



Accelerating Deep Learning for Embedded Vision at the Edge

May 22, 2019

Note: participants are muted upon entering the webinar.
Please use the chat feature to ask questions.

Hugh Pollitt-Smith, CMC Microsystems, Pollitt-smith@cmc.ca

- Demonstrate an integrated Machine Learning (ML) training and inference flow utilizing tools and hardware available to CNDN
 - Xilinx SDSoc/DNNDK
 - CMC HPP/HCC
 - Xilinx ZCU102 Development Kit
- Exploit reconfigurable, heterogeneous processing
- Builds on previous webinar, *Accelerating Deep Learning for Vision Using Caffe* (February 27, 2019), posted on CMC's YouTube channel

Note: this work was undertaken through National Defence Innovation for Defence Excellence and Security (IDEaS) competitive project

Agenda



- CMC Microsystems
- Overview
- Hardware and software environment
- CNN Training and Quantization Flow
- Inference Demonstration
- How to access
- Q&A

Agenda

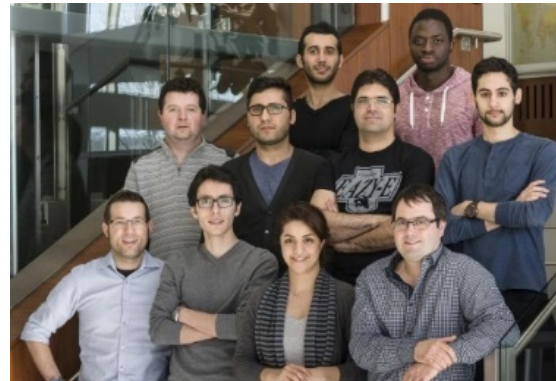


- **CMC Microsystems**
- Overview
- Hardware and software environment
- CNN Training and Quantization Flow
- Inference Demonstration
- How to access
- Q&A

What is CMC and what is its role?



- Not for profit – federally incorporated 1984
- Manages Canada's National Design Network[®]
- Delivers micro-nano innovation capabilities across Canada



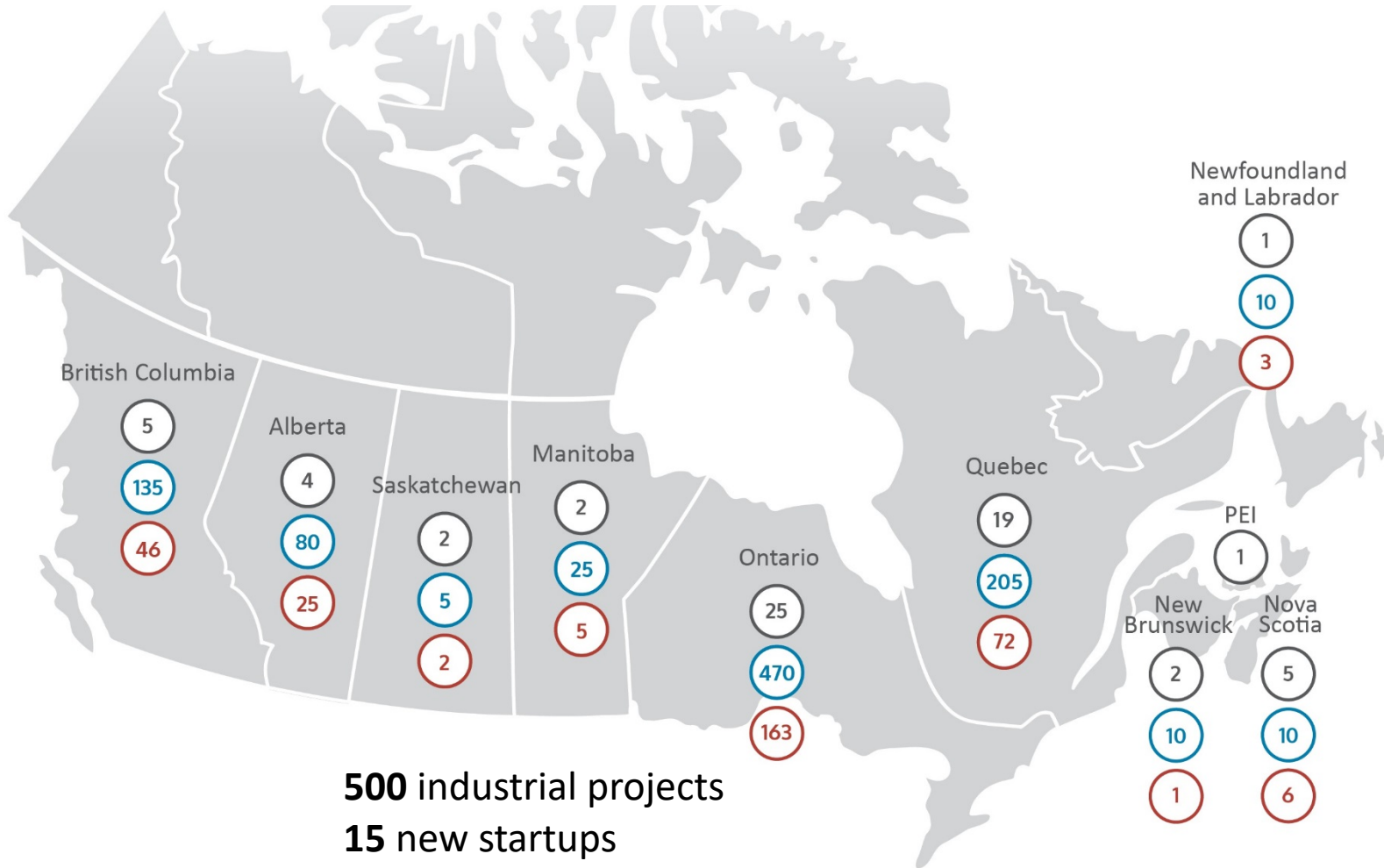
Academic and
Industrial Users



Canada's National Design Network

A Canada-wide collaboration between **66** universities/colleges to connect **10,000** academic participants with **950** companies to design, make and test micro-nanosystem prototypes. CMC Microsystems manages Canada's National Design Network®.

- Post-secondary institutions
- Collaborating companies
- Companies manufacturing micro-nanosystems products in Canada



Annually:

1200 connected professors
4200 researchers on professors' teams
5700 users of computer-aided design tools
300 physical prototypes
80 test equipment loan items otherwise unaffordable to users

2017 Outcomes:

3780 publications
110 awards
160 patents awarded & applied

500 industrial projects
15 new startups
780 trained HQP moved to industry in Canada

Measured outcomes, published annually

Research Outcomes for 2017:

1662 journal publications

2116 other publications

53 national awards

57 international awards

443 graduate student courses

606 undergraduate student courses

Commercialization Outcomes for 2017:

14 startup companies

160 patents (applied for/issued)

27 licences

442 interactions with industry in Canada, valued at **\$21.9M**

57 interactions with foreign industry, valued at **\$4.5M**

Value to industry is measured in research collaborations, transfer of highly qualified people, and direct company access of tools and technologies for research collaborations.

700 highly trained researchers joined industry in 2017

LOWERING BARRIERS TO TECHNOLOGY ADOPTION



CAD



Services for making working prototypes

- ✓ Selection of high-performance Computer Aided Design (CAD) tools and design environments
- ✓ Available via desktop or through CMC Cloud
- ✓ User guides, application notes, training materials and courses

 [CMC.ca/CAD](https://cmc.ca/CAD)

FAB



Services for making working prototypes

- ✓ Multi-project wafer services with affordable access to foundries worldwide
- ✓ Fabrication and travel assistance to prototype at a university-based lab
- ✓ Value-added packaging and assembly services
- ✓ In-house expertise for first-time-right prototypes

 [CMC.ca/FAB](https://cmc.ca/FAB)

LAB



Device validation to system demonstration

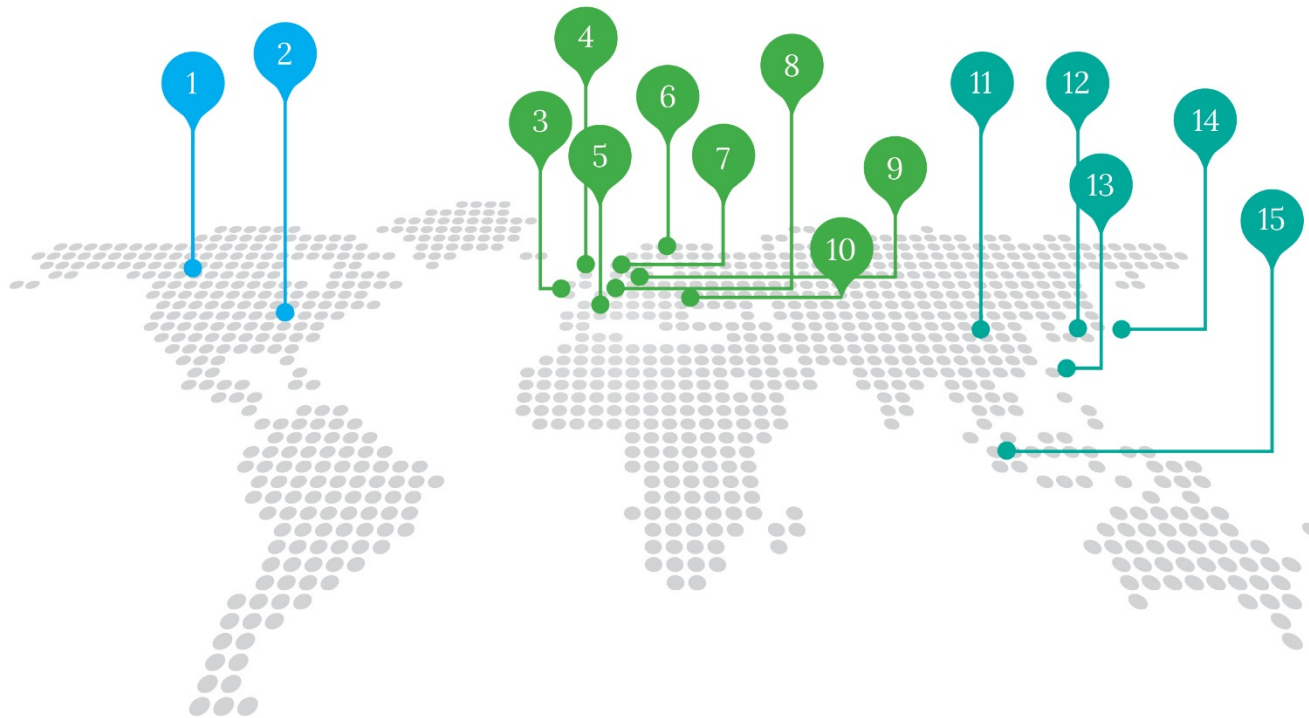
- ✓ Access to platform-based microsystems design and prototyping environments
- ✓ Access to test equipment on loan
- ✓ Access to contract engineering services

 [CMC.ca/LAB](https://cmc.ca/LAB)

ENGAGING STRATEGICALLY in Canada and worldwide



Strategic Engagements, Global Partners



North America

1. Canada	14 CAD 8 FAB 13 LABs 19 Systems & Components 42 University MNT LABS
2. USA	1 Co-operative Initiative 15 CAD 5 FAB 11 LABs 8 Systems & Components

Asia

11. China	1 Co-operative Initiative
12. South Korea	1 Co-operative Initiative
13. Taiwan	1 Co-operative Initiative 2 FAB 1 LAB 2 Systems & Components
14. Japan	1 Co-operative Initiative
15. Singapore	3 FAB

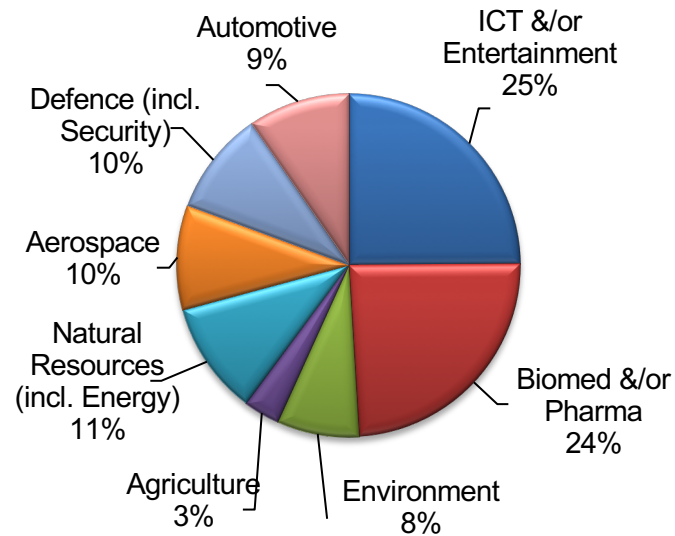
Europe 1 Co-operative Initiative

3. Ireland	1 FAB
4. UK	1 CAD 1 Systems & Components
5. France	2 FAB 1 Co-operative Initiative
6. Sweden	1 CAD
7. Netherlands	2 FAB
8. Belgium	1 FAB
9. Germany	1 CAD 2 FAB
10. Austria	1 FAB

