УДК 004.384:004.272:004.414.2

Э.И. Ватутин, И.В. Зотов, В.С. Титов

Курский государственный технический университет

АЛГОРИТМ И УСТРОЙСТВО ВЫЯВЛЕНИЯ ИЗОМОРФНЫХ ВХОЖДЕНИЙ R-ВЫРАЖЕНИЙ ПРИ ПОСТРОЕНИИ МНОЖЕСТВА СЕЧЕНИЙ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ ЛОГИЧЕСКОГО УПРАВЛЕНИЯ

В статье предложен алгоритм выяснения отношения изоморфизма *R*-выражений (сечений параллельного алгоритма), основанный на ряде их особых свойств и ориентированный на аппаратную реализацию. Приведено описание устройства (акселератора) на его основе, позволяющего проводить проверку отношения изоморфизма за линейное время.

Реализация параллельных управляющих алгоритмов в базисе логических мультиконтроллеров (ЛМК) требует их декомпозиции на множество частных алгоритмов ограниченной сложности [1]. Получение оптимального набора частных алгоритмов (разбиения) – сложная комбинаторная задача. Качество ее решения существенно влияет на аппаратную сложность ЛМК и определяет, в конечном счете, время выполнения алгоритма. Один из наиболее эффективных путей решения данной задачи дает развиваемый авторами параллельно-последовательный метод декомпозиции [2-8]. Как показано в [9], он позволяет формировать наиболее близкие к оптимальным разбиения с учетом основных структурных и технологических ограничений базиса ЛМК.

Один из ключевых этапов параллельно-последовательной декомпозиции – построение множества сечений, покрывающего все вершины исходного алгоритма. Формирование сечений осуществляется путем выполнения трудоемких операций подстановки над множеством так называемых *R*-выражений, описывающих алгоритм управления. Как показывают исследования, упрощение и ускорение этих операций возможно путем их сведения к действиям над деревьями, в частности, к проверке изоморфизма.

В данной статье предложен алгоритм определения изоморфных вхождений *R*-выражений, основанный на ряде их специфических свойств, не присущих графам или деревьям общего вида, и обеспечивающий проверку изоморфизма за полиномиальное время. (Как известно [10], для решения задачи распознавания изоморфизма графов общего вида до сих пор не только не придумано эффективного универсального алгоритма, но и не доказана ее принадлежность к классу *P* или *NP* [11].) Следует отметить, что рассматриваемый вид изоморфизма *R*-выражений определяется исходя из возможности проведения операции подстановки [1, 3, 4] и несколько отличается от «классического» понятия изоморфизма [10], определяемого для графов, ввиду чего далее будем именовать его *r*-изоморфизмом.

При практической реализации операций над *R*-выражениями удобным является их представление в виде деревьев, допускающее преобразование в табличный вид (рис. 1). Каждый элемент дерева *X*, представленного совокупностью наборов листьев $L_1^X, L_2^X, ..., L_{n_L(X)}^X$, узлов $T_1^X, T_2^X, ..., T_{n_T(X)}^X$ и связей между ними, кодируется набором полей. Учитывая ряд особенностей обработки, наборы листьев и узлы дерева кодируются отдельно. Узлы дерева представлены полями типа узла (ТУ) – параллельный или альтернативный, ссылки на предка (СП) – номер узла-предка, номера соответствия (НС) – номер изоморфного эквивалента в соседнем дереве, типа соответствия (ТС) – может принимать значения «0*» – соответствие отсутствует, «10» – неполное (частичное) соответствие, «11» – полное соответствие; наборам листьев дерева при этом соответствуют поля множества вершин (МВ) – двоичный вектор с единичными битами в позициях, соответствующих номерам присутствующих в наборе вершин, а также перечисленные выше поля СП, НС и ТС. Во избежание путаницы при обозначении одноименных полей, соответствующих узлам и наборам листьев, будем обозначать их с указанием принадлежности к элементам дерева (например, СП«у» и СП«нл»).

Краткое описание свойств *R*-выражений, гарантирующих наличие единственно возможного изоморфного эквивалента поддерева и позволяющих исключить из рассмотрения проверку соответствия типов узлов деревьев, приведено ниже (ввиду ограниченного объема статьи доказательства лемм и теорем не приведены).

