

# Цифровая схемотехника ПЛИС применение

---

КУРС ЛЕКЦИЙ

ЧУ ПО «СОЦИАЛЬНО-ТЕХНОЛОГИЧЕСКИЙ КОЛЛЕДЖ»

ПРЕПОДАВАТЕЛЬ: БОРИСОВ АЛЕКСЕЙ АЛЬБЕРТОВИЧ

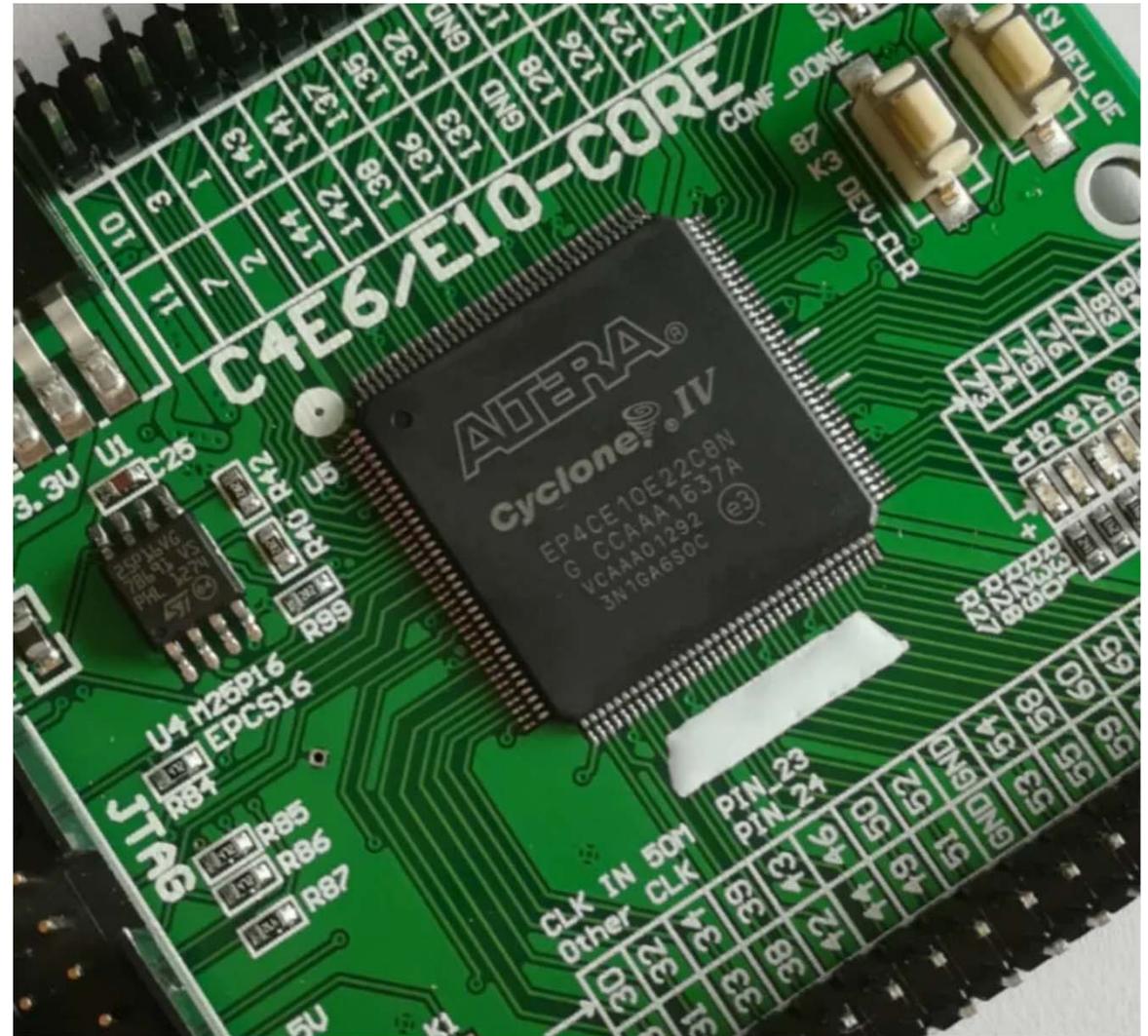
**RAZUMDOM**



# Программируемые логические интегральные схемы ПЛИС FPGA

**FPGA — Field-Programmable Gate Array**, то есть программируемая логическая матрица (ПЛМ), программируемая логическая интегральная схема (ПЛИС).