Canada's National Design Network Academic Landscape 2017-2018

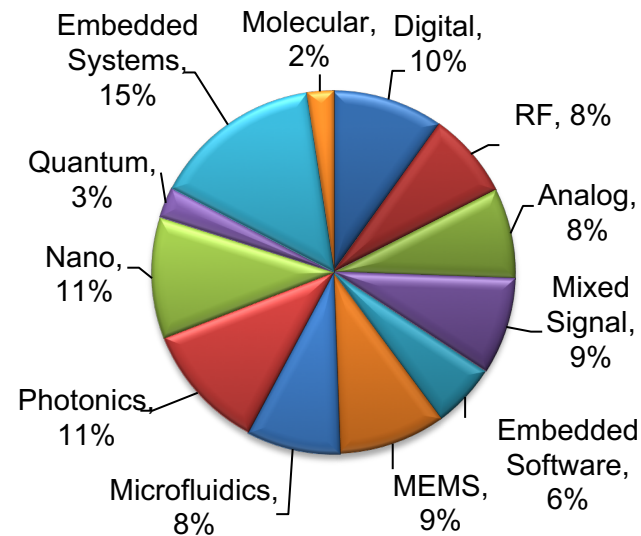


Technology Application



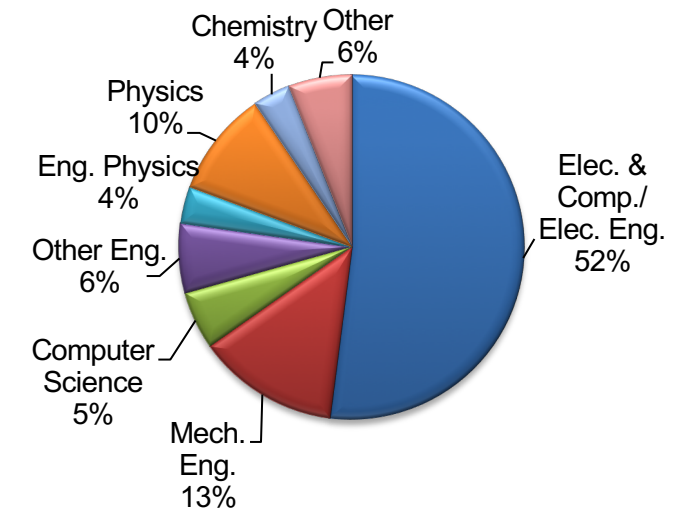
of professors = 1117
of interests = 3079

Design-Oriented Interests



of professors = 1126
of interests = 4513

Disciplines/Departments



of professors = 1209

CADnet: Canadian Design Network for Circuits and Systems



- \$20M infrastructure project targeting IF 2020 competition to extend access to CAD tools in the 2021-2025
 - 40 participating institutions, 750+ faculty
- Key infrastructure:
 - CAD tools
 - Centralized servers (Canada)
 - Next-generation (access via equipment loan)
 - Design
 - F (centralized access)
- NOI (September 2019)
- Proposal submission: January 2020

<https://community.cmc.ca/community/infrastructure-fund-project>

Agenda

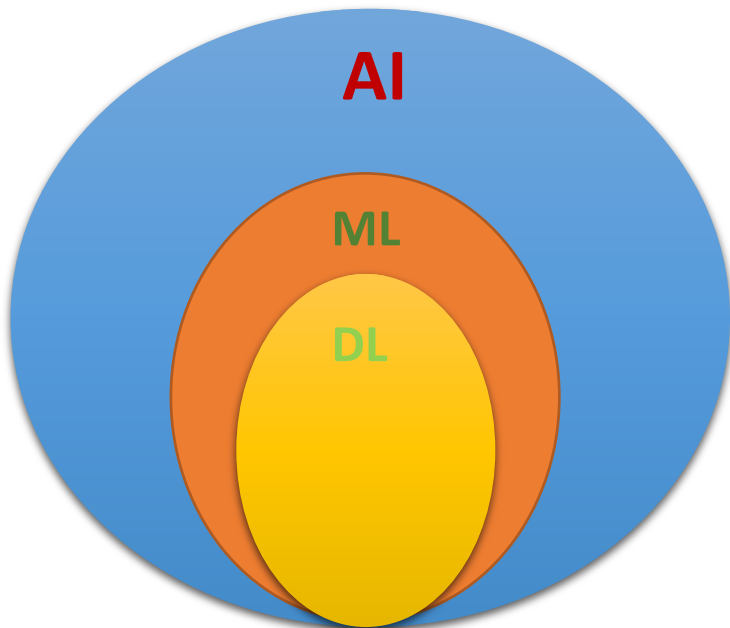
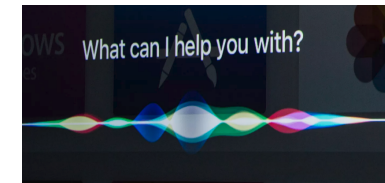


- CMC Microsystems
- **Overview**
- Hardware and software environment
- CNN Training and Quantization Flow
- Inference Demonstration
- How to access
- Q&A

AI and Machine Learning



- Machine learning is programming computers to optimize a performance criterion using example data or past experience
- Transforming many industries
- Exploding ecosystem of tools and platforms

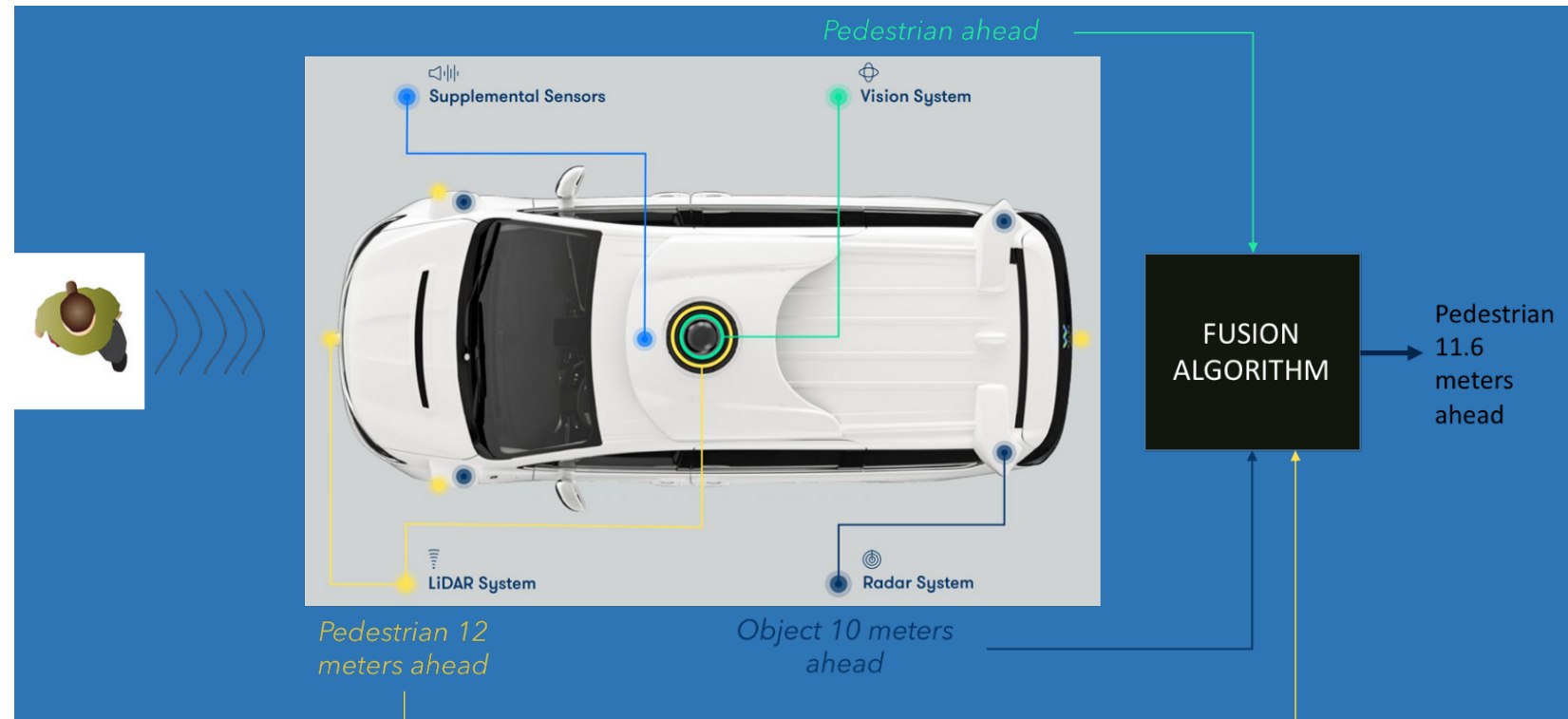


- **AI: Artificial Intelligence**
 - Sense, reason, act and adapt
- **ML: Machine Learning**
 - Algorithm that improve as they are exposed to data over time
- **DL: Deep Learning**
 - Multilayered neural networks learn from vast amounts of data

Source: What's the Difference Between Artificial Intelligence (AI), Machine Learning, and Deep Learning?
by [Glenn Evan Touger](#)

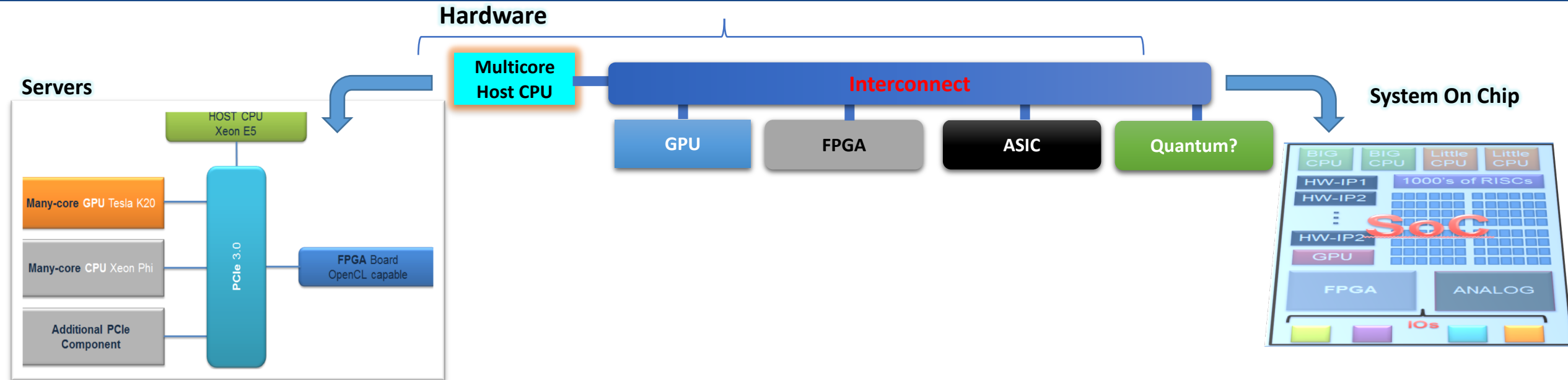
Sensor Fusion, Edge AI

- AI/ML processing increasingly moving to the edge, close to the sensors:
 - Sensor fusion
 - Low latency, fast response
 - Power-constrained
 - Harsh environment



Source: <https://towardsdatascience.com/sensor-fusion-90135614fde6>

Heterogeneous Systems Architecture



Deep Learning on the HPP/HCC:

- Objective: Accelerating Deep Learning on:
 - Programmable logic (FPGAs)
 - GPGPUs

Deep Learning on the ZCU102:

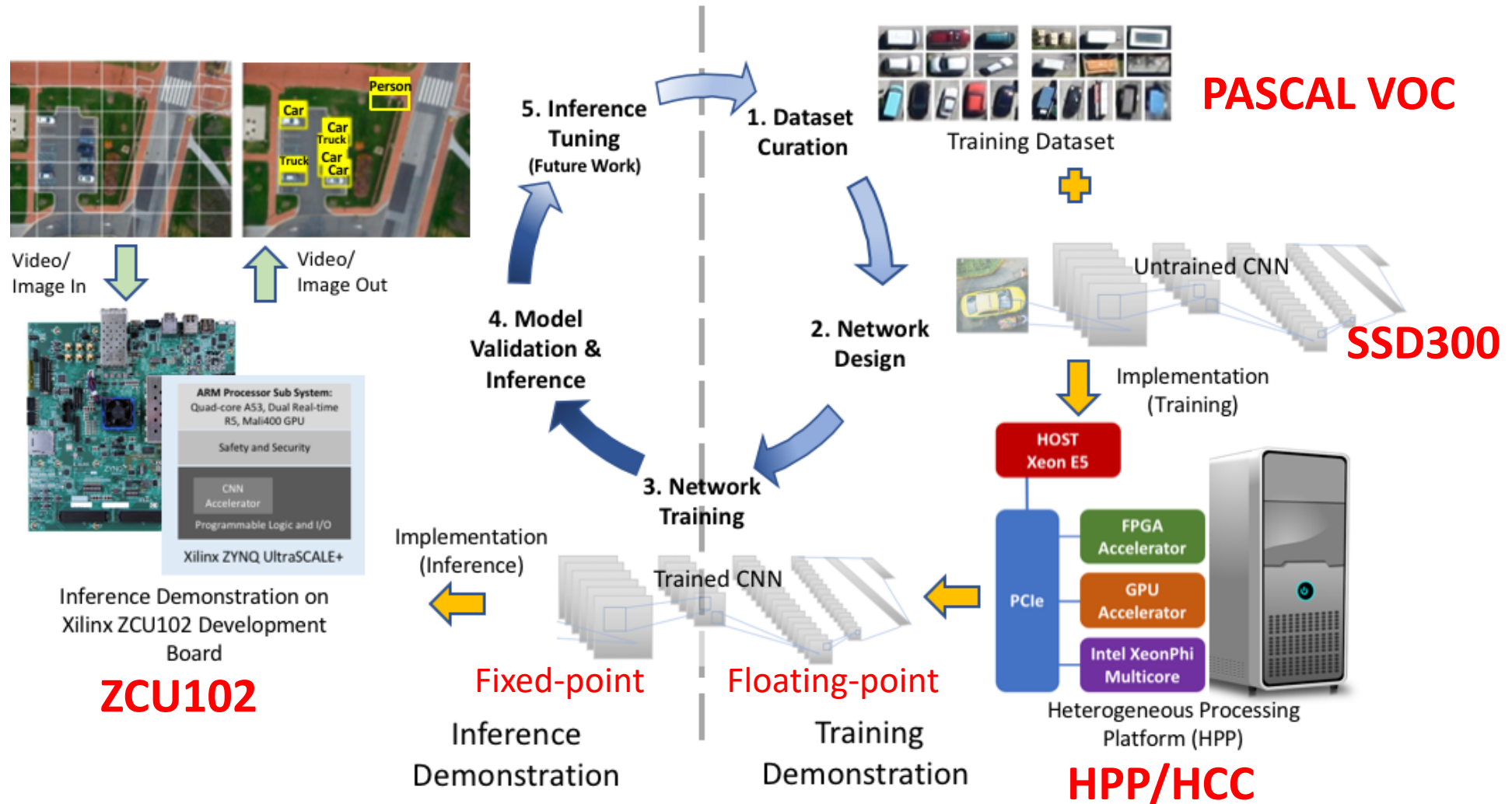
- Objective: Accelerating Inference:
 - Multicore ARM processors
 - Programmable logic (FPGAs)
 - Embedded GPU, real-time processor

