

УДК 681.324

В. А. Воробьев, Н. В. Лаходынова
(Томск)

**ПРОЦЕССОРНАЯ МАТРИЦА С ПЕРЕСТРАИВАЕМОЙ СТРУКТУРОЙ И
ПЕРЕСТРАИВАЕМЫМ РЕЗЕРВОМ**

Предложен метод обеспечения отказоустойчивости процессорных матриц путем программного назначения резервных строк и столбцов. Разработаны алгоритмы перестройки структуры процессорной матрицы, реализующие гибкое управление резервом. Дана оценка эффективности предложенных алгоритмов.

Современная элементная база позволяет строить процессорные матрицы (ПМ) из сотен процессорных элементов (ПЭ). Стандартным требованием к техническому состоянию ПМ является сохранение работоспособности при кратных отказах ПЭ. Разработка отказоустойчивых ПМ становится особенно актуальной при изготовлении СБИС на кремниевых пластинах по неразрезной технологии. Вопрос отказоустойчивости напрямую связан с использованием пластин с большим числом неисправностей. Общеизвестным средством повышения отказоустойчивости ПМ являются введение в систему резервных элементов и перестройка структуры путем замены вышедших из строя ПЭ на резервные [1, 2]. Известные алгоритмы перестройки структуры ПМ (алгоритмы реконфигурации) ориентированы на фиксированное количество и расположение резервных элементов. Как правило, это строка, столбец или то и другое по краю матрицы. В [3] показано, что такой подход не обеспечивает требуемый уровень отказоустойчивости при увеличении размеров ПМ. Выходом из создавшегося положения могут служить увеличение числа резервных элементов и равномерное распределение резерва в матрице. В данной работе предлагаются алгоритмы реконфигурации структуры ПМ, допускающие гибкое управление резервом: любая строка и любой столбец могут быть зарезервированы. В результате с увеличением числа отказов будет падать производительность ПМ, но сохранится ее логическая структура — квадратная решетка.

Идея алгоритмов реконфигурации. Сохранение логической структуры ПМ требует не только резервных ПЭ, но и резервных связей. Исходная структура вертикальных и горизонтальных связей задается в зависимости от алгоритма реконфигурации при его описании. Реконфигурация заключается в отображении логической структуры в исправную часть физической структуры ПМ. С этой целью каждый ПЭ снабжается коммутационным окружением для выбора требуемых линий связи. Коммутационное окружение работает под управлением сигналов перестройки, получаемых схемами управления перестройкой. ПЭ, вместе с коммутационным окружением и схемами управления перестройкой, в дальнейшем называется элементом матрицы или элементарной машиной (ЭМ) (рис. 1).

Каждая ЭМ имеет собственные средства самодиагностики, вырабатывающие сигнал отказа $e(i, j)$: $e(i, j) = 1$, если ЭМ неисправна, и $e(i, j) = 0$, если ЭМ исправна.

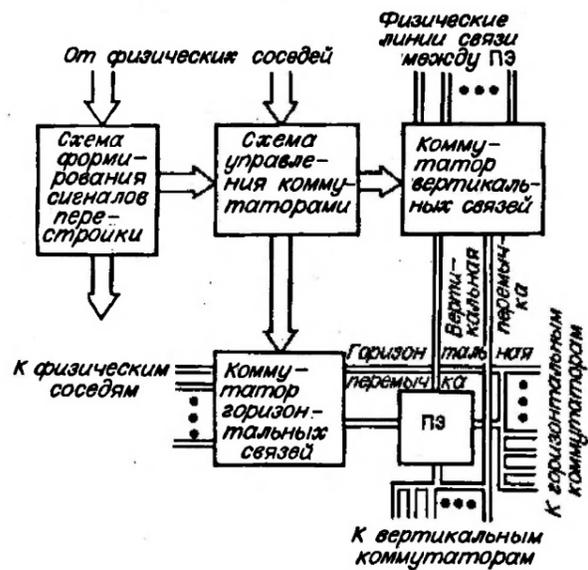


Рис. 1

Предполагается, что неисправные ЭМ распределены по матрице равномерно и выходы из строя ЭМ происходят независимо друг от друга. Кроме того, отказы в линиях связи не зависят от отказов ЭМ и анализируются отдельно.

