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

Ivo Bolsens
Traditional Compute Architectures Are Not Scalable

Requires Heterogeneous Architectures & Acceleration
Heterogeneous Domain Optimized Computing Platforms
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

© Copyright 2017 Xilinx
Data Center enabling the Cloud

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

- Network Acceleration
- NVF NICs
- SDN Controllers
- Compression

- Flash Arrays Acceleration
- SSDs Acceleration
- NVDIMM
- NVMe over Fabric
Unified HW to address compute, storage, and networking apps

- HPC
- Deep Learning Training

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

- Network Acceleration
- NFV NICs
- SDN Controllers
- Compression

- Flash Arrays Acceleration
- SSDs Acceleration
- NVDIMM
- NVMe over Fabric
Heterogeneous Compute Platforms in Datacenter

How Server Components Change with Machine Learning

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

Source: ARK Investment Management LLC
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

- Logic
- BRAM
- DSP

- Dual
  - A9
  - acc1
  - acc2

- Quad
  - A53
  - Dual
    - R5
    - GPU
    - H.265
  - acc1
  - acc2

- C/C++
- RTL
- VHDL

- C + VHDL

- C + High Level Synthesis

- OpenCL

- Bring power of C++ to OpenCL

- 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)
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

Memory allocation special? Yes
Data movement special? Yes
Accelerator memory access special? Yes
CCIX : Cache Coherent Interface for Accelerators

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

CCIX : Open Source

© Copyright 2017 Xilinx
CCIX: Accelerator – CPU Configurations
Domain Specific Platform Infrastructure

Networking Shell
- CPU
- Host Bridge
- EMAC
- DMA
- MEMC
- PPP
- PPP
- PPP

Video Shell
- CPU
- Host Bridge
- SDI
- AXI
- HDMI
- PR Region
- PR Region

Wireless Shell
- CPU
- Host Bridge
- DMA
- MEMC
- PR Region
- JESD In
- CPRI Out

Shell : Domain specific infrastructure (gray)
Role : ‘Donut holes’ in FPGA or on CPU executing programmable functions (white)

One Board
Multiple Shells
Use Model and Personas

- HW/PCB designer
  - CAD tools, PCB design tools
- HW designer
- Firmware designer
  - SDK
- Application developer

Silicon and boards

HW platform

SW platform

End application
Hardware : HLx – High-Level Design

UltraFast High-Level Design Methodology

C, C++ or SystemC

Vivado™ HLS

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)
Hardware: HLS Control & Datapath Synthesis

**Code**

```c
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;
}
```

**Operations**

- RDx
- RDc
- >=
- -
- ==
- +
- *
- +
- *
- WRy

**Control Behavior**

Finite State Machine (FSM) states

- 0
- 1
- 2

**Control & Datapath Behavior**

A unified control dataflow behavior is created.

From any C code example...

Operations are extracted...

The control is known...
Hardware : IPI Automated IP Integration

IP Assembly Example:
Zynq Processor Subsystem
+ Video Subsystem
+ 6 IP Blocks

Video Processing IP Subsystem

4700 lines of VHDL
(top-level connectivity only)
Creation of fixed Shell infrastructure
Hardware Abstraction : Runtime Layers

- **API**
  - OpenCL

- **Xilinx Runtime (XRT)**

- **Hardware Abstraction Layer (HAL)**
  - XDMA
    - libxcldrv.so
  - MPSoC/Zynq
    - libzynqdrv.so

- **Linux Kernel Driver**
  - XDMA
    - xdma.ko, xclmgmt.ko
  - MPSoC/Zynq
    - zoclsvm.ko

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

- **Application developer**
  - Vivado IPI
  - Vivado HLS, SysGen
  - Vivado RTL

- **SDK**
  - Linux
  - Yocto

- **Software Platform** (Board Support Package)

- **HW platform** (DSA/...)

- **Hardware description (.hdf)**

- **Firmware Designer**
  - SDK

- **Reference component**: HLx
Example: Xilinx Machine Learning Stack

- Reference Networks (e.g., AlexNet) & Custom Networks
- CNN Network Design Tools & Training Frameworks (e.g., CAFFE)
- CNN Compiler & Runtime
- APIs & Libraries (Primitive operators)
- SDx
- HW-SW development platform

Customers

OpenSource

Xilinx

Xilinx/Partner
The Future: Single Source C++

- OpenCL
- SYCL
- DSELS
- SPIR-V
- Heterogenous
  Parallel & Concurrent
  Pipes & fine-grain dataflow
- OpenMP 4+
- #pragmas
- C++ Based Frontends
- SDx Backend Interface
- Run-time
- Platforms DSA
- Debug Profile
- Compiler HLS
- SDK
- Debug
- DSA
- HLS
- SDK

SDx Backend Infrastructure
From the Cloud to IOT

Real-time
Deterministic

The Edge/Fog

Performance
Scale
### Productivity languages and Efficiency languages

<table>
<thead>
<tr>
<th>Application</th>
<th>Hardware Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applications, Programming Frameworks</td>
<td>OS, hypervisor, drivers</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Productivity Languages</th>
<th>Efficiency Languages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Python, Scala, ..</td>
<td>C, C++, OpenCL</td>
</tr>
</tbody>
</table>

#### Programming ZYNQ/MPSoC in a productivity language

### About Python

**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**

<table>
<thead>
<tr>
<th>Language</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Python</td>
<td>30</td>
</tr>
<tr>
<td>Java</td>
<td>25</td>
</tr>
<tr>
<td>MATLAB</td>
<td>15</td>
</tr>
<tr>
<td>C</td>
<td>10</td>
</tr>
<tr>
<td>C++</td>
<td>5</td>
</tr>
<tr>
<td>Scheme</td>
<td>2</td>
</tr>
<tr>
<td>Scratch</td>
<td>1</td>
</tr>
</tbody>
</table>

*Analytic done by Philip Greenspan (www.gzhimese.net) in July 2014, last updated 2014-07-30*
Architecture emphasizes:

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

Web Server

<table>
<thead>
<tr>
<th>Python VM</th>
<th>Pynq Libraries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ubuntu Server</td>
<td>Overlays</td>
</tr>
<tr>
<td>ARM</td>
<td>FPGA</td>
</tr>
</tbody>
</table>

SW running natively on Zynq

Overlays:
- IO programming
- OpenCV
- CNN
Summary

HW designers:
C-based IP development + high-level IP assembly

SW developers:
FPGA-based acceleration using SDx

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