

Н.Л. Ерёмина

АЛГОРИТМ ДИАГОНАЛЬНОГО ЗАХВАТА ДЛЯ РЕКОНФИГУРАЦИИ ПРОЦЕССОРНОЙ МАТРИЦЫ И ЕГО ЭФФЕКТИВНОСТЬ

Томский государственный педагогический университет

Введение

Вторая половина завершающегося столетия явилась периодом бурного развития электронно-вычислительной техники, ставшей необходимым инструментом во всех ключевых областях человеческой жизнедеятельности. От надежности ЭВМ зависит не только прогресс, но зачастую и само существование человека. Особое значение проблема повышения надежности ЭВМ приобретает в таких областях, как космическая и военная техника, атомная энергетика и др.

Современная технология проектирования и изготовления сверхбольших интегральных схем (СБИС) позволяет создавать логические устройства, содержащие сотни тысяч элементов в одном кристалле. При этом размеры отдельных элементов уменьшаются до тысячных долей миллиметра. Эти обстоятельства увеличивают вероятность возникновения отказа системы, т.е. такого состояния, при котором предоставляемые системой услуги отличаются от предусмотренных техническими условиями. Заслуживает особого внимания класс опасных, или катастрофических, отказов [1]. Последствия таких отказов по тяжести не сопоставимы с результатами нормальной работы системы при отсутствии отказов. Для такого класса отказов проблема обеспечения надежности становится особо актуальной.

Методы обеспечения надежности, или гарантируемости [1], систем разнообразны: разработка сверхнадежных компонентов, прогнозирование неисправностей и отказов, сложные процедуры аттестации изделия, контроль, диагностика и устранение неисправностей в период эксплуатации и пр. Однако на современном уровне развития технологии изготовления СБИС создать неразрезную пластину, содержащую сотни и тысячи элементов и при этом не имеющую ни одного дефекта, невозможно. Замена элементов на такой пластине также невозможна. Поэтому наиболее перспективным методом обеспечения надежности является создание отказоустойчивых систем, обладающих способностью обеспечивать предоставление услуг согласно техническому заданию даже при наличии отказов.

Метод реконфигурации для обеспечения отказоустойчивости процессорной матрицы.

Рассмотрим процессорную матрицу, представляющую собой прямоугольный массив процессо-

ров, расположенных на одной пластине СБИС. Введем следующие обозначения: I – число строк в матрице, J – число столбцов. Процессор, находящийся в строке i , $0 \leq i \leq I-1$, и в столбце j , $0 \leq j \leq J-1$, имеет физические координаты (i, j) , отражающие его положение в матрице. Каждый процессор (за исключением находящихся на краю матрицы, т.е. в строках с номерами 0 и $I-1$ либо в столбцах с номерами 0 и $J-1$) соседствует с четырьмя процессорами, имеющими физические координаты $(i-1, j)$, $(i, j-1)$, $(i+1, j)$, $(i, j+1)$. С точки зрения программирования процессорная матрица представляет собой квадратную решетку размера $I \times J$, в узлах которой расположены процессорные элементы. Будем считать систему работоспособной в том случае, если она сохраняет исходную структуру квадратной решетки.

Общепризнанным методом обеспечения отказоустойчивости процессорной матрицы является введение в нее избыточных элементов и связей и реконфигурация системы путем замещения неисправных процессоров резервными. В [2] избыточные элементы вводятся только по краю матрицы в виде резервной строки и резервного столбца или только резервного столбца; в [3] они располагаются в произвольных строках и столбцах исходной матрицы. Возможна постепенная деградация процессорной матрицы, т.е. уменьшение размеров решетки при условии неперемещения сохранения структуры связей.

Для того чтобы обеспечить возможность реконфигурации системы, необходимо ввести в нее дополнительные информационные связи между элементами. Таким образом, количество физических соседей каждого процессора достигает 14-16 элементов в зависимости от метода реконфигурации. Если резервные элементы не используются, то дополнительные связи находятся в пассивном состоянии. При включении резервного элемента в работу некоторые из них активизируются.

