Design and Application of Energy Efficient Emerging NoC Architectures (E3NoC)

Salvatore Monteleone, PhD

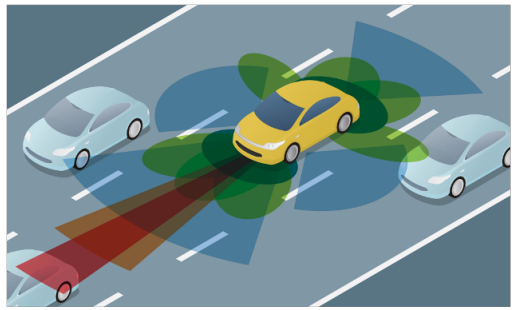
Agenda

- Introduction
- Network on Chip (NoC) Paradigm
- Emerging NoC Architectures
- Design and Simulation
- Applications

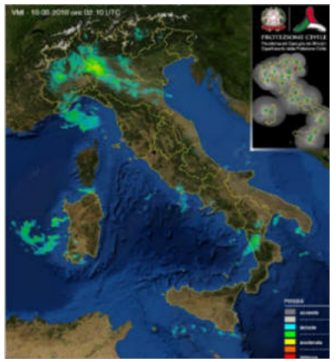
Introduction

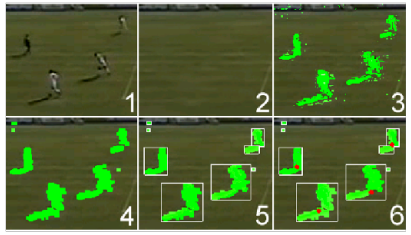
Demand for computation

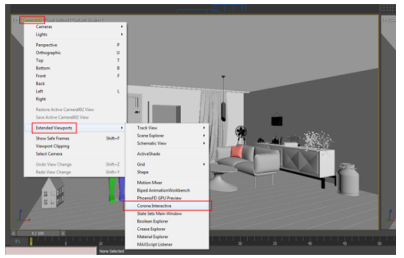
Mobility scenarios

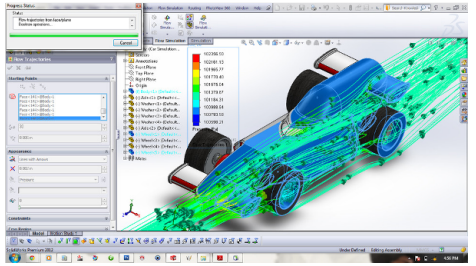


Computate intensive applications









Moore's Law

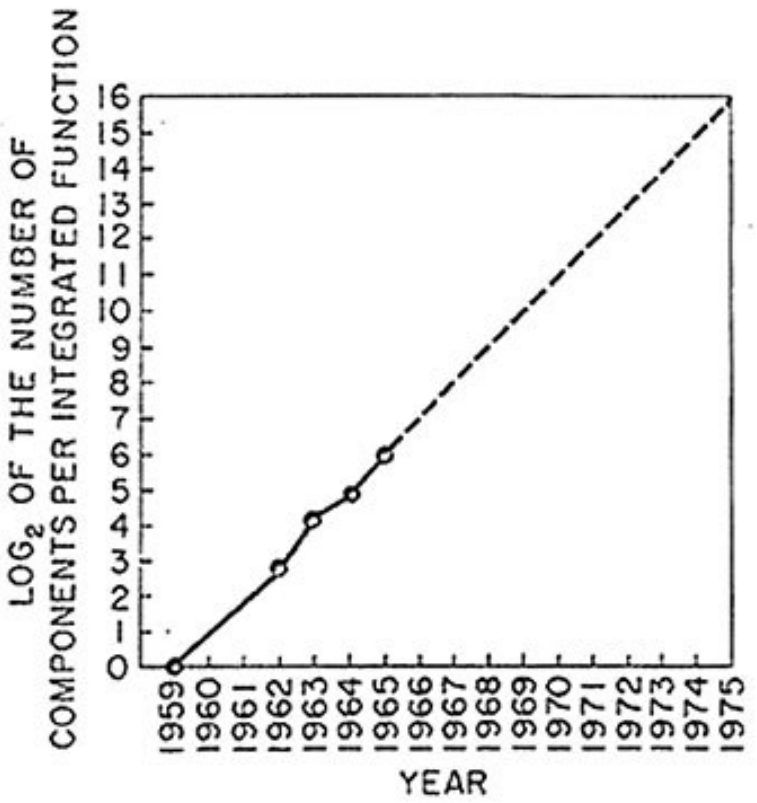
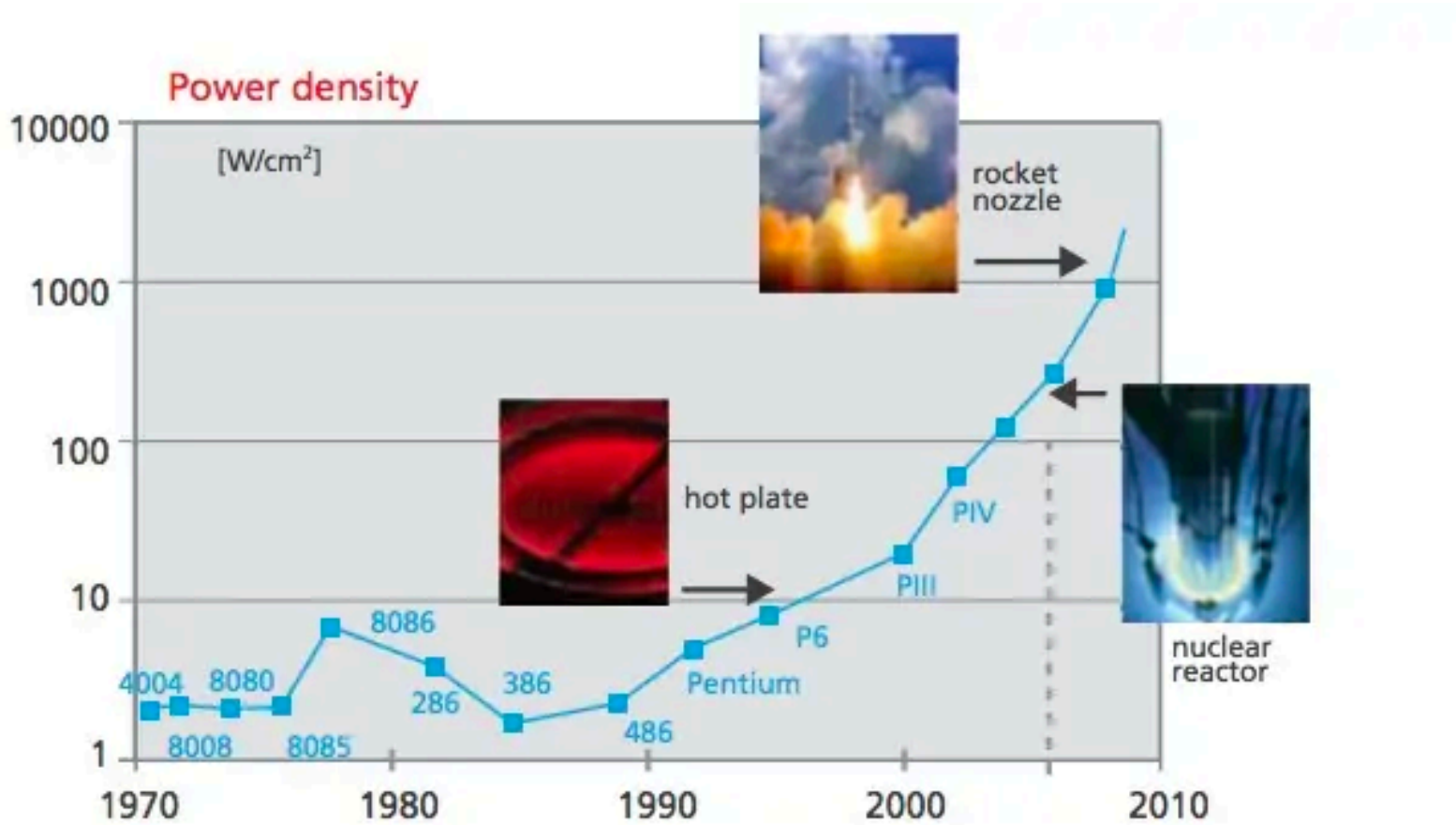


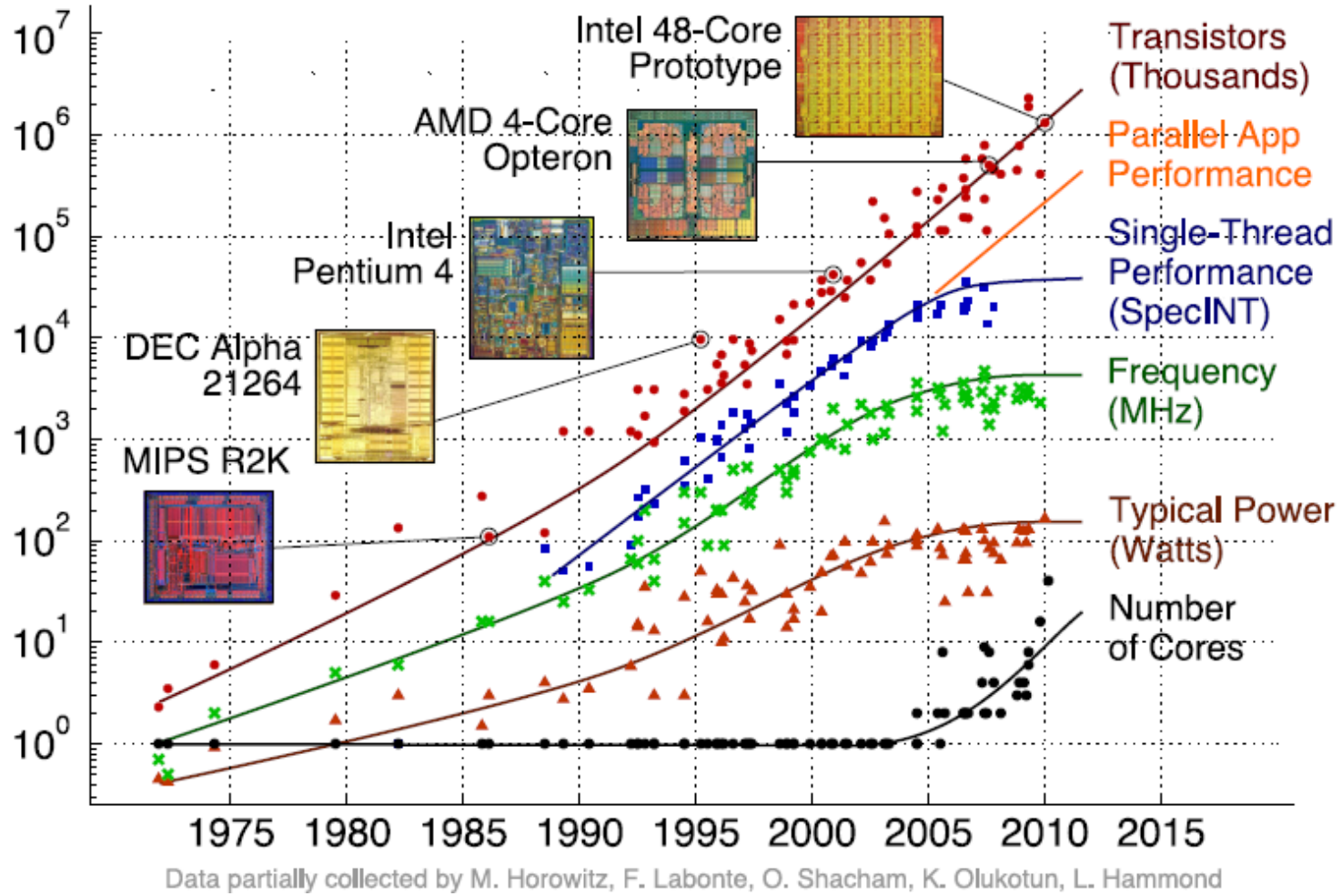
Fig. 2 Number of components per integrated function for minimum cost per component extrapolated vs time.

Power density problem

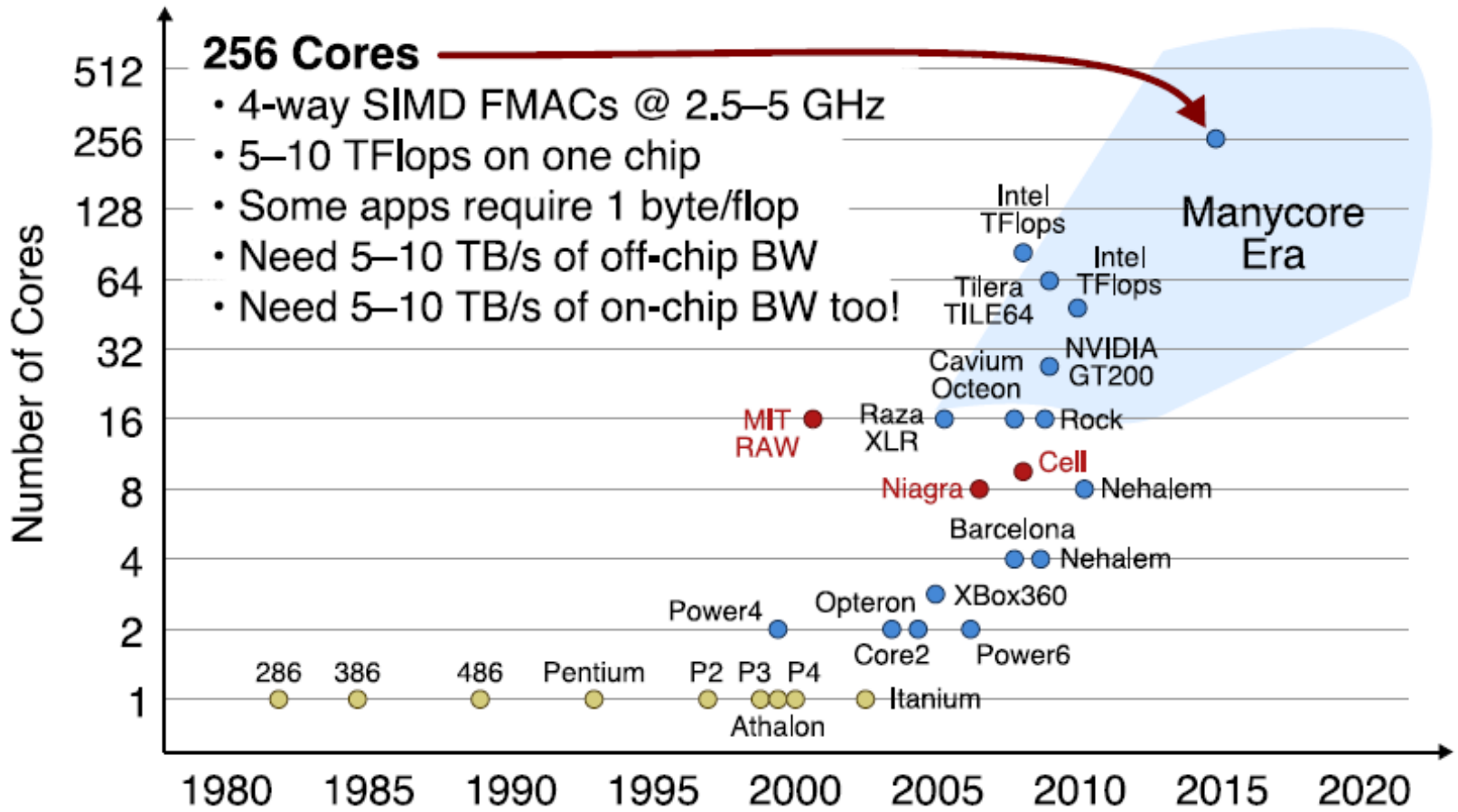


The growing power density (measured in W/cm2) of Intel's microchip processor families. (Source: Intel)

Trends in SoCs

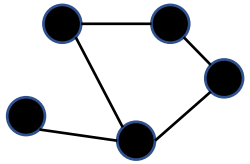


Trends in SoCs

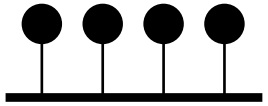


Network on Chip (NoC) Paradigm

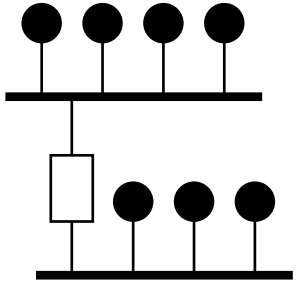
Communication Architectures



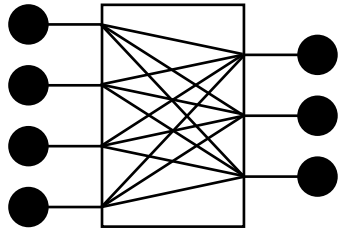
1. Custom



2. Shared Bus

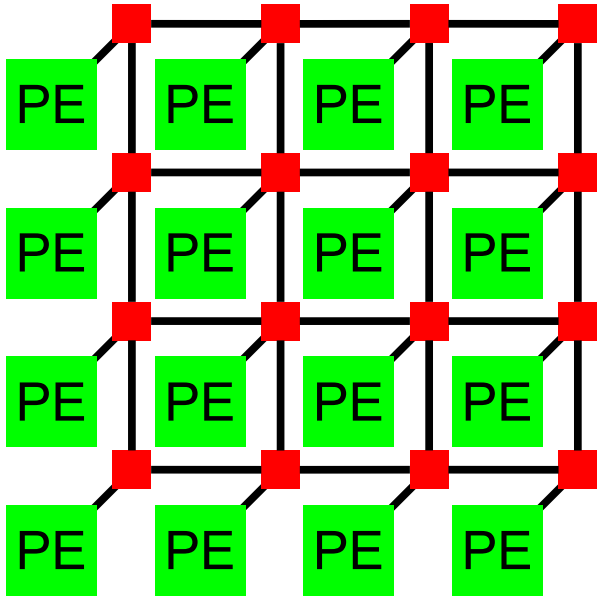


3. Hierarchical Bus



4. Bus Matrix

Communication Architectures

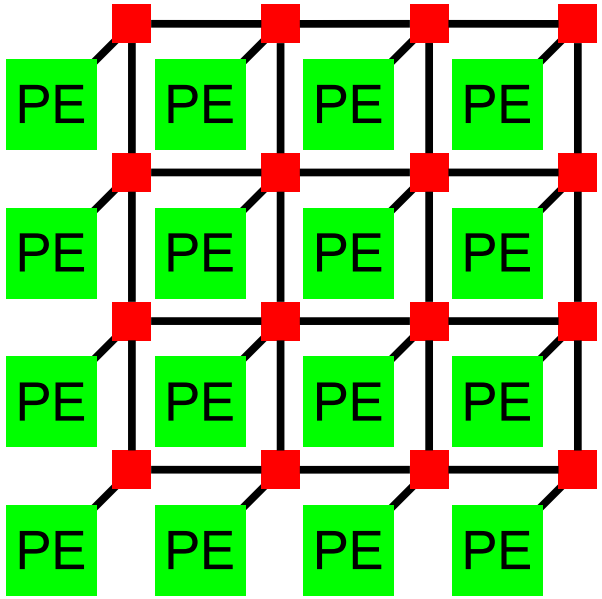


5. Network-on-Chip

A Network-on-chip (NoC) is a packet switched on-chip communication network designed using a layered methodology.

