НАНОТЕХНОЛОГИЯ НА ПУТИ К РЕЛЯТИВИСТСКИМ КОМПЬЮТЕРАМ

Проф. Воробьёв Владимир Анатольевич
Архангельск, ПГУ им. М.В. Ломоносова,
Математический факультет
vva100@atnet.ru

ПРОБЛЕМА БЫСТРОДЕЙСТВИЯ

Фундаментальная задача современной науки и техники - достижение быстродействия компьютеров

$$10^{12} \div 10^{16}$$ **Flops и более.**

От решения этой проблемы зависит дальнейшее продвижение во многих стратегически важных отраслях науки, техники, экономики.

ИСТОРИЯ СУПЕРКОМПЬЮТЕРОВ В РОССИИ

1958 – **клеточные автоматы** фон Неймана (теория)

1962 – **однородная машина** Холланда (теория).

1964 – **ОВС МИНСК-222** (Э.В.Евреинов, Ю.Г.Косарев)

1975 – **ОВС СУММА** (В.Г. Хорошевский, В.А.Воробьёв, С. Г.Седухин и др.)

1978 – **ОВС МИНИМАКС** (В.Г. Хорошевский, Ю.К. Димитриев, Л.С. Шум, Н.Н. Меренков и др.)

1980 – **ПС-2000** (И.В.Прангишвили, С.И.Медведев, Я.С. Виленкин и др.)

1995 – **МВС-100** (А.В.Забродин, В.К.Левин, В.В.Корнеев)

2000 – **МВС-100Х** (А.В.Забродин, В.К.Левин, В.В.Корнеев)

2005-2010 – Вычислительные кластеры «Чебышев» и «Ломоносов», ВЦ МГУ

ПУТИ ПОВЫШЕНИЯ БЫСТРОДЕЙСТВИЯ

1. БУФЕРИЗАЦИЯ,

обеспечивающая сглаживание потоков команд и данных. Буферизация применяется, в первую очередь, при организации иерархической памяти компьютера: сверхбыстродействующая регистровая память, КЕШ, основное ОЗУ, встроенные накопители, сменные накопители.

Не менее успешно применяются и буферы команд, которые открывают новые возможности для управления счётом: конвейер обработки команд, захват циклов, планирование загрузки устройств и так далее.

ПУТИ ПОВЫШЕНИЯ БЫСТРОДЕЙСТВИЯ

2. КОНВЕЙЕРИЗАЦИЯ,

позволяющая ускорить исполнение повторяющихся последовательностей действий при приемлемом увеличении оборудования.

Наиболее известны: векторные арифметические устройства (**микроконвейеры**).

Конвейеризации поддаются и потоки команд (**командные магистрали**),

и последовательности этапов обработки (**макроконвейеры**).

ПУТИ ПОВЫШЕНИЯ БЫСТРОДЕЙСТВИЯ

3. **СТРУКТУРНАЯ РЕАЛИЗАЦИЯ**,

позволяющая настроить аппаратуру специально для решения данной задачи. **Структурная (аппаратурная) реализация – основа вычислительной техники.** В современных компьютерах структурно реализованы алгоритмы управления внешними устройствами, функциональные устройства для отдельных команд, протоколы обменов. Архитектура компьютера с сокращённой системой команд (RISC-архитектура) открывает возможности для структурной реализации команд и соответствующего роста производительности.

На системном уровне структурная реализация состоит в соединении элементов системы в соответствии со структурой задачи – **программировании структуры.**

ПУТИ ПОВЫШЕНИЯ БЫСТРОДЕЙСТВИЯ

4. РАСПАРАЛЛЕЛИВАНИЕ,

позволяющее максимально использовать естественный параллелизм вычислений и привлечь к вычислениям большое число процессоров. Параллелизм используется повсеместно:

1. параллельное считывание слов в расщеплённом ОЗУ,
2. параллельные процессоры для управления периферией,
3. параллельно работающие векторные и скалярные арифметические устройства конвейерных супер-компьютеров,
4. многопроцессорные системы и суперкомпьютеры,
5. **Однородные Вычислительные Системы (ОВС)** и их зарубежные реализации − Массово Параллельные Процессоры (МПП).

ПУТИ ПОВЫШЕНИЯ БЫСТРОДЕЙСТВИЯ

5. **БАЛАНСИРОВКА НАГРУЗКИ**,

то есть такое планирование работ, при котором вся аппаратура загружается равномерно и не простаивает в ожидании. Методы балансировки могут быть как аппаратурными, так и программными и зависят от архитектуры компьютера.

Упомянутые принципы встречаются во всех современных компьютерах.

ИСТОЧНОКИ ПАРАЛЛЕЛИЗМА

Существует два источника параллелизма

1. **Независимые операторы** алгоритма.
2. **Независимые вычисления** над разными данными.

Независимость операторов дает параллелизм, ограниченный малым числом операторов.

Независимые вычисления над данными могут выполняться параллельно, причем число параллельных ветвей вычисления задается количеством получаемых результатов, обычно очень большим.

Иными словами, в первом случае стоит задача распараллеливания на множестве операторов в заданном алгоритме, во втором случае распараллеливается множество возможных исполнений этих операторов.

Второе множество значительно больше первого.

СИЛА ПАРАЛЛЕЛИЗМА

Пусть происходит умножение матриц $C=A\cdot B$ размера ($N\times N$). Тогда в каждом витке ijk-цикла выполняется

$$c_{ij}^{(k+1)} := a_{ik} \times b_{kj} + c_{ij}^{(k)}, \; k = 1,\ldots, N$$

Если иметь N^2 вычислителей для каждого элемента c_{ij} матрицы C, вычисление было бы завершено в N^2 раз быстрее. Такова «сила параллелизма».

Емкостная сложность последовательного алгоритма $O(3N^2)$. В параллельном способе каждый из N^2 вычислителей должен иметь $2N$ чисел в качестве исходных данных и одно число c_{ij} на выходе. Емкостная сложность, следовательно,

$$O(2N^3 + N^2).$$

Если мы хотим сохранить оценку $O(3N^2)$, мы должны обеспечить одну из следующих альтернатив.

СИЛА ПАРАЛЛЕЛИЗМА

1. Работу всех N^2 вычислителей над общей памятью.

2. Хранение в памяти каждого вычислителя минимальной части информации (3 числа: c_{ik}, b_{kj}, c_{ij}) и обмен данными N^3 раз.

Параллельные компьютеры, реализующие первую альтернативу, называются *системами с общей* (*разделяемой*) памятью.

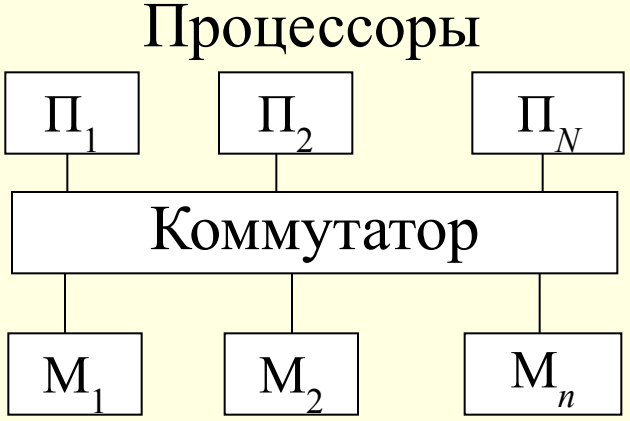
Параллельные компьютеры, реализующие вторую альтернативу, называются *системами с локальной* (*распределенной*) памятью.

В системах с общей памятью коммутатор соединяет процессоры с блоками расщеплённой памяти.

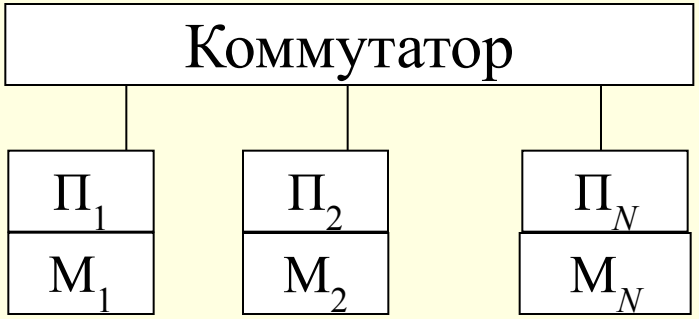
В системах с локальной памятью коммутатор соединяет только процессоры

Поэтому иногда говорят о системах с *разделённым* и *неразделённым* коммутатором.

ДВА КЛАССА ПАРАЛЛЕЛЬНЫХ СИСТЕМ



(а)



(б)

Два класса вычислительных систем:

а) мультипроцессоры с общей памятью,

б) мультикомпьютеры с локальной памятью.

КЛАССИФИКАЦИЯ ПАРАЛЛЕЛЬНЫХ СИСТЕМ

Э.Таннен-баум. Архи-тектура компью-тера. – Питер, 2006.

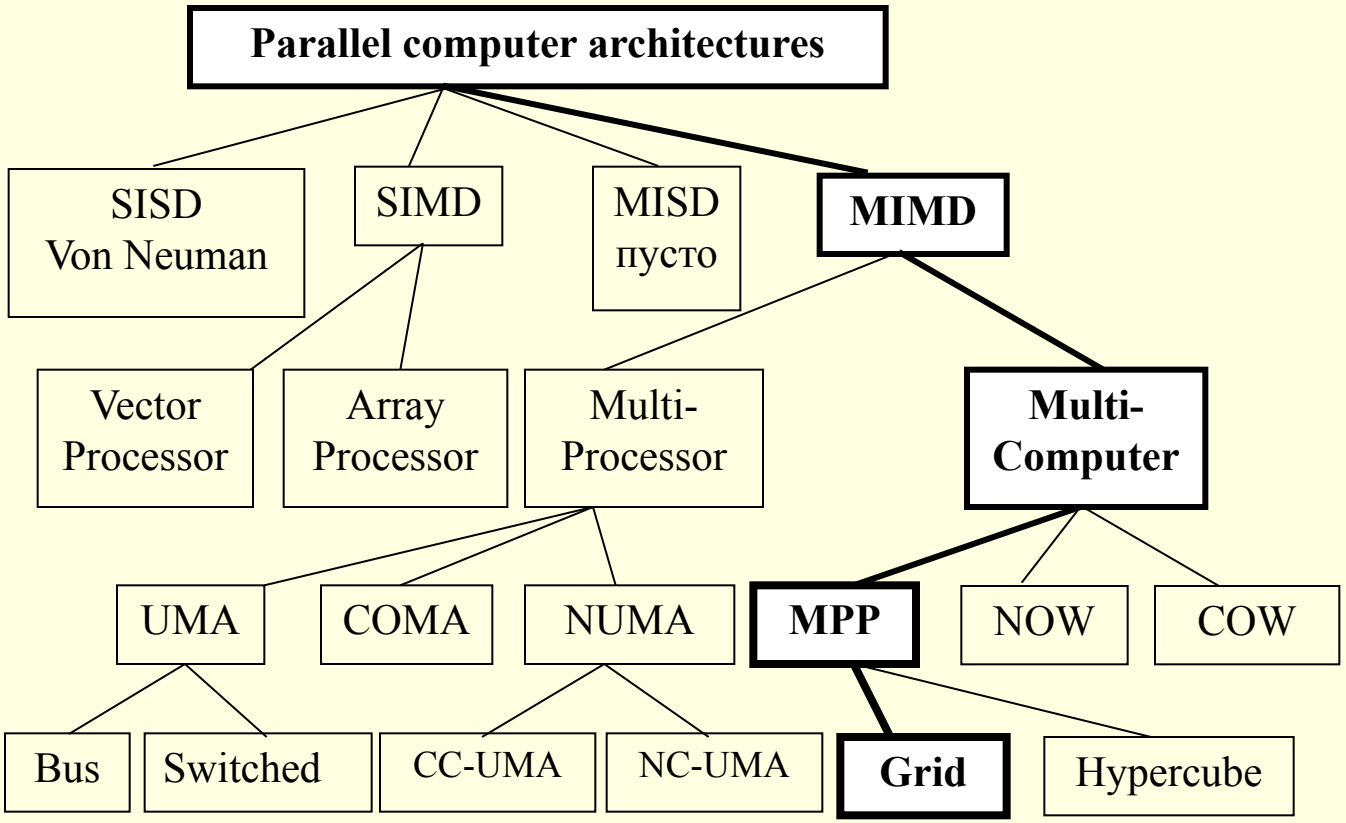


Рис. 1. Классификация параллельных компьютеров в Европе и США.

КЛАССИФИКАЦИЯ ПАРАЛЛЕЛЬНЫХ СИСТЕМ

Пусть **(О/М)К МД** означает ОКМД либо МКМД. Например, **ОКМД (О/Л)П** – класс ОКМД систем, включающий системы и с общей **О**П, и с локальной **ЛП** памятью **П**. Введём дополнительно:

1) для указания типа коммутатора К – буквы **Ц** и **Р**:

ЦК– централизованный, **РК** – распределённый коммутатор

2) типа структуры **С**: **ЖС** – жесткая или **ПС** – программируемая структура.

