# Efficient AI Modeling

John Li, Jacob Lin, and Ray Kuo
Digital Center and AI Center

創新 Innovation

品質 Quality

虛心 Open Mind

力行 Execution

Inventec

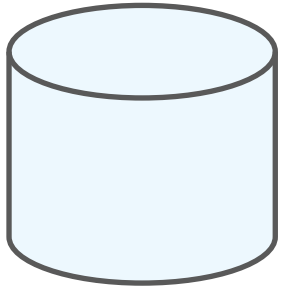
# Outline

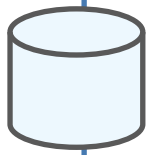
- Overview
- AI on Chip (AIoC) team work flow
- Quantization
- Testing System
- Hardware design
  - NPU
  - ASIC (Application Specific Integrated Circuit)
- System-hardware co-design

# Outline

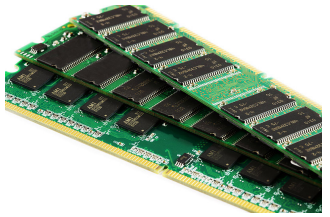
- **Overview**
- AI on Chip (AIoC) team work flow
- Quantization
- Testing System
- Hardware design
  - NPU
  - ASIC (Application Specific Integrated Circuit)
- System-hardware co-design

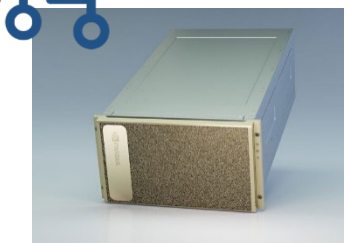
# Efficient AI

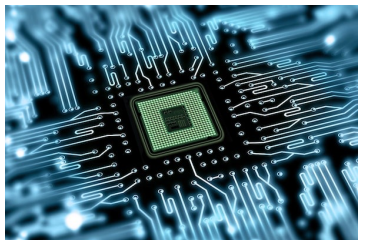


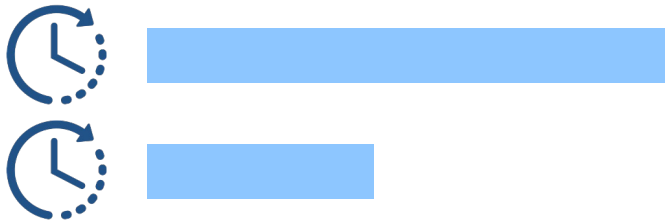


| | Training | Inference |
|---|---|---|
| | Data efficiency | Efficient architectures |
| | Training efficiency | Inference efficiency |







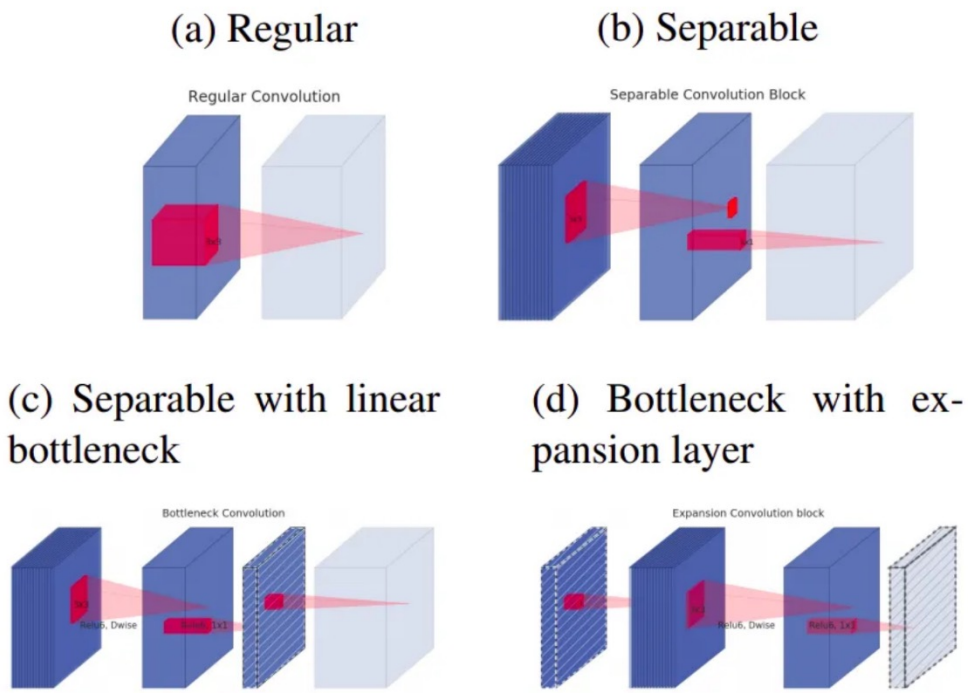


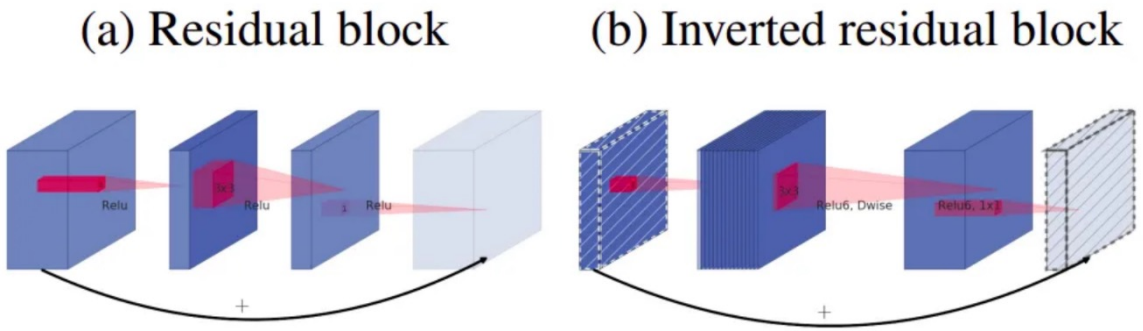
Slide credit: John

# Model Design

## MobilenetV2 and MobilenetV3

- V2: Bottleneck Residual Block
- V3: Squeeze-and-Excitation, NAS and NetAdapt



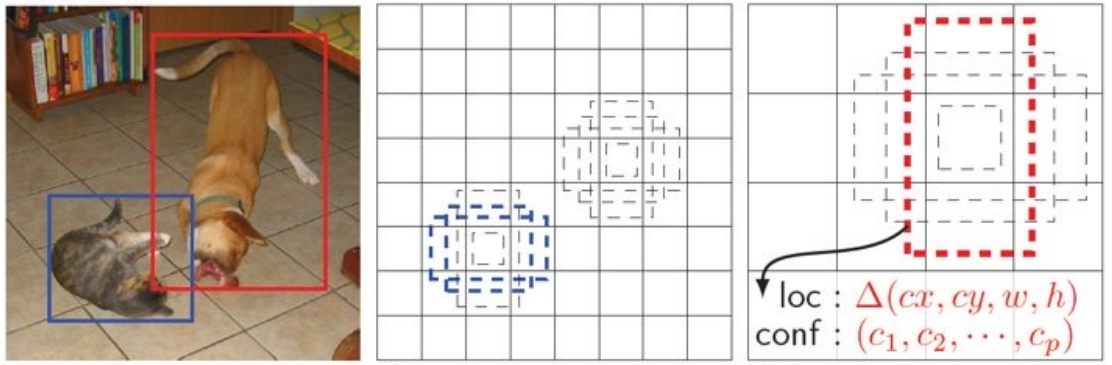

Sandler, Mark, et al. "Mobilenetv2: Inverted residuals and linear bottlenecks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

Slide credit: John

# Model Design

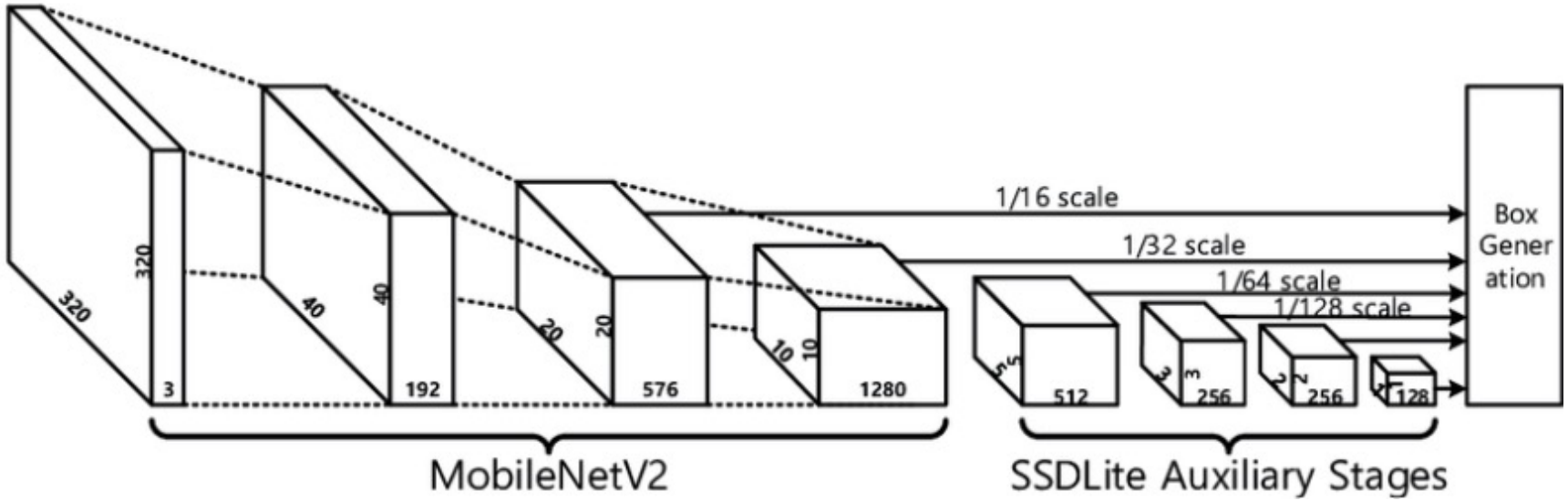
## Face detection

- Network: MobileNet + SSD (Single Shot Detector)




(a) Image with GT boxes (b) $8 \times 8$ feature map (c) $4 \times 4$ feature map

Prior boxes




https://github.com/Linzaer/Ultra-Light-Fast-Generic-Face-Detector-1MB

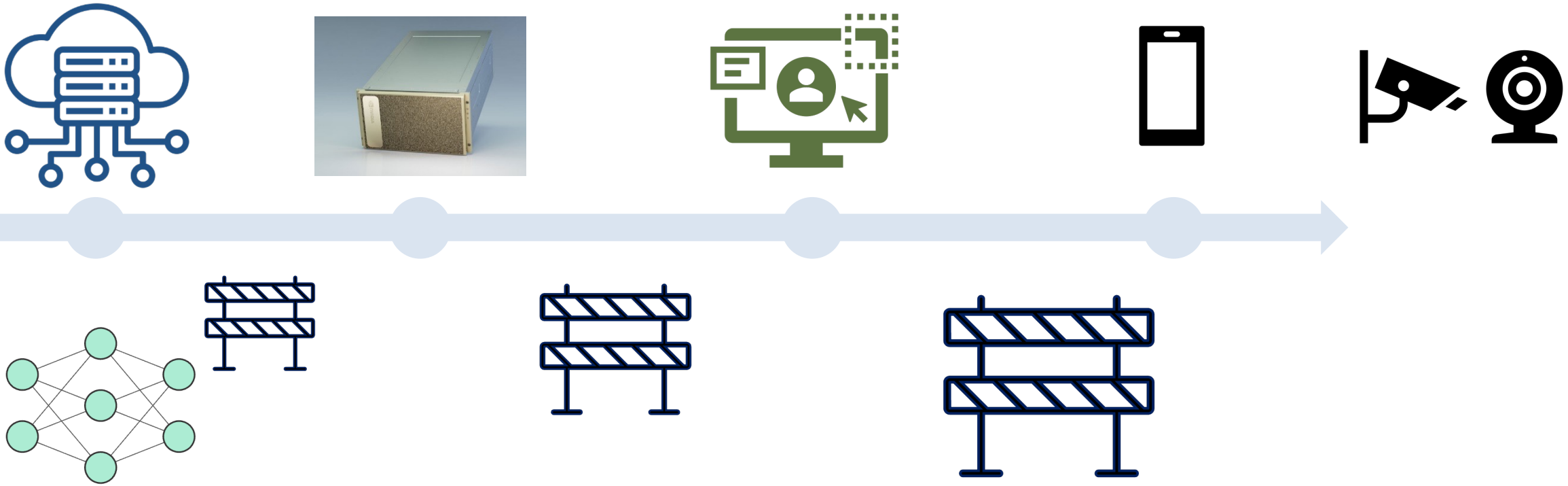
Hyeong-Ju Kang, "SSDLiteX: Enhancing SSDLite for Small Object Detection," *Applied Sciences*, 2023

Slide credit: Jacob

# Gap of Efficiency Expectation

## New challenges

- Edge-of-the-edge



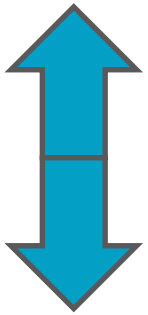
Slide credit: John

# Efficient AI

## AI on the edge

- Define the use case
- Select the right hardware
- Develop the AI model
- Optimize the model for edge deployment
- Test the system
- Deploy the system to production

**Face Detection**



**AI on Chips**

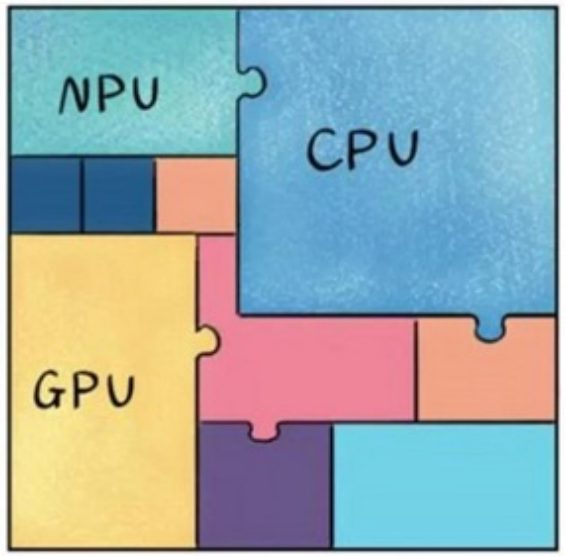
Slide credit: John

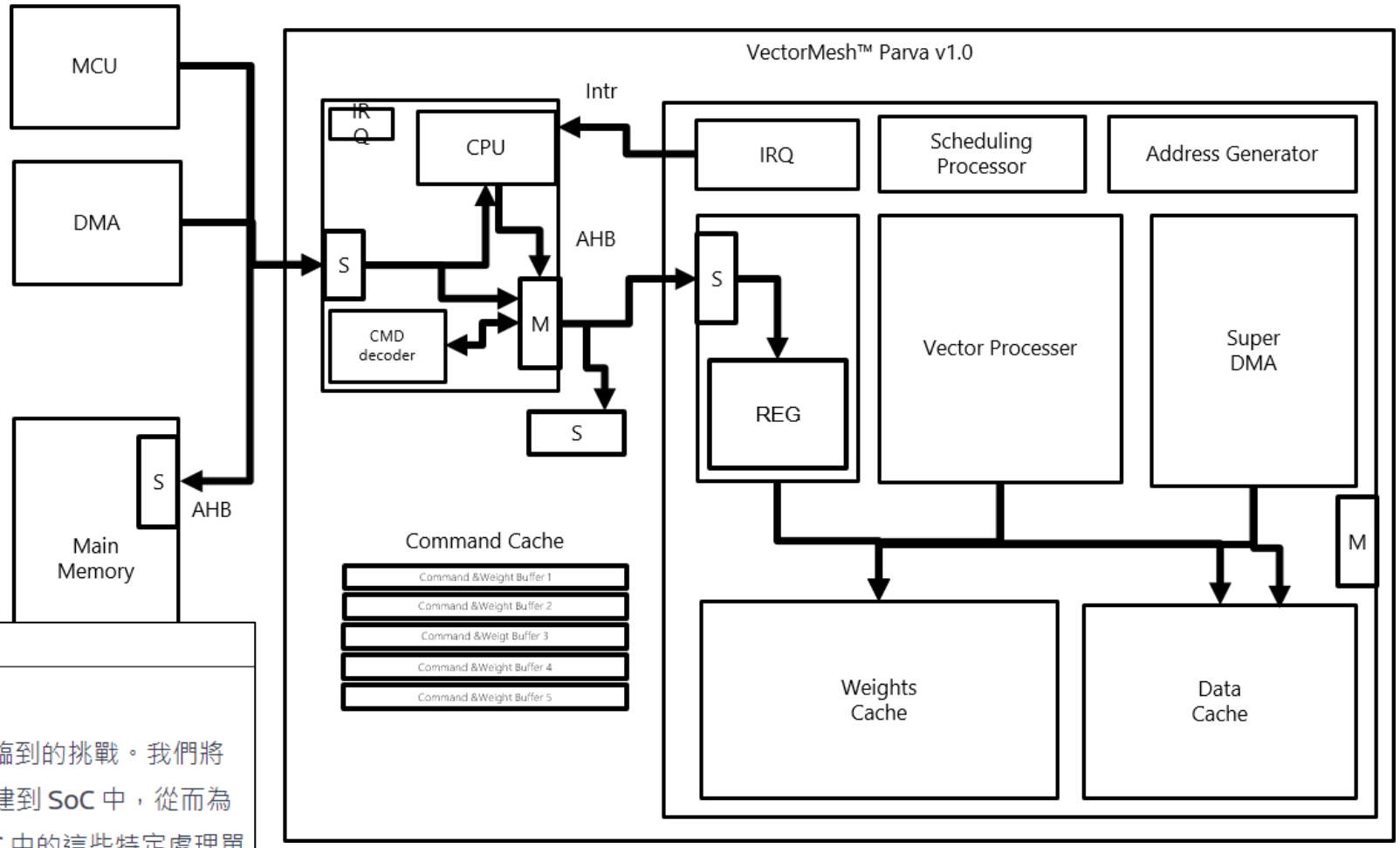
# Outline

- Overview
- **AI on Chip (AIoC) team work flow**
- Quantization
- Testing System
- Hardware design
  - NPU
  - ASIC (Application Specific Integrated Circuit)
- System-hardware co-design

# What is NPU IP?

## System on the chip (SoC)




### 聯發科技 NeuroPilot

我們以聯發科技 NeuroPilot 對應在終端人工智慧所面臨到的挑戰。我們將 CPU、GPU 和 APU（AI 處理單元）等異構運算功能內建到 SoC 中，從而為人工智慧功能和應用提供了高性能和低功耗。針對 SoC 中的這些特定處理單元，開發人員可以讓聯發科技 NeuroPoint SDK 智慧地為他們處理所分配到的任務。