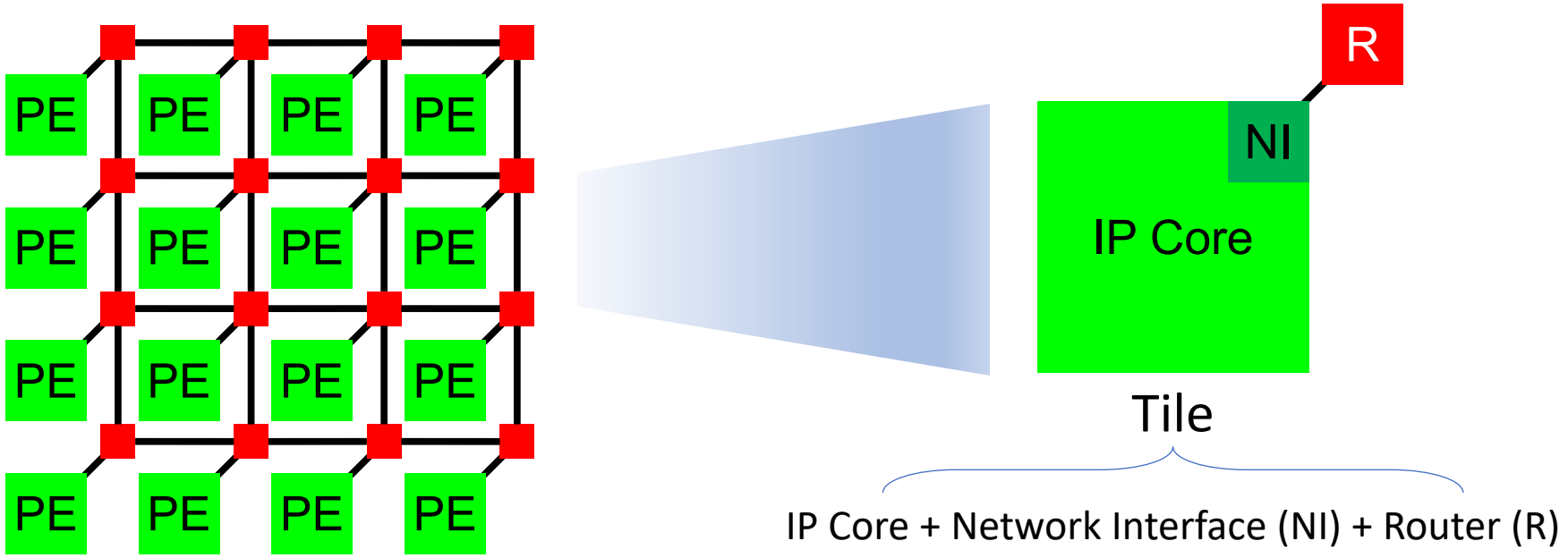
It "routes packets, not wires”.

Network-on-Chip



NoCs are an attempt to scale down the concepts of largescale networks, and apply them to the embedded System-on-Chip (SoC) domain

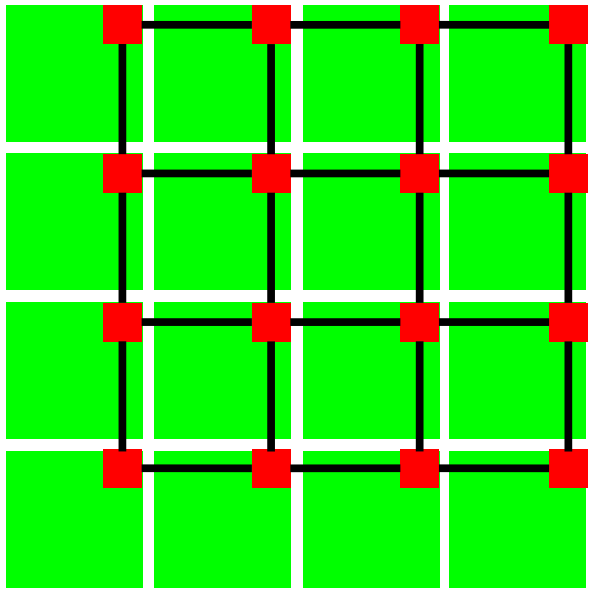
Network-on-Chip



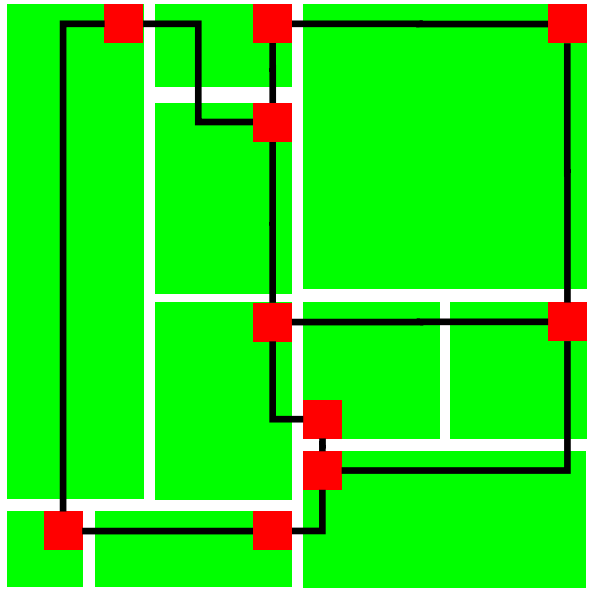
Network-on-Chip: Properties

- Regular geometry that is scalable
- Flexible QoS guarantees
- Higher bandwidth
- Reusable components
  - Buffers, arbiters, routers, protocol stack
- No long global wires (or global clock tree)
  - No problematic global synchronization
  - GALS: Globally asynchronous, locally synchronous design
- Reliable and predictable electrical and physical properties

Network-on-Chip: Homogeneous vs. Heterogeneous

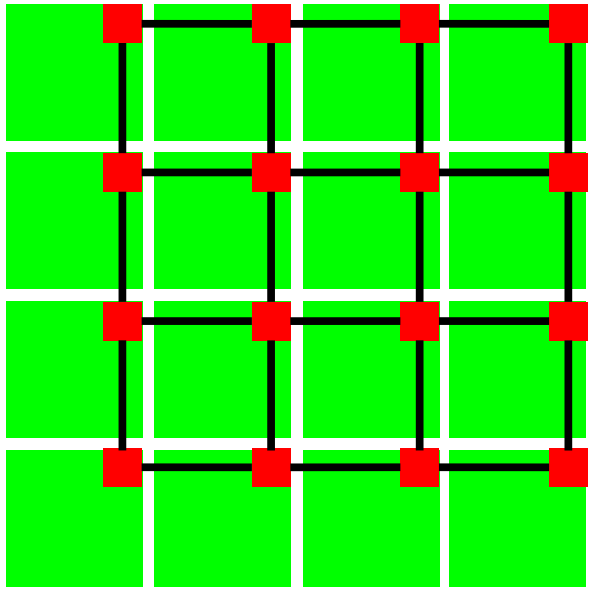


Homogeneous



Heterogeneous

Network-on-Chip: Homogeneous vs. Heterogeneous

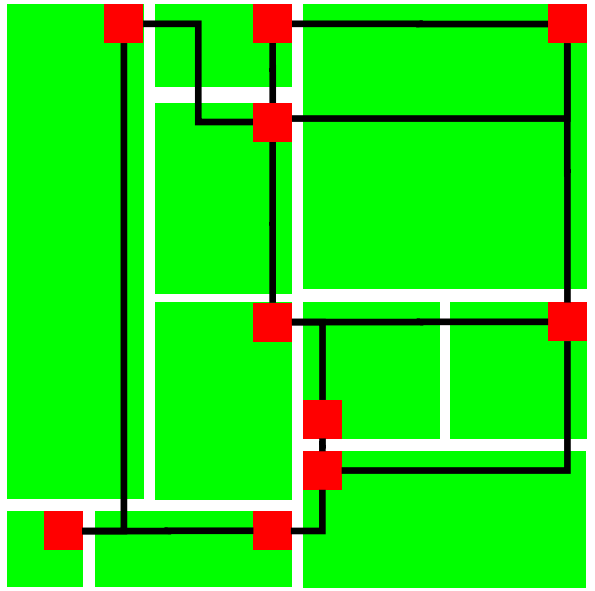


Homogeneous

- Each tile is a simple processor
- Easy tile replication (scalability, predictability)
- Less performance
- Low network resource utilization

Network-on-Chip: Homogeneous vs. Heterogeneous

- IPs can be: General purpose/DSP processor, Memory, FPGA, I/O core
- Better fit to application domain
- Most modern systems are heterogeneous
- Topology synthesis: more difficult
- Needs specialized routing



Heterogeneous

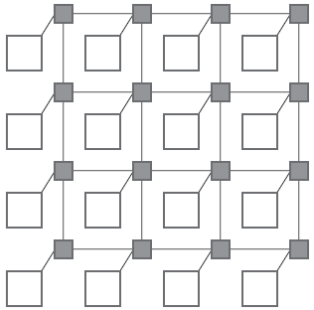
Network-on-Chip: Performance

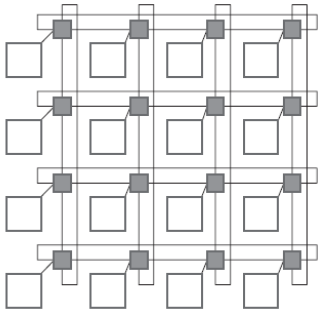
Factors that influence the performance of a NoC are

- Topology
- Router Architecture
- Routing Technique
- Flow Control
- Traffic Pattern

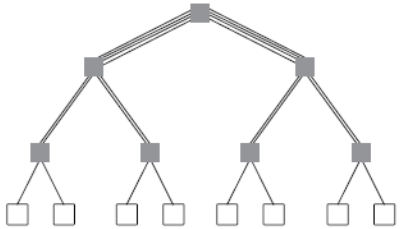
NoC Topologies

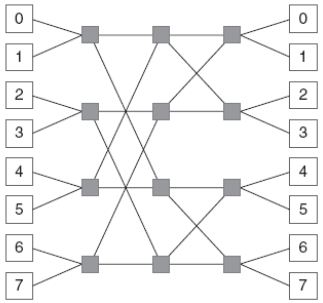
- Direct Topologies



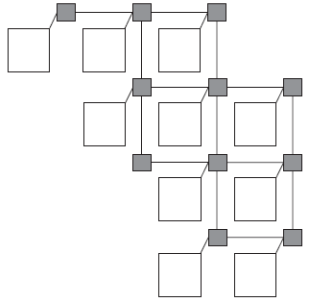


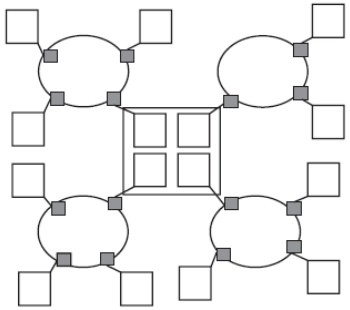
- Indirect Topologies



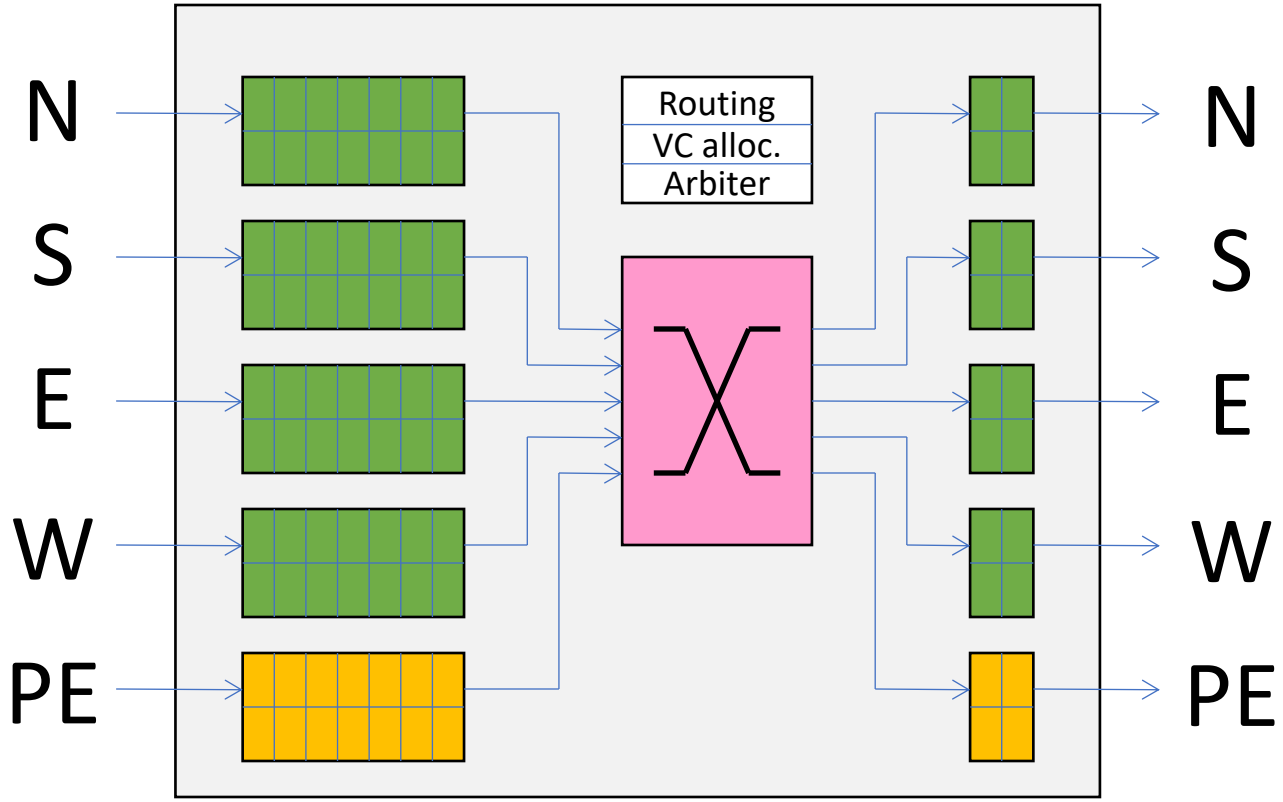


- Irregular or ad-hoc network topologies





Router Architecture

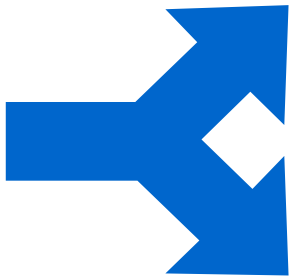


Taxonomy of Routing Algorithms

Oblivious

- Deterministic
- Random

Adaptive

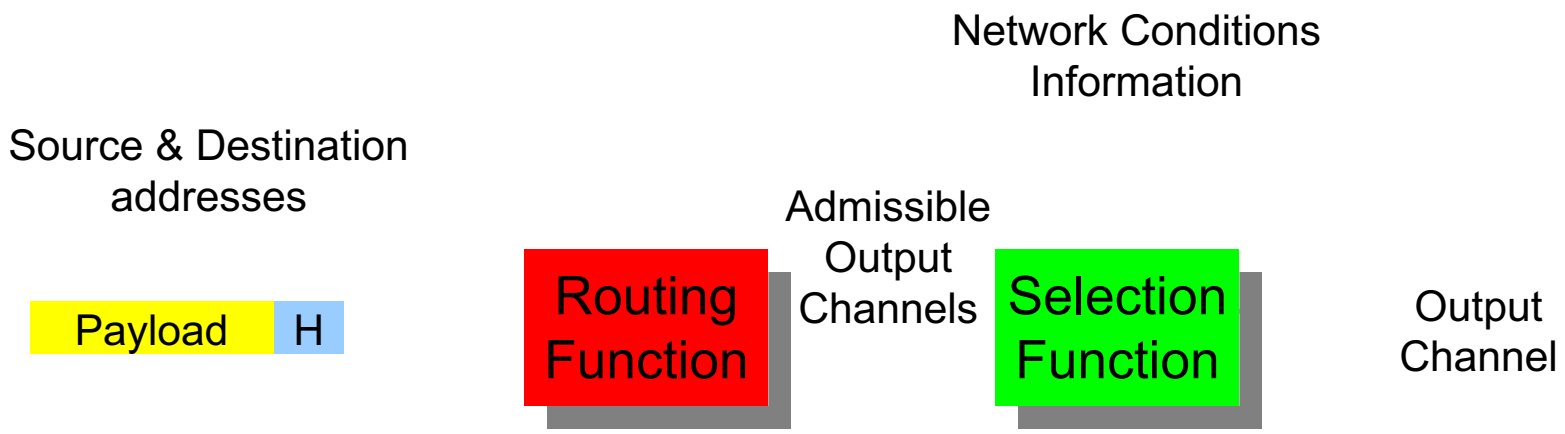


Minimal

Non-minimal

Routing and Selection

- The routing algorithm can be represented as a *routing relation R* and a *selection function S*
- *R* returns a set of paths or channels and *S* selects between the route to be taken



