



Проектирование цифровых устройств на языке VHDL

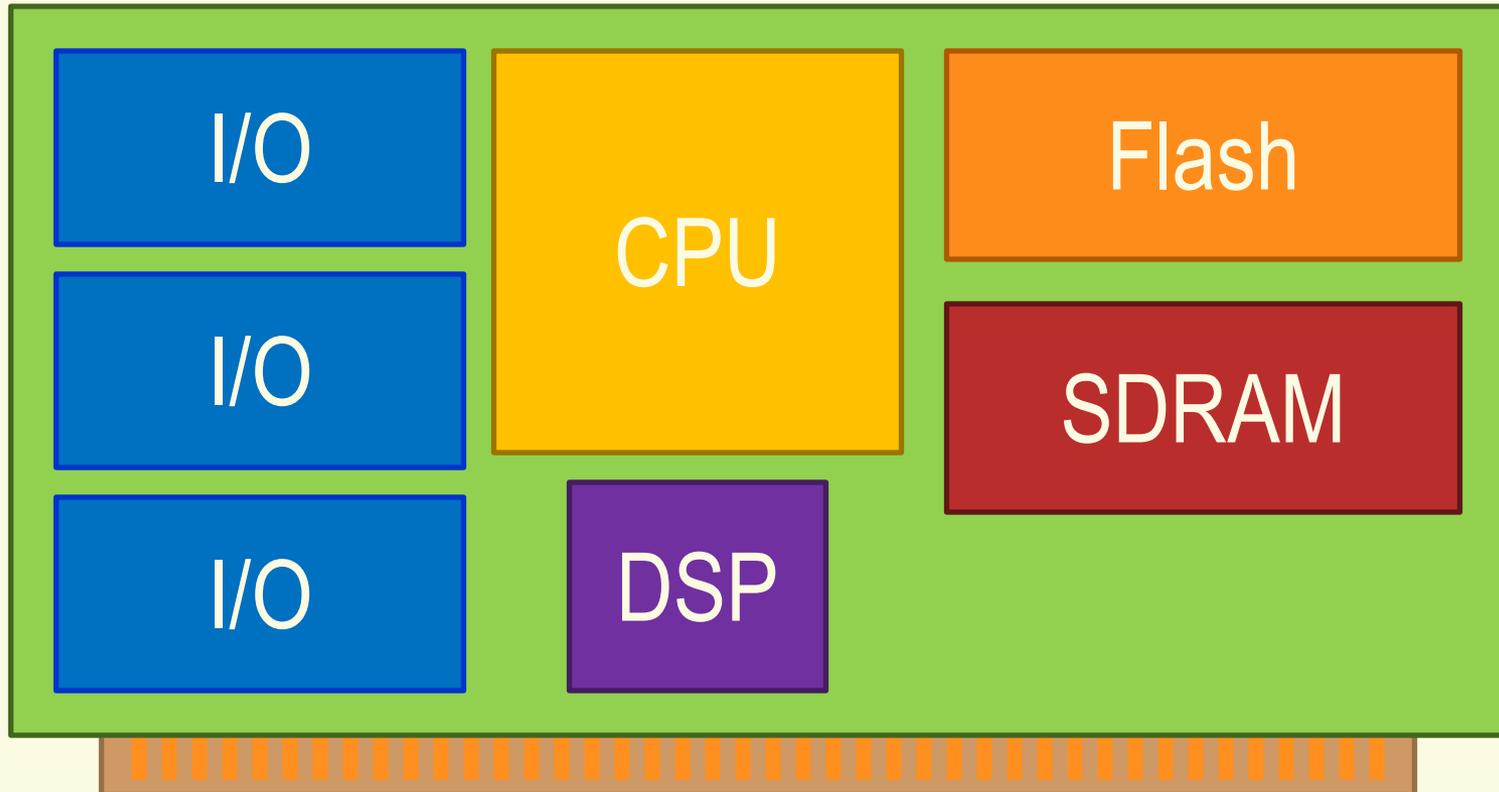
Основы программируемой логики



Применение ПЛИС

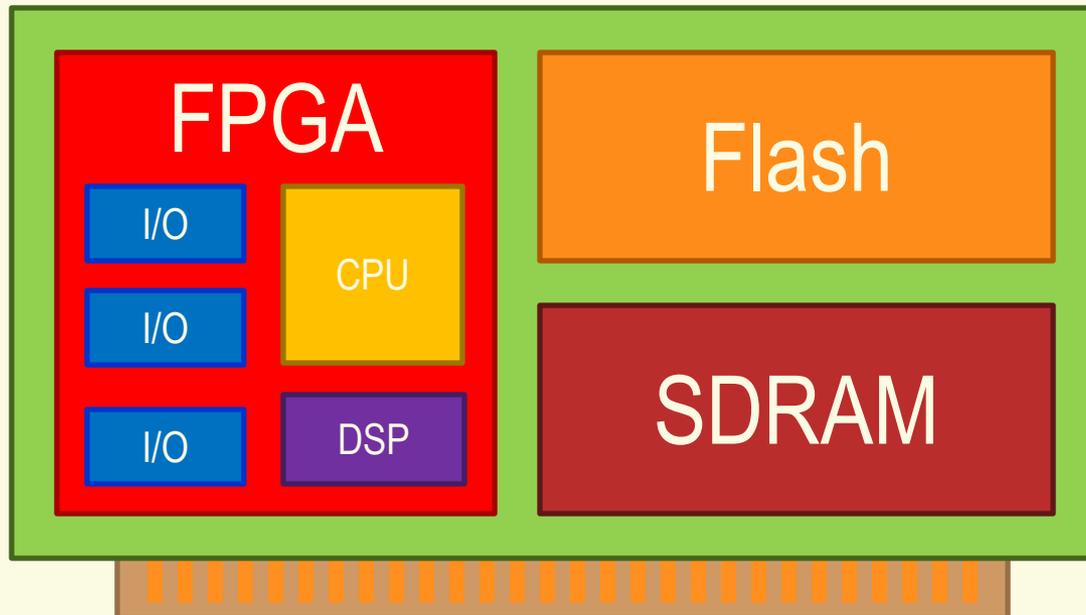
Цифровое устройство – дискретные микросхемы

- Большое число различных компонентов
- Сложные печатные платы
- Плохая взаимозаменяемость
- Проблемы обеспечения нестандартных функций



Цифровое устройство – ПЛИС

- Совмещение различных компонент в одной МС
- Упрощение устройства
- Возможность введения новых функций без изменения конструкции
- Снижение стоимости и энергопотребления



Некоторые сферы применения

- Бытовая техника
 - Обработка сигналов
 - Цифровое телевидение
- Промышленность
 - САР / САУ
 - Измерительное оборудование
- Телекоммуникации
 - Коммутационное оборудование
- **Безопасность**
 - Защищенные системы связи
 - Специальные электронные устройства



