# Развёртывание нейронной сети на платформе Zynq UltraScale+

Дмитрий Шадрин,
Инженер по применению Xilinx

Макро Групп – официальный партнёр Xilinx в России

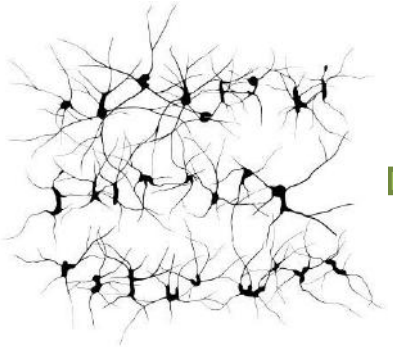
# План вебинара

- Виды нейронных сетей и сферы их применения;
- Методы повышения производительности нейронных сетей;
- Обзор продуктов Xilinx для развёртывания нейронных сетей;
- Обзор поддерживаемых нейронных сетей ;
- Перспективные аппаратные решения;
- Пример развёртывания готовой нейронной сети на базе отладочной платы ZCU104.

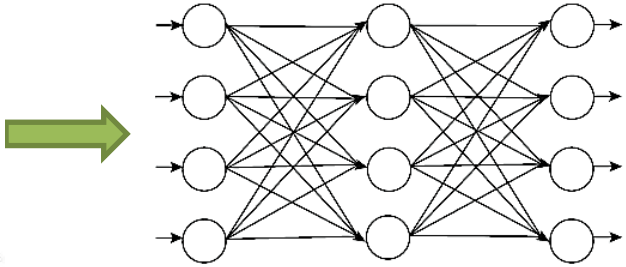
# 1. Виды нейронных сетей и сферы их применения

# Виды нейронных сетей и сферы их применения

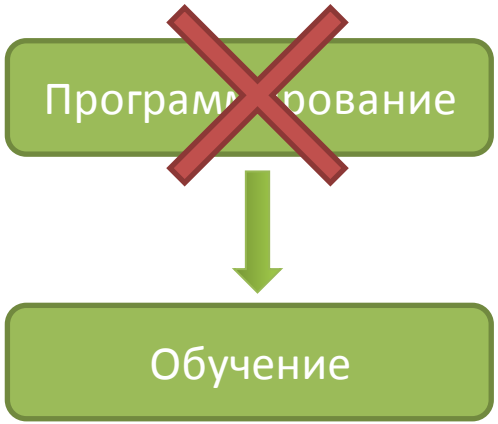
Нейронная сеть – это математическая модель, а также её программная или аппаратная реализации, построенная по принципам функционирования биологических нейронных сетей — сетей нервных клеток живого организма.



Нервные клетки



Нейронная сеть



# Виды нейронных сетей и сферы их применения

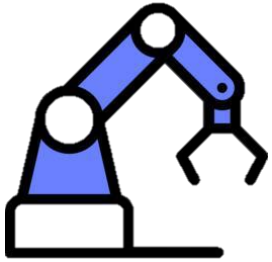
Основными сферами применения нейронных сетей в мире являются:

Экономика

Медицина

Транспорт

Телекоммуникации



Производство

Безопасность

Обработка данных

# Виды нейронных сетей и сферы их применения

Xilinx делает упор на 2 сферы применения нейронных сетей:

Безопасность

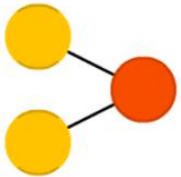
- Классификация объектов
- Детектирование и распознавание лиц
- Детектирование и распознавание позы человека

Транспорт

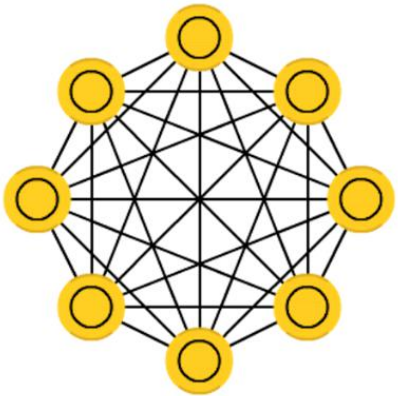
- Семантическая сегментация
- Определение дорожной разметки
- Обнаружение пешеходов и автомобилей
- Распознавание номерных знаков автомобиля

# Виды нейронных сетей и сферы их применения

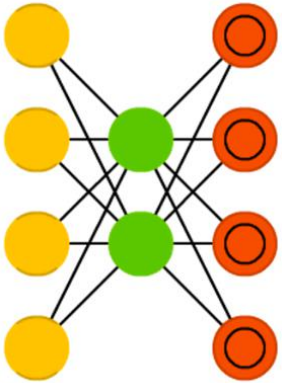
Виды нейронных сетей:



