From PAMELA to EuroEXA and RAIN in Manchester

Mikel Luján

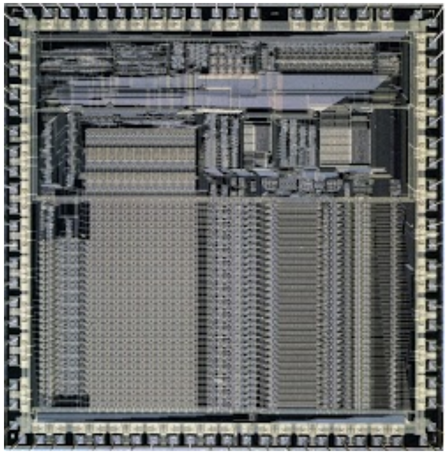
Department of Computer Science

Jornadas Sarteco, 18-20 September 2019

Taming Heterogenous System-on-Chips

1. How to program them?
   - TornadoVM
     - https://github.com/beehive-lab/TornadoVM
     - VEE'17, ManLang'18, VEE'19
2. How to design, simulate, prototype, … them?
   - SLAMBench,
     - https://github.com/pamela-project/slambench2
     - ICRA'19, ICRA'18, PACT'16, ICRA'15
   - MAMBO & SimAcc
     - https://github.com/beehive-lab/mambo
     - FPL'14, TACO 2016, FCCM'17, FCCM'19
   - RAIN Hub, EuroEXA

System on Chip



ARM1

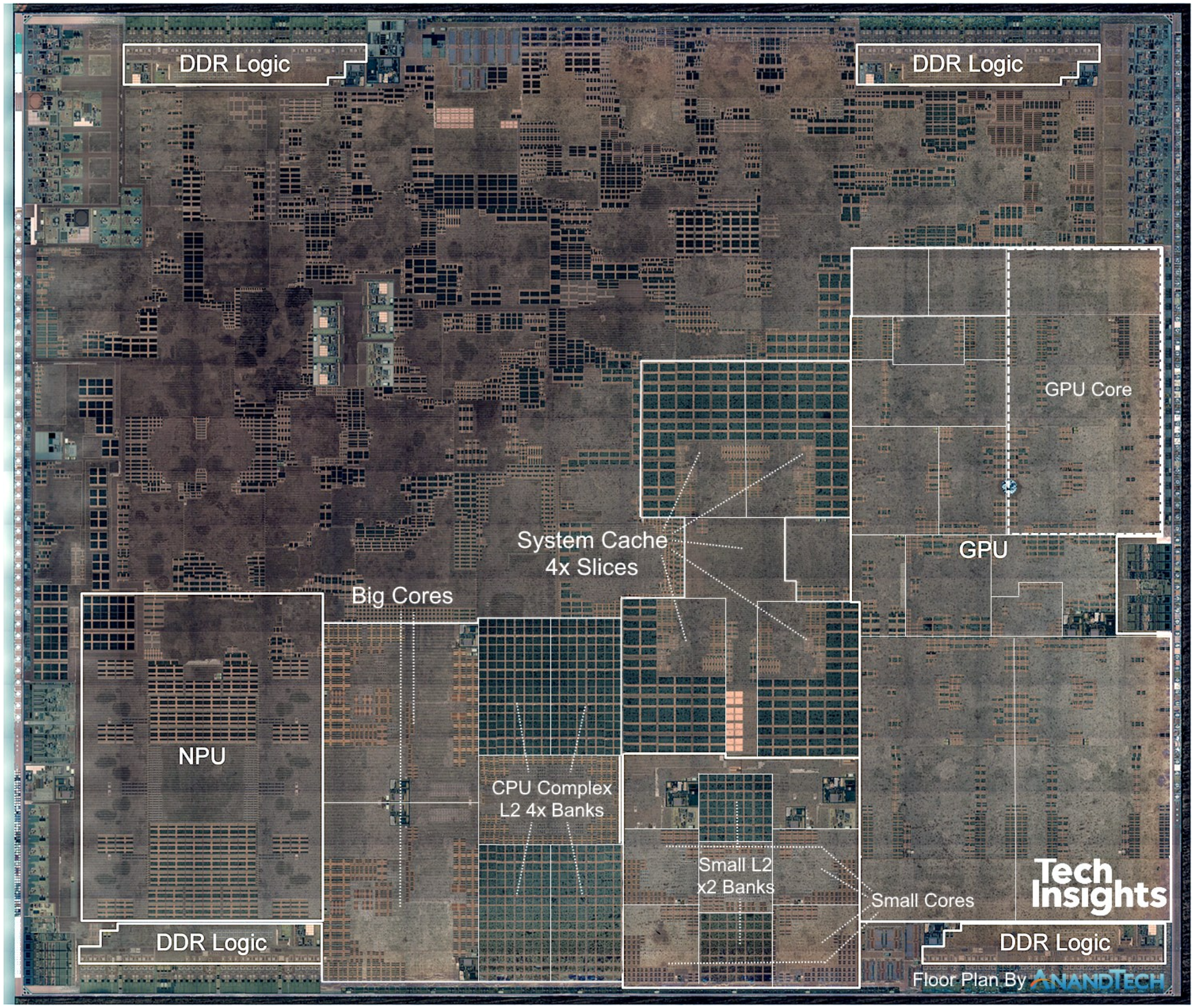
25K transistors

50mm² 3 micron

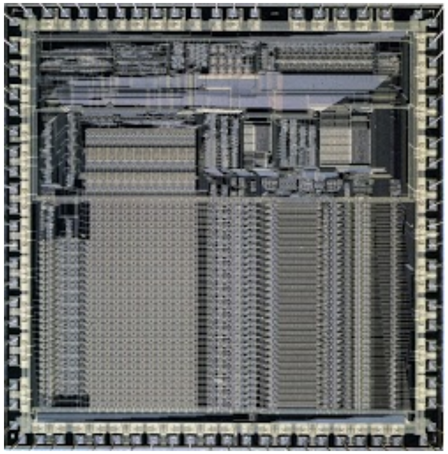
Apple A12 – 2018

6,9 Billion transistors

83mm² TSMC 7nm



Low Power System on Chips

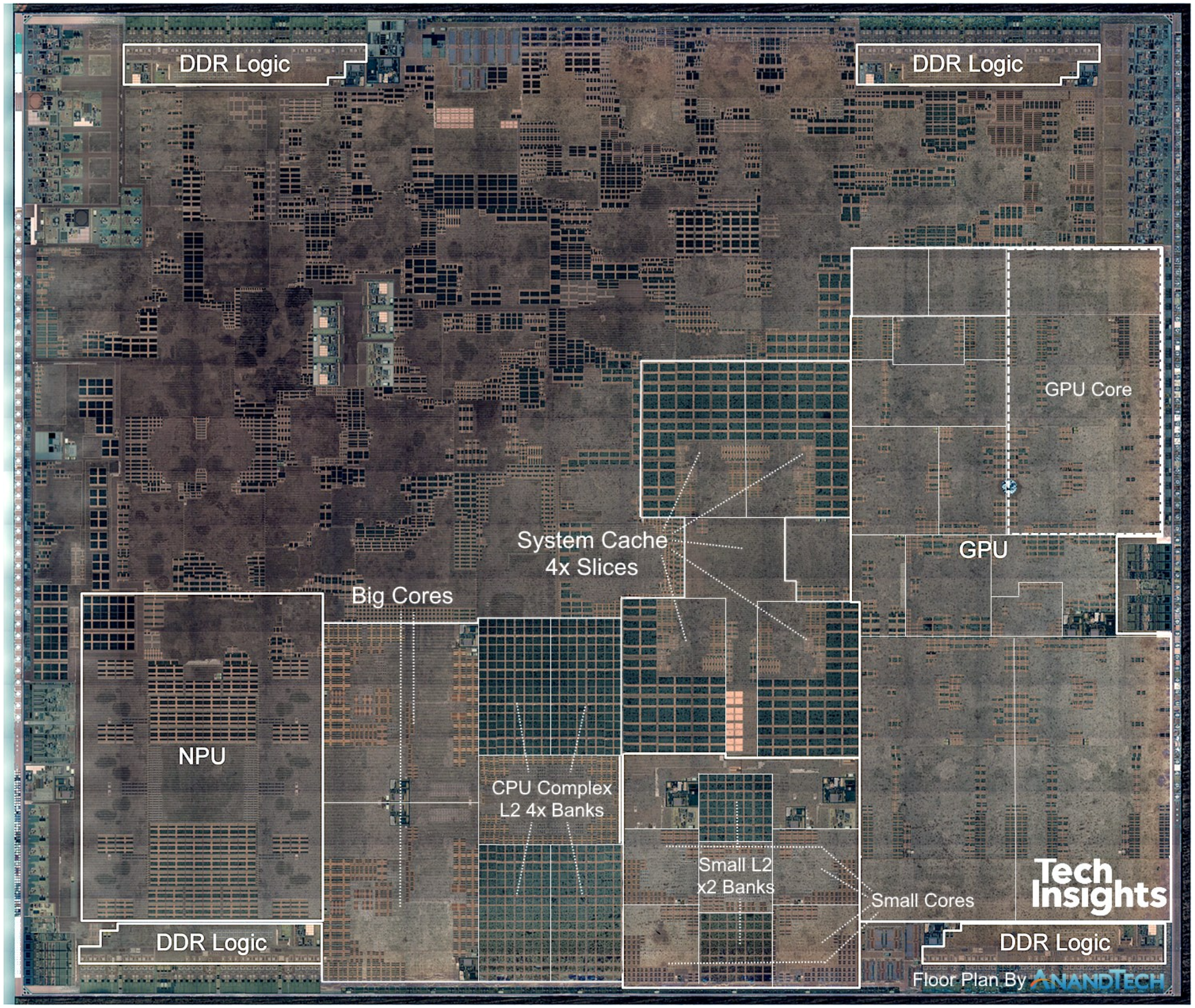


ARM1

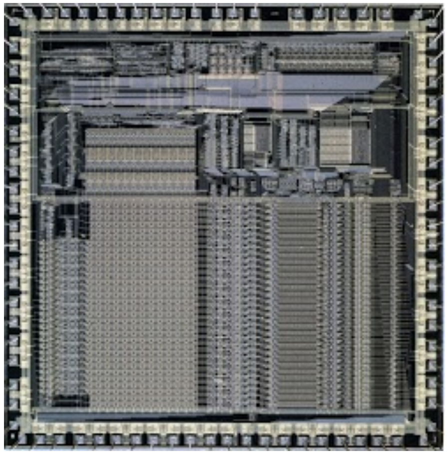
0.1 Watt

Apple A12 – 2018

Max 3 Watts

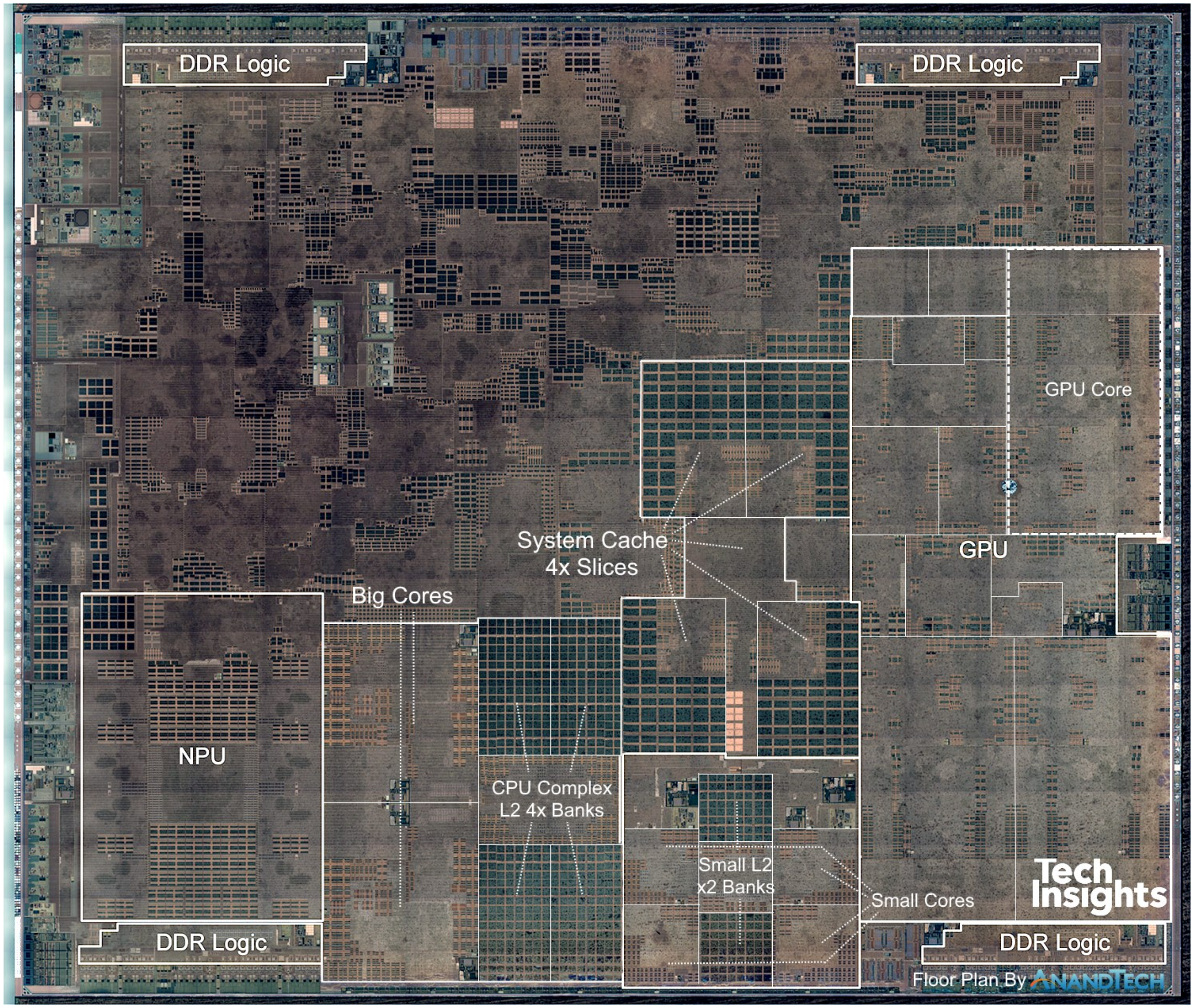


Heterogenous System on Chips



Apple A12 – 2018

- Big/Little cores
- GPU
  - graphics accelerator
- NPU
  - Deep Neural Network accelerator
  - 5,000 billion ops/sec



A12 SoC NPU accelerator - 5,000 billion ops/sec

- 5 Tera ops/sec

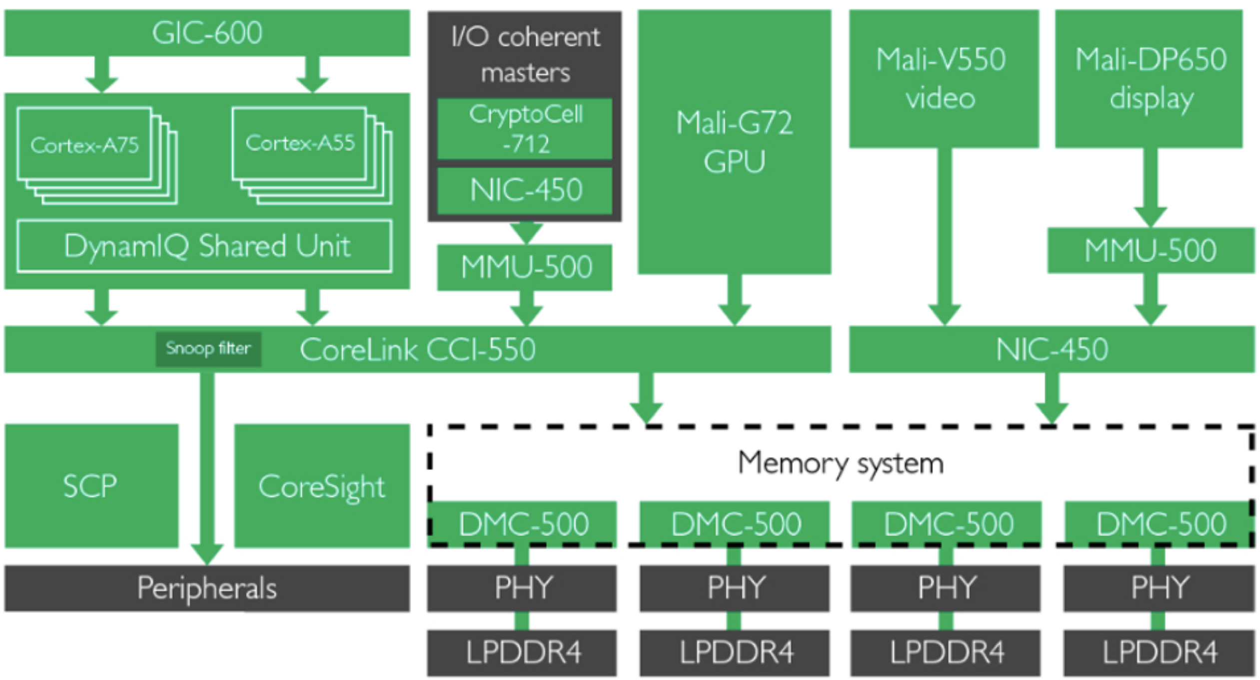
The Milky Way contains between 200 and **400 billion** stars and at least **100 billion** planets. The exact figure depends on the number of very-low-mass stars, which are hard to detect, especially at distances of more than 300 ly (90 pc) from the Sun.

