AI acceleration – image processing as a use case

Masoud Daneshtalab
Associate Professor at MDH
www.idt.mdh.se/~md/

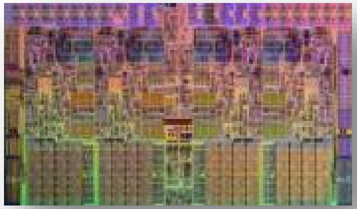
MÄLARDALEN UNIVERSITY SWEDEN

Outline

- Dark Silicon
- Heterogeneouse Computing
- Approximation
  - Deep Neural Networks

The glory of Moore's law

Intel 4004
2300 transistors
740 kHz clock
10um process
10.8 usec/inst



Intel Core i7 980X
1.17B transistors
3.33 GHz clock
32nm process
73.4 psec/inst

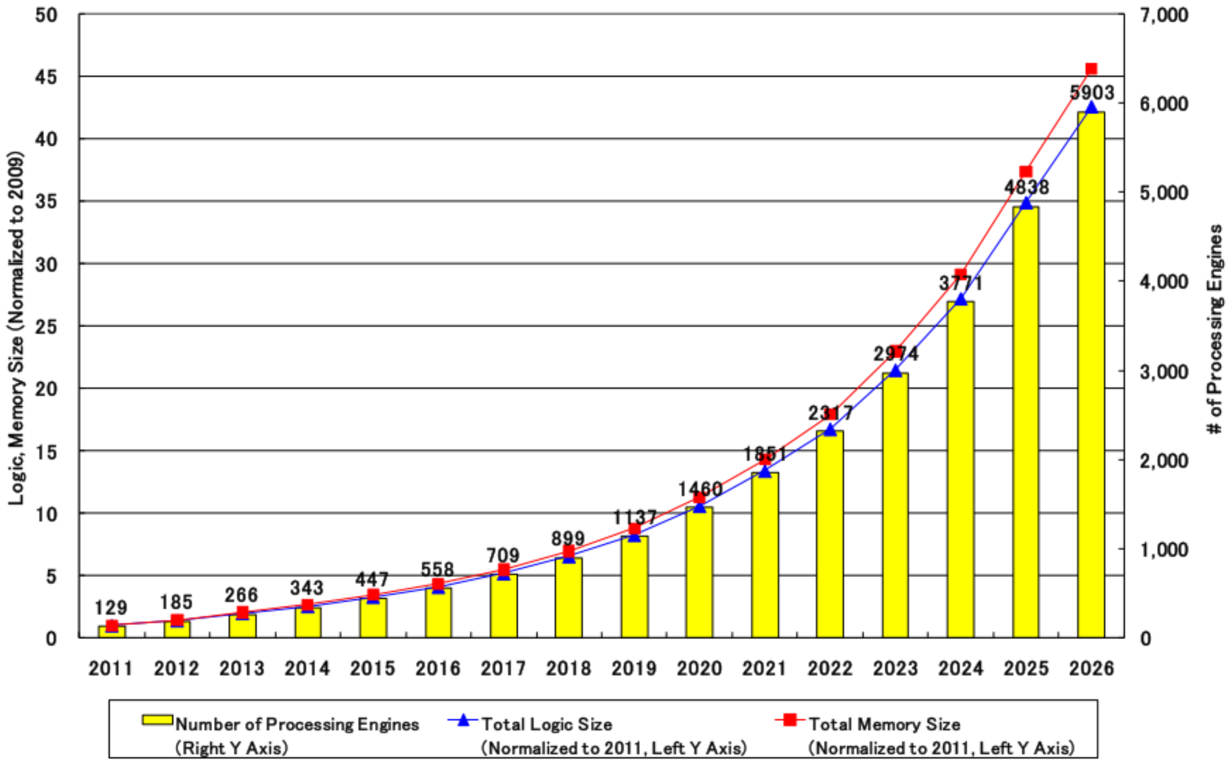
Every 2 Years

- Double the number of transistors
- Build higher performance general-purpose processors
  - Make the transistors available to masses
  - Increase performance (1.8×↑)
  - Lower the cost of computing (1.8×↓)

Semiconductor trends

ITRS roadmap for SoC Design Complexity Trens

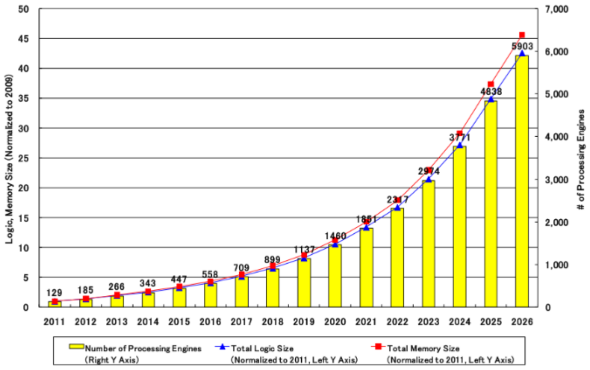
ITRS: International Technology Roadmap for Semiconductors

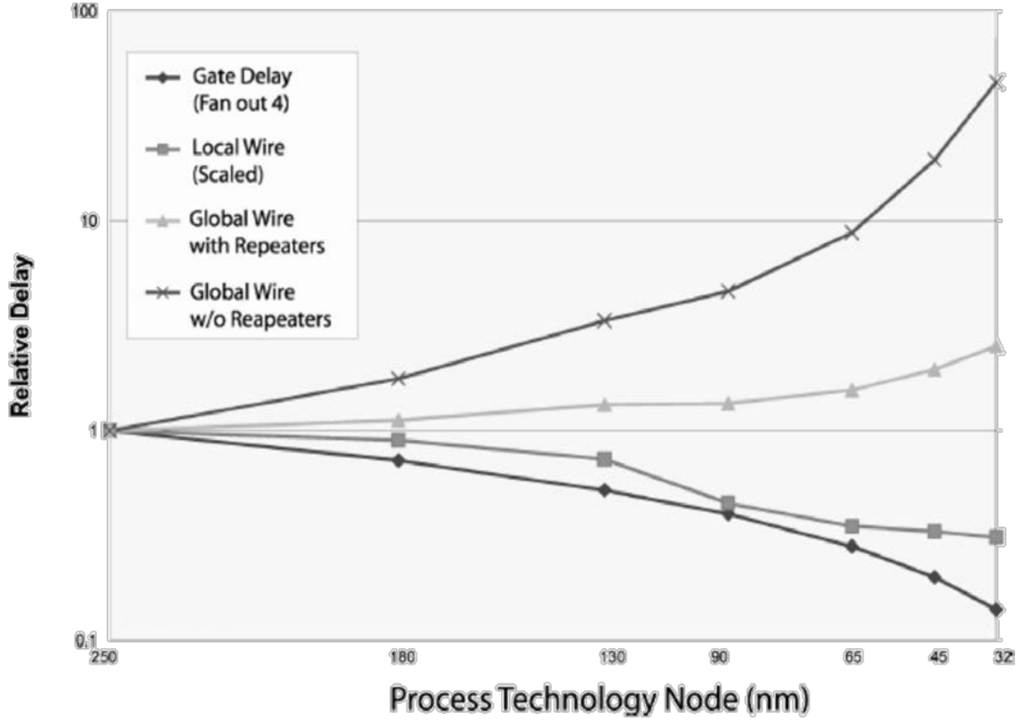


Expected number of processing elements into a System-on-Chip (SoC).

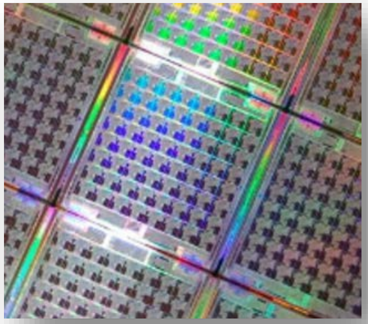
Semiconductor trends

<u>ITRS roadmap for SoC Design Complexity Trens</u>

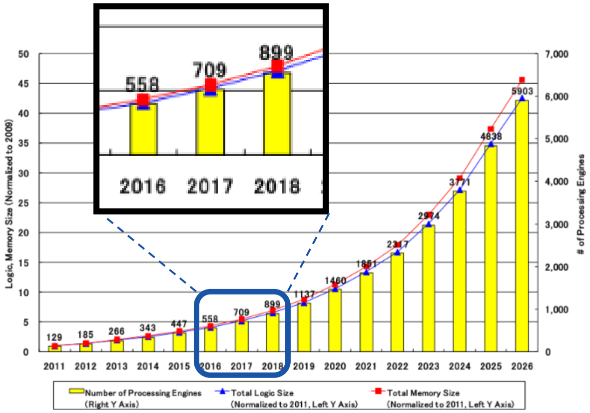


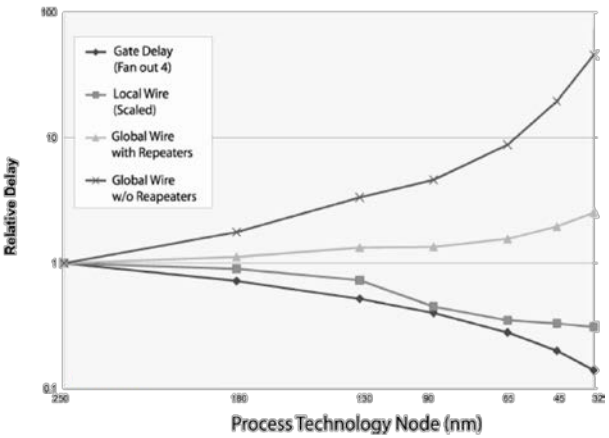


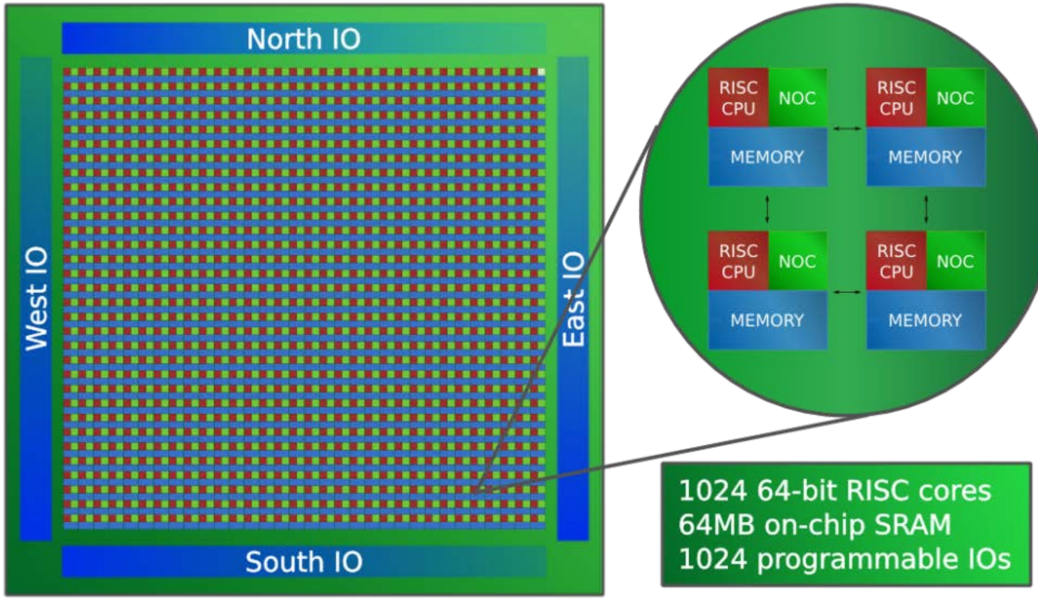
Semiconductor trends



Network-on-Chip (NoC) –based Multi/Many-core Systems







"**Adapteva**, Inc." http://www.adapteva.com/

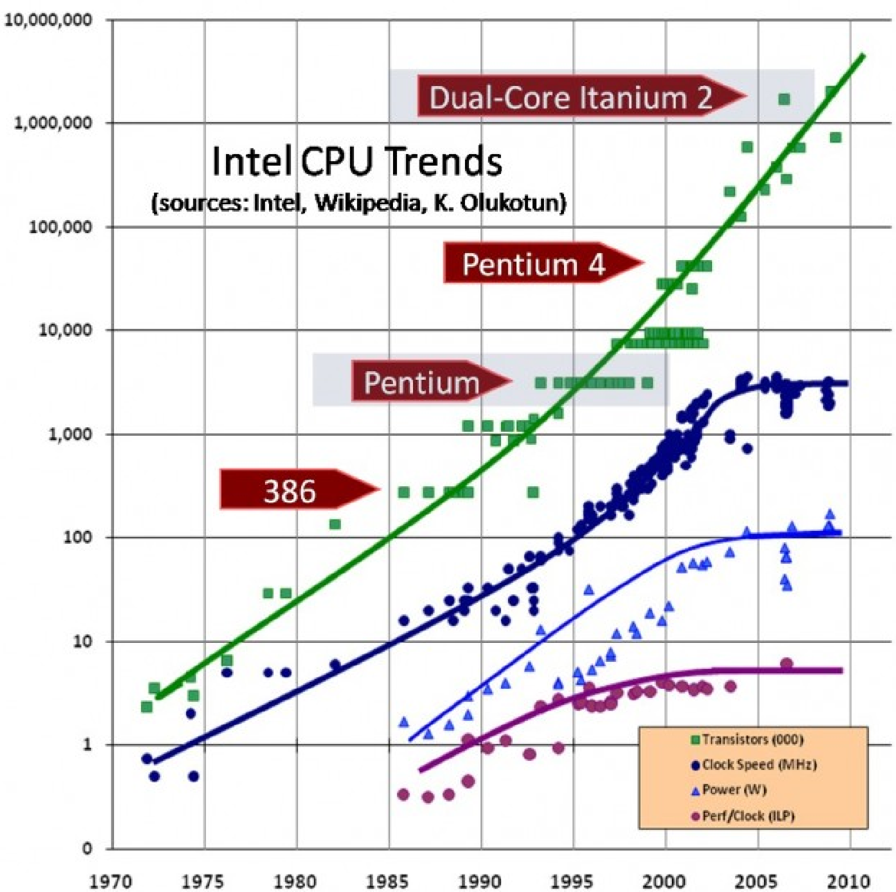
"**Arteris**, Inc." http://www.arteris.com/

