

Э.И. Ватутин (аспирант),
И.В. Зотов (д-р техн. наук, профессор)¹

Курский государственный технический университет (*кафедра вычислительной техники*)

Повышение качества разбиения алгоритмов при синтезе логических мультиконтроллеров с использованием метода параллельно-последовательной декомпозиции

Одним из перспективных подходов к созданию бортовых систем логического управления является их реализация в базисе логических мультиконтроллеров [1]. Подобные системы способны выполнять алгоритмы управления теоретически неограниченной сложности (выражаемой количеством вершин алгоритма, его параллельных ветвей, степенью вложенности базовых фрагментов и т.д.) за счет их разбиения (декомпозиции) на последовательные фрагменты (блоки) фиксированной сложности в соответствии с ограничениями, налагаемыми элементной базой [2]. Задача выбора разбиения имеет ярко выраженный комбинаторный характер, что не позволяет получить оптимальное решение за приемлемое время. В связи с этим при декомпозиции алгоритмов управления реальной размерности используется ряд эвристических методов [3-5], существенно отличающихся по качеству получаемых решений и временным затратам на его достижение. Проведенные авторами исследования показывают, что среди рассматриваемых методов наиболее перспективным является параллельно-последовательный метод [1-3], обеспечивающий более высокое качество решений при сохранении приемлемых для практики временных затрат. Целью данной работы является развитие указанного метода в направлении дальнейшего повышения качества синтезируемых разбиений.

Качество разбиения алгоритма управления определяется степенью минимизации отдельных критериев, к которым относятся: количество блоков, степень дублирования логических условий и микроопераций, интенсивность межблочных взаимодействий и сложность сети межблочных связей. Интегральный критерий качества, отражающий все перечисленные выше компоненты, может быть представлен в следующем виде [6]:

$$f\left(\text{Sep}_{B_k}\left(A_l^0\right)\right)=\frac{K_H}{\omega_{\max}} H+\frac{K_X}{\left|X\left(A_l^0\right)\right|}\left(\sum_{i=1}^H\left|X\left(A_i\right)\right|-\left|X\left(A_l^0\right)\right|\right)+\frac{K_Y}{\left|Y\left(A_l^0\right)\right|}\left(\sum_{i=1}^H\left|Y\left(A_i\right)\right|-\left|Y\left(A_l^0\right)\right|\right)+\frac{K_{\delta}}{\delta\left(A_l^0\right)} \sum_{i=1}^{H-1} \sum_{j=i+1}^H \delta\left(A_i, A_j\right)+\frac{K_{\alpha}}{\omega_{\max}\left(\omega_{\max}-1\right)} \sum_{i=1}^{H-1} \sum_{j=i+1}^H \alpha\left(A_i, A_j\right)+K_W \Delta W,$$

где H – количество блоков; A_l^0 – исходное множество всех вершин l -го алгоритма выборки, подлежащих распределению по блокам; A_i – i -ый блок разбиения, $i=\overline{1, H}$; $X\left(A_i\right)$ и $Y\left(A_i\right)$ – множества логических условий и микроопераций i -го блока; $\delta\left(A_i, A_j\right)$ – интенсивность межблочного взаимодействия между блоками A_i к A_j ; $\alpha\left(A_i, A_j\right)$ – функция связности блоков (0 – блоки не связаны по управлению, 1 – связаны); ω_{\max} – степень параллелизма алгоритма управления (минимальное теоретическое количество блоков); $\text{Sep}_{B_k}\left(A_l^0\right)$ – разбиение алгоритма A_l^0 , полученное с использованием метода B_k . При этом параметр $\Delta W=\max _{i=\overline{1, H}} W\left(A_i\right)-\min _{i=\overline{1, H}} W\left(A_i\right)$, оценивающий разность алгоритмов по сложности и по предварительным оценкам мешающий минимизации более важных параметров, решено исключить из рассмотрения путем задания весового коэффициента $K_W=0$. Значения остальных весовых коэффициентов не претерпели

¹ Исследования И.В.Зотова выполнены при поддержке гранта Президента РФ для молодых ученых МК-3073.2007.8.