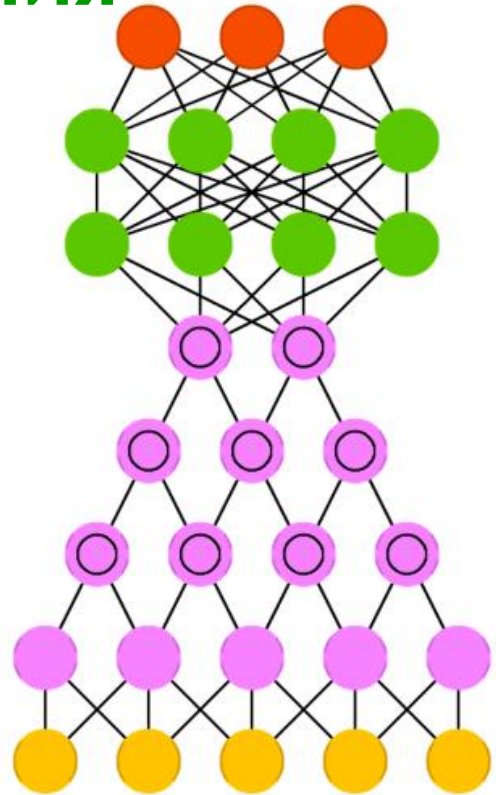
Перцептрон



Сеть Хопфилда



Автокодировщик

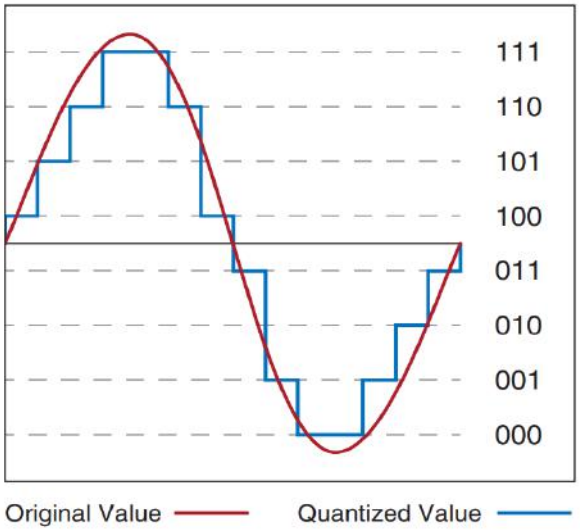


Свёрточная нейронная сеть

# Методы повышения производительности нейронных сетей

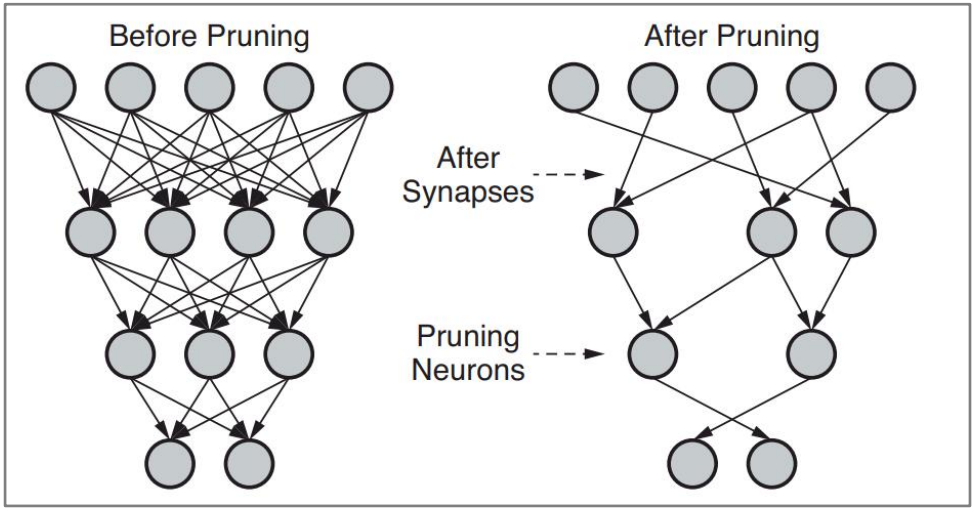
Квантование

Float32 → Int8




Прореживание (pruning)

Удаление связей



# Обзор продуктов Xilinx для развёртывания нейронных сетей

DNNDK (Deep Neural Network Development Kit)

Особенности:

- Решение с полным стеком для разработки приложений глубокого обучения;
- Полный набор оптимизированных инструментов;
- API для программирования на C / C ++.

Инструменты:

# Обзор продуктов Xilinx для развёртывания нейронных сетей

| | |
|---|---|
| DECENT | Прореживание (pruning) и квантование нейронных сетей |
| DNNC | Компилятор для DPU – процессора глубокого обучения |
| DNNAS | Сборка инструкций DPU в двоичный код ELF |
| N2Cube | Загрузчик приложений DNNDK |
| Simulator | Симулятор DPU |
| Profiler | Сбор и визуализация данных |

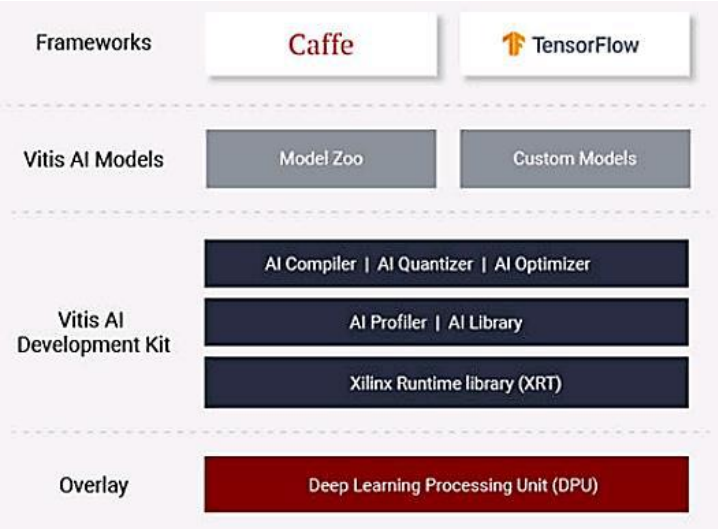
# Обзор продуктов Xilinx для развёртывания нейронных сетей