Получим классификацию параллельных компьютеров, содержащую 16 типов:

(О/М)К МД (О/Л)П (Ц/Р)К (Ж/П)С.

ПРЕПЯТСТВИЯ НА ПУТИ ПАРАЛЛЕЛЬНЫХ КОМПЬЮТЕРОВ

1. ВЫСОКАЯ СТОИМОСТЬ.

В соответствии с ***законом Гроша*** (**Grosch**), производительность компьютера **П** возрастает пропорционально квадрату его стоимости **Ц**:

$$П = kЦ^2$$

В результате гораздо выгоднее получить требуемое быстродействие приобретением одного производительного процессора, чем использование нескольких менее быстродействующих процессоров.

ПРЕПЯТСТВИЯ НА ПУТИ ПАРАЛЛЕЛЬНЫХ КОМПЬЮТЕРОВ

2. ПОТЕРИ ПРОИЗВОДИТЕЛЬНОСТИ ПРИ ОРГАНИЗАЦИИ ПАРАЛЛЕЛИЗМА.

Согласно ***гипотезе Минского*** (**Minsky**), ускорение, достигаемое при использовании параллельной системы, пропорционально двоичному логарифму от числа N процессоров (??):

$$S = O(\log_2 N)$$

Так на 1000 процессорах возможное ускорение оказывается равным

$$\log_2 1000 \approx 10 \text{ (??)}$$

ПРЕПЯТСТВИЯ НА ПУТИ ПАРАЛЛЕЛЬНЫХ КОМПЬЮТЕРОВ

3. ОПЕРЕЖАЮЩЕЕ СОВЕРШЕНСТВОВАНИЕ ПОСЛЕДОВАТЕЛЬНЫХ КОМПЬЮТЕРОВ.

По ***закону Мура*** **(Moore)** быстродействие последовательных процессоров возрастает практически **в 2 раза каждые 18** месяцев. История показывает, что быстродействие компьютера увеличивалось на порядок каждые 5 лет.

Необходимая производительность могла быть достигнута и на последовательных компьютерах.

ПРЕПЯТСТВИЯ НА ПУТИ ПАРАЛЛЕЛЬНЫХ КОМПЬЮТЕРОВ

4. НЕИЗБЕЖНОСТЬ ПОСЛЕДОВАТЕЛЬНЫХ ВЫЧИСЛЕНИЙ.

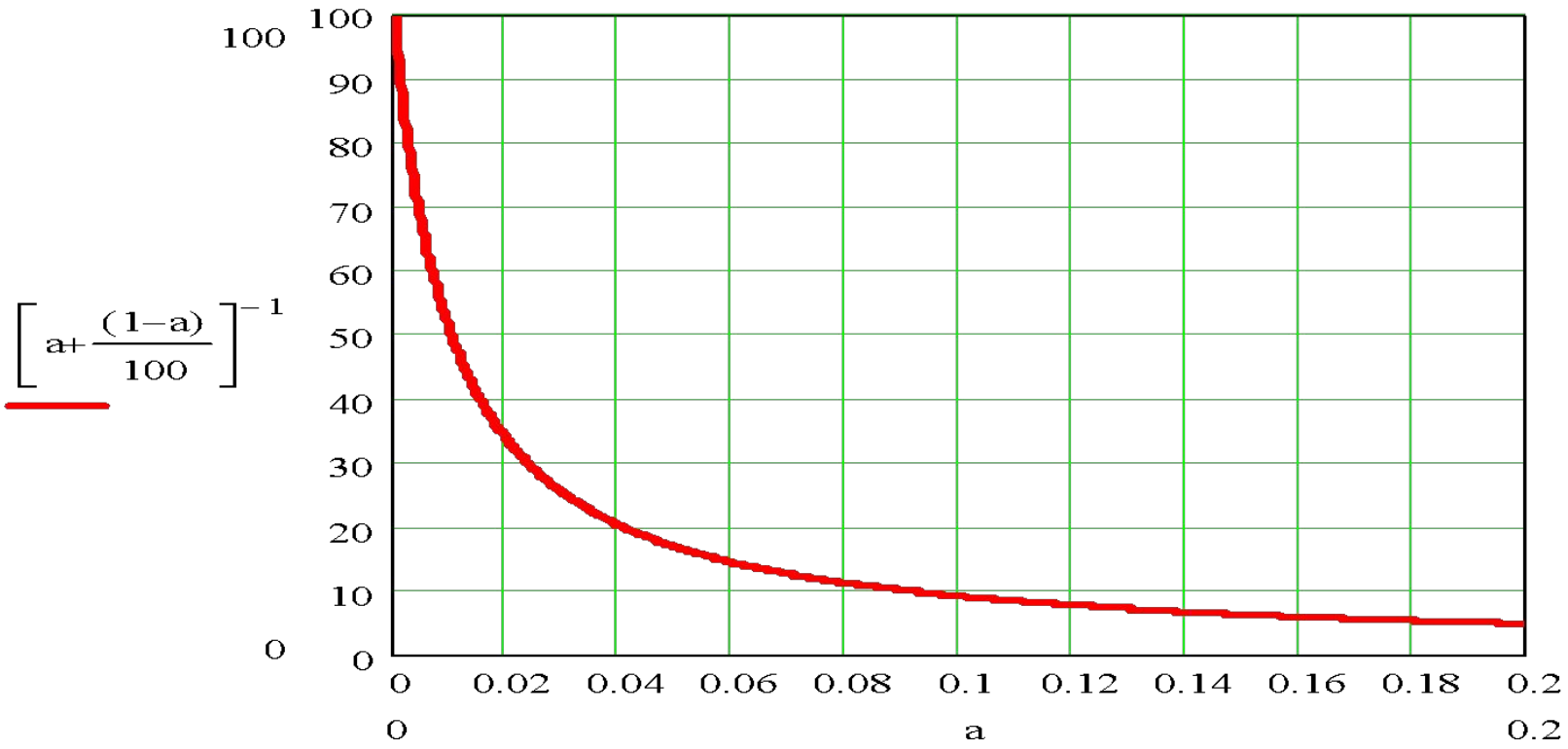
В соответствии с ***законом Амдаля*** **(Amdahl)** ускорение **S** процесса вычислений при использовании ***N*** процессоров составит всего

$$S \le [\alpha - (1- \alpha)/N]^{-1}$$

где α – доля последовательных вычислений, т.е., при наличии 10% последовательных команд, эффект использования параллелизма не может превышать 10-кратного ускорения обработки данных.

ПРЕПЯТСТВИЯ НА ПУТИ ПАРАЛЛЕЛЬНЫХ КОМПЬЮТЕРОВ

Закон Амдаля



a - доля последовательных вычислений

ПРЕПЯТСТВИЯ НА ПУТИ ПАРАЛЛЕЛЬНЫХ КОМПЬЮТЕРОВ

5. ЗАВИСИМОСТЬ ЭФФЕКТИВНОСТИ ОТ СТРУКТУРЫ СИСТЕМЫ

Параллельные системы отличаются существенным разнообразием архитектурных принципов построения, и максимальный эффект от использования параллелизма может быть получен

только при полном использовании всех особенностей аппаратуры.

Перенос параллельных алгоритмов и программ между разными типами систем становится затруднительным (если вообще возможен).

ПРЕПЯТСТВИЯ НА ПУТИ ПАРАЛЛЕЛЬНЫХ КОМПЬЮТЕРОВ

6. ОТСУТСТВИЕ ЭФФЕКТИВНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ.

Существующее программное обеспечение ориентировано, в основном, на последовательные компьютеры.

Для большого количества задач уже имеются готовые программы и все они ориентированы, главным образом, на последовательные компьютеры.

Переработка такого количества программ для параллельных систем не всегда оправдана.

ПАРАМЕТРЫ ПАРАЛЛЕЛИЗМА

Степенью параллелизма алгоритма называется число ***p*** операций, которые можно выполнять одновременно.

Пусть:

$V_p(N)$ – число операций параллельного алгоритма

k **-** число этапов параллельного алгоритма.

Средней степенью параллелизма $p_{ср}$ называется отношение

$$p_{ср} = V_p(N)/k$$

ПАРАМЕТРЫ ПАРАЛЛЕЛИЗМА

Пусть:

T_1 – время исполнения алгоритма на одном процессоре.

$T_{пар}$ – время исполнения параллельного алгоритма на p процессорах.

Ускорением параллельного алгоритма называется величина:

$$S_{пар} = T_1 / T_{пар}$$

Эффективностью параллельного алгоритма называется величина:

$$E_{пар} = S_{пар} / p,$$

где p – максимальная степень параллелизма

ВЫРОЖДЕНИЕ ПАРАЛЛЕЛИЗМА

Редукция вектора – пример вырождения параллелизма

Редукция вектора $\boldsymbol{X} = \langle x_0, x_1, \ldots, x_{n-1} \rangle$ – получение суммы его элементов

$$x_{0n} = \sum_{j=0\ldots n-1} x_j$$

получается сдваиванием. Получение частичных сумм

$$x_{0i} = \sum_{j=0\ldots i} x_j$$

получается рекурсивным сдваиванием.

ВЫРОЖДЕНИЕ ПАРАЛЛЕЛИЗМА

Для алгоритма сдваивания число операций равно

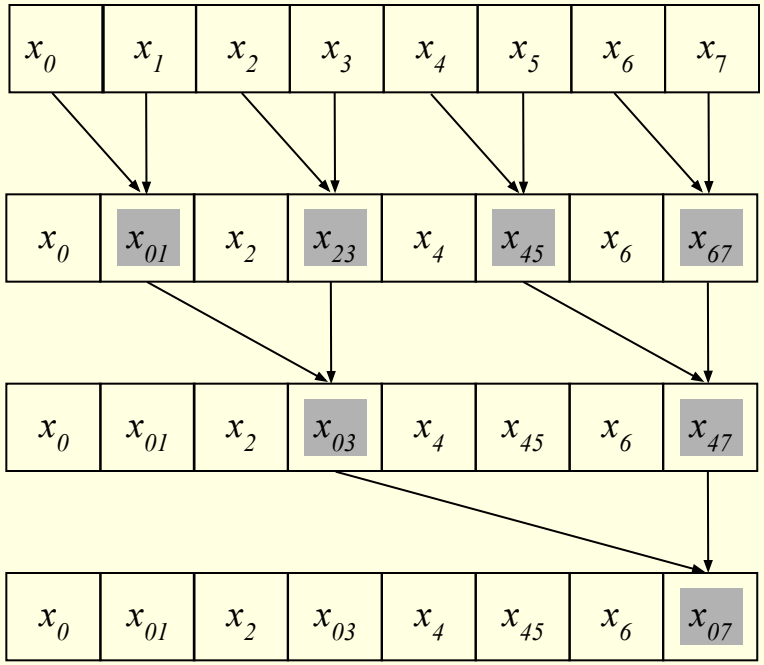
$$V_p(N) = N - 1,$$

число этапов сдваивания

$$k = \log_2 N,$$

средняя степень параллелизма равна:

$$\boldsymbol{p}_{\mathbf{ср}} = (N\text{-}1)/\log_2 N$$



Параллельное суммирование методом сдваивания.
Здесь x_{ij} – сумма от i до j.

ВЫРОЖДЕНИЕ ПАРАЛЛЕЛИЗМА

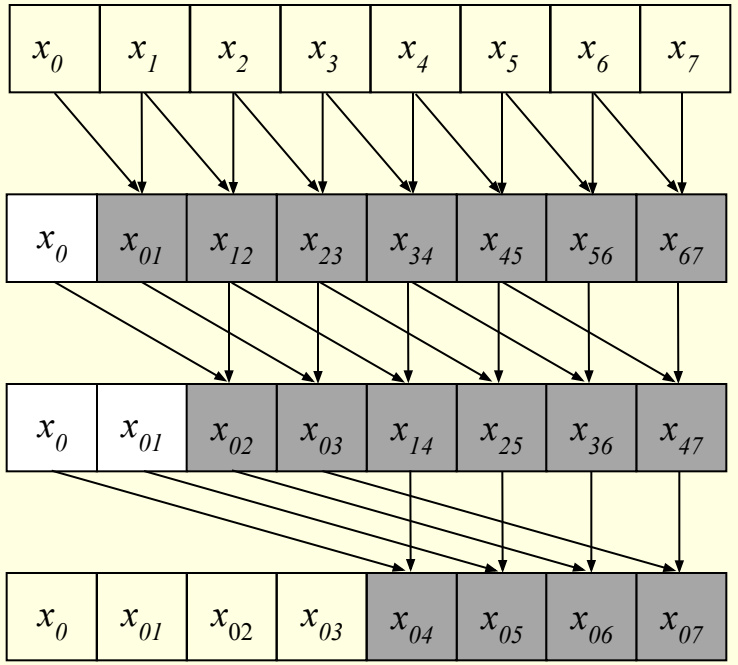
Для рекурсивного сдваивания

$$V_p(n) = N \log_2 N - (N + 1)$$

число этапов сдваивания равно $k = \log_2 N$

$$p_{cp} = N - (N + 1)/ \log_2 N$$

Здесь степень параллелизма $p = N / 2$



Параллельное вычисление частичных сумм.
Здесь x_{ij} – сумма от i до j.