изменений по сравнению с предложенными в [6]: $K_H = 0,6$; $K_X = 0,4$; $K_Y = 0,3$; $K_\alpha = 0,6$; $K_\delta = 0,6$. Для более объективного выявления влияния предлагаемых в работе улучшений на результирующее качество разбиений сравнение качества решений будем проводить на выборке из K алгоритмов со случайной структурой и распределением микроопераций, логических условий и весов [7]. При этом в качестве критериев сравнения будем рассматривать как средневывборочные значения перечисленных выше критериев

$$\begin{aligned}\bar{\gamma}(H(B_i)) &= \frac{\sum_{j=1}^K H(Sep_{B_i}(A_j^0))}{K}, \\ \bar{\gamma}(X(B_i)) &= \frac{\sum_{j=1}^K \gamma(X(Sep_{B_i}(A_j^0)))}{K} = \frac{\sum_{j=1}^K \left(\sum_{k=1}^{H(Sep_{B_i}(A_j^0))} |X(A_k \in Sep_{B_i}(A_j^0))| - |X(A_j^0)| \right)}{K}, \\ \bar{\gamma}(Y(B_i)) &= \frac{\sum_{j=1}^K \gamma(Y(Sep_{B_i}(A_j^0)))}{K} = \frac{\sum_{j=1}^K \left(\sum_{k=1}^{H_i(Sep_{B_i}(A_j^0))} |Y(A_k \in Sep_{B_i}(A_j^0))| - |Y(A_j^0)| \right)}{K}, \\ \bar{\gamma}(\alpha(B_i)) &= \frac{\sum_{j=1}^K \gamma(\alpha(Sep_{B_i}(A_j^0)))}{K} = \frac{\sum_{j=1}^K \left(\sum_{k=1}^{H(Sep_{B_i}(A_j^0))} \sum_{l=k+1}^{H(Sep_{B_i}(A_j^0))} \alpha(A_k \in Sep(A_j^0), A_l \in Sep(A_j^0)) \right)}{K}, \\ \bar{\gamma}(\delta(B_i)) &= \frac{\sum_{j=1}^K \gamma(\delta(Sep_{B_i}(A_j^0)))}{K} = \frac{\sum_{j=1}^K \left(\sum_{k=1}^{H(Sep_{B_i}(A_j^0))} \sum_{l=k+1}^{H(Sep_{B_i}(A_j^0))} \delta(A_k \in Sep(A_j^0), A_l \in Sep(A_j^0)) \right)}{K}, \\ \bar{\gamma}(f(B_i)) &= \frac{\sum_{j=1}^K f(Sep(A_j^0))}{K},\end{aligned}$$

где под $A_k \in Sep_{B_i}(A_j^0)$ понимается k -й блок разбиения j -го алгоритма выборки, полученного с использованием метода B_i , так и вероятности

$$\rho(z(Sep_{B_i}(A^0))) = \frac{\sum_{j=1}^K \mu(z(Sep_{B_i}(A^0)))}{K} \cdot 100\%$$

получения разбиения с минимальным значением выбранного параметра $z = \{H, X, Y, \alpha, \delta, f\}$, определяемые как отношение количества разбиений с минимальным значением параметра z (полученных методом B_i , среди разбиений того же алгоритма A^0 , полученных другими методами), к общему количеству алгоритмов в выборке. Здесь

$$\mu(z(Sep_{B_i}(A^0))) = \begin{cases} 1, & z(Sep_{B_i}(A^0)) = \min_{j=1, B_N} z(Sep_{B_j}(A^0)) \\ 0, & \text{иначе} \end{cases} \quad \text{— функция определения близости к}$$

минимуму значения выбранного параметра z в разбиении $Sep_{B_i}(A^0)$ среди других имеющихся разбиений алгоритма A^0 .

Очевидно, что метод, для которого $\bar{\gamma}(z) \rightarrow \min$ и $\rho(z) \rightarrow \max$ для всех критериев z (либо для некоторого подмножества критериев при сохранении значений остальных) будет считаться наиболее предпочтительным.

Для иллюстрации приведенных выше формул рассмотрим пример (табл. 1), в котором построены разбиения выборки из двух алгоритмов тремя методами.