Vitis AI

Особенности:

- Набор предварительно оптимизированных моделей;
- Большая коллекция инструментов для работы с свёрточными сетями;
- Унифицированные высокоуровневые API C ++ и Python для максимальной переносимости;
- Настраивает эффективные и масштабируемые IP-ядра

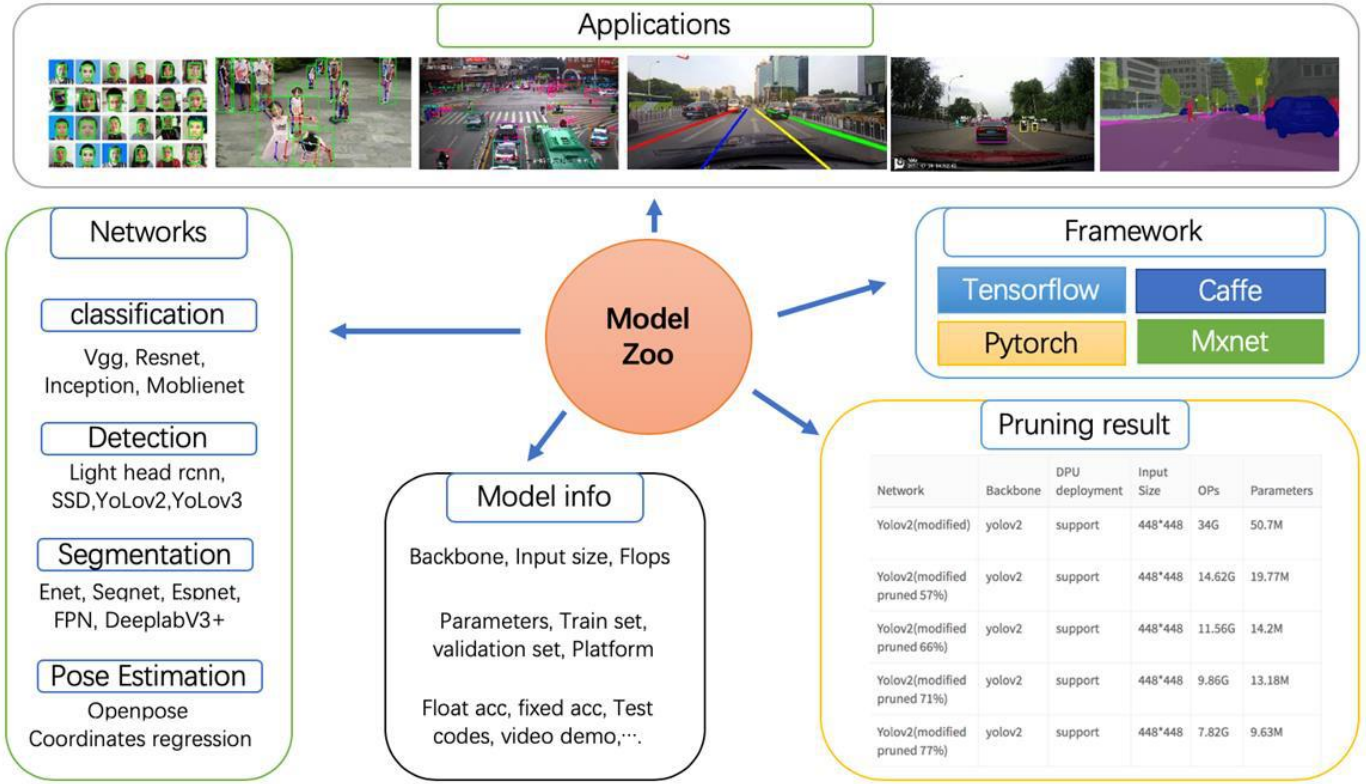
Инструменты:



# Обзор продуктов Xilinx для развёртывания нейронных сетей

| | |
|---|---|
| DPU | Deep-Learning Processor Unit – инструмент для работы с глубокими нейронными сетями |
| AI Optimizer | Инструмент прореживания (pruning) нейронных сетей |
| AI Quantizer | Инструмент квантования нейронных сетей |
| AI Compiler | Компилятор моделей и сборщик инструкций |
| AI Profiler | Профилирование и визуализация приложений |
| AI Library | Набор высокоуровневых библиотек и API для DPU |
| AI Runtime | Загрузчик приложений |

# Обзор продуктов Xilinx для развёртывания нейронных сетей



# Обзор поддерживаемых нейронных сетей

| Применение | Задача | Алгоритм |
|---|---|---|
| Общее | Классификация изображений | Resnet50, Inception v1, BN-inception, VGG16, SqueezeNet, MobilenetV2 |
| | Обнаружение объекта | MobilnetV2-SSD, SSD, YOLO v2, YOLO v3, Tiny YOLO v2, Tiny YOLO v3 |
| | Сегментация | ENet, ESPNet |
| Лицо | Обнаружение лица | SSD, Densebox |
| | Распознавание лица | ResNet + Triplet / A-softmax Loss |
| | Распознавание атрибутов лица | Classification and regression |
| Пешеход | Обнаружение пешеходов | SSD |
| | Оценка позы | Coordinates Regression |

