Leverage the i.MX 6 Series Applications Processor for Industrial Applications

FTF-IND-F0244

Hans-Peter Rosinger | FAE

A P R . 2 0 1 4
Session Introduction

• This session talks about how to maximize computing performance and take advantage of flexible interfaces using the i.MX 6 series applications processor
  – Learn how to enable and use common acceleration units on i.MX 6 series
  – Recognize the benefit and flexibility i.MX 6 series offers for attaching peripherals
• About myself:
  – My name is Hans-Peter Rosinger, I am an Freescale field application engineer for the i.MX processor and Vybrid controller solution product lines.
  – I am based in Munich, Germany, and support industrial and general embedded customers.
• This session will last about 50min with Q&A at the end.
Session Objectives

After completing this session you will be able to:

- Specify industrial grade i.MX 6 series processors which are pin compatible and part of the Freescale product longevity program
- Describe acceleration units the i.MX 6 series processor offers
- Enable and use them on i.MX 6 series processors running Linux
- Select the most appropriate interface on the i.MX 6 series for connecting peripherals depending on your application requirements
Agenda

• Overview of i.MX 6 series processors
  - Target applications
  - Features and industrial highlights
• Maximize i.MX 6 series computing performance
  - Take advantage of the NEON on chip acceleration unit
  - Integrate it into your application software
  - Let the GPU do the job featuring OpenCL
  - Power vs. performance trade off
• Connect peripherals to i.MX 6 series
  - Leverage many feature rich and flexible interfaces
  - Low latency / high throughput options on i.MX 6 series connecting an ASIC or FPGA
• Summary
i.MX 6 Series Target Applications

Automotive

- Infotainment
- Telematics
- Instrument Clusters
- Vision/Camera Systems

eReaders

- Monochrome eReader
- Color eReaders

Smart Devices

- IPTV/Streaming Media
- Transportation – in-flight infotainment, marine navigation, in-car signage
- Health Care – patient monitoring, telehealth, fitness equipment
- Factory, process and building automation
- Handheld scanners and printers
- Digital signage

- Thin Client and cloud server
- Point of sale terminal and printer
- Tablets – medical, educational, industrial
- IPTV/Streaming media
- IP phone
- Test and Measurement
- Appliances
- Home audio
i.MX 6 Series Industrial Grade Features

- **ARM® Cortex™-A9 based solutions** with up to 800MHz
- **Video playback in high definition** for stunning visual experience
- **Triple Play graphics architecture** enables graphics with separate 3D and 2D graphics engines
- **Display interfaces** LVDS, Parallel and HDMI display technology
- **Optimized peripheral sets:** PCIe/SATA/GbE, USB/SD/MIPI and CAN/MLB150 tailored for the auto infotainment, consumer and industrial markets
- **64-bit memory bus for optimal performance**, and memory interfaces support DDR3, DDR3L and LP-DDR2
- **Integrated power management** simplifies design and reduces external components
- **Specialized support for harsh conditions** unique to automotive and industrial environments (i.e. extreme temperatures)
i.MX 6 Series: One Platform, Differentiated Products

- Provide customers and partners with the **broadest range** of choices
- Reduce **development costs** and **improve** time to market
- **Performance scalability** is key to implement this strategy
**6 Series At a Glance**