Каждый элемент матрицы имеет средства для хранения сигналов a и b , которые задаются программно. Признак $a(i, j)$ определяет принадлежность элемента резервному столбцу, признак $b(i, j)$ — резервной строке; таким образом, любые строку и столбец матрицы можно сделать резервными.

Регулярная матрица размером $N \times N$ разбивается резервными строками и столбцами либо только столбцами на подматрицы. Процесс перестройки происходит внутри каждой подматрицы. Логические индексы (i', j') подматрицы отображаются на множество физических индексов (i, j) исправных элементов той же подматрицы, включая резервные. Если в подматрице есть отказавшие элементы, то логические и физические индексы не совпадают. Реконфигурация производится так, что логические соседи соединяются активными линиями связи непосредственно.

Итак, задача заключается в выборе способа отображения множества логических индексов на физические внутри каждой подматрицы. Этот выбор определяет эффективность алгоритмов реконфигурации, сложность реализации схем формирования сигналов перестройки и управления коммутаторами, число дополнительных линий связи и т. п.

Ниже предлагаются три алгоритма реконфигурации и проводится их сравнительный анализ.

Алгоритм прямой перестройки. Исходная матрица разбивается на подматрицы резервными строками и столбцами. После обнаружения отказа вся подматрица перестраивается. Перестройка каждой подматрицы проводится в двух направлениях: горизонтальный сдвиг вправо и вертикальный сдвиг вниз.

Процедура перестройки выглядит следующим образом:

А. Каждый столбец просматривается снизу вверх, и первый неисправный элемент помечается как вертикальный отказ.

Б. Все остальные отказы считаются горизонтальными.

В. Если каждая строка содержит не более одного горизонтального отказа, то выполняется сдвиг вправо. Каждому горизонтальному отказу присваивается логический индекс $(i, j + 1)$.

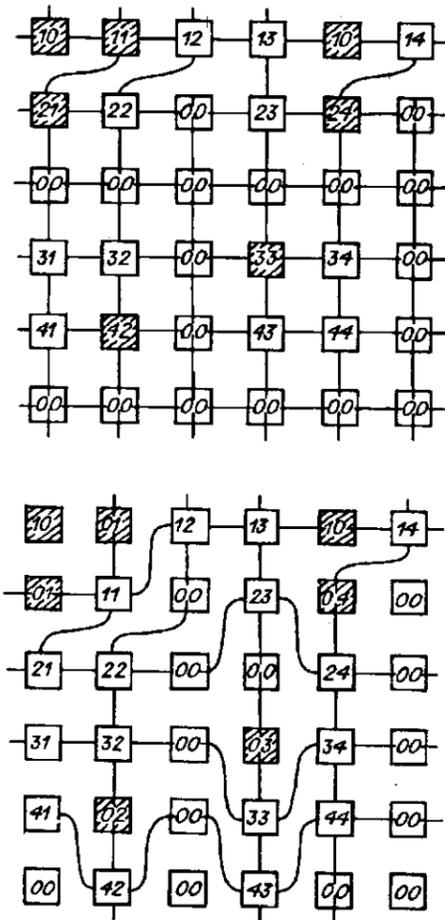


Рис. 2

Г. Если в столбце имеется не более одного вертикального отказа, выполняется вертикальный сдвиг и каждому вертикальному отказу присваивается логический индекс $(i + 1, j)$.

При нарушении условий В, Г вырабатывается сигнал фатального отказа. Действия А—Г выполняются в каждой подматрице. На рис. 2 показаны горизонтальный и вертикальный сдвиги, дефектные элементы заштрихованы, неиспользованные резервные элементы имеют логические индексы $(0, 0)$ и перекрыты вертикальными или горизонтальными перемычками. Если в любой из подматриц произошел фатальный отказ, процесс реконфигурации прекращается во всей матрице и может быть проведено переназначение резерва.