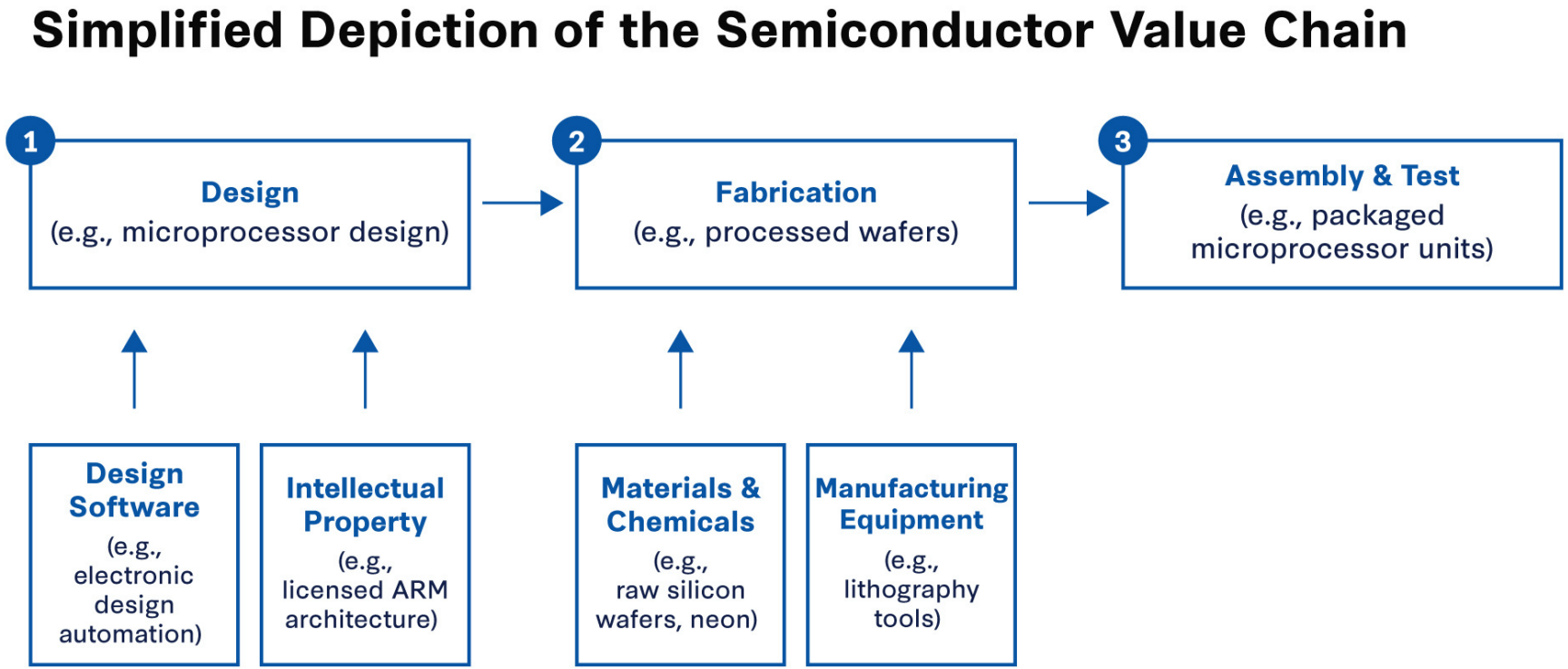
https://www.mediatek.tw/innovations/artificial-intelligence

Slide credit: Jacob

# Why NPU IP?

## System on the chip (SoC)

- SIP: Silicon Intellectual Property (矽智財)
- Faster, Better, Cheaper



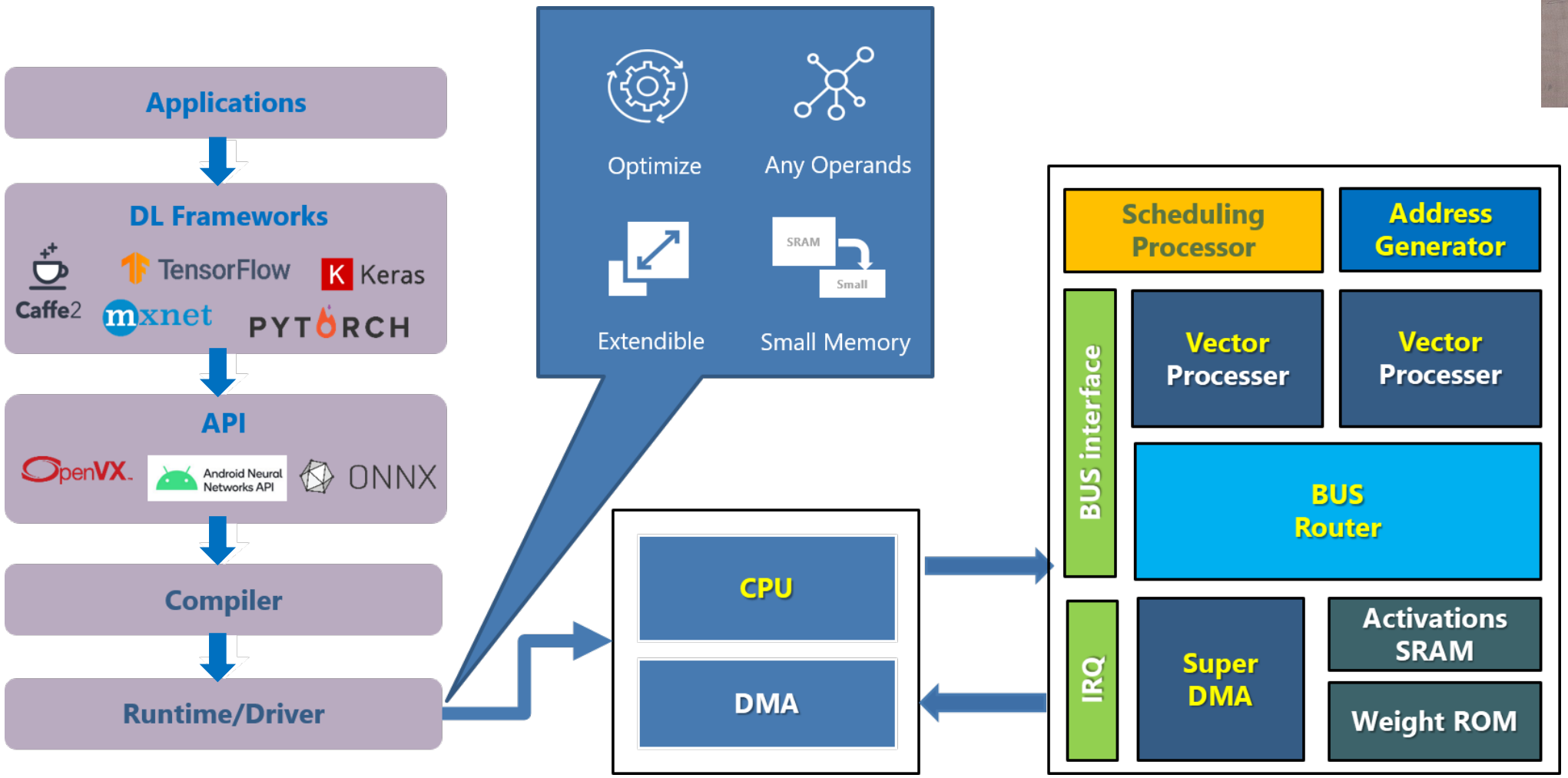

https://www.csis.org/analysis/mapping-semiconductor-supply-chain-critical-role-indo-pacific-region

Slide credit: Jacob

# One-stop Service?

## From AI Model Training to NPU IP Design

Opposite: Separate solutions for AI model, compiler, NPU design, … etc




https://www.guwenxuexi.com/classical/21785.html

Slide credit: Jacob

# AIoC Team Workflow

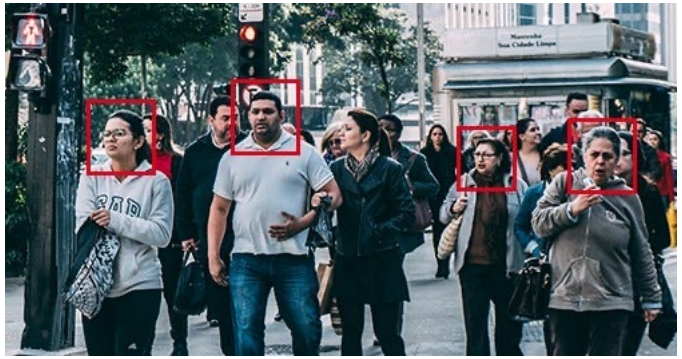
Feasibility Assessment

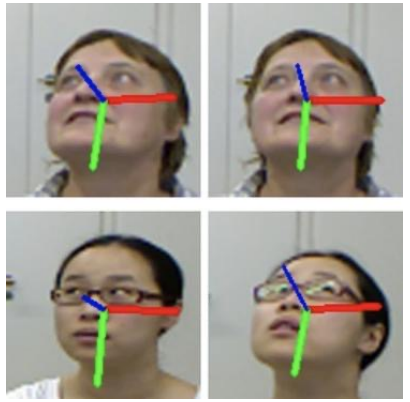
AI Model Design & Training

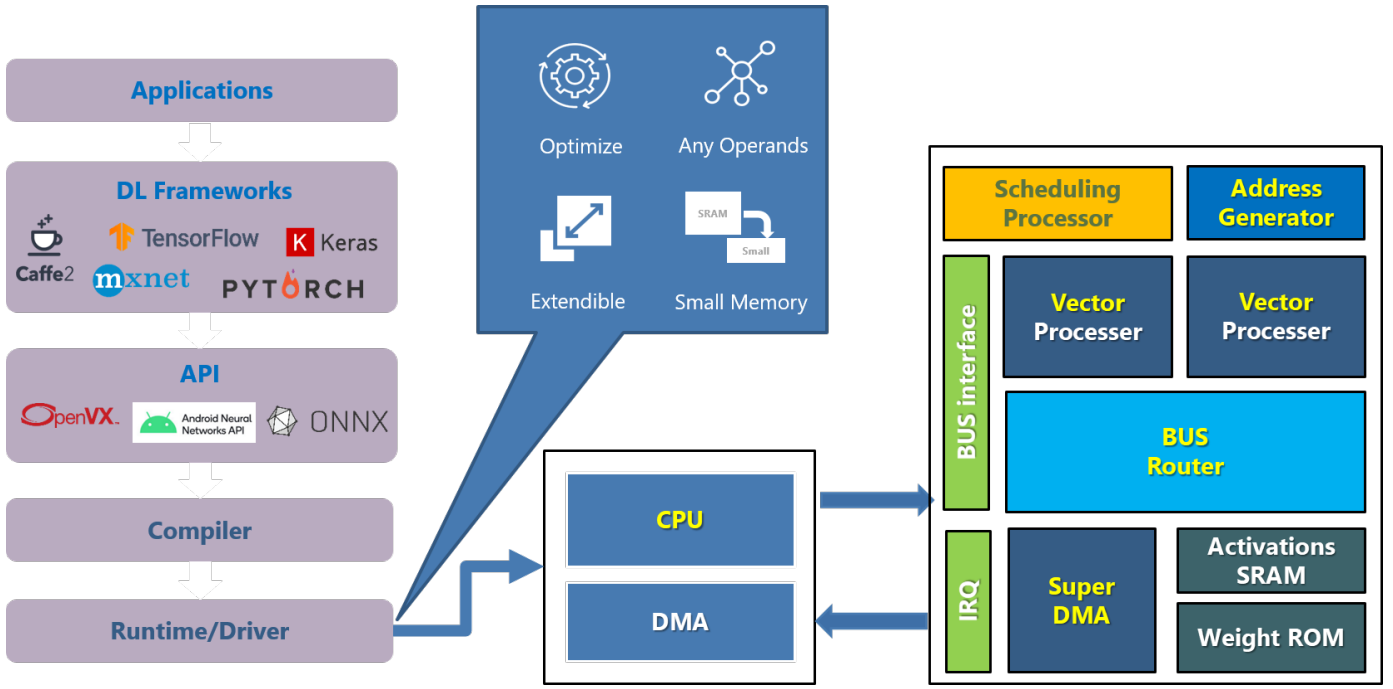
Quantization

Compile for NPU

Simulation & Testing





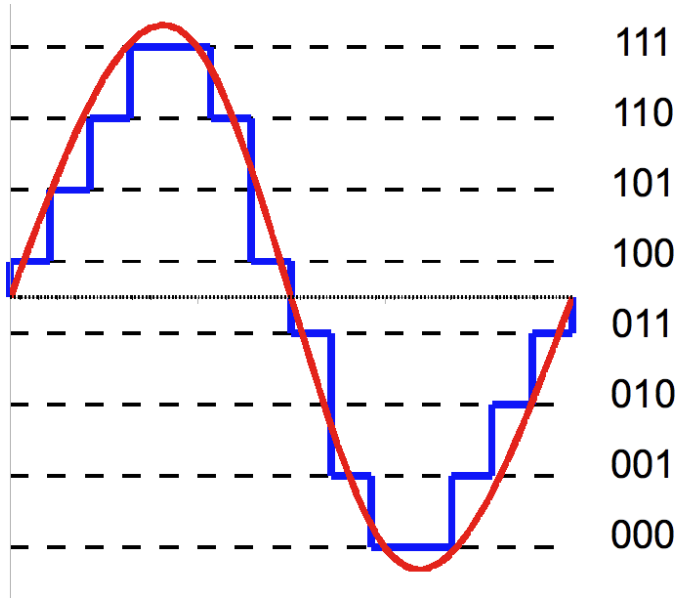


# Outline

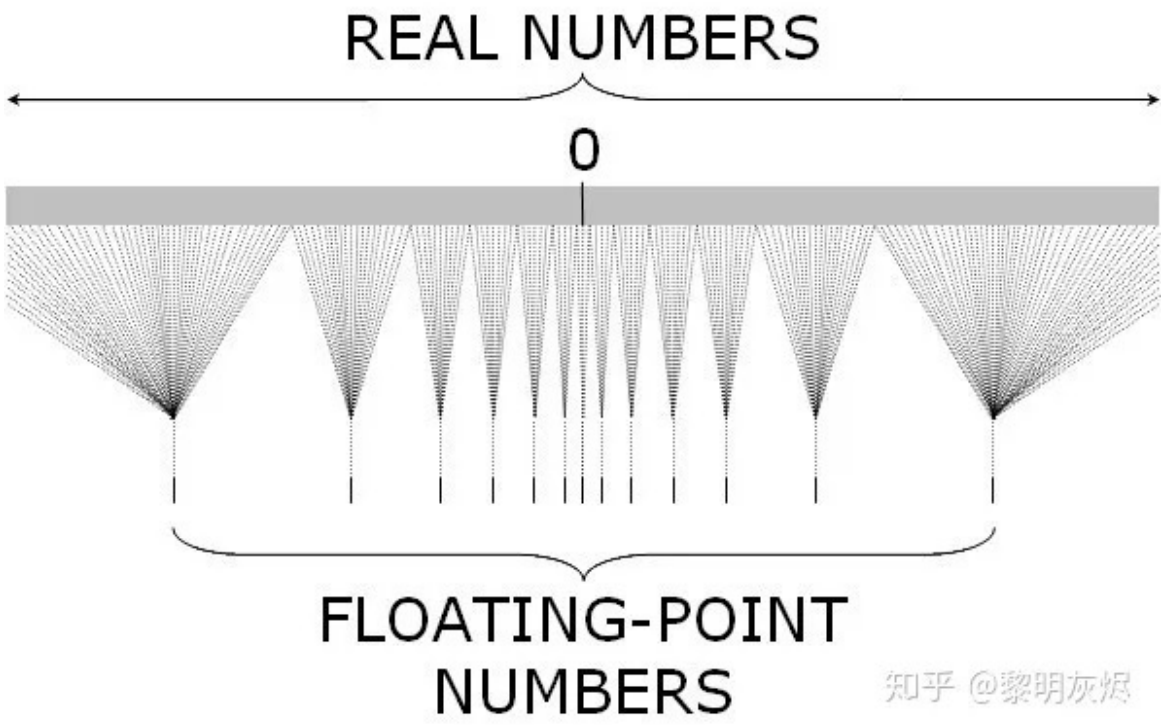
- Overview
- AI on Chip (AIoC) team work flow
- **Quantization**
- Testing System
- Hardware design
  - NPU
  - ASIC (Application Specific Integrated Circuit)
- System-hardware co-design

# What is Quantization?

- Process of turning continuous values into a finite set of discrete values.
- Computer Science & DSP: Used to reduce data precision.
  - Resource Efficiency
  - Faster Computation
  - **Hardware Implementation**



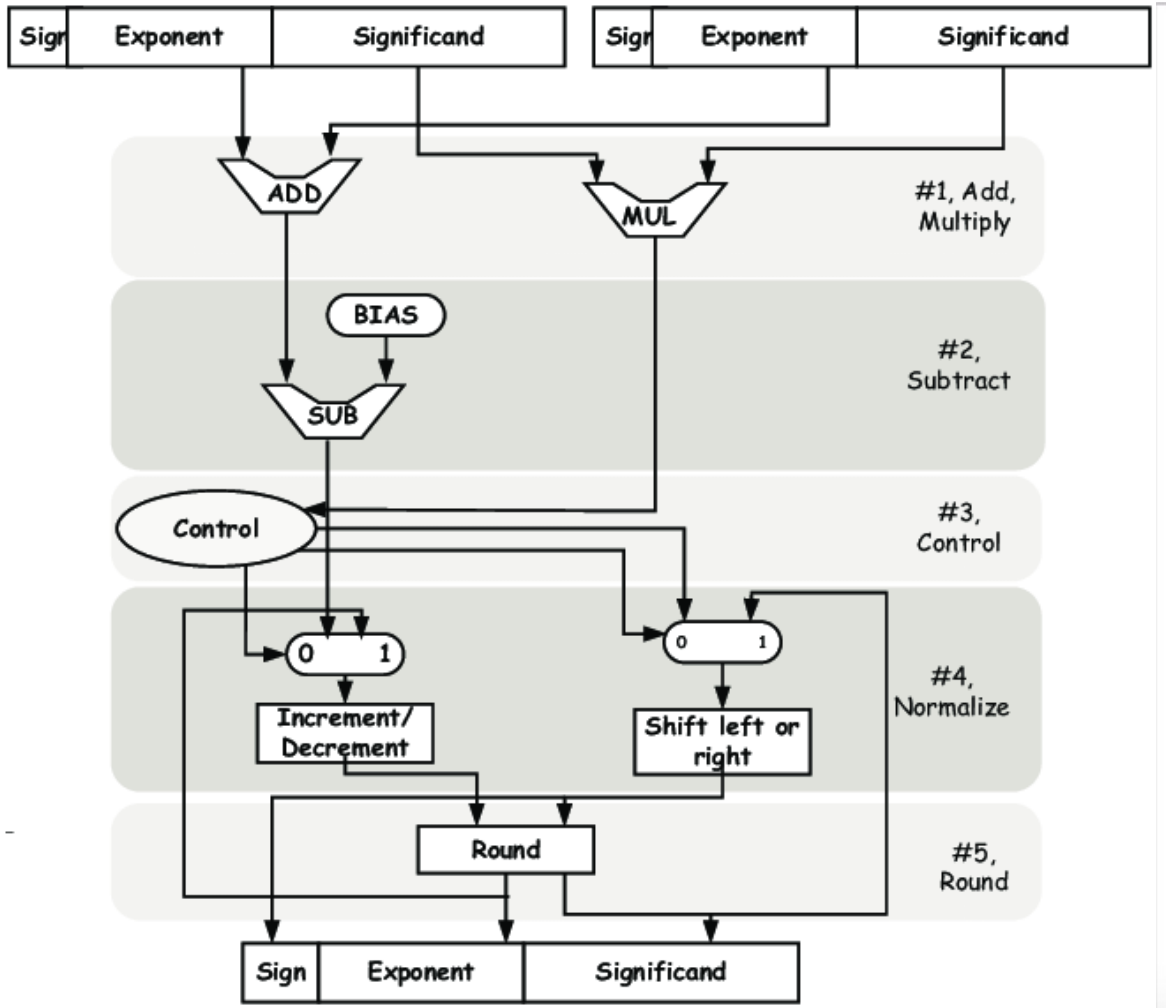

https://www.differencebetween.com/difference-between-uniform-and-nonuniform-quantization/