Flow Control

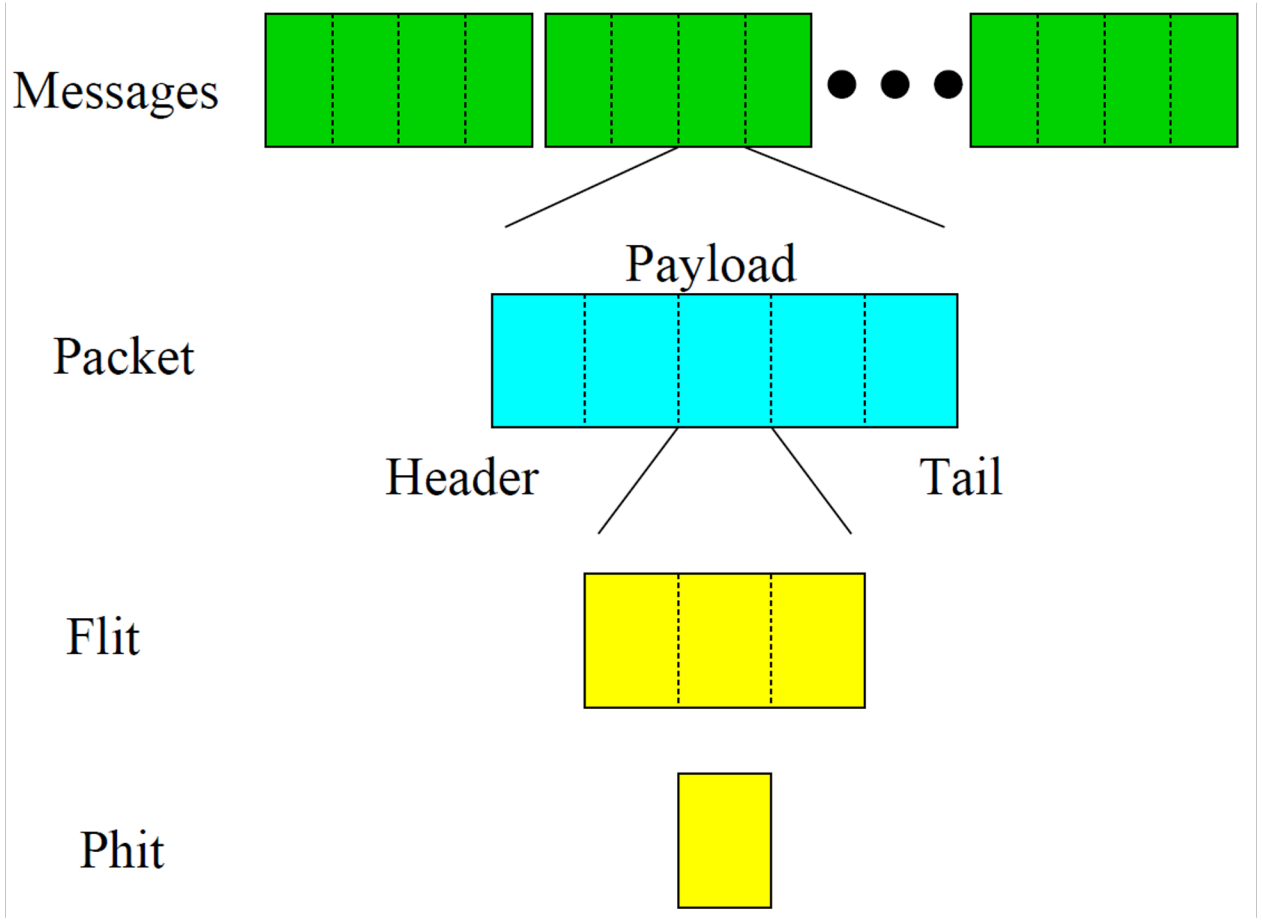
- Flow Control can be viewed as a problem of
  - Resource allocation
  - Contention resolution
- Resources in form of channels, buffers and state must be allocated to each packet
- If two packets compete for the same channel flow control can only assign the channel to one packet, but must also deal with the other packet

Flow Control

Flow Control can be divided into:

- Bufferless flow control: Packets are either dropped or misrouted
- Buffered flow control: Packets that cannot be routed via the desired channel are stored in buffers