При перестройке матрицы в целом столбцы также просматриваются снизу вверх. При этом требуется обходить элементы резервных строк и так организовать процедуру обнаружения и обхода неисправностей, чтобы перестройка происходила внутри каждой подматрицы на ее резерве, а соединения между ними (подматрицами) были согласованы. Для организации такой процедуры недостаточно иметь только сигналы отказов каждого из элементов. Сигналы отказов имеют локальный характер, а процедура перестройки должна вносить глобальные изменения в структуру, учитывать ситуацию за пределами ближайшей окрестности отказавшего элемента. В связи с этим требуется обработка сигналов отказа и формирование на их основе специальных сигналов перестройки, которые используются для вычисления входов элементов.

Пусть $a(i, j)$ — признак принадлежности (i, j) -го ПЭ резервному столбцу j : $a(i, j) = 1$, если столбец j резервный; $b(i, j)$ — признак того, что элемент (i, j) принадлежит резервной строке; в этом случае $b(i, j) = 1$; $e(i, j)$ — сигнал отказа элемента (i, j) : $e(i, j) = 1$, если элемент (i, j) неисправен; $y(i, j)$ — входной сигнал вертикального отказа:

$$y(i, j) = \bigvee b(i, j)y'(i + 1, j).$$

Сигнал $y(i, j)$ распространяется снизу вверх по столбцам и принимает значение, равное 1, после прохождения самой нижней из неисправных ЭМ. При прохождении элемента резервной строки ($b(i, j) = 1$) сигнал обнуляется. Это дает возможность пометить в столбце j столько вертикальных отказов, сколько имеется резервных строк; $y'(i, j)$ — выходной сигнал вертикального отказа: $y'(i, j) = y(i, j) \vee e(i, j)$; $y_0(i, j)$ — сигнал отказа в j -м столбце, идущий вниз от (i, j) -го ПЭ:

$$y_0(i, j) = b(i, j)y'(i+1, j) \vee \neg b(i, j)y_0(i-1, j).$$

Сигнал $y_0(i, j) = 1$, если в j -м столбце подматрицы есть отказавшие элементы. Все элементы, для которых $y_0(i-1, j) = 1$ и $y(i, j) = 0$, вырабатывают сигнал вертикального сдвига $e_y(i, j) = 1$:

$$e_y(i, j) = y_0(i-1, j) \neg y(i, j).$$

При выходе с элемента резервной строки сигнал e_y обнуляется для того, чтобы не вызвать ложного запроса на вертикальный сдвиг у элементов с номерами $(i+1, j)$, $(i+2, j)$ и т. д.

Как было указано выше, в каждом столбце подматрицы может быть не более одного вертикального отказа. Остальные отказы помечаются как горизонтальные, при этом вырабатывается сигнал $e_x(i, j)$; $e_x(i, j)$ — сигнал сдвига вправо (горизонтальная перестройка):

$$e_x(i, j) = e'_x(i, j-1) \vee y(i, j)e(i, j);$$

$e'_x(i, j)$ — выходной сигнал сдвига вправо:

$$e'_x(i, j) = e_x(i, j) \neg a(i, j).$$

При прохождении резервного столбца ($a(i, j) = 1$) $e'_x(i, j) = 0$; $\Phi(i, j)$ — сигнал фатального отказа от элемента (i, j) , распространяющийся слева направо:

$$\Phi(i, j) = \Phi(i, j-1) \vee e'_x(i, j-1)e(i, j)y(i, j).$$

При появлении фатального отказа хотя бы в одной из подматриц вся матрица выходит из строя и следует сделать переназначение резервных строк или столбцов; $d_y(i, j)$ — сигнал вертикальной переемычки элемента (i, j) :

$$d_y(i, j) = e_y(i, j) \neg e_y(i-1, j) \vee b(i, j) \neg e_y(i, j).$$

Вертикальная переемычка включается в случае вертикального отказа элемента (i, j) или если элемент (i, j) принадлежит резервной строке и не востребован для вертикальной перестройки; $d_x(i, j)$ — сигнал горизонтальной переемычки элемента (i, j) :

$$d_x(i, j) = \neg e'_x(i-1, j)e_x(i, j) \vee a(i, j) \neg e_x(i, j) [\neg e_x(i-1, j) \vee \neg e_y(i-1, j)].$$

