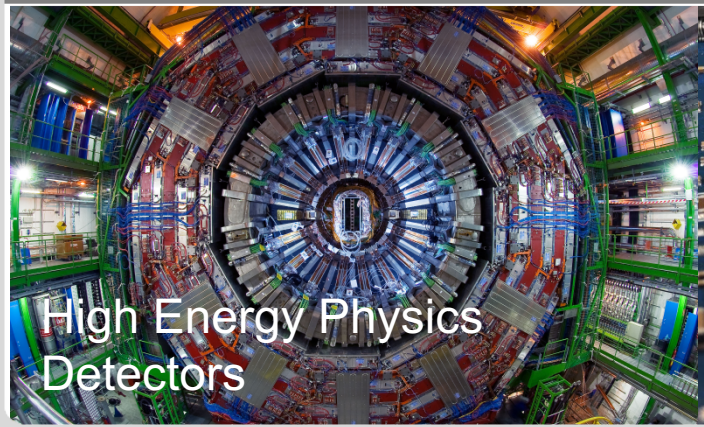
Karlsruhe Institute of Technology

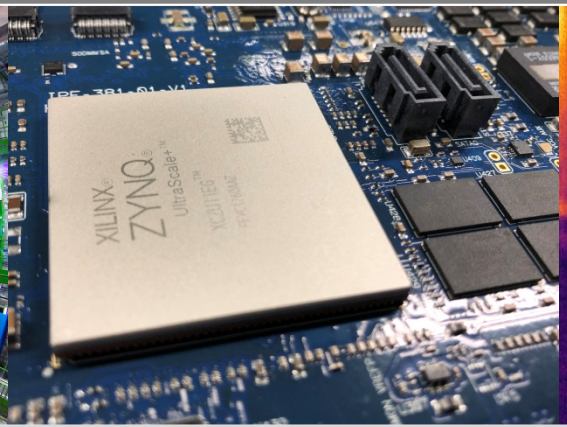
FPGAs in Detector Instrumentation: concepts and trends

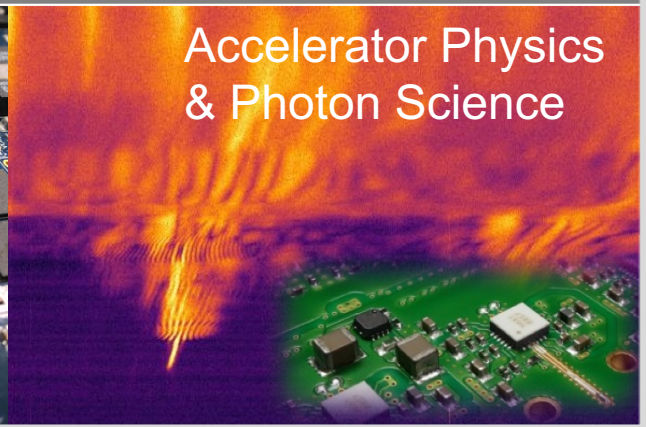
Michele Caselle

19-21 February 2020 Best Western Hotel

Helmholtz Research Field Matter



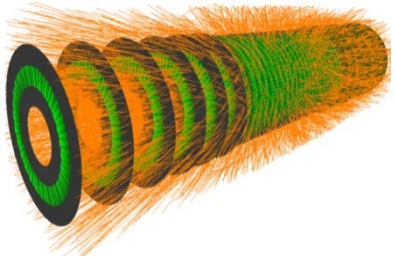




Challenges for the future

The next generation of detectors are extremely challenging: HEP, astrophysics, photon science, etc.

Particle physics

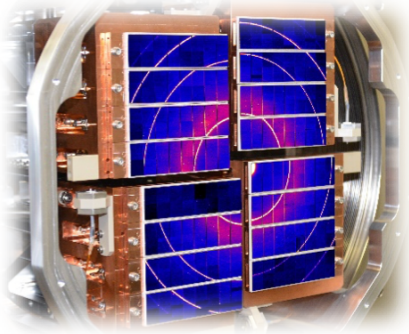


Unprecedent luminosity operations,

4D detectors: excellent spatial and time resolution

HL-CMS: Unprecedented data rate of up to 50 Tb/s to be processed in < 4 µs with high efficiency

Photon Science



Terapixel per second imaging:

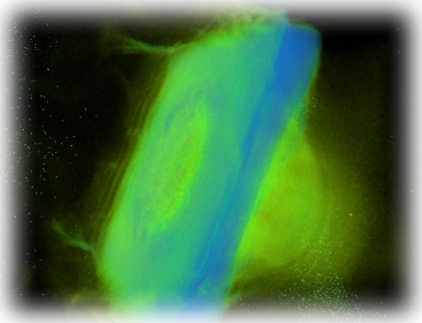
- 100 million pixel,
- MHz- frame rates
- High dynamic range

Astroparticle physics

Cryogenic detectors of unique energy resolution for dark matter searches and neutrino physics

CTA, IceCube-Gen2, etc.

Accelerators and beam physics



Complex dynamics on short time scale,

Multi-spectral THz detectors for beam diagnostics, for plasma accelerators

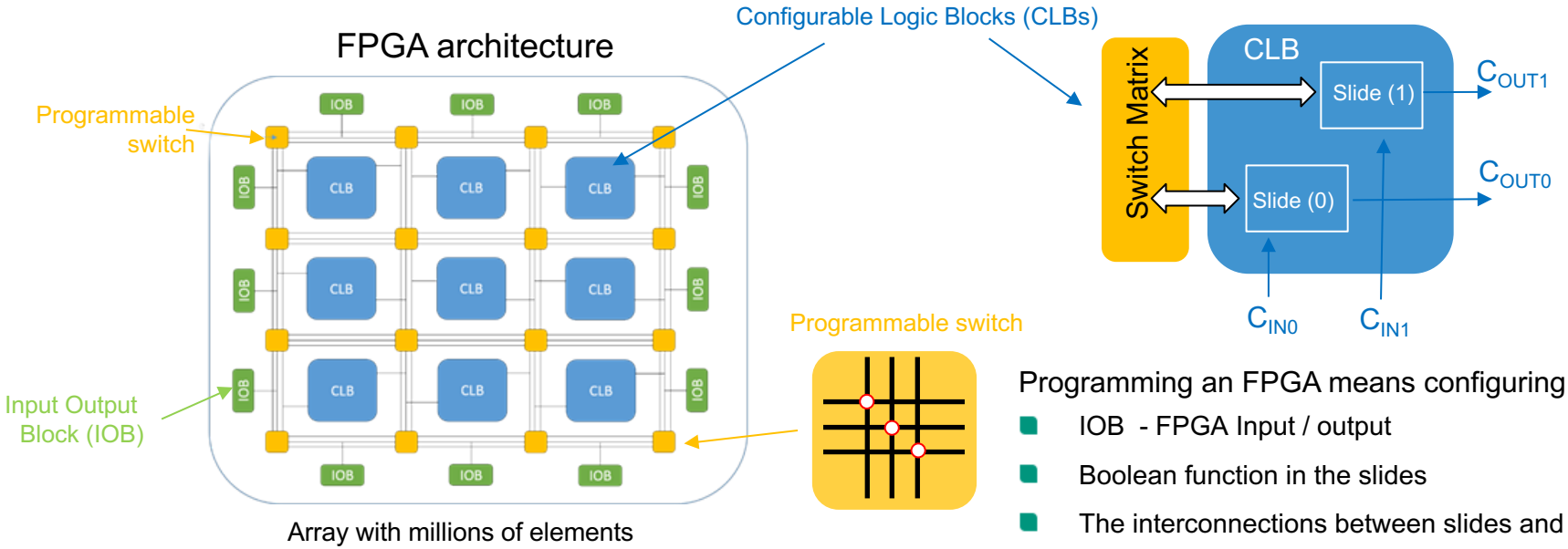
FPGAs in Detector Instrumentation - Outlook

Karlsruher Institut für Technologie

- **Motivations** - *How to cope with the data deluge from the next generation of detectors?*
- **Novel heterogenous programmable devices** – *What is the evolution of Field Programmable Gate Array (FPGA) devices?*
- **Machine learning (ML) and artificial intelligence** - *Will enable us to process the data deluge in real-time?*
- **AI on FPGA** - *How develop a fast ML inference on FPGA?*
- **Applications:** *fast track-reconstruction based on ML, control of the beam dynamics in complex synchrotron machines e.g. autonomous accelerator?*

Field Programmable Gate Arrays (FPGA)

- The first commercially viable Field Programmable Gate Array (FPGA), named XC2064, has been invented in 1985 by Ross Freeman and Bernard Vonderschmitt, the Xilinx co-founders.
- A FPGA is a semiconductor *Integrated Circuit* (*IC*) device on which the function can be defined after manufacturing ("in the field") using software-like languages (ex: VHDL, Verilog). FPGAs can be reconfigured at any times.

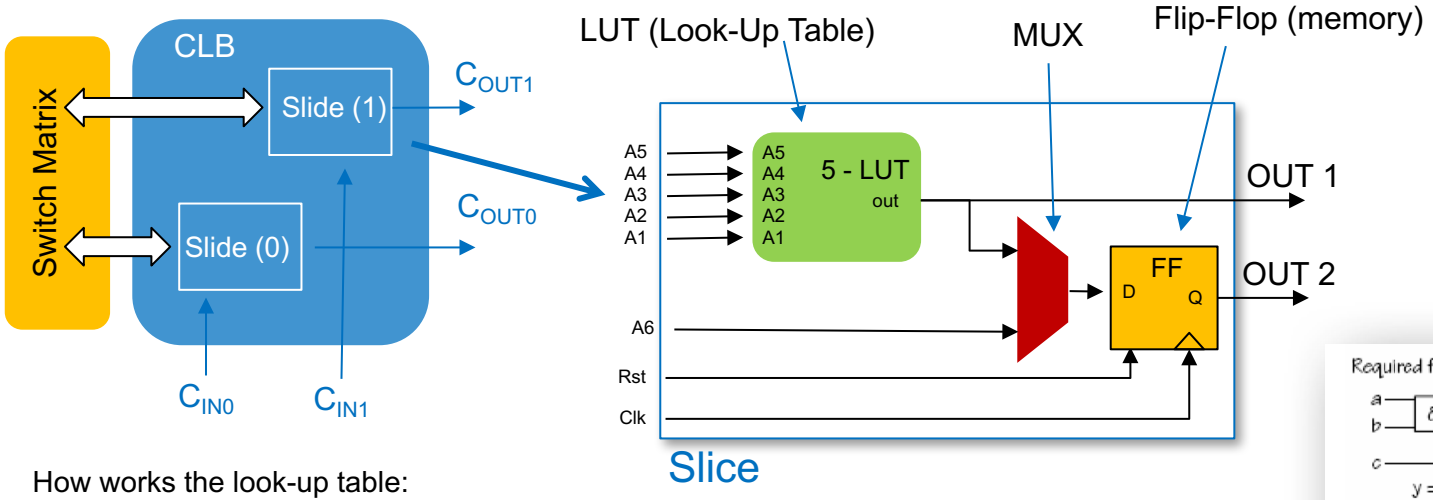


Programming an FPGA means configuring:

- IOB - FPGA Input / output
- Boolean function in the slides
- The interconnections between slides and IOB

Field Programmable Gate Arrays (FPGA)

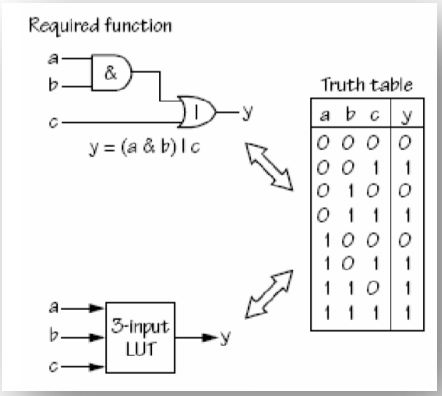
- A Look-Up Table (LUT) can implement any arbitrary Boolean function of its inputs and can be cascaded to other LUTs to perform more complex functions



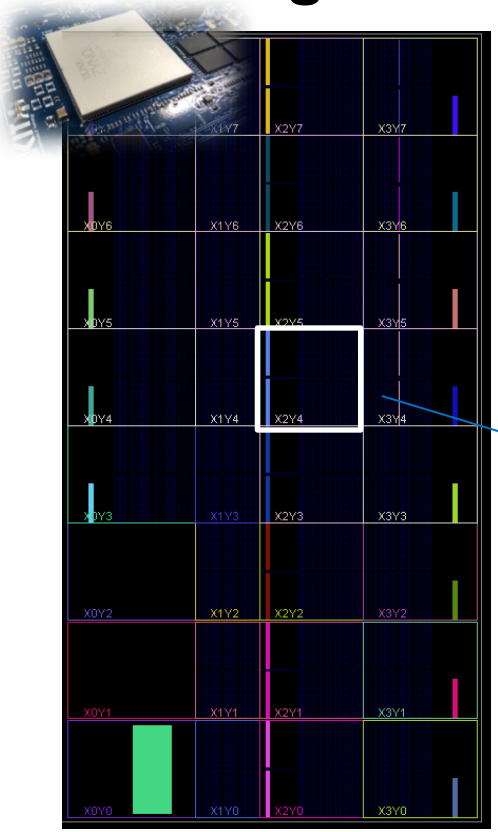
How works the look-up table:

- Can be programmed
- LUT contains memory cells to implement any arbitrary “small” logic function
- Each cell holds “0” or “1”
- Programmed with outputs of Truth Table

Example



Field Programmable Gate Arrays (FPGA)

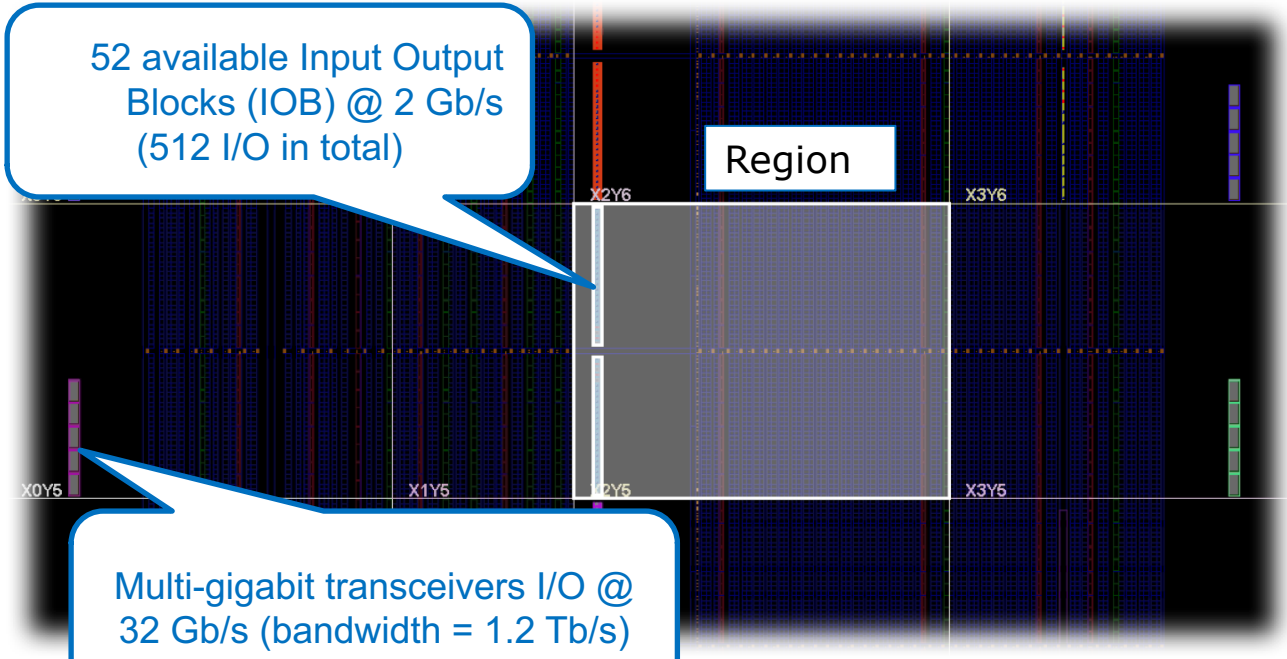


Zynq ZU11EG Ultrascale+ (Xilinx)

FPGA is divided in regions

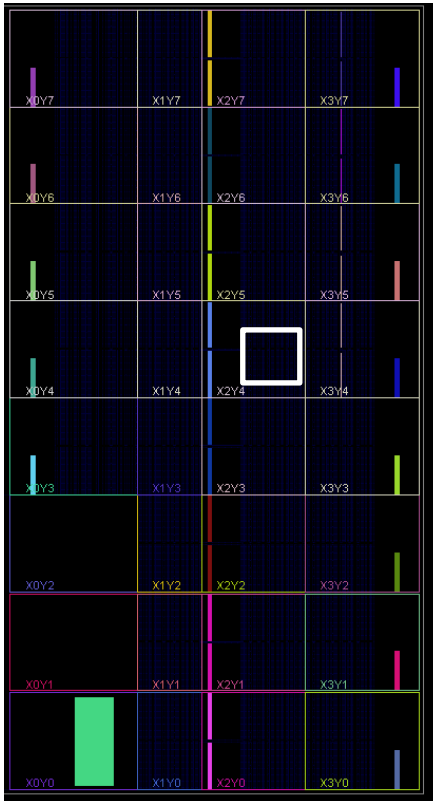
52 available Input Output Blocks (IOB) @ 2 Gb/s (512 I/O in total)

Region

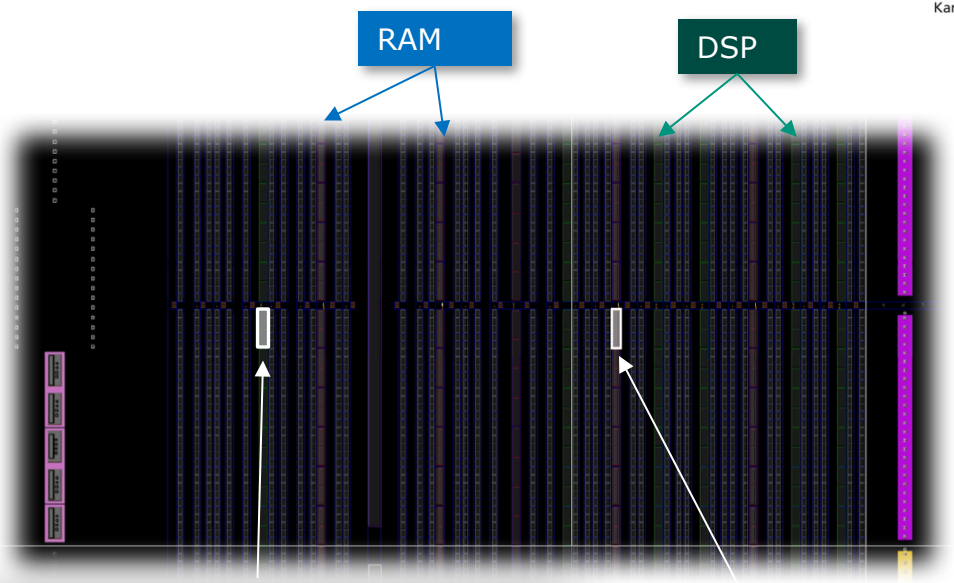


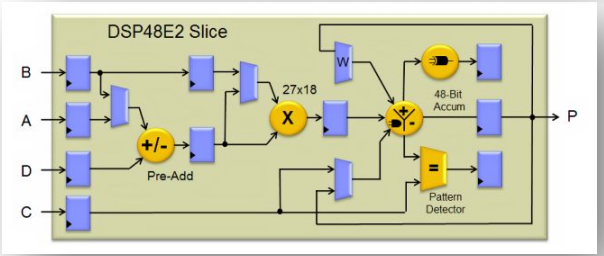
Multi-gigabit transceivers I/O @ 32 Gb/s (bandwidth = 1.2 Tb/s)

Field Programmable Gate Arrays (FPGA)



Zynq ZU11EG Ultrascale+ (Xilinx)

