PRESENTED BY

AVNET

Adding Smarter Vision to your Product with Zynq®-7000 All Programmable SoCs

Accelerating Your Success™

The Image Sensor … The Smart Sensor

- Traditionally
  - Image sensor

  - Intent
    - Capture images
    - Transport images
  - Challenges
    - Large storage
    - High bandwidth I/O

- Today
  - Image Sensor + Processing

  - Intent
    - Understand scene
    - Video analytics
  - Challenges
    - high pixel rate image and video processing
    - complex analytics operations

Course Objectives

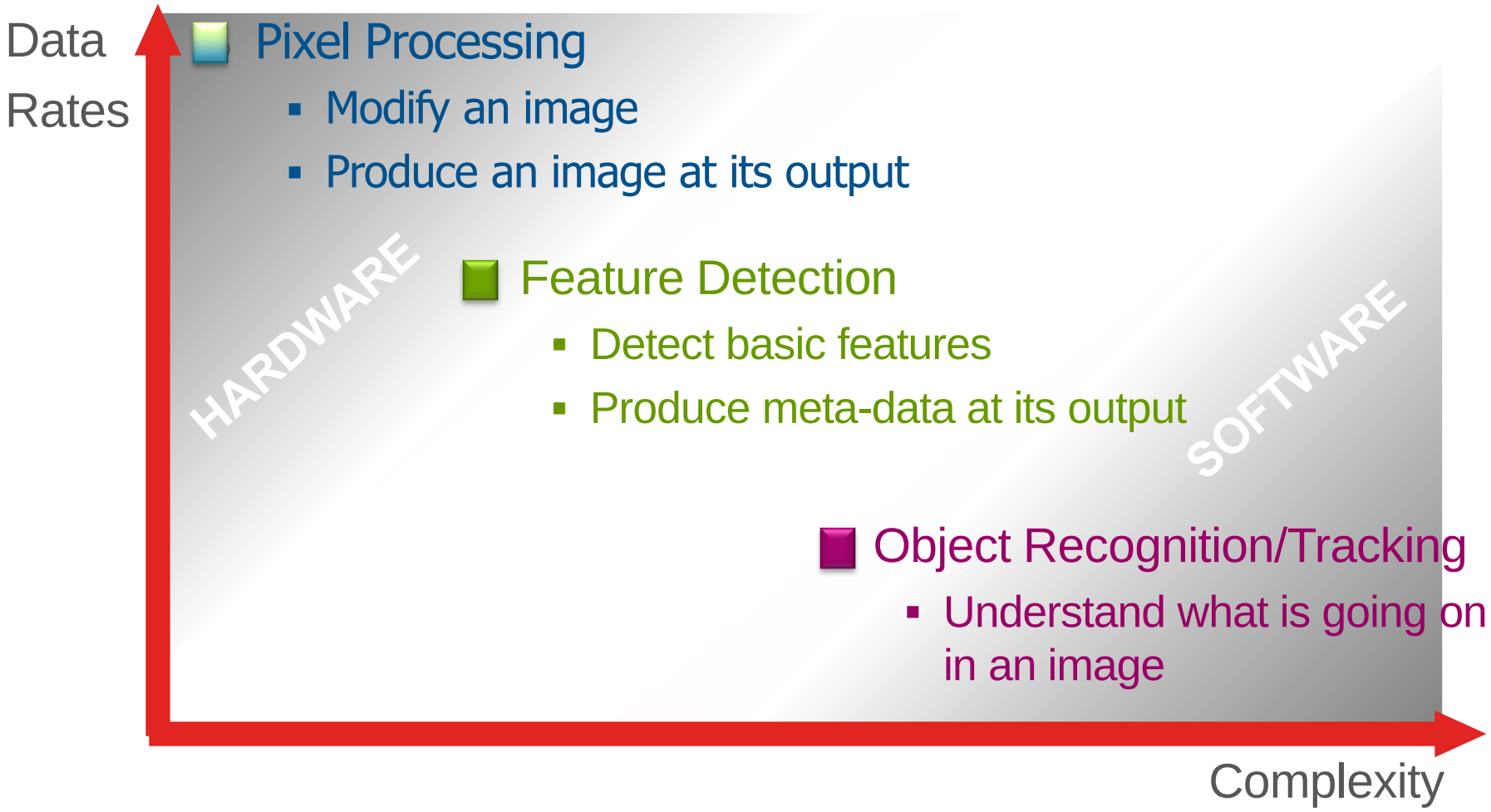
- Learn the latest developments in high pixel-rate video and image processing for Zynq®
- Become familiar with the MicroZed Embedded Vision Kit
- Explore the design techniques used by leading IP providers to deploy and accelerate smarter vision algorithms on Zynq
- Understand the advantages of system level design flows using high-level C-synthesis, OpenCV, and model-based design with MATLAB®/Simulink®

Agenda

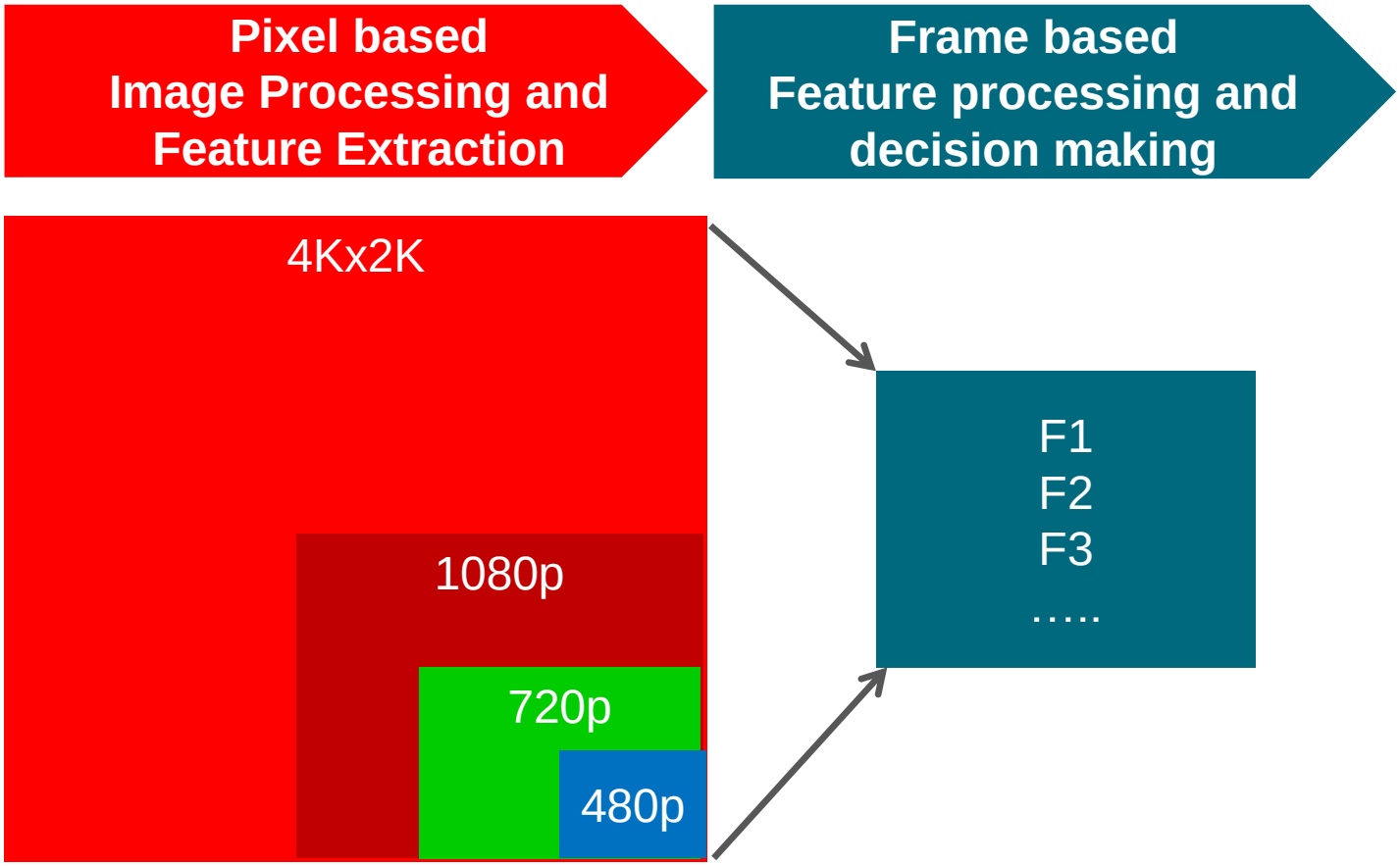
- Smarter Vision Concepts
- Overview of the MicroZed Embedded Vision Kit
- Accelerating Vision algorithms on Zynq
- System level design tools for Smarter Vision

