler
  - The scalar MCU is like any other MCU
  - The vector MCU has all the equivalent scalar instructions plus dedicated vector instructions (vif, vany, vall, vget, vput…)

- NO ASSEMBLER required; compiler has internal optimized inline assembler instructions.  No instruction set manual available.

- Unified C/Cpp file required for both scalar and vector MCU.
  - The compiler, based on instructions syntax, will handle the appropriate generation of scalar/vectors instructions, registers and memory references.

# Vision SDK Overview




- **Build Environment**
  - Makefile based
  - single point of build for all components
- **Rich set of demo applications including source code**
  - Face detection
  - ORB homography
  - LDW
  - And many more

- **Basic example applications including sources**
  - Histogram
  - Gaussian blur
  - Rotate
  - Image/movie capture, display control
- **Rich set of optimized kernels including source code**
  - Arithmentic: add, diff, dot division, dot sqr, max, min, ...
  - Filter: gauss, gratient, saturate, median, sobel, ...
  - Object/feature detection: haar, lbp, fast9, harris, sad
  - Geometry: bilinear interpolation, hough transform, rotate, ...
  - Full list of kernels included in Vision SDK
- **OpenCV Integration**
  - Seamless integration of HW accelerators with OpenCV (for supported OSs)
  - Integration of sensor drivers and display output
- **Middleware**
  - Driver and SW integration of Vision processing pipeline (HW accelerators, sensor I/O, display)
- **OS Support**
  - OS Abstraction layer to seamlessly support
  - Green Hills INTEGRITY and
  - Linux
  - OS Abstraction Layer allows for easy support of arbitrary operating systems

- **Not included**
  - Operating System
  - Commercial tools (compiler, debugger)

# ISO26262 Fault Prevention and Control Measures

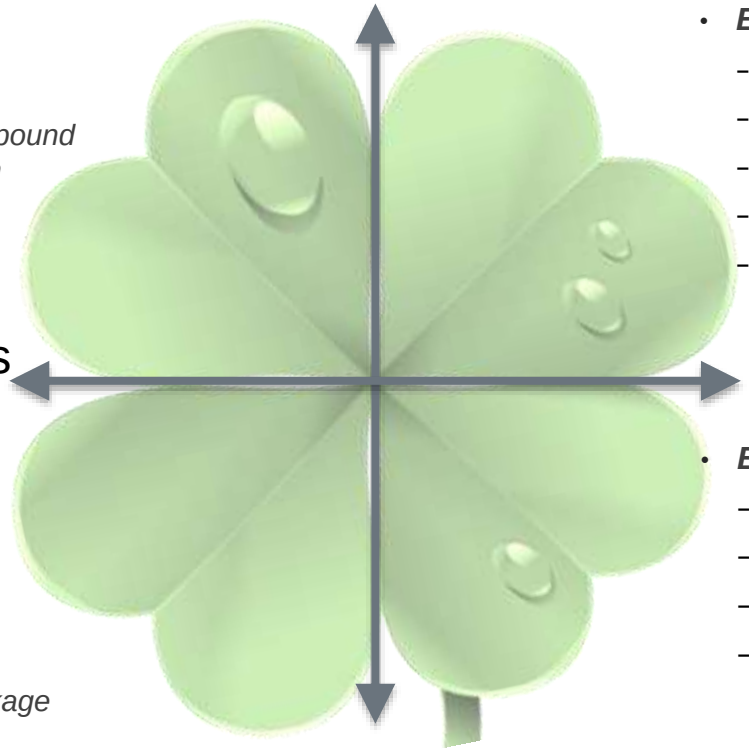
... implemented as **product** features
against **random** faults (architecture, function)

- ***Examples***:
  - *Triple voted flip flops*
  - *ECC on memories*
  - *Redundant vias*
  - *Ultra low alpha mould compound to reduce effect of radiation*

- ***Examples***:
  - *Independent compute engines – 2 x MP2 cluster*
  - *Independent checker engine – Safe State engine*
  - *EDC on memory and buses*
  - *Logic BIST/Memory BIST*
  - *Voltage/temperature monitors*

... to **prevent** faults
(robustness)

- ***Examples***:
  - *ISO Design Process*
  - *Design Margin*
  - *Process Margin*
  - *Automotive Process Package*