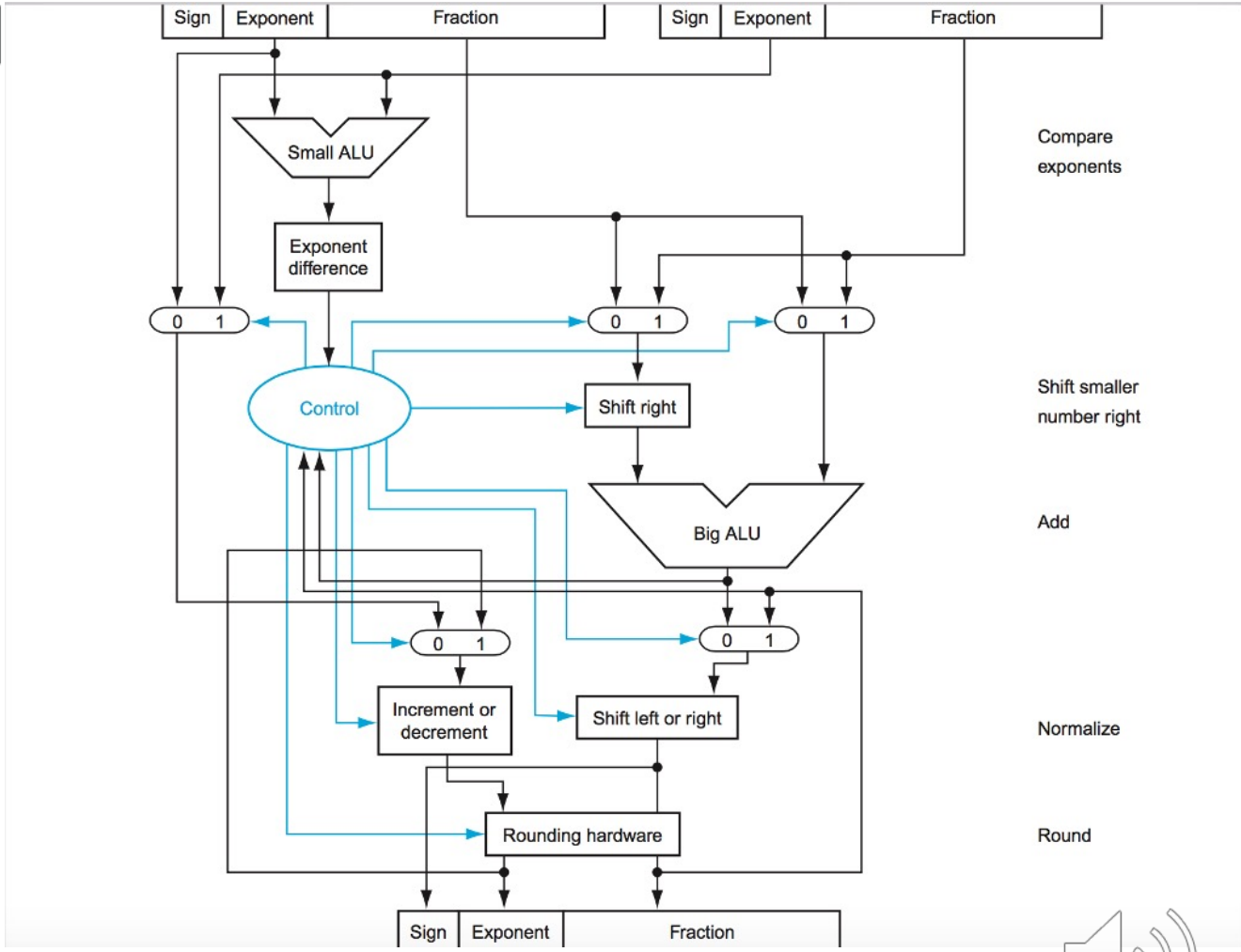

https://iter01.com/175359.html

# Hardware Implementation

## Floating point vs fixed point

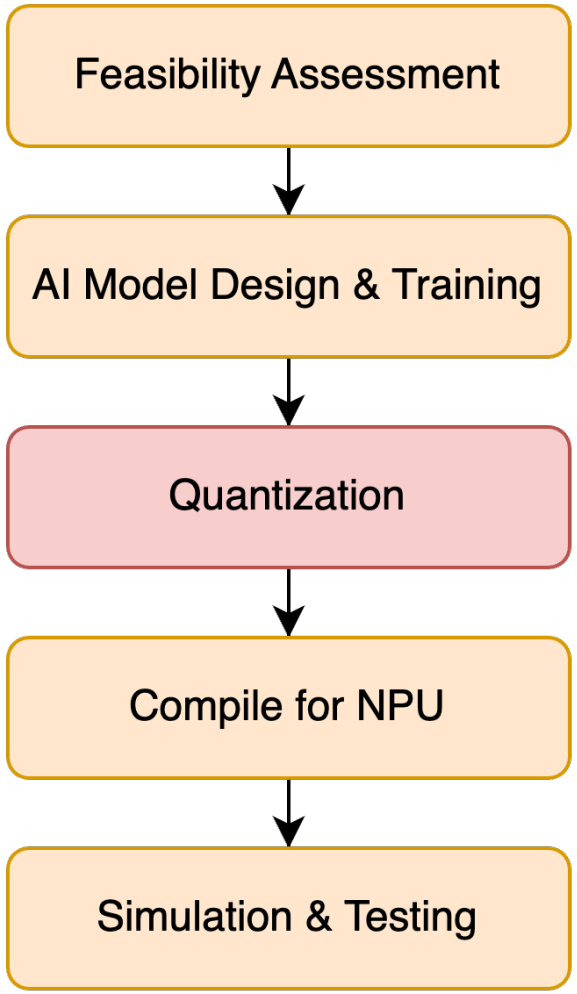


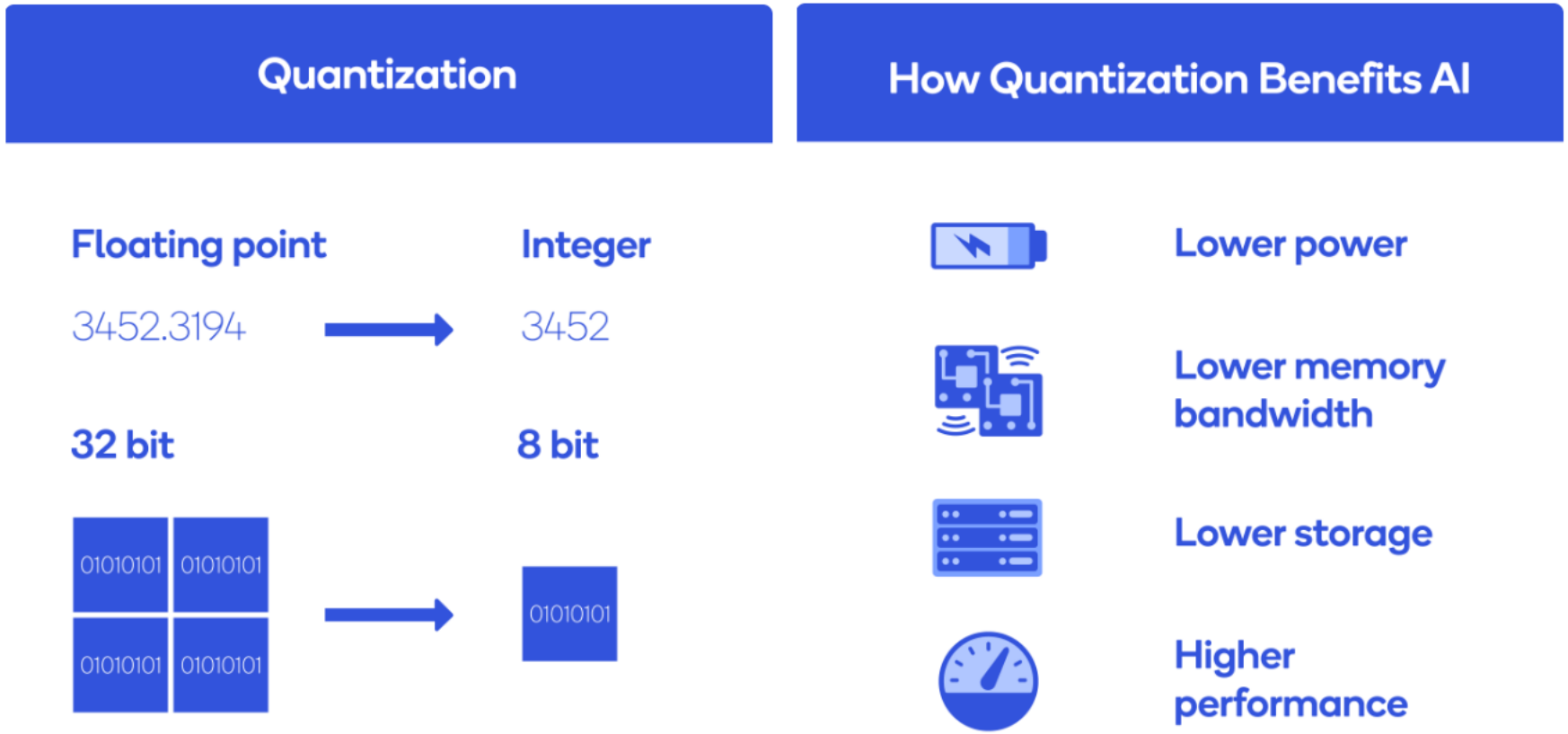

https://www.researchgate.net/figure/Floating-Point-Multiplier-Architecture_fig2_224698133




https://www.chegg.com/homework-help/questions-and-answers/design-floating-point-add...-unit-adding-two-32-bit-floating-point-numbers-ieee-754-form-q21342231

# Quantization for Neural Network




https://www.qualcomm.com/news/onq/2019/03/heres-why-quantization-matters-ai

# Quantization for Neural Network

## Asymmetric vs. Symmetric

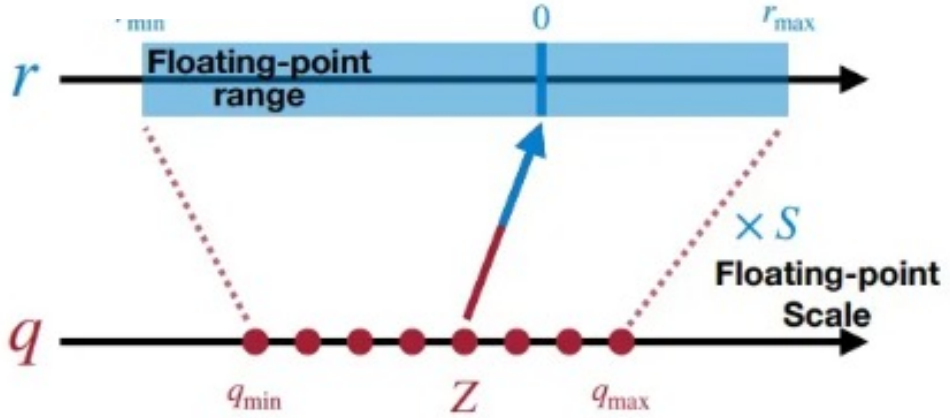
- r: floating point
- q: fixed point after quantization
- S: scale
- Z: zero point

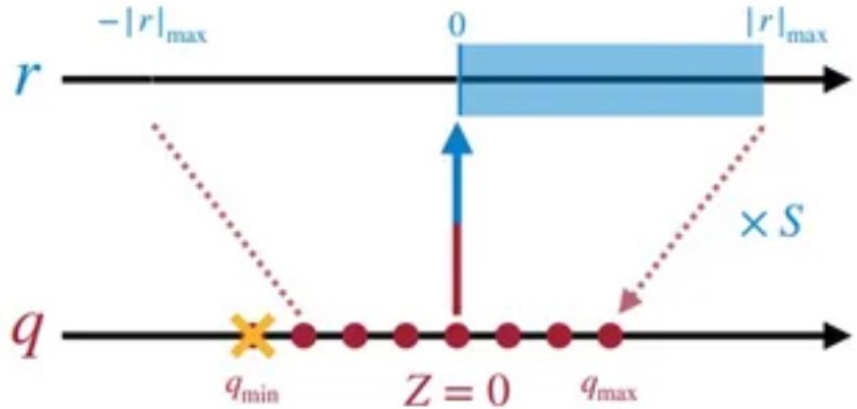
$$r = S(q - Z)$$

$$q = round(\frac{r}{S} + Z)$$

$$S = \frac{r_{max} - r_{min}}{q_{max} - q_{min}}$$

$$Z = round(q_{max} - \frac{r_{max}}{S})$$




https://zhuanlan.zhihu.com/p/629593666

# Quantization for Neural Network

## Quantization of matrix operations

- 2D Convolution can be converted to matrix multiplication
- **$R_3 = R_1 \times R_2$**

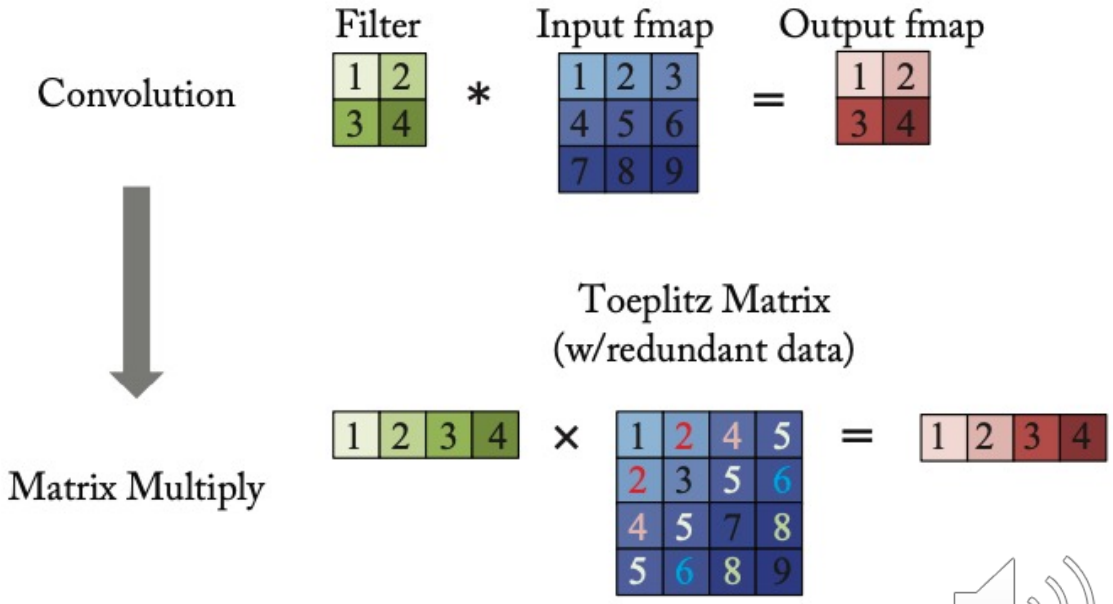
$$r_3^{i,k} = \sum_{j=1}^{N} r_1^{i,j} r_2^{j,k}$$

$$S_3(q_3^{i,k} - Z_3) = \sum_{j=1}^{N} S_1(q_1^{i,j} - Z_1) S_2(q_2^{j,k} - Z_2)$$

$$q_3^{i,k} = \frac{S_1 S_2}{S_3} \sum_{j=1}^{N} (q_1^{i,j} - Z_1)(q_2^{j,k} - Z_2) + Z_3$$

$$r = S(q - Z)$$

$$q = round(\frac{r}{S} + Z)$$




https://zhuanlan.zhihu.com/p/149659607

https://www.researchgate.net/figure/Mapping-convolution-functions-to-the-matrix-multiplication-for-the-conversion-of-the_fig2_347268474

