

# **ОДНОРОДНЫЕ СТРУКТУРЫ**

**Анализ. Синтез. Поведение**



**«ЭНЕРГИЯ»**

**МОСКВА 1973**

**6Ф.7**

**О 43**

**УДК 62.52·681.3**

**Авторы: В. И. Варшавский, В. Б. Мараховский, В. А. Песчанский,  
Л. Я. Розенблум.**

**О 43 Однородные структуры. (Анализ. Синтез. Поведение). М., «Энергия», 1973.**

**152 с. с ил.**

**На обороте тит л авт В. И. Варшавский, В. Б. Мараховский, В. А. Песчанский, Л. Я. Розенблум.**

**В книге рассматриваются новые методы построения вычислительных и управляющих схем, реализуемых на базе интегральной технологии. Особое внимание уделяется способам синтеза однородных цепочек и решеток из однотипных элементов с одинаковыми связями, а также анализу их поведения.**

**Книга рассчитана на широкий круг специалистов по автоматике и вычислительной технике, аспирантов и студентов соответствующих специальностей.**

**3-3-13**

**238-72**

**6Ф.7**

## **Предисловие**

В работе нашли свое отражение вопросы анализа и синтеза однородных структур, необходимость детального рассмотрения которых диктуется влиянием микроэлектроники на развитие теории автоматов и внутренними потребностями теории коллективного поведения.

Авторы сочли необходимым довольно подробно изложить наиболее существенные результаты работы Ф. Хенни «Итеративные логические сети».

Некоторые смежные вопросы, касающиеся однородных структур, читатель может найти в книгах [Л. 8, 16]. В работе использованы многие статьи советских и, главным образом, зарубежных авторов, включены также собственные результаты авторов. Помимо прилагаемого списка литературы читатель может использовать дополнительную библиографию, имеющуюся в [Л. 8, 16], а также в обзоре [Л. 44].

Авторы благодарят А. Я. Макаревского, написавшего § 4-6, Д. А. Поспелова, О. П. Кузнецова, В. П. Чистова, М. Г. Карповского, И. Н. Боголюбова, Б. Л. Овсиевича, обсуждавших затронутые в работе вопросы и высказавших ряд замечаний, а также Н. И. Глазунова, внимательное и тщательное редактирование которого способствовало улучшению качества книги.

## **Введение**

Рассматриваются схемы из одинаковых элементов или групп элементов, связанных между собой. Элементами являются конечные автоматы, в том числе автоматы без памяти, т. е. комбинационные схемы.

Бесконечная схема называется *однородной*, если она образована повторением данного элемента (или связанной группы элементов) с его связями группой конечных переносов (трансляцией вдоль связей).

*Однородной сетью* будем называть конечную часть однородной схемы, полученную разрезом по связям. Связи, попавшие в разрез, будем называть *границыми*, а сигналы, поступающие по этим связям, *границными сигналами*. На рис. В-1 приведены примеры однородных сетей.

В этой работе мы рассмотрим лишь некоторые типы однородных структур. Это в определенной степени оправдывает отказ от попытки формально определить понятие однородной бесконечной схемы и производного от него понятия однородной сети. Для этого потребовалось бы применить теоретико-групповые построения типа тех, которые используются в кристаллографии. Такую задачу авторы перед собой не ставили. Однородные сети будут классифицироваться по следующим признакам:

1. По типу связей:

а) по направленности связей — на

*односторонние сети* — сети, у которых подача выходного сигнала  $i$ -го элемента на вход  $j$ -го элемента исключает связь  $j$ -го элемента со входом  $i$ -го элемента;

*двусторонние* сети — сети, у которых подача выходного сигнала  $i$ -го элемента на вход  $j$ -го элемента требует подачи выхода  $j$ -го элемента на вход  $i$ -го элемента;

б) по кратности связей:

*кратность по входу* равна числу элементов, выходы которых являются входами одного и того же элемента;

*кратность по выходу* равна числу элементов, входами которых является выход данного элемента (у двусторонних сетей кратности по входу и по выходу совпадают).

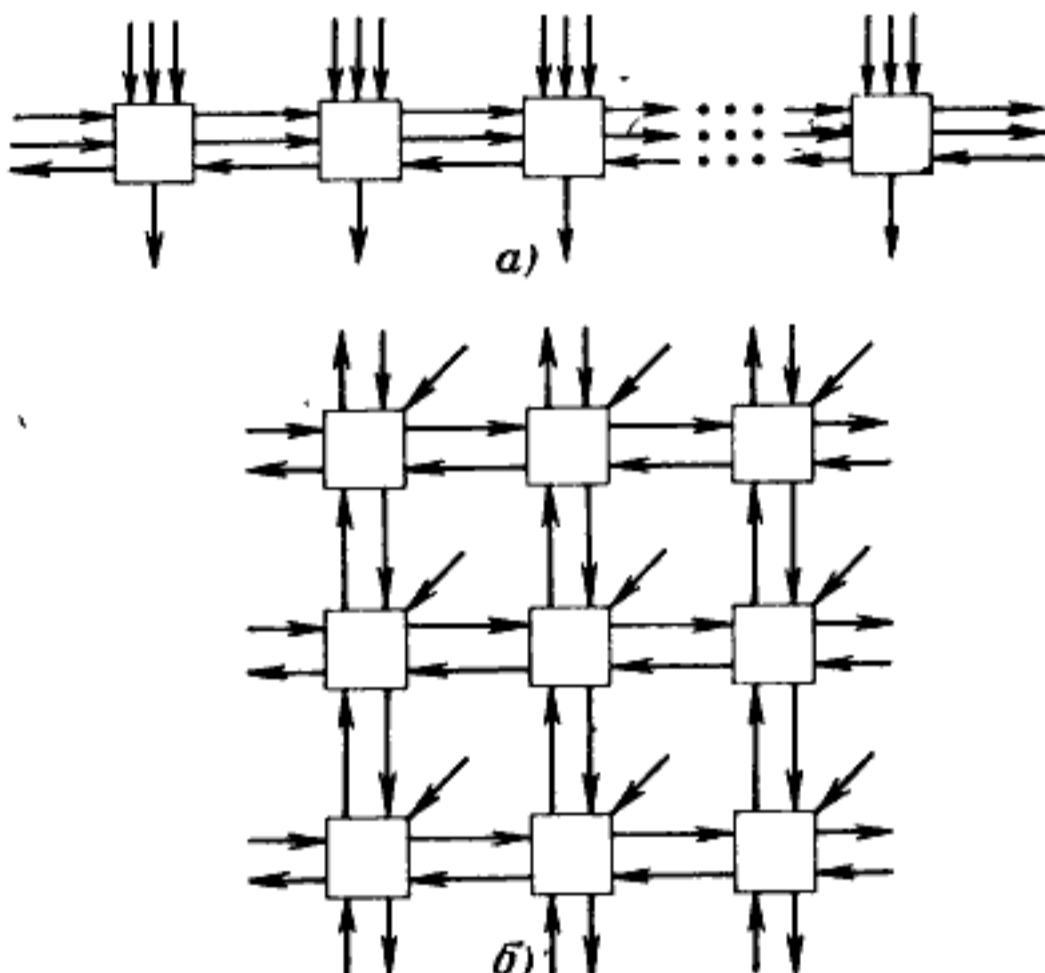


Рис. В-1. Примеры однородных сетей.

а — одномерная; б — двумерная

2. По размерности пространства в принятом способе расположения элементов сети: *одномерные*, обычно называемые *цепочками* (рис. В-1, а); *двумерные*, или *решетки* (рис. В-1, б); *трехмерные*. Классификация по этому признаку является в некоторой степени условной и выбирается в соответствии с технологическими особенностями реализации однородной сети.

3. По типу элементов: сети из элементов с памятью и сети из комбинационных элементов.

4. По топологии сети: замкнутые сети и сети с границами (для одномерных сетей — кольца и линейные сети, для двумерных — торы и плоские сети).

Все входы и выходы элемента, которые мы будем без нарушения общности считать двоичными, можно разделить на *межэлементные* (в цепочках — *боковые*) и *внешние*. На боковые входы элемента поступают сигналы с выходов других, непосредственно с ним связанных элементов. Если число возможных боковых сигналов равно  $m$ , то число двоичных входов элементов должно быть не менее  $\lceil \log_2 m \rceil$ . (Здесь через  $\lceil k \rceil$  обозначено целое число, ближайшее сверху к  $k$ .) В некоторых случаях нам будет удобно пользоваться термином *межэлементный (боковой) канал*, под которым будем понимать связь входа данного элемента с выходом другого элемента, предполагая, что по каналу передается двоичный сигнал. Остальные входы считаются *внешними*. Иногда внешние входы удобно делить на *информационные* и *настроечные*.

На информационные входы подаются входные двоичные переменные, на настроечные — сигналы настройки (константы), набор которых определяет режим работы настраиваемого элемента. При этом сети, содержащие лишь элементы без настроечных входов, в литературе часто называют *итеративными*.

Множество сетей из конечного числа элементов, полученных из бесконечной однородной схемы путем разрезов по связям в соответствии с определенным законом выбора границ, будем называть *однородной системой*. Одной двумерной системе могут принадлежать, например, все решетки с прямоугольными границами, вырезанные из одной и той же бесконечной схемы. Что касается одномерной системы, то в этом случае можно предложить рекурсивное ее определение. Возьмем элемент. Его можно соединить с другим таким же элементом, этот второй — с третьим и т. д. Множество целочек, составленных из  $1, 2, \dots, n, \dots$  элементов, будем называть *системой*. Таким образом, сети, принадлежащие одномерной системе, отличаются лишь числом элементов.

В зависимости от постановки задач будем различать одновходные сети и системы и многовходные. Сети и системы, в которых внешние входы отсутствуют, будем называть *автономными*.

Внутреннее состояние элемента (если он является элементом с памятью) и набор значений всех входов назовем *полным состоянием элемента*. Состояние одно-

родной сети описывается состоянием всех ее элементов. Из множества возможных состояний сети можно выделить некоторое ее фиксированное состояние, которое назовем начальным.

Для решения задач, рассматриваемых в этой работе, предположим, что однородная сеть функционирует таким образом, что внутреннее состояние любого ее элемента в момент времени  $t$  зависит только от входов и внутреннего состояния этого элемента в момент  $t-1$ ; эта единичная задержка определяет такт работы элемента. Будем также считать, что комбинационная часть элемента сети обладает такой временной задержкой, что суммарная задержка комбинационных частей всех элементов любой сети много меньше такта работы элемента.

Будем считать также, что переменные подаются на внешние входы сети одновременно. Значения переменных при этом не изменяются в течение некоторого времени работы сети. Длительность переходного процесса зависит от ряда характеристик сети, которые будут введены далее, и ее элементов. Иными словами, «вычисления» в сетях происходят не в реальном масштабе времени.

Если после подачи некоторого набора значений внешних входов сеть, находящаяся перед этим в некотором состоянии, устанавливается в определенное состояние, то такое состояние будем называть *равновесным*. Для достижения равновесного состояния требуется определенное время, зависящее от числа элементов сети и числа их внутренних состояний.

Если каждому набору значений внешних входов при любом фиксированном предыдущем состоянии сети соответствует одно и только одно равновесное состояние, то такая сеть называется *регулярной*. После того как регулярная сеть попала в равновесное состояние и считаны значения выходных переменных, на ее внешние входы можно подать новый набор входных переменных, не сбрасывая ее в начальное состояние.

Если при некотором наборе значений внешних входов сеть, находящаяся до этого в некотором состоянии, не достигает равновесного состояния, то говорят, что такая сеть имеет *цикл*. Если сеть не имеет цикла для каждого из возможных наборов значений входов, ее называют *стабильной*. До подачи нового набора входных

переменных стабильную сеть нужно принудительно перевести в начальное состояние.

Однородная система называется регулярной (стабильной), если все сети системы регулярны (стабильны).

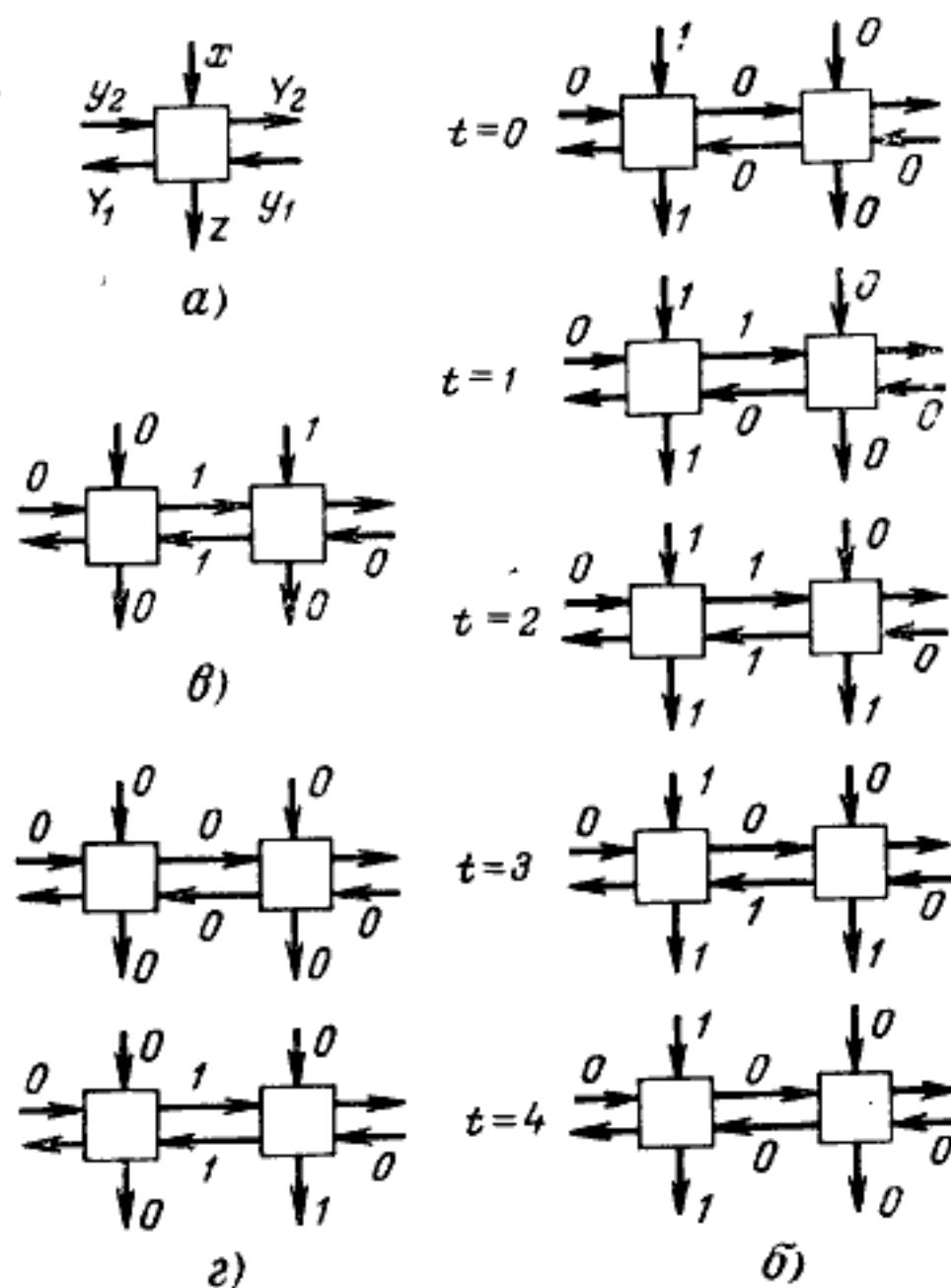


Рис. В-2. Пример сети, иллюстрирующий наличие равновесных состояний и циклов.

*а* — элемент сети, *б* — возникновение цикла, *в* — наличие единственного равновесного состояния, *г* — случай, когда возможны два состояния равновесия.

Из определений видно, что любая регулярная сеть является стабильной. Обратное утверждение в общем случае неверно.

Для пояснения введенных определений приведем пример сети (рис. В-2). Даны двусторонняя сеть из двух комбинационных элементов типа рис. В-2,*а*. Каждый элемент имеет три входа (внешний  $x$ , левый  $y_2$  и правый  $y_1$ ) и три выхода (внешний  $z$ , правый  $Y_2$  и левый  $Y_1$ ).

вый  $Y_1$ ). Поведение одного такого элемента представлено таблицей истинности

$x \ y_2 \ y_1$	$Y_1 \ Y_2 \ z$	$x \ y_2 \ y_1$	$Y_1 \ Y_2 \ z$
0 0 0	0 0 0	1 0 0	1 1 1
0 0 1	1 1 0	1 0 1	1 0 1
0 1 0	1 0 1	1 1 0	1 0 0
0 1 1	1 1 0	1 1 1	1 0 0

Заданы начальные условия: входы  $y_2$  левого крайнего и  $y_1$  крайнего правого элемента равны 0. Для того чтобы иметь возможность проанализировать поведение рассматриваемой сети, необходимо знать и начальные значения межэлементных (боковых) сигналов. Очевидно, что эту информацию получить трудно, потому что появление тех или иных значений межэлементных сигналов в начальный момент времени связано со способом технической реализации сети. Поэтому здесь, как и в общем случае, остается лишь рассмотреть все возможные комбинации значений межэлементных сигналов, позволяющие получить интересующие нас данные. Предположим сначала, что сигналы на межэлементных входах равны 0. Пусть набор значений внешних входов равен  $x_1=1, x_2=0$ . В соответствии с таблицей истинности определяем значения всех выходов каждого элемента в последовательные моменты времени. Поведение сети за 5 тактов приведено на рис. В-2,б. Видно, что состояния сети в первый и пятый моменты времени совпадают.

Следовательно, в этом случае сеть входит в цикл и, значит, является нестабильной. Это утверждение не означает, однако, что при некотором наборе значений внешних входов и значений межэлементных сигналов в начальный момент времени сеть не может прийти в состояние равновесия. В нашем примере для набора значений внешних входов  $x_1=0, x_2=1$  при начальных боковых сигналах, равных 1, сеть имеет единственное состояние равновесия (рис. В-2,в). Однако при наборе  $x_1=x_2=0$  в зависимости от значений межэлементных сигналов в начальный момент сеть имеет два состояния равновесия.

Приведенный пример не описывает всех возможных случаев возникновения циклов и равновесных состояний.

Можно привести пример сети, в которой при одном и том же наборе значений внешних входов могут наблюдаться как равновесные состояния, так и циклы, в зависимости от распределения межэлементных сигналов.

Остается привести пример стабильной, но нерегулярной сети. Пусть задана однородная одномерная цепочка из двух элементов с памятью, поведение каждого из которых полностью описывается таблицей

$x = 0$		$x = 1$	
$A$	$B$	$A$	$B$
$S_0$	$A, S_0, 1$	$B, S_0, 0$	$B, S_1, 0$
$S_1$	$B, S_0, 0$	$B, S_1, 0$	$A, S_1, 0$

Запись  $A, S_0, 1$  в верхнем левом квадрате таблицы означает, что если начальное внутреннее состояние элемента  $A$ , внешний вход  $x=0$  и боковой вход  $S_0$ , то элемент останется в состоянии  $A$ , выдаст в качестве бокового выхода сигнал  $S_0$ , а его внешний выход будет равен единице.

Зададимся граничным сигналом  $S_0$ . Читатель может убедиться в том, что эта сеть ни при каких условиях не входит в цикл, пользуясь способом предыдущего примера. В то же время легко установить, что данная сеть не является регулярной. Действительно, при комбинации значений внешних входов  $x_1=1, x_2=0$  на внешних выходах сети устанавливаются значения  $z_1=0, z_2=0$ , если внутренние состояния элементов  $AA$ , и  $z_1=0, z_2=1$ , если внутренние состояния  $BA$  (рис. В-3). Следовательно, сеть не регулярна.

Предметом исследования в работе является анализ поведения рассматриваемого класса структур, выяснение их функциональных возможностей и решение некоторого набора задач, которые нас будут интересовать. Под анализом мы будем понимать установление факта стабильности и регулярности систем.

Проверка соответствия синтезированной сети заданному закону функционирования (при соблюдении установленных ограничений) также является традиционной задачей анализа, однако в этой работе она не рассматривается. Важным этапом является выяснение функциональных возможностей разных типов однородных

структур. В частности, представляет интерес решение задач о реализуемости произвольной логической функции или определенного класса логических функций системами из некоторого числа элементов при определенных ограничениях.

Дадим здесь обобщенные неформальные постановки задач, преследуя цель предварительно обрисовать круг вопросов, нашедших отражение в работе.

1. Задана булева функция  $n$  переменных. Указан набор признаков, приведенных в классификации, уст-

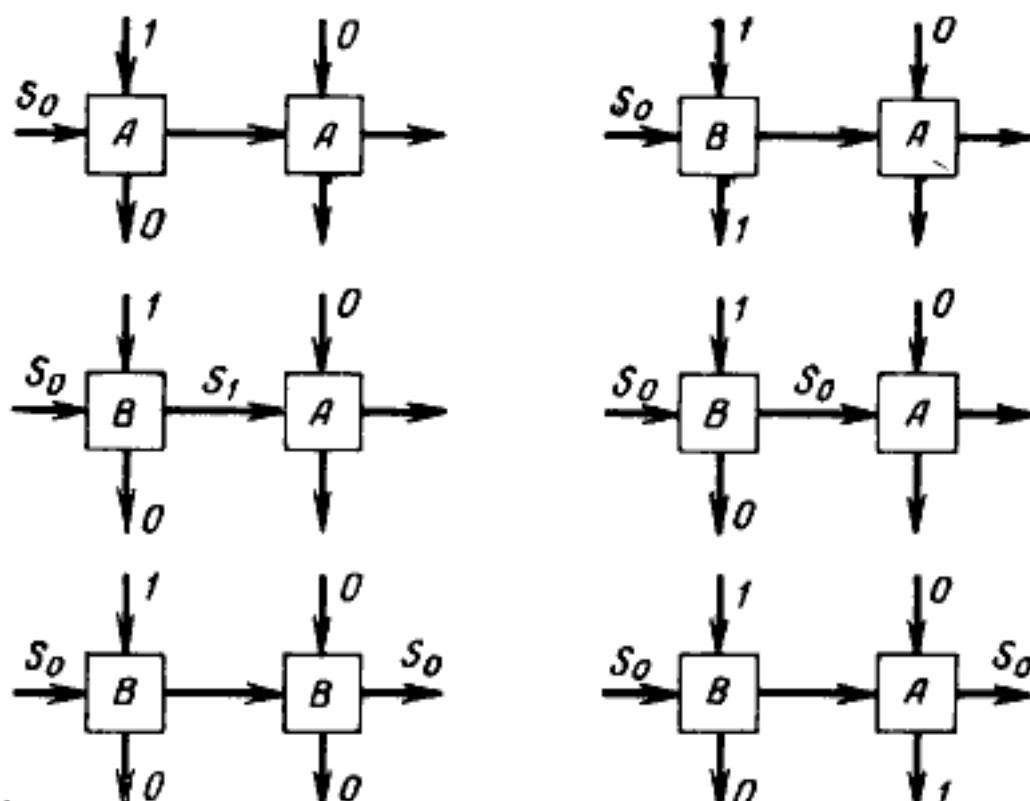


Рис. В-3. Пример стабильной, но нерегулярной сети.

авливающий ограничения на используемые типы элементов и вид сети. Требуется определить реализуемость этой функции при заданных ограничениях; найти ее реализацию, если она существует.

Задачи этого типа являются задачами структурного синтеза. Изложение методов решения этих задач составляет основное содержание гл. 1 (кроме § 1-7 и частично § 1-1) и гл. 4 (кроме § 4-1).

2. На внешние входы каждой сети системы подается некоторая конфигурация значений входных переменных. Под словом конфигурация в одномерном случае здесь понимается какой-либо набор значений внешних переменных (двоичный код), в двумерном — двоичная матрица, элементы которой компонуют некоторый образ, в трехмерном — некоторое объемное изображение, элементы которого также кодируются символами 1 и 0.

Синтезировать элемент системы — такой, что сеть, составленная из некоторого количества этих элементов, позволяет: опознать требуемые свойства предъявляемых конфигураций; установить локальные или позиционные свойства конфигураций, определенные условиями задачи; преобразовать конфигурации по заданному закону.

3. Синтезировать элемент системы — такой, что сеть из этих элементов выдает сигнал о наличии некоторых интересующих нас соотношений между определяющими ее характеристиками.

4. Синтезировать элемент системы — такой, что сеть из этих элементов позволяет в некоторый момент времени синхронизировать ее элементы<sup>1</sup>.

Задачи типов 2—4 следует отметить особо. Они обладают определенной спецификой и, по-видимому, окажутся несколько непривычными даже для читателя, хорошо знакомого с традиционными разделами теории автоматов. Дело в том, что в этих задачах требуется получить такое решение, которое справедливо не только для какой-либо сети, содержащей фиксированное число элементов, но справедливое для всех сетей системы без относительно к числу содержащихся в ней элементов. Иначе говоря, полученное решение какой-либо задачи является решением целого класса задач. Сейчас вряд ли имеет смысл останавливаться на этом подробнее. Сделаем попытку разобраться во всех возникающих вопросах в гл. 2 и 3, а также в § 1-1 (частично), 1-7 и 4-1. Будем считать, что при решении задач типов 2—4 можно ограничиться лишь этапом абстрактного синтеза, т. е. получением таблицы функционирования элемента системы. Переход от такого задания элемента к получению его структурной схемы может быть выполнен обычными методами и в работе не рассматривается.

---

<sup>1</sup> Задачи этих типов решаются автономными системами.

## Глава первая

# ОДНОСТОРОННИЕ ОДНОМЕРНЫЕ СЕТИ И СИСТЕМЫ ИЗ КОМБИНАЦИОННЫХ ЭЛЕМЕНТОВ

Рассмотрим одностороннюю цель (рис. 1-1), состоящую из фиксированного числа  $n$  комбинационных элементов. На каждый ее элемент подается соответствующая переменная  $x_i$  ( $1 \leq i \leq n$ ). Пусть боковые сигналы могут принимать любые значения из множества  $S = \{S_0, \dots, S_{k-1}\}$ . Если в момент времени  $t$  на боковой вход любого элемента цепи поступает сигнал  $S(t)$ , а на внешний вход — переменная  $x_i$ , то на боковом выходе этого элемента в момент времени  $t+1$  появится сигнал  $S(t+1)$ , причем  $S(t+1) = f(S(t), x_i(t))$ .

В принципе вместо  $x_i(t)$  можно было написать просто  $x_i$ , не отмечая времени, поскольку набор внешних переменных подается на все элементы цепи в начальный момент времени и стоит до тех пор, пока не будет получен результат, т. е.  $x_i(t=0) = x_i(t=1) = \dots = x_i(t=n)$ . Таким образом, при подаче внешних входных переменных и граничном условии  $S_0 = S(t=0)$  на боковом выходе первого слева элемента в момент времени  $t=1$  появится сигнал

$$S(t=1) = f(S(t=0), x_1(t=0)),$$

на боковом выходе второго в момент  $t=2$

$$S(t=2) = f(S(t=1), x_2(t=1)) \text{ и т. д.}$$

За  $n$  тактов работы на боковом выходе  $n$ -го элемента цепи появится боковой сигнал  $S(t=n)$ .