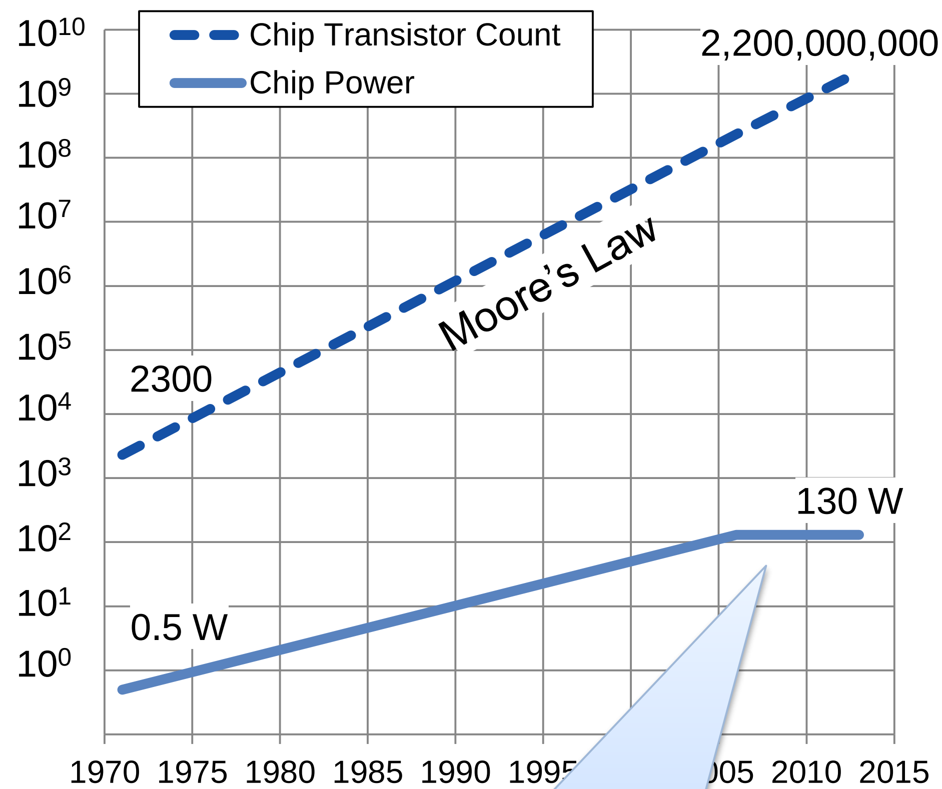
"**Sonics**, Inc." http://sonicsinc.com/

Founder: Andreas Olofsson
Sponsored by Ericsson AB

Dark Silicon Era

The catch is powering exponentially increasing number of transistors without melting the chip down.

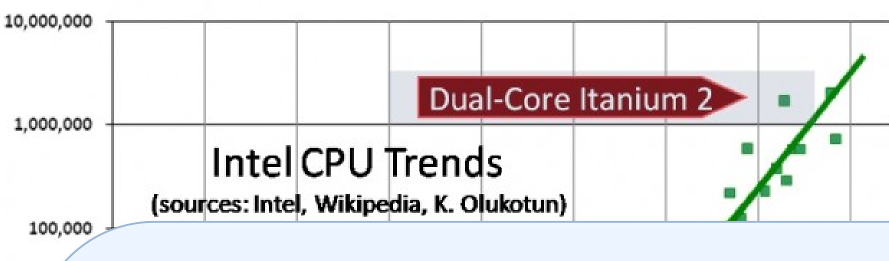


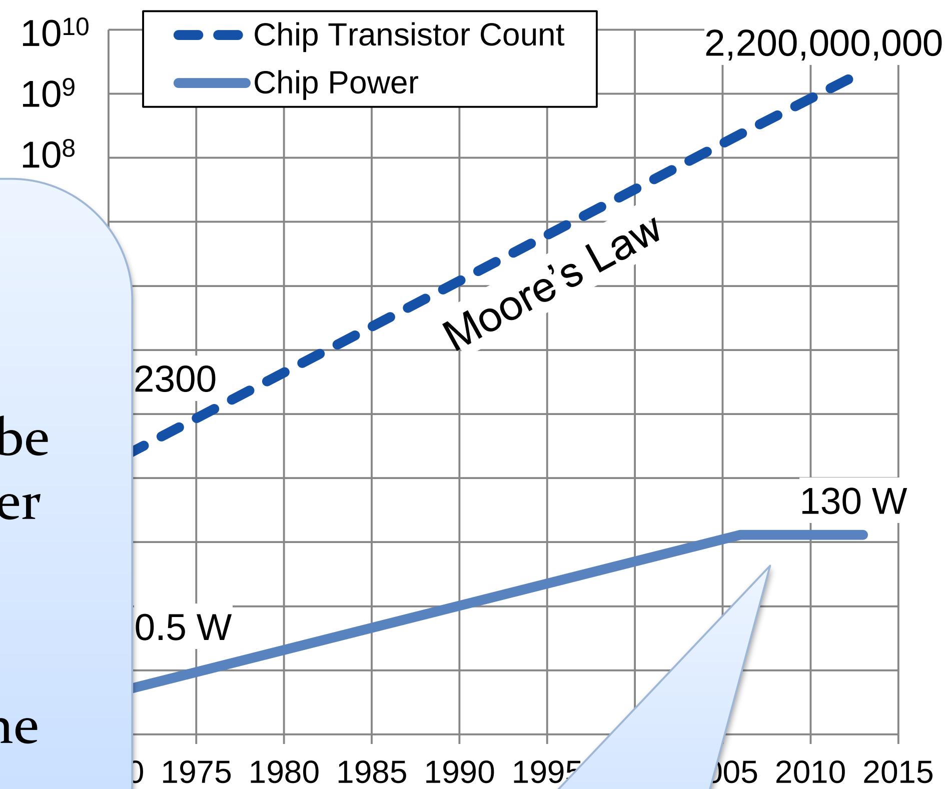


If you cannot power them, why bother making them?

Dark Silicon Era

The catch is powering exponentially increasing number of transistors without melting the chip down.





Dark Silicon

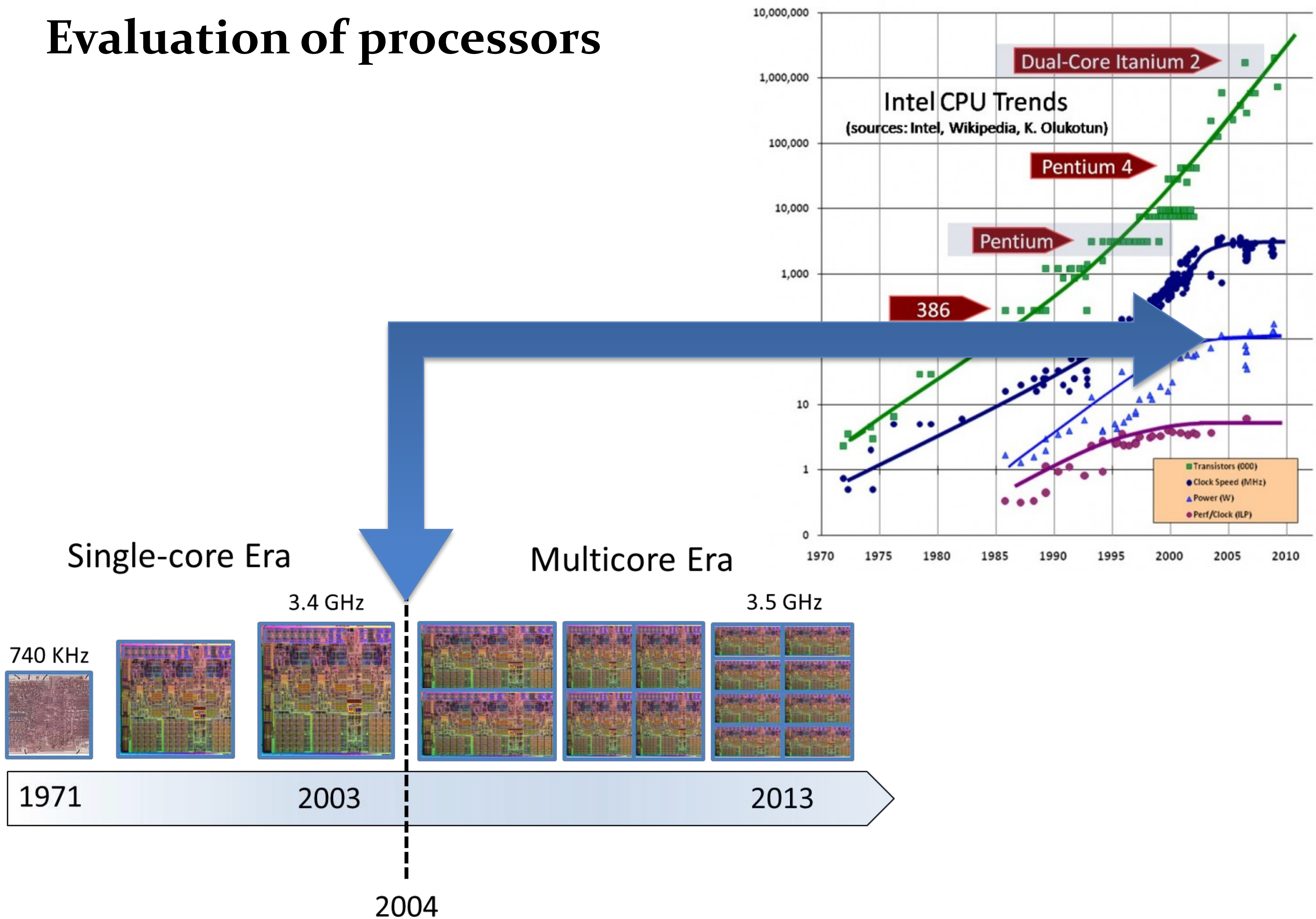
(**<u>Utilization Wall</u>**)

Fraction of transistors that need to be powered off at all times due to power constraints.

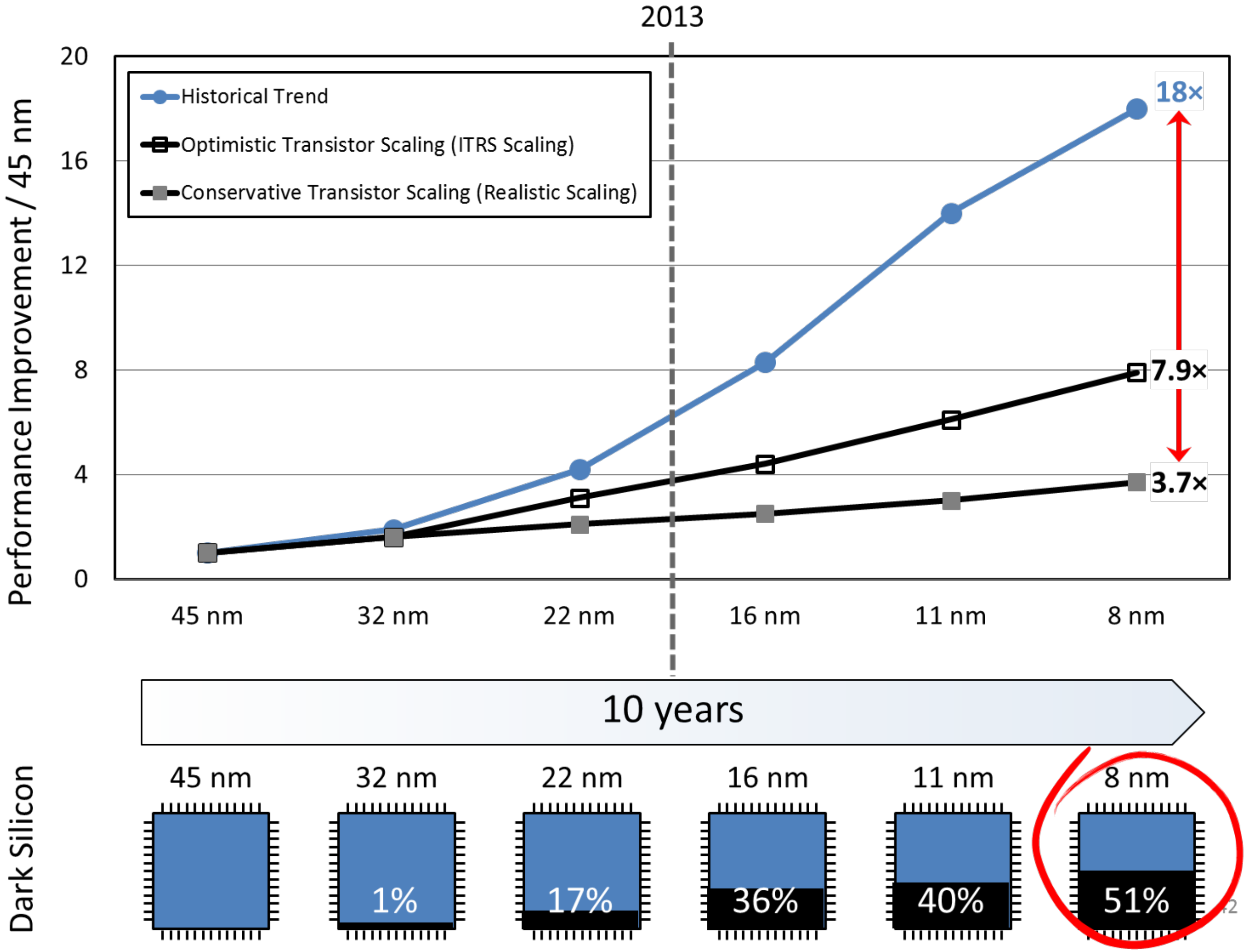
Lower utilization, Lower performane

If you cannot power them, why bother making them?

Evaluation of processors



Even multicores could not help!

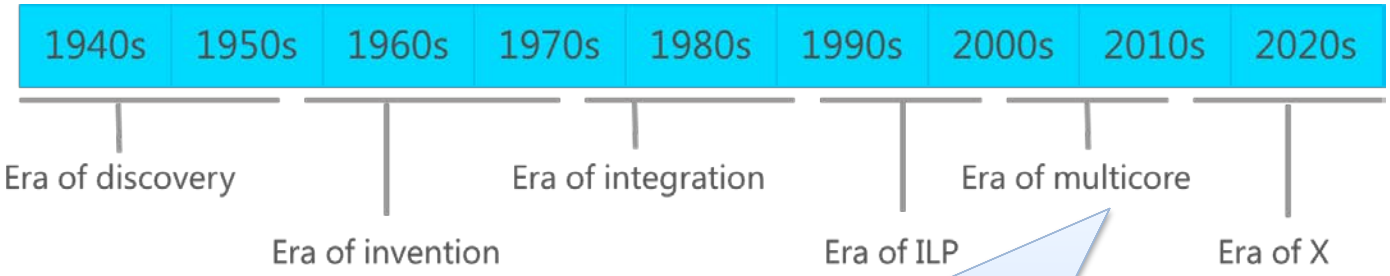


[Esmaeilzadeh, Blem, St. Amant, Sankaralingam, Burger, ISCA 2011]

End to Moore's law?