# Quantization for Neural Network

- M is usually between 0 and 1 (statistical results)

$$q_3^{i,k} = \frac{S_1 S_2}{S_3} \sum_{j=1}^{N} (q_1^{i,j} - Z_1)(q_2^{j,k} - Z_2) + Z_3$$

$$M = \frac{S_1 S_2}{S_3}$$

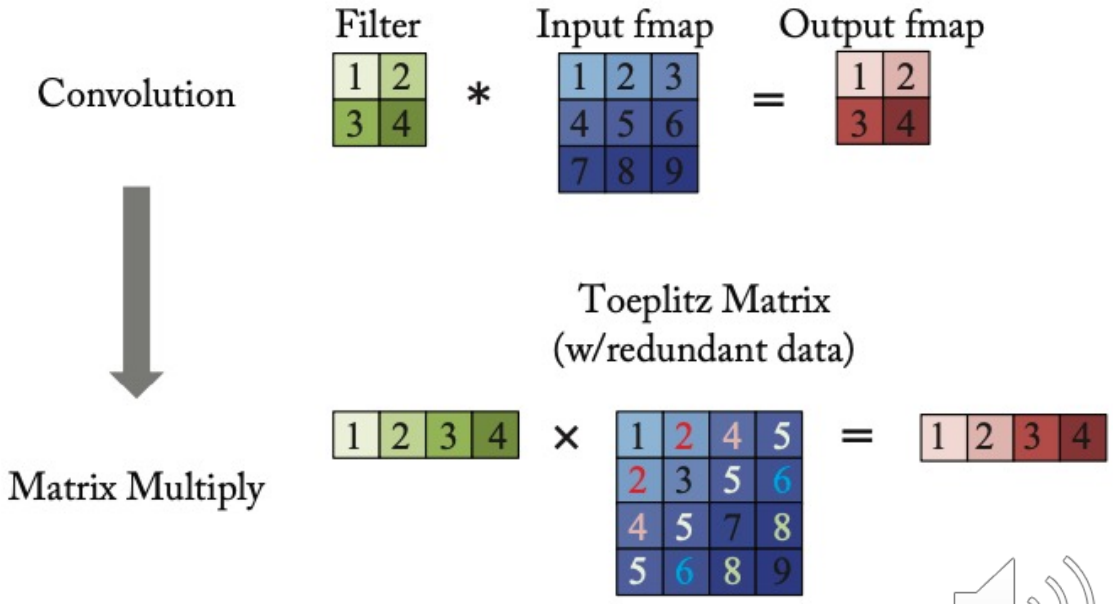
$$q_3^{i,k} = M \sum_{j=1}^{N} (q_1^{i,j} - Z_1)(q_2^{j,k} - Z_2) + Z_3 = MP + Z_3$$

$$M = 2^{-n} M_0$$

$$q_3^{i,k} = 2^{-n} M_0 P + Z_3$$

$$r = S(q - Z)$$

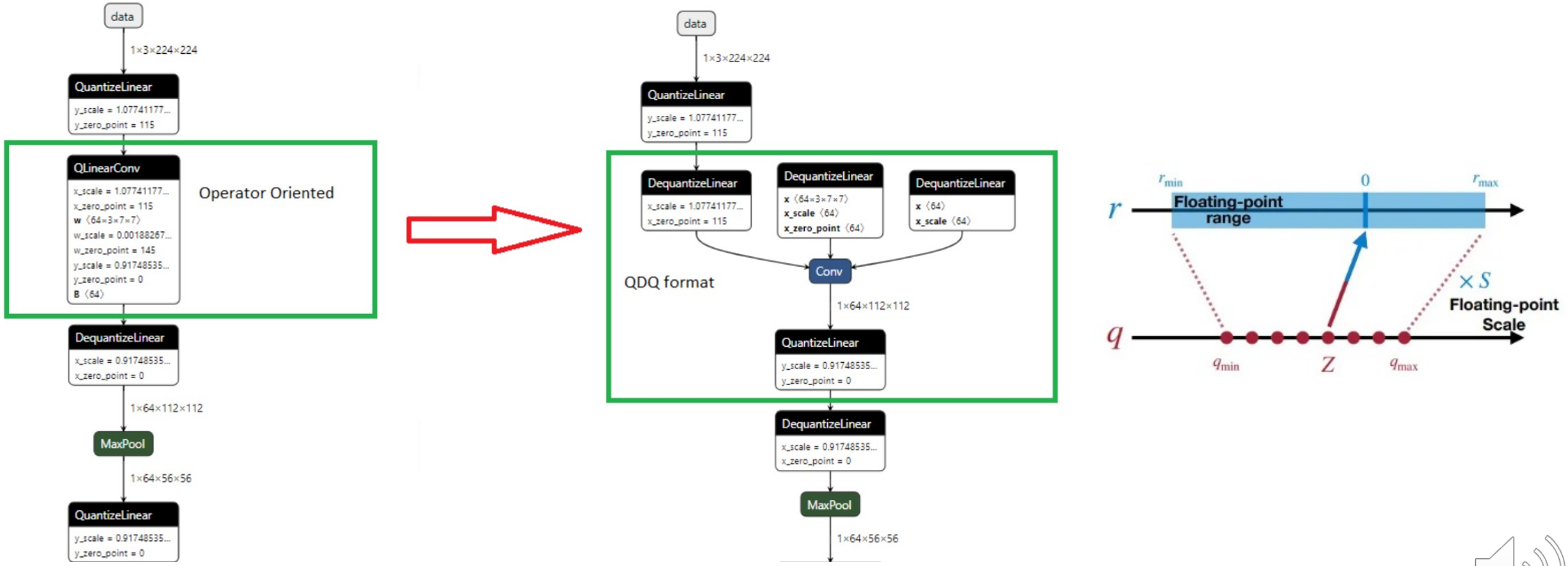
$$q = round(\frac{r}{S} + Z)$$




https://zhuanlan.zhihu.com/p/149659607

# Onnx Quantization

- Convert floating point model to QDQ format and perform **calibration**




https://onnxruntime.ai/docs/performance/model-optimizations/quantization.html

# Quantization

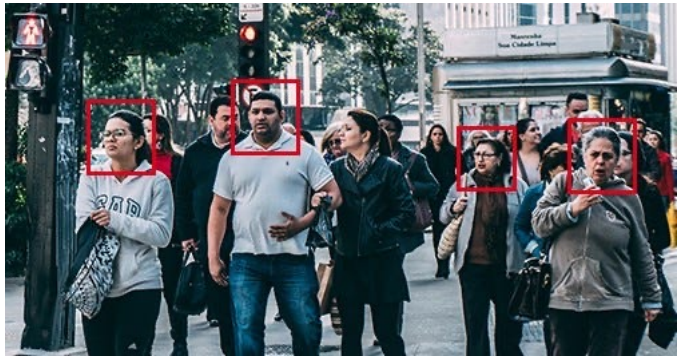
Feasibility Assessment

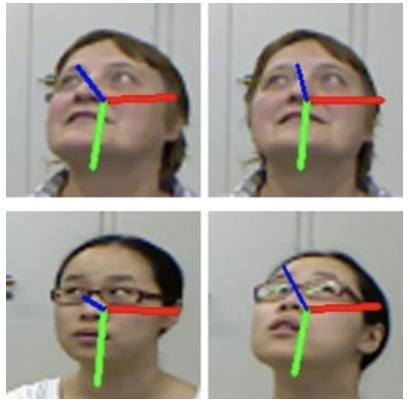
AI Model Design & Training

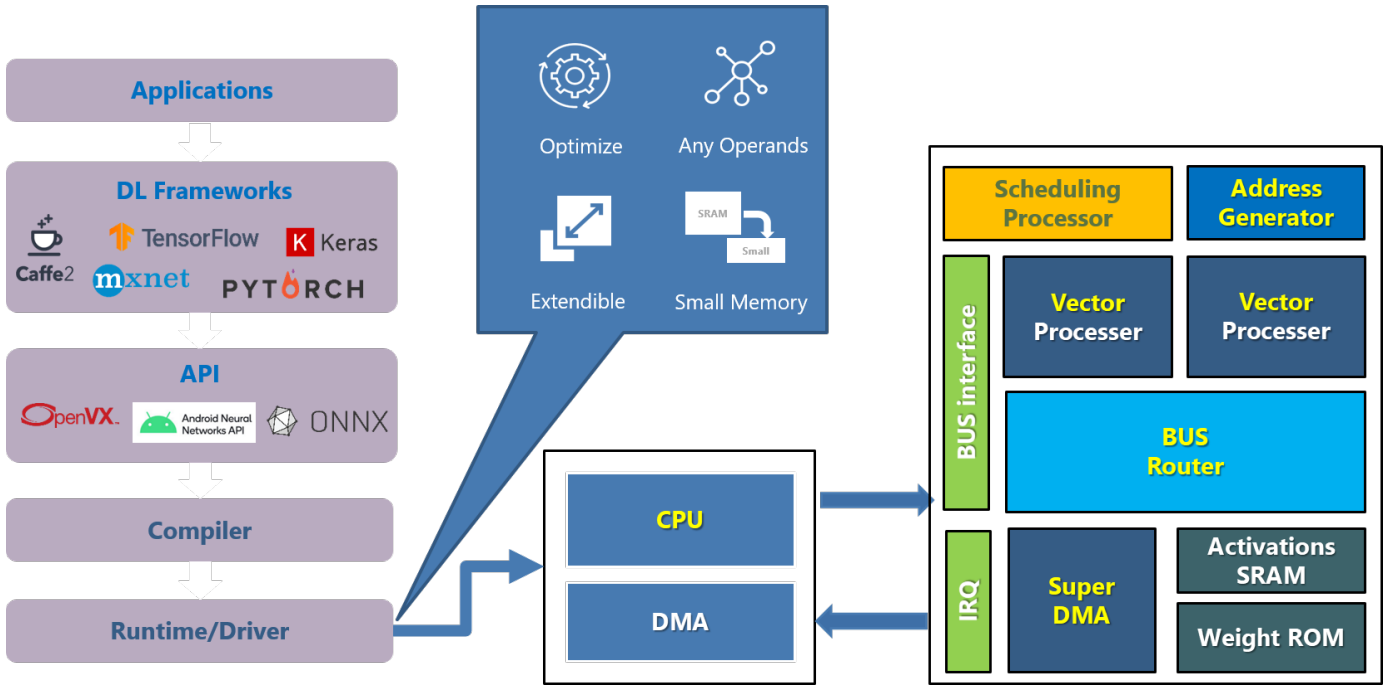
Quantization

Compile for NPU

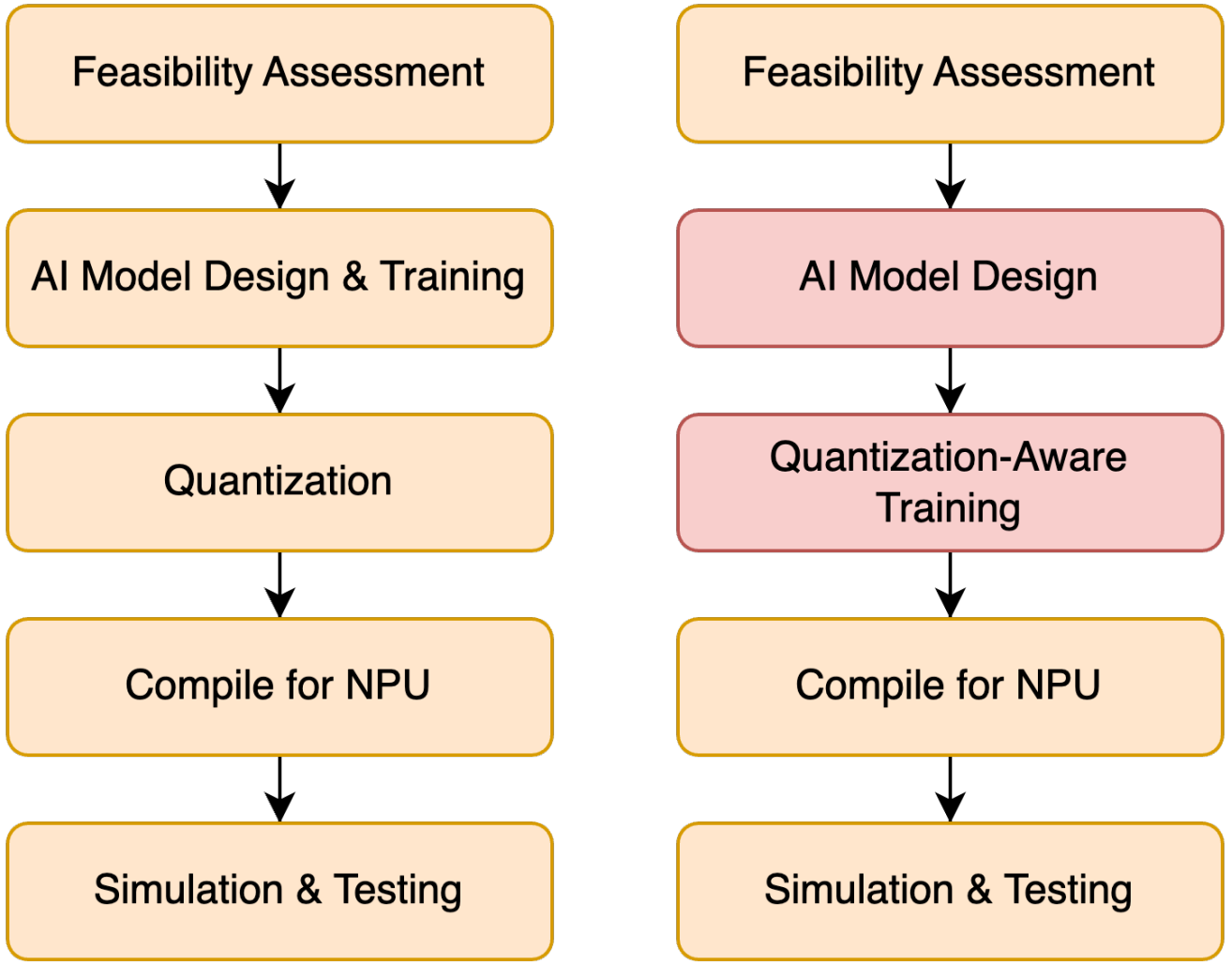
Simulation & Testing







# Quantization Aware Training



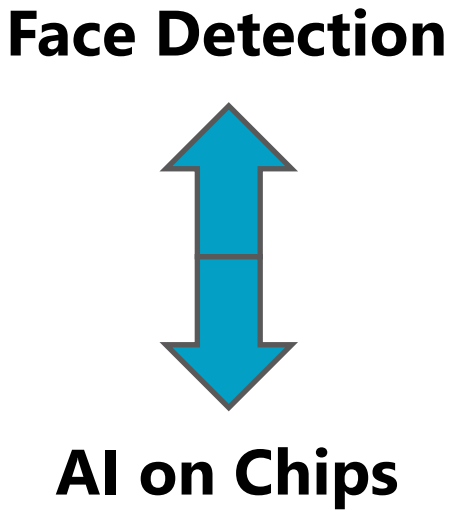
# Outline

- Overview
- AI on Chip (AIoC) team work flow
- Quantization
- **Testing System**
- Hardware design
  - NPU
  - ASIC (Application Specific Integrated Circuit)
- System-hardware co-design

# Efficient AI

## AI on the edge

- Define the use case
- Select the right hardware
- Develop the AI model
- Optimize the model for edge deployment
- Test the system
- Deploy the system to production



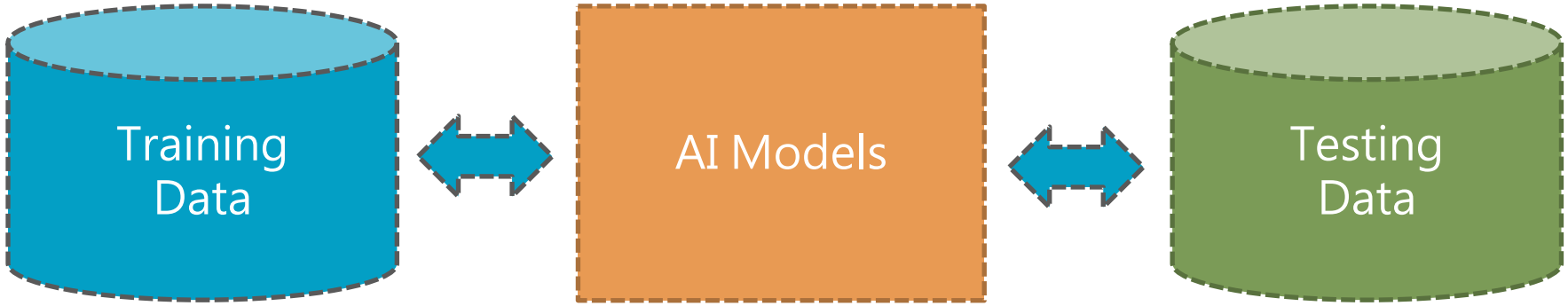

Slide credit: John

# What is Learning in AI?

## A program that writes itself… using framework and data

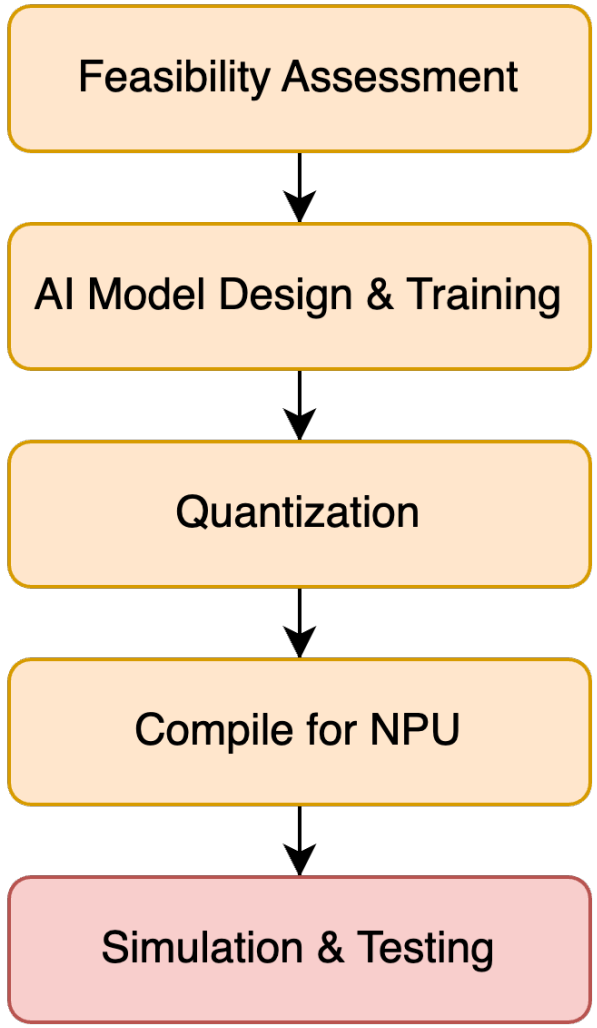
- **Applications programmed against spec and verified against test cases.**

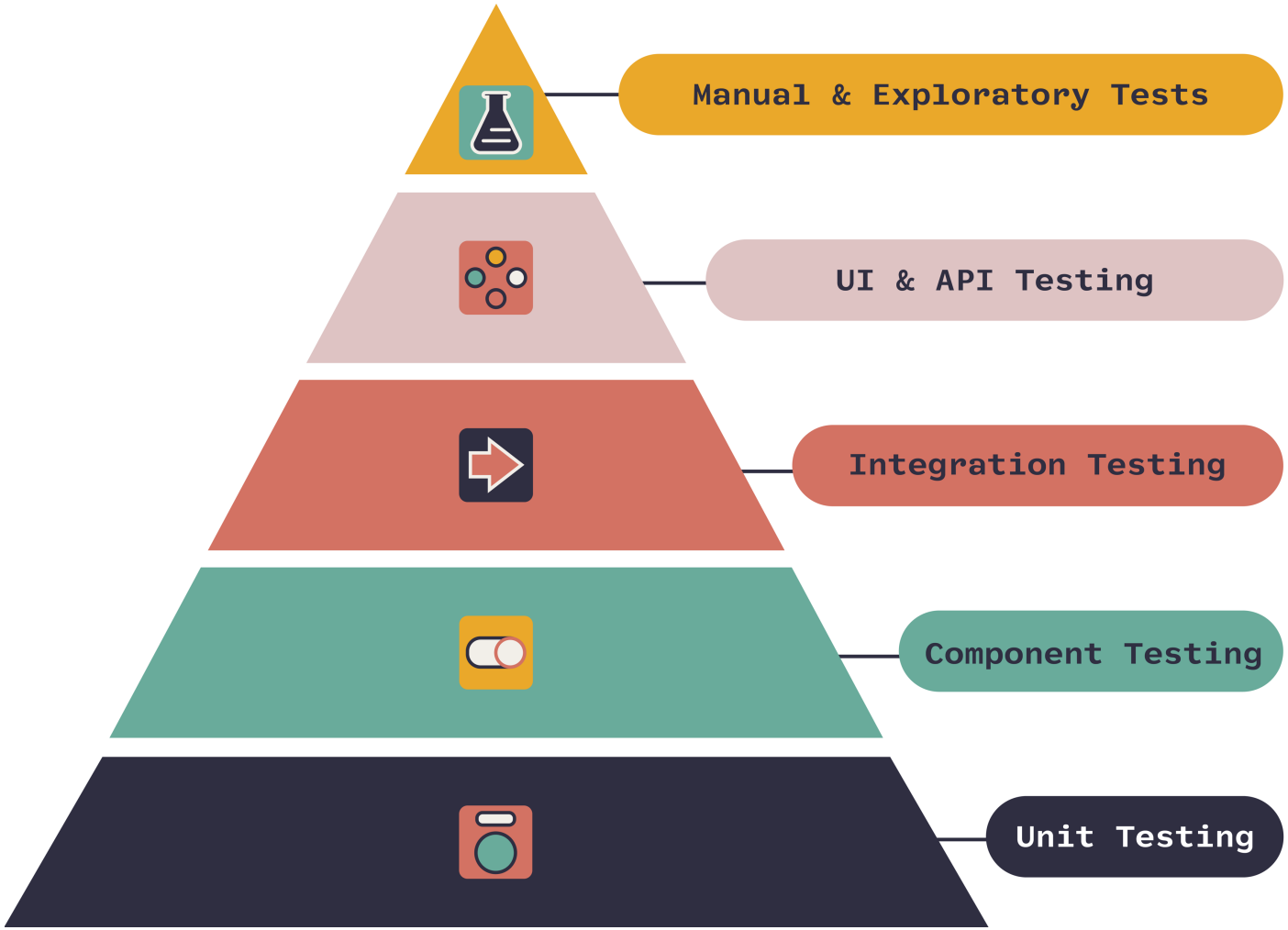

Sounds familiar?




