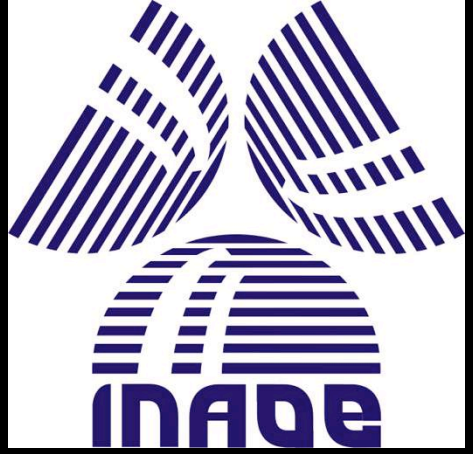


Reconfigurable and High Performance Computing Lab
INAOE – Puebla, Mexico

Embedded vision with FPGA vs CUDA processing. Directions and platform proposal

WASC 2014
20 June 2014

Dr. Miguel Arias Estrada
ariasmo@inaoep.mx

Content

1. Introduction
2. Previous work on FPGA architectures
3. FPGA cameras
4. Platform proposal: FPGA vs CUDA
5. Long term project
6. Conclusions

Reconfigurable and High Performance Computing Laboratory

- Computer Science Department
  - 4 Researchers
  - 10+ M.Sc. Students
  - 5+ Ph.D. Students
- Active since1998
- Research on:
  - Real time computer vision
  - Criptography and Cipher
  - Hardware Signal Processing

1. Introduction

Smart camera approach

High performance low level vision computing at camera

3D Vision, tracking / surveillance applications

2. Previous work on FPGA architectures

Edge / Corner Detection

Stereo disparity

Target Tracking

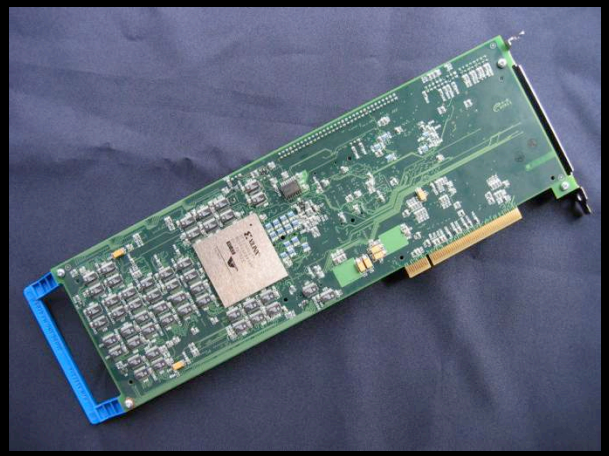
Motion correlation and Optical Flow

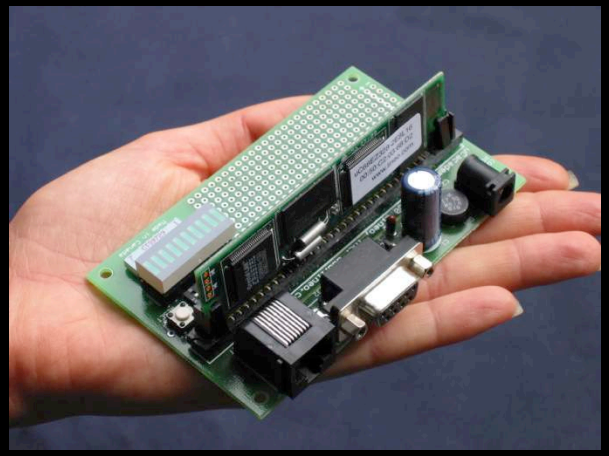
3D from Optical Flow

SIFT / SURF / LISF feature detection

Approach

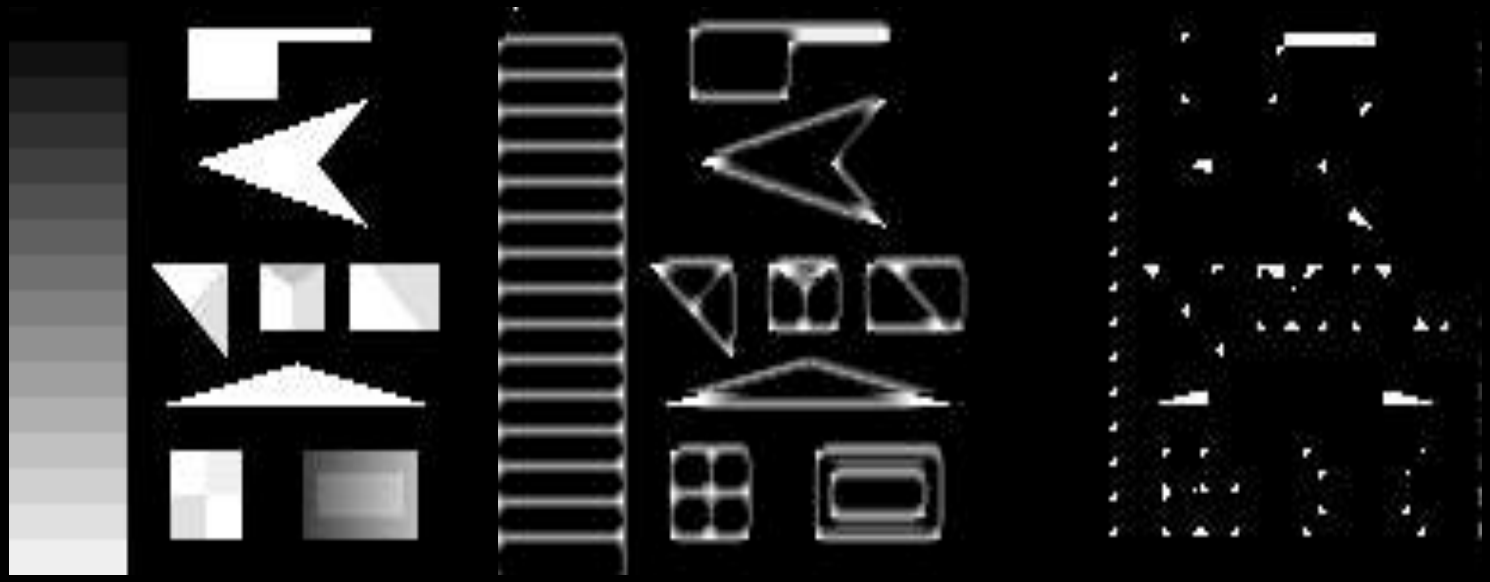
- Off-the-shelf development boards
- Focus on FPGA architecture. Application can be built in parallel
- Goal: Reach video rate processing (i.e. 30 fps)



