

## **БУЛЕВА МОДЕЛЬ ЛОГИКИ ПЕРЕСТРАИВАЕМЫХ СТРУКТУР**

---

---

### **2.1. ВВОДНЫЕ ПОНЯТИЯ**

Построение систем логического управления разнообразными объектами промышленной автоматики, обеспечивающих их эффективную согласованную работу в соответствии с заданным алгоритмом функционирования, всегда считалось одной из важнейших задач автоматизации производственных процессов.

За последние годы элементная база систем логического управления быстро изменялась, пройдя путь от релейно-контактных элементов, микросхем малого и среднего уровней интеграции к программным логическим устройствам, а затем к большим и сверхбольшим интегральным схемам, частным случаем которых являются микропроцессоры и микроконтроллеры, используемые в качестве вычислительного ядра, в том числе и в промышленных (управляющих) компьютерах, и в программируемых логических контроллерах.

Особенности построения систем логического управления на основе многофункциональных логических модулей состоят в том, что средством реализации алгоритмов функционирования проектируемых систем являются функциональные модули, которые могут настраиваться на реализацию конкретного логического алгоритма из заданного класса.

Настраивая один (или несколько) однотипный элемент, на основе которого создается логическое устройство управления, на реализацию алгоритма функционирования последнего, можно построить заданное устройство управления в базе лишь одного универсального элемента [9, 15]. Практически удобнее иметь набор различных элементов (универсальный базис) не на основе одного типа перестраиваемого элемента, а в виде набора небольшого числа (обычно не более пяти-шести) типов перестраиваемых элементов [4].

Разработки же формальных методов аппаратной и программной реализации алгоритмов логического управления были актуальны на протяжении многих лет и являются актуальными в настоящее время.

В логическом проектировании кроме задач, непосредственно возникающих из потребностей практики, имеется также уже ставшая традиционной область теоретических задач, связанных с исследованиями как класса всех булевых функций, являющихся одним из важнейших объектов дискретной математики, так и его подклассов.

Таким образом, большой интерес представляют исследования булевых формул одной из реализаций булевых функций, первичным заданием которых являются таблицы истинности. При этом особую важность имеют простейшие из булевых формул – бесповторные и повторные.

## 2.2. КЛАССИФИКАЦИЯ БУЛЕВЫХ ФОРМУЛ

Функция, у которой ее значения и значения аргументов двоичны, называется булевой [6].

Булевы функции (БФу) задаются в виде таблиц истинности (ТИ). Аналогичная запись БФу в виде формулы, использующей логические операции (базис формулы), называется булевой формулой (БФ).

В настоящей работе рассматриваются формулы в базисе И, ИЛИ, НЕ. Зафиксируем базис логических операций. При этом основной метрикой БФ является число  $h$  символов переменных и их инверсий в правой ее части, которые будем называть буквами.

В настоящее время БФ в фиксированном базисе из  $h$  букв могут быть разделены на два класса: бесповторные, для которых  $h = n$ , и повторные, для которых  $h > n$  (рис. 2.1). В настоящей работе, если это не оговаривается особо, рассматриваются одиночные, полностью определенные БФу  $n$  переменных, заданных на  $2^n$  входных наборах. Основной метрикой БФу является число переменных  $n$ .

Каждый логический элемент, реализующий отрицание и дизъюнкцию либо отрицание и конъюнкцию, в определенном смысле универсален, ибо, выбирая нужное число элементов и схему их соединения, можно построить любой логический автомат. Согласно [1] основную (93–99,7%) долю реализуемых в устройствах управления составляют бесповторные БФ или близкие к ним.

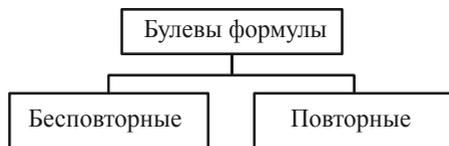


Рис. 2.1. Классификация булевых формул

Формулу будем называть бесповторной, если каждый аргумент, взятый в прямой или инверсной форме, входит в нее не более одного раза. Во всех остальных случаях формула является повторной.

Бесповторные БФ подвергнем дальнейшей классификации (рис. 2.2) [11]. Повторные БФ имеют аналогичную классификацию.

Если в записи бесповторной булевой формулы  $f(x_i)$ , где  $i = 1, 2, \dots, n$ ,  $n$  – число аргументов, индекс  $i$  при логических аргументах возрастает слева направо, то будем считать, что эта формула упорядочена. Упорядоченной будем считать формулу и в том случае, если существуют тождественные преобразования, в результате которых получается запись формулы с возрастающими слева направо индексами аргументов. Если в записи бесповторной булевой формулы аргументы с теми или иными индексами отсутствуют, будем считать, что эта формула содержит пропуски соответствующих аргументов, и назовем ее формулой с пропусками аргументов. Формулами с пропусками аргументов будем считать и такие бесповторные булевы формулы, для которых  $\mu < n$ , где  $\mu$  – максимальный индекс при логическом аргументе.

**Пример 1.** Формула  $f_1 = A_1 \vee A_2 \bar{A}_3 \vee A_4$  зависит от аргументов  $A_1, A_2, A_3, A_4$ , является упорядоченной, представлена в дизъюнктивно нормальной форме (ДНФ) и не содержит пропусков аргументов.

**Пример 2.** Формула  $f_2 = (\bar{A}_1 \vee A_2)(\bar{A}_3 \vee A_4)$  зависит от аргументов  $A_1, A_2, A_3, A_4$ , является упорядоченной, представлена в конъюнктивно нормальной форме (КНФ) и не имеет пропусков аргументов.

**Пример 3.** Формула  $f_3 = [(A_1 \vee A_2)A_3 \vee A_4]A_5$  зависит от аргументов  $A_1, A_2, A_3, A_4, A_5$ , представлена в форме четвертого порядка, является упорядоченной и не содержит пропусков аргументов.



Рис. 2.2. Классификация бесповторных булевых формул

Пример 4. Формула  $f_4 = A_1 \vee A_3 A_4$  зависит от аргументов  $A_1$ ,  $A_2$ ,  $A_3$ ,  $A_4$ , упорядочена, представлена в ДНФ с пропуском аргумента  $A_2$ .

Пример 5. Формула  $f_5 = (A_1\bar{A}_2 \vee A_3\bar{A}_4)A_5$  зависит от аргументов  $A_1, A_2, A_3, A_4, \dots, A_8$ , является упорядоченной, представлена в КНФ с пропуском аргументов  $A_6, A_7, A_8$ .

Пример 6. Формула  $f_6 = A_1\bar{A}_3 \vee A_2\bar{A}_4$  зависит от аргументов  $A_1, A_2, A_3, A_4$ , является неупорядоченной, представлена в ДНФ и не содержит пропусков аргументов.

В дальнейшем будем пользоваться приведенной классификацией.

Таким образом, в соответствии с приведенной классификацией (см. рис. 2.2) множество булевых формул распадается на попарно непересекающиеся классы – множества однотипных формул. Каждую формулу определенного класса можно выбрать в качестве представителя этого класса. Булевы формулы, принадлежащие одному классу, реализуются физически одинаковыми схемами, поэтому для каждого класса достаточно реализовать лишь одну схему, структура которой описывается формулой представителя класса.

### 2.3. БУЛЕВА МОДЕЛЬ ЛОГИКИ ПЕРЕСТРАИВАЕМЫХ СТРУКТУР ДЛЯ ОПРЕДЕЛЕННЫХ КЛАССОВ БУЛЕВЫХ ФОРМУЛ

Разработка сложных дискретных управляющих устройств требует обычно больших трудозатрат, поэтому является экономически оправданным создание логических управляющих устройств, которые способны при соответствующей перестройке, не затрагивающей их структуры, управлять работой других объектов.

Любой перестраиваемый автомат может рассматриваться как множество автоматов с одними и теми же выходами, причем настройка определяет тот автомат, выходы которого считаются при этой настройке выходами всего перестраиваемого автомата.

Многофункциональные логические модули, настраиваемые на реализацию любой из  $2^{2^n}$  булевых формул от  $n$  входных переменных, на-

зываются универсальными логическими модулями (УЛМ) от  $n$  переменных и описываются, согласно [7], следующей булевой формулой:

$$U = a_1 z_1 \vee a_2 z_2 \vee \dots \vee a_{2^n} z_{2^n},$$

где  $a_1, \dots, a_{2^n}$  – все возможные конъюнкции от  $n$  переменных и их отрицаний. Приравнивая некоторые  $z$  нулю, а остальные – единице, получим формулу, зависящую только от  $x_1, \dots, x_n$  и заданную в совершенной дизъюнктивной нормальной форме (СДНФ). Меняя подстановку 0 и 1 на множестве  $z_i$  ( $i=1, \dots, 2^n$ ), можно получать различные СДНФ.

На практике часто необходимо создавать перестраиваемый логический модуль, реализующий не все возможные булевы формулы, а любую формулу из заранее заданного множества, т.е. перестраиваемый автомат, универсальный на множестве булевых формул. Перестраиваемые модули, универсальные на множествах формул, по сравнению с УЛМ имеют меньшее число настроечных входов и меньшую сложность, поэтому разработка моделей логики перестраиваемых структур имеет большое значение для практики. Ниже приведем одну из них, позволяющую получить перестраиваемые автоматы с простой настройкой (настройка, при которой используется два сигнала: 0 и 1).

Суть заключается в следующем:

1. Пусть задано множество функций алгебры логики, каждая из которых может быть реализована на выходе перестраиваемого автомата:  $f_1(x_1, \dots, x_n), f_2(x_1, \dots, x_n), \dots, f_R(x_1, \dots, x_n)$ .

2. Введем  $t = \log_2 R$  дополнительных переменных  $z_1, \dots, z_t$ .

3. Поставим каждой из исходных функций алгебры логики  $f_i$  ( $i=1, \dots, t$ ) конъюнкцию  $a_i = z_1^{\sigma_{i,1}} \dots z_t^{\sigma_{i,t}}$ , где  $\sigma_{i,j} \in \{0,1\}$  ( $j=1, \dots, t$ ),  $z_j^0 = \bar{z}_j$ ,  $z_j^1 = z_j$ , так, чтобы разным функциям алгебры логики сопоставлялись различные конъюнкции.

4. Образует формулу

$$U = a_1 f_1(x_1, \dots, x_n) \vee a_2 f_2(x_1, \dots, x_n) \vee \dots \vee a_R f_R(x_1, \dots, x_n).$$

Логическая схема, реализующая заданную формулу, имеет информационные входы  $x_1, \dots, x_n$  и дополнительные входы  $z_1, \dots, z_n$ , исполь-

зубые для настройки. При этом если, например, нужно настроить полученный перестраиваемый автомат на реализацию функции алгебры логики  $f_i$ , которой сопоставлена конъюнкция  $z_1^{\sigma_{i1}} \dots z_i^{\sigma_{ii}}$ , то на настроечные входы  $z_1, \dots, z_i$  подаются соответственно сигналы-константы  $\sigma_{i1}, \dots, \sigma_{ii}$ .

Таким образом, используя булеву модель логики перестраиваемых структур, можно создать логическое устройство управления с перестраиваемой структурой, реализующее алгоритм управления из заранее заданного множества булевых формул.

## 2.4. МНОГОФУНКЦИОНАЛЬНЫЕ ЛОГИЧЕСКИЕ МОДУЛИ. ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ

Напомним, что под МЛМ будем понимать перестраиваемый автомат, для любых настроек которого значение на его выходах зависит от значений сигналов на его входах.

Сигналы, подаваемые на входы МЛМ, разделяются на две группы. Сигналы, несущие информацию, предназначенную для обработки МЛМ, назовем *информационными* и обозначим  $A_1, A_2, \dots, A_n$ , где  $n$  – число аргументов. Эти сигналы включаются на  $n$  входов МЛМ. На остальные  $\eta = (h - n)$  входов подаются сигналы, которые назовем *настроечными*, где  $h$  – общее количество входов МЛМ.

Настройку, при которой используется два сигнала (0, 1), назовем *простой*, а настройку, которая производится  $n + 2$  сигналами  $(0, 1, z_1, z_2, \dots, z_n)$ , будем называть *нормальной*. Нормальная настройка по сравнению с простой зачастую позволяет увеличивать число функций, реализуемых МЛМ. Однако использование для настройки сигналов  $z_1, z_2, \dots, z_i$  приводит к необходимости подавать сигнал  $z_i$ , где  $i \in \{1, 2, \dots, n\}$ , не на один, а сразу на несколько входов.

При использовании для настройки всех  $2n + 2$  сигналов, а именно  $(0, 1, z_1, z_2, \dots, z_n, \bar{z}_1, \bar{z}_2, \dots, \bar{z}_n)$ , настройка называется *сложной*. По отношению к нормальной сложная настройка позволяет увеличить число функций, реализуемых МЛМ. Однако ее использование оказывается возможным лишь при наличии информационных сигналов и их отрицаний.

В отношении выбора входов МЛМ, на которые подаются сигналы настройки, существуют две возможности. Если для подачи  $\eta$  сигналов настройки выделяется  $\eta$  определенных входов МЛМ, то говорят, что он имеет *фиксированные* входы настройки. Если же для настройки могут быть использованы любые  $\eta$  из  $h$  входов МЛМ, то будем считать, что он имеет *произвольные* входы настройки [20].

Также будем считать, что имеется некоторый алфавит  $N$ ,  $N = (n_1, n_2, \dots, n_k)$ , который используется для кодирования букв алфавита  $X, Y, Z$ .

Пусть  $X = \{x_1, x_2, \dots, x_n\}$ ,  $Y = \{y_1, y_2, \dots, y_m\}$ ,  $Z = \{z_1, z_2, \dots, z_p\}$  – конечные алфавиты. Функции  $f$ , определенные на  $X^*$  и принимающие значения  $Y^*$  в зависимости от значений  $Z^*$ , могут быть использованы для описания поведения МЛМ в дискретные моменты времени  $t = 1, 2, \dots$ , где  $X^*$ ,  $Y^*$  и  $Z^*$  – множество слов в алфавитах  $X, Y, Z$ . При этом слово  $x$ ,  $x \in X^*$ , интерпретируется как последовательность входных сигналов,  $f(x) \in Y^*$  – как последовательность сигналов, возникающих на выходе МЛМ, в зависимости от содержания слов  $z$ ,  $z \in Z^*$ .

Множества слов алфавитов  $X$  и  $Z$  могут быть составными частями некоторого множества  $M$ . Введем следующие понятия.

1. Если настроенный код подается на настроенные входы МЛМ, то будем говорить, что речь идет о прямом варианте реализации функции  $N_1$ .

2. Если же настроенный код подается на место входных сигналов МЛМ, а входные сигналы – на настроенные входы, то условимся называть это обратным вариантом реализации функции ( $N_2$ ), т.е. слова из алфавитов  $X$  и  $Z$  меняются местами. Тогда имеем

$$M_1 = \bar{N}_1 \cap N_2 \neq 0;$$

$$M_2 = N_1 \cap \bar{N}_2 \neq 0;$$

$$M_3 = N_1 \cap N_2 \neq 0.$$

Функции из множеств  $M_2$  и  $M_3$  соответствуют прямой реализации, а функции из множеств  $M_1$  могут быть реализованы лишь в обратном варианте.

В МЛМ с фиксированными входами настройки сигналы подаются только на входы, выделенные для этой цели. Важной особенностью элементов с фиксированными входами настройки (по сравнению с элементами, имеющими произвольные выходы настройки) является возможность уменьшения числа внешних вводов. Правда, это достигается за счет усложнения логической структуры элемента и принятия  $n = \text{const}$ . Кроме того, при использовании элементов с фиксированной настройкой нет необходимости в переключении информационных сигналов, что позволяет строить логические автоматы, в которых функциональные возможности элементов изменяются во время переработки информации.

Таким образом, фиксация входов ячейки (МЛМ) позволяет сделать внутренними все входы настройки и подавать на них сигналы. При этом заметим, что появляются новые (дополнительные) внешние входы настройки.

Будем различать МЛМ с внешней и внутренней настройкой. Структурная схема ячейки с внешней настройкой (рис. 2.3), использующая многопозиционный код, может быть описана выражением [20]:

$$Y_j = \bigvee_{i=1}^{\gamma} f_i A_i, \quad (2.1)$$

где  $f_i$  – базовая формула, т.е. формула, принадлежащая произвольному множеству формул  $f_1, f_2, \dots, f_\gamma$ ;  $A_i$  – сигнал настройки,  $j \in \{1, 2, \dots, m\}$ ;  $\gamma$  – число базовых формул.

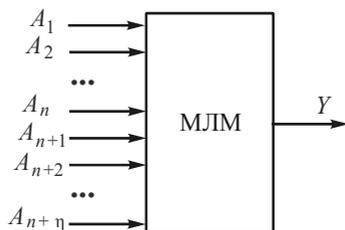


Рис. 2.3. Структурная схема элемента с внешней настройкой

Структура МЛМ с внутренней настройкой представлена на рис. 2.4. Основной логический блок описывается в зависимости от выбранного

метода кодирования уравнением (2.1) либо уравнением вида

$$Y_j = \bigvee_{i=1}^{\gamma} f_i A_1^i \dots A_n^i \text{ при использовании однопозиционного кода.}$$

На входы  $A_1, A_2, \dots, A_n$  подаются информационные сигналы, на входы  $A_{n+1}, A_{n+2}, \dots, A_{n+\eta}$ , являющиеся внутренними, подаются сигналы настройки  $I_1, I_2, \dots, I_\eta$ , генерируемые блоками настройки  $E_1, E_2, \dots, E_\eta$ .