Slide credit: Wei-Chao Chen

# Importance of Testing



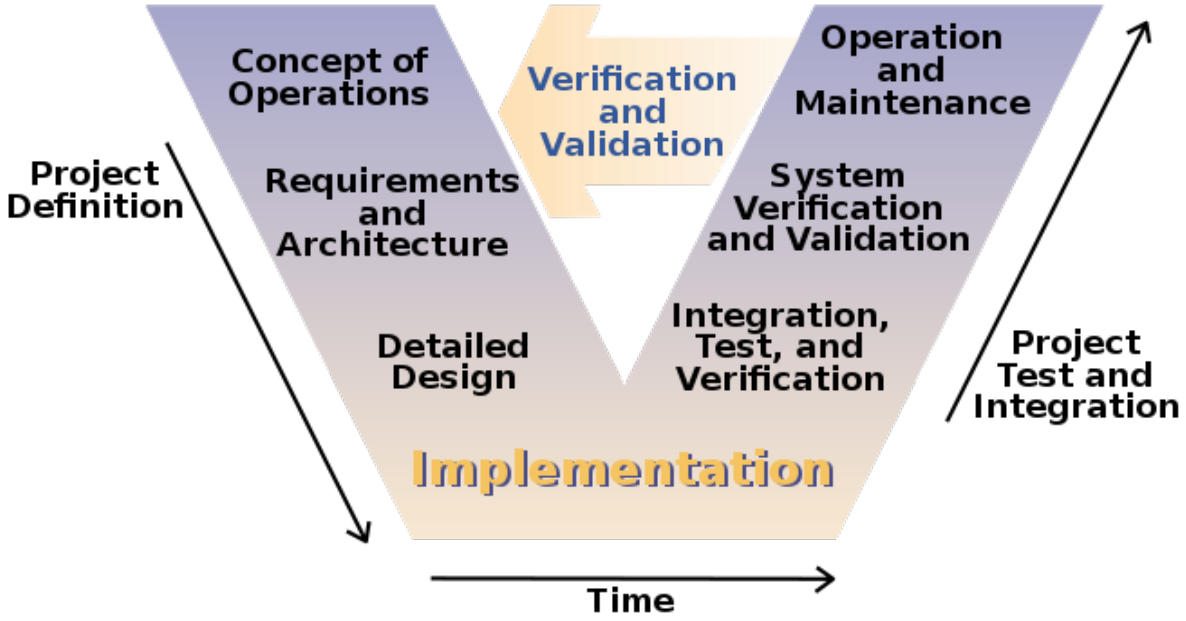



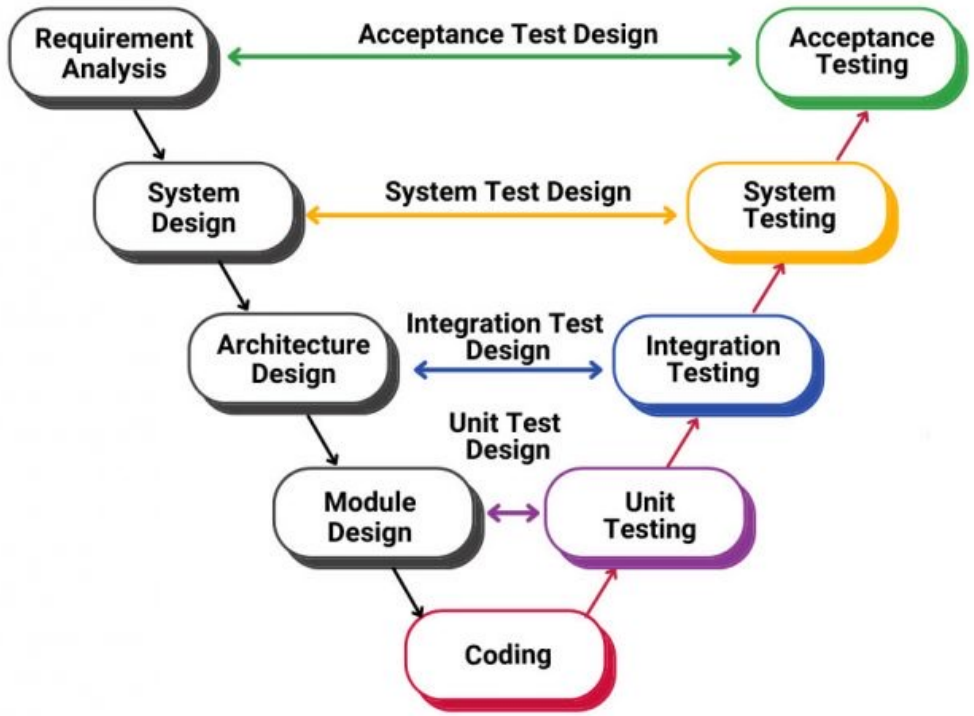

https://www.sipsa.net/en/levels-of-software-testing/

# V-Model

## Standard software development process

- Verification: Are we building the product right (according to spec)?
- Validation: Are we building the right product (meet user requirements/expectations)?

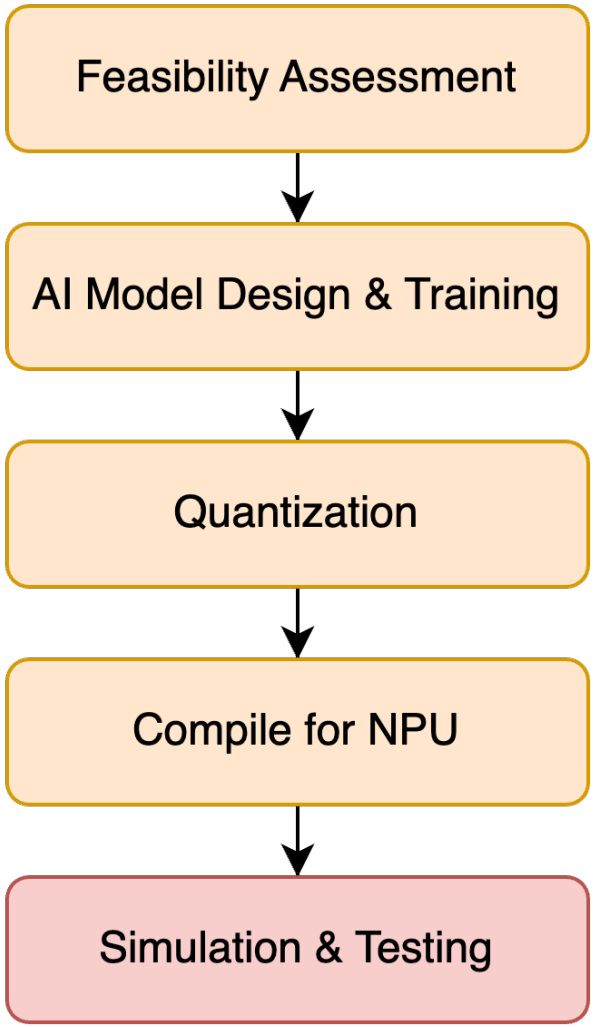


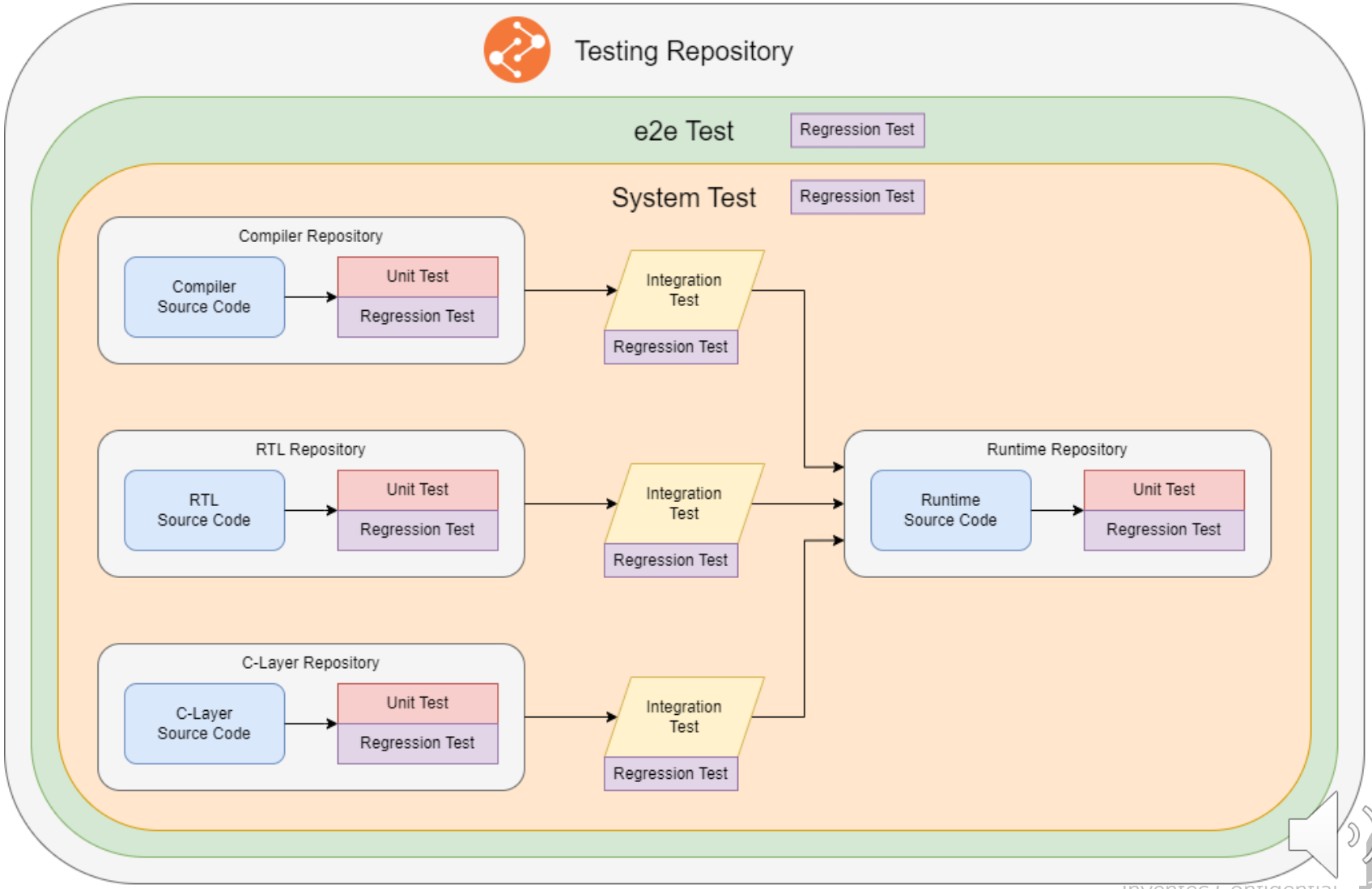




https://en.wikipedia.org/wiki/V-model_(software_development)
https://www.coleyconsulting.co.uk/v-model-verification-and-validation.htm

Slide credit: Jacob

# Testing System



# Outline

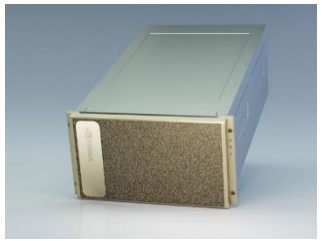
- Overview
- AI on Chip (AIoC) team work flow
- Quantization
- Testing System
- **Hardware design**
  - NPU
  - ASIC (Application Specific Integrated Circuit)
- System-hardware co-design

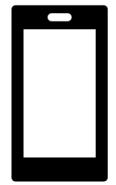
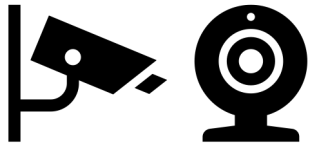
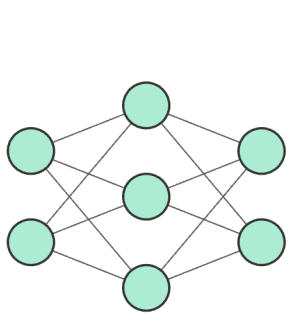
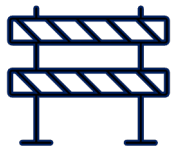
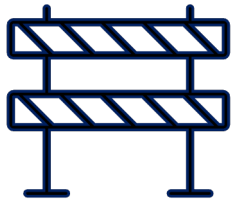
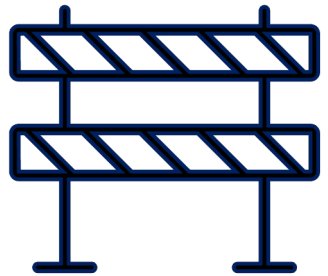
# Software Engineer

