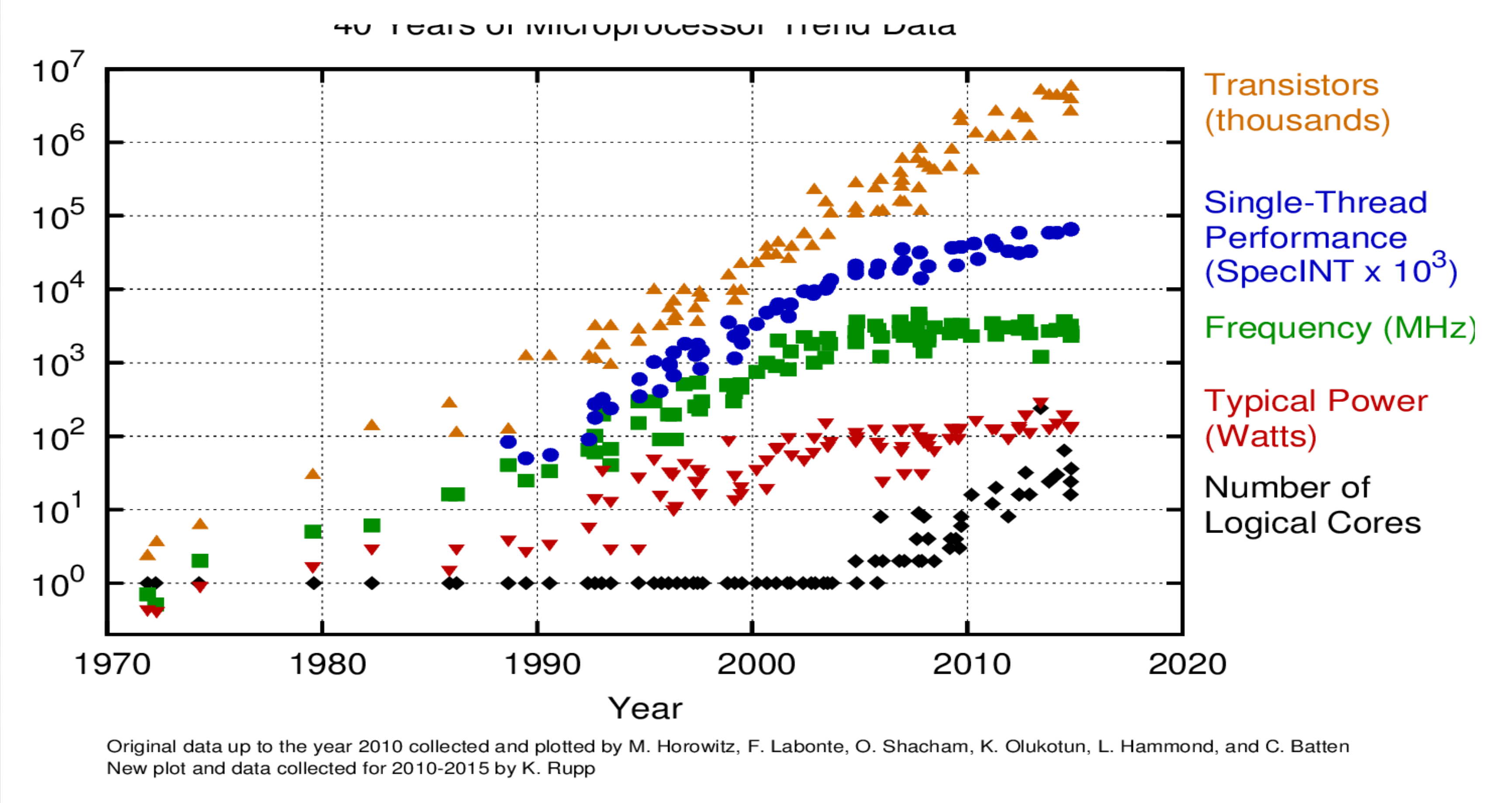
ALL PROGRAMMABLE
ANY MEDIA
5G
4K/8K
ANY STANDARD
ANY MACHINE
ANY NETWORK
5G Wireless • SDN/NFV • Video/Vision • ADAS • Industrial IoT • Cloud Computing

XILINX
ALL PROGRAMMABLE™

Unleashing the full performance of the All Programmable FPGA while abstracting the hardware details

Ivo Bolsens

Traditional Compute Architectures Are Not Scalable



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2015 by K. Rupp

Requires Heterogeneous Architectures & Acceleration

Heterogeneous Domain Optimized Computing Platforms

Wiley Series on Parallel and Distributed Computing
Albert Y. Zomaya, Series Editor
High Performance Heterogeneous Computing
Alexey L. Lastovetsky and Jack J. Dongarra
WILEY
Wim Vanderbauwhede · Khaled Benkrid
Editors
High-Performance Computing Using FPGAs
Springer
Heterogeneous Computing with OpenCL 2.0
David Kaeli
Perhaad Mistry
Dana Schaa
Dong Ping Zhang
MK

Data Center enabling the Cloud

- HPC
- Deep Learning Training
- Deep Learning Inference
- Image & Video Acceleration
- Data Analytics
- Genomics

Computing

- Network Acceleration
- NFV NICs
- SDN Controllers
- Compression

Networking

- Flash Arrays Acceleration
- SSDs Acceleration
- NVDIMM
- NVMe over Fabric

Storage

Data Center enabling the Cloud

GPU

- HPC
- Deep Learning Training
- Deep Learning Inference
- Image & Video Acceleration
- Data Analytics
- Genomics

Computing

NIC

- Network Acceleration
- NFV NICs
- SDN Controllers
- Compression

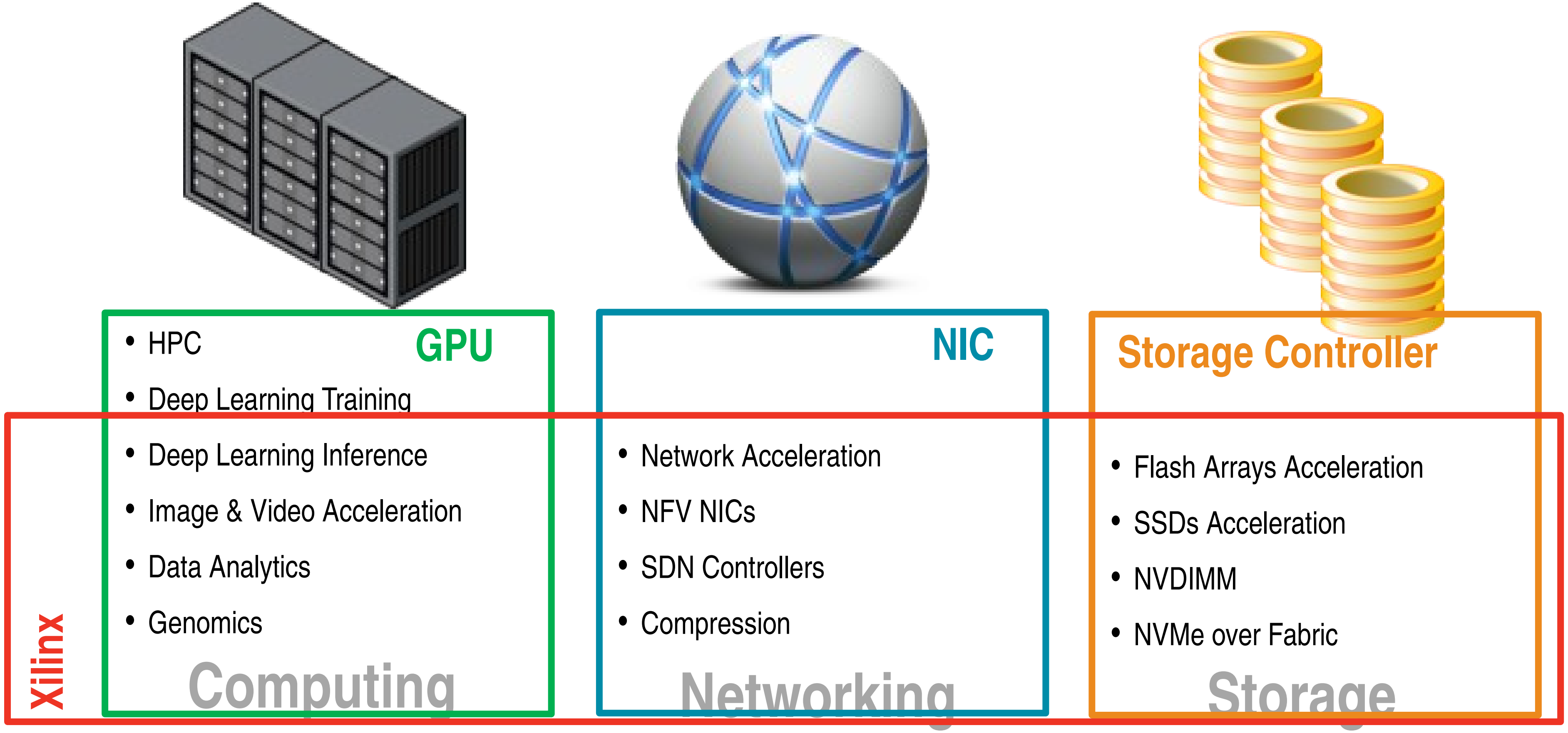
Networking

Storage Controller

- Flash Arrays Acceleration
- SSDs Acceleration
- NVDIMM
- NVMe over Fabric

Storage

Data Center enabling the Cloud



GPU

- HPC
- Deep Learning Training

Xilinx

- Deep Learning Inference
- Image & Video Acceleration
- Data Analytics
- Genomics

Computing

NIC

- Network Acceleration
- NFV NICs
- SDN Controllers
- Compression

Networking

Storage Controller

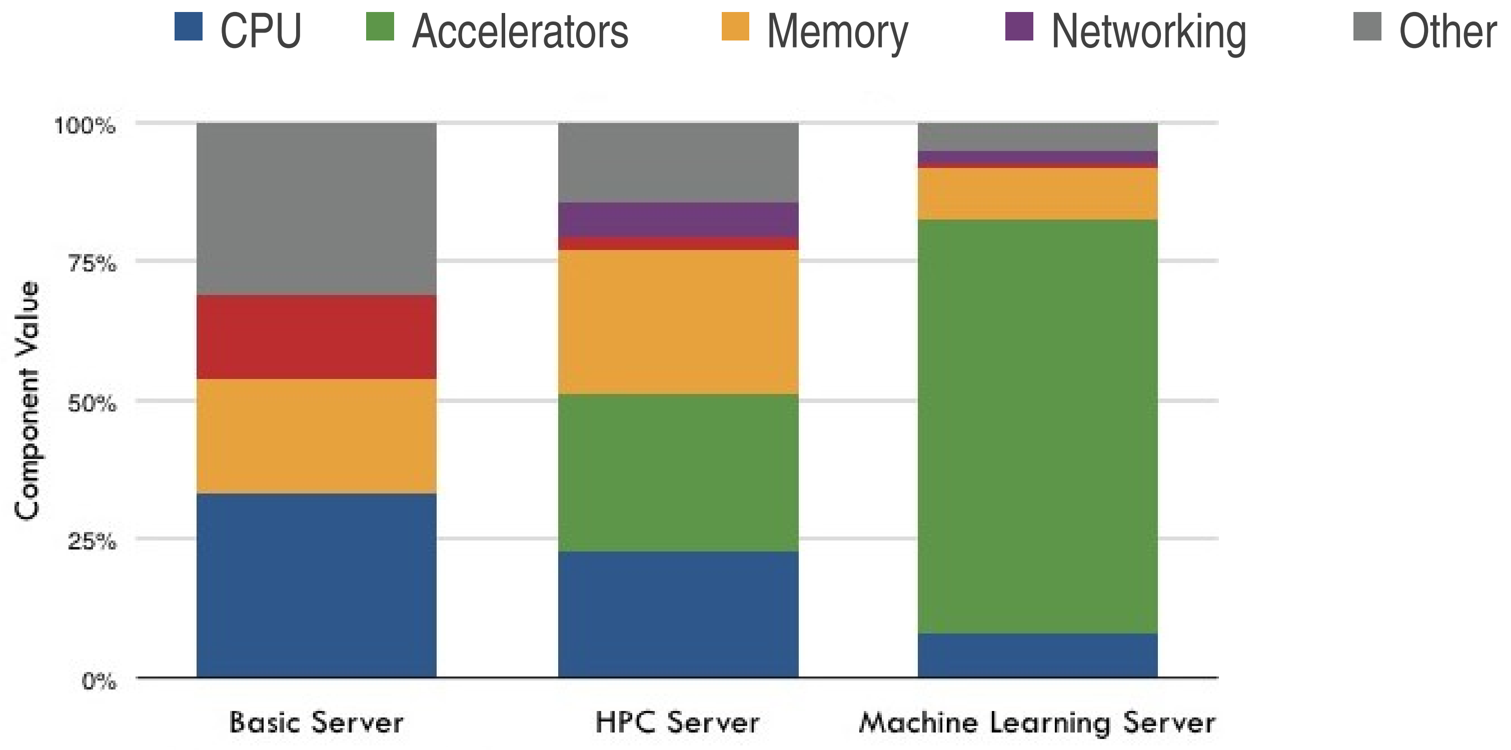
- Flash Arrays Acceleration
- SSDs Acceleration
- NVDIMM
- NVMe over Fabric

