

В. П. Семеренко, канд. техн. наук  
(Винницкий политехнический ин-т)

## Систолическая реализация кубических функций

Рассмотрено применение булевой алгебры кубических функций в качестве теоретической основы для проектирования нового класса вычислительных структур. Предложен вариант универсальной программируемой систолической структуры, в которой обеспечивается принцип однородности на уровне составляющих ее ячеек.

Розглянуто застосування бульової алгебри кубічних функцій як теоретичної основи для проектування нового класу обчислювальних структур. Запропоновано варіант універсальної програмуємої систолическої структури, у якій забезпечується принцип однорідності на рівні ячеек, що її складають.

*Ключевые слова:* булева алгебра, программируемая логическая интегральная схема, систолическая структура.

В настоящее время все более расширяется применение программируемых логических интегральных схем (ПЛИС): от простейших ПЛМ до сложных многоступенчатых и многоблочных матриц, содержащих до 20 000 эквивалентных вентилях [1, 2].

Несмотря на достигнутые успехи, имеется много проблем, связанных с проектированием, изготовлением и использованием ПЛИС. В современных программируемых БИС так и не удалось разрешить противоречие между универсальностью и специализацией.

Существенным недостатком выпускаемых схем является большое время реализации заданной функции. В многоступенчатых ПЛИС можно реализовать любую многоуровневую комбинационную схему путем объединения нескольких уровней внутренних вентилях, однако задержка распространения сигнала на каждый логический уровень составляет 10—20 нс [2].

К числу нерешенных проблем ПЛИС можно отнести неэффективное использование площади кристалла (50—80 %) и сложность репрограммирования.

Одной из причин, обуславливающей наличие указанных проблем, является то, что внутренняя структура всех ПЛИС по сути содержит точные функциональные эквиваленты стандартных логических микросхем средней и малой степени интеграции. Таким образом, с объединением элементов в БИС суммируются и недостатки, присущие традиционному методу проектирования цифровых устройств на основе заданного набора отдельных микросхем.

Рассмотрим новый способ вычисления булевых функций, используемый для получения нового класса программируемых схем.

© В. П. СЕМЕРЕНКО, 1992.

**Булева алгебра кубических функций.** С момента становления теории проектирования ЭВМ формальным аппаратом для описания логической стороны процессов в цифровых устройствах служит булева алгебра логических функций, или алгебра Линденбаума-Тарского, которая изоморфна алгебре логических функций [3].

Основываясь на известном определении булевой алгебры (булевой структуры) как математического объекта, можно предложить ее различные конкретные реализации [4].

Рассмотрим пример реализации булевой алгебры на основе математического аппарата исчисления кубических комплексов, в котором используются понятия  $r$ -кубов и операций над ними ( $r=0, 1, \dots, n$ ) [5]. В общем виде  $r$ -куб представляет собой  $n$ -мерный набор  $e_1 e_2 \dots e_n$  ( $e_i = \{0, 1, \times\}$ ), причем число  $r$  независимых координат в кубе определяет его размерность ( $i=1 \div n$ ). Два  $r$ -куба, содержащие  $r$  одноименных независимых координат  $\times$  и различающиеся только одной координатой, могут образовывать один  $(r+1)$ -куб,  $(r+1)$ -я независимая координата которого соответствует координате, которой различаются  $r$ -кубы.

Рассмотрим структуру вида

$$H = \langle A_H, \cap, \cup, - \rangle, \quad (1)$$

где на множестве элементов  $A_H$ ,

$$A_H = \{\alpha, \beta, \gamma, \dots, O, I\},$$

выделены два «особых» элемента  $O$  и  $I$ , определены две бинарные операции (пересечение  $\cap$  и объединение  $\cup$  кубов), одна унарная операция (дополнение  $-$ ).

Элементами множества  $A_H$  являются  $r$ -кубы ( $r=0, 1, \dots, n-1$ ), а также один  $n$ -куб («особый» элемент  $I$ ) и один пустой куб («особый» элемент  $O$ ).

Если для определения операций пересечения, объединения и дополнения кубов использовать систему аксиом аналогично как и для общепринятого определения булевой структуры [4], то структуру вида (1) можно рассматривать как булеву структуру.

Рассмотрим множество  $Z$ , содержащее все  $O$ -кубы из множества  $A_H$  ( $Z \subset A_H$ ). Сопоставим каждому из  $2^n$  кубов множества  $Z$  число 0 или 1, т. е. проведем однозначное отображение множества  $Z$  на множество  $Y = \{0, 1\}$ .