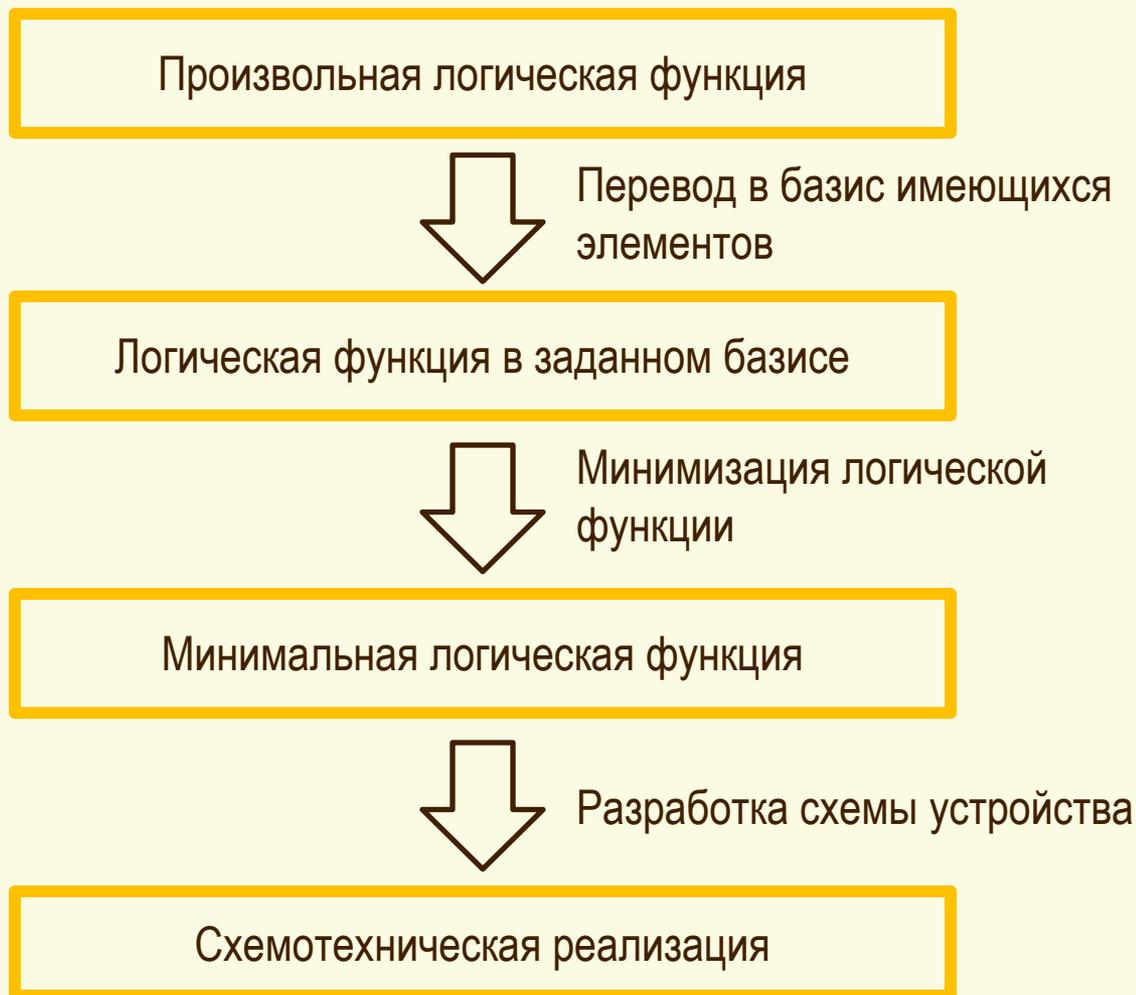
История развития программируемой ЛОГИКИ

Дискретная TTL-логика



- Прimitives реализованы в отдельных микросхемах:
 - NAND, OR, ...
 - мультиплексоры, дешифраторы, сумматоры...
 - триггеры, регистры, ...
- Конструкция устройства определяется ценой и доступной номенклатурой элементов

Дискретная TTL-логика (разработка)

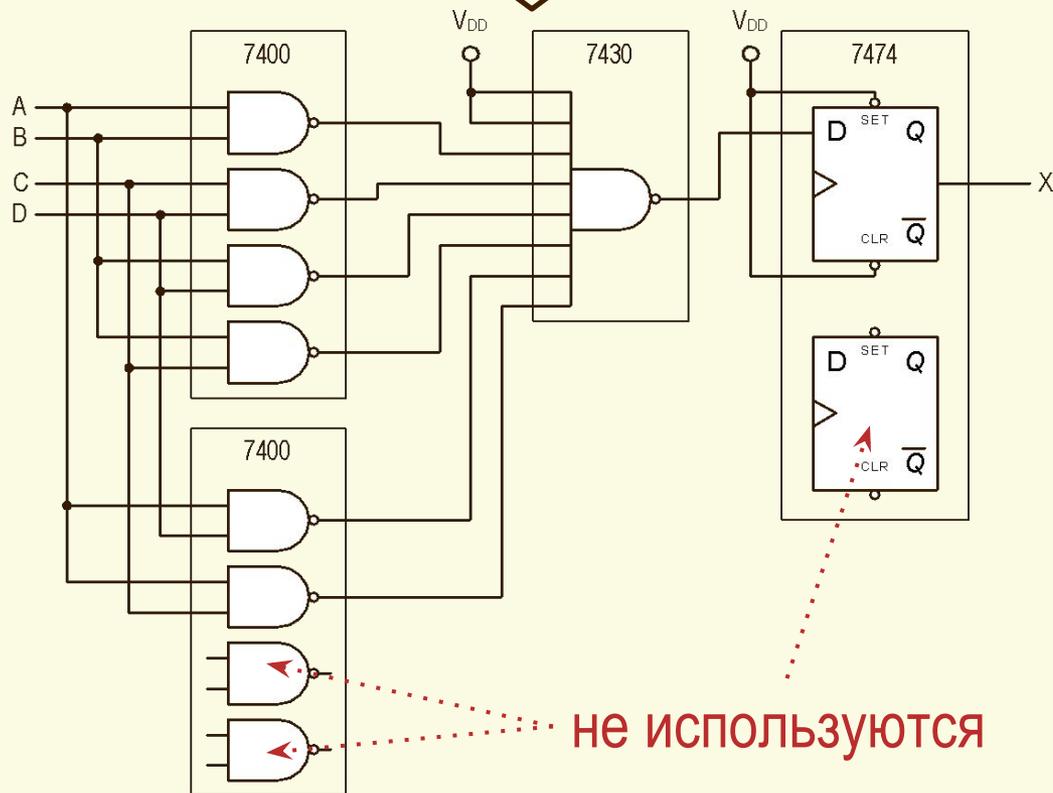
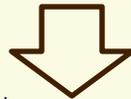


Дискретная TTL-логика (разработка)

$$X = AB + CD + BD + BC + AD + AC$$



$$X = \overline{AB \cdot CD \cdot BD \cdot BC \cdot AD \cdot AC}$$

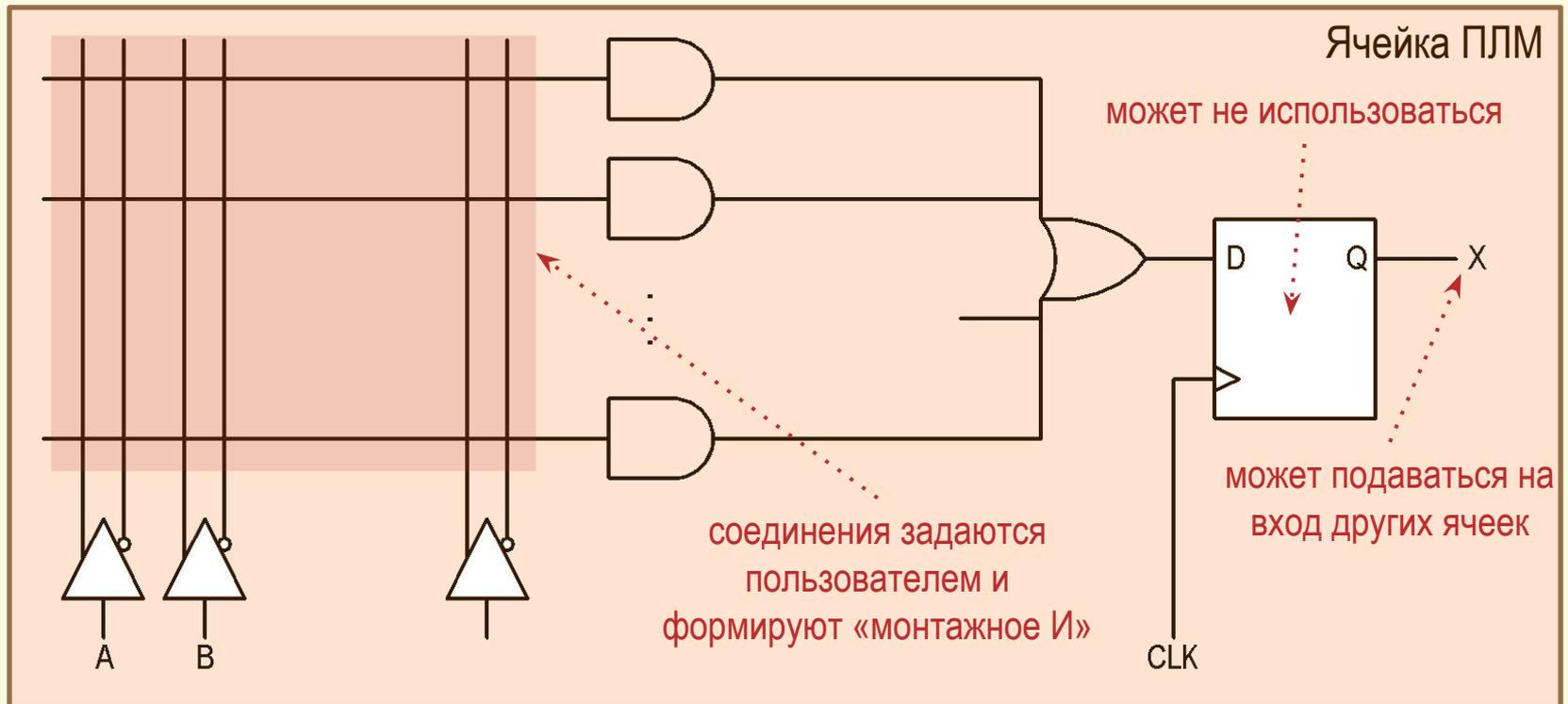


Реализация логических функций

- Логические функции...
 - Могут быть выражены в базисе И-ИЛИ-НЕ
 - Различные функции задаются соединением элементов
 - Для хранения результата используются триггеры
- Что если...
 - Логический базис фиксирован
 - Все логические элементы совмещены на **одном** устройстве
 - Соединения задаются **пользователем**

Программируемые логические матрицы (ПЛМ)

- Programmable **A**rray **L**ogic (**PAL**)
- Наиболее простая реализация логических функций
- Логические элементы и триггеры фиксированы
- Соединения задаются пользователем

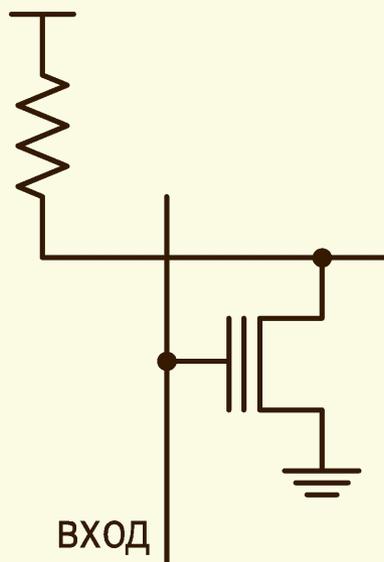


Технология программирования

- При программировании транзисторы с плавающим затвором переводятся в непроводящее состояние

