

АРХИТЕКТУРА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Лекция 7: Вычислительные системы класса SIMD

Векторные и векторно-конвейерные ВС



Векторные ВС

Мотивация:

- много задач моделирования и обработки сигналов
 - большие регулярные массивы (в т.ч., с плавающей запятой)
 - алгоритмы: матричные и векторные операции
- скалярные ВС: только последовательная обработка
- векторная обработка: с помощью единой команды действие сразу над всеми элементами массивов



Размещение матриц в памяти

- Размещение

- Построчное размещение

- элементы строки расположены последовательно

- По столбцам

- шаг по столбцу = 1

- Характеристики вектора:

- Шаг по индексу (stride) – шаг извлечения элементов

- Длина вектора

a_{11}	a_{12}	a_{13}	a_{14}	a_{15}
a_{21}	a_{22}	a_{23}	a_{24}	a_{25}
a_{31}	a_{32}	a_{33}	a_{34}	a_{35}
a_{41}	a_{42}	a_{43}	a_{44}	a_{45}

a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{21}	a_{22}	a_{23}	a_{24}	a_{25}	a_{31}	a_{32}	a_{33}	a_{34}	a_{35}	a_{41}	a_{42}	a_{43}	a_{44}	a_{45}
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

по строке

по столбцу (шаг=5)

a_{11}	a_{21}	a_{31}	a_{41}	a_{12}	a_{22}	a_{32}	a_{42}	a_{13}	a_{23}	a_{33}	a_{43}	a_{14}	a_{24}	a_{34}	a_{44}	a_{15}	a_{25}	a_{35}	a_{45}
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

по столбцу (шаг=1)

по строке (шаг=4)

Векторный процессор

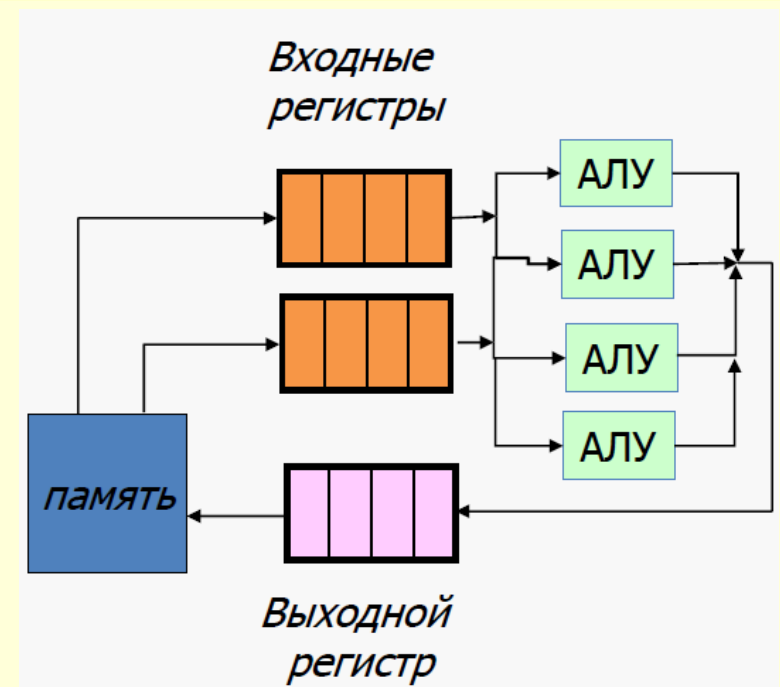
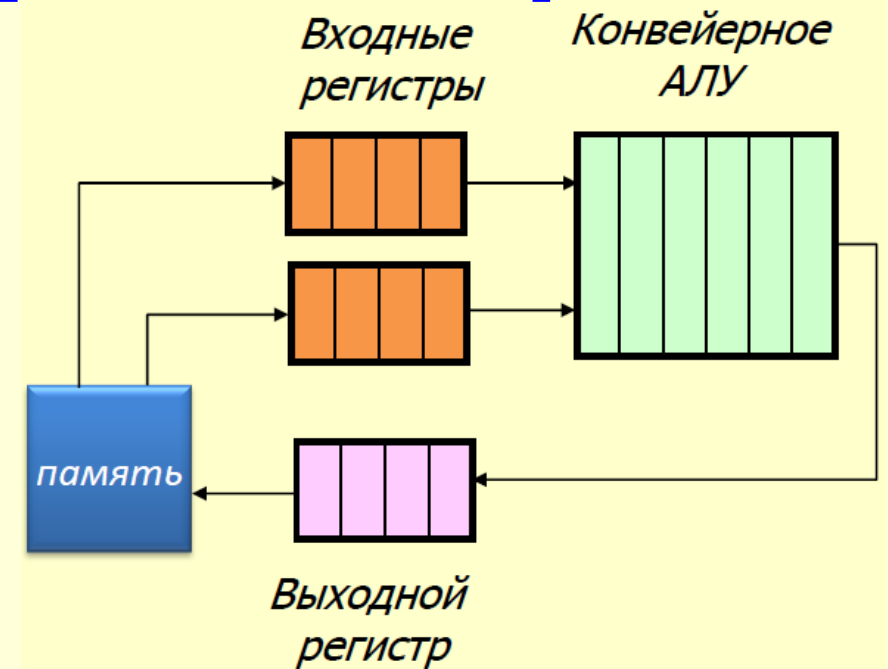
Подходы к векторной обработке:

- **Векторный процессор:** векторные операнды команд

- конвейерное АЛУ
- массив АЛУ

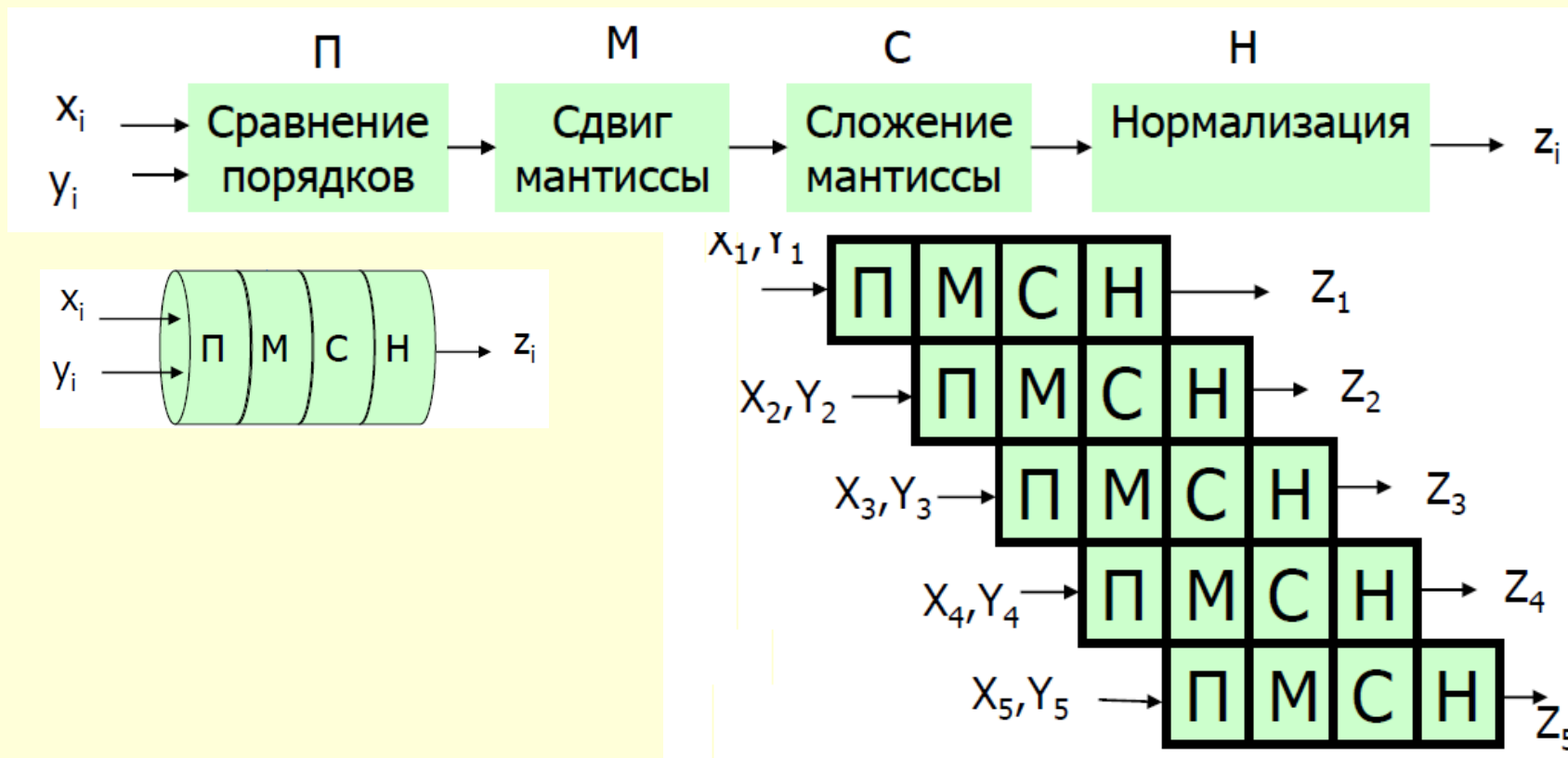
- Массив процессорных элементов

- (матричные ВС)



