

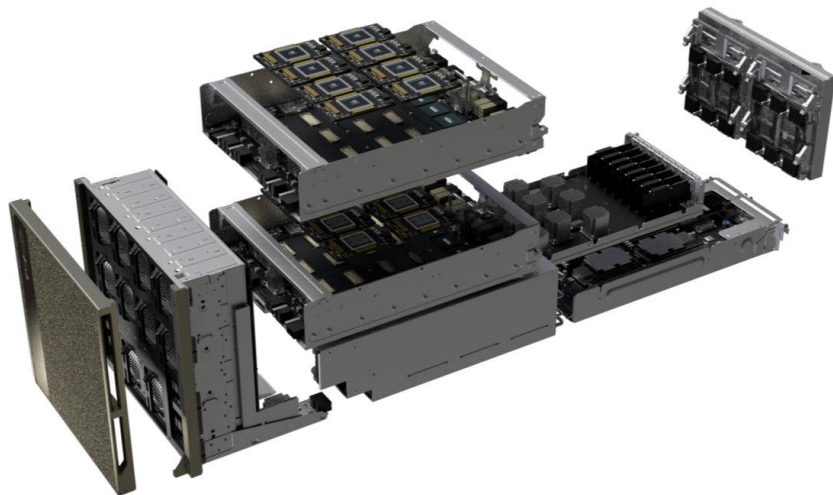


Студенческих практикум

Разработка Ускорителей вычислений

Супер ЭВМ DGX2

Производительность:	2 ПФлопс
Ускорителей NVidia Tesla V100:	16
Память GPU:	512 ГБ
Количество CUDA ядер:	81920
Хранилище данных SSD:	30 ТБ



Студенческие исследовательские
проекты ИИ

$\frac{1}{4}$

Поддержка
учебного процесса в области ИИ

$\frac{1}{4}$

Коммерческие
проекты и фундаментальные
исследования

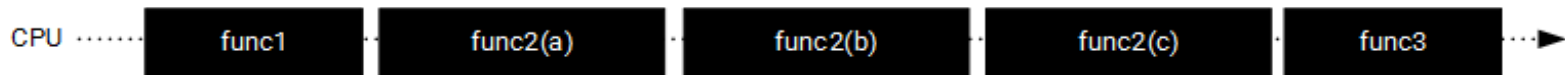
$\frac{1}{2}$

Сравнение принципов ускорения CPU, GPU ~ FPGA

- Как CPU, так GPU имеют predetermined архитектуру с фиксированным количеством ядер, фиксированным набором инструкций и жесткой архитектурой памяти.
- Традиционная разработка программного обеспечения связана с программированием функциональности на заранее определенной архитектуре.
- CPU достигают высокой скорости работы благодаря глубокой конвейеризации, аппаратной предвыборке данных и скорости доступа к памяти.
- Графические процессоры масштабируют производительность за счет количества ядер и использования параллелизма SIMD / SIMT.

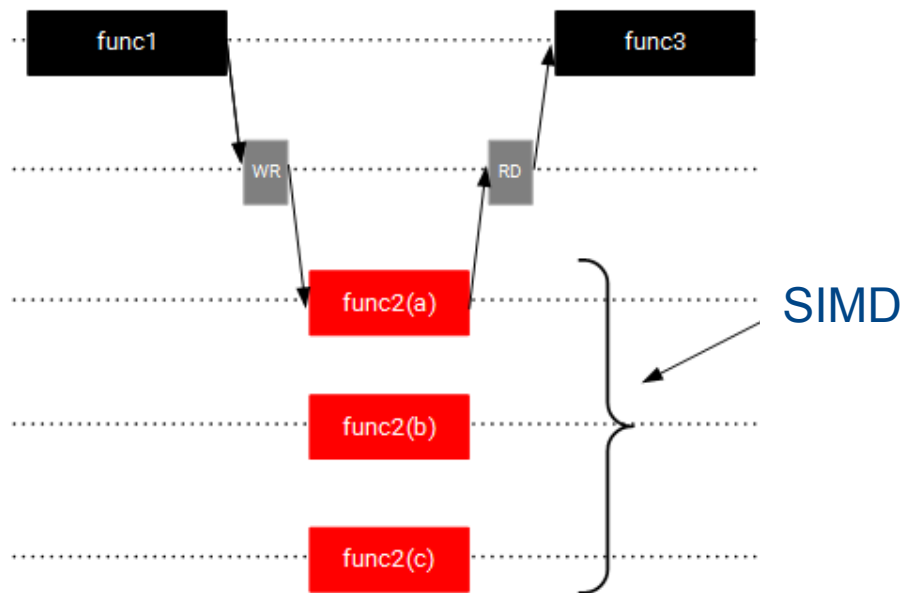
- Программируемые устройства представляют собой полностью настраиваемые архитектуры. Разработчик создает вычислительные блоки, оптимизированные для нужд приложений.
- Производительность достигается за счет создания глубоко конвейерных каналов данных, а не за счет увеличения количества вычислительных единиц.
- Разработка программируемых устройств - это программирование архитектуры для реализации желаемой функциональности.