Smarter Vision Concepts

Overview of Image/Vision Algorithms



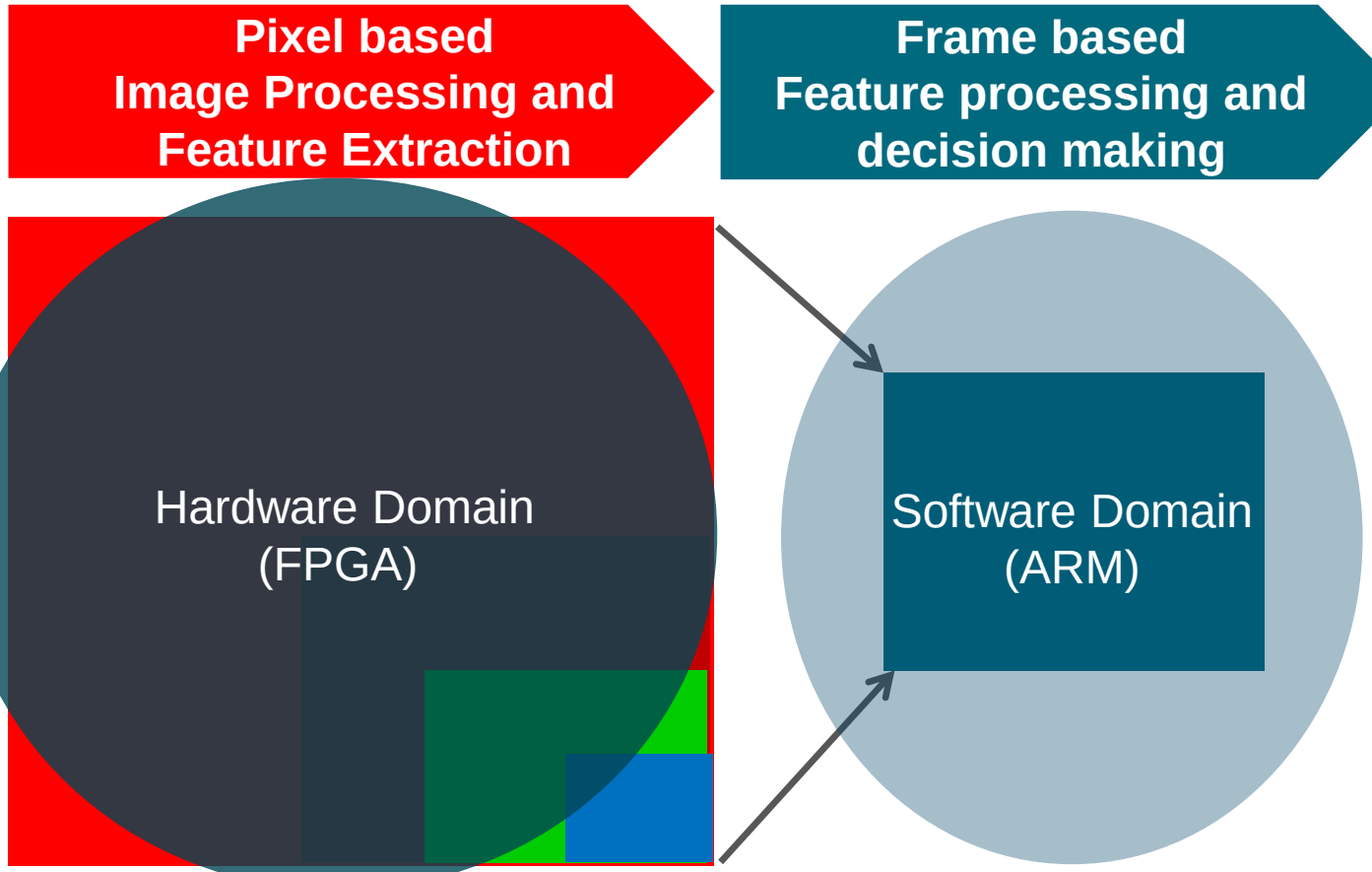
Real-time Video Analytics Processing



100's Ops/pixel
8MPx100 Pixels / frame
= 100's Gops

10000's Ops/feature
1000's of features/sec
= Mops

Heterogeneous Implementation of Real-time Video Analytics



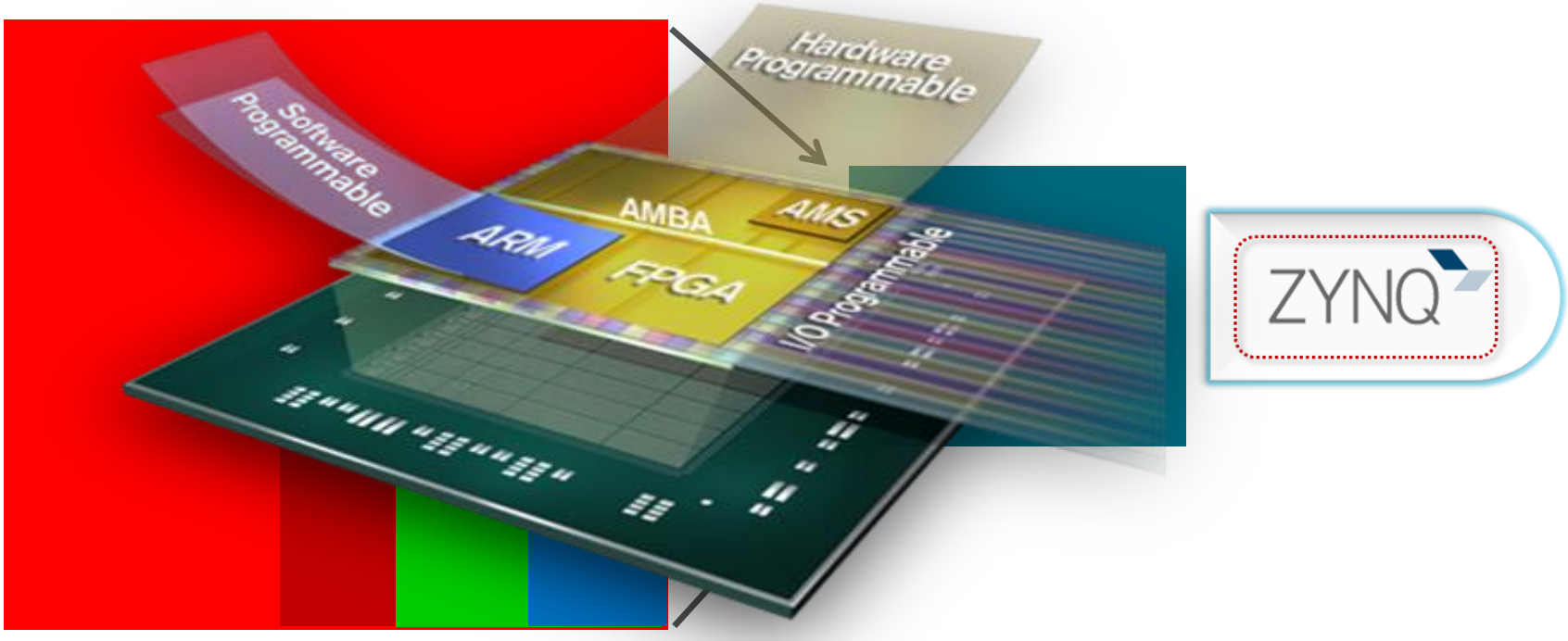
100's Ops/pixel
8MPx100 Pixels / frame
= 100's Gops

10000's Ops/feature
1000's of features/sec
= Mops

Xilinx Real-time Image Analytics Implementation: Zynq All Programmable SoC

Pixel based
Image Processing and
Feature Extraction

Frame based
Feature processing and
decision making



100's Ops/pixel
8MPx100 Pixels / frame
= 100's Gops

10000's Ops/feature
1000's of features/sec
= Mops

Overview of the MicroZed Embedded Vision Kit

MicroZed Embedded Vision Development Kit

- Hardware Overview
  - MicroZed 7020
  - Embedded Vision Carrier Card
    - HDMI input/output (Analog Devices)
    - Power over Ethernet (ST)
  - Connector for Camera Module
- Reference Designs
  - Getting Started / Tutorials
    - HDMI Pass-through Tutorial
    - HDMI Frame Buffer Tutorial
  - Advanced Designs
    - XAPP1167 – OpenCV on Zynq
- Supported by www.microzed.org

AES-MBCC-EMBV-DEV-KIT

HDMI Input/Output

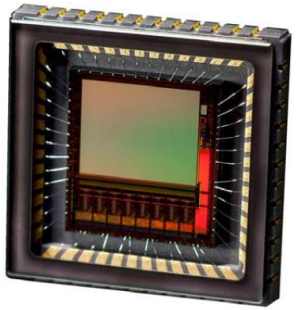
- HDMI Input Interface
  - Analog Devices ADV7611 (non-HDCP device)
    - 1080P60 capable
    - 16 bits YCbCr 4:2:2, embedded syncs
- HDMI Output Interface
  - Analog Devices ADV7511
    - 1080P60 capable
    - 16 bits YCbCr 4:2:2, embedded syncs
- Linux Drivers
  - Standard Linux drivers for video (V4L2, DRM/KMS)
  - Standard Linux drivers for audio (ALSA)

Power over Ethernet

- Power over Ethernet (PoE) Interface
  - ST Microelectronics PM8803
    - Single chip solution
    - IEEE 802.3at compliant PD interface
    - Current mode PWM controller
    - Flyback topology (lower cost, simpler design)
    - designed to work with power either from the Ethernet cable or from an external power source

PYTHON-1300 Camera Module

- Hardware Overview
  - PYTHON-1300 color sensor
    - 1280 x 1024 @ 210 frames/sec
    - 10 bits raw pixels (RGB bayer)
    - Pixel size = 4.8 µm x 4.8 µm
    - **Global shutter** and rolling shutter
  - Optics
    - C mount lens (2/3” optical size, 8mm)
    - IR cut filter
- Reference Designs
  - Camera Frame Buffer Tutorial



ON Semiconductor®

AES-CAM-ON-P1300C-G

PYTHON Image Sensor

- PYTHON family overview

Image Sensor	Active Area	Frame Rate	Throughput
PYTHON-300	640 x 480	840 fps	2.88 Gbps
PYTHON-500	800 x 600	560 fps	2.88 Gbps
PYTHON-1300	1280 x 1024	210 fps	2.88 Gbps
PYTHON-2000	1920 x 1200	245 fps	5.76 Gbps
PYTHON-5000	2592 x 2048	105 fps	5.76 Gbps
PYTHON-10K	3840 x 2896	180 fps	19.84 Gbps
PYTHON-12K	4096 x 3072	160 fps	19.84 Gbps
PYTHON-16K	4096 x 4096	120 fps	19.84 Gbps
PYTHON-25K	5120 x 5120	75 fps	19.84 Gbps