Назовем кубической функцией, которая дает однозначное отображение  $Z$  в  $Y$ . Множество  $O$ -кубов, на которых кубическая функция  $\varphi$  равна 1 (0), будем называть единичным (нулевым) кубическим комплексом  $K^0_{(1)}$  ( $K^0_{(0)}$ ), а булеву структуру вида (1) — булевой алгеброй кубических функций, которая является изоморфной булевой алгебре логических функций.



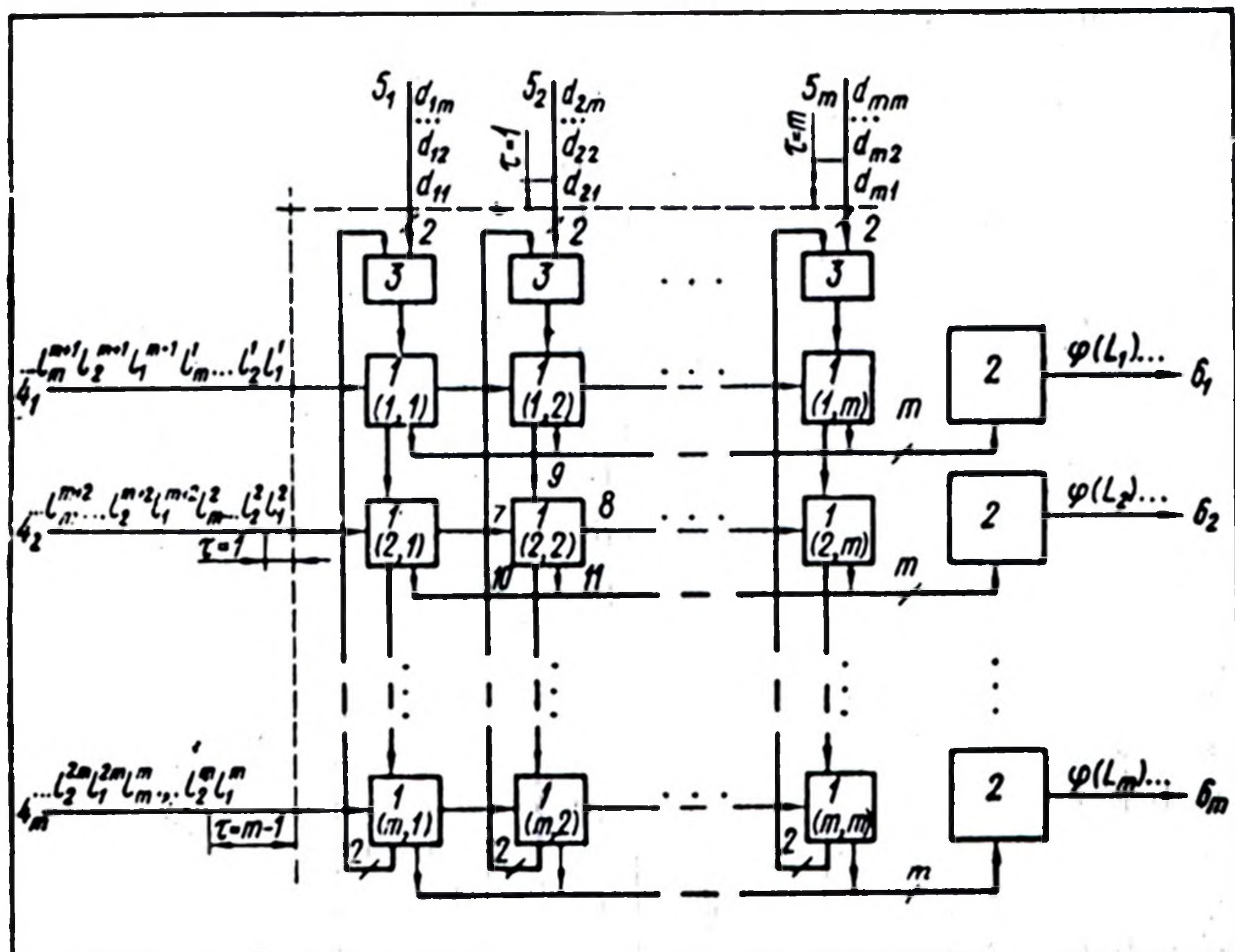


Рис. 1.

Операциям дизъюнкции, конъюнкции и отрицания в булевой алгебре логических функций соответствуют операции объединения, пересечения и дополнения кубов в булевой алгебре кубических функций II вида (1). Комплекс  $K^0_{(1)}$  ( $K^0_{(0)}$ ) соответствует множеству элементарных конъюнкций совершенной дизъюнктивной нормальной формы (СДНФ) логической функции  $\psi$  (инверсной логической функции  $\bar{\psi}$ ).

