

МЕТОДЫ РЕКОНФИГУРАЦИИ КРИТИЧЕСКИХ СИСТЕМ

Д.Б. Борзов

Юго-западный государственный университет
Россия, 305040, Курск, 50 лет Октября ул., 94
E-mail: borzovdb@kursknet.ru

К.А. Гуляев

Юго-западный государственный университет
Россия, 305040, Курск, 50 лет Октября ул., 94
E-mail: borzovdb@kursknet.ru

В.С. Титов

Юго-западный государственный университет
Россия, 305040, Курск, 50 лет Октября ул., 94
E-mail: borzovdb@kursknet.ru

Ключевые слова: планирование, размещение, перераспределение, критические системы, логическое управление, процессор, подпрограмма

Аннотация: В работе приведен анализ критических систем, таких как системы слежения, наблюдений, бортовой авиации и т.п. Показана необходимость оперативной реакции системы в случае возникновения внештатной ситуации, такой как отказ внутреннего процессорного модуля или модуля, построенного на базе ПЛИС. В этом случае необходима оперативная реакция системы, когда программные решения неприемлемы. Один из вариантов решения это перераспределения внутренних модулей вследствие изменения внутренней топологии системы. В работе предложены методы и аппаратно-ориентированные алгоритмы перераспределения и приведены соответствующие оценки временной сложности предложенных подходов.

1. Введение

Параллельная структурная организация – основной способ построения многозадачных систем логического управления (СЛУ), представляющих собой ансамбли множеств сверх больших интегральных схем (СБИС) многопроцессорных мультиконтроллеров или реконфигурируемых ПЛИС. Такой подход к организации позволяет повысить производительность СЛУ без увеличения тактовой частоты и систем нагрева СБИС, а также достичь отказоустойчивости и высокой готовности СЛУ [1]. При этом использование в качестве СЛУ критических систем (системы слежения, наблюдения, бортовой авиации и т.п.) накладывает дополнительные ограничения, связанные с дополнительным контролем за работоспособностью системы [2].

В этом случае важнейшим требованием к СЛУ является необходимость обеспечения высокой величины коэффициента готовности, приближающейся к единице [3]. В случае возникновения отказа функционального узла выполняется реконфигурация внутренней структуры, нарушается оптимальность первоначального размещения парал-

тельных процедур и может потребоваться повторное их перераспределение для уменьшения возросшей коммуникационной задержки и восстановления быстродействия СЛУ.

Многokратное перераспределение параллельных процедур необходимо также для сокращения быстродействия СЛУ ситуационного и адаптивного управления, так как в них многократно изменяются алгоритмы задач управления, решаемых множеством параллельных процессоров и/или ПЛИС. Наиболее высокой готовности и быстродействия таких СЛУ можно достичь, если построить их по архитектуре реконфигурируемых вычислительных систем на ПЛИС, позволяющих обеспечить максимальную скорость перезагрузки задач и существенно повысить быстродействие по сравнению с их решением мультипроцессорами. В каждом сеансе перезагрузки параллельно решаемых задач управления необходимо для уменьшения коммуникационных задержек выполнять планирование размещения параллельных процедур по их функциональным узлам [3,4].

Таким образом, актуальной является задача поиска методов и аппаратно-ориентированных алгоритмов планирования размещения и перераспределения подпрограмм в критических системах логического управления.

2. Метод реконфигурации в мультипроцессорных системах

Подмножество взаимодействующих подпрограмм описывается графом взаимодей-

ствия задач $G = \langle X, E \rangle$, где $X = \left\{ \begin{array}{cccccc} x_{1.1} & x_{1.2} & \dots & x_{1.k} & \dots & x_{1.n} \\ x_{2.1} & x_{2.2} & \dots & x_{2.k} & \dots & x_{2.n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_{q.1} & x_{q.1} & \dots & x_{q.k} & \dots & x_{q.n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_{n.1} & x_{n.2} & \dots & x_{n.k} & \dots & x_{n.n} \end{array} \right\}$ – множество вершин

графа G , вершины $x_i \in X$ которого соответствуют подпрограммам, а взвешенные дуги $e_{ij} \in E$ описывают связи между подпрограммами и задают объемы данных m_{ij} (в байтах) передаваемых при обмене между ними. Граф G представляется матрицей обмена информацией (МОИ): $M = \|m_{ij}\|$, где m_{ij} – вес дуги e_{ij} .

Мультипроцессорную систему будем описывать топологической моделью в виде

графа $H = \langle P, V \rangle$, где $P = \left\{ \begin{array}{cccc} p_{1,1} & p_{1,2} & \dots & p_{1,n} \\ p_{2,1} & p_{2,2} & \dots & p_{2,n} \\ \dots & \dots & \dots & \dots \\ p_{n,1} & p_{n,2} & \dots & p_{n,n} \end{array} \right\}$ – множество идентификаторов процес-

сорных модулей, организованных в матрицу $|P|_{n \times n}$, где мощность $|P| = N = n^2$ – число процессорных модулей базового блока; V – множество межмодульных связей, задаваемых матрицей смежности $\|W\|_{N \times N}$ размером $n^2 \times n^2$ [5-7].