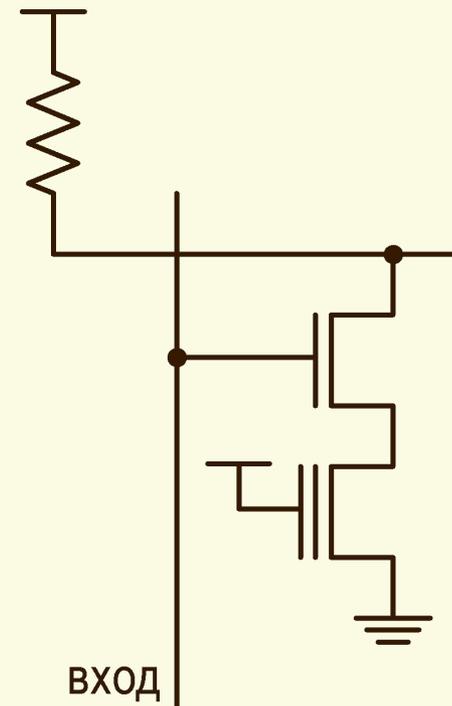
FAMOS

(Floating-gate Avalanche injection MOS transistor)



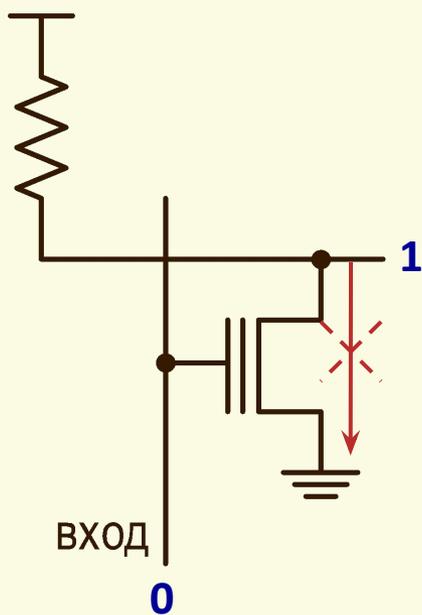
FLOTOX

(Floating-gate Tunnel Oxide transistor)

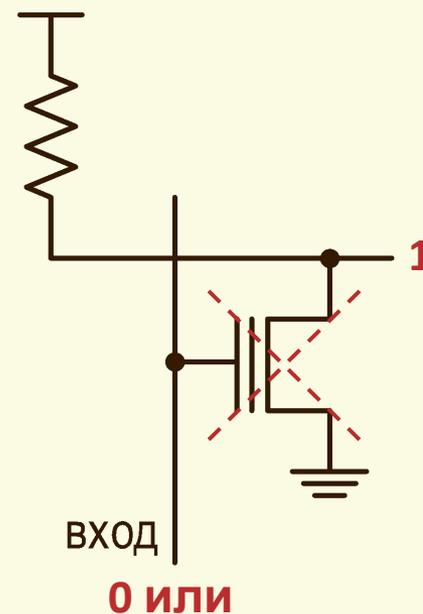
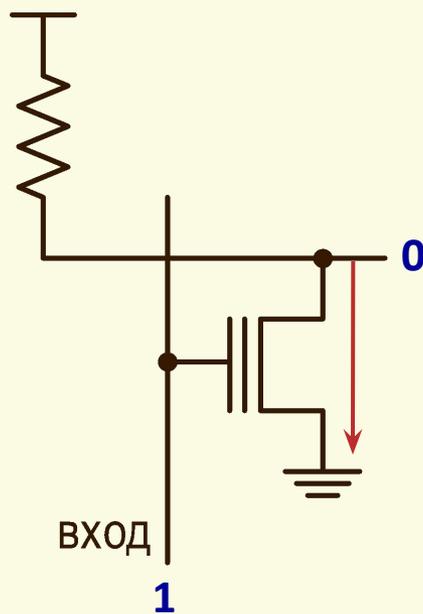


Технология программирования – FAMOS

- Запись напряжением
- Стирание ультрафиолетом (EPROM)



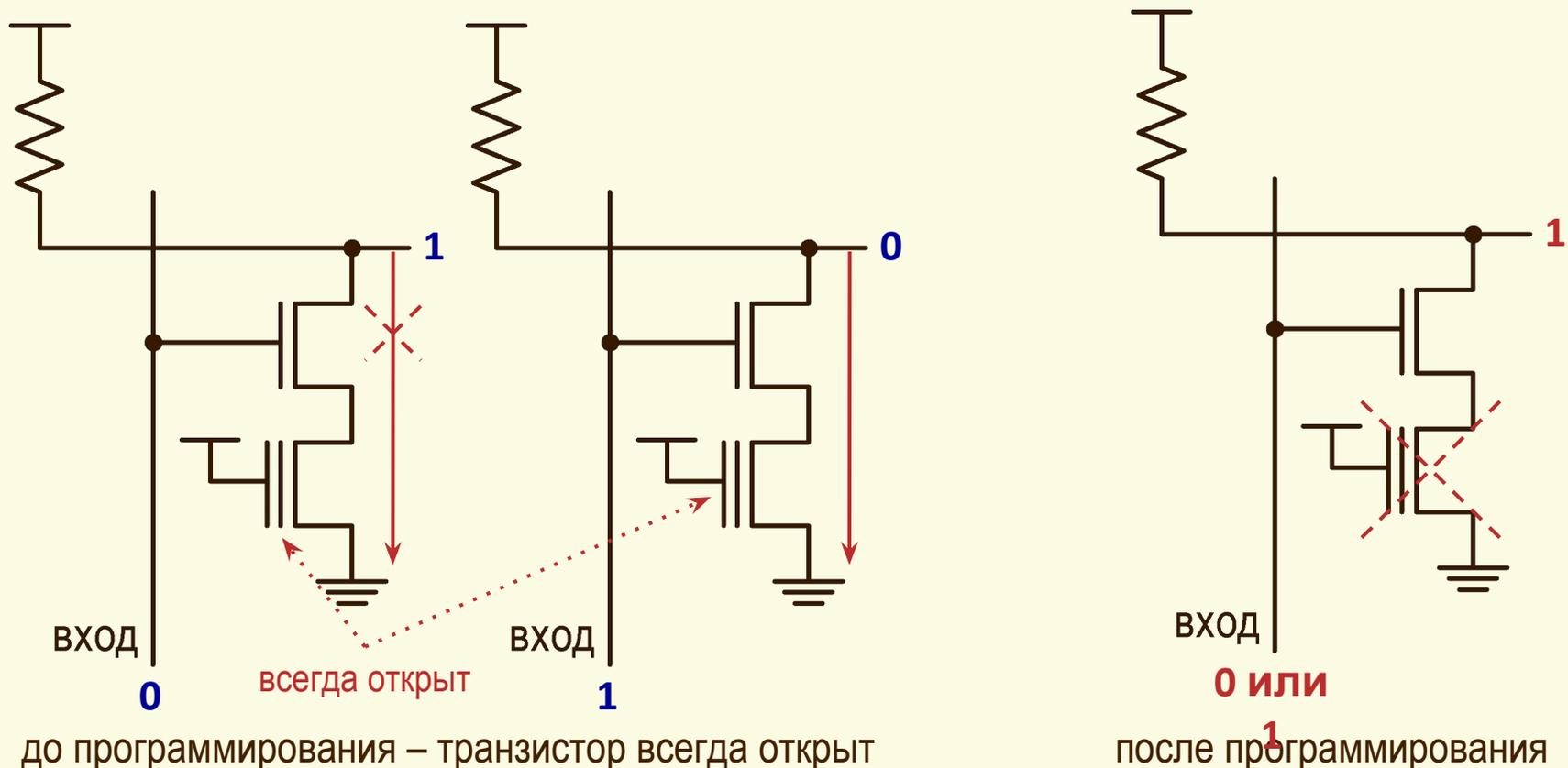
до программирования – обычный транзистор



после программирования

Технология программирования – FLOTOX

- Запись напряжением
- Стирание **обратным напряжением** (EEPROM)