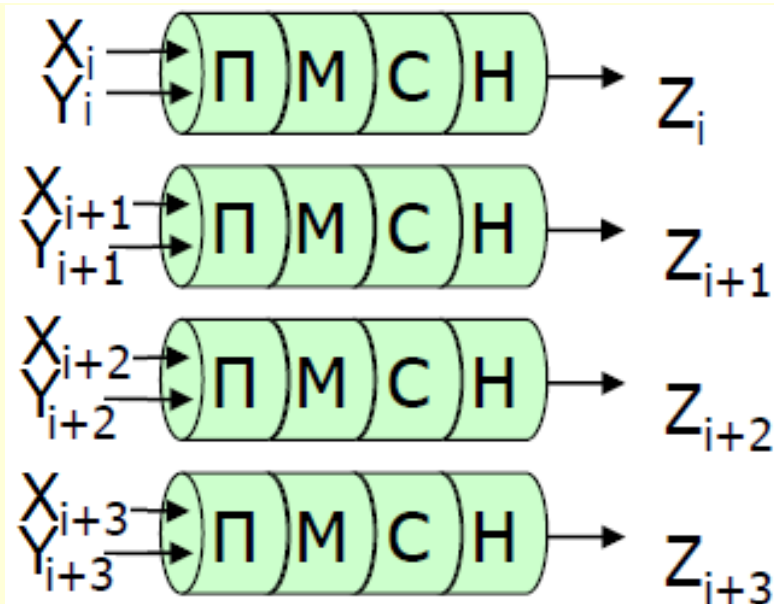
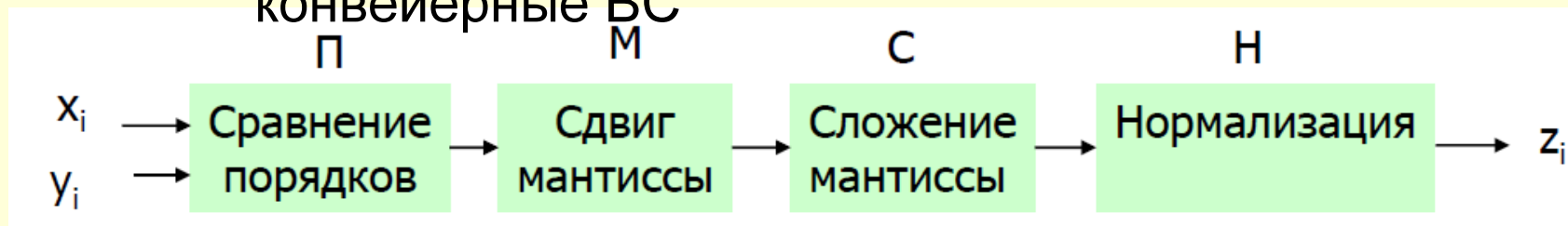
Вариант с конвейерным АЛУ

- Операции разбиваются по шагам
- Каждый этап реализуется в виде ступени конвейерного АЛУ
- Очередной элемент поступает при освобождении первой ступени конвейера

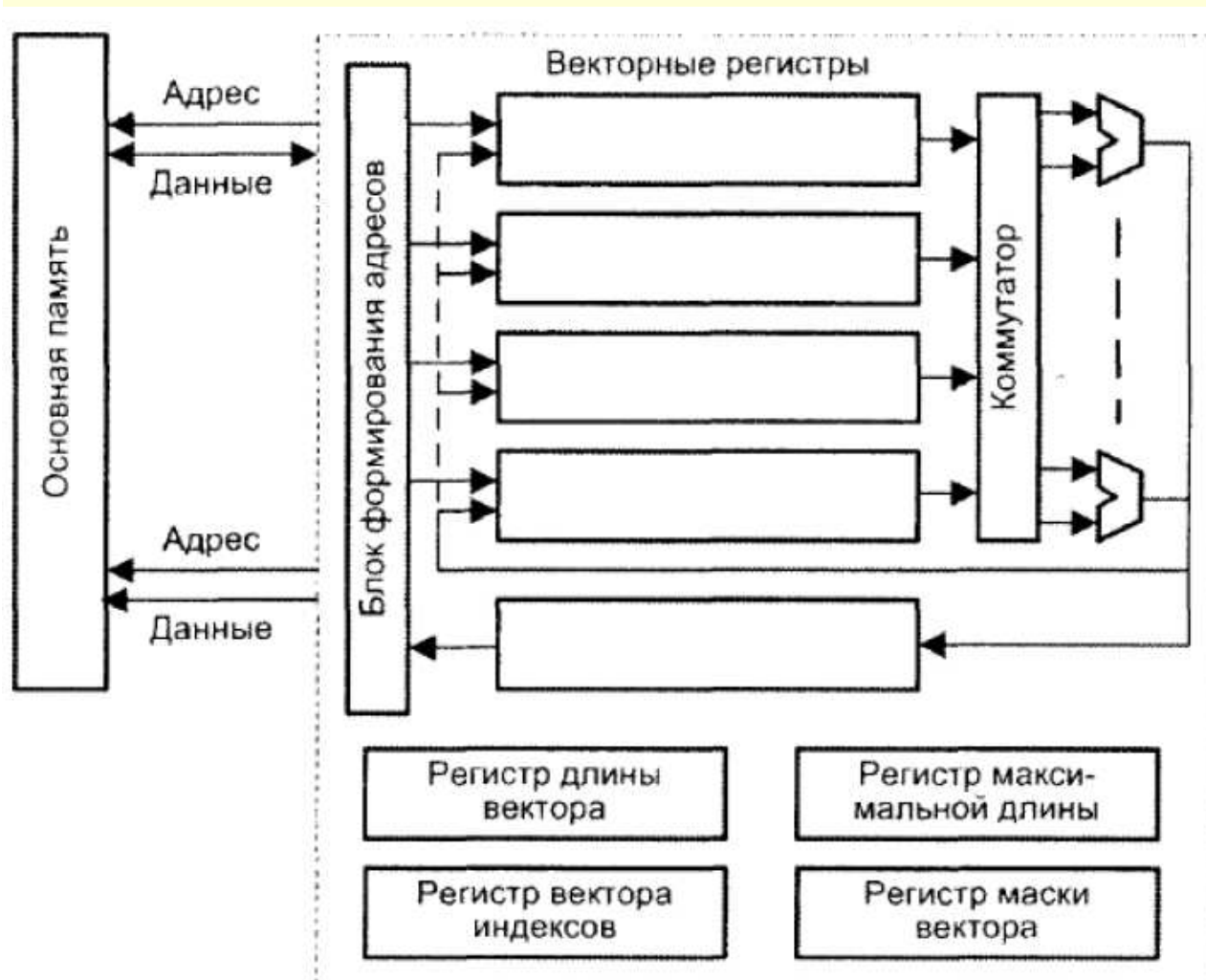


Вариант с несколькими АЛУ

- Параллельная обработка пар элементов в нескольких АЛУ
- Каждое АЛУ может быть конвейерным
 - возможен новый уровень конвейеризации – векторно-конвейерные ВС



Структура векторного процессора



- АЛУ – блоки:
 - сложения
 - умножения
 - инверсии ($1/x$)
- Векторные регистры
 - FIFO из скалярных регистров
- Система команд:
 - загрузка векторного регистра
 - поэлементные операции над вектором
 - сохранение содержимого регистра в память



Структура векторного процессора

- Регистр длины вектора
 - число элементов обрабатываемого вектора
- Регистр максимальной длины вектора
 - лимитирует число одновременно обрабатываемых элементов
 - используется при делении длинного вектора на сегменты
- Маска вектора
 - битовое поле для избирательного применения операций к элементам
- Регистр вектора индексов
 - вектор индексов – битовая маска
 - применяется для выборки элементов из вектора (операции упаковки/распаковки)



Пример векторной операции: умножение матриц

$$\begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \times \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \quad 1$$