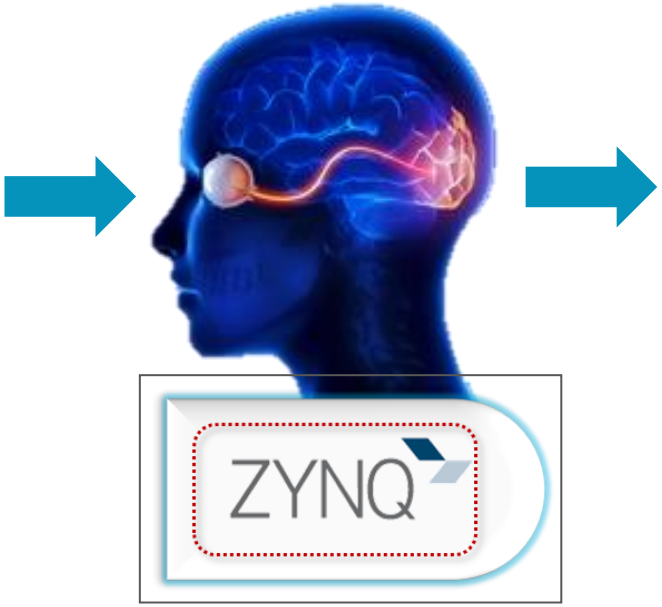
ON Semiconductor®

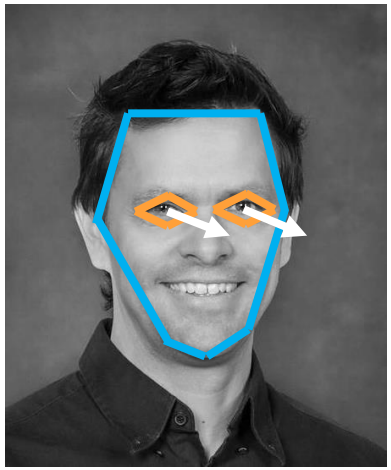
ISP8 pin compatible package

ISP32

MicroZed Embedded Vision Kit - Demonstration

- Face Analytics
  - Algorithm by Visage Technologies
  - Zynq implementation by XYLON





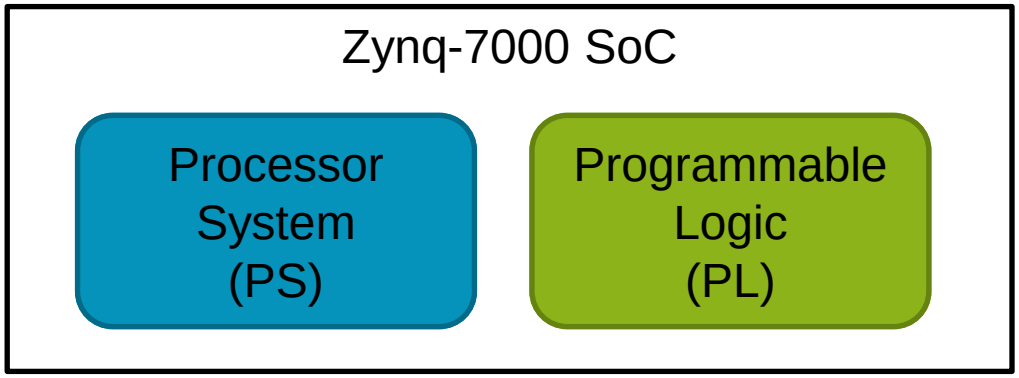
face presence/position

eyes open/closed

direction of gaze

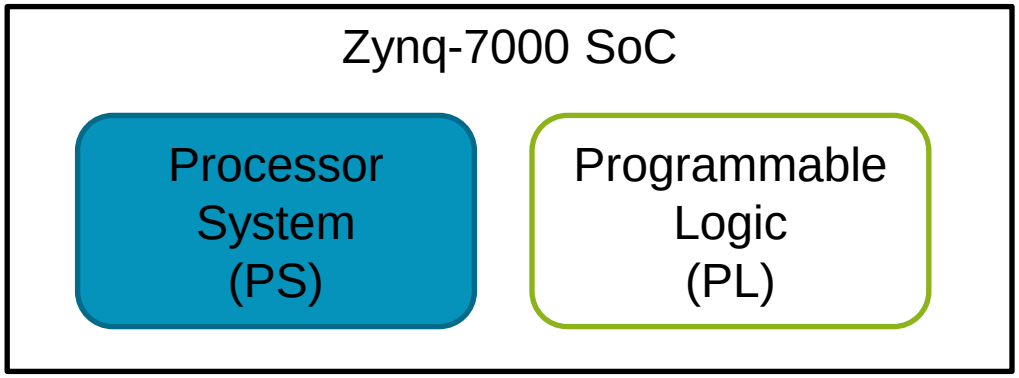
Accelerating Vision algorithms on Zynq

Accelerating Vision algorithms on Zynq



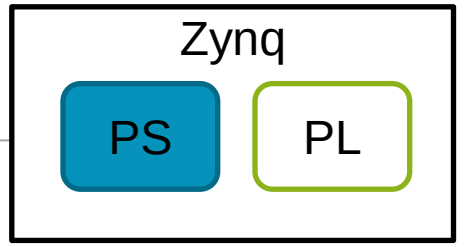
- Acceleration can be achieved
  - in the Processor System (PS)
    - 1-2 Gops
  - in the Programmable Logic (PL)
    - 10-500 Gops

Accelerating Vision algorithms on Zynq



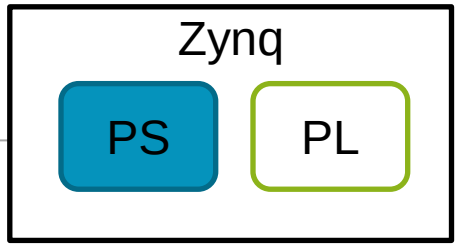
- Acceleration in the Processor System (PS)
  - Improving Memory Access Efficiency
  - Optimizing performance with NEON
  - Using NEON optimized libraries

Improving Memory Access Efficiency

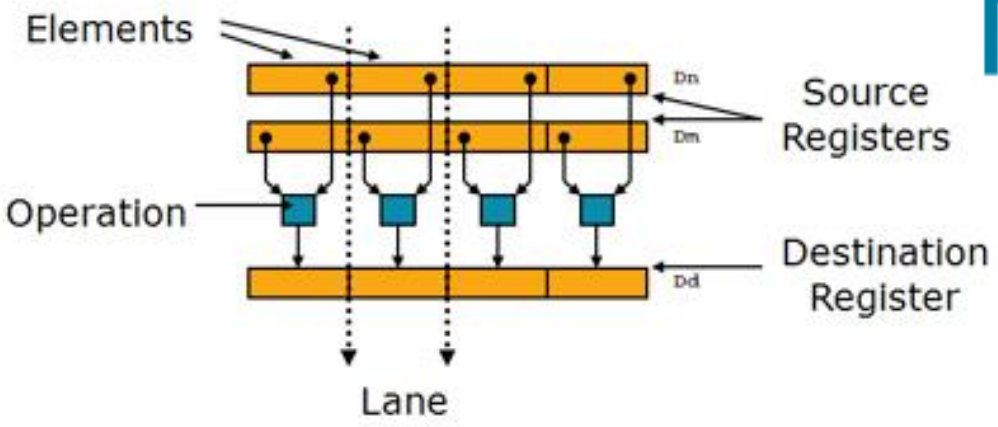


- Challenge
  - Images are stored in external memory
  - Internal memory is typically only 1-10 Mbits
  - 1080P60 images require 1920x1080 x 24bits = 45 Mbits
- Solution
  - Implement Data Locality
    - Combine multiple loops into single loop
    - Work on image blocks (ie. block processing)
      - size of internal memory
      - size of data cache
    - Use DMA to load/store image blocks from/to external memory

Overview of NEON

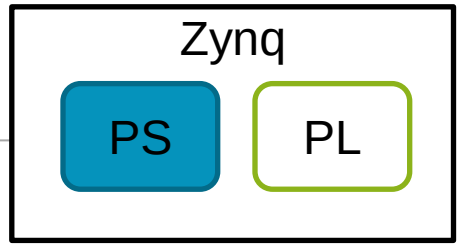


- Performs “Packed SIMD" processing
  - Registers are considered as **vectors** of **elements** of the same **data type**
  - Data types can be: signed/unsigned 8-bit, 16-bit, 32-bit, 64-bit, single precision floating point
  - Instructions perform the same **operation** in all (4) **lanes**
- 4X peak performance increase



NEON™

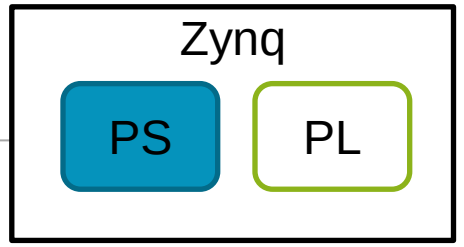
Compiler Automatic Vectorization



- GCC Compiler Options
  - -O3
  - -mcpu=cortex-a9
  - -mfpu=neon
  - -ftree-vectorize : loop vectorization on trees
  - -mvectorize-with-neon-quad : vectorize for quad-word
  - -ffast-math : faster math routines (not IEEE/ISO compliant)

- Example
  - OpenCV face detector => 2X faster with “–mfpu=neon”
    http://www.embedded-vision.com/forums/2011/09/15/opencv-arm-neon-accelerating

Compiler Automatic Vectorization



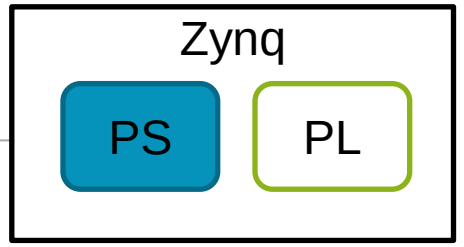
- C Code Modifications
  - Indicate the number of loop iterations

```
for ( i = 0; i < (M*N); i++ ) {
   out[i] = in[i] * gain;
}
```

```
for ( i = 0; i < ((M*N) & ~3); i++ ) {
   out[i] = in[i] * gain;
}
```