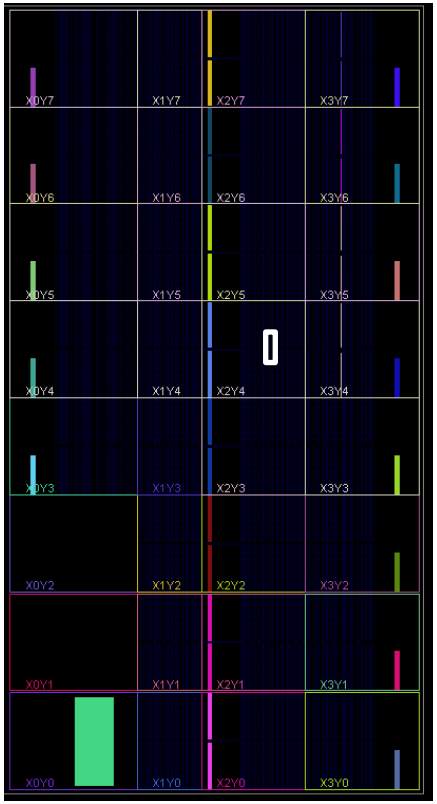




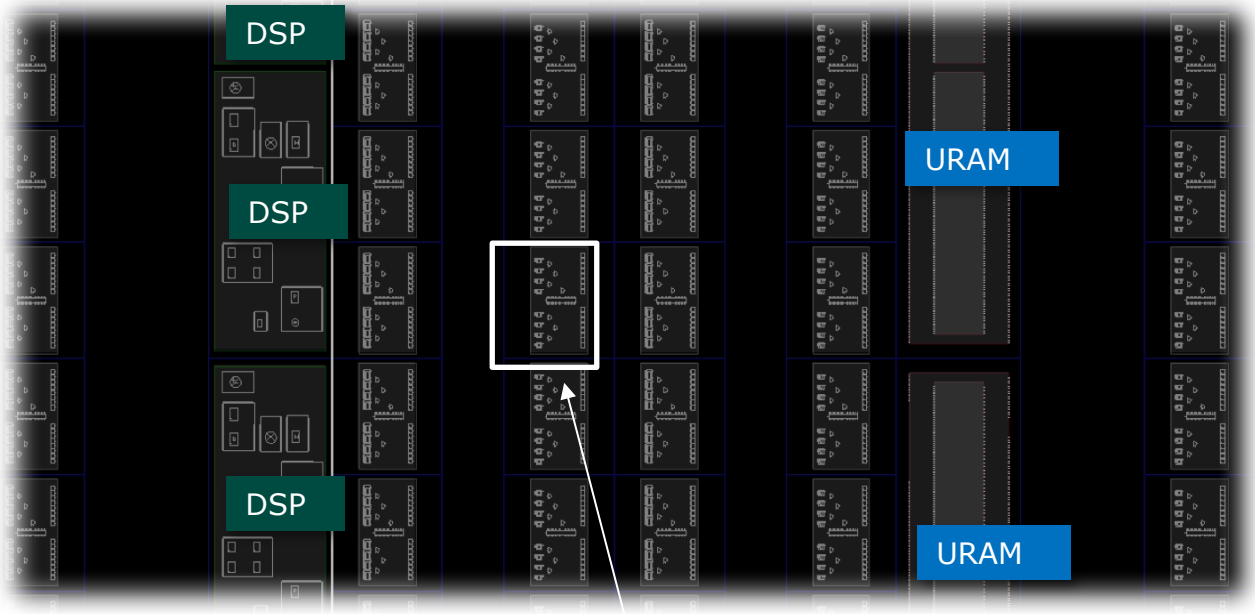
Block RAM 21 Mbit + 22 Mbit (Ultra RAM)

DSP -Digital Signal Processing (2928 units)

Field Programmable Gate Arrays (FPGA)

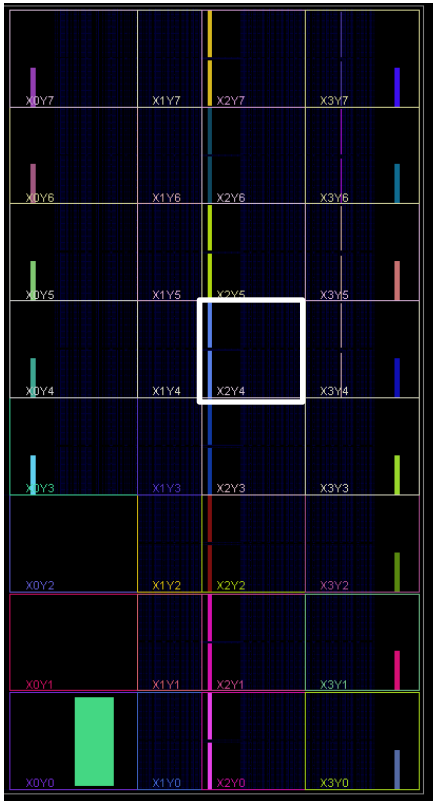


Zynq ZU11EG Ultrascale+ (Xilinx)

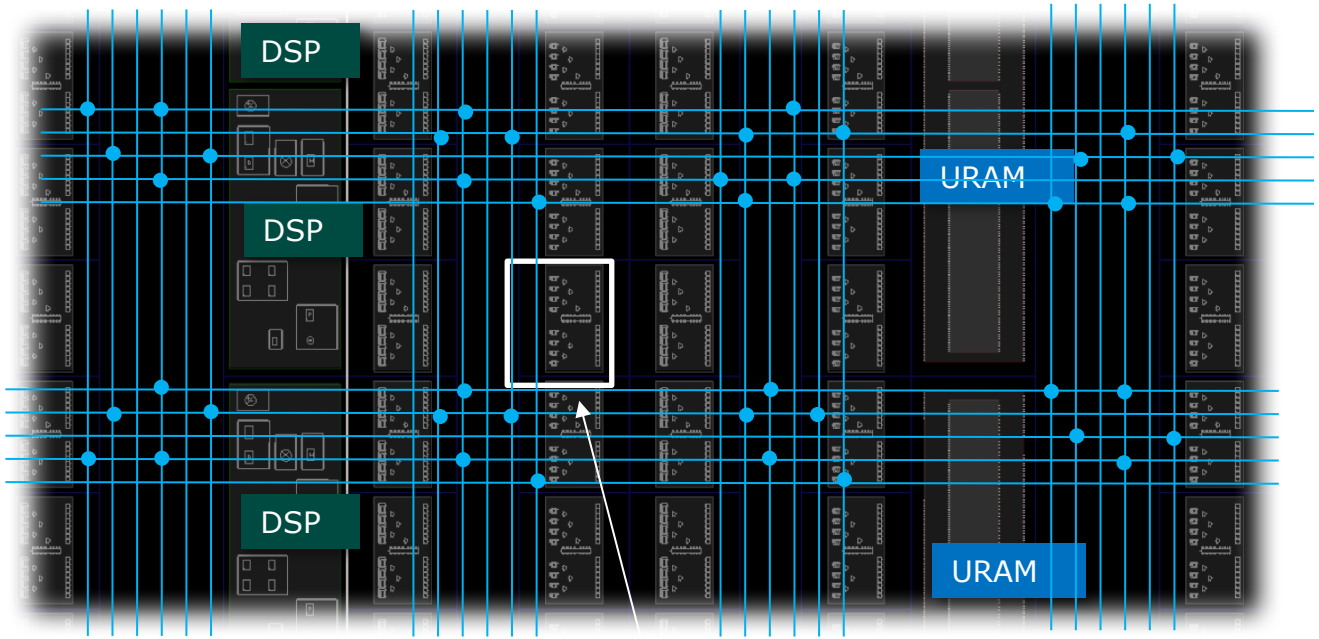


CLBs - Configurable Logic Blocks (> 1 Million of CLBs)

Field Programmable Gate Arrays (FPGA)



Zynq ZU11EG Ultrascale+ (Xilinx)



CLBs - Configurable Logic Blocks (> 1 Million of CLBs)

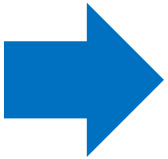
Sea of Programmable Logic and Routing resourses

Latency & bandwidth comparable to ASICs

From *FPGA* …

… to *Adaptive Compute Acceleration Platform*

New generation of heterogenous programmable platform



A New Class of Devices for Today's Challenges

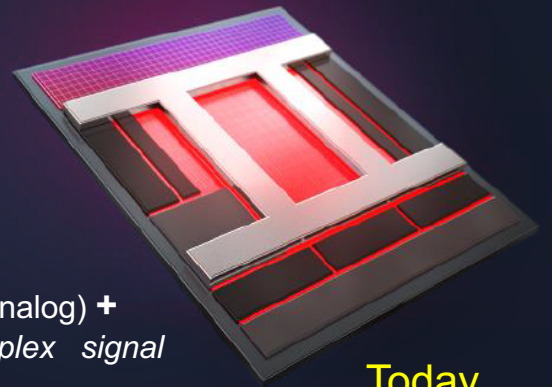
16 nm TSMC (FinFET) MPSoC (Zynq US+)
1.5 GHz Quad-core ARM processor +
FPGA @ 890 MHz F_{MAX} (all register used)

20 nm TSMC (FinFET) MPSoC (Zynq US)
1 GHz Dual-core ARM processor +
FPGA @ 740 MHz F_{MAX} (all register used)

90 nm CMOS SoC (Virtex-4)
FPGA @ 300 MHz F_{MAX} +
450 MHz PowerPC processor

VERSAL architecture

FPGA based on +
Multi-core processors +
Radio frequency front-end (analog) +
Artificial intelligent & complex signal processing (hard-core)
7 nm TSMC (FinFET)



Today
ACAP

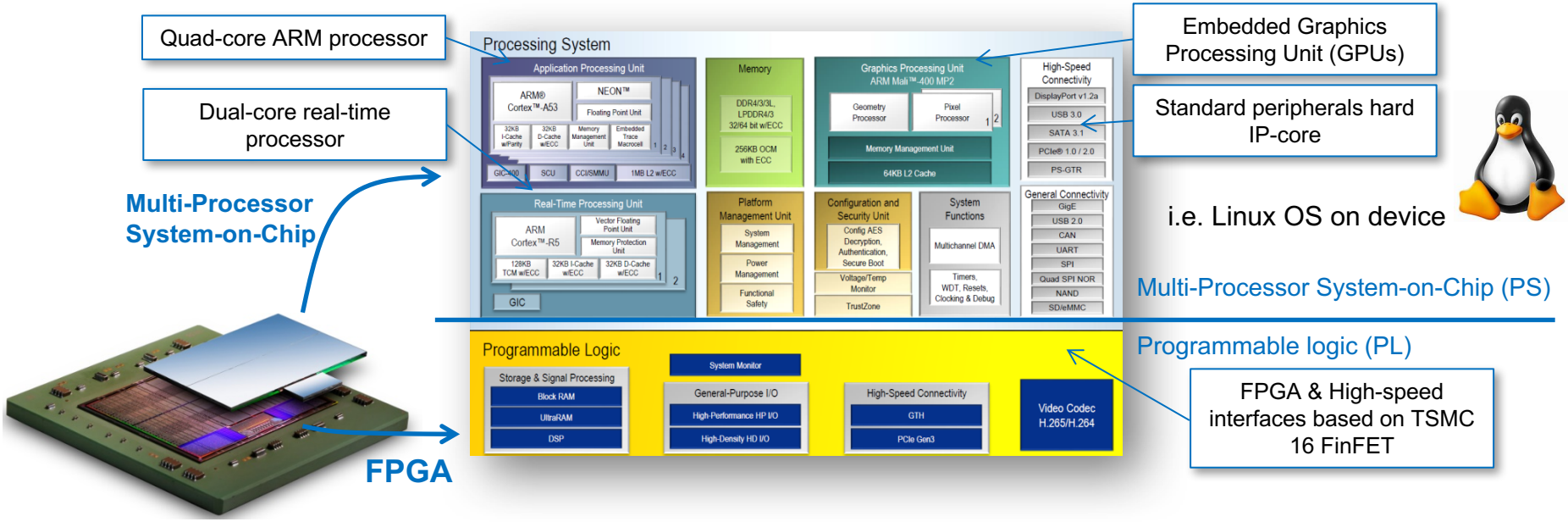
Invented (XC2064)
1985
FPGA
2005
SoC
2014
2016
MPSoC
2018
RFSoC
16 nm TSMC (FinFET)

RFSoC (Zynq US+)
1.5 GHz Quad-core ARM processor +
890 MHz F_{MAX} (all register used) +
Radio frequency front-end (analog)

Device Category

Heterogeneous: ZYNQ MPSoC technology

- Heterogeneous platform of the Zynq System-on-Chip (SoC) integrates, in a monolithic device, FPGA resources with a back-end software running on a hard-core ARM-based processor.

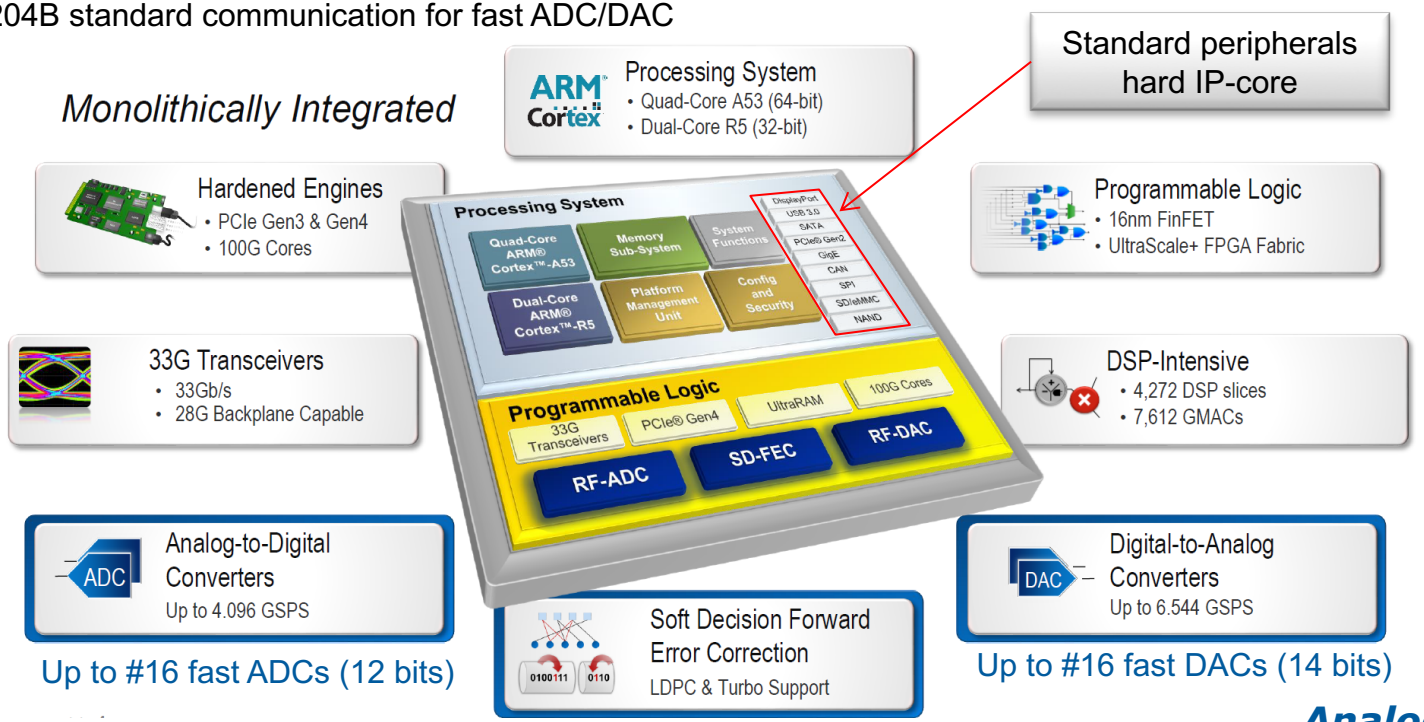


System-on-Chip (SoC) workshop – CERN , 12-14 June 2019
https://indico.cern.ch/event/799275/

Ref. https://www.xilinx.com/support/documentation/selection-guides/zynq-ultrascale-plus-product-selection-guide.pdf

Heterogeneous: ZYNQ *RF*SoC technology

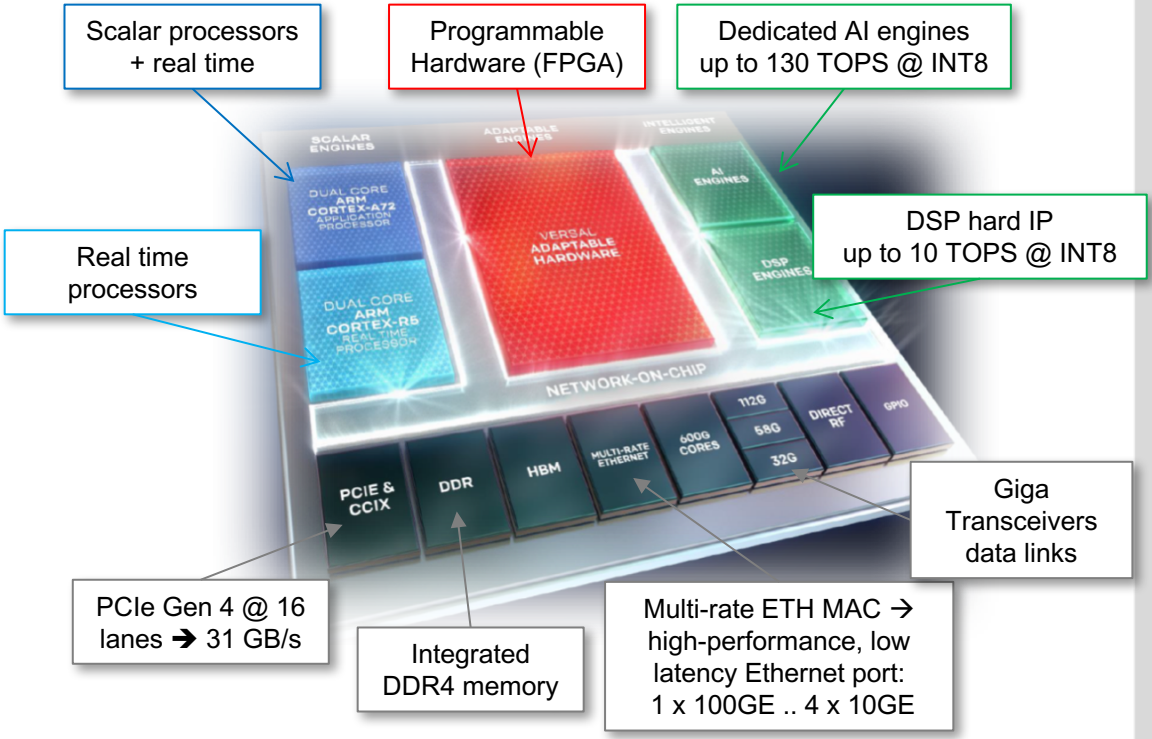
- The bandwidth bottleneck of previous ZYNQ MPSoC was the bandwidth and complexity of the JESD204B standard communication for fast ADC/DAC



Ref. https://www.xilinx.com/support/documentation/selection-guides/zynq-usp-rfsoc-product-selection-guide.pdf

Versal – ACAP (Overview)

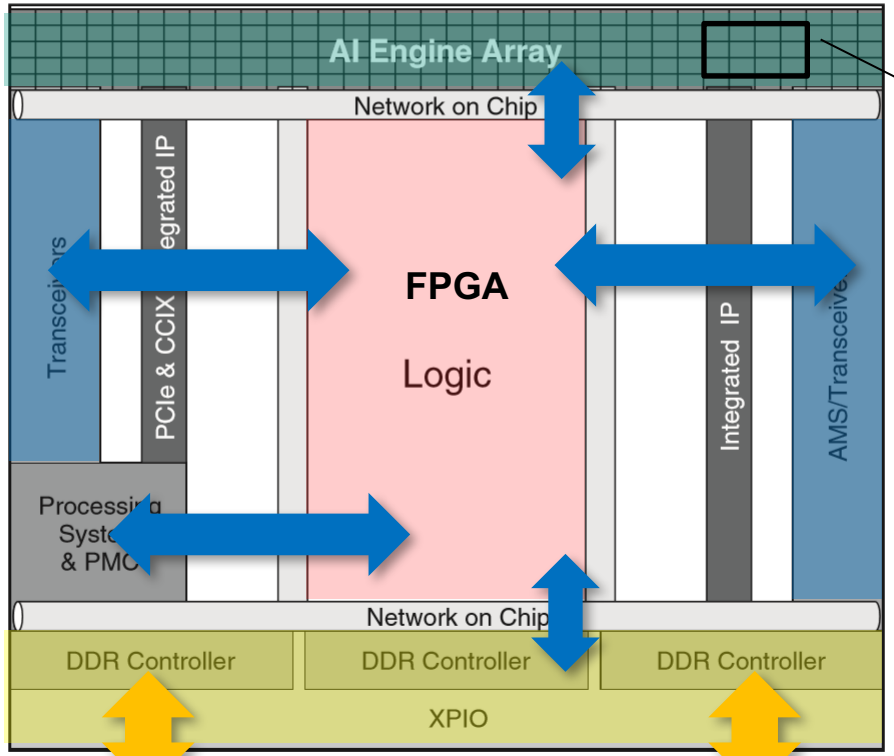
- New Xilinx architecture Versal - ACAP (Adaptive Compute Acceleration Platform) develop in TSMC 7nm FinFET technology, key features:
- More heterogeneous: *Scalar processors + FPGA + HBM + RF + AI and DSP engines*
- *Network-on-Chip:* is a full blocking crossbar between memories, PL, processors, AI engines. It is built using stacked silicon interconnect (SSI) technology
- *Adaptable engine* (FPGA): memory elements tightly coupled with programmable logic
- AI engines: two-dimensional array of AI engine optimized for real-time computation.