Необходимое условие 1 отсутствия *r*-изоморфизма. Если в дереве *A* найдется набор листьев L_i^A, не находящийся в отношении эквивалентности (обозначаемом как $L_1[\sim]L_2 \Leftrightarrow (L_1 = L_2) \vee (L_1 \subset L_2) \vee (L_2 \subset L_1)$) ни с одним набором листьев дерева *B*, то дерево *A* не является *r*-изоморфным дереву *B*: $\exists L_i^A \in A, \forall L_j^B \in B, L_i^A[\nsim]L_j^B \to A[\not\subseteq]B$.

Необходимое условие 2 отсутствия *r*-изоморфизма. Если в деревьях *A* и *B* присутствует более одной пары наборов листьев с частичным соответствием ($L_i^A \subset L_j^B$, где L_i^A – *i*-й набор листьев дерева *A*), рассматриваемые деревья не *r*-изоморфны.

Аксиома 1. Невозможно найти такие два набора листьев L_i^A и L_j^A, предком которых являлся бы один и тот же узел.

Аксиома 2. При движении по узлам любой ветви дерева типы узлов строго чередуются.

Лемма 1 (об ортогональности наборов листьев в составе дерева). В пределах дерева все наборы листьев ортогональны: $L_i^A \cap L_j^A = \varnothing, \forall i \neq j$.

Лемма 2 (о совпадении типа узлов-предков для вершин в составе различных наборов листьев). Любые две вершины, входящие в состав одного набора листьев L_i^X дерева *X*, могут одновременно войти в состав набора листьев L_j^Y дерева *Y* только в том случае, если тип узла предков наборов листьев L_i^X и L_j^Y совпадает.

Следствие. У полностью или частично эквивалентных наборов листьев не может быть предков разного типа.

Теорема 1 (о единственности *r*-изоморфной пары наборов листьев). Набору листьев L_i^A может соответствовать не более одного полностью или частично эквивалентного набора листьев L_j^B: $L_i^A\left[\sim\right]L_j^B$: $\exists L_k^B$, $L_i^A\left[\sim\right]L_k^B$, $k \neq j$.

Теорема 2 (о единственности *r*-изоморфной пары поддеревьев). В деревьях *A* и *B* не может быть более одной пары совпадающих поддеревьев.

Следствие. Дереву *A* может быть изоморфно не более одного поддерева из дерева *B*.

Приведенные выше аксиомы, леммы, теоремы и необходимые условия позволяют сформулировать алгоритм выявления *r*-изоморфизма пары деревьев *A* и *B*, ориентированный на параллельную аппаратную реализацию, в следующем виде.

1. (инициализация) Установить значения полей ТС всех наборов листьев дерева *A* в «00» (соответствия нет), значения полей НС всех наборов листьев дерева *A* в «11…1» (ссылка на несуществующий элемент дерева *B*), значения полей ТС всех узлов дерева *A* в «11» (полное соответствие), значения полей НС всех узлов дерева *A* в «11…1».
2. Если $n_L\left(A\right) > n_L\left(B\right)$ (количество наборов листьев в подставляемом дереве больше количества наборов листьев в объемлющем), установить признак $\varphi = 0$ отсутствия изоморфного дереву *A* поддерева в составе дерева *B*. Перейти к п. 7.
3. Поочередно выбирать все наборы листьев дерева *B*. Для выбранного набора листьев L_j^B дерева *B* осуществить параллельное во времени сравнение поля МВ с полями МВ всех наборов листьев дерева *A*, сформировать признаки полного $\delta^+ = \left(L_i^A = L_j^B\right)$ и частичного $\delta^- = \left(L_i^A \subseteq L_j^B\right)$ соответствия наборов листьев с их последующим сохранением в поле ТС набора листьев L_i^A в

формате $\left[\delta^- \mid \delta^+\right]$ (на позиции старшего бита – признак δ^-, младшего – δ^+) и в поле ТС предка набора листьев L_i^A. В случае присутствия полного или частичного соответствия между наборами листьев L_i^A и L_j^B сохранить номер *j*, соответствующий предположительно изоморфному эквиваленту L_i^A в составе дерева *B*, в поле НС набора листьев L_i^A, а также значение поля СП набора листьев L_j^B в поле НС предка набора листьев L_i^A (предположительно изоморфный эквивалент предка набора листьев L_i^A).

