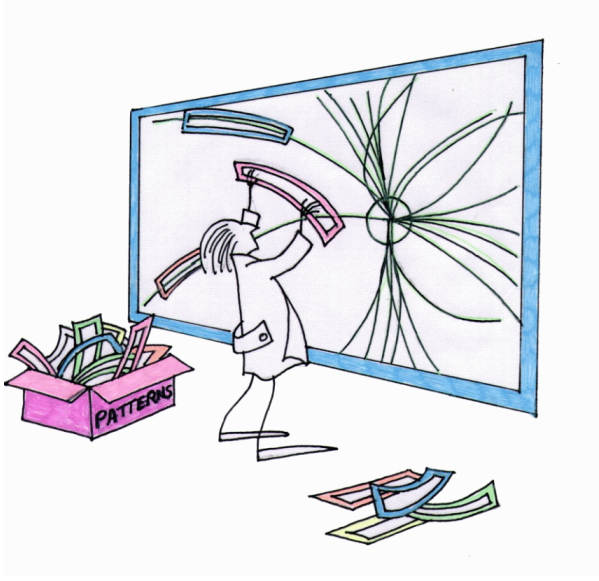
MARIE CURIE IAPP: FAST TRACKER FOR HADRON COLLIDER EXPERIMENTS

1ST FTK IAPP WORKSHOP

PISA, JULY 2013

Programmable Logic (FPGAs) for Computationally Intensive Applications

Calliope-Louisa Sotiropoulou
PhD Candidate/Researcher
Aristotle University of Thessaloniki



AUTH e-LAB
Aristotle University of Thessaloniki-Electronics Laboratory

Programmable Logic (FPGAs) for Computationally Intensive Applications

- Embedded Systems
- FPGAs
- VHDL
- FPGA MPSoCs
- Canny Edge Detection
- Machine Vision Implementation

Programmable Logic (FPGAs) for Computationally Intensive Applications

- Embedded Systems
- FPGAs
- VHDL
- FPGA MPSoCs
- Canny Edge Detection
- Machine Vision Implementation

Introduction – Embedded Systems

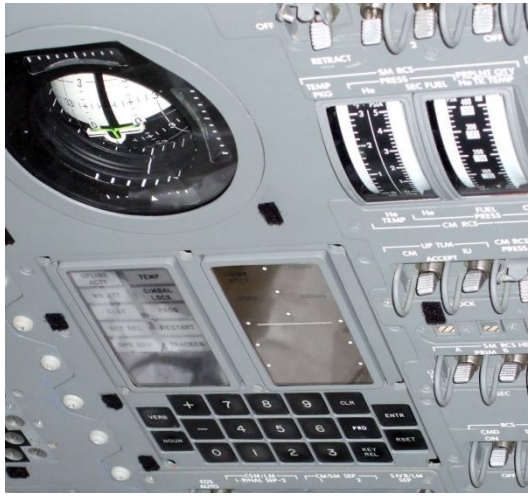
- An **embedded system** is a computer system designed to do one or a few dedicated and/or specific functions

- Often with **real-time** computing constraints
  (something which needs to be done immediately)

- It is **embedded** as part of a complete device often including **hardware** and **mechanical parts**

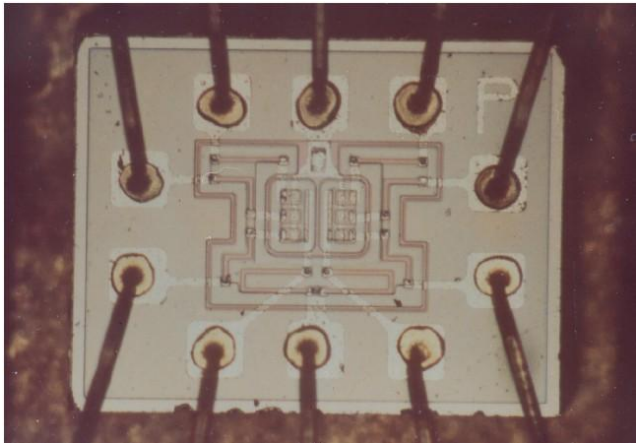
Introduction – Embedded Systems

- One of the first modern embedded systems was the Apollo Guidance Computer (MIT Instrumentation Laboratory)



AGC User Interface

AGC dual nor gate



Introduction – Embedded Systems

- What about today?

Embedded Systems are everywhere!

Embedded Systems application domains

- High-throughput Multimedia
- Security
- Smart devices
- Gaming Platform

- Such applications have [illegible] or
  - Performance
  - Flexibility
  - Memory Space

  Combined with area and power consumption constraints

Embedded Systems

- Where can we implement embedded systems?

Programmable Logic (FPGAs) for Computationally Intensive Applications

- Embedded Systems
- FPGAs
- VHDL
- FPGA MPSoCs
- Canny Edge Detection
- Machine Vision Implementation

Introduction – FPGA Technology

- **Field Programmable Gate Arrays (FPGAs)** are programmable semiconductor devices that are based around a matrix of configurable logic blocks (CLBs) connected via programmable interconnects.

- As opposed to **Application Specific Integrated Circuits (ASICs)** where the device is custom built for the particular design, FPGAs can be programmed to the desired application or functionality requirements.

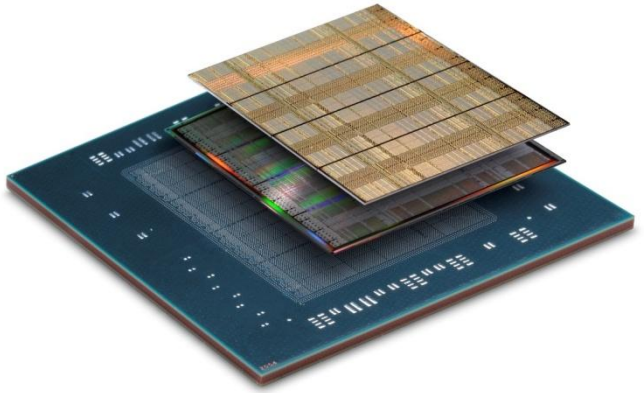
In simple words…

- FPGAs:
  - Can be reprogrammed many times
  - Can be reused for different applications

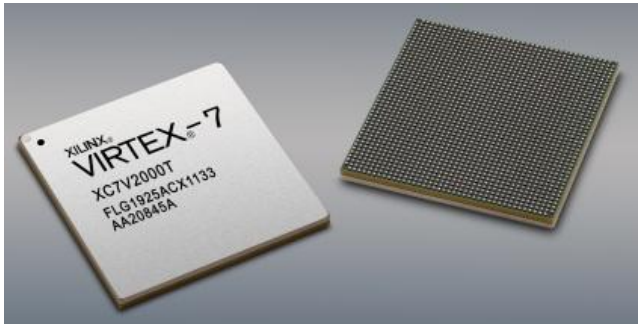
- ASICs:
  - Are printed circuits
  - Used for very specific applications
  - Designed once for one task

Field Programmable Gate Arrays

- Reconfigurability
- Versatility
- Reduced cost
- …and limited time to market

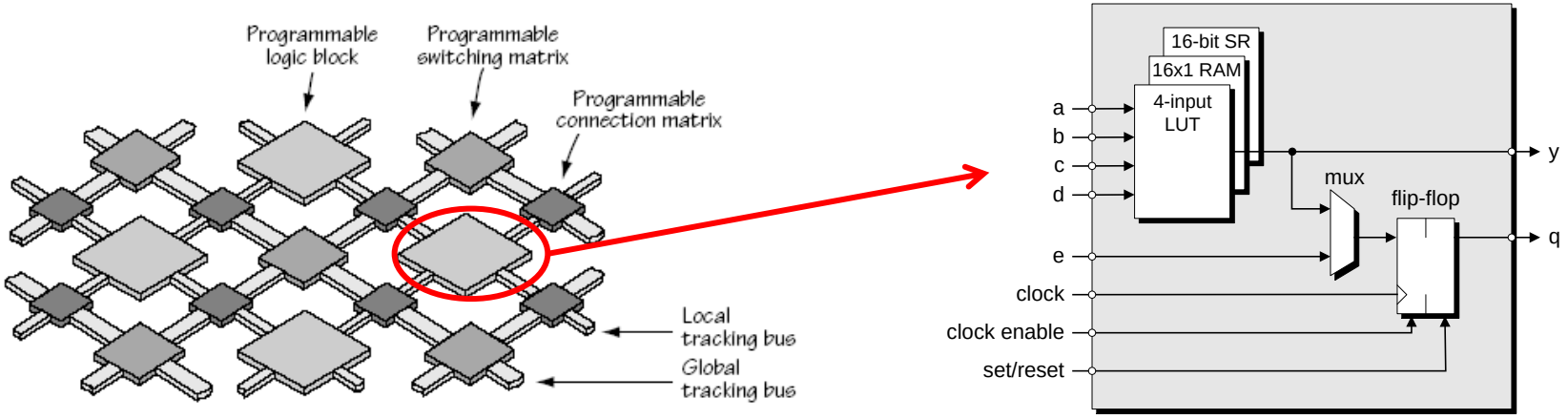


- Modern FPGA devices offer a great number and variety of resources at a reasonable cost