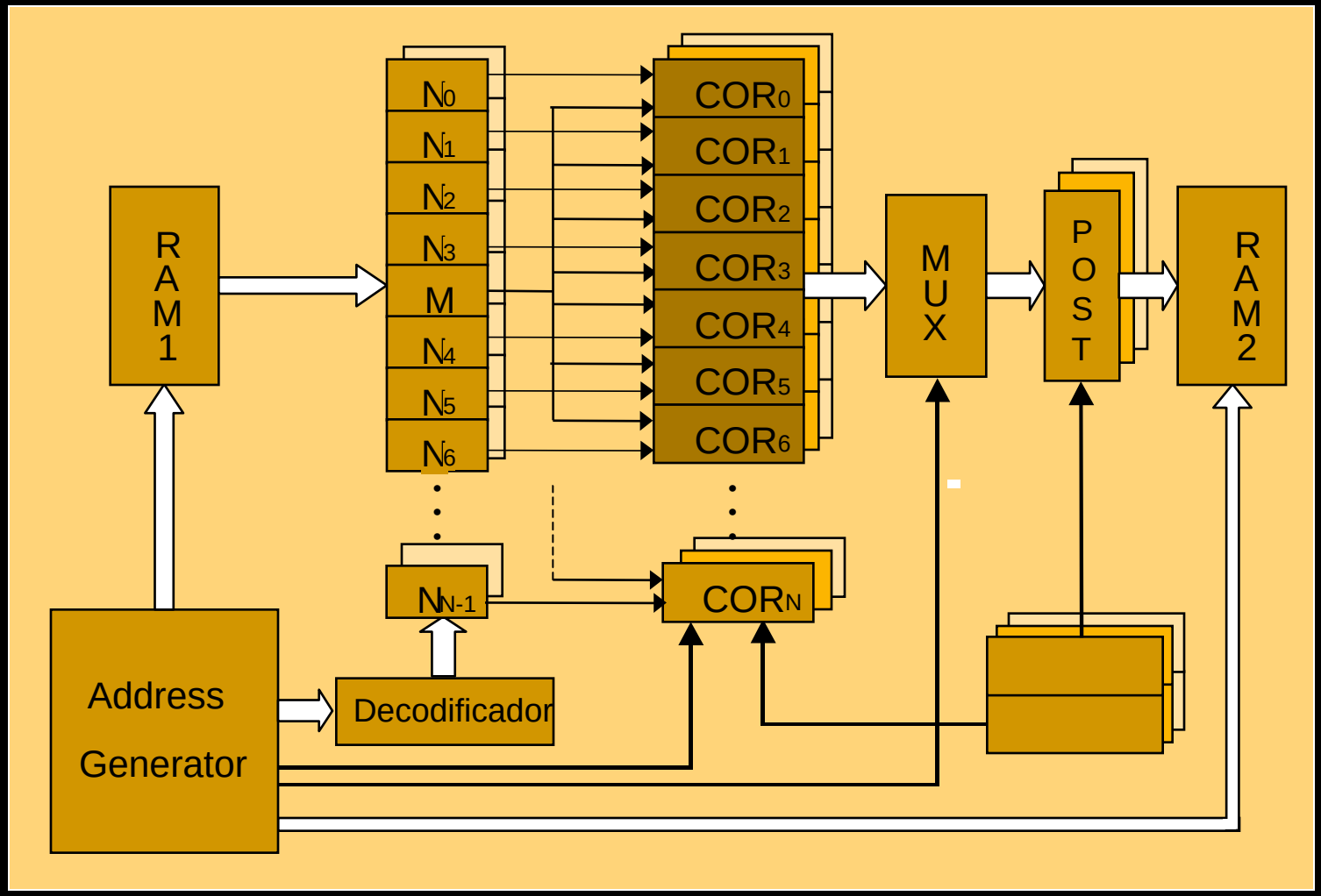


Edge and corner detection

- Industrial applications
- Basis for other image processing applications

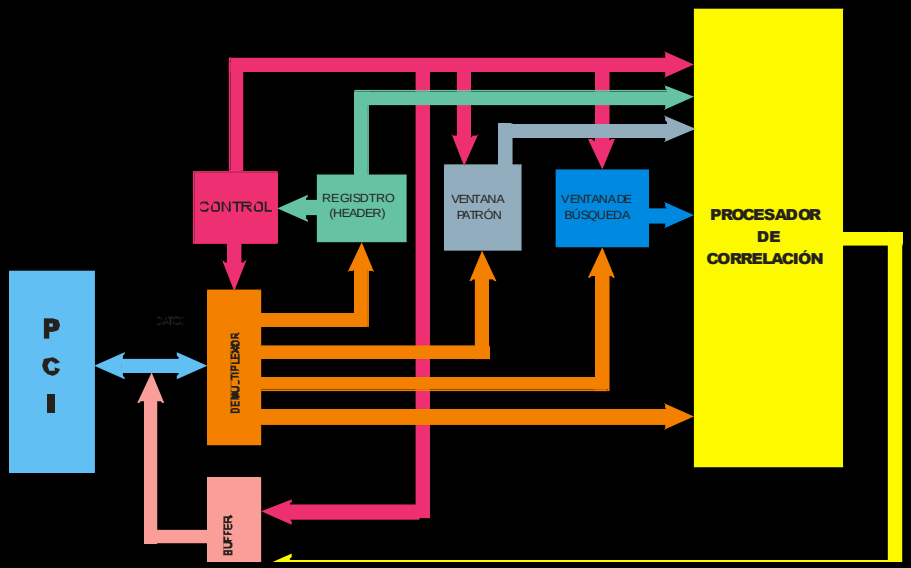


Edge and corner detection architecture

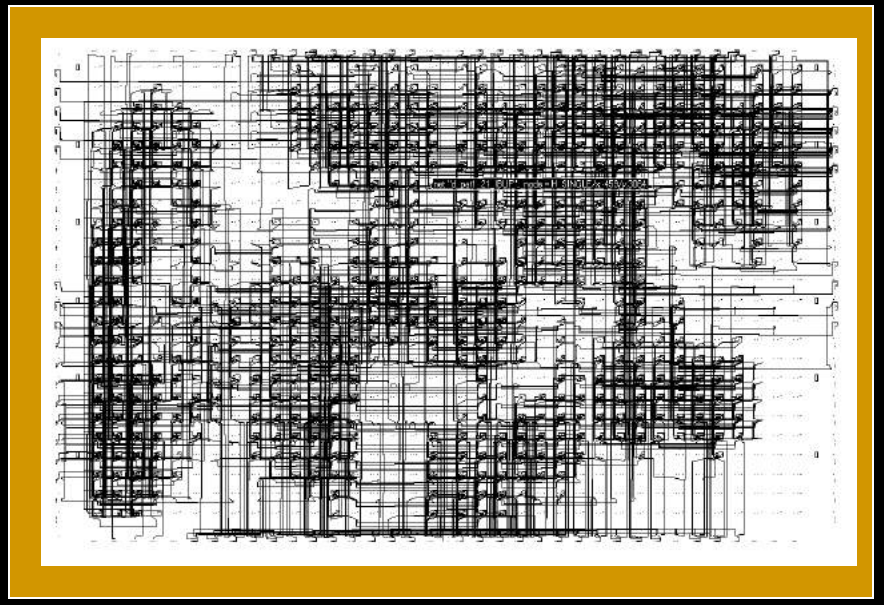


Demostration with RC200

Target tracking



Target processor