Advantages of Reconfigurable, Heterogeneous Computing

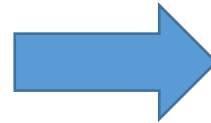


- Flexibility to target parts of algorithm in software and parts in reconfigurable hardware or specialized accelerators to achieve performance speedups, low latency and power efficiency while maintaining software programmability
- Scale-up/down to meet higher performance or lower power requirements by selecting same family of chips with more/fewer resources while maintaining the same design/architecture
- In-field modifications with no changes to equipment:
 - New network models
 - New applications
 - Better-trained models
 - Continuous learning

Convolutional Neural Network (CNN) Training and Inference Development Flow

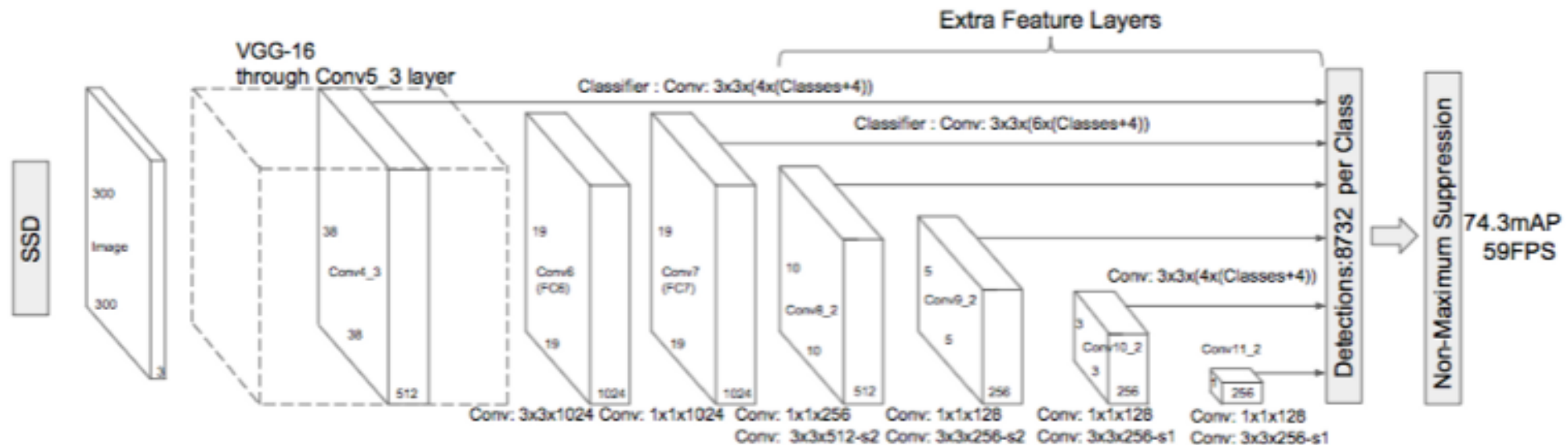


- Video-based object classification and detection:
 - Video input (e.g., camera)
 - Identify multiple objects in each from a library of classes
 - Mark each detected object with a colour-coded bounding box
 - Output processed frames with bounding boxes



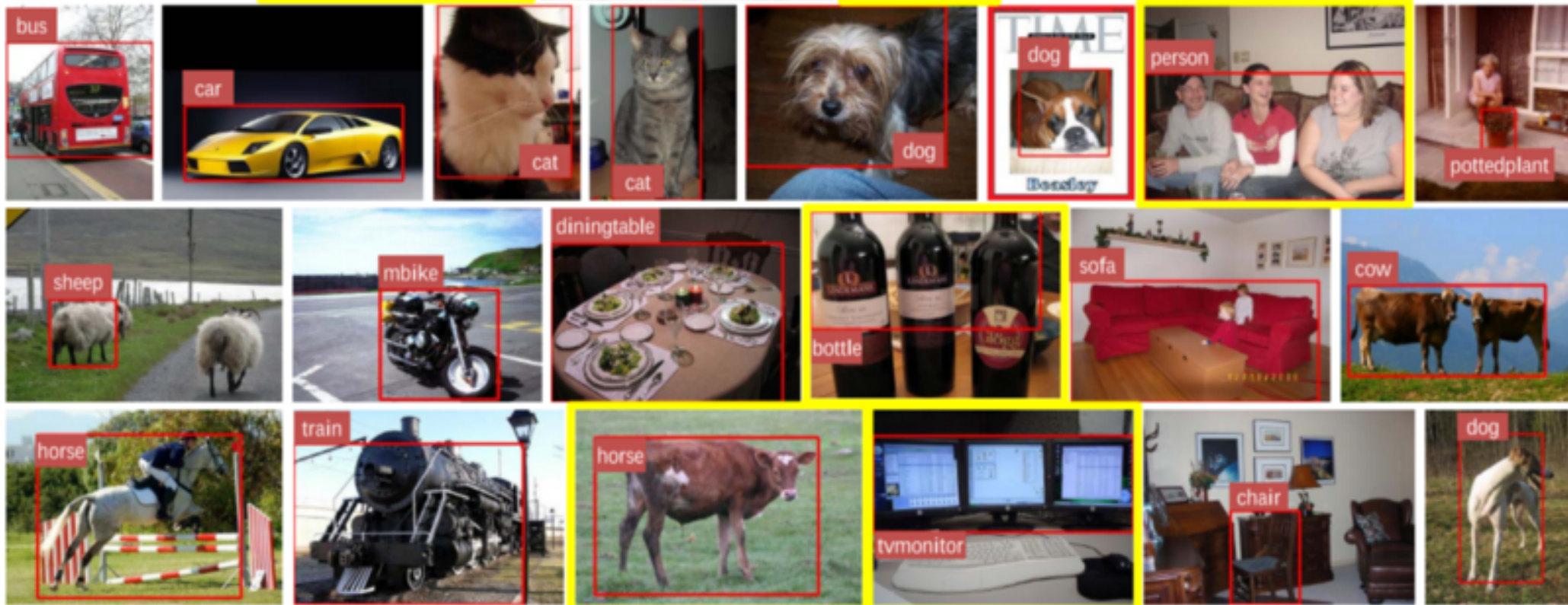
SSD: Single Shot MultiBox Detector

- SSD takes one single forward pass through the network to detect multiple objects within an image
- For object detection, outputs classification ID/confidence and location coordinates for a bounding box containing the object
- <https://towardsdatascience.com/understanding-ssd-multibox-real-time-object-detection-in-deep-learning-495ef744fab>



PASCAL Visual Object Classes (VOC) Dataset

- <http://host.robots.ox.ac.uk/pascal/VOC/>
- Standardised image data sets for object class recognition
- >20,000 images containing >45,000 annotated objects



PASCAL VOC Classes



- | | |
|---------------|------------------|
| 1. Background | 12. Dining Table |
| 2. Aeroplane | 13. Dog |
| 3. Bicycle | 14. Horse |
| 4. Bird | 15. Motorbike |
| 5. Boat | 16. Person |
| 6. Bottle | 17. Potted Plant |
| 7. Bus | 18. Sheep |
| 8. Car | 19. Sofa |
| 9. Cat | 20. Train |
| 10. Chair | 21. TV Monitor |
| 11. Cow | |

Agenda



- CMC Microsystems
- Overview
- **Hardware and software environment**
- CNN Training and Quantization Flow
- Inference Demonstration
- How to access
- Q&A

- Training:
 - Synodic workstation, 2x Intel E5-2630v2 CPU, NVIDIA Tesla K40, Ubuntu 16.04
 - Colfax ProEdge SXT9700, 2x Intel Xeon Bronze 3104 CPU, NVIDIA Tesla Pascal P100, Ubuntu 18.04
 - *Coming soon: CMC Heterogeneous Compute Cluster (HCC)*
- Inference:
 - Xilinx Zynq Ultrascale+ MPSoC ZCU102 Development Kit
 - Leopard Imaging LI-IMX274MIPI-FMC camera
 - HDMI monitor

- Xilinx DNNDK (Deep Neural Network Development Kit) v2.08
 - SDSoC 2018.3
 - DNNDK for SDSoC
 - ZCU102 SDSoC Revision Stack for DNNDK
- Caffe v1.0
- CUDA v8.0
- CuDNN v7.0.5
- NCCL v1.2.3

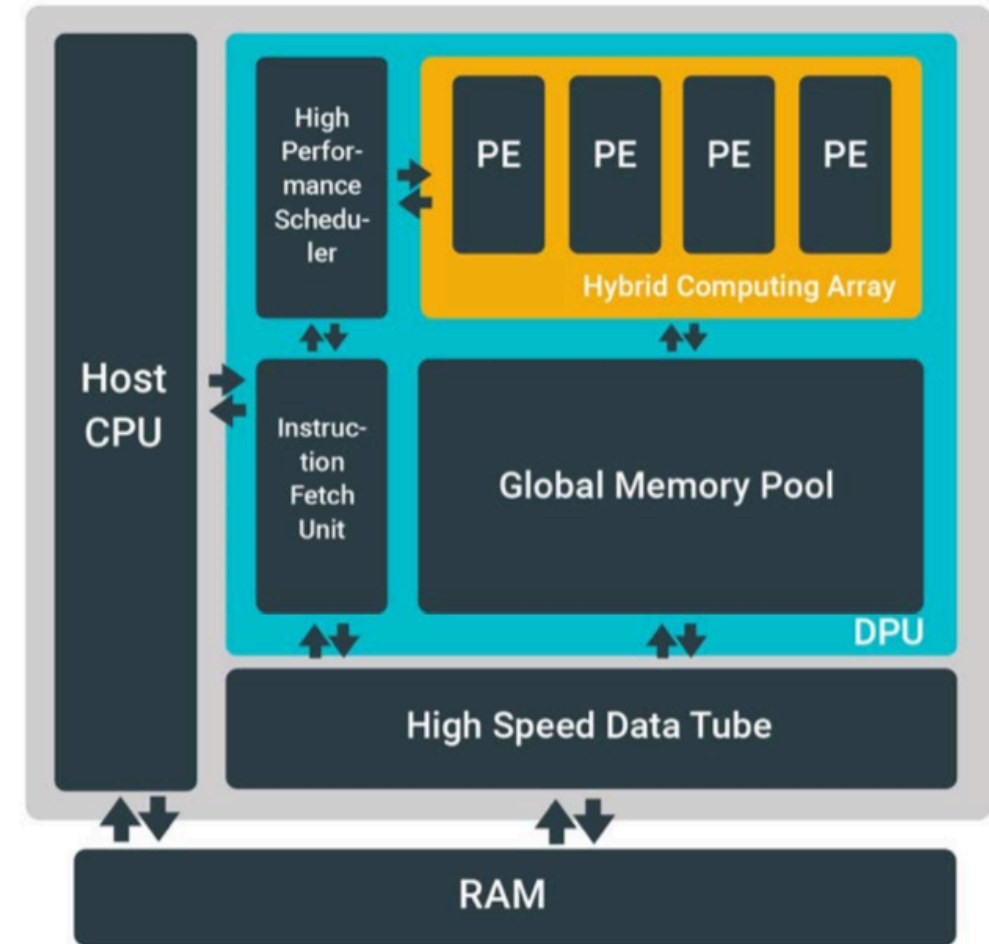
<https://github.com/Xilinx/Edge-AI-Platform-Tutorials/tree/master/docs/ML-SSD-PASCAL>

- Full-stack SDK for the Deep-learning Processor Unit (DPU)
 - Supports CNN quantization, compilation, optimization and runtime support
 - Network pruning supported by separate license
 - Supports Caffe framework
- Freely downloaded from Xilinx (registration required)
- Compatible with existing Xilinx tools/flows (Vivado, SDSoC)
- Supported evaluation boards:
 - **ZCU102**
 - ZCU104
 - Ultra96
- DNNDK v3.0 – support for TensorFlow added

Xilinx Deep-learning Processor Unit (DPU)



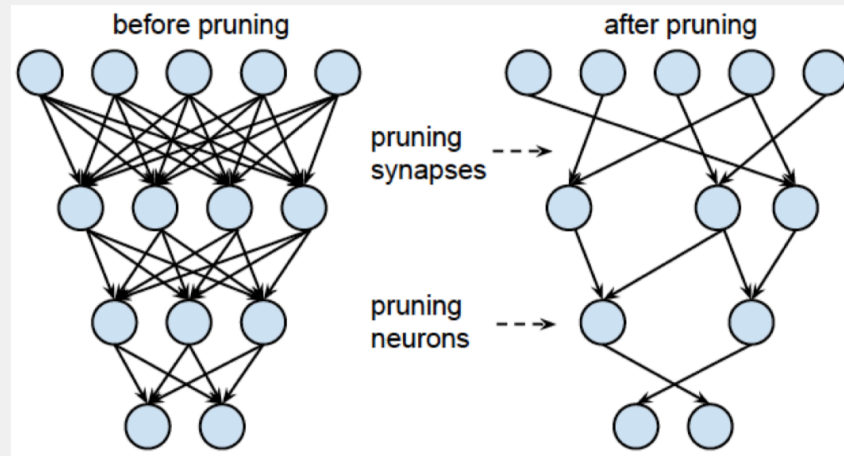
- Co-processor/overlay for Zynq embedded ARM cores
- Supports commonly used network layers, using hardware acceleration from the underlying FPGA architecture
- DPU hardware generated from SDSoC project
- Supports multi-threading, up to 4 DPU core on chip (limited by available FPGA resources)



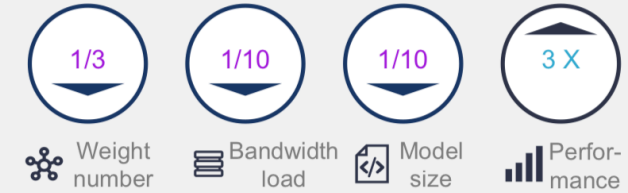
Courtesy: Xilinx Inc.