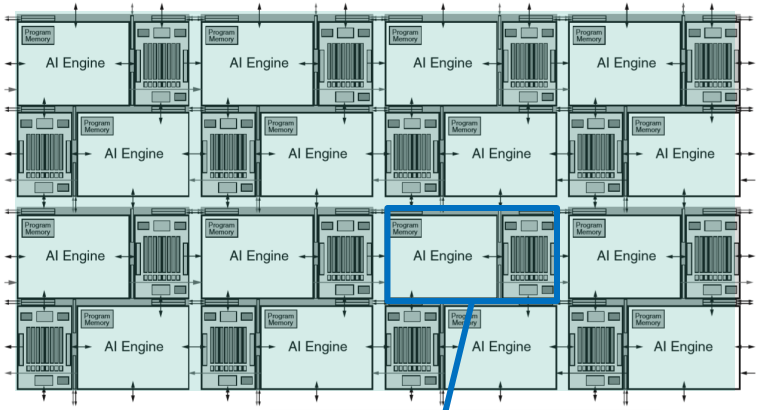


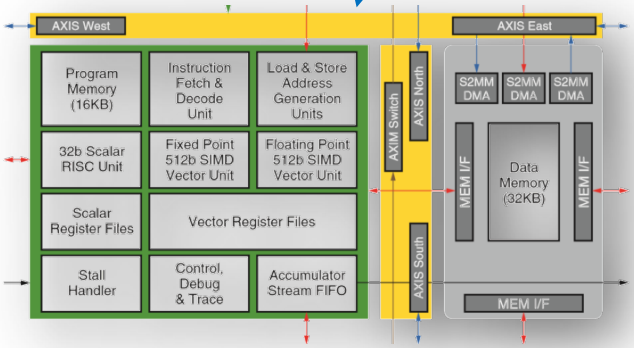
Ref: https://www.xilinx.com/support/documentation/white_papers/wp505-versal-acap.pdf

Versal- ACAP architecture

Matrix up to 400 AI engines (VC1902)
Data Memory up to 100 Mb (VC1902)



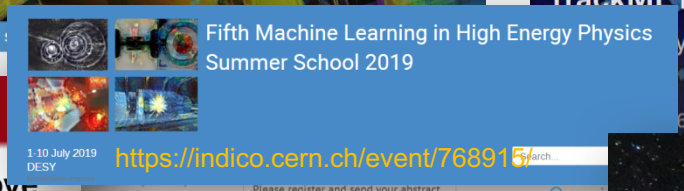


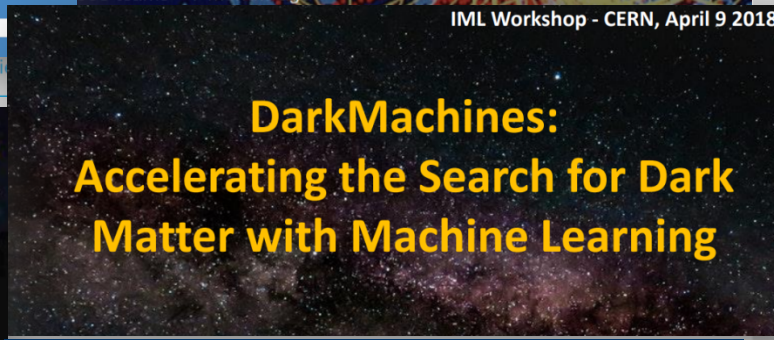


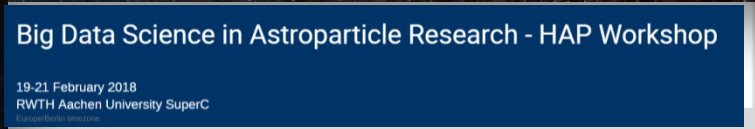
Next generation of data processing

Machine Learning

The generation of detectors will face a peak luminosity of 30 x 10^{34} cm^{-2}/s with a pile-up 1000 and radiation levels that are 1-2 orders of magnitude larger than those at the HL-LHC. Timing precision of 5 – 10 ps is required for pile-up mitigation. *The resulting data rates lie in the hundreds of TB/s.*





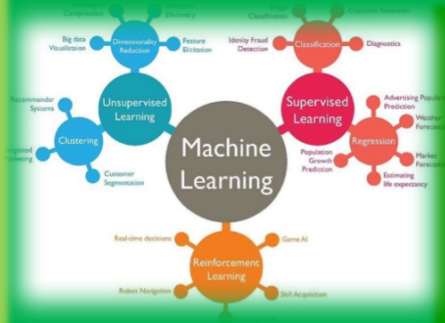


What's the different between Artificial lintelligence, Machine Learning, and Deep Learning?

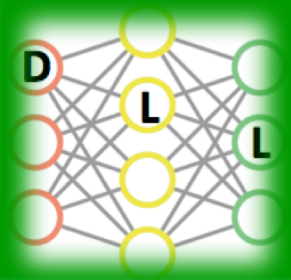
AI involves machines that can perform tasks that are characteristic of human intelligence
While this is rather general, it includes things like planning, understanding language, recognizing objects and sounds, learning, and problem solving

At its core, machine learning is simply a way of achieving AI

"*the ability to learn without being explicitly programmed*." You see, you can get a traditional program without using machine learning, but this would require building millions of lines of codes with complex rules and decision-trees.

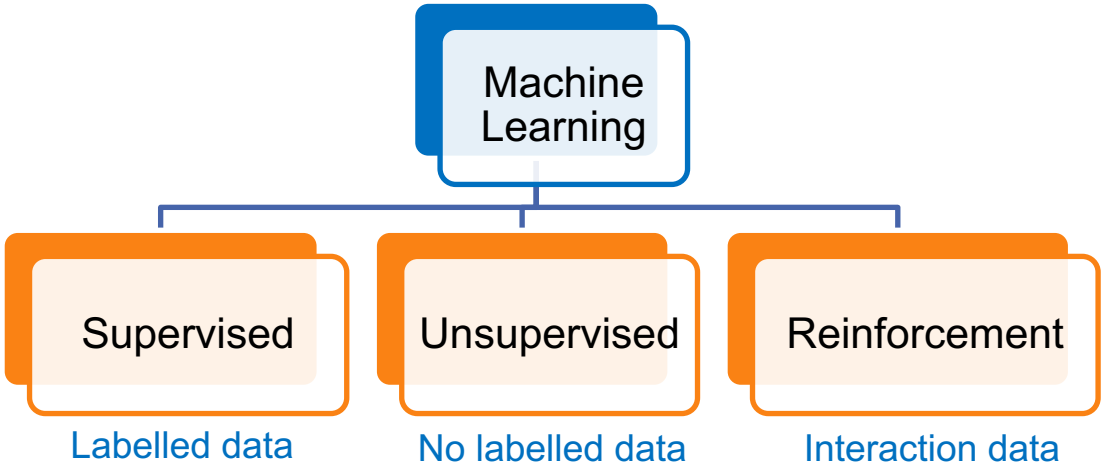


Deep learning is one of many approaches to machine learning



Deep learning was inspired by the structure and function of the brain, namely the interconnecting of many neurons. Artificial Neural Networks (ANNs) are algorithms that mimic the biological structure of the brain.

Machine learning

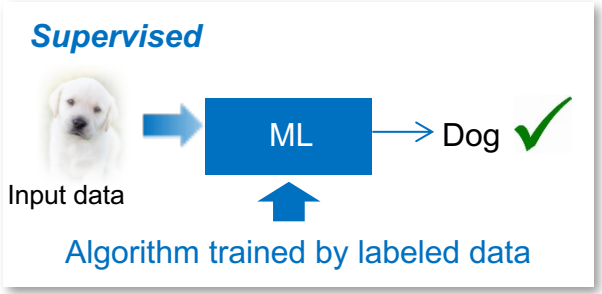


Supervised: All data is *labeled* and the algorithms learn to predict the output from the input data.

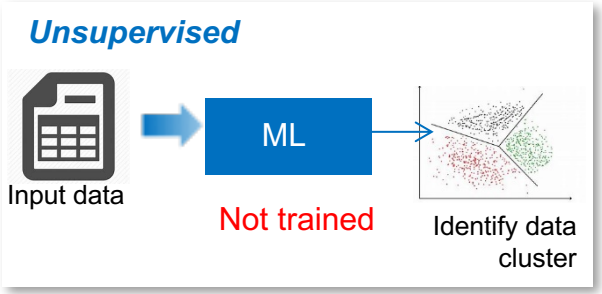
Unsupervised: All data is *unlabeled* and the algorithms learn to inherent structure from the input data

Reinforcement: The learning algorithm is trained not on present data but rather based on a *feedback* system.

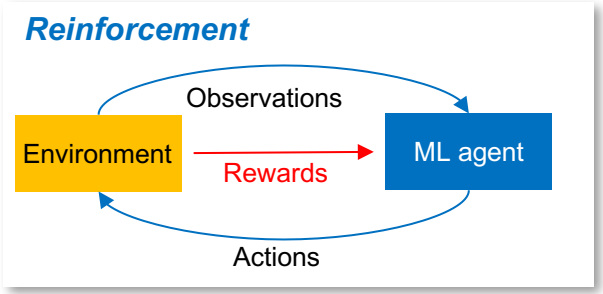
Task driven



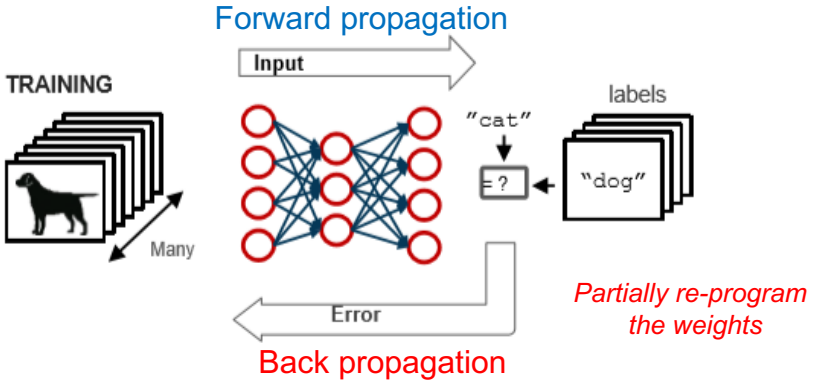
Data driven



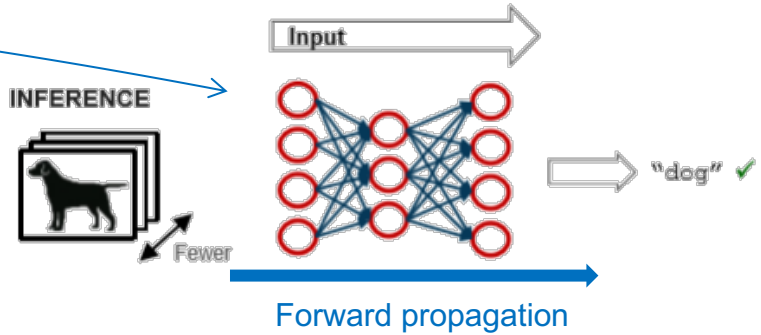
Learn from Mistakes



Supervised Machine Learning

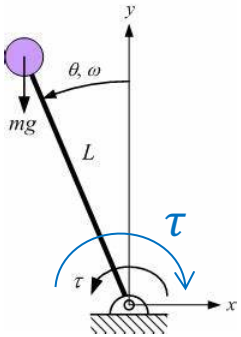


STEP 1 (Training): Process for machine to "learn" and optimize model from data, ***off-line.*** *Can be performed on CPU/GPU rarely on FPGA*



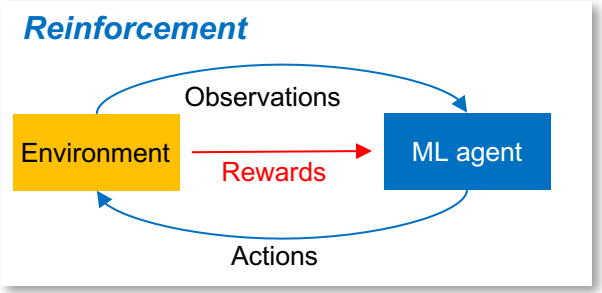
STEP 2 (Inference): Using trained models to predict/estimate outcome from new observations in efficient deployments, ***on-line.*** *Can be performed on CPU/GPU and FPGA*

Reinforcement ML – Inverted pendulum application

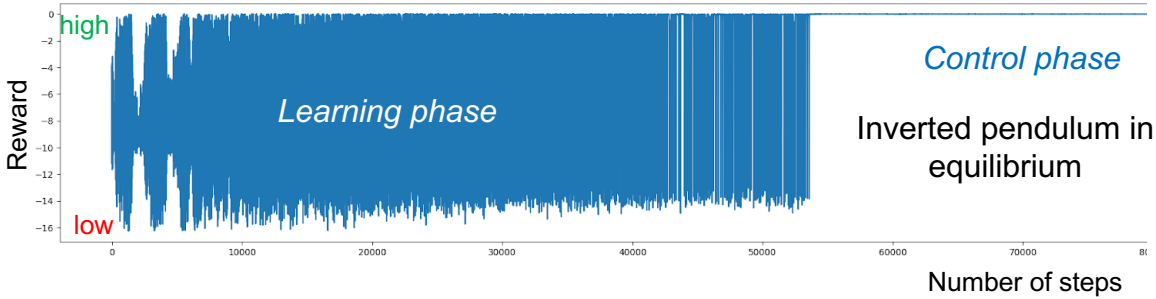


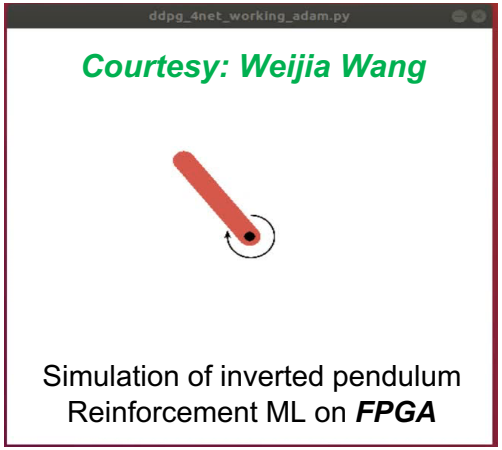
Inverted pendulum

- ***GOAL:*** to keep the pendulum in equilibrium
- ***Action:*** to apply a “torque” or “rotational force” (τ)
- Reinforcement ML: which learns from the rewards / mistakes
- ML running on FPGA

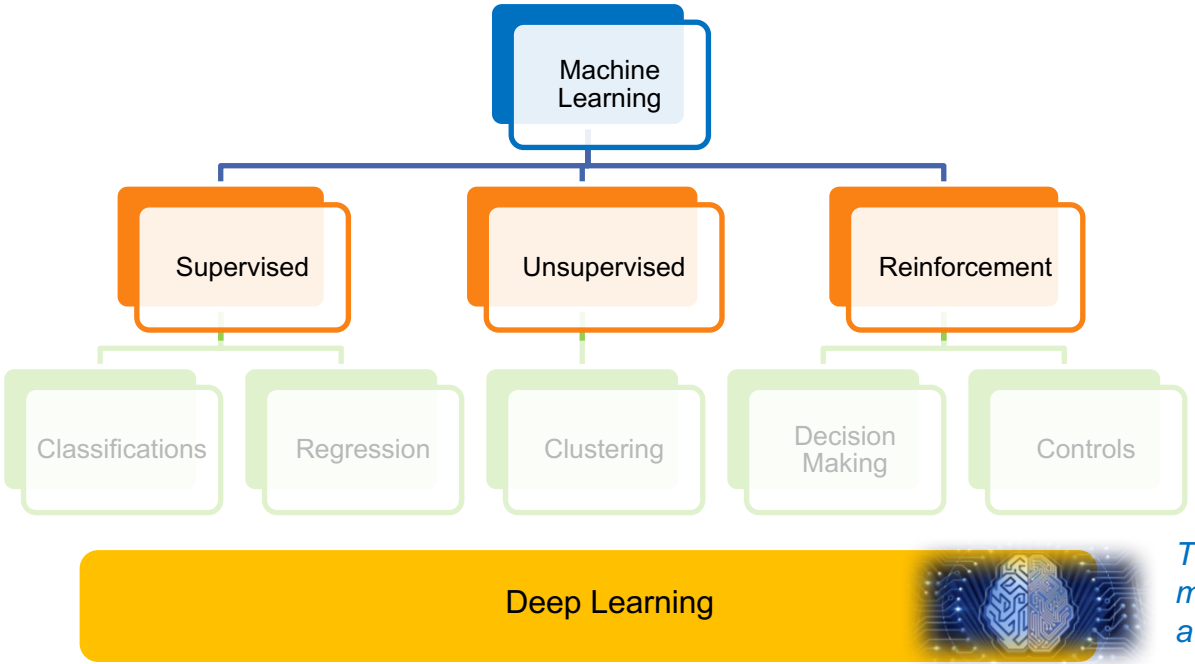


Rewards time evolution learning/controlling





Machine learning



Supervised: All data is *labelled* and the algorithms learn to predict the output from the input data.

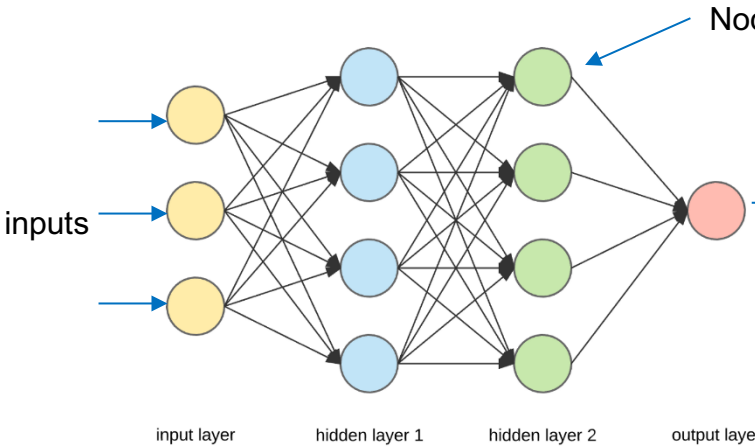
Unsupervised: All data is *unlabelled* and the algorithms learn to inherent structure from the input data

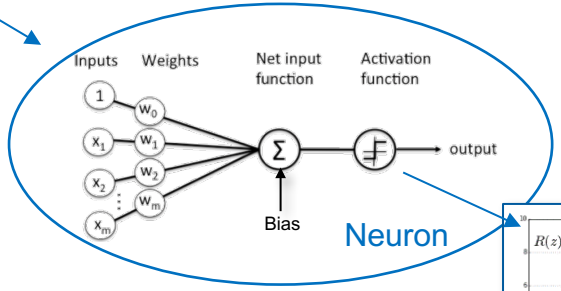
Reinforcement: The learning algorithm is trained not on present data but rather based on a *feedback* system.