4. Если хотя бы для одного набора листьев дерева *A* не нашлось полностью или частично эквивалентного предположительно изоморфного набора листьев в составе дерева *B* $\left(\exists L_k^A : \delta_k^- = 0 \Leftrightarrow \lambda = \underset{i=\overline{0,\, N_L(A)-1}}{\&} \delta^-\left(L_i^A\right) \neq 1\right)$, установить признак $\varphi = 0$ отсутствия изоморфного дереву *A* поддерева в дереве *B*. Перейти к п. 7.
5. Просмотреть все узлы дерева *A*, кроме корня, в направлении от узлов большим номером к узлам с меньшим номером. Откорректировать значения поля ТС предка каждого *i*-го узла как $TC^{\uparrow\prime} = f\left(TC^{\downarrow}, TC^{\uparrow}\right)$ в случае наличия предка у предположительно изоморфного эквивалента текущего узла $\left(\tilde{\gamma} = 0\right)$, установить $TC^{\uparrow\prime} = 00$ в противном случае $\left(\gamma = 1\right)$. (Здесь $f\left(x, y\right)$ – функция корректировки значения поля ТС узла-предка, *x* – значение поля ТС узла-предка, *y* – узла-потомка, эквивалентная функции минимума двух аргументов для всех случаев, кроме $f\left(10,10\right) = f\left(10,11\right) = 00$). Задать значения поля НС предка рассматриваемого узла $a\left[i\right]$ как $HC^{\uparrow} = b[a[i].HC].C\Pi$, где $HC^{\uparrow}$ – значение поля НС узла-предка текущего узла.
6. Если поля ТС всех узлов дерева *A* имеют значение «11» или «10» $\left(\varphi_0 = \underset{i=\overline{0, N_T(A)-1}}{\&} \delta^-\left(T_i^A\right) = 1\right)$, в дереве *B* есть поддерево, изоморфное дереву *A*, причем подстановка изоморфизма определяется значениями полей НС (*i*-му

элементу дерева A соответствует изоморфный эквивалент $a[i].HC$ в дереве B). В противном случае в составе дерева B нет поддерева, изоморфного дереву A. Установить признак $\varphi = \varphi_0$.

7. Конец алгоритма.

Рассмотрим пример работы описанного алгоритма. В качестве объемлющего дерева B возьмем дерево, приведенное на рис. 1. Пример (табл. 1) показывает, что в состав дерева B (рис. 1) входит поддерево, r-изоморфное изображенному на рис. 2 дереву A.

Таблица 1. Пример определения r-изоморфизма

№ итерации	Пояснение к действию	Значения полей элементов дерева А											
		л0		л1		л2		у0		у1		у2	
		ТС	НС	ТС	НС	ТС	НС	ТС	НС	ТС	НС	ТС	НС
1	–	00	–1	00	–1	00	–1	11	–1	11	–1	11	–1
2	$N_L(A)=3,\ N_L(B)=5,\ n_L(A)\le n_L(B)$	00	–1	00	–1	00	–1	11	–1	11	–1	11	–1
3.1	$j=0,\ B.л_0.MB=a_{99}$	00	–1	00	–1	00	–1	11	–1	11	–1	11	–1
3.2	$j=1,\ B.л_1.MB=a_{94}a_{95}a_{128}$	00	–1	00	–1	00	–1	11	–1	11	–1	11	–1
3.3	$j=2,\ B.л_2.MB=a_{65}a_{66}a_{82}a_{83},\ i=0,\ A:л_0\to y_0, B:л_2\to y_2$	10	2	00	–1	00	–1	10	2	11	–1	11	–1
3.4	$j=3,\ B.л_3.MB=a_{55}a_{56}a_{112},\ i=2,\ A:л_2\to y_2, B:л_3\to y_3$	10	2	00	–1	11	3	10	2	11	–1	11	3
3.5	$j=4,\ B.л_4.MB=a_{59}a_{104}a_{111},\ i=1,\ A:л_1\to y_1, B:л_4\to y_4$	10	2	11	4	11	3	10	2	11	4	11	3
4	$\lambda=1\&1\&1=1$	**1**0	2	**1**1	4	**1**1	3	10	2	11	4	11	3
5.1	$i=2,\ A:y_2\to y_0,\ A.y_2[\sim]B.y_3$ $B:y_3\to y_2,\ \gamma=0,\ y_0.TC=f(11,10)=10$	10	2	11	4	11	3	10	2	11	4	11	3
5.2	$i=1,\ A:y_1\to y_0,\ A.y_1[\sim]B.y_4$ $B:y_4\to y_2,\ \gamma=0,\ y_0.TC=f(11,10)=10$	10	2	11	4	11	3	10	2	11	4	11	3
6	$\varphi=1\&1\&1=1$	10	2	11	4	11	3	**1**0	2	**1**1	4	**1**1	3