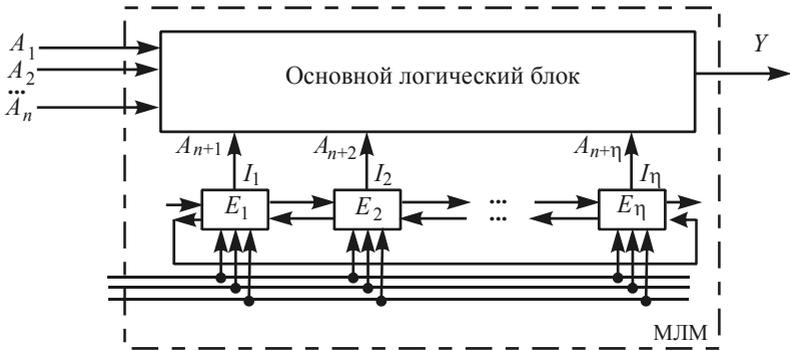


Рис. 2.4. Структура элемента с внутренней настройкой

С помощью рассматриваемых в данной работе МЛМ можно строить различные однородные среды. Напомним, что под *однородной* средой в дальнейшем будем понимать дискретное устройство с итеративной структурой его функциональной схемы, если в качестве элементов этой схемы используются *однотипные* МЛМ. Если же дискретное устройство построено на основе *разнотипных* МЛМ, то подобные среды будем называть *квазиоднородными*. Квазиоднородные среды позволяют, с одной стороны, с помощью применения нескольких типов МЛМ существенно сократить избыточность, которая неизбежна при применении одного типа МЛМ, а с другой – сохранить все основные достоинства однородных сред.

Таким образом, используя различные МЛМ, можно создавать электронные устройства с перестраиваемой структурой, которые за счет перестройки в процессе или перед началом работы могут решать различные задачи.

## 2.5. ВЫЧИСЛЕНИЕ БУЛЕВЫХ ФОРМУЛ

### 2.5.1. Вычисление неповторных ДНФ и КНФ булевых формул

#### 2.5.1.1. Вычисление конъюнктивной нормальной формы формул на структуре для дизъюнктивной нормальной формы

В работах [16, 18] представлен МЛМ, обеспечивающий вычисление неповторных ДНФ функций алгебры логики. МЛМ (рис. 2.5, а) описывается формулами вида

$$\varepsilon_{i-1} = A_{i-1}\varepsilon_{i-2} \vee T_{i-1}; \quad (2.2)$$

$$\varphi_i = \varphi_{i-1} \vee \varepsilon_{i-1}A_iT_i, \quad (2.3)$$

где  $A_i$  – логические аргументы, обозначающие информационные входы ячейки;  $T_i$  – логические аргументы, обозначающие настроечные входы ячейки;  $i = 1, 2, \dots, n$ ,  $n$  – число ячеек линейной среды (см. рис. 2.5, б).

Рассматриваемая структура (которую в дальнейшем будем называть  $F$ -структурой) предназначена для реализаций формул, представленных в ДНФ, но для реализации формул, представленных в КНФ, нет необходимости в использовании отдельной схемы, так как схема, предназначенная для ДНФ, может быть использована и для КНФ [3, 5].

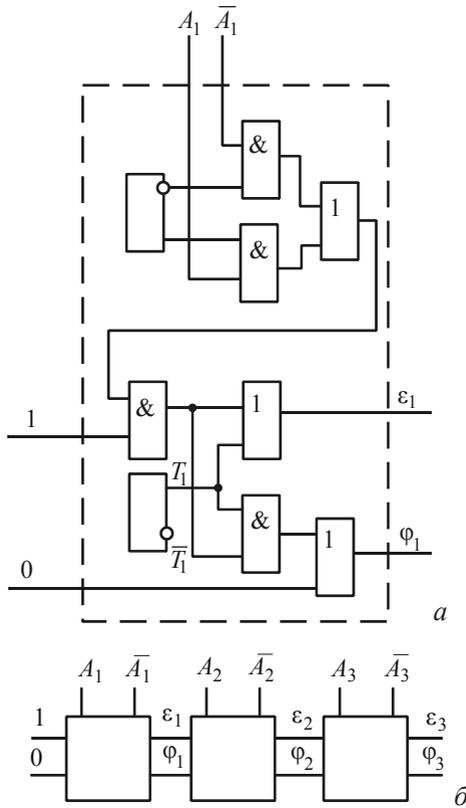
Поставим в соответствие каждому логическому аргументу неповторной упорядоченной формулы двоичный разряд и единицей условимся обозначать последний справа аргумент каждой конъюнкции. Все остальные аргументы будем обозначать нулями. В результате получим  $n$ -разрядное двоичное число, где  $n$  – число аргументов формулы. Условимся называть такое число  $\alpha$ -кодом. Для примера рассмотрим формулу (примем  $n = 7$ ):

$$f = A_1 \vee A_2A_3 \vee A_4A_5A_6.$$

Ее  $\alpha$ -код имеет вид

$$\alpha = 1010010.$$

Очевидно, что всякой булевой формуле, представленной в ДНФ и имеющей не более  $n$  вхождений аргументов, можно поставить в соответствие некоторый  $\alpha$ -код. При этом одной и той же формуле могут соответствовать различные  $\alpha$ -коды в зависимости от расположения в этой формуле конъюнкций различной длины [7].

Рис. 2.5. Ячейка  $F$ -структуры

Чтобы отличать инверсные аргументы от неинверсных, каждому разряду  $\alpha$ -кода поставим в соответствие еще один двоичный разряд. Если аргумент находится в неинверсной форме, то условимся его обозначать нулем, если в инверсной – единицей. Полученное при этом  $n$ -разрядное двоичное число будем называть  $\beta$ -кодом. Например, для формулы ( $n = 7$ )

$$f = A_1 A_2 \vee \bar{A}_3 A_4 \vee \bar{A}_5 \bar{A}_6$$

$\beta$ -код имеет вид

$$\beta = 0010110.$$

Два кода ( $\alpha$  и  $\beta$ ) полностью определяют структуру неповторной упорядоченной булевой формулы, представленной в ДНФ и содержащей инверсные и неинверсные аргументы.

Для реализации конъюнктивных форм формул достаточно записать  $\gamma$ -код точно так же, как и  $\alpha$ -код, с той лишь разницей, что группа нулей с примыкающей к ним справа единицей будет соответствовать дизъюнкции, записанной в скобках. Например, для формулы

$$f = (A_1 \vee A_2 \vee \bar{A}_3 \vee \bar{A}_4)(A_5 \vee \bar{A}_6 \vee A_7)(\bar{A}_8 \vee A_9)A_{10}$$

$\gamma$ -код представится в виде

$$\gamma = 0001001011.$$

Если этот код ввести в  $F$ -структуру, то будет реализована формула

$$f_1 = A_1A_2A_3A_4 \vee A_5A_6A_7 \vee A_8A_9 \vee A_{10},$$

в которой все аргументы не содержат инверсий. Чтобы учесть инверсии, достаточно записать  $\beta$ -код и проинвертировать его:

$$\bar{\beta} = 1100101011.$$

С учетом  $\bar{\beta}$ -кода  $F$ -структура обеспечит реализацию формулы

$$f_2 = \bar{A}_1\bar{A}_2A_3A_4 \vee \bar{A}_5A_6\bar{A}_7 \vee A_8\bar{A}_9 \vee \bar{A}_{10}.$$

Очевидно, что  $f_2 = \bar{f}$ .

Таким образом, чтобы с помощью  $F$ -структуры реализовать конъюнктивную форму формулы, достаточно:

- 1) найти  $\gamma$ -код;
- 2) найти  $\bar{\beta}$ -код;
- 3) проинвертировать сигнал, снимаемый с выхода  $F$ -структуры.

В справедливости этого утверждения нетрудно убедиться, если учесть, что в случае конъюнктивных форм определение  $\gamma$ -кода соответствует операции замены знаков дизъюнкции на знаки конъюнкции и наоборот, а инвертирование  $\beta$ -кода соответствует замене инверсных аргументов неинверсными и наоборот. Этими двумя операциями реализуются правила инвертирования формул по теореме де Моргана.

Таким образом, с помощью одной и той же ячеистой структуры можно реализовать и дизъюнктивные, и конъюнктивные нормальные формы неповторных упорядоченных булевых формул.

## 2.5.1.2. Анализ работы модуля

Работа структуры рассматривается в предположении равной доступности прямых и инверсных выходов источников информации при помощи  $\beta$ -кода.

Исследуем возможности  $F$ -структуры в предположении, что настроечными являются входы  $A_i$ , а информационными – входы  $T_i$  ( $i = 1, 2, \dots, n$ ), где  $n$  – число ячеек изотропной структуры. Подставим формулу (2.2) в (2.3) и получим

$$\varphi_i = \varphi_{i-1} \vee (A_{i-1} \varepsilon_{i-2} \vee T_{i-1}) A_i T_i = \varphi_{i-1} \vee A_{i-1} T_i A_i \varepsilon_{i-2} \vee T_{i-1} T_i A_i.$$

Здесь  $\varepsilon_{i-2}$  состоит из конъюнкций, в каждую из которых  $T$ -аргументы входят не более одного раза. Следовательно, в ДНФ выражения  $\varphi_i$ , записанного через  $A$ - и  $T$ -аргументы, нет ни одной конъюнкции, содержащей более двух  $T$ -аргументов. Представим формулу (2.3) в ДНФ и выразим ее через  $A$ - и  $T$ -аргументы, например для  $n = 8$ :

$$\begin{aligned} \varphi_i = & A_1 T_1 \vee A_1 A_2 T_2 \vee A_2 T_1 T_2 \vee A_1 A_2 A_3 T_3 \vee A_1 A_2 T_1 T_3 \vee \dots \vee T_i \bigwedge_{i=1}^n A_i \vee \\ & \vee T_1 T_8 \bigwedge_{i=2}^n A_i \vee T_2 T_8 \bigwedge_{i=3}^n A_i \vee \dots \vee T_7 T_8 A_8. \end{aligned}$$

Эта формула состоит из дизъюнкции конъюнкций двух видов:

а) конъюнкций, содержащих только один  $T$ -аргумент, вида

$$T_h \bigwedge_{i=1}^h A_i, \quad h = 1, 2, \dots, n.$$

Всего их существует  $n$ ;

б) конъюнкций, содержащих только два различных  $T$ -аргумента; всего их  $C_n^2$ . При этом каждая конъюнкция, содержащая  $\lambda$   $A$ -аргументов, имеет вид

$$T_h T_{h+\lambda} \bigwedge_{i=1}^{h+\lambda} A_{h+i},$$

т.е. индексы  $A$ -аргументов образуют натуральную последовательность, начиная с  $h+1$  и заканчивая  $h+\lambda$ .

Таким образом, если  $A$ -аргументы использовать в качестве настроечных, а  $T$ -аргументы считать логическими независимыми переменными, то с помощью  $F$ -структуры невозможно реализовать формулы, ми-

нимальные ДНФ которых содержат хотя бы одну конъюнкцию, охватывающую более двух аргументов.

Выясним, как по настроечному коду, поданному на  $A$ -входы, определить вид формулы, реализуемой  $F$ -структурой. Для этого сначала введем понятие полной группы единиц настроечного кода. Группу единиц в настроечном коде будем называть полной в следующих случаях:

а) если группа единиц, между которыми нет нулей, ограничена с обеих сторон нулями;

б) если группа единиц, между которыми нет нулей, находится либо в начале кода, либо в его конце.

Пр и м е р. Настроечный код вида

1 1 1 0 0 0 1 0 0 1 1 0 0 1 1 1 1

содержит четыре полные группы единиц. Первая из них содержит три единицы и находится в начале кода, вторая состоит из одной единицы, третья – из двух и четвертая – из расположенных в конце кода четырех единиц.

Длиной полной группы единиц условимся называть число образующих ее единиц. Для вышеприведенного кода длины полных групп есть соответственно числа 3, 1, 2, 4.

**Т е о р е м а 2.1.** Если настроечный код, подаваемый на  $A$ -входы  $F$ -структуры, состоит из  $r$  групп двоичных знаков, каждая из которых содержит полную группу единиц и один примыкающий к ним слева нуль, то  $F$ -структура реализует формулу

$$\begin{aligned} f_i &= S_{2,3,\dots,n_1}(T \in N_1) \vee S_{2,3,\dots,n_2}(T \in N_2) \vee \dots \vee S_{2,3,\dots,n_r}(T \in N_r) = \\ &= \bigvee_{i=1}^r S_{2,3,\dots,r_i}(T \in N_i), \end{aligned} \quad (2.4)$$

где  $n_i$  – число двоичных знаков, образованных полной группой единиц и примыкающим к ним слева нулем;  $N_i$  – множество  $T$ -аргументов, соответствующих  $i$ -й полной группе единиц и примыкающему к ней слева нулю,  $S_{2,3,\dots,r_i}$  – симметрическая реализация БФУ [6] соответствующих аргументов.

**Д о к а з а т е л ь с т в о.** Пусть дан некоторый настроечный код. Выделим в нем какую-либо полную группу единиц с примыкающим к ней слева нулем. Будем считать, что этому нулю соответствует аргу-

мент  $A_h$ , а единицам – аргументы  $A_{h+1}, A_{h+2}, \dots, A_{h+\lambda}$ , где  $\lambda$  – длина полной группы ( $h, h + \lambda \in \{1, 2, \dots, n\}$ ). Тогда  $A_h = 0$ ;

$$A_{h+1} = A_{h+2} = \dots = A_{h+\lambda} = 1.$$

Подставим значения этих аргументов в формулу  $\varphi_i$ . Тогда конъюнкции формулы  $\varphi_i$ , содержащие одиночные  $A$ -аргументы (кроме  $T$ -аргументов), дадут  $\lambda$  различных двухсимвольных конъюнкций  $T$ -аргументов. Эти  $\lambda$  конъюнкции войдут в искомую формулу  $\varphi_i$ . Кроме того, в формуле  $\varphi_i$  существует  $C_\lambda^2$  конъюнкций, включающих в себя различные комбинации  $A$ -аргументов группы  $\lambda$ . При подстановке единиц вместо этих аргументов получим еще  $C_\lambda^2$  различных двухсимвольных конъюнкций  $T$ -аргументов, которые также войдут в формулу  $f_i$ . Всего данной полной группе единиц настроечного кода соответствует  $P$  двухсимвольных конъюнкций  $T$ -аргументов, входящих в искомую формулу  $f_i$ , где

$$P = \lambda \vee C_\lambda^2.$$

Все эти  $P$  конъюнкций являются различными и образуют их  $\lambda + 1$  аргументов  $T_h, T_{h+1}, \dots, T_{h+\lambda}$ .

Из [1] известно, что число выражений вида

$$S_{2,3,\dots,\lambda+1}(T_h, T_{h+1}, \dots, T_{h+\lambda})$$

равно  $Q$  где  $Q = C_{\lambda+1}^2$ .

Поскольку  $P = Q$ , то множество двухсимвольных конъюнкций, полученных на основе полной группы настроечного кода, равно множеству всех конъюнкций симметрической реализации БФу тех же аргументов, представленной в минимальной ДНФ. Так как и  $A$ -, и  $T$ -аргументы, входящие в различные полные группы единиц настроечного кода, не повторяются, то каждой полной группе единиц соответствует определенная симметрическая реализация БФу соответствующих аргументов. Пусть  $\lambda_i$  – длина  $i$ -й полной группы единиц ( $i = 1, 2, \dots, r$ ), тогда

$$n_i = \lambda_i + 1$$

и формула, которая реализуется  $F$ -структурой при помощи настроечного кода, содержащего  $r$  полных групп единиц, представится в виде

$$\begin{aligned} f_i &= S_{2,3,\dots,n_1}(T \in N_1) \vee S_{2,3,\dots,n_2}(T \in N_2) \vee \dots \vee S_{2,3,\dots,n_r}(T \in N_r) = \\ &= \bigvee_{i=1}^r S_{2,3,\dots,r_i}(T \in N_i), \end{aligned}$$

что и требовалось доказать.

**С л е д с т в и е 1.** Если полные группы единиц отделены одна от другой  $K$  нулями, то в минимальной ДНФ формулы  $f_i$  отсутствуют  $K - 1$   $T$ -аргументов. Справедливость этого следствия вытекает из того, что  $T$ -аргументы, образующие каждую симметрическую реализацию БФу выражения (2.4), определяются только единицами настроечного кода и одиночными нулями, примыкающими слева к каждой полной группе единиц.  $T$ -аргументы, соответствующие остальным нулям, в формулу  $f_i$  не входят.

**С л е д с т в и е 2.** Если настроечный код оканчивается  $m$  нулями, то последние  $m$   $T$ -аргументов в формулу  $f_i$  не войдут.

**С л е д с т в и е 3.** Если настроечный код начинается с единицы, то полной первой группе, насчитывающей  $\lambda$  единиц, соответствует дизъюнкция  $T$ -аргументов, входящая в формулу  $f_i$ :

$$f_i = \bigvee_{i=1}^{\lambda} T_i \vee \xi,$$

где  $\xi$  – дизъюнкция симметрических реализаций БФу  $T$ -аргументов, соответствующих остальным полным группам единиц настроечного кода.

Для доказательства этого следствия достаточно заметить, что если настроечный код начинается с единицы, то

$$A_1 = A_2 = \dots = A_{\lambda},$$

где  $\lambda$  – длина первой полной группы единиц. Тогда

$$f_i = T_1 \vee T_2 \vee \dots \vee T_{\lambda} \vee \xi = \bigvee_{i=1}^{\lambda} T_i \vee \xi,$$

что и требовалось доказать.

**С л е д с т в и е 4.** Если  $K_1$  и  $K_2$  – различные настроечные коды, то соответствующие им формулы, реализуемые  $F$ -структурой, на  $A$ -входы которой подаются коды  $K_1$  и  $K_2$ , не совпадают, что следует из выражения (2.4), если учесть, что

$$N_i \cap N_j = 0,$$

где  $i \neq j$ ;  $i, j = 1, 2, \dots, r$ .