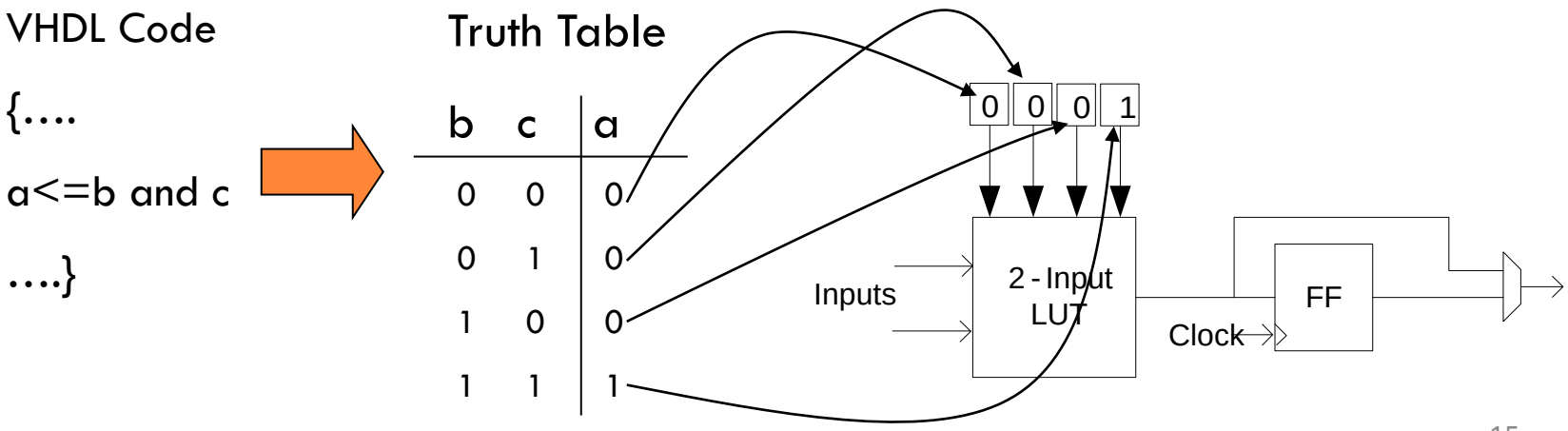


Programmable logic blocks

- The "heart" of every FPGA
  - Logic functions implemented in look-up tables (LUTs)
  - Clocked storage elements (flip-flops)
  - N-to-1 Multiplexers



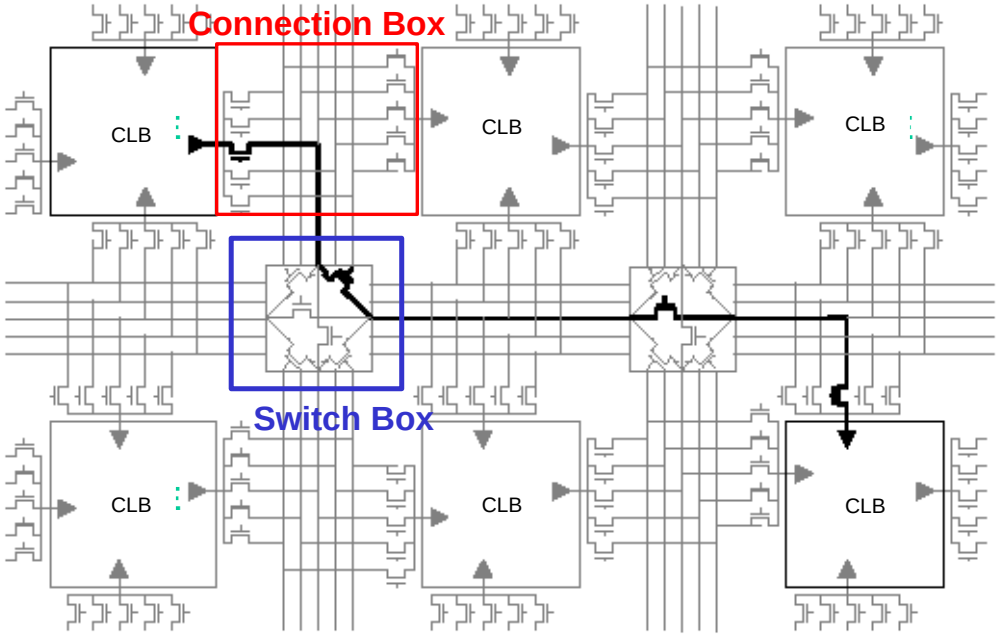
Logic Implementation on an FPGA



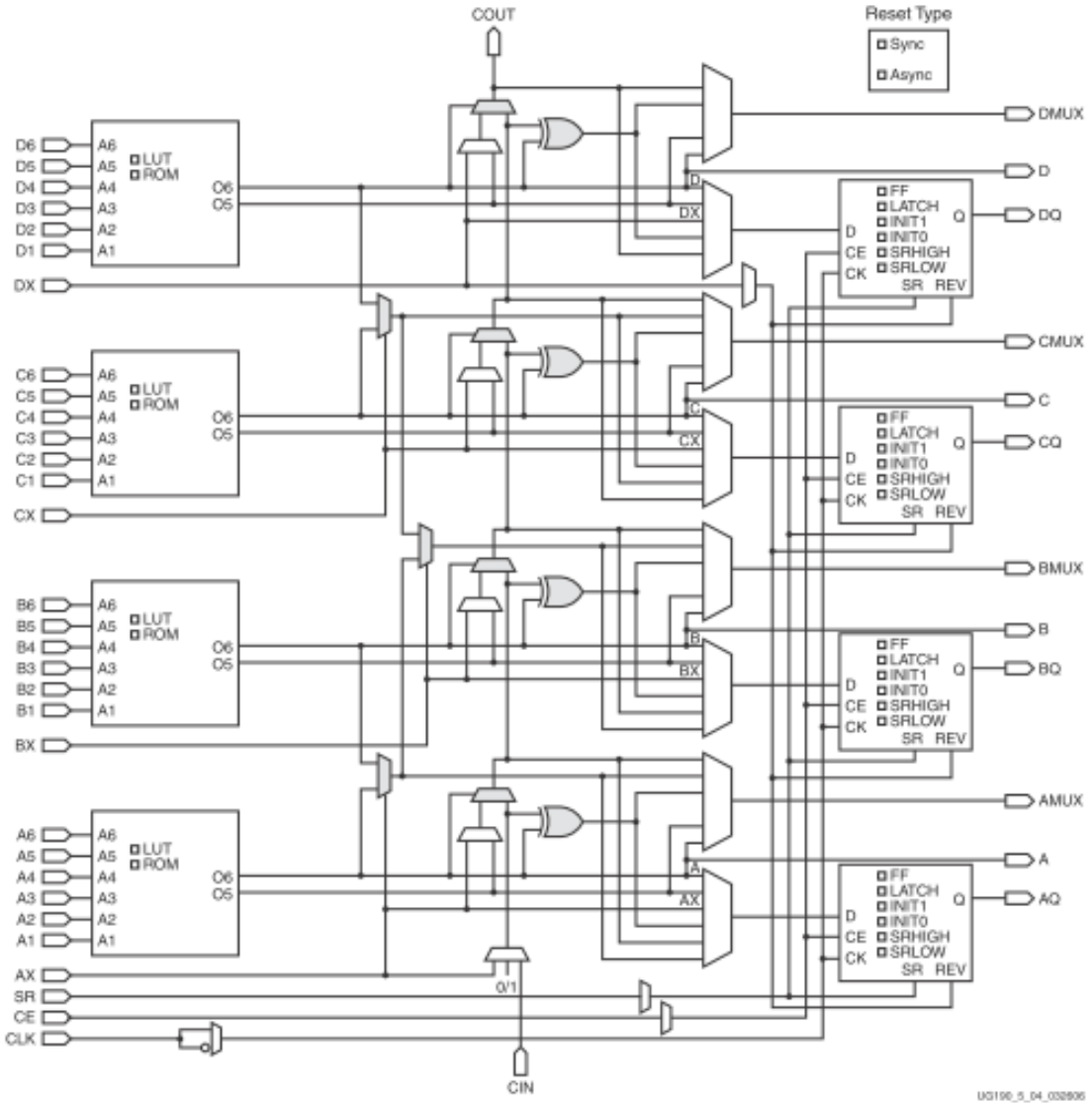
- The truth table of a simple operator is loaded to a LUT
- The two combined inputs are used as address
- The output is stored on a flip-flop for synchronization