Предполагается, что процессорная матрица имеет средства самотестирования и для каждого процессорного элемента с физическими координатами (i, j) известно значение двоичного сигнала неисправности $e(i, j)$. Множество

$$\{e(i, j) / 1 \leq i \leq I-1, 1 \leq j \leq J-1\} \quad (1)$$

образует синдром неисправности процессорной

матрицы и служит исходной информацией для ее коррекции. Способ получения синдрома неисправности (1) не рассматривается в данном исследовании. Если в вычислительной системе есть отказы, т.е.

$$\exists (i, j), 1 \leq i \leq I-1, 1 \leq j \leq J-1 \{e(i, j) = 1\}, \quad (2)$$

происходит перестройка процессорной матрицы: избыточные связи исправных элементов активизируются так, чтобы система сохраняла исходную структуру; у неисправных элементов, вырабатывающих сигнал $e(i, j) = 1$, наоборот, все связи блокируются, в связи с чем они не могут участвовать в вычислениях и фактически исключаются из процессорной матрицы.

Таким образом, задача реконфигурации сводится к отображению логической структуры квадратной решетки в избыточную физическую структуру. С этой целью каждый процессорный элемент снабжается коммутационным окружением, состоящим из двух коммутаторов, вертикального и горизонтального, и блока управления, вырабатывающего сигналы перестройки и управления коммутаторами. Коммутационное окружение процессорного элемента использует сигналы $e(i, j)$ от самого процессора и его соседей и обеспечивает корректирующую реконфигурацию структуры, т.е. упомянутое отображение. При этом процессор (i, j) выбирает соседей сверху и слева от себя (при просмотре матрицы начиная с нижнего правого ее элемента). Выбор соседей снизу и справа реализуют соседние процессорные элементы. В том случае, если реконфигурация невозможна, т.е. нельзя получить исправную квадратную решетку требуемого размера, вырабатывается сигнал фатального отказа.

Кроме физических координат (i, j) , каждый процессор имеет логические координаты (i', j') . Первоначально логические координаты совпадают с физическими. В процессе реконфигурации логические координаты изменяются и отображают положение процессорного элемента в исправной матрице, на которой будут происходить вычисления. Неисправные процессоры не имеют логических координат. Логические координаты исправных процессоров вычисляются с помощью специальных алгоритмов адресации. Наличие логических координат процессорного элемента существенно облегчает программирование матрицы.

Результат перестройки не в последнюю очередь зависит от выбранного алгоритма реконфигурации. При одном и том же расположении отказов в процессорной матрице различные алгоритмы могут получить различные конфигурации связей между элементами. Возможна ситуация, когда, следуя одному алгоритму, можно получить исправную квадратную решетку, а при исполь-

зовании другого алгоритма возникает ситуация фатального отказа матрицы.

Сравнительный анализ предложенных в [2] и [3] алгоритмов реконфигурации показал, что наибольшую эффективность имеет алгоритм свободного захвата [4]. Вместе с тем для алгоритма свободного захвата ситуация фатального отказа возникает при наличии двух пар отказов

$$\{(i, j_1), (i+1, j_1)\}, \{(i, j_2), (i+1, j_2)\}, j_1 \neq j_2, \quad (3)$$

т.е. структура любого, в том числе достаточно большого, размера может выйти из строя при наличии всего лишь четырех неисправных процессоров. С целью повышения отказоустойчивости процессорной матрицы нами предлагается алгоритм диагонального захвата, свободный от указанного недостатка.

Алгоритм диагонального захвата

Для упрощения описания алгоритма введем специальную терминологию.

