FTF 2016
TECHNOLOGY FORUM

USING THE S32V234 EVB DEVELOPMENT PLATFORM

FTF-AUT-N1808

STEVE MCASLAN
PRINCIPAL ENGINEER
FTF-AUT-N1808
MAY,18 2016

AGENDA

- Introduction to the S32V234 vision processor development platform
- Peripheral and real-time processing
- Application processor cluster
- Vision processing elements
- The Image Signal Processor
- Vision SDK

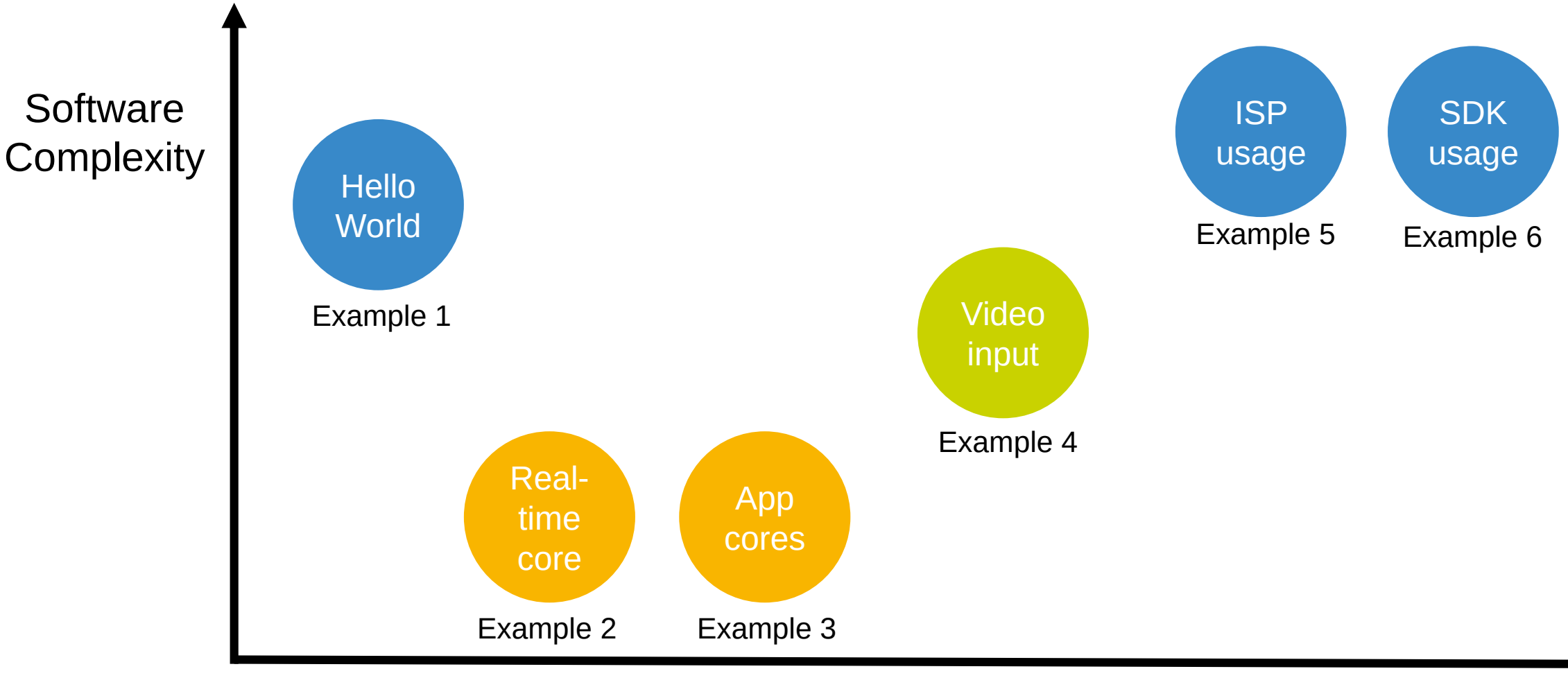
S32V234

- Vision processing MPU
  - multiple ARM® Cortex®-A53 CPUs and one Cortex-M4 CPU
  - many dedicated modules and processors for acceleration of vision processing tasks
  - Developed according to ISO 26262 with an integrated safety concept
- Automotive applications include front view, surround view and Data Fusion systems
  - Scalable
- Includes leading edge camera vision modules like APEX2, ISP, and GPU