Теперь рассмотрим некоторый конечный автомат Мура, состояния которого принадлежат множеству  $S$ ,

причем  $S_0$  есть начальное состояние. Выражение для функции перехода его имеет тот же вид, что и выражение для бокового выхода элемента цепи:

$$S(t+1) = F(S(t), x_i(t)).$$

Если положить  $F=f$ , то при подаче на вход этого автомата входной последовательности, состоящей из  $n$  переменных, автомат через  $n$  тактов также установится в состоянии  $S(n)$ . Для этого автомата Мура можно найти эквивалентный ему автомат Мили<sup>1</sup>.

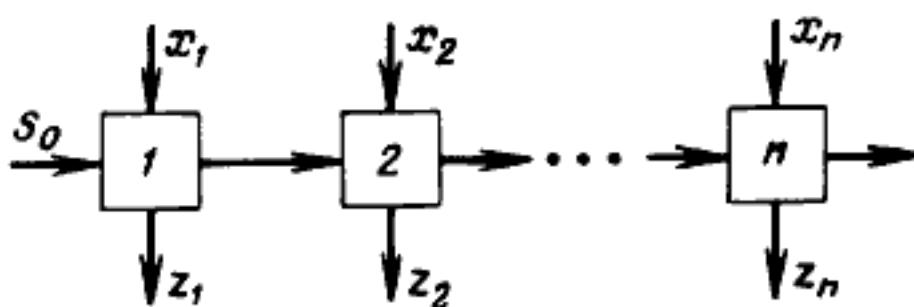


Рис. 1-1. Односторонняя сеть из комбинационных элементов.

Таким образом, цепи с фиксированным числом элементов можно сопоставить конечный автомат с фиксированной длиной входной последовательности, причем ее длина равна числу элементов в цепи [Л. 39, 21]. Такая цепь и соответствующий ей конечный автомат Мили обладают одинаковым поведением в том смысле, что, подавая на вход цепи какой-либо код параллельно или на вход автомата соответствующую последовательность, получаем на выходах цепи и автомата одинаковые сигналы.

Односторонним системам из комбинационных элементов можно поставить в соответствие конечный автомат с произвольной длиной входной последовательности и обратно: конечному автомatu с произвольной длиной входной последовательности можно поставить в соответствие одностороннюю систему из комбинационных элементов. Отсюда вытекает, что функциональные возможности односторонних одномерных систем из комбинационных

<sup>1</sup> Определения автоматов Мура и Мили можно найти, например, в книге В. М. Глушкова «Синтез цифровых автоматов», Физматгиз, 1962.

элементов и конечных автоматов совпадают. Можно утверждать, что любой язык типа 3 (язык регулярных событий) [Л. 7] представим такими системами.

Необходимости в анализе односторонних одномерных систем из комбинационных элементов не возникает; они регулярны, так как в них нет путей обратной связи.

В § 1-2—1-6 рассматривается поведение сетей при различных ограничениях, а в § 1-1, 1-7 поставлены и решены некоторые задачи для однородных систем.

### **1-1. Одноканальные сети и системы из ненастраиваемых элементов**

Рассмотрим одновыходные системы из комбинационных элементов с одним двоичным боковым каналом и одним двоичным внешним входом на каждый элемент; внешний выход каждого элемента совпадает с боковым. Существует 16 таких систем, соответствующих 16 функциям двух переменных. Интерес представляют только 10 функций, существенно зависящих от двух переменных. Очевидно, что такие системы реализуют только функции типа

$$f(x_n, f(x_{n-1}, f(x_{n-2} \dots f(x_1, S_0) \dots)).$$

К таким функциям принадлежат функции

$$f_1 = \bigvee_{j=1}^n x_j$$

(границный сигнал  $S_0 = 0$ );

$$f_2 = \bar{f}_1, (S_0 = 1);$$

$$f_3 = \bigvee_{j=1}^n \bar{x}_j, (S_0 = 0);$$

$$f_4 = \bar{f}_3, (S_0 = 1);$$

$$f_5 = \sum_{j=1}^n x_j \pmod{2}, (S_0 = 0);$$

$$f_6 = \bar{f}_5, (S_0 = 1);$$

$$f_7 = x_n \vee \bar{x}_{n-1} (x_{n-2} \vee \bar{x}_{n-3} (x_{n-4} \vee$$

$$\vee \bar{x}_{n-5} (x_{n-6} \vee \dots) \dots), (S_0 = 1).$$

Функция  $\bar{f}_7(x_1, \dots, x_n)$  является пороговой функцией<sup>1</sup> вида

$$\bar{f}_7(x_1, \dots, x_n) = \text{sign} \left( \sum_{j=1}^n (-1)^{n-j} \varphi_j x_j - (1 - (-1)^n)/2 \right),$$

где  $\varphi_j$  —  $j$ -е число Фибоначи ( $\varphi_j = \varphi_{j-1} + \varphi_{j-2}$ ,  $\varphi_1 = \varphi_2 = 1$ ),

$$f_8 = \bar{f}_7, (S_0 = 0);$$

$$\begin{aligned} f_9(x_1, \dots, x_n) &= \bar{x}_n \vee x_{n-1} (\bar{x}_{n-2} \vee \\ &\vee x_{n-3}) (\bar{x}_{n-4} \vee x_{n-5} (\bar{x}_{n-6} \vee \dots)) = \\ &= \text{sign} \left( - \sum_{j=1}^n (-1)^{n-j} \varphi_j x_j + \varphi_{n-1} - \right. \\ &\quad \left. - (1 + (-1)^n)/2 \right), (S_0 = 0); \\ f_{10} &= \bar{f}_9, (S_0 = 1). \end{aligned}$$

Таким образом, мы перечислили все задачи, которые могут быть решены системами с одним боковым каналом и двузначным входом. Как ясно из определения систем, переменные подаются на внешние входы элементов сетей, принадлежащих системам, лишь однократно. Метод синтеза сетей таких систем является частным случаем метода, изложенного в § 1-2.

Исследуем теперь возможности повторных сетей, к которым отнесем сети, допускающие повторную подачу переменных или подачу вместо некоторых переменных констант, и сравним их с возможностями рассмотренных выше систем. Но прежде напомним следующие достаточно известные определения.

*Остаточными функциями* будем называть функции, образованные из данной логической функции фиксацией ее переменных (т. е. заменой этих переменных константами 0 и 1) или заменой одних переменных другими или их инверсиями (отождествлением ее переменных).

<sup>1</sup> Определение пороговой функции можно найти, например, в книге Д. А. Поспелова «Логические методы анализа и синтеза схем», М., «Энергия», 1968.

Примем следующие обозначения:

$$f(x_i=1) = f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n),$$

$$f(x_i=0) = f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n),$$

$$f(x_i=x_j) = f(x_1, \dots, x_{i-1}, x_j, x_{i+1}, \dots, x_{j-1}, \\ x_j, x_{j+1}, \dots, x_n),$$

$$f(x_i=\bar{x}_j) = f(x_1, \dots, x_{i-1}, \bar{x}_j, x_{i+1}, \dots, x_{j-1}, \bar{x}_j, \\ x_{j+1}, \dots, x_n).$$

Первые две функции образованы фиксациями переменной  $x_i$ , вторые — тождествением переменных  $x_i$  и  $x_j$ . Будем говорить, что функция  $f(x_i=1)$  *парна* функции  $f(x_i=0)$ , а  $f(x_i=x_j)$  ларна функции  $f(x_i=\bar{x}_j)$ , и наоборот. Если функция  $f(x_1, \dots, x_n)$  существенно зависит от всех  $n$  аргументов, то при фиксации или замене  $k$  переменных число остаточных функций в общем случае равно  $2^k$ , хотя некоторые из них могут оказаться одинаковыми.

Итак, мы займемся выяснением следующего вопроса: дает ли повторная подача переменных или подача вместо некоторых переменных констант какой-либо эффект, т. е. позволяет ли она реализовать хотя бы некоторые классы функций, которые не могут быть представлены бесповторными сетями систем  $f_1-f_{10}$ ?

Рассмотрим сеть из  $n$  элементов, принадлежащую системе  $f_1$ . Очевидно, что если на какие-либо два элемента этой сети  $i$  и  $j$ ,  $i \neq j$  подается одна и та же переменная, скажем  $x_i$ , то сеть реализует функцию

$$\bigvee_{k=1, k \neq i}^n x_k, \text{ т. е. дизъюнкцию переменных } x_1, \dots, x_{j-1},$$

$x_{j+1}, \dots, x_n$ . Такую же функцию реализует сеть системы  $f_1$ , состоящая из  $n-1$  элемента. Естественно, что при многократной подаче одной и той же переменной такие сети могут реализовать лишь те функции, которые реализуются сетями системы  $f_1$ .

Подача константы 0 равносильна повторной подаче любой переменной, подача константы 1 приводит к реализации тождественной единицы.

Аналогично можно убедиться в том, что повторная подача переменных и подача констант в сетях систем

$f_2-f_6$  не приводит к реализации каких-либо новых функций, за исключением констант

Теперь рассмотрим сеть из  $n$  элементов, принадлежащую системе  $f_7$ . Оказывается, что подача любой переменной на входы двух элементов такой сети равносильна подаче константы 0 на вход одного из этих двух элементов (а именно элемента с меньшим номером), поскольку справедливо равенство

$$f_7(x_k=x_{k-j})=f_7(x_{k-j}=0),$$

$$1 \leq k \leq n, 1 \leq j \leq k-1.$$

Аналогичное справедливо и для сети из  $n$  элементов, принадлежащей системе  $f_9$ , поскольку

$$f_9(x_k=x_{k-j})=f_9(x_{k-j}=1),$$

$$1 \leq k \leq n, 1 \leq j \leq k-1.$$

Тот же эффект получается при повторной подаче переменных или подаче констант вместо них в сетях систем  $f_8$  и  $f_{10}$ , описываемых функциями, являющимися инверсиями функций  $f_7$  и  $f_9$  соответственно.

Заметим, что отождествление или фиксация переменных приводит к получению некоторых новых функций, относительно которых можно сказать, что они принадлежат классу пороговых функций, поскольку функции  $f_7-f_{10}$  являются пороговыми. Из пороговой логики известно, что остаточные функции пороговых функций также являются пороговыми.

Далее выясним еще один вопрос: не пересекаются ли классы реализуемых функций в сетях типов  $f_1-f_{10}$ , если допустить повторную подачу переменных и их фиксацию? Оказывается, что функциональные возможности сетей, принадлежащих системам  $f_1-f_{10}$ , полностью покрываются возможностями сетей трех типов:  $f_5$ ,  $f_7$  и  $f_9$ . Иначе говоря, имея элементы лишь трех типов, можно построить повторные сети, реализующие все функции, представимые сетями одноканальных систем рассматриваемого типа.

Не давая дополнительных разъяснений, приведем здесь формулы, доказывающие сделанное утверждение:

$$f_5(x_1, \dots, x_n, x_{n+1}=1)=f_6(x_1, \dots, x_n);$$

$$f_7(x_1, \dots, x_n, x_{n+1}=0)=\bar{f}_7(x_1, \dots, x_n)=f_8(x_1, \dots, x_n);$$

$$f_7(x_1, \dots, x_{2n}, x_1=x_3=\dots=x_{2n-1}=0)=f_1(x_1, \dots, x_n);$$

$$\begin{aligned} f_7(x_1, \dots, x_{2n}, x_2=x_4=\dots=x_{2n}=0) = \\ = \tilde{f}_1(x_1, \dots, x_n) = f_2(x_1, \dots, x_n); \end{aligned}$$

$$f_9(x_1, \dots, x_{n+1}, x_{n+1}=1)=f_{10}(x_1, \dots, x_n);$$

$$\begin{aligned} f_9(x_1, \dots, x_{2n}, x_1=x_3=\dots=x_{2n-1}=1)=f_3(x_1, \dots, x_n); \\ f_9(x_1, \dots, x_{2n}, x_2=x_4=\dots=x_{2n}=1)=\tilde{f}_3(x_1, \dots, x_n)= \\ = \tilde{f}_4(x_1, \dots, x_n). \end{aligned}$$

## 1-2. Одноканальные бесповторные сети из настраиваемых элементов

В качестве элемента структуры возьмём элемент, реализующий любую из 16 логических функций двух переменных. Настроечные входы (на которые подаются константы) служат для выбора одной из этих 16 логических функций. В этом случае число настроечных входов должно быть равно 4 ( $2^4=16$ ). Если допустить возможность подачи на внешний вход элемента как переменной, так и ее инверсии, то, как легко показать, можно использовать элемент, который реализует лишь три логические операции:  $\wedge$ ,  $\vee$ ,  $\oplus$  (конъюнкцию, дизъюнкцию и сумму по модулю 2 соответственно). Если подача инверсий переменных не допускается, то введением еще одного настроечного входа  $\sigma$  можно это сделать:  $\tilde{x}_j = x_j \oplus \sigma$ ;  $\sigma = 0,1$ . В литературе такие сети часто называют майтровскими каскадами или трибутарными (от англ. tributary — приток).

Таким образом, в этом параграфе задача синтеза однородных сетей сводится к синтезу сетей, составленных из элементов трех типов. В таком случае настроечные входы не будем принимать во внимание. Кроме того, сделаем допущение: вместо постоянного граничного условия (константы) будем подавать на боковой вход крайнего левого элемента переменную.

Ограничимся сначала рассмотрением одномерных сетей, в которых каждая переменная  $x_j$  ( $1 \leq j \leq n$ ) подается только на один (внешний) вход  $(j-1)$ -го элемента. (Исключение составляет только крайний левый элемент, на который подаются две переменные.) Такую схему

естественно назвать *бесповторной*. Вообще говоря, входы  $x_1, \dots, x_n$  схемы могут быть упорядочены произвольным образом. Поведение схемы, изображенной на рис. 1-2, описывается выражением

$$f(x_1, \dots, x_n) = f_{n-1}(x_n, f_{n-2}(x_{n-1}, f_{n-3}(\dots(f_3(x_3, f_2(x_2, x_1)) \dots))),$$

где через  $f_k$  обозначена функция на выходе  $k$ -го элемента, причем если  $f(x_1, \dots, x_n)$  существенно зависит от

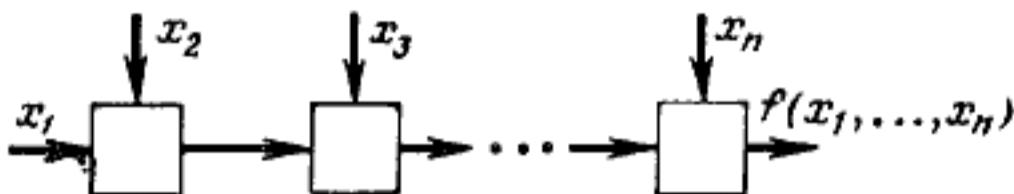


Рис. 1-2. Бесповторная одноканальная сеть из настраиваемых элементов.

всех своих переменных, то каждая из функций  $f_k$  ( $1 \leq k \leq n-1$ ) должна существенно зависеть от  $k+1$  переменных.

**Логические возможности.** Первый вопрос, возникающий при рассмотрении одномерных бесповторных схем, состоит в том, является ли такая схема универсальной, т. е. может ли быть реализована посредством такой схемы любая логическая функция  $n$  переменных. На этот вопрос можно дать отрицательный ответ. Например, мажоритарная функция

$$\text{maj } x_1, x_2, x_3 = x_1 x_2 \vee x_1 x_3 \vee x_2 x_3$$

не может быть реализована двухкаскадной одномерной бесповторной схемой, что следует из решения примера 1-2.

Легко показать, что относительное число функций, реализуемых одномерными бесповторными схемами, стремится к нулю при  $n \rightarrow \infty$ .

**Реализуемость.** Поскольку не любая логическая функция может быть реализована одномерной бесповторной схемой, то возникают вопросы, как по заданной логической функции определить, реализуется ли она такой схемой и каковы признаки реализуемости.

Приведенная ниже процедура позволяет определить, является ли заданная функция реализуемой или нереализуемой. В первом случае процедура дает метод синтеза бесповторной сети.

**Синтез.** Процедура синтеза основана на приведенных ниже теоремах.

**Теорема 1-1.** Для того чтобы логическая функция  $f(x_1, \dots, x_n)$  допускала тривиальную декомпозицию относительно переменной  $x_i$  с каким-либо из образов декомпозиции  $\vee$ ,  $\wedge$  и  $\oplus$ , необходимо и достаточно, чтобы выполнялось хотя бы одно из двух условий: а) одна из остаточных функций  $f(x_i=1)$  или  $f(x_i=0)$  обращалась в константу; б)  $f(x_i=0) = \bar{f}(x_i=1)$ . (*Тривиальной декомпозицией* логической функции  $f(x_1, \dots, x_n)$  относительно переменной  $x_j$  называется ее представление в виде  $f(x_1, \dots, x_n) = F(x_j, g(x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n))$ . При этом  $F$  называется *образом декомпозиции*, а  $f(x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n)$  — *составляющей декомпозиции*.)

Доказательство этой теоремы с очевидностью следует из справедливости разложения Шеннона:

$$f(x_1, \dots, x_n) = x_i f(x_i=1) \vee \bar{x}_i f(x_i=0)$$

и легко может быть получено читателем.

Из теоремы вытекает следствие:

по значению остаточной функции вид тривиальной декомпозиции может быть определен (если она возможна) следующим образом:

если  $f(x_i=0) = 0$ , то  $f = x_i f(x_i=1)$ ;

если  $f(x_i=1) = 0$ , то  $f = \bar{x}_i f(x_i=0)$ ;

если  $f(x_i=0) = 1$ , то  $f = \bar{x}_i \vee f(x_i=1)$ ;

если  $f(x_i=1) = 1$ , то  $f = x_i \vee f(x_i=0)$ ;

если  $f(x_i=0) = \bar{f}(x_i=1)$ , то  $f = x_i \oplus f(x_i=0)$  или  
 $f = \bar{x}_i \oplus f(x_i=1)$ .

**Теорема 1-2.** Если логическая функция  $f(x_1, \dots, x_n)$ , существенно зависящая от  $n$  переменных, представима в одном из следующих видов:

$$f(x_1, \dots, x_n) = x_i^{\sigma_i} \vee f(x_i^{\sigma_i}=0);$$

$$f(x_1, \dots, x_n) = x_j^{\sigma_j} f(x_j^{\sigma_j}=1);$$

$$f(x_1, \dots, x_n) = x_k^{\sigma_k} \oplus f(x_k^{\sigma_k}=0),$$

то она не может быть представлена ни в одном из других двух видов. Справедливость этой теоремы была до-

казана А. В. Кузнецовым<sup>1</sup>. Наконец, следует также подчеркнуть наличие еще одного факта. Если заданная функция реализуется в виде

$$f(x_1, \dots, x_n) = F[x_i^{\sigma_i}, f_1(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)]$$

и в виде

$$f(x_1, \dots, x_n) = F[x_j^{\sigma_j}, f_2(x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n)],$$

то, если реализуема остаточная функция  $f_1$ , остаточная функция  $f_2$  также реализуема, и наоборот. Отсюда следует, что порядок выбора переменных безразличен и не влияет на факт реализуемости.

Синтез ведется «справа налево», от выхода схемы, последовательным исключением переменных, как это делается в «методе каскадов».

Основные этапы процедуры синтеза заключаются в следующем.

1. Исходную функцию или остаточную функцию, полученную в последующих шагах, проверяем на существенную зависимость от всех своих переменных. Для этого остаточная функция должна быть отлична от своей парной. В дальнейшем рассматриваются остаточные функции, полученные фиксацией только тех переменных, от которых они зависят существенно. Подсчитывается число существенных переменных. Если оно больше двух, переходим к выполнению п. 2; если равно двум, то процедуру синтеза заканчиваем. Зафиксируем результаты (вид образа декомпозиции и переменную) в соответствии с приведенным следствием из теоремы 1-1.

2. Каждая из остаточных функций, образованных заменой существенных переменных на 1 и 0, последовательно сравнивается с константами 1 и 0. Если одна из них оказалась равной константе, то это обстоятельство фиксируется, после чего вновь переходим к выполнению п. 1, рассматривая остаточную функцию, парную той, которая равна константе. В противном случае переходим к выполнению п. 3.

3. Одна из парных остаточных функций инвертируется и сравнивается с парной ей функцией. Если равенст-

<sup>1</sup> Кузнецов А. В. Об одном свойстве функций, реализуемых неплоскими бесповторными схемами. Труды МИАН, т. 51, 1958.

во не выполняется ни для одной из существенных переменных, то исходная функция не реализуется одномерной бесповоротной схемой. Если для какой-либо переменной это равенство выполняется, фиксируем это обстоятельство и переходим к выполнению п. 1, рассматривая одну из парных остаточных функций.

Пример 1-2. Построить, если это возможно, одномерную бесповоротную схему, реализующую следующую логическую функцию:

$$f(x_1, \dots, x_6) = \bar{x}_3 x_6 \vee x_2 x_3 \bar{x}_6 \vee x_1 x_3 x_5 \bar{x}_6 \vee x_3 \bar{x}_4 x_5 \bar{x}_6 \vee \\ \vee \bar{x}_1 x_3 x_4 \bar{x}_5 \bar{x}_6 \vee x_1 \bar{x}_2 \bar{x}_3 \bar{x}_5 \vee \bar{x}_2 \bar{x}_3 \bar{x}_4 \bar{x}_5 \vee \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 x_5.$$

Легко убедиться в том, что заданная функция существенно зависит от всех переменных, поскольку

$$f(x_i=1) \neq f(x_i=0), \quad 1 \leq i \leq 6.$$

Ни одна из остаточных функций  $f(x_i^{\sigma_i} = 1)$ ,  $1 \leq i \leq 6$ , не обращается в константу. Условие

$$f(x_i=1) = \bar{f}(x_i=0)$$

выполняется лишь для одной пары остаточных функций:

$$f(x_3=1) = \bar{f}(x_3=0),$$

так как

$$f(x_3=1) = x_2 \bar{x}_6 \vee x_1 x_3 \bar{x}_6 \vee \bar{x}_4 x_5 \bar{x}_6 \vee \bar{x}_1 x_4 \bar{x}_5 \bar{x}_6;$$

$$f(x_3=0) = x_6 \vee x_1 \bar{x}_2 \bar{x}_5 \vee \bar{x}_2 \bar{x}_4 \bar{x}_5 \vee \bar{x}_1 \bar{x}_2 x_4 x_5.$$

Таким образом, исходная функция может быть тривиально декомпозирована относительно переменной  $x_3$ , причем в соответствии со следствием из теоремы

$$f(x_1, \dots, x_6) = x_3 \oplus f(x_3=0).$$

Далее обратим внимание на то, что функция  $f(x_3=0)$  в свою очередь может быть представлена в виде

$$f(x_3=0) = x_6 \vee f(x_3=0, x_6=0),$$

поскольку  $f(x_3=0, x_6=1)=1$ . Далее вследствие того что

$$f(x_3=0, x_6=0) = \bar{x}_1 \bar{x}_2 x_4 x_5 \vee x_1 \bar{x}_2 \bar{x}_5 \vee \bar{x}_2 \bar{x}_4 \bar{x}_5 = \\ = \bar{x}_2 (x_1 x_4 x_5 \vee x_1 \bar{x}_5 \vee \bar{x}_4 \bar{x}_5)$$

и

$$f(x_3=0, x_6=0, x_2=1)=0,$$

$$f(x_3=0, x_6=0, x_2=0) = \bar{x}_2 f(x_3=0, x_6=0, x_2=0)$$

и

$$f(x_3=0, x_6=0, x_2=0) = \bar{x}_1 x_4 x_5 \vee x_1 \bar{x}_5 \vee x_4 \bar{x}_5.$$

В этой последней остаточной функции можно выделить переменную  $x_5$ , что возможно, поскольку

$$f(x_3=0, x_6=0, x_2=0, x_5=1) = \bar{x}_1 x_4 = \\ = \bar{f}(x_3=0, x_6=0, x_2=0, x_5=0).$$

Таким образом,

$$f(x_3=0, x_6=0, x_2=0) = x_5 \oplus f(x_3=0, \\ x_6=0, x_2=0, x_5=0) = x_5 \oplus (x_1 \vee \bar{x}_4).$$

Наконец,  $f(x_3=0, x_6=0, x_2=0, x_5=0) = x_1 \vee \bar{x}_4$  реализуется одним элементом  $\vee$ .

В результате имеем следующую одномерную бесповторную схему (рис. 1-3).

**Пример 1-3.**  $f(x_1, x_2, x_3) = x_1 x_2 \vee x_1 x_3 \vee x_2 x_3$ . Ни одна из остаточных функций, полученных фиксацией переменной  $x_1$ , не обращается в константу:  $f(x_1=1) = x_2 \vee x_3$ ;  $f(x_1=0) = x_2 x_3$ . Остается выполнить проверку по п. 3 процедуры. Видно, что  $f(x_1=1) \neq \bar{f}(x_1=0)$ .

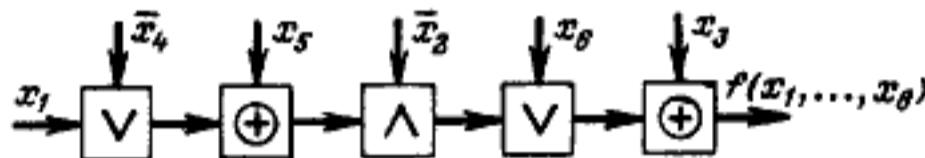


Рис. 1-3. Пример реализации логической функции бесповторной сетью.

Соответствующая проверка двух других пар остаточных функций, полученных фиксацией переменных  $x_2$  и  $x_3$ , может быть опущена вследствие того, что мажоритарная функция инвариантна по всем своим переменным. Следовательно, заданная функция не реализуется одномерной бесповторной схемой.

Схема, реализующая функцию  $n$  переменных, содержит точно  $n-1$  элемент, поэтому проблемы минимизации (по числу элементов) в классе одномерных бесповторных схем не существует. Однако из этого утверждения еще не следует, что реализуемой функции соответствует одна сеть.

Участки одномерной бесповторной схемы, состоящие из одноименных элементов, будем называть *итеративными участками*. Поскольку каждая из операций  $\vee$ ,  $\wedge$  и  $\oplus$  коммутативна, то в пределах итеративного участка порядок следования переменных может быть выбран произвольно, что не допускается в отношении переменных, принадлежащих различным участкам. Отсюда следует, что если найдена какая-либо одномерная бесповторная сеть, соответствующая заданной функции, то

другие реализующие ту же функцию сети могут быть найдены перестановками переменных внутри итеративных участков сети. Кроме того, различные вариации сетей могут быть получены применением правил де Моргана всеми возможными способами.