Возьмем в матрице произвольный процессорный элемент (i, j) . Он соседствует с двумя элементами по вертикали и с двумя по горизонтали. Процессорный элемент соединен вертикальными связями с элементом $(i+1, j)$ из нижней строки и с элементом $(i-1, j)$ из верхней строки. Назовем элемент $(i+1, j)$ «соседом снизу» рассматриваемого элемента (i, j) , а элемент $(i-1, j)$ его «соседом сверху». Элементы $(i, j-1)$ и $(i, j+1)$, соединенные с элементом (i, j) горизонтальной связью, назовем его «соседом слева» и «соседом справа» соответственно. В процессе реконфигурации процессорной матрицы логические индексы элементов изменяются, и, возможно, функции элемента (i, j) будет исполнять другой процессорный элемент, до перестройки имевший логические индексы (i', j') . В этом случае будем говорить, что элемент (i', j') является заместителем элемента (i, j) .

Идея алгоритма перестройки с диагональным захватом состоит в следующем. Наличие избыточных связей и избыточных процессоров в матрице позволяет заменять элемент его соседями не только по вертикали и горизонтали, как в [Z], но и по диагонали, что позволяет снизить вероятность фатального отказа процессорной матрицы. Каждый процессорный элемент, будучи неисправным, передает свои функции одному из возможных заместителей. В качестве же заместителей могут выступать не только «сосед сверху» $(i-1, j)$ и «сосед справа» $(i, j+1)$, как в алгоритмах захвата [Z], но и элемент $(i-1, j-1)$ – «диагональный сосед сверху» отказавшего процессорного элемента (i, j) .

Как и прежде, будем рассматривать процессорную матрицу размера $I \cdot J$. Резервные элементы расположим в столбце и строке с максималь-

ными номерами, т.е. справа и сверху на периферии матрицы. Дадим следующие определения.

Определение 1. Пусть неисправный процессор имеет физический номер (k, l) , $0 \leq k \leq I-2$, $0 \leq l \leq J-1$, и его «сосед сверху» $(k-1, l)$ исправен. Процедуру отображения функций неисправного элемента (k, l) на исправный процессор $(k-1, l)$ назовем прямым захватом, а элемент $(k-1, l)$ – прямым заместителем элемента (k, l) .

Определение 2. Пусть неисправный процессор имеет физический номер (k, l) , $0 \leq k \leq I-2$, $0 \leq l \leq J-2$, и его «диагональный сосед сверху» $(k-1, l+1)$ исправен. Процедуру отображения функций неисправного элемента (k, l) на исправный процессор $(k-1, l+1)$ назовем прямым захватом, а элемент $(k-1, l+1)$ – диагональным заместителем элемента (k, l) .

Перейдем к изложению собственно алгоритма диагонального захвата.

Процессорная матрица просматривается построчно снизу вверх, начиная от строки с наибольшим номером $i = I-1$ и заканчивая второй сверху строкой с номером $i = 1$. В каждой строке процесс реконфигурации идет слева направо, от элемента с наименьшим номером столбца $j = 0$ до его максимального номера $j = J-1$.

Один из отказов в строке объявляется горизонтальным. Горизонтальным отказом считается отказавший элемент, для которого невозможно выполнить ни прямой, ни диагональный захват. В том случае, если таких отказов в строке нет, горизонтальным объявляется крайний справа отказ, т.е. отказавший процессор (k, l) с максимальным номером столбца l .

Для нейтрализации горизонтального отказа осуществляется перестройка вправо: отказавший элемент замещается своим «соседом справа», тот, в свою очередь, также замещается своим «соседом справа», и т.д. Предпоследний элемент в строке замещается элементом из резервного столбца. Таким образом, происходит сдвиг процессорных элементов в строке матрицы слева направо. Важно отметить, что только один отказ в каждой строке может быть замещен подобным образом.

Остальные отказы в этой строке замещаются с помощью прямого либо диагонального захвата. Если отказавший элемент невозможно заместить исправным, возникает ситуация фатального отказа, и построение квадратной решетки требуемого размера из исправных элементов процессорной матрицы невозможно.

Итак, для каждого отказа либо захваченного ранее элемента (k, l) , $0 \leq k \leq I-2$, $0 \leq l \leq J-1$, выполняется следующая процедура.