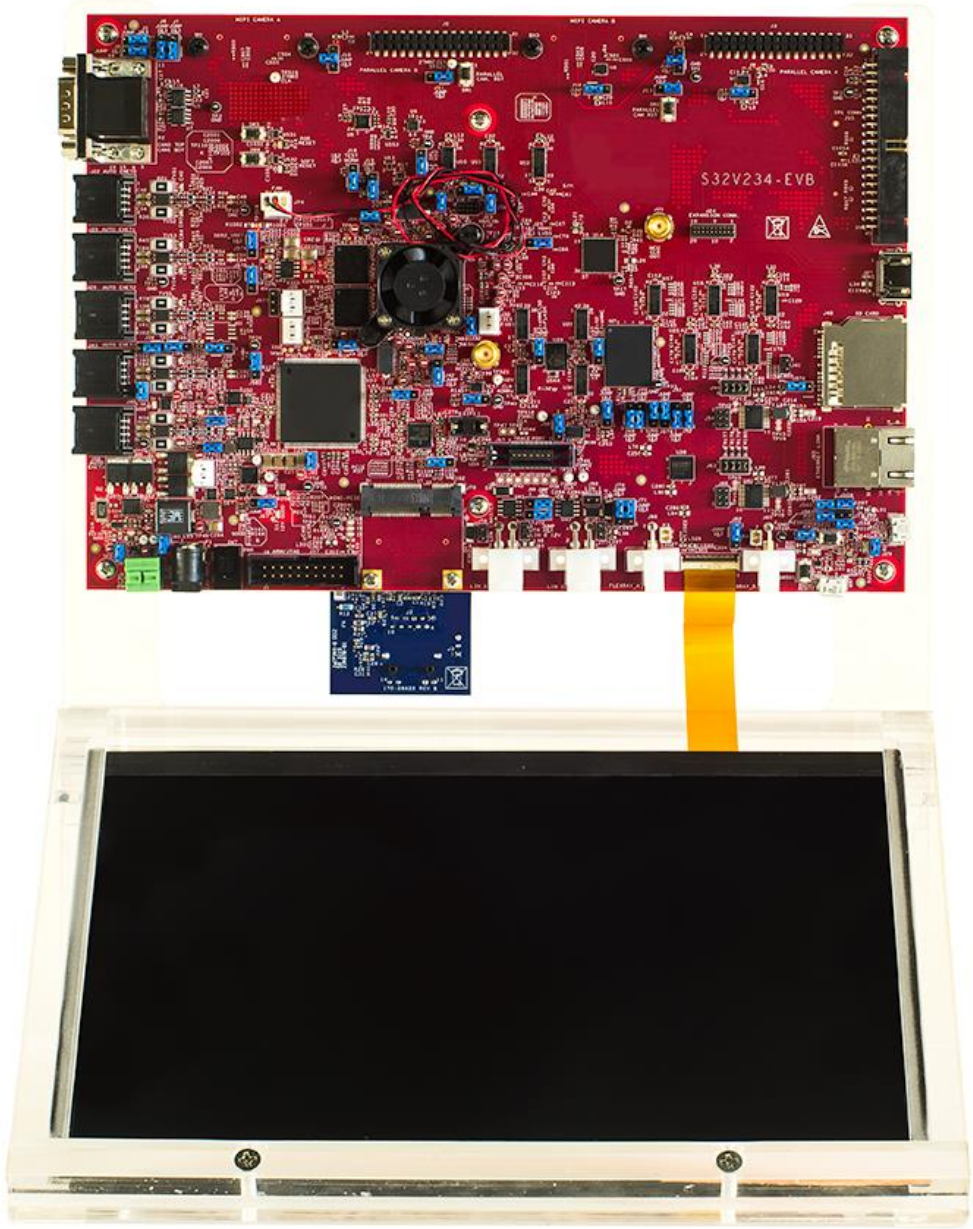
S32V234 Development Environment and This Session