**П р и м е р.** Найти формулу, которую реализует  $F$ -структура на основе кода 0 1 0 0 1 1 1 0, поданного на  $A$ -входы этой структуры.

Согласно записи кода в нем имеется две полные группы единиц. Первая состоит из одной единицы. Добавив к ней нуль, расположенный слева, находим, что в искомую формулу  $f_8$  войдет:

$$S_2(T_1T_2) = T_1T_2.$$

Вторая полная группа состоит из трех единиц, которым соответствуют аргументы  $T_5, T_6, T_7$ . С учетом нуля, расположенного левее полной группы, находим, что в формулу  $f_8$  войдет:

$$S_{2,3,4}(T_4, T_5, T_6, T_7) = T_4T_5 \vee T_4T_6 \vee T_4T_7 \vee T_5T_6 \vee T_5T_7 \vee T_6T_7.$$

Других групп единиц в настроечном коде нет, следовательно, искомая формула  $f_8$  примет вид

$$f_8 = T_1T_2 \vee T_4T_5 \vee T_4T_6 \vee T_4T_7 \vee T_5T_6 \vee T_5T_7 \vee T_6T_7.$$

Полные группы единиц отделены двумя нулями. Согласно следствию 1 один  $T$ -аргумент в формуле  $f_8$  отсутствует. Это аргумент  $T_3$ , соответствующий второму нулю, расположенному слева от второй полной группы единиц. Настроечный код оканчивается нулем. Согласно следствию 2 в формуле  $f_8$  отсутствует аргумент  $T_8$ .

Таким образом, теорема 2.1 и следствия позволяют по настроечному коду найти формулу, которую реализует  $F$ -структура, если на ее  $A$ -входы подать этот настроечный код. При этом, согласно следствию 4, всякому настроечному коду соответствует единственная  $f$  формула.

### 2.5.1.3. Нахождение настроечных кодов по виду булевой формулы

С практической точки зрения наибольший интерес представляет вопрос о том, как найти настроечный код по заданной булевой формуле.

Очевидно, что если формула  $f$  представлена в ДНФ, не содержит повторяющихся аргументов и пропусков аргументов, то она реализуется  $F$ -структурой. Если настроечный код подается на  $T$ -входы, то будем говорить, что речь идет о прямом варианте реализации формулы. Если же настроечный код подается на  $A$ -входы, то условимся называть это обратным вариантом реализации.

Относительно всякой формулы имеют место следующие вопросы.

1. Если формула  $f$  не входит в множество  $N_1$ , рассмотренное в разделе 2.4, то входит ли она в множество  $M_1$ ?

2. Если на вопрос 1 дан положительный ответ, то как найти настроечный код?

Для ответа на первый вопрос воспользуемся доказанной теоремой 2.1. Непосредственно из формулы (2.4) следует, что всякая формула реализуется  $F$ -структурой в обратном варианте, если эта формула представима в виде дизъюнкции симметрических реализаций БФу, минимальные ДНФ которых содержат конъюнкции не более чем по два аргумента, и если множества аргументов, входящих в различные симметрические реализации БФу, не пересекаются.

Пример 1. Формула

$$f = A \vee B \vee CD \vee CE \vee DE \quad (2.5)$$

реализуется в обратном варианте (и не реализуется в прямом, поскольку в ней есть повторяющиеся аргументы), так как можно записать

$$f = A \vee B \vee S_{2,3}(C, D, E) = S_{1,2}(A, B) \vee S_{2,3}(C, D, E).$$

Пример 2. Формула

$$f = A \vee B \vee CD \vee CE \vee CF \vee DE \vee DF$$

не реализуется  $F$ -структурой ни в прямом, ни в обратном вариантах, поскольку она не сводится к дизъюнкции симметрических реализаций БФу, множества аргументов которых не пересекаются.

Ответ на второй вопрос не является однозначным. Если известно, что формула реализуема в обратном варианте, то можно найти несколько различных настроечных кодов, каждый из которых можно использовать для настройки  $F$ -структуры на заданную формулу. Эта неодно-

значность нахождения кодов обусловлена следствием 1, т.е. в зависимости от того, какому входу  $F$ -структуры поставлен в соответствие тот или иной  $T$ -аргумент, настроечный код примет тот или иной вид. Если соответствие между входами  $F$ -структуры и аргументами заданной формулы зафиксировано, то настроечный код находится однозначно. Рассмотрим, например, формулу (2.5). Пусть соответствие между ее аргументами имеет вид

$$\begin{array}{ccccc} A & B & C & D & E \\ 1 & 2 & 3 & 4 & 5, \end{array}$$

где цифры являются номерами  $T$ -входов. Тогда для 8-входовой  $F$ -структуры получаем настроечный код 11011000. Выберем другое соответствие:

$$\begin{array}{ccccc} A & B & C & D & E \\ 1 & 2 & 5 & 6 & 7, \end{array}$$

получим настроечный код 11000110.

Рассмотрим еще пример. Найти настроечный код для формулы

$$f = AB \vee CD \vee FK \vee LM$$

при условии, что аргумент  $E$  является пропущенным.

Хотя формула представлена в ДНФ и является неповторной, реализовать ее в прямом варианте невозможно из-за пропуска аргумента  $E$ . Установим соответствие для 10-входовой  $F$ -структуры:

$$\begin{array}{cccccccccc} A & B & C & D & E & F & K & L & M & N \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10. \end{array}$$

Настроечный код имеет вид 01010001010, которому соответствует формула, выраженная через  $T$ -аргументы:

$$f_{10} = T_1 T_2 \vee T_3 T_4 \vee T_6 T_7 \vee T_8 T_9,$$

где отсутствует аргумент  $T_5$ .

Таким образом, благодаря обратному варианту реализации формул возможности  $F$ -структуры значительно расширились.

**2.5.2. Вычисление неповторных булевых формул с пропусками аргументов**

*2.5.2.1. Операция удаления аргументов из булевой формулы*

Рассмотренный в разделе 2.5.1 МЛМ не обеспечивает реализацию неповторных упорядоченных булевых формул классов 3, 4, 5, 6 из приведенной классификации.

Рассмотрим цепь ячеек, приведенных на рис. 2.6, которая представляет собой линейную изотропную (однородную) среду. Так как за каждой ячейкой изотропной среды закреплен вполне определенный логический аргумент, возникает вопрос, что значит не учитывать тот или иной аргумент при вычислении заданной формулы?

Например, формула

$$f = A_1 \vee A_3 A_4$$

не зависит от аргумента  $A_2$ . Схемно это будем осуществлять путем электрического соединения входов и выходов ячейки  $A_2$ , как показано пунктиром на рис. 2.6.

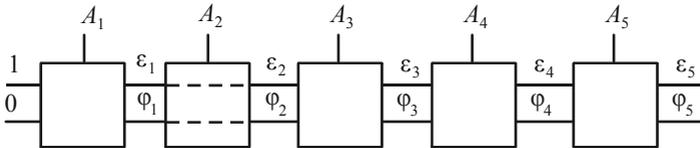


Рис. 2.6. Линейная изотропная среда

Математически такое соединение ячеек соответствует некоторой операции, которая в случае неповторной формулы сводится к подстановке в нее нуля или единицы вместо удаляемого аргумента. Пусть, например, при  $n = 5$   $F$ -структура настроена на формулу

$$f_1 = A_1 A_2 \vee A_3 A_4 \vee A_5.$$

Если принять  $A_3 = 1$ , то получим новую формулу

$$f'_1 = A_1 A_2 \vee A_4 \vee A_5,$$

на  $F$ -структуре ее можно реализовать лишь в том случае, если в нее ввести код 01011 и удалить ячейку 3.

Рассмотрим другую формулу:

$$f_2 = A_1 A_2 \vee A_3 \vee A_4 A_5.$$

Если по аналогии с формулой  $f_1$  в формулу  $f_2$  подставить  $A_3 = 1$ , чтобы удалить аргумент  $A_3$ , то получим

$$f_2' = A_1 A_2 \vee 1 \vee A_4 A_5 \equiv 1$$

вместо ожидаемого результата

$$f_2'' = A_1 A_2 \vee A_4 A_5,$$

который получился бы при вводе в  $F$ -структуру кода 01101 с последующим удалением ячейки 3.

Отсюда следует, что для неповторных упорядоченных формул операции удаления аргумента соответствует подстановка вместо него единицы, если он входит в конъюнкцию с другими аргументами, и нуля, если он входит в дизъюнкцию. Однако это справедливо лишь для формул, не содержащих инверсных аргументов. В последнем случае, прежде чем выполнять подстановку, необходимо выяснить, не является ли инверсным удаляемый аргумент [19]. Например, чтобы удалить аргумент  $A_2$  в формуле

$$f = A_1 A_2 \vee A_3,$$

необходимо выполнить подстановку  $A_2 = 1$ . Чтобы удалить тот же аргумент в формуле

$$f = A_1 \bar{A}_2 \vee A_3,$$

необходима подстановка  $A_2 = 0$ .

В общем случае, когда формула содержит повторные аргументы, принцип подстановки не применим. Поясним это на примере формулы

$$f = A_1 A_2 \vee A_2 A_3. \quad (2.6)$$

Чтобы из (2.6) получить формулу

$$f = A_1 A_2 \vee A_3, \quad (2.7)$$

необходимо вместо второго вхождения аргумента  $A_2$  подставить единицу, а первое вхождение оставить без изменения. Очевидно, что

подстановкой в формулу (2.6)  $A_2 = 1$  выражение (2.7) получить невозможно. При  $A_2 = 0$  получаем  $f \equiv 0$ .

В дальнейшем будем полагать, что операции удаления аргумента соответствует не подстановка вместо него какого-либо значения, а его *вычеркивание* без изменения всех логических операций, связывающих аргументы функции, при этом инверсные аргументы удаляются вместе со знаками отрицания. Если аргумент входит в формулу многократно, то вычеркиванию подлежит не сам аргумент, а его вхождение в формулу.

Например, формула

$$f = A_1 A_2 A_3 \vee A_2 A_4 \vee A_2 A_5$$

имеет семь вхождений аргументов. Чтобы из нее получить формулу

$$f = A_1 A_2 A_3 \vee A_4 \vee A_5,$$

необходимо удалить второе и третье вхождение аргумента  $A_2$ , а первое оставить без изменения.

В случае неповторных упорядоченных формул аргумент и его вхождение неразличимы.

Таким образом, операцию вычеркивания аргумента реализует принцип удаления ячейки, основанный на создании полной прямой проводимости между входами удаляемой ячейки и соответствующими ее выходами.

### 2.5.2.2. Реализация операции удаления аргумента из формулы

Структура рассматриваемого МЛМ (рис. 2.7) описывается системой вида [10]:

$$\begin{cases} \varepsilon_i = (A_i \varepsilon_{i-1} \vee T_i) \bar{S}_i \vee \varepsilon_{i-1} S_i, \\ \varphi_i = A_i \varepsilon_{i-1} T_i \bar{S}_i \vee \varphi_{i-1}. \end{cases} \quad (2.8)$$

Если на ее основе построить линейную среду ( $S$ -структуру), то получим возможность вычислять неповторные булевы формулы, аргументы которой упорядочены, но некоторые из них отсутствуют, например:

$$f_1 = A_1 A_2 \vee A_4 A_6.$$

Здесь аргументы  $A_3$  и  $A_5$  удалены, однако в  $\alpha$ -коде их необходимо учесть так, как будто они входят в формулу, т.е.

$$\alpha = \begin{array}{cccccc} 0 & 1 & 0 & 0 & 0 & 1 \\ A_1 & A_2 & A_3 & A_4 & A_5 & A_6. \end{array}$$

Код, содержащий информацию об удаленных аргументах и задаваемый с помощью  $S$ -триггеров, будем называть  $S$ -кодом. В данном случае  $S$ -код имеет вид

$$S = \begin{array}{cccccc} 0 & 0 & 1 & 0 & 1 & 0 \\ A_1 & A_2 & A_3 & A_4 & A_5 & A_6, \end{array}$$

где единицами обозначены отсутствующие аргументы.

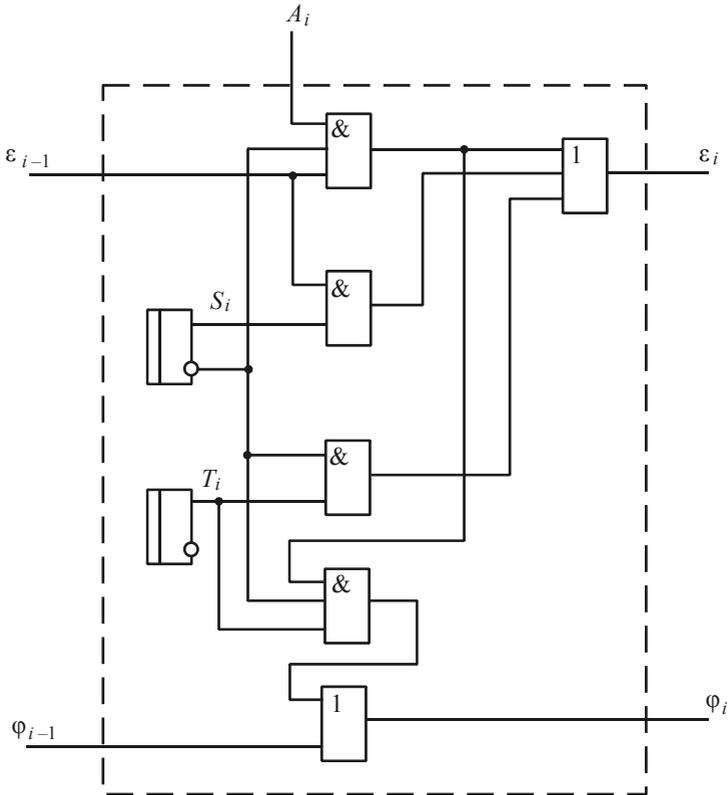


Рис. 2.7. МЛМ  $S$ -структуры

Если формула неповторна, но аргументы в ее записи не упорядочены, то вычислить ее значение с помощью  $S$ -структуры не всегда воз-

можно. Например, формулу  $f = A_5 A_4 \vee A_2 A_1 \vee A_6$  можно вычислить, если записать ее в виде  $f = A_1 A_2 \vee A_4 A_5 \vee A_6$ , соответствующие  $\alpha$ - и  $S$ -коды имеют вид  $\alpha = 010011$ ,  $S = 001000$ . Формулу  $f = A_1 A_3 \vee A_2$  вычислить невозможно [14].

Сформулируем *правило*, в соответствии с которым можно было бы определить вычислимость формулы.

Пусть формула представлена неповторной ДНФ, аргументы которой не упорядочены по вхождению. Находим аргумент с наименьшим индексом и записываем всю конъюнкцию, куда входит этот аргумент, предварительно расставив аргументы в конъюнкции в порядке возрастания: слева направо. Последний аргумент будет иметь индекс  $t$ . Если в оставшейся части формулы найдется хотя бы один аргумент с индексом, меньшим  $t$ , то формула не вычислима  $S$ -структурой. Если же таких нет, то в оставшейся части формулы находим аргумент с наименьшим индексом и выписываем вторую конъюнкцию так же, как и в первом случае.

Таким образом, с помощью  $S$ -структуры осуществляется вычисление неповторных упорядоченных булевых формул, представленных в ДНФ или КНФ как с пропусками аргументов, так и без них.

### 2.5.2.3. О неоднозначности записи настроечных кодов в $S$ -структуре

Рассмотрим формулу

$$f = A_1 A_2 \vee A_4 A_6, \quad (2.9)$$

ее настроечные коды имеют вид

$$\alpha_1 = 010001,$$

$$S = 001010.$$

Поскольку аргументы  $A_3$  и  $A_5$  удалены, то формально  $\alpha$ -код можно записать в виде двух других вариантов:

$$\alpha_2 = 001001,$$

$$\alpha_3 = 011001.$$

Эта неоднозначность записи  $\alpha$ -кодов обусловлена тем, что формула (2.9) может быть получена путем вычеркивания аргументов  $A_3$  и  $A_5$  из следующих трех различных формул:

$$f_1 = A_1 A_2 \vee A_3 A_4 A_5 A_6,$$

$$f_2 = A_1 A_2 A_3 \vee A_4 A_5 A_6,$$

$$f_3 = A_1 A_2 \vee A_3 \vee A_4 A_5 A_6.$$

Однако применительно к  $S$ -средам это справедливо лишь для формул  $f_1$  и  $f_3$ . Если аргументы  $A_3$  и  $A_5$  вычеркнуть из формулы  $f_2$ , то формально получается формула (2.9), а при вводе в  $S$ -структуру настроечных кодов

$$\alpha_2 = 001001,$$

$$S = 001010$$

будет реализовываться

$$f = A_1 A_2 A_3 A_6,$$

не совпадающая с выражением (2.9).

Пусть  $\Psi$  – формула, которая получилась на основе некоторой формулы  $f$  путем вычеркивания из нее определенных аргументов;  $\Psi_2$  – формула, реализуемая  $S$ -структурой, настроенной  $\alpha$ - и  $S$ -кодами, где  $\alpha$ -код записан на основе формулы  $f$ , а  $S$ -код отражает отсутствующие аргументы. Выясним, в каких случаях

$$\Psi_1 \neq \Psi_2.$$

Ответ на этот вопрос дают следующие теоремы.