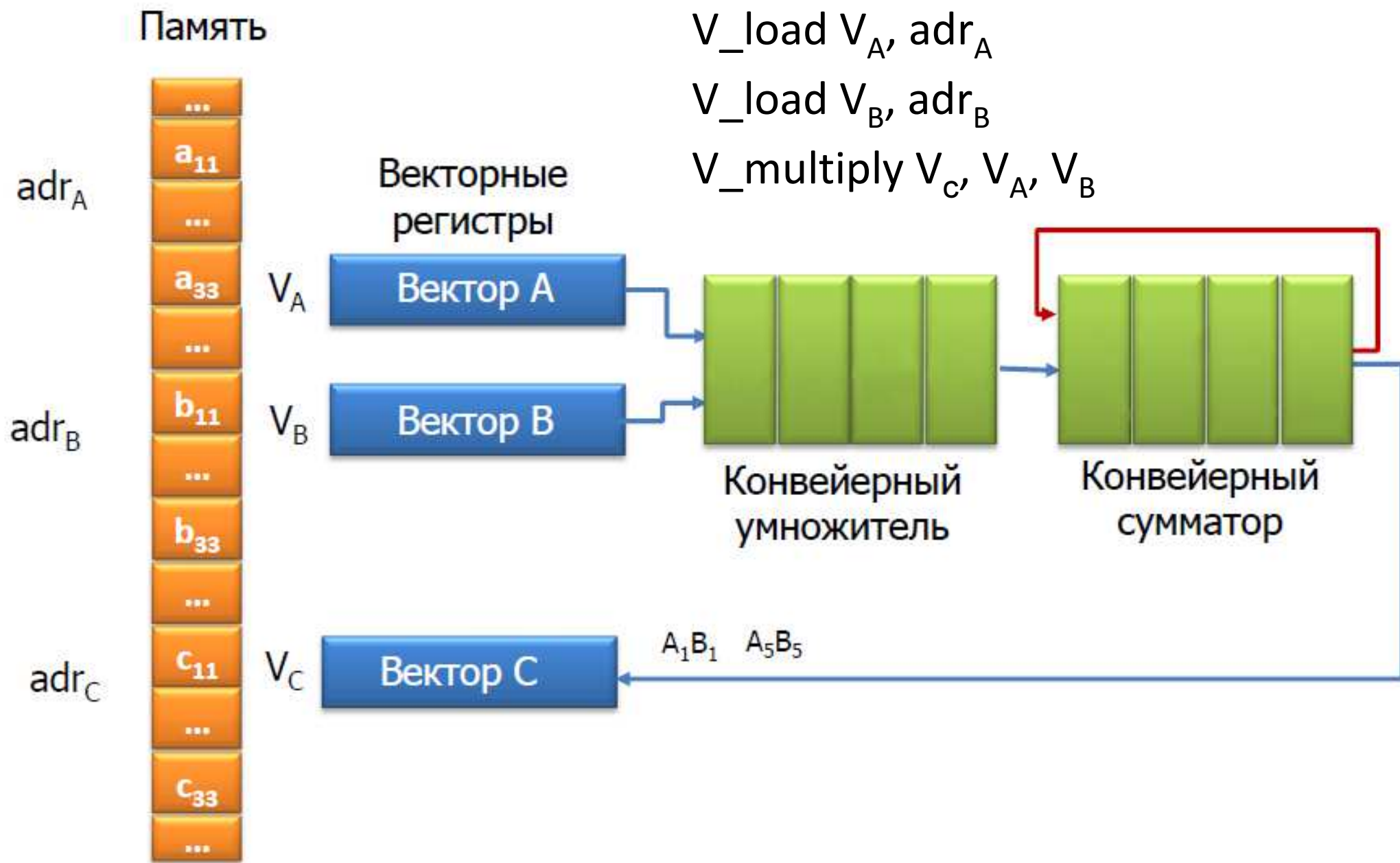
$$c_{ij} = \sum_{k=1}^3 a_{ik} \times b_{kj} \quad 2$$

$$c_{11} = a_{11} \times b_{11} + a_{12} \times b_{21} + a_{13} \times b_{31} \quad 3$$

$$c = a \times b \quad 4$$

$$C = A_1 B_1 + A_2 B_2 + A_3 B_3 + A_4 B_4 + \dots + A_k B_k \quad 5$$

Векторный процессор для вычисления скалярного произведения



Преимущества векторных ВС

- Эффективная выборка команд
 - только одна команда для набора операций
 - снижаются издержки управляющего устройства
 - ниже требования к пропускной способности памяти
- Упорядоченные данные
 - чередование адресов увеличивает скорость загрузки данных из памяти



Структуры типа «память-память» и «регистр-регистр»

По способу доступа к операндам:

- **«память-память»**
 - элементы векторов извлекаются из памяти поочерёдно и непосредственно в ФБ
 - результат сразу заносится в память
 - более гибкая схема, но выше издержки
- **«регистр-регистр»**
 - операнды загружаются в векторные регистры
 - нужно сегментировать длинные векторы, но ниже время запуска

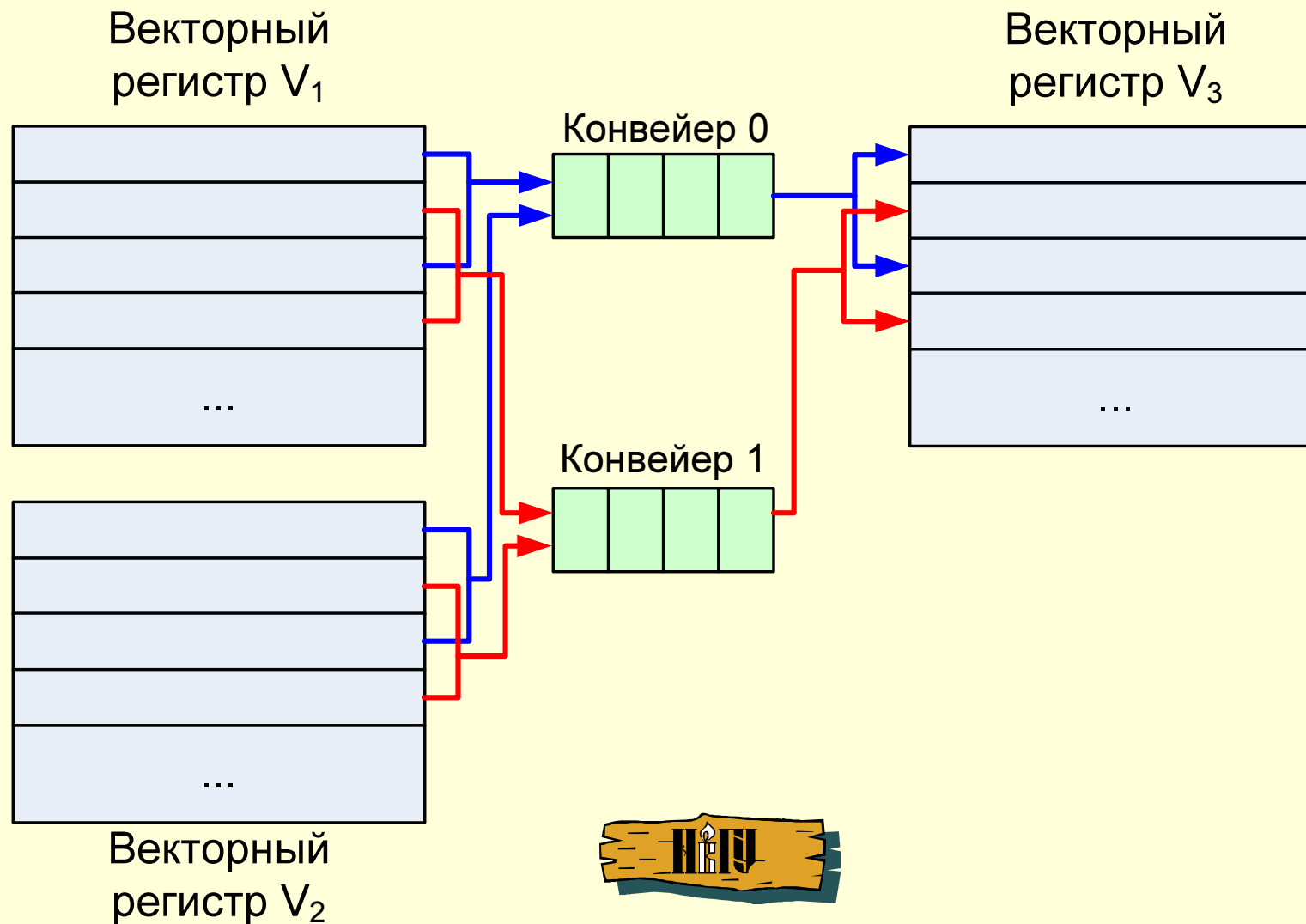
Время работы процессора: $T = s + a \times N$

- s – время запуска
- a – константа (зависит от типа команды, обычно $\frac{1}{2}$, 1 или 2)
- N – длина вектора