1. ЕСЛИ (k, l) – самый правый отказ в строке и в строке k не было горизонтального отказа,

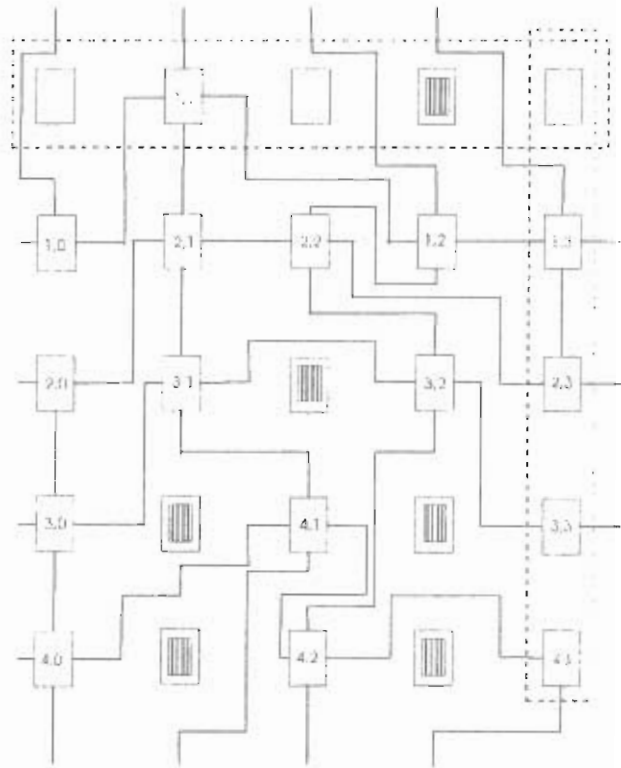


Рис. 1. Пример реконфигурации процессорной матрицы по алгоритму диагонального захвата.

ТО (k, l) считается горизонтальным отказом и нейтрализуется перестройкой вправо; переход к п. 5;

ИНАЧЕ переход к п. 2.

2. ЕСЛИ элемент $(k-1, l)$ исправен и не захвачен по диагонали элементом $(k, l-1)$,

ТО происходит прямой захват; переход к п. 5; ИНАЧЕ переход к п. 3.

3. ЕСЛИ отказ (k, l) не принадлежит резервному столбцу, т.е. $l \neq J-1$, и элемент $(k-1, l+1)$ исправен,

ТО происходит диагональный захват; переход к п. 5;

ИНАЧЕ переход к п. 4;

4. ЕСЛИ в строке k не было горизонтального отказа,

ТО (k, l) считается горизонтальным отказом и нейтрализуется перестройкой вправо; переход к п. 5;

ИНАЧЕ возникает ситуация фатального отказа и перестройка матрицы невозможна (аварийное завершение процедуры);

5. КОНЕЦ процедуры.

Условием фатального отказа предлагаемого алгоритма является невозможность захвата для неисправного элемента при наличии в данной строке горизонтального отказа.

Пример реконфигурации процессорной матрицы по алгоритму диагонального захвата показан на рис. 1.

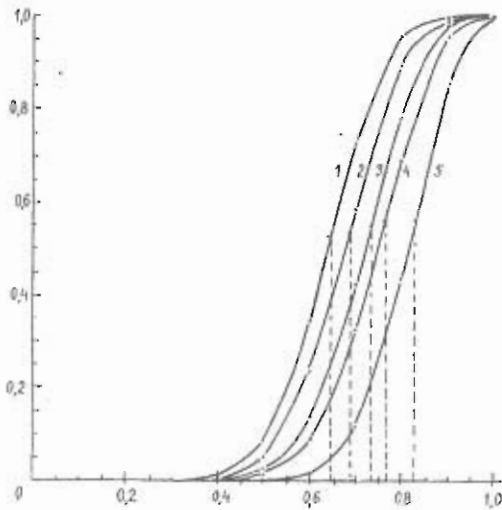


Рис. 2. Вероятность $P(5,5,p)$ сохранения работоспособности процессорной матрицы 5×5 при различных способах реконфигурации: 1, 2, 3, 4 – алгоритмы диагонального захвата, свободного захвата, прямой перестройки и ограниченного захвата, соответственно, с резервной строкой и столбцом; 5 – алгоритмы захвата, в которых резервировался только столбец (значения $P(I,J,p)$ очень близки, так что кривые сливаются).