Таблица 1. Построения разбиений тремя методами на выборке из двух алгоритмов

Выборка алгоритмов	Метод 1	Метод 2	Метод 3
Алгоритм 1	$H = 14$ $\gamma(X) = 0$ $\gamma(Y) = 26$ $\gamma(\alpha) = 28$ $\gamma(\delta) = 30,00$ $f = 1,86$	$H = 14$ $\gamma(X) = 1$ $\gamma(Y) = 28$ $\gamma(\alpha) = 24$ $\gamma(\delta) = 29,00$ $f = 1,92$	$H = 14$ $\gamma(X) = 1$ $\gamma(Y) = 31$ $\gamma(\alpha) = 28$ $\gamma(\delta) = 30,00$ $f = 1,99$
Алгоритм 2	$H = 4$ $\gamma(X) = 0$ $\gamma(Y) = 10$ $\gamma(\alpha) = 6$ $\gamma(\delta) = 11,32$ $f = 1,69$	$H = 4$ $\gamma(X) = 2$ $\gamma(Y) = 8$ $\gamma(\alpha) = 6$ $\gamma(\delta) = 11,58$ $f = 1,73$	$H = 5$ $\gamma(X) = 4$ $\gamma(Y) = 14$ $\gamma(\alpha) = 8$ $\gamma(\delta) = 11,99$ $f = 2,25$
Средние значения	$\bar{\gamma}(H) = \frac{14+4}{2} = 9$ $\bar{\gamma}(X) = \frac{0+0}{2} = 0$ $\bar{\gamma}(Y) = \frac{26+10}{2} = 18$ $\bar{\gamma}(\alpha) = \frac{28+6}{2} = 17$ $\bar{\gamma}(\delta) = \frac{30,00+11,32}{2} = 20,66$ $\bar{\gamma}(f) = \frac{1,86+1,69}{2} = 1,78$	$\bar{\gamma}(H) = \frac{14+4}{2} = 9$ $\bar{\gamma}(X) = \frac{1+2}{2} = 1,5$ $\bar{\gamma}(Y) = \frac{28+8}{2} = 18$ $\bar{\gamma}(\alpha) = \frac{24+6}{2} = 15$ $\bar{\gamma}(\delta) = \frac{29,00+11,58}{2} = 20,29$ $\bar{\gamma}(f) = \frac{1,92+1,73}{2} = 1,83$	$\bar{\gamma}(H) = \frac{14+5}{2} = 9,5$ $\bar{\gamma}(X) = \frac{1+4}{2} = 2,5$ $\bar{\gamma}(Y) = \frac{31+14}{2} = 22,5$ $\bar{\gamma}(\alpha) = \frac{28+8}{2} = 18$ $\bar{\gamma}(\delta) = \frac{30,00+11,99}{2} = 21,00$ $\bar{\gamma}(f) = \frac{1,99+2,25}{2} = 2,12$
Вероятности	$\rho(H) = \frac{1+1}{2} \cdot 100\% = 100\%$ $\rho(X) = \frac{1+1}{2} \cdot 100\% = 100\%$ $\rho(Y) = \frac{1+0}{2} \cdot 100\% = 50\%$ $\rho(\alpha) = \frac{0+1}{2} \cdot 100\% = 50\%$ $\rho(\delta) = \frac{0+1}{2} \cdot 100\% = 50\%$ $\rho(f) = \frac{1+1}{2} \cdot 100\% = 100\%$	$\rho(H) = \frac{1+1}{2} \cdot 100\% = 100\%$ $\rho(X) = \frac{0+0}{2} \cdot 100\% = 0\%$ $\rho(Y) = \frac{0+1}{2} \cdot 100\% = 50\%$ $\rho(\alpha) = \frac{1+1}{2} \cdot 100\% = 100\%$ $\rho(\delta) = \frac{1+0}{2} \cdot 100\% = 50\%$ $\rho(f) = \frac{0+0}{2} \cdot 100\% = 0\%$	$\rho(H) = \frac{1+0}{2} \cdot 100\% = 50\%$ $\rho(X) = \frac{0+0}{2} \cdot 100\% = 0\%$ $\rho(Y) = \frac{0+0}{2} \cdot 100\% = 0\%$ $\rho(\alpha) = \frac{0+0}{2} \cdot 100\% = 0\%$ $\rho(\delta) = \frac{0+0}{2} \cdot 100\% = 0\%$ $\rho(f) = \frac{0+0}{2} \cdot 100\% = 0\%$

Замечание. Сумма вероятностей получения минимального значения выбранного критерия $\sum_{i=1}^{B_N} \rho(z \in Sep_{B_i}(A_j^0))$ вполне может быть больше 100%, что не является ошибкой, т.к. минимальное значение выбранного критерия z может быть получено одновременно несколькими методами.