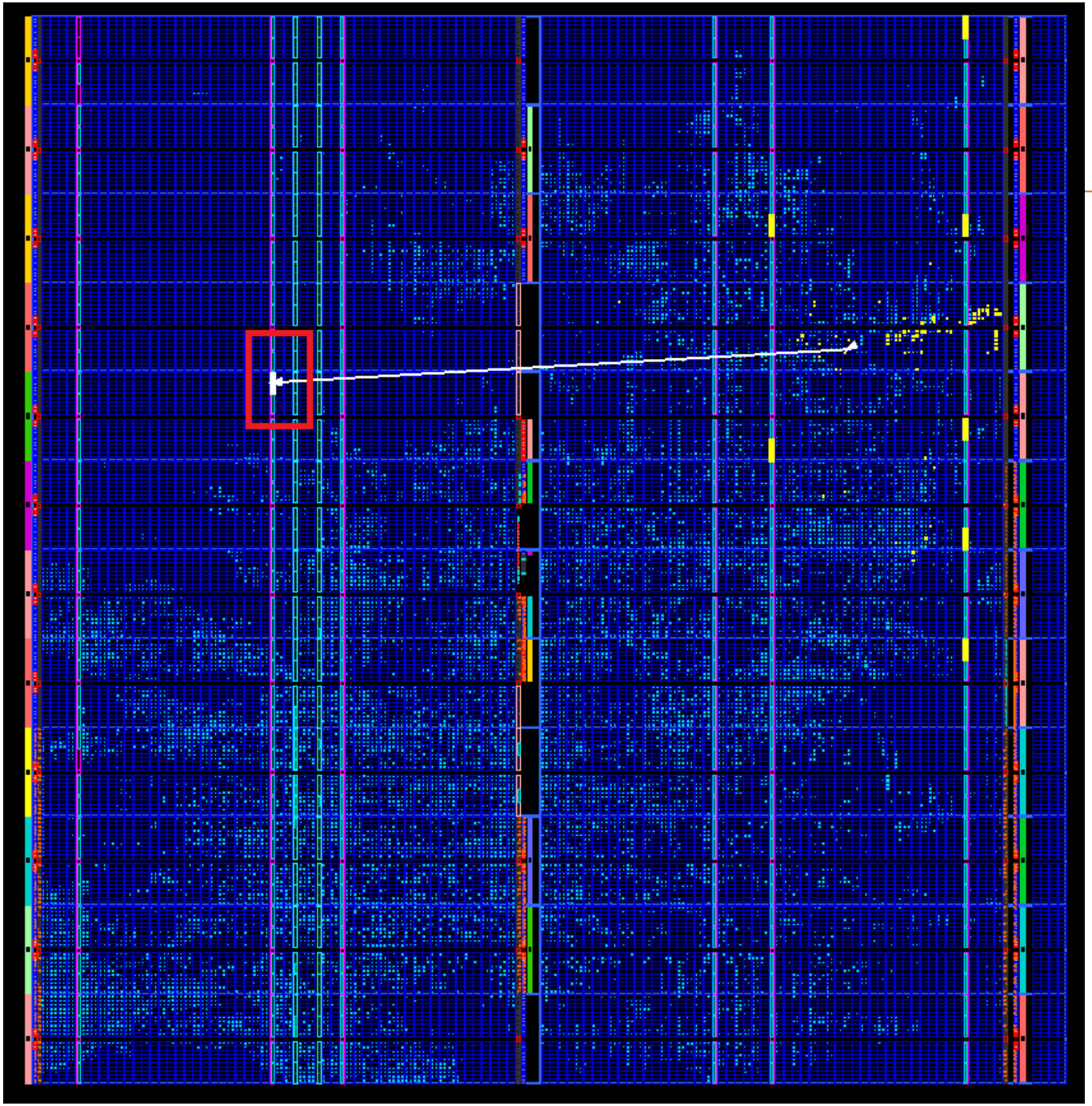
Routing on an FPGA

- Programmable connection and switch boxes



Xilinx Virtex-5, Slice (SLICEM)





FPGA Vendors

- Xilinx

- Altera

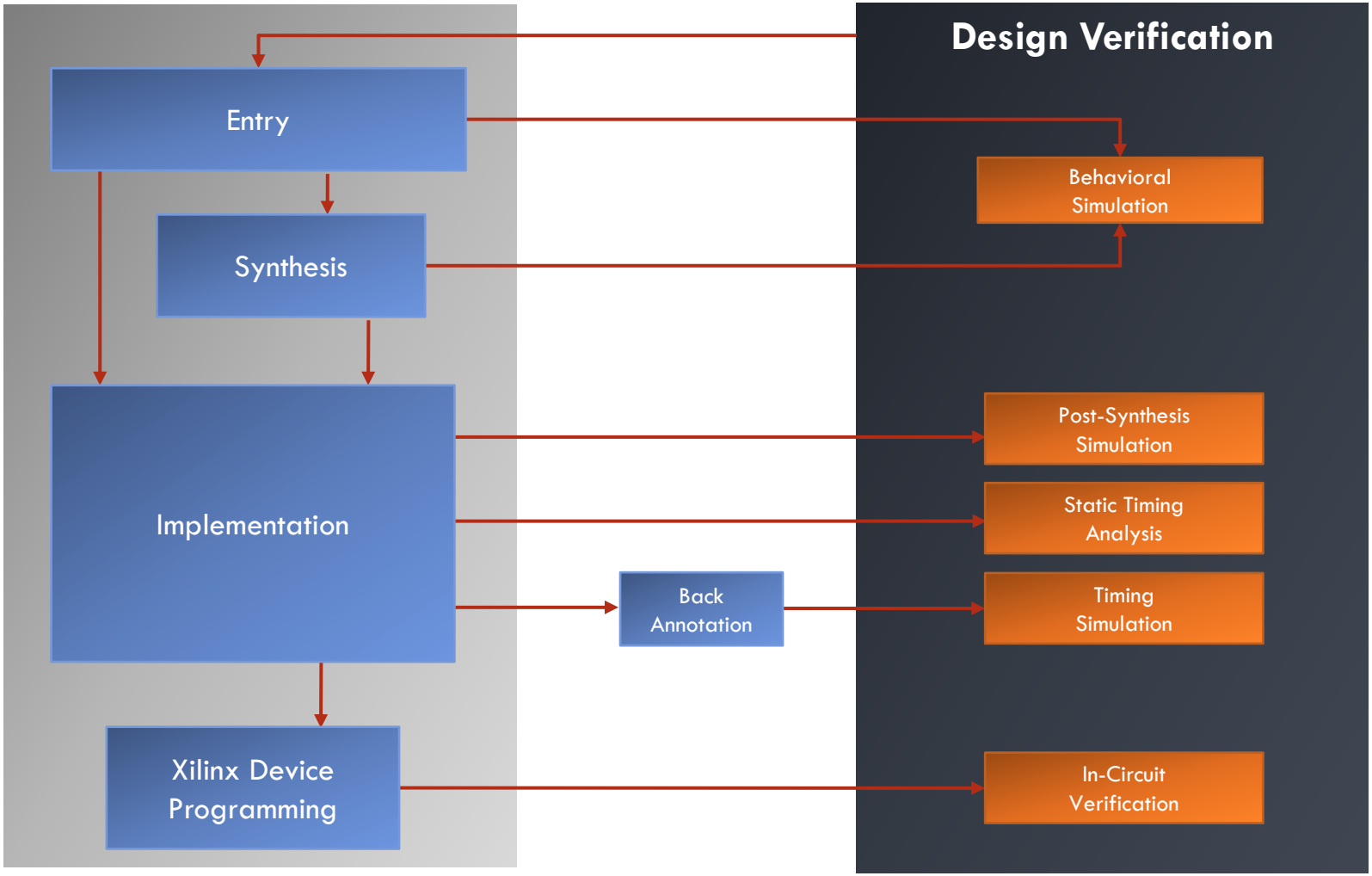
- Lattice Semiconductor

- Actel

FPGA Tools

- All FPGA Vendors offer tools for FPGA implementation
- There is almost always a version you can download for free
- You just need a pc to use it!
- (And an FPGA to implement it… but not necessarily)

Xilinx ISE Design Flow



Programmable Logic (FPGAs) for Computationally Intensive Applications

- Embedded Systems
- FPGAs
- VHDL
- FPGA MPSoCs
- Canny Edge Detection
- Machine Vision Implementation

Introduction – VHDL

- What is VHDL?