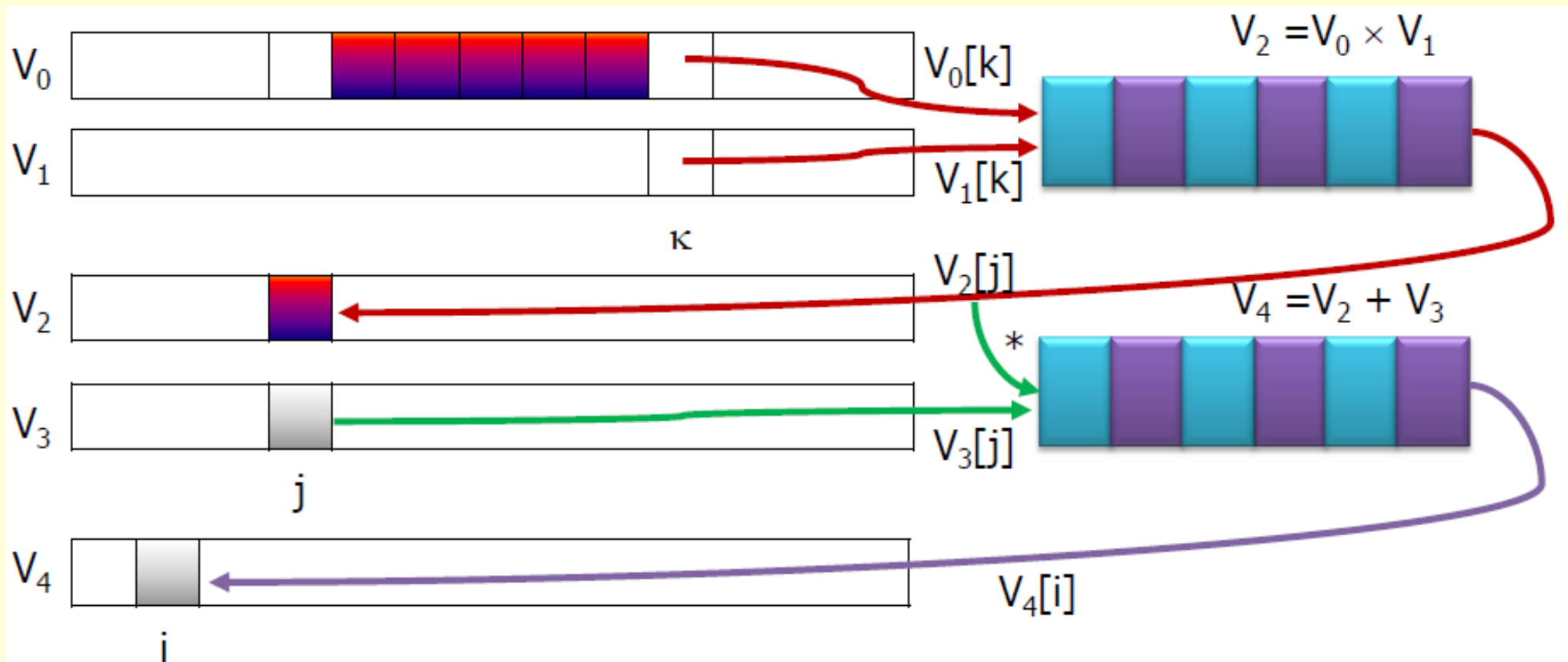
Ускорение вычислений

- Дублирование конвейеров всех функциональных блоков



Ускорение вычислений

- **Сцепление векторов** в схемах «регистр-регистр»
 - векторный регистр результат одной операции используется в качестве входного регистра следующей операции
 - в системах «память-память» реализация требует дополнительных каналов



Матричные ВС



Матричные ВС

- Назначение – обработка больших массивов данных
- В основе – матричный процессор
 - регулярный массив процессорных элементов
 - общее управляющее устройство (один поток команд)
 - все процессорные элементы работают синхронно



Структура матричной ВС



- Параллельная обработка выполняется массивом процессоров (МП)
- Единый поток команд генерируется контроллером МП
- По шине широковещательной рассылки передаются команды, данные и сигналы управления



Структура матричной ВС

- Интерфейсная ВМ (front-end computer)
 - универсальная ЭВМ
 - интерфейс программирования и отладки
 - в процессе выполнения загружает программу в контроллер управления МП
- Массив процессоров
 - процессоры, память, сеть межсоединений
- Механизм маскирования
 - к вычислениям привлекается подмножество процессоров на основе маски
 - на этапе компиляции
 - во время выполнения



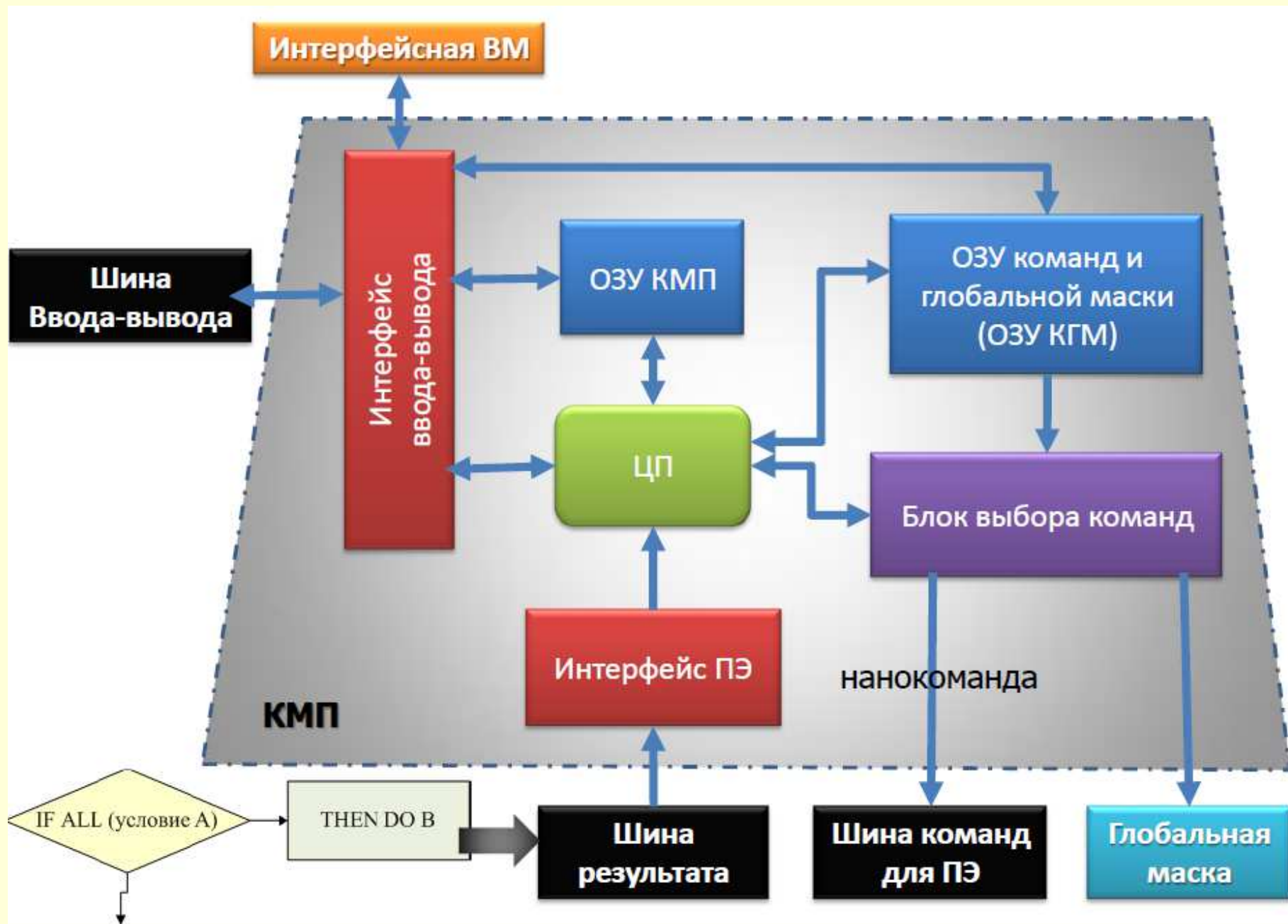
Структура матричной ВС

- Контроллер массива процессоров
 - программа при загрузке заносится в ОЗУ КМП
 - команды для ПЭ и глобальная маска (компил.) заносятся в ОЗУ команд и глобальной маски
 - затем КМП начинает исполнение
 - извлекает скалярную команду из ОЗУ КМП
 - команды для ЦП
 - или множественные команды из ОЗУ КГМ
 - преобразуются в блоке выборки в **нанокоманды**
 - нанокоманды + маска пересылаются в ПЭ на исполнение