High Volume Manufacturing	2008	2010	2012	2014	2016	2018	2020	2022
Technology Node (nm)	45	32	22	16	11	8	6	4
Integration Capacity (BT)	8	16	32	64	128	256	512	1024

Source: Shekhar Borkar, Intel Corporation



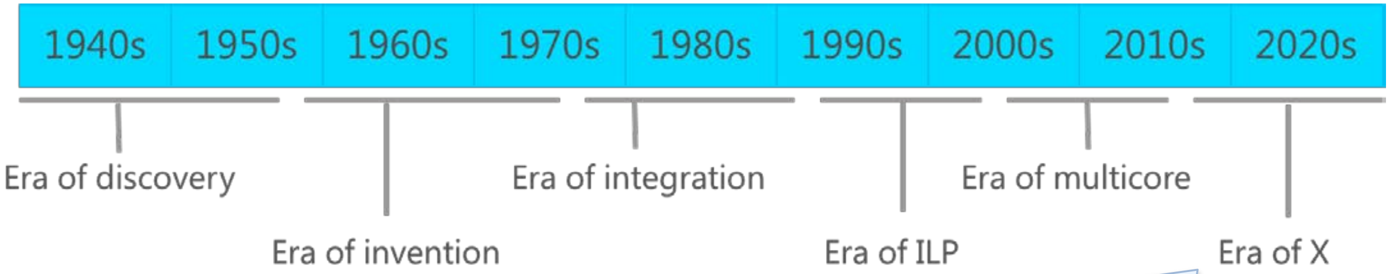
Multicores are likely to be a stopgap

- Not likely to continue the historical trends
- Do not overcome the transistor scaling trends
- The performance gap is significantly large

End to Moore's law?

High Volume Manufacturing	2008	2010	2012	2014	2016	2018	2020	2022
Technology Node (nm)	45	32	22	16	11	8	6	4
Integration Capacity (BT)	8	16	32	64	128	256	512	1024

Source: Shekhar Borkar, Intel Corporation

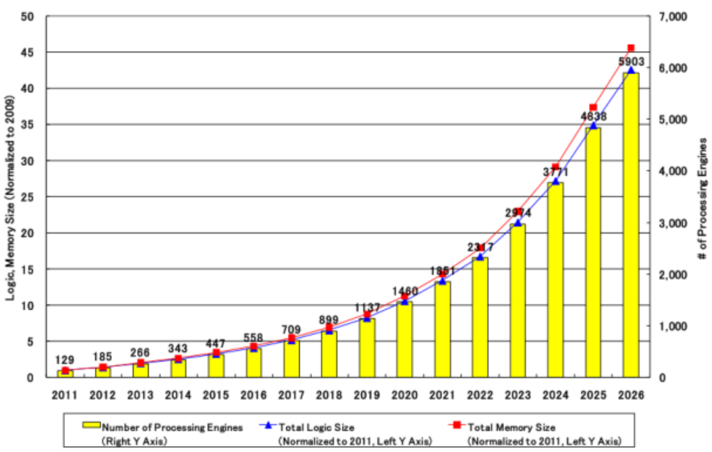


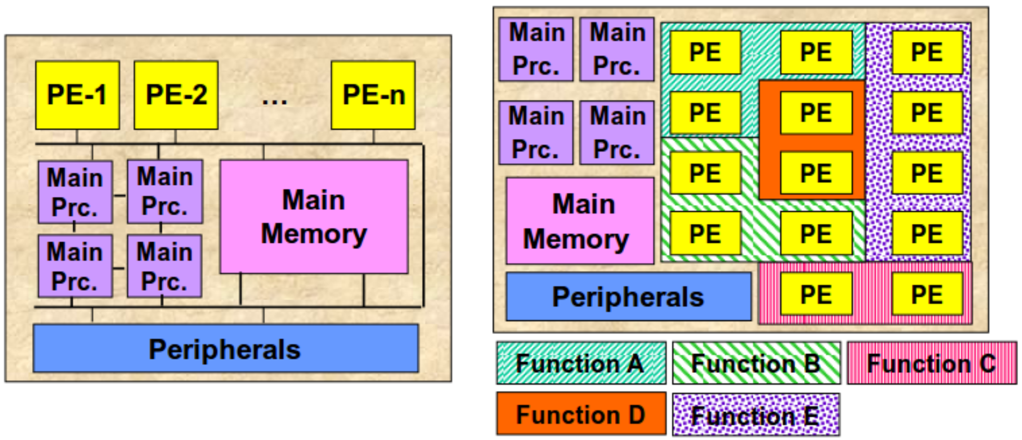
- **HW/SW specialization and heterogeneity**
- **Approximate computing**
- New emerging technologies (under development)

Where we are now and what's the trend?

ITRS roadmap for SoC Portable Design Complexity Trens

ITRS: International Technology Roadmap for Semiconductors





SoC Architecture Template.

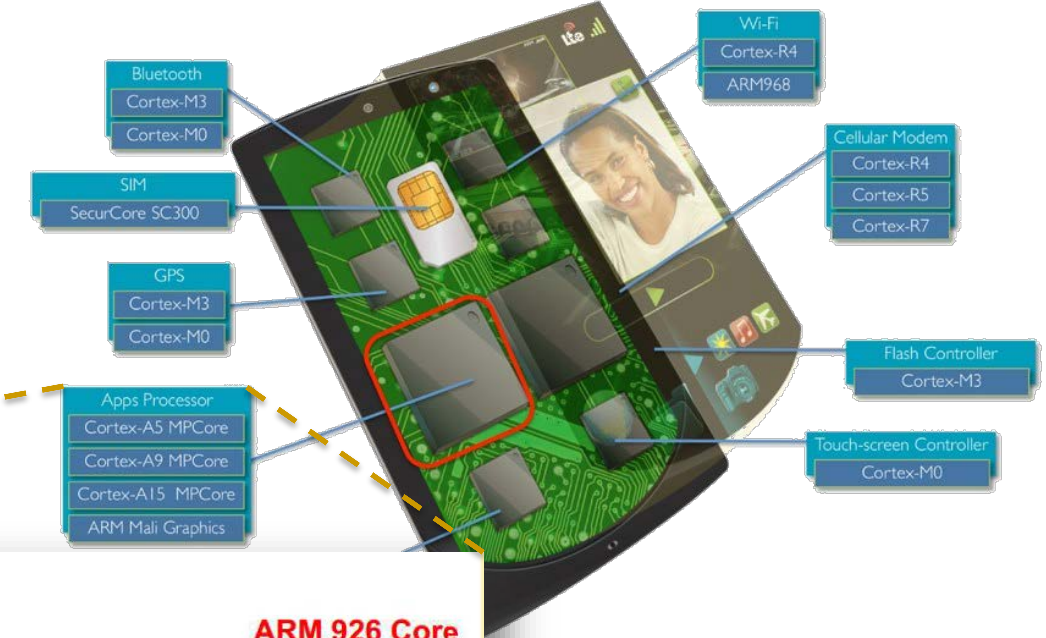
The SOC embodies a highly parallel architecture consisting of

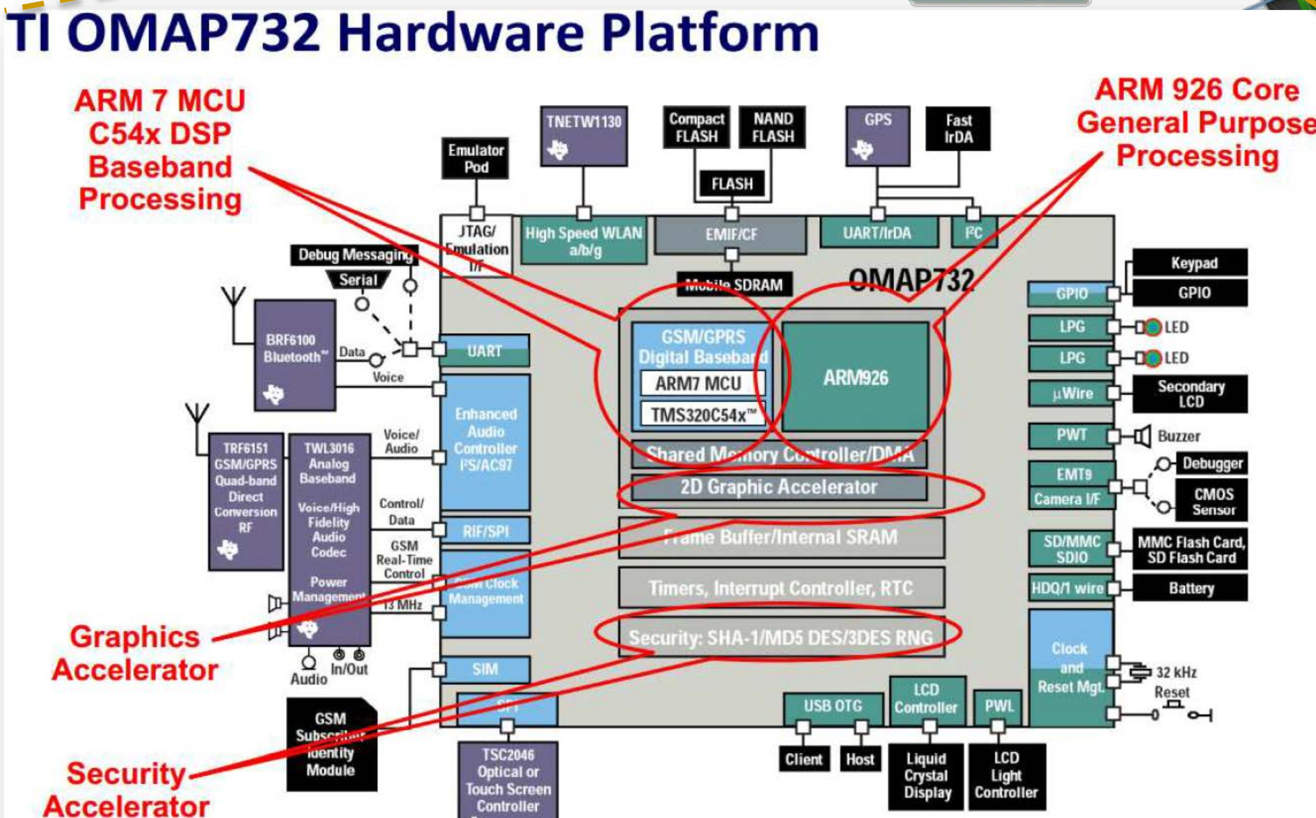
- **Main processors (grow slowly)**
- **PEs (Processing Engines) (grow faster)**
  - **customized processor**
  - **Accelerators (function).**
- **Peripherals**
- **Memories (proportional to #PEs)**
- **Die size of 49mm^2 gradually decreases to 44mm^2**

This architecture template enables both **high processing performance** and **low power consumption** by virtue of <u>parallel processing and hardware realization of specific functions</u>.

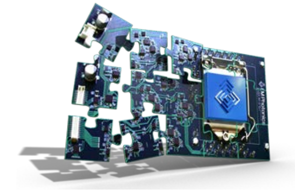
Example:

OMAP (Open Multimedia Applications Platform)

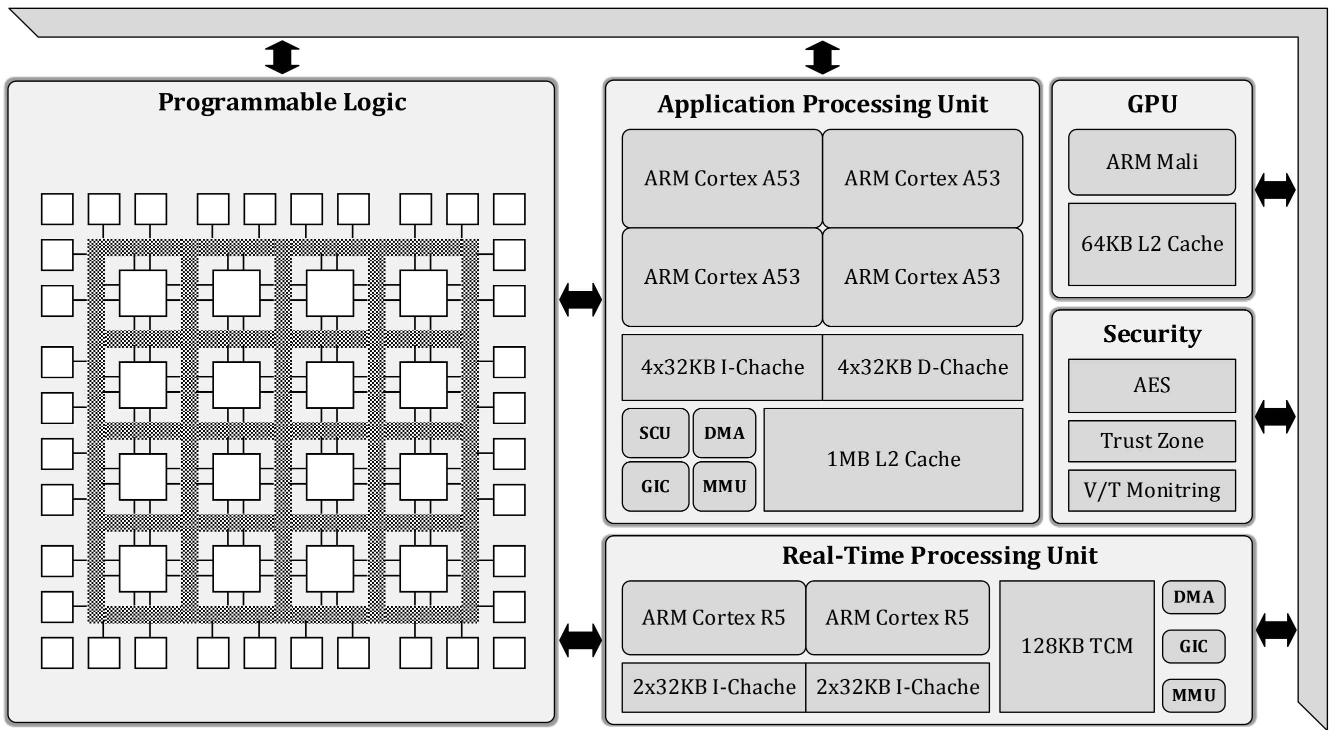




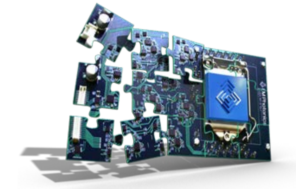
Paradigm Shift from Homogeneity to Heterogeneity