Преимущества программируемой логики

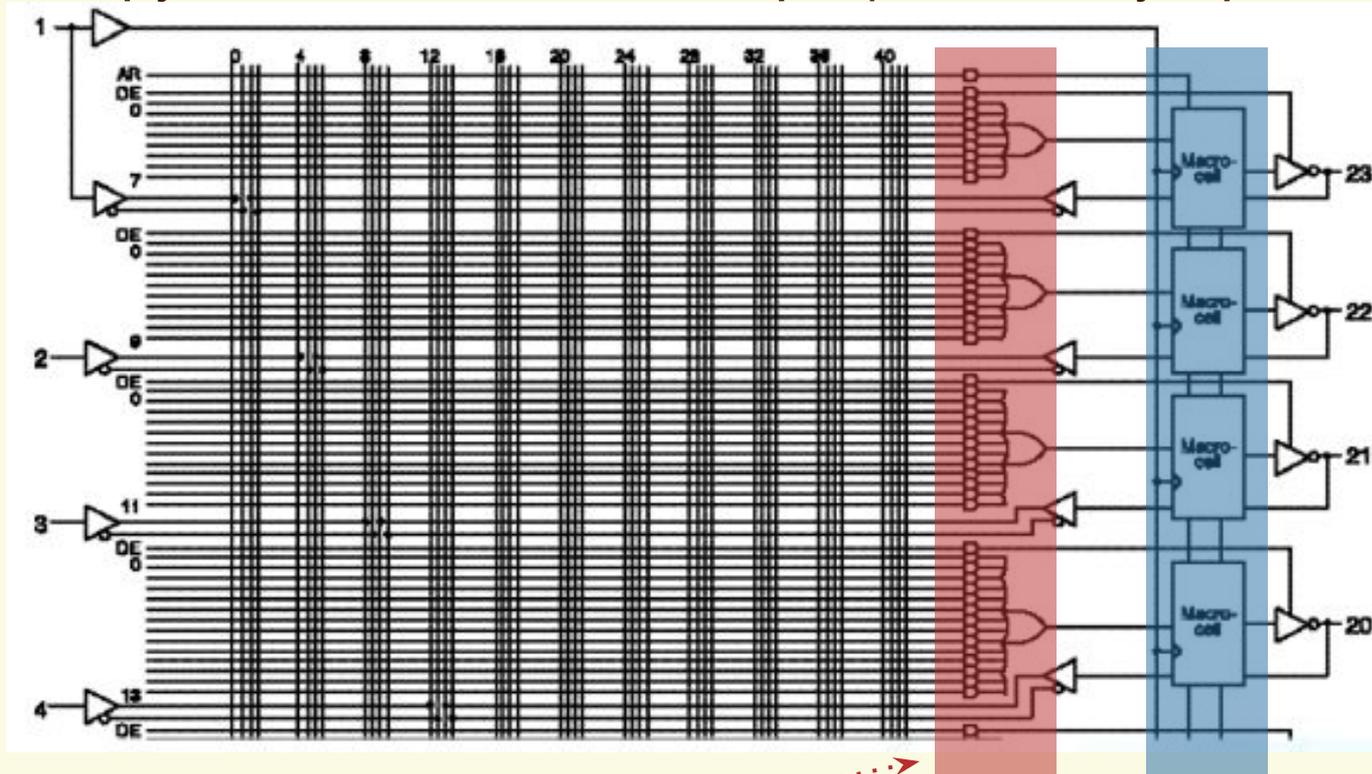
- Снижение числа микросхем
 - Ниже стоимость и энергопотребление
 - Меньше занимаемая площадь
- Повышение безопасности
 - Труднее восстановить схему
- Простота внесения изменений
 - Совместимость по выводам микросхем
 - Многократное программирование (в том числе **внутрисхемное**)
- Наличие средств автоматизированного проектирования



Основы программируемой логики

От PAL к PLD

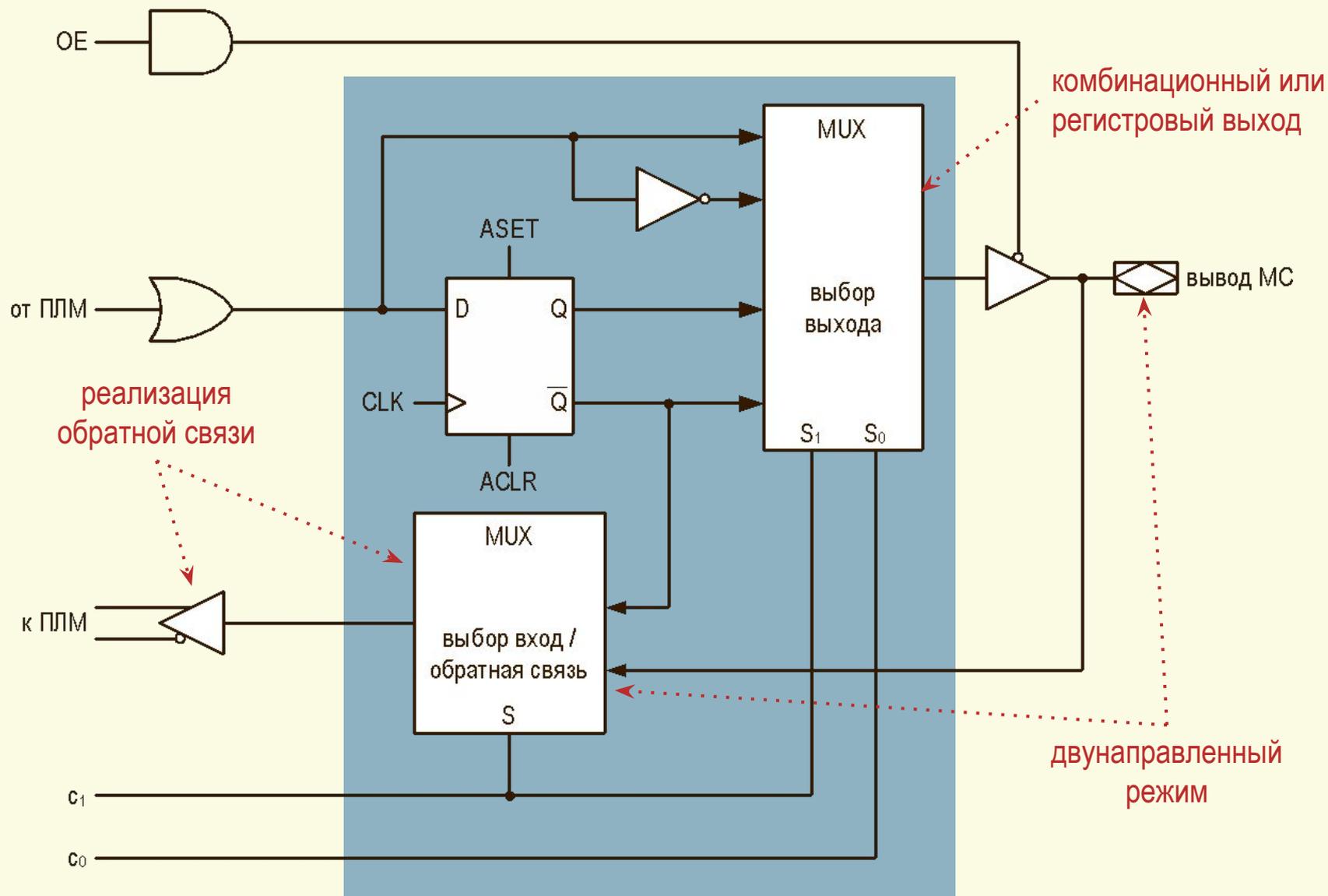
- **PLD** (**P**rogrammable **L**ogic **D**evice) – соединение нескольких программируемых логических матриц в одном устройстве