ВЫРОЖДЕНИЕ ПАРАЛЛЕЛИЗМА

Пусть

t – время сложения,

β – коэффициент затрат на передачу данных

βt – время переноса данных на каждом этапе сдваивания. Игнорируя прочие издержки, имеем :

$$T_1 = (N - 1)t = (2p - 1)t$$

$$T_{пар} = [\log_2 N]\ (t + \beta t) = (1 + \log_2 p)(1 + \beta)t$$

Ускорение $S_{пар} = T_1 / T_{пар} = (N-1)/(1 + \beta)\log_2 p$

Даже при $\beta = 1$ имеем $S_{пар} = (N-1)/2\log_2 p$

т.е. **ускорение падает вдвое** по сравнению с идеальным случаем, когда нет потерь на передачу информации и $\beta = 0$.

ВЫРОЖДЕНИЕ ПАРАЛЛЕЛИЗМА как следствие его недостаточности

При больших p и при $\beta = p$

ускорение параллельного сдваивания:

$$S_{\text{пар}} = O(p/\log_2 p)$$

эффективность параллельного сдваивания:

$$E_{\text{пар}} = 2/\log_2 N \to 0 \quad \text{при } p = N/2 \to \infty$$

Итак, **недостаток параллелизма** приводит к неполной загрузке параллельной системы и к низкой эффективности.

Возникающая отсюда проблема называется *балансировкой нагрузки* или *задачей вложения* параллельных вычислений в систему. Плохое вложение может не только снижать полезную нагрузку на процессор, но и увеличивать время обмена информацией, т.е. величину β.

ФУНДАМЕНТАЛЬНЫЕ ПРЕДЕЛЫ ПРОИЗВОДИТЕЛЬНОСТИ

Наблюдение за развитием современной вычислительной техники говорит о достижении **ФУНДАМЕНТАЛЬНЫХ ПРЕДЕЛОВ роста производительности и последовательных и параллельных компьютеров:**

светового и ***теплового*.**

Старые методы вычислений оказываются менее чувствительными к этим пределам.

Это не удивительно, поскольку они разрабатывались для человека с его низкой скоростью последовательной обработки информации и высшей степени параллелизма восприятия, распознавания и обработки зрительных образов.

СВЕТОВОЙ БАРЬЕР

Световой барьер ограничивает размер синхронного компьютера соотношением:

$$f \cdot d \leq c$$

c – скорость света

$$(2 \cdot 10^{11} \text{мм/с} \leq c \leq 3 \cdot 10^{11} \text{мм/с}),$$

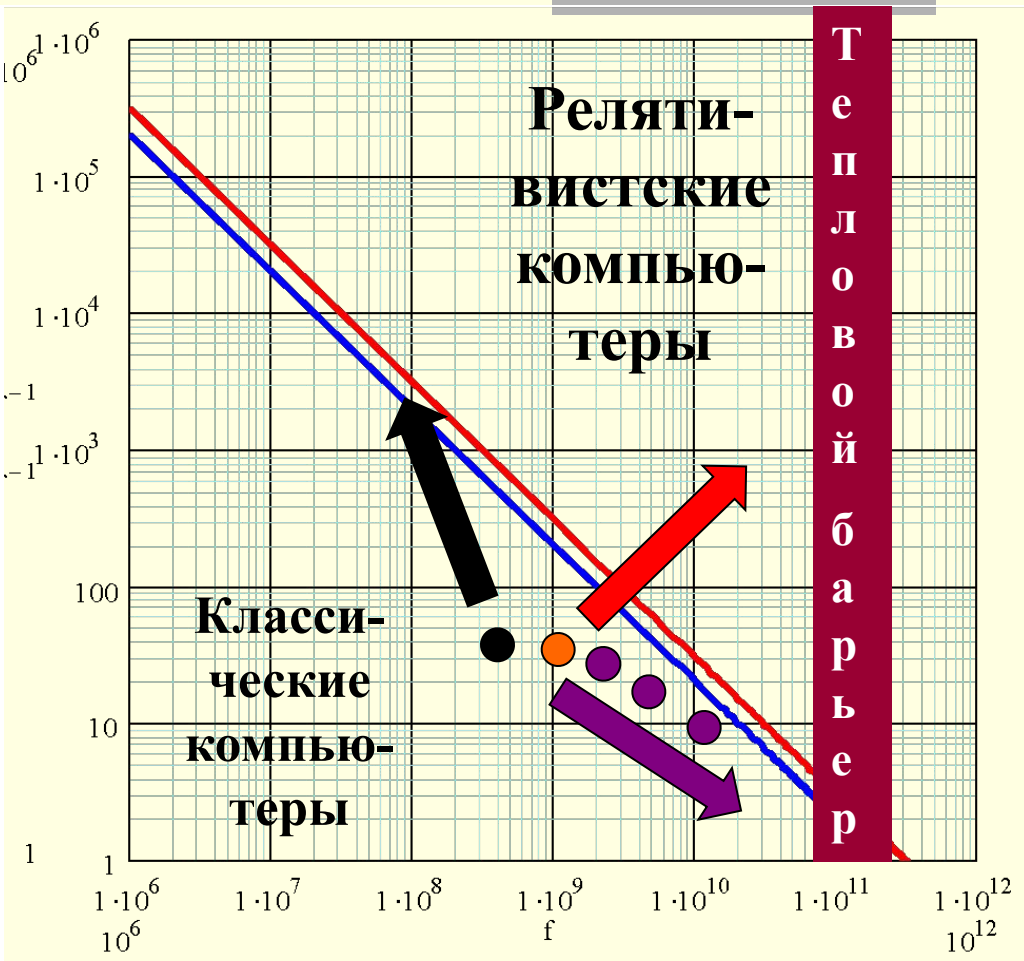
f – тактовая частота,

d – диаметр системы.

СВЕТОВОЙ И ТЕПЛОВОЙ БАРЬЕРЫ

ТРИ НАПРАВЛЕНИЯ развития параллельных вычислительных систем:

1. Классическое
2. Технологическое
3. Релятивистское



ЕСТЬ ДВЕ ТЕОРИИ ВЫЧИСЛЕНИЙ: КЛАССИЧЕСКАЯ И РЕЛЯТИВИСТСКАЯ

Классическая теория вычислений пренебрегает временем доставки данных к процессору. Это **теория медленных вычислений**, как и вся классическая механика – теория медленных движений. И уже там происходит вырождение!

Релятивистская теория вычислений основное внимание уделяет доставке данных к процессорам. Это, как и релятивистская механика, **теория быстрых вычислений**.

ВЫРОЖДЕНИЕ ПАРАЛЛЕЛИЗМА как следствие светового барьера

В общем случае для *p* процессоров имеем ускорение:

$S_p = T_1 / [T_1(\alpha_1 + k^{-1}\alpha_2 + p^{-1}\alpha_3) + t_d]$ для обменов данными перемежающихся с вычислениями.

$S_p = T_1 / max[T_1(\alpha_1 + k^{-1}\alpha_2 + p^{-1}\alpha_3), t_d]$ для обменов данными параллельных с вычислениями.

Где:

α_1 – доля операции, выполненные одним процессором,

α_2 – доля операции со средним ускорением k,

α_3 – доля операции с максимальным ускорением p,

t_d – **время доставки данных**, т.е. задержка, необходимая для перемещения и размещения данных в местах продолжения вычислений.

ВЫРОЖДЕНИЕ ПАРАЛЛЕЛИЗМА как следствие светового барьера

Закон Амдаля получается как в классической механике, если положить,

$$t_d = 0,\ \alpha_2 = 0,\ \alpha_1 = \alpha,\ \alpha_3 = 1 - \alpha$$

где α – доля последовательных операций.

Поступим наоборот. В духе релятивистской теории

не будем пренебрегать величиной t_d. Положим $\alpha_1 = \alpha_2 = 0$, $\alpha_3 = 1$, т.е. все вычисления параллельны в максимальной степени p.

где l_1 и l_2 – некоторые константы.

ВЫРОЖДЕНИЕ ПАРАЛЛЕЛИЗМА как следствие светового барьера

Реалистично полагать для одномерных систем:

$$t_d = l_1 p\, f(T_1, p)$$

а для двумерных систем:

$$t_d = l_2 p^{1/2} f(T_1, p)$$

Пусть $f(T_1, p) = \mathrm{O}(T_1)$,

т.е. пропорционально числу T_1 вычислительных операций,

Тогда даже при максимальном параллелизме имеем

$S_{\text{пар},1} = \mathrm{O}(p/(l_1 p^2)) \rightarrow 0$ при $p \rightarrow \infty$

$S_{\text{пар},2} = \mathrm{O}(p/(l_1 p^{1,5}) \rightarrow 0$ при $p \rightarrow \infty$

В ОБЩЕМ СЛУЧАЕ ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ МЕДЛЕННЕЕ ПОСЛЕДОВАТЕЛЬНЫХ!?

ПРИНЦИП ЛОКАЛЬНОСТИ

Для параллельных вычислений нужны такие функции $t_d = f(T_1, p)$, которые

1. отражают технические возможности коммутаторов,
2. ограничивают рост затрат на взаимодействие t_d.

Подходящие функции

$$f_1(T_1, p) = \mathbf{O}(T_1 \cdot p^{-1}),$$

$$\mathbf{O}(T_1 \cdot p^{-1}) \le f_2(T_1, p) \le \mathbf{O}(T_1 \cdot p^{-1/2})$$

ПРИНЦИП ЛОКАЛЬНОСТИ

Параллельные вычисления **не вырождаются** при

$$t_{d1} = l_1 p\, f(T_1, p) = l_1 p k T_1 \cdot p^{-1} = l T_1$$

$$p^{-1/2} \le t_{d2} \le l T_1$$

Требуемый вид функции $f(T_1,p)$ обеспечивается **параллельными операциями обмена информаций**.

Это возможно в двух случаях.

ПРИНЦИП ЛОКАЛЬНОСТИ

1. **В системе с общей памятью** все запросы от ***p*** процессоров идут к различным банкам памяти, а коммутатор между процессорами и памятью допускает все ***p*** соединений одновременно.
2. **В системе с локальной памятью** обмены данными происходят так, что все ***p*** (или $p^{1/2}$) процессоров передают, и все ***p*** (или $p^{1/2}$) процессоров принимают информацию **локально (соседям) и одновременно.**

Второй подход назван нами **мелкозернистое локально-параллельное программирование (МЛП-программирование).**

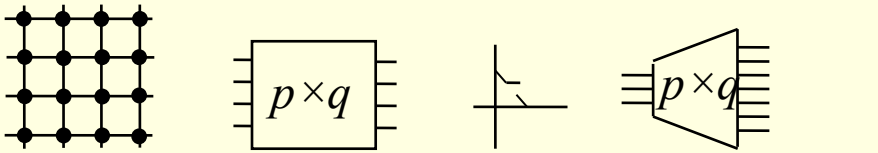
КЛАССИЧЕСКИЕ КОММУТАТОРЫ

Коммутатор – схема обеспечивающая соединение p входов и q выходов.

Рис. 1 – разделённый коммутатор кроссбар

Рис. 2 – неразделённый коммутатор кроссбар

Рис. 3 – разделённый коммутатор Баттерфляй (Δ - сеть)



(а) (б) (в) (г)

Рис.1. Перекрёстные коммутаторы.

(а) (б)

Рис.2. Треугольный коммутатор.

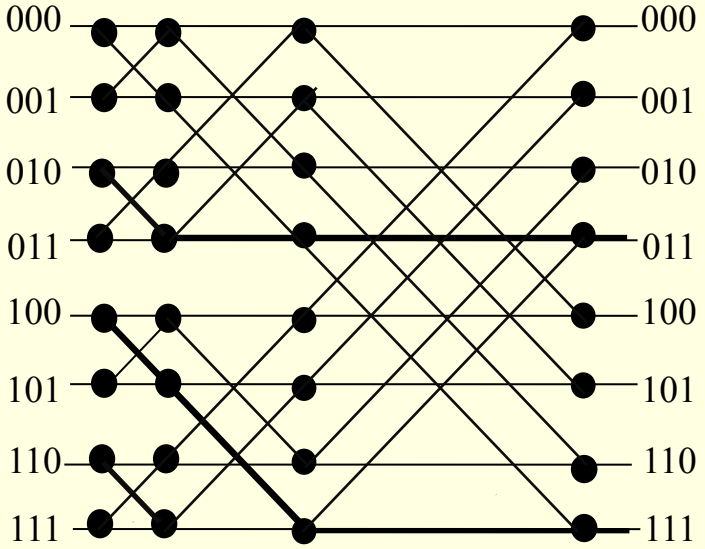


Рис.3. Баттерфляй

СЛОЖНОСТЬ и ЗАДЕРЖКА КЛАССИЧЕСКИХ КОММУТАТОРОВ

ТЕОРЕМА ШЕННОНА.