Storage

Unified HW to address compute, storage, and networking apps

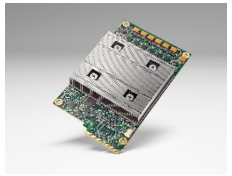
Heterogeneous Compute Platforms in Datacenter

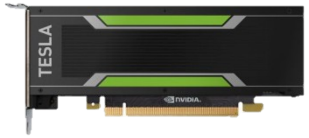
How Server Components Change with Machine Learning



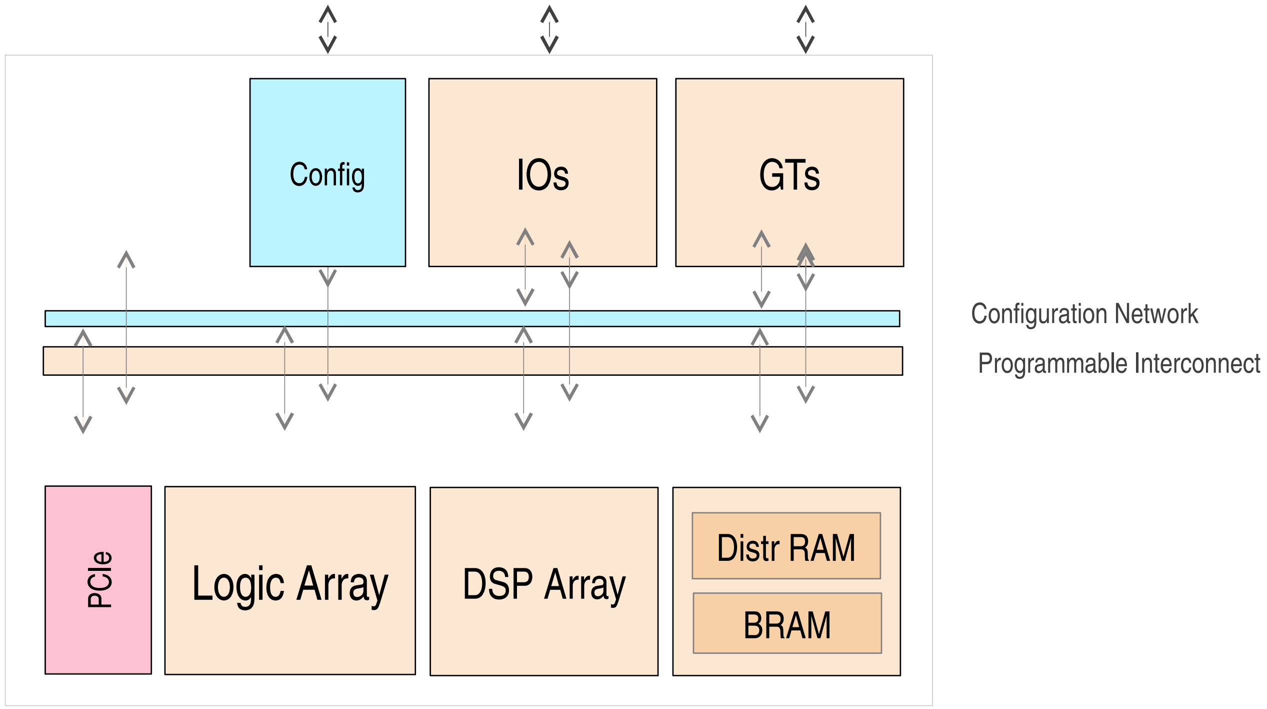
Source: ARK Investment Management LLC |

Machine Learning
ASICs (TPU by Google)
FPGA (Xilinx, Intel)
GPU (NVidia Tesla P4)