На основании приведенного выше алгоритма синтезировано специализированное устройство, функциональная схема которого приведена на рис. 3. Регистры 1, 2 и 27 предназначены для хранения значений $n_L(B)$, $n_T(A)$ и $n_L(A)$ соответственно, при этом значения полей обрабатываемых деревьев хранятся в элементах однородной среды электронной модели дерева (ОСЭМД) [12] B.MB, A.MB, A.СП(нл), A.TC(нл), A.HC(нл), B.СП(нл), A.СП(у), B.СП(у), A.TC(у) и A.HC(у), схема ячейки которой приведена на рис. 4. Коммутаторы 3, 23–25 управляются сигналом ρ и предназначены для коммутации значений на разных этапах работы алгоритма. Дешифраторы 4, 22, 28, 33, 34, 38 используются для преобразования номеров позиций элементов дерева в табличном представлении из двоичного кода в унитарный, используемый на входах *ra* (сокр. от Read Address – адрес чтения) элементов ОСЭМД. Шифратор 21 используется для обратного преобразования. Сдвиговый регистр 5 используется для поочередного выбора элементов обрабатываемых деревьев. Блоки элементов ИЛИ 7, запрета 8, ИЛИ 9–12 используются для формирования начальных значений полей на этапе инициализации. Элементы ИЛИ 6, ИЛИ 13 и И 36 используются для коммутации синхросигналов. Блок элементов запрета 26 в совокупности с элементом И 29 и схемой маскировки неиспользуемых позиций (СМНП) 1 используются для формирования значения признака λ. Схема СМНП 2 в совокупности с блоком элементов запрета 37, элементом ИЛИ 39, запрета 30, блоком элементов И 31 и элементом ИЛИ 32 используются для формирования результирующего признака *r*-изоморфизма φ, при этом на выходе элемента ИЛИ 39 формируется значение признака σ отсутствия узлов у дерева *A*. Элемент И 35, на выходе которого формируется значение признака $\tilde{\gamma}$ в совокупности со схемой определения типа соответствия предка (СОТСП) отвечают за формирование обновленного значения поля типа соответствия узла-предка (фактически схема СОТСП реализуется рассмотренную выше функцию *f*). Блок элементов НЕ 14 в совокупности со схемами сравнения битовых векторов (ССБВ) 1–n, образованными блоками элементов И 15 и 16, ИЛИ 17, И-НЕ 19 и элементами И 18 и 20 используются для формирования признаков δ_i^+ и δ_i^- полного и частичного соответствия наборов листьев. Элемент задержки предназначен для задержки синхросигнала сдвига содержимого регистра 5 на время, достаточное для формирования и записи обновленных значений полей на предыдущей итерации алгоритма.

По сравнению с программной реализацией предлагаемого алгоритма выигрыш в скорости достигается за счет параллельной инициализации значений полей ТС и НС узлов и наборов листьев дерева A; параллельного сравнения выбранного j-го набора листьев дерева B со всеми наборами листьев дерева A; параллельного сравнения компонентов битовых векторов, соответствующих наборам листьев; параллельного чтения и записи значений различных полей (А.ТС(нл), А.НС(нл), А.НС(у) и т.д.) из/в разные элементы ОСЭМД.

При программной реализации [3, 5, 6] имеют место дополнительные затраты на организацию стека возвратов при рекуррентном сравнении деревьев, сканирование наборов листьев при этом является фактически условием завершения рекурсии: обработка элементов наборов листьев производится последовательно. Просмотр узлов деревьев как при программной, так и при аппаратной реализации осуществляется последовательно.