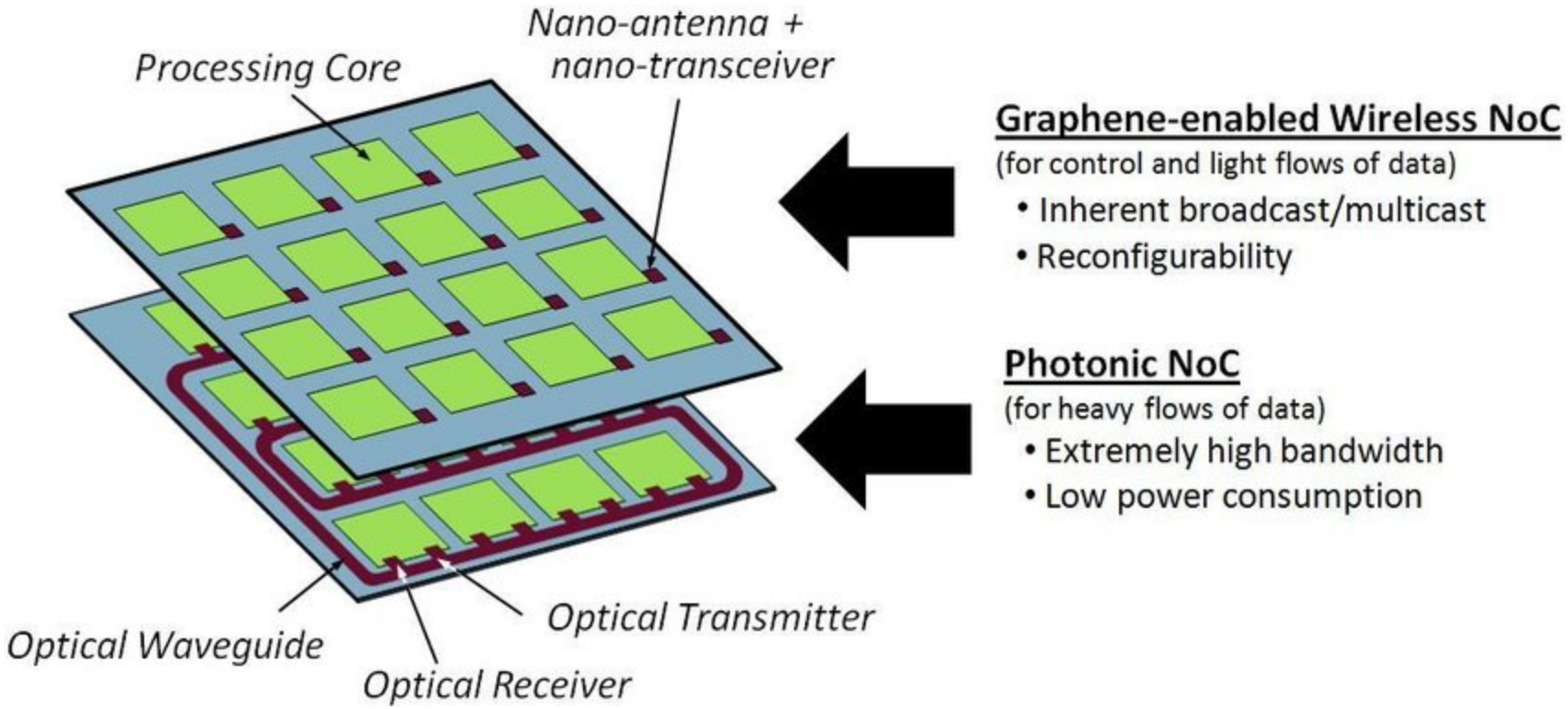
Data Units



Emerging NoC Architectures

Emerging NoC tecnologies

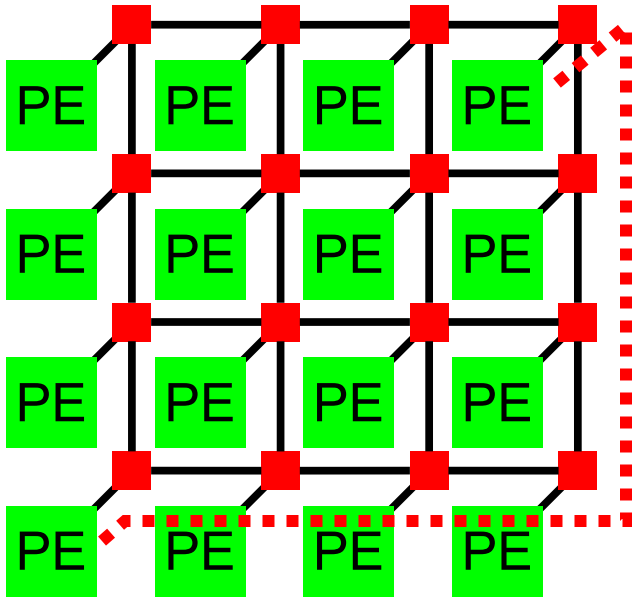
- Optical NoC
- RF-NoC
- Hybrid NoC



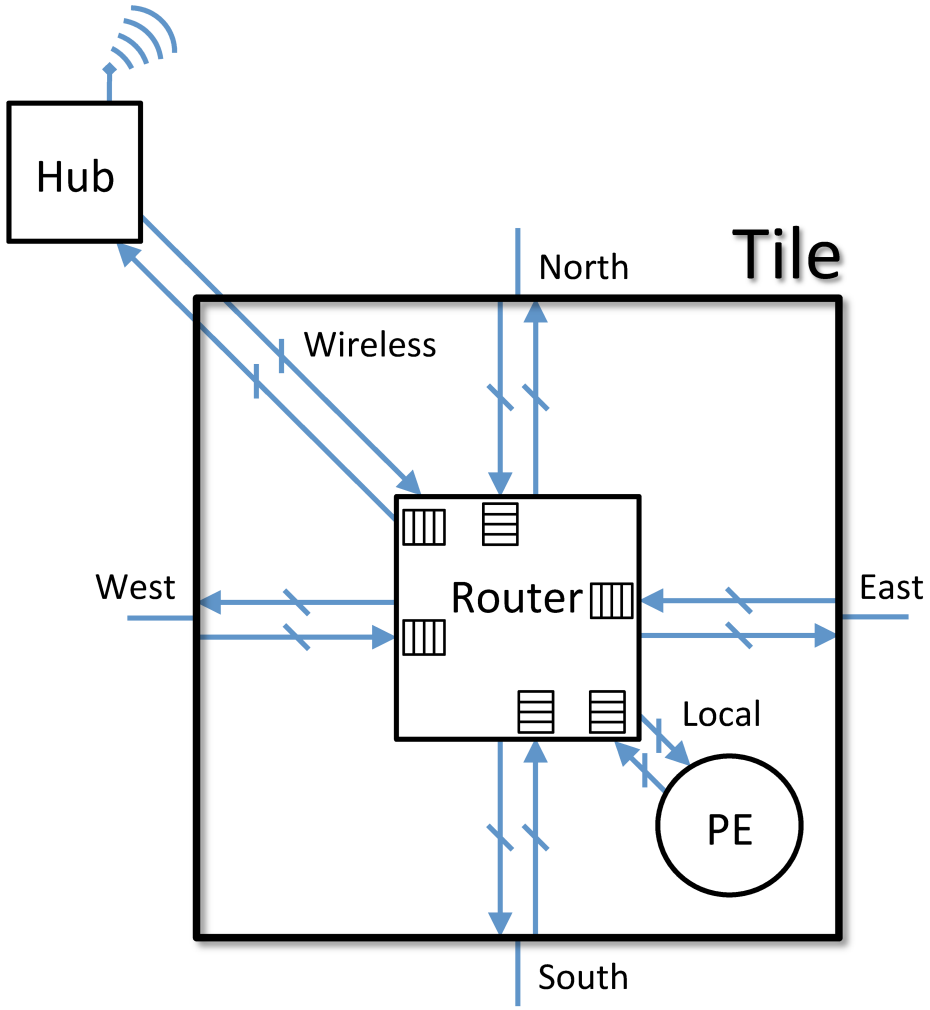
Wireless NoC Motivation

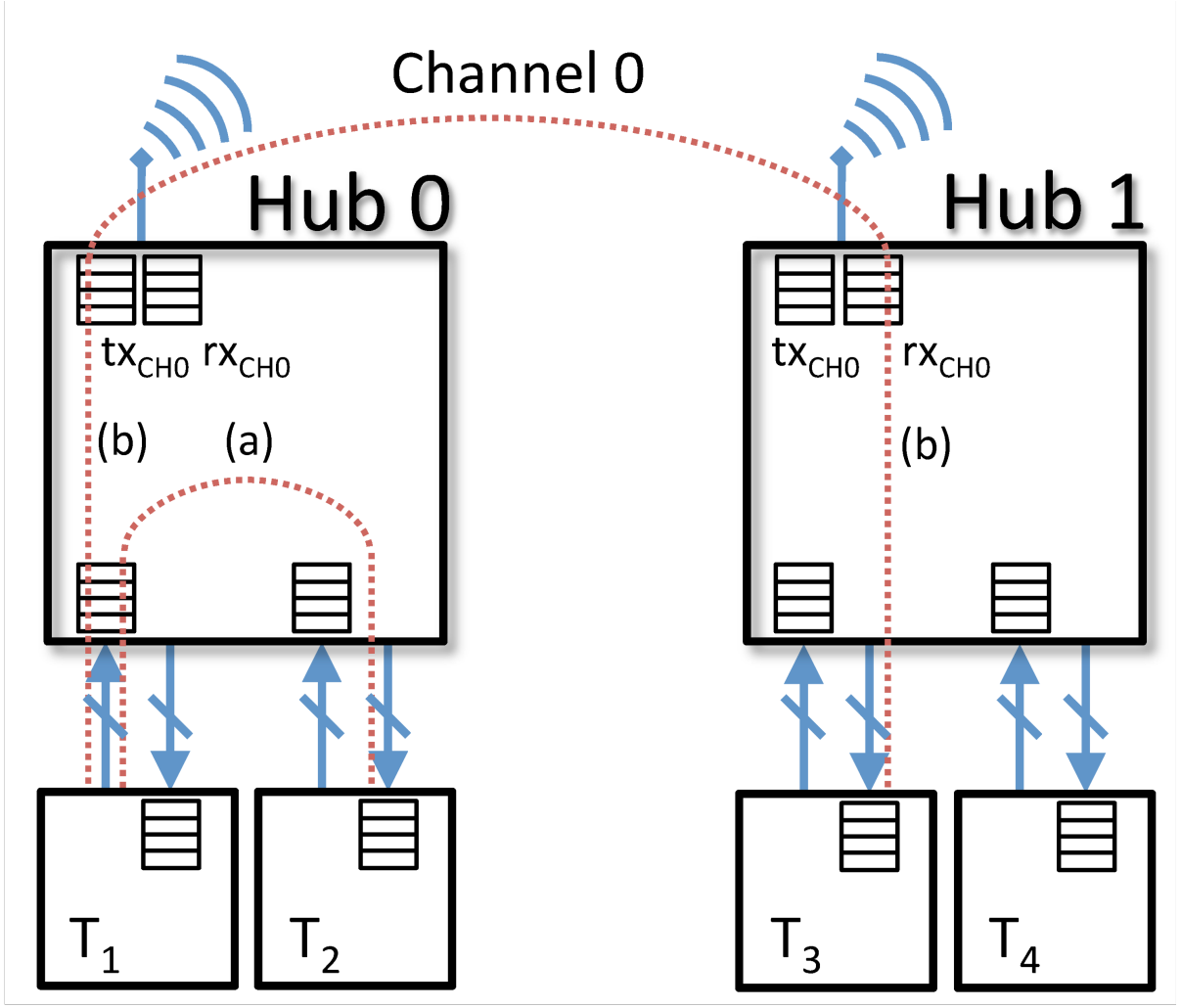
$$D = 2(\sqrt{n} - 1)$$

$$E_{flit} = n(E_{switch} + E_{link})$$

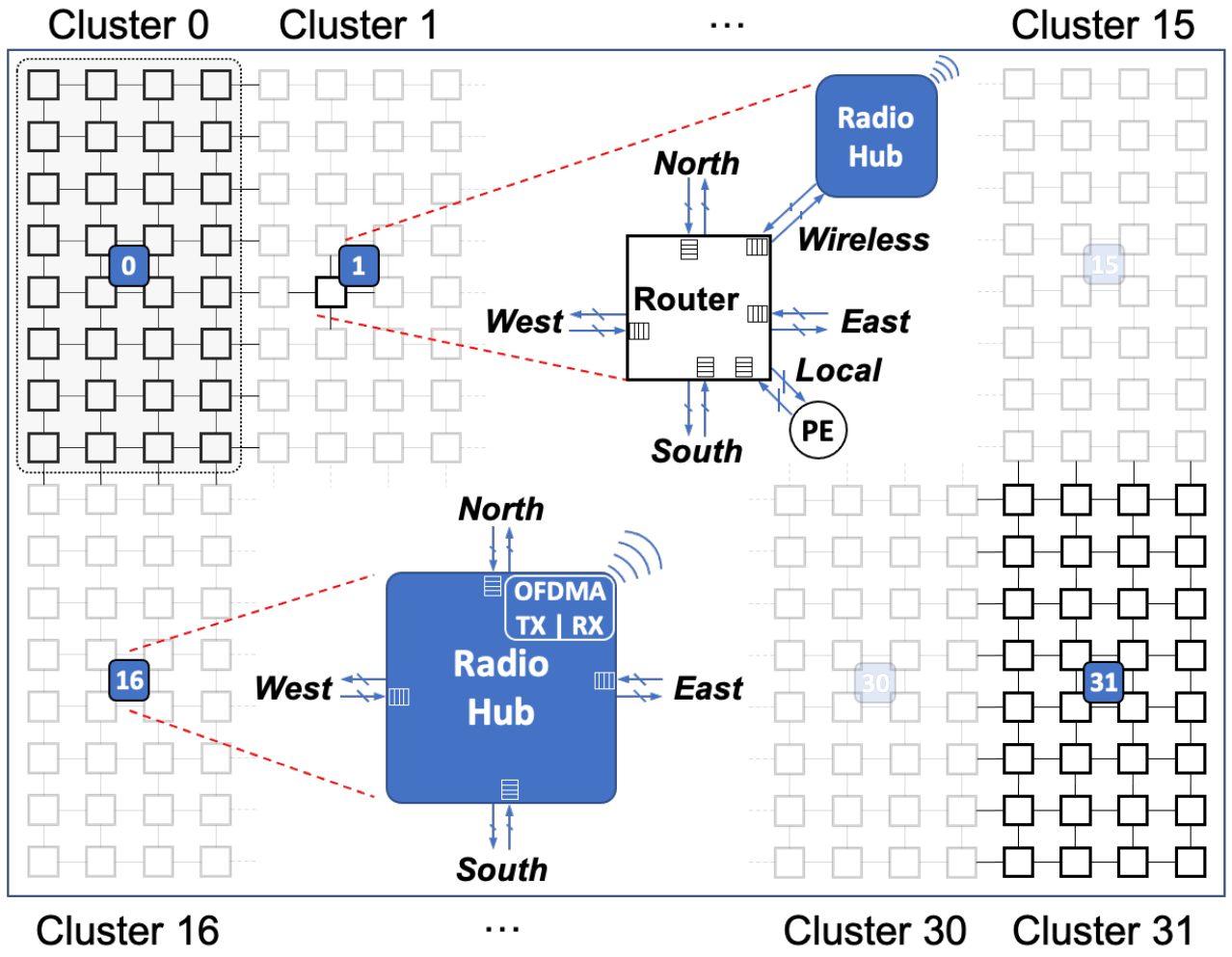


Wireless NoC elements

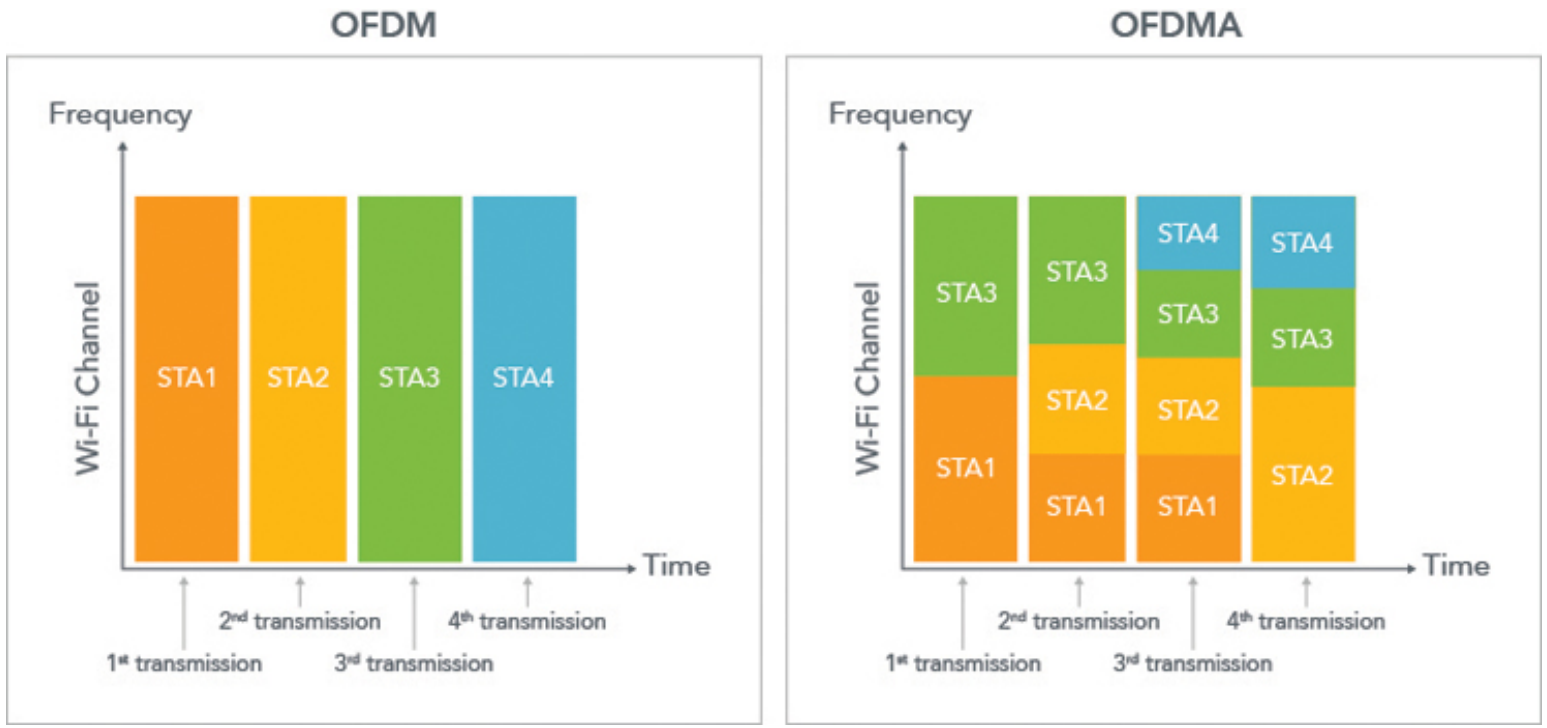




RF-NoC Architecture

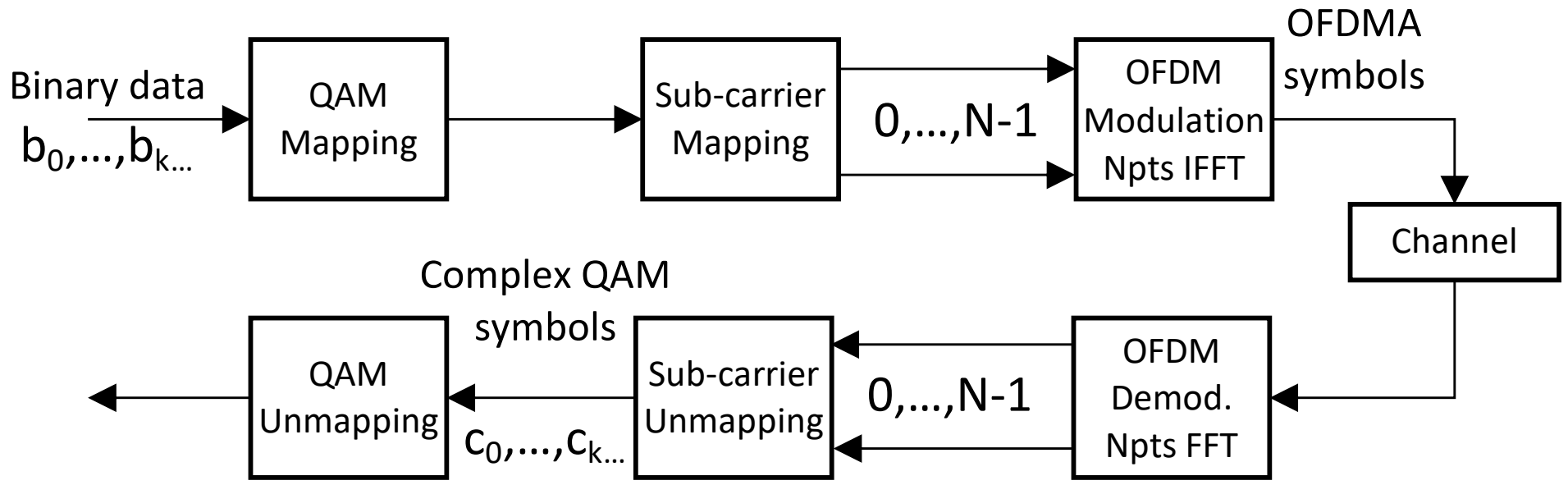


RF-NoC Architecture: OFDMA



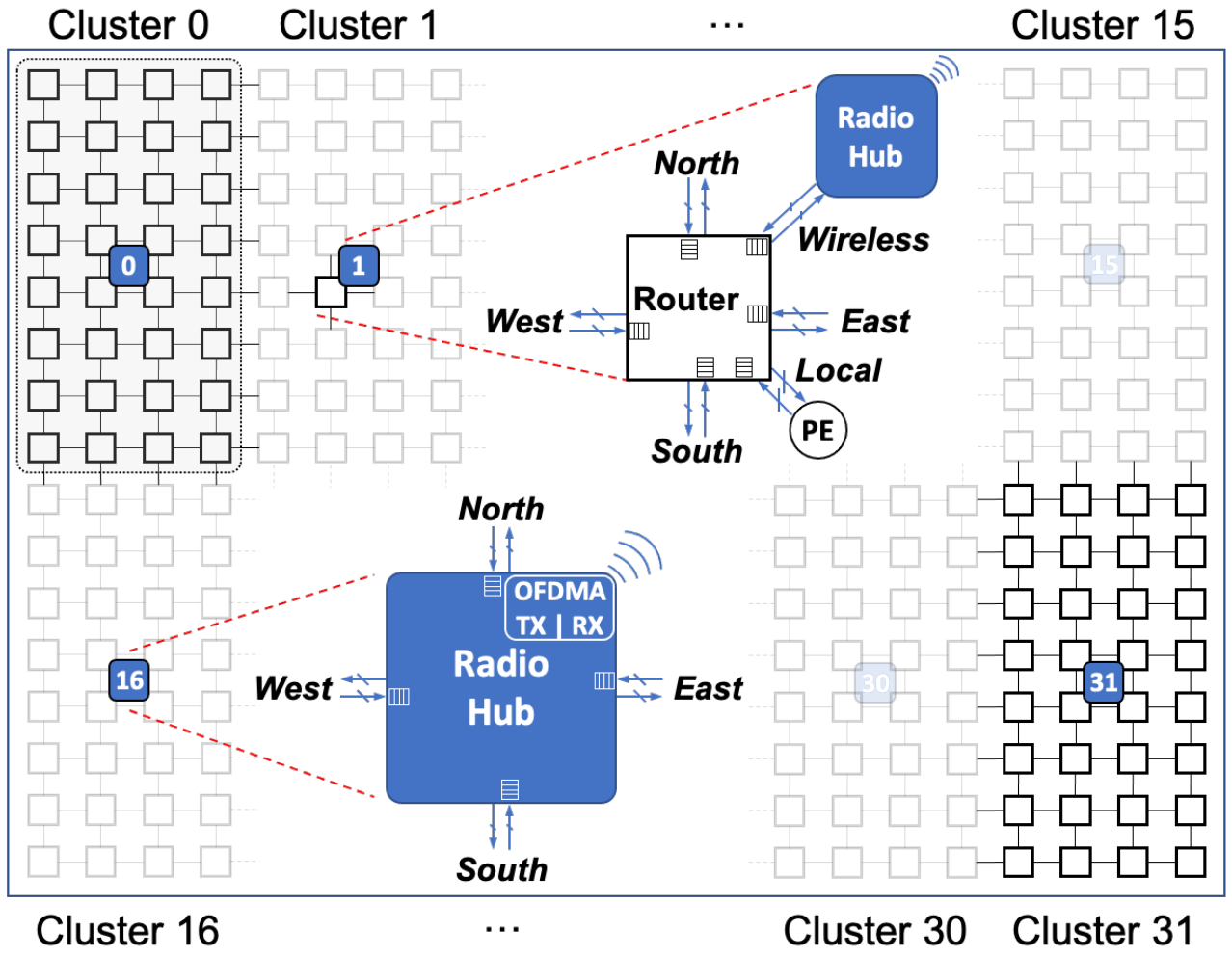
Source: https://www.rfglobalnet.com/doc/wi-fi-s-ofdma-challenges-make-verification-crucial-0001

RF-NoC Architecture: OFDMA



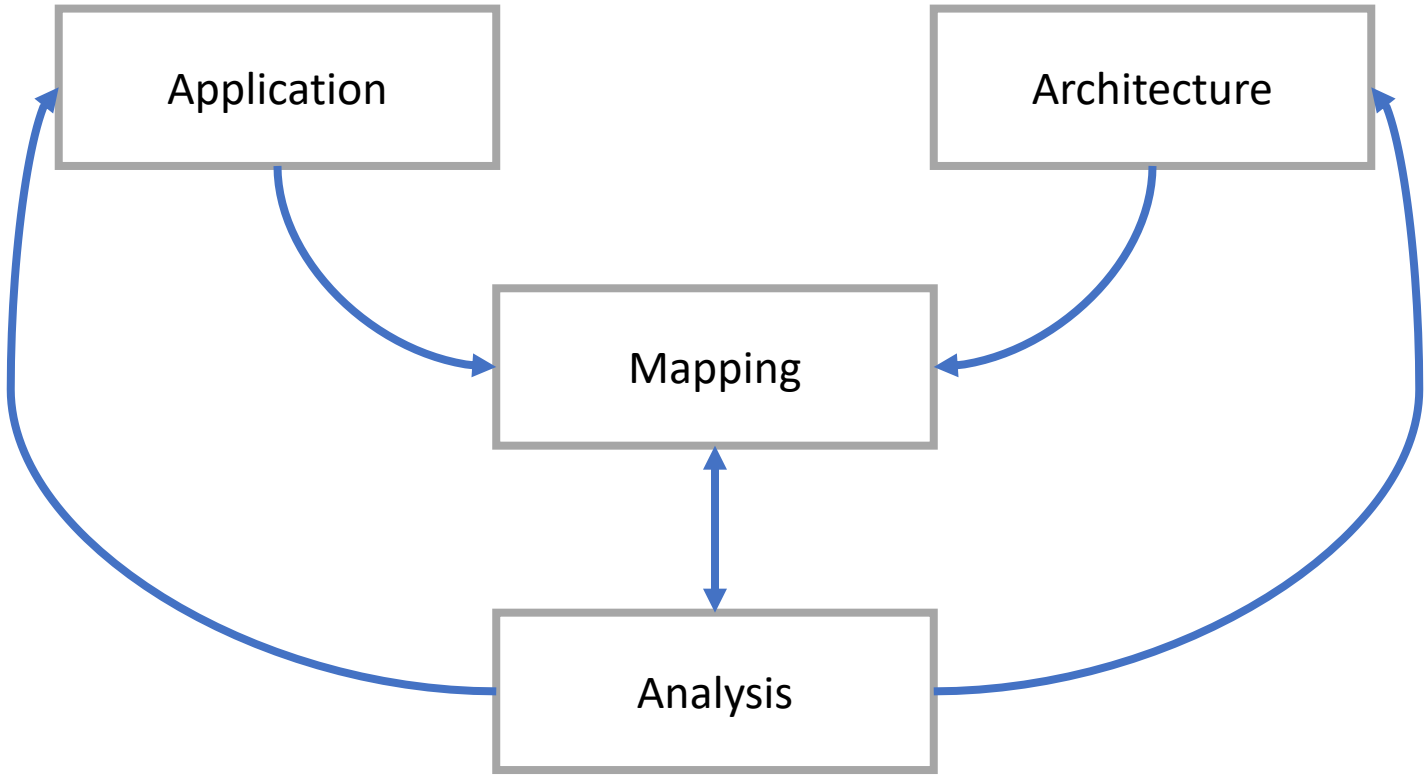
Orthogonal Frequency-Division Multiple Access (OFDMA) block diagram