**Т е о р е м а 2.2.** Если в записи некоторой неповторной упорядоченной булевой формулы  $f$ , представленной в виде дизъюнкции  $k$  конъюнкций, пронумерованных в последовательности  $1, 2, \dots, k$ :

$$f = \tau_1 \vee \tau_2 \vee \dots \vee \tau_k,$$

удалить последний аргумент в  $i$ -й конъюнкции, где  $i = 1, 2, \dots, k - 1$ , и записать  $S$ -код, то при вводе в  $S$ -структуру этого  $S$ -кода и  $\alpha$ -кода, записанного на основе формулы  $f$ ,  $S$ -структура настроится на формулу  $\Psi$ , отличающуюся от формулы  $f$  тем, что в ней вместо дизъюнкции  $\tau_{j-1} \vee \tau_j$  ( $j = 1, 2, \dots, k$ ) будет записана конъюнкция  $t_{j-1} \tau_j$ , где  $t_{j-1}$  – конъюнкция, полученная на основе конъюнкции  $\tau_{j-1}$  путем удаления последнего аргумента.

**Д о к а з а т е л ь с т в о.** Пусть задана неповторная упорядоченная булева формула  $n$  аргументов

$$f = \tau_1 \vee \tau_2 \vee \dots \vee \tau_k,$$

где

$$\begin{aligned} \tau_1 &= A_1 A_2 \dots A_{S_1-1} A_{S_1}, \\ \tau_2 &= A_{S_1+1} A_{S_1+2} \dots A_{S_2-1} A_{S_2}, \\ \tau_3 &= A_{S_2+1} A_{S_2+2} \dots A_{S_3-1} A_{S_3}, \\ &\dots \\ \tau_k &= A_{S_{k-1}+1} A_{S_{k-1}+2} \dots A_{S_k-1} A_{S_k}. \end{aligned}$$

Найдем для нее  $\alpha$ -код. Затем удалим последний аргумент в  $i$ -й конъюнкции и определим соответствующий  $S$ -код. Оба найденные кода введем в  $S$ -структуру. Благодаря  $S$ -коду ячейка  $A_{S_i}$  будет исключена.

Выходы ячейки с номером  $S_{i-1}$  соединятся с соответствующими входами ячейки с номером  $S_{i+1}$ . В разрядах  $\alpha$ -кода с указанными номерами содержатся нули, обозначающие операцию конъюнкции соответствующих аргументов. Следовательно, вместо дизъюнкции вида  $\tau_{j-1} \vee \tau_j$  в формуле  $f$  окажется конъюнкция  $t_{j-1} \tau_j$ , где

$$t_{j-1} = A_{S_{j-1}+1} A_{S_{j-1}+2} \dots A_{S_{j-1}},$$

что и требовалось доказать.

**Т е о р е м а 2.3.** Если формула представлена единственной конъюнкцией, то она всегда вычислима и имеет  $2^r$  вариантов  $\alpha$ -кода, где  $r$  – число единиц в  $S$ -коде, оканчивающемся нулем.

**Д о к а з а т е л ь с т в о.** Пусть задана формула

$$f = A_1 A_2 \dots A_{k-1} A_{k+1} A_{k+2} \dots A_n,$$

где  $n > k$ . Если аргументы формулы не упорядочены, то, расположив их в порядке возрастания индексов, получим вычислимую запись. Ее  $S$ -код имеет вид

$$\underbrace{000\dots 01}_{k-1 \text{ нулей}} \underbrace{00\dots 0}_{n-k \text{ нулей}},$$

$\alpha$ -код формулы представится в виде

$$\alpha = \underbrace{00000\dots 001}_{n-1 \text{ нулей}}.$$

Согласно теореме 2.1  $\alpha$ -код можно записать и в виде

$$\alpha = \underbrace{000\dots 01}_{k-1 \text{ нулей}} \underbrace{1000\dots 01}_{n-k \text{ нулей}},$$

что соответствует выражению

$$f^n = A_1 A_2 \dots A_k \vee A_{k+1} \dots A_n,$$

где аргумент  $A_k$ , являющийся последним в первой конъюнкции, удален.

Все сказанное справедливо для любой ячейки, следовательно, если в  $S$ -коде содержится  $k$  единиц, причем  $S$ -код оканчивается нулем, то  $r$  триггеров, не влияющих на значение формулы, могут находиться в одном из  $2^r$  состояний, что и требовалось доказать.

**Т е о р е м а 2.4.** Если  $S$ -код совпадает с  $\alpha$ -кодом, то ячеистая структура реализует формулу  $f \equiv 0$ .

**Д о к а з а т е л ь с т в о.** Если длина ячеистой структуры равна единице, то формула, описывающая ее выход, имеет вид

$$\varphi_1 = A_1 T_1 \bar{S}.$$

Это выражение можно получить из (2.8), если учесть, что  $\varepsilon_0 = 1$  и  $\varphi_0 = 1$ . Очевидно, что при  $S_1 = T_1$  имеем  $\varphi_1 \equiv 0$ .

Пусть длина структуры равна двум ячейкам. Выход ее представится в виде

$$\varphi_2 = A_2 \varepsilon_1 T_2 \bar{S}_2 \vee \varphi_1.$$

При  $T_2 = S_2$  первая конъюнкция равна нулю, а при  $T_1 = S_1$ , как было показано выше,  $\varphi_1 \equiv 0$ , следовательно,  $\varphi_2 \equiv 0$ .

Рассуждая аналогично, замечаем, что в формуле

$$\varphi_i = A_i \varepsilon_{i-1} T_i \bar{S}_i \vee \varphi_{i-1},$$

где  $i = 1, 2, \dots, n$ ,  $n$  – число ячеек, первая конъюнкция равна нулю, так как  $T_i = S_i$ , а вторая равна нулю по причине  $T_j = S_j$  ( $j = 1, 2, \dots, i-1$ ). Следовательно, если  $\alpha$ -код и  $S$ -код равны, то структура реализует формулу, тождественно равную нулю, что и требовалось доказать.

**С л е д с т в и е и з т е о р е м ы 2.4.** Если  $n$  – длина ячеистой структуры, то формула  $f \equiv 0$  может быть представлена  $2^n$  различными  $\alpha$ -кодами и  $2^n$   $S$ -кодами при равенстве  $S$ - и  $\alpha$ -кодов. Справедливость этого утверждения следует из того, что всего существует  $2^n$   $n$ -разрядных двоичных чисел, каждое из которых может быть  $\alpha$ - и  $S$ -кодами.

Формулу  $f \equiv 0$  в общем случае можно задать многими способами. Кроме уже указанного случая, когда  $\alpha = S$ , существуют и другие варианты. Например, если  $\alpha = 0$ , т.е.  $\alpha$ -код не содержит единиц, то структура реализует формулу  $f \equiv 0$  независимо от  $S$ -кода. Это значит, что при

$\alpha = 0$  существует  $2^n$  вариантов настройки  $S$ -структуры на формулу  $f \equiv 0$ , образуемых путем изменения  $S$ -кода. Если  $S$ -код не содержит нулей, то  $S$ -структура также реализует формулу  $f \equiv 0$   $2^n$  способами, каждому из которых соответствует определенный  $\alpha$ -код. Справедливость этого утверждения следует из того, что при  $S$ -коде, не содержащем нулей, выход структуры соединен с входом первой ячейки, на который постоянно подан нулевой уровень напряжения. Общий случай распознавания формул  $f \equiv 0$  по виду настроечного кода освещен в следующем разделе.

#### 2.5.2.4. Распознавание формулы константа нуль по виду настроечных кодов

В общем случае варианты настройки  $S$ -структуры на формулу  $f \equiv 0$  определяются нижеприведенной теоремой. Но сначала введем некоторые понятия. Формула, описывающая  $S$ -структуру, зависит от служебных аргументов  $T_i$  и  $S_i$ , каждому из которых соответствует отдельный триггер. Будем рассматривать эти триггеры как двухразрядный двоичный регистр, где  $T_i$  соответствует старшему разряду, а  $S_i$  – младшему. В каждом регистре может быть записано одно из четырех чисел 00, 01, 10, 11, которые рассмотрим как цифры четверичной системы счисления 0, 1, 2, 3. Тогда каждому  $n$ -разрядному четверичному числу будет соответствовать некоторая булева формула, на которую настроится  $S$ -структура, и всякий настроечный код можно представить в четверичной системе.

**Т е о р е м а 2.5.** Если в настроечном коде отсутствует цифра 2, то структура реализует формулу  $f \equiv 0$ .

**Д о к а з а т е л ь с т в о.** Пусть дана структура, настроечный код которой содержит цифры 0, 1, 3 и не содержит цифру 2. Ячейки, в которые введены цифры 1 и 3, можно удалить, поскольку их входы соединены с соответствующими выходами. Тогда останется структура, настроечный код которой состоит из одних нулей, т.е. коды  $S$  и  $\alpha$  будут равными. Согласно теореме 2.4 такая структура реализует формулу  $f \equiv 0$ , что и требовалось доказать.

**Т е о р е м а 2.6.** Если в настроечный код входит цифра 2, то структура реализует формулу  $f \neq 0$ .

**Д о к а з а т е л ь с т в о.** Пусть дана структура, настроечный код которой содержит все четверичные цифры. Удалим ячейки, содержащие цифры 1 и 3. Тогда останется структура, содержащая настроечные коды

0 и 2. Такая структура совпадает с  $F$ -структурой. Согласно правилам нахождения  $\alpha$ -кодов единица в коде обозначает конец конъюнкции (ее последнюю букву). Следовательно, присутствие единицы в  $\alpha$ -коде говорит о наличии соответствующей конъюнкции в заданной формуле, тождественно не равной нулю.

Таким образом, если в настроечном коде имеется цифра 2, то  $S$ -структура реализует формулу  $f \neq 0$ , что и требовалось доказать.

**С л е д с т в и е 1** из теорем 2.5 и 2.6.  $S$ -структура, состоящая из  $n$  ячеек, реализует  $k$  тождественно равных нулю булевых формул, где  $k = 3^n$ , и  $N$  тождественно не равных нулю булевых формул, где

$$N = 4^n - 3^n. \quad (2.10)$$

Справедливость этих утверждений следует из того, что настроечные коды, приводящие к формуле  $f \equiv 0$ , могут состоять из последовательности длины  $n$ , образованной тремя цифрами 0, 1, 3. Поставим их в соответствие цифрам троичной системы счисления. Тогда каждое  $n$ -разрядное троичное число будет представлять некоторый настроечный код для  $f \equiv 0$ . Количество всех троичных  $n$ -разрядных чисел равно  $3^n$ , следовательно, и  $k = 3^n$ . Всего существует  $4^n$  настроечных кодов, следовательно, число настроечных кодов для  $f \neq 0$  равно  $4^n - 3^n$ , что и требовалось доказать.

**С л е д с т в и е 2.** Если настроечный код состоит только из двоек, то  $S$ -структура реализует дизъюнкцию  $n$ -аргументов.

**С л е д с т в и е 3.** Если настроечный код состоит из  $n - 1$  нулей и оканчивается двойкой, то  $S$ -структура реализует конъюнкцию  $n$ -аргументов.

**С л е д с т в и е 4.** Если в  $i$ -м разряде настроечного кода находится двойка, а в разрядах с номерами  $i + 1, i + 2, \dots, n$  двоек нет, то  $S$ -структура реализует формулу, зависящую не более чем от  $i$  аргументов.

#### 2.5.2.5. Об избыточности в кодировании булевых формул

В данном разделе покажем, что количество неповторных упорядоченных булевых формул, реализуемых  $S$ -структурой, вследствие неоднозначности представления их настроечными кодами значительно меньше числа  $N$ , приведенного в выражении (2.10). Например, при  $n = 2$  имеем

$$N = 4^2 - 3^2 = 7.$$

В приложении 1 (П.1.1) приведены все варианты реализации формул двух аргументов, из которой видно, что среди семи тождественно не равных нулю формул лишь четыре являются различными:

$$\begin{aligned} f_1 &= A_1 A_2; \quad f_3 = A_1; \\ f_2 &= A_2; \quad f_4 = A_1 \vee A_2. \end{aligned}$$

При  $n = 3$  имеем

$$N = 4^3 - 3^3 = 37.$$

В приложении 1 (П.1.2) приведены все 64 настроечных кода. Находим, что существует 13 различных, тождественно не равных нулю формул трех аргументов, которые могут быть реализованы  $S$ -структурой при  $n = 3$ :

$$f_1 = A_1 (200, 201, 203, 210, 230, 211, 213, 231, 233);$$

$$f_2 = A_2 (120, 320, 121, 123, 321, 323);$$

$$f_3 = A_3 (112, 132, 312, 332);$$

$$f_4 = A_1 A_2 (020, 021, 023);$$

$$f_5 = A_1 A_3 (012);$$

$$f_6 = A_2 A_3 (102, 302);$$

$$f_7 = A_1 A_2 A_3 (002);$$

$$f_8 = A_1 \vee A_2 (220, 221, 223);$$

$$f_9 = A_1 \vee A_3 (032, 212, 232);$$

$$f_{10} = A_2 \vee A_3 (122, 322);$$

$$f_{11} = A_1 \vee A_2 \vee A_3 (222);$$

$$f_{12} = A_1 \vee A_2 A_3 (202);$$

$$f_{13} = A_1 A_2 \vee A_3 (022).$$

В скобках перечислены настроечные коды для каждой из 13 формул. Коды представлены в четверичной системе счисления.

Выясним, сколько существует различных формул, которые может реализовать  $S$ -структура, состоящая из  $n$  ячеек, т.е. выведем формулу, связывающую величины  $N$  и  $n$ .

Пусть даны  $n$  аргументов  $A_1, A_2, \dots, A_n$ . Выделим из них группу, содержащую  $i$  аргументов,  $i = 1, 2, \dots, n$ . Количество таких групп равно  $C_n^i$  [2]. Упорядочим аргументы в группе. Чтобы образовать из них формулу, между соседними аргументами необходимо поставить знак дизъюнкции либо конъюнкции.

В последовательности, содержащей  $i$  аргументов, существует  $i - 1$  промежутков, куда можно записывать знаки конъюнкции или дизъюнкции. Следовательно,  $i$  аргументов дадут  $2^{i-1}$  различных формул для одной группы аргументов. А так как число групп  $C_n^i$ , то при  $i$  аргументах получим  $2^{i-1} C_n^i$  соответствующих формул. Поскольку  $i = 1, 2, \dots, n$ , то общее число  $N$  различных функций равно:

$$N = 2^0 C_n^1 + 2^1 C_n^2 + 2^2 C_n^3 + \dots + 2^{n-1} C_n^n = \sum_{i=1}^n 2^{i-1} C_n^i.$$

Полученная формула не только дает количество различных бесповторных булевых формул с упорядоченными аргументами, но и предоставляет возможность записать все такие формулы. Для примера запишем все формулы при  $n = 4$ :

$$N = C_4^1 + 2C_4^2 + 4C_4^3 + 8C_4^4 = 40.$$

Первое слагаемое говорит о том, что соответствующие ему формулы содержат по одному аргументу:

$$f_1 = A_1; f_3 = A_3;$$

$$f_2 = A_2; f_4 = A_4.$$

Второму слагаемому выражения соответствуют формулы, зависящие от двух аргументов. Число выборов двух аргументов из четырех равно  $C_4^2 = 6$ . Два аргумента дают только один промежуток для записи

знаков дизъюнкции и конъюнкции, следовательно, второе слагаемое дает 12 функций:

$$f_5 = A_1 A_2; f_{11} = A_1 \vee A_2;$$

$$f_6 = A_1 A_3; f_{12} = A_1 \vee A_3;$$

$$f_7 = A_1 A_4; f_{13} = A_1 \vee A_4;$$

$$f_8 = A_2 A_3; f_{14} = A_2 \vee A_3;$$

$$f_9 = A_2 A_4; f_{15} = A_2 \vee A_4;$$

$$f_{10} = A_3 A_4; f_{16} = A_3 \vee A_4.$$

Третьему слагаемому соответствуют формулы, зависящие от трех аргументов. Три аргумента можно выбрать  $C_4^3 = 4$  способами. Между тремя аргументами имеется два промежутка, куда можно записывать знаки дизъюнкции и конъюнкции. Следовательно, имеем 16 формул:

$$f_{17} = A_1 A_2 A_3; f_{25} = A_1 \vee A_2 A_3;$$

$$f_{18} = A_1 A_2 A_4; f_{26} = A_1 \vee A_2 A_4;$$

$$f_{19} = A_1 A_3 A_4; f_{27} = A_1 \vee A_3 A_4;$$

$$f_{20} = A_2 A_3 A_4; f_{28} = A_2 \vee A_3 A_4;$$

$$f_{21} = A_1 A_2 \vee A_3; f_{29} = A_1 \vee A_2 \vee A_3;$$

$$f_{22} = A_1 A_2 \vee A_4; f_{30} = A_1 \vee A_2 \vee A_4;$$

$$f_{23} = A_1 A_3 \vee A_4; f_{31} = A_1 \vee A_3 \vee A_4;$$

$$f_{24} = A_2 A_3 \vee A_4; f_{32} = A_2 \vee A_3 \vee A_4.$$

Четвертому слагаемому соответствуют формулы, в которые входят все четыре аргумента. Для четырех аргументов существует 8 способов соединения их знаками дизъюнкции и конъюнкции:

$$f_{33} = A_1 A_2 A_3 A_4; f_{37} = A_1 \vee A_2 A_3 A_4;$$

$$f_{34} = A_1 A_2 A_3 \vee A_4; f_{38} = A_1 \vee A_2 A_3 \vee A_4;$$

$$f_{35} = A_1 A_2 \vee A_3 A_4; f_{39} = A_1 \vee A_2 \vee A_3 A_4;$$

$$f_{36} = A_1 A_2 \vee A_3 \vee A_4; f_{40} = A_1 \vee A_2 \vee A_3 \vee A_4.$$

Других формул, зависящих от четырех аргументов, являющихся неповторными и записанных в ДНФ, с упорядоченными аргументами не существует.

Вычислим величину  $N$  и найдем число настроечных кодов и количество не равных нулю формул для значений  $n = 1, 2, 3, 4, 5, 6$  (табл. 2.1); рассмотренное кодирование булевых формул обладает значительной избыточностью.