Приведенный пример является весьма условным ввиду очень малого объема выборки и приведен исключительно для демонстрации процесса вычисления значений критериев. Реальные сравнения необходимо проводить на выборках большего размера, ограничиваемого продолжительностью вычислительного эксперимента и затратами памяти. Влияние предлагаемых ниже улучшений параллельно-последовательного метода на качество разбиений оценивалось на выборке из 1000 сгенерированных алгоритмов со случайной структурой (средний размер алгоритма – 100 вершин) с использованием разработанной программной системы РАЕ [8]. Длительность вычислительного эксперимента для выявления влияния каждого улучшения составляло величину порядка 15–20 мин.

Оценку влияния каждого из улучшений будем проводить следующим образом: рассматривать версию метода со всеми предлагаемыми улучшениями и сравнивать качество его разбиений с тем же методом, но без одного выбранного улучшения.

Улучшение 1 – дополнительный параметр весовой функции, обеспечивающий совместную группировку условных вершин в блоках. Необходимость улучшения рассмотрим на следующем примере (рис. 1).

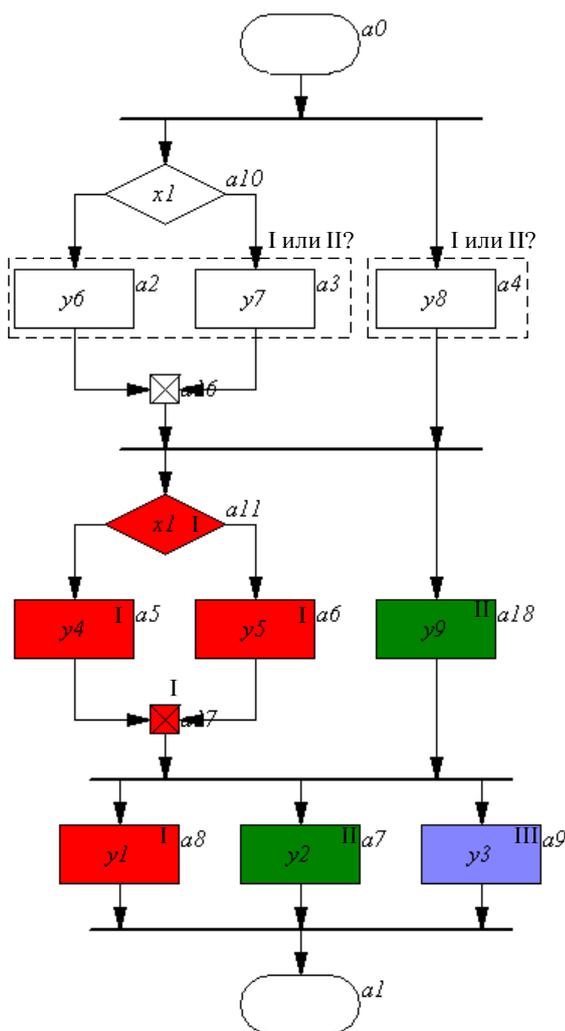


Рис. 1. Пример алгоритма логического управления

В приведенном примере базовое сечение [1-3], с которого начинается построение блоков разбиения, образовано вершинами $\{a_7, a_8, a_9\}$. На некотором i -м шаге работы алгоритма синтеза блоков разбиения (при движении «вверх») сформированы блоки $A_1 = \{a_5, a_6, a_8, a_{11}, a_{17}\}$, $A_2 = \{a_7, a_8\}$, $A_3 = \{a_9\}$ и имеется два субсечения $\rho_1 = \{a_2, a_3\}$, $\rho_2 = \{a_4\}$. Оба субсечения одинаково «плохо» включать в блок A_3 , т.к. при этом появляются избыточные дуги передачи управления (что ведет к увеличению сложности сети связей и интенсивности межблочных взаимодействий), и одинаково «хорошо» включать в блоки A_1 и A_2 . При более глубоком рассмотрении видно, что на самом деле субсечение ρ_1 лучше включить в блок A_1 , т.к. следующая вершина-кандидат на распределение a_{10} имеет в своем составе логическое условие x_1 , уже имеющееся в блоке A_1 . Однако без дополнительных проверок значение весовой функции для включений $\begin{bmatrix} \rho_1 \rightarrow A_1 \\ \rho_2 \rightarrow A_2 \end{bmatrix}$ и $\begin{bmatrix} \rho_1 \rightarrow A_2 \\ \rho_2 \rightarrow A_1 \end{bmatrix}$ одинаково, что на практике примерно в половине подобных случаев приводит к неоптимальному распределению условной вершины. С ростом размерности алгоритмов управления дублирование логических условий все более увеличивается, т.к. приведенная в примере ситуация начинает встречаться чаще.