  - Avoid conditions inside loops
  - Avoid loop-carried dependencies

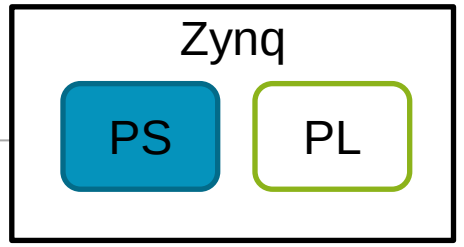
Using NEON Intrinsics



- Using NEON Intrinsics
  - Standardized by ACLE (ARM C language extension)
  - Accessed by including <arm_neon.h> header file
  - Format : V<op>.<dt> <dest> ,<src1>, <src2>
    https://gcc.gnu.org/onlinedocs/gcc/ARM-NEON-Intrinsics.html

- Example - NE10 (ARM)
  - ARM accelerated two OpenCV functions
    - imresize : image resizing / video scaling
    - imrotate : image rotation
  - Code had to be re-written for data locality & NEON
    https://github.com/projectNe10

Use Case – unCannyCV Library

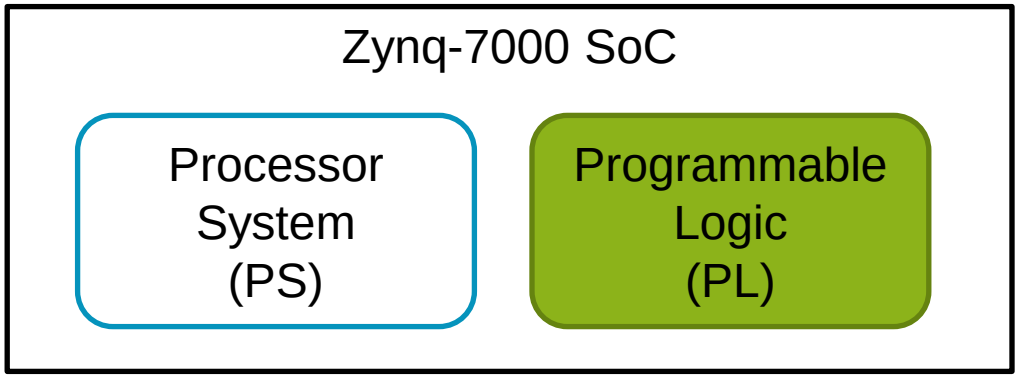


- unCannyCV (unCannyVision)
  - Code re-written for data locality
  - Explicit use of cache preload engine
  - Code explicitly written for NEON acceleration

- Acceleration achieved
  - **4.7x** for Canny Edge Detection
  - **9.4x** for Sobel Edge Detector
  - **11.38x** for 3x3 Convolution Filter
  - **15.5x** for 5x5 Convolution Filter

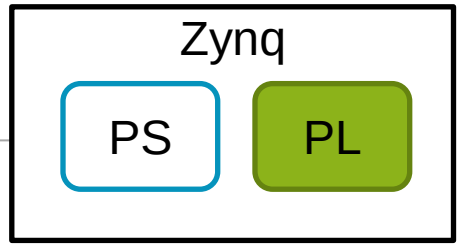
} VGA on 1GHz ARM

Accelerating Vision algorithms on Zynq



- Acceleration in the Programmable Logic (PL)
  - Overview of hardware resources for video
  - Overview of hardware architectures
  - Using hardware optimized video IP

Overview of hardware resources

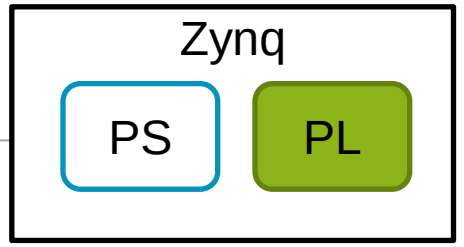


- Programmable Logic (PL) resources
  - Video processing : Logic Cells, DSP Slices
  - Video storage
    - Local (line buffers) : Block RAM
    - External (frame buffers) : DDR3 Interface

Resource	Zynq-7020	Zynq-7100
Fabric	Artix-7	Kintex-7
Logic Cells	85K	444K
DSP Slices	220 (276 GMACs)	2020 (2622 GMACs)
Block RAMs (32Kbits each)	140 (4Mbits)	755 (24Mbits)
DDR3 Interface	1,066 Mbps	1,066 Mbps

1080P60 4K

Overview of hardware resources

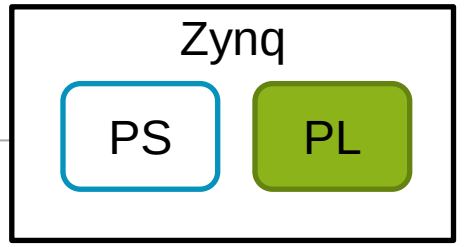


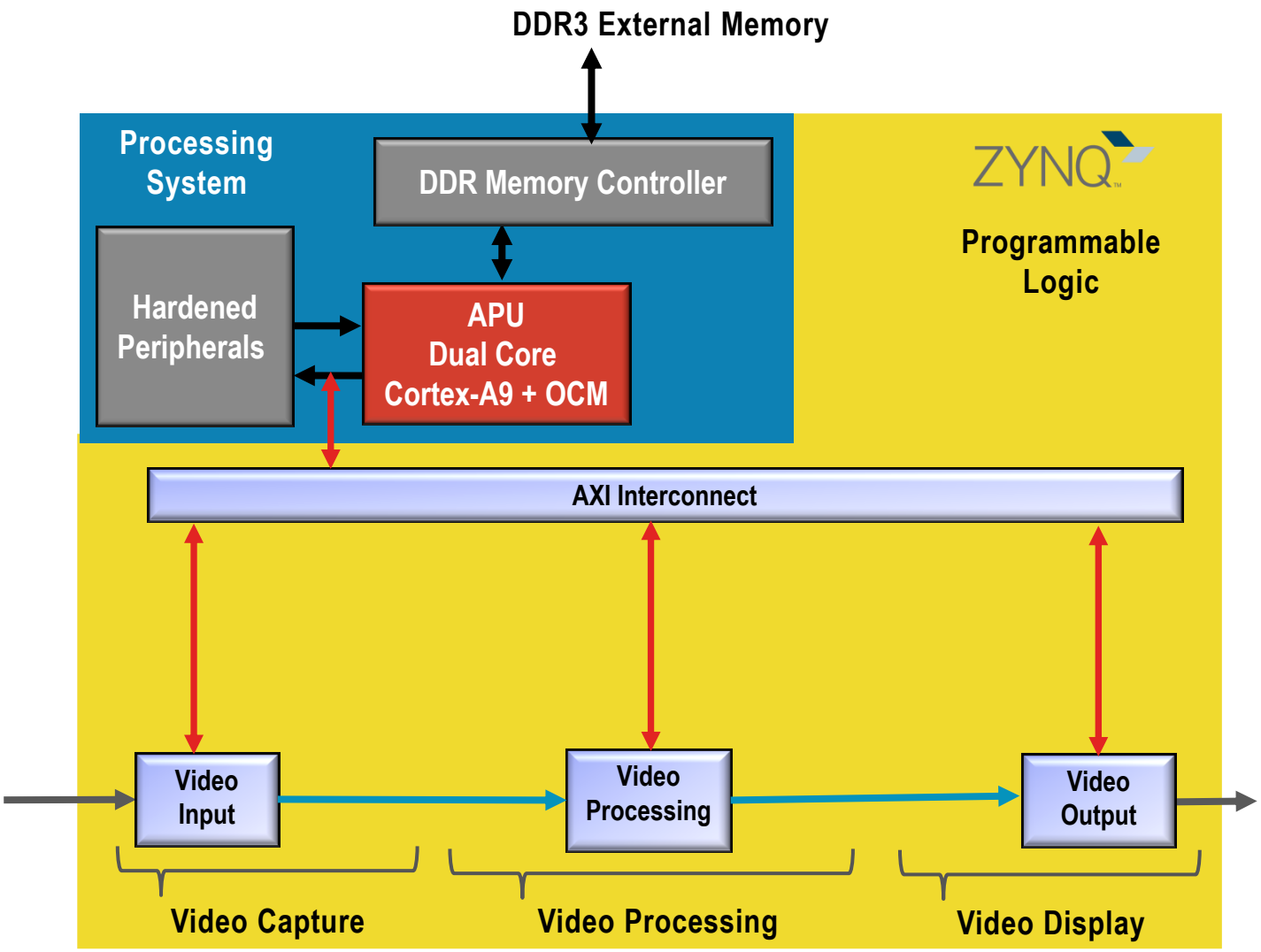
- AXI Interconnect resources
  - Industry standard interconnect (ARM)
  - Ensures plug-and-play between IP cores

AXI Interconnect	Recommended Usage
AXI4-Lite	Control Interface for IP cores
AXI4-Memory Mapped	High Performance access to external DDR3 memory
AXI4-Stream	Point to point interconnect for data (video) streams. Supports back pressure.