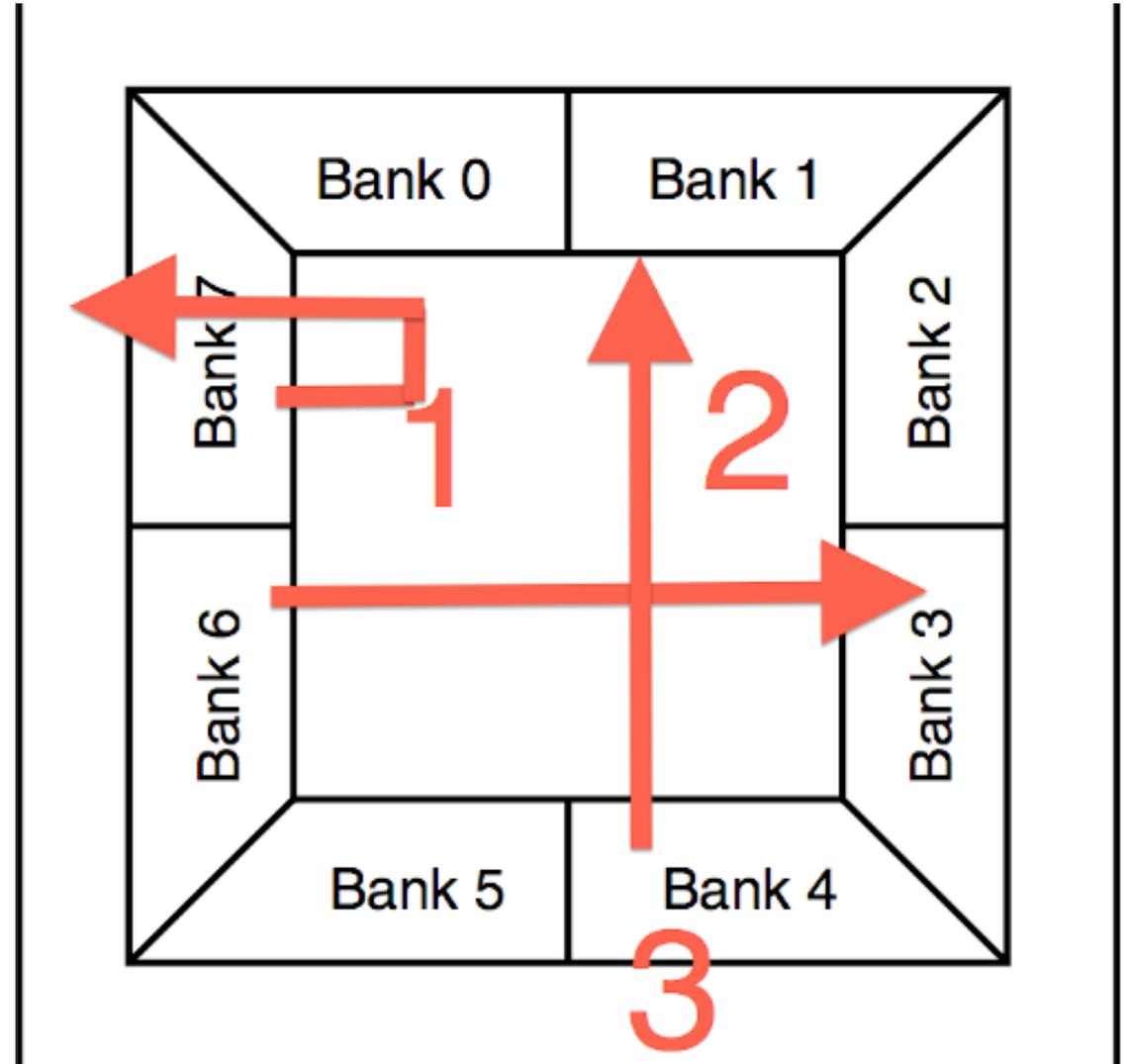
Это технология, при которой создается микросхема с набором логических элементов, триггеров, иногда оперативной памяти и программируемых электрических связей между ними. При этом программирование FPGA больше похоже на разработку электрической схемы, а не программы.





