

Л.В. ДЕРБУНОВИЧ, д-р техн. наук, проф. НТУ «ХПИ»,
М.А. БЕРЕЖНАЯ, канд. техн. наук, доц. ХНУРЭ,
Я.Ю. КОРОЛЕВА, асс. ХНУРЭ

ОДНОРОДНЫЕ СЕТИ С РАСПРЕДЕЛЕННОЙ СИСТЕМОЙ РЕКОНФИГУРАЦИЙ

Запропоновано метод та алгоритм реконфігурації розрядномодульних однорідних мереж (PMOM) з розподіленою реконфігурацією резервних та функціонуючих модулів. Запропонований підхід може застосовуватися до PMOM різної розмірності та призначення, в яких несправний модуль виявляється вбудованими засобами діагностування, а реконфігурація здійснюється під управлінням HOST процесора.

This paper presents an effective reconfiguration method and algorithm of reconfiguring one and two-dimensional degradable arrays with four – port switches, when processing elements of arrays become faulty.

Постановка проблемы. Развитие электронных технологий и возможность использования СБИС, у которых на одном кристалле размещается определенное число процессорных модулей (ПМ) позволяет создавать высокопроизводительные одномерные и двумерные сети конвейерной и параллельной обработки информации. С возрастанием числа ПМ в таких PMOC проблема повышения надежности и отказоустойчивости функционирования сетей становится особенно важной и актуальной.

Анализ литературы. В [1,2] проведен анализ существующих подходов, методов и алгоритмов реконфигурации PMOC. В одном из них используются резервные ПМ, позволяющие реконфигурировать PMOC при обнаружении неисправных ПМ и замены их исправными модулями, что обеспечивает работоспособность сети без снижения ее производительности [2, 3, 4]. В другом подходе неисправный ПМ исключается из сети путем перекоммутации маршрутов и связей между модулями и распределение функций неисправного ПМ между исправными. Это обеспечивает экономичность сети, но снижает ее производительность [5, 6, 7, 8].

Цель статьи. Основным недостатком существующих методов и алгоритмов реконфигурации PMOC является возможность нарушения работоспособности сети при наличии в ней неисправных коммутирующих модулей. Для исключения этого недостатка предлагается использовать самопроверяемые четырехпортовые коммутаторы входных и выходных шин процессорных модулей сети [9,10,11,12].

Одномерные РМОС с резервными модулями. Распределенная процедура реконфигурации РМОС реализуется путем использования четырехпортовых коммутаторов. На рис.1 представлена обобщенная структура процессорного модуля, входы и выходы которого коммутируются под управлением HOST процессора.

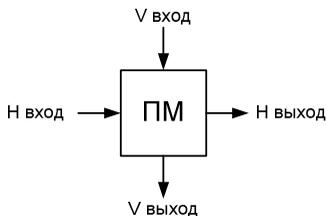
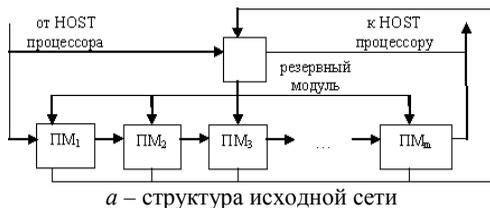


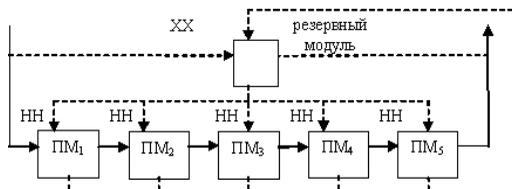
Рис. 1. Процессорный модуль реконфигурируемой РМОС

Модуль имеет множество горизонтальных (H) и вертикальных (V) входов и выходов. В каждом модуле сети передача данных может осуществляться по четырем маршрутам HH , HV , VH и VV , где HH обозначает передачу с H входа на H выход и т.д.