# Обзор поддерживаемых нейронных сетей

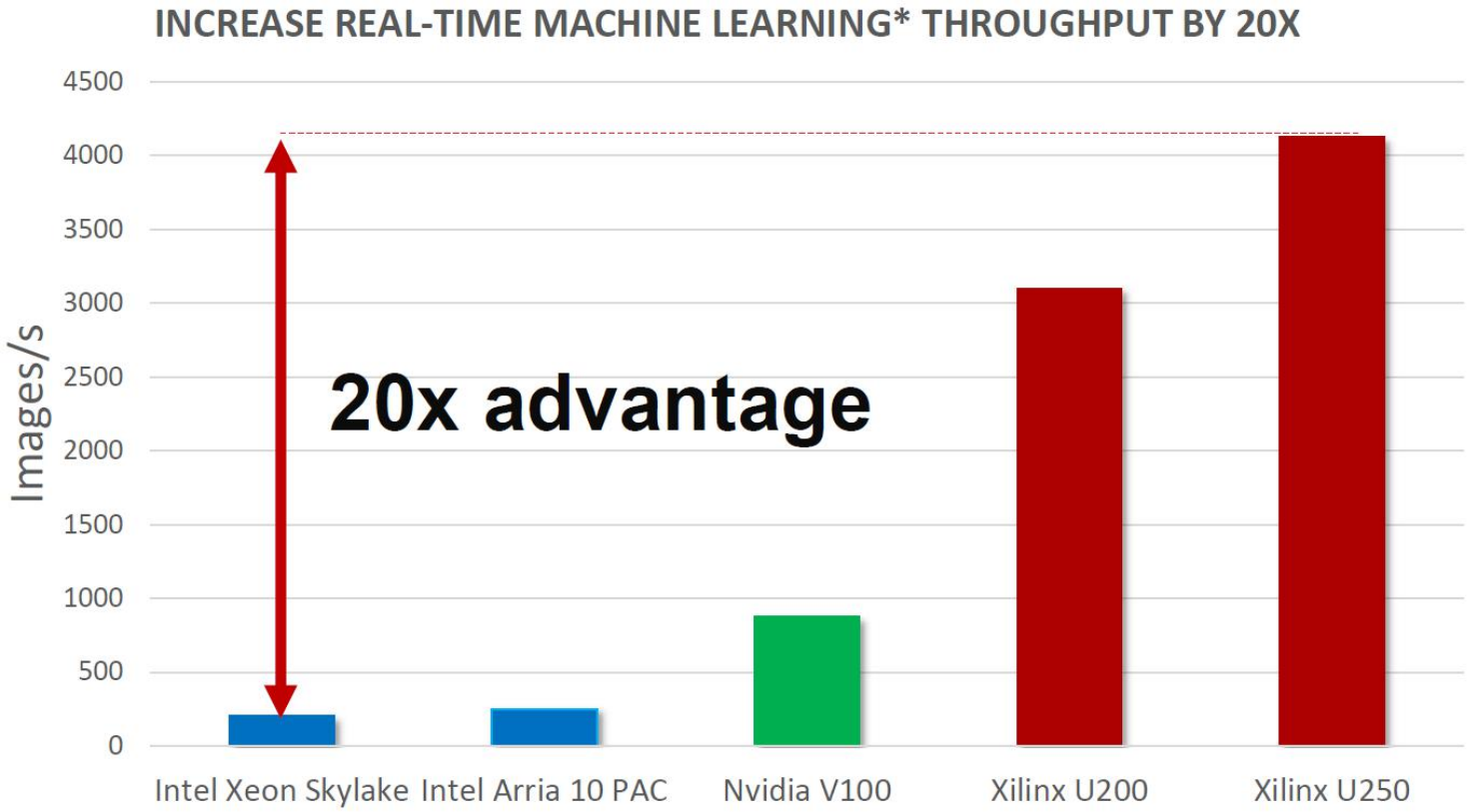
| Применение | Задача | Алгоритм |
|---|---|---|
| Видео Аналитика | Обнаружение объекта | SSD, RefineDet |
| | Распознавание пешеходов | GoogleNet |
| | Распознавание автомобильных атрибутов | GoogleNet |
| | Распознавание логотипа автомобиля | Модифицированный Densebox + GoogleNet |
| | Обнаружение номерного знака | Модифицированный DenseBox |
| | Распознавание номерных знаков | GoogleNet + Многозадачное обучение |
| ADAS/AD | Обнаружение объекта | SSD, YOLOv2, YOLOv3 |
| | Обнаружение переулка | VPGNet |
| | Семантическая сегментация | FPN |