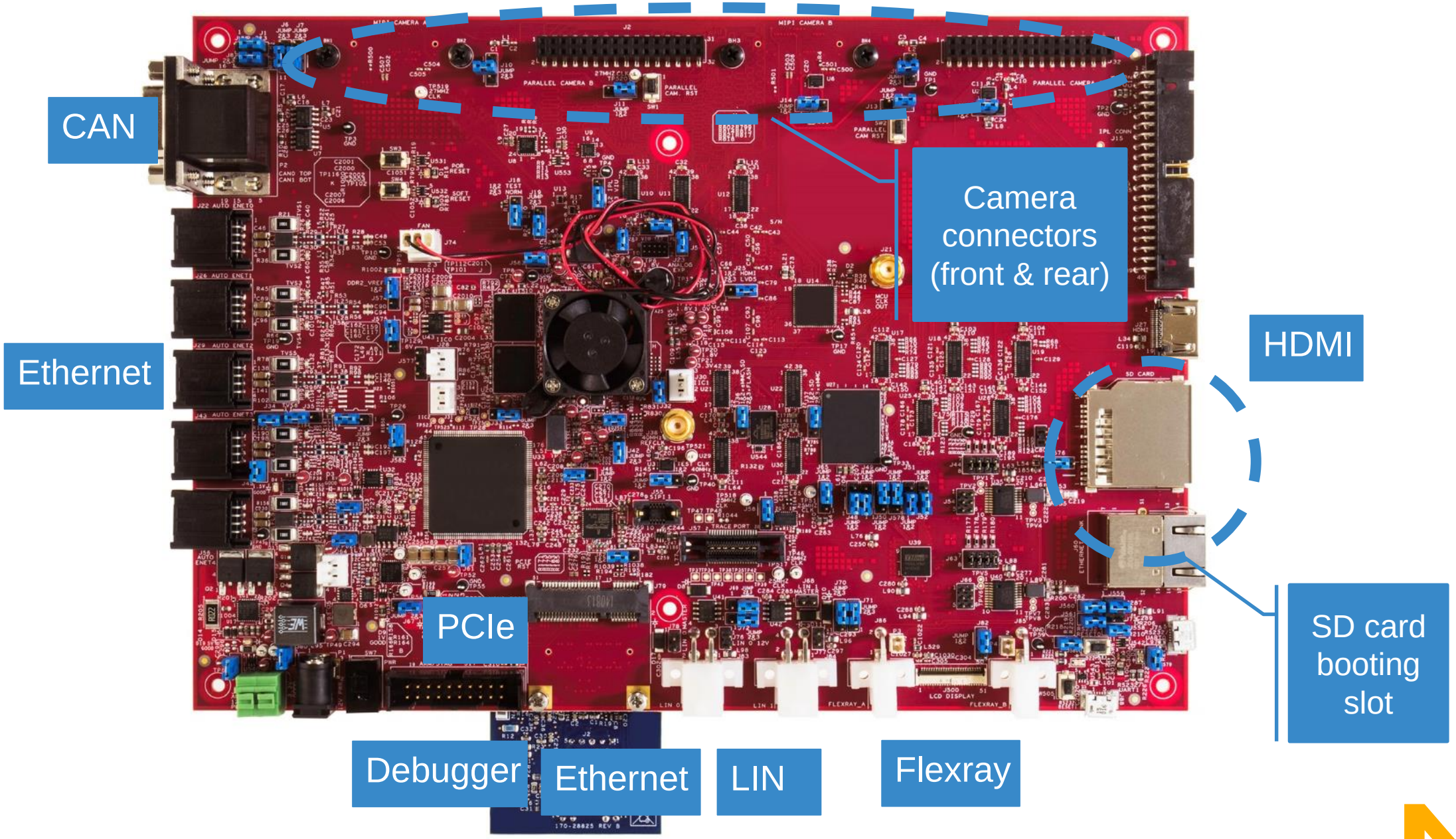
- The NXP development environment consists of hardware platforms and software tools, libraries and operating systems
- This session uses the full-featured EVB for convenience however the examples are also suitable for low cost boards under development
- There are 6 examples to follow which illustrate the features of the board, tools and software
  - Illustrate the low-level boot behaviour of the device
  - Higher level OS and SDK capability
  - Opportunity to modify software functions and run these on the hardware