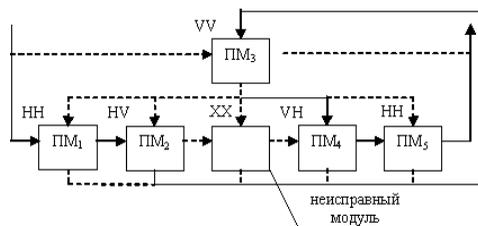
На рис. 2, *a* представлена структура одномерной однонаправленной реконфигурируемой сети из « m » процессорных модулей и одного резервного ПМ. На примере сети из пяти ПМ ($m = 5$) и одного резервного модуля ($s = 1$) представлены структуры исправной сети (рис. 2, *б*) и реконфигурированной сети при неисправном ПМ₃. Сплошными линиями показаны маршруты передачи данных в сети, в которые активизируются $HOST$ процессором в соответствии с алгоритмом управления распределенной системой реконфигурации.



a – структура исходной сети



б – структура сети при нормальном функционировании с $m = 5$



в – структура сети с заменой неисправного модуля ПМ₃

Рис. 2. Система распределенной реконфигурации с одним резервным модулем

Основным недостатком такой структуры является возможность искажения данных на общей шине в случае ненадежного отключения неисправного модуля.

Двумерные РМОС с резервными модулями. Процессорные модули с четырехпортовыми коммутаторами входов и выходов могут быть соединены в двумерный массив, как показано на рис. 3, а. на рис. 3, б, в показан процесс реконфигурации.

Как видно из рис. 3, все 9 модулей могут быть соединены в одну цепочку. При отказе любого модуля оставшиеся могут выполнять операции. Для $m = 8$ и $s = 1$ такая РМОС является отказоустойчивой для класса неисправностей одного произвольного ПМ_{*i*}. Для $m = 6$ и $s = 3$ она является отказоустойчивой по отношению к произвольным двум и к некоторым трем неисправным модулям.

Синтез алгоритма реконфигурации РМОС определяется необходимостью решения двух следующих задач:

1. Структурная организация РМОС, т.е. каким образом следует соединить модули для обеспечения отказоустойчивости РМОС для класса неисправностей ПМ возможно большей кратности, если задано число m - функционирующих и число s - резервных модулей;

2. Какое число s - резервных модулей и какая исходная конфигурация необходима для обеспечения заданной отказоустойчивости РМОС, состоящей из m функционирующих модулей.

В настоящее время общего решения рассматриваемых задач не существует. Однако для РМОС с прямоугольной конфигурацией и размерностью $h \times c$ (рис. 4) в которой число столбцов « c » является четным числом и соединения внутри строк всегда выполняются слева направо, а соединения внутри столбцов в виде замкнутого кольца, можно найти детерминированную процедуру реконфигурации сети.

Утверждение 1. В сети с прямоугольной структурой размерностью $h \times c$ (рис. 4) число модулей n_{HV} в состоянии HV равно числу модулей n_{VH} в состоянии VH и имеет следующие границы:

$$\frac{(m-s)}{(h-1)} \leq n_{HV} = n_{VH} \leq c,$$

где $m(s)$ – число функционирующих (резервных) модулей.

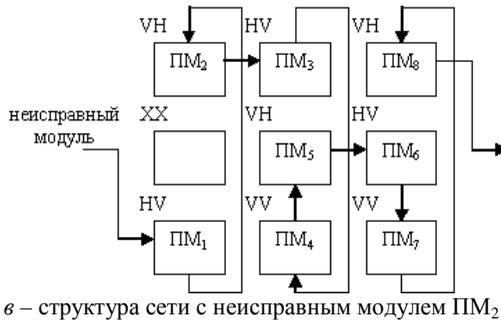
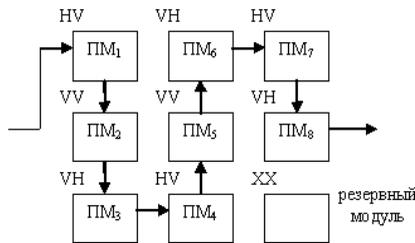
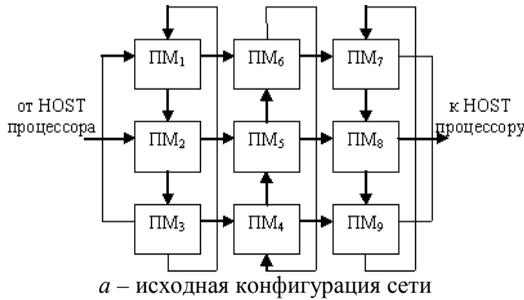