Рекуррентное соотношение для нахождения числа представителей классов эквивалентности функций, существенно зависящих от  $n$  переменных и реализуемых одномерными бесповторными сетями, найдено в [Л. 50]:

$$M(n) = 3M(n-1) - M(n-2).$$

При этом принята так называемая PN-классификация, в которой в один класс эквивалентности объединяются функции, полученные из представителя всеми возможными перестановками и инверсиями переменных. Приближенное решение этого соотношения равно для больших  $n$  [Л. 50]:

$$M(n) \approx ((2,62)^n)/2,24, M(1)=1, M(2)=3.$$

Число представителей функций, существенно зависящих от 2, 3 и 4 переменных и реализуемых бесповторными ( $M(n)$ ) и повторными ( $T(n)$ ) сетями, приведено в следующей таблице:

$n$	$M(n)$	$T(n)$
2	3	3
3	8	13
4	21	116

### 1-3. Одноканальные повторные сети из настраиваемых элементов

Рассмотрим одномерные сети, в которых каждая переменная  $x_j$  ( $1 \leq j \leq n$ ) может подаваться на внешние входы нескольких элементов (очевидно, что подача констант с точки зрения увеличения функциональных возможностей смысла не имеет).

Класс функций, реализуемых такими сетями (будем называть их повторными), несколько шире, чем в случае бесповторных сетей. Например, функция  $f(x_1, x_2, x_3) = x_1\bar{x}_3 \vee \bar{x}_1\bar{x}_2x_3$ , не реализуемая бесповторной сетью, представима повторной в виде

$$f(x_1, x_2, x_3) = \bar{x}_3 \oplus \bar{x}_1 (\bar{x}_2 \vee \bar{x}_3).$$

С другой стороны, не все функции могут быть реализованы повторными сетями, в том числе функция

$$\text{maj}(x_1, x_2, x_3) = x_1x_2 \vee x_1x_3 \vee x_2x_3 \text{ (см. пример 1-4).}$$

Как и в предыдущем случае, ответ на вопрос о реализуемости заданной функции дает применение алгоритма синтеза сети.

Заметим, что многократная подача переменных на внешние входы элементов не всегда приводит к расширению логических возможностей. Сформулируем следующие утверждения.

1. Если в сети на  $j$ -й элемент (счет элементов ведется справа налево, начиная с выходного элемента), являющийся элементом типа  $\vee$  или  $\wedge$ , подается переменная  $x_i$  и если же переменная  $x_i$  подается на один из последующих элементов, то по крайней мере один элемент сети является избыточным.

Действительно, пусть  $j$ -й элемент является элементом типа  $\wedge$ . Тогда на выходе  $j$ -го элемента реализуется функция

$$f = x_i f_1(x_1, \dots, x_k).$$

Так как по предположению  $f_1(x_1, \dots, x_k)$  зависит и от  $x_i$ , то

$$f_1(x_1, \dots, x_k) = x_i f_1(x_i=1) \vee \bar{x}_i f_1(x_i=0).$$

Тогда

$$f = x_i [x_i f_1(x_i=1) \vee \bar{x}_i f_1(x_i=0)] = x_i f_1(x_i=1),$$

т. е. функция  $f(x_1, \dots, x_k)$  может не зависеть от переменной  $x_i$ . Если  $j$ -й элемент является элементом типа  $\vee$ , то доказательство проводится аналогично.

Таким образом, в безызбыточной повторной сети переменные должны подаваться неоднократно лишь на элементы типа  $\oplus$ .

Легко доказать также следующее утверждение.

2. В неизбыточной реализации одна и та же переменная не может подаваться на рядом стоящие элементы, а также на элементы двух соседних итеративных участков.

**Синтез.** Если заданная функция не может быть реализована в виде бесповторной сети, что выясняется на некотором шаге применения соответствующего алгоритма, то начиная с этого шага, следует искать повторную реализацию остаточной функции, пользуясь *разложением Рида*:

$$f(x_1, \dots, x_n) = f(x_i=0) \oplus (x_i(f(x_i=0) \oplus f(x_i=1))),$$

или

$$f(x_1, \dots, x_n) = f(x_i=1) \oplus (\bar{x}_i(f(x_i=0) \oplus f(x_i=1))),$$

справедливым для любой логической функции.

Из вида разложения следует, что переменную  $x_i$  можно исключить, если одна из остаточных функций  $f(x_i=0)$  или  $f(x_i=1)$  представима в виде суммы по модулю 2 своих переменных. Отсюда, в частности, следует, что если какие-либо переменные подаются повторно, то элементы типов  $\vee$  или  $\wedge$ , управляемые этими переменными, должны быть расположены дальше от выхода цепи, чем элементы типа  $\oplus$ , управляемые теми же переменными.

Таким образом, алгоритм синтеза повторной сети по сравнению с алгоритмом синтеза бесповторной сети включает в себя дополнительный шаг.

3. Проверяется, представима ли одна из каждой пары остаточных функций в виде суммы по модулю 2 переменных (или их инверсий), от которых она зависит. Если не представима, то заданная функция не реализуема повторной сетью. Если представима, то запоминаем реализацию этого участка схемы, образуем функцию

$$x_i(f(x_i=0) \oplus f(x_i=1))$$

и переходим к выполнению п. 1 алгоритма применительно к этой функции.

Может оказаться, что при разложении по Риду

$$f = g_1 \oplus x_i^{\sigma_i} g_2 \text{ и } f = g_3 \oplus x_j^{\sigma_j} g_4,$$

причем  $g_1$  и  $g_3$  являются суммой по модулю 2 своих переменных. Тогда из факта реализуемости  $g_2$  следует реализуемость  $g_4$ , и наоборот. Доказательство этого факта читатель может найти в [Л. 53].

Пример 1-4. Построить сеть, реализующую функцию

$$\begin{aligned} f(x_1, \dots, x_4) = & \bar{x}_1 \bar{x}_2 x_3 \bar{x}_4 \vee \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 \vee x_1 x_2 \bar{x}_3 x_4 \vee \\ & \vee x_1 \bar{x}_2 x_3 x_4 \vee \bar{x}_1 x_2 x_3 x_4. \end{aligned}$$

Легко убедиться в том, что, пользуясь алгоритмом синтеза бесповторных сетей, нельзя исключить ни одной из переменных. С другой стороны, используя разложение Рида, убеждаемся в том, что

п. 4 алгоритма удовлетворяет лишь исключение переменной  $x_4$ , поскольку

$$f(x_4 = 1) = x_1 x_2 x_3 \vee x_1 x_2 \bar{x}_3 \vee x_1 \bar{x}_2 x_3 \vee \bar{x}_1 x_2 x_3 = x_1 \oplus x_2 \oplus x_3,$$

$$f(x_4 = 0) = x_1 x_2 x_3.$$

Далее,

$$\bar{x}_4 (f(\bar{x}_4 = 1) \oplus f(x_4 = 0)) = x_4 (x_1 \bar{x}_2 \vee \bar{x}_1 x_3 \vee \bar{x}_2 x_3 \vee x_1 x_2 \bar{x}_3) = \\ = \bar{x}_4 g_1(x_1, x_2, x_3).$$

Итак, за итеративным участком из трех элементов типа  $\oplus$ , на которые подаются внешние переменные  $x_1, x_2, x_3$ , следует элемент типа  $\wedge$  с внешним входом  $\bar{x}_4$  и боковым входом  $g_1(x_1, x_2, x_3)$ .

Исключая далее, например, переменную  $x_3$ , получаем:

$$g_1(x_3 = 1) = \bar{x}_1 \vee \bar{x}_2,$$

$$g_1(x_3 = 0) = \bar{x}_1 \bar{x}_2 \vee x_1 x_2 = \bar{x}_1 \oplus x_2;$$

$$x_3 (g_1(x_3 = 1) \oplus g_1(x_3 = 0)) = x_3 (x_1 \vee x_2).$$

Искомая сеть представлена на рис. 1-4.

Пример 1-5. Мажоритарная функция (см. пример 1-3), очевидно, не реализуется одномерной повторной сетью, так как ни одна из ее остаточных функций не представима в виде суммы своих аргументов по модулю 2.

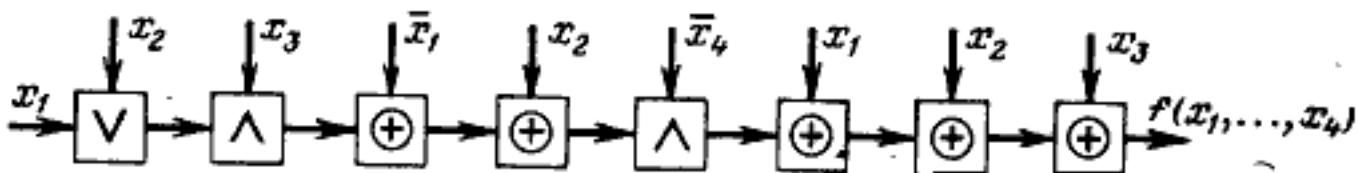


Рис. 1-4. Пример реализации логической функции повторной сетью.

Как и в случае бесповторных сетей, применением правил де Моргана и перестановкой переменных в пределах итеративных участков можно получить семейство повторных сетей, реализующих заданную функцию.

На основании сказанного можно заключить, что в случае повторных сетей существует проблема минимизации. Некоторые правила минимизации могут быть предложены на основании условий безызбыточности повторных сетей. Более подробно этот вопрос никем не рассматривался.

Возникает вопрос, какова наименьшая длина  $C(n)$  повторной сети, реализующей «наихудшую из реализуемых» функцию  $n$  переменных? Ответ на него получен в [Л. 53]:

$$C(n) = n^2 + n - 4)/2.$$

Там же предложен конструктивный способ построения такой сети.

Точная оценка числа функций, реализуемых повторными сетями, не известна; верхняя [Л. 53] приводится ниже:

$$T(n) < 3n! \cdot 2^n \cdot 2^{(n+4)(n-1)/2}.$$

Отношение числа реализуемых функций  $n$  переменных к общему числу функций  $n$  переменных с ростом  $n$  стремится к 0. Однако их число все же больше, чем бесповторных. Для малых  $n$  число представителей функции, реализуемых бесповторными  $M(n)$  — и повторными —  $T(n)$  — сетями, представлено в таблице на стр. 25.

Попутно заметим, что любая троичная логическая функция  $n$  переменных (такая функция и ее аргументы принимают три значения — например, 0, 1, 2 или  $-1, 0, +1$ ) может быть реализована повторной односторонней одноканальной сетью длины  $L(n)$ , где  $L(n) = 3(L(n-1) + 1)$  и  $L(1) = 1$ , каждый элемент которой реализует произвольную троичную функцию двух переменных [Л. 63].

#### 1-4. Двухканальные сети

Ранее рассматривались одномерные сети из элементов, реализующих логические функции двух переменных, т. е. элементов с одним информационным внешним и одним боковым входами.

Возникает вопрос: как возрастают логические возможности таких сетей при увеличении числа только внешних информационных входов на элемент? При такой постановке каждый элемент должен реализовать любую одну функцию от  $m$  переменных, где  $m > 2$  (рис. 1-5). Однако оказывается, что такое усложнение элемента не приводит к существенному увеличению доли реализуемых функций, что показано в [Л. 33].

Другим путем увеличения логических возможностей одномерных сетей из комбинационных элементов является использование большего числа боковых каналов передачи информации.

Потребуем, чтобы элемент сети мог реализовать любую пару логических функций трех переменных (рис. 1-6):

$$f_1 = f_1(x, y_1, y_2);$$

$$f_2 = f_2(x, y_1, y_2);$$

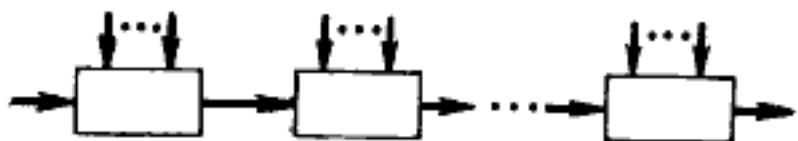


Рис. 1-5. Одноканальная сеть из многовходовых комбинационных элементов.

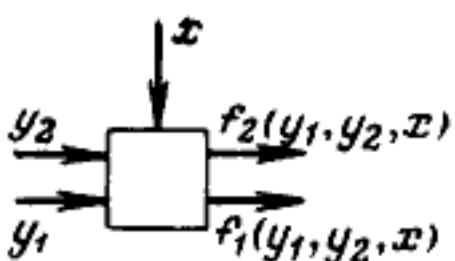


Рис. 1-6. Элемент двухканальной сети.

(внешний выход элемента отсутствует, в качестве него может быть использован любой боковой выход). Будем рассматривать возможности одномерной сети из таких элементов (рис. 1-7), допуская повторную подачу внешних переменных.

**Логические возможности.** Легко показать, что любая логическая функция четырех переменных может быть реализована сетью из двух таких элементов, что видно из рис. 1-8.

Оказывается также, что вообще любая функция любого числа переменных может быть реализована сетями

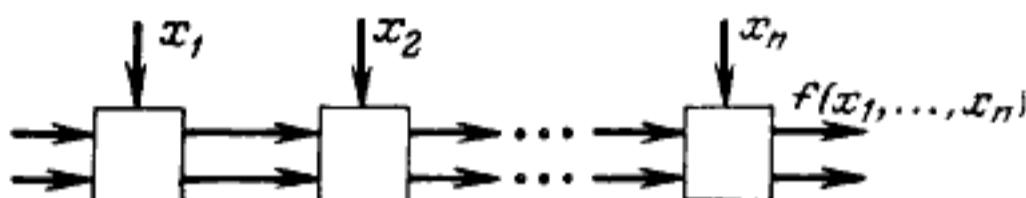


Рис. 1-7. Двухканальная сеть из комбинационных элементов.

из элементов типа рис. 1-6, если от элемента потребовать, чтобы на боковых выходах  $f_1$  и  $f_2$  можно было реализовать любую из следующих функций:

$$f_1 \in \{xy_1, \bar{x}y_1, 1\},$$

$$f_2 \in \{xy_1 \vee y_2, \bar{x}y_1 \vee y_2, y_2\}.$$

Тогда можно формировать последовательно импликанты (имплиценты) нормальной формы заданной булевой функции на боковом выходе  $f_1$  (нижнем канале), а верхний канал передачи информации использовать для последовательного суммирования (умножения) сформированных импликантов (имплицентов) и хранения («накопления») текущего значения дизъюнктивной (конъюнктивной) нормальной формы.

Например, функция  $f(x_1, \dots, x_5) = x_1x_2 \vee x_3x_4x_5$  реализуется схемой, показанной на рис. 1-9. Ясно, что макси-

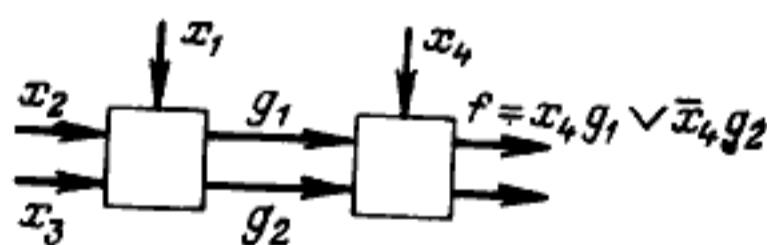


Рис. 1-8. Реализация произвольной функции четырех переменных.  $g_1$  и  $g_2$  — произвольные функции трех переменных ( $x_1$ ,  $x_2$  и  $x_3$ ).

мальное число элементов в такой сети, реализующей функцию или ее отрицание, не должно превышать  $n2^{n-1}$  ( $n$ —максимальное число букв в импликанте,  $2^{n-1}$ —максимальное число импликантов). (Если число конституент единицы заданной функции превышает  $2^{n-1}$ , то следует реализовать дополнение ее, в котором будет не

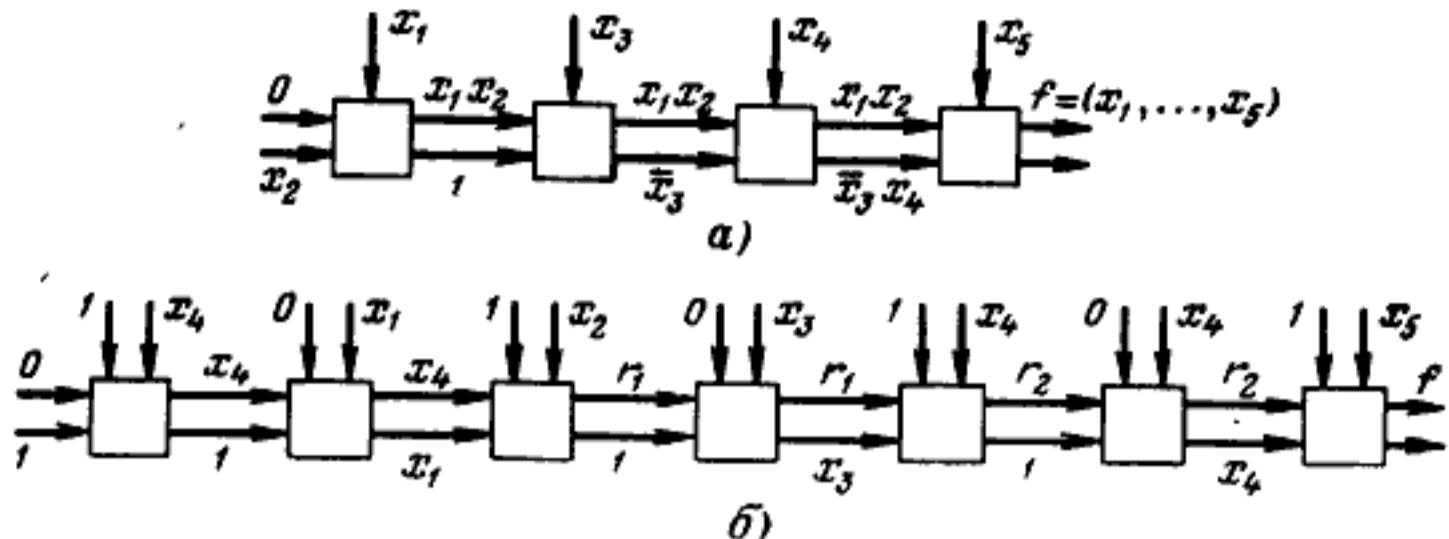


Рис. 1-9. Примеры реализации логических функций двухканальными сетями.

*a*—на элементах, реализующих произвольную пару функций трех переменных;  
*б*—на элементах с одним настречным входом ( $r_1 = x_4 \oplus x_1x_2$ ;  $r_2 = x_4 \oplus x_1x_2 \oplus x_3x_4$ ).

более  $2^{n-1}$  конституент единицы.) Разложение по полиному Жегалкина приводит в общем случае к тому же числу элементов.

**Синтез.** Лучшие результаты могут быть получены при использовании метода последовательного исключения переменных в разложении Рида:

$$f(x_1, \dots, x_n) = g_1(x_1, \dots, x_{n-1}) \oplus x_n g_2(x_1, \dots, x_{n-1}).$$

В этом случае можно отдельно реализовать функции  $g_1$  и  $g_2 x_n$ , что дает некоторое уменьшение числа элементов. Например, любая функция пяти переменных может быть разложена следующим образом:

$$\begin{aligned} f(x_1, \dots, x_5) &= g_1(x_1, \dots, x_4) \oplus x_5 g_2(x_1, \dots, x_4) = \\ &= (h_0(x_1, x_2) \oplus x_3 h_1(x_1, x_2) \oplus x_4 h_2(x_2, x_3) \oplus \\ &\quad \oplus x_1 x_4 h_3(x_2, x_3)) \oplus x_5 g_2(x_1, \dots, x_4), \end{aligned}$$

где  $g_1$ ,  $g_2$ — некоторые функции четырех переменных, а  $h_0$ ,  $h_1$ ,  $h_2$ ,  $h_3$ — некоторые функции двух переменных, причем функция  $g_2$  может быть реализована на двух элементах, как на рис. 1-8. «Накопление» в верхнем ка-

нале ведется по модулю 2. Соответствующая сеть представлена на рис. 1-10. Можно показать, что эта каноническая реализация является оптимальной по числу элементов.

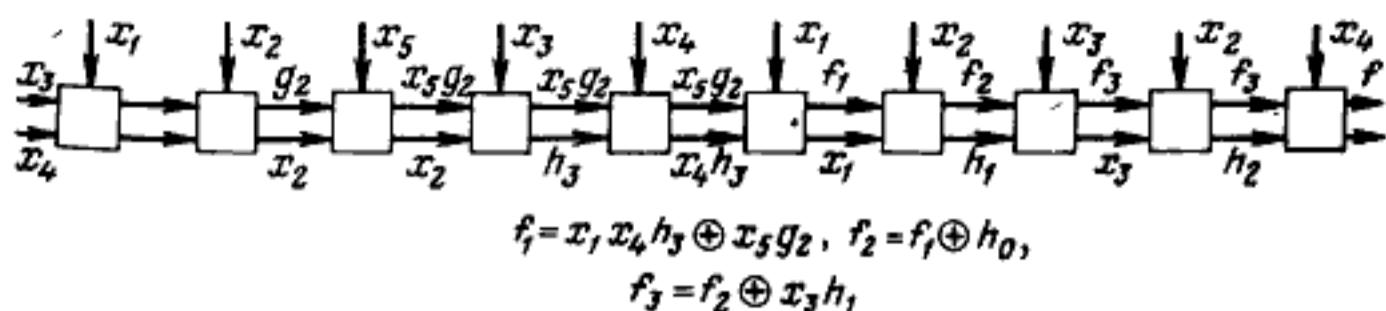


Рис. 1-10. Реализация произвольной функции пяти переменных с использованием разложения Рида.

Можно получить рекуррентную оценку оптимального числа элементов канонической сети, реализующей произвольную функцию  $n$  переменных<sup>1</sup>:

$$N(n) \leq N(n-1) + (n-1)2^{n-4}.$$

Верхняя оценка числа элементов при реализации произвольной функции  $n$  переменных по Шеннону и Риду (для  $n \leq 10$ ) приведена ниже:

$n$	$n2^{n-1}$	$N(n)$
3	12	1
4	32	2
5	80	10
6	192	30
7	448	78
8	1 024	190
9	2 304	446
10	5 120	1 022

Вообще говоря, сложность сети следовало бы оценивать не только числом ее элементов, но также надо учитывать сложность отдельного элемента. Заметим, что требование реализации любой пары функций трех переменных предъявляется только к первому слева элементу

<sup>1</sup> Очевидно, что для больших  $n$   $N(n) < n2^{n-3}$  и, более того,  $N(n) = n2^{n-3}(1+o(1))$ . (Прим. ред.)

сети. Если ограничиться только пятью типами элементов (пятью парами функций):

1-й тип:  $f_1 = x, f_2 = y_2 \oplus xy_1;$

2-й тип:  $f_1 = y_1, f_2 = xy_2;$

3-й тип:  $f_1 = h(x, y_1), f_2 = y_2;$

4-й тип:  $f_1 = xy_1, f_2 = y_2;$

5-й тип:  $f_1 = h(x, y_1), f_2 = y_2 \oplus h(x, y_2)$

( $h$  — произвольная булева функция двух переменных), то произвольная функция может быть также реализована на элементах этих типов, причем число элементов сети увеличивается на три.

Минимальное число типов элементов может быть достигнуто, если использовать разложение функции по полиному Жегалкина. Тогда с помощью только двух пар функций можно реализовать произвольную логическую функцию:

$$f_1 = y_1, f_2 = xy_2;$$

$$f_1 = y_1 \oplus xy_2, f_2 = 1.$$

Настройку элемента можно осуществлять одним настроечным входом  $a$ . Тогда элемент реализует следующую пару функций:

$$f_1 = y_1 \bar{a} \vee (y_1 \oplus xy_2) a,$$

$$f_2 = a \vee xy_2.$$

Если  $a=0$ , то в нижнем канале происходит формирование члена полинома, а в верхнем сохраняется текущее значение полинома; если же  $a=1$ , то в верхнем канале к текущему значению прибавляется по модулю 2 значение сформированного члена полинома. (Под текущим значением полинома Жегалкина будем здесь и далее понимать то значение его, которое получается сложением по модулю 2 очередной образованной конъюнкции с «накопленным» значением полинома.)

Например, реализация функции

$$f(x_1, \dots, x_5) = x_4 \oplus x_1x_2 \oplus x_3x_4 \oplus x_4x_5$$

представлена на рис. 1-9,б.

На основании результатов, изложенных ранее, можно сделать вывод, что для реализации произвольной логической функции одномерной сетью достаточно иметь

элемент с шестью выводами [Л. 15]: два боковых входа, два боковых выхода, один информационный вход и один настроочный.

В [Л. 61] развивается теоретико-групповой подход к синтезу двухканальных односторонних сетей. Предложены процедуры реализации произвольной пары функций 4—5 переменных, основанные на некоторых свойствах симметрии и функциональной разделимости.

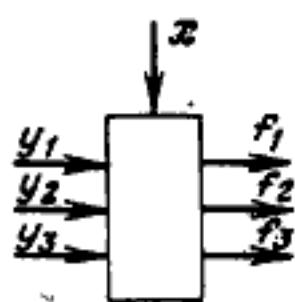
### 1-5. Трехканальные сети

Ранее было показано, что универсальностью (т. е. способностью реализовать любую логическую функцию) обладают лишь повторные сети с числом боковых каналов более одного, с общим числом внешних входов — не менее двух. Возникает соображение,

нельзя ли добиться универсальности за счет увеличения числа боковых каналов при сохранении одного внешнего входа на каждый элемент. Что касается бесповторных сетей, то доказано, что они не могут быть универсальными при любом фиксированном числе каналов. Это доказательство (построением опровергающего примера) дано в [Л. 27] и приведено на стр. 55. Следовательно, универсальными могут

Рис. 1-11. Элемент трехканальной сети.

оказаться лишь повторные сети. Действительно, существует универсальная однородная односторонняя сеть из элементов с тремя боковыми каналами и одним двузначным внешним входом, реализующая произвольную функцию  $p$  переменных. Настройка сети производится подачей констант на нечетные элементы. Для построения сети используем представление функции в виде полинома Жегалкина. Рассмотрим схему с тремя двоичными боковыми каналами. По первому каналу передается текущее значение полинома, во втором канале происходит формирование очередной конъюнкции. По третьему каналу передается сигнал четности (нечетности) позиции элементов. Полином Жегалкина представляет собой формулу, в которой на нечетных позициях стоят переменные, а на четных — операции. Элемент схемы работает следующим образом: на четных позициях он умножает значение поданной на него переменной на значение сформированной части конъюнкции и передает на следующий элемент новое значение конъюнкции и преды-



дущее текущее значение полинома. На нечетных позициях текущие значения полинома и значения сформированной части конъюнкции передаются без изменений, если входной сигнал равен 0 (операция умножения в формуле). Если внешний вход равен 1, то значение

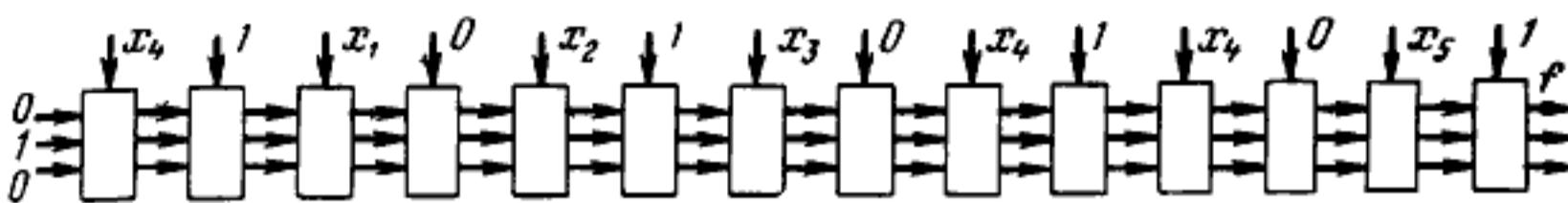


Рис. 1-12. Пример реализации логической функции трехканальной сетью.