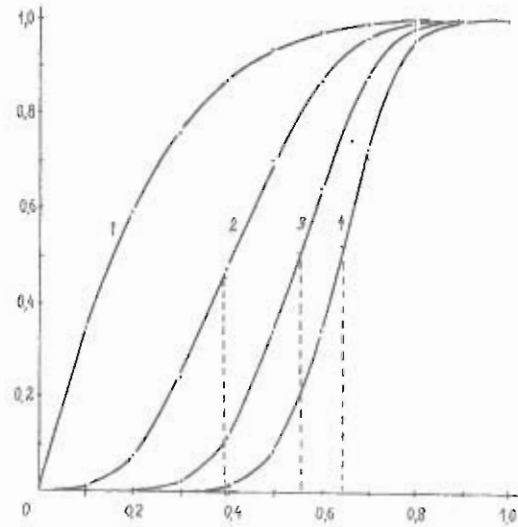


Рис. 3. Вероятность $P(I,J,p)$ сохранения работоспособности процессорных квадратных ($I=J$) матриц различной размерности при реконфигурации по алгоритму диагонального захвата с резервным столбцом и резервной строкой:
1 – матрица 2×2 ; 2 – матрица 3×3 ;
3 – матрица 4×4 ; 4 – матрица 5×5 .

Эффективность алгоритма диагонального захвата

Легко видеть, что множество конфигураций отказов в процессорной матрице, для которого по алгоритму свободного захвата может быть построена квадратная решетка заданного размера, является подмножеством аналогичного множества для алгоритма свободного захвата [2]. Вместе с тем ситуация (3) существования четырех неисправных процессорных элементов в системе не вызывает, как при свободном захвате, сигнала фатального отказа. Исключением является случай, когда неисправны две пары процессоров на правом краю матрицы:

$$\{(i, J-2), (i+1, J-2)\}, \{(i, J-1), (i+1, J-1)\}, i < I-1. \quad (4)$$

Поэтому естественно ожидать, что эффективность предложенного алгоритма превысит эффективность алгоритмов захвата, особенно при небольшом числе отказов в процессорной матрице.

Для подтверждения этой гипотезы был проведен сравнительный анализ эффективности алгоритма диагонального захвата и алгоритмов реконфигурации, предложенных в [2, 3].

Отказоустойчивость процессорной матрицы характеризуется вероятностью сохранения работоспособности $P(I, J, p)$, где I – количество строк в процессорной матрице, J – количество столбцов, p – вероятность исправности одного процессорного элемента. Эффективность алгоритмов

реконфигурации сравнивалась методом второй производной [5]. Для матриц небольших размеров вид функции $P(I, J, p)$ может быть получен аналитически:

$$P(I, J, p) = \sum_{i=0}^{I+J} a_i p^i (1-p)^{I+J-i}, \quad (5)$$

где коэффициент a_i равен количеству различных конфигураций i отказов в процессорной матрице. Эти коэффициенты определяются полным перебором всех возможных конфигураций отказов и подсчетом числа тех из них, которые позволяют получить квадратную решетку из оставшихся исправными процессоров. Отсюда легко определить первую и вторую производные функции $P(I, J, p)$, $P'(I, J, p)$:

$$P'(I, J, p) = \sum_{i=1}^{I+J} b_i p^{i-1} (1-p)^{I+J-i}, \quad (6)$$

где

$$b_i = ia_i + (i-1-I \times J)a_{i-1}; \quad (7)$$

$$P''(I, J, p) = \sum_{i=2}^{I+J} c_i p^{i-2} (1-p)^{I+J-i}, \quad (8)$$

где

$$c_i = (i-1)b_i + (i-1-I \times J)b_{i-1} = [(i-1)i]a_i + [2(i-1)(i-1-I \times J)]a_{i-1} + [2(i-1-I \times J)(i-2-I \times J)]a_{i-2}. \quad (9)$$