Рис. 3. Структура двумерной РМОС

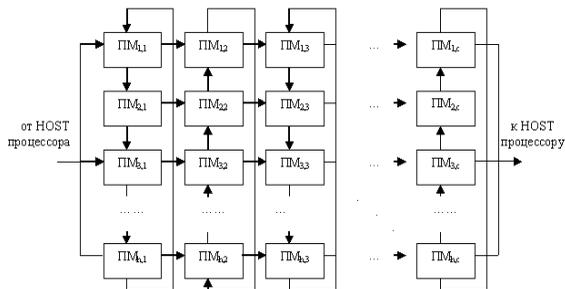


Рис. 4. Прямоугольная структура РМОС размерностью $h \times c$ (h – число строк, c – число столбцов)

Доказательство. Рабочей конфигурации соответствует цепочка, состоящая из m модулей, начинающаяся со входного и заканчивающаяся выходным модулем. Рассмотрим i -ый столбец. Маршрут данных проходит через некоторое число модулей этого столбца, и затем переходит к следующему столбцу. Если в столбце используется только один модуль, то последний находится в состоянии HH и такой столбец не оказывает влияния на величину чисел n_{HV} и n_{VH} . Если в столбце i используется несколько модулей, первый из них будет находиться в состоянии HV , последний в состоянии VH , а оставшиеся – в состоянии VV . Таким образом, i -ый столбец увеличивает на единицу как число n_{HV} , так и n_{VH} и верхняя граница определяется неравенством $n_{HV} = n_{VH} \leq c$. Для определения нижней границы очевидно, что маршрут данных длиной m модулей должен включать по два или более модулей, по меньшей мере из $(m - c) / (h - 1)$ столбцов \square .

Пример. Рассмотрим РМОС, изображенную на рис. 3, для которой $c = h = 3$ и $m = 8$. Следовательно, $n_{HV} = n_{VH} \leq 3$. Используя доказанное утверждение, можно определить также значения n_{HH} и n_{VV} :

$$n_{HH} \leq \frac{h \cdot c - m}{h - 1};$$

$$n_{VV} = m - n_{HH} - n_{HV} - n_{VH} = m - c - n_{HV}.$$

Таким образом,

$$m - 2 \cdot c \leq n_{VV} \leq \frac{(m - c) \cdot (h - 2)}{h - 1}.$$

Для сети, представленной на рис. 3, $n_{HH} = 0$ и $n_{VV} = 2n$. Полученные предельные значения состояний четырехпортовых коммутаторов использу-

ются при проверке корректности конфигурации системы в процессе реализации процедуры восстановления работоспособности сети.

Основным достоинством двумерной РМОС с распределенной реконфигурацией является то, что процедура реконфигурации не является частью ядра системы. Такая структура имеет следующие достоинства:

- 1) неисправность механизма коммутации равноценна неисправности модуля;
- 2) рабочая конфигурация поддерживается исключительно исправными модулями и, таким образом, не зависит от поведения неисправных модулей.

Следует отметить, что неисправный ПМ может воздействовать на выходы РМОС. Это можно устранить путем использования специального модуля выходного селектора, как показано на рис. 5. Основными недостатками такой структуры являются сложность алгоритма реконфигурации и тот факт, что схема с s резервными модулями будет устойчива к неисправностям, имеющим кратность, меньшую s .