сформированной части конъюнкции прибавляется по модулю 2 к текущему значению полинома, а передаваемое значение конъюнкции устанавливается равным 1.

Легко показать, что элемент универсальной трехканальной сети (рис. 1-11) реализует следующую систему трех функций:

$$f_1 = y_1 \oplus y_2 y_3 x;$$

$$f_2 = y_2 \# y_3 \# x = y_2 y_3 \vee y_2 x \vee y_3 x;$$

$$f_3 = y_3 \oplus 1.$$

Число элементов, требуемых для реализации любой функции  $n$  переменных, в этом случае равно  $n2^n$ .

Реализация функции  $x_4 \oplus x_1 x_2 \oplus x_3 x_4 \oplus x_4 x_5$  представлена на рис. 1-12.

Аналогичным образом может быть построена схема, основанная на представлении функции дизъюнктивными нормальными формами. Однако реализация получается сложнее.

### **1-6. Об эквивалентности произвольных односторонних сетей и сетей из элементов с одним внешним входом**

Настройка одномерной односторонней сети из комбинационных элементов может быть выполнена двумя способами. Первый состоит в том, что на каждом элементе выделяются специальные настроечные входы, и настройка производится за счет подачи сигналов на эти входы, второй заключается в подаче настроечных сигналов на

некоторые элементы вместо переменных. При этом втором способе настройки входы, на которые подаются настроочные сигналы, опознаются как таковые благодаря наличию дополнительных боковых каналов, позволяющих «отсчитывать» те позиции сети, на которых стоят элементы, служащие лишь целям настройки. Такая ситуация имела место в предыдущем параграфе, где третий канал использовался для отсчета четной-нечетной позиции элементов сети.

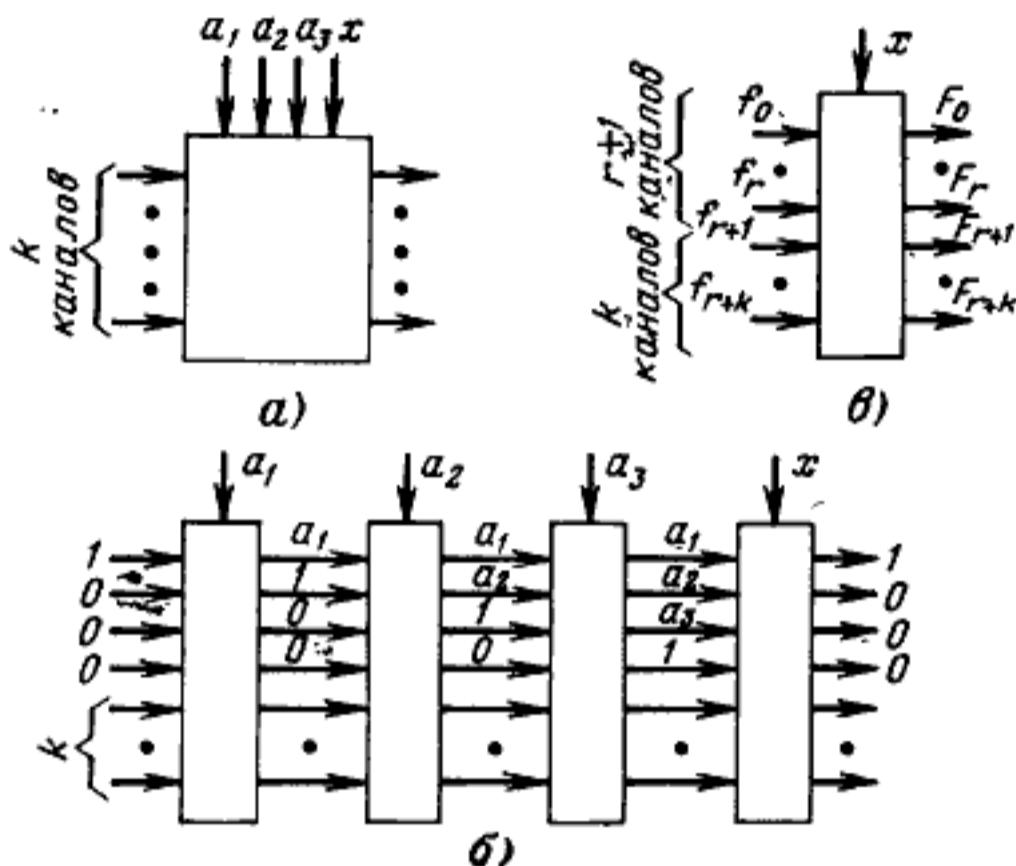


Рис. 1-13. Элемент многоканальной сети с индивидуальной настройкой (a); участок сети, эквивалентный этому элементу (б); элемент участка сети, имеющий один внешний вход,  $r+1$  канал настройки и  $k$  боковых каналов (в).

Можно показать, что сети с одним типом настройки можно сопоставить эквивалентную сеть с другим видом настройки. Для этого элементу сети (рис. 1-13, а) с одним информационным входом,  $r$  настроечными входами и  $k$  боковыми каналами поставим в соответствие участок однородной сети из  $r+1$  элементов, каждый из которых имеет один внешний вход и  $k+r+\lceil \log_2 r \rceil$  боковых входов (выходов). Тогда  $\lceil \log_2 r \rceil$  каналов достаточно для того, чтобы опознать каждый элемент участка;  $r$  каналов служат для передачи значений настроечных сигналов на  $(r+1)$ -й элемент участка. Последний,  $(r+1)$ -й элемент участка имеет, таким образом, один

внешний информационный вход,  $r$  боковых входов, определяющих соответствующую настройку элемента,  $\lceil \log_2 r \rceil$  входов, позволяющих отметить, что этот элемент является «рабочим», а не настроенным. «Рабочий» элемент реализует ту же функцию  $k+1$  переменных, что и исходный настраиваемый элемент.

Покажем, что приведенную выше оценку числа боковых каналов можно уменьшить по крайней мере до  $k+r+1$ . Рассмотрим рис. 1-13,*a*, *b*, где элементу с тремя настроечными входами и одним информационным поставлен в соответствие участок из четырех элементов. На рис. 1-13,*b* изображены только  $r+1=4$  каналов для передачи настроечных сигналов и идентификации каждого элемента участка. Заметим, что в  $i$ -м элементе участка под передачу настроечных сигналов занято  $i$  выходных каналов. Из рисунка видно, что граничный сигнал, равный единице, на верхнем канале передается от элемента к элементу со сдвигом на один канал, указывая номер элемента участка. Когда единица окажется на  $(r+1)$ -м канале, это означает, что  $(r+1)$ -й элемент является «рабочим».

Элемент участка (рис. 1-13,*b*) реализует следующую систему функций:

$$\begin{aligned} F_0 &= (x \vee f_1 \vee f_2 \vee \dots \vee f_{r-1}) f_0 \vee f_r; \\ F_1 &= ((x \vee f_2 \vee f_3 \vee \dots \vee f_{r-1}) f_1 \vee f_0 \bar{f}_1 \bar{f}_2 \dots \bar{f}_{r-1}) \bar{f}_r; \\ F_2 &= ((x \vee f_3 \vee f_4 \vee \dots \vee f_{r-1}) f_2 \vee f_1 \bar{f}_2 \bar{f}_3 \dots \bar{f}_{r-1}) \bar{f}_r; \\ &\dots \\ F_{r-1} &= (f_{r-1} x \vee f_{r-2} \bar{f}_{r-1}) \bar{f}_r, \\ F_r &= f_{r-1} \bar{f}_r; \\ F_{r+1} &= f_r F_{r+1}(f_0, \dots, f_{r-1}, f_{r+1}, \dots, f_{r+k}) \vee \bar{f}_r f_{r+1}; \\ &\dots \\ F_{r+k} &= f_r F_{r+k}(f_0, \dots, f_{r-1}, f_{r+1}, \dots, f_{r+k}) \vee \bar{f}_r f_{r+k}. \end{aligned}$$

Аналогичным способом может быть произведено разделение информационных входов, если их число более одного.

Найденная оценка числа боковых каналов не является минимальной и во многих случаях допускает улучшение.

## **1-7. Системы из комбинационных элементов**

В § 1-1 были рассмотрены все возможные одноканальные системы. При большем числе боковых каналов перечислить возникающие задачи практически невозможно. Поэтому мы вынуждены ограничиться рассмотрением некоторого списка задач, которые в определенной мере характеризуют поведение систем и их функциональные возможности.

Несколько слов о языке, который нами будет использован для описания поведения систем. В начале гл. 1 мы установили, что одномерные системы из комбинационных элементов и конечные автоматы эквивалентны в том смысле, что они обладают одинаковым поведением. Каждому конечному автомatu с входной последовательностью из  $n$  символов, подаваемой на вход автомата последовательно, такт за тактом, можно поставить в соответствие сеть системы, состоящую из  $n$  элементов, на которую параллельно подается код, совпадающий с входной последовательностью автомата, по одному символу на каждый элемент.

Как известно, поведение конечного автомата может быть задано на языке регулярных выражений. Если задание на синтез удается записать на этом языке, то реализующий его автомат всегда можно построить с помощью известных методов (В. М. Глушков, Синтез цифровых автоматов. М., ФМГ, 1962, М. А. Айзerman и др. Логика. Автоматы. Алгоритмы. М., ФМГ, 1963). «Построить автомат» — в данном случае означает построить граф его переходов или таблицу переходов и выходов. Из эквивалентности автомата и системы из комбинационных элементов следует, что полученная таблица переходов и выходов может в модифицированном виде служить описанием элемента системы.

Для объяснения сказанного приведем простой пример — задачу.

**Задача 1.** (М. Минский, Вычисления и автоматы. М., «Мир», 1971). Построить систему, внешний выход каждого элемента которой равен 1, когда число единиц в наборе внешних входов слева от него (включая сам элемент) делится на 3, и равен 0 в противном случае.

Регулярное выражение, описывающее данное событие, может быть записано в следующем виде:

$$(0 * 10 * 10 * 1) * 0 * .$$

Диаграмма переходов автомата, реализующего данное выражение, приведена на рис. 1-14. В скобках отмечено значение выхода автомата, равного 1. Таблица переходов и выходов автомата, реализующего регулярное выражение

$$(0 * 10 * 10 * 1) * 0,$$

	$x = 0$	$x = 1$
$S_0$	$S_0, 1$	$S_1, 0$
$S_1$	$S_1, 0$	$S_2, 0$
$S_2$	$S_2, 0$	$S_0, 1$

одновременно описывает поведение элемента системы, обладающей заданным поведением.

Однако здесь символы  $S_i (0 \leq i \leq 2)$  обозначают боковые сигналы, а не состояния автомата. Границным сигналом, соответствующим начальному состоянию автомата, является сигнал  $S_0$ .

В данном варианте решения задачи имеется одна особенность — составление регулярного выражения по словесному описанию задачи. Этот этап может быть в некоторых случаях выполнен лишь специалистом, хорошо знакомым с языком регулярных выражений, на эвристическом уровне, потому что не существует формального метода перевода словесного задания на язык регулярных выражений.

Развиваемый в этой работе подход основывается на том, что этап составления регулярных выражений исключается. По словесному заданию непосредственно строятся таблицы (будем называть их *модифицированными*), описывающие поведение элемента системы, подобно тому, как это часто делается в теории автоматов. Эти таблицы будут являться частным случаем широко известных таблиц переходов и выходов.

В последующих главах модифицированными таблицами описывается и поведение систем из элементов с памятью. Используемый язык модифицированных таблиц переходов обладает тем же недостатком, что и язык ре-

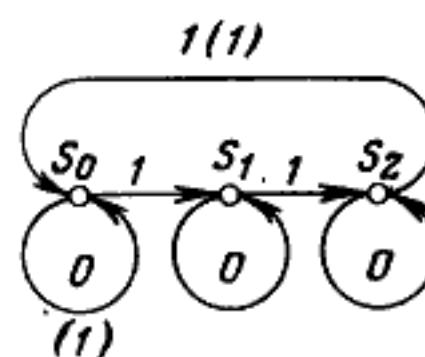


Рис. 1-14. Диаграмма переходов автомата, рассматриваемого в задаче 1.

гулярных выражений: этап составления их не удается формализовать. Тем не менее, нам кажется, что этот язык обладает по крайней мере тем преимуществом, что его можно применять как для синтеза систем из комбинационных элементов, так и для систем из элементов с памятью. Представляется, что язык модифицированных таблиц переходов удобен тем, что он ориентирован больше на описание поведения систем, чем на описание событий, представляемых этими системами. Отмеченные особенности языка определили его применение для решения класса рассматриваемых задач.

Следует объяснить, чем, с нашей точки зрения, интересен подход, связанный с постановками задач о системах, а не об отдельных сетях.

Во-первых, при реализации логических функций сложность пирамидальных схем из функциональных элементов растет, как известно, при  $n \rightarrow \infty$  как  $C(2^n/n)$ , т. е. как показательная функция. Требование однородности ведет к еще большему расходу элементов, при  $n \rightarrow \infty$  сложность оценивается уже как  $C2^n$ , а при некоторых асимптотически неоптимальных конструкциях — как  $Cn2^n$  (см. гл. 4). Сложность же реализации сетей, принадлежащих одной системе, линейно зависит от числа ее элементов и равна  $Cn$  (здесь  $C$  — сложность элемента системы). Естественно, что с такой сложностью могут быть реализованы только некоторые классы функций, причем каждая система представляет вполне определенный класс. Приведенные ниже решения задач свидетельствуют о том, что множество этих реализуемых с линейной сложностью классов функций не так узко, как это кажется на первый взгляд.

Во-вторых, рассматриваемые задачи интересны сточки зрения поведенческих аспектов теории автоматов. Функционирование системы задается локальным алгоритмом работы отдельного элемента и описанием его связей с соседями. Оказывается, что системы с достаточно простыми элементами и межэлементными связями обладают достаточно сложным поведением, что видно хотя бы из ниже следующих задач на стр. 65, 92.

Сказанное может быть отнесено в равной степени и к последующим главам.

Теперь вернемся к системам из комбинационных элементов и попытаемся в общих чертах описать принципы синтеза однородных одномерных регулярных систем.

Процедуру синтеза можно разделить на четыре шага:

- 1) составление логического описания в соответствии с требованиями задачи;
- 2) описание структуры индивидуального элемента;
- 3) перевод упрощенного логического описания на язык булевых функций;
- 4) реализация этих функций на реальных схемных элементах.

Поскольку третий и четвертый шаги логического проектирования достаточно подробно рассматривались в литературе и не имеют особой специфики, здесь мы ограничимся рассмотрением вопросов, связанных с первыми двумя шагами.

Выполнение первого этапа синтеза систем заключается в разбиении множества наборов внешних входов элементов, находящихся слева от рассматриваемого, на подмножества, соответствующие различным боковым сигналам, в соответствии с условиями задачи. При этом в один класс объединяются наборы значений внешних входов, неразличимые с точки зрения их воздействия на данный элемент. Такое разбиение не всегда является единственным, а следовательно, не всегда приводит к минимальному числу боковых сигналов. На первом этапе синтеза вопрос минимизации по числу классов не рассматривается.

На втором этапе выясняется, как наборы, принадлежащие каждому классу, переходят в наборы другого класса под воздействием внешнего входа данного элемента.

Возвращаясь к задаче 1, решим ее, используя язык модифицированных таблиц переходов.

Все наборы значений внешних входов слева от данного элемента можно разбить на 3 класса:

- 1) число единиц в наборе делится на 3;
- 2) остаток от деления числа единиц в наборе на 3 равен 1;
- 3) остаток от деления числа единиц в наборе на 3 равен 2.

Сопоставим этим классам три боковых сигнала —  $S_0$ ,  $S_1$  и  $S_2$  соответственно. Легко видеть, что искомый элемент системы преобразует эти боковые сигналы в соответствии с таблицей переходов и выходов (стр. 39).

**Задача 2 [Л. 27].** Построить систему, внешний выход каждого элемента которой равен единице, когда на-

бор внешних входов слева от него содержит серию из трех или более единиц, и равен нулю в противном случае.

Значение трех внешних переменных, подаваемых на три элемента слева от данного, можно разделить на четыре подмножества:

$x_{t-3}$	$x_{t-2}$	$x_{t-1}$	$S_j$
0	0	0	$S_0$
1	0	0	$S_0$
0	1	0	$S_0$
1	1	0	$S_0$
0	0	1	$S_1$
-1	0	1	$S_1$
0	1	1	$S_2$
1	1	1	$S_3$

Составим таблицу переходов подмножеств  $S_i$  под воздействием внешнего входа данного элемента и укажем также значение его внешнего выхода.

	$x = 0$	$x = 1$
$S_0$	$S_0,0$	$S_1,0$
$S_1$	$S_0,0$	$S_2,0$
$S_2$	$S_0,0$	$S_3,0$
$S_3$	$S_3,1$	$S_3,1$

Такая таблица полностью задает поведение элемента системы.

Внешний выход элемента принимает значение 1 тогда и только тогда, когда он получает слева сигнал  $S_3$ .

**Задача 3 [Л. 15].** Построить одномерную одностороннюю систему, реализующую симметрическую логическую функцию с заданными рабочими числами. Некоторые частные случаи этой задачи можно найти в [Л. 21].

Как известно, любая симметрическая функция полностью задается рабочими числами, т. е. указанием числа единиц в наборах, на которых она принимает значение, равное 1. Например, симметрическая функция

$$f = \overline{x_1}x_2\overline{x_3} \vee x_1\overline{x_2}\overline{x_3} \vee \overline{x_1}x_2x_3 \vee x_1\overline{x_2}x_3$$

может быть задана рабочими числами 0 и 2. Рассмотрим сначала реализацию симметрической функции, заданной одним рабочим числом  $r$ . Пусть имеется односторонняя сеть из комбинационных элементов с  $r+2$  боковыми сигналами  $S_0, S_1, \dots, S_r, S_{r+1}$  и граничным условием  $S_0$ . Установим следующее правило перехода:

$$S_j = \begin{cases} S_{j+1}, & \text{если } x=1 \\ S_j, & \text{если } x=0 \end{cases}, \quad 0 \leq j \leq r;$$

если значение бокового выхода равно  $S_{r+1}$ , то значение его остается без изменения вне зависимости от значения внешнего входа  $x$ .

Внешний выход каждого элемента равен 1, если боковой выход этого элемента равен  $S_r$ , и равен 0 в противном случае.

Теперь предложим конструкцию системы для реализации функции, заданной несколькими рабочими числами  $r_1, r_2, \dots, r_k$ . Выберем из них некоторое число  $r_i = \max(r_1, r_2, \dots, r_k)$ . Число боковых сигналов для реализации такой функции будет равно  $r_i + 2$ , правила перехода — такие же, как и указанные ранее, и  $r=r_i$ , а внешний выход каждого элемента равен 1, если боковой выход этого элемента равен  $S_{r_1}$ , или  $S_{r_2}, \dots$ , или  $S_{r_k}$ .

**Задача 4 [Л. 15].** Построить систему, сигнал на внешнем выходе каждого элемента которой равен 1, когда число, являющееся эквивалентом двоичного набора внешних входов, начиная с крайнего элемента слева до данного элемента включительно, делится на 3, и равен 0 в противном случае.

Как может решаться такая задача «вручную»? Известно, что признаком делимости числа в системе счисления с основанием  $m$  на число  $m-1$  является делимость суммы цифр этого числа на  $m-1$ . Поэтому основанием системы счисления при делении на 3 должно быть выбрано число 4, т. е. два разряда двоичного кода. Разобьем двоичный набор внешних входов на участки слева направо так, чтобы в каждом участке содержалось точно по два разряда. Деление на 3 производится путем вычитания тройки из текущей суммы цифр, если эта сумма не менее 3. Это может быть выполнено следующим образом. Берется число, представленное первым слева участком из двух разрядов, и сравнивается с числом 3. Если число, представляемое двумя двоичными

разрядами, равно 11, то из него вычитается тройка, и, следовательно, текущая сумма будет равна 0. Если это число отлично от 11, то вычитания не происходит, и это число складывается с числом, представляемым следующей парой разрядов. Эта новая сумма сравнивается с числом 3, и если она не меньше 3, из нее вычитается тройка, а остаток складывается со следующей парой разрядов и т. д.

Задача может быть решена с шестью боковыми сигналами:  $S_0^1, S_0^2, S_1^1, S_1^2, S_2^1, S_2^2$ , нижний индекс в которых есть значение текущей суммы на выходе элемента, а верхний индекс равен 1 и 2 в соответствии с тем, является ли этот элемент сети нечетным или четным. Границное условие представляется сигналом  $S_0^1$ . Исходная (a) и упрощенная (б) таблицы переходов и значений внешнего выхода элемента имеют вид:

	$x = 0$	$x = 1$
$S_0^1$	$S_0^2, 1$	$S_1^2, 0$
$S_0^2$	$S_0^1, 1$	$S_2^1, 0$
$S_1^1$	$S_1^2, 0$	$S_2^2, 0$
$S_1^2$	$S_1^1, 0$	$S_0^1, 1$
$S_2^1$	$S_2^2, 0$	$S_0^2, 1$
$S_2^2$	$S_2^1, 0$	$S_1^1, 0$

a)

	$x = 0$	$x = 1$
$a$	$a, 1$	$b, 0$
$b$	$c, 0$	$a, 1$
$c$	$b, 0$	$c, 0$

б)

Аналогично могут быть решены задачи деления на  $2^r - 1$ , где  $r$  — целое число.

Рассмотрим попутно на этом примере возможность упрощения таблиц переходов. Будем говорить, что два

сигнала  $S_i$  и  $S_j$  неэквивалентны, если существует некоторая пара комбинаций значений бокового и внешнего входа  $(S_i, x_*)$  и  $(S_j, x_*)$ , для которой значения внешнего выхода или бокового сигнала отличаются. В противном случае сигналы  $S_i$  и  $S_j$  эквивалентны (записывается  $S_i=S_j$ ). Можно показать, что отношение эквивалентности обладает свойством транзитивности, т. е. если  $S_i=S_j$  и  $S_j=S_k$ , то  $S_i=S_k$  [Л. 27].

Из таблицы видно, что  $S_0^1 = S_0^2$ , если  $S_1^2 = S_2^1$ , а  $S_1^2 = S_2^1$ , если в свою очередь  $S_2^2 = S_1^1$ , и, наконец,  $S_2^2 = S_1^2$ , если  $S_0^1 = S_0^2$ . Следовательно,  $S_0^1 = S_0^2, S_1^2 = S_2^1, S_2^2 = S_1^1$ . Эти пары эквивалентных боковых сигналов обозначим соответственно через  $a, b, c$ .

Таблицу переходов, содержащую лишь неэквивалентные боковые сигналы, будем называть неизбыточной.

Неизбыточная таблица переходов приведена в таблице б. Заметим, что эта таблица могла быть получена сразу, если имелся бы навык решения задач такого типа<sup>1</sup>.

Рассмотренные задачи можно трактовать как задачи, предназначенные для распознавания (представления, индикации, классификации) каких-либо свойств входных наборов (кодов). В таких случаях мы интересуемся

<sup>1</sup> Действительно, для того чтобы определить остаток от деления на три (и на любое другое число) числа  $K$  в двоичном коде, достаточно знать значение последнего разряда (внешнего входа данного элемента) и остаток от деления числа, образованного только старшими разрядами ( $[k/2]$ ). Отсюда сразу получается таблица, эквивалентная упрощенной. Для аналогичной задачи с делением на 5 сразу можно написать таблицу

	$x=0$	$x=1$
$a$	$a, 1$	$b, 0$
$b$	$c, 0$	$d, 0$
$c$	$e, 0$	$a, 1$
$d$	$b, 0$	$c, 0$
$e$	$d, 0$	$e, 0$

(Прим. ред.)

лишь выходом крайнего правого элемента сети, принадлежащей рассматриваемой системе.

Односторонние системы могут решать также некоторый класс задач, которые обычно рассматриваются как задачи переработки (преобразования) входных кодов (наборов) в выходные по определенному закону. В этих случаях нас интересуют выходы всех элементов сети.

**Задача 5.** Построить систему для преобразования двоичного кода в код Грея.

Известно, что код Грея может быть получен из двоичного кода сложением его по модулю 2 с тем же кодом, сдвинутым в сторону младших разрядов на один разряд. При этом младший разряд сдвинутого числа теряется<sup>1</sup>. Решение задачи представлено в следующей таблице:

	$x = 0$	$x = 1$
$S_0$	$S_0,0$	$S_1,1$
$S_1$	$S_0,1$	$S_1,0$

Сигнал  $S_0$  соответствует случаю, когда внешний вход предыдущего элемента имеет значение 0, сигнал  $S_1$  — 1. Граничный сигнал —  $S_0$ .

**Задача 6.** Построить систему для преобразования кода Грея в двоичный код.

Известно, что  $i$ -й разряд двоичного кода равен сумме по модулю 2 значений всех разрядов кода Грея, начиная со старшего и кончая  $i$ -м. Решение дано в следующей таблице:

	$x = 0$	$x = 1$
$S_0$	$S_0,0$	$S_1,1$
$S_1$	$S_1,1$	$S_0,0$

Боковой сигнал  $S_0(S_1)$  соответствует случаю, когда текущая сумма по модулю 2 внешних входов со старшего по данный элемент включительно равна 0(1).

**Задача 7.** Построить систему, вычисляющую сумму (разность) двух кодов  $Z=X+Y$ , где

$$X=2^{n-1}x_1+\dots+2^0x_n;$$

$$Y=2^{n-1}y_1+\dots+2^0y_n.$$

Решения этих задач общеизвестны.

<sup>1</sup> См., например, Гитис Э. И. «Преобразователи информации для электронных цифровых вычислительных устройств». ГЭИ, М.—Л., 1961.

Возможно, следует выделить также определенный класс задач на сравнение нескольких кодов (наборов), к которым, в частности, относятся следующие.

**Задача 8** [Л. 1, 2, 39]. Построить одновыходную систему для сравнения двух подаваемых двоичных чисел  $X$  и  $Y$ , внешний выход каждой сети которой равен 1 тогда и только тогда, когда удовлетворяется неравенство  $X \leq Y$ . Здесь

$$X = 2^{n-1}x_1 + \dots + 2^0x_n;$$

$$Y = 2^{n-1}y_1 + \dots + 2^0y_n.$$

Для решения достаточно ввести три боковых сигнала. Пусть  $S_0$  соответствует случаю, когда  $x_i = y_i$  (т. е. когда  $x_i \oplus y_i = 1$ );  $S_1$  — случаю, когда  $x_i > y_i$  (т. е.  $x_i y_i = 1$ );  $S_2$  — случаю, когда  $x_i < y_i$  (т. е.  $\bar{x}_i y_i = 1$ ). Сравнение значений  $x_i$  и  $y_i$  производится слева направо, начиная со старших разрядов. Границное условие —  $S_0$ . Легко сообразить, что таблица переходов и выхода элемента будет иметь следующий вид:

	$x_t = 0, y_t = 0$	$x_t = 1, y_t = 0$	$x_t = 0, y_t = 1$	$x_t = 1, y_t = 1$
$S_0$	$S_0, 1$	$S_1, 0$	$S_2, 1$	$S_0, 1$
$S_1$	$S_1, 1$	$S_1, 0$	$S_1, 0$	$S_1, 0$
$S_2$	$S_2, 1$	$S_2, 1$	$S_2, 1$	$S_2, 1$

Заметим, что для представления трех боковых сигналов требуется два канала, причем соответствующей кодировкой можно добиться, чтобы выход совпадал со значением бокового сигнала в одном из каналов. Легко получить решение для случая, когда должно выполняться строгое неравенство [Л. 39]. Приведенное решение дает ключ к решению следующей задачи, в которой число внешних входов каждого элемента системы равно трем.