A **V**ery **H**ard and **D**ifficult **L**anguage…

Introduction – VHDL

- **V**ery High Speed Integrated Circuit –
  Hardware **D**escription **L**anguage

- VHDL originated in the early 1980s
  - The American Department of Defense initiated the development of VHDL in the early 1980s
    - because the US military needed a standardized method of describing electronic systems
- VHDL was standardized in 1987 by the IEEE

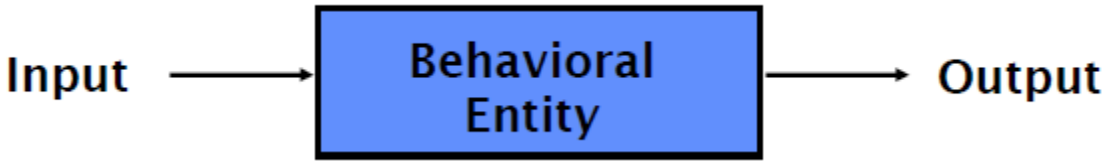
VHDL

- VHDL is a programming language that allows one to model and develop complex digital systems
- Allows you to define in/out ports and specify behavior or response of the system
- With VHDL we can design a system/circuit → Something that physically exists
- Or we can model a system's/circuit's behavior

VHDL - Hierarchy

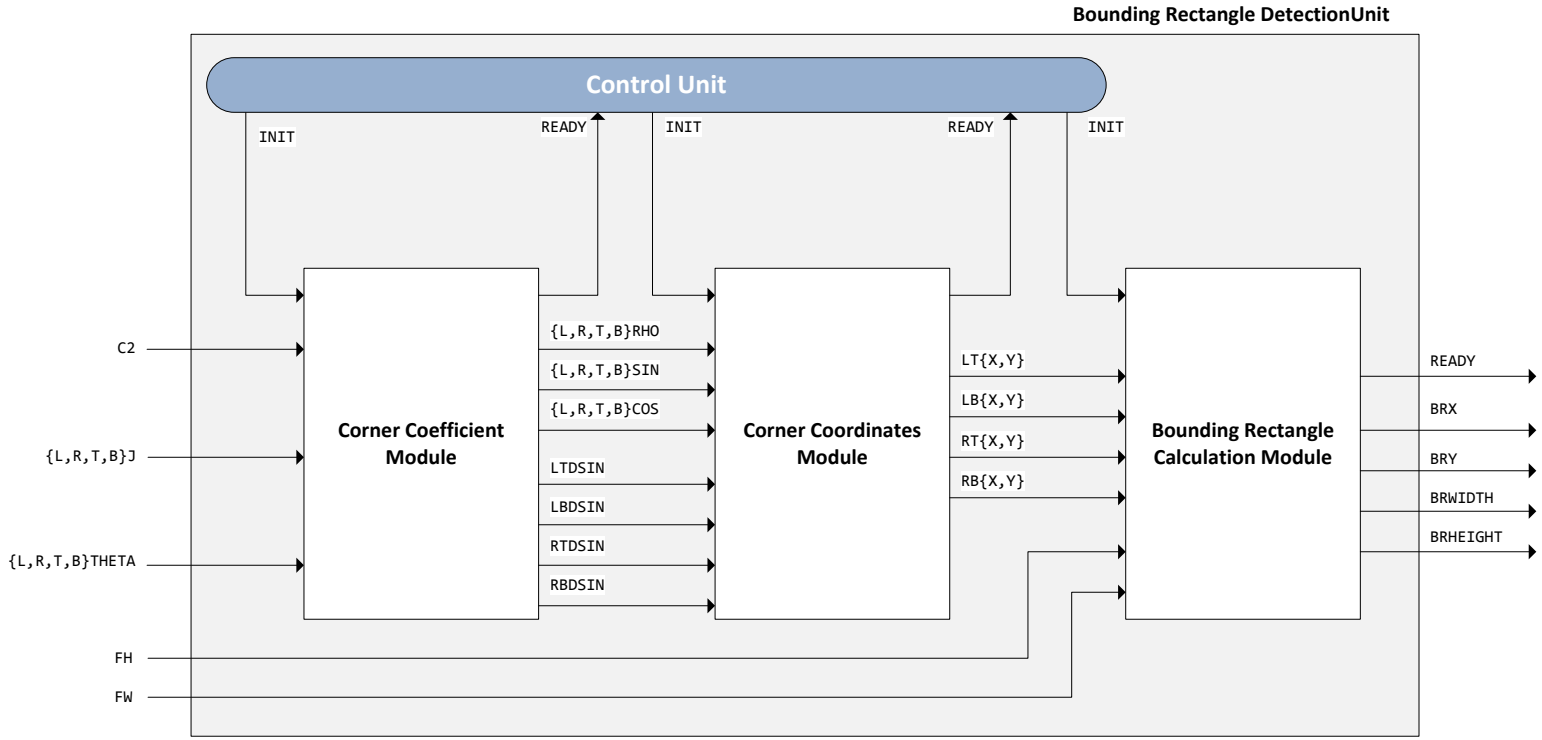
- Black Box Principle:
  At every hierarchy level only the absolutely necessary information is disclosed

- Input/Output Ports and their behavior



VHDL - Hierarchy

- Therefore you can design from simple modules to complete systems with the same tools



Programmable Logic (FPGAs) for Computationally Intensive Applications

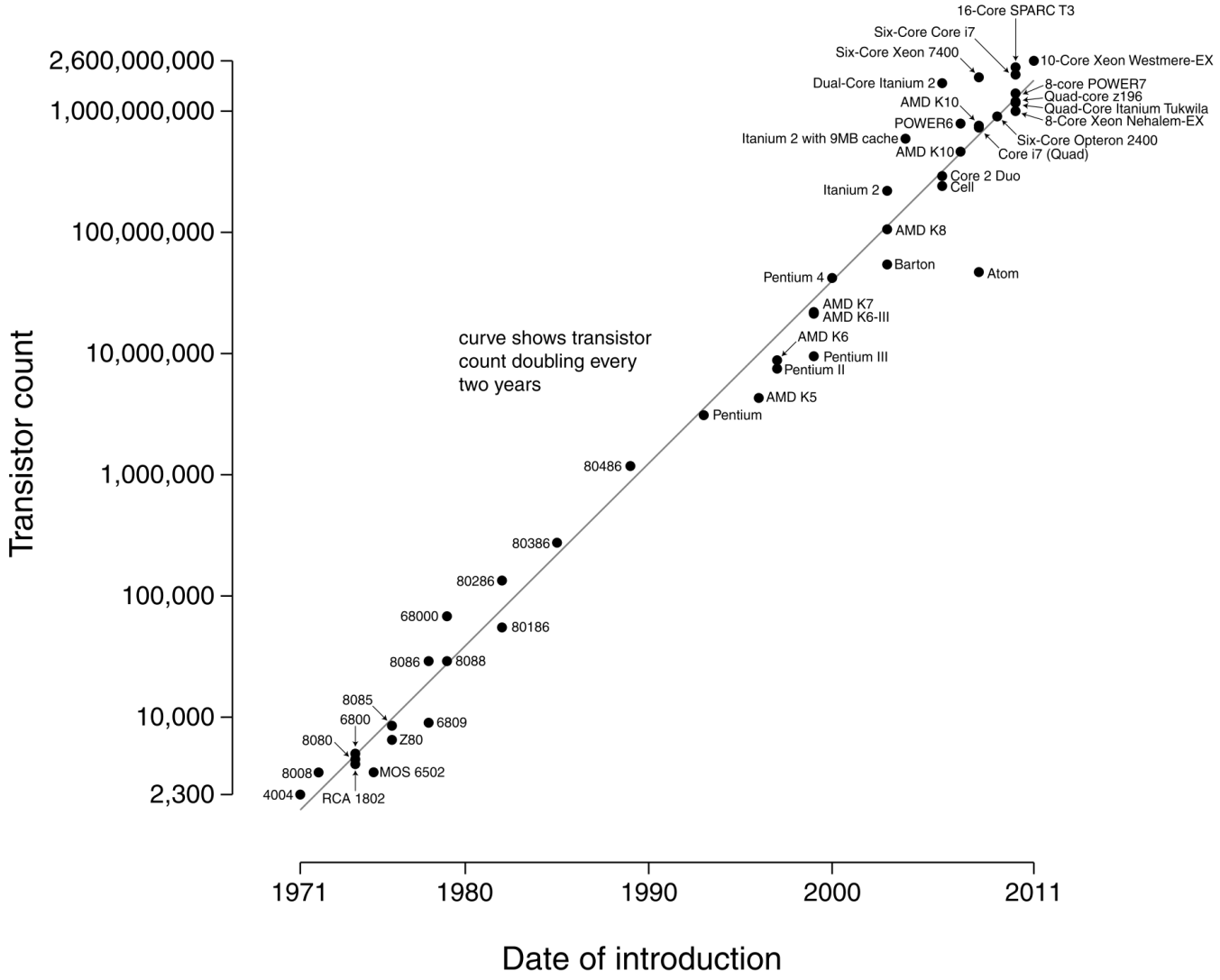
- Embedded Systems
- FPGAs
- VHDL
- FPGA MPSoCs
- Canny Edge Detection
- Machine Vision Implementation

The need for performance…

- Performance requirements can no longer be supported by Embedded System Architectures based on a single processor

- Multiprocessing architectures are being used

Microprocessor Transistor Counts 1971-2011 & Moore's Law



FPGA based MPSoC

On an FPGA device a designer can configure:

- The number of processors
- The types of the interconnection buses
- The size and type of onboard or external memory
- The possible substitution of the execution of a computational task by a processor with dedicated hardware