различное число
аргументов

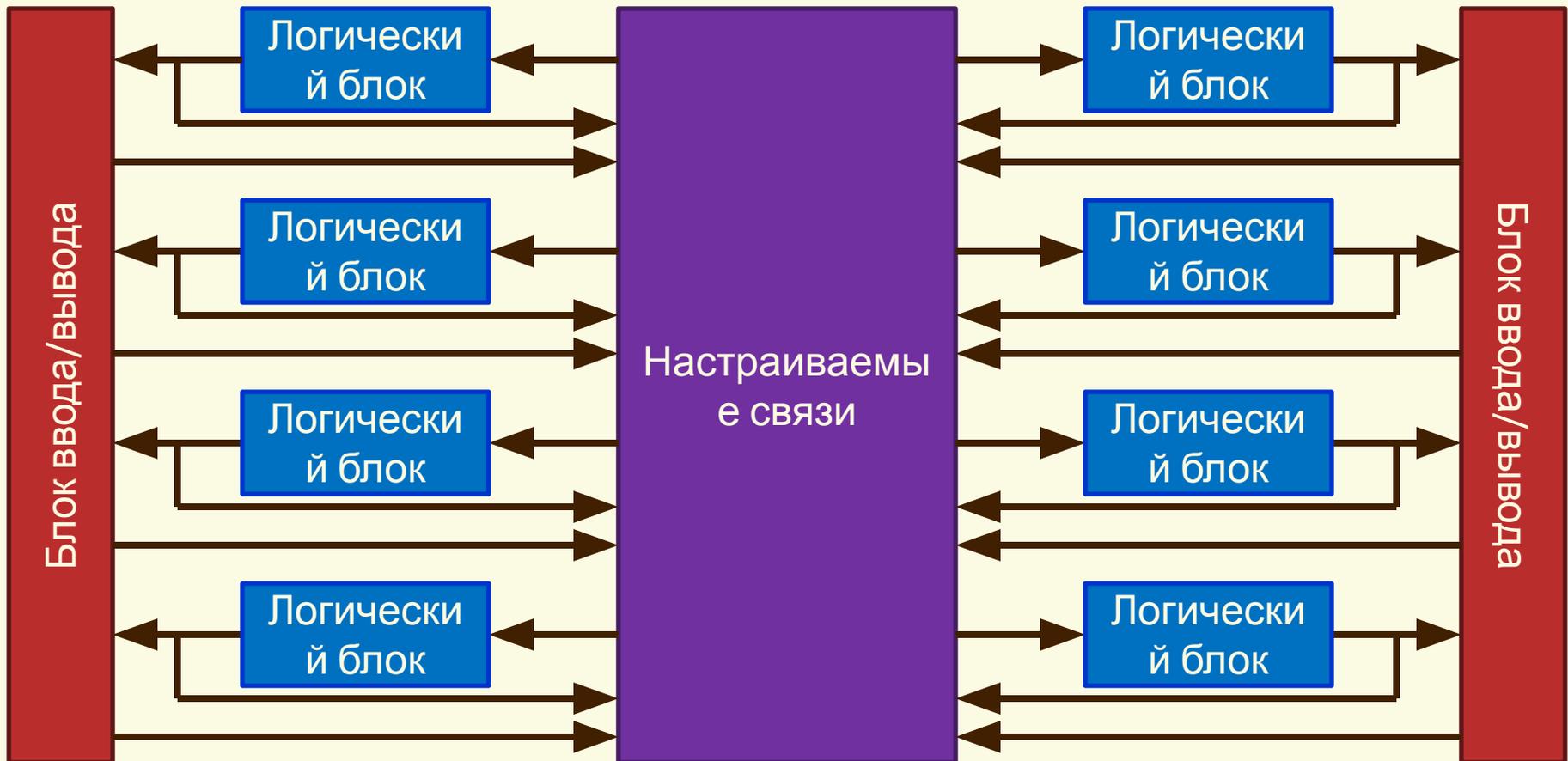
настраиваемые
макроячейки

Настраиваемая макроячейка



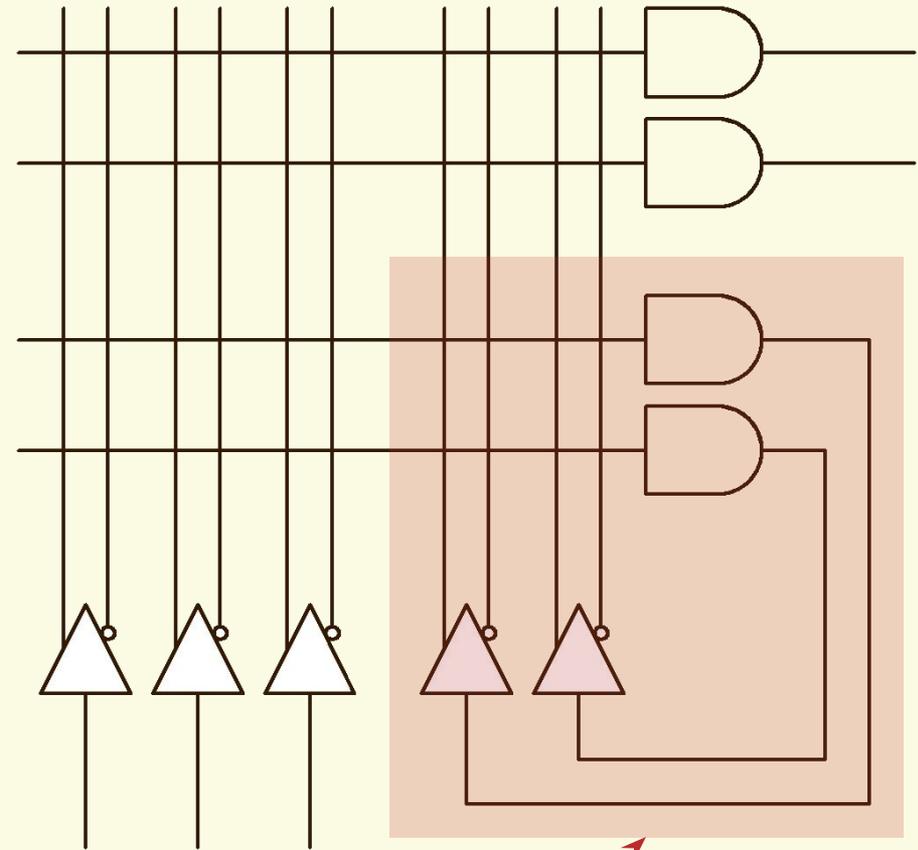
От PLD к CPLD

- **CPLD** (**C**omplex **PLD**) – соединение нескольких PLD в одном устройстве



Логический блок CPLD

- Обычно называется **LAB** (**L**ogic **A**rray **B**lock)
- Содержит несколько макроячеек (от 4 до 20)
- Связи программируются (как в PLD)
- Обратные связи позволяют вычислять конъюнкции только один раз для всех макроячеек



обратные связи

Настраиваемые связи

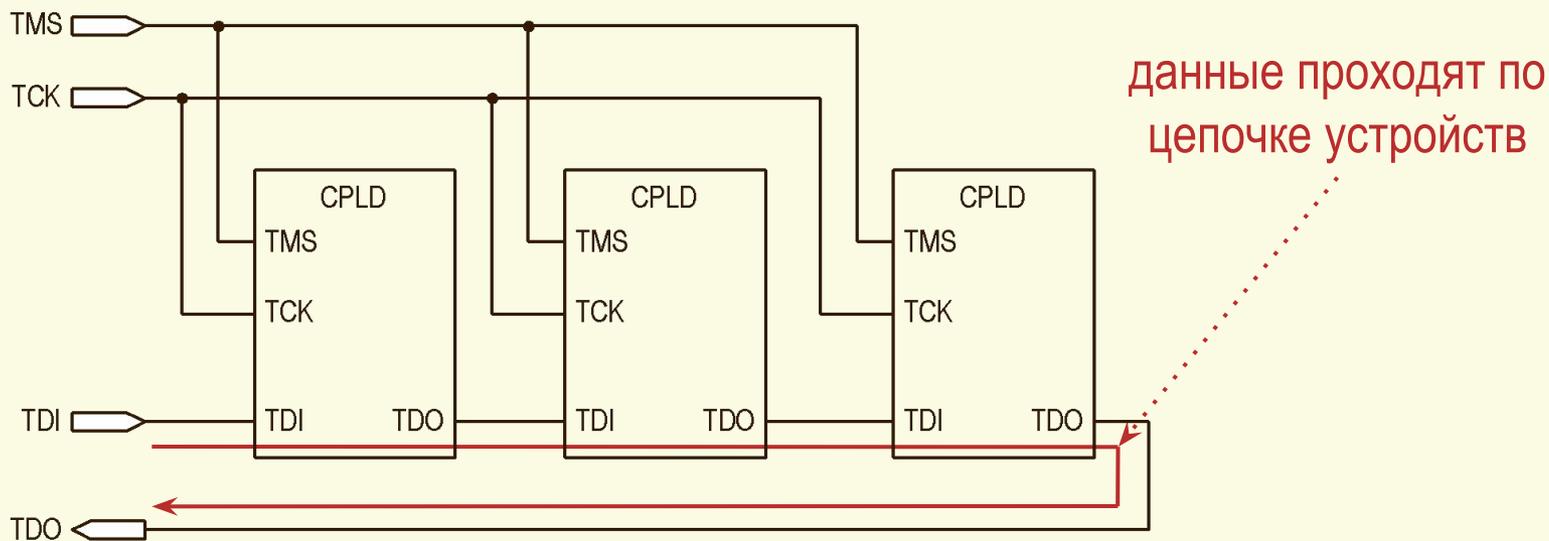
- Programmable Interconnect Array (PI или PIA)
- Принцип аналогичен ПЛМ
- Глобальные связи позволяют соединить **любой сигнал с любым другим**:
 - Между логическими блоками (LAB)
 - Между логическими блоками и выводами микросхемы
- Программирование по технологии EPROM / EEPROM / Flash

Блоки ввода-вывода

- Обеспечивают интерфейс с внешним миром
- Соединены с логическими блоками через PIA (в PLD – напрямую)
- Различные режимы ввода-вывода
 - Pull-up или pull-down
 - Несколько стандартов
 - Направление: входной, выходной или двунаправленный (через тристабильные буферы)

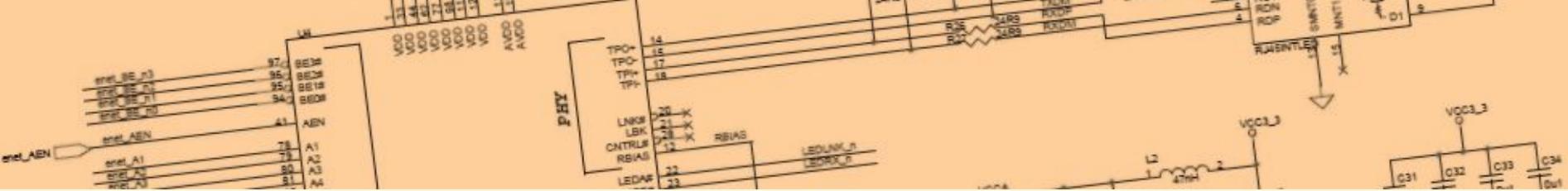
Внутрисхемное программирование

- **In-System Programming (ISP)**
 - Напряжение программирования вырабатывается CPLD
 - Управляющий интерфейс указывает, куда прикладывается напряжение
- Простой 4-проводной последовательный интерфейс JTAG
 - Позволяет работать с несколькими устройствами
 - Может использоваться также для самотестирования