DNNDK: network pruning

Deep compression
Makes algorithm smaller and lighter



Highlight



Compression efficiency

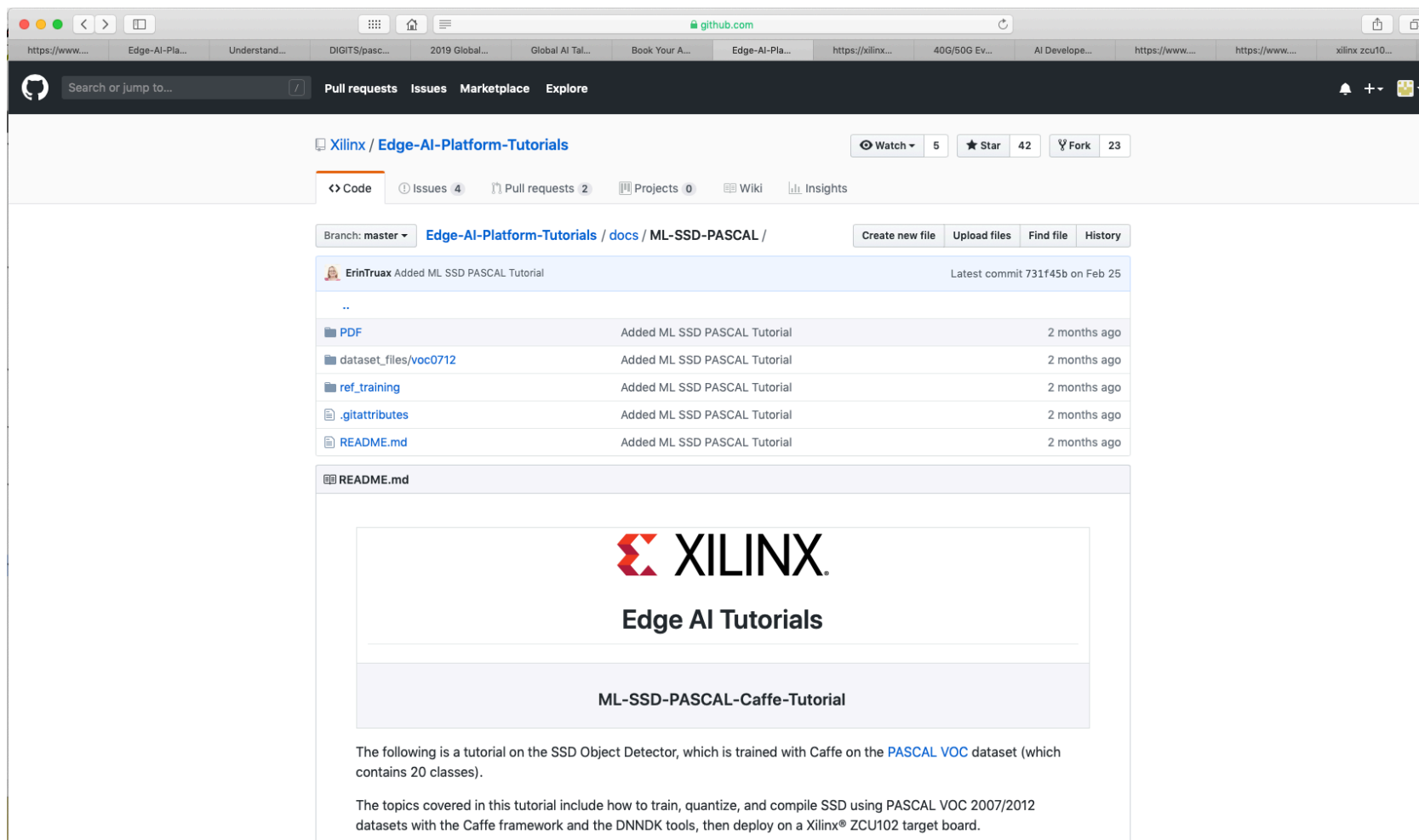
Deep Compression Tool can achieve significant compression on **CNN** and **RNN**

Accuracy

Algorithm can be **compressed 7 times without losing accuracy** under SSD object detection framework

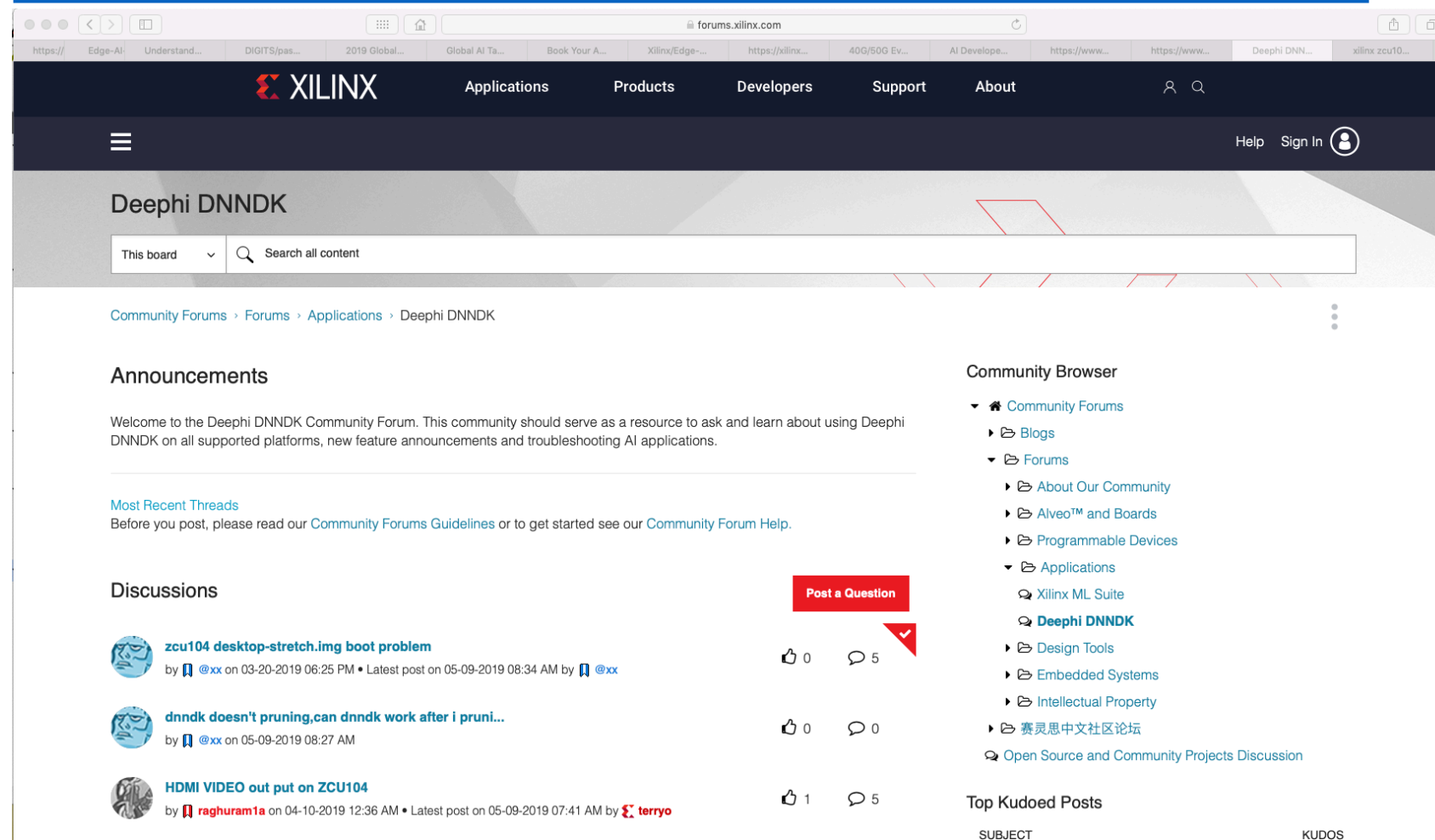
Source: <https://www.xilinx.com/publications/events/developer-forum/2018-frankfurt/machine-learning-for-embedded-deep-dive.pdf>

<https://github.com/Xilinx/Edge-AI-Platform-Tutorials>



The screenshot shows the GitHub repository page for Xilinx/Edge-AI-Platform-Tutorials. The repository is on the master branch and contains a folder named 'docs' with a subfolder 'ML-SSD-PASCAL'. The commit history shows that ErinTruax added the ML SSD PASCAL Tutorial on Feb 25. The README.md file is displayed, featuring the Xilinx logo and the title 'Edge AI Tutorials'. The tutorial is titled 'ML-SSD-PASCAL-Caffe-Tutorial' and describes a tutorial on the SSD Object Detector, which is trained with Caffe on the PASCAL VOC dataset (which contains 20 classes). The topics covered in this tutorial include how to train, quantize, and compile SSD using PASCAL VOC 2007/2012 datasets with the Caffe framework and the DNNDK tools, then deploy on a Xilinx® ZCU102 target board.

<https://forums.xilinx.com/t5/Deepphi-DNNDK/bd-p/Deepphi>



The screenshot shows the Xilinx forums website. The top navigation bar includes the Xilinx logo and links for Applications, Products, Developers, Support, and About. Below this is a search bar and a 'Sign In' button. The main heading is 'Deepphi DNNDK'. A search bar is present with the text 'Search all content'. The page is divided into three main sections: Announcements, Discussions, and a Community Browser. The Announcements section contains a welcome message. The Discussions section lists three threads: 'zcu104 desktop-stretch.img boot problem', 'dnndk doesn't pruning, can dnndk work after i pruni...', and 'HDMI VIDEO out put on ZCU104'. The Community Browser section lists various categories like Community Forums, Blogs, Forums, Applications, Design Tools, and Embedded Systems. A 'Post a Question' button is visible in the Discussions section.

Announcements

Welcome to the Deepphi DNNDK Community Forum. This community should serve as a resource to ask and learn about using Deepphi DNNDK on all supported platforms, new feature announcements and troubleshooting AI applications.

Most Recent Threads

Before you post, please read our [Community Forums Guidelines](#) or to get started see our [Community Forum Help](#).

Discussions

zcu104 desktop-stretch.img boot problem
by [@xx](#) on 03-20-2019 06:25 PM • Latest post on 05-09-2019 08:34 AM by [@xx](#)

dnndk doesn't pruning, can dnndk work after i pruni...
by [@xx](#) on 05-09-2019 08:27 AM

HDMI VIDEO out put on ZCU104
by [raghuram1a](#) on 04-10-2019 12:36 AM • Latest post on 05-09-2019 07:41 AM by [terry](#)

Community Browser

- Community Forums
 - Blogs
 - Forums
 - About Our Community
 - Alveo™ and Boards
 - Programmable Devices
 - Applications
 - Xilinx ML Suite
 - Deepphi DNNDK**
 - Design Tools
 - Embedded Systems
 - Intellectual Property
 - 赛灵思中文社区论坛
 - Open Source and Community Projects Discussion