- AXI Video DMA
  - Transfer images between PL and shared DDR3

Direct streaming architecture

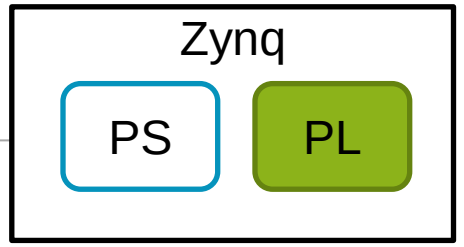


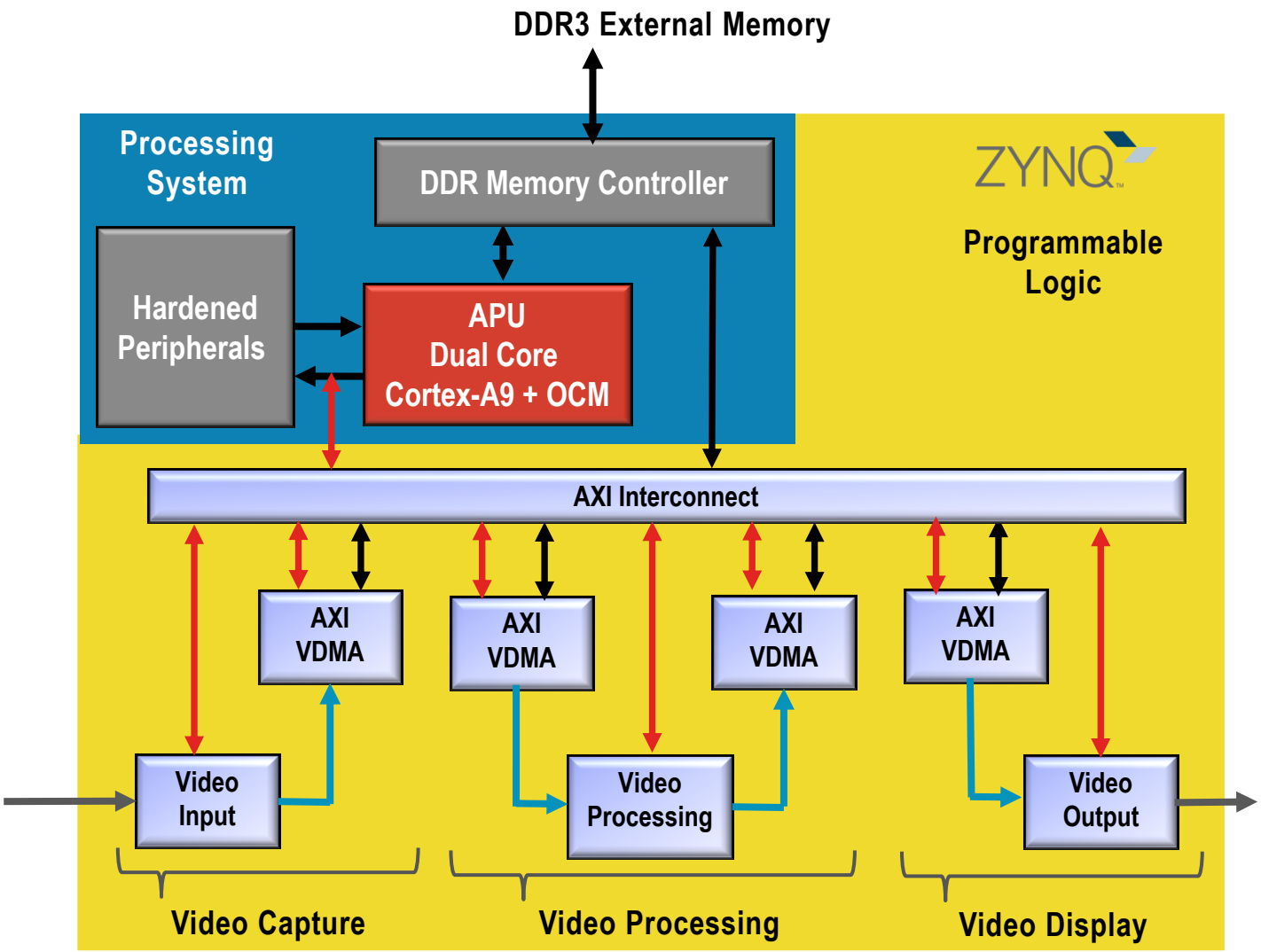


LEGEND:

- Video I/O
- AXI4-Stream
- AXI4-Lite

Frame Buffer Architecture



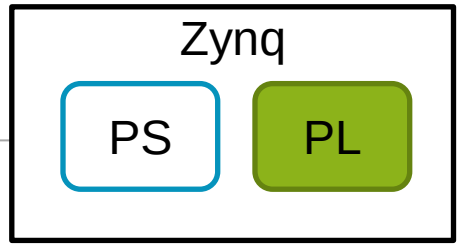


DDR3 Size
- 1 GBytes
- 128 x 1080P60 (32 bit pixels)

DDR3 Bandwidth
- 32 Gbps
- 8 x 1080P60 (32 bit pixels)

LEGEND:
- Video I/O
- AXI4-Stream
- AXI4-Lite
- AXI4-MM

Using hardware optimized video IP

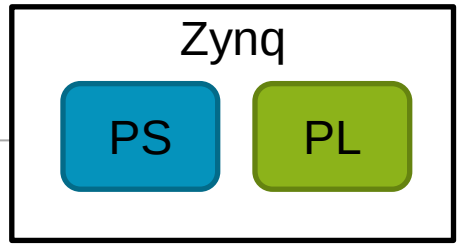


- Xilinx – OpenCV functions
  - XAPP1167 : Targeting OpenCV on Zynq

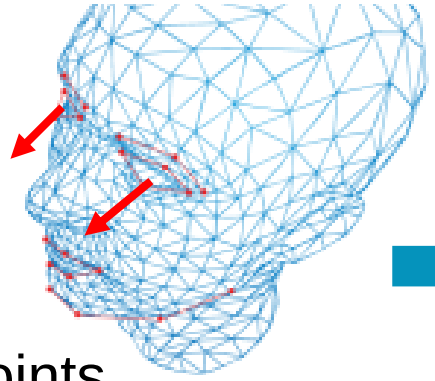
- Acceleration achieved
  - **8x** for Pass-Through
  - **60x** for Fast Corners
  - **78x** for 3x3 Convolution Filter (sobel)
  - **72x** for 5x5 Convolution Filter (gaussian)
  - **150x** for Canny Edge Detection

1080P60
on Zynq-7020

Using hardware optimized video IP

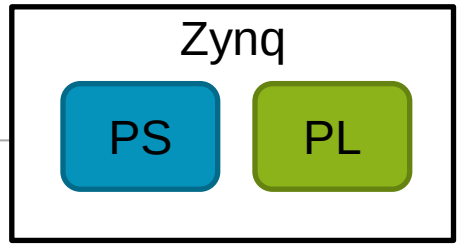


- Xylon - Face Analytics
  - Algorithm from Visage Technologies
    - 3D head pose
    - facial expressions
    - eye closure
    - gaze direction
    - 2D/3D facial feature points
    - Textured 3D face model
  - Xylon ported algorithm to Zynq

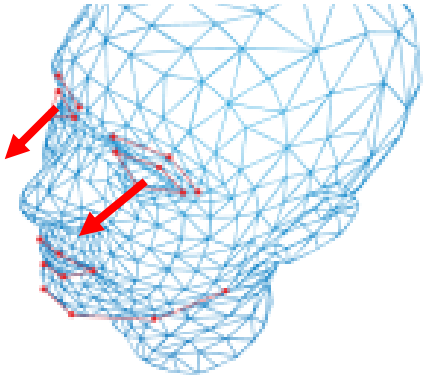


http://www.visagetechnologies.com

Using hardware optimized video IP



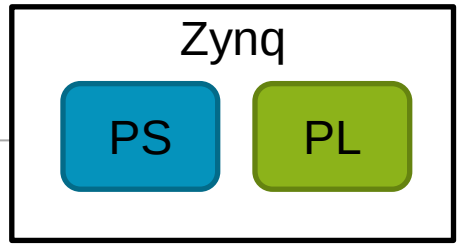
- Xylon - Face Analytics
  - Xylon ported Visage Technologies algorithm to Zynq
  - Resolution achieved :
    - 1280x1024 @ 30fps
  - Hardware acceleration :
    - PL : Image pre-processing
    - PL : Video scaling down to < VGA
    - PS : Complex Analytics



logicBRICKS™ Designed by XYLON

http://www.logicbricks.com/Solutions/Xylon-Face-Detection-Tracing-on-Xilinx-Zynq-SoC.aspx

Using hardware optimized video IP



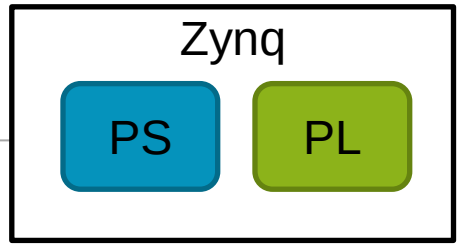
- Van Gogh Imaging – Starry Night
  - Middleware for 3D Scanner
    - Capture, Recognize, Analyze objects in a 3D scene

Example:
Locate Toy Bear

VanGogh Advantages

- Fully Automated Process
- No Manual Post Processing
- Accurate Fully-Formed 3D Model
- Superior Noise Tolerance

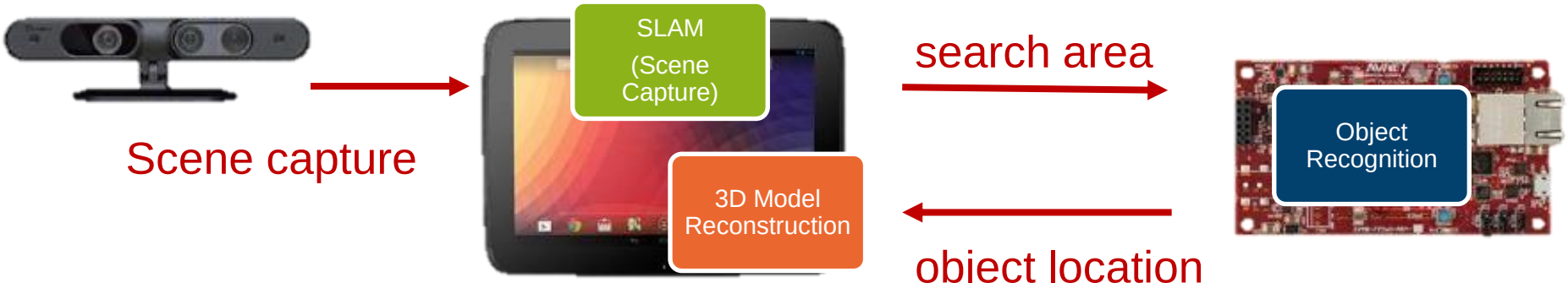
Using hardware optimized video IP



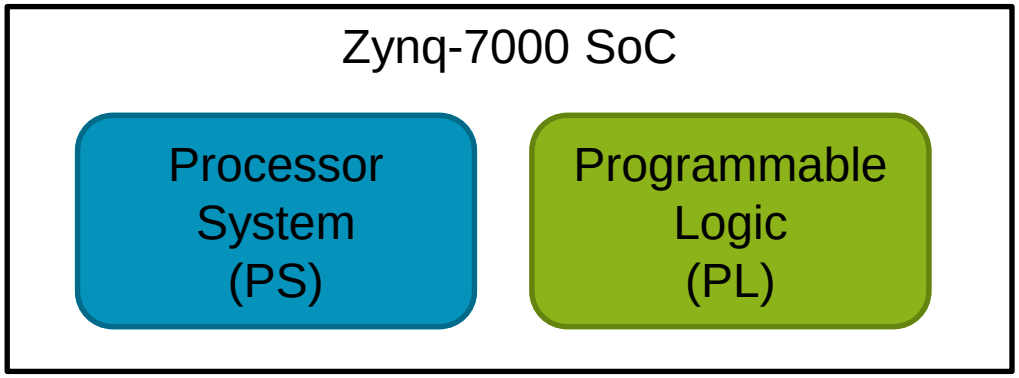
- Van Gogh Imaging – Starry Night
  - Resolution : 320x240
  - Acceleration achieved = 10x faster with MicroZed
    - Tablet : SLAM, 3D Model Reconstruction
    - Zynq : Object Recognition
      - PS : Data Management + Floating-Point
      - PL : Matrix Operations