Назовем  $D$ -покрытием ( $R$ -покрытием) минимальное число кубов максимальной размерности  $r$  ( $r \geq 0$ ), которые в совокупности содержат все кубы комплекса  $K^0_{(1)}$  ( $K^0_{(0)}$ ).

Каждый куб  $d_j(r_j)$   $D$ -покрытия ( $R$ -покрытия) кубической функции  $\psi$  соответствует одной импликанте минимальной ДНФ изоморфной ей логической функции  $\psi$  (инверсной логической функции  $\bar{\psi}$ ) таким образом, что единичное (нулевое) значение компоненты куба соответствует прямому (инверсному) значению переменной в импликанте минимальной ДНФ, причем число  $n$  компонент куба равно числу переменных логической функции ( $j = 1 \div m$ ).

Например, минимальной ДНФ логической функции  $\psi$  (инверсной логической функции  $\bar{\psi}$ ),

$$\psi(a, b, c, e) = \bar{a}\bar{c} \vee \bar{b}\bar{c} \vee e,$$

$$\bar{\psi}(a, b, c, e) = ab\bar{e} \vee c\bar{e},$$

соответствуют следующие покрытия:

$$D = \left\{ \begin{array}{cccc} a & b & c & e \\ 0 & \times & 0 & \times \\ \times & 0 & 0 & \times \\ \times & \times & \times & 1 \end{array} \right\}; \quad R = \left\{ \begin{array}{cccc} a & b & c & e \\ 1 & 1 & \times & 0 \\ \times & \times & 1 & 0 \end{array} \right\}. \quad (2)$$

Эти покрытия однозначно определяют функционирование комбинационной схемы, поэтому

используется только одно из них, а именно покрытие, которое содержит меньшее число кубов. В дальнейшем будет рассматриваться только  $D$ -покрытие.

Традиционный метод синтеза комбинационной схемы (КС) на основе заданных МДНФ логической функции  $\psi$  (или покрытий кубической функции  $\psi$ ) заключается в выборе совокупности логических элементов из заданной системы логических элементов, которые соединяются таким образом, чтобы КС реализовала функцию  $\psi$  (или  $\bar{\psi}$ ).

Выполняемую КС операцию можно интерпретировать как установление принадлежности входного набора  $L_h$  множеству наборов, на которых функция  $\psi(L_h)$  или  $\bar{\psi}(L_h)$  принимает значение логической единицы (логического нуля).

В алгебре кубических функций установление принадлежности входного набора  $L_h$  указанному множеству наборов может быть выполнено аналитически с помощью операции пересечения кубов.

Суть данной операции заключается в следующем. По определению [5] операция пересечения куба  $\alpha = \alpha_1 \alpha_2 \dots \alpha_n$  и куба  $\beta = \beta_1 \beta_2 \dots \beta_n$  обозначается как  $\gamma = \alpha \cap \beta$  и служит для выделения куба  $\gamma = \gamma_1 \gamma_2 \dots \gamma_n$ , являющегося общей частью кубов  $\alpha$  и  $\beta$ . Значение компоненты  $\gamma_i$  определяется по табл. 1:  $\gamma_i = \alpha_i \cap \beta_i$  ( $i = 1 \div n$ ). Знак  $\emptyset$  означает пустое пересечение. Например, если  $\alpha = \{1 \times 1 0\}$ ,  $\beta = \{\times 0 \times 0\}$ , то куб  $\gamma$  равен 1010:

$$\begin{array}{c} \cap \\ \begin{array}{cccc} 1 & \times & 1 & 0 \\ \times & 0 & \times & 0 \\ \hline 1 & 0 & 1 & 0 \end{array} \end{array}$$

Входной набор  $L_h$  будет принадлежать множеству наборов, на которых функция  $\psi(L_h)$  принимает значение единицы (нуля), если существует непустое пересечение набора  $L_h$  хотя бы с одним кубом  $D$ -покрытия (имеет место пустое пересечение набора  $L_h$  со всеми кубами  $D$ -покрытия):

$$\psi(L_h) = \begin{cases} 1, & \text{если } L_h \cap d_j \neq \emptyset \text{ для некоторого } j, \\ 0, & \text{если } L_h \cap d_j = \emptyset \text{ для всех } j, \end{cases} \quad (3)$$

$$L_h = (l_1^h \dots l_k^h \dots l_n^h);$$

$$d_j = (d_{j1} \dots d_{jk} \dots d_{jn}), \quad j = 1 \div m_D, \quad d_j \in D.$$