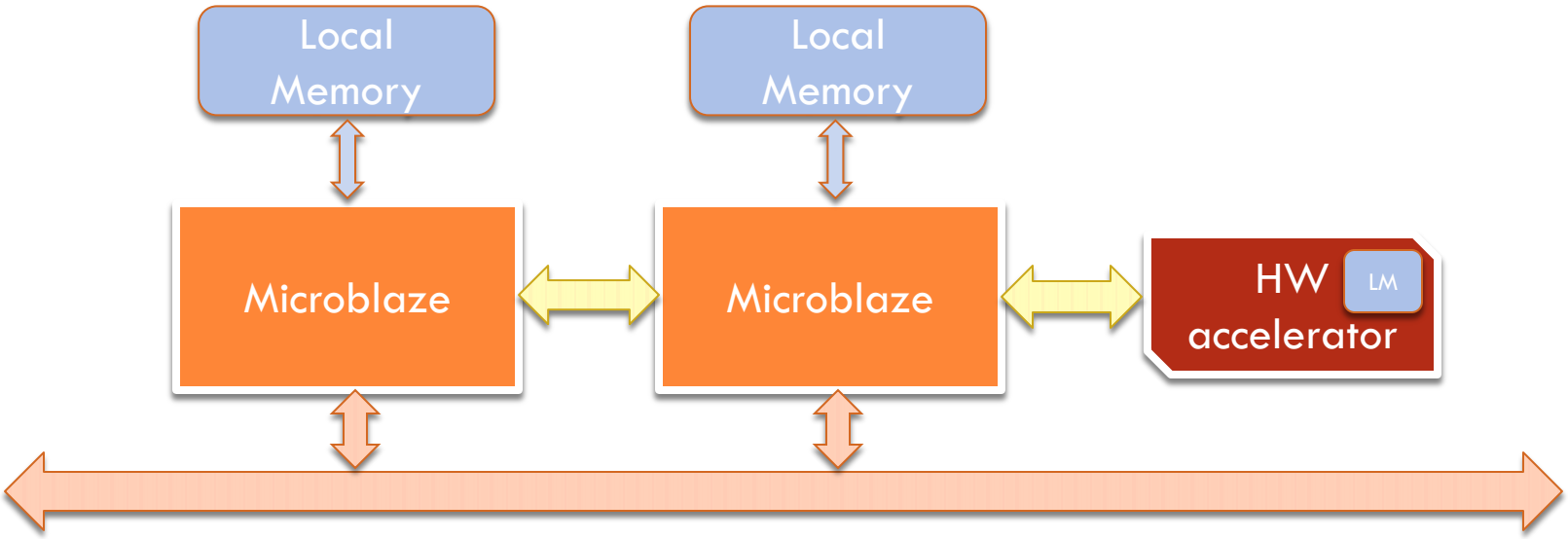
- Effort and time needed for design space exploration of such magnitude which eliminates the advantage of limited time to market

Proposed Solution

- Formulation of different design models that identify the optimal **hybrid MPSoC design** for each application, taking into account **constraints** by the designer

- One of the approaches used is formulating and solving the problem by using **Integer Linear Programming**

MPSoC Architecture Model

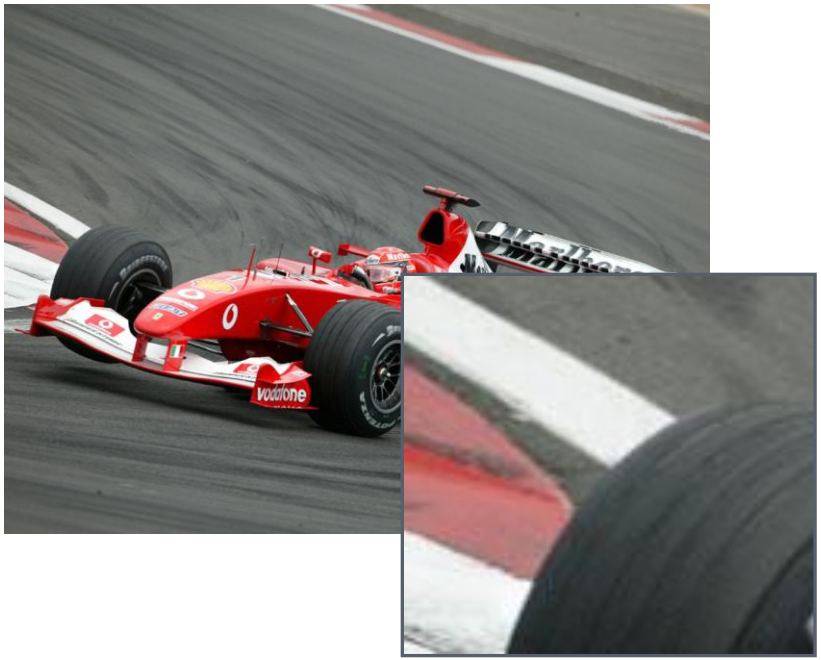


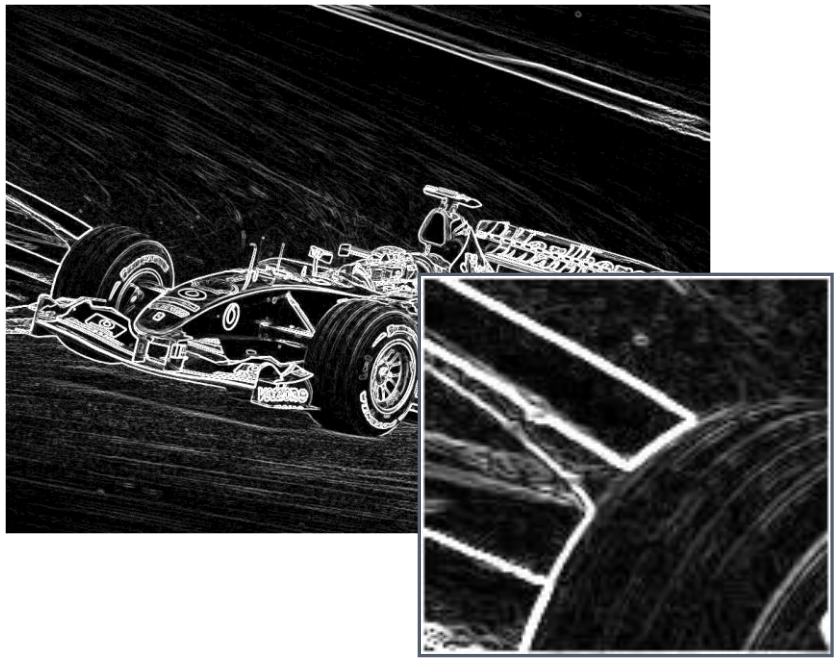
Programmable Logic (FPGAs) for Computationally Intensive Applications

- Embedded Systems
- FPGAs
- VHDL
- FPGA MPSoCs
- Canny Edge Detection
- Machine Vision Implementation

Edge Detection

- Edge Detection is used to identify sharp discontinuities in an image, such as changes in luminosity or in the intensity due to changes in scene structure.





Edge Detection

- Edge Detection implementation leads to a set of connected curves that indicate the boundaries of objects or surfaces.
- It is used to reduce the amount of data to be processed and filter out information that is less relevant to the next processing step.
- For grayscale images it produces a binary output.

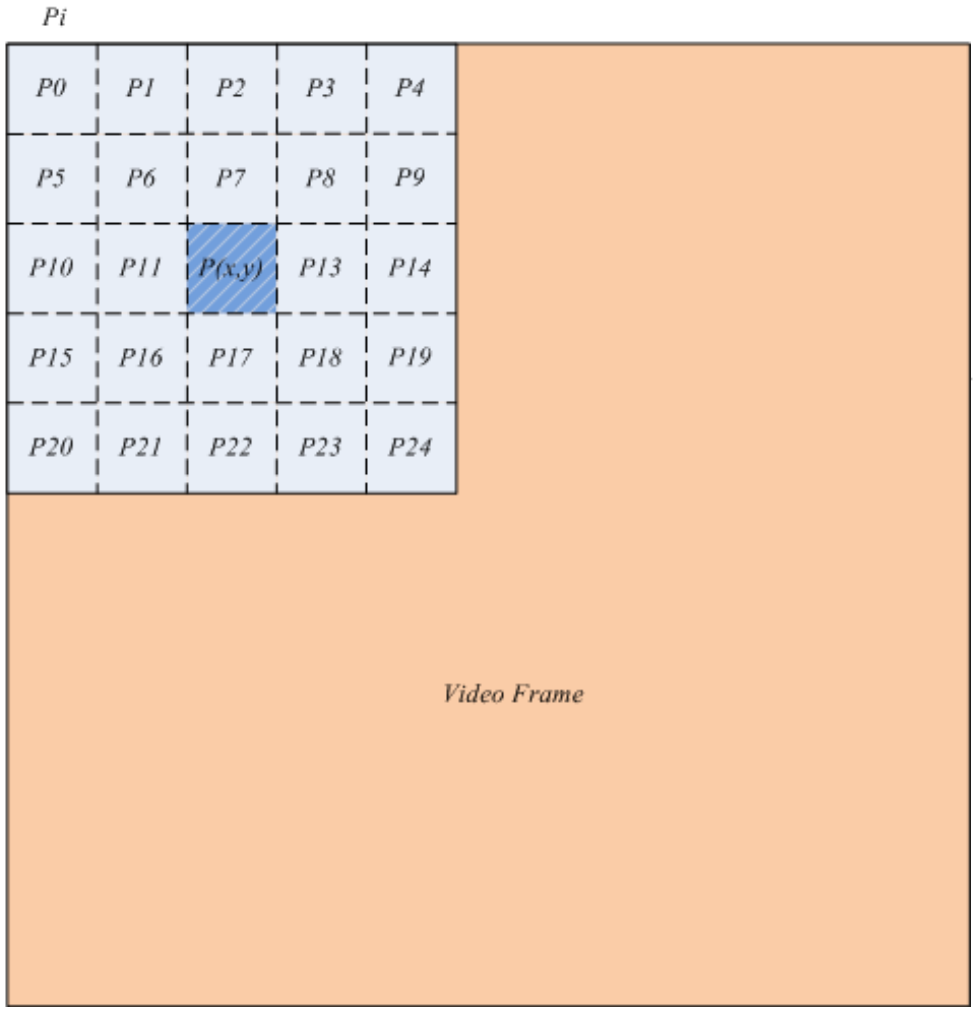
- It is used as the first step in many image processing algorithms, such as pattern matching, motion estimation, feature extraction, texture analysis etc.