Горизонтальная переемычка возникает в случае горизонтального отказа элемента (i, j) , когда элемент (i, j) требует горизонтального сдвига, а сосед слева остается на месте. Второй случай включения горизонтальной переемычки: элемент (i, j) принадлежит резервному столбцу ($a(i, j) = 1$) и не участвует в горизонтальной перестройке, а его сосед сверху $(i-1, j)$ не участвует в горизонтальной и вертикальной перестройках одновременно, иначе (i, j) -й элемент будет востребован.

Выпишем сигналы управления коммутации.

Сигналы горизонтальной коммутации: $x_1 = d_x(i, j)$ — сигнал управления горизонтальной переемычкой:

$$x_1 = \neg e'_x(i-1, j)e_x(i, j) \vee a(i, j) \neg e_x(i, j) [\neg e_x(i-1, j) \vee \neg e_y(i-1, j)];$$

x_3 — сигнал коммутации $(i - 1, j - 1) \leftrightarrow (i, j)$:

$$x_3 = e_y(i, j) \uparrow e_y(i, j - 1) \uparrow d_y(i, j);$$

x_4 — сигнал коммутации $(i + 1, j - 1) \leftrightarrow (i, j)$:

$$x_4 = \uparrow e_y(i, j) e_y(i + 1, j - 1) \uparrow d_y(i + 1, j - 1).$$

Сигналы вертикальной коммутации.
Вертикальная перемычка:

$$y_1 = d_y(i, j) = e_y(i, j) \uparrow e_y(i - 1, j) \vee b(i, j) \uparrow e_y(i, j).$$

Коммутация $(i - 1, j) \leftrightarrow (i, j)$:

$$y_2 = \uparrow d_x(i, j) \uparrow d_x(i - 1, j) [e_x(i, j) = e_x(i - 1, j)].$$

Коммутация $(i - 1, j - 1) \leftrightarrow (i, j)$:

$$y_3 = e_x(i, j) \uparrow e_x(i - 1, j - 1) \uparrow d_x(i, j) [e_y(i, j) = e_y(i - 1, j - 1)].$$

Коммутация $(i - 1, j + 1) \leftrightarrow (i, j)$:

$$y_4 = \uparrow e_x(i, j) e_x(i - 1, j) [e_y(i, j) = e_y(i - 1, j + 1)].$$

Коммутация $(i, j - 1) \leftrightarrow (i, j)$:

$$y_5 = e'_x(i, j) \uparrow e_x(i - 1, j) e_y(i - 1, j - 1) \uparrow e_y(i, j) \uparrow d_x(i, j).$$

Коммутация $(i, j + 1) \leftrightarrow (i, j)$:

$$y_6 = \uparrow e_x(i, j) e_x(i - 1, j) e_y(i, j + 1) \uparrow e_y(i, j) \uparrow d_x(i, j).$$

Коммутация $(i - 2, j - 1) \leftrightarrow (i, j)$:

$$y_7 = e_x(i - 1, j) \uparrow e_x(i - 2, j - 1) \uparrow d_x(i, j) e_y(i - 1, j) \uparrow e_y(i - 2, j - 1).$$

Коммутация $(i - 2, j + 1) \leftrightarrow (i, j)$:

$$y_8 = \uparrow e_x(i - 1, j) e_x(i - 2, j) e_y(i - 1, j) \uparrow e_y(i - 2, j + 1).$$

Алгоритмы захвата. Захват резервных элементов заключается в том, что в случае отказа поиск резервных элементов можно проводить не только в той же строке или столбце, где произошел отказ, но и в соседней строке (столбце) по некоторым правилам. Изменяя эти правила, можно получать различные алгоритмы захвата.