CPU



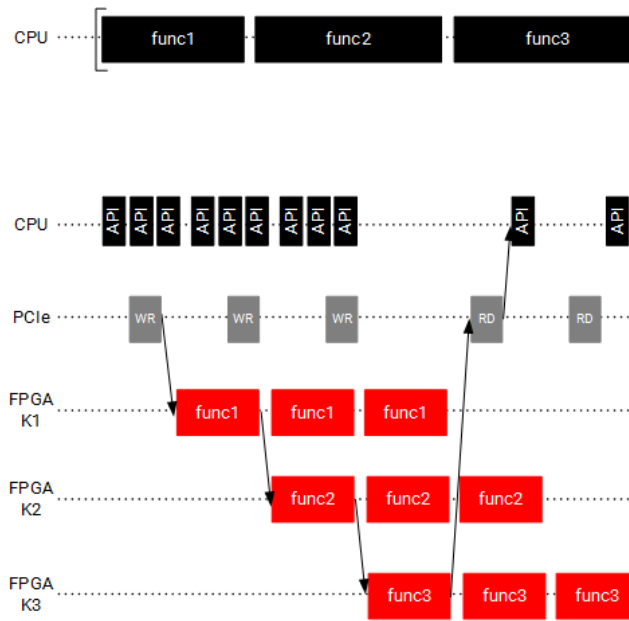
Сравнение принципов ускорения

GPU



Конвейерность,
многопоточность

FPGA



Преимущества FPGA на многих видах задач

AREA	PARTNER WORKLOAD	ALVEO ACCELERATION VS CPU
Database Search and Analytics	BlackLynx Unstructured Data Elasticsearch	90X
Financial Computing	Maxeler Value-at-Risk (VAR) Calculation	89X
Machine Learning	Xilinx Real-Time Machine Learning Inference	20X
Video Processing / Transcoding	NGCodec HEVC Video Encoding	12X
Genomics	Falcon Computing Genome Sequencing	10X