Milky Way - Wikipedia
https://en.wikipedia.org/wiki/Milky_Way

Heterogenous System on Chips

Challenge Statement

Programming heterogenous SoCs with hardware accelerators is
- highly human labour intensive,
- not portable, and
- lacks appropriate tools



Block diagram with Arm IP

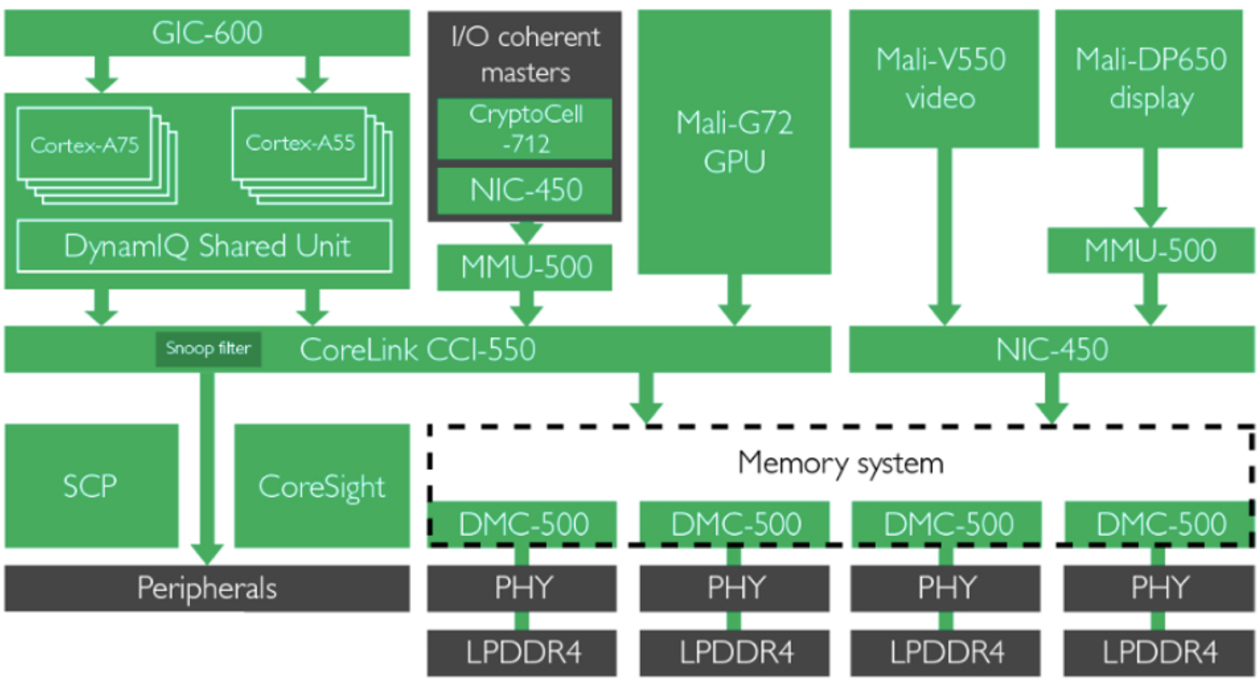
Example Arm SoC for mobile phone

Heterogenous System on Chips

Challenge Statement

Designing heterogenous SoCs with hardware accelerators is

- highly human labour intensive,
- restricted by simulation capabilities, and
- lacks appropriate integrated methodologies and tools



Block diagram with Arm IP

Example Arm SoC for mobile phone

Taming Heterogenous System-on-Chips

1. How to program them?
   - TornadoVM
     - https://github.com/beehive-lab/TornadoVM
     - VEE'17, ManLang'18, VEE'19
2. How to design, simulate, prototype, … them?
   - SLAMBench,
     - https://github.com/pamela-project/slambench2
     - ICRA'19, ICRA'18, PACT'16, ICRA'15
   - MAMBO & SimAcc
     - https://github.com/beehive-lab/mambo
     - FPL'14, TACO 2016, FCCM'17, FCCM'19
   - RAIN Hub, EuroEXA

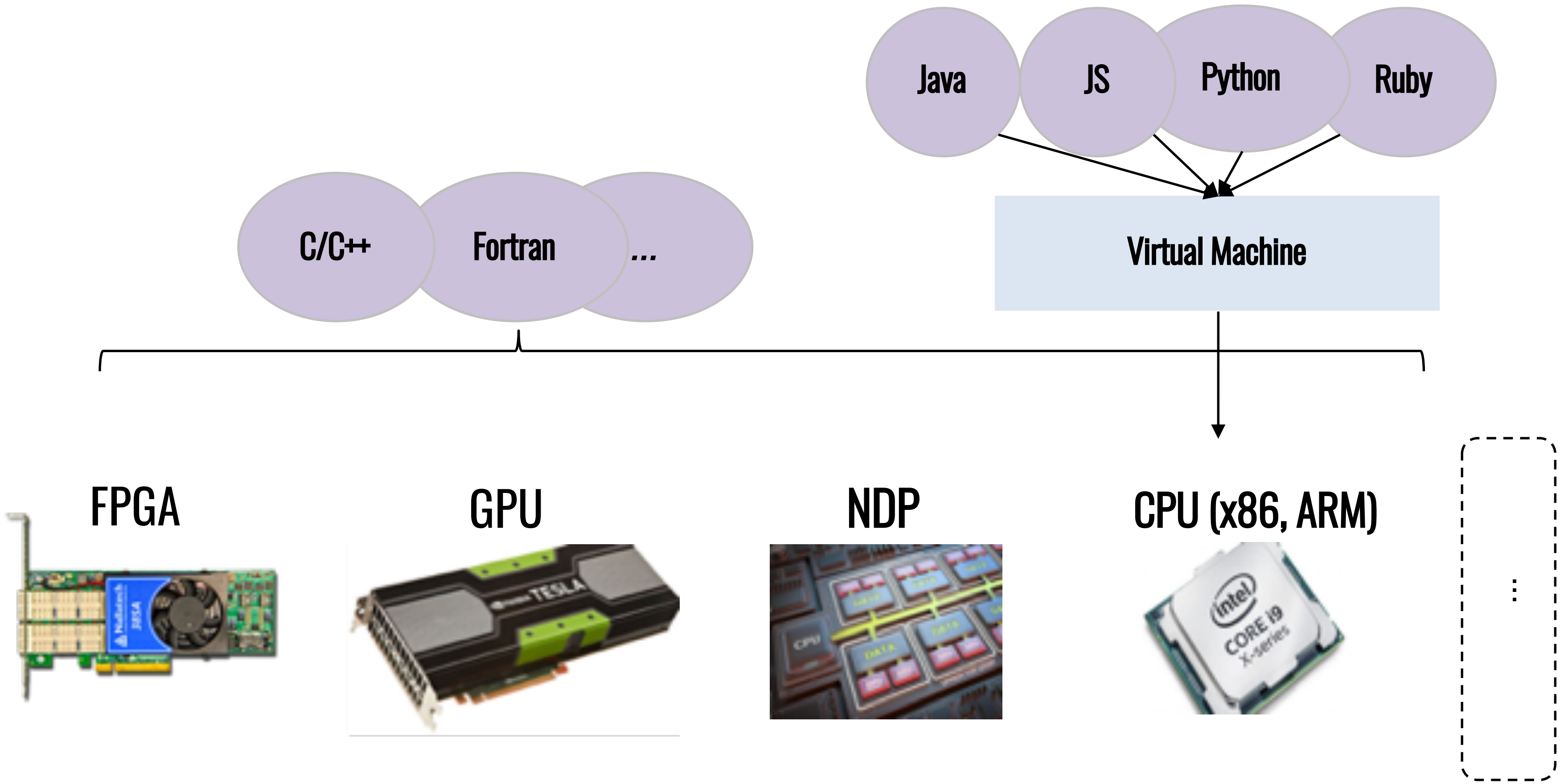
PAMELA

A Panoramic View of the Many-core Landscape

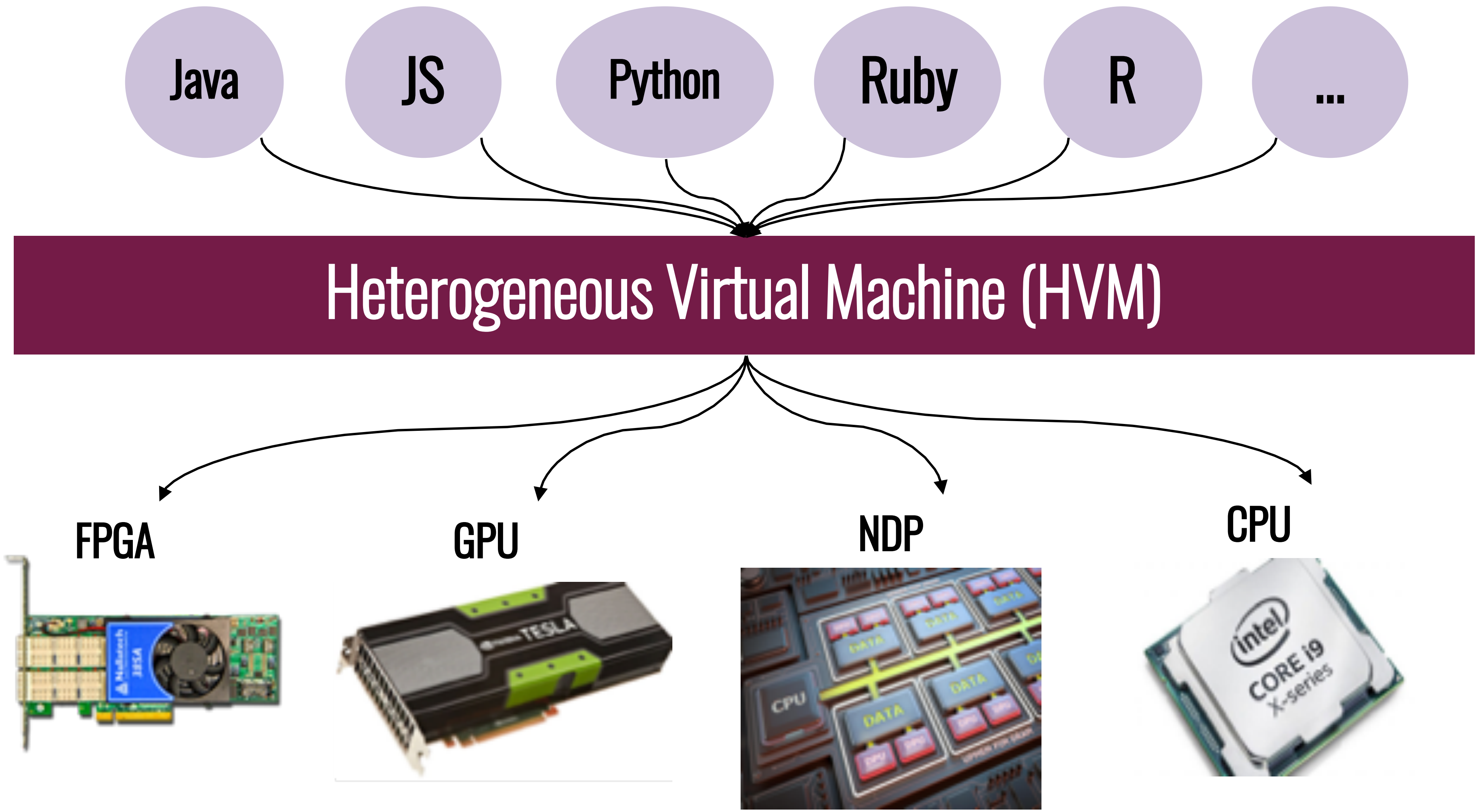
Bring together application developers, system software researchers and computer architects to develop novel approaches that can adapt with and drive heterogeneous, many-core architectural evolution.

3D scene understanding, as a driver to bring together the project. An important application in next generation smart-phones and future intelligent devices.

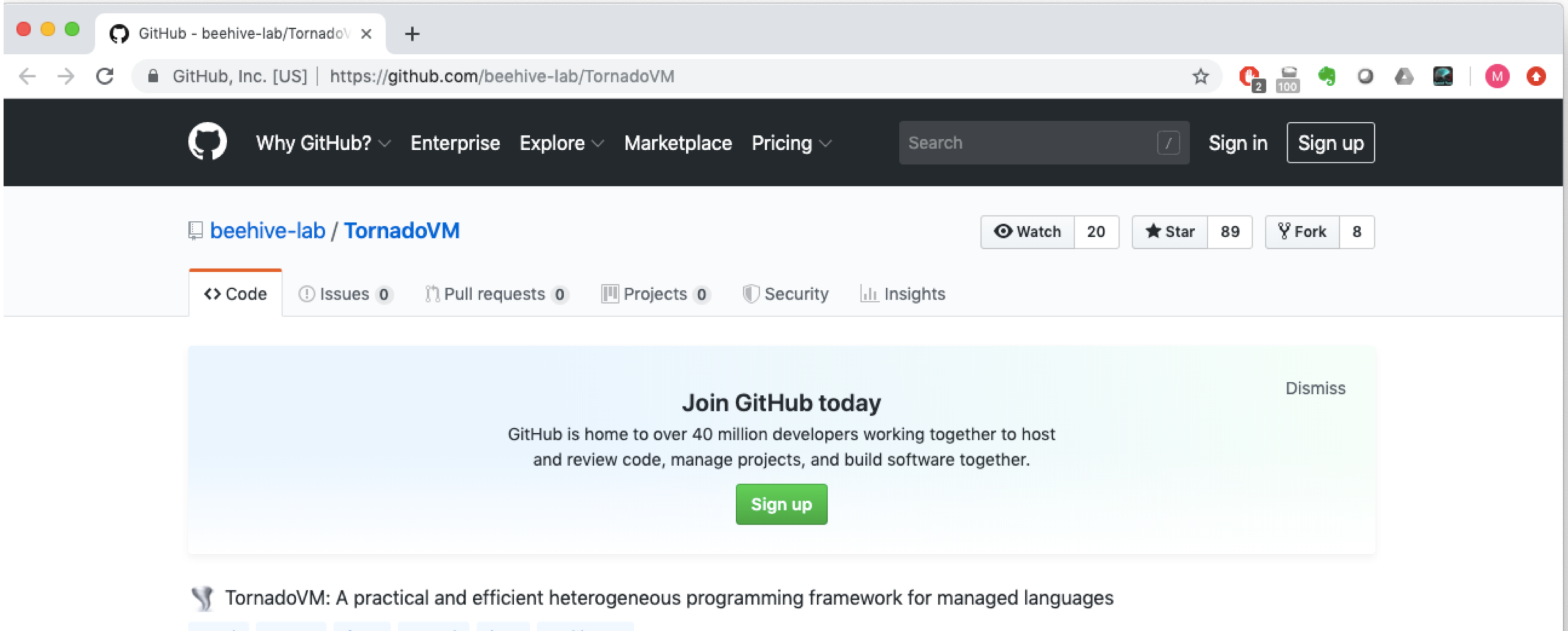
TornadoVM - Practical HW acceleration of Managed Languages



Ideal Scenario for Managed Languages



TornadoVM


GitHub - beehive-lab/TornadoV
GitHub, Inc. [US] | https://github.com/beehive-lab/TornadoVM
Why GitHub?
Enterprise
Explore
Marketplace
Pricing
Search
Sign in
Sign up
beehive-lab / TornadoVM
Watch 20
Star 89
Fork 8
Code
Issues 0
Pull requests 0
Projects 0
Security
Insights
Dismiss
Join GitHub today
GitHub is home to over 40 million developers working together to host and review code, manage projects, and build software together.
Sign up
TornadoVM: A practical and efficient heterogeneous programming framework for managed languages

TornadoVM – Design principles

- Simplicity
  - Vanilla Java code, Legacy, Libraries
- Extensibility
  - Coverage of wide range of devices
- Performance
  - General and device-specific optimizations
- Dynamism
  - Automatically discover “best” device