Улучшение заключается в том, что к весовой функции, используемой при построении блоков разбиения [1-3]

$$t(\rho_i, A_j) = \begin{cases} \text{"-"}, \exists a_m \in \rho_i, a_n \in A_j : a_m \omega a_n \\ \text{"+"}, (W(\rho_i) + W(A_j) > W_{\max}) \vee (|X(\rho_i) \cup X(A_j)| > n_{IV}) \\ K_1^Y |Y(\rho_i) \cap Y(A_j)| - K_2^Y |Y(\rho_i) \setminus Y(A_j)| + \\ + K_1^X |X(\rho_i) \cap X(A_j)| - K_2^X |X(\rho_i) \setminus X(A_j)| - \\ - [K_1^Z \Delta Z_1 + K_2^Z \Delta Z_2 + K_W \Delta W] \text{ в противном случае} \end{cases}$$

добавлен параметр $K_\Psi \cdot \xi(\rho_i) \cdot |\Psi(A_j)|$, где K_Ψ – весовой коэффициент, $\Psi(A) \subseteq A$ – подмножество альтернативных вершин множества A , $\xi(\rho_i)$ – функция наличия в субсечении альтернативных вершин, такая что

$$\xi(\rho_i) = \begin{cases} 1, \Psi(\rho_i) \neq \emptyset, \\ 0, \text{ иначе.} \end{cases}$$

Влияние предлагаемого улучшения на качество решений отражено в табл. 2 (наиболее значимые изменения значений параметров выделены серым).

Таблица 2. Влияние улучшения 1 на качество разбиений

Метод	Критерии качества										
	$\bar{\gamma}(H)$	$\rho(H)$	$\bar{\gamma}(X)$	$\rho(X)$	$\bar{\gamma}(Y)$	$\rho(Y)$	$\bar{\gamma}(\alpha)$	$\rho(\alpha)$	$\bar{\gamma}(\delta)$	$\rho(\delta)$	$t, \text{ мс}$
С улучшением	14,836	99,9%	7,440	92,6%	49,615	84,9%	39,866	81,2%	37,988	80,2%	448,1
Без улучшения	14,835	100,0%	8,055	63,6%	49,998	73,0%	39,807	83,6%	38,017	74,8%	448,3

Из приведенных в таблице данных следует, что предлагаемое улучшение позволяет уменьшить среднее значение степени дублирования логических условий (с 8,055 до 7,440) и увеличить вероятность получения разбиений с меньшей степенью дублирования логических условий (с 63,6% до 92,6%) при практически неизменной степени приближения к оптимальному для остальных критериев и сохранении аналогичного (в пределах погрешности) времени

нахождения разбиений. Следует отметить, что введенное улучшение также увеличивает вероятность получения меньшей степени дублирования микроопераций (с 73,0% до 84,9%) и вероятность получения меньшей интенсивности межблочных взаимодействий (с 74,8% до 80,2%), хотя напрямую не влияет на эти параметры.

Улучшение 2 – разбиение субсечений на отдельные вершины при попытке образовании нового блока. Рассмотрим пример алгоритма управления (рис. 2).