Scalable series of **five** ARM Cortex A9-based SoC families

<table>
<thead>
<tr>
<th>i.MX 6SoloLite</th>
<th>i.MX 6Solo</th>
<th>i.MX 6DualLite</th>
<th>i.MX 6Dual</th>
<th>i.MX 6Quad</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single ARM® Cortex™-A9 at 1.0GHz</td>
<td>Single ARM Cortex-A9 at 1.0GHz</td>
<td>Dual ARM Cortex-A9 at 1/1.2GHz</td>
<td>Dual ARM Cortex-A9 at 1/1.2GHz</td>
<td><strong>Quad</strong> ARM Cortex-A9 at 1.2GHz</td>
</tr>
<tr>
<td>256KB L2 cache, Neon, VFPvd16, Trustzone</td>
<td>512KB L2 cache, Neon, VFPvd16, Trustzone</td>
<td>1 MB L2 cache, Neon, VFPvd16, Trustzone</td>
<td>1 MB L2 cache, Neon, VFPvd16, Trustzone</td>
<td>1 MB L2 cache, Neon, VFPvd16, Trustzone</td>
</tr>
<tr>
<td>2D graphics</td>
<td>3D graphics with 1 shader</td>
<td>3D graphics with 1 shader</td>
<td>3D graphics with 4 shaders</td>
<td>3D graphics with 4 shaders</td>
</tr>
<tr>
<td>32-bit DDR3 and LPDDR2 at 400MHz</td>
<td>2D graphics</td>
<td>2D graphics</td>
<td>Two 2D graphics engines</td>
<td>Two 2D graphics engines</td>
</tr>
<tr>
<td>Integrated EPD controller</td>
<td>3D graphics with 1 shader</td>
<td>64-bit DDR3 and 2-channel 32-bit LPDDR2 at 400MHz</td>
<td>64-bit DDR3 and 2-channel 32-bit LPDDR2 at 533MHz</td>
<td>64-bit DDR3 and 2-channel 32-bit LPDDR2 at 533MHz</td>
</tr>
<tr>
<td><strong>i.MX 6 Series Highlights</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARM Cortex-A9 based solutions ranging up to 1.2GHz</td>
<td>HD 1080p encode and decode (except 6SL)</td>
<td>3D video playback in High definition (except 6SL)</td>
<td>SW support: Google Android™, Windows® Embedded CE, Linux®</td>
<td></td>
</tr>
</tbody>
</table>

Features vary by product family
i.MX 6 Series for Industrial

• Broad Software Support
  - Mature Freescale Linux offering with LTS Kernel support
  - Broad Android offering with 3.x and 4.x
  - Windows Embedded support through proven partners
  - RTOS support including QNX, VxWorks, Integrity
  - Strong Ecosystem with Module Manufacturers and Software Integrators

• Industrial Environment and Peripheral Support
  - 3.3V IO
  - 0.8mm pitch package to reduce PCB & manufacturing cost
  - Flexible interface supporting LP-DDR2, DDR3 and LV-DDR3
  - Broad peripheral support – Gbit ethernet, CAN, SATA, PCI-E, LVDS, MIPI
  - Extended Temp range available with full Industrial qualified parts

• Up to 15 year lifetime support for manufacturing
i.MX 6 Series Product Longevity

- Certain devices in the i.MX 6 series have been added to Freescale’s product longevity program.
- Visit [freescale.com/productlongevity](freescale.com/productlongevity) for more information.

<table>
<thead>
<tr>
<th>i.MX 6 Family</th>
<th>Part #</th>
<th>Description</th>
<th>Package</th>
<th>Temp Range</th>
<th>Longevity</th>
<th>Product Launch</th>
</tr>
</thead>
<tbody>
<tr>
<td>i.MX 6Quad</td>
<td>MCIMX6Q6AVT10AD</td>
<td>1GHz Automotive</td>
<td>21x21, 0.8mm FCBGA lidded</td>
<td>-40C to +125C</td>
<td>15yrs</td>
<td>2012-11</td>
</tr>
<tr>
<td>i.MX 6Quad</td>
<td>MCIMX6Q7CVT08AD</td>
<td>800MHz Industrial</td>
<td>21x21, 0.8mm FCBGA lidded</td>
<td>-40C to +105C</td>
<td>10yrs</td>
<td>2012-11</td>
</tr>
<tr>
<td>i.MX 6Dual</td>
<td>MCIMX6D6AVT10AD</td>
<td>1GHz Automotive</td>
<td>21x21, 0.8mm FCBGA lidded</td>
<td>-40C to +125C</td>
<td>15yrs</td>
<td>2012-11</td>
</tr>
<tr>
<td>i.MX 6Dual</td>
<td>MCIMX6D7CVT08AD</td>
<td>800MHz Industrial</td>
<td>21x21, 0.8mm FCBGA lidded</td>
<td>-40C to +105C</td>
<td>10yrs</td>
<td>2012-11</td>
</tr>
<tr>
<td>i.MX 6DualLite</td>
<td>MCIMX6U6AVM08AC</td>
<td>800MHz Automotive</td>
<td>21x21, 0.8mm MAPBGA</td>
<td>-40C to +125C</td>
<td>15yrs</td>
<td>2012-11</td>
</tr>
<tr>
<td>i.MX 6DualLite</td>
<td>MCIMX6U7CVM08AC</td>
<td>800MHz Industrial</td>
<td>21x21, 0.8mm MAPBGA</td>
<td>-40C to +105C</td>
<td>10yrs</td>
<td>2012-11</td>
</tr>
<tr>
<td>i.MX 6Solo</td>
<td>MCIMX6S6AVM08AC</td>
<td>800MHz Automotive</td>
<td>21x21, 0.8mm MAPBGA</td>
<td>-40C to +125C</td>
<td>15yrs</td>
<td>2012-11</td>
</tr>
<tr>
<td>i.MX 6Solo</td>
<td>MCIMX6S7CVM08AC</td>
<td>800MHz Industrial</td>
<td>21x21, 0.8mm MAPBGA</td>
<td>-40C to +105C</td>
<td>10yrs</td>
<td>2012-11</td>
</tr>
</tbody>
</table>
Maximize i.MX 6 Series Computing Performance
Why Computing Performance in Industrial?

