

УДК 681.324.001+681.326

Д.Б. Борзов, Э.И. Ватутин, И.В. Зотов, В.С. Титов
Курский государственный технический университет

ПРОЦЕДУРА И УСТРОЙСТВО ДЛЯ СУБОПТИМАЛЬНОГО РАЗБИЕНИЯ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ

Рассмотрена задача поиска субоптимальных разбиений параллельных алгоритмов логического управления. Описана аппаратно-ориентированная процедура разбиения. Приведено описание устройства (акселератора) на ее основе. Даны результаты оценки сложности и быстродействия устройства.

Одной из важных задач синтеза систем логического управления (СЛУ) является разбиение управляющего алгоритма на подалгоритмы с целью их последующего распределения между процессорами (модулями) [1]. Существующие подходы к ее решению [2–4] разработаны в расчете на программную реализацию, которая не способна обеспечить достаточную для практики степень приближения разбиения к оптимальному за ограниченное время. Однако, такое требование характерно для многих СЛУ, особенно, функционирующих в условиях оперативного изменения состава реализуемых алгоритмов управления. Преодолению указанной проблемы может способствовать перенос решения задачи разбиения на аппаратный уровень. Применительно к ряду аналогичных задач эта идея уже нашла воплощение в соответствующих устройствах-акселераторах [5]. Однако, задача выбора разбиения алгоритмов пока не решалась ни аппаратными средствами, ни на аппаратно-программном уровне.

В настоящей работе предпринята попытка решить задачу выбора субоптимального разбиения на аппаратном уровне. В основе решения предлагается использовать процедуру разбиения последовательного характера, подобную описанной в [6].

Формализованное представление задачи имеет следующий вид.

Пусть H – число подалгоритмов разбиения; W – емкость памяти модуля, выделенной для хранения команд подалгоритма; \bar{X}, \bar{Y} – допустимое число логических условий (ЛУ) и микроопераций (МО) подалгоритма соответственно. Требуется выбрать разбиение $Sep(A) = \{A_1, A_2, \dots, A_H\}$ множества вершин A исходного алгоритма управления, удовлетворяющее следующим критериям оптимальности:

$$\begin{cases} H \rightarrow \min; \\ Z^1 = \sum_{k=1}^{H-1} \sum_{l=k+1}^H \alpha(A_k, A_l) \rightarrow \min; \\ Z^2 = \sum_{k=1}^{H-1} \sum_{l=k+1}^H \delta(A_k, A_l) \rightarrow \min; \\ Z^3 = \max_{t, h=1, H, t \neq h} \{ |E(A_t) - E(A_h)| \} \rightarrow \min, \end{cases}$$

где Z^1 – сложность сети связей подалгоритмов для разбиения $Sep(A)$; $\alpha(A_k, A_l)$ – коэффициент связи подалгоритмов: $\alpha(A_k, A_l) = 1$, если подалгоритмы A_k и A_l связаны, т.е. $\exists a_i \in A_k, a_j \in A_l : a_i \varphi a_j$, где φ – отношение связи [2], $\alpha(A_k, A_l) = 0$ иначе; Z^2 – суммарное число взаимодействий подалгоритмов при заданной сети связей; $\delta(A_k, A_l)$ – интенсивность взаимодействия подалгоритмов A_k и A_l , т.е. число взаимодействий подалгоритма A_k с подалгоритмом A_l ; Z^3 – признак различия подалгоритмов по сложности; $E(A_f)$ – оценка сложности подалгоритма $A_f, f=t, h$.

Разбиение $Sep(A)$ должно соответствовать ограничениям:

$$\bigcup_{h=1}^H A_h = A, A_h \neq \emptyset, A_h \cap A_f = \emptyset, h, f = \overline{1, H}, h \neq f; \quad (1)$$

$$\neg(a_k \omega a_l) \quad \forall a_k, a_l \in A_h, k \neq l, h = \overline{1, H}; \quad (2)$$

$$W(A_h) \leq W, |X(A_h)| \leq \bar{X}, |Y(A_h)| \leq \bar{Y}, h = \overline{1, H}, \quad (3)$$

где $W(A_h)$ – число команд h -го подалгоритма: $X(A_h) = \bigcup_{a_k \in A_h} X(a_k)$ – множество ЛУ

h -го подалгоритма; $Y(A_h) = \bigcup_{a_k \in A_h} Y(a_k)$ – множество МО h -го подалгоритма; ω –

символ отношения параллельности [2].

Предлагаемая аппаратно-ориентированная процедура разбиения включает следующие шаги.

1. Положить $H = 1$, $A_1 = \{a_0\}$, $A^* = A \setminus \{a_0\}$, где a_0 – начальная вершина алгоритма управления, A^* – множество не распределенных вершин.