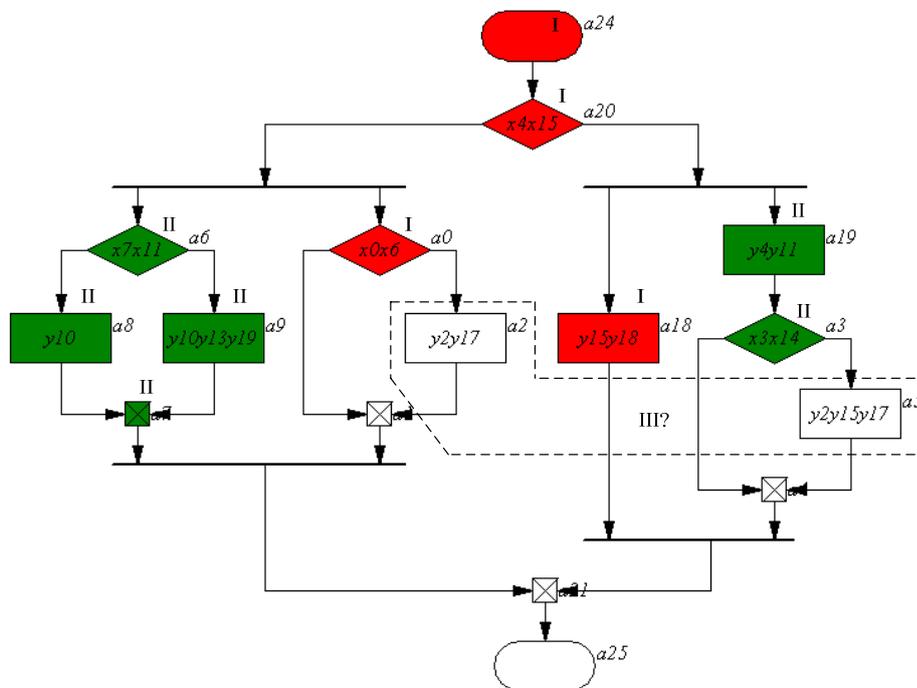


Рис. 2. Пример алгоритма управления

На i -м шаге работы алгоритма построения блоков разбиения сформирована пара блоков $A_1 = \{a_0, a_{18}, a_{20}, a_{24}\}$ и $A_2 = \{a_3, a_6, a_7, a_8, a_9, a_{19}\}$. Необходимо разместить в блоках сечение $\{a_2, a_5\}$, которое представляется одним субсечением $(a_2 \psi a_5)$. В первый блок данное субсечение включить нельзя из-за нарушения ограничения параллельности $(a_5 \omega a_{18})$, во второй блок данное сечение также не может попасть из-за нарушения ограничения параллельности $(a_2 \omega a_6)$: сечение образует новый («лишний») блок (с увеличением размерности алгоритма вероятность появления данного конфликта возрастает). Несложно заметить, что если рассмотреть сечение в виде двух субсечений $\rho_1 = \{a_2\}$ и $\rho_2 = \{a_5\}$, содержащих по одной вершине, то будут возможны включения $\begin{bmatrix} \rho_1 \rightarrow A_1 \\ \rho_2 \rightarrow A_2 \end{bmatrix}$ и образования лишнего блока не произойдет.

Рассмотренное улучшение можно формализовать следующим образом: если рассматриваемое субсечение состоит более чем из одной вершины ($|\rho_i| > 1$) и его невозможно включить ни в один из существующих блоков из-за нарушения отношения параллельности $(\forall A_j : \rho_i \omega A_j)$, то необходимо представить субсечение в виде множества субсечений $\rho'_i, \rho'_{i_2}, \dots, \rho'_{i_L}$, $L = |\rho_i|$, каждое из которых содержит одну вершину $(\forall \rho'_{i_k} : |\rho'_{i_k}| = 1)$, заполнить для них таблицу включений и произвести их распределение по блокам.

Влияние предлагаемого улучшения на качество решений отражено в табл. 3.

Таблица 3. Влияние улучшения 2 на качество разбиений

Метод	Критерии качества										
	$\bar{\gamma}(H)$	$\rho(H)$	$\bar{\gamma}(X)$	$\rho(X)$	$\bar{\gamma}(Y)$	$\rho(Y)$	$\bar{\gamma}(\alpha)$	$\rho(\alpha)$	$\bar{\gamma}(\delta)$	$\rho(\delta)$	$t, \text{мс}$
С улучшением	14,836	100,0%	7,440	99,0%	49,615	99,6%	39,866	100,0%	37,988	99,6%	613,3
Без улучшения	14,941	93,7%	7,469	98,0%	49,788	94,7%	40,266	93,7%	38,046	94,1%	615,0

Анализ таблицы показывает, что предложенное улучшение увеличивает вероятность построения разбиения с минимальным количеством блоков с 93,7% до 100%, при этом все остальные критерии также улучшаются. Это в очередной раз доказывает, что количество блоков является наиболее важным критерием, косвенно влияющим на все остальные критерии.

В продолжении улучшения 2 возникает идея о представлении субсечений в виде отдельных вершин всегда, а не только в случае получения «лишнего» блока. Однако в таком случае (табл. 4) увеличивается среднее количество блоков, что косвенно влияет и на ухудшение остальных критериев, поэтому данную идею следует считать неоправданной.