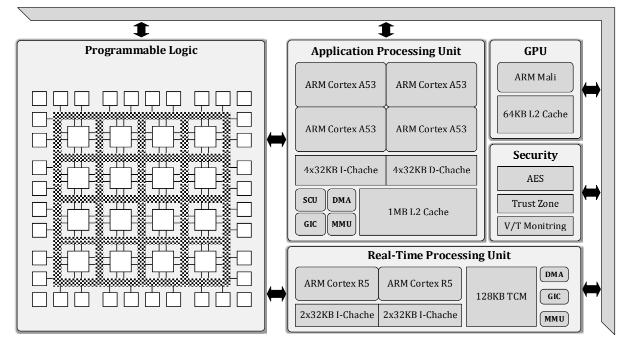
Heterogenous (Zynq/HSA-like) HW/SW SoC platform

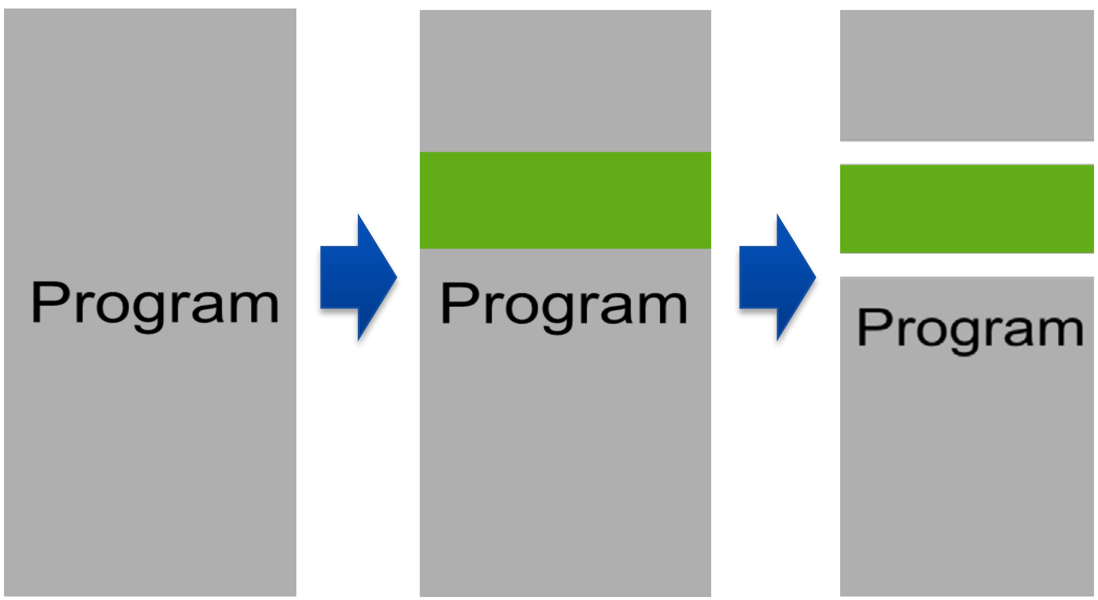


Heterogeneous Computing



Transforming von Neumann to heterogeneity





Find critical part of program component

Compile the program & specilized SW/HW

Execute on a fast specialized processing engine (PE)

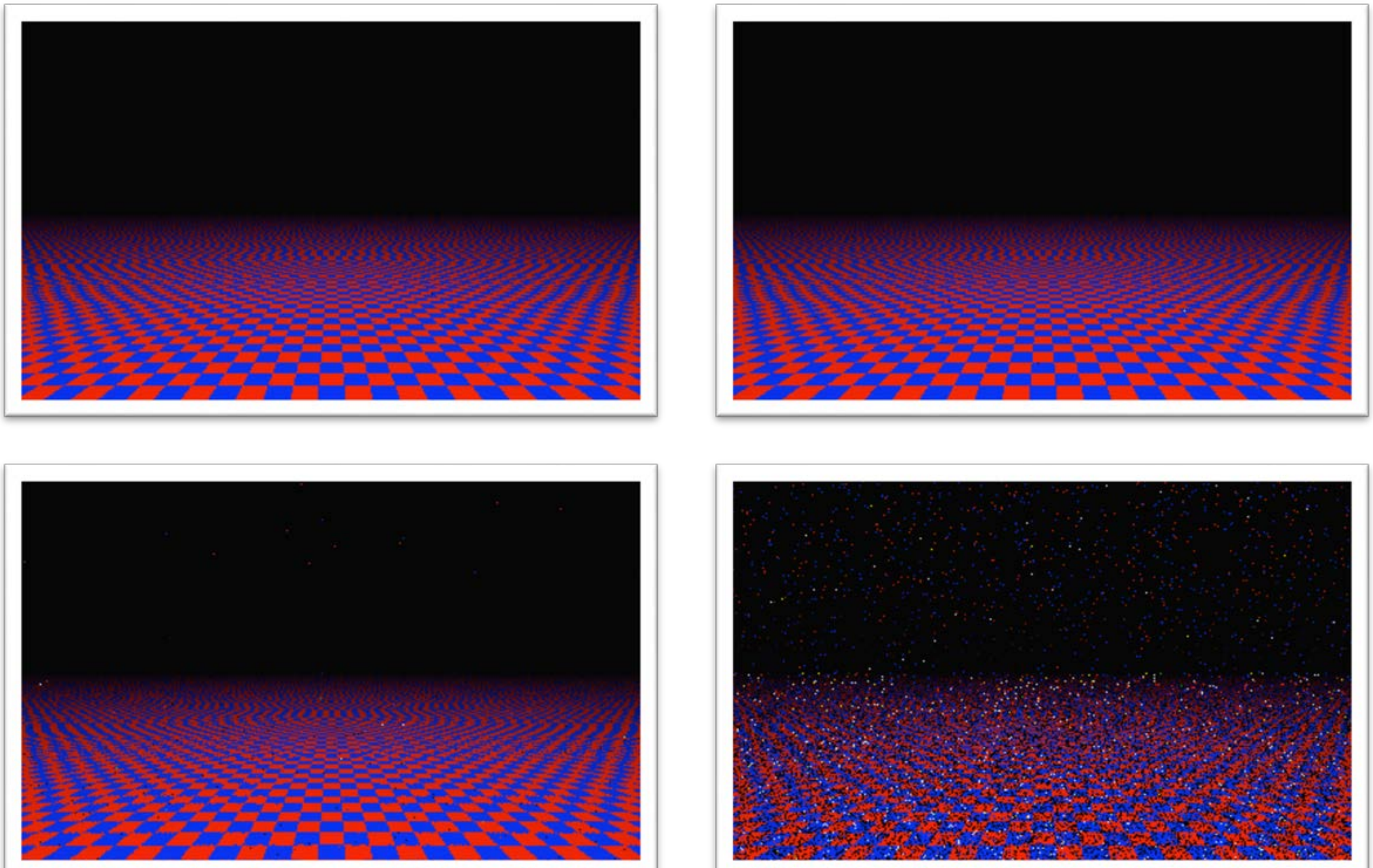
Approximate computing

- Relax the abstraction of near-perfect accuracy in general-purpose computing
- **Allow errors to happen in the computation**
    - **Run faster**
    - **Power efficient**

- This sounds a bit crazy but a large body of important applications having some amount of errors in the computation, entirely acceptable!
    - Computer vision, multimedia, stream applications
    - Large-scale machine learning
    - Bioinformatics
    - Mining big data
    - Speech and AI

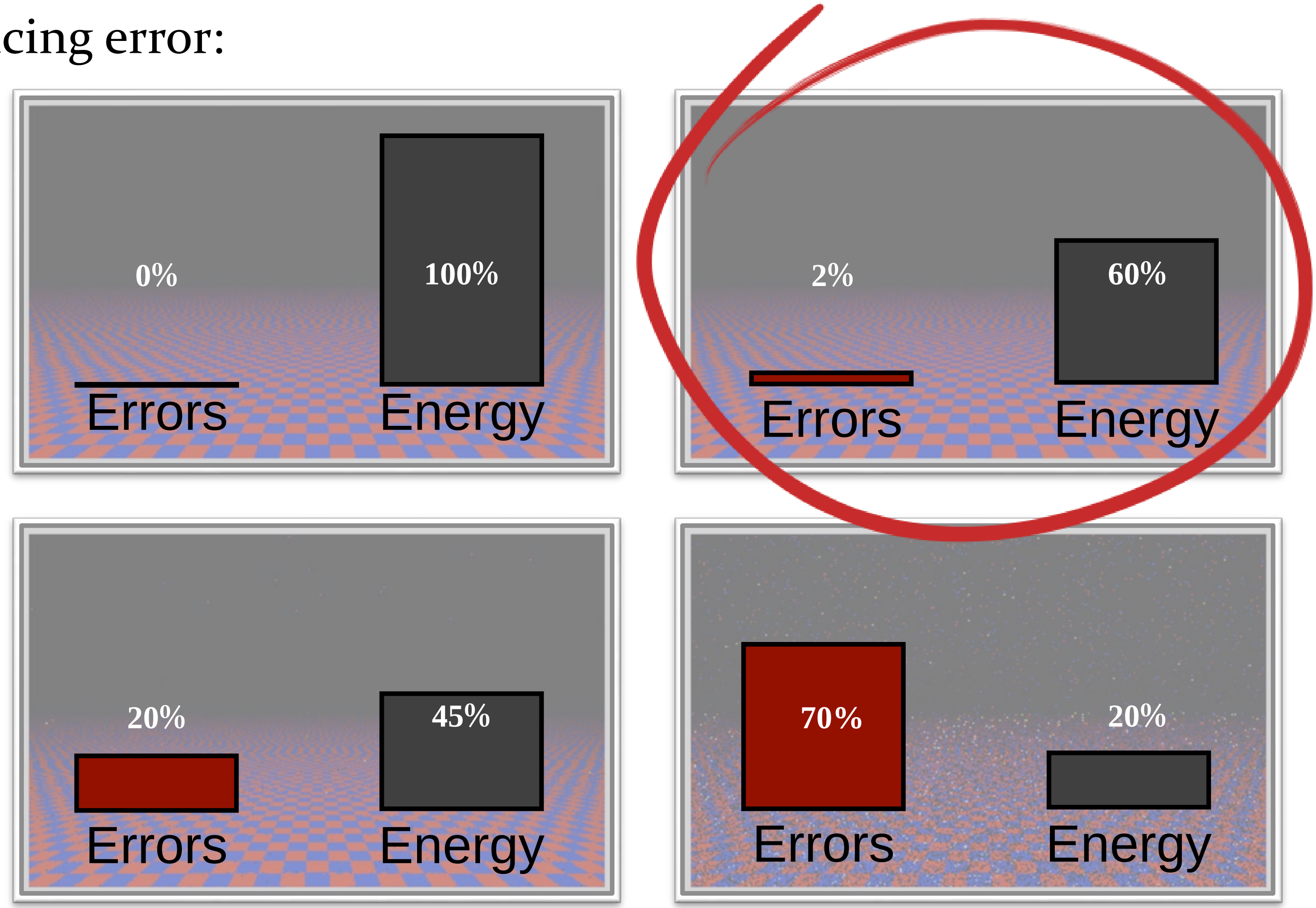
Approximate computing (cont.)

Embracing error:

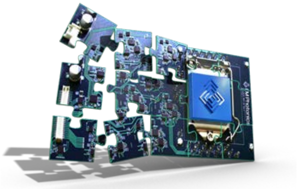


Approximate computing (cont.)

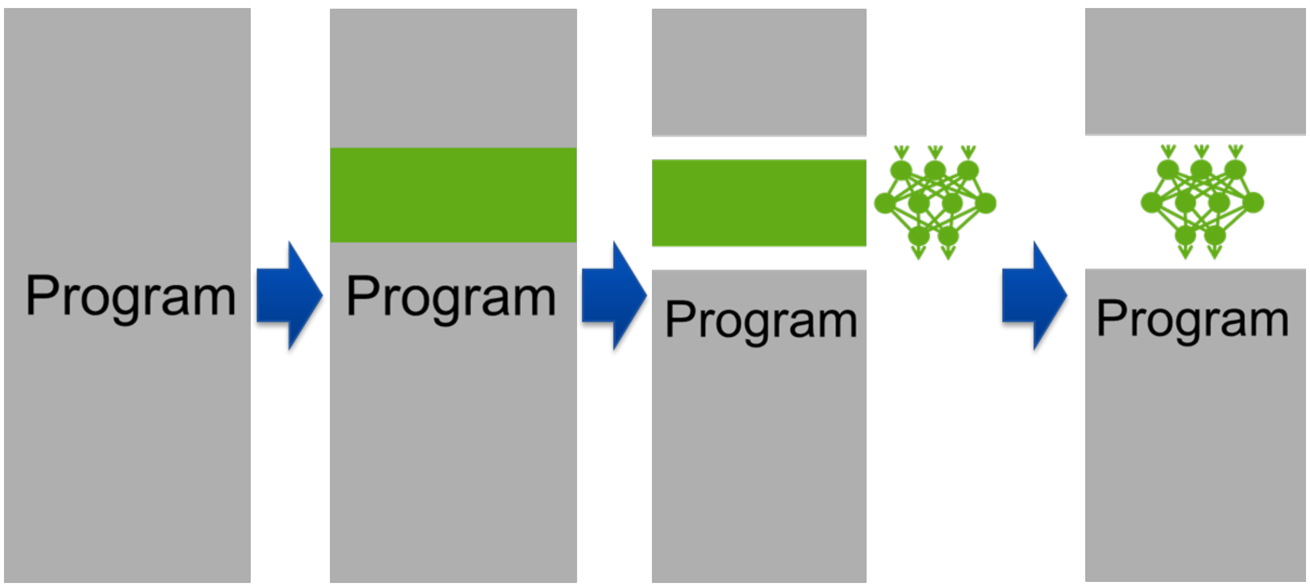
Embracing error:



Approximate computing (cont.)