Преимущества CPLD

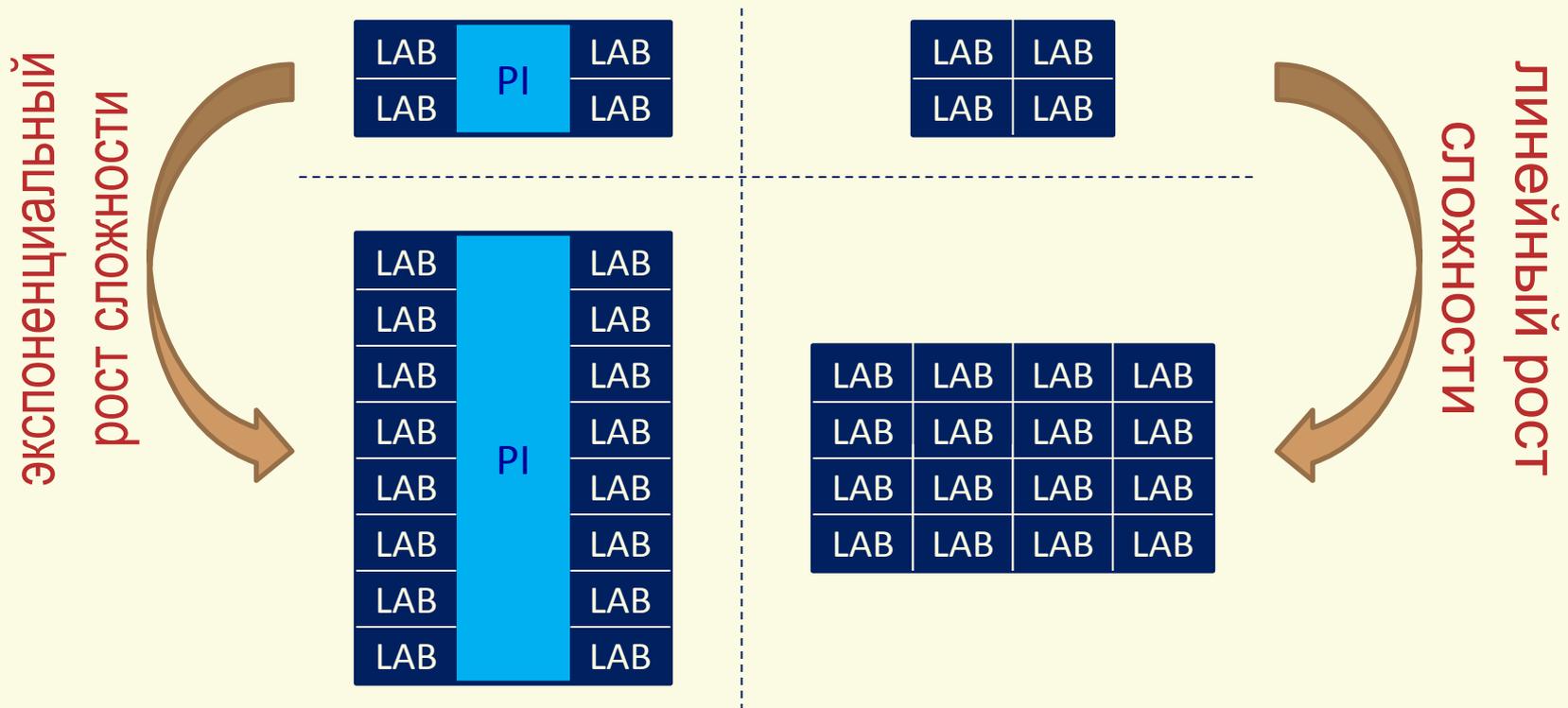
- Низкая стоимость
- Большая логическая емкость
- Программируемые внутренние связи
- Готовы к работе сразу после подачи питания
- Энергонезависимая конфигурация
- Возможность многократного программирования



Что такое FPGA?

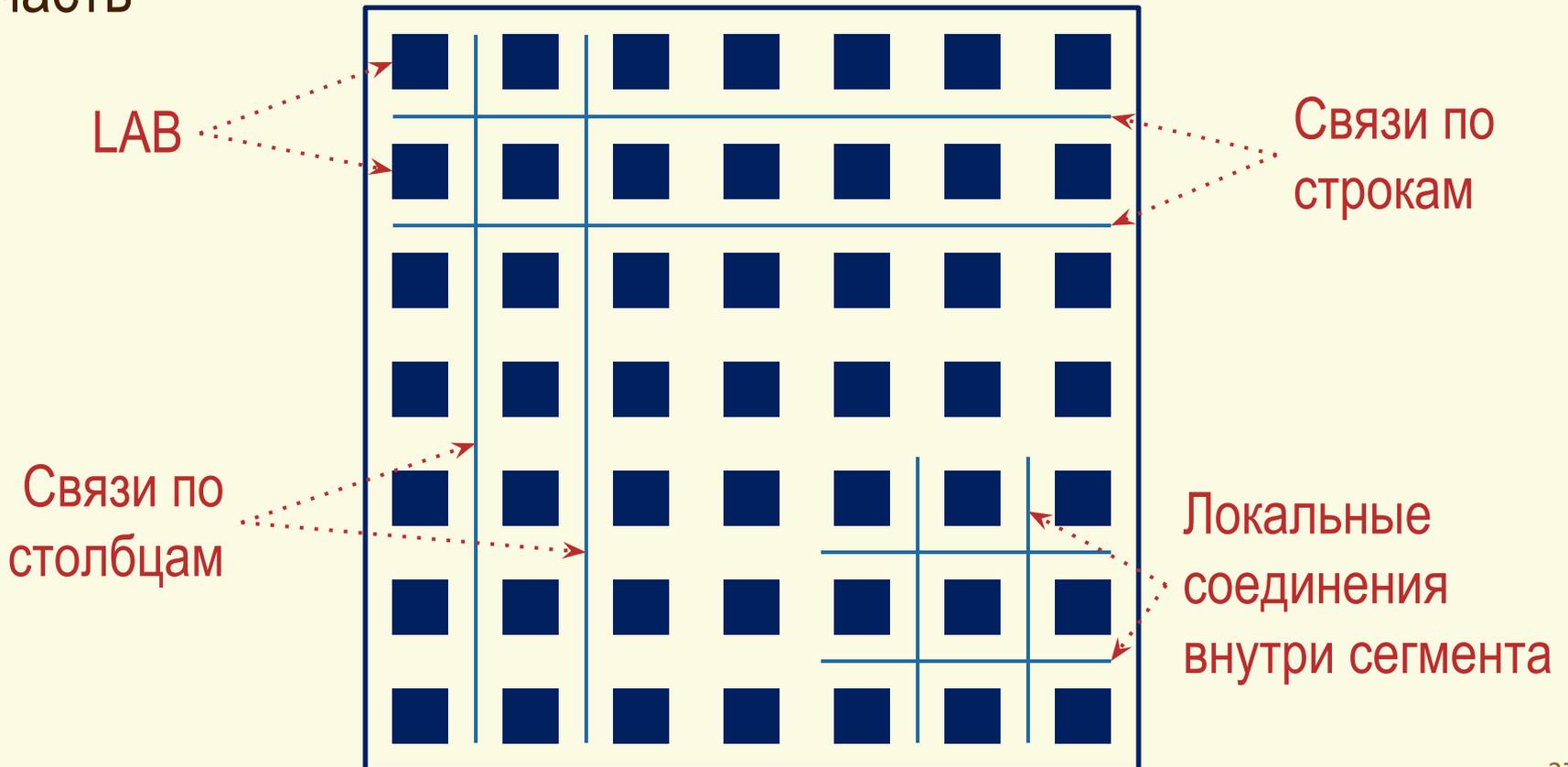
Увеличение емкости CPLD

- При увеличении логической емкости CPLD сложность соединений растет экспоненциально
 - А если упорядочить блоки LAB в массив и соединять только соседние?