RF-NoC Architecture

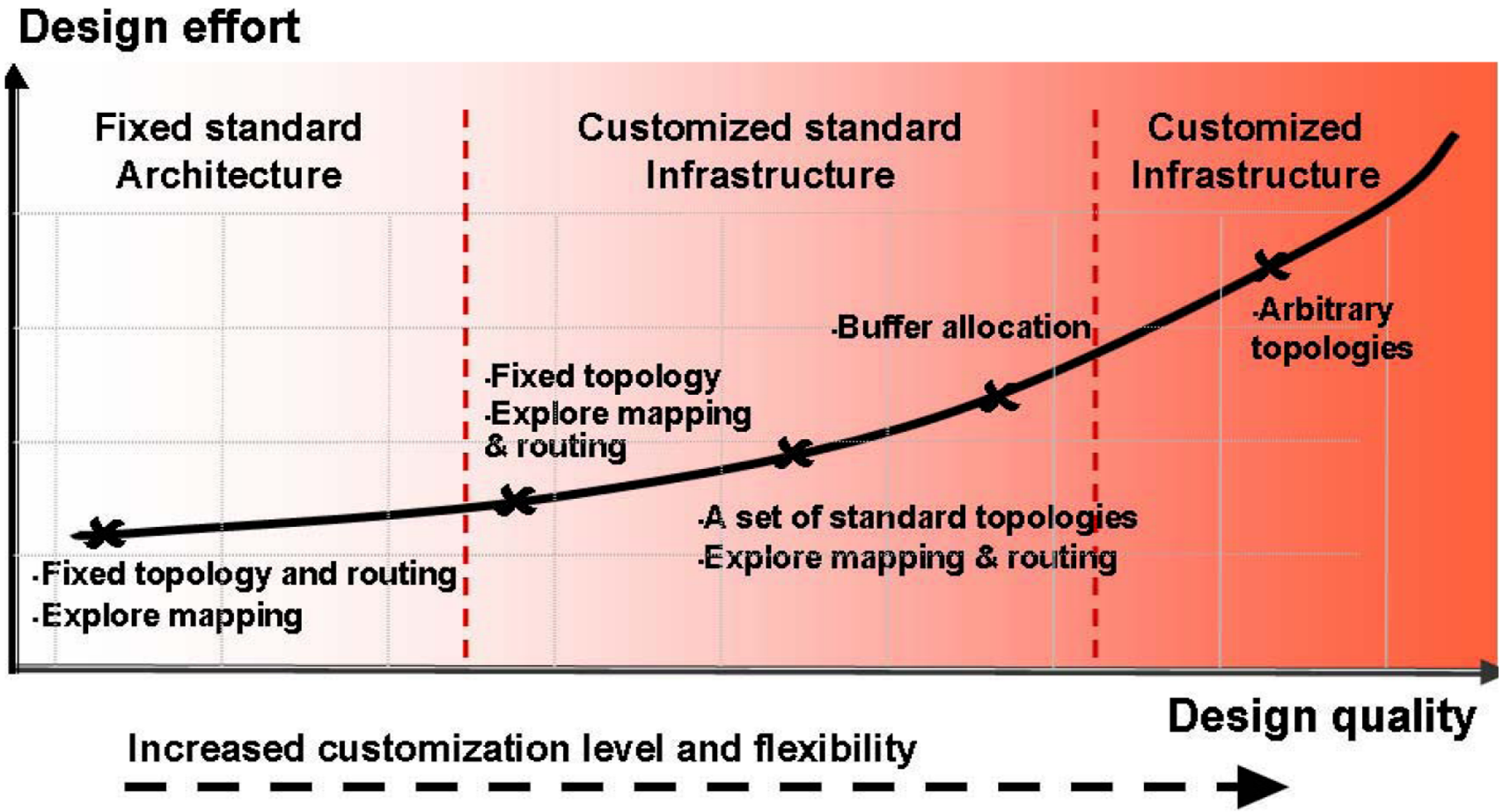


Design and Simulation

Design Space Exploration

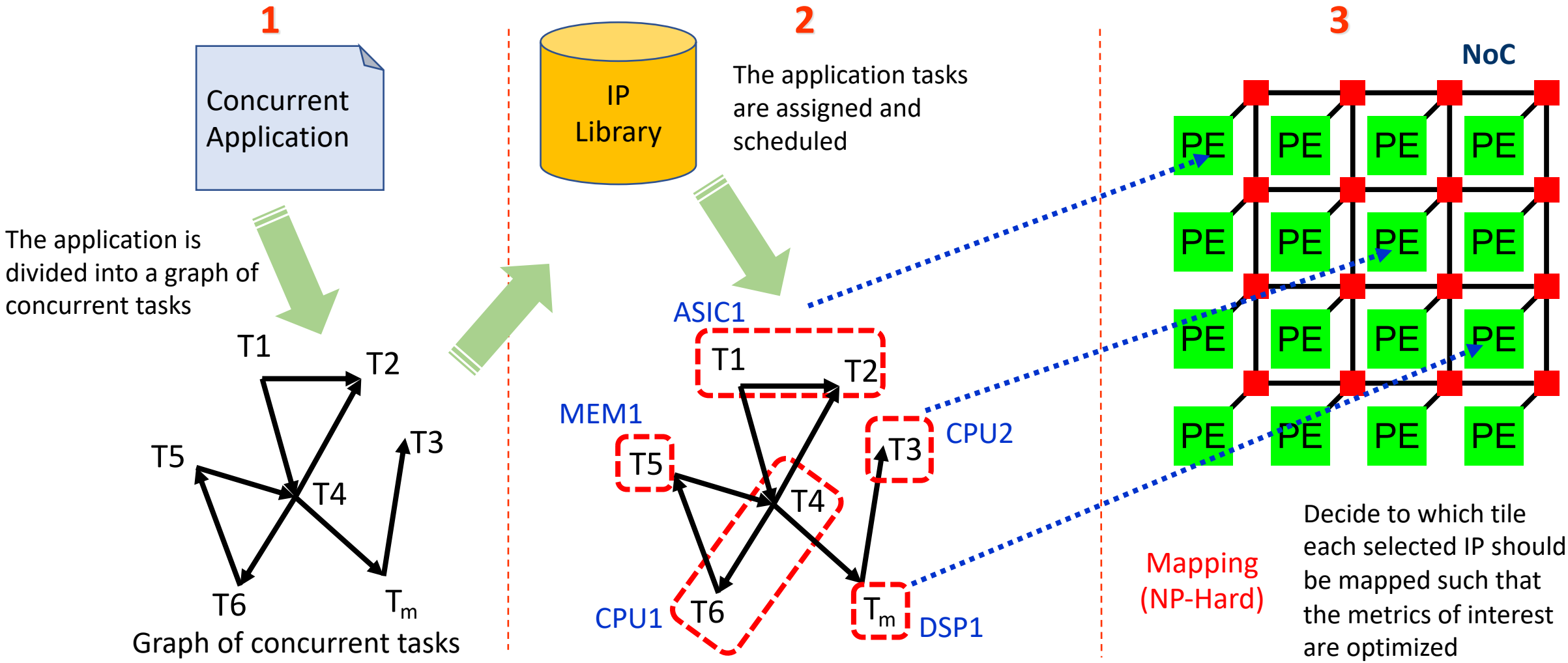


Design Space Exploration for NoC Architectures



Ogras *et al.*, ASAP'05

The Mapping Problem



Noxim: NoC Simulator

- Cycle accurate, Open Source:
  - https://github.com/davidepatti/noxim
- SystemC signals level simulation in C++
- Performance & Power estimation
- Modular plugin-like addition of Routing/Selection strategies
- Wireless transmissions simulation

Catania *et al.*, ACM TOMACS 2016

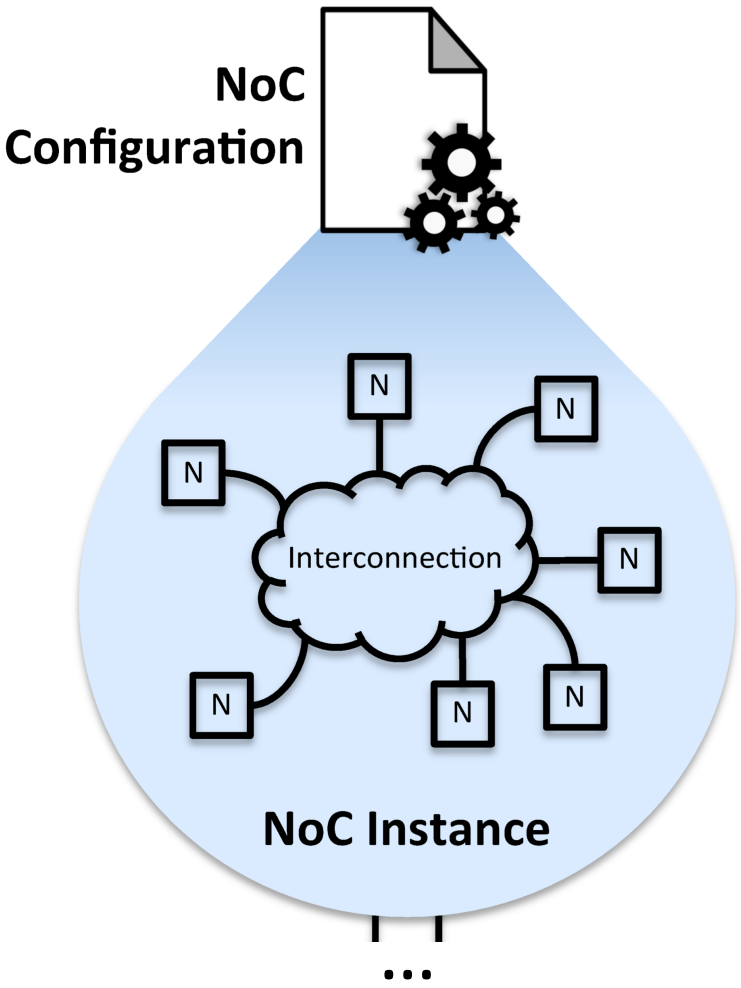
Noxim: Simulation Flow

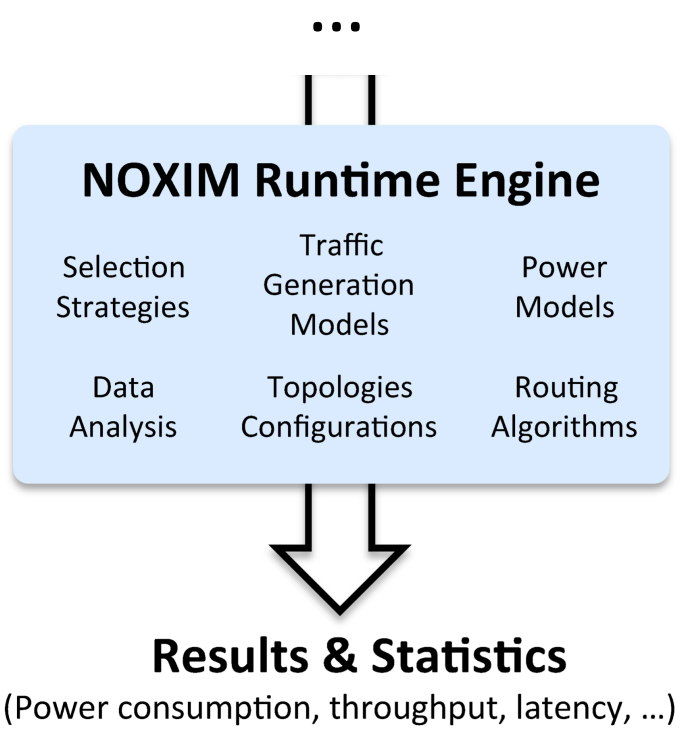
Topology & Structure
- size (number of nodes)
- Buffers
- wireless channels

Workload
- packet size
- injection rate
- traffic distribution

Dynamic behaviour
- routing strategy
- wireless access

Simulation Runtime
- duration (cycles)
- statistics collection

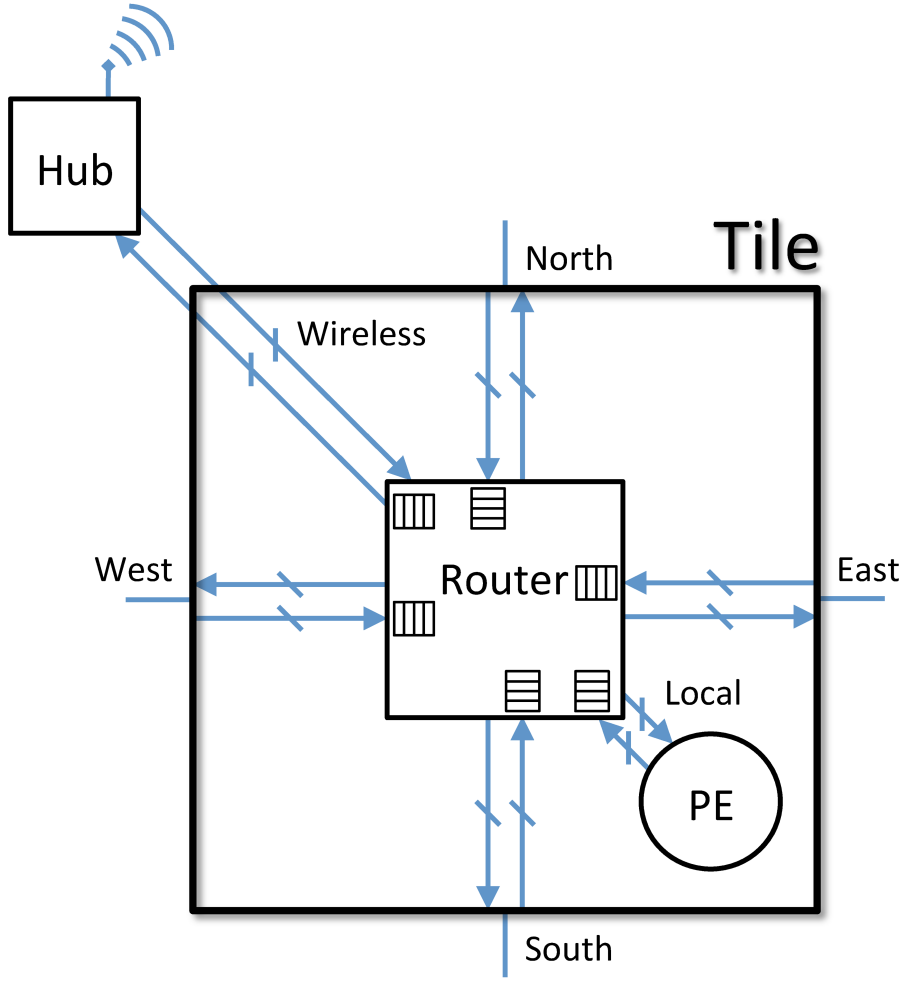




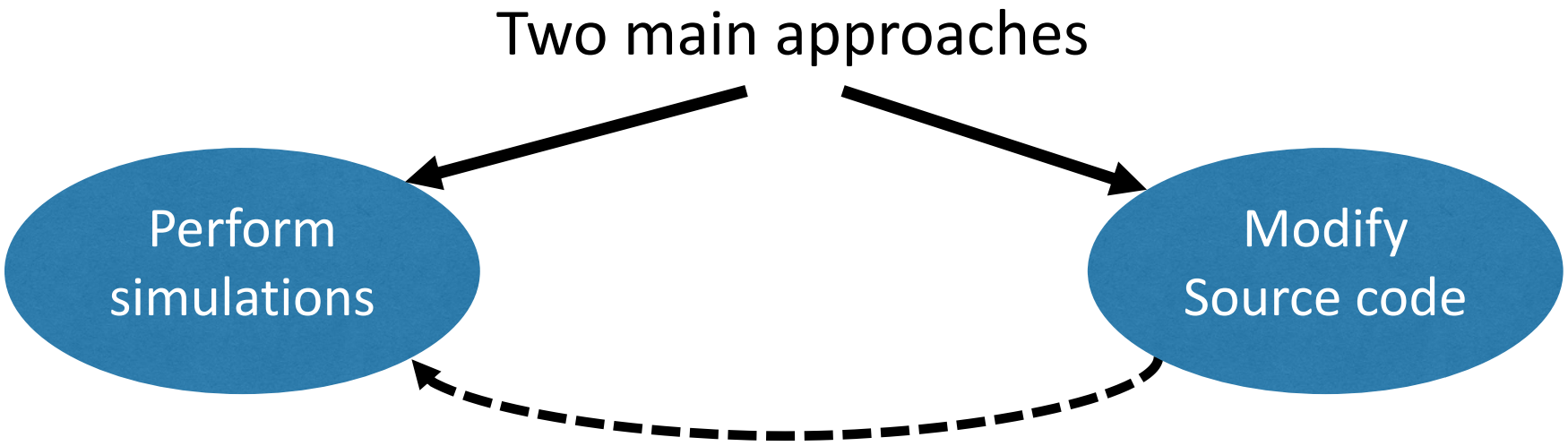
Noxim: Default Scenario

Mesh of Tile nodes

- Each Tile, as aforementioned, contains a Router and a Processing Element (PE)
- Each Tile is connected to the 4 neighbors (N, S, E, W)
- Optionally, some nodes can be connected to Radio-hub allowing wireless transmissions.