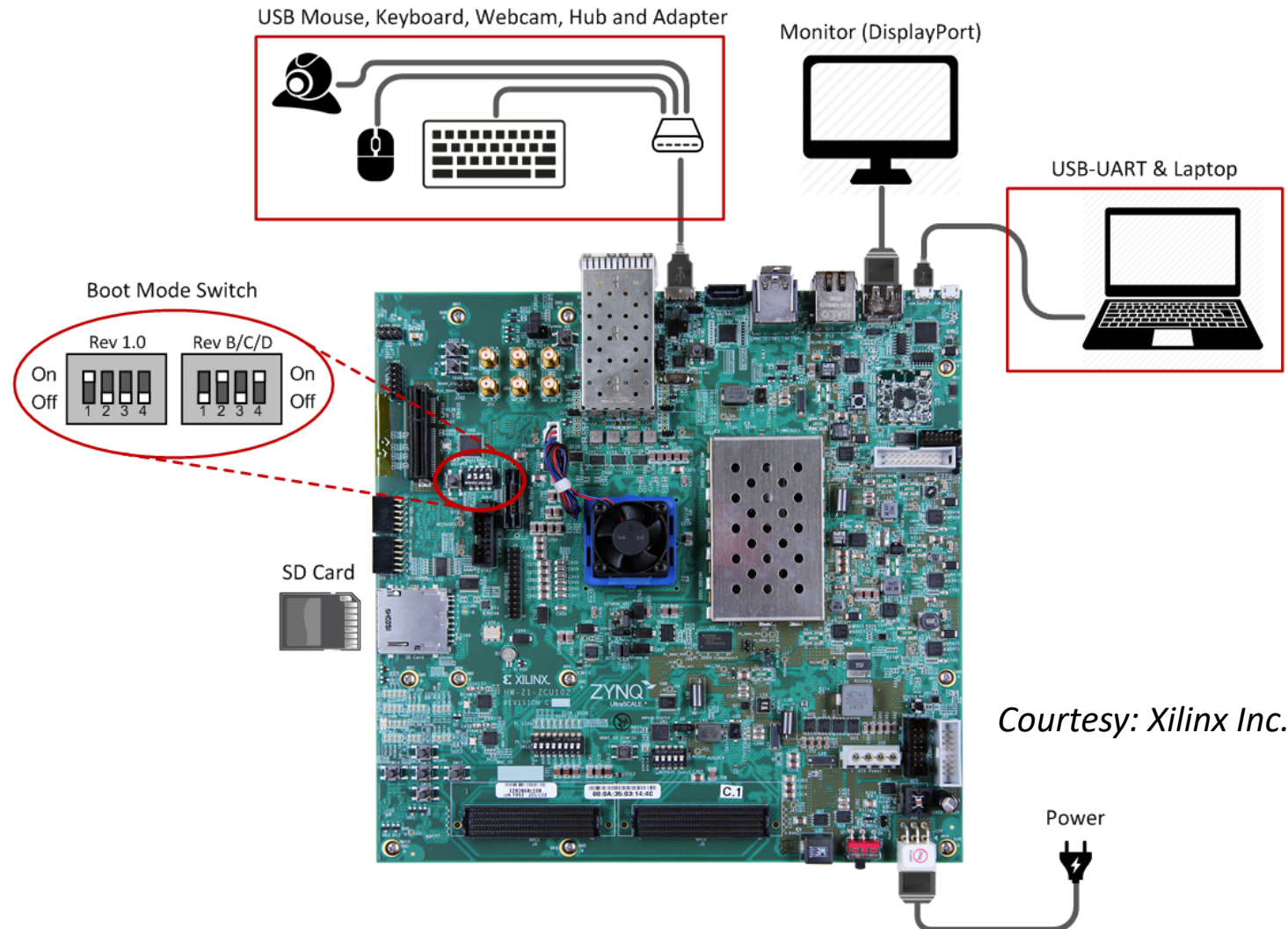
Top Kudoed Posts

SUBJECT	KUDOS

Xilinx ZCU102 Features



- Xilinx Zynq Ultrascale+ MPSoC (ZU9EG)
 - Quad-core ARM A53
 - Dual-core ARM R5
 - ARM GPU
 - 16nm FinFET+ programmable logic
- 4GB 64-bit DDR4 (processor)
- 512MB 16-bit DDR4 (FPGA)
- 2x FMC-HPC connectors
- HDMI video input and output
- DisplayPort video output
- SD Card
- Push buttons, DIP switches, LEDs
- USB UART
- Ethernet

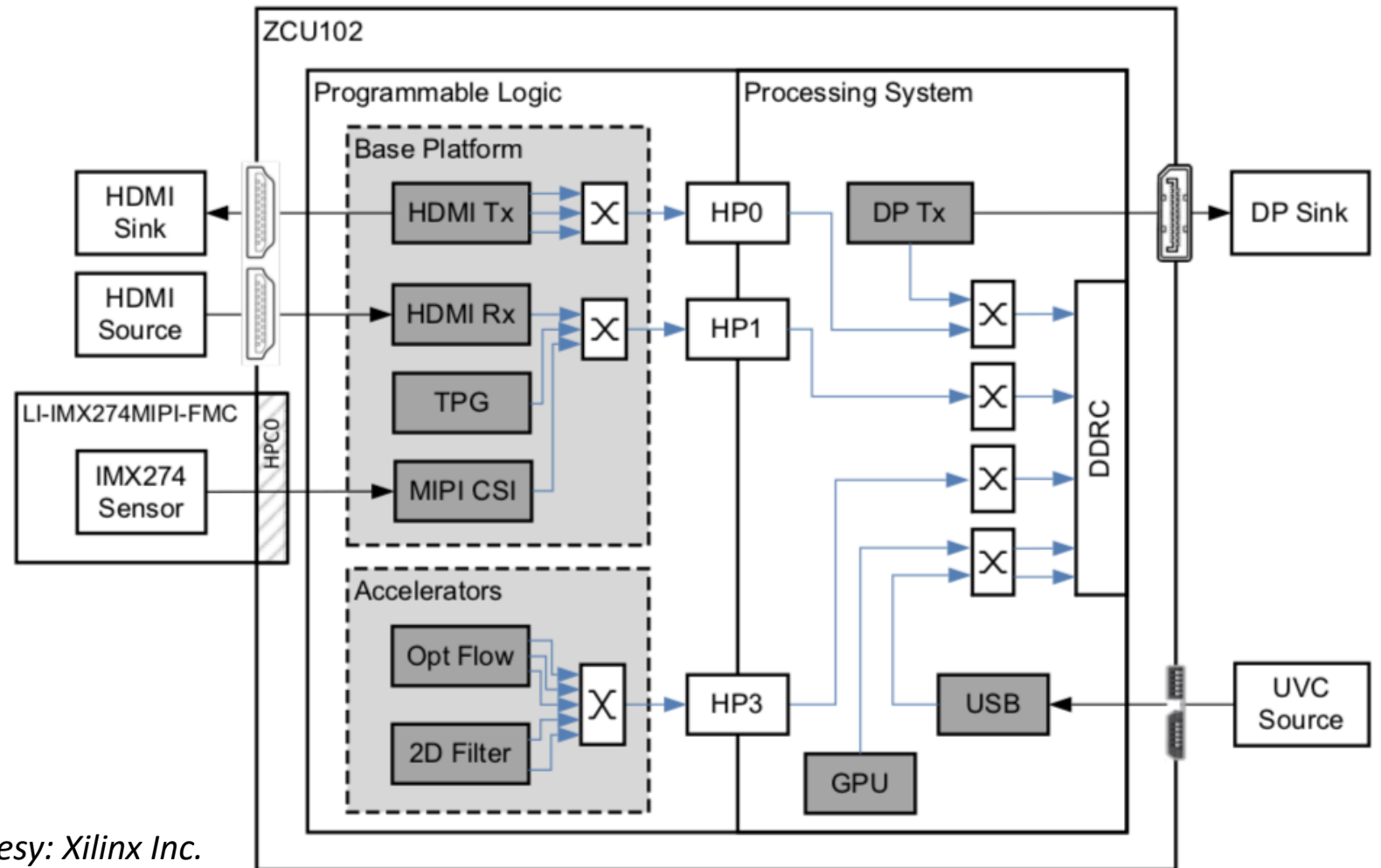


Xilinx ReVISION Stack

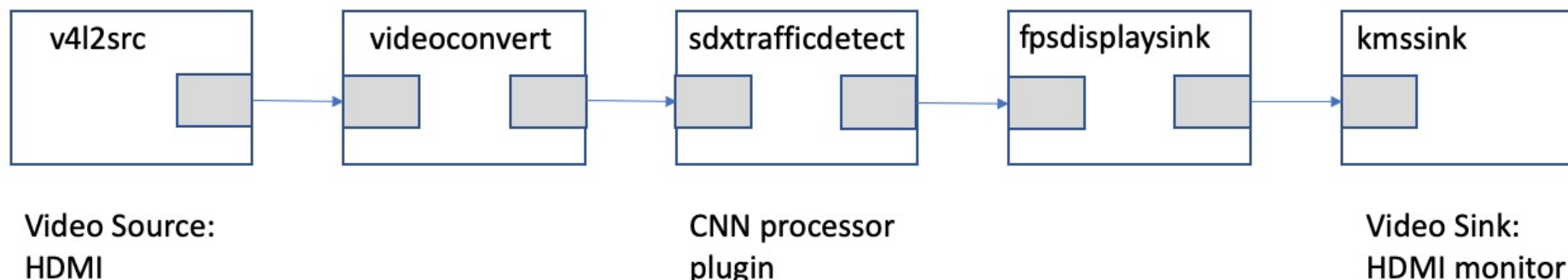


- Pre-built platform for algorithm and application development for embedded vision on Xilinx boards (e.g., ZCU102)
- Video capture and sink pipelines
- xfOpenCV library: acceleration-ready OpenCV functions
- PetaLinux BSP
- Design examples (machine vision, CNN)

Courtesy: Xilinx Inc.



- Modular, open framework for creating streaming multimedia applications
- Individual processing elements (sources, sinks, filters) are called *plugins*
- In an application, plugins are linked and arranged into *pipelines*
- Pipelines can be constructed/executed within application (e.g., C/C++, Python) at the command line or through *gst-launch-1.0*
- Large library of plugins available (good, bad, ugly)
- Supported in Xilinx reVISION Stack Linux kernel



<https://gstreamer.freedesktop.org>

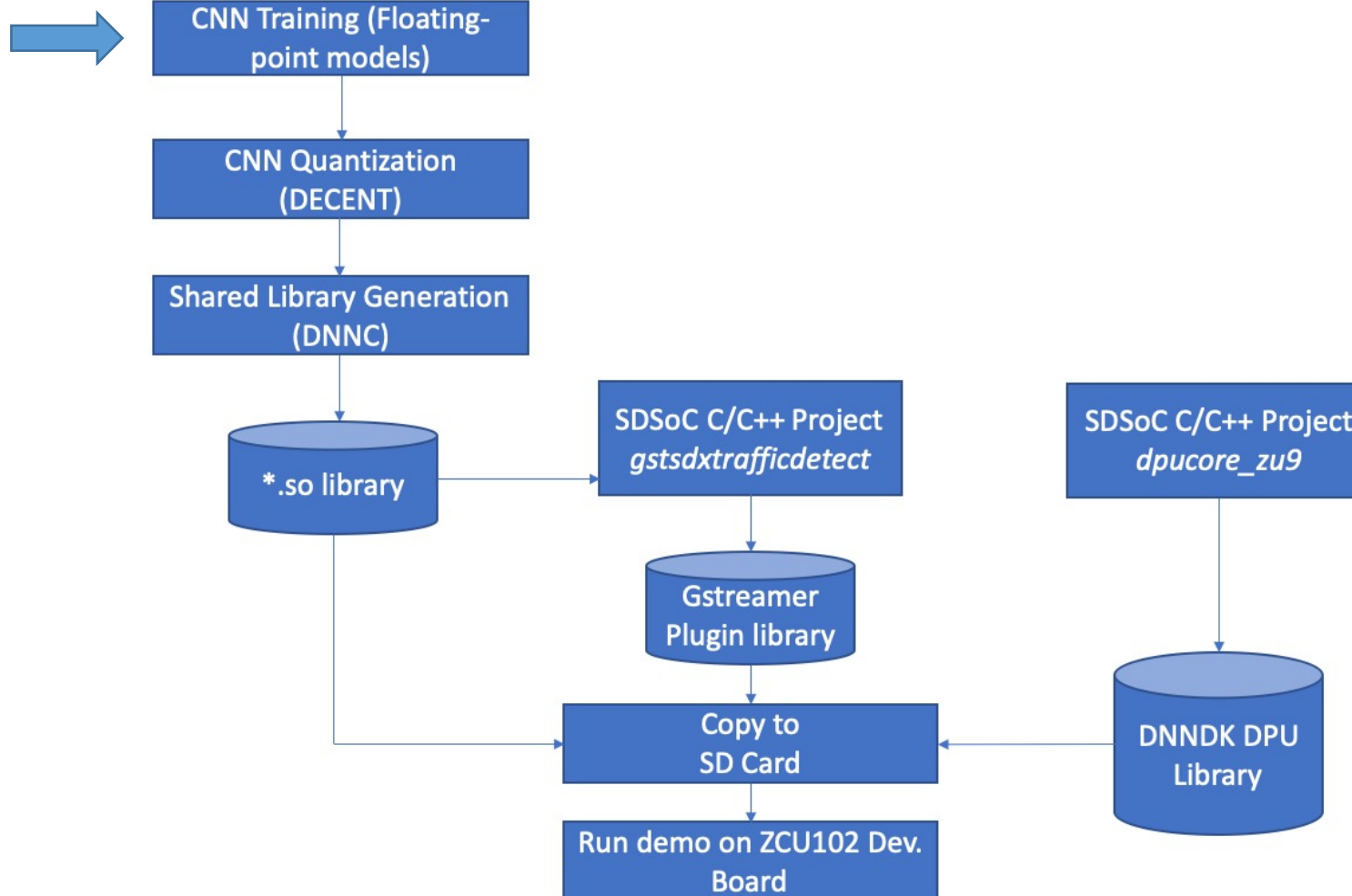
Agenda



- CMC Microsystems
- Overview
- Hardware and software environment
- **CNN Training and Quantization Flow**
- Inference Demonstration
- How to access
- Q&A

Inference Development Flow

- Dataset
- Untrained CNN



Step 1: Install the Caffe tool for SSD



- Nvidia libraries/drivers
- CUDA v8.0
- CuDNN v7.0.5
- NCCL v1.2.3
- SSD Caffe

Step 2: Prepare the dataset and database



- Download pre-trained VGG network files

VGG_ILSVRC_16_layers_fc_reduced_deploy.prototxt (description)

VGG_ILSVRC_16_layers_fc_reduced.caffemodel (weights)

- Download PASCAL VOC dataset

http://host.robots.ox.ac.uk/pascal/VOC/voc2012/VOCtrainval_11-May-2012.tar

http://host.robots.ox.ac.uk/pascal/VOC/voc2007/VOCtrainval_06-Nov-2007.tar

http://host.robots.ox.ac.uk/pascal/VOC/voc2007/VOCtest_06-Nov-2007.tar

- Create training validation database and test database files

create_list.sh → VOC0712_test_lmdb/data.mdb

create_data.sh → VOC0712_trainval_lmdb/data.mdb

- Execute python script to add SSD framework layers to VGGNet

ssd_pascal.py → solver.prototxt, deploy.prototxt, test.prototxt, train.prototxt

Step 3: Train the SSD Network



- Modify SSD Prototxt files for compatibility with DPU/DNNDK
- Run training script (will execute `caffe train`):

```
$CAFFE_ROOT/jobs/VGGNet/VOC0912/SSD_300x300/VGG_VOC0912_SSD_300x300.sh
```