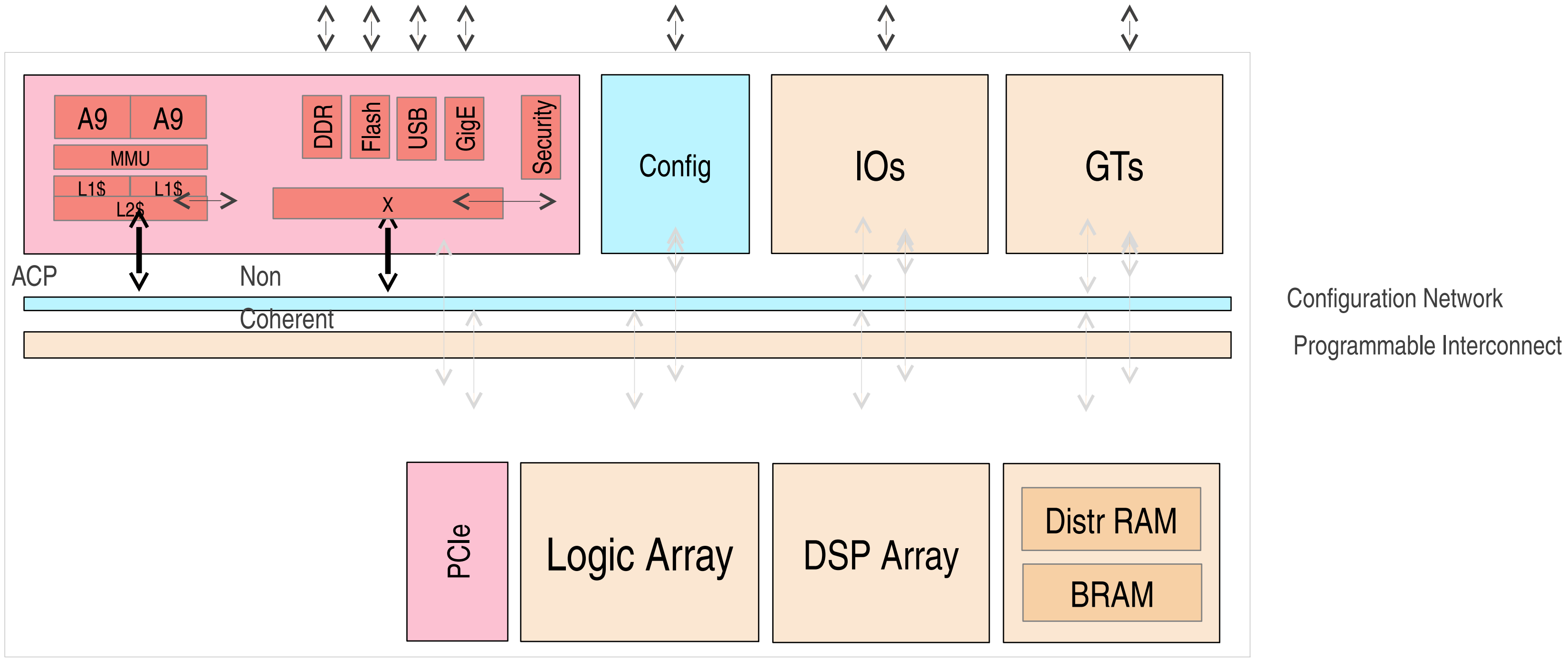


FPGA Silicon Architecture



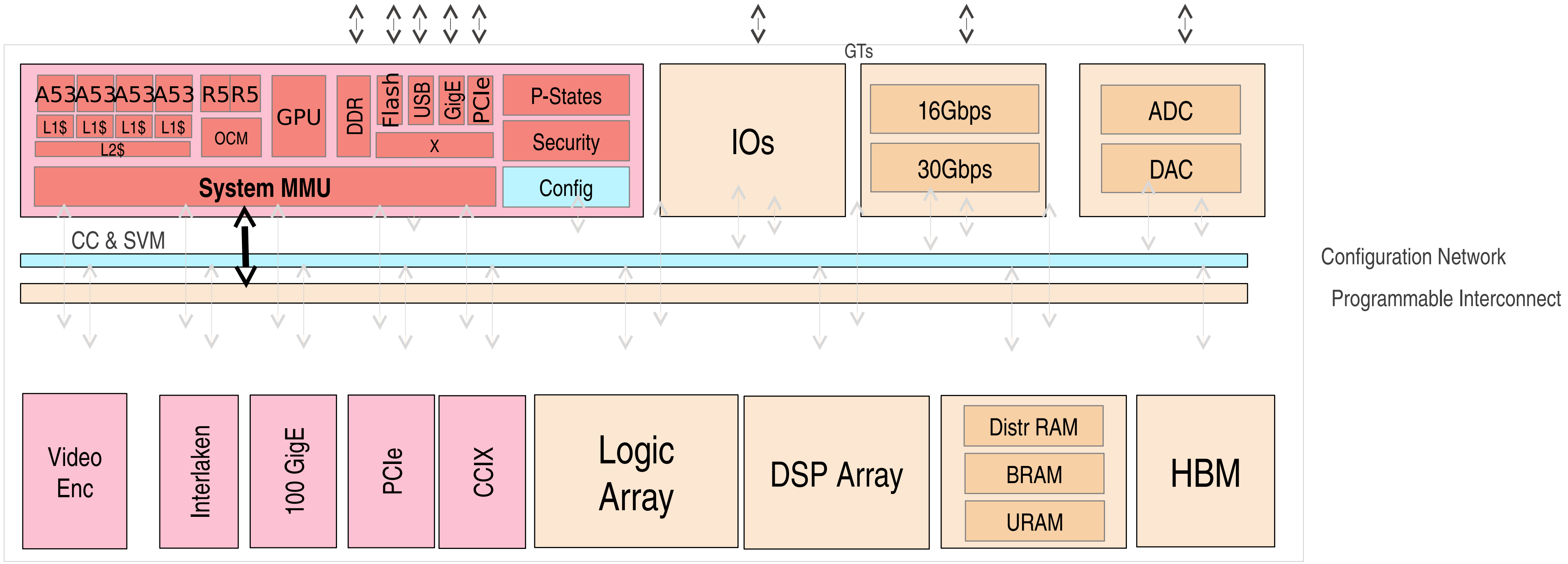
Standard FPGA

FPGA Silicon Architecture



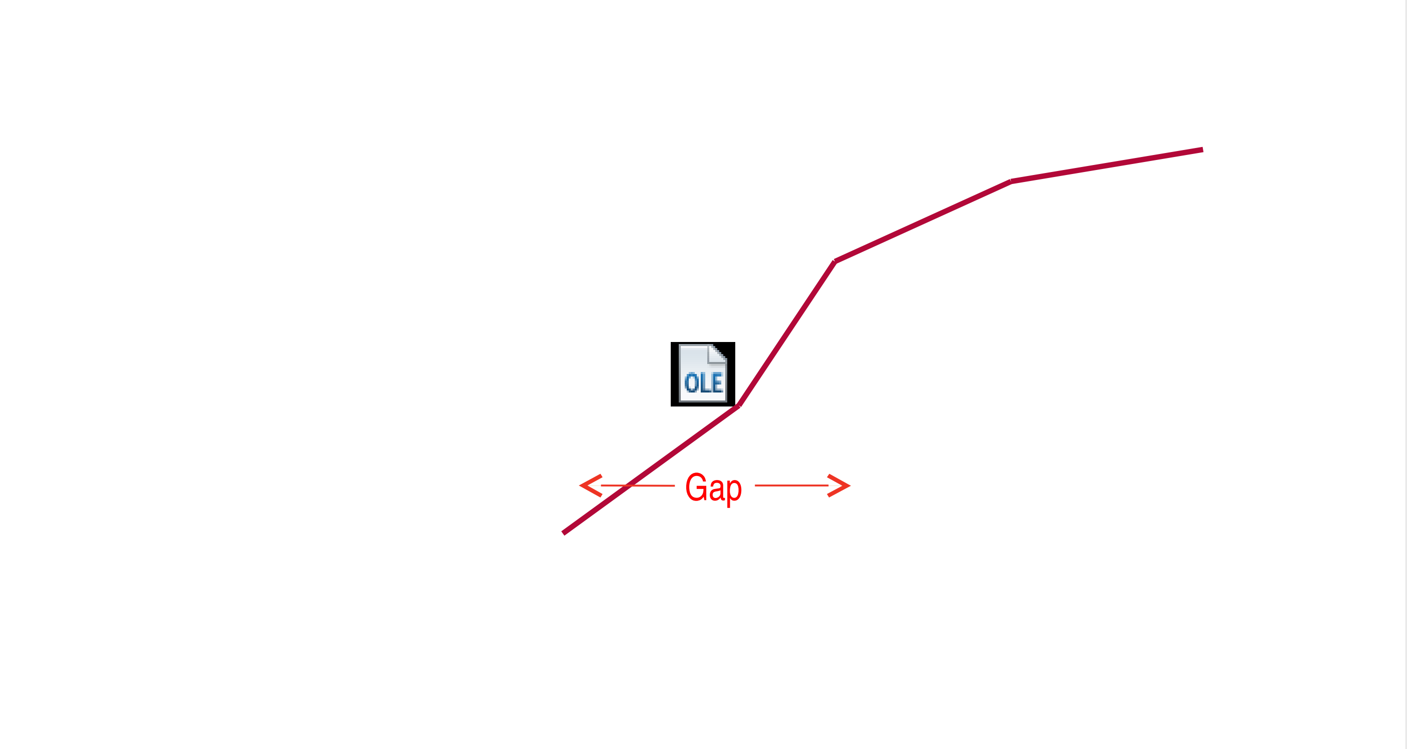
Zynq 7000 (28nm)

All Programmable FPGA Silicon Architecture



Zynq MPSoC (16nm)

Efficiency Programming Experience



FPGA: Performance Advantage, But Productivity Gap

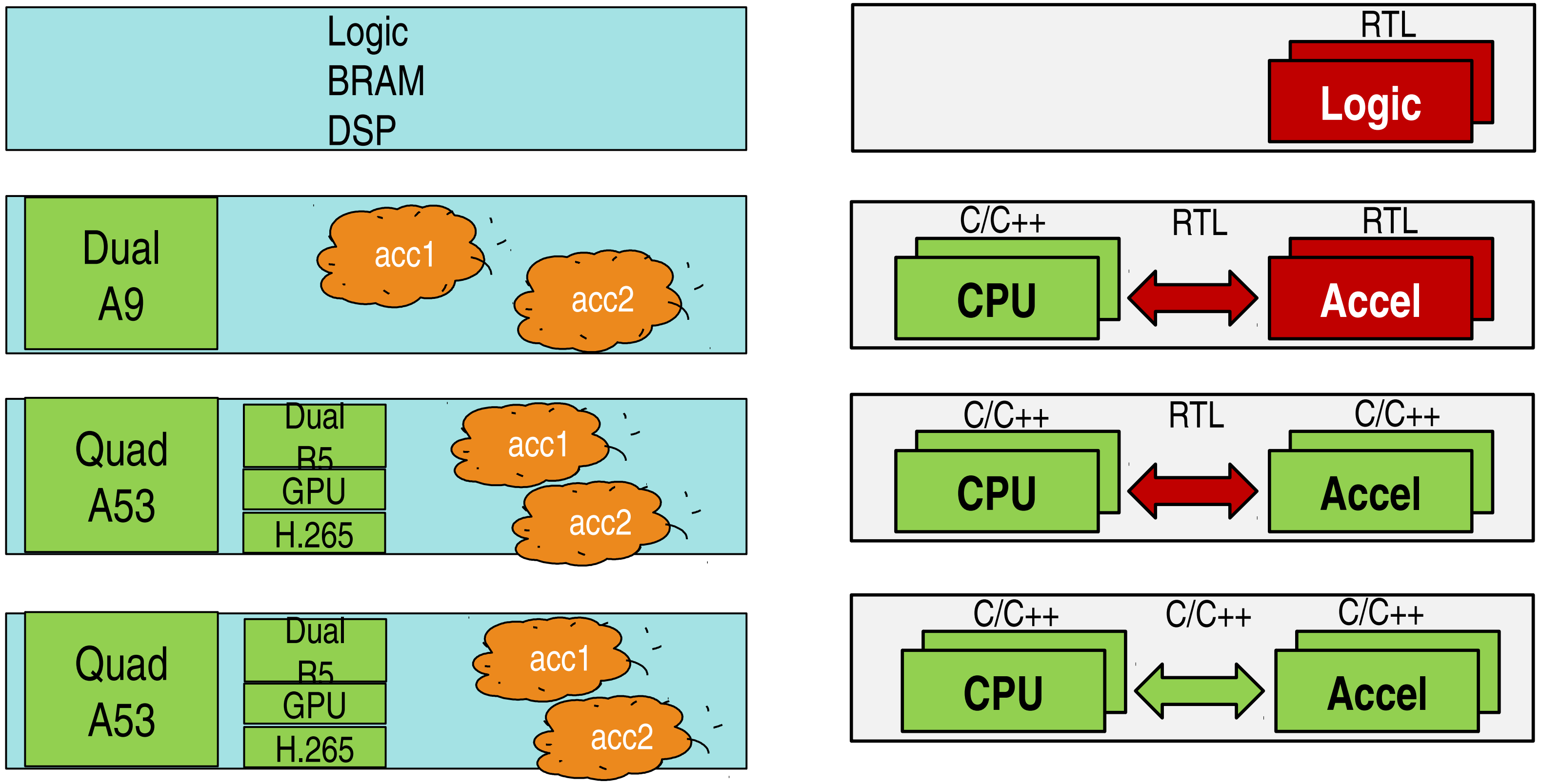
Source: David Thomas, Imperial College London
Adopted and extended by Xilinx

Programming Heterogeneous Parallel Platforms

ASIC Refugees

HW/SW co-design

Software Programming



VHDL

C + VHDL

C + High Level Synthesis

HLx

OpenCL

SDx Environments

OpenCL

Heterogeneous Parallel Programming

SYCL

Bring power of C++ to OpenCL

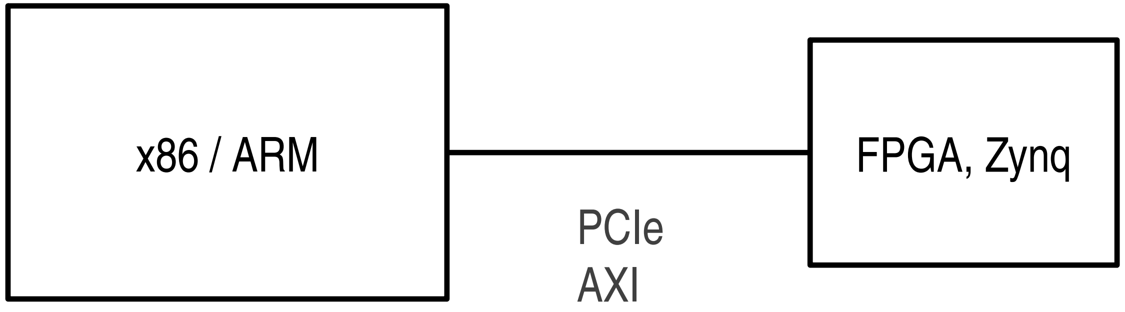
SPIR

Intermediate Representation

Towards single source C++

- No special memory allocation (malloc)
- No special data movement (no copy needed)
- No special accessing accelerator memory (private memories)

x86 / ARM Host: Non-Coherent

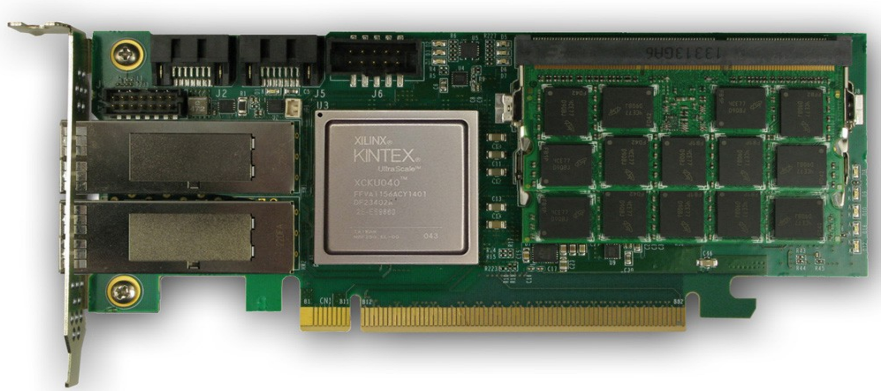


x86 / ARM

FPGA, Zynq

PCIe
AXI

Memory allocation special?	Yes
Data movement special?	Yes
Accelerator memory access special?	Yes



Product Description

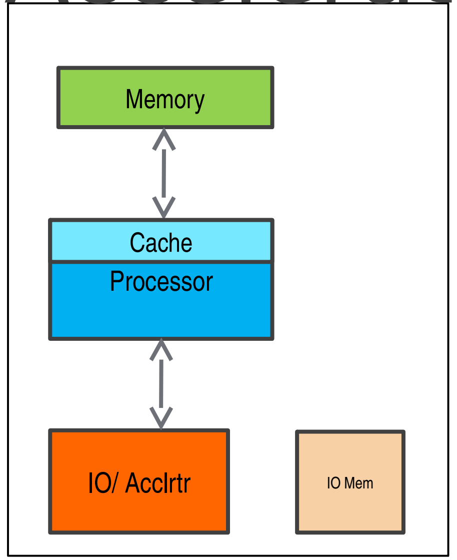
The ADM-PCIE-KU3 is a high performance reconfigurable Half-Length, low profile x16 PCIe form factor board based on the Xilinx Kintex UltraSCALE range of Platform FPGAs. The ADM-PCIE-KU3 features two independent channels of DDR3 memory capable of 1600MT/s (fitted with two 8GB SODIMMs), high speed I/O, SATA connections, Dual QSFP ports supporting 10G Ethernet, voltage/temperature/current control and monitoring, passive air-cooled heat sink.