**Задача 9** [Л. 1, 2]. Построить систему, внешний выход каждой сети которой равен 1, если выполняется неравенство  $X \leq Y \leq Z$ , где

$$X = 2^{n-1}x_1 + \dots + 2^0x_n;$$

$$Y = 2^{n-1}y_1 + \dots + 2^0y_n;$$

$$Z = 2^{n-1}z_1 + \dots + 2^0z_n,$$

и равен 0 в противном случае.

При решении задачи оказывается удобным разбить синтезируемую систему на две односторонние подсистемы, одна из которых работает на решение  $X \leq Y$ , вторая —  $Y \leq Z$  (см. предыдущую задачу), и выходной преобразователь (состоящий из двухходовой схемы совпадения), который выдает 1 на выход элемента, если значения сигналов на выходе обеих подсистем равны (рис. 1-15).

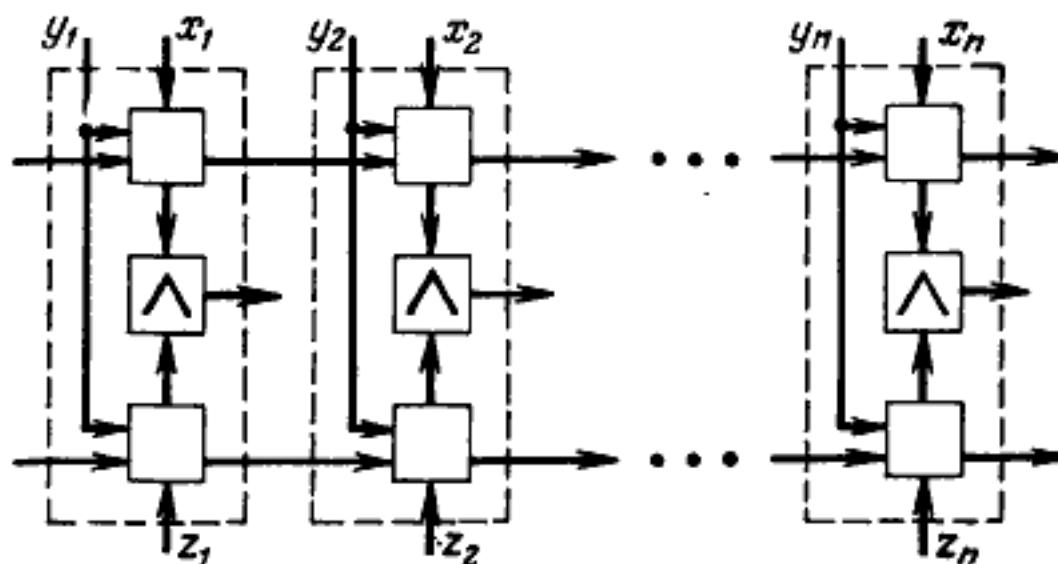


Рис. 1-15. Способ решения задачи 8.

**Задача 10<sup>1</sup>.** Построить систему, внешний выход каждой сети которой равен 1, если выполняется неравенство

$$|X - Y| \leq Z,$$

где  $X$ ,  $Y$  и  $Z$  определяются так же, как в предыдущей задаче, и равен 0 в противном случае.

Элемент системы имеет три внешних входа, на которые подаются одноименные разряды чисел. Можно показать, что при поразрядном выполнении вычитания и сравнения чисел, начиная со старших разрядов, достаточно иметь 7 боковых сигналов, для представления которых необходимы 3 двоичных боковых канала. Неизбыточная таблица переходов и выходов элемента системы (не содержащая эквивалентных сигналов) приведена ниже.

Через  $x_i$ ,  $y_i$ ,  $z_i$  в таблице обозначены внешние входы элемента. Границным сигналом является  $S_0$ .

В некоторых случаях условия задачи могут потребовать более одного внешнего выхода. В качестве примера приведем две задачи (ограничившись только их формулировками).

<sup>1</sup> Решение этой задачи при  $Z=2^m$ ,  $0 \leq m \leq n-1$  дано в [Л. 2].

$x_t$	0	1	0	1	0	1	0	1
$y_t$	0	0	1	1	0	0	1	1
$z_t$	0	0	0	0	1	1	1	1
$S_0$	$S_0, 1$	$S_1, 0$	$S_2, 0$	$S_0, 1$	$S_5, 1$	$S_3, 1$	$S_4, 1$	$S_5, 1$
$S_1$	$S_6, 0$	$S_6, 0$	$S_1, 0$	$S_6, 0$	$S_1, 0$	$S_6, 0$	$S_3, 1$	$S_1, 0$
$S_2$	$S_6, 0$	$S_2, 0$	$S_6, 1$	$S_6, 0$	$S_2, 0$	$S_4, 1$	$S_6, 0$	$S_2, 0$
$S_3$	$S_2, 1$	$S_1, 0$	$S_5, 1$	$S_3, 1$	$S_5, 1$	$S_3, 1$	$S_5, 1$	$S_5, 1$
$S_4$	$S_4, 1$	$S_5, 1$	$S_2, 0$	$S_4, 1$	$S_5, 1$	$S_5, 1$	$S_4, 1$	$S_5, 1$
$S_5$	$S_5, 1$							
$S_6$	$S_6, 0$							

Задача 11 [Л. 1]. Построить систему, внешние выходы каждой сети которой  $z_1, z_2, z_3$  принимают следующие значения:

$$z_1 = \begin{cases} 1, & \text{если } X - Y < Z, \\ 0 & \text{в остальных случаях;} \end{cases}$$

$$z_2 = \begin{cases} 1, & \text{если } X - Y = Z, \\ 0 & \text{в остальных случаях;} \end{cases}$$

$$z_3 = \begin{cases} 1, & \text{если } X - Y > Z, \\ 0 & \text{в остальных случаях,} \end{cases}$$

где  $X, Y, Z$  — положительные или отрицательные двоичные числа.

Задача 12. Построить систему, внешние выходы каждой сети которой  $z_1, \dots, z_5$  принимают следующие значения:

$$z_1 = \begin{cases} 1, & \text{если } X - Y > Z, \\ 0 & \text{в остальных случаях;} \end{cases}$$

$$z_2 = \begin{cases} 1, & \text{если } X - Y = Z, \\ 0 & \text{в остальных случаях;} \end{cases}$$

$$z_3 = \begin{cases} 1, & \text{если } U < X - Y < Z, \\ 0 & \text{в остальных случаях;} \end{cases}$$

$$z_4 = \begin{cases} 1, & \text{если } X - Y = U, \\ 0 & \text{в остальных случаях;} \end{cases}$$

$$z_5 = \begin{cases} 1, & \text{если } X - Y < U, \\ 0 & \text{в остальных случаях.} \end{cases}$$

где  $X, Y, Z, U$  — положительные или отрицательные двоичные числа.

Список задач на построение одномерных односторонних систем из комбинационных элементов можно было бы продолжить [Л. 21].

# ОДНОМЕРНЫЕ ОДНОСТОРОННИЕ СЕТИ И СИСТЕМЫ ИЗ ЭЛЕМЕНТОВ С ПАМЯТЬЮ<sup>1</sup>

## 2-1. Стабильные сети и системы

Нерегулярные сети имеют более чем одно равновесное состояние, соответствующее некоторому набору значений внешних переменных. Этому набору значений внешних переменных может соответствовать более чем один набор значений выходных переменных (какой

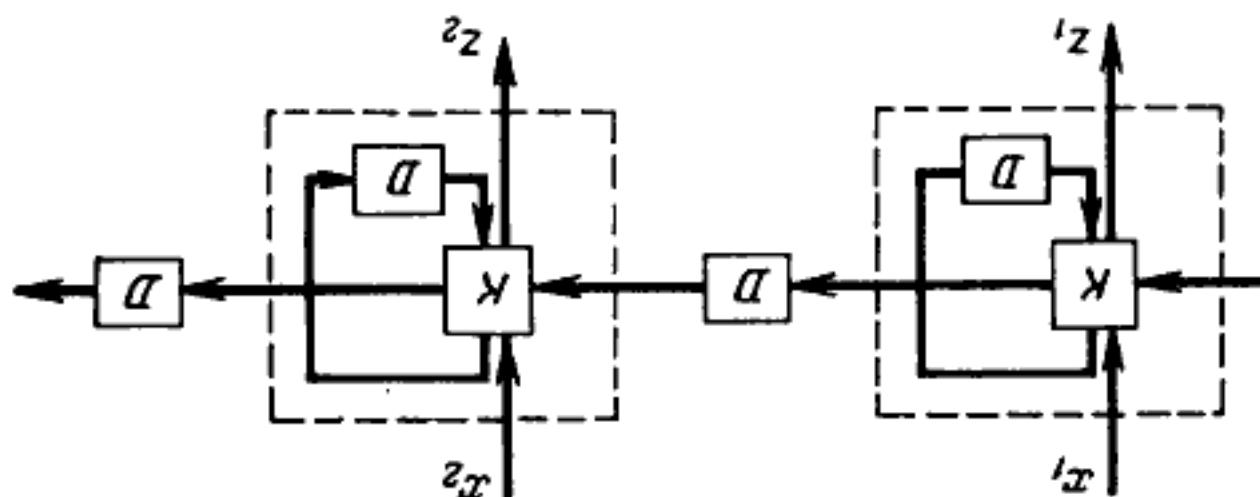


Рис. 2-1. Односторонняя сеть из элементов с памятью.

*K* — комбинационная схема; *D* — единичная задержка.

именно — это зависит от предыстории сети, т. е. от того, в каких состояниях находились элементы сети). Отсюда следует, что перед подачей нового набора переменных большинство сетей из элементов с памятью необходимо устанавливать в одно и то же начальное состояние. Нерегулярной сетью простейшего типа является односторонняя одномерная сеть из элементов с памятью (рис. 2-1).

Если предположить (как обычно), что все элементы обладают единичной задержкой, то от межэлементных задержек можно отказаться, не нарушая общности, что облегчает анализ поведения рассматриваемых сетей и систем (рис. 2-2)<sup>1</sup>.

**Анализ.** При подаче внешних входов односторонняя сеть из элементов с памятью, находящаяся в некотором начальном состоянии, либо переходит в состояние равновесия, либо входит в цикл. Даже в том случае, если

<sup>1</sup> К этому классу схем относятся такие широко известные устройства вычислительной техники, как сдвиговые регистры, счетчики и т. п.

каждому набору внешних входов сети, находящейся в некотором начальном состоянии, соответствует единственное состояние равновесия, нельзя делать вывод о стабильности такой сети, поскольку, возможно, существуют другие начальные состояния, из которых сеть попадает в цикл.

Тест на стабильность сети типа рис. 2-2 мог бы состоять, как кажется на первый взгляд, из проверки на стабильность одного элемента сети для любых начальных состояний и любых значений внешних и боковых входов, поскольку единственным источником возникновения цикла в сети является нестабильность элемента. Действительно, одномерные односторонние сети и системы из стабильных элементов обязательно стабильны. Однако нестабильность элемента еще не означает нестабильности сети, составленной из некоторого числа таких элементов, потому что заданием граничных условий исключается часть возможных значений боковых сигналов. Поэтому тест на стабильность сети должен состоять из последовательного анализа элементов сети, начиная от крайнего левого, по матрице переходов. Если элемент сети имеет  $k$  внутренних состояний и сам по себе не является стабильным, то для проверки сети из  $n$  элементов на стабильность или нестабильность в общем случае следует просмотреть каждое из  $k^n$  состояний (приняв его за начальное) при любом наборе внешних входов, число которых равно  $2^n$ .

Понятно, что если помимо граничных условий задано лишь одно начальное состояние, то трудоемкость проверки на стабильность значительно снижается, хотя и остается весьма большой для больших  $n$ .

Менее трудоемких методов проверки сетей на стабильность, насколько нам известно, предложено не было.

Процедуру проверки на стабильность проиллюстрируем примером. Поведение элемента сети, имеющего

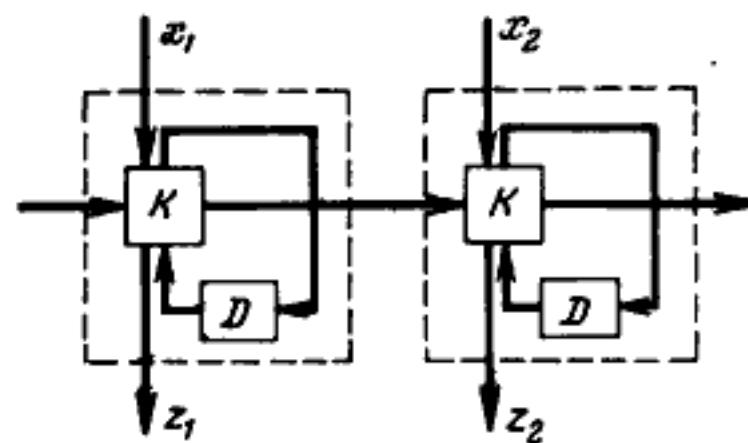


Рис. 2-2. Односторонняя сеть из элементов с памятью.

$K$  — комбинационная схема;  $D$  — единичная задержка.

не означает нестабильности сети, составленной из некоторого числа таких элементов, потому что заданием граничных условий исключается часть возможных значений боковых сигналов. Поэтому тест на стабильность сети должен состоять из последовательного анализа элементов сети, начиная от крайнего левого, по матрице переходов. Если элемент сети имеет  $k$  внутренних состояний и сам по себе не является стабильным, то для проверки сети из  $n$  элементов на стабильность или нестабильность в общем случае следует просмотреть каждое из  $k^n$  состояний (приняв его за начальное) при любом наборе внешних входов, число которых равно  $2^n$ .

Понятно, что если помимо граничных условий задано лишь одно начальное состояние, то трудоемкость проверки на стабильность значительно снижается, хотя и остается весьма большой для больших  $n$ .

Менее трудоемких методов проверки сетей на стабильность, насколько нам известно, предложено не было.

Процедуру проверки на стабильность проиллюстрируем примером. Поведение элемента сети, имеющего

два внутренних состояния  $A$  и  $B$ , два значения внешнего входа  $0$  и  $1$  и три значения боковых сигналов  $S_0, S_1, S_2$ , описывается таблицей

	$x = 0$		$x = 1$	
	$A$	$B$	$A$	$B$
$S_0$	$S_0, A$	$S_1, B$	$S_2, A$	$S_1, A$
$S_1$	$S_0, A$	$S_2, B$	$S_2, A$	$S_1, B$
$S_2$	$S_2, A$	$S_1, A$	$S_0, B$	$S_1, A$

Из рис. 2-3 видно, что при граничном условии  $S_0$ , внутреннем состоянии первого элемента  $A$ , значениях внешних входов первого и второго элемента  $I$  второй элемент входит в цикл.

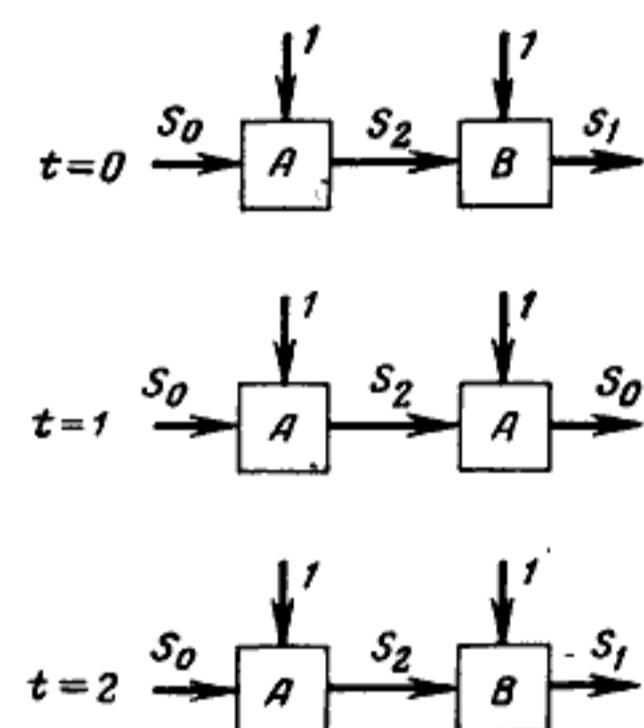


Рис. 2-3. Пример нестабильной односторонней системы.

Для нестабильной сети может найтись такое начальное состояние, для которого она переходит в равновесные состояния при всех возможных наборах значений внешних входов. Отсюда возникает возможность сравнения поведения двух различных сетей.

Про две нерегулярные сети с одинаковым числом элементов говорят, что они эквивалентны, если существуют такие начальные состояния их, при

которых: а) сети не имеют цикла при любых значениях внешних входов; б) имеется взаимооднозначное соответствие между подаваемыми на сети наборами внешних входов и снимаемыми внешними выходами. Аналогично, если каждой сети из  $k$  элементов одной системы найдется эквивалентная сеть другой системы (также из  $k$  элементов),  $k=1, 2, \dots$ , то можно говорить об эквивалентности систем.

**Функциональные возможности.** Попытаемся оценить возможности нерегулярных систем типа рис. 2-2 и сравнить их с возможностями регулярных систем, т. е. систем из комбинационных элементов. В гл. 1 мы говорили о том, что каждому автомата с конечной длиной ленты (т. е. конечной длиной входной последовательно-

сти) можно поставить в соответствие сеть из комбинационных элементов, число элементов в которой равно длине ленты. На этом основании мы делали вывод, что конечному автомату с произвольной длиной ленты соответствует система из комбинационных элементов (и наоборот).

Теперь рассмотрим систему из элементов с памятью. Выделим из этой системы сеть с фиксированным числом элементов. Поскольку эта сеть является конечным автоматом, то ей соответствует некоторая система из комбинационных элементов. Таким образом, каждой сети системы из элементов с памятью соответствует система из комбинационных элементов. Одновременно возникает вопрос о возможностях всего множества систем из элементов с памятью. Насколько нам известно, эта проблема еще не решена.

Тогда системе из элементов с памятью соответствует целый класс систем из комбинационных элементов. Поэтому следует сделать вывод, что некоторые задачи, которые не могут быть решены системами из комбинационных элементов, решаются системами из элементов с памятью. Подтверждением этого вывода является решение следующего примера.

**Задача 1 [Л. 27].** Построить систему, обладающую следующим поведением. Значение сигнала на внешнем выходе элемента системы равно 1, если значение внешних входов этого элемента и элементов слева от него представляет собой серию нулей, длина которой больше, чем длина серии единиц, непосредственно предшествующей этой серии, и равно 0 в остальных случаях.

Как решается такая задача с помощью карандаша и бумаги? Запишем последовательность единиц и нулей, соответствующую набору значений внешних входов системы. Просматривая эту последовательность слева направо, вычеркнем первую единицу в каждой серии единиц и первый нуль в каждой серии нулей, следующей за серией единиц. Затем вычеркнем в каждой серии единиц вторую единицу, если она имеется. Вторые нули вычеркиваются только в тех сериях, которым предшествуют серии из двух или более единиц. Процесс продолжим, пока не будут вычеркнуты все единицы. Оставшиеся невычеркнутыми нули указывают те позиции, на которых внешние выходы должны быть равны 1. Например, вычеркивая (выделяя жирным) по приведенному

алгоритму единицы и нули из последовательности 000111000011, получим: 000111000011. На внешних выходах будут следующие сигналы: 111000000100.

Логическое описание соответствующей структуры элемента сети с двумя внутренними состояниями —  $A$  («не вычеркнуто») и  $B$  («вычеркнуто») и двумя межэлементными (боковыми) сигналами  $S_0$  и  $S_1$  — можно представить в виде таблицы.

	$x = 0$		$x = 1$	
	$A$	$B$	$A$	$B$
$S_0$	$S_0, A, 1$	$S_0, B, 0$	$S_1, B, 0$	$S_0, B, 0$
$S_1$	$S_0, B, 0$	$S_1, B, 0$	$S_1, A, 0$	$S_0, B, 0$

Сначала все элементы сети переводятся в состояние  $A$ . На левый крайний элемент подается граничный боковой сигнал  $S_0$ . Этот сигнал передается слева направо по элементам сети, пока не достигнет элемента с  $x=1$  и внутренним состоянием  $A$ . Он переводит этот элемент в состояние  $B$  («вычеркнуто»), после чего вправо распространяется сигнал  $S_1$ . Когда сигнал  $S_1$  достигнет элемента с  $x=0$  в состоянии  $A$ , он переведет элемент в состояние  $B$ , после чего вправо вновь распространяется боковой сигнал  $S_0$ . Этот процесс повторяется, пока не вычеркиваются все единицы. Боковой сигнал  $S_1$  вновь переходит в  $S_0$ , если достигает первого элемента с  $x=1$  в состоянии  $B$ . Это делается для того, чтобы ошибочно не вычеркнуть 0 из серии нулей, которой не предшествует серия единиц.

Внешний выход элемента будет равен 1 лишь в том случае, если он находится в состоянии  $A$ , а на него приходят боковой сигнал  $S_0$  и внешний вход  $x=0$ \*. После окончания переходного процесса сеть устанавливается в равновесное состояние. Значения выходов этой сети в равновесном состоянии и являются решением задачи.

Заметим, что хотя рассматриваемая система не является регулярной, каждому набору значений внешних

\* Читатель, по-видимому, обратил внимание на то, что описание работы элемента системы во многом напоминает составление функциональной таблицы машины Тьюринга. В гл. 3 будет отмечено, что любой двусторонней системе из элементов с памятью можно поставить в соответствие обладающую таким же поведением машину Тьюринга.

входов соответствует одно и только одно состояние равновесия. Это достигается принудительной начальной установкой всех элементов в исходное состояние  $A$ .

Теперь покажем, что система с требуемым поведением не может быть построена на комбинационных элементах. Доказательство будем вести от противного. Пусть система из комбинационных элементов имеет  $Q$  боковых сигналов. Построим сеть системы, состоящую из  $n=2Q+1$  элементов, у которой внешние выходы первых  $Q$  элементов равны 1, а входы

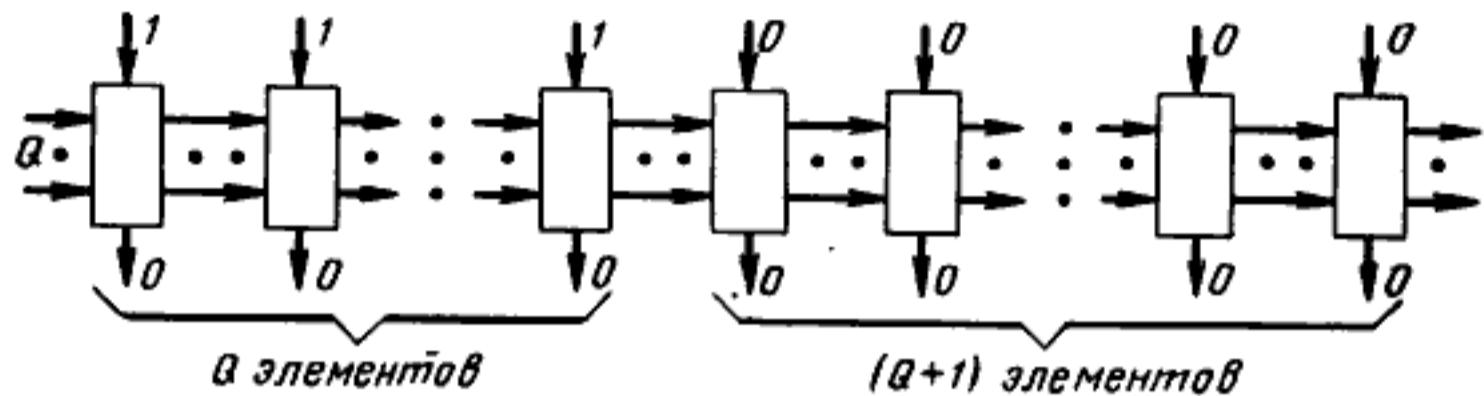


Рис. 2-4. Пример, иллюстрирующий доказательство того, что система с  $Q$  боковыми сигналами не может реализовать произвольную функцию  $n$  переменных,  $n=2Q+1$ .

остальных  $Q+1$  элементов равны 0 (рис. 2-4). Покажем, что такая сеть системы (а следовательно, и собственно система) не обладает заданным поведением.

В соответствии с условиями работы элемента внешние выходы первых  $Q$  элементов должны быть равны 0, поскольку у них  $x=1$ , а боковые сигналы, приходящие на них, должны быть различными. Так как в системе имеется только  $Q$  боковых сигналов, сигнал, приходящий с левой стороны  $(2Q+1)$ -го элемента, должен быть идентичен с сигналом, приходящим слева на один из  $Q$  предыдущих элементов.

Но этот элемент имеет внешний вход, равный 0. Следовательно, и на выходе  $(2Q+1)$ -го элемента должен быть сигнал 0, что противоречит условию. Отсюда, в частности, следует такой результат: существует задача, которая может быть реализована односторонней одномерной одновыходной системой из элементов с памятью, но не может быть реализована односторонней системой из комбинационных элементов.

В [Л. 27] показано, что не существует конечного алгоритма, который позволил бы определить, реализуется ли произвольная задача односторонней системой из комбинационных элементов, если известно, что она

реализуется односторонней системой из элементов с памятью. Это не означает, что нельзя предложить достаточного критерия реализуемости определенной задачи или типа задач, но означает, что нельзя найти универсальный критерий или конечное множество критериев реализуемости любой возможной задачи.

**Синтез.** Рассмотрим две задачи и найдем их решение. Хотя выбор этих задач в определенной степени случаен, тем не менее их внимательное рассмотрение поможет читателю получить достаточный опыт и развить интуицию до такой степени, чтобы он смог решать практические задачи такого типа.

Процедура, как и при решении рассмотренных ранее задач, сводится сначала к поиску «ручного» алгоритма решения и затем к составлению правил поведения на принятом нами языке модифицированных таблиц переходов.

Первая из задач — о реализации класса мажоритарных функций. Напомним, что мажоритарной функцией  $n$  переменных обычно называют булеву функцию, принимающую то значение, которое принимает более половины ее аргументов. Эта задача находит применение при декодировании сообщений. В предлагаемом способе решения сложность реализации зависит от  $n$  (длины кода) линейно, как  $Cn$ , где  $C$  — сложность элемента системы. Переход от одного числа аргументов мажоритарной функции к большему требует лишь увеличения числа элементов, по одному на каждый аргумент.