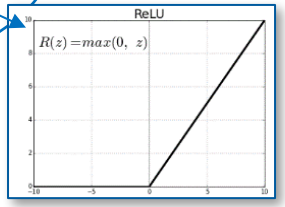
The idea behind a deep neural network is to mimic the biological structure of the brain with a similar structure with layers of artificial neurons

Deep learning is part of a broader family of machine learning methods based on artificial neural networks. Learning can be supervised, unsupervised and reinforcements

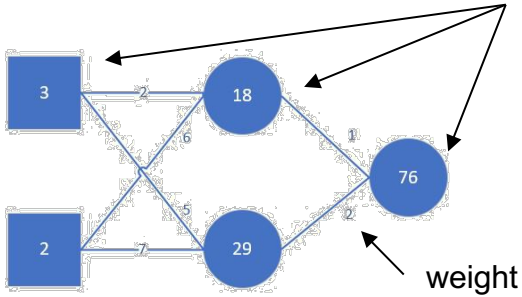
Deep Neural Network







ReLU activation function



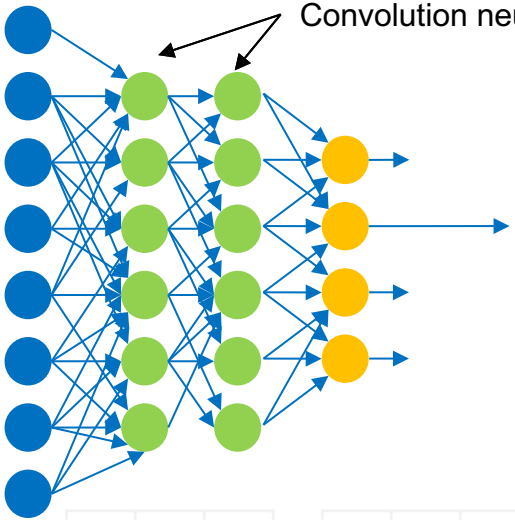
Very simple example of network

$$\begin{bmatrix} 3 & 2 \end{bmatrix} \times \begin{bmatrix} 2 & 5 \\ 6 & 7 \end{bmatrix} = \begin{bmatrix} 18 & 29 \end{bmatrix}$$

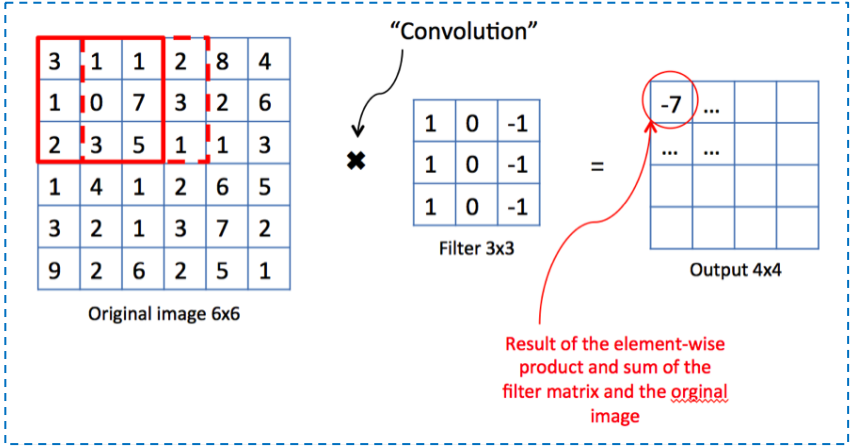
$$\begin{bmatrix} 18 & 29 \end{bmatrix} \times \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 76 \end{bmatrix}$$

Weight matrix

Deep Neural Network (Convolution layer)



Convolution



Juts a simple weight

-1	-1	-1
1	1	1
0	0	0

Filter 1

-1	1	0
-1	1	0
-1	1	0

Filter 2

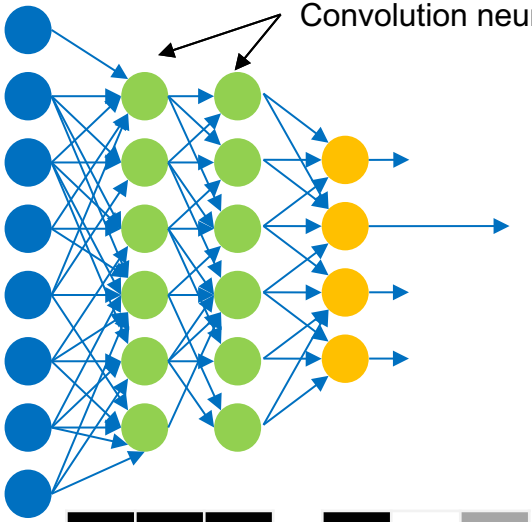
0	0	0
1	1	1
-1	-1	-1

Filter 3

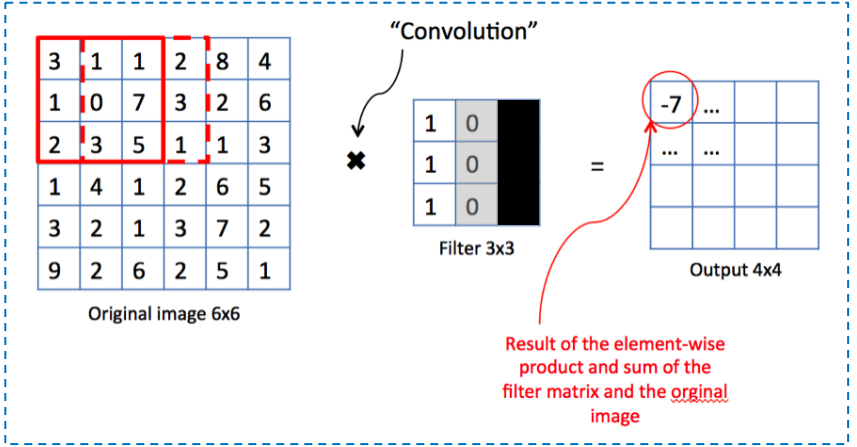
0	1	-1
0	1	-1
0	1	-1

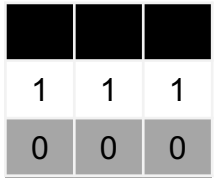
Filter 4

Deep Neural Network (Convolution layer)

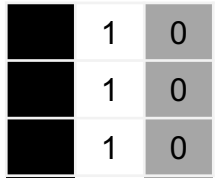


Convolution

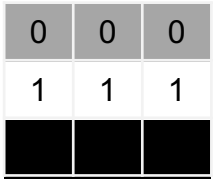




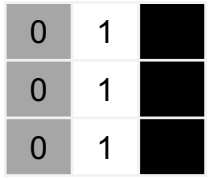
Filter 1



Filter 2

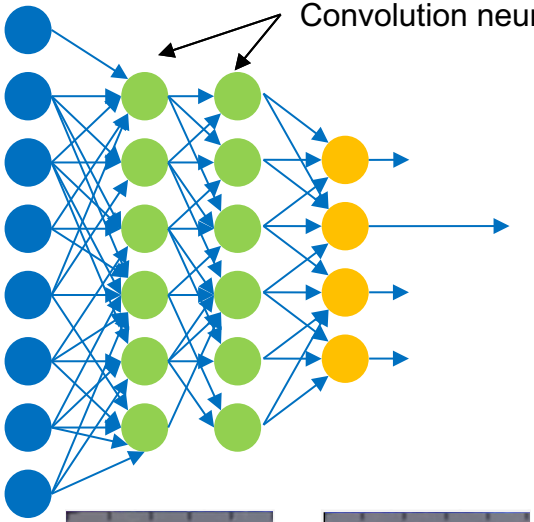


Filter 3



Filter 4

Deep Neural Network (Convolution layer)



Outputs

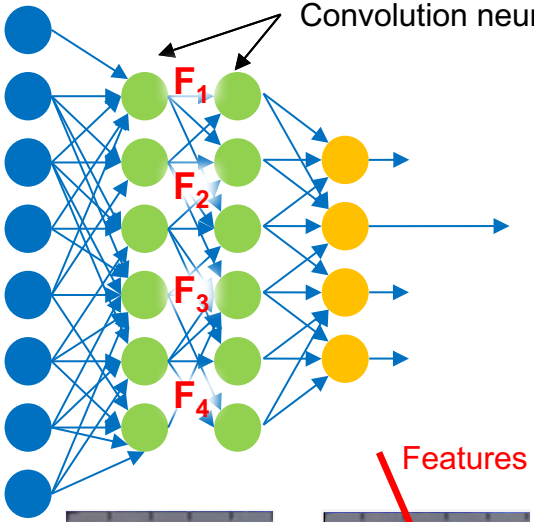

Filter 1


Filter 2


Filter 3


Filter 4

Deep Neural Network (Convolution layer)



Features 1

Filter 1

Features 2

Filter 2

Features 3

Filter 3

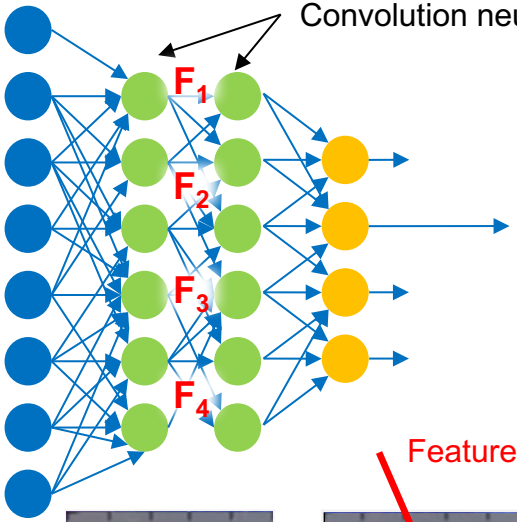
Features 4

Filter 4

The convolution layer will extract several “special features” from the original picture

Feature extraction

Deep Neural Network (Convolution layer)



All extracted features are combine in the output layer in order to recognize more complex structures

Features 1

Filter 1

Features 2

Filter 2

Features 3

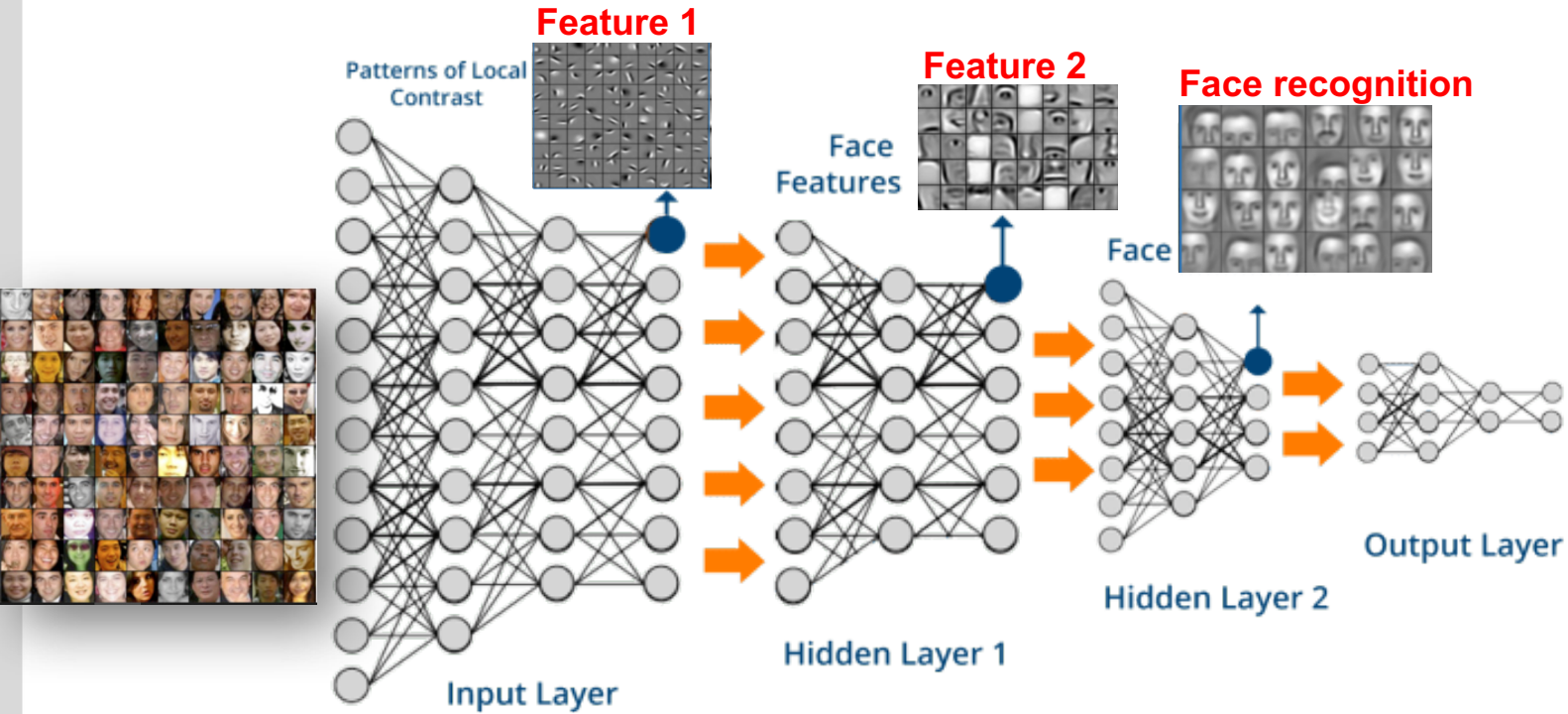
Filter 3

Features 4

Filter 4

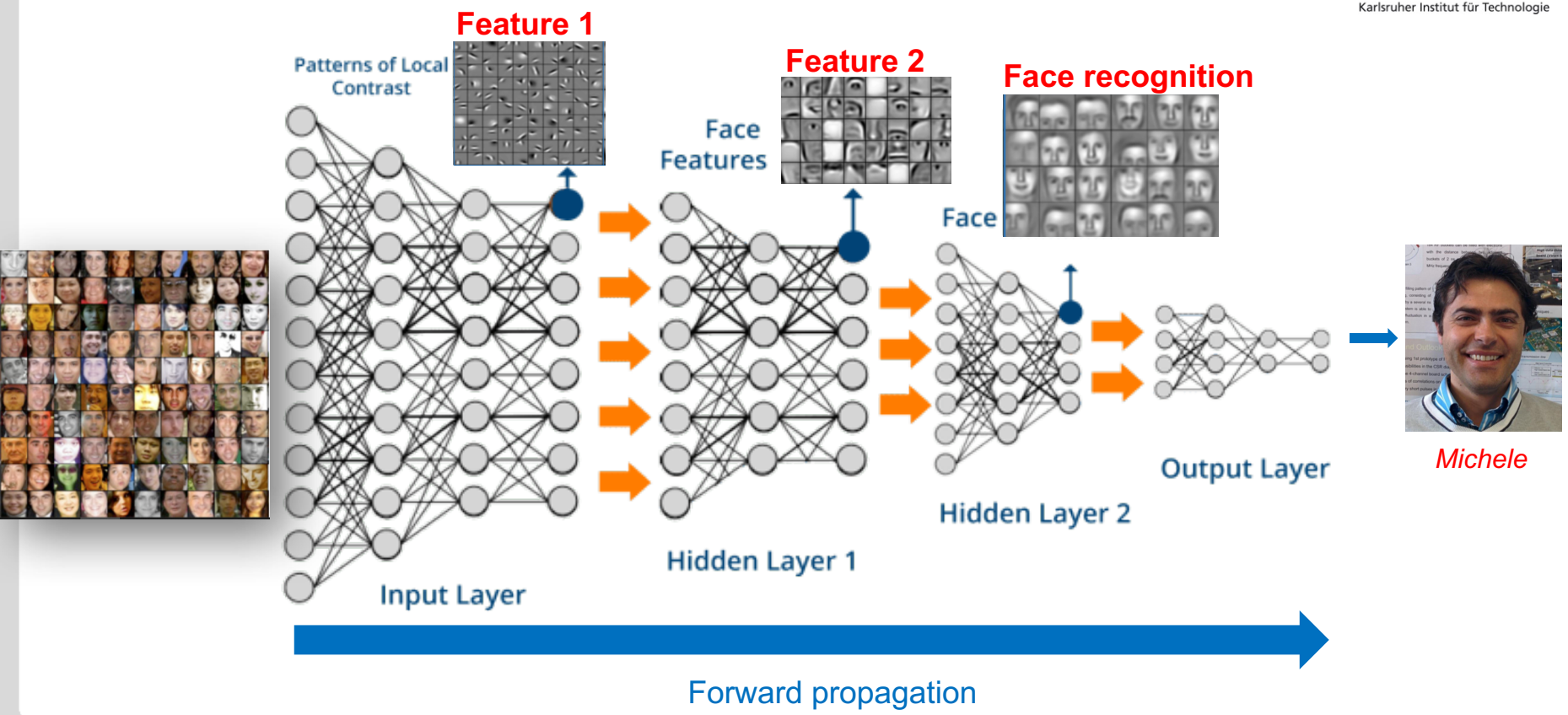
The convolution layer will extra a “special feature” from the original picture

Deep Neural Network (Convolution layer)



- A deep neural network consists of a hierarchy of layers, whereby each layer transforms the input data into more abstract representations (e.g. edge -> nose -> face). The output layer combines those features to make predictions

Deep Neural Network (Convolution layer)

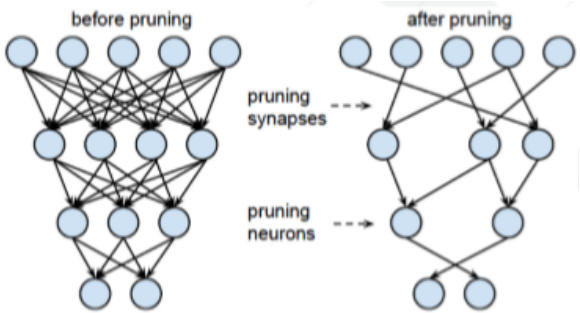


How to implement a fast ML inference on FPGA?

High-Performance DNN on FPGA

- DNN produces many parameters for the model, which increases the compute cost and requires high memory bandwidth. There are two main ways to optimize a DNN application:

Pruning: This is a form of DNN compression. It reduces the number of "synaptic" connections and "neurons" and so that the overall amount of data is reduced. Typically, weights close to zero are removed.



What we lose in accuracy?

Comparison *GPU (FP 32)* *vs* *FPGA (INT8)*

Top-5 Accuracy	FP-32	FIXED-16 (INT16)	FIXED-8 (INT8)	Difference vs FP32
VGG-16	86.6%	86.6%	86.4%	(0.2%)
GoogLeNet	88.6%	88.5%	85.7%	(2.9%)
SqueezeNet	81.4%	81.4%	80.3%	(1.1%)

Quantization: To bring the neural network to a reasonable size while also achieving high-performance accuracy. In this method, the process of approximating a neural network that uses floating-point numbers (FTP32) by a neural network of low-bit width numbers (INT8) is performed.

Machine learning on FPGA

Nowadays, several High Level tools are available to generate “fast” ML inferences running on FPGA

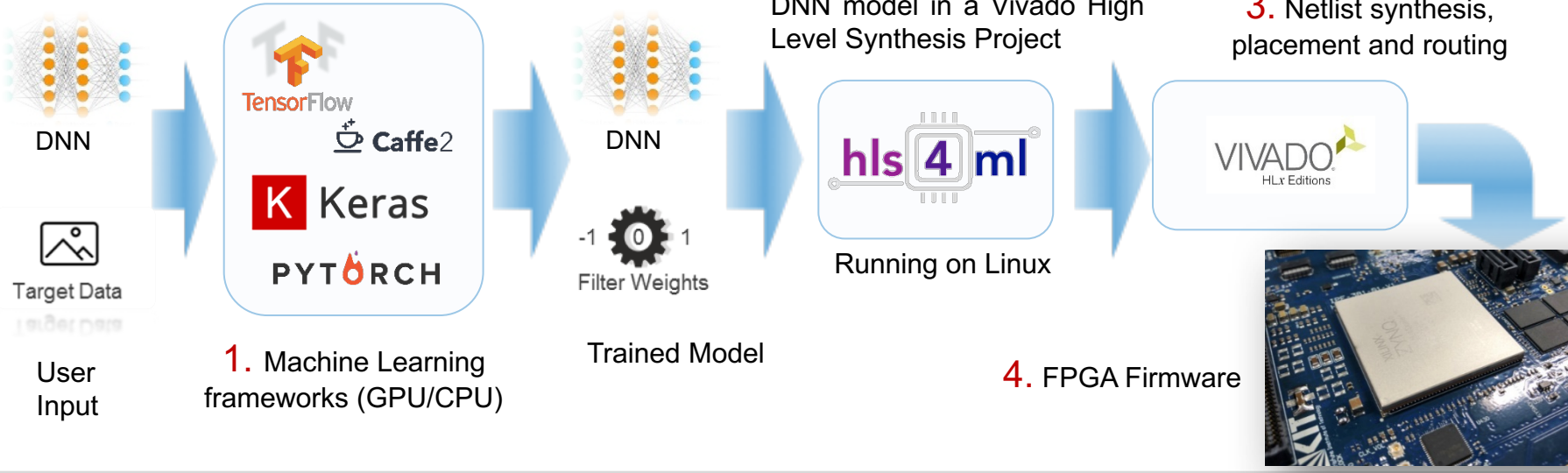
- High Level Synthesis four Machine Learning, developed at CERN
  - *What is **hls4ml*** → framework removes major barrier on hardware development of ML algorithms allowing developers *with little or no FPGA expertise* to program the FPGA
- Machine learning for Data Quality Monitoring (on-line), developed at CERN (HL-CMS)
  - Based on supervised learning → binary classification problem: good plot *vs* bad plot
- Xilinx Deep Learning Processing Unit (DPU) which is a configurable computation engine dedicated to convolutional neural networks
- …
- Custom frameworks: Snowflake: arXiv:1708.02579, DNNWeaver: http://act-lab.org/artifacts/dnnweaver/, fpgaConvNet: http://cas.ee.ic.ac.uk/people/sv1310/ fpgaConvNet.html, CNNECST, many others …

hls4ml – Workflow

- ***Very success framework***, developed at CERN for CMS jet classification is now integrated and distributed in the Xilinx Software Development Kit (XSDK)

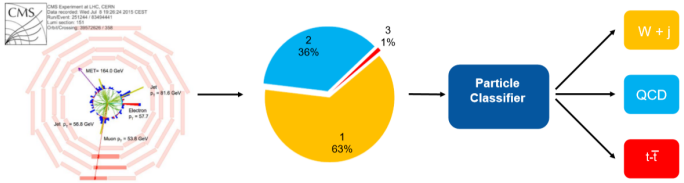
  Webpage: https://fastmachinelearning.org/hls4ml/