Сложность (число узлов коммутации) неблокирующего коммутатора на p входов и p выходов

$$s \geq p \log_2 p$$

Сложность неразделённого коммутатора на p входов-выходов

$$s \geq (p/2)\log_2(p/2)$$

Задержка коммутатора $\tau_{min} \geq O(\log_2 p)$

Сложность коммутатора кросс-бар $O(p^2)$

Задержка коммутатора кросс-бар $\tau \geq O(2p)$

ВЫВОД: КЛАССИЧЕСКИЕ КОММУТАТОРЫ НЕ ОБЕСПЕЧИВАЮТ РЕЛЯТИВИСТСКИЕ ВЫЧИСЛЕНИЯ

ЛОКАЛЬНЫЕ КОММУТАТОРЫ

ЛОКАЛЬНЫЕ КОММУТАТОРЫ СОЕДИНЯЮТ ТОЛЬКО СОСЕДЕЙ И ТОЛЬКО НА ОГРАНИЧЕННОМ РАССТОЯНИИ

СЛОЖНОСТЬ ЛОКАЛЬНОГО КОММУТАТОРА ПРОПОРЦИОНАЛЬНА РАЗМЕРУ СИСТЕМЫ

$$s_{\text{лок}} \geq p_{\text{соседей}} \log_2 p_{\text{соседей}} = \text{const},$$

$$S = p \times s_{\text{лок}} = p \times \text{const}$$

ЗАДЕРЖКА ЛОКАЛЬНОГО КОММУТАТОРА НЕ ЗАВИСИТ ОТ РАЗМЕРА СИСТЕМЫ

$$\tau = \log_2 p_{\text{соседей}} = \text{const},$$

ВЫВОД:

ТОЛЬКО ЛОКАЛЬНЫЕ КОММУТАТОРЫ ОБЕСПЕЧИВАЮТ РЕЛЯТИВИСТСКИЕ ВЫЧИСЛЕНИЯ

ЛОКАЛЬНЫЕ ОДНОРОДНЫЕ СТРУКТУРЫ

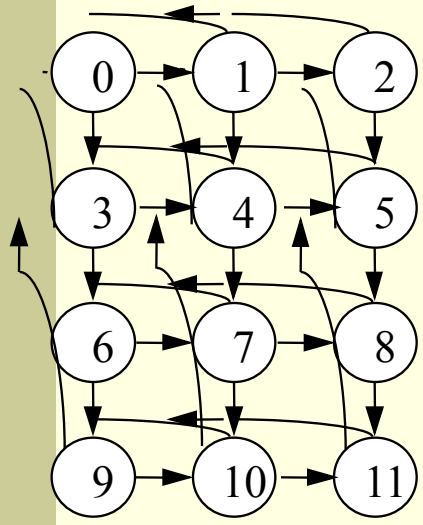
Структура вычислительной системы – граф связей между её элементами. Структура ***однородна***, если она обладает транзитивной группой автоморфизмов, сохраняющей отметки рёбер.

Теорема Вагнера. Структура однородна тогда и только тогда, когда она есть групп-граф с точностью до направлений дуг.

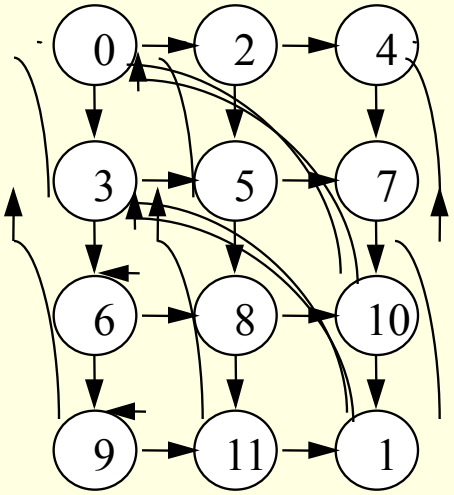
Структура ***локальна*****,** если она вкладывается в физическое пространство так, что сохраняется ограниченное расстояние между соседями.

Теорема Воробьёва. Однородны и локальны двумерные структуры, порождаемые квадратной решёткой и расположенные на торе, и только они.

ЛОКАЛЬНЫЕ ОДНОРОДНЫЕ СТРУКТУРЫ

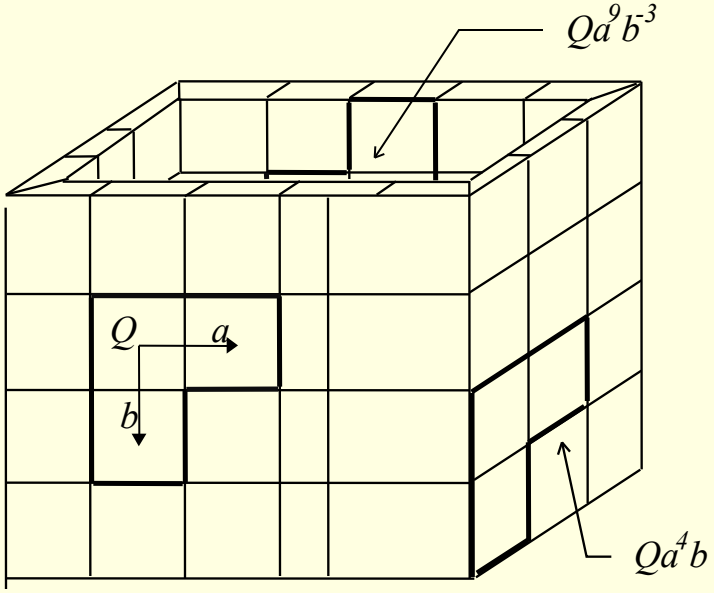


(а)



(б)

E_2 и D_2 –графы размеров 3×4 и {12,2,3}



КАИС-структура E_2: {16×8} на торе

Примеры локальных однородных структур и тороидальный конструктив системы

ПРОБЛЕМА РЕЛЯТИВИСТСКОГО КОМПЬЮТЕРА

Релятивистский компьютер технически возможен поскольку:

1. Существуют локальные однородные структуры связей с линейной оценкой сложности и постоянной задержкой передачи между соседями.

2. Очевидно существование локально-параллельных протоколов взаимодействий и синхронизации событий.

Для релятивистских вычислений необходимы:

1. Локально-параллелльные вычислительные алгоритмы и

2. МЛП-программирование.

РЕЛЯТИВИСТСКАЯ Однородная Вычислительная Система

Архитектура ОВС на **Н**еразрезных **П**роцессорных **М**атрицах СБИС

ПК - Мониторная подсистема на основе персонального компьютера

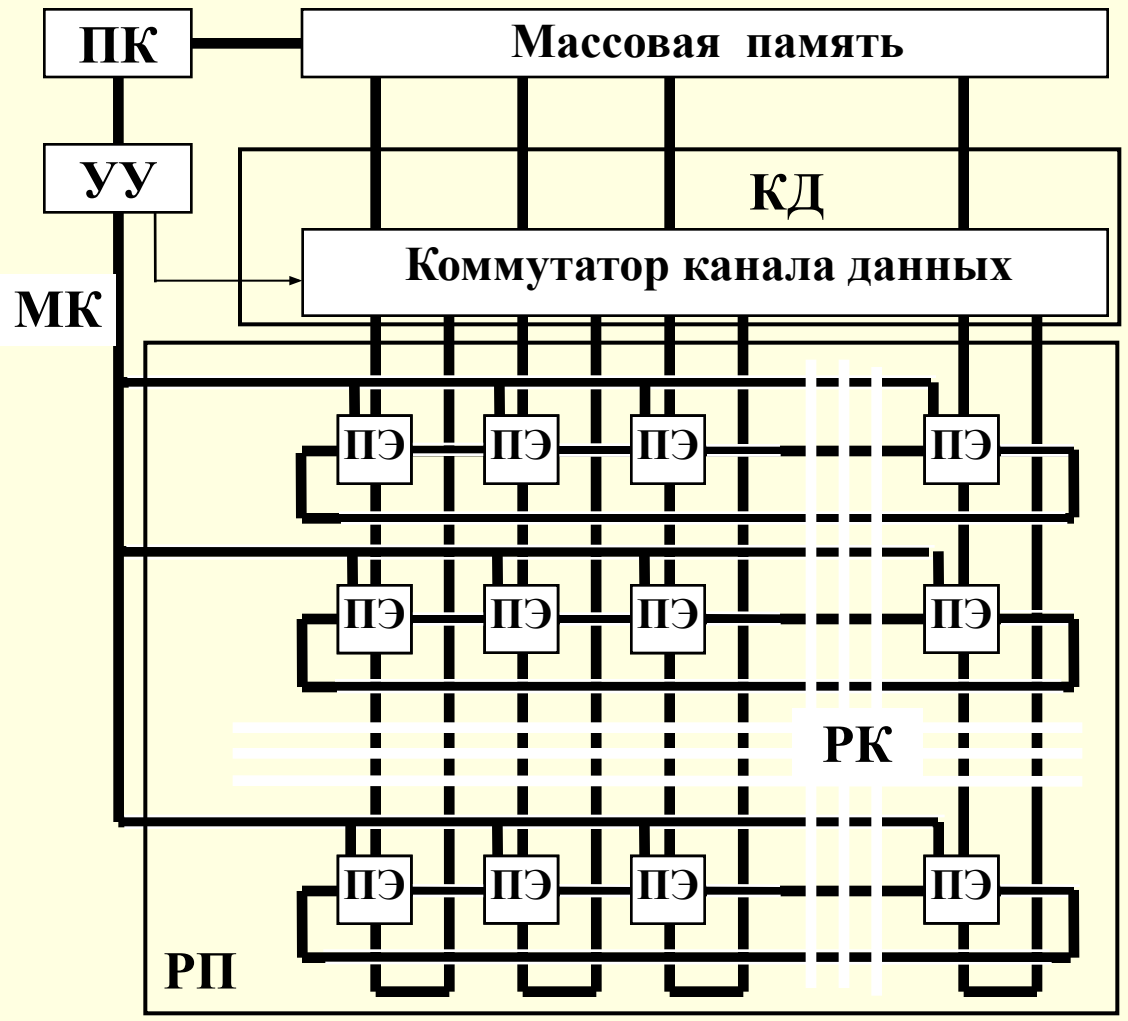
УУ – Устройство управления решающим полем

МК – Магистральный канал (магистральная шина)

РК – Регулярный канал (**КАИС** - структура)

КД – Канал данных

РП – Решающее поле на неразрезных процессорных матрицах **НПМ СБИС** (**КАИС- структура**)



РЕШАЮЩЕЕ ПОЛЕ – НЕРАЗРЕЗНАЯ ПРОЦЕССОРНАЯ МАТРИЦА СБИС

- Современная НАНОТЕХНОЛОГИЯ позволяет вплотную подойти к изготовлению неразрезных процессорных матриц (НПМ) СБИС, содержащих сотни процессорных элементов с локальной памятью и программируемой КАИС-структурой коммутатора
- НПМ СБИС наилучшим образом соответствуют идеологии ОВС и потребностям дальнейшего развития компактных массово-параллельных (мозгоподобных) вычислительных устройств.

КЛАССИЧЕСКАЯ ПАРАДИГМА ОТКАЗОУСТОЙЧИВОСТИ

1. Контролируемое и диагностируемое устройство присутствует в системе в единственном экземпляре.
2. Это устройство имеет высокую цену.
3. Элементы устройства можно заменять по мере их отказов. Минимальный заменяемый блок – типовой элемент замены (ТЭЗ) можно контролировать, диагностировать и ремонтировать на специальном стенде вплоть до замены отдельных электронных компонент.
4. Связи между ТЭЗ-ами дёшевы и либо абсолютно надёжны, либо их отказы приписываются элементам. И, вообще, проверка связей - вопрос отдельный.
5. Имеются абсолютно надёжные элементы, например, смесители Дж. фон Неймана

Важнейшие особенности неразрезной технологии СБИС

1. Невозможность замены отказавших элементов;
2. Отказы не только процессорных элементов, но и связей между ними;
3. Однородность и близкодействие структуры связей;
4. Стремление к экономии коммутационного оборудования

ПРОБЛЕМА НПМ СБИС - ОТКАЗОУСТОЙЧИВОСТЬ

Указанные особенности СБИС создают принципиальные сложности и фундаментальные ограничения при производстве и обеспечения отказоустойчивости НПМ СБИС.