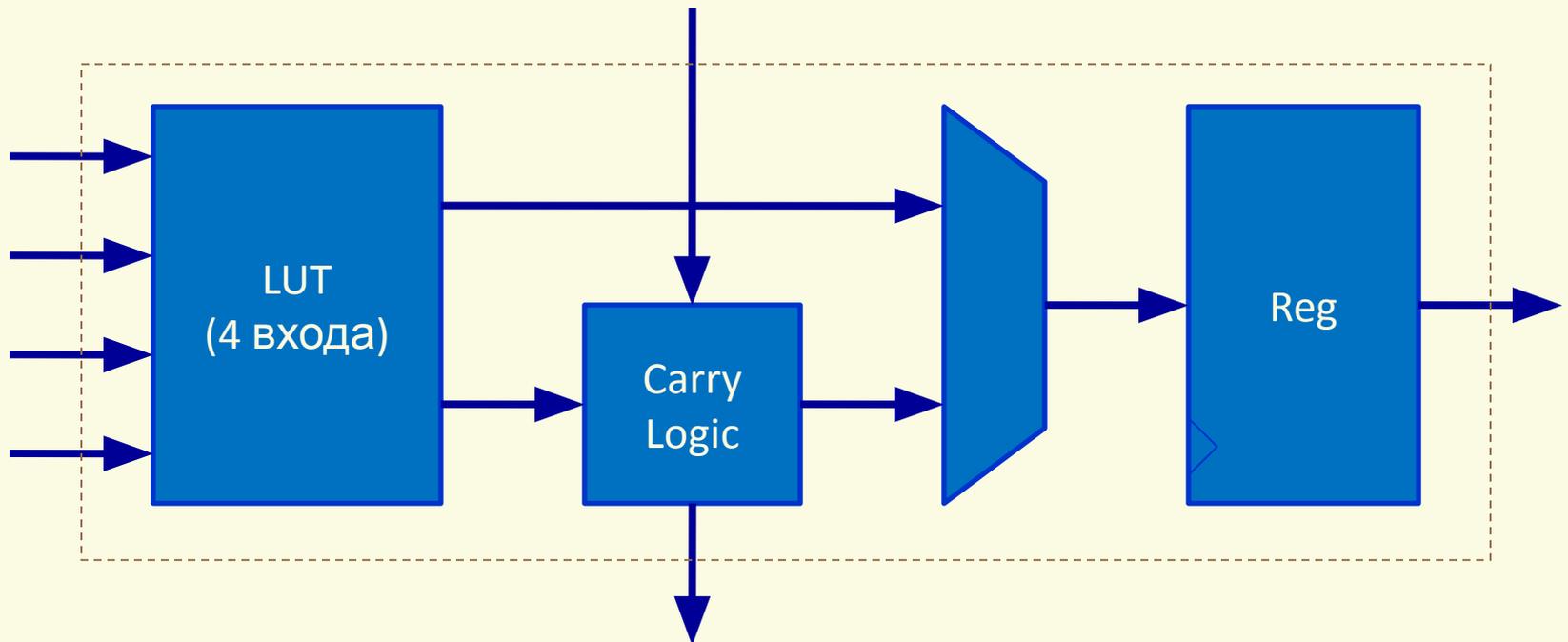
Field Programmable Gate Array (FPGA)

- LAB упорядочены в двумерную матрицу
- Связи располагаются между столбцами и строками матрицы
- Линии связей могут распространяться на всю FPGA или ее часть



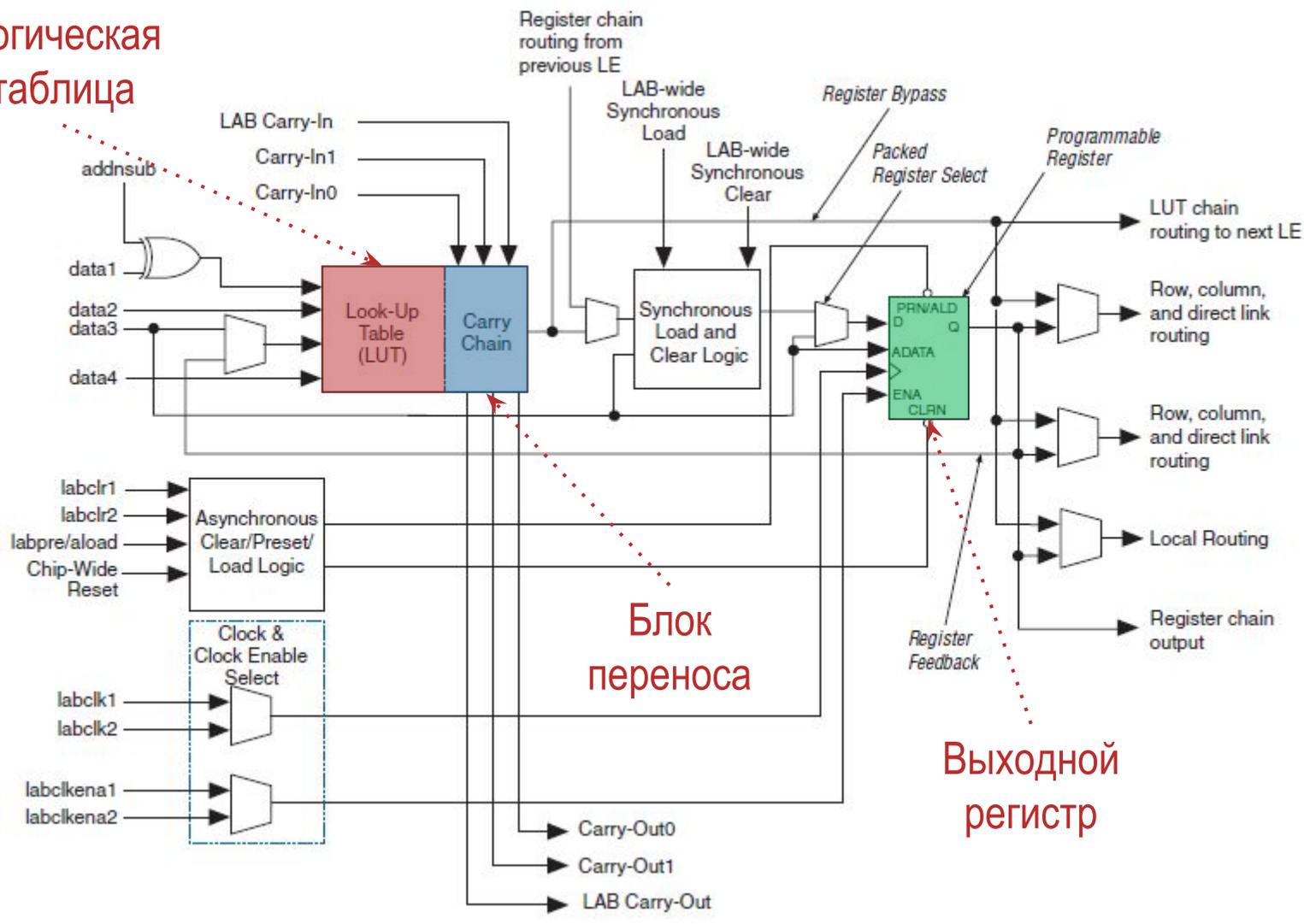
FPGA LE

- LAB состоят из логических элементов (**L**ogical **E**lement, **LE**)
 - В CPLD – из программируемой матрицы соединений
- Основные блоки LE:
 - Логическая таблица (**L**ook-**U**p **T**able, **LUT**)
 - Блок переноса (Carry Logic)
 - Настраиваемый регистр



FPGA LE (Altera Cyclone)

Логическая
таблица

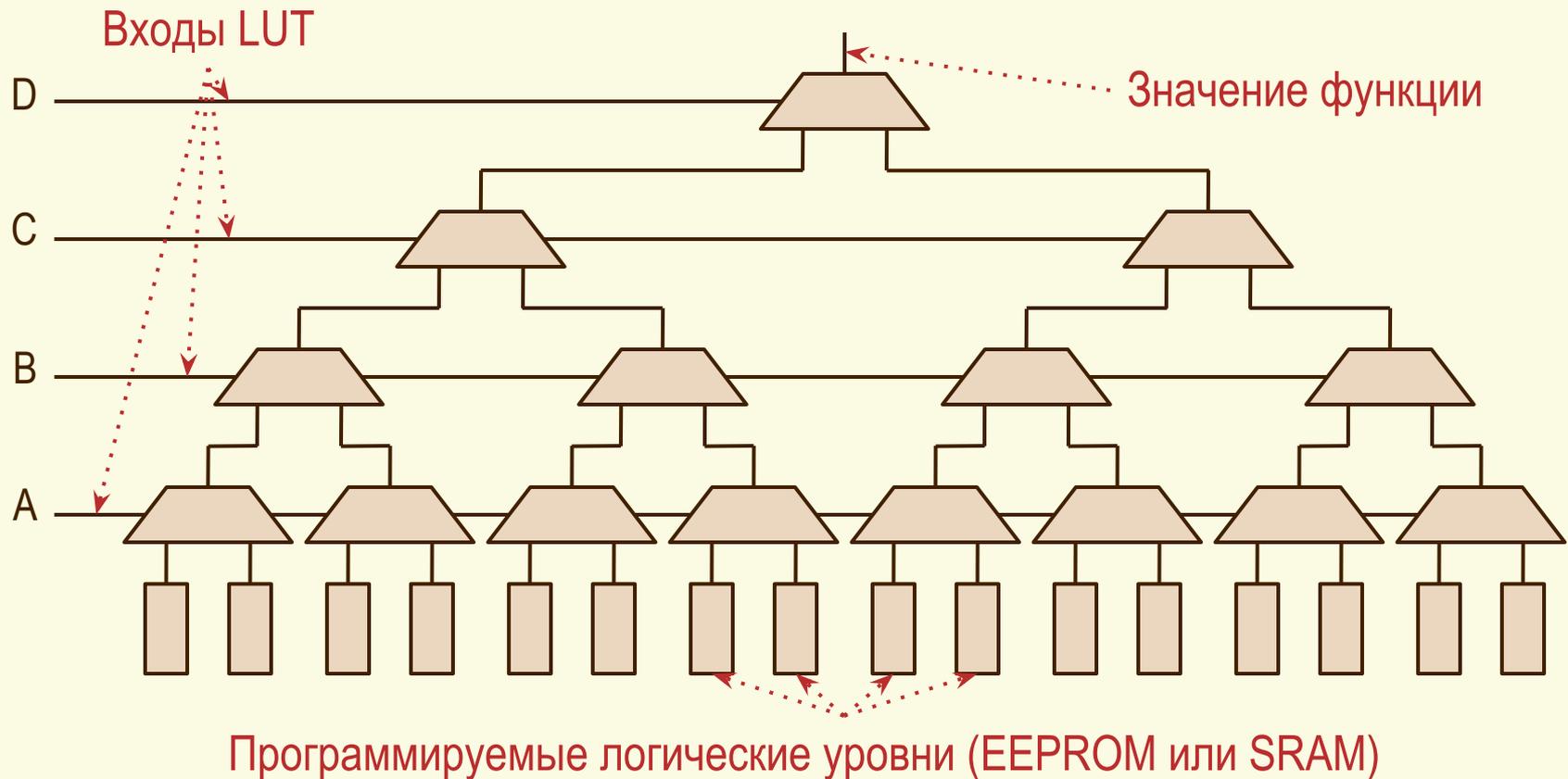


Блок
переноса

Выходной
регистр

Логическая таблица (LUT)