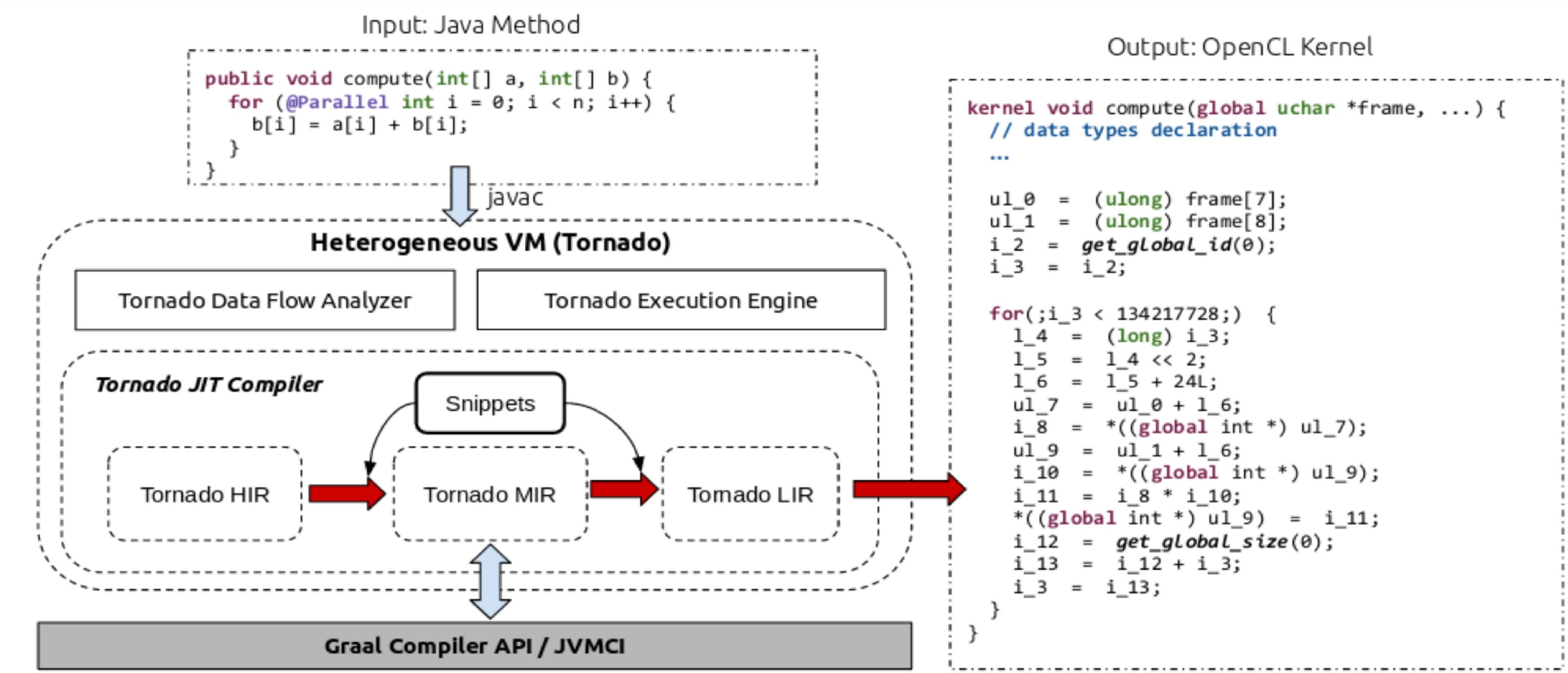
TornadoVM API

```java
public static void add(int[] a, int[] b, int[] c) {
    for (@Parallel int i = 0; i < c.length; i++) {
        c[i] = a[i] + b[i];
    }
}

// execute array addition on an accelerator
int[] a = new int[100];
int[] b = new int[100];
int[] c = new int[100];


new TaskSchedule("s0")
    .task("t0", ArrayAddInt::add, a, b, c)
    .streamOut(c)
    .execute();
```

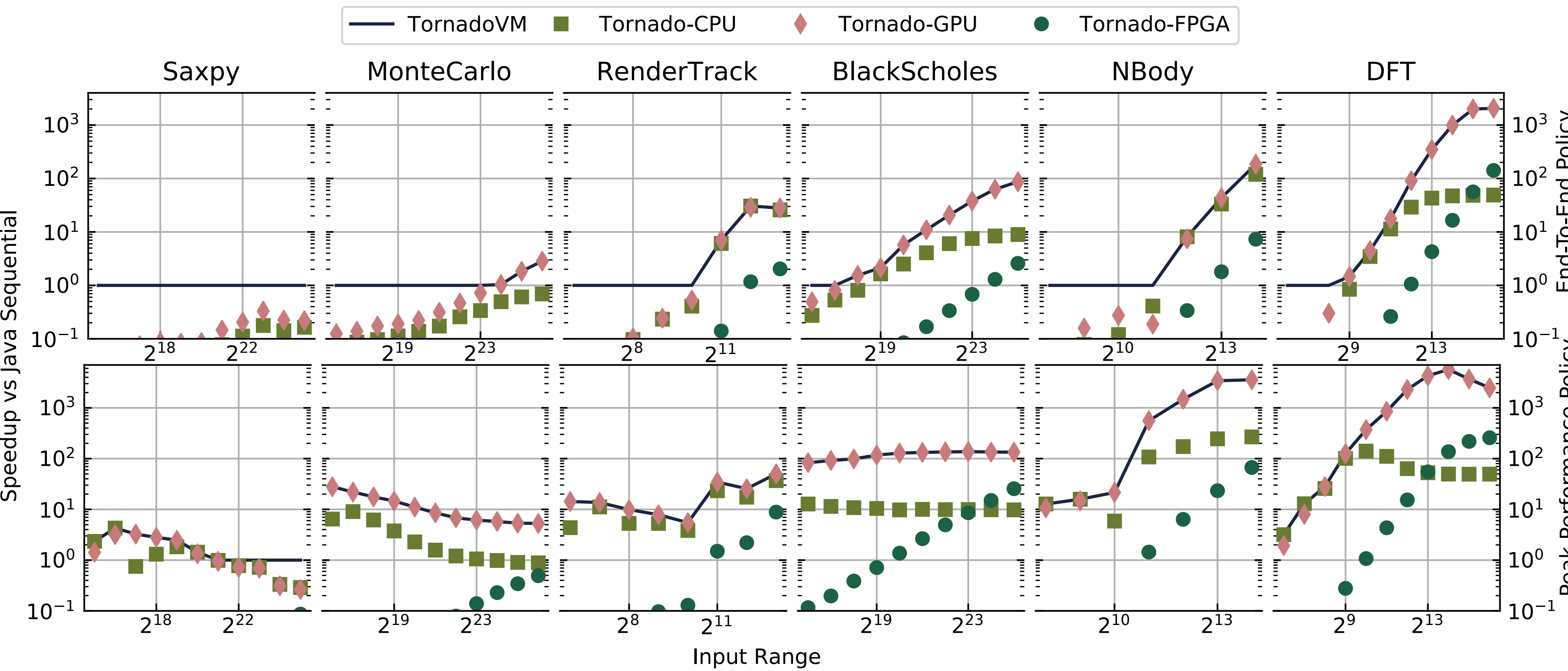
Auto-parallelization directive

Task-based execution model

TornadoVM - JIT Compiler and Runtime



Evaluation – VEE'19



Taming Heterogenous System-on-Chips

1. How to program them?
   - TornadoVM
     - https://github.com/beehive-lab/TornadoVM
     - VEE’17, ManLang’18, VEE’19
2. How to design, simulate, prototype, … them?
   - SLAMBench,
     - https://github.com/pamela-project/slambench2
     - ICRA’19, ICRA’18, PACT’16, ICRA’15
   - MAMBO & SimAcc
     - https://github.com/beehive-lab/mambo
     - FPL’14, TACO 2016, FCCM’17, FCCM’19
   - RAIN Hub, EuroEXA

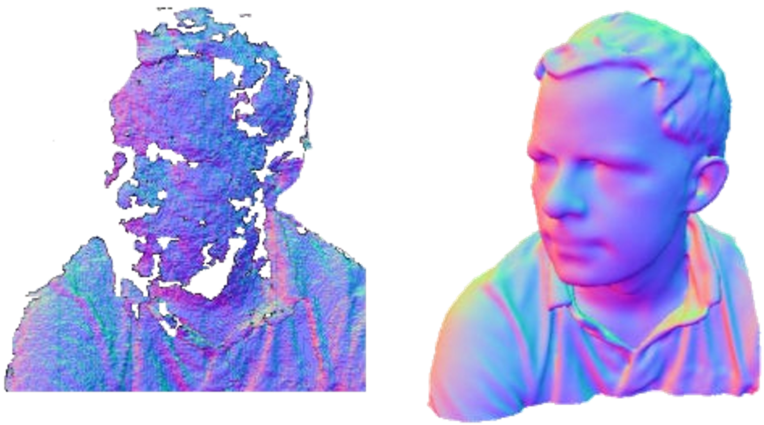
SLAMBench 1, 2, 3 …

HOW DID IT START, WHERE DOES IT GO ?

The University of Manchester

MANCHESTER 1824

Why did it start?



- **KinectFusion [Newcombe 2011]**
  - Efficient implementation with GPU (2011)
  - But not applicable on Mobile

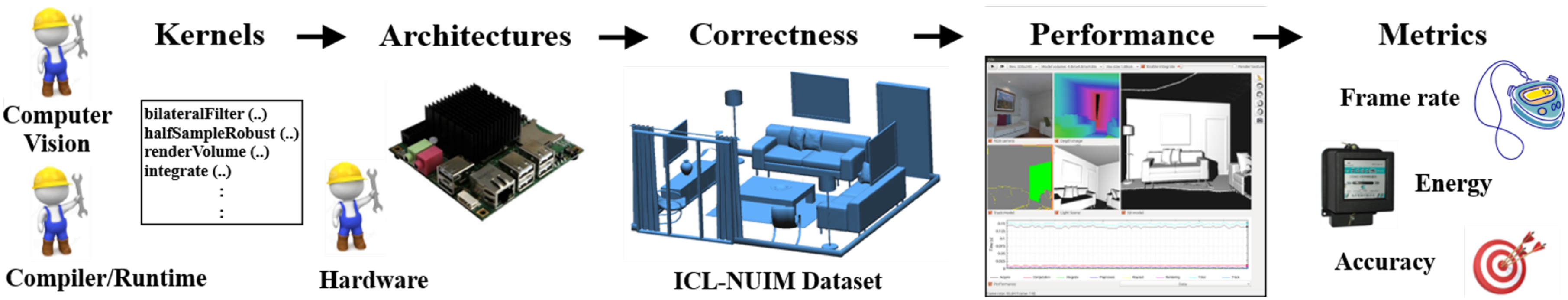
The future of SLAM is embedded,

can we make this happen ?

Experts from different fields

- Feedback: "we cannot optimise KinectFusion", reproducibility issue!
- True benefit of interdisciplinary collaboration in action
- **How can we modify a SLAM system without breaking it?**

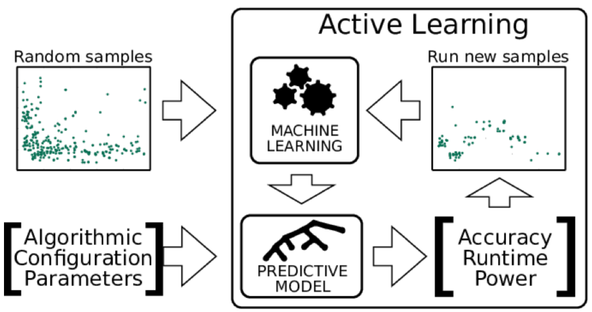
SLAMBench 1

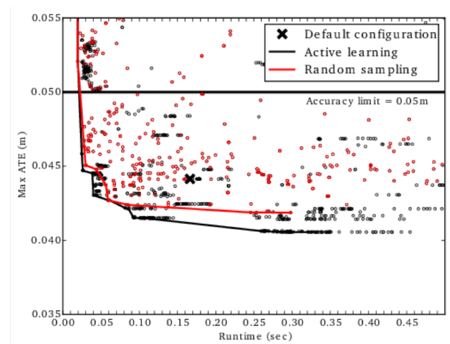


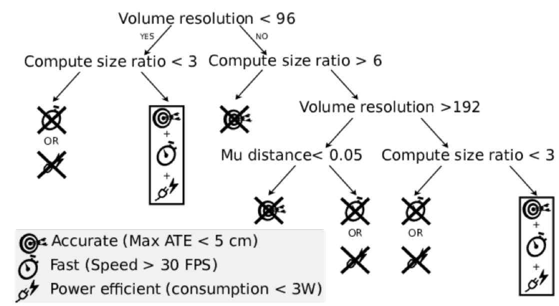
- Multiple implementations of KinectFusion (C++, OpenMP, OpenCL, CUDA)
- Three metrics (Speed, power, Accuracy)
- Scripts to run the system, collect data, and analyse performance.
- Tested on multiple O.S. (Android, Linux, Windows, MacOS)

[ICRA'15] Introducing SLAMBench, a performance and accuracy benchmarking methodology for SLAM,
Luigi Nardi, Bruno Bodin, M Zeeshan Zia, John Mawer, Andy Nisbet, Paul HJ Kelly, Andrew J Davison, Mikel Luján, Michael FP O'Boyle, Graham Riley, Nigel Topham, Steve Furber

PACT 2016 – Kinnect fusion on low power SoC







Learning Best configurations Knowledge

[PACT'16] Integrating algorithmic parameters into benchmarking and design space exploration in 3D scene understanding
Bruno Bodin, Luigi Nardi, M Zeeshan Zia, Harry Wagstaff, Govind Sreekar Shenoy, Murali Emani, John Mawer, Christos Kotselidis, Andy Nisbet, Mikel Lujan, Björn Franke, Paul HJ Kelly, Michael O'Boyle

What Robotics People Need?

Comparing easily SLAM systems to pick the right one!

A SLAM system that works!

Can We Compare SLAM systems **easily**?

- **Algorithms** (Dense, Sparse, Direct, Indirect...)
- **Sensors** (Monocular, Stereo, Depth, Event-based...)
- **Environments** (Street, Forest, in-door...)
- **Computation hardware** (Desktop, Cloud, Mobile phones...)
- **Performance metrics** (Speed, Accuracy, Power, Memory...)

LONG AND PAINFUL

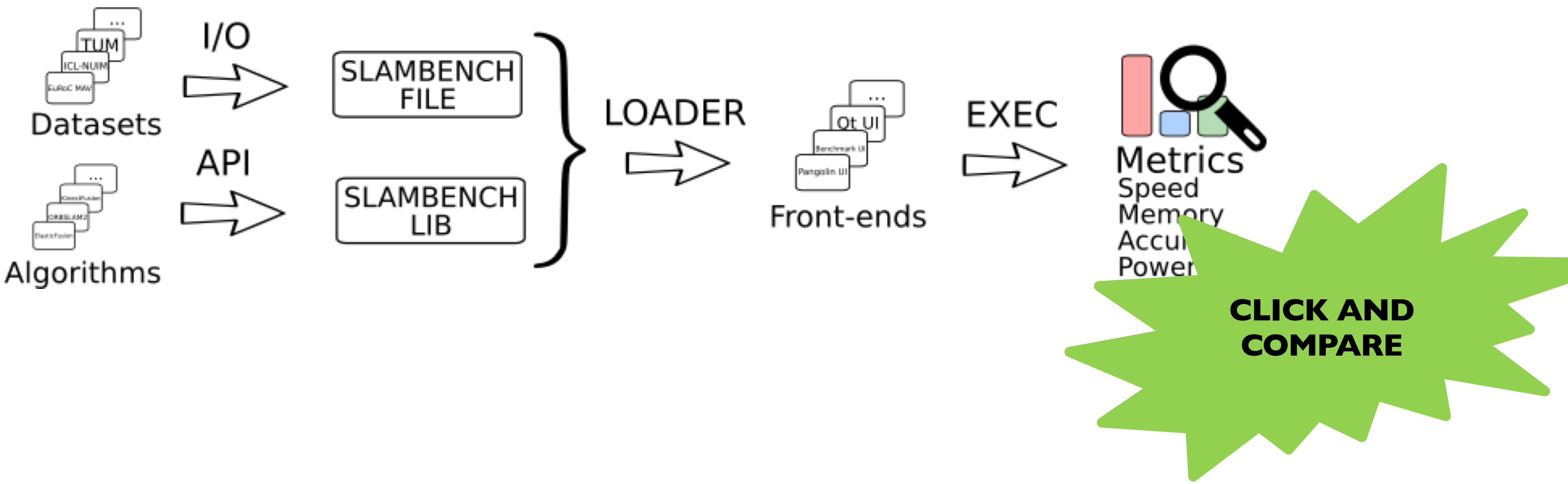
What is required to evaluate and compare SLAM systems?

ALGORITHM-AGNOSTIC: DEFINE SLAM INPUTS AND OUTPUTS.

DATASET-AGNOSTIC: SUPPORT EXISTING AND FUTURE SENSORS.

CONTEXT-AGNOSTIC: FAIRLY EVALUATE PERFORMANCE.