• Application for object detection / gesture recognition
  – Require mathematical algorithms

• Growing display and camera resolution required for HMI
  – Image processing capabilities

• Demand for similar “consumer like” user experience
  – Graphics acceleration

• Battery powered devices in industrial automation
  – Compute smart and power efficient
i.MX 6Quad/6Dual Applications Processor

- ARM® Cortex–A9 widely adopted multicore processor
  - Dual-issue, Out of order pipeline
  - Scalable SMP Processing

- Flexible system architecture
  - 32KB L1 / D L1 cache per core
  - Optimized L2 Cache controller available
  - Cortex-A9 MP Interrupt controller (GIC) and timers / watchdog functions built-in
ARM® NEON Acceleration Unit

• NEON® is a wide SIMD (Single Instruction, Multiple Data) data processing architecture
  - Extension for ARM Cortex-A series processors and ARM instruction set
  - It provides support for the ARM v7 Advanced SIMD and Vector Floating-Point v3 (VFPv3) instruction sets.
• 32 registers, 64 bite wide (dual view as 16 registers, 128-bits wide)
• NEON instructions perform “Packed SIMD” processing
  - Registers are considered as vectors of elements of the same data type
  - Data types can be:
    ▪ Singed/unsigned 8-bit, 16-bit, 32-bit 64-bit, single precision or floating point
    ▪ Instructions perform same operation in all lane
• With the exception of simultaneous loads / stores, the processor can execute SIMD(VFP) instructions in parallel with ARM or Thumb instructions
Software Enablement

• NEON can be enabled on Bare Metal Code
  – No OS required
  – Inline Assembler
  – NEON Intrinsics available (for GCC when mfpu=neon)

• Freescale offers Platform SDK
  – Bare metal SDK for i.Mx6 Series
  – starts Cortex-A9 (optional enable, NEON, L1 Cache, MMU, etc)
  – Reusable drivers and tests for many peripherals
  – [https://community.freescale.com/docs/DOC-94139](https://community.freescale.com/docs/DOC-94139)

• But how to enable NEON on an OS?
ARM® Ne10 Project

- **Some facts:**
  - An Open Optimized Software Library Project for the ARM architecture
  - The Ne10 library provides a set of the most commonly used functions that have been heavily optimized for ARM based CPUs with NEON
  - The library focuses on matrix/vector algebra and signal processing
  - Ne10 will evolve over time

- **Software Enablement**
  - Easy enablement on i.MX 6 series Linux or Android platforms
  - No assembly coding required to use libraries
  - Chose function from Ne10 pre-built library
## Ne10 Supported Functions

### Math Functions
- Vector Add / Sub
- Matrix Add / Sub
- Vector Multiply / Multiply-Accumulator
- Matrix Multiply / Vector Multiply
- Vector Div / Setc / Len /Normalize / Abs / Dot / Cross / Rsbc
- Matrix Div / Determinant / Invertible / Transpose / Identify

### Signal Processing Functions
- Complex FFT
- Float/Fixed point Complex FFT
- Finite Impulse Response (FIR) Filters
- Finite Impulse Response (FIR) Decimator
- Finite Impulse Response (FIR) Interpolator
- Finite Impulse Response (FIR) Lattice Filters
- Finite Impulse Response (FIR) Sparse Filters