Таблица 2.1

$n$	Число настроечных кодов $4^n$	Число тождественно не равных нулю формул $4^n - 3^n$	Число различных формул $N = \sum_{i=1}^n 2^{i-1} C_n^i$
1	4	1	1
2	16	7	4
3	64	37	13
4	256	175	40
5	1024	791	121
6	4096	3367	364

Например, при  $n = 6$  из всех возможных 4096 настроечных кодов 729 являются неиспользуемыми, поскольку они настраивают  $S$ -структуру на вычисление одной и той же формулы  $f \equiv 0$ . Все оставшиеся 3367 настроечных кодов делятся на 364 подмножества, где каждое подмножество соответствует определенной формуле, на которую может быть настроена  $S$ -структура. Все настроечные коды, принадлежащие одному и тому же множеству, настраивают  $S$ -структуру на одну и ту же формулу.

#### 2.5.2.6. Нахождение булевой формулы по настроечному коду

Если в классе ДНФ задана неповторная упорядоченная булева формула  $f$  без пропусков аргументов и к ней применена  $S$ -операция (вычеркивание аргументов на основе  $S$ -кода), в результате которой получена формула  $\varphi$ , то согласно теореме 2.2 по  $\alpha$ -коду формулы  $f$  и заданному  $S$ -коду  $S$ -структурой будет реализована формула  $z$ , не всегда совпа-

дающая с формулой  $\varphi$ . Сформулируем правило, позволяющее находить формулу  $z$ . Чтобы выяснить, какую формулу будет реализовывать  $S$ -структура, необходимо записать  $\alpha$ - и  $S$ -коды один под другим так, чтобы соответствующие разряды находились в одних и тех же колонках. После этого вычеркиваем разряды в  $\alpha$ -коде, соответствующие единицам  $S$ -кода. Оставшиеся разряды образуют новый код, который условимся называть  $\alpha_0$ -кодом; формула, записанная на основе  $\alpha_0$ -кода, и будет являться искомой формулой  $z$ .

**Пр и м е р 1.** Пусть даны  $\alpha$ - и  $S$ -коды:

$$\alpha = 01011011,$$

$$S = 01001010.$$

Удаляем в  $\alpha$ -коде разряды с номерами 2, 5, 7 (нумерация разрядов слева направо), тогда получим:

$$\alpha_0 = 00101.$$

Чтобы записать алгебраическое выражение формулы  $z$ , каждому разряду  $\alpha_0$ -кода необходимо поставить в соответствие аргументы согласно  $S$ -коду:

$$\alpha_0 = \begin{matrix} 0 & 0 & 1 & 0 & 1 \\ A_1 & A_3 & A_4 & A_6 & A_8. \end{matrix}$$

Искомая формула  $z$  имеет вид:

$$z = A_1 A_3 A_4 \vee A_6 A_8.$$

Если настроечный код задан в четверичной системе счисления, то для того чтобы восстановить по нему формулу, на которую настроится  $S$ -структура, необходимо вычеркнуть из настроечного кода цифры 1 и 3, которым соответствуют единицы в  $S$ -коде. Тогда в настроечном коде останутся цифры 0 и 2. Группа нулей с примыкающей к ним справа цифрой 2 обозначает конъюнкцию соответствующих аргументов. Если таких групп в настроечном коде несколько, то соответствующие конъюнкции соединяются знаками дизъюнкции.

**Пр и м е р 2.** Какую формулу реализует  $S$ -структура при настроечном коде 0121221010?

Запишем настроечный код и под его цифрами укажем соответствующие им аргументы:

$$\begin{array}{cccccccccc} 0 & 1 & 2 & 1 & 2 & 2 & 1 & 0 & 1 & 0 \\ A_1 & A_2 & A_3 & A_4 & A_5 & A_6 & A_7 & A_8 & A_9 & A_{10}. \end{array}$$

В коде нет троек, следовательно, вычеркиваем только единицы. Искомая формула имеет вид

$$z = A_1 A_3 \vee A_5 \vee A_6.$$

Заметим, что в найденной формуле  $z$  нет аргументов  $A_8$  и  $A_{10}$ , хотя разряды 8 и 10 в  $\alpha$ -коде не вычеркнуты. Чтобы убедиться в справедливости формулы  $z$ , достаточно отметить, что в заданном настроечном коде после шестого разряда нет цифр 2, ограничивающих длину конъюнкции в записи формулы.

**Пример 3.** Какую формулу реализует настроечный код 130013313?

Вычеркивая единицы и тройки, получим код, состоящий из двух нулей, что соответствует формуле  $z \equiv 0$ . К этому же результату придем, если будем учитывать, что согласно теореме 2.5 код, не содержащий цифры 2, настраивает  $S$ -структуру на формулу  $z \equiv 0$ .

В заключение отметим, что нахождение формулы по настроечному коду осуществляется однозначно.

### 2.5.3. Вычисление неповторных упорядоченных булевых формул выше второго порядка

Изложим кратко результаты анализа МЛМ (рис. 2.8), позволяющего строить изотропные среды для вычисления булевых формул высоких порядков.

Структура рассматриваемого МЛМ описывается следующей системой булевых формул [17, 22]:

$$\begin{cases} f_1 = y_1 \bar{z}_1 z_2 \vee (y_1 \vee z_1 \vee z_2) x \bar{z}_3, \\ f_2 = y_1 z_1 z_2 \vee (y_1 \vee \bar{z}_1 \vee z_2) y_2, \end{cases}$$

где  $z_1, z_2, z_3$  – настроечные входы МЛМ;  $x, y_1, y_2$  – информационные входы;  $f_1, f_2$  – выходы МЛМ.

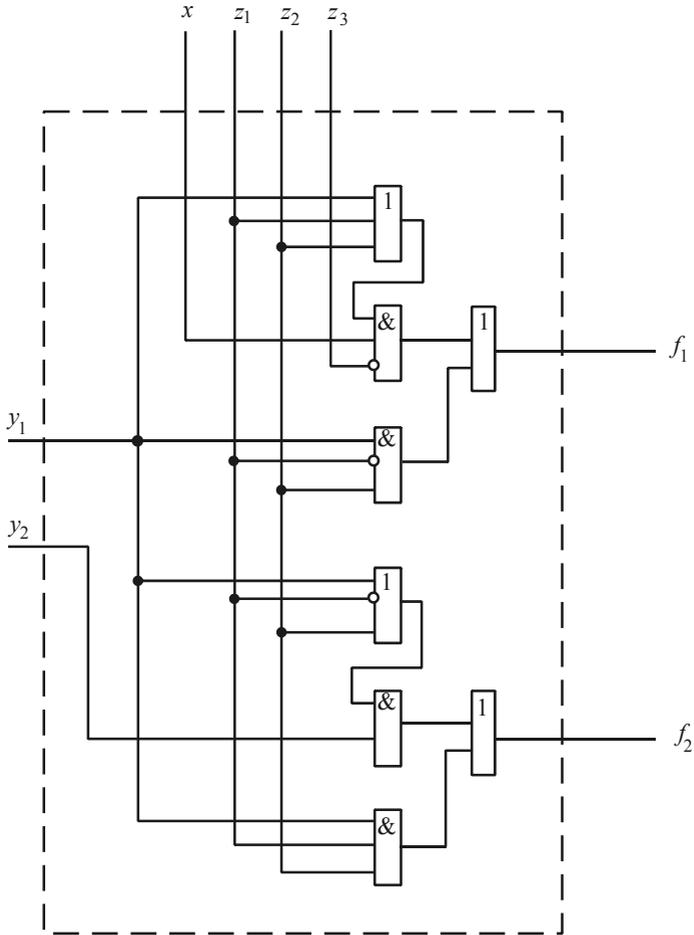


Рис. 2.8. Ячейка T-структуры

Многофункциональный логический модуль, который в дальнейшем будем называть T-ячейкой, реализует следующие системы формул:

- 1) при  $z_1 = 0, z_2 = 0, z_3 = 0$       2) при  $z_1 = 0, z_2 = 0, z_3 = 1$   
 (см. рис. 2.9, а)                              (см. рис. 2.9, б)

$$\begin{cases} f_1 = y_1x, \\ f_2 = y_2; \end{cases} \quad \begin{cases} f_1 = 0, \\ f_2 = y_2; \end{cases}$$

3) при  $z_1 = 0, z_2 = 1, z_3 = 0$   
(рис. 2.9, в)

$$\begin{cases} f_1 = y_1 + x, \\ f_2 = y_2; \end{cases}$$

4) при  $z_1 = 0, z_2 = 1, z_3 = 1$   
(рис. 2.9, г)

$$\begin{cases} f_1 = y_1, \\ f_2 = y_2; \end{cases}$$

5) при  $z_1 = 1, z_2 = 0, z_3 = 0$   
(рис. 2.9, д)

$$\begin{cases} f_1 = x, \\ f_2 = y_1 y_2; \end{cases}$$

6) при  $z_1 = 1, z_2 = 0, z_3 = 1$   
(рис. 2.9, е)

$$\begin{cases} f_1 = 0, \\ f_2 = y_1 y_2; \end{cases}$$

7) при  $z_1 = 1, z_2 = 1, z_3 = 0$   
(рис. 2.9, ж)

$$\begin{cases} f_1 = x, \\ f_2 = y_1 + y_2; \end{cases}$$

8) при  $z_1 = 1, z_2 = 1, z_3 = 1$   
(рис. 2.9, з)

$$\begin{cases} f_1 = 0, \\ f_2 = y_1 + y_2. \end{cases}$$

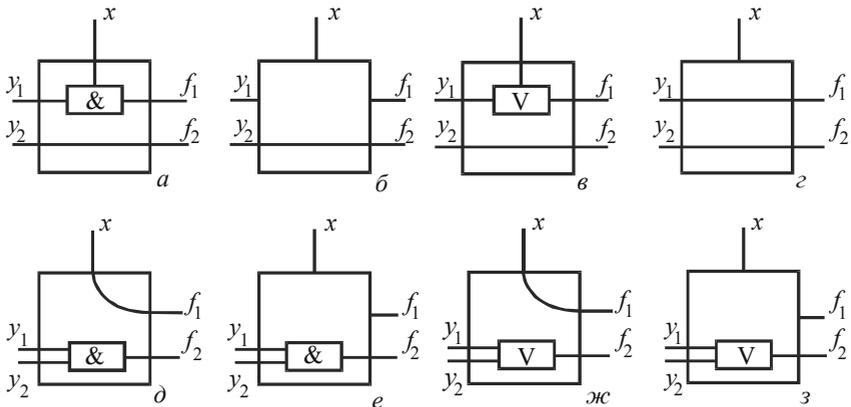


Рис. 2.9. Структурные схемы, полученные путем настройки

Таким образом, с помощью изотропной структуры ( $T$ -структуры), построенной из рассматриваемых  $T$ -ячеек, можно реализовать бесповторные упорядоченные ДНФ или КНФ, скобочные формы, а также все формулы с пропусками аргументов.

#### 2.5.4. Вычисление неупорядоченных булевых формул

Приведенные в разделах 2.5.1 – 2.5.3  $F$ -,  $S$ - и  $T$ -структуры обеспечивают вычисление только упорядоченных булевых формул. Выясним,

как можно вычислять неупорядоченные формулы, представленные в ДНФ.

Пусть задана произвольная формула  $n$  аргументов в ДНФ и пусть она имеет  $k$  конъюнкций. Очевидно, что длина конъюнкций не может превышать  $n$  букв, поскольку  $n$  – это длина минтерма (члена СДНФ) [7]. В разделе 2.5.2 было отмечено, что аргументы каждой из конъюнкций, входящих в указанную ДНФ формулы, можно упорядочить. Следовательно, заданную формулу  $n$  аргументов можно представить дизъюнкцией неповторных упорядоченных формул с пропусками аргументов:

$$f = \lambda_1 \vee \lambda_2 \vee \dots \vee \lambda_k = \bigvee_{i=1}^k \lambda_i, \quad (2.11)$$

где  $\lambda_i$  – конъюнкции заданной формулы  $f$ .

Исследуем три варианта вычисления формулы (2.11):

1) комбинационной схемой, представленной изотропной либо квазиизотропной матрицей (т.е. составленной из  $S$ - и  $T$ -ячеек);

2) многотактным автоматом путем последовательного вычисления за  $k$  тактов, когда в течение каждого такта вычисляется значение только одной конъюнкции формулы (2.11);

3) линейной изотропной структурой, построенной из МЛМ, ориентированных на вычисление как упорядоченных, так и неупорядоченных булевых формул.

**Первый вариант.** Построим изотропную матрицу на основе  $S(T)$ -структур, где символ  $S(T)$  обозначает, что структура состоит из  $S$ - либо  $T$ -ячеек (рис. 2.10).

Вместо матрицы можно рассматривать линейную  $S(T)$ -структуру, состоящую из  $n \times k$  ячеек, где входы  $A_1, A_2, \dots, A_{n+1}, A_{2n+1}, \dots, A_{(k-1)n+1}$  соединяются между собой и подключаются к устройству, моделирующему вход  $A_1$ . Входы  $A_2, A_{n+2}, A_{2n+2}, \dots, A_{(k-1)n+2}$  также соединяются между собой и объявляются входом аргумента  $A_2$  и т.д. до  $A_n$ , которому соответствует соединение  $A_n, A_{2n}, \dots, A_{kn}$ . Такую структуру целесообразно применять, если основным требованием, предъявляемым к устройству, является быстрое действие.

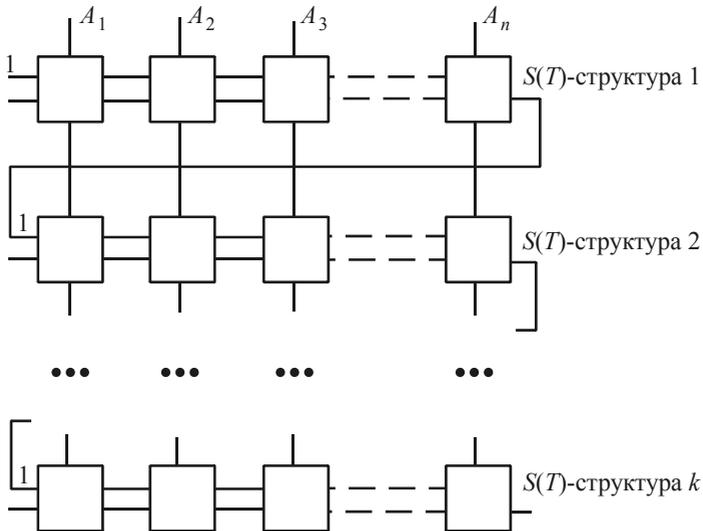


Рис. 2.10. Изотропная матрица

**Второй (многоактный) вариант.** Все настроечные коды для каждой конъюнкции  $\lambda_i$  записываются в запоминающее устройство (ЗУ), откуда последовательно поступают на настроечные входы  $S(T)$ -структуры (рис. 2.11).

С каждым тактом смены настроечных кодов на синхровход триггера  $E$  поступает импульс. Если на каком-либо настроечном коде выходной сигнал триггера  $E$  принимает единичное значение, то триггер  $E$  переходит в единичное состояние, что сигнализирует о равенстве единице формулы  $f$ . Если же после полного просмотра ЗУ триггер  $E$  остался в нулевом состоянии, то формула на данном наборе значений аргументов равна нулю. ЗУ можно организовать различными способами. Условимся считать, что каждому настроечному коду соответствует определенный адрес ЗУ. Тогда можно пронумеровать конъюнкции  $\lambda_i$  и считать, что номер этой конъюнкции совпадает с адресом, по которому в ЗУ находится ее настроечный код. Если не учитывать код  $\beta$ , которым задается распределение инверсий над аргументами, то длина выходного кода, хранящегося по заданному адресу ЗУ, равна  $2n$ . Рассмотрим пример для  $n = 7$ .

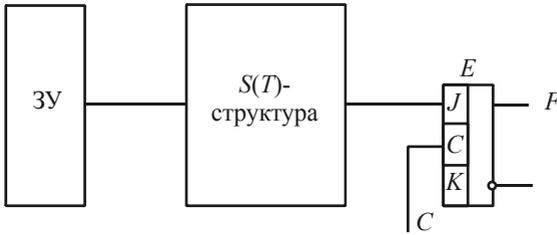


Рис. 2.11. Многотактный автомат

Пусть дана формула

$$f = A_1A_2 \vee A_1A_5A_6 \vee A_1A_3A_7 \vee A_1A_6A_7 \vee A_4A_5A_7. \quad (2.12)$$

Эта формула не является бесповторной, но каждая ее конъюнкция есть бесповторная формула с пропусками:

$$\lambda_1 = A_1A_2; \lambda_2 = A_1A_5A_6; \lambda_3 = A_1A_3A_7; \lambda_4 = A_1A_6A_7; \lambda_5 = A_4A_5A_7.$$

В ЗУ будут записаны следующие коды:

	$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	$A_6$	$A_7$	
$S =$	0	0	1	1	1	1	1,	
$\alpha =$	0	1	0	0	0	0	0	по адресу 1,
$S =$	0	1	1	1	0	0	1,	
$\alpha =$	0	0	0	0	0	1	0	по адресу 2,
$S =$	0	1	0	1	1	1	0,	
$\alpha =$	0	0	0	0	0	0	1	по адресу 3,
$S =$	0	1	1	1	1	0	0,	
$\alpha =$	0	0	0	0	0	0	1	по адресу 4,
$S =$	1	1	1	0	0	1	0,	
$\alpha =$	0	0	0	0	0	0	1	по адресу 5.