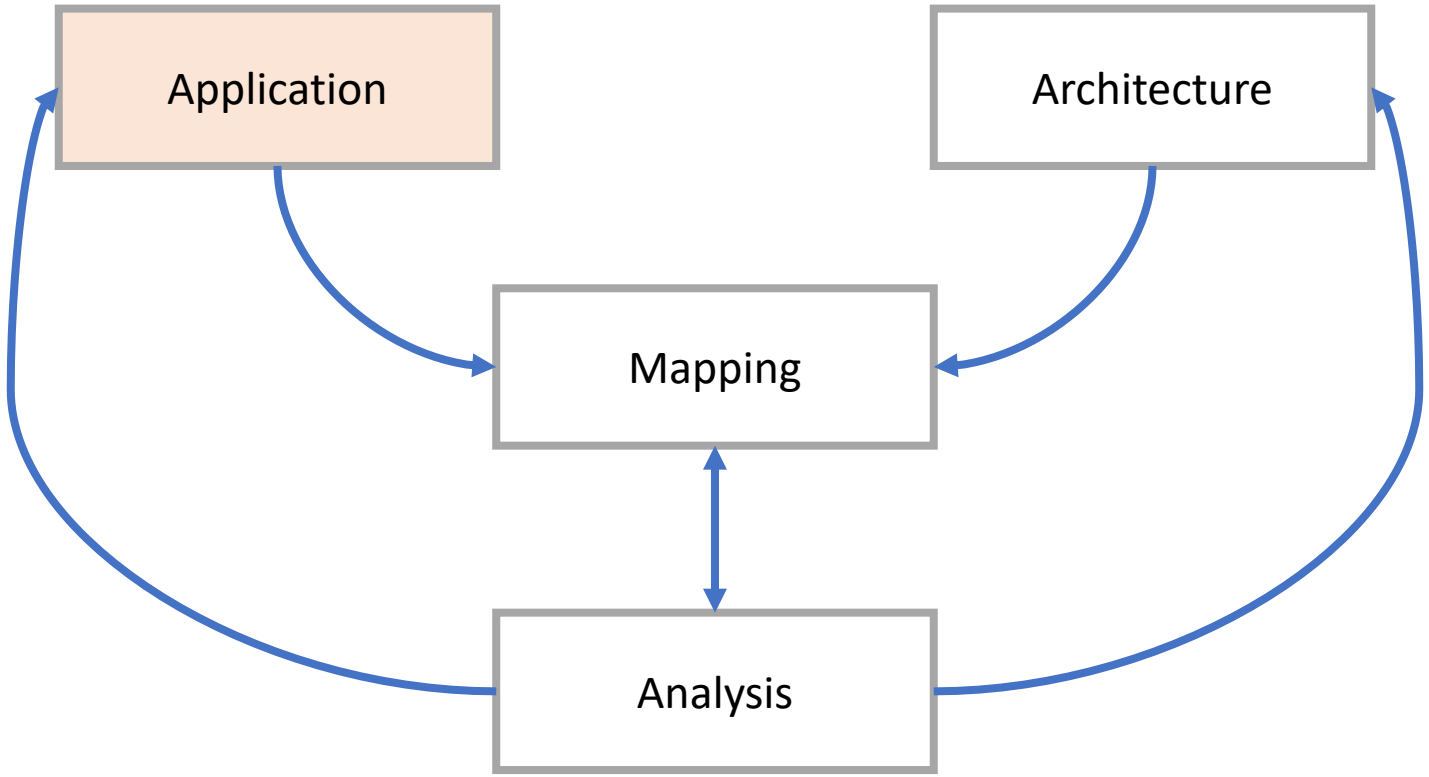
Noxim: What can be done?



- Design Space Exploration
- Effect of Traffic Patterns
- Power Estimation
- Comparison Analysis

- New Routing Algorithms
- Different topologies
- Power Saving Strategies
- And so on …

Design Space Exploration



Simulation Framework

- With Noxim it is possible to simulate application traffic starting from
  - Common traffic models
  - Input traffic table

It could be a limitation when specific real application traffic is needed

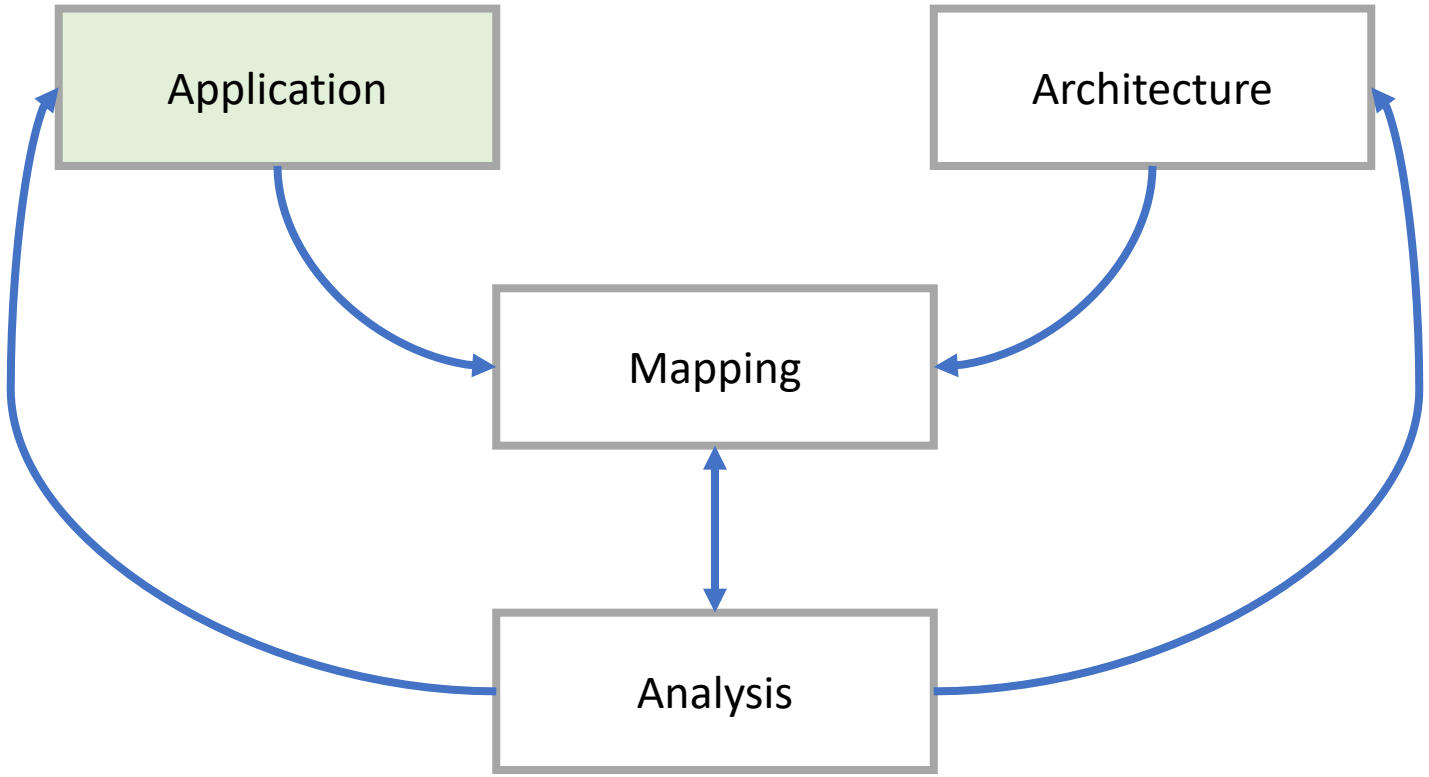
Simulation Framework: Sniper

For this purpose we get help from a different tool, a system-level simulator called Sniper.

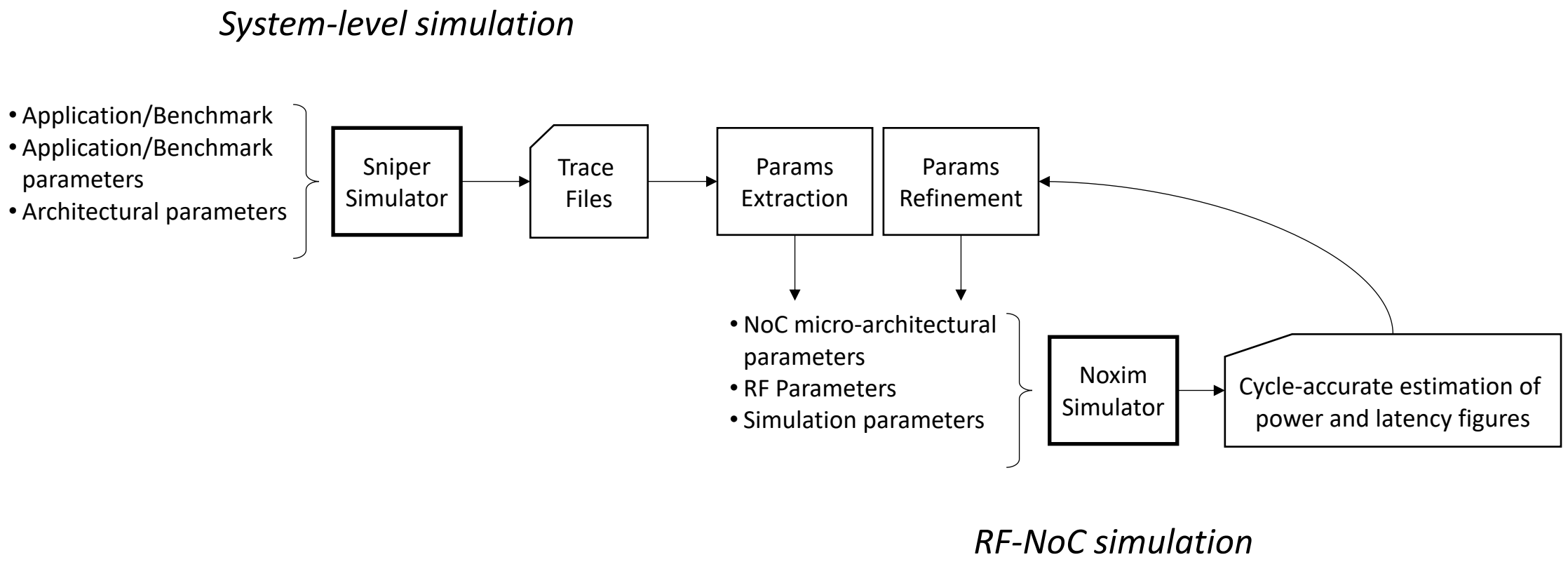
“Sniper is a next generation parallel, high-speed and accurate x86 simulator. This multi-core simulator is based on the interval core model and the Graphite simulation infrastructure, allowing for fast and accurate simulation and for trading off simulation speed for accuracy to allow a range of flexible simulation options when exploring different homogeneous and heterogeneous multi-core architectures.”

Source: http://snipersim.org/w/The_Sniper_Multi-Core_Simulator

Design Space Exploration



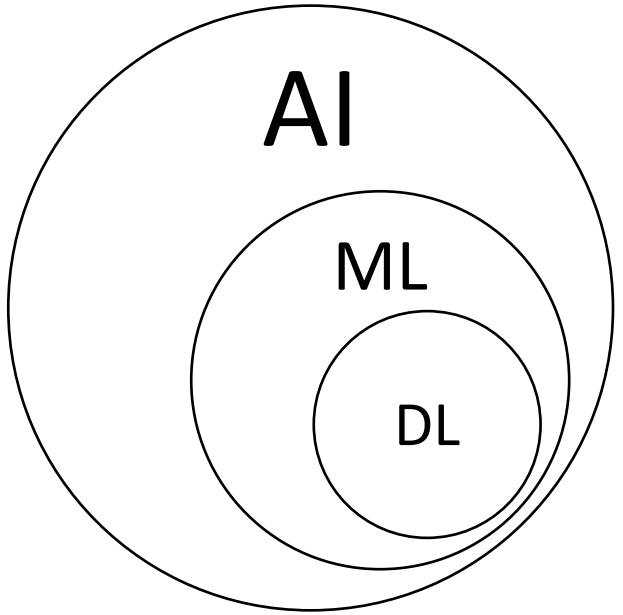
Simulation Framework



Applications

Application Development Trends

- Spread of applications exploiting Artificial Intelligence (AI)
  - In particular Machine Learning (ML) applications
    - Through Deep Learning (DL) techniques



Application Development Trends

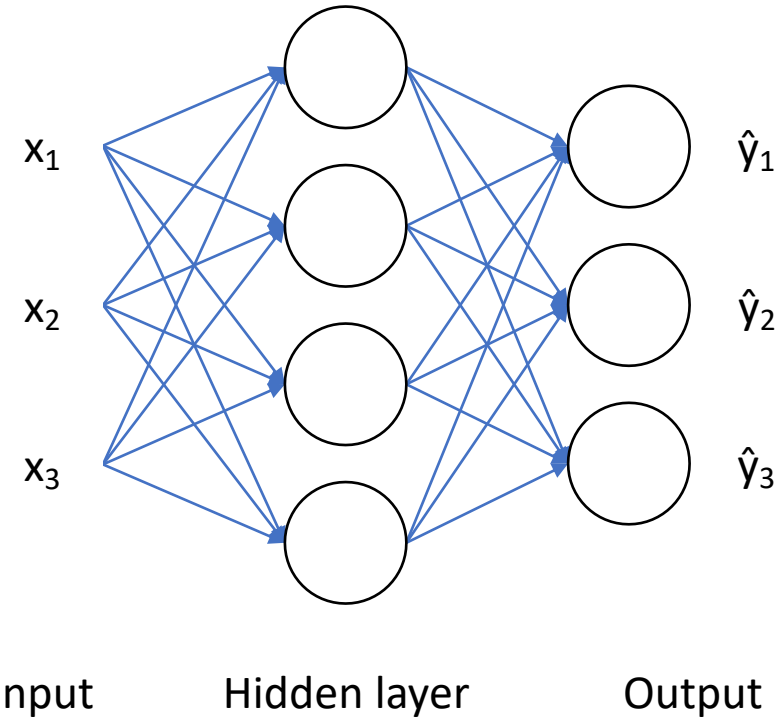
Some examples using Neural Networks:

- Visual data processing
- Speech and audio processing
- Autonomous driving

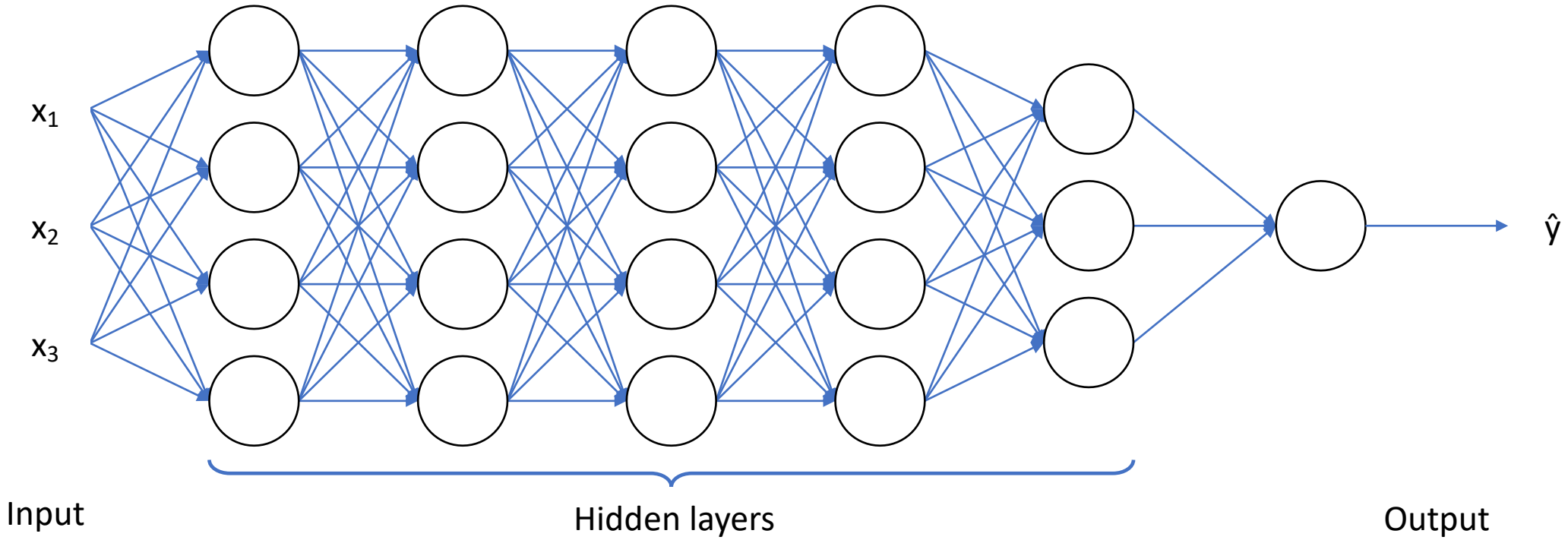
Again two main domains of interest:

- High-Performance Computing
- Resource constrained devices

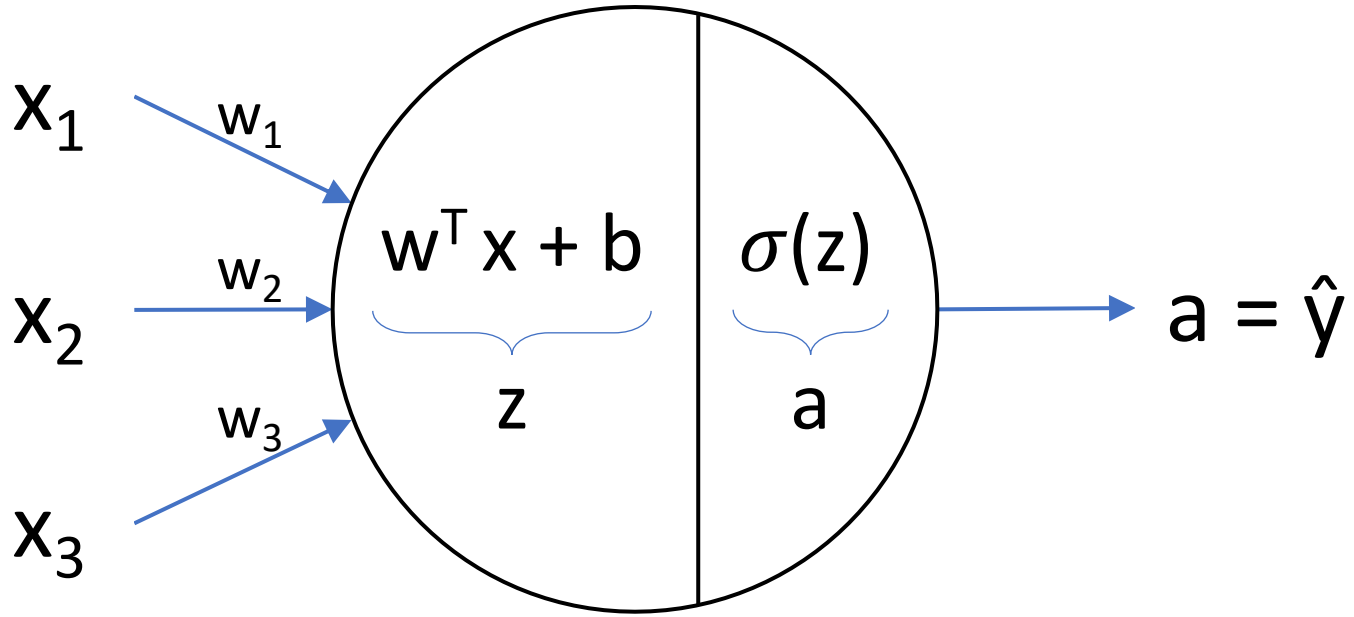
Neural Networks



Deep Neural Networks (DNNs)