Runs 120,000 training iterations:


- Synodic Tower with NVIDIA Tesla K40: 6 days
- Colfax ProEdge SXT9700 with NVIDIA Tesla Pascal P100: 2 days

```
ideasuser@ideasubuntu16:~/Caffe-SSD/caffe-ssd/models/VGGNet/VOC0712/SSD_300x300$ ls -al
I0422 14:22:34.587582 27953 solver.cpp:259] Train net output #0: mbox_loss = 2.31304 (* 1 = 2.31304 loss)
I0422 14:22:34.587594 27953 sgd_solver.cpp:138] Iteration 119990, lr = 1e-05
I0422 14:23:14.649554 27953 solver.cpp:596] Snapshotting to binary proto file models/VGGNet/VOC0712/SSD_300x3
00/VGG_VOC0712_SSD_300x300_iter_120000.caffemodel
I0422 14:23:14.991468 27953 sgd_solver.cpp:307] Snapshotting solver state to binary proto file models/VGGNet/
VOC0712/SSD_300x300/VGG_VOC0712_SSD_300x300_iter_120000.solverstate
I0422 14:23:15.576786 27953 solver.cpp:332] Iteration 120000, loss = 2.33714
I0422 14:23:15.576824 27953 solver.cpp:433] Iteration 120000, Testing net (#0)
I0422 14:23:15.576895 27953 net.cpp:693] Ignoring source layer mbox_loss
I0422 14:27:30.300951 27953 solver.cpp:546] Test net output #0: detection_eval = 0.762029
I0422 14:27:30.301141 27953 solver.cpp:337] Optimization Done.
I0422 14:27:30.301154 27953 caffe.cpp:254] Optimization Done.
ideasuser@ideasubuntu16:~/Caffe-SSD/caffe-ssd/jobs/VGGNet/VOC0712/SSD_300x300$
-rw-rw-r-- 1 ideasuser ideasuser 97443334 Apr 20 13:27 VGG_VOC0712_SSD_300x300_iter_80000.solverstate
ideasuser@ideasubuntu16:~/Caffe-SSD/caffe-ssd/models/VGGNet/VOC0712/SSD_300x300$
```

Step 4: Evaluate the Floating Point Network

Command: `$CAFFE_DIR/evaluation/score.sh`

```
Ideasuser@ideasubuntu16: ~/Caffe-SSD/caffe-ssd
I0509 14:03:10.999877 27352 net.cpp:228] relu3_1 does not need backward computation.
I0509 14:03:10.999883 27352 net.cpp:228] conv3_1 does not need backward computation.
I0509 14:03:10.999891 27352 net.cpp:228] pool2 does not need backward computation.
I0509 14:03:10.999897 27352 net.cpp:228] relu2_2 does not need backward computation.
I0509 14:03:10.999903 27352 net.cpp:228] conv2_2 does not need backward computation.
I0509 14:03:10.999910 27352 net.cpp:228] relu2_1 does not need backward computation.
I0509 14:03:10.999917 27352 net.cpp:228] conv2_1 does not need backward computation.
I0509 14:03:10.999923 27352 net.cpp:228] pool1 does not need backward computation.
I0509 14:03:10.999930 27352 net.cpp:228] relu1_2 does not need backward computation.
I0509 14:03:10.999938 27352 net.cpp:228] conv1_2 does not need backward computation.
I0509 14:03:10.999943 27352 net.cpp:228] relu1_1 does not need backward computation.
I0509 14:03:10.999950 27352 net.cpp:228] conv1_1 does not need backward computation.
I0509 14:03:10.999958 27352 net.cpp:228] data_data_0 does not need backward computation.
I0509 14:03:10.999965 27352 net.cpp:228] data does not need backward computation.
I0509 14:03:10.999972 27352 net.cpp:270] This network is not initialized.
I0509 14:03:11.000078 27352 net.cpp:283] Network initialization done.
Could not create logging file: No such file or directory
COULD NOT CREATE A LOGGINGFILE 20190509-140311.27352!
Loading done.
I0509 14:03:11.006572 27352 caffe.cpp:155] Finetuning 300x300_iter_120000.caffemodel
I0509 14:03:11.083011 27352 upgrade_proto.cpp:77] Attempting to upgrade models/VGGNet/VOC0712/SSD_300x300/VGG_VOC0712_SSD_300x300_iter_120000.caffemodel to version 1.
I0509 14:03:11.083052 27352 upgrade_proto.cpp:80] Successfully upgraded model.
I0509 14:03:11.157622 27352 upgrade_proto.cpp:77] Attempting to upgrade models/VGGNet/VOC0712/SSD_300x300/VGG_VOC0712_SSD_300x300_iter_120000.caffemodel to version 1.
I0509 14:03:11.157660 27352 upgrade_proto.cpp:80] Successfully upgraded model.
I0509 14:03:11.180361 27352 net.cpp:761] Ignoring source layer data_0.
I0509 14:03:11.180789 27352 caffe.cpp:251] Starting Optimization.
I0509 14:03:11.180804 27352 solver.cpp:294] Solving VGGNet/VOC0712/SSD_300x300/VGG_VOC0712_SSD_300x300_iter_120000.caffemodel.
I0509 14:03:11.180811 27352 solver.cpp:295] Learning Rate: 0.001.
I0509 14:03:11.622129 27352 solver.cpp:332] Iteration 1000.
I0509 14:03:11.622176 27352 solver.cpp:433] Iteration 1000.
I0509 14:03:11.639858 27352 net.cpp:693] Ignoring source layer data_0.
I0509 14:07:25.647696 27352 solver.cpp:546] Test set accuracy: 0.99757165.
I0509 14:07:25.647825 27352 solver.cpp:337] Optimization Done.
I0509 14:07:25.647835 27352 caffe.cpp:254] Optimization Done.
Ideasuser@ideasubuntu16:~/Caffe-SSD/caffe-ssd$
```



Step 5: Quantize the SSD Network with DECENT



- Input files: float.caffemodel, float.prototxt, calibration dataset (100-1000 images)
- Output files: deploy.caffemodel, deploy.prototxt
- Default bitwidth: 8 (currently only supported by DPU)
- Command:

```
decent quantize \
    -model ${model_dir}/float.prototxt \
    -weights ${model_dir} float.caffemodel \
    -output_dir ${output_dir} -gpu 0 -auto_test
```

```
I0509 13:18:13.569810 26696 net_test.cpp:207] Test iter: 46/50
I0509 13:18:13.978898 26696 net_test.cpp:207] Test iter: 47/50
I0509 13:18:14.402451 26696 net_test.cpp:207] Test iter: 48/50
I0509 13:18:14.814522 26696 net_test.cpp:207] Test iter: 49/50
I0509 13:18:15.209785 26696 net_test.cpp:207] Test iter: 50/50
I0509 13:18:15.218399 26696 net_test.cpp:254] Test Results:
I0509 13:18:15.218410 26696 net_test.cpp:255] Test net output #0: detection_eval = 0.790698
I0509 13:18:15.218444 26696 net_test.cpp:387] Test Done!
I0509 13:18:15.574645 26696 decent.cpp:333] Start Deploy
I0509 13:18:15.913386 26696 decent.cpp:341] Deploy Done!
-----
Output Deploy Weights: "/home/ideasuser/Caffe-SSD/caffe-ssd/DNNDK_Project/decent_output/deploy.caffemodel"
Output Deploy Model:  "/home/ideasuser/Caffe-SSD/caffe-ssd/DNNDK_Project/decent_output/deploy.prototxt"
ideasuser@ideasubuntu16:~/Caffe-SSD/caffe-ssd/DNNDK_Project$
```

Step 6: Compile the SSD Network with DNNC



- Output file: dpu_ssd.elf

- Command:

```
dnnc --prototxt=${model_dir}/deploy.prototxt \
     --caffemodel=${model_dir}/deploy.caffemodel \
     --net_name=ssd \
     --dpu=4096FA \
     --cpu_arch=arm64 \
     --abi=0
```

```
DNNC Kernel Information
1. Overview
kernel numbers : 1
kernel topology : ssd_kernel_graph.jpg
2. Kernel Description in Detail
kernel id      : 0
kernel name    : ssd
type          : DPUKernel
nodes         : NA
input node(s) : conv1_1(0)
output node(s) : mbox_loc(0) mbox_conf(0)
ideasuser@ideasubuntu16:~/Caffe-SSD/caffe-ssd/DNNDK_Project$
```

Step 7: Compile .elf to shared library



- Input file: dpu_ssd.elf
- Output file: libdpumodelssd.so
- Command:

```
aarch64-linux-gnu-gcc -fPIC -shared dpu_ssd.elf -o libdpumodelssd.so
```

```
ideasuser@ideasubuntu16:~/Caffe-SSD/caffe-ssd/DNNDK_Project/dnnc_output$ aarch64-linux-gnu-gcc -fPIC -shared  
dpu_ssd.elf -o libdpumodelssd.so  
ideasuser@ideasubuntu16:~/Caffe-SSD/caffe-ssd/DNNDK_Project/dnnc_output$ ls -al  
total 48528  
drwx----- 2 ideasuser ideasuser    4096 May  9 13:33 .  
drwxrwxr-x 6 ideasuser ideasuser    4096 May  9 13:17 ..  
-rw-rw-r-- 1 ideasuser ideasuser 24834008 May  9 13:25 dpu_ssd.elf  
-rwxrwxr-x 1 ideasuser ideasuser 24850256 May  9 13:33 libdpumodelssd.so  
ideasuser@ideasubuntu16:~/Caffe-SSD/caffe-ssd/DNNDK_Project/dnnc_output$
```

Step 8: Build Hardware and Application Projects in the SDSoC Development Environment



workspace - dpucore_zcu102/project.sdx - Xilinx SDx

Project Explorer

- dpucore_zcu102
 - Binaries
 - Archives
 - Debug
 - lib
 - src
 - project.sdx
 - Readme.md
- gstsdxttraffictdetect
 - Binaries
 - Includes
 - Debug
 - src

SDx Application Project Settings

General

Project name: dpucore_zcu102

Project flow: SDSoc

Platform: zcu102_rv_ss

Runtime: C/C++

System configuration: A53 SMP Linux

Domain: A53 SMP Linux

OS: Linux

Options

Target: Hardware

☐ Estimate performance

☐ Enable event tracing

☐ Insert AXI performance monitor

Data motion network clock frequency (MHz): 299.97

Emulation model: Debug

☒ Generate SD card image

Root function: main

Hardware Functions

Name	Clock Frequency (MHz)	Path
dpu_cache_sync	299.97	src/dpustubs.cpp
dpu_memcpy	299.97	src/dpustubs.cpp
dpu_memset	299.97	src/dpustubs.cpp

Assistant

dpucore_zcu102 [SDSoC]

- Debug [Hardware]
- Release [Hardware]
- gstsdxttraffictdetect

Problems

SDx Build Console [dpucore_zcu102, Debug]

Software tracing enabled

Compile hardware access API functions

Link application ELF file

SD card folder created /eng/home/hugh/ENGDEV/xilinx/SDx_2018.2/DNNDK/xilinx_dnndk_v2.0

All user specified timing constraints are met.

sds++ log file saved as /eng/home/hugh/ENGDEV/xilinx/SDx_2018.2/DNNDK/xilinx_dnndk_v2.0

Finished building target: libdpucore_zcu102.so

14:00:39 Build Finished (took 3h:26m:2s.420ms)

Target Connections

- Hardware Server
- Linux TCF Agent
- QEMU TcfGdbClient

Implementation results: Xilinx Vivado



prj - [/eng/home/hugh/ENGDEV/xilinx/SDx_2018.2/DNNDK/xilinx_dnndk_v2.08_for_sdsoC/dnndk_ws/dpucore_zu9/Debug/_sds/p0/vivado/prj/prj.xpr] - Vivado 2018.3

File Edit Flow Tools Reports Window Layout View Help Q Quick Access

Synthesis and Implementation Out-of-date details

Default Layout

Flow Navigator

- PROJECT MANAGER
 - Settings
 - Add Sources
 - Language Templates
 - IP Catalog
- IP INTEGRATOR
 - Create Block Design
 - Open Block Design
 - Generate Block Design
- SIMULATION
 - Run Simulation
- RTL ANALYSIS
 - Open Elaborated Design
- SYNTHESIS
 - Run Synthesis
 - Open Synthesized Design
- IMPLEMENTATION
 - Run Implementation
 - Open Implemented Design
 - Constraints Wizard
 - Edit Timing Constraints
 - Report Timing Summary
 - Report Clock Networks
 - Report Clock Interaction
 - Report Methodology
 - Report DRC

IMPLEMENDED DESIGN - xczu9eg-ffvb1156-2-e (active)

Sources Netlist

- zcu102_rv_ss_wrapper
 - Nets (86)
 - Leaf Cells (20)
 - zcu102_rv_ss_i (zcu102_rv_ss)

Properties

Select an object to see properties

Project Summary Device

Step 7: Build the Application in the SDSoC Development Environment

Tcl Console Messages Log Reports Design Runs Power DRC Methodology Timing

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT %	FF %	BRAM %	URA ...	DSP %	Start	Elapsed	Run Strategy
synth_1 (active)	constrs_1	Synthesis Out-of-date								0.00	0.00	0.00	0.00	0.00	4/15/19, 11:02 AM	00:04:59	Vivado Synthesis Defaults
impl_1	constrs_1	Implementation Out-of-date	0.052	0.000	0.008	0.000	0.000	20.728	0	61.72	56.14	77.52	0.00	53.21	4/15/19, 11:23 AM	02:37:11	Congestion_SpreadLogic_high
Out-of-Context Module Runs																	
zcu102_rv_ss		Submodule Runs Out-of-date													4/15/19, 10:49 AM	00:33:15	