Hls4ml workflow



hls4ml – reconstruction chain for jet

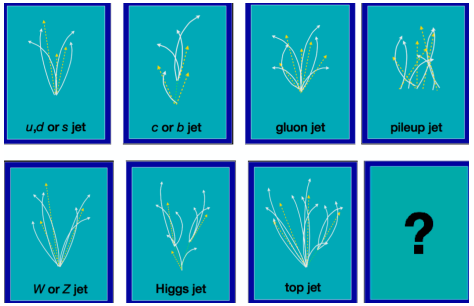
- **GOAL:** Machine Learning for low-level trigger system based on FPGA → fast jet substructure classification

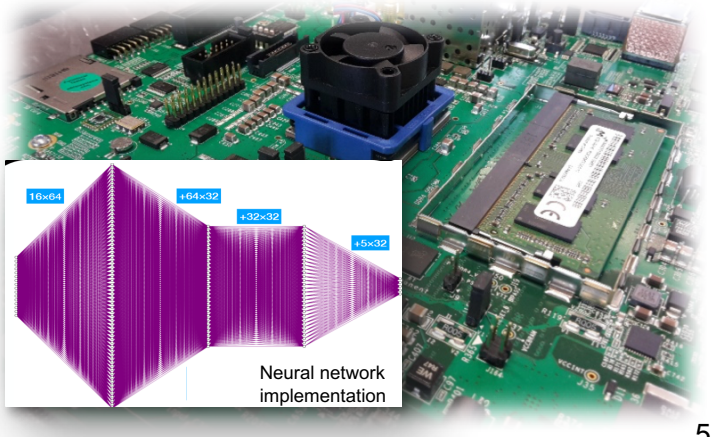


Monte Carlo data simulation

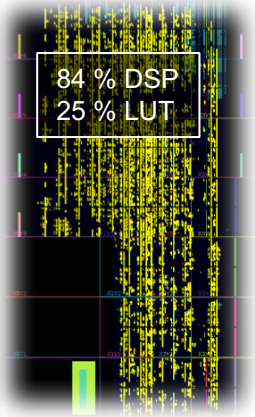
Jets classifications:

- *Gluon (g)*
- *Quark (q)*
- *Boson (W)*
- *Boson (Z)*
- *Top quark (t)*





Implementation on ZYNQ UltraScale+ XCZU9EG



5 layers of fully connected layers neural network

FPGA *vs* GPU

Res_V	FPGA prediction	Python Keras calculation (GPU)
Gluon (g)	0.118164	0.12993355
Quark (q)	0.639648	0.6487177
Boson (W)	0.118164	0.10633943
Boson (Z)	0.118164	0.10616959
top quark (t)	0.015625	0.00883975

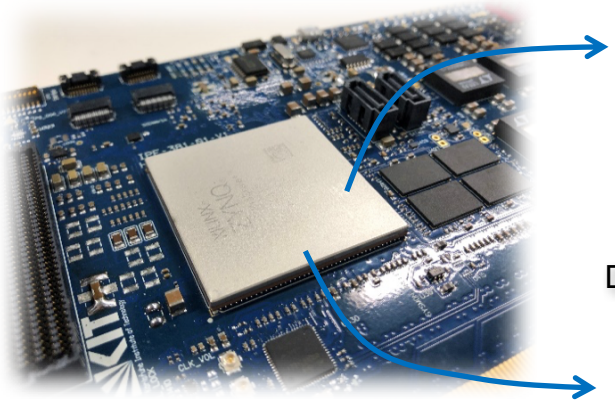
Latency time:

- *GPUs: of ~ tens µs*
- *FPGA: 16 clock cycles @ 200 MHz = 80 ns*

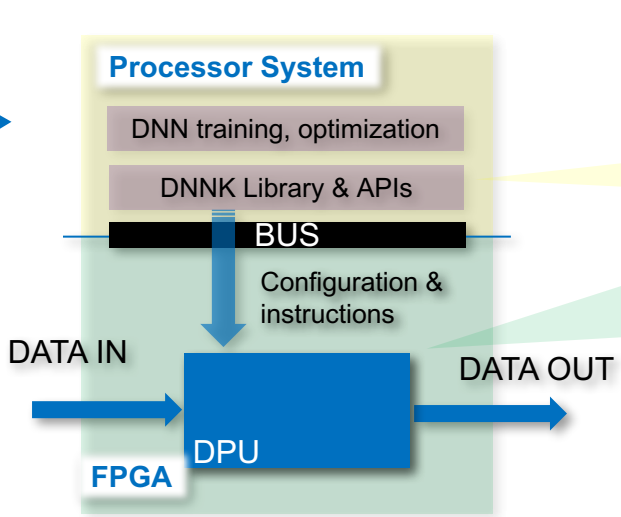
Courtesy: Weijia Wang

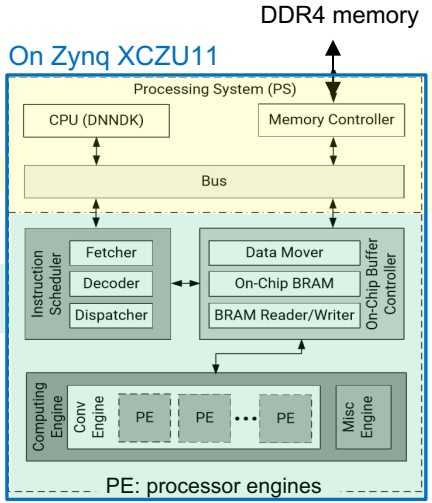
Xilinx Deep Learning Processing Unit (DPU)

- The Xilinx Deep Learning Processor Unit (DPU) is a configurable computation engine dedicated for convolutional neural networks.
- The DNNDK (Deep Neural Network Development Kit) is designed as an integrated framework, which aims to simplify and accelerate deep learning application development and deployment on the Deep Learning Processor Unit (DPU).



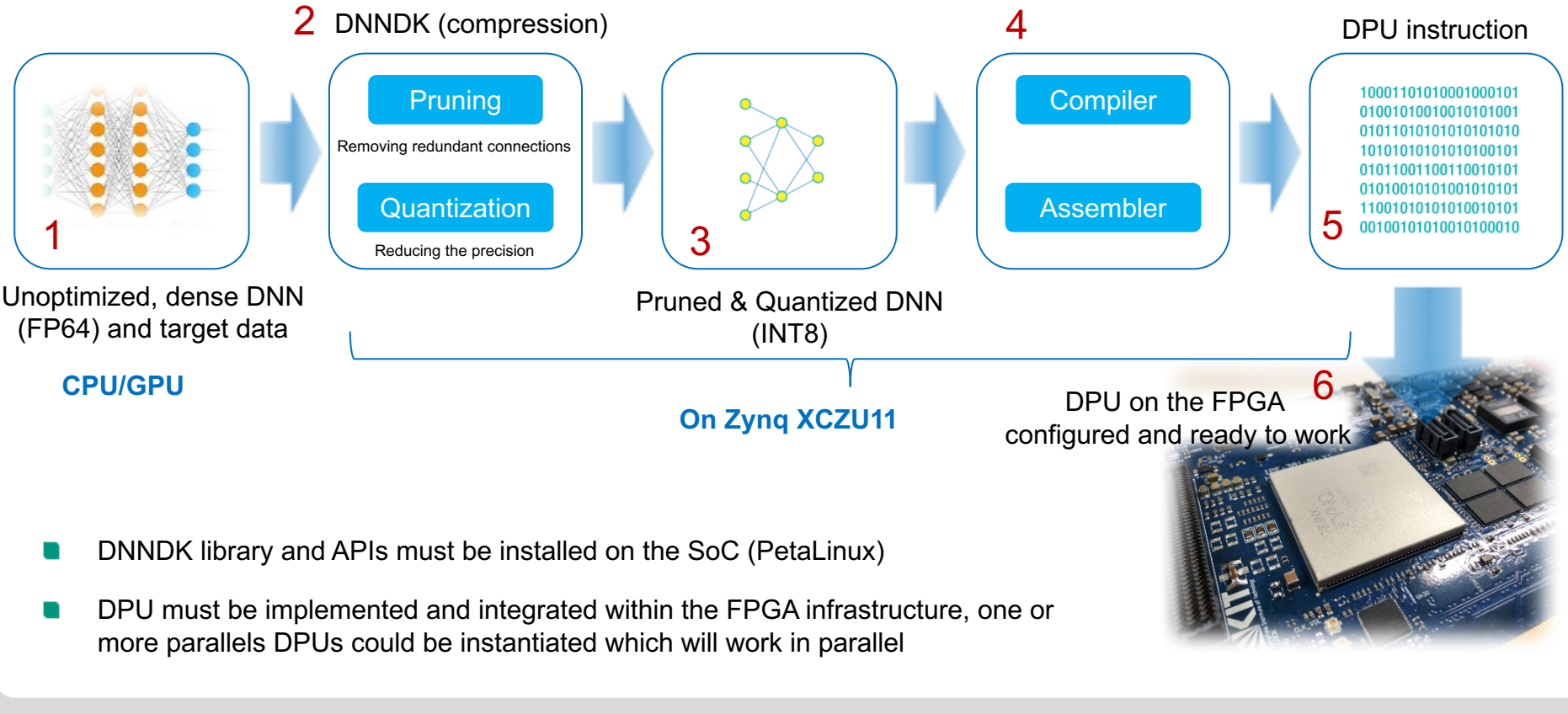
High-Flex 2 multi-purpose PCIe card
FPGA processor: Zynq XCZU11





Internal architecture

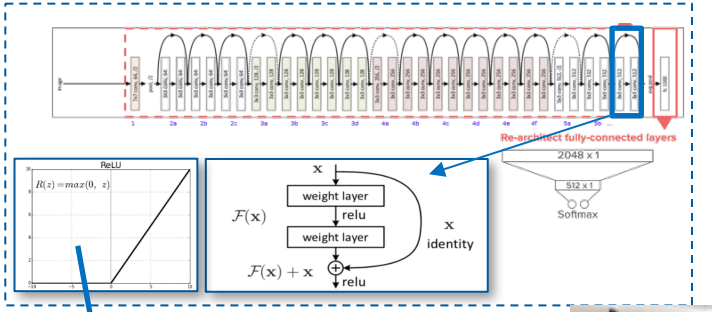
Xilinx Deep Learning - Workflow



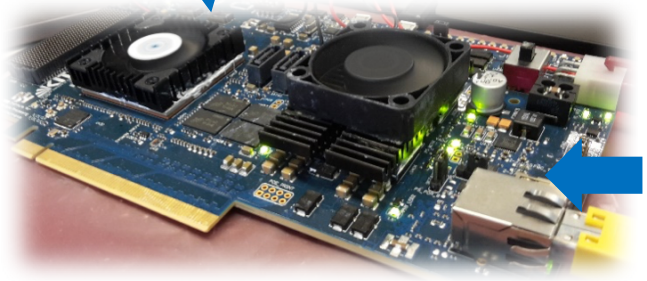
- DNNDK library and APIs must be installed on the SoC (PetaLinux)
- DPU must be implemented and integrated within the FPGA infrastructure, one or more parallels DPUs could be instantiated which will work in parallel

Examples: Objects recognition/classification

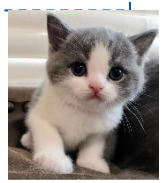
- Neural network implemented resnet50
- Xilinx Deep Learning Processor Unit (DPU)



Courtesy: Weijia Wang



High-Flex 2 multi-purpose PCIe card



cat.jpg

cab.jpg

xian.jpg

HF-2.jpg

Linux + drivers + Resnet 50 (DNN) → on-FPGA

```
Save
weijia@weijia: ~/projects/zcu102-dpu-trd-2019-1/zcu102-dpu-trd-2019-1-timer/images
File  Edit  View  Search  Terminal  Tabs  Help
weijia@weijia: ~/projects/zcu102-...   weijia@weijia: ~/projects/zcu102-...   weijia@weijia: ~/projects/zcu102-...
KIT_shuttle.jpg    Xian.jpg          cat.jpg
root@hiflex2-peta-prj-0220:~/resnet50/image# rm KIT_shuttle*
root@hiflex2-peta-prj-0220:~/resnet50/image# ls
HiFlex2_new.jpg  Xian.jpg         cab.jpg          cat.jpg          words.txt
root@hiflex2-peta-prj-0220:~/resnet50/image# cd ..
root@hiflex2-peta-prj-0220:~/resnet50# ./resnet50

##################################################
Warning:
The DPU in this TRD can only work 8 hours each time!
Please consult Sales for more details about this!
##################################################

total image : 4

Load image: cat.jpg
[Top 0] prob = 0.379113  name = Egyptian cat,
[Top 1] prob = 0.229943  name = tabby, tabby cat,
[Top 2] prob = 0.108618  name = lynx, catamount,
[Top 3] prob = 0.065880  name = Siamese cat, Siamese,
[Top 4] prob = 0.024236  name = dishwasher, dish washer, dishwashing machine,
```

```
Load image: cab.jpg
[Top 0] prob = 0.538945  name = cab, hack, taxi, taxicab,
[Top 1] prob = 0.093655  name = trailer truck, tractor trailer, trucking rig, rig, articulated
, semi,
[Top 2] prob = 0.093655  name = school bus,
[Top 3] prob = 0.093655  name = police van, police wagon, paddy wagon, patrol wagon, wagon, bl
ria,
[Top 4] prob = 0.034454  name = tow truck, tow car, wrecker,
```

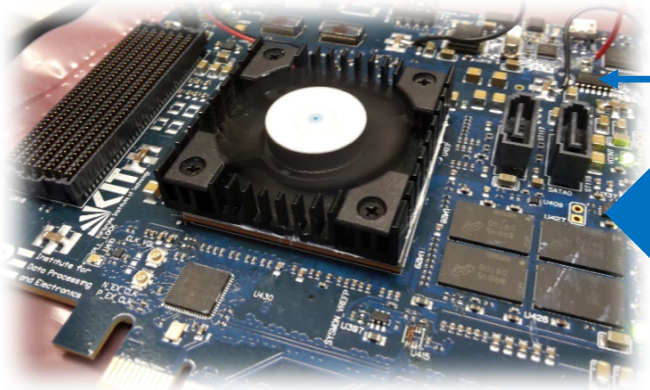
```
Load image: Xian.jpg
[Top 0] prob = 0.716057  name = bell cote, bell cot,
[Top 1] prob = 0.159774  name = palace,
[Top 2] prob = 0.096908  name = castle,
[Top 3] prob = 0.007955  name = church, church building,
[Top 4] prob = 0.003758  name = monastery,

Load image: HiFlex2_new.jpg
[Top 0] prob = 0.638194  name = hard disc, hard disk, fixed disk,
[Top 1] prob = 0.182846  name = oscilloscope, scope, cathode-ray oscilloscope, CRO,
[Top 2] prob = 0.024745  name = tape player,
[Top 3] prob = 0.024745  name = modem,
[Top 4] prob = 0.024745  name = cassette,
```

No high-flex trained data

Examples: Advanced driver-assistance systems (ADAS)

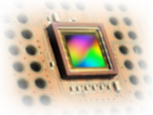
- **GOAL:** Real-time recognition of cars, trucks, people, bikes, motorcycles, etc.



High-Flex 2 multi-purpose PCIe card

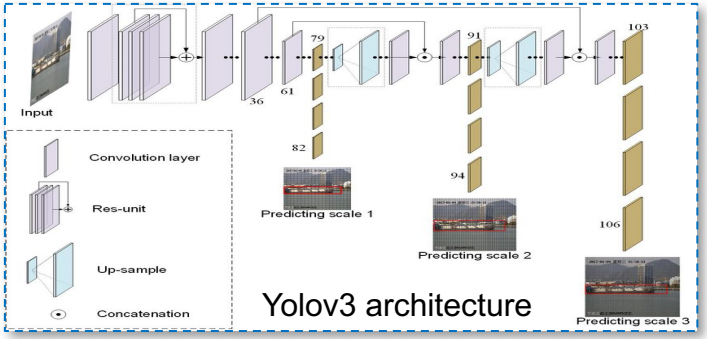
ETH

Video stored in the on-board DDR4 memory and processed by **yolov3detection** on FPGA



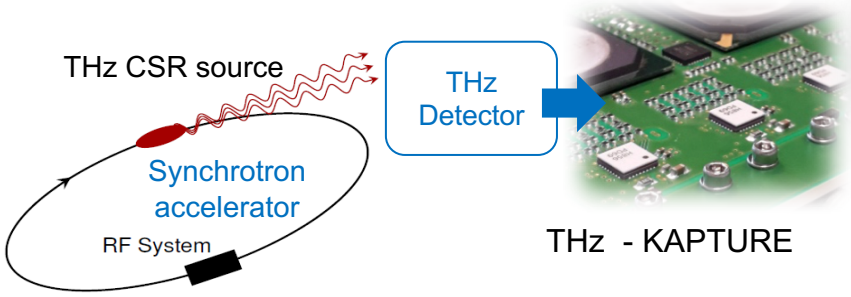
UFO Camera

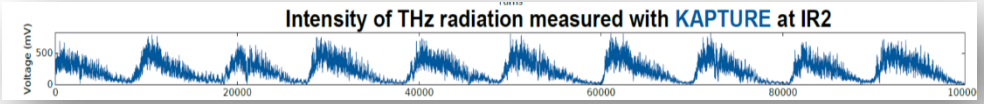
Courtesy: Weijia Wang



Advanced beam control (KARA)

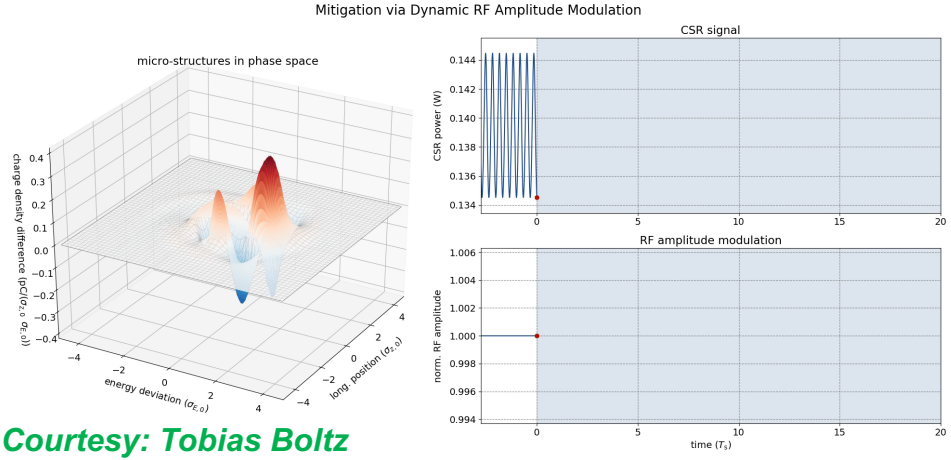
- ***Goal:*** to keep the coherent synchrotron radiation (CSR) THz intensity stable by fast feedback to RF System
- Complex and nonlinear dynamics in longitudinal / transverse bunch profiles → described by a nonlocal nonlinear partial differential vlasov fokker planck equation describing the time evolution of the probability distribution of a particle in synchrotron machine.