Motivation

- Why use the Canny algorithm
  - It is one of the most reliable edge detection algorithms
  - Achieves low error-rate in the detected edges
  - Improves the localization of the identified edges
- The need for Real-Time/High Throughput Implementation
  - Multiplication of camera resolutions in recent years
  - Real-time applications
- The performance of modern FPGA devices
  - Powerful, efficient, availability of memory resources and DSP specific slices

Canny Edge Detection

$$G = \frac{1}{159}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} \circledast$$

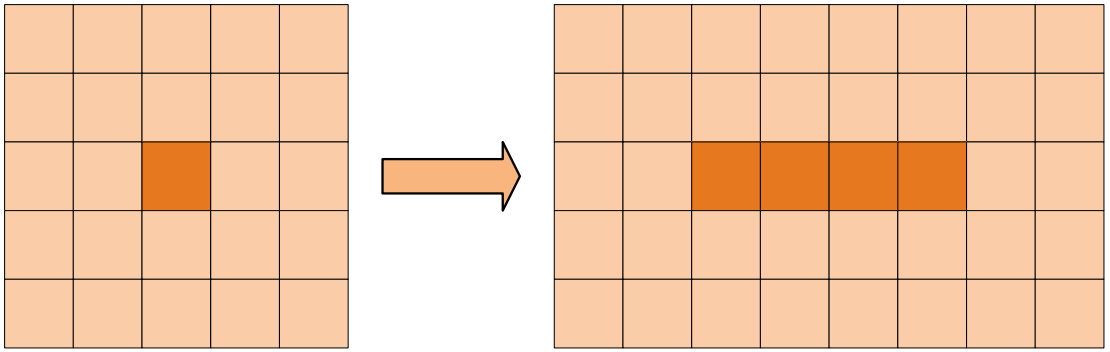


Canny Edge Detection

- The computational stages of Canny Edge Detection sequence

Gaussian Smoothing

- 5 x 5 convolution
  - Introducing 4-pixel parallelism

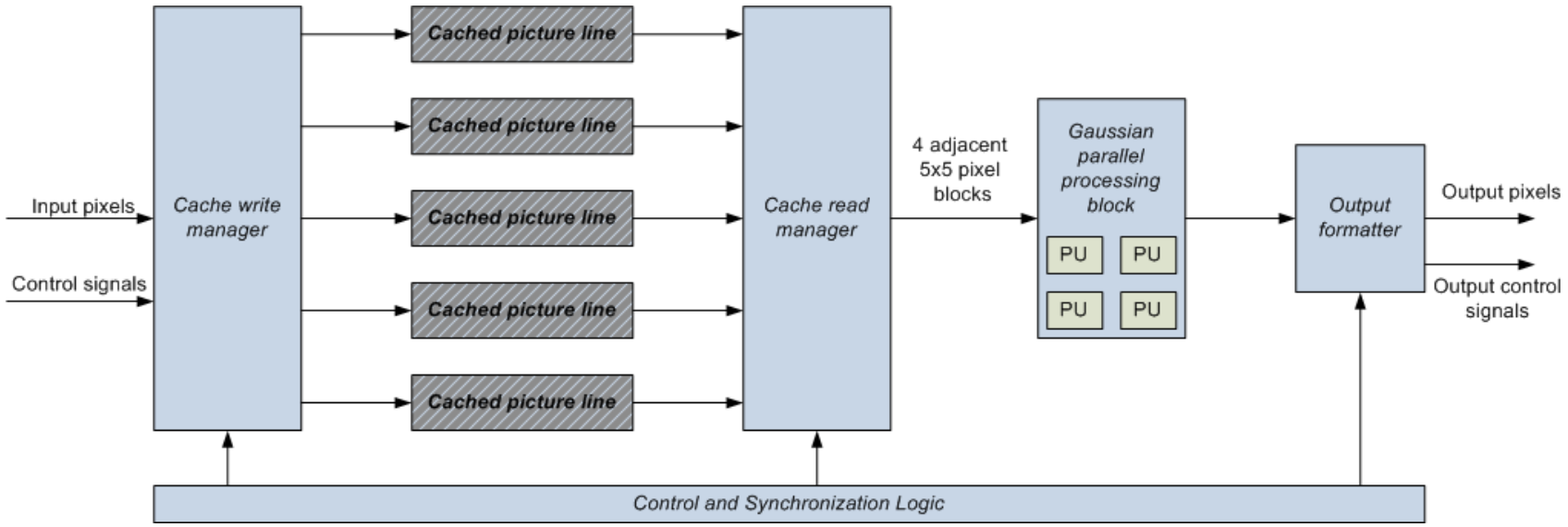


25 pixels

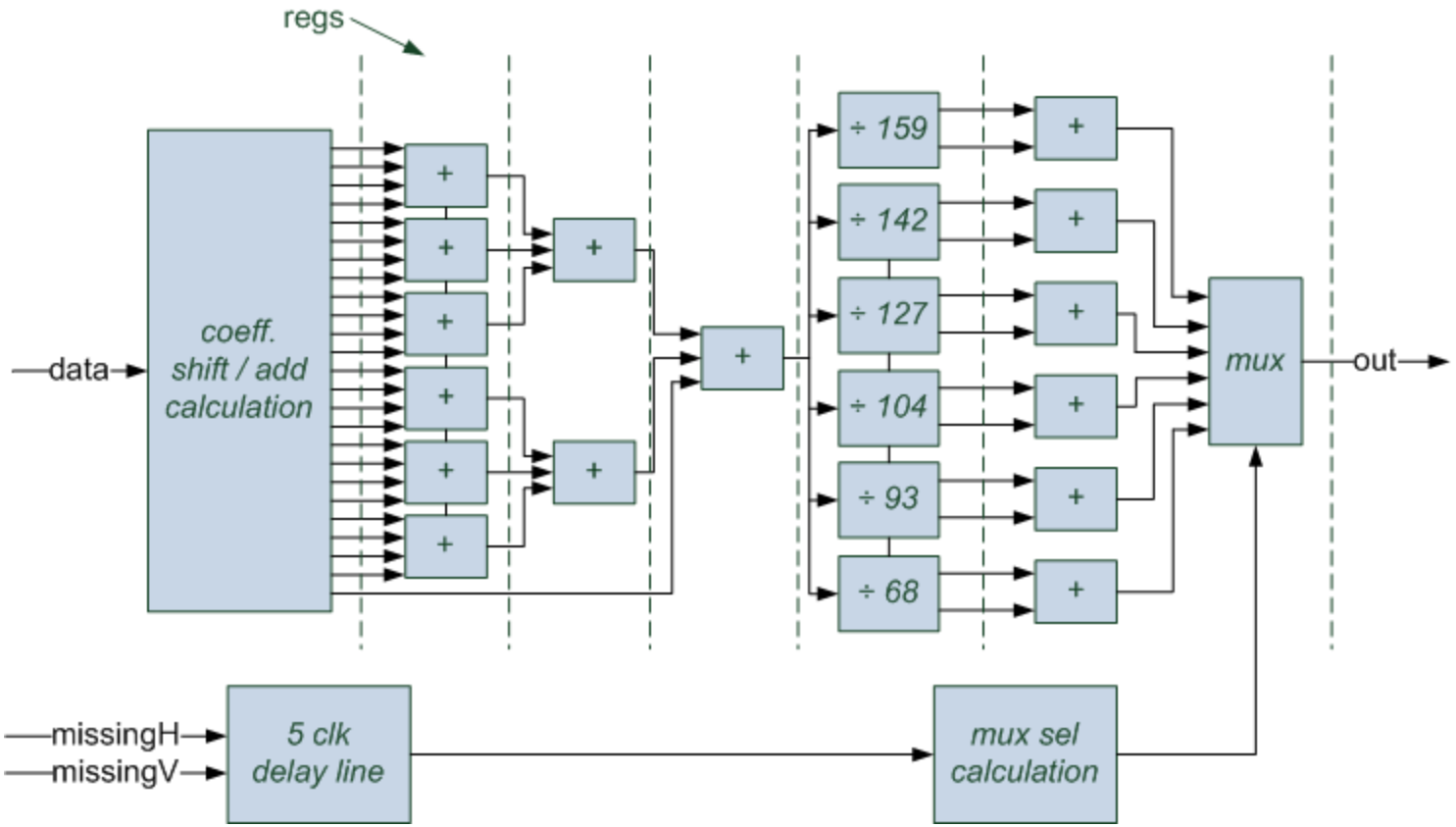
40 pixels

  - Substitution of the multiplications and divisions with shifts additions and subtractions