Таким образом, значение этой формулы будет определено не более чем за пять тактов. Формулу (2.12) можно вычислить и за четыре такта, если учесть, что первая и последняя конъюнкции образуют бесповторную упорядоченную формулу с пропусками аргументов  $A_3, A_6$ :

$$\lambda_1 = A_1 A_2 \vee A_4 A_5 A_7,$$

$S$ - и  $\alpha$ -коды (см. разд. 2.5.2) имеют вид

	$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	$A_6$	$A_7$
$S =$	0	0	1	1	1	1	1,
$\alpha =$	0	1	0	0	0	0	0.

Возникает вопрос о наименьшем количестве тактов, за которые можно вычислить значение формулы, ответ на который дан в следующем разделе.

### 2.5.5. Метод декомпозиции для вычисления произвольных булевых формул

Пусть задана произвольная формула  $f$ , представленная в ДНФ и зависящая от  $n$  аргументов. Подвергнем ее операции декомпозиции следующим образом:

$$f = \varphi_1 \vee \varphi_2 \vee \dots \vee \varphi_n, \quad (2.13)$$

где  $\varphi_i$  ( $i = 1, 2, \dots, n$ ) – неповторные упорядоченные формулы с пропусками или без пропусков аргументов.

Чтобы формулу  $f$  вычислить за наименьшее число тактов и с наименьшими затратами памяти, необходимо найти минимальный список формул  $\varphi_i$ , дизъюнкция которых дает формулу  $f$ . В процессе решения данной задачи разработан метод нахождения минимального списка формул  $\varphi_i$ , которые в дальнейшем будем называть  $\varphi$ -импликантами.

Суть метода состоит в следующем. Находим конъюнкцию, содержащую наименьший номер аргумента. Если их несколько, то выбираем из них первый произвольным образом. Если выбранная конъюнкция состоит более чем из одного аргумента, то она оканчивается аргументом с номером  $\delta_1$ . Затем находим конъюнкцию, содержащую первым аргумент, индекс которого является ближайшим к  $\delta_1$ . Это будет вторая конъюнкция, которая войдет в  $\varphi$ -импликанту. Вторая конъюнкция оканчивается аргументом с индексом  $\delta_2$ . Аналогично находим третью, четвертую и т.д. конъюнкции, пока не получим конъюнкцию, оканчивающуюся аргументом с номером  $\delta_i$ , таким, что других конъюнкций, первый аргумент которых превышает  $\delta_i$ , в заданной формуле  $f$  нет.

Закодируем двоичными числами все найденные  $\varphi$ -импликанты. Для этого запишем в ряд конъюнкции, образующие формулу (2.13). Каждой из них поставим в соответствие двоичный разряд и единицами будем отмечать конъюнкции, входящие в ту или иную  $\varphi$ -импликанту. Перечислив все  $\varphi$ -импликанты, получим таблицу двоичных чисел, где каждому числу будет соответствовать определенная  $\varphi$ -импликанта. Обозначим  $\varphi$ -импликанты символами  $\varphi_1, \varphi_2, \dots, \varphi_n$ . Затем составим булево уравнение, представляющее собой конъюнкцию скобочных выражений, где в каждой скобке записана дизъюнкция  $\varphi$ -импликант. Число скобочных выражений в уравнении равно числу конъюнкций заданной формулы, т.е. каждой конъюнкции заданной формулы соответствует скобочное выражение в уравнении. Чтобы найти  $\varphi_1, \varphi_2, \dots, \varphi_n$ , входящие в скобочные выражения, достаточно обратить внимание на единицы в соответствующей колонке полученной таблицы.

Представим булево уравнение в ДНФ. После минимизации получим дизъюнкцию конъюнкции символов  $\varphi_1, \varphi_2, \dots, \varphi_n$ . Каждая конъюнкция дает одно решение задачи декомпозиции.

Для иллюстрации метода рассмотрим пример. Дана формула

$$f = A_1A_2A_3 \vee A_1A_4A_6 \vee A_4A_7A_8A_9 \vee A_5A_7A_8 \vee \\ \vee A_9A_{10}A_{12}A_{14} \vee A_9A_{13} \vee A_{10}A_{11}A_{12} \vee A_{13}A_{14},$$

которая не является неповторной, следовательно, ее необходимо подвергнуть операции декомпозиции. Наиболее простой вариант декомпозиции состоит в том, что вычисление ведется отдельно по каждой конъюнкции. При этом потребуется 8 тактов. Выясним, сколько тактов потребуется при минимальном варианте. Составляем таблицу (прил. 1, табл. П.1.3), где полный список  $\varphi$ -импликант, выраженных через исходные аргументы заданной формулы, имеет вид

$$\varphi_1 = A_1A_2A_3 \vee A_4A_7A_8A_9 \vee A_{10}A_{11}A_{12} \vee A_{13}A_{14};$$

$$\varphi_2 = A_1A_2A_3 \vee A_5A_7A_8 \vee A_9A_{10}A_{12}A_{14};$$

$$\varphi_3 = A_1A_2A_3 \vee A_5A_7A_8 \vee A_9A_{13};$$

$$\varphi_4 = A_1A_2A_3 \vee A_5A_7A_8 \vee A_{10}A_{11}A_{12} \vee A_{13}A_{14};$$

$$\varphi_5 = A_1A_4A_6 \vee A_9A_{10}A_{12}A_{14};$$

$$\varphi_6 = A_1 A_4 A_6 \vee A_9 A_{13};$$

$$\varphi_7 = A_1 A_4 A_6 \vee A_{10} A_{11} A_{12} \vee A_{13} A_{14}.$$

Составляем булево уравнение. Прежде всего рассмотрим колонку таблицы, где записана конъюнкция  $A_4 A_7 A_8 A_9$ , так как в ней только одна единица. Это значит, что конъюнкция  $A_4 A_7 A_8 A_9$  войдет в какую-либо формулу искомого списка только в единственном случае: когда в список будет включена  $\varphi$ -импликанта  $\varphi_1$ . Очевидно, что эта  $\varphi$ -импликанта войдет во все варианты декомпозиции. Далее, поскольку импликанта  $\varphi_1$  введена в список, в него будут введены и конъюнкции  $A_1 A_2 A_3$ ,  $A_{10} A_{11} A_{12}$ ,  $A_{13} A_{14}$ . Вычеркиваем из таблицы верхнюю строку и колонки, обозначенные единицами в этой строке. Построим новую таблицу (см. прил. 1, табл. П.1.4), а по ней – булево уравнение в виде конъюнкции дизъюнкций:

$$(\varphi_5 \vee \varphi_6 \vee \varphi_7) (\varphi_2 \vee \varphi_3 \vee \varphi_4) (\varphi_2 \vee \varphi_5) (\varphi_3 \vee \varphi_6) = 1.$$

Раскроем скобки:

$$\begin{aligned} (\varphi_5 \vee \varphi_2 \varphi_6 \vee \varphi_2 \varphi_7) (\varphi_3 \vee \varphi_2 \varphi_6 \vee \varphi_4 \varphi_6) = & \varphi_3 \varphi_5 \vee \varphi_2 \varphi_3 \varphi_6 \vee \varphi_2 \varphi_3 \varphi_7 \vee \\ & \vee \varphi_2 \varphi_5 \varphi_6 \vee \varphi_2 \varphi_6 \vee \varphi_2 \varphi_6 \varphi_7 \vee \varphi_4 \varphi_5 \varphi_6 \vee \varphi_2 \varphi_4 \varphi_6 \vee \varphi_2 \varphi_4 \varphi_7. \end{aligned}$$

Выполним все операции поглощения, тогда получим минимальную ДНФ:

$$\varphi_3 \varphi_5 \vee \varphi_2 \varphi_6 \vee \varphi_2 \varphi_3 \varphi_7 \vee \varphi_4 \varphi_5 \varphi_6 = 1.$$

Отсюда следует, что существует четыре варианта искомого списка формул, из которых два варианта соответствуют случаю, когда список содержит три формулы. Эти два минимальных варианта имеют вид

$$\begin{cases} \varphi_1 = A_1 A_2 A_3 \vee A_4 A_7 A_8 A_9 \vee A_{10} A_{11} A_{12} \vee A_{13} A_{14}, \\ \varphi_3 = A_1 A_2 A_3 \vee A_5 A_7 A_8 \vee A_9 A_{13}, \\ \varphi_5 = A_1 A_4 A_6 \vee A_9 A_{10} A_{12} A_{14}; \end{cases} \quad (2.14)$$

$$\begin{cases} \varphi_1 = A_1 A_2 A_3 \vee A_4 A_7 A_8 A_9 \vee A_{10} A_{11} A_{12} \vee A_{13} A_{14}, \\ \varphi_2 = A_1 A_2 A_3 \vee A_5 A_7 A_8 \vee A_9 A_{10} A_{12} A_{14}, \\ \varphi_6 = A_1 A_4 A_6 \vee A_9 A_{13}. \end{cases} \quad (2.15)$$

В списке (2.14) конъюнкция  $A_1A_2A_3$  встречается дважды. Одну из них можно удалить, например, из формулы  $\varphi_3$ . Аналогично удаляем эту конъюнкцию и из формулы  $\varphi_2$  списка (2.15).

В связи с тем, что в формуле  $\varphi_3$  системы (2.14) конъюнкцию  $A_1A_2A_3$  можно было и не вычеркивать, делаем вывод, что в общем случае представление минимального списка  $\varphi$ -импликант  $\alpha$ - и  $S$ -кодами неоднозначно.

### 2.5.6. Однотактное вычисление неупорядоченных булевых формул

Как отмечалось в разделе 2.5.4, существует третий путь вычисления неупорядоченных булевых формул с помощью линейных изотропных сред на многофункциональных логических модулях, обеспечивающих вычисление как упорядоченных, так и неупорядоченных булевых формул. Применение МЛМ подобного типа сопряжено с трудностями их разработки. Проиллюстрируем МЛМ [8], ориентированный на вычисление вышеназванных булевых формул, представленных классами 1–6 и подклассом  $J$ , определяемым следующим образом.

Пусть дана формула, явно или неявно зависящая от аргументов  $x_1, x_2, \dots, x_n$ . Запишем эти аргументы в порядке возрастания их индексов слева направо. Аргумент, имеющий наименьший индекс, будем называть минимальным, а наибольший – максимальным. Диапазоном формулы назовем замкнутый интервал, границы которого образуют индексы минимального и максимального аргументов. Условимся, что интервалы двух различных формул пересекаются, если минимальный аргумент одной из формул входит в интервал другой.

Если формула  $f$  представлена в виде [23]:

$$f = l * \omega * q,$$

то она входит в подкласс  $J$ , где  $l$  и  $\omega$  – упорядоченные формулы с пересекающимися диапазонами;  $q$  – упорядоченная формула, минимальный аргумент которой не входит в диапазон функций  $l$  и  $\omega$ .

В общем случае формула  $q$  может быть тождественно равна нулю; \* – знак конъюнкции или дизъюнкции. Формулы  $l$  и  $\omega$  могут быть любого порядка.

Логическая схема МЛМ (рис. 2.12) описывается следующей системой булевых формул [21]:

$$\begin{cases} f_1 = y_1(\bar{z}_1z_2 \vee \bar{z}_1z_3\bar{z}_4 \vee x\bar{z}_3), \\ f_2 = (\bar{z}_1x \vee z_3x \vee z_2 \vee \bar{z}_2y_1 \vee \bar{z}_1\bar{z}_3 \vee \bar{z}_1z_4)y_2 \vee \\ \vee xz_4 \vee y_1z_1z_2, \end{cases} \quad (2.16)$$

где  $x, y_1, y_2$  – информационные входы;  $z_1, z_2, z_3, z_4$  – настроечные входы;  $f_1, f_2$  – выходы МЛМ.

Многофункциональный логический модуль, который в дальнейшем будем называть *H*-ячейкой, реализует следующие системы булевых формул:

1) при  $z_4 = 0, z_3 = 0, z_2 = 0, z_1 = 0$   
(рис. 2.13, а)

$$\begin{cases} f_1 = y_1x, \\ f_2 = y_2; \end{cases}$$

2) при  $z_4 = 0, z_3 = 0, z_2 = 0, z_1 = 1$   
(рис. 2.13, б)

$$\begin{cases} f_1 = x, \\ f_2 = y_1y_2; \end{cases}$$

3) при  $z_4 = 0, z_3 = 0, z_2 = 1, z_1 = 0$   
(рис. 2.13, в)

$$\begin{cases} f_1 = y_1 \vee x, \\ f_2 = y_2; \end{cases}$$

4) при  $z_4 = 0, z_3 = 0, z_2 = 1, z_1 = 1$   
(рис. 2.13, г)

$$\begin{cases} f_1 = x, \\ f_2 = y_1 \vee y_2; \end{cases}$$

5) при  $z_4 = 0, z_3 = 1, z_2 = 0, z_1 = 0$   
(рис. 2.13, д)

$$\begin{cases} f_1 = y_1, \\ f_2 = y_2x; \end{cases}$$

6) при  $z_4 = 0, z_3 = 1, z_2 = 0, z_1 = 1$   
(рис. 2.13, е)

$$\begin{cases} f_1 = 0, \\ f_2 = xy_2; \end{cases}$$

7) при  $z_4 = 0, z_3 = 1, z_2 = 1, z_1 = 0$   
(рис. 2.13, ж)

$$\begin{cases} f_1 = y_1, \\ f_2 = y_2; \end{cases}$$

8) при  $z_4 = 0, z_3 = 1, z_2 = 1, z_1 = 1$   
(рис. 2.13, з)

$$\begin{cases} f_1 = 0, \\ f_2 = y_1 \vee y_2; \end{cases}$$

9) при  $z_4 = 1, z_3 = 0, z_2 = 0, z_1 = 0$   
(рис. 2.13, и)

$$\begin{cases} f_1 = y_1x, \\ f_2 = y_2 \vee x; \end{cases}$$

10) при  $z_4 = 1, z_3 = 0, z_2 = 0, z_1 = 1$  (рис. 2.13, к)

$$\begin{cases} f_1 = x, \\ f_2 = y_1y_2 \vee x; \end{cases}$$

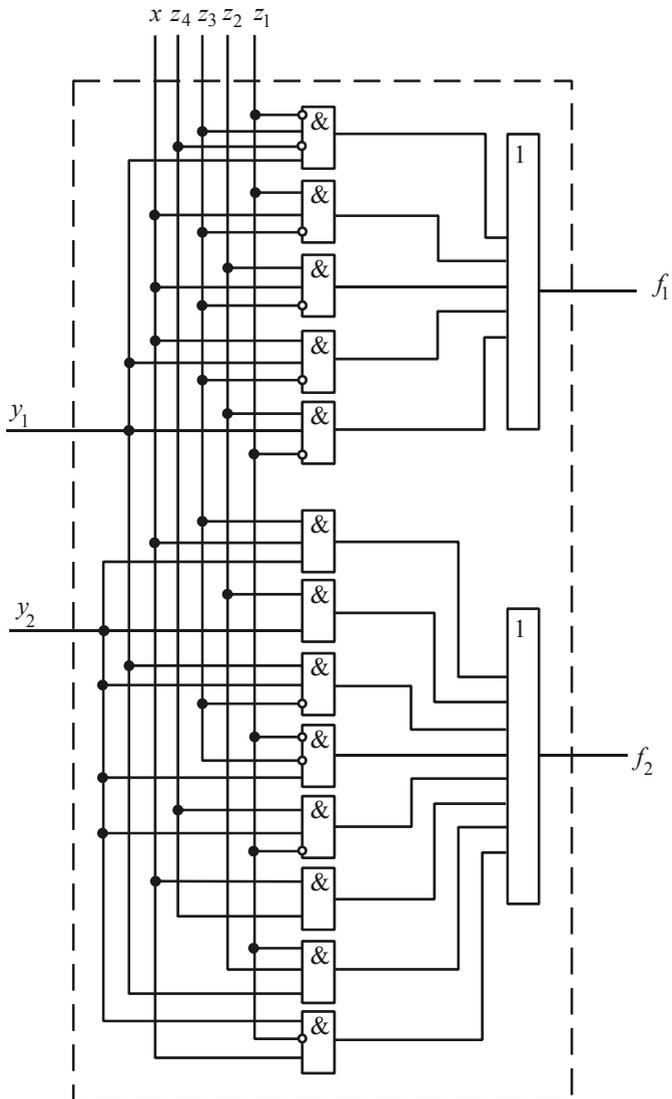


Рис. 2.12. Ячейка H-структуры

11) при  $z_4 = 1, z_3 = 0, z_2 = 1, z_1 = 0$  (рис. 2.13, л)

$$\begin{cases} f_1 = y_1 \vee x, \\ f_2 = y_2 \vee x; \end{cases}$$

12) при  $z_4 = 1, z_3 = 0, z_2 = 1, z_1 = 1$  (рис. 2.13, м)

$$\begin{cases} f_1 = x, \\ f_2 = y_1 \vee y_2 \vee x; \end{cases}$$

13) при  $z_4 = 1, z_3 = 1, z_2 = 0, z_1 = 0$  (рис. 2.13, н)

$$\begin{cases} f_1 = 0, \\ f_2 = y_2 \vee x; \end{cases}$$

14) при  $z_4 = 1, z_3 = 1, z_2 = 0, z_1 = 1$  (рис. 2.13, о)

$$\begin{cases} f_1 = 0, \\ f_2 = x; \end{cases}$$

15) при  $z_4 = 1, z_3 = 1, z_2 = 1, z_1 = 0$  (рис. 2.13, п)

$$\begin{cases} f_1 = y_1, \\ f_2 = y_2 \vee x; \end{cases}$$

16) при  $z_4 = 1, z_3 = 1, z_2 = 1, z_1 = 1$  (рис. 2.13, р)

$$\begin{cases} f_1 = 0, \\ f_2 = y_1 \vee y_2 \vee x. \end{cases}$$

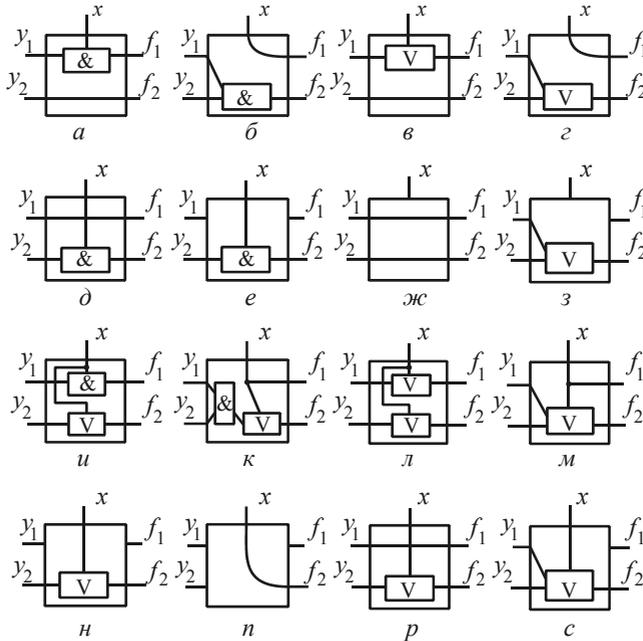


Рис. 2.13. Структурные схемы, полученные путем настройки

Таким образом, изотропные среды, построенные на  $H$ -ячейках, обеспечивают реализацию как класса неповторных упорядоченных произвольных нормальных формул из  $h$  букв (в том числе любых скобочных), так и формул из класса неупорядоченных и повторных формул, определяемых подклассом  $J$ .