THz intensity is not stable period about ms

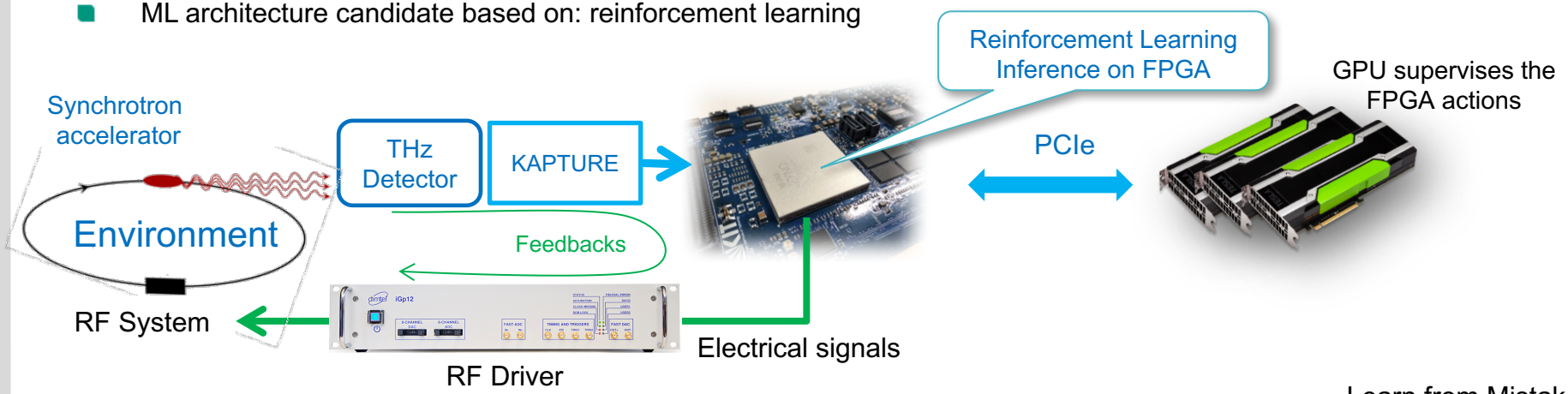
- We would like to **control the phase space of the bunches by ML on FPGA**
- **Action**: to control the nonlinear dynamics of the phase space of the beam by a modulation of the RF system
- **RF System parameters are**: RF Amplitude, phase and frequency



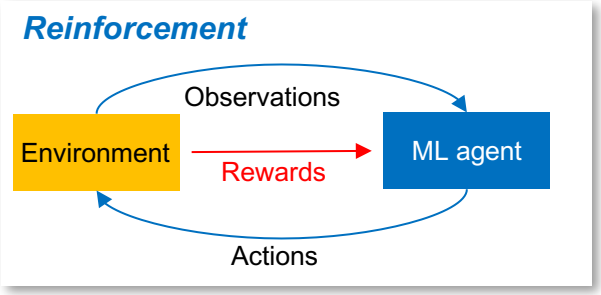
Courtesy: Tobias Boltz

Advanced beam control (KARA) - Reinforcement

- Complex and multi-dimensional control system working in real-time
- ML architecture candidate based on: reinforcement learning

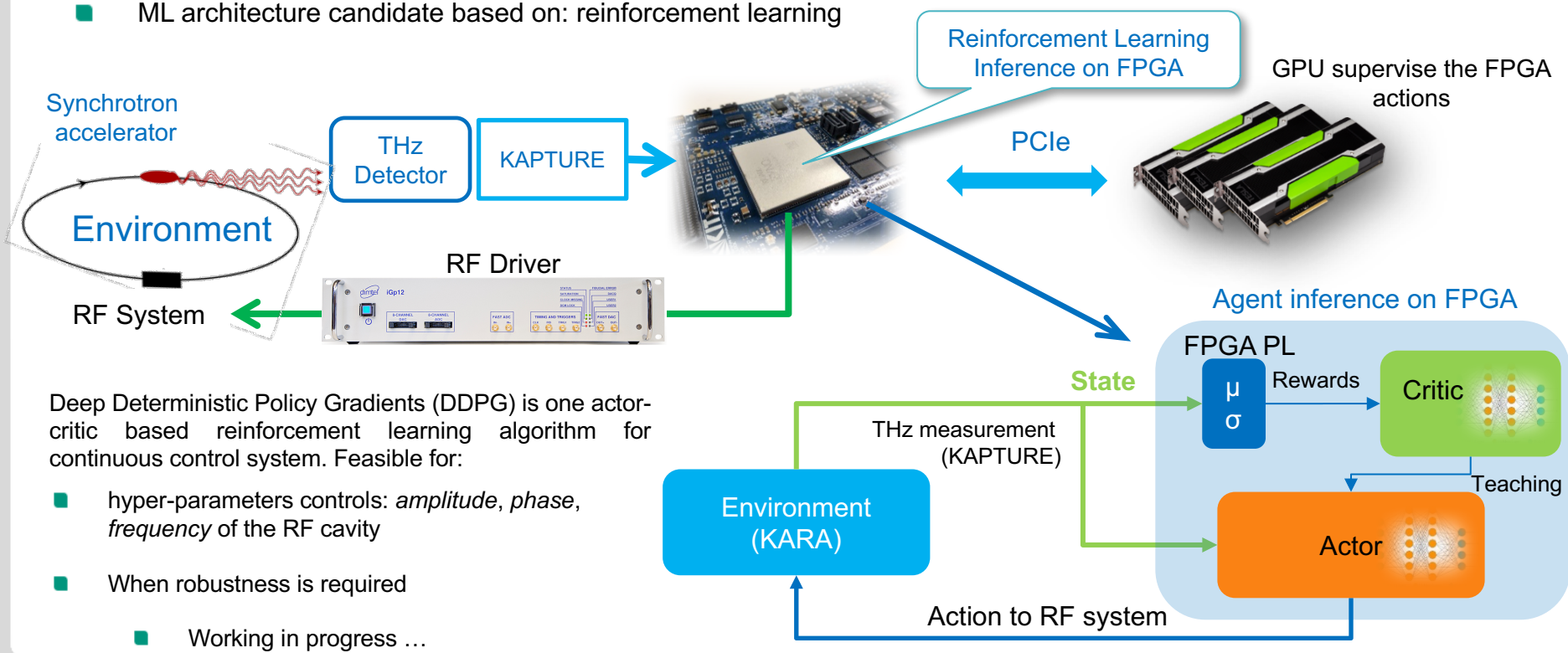


Learn from Mistakes



Advanced beam control (KARA) - Reinforcement

- Complex and multi-dimensional control system working in real-time
- ML architecture candidate based on: reinforcement learning

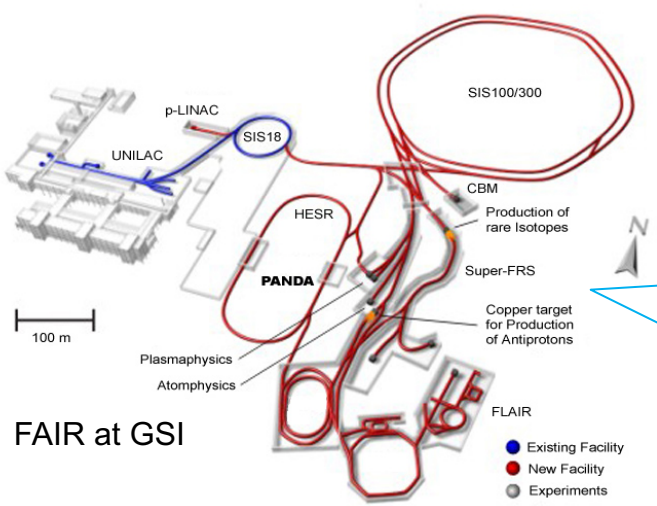


Deep Deterministic Policy Gradients (DDPG) is one actor-critic based reinforcement learning algorithm for continuous control system. Feasible for:

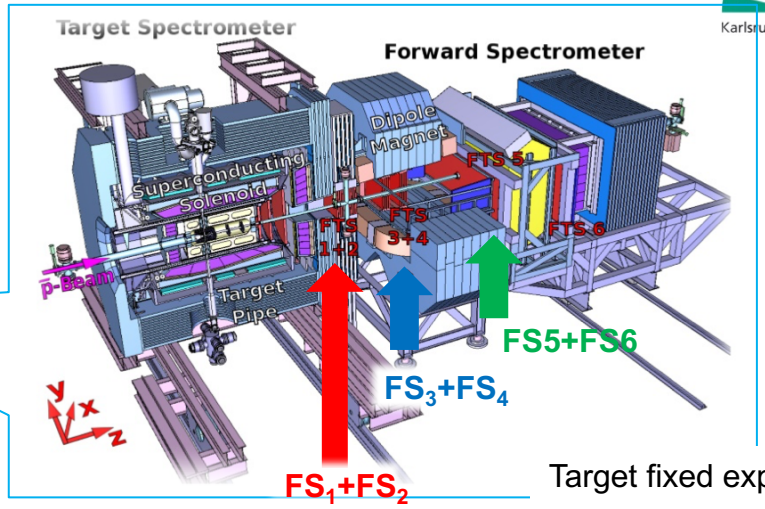
- hyper-parameters controls: *amplitude*, *phase*, *frequency* of the RF cavity
- When robustness is required
  - Working in progress …

Courtesy: Weijia Wang and Tobias Boltz

PANDA – fast track-reconstruction



FAIR at GSI



Target fixed experiment

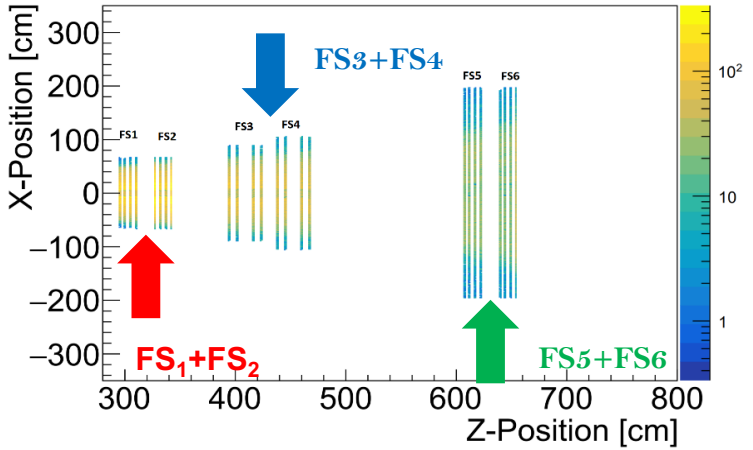
- Trigger-less readout system (no hardware trigger)
- **GOAL**: to explore ML for fast-track reconstruction of PANDA
  - Forward Tracking System detector
  - Barrel Tracking detector

Courtesy: Waleed Esmail (Forschungszentrum Jülich) and Weijia Wang

Two-steps fast-track reconstruction

- **STEP 1**: Find “local” *Track Segments* by DNN (under investigation by unsupervised ML)

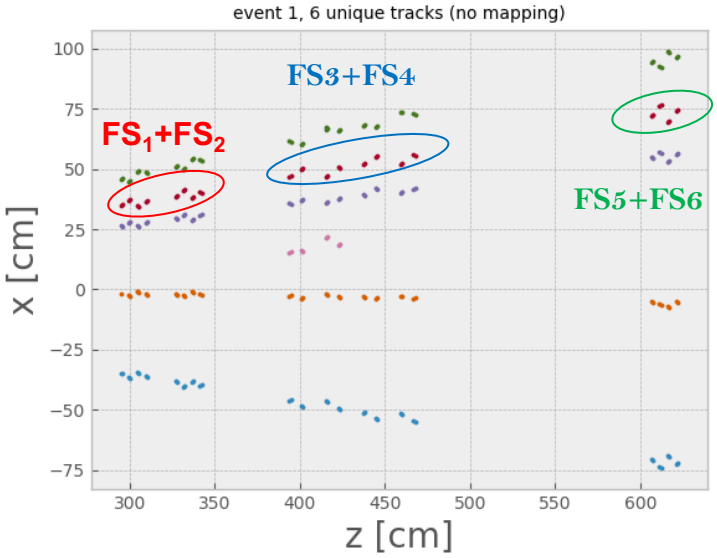
Monte Carlo event simulation (Forward Tracking System)



Two-steps tracking reconstruction

- **STEP 1**: Find “local” *Track Segments* by CNN (under investigation by unsupervised ML)

To find local track segments

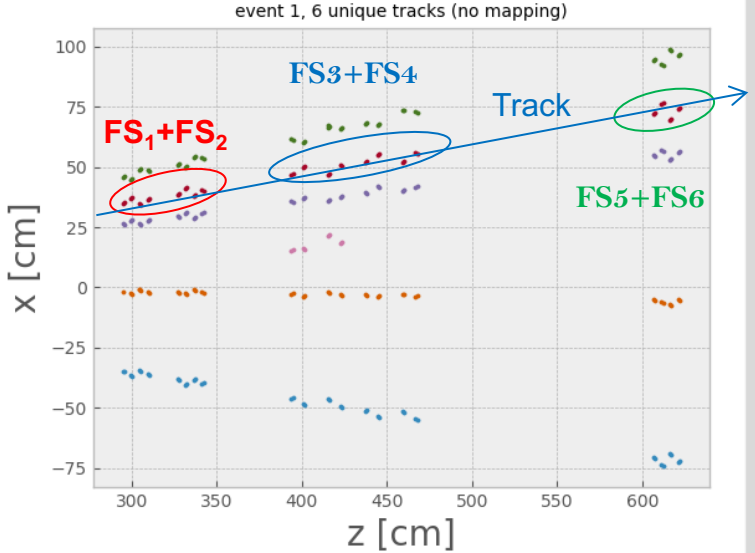


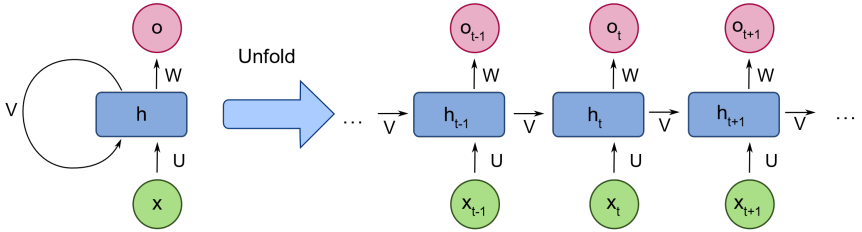
Two-steps tracking reconstruction

- By Unsupervised (*K*-means) and supervised (recurrent neural network) learning:
  - **STEP 1**: Create *Track Segments* by using K-means clustering algorithm

    To find local track segments

  - **STEP 2**: interpolate the track segments from the different parts of the FTS to form a full track candidate, it is based on a Recurrent Neural Network (RNN)





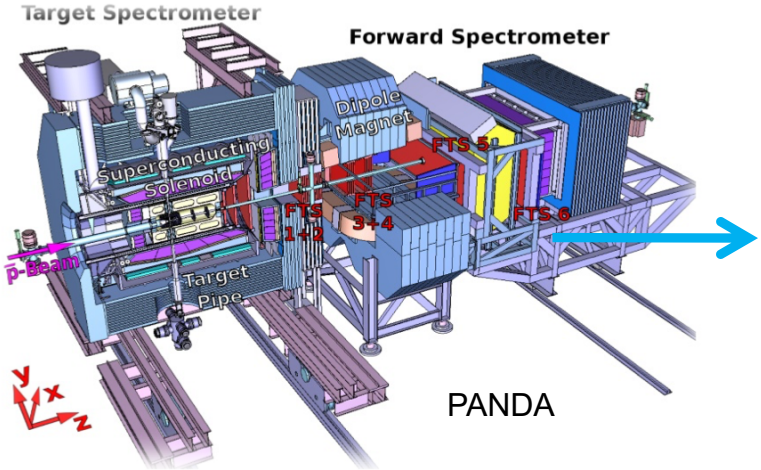
Temporal dynamic behavior

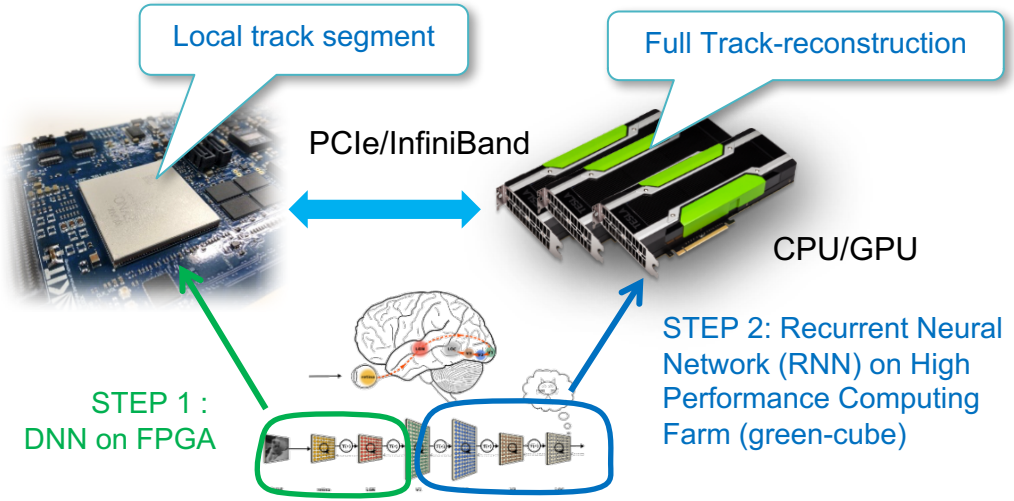
Recurrent neural networks have a memory that enables them to remember important events that happened many time steps in the past.

For: handwriting recognition or speech recognition.

PANDA – A possible ML implementation

- Distributed AI architecture running on heterogenous FPGA-GPUs infrastructure, local track reconstruction on FPGA and RNN on GPUs and integrated within the on-line event selection framework





Courtesy: Waleed Esmail (Forschungszentrum Jülich) and Weijia Wang

	FT1,FT2	FT3,FT4	FT5,FT6
Purity	99%	99%	99%
Efficiency	96%	95%	96%

Local track segment "ONLY" Preliminary

- The ***purity*** specifies which fraction of hits in one track come from the correct particle.
- The ***efficiency*** is defined as the ratio of the number of correctly reconstructed tracks to all generated tracks.

Impact

… we are just at the beginning of a new disruptive technology …

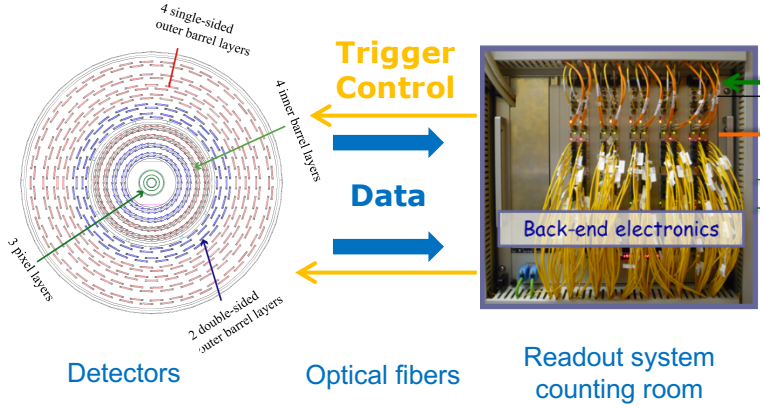
- **Motivations** - Next generation of detectors will generate huge and complex data volume (petabyte/sec), serious technological challenges → to push the technology envelope (especially for trigger and data acquisition)
  - **Machine learning** is one prominent technique for data processing (trigger, fast-track reconstruction), intelligent detector configuration/calibrations, data quality monitor, etc.
    - **Novel programmable** devices families with dedicated *AI engines* opens new prospective for future detectors.
      - **New generation of development tools** *for users without knowledge of deep-learning or FPGA*.

What's about FPGA also for data analysis ?

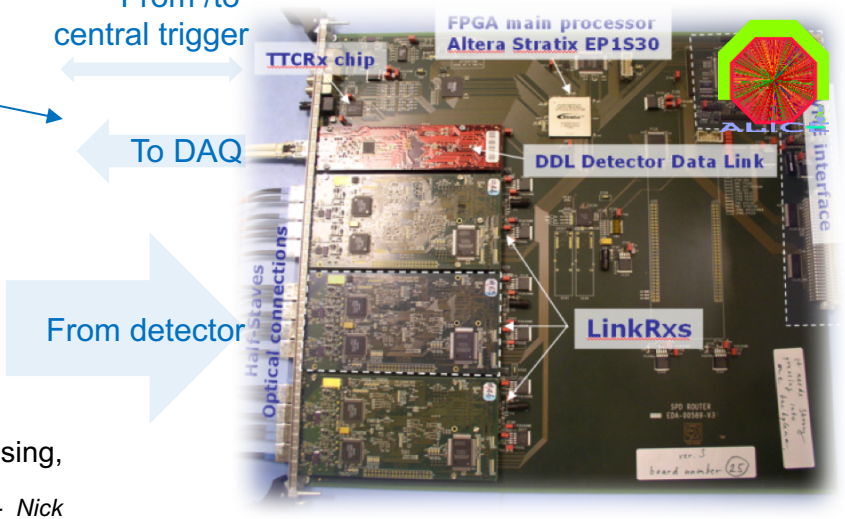
Thank you for your attention

Backup slides

FPGA in detectors



Back-end readout card for Alice Pixel Detector (CERN 2008-2010)



FPGA (*Field Programmable Gate Arrays*) is intensively employed in:

- Back-end readout electronics from detectors
- Real-time data processing, i.e. signal filtering, fast Fourier processing, etc. (*Wideband Readout System for Microwave-Resonator Multiplexed Sensors - Nick Karcher)*
- Low-level trigger process for real-time event filtering (*FPGA-based real-time track reconstruction for the CMS phase-2 tracker upgrade -Luis Eduardo Ardila Perez)*

VME –CERN cards with # 5 FPGAs: Real-time data processing, merging and formatting and low-level trigger

Three Ages of FPGA

Age of Invention

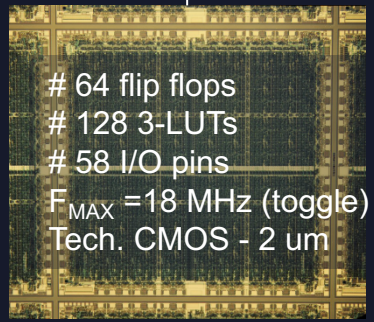
- FPGAs are much smaller than the application problem size
- No tools, manual P&R

Age of accumulation

- Communication is main market
- Dedicated logic blocks (e.g. high-speed I/O, multiplier)
- Low cost FPGAs
- IP cores for large FPGAs

Age of heterogeneous

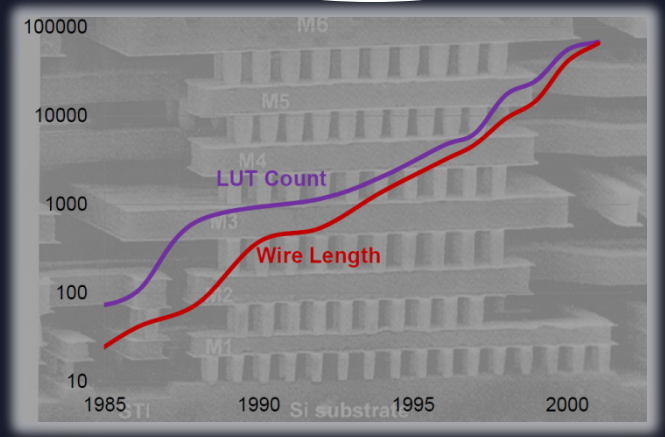
1984 1992 2000 2008 2014



XC2064 first FPGA (1985)

Age of expansion

- Rapidly growing FPGA sizes
- Increasing Demand for Design Automation
- SRAM FPGAs first for new technology – domination



Charting an Aggressive Course Forward

Courtesy of Hanak, Georg (Avnet Silica)

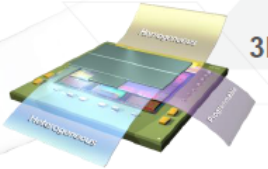
Surpassing Moore's Law

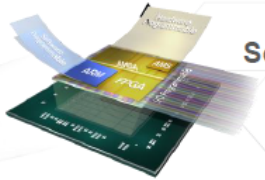
Unprecedented Capacity & Performance

System Intelligence & Integration

Silicon & Architecture for Price/Performance/Watt

Break Out





Low price

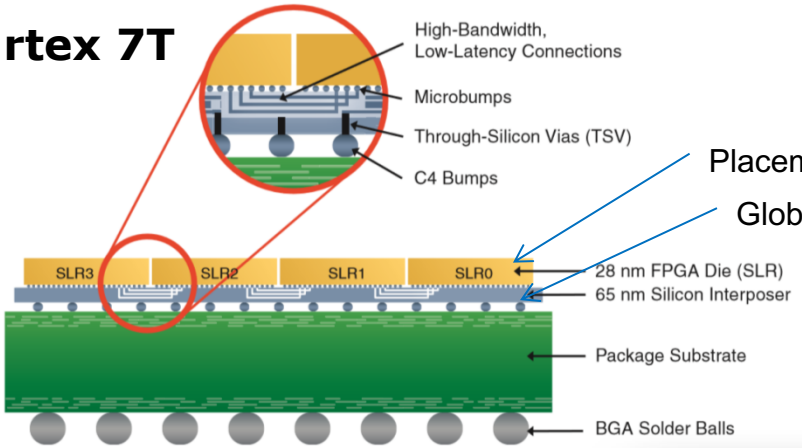
Next technology

Tools	VIVADO	UltraFAST Design Methodology	SDx	**VERSAL**
Architecture	7 Series	UltraScale™ Architecture		**ACAP**
Process	28HPL	20SoC	16FinFET+	7nm

XILINX.