Алгоритмы захвата отличаются от алгоритма прямой перестройки меньшей вероятностью фатального отказа и более эффективным использованием резерва.

По-прежнему сигналы $a(i, j)$ и $b(i, j)$, задаваемые программно, определяют принадлежность элемента (i, j) резервному столбцу и строке соответственно. Изначально логические индексы исправных и неисправных элементов совпадают с физическими. Захват элемента производится из верхней строки, поэтому желательно верхнюю строку подматрицы сделать резервной. Алгоритмы выполняются внутри каждой подматрицы.

1. Алгоритм ограниченного захвата заключается в следующем. Просмотр матрицы проводится построчно и начинается с нижней строки. При просмотре i -й строки обнаружение самого левого отказа (i, l) вызывает сдвиг вправо. Все исправные элементы, в том числе и резервные, получают логические индексы $(i', j') = (i, j - 1)$. Оставшиеся неисправные элементы (i, k) , $k \geq l$, захватывают элементы $(i - 1, k)$ из верхней строки. Захваченный элемент $(i - 1, k)$ получает логический индекс (i, k) .

При просмотре $(i - 1)$ -й строки захваченные элементы считаются неисправными. Фатальный отказ возникает при попытке захвата неисправного элемента или необходимости захвата для резервной строки.

Пусть, как и прежде, $e(i, j)$ — сигнал отказа элемента (i, j) , $e_x(i, j)$ — сигнал наличия неисправных или захваченных элементов в i -й строке. Для резервного элемента $e_x(i, j) = 0$:

$$e_x(i, j) = \bigwedge a(i, j) [e(i, j) \vee e_x(i, j - 1) \vee e_y(i, j)];$$

$e_y(i, j)$ — сигнал захвата: $e_y(i, j) = 1$, если элемент (i, j) захвачен или неисправен и не является самым левым отказом в строке, т. е. требует захвата в $(i - 1)$ -й строке. Сигнал требования захвата от элемента (i, j) обозначим $e'_y(i, j)$:

$$e'_y(i, j) = e_x(i, j - 1) [e_y(i, j) \vee e(i, j)],$$

$$e_y(i, j) = e'_y(i + 1, j) \bigwedge b(i + 1, j);$$

$\Phi(i, j)$ — сигнал фатального отказа, распространяющийся вправо:

$$\Phi(i, j) = \Phi(i, j - 1) \vee e'_y(i, j) e(i, j) \vee b(i, j) e'_y(i, j);$$

$d_x(i, j)$ — сигнал горизонтальной перемычки:

$$d_x(i, j) = e(i, j) \bigwedge a(i, j) \bigwedge e_y(i, j) \vee a(i, j) \bigwedge e_x(i, j) \vee \Phi(i, j - 1);$$

сигнал вертикальной перемычки:

$$d_y(i, j) = b(i, j) \bigwedge e'_y(i + 1, j) \vee \bigwedge b(i, j) e(i, j) e_y(i, j).$$

Используя сигналы перестройки, можно выписать сигналы управления коммутациями.

Пример перестройки по алгоритму ограниченного захвата дан на рис. 3.

2. Алгоритм свободного захвата является модификацией алгоритма ограниченного захвата, уменьшающей вероятность возникновения фатального отказа. Сложность соединений при этом не меняется. Отличие от алгоритма ограниченного захвата состоит в том, что в нем сдвиг вправо производится для самого левого отказавшего элемента, т. е. заранее определен элемент, относи-