# Перспективные аппаратные решения

## Ускорительные карты Alveo

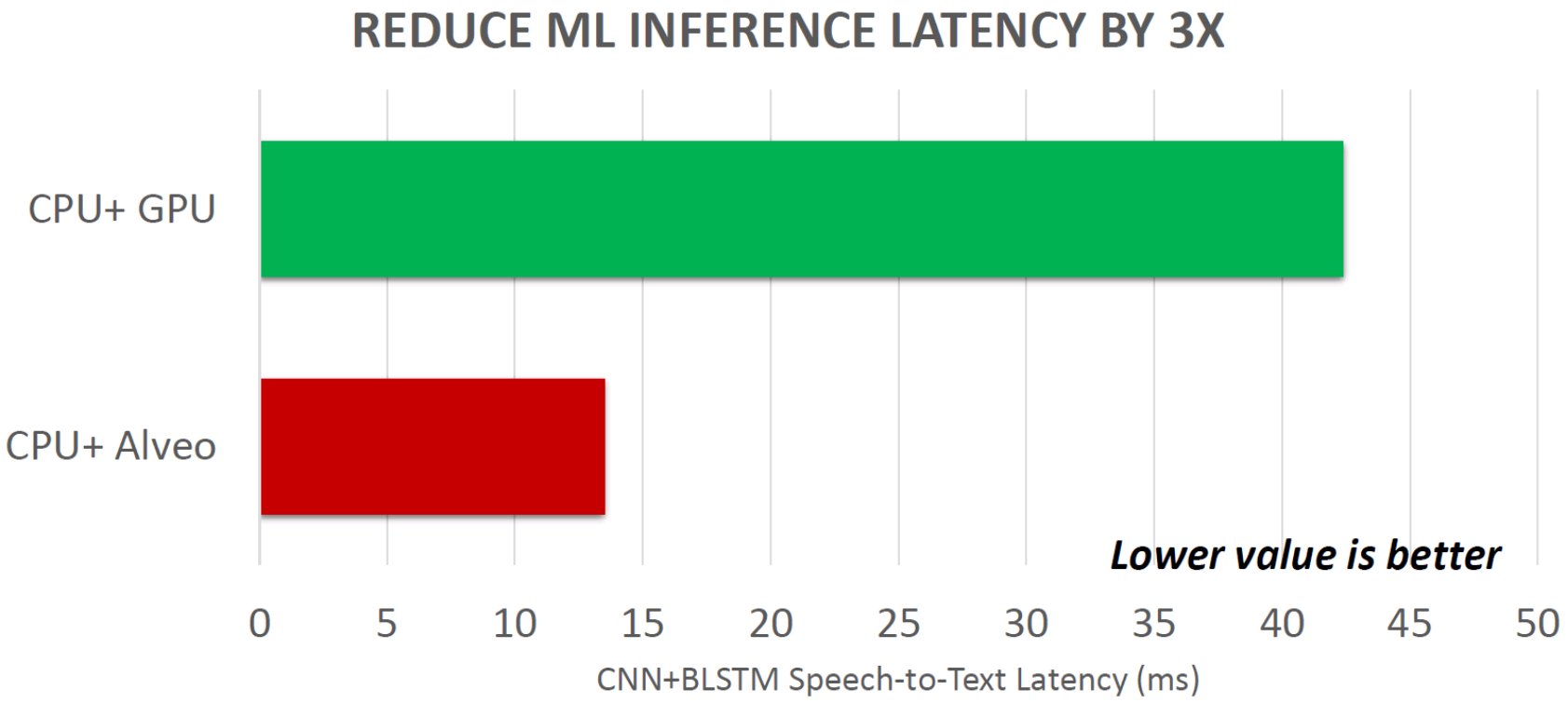

# Перспективные аппаратные решения

## Ускорительные карты Alveo



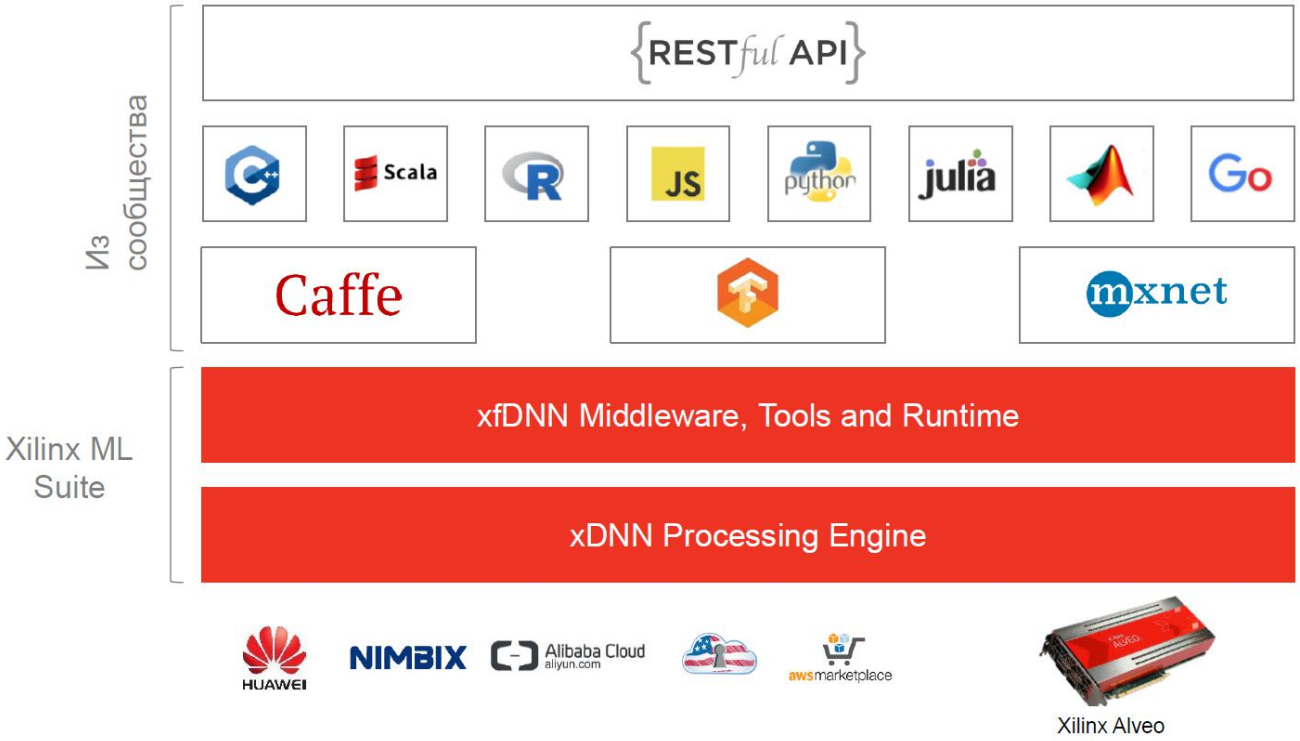
# Перспективные аппаратные решения