Размещение пакета программ (подпрограмм), описываемых графом G в мультипроцессорной системе может быть аналитически описано отображением

$$(1) \quad \beta_s = \left\{ \begin{array}{cccccc} x_{S_{1,1}} & x_{S_{1,2}} & \dots & x_{S_{1,k}} & \dots & x_{S_{1,n}} \\ x_{S_{2,1}} & x_{S_{2,2}} & \dots & x_{S_{2,k}} & \dots & x_{S_{2,n}} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_{S_{q,1}} & x_{S_{q,2}} & \dots & x_{S_{q,k}} & \dots & x_{S_{q,n}} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_{S_{n,1}} & x_{S_{n,2}} & \dots & x_{S_{n,k}} & \dots & x_{S_{n,n}} \end{array} \right\} \rightarrow \left\{ \begin{array}{cccccc} p_{1,1} & p_{1,2} & \dots & p_{1,k} & \dots & p_{1,n} \\ p_{2,1} & p_{2,2} & \dots & p_{2,k} & \dots & p_{2,n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ p_{q,1} & p_{q,2} & \dots & p_{q,k} & \dots & p_{q,n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ p_{n,1} & p_{n,2} & \dots & p_{n,k} & \dots & p_{n,n} \end{array} \right\},$$

где $s = \overline{1, N!}$, $k = \overline{1, n}$, $q = \overline{1, n}$.

Здесь s – это номер очередной перестановки, соответствующий S -му варианту размещения. Мощность множества $\psi = \{\beta_s\}$ всевозможных отображений (1) равна числу всевозможных перестановок задач $\{x_{qk}\}$ в матрице X : $|\psi| = N!$.

Введем матрицу минимальных расстояний (ММР) $D = \|d_{ij}\|_{N \times N}$, $N = n^2 = |H|$. ММР строится по матрице смежности $\|W\|_{N \times N}$ графа H . Если непосредственной связи между модулями в базовом блоке нет, то элемент ММР: d_{ij} – минимальное межмодульное топологическое расстояние, которое вычисляется как кратчайший путь от вершины p_i до вершины p_j в графе H , измеренный числом последовательно соединенных каналов.

Пусть Ψ – множество всевозможных отображений вида (1). Тогда задачу размещения, можно сформулировать как поиск отображения $\beta^* \in \Psi$, такого, что

$$(2) \quad T_{\beta^*} = \min_{\Psi} \{ \max_{\beta_s \in \Psi} \{ T_{\beta_s}(p_{a,b}, p_{x,y}) \} \},$$

где $T_{\beta_s}(p_{a,b}, p_{x,y})$ – коммутационная задержка при передаче данных между модулями $p_{a,b}$ и $p_{x,y}$, соответствующая отображению β_s , которая рассчитывается как произведение

$$(3) \quad T_{\beta_s}(p_{a,b}, p_{x,y}) = d_{ij} \cdot m_{ij},$$

где $i = (a-1) \cdot n + b$ и $j = (x-1) \cdot n + y$.

При отказе процессора в критических системах происходит нарушение внутренней топологической организации системы и возникает необходимость оперативной перестройки внутренних связей. В связи с этим в [5-7] предложен метод и аппаратно-ориентированный алгоритм оперативного перераспределения подпрограмм.

В связи с вышесказанным в ходе поиска необходимо контролировать степень уменьшения величины образующихся коммуникационных задержек (3) и принимать решение о целесообразности продолжения поисковых перестановок строк и столбцов матрицы МОИ. Это может быть сделано в результате анализа величины отношения достигнутого значения задержки (2) к гипотетической минимально возможной ее величине.

Процедуру поиска минимальной нижней оценки можно представить, состоящим из следующих этапов:

1) Переписать элементы $d_{kl} \neq 0$ матрицы D в вектор-строку $D' = \|d_{kl}^z\|$ так, что

$$d_{k'l'}^{z_1} \leq d_{k'l'}^{z_2} \Leftrightarrow z_1 > z_2, \text{ где } z_1 \text{ и } z_2 \text{ – порядковые номера элементов в } D'.$$

2) Переписать элементы $m_{ij} \neq 0$ матрицы M в вектор-строку $M' = \|m_{ij}^z\|$ так, что

$$m_{i'j'}^{z_1} \geq m_{i'j'}^{z_2} \Leftrightarrow z_1 > z_2, \text{ где } z_1 \text{ и } z_2 \text{ – порядковые номера элементов в } M'.$$

3) Положить

$$(4) \quad T_{\inf} = \max \{ m^z d^z \},$$

где $z = \overline{1, |E|}$, m^z , d^z – одноименные элементы векторов M' и D' с одинаковыми порядковыми номерами.