Key Features & Benefits

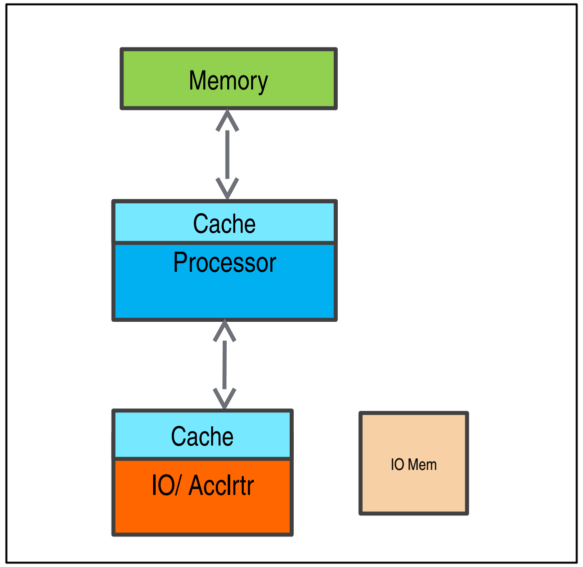
- Dual QSFP High Speed Communications ports
- Dual SATA High Speed Data Storage ports
- PCI Express x16 Interface
- TWO SODIMM slots

CCIX : Cache Coherent Interface for Accelerators

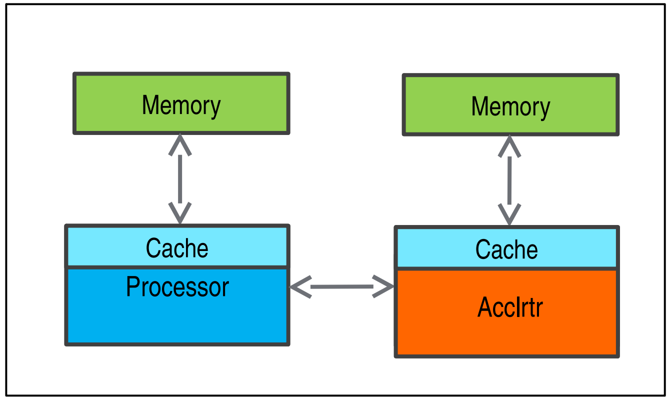
IO Coherent



Coherent Accelerator



Peer- Peer



IO Coherency

- Allows DMA with IO as master
- IO agent sees limited memory range
- One-way coherency. Processor memory not coherent with IO Memory
- Interface optimized for large transfers
- E.g.: PCIE

Coherent Accelerator

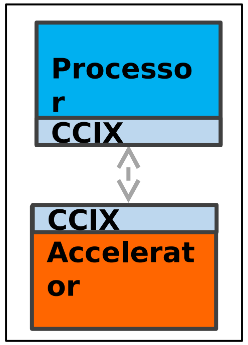
- Caching (always), Home Node (limited) capability
- Typically a standard interface
- Protocol Bridging function
- E.g.: IBM CAPI, nVidia nvLink, OpenCAPI

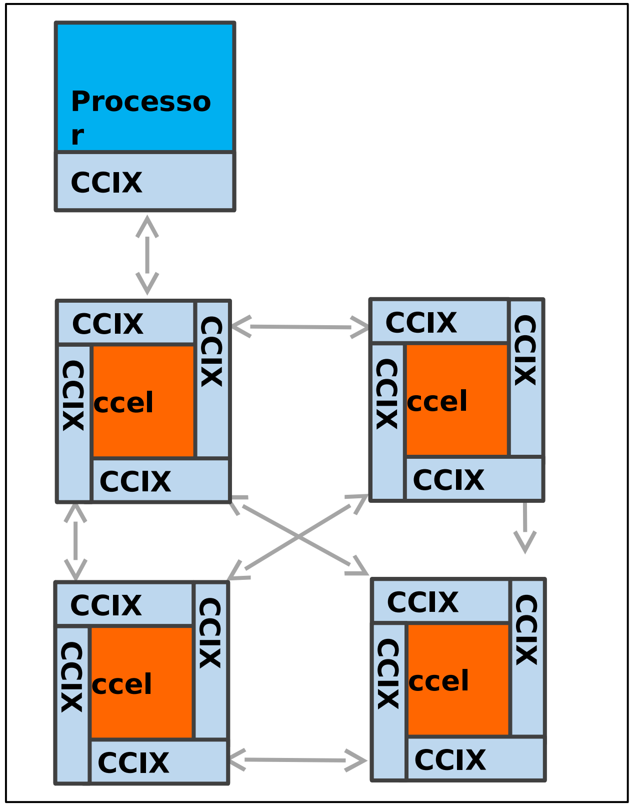
Coherent Peer-peer

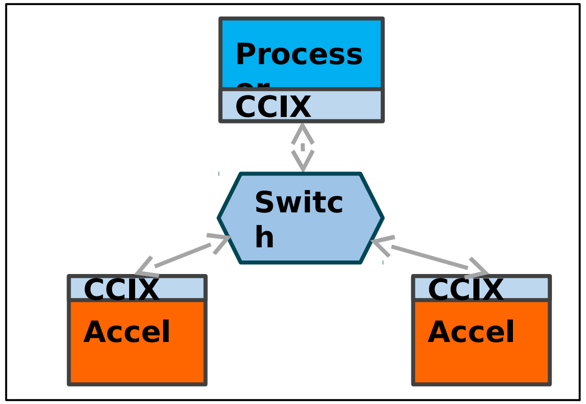
- All agents can cache other agents memory
- Each agent is peer and home node
- Mostly proprietary interface
- E.g.: CCIX

CCIX : Open Source

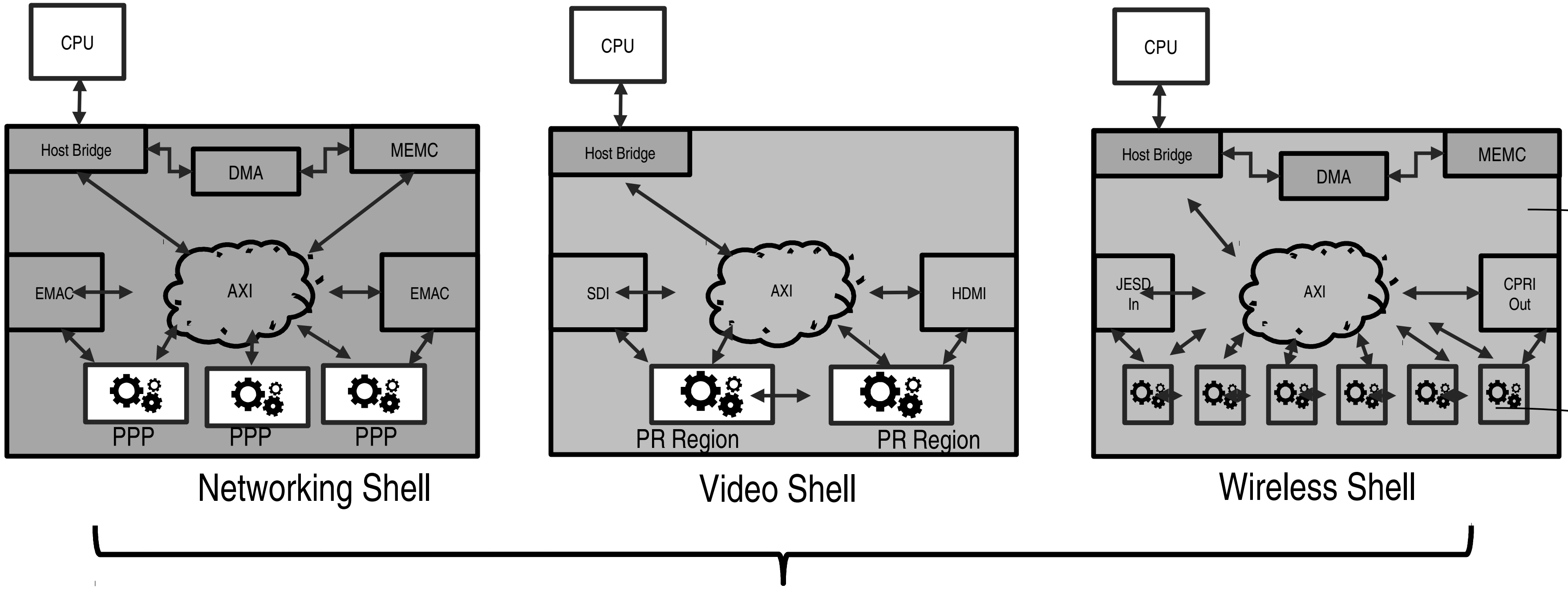
CCIX: Accelerator – CPU Configurations

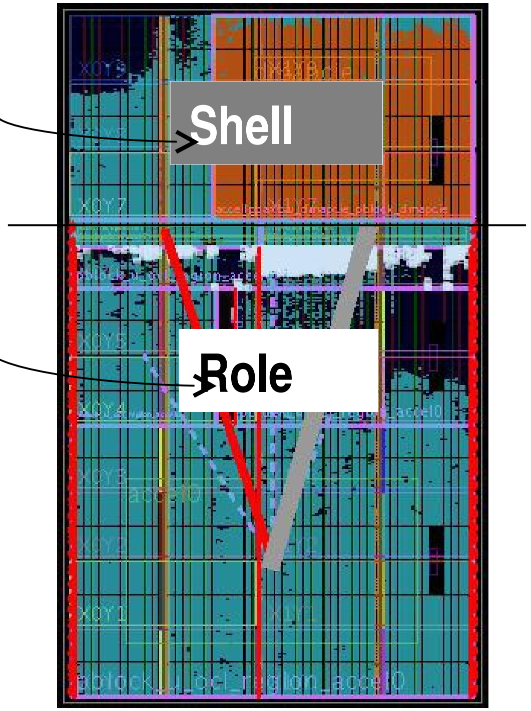






Domain Specific Platform Infrastructure





One Board

Multiple Shells

Shell : Domain specific infrastructure (gray)

Role : ‘Donut holes’ in FPGA or on CPU executing programmable functions (white)