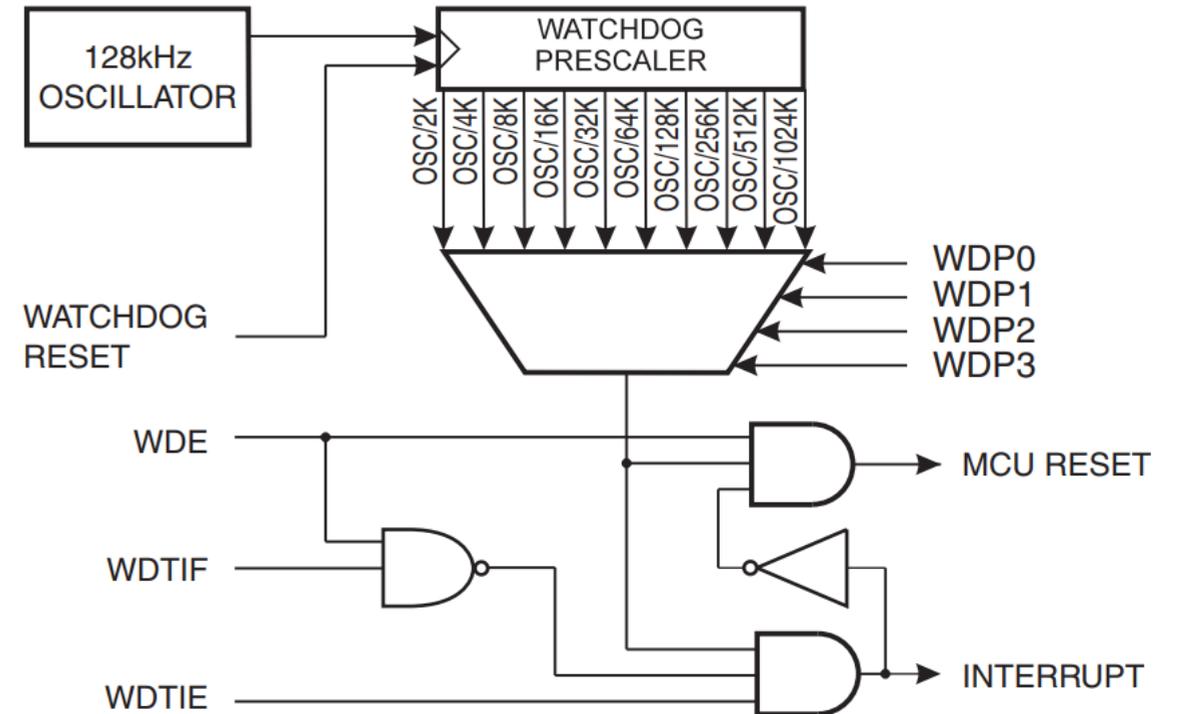
## Согласование уровней сигналов

Если часть микросхем имеет интерфейс 1.2В, другая 1.5, 1.8, 2.5, 3.3В, то соединять между собой их напрямую соединять нельзя, нужно применить устройство согласования уровней напряжений. Один из вариантов решения этой задачи - все эти микросхемы можно подключить к одной FPGA и обеспечить двусторонний обмен за счет того, что любая FPGA имеет несколько банков ввода-вывода, каждый из которых может иметь свое опорное напряжение сигналов.