При поиске варианта размещения по критерию (1) коммуникационную задержку при обработке пакета программ (подпрограмм) будем определять по формуле:

$$(5) \quad T = \max\{m_{i,j} \cdot d_{ij}\},$$

где $i, j = \overline{1, N}$.

Для ускорения поиска варианта размещения применяется метод [5], основанный на применении целенаправленных перестановок строк и столбцов матрицы МОИ с выбором в ней αk -го места перестановки ее элемента $m_{\alpha\beta}$ по критерию:

$$(6) \quad d_{\alpha k} < d_{\alpha\beta},$$

где $d_{\alpha k}, d_{\alpha\beta}$ – одноименные элементы матрицы ММР; $m_{\alpha\beta}$ – элемент МОИ, которому соответствует $\max\{m_{ij} \cdot d_{ij}\}$, найденный в предыдущем шаге перестановок.

Дополнительные критерии, используемые для выполнения текущего контроля целесообразности перестановок:

$$(7) \quad P > m_{\alpha i} \cdot d_{\alpha i}$$

$$(8) \quad P > m_{\alpha i} \cdot d_{\alpha\beta}$$

где произведение $P = m_{\alpha\beta} \cdot d_{\alpha\beta}$ соответствует оценке текущего достигнутого значения коммуникационной задержки.

Таким образом, оценка задержки P , которую дает текущий элемент МОИ $m_{\alpha\beta}$, находясь на расстоянии $d_{\alpha\beta}$ должна быть:

- а) больше той, что дает элемент $m_{\alpha i}$ на расстоянии $d_{\alpha i}$, на место которого целесообразно выполнить перестановку;
- б) больше той, что даст элемент МОИ $m_{\alpha i}$, после его перестановки в $\alpha\beta$ -ю позицию на расстояние $d_{\alpha\beta}$.

Тогда исходя из приведенных теоретических положений можно сформулировать процедуру размещения, состоящую из следующих шагов:

- а) Производится первичное произвольное размещение вершин графа G , представляющего исходный комплекс решаемых задач в базовом кластерном блоке. Размещение реализуется путем наложения матрицы МОИ $M = \|m_{ij}\|_{N \times N}$ на матрицу ММР $D = \|d_{ij}\|_{N \times N}$. По формуле (6) находится максимальную коммутационную задержку T_n соответствующую данному начальному варианту размещения.
- б) Для сопоставления вариантов размещения по критерию (1) вначале осуществляется поиск нижней оценки T_{inf} для графа G . Затем вычисляется степень близости максимальной коммутационной задержки T_n , соответствующей первичному варианту размещения, к нижней оценке T_{inf} в виде:

$$(9) \quad \eta_n = \frac{T_n}{T_{inf}},$$

- в) Начиная с элемента m_{ij} , которому соответствует $\max\{m_{i,j} \cdot d_{ij}\}$ в первичном варианте, пытаемся переместить столбец, его содержащий, на место другого столбца так, чтобы после перестановки и расчета T по формуле (7) оценка:

$$(10) \quad \eta = \frac{T}{T_{inf}}$$

снижалась по сравнению с η_n (9) и оценками η по предыдущим вариантам размещения. Место целенаправленной перестановки элемента m_{ij} выбирается по критерию:

$$(11) \quad d_{ik} < d_{ij},$$

где ij – координаты перемещаемого элемента МОИ m_{ij} ; d_{ij} – расстояние в ММР, с которым совпадает элемент МОИ m_{ij} ; d_{ik} – расстояние в ММР, куда целесообразно переместить элемент m_{ij} ; ik – координаты элемента m_{ik} на место которого перемещается в ходе перестановки элемент m_{ij} ; i – номер строки в МОИ и ММР в пределах которой выполняется перестановка элементов столбцов j и k ; j, k – номера переставляемых столбцов МОИ с обязательной последующей перестановкой соответствующих им строк.

д) Анализ достигнутой величины η (9) и оценка степени улучшения размещения выполняется по следующей формуле выигрыша в снижении коммуникационной задержки.

$$(12) \quad \sigma = \frac{\eta_n}{\eta} = \frac{T_n}{T}.$$

Вывод результатов перестановки в виде новой матрицы МОИ, размещенной в кластерном блоке с заданной матрицей ММР топологией.

Было проведено программное моделирование предложенного алгоритма с целью определение выигрыша в разгах σ относительно T_n в результате применения разработанного метода при разных видах и степенях заполнения МОИ, соответствующих широкому диапазону изменения степени связности между подпрограммами, запланированных к обработке. Анализировались результаты, соответствующие матрице наилучшего размещения M_k , начальному и достигнутому отклонению задержки T от T_{inf} . η_n и η соответственно, достигнутому выигрышу σ в разгах, а так же времени, затраченному на поиск. В результате были получены графики, представленные на рис. 1-4.