FPGA implementation

Multiple object tracking

Performance gain

Algorithm acceleration

- 25 x to 50x compared to PC computer

Drawback

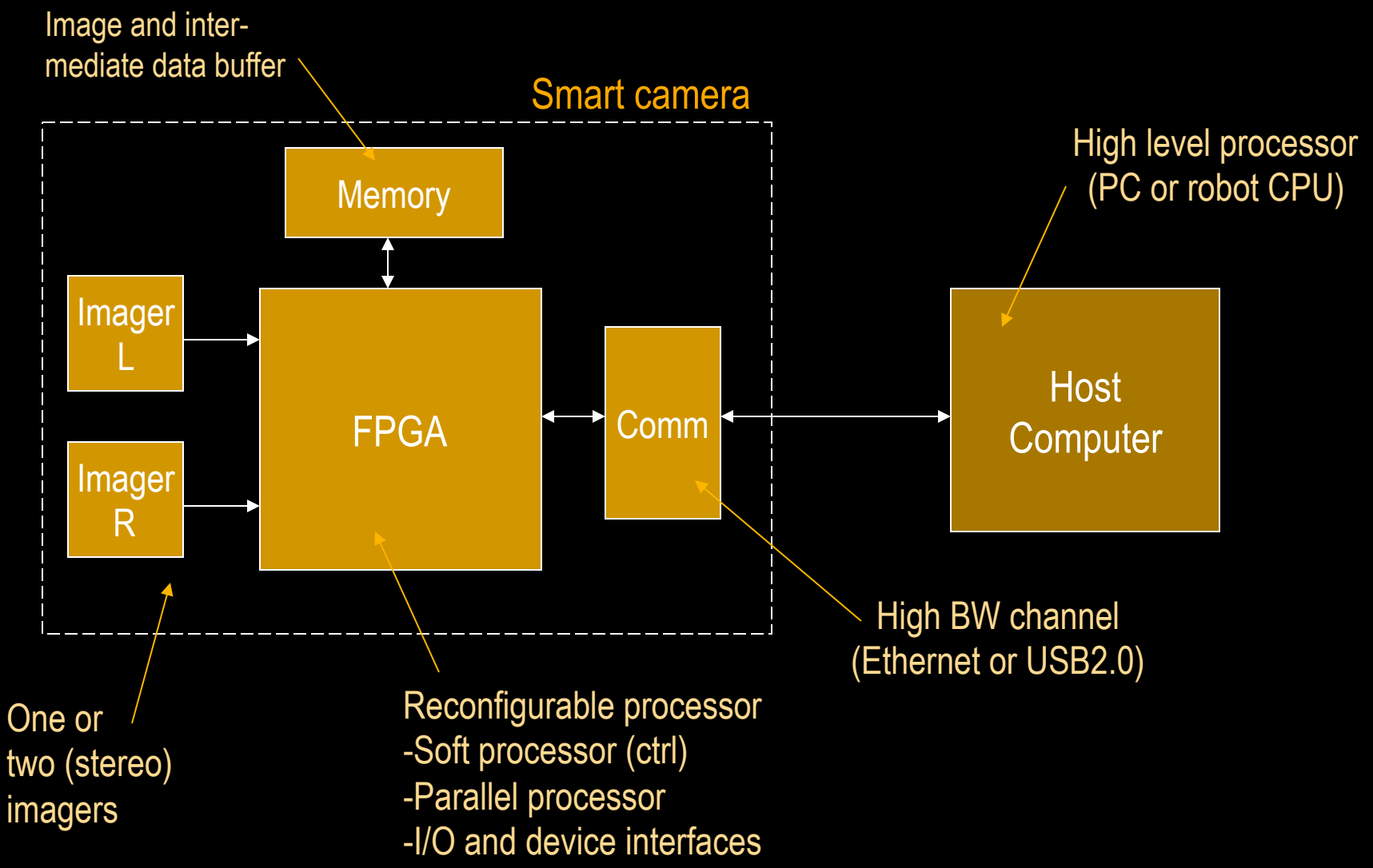
- Modularity and reuse
- Lack of standards for vision cores

3. FPGA based cameras

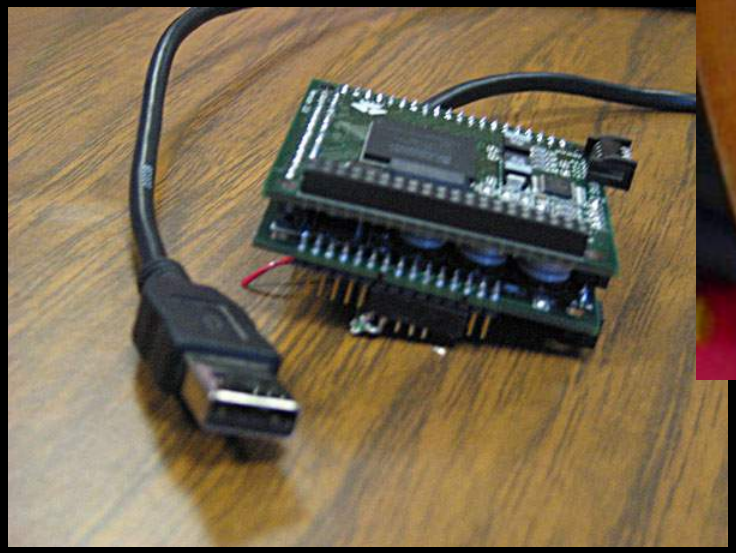
Overview of :