Мы рассмотрим два варианта решения этой задачи не только для того, чтобы показать возможность использования разных подходов к решению одной и той же задачи, но и для того, чтобы продемонстрировать способы работы с таблицами переходов.

**Задача 2 [Л. 27].** Построить одностороннюю одновходную систему, внешний выход которой равен 1, если среди значений внешних входов больше единиц, чем нулей, и равен 0 в противном случае.

Вначале попытаемся разбить наборы значений внешних входов на классы. Наборы, в которых разность числа единиц и числа нулей одинакова, объединим в один класс. Обозначим  $d = N_0 - N_1$ , где  $N_0$  — число нулей,  $N_1$  — число единиц. Внешний выход системы по условию задачи равен 1, если  $d \leq 0$ . В этом случае для представления подмножеств внешних входов достаточно, как

станет понятно ниже, иметь четыре боковых сигнала:  $S_0$ ,  $S_1$ ,  $S_2$ ,  $S_3$ .

Превышение числа нулей над числом единиц, равное  $d$ , будем обозначать серией из  $d$  сигналов  $S_1$ . Превышение числа единиц над числом нулей, равное  $-d$ , будем обозначать серией из  $d$  сигналов  $S_2$ . Предположим, что серия из  $d$  сигналов  $S_1$  поступает на боковой вход элемента с  $x=0$ . В этом случае на боковом выходе появится серия из  $d+1$  сигналов  $S_1$ . Это может быть сделано заменой первого следующего за серией сигналов  $S_1$  сигнала  $S_0$  на  $S_1$ . Такой серии может быть предпослана серия сигналов  $S_3$  (которая не влияет на интерпретацию последующих серий сигналов  $S_1$  или  $S_2$ ).

Кроме того, всем последовательностям припишем неопределенно длинную серию сигналов  $S_0$ . (Основания необходимости введения этих серий читателю станут понятными при ознакомлении с алгоритмом.)

Например, появление следующей последовательности сигналов:

$$S_3 S_3 S_3 S_3 S_1 S_1 S_1 S_0 \dots S_0$$

между элементами  $j$  и  $j+1$  показывает, что набор значений внешних входов первых  $j$  элементов системы содержит на три нуля больше, чем единиц. На крайний элемент слева в любой сети должен подаваться граничный сигнал  $S_0$ . Если элемент получает временную последовательность, состоящую из  $d$  сигналов  $S_2$ , а на боковом выходе его должна появиться серия из  $d-1$  сигналов  $S_2$ , то это можно осуществить, заменив первый из серий сигналов  $S_2$  на  $S_3$ . Если входная последовательность не содержит сигналов  $S_4$  или  $S_2$ , то первый сигнал  $S_0$  должен быть заменен сигналом  $S_1$  или  $S_2$  в соответствии со значением внешнего входа.

$x = 0$			$x = 1$			
$A$	$B$	$C$	$A$	$B$	$C$	
$S_0$	$S_1, B$	$S_0, B$	$S_0, C$	$S_2, C$	$S_0, B$	$S_0, C$
$S_1$	$S_1, A$	*	*	$S_3, B$	$S_1, B$	*
$S_2$	$S_3, B$	$S_2, C$	$S_2, C$	$S_2, A$	*	*
$S_3$	$S_3, A$	*	*	$S_3, A$	*	*
$z$	0	0	1	0	0	1
.....			.....			

Левая часть таблицы первого варианта<sup>1</sup>, задающей поведение элементов этой системы, соответствует случаю, когда внешний вход равен 0. Каждый элемент первоначально принудительно переводится в состояние  $A$  и остается в нем, пока получает слева только сигналы  $S_3$  и  $S_1$ . Последовательность из сигналов  $S_3$  и  $S_1$ , которые он получает в течение этого времени, повторяется в последовательности, которая им выдается на соседний элемент справа. Если впоследствии получен сигнал  $S_0$ , он заменяется сигналом  $S_1$  в выходной последовательности, индицируя увеличение числа нулевых входных сигналов. Затем элемент переходит в состояние  $B$ , так что впоследствии входящие сигналы  $S_0$  будут повторены в выходной последовательности. Если же после серии сигналов  $S_3$  получен сигнал  $S_2$ , он заменяется на сигнал  $S_3$  в выходной последовательности, и элемент опять-таки переходит в состояние  $B$ . Серия сигналов  $S_2$  показывает, что число единиц в наборе значений внешних входов превышает число нулей и что внешний выход должен быть равен 1. Элемент переводится в состояние  $C$ , ассоциированное с единичным значением внешнего выхода.

Подобным же образом можно расписать правую часть таблицы, в которой описывается поведение элемента при  $x=1$ .

Итак, если все элементы сети первоначально находятся в состоянии  $A$ , граничным сигналом является сигнал  $S_0$ , то на внешнем выходе крайнего правого элемента сети будет сигнал 1, если только число единичных значений внешних входов сети превышает число нулевых значений<sup>2</sup>.

Заметим, что при решении задачи получилось, что внешний выход каждого элемента удовлетворяет услов-

<sup>1</sup> Знакок \* ставится, когда переход может быть задан произвольно.

<sup>2</sup> Если последнюю строку таблицы доопределить так:

$$\begin{array}{ll} x = 0 & x = 1 \\ S_3 \quad S_3, A_2 \quad S_3 \quad A \quad S_3, A \mid S_3, A & S_3, A \quad S_3, A, \end{array}$$

то для работы сети будет достаточно, не устанавливая сеть в исходное состояние, предпослать серии граничных сигналов  $S_0$  один сигнал  $S_3$ . (Прим. ред.)

вию задачи, если рассматривать часть сети, в которую входят этот элемент и все элементы слева от него.

Приведем второй вариант решения задачи.

Процедура решения с помощью карандаша и бумаги могла бы быть также следующей. Если в любом наборе значений внешних входов мы вычеркнем один 0 и одну 1, разница между числом нулей и единиц не изменится. Следовательно, за некоторое число шагов мы можем «упростить» набор, вычеркивая пары 0—1. Если на каком-либо шаге останутся только единицы, мы делаем вывод, что число единиц в первоначальном наборе значений внешних входов превышает число нулей.

Процесс «вычеркивания» может быть выполнен элементом с двумя внутренними состояниями  $A$  («не вычеркнуто») и  $B$  («вычеркнуто») и четырьмя боковыми сигналами.

Таблица переходов элемента системы в этом случае имеет вид (второй вариант):

$x = 0$		$x = 1$	
$A$	$B$	$A$	$B$
$S_0$	$S_1, B$	$S_0, B$	$S_2, B$
$S_1$	$S_1, A$	$S_1, B$	$S_3, B$
$S_2$	$S_3, B$	$S_2, B$	$S_2, A$
$S_3$	$S_3, A$	$S_3, B$	$S_3, A$

Все элементы устанавливаются в состояние  $A$ , граничным сигналом является сигнал  $S_0$ . В течение первого промежутка времени, не превышающего  $k$ , где  $k$  — число элементов в сети, каждый из которых обладает единичной задержкой, первый элемент переходит в состояние  $B$ , а направо посыпаются: сигнал  $S_1$ , если внешний вход этого элемента равен 0, и  $S_2$ , если  $x=1$ . Сигналы  $S_1$  или  $S_2$  распространяются вправо, пока не достигнут элемента с внешним входом 1 и 0 соответственно. Эти элементы также переходят в состояние  $B$ , выдавая вправо сигнал  $S_3$ , который не изменяет внутренних состояний элементов, через которые проходит. В этот промежуток времени устанавливаются местоположения крайнего элемента слева с внешним входом  $x=0$  и ближайшего от него элемента справа с  $x=1$  и происходит «вычеркивание» этих элементов переводом их в состояние  $B$ . В последующие промежутки времени вновь выявляются соответствующие пары элементов. Для того чтобы искомая величина выходного сигнала схемы была равна 1, необходимо, чтобы среди невычеркнутых внешних входов были только единицы. С целью различия внешних выходов вводится третье состояние элемента  $C$ , выполняющее те же функции, что и состояние  $B$  во время процесса вычеркивания пар, но отличающееся от него тем, что оно соответствует внешнему входу  $x=1$ , в то время как  $B$  — входу  $x=0$ . Таким образом, когда боковой сигнал  $S_0$  поступает на элемент с вну-

тrenним состоянием  $A$  и внешним входом 1, элемент переходит в состояние  $C$ , а не  $B$ , что показано ниже. [Пунктиром показаны изменение (по сравнению с предыдущей таблицей) элементы.]

$x = 0$			$x = 1$			
$A$	$B$	$C$	$A$	$B$	$C$	
$S_0$	$S_1, B$	$S_0, B$	$S_0, C$	$\boxed{S_2, C}$	$S_0, B$	$S_0, C$
$S_1$	$S_1, A$	$S_1, B$	$S_1, C$	$S_3, B$	$S_1, B$	$S_1, C$
$S_2$	$S_2, B$	$\boxed{S_2, C}$	$S_2, C$	$S_2, A$	$S_2, B$	$S_2, C$
$S_3$	$S_3, A$	$S_3, B$	$S_3, C$	$S_3, A,$	$S_3, B$	$S_3, C$
$z$	0	0	1	0	0	1

Еще раз отметим, что боковые выходы в состояниях  $B$  и  $C$  одинаковые. Итак, если боковой сигнал  $S_2$  поступает на элемент с состоянием  $B$  и  $x=0$ , элемент переходит в состояние  $C$ . Любой элемент, перешедший в состояние  $C$ , сохраняет его, но элемент продолжает вести себя так, как если бы он находился в состоянии  $B$ .

Поведение сети из 10 элементов, работающих в соответствии с этой таблицей, проиллюстрировано пространственно-временной диаграммой на рис. 2-5. Сверху отмечены внешние входы. Зачернены элементы, внешний выход которых равен 1. Каждая строка рис. 2-5 соответствует вычеркиванию одной пары 0 и 1. Эта диаграмма позволяет наблюдать поведение сети за все время ее работы и способ смены внутренних состояний элементов при воздействии боковых сигналов.

Остается сравнить два способа решения задачи. С первого взгляда решения могут показаться различными. Однако незаполненные места таблицы первого варианта (стр. 57) могут быть доопределены произвольным образом, в том числе и так, что таблицы первого и второго вариантов, а следовательно, и поведение обеих систем будет идентичным.

На этом примере вновь рассмотрим возможность упрощения таблиц переходов. Заметим, что на стр. 45 был дан способ проверки боковых сигналов на эквивалентность. Внутренние состояния элемента играют роль, аналогичную межэлементным сигналам. Вообще говоря, незаполненные места таблицы первого варианта могут быть выбраны любыми, потому что в соответствии с применяемым методом решения задачи в эти состояния элемент попасть не может. Состояние  $C$  не эквивалентно состояниям  $A$  или  $B$ , так как только в этом состоянии

$x=1$ . Состояния  $A$  и  $B$  также не эквивалентны, так как элементы из этих состояний под воздействием сигнала  $S_0$  и  $x=1$  переходят в неэквивалентные состояния  $C$  и  $B$  соответственно. Подобным же образом можно показать, что в таблице на стр. 60 нельзя найти эквивалентных состояний, так что эта таблица уже является неиз-

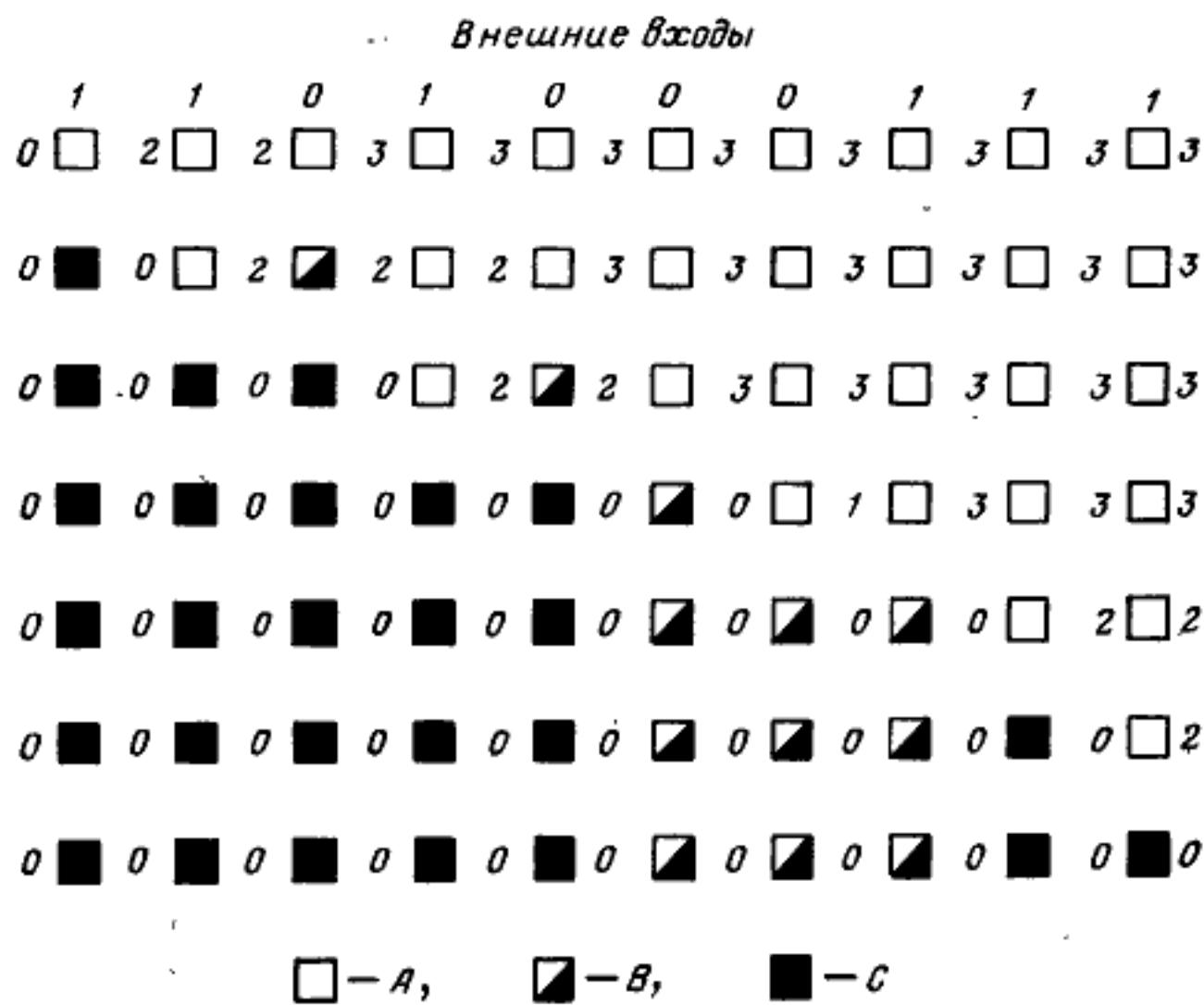


Рис. 2-5. Пространственно-временная диаграмма поведения сети из 10 элементов к задаче 2.

быточной. (Таким образом, *неизбыточной* таблицей теперь мы будем называть такую таблицу переходов, в которой содержатся лишь неэквивалентные боковые сигналы и состояния).

Следует заметить, что элементу одной системы могут соответствовать различные неизбыточные таблицы переходов.

Продемонстрируем на нашем примере возможность перехода от одной неизбыточной таблицы к другой. Потребуем, чтобы каждое внутреннее состояние элемента

соответствовало определенному значению внешнего входа. Тогда из последней таблицы получим таблицу

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>
<i>S</i> <sub>0</sub>	<i>S</i> <sub>1</sub> , <i>B</i>	<i>S</i> <sub>0</sub> , <i>B</i>	<i>S</i> <sub>0</sub> , <i>C</i>	<i>S</i> <sub>2</sub> , <i>F</i>	<i>S</i> <sub>0</sub> , <i>E</i>	<i>S</i> <sub>0</sub> , <i>F</i>
<i>S</i> <sub>1</sub>	<i>S</i> <sub>1</sub> , <i>A</i>	<i>S</i> <sub>1</sub> , <i>B</i>	*	<i>S</i> <sub>3</sub> , <i>E</i>	<i>S</i> <sub>1</sub> , <i>E</i>	*
<i>S</i> <sub>2</sub>	<i>S</i> <sub>3</sub> , <i>B</i>	<i>S</i> <sub>2</sub> , <i>C</i>	<i>S</i> <sub>2</sub> , <i>C</i>	<i>S</i> <sub>2</sub> , <i>D</i>	<i>S</i> <sub>2</sub> , <i>C</i>	<i>S</i> <sub>2</sub> , <i>C</i>
<i>S</i> <sub>3</sub>	<i>S</i> <sub>3</sub> , <i>A</i>	*	*	<i>S</i> <sub>3</sub> , <i>D</i>	*	*
<i>z</i>	0	0	1	0	0	1

*a)*

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
<i>S</i> <sub>0</sub>	<i>S</i> <sub>1</sub> , <i>B</i>	<i>S</i> <sub>0</sub> , <i>B</i>	<i>S</i> <sub>0</sub> , <i>C</i>	<i>S</i> <sub>2</sub> , <i>C</i>
<i>S</i> <sub>1</sub>	<i>S</i> <sub>1</sub> , <i>A</i>	<i>S</i> <sub>1</sub> , <i>B</i>	*	<i>S</i> <sub>3</sub> , <i>B</i>
<i>S</i> <sub>2</sub>	<i>S</i> <sub>3</sub> , <i>B</i>	<i>S</i> <sub>2</sub> , <i>C</i>	<i>S</i> <sub>2</sub> , <i>C</i>	<i>S</i> <sub>2</sub> , <i>D</i>
<i>S</i> <sub>3</sub>	<i>S</i> <sub>3</sub> , <i>A</i>	*	*	<i>S</i> <sub>3</sub> , <i>D</i>
<i>z</i>	0	0	1	0

*б)*

Начальное состояние при  $x=0$  обозначено *A*, при  $x=1$  — *D*. Величина внешнего выхода теперь как бы «запоминается» внутренним состоянием элемента: состояния *A*, *B*, *D*, *E* соответствуют  $z=0$ , состояния *C* и *F* —  $z=-1$ . Состояния *B* и *E*, *C* и *F* теперь стали эквивалентными.

В результате исключения эквивалентных состояний из таблицы *a* (избыточной таблицы) получим таблицу *б*, которая является неизбыточной.

Число состояний элемента в таблице увеличилось на единицу, но зато отпала необходимость в реализации внешнего выхода *z*. Указанные преобразования таблиц переходов могут оказаться полезными на этапе структурного синтеза, когда перед проектировщиком стоит задача соблюсти приемлемое соотношение между сложностями реализации комбинационной части автомата и его памяти. Добавим к этому, что, несмотря на проделанные с таблицами переходов манипуляции, системы из синтезированных по этим таблицам элементов сохраняют эквивалентность. Поэтому решение, полученное в таблице, может быть использовано как иллюстрация обоих способов решения задачи.

Теперь рассмотрим второй, несколько более сложный пример.

**Задача 3 [Л. 27].** Построить одновыходную систему, выход которой равен 1, если на внешних входах имеют место следующие наборы:

```

0 . . . . . . . . 010110 . . . . . . . . 0
0 . . . . . . . . 0110011110 . . . . . . . 0
0 . . . . . . . . 01110001111110 . . . . . 0
. . . . . . . . . . . . . . . . . . . . . . .
0 . . . 0 1 . . . 1 0 . . . 01 . . . 10 . . . 0
      { k }     { k }     { 2k }
. . . . . . . . . . . . . . . . . . . . . . .

```

и равен 0 в противном случае.

Пусть число единиц в первой серии единиц равно  $k$ . Каждую из единиц представим двумя сигналами  $S_1$ , следующими один за другим. Сравним число единиц с числом нулей заменой пар сигналов  $S_1$  на пары  $S_2$ . Если число сигналов  $S_1$  и  $S_2$  совпадает, проверим число единиц в следующей серии, заменяя каждый из сигналов  $S_2$  сигналом  $S_3$ .

Можно выделить шесть классов наборов внешних входов.

1. Любой набор, состоящий только из нулей. Этот набор представляется серией сигналов  $S_0$  (граничным условием также является  $S_0$ ).

2. Наборы, кончающиеся серией из  $k$  единиц, причем эта серия является единственной. Эти наборы представляются серией из  $2k$  сигналов  $S_1$ , за которой следует серия из некоторого числа сигналов  $S_0$ .

3. Наборы, в которых за единственной серией из  $k$  единиц следует серия из  $m$  нулей. Если  $m \leq k$ , то набор представляется последовательностью из  $2m$  сигналов  $S_2$ ,  $2(k-m)$  сигналов  $S_1$  и серией сигналов  $S_0$ . Если  $m > k$ , то набор представляется серией из  $2k$  сигналов  $S_2$  и серией сигналов  $S_4$ . Ясно, что такая последовательность не принадлежит к классу рассматриваемых. Поэтому сигнал  $S_4$  есть сигнал ошибки.

4. Наборы, в которых за серией из  $k$  единиц следует серия из  $k$  нулей и серия из  $p$  единиц. Если  $p \leq 2k$ , то набор может принадлежать классу рассматриваемых и представляется последовательностью из  $p$  сигналов  $S_3$ ,  $2k-p$  сигналов  $S_2$  и серией сигналов  $S_0$ . Если  $p > 2k$ , то набор не принадлежит к классу рассматриваемых и представляется последовательностью из  $2k$  сигналов  $S_3$  и серией сигналов  $S_4$ .

5. Наборы, в которых за опознаваемой конфигурацией следует серия из нулей. Эти наборы представляются последовательностью из  $2k$  сигналов  $S_3$  и серией сигналов  $S_0$ .

6. Наборы, в которых за классом опознаваемых конфигураций следует серия лишних единиц. Эти наборы представляются последовательностью из  $2k$  сигналов  $S_3$  и серией сигналов  $S_4$ .

Если учитывать только наборы, принадлежащие классам 1 и 3 (при  $m < k$ ), то таблица переходов для элемента с внешним входом  $x=0$  будет иметь вид:

	$x = 0$			
	A	B	C	D
$S_0$	$S_0, A$	*	*	$S_0, D$
$S_1$	*	$S_2, C$	$S_2, D$	$S_1, D$
$S_2$	$S_2, B$	$S_2, B$	*	*
$S_3$	*	*	*	*
$S_4$	*	*	*	*

Видно, что нам потребовалось 4 состояния:  $A, B, C, D$ . Состояние  $A$  является начальным. Оно переходит в состояние  $B$  при боковом сигнале  $S_2$ . Состояния  $C$  и  $D$  служат для замены двух следующих друг за другом сигналов  $S_1$  двумя сигналами  $S_2$ .

Учитывая наборы, принадлежащие остальным классам, и рассматривая работу элемента при любых значениях внешнего входа, мы в конце концов придем к таблице вида

	$x = 0$					$x = 1$				
	A	B	C	D	E	A	B	C	D	E
$S_0$	$S_0, A$	$S_4, E$	$S_0, C$	$S_0, D$	$S_4, E$	$S_1, B$	$S_1, C$	$S_0, C$	$S_4, E$	$S_4, E$
$S_1$	$S_2, C$	$S_2, C$	$S_2, D$	$S_1, D$	$S_4, E$	$S_1, A$	*	$S_4, E$	*	$S_4, E$
$S_2$	$S_2, B$	$S_2, B$	$S_4, E$	*	$S_4, E$	$S_3, C$	*	$S_2, C$	$S_3, C$	$S_4, E$
$S_3$	$S_3, C$	*	$S_3, C$	*	$S_4, E$	$S_3, D$	*	*	$S_3, D$	$S_4, E$
$S_4$	$S_4, E$									

В ней появляется пятое состояние  $E$ , в которое элемент переходит, когда оказывается, что набор внешних входов слева от элемента (включая сам элемент) не принадлежит опознаваемым конфигура-

#### Внешние входы:

0	1	1	0	0	1	1	1	1	0
0□	0□	1□	1□	2□	2□	3□	3□	3□	3□3
0□	0■	1□	1■	2■	2■	2■	3■	3■	3■3
0□	0■	0□	1■	1■	2■	2■	2■	3■	3■3
0□	0■	0■	1■	1■	2■	2■	2■	2■	3■3
0□	0■	0■	0■	0■	0■	0■	0■	0■	0■0

□—A, ■—B, ■—C, ■—D, □—C ( $z=1$ )

Рис. 2-6. Пространственно-временная диаграмма к задаче 3.

циям. Внешний выход элемента системы равен единице лишь тогда, когда он находится в состоянии  $C$ , получает боковой сигнал  $S_0$ , а значение его внешнего входа равно 1. Временная диаграмма для одного примера представлена на рис. 2-6.

**Замечание.** В односторонних стабильных системах в течение переходного режима работы на внешних выходах элементов могут появляться неправильные значения сигналов. Для того чтобы эти значения не воспринимались как верные, можно ввести дополнительное значение выходного сигнала, указывающего на то, что система находится в переходном режиме. Конструктивно это может быть выполнено введением дополнительного внешнего выхода  $z_0$ , бокового входа  $c$ , бокового выхода  $C$  и некоторым изменением структуры элемента (рис. 2-7). Границное условие  $c=1$ . Появление сигнала  $z_0=1$  на выходе элемента говорит о том, что все элементы слева от данного достигли равновесного состояния. Заметим, что стабильная сеть достигает устойчивого состояния за время, не превышающее величины  $kn$ , где  $k$  — число внутренних состояний элемента, а  $n$  — число элементов в сети.

Некоторые вопросы, связанные с необходимостью получения сигнала окончания переходного процесса, читатель найдет в [Л. 58].

## 2-2. Нестабильные сети и системы

Системы, описанные в задачах 1, 2 и 3 (§ 2-1), как можно показать, являются стабильными, т. е. любая сеть, принадлежащая этим системам, должна в конечном счете достигать состояния равновесия независимо от набора значений внешних входов. Но имеется ряд задач, для которых, по-видимому, невозможно получить решение, используя стабильные системы. Одной из таких задач является следующая.

**Задача 1.** [Л. 27]. Построить автономную одновходовую сеть, внешний выход которой равен 1, когда число элементов в сети — простое число, и равен 0 в противном случае.

Решения этой задачи посредством использования стабильной сети получить еще не удалось, хотя и не показано, что такая задача

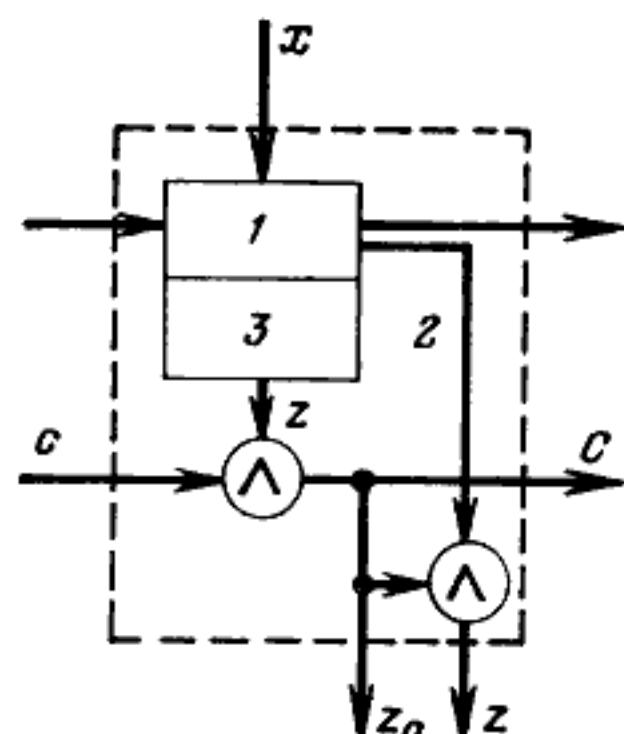


Рис. 2-7. Способ предотвращения появления ложных выходных сигналов в стабильных сетях.