2. Найти подмножество вершин – кандидатов на включение $A' = \{a_i | (a_i \in A^*) \wedge (a_i \nu a_j, a_j \notin A^*)\}$, где ν – отношение следования [2].

3. Рассчитать для каждой вершины $a_i \in A'$ и каждого подалгоритма $A_k, k = \overline{1, H}$, значение критерия оптимальности распределения:

$$F(a_i, A_k) = \begin{cases} K_1 |Y(a_i) \cup Y(A_k)| + K_2 |X(a_i) \cup X(A_k)| + \\ + K_3 \alpha(a_i, A_k), \text{ если } \pi(a_i, A_k) = 1; \\ \infty, \text{ если } \pi(a_i, A_k) = 0, \end{cases} \quad (4)$$

где $\alpha(a_i, A_k)$ – коэффициент связи i -й вершины с вершинами k -го подалгоритма; K_1, K_2, K_3 – коэффициенты значимости; $\pi(a_i, A_k)$ – предикат, определяющий допустимость включения i -й вершины в k -й подалгоритм: $\pi(a_i, A_k) = 1$, если $\forall a_q \in A_k : \neg(a_q \omega a_i)$, при этом выполняются ограничения (3); $\pi(a_i, A_k) = 0$, если $\exists a_q \in A_k : a_q \omega a_i$ или нарушено хотя бы одно из ограничений (3).

4. Если $\exists a_j \in A', \forall A_l : F(a_j, A_l) = \infty$, то для такой вершины ввести новый подалгоритм A_{H+1} , положить $H = H + 1$ и $F(a_j, A_H) = 0$.

5. Повторять п.4, пока $\exists a_j \in A', \forall A_l : F(a_j, A_l) = \infty$.

6. Распределить вершины $a_i \in A'$ в формируемые подалгоритмы в соответствии с минимумом значений $F(a_i, A_k)$, рассчитанных в пп.3, 4.

7. Пересчитать параметры подалгоритмов $Y(A_k), X(A_k)$ согласно выбранному в п.6 распределению вершин.

8. Положить $A^* = A^* \setminus A'$. Если $A^* = \emptyset$, то конец, иначе идти к п.2.

Структурная схема устройства-акселератора, реализующего описанную процедуру, показана на рис.1. Здесь ЭМАУ – электронная модель алгоритма управления; БПВ – блок перебора вершин; БРВ – блок распределения вершин; БС – блок синхронизации. ЭМАУ хранит структуру разбиваемого алгоритма, для чего в нее введены N идентичных электронных моделей вершин (ЭМВ). Также в ЭМАУ содержатся списки МО и ЛУ и ограничения (3). БПВ выполняет перебор вершин, которые будут кандидатами на включение в подалгоритм согласно п.2 процедуры разбиения. БРВ служит для оценки выбранного подмножества вершин и распределения его по подалгоритмам. Оценка выполняется на основе критерия (4) по его аппаратному аналогу. После оценки БРВ передает в ЭМАУ данные о вычисленном распределении вершин. Цепи взаимодействия ЭМАУ и БПВ построены так, что БПВ игнорирует уже распределенные вершины, а среди нераспределенных выбирает вершины с учетом отношений между ними.

Структурная схема ЭМВ представлена на рис.2. В нее входит регистр микроопераций и логических условий (РгМОЛУ), регистр маски связей (РгМС), регистр маски распределения вершин (РгМРВ), регистр маски параллельности вершин (РгМПВ), триггер вида вершины (ТВВ), буферный блок (ББ), логические элементы. Регистр РгМОЛУ содержит унитарный код МО или ЛУ, соответствующих текущей вершине. Содержимое данного регистра интерпретируется в зависимости от состояния ТВВ. Регистр РгМРВ хранит двоичный код номера подалгоритма, в который распределена текущая вершина. Регистр РгМС хранит унитарный код всех вершин, которые непосредственно следуют за текущей вершиной. Регистр РгМПВ содержит унитарный код вершин, параллельных текущей вершине. Триггер ТВВ индицирует вид текущей вершины (операторная или условная).

На рис.3 изображена функциональная схема БПВ. В ней имеется две группы блоков элементов И ($1.1-1.N$ и $2.1-2.N$, где $N \equiv |A|$ – число вершин алгоритма управления), группа элементов ИЛИ, а также элемент ИЛИ-НЕ. Блоки элементов И $1.1-1.N$ формируют набор двоичных векторов, которые кодируют множество

всех непосредственных «последователей» i -й вершины, еще не распределенных по подалгоритмам. Блоки элементов И 2.1-2. N вырабатывают набор двоичных векторов, кодирующих множество всех непосредственных «последователей» уже распределенной i -й вершины, которые сами еще не распределены по подалгоритмам. Элементы ИЛИ формируют код группы выбранных по п.2 процедуры разбиения вершин – кандидатов на включение. Элемент ИЛИ-НЕ генерирует признак наличия вершин-кандидатов. Его значение 0, если имеется хотя бы один кандидат, и 1 при отсутствии кандидатов (признак завершения разбиения или ошибки в алгоритме).