### Image Processing Functions
- Image Resize
- Image Rotate

Four Steps to Use Ne10 on i.MX 6 Series Running Linux

- Prepare target or host
  - ARM GCC, cmake, make, git
- Download Ne10 project
  - $ git clone git://github.com/projectNe10/Ne10
- Compile and use library
  - $cd $NE10PATH, mkdir build, cd build, cmake .., make
  - libNE10.a is placed in $NE10PATH /build/modules/
  - A test program "NE10_test_static" is placed in $NE10PATH /build/samples/.
  - More tests can be compiled $NE10PATH /build/test/
- You can run it
Demo on i.MX 6Quad Sabre SDB

```
root@imx6qsabresd:/Ne10/build# make
Scanning dependencies of target NE10
[ 0%] Building C object modules/CMakeFiles/NE10.dir/math/NE10_abs.c.o
[ 1%] Building C object modules/CMakeFiles/NE10.dir/math/NE10_addc.c.o
root@imx6qsabresd:/Ne10/build# make
Scanning dependencies of target NE10
[ 2%] Building C object modules/CMakeFiles/NE10.dir/math/NE10_add.c.o
[ 3%] Building C object modules/CMakeFiles/NE10.dir/math/NE10_divc.c.o
root@imx6qsabresd:/Ne10/build/test# ls
CMakeFiles Makefile NE10dsp_unit_test_static NE10_imgproc_unit_test_static NE10_math_unit_test_static
------ /home/root/Ne10/modules/imgproc/test/test_suite_resize.c ------
  1 run  0 failed

------ /home/root/Ne10/modules/imgproc/test/test_suite_rotate.c ------
  1 run  0 failed

--------------------- SEATEST v0.5 ---------------------

  ALL TESTS PASSED
  2 tests run
  in 1009 micro-s

root@imx6qsabresd:/Ne10/build/test#```
i.MX 6Quad/6Dual Applications Processor

• Introducing the i.MX Triple-Play Graphics Solution
  - High-Performance OpenGL/OpenCL GPU Core
    - Vivante GC2000 @ 528MHz (600 MHz shader)
    - 88M triangles / sec (176MT/s @ 50% cull)
    - 4 Shader Cores: 21.6 GFLOPS
    - 1056 M Pixels/s, 1056 M Texels/s

• High-Performance Composition Engine 2D Core
  - Up to 633M pixels / sec raw performance
  - Vivante GC320

• High-Performance Vector Graphics Core
  - 264M Pixels / sec raw performance
  - Native OpenVG 1.1 vector acceleration with hardware tessellation
  - Vivante GC355
Vivante GC2000 ScalarMorphic™ Architecture

- Architectural Features
- CoolThreads™ multilevel clock gating and massive data parallelism
- 4X SIMD Vec-4 unified shaders (IEEE single precision FP)
- Point sample, bi-linear/tri-linear and cubic/volumetric textures
- Balanced performance/ bandwidth
- Memory locality / Texture compression
- Frame buffer/Z compression
- Native OpenGL ES 2.0 rendering
- Quality 4x MSAA anti-aliasing
- Depth buffering

- System Functionality
- Up to 1024 independent threads
- Low bandwidth characteristics for both simple and complex scenes
- Comprehensive API support
- OpenGL ES 1.1 / 2.0
- OpenVG 1.1
- OpenGL 2.1 & 3.0 (w/ GLX)
- OpenCL 1.1 EP
i.MX 6 Series GPU Driver Overview

- The GPU driver is structured in a “User space” part and a “Kernel space” part that communicate through ioctl
- The user side contains command buffers where the application puts commands for the GPU
- On the kernel side, command queues are used to link and manage the command buffers supplied
- The hardware is synchronized with the software through hardware events
OpenCL Overview

• OpenCL is an open specification API from the Khronos Group
  - Same standards body that made OpenGL, OpenVG, OpenMAX, etc.
• OpenCL enables asynchronous multi-core programming for heterogeneous computing environments
  - Allows the user to run work ‘kernels’ that run over large data sets and across many processors
  - Effectively makes GPU look like a cluster of DSPs executing on a task
  - i.MX 6Q/D implements the OpenCL 1.1 Embedded Profile
• Language Specification
  - C-based cross-platform programming interface
  - Subset of ISO C99 with language extensions - familiar to developers