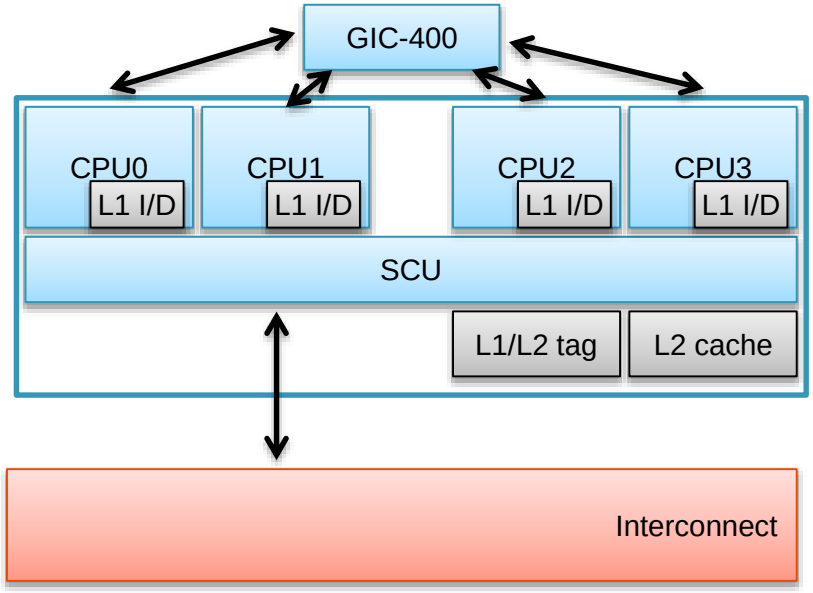
... to **control** faults
(detect and react)

- ***Examples***:
  - *Wafer level stress testing*
  - *Test in Burn-In*
  - *Iddq testing*
  - *AECQ100 qualification*

... during **develoment** and **production**
against **systematic** faults (process, procedures)

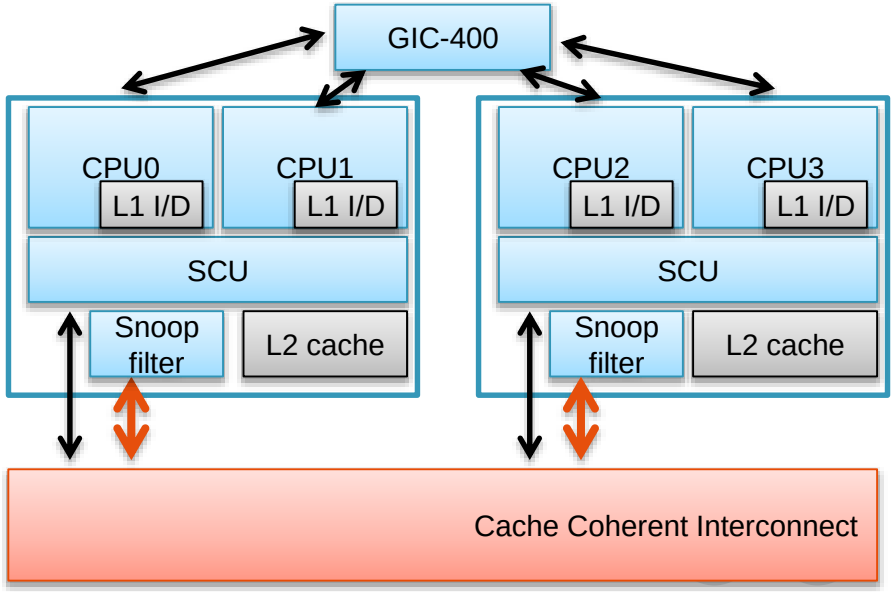
**ISO26262 can NOT be retro-fitted to a device**