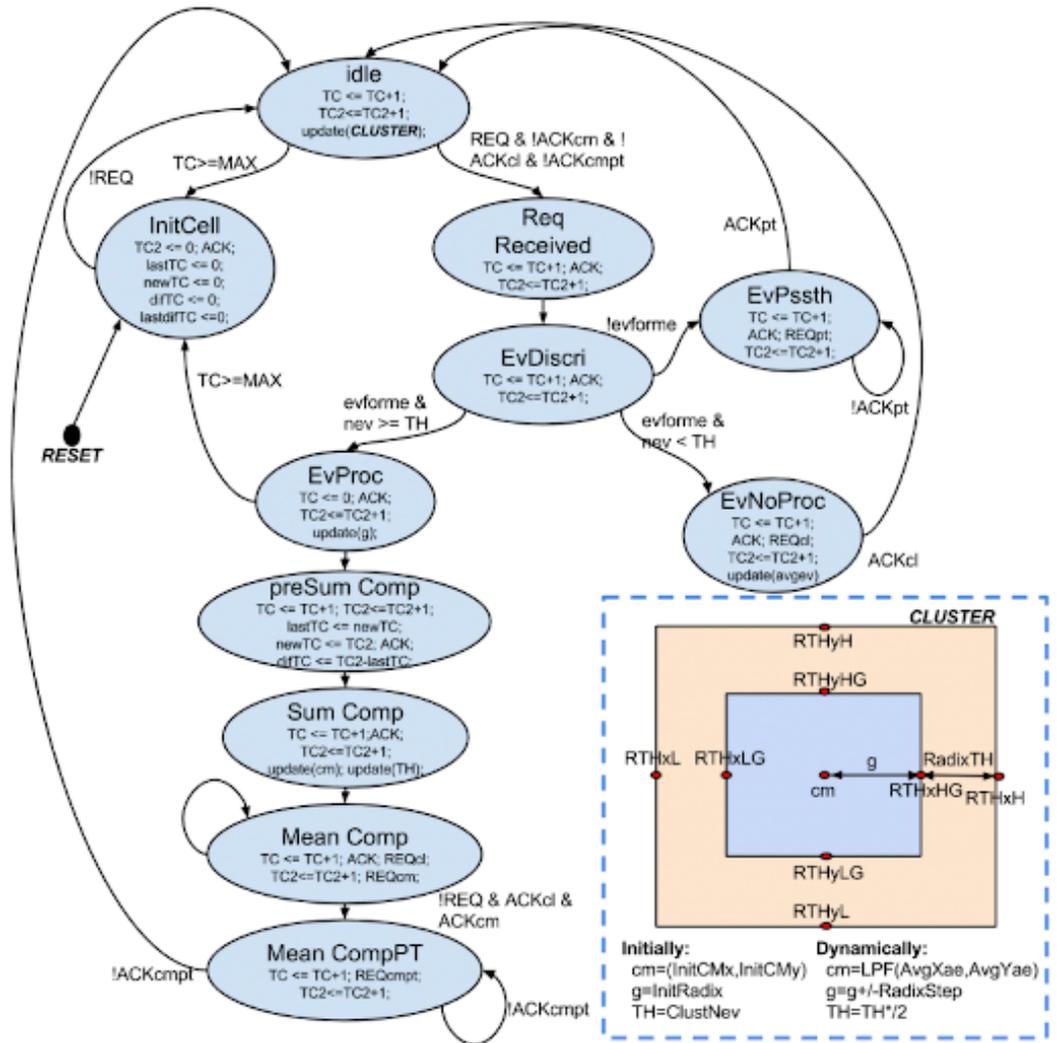
## Обеспечение надежности устройства

FPGA достаточно дорогие, но надежные устройства. Они начинают включаться при меньшем напряжении, чем номинальное, выдерживают импульсные наводки, часто короткое замыкание на ножках IO, быстро загружаются и могут использоваться для контроля и управления процессорами и умной периферией. Плюс могут реализовывать вспомогательные функции коммутации, задержек, моргания светодиодом и так далее. Можно использовать FPGA (маленький PLD) как умный сторожевой таймер и схему запуска.