Two FPGA technology solutions

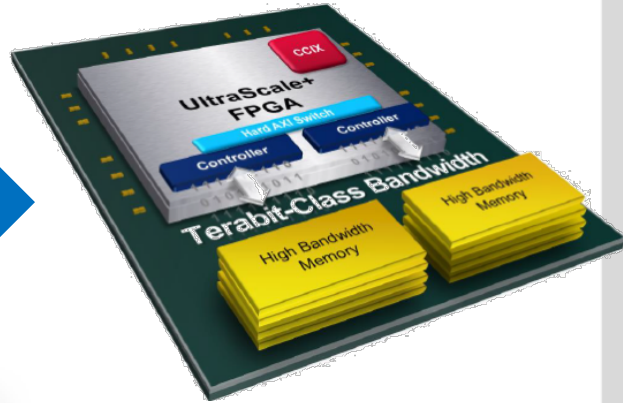
Virtex 7T



Virtex 7T FPGA Enabled by SSI (Stacked Silicon Interconnect) technology

Solution

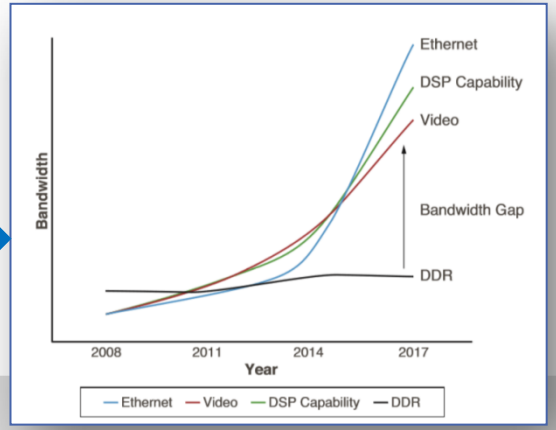
Virtex Ultrascale+



HBM (High Bandwidth Memory) integrated into the FPGA

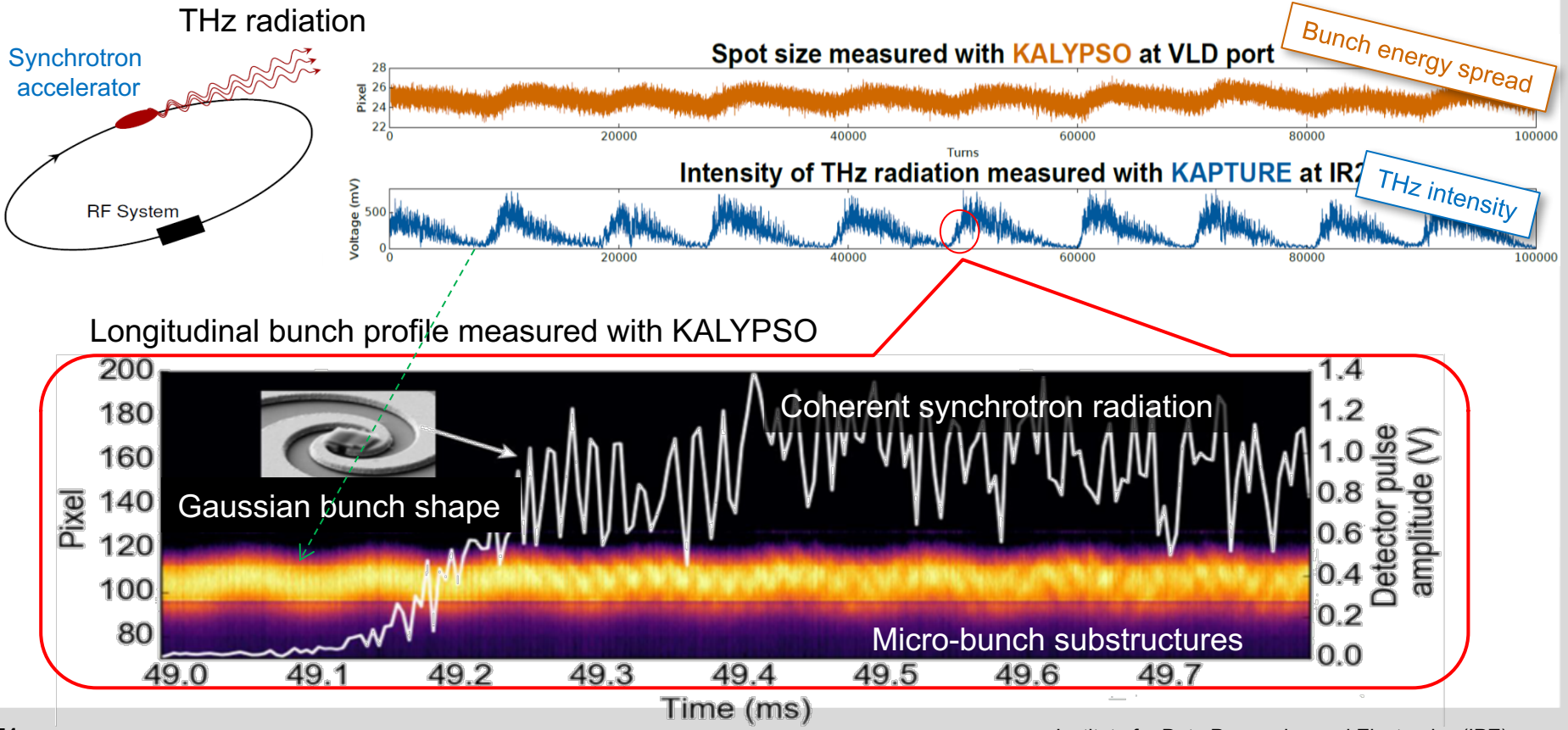
- DRAM: up to 4 GByte
- Bandwidth: 1.84 Tb/s

DDR memory device bandwidth GAP



Ref. Xilinx white paper WP380

High-performance Deep Machine Learning (II)



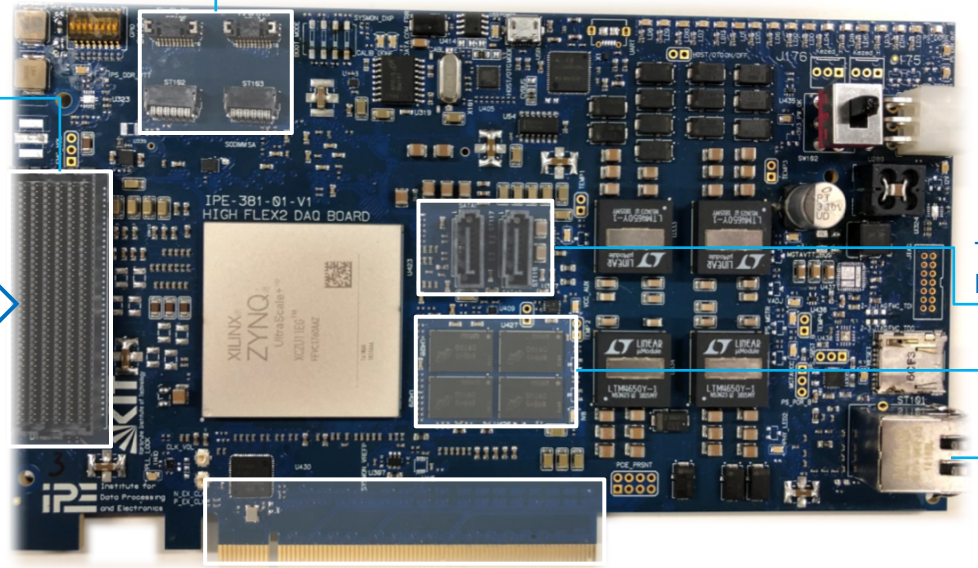
New Readout card

Optical link: 12 lanes @ 28 Gb/s (full-duplex) > 330 Gb/s.

SODIMM-DDR4 for PS

FMC+ 57.4 connector:
32 transvers @ 28 Gbps
160 lines @ 2 Gbps

µUSB + UART connectors



Detector

Two **SATA** connections:
Data saved directly on SSDs

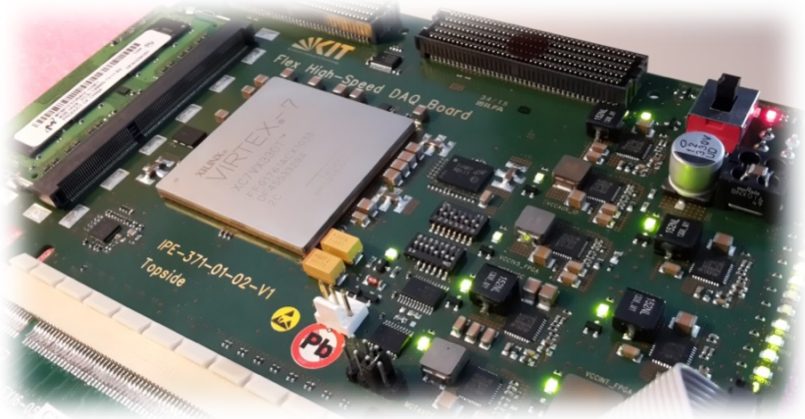
DDR4 for PL (FPGA)

ETH connected to processor (TCP/IP)

PCIe Gen 3 and 4, 16 lanes → up to 240 Gb/s full-duplex

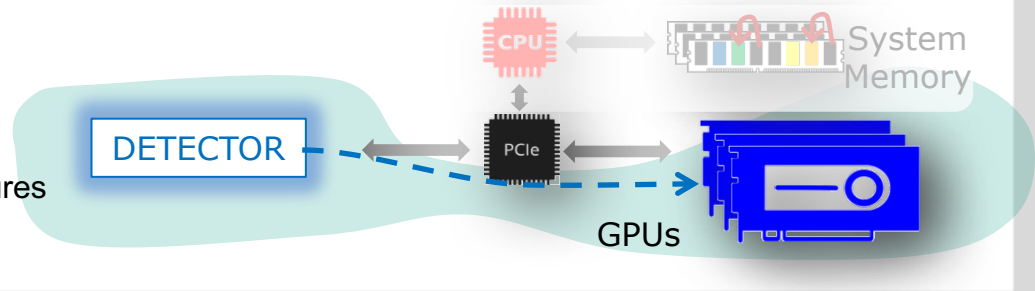
High-performance heterogeneous FPGA-GPU DAQ

- Modern photon science detectors generate huge raw data volumes (~120 Gb/s)
- Observed slow changes in synchrotron machine (e.g. current) → sec - hrs



PCIe readout card of UFO DAQ Platform

- Heterogeneous FPGA/GPU-based readout system, the UFO DAQ platform, has been developed
  http://ufo.kit.edu/ufo
- The core component is a “novel” Direct Memory Access (DMA) architecture
- Direct FPGA ↔ GPU communication enables real-time data processing



- DMA working close to theoretical limit of data link
- Data latency 5 times better than other DMA architectures

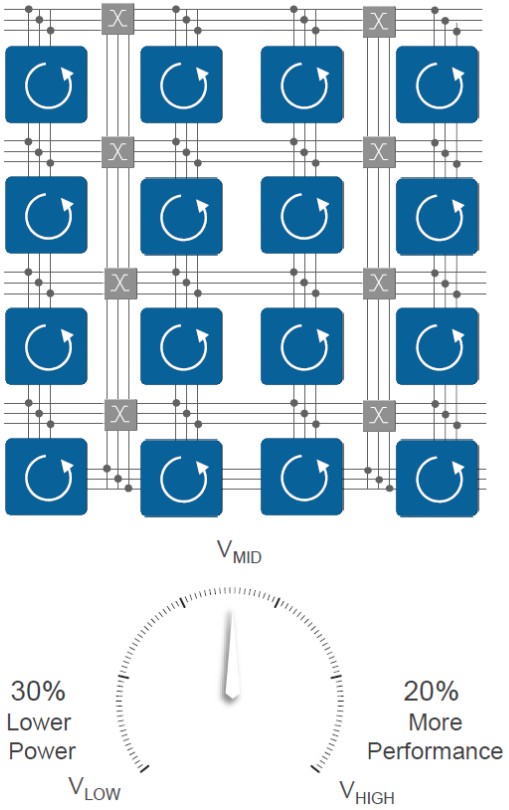
M. Caselle et al., JINST 12 C03015 (2017)

Versal – Adaptable Engines

For traditionalists: This is the FPGA part

Some known facts

- 6 Input LUTs
- **Each CLB has 32 LUTs and 64 FF (4x density compared to US+)**
- 16 LUTs in a slice can be
  - a 64 bit RAM
  - 32-bit shift registers (SRL32) or two SRL16
- **Internal connection of LUTs possible**
- 4x clock, 4x set/reset, 16 clock enable
- **3 step voltage-scaling supported**



Versal – AI tile architecture

Karlsruher Institut für Technologie

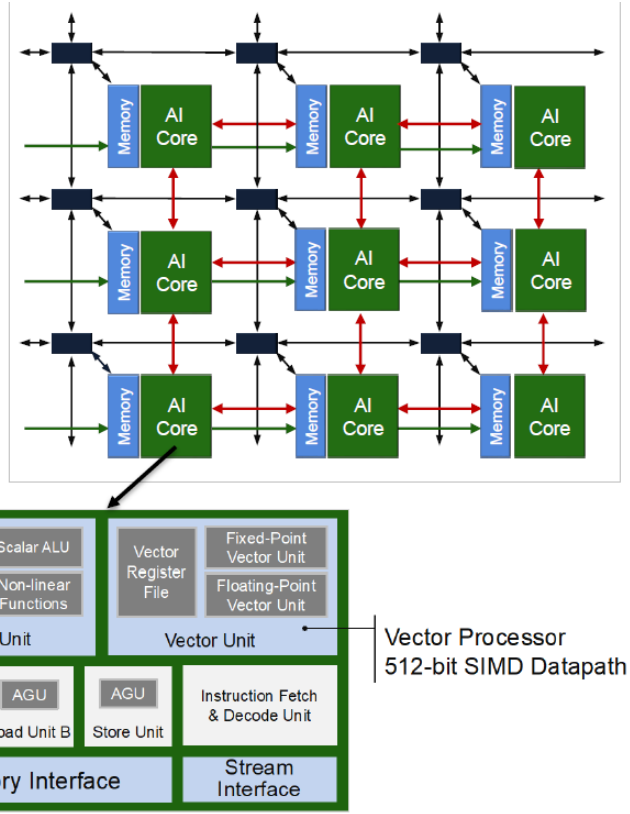
1.3 GHz VLIW / SIMD vector processors

Parallelity

- VLIW: 7+ operations / clock cycle
- SIMD: 512 bit vector datapath
  (8 / 16 / 32 bit & SPFP operands)
- Up to 128 INT8 MACs / clock cycle / core

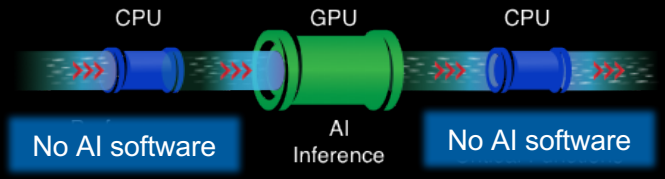
Memory

- 16 KB Internal program memory
- 32 KB data memory (parallel)
- Integrated DMA logic



CPU & GPU *vs* Versal-ACAP

GPU & CPU - Mismatched Throughput

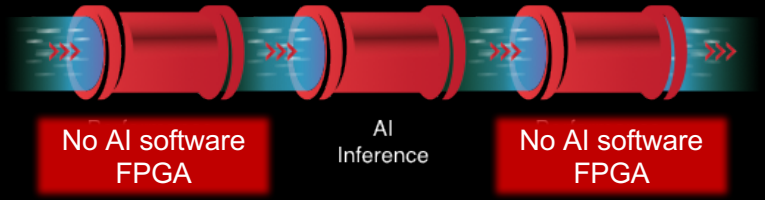


In heterogenous CPU / GPU systems the GPU, the other performance-critical functions of the application must still run in software (CPU)

Data copy from system memory to GPU global memory is still a limitation

High flexibility degree

Xilinx – Matched Throughput



In Versal devices the other performance-critical functions will be implemented in the FPGA, therefore, deep pipeline implementation on massive parallel operations

Data copy/moving by a dedicated high-bandwidth “Network on Chip” bus

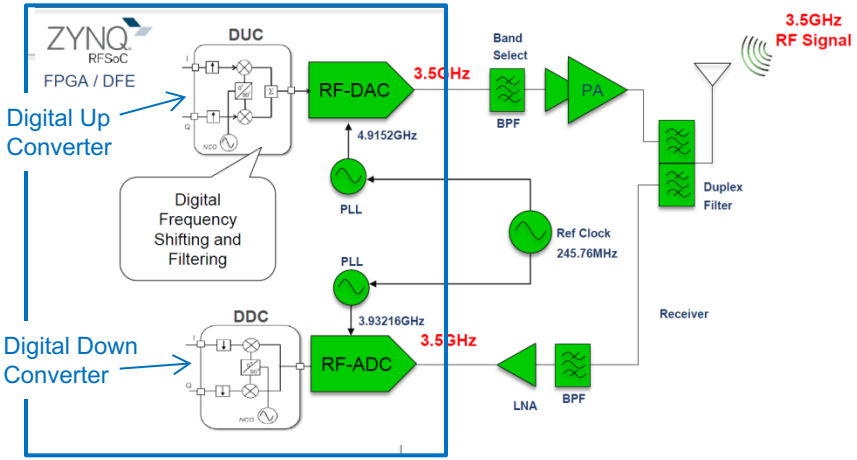
Less flexible, but …

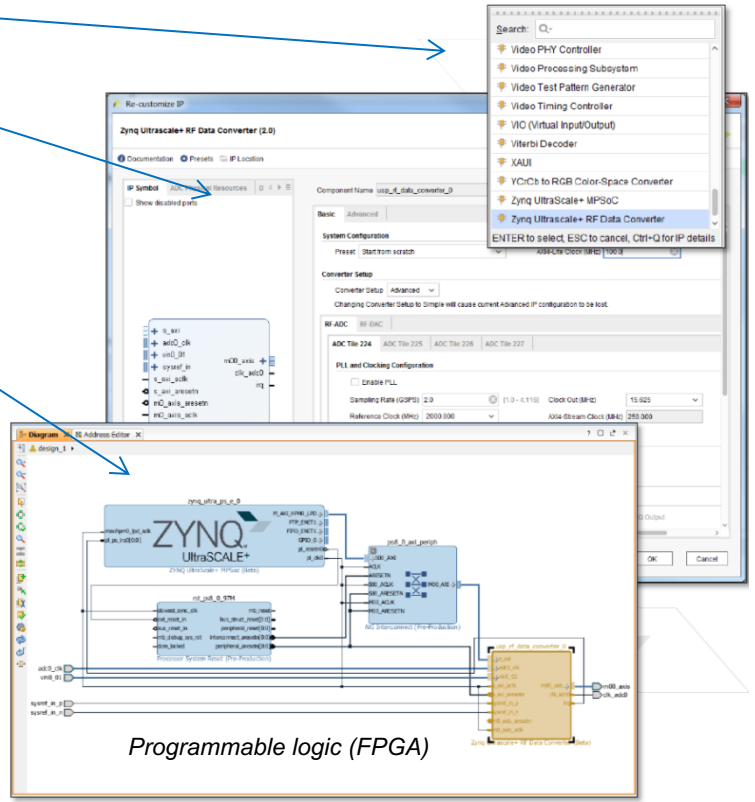
Vitis simplifies the use of deep-learning neural networks, *even for users without knowledge of deep-learning or FPGAs*.
The Vitis AI Library allows users *to focus more on the development* of their applications, *rather than the underlying hardware*.