Transforming von Neumann to Neural Networks



Find an approximate program component

Compile the program & train a NN

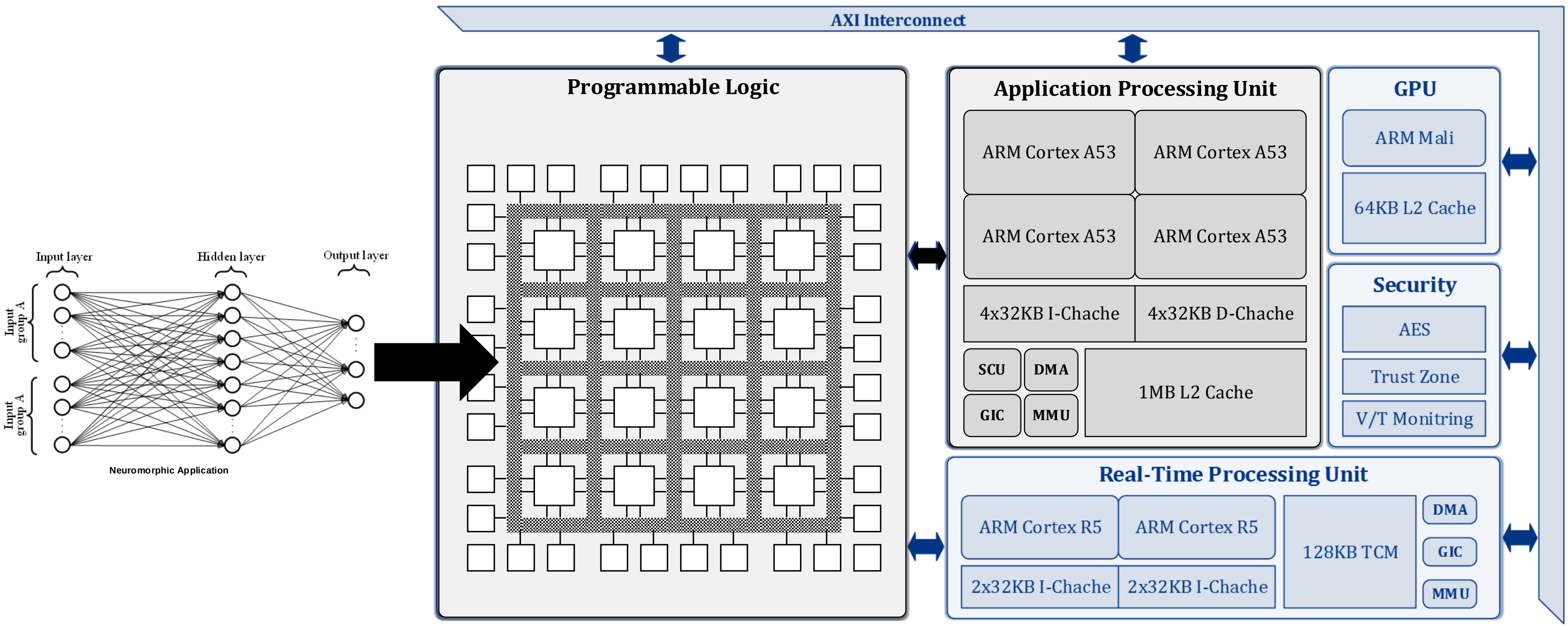
Execute on a fast Neural Processing Engine(NPE)

Speed: ~4×↑,
Energy: ~10×↓,
Quality: 5%↓)

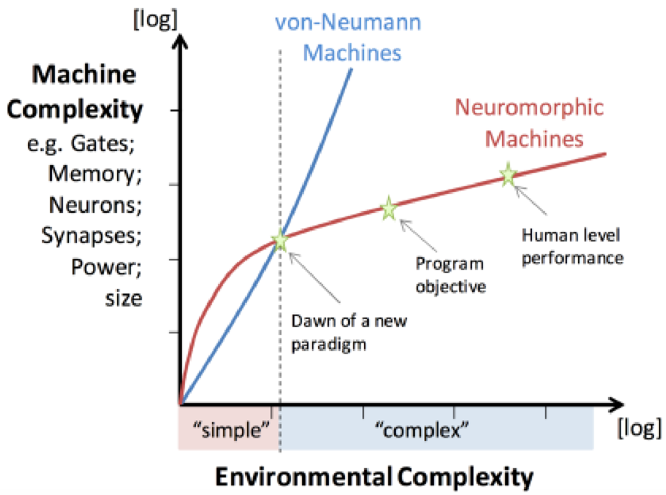
[Esmaeilzadeh, and Burger, MICRO 2012]

Approximate computing (cont.)

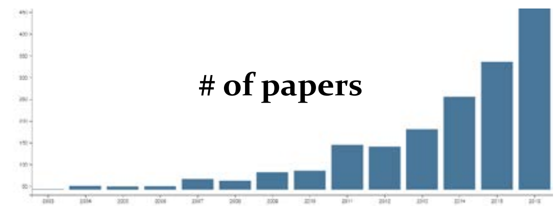
Zynq/HSA-like HW/SW SoC platform

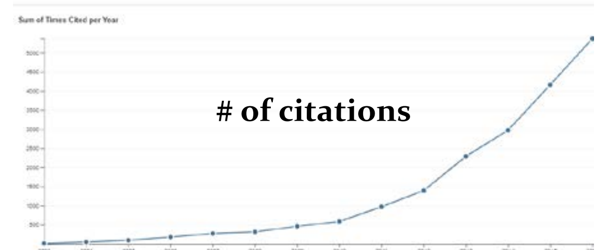


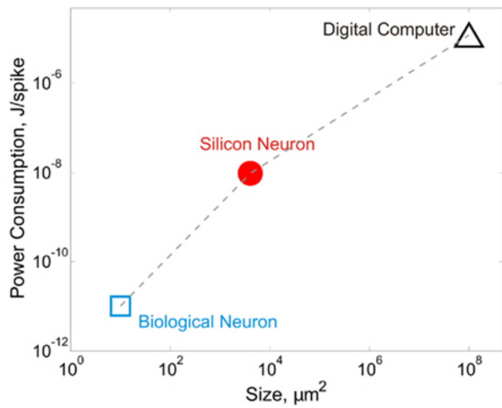
Why Deep Learning (Neuromorphic)?



Neuromorphic computing scales well increasing complexity of problems.



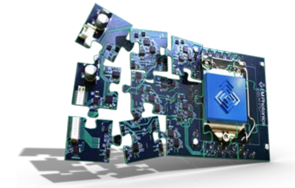




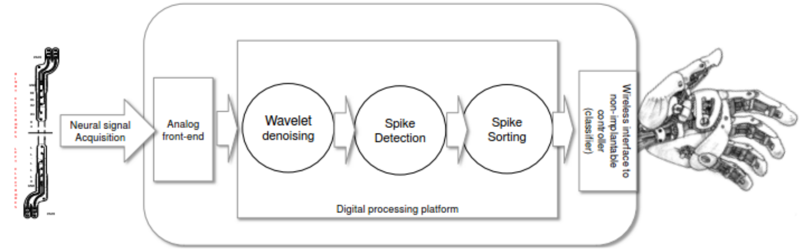
Biological and silicon neurons have much better power and space efficiencies than digital computers [**MIT & Intel**]

Intel: Nervana and Movidius
Google: Tensor
Nvidia: Jetson TX1 and TX2 specialized for DNN
Microsoft: BrainWave
Qualcomm: Zeroth Processors, extending with NVM
IBM: TrueNorth
Auviz/Xilinx: CNN accelarator

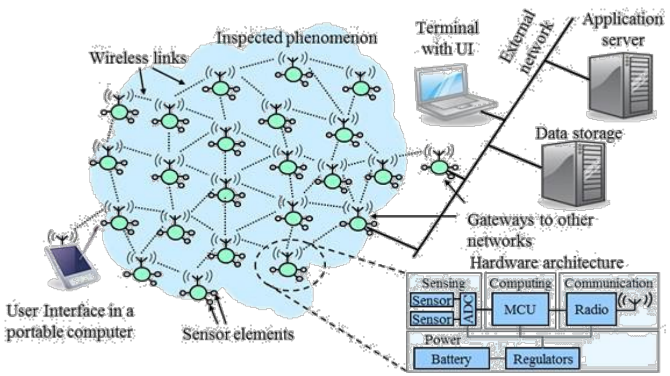
Computing Platform

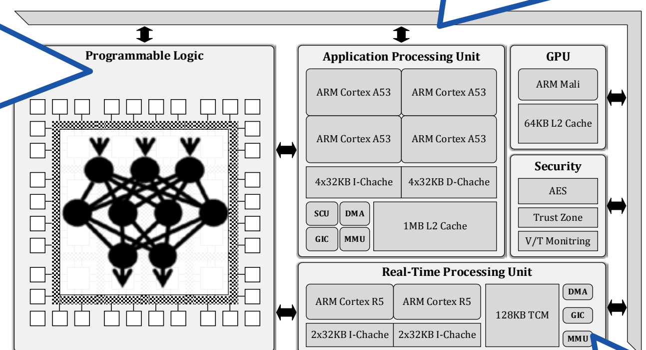


Potential applications:



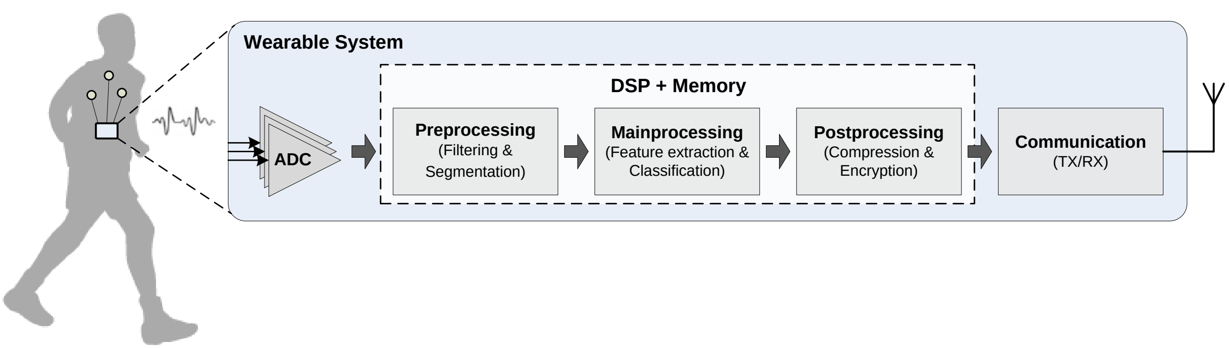
Bionix

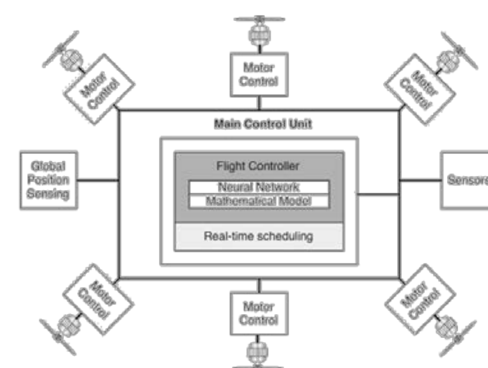




WSN/IoT/Wearables

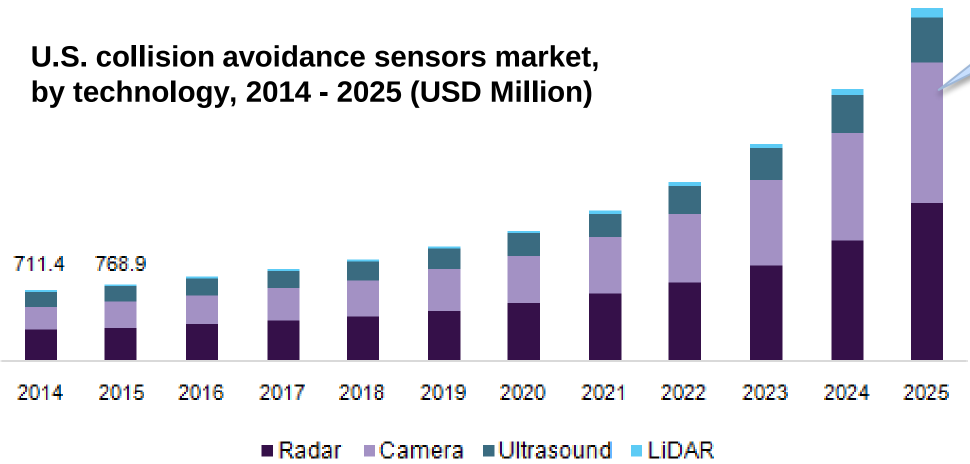
Autonomous UAV/vehicle/robot



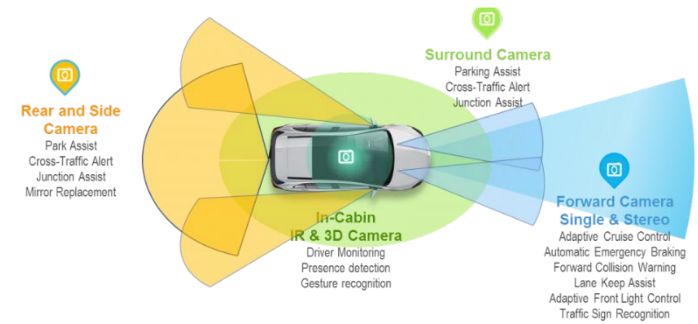


Autonomous vehicles & Technology Used

- Vision Camera:
  - Cameras are the only sensor technology that can *capture texture, color and contrast information* and the *high level of detail captured by cameras allow them to be the leading technology for classification*.
    - make camera sensors indispensable for autonomous systems.
  - The camera sensor technology play a very large role in autonomous vehicle.