# Автоматы состояний или аппаратное программирование

Если на процессоре сначала создается «исполнитель команд», то есть процессор, а потом в него загружается последовательность команд, то на FPGA можно писать программу с командами, вшитыми в структуру прошивки. При этом отсутствует избыточность процессора, появляется малое потребление при той же скорости и функциональности, гарантированное время выполнения и высокая надежность. Пример такого автомата состояний:

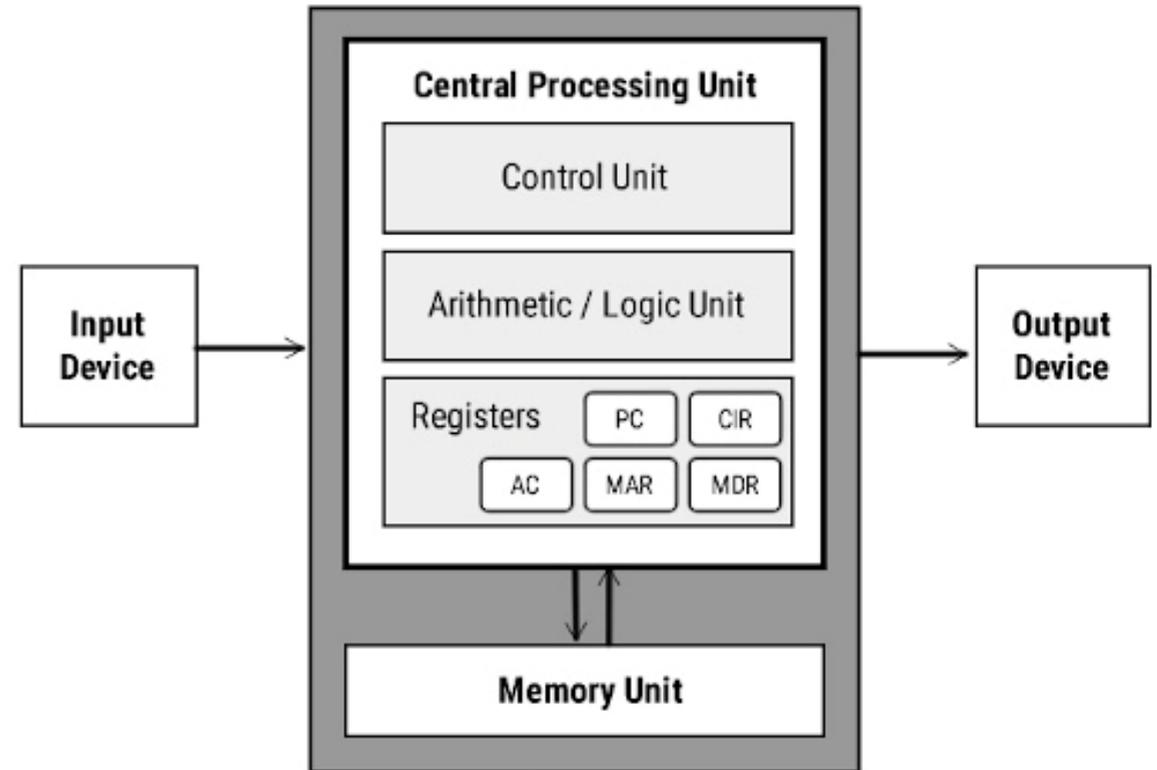


## Создание процессора внутри FPGA

Считается, что каждый программист должен написать хотя бы один компилятор, а каждый инженер должен разработать хотя бы один процессор. Это очень интересный и важный процесс, позволяющий лучше понять как работают микропроцессоры, при этом можно оптимизировать под свои задачи систему команд, встраивать большое количество одновременно работающих процессоров в одну микросхему и получать настоящую многозадачность с низким энергопотреблением.

При этом структура процессора проста и легко реализуема на FPGA.