Последовательная программная реализация предложенного в статье алгоритма проверки r-изоморфизма предполагает попарное сравнение наборов листьев деревьев A и B (каждое сравнение наборов листьев требует $O\left(L_{\max}\right)$ действий) с последующим просмотром узлов дерева A снизу вверх. Ее асимптотическая временная сложность составляет $O\left(N_{L_{\max}}^2 L_{\max} + N_T\right) \simeq O\left(L_{\max}^3\right)$, где $N_{L_{\max}}$ – максимально возможное число наборов листьев в дереве, $L_{\max}$ – размерность битовых векторов МВ наборов листьев (фактически число вершин в алгоритме управления). Предложенное аппаратное решение обладает асимптотической временной сложностью $O\left(N_{L_{\max}}+N_{T_{\max}}\right)\simeq O\left(L_{\max}\right)$, где $N_{T_{\max}}$ – максимально возможное число узлов в дереве.

СПИСОК ЛИТЕРАТУРЫ

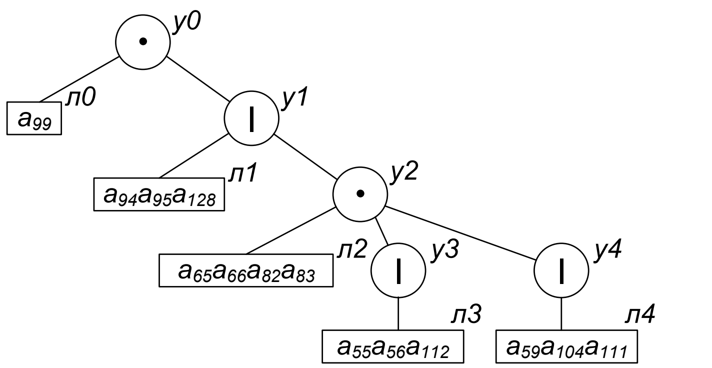
1. Организация и синтез микропрограммных мультимикроконтроллеров / *И.В. Зотов, В.А. Колосков, В.С. Титов* [и др.]. Курск: КурскГТУ, 1999. 368 с.
2. *Зотов И.В., Колосков В.А., Титов В.С.* Выбор оптимальных разбиений алгоритмов при проектировании микроконтроллерных сетей // Автоматика и вычислительная техника. 1997. № 5. С. 51̃62.
3. Поиск базового сечения в задаче разбиения параллельных алгоритмов / *Ватутин Э.И., Зотов И.В.*; КГТУ. Курск, 2003. 30 с. Рус. деп. в ВИНИТИ 24.11.03 № 2036-В2003.
4. *Ватутин Э.И., Зотов И.В., Титов В.С.* Построение множества сечений в задаче оптимального разбиения параллельных управляющих алгоритмов // Известия ТулГУ. Вычислительная техника. Информационные технологии. Системы управления. Тула: ТулГУ, 2003. Т. 1. Вып. 2. С. 70–77.
5. *Ватутин Э.И., Зотов И.В.* Метод формирования субоптимальных разбиений параллельных управляющих алгоритмов // Параллельные вычисления и задачи управления (PACO'04). М.: ИПУ РАН. 2004. С. 884–917.
6. *Ватутин Э.И., Зотов И.В.* Параллельно-последовательный метод формирования субоптимальных разбиений параллельных управляющих алгоритмов // Свидетельство об официальной регистрации программы для ЭВМ № 2005613091 от 28.11.05.
7. *Ватутин Э.И., Зотов И.В.* Программная система для построения разбиений параллельных управляющих алгоритмов // Идентификация систем и задачи управления (SICPRO'06). М.: ИПУ РАН, 2006. С. 2239–2250.
8. *Ватутин Э.И., Зотов И.В.* Визуальная среда синтеза разбиений параллельных алгоритмов логического управления // Свидетельство об официальной регистрации программы для ЭВМ № 2007613222 от 30.07.07.
9. *Ватутин Э.И., Волобуев С.В., Зотов И.В.* Комплексная сравнительная оценка методов выбора разбиений при проектировании логических мультиконтроллеров // Идентификация систем и задачи управления (SICPRO'08). М.: ИПУ РАН, 2008. С. 1917–1940.
10. *Зыков А.А.* Основы теории графов. – М.: «Наука», 1987. – 381 с.
11. *Хопкрофт Дж., Мотвани Р., Ульман Дж.* Введение в теорию автоматов, языков и вычислений, 2-е изд.: Пер. с англ. – М.: «Вильямс», 2002. – 528 с.

12. *Ватутин Э.И.* Однородная среда электронной модели дерева для аппаратно-ориентированной обработки R-выражений // Оптико-электронные приборы и устройства в системах распознавания образов, обработки изображений и символьной информации (Распознавание – 2008). Ч. 1. Курск: изд-во КурскГТУ, 2008. С. 90–92.