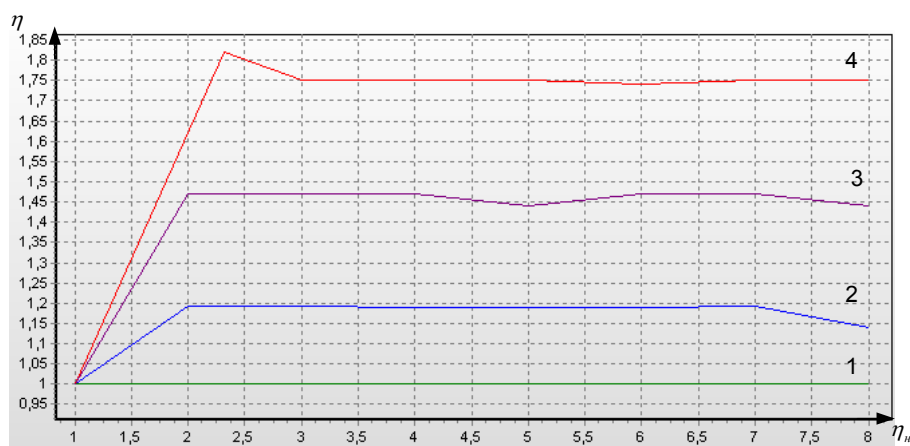


Рис. 1. Графики зависимости η от η_n .

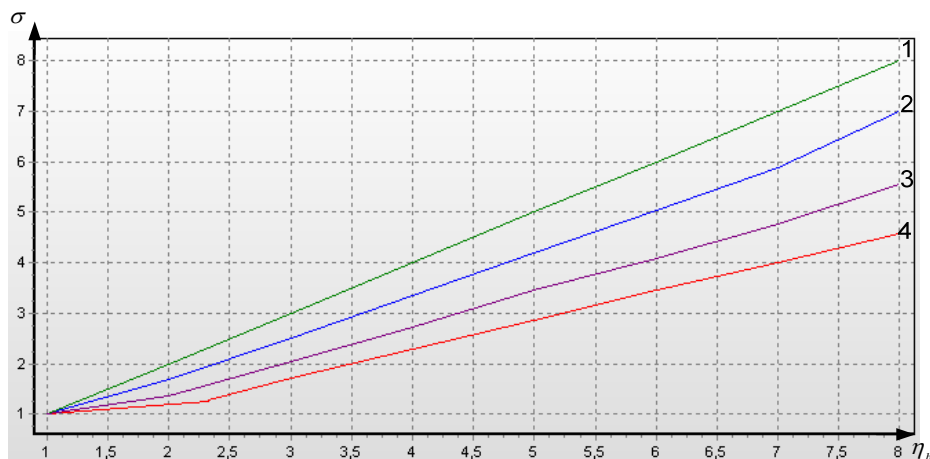


Рис. 2. Графики зависимости σ от η_n .

На рис. 2 приняты следующие обозначения:

- график 1 – матрица МОИ заполнена так, что отношение элементов $\frac{m_{\min}}{m_{\max}} = \frac{1}{10}$;
- график 2 – матрица МОИ заполнена так, что отношение элементов $\frac{m_{\min}}{m_{\max}} = \frac{1}{5}$;
- график 3 – матрица МОИ заполнена так, что отношение элементов $\frac{m_{\min}}{m_{\max}} = \frac{1}{4}$;
- график 4 – матрица МОИ заполнена так, что отношение элементов $\frac{m_{\min}}{m_{\max}} = \frac{1}{3}$.

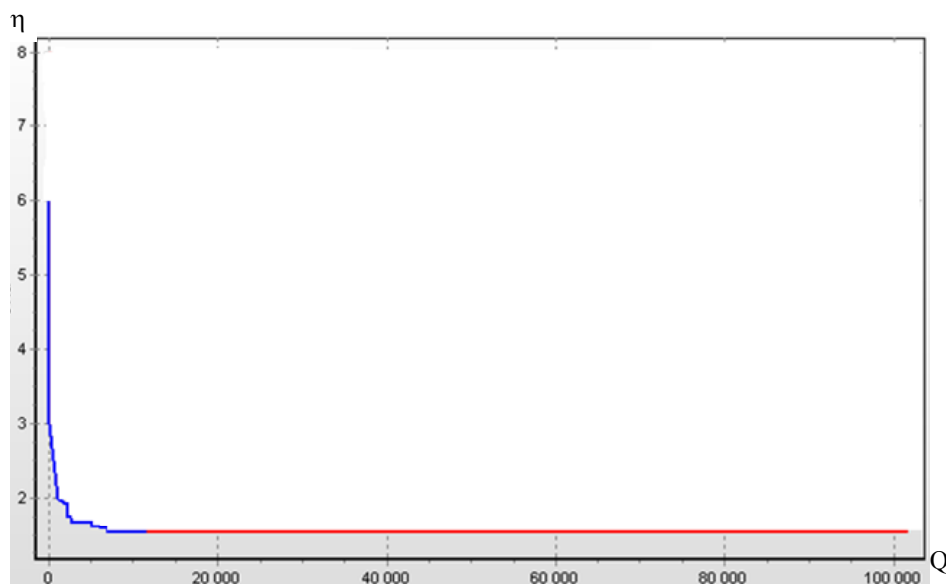


Рис. 3. График зависимости η от количества последовательных перестановок Q для $\eta_n = 6$ при перестановках по критерию б.