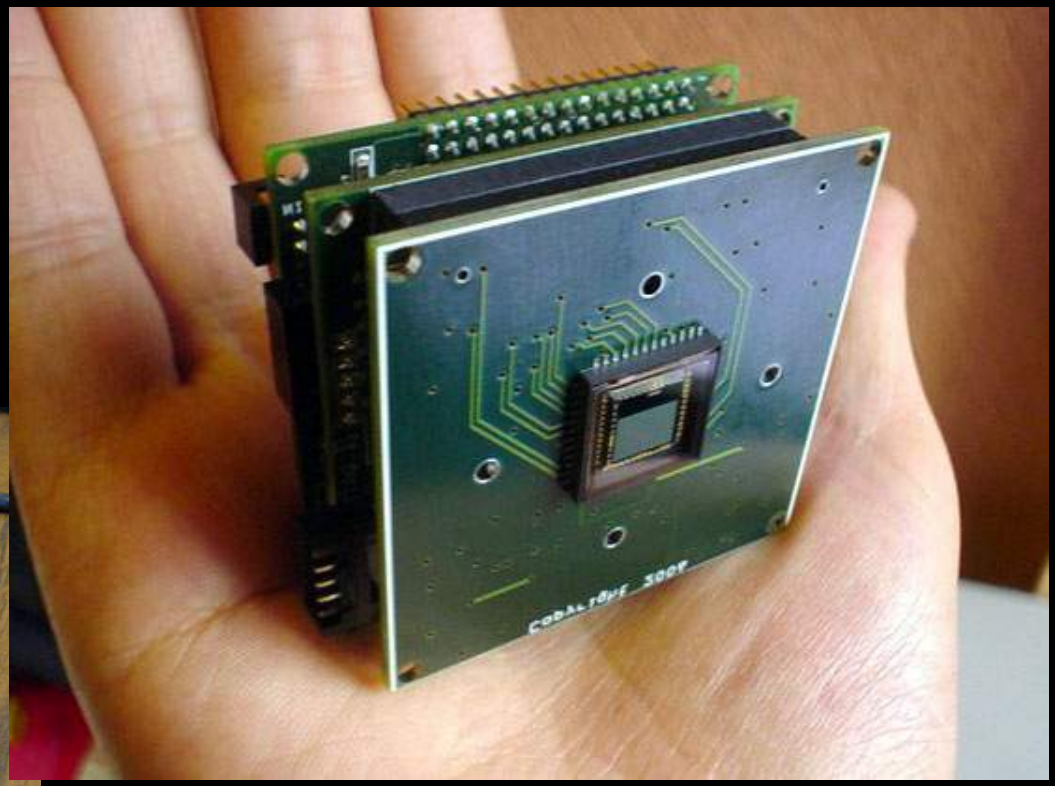
- Early concept
- Current approaches

Smart camera architecture

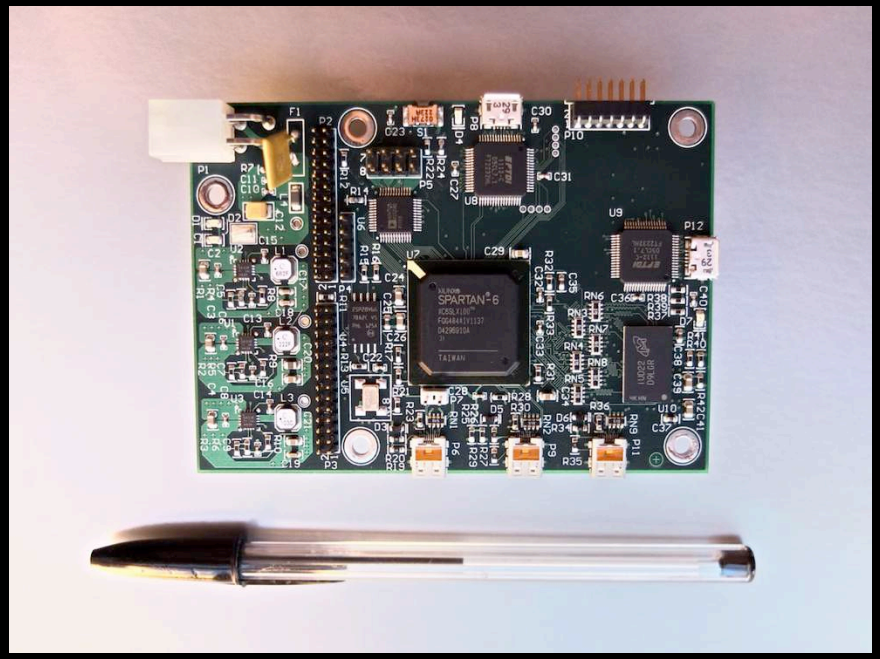


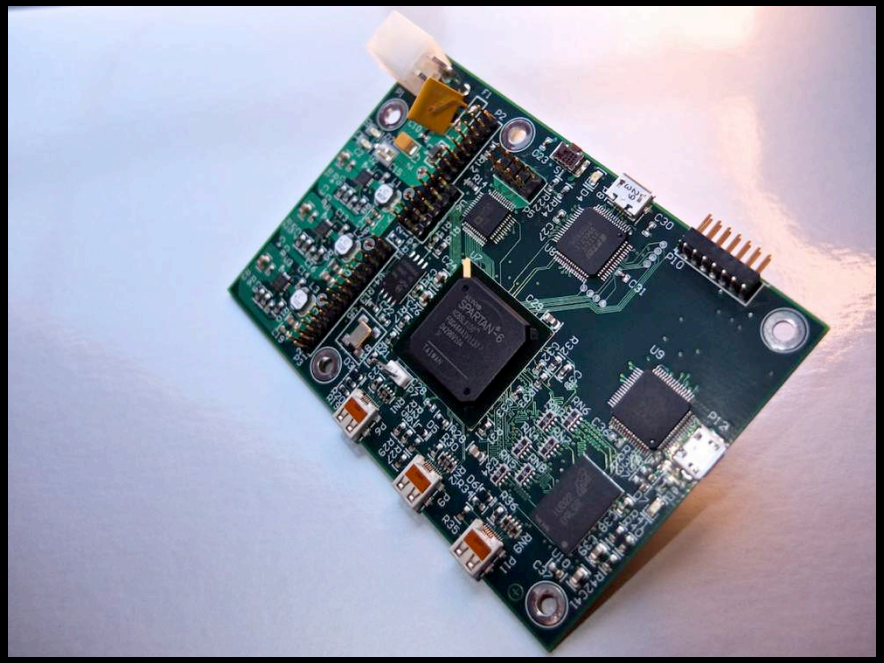
FPGA camera





Custom Spartan6 development board





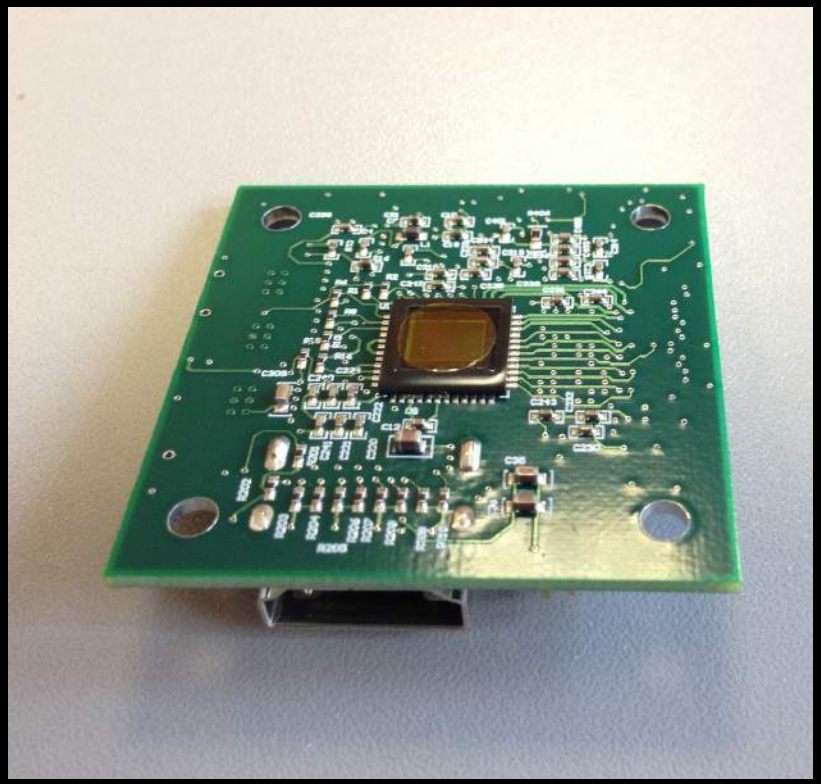
FPGA camera – 2012/2013

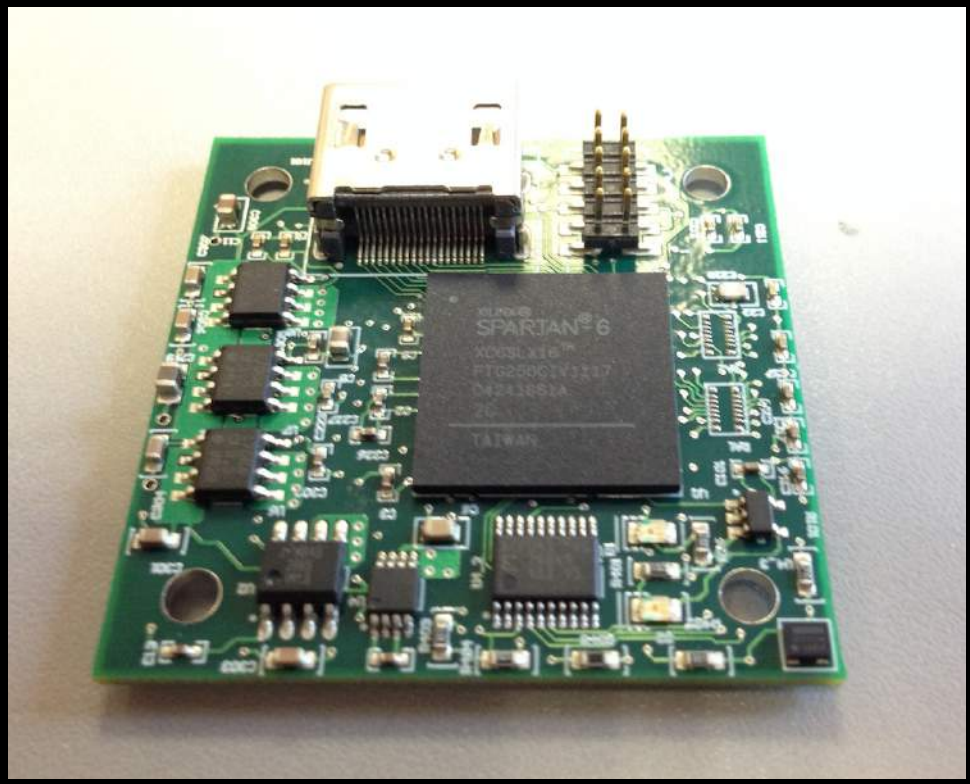
USB 3.0
5 Megapixeles
Spartan 6 device

FPGA for sensor control and data packaging