Workshop Flow



INTRODUCTION TO THE S32V234 DEVELOPMENT PLATFORM


S32V234-EVB

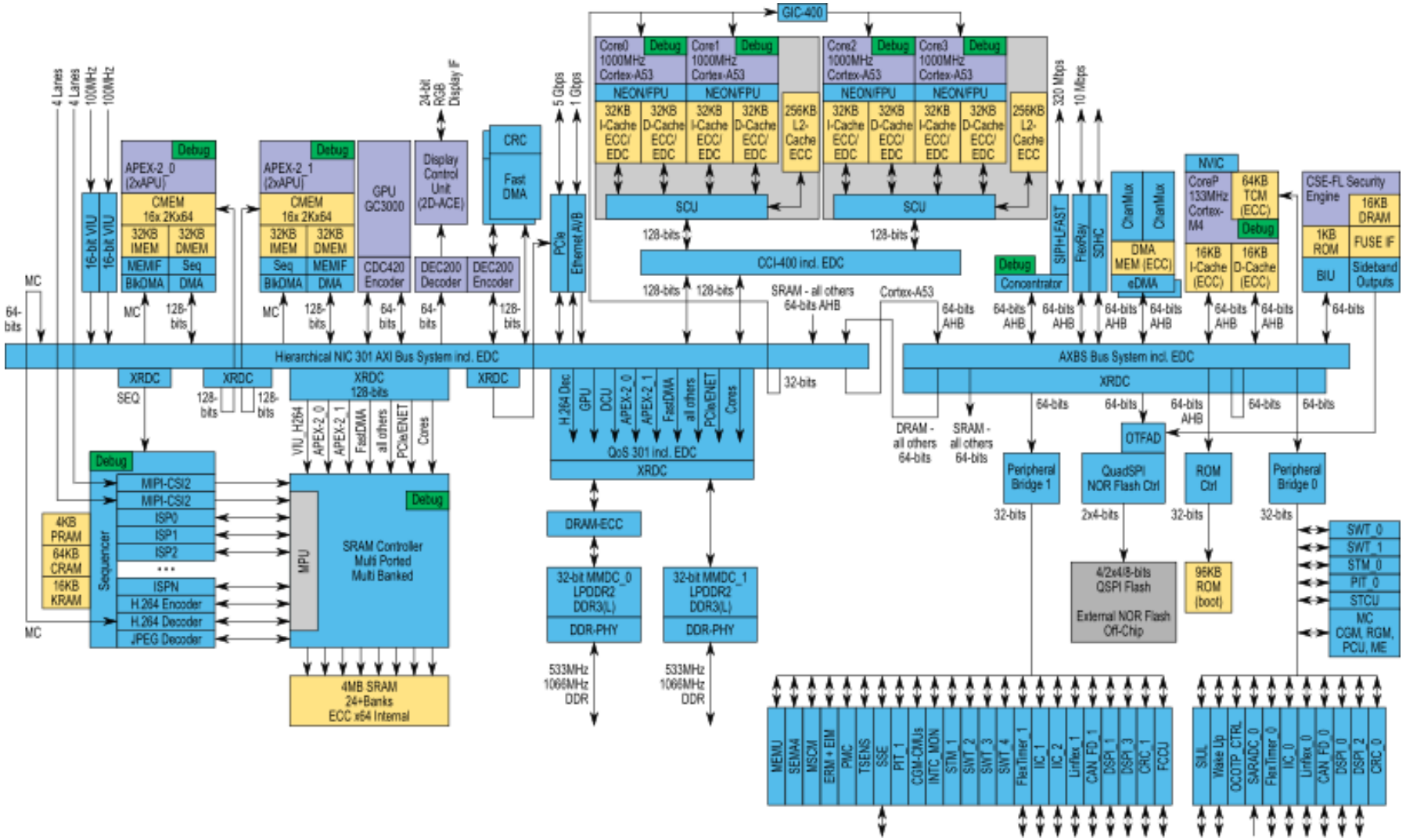

CAN
Ethernet
Camera connectors (front & rear)
HDMI
SD card booting slot
PCIe
Debugger
Ethernet
LIN
Flexray

Example 1: Hello World

This example is here to make sure the system is functional and you are familiar with the EVB features