На рис.4 показана структурная схема БРВ. Блок перебора кандидатов (БПК) обеспечивает цикл последовательного перебора вершин – кандидатов на включение, выбранных в БПВ. Цикл перебора подалгоритмов реализует схема перебора подалгоритмов (СПП), содержащая кольцевой счетчик с варьируемым коэффициентом пересчета. Блок распределения микроопераций (БРМ) решает три подзадачи. Во-первых, он хранит распределение МО, во-вторых, вычисляет новое множество МО подалгоритма с учетом выбранной вершины-кандидата, и в-третьих, проверяет выполнение ограничений (3) по числу МО в подалгоритме. Блок распределения логических условий (БРЛУ) по структуре аналогичен БРМ. Он хранит распределение ЛУ, вычисляет новое множество ЛУ подалгоритма с учетом выбранной вершины-кандидата, а также проверяет выполнение ограничений (3) по числу ЛУ в подалгоритме. Оценка связности вершин-кандидатов с подалгоритмами разбиения выполняется блоком оценки связности (БОС) на основе данных из РгМС, РгМПВ, РгМРВ (рис.2). Этот же блок запрещает включение в подалгоритм вершин, для которых нарушается ограничение (2) на непараллельность. Блок выбора кандидатов (БВК) рассчитывает критерий (4) по значениям, полученным из БОС, БРЛУ и БРМ, с учетом ограничений и запоминает номер наилучшего кандидата.

Время разбиения алгоритма предлагаемым устройством (без учета процессов записи информации об алгоритме и считывания разбиения) характеризуется следующей асимптотикой:

$$\theta = O(Hqz \log H), \quad (5)$$

где z – среднее число подмножеств вершин-кандидатов, выбираемых при формировании разбиения, а q – средняя мощность подмножества вершин-кандидатов. В пределе $q = z = O(N)$, поэтому с учетом (5):

$$\theta = O(N^2 H \log H). \quad (6)$$

По формуле (6) можно установить, что даже для сложных случаев ($N > 100$, $H > 10$) предельное время поиска разбиения составляет доли секунды. Так, при $N = 300$ и $H = 30$ получаем $\theta \approx 0.092$ с. Это на 2 порядка быстрее, чем при программной реализации аналогичных процедур.

Сложность устройства в числе эквивалентных логических элементов (ЭЛЭ) оценивается следующей асимптотической зависимостью:

$$T = O(N(N + H)) \approx O(N^2). \quad (7)$$

Например, при $N = 50$ согласно (7) имеем $T = 2500$ ЭЛЭ, при $N = 100$ получаем $T = 10000$ ЭЛЭ, а при $N = 200$ $T = 40000$ ЭЛЭ. Схема из $O(10000)$ вентилях считается в настоящее время схемой средней сложности, поэтому применение разработанного устройства (при условии его реализации на кремниевой пластине или на отдельной плате) целесообразно даже для довольно больших алгоритмов, имеющих $O(100)$ вершин.

Разработанные процедура и устройство могут найти применение при решении широкого круга задач, аналогичных рассмотренной в статье, например, стать прототипом при построении средств аппаратной поддержки распараллеливающих компиляторов, используемых в однородных параллельных вычислительных и управляющих системах широкого класса.

Библиографический список

Харченко В.С., Никольский С.Б., Сазонов А.Е. Один подход к синтезу дискретных микроконтроллерных сетей // АВТ.– 1989. – №4. – С.87–95.

Харченко В.С., Кальченко С.Б., Сазонов А.Е. Декомпозиция параллельных матричных схем алгоритмов в задачах синтеза микроконтроллерных сетей // АВТ. – 1990. – №4. – С. 81–89.

Соловьев В.В. Реализация на программируемых матрицах логики параллельных алгоритмов логического управления // УС и М. – 1995. – №6. – С. 24–30.

Зотов И.В., Колосков В.А., Титов В.С. Выбор оптимальных разбиений алгоритмов при проектировании микроконтроллерных сетей // АВТ. – 1997. – №5. – С. 51–62.

Курейчик В.М., Глушань В.М., Щербаков Л.И. Комбинаторные аппаратные модели и алгоритмы в САПР. – М.: Радио и связь, 1990. – 216 с.

Баранов С.И., Журавина Л.Н., Песчанский В.А. Обобщенный метод декомпозиции граф-схем алгоритмов // АВТ. – 1982. – №5. – С. 43–51.