Таблица 1

$\alpha_i$	$\beta_i$		
	0	1	$\times$
0	0	$\emptyset$	0
1	$\emptyset$	1	1
$\times$	0	1	$\times$

Таблица 2

$l_k^i$	$d'_{jk}$	$d''_{jk}$	$\psi$
0	0	1	0
0	1	1	1
0	1	0	0
1	0	1	1
1	1	1	0
1	1	0	0



Например, набор  $L_n=1101$  принадлежит множеству наборов, на которых функция  $\psi(a, b, c, e)$  принимает единичное значение, поскольку существует непустое пересечение с одним кубом  $D$ -покрытия (2):

$$\begin{array}{cccc} \cap & 1 & 1 & 0 & 1 \\ & \times & \times & \times & 1 \\ \hline & 1 & 1 & 0 & 1 \end{array}$$

**Вычисление кубических функций.** Отличительной особенностью выполнения операций над кубами является возможность разбить весь процесс выполнения операций пересечения входных наборов аргументов кубической функции с кубами покрытий этой функции на элементарные независимые один от другого фрагменты вычислений и организовать параллельную обработку данных.

Схема для реализации кубической функции представляет собой устройство, в котором можно выделить операционный и управляющий автоматы. Операционный автомат, именуемый в дальнейшем программируемой систолической структурой (ПСС), содержит  $m \times m$  ячеек 1,  $m$  элементов свертки (ЭС) 2 и  $m$  элементов ИЛИ 3 (рис. 1). Управляющий автомат синтезируется традиционным способом.

Ячейка 1, представленная на рис. 2, предназначена для выполнения компонентной операции пересечения  $l_k^i \cap d_{jk}$ , а также хранения исходных данных и результатов указанной операции в течение одного цикла работы ПСС. Элемент свертки, изображенный на рис. 3, предназначен для формирования окончательного результата вычисления кубической функции на соответствующих входных наборах ее аргументов.

Число  $m_D$  кубов покрытия и число  $n$  компонент кубов связаны с параметром  $m$  ПСС следующим образом:  $m_D \leq m, n \leq m$ .

Если  $n < m$ , то  $(m-n)$  разрядов соответствующего покрытия дополняется значениями  $\times$ . Если  $m_D < m$ , тогда к соответствующему покрытию добавляется  $(m-m_D)$  строк, являющихся копиями одной из строк этого покрытия. Для простоты изложения будем предполагать, что  $m_D = m, n = m$ .

Процесс вычисления значений кубической функции осуществляется по циклам, каждый из которых состоит из трех тактов.

$D$ -покрытие кубической функции  $\varphi$ , значения которой определяются с помощью ПСС, записывается в матрицу ячеек 1 через вертикальную группу входов  $5_1, \dots, 5_m$ . На  $j$ -й вертикальный вход  $5_j$  со сдвигом во времени на  $(j-1)$  цикл относительно поступления компонент первого куба  $d_1$  поступают компоненты  $j$ -го куба  $d_j$ ,

$$d_j = (d_{j1}d_{j2} \dots d_{jm}),$$

начиная с компоненты  $d_{j1}$  ( $j=1 \div m$ ).