Структура матричной ВС

Контроллер массива процессоров

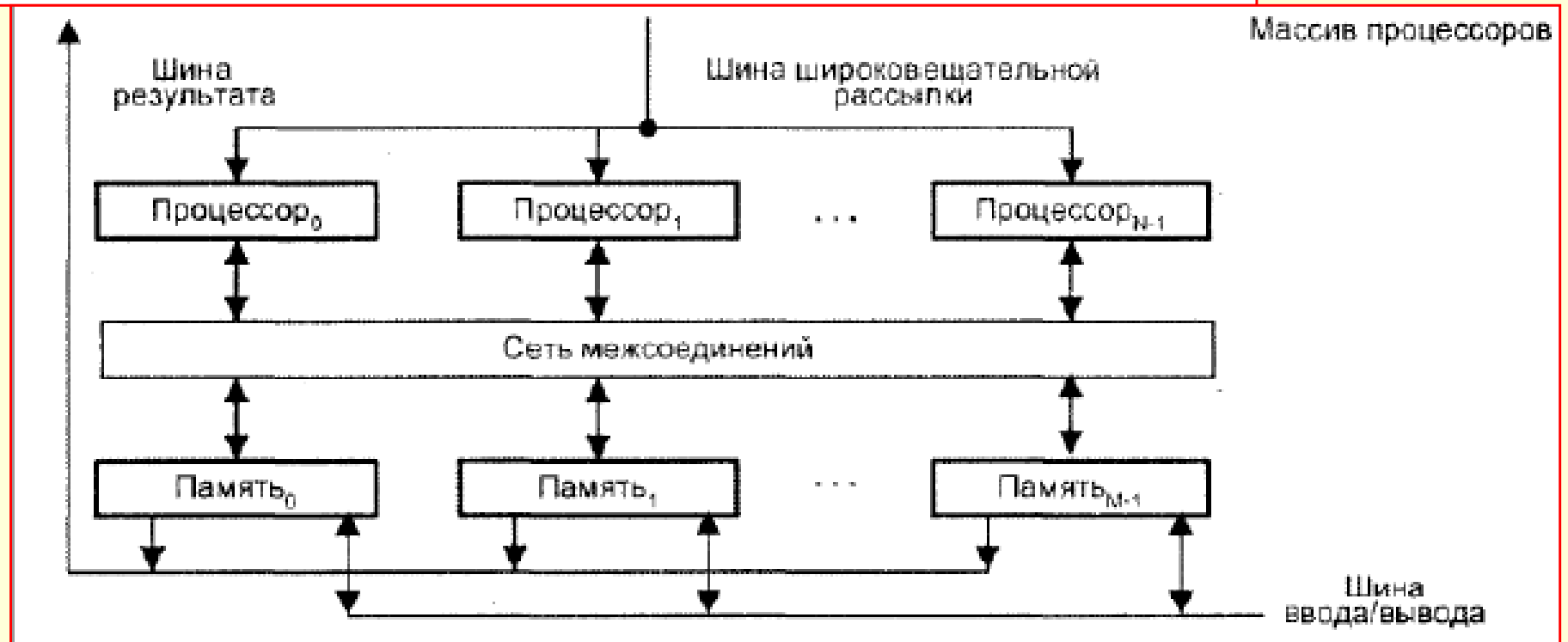
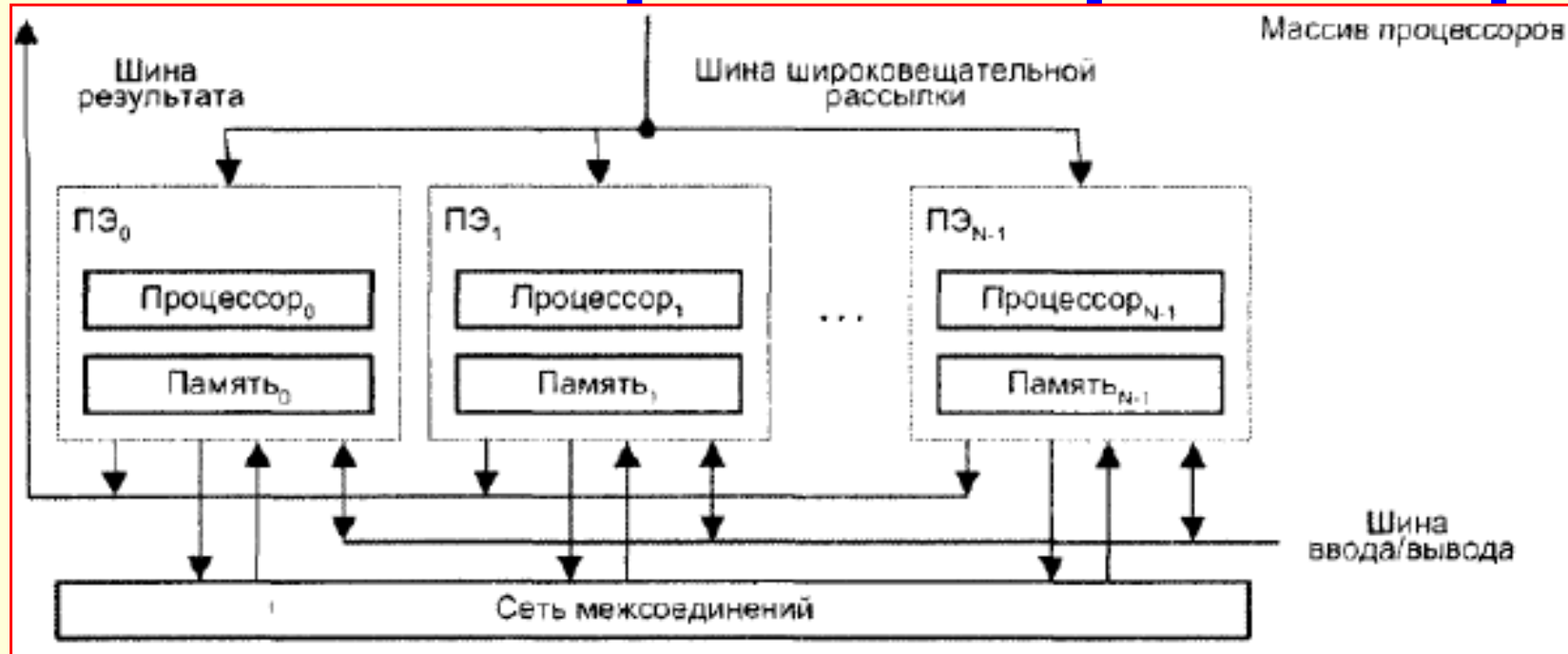