Базы
данных



90x

Финансы



89x

Машинное
Обучение



20x

Видео



12x

HPC & Life
Sciences



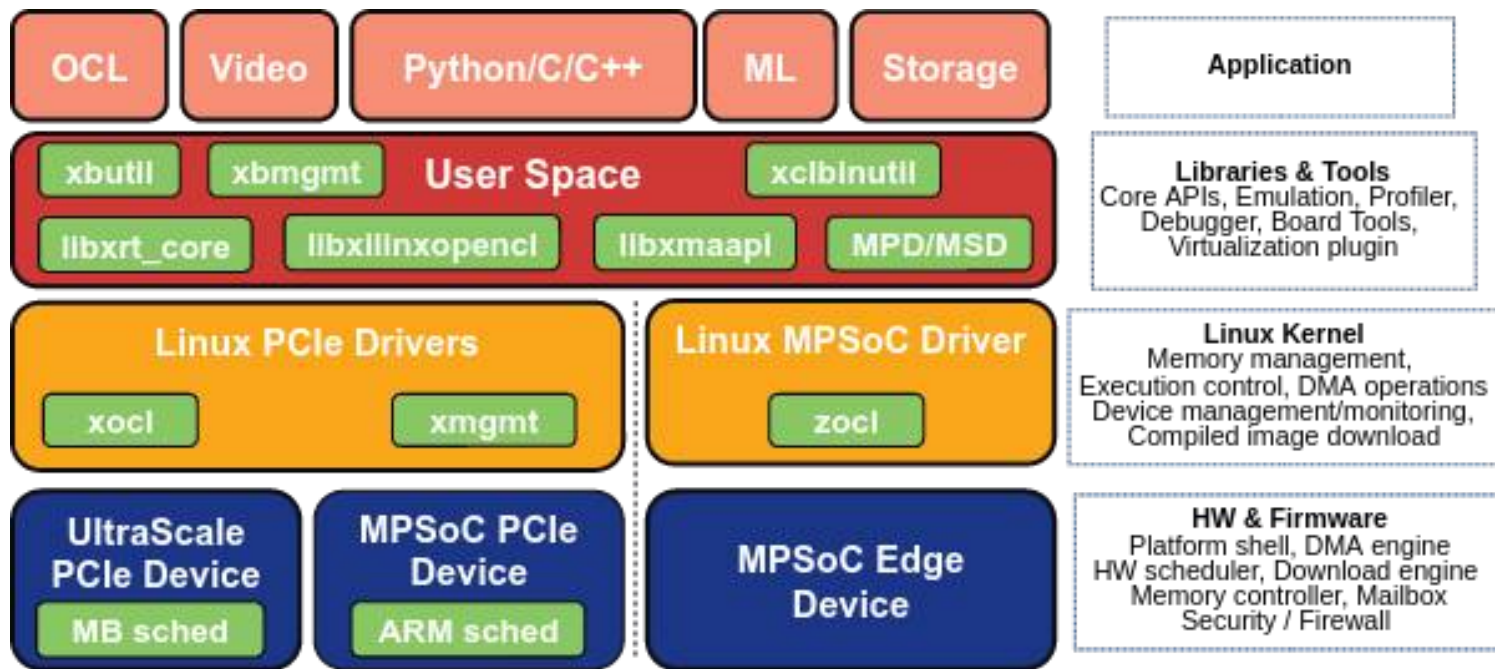
10x

Ускорительная карта Xilinx Alveo U200



- Интерфейс - PCI Express ® Gen3 x16
- Максимальное энергопотребление - 225W
- Память: 4 x 288-pin DDR4 DIMM, каждая планка DIMM 2Gb x 72, ECC, Single rank, Номинал микросхем памяти: MTA18ASF2G72PZ-2G3B1
- Базовая микросхема ПЛИС FPGA: Xilinx ® UltraScale+ XCU200.
- Ресурсы ПЛИС: Look-Up tables (LUTs) (K) - 892; Registers (K) - 1831; 36 Kb block RAMs - 1766; 288 Kb UltraRAMs - 800; DSP slices - 5867