Таблица 4. Влияние на качество разбиений генерации субсечения из отдельных вершин

Метод	Критерии качества										
	$\bar{\gamma}(H)$	$\rho(H)$	$\bar{\gamma}(X)$	$\rho(X)$	$\bar{\gamma}(Y)$	$\rho(Y)$	$\bar{\gamma}(\alpha)$	$\rho(\alpha)$	$\bar{\gamma}(\delta)$	$\rho(\delta)$	$t, \text{мс}$
Субсечения по отношению параллельности вершин	14,836	100,0%	7,440	76,9%	49,615	87,2%	39,866	97,8%	37,988	94,7%	411,8
Субсечения в виде отдельных вершины	18,290	9,0%	7,628	70,5%	53,596	21,3%	47,993	5,9%	40,592	5,4%	534,0

Улучшение 3 – «интеллектуальное» разбиение сечения на субсечения с использованием весовой функции. В описании параллельно-последовательного метода, приводимом в [1-3], каждое из сечений разбивается на субсечения таким образом, чтобы в составе каждого из получаемых субсечений не было параллельных вершин. Подобное разбиение неоднозначно: сечение можно разбить на субсечения более чем одним способом. Вызывает интерес поиск такого способа, который обеспечивал бы меньшее дублирование микроопераций и логических условий за счет более оптимальной группировки вершин в субсечениях (в предыдущей реализации метода в качестве множества субсечений принималось любое из распределений). Для этого модифицируем алгоритм разбиения сечения на субсечения следующим образом: для каждой новой нераспределенной вершины сечения будем вычислять значение функции

$$g(a_i, \rho_j) = \begin{cases} "- ", a_i \cap \rho_j, \\ K_1^X \cdot |X(\rho_j) \cap X(a_i)| - K_2^X \cdot |X(\rho_j) \setminus X(a_i)| + \\ + K_1^Y \cdot |Y(\rho_j) \cap Y(a_i)| - K_2^Y \cdot |Y(\rho_j) \setminus Y(a_i)| \text{ иначе} \end{cases}$$

и помещать вершину в то субсечение, для которого значение функции максимально. Результат применения улучшения приведен в табл. 5.

Таблица 5. Влияние улучшения 3 на качество разбиений

Метод	Критерии качества										
	$\bar{\gamma}(H)$	$\rho(H)$	$\bar{\gamma}(X)$	$\rho(X)$	$\bar{\gamma}(Y)$	$\rho(Y)$	$\bar{\gamma}(\alpha)$	$\rho(\alpha)$	$\bar{\gamma}(\delta)$	$\rho(\delta)$	$t, \text{мс}$
С улучшением	14,836	99,6%	7,440	90,0%	49,615	85,9%	39,866	79,3%	37,988	65,2%	412,3
Без улучшения	14,832	100,0%	7,415	91,9%	49,992	70,1%	39,655	88,0%	37,798	86,9%	418,5

Анализ приведенных в таблице данных показывает, что улучшение весьма неоднозначно: на фоне увеличения вероятности получения меньшего дублирования микроопераций (с 70,1% до 85,9%) наблюдается ухудшение остальных характеристик (наиболее сильно ухудшаются вероятности получения минимальной сложности сети связей – с 88,0% до 79,3% и вероятность получения минимальной интенсивности межблочных взаимодействий – с 86,9% до 65,2%). При наложении ограничений в целом ситуация не изменяется.

Также вызывает интерес анализ влияния порядка рассмотрения сечений на качество разбиений. В исходном описании метода [1-3] предлагается начинать рассмотрение сечений с базового сечения, обладающего максимальной ω -мощностью, двигаясь по сечениям «вверх» и «вниз» поочередно (вверх, вниз, вверх, вниз, ...). В качестве альтернативы такому порядку рассмотрения сечений введем еще два способа: сперва преимущественное движение от базового сечения «вверх», а затем «вниз» (вверх, вверх, ..., вверх, вниз, вниз, ..., вниз), и случайный порядок рассмотрения сечений, начиная с базового (в первую очередь по блокам распределяется базовое сечения, остальные сечения распределяются в случайном порядке). Вычислительный эксперимент проведем для случая без ограничений (табл. 6) и для случая сильного ограничения $W_{\max} = 10$ (табл. 7).