- Заменяет программируемую матрицу соединений в LAB CPLD
- Реализует произвольную функцию
 - Обычно не более 4 входов
 - Входы LUT выбирают значения мультиплексоров



Блок переноса и настраиваемый регистр

- Блок переноса
 - Позволяет каскадировать LE
 - Не задействует LUT и регистр при построении каскада
- Настраиваемый регистр
 - Режим работы: D-, T-, JK- или RS-триггер
 - Входы асинхронной установки и сброса
 - Подача выхода на вход LUT (обратная связь)
 - Обход регистра или LUT

Маршрутизация сигналов

- Любой LE может обмениваться информацией с любым другим LE
- Локальные соединения
 - Соединяют LE внутри LAB
 - Могут также соединять соседние LAB в FPGA
- Связи по столбцам и строкам
 - Состоят из сегментов фиксированной длины
 - Охватывают несколько LAB или всю FPGA

Блоки ввода/вывода

- Также называются элементами ввода/вывода (I/O elements)
- Подключены напрямую к связям по строкам и столбцам
- Обеспечивают настраиваемый ввод/вывод
 - Направление (вход / выход / двунаправленный)
 - Стандарт (LVTTL, LVCMOS, PCI, ...)
 - Формирование дифференциального сигнала
 - Выходной ток
 - Pull-up / pull-down
 - ...

Другие особенности FPGA

- Встроенные блоки памяти
 - Упрощают создание структур памяти (RAM, ROM, ...)
 - Могут быть инициализированы при подаче питания
- Аппаратные умножители
 - Используются в DSP
 - Высокоскоростные операции умножения / сложения
- Высокоскоростные трансиверы
- Замена некоторых LAB специализированными блоками

Синхронизация

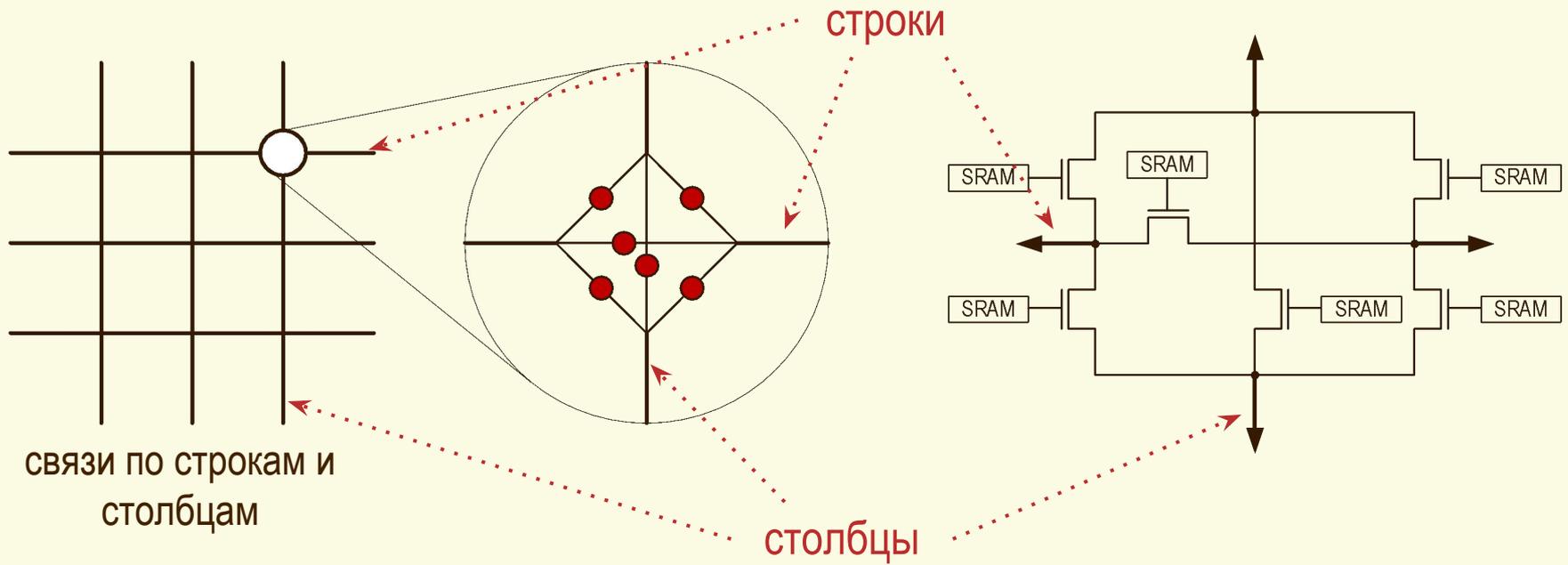
- Особенности синхросигнала
 - Очень чувствителен к временным параметрам
 - Высокий коэффициент разветвления
- Маршрутизация синхросигнала
 - **Отдельные выводы** микросхемы FPGA
 - Отдельная сеть маршрутизации (clock routing network)
 - Глобальная сеть снабжает всю FPGA
 - Региональные сети обслуживают отдельные области FPGA
- Специальные структуры для работы с синхросигналом
 - Блоки фазовой автоподстройки частоты (PLL)
 - Блоки управления синхросигналом

Структуры для работы с синхросигналом

- Блок фазовой автоподстройки частоты (**Phase-Locked Loop, PLL**)
 - Умножение и деление частоты
 - Сдвиг фазы
 - Настраиваемая скважность
 - Переключение между несколькими входными синхросигналами
 - Различные режимы компенсации задержек
- Блок управления синхросигналом (**Clock Control Block, ССВ**)
 - Является промежуточным звеном между источником синхросигнала (выводом МС, PLL или внутренней логикой) и глобальной сетью маршрутизации
 - Позволяет динамически выбирать источник синхросигнала
 - Обеспечивает отключение сети при необходимости

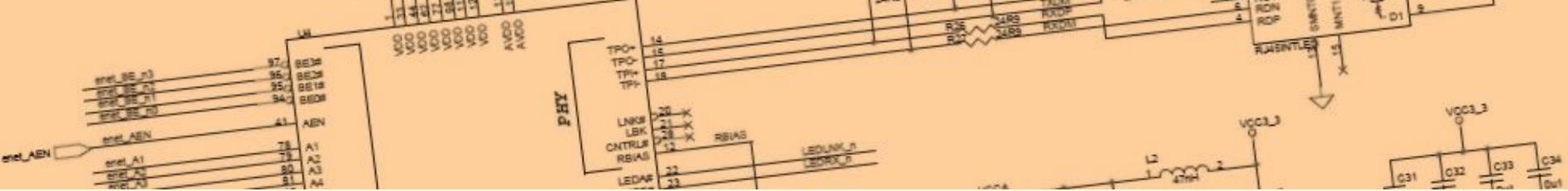
Программирование FPGA

- Большинство FPGA используют SRAM для задания связей и программирования LUT



Программирование FPGA

- SRAM является **энергозависимой**
 - Необходима инициализация после подачи питания
 - Долговременное хранение конфигурации вне FPGA
- Режимы программирования:
 - Активный: управляется FPGA после подачи питания
 - Пассивный: управляется внешним устройством (CPLD, CPU, ...)
- Возможно программирование по интерфейсу JTAG



Сравнение CPLD и FPGA

Аппаратные особенности

	CPLD	FPGA
Базовые логические компоненты	LAB, состоящие из макроячеек	LAB, состоящие из LE
Реализация логических функций	Программируемые логические матрицы	Логические таблицы
Размещение блоков	LAB размещаются вокруг PI	LAB упорядочены в двумерную матрицу
Связи	Локальные внутри LAB и глобальные произвольной структуры	Локальные внутри LAB и по строкам / столбцам
Встроенная память	Нет	Есть
Встроенные блоки цифровой обработки сигналов (DSP)	Нет	Аппаратные умножители
Технология программирования	EPROM, EEPROM или Flash	SRAM

Отличительные черты

	CPLD	FPGA
Быстродействие	От низкого до среднего	Высокое
Логическая емкость	Достаточна для небольших устройств	Тысячи и сотни тысяч LE для сложных устройств
Ввод / вывод	Программируемый	Программируемый с расширенными возможностями
Высокоскоростной обмен данными	Отсутствует	Трансиверы с поддержкой различных протоколов
Энергопотребление	Низкое	От низкого до высокого в зависимости от задействованных ресурсов
Программирование	Внутрисхемное; работоспособна сразу после подачи питания	Внутрисхемное; после подачи питания необходима загрузка конфигурации