# Core-cluster separation

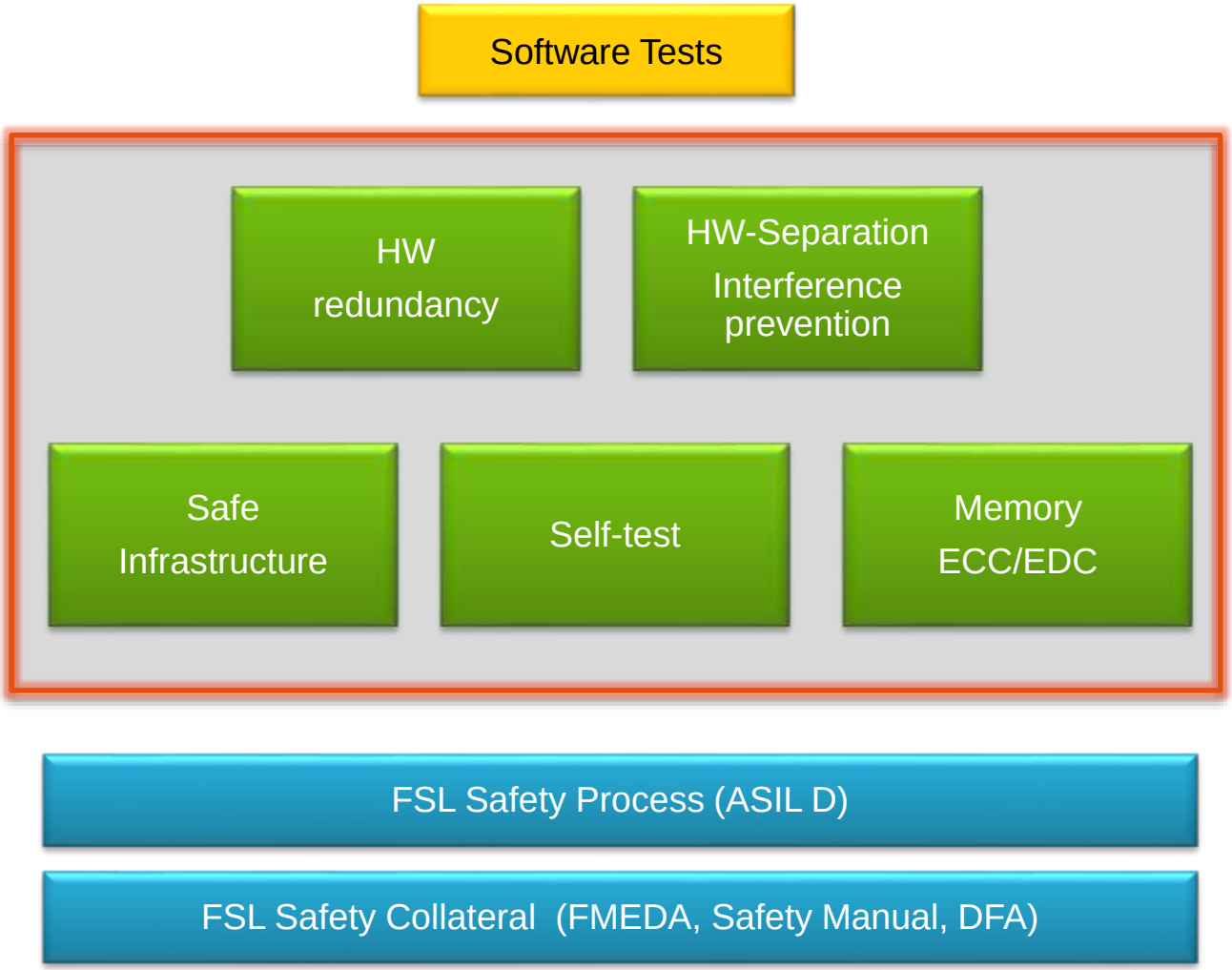



- SCU is a potential common cause of failure for all CPUs
- L1/L2 tag RAM & control failures affect all CPUs
- L2 data RAM failures affect all CPUs

- CPU can snoop data from other cluster
- Snoop can be disabled for predefined regions
- HW-separation of clusters