Структура матричной ВС

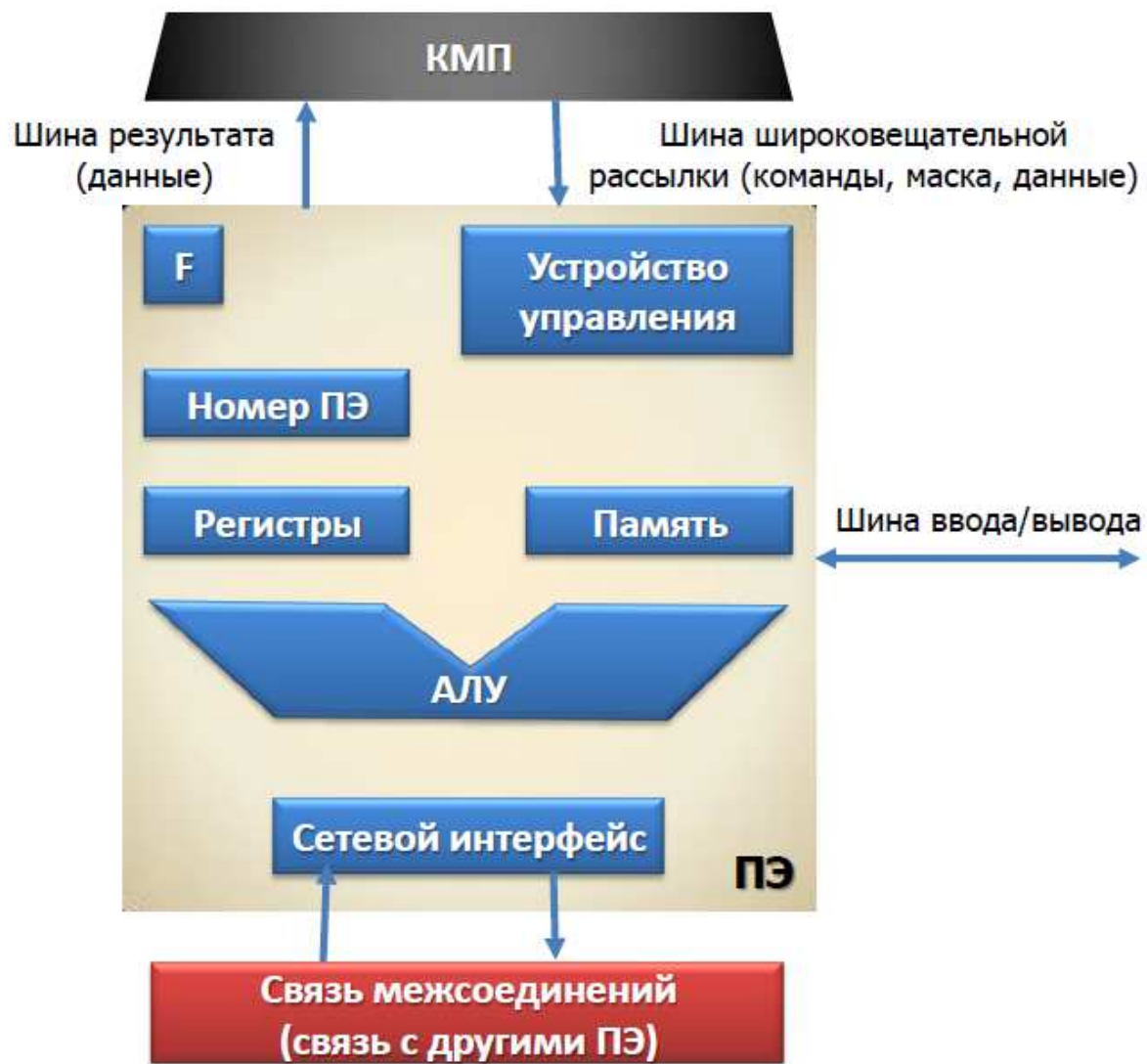
- Массив процессоров
 - архитектура «ПЭ-ПЭ»
 - ПЭ связаны между собой сетью межсоединений
 - каждый ПЭ – процессор с локальной памятью
 - обмен данными по сети
 - шина ввода-вывода – к периферийным устройствам
 - шина результата – трансляция результата из ПЭ в КМП
 - примеры: MasPar MP-1, Connection Machine CM-2, GF11, DAP, MPP, STARAN, PEPE, ILLIAC IV
 - архитектура «процессор-память»
 - двунаправленная сеть связывает N процессоров с M модулями памяти
 - обмен данными между ПЭ – через сеть или память
 - шина ввода-вывода – между модулями памяти и к периферийным устройствам
 - примеры: Burroughs Scientific Processor (BSP), Texas Reconfigurable Array Computer TRAC.



Массив процессоров: варианты



Структура процессорного элемента



- Типичный ПЭ – RISC-процессор
- Реализация массива в виде одной микросхемы
- Основные компоненты:
 - АЛУ
 - регистры данных
 - сетевой интерфейс
 - номер процессора (адрес ПЭ)
 - регистр флага разрешения маскирования (F)
 - локальная память



Маскирование процессорных элементов

- На этапе компиляции
- Во время выполнения
 - маскирование, определяемое данными
 - ПЭ самостоятельно определяет свой статус
 - маскируемые команды в системе команд
 - исполняются в зависимости от флага маскирования
 - глобальное маскирование
 - КМП посылает глобальную маску во все ПЭ вместе с командами
 - маска может быть битовой или содержать шаблон адресов ПЭ



Ассоциативные ВС



Ассоциативные ВС

- В основе – ассоциативный процессор
 - построен на базе ассоциативного ЗУ
 - выборка информации по ассоциативным признакам, а не по адресу операнда
 - запись не по адресу, а в свободную ячейку
 - мультizaпись – параллельная запись во все ячейки, совпадающие с ассоциативным признаком
 - считывание и запись – по двум срезам запоминающего массива
 - все разряды одного слова
 - один и тот же разряд всех слов
 - с маскированием лишних позиций
 - каждый разряд среза снабжён ПЭ
 - параллельно выполняются операции над данными



Ассоциативные ВС

- n ПЭ (вертикальный разрядный срез памяти) – простые устройства поразрядной обработки
- каждое слово (ячейка) памяти имеет собственное устройство обработки (сумматор)
- Операция осуществляется синхронно всеми ПЭ

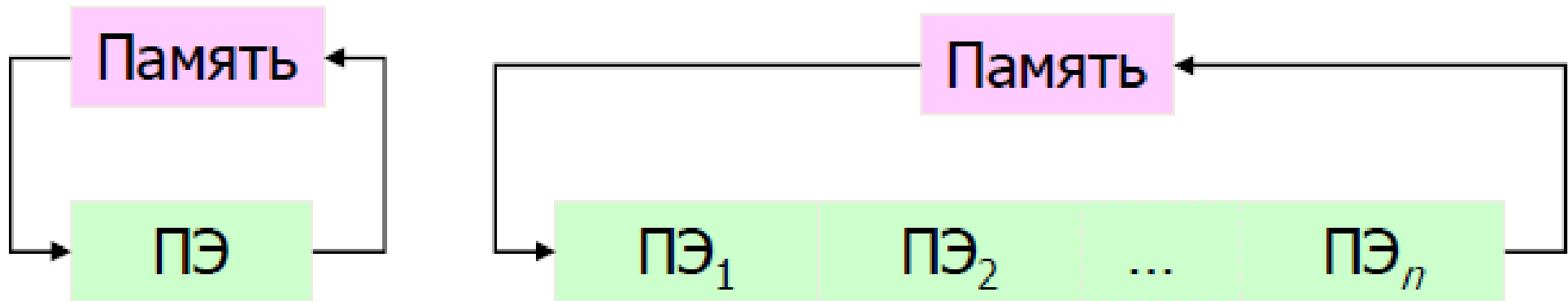


ВС с систолической структурой



Систолические ВС

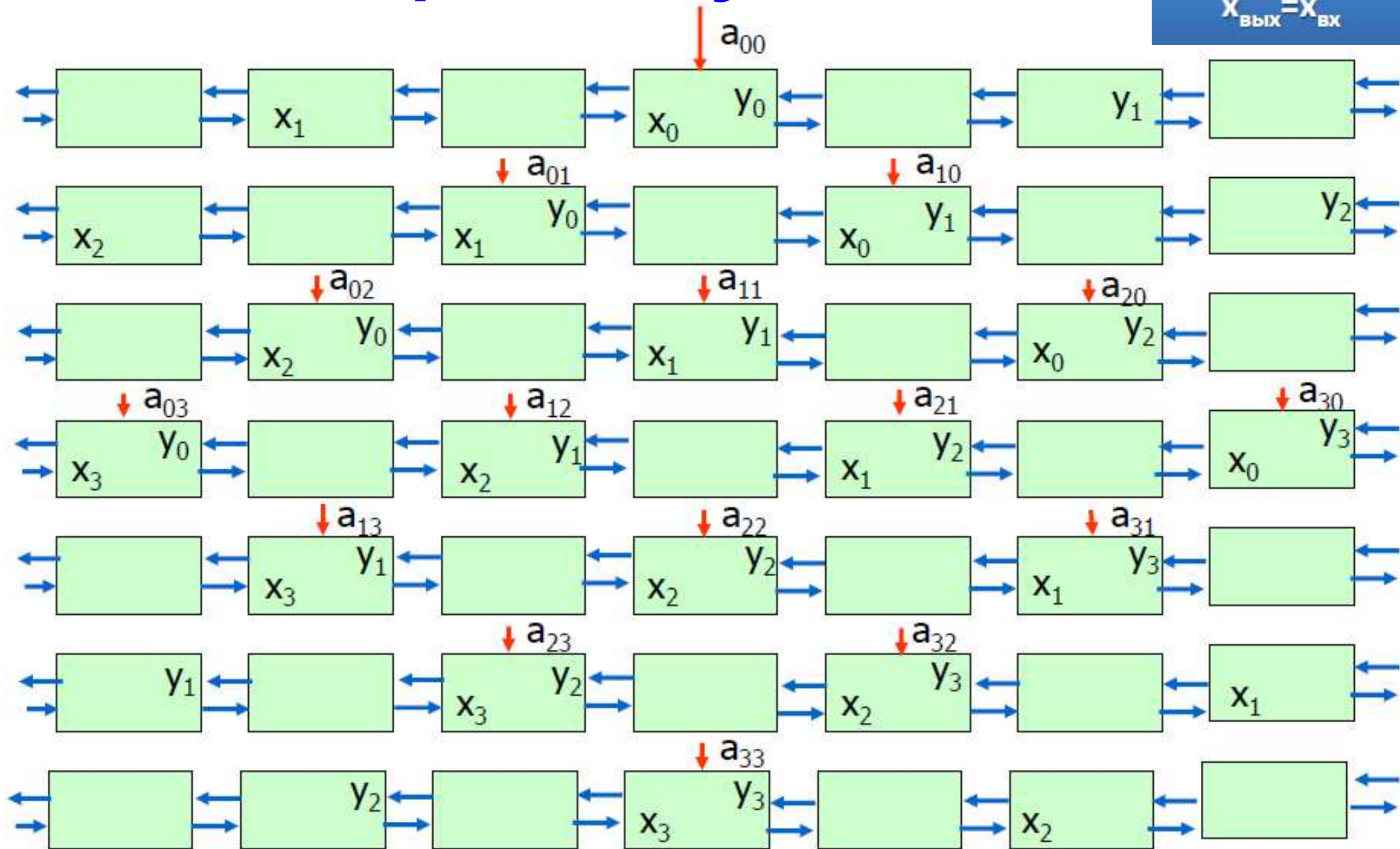
- Идея: пропускать данные на пути от считывания до записи через максимальное число ПЭ
- Эффективность
 - матричные вычисления
 - обработка сигналов
 - сортировка данных
 - и т.д.