Рекомендована кафедрой вычислительной техники

Поступила в редакцию 01.09.08

$$a_{99}\bullet(a_{94}|a_{95}|a_{128}|(a_{65}\bullet a_{66}\bullet a_{82}\bullet a_{83}\bullet(a_{55}|a_{56}|a_{112})\bullet(a_{59}|a_{104}|a_{111})))$$



Узлы

№	Тип узла	Ссылка на предка
у0	•	-
у1	|	у0
у2	•	у1
у3	|	у2
у4	|	у2
у5	-	-
...	...	...
$уN_{Tmax}$	-	-

Текущее количество узлов (N_T)
5

Текущее количество наборов листьев (N_L)
5

Наборы листьев

№	Множество вершин	Ссылка на предка
л0	99	у0
л1	94, 95, 128	у1
л2	65, 66, 82, 83	у2
л3	55, 56, 112	у3
л4	59, 104, 111	у4
л5	-	-
...	...	...
$лN_{Lmax}$	-	-

Рис. 1.

$$a_{66}\bullet a_{83}\bullet(a_{59}|a_{104}|a_{111})\bullet(a_{55}|a_{56}|a_{112})$$

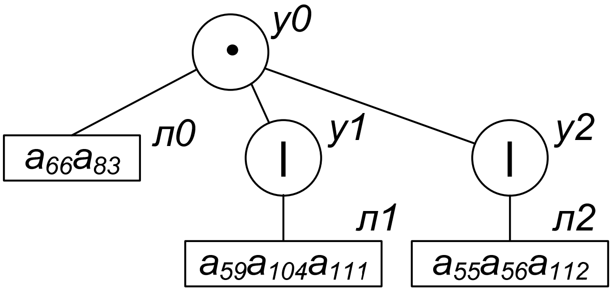


Рис. 2.

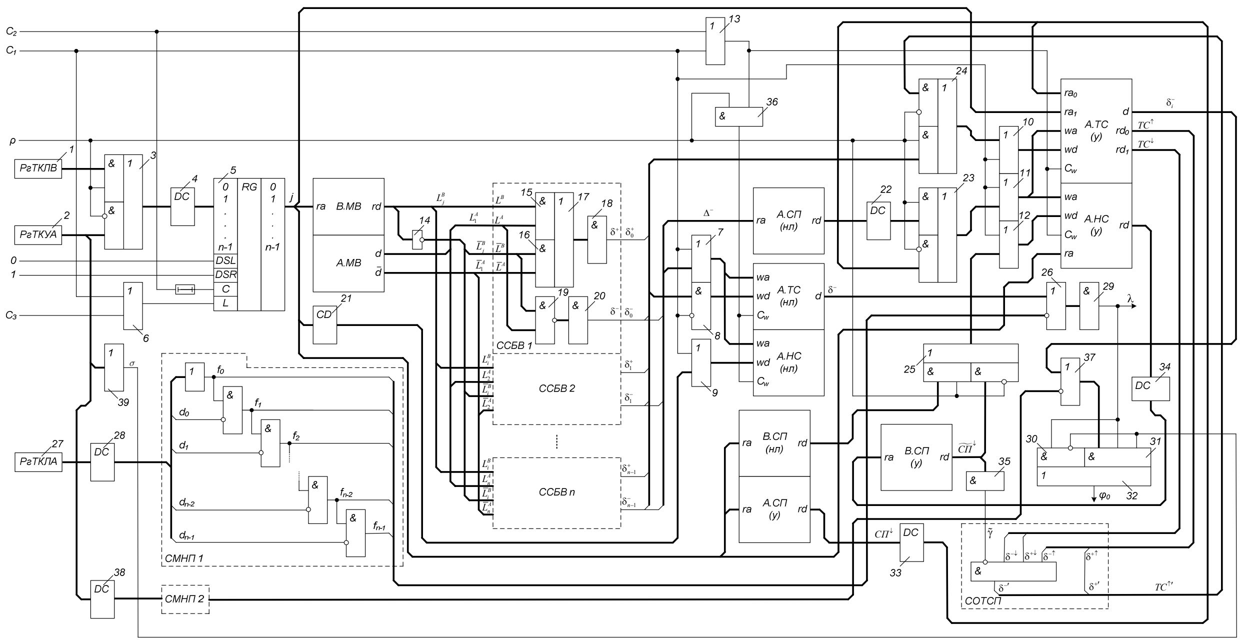


Рис. 3.