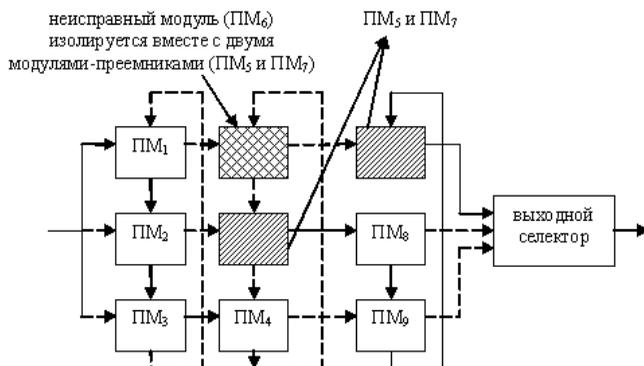


Рис. 5. Структура сети со схемой выходного селектора

Ниже приведен обобщенный алгоритм реконфигурации двумерной РМОС при отсутствии резервных модулей. В алгоритме используются следующие исходные данные: 1) $vx_вект$, $вых_вект$ – двоичные векторы входов и выходов РМОС и каждого исправного модуля; 2) $путь$ – вектор текущей конфигурации РМОС. Этот вектор определяет последовательность номеров ПМ функционирующих в сети; 3) $матрица$ -преемник – матрица размерностью $2 \times n$, где n – общее число модулей, в которой $вых_вект$ каждого исправного модуля отмечается «1», а наличие неисправного модуля приемника, либо отсутствие приемника отмечается «0».

Входными данными алгоритма реконфигурации РМОС является размерность сети и номер неисправного ПМ, значения векторов $vx_вект$,

вых_вект, *путь*, *матрицы* преемников, а также инициализация процесса пересылки данных.

Алгоритм реконфигурации РМОС можно представить в виде последовательности следующих шагов:

Шаг 1. Исключить неисправный модуль из векторов *вх_вект* и *вых_вект*.

Шаг 2. Модифицировать *матрицу* преемников путем изменения всех значений *вх_вект*, *вых_вект* каждого ПМ с учетом номера неисправного ПМ.

Шаг 3. Если хотя бы один из векторов *вх_вект* и *вых_вект* РМОС не принадлежит ни одному ПМ, выполнить функцию *отказ_РМОС* и завершить выполнение алгоритма, в противном случае перейти к шагу 4 алгоритма.

Шаг 4. Выполнить функцию *поиск_путей* для нахождения всех путей в РМОС при известном номере неисправного ПМ. Результат представить в виде матрицы, *все_пути*, в каждой строке которой описан отдельный путь.

Шаг 5. Удалить из матрицы *все_пути* те пути, в которых последний модуль не связан с выходом РМОС.

Шаг 6. Если матрица *все_пути* не содержит ни одного модуля, связанного со входами и выходом РМОС, выполнить функцию *отказ_системы* и завершить выполнение алгоритма, в противном случае перейти к шагу 7.

Шаг 7. Выбрать в матрице *все_пути* путь, которому соответствует наименьшее число пересылок данных и максимальное число функционирующих модулей.

Шаг 8. Выполнить функцию пересылки данных между модулями. В случае разрушения данных выполнить процедуру их восстановления путем повторного выполнения загрузочной программы

Шаг 9. Конец алгоритма.

Предложенный алгоритм рассмотрим на примере РМОС, состоящей из девяти ПМ (рис. 5). Пусть исходная конфигурация сети представлена маршрутом обработки данных в виде последовательности: (ПМ₁, ПМ₂, ПМ₃, ПМ₄, ПМ₆, ПМ₅, ПМ₈, ПМ₉, ПМ₇), которая определена установленной структурой соединений между ПМ сети. Предположим, что в процессе диагностирования исправности сети модуль ПМ₆ – неисправен. Тогда на шаге 2 алгоритма модифицируется матрица приемников РМОС, в которой модули преемников ПМ₅ и ПМ₇ неисправного модуля ПМ₆ исключаются из матрицы преемников, что в свою очередь исключает все пути обработки данных, содержащие пересылки (ПМ₆ → ПМ₅, ПМ₆ → ПМ₇). На шаге 7 алгоритма определяются две возможных конфигурации РМОС 1) ПМ₁, ПМ₂, ПМ₃, ПМ₄, ПМ₉, ПМ₇, ПМ₈; 2) ПМ₁, ПМ₂, ПМ₅, ПМ₄, ПМ₉, ПМ₇, ПМ₈. Анализ вариантов реконфигурации РМОС 2) ПМ₁, ПМ₂, ПМ₅, ПМ₄, ПМ₉, ПМ₇, ПМ₈.