### 2.5.7. Вычисление систем булевых формул из классов неповторных упорядоченных и неупорядоченных формул

В разделе 2.5.5 показано, что существующие МЛМ охватывают классы 1–6 и подкласс  $J$  классификации (см. рис. 2.2). В данном и следующем разделах покажем реализацию разработанных МЛМ, ориентированных на вычисления систем булевых формул из классов неповторных упорядоченных и неупорядоченных формул, а также МЛМ, реализующих класс повторных упорядоченных произвольных нормальных булевых формул из  $h$  букв и систем булевых формул как с пропусками аргументов, так и без них.

Ячейка (рис. 2.14) реализует системы булевых формул из классов неповторных упорядоченных и неупорядоченных формул.

Напомним, что формулу будем считать неповторной, если каждый аргумент входит в нее не более одного раза. Неповторной будем считать формулу и в том случае, если существуют тождественные преобразования, в результате которых формула, содержащая повторные аргументы, приводится к виду, не содержащему повторных аргументов. Во всех остальных случаях формула является повторной.

Структура предлагаемой ячейки описывается следующей системой формул [12]:

$$\begin{cases} f_1 = y_1(\bar{Z}_1 Z_2 \vee Z_3) \vee (y_1 \vee Z_1 \vee Z_2) x \bar{Z}_3, \\ f_2 = y_1 Z_1 Z_2 \bar{Z}_3 \vee y_2 Z_1 \bar{Z}_2 Z_3 \vee (y_1 \vee \bar{Z}_1 \vee Z_2) y_2 \bar{Z}_3 \vee \\ \quad \vee (y_2 \vee Z_1 \vee Z_2) x Z_3, \\ f_3 = y_3 \bar{Z}_3 \vee (y_2 \vee \bar{Z}_1 \vee \bar{Z}_2) y_3 Z_3 \vee y_2 \bar{Z}_1 Z_2 Z_3. \end{cases} \quad (2.17)$$

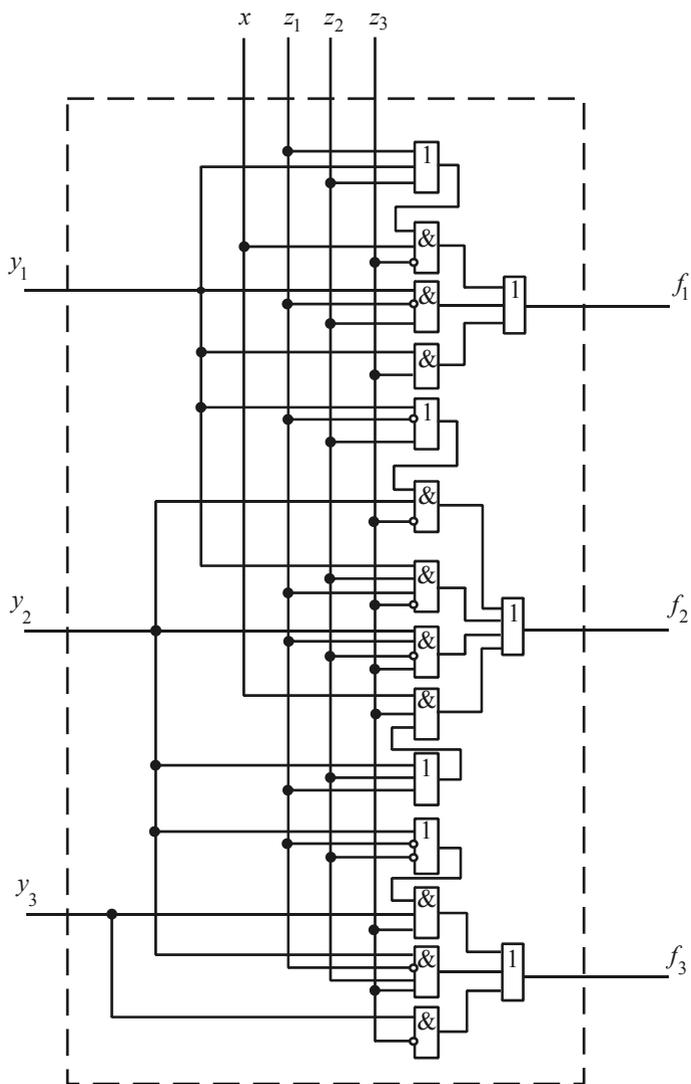


Рис. 2.14. Ячейка L-структуры

Ячейка путем настройки реализует следующие системы формул:

1) при  $Z_1 = 0, Z_2 = 0, Z_3 = 0$

(рис. 2.15, а)

$$\begin{cases} f_1 = xy_1, \\ f_2 = y_2, \\ f_3 = y_3; \end{cases}$$

5) при  $Z_1 = 1, Z_2 = 0, Z_3 = 0$

(рис. 2.15, б)

$$\begin{cases} f_1 = x, \\ f_2 = y_1y_2, \\ f_3 = y_3; \end{cases}$$

2) при  $Z_1 = 0, Z_2 = 0, Z_3 = 1$

(рис. 2.15, в)

$$\begin{cases} f_1 = y_1, \\ f_2 = xy_2, \\ f_3 = y_3; \end{cases}$$

6) при  $Z_1 = 1, Z_2 = 0, Z_3 = 1$

(рис. 2.15, г)

$$\begin{cases} f_1 = y_1, \\ f_2 = x \vee y_2, \\ f_3 = y_3; \end{cases}$$

3) при  $Z_1 = 0, Z_2 = 1, Z_3 = 0$

(рис. 2.15, д)

$$\begin{cases} f_1 = x \vee y_1, \\ f_2 = y_2, \\ f_3 = y_3; \end{cases}$$

7) при  $Z_1 = 1, Z_2 = 1, Z_3 = 0$

(рис. 2.15, е)

$$\begin{cases} f_1 = x, \\ f_2 = y_1 \vee y_2, \\ f_3 = y_3; \end{cases}$$

4) при  $Z_1 = 0, Z_2 = 1, Z_3 = 1$

(рис. 2.15, з)

$$\begin{cases} f_1 = y_1, \\ f_2 = x, \\ f_3 = y_2 \vee y_3; \end{cases}$$

8) при  $Z_1 = 1, Z_2 = 1, Z_3 = 1$

(рис. 2.15, ж)

$$\begin{cases} f_1 = y_1, \\ f_2 = x, \\ f_3 = y_2y_3. \end{cases}$$

Проиллюстрируем работу однородных сред, построенных из предлагаемых ячеек на следующих примерах.

**Пример 1.** Для реализации системы булевых формул из классов бесповторных упорядоченных и неупорядоченных формул вида

$$\begin{cases} f_1 = x_1x_2x_3 \vee x_4x_5 \vee x_6, \\ f_2 = (x_7 \vee x_9)(x_8 \vee x_{10}) \end{cases}$$

строится однородная среда с настроечными кодами (рис. 2.16).

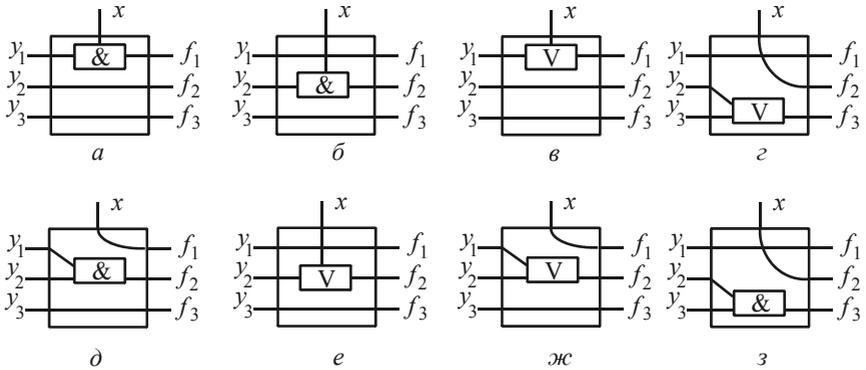


Рис. 2.15. Структурные схемы, полученные путем настройки

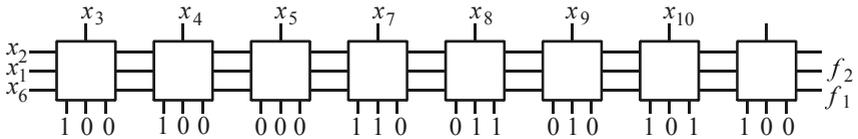


Рис. 2.16. Иллюстрация к примеру 1

Пример 2. Аналогично можно найти настроечные коды для системы формул вида

$$\begin{cases} f_1 = (x_1 x_3 \vee x_4) x_2, \\ f_2 = x_5 \vee x_6 x_7. \end{cases}$$

Ее однородная среда и настроечные коды представлены на рис. 2.17.

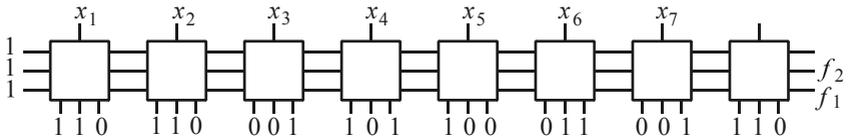


Рис. 2.17. Иллюстрация к примеру 2

Пример 3. На рис. 2.18 показаны настроечные коды каждой ячейки однородной среды, реализующей систему формул вида

$$\begin{cases} f_1 = (x_1 x_2 \vee x_3 x_4) x_5, \\ f_2 = (x_6 \vee x_7)(x_8 \vee x_9). \end{cases}$$

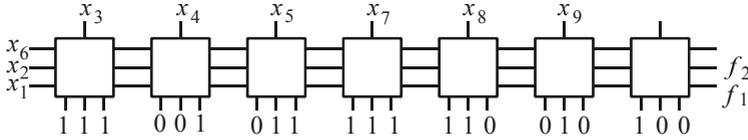


Рис. 2.18. Иллюстрация к примеру 3

Таким образом, изотропная среда, построенная на  $L$ -ячейках, обеспечивает реализацию систем булевых формул из классов неповторных упорядоченных и неупорядоченных булевых формул.

**2.5.8. Вычисление повторных упорядоченных произвольных нормальных булевых формул из  $h$  букв и систем булевых формул как с пропусками аргументов, так и без них**

Ячейка (рис. 2.19) реализует класс повторных упорядоченных произвольных нормальных булевых формул из  $h$  букв, а также системы булевых формул как с пропусками аргументов, так и без них.

Структура предлагаемой ячейки описывается следующей системой формул [13]:

$$\begin{cases} f_1 = \bar{Z}_3\bar{Z}_4[x(Z_2 \vee y_1 \vee Z_1) \vee \bar{Z}_1Z_2y_1] \vee Z_3\bar{Z}_4y_1 \vee \bar{Z}_3Z_4x \vee \\ \vee Z_3Z_4[x(Z_1 \vee \bar{Z}_2 \vee y_1) \vee \bar{Z}_1Z_2y_1], \\ f_2 = \bar{Z}_3\bar{Z}_4[y_2(Z_2 \vee \bar{Z}_1 \vee y_1) \vee Z_1Z_2y_1] \vee Z_3\bar{Z}_4y_2(x \vee Z_2) \vee \bar{Z}_3Z_4[y_2(Z_1 \vee \\ \vee \bar{Z}_2y_1 \vee y_1x) \vee Z_1(y_1 \vee Z_2x)] \vee Z_3Z_4[\bar{Z}_1Z_2(y_2 \vee x) \vee y_1(\bar{Z}_2 \vee x)], \\ f_3 = \bar{Z}_3\bar{Z}_4[y_3(Z_1 \vee \bar{Z}_2 \vee x) \vee \bar{Z}_1\bar{Z}_2x] \vee Z_3\bar{Z}_4[y_3(Z_2 \vee \bar{Z}_1 \vee x) \vee Z_1Z_2x] \vee \\ \vee \bar{Z}_3Z_4[y_3(\bar{Z}_1 \vee Z_2 \vee x) \vee \bar{Z}_1\bar{Z}_2x] \vee Z_3Z_4[y_3(Z_1 \vee Z_2 \vee y_2) \vee Z_1y_2]. \end{cases}$$

Ячейка путем настройки реализует следующие системы формул:

1) при  $Z_4 = 0, Z_3 = 0, Z_2 = 0,$   
 $Z_1 = 1$  (рис. 2.20, а)

$$\begin{cases} f_1 = x, \\ f_2 = y_1y_2, \\ f_3 = y_3; \end{cases}$$

2) при  $Z_4 = 0, Z_3 = 0, Z_2 = 1,$   
 $Z_1 = 0$  (рис. 2.20, б)

$$\begin{cases} f_1 = x \vee y_1, \\ f_2 = y_2, \\ f_3 = xy_3; \end{cases}$$

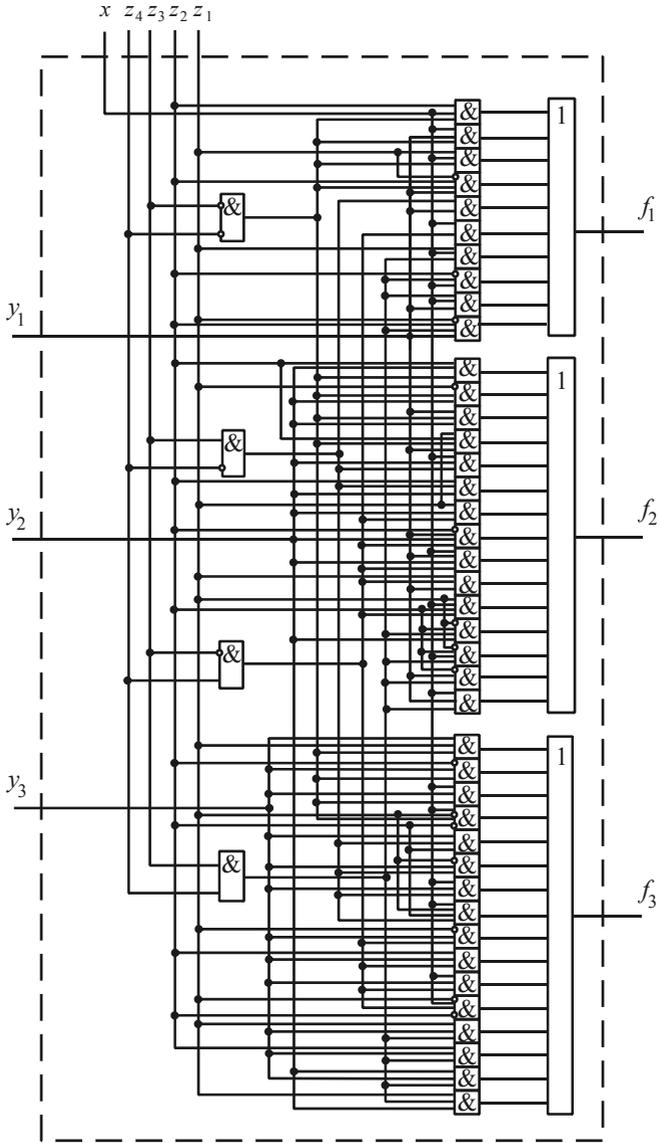


Рис. 2.19. Ячейка V-структуры

3) при  $Z_4 = 0, Z_3 = 0, Z_2 = 1, Z_1 = 1$  (рис. 2.20, в)

$$\begin{cases} f_1 = x, \\ f_2 = y_1 \vee y_2, \\ f_3 = y_3; \end{cases}$$

4) при  $Z_4 = 0, Z_3 = 1, Z_2 = 0, Z_1 = 0$  (рис. 2.20, г)

$$\begin{cases} f_1 = y_1, \\ f_2 = xy_2, \\ f_3 = y_3; \end{cases}$$

5) при  $Z_4 = 0, Z_3 = 1, Z_2 = 0, Z_1 = 1$  (рис. 2.20, д)

$$\begin{cases} f_1 = y_1, \\ f_2 = xy_2, \\ f_3 = xy_3; \end{cases}$$

6) при  $Z_4 = 0, Z_3 = 1, Z_2 = 1, Z_1 = 0$  (рис. 2.20, е)

$$\begin{cases} f_1 = y_1, \\ f_2 = y_2, \\ f_3 = y_3; \end{cases}$$

7) при  $Z_4 = 0, Z_3 = 1, Z_2 = 1, Z_1 = 1$  (рис. 2.20, ж)

$$\begin{cases} f_1 = y_1, \\ f_2 = y_2, \\ f_3 = x \vee y_3; \end{cases}$$

8) при  $Z_4 = 1, Z_3 = 0, Z_2 = 0, Z_1 = 0$  (рис. 2.20, з)

$$\begin{cases} f_1 = x, \\ f_2 = y_1 y_2, \\ f_3 = x \vee y_3; \end{cases}$$

9) при  $Z_4 = 1, Z_3 = 0, Z_2 = 0, Z_1 = 1$  (рис. 2.20, и)

$$\begin{cases} f_1 = x, \\ f_2 = y_1 \vee y_2, \\ f_3 = xy_3; \end{cases}$$

10) при  $Z_4 = 1, Z_3 = 0, Z_2 = 1, Z_1 = 0$  (рис. 2.20, к)

$$\begin{cases} f_1 = x, \\ f_2 = xy_1 y_2, \\ f_3 = y_3; \end{cases}$$

11) при  $Z_4 = 1, Z_3 = 0, Z_2 = 1, Z_1 = 1$  (рис. 2.20, л)

$$\begin{cases} f_1 = x, \\ f_2 = x \vee y_1 \vee y_2, \\ f_3 = y_3; \end{cases}$$