VanGogh Imaging

http://vangoghimaging.com



Summary



- Acceleration can be achieved
  - in the Processor System (PS)
    - re-written code + NEON => VGA @ 30 fps
  - in the Programmable Logic (PL)
    - Zynq-7020 (Artix-7) => 1080P @ 60 fps
    - Zynq-7045 (Kintex-7) => 8 * 1080P60, 4K2K

1280x1024 @ 30 fps

System level design tools for Smarter Vision

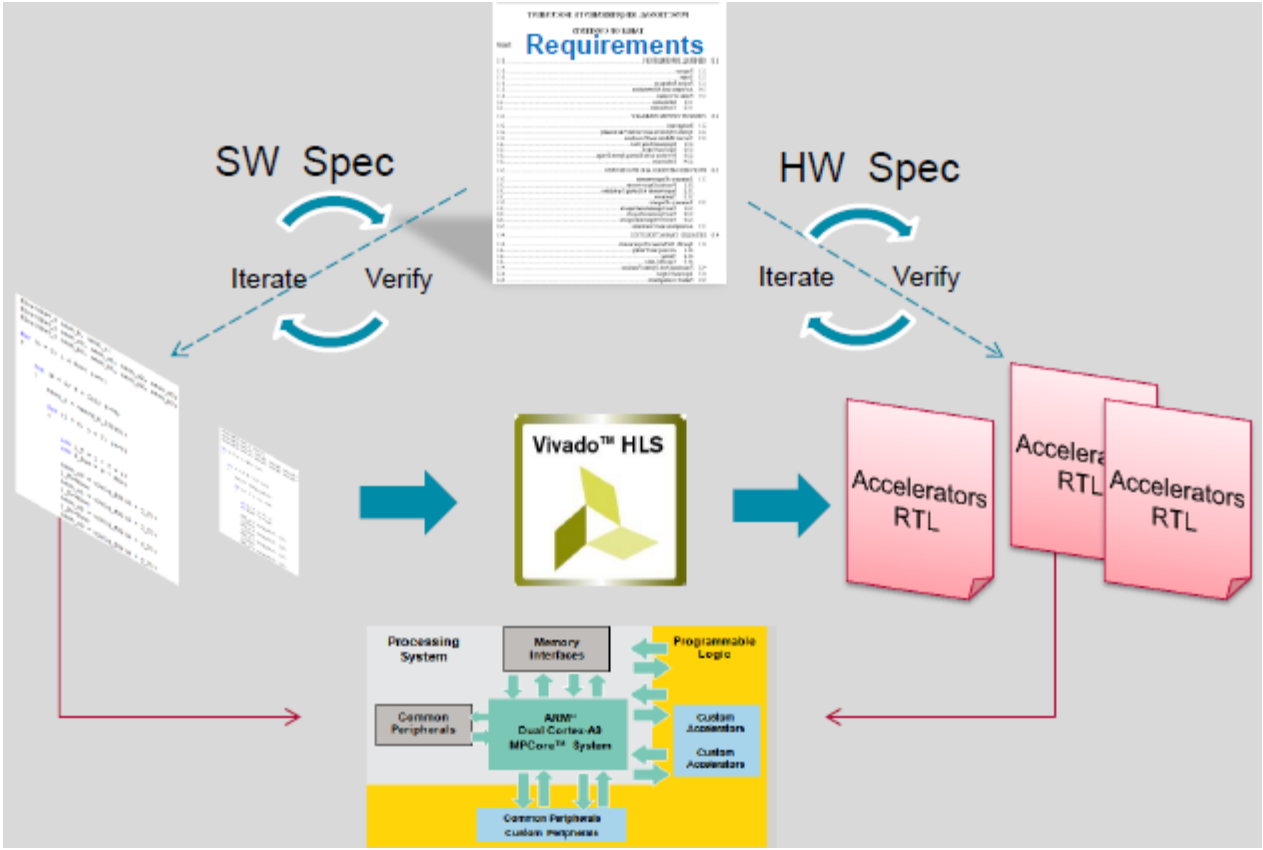
System level design tools for Smarter Vision

- Xilinx - Vivado High-Level Synthesis (HLS)
  - C to RTL Synthesis
  - Optimized video libraries
  - Native vs Accelerated OpenCV application

- Mathworks – Model-Based Design
  - Model-Based Design Flow
  - Model-Based Design Tools
  - IP Core Generation

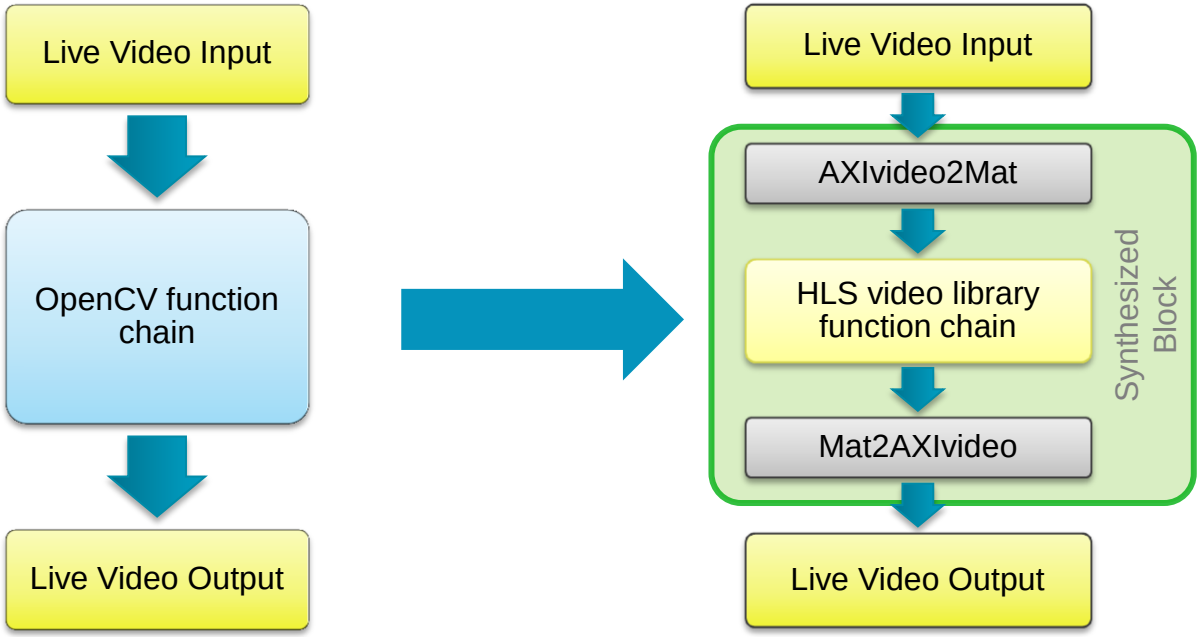
Vivado High-Level C-Synthesis (HLS)

- C to RTL synthesis
  - Hardware accelerator created with software environment



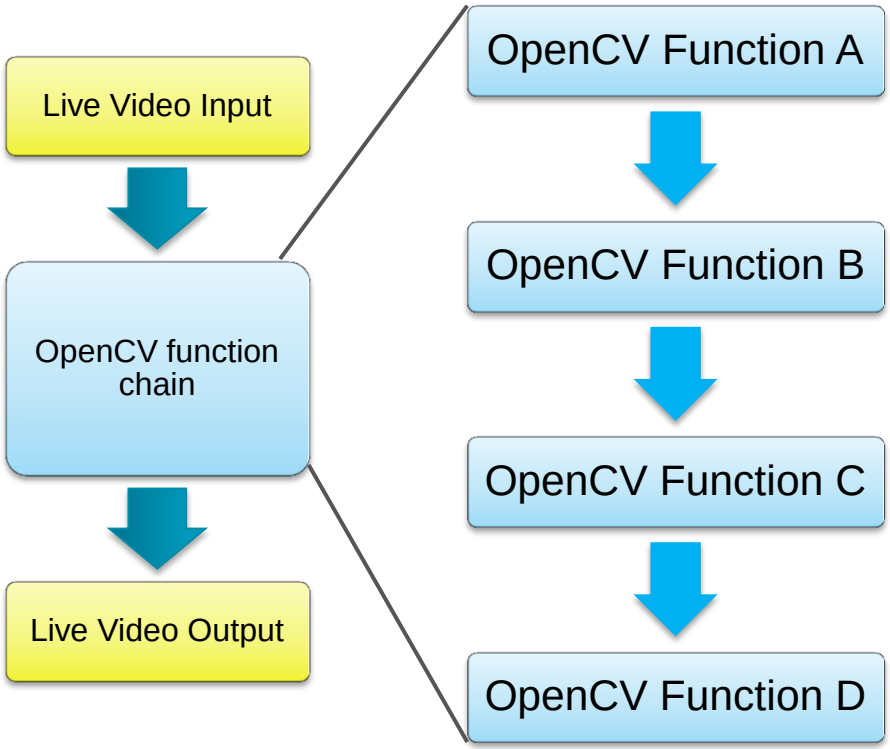
Vivado High-Level C-Synthesis (HLS)