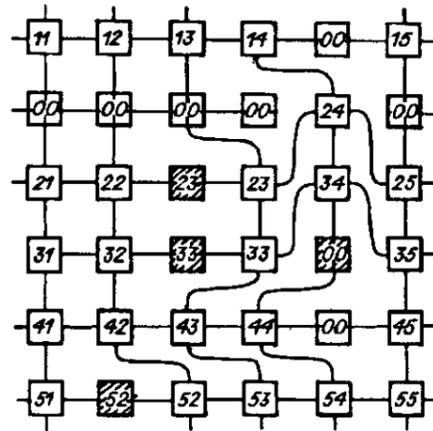


Рис. 3

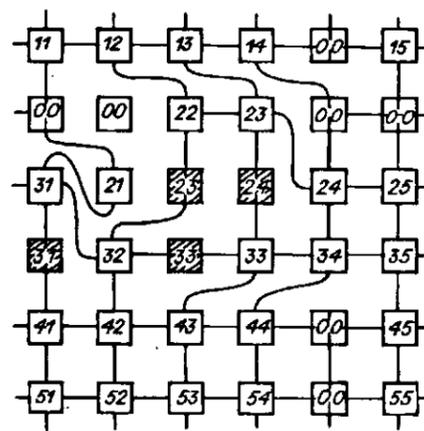


Рис. 4

тельно которого делается сдвиг вправо. В алгоритме свободного захвата сдвиг вправо в i -й строке производится относительно элемента (i, j) , которому соответствует неисправный или захваченный элемент $(i - 1, j)$. Все остальное выполняется как в алгоритме ограниченного захвата. Если ни один элемент (i, j) не имеет неисправного или захваченного соседа $(i - 1, j)$, то применяется алгоритм ограниченного захвата. Ситуация, изображенная на рис. 4, дает фатальный отказ при применении алгоритма ограниченного захвата. Структуру можно перестроить, используя алгоритм свободного захвата. Фатальный отказ в алгоритме свободного захвата получается, во-первых, если имеется строка i , в которой двум элементам (i, j) , (i, j') соответствуют неисправные или захваченные элементы $(i - 1, j)$, $(i - 1, j')$, во-вторых, если возникает необходимость захвата для элементов резервной строки.

Опишем сигналы перестройки.

Сигналы $a(i, j)$, $b(i, j)$, $e(i, j)$ те же, что и раньше.

$E(i, j)$ распространяется вправо от неисправного или захваченного элемента, если элемент $(i - 1, j)$ неисправен, т. е. (i, j) -й элемент не может осуществить захват:

$$E(i, j) = e(i - 1, j) [e_y(i, j) \vee e(i, j)] \vee a(i, j) \vee E(i, j - 1) \vee a(i, j);$$

$E_0 = \vee E(i, j)$, $E_0 = 1$, если хотя бы одному элементу в строке i соответствует

$E(i, j) = 1$; $e_x(i, j)$ — сигнал горизонтального сдвига:

$$e_x(i, j) = \{ [e(i, j) \vee e_y(i, j)] \vee e_x(i, j - 1) \vee E_0 \vee e(i - 1, j) \vee e_x(i, j - 1) \} a(i, j);$$

$e'_y(i, j)$ — сигнал захвата от элемента (i, j) ; $e_y(i, j)$ — сигнал о том, что элемент (i, j) захвачен:

$$e_y(i, j) = e'_y(i + 1, j) \vee b(i + 1, j),$$

$$e'_y(i, j) = [e(i, j) \vee e_y(i + 1, j)] \vee e_x(i, j - 1) E_0 \vee e_x(i, j - 1);$$

$\Phi(i, j)$ — сигнал фатального отказа:

$$\Phi(i, j) = \Phi(i, j - 1) \vee b(i, j) e'_y(i, j) \vee [e(i - 1, j) (e_y(i + 1, j) \vee e(i, j))] \vee a(i, j).$$

Сигналы $d_x(i, j)$, $d_y(i, j)$ такие же, как для алгоритма ограниченного захвата.