ПРОБЛЕМА НПМ СБИС - ОТКАЗОУСТОЙЧИВОСТЬ

1. Отказавшие ПЭ следует обходить, используя резервные элементы
2. Отказавшие связи следует заменять резервными связями.
3. Однородность и локальность структуры следует сохранить.
4. Резерв ПЭ и Связей следует минимизировать

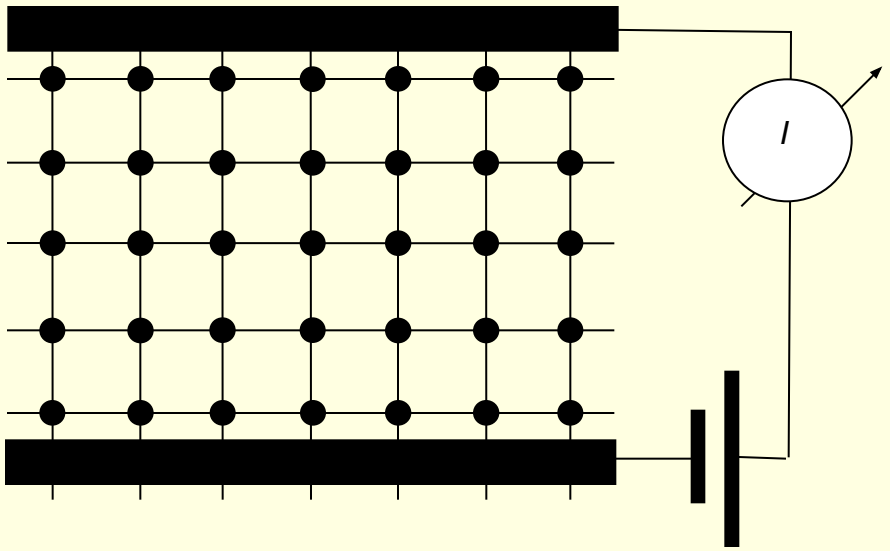
ПРОСАЧИВАНИЕ – ФУНДАМЕНТАЛЬНЫЙ ПРЕДЕЛ ОТКАЗОУСТОЙЧИВОСТИ НПМ

Задача Хаммерсли (1963г.)

Будем удалять узлы и связи случайным образом, до тех пор пока ток не прекратится.

Каковы вероятности

исправных узлов p_H и связей r_H в момент прекращения тока?



Эксперимент с просачиванием электрического тока в квадратной решётке К

КРИТЕРИЙ ПРОСАЧИВАНИЯ (ПЕРКОЛЯЦИИ)

Пусть на множестве локальных однородных структур одного типа, вложенных в R^2 с абсолютно надёжными связями,

p – вероятность исправности ПЭ,

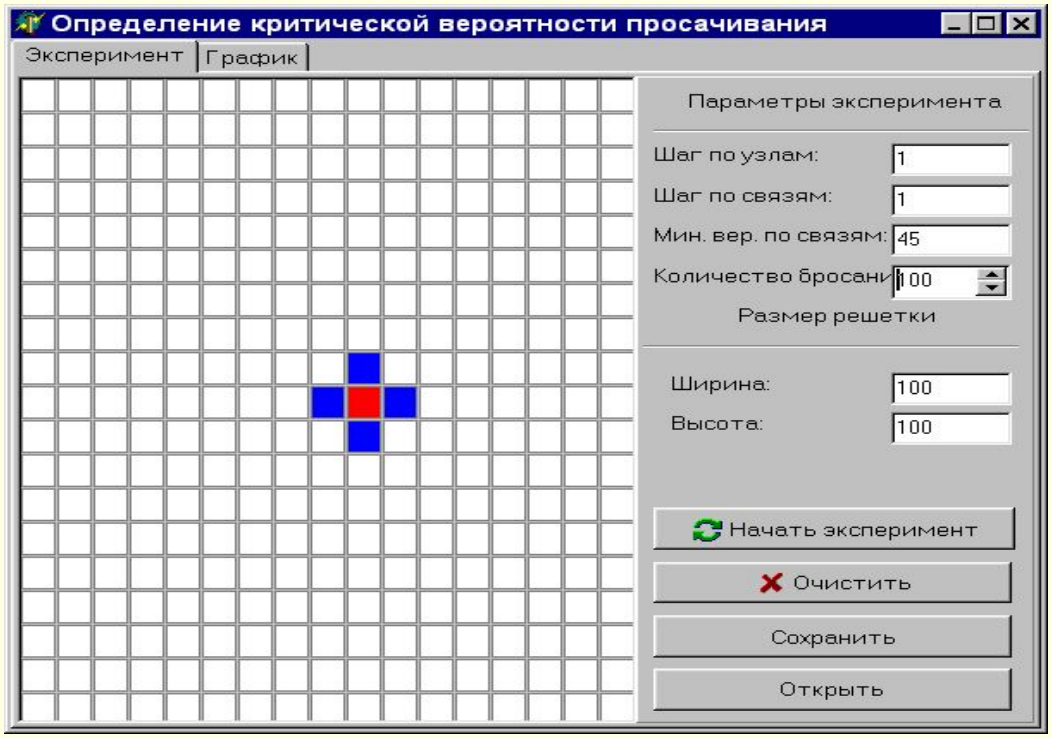
p_H – критическая вероятность просачивания по узлам

Утверждение 1. С ростом избыточности НПМ СБИС становится сколь угодно надёжной,

если $p > p_н$, и сколь угодно ненадёжной,

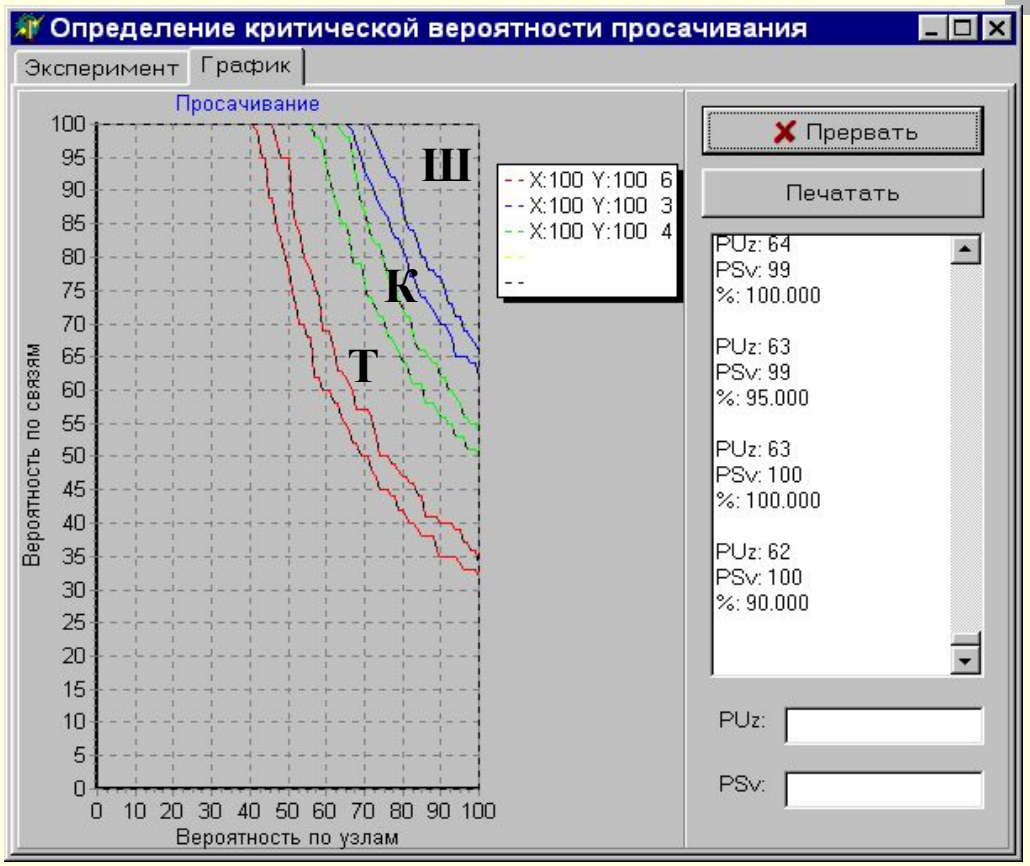
если $p < p_н$.

МОДЕЛЬ ПРОСАЧИВАНИЯ (ПЕРКОЛЯЦИИ)



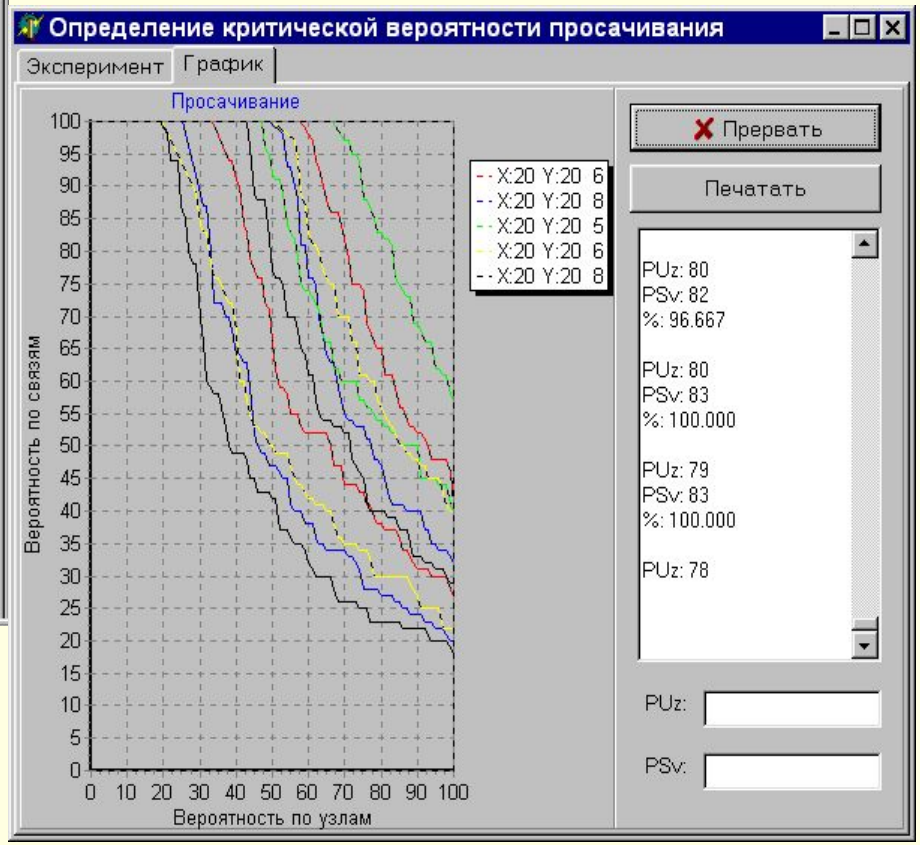
Шаблон соседства для решетки К

МОДЕЛЬ ПРОСАЧИВАНИЯ (ПЕРКОЛЯЦИИ)



Области просачивания для решеток Т, К и Ш.

МОДЕЛЬ ПРОСАЧИВАНИЯ (ПЕРКОЛЯЦИИ)



Шаблоны соседства и области просачивания для некоторых решеток

ПОРОГИ ПРОСАЧИВАНИЯ

Таблица 1.

Критические вероятности просачивания для различных решёток				
Размерность	**Решётка**	**Степень узла**	**По узлам**	**По связям**
2	**Ш**	**3**	**0,70**	**0,65**
2	**К**	**4**	**0,59**	**0,50**
2	**Т**	**6**	**0,50**	**0,35**
2	**К***	**8**	**0,41**	**0,25**
3	**алмаз**	**4**	**0,43**	**0,39**
3	**куб**	**6**	**0,31**	**0,25**
3	**Центрированный куб**	**8**	**0,25**	**0,18**
3	**Гранецентрированный куб**	**10**	**0,20**	**0,12**

ДИНАМИЧЕСКАЯ МОДЕЛЬ ОТКАЗОУСТОЙЧИВОСТИ НПМ

Пусть время безотказного функционирования ЭМ распределено по экспоненциальному закону с параметром α при условии $p \geq p_H$
Среднее время наработки на отказ для ЭМ равно $\tau = \alpha^{-1}$.
Величина α − интенсивность отказов ЭМ, τ − время жизни.
Отказавшая ЭМ обнаруживается и восстанавливается с интенсивностью β за среднее время восстановления β^{-1}.

$dp(t) = [\beta(1 - p(t)) - \alpha\, p(t)]\, dt$ - уравнение процесса гибели − восстановления в бесконечном ансамбле ЭМ.

Начальные условия:

$p_0 = p(0)$

$p_0 = 1$ если бракованные ЭМ можно заменить

Решение уравнения при указанных начальных условиях имеет вид

$$p = p_\infty + [p_0 - p_\infty]\, e^{-t(\alpha + \beta)},$$

где − финальная вероятность исправного состояния ЭМ.

ОТКАЗОУСТОЙЧИВОСТЬ НПМ БЕЗ ВОССТАНОВЛЕНИЯ

Утверждение 2. Время существования бесконечного исправного кластера в локальной однородной структуре без восстановления равно

$$t = \tau \, ln(p_0 / p_H) \leq \tau \, ln(p^{-1})$$

где $\tau = \alpha^{-1}$ – время жизни одного элемента.