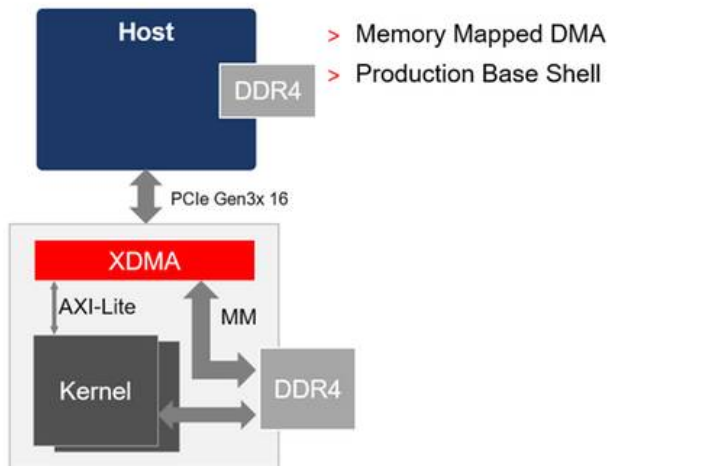
Структура Xilinx RunTime Library



XRT Software Stack

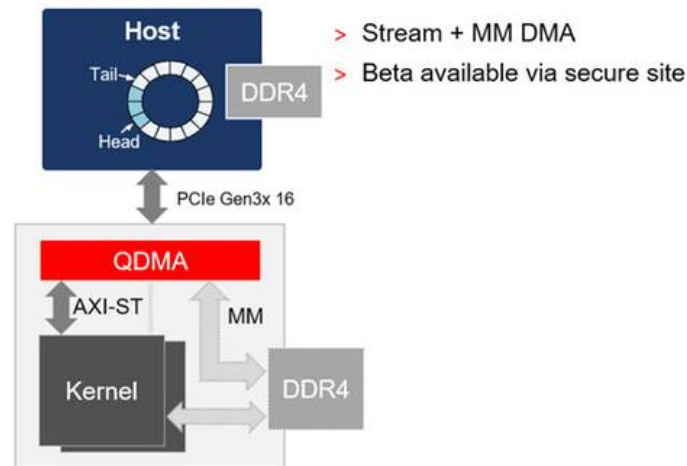
Сравнение версий окружения XRT

Typical XDMA Shell



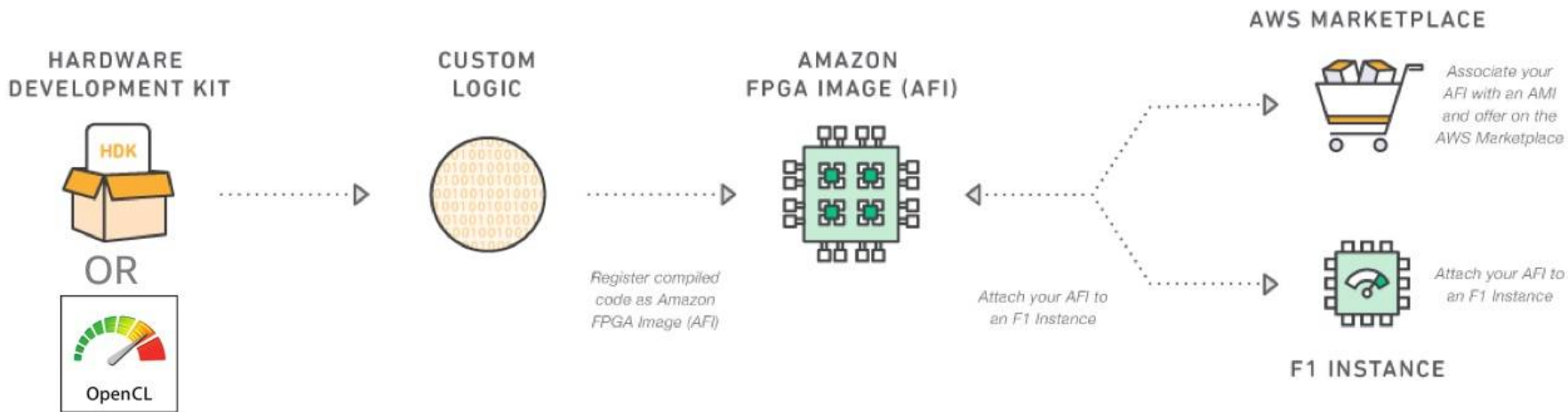
- > FPGA memory accessible by x86 Host
- > Maps to DIMM (off-chip DRAM) and PLRAM (on-chip SRAM)
- > 15 kernel interfaces per SLR
 - > Alveo U200=45 and Alveo U250=60

QDMA Shell



- > Streaming for direct kernel access
- > Optimized for high bandwidth and low latency transfers
- > 48 kernel interfaces and 48 queue sets