Систолические ВС

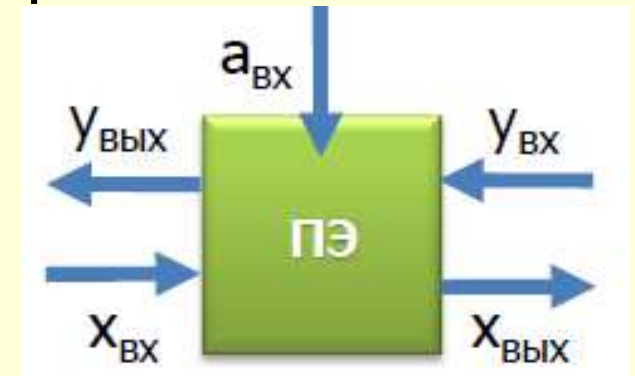
матричное умножение

$$y_{\text{ВЫХ}} = y_{\text{ВХ}} + x_{\text{ВХ}} * a_{\text{ВХ}}$$
$$x_{\text{ВЫХ}} = x_{\text{ВХ}}$$



Систолические ВС

- В основе схемы – ритмическое прохождение двух потоков данных навстречу друг другу
- Последовательные элементы разделены одним тактом
- Вычисления выполняют ПЭ
 - один шаг скалярного произведения – IPS-элементы (Inner Product Step)



Особенности систолических ВС

- совмещают свойства конвейерной и матричной обработки
- вычисления: непрерывная передача данных между ПЭ без сохранения промежуточных результатов
- входные данные выбираются однократно, ввод – через крайние ПЭ матрицы
- ПЭ однотипны и узкоспециализированы
- регулярные потоки данных и управляющих сигналов – позволяют объединять ПЭ связями минимальной длины
- производительность линейно улучшается добавлением новых ПЭ



Классификация систолических ВС

- По степени гибкости:
 - специализированные
 - жёсткий алгоритм
 - определённые топология систолической структуры и тип операции
 - алгоритмически ориентированные
 - программируемые связи в матрице
 - либо программируемые ПЭ
 - программируемые
 - возможность программирования самих ПЭ и связей между ними
 - локальная память ПЭ, различные операции в разных ПЭ
 - смена направления передачи операндов



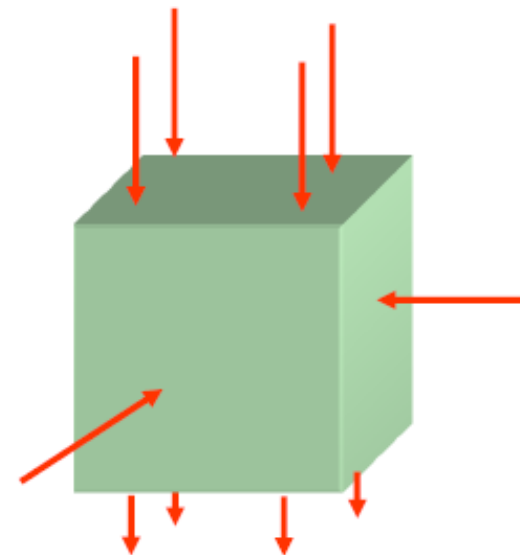
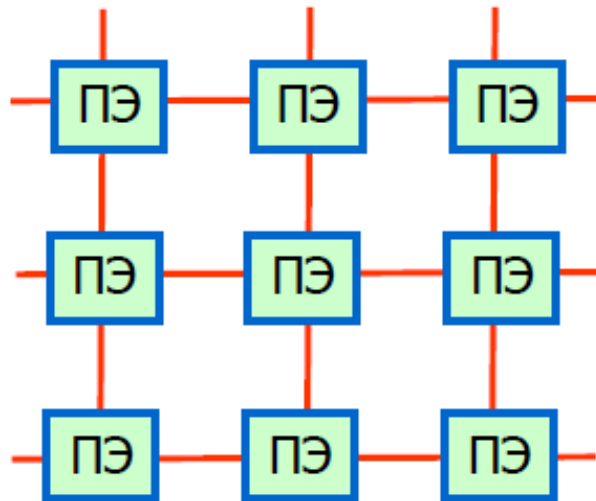
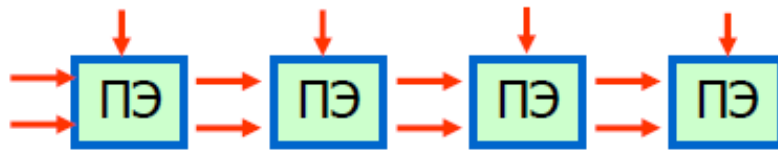
Классификация систолических ВС

- По разрядности процессорных элементов:
 - одноразрядные
 - ПЭ выполняют операцию над одним двоичным разрядом
 - многоразрядные
 - операции над словами фиксированной длины
- По характеру локально-пространственных связей:
 - одномерные
 - двумерные
 - трехмерные



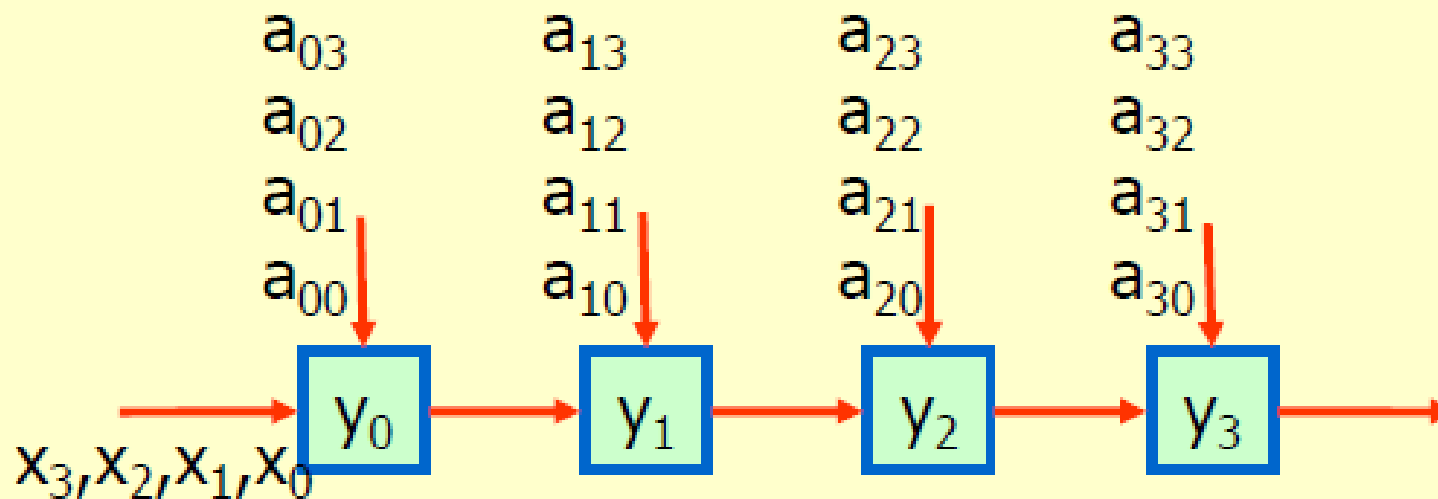
Топология систолических структур

- Систолическая структура может представлять собой:
 - набор отдельных матриц
 - сложную сеть взаимосвязанных матриц
 - обрабатывающую поверхность
 - бесконечная прямоугольная сетка ПЭ, связанных с 4 соседями



Топология систолических структур

- **ULA** (Unidirectional Linear Array) –
однонаправленный линейный процессорный
массив