Избыточность не увеличивает времени жизни локальной однородной структуры.

РЕКОНФИГУРАЦИЯ НПМ по САМИ и СТЕФАНЕЛЛИ

Сами и Стефанелли предложили несколько алгоритмов реконфигурации НПМ для обхода неисправных узлов решётки в следующих предположениях:

1. Края НПМ (строка или столбец) резервируются.
2. Связи абсолютно надёжны.
3. Неисправные узлы обнаруживаются при тестировании извне.
4. Каждый ПЭ имеет коммутационное окружение, вычисляющее реконфигурацию структуры связей по сигналам неисправностей ПЭ.

НПМ с ОГРАНИЧЕННЫМ ВОССТАНОВЛЕНИЕМ

Пусть N – число ЭМ; m – число восстанавливающих органов, работающих параллельно; α и β – интенсивности потока отказов и восстановлений.

При $m > N(1-p)$ система ведёт себя как самовосстанавливающаяся.

При $m < N(1-p)$ имеет место ограниченное восстановление.

Условие равенства потоков отказов и восстановлений :

$$N \alpha p = m \beta$$

При полной загрузке ограниченной восстанавливающей системы математическое ожидание числа исправных элементов равно $m\beta\alpha^{-1}$.

Потребуем, чтобы это число превышало N_0 и, кроме того, выполнялось

$$p > p_H .$$

Утверждение 3. НПМ с ограниченным восстановлением работоспособна если выполняется условие

$$N_0 < m\,\beta\,\alpha^{-1} \le N \le m\,\beta\,(\alpha\, p_H)^{-1}$$

При $N > m\beta(\alpha p_H)^{-1}$, стационарное состояние НПМ есть состояние отказа.

КОММУТАЦИОННОЕ ОКРУЖЕНИЕ ПЭ по САМИ И СТЕФАНЕЛЛИ

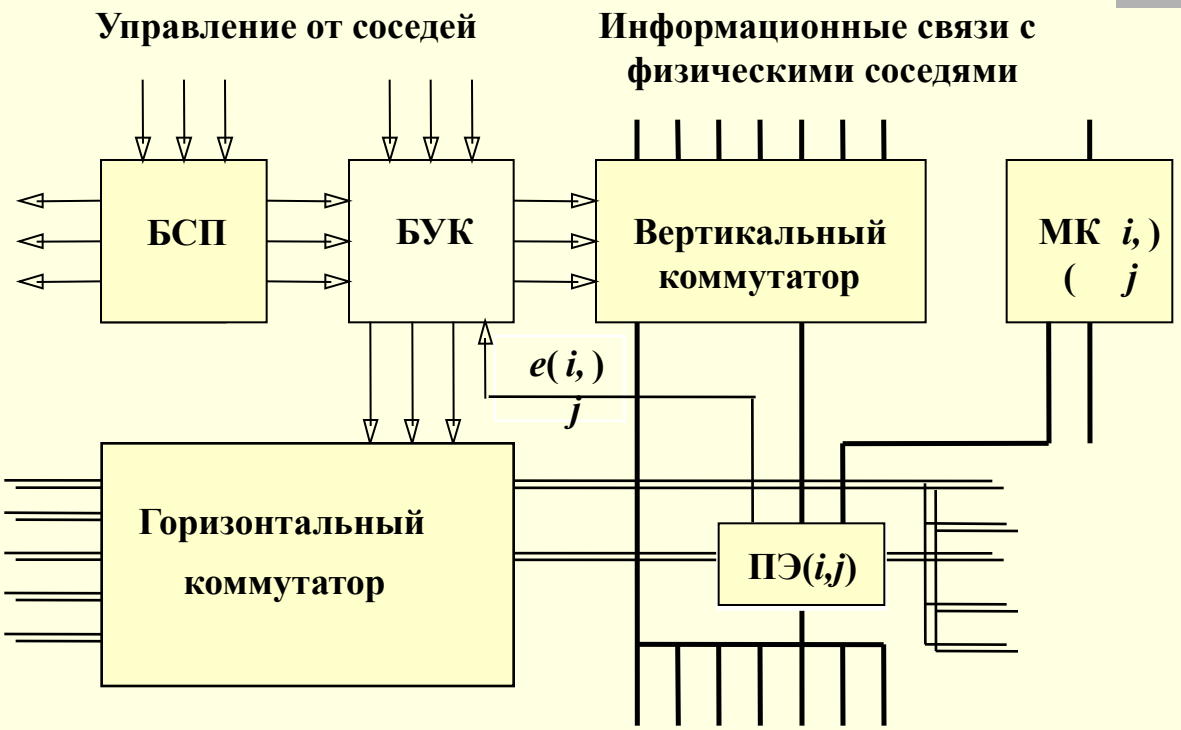


Рис. 4.3.4. Коммутационное окружение процессорного элемента.

БСП - блок формирования сигналов перестройки матрицы;

БУК - блок управления коммутациями;

МК - магистральный канал.

РЕЗЕРВНЫЕ СВЯЗИ ПЭ по Сами и Стефанелли

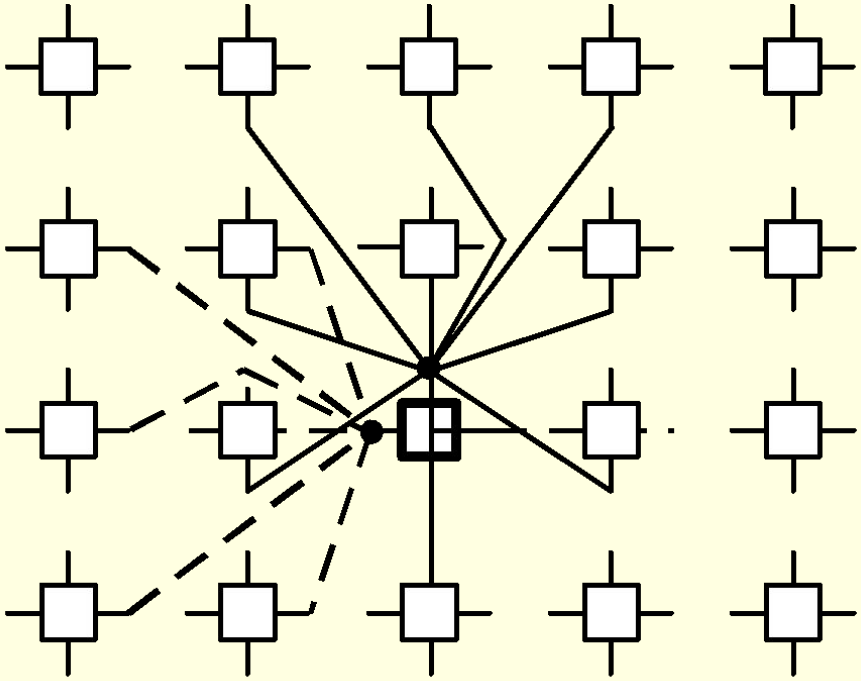


Рис. 4.3.3. Информационные связи ПЭ(i,j) в матрице с программируемым резервом (используются в зависимости от алгоритма):

- - - - – горизонтальные линии связи

——— – вертикальные линии связи

РЕКОНФИГУРАЦИЯ НПМ по В.А.Воробьёву – Н.В.Лаходыновой

В.А. Воробьёв и Н.В. Лаходынова модифицировали те же алгоритмы реконфигурации НПМ для обхода неисправных узлов решётки в иных предположениях:

1. Резервируются любые строки и столбцы в любых количествах.
2. Связи абсолютно надёжны.
3. Неисправные узлы самодиагностируются тестовой программой или аппаратно.
4. Каждый ПЭ имеет коммутационное окружение и программу вычисляющую реконфигурацию структуры связей по сигналам неисправностей от ПЭ.

КОММУТАЦИОННОЕ ОКРУЖЕНИЕ по В.А.Воробьёву– Н.В.Лаходыновой

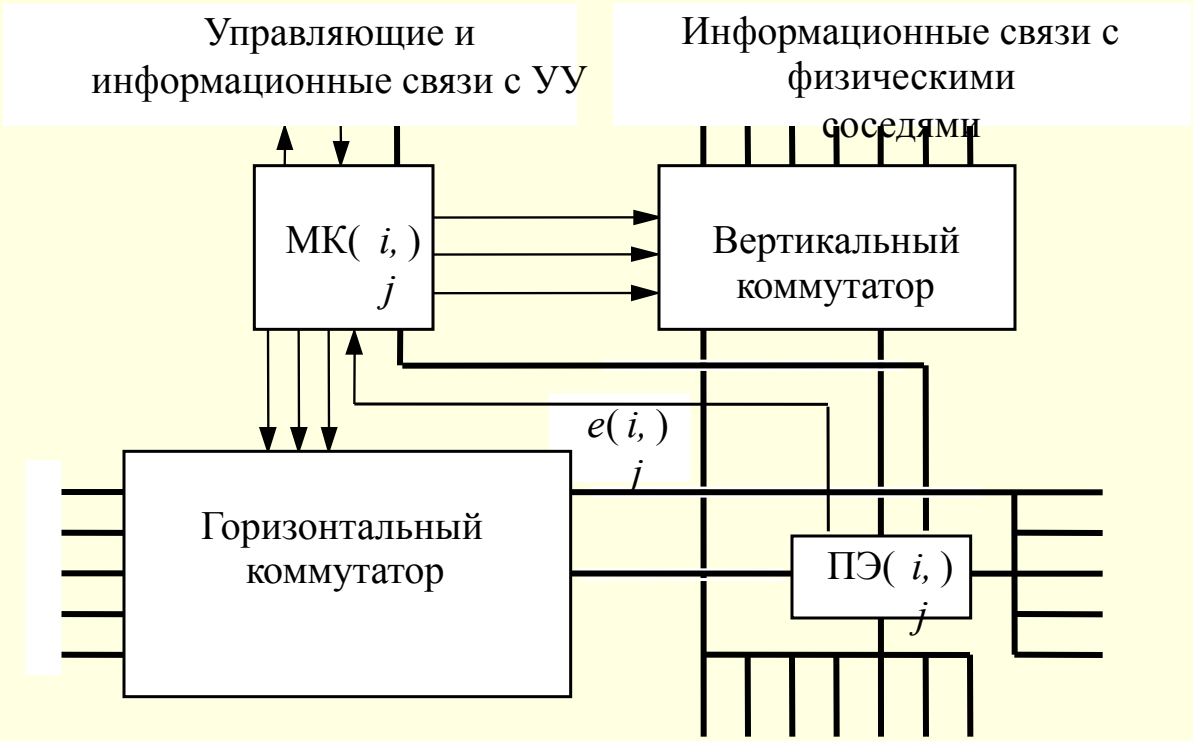
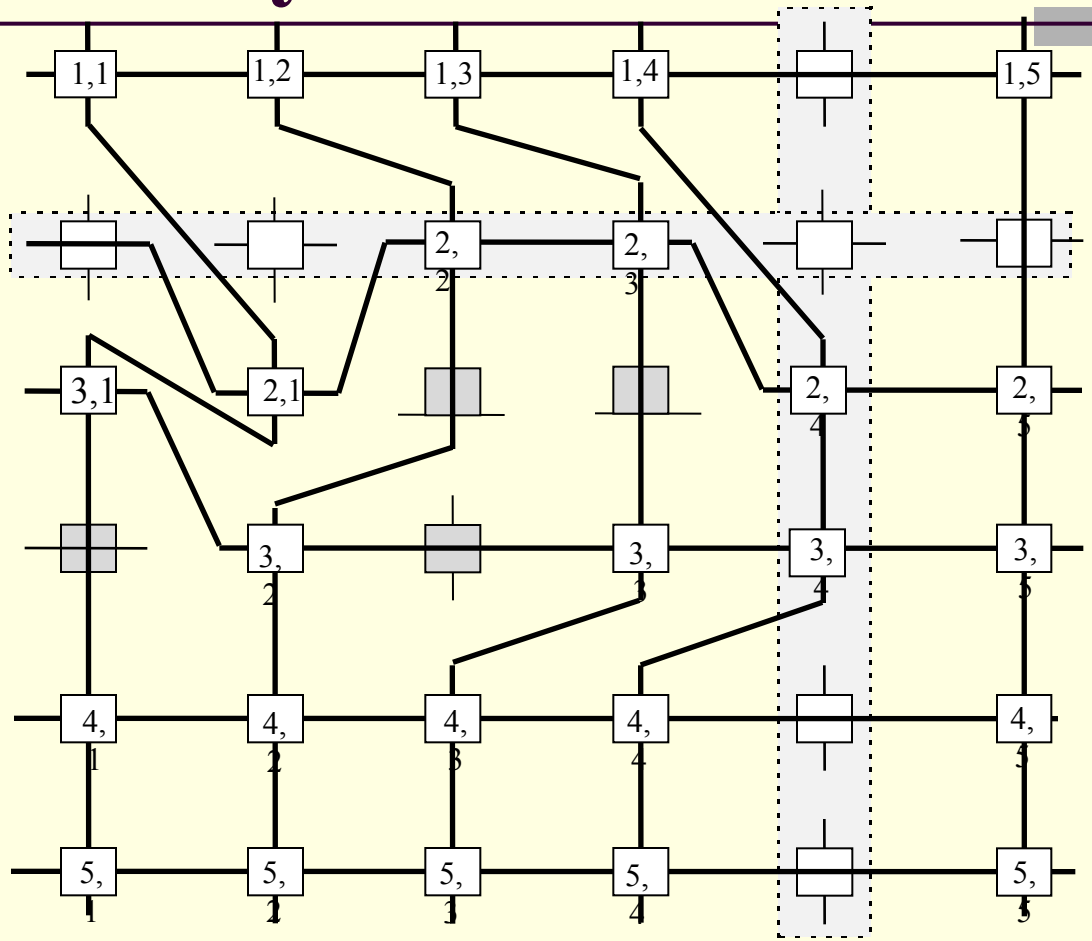


Рис. 4.4.11. Коммутационное окружение ПЭ(i,j) в матрице с программной реализацией алгоритмов реконфигурации.