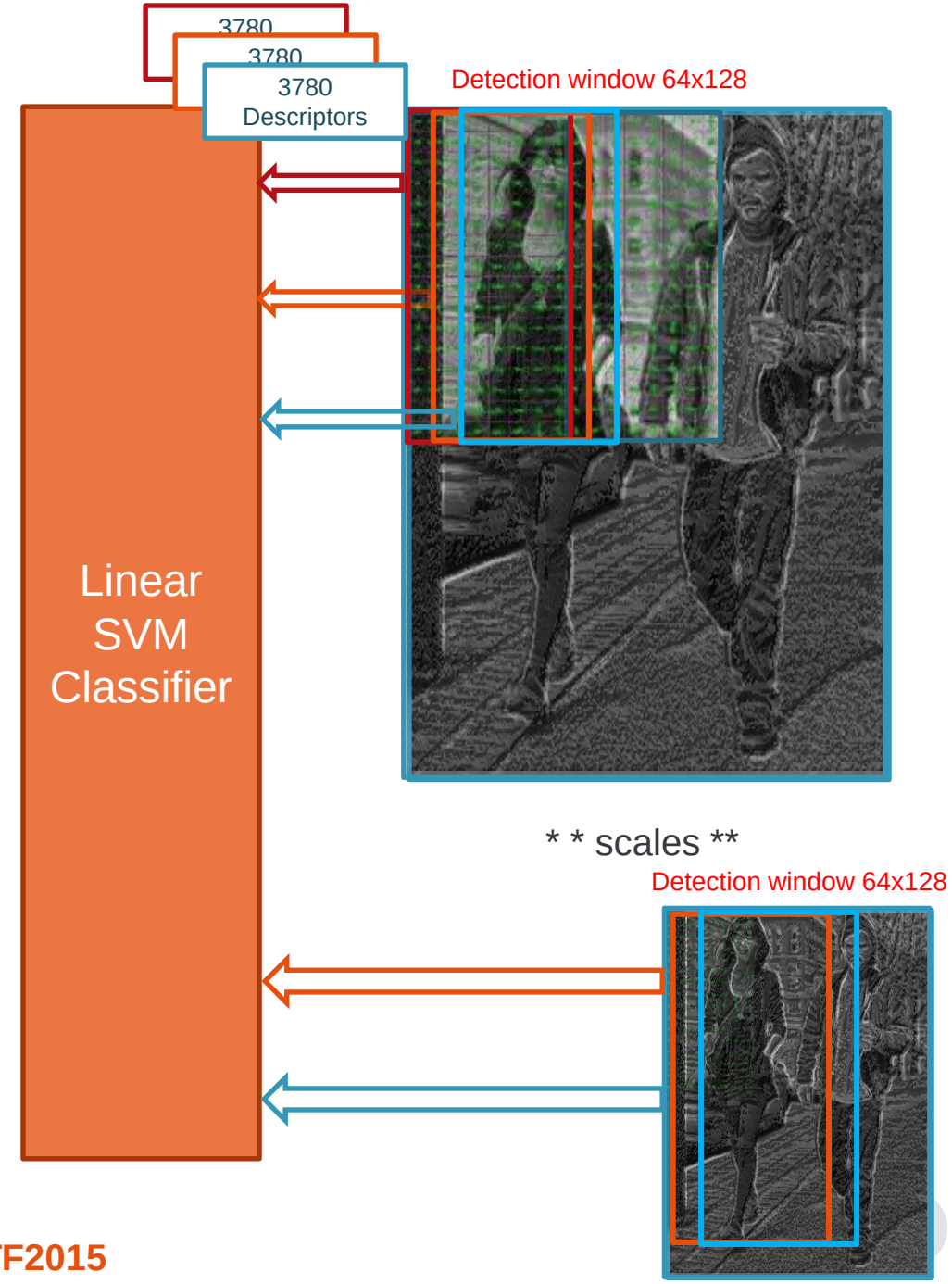


# S32V234: targeting – but not limited to – ASIL B applications



# HOG + SVM

- Detection window is moved around the image.
- HOG descriptor are collected at each detection window and
- Given to Linear SVM for classification



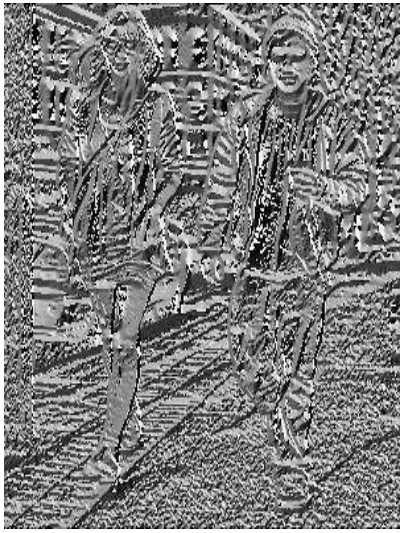
# Processing steps involved to obtain Histogram of oriented Gradients(HOG)