Use Model and Personas

HW/PCB designer

CAD tools, PCB design tools

HW designer

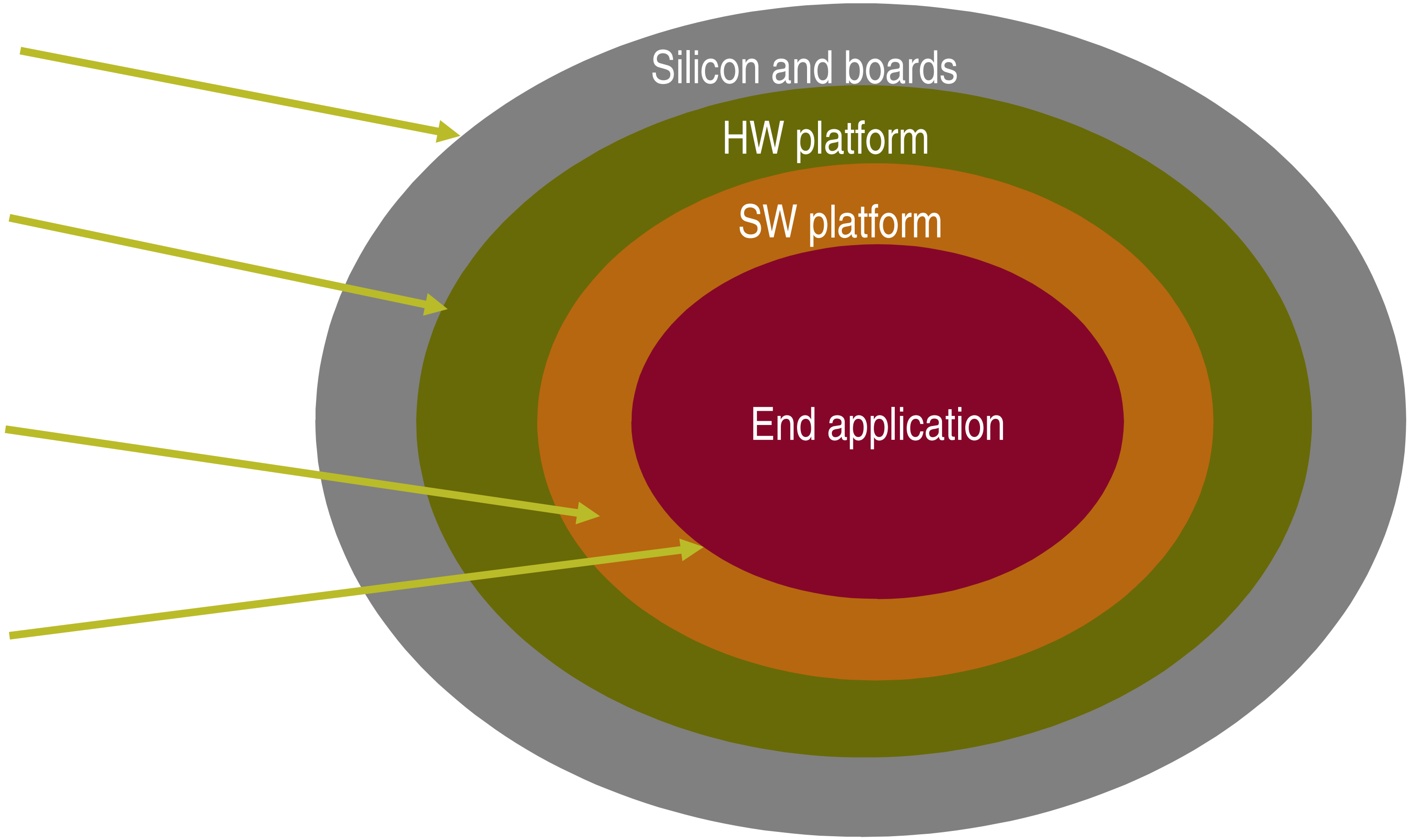
HLx

Firmware designer

SDK

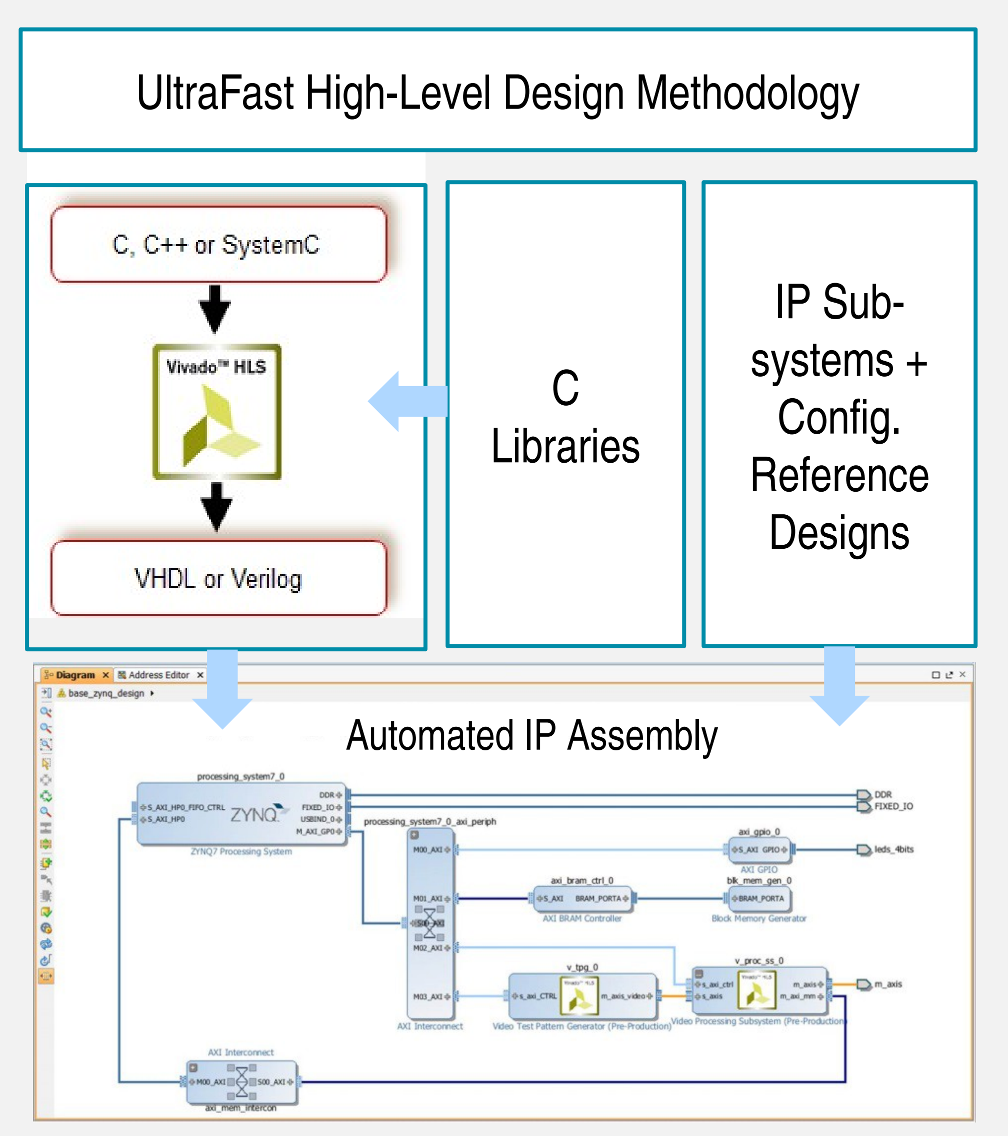
Application developer

SDx Environments



Hardware : HLx – High-Level Design

HLS
High-Level IP Creation



UltraFast High-Level Design Methodology

C, C++ or SystemC

VHDL or Verilog

C Libraries

IP Sub-systems + Config. Reference Designs

Automated IP Assembly

Typically 5-15x productivity improvement via:

- Creation of HW-optimized functions in C/C++
- Accelerated verification (>1000X RTL)
- Automated, intelligent assembly (15x manual)

XILINX ALL PROGRAMMABLE.

Hardware : HLS Control & Datapath Synthesis

Code

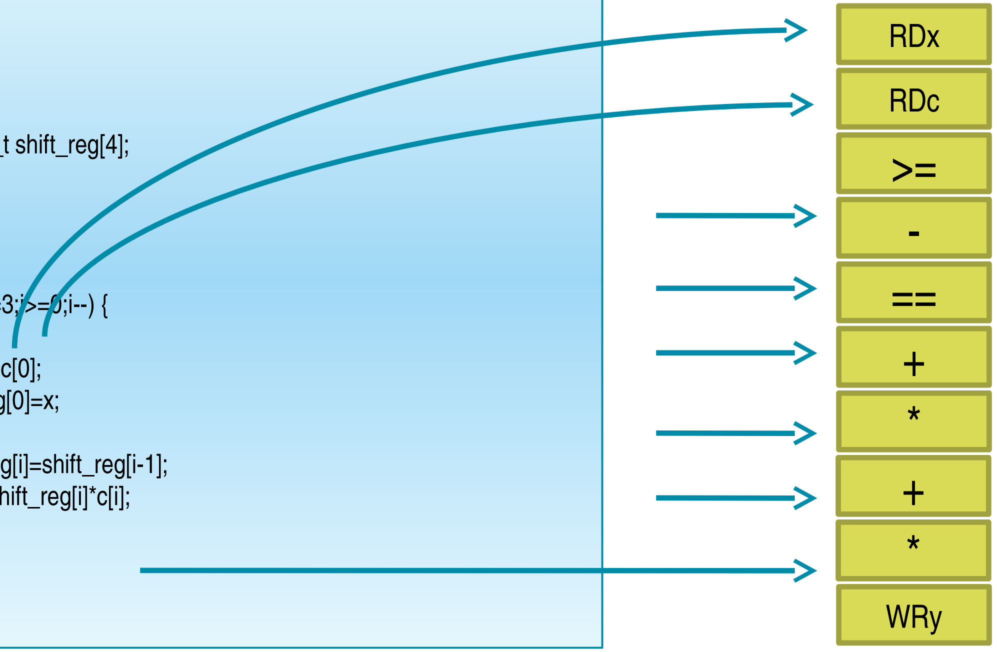
```
void fir (
  data_t *y,
  coef_t c[4],
  data_t x
  ) {

  static data_t shift_reg[4];
  acc_t acc;
  int i;

  acc=0;
  loop: for (i=3;i>=0;i--) {
     if (i==0) {
       acc+=x*c[0];
       shift_reg[0]=x;
     } else {
        shift_reg[i]=shift_reg[i-1];
        acc+=shift_reg[i]*c[i];
     }
  }
  *y=acc;
}
```

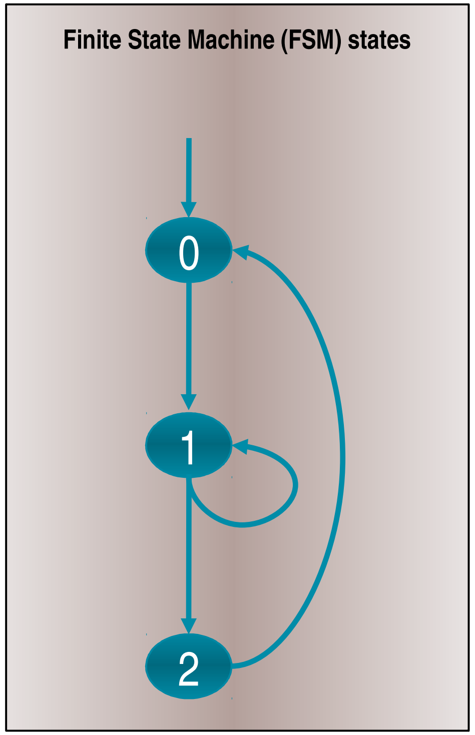
From any C code example ..