FPGA room for additional processing

FPGA camera prototype





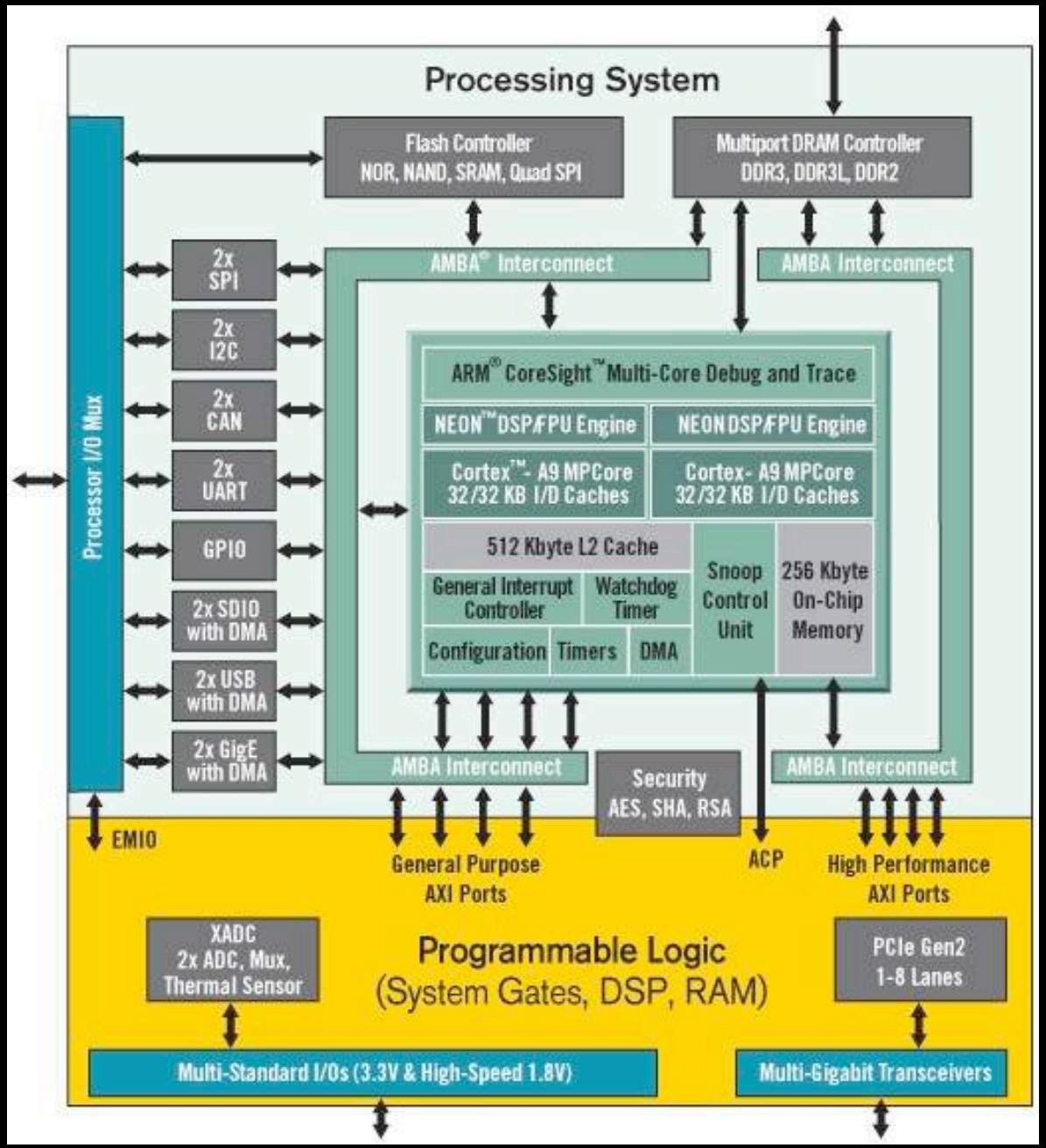
4. Platform proposal

Current work

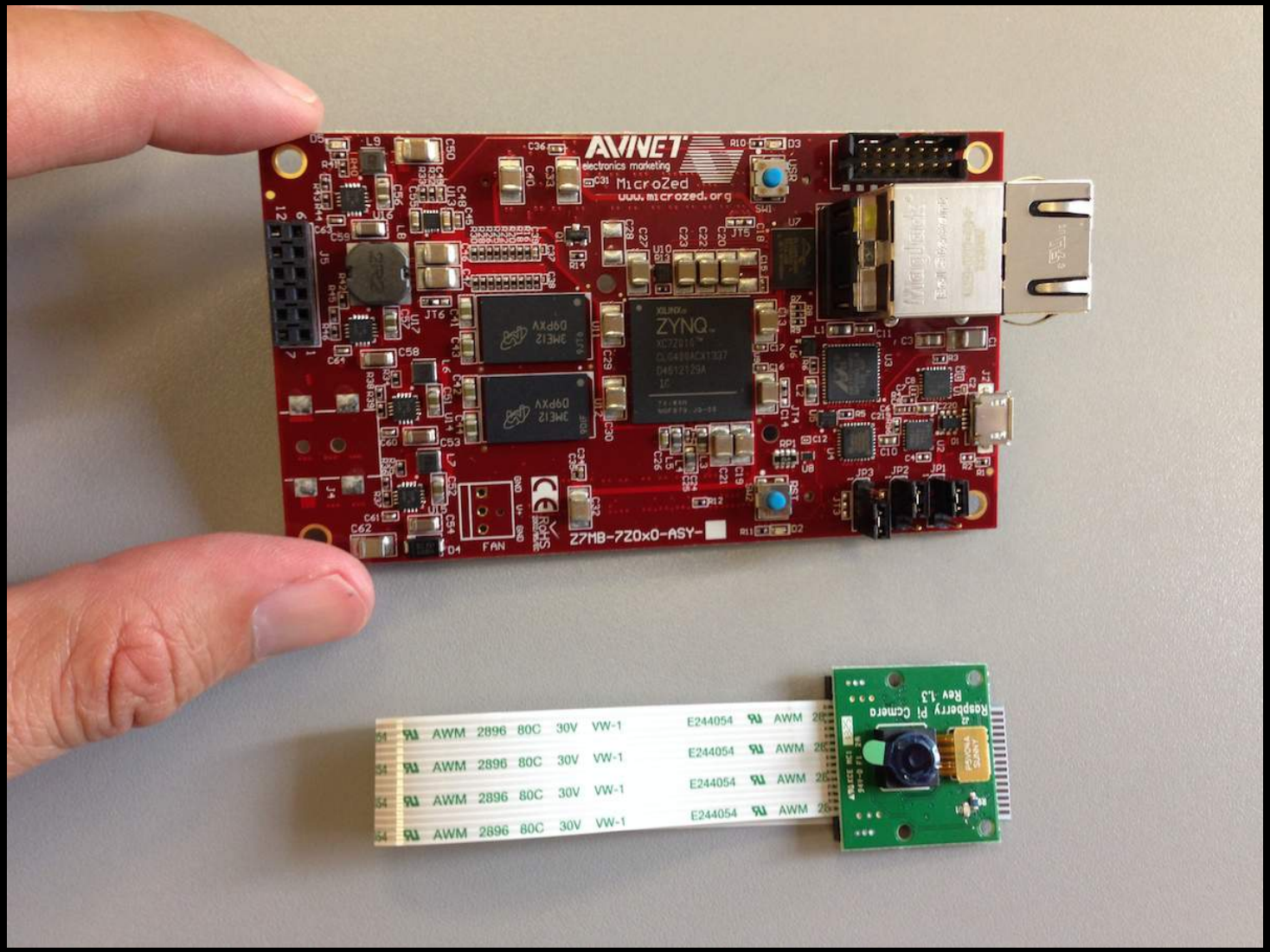
- FPGA/Arm platform + Camera
- Tegra K1 platform + Camera

4.1 FPGA based Proposal

- Use of a SoC (System on a Chip)
- FPGA + ARM processor + Embedded Linux
- Xilinx Zynq7000 + support electronics
- Reconfiguration + I/O flexibility