RF-ADC/DAC Implementation steps

- Add RF-ADC/DAC instance using the IP-Core
- Use GUI to configure and customize the analog block
- Connect the ADC/DAC to ZYNQ by
  “Running_connection_automation”

Direct RF Sampling & digital Signal Processing

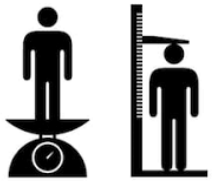




Programmable logic (FPGA)

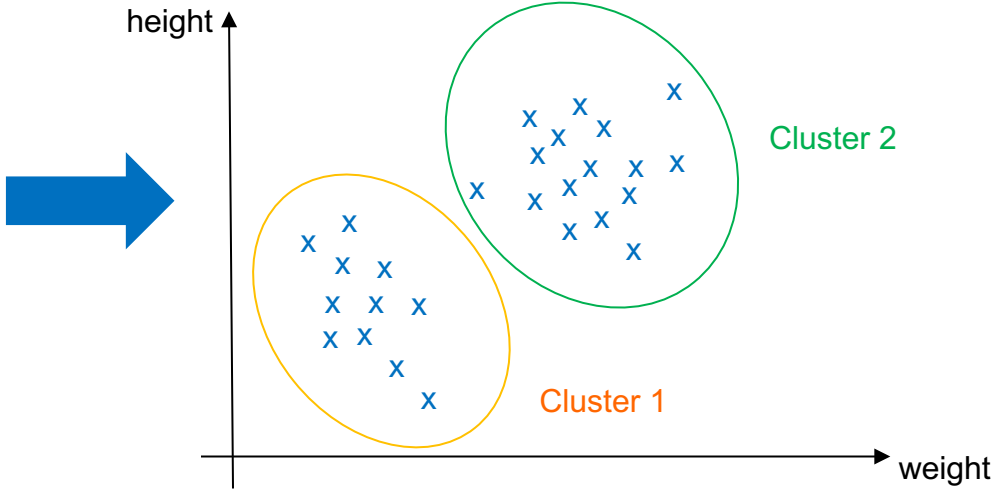
Unsupervised Learning

Unsupervised learning finds hidden patterns or intrinsic structures in data.
Clustering is the most common unsupervised learning technique. *It is used for exploratory*



Unlabelled data

x	y
170	67
150	51
….	…
190	87



In cluster analysis, data is partitioned into groups based on some measure of similarity or shared characteristic. Clusters are formed so that objects in the same cluster are very similar and objects in different clusters are very distinct.

Unsupervised Learning

k-Means

How to work: Partitions data into k number of mutually exclusive clusters

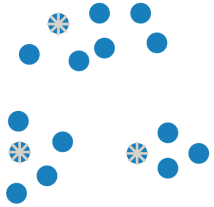
Result: Cluster centers

Best Used…

- When the number of clusters is known
- For fast clustering of large data sets

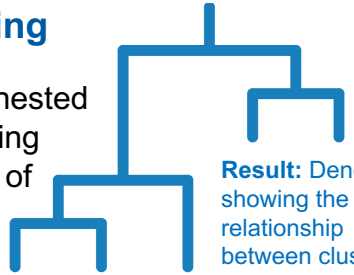
k-Medoids

How to work: Similar to k-means, but with the requirement that the cluster centers coincide with points in the data.



Result: Cluster centers that coincide with data points

Hierarchical Clustering



How to work: Produces nested sets of clusters by analyzing similarities between pairs of points

Result: Dendrogram showing the hierarchical relationship between clusters

Best Used...

When you don't know in advance how many clusters are in your data

Self-Organizing Map

How It Works: Neural-network based clustering that transforms a dataset into a topology-preserving 2D map.

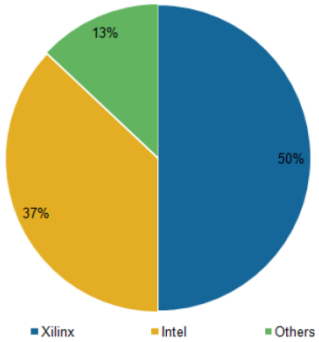
Result: Lower-dimensional (typically 2D) representation

Best Used...

To visualize high-dimensional data in 2D or 3D

FPGA market or which vendors did survive?

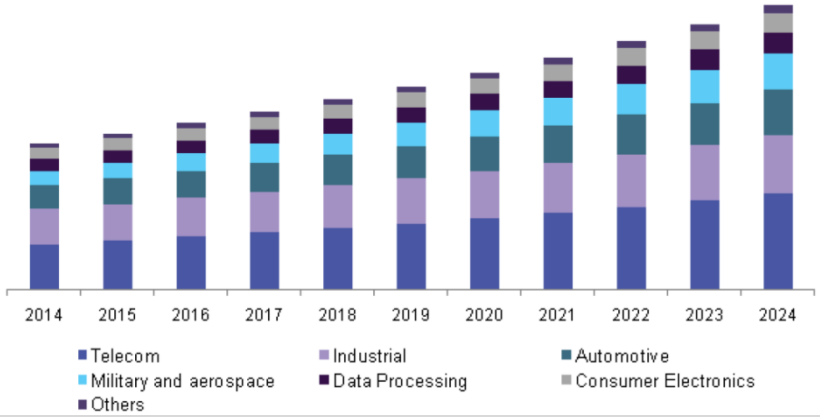
achronix SEMICONDUCTOR CORPORATION FPGA IP core for SoC designs	intel FPGA SRAM based FPGAs Broad range	LATTICE SEMICONDUCTOR Low power & cost efficient FPGAs
MICROCHIP 2 small SRAM FPGA families	Microsemi Flash & Antifuse FPGAs	XILINX ALL PROGRAMMABLE SRAM based FPGAs Broad range



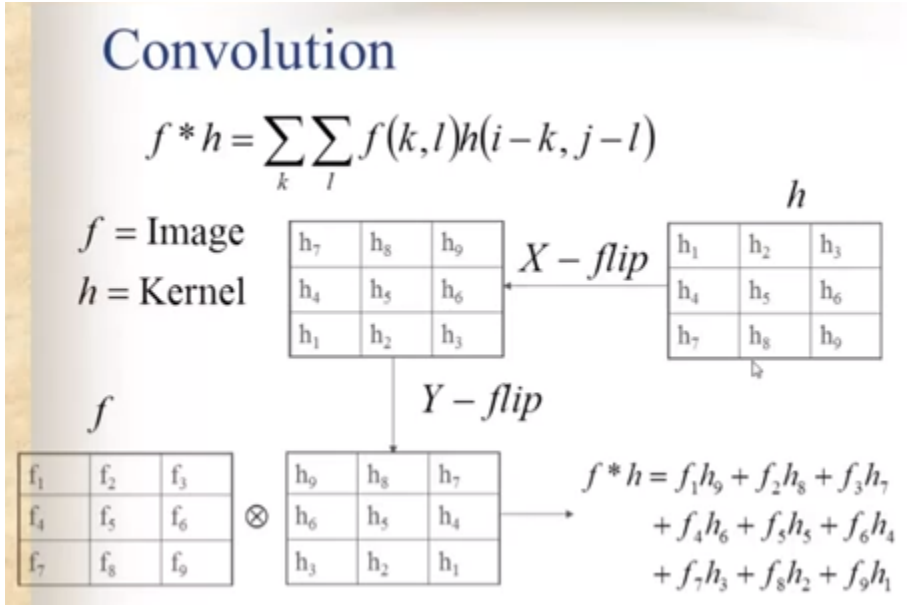
Xilinx: 50 %

Altera / Intel: 37 %

Others: 13 %



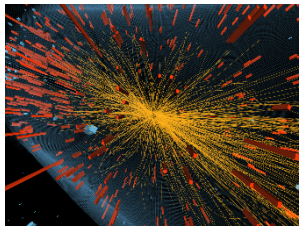
Convolution calculation



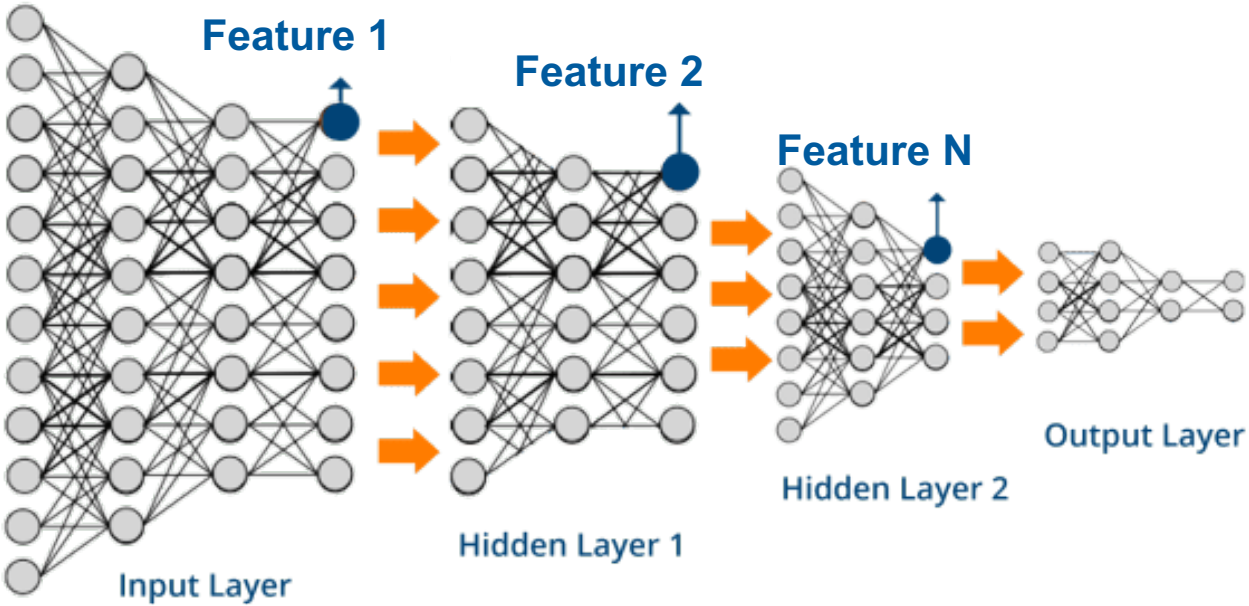
Deep Neural Network (Convolution layer)

Neural network for physics:

- **Supervised:** Neural network trained by Monte Carlo simulations
- **Unsupervised (no trained):** to find “New Physics” *without being explicitly programmed*

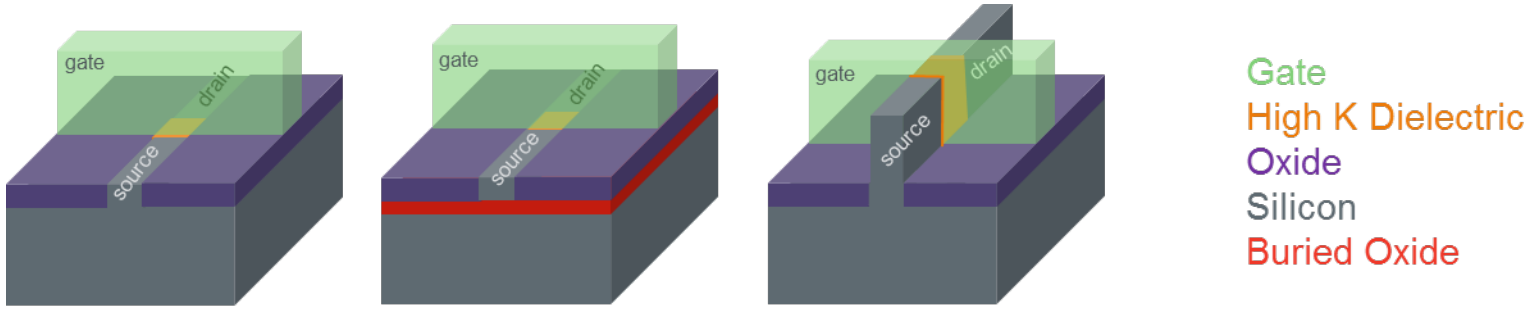


LH-CMS event



Perform leading data analyses. Physics event, Track reconstruction

CMOS technology



planar

> 28 nm

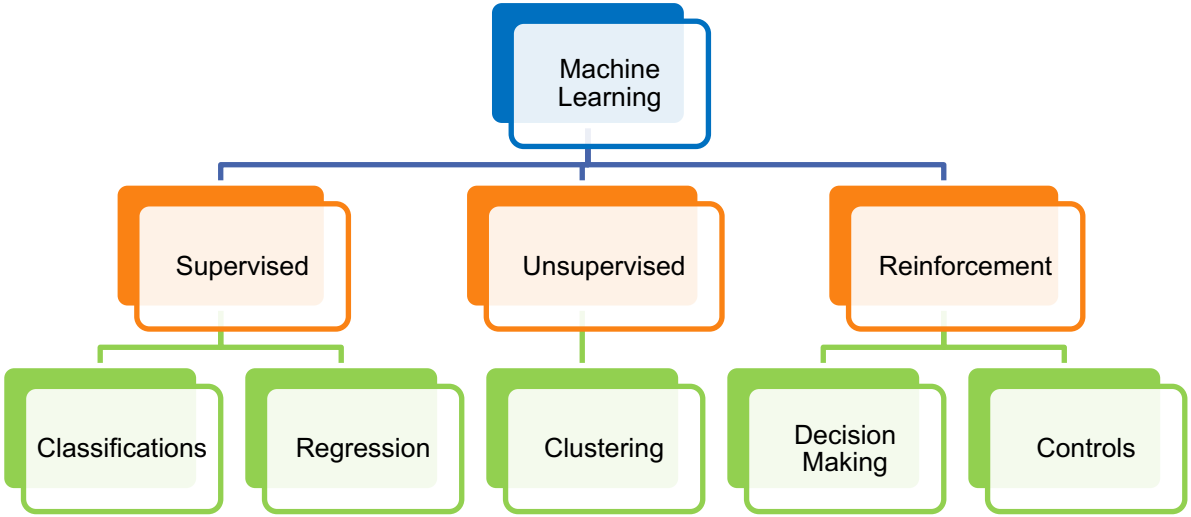
FD-SOI

12 nm, 22 and 28 nm

finFET

5 nm (by intel) 7 nm, 16 and 20 nm

Machine learning



Supervised: All data is *labelled* and the algorithms learn to predict the output from the input data.

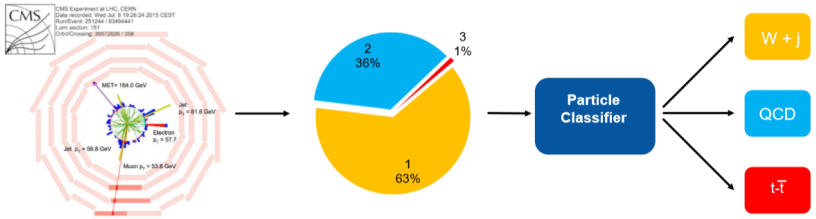
Unsupervised: All data is *unlabelled* and the algorithms learn to inherent structure from the input data

Reinforcement: The learning algorithm is trained not on present data but rather based on a *feedback* system.

- **Classification techniques** predict discrete responses, classify input data into categories. Example: whether an email is genuine or spam, or whether a tumor is cancerous or benign.
- **Regression techniques** predict continuous responses. Example: changes in temperature or fluctuations in power demand.
- **Clustering** is employed for exploratory data analysis to find hidden patterns or groupings in data.
- **Reinforcement** is employed for fast feedback control systems (i.e. beam stability in accelerators).

From data to understanding …

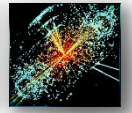
The generation of detectors will face a peak luminosity of 30 x 10^{34} cm^{-2}/s with a pile-up 1000 and radiation levels that are 1-2 orders of magnitude larger than those at the HL-LHC. Timing precision of 5 – 10 ps is required for pile-up mitigation. *The resulting data rates lie in the hundreds of TB/s.*



Data Analysis

- Traditional data analysis techniques in HEP use a sequence of Boolean decisions followed by statistical analysis on the selected data.

The last big success of the traditional science is the Higgs boson



- Find “a new physics” without being explicitly written in the analysis code.

Mathematical models learnt from data

From Detectors and DAQ systems

- The data deluge MUST BE readout and real-time processed without any data/information loss
- The radiation load on the detectors → will dramatically impact on the noise/efficiency of each single sensitive cell

Fast track-reconstruction and triggering

Data Quality Monitoring

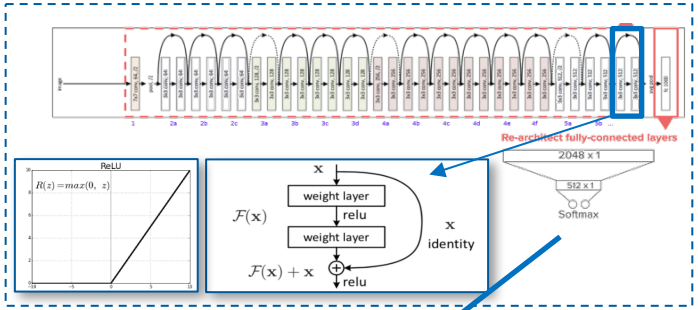
Autonomous detector configuration

Sophisticated slow-controls

On-line self-calibration

Examples: Objects recognition/classification

- Neural network implemented resnet50

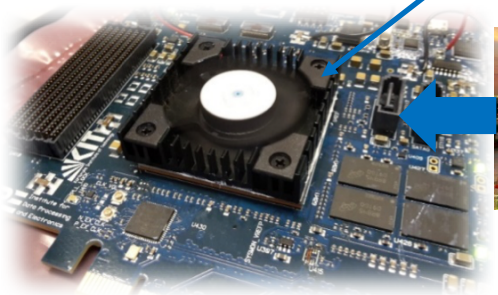


Linux + drivers + Resnet 50 (DNN) → on-FPGA

```
root@xilinx-zcu102-2019_2:~/vitis_ai_samples_zcu102# ls
adas_detection  images          inception_v1_mt_py  resnet50        segmentation
common          images_backup   pose_detection      resnet50_mt_py  video_analysis
root@xilinx-zcu102-2019_2:~/vitis_ai_samples_zcu102# cd resnet50
root@xilinx-zcu102-2019_2:~/vitis_ai_samples_zcu102/resnet50# ls
Makefile  build  dpuv2_rundir  model  picture.jpg  resnet50  src  words.txt
root@xilinx-zcu102-2019_2:~/vitis_ai_samples_zcu102/resnet50# ./resnet50 dpuv2_rundir/

Image : xian.jpg
top[0] prob = 0.416849  name = palace
top[1] prob = 0.119429  name = seashore, coast, seacoast, sea-coast
top[2] prob = 0.119429  name = fountain
top[3] prob = 0.043936  name = dome
top[4] prob = 0.043936  name = church, church building
```

Courtesy: Weijia Wang



High-Flex 2 multi-purpose PCIe card

xian.jpg

```
Image : burger.jpg
top[0] prob = 0.794760  name = cheeseburger
top[1] prob = 0.138109  name = plate
top[2] prob = 0.014557  name = corn
top[3] prob = 0.008829  name = tray
top[4] prob = 0.006876  name = hotdog, hot dog, red hot
```

burger.jpg

car.jpg

```
Image : car.jpg
top[0] prob = 0.929687  name = cab, hack, taxi, taxicab
top[1] prob = 0.021864  name = minivan
top[2] prob = 0.010328  name = car mirror
top[3] prob = 0.008043  name = motor scooter, scooter
top[4] prob = 0.003799  name = police van, police wagon, paddy
root@xilinx-zcu102-2019_2:~/vitis_ai_samples_zcu102/resnet50#
```