МК - магистральный канал

РЕКОНФИГУРАЦИЯ НПМ по В.А. Воробьёву – Н.В. Лаходыновой



Перестройка матрицы по алгоритму свободного захвата в ситуации фатального отказа алгоритма ограниченного захвата

ВИЗАНТИЙСКАЯ ПАРАДИГМА ОТКАЗОУСТОЙЧИВОСТИ НПМ

1. Элементом замены является элементарная машина (ЭМ) – устройство, которое обычно само по себе снабжается средствами самодиагностики. Для неразрезной пластины СБИС – замена вообще невозможна, поэтому выбор минимального диагностируемого элемента является предметом особого рассмотрения.
2. Структура системы однородна и локальна - связаны только соседи.
3. В силу одинаковости ЭМ каждый элемент окрестности ЭМ является для неё эталоном и таких эталонов достаточно много (8 ÷16 и более).
4. Наличие большого числа эталонных модулей позволяет упростить ЭМ за счёт удаления средств индивидуальной самодиагностики, но в то же время требует системной самодиагностики. Основная идея состоит в том, что достаточно сложные модули могут диагностировать друг друга.
5. Алгоритм самодиагностики пытается восстановить ***образ неисправности***, то есть выделить неисправные модули. Эта задача известна как **проблема византийских генералов**, пытающихся путём взаимопроверок и обменов результатами найти генералов-предателей, подсчитать число верных генералов и всех их включить в систему.
6. Для византийского согласия характерны все исходные предположения о диагностируемой системе, кроме единственности устройства.

КОНСЕНСУС-ПАРАДИГМА ОТКАЗОУСТОЙЧИВОСТИ НПМ

1. ПЭ много и они связаны однородной, локальной сетью связи.
2. Число связей у каждого ПЭ ограничено.
3. Связи можно программно включать, выключать и коммутировать.
4. Отдельные ПЭ и связи дёшевы относительно цены исправной НПМ.
5. Абсолютно надёжных ПЭ и связей нет.
6. Заменить отказавшие ПЭ и связи невозможно.
7. Византийское согласие всех исправных ПЭ ненужно и нереально.
8. Аппаратура самодиагностики не нужна. Соседние ПЭ могут служить эталонами друг для друга, их число достаточно, чтобы изолировать неисправный ПЭ надёжнее чем самодиагностика.
9. Резервные ПЭ и связи не выделяются.
10. Отказоустойчивость достигается за счёт программирования структуры и избыточного числа ПЭ и связей между ними. Задача состоит в том, чтобы вложить требуемый граф в избыточный граф с отказами элементов и связей.

ВОЛНОВОЙ АЛГОРИТМ ПОИСКА КОНСЕНСУСА

Каждый ПЭ тестируется и результаты тестирования сравниваются с результатами тестирования соседей. По результатам тестирования в каждом ПЭ создается список соседей, с которыми ПЭ согласен. Дуга отмечается флагом несогласия, если хотя бы один инцидентный ей ПЭ “не согласен” с соседом.

Прямой ход. Начиная с верхнего левого угла матрицы ПЭ передают свои логические номера согласным соседям. Для того, чтобы логический номер (i,j) стал возможен необходимо найти среди предшественников некоторое множество уже вычисленных номеров.

Для квадратной решётки это пара $\{(i1, j1),(i2, j2)\}$, удовлетворяющая условиям

(1) $[\,|\, i1 - i2\,|\, = 1] \wedge [\,|\, j1 - j2\,|\, = 1]$

(2) $[(i1 < i2) \rightarrow (j1 > j2)] \vee [(i1 > i2\ ) \rightarrow (j1 < j2)]$

Для треугольной решётки это тройка $\{(i0, j0),(i1, j1),(i2, j2)\}$, удовлетворяющая ещё одному условию:

(3) $(i0, j0) = (min\ (i1,i2\ ),\ min\ (j1,j2))$

Для решётки К* это уже четвёрка $\{(i0, j0),\ (i1, j1),\ (i2\ , j2),\ (i4, j4)\}$, удовлетворяющая ещё одному дополнительному условию:

(4) $(i4, j4) = (max\ (i1,\ i2) +1,\ min\ (j1, j2))$

Наконец, для решётки Ш необходимая конфигурация передающих, согласных соседей есть “прореженная” конфигурация квадратной решётки. Например, если чётности строк и столбцов совпадают, требуются условия (1) и (2), если не совпадают, достаточно одного передающего и согласного соседа слева.

Логический номер (i,j) во всех случаях вычисляется по формуле

$(\ i, j\ ) = (\ max\ (i1,\ i2),\ max\ (j1, j2)\ )$

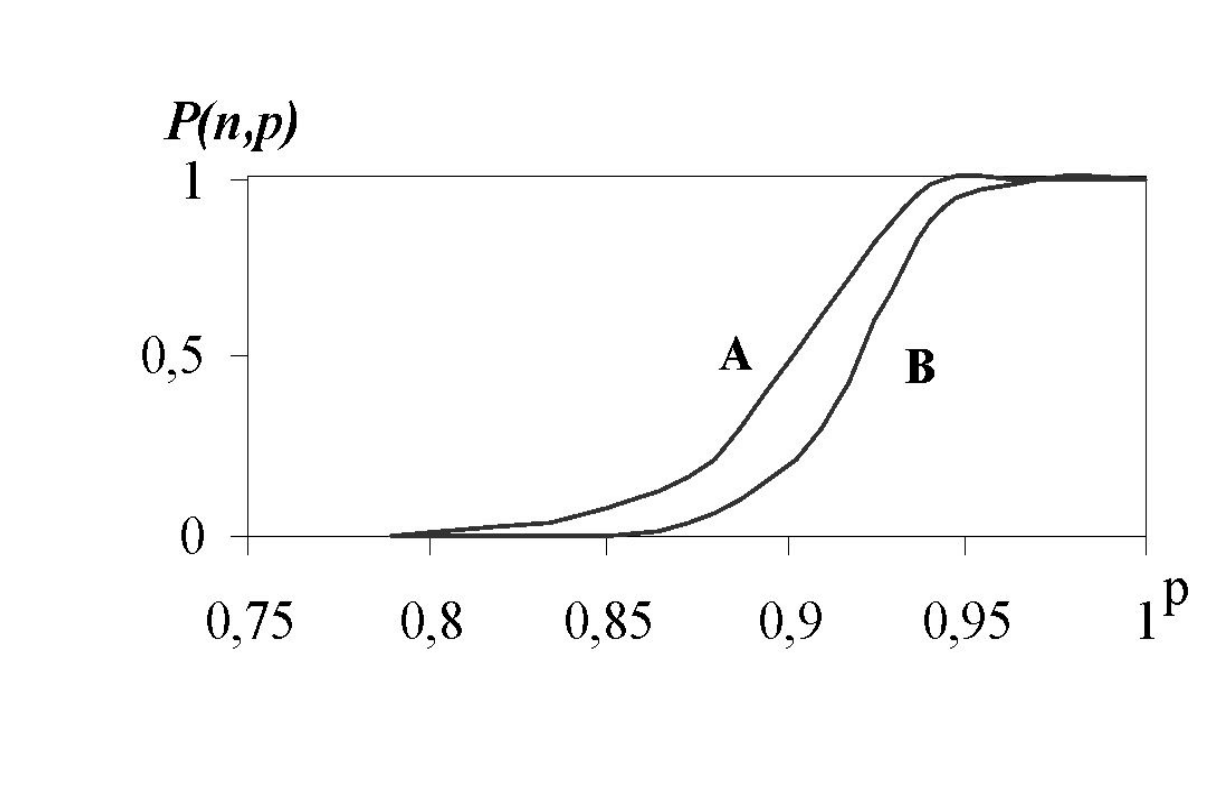
Обратный ход подобен прямому ходу, только передача информации идёт в обратном направлении, а логический номер (i'', j'') вычисляется по правилу

$(i'', j'') = (min(i,\ i'),\ min(j, j'))$.

Вычисленный номер (i'', j'') считается допустимым в ПЭ, если ранее он уже был получен в процессе прямого хода.

Таким образом, после обратного хода в списках допустимых номеров ПЭ остаётся только пересечение множеств номеров прямого и обратного хода. Остальные номера не имеют отношения к искомой решётке и уничтожаются. ПЭ с фиксированными логическими номерами отмечается как включённый в решётку

СРАВНЕНИЕ АЛГОРИТМОВ ОТКАЗОУСТОЙЧИВОСТИ

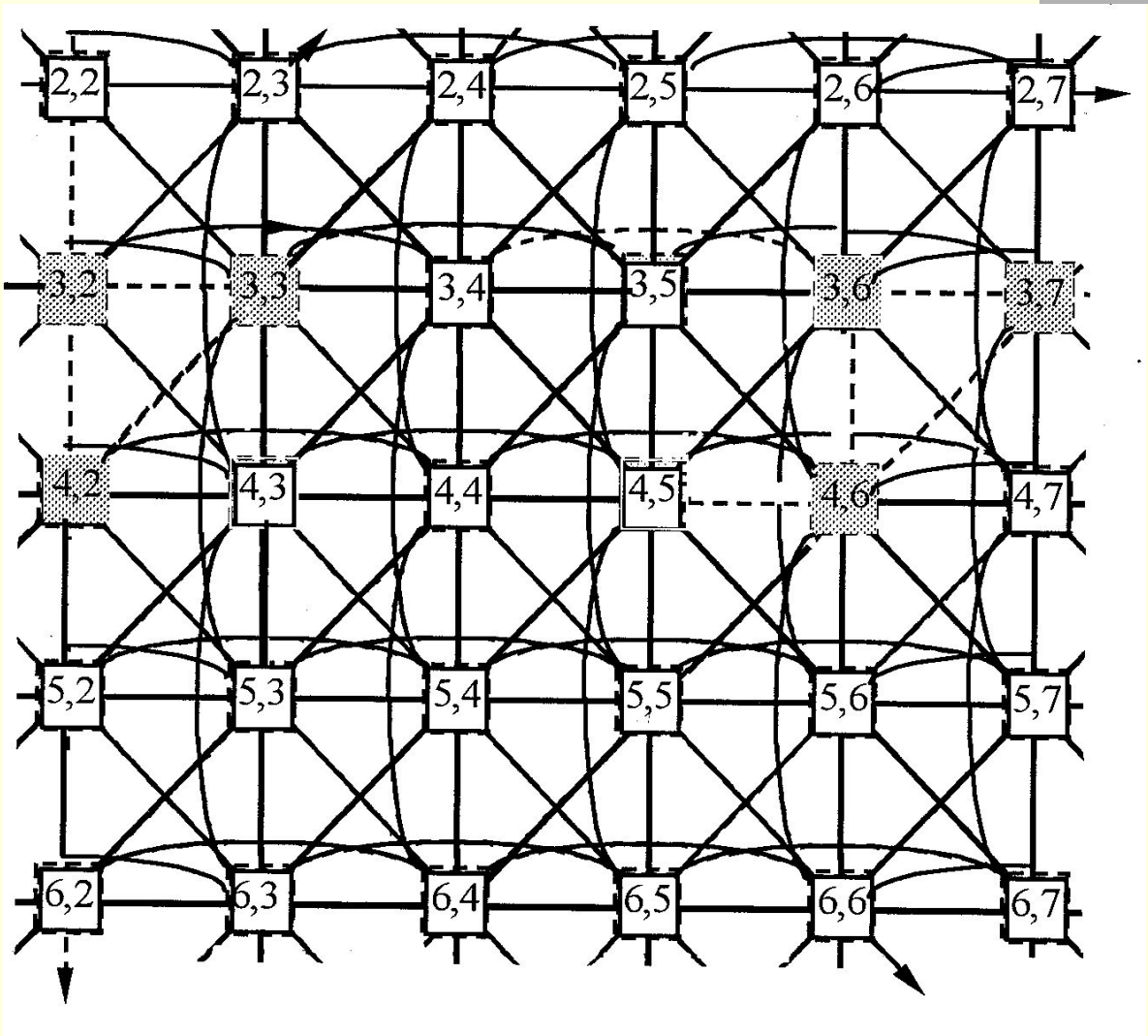


Вероятность ***P(n,p)*** сохранения работоспособности НПМ 20×20.