Таблица 6. Влияние порядка рассмотрения сечений на качество разбиений (без ограничений)

Метод	Критерии качества										
	$\bar{\gamma}(H)$	$\rho(H)$	$\bar{\gamma}(X)$	$\rho(X)$	$\bar{\gamma}(Y)$	$\rho(Y)$	$\bar{\gamma}(\alpha)$	$\rho(\alpha)$	$\bar{\gamma}(\delta)$	$\rho(\delta)$	t , мс
Поочередная смена направлений	14,834	99,9%	7,583	53,2%	49,899	44,3%	40,478	45,1%	38,184	51,5%	450,2
Преимущественные направления	14,836	99,7%	7,440	59,7%	49,615	51,9%	39,866	62,3%	37,988	63,2%	436,1
Случайный порядок	14,836	99,7%	7,767	49,6%	50,684	34,9%	42,091	23,9%	39,585	13,4%	433,0

Таблица 7. Влияние порядка рассмотрения сечений на качество разбиений ($W_{\max} = 10$)

Метод	Критерии качества										
	$\bar{\gamma}(H)$	$\rho(H)$	$\bar{\gamma}(X)$	$\rho(X)$	$\bar{\gamma}(Y)$	$\rho(Y)$	$\bar{\gamma}(\alpha)$	$\rho(\alpha)$	$\bar{\gamma}(\delta)$	$\rho(\delta)$	t , мс
Поочередная смена направлений	15,099	93,4%	10,432	45,0%	54,753	46,6%	44,527	54,1%	39,798	51,0%	440,0
Преимущественные направления	15,310	74,9%	10,290	51,2%	55,559	30,6%	44,612	50,3%	39,760	53,0%	430,8
Случайный порядок	15,385	67,1%	10,182	55,8%	54,865	47,1%	47,193	19,9%	42,307	8,8%	432,7

Прежде всего следует отметить, что случайный порядок обработки сечений равномерно ухудшает все критерии как при отсутствии, так и при наличии ограничений (исключение составляет только незначительное улучшение распределения логических условий и микроопераций при наличии ограничений). В отсутствие ограничений лучше по всем параметрам себя проявляет порядок просмотра сечений с преимущественными направлениями. При появлении ограничений ситуация меняется: поочередная смена направлений при рассмотрении сечений приводит к увеличению вероятности получения минимального количества блоков (с 74,9% до 93,4%) при незначительном ухудшении распределения логических условий и незначительном улучшении распределения микроопераций и сложности сети связей.

Также следует отметить, что кроме перечисленных выше наиболее значительных усовершенствований было улучшено поведение метода при возникновении конфликта по включению и при построении разбиений в «предельном» случае (когда на протяжении нескольких

итераций количество субсечений в точности равно количеству блоков и любая ошибка ведет к образованию нового блока, что крайне нежелательно). Как отражено в таблицах 2–7, все положительно влияющие на качество разбиений улучшения не оказывают существенного влияния на время построения разбиений (увеличение временных затрат составляет не более 3%), что в совокупности с интегральным улучшением качества разбиений подтверждает их целесообразность.

Библиографический список

1. Архитектура и синтез параллельных логических мультимикроконтроллеров: Учебное пособие: в 2 ч. Ч.1 / Зотов И.В. и др.; Курск: Изд-во КурскГТУ, 2006. 200 с.
2. Зотов И.В., Колосков В.А., Титов В.С. Выбор оптимальных разбиений алгоритмов при проектировании микроконтроллерных сетей // Автоматика и вычислительная техника. 1997. № 5. С. 51–62.
3. Ватутин Э.И., Зотов И.В. Метод формирования субоптимальных разбиений параллельных управляющих алгоритмов // Параллельные вычисления и задачи управления (РАСО'04). М.: Институт проблем управления им. В.А. Трапезникова РАН. 2004. С. 884–917.
4. Баранов С.И., Журавина Л.Н., Песчанский В.А. Метод представления параллельных граф-схем алгоритмов совокупностями последовательных граф-схем // Автоматика и вычислительная техника. 1984. № 5. С. 74–81.
5. Закревский А.Д. Параллельные алгоритмы логического управления. Минск.: ИТК НАН Б. 1999. 202 с.
6. Ватутин Э.И. Оценка качества разбиений параллельных управляющих алгоритмов на последовательные подалгоритмы с использованием весовой функции // Материалы межрегиональной научно-технической конференции «Интеллектуальные и информационные системы» (Интеллект-2005). – Тула, 2005. С. 29–30.
7. Vatutin E.I. Constructing Random Sample Parallel Logic Control Algorithms // 11th International Student Olympiad on Automatic Control (Baltic Olympiad BOAC'06). Saint-Petersburg, 2006. PP. 162–166.
8. Ватутин Э.И., Зотов И.В. Программная система для построения разбиений параллельных управляющих алгоритмов // Идентификация систем и задачи управления (SICPRO'06). М.: Институт проблем управления им. В.А. Трапезникова РАН, 2006. С. 2239–2250.