1 — собственно элемент; 2 — его внешний выход; 3 — схема, определяющая момент перехода в стабильное состояние.

не может быть решена стабильной сетью. С другой стороны, эта задача может быть решена системой, сети которой не достигают в конечном счете состояния равновесия. Приведем такую реализацию, которая показывает широкие возможности нестабильных систем.

Структура описываемой системы состоит из двух частей — генератора чисел, вырабатывающего временную последовательность межэлементных сигналов, представляющих целые положительные числа, и схемы проверки, определяющей, кратно ли число элементов сети этим числам. Выход сети равен 1 тогда и только тогда, когда число элементов в сети не делится ни на какие числа, кроме 1 и самого себя.

Генератор чисел имеет три внутренних состояния  $A$ ,  $B$ ,  $C$  и четыре боковых сигнала —  $S_0$ ,  $S_1$ ,  $S_M$ ,  $S_L$ . Сигнал  $S_L$  — это левый граничный сигнал, который появляется лишь на входе крайнего левого элемента, исходное внутреннее состояние элементов генератора —  $A$ .

Пусть последовательность сигналов, появляющихся на выходе  $j$ -го элемента, т. е. между  $j$ -м и  $(j+1)$ -м элементами, обладает периодичностью и имеет вид:

$$\dots \underbrace{S_1, S_1, \dots, S_1}_{j-1}, S_M, S_0, S_0, \dots, S_0, \underbrace{S_1, S_1, \dots, S_1}_{j-1}, S_M, S_0, \dots$$

Каждый период последовательности содержит серию, состоящую из  $(j-1)$ -го сигнала  $S_1$ , за которым следует одиночный сигнал  $S_M$ . За сигналом  $S_M$  в свою очередь следует один (или более) сигнал  $S_0$ , играющий роль пробела. Для удобства изложения последовательность из некоторого числа сигналов  $S_1$  и одиночного сигнала  $S_M$  будем называть блоком. Число символов в блоке, таким образом, характеризует число элементов в сети. Если на выходе  $j$ -го элемента сети появляется последовательность, состоящая из блоков  $S_1, \dots, S_1, S_M$ , разделенных серией из сигналов  $S_0$ , то, значит, этот элемент является  $j$ -м слева. На выходе следующего,  $(j+1)$ -го элемента длина серии сигналов  $S_1$  в блоке должна увеличиться на единицу. Это делается заменой сигнала  $S_M$  в блоке на  $S_1$  и заменой непосредственно следующего за ним сигнала  $S_0$  на  $S_M$ .

Если число сигналов  $S_0$ , разделяющих блоки, фиксировано, то на выходе некоторого элемента сети может образоваться такая последовательность, в которой за блоком следует лишь один сигнал  $S_0$ . Тогда на выходе следующего элемента блоки окажутся без пробела. Поэтому необходимо предпринять меры предосторожности. С этой целью вводится состояние  $C$ . Если за сигналом  $S_M$  следует лишь один сигнал  $S_0$ , то последующий блок сигналов должен быть заменен сигналами  $S_C$ . После такой замены вновь может продолжаться «нормальная работа», а сигналы  $S_0$  должны периодически «восстанавливаться» по мере необходимости.

Таблицы переходов генератора чисел (а) и схемы проверки (б) представлены следующим образом:

	$A$	$B$	$C$		$A$	$B$	$C$
$S_0$	$S_0, A$	$S_M, C$	$S_0, A$	$S_0$	$S_0, A$	$S_1, A$	$S_M, A$
$S_1$	$S_1, A$	*	$S_0, C$	$S_1$	$S_0, B$	$S_1, B$	$S_1, C$
$S_M$	$S_1, B$	*	$S_0, C$	$S_M$	$S_0, C$	$S_M, B$	*
$S_L$	$S_M, B$	$S_0, A$	*				

а)

б)

Пространственно-временная диаграмма для генератора (в), состоящего из пяти элементов, и его схема проверки (г) имеют вид:

$t = 0$	L M 1 1 1 1	0   1   0 0 0
	L 0 M 1 1 1	0   M   M 0 0
	L M 0 M 1 1	0 0 1 1 0
	L 0 0 0 M 1	0 0 0 M M
	L M 1 1 0 M	0 0   1   0 1
$t = 5$	L 0 M 1 0 0	0 0   1   1 0
	L M 0 M 0 0	0 0   M   M M
	L 0 0 0 0 0	0 0 0 1 1
	L M 1 1 1 1	0 0 0 0 1
	L 0 M 1 1 1	0 0 0 0 0
$t = 10$	L M 0 M 1 1	0 0 0 0 0
	L 0 0 0 M 1	0 0 0 0 0
	L M 1 1 0 M	0 0 0 0 0
	L 0 M 1 0 0	0 0 0 0 0
	L M 0 M 0 0	0 0 0 0 0
$t = 15$	L 0 0 0 0 0	0 0 0 0 0
	L M 1 1 1 1	0 0 0   1   0
	L 0 M 1 1 1	0 0 0   1   1
	L M 0 M 1 1	0 0 0   1   1
	L 0 0 0 M 1	0 0 0   M   M
$t = 20$	L M 1 1 0 M	0 0 0 0 1
	L 0 M 1 0 0	0 0 0 0 0
	L M 0 M 0 0	0 0 0 0 0
	L 0 0 0 0 0	0 0 0 0 0
	L M 1 1 1 1	0 0 0 0 0   1
$t = 25$	L 0 M 1 1 1	0 0 0 0 0   1
	L M 0 M 1 1	0 0 0 0 0   1
	L 0 0 0 M 1	0 0 0 0 0   1
	L M 1 1 0 M	0 0 0 0 0   M

в)

г)

Для упрощения записи обозначения боковых сигналов заменены их индексами.

Рассмотрим работу схемы проверки.

Начнем с задачи об автономной одновходной системе, которая выдает внешний сигнал 1 тогда и только тогда, когда число элементов сети кратно трем. Предположим, что на первый элемент сети можно подать последовательность сигналов ...,  $S_0, S_0, \dots, S_0, S_1, S_1$ ,

$S_M, S_0, \dots$ , в которой число 3 представляется блоком  $S_1, S_1, S_M$ ; этому блоку предшествует блок из  $n$  сигналов  $S_0$ , за ней следует серия из неопределенного числа сигналов  $S_0$ . Если поведение элемента описывается таблицей б, то последовательность сигналов, появляющихся между первым и вторым элементами сети, будет состоять из серии сигналов  $S_0$ , за которыми следует блок  $S_1, S_M, S_1$ , за которым, в свою очередь, следует серия сигналов  $S_0$ . Последовательность сигналов, выдаваемых любым элементом направо, будет

0	□	0	□	0	□	0	□	0	□	0	□
0	□	0	□	0	□	0	□	0	□	0	□
1	□	0	□	0	□	0	□	0	□	0	□
1	■	1	□	0	□	0	□	0	□	0	□
M	■	M	■	M	■	0	□	0	□	0	□
0	■	1	■	1	■	1	□	0	□	0	□
0	□	0	■	1	■	1	■	1	□	0	□
0	□	0	□	0	■	M	■	M	■	M	■
0	□	0	□	0	□	0	■	1	■	1	■
0	□	0	□	0	□	0	□	0	□	1	■
<b>□—A ■—B ■—C ■—A(z=1)</b>											

Рис. 2-8. Пространственно-временная диаграмма, иллюстрирующая работу схемы проверки.

режиме, оценивать позиции тех элементов, которые кратны числу, представляющему любым из блоков. Несколько иначе работает схема проверки в первых двух элементах, однако на это не нужно обращать особого внимания потому, что предусмотрена особая схема, блокирующая схемы проверки в первых двух элементах, о которой мы расскажем далее.

Задача дополнительной схемы с памятью, названной на рис. 2-9 преобразователем, состоит в том, чтобы в соответствующее время ввести в схему проверки блоки, выделенные в таблице г. Преобразователь принимает боковые сигналы от предыдущего разряда генератора чисел и выдает их в схему проверки. Границные условия выбираются такими, чтобы в первых двух элементах преобразователь не был задействован. В третьем слева элементе он

отличаться от последовательности, которую этот элемент получает слева, лишь тем, что сигнал  $S_M$  в блоке сместится на одну позицию влево. Поведение этой сети ( начальное состояние —  $A$ , граничный сигнал —  $S_0$ ) во времени показано на рис. 2-8. Заметим, что в каждом элементе, расстояние от которого до левого крайнего элемента сети кратно трем, первым получающим сигналом из блока является сигнал  $S_M$ . Внешний выход равен 1, если элемент находится во внутреннем состоянии  $A$  и получает сигнал  $S_M$  (эти элементы заштрихованы на рис. 2-8). Таким образом, сеть из элементов, поведение которых описывается в таблице г, способна определить, кратно ли трех число элементов в любой части схемы, считая слева.

Затем предположим, что последовательность сигналов, поступающих на первый элемент, содержит, кроме блока  $S_1, S_1, S_M$ , обозначающего число 3, другие блоки вида  $S_1, S_1, \dots, S_1, S_M$ . Если эти блоки, следующие последовательно во времени, отделяются друг от друга по крайней мере одним сигналом  $S_0$ , схема проверки будет работать с каждым блоком независимо. Описанная схема будет обнаруживать позиции тех элементов, которые кратны числам, представляемым любым из блоков. Несколько иначе работает схема проверки в первых двух элементах, однако на это не нужно обращать особого внимания потому, что предусмотрена особая схема, блокирующая схемы проверки в первых двух элементах, о которой мы расскажем далее.

Задача дополнительной схемы с памятью, названной на рис. 2-9 преобразователем, состоит в том, чтобы в соответствующее время ввести в схему проверки блоки, выделенные в таблице г. Преобразователь принимает боковые сигналы от предыдущего разряда генератора чисел и выдает их в схему проверки. Границные условия выбираются такими, чтобы в первых двух элементах преобразователь не был задействован. В третьем слева элементе он

передает на схему проверки блок  $S_1, S_m$ . После того как этот блок кончается (что видно по следующему за ним сигналу  $S_0$ ), преобразователь третьего элемента вырабатывает специальный сигнал, который подготавливает к работе преобразователь четвертого элемента, а сам отключается, не пропуская в дальнейшем блок  $S_1, S_m$  на свою схему проверки. Преобразователь четвертого элемента ждет дополнительного сигнала  $S_0$ , который поступает со схемы генератора третьего элемента, пропускает следующий блок  $S_1, S_1, S_m$  на свою схему проверки и отключается. Когда эта операция завершена, подготавливается к работе преобразователь пятого элемента, и процесс продолжается (нормальная работа схемы проверки показана в таблице а).

Для того чтобы удовлетворить требованиям задачи, нужно также предусмотреть следующее. Заметим, что блок, представляющий число  $j$ , передается на схему проверки  $(j+1)$ -го элемента; следовательно, тот факт, что элементом  $j$  заканчивается схема, длина которой кратна  $j$ , не может быть обнаружен схемой проверки. Поэтому 1 на выходе  $j$ -го элемента появится тогда и только тогда, когда существует некоторое число  $k$ ,  $1 < k < j$ , которое является делителем  $j$ . Если схема проверки  $j$ -го элемента не выдала сигнал 1, то  $j$  есть простое число. Часть схемы с петлей обратной связи в верхнем правом углу рис. 2-9 служит для запоминания того, вырабатывала ли схема проверки когда-либо единицу на выходе элемента. Если 1 была когда-либо выдана, то ее необходимо инвертировать, т. е. выдать сигнал 0.

Итак, элемент системы, имеющий структуру, показанную на рис. 2-9, отвечает условию задачи.

Время, необходимое для выдачи правильного ответа, имеет порядок  $j^2$ , если сигналы кодируются так, как это принято при решении задачи, и  $j \log_2 j$  при двоичном кодировании.

Очевидно, что может быть получено решение следующей задачи.

**Задача 2 [Л. 27].** Построить систему, все сети которой имеют выход, равный 1, если существует целое число  $m$  ( $1 < m < n$ ,  $n$  — число элементов сети) такое, что сеть может быть разделена на  $m$  блоков равной длины, каждый из которых содержит одинаковое число внешних входов, равных единице, и равен 0 в противном случае.

Обобщением этой задачи является построение системы, сети которой определяют, существует ли положи-

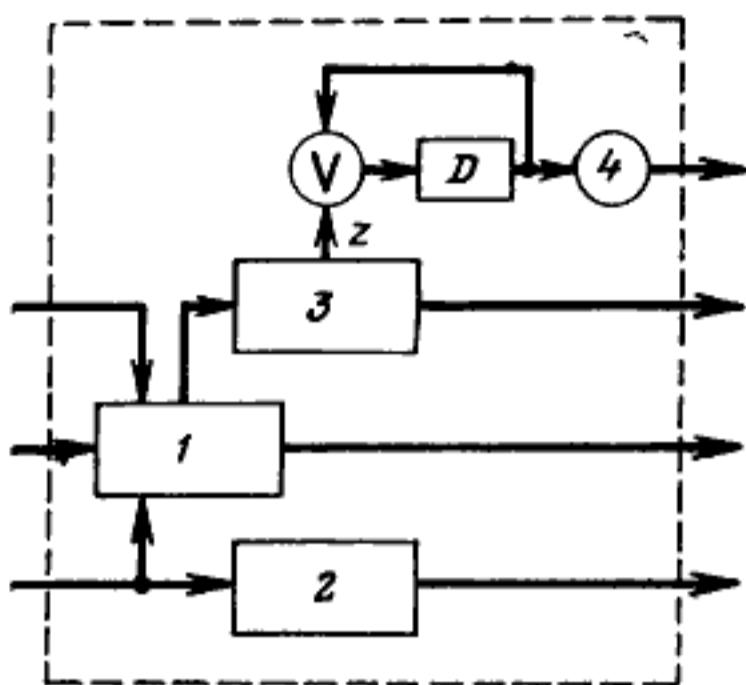


Рис. 2-9. Структура элемента системы.

1 — преобразователь; 2 — генератор чисел; 3 — схема проверки; 4 — инвертор;  $D$  — единичная задержка.

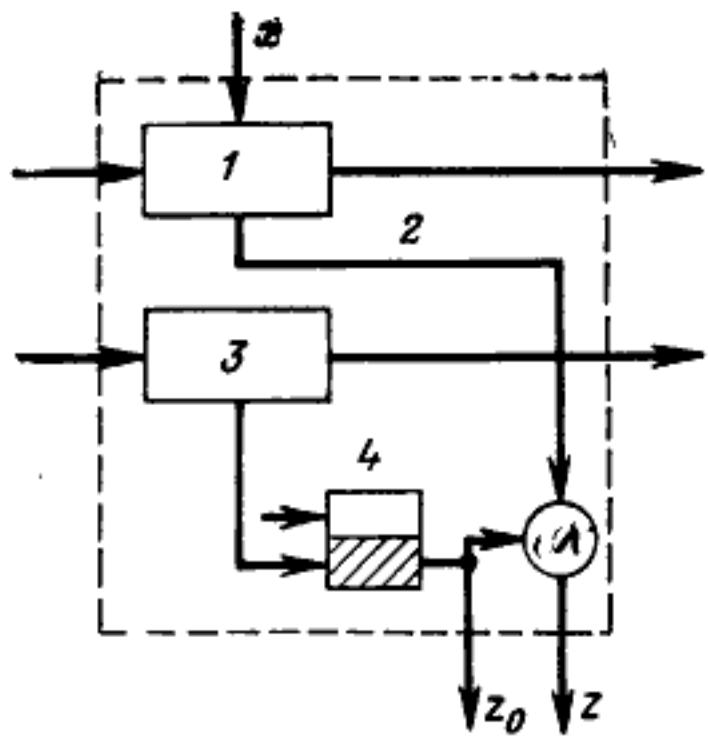


Рис. 2-10. Способ предотвращения появления ложных выходных сигналов в нестабильных сетях.

1 — собственно элемент; 2 — его внешний выход; 3 — разряд  $(k+1)$  счетчика, 4 — триггер переполнения счетчика.

сигнала (рис. 2-7), должен быть внесен разряд  $(k+1)$ -ичного счетчика (рис. 2-10) и триггер, запоминающий переполнение счетчика.

### 2-3. Реализация булевых функций одномерными сетями из элементов с памятью<sup>1</sup>

Пусть задана функция  $n$  переменных  $f(x_1, \dots, x_n)$ . Запишем ее двоичным кодом длины  $2^n$ , так чтобы значение функции на  $i$ -м наборе ( $i = 2^0x_1 + \dots + 2^{n-1}x_n$ ) было равно  $(i+1)$ -му символу кода. Совершенно очевидно, что при таком способе задания функции процесс вычисления ее значения на заданном наборе сводится к нахождению соответствующей позиции в коде. Так, например, если при вычислении функции четырех переменных предъявлен набор  $x_1=1, x_2=0, x_3=1, x_4=0$ , номер которого равен 5, то шестой символ кода представ-

<sup>1</sup> Идея способа реализации принадлежит А. Я. Макаревскому (см. А. Я. Макаревский. О реализации булевых функций одномерной итеративной сетью автоматов. — «Автоматика и телемеханика», 1970, № 11). В его конструкции используются двусторонние сети из автоматов, имеющих 7 внутренних состояний. Конструкция, предлагаемая читателю, отличается, на наш взгляд, большей наглядностью.

тельное целое число, имеющее некоторое специфическое свойство по отношению к набору значений внешних входов.

В связи с замечанием на стр. 65 продемонстрируем возможность устранения выдачи ложных значений сигналов, появляющихся на внешних выходах односторонних нестабильных сетей в течение времени переходного процесса. Внешние выводы сетей таких систем достигают установившихся значений за время, не превышающее  $k^n$ . Поэтому в структуру элемента, предотвращающего преждевременную выдачу внешнего выходного

ляющего функцию, является ее значением на этом наборе.

Такой способ вычисления может быть реализован сетями из элементов с памятью типа рис. 2-2. Сеть, реализующая функцию  $n$  переменных, состоит из  $n+2^n$  элементов, среди которых первые  $n$  элементов используются для подачи входного набора, а остальные  $2^n$  элементов — для задания функции.

Каждый элемент сети имеет 6 состояний:  $A, B, C, D, E, F$ .

Перед началом работы на боковой вход первого элемента подается сигнал  $S_0$ , первые  $n$  элементов устанавливаются в состояние  $A$ , а остальные — в состояние  $D$ . После такой начальной установки на внешние входы первых  $n$  элементов подается входной набор, а на внешние входы остальных  $2^n$  элементов слева направо — значения функции на наборах от нулевого до  $(2^n-1)$ -го соответственно. При этом правила перехода элемента (см. таблицу) таковы, что сеть находится в состоянии равновесия.

$$x=0$$

	$A$	$B$	$C$	$D$	$E$	$F$
$S_0$	$A, S_0, 0$	*	$A, S_0, 0$	$D, S_0, 0$	$D, S_0, 0$	$D, S_0, 0$
$S_1$	$C, S_1, 0$	$C, S_1, 0$	$B, S_2, 0$	$F, S_3, 0$	$E, S_1, 0$	$F, S_3, 0$
$S_2$	$B, S_2, 0$	$B, S_2, 0$	$C, S_2, 0$	$E, S_0, 0$	$E, S_2, 0$	$F, S_3, 0$
$S_3$	*	*	*	$D, S_3, 0$	*	*
$S_4$	*	*	*	$D, S_4, 1$	*	*

$$x=1$$

	$A$	$B$	$C$	$D$	$E$	$F$
$S_0$	$A, S_0, 0$	*	$A, S_0, 0$	$D, S_0, 0$	$D, S_0, 0$	$D, S_0, 0$
$S_1$	$B, S_2, 0$	$C, S_1, 0$	$B, S_2, 0$	$F, S_4, 1$	$E, S_1, 0$	$F, S_4, 1$
$S_2$	$C, S_2, 0$	$B, S_2, 0$	$C, S_2, 0$	$E, S_0, 0$	$E, S_2, 0$	$F, S_4, 1$
$S_3$	*	*	*	$D, S_3, 0$	*	*
$S_4$	*	*	*	$D, S_4, 1$	*	*

Вычисление значения функции начинается с подачи на боковой вход первого элемента сигнала  $S_1$  вместо сигнала  $S_0$ .

В соответствии с принятым способом вычисления значения функции на первых  $n$  элементах сети организуется двоичная пересчетная схема. Для хранения 0 или 1 в каждом разряде пересчетной схемы используются состояния  $B$  или  $C$  соответственно. В каждом такте работы сети происходит вычитание единицы из текущего значения кода, хранимого в пересчетной схеме. В первом такте по сигналу  $S_1$  в пересчетную схему заносится код, на единицу меньший код входного набора. На выходе пересчетной схемы появляется сигнал  $S_1$ , если текущее значение кода равно 0, и  $S_2$  в противном случае. Работа элемента пересчетной схемы полностью описывается частью таблицы, включающей столбцы  $A$ ,  $B$ ,  $C$ .

Ясно, что число сигналов  $S_2$ , появившихся на выходе пересчетной схемы (на выходе  $n$ -го элемента сети), равно номеру набора значений внешних переменных, поданному на сеть. Каждый сигнал  $S_2$ , распространяясь по сети, переводит первый встретившийся ему элемент, находящийся в состоянии  $D$ , в состояние  $E$ ; при этом направо передается сигнал  $S_0$ .

Если на выходе пересчетной схемы появился сигнал  $S_1$ , то этот сигнал переводит первый встретившийся элемент, находящийся в состоянии  $D$ , в состояние  $F$ . При этом направо посыпается сигнал  $S_3$ , если значение внешнего входа равно 0, и сигнал  $S_4$ , если оно равно 1. Сигналы  $S_3$  и  $S_4$  распространяются без изменения до правого края сети. Значения внешнего выхода элемента сети (искомое значение функции) равно 0, если на его боковом выходе появляется сигнал  $S_3$ , и 1, если на боковом выходе появляется сигнал  $S_4$ . Таким образом, на внешнем выходе крайнего правого элемента сетирабатывается искомое значение реализуемой функции.

Заметим, что пересчетная схема продолжает работать и после вычисления искомого значения функции, т. е. сеть, как и в задаче 1 (§ 2-2), является нестабильной. Однако внешний выход сети остается в этом случае неизменным. Для установки схемы в исходное состояние достаточно подать на боковой вход ее правого элемента сигнал  $S_0$ .

Нетрудно показать, что при реализации любой булевой функции  $n$  аргументов требуется по порядку не меньше чем  $2^n$  элементов. Таким образом, описанная конструкция с точностью до порядка является минимальной.

## ДВУСТОРОННИЕ ОДНОМЕРНЫЕ СЕТИ И СИСТЕМЫ

Двусторонняя одномерная сеть представлена на рис. 3-1. Как видно из рисунка, двусторонние сети отличаются от односторонних возможностью передачи ин-

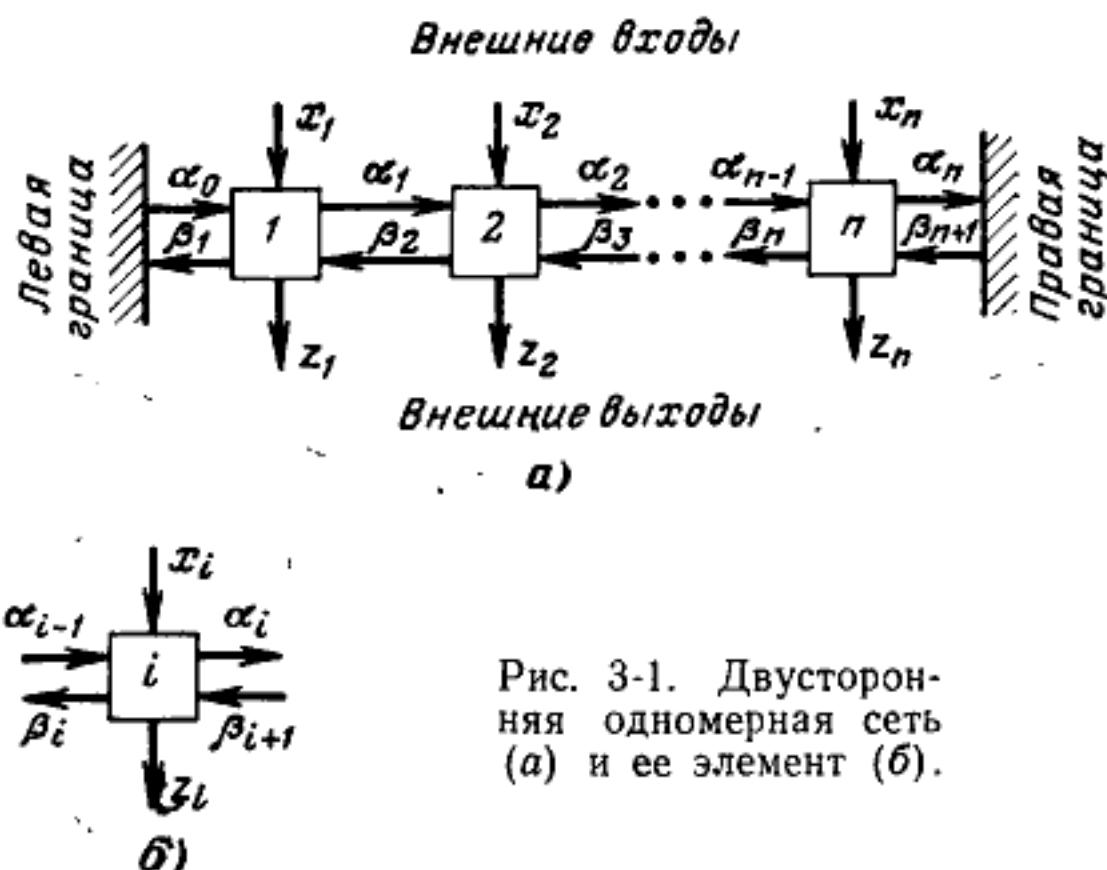


Рис. 3-1. Двусторонняя одномерная сеть (a) и ее элемент (б).

формации не только слева направо, но и справа налево. В общем случае внешний и боковой выходы каждого элемента зависят от внешнего и боковых входов на этот элемент. Поэтому не исключена возможность возникновения обратных связей, а следовательно, и циклов.

### 3-1. Двусторонние одномерные системы из комбинационных элементов

**Анализ.** Так как типовой комбинационный элемент не обладает памятью, то по полному состоянию элемента, которое определяется набором значений всех его входов, могут быть найдены единственным образом значения всех его выходов. Таким образом, работа типового элемента полностью описывается таблицей значений всех его выходов, составленной для всех возможных наборов значений входов. Эта таблица совместно с граничными условиями является отправной точкой анализа таких систем.