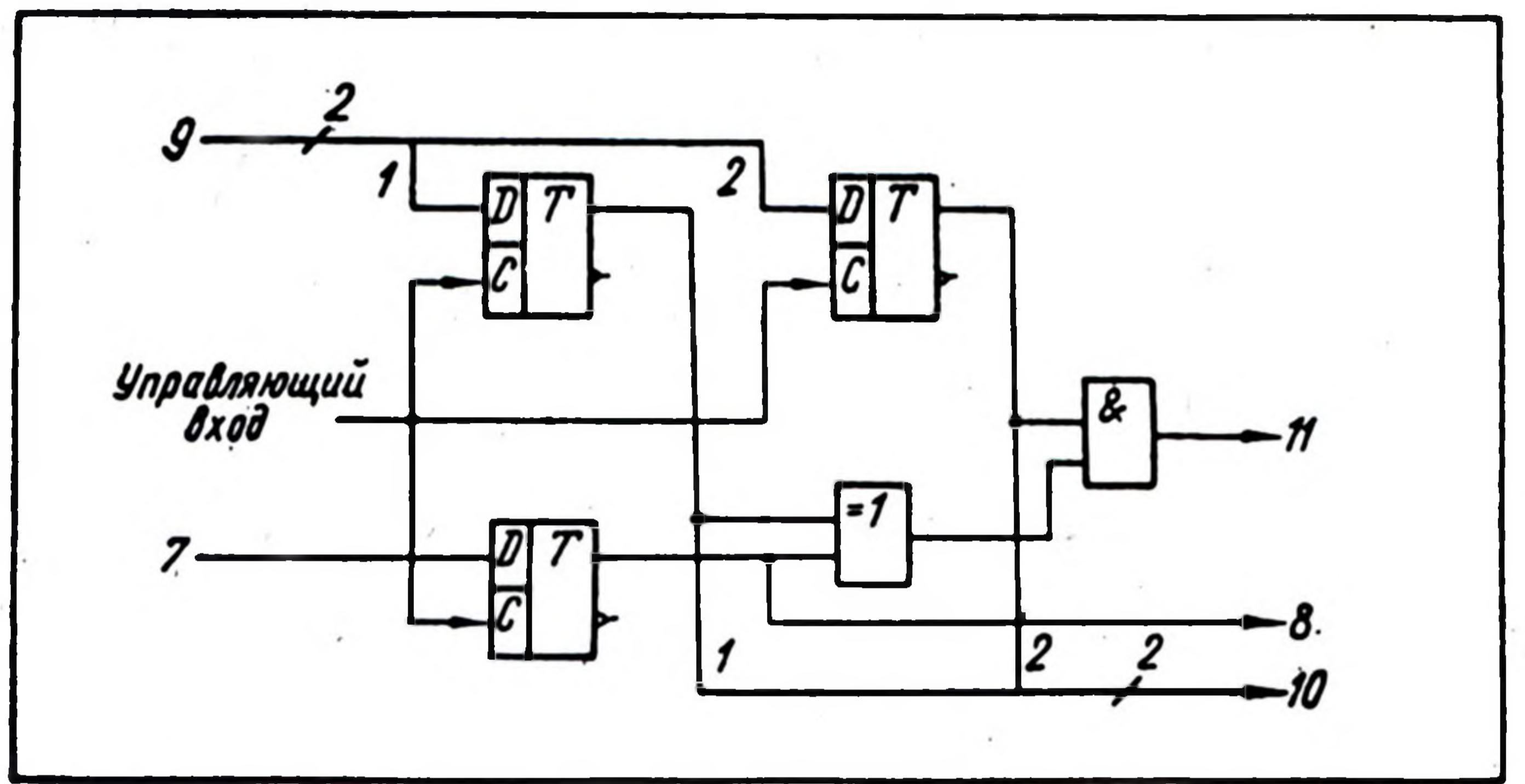


Рис. 2.