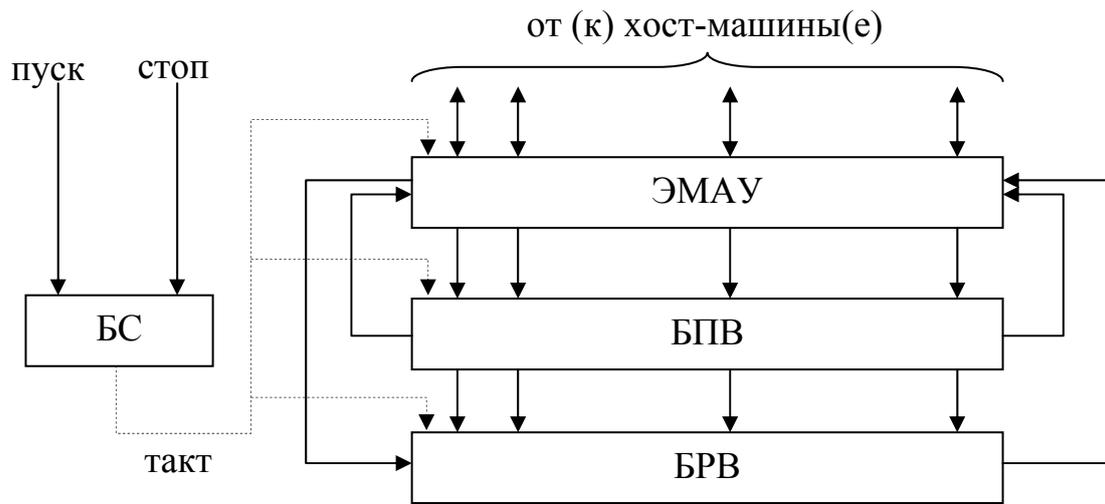


Рис.1

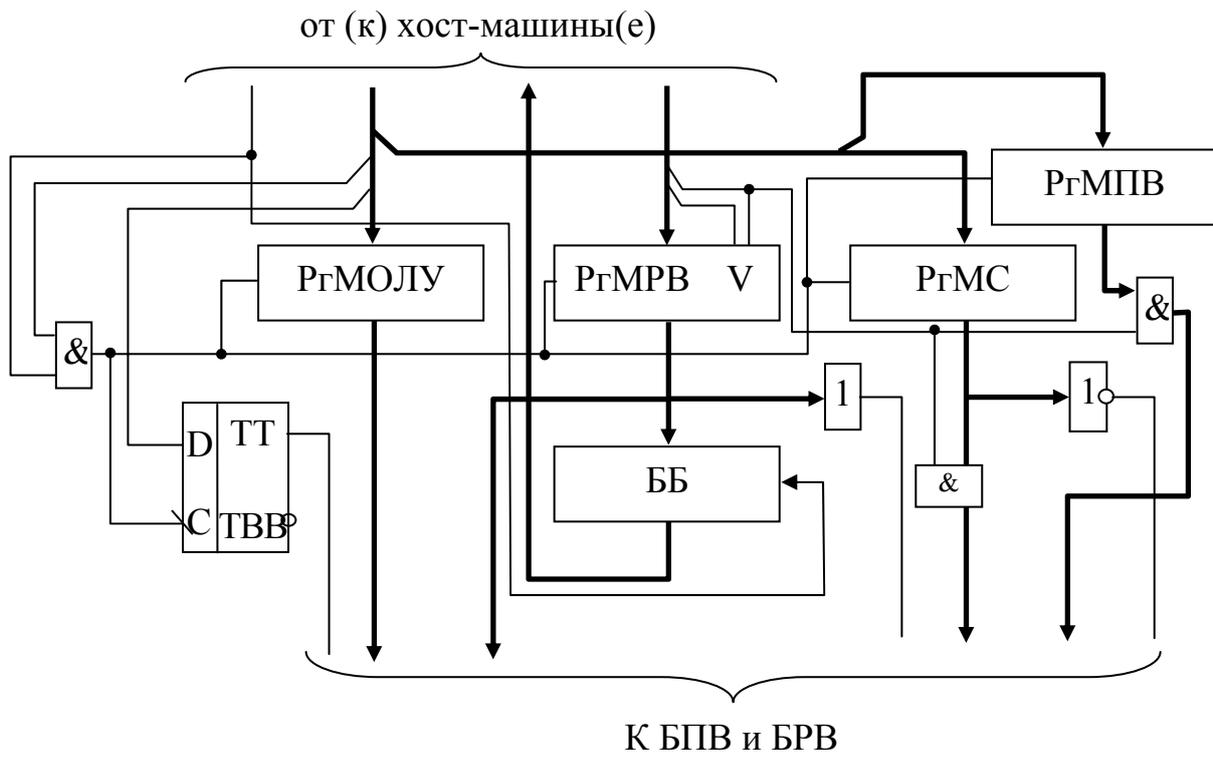


Рис.2