  - Well-defined numerical accuracy (IEEE 754 rounding with specified max error)
  - Online or offline compilation and build of compute kernel executables
  - Includes a rich set of built-in functions
• Platform Layer API
  - A hardware abstraction layer over diverse computational resources
  - Query, select and initialize compute devices
  - Create compute contexts and work-queues
• Runtime API
  - Execute compute kernels
  - Manage scheduling, compute, and memory resources
OpenCL Example Use Cases

OpenCL efficient functions

- Haar-like Features / Wavelets
- Sobel Line Detection
- Computing Eigenvalues
- FFT
- Noise Generation
- DCT
- Histogram Generation
- Histogram of Oriented Gradients (HOGs)
- Demosiac
## i.MX 6 Series Open CL Execution Model

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compute Units per device (Shader cores)</td>
<td>4</td>
</tr>
<tr>
<td>Processing Elements per compute unit</td>
<td>4</td>
</tr>
<tr>
<td>Profile</td>
<td>Embedded</td>
</tr>
<tr>
<td>Preferred work-group/ thread group size</td>
<td>16</td>
</tr>
<tr>
<td>Max count global work-items each dim</td>
<td>64K</td>
</tr>
<tr>
<td>Max count of work-items each dim per work-group</td>
<td>1K</td>
</tr>
<tr>
<td>Instruction Memory</td>
<td>512</td>
</tr>
<tr>
<td>Texture Samplers available to OCL</td>
<td>4</td>
</tr>
<tr>
<td>L1 Cache Size</td>
<td>4KB</td>
</tr>
</tbody>
</table>
Power Versus Performance Tradeoff

- Simple test on i.MX 6Quad SabreSDB – Yocto 3.10.17 Beta Release
- Enabled two Cortex-A9 cores
- Vector ADD (Data type float32)
  - Cortex-A9
  - Cortex-A9 + NEON
  - Cortex-A9 + OpenCL (GC2000)
- Depending on size NEON / OpenCL can show significant performance boost
- On small data size OpenCL sub optimal or even worse because of OpenCL overhead
- Significant power consumption increase (~+60%) when enabling OpenCL
Best Reasons When to Choose NEON / GPU OpenCL

• When CPU utilization is high due to DSP and MATH computing
  – Filter (IIR, FIR) MAC, Floats
• When dealing with vectors
• NEON runs at CPU speed and takes advantage of L1/L2 cache
• CPU and NEON can make use of local SRAM
• Make use of Ne10 Library to use NEON for Linux or Android applications
• OpenCL can bring more performance and is more flexible
Connect Peripherals to i.MX 6
Why Attach or Integrate Peripherals in Industrial?

• Industrial IO Systems
  – Support (different) industrial Ethernet protocols
  – Require often Integration of proprietary ASIC or FPGA
  – Flexible and efficient interface

• Compact form factors
  – CPU needs to service additional channels (UART, USB, etc)

• Cooling due to compact size or closed body an issue
  – Power efficient system architectures required

• At the same time data throughput demand growing
  – An high throughput interface is required
i.MX 6 Series Interface Offering

- i.MX 6 series Industrial Ethernet support (partners)
  - EtherCAT master
  - PROFINET RT (Not IRT)
  - Modbus TCP
  - IEEE 1588
  - DeviceNet (CAN)
- EtherCAT slave or other proprietary Interfaces require ASIC / FPGA
External Interface Module (EIM)

- The EIM handles the interface to devices peripherals and memory to the chip
- Feature Highlights:
  - Up to four chip selects for external devices
  - Support for multiplexed address / data bus operation
  - Programmable wait-state generator for each CS
  - Asynchronous or Synchronous access to devices
  - Flexible 8/16 or 32-bit configurable data width
  - Asynch. page Mode access and Burst support for synch mode
  - External Interrupt Support
EIM Benefit and Software Enablement

• Connect FPGA / ASIC directly through i.MX 6 series EIM
• No protocol overhead thus efficient when accessing small data junks (single bytes, half or words)