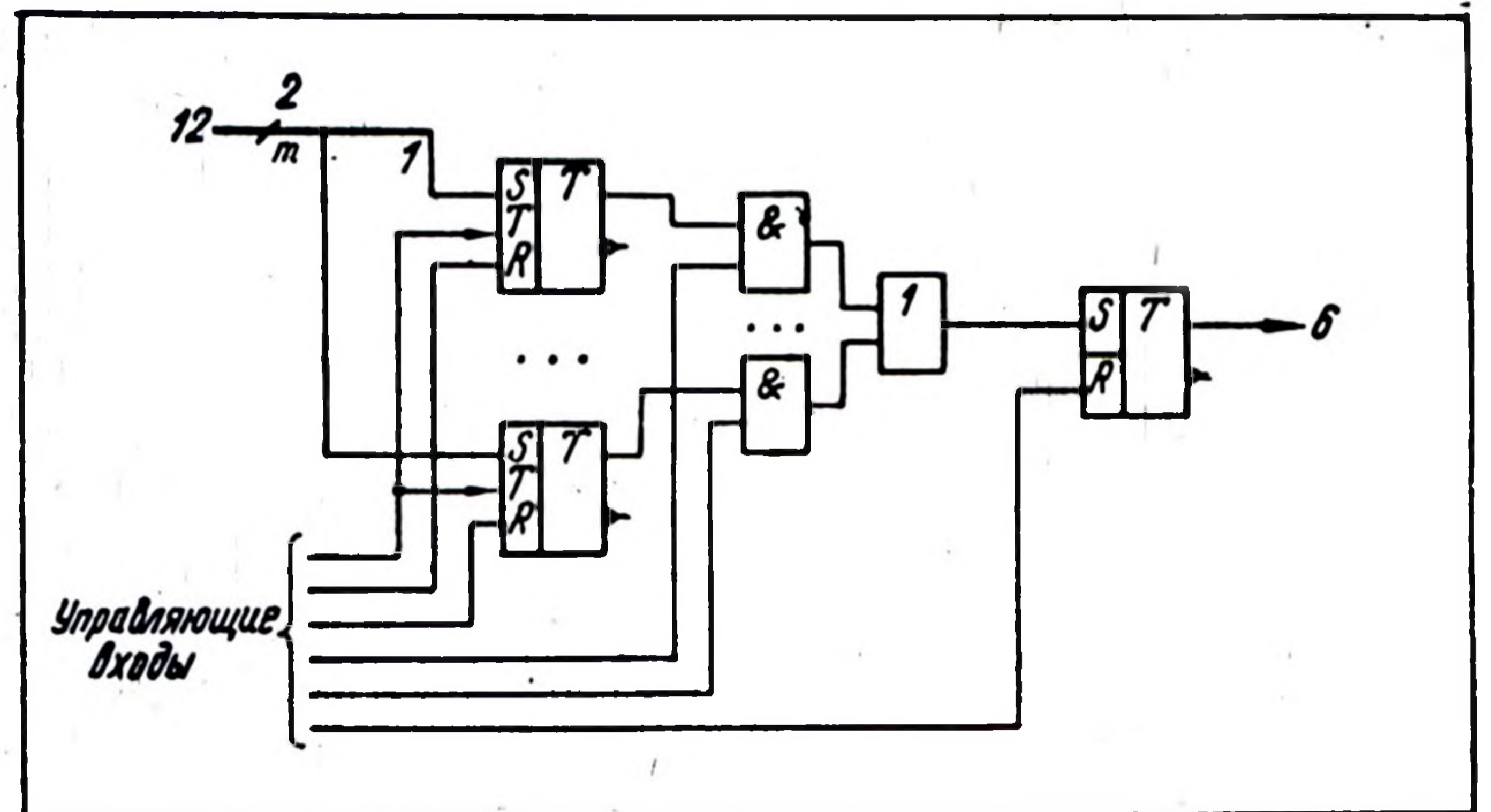


Рис. 3.

В начале каждого цикла компоненты кубов  $D$ -покрытия с  $i$ -й ячейки 1 переписываются в соседнюю снизу  $(i+1)$ -ю ячейку 1 ( $i=1 \div m$ ); с  $m$ -й ячейки 1 по цепи обратной связи через схему ИЛИ 3 переписываются в первую ячейку этого же столбца.

На  $i$ -й горизонтальный вход  $4_i$  со сдвигом во времени на  $(i-1)$  циклов относительно поступления компонент входного набора  $L_1$  аргументов кубической функции  $\varphi$  поступают компоненты набора  $L_i$ ,

$$L_i = (l_1^i l_2^i \dots l_m^i),$$

начиная с компоненты  $l_1^i$  ( $i=1 \div m$ ). После поступления на  $i$ -й горизонтальный вход  $4_i$  последней компоненты  $l_m^i$  набора  $L_i$  на следующем цикле на указанный вход поступает первая компонента  $l_1^{i+m}$  набора  $L_{i+m}$ :

$$L_{i+m} = (l_1^{i+m} l_2^{i+m} \dots l_m^{i+m}).$$

На рис. 1 величина сдвига во времени в циклах обозначена буквой  $\tau$ .

Процесс вычислений в матрице ячеек 1 начинается с активизации ячейки 1 с координатами  $(1, 1)$ .

В первом такте первого цикла работы в этой ячейке выполняется операция

$$l_1^1 \cap d_{11}. \quad (4)$$



Поскольку в настоящее время отсутствуют элементы для непосредственной реализации операции пересечения кубов, ячейки ПСС должны реализоваться на стандартных логических элементах. Такая реализация возможна, если перейти к двоичному представлению  $d'_{jk}d''_{jk}$  компоненты  $d_{jk}$   $D$ -покрытия следующим образом:

$$0 \rightarrow 01$$

$$1 \rightarrow 11$$

$$\times \rightarrow 10.$$

Тогда таблица истинности логической функции  $\psi$ , реализуемой ячейкой ПСС, представится табл. 2, а ее аналитическая форма будет иметь вид

$$\psi = l_k^i \cap d'_{jk}d''_{jk} = (l_k^i \oplus d'_{jk}) \& d''_{jk}, \quad j = 1 \div m, \\ i = 1 \div m, \quad k = 1 \div m.$$

На втором такте первого цикла работы в первый  $RST$ -триггер первого ЭС 2 записывается результат (4).

После циклического сдвига в матрице ячеек 1 на их вертикальные входы с координатами (1,1), (1,2), (2,1) поступают соответствующие компоненты  $D$ -покрытия, а на горизонтальные входы указанных ячеек 1 — соответствующие компоненты входных наборов. В результате на первом такте второго цикла работы выполняются следующие операции:

$$l_2^1 \cap d_{12} \text{ в ячейке 1 с координатами (1,1),}$$

$$l_1^1 \cap d_{21} \text{ в ячейке 1 с координатами (1,2), (5)}$$

$$l_1^2 \cap d_{11} \text{ в ячейке 1 с координатами (2,1).}$$

На втором такте второго цикла работы (см. рис. 1) полученные результаты (5) в ячейках 1 с координатами (1,1) и (1,2) записываются в первый ЭС 2 (соответственно в первый и второй  $RST$ -триггеры), а результат (5) в ячейке 1 с координатами (2,1) записывается во второй ЭС 2 (в первый  $RST$ -триггер).