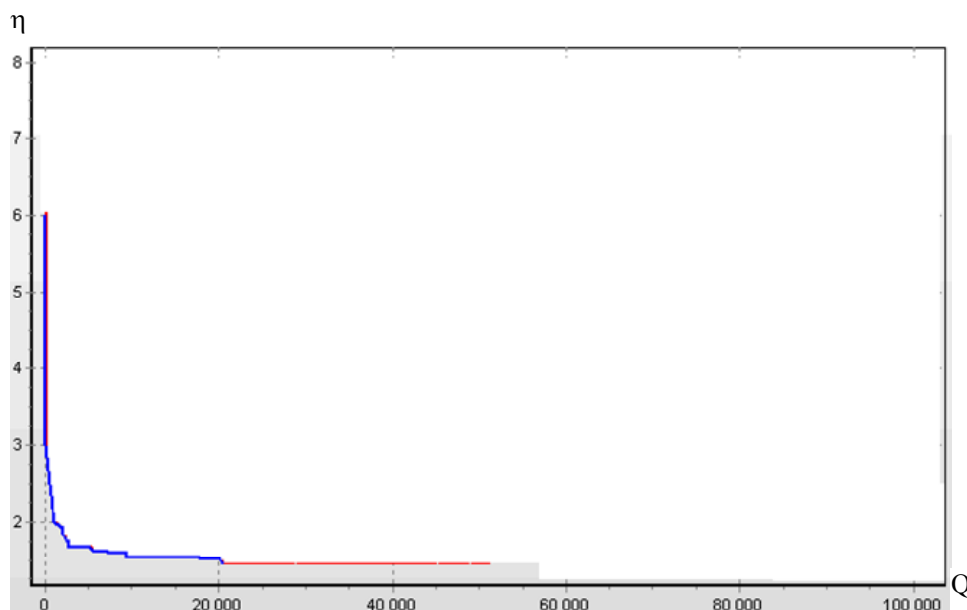


Рис. 4. График зависимости η от количества последовательных перестановок Q для $\eta_n = 6$ при перестановках по комплексу критериев 6, 7, 8.

Анализ полученных графиков (рис. 1-4) показывает следующее.

- а) При начальном заполнении МОИ объемами передаваемых между задачами данных, когда $\eta_n \leq 2$, выигрыш в снижении коммуникационной задержки составляет всего $\sigma = 1,3 \dots 2$ раза. Причем наибольший выигрыш может быть достигнут лишь в изначально наихудшем по разбросу объемов передаваемых данных случае заполнения МОИ, когда отношение элементов $\frac{m_{\min}}{m_{\max}} = \frac{1}{10}$, где m – элемент матрицы МОИ с размерностью $[m]$ =байт.
- б) Сходимость поиска можно значительно улучшить путем ввода дополнительных критериев (7), (8) контроля целесообразности перестановки по ускоренному алгоритму. Это позволяет избавиться от заведомо неудачных перестановок и сократить вдвое общее число требуемых перестановок Q .
- в) В связи с необходимостью выполнения по алгоритму избыточного количества перестановок, большая часть которых приходится на заключительную часть процедуры поиска, когда значение η перестает существенно снижаться, целесообразно вести порог η_n эффективности перестановок, который отсекал бы эти избыточные перестановки по неравенству $\eta \leq \eta_n$.

Как следует из зависимостей η от η_n при разных распределениях заполнения $\frac{m_{\min}}{m_{\max}}$ матрицы МОИ, величина порога η_n может быть выбрана в диапазоне $1.8 \leq \eta_n \leq 2$.

3. Метод реконфигурации в программируемых логических интегральных схемах

Программу (подпрограмму) будем представлять графом взаимодействия задач.

$$(13) \quad G = \langle X, E \rangle, \text{ где } X = \left\{ \begin{array}{cccccc} x_{1,1} & x_{1,2} & \dots & x_{1,k} & \dots & x_{1,n} \\ x_{2,1} & x_{2,2} & \dots & x_{2,k} & \dots & x_{2,n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_{q,1} & x_{q,2} & \dots & x_{q,k} & \dots & x_{q,n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_{n,1} & x_{n,2} & \dots & x_{n,k} & \dots & x_{n,n} \end{array} \right\}$$

– множество вершин графа G , вершины $x_{qk} \in X$ которого соответствуют подпрограммам, а дуги $e_{ij} \in E$ связям между ними, которые передаются между подпрограммами и сведены в матрицу смежности (МС) $M = \|m_{ij}\|_{N \times |E|}$, где $N = |X|$.