Gaussian Smoothing



Gaussian Smoothing



Simulations - Results

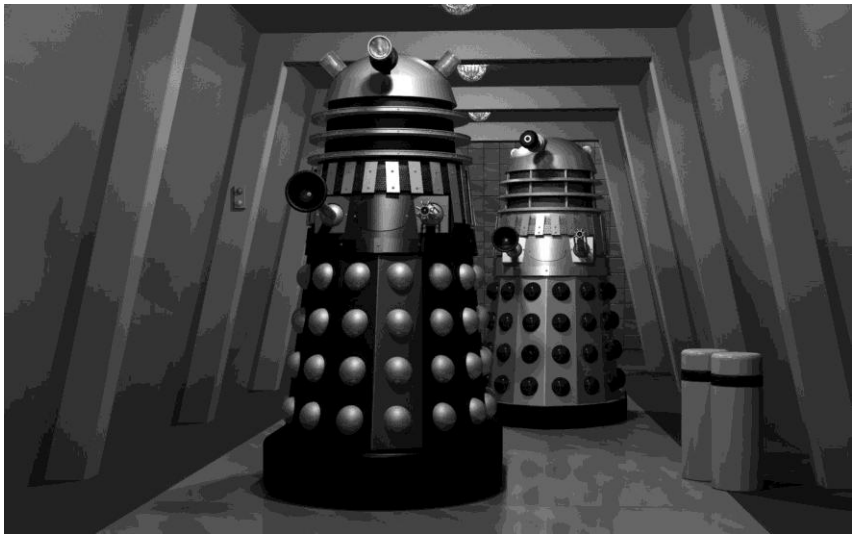
- Synthesis results for three different FPGAs

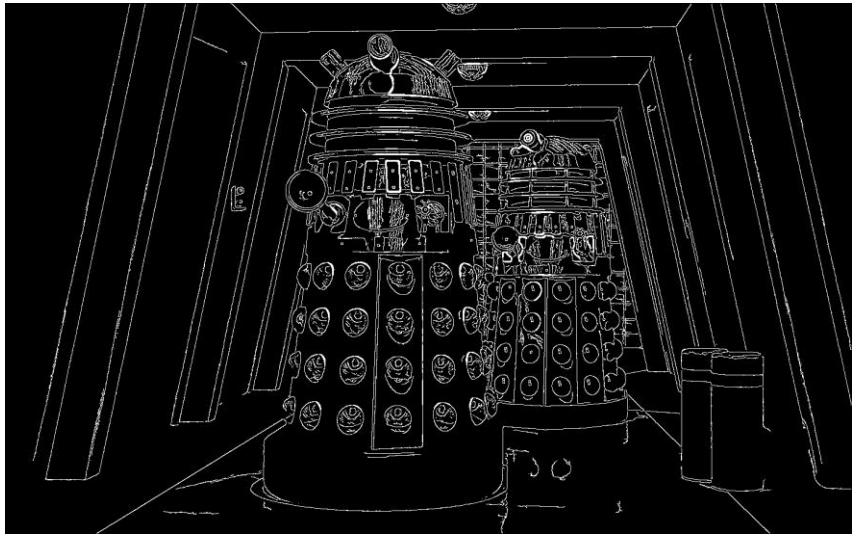
Synthesis Results	Gauss	Sobel	NMS	Db_Thres	Hysteresis	Total	Total(%)	Frequency (MHz)
Spartan 3E Slices	2578	1095	656	45	84	4284	29%	167
Spartan 6 Slices	4075	1482	823	43	104	6470	7%	214
Virtex 5 Slices	3815	1426	960	40	96	6350	9%	350

Simulations - Results

- Timing results

Image File	Size	Time (ms) Spartan-3E	Time (ms) Spartan-6	Time (ms) Virtex-5
Lena	512x512	0.78	0.61	0.37
VGA resolution	640x480	0.91	0.71	0.43
Daleks	1440X900	3.88	3.02	1.85