Начальные состояния всех  $RST$ -триггеров нулевые.

Далее фронт выполняемых операций перемещается к ячейкам 1 с координатами (3,1), (2,2) и (1,3) и, следовательно, порождается волна вычислений, бегущая вниз по матрице ячеек 1.

В ячейке 1 с координатами  $(i, j)$ , начиная с  $(i+j-1)$ -го цикла и в течение последующих  $(m-1)$  циклов, выполняется операция пересечения компонент набора  $L_i$  с компонентами одного куба  $d_j$ , при этом определяется следующая

подфункция:

$$\varphi_j(L_i) = \begin{cases} 1, & \text{если } l_k^i \cap d_{jk} = \emptyset \text{ для некоторого } k, \\ 0, & \text{если } l_k^i \cap d_{jk} \neq \emptyset \text{ для всех } k, \\ & i = 1 \div m, j = 1 \div m, k = 1 \div m. \end{cases} \quad (6)$$

Значение подфункции  $\varphi_j(L_i)$  фиксируется в  $j$ -м  $RST$ -триггере  $i$ -го ЭС 2.

При прохождении компонент набора  $L_i$  через все ячейки 1  $i$ -й строки поочередно формируется результат выполнения операции пересечения  $L_i$  со всеми кубами  $D$ -покрытия. Эти результаты будут зафиксированы в соответствующих  $RST$ -триггерах  $i$ -го ЭС 2.

На первом такте  $(i+j+m-1)$ -го цикла  $j$ -й  $RST$ -триггер  $i$ -го ЭС 2 приравнивается нулю и тем самым подготавливается к формированию результата операции пересечения куба  $d_j$  с набором  $L_{i+m}$ .

Окончательный вывод о результатах операции пересечения набора  $L_i$  с  $D$ -покрытием формируется на третьем такте  $(m+m+i-2)$ -го цикла работы на основе следующего соотношения:

$$\varphi(L_i) = \begin{cases} 1, & \text{если } \varphi_j(L_i) = 0 \text{ для некоторого } j, \\ 0, & \text{если } \varphi_j(L_i) = 1 \text{ для всех } j, j = 1 \div m. \end{cases} \quad (7)$$

Соотношения (6) и (7) эквивалентны соотношению (3).

Значение функции  $\varphi(L_i)$  фиксируется в  $RS$ -триггере  $i$ -го ЭС 2, т. е. на выходе ПСС.

Значение кубической функции  $\varphi$ , начиная с  $(2m-1)$ -го цикла, будут появляться на каждом цикле поочередно на выходах ПСС в последовательности, приведенной на рис. 1. ПСС может программироваться на вычисление любой кубической функции.

Поскольку булева алгебра логических функций изоморфна булевой алгебре кубических функций, с помощью ПСС, синтезированных на стандартных логических элементах, вычисляются по сути и логические функции.

Систематическая реализация стандартных вычислительных узлов. Для реализации на основе ПСС любого вычислительного узла необходимо лишь найти соответствующие кубические покрытия, которые определяют функционирование заданного узла, и записать их в ПСС.

Очевидно, что представление многих стандартных вычислительных узлов (умножителей, многоадресных сумматоров и т. д.) в виде одного покрытия практически нереально ввиду его большой размерности. Такие узлы могут быть описаны конечным набором покрытий отдельных его подсхем. Тогда каждая подсхема представится одной ПСС небольших размеров, а межсоединения отдельных ПСС будут определяться конфигурацией связей подсхем в узле.

Например,  $n$ -разрядный комбинационный сумматор целесообразно представить в виде  $n$   $D_c$ -покрытий функции суммы и  $n$   $D_n$ -покрытий



функции переноса одноразрядного сумматора вида

$$D_c = \begin{pmatrix} 0 & 0 & 1 & \times \\ 0 & 1 & 0 & \times \\ 1 & 0 & 0 & \times \\ 1 & 1 & 1 & \times \end{pmatrix};$$

$$D_n = \begin{pmatrix} 1 & 1 & 0 & \times \\ 0 & 1 & 1 & \times \\ 1 & 0 & 1 & \times \\ 1 & 1 & 1 & \times \end{pmatrix}.$$

Следует отметить, что в систолическом сумматоре параллельно и независимо функционируют четыре  $n$ -разрядных последовательных сумматора. Общее число  $N_{cc}$  логических элементов (триггеров и вентилях) в нем составит  $144n$ .