Эффективность алгоритмов реконфигурации. Сравнительный анализ алгоритмов реконфигурации дан в [1, 2] для фиксированного числа и расположения резервных элементов. Указаны показатели, по которым можно оценить целесообразность выбора того или иного алгоритма и структуры связей. Показатели таковы: вероятность сохранения работоспособности при данном количестве отказов, временные затраты на перестройку, аппаратные издержки на резервирование и дополнительные соединения. В [1] показано, что алгоритм прямой перестройки и алгоритм ограниченного захвата обеспечивают примерно одинаковую вероятность сохранения работоспособности, но первый алгоритм имеет лучшие характеристики при малом числе отказов, второй — снижает среднюю вероятность фатального отказа при большом числе неисправностей. Значительное увеличение вероятности сохранения работоспособности дает алгоритм свободного захвата. Его рекомендуется применять в случаях, когда требования по быстродействию не слишком жестки, а процессорные элементы достаточно велики. Все алгоритмы требуют дополнительных аппаратных затрат и избыточной площади кристалла, если речь идет о СБИС. Однако их преимущества по сравнению с тройным резервированием очевидны.

Введение программируемых резервных элементов, как было сделано выше, усложняет схемы формирования сигналов перестройки, увеличивает избыточность по площади, занимаемой резервными элементами, но структура соединения меняется незначительно. Очевидно, что увеличение числа резервных элементов удорожает систему и предложенные алгоритмы имеют целью только повышение вероятности сохранения работоспособности (уменьшение вероятности фатального отказа). Такая постановка задачи актуальна, во-первых, потому, что использование вычислительной техники в таких областях, как ядерная энергетика, вынуждает обеспечивать высокую надежность вычислений. Никакие затраты на усложнение вычислительных систем не могут сравниться с последствиями «фатального отказа». Во-вторых, при увеличении числа процессорных элементов алгоритмы [1] не обеспечат необходимый уровень надежности. Кроме того, немаловажна возможность манипулировать резервными элементами, назначать их по произволу разработчика.

На рис. 5 показана зависимость вероятности сохранения работоспособности $P(N, p)$ от вероятности исправности ЭМ p для разных алгоритмов: A — алгоритм прямой перестройки; B — алгоритм ограниченного захвата; C — алгоритм свободного захвата. Штриховыми линиями изображены результаты для алгоритмов [1], сплошными — для алгоритмов, предлагаемых выше.

Результаты получены моделированием на ЭВМ случайного числа отказов в матрице размером 20×20 , разбитой на четыре квадратные подматрицы резервной строкой и столбцом, кроме того, имеются резервные строка и стол-

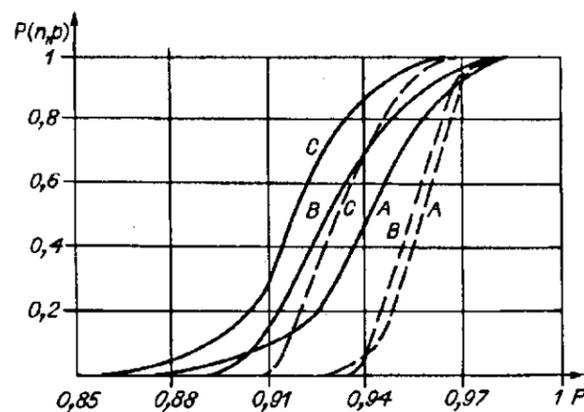


Рис. 5

без на периферии. Для алгоритмов [1] резервные строка и столбец есть только на периферии. Под вероятностью работоспособности понимается отношение фигурации.

СПИСОК ЛИТЕРАТУРЫ

1. Сами М., Стефанели Р. Перестраиваемые архитектуры матричных процессорных СБИС // ТИИЭР.—1986.—74, № 5.
2. Галушкин А. Н., Грачев Л. В., Толстых М. М., Точенов В. А. Оценка алгоритмов реконфигурации структуры вычислительной системы с МИМД-архитектурой // Кибернетика.—1990.—№ 2.
3. Воробьев В. А., Лаходынова Н. В. Пределы надежности однородных структур // Изв. АН СССР. Техн. кибернетика.—1989.—№ 3.

Поступила в редакцию 25 февраля 1994 г.