## Ускорительные карты Alveo



# Перспективные аппаратные решения

## Ускорительные карты Alveo

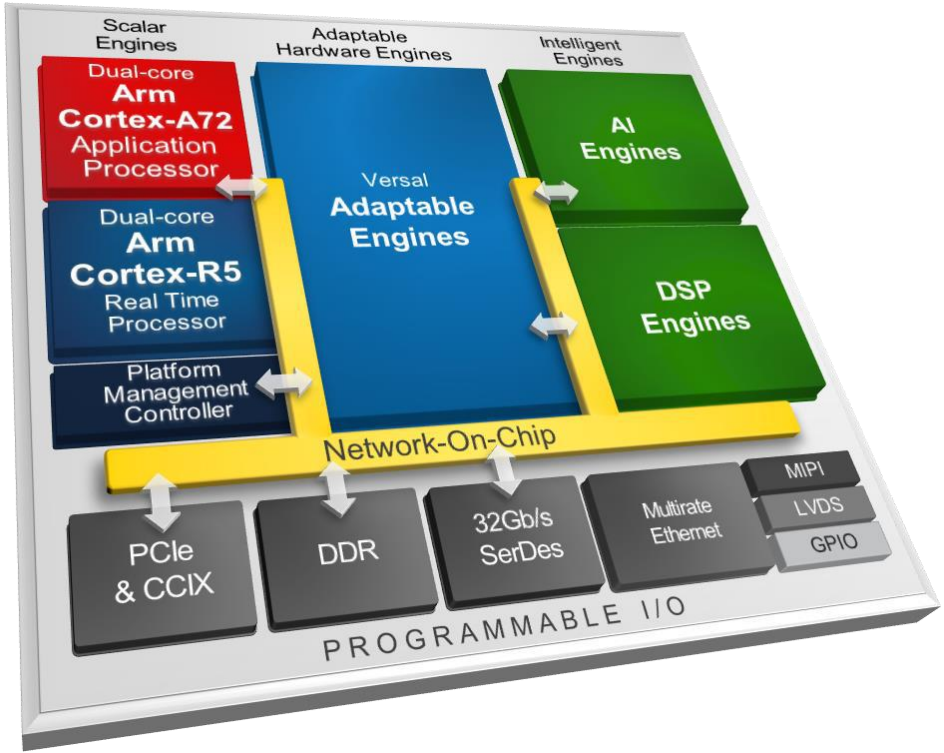



- вывод быстрее в 4 раза, чем на GPU
- высочайшая производительность на ватт
- лёгкие в работе ML-фреймворки и API