Топология ПЛИС задана графом H , который представлен как [8]:

$$H = \left\{ \begin{array}{cccccc} P_{1,1} & P_{1,2} & \dots & P_{\varpi, \theta} & \dots & P_{1,n} \\ P_{2,1} & P_{2,2} & \dots & P_{\varpi, \theta} & & P_{2,n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ P_{m,1} & P_{m,2} & \dots & P_{\varpi, \theta} & \dots & P_{m,n} \end{array} \right\},$$

где $P_{\varpi, \theta}$ – отдельные модули ПЛИС, причем ($\varpi = \overline{1, m}$, $\theta = \overline{1, n}$).

Модуль $P_{\varpi, \theta}$ представляется в виду функции $F(O, X)$, т.е.

$$(14) \quad P_{\varpi, \theta} = F(O, X),$$

где $O = o_1, o_2, \dots, o_\xi$ – множество входных модуля, а $X = x_1, x_2, \dots, x_\xi$ – множество выходов. Выводы модулей ПЛИС соединяется с выводами других модулей. При этом общее количество сигналов, которые передаются в ПЛИС от одного модуля к другому, определяют итоговую коммуникационную задержку, которую необходимо минимизировать для увеличения производительности ПЛИС. Схематично модуль ПЛИС может быть представлен так, как показано на рис. 5.

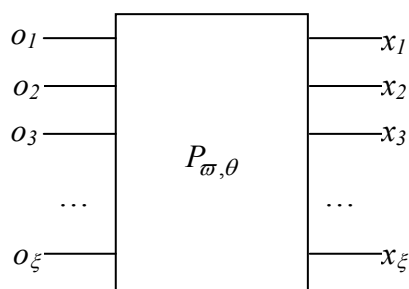


Рис. 5. Модуль ПЛИС.

Как видно из рис. 5, величина ξ – количество входов и/или выходов, представленная в (14) зависит от конкретной схемной реализации модуля ПЛИС и неизвестна заранее.

Под межсоединением в модуле ПЛИС будем понимать проложенную трассу от одного вывода o_i или x_j ($i = \overline{1, \xi}$, $j = \overline{1, \xi}$) модуля к другому. На рис. 6 проиллюстрировано понятие межсоединения.

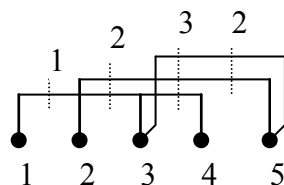


Рис. 6. Межсоединения в модуле ПЛИС.

На рис. 6 точками обозначены выходы фиктивных модулей ПЛИС. Числа над пунктирными линиями обозначают степени загрузки межсоединений между парами смежных контактов.

Матрицей цепей (МЦ) будем называть, описывающая вариант размещения модулей ПЛИС прямоугольная матрица $V = |v_{i,j}|_{n,\alpha}$, где $i = \overline{1, N}$, $j = \overline{1, \alpha}$, $n = |X|$, α представляет собой суммарное количество межсоединений, полученных в результате размещения подпрограмм в модулях ПЛИС.

Мощность множества $|P|$ в конечном итоге зависит от числа межсоединений, полученных в результате размещения модулей в ПЛИС. Параметр α соответствует количеству межсоединений и заранее неизвестен. Показано, что должно выполняться соотношение [9]:

$$(15) \quad \alpha \rightarrow \min$$

Тогда размещение модулей в ПЛИС может быть описано отображением:

$$(16) \quad \beta_s = \left\{ \begin{array}{cccccc} x_{S_{1,1}} & x_{S_{1,2}} & \dots & x_{S_{1,k}} & \dots & x_{S_{1,\alpha}} \\ x_{S_{2,1}} & x_{S_{2,2}} & \dots & x_{S_{2,k}} & \dots & x_{S_{2,\alpha}} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_{S_{q,1}} & x_{S_{q,1}} & \dots & x_{S_{q,k}} & \dots & x_{S_{q,\alpha}} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_{S_{n,1}} & x_{S_{n,2}} & \dots & x_{S_{n,k}} & \dots & x_{S_{n,\alpha}} \end{array} \right\} \rightarrow \left\{ \begin{array}{cccccc} p_{1,1} & p_{1,2} & \dots & p_{1,k} & \dots & p_{1,\alpha} \\ p_{2,1} & p_{2,2} & \dots & p_{2,k} & \dots & p_{2,\alpha} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ p_{q,1} & p_{q,2} & \dots & p_{q,k} & \dots & p_{q,\alpha} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ p_{n,1} & p_{n,2} & \dots & p_{n,k} & \dots & p_{n,\alpha} \end{array} \right\} \triangleright,$$

где $S = \overline{1, N!}$. В (16) символ \rightarrow означает отображение одной из вершин $x_{S_{q,k}} \in X$ на один из модулей $p_{q,k} \in H$. Здесь s – это номер очередной перестановки, соответствующий S -му варианту размещения. Мощность множества $\psi = \{\beta_s\}$ всевозможных отображений (16) равна числу всевозможных перестановок подпрограмм $x_{qk} \in X$ в матрице M : $|\psi| = N!$.