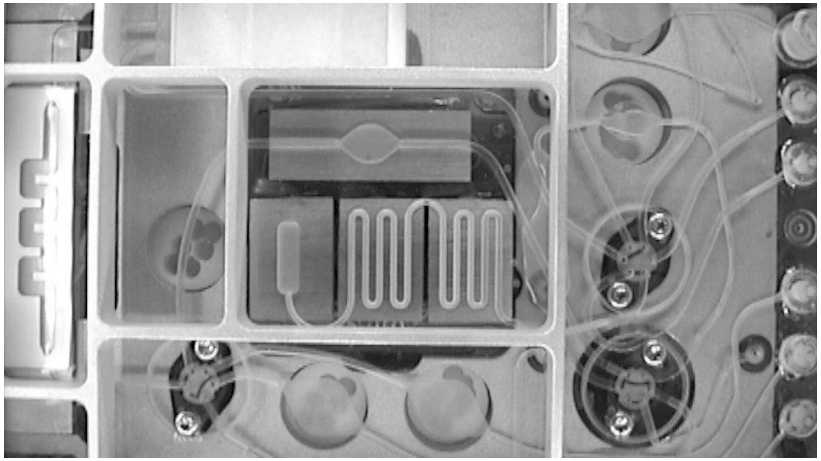


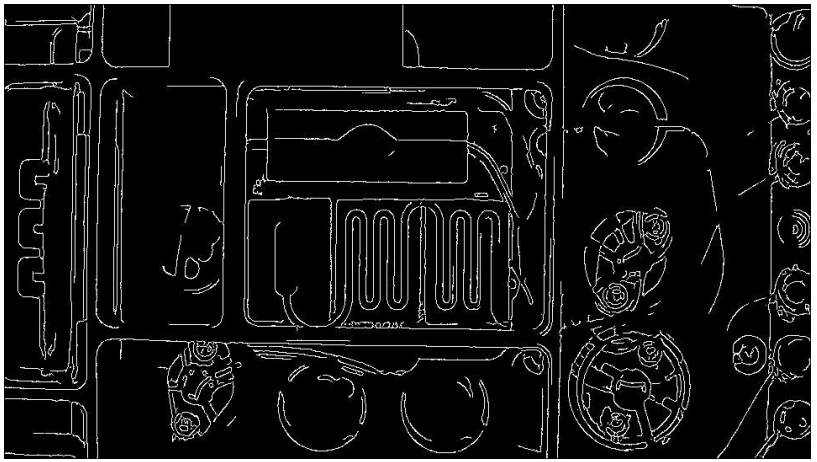


Simulations - Results

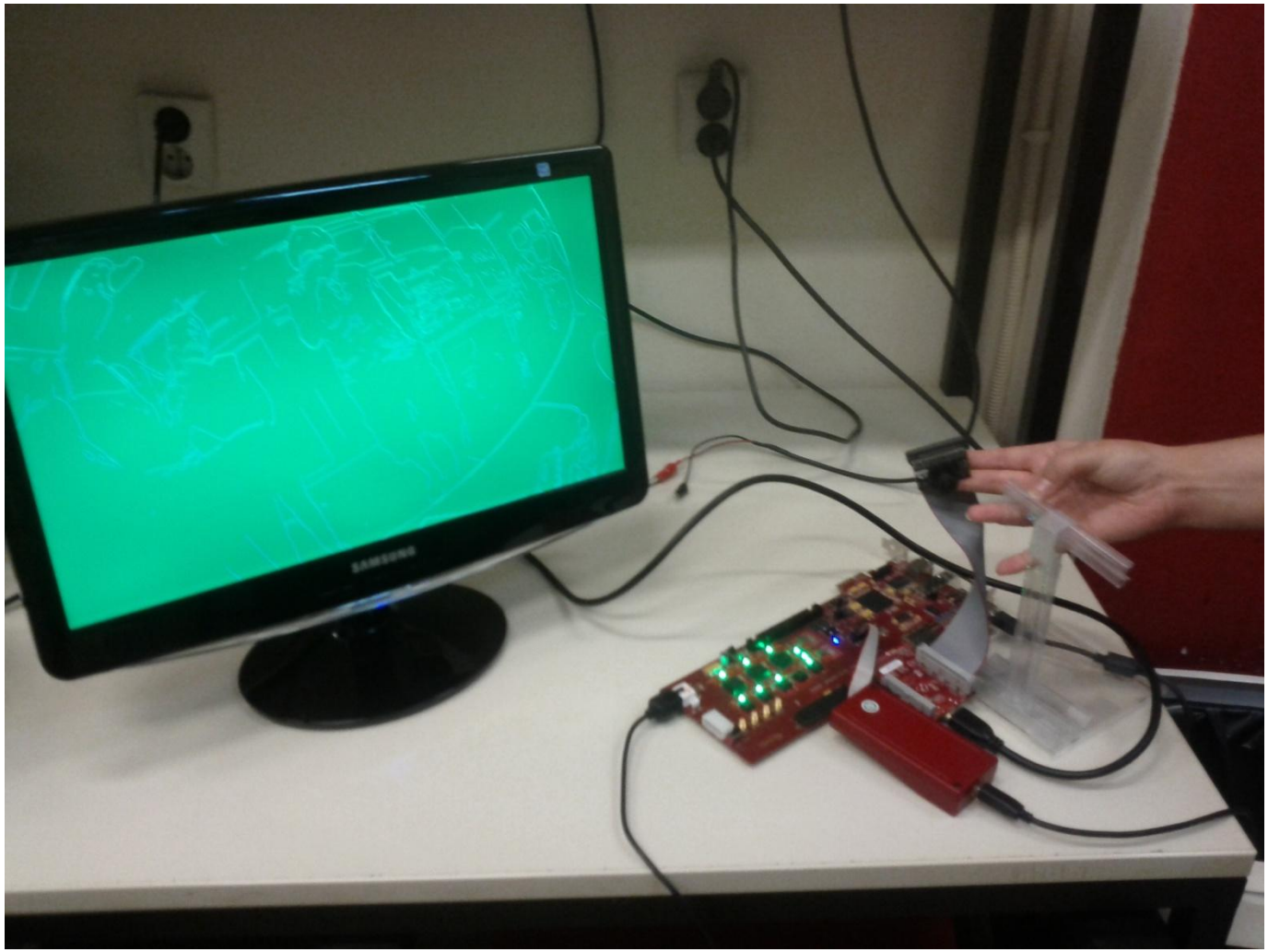
- Throughput

Image File Size	fps Spartan-3E	fps Spartan-6	fps Virtex-5
1Mpixel	318	408	667





Canny Implementation with Camera

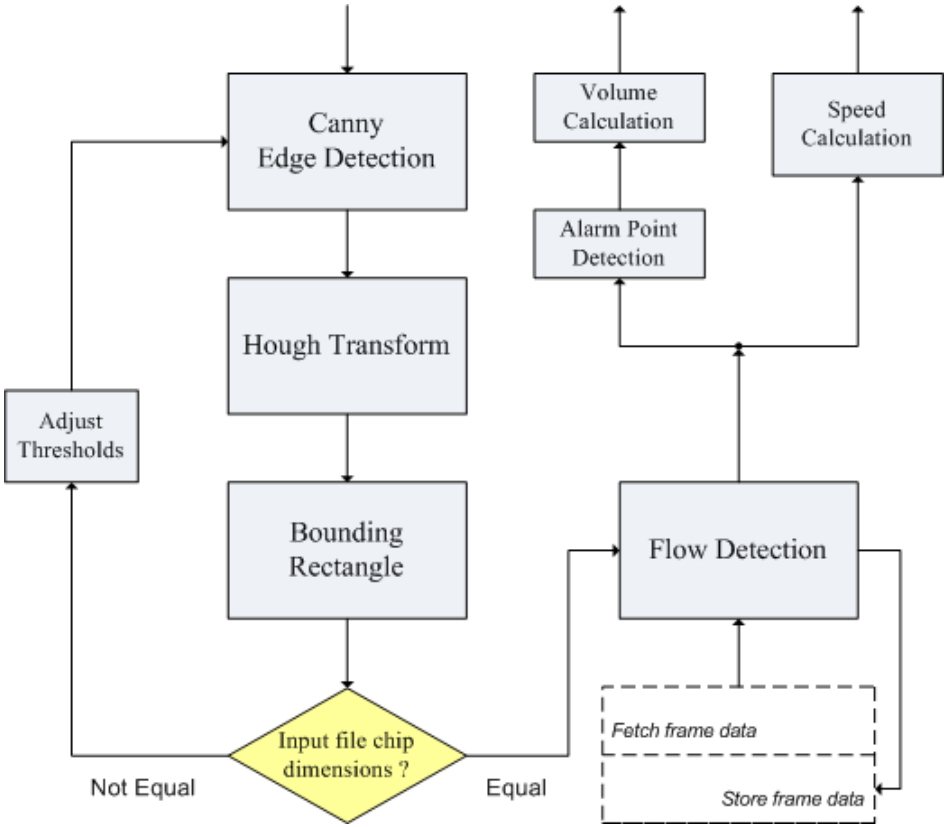


Programmable Logic (FPGAs) for Computationally Intensive Applications

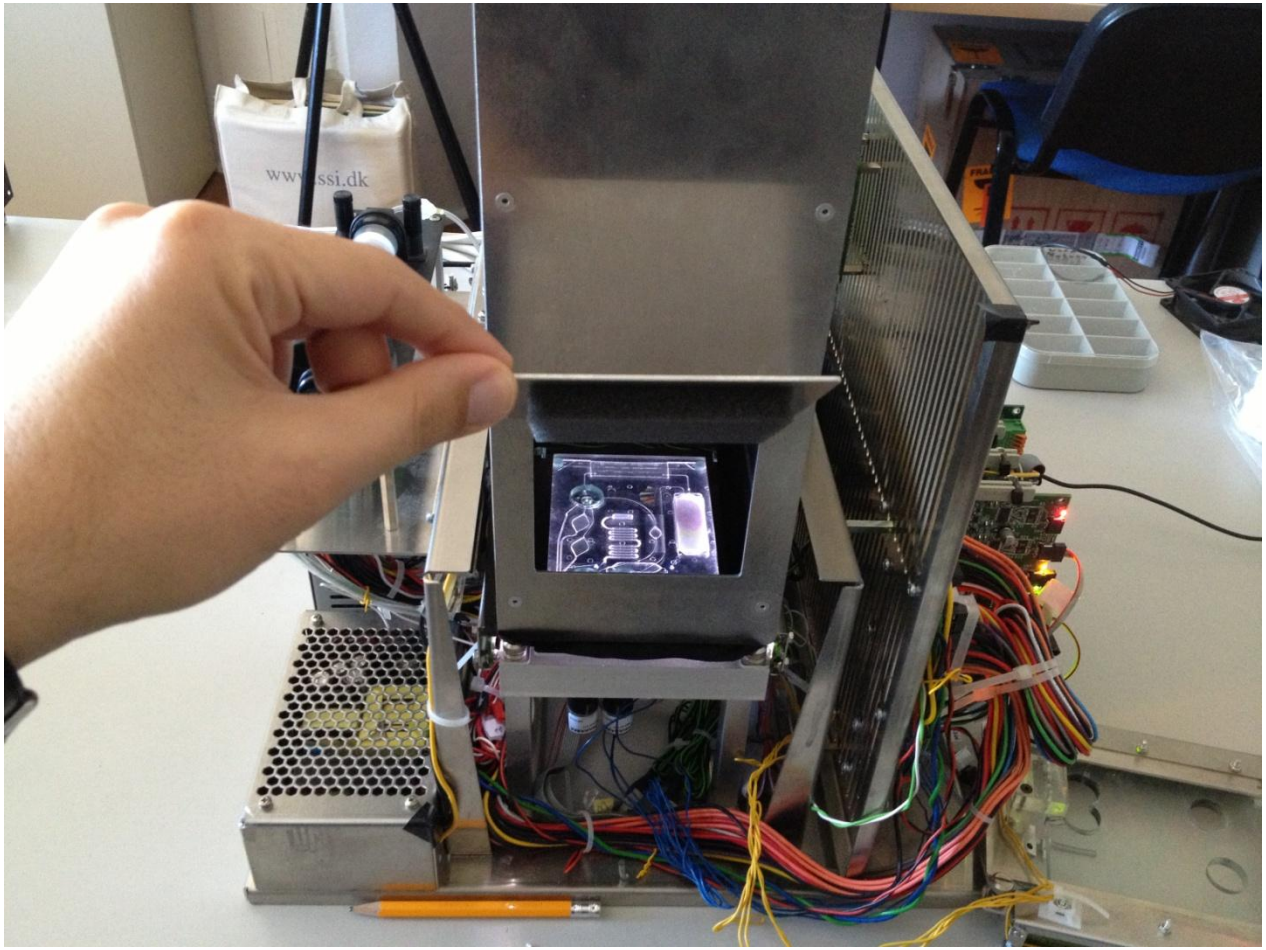
- Embedded Systems
- FPGAs
- VHDL
- FPGA MPSoCs
- Canny Edge Detection
- Machine Vision Implementation

Machine Vision Implementation

- Machine Vision Flow chart



Machine Vision Implementation



Machine Vision Implementation

- Original Video and Video with Flow Detection Implemented

- Specifications: 60fps for 1Mpixel input video (it actually achieves more than 70fps)

Conclusions

- FPGAs are powerful reconfigurable devices which can be used to implement computationally intensive applications
- The tools are provided by the FPGA vendors and usually there is a free version available
- FPGAs can be programmed using VHDL
- Very powerful algorithms can be implemented with these tools
- And all this can be done with just one pc (and one FPGA…)

And now….

- You can start your own designs! :-)

Thank you!