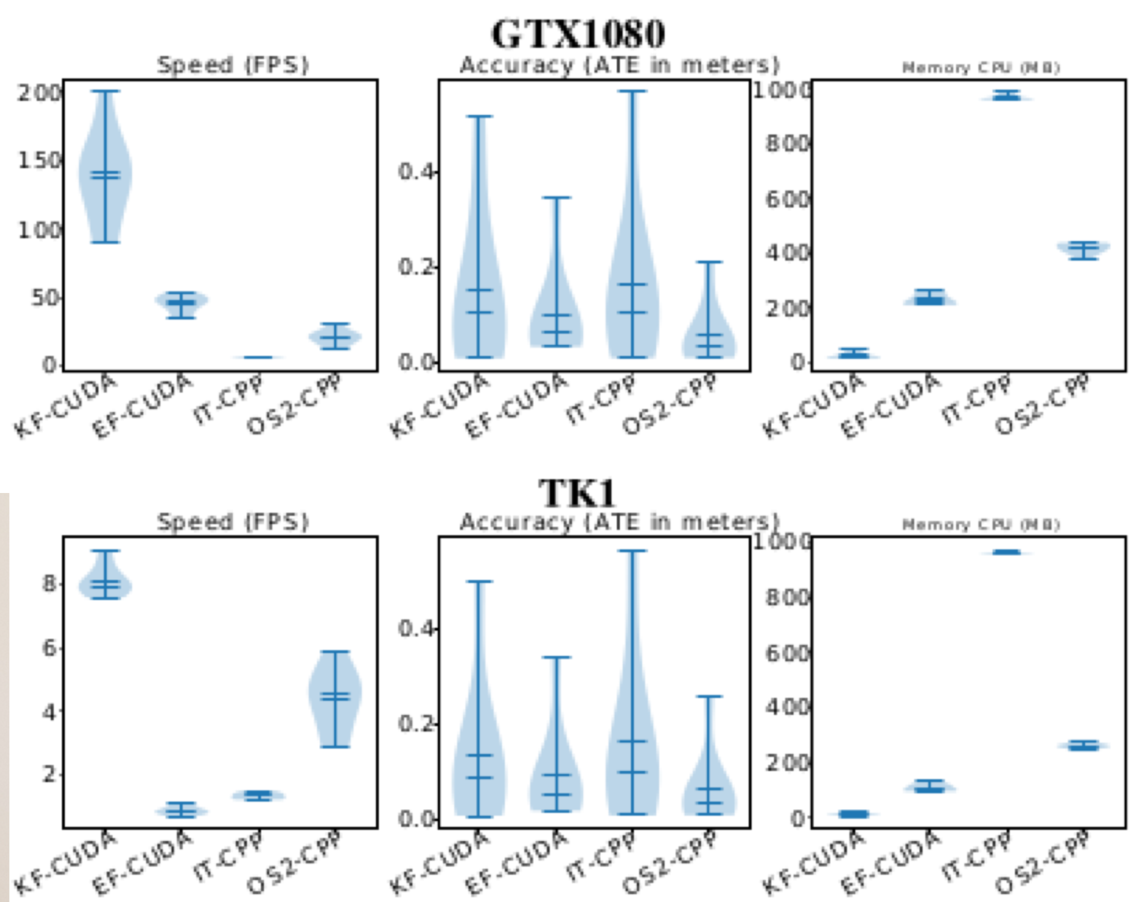
SLAMBench 2



Datasets
TUM
ICL-NUIM
I/O
SLAMBENCH FILE
API
SLAMBENCH LIB
Algorithms
LOADER
Qt UI
Front-ends
EXEC
Metrics
Speed
Memory
Power
CLICK AND COMPARE

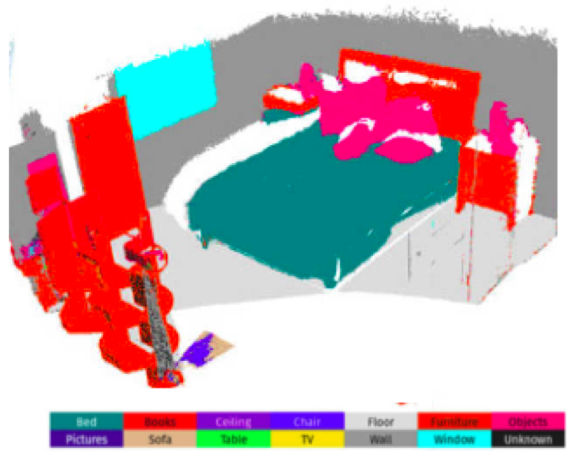
Comparing SLAM Systems

- 4 different SLAM implementations
- 2 different architectures
- 3 different metrics
- 6 dataset trajectory



SLAMBench 3.0: Beyond Traditional SLAM

Dynamic Reconstruction



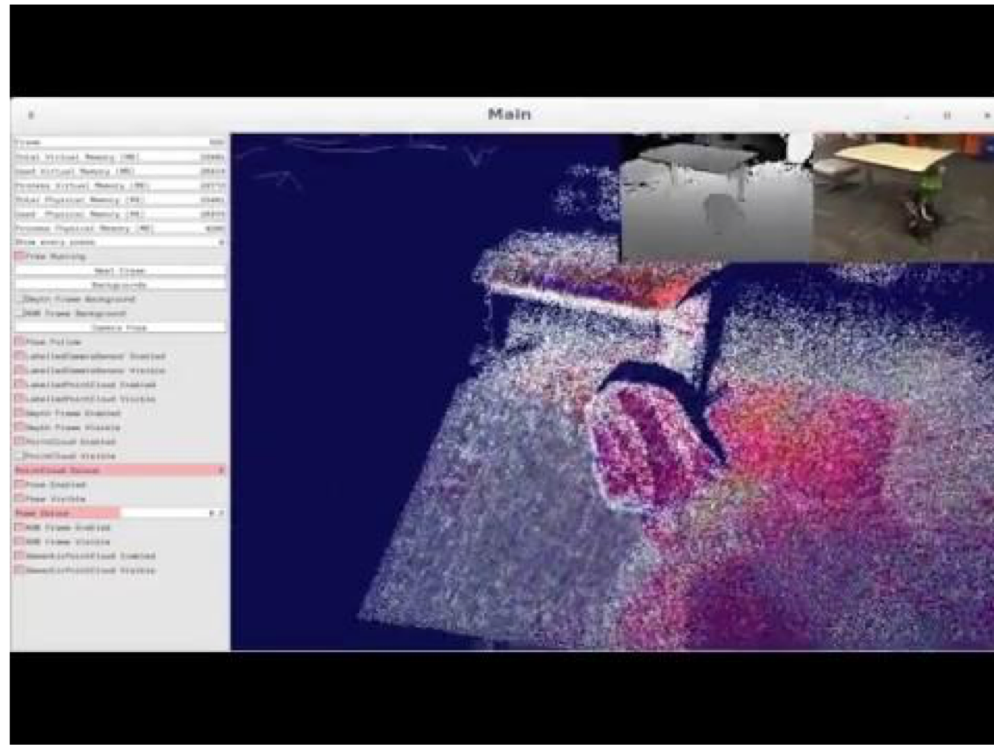
Semantic Reconstruction

Depth Estimation

[ICRA'19] SLAMBench 3.0: Systematic Automated Reproducible Evaluation of SLAM Systems for Robot Vision Challenges and Scene Understanding
Mihai Bujanca, Paul-Adrian Gafton, Sajad Saeedi, Andrew Nisbet, Bruno Bodin, Michael F P O'Boyle, Andrew J Davison, Paul H J Kelly, Graham Riley, Barry Lennox, Mikel Luján, Steve Furber

Semantic Reconstruction

- Metric: reprojected pixel accuracy
- Datasets:
  - NYU-RGB-D
  - ScanNet
- Algorithms:
  - SemanticFusion
  - ORB-SLAM-CNN

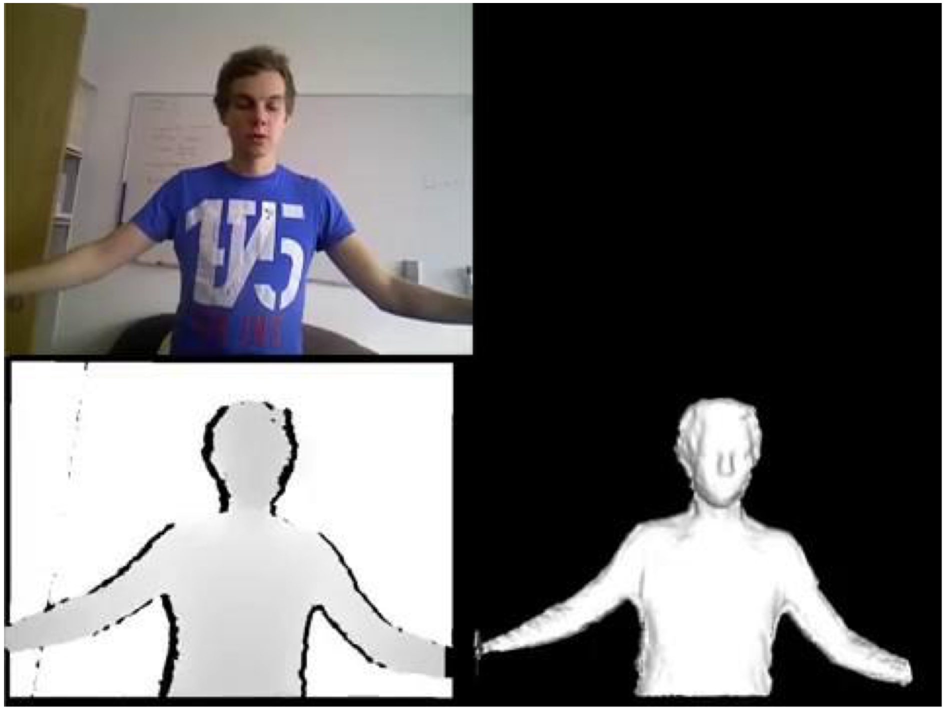


Depth Estimation

- Metrics:
  - Absolute Relative Difference
  - Accurate depth percentage
- Datasets
  - Any RGB-D dataset
- Algorithms
  - FLAME

Non-Rigid Reconstruction

- Metric: reconstruction error
- Datasets:
  - VolumeDeform (Kinect, RGB-D)
  - *Elanattil et al.* (Synthetic, RGB-D)
- Algorithm:
  - DynamicFusion



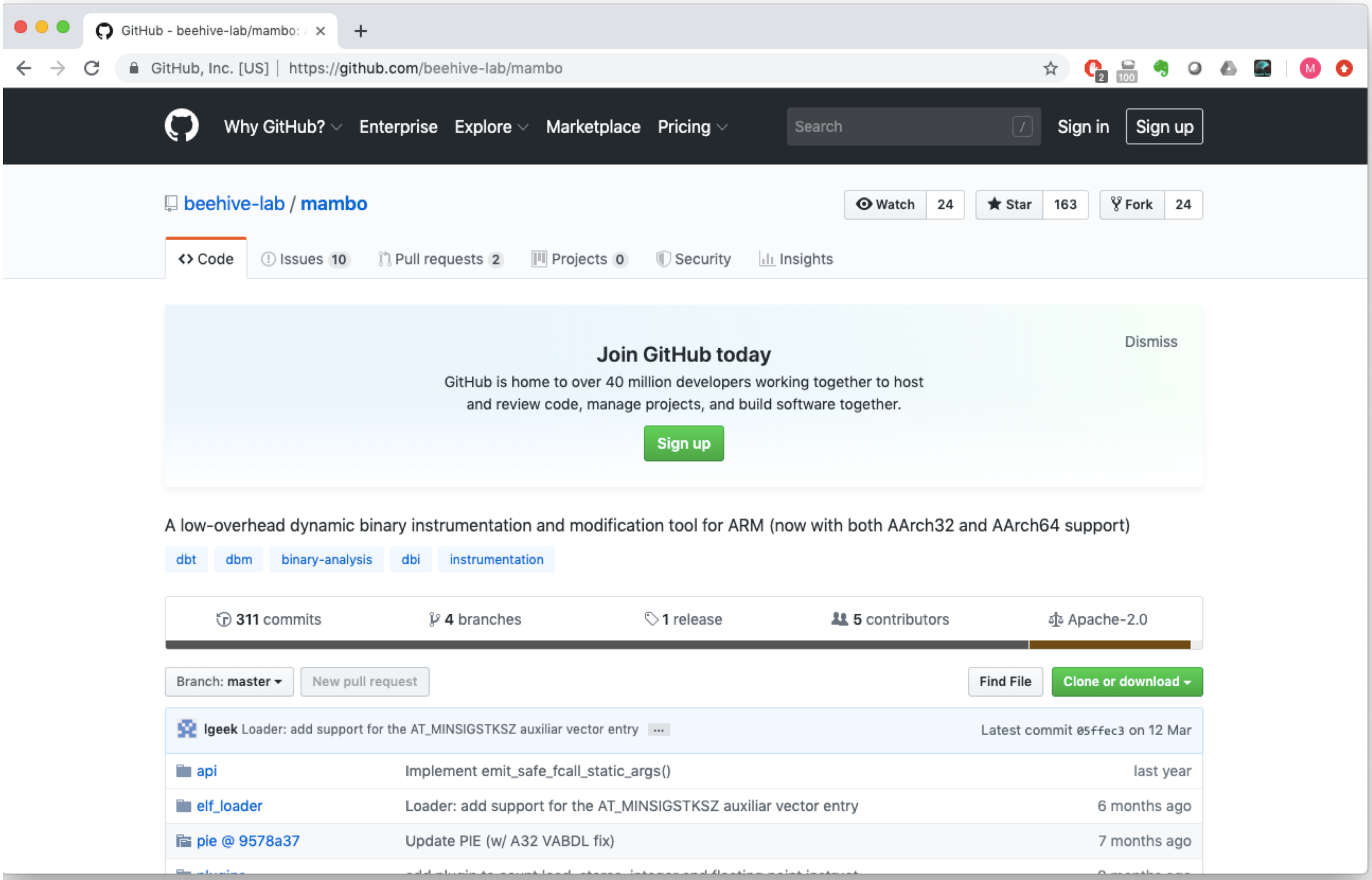
SLAM algorithms

Algorithm	Type	Sensors	Implementations	Year
ORB-SLAM	Sparse	RGB-D, Stereo, Monocular	C++	2016
OKVIS	Sparse	Stereo, IMU	C++	2015
SVO	Sparse	Monocular	C++	2014
MonoSLAM	Sparse	Monocular	C++, OpenCL	2007
PTAM	Sparse	Monocular	C++	2007
BundleFusion	Dense	RGB-D	CUDA	2016
ElasticFusion	Dense	RGB-D	CUDA	2015
InfiniTAM	Dense	RGB-D	C++, OpenMP, CUDA	2015
KinectFusion	Dense	RGB-D	C++, OpenMP, OpenCL, CUDA	2011
LSD-SLAM	Semi-Dense	Monocular	C++, PThread	2014
SemanticFusion	Dense, semantic	RGB-D	CUDA	2016
ORB-SLAM2-CNN	Sparse, semantic	Monocular	C++	2018
DynamicFusion	Dense, non-rigid	RGB-D	CUDA	2015
FLaME	Depth estimation	Monocular	C++	2017

Taming Heterogenous System-on-Chips

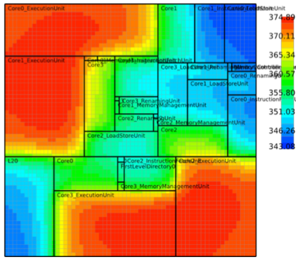
1. How to program them?
   - TornadoVM
     - https://github.com/beehive-lab/TornadoVM
     - VEE’17, ManLang’18, VEE’19
2. How to design, simulate, prototype, … them?
   - SLAMBench,
     - https://github.com/pamela-project/slambench2
     - ICRA’19, ICRA’18, PACT’16, ICRA’15
   - MAMBO & SimAcc
     - https://github.com/beehive-lab/mambo
     - FPL’14, TACO 2016, FCCM’17, FCCM’19
   - RAIN Hub, EuroEXA

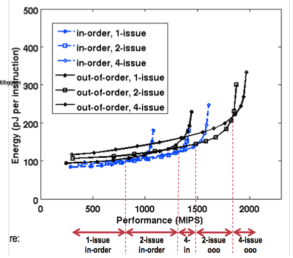
MAMBO – Dynamic Binary Instrumentation

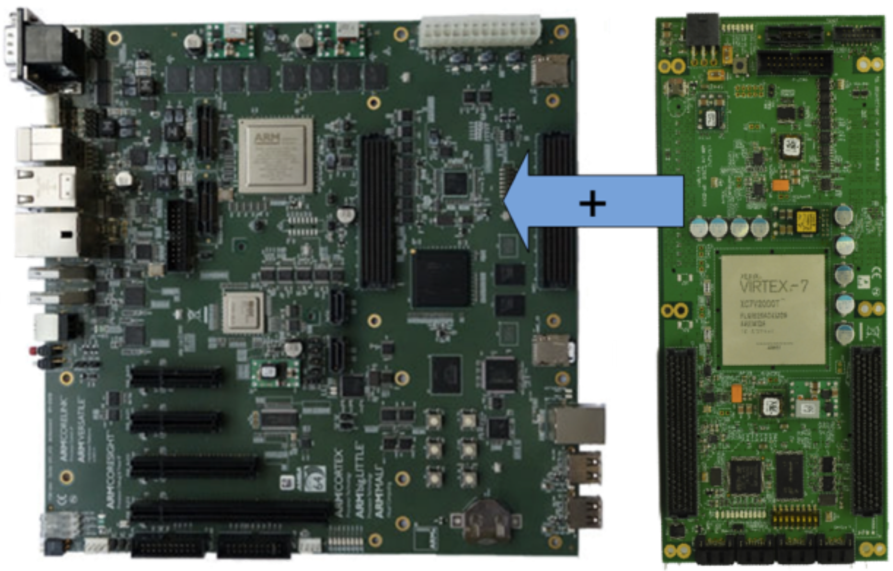


Simulation, Prototyping and Testing Heterogenous SoCs

- Gem5 – microarchitecture
- Compose IP blocks
  - EDA tools
  - FPGA prototyping
  - RTL emulation





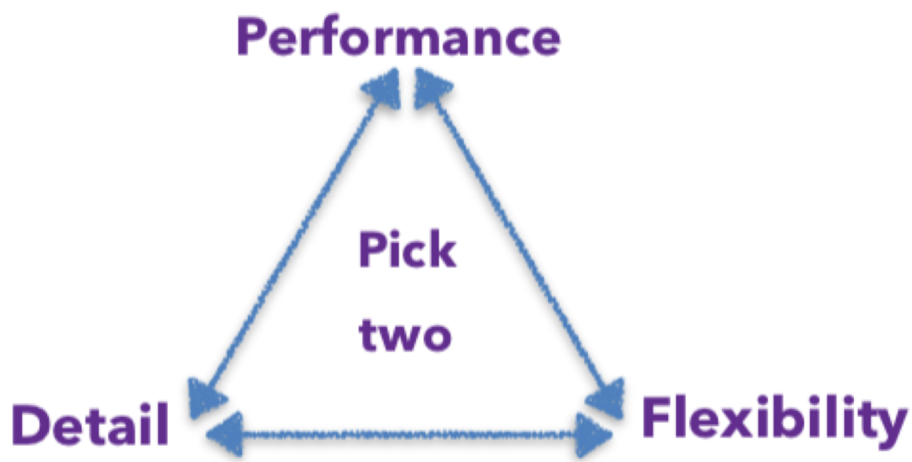


Motivation

In simulator's world, "good" and effective simulators are:

Fast, Accurate, Complete, Transparent, Inexpensive & Current.