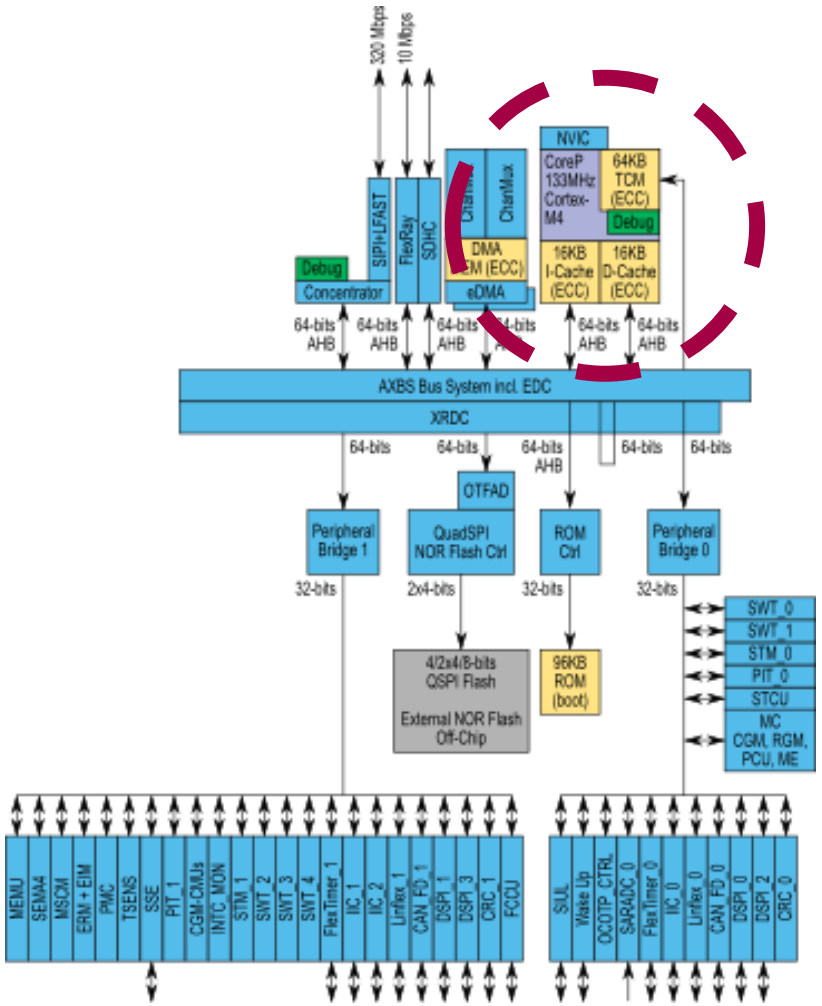
PERIPHERAL AND REAL-TIME PROCESSING

S32V234 Detailed Block Diagram



Peripheral and Real-time Processing

- This portion of the processor is effectively an embedded real-time processor with optimised connection to the device peripherals
  - cf Kinetis family
- It has its own dedicated memory (TCM)
- Boot sources are close by
  - SD/eMMC and QuadSPI
- It is always the boot processor