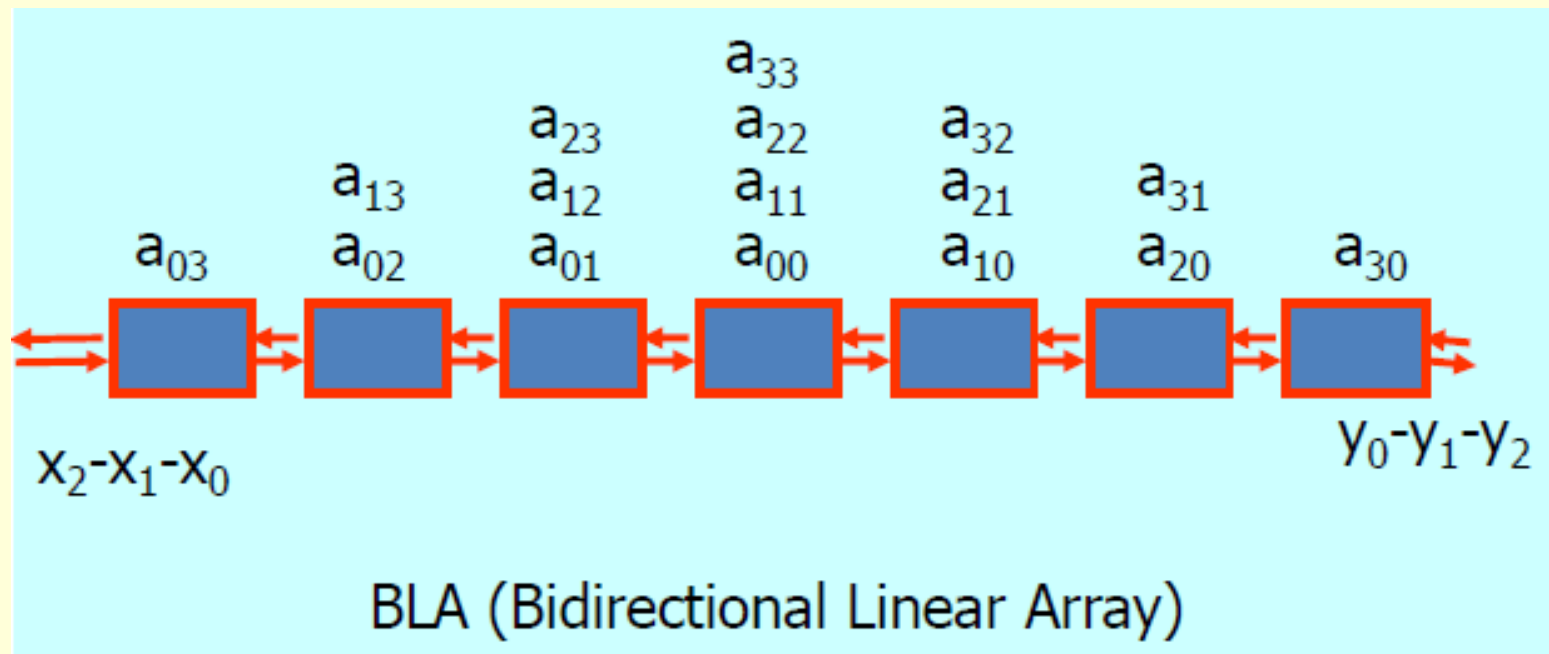


ULA (Unidirectional Linear Array)



Топология систолических структур

- **BLA** (Bidirectional Linear Array) — это двунаправленный линейный процессорный массив,
 - два потока данных движутся навстречу друг другу.
- Регулярный BLA – BLA, где один из потоков является

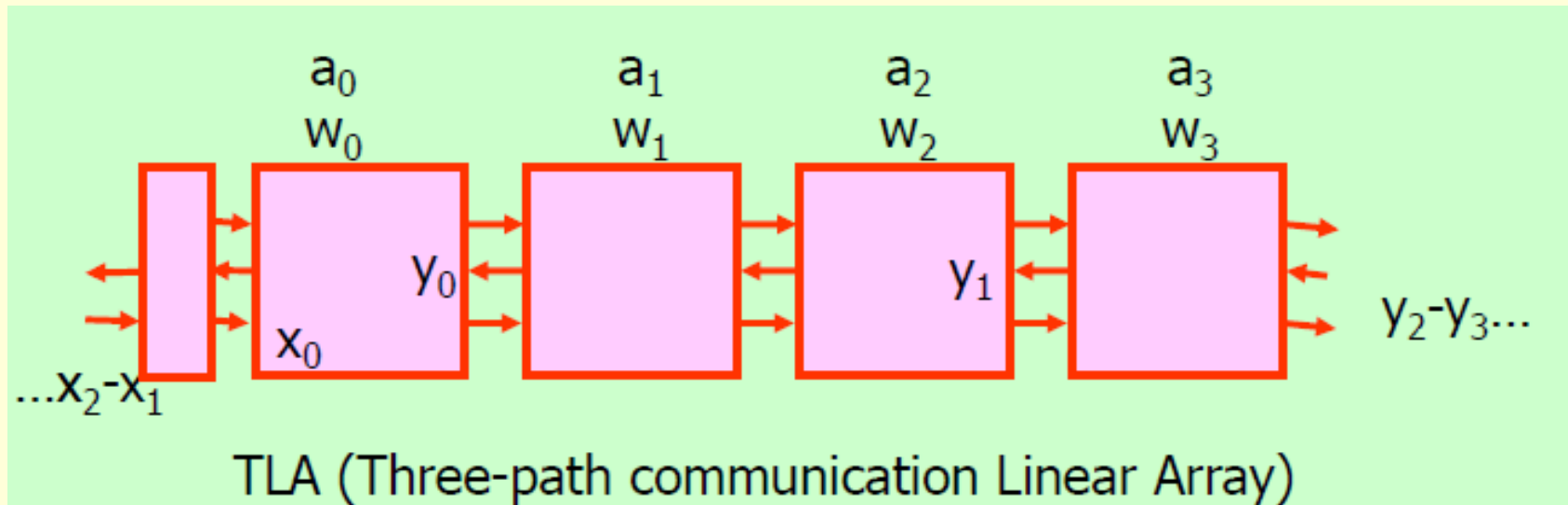


Топология систолических структур

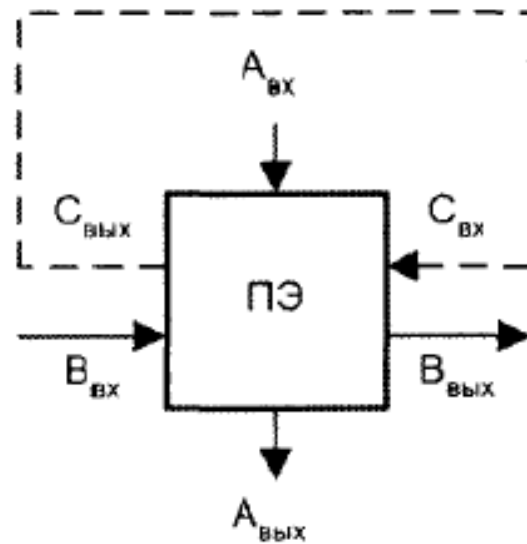
- **TLA** (Three-path communication Linear Array) — линейный процессорный массив с тремя коммуникационными трактами
 - по разным направлениям перемещаются три потока данных
 - часто называют сдвоенным конвейером (можно разделить на 2 BLA)

Пример фильтра ARMA, предложенного Кунгом и построенного по схеме TLA:

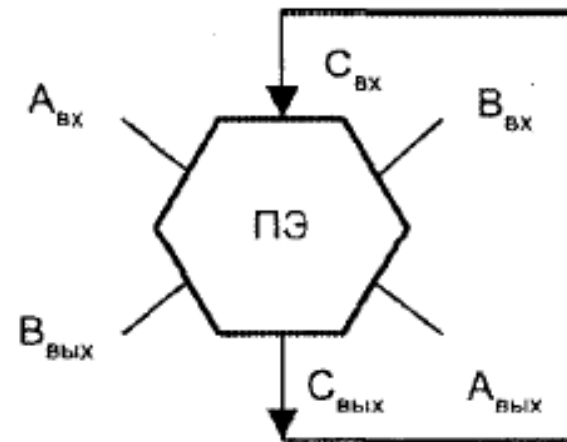
- ПЭ могут использовать как хранимые в памяти значения, так и внешние данные



Структура процессорных элементов



а



б

Структура ПЭ:

а - для прямоугольной систолической матрицы;
б - для гексагональной систолической матрицы

$$C_{\text{вых}}^{(n)} = C_{\text{вх}}^{(n-1)} + A_{\text{вх}}^{(n-1)} \times B_{\text{вх}}^{(n-1)}$$

$$A_{\text{вых}}^{(n)} = A_{\text{вх}}^{(n-1)}, B_{\text{вых}}^{(n)} = B_{\text{вх}}^{(n-1)}$$