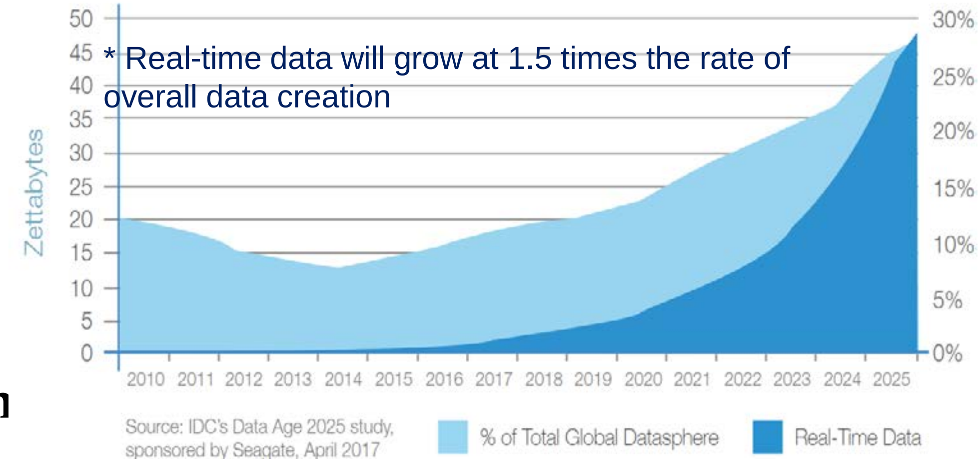
Camera will becomes domnant



Some examples of camera in the Advanced driver-assistance systems (ADAS) application:

- Adaptive Cruise Control (ACC)
- Automatic High Beam Control (AHBC)
- Traffic Sign Recognition (TSR)
- Lane Keep Systems (LKS)

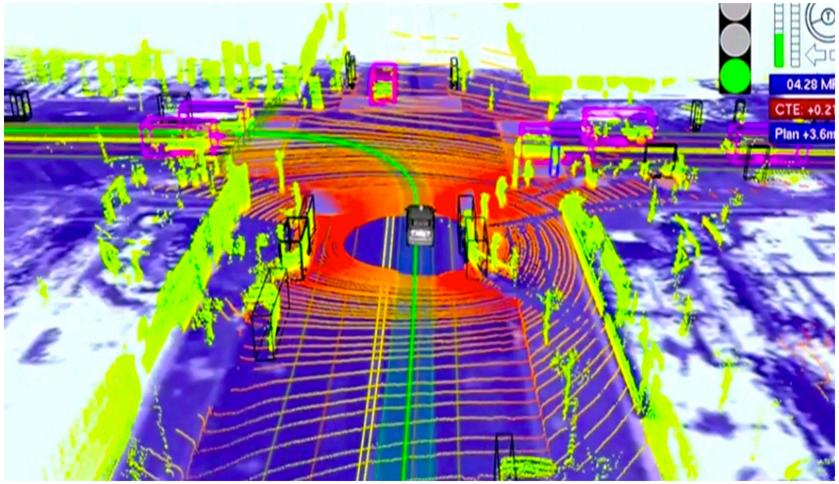
Heterogeneous Era



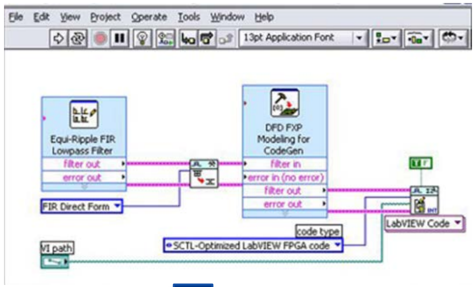
Heterogenous embedded platform

- We need high-performance embedded computing machines to deal with **huge amount of heterogeneous sensor inputs** while executing multiple algorithms for autonomy!
- Examples:
  Perception:
  - 3-D imaging with multiple lasers (LIDAR).
  - Edge-Detection Algorithm
  - Motion-Detection algorithm
  - Tracking algorithm

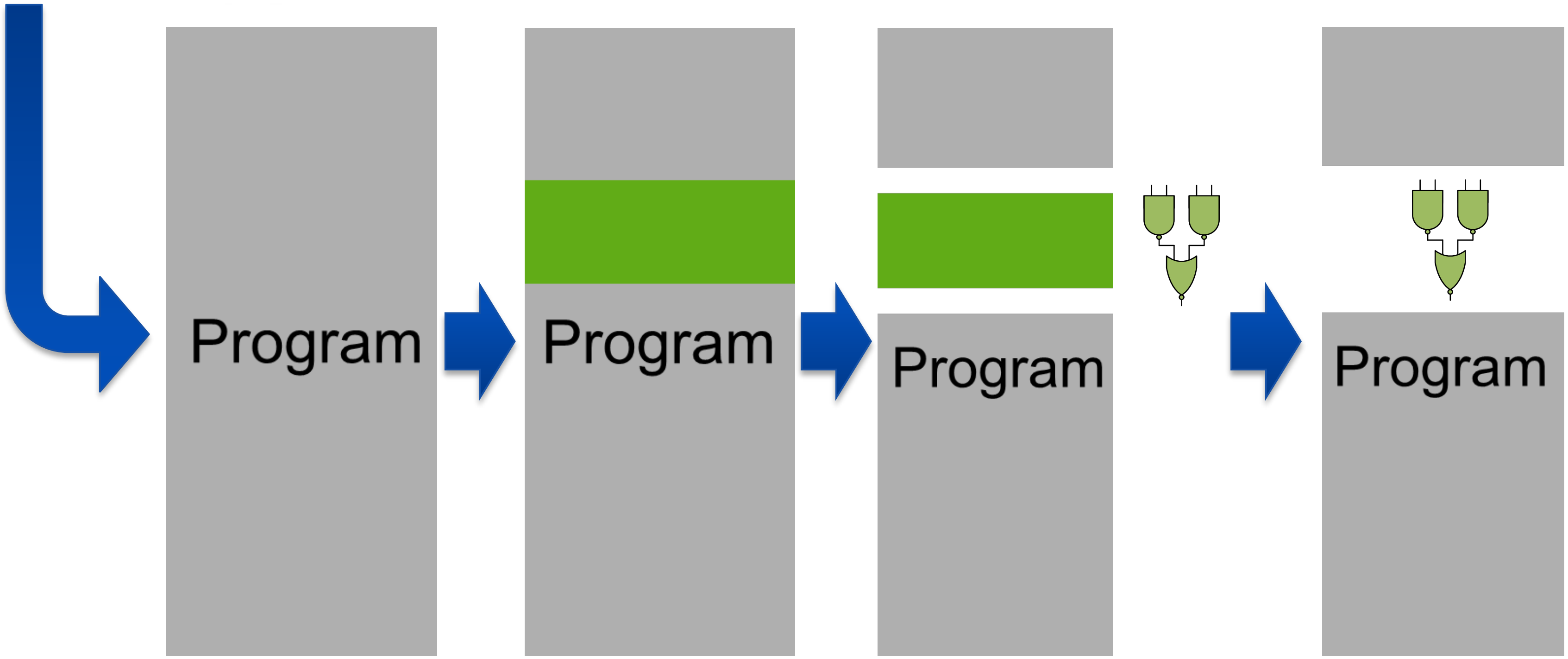
- (parallel) Heterogenenous Computing
- Deep Learning



Heterogeneous Era (cont.)



- HW&SW Integration
- Based on legacy code
- Map the legacy code to heterogeneous architgectures



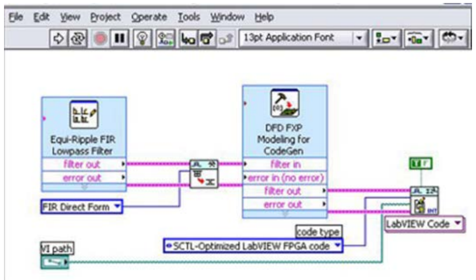
Code generation from modeling

Find a critical program component

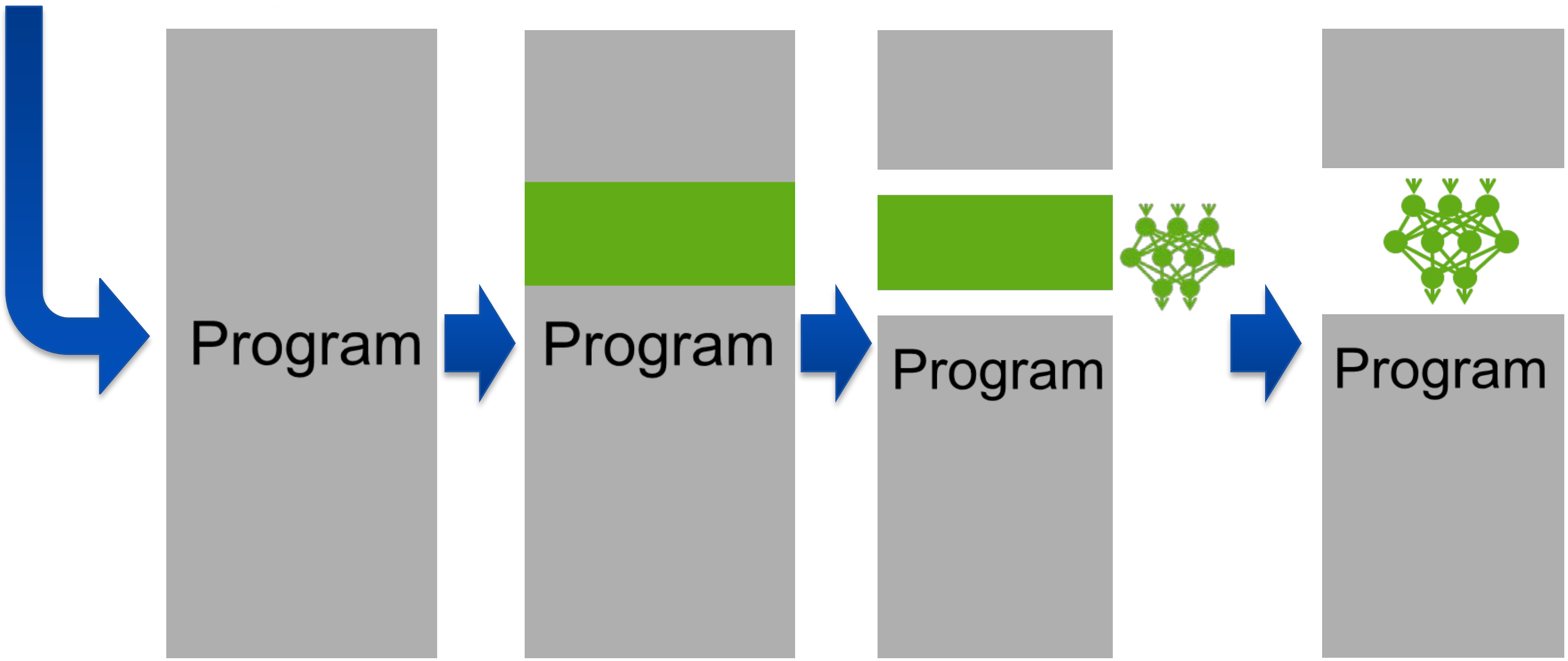
Compile the program & generate HW logics

Execute on the HW logic

Heterogeneous Era (cont.)



- HW&SW Integration
- Based on legacy code
- Generting and optimizing deep neural nerwork



Code generation from modeling

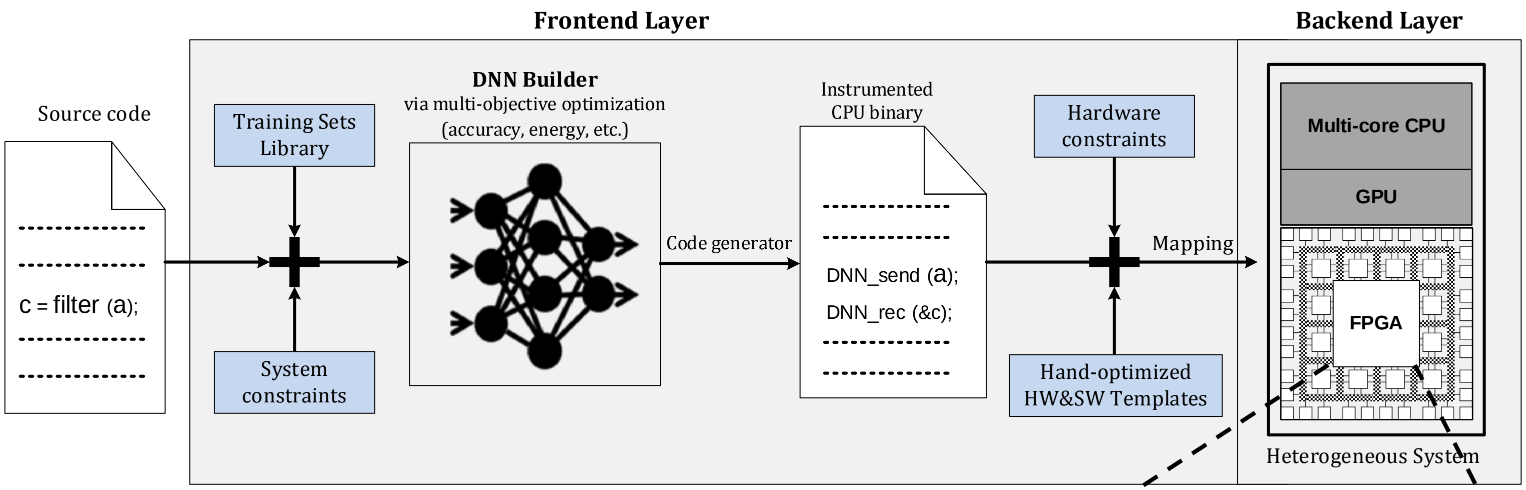
Find a critical program component

Compile the program , generate & train a NN

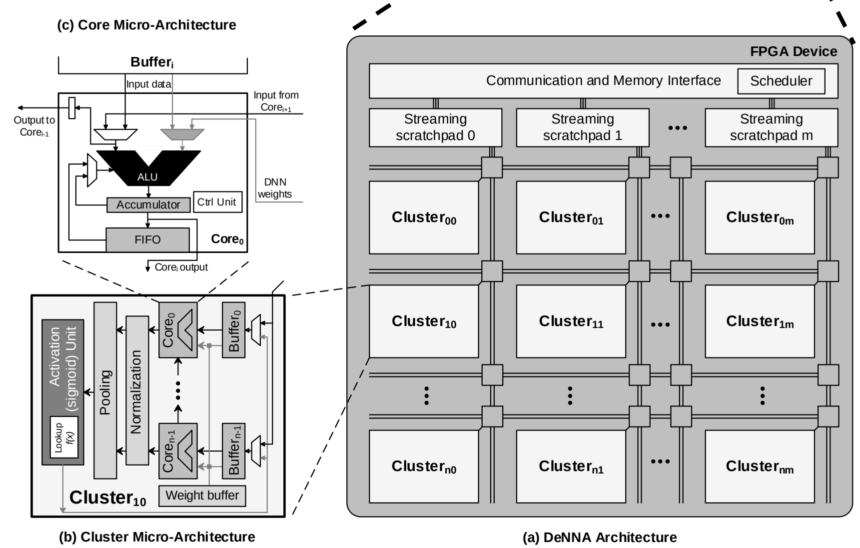
Execute on the smart and faster logic

Heterogeneous Era (cont.)