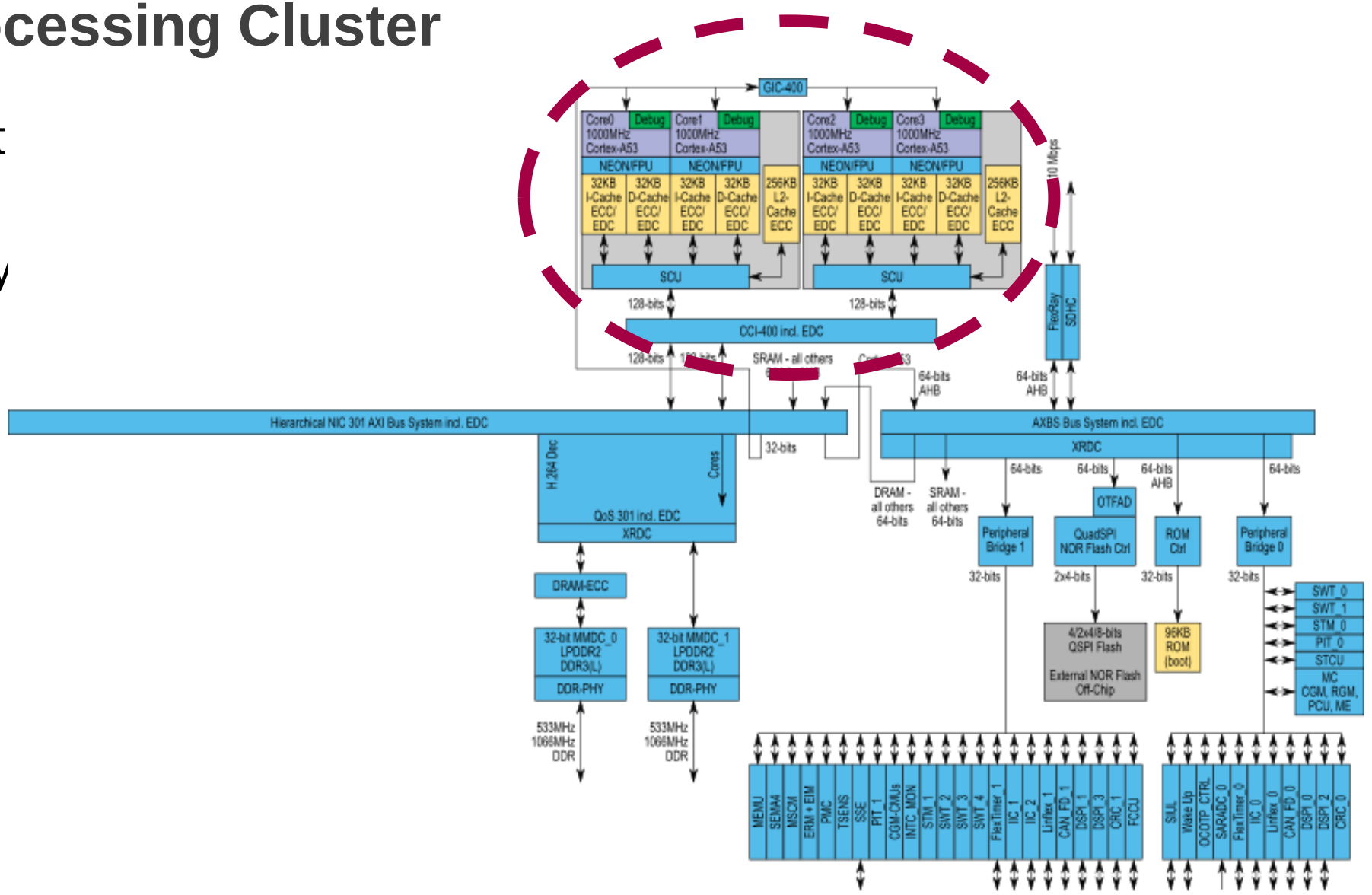
Example 2: Peripheral Use

This example allows you to take control of the UART and use it to interact with the real-time part of the system