Operations



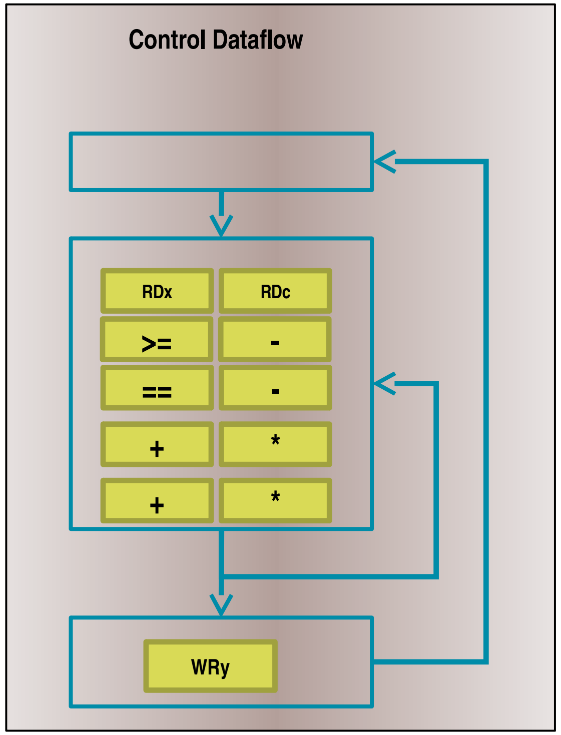
Operations are extracted…

Control Behavior



The control is known

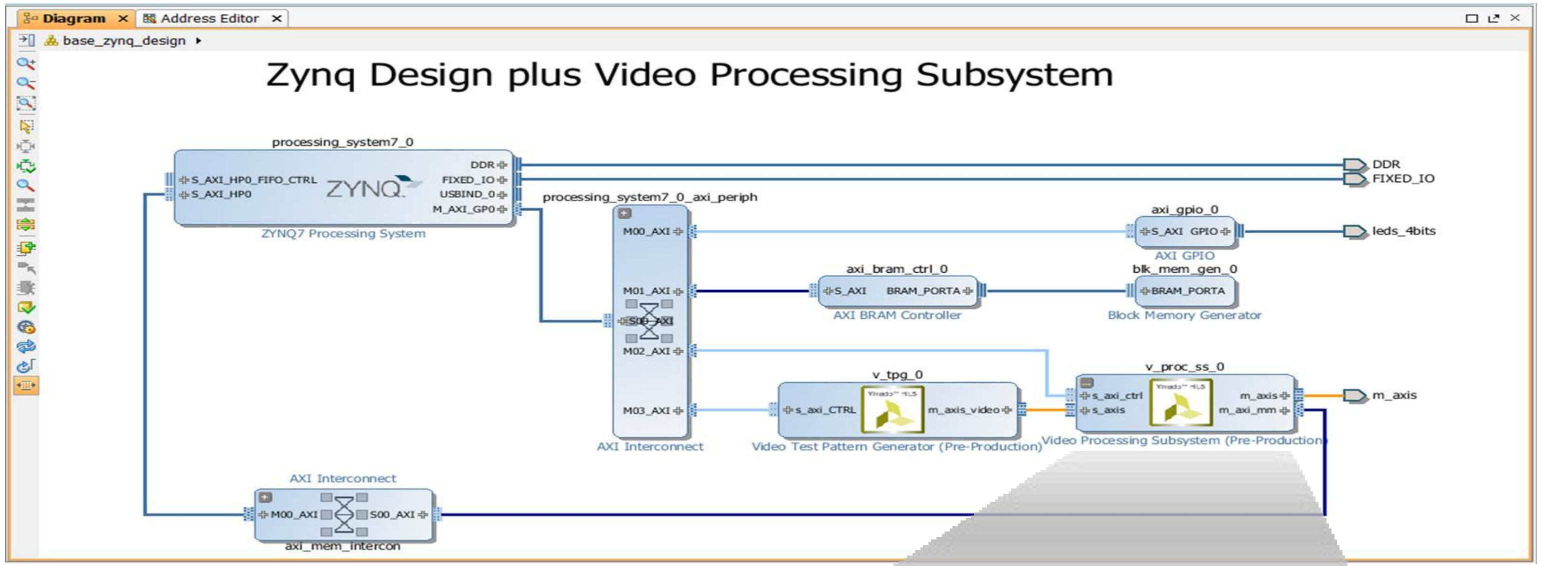
Control & Datapath Behavior

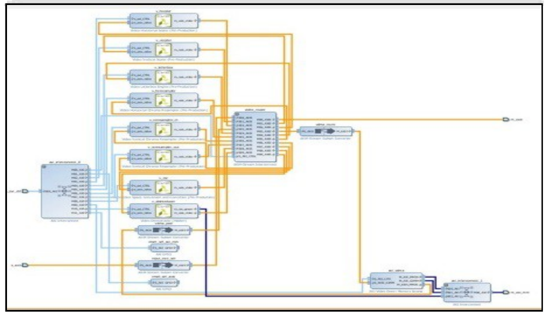


A unified control dataflow behavior is created.

XILINX ALL PROGRAMMABLE.

Hardware : IPI Automated IP Integration



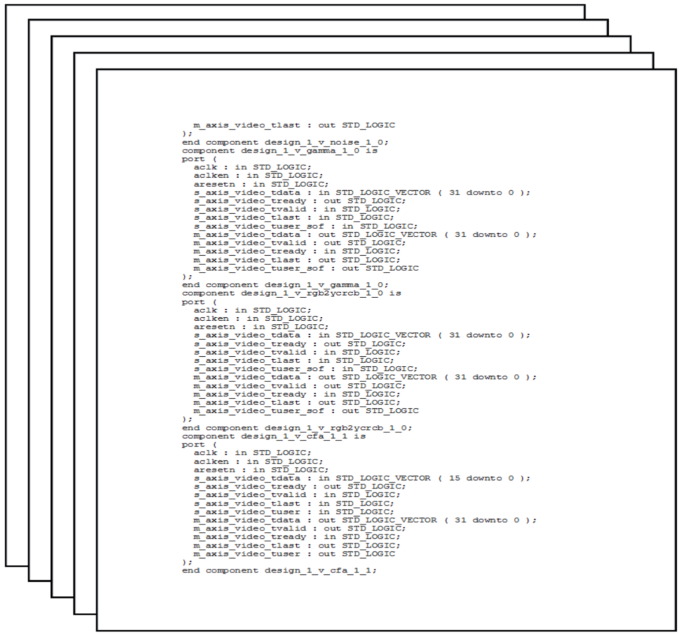


Video Processing IP Subsystem

IP Assembly Example:

Zynq Processor Subsystem
+ Video Subsystem
+ 6 IP Blocks

=

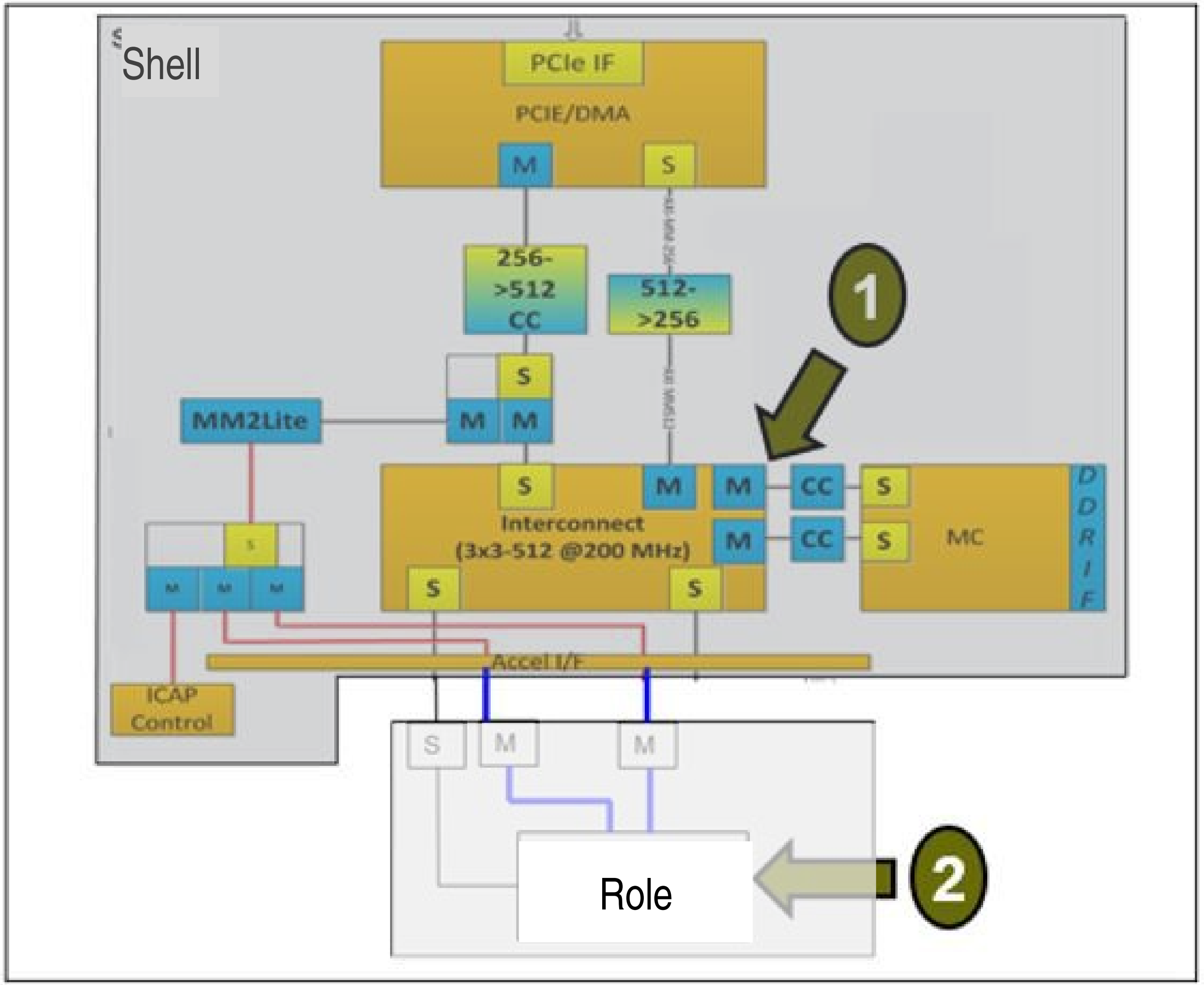


```
    m_axis_video_tlast : out STD_LOGIC
  );
  end component design_1_v_noise_1_0;
  component design_1_v_gamma_1_0 is
  port (
    aclk : in STD_LOGIC;
    aclken : in STD_LOGIC;
    aresetn : in STD_LOGIC;
    s_axis_video_tdata : in STD_LOGIC_VECTOR ( 31 downto 0 );
    s_axis_video_tready : out STD_LOGIC;
    s_axis_video_tvalid : in STD_LOGIC;
    s_axis_video_tlast : in STD_LOGIC;
    s_axis_video_tuser_sof : in STD_LOGIC;
    m_axis_video_tdata : out STD_LOGIC_VECTOR ( 31 downto 0 );
    m_axis_video_tvalid : out STD_LOGIC;
    m_axis_video_tready : in STD_LOGIC;
    m_axis_video_tlast : out STD_LOGIC;
    m_axis_video_tuser_sof : out STD_LOGIC
  );
  end component design_1_v_gamma_1_0;
  component design_1_v_rgb2ycrcb_1_0 is
  port (
    aclk : in STD_LOGIC;
    aclken : in STD_LOGIC;
    aresetn : in STD_LOGIC;
    s_axis_video_tdata : in STD_LOGIC_VECTOR ( 31 downto 0 );
    s_axis_video_tready : out STD_LOGIC;
    s_axis_video_tvalid : in STD_LOGIC;
    s_axis_video_tlast : in STD_LOGIC;
    s_axis_video_tuser_sof : in STD_LOGIC;
    m_axis_video_tdata : out STD_LOGIC_VECTOR ( 31 downto 0 );
    m_axis_video_tvalid : out STD_LOGIC;
    m_axis_video_tready : in STD_LOGIC;
    m_axis_video_tlast : out STD_LOGIC;
    m_axis_video_tuser_sof : out STD_LOGIC
  );
  end component design_1_v_rgb2ycrcb_1_0;
  component design_1_v_cfa_1_1 is
  port (
    aclk : in STD_LOGIC;
    aclken : in STD_LOGIC;
    aresetn : in STD_LOGIC;
    s_axis_video_tdata : in STD_LOGIC_VECTOR ( 15 downto 0 );
    s_axis_video_tready : out STD_LOGIC;
    s_axis_video_tvalid : in STD_LOGIC;
    s_axis_video_tlast : in STD_LOGIC;
    s_axis_video_tuser : in STD_LOGIC;
    m_axis_video_tdata : out STD_LOGIC_VECTOR ( 31 downto 0 );
    m_axis_video_tvalid : out STD_LOGIC;
    m_axis_video_tready : in STD_LOGIC;
    m_axis_video_tlast : out STD_LOGIC;
    m_axis_video_tuser : out STD_LOGIC
  );
  end component design_1_v_cfa_1_1;
```