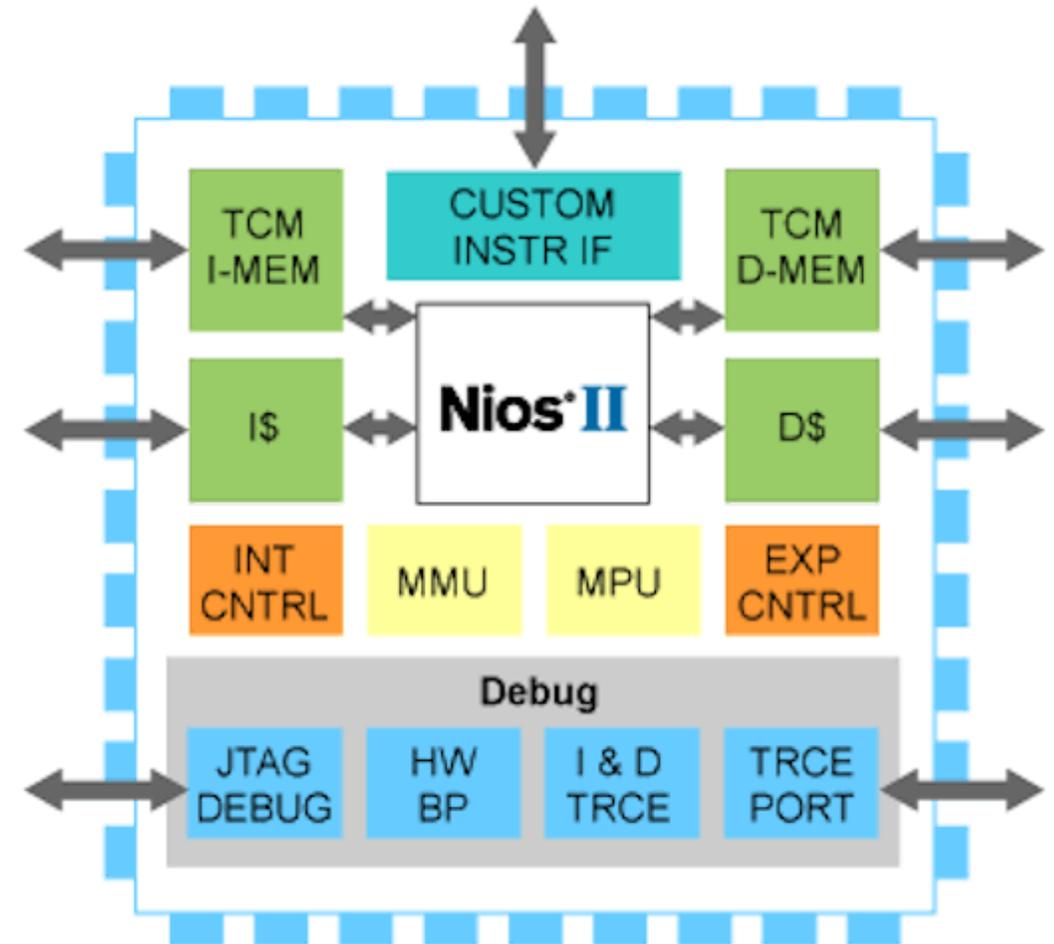
Недостатком такого процессора является отсутствие готовых компиляторов и отладчиков.