APPLICATION PROCESSOR CLUSTER

Applications Processing Cluster

- This processor clust
  OS tasks
- We can boot directly



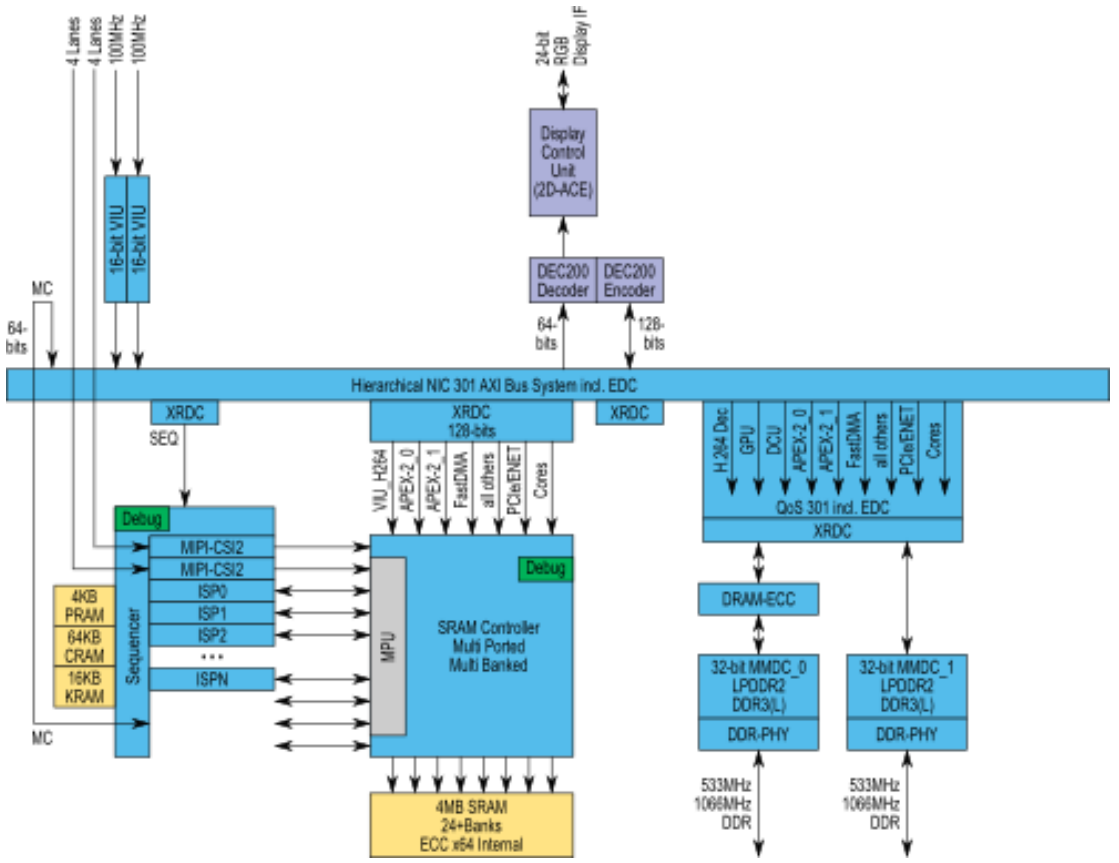
Example 3: Application Cluster Use

- This example allows you to use the system DRAM and launch the Cortex M4 and further Cortex A53 cores under the control of the boot core
- This shows the effect that multiple cores can have on throughput

VISION PROCESSING ELEMENTS

Vision Processing Elements

- This part of the processor captures incoming RAW video from the cameras and performs basic processing on it before storing it into DRAM
- We will use the 2D-ACE to see the effect that the processing steps have on the video



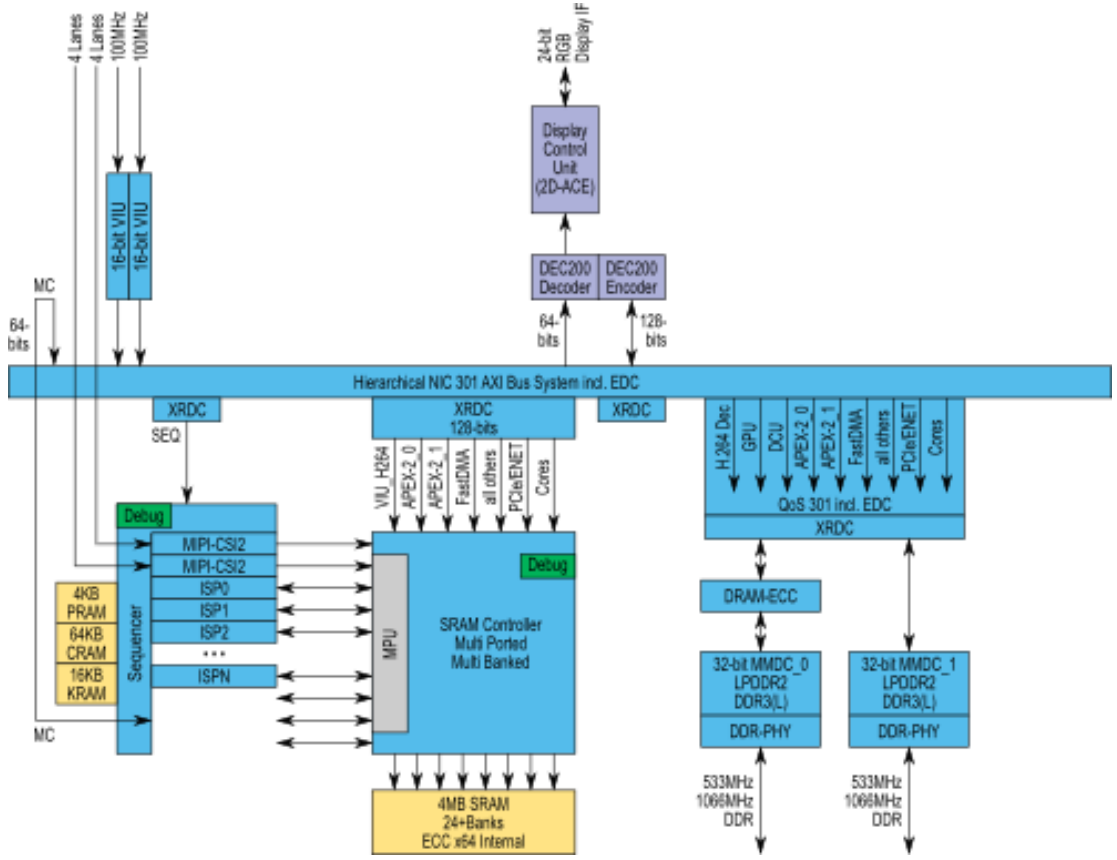
Example 4: Camera Input

- This example captures the video from the camera and uses the 2D-ACE to display it in real time on the display panel
- Using some simple software settings we will adjust the appearance of the video