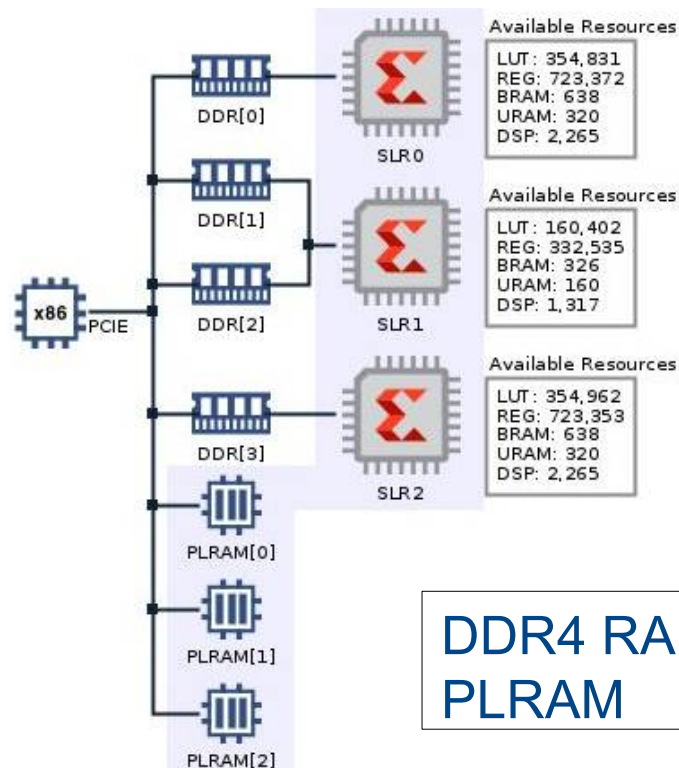
Облачная платформа Amazon AWS EC2



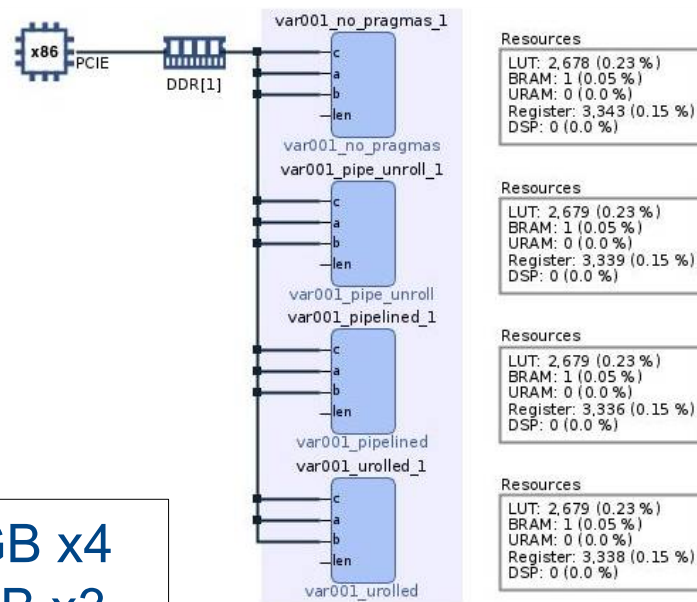
AWS EC2 FPGA Development Kit - это набор инструментов разработки, моделирования, отладки, компиляции и запуска приложений с аппаратным ускорением на узлах Amazon EC2 F1.