- Optimized video libraries
  - OpenCV functions, re-written for Vivado HLS
  - Pipelined Streaming Architecture (ie. data locality)



Native OpenCV Application

- OpenCV makes extensive use of external memory
  - each function loops over the entire image



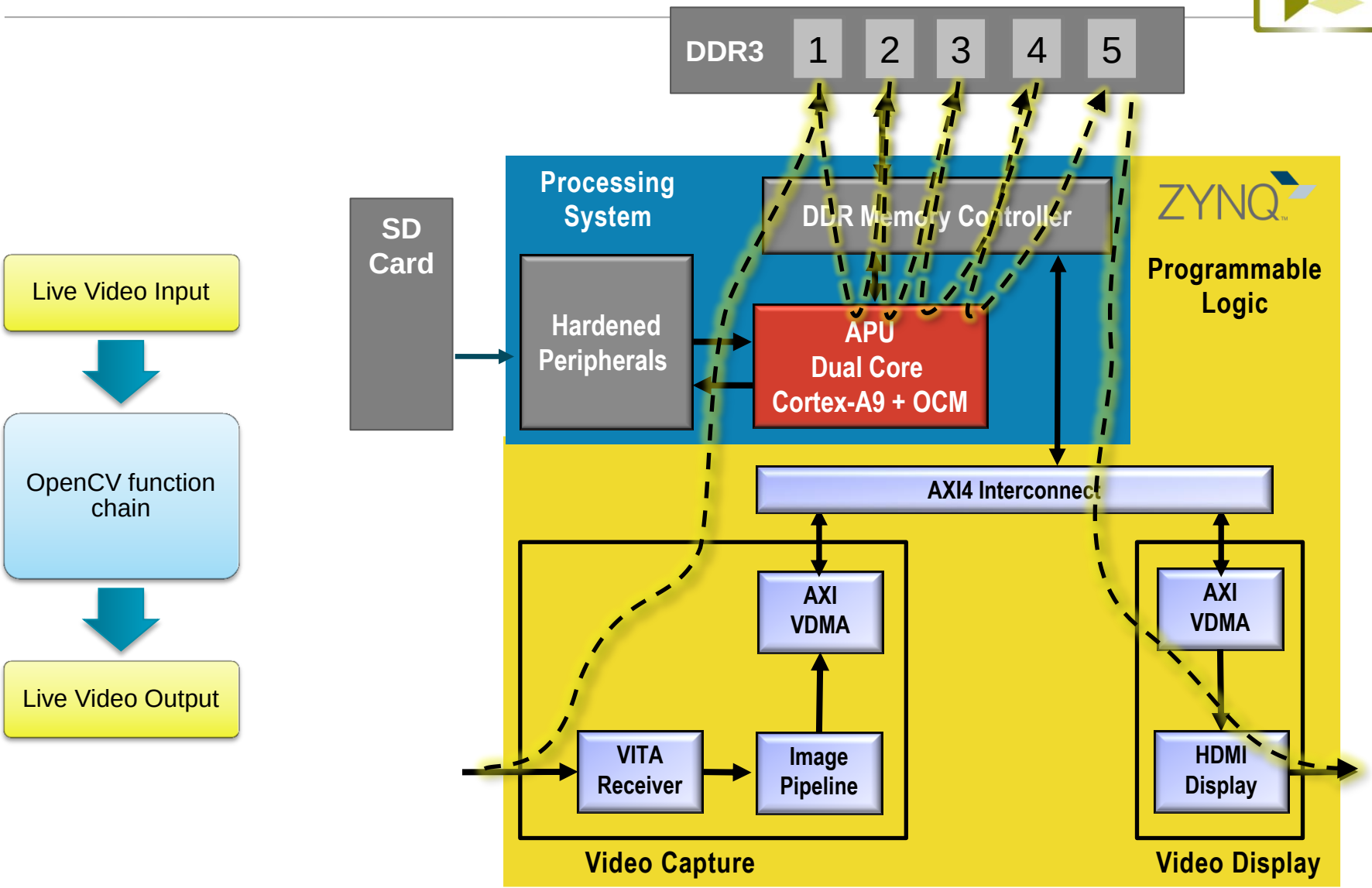

```
for ( i = 0; i < (M*N); i++ ) {
   out[i] = func_A( in[i] );
}
```

```
for ( i = 0; i < (M*N); i++ ) {
   out[i] = func_B( in[i] );
}
```

```
for ( i = 0; i < (M*N); i++ ) {
   out[i] = func_C( in[i] );
}
```

```
for ( i = 0; i < (M*N); i++ ) {
   out[i] = func_D( in[i] );
}
```

Native OpenCV Application



Accelerated OpenCV Application

- Vivado HLS automatically pipelines loops

```
for ( i = 0; i < (M*N); i++ ) {
   out[i] = func_A( in[i] );
}
```

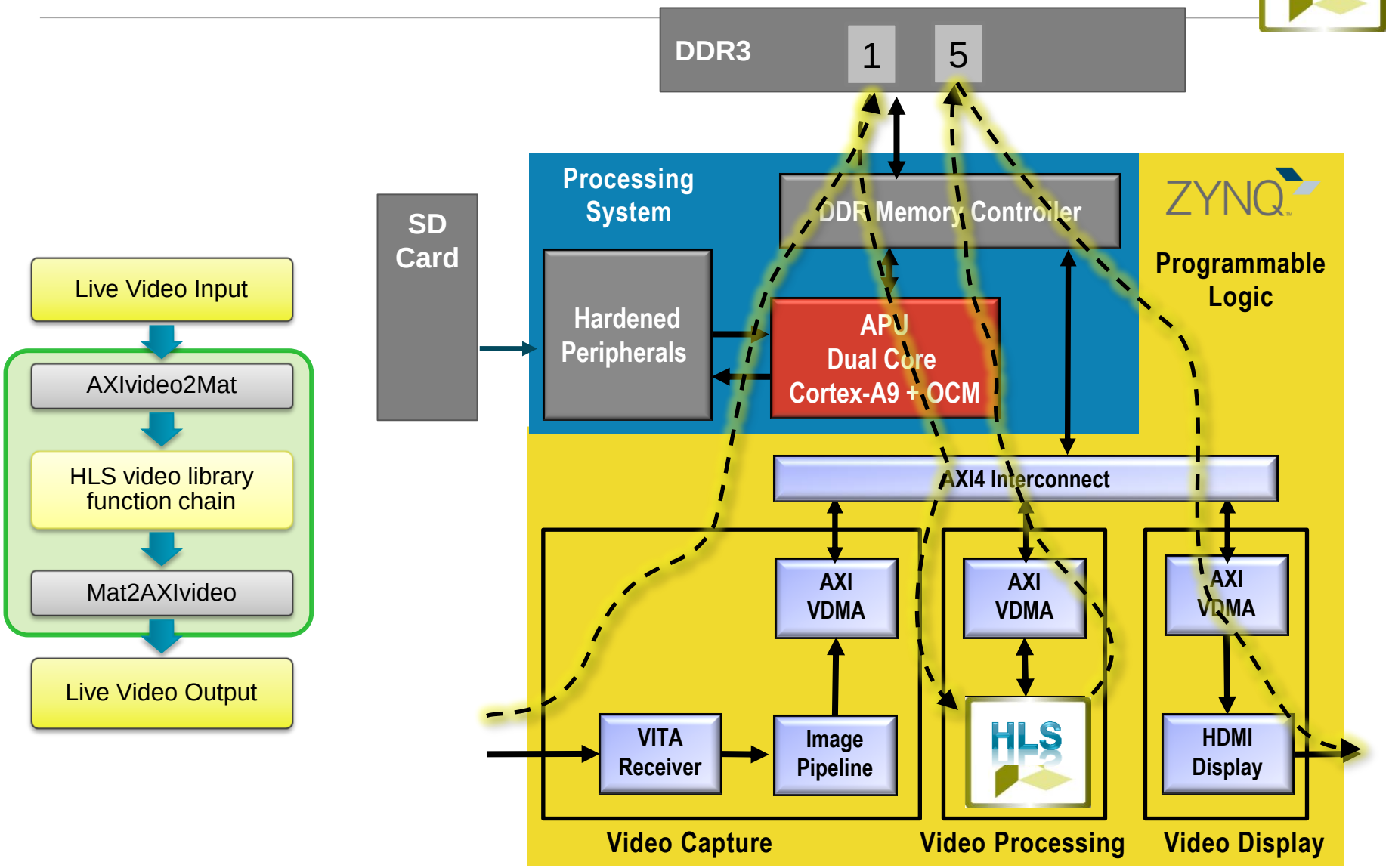
```
for ( i = 0; i < (M*N); i++ ) {
   out[i] = func_B( in[i] );
}
```

```
for ( i = 0; i < (M*N); i++ ) {
   out[i] = func_C( in[i] );
}
```

```
for ( i = 0; i < (M*N); i++ ) {
   out[i] = func_D( in[i] );
}
```

```
for ( i = 0; i < (M*N); i++ ) {
   out1[i] = func_A( in[i] );
   out2[i] = func_B( out1[i] );
   out3[i] = func_C( out2[i] );
   out4[i] = func_D( out3[i] );
}
```