THE IMAGE SIGNAL PROCESSOR

Vision Processing Elements

- This part of the processor captures incoming RAW video from the cameras and performs basic processing on it before storing it into DRAM
- We will use the 2D-ACE to see the effect that the processing steps have on the video



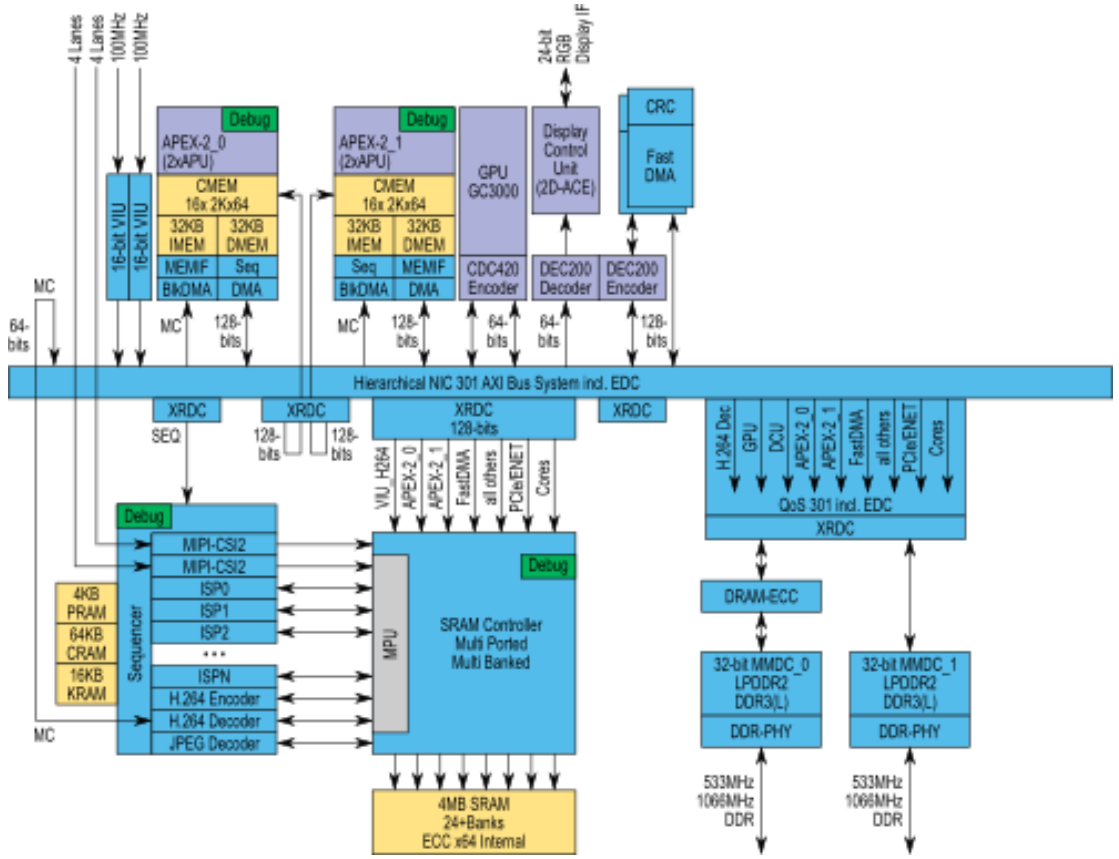
Example 5: ISP Use

- This example uses the Cortex A53 to manipulate the parameters of the ISP processing steps
- This is our first use of the S32V234 Vision SDK
- We will run examples from the file system on the laptops

VISION SDK

Vision SDK

- The full vision SDK includes support for the APEX engines, the ISP, the cameras and the 2D-ACE
- This allows us to extract information from the incoming video and identify items of interest



Example 6: Vision SDK Use

We will examine various Vision SDK example projects to review the range of features supported

Summary

- In this session we have seen how the S32V234 vision processor boots from external memory and how the code in that memory is structured
- We can choose the initial target processor and have it launch the other processing elements in the system
- Real-time and peripheral management has a dedicated zone on chip and example support on board
- Capturing and displaying video is optimised by the architecture of the S32V234 and the EVB
- We can create an optimised development environment for our vision processing using the software tools running under a Linux environment with a remote file system

SECURE CONNECTIONS
FOR A SMARTER WORLD

ATTRIBUTION STATEMENT