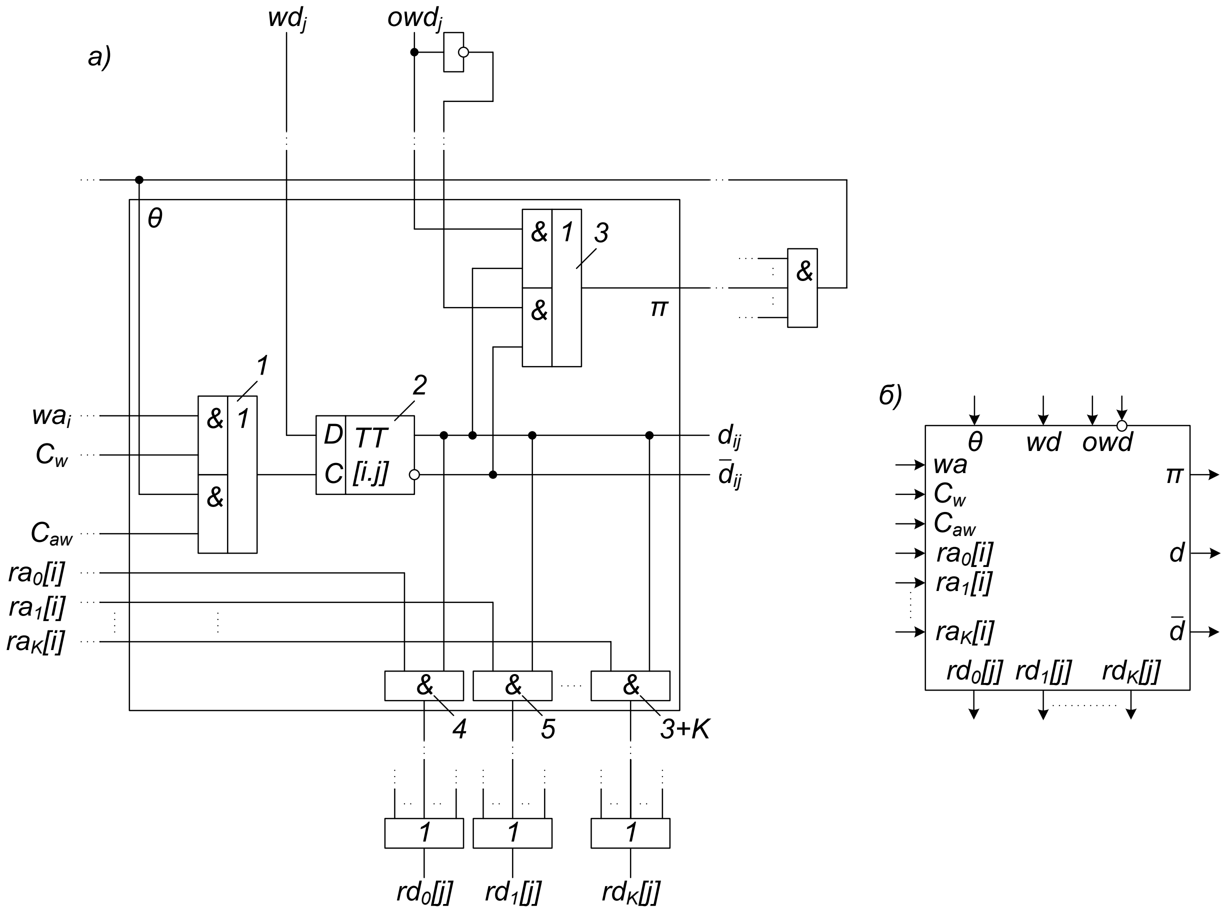

а)
wdj
owdj
θ
π
1
2
3
D TT
C [i.j]
wai
Cw
Caw
ra0[i]
ra1[i]
raK[i]
dij
4
5
3+K
rd0[j]
rd1[j]
rdK[j]
б)
θ wd owd
wa
Cw
Caw
ra0[i]
ra1[i]
raK[i]
π
d
rd0[j] rd1[j] rdK[j]

Рис. 4.