## Использование готовых библиотек процессоров для FPGA

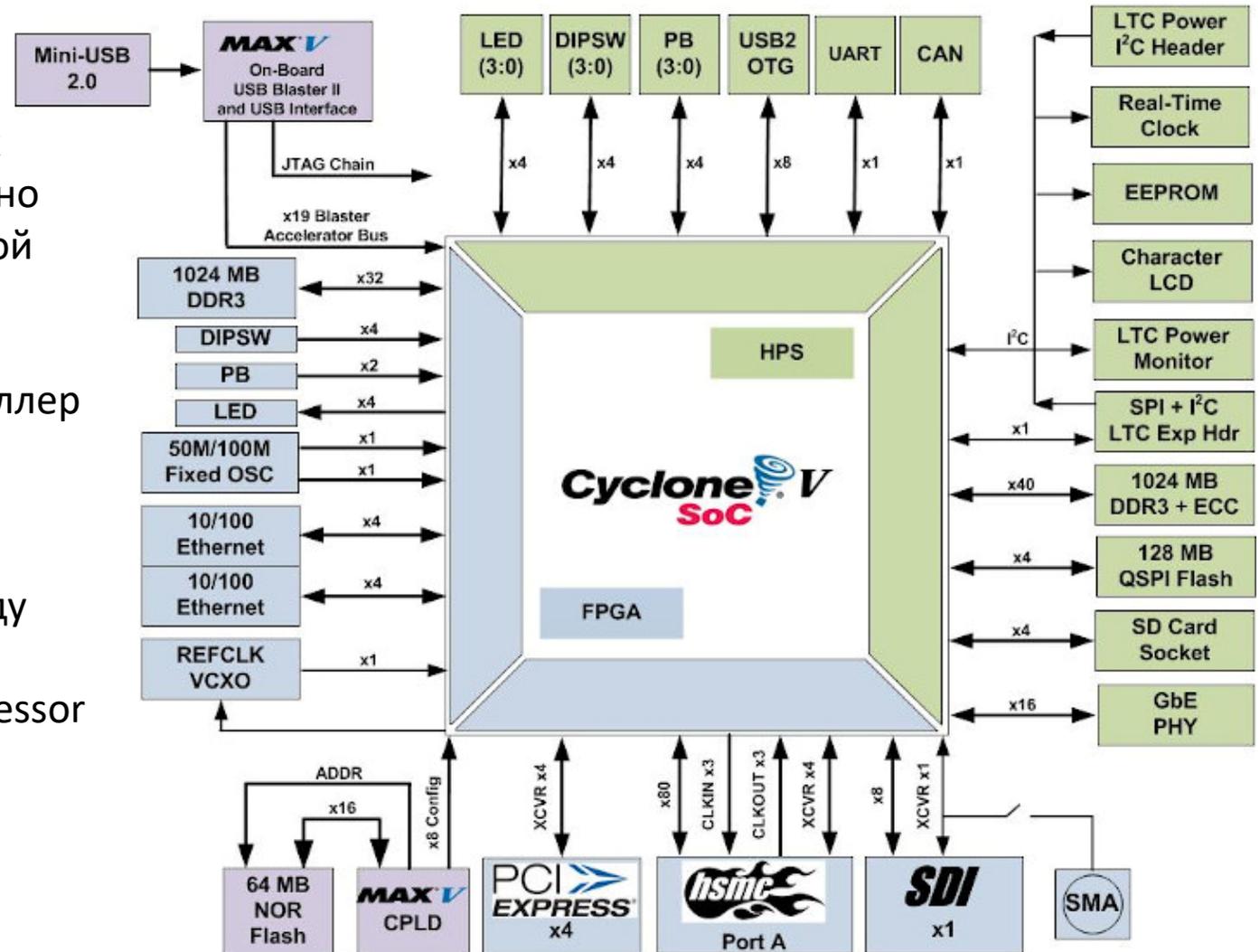
Библиотеки готовых процессоров есть у любого производителя FPGA (от 8086 до ARM), позволяют быстро создать процессор с определенным набором периферии и вставить его в проект FPGA. К процессору прилагается компилятор и отладчик. Быстро, удобно, но избыточно и потому ограничено по быстродействию.

Пример структуры готовой библиотеки процессора:



# Объединение процессора и периферии в одной микросхеме — SoC (System-On-Chip)

SoC — достаточно новая технология, решающая сложную проблему инженера, необходимость протаскивать по плате много высокоскоростных интерфейсов, и которые необходимо программно поддерживать. Технология SoC позволяет в одной микросхеме иметь полноценный центральный процессор (например, поддерживающий операционную систему Linux) или микроконтроллер и большую FPGA, соединенные логическими сигналами, общей внутренней памятью и интерфейсами. То есть проблема эффективной, простой и быстрой передачи информации между FPGA и процессором может быть решена с помощью SoC. Видно, что FPGA и HPS (Host Processor System, процессор) находятся внутри одной микросхемы и окружены программируемыми ножками ввода-вывода. Действительно это многофункциональная система на кристалле.