На основе ПСС может быть также реализовано вычислительное устройство с памятью (конечный автомат) при условии представления его в виде комбинационной схемы, содержащей контуры обратной связи с задержками. Комбинационная часть такого устройства может быть выполнена в виде набора ПСС. При этом обратные связи сохраняются, однако моделировать задержки нет необходимости, поскольку ПСС уже является автоматом.

Полученный в итоге систолический автомат (СА) позволяет в конвейерном режиме вычислять автоматные функции от  $m$  входных последовательностей (если ПСС содержит  $m \times m$  ячеек).

В синхронном автомате смена внутренних состояний происходит через каждый цикл длительностью  $t_{ca}$ . За время  $T_{ca} = mt_{ca}$  в СА выполняются действия, которые моделируют смену одного состояния традиционного автомата (ТА) в течение времени  $t_{ta}$ . Другими словами, величины  $t_{ca}$  и  $t_{ta}$  представляют собой длительности автоматного такта соответственно в СА и в моделируемом ТА. При этом величина  $t_{ca}$  всегда постоянна, поскольку ПСС неизменна, а величина  $t_{ta}$  зависит от сложности комбинационной части ТА.

Пусть для вычисления некоторой автоматной функции  $\delta(L_i)$  необходимо  $r$  автоматных тактов. Тогда период времени вычисления указанной функции от  $m$  входных последовательностей  $L_1, \dots, L_m$  составит соответственно в ТА и СА

$$T_{TA} = mrt_{TA},$$

$$T_{CA} = (m-1)t_{ca} + mrt_{ca}.$$

Следовательно, при обработке больших массивов информации производительность СА больше, чем производительность ТА, если  $t_{ca} < t_{ta}$ . Отличительной особенностью СА, отсутствующей в ТА, является его универсальность.

Принципиальное отличие кубических покрытий от других форм представления булевых функций, в том числе от матричного представления, заключается в возможности параллельно-конвейерной обработки покрытий. При этом уровень конвейерной обработки имеет такой минимальный цикл, который недостижим при других формах представления булевых функций. В результате существенно повышается тактовая частота и, следовательно, производительность ПСС по сравнению с известными ПЛИС.

ПСС является универсальным устройством и его архитектура не зависит от вида реализуемой функции. Настройка ПСС на выполнение заданной функции осуществляется путем ее записи в ячейки ПСС. В процессе функционирования ПСС обеспечивается равномерное и полное использование пространства схемы.

При большом числе  $n$  переменных булевой функции ПСС экономичнее ПЗУ, поскольку в ПЗУ необходимо записывать все  $2^n$  импликант совершенной ДНФ булевой функции.

По сравнению с традиционными систолическими структурами [6] рассматриваемый подход обеспечивает регулярность вычислительной структуры не только на макроуровне, но и на уровне ячеек. Полученная в итоге микрорегулярная систолическая структура является более быстродействующей и универсальной. При этом, однако, увеличиваются аппаратные затраты, которые могут быть существенно снижены за счет учета конкретных особенностей реализуемой схемы.

V. P. Semerenko

#### SYSTOLIC REALIZATION OF CUBIC FUNCTIONS

Polytechnical Institute, Vinnitsa

The use of the Boolean algebra of cubic functions as a theoretical fundamentals for design of a new class of computer structures is considered. A variant of the universal programmed systolic structure is suggested where the principle of uniformity at the level of its cells is provided.

1. Баранов С. И., Скляр В. А. Цифровые устройства на программируемых БИС с матричной структурой.— М.: Радио и связь, 1986.— 272 с.
2. Бернгард К. Коул. Второе поколение программируемых логических ИС // Электроника.— 1988.— № 10.— С. 18—22.
3. Кузнецов О. И., Адельсон-Вельский Г. М. Дискретная математика для инженера. 2-е изд.— М.: Энергоатомиздат, 1988.— 480 с.
4. Яглом И. М. Булева структура и ее модели.— М.: Сов. радио, 1980.— 192 с.
5. Миллер Р. Теория переключательных схем.— М.: Наука, 1970.— Т. 1.— 416 с.
6. Гун Сюньюань. Систолические и волновые матричные процессоры для высокопроизводительных вычислений. // Тр. Ин-та инж. по электротехнике и радиоэлектрон.— 1984.— 72, № 7.— С. 133—153.

Поступила 11.02.91;  
после доработки 29.05.91