# Перспективные аппаратные решения

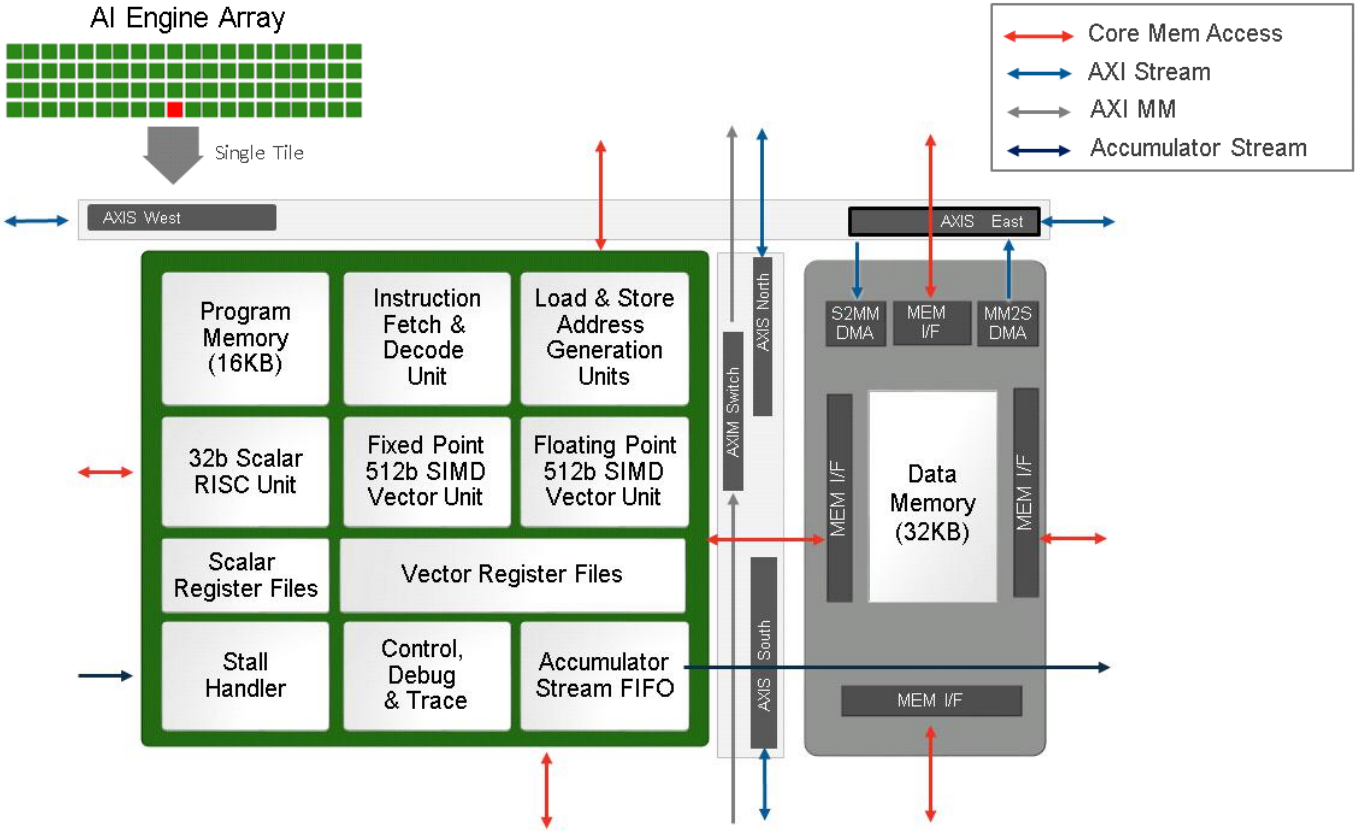
## Новая архитектура Versal ACAP

- Scalar Engines
  - Arm® Cortex™-A72 APU
  - Arm Cortex-R5 RPU
- Adaptable Engines
  - CLBs
  - Internal Memory
- Intelligent Engines
  - AI Engine
  - DSP Engine
- Connectivity
  - PCIe w/CCIX
  - Ethernet
  - DDR Memory Controllers
  - Transceivers
  - I/O
- Platform Resources
  - Network-On-Chip
  - Platform Management Controller



# Перспективные аппаратные решения

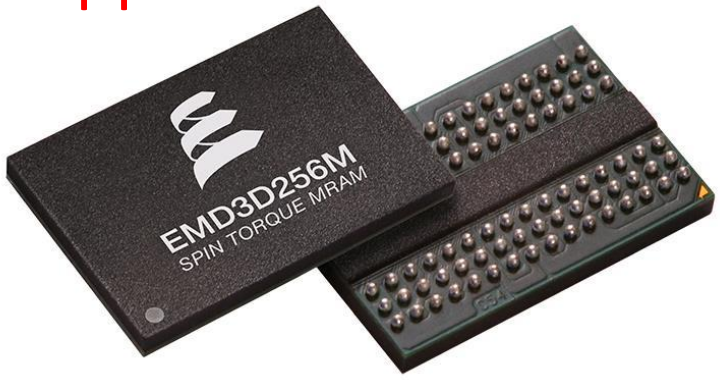
## Новая архитектура Versal ACAP

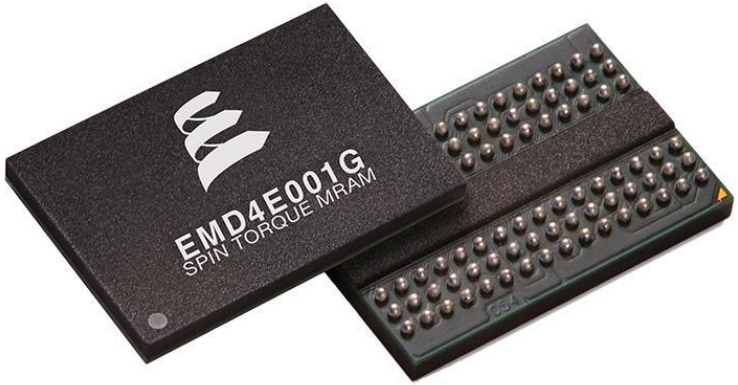


# Перспективные аппаратные решения

## Everspin STT-MRAM для Xilinx FPGA

- Энергонезависимость
- Объем 256 Мбит и 1 Гбит
- 256 Мбит организация (32Mb x 8, 16Mb x 16)
- 256Мбит: VDD = 1.5v +/- 0.075v
- 1 Гбит организация (128Mb x8, 64Mb x16)
- 1 Гбит: VDD = VDDQ = 1.2v
- Тактовая частота 667MHz
- Интерфейс ST-DDR3 и ST-DDR4
- Корпус 78-BGA и 96-BGA