## New challenges

- Edge-of-the-edge

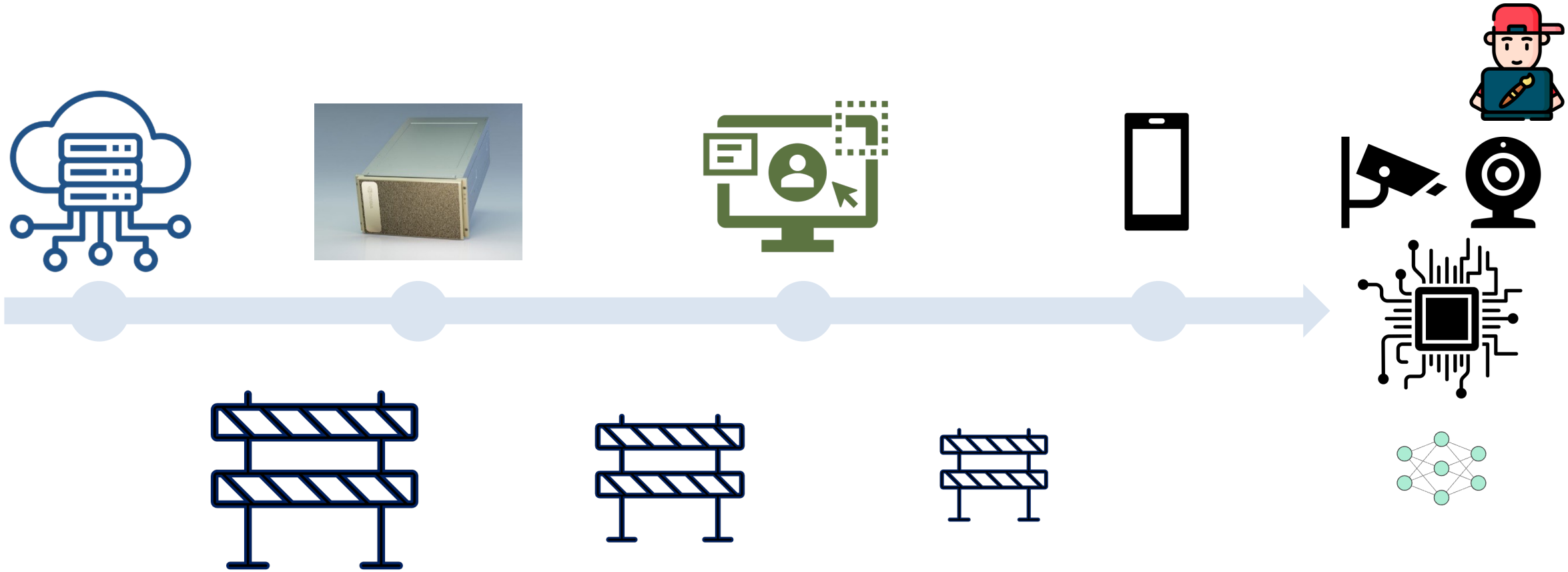




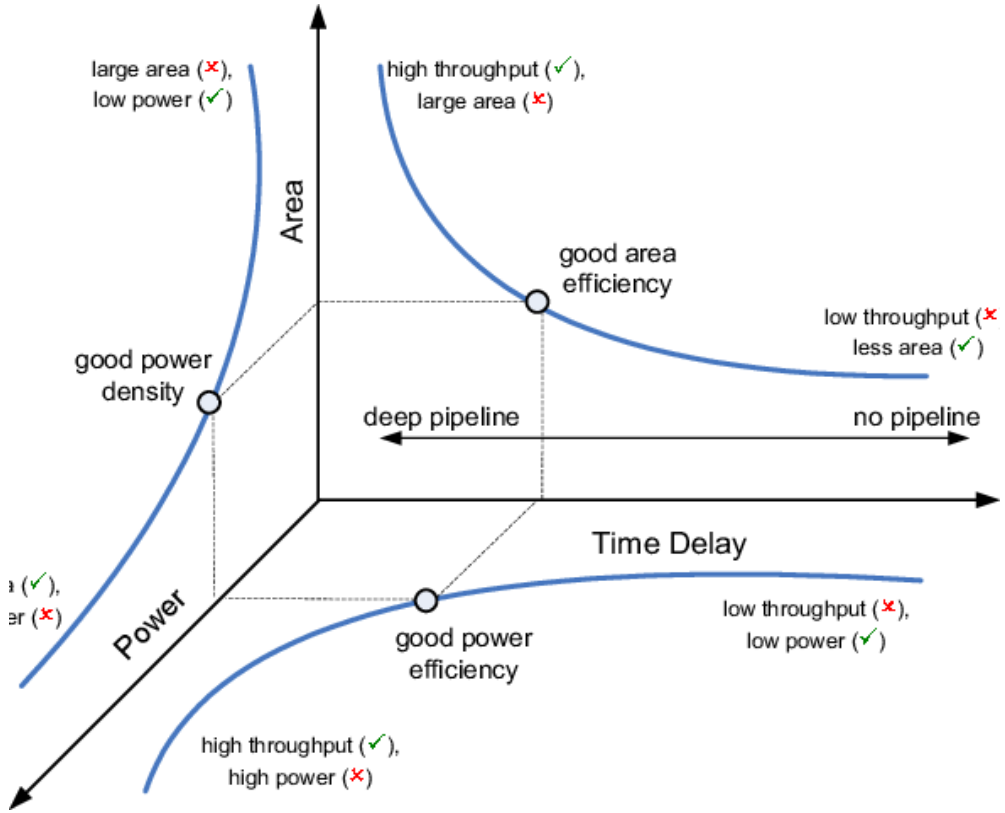







# Hardware Engineer

## New challenges



# Measurement of Efficiency



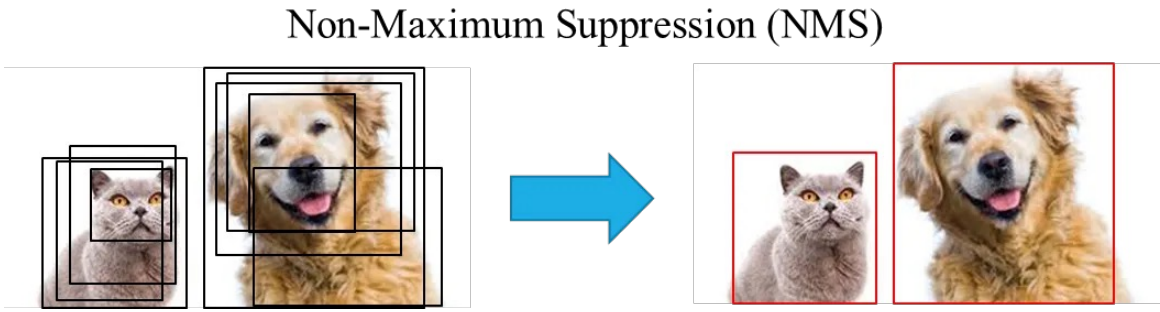

- **Hardware-specific**
  - Area (Gate count)
    - Number of computing unit
    - Memory footprint
  - Time Delay
  - Power (W/s)
- **Test metric – might be very different**
  - Accuracy: The quality of the result for a given task
  - Throughput: Inferences per second.
  - Latency: The time between the input data arrives and the result is generated.
  - Energy: Total energy of executing a task.

Bartoldson, Brian R., Bhavya Kailkhura, and Davis Blalock. "Compute-Efficient Deep Learning: Algorithmic Trends and Opportunities." *Journal of Machine Learning Research* 24 (2023): 1-77.

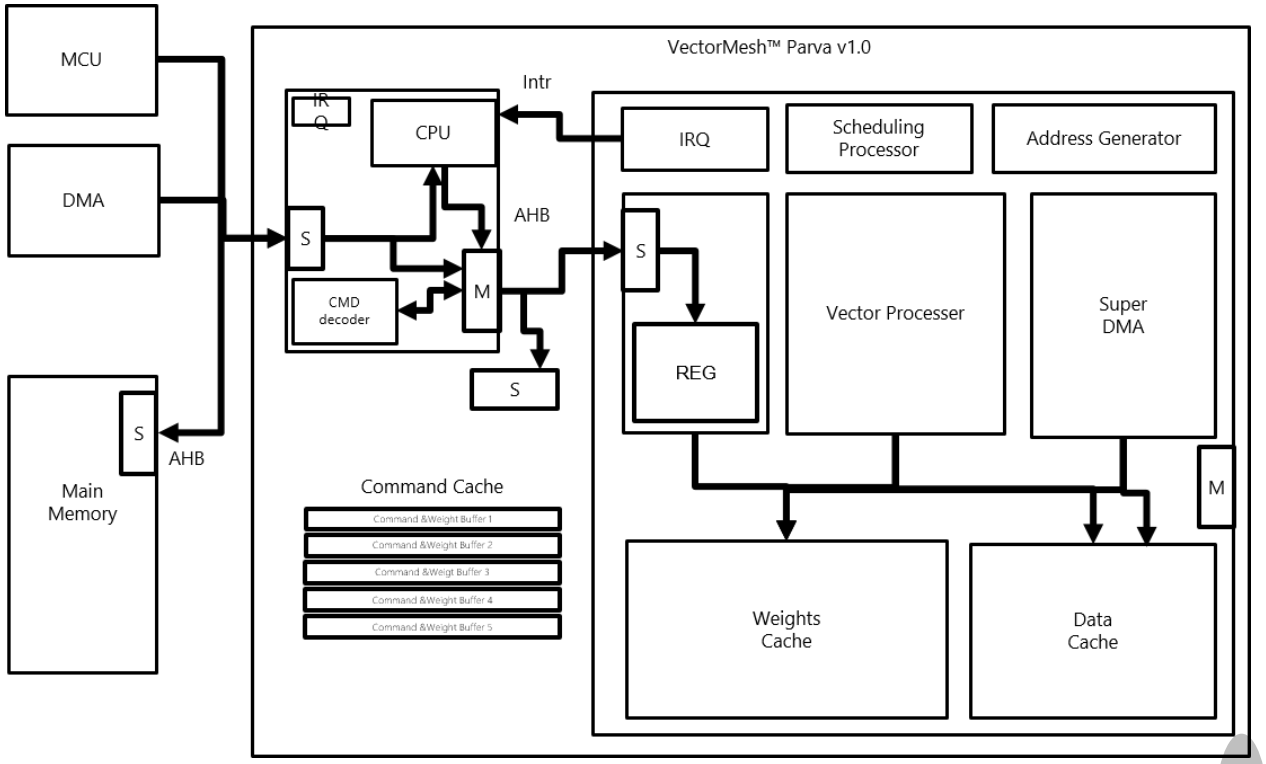
# Outline

- Overview
- AI on Chip (AIoC) team work flow
- Quantization
- Testing System
- Hardware design
  - **NPU**
  - ASIC (Application Specific Integrated Circuit)
- System-hardware co-design

# NPU – Design Philosophy of RISC-V




- What computations do we need?
  - Image pre-processing (denoising, gamma, …)
  - Neural network inference (CNN kernel computation, activation, …)
  - Data post-processing (NMS, ID matching, …)
- What kinds of hardware do we have?
  - NPU: CNN operation
  - DMA: resize, crop
  - MCU: runtime, NMS
  - ISP: image enhancement




Slide credit: Jacob

https://chih-sheng-huang821.medium.com/%E6%A9%9F%E5%99%A8-%E6%B7%B1%E5%BA%A6%E5%AD%B8%E7%BF%92-%E7%89%A9%E4%BB%B6%E5%81%B5%E6%B8%AC-non-maximum-suppression-nms-aa70c45adffa

# Outline

- Overview
- AI on Chip (AIoC) team work flow
- Quantization
- Testing System
- Hardware design
  - NPU
  - **ASIC (Application Specific Integrated Circuit)**
- System-hardware co-design

# Systolic Array

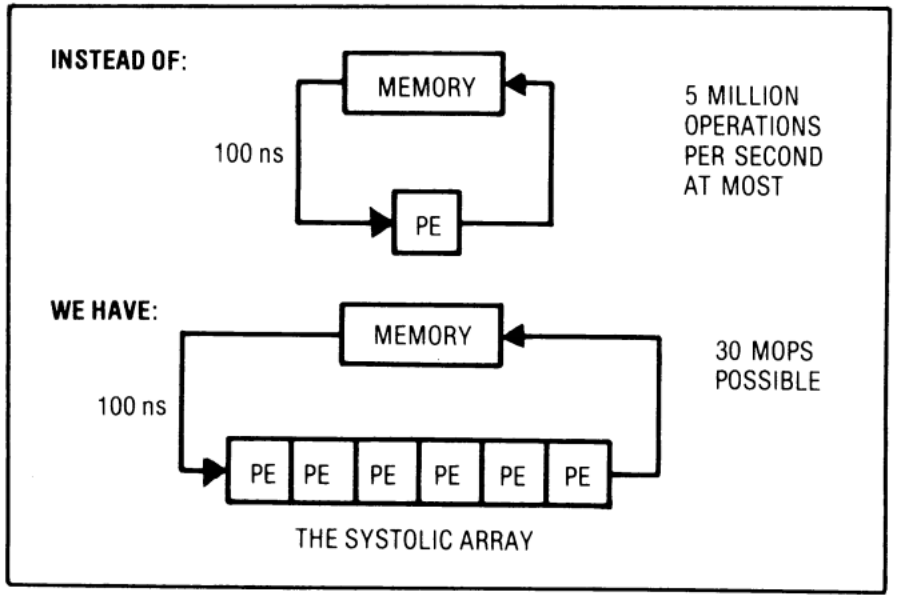
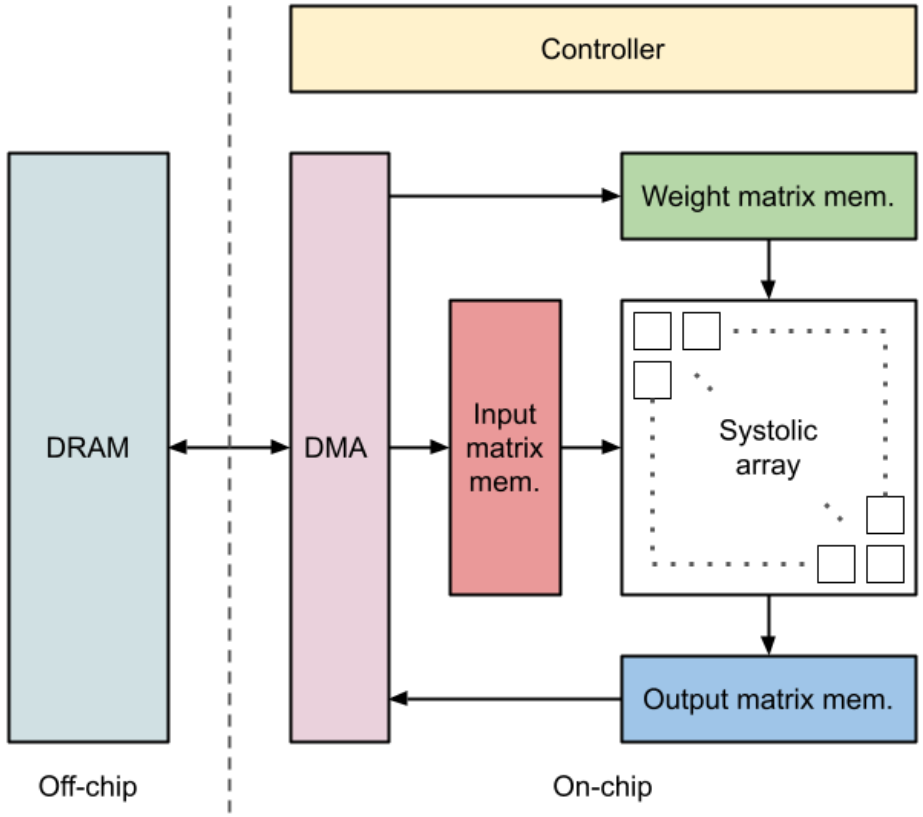



**Figure 1. Basic principle of a systolic system.**

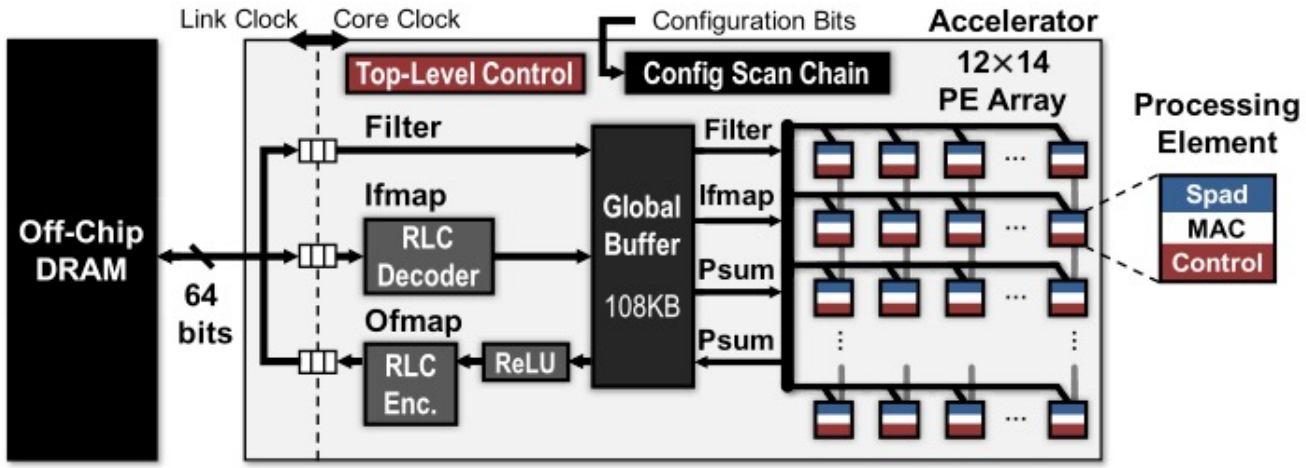


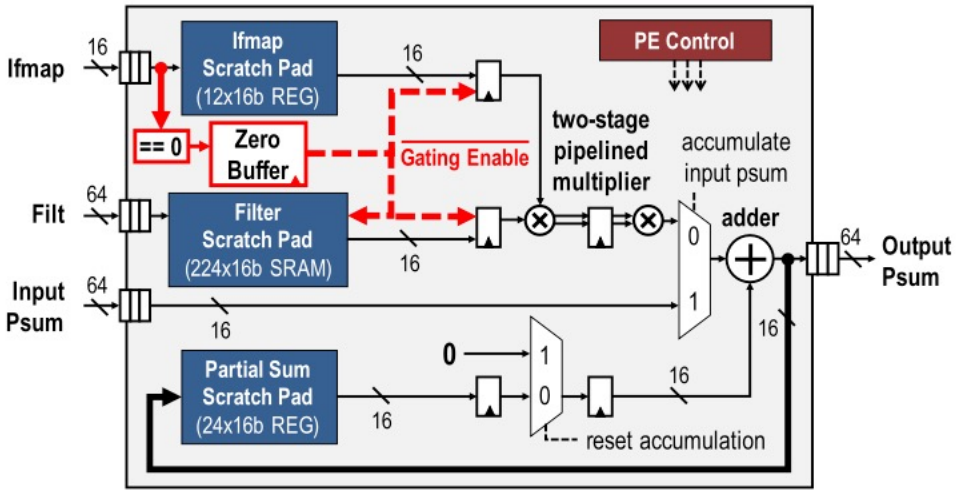
https://blog.csdn.net/wangwangmoon_light/article/details/121890472




https://towardsdatascience.com/accelerating-neural-networks-on-hardware-baa3c14cd5ba

# Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks

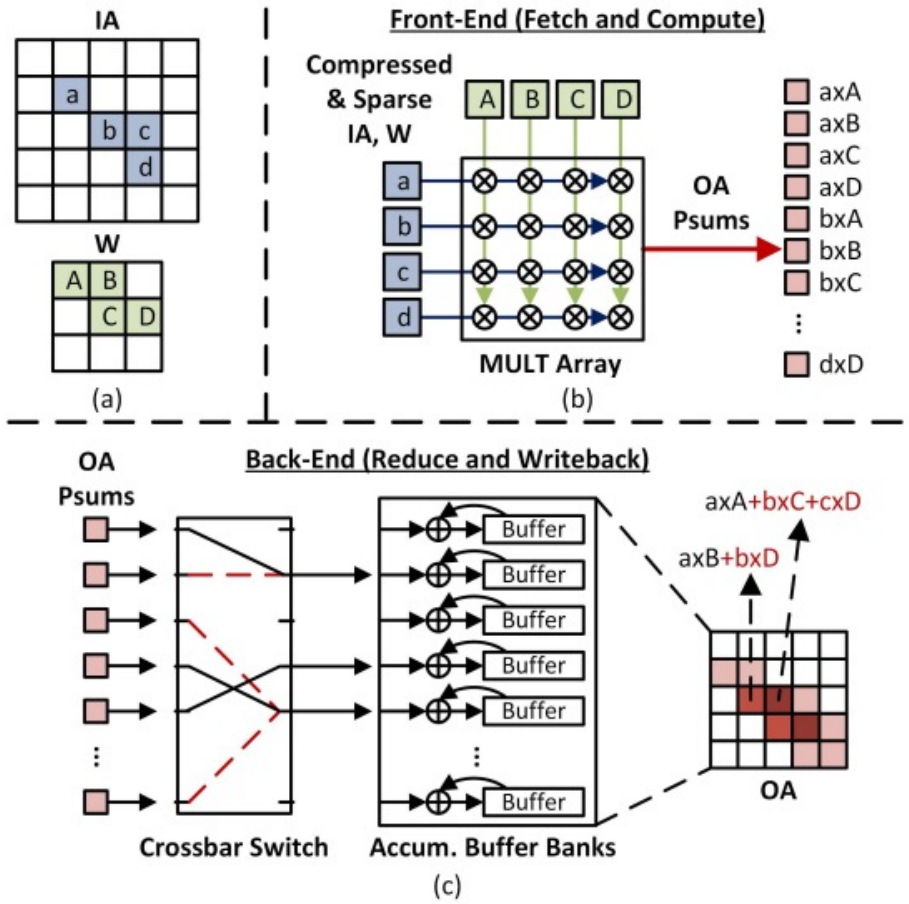


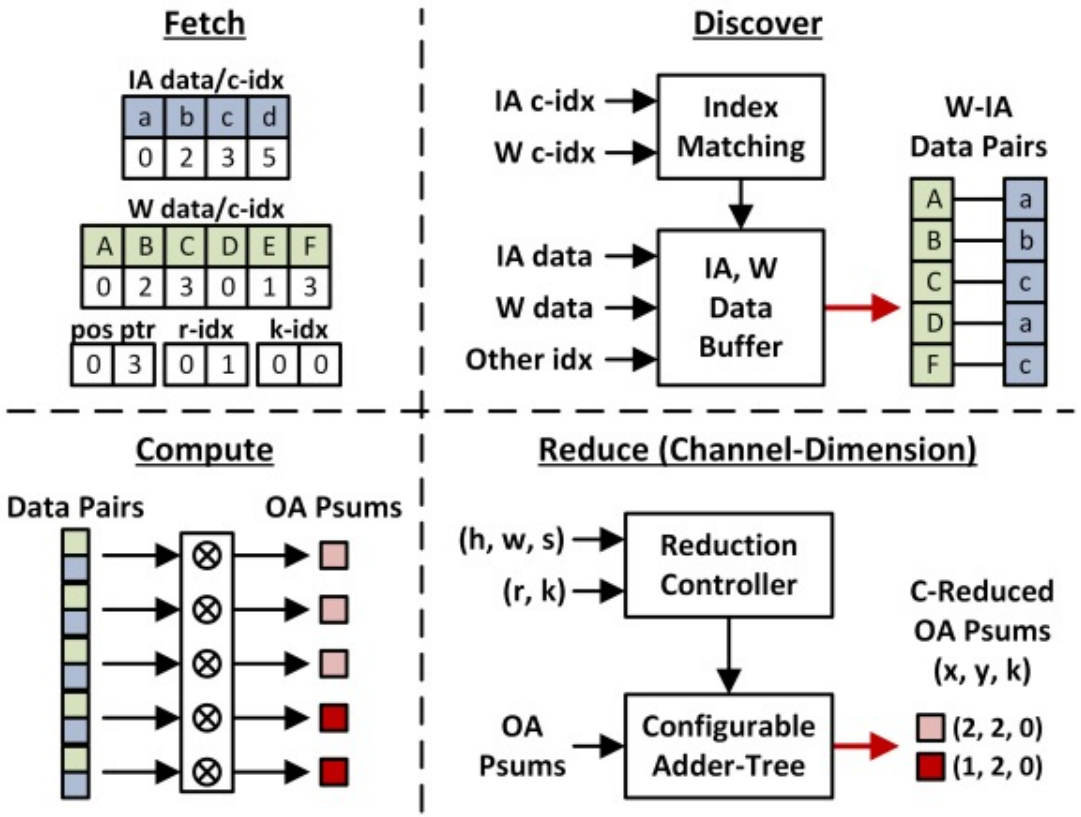




Chen, Yu-Hsin et al. "Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks" IEEE International Solid-State Circuits Conference, ISSCC 2016, Digest of Technical Papers