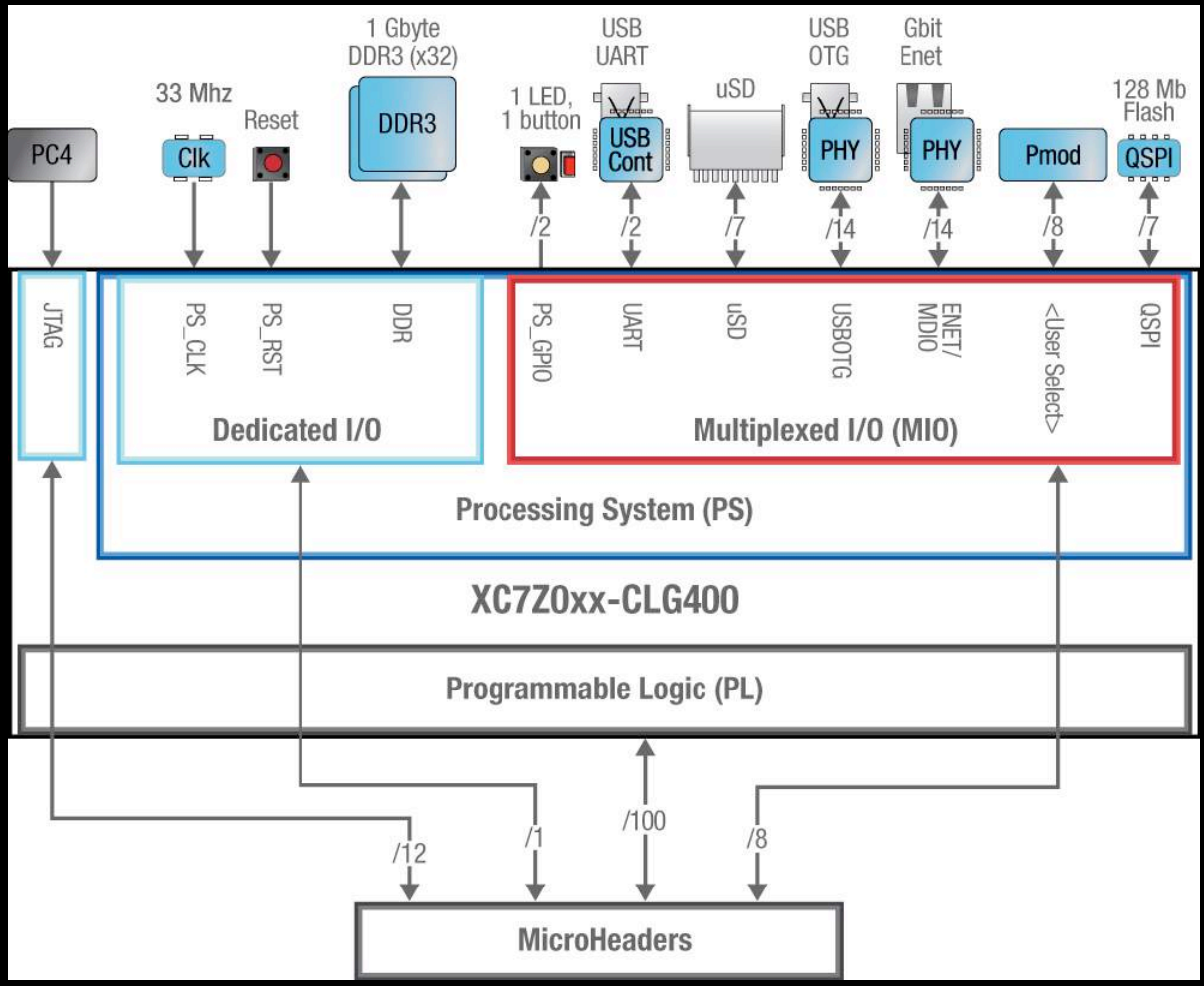

Processing System
Flash Controller
NOR, NAND, SRAM, Quad SPI
Multiport DRAM Controller
DDR3, DDR3L, DDR2
Processor I/O Mux
2x SPI
2x I2C
2x CAN
2x UART
GPIO
2x SDIO with DMA
2x USB with DMA
2x GigE with DMA
AMBA® Interconnect
AMBA Interconnect
ARM® CoreSight™ Multi-Core Debug and Trace
NEON™ DSP/FPU Engine
NEON DSP/FPU Engine
Cortex™- A9 MPCore 32/32 KB I/D Caches
Cortex- A9 MPCore 32/32 KB I/D Caches
512 Kbyte L2 Cache
General Interrupt Controller
Watchdog Timer
Configuration
Timers
DMA
Snoop Control Unit
256 Kbyte On-Chip Memory
AMBA Interconnect
AMBA Interconnect
Security AES, SHA, RSA
EMIO
General Purpose AXI Ports
ACP
High Performance AXI Ports
XADC 2x ADC, Mux, Thermal Sensor
Programmable Logic (System Gates, DSP, RAM)
PCIe Gen2 1-8 Lanes
Multi-Standard I/Os (3.3V & High-Speed 1.8V)
Multi-Gigabit Transceivers