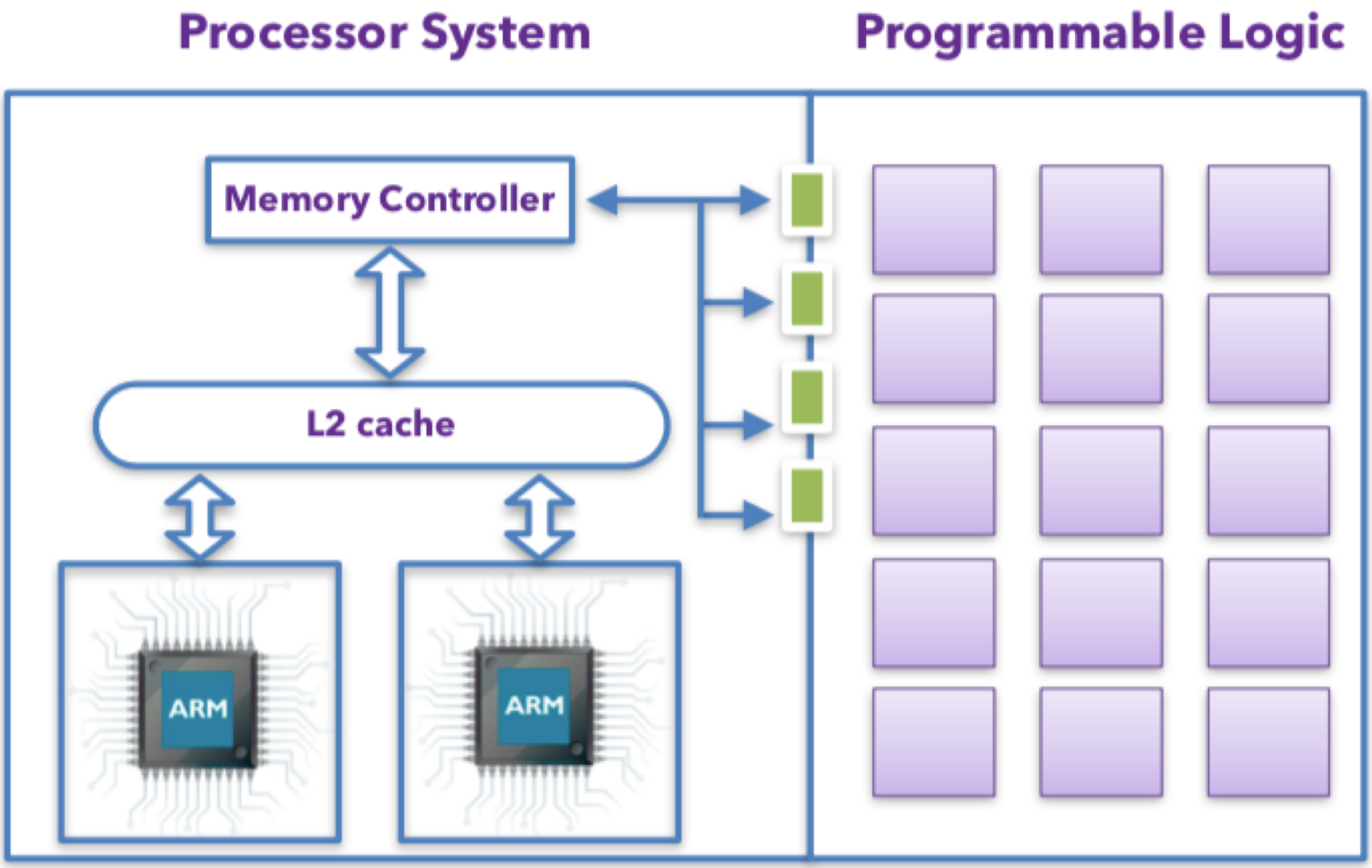
However… many of the above properties conflict with each other.



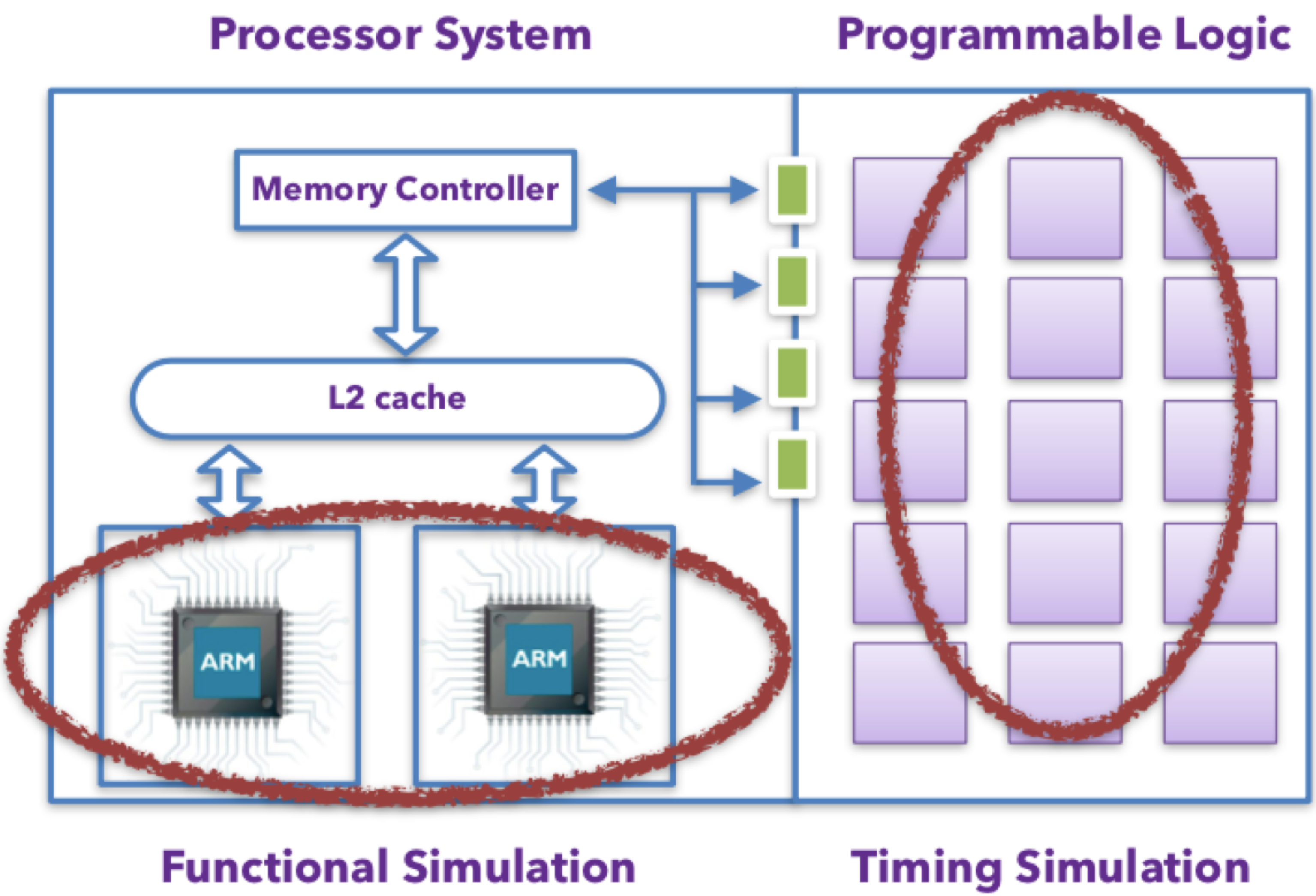
Evaluate next generation processor and system architectures:

- software-based cycle-accurate simulators.
- These simulators are: **transparent, easy-to-use and can be cycle-accurate but are generally not fast or complete and often not current.**
- There is a key research problem that needs to be addressed: **How do hardware accelerators interact with the processors of a system and what is the impact on overall performance?**

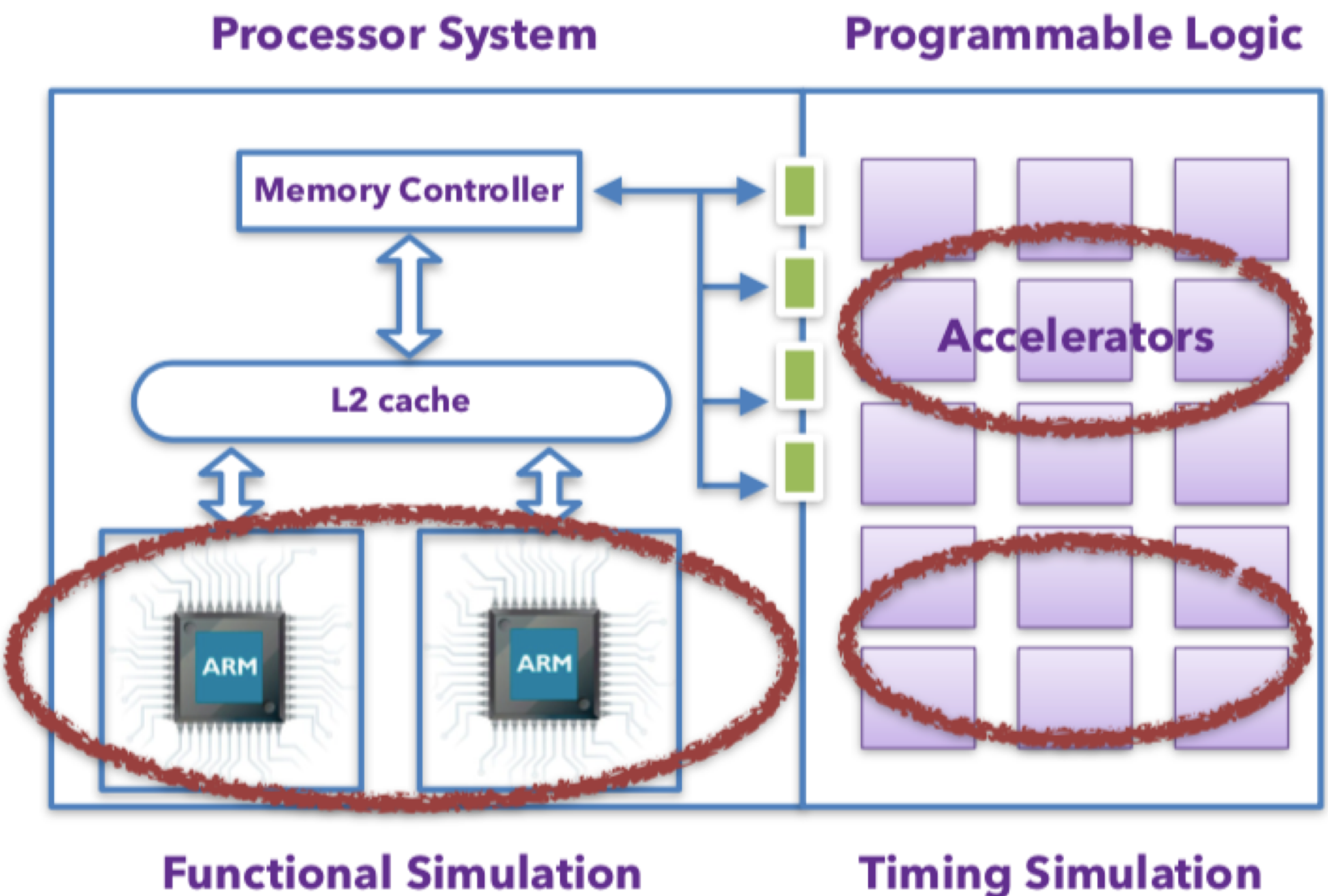
SoC with FPGA – e.g. Xilinx Zynq



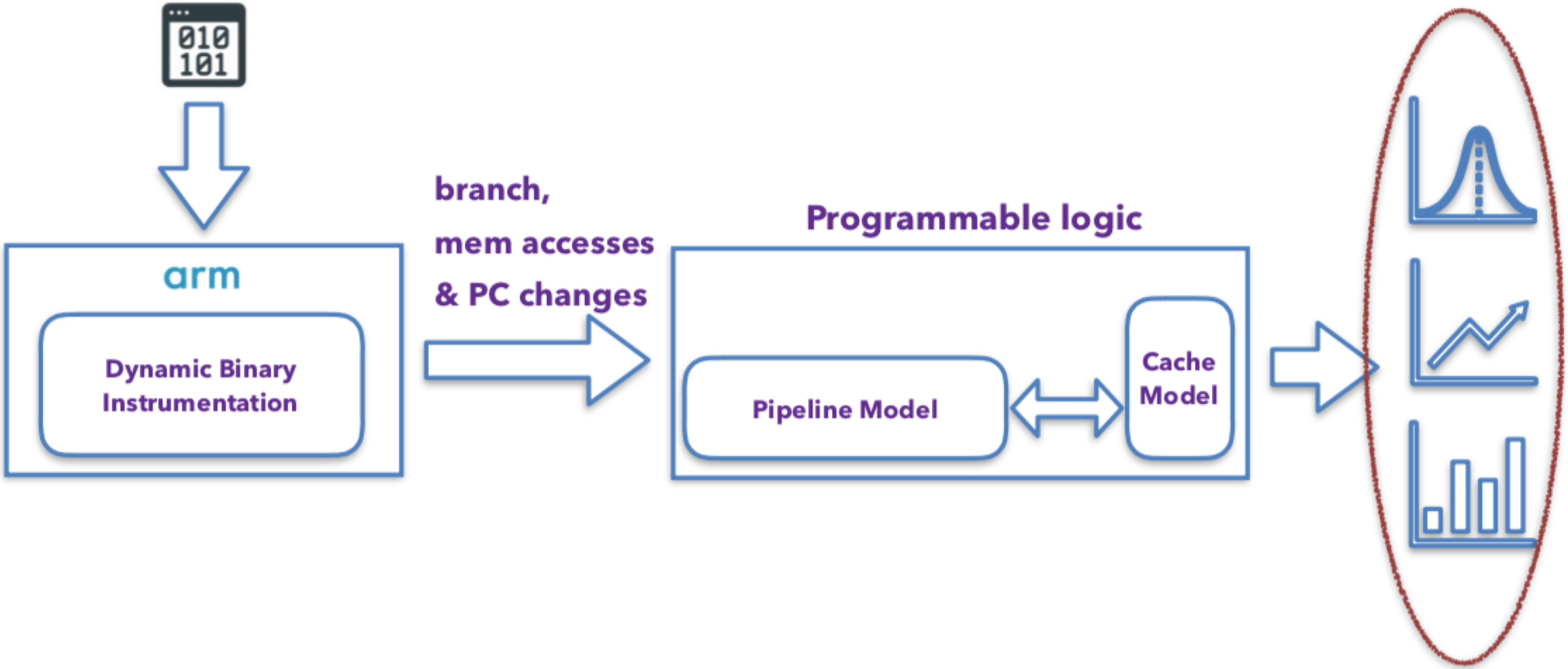
How can we design fast and accurate simulators?