# Перспективные аппаратные решения

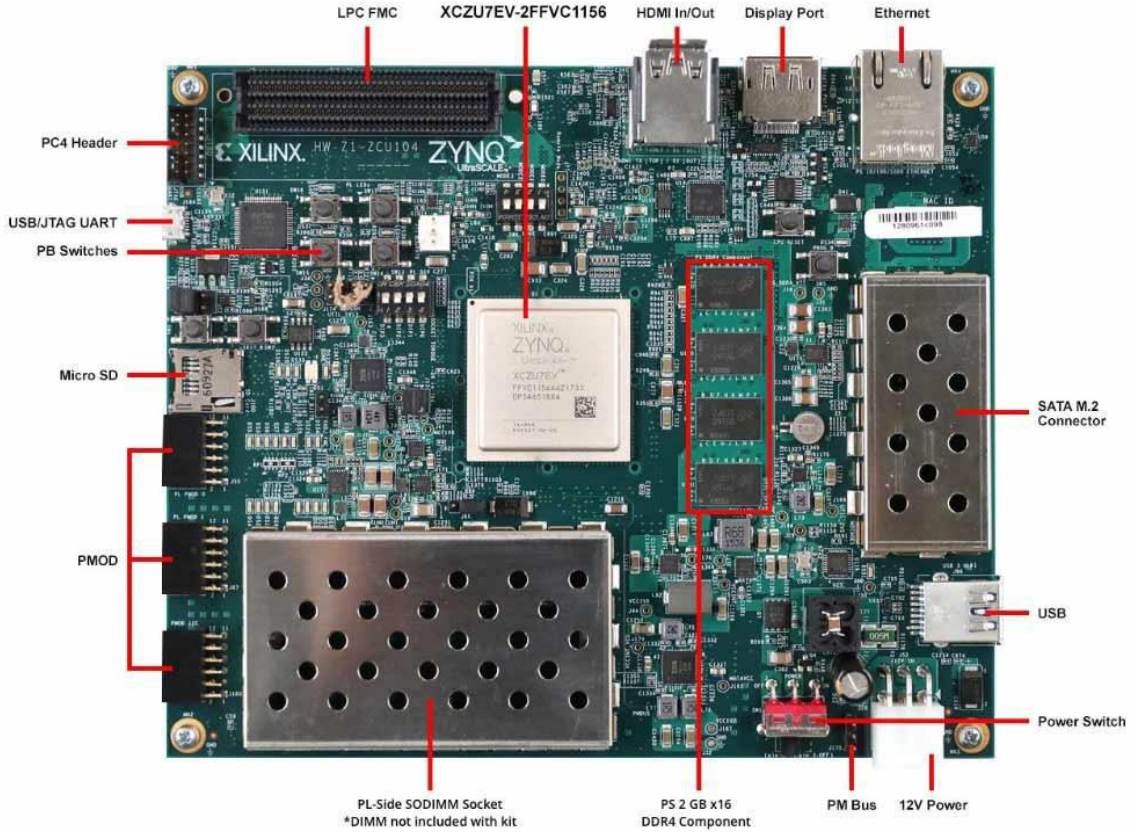
## Everspin nvNITRO Storage Accelerator

# Перспективные аппаратные решения

## Everspin nvNITRO Storage Accelerator

| |  |  |
|---|---|---|
| Form Factor | PCIe (HHHL) | U.2 (15mm) |
| Interface | PCIe Gen3 x8 | PCIe Gen3 x4 |
| Capacity | 1GB, | 1GB |
| Protocol/Access Modes | NVMe 1.2.1 & Direct Memory Access (PCIe MMIO, CMB, P2Pmem, PMR) | |
| Performance IOPs (R/W) (4K Random R/W) | 1.5M / 1.5M (x8 PCIe) | 750K / 750K |
| Latency (R/W) QD=1 | 5.1µS (Read) / 5.9µS (Write) | |
| BER / Data Retention | < 1 $e^{-18}$ / Powered down DR is 3+ months @ 50C, Powered up DR is lifetime at full operating temperature | |
| Endurance | 1$e^{9}$ Access to each and every page, Unlimited uniform access for 10+ years | |

# Пример развёртывания нейронной сети
## на базе отладочной платы ZCU104




Подробное описание процесса установки в документе UG1354

# Пример развёртывания нейронной сети
## на базе отладочной платы ZCU104

Настройка загрузки платы с microSD-карты

Необходимое ПО для Windows:

Xming

Tera Term

# Спасибо за внимание!

Компания Макро Групп:

- официальный партнер Xilinx
- комплексная поставка электронных компонентов
- техническая поддержка по всем вопросам применения продукции и ПО Xilinx
- контрактное производство электроники

Обращайтесь:


- Dmitriy.Shadrin@macrogroup.ru
- Dmitry.Khorkov@macrogroup.ru
- fpga@macrogroup.ru


МАКРО ГРУПП

# Ссылки на материалы

Инструкция по установке и запуску (GitHub)

UG1354-Vitis AI Library User Guide

Образ для ZCU104

ZCU104 AI Model

Vitis AI Library 1.0

# Команды для терминала

**Задаём IP-адрес платы (через Com-порт):**

```
sudo ip link set eth0 up
sudo ip addr add 169.254.177.200/255.255.0.0 dev eth0
```

**Устанавливаем пакеты:**

```
dpkg -i vitis_ai_model_ZCU104_2019.2-r1.0.deb
dpkg -i vitis_ai_library_2019.2-r1.0.deb
```

**Запуск facedetect:**

```
cd /usr/share/vitis_ai_library/samples/facedetect
./test_video_facedetect densebox_640_360 0 -t 8
```

**Запуск Yolov3 (дорожная ситуация):**

```
cd /usr/share/vitis_ai_library/samples/yolov3
./test_video_yolov3 yolov3_bdd 0 -t 8
```

**Запуск posedetect:**

```
cd /usr/share/vitis_ai_library/samples/posedetect
./test_video_posedetect sp_net 0 -t 8
```

**Запуск classification:**

```
cd /usr/share/vitis_ai_library/samples/classification
./test_video_classification resnet50 0 -t 8
```

**Запуск segmentation:**

```
cd /usr/share/vitis_ai_library/samples/segmentation
./test_video_segmentation fpn 0 -t 8
```