FPGA platform

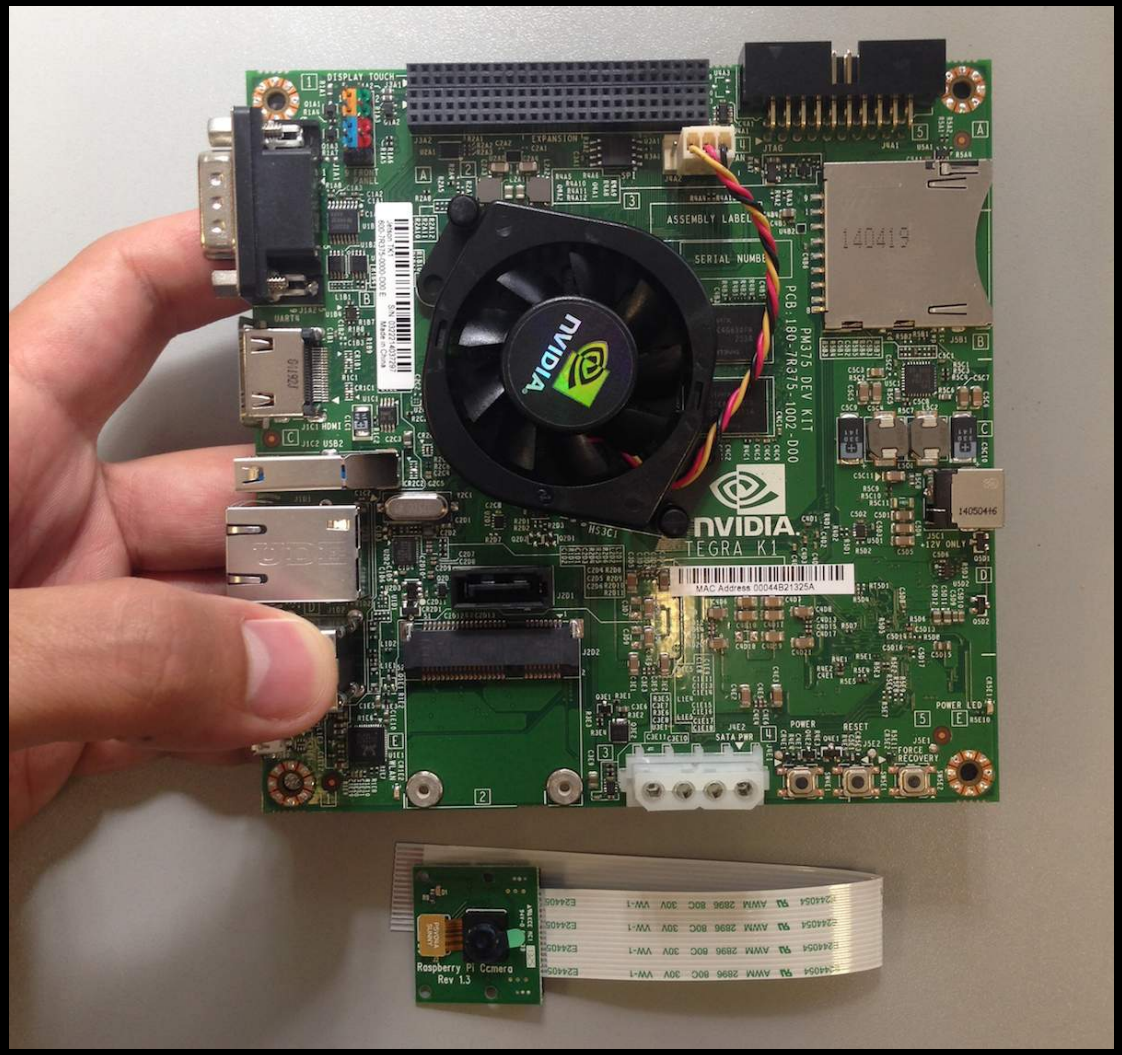


FPGA Platform :: MicroZed

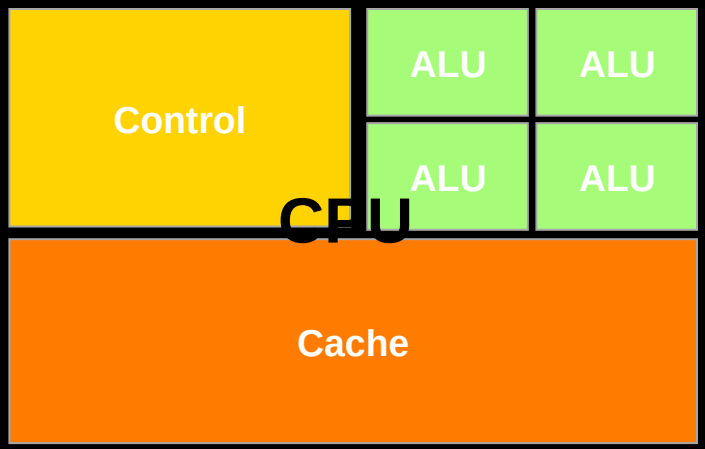
- Xilinx XC7Z010
- USB 2.0
- Gbit Ethernet
- 1 Gbyte SRAM DDR3
- 128 Mb Flash
- Micro SD card
- 100 I/O
- Embedded Linux

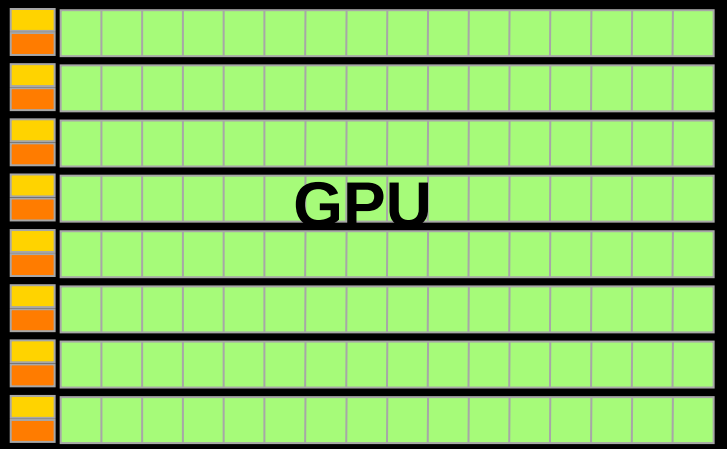


4.2 CUDA platform



CPU vs GPU





CUDA programming

Optimización de aplicaciones complejas

FPGA vs Embedded CUDA

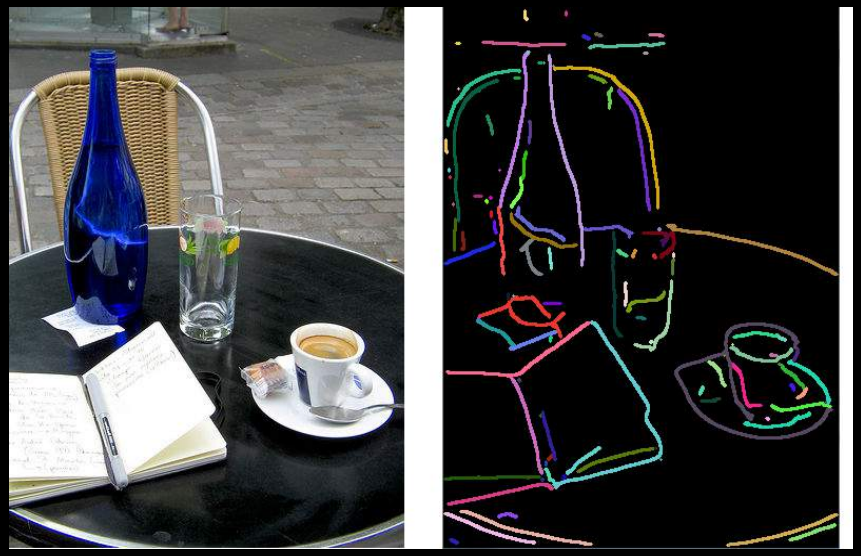
	FPGA	CUDA
Advantages	• Low power • High performance • Small size, possible to migrate to VLSI	• Easy to program • Speed up • Floating point
Inconvenient	• Complex to implement • Long to learn • Architecture complexity vs speedup	• Reformulate in parallel: core + memory use • Power consumption

5. Long term project

- Image + Feature extraction in Camera
- Form descriptor extraction at the camera level (best for CUDA programing)
- Host computer or Cloud for high level cognitive modeling / BigData techniques
- Network of cameras can open new research possibilities

PARTIAL CORRESPONDENCE OF FORM

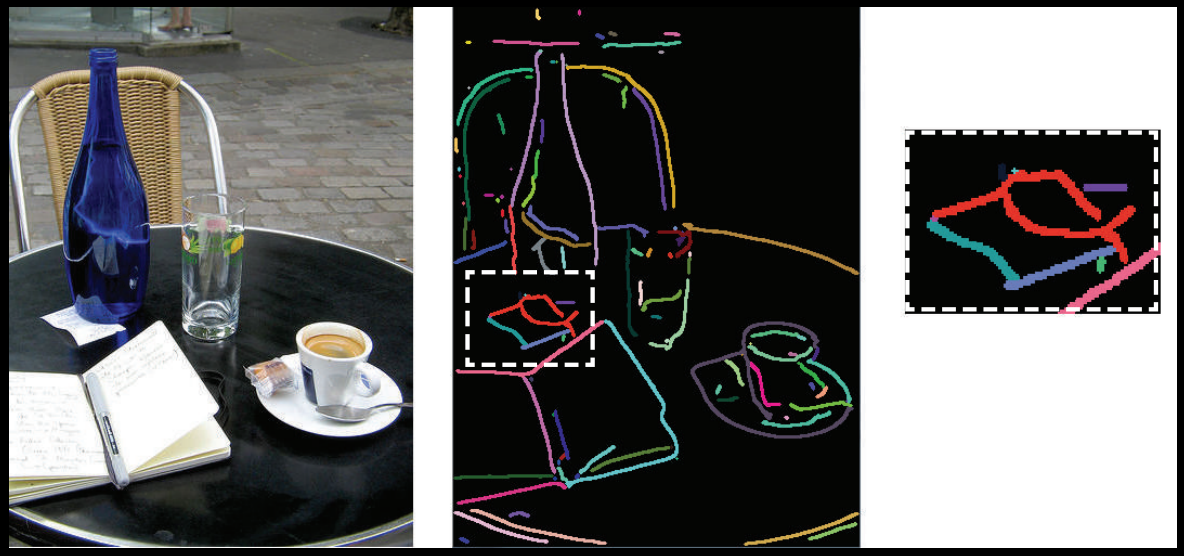
- Object recognition using form
  - From a given object model, select a subset of corresponding edge segments.