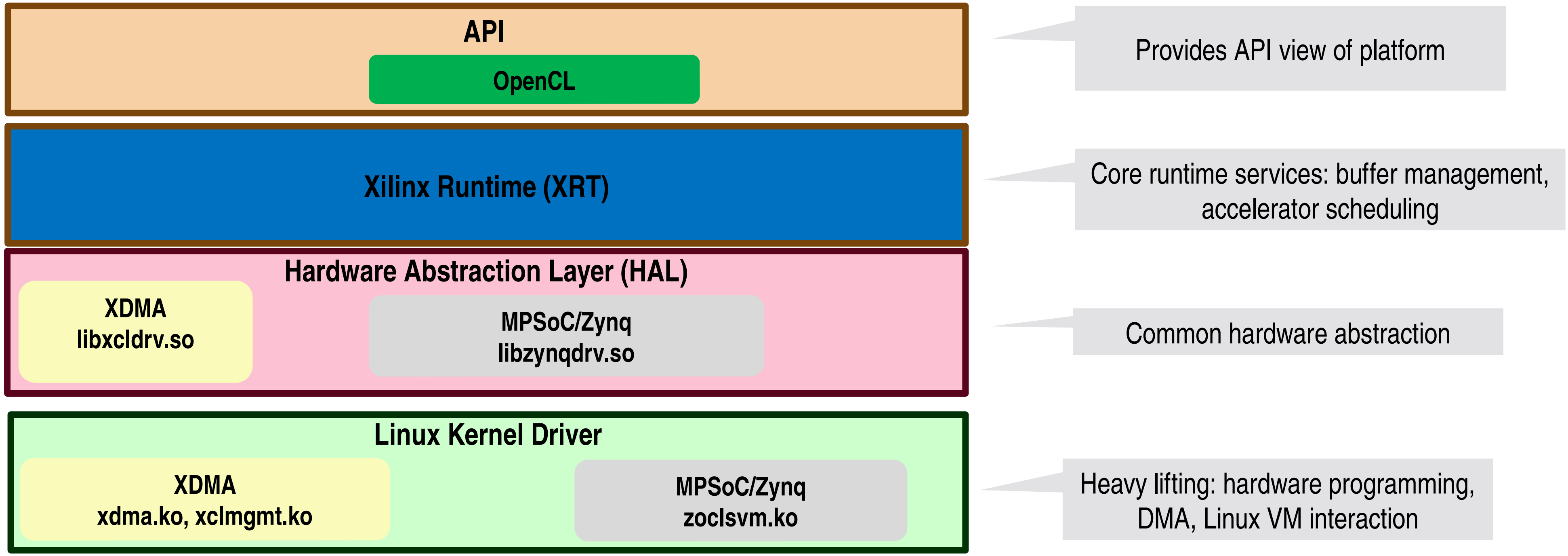
4700 lines of VHDL

(top-level connectivity only)

Creation of fixed Shell infrastructure



Hardware Abstraction : Runtime Layers



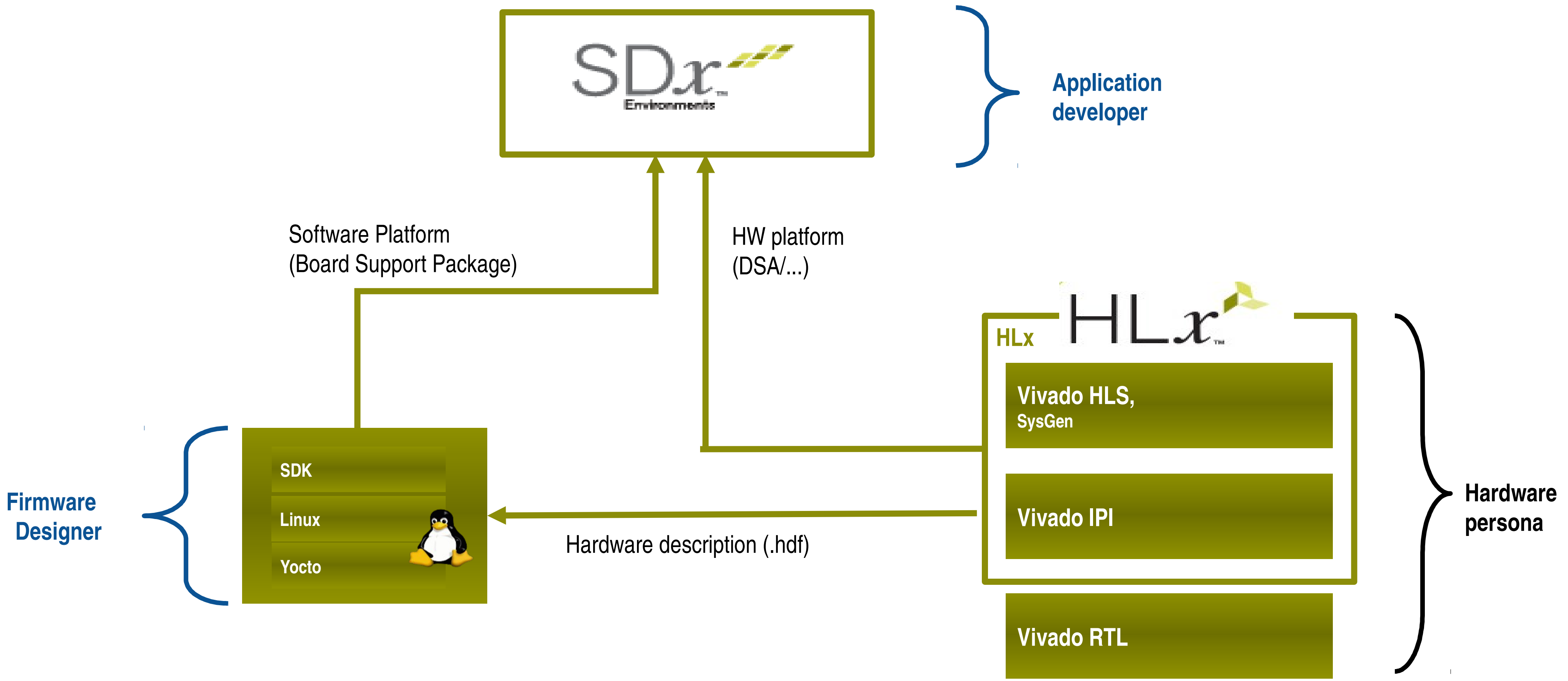
Provides API view of platform

Core runtime services: buffer management, accelerator scheduling

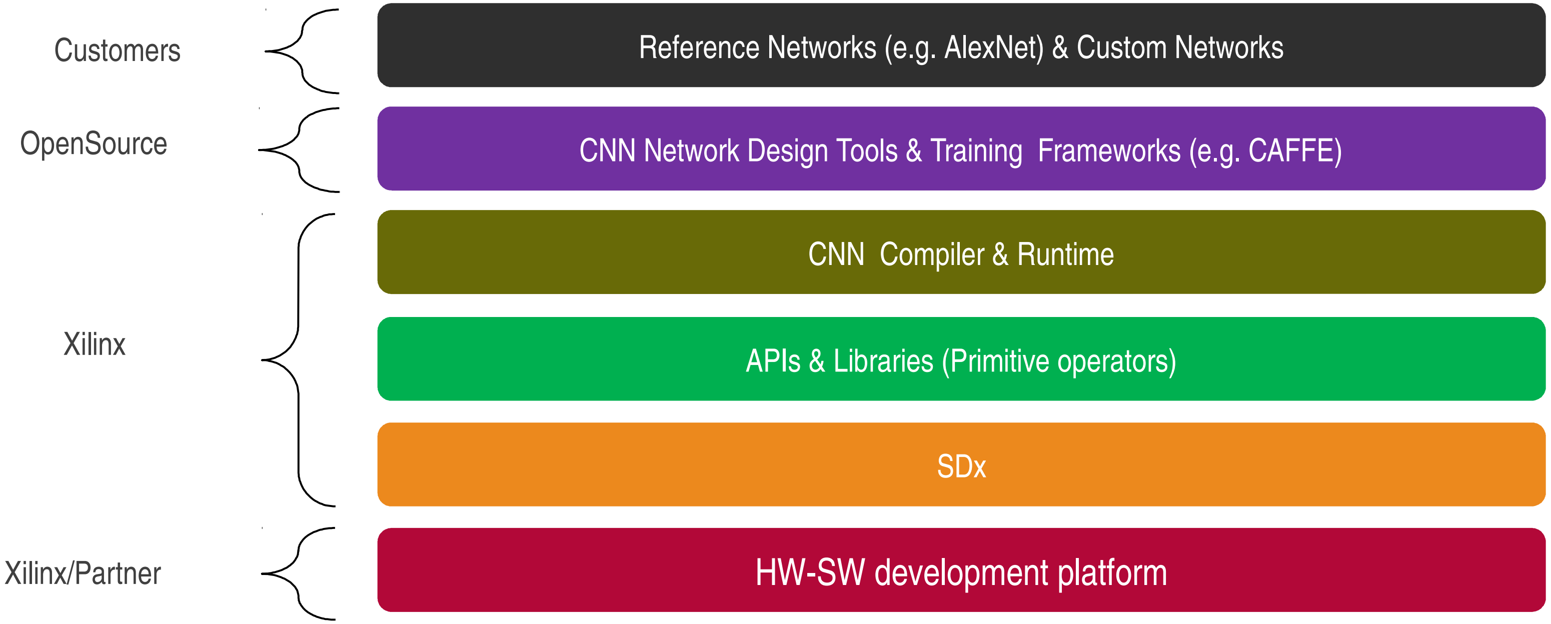
Common hardware abstraction

Heavy lifting: hardware programming, DMA, Linux VM interaction

Overall Platform and SDx Flow



Example : Xilinx Machine Learning Stack



Customers

Reference Networks (e.g. AlexNet) & Custom Networks

OpenSource

CNN Network Design Tools & Training Frameworks (e.g. CAFFE)

Xilinx

CNN Compiler & Runtime

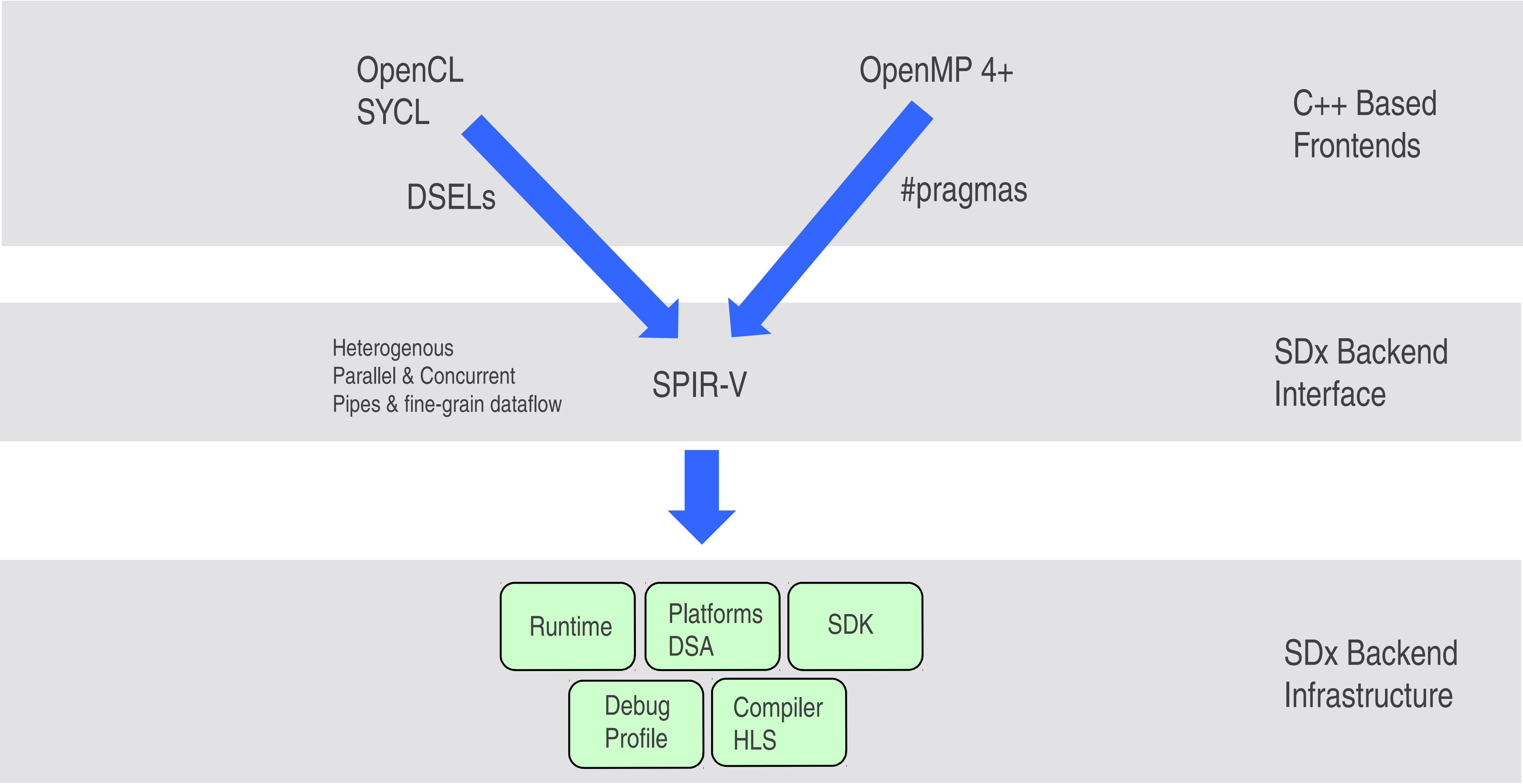
APIs & Libraries (Primitive operators)