Пуем сравнения вариантов реконфигурации РМОС с ее исходной конфигурацией определяется число «к» необходимых перезагрузок данных между ПМ сети, которые в каждом варианте отмечены сплошной линией. Анализ вариантов реконфигурации РМОС показывает, что при одинаковом числе

функционирующих модулей ($n = 7$), число перезагрузок данных между модулями реконфигурируемой РМОС в 1-ом варианте $\kappa = 2$, во 2-ом $\kappa = 3$. Следовательно, 1-ый вариант реконфигурации является более предпочтительным второму.

Выводы. Рассмотрены процедуры реконфигурации одномерных и двумерных РМОС с использованием четырехпортовых коммутаторов и распределенной реконфигурации структуры РМОС, содержащей резервные и функциональные модули. Определены границы предельных значений состояний коммутирующих модулей, обеспечивающих оптимальную структуру РМОС. Предложен алгоритм реконфигурации двумерной РМОС с выходным селектором данных без использования резервных модулей.

Список литературы: 1. *Аладьев В.З.* Однородные структуры / Аладьев В.З. – К.: Техника, 1990. – 272 с. 2. *Евреинов Э.В.* Однородные вычислительные системы, структуры и среды / Евреинов Э.В. – М.: Радио и связь, 1981. – 208 с. 3. *Прангшвили И.В.* Параллельные вычислительные системы с общим управлением / Прангшвили И.В., Виленкин С.Я., Медведев И.Л. – М.: Энергоатомиздат, 1983. – 312 с. 4. *Палагин А.В.* Реконфигурируемые вычислительные системы / А.В. Палагин, В.Н. Опанасенко. – К.: Просвіта, 2006. – 295 с. 5. *Kaklamanis C.* Asymptotically Tight Bounds for Computing with Faulty Arrays of Processors (extended abstract) / C. Kaklamanis, A.R. Karlin // Proc. IEEE Symp. Foundations of Computer Science. – 1990. – P. 285-296. 6. *Leighton F.T.* Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes. San Mateo, Calif.: Morgan-Kaufmann. – 1992. 7. *Leighton T.* 'On the Fault Tolerance of Some Popular Bounded-Degree Networks' / T. Leighton, B. Maggs, R. Sitaraman // Proc. IEEE Symp. Foundations of Computer Science. – 1992. – P. 542 – 552. 8. *Ajtai M., Fault N.* Tolerant Graphs, Perfect Hash Functions and Disjoint Paths / M. Ajtai, N. Alon // Proc. IEEE Symp. Foundations of Computer Science. – 1992. – P. 693 – 702. 9. *Alon N.* Explicit Construction of Linear Sized Tolerant Networks / N. Alon, F. Chung // Discrete Math. – 1988. – Vol. 72. – P. 15 – 19. 10. *Bruck J.* 'Wildcard Dimensions, Coding Theory and Fault-Tolerant Meshes and Hypercubes' / J. Bruck, R. Cypher, C. Ho // IEEE Trans. Computers – 1995. – Vol. 44. – P. 150 – 155. 11. *Rosenberg A.L.* 'The Diogenes Approach to Testable Fault-Tolerant Arrays of Processors' / A.L. Rosenberg // IEEE Trans. Computers. – 1983. – Vol. 32. – P. 480 – 489. 12. *Rosenberg A.L.* 'Fault-Tolerant Interconnection Networks: A Graph-Theoretic Approach' / A.L. Rosenberg // Proc. Ninth Workshop Graph-Theoretic Concepts in Computer Science. – 1983. – P. 286 – 297.

Поступила в редколлегию 02.04.10