DeepMaker: Deep Learning Accelerator on Commercial Programmable Devices



Deep Neural Network Accelerators (DeNNA):



DeepMaker

- Mohammad will continue with some of the latest results
  - How we generate optimal models for deep networks
  - Some results for image processing with industrial dataset

DeepMaker Framework

Mohammad Loni, Masoud Daneshtalab
{mohammad.loni, masoud.daneshtalab}@mdh.se

School of Innovation, Design and Engineering
Mälardalen University, Sweden

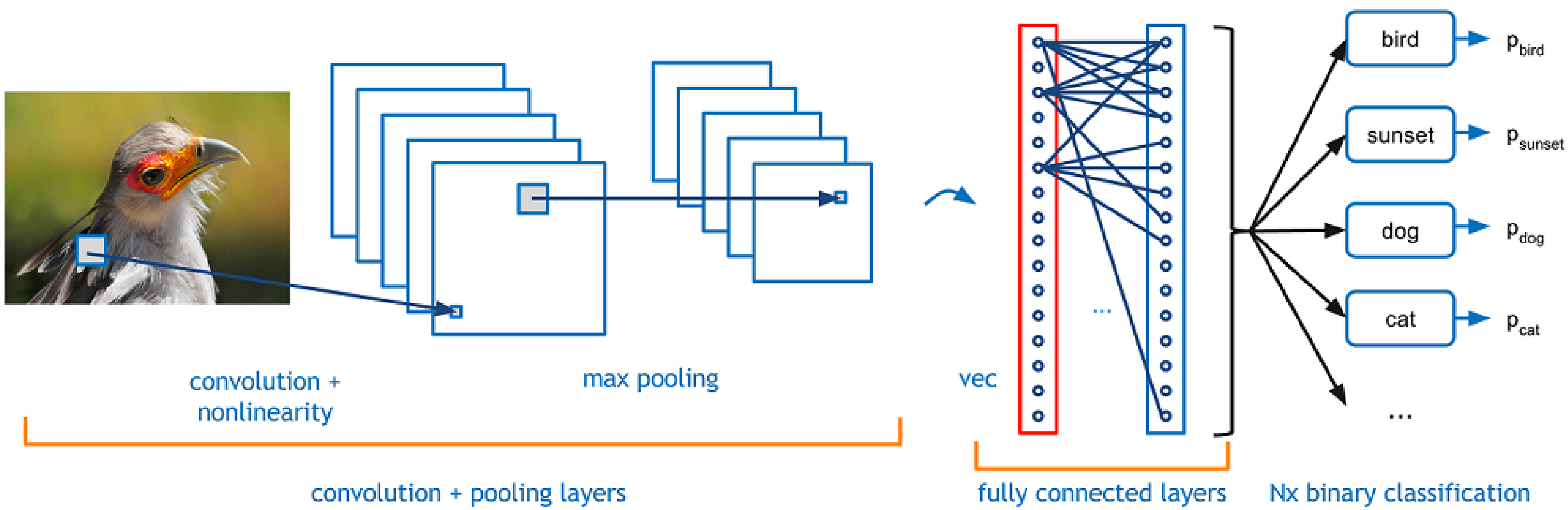
12 Dec. 2018, Västerås

Agenda

- Convolutional Neural Networks (CNNs)
- Processing Challenges of CNNs
- DeepMaker Framework
- Classification/Implementation Results
- Conclusion
- References

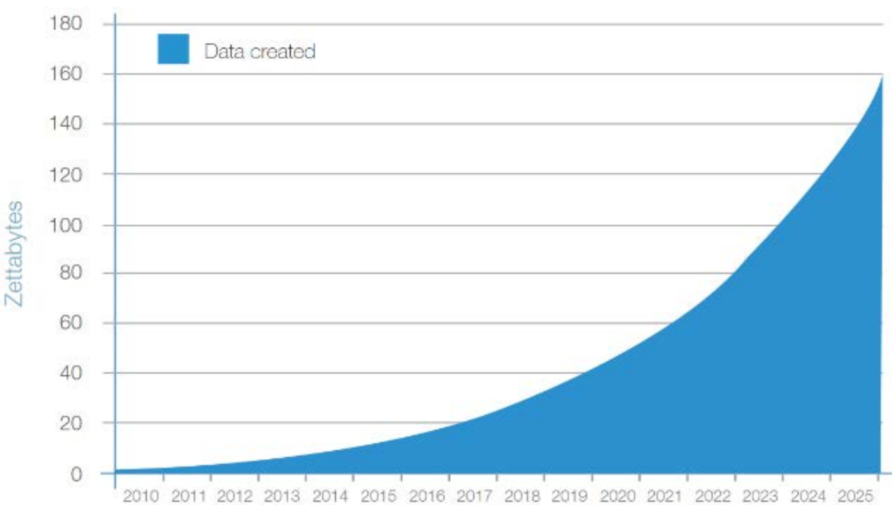
CNN

- CNN is composed of multiple layers running in sequence, where input data is fed to the first layer and output is a series of feature extraction kernels applied on the input image.
- Convolution, normalization, pooling, and activation layers are responsible for feature extraction, while fully-connected layers are for classification.

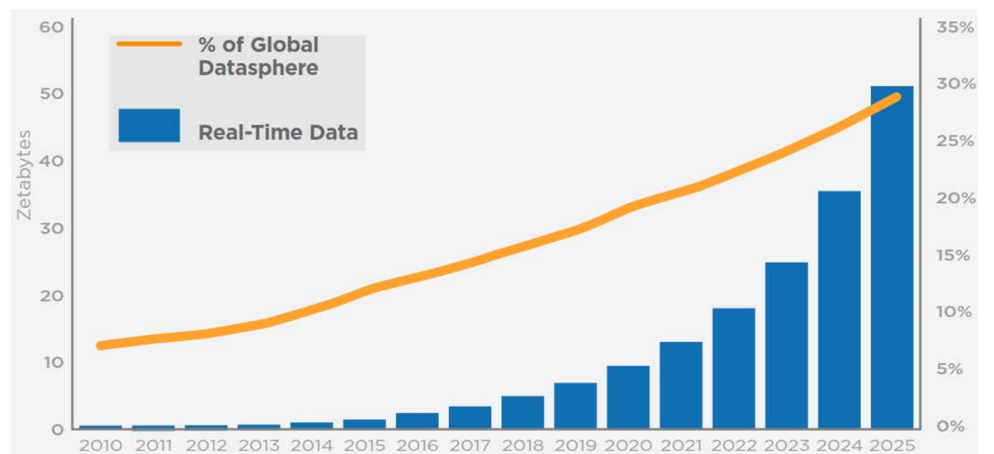


Processing Challenges

1. Dealing with **big amount of raw data** for modern applications
   - Modern Vision Camera: **10 Megapixel, 40 frame/sec**.
   - By 2025, **real-time data** (generated by IoT) will constitute nearly **30%** of all data created.



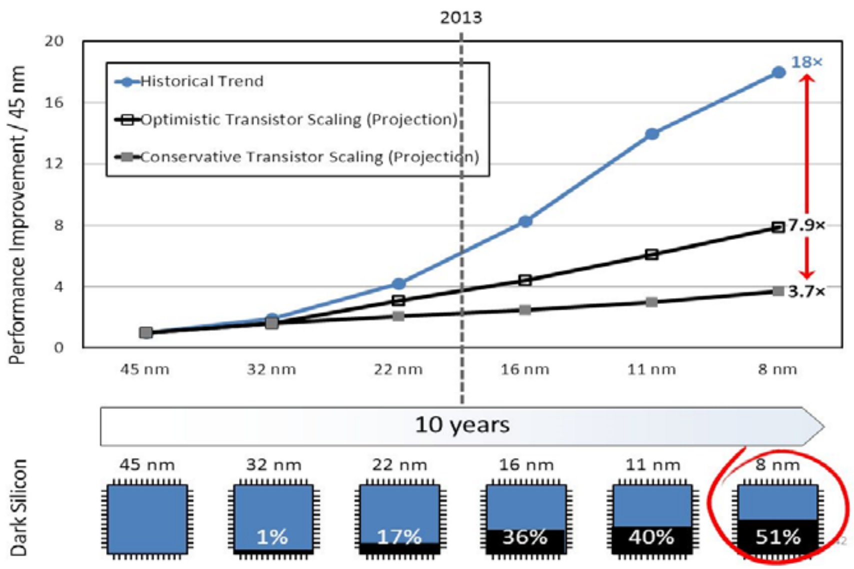
Annual Size of the Global Datasphere [1]

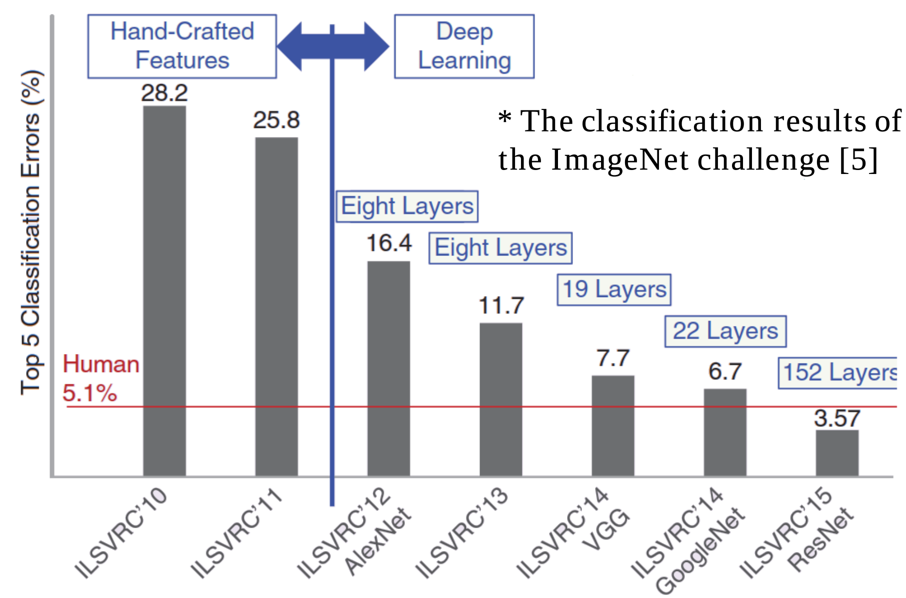


Data Creation by Type [1]

Processing Challenges in Big Data Era (cont.)

2. Traditional CMOS scaling no longer provides performance and efficiency gains due to the failure of Dennard scaling and Moore's law [3].



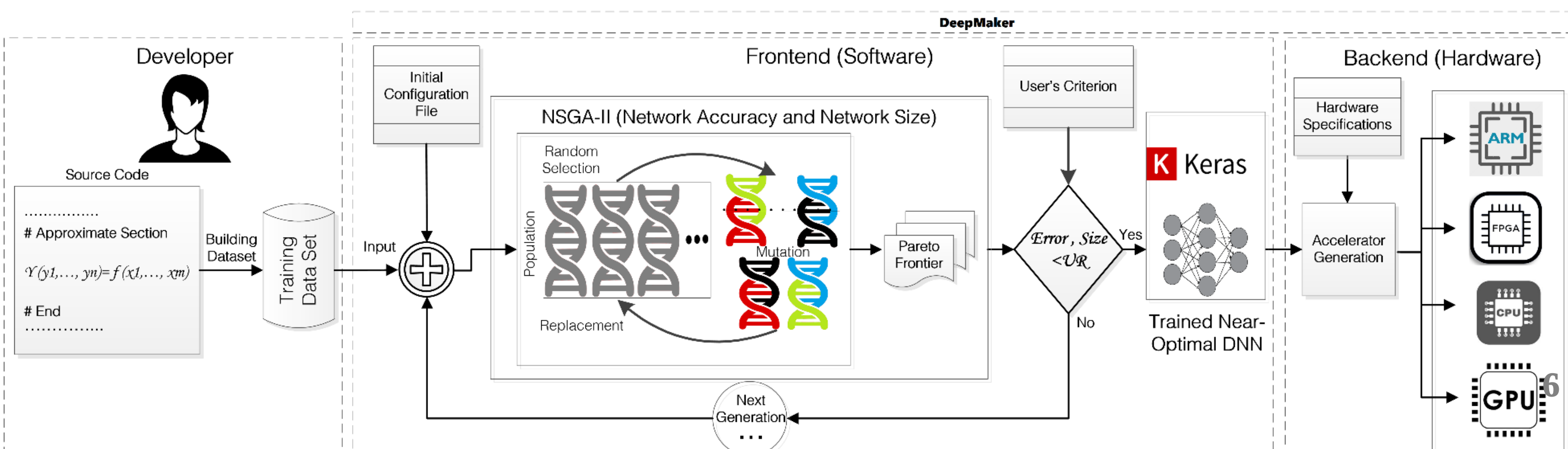


* The classification results of the ImageNet challenge [5]

3. Increasing the **complexity** of DL algorithms for achieving better accuracy [4].

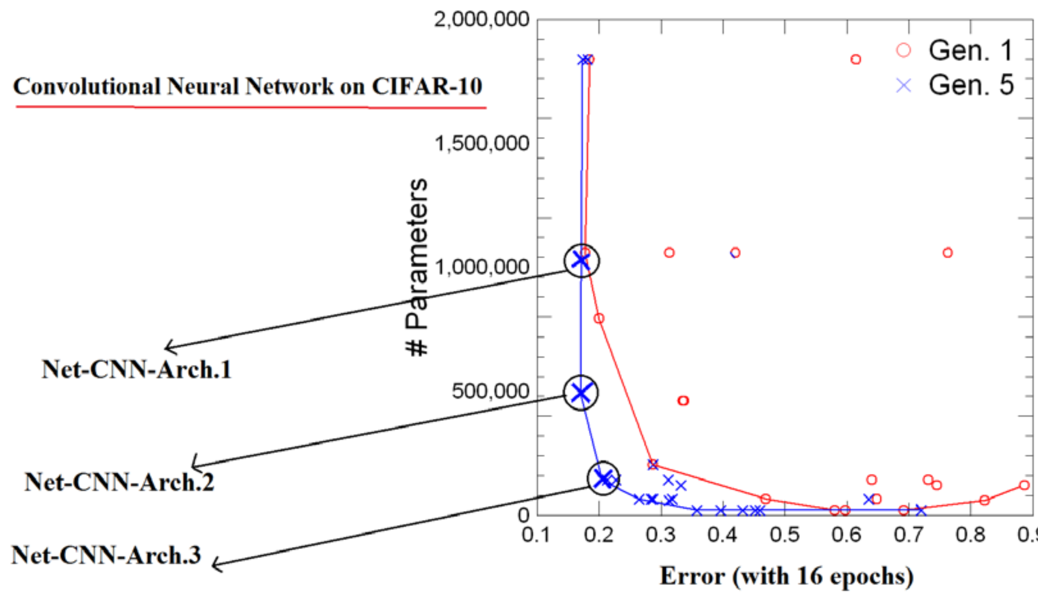
DeepMaker [4]

- We aim to tackle these challenges by providing a framework which generates synthesizable accelerators for CNNs
- **Front-end:** is responsible for Designing an **accurate** and **optimized** CNN architecture
  - The network *generalization proficiency*, *network complexity*, and execution time are depending on network architecture.
- **Back-end:** Efficient Implementation of generated CNN on different COTS processing platforms