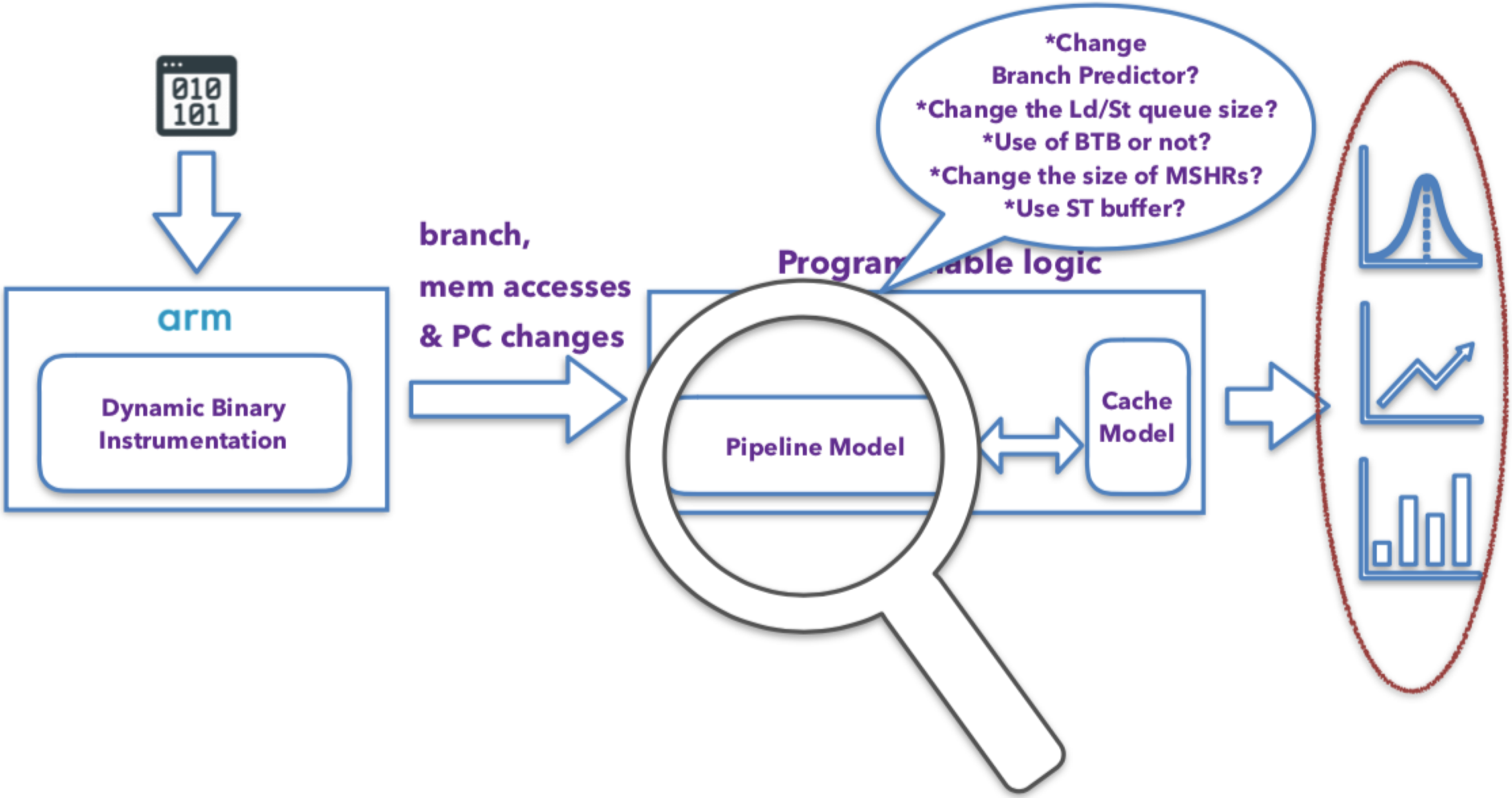
How can we design fast and accurate simulators while also prototyping an accelerator?