Step 9: Copy files to SD Card



- From DNNDK pre-built for ZCU102:

`video_cmd`

`libdputils.so→lib`

`libn2cube.so→lib`

- From DNNC output:

`libdpumodelssd.so→lib`

- From dpucore_zcu102 project:

`BOOT.BIN`

`image.ub`

`libdpucore.so→lib`

- From gststdxtraffictdetect project:

`libgststdxtraffictdetect.so→lib`

Step 10: Boot ZCU102



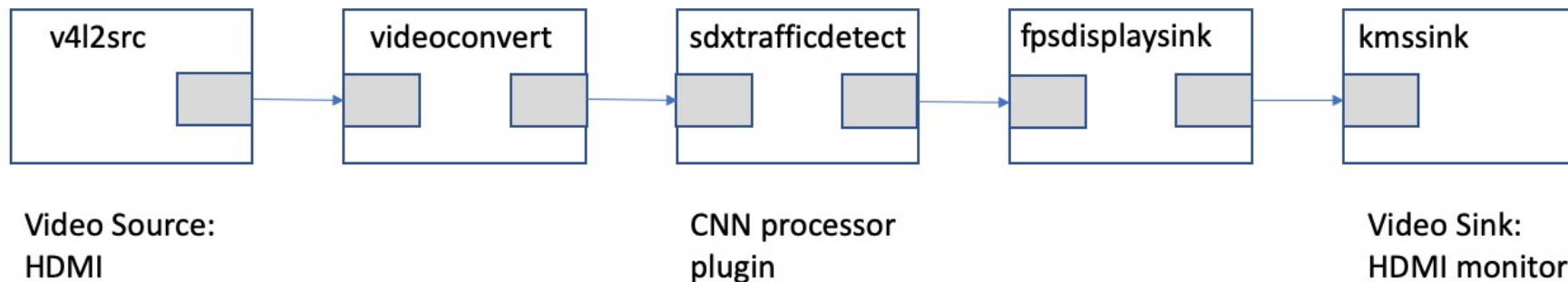
```
dev — screen /dev/tty.SLAB_USBtoUART 115200 • SCREEN — 138x47
[ 9.398413] xilinx-vphy a0000000.vphy: probed
[ 9.412404] VPhy version : 02.02 (0000)
[ 9.413486] dp159 3-005e: probe successful
[ 9.420766] xilinx-vphy a0000000.vphy: probe successful
[ 9.428894] xilinx-hdmi-rx a1000000.hdmi_rxss: probed
[ 9.434176] xvphy_phy_init(fffffc87b11f800).
[ 9.438593] xvphy_phy_init(fffffc87b19a000).
[ 9.443044] xvphy_phy_init(fffffc87bb21c00).
[ 9.455063] xilinx-hdmi-rx a1000000.hdmi_rxss: Direct firmware load for xilinx/xilinx-hdmi-rx-edid.bin failed with error -2
[ 9.466246] xilinx-hdmi-rx a1000000.hdmi_rxss: Using Xilinx built-in EDID.
[ 9.473275]
[ 9.473275] Successfully loaded edid.
[ 9.478621] xilinx-video amba:vcap_hdmi: Entity type for entity a1000000.hdmi_rxss was not initialized!
[ 9.493520] xilinx-hdmi-rx a1000000.hdmi_rxss: probe successful
[ 9.499614] xlnx-drm-hdmi a0080000.hdmi_txss: probed
[ 9.504648] xlnx-drm-hdmi a0080000.hdmi_txss: hdmi tx audio disabled in DT
[ 9.514695] xlnx-drm-hdmi a0080000.hdmi_txss: probe successful
[ 9.526397] [drm] Supports vblank timestamp caching Rev 2 (21.10.2013).
[ 9.533043] [drm] No driver support for vblank timestamp query.
[ 9.539234] xlnx-drm xlnx-drm.0: bound b00c0000.v_mix (ops 0xfffff8008b33eb8)
[ 9.546556] xlnx-drm xlnx-drm.0: bound a0080000.hdmi_txss (ops xlnx_drm_hdmi_component_ops [xilinx_hdmi_tx])
[ 9.556337] [drm] Cannot find any crtc or sizes
[ 9.647540] xlnx-mixer b00c0000.v_mix: fb0: frame buffer device
[ 9.680990] [drm] Initialized xlnx 1.0.0 20130509 for b00c0000.v_mix on minor 1
Starting internet superserver: inetd.
Configuring packages on first boot....
(This may take several minutes. Please do not power off the machine.)
Running postinst /etc/rpm-postinsts/100-xserver-nodm-init...
Running postinst /etc/rpm-postinsts/101-sysvinit-inittab...
update-rc.d: /etc/init.d/run-postinsts exists during rc.d purge (continuing)
INIT: Entering runlevel: 5
Configuring network interfaces... [ 10.221239] pps pps0: new PPS source ptp0
[ 10.225304] macb ff0e0000.ethernet: gem-ptp-timer ptp clock registered.
[ 10.231978] IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
udhcpc (v1.24.1) started
Sending discover...
Sending discover...
Sending discover...
No lease, forking to background
done.
Starting system message bus: dbus.
Starting Dropbear SSH server: dropbear.
Starting syslogd/klogd: done
Starting tcf-agent: OK

Setting console loglevel to 0 ...
root@xilinx:~#
```

Step 11: Run application with *gst-launch-1.0*



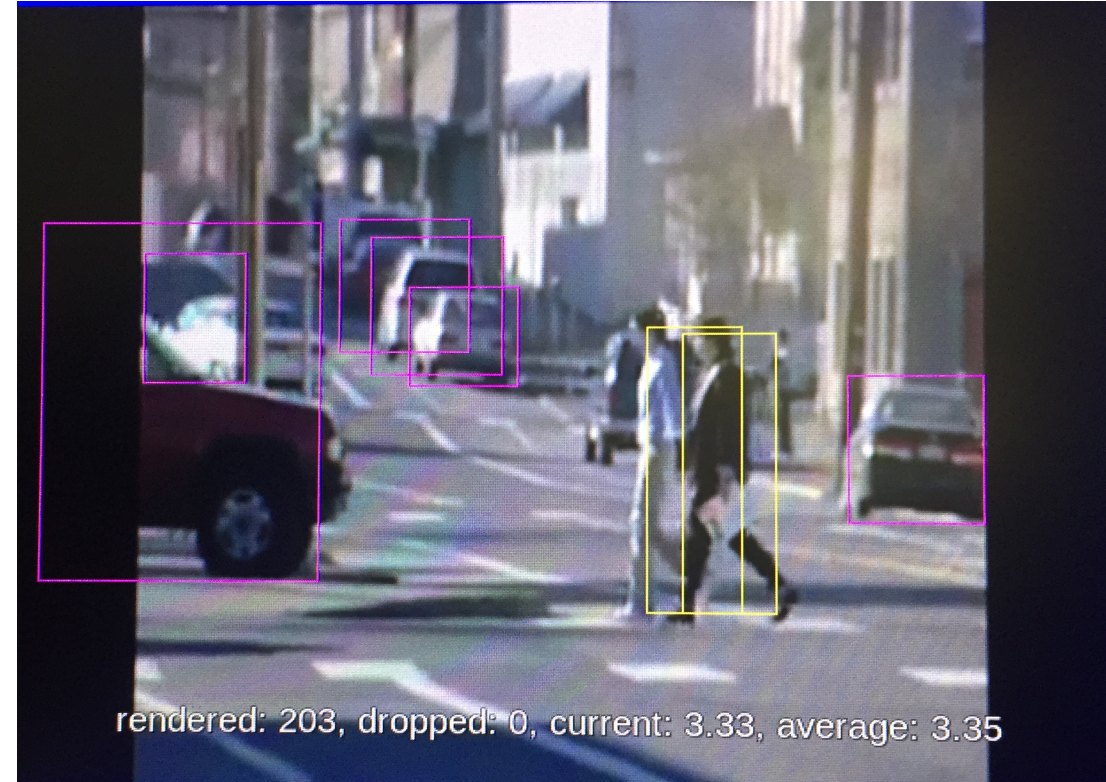
```
root@xilinx:~# cd /media/card
root@xilinx:~# video_cmd -s 1 -i 640x480@YUY2 -X
root@xilinx:~# video_cmd -d 1 &
root@xilinx:~# gst-launch-1.0 \
    v4l2src device=/dev/video2 force-aspect-ratio=false ! \
    "video/x-raw, width=640, height=480, format=YUY2, framerate=3/1, pixel-aspect-ratio=4/3" ! \
    videoconvert ! \
    "video/x-raw, width=640, height=480, format=BGR, framerate=3/1, pixel-aspect-ratio=4/3" ! \
    sdxttraffictdetect ! \
    fpsdisplaysink video-sink=" kmssink sync=false plane-id=29 bus-id="b00c0000.v_mix" render-
rectangle="\<0,0,640,480>" " text-overlay=true sync=false
```



Results



HDMI Video Input



HDMI Monitor 640x480

Agenda



- CMC Microsystems
- Overview
- Hardware and software environment
- CNN Training and Quantization Flow
- **Inference Demonstration**
- How to access
- Q&A

DEMO

Agenda



- CMC Microsystems
- Overview
- Hardware and software environment
- CNN Training and Quantization Flow
- Inference Demonstration
- **How to access**
- Q&A

Getting Started: Xilinx Tools



- Xilinx tools & licenses available for academic use
- Local and CMC cloud installation

The screenshot shows the CMC Microsystems website. At the top right is the CMC Microsystems logo. Below it is a search bar and buttons for 'Get an Account' and 'Sign In'. A navigation bar contains 'Products and Services', 'News & Events', 'NDN Hub', and 'About Us'. Below the navigation bar, a breadcrumb trail reads 'You are here: Home > Products and Services > CAD: Xilinx SDSoC'. To the right of the breadcrumb are a 'Share' button and a 'Checkout' button. On the left side of the main content area is a vertical menu with options: CAD, FAB, LAB, Support, and Subscription. The main content area is titled 'CAD: Xilinx SDSoC' and features the Xilinx logo. To the right of the logo, it states 'Minimum Subscription Required: Research'. Further right is a box titled 'How to access this item?' with links for 'Access Requirements' and 'Download Software'. Below this is a 'Description' section. The description states: 'Xilinx SDSoC provides a comprehensive and easy to use application development environment for embedded C/C++ applications targeting Xilinx Zynq SoCs. The environment includes:'. This is followed by a bulleted list: '• A C/C++ full-system optimizing compiler', '• System-level profiling', '• Automated software acceleration', '• Automated system connectivity generation', '• Libraries to speed programming', and '• Support for bare metal, Linux and FreeRTOS operating systems'. The description continues: 'The SDSoC installation includes the Xilinx Vivado and Vivado HLS tools for design implementation and high-level synthesis. You can find more information on the Xilinx product page at <http://www.xilinx.com/products/design-tools/software-zone/sdsoc.htm>. You can also join the [NDN Development Systems Community](#) to learn more and interact with other researchers using these environments.'

CAD: Xilinx SDSoC

Minimum Subscription Required: Research

How to access this item?

- Access Requirements
- Download Software

Description

Xilinx SDSoC provides a comprehensive and easy to use application development environment for embedded C/C++ applications targeting Xilinx Zynq SoCs. The environment includes:

- A C/C++ full-system optimizing compiler
- System-level profiling
- Automated software acceleration
- Automated system connectivity generation
- Libraries to speed programming
- Support for bare metal, Linux and FreeRTOS operating systems

The SDSoC installation includes the Xilinx Vivado and Vivado HLS tools for design implementation and high-level synthesis. You can find more information on the Xilinx product page at <http://www.xilinx.com/products/design-tools/software-zone/sdsoc.htm>. You can also join the [NDN Development Systems Community](#) to learn more and interact with other researchers using these environments.

Getting Started: ZCU102 Zynq Ultrascale+ MPSoC Development Kit

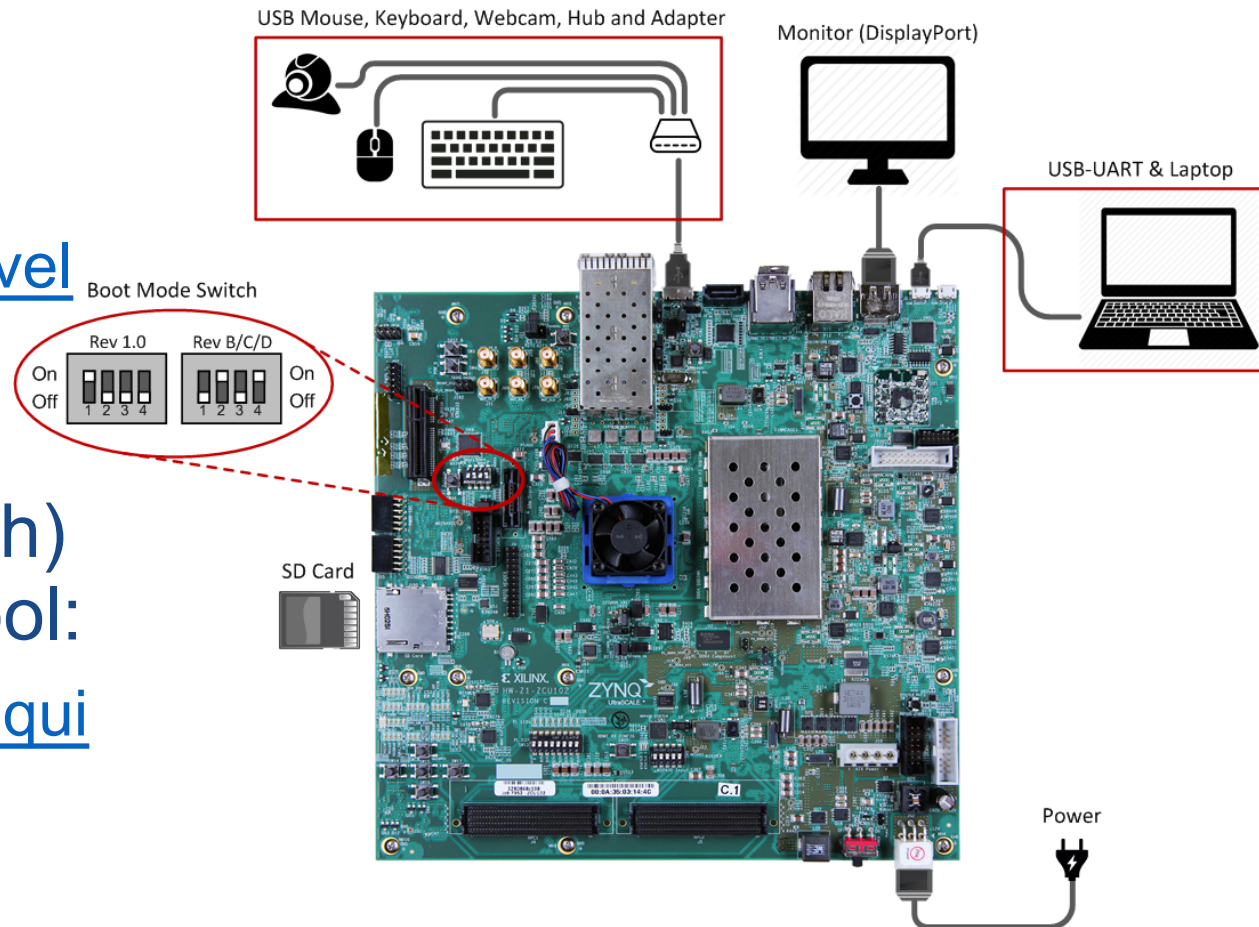


- Available for shared access at universities via emSYSCAN CFI project:

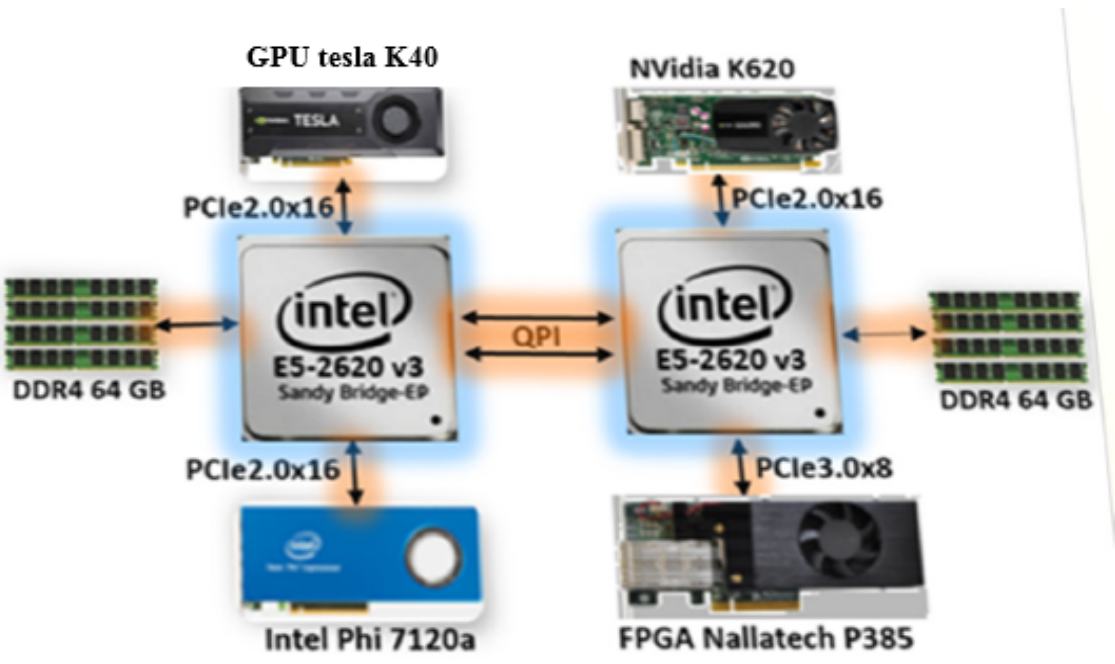
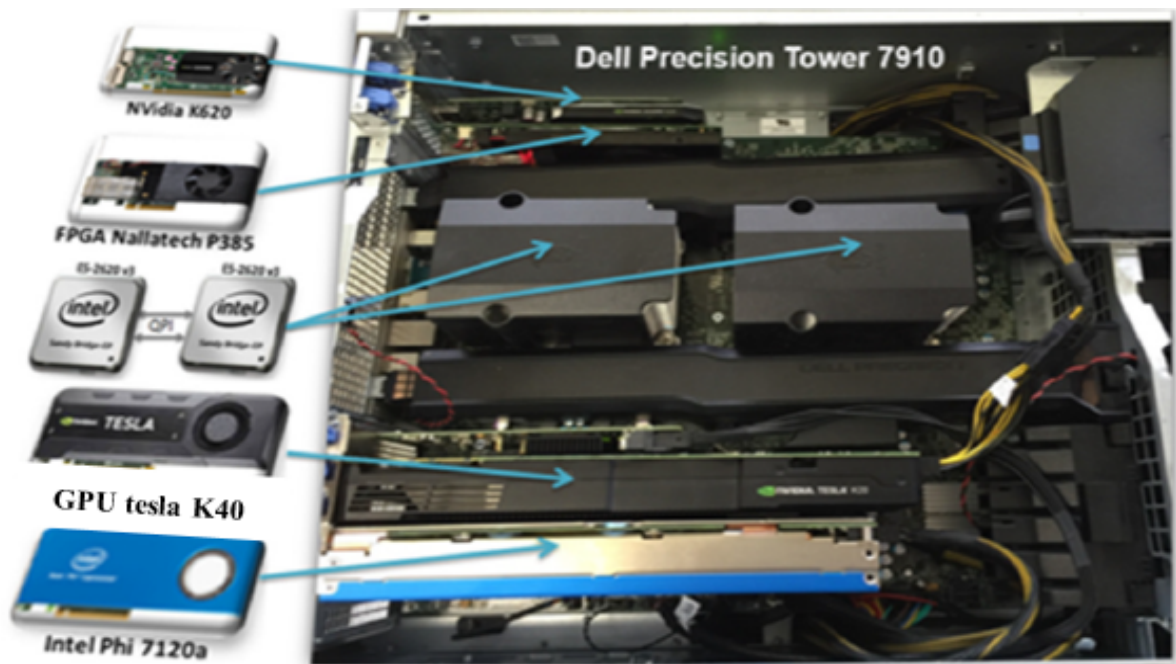
<https://community.cmc.ca/community/development-systems>

- Available for short-term (~6-month) loan through CMC Equipment Pool:

<https://www.cmc.ca/WhatWeOffer/Test/EquipmentLoan.aspx>



CMC Heterogeneous Systems

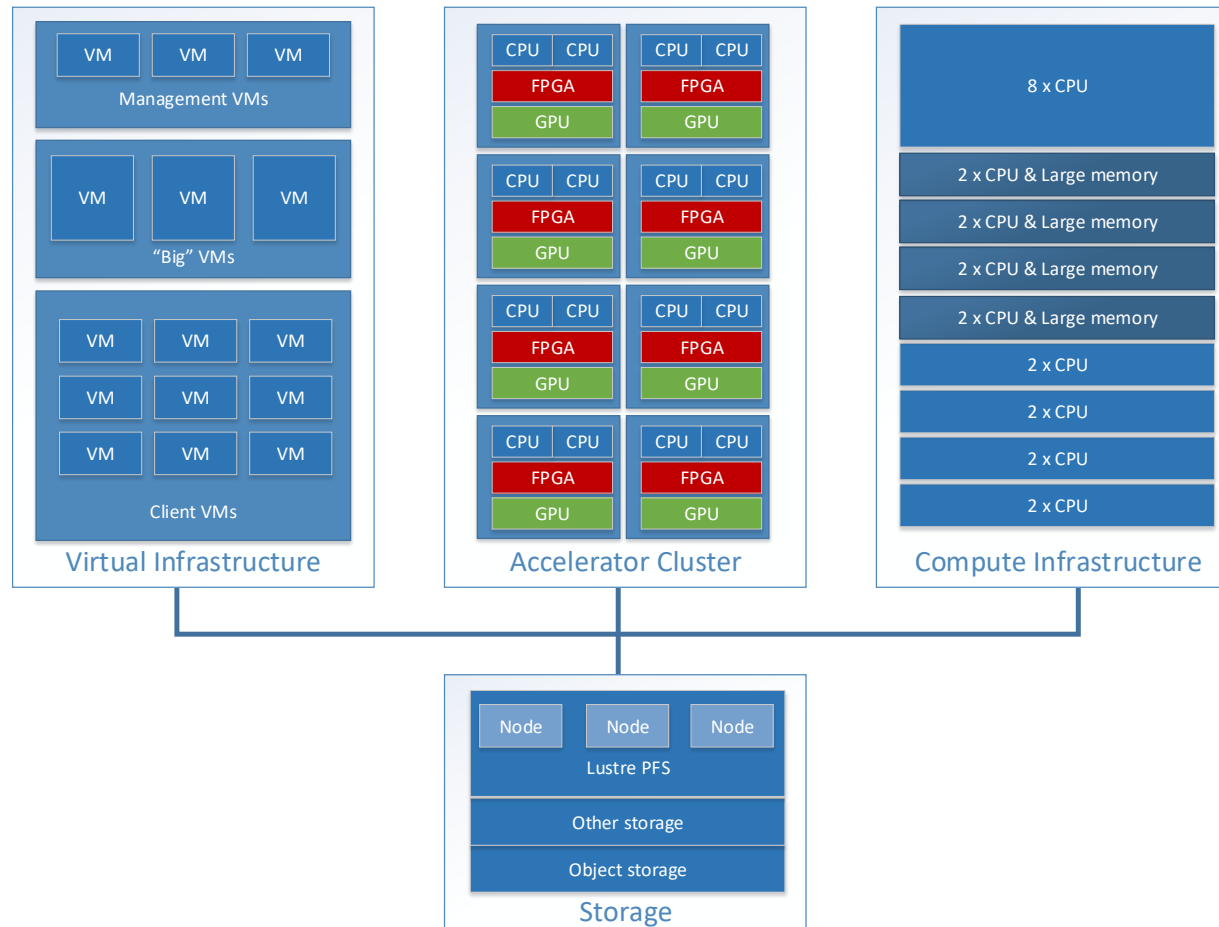


Accelerator	Features	Host Interface	Compute Performance	Power
Nallatech 385	<ul style="list-style-type: none">Altera Stratix VMemory: 2 banks of 4 GB	PCIe 3.0 x 8	Unavailable	Typical application ≤ 25 W
TESLA K40	<ul style="list-style-type: none">2880 CUDA coresMemory: 12 GB at 288 GB/s	PCIe 2.0 x 16	4.29 TFLOPS (single precision) 1.43 TFLOPs (double precision)	225 W
Xeon Phi 7120a	<ul style="list-style-type: none">61 Cores, 1.33 GHzMemory: 16 GB at 352 Gb/s	PCIe 2.0 x 16	Peak Double Precision 1.003 TFLOPs	300 W

Supported platforms

- SAP:** Simulation Acceleration platform; CPU + FPGA
- MPA:** Multiprocessor Array Platform; CPU + GPU or Xeon Phi
- HPP:** Heterogeneous Processing Platform; MPA + SAP

Getting Started: CMC Cloud: Unified Architecture



Seamless Transition Between Environments

- **CAD** - Design using CMC Cloud desktop
- **FAB** - Simulate on the CAD Compute cluster
- **LAB** - Prototype on the FPGA+GPU cluster

More info: www.cmc.ca/cmcccloud

CMC Cloud: CAD Compute Cluster



Speed up your simulations

- CMC engineers provide assistance in utilizing the infrastructure as well as domain knowledge on utilizing HPC infrastructure
- Documentation/reference designs available for ANSYS, COMSOL, Xilinx and more
- Uniform array available in standard and large memory configurations



CAD Compute Cluster – 8 nodes

- Dual 16-core 2.1-.3.7 GHz CPU
- 4 nodes each with 384GB RAM
- 4 nodes each with 768GB RAM
- 300GB local storage
- 100Gb EDR node interconnect / 10GbE storage

More info: www.cmc.ca/cmccloud

CMC Cloud: Multi-FPGA+GPU Cluster



CPU, GPU and FPGA in pre-validated cluster to scale heterogeneous computing workloads

- CMC engineers provide assistance with access and application best practices
- Hosted and managed by CMC as a cloud resource; accessible at your desktop
- Reference designs using software stack for OpenCL + MPI heterogeneous cluster computing



Heterogeneous Compute Cluster – 8 nodes

- Dual 12 core 2.2-3.0 GHz CPU
- 192GB RAM
- 300GB local storage
- 100Gb EDR node interconnect / 10GbE storage
- **Xilinx Alveo U200 FPGA + NVIDIA V100 GPU**

More info: www.cmc.ca/cmccloud

- Presented flow for CNN training and embedded inference using Xilinx DNNDK and Zynq Ultrascale+ MPSoC Development Kit
- Fast, stream-lined flow for rapid prototyping
- Tools and equipment available through CMC

- Performance improvements:
 - DNNDK network pruning tool
 - Multi-threading/multi-DPU
 - Other networks/models
- Training (Xilinx SDS, Intel Quartus, UltraScale+ MPSoC, DNNDK)
- Release CMC Cloud Heterogeneous Computing Cluster (HCC)
- AI to ASIC reference design and flow

contact us to express interest, become a lead client!

Thanks!



Questions?

For more information, contact:

Hugh Pollitt-Smith, CMC Microsystems

Pollitt-smith@cmc.ca