Designing a Optimal CNN Architecture

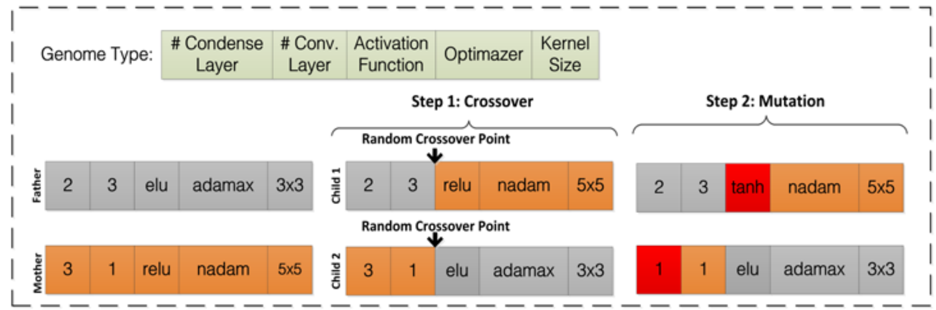
- Using a metaheuristic evolutionary solution for efficiently exploring the design space of CNN architectures
- Leveraging a **multi-objective** exploration strategy
  - ***Validation Accuracy***
  - ***Network Architectural Complexity***: Total number of trainable parameters
- **Output**: a set of Pareto frontiers including improved architectures



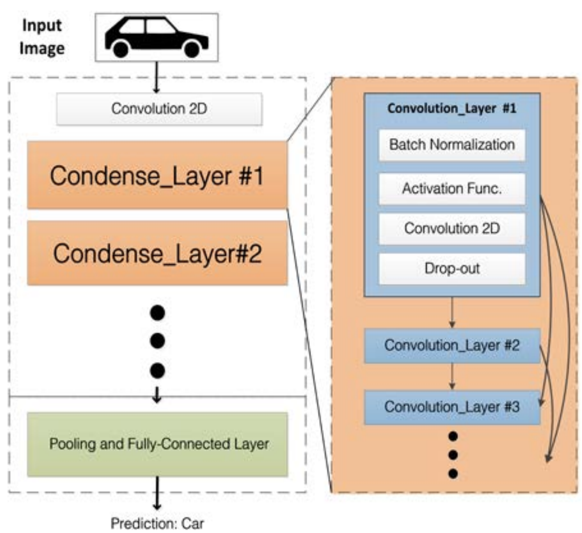
Selection Policy

- Representing the CNN architecture as a genome type

Parameter	Deep CNN
Activation Function	hard-sigmoid, relu, elu, tanh, sigmoid, softplus, linear
# Condense_Layer	1, 2, 3, 4
# Convolution_Layer	16, 28, 40, 52
Kernel Size	3x3, 5x5
Optimizer	rmsprop, adam, sgd, adagrad, adadelta, adamax, nadam



- Pruning the design space by taking inspirations from DenseNet arch.



The template architecture of generated networks

Results

- Training Datasets
  - **MNIST**: This is a dataset of black and white images for handwritten digit recognition
  - **CIFAR-10**: This is a complex colorful benchmark dataset of natural images used for object recognition
- Getting the **total execution time** as the evaluation metric since communication time is vital for embedded implementations
- We also did not use any network compression technique to only assess the influence of network architecture on inference time.
- Initial Configuration:
  - **Epoch**=30, **batch size**=128, **number of generations**=5, **initial population**=2.
- **Back-end side:**

CPU	Core i7-7820
GPU	Tesla M60
ARM	Cortex-A15
FPGA	Xilinx UltraScale+

Platform	CPU	GPU	ARM	FPGA
Frequency (GHz)	2.9	1.178	1.9	.8
Technology (nm)	14	28	28	16 (FinFET+)
TDP (W)	45	300	5	-
Cores/Total Thread	4/8	4096 CUDA Cores	8/8	FF= $2.5(\mathrm{x}10^6)$ LUT= $1.18(\mathrm{x}10^6)$ DSP= 6800
Memory	8MB Cache	16GB GDDR5	2.5MB Cache	BRAM= 75.9 Mb
Approx. Price (USD)	378$	7,532$	60$/board	-

Classification Results

- **MNIST** (Compare to MetaQNN by ***Google***): **43x** compression rate, **0.06%** accuracy loss
- **CIFAR-10** (Compare to the most accurate): **26.4x** compression Rate, **4%** accuracy loss
- **CIFAR-10** (Compare to MetaQNN by ***Google***): **6.92x** compression Rate, **4.2% better accuracy**

Dataset	Method	#Params (x10^6)	Error (%)
Google	MetaQNN [21]	5.59	.35
	EDEN [28]	1.8	1.6
	SimpleNet [29]	.3	.25
MNIST	Wan et al. [30]	-	.21
	Our MNIST-MLP	.19	1.2
	Our MNIST-CNN	**.13**	**.41**
	NAS-v1/v3 [22]	4.2/37.4	5.50/3.65
	SimpleNet [29]	5.48	4.68
	VGG-16 [31]	138	7.55
	DenseNet (k=12)-40 [6]	1.0	7.0
	DenseNet (k=12)-100 [6]	7	5.77
	DenseNet (k=24)-100 [6]	27.2	5.83
	EDEN [28]	.17	25.6
Most Popular	ResNet-20 [27]	0.27	8.75
	ResNet-110 [27]	1.7	6.43
CIFAR-10	Masanori et al. [24]	1.68	5.98
	Block-QNN-22L [23]	39.8	3.54
Google (RL)	MetaQNN [21]	6.92	11.18
	Real et al. [25]	5.4	5.4
	Gastaldi et al. [26]	26.4	2.86
	Our Net-MLP	0.66	37.0
	Our Net-CNN-Arch.1	**1.0**	**6.9**
	Our Net-CNN-Arch.2	**0.49**	**8.7**
	Our Net-CNN-Arch.3	**0.14**	**14.1**

1.7x compression rate, **0.5%** accuracy loss

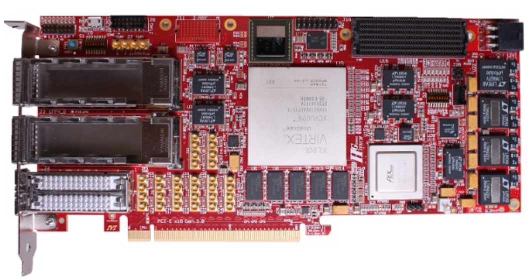
5.4x compression rate, **1.5%** accuracy loss

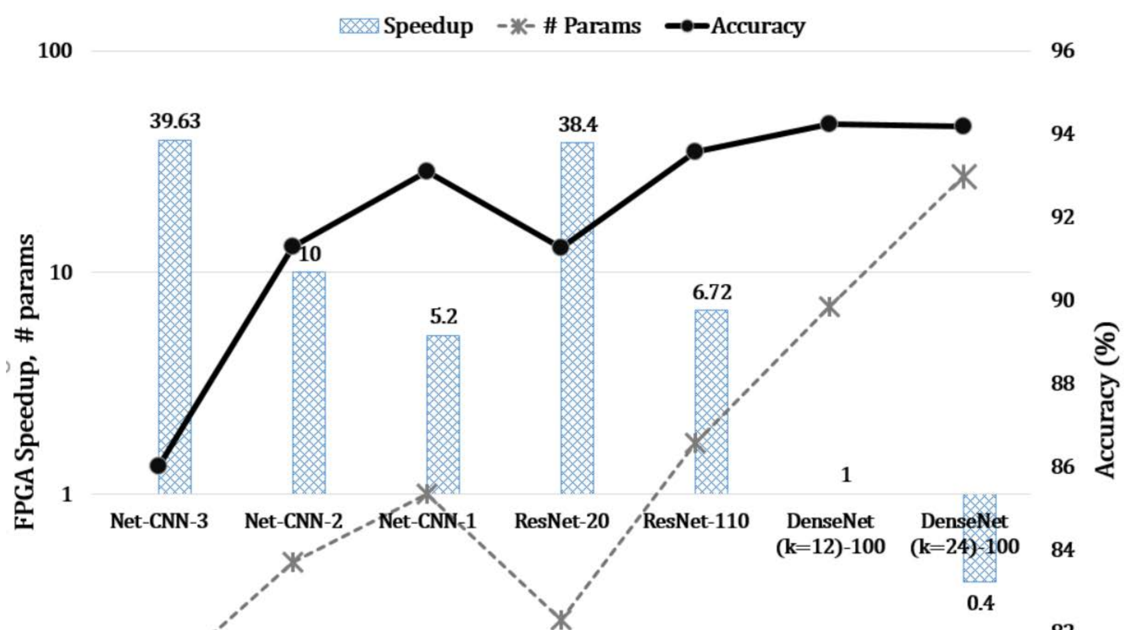
Implementation Results

- All the results have been compared with *DenseNet* (7 M params) as the most accurate network

- **FPGA execution time**

Generated Network	Speedup	Accuracy Loss (%)
Net-CNN-Arch.1	5.2x	1.13
Net-CNN-Arch.2	10x	2.93
Net-CNN-Arch.3	39.63x	8.33

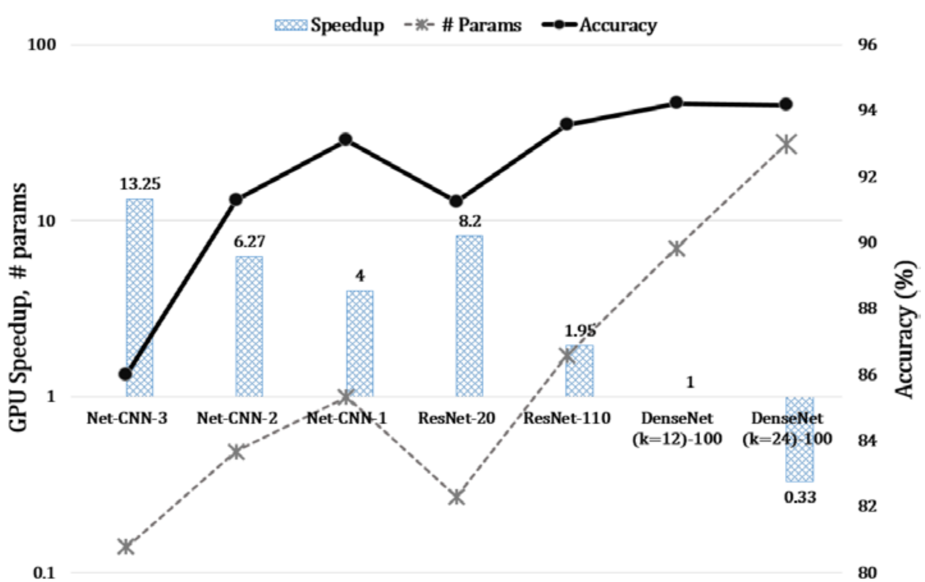




Implementation Results on GPU and ARM

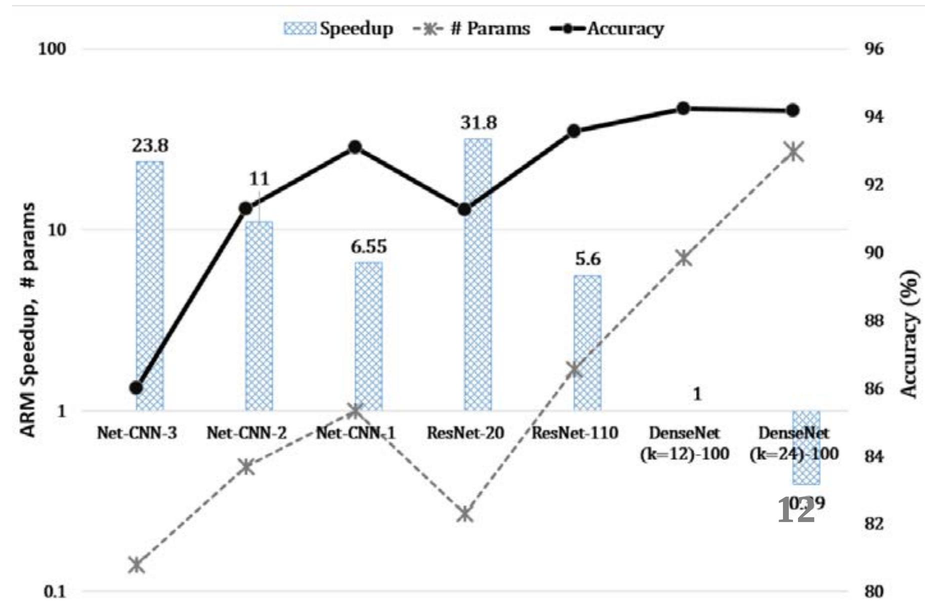
- **GPU**

Generated Network	Speedup	Accuracy Loss (%)
Net-CNN-Arch.1	4x	1.13
Net-CNN-Arch.2	6.27x	2.93
Net-CNN-Arch.3	13.25x	8.33



- **ARM Processor**

Generated Network	Speedup	Accuracy Loss (%)
Net-CNN-Arch.1	6.55x	1.13
Net-CNN-Arch.2	11x	2.93
Net-CNN-Arch.3	23.8x	8.33



Results

- All the results have been achieved by running on NVIDIA GTX 1080ti.

- **4x** better inference time (total execution time)
- **4.22x** more energy efficiency
- **4.15 %** more accurate results

Solution	Facebook, 2018 [2]	DeepMaker
AVG. Accuracy	86.95 %	**91.1%**
Inference Time (ms)	63	**16**
Frame/Second	15	**62**

Conclusion

- Deep convolutional neural networks are complex processing models which their implementation is challenging especially on embedded devices
- To tackle these challenges, we proposed a multi-objective evolutionary approach which automatically design a highly optimized CNN arc. for COTS processing platforms.
- The evaluation results demonstrate the effectiveness of DeepMaker on complex image datasets

References

- [1] D. Reinsel, J. Gantz, and J. Rydning, Data Age 2025 - The Evolution of Data to Life-Critical: Don not Focus on Big Data; Focus on the Data That is Big, IDC White Pap., no. April, pp. 125, 2017.
- [2] Lin, Tsung-Yi, Priyal Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. "Focal loss for dense object detection." *IEEE transactions on pattern analysis and machine intelligence*(2018).
- [3] H. Esmaeilzadeh, E. Blem, R. St. Amant, K. Sankaralingam, and D. Burger, Dark silicon and the end of multicore scaling, IEEE Micro, vol. 32, no. 3, pp. 122134, 2012.
- [4] M. Loni, M. Daneshtalab, and M. Sjödin, ADONN: Adaptive Design of Optimized Deep Neural Networks for Embedded Systems, Euromicro Conference on Digital System Design (DSD), Prague, Czech, 2018.

Thank you!