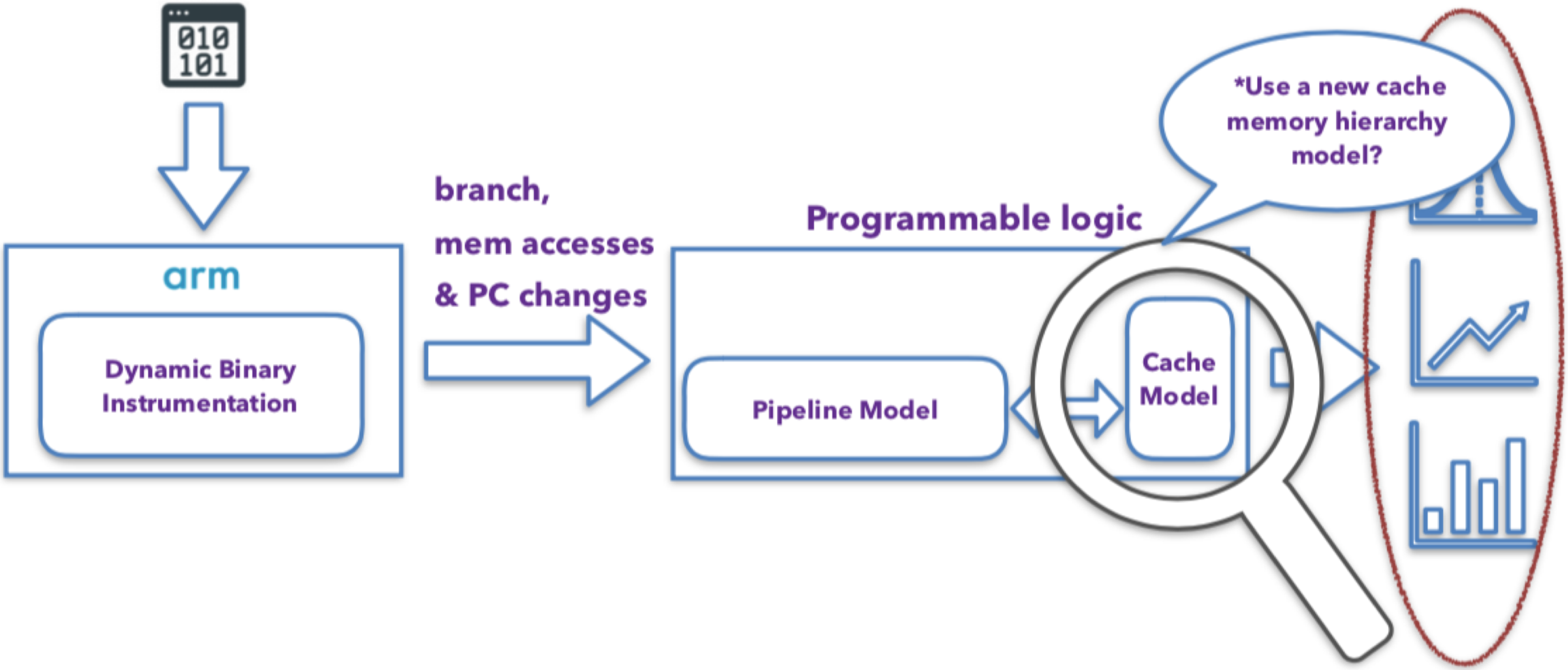
Simulator Infrastructure



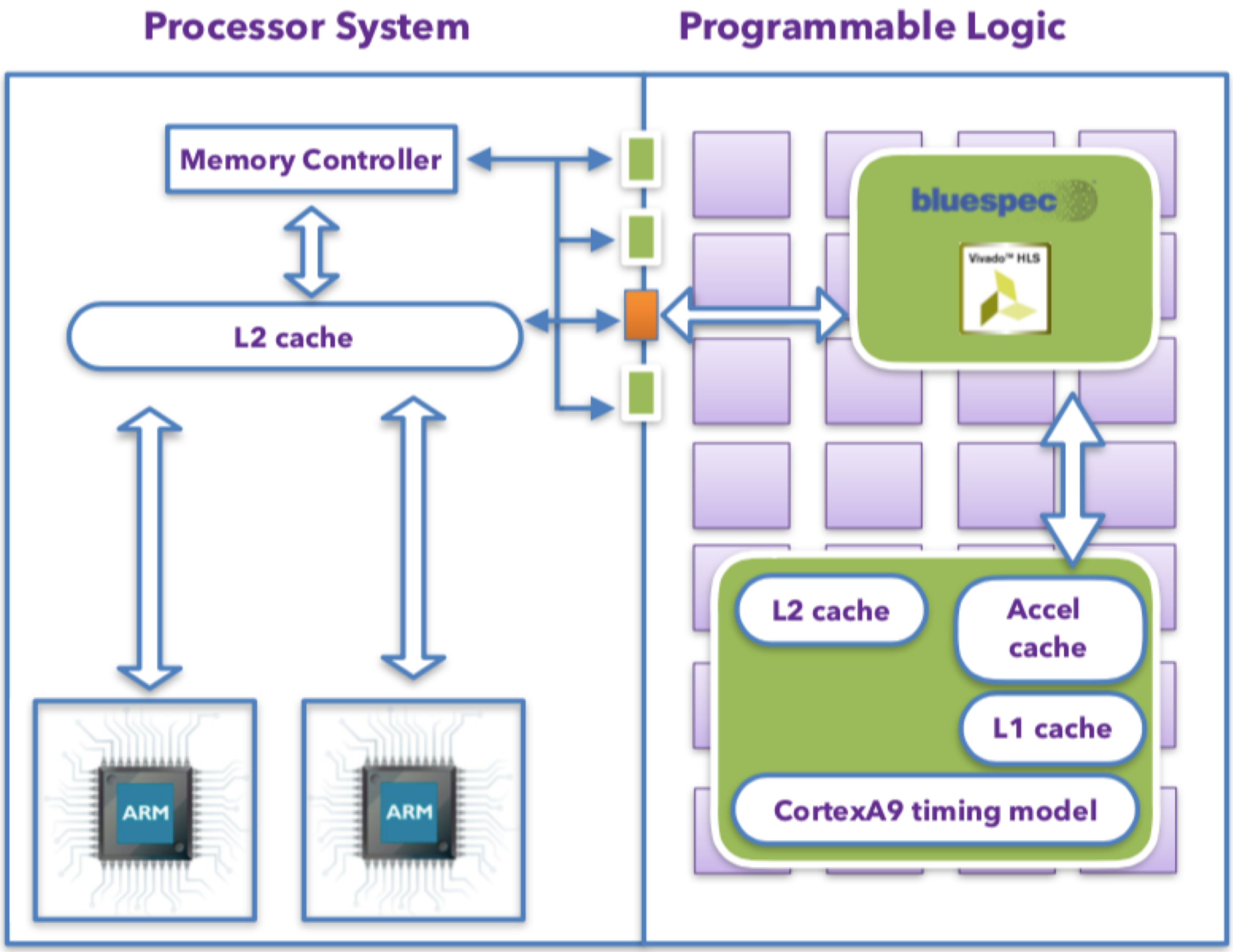
Simulator Infrastructure



Simulator Infrastructure

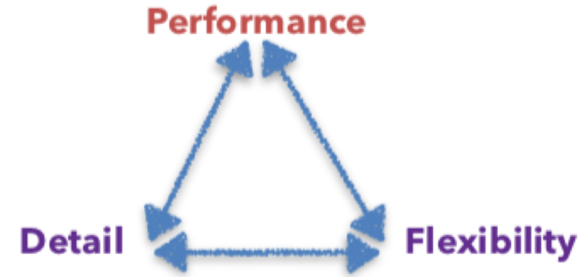


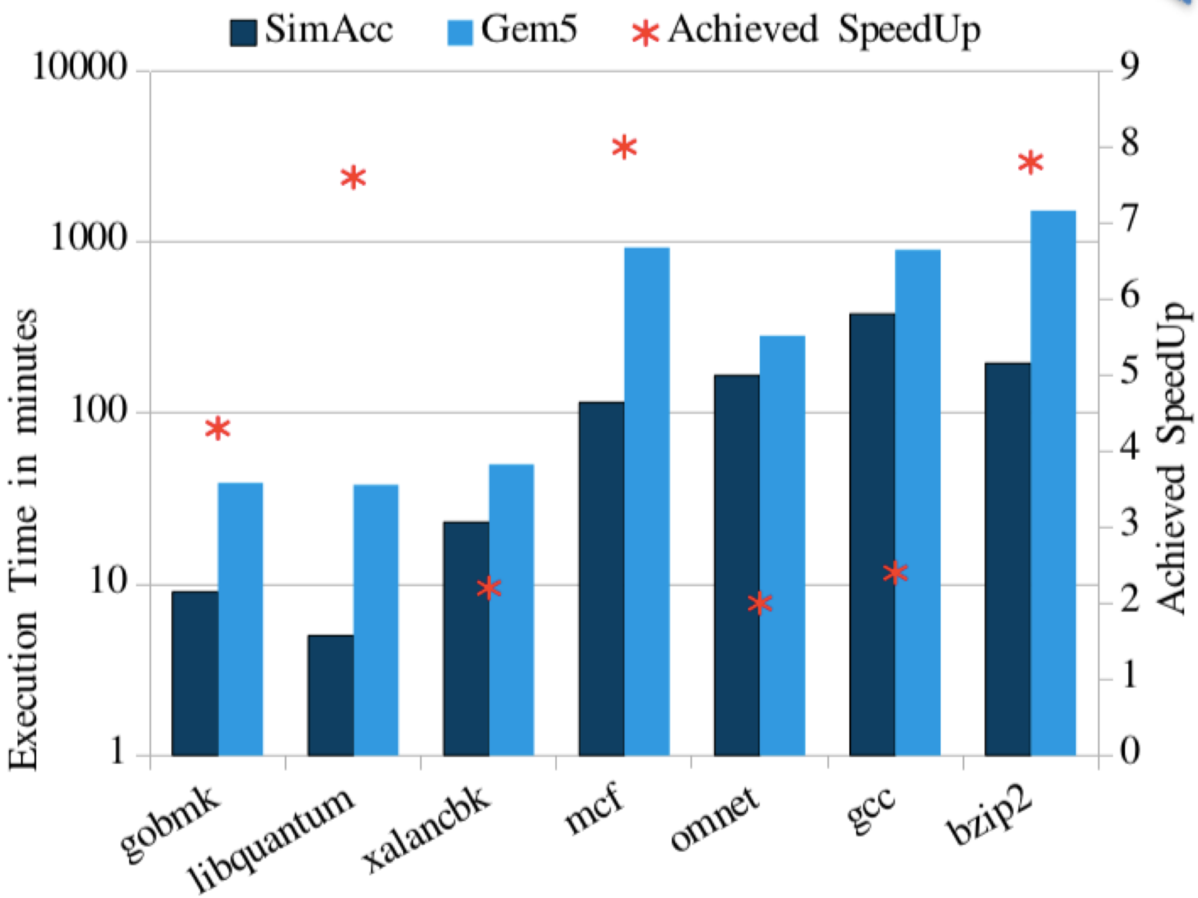
SoC with ORB-SLAM accelerators

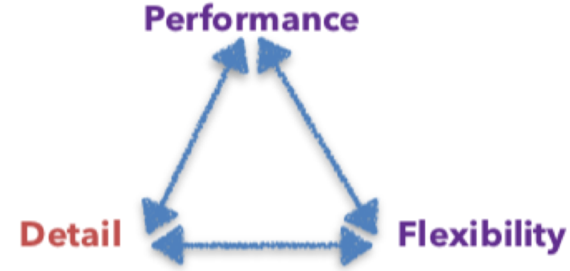


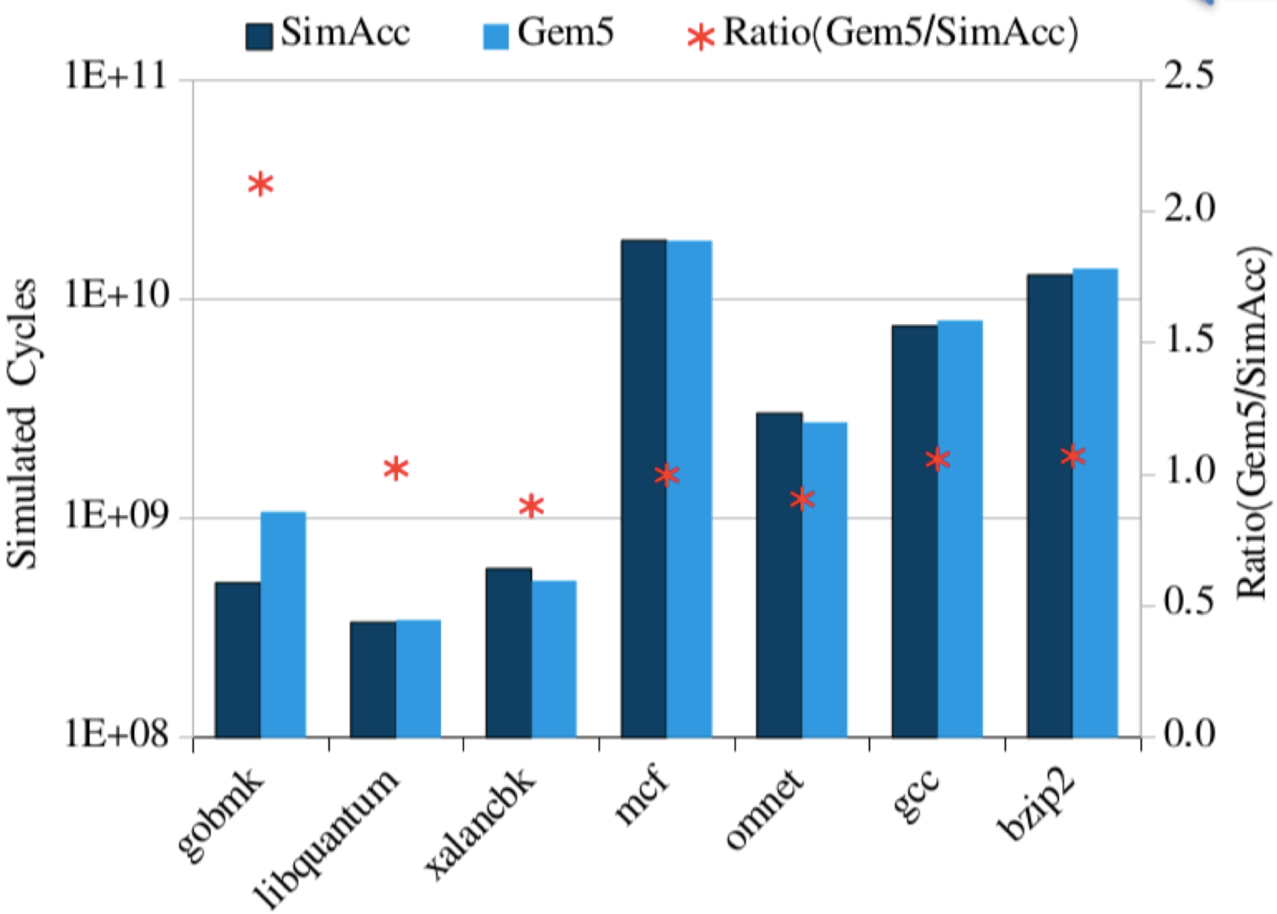
Experimental Evaluation

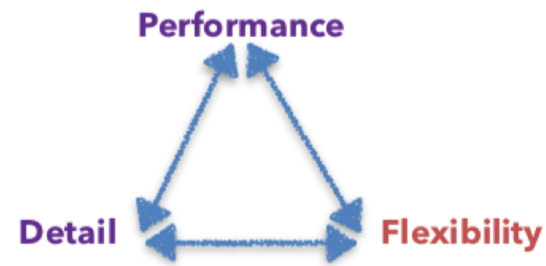
- SimAcc uses:
  - Xilinx Zynq-7000 XC7Z045 evaluation board running Ubuntu 14.04 with 1GB DRAM (no swap), and dual 667MHz arm Cortex-A9 processors.
- For the experimental evaluation, we are using gem5:
  - DerivO3 CPU model.
- Benchmarks:
  - SPEC CPU 2006.
  - Mach Benchmark Suite, Computer Vision Applications.

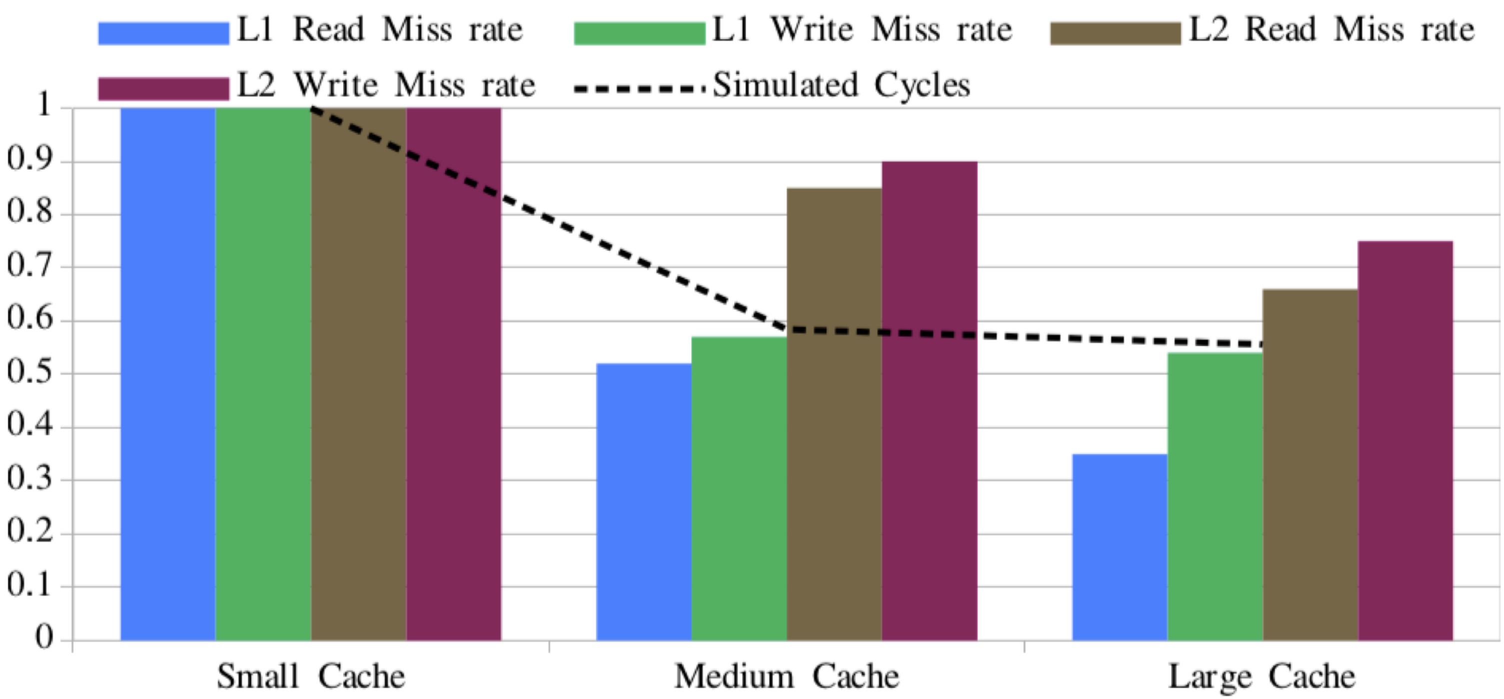

Performance
Detail
Flexibility


SimAcc
Gem5
Achieved SpeedUp
Execution Time in minutes
Achieved SpeedUp
10000
1000
100
10
1
9
8
7
6
5
4
3
2
1
0
gobmk
libquantum
xalancbk
mcf
omnet
gcc
bzip2


Performance
Detail
Flexibility


SimAcc
Gem5
Ratio(Gem5/SimAcc)
Simulated Cycles
Ratio(Gem5/SimAcc)
1E+11
1E+10
1E+09
1E+08
2.5
2.0
1.5
1.0
0.5
0.0
gobmk
libquantum
xalancbk
mcf
omnet
gcc
bzip2


Performance
Detail
Flexibility


L1 Read Miss rate
L1 Write Miss rate
L2 Read Miss rate
L2 Write Miss rate
Simulated Cycles
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Small Cache
Medium Cache
Large Cache

SimAcc

- We have demonstrated the potential of combining a flexible IP hardware library, a user-level driver library and dynamic binary instrumentation for microarchitecture simulation and prototyping.
- We exploit the advantages of an FPGA SoC to accelerate at a very fine granularity (instructions).
- We can benefit from the accuracy and speed of FPGA-based modelling and the ability to run binaries.
- It is the first FPGA-based simulator for microarchitecture combined with accelerators, significantly extending the options for simulating heterogenous SoC.

Taming Heterogenous System-on-Chips

1. How to program them?
   - TornadoVM
     - https://github.com/beehive-lab/TornadoVM
     - VEE’17, ManLang’18, VEE’19
2. How to design, simulate, prototype, … them?
   - SLAMBench,
     - https://github.com/pamela-project/slambench2
     - ICRA’19, ICRA’18, PACT’16, ICRA’15
   - MAMBO & SimAcc
     - https://github.com/beehive-lab/mambo
     - FPL’14, TACO 2016, FCCM’17, FCCM’19
   - RAIN Hub, EuroEXA

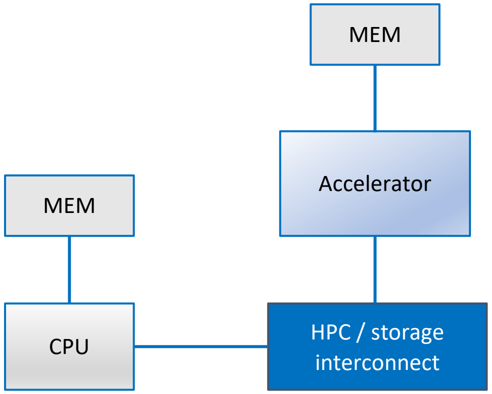
EUROEXA FUNDED BY THE EUROPEAN UNION

EuroEXA in a nutshell

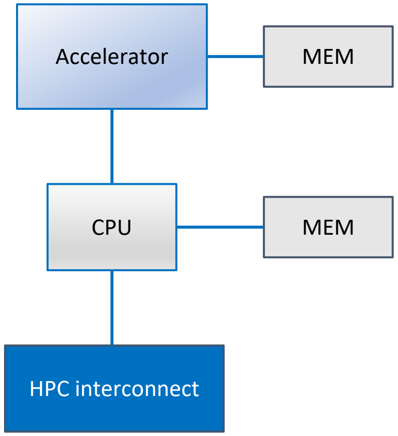
Start: Sep. 2017
Duration: 42 months
Funded under: H2020-EU.1.2.2

Commercial Partners	Academic/Gov. Partners	Supporters
ICEOTOPE, SYNELIXIS, arm, MAXELER Technologies, ZeroPoint Technologies	ICCS, The University of Manchester, INAF, Barcelona Supercomputing Center, imec, FORTH, Fraunhofer, Hartree Centre, ECMWF, NEURASMUS, INFN	XILINX, SYNOPSYS, SKA, Met Office, cnag, janssen, CRS4, NYU School of Medicine, Noldus, Schneider Electric, SOLVAY

EuroEXA node architecture

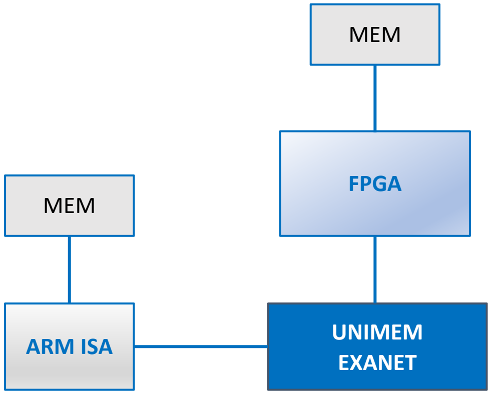


EuroEXA node architecture



Traditional HPC node architecture

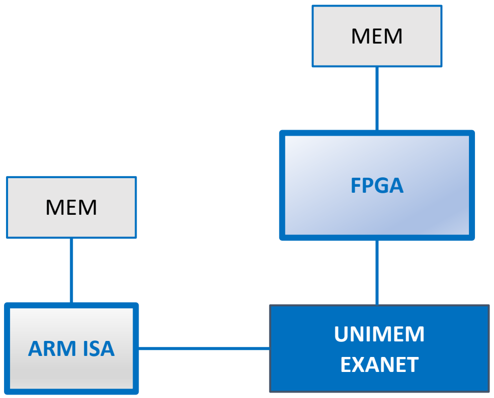
EuroEXA node architecture: some details



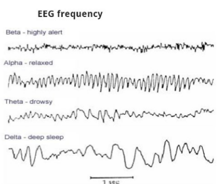
EuroEXA node architecture

- We employ **FPGAs** as our compute accelerator
- We innovate around the **ARM** ISA HW and SW ecosystem
- We scale-up with **EXANET** a low-latency, HPC network
- We support Global Shared Address Space (GSAS) with **UNIMEM**

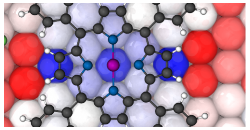
EUROEXA Application porting and optimization



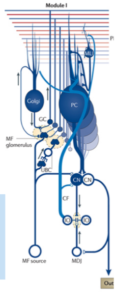
14 applications being ported and optimized for ARM + FPGA

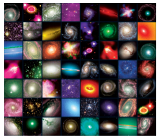
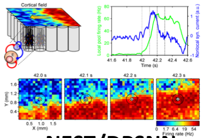
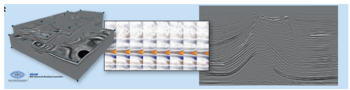


Neuromarketing



Quantum Espresso



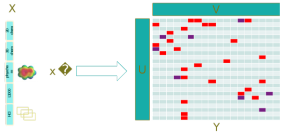
NEMO

Astronomy image classification

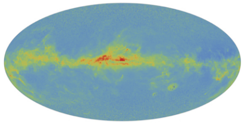
NEST/DPSNN

FRTM

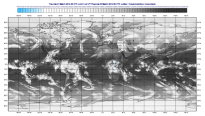
InfOli



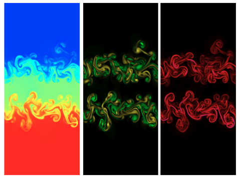
SMURFF



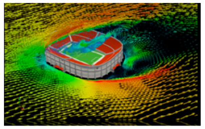
AVU-GSR



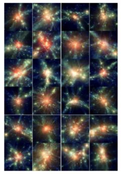
IFS



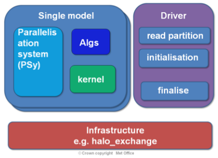
LBM



Alya

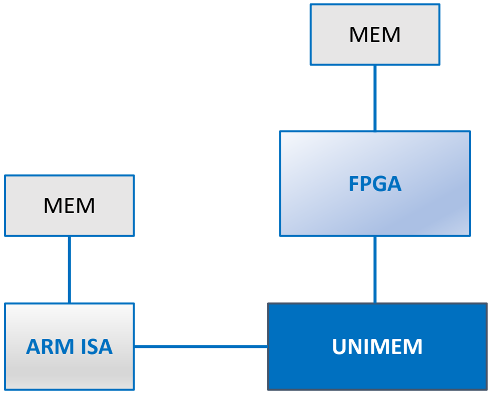


GADGET



LFRic

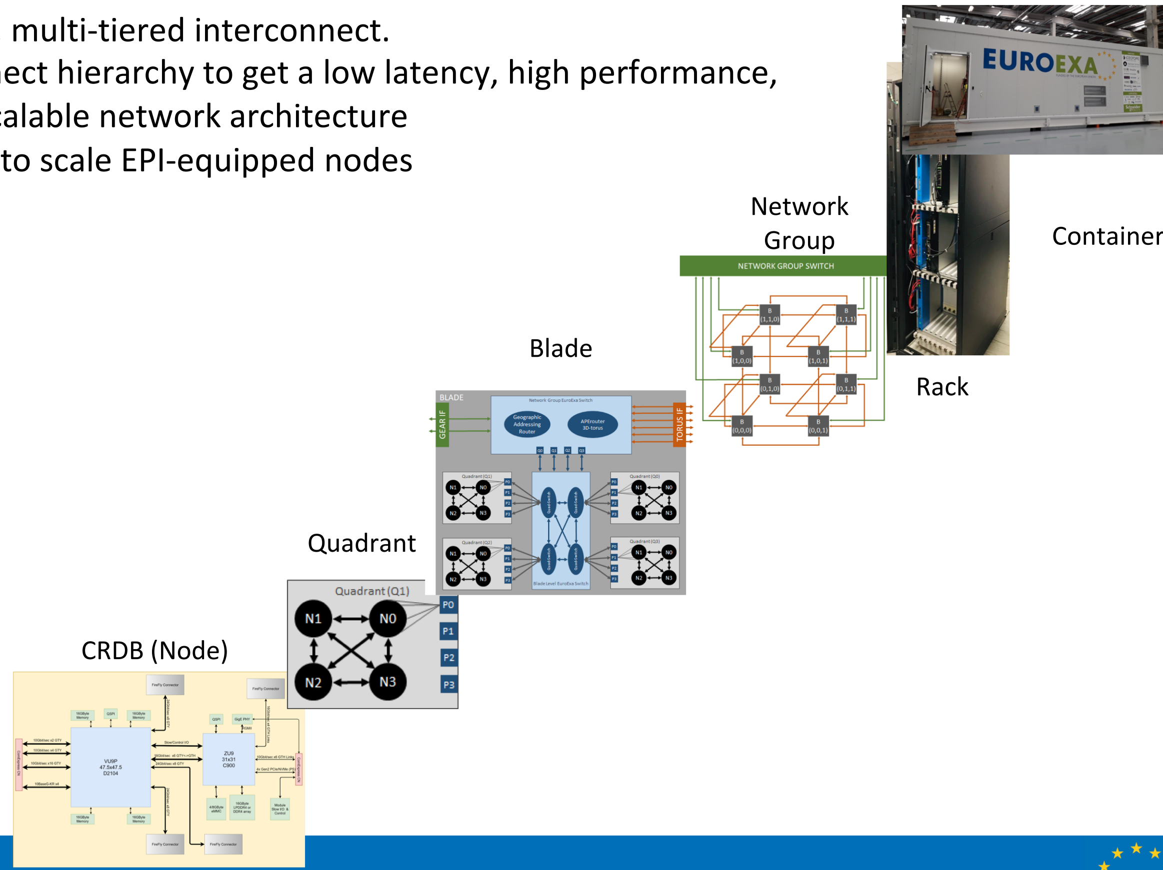
EuroEXA node architecture: GSAS with UNIMEM