12) при  $Z_4 = 1, Z_3 = 1, Z_2 = 0, Z_1 = 0$  (рис. 2.20, м)

$$\begin{cases} f_1 = x, \\ f_2 = y_1, \\ f_3 = y_2 y_3; \end{cases}$$

13) при  $Z_4 = 1, Z_3 = 1, Z_2 = 0, Z_1 = 1$  (рис. 2.20, н)

$$\begin{cases} f_1 = x, \\ f_2 = y_1, \\ f_3 = y_2 \vee y_3; \end{cases}$$

14) при  $Z_4 = 1, Z_3 = 1, Z_2 = 1, Z_1 = 0$  (рис. 2.20, о)

$$\begin{cases} f_1 = y_1, \\ f_2 = x \vee y_2, \\ f_3 = y_3; \end{cases}$$

15) при  $Z_4 = 1, Z_3 = 1, Z_2 = 1,$   
 $Z_1 = 1$  (рис. 2.20, *p*)

$$\begin{cases} f_1 = x, \\ f_2 = xy_1, \\ f_3 = y_2 \vee y_3; \end{cases}$$

16) при  $Z_4 = 0, Z_3 = 0, Z_2 = 0,$   
 $Z_1 = 0$  (рис. 2.20, *c*)

$$\begin{cases} f_1 = xy_1, \\ f_2 = y_2, \\ f_3 = x \vee y_3. \end{cases}$$

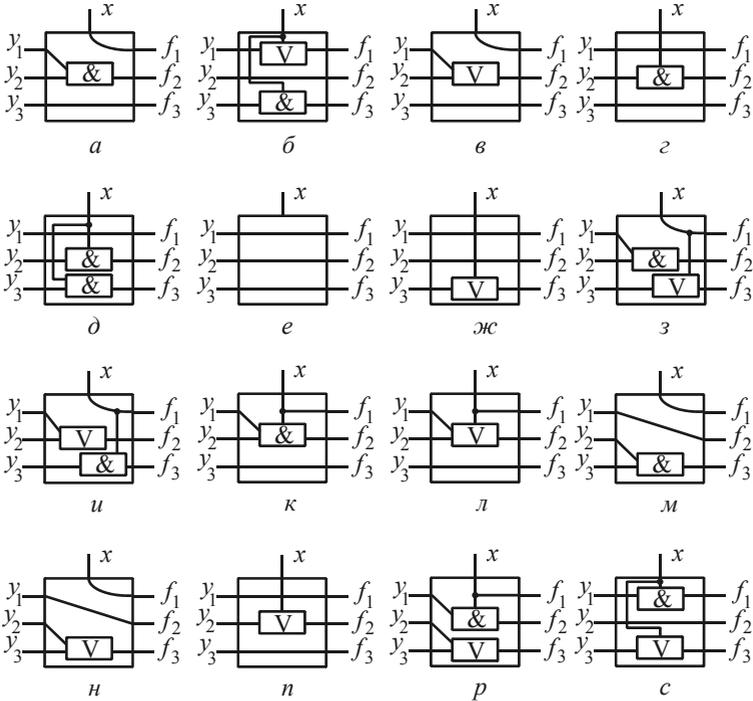


Рис. 2.20. Структурные схемы, полученные путем настройки

Работа однородных сред, построенных из предлагаемых ячеек, показана на следующих примерах.

Пример 1. Реализация повторной упорядоченной формулы без пропусков аргументов, представленной в КНФ,

$$f_1 = (x_1 \vee x_2)(x_2 \vee x_3)(x_3 \vee x_4)$$

приведена на рис. 2.21.

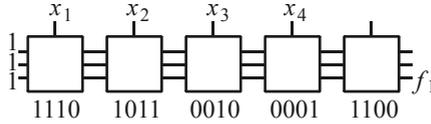


Рис. 2.21. Иллюстрация к примеру 1

Пример 2. Повторная упорядоченная формула без пропусков аргументов, представленная в ДНФ,

$$f_2 = x_1x_2 \vee x_2x_3 \vee x_3x_4$$

реализуется линейной однородной структурой (рис. 2.22).

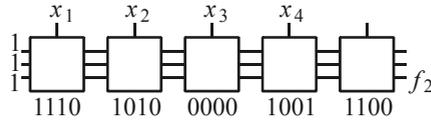


Рис. 2.22. Иллюстрация к примеру 2

Пример 3. На рис. 2.23 показаны настроечные коды каждой ячейки однородной структуры, реализующей повторную упорядоченную формулу без пропусков аргументов, представленную в форме высших (4-го) порядков:

$$f_3 = (x_1 \vee x_2)x_3 \vee x_3(x_4 \vee x_5 \vee x_6) .$$

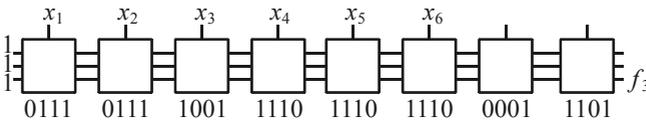


Рис. 2.23. Иллюстрация к примеру 3

Пример 4. Реализация повторной упорядоченной формулы с пропусками аргументов  $x_2$  и  $x_6$ , представленной в ДНФ,

$$f_4 = x_1x_3 \vee x_1x_4 \vee x_3x_4 \vee x_5 \vee x_7$$

показана на рис. 2.24.

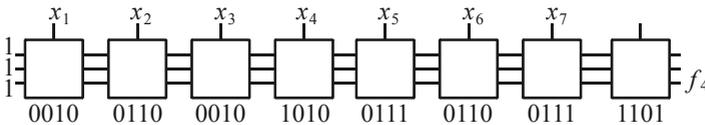


Рис. 2.24. Иллюстрация к примеру 4

Пример 5. Неупорядоченная повторная формула, представленная в ДНФ,

$$f_5 = x_1x_3x_6 \vee x_2x_4x_6 \vee x_5x_7$$

реализуется структурой, приведенной на рис. 2.25.

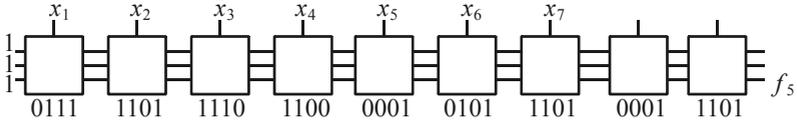


Рис. 2.25. Иллюстрация к примеру 5

Пример 6. На рис. 2.26 представлена реализация системы формул высокого порядка с пропусками аргументов вида

$$\begin{cases} f_6 = \{[(x_1x_6 \vee x_8)x_{10} \vee x_{11}]x_{12} \vee x_{13}\} \{[(x_2x_3x_4 \vee x_5)x_7x_9 \vee x_{14}]\} \vee \\ \vee [(x_{15}x_{16} \vee x_{17})x_{18} \vee x_{20}]x_{21}, \\ f_7 = \{[(x_1x_2 \vee x_3 \vee x_4)x_5x_6 \vee x_7 \vee x_9]x_{14} \vee x_{16}\} x_{17} \vee x_{18} \} x_{20} \vee x_{21}. \end{cases}$$

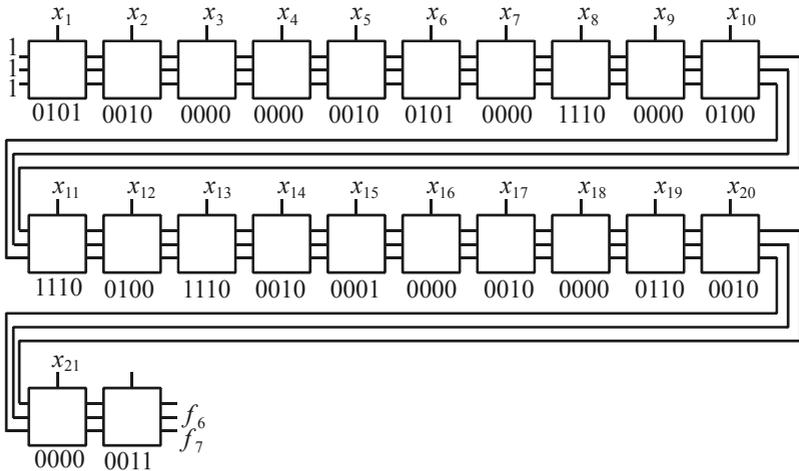


Рис. 2.26. Иллюстрация к примеру 6

Пример 7. Для реализации неповторной упорядоченной формулы вида

$$f_8 = [(x_1 \vee x_2)(x_3 \vee x_4) \vee (x_5 \vee x_6)(x_7 \vee x_8)][(x_9 \vee x_{10})(x_{11} \vee x_{12}) \vee (x_{13} \vee x_{14})(x_{15} \vee x_{16})]$$

можно построить древовидную структуру с выделенными каскадами (рис. 2.27). Схема каскада и настроечные коды для его ячеек показаны на рис. 2.28.

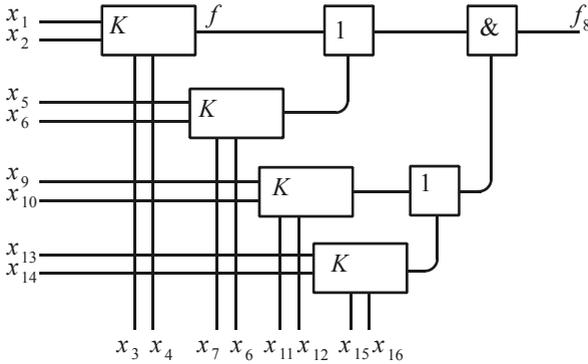


Рис. 2.27. Иллюстрация к примеру 7

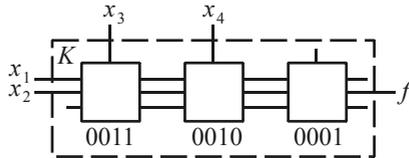


Рис. 2.28. Схема каскада из примера 7

Пример 8. Повторная неупорядоченная формула без пропусков аргументов, представленная в ДНФ,

$$f_9 = x_1x_3 \vee x_2x_3 \vee x_2x_4$$

реализуется однородной структурой (рис. 2.29).

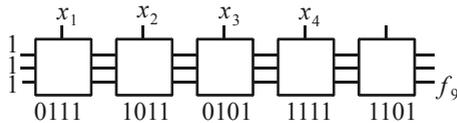


Рис. 2.29. Иллюстрация к примеру 8

Таким образом, изотропная среда, построенная на  $V$ -ячейках, реализует класс повторных упорядоченных произвольных нормальных булевых формул из  $h$  букв, а также системы булевых формул как с пропусками аргументов, так и без них.

## 2.6. ВЫВОДЫ

Основной результат исследований, изложенный в данной главе, заключается в решении проблемы вычисления систем булевых формул из классов неповторных упорядоченных и неупорядоченных, а также класса повторных упорядоченных произвольных нормальных булевых формул из  $h$  букв и систем булевых формул как с пропусками аргументов, так и без них.

1. Предложена классификация булевых формул. Установлено, что для каждого класса достаточно реализовать лишь одну схему, структура которой описывается формулой представителя класса.

2. Предложена булева модель логики перестраиваемых структур, используя которую можно создавать практические логические управляющие устройства с перестраиваемой структурой, реализующие алгоритм управления из заранее заданного класса.

3. Проведен анализ существующих многофункциональных логических модулей и на их основе синтезированы новые структуры.

3. Предложены многоканальные линейные однородные структуры из  $h-1$  ячеек для реализации булевых формул в рассматриваемом базисе из  $h$  букв.

4. Предложены плоскостные однородные структуры для реализации указанных классов булевых формул.

5. Установлено, что для настройки структуры логического управляющего устройства на конкретную формулу достаточно знать только код этой формулы.

## ЛИТЕРАТУРА

1. *Артюхов В.Л., Копейкин Г.А., Шальто А.А.* Настраиваемые модули для управляющих логических устройств. – Л.: Энергоатомиздат, 1981. – 168 с.
2. *Гиндикин С.А.* Алгебра логики в задачах. – М.: Наука, 1972. – 288 с.
3. *Колдуэлл С.* Логический синтез релейных устройств / Пер. с англ. Г.К. Москатова, А.Д. Таланцева. – М.: ИЛ, 1962. – 740 с.

4. Лазарев В.Г., Пийль Е.И., Турута Е.Н. Построение программируемых управляющих устройств. – М.: Энергоатомиздат, 1984. – 192 с.

5. Миллер Р. Теория переключательных схем. – М.: Наука, 1970. Т. 1. – 416 с.

6. Шальто А.А. Логическое управление. Методы аппаратной и программной реализации алгоритмов. – СПб.: Наука, 2000. – 780 с.

7. Шевелев Ю.П. Дискретная математика. – Томск: ТУСУР, 1998, 1999. – Ч. 1, 2. – 114, 120 с.

8. Шевелев Ю.П., Шидловский В.С. А.с. 1476456 (СССР). Ячейка однородной среды // Б.И. – 1989. – № 16.

9. Шидловский С.В. Изотропная среда в системе автоматизированного управления и контроля // Труды VI Междунар. науч.-практич. конф. студентов, аспирантов и молодых ученых «Современные техника и технологии». – Томск: Изд-во ТПУ, 2000. – С. 161–162.

10. Шидловский С.В. Многофункциональный логический модуль для реализации операций удаления аргументов из булевых функций // Радиотехнические устройства, информационные технологии и системы управления: Тез. докл. региональной науч.-техн. конф. студентов и молодых специалистов. – Томск, 2001. – С.15–16.

11. Шидловский С.В. Перестраиваемые структуры на многофункциональных логических модулях // Информационные системы: Труды постоянно действующей науч.-техн. школы-семинара студентов, аспирантов и молодых специалистов «Информационные системы мониторинга окружающей среды». – Томск: ТУСУР, 2003. – Вып. 2. – С.105–117.

12. Шидловский С.В. Ячейка однородной среды. Патент РФ на изобретение № 2251140 // БИ. – 2005. – № 12. – 9 с.

13. Шидловский С.В. Ячейка однородной среды. Патент РФ на изобретение № 2251141 // БИ. – 2005. – № 12. – 13 с.

14. Шидловский С.В., Светлаков А.А. Исследование функциональных возможностей многофункционального логического модуля, реализующего операции удаления аргументов из булевых функций // Вестник Сибирского отделения АН ВШ. – 2002. – №1(8). – С. 72–78.

15. Шидловский С.В., Светлаков А.А. Однородная среда в системе контроля знаний // Тез. докл. науч.-методич. конф. «Современное образование: Качество и новые технологии». – Томск: ТУСУР, 2000. – С. 69–70.

16. Шидловский С.В., Светлаков А.А., Шидловский В.С. Многофункциональный логический модуль для проблемно-ориентированных процессоров // Измерение, контроль, информатизация: Материалы Второй Междунар. науч.-техн. конф. – Барнаул, 2001. – С. 289–292.

17. Шидловский С.В., Светлаков А.А., Шидловский В.С. Преобразование структур данных в специализированных изотропных процессорах // Методы и

средства измерения в системах контроля и управления: Материалы Всерос. науч.-техн. конф. – Пенза, 2001. – С. 213–217.

18. *Шидловский С.В., Фоминых Ю.И.* Исследование функциональных возможностей однородной структуры, обеспечивающей вычисление неповторных ДНФ булевых функций // Вестник Сибирского отделения АН ВШ. – 2001. – № 1(7). – С. 50–59.

19. *Шидловский С.В., Шевелев Ю.П.* Автомат с перестраиваемой структурой в системе контроля знаний // VI Междунар. науч.-методич. конф. вузов и факультетов телекоммуникаций. – М., 2000. – С. 97–98.

20. *Якубайтис Э.А.* Логические автоматы и микромодули. – Рига: Зинатне, 1975. – 259 с.

21. *Shidlovskiy S.V.* Multifunctional Automaton for Computation of Disordered Boolean Functions // The VIII International Scientific and Practical Conference of Students, Post-graduates and Young Scientist «Modern Techniques and Technology» (МТТ'2002). – Tomsk: Tomsk Polytechnic University, 2002. – P. 40–41.

22. *Shidlovskiy S.V.* Computation of nonrecurrent ordered Boolean functions of higher than the second order // SIBEDEM 2002. Proceeding. – Tomsk: The Tomsk IEEE Chapter & Student Branch. Russia, 2002. – P. 45–48.

23. *Shidlovskiy S.V., Svetlakov A.A., Shidlovskiy V.S.* Computation of disordered Boolean functions // SIBEDEM 2002. Proceeding. – Tomsk: The Tomsk IEEE Chapter & Student Branch, 2002. – P. 41–44.