# SNAP: An Efficient Sparse Neural Acceleration Processor for Unstructured Sparse Deep Neural Network Inference

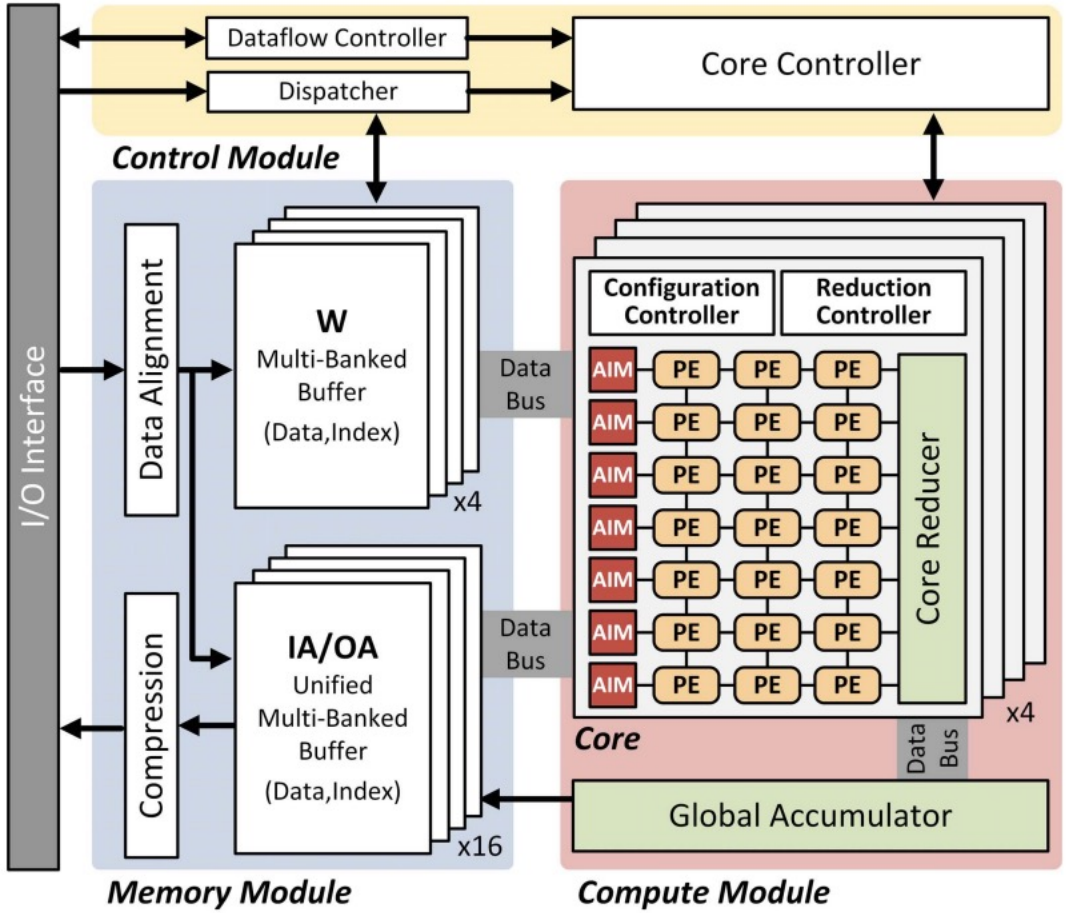
- Channel last vs channel first

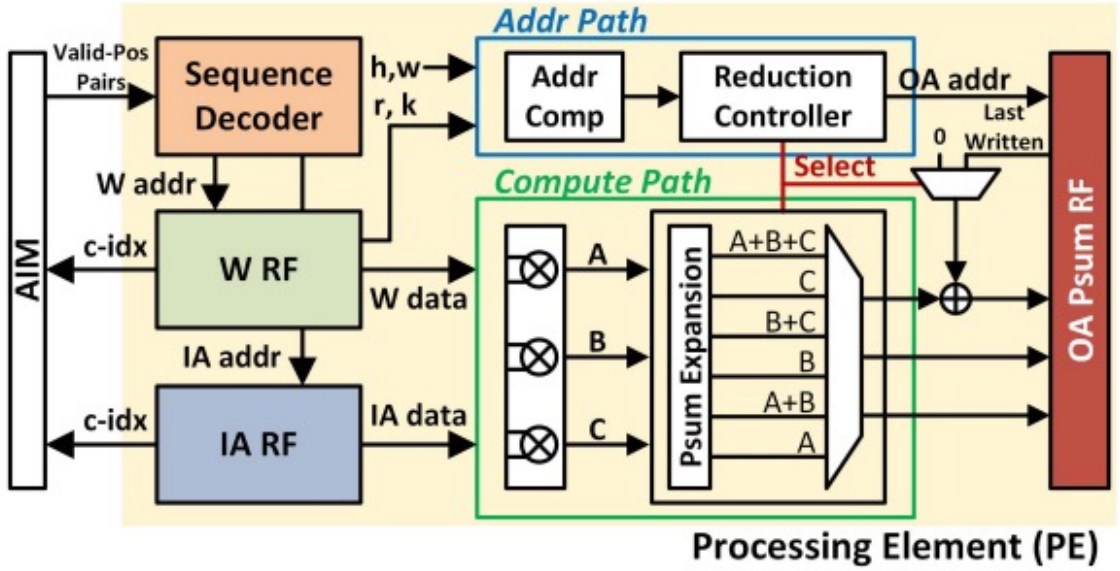



Zhang, Jie-Fang et al. "SNAP: An Efficient Sparse Neural Acceleration Processor for Unstructured Sparse Deep Neural Network Inference" IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 56, NO. 2, FEBRUARY 2021

# SNAP: An Efficient Sparse Neural Acceleration Processor for Unstructured Sparse Deep Neural Network Inference




Zhang, Jie-Fang et al. "SNAP: An Efficient Sparse Neural Acceleration Processor for Unstructured Sparse Deep Neural Network Inference" IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 56, NO. 2, FEBRUARY 2021

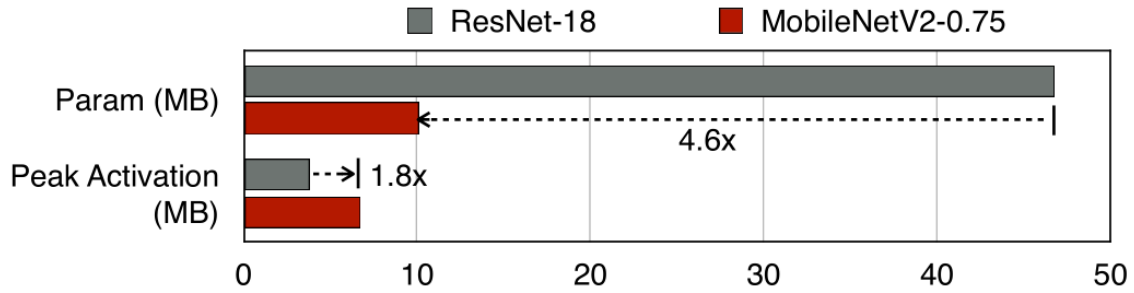
# Outline

- Overview
- AI on Chip (AIoC) team work flow
- Quantization
- Testing System
- Hardware design
  - NPU
  - ASIC (Application Specific Integrated Circuit)
- **System-hardware co-design**

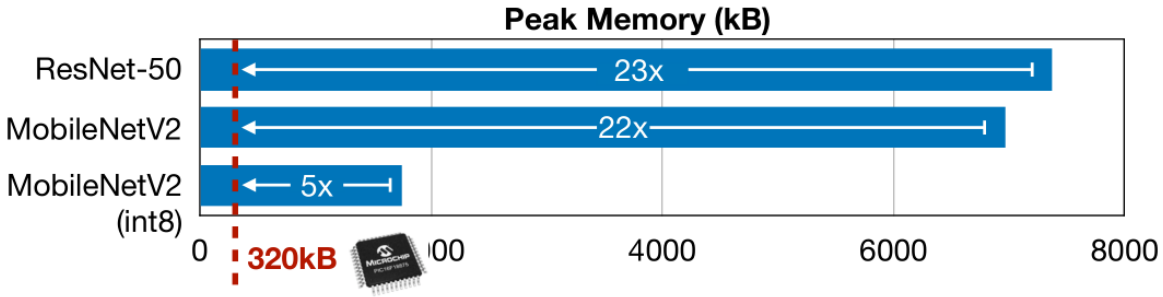
# MCUNet: Tiny Deep Learning on IoT Devices

## System-Algorithm Co-design

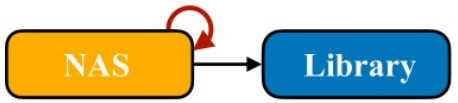
- Existing Methods Reduce Model Size, but not the Activation Size



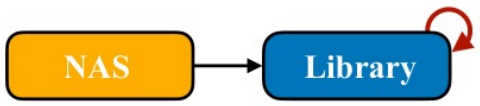

- Existing network **CANNOT** fit the tight memory constraints on MCU



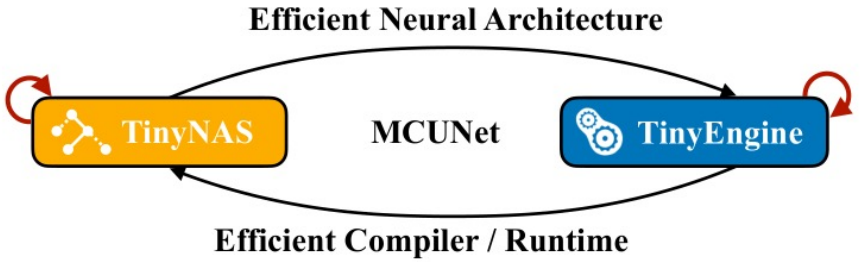

| | ResNet<br>Cloud AI | MobileNet<br>Mobile AI | MCUNet<br>Tiny AI |
|---|---|---|---|
| **Memory** (Activation) | 16GB | 4GB ← 13,000x smaller | 320kB |
| **Storage** (Weights) | ~TB/PB | 256GB | 1MB |