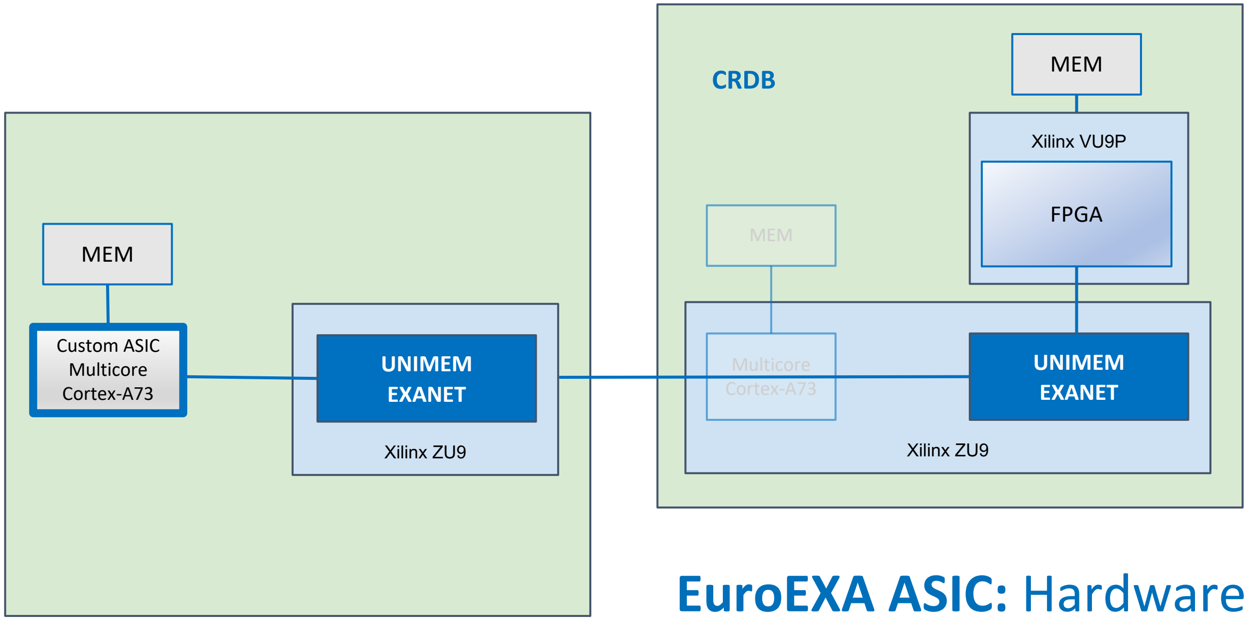
The UNIMEM technology developed within EuroEXA (and parent projects) can support Global Shared Address Space (GSAS) in a large scale system

EuroEXA node architecture: scaling out with ExaNet

ExaNet is the EuroEXA approach for large-scale, multi-tiered interconnect.
ExaNet is tailored to each level of the interconnect hierarchy to get a low latency, high performance, power effective, low resources consumption, scalable network architecture
ExaNet's system level architecture can be used to scale EPI-equipped nodes



EuroEXA system: implementation

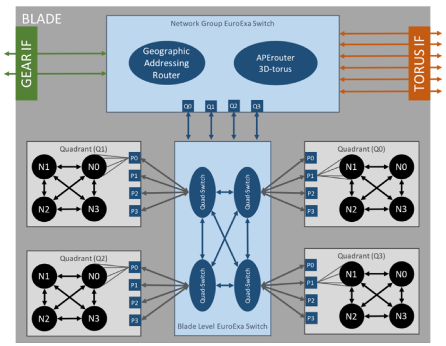


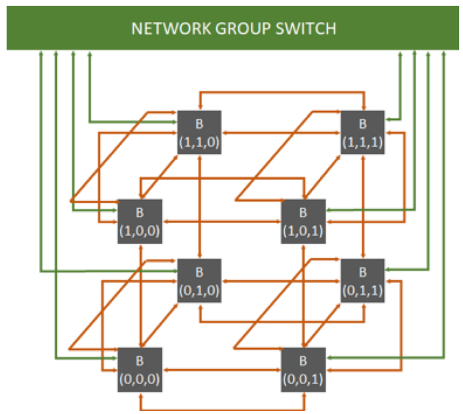
EuroEXA testbed 3

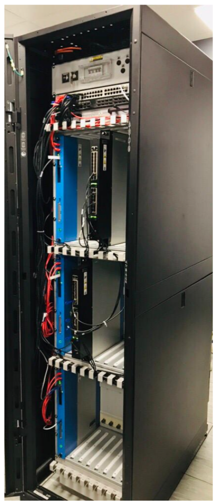
Number of nodes: TBD

EuroEXA ASIC: Hardware support for UNIMEM, EXANET and Memory Compression

Integration and scale out







Blade
16 slots (COME-Extended open standard)
Each slot could be CPU+Accelerator

Network Group:
- 8 Blades
- 6400Gbps of Interconnect
- Torus topology

Rack
- 4 netgroups
- (32 blades, 512 slots)
- more than 110kW

System
- 2MW in a modular facility
- PUE 1.0x
- Heat Reuse Capable
- Low Cost Facilities

Robotics in Nuclear Environments

RAIN

ROBOTICS AND AI IN NUCLEAR

£12.2M between 2017-2021.
Involving UoM: EEE, CS, MACE, Physics
Universities of Bristol, Lancaster, Liverpool, Oxford, Sheffield and Nottingham.
UKAEA's RACE centre.
Focus on transfer of technology to industry.

RNE

ROBOTICS FOR NUCLEAR ENVIRONMENTS

£4.6M Programme Grant (2016-2021)
Involving UoM: EEE
Universities of West of England and Birmingham.
Focus on fundamental research to support nuclear sector in Cumbria.

UK Research and Innovation

Sellafield Ltd · EDF ENERGY · JACOBS · National Nuclear Laboratory · AWE · Rolls Royce · UK Atomic Energy Authority · bp

EPSRC
Engineering and Physical Sciences Research Council

What are the challenges that we're trying to address?

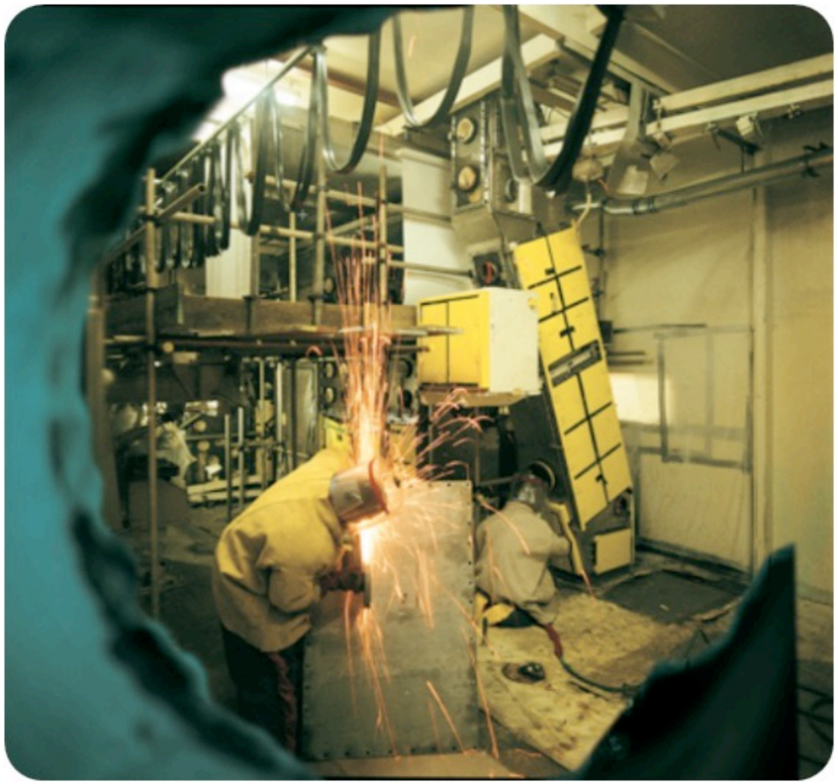
Sellafield Site

guardian.co.uk | The Observer

Sellafield: the most hazardous place in Europe

Last week the government announced plans for a new generation of nuclear plants. But Britain is still dealing with the legacy of its first atomic installation at Sellafield - a toxic waste dump in one of the most contaminated buildings in Europe. As a multi-billion-pound clean-up is planned, can we avoid making the same mistakes again?

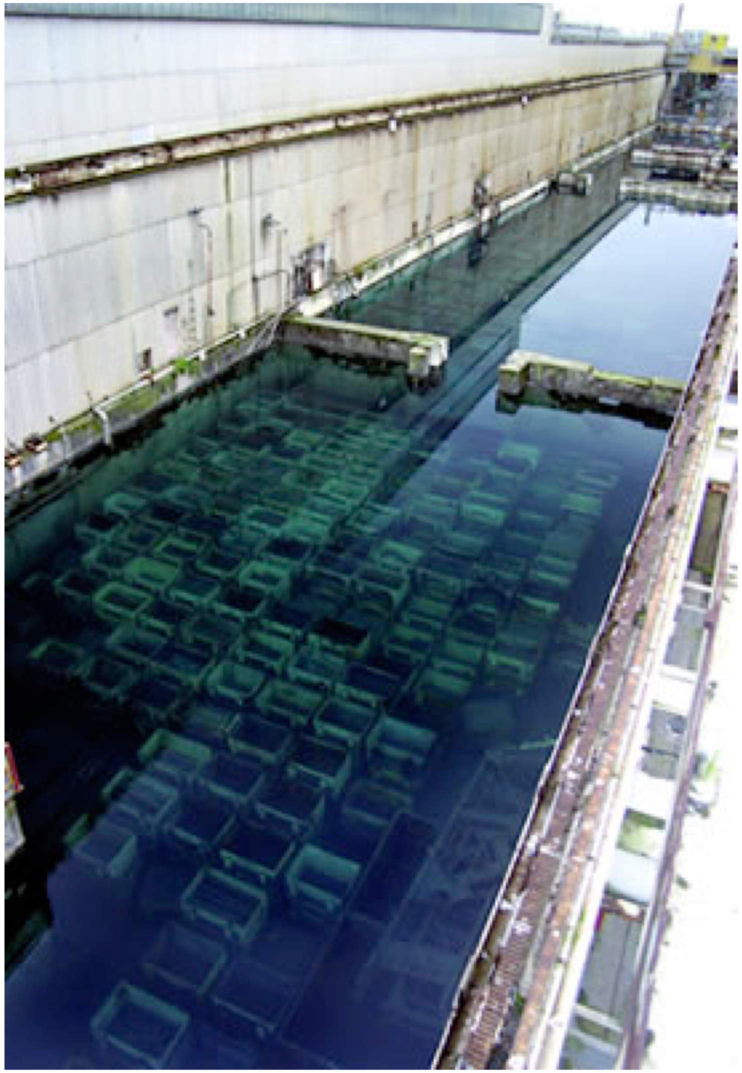
Sellafield Challenges



Alpha Decommissioning (Sellafield):
- Minimise airborne contamination.
- Minimise waste volumes.

Diverse range of glove boxes require decommissioning.

First Generation Magnox Storage Pond



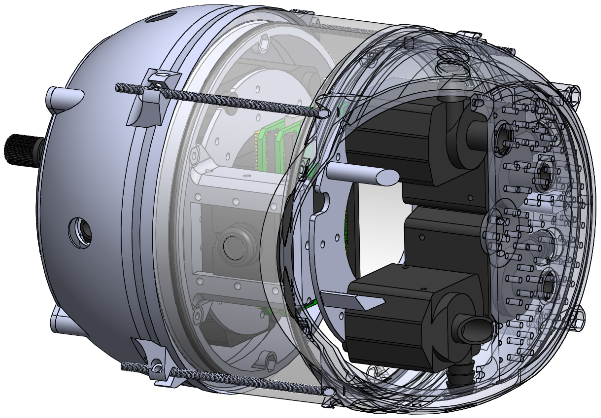
Legacy pond inspection: AVEXIS

- Autonomous Vehicle for Exploration and In-situ Sensing (AVEXIS) has been developed in partnership with Sellafield and Forth Engineering.

- Two versions exist
  - AVEXIS 150 for restricted access deployment
  - AVEXIS Prime for untethered monitoring

The University of Manchester

AVEXIS 150



- Vehicle fits through a 150 mm access port.
- Vehicle is 3D printed to reduce development time and construction cost (it can be disposable).
- 3D print material is porous:
  - Dual hull design is used to ensure vehicle is waterproof.
  - Thickness of outer shell is minimised to avoid contamination.
- Contains camera and radiological sensor.

Summary

Taming Heterogenous System-on-Chips

1. How to program them?
   - TornadoVM
     - https://github.com/beehive-lab/TornadoVM
     - VEE'17, ManLang'18, VEE'19
2. How to design, simulate, prototype, … them?
   - SLAMBench,
     - https://github.com/pamela-project/slambench2
     - ICRA'19, ICRA'18, PACT'16, ICRA'15
   - MAMBO & SimAcc
     - https://github.com/beehive-lab/mambo
     - FPL'14, TACO 2016, FCCM'17, FCCM'19
   - RAIN Hub, EuroEXA

MANCHESTER
1824
The University of Manchester

Acknowledgements

European Commission

ROYAL ACADEMY OF ENGINEERING

EPSRC
Engineering and Physical Sciences Research Council

THE ROYAL SOCIETY

arm

Microsoft

ORACLE

Links and References (1/4)

- RAIN hub https://rainhub.org.uk/
- EuroEXA https://euroexa.eu/
- TornadoVM https://github.com/beehive-lab/TornadoVM
  - Main contact point: Dr Kristos Kotselidis
    - https://www.kotselidis.net/
    - Collaboration with Flink: H2020 E2DATA https://e2data.eu/
- SLAMBench
  - https://github.com/pamela-project/slambench2
- MAMBO
  - https://github.com/beehive-lab/mambo
- Other open-source projects that I did not have time to talk about
  - Maxine VM https://github.com/beehive-lab/Maxine-VM

Links and References (2/4)

VEE’17

Heterogeneous Managed Runtime Systems: A Computer Vision Case Study

DOI https://doi.org/10.1145/3050748.3050764

ManLang’18

Exploiting high-performance heterogeneous hardware for Java programs using Graal DOI https://doi.org/10.1145/3237009.3237016

VEE’19

Dynamic Application Reconfiguration on Heterogeneous Hardware

DOI https://doi.org/10.1145/3313808.3313819

ICRA’15

Introducing SLAMBench, a performance and accuracy benchmarking methodology for SLAM DOI https://doi.org/10.1109/ICRA.2015.7140009

PACT’16

Integrating Algorithmic Parameters into Benchmarking and Design Space Exploration in 3D Scene Understanding

DOI https://doi.org/10.1145/2967938.2967963

Links and References (3/4)

ICRA'18

SLAMBench2: Multi-Objective Head-to-Head Benchmarking for Visual SLAM

DOI https://doi.org/10.1109/ICRA.2018.8460558

ICRA'19

SLAMBench 3.0: Systematic Automated Reproducible Evaluation of SLAM Systems for Robot Vision Challenges and Scene Understanding

DOI https://doi.org/10.1109/ICRA.2019.8794369

TACO 2016

MAMBO: A Low-Overhead Dynamic Binary Modification Tool for ARM

DOI https://doi.org/10.1145/2896451

FPL'14

An empirical evaluation of High-Level Synthesis languages and tools for database acceleration

DOI https://doi.org/10.1109/FPL.2014.6927484

Links and References (4/4)

FCCM'17

The Potential of Dynamic Binary Modification and CPU-FPGA SoCs for Simulation DOI https://doi.org/10.1109/FCCM.2017.36

FCCM'19

SimAcc: A Configurable Cycle-Accurate Simulator for Customized Accelerators on CPU-FPGAs SoCs

DOI https://doi.org/10.1109/FCCM.2019.00031

Newcombe 2011 - KinnectFusion

KinectFusion: Real-time dense surface mapping and tracking

DOI https://doi.org/10.1109/ISMAR.2011.6092378

TornadoVM implementation https://github.com/beehive-lab/kfusion-tornadovm

ORB-SLAM & ORB-SLAM2

https://github.com/raulmur