SDx

Xilinx/Partner

HW-SW development platform

The Future : Single Source C++



OpenCL
SYCL

DSELs

OpenMP 4+

#pragmas

C++ Based
Frontends

Heterogenous
Parallel & Concurrent
Pipes & fine-grain dataflow

SPIR-V

SDx Backend
Interface

Runtime

Platforms
DSA

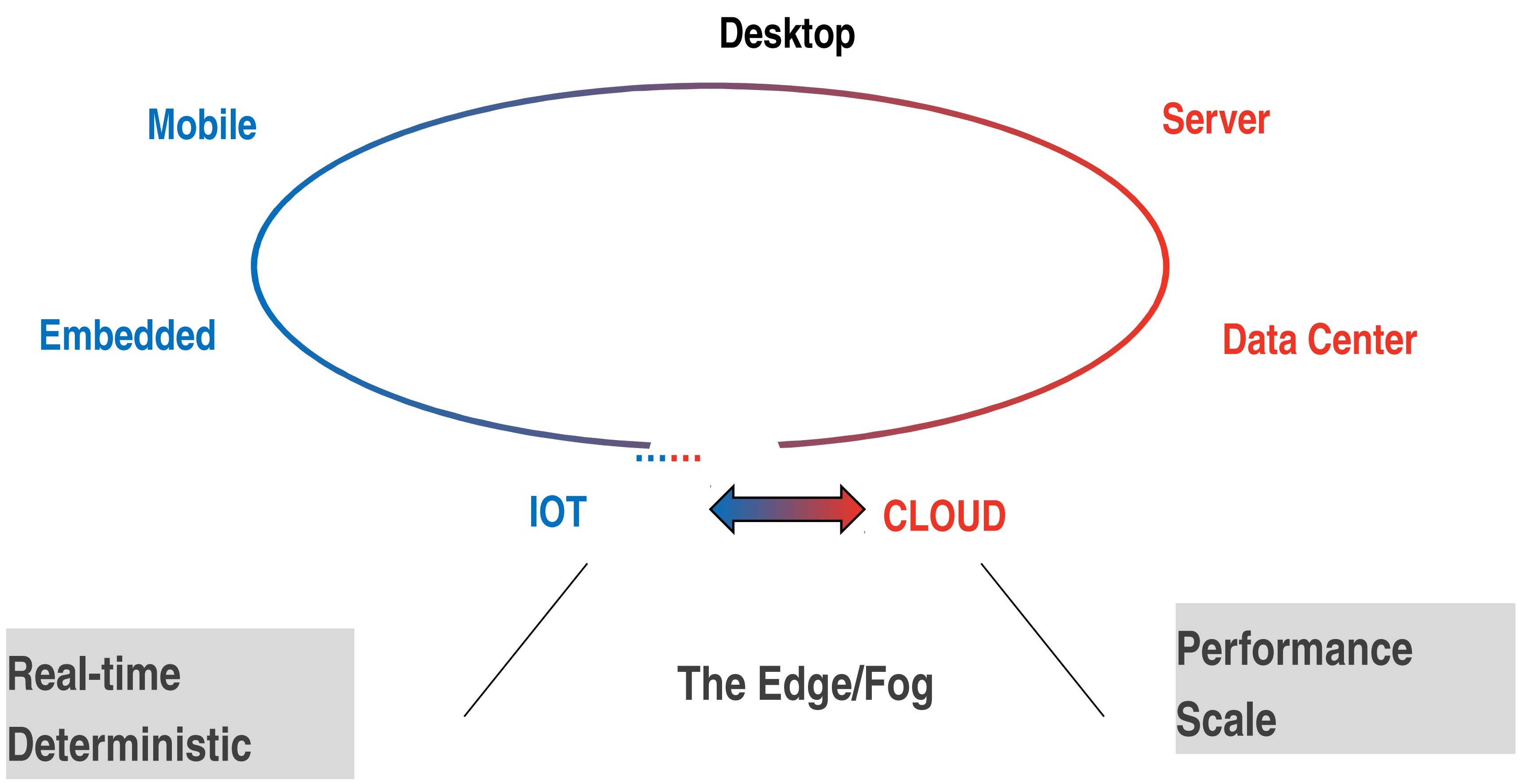
SDK

Debug
Profile

Compiler
HLS

SDx Backend
Infrastructure

From the Cloud to IOT



Desktop

Mobile

Server

Embedded

Data Center

IOT

CLOUD

Real-time

Deterministic

The Edge/Fog

Performance

Scale

Productivity languages and Efficiency languages

Application

Implementation

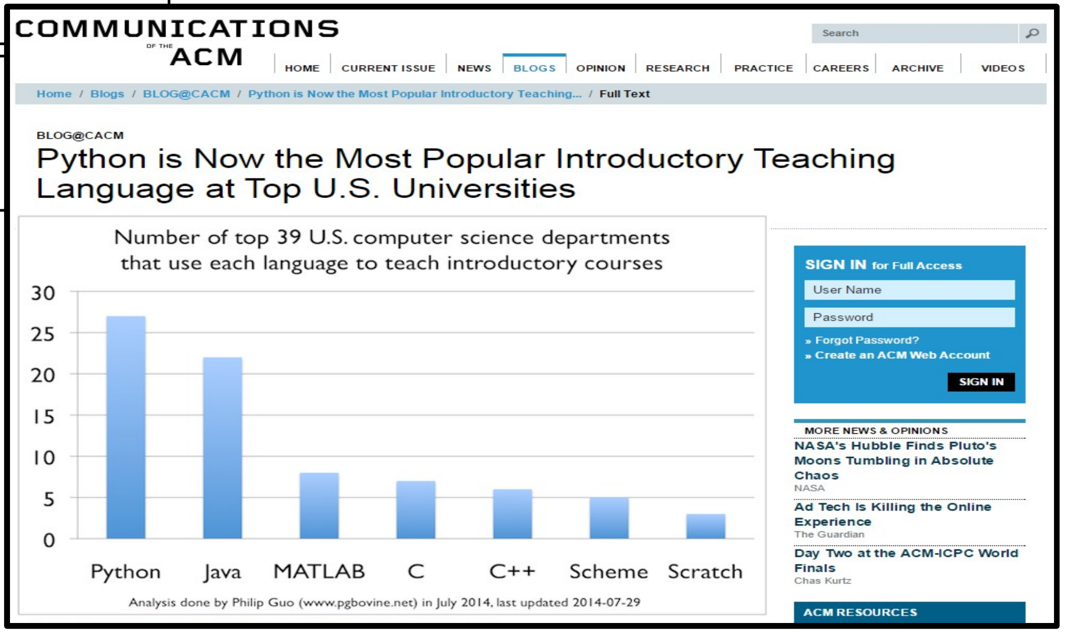
Applications,
Programming Frameworks

Productivity Languages
Python, Scala, ..

Efficiency Languages
C, C++, OpenCL

Hardware Systems
OS, hypervisor, drivers

Programming ZYNQ/MPSoC in a productivity language


COMMUNICATIONS OF THE ACM
Search
HOME
CURRENT ISSUE
NEWS
BLOGS
OPINION
RESEARCH
PRACTICE
CAREERS
ARCHIVE
VIDEOS
Home / Blogs / BLOG@CACM / Python is Now the Most Popular Introductory Teaching... / Full Text

BLOG@CACM

Python is Now the Most Popular Introductory Teaching Language at Top U.S. Universities

Number of top 39 U.S. computer science departments that use each language to teach introductory courses

Analysis done by Philip Guo (www.pgbovine.net) in July 2014, last updated 2014-07-29

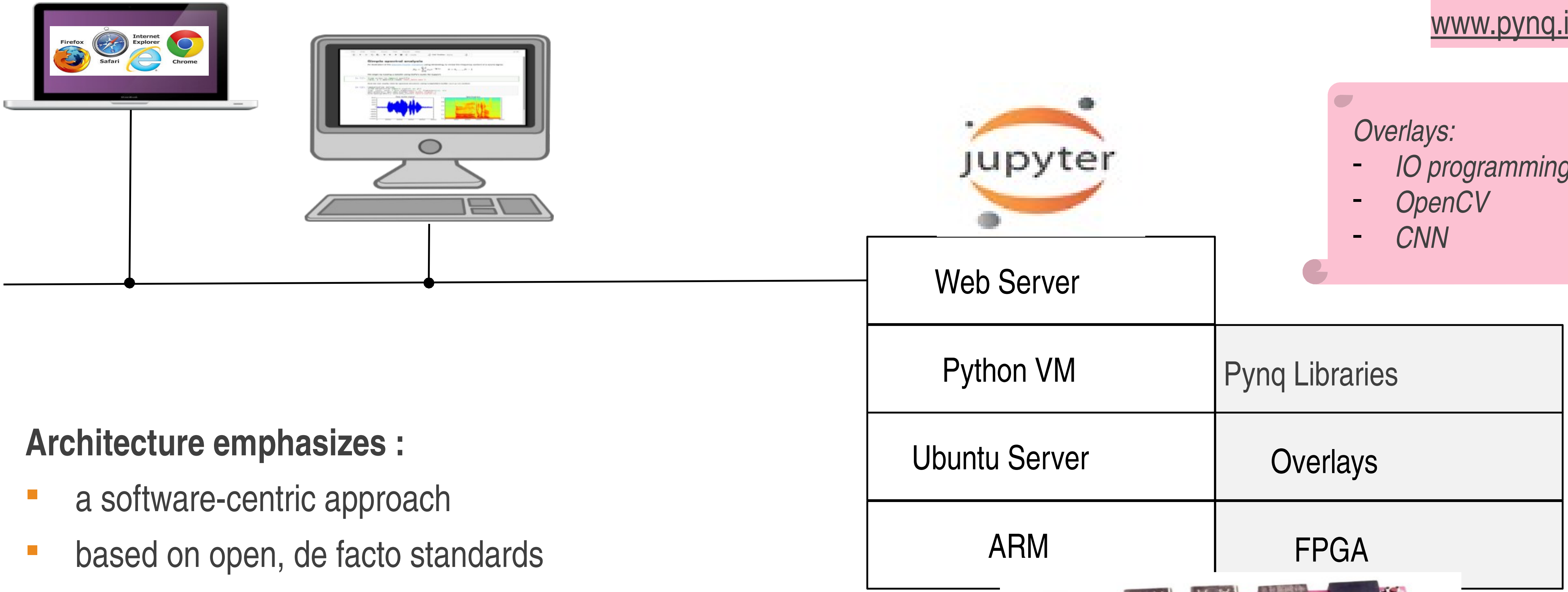
NASA's Hubble Finds Pluto's Moons Tumbling in Absolute Chaos
NASA

Ad Tech Is Killing the Online Experience
The Guardian

Day Two at the ACM-ICPC World Finals
Chas Kurtz

Python-based Open Source Platform : PYNQ

www.pynq.io



Overlays:

- *IO programming*
- *OpenCV*
- *CNN*

Architecture emphasizes :

- a software-centric approach
- based on open, de facto standards
- platform, OS and browser agnostic
- minimal learning curve
- no proprietary methodologies

SW running natively on Zynq

Summary

HW designers:

C-based IP development + high-level IP assembly

SW developers:

FPGA-based acceleration using SDx

XILINX

Committed to major investments in next generation silicon and tools that will revolutionize programming <u>All Programmable FPGA</u>