Пусть Ψ – множество всевозможных отображений вида (16). Тогда задачу размещения, можно сформулировать как поиск отображения $\beta^* \in \Psi$, такого, что

$$(17) \quad T_{\beta^*} = \min_{\Psi} \{ \max_{\beta_s \in \Psi} \{ T_{\beta_s} (|V|_{p_{q,k}}) \} \},$$

где $T_{\beta_s} (|V|_{p_{q,k}})$ – задержка при передаче данных в модуле $p_{q,k}$, соответствующая отображению β_s . В выражении (17) $\max_{\beta_s \in \Psi}$ означает поиск максимальной задержки при передаче данных в процессоре $p_{q,k}$, где $q = \overline{1, n}$, $k = \overline{1, \alpha}$; выражение \min_{Ψ} соответствует

поиску минимально возможного значения задержки для максимального

$$\max_{\beta_s \in \Psi} \left\{ T_{\beta_s} \left(|V|_{p,q,k} \right) \right\}.$$

Под смежным контактом (СК) в ПЛИС будем понимать такое расположение выводов модулей ПЛИС, при котором выполняется условие:

$$(18) \quad \theta = |v_{i,j} - v_{i,j-1}| = 1.$$

При этом справедливо условие при поиске варианта размещения $|\Psi| = N!$ модулей в ПЛИС:

$$(19) \quad \sum_{\Delta} \theta_{\Delta} \rightarrow \max,$$

где $\Delta = \overline{1, n}$.

Для уменьшения общей длины межсоединений проводников ПЛИС желательно, чтобы наиболее интенсивно взаимодействующие модули ПЛИС располагались в непосредственной близости друг от друга и тем самым уменьшали общую длину межсоединений.

Поиск варианта размещения является трудоемкой переборной задачей. Решение может быть принято в результате анализа величины отношения достигнутого минимаксного значения (17) к минимально возможной ее величине, алгоритм вычисления которого представлен ниже:

а) Переписать элементы $m_{kl} \neq 0$ ($k = \overline{1, N}$, $l = \overline{1, N}$) матрицы $M = \|m_{ij}\|_{N \times |E|}$ в вектор-строку $M' = \|m_{kl}^z\|$, где z – это порядковый номер элемента в M' , причем $k = \overline{1, N}$, $l = \overline{1, N}$. При этом $z_1 > z_2$ и z_1, z_2 – порядковые номера элементов в M' .

б) Положить

$$(20) \quad T_{\inf} = \sum_{i=1}^{j=|E|} m_i^z,$$

где $i = \overline{1, |E|}$, $z = \overline{1, |E|}$, $|E|$ – мощность множества E в графе G ; m^z – одноименные элементы вектора M' .

Для решения задачи планирования размещения в ПЛИС предлагается процедура, в которой условие оптимальности размещения является степень отклонения η минимаксного значения условия (17) от нижней оценки T_{\inf} (20).

Процедура поиска решения включает следующие этапы [10]:

1) Производится первичное произвольное размещение вершин графа G , представляющего исходный комплекс решаемых в ПЛИС подпрограмм, описанных графом H . Размещение реализуется путем наложения матрицы МС $M = \|m_{ij}\|_{N \times |E|}$ на матрицу

МЦ $V = |v_{i,j}|_{n,\alpha}$. По формуле (20) находим максимальную задержку T_n , соответствующую данному начальному варианту размещения.

2) Для сопоставления вариантов размещения по условию (17) вначале осуществляется поиск нижней оценки T_{\inf} для графа G по алгоритму поиска нижней оценки. Затем вычисляется степень близости максимальной задержки T_n , соответствующей первичному варианту размещения, к нижней оценке T_{\inf} в виде:

$$(21) \quad \eta_n = \frac{T_n}{T_{\inf}}.$$

- 3) Начиная с элемента $m_{ij} \in H$, которому соответствует $\max\{\sum v_j\}$ в первичном варианте, пытаемся переместить строку, его содержащую, на место другой строки так, чтобы после перестановки и расчета T по формуле (21) оценка:

$$(22) \quad \eta = \frac{T}{T_{\text{inf}}}$$

снижалась по сравнению с η_n (21) и оценками η по предыдущим вариантам размещения.

- 4) Анализ достигнутой величины η (22) и оценка степени улучшения размещения выполняется по следующей формуле выигрыша в снижении коммуникационной задержки.

$$(23) \quad \sigma = \frac{\eta_n}{\eta} = \frac{T_n}{T}.$$

Проведено программное моделирование разработанного метода и алгоритма разработан пакет программ, в результате чего были получены зависимости, результаты которых были сведены в таблицах 1 и 2.

Таблица 1. Требуемое количество перестановок Q при варьировании размеров матрицы цепей.

V	5	9	10	20	25	50	60
ψ	5!	9!	10!	20!	25!	50!	60!
Q	5	16	21	51	66	141	171

Таблица 2. Результаты моделирования методики и алгоритма размещения.