Вспоминая, что FPGA — это набор логических ячеек и триггеров, работающих параллельно, на FPGA можно проводить много параллельных операций, что отличает от процессора, параллельность которого ограничена количеством ядер и потоков. Поэтому можно использовать FPGA как сопроцессор к центральному процессору, вынося на FPGA все самые требовательные к вычислительной мощности операции.

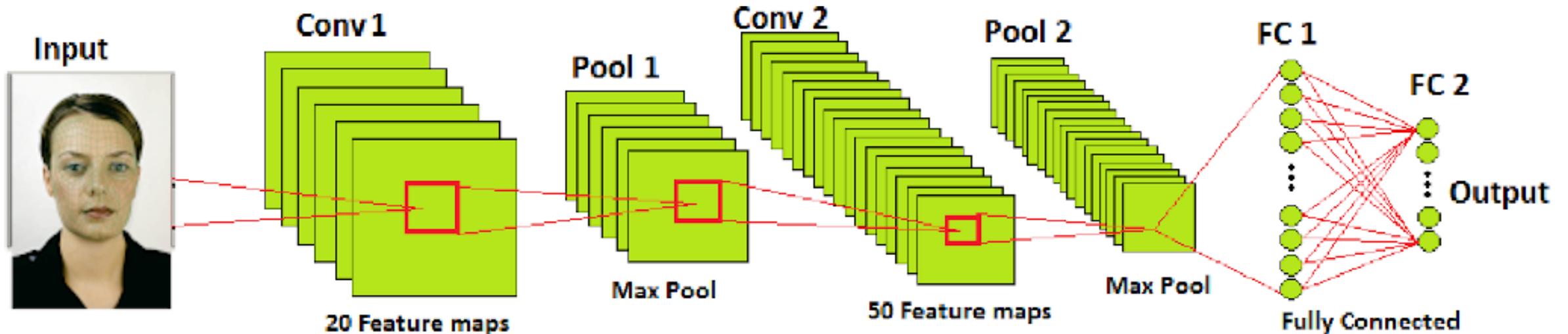
Например, центральный процессор занимается логической обработкой задачи, а FPGA параллельно вычисляет контрольные суммы, хэши, ищет совпадения, перебирает варианты и так далее. Быстродействие FPGA ограничено только количеством параллельных блоков и временем выполнения одной операции. Отладив таким образом вычисления можно заказать **ASIC**, то есть заказную микросхему, выполняющую те же функции, но дешевле (при массовом производстве) и с меньшим энергопотреблением. И данная идея оказалась настолько перспективной и удобной, что гиганты разработки FPGA начали создавать специальное ПО, позволяющее интерактивно переносить части вычислений из программы на C/C++ в FPGA и контролировать быстродействие (HLS, High-Level Synthesis). Для такой задачи есть готовые платы с быстрыми интерфейсами и средства отладки.



# Реализация нейронных сетей на FPGA

**Нейронные сети и глубокие нейронные сети** сейчас активно используются в разных областях, но реализация их на процессоре оказывается неэффективной — существует много вычислений, которые можно распараллелить (нейроны одного слоя, например, вычисляются независимо).

Поэтому перенеся нейронную сеть на FPGA удастся на много порядков ускорить работу нейронной сети, остается обеспечить высокоскоростной интерфейс для загрузки исходных данных и получения результата. В качестве примера — реализация системы распознавания лиц на процессоре i7/9Gen распознает до 20 лиц за секунду с одной видеокамеры HD, реализация на FPGA — порядка 1000 лиц с нескольких камер. Структура используемой глубокой нейронной сети:



**Спасибо за внимание**

ЧУ ПО «Социально-технологический колледж»

Преподаватель: Борисов Алексей Альбертович