Source Image

Dx = Sobel X $[-1, 0, 1]$

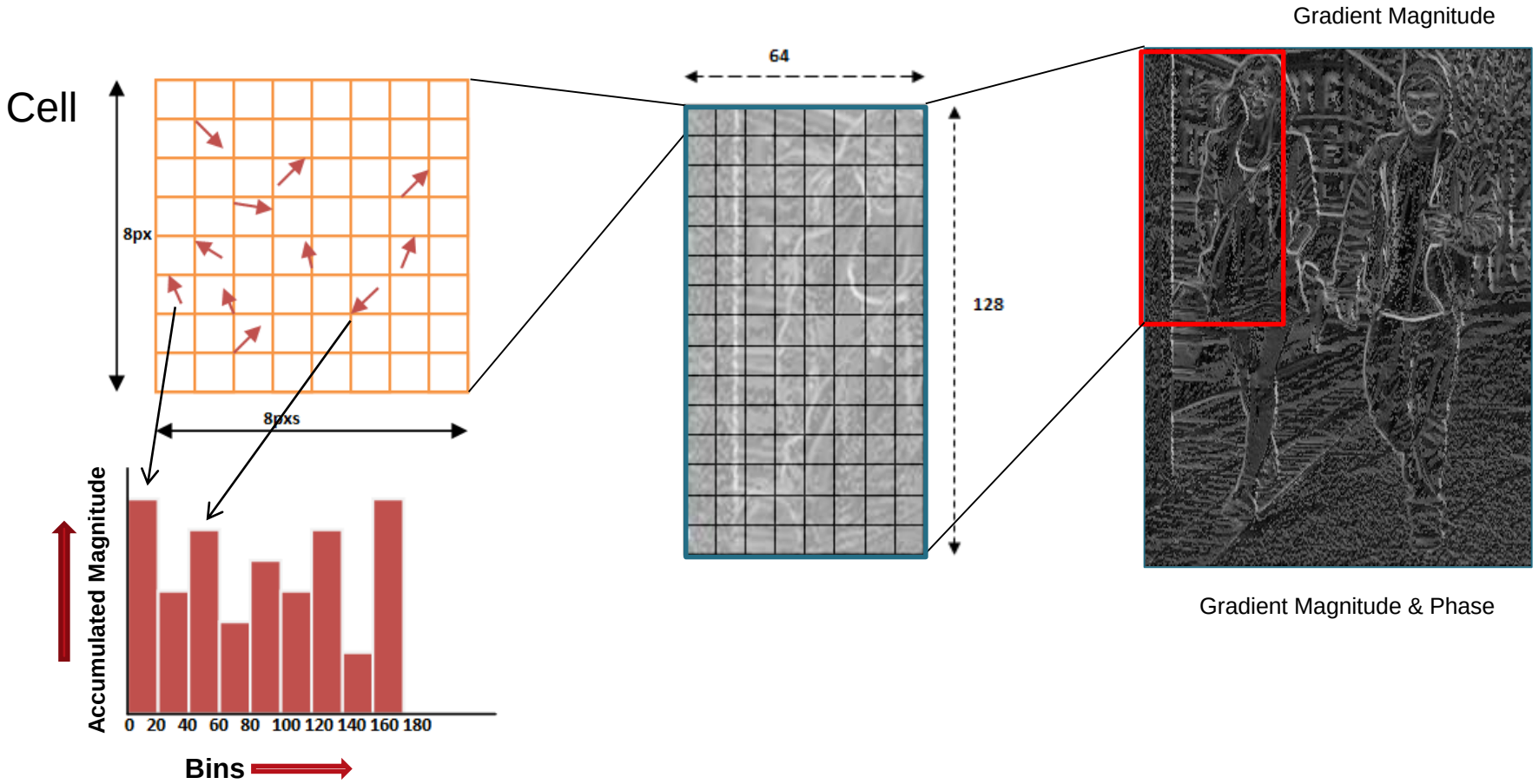
Dy = Sobel Y $\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$