Облачная платформа Amazon AWS EC2

Структурная схема Alveo U200

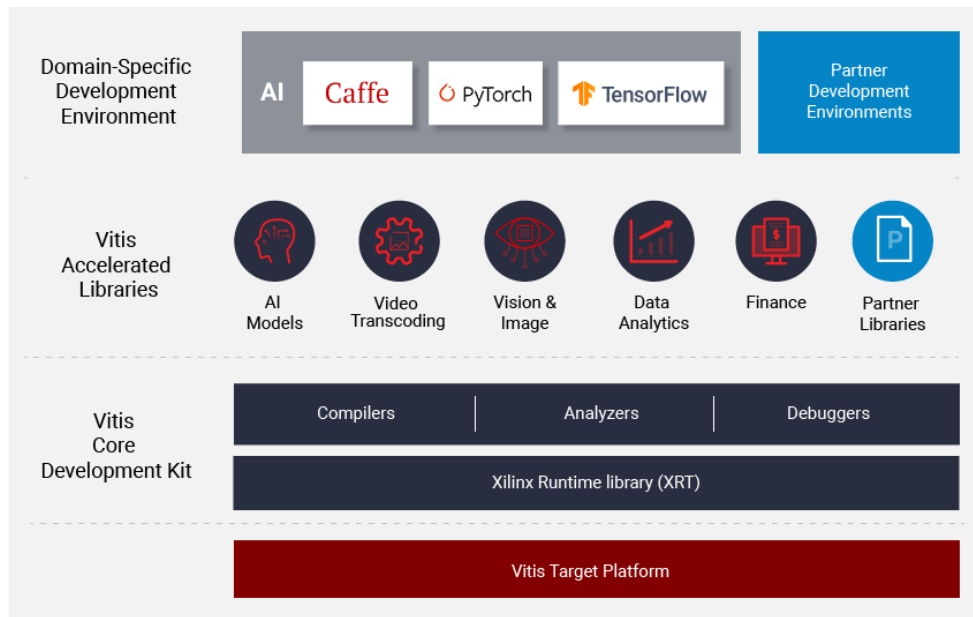


Пример ускорителя

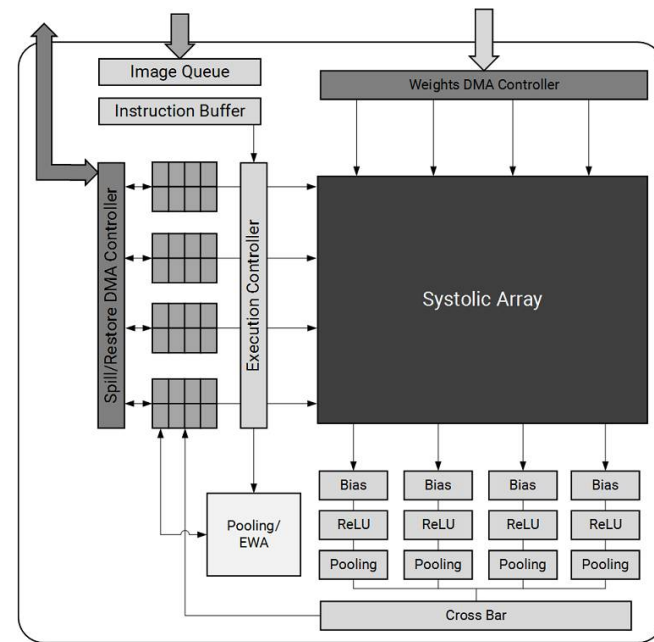


Vitis AI : платформа для ИИ на FPGA

Vitis™ Unified Software Platform



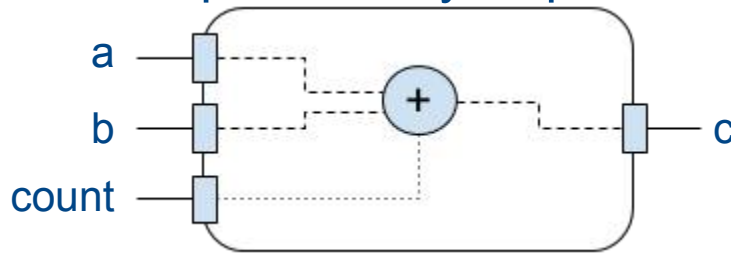
Архитектура xDNN ядра Xilinx (DPUCADX8G)




XZ4609-091/020












Пример разработки устройства ускорения с Xilinx Vivado High Level Synthesis Tool

1. Разработать ускоритель



jupyter 1-building-with-vitis Last Checkpoint: 25.02.2021 (autosaved)  Logout

File Edit View Insert Cell Kernel Help Trusted Python 3 C

       Run    Markdown 

```
In [1]: %%writefile vadd.c

void vadd(int* in_a, int* in_b, int* out_c, int count) {
    #pragma HLS INTERFACE m_axi port=in_a offset=slave
    #pragma HLS INTERFACE s_axilite port=in_a bundle=control
    #pragma HLS INTERFACE m_axi port=in_b offset=slave
    #pragma HLS INTERFACE s_axilite port=in_b bundle=control
    #pragma HLS INTERFACE m_axi port=out_c offset=slave
    #pragma HLS INTERFACE s_axilite port=out_c bundle=control
    #pragma HLS INTERFACE s_axilite port=count bundle=control
    #pragma HLS INTERFACE s_axilite port=return bundle=control
    for (int i = 0; i < count; ++i) {
        *out_c++ = *in_a++ + *in_b++;
    }
}
```