Абсцисса точки перегиба кривой функции $P(I, J, p)$, $0 < p < 1$ задает критическое значение вероятности $p = p_{II}$ исправности процессорного элемента. Если $p < p_{II}$, значение функции $P(I, J, p)$ резко падает до нуля при уменьшении p , при $p > p_{II}$ оно стремится к единице. Значение p_{II} является корнем уравнения

$$P(I, J, p) = 0. \quad (10)$$

Приближенное значение корня уравнения (10) легко можно найти любым известным способом, например методом деления отрезка пополам.

Следует подчеркнуть, что полученный результат носит аналитический, а не статистический характер.

Описанная процедура реализована на персональном компьютере в виде пакета программ с использованием Turbo C (версия 2.0) фирмы Borland International и среды программирования «Математика» (версия 3.0) фирмы Wolfram Research.

Были исследованы структуры, содержащие до 30 элементов. Для каждой из них и каждого алгоритма реконфигурации получен аналитический вид функций $P(I, J, p)$, построены графики $P(I, J, p)$ и найдено значение критической вероятности p_{II} .

На рис. 2 показаны графики функций $P(I, J, p)$ для различных алгоритмов при фиксированном размере матрицы 5×5 . Видно, что для алгоритма захвата чем меньше p_{II} , тем выше лежит график $P(I, J, p)$, и, следовательно, тем меньше требования к выходу годных процессорных элементов. Полностью подтвердилось предположение о наибольшей эффективности алгоритма диагонального захвата. Так, для матрицы 5×5 с 16 ис-

правными элементами возможно 2 042 975 различных конфигураций отказов. Из них для 1 228 794 конфигураций возможно построение исправной квадратной решетки 4×4 методом диагонального захвата, тогда как методом свободного захвата – для 458 331, ограниченного захвата – для 340 619, непосредственной перестройки – лишь для 131 525 конфигураций. Таким образом, вероятность восстановления решетки 4×4 в избыточной процессорной матрице 5×5 с девятью отказами для алгоритма диагонального захвата выше, чем для свободного захвата в 2,7 раза, ограниченного захвата в 3,6 раза и непосредственной перестройки – в 9,3 раза. С ростом размерности структуры значение критической вероятности для алгоритма диагональной перестройки также возрастает, стремясь в пределе к единице. Тем не менее оно остается ниже, чем значение критической вероятности для других алгоритмов. Итак, можно утверждать, что применение алгоритма диагонального захвата для обеспечения отказоустойчивости процессорной матрицы дает наилучшие результаты.

На рис. 3 демонстрируется зависимость $P(I, J, p)$ от размеров структуры для алгоритма диагонального захвата (рассмотрены квадратные процессорные матрицы, у которых число строк равно числу столбцов). Видно, что с ростом числа процессорных элементов требования к надежности отдельного процессора возрастают. Абсцисса p_{II} точки перегиба кривой $P(I, J, p)$ «дрейфует» вправо с ростом $I \cdot J$; чем меньше p_{II} , тем выше лежит кривая. Эти результаты полностью согласуются с [5, 6].

Значения p_{II} , полученные для решеток разных размеров, содержащих до 30 процессорных эле-

Таблица
Значение критической вероятности для матриц $I \times J$ с перестройкой по алгоритму диагонального захвата

$I \setminus J$	2	3	4	5	6	7	8	9	10	11	12	13	14
2	0,000	0,207	0,305	0,366	0,408	0,441	0,466	0,487	0,505	0,521	0,535	0,547	0,557
3	0,216	0,392	0,474	0,525	0,560	0,586	0,607	0,624					
4	0,320	0,480	0,555	0,601	0,633	0,657							
5	0,383	0,532	0,603	0,646									
6	0,426	0,567	0,635										
7	0,457	0,592	0,658										
8	0,481	0,611											
9	0,501	0,626											
10	0,517												
11	0,531												
12	0,543												
13	0,553												
14	0,563												
15	0,570												

ментов, с перестройкой по алгоритму диагонального захвата, приведены в таблице.