V	9×9	10×10	10×15	15×15	20×20	20×25	25×25	50×50	60×60
T_{inf}	33	53	67	73	204	243	317	1239	1832
T_n	18	35	36	45	109	127	163	450	585
T	23	40	49	54	125	142	179	461	620
$t, \text{ мкс}$	73	110	155	635	754	828	839	13323	16677
Q	16	20	21	19	51	51	66	141	171
ψ	9!	10!	10!	15!	20!	25!	25!	50!	60!
$\sigma = \frac{T_n}{T}$	1.2	1.14	1.3	1.2	1.14	1.11	1.09	1.0	1.0
$\eta = \frac{T}{T_{\text{inf}}}$	1.43	1.3	1.36	1.35	1.63	1.71	1.77	2.68	2.95
$\eta_n = \frac{T_n}{T_{\text{inf}}}$	1.8	1.5	1.0	1.6	1.8	1.9	1.9	2.75	3.8

В таблице 1 представлены данные для разных размеров МЦ. В строках показаны данные для вариантов полного перебора всех возможных перестановок ψ и их количества при применении предложенного метода Q . Из анализа данных таблицы 1 можно сделать вывод, что при малом количестве внутренних модулей ПЛИС (5-10), количество перестановок для поиска варианта размещения не большое. Однако в случае применения большего количества контактов, например 20 или 25, при полном переборе требуется 20! и 25! перестановок. Одновременно при применении предложенного метода необходимо соответственно 51 и 66 перестановок.

На основе данных таблиц 1, 2 можно сделать следующие выводы. При малых размерах МЦ степень удаления T от T_n незначительно. С увеличением размеров МЦ (таб-

лица 2) разница значений между T и T_n составляет около 20. Это свидетельствует о целесообразности применения предложенного метода при выполнении «больших» вычислительных задач с размерами МЦ порядка 1000×1000 , так как в критических системах, как правило, встречаются именно такие задачи.

4. Заключение

В статье приведен один из подходов для увеличения коэффициента готовности критических систем, когда выход из строя внутреннего модуля может привести к катастрофе. Предложены методы и аппаратно-ориентированные алгоритмы перераспределения подпрограмм в многопроцессорных системах и СЛУ, построенных на базе ПЛИС. В работе приведены оценки и временные показатели скорости работы предложенных алгоритмов, подтверждающих перспективность их применения в системах будущих поколений, в которых размерность используемых процессорных модулей будет порядка тысячи. Предложенные алгоритмы целесообразно использовать в виде аппаратных средств, встроенных в СЛУ и выполняющих роль помощника в случае возникновения критической ситуации.

Работа выполнена в рамках гранта президента РФ № НШ-2357.2014.8

Список литературы

1. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. СПб.: БХВ-Петербург. 2002. 608 с.
2. Чечкин А.В., Пирогов М.В. Метод интеллектуализации критических систем с использованием таблиц радикалов // Нейрокомпьютеры: разработка, применение. 2012. № 2. С. 3-12.
3. Борзов Д.Б., Мараят Б.И., Типикин А.П. Акселератор планирования размещения задач в кластерных вычислительных системах высокой готовности // Известия вузов. Приборостроение. 2008. № 2. С. 29-33.
4. Борзов Д.Б., Зотов И.В., Минайлов В.В. Акселераторные средства составления плана топологии программируемых логических интегральных схем // Известия Юго-Западного государственного университета. Серия. Управление, вычислительная техника, информатика. Медицинское приборостроение. 2012. № 2. Ч. 2. С. 95-98.
5. Борзов Д.Б., Мараят Б.И., Масолов А.С. Метод снижения коммуникационной задержки путем субоптимального размещения задач в матричных базовых блоках кластера // Телекоммуникации. 2008. № 4. С. 21-25.
6. Борзов Д.Б., Мараят Б.И., Типикин А.П. Алгоритмы и принцип организации аппаратных средств ускорения составления плана размещения задач в кластерных мультимикомпьютерах / Деп. в ВИНТИ 25.10.07. № 998-В 2007.
7. Борзов Д.Б., Мараят Б.И. Методика планирования размещения задач в матрично-торроидальных базовых блоках кластерных мультимикомпьютеров / Деп. в ВИНТИ 18.07.06. № 961-В 2006.
8. Борзов Д.Б. Методика минимизации длины межмодульных связей в ПЛИС // Машиностроение и техносфера XXI века: сборник трудов XVIII МНТК. Курск: ДонНТУ, 2011. Т. 2. С. 86-89.
9. Борзов Д.Б. Метод и алгоритм планирования топологии ПЛИС в многопроцессорных системах // Известия Юго-Западного государственного университета. Серия. Управление, вычислительная техника, информатика. Медицинское приборостроение. 2012. № 2. Часть 2. С. 23-27.
10. Борзов Д.Б., Минайлов В.В. Метод и алгоритм планирования топологии ПЛИС в многопроцессорных системах // Известия Юго-Западного государственного университета. Серия. Управление, вычислительная техника, информатика. Медицинское приборостроение. 2012. № 2. Часть 2. С. 23-27.