Пример разработки устройства ускорения с Xilinx Vivado High Level Synthesis Tool

2. Подключить окружение платформы Alveo

```
In [2]: ► import glob  
platform = glob.glob("/opt/xilinx/platforms/*/*.xpfm")[0]
```

3. Синтезировать kernel ускорителя

```
In [3]: ► !v++ -c vadd.c -t hw --kernel vadd -f $platform -o vadd.xo
```

3. Выполнить линковку нескольких ускорителей

```
In [4]: ► !v++ -l -t hw -o vadd.xclbin -f $platform vadd.xo
```

Пример разработки устройства ускорения с Xilinx Vivado High Level Synthesis Tool

5. Запустить вычисления на ускорителе

```
import pynq
import numpy as np

ol = pynq.Overlay('vadd.xclbin')

vadd = ol.vadd_inst1

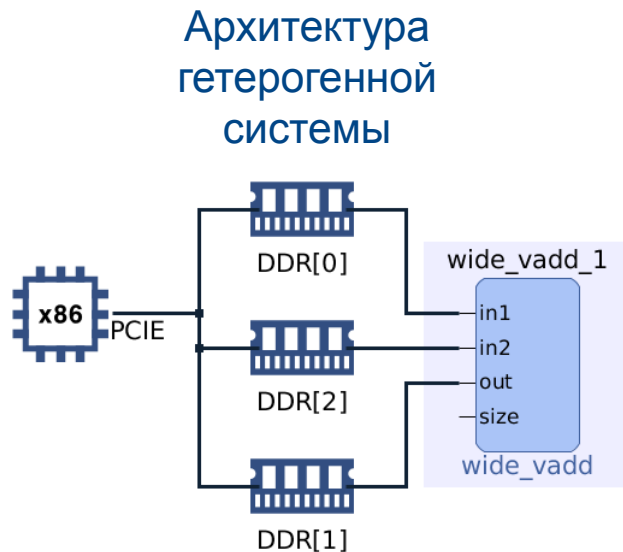
in1 = pynq.allocate((10000000,), 'u4')
in2 = pynq.allocate((10000000,), 'u4')
out = pynq.allocate((10000000,), 'u4')

in1[:] = np.random.randint(low=0, high=1000000, size=(10000000,), dtype='u4')
in2[:] = 200

in1.sync_to_device()
in2.sync_to_device()

vadd.call(in1, in2, out, 10000000)

out.sync_from_device()
np.array_equal(in1 + in2, out)
```



Практикум

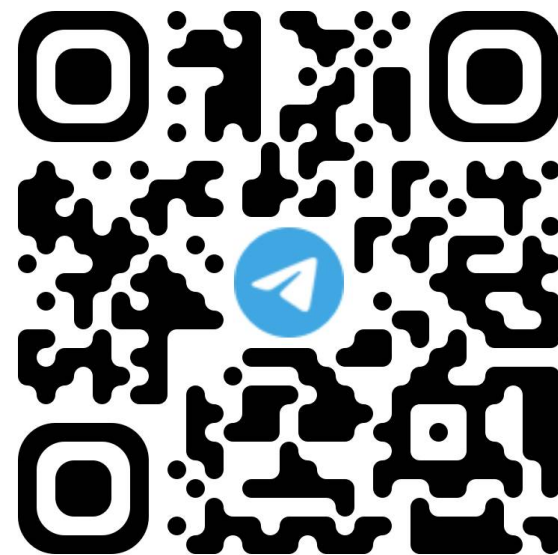
8 ноября — 7 декабря

- I. Лабораторная работа №1 Методология разработки и верификации ускорителей вычислений на платформе Xilinx Alveo
- II. Лабораторная работа №2. Разработка ускорителей вычислений средствами САПР высокоуровневого синтеза Xilinx Vitis HLS
- III. Хакатон. Командная разработка концепции ускорителя вычислений

Вторник	13.50 — 17.00
Среда	10.15 — 13.30
Суббота	10.15 — 13.30

Веб аудитория: <https://webinar10.bmstu.ru/b/2mg-euu-zpz-pdu>

Страница практикума: <https://alexbmstu.github.io/2021/>



Телеграмм:
Электронная почта:

@alexpopov_bmstu
alexpopov@bmstu.ru