Accelerated OpenCV Application



Xilinx HLS Video Library 2014.1

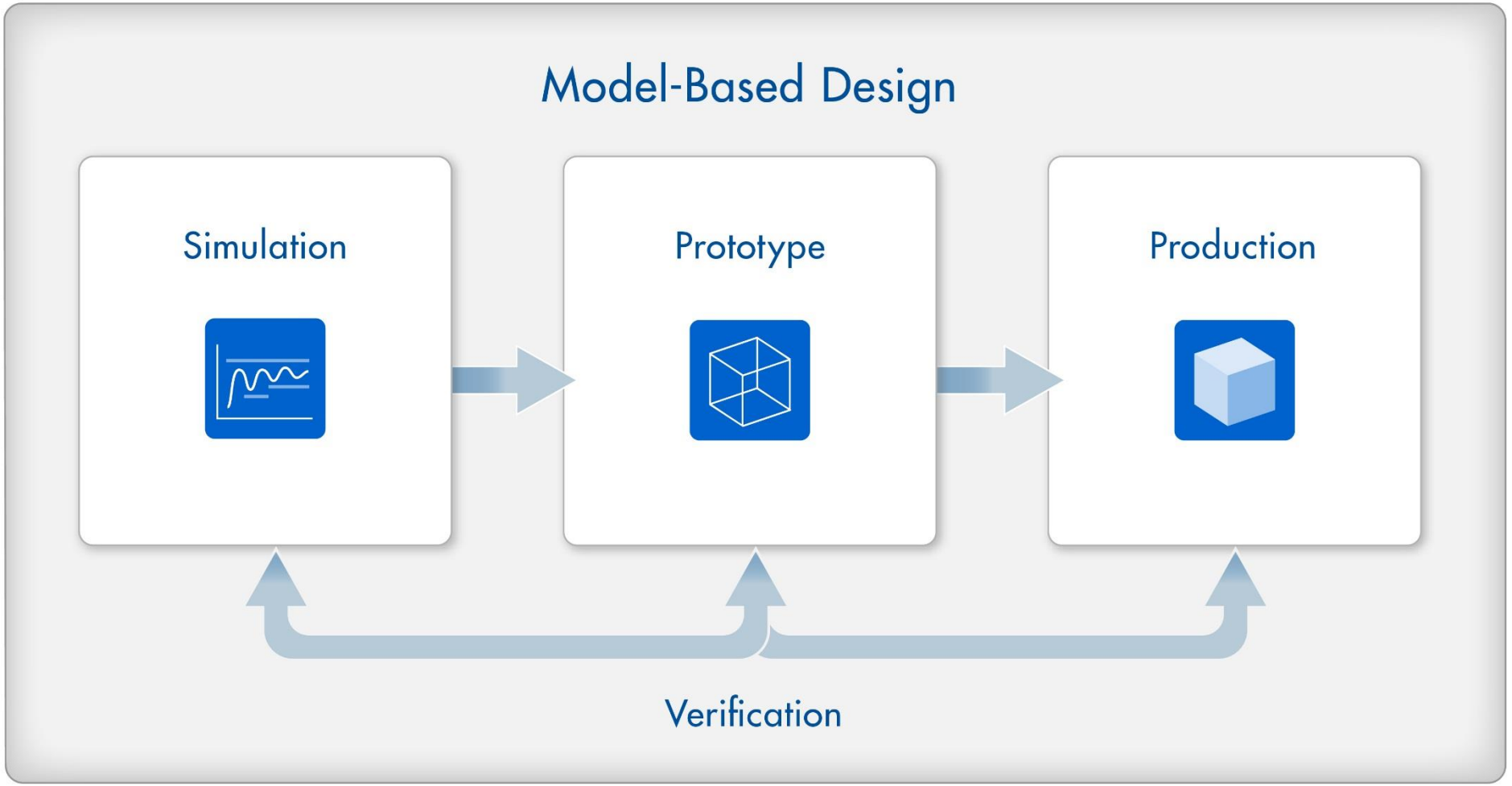
- UG 902 – Vivado HLS User Guide
  - Video Processing Functions
    - Compatible with standard OpenCV functions
    - Not direct replacements for OpenCV functions

Video Functions			
AbsDiff	Duplicate	MaxS	Remap
AddS	EqualizeHist	Mean	Resize
AddWeighted	Erode	Merge	Scale
And	FASTX	Min	Set
Avg	Filter2D	MinMaxLoc	Sobel
AvgSdv	GaussianBlur	MinS	Split
Cmp	Harris	Mul	SubRS
CmpS	HoughLines2	Not	SubS
CornerHarris	Integral	PaintMask	Sum
CvtColor	InitUndistortRectifyMap	Range	Threshold
Dilate	Max	Reduce	Zero

Vivado HLS Summary

- Optimized video libraries
  - Accelerates OpenCV functions to programmable logic
- Pipelined streaming architecture
  - Efficiently implement real-time video processing
  - Throughput stays constant with more complex programs
- Zynq offers power-optimized integrated solution
  - Less power than 2 chip solution
  - Pipelined architecture reduces ext. memory bandwidth

Model-Based Design Flow



Model-Based Design Flow

Simulation

Prototype

Production

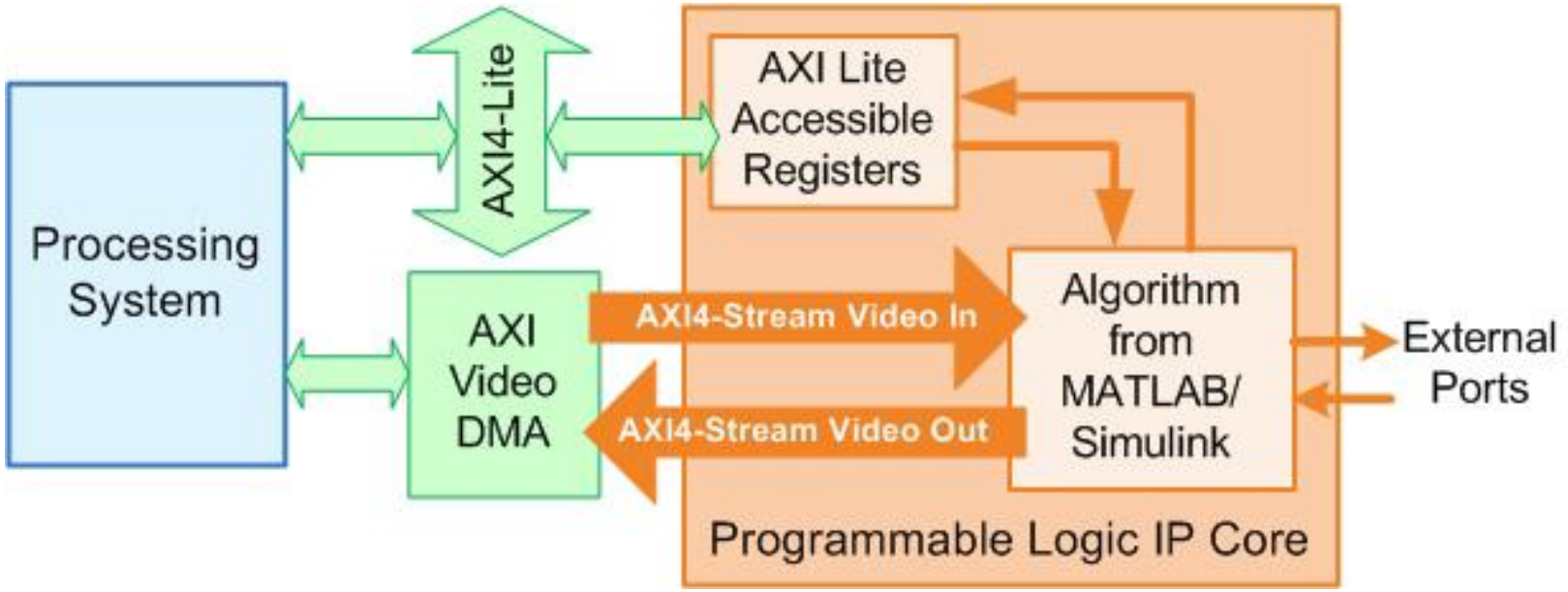
- **Verify operation before committing to hardware**
  - Model complex multidomain systems
  - Evaluate design options
  - Create system-level test benches
- **Validate performance on chip**
  - Convert floating-point designs to fixed-point implementations
  - Generate IP cores from Simulink models
  - Build Linux executables targeting ARM on Zynq
- **Deploy design on target system**
  - Export IP to downstream Xilinx tools for system integration
  - Support workflow certification

Model-Based Design Tools

- Model-based design with Matlab/Simulink
  - Computer Vision System Toolbox
  - Embedded Coder
    - Generates C code for the ARM processors (PS)
    - Block Processing,
      allows chunks of larger image to be processed in local memory
    - Code Replace Library for ARM Cortex-A Processors,
      makes use of the NE10 library for NEON acceleration
    - Multi-tasking, allows model to be executed on both ARM cores for Linux SMP
  - HDL Coder
    - Generates VHDL/Verilog code for the programmable logic (PL)
    - Provides high-level modeling for image and video processing applications

Model-Based Design Tools

- IP Core Generator
  - Supports AXI4-Lite interconnect for control signal and parameter tuning
  - Supports AXI4-Stream to transport video content
  - Makes use of AXI Video DMA to access video content



Conclusion

Next Steps

- X-fest 2014 Slides
  - www.xfest2014.com
- MicroZed Embedded Vision Kit
  - www.microzed.org => Products
- NEON optimization on Zynq
  - XAPP1206 – Boost Zynq performance with NEON
- Vivado HLS + optimized video libraries
  - XAPP1167 – Targeting OpenCV on Zynq
  - **Smarter Vision Design Seminar**
    www.microzed.org => Training and Videos


Thank
You
Maake
Asante
Kiitos
köszönöm
спасибо
Dankscheen
Vinaka
Shukria
Dhanyavadagalu
Manana
Dankon
Kam Sah Hammida
감사합니다
Dank Je
Maururu
Biyan
Matondo
Terima Kasih
Taiku
Tack
Diolch i Chi
Chokrane
Blagodaram
Ngiyabonga
Dziekuje
Juspaxar
Arigato
Grazie
Mochchakkeram
Gracias
Tingki
Paldies
Khap
Khun
Kop
Kia Ora
Djiere Dieuf
Obrigado
Gratias Tibi
cảm ơn bạn
ধন্যবাদ
நன்றி
Ua Tsaug Rau Koj
Bedankt
Dakujem
Grazas
Nirringrazzjak
Děkuji
Suksama
Hvala
Welalin
Di Ou Mèsi
Merci
Go Raibh Maith Agat
ขอขอบคุณคุณ
Salamat
Najis Tuke
Eskerrik Asko
Misaotra
Rahmat
Nuwun
Matur
谢
谢
хвала
Danke
ありがとう