Gradient Magnitude **= abs(Dx) + abs(Dy)**

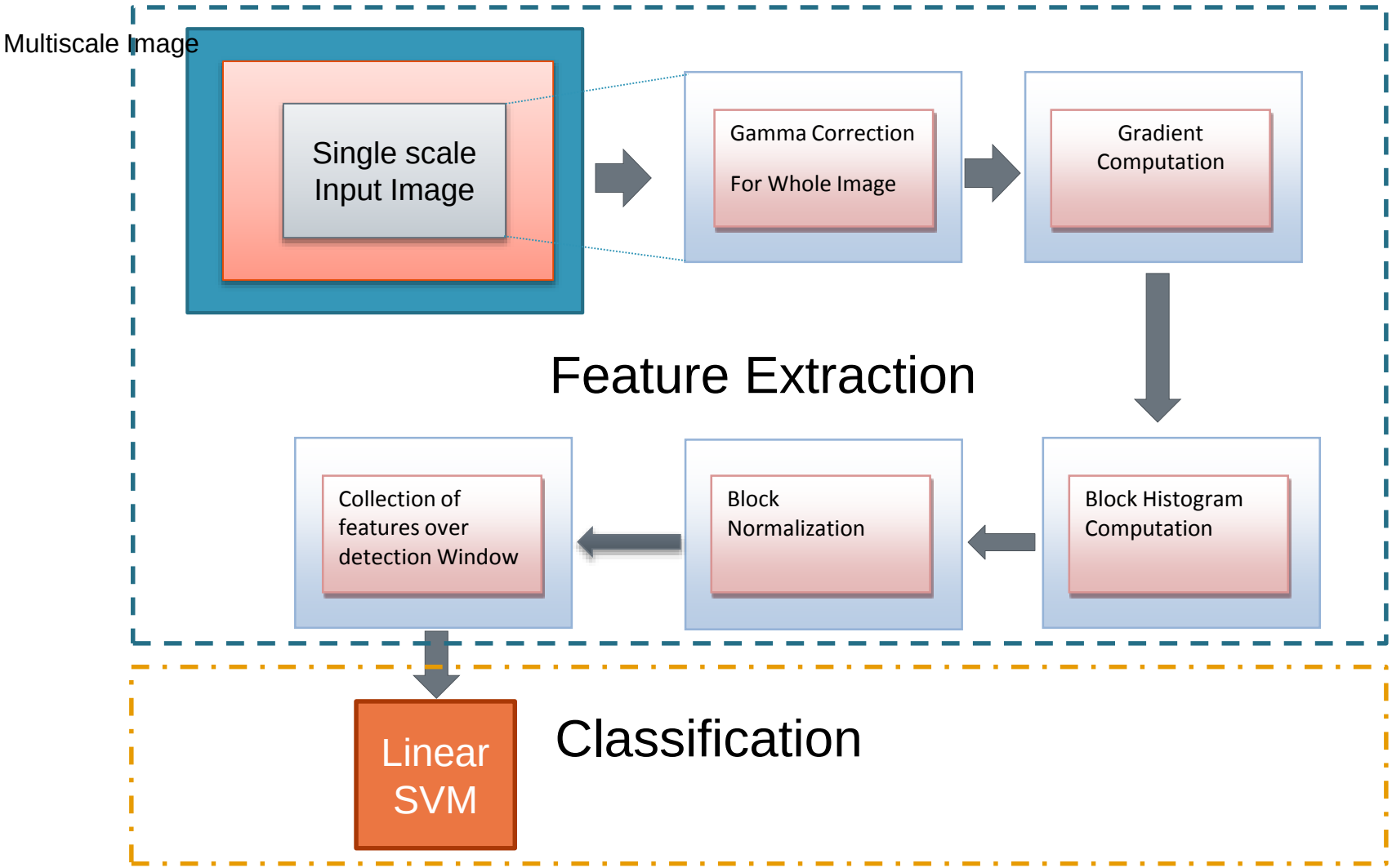


Gradient phase **= atan2(Dy,Dx)**

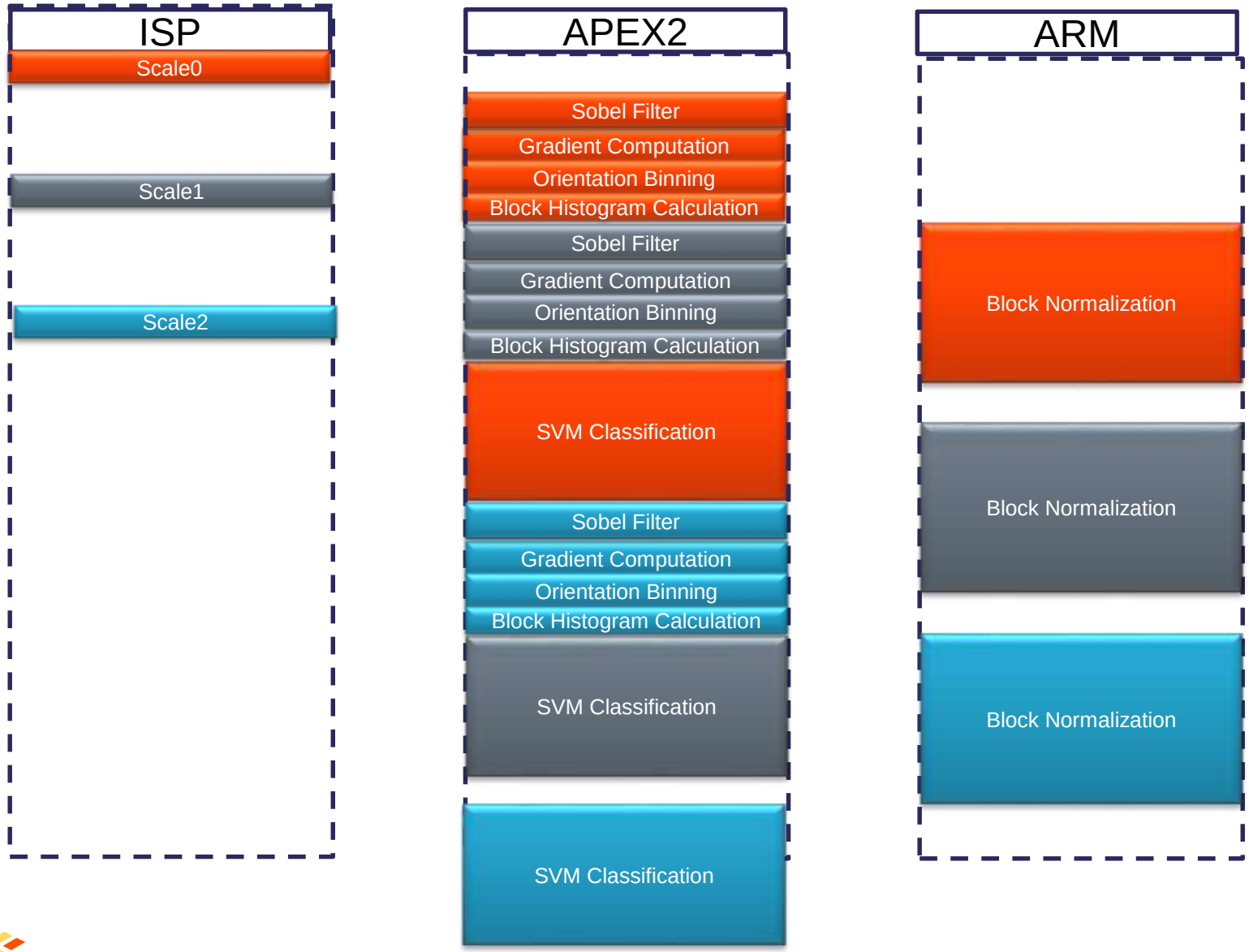
# What are Histogram of Oriented Gradients



# Overview of HOG Algorithm

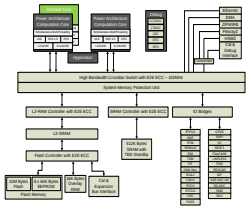


# Pedestrian Detection Pipeline



# Summary

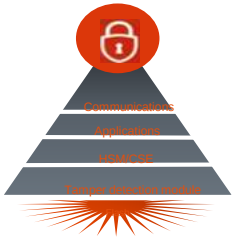
- **Robust:** Fully targeted at ISO26262, Embedded security, Reliable, dependable automotive design and integration
- **Efficient:** Intelligent partitioning to reduce cost, Dedicated Acceleration to improve performance, Best in class power
- **Flexible:** Open programming models, Supported by off-the-shelf RTOS & tools, Enabled by 360º EcoSystem



*Performance*


*Quality*


*Safety*



*Security*

# Thank you!
# Questions?


www.Freescale.com