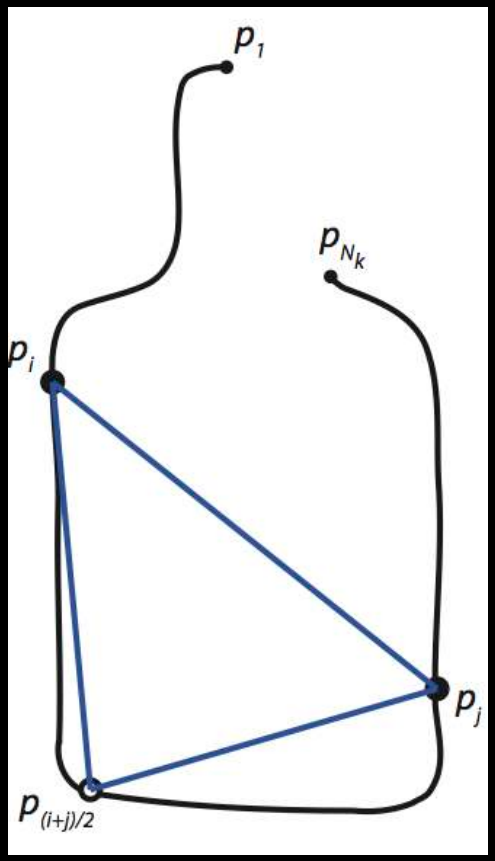
PARTIAL CORRESPONDENCE OF FORM

CHALLENGE

Part of the contour can be connected incorrectly with the background or other object, giving a wrong edge to be matched



OCTAR FORM DESCRIPTOR



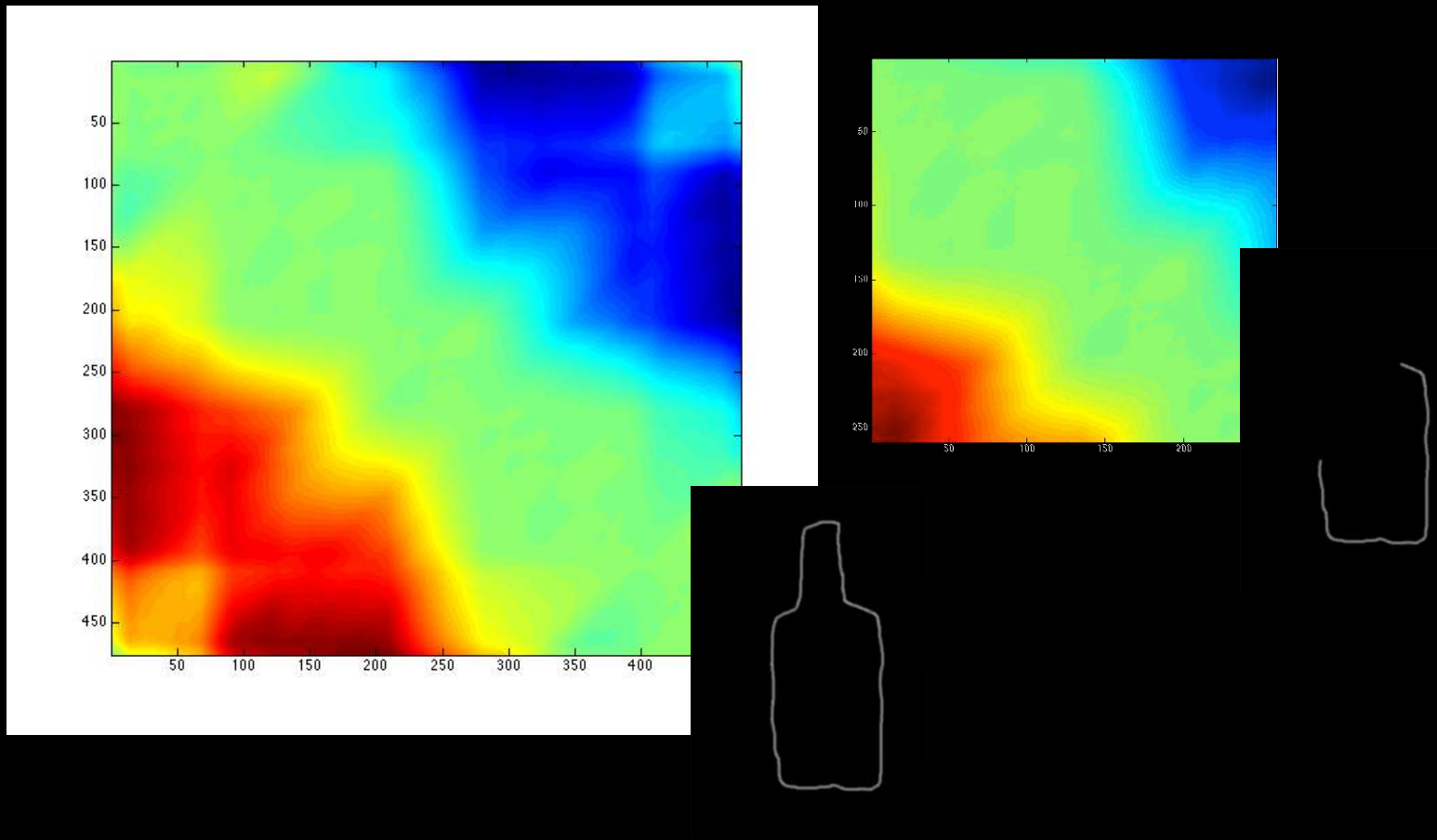
$$OCTAR(i,j) = \frac{1}{2}\begin{vmatrix} x_i & y_i & 1 \\ x_{(i+j)/2} & y_{(i+j)/2} & 1 \\ x_j & y_j & 1 \end{vmatrix}$$

$$\Phi(A,B) = 1 - \frac{1}{N^2}\sum_{i}^{N}\sum_{j}^{N}\left(\frac{|OCTAR_A(i,j) - OCTAR_B(i,j)|}{2\,\text{máx}(|OCTAR_A(i,j)|, |OCTAR_B(i,j)|)}\right)$$

- Open contour
- Self-contained
- Rotation and translation invariant

PARTIAL CORRESPONDENCE OF FORM

FORM DESCRIPTOR



PARTIAL CORRESPONDENCE OF FORM

OBJECT LOCATION

- Each partial correspondence vote for the center of the object

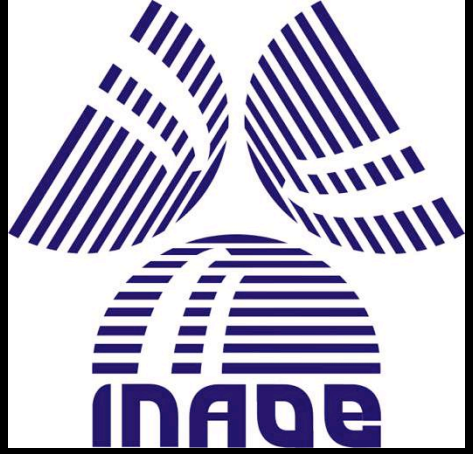
PARTIAL CORRESPONDENCE OF FORM

OBJECT LOCATION

- Fragments that can be part of the object

6. Conclusions

- FPGA based processing for low level feature extraction
- Form descriptors and medium level processing is better with CUDA based platform
- Potential to combine networks of cameras with embedded vision processing and Cloud computing



Laboratorio de Cómputo Reconfigurable y de Alto Desempeño

Miguel Arias – Computer Sc. Dept.
ariasmo@inaoep.mx