• If real-time data are computed in external peripheral they can be transferred with low latency to i.MX 6 series (could run Linux RT Xenmai)
• To burst data is are possible
• Driver can directly memory map EIM address space
• Freescale provides reference driver in Platform SDK
  – Bare metal driver can be ported to Linux, WinCE or RTOS
i.MX 6 Series Interface Offering

• But what if higher throughput is required?
PCI Express Overview

• PCIe Specification

• PCI Express architecture is a high performance, IO interconnect for peripherals in computing/communication platforms

• Evolved from PCI and PCI-X architectures
  – 32-bit or 64-bit data bus, addressing scheme
  – Bus arbitration

• PCIe enhancements
  – Point-to-point connection
  – Serial high Speed Interface (1,2,4,8,16, etc Lanes)
  – PCIe uses differenc encoding shemes like 8b/10b
  – Higher throughput

• PCIe Topology
  – PCIe Root Complex (typical Master or main CPU)
  – PCIe Endpoint (typical high speed Peripheral)
  – PCIe Switch (Peripheral Expander)
i.MX 6 Series PCIe Integration

- PCIe include the following cores
  - PCIe Dual Mode (DM) core
  - PCIe Root Complex (RC) core
  - PCIe Endpoint (EP) core
  - AXI Bridge to Interconnect

- PCI Express PHY supports
  - PCIe 2.0 / 5Gbps
  - PCIe 1.1 / 2.5 Gbps
  - IEEE 1149.6 (JTAG) Bscan
  - Built in self Tester (BIST)
PCIe Protocol Stack

- PCI Express Dual Mode controller and PCIe PHY hide most of the complexity of the protocol from application.
- PCIe complex implements standard PIPE interface between digital logic (PCIe controller) and interface PHY.
- i.MX 6 series BSP supports PCIe.
How to Enable PCIe Driver in FSL Linux

- i.MX 6 series FSL Yocto 3.10.17 release supports PCIe
  - Root Complex or Endpoint Driver
Demo of Using and Testing PCIe WLAN Card

• Follow Linux RM – PCIe How to test
  - Generic IEEE 802.11 Networking Stack (mac80211) used by WIFI devices
    ▪ CONFIG_IWLAGN: Select to build the driver supporting the: Intel Wireless Wi-Fi Link Next-Gen AGN
  ▪ WIFI firmware configurations:
    • iwlagn: Intel(R) Wireless WiFi Link AGN driver for Linux, in-tree:
    • iwlagn: Copyright(c) 2003-2011 Intel Corporation
    • PCI: enabling device 0000:01:00.0 (0140 -> 0142)
    • iwlagn 0000:01:00.0: Detected Intel(R) Centrino(R) Ultimate-N 6300 AGN, REV=0x74
    • iwlagn 0000:01:00.0: device EEPROM VER=0x43a, CALIB=0x6
    • iwlagn 0000:01:00.0: Device SKU: 0Xb
    • iwlagn 0000:01:00.0: Valid Tx ant: 0X7, Valid Rx ant: 0X7
    • iwlagn 0000:01:00.0: Tunable channels: 13 802.11bg, 24 802.11a channels
## i.MX 6 Series PCI Express Throughput Results

<table>
<thead>
<tr>
<th></th>
<th>ARM Core used as the bus master and cache disabled</th>
<th>ARM Core used as the bus master and cache enabled</th>
<th>IPU used as the bus master (DMA)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data size in one write tlp</strong></td>
<td>8 bytes</td>
<td>32 bytes</td>
<td>64 bytes</td>
</tr>
<tr>
<td><strong>Write speed</strong></td>
<td>~109 MB/s</td>
<td>~298 MB/s</td>
<td>~344 MB/s</td>
</tr>
<tr>
<td><strong>Data size in one read tlp</strong></td>
<td>32 bytes</td>
<td>64 bytes</td>
<td>64 bytes</td>
</tr>
<tr>
<td><strong>Read speed</strong></td>
<td>~29 MB/s</td>
<td>~100 MB/s</td>
<td>~211 MB/s</td>
</tr>
</tbody>
</table>
Summary

• i.MX 6 series is an successful and optimal choice for industrial applications
  – i.MX 6Quad to i.MX 6Solo full pin to pin compatible
    ▪ Industrial qualified devices
    ▪ Part of the Freescale product longevity program
  – i.MX 6 series offers several computing engines
    ▪ ARM Cortex-A9 including NEON / VFPv3
    ▪ Vivante GP GPU supporting OpenCL
  – i.MX 6 series rich and flexible interfaces
    ▪ Choose between flexible and low latency (EIM) vs. high data throughput (PCIe)