Neural Networks: Perceptron Model

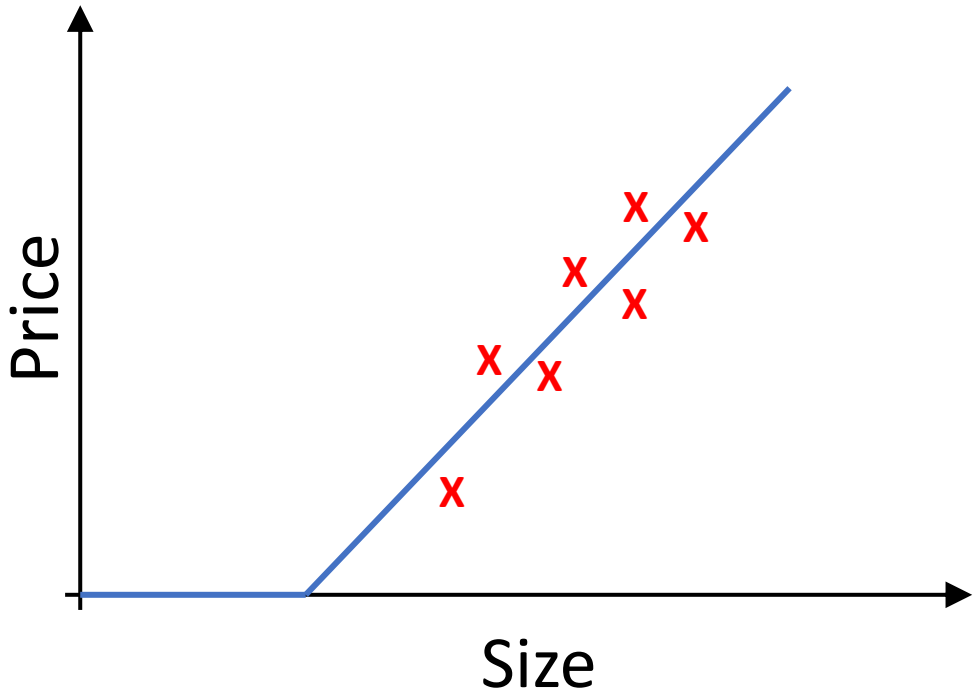


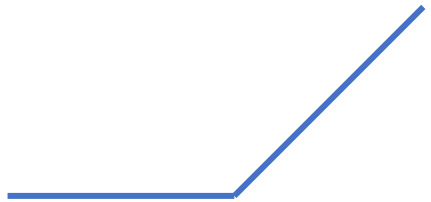
$$\mathrm{z} = w^T x + b = [w_1 \quad w_2 \quad w_3] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + b$$

$$\mathrm{a} = \sigma(z)$$

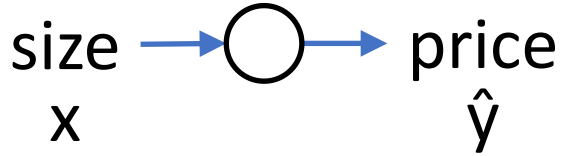
Source: Deep Learning Specialization - Prof. Andrew Ng

Neural Networks: House Price Prediction



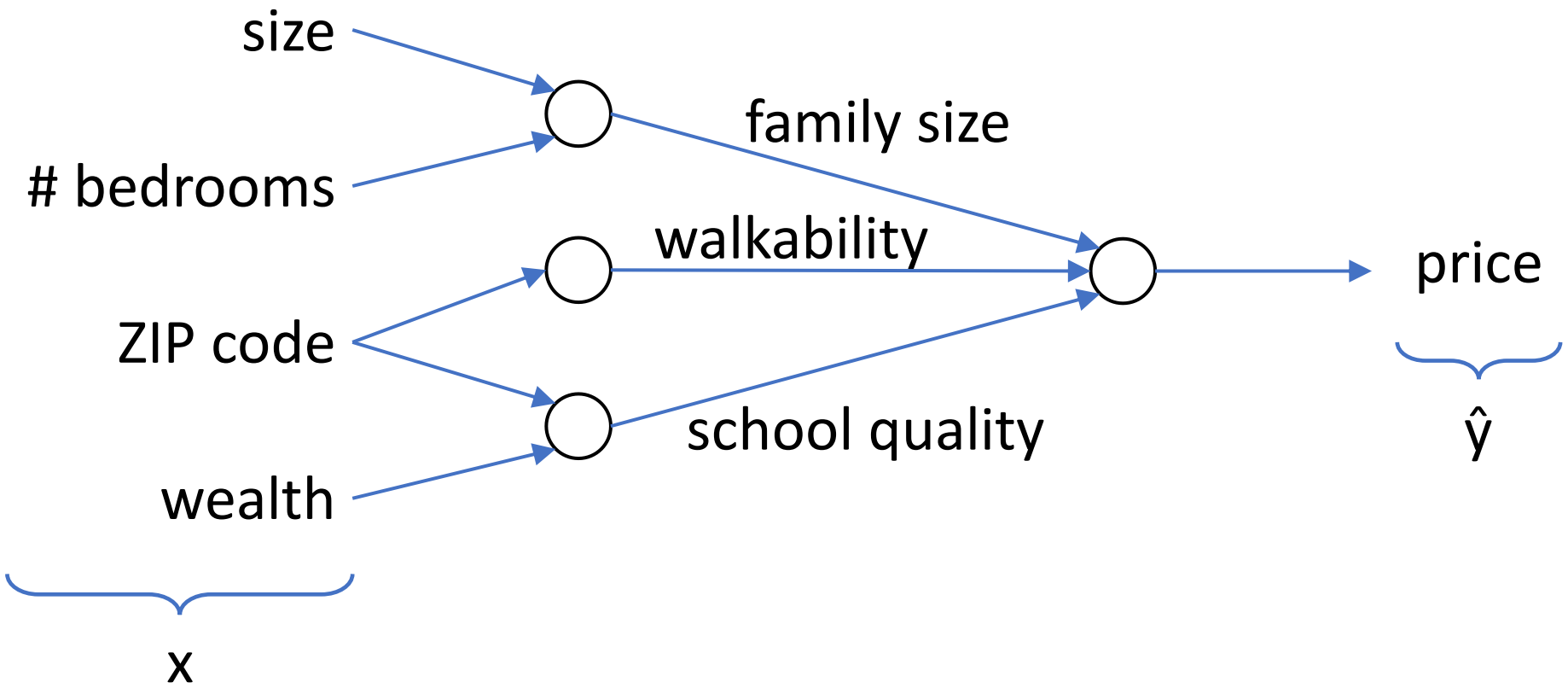


ReLU:
Rectified
Linear
Unit



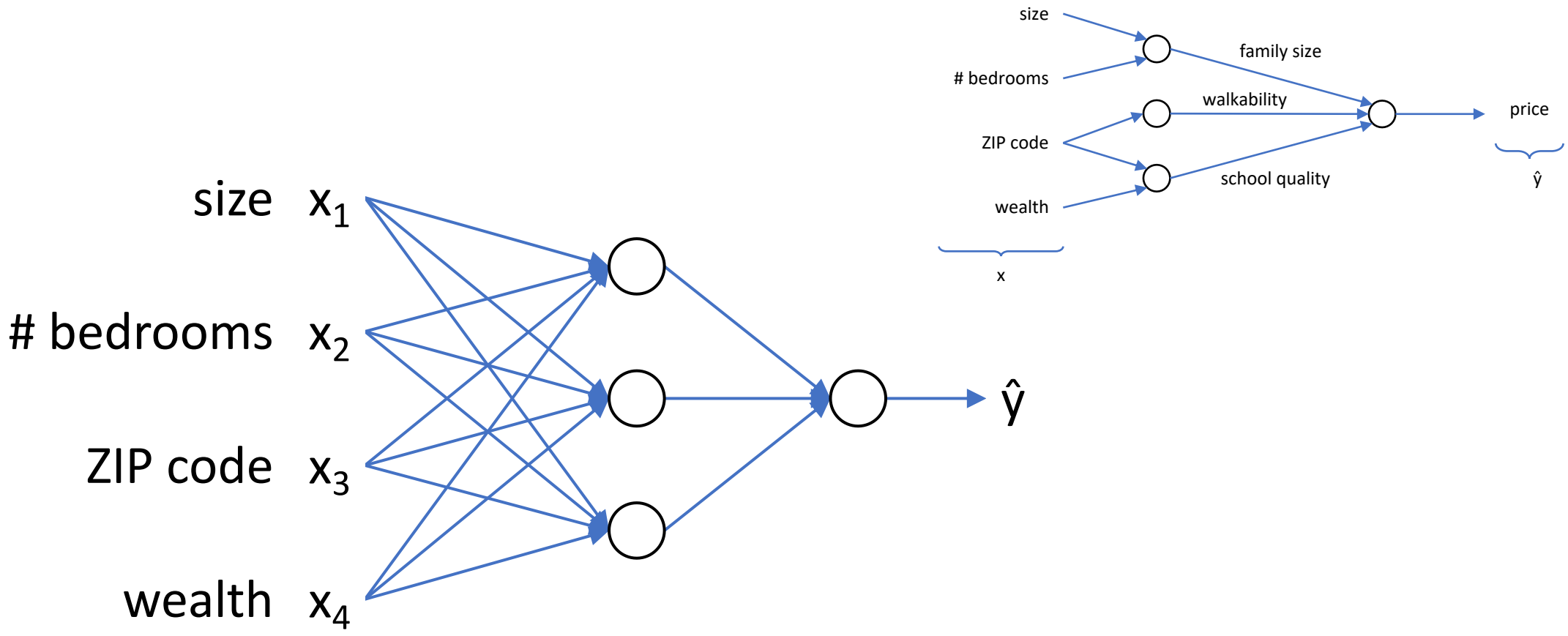
Source: Deep Learning Specialization - Prof. Andrew Ng

Neural Networks: House Price Prediction



Source: Deep Learning Specialization - Prof. Andrew Ng

Neural Networks: House Price Prediction



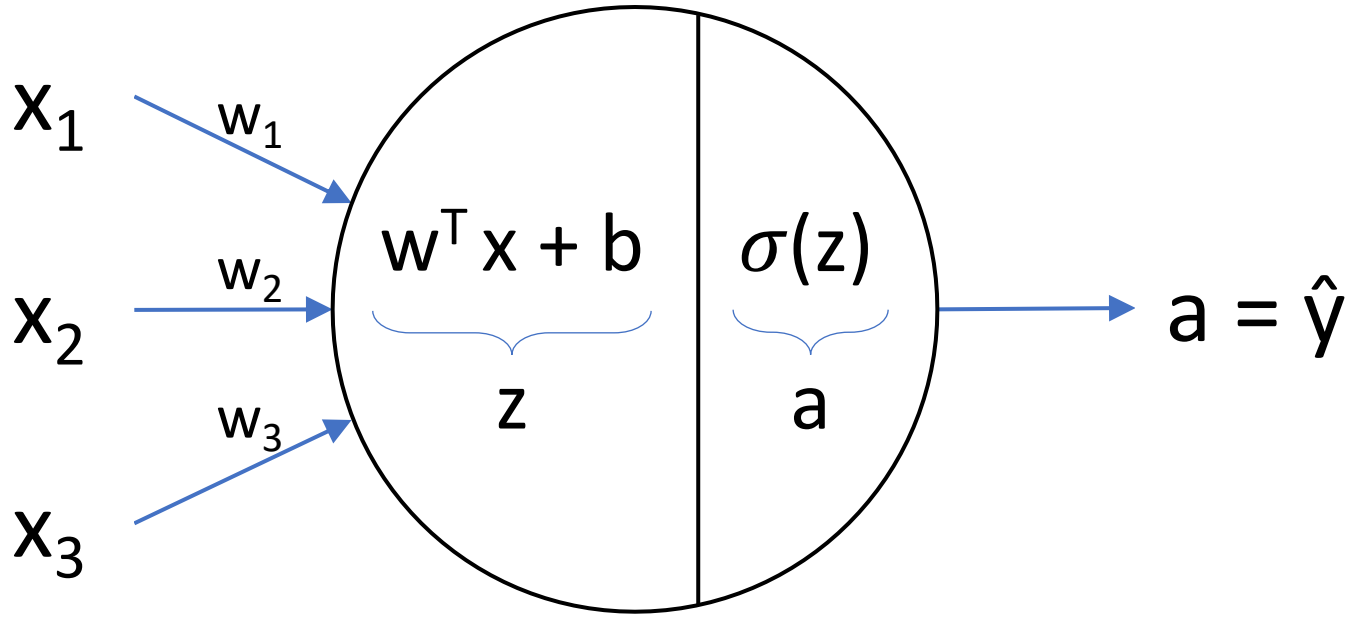
Source: Deep Learning Specialization - Prof. Andrew Ng

Neural Networks: some applications

Input(x)	Output (y)	Application
Home features	Price	Real Estate
Ad, user info	Click on ad? (0/1)	Online Advertising
Image	Object (1,…,1000)	Photo tagging
Audio	Text transcript	Speech recognition
English	Chinese	Machine translation
Image, Radar info	Position of other cars	Autonomous driving

Source: Deep Learning Specialization - Prof. Andrew Ng

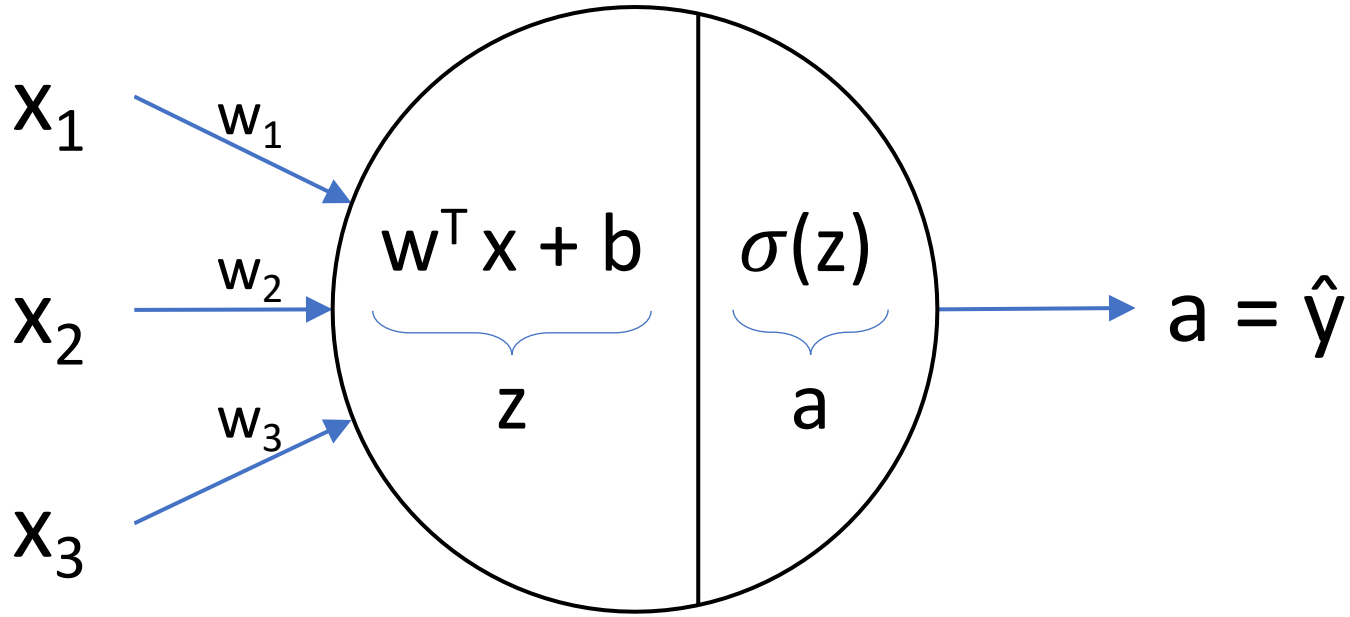
Neural Networks: Perceptron Model



$$\mathrm{z} = w^T x + b = [w_1 \quad w_2 \quad w_3] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + b$$

$$\mathrm{a} = \sigma(z)$$

Neural Networks: Perceptron Model



multiply and accumulate (MAC)