(a) Search NN model on an existing library. e.g., *ProxylessNAS, MnasNet*



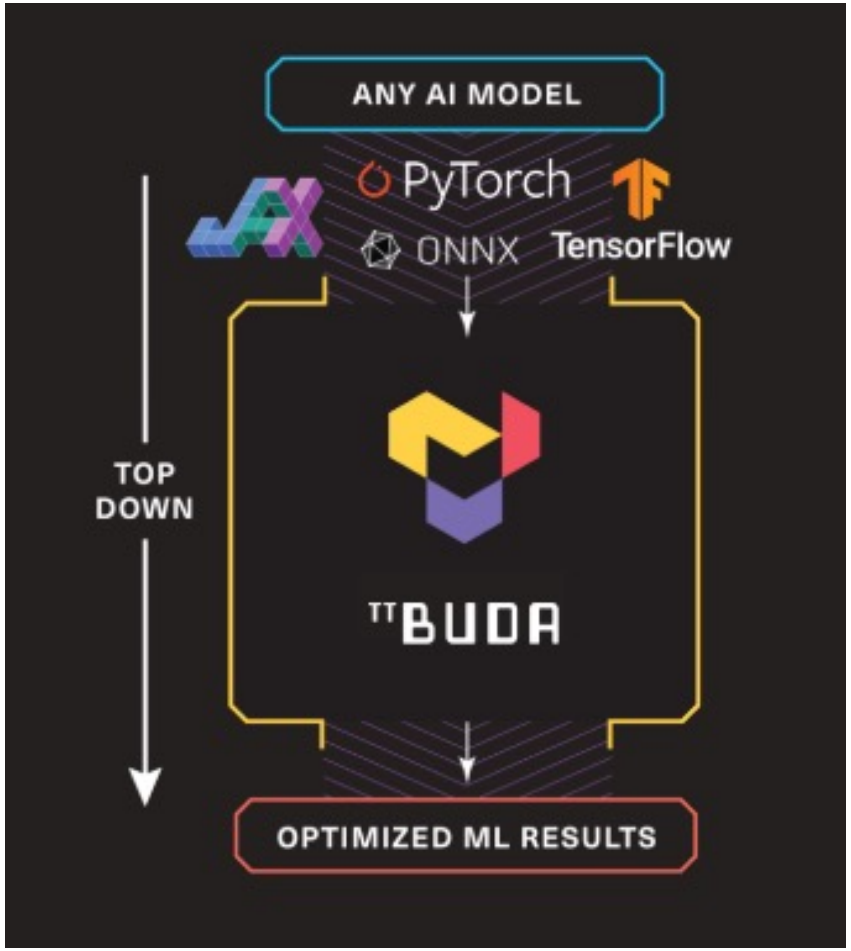

(b) Tune deep learning library given a NN model. e.g., *TVM*

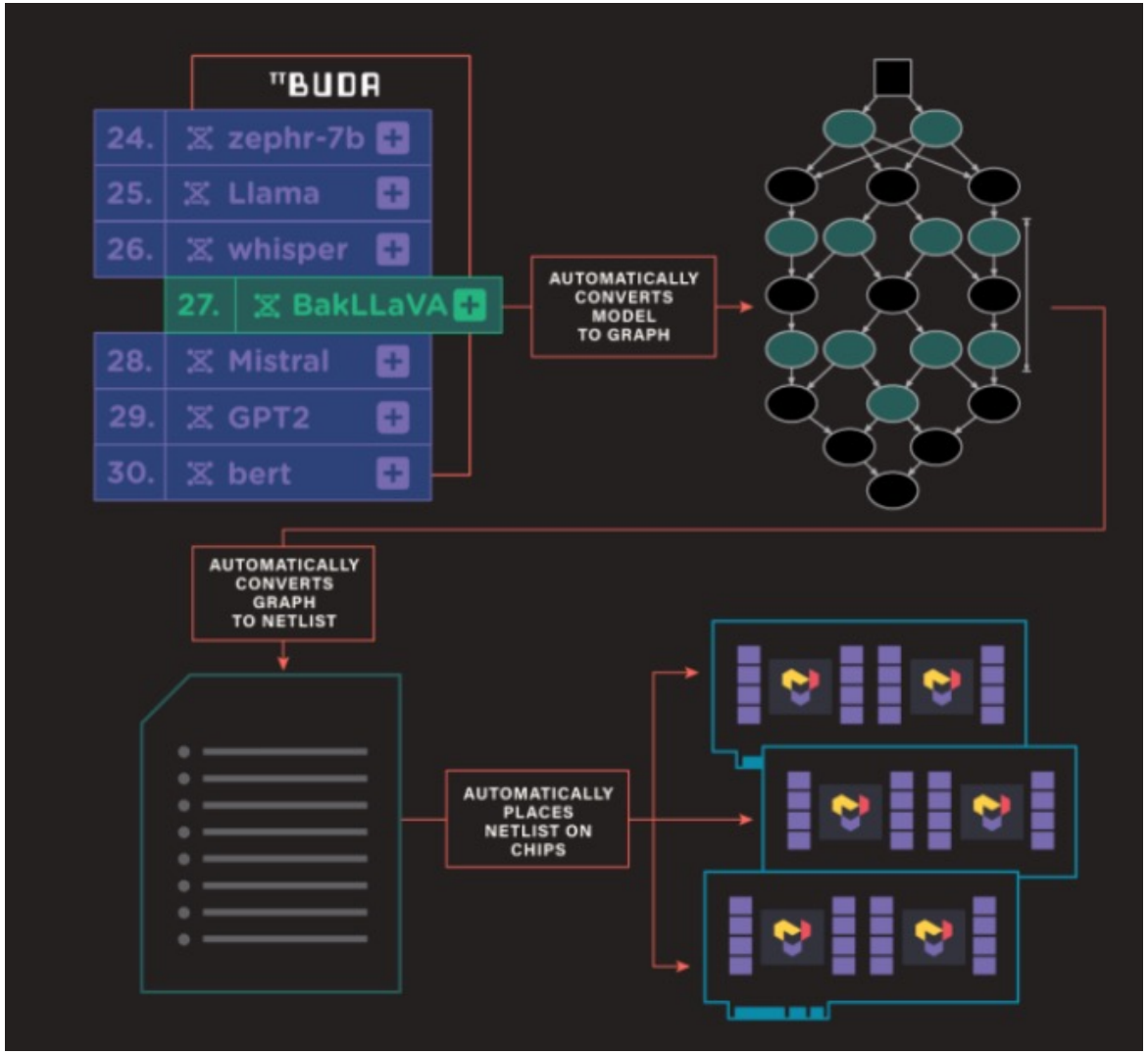



Lin, Ji et al. "MCUNet: Tiny Deep Learning on IoT Devices" NeurIPS 2020

# tenstorrent

## Top down

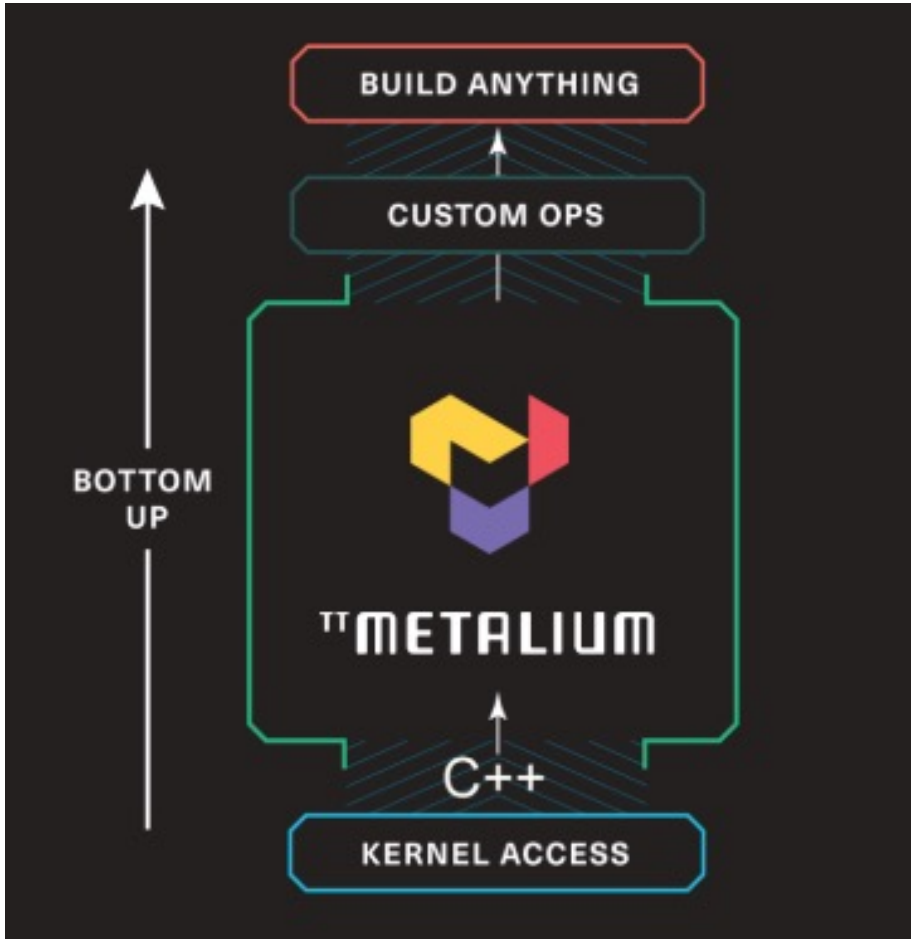


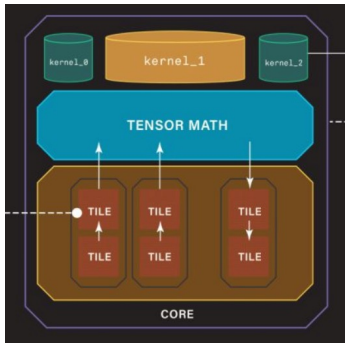

https://tenstorrent.com/

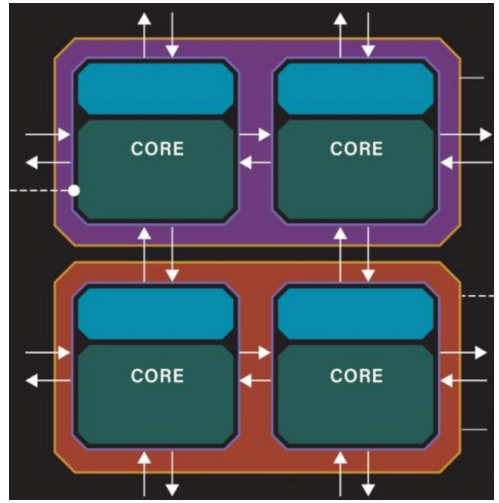
# tenstorrent

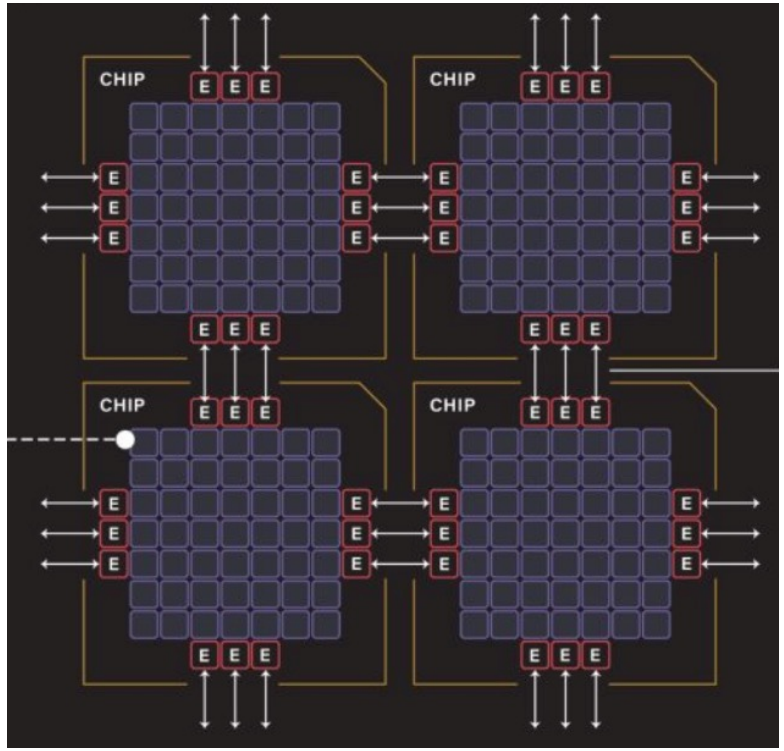
## Bottom Up





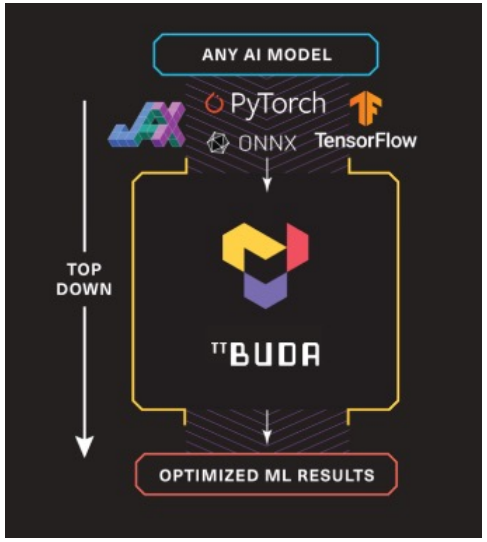


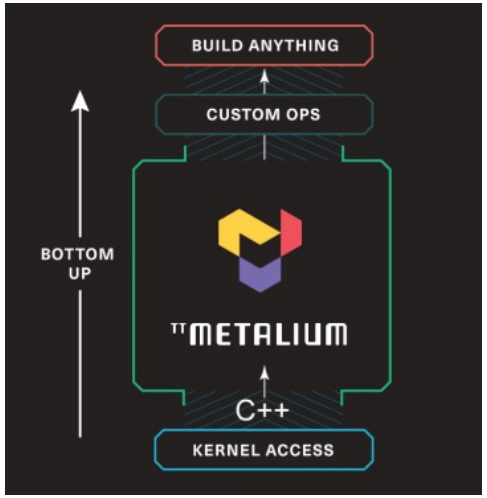



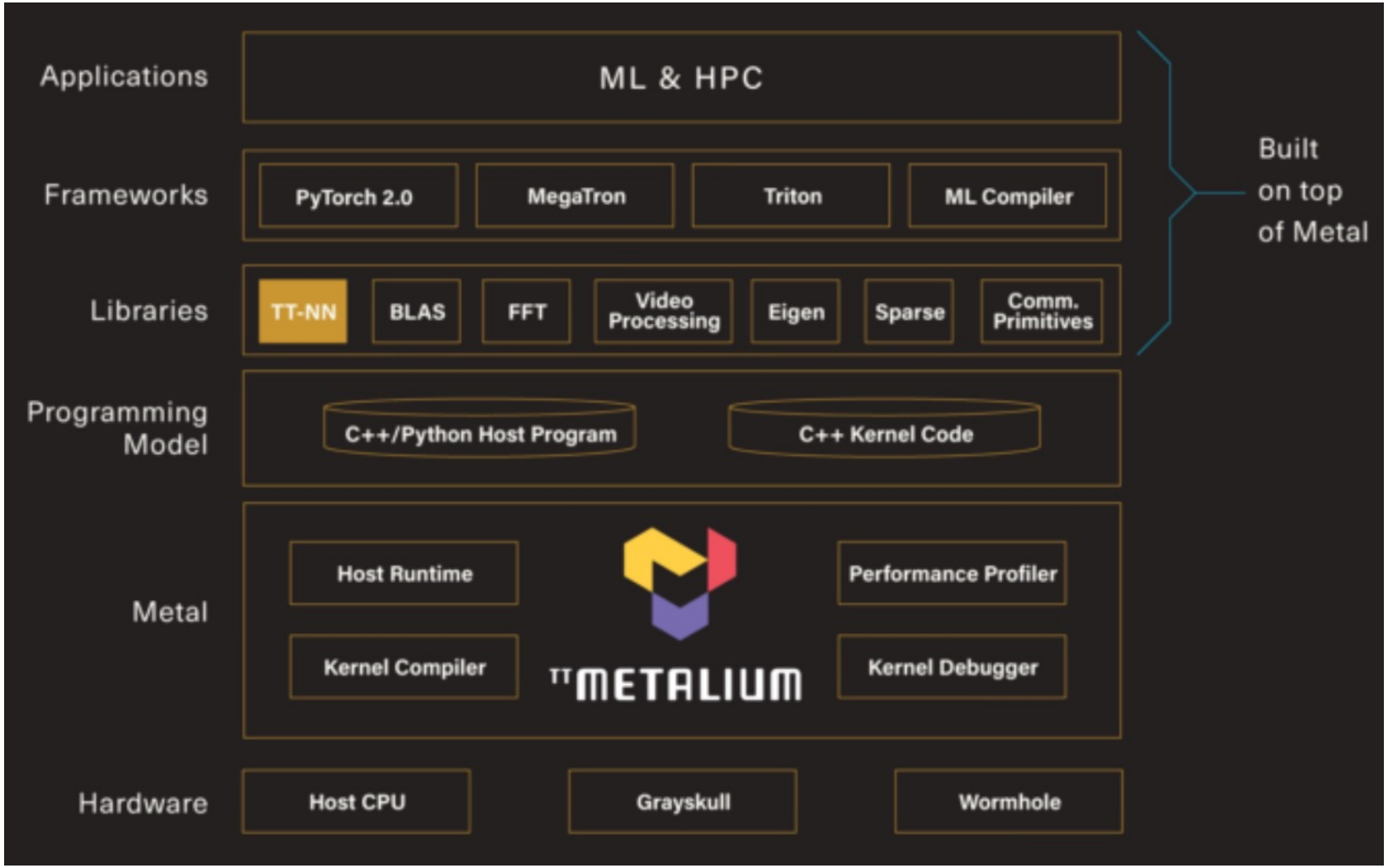



# tenstorrent

## Integration

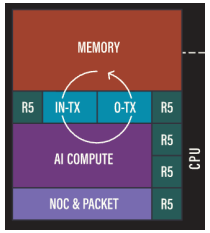


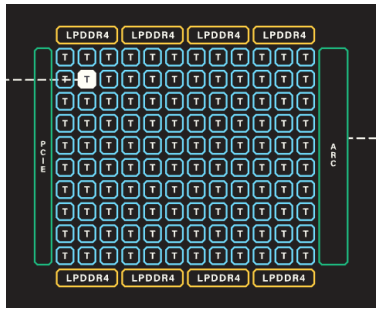



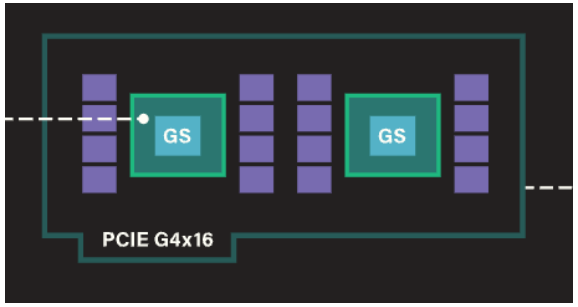



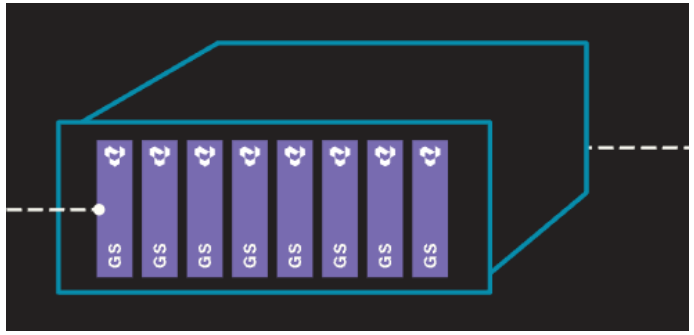
# AI Cloud

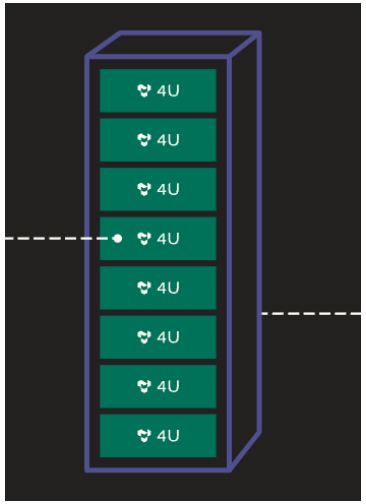
## Scale Up

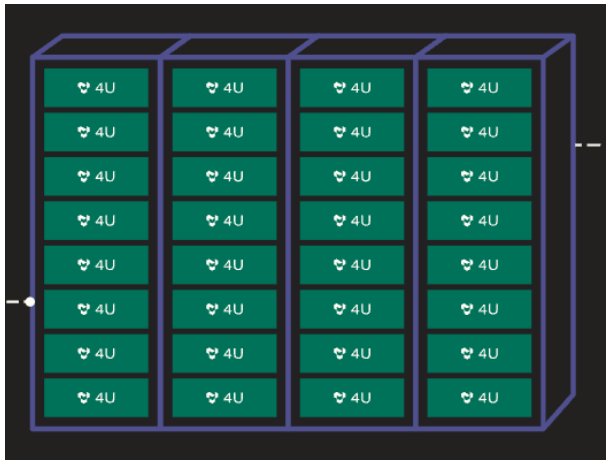


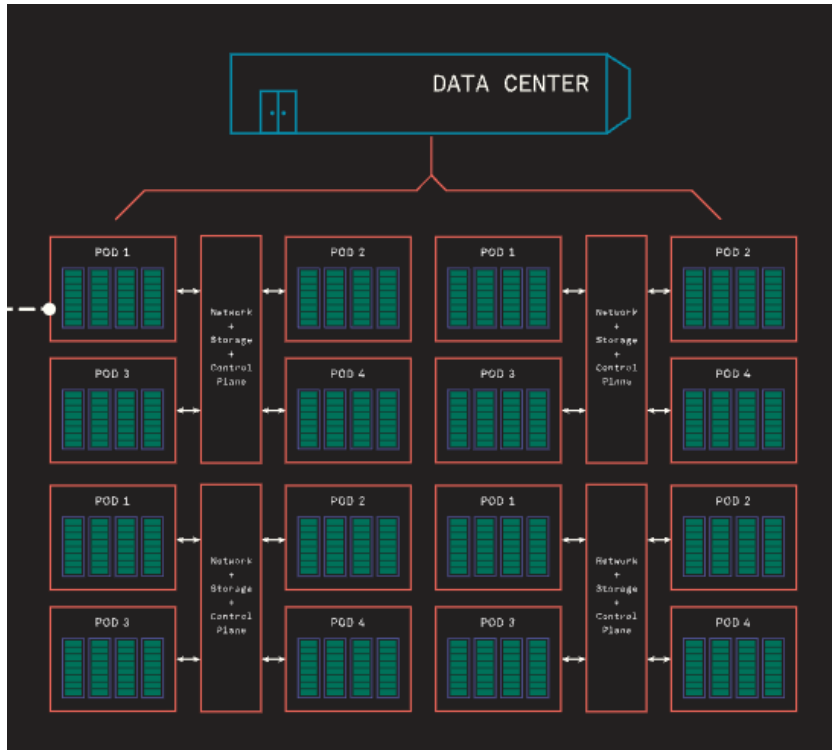



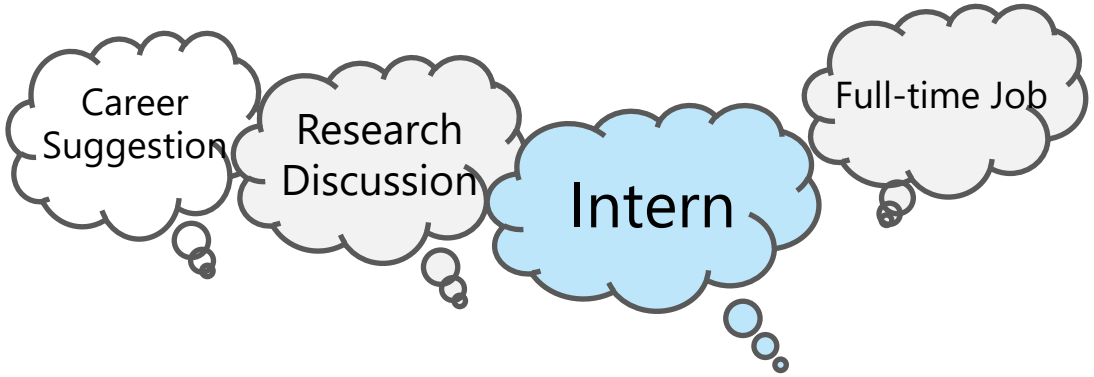






# Welcome to Contact Us!

https://ai.inventec.com

Shang-Jui (Ray) Kuo (郭尚睿)
Inventec AI 研究員
kuo.raysj@inventec.com

Jeng-Lin (John) Li | 李政霖
Inventec 資深資料科學家
li.johncl@inventec.com

Chia-Ching (Jacob) Lin | 林家慶
Inventec 資深資料科學家
lin.jacob@inventec.com

Inventec

# Thank you!

Inventec