Исходя из полученных результатов, можно сделать вывод, что для алгоритма диагонального захвата наибольшей надежностью обладают структуры, вытянутые в горизонтальном направлении.

Для уменьшения вероятности возникновения фатального отказа в [3] алгоритмы реконфигурации модифицированы следующим образом. Матрица разбивается резервными строками и столбцами на подматрицы. Перестройка структуры происходит внутри каждой подматрицы. Если перестройка в одной из подматриц невозможна, вырабатывается сигнал отказа всей процессорной матрицы.

Выводы, сделанные для матриц малых размеров, естественно отнести к подматрицам, кото-

рые получаются разбиением исходной процессорной матрицы резервными строками и столбцами.

Для алгоритма диагонального захвата целесообразно размещать резервные элементы преимущественно в строках процессорной матрицы, разбивая ее на подматрицы, вытянутые в горизонтальном направлении.

Заключение

Итак, в статье описан алгоритм диагонального захвата, обеспечивающий отказоустойчивость процессорной матрицы на СБИС. Он позволяет полнее использовать введенную избыточность и тем самым повысить надежность схемы. Показана высокая эффективность диагонального захвата по сравнению с ранее предложенными алгоритмами [2, 3]. Даны рекомендации по выбору расположения резерва.

Литература

1. Авиженис А., Лалри Ж.-К. Гарантоспособные вычисления: от идеи до реализации в проектах // ТИИЭР. Т. 74. № 5. 1986. С. 8-21.
2. Сами М., Стефанелли Р. Перестраиваемые архитектуры матричных процессорных СБИС // ТИИЭР. 1986. Т. 74. № 5. С. 93-107.
3. Воробьев В.А., Лаходынова Н.В. Процессорная матрица с перестраиваемой структурой и перестраиваемым резервом // Автоматика. 1994. № 5. С. 90-98.
4. Воробьев В.А., Ерёмкина Н.Л., Лаходынова Н.В. Анализ алгоритмов перестройки структуры процессорной матрицы // Автоматика. 1996. № 2. С. 69-77.
5. Воробьев В.А., Лаходынова Н.В. Пределы надежности однородных вычислительных систем // Вычислительные системы. 1988. Вып. 126. С. 122-149.
6. Воробьев В.А., Лаходынова Н.В. Пределы надежности однородных структур // Изв. АН СССР. Технич. киб-ка. 1989. № 3. С. 110-113.

УДК 681.324

Н.Л. Ерёмкина

АЛГОРИТМ АДРЕСАЦИИ ДЛЯ РЕКОНФИГУРАЦИИ ПРОЦЕССОРНОЙ МАТРИЦЫ ПУТЕМ ДИАГОНАЛЬНОГО ЗАХВАТА

Томский государственный педагогический университет

Введение

Предлагаются алгоритм адресации отказоустойчивой процессорной матрицы с перестраиваемой структурой.

Матрица разбивается резервными строками и столбцами на подматрицы. Перестройка структуры происходит внутри каждой подматрицы за счет использования резервных процессоров и линий связи. Каждый процессорный элемент (i, j) имеет коммутационное окружение, соединяющее его с исправными соседями на основе сигнала неисправности от самого элемента и

его соседей так, чтобы в результате получилась квадратная решетка. Если перестройка невозможна, вырабатывается сигнал отказа всей матрицы. Алгоритм диагонального захвата заключается в том, что отказы процессорных элементов компенсируются сдвигами используемых процессоров по физической структуре.

В данной работе получены логические уравнения для вычисления управляющих сигналов перестройки, на основе которых выполняется переконмутация матрицы и вычисляются новые логические индексы процессорных элементов.