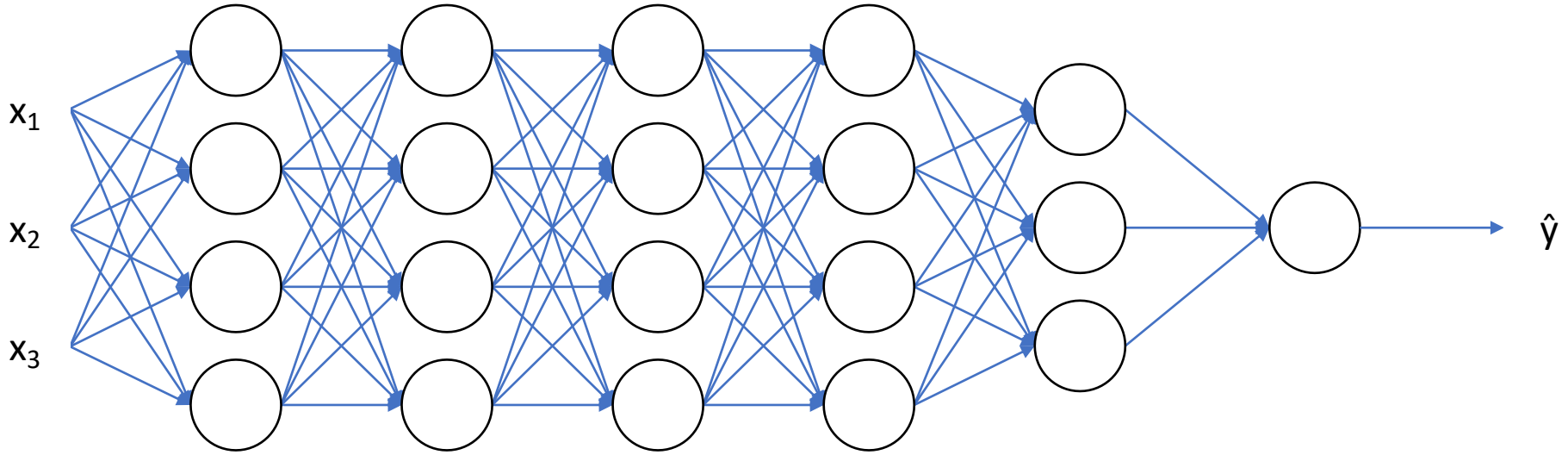
$$z = w^T x + b = w_1 x_1 + w_2 x_2 + w_3 x_3 + b$$

$$a = \sigma(z)$$

Neural Networks: How to use them

Two modes:

1. Training
2. Inference



Popular DNN Models

Metrics	LeNet-5	AlexNet	VGG-16	GoogLeNet (v1)	ResNet-50	EfficientNet-B4
Top-5 error (ImageNet)	n/a	16.4	7.4	6.7	5.3	3.7*
Input Size	28x28	227x227	224x224	224x224	224x224	380x380
# of CONV Layers	**2**	**5**	**16**	**21 (depth)**	**49**	**96**
# of Weights	2.6k	2.3M	14.7M	6.0M	23.5M	14M
# of MACs	283k	666M	15.3G	1.43G	3.86G	4.4G
# of FC layers	**2**	**3**	**3**	**1**	**1**	**65****
# of Weights	58k	58.6M	124M	1M	2M	4.9M
# of MACs	58k	58.6M	124M	1M	2M	4.9M
Total Weights	**60k**	**61M**	**138M**	**7M**	**25.5M**	**19M**
Total MACs	**341k**	**724M**	**15.5G**	**1.43G**	**3.9G**	**4.4G**
Reference	**Lecun**, *PIEEE* 1998	**Krizhevsky**, *NeurIPS* 2012	**Simonyan**, *ICLR* 2015	**Szegedy**, *CVPR* 2015	**He**, *CVPR* 2016	**Tan**, *ICML* 2019

* Does not include multi-crop and ensemble

** Increase in FC layers due to squeeze-and-excitation layers (much smaller than FC layers for classification)

Source: How to Evaluate Efficient Deep Neural Network Approaches - Prof. V. Sze, MIT

Key metrics for an implementation

- Accuracy
- Throughput
- Latency
- Energy and Power consumption
- HW cost
- Flexibility
- Scalability

Source: How to Evaluate Efficient Deep Neural Network Approaches - Prof. V. Sze, MIT

Key design objectives for an implementation

- Increase Throughput and Reduce Latency
- Avoid unnecessary MACs (save cycles)
- Increase number of processing elements (PE)
- Increase PE utilization

Source: How to Evaluate Efficient Deep Neural Network Approaches - Prof. V. Sze, MIT

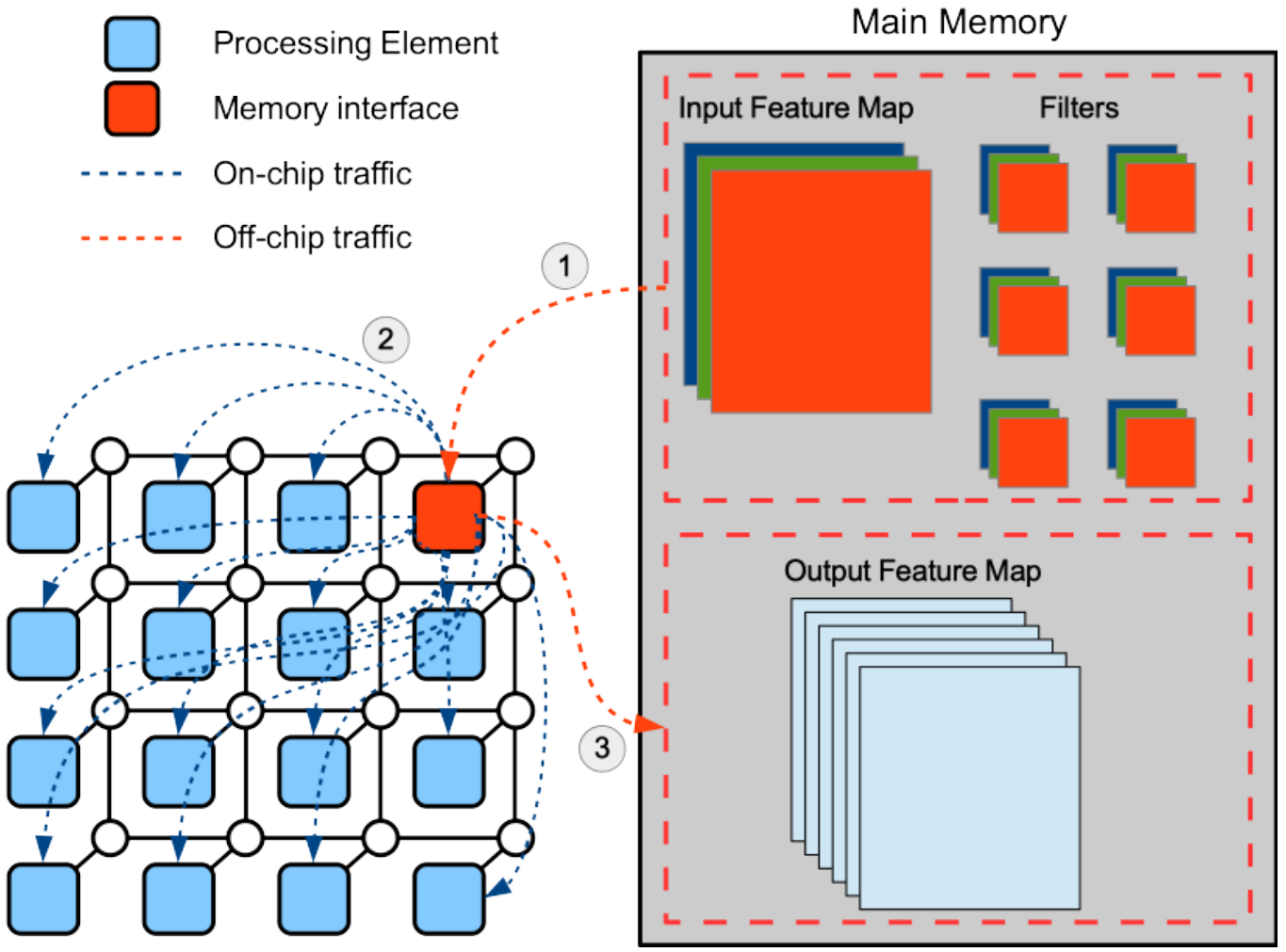
Key design objectives for an implementation

Inference latency and energy dominated by

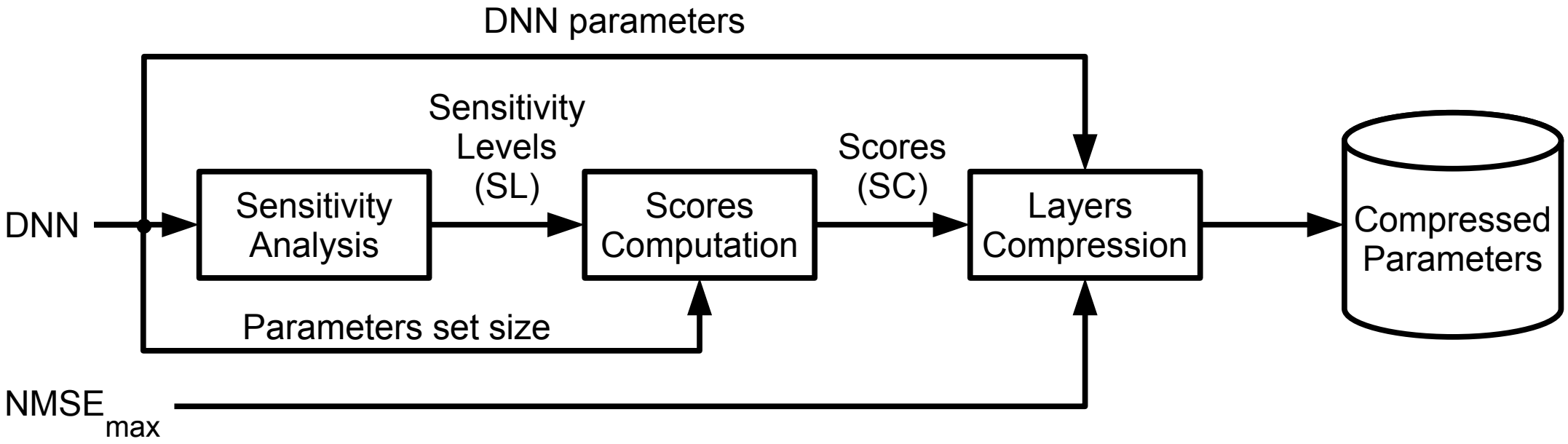
- Memory sub-system
- Communication sub-system

Memory traffic dominated by model parameters

Hardware implementation

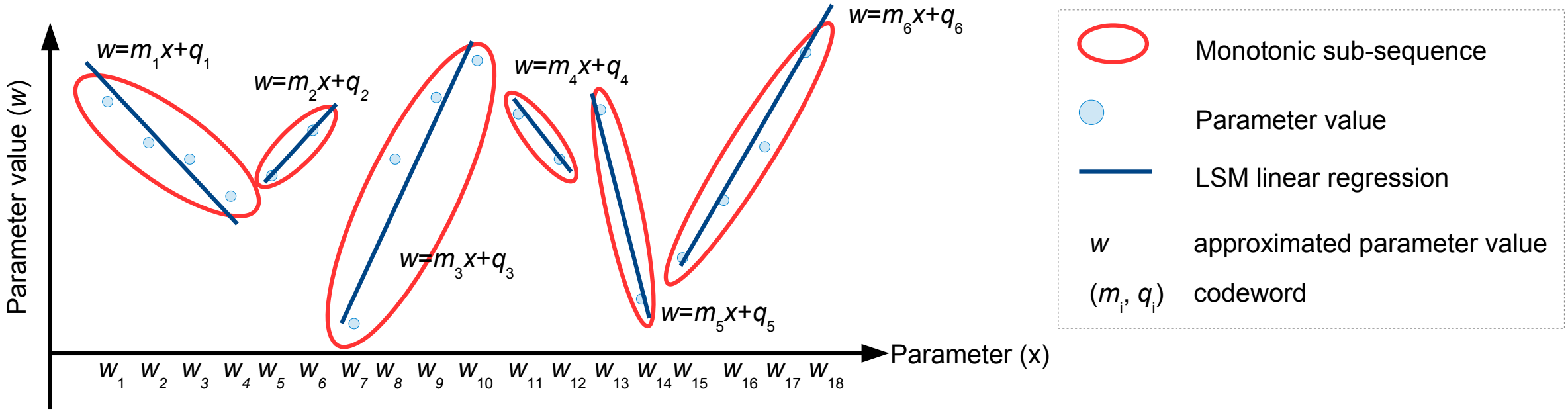


Hardware implementation: Weight Compression



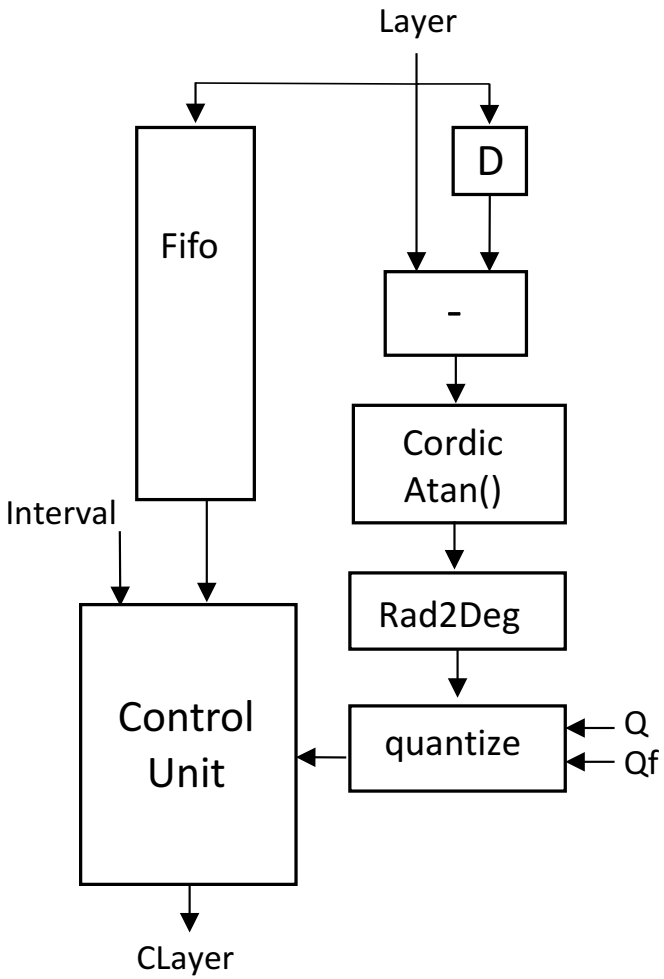
Selective Layers Compression Technique in Deep Neural Network Accelerators,
H. Lahdhiri et al., 2020

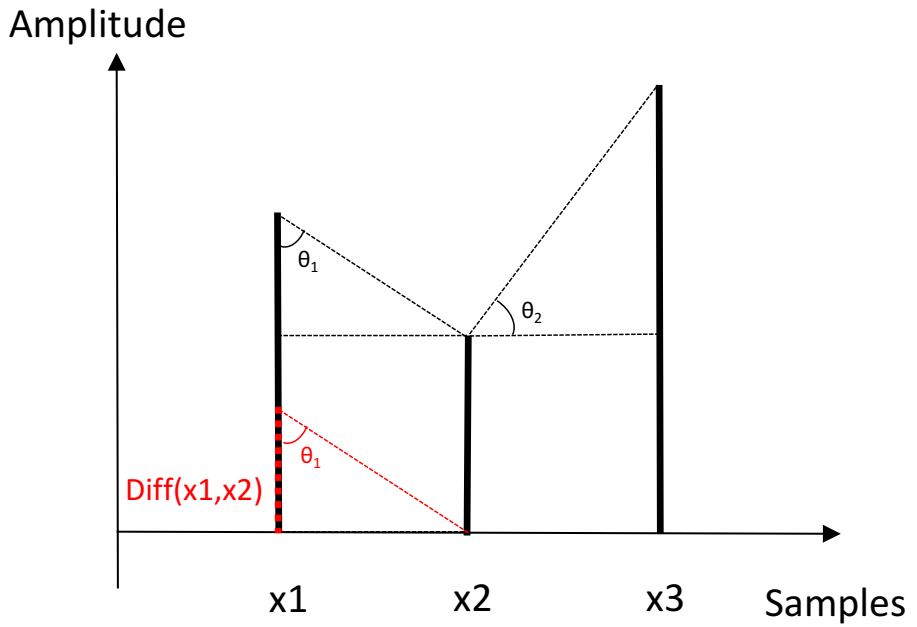
Hardware implementation: Weight Compression



Selective Layers Compression Technique in Deep Neural Network Accelerators,
H. Lahdhiri et al., 2020

Hardware implementation: Weight Compression





Efficient Compression Technique for NoC-based Deep Neural Network Accelerators,
J. Lorandel et al., 2020

People

Salvatore Monteleone, PhD
(formerly @ University of Catania)
salvatoremonteleone@gmail.com

Emmanuelle Bourdel, PhD
ETIS UMR 8051, CY Cergy-Paris Université, ENSEA, CNRS.

Jordane Lorandel, PhD
ETIS UMR 8051, CY Cergy-Paris Université, ENSEA, CNRS.

Habiba Lahdhiri, PhD Student
ETIS UMR 8051, CY Cergy-Paris Université, ENSEA, CNRS.

Maurizio Palesi, PhD
University of Catania

Davide Patti, PhD
University of Catania