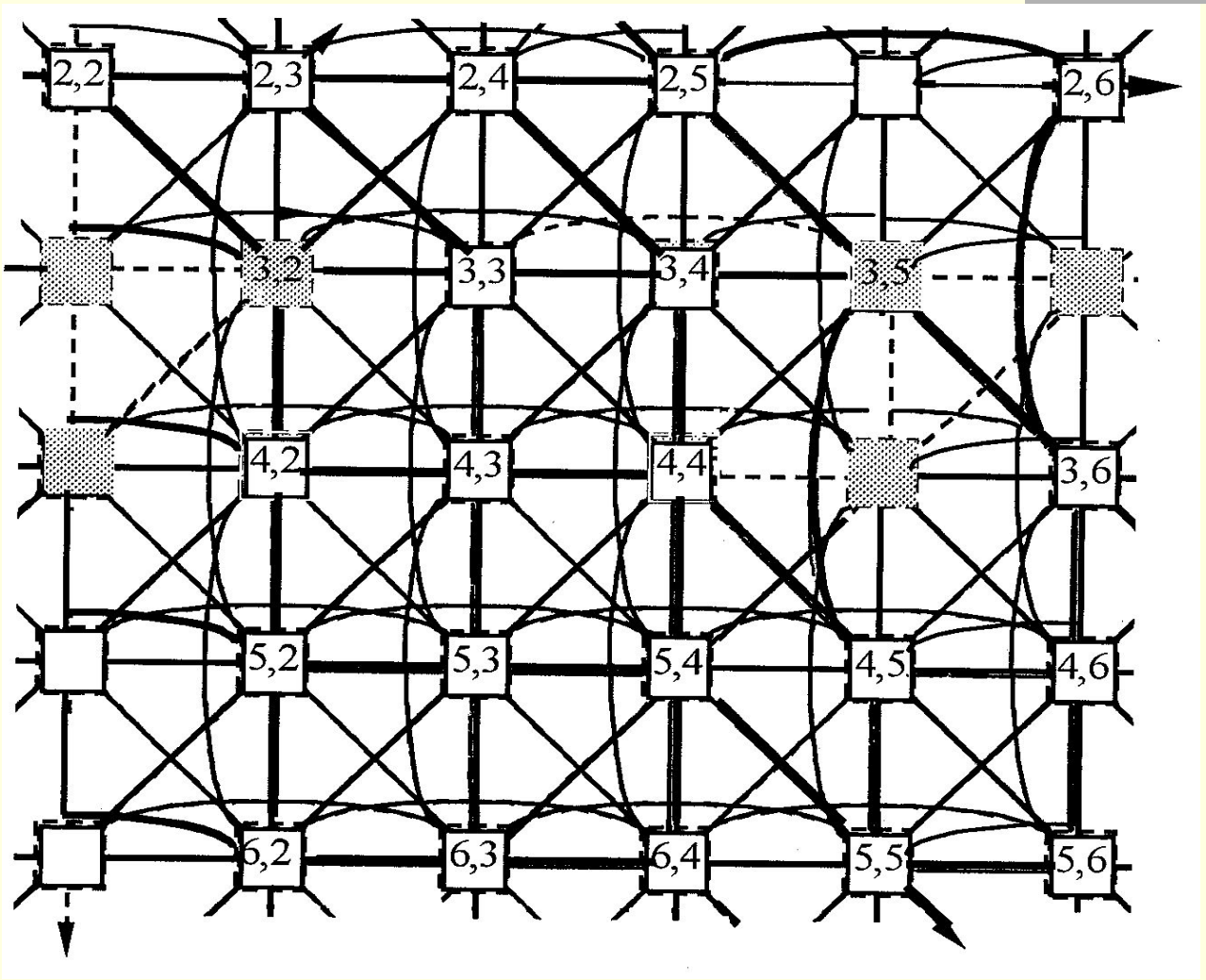
A - волновой алгоритм

B - алгоритм свободного захвата с программируемым резервом.

СИНДРОМ НЕСОГЛАСИЯ в НПМ размера 8×8 (фрагмент)



КОНСЕНСУС В НПМ размера 8×8 (фрагмент)



МЛП-ВЫЧИСЛЕНИЕ

1. Выполняется на локальных однородных структурах

2. Параллелизм максимален

3. Объём данных и вычислений в ЭМ минимален (1 виток цикла)

4. Взаимодействуют только соседи

5. Глобальные взаимодействия единичны: типа **пуск** и **останов**.

МЛП-РЕДУКЦИЯ СТРОК МАТРИЦЫ

суммирует параллельно N строк за время Nt с ускорением $(N-1)$ и эффективностью

$(N-1)/N \to 1$ при $N \to \infty$.

Каждая ЭМ содержит в локальной памяти один столбец длинны N и на индекс-регистре адреса – номер строки, сумму которой она накапливает на текущем этапе.

Символ s_{ij}^{k} обозначает сумму от i-го до j-го элементов k-той строки, перечисляемых циклически слева направо.

сдвиг суммы →

процессоры

сумма	0^1	0^2	0^3	0^4
индекс	1	2	3	4
память	x_{11}	x_{12}	x_{13}	x_{14}
	x_{21}	x_{22}	x_{23}	x_{24}
	x_{31}`	x_{32}	x_{33}	x_{34}
	x_{41}	x_{42}	x_{43}	x_{44}

начальное состояние

s_1^1	s_2^2	s_3^3	s_4^4
1	2	3	4
x_{11}	x_{12}	x_{13}	x_{14}
x_{21}	x_{22}	x_{23}	x_{24}
x_{31}`	x_{32}	x_{33}	x_{34}
x_{41}	x_{42}	x_{43}	x_{44}

этап 1

s_{41}^1	s_{12}^2	s_{23}^3	s_{34}^4
4	1	2	3
x_{11}	x_{12}	x_{13}	x_{14}
x_{21}	x_{22}	x_{23}	x_{24}
x_{31}`	x_{32}	x_{33}	x_{34}
x_{41}	x_{42}	x_{43}	x_{44}

этап 2

s_{31}^1	s_{42}^2	s_{13}^3	s_{24}^4
3	4	1	2
x_{11}	x_{12}	x_{13}	x_{14}
x_{21}	x_{22}	x_{23}	x_{24}
x_{31}`	x_{32}	x_{33}	x_{34}
x_{41}	x_{42}	x_{43}	x_{44}

этап 3

s_{21}^1	s_{32}^2	s_{43}^3	s_{14}^4
2	3	4	1
x_{11}	x_{12}	x_{13}	x_{14}
x_{21}	x_{22}	x_{23}	x_{24}
x_{31}`	x_{32}	x_{33}	x_{34}
x_{41}	x_{42}	x_{43}	x_{44}

этап 4

МЛП-УМНОЖЕНИЕ МАТРИЦ

Одним из примеров мелкозернистого распараллеливания может служить умножение матриц размера n^2 при использовании n^2 процессоров. Все три матрицы можно поэлементно распределить по клеткам, так что в каждой клетке $К_{ij}$ в начальный момент располагаются три элемента

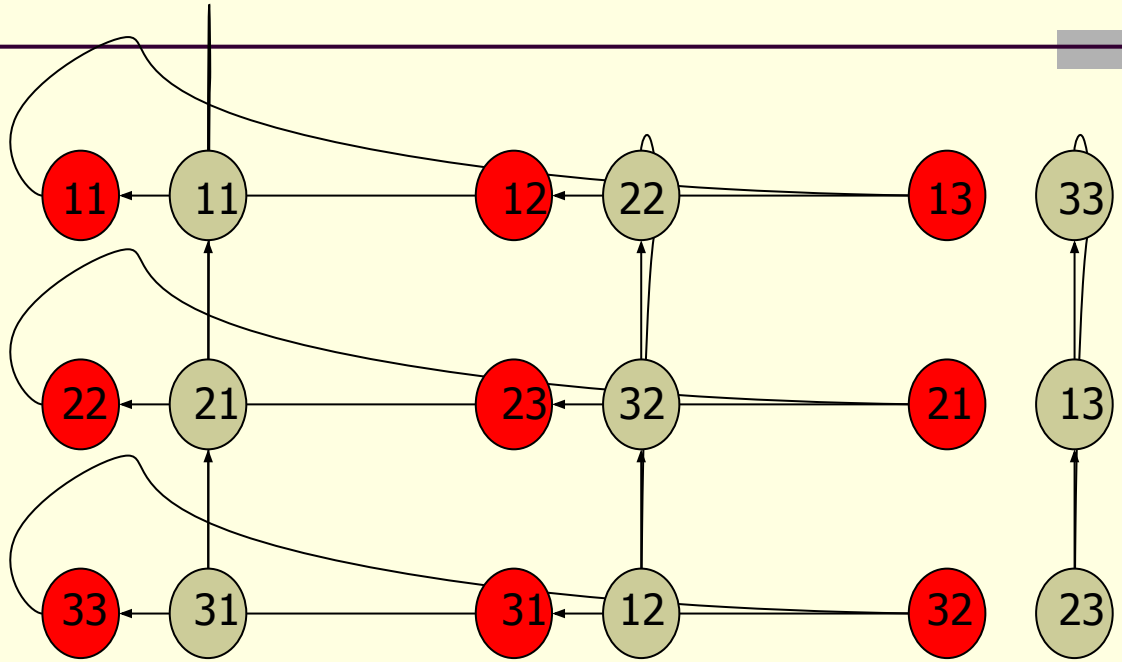
$$a_{ik}\ ,\ b_{kj} \text{ и } c_{ij} = 0.$$

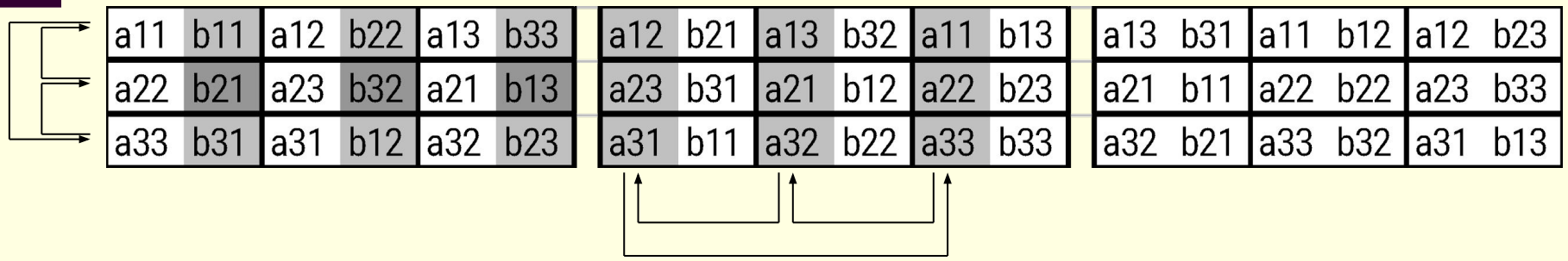
Тогда, при соответствующих сдвигах строк и столбцов, в каждом процессе можно выполнять операцию накопления

$$c_{ij} = a_{ik} \times b_{kj} + c_{ij}$$

Для выполнения такого параллельного алгоритма требуется всего по $O(n)$ этапа сдвига, умножения и сложения.

МЛП-УМНОЖЕНИЕ МАТРИЦ





ОЦЕНКА
МЛП-умножения матриц

Сложность параллельного алгоритма O(n).

Процесс вычислений состоит из 3-х команд:

1. пересылка исходных данных a_{ik} и b_{kj} на каждый процессор;
2. вычисление произведения блоков;
3. суммирование и запоминание результата c_{ij}.

Всего потребуется n таких действий. Тогда имеем:

$t_1(n) = n^3 T_{mult}$ – время умножения матриц размерности $n \times n$

$t_p(n) = n T_{mult}$ – время умножения матриц в $p = n^2$ ЭМ

$$S_p(n) = t_1(n)/ t_p(n) = n^3 T_{mult}/(n T_{mult}) = n^2$$

Итак, ускорение МЛП- алгоритма по сравнению с наилучшим последовательным алгоритмом равно n^2.

РЕЛЯТИВИСТСКИЙ КОМПЬЮТЕР ВОЗМОЖЕН,

МЛП-АЛГОРИТМЫ СУЩЕСТВУЮТ

для всех важнейших задач математической физики, линейной алгебры и дискретной математики.

ДЕРЗАЙТЕ!

Благодарю за внимание