Ясно, что если значения правых (левых) боковых выходов элемента двусторонней сети не зависят от правых (левых) боковых входов соответственно, т. е. (рис. 3-1,б)

$$\alpha_i = f(x_i, \alpha_{i-1});$$

$$\beta_i = f(x_i, \beta_{i+1}),$$

то сеть из таких элементов регулярна. Аналогичный вывод можно сделать и в том случае, если

$$\alpha_i = f(x_i, \alpha_{i-1}, \beta_{i+1}), \text{ но } \beta_i = f(x_i, \beta_{i+1}),$$

или

$$\alpha_i = f(x_i, \alpha_{i-1}), \text{ но } \beta_i = f(x_i, \beta_{i+1}, \alpha_{i-1}).$$

Рассмотрим двустороннюю одномерную систему из комбинационных элементов общего типа, когда на нее не накладывается дополнительных ограничений, т. е. когда  $\alpha_i$  и  $\beta_i$  существенно зависят от всех трех аргументов:

$$\alpha_i = f(x_i, \alpha_{i-1}, \beta_{i+1});$$

$$\beta_i = f(x_i, \beta_{i+1}, \alpha_{i-1}).$$

Попытаемся найти условия существования равновесных состояний<sup>1</sup>. Множество состояний, в которых может находиться произвольный элемент  $j$  при установлении равновесия в сети, определяется значениями внешних входов слева от элемента  $j$ , значениями внешних входов справа от элемента  $j$ , значениями внешнего входа самого элемента  $j$ .

Каждая из этих трех групп определяет свое множество полных состояний элемента  $j$ . Если для некоторого набора значений внешних входов существует по крайней мере один элемент, для которого пересечение этих трех множеств полных состояний пусто, то очевидно, что для этого набора не существует равновесного состояния сети.

Способ определения наличия равновесных состояний легче всего пояснить на примере. На рис. 3-2 приведен пример элемента двусторонней системы, который задается в виде полностью определенной таблицы истинно-

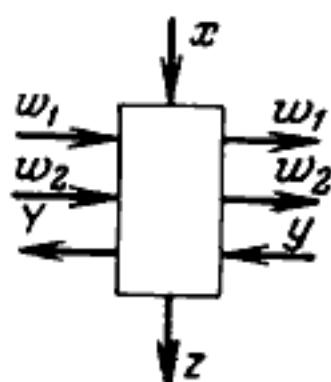
<sup>1</sup> Имеет место следующая теорема [Л. 27]: если сеть или система из комбинационных элементов стабильна, то она должна быть также и регулярна. Доказательство приводится в гл. 4.

сти (в этом параграфе рассматривается только случай, когда таблица истинности элемента определена полностью). Эта таблица составлена таким образом, что все переменные, относящиеся к каждой стороне элемента, сгруппированы вместе.

$S_i$	$x$	$w_1$	$w_2$	$Y$	$W_1$	$W_2$	$y$	$z$
$S_0$	0	0	0	0	0	1	0	0
$S_1$	0	0	0	0	0	1	1	0
$S_2$	0	0	1	0	0	0	0	0
$S_3$	0	0	1	0	0	0	1	0
$S_4$	0	1	0	0	0	0	0	0
$S_5$	0	1	0	0	0	0	1	0
$S_6$	0	1	1	0	0	1	0	0
$S_7$	0	1	1	0	0	1	1	0
$S_8$	1	0	0	0	1	0	0	0
$S_9$	1	0	0	1	1	0	1	1
$S_{10}$	1	0	1	0	1	0	0	0
$S_{11}$	1	0	1	1	1	0	1	1
$S_{12}$	1	1	0	0	1	1	0	0
$S_{13}$	1	1	0	1	1	1	1	1
$S_{14}$	1	1	1	1	1	1	0	1
$S_{15}$	1	1	1	1	1	1	1	1

Предположим, что сеть из таких элементов находится в равновесии. Определим, в каких состояниях может находиться элемент  $j$ , если  $(j-1)$ -й элемент находится

Рис. 3-2. Элемент двусторонней системы (пример). Границные условия системы: слева —  $w_1=w_2=0$ , справа —  $y=0$ .



в состоянии  $S_0$ . Из таблицы на рис. 3-2 находим, что  $(j-1)$ -й элемент имеет правые выходы  $W_1=0$ ,  $W_2=1$  и правый вход  $y=0$ . Тогда, чтобы  $(j-1)$ -й и  $j$ -й элементы

находились в состоянии равновесия,  $j$ -й элемент должен иметь значения левых входов  $w_1=0$ ,  $w_2=1$  и значение левого выхода  $Y=0$ . Из таблицы видим, что значения переменных  $w_1=0$ ,  $w_2=1$  и  $Y=0$  слева от элемента соответствуют состояниям  $S_2$ ,  $S_3$ ,  $S_{10}$ .

Повторив ту же процедуру для остальных состояний  $S_1-S_{15}$ , получим таблицу, которую условно будем называть *матрицей «левых» состояний элемента (a)*. Аналогично получена *матрица «правых» состояний элемента (б)*, в которой подмножества состояний  $(j+1)$ -го элемента поставлены в соответствие состояниям  $j$ -го элемента. Каждая из таблиц «левых» и «правых» состояний элемента для облегчения дальнейшего анализа разбита на две части, относящиеся к различным значениям внешнего входа 0 и 1  $(j-1)$ -го и  $(j+1)$ -го элементов соответственно.

	$x = 0$	$x = 1$	$x = 0$	$x = 1$
$S_0$	$S_2, S_3, S_{10}$		$S_2, S_4$	$S_0$
$S_1$	$S_{11}$		$S_2, S_4$	$S_1$
$S_2$	$S_0, S_1, S_8$		$S_0, S_6$	$S_2$
$S_3$	$S_9$		$S_0, S_6$	$S_3$
$S_4$	$S_0, S_1, S_8$		$S_8, S_{10}$	$S_4$
$S_5$	$S_9$		$S_8, S_{10}$	$S_5$
$S_6$	$S_2, S_3, S_{10}$		$S_{12}, S_{14}$	$S_6$
$S_7$	$S_{11}$		$S_{12}, S_{14}$	$S_7$
$S_8$		$S_4, S_5, S_{12}$	$S_2, S_4$	$S_8$
$S_9$		$S_{13}$	$S_3, S_5$	$S_9$
$S_{10}$		$S_4, S_5, S_{12}$	$S_0, S_6$	$S_{10}$
$S_{11}$		$S_{13}$	$S_1, S_7$	$S_{11}$
$S_{12}$		$S_6, S_7$	$S_8, S_{10}$	$S_{12}$
$S_{13}$		$S_{14}, S_{15}$	$S_9, S_{11}$	$S_{13}$
$S_{14}$		$S_6, S_7$	$S_{13}, S_{15}$	$S_{14}$
$S_{15}$		$S_{14}, S_{15}$	$S_{13}, S_{15}$	$S_{15}$

Используя эти матрицы, нетрудно найти подмножество состояний, в которых может находиться данный элемент, если известны значения внешних входов всех элементов слева (справа) от него и соответствующие граничные сигналы. Для всех возможных наборов значений внешних входов слева (справа) от элемента опре-

делим все такие подмножества и обозначим различные подмножества состояний различными символами. Состояния левого крайнего элемента, определяемые граничными сигналами, обозначим символом  $a_0$ . Если внешний вход первого элемента 0, то множество состояний второго элемента, если оно не совпадает с множеством  $a_0$ , обозначим символом  $a_1$ . Если первый элемент имеет значение внешнего входа 1, то подмножество состояний, в которых может находиться второй элемент, обозначается символом  $a_2$ , если это подмножество отлично от подмножеств  $a_0$  и  $a_1$ , и т. д. Процесс прекращается, как только перестают возникать новые подмножества состояний.

В нашем примере левые граничные сигналы  $w_1 = w_2 = 0$ . Пользуясь таблицей истинности к рис. 3-2, определяем, что крайний левый элемент может находиться в состояниях  $S_0, S_1, S_8$  и  $S_9$ . Это подмножество обозначим  $a_0$ . Допустим, что некоторый  $(j-1)$ -й элемент находится в одном из состояний подмножества  $a_0$  и имеет значение внешнего входа 0. Из матрицы «левых» состояний выпишем все состояния, которые принадлежат столбцу 0 и строкам  $S_0, S_1, S_8$  и  $S_9$ . Тогда  $j$ -й элемент должен быть в одном из состояний  $S_2, S_3, S_{10}$  или  $S_{11}$ . Обозначим это новое подмножество  $a_1$ . С другой стороны, если  $(j-1)$ -й элемент находится в одном из состояний подмножества  $a_0$  и имеет внешний вход 1, то  $j$ -й элемент должен принимать состояния из подмножества  $a_2 = \{S_4, S_5, S_{12}, S_{13}\}$ . Таким образом, подмножество состояний  $a_0$   $(j-1)$ -го элемента под воздействием значений 0 и 1 его внешнего входа переходит в подмножества состояний  $a_1$  и  $a_2$   $j$ -го элемента соответственно. Действуя подобным образом, можно построить матрицу «левых» символов ( $a$ ).

	$x = 0$	$x = 1$	
$a_0$	$a_1$	$a_2$	$a_0 = \{S_0, S_1, S_8, S_9\}$
$a_1$	$a_0$	$a_2$	$a_1 = \{S_2, S_3, S_{10}, S_{11}\}$
$a_2$	$a_0$	$a_3$	$a_2 = \{S_4, S_5, S_{12}, S_{13}\}$
$a_3$	$a_1$	$a_3$	$a_3 = \{S_6, S_7, S_{14}, S_{15}\}$
			$a)$

Аналогичным способом, зная правые граничные условия, нетрудно найти подмножества состояний, обозна-

чаемые символами  $b_0, b_1, \dots$  и построить матрицу «правых» символов (табл. б)

$x = 0 \quad x = 1$

$b_0$	$b_1$	$b_0$	$b_0 = \{S_0, S_2, S_4, S_6, S_8, S_{10}, S_{12}, S_{14}\}$
$b_0$	$b_2$	$b_1$	$b_1 = \{S_0, S_2, S_4, S_6, S_8, S_{10}, S_{13}, S_{15}\}$
$b_0$	$b_3$	$b_2$	$b_2 = \{S_0, S_2, S_4, S_6, S_9, S_{11}, S_{13}, S_{15}\}$
$b_0$	$b_3$	$b_3$	$b_3 = \{S_1, S_3, S_5, S_7, S_9, S_{11}, S_{13}, S_{15}\}$
<b>б)</b>			

Эта матрица дает нам возможность по набору значений внешних входов найти последовательность «правых» символов, определяющих подмножества состояний, которые могут принимать элементы сети в зависимости от значений внешних входов справа от этих элементов. Заметим, что матрицы символов не накладывают ограничений на число элементов в сети и действительно описывают систему сетей, составленных из элементов данной структуры.

Теперь составим еще одну матрицу, строками которой являются «левые» символы, столбцами — «правые» символы, а элементами — подмножества, образованные пересечением подмножеств, входящих в пару «левый» символ — «правый» символ, т. е.  $a_i \cap b_j$ . Эту матрицу назовем *матрицей пересечений*.

В нашем примере матрица пересечений будет иметь вид:

	$b_0$	$b_1$	$b_2$	$b_3$
$a_0$	$S_0, S_8$	$S_0, S_8$	$S_0, S_9$	$S_1, S_9$
$a_1$	$S_2, S_{10}$	$S_2, S_{10}$	$S_2, S_{11}$	$S_3, S_{11}$
$a_2$	$S_4, S_{12}$	$S_4, S_{13}$	$S_4, S_{13}$	$S_5, S_{13}$
$a_3$	$S_6, S_{14}$	$S_6, S_{15}$	$S_6, S_{15}$	$S_7, S_{15}$

Ее элемент, соответствующий символам  $a_0$  и  $b_0$ , находится следующим образом:

$$a_0 = \{S_0, S_1, S_8, S_9\},$$

$$b_0 = \{S_0, S_2, S_4, S_6, S_8, S_{10}, S_{12}, S_{14}\},$$

$$a_0 \cap b_0 = \{S_0, S_8\}.$$

По матрице пересечений нетрудно определить, является ли данная система регулярной или нет. Из того, что мы говорили, следует, что каждому значению внешнего входа в любом элементе матрицы пересечений регулярной системы должно соответствовать в точности

одно состояние. Из таблицы видно, что система, составленная из элементов на рис. 3-2, регулярия.

Если в каком-либо элементе матрицы пересечений какому-либо значению внешнего входа соответствует не в точности одно состояние, то система не регулярна. В частности, если имеется два или более состояний, то система при некоторых значениях внешних входов имеет два или более равновесных состояний; если не имеется ни одного состояния, то система не достигает состояния равновесия и, следовательно, имеет цикл.

Итак, двусторонняя одномерная система из комбинационных элементов регулярна тогда и только тогда, когда левый символ, правый символ и значение внешнего входа определяют точно одно равновесное состояние элемента.

Для полного описания поведения регулярной сети или системы необходимо определить значения внешнего выхода каждого элемента. Эта информация содержится в *матрице внешних выходов*. Элементами матрицы являются значения внешнего выхода для каждого из значений внешнего входа. Алгоритм построения этой матрицы очень простой: по таблице истинности отыскивается значение внешнего выхода для каждого элемента матрицы пересечений (для  $x=0, x=1$ ):

	$b_0$	$b_1$	$b_2$	$b_3$
$a_0$	0,0	0,0	0,1	0,1
$a_1$	0,0	0,0	0,1	0,1
$a_2$	0,0	0,1	0,1	0,1
$a_3$	0,1	0,1	0,1	0,1

Мы привели алгоритм проверки двусторонней системы из комбинационных элементов на регулярность. Если в результате анализа мы сделали вывод о нерегулярности системы, это не значит, что какая-либо ее сеть из ограниченного числа элементов не может быть регулярной. Если петля обратной связи при некоторых граничных условиях замыкается через  $k$  элементов сети, то сеть из  $n < k$  элементов при этих граничных условиях регулярна. Анализ двусторонней сети на наличие или отсутствие равновесных состояний может быть проведен способом, описанным В. Килмером [Л. 30], который, однако, не эффективен при больших  $k$ . Некоторые во-

просы динамики (переходного поведения) одномерных сетей из элементов с памятью рассматриваются в [Л. 31]. В частности, показано, что если сеть находится в состоянии равновесия в некоторый момент времени, то при смене значений внешних переменных в следующий момент времени сеть в дальнейшем может либо войти в цикл, либо перейти в равновесное состояние, что зависит от числа элементов в сети.

Матрицы символов и выходов полностью описывают равновесное поведение системы.

Однако некоторые символы этих матриц могут быть избыточными, так как в приведенном выше методе анализа все построения проводились с использованием таблицы истинности элемента, в которой не были устранины несущественные переменные. Будем говорить, что два левых (правых) символа не эквивалентны, если

а) строки или столбцы матрицы выходов, соответствующие этим символам, различны или

б) строки левой (правой) матрицы символов, соответствующие этим символам, различны.

Если эти условия одновременно не выполняются, то два левых (правых) символа эквивалентны. Эквивалентные символы могут быть объединены, если использовать ту же процедуру, которая предлагалась для устранения избыточных боковых сигналов в гл. 1 (стр. 45).

В нашем примере по таблице выходов для  $x=0$ ,  $x=-1$  делаем предположение, что  $a_0=a_1$  и  $b_2=b_3$ . Это предположение оказывается верным, потому что строкам  $a_0$  и  $a_1$ ,  $b_2$  и  $b_3$  в таблице «левых» и «правых» символов (стр. 77) в этом случае соответствуют одинаковые строки. В результате имеем следующие неизбыточные матрицы символов и выходов ( $a$  — левая матрица символов,  $b$  — правая матрица символов,  $c$  — матрица выходов):

$x = 0$			$x = 1$			$x = 0$			$x = 1$			$b_0$	$b_1$	$b_2$
$a_1$	$a_1$	$a_2$	$b_0$	$b_1$	$b_0$	$a_1$	$0,0$	$0,0$	$b_0$	$b_1$	$b_2$	$a_1$	$0,0$	$0,1$
$a_2$	$a_1$	$a_3$	$b_0$	$b_2$	$b_1$	$a_2$	$0,0$	$0,1$	$b_0$	$b_2$	$b_1$	$a_2$	$0,0$	$0,1$
$a_3$	$a_1$	$a_3$	$b_0$	$b_2$	$b_2$	$a_3$	$0,1$	$0,1$	$b_0$	$b_2$	$b_2$	$a_3$	$0,1$	$0,1$
a)			б)			в)								

Каждая полностью определенная матрица символов имеет единственную (с точностью до переобозначения

символов) неизбыточную форму, которая не содержит эквивалентных символов. Отсюда следует, что две регулярные двусторонние одномерные системы эквивалентны тогда и только тогда, когда их неизбыточные матрицы символов и выходов совпадают с точностью до переобозначения символов [Л. 27]. В общем случае эквивалент-

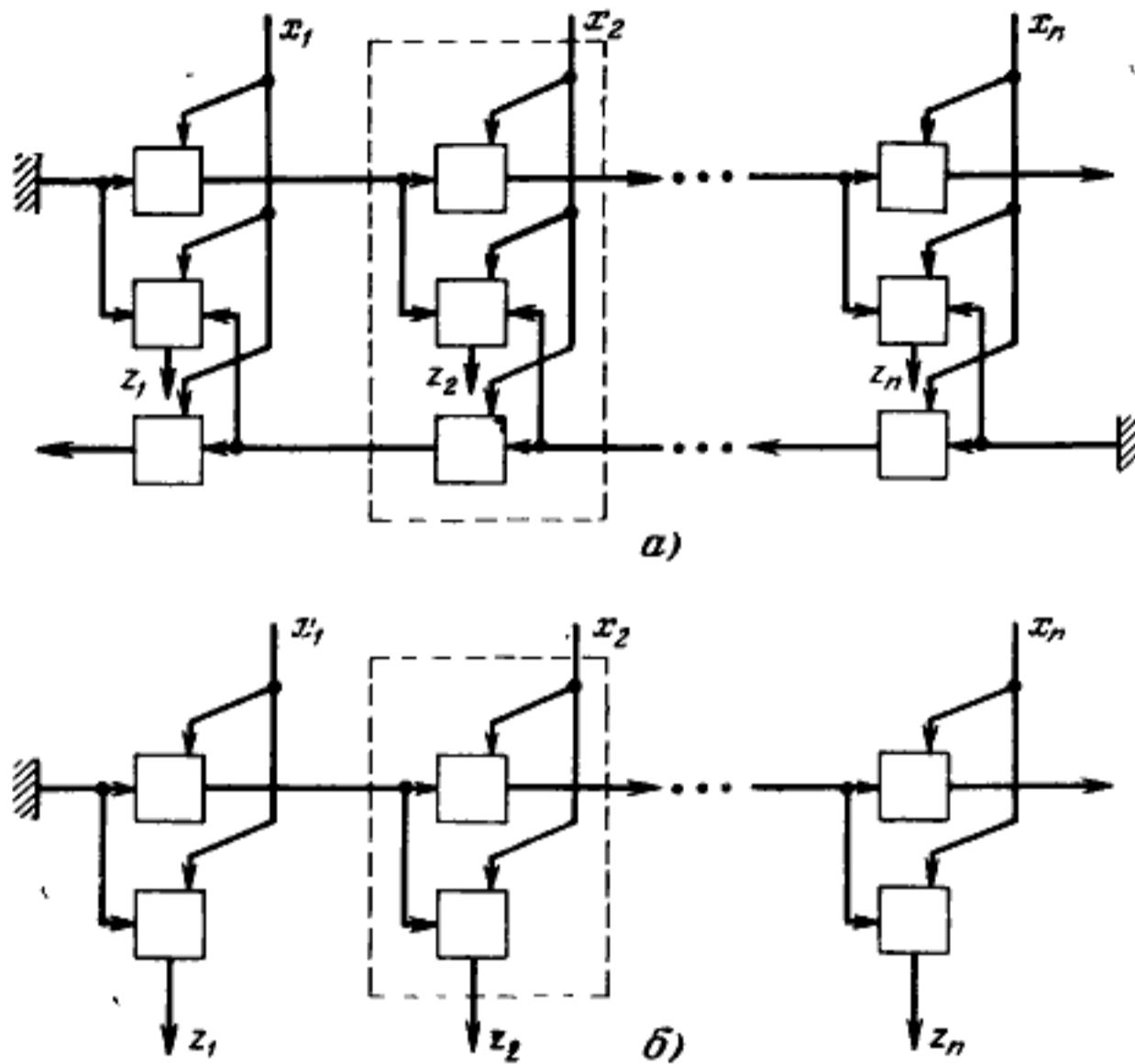


Рис. 3-3. Двусторонняя система.

*а* — каноническое представление, *б* — эквивалентная ей одновходная односторонняя система.

ными могут оказаться двусторонние системы с разным числом каналов связи. Подтверждением этому является рассматриваемая далее задача 1.

Очевидно, что неизбыточные левая и правая матрицы символов совместно с матрицей выходов представляют собой каноническое описание системы. Каждую из матриц символов можно рассматривать как представление односторонней системы из комбинационных элементов. Таким образом, двустороннюю регулярную сеть можно изобразить в каноническом виде (рис. 3-3,*а*).

**Функциональные возможности.** Функциональные возможности многовыходной двусторонней сети, естественно, выше, чем односторонней, однако если ограничиться рассмотрением одновыходных систем, эти возможности совпадают. Действительно, выход крайнего элемента сети справа на рис. 3-3,*a* зависит лишь от значений наборов внешних входов слева от него, собственного внешнего входа и правого граничного условия (последнее можно внести в структуру элемента). Тогда одновыходной системе рис. 3-3,*a* эквивалентна система на рис. 3-3,*b*.

**Синтез.** Общий подход к синтезу регулярных двусторонних систем состоит в том, чтобы разбить такую систему на две односторонние в соответствии с рис. 3-3,*a* и затем синтезировать последние и выходной преобразователь элемента. Поэтому задачу желательно переформулировать так, чтобы можно было перейти к решению двух задач синтеза односторонних сетей. Рассмотрим пример.

**Задача 1 [Л. 27].** Построить систему, внешних выход элемента которой равен 1, если он относится к блоку из четырех или более элементов, внешние входы которых равны 1, и равен 0 в противном случае.

Задачу можно решить, зная число элементов слева и справа от данного элемента с внешними входами, равными 1. Ясно, что можно ограничиться рассмотрением только трех элементов с каждой стороны. Заметим, что все наборы значений внешних входов слева (или справа) от элемента можно разбить на четыре класса  $a_j(b_j)$ :

	$a_3$	$b_3$
0	0	0
1	0	0
0	1	0
1	1	0
0	0	1
1	0	1
0	1	1
1	1	1

Нетрудно получить матрицы левых ( $a$ ) и правых ( $b$ ) символов и выходов ( $v$ ), по которым можно построить

элемент сети в канонической форме (т. е. в соответствии с рис. 3-3,а).

	$x=0$	$x=1$	$x=0$	$x=1$			$b_0$	$b_1$	$b_2$	$b_3$
$a_0$	$a_0$	$a_1$	$b_0$	$b_1$	$b_0$	$a_0$	00	00	00	01
$a_1$	$a_0$	$a_1$	$b_0$	$b_2$	$b_1$	$a_1$	00	00	01	01
$a_2$	$a_0$	$a_3$	$b_0$	$b_3$	$b_2$	$a_2$	00	01	01	01
$a_3$	$a_0$	$a_3$	$b_0$	$b_3$	$b_3$	$a_3$	01	01	01	01

а)

б)

в)

Полное описание повторения системы для нашего случая содержится в таблице г.

	$x=0$				$x=1$				
	$\beta_0$	$\beta_1$	$\beta_2$	$\beta_3$		$\beta_0$	$\beta_1$	$\beta_2$	$\beta_3$
$\alpha_0$	$\alpha_0\beta_00$	$\alpha_0\beta_00$	$\alpha_0\beta_00$	$\alpha_0\beta_00$		$\alpha_1\beta_10^*$	$\alpha_1\beta_20^*$	$\alpha_1\beta_30$	$\alpha_1\beta_31$
$\alpha_1$	$\alpha_0\beta_00$	$\alpha_0\beta_00$	$\alpha_0\beta_00$	$\alpha_0\beta_00$		$\alpha_2\beta_10$	$\alpha_2\beta_20$	$\alpha_2\beta_31$	$\alpha_3\beta_31$
$\alpha_2$	$\alpha_0\beta_00$	$\alpha_0\beta_00$	$\alpha_0\beta_00$	$\alpha_0\beta_00$		$\alpha_3\beta_10$	$\alpha_3\beta_21$	$\alpha_3\beta_31$	$\alpha_3\beta_31$
$\alpha_3$	$\alpha_0\beta_00$	$\alpha_0\beta_00$	$\alpha_0\beta_00$	$\alpha_0\beta_00$		$\alpha_3\beta_11$	$\alpha_3\beta_21^*$	$\alpha_3\beta_31^*$	$\alpha_3\beta_31^*$

г)

Боковые сигналы, распространяющиеся слева направо (справа налево), обозначены в ней буквами  $\alpha_i(\beta_i)$ . Им взаимооднозначно соответствуют левые (правые) символы  $a_i(b_i)$ . Таблицы такого типа, представляющие собой некоторое видоизменение модифицированных таблиц переходов, приспособленные для описания поведения элемента двусторонней сети, будет удобно называть в дальнейшем *канонической матрицей* элемента. Подчеркнем еще раз, что под термином «каноническая» мы понимаем лишь такую матрицу элемента системы, которая может быть представлена в виде рис. 3-3,а.

Заметим, что матрицы символов составляют абстрактное описание работы системы, в то время как каноническая матрица элемента — специфическое описание поведения элемента.

Спроектированные таким образом системы являются регулярными. При переходе к форме, отличной от канонической, иногда возможна некоторая минимизация чис-

ла состояний элемента, а следовательно, и числа боковых сигналов. Однако минимизация должна проводиться так, чтобы свойство регулярности системы сохранялось.

Напомним, что полное состояние элемента определяется значениями левого символа, правого символа и внешнего входа. Будем говорить, что два состояния *соизбыточны* тогда и только тогда, когда присвоение значений боковых и внешнего выходов, соответствующих одному состоянию, другому состоянию и наоборот, приводит к системе, эквивалентной исходной. Заметим, что соизбыточные состояния не являются эквивалентными, так как им могут соответствовать различные значения боковых выходов.

В простейшем случае два соизбыточных состояния имеют одинаковые значения внешнего входа, одинаковые правые (левые) символы, но различные левые (правые) символы. Пусть состояния  $(a_i, b_j, x_k)$  и  $(a_l, b_j, x_k)$  имеют различные левые символы. Обозначим через  $a'_i$  и  $a'_l$  символы левой матрицы символов в столбце  $x_k$  и строках  $a_i$  и  $a_l$  соответственно. Пусть  $L(b_j)$  — множество пар, составленных из правых символов и значений внешнего входа, обозначающих в правой матрице символы соответственно строки и столбцы, на пересечении которых находится символ  $b_j$ . Тогда состояния  $(a_i, b_j, x_k)$  и  $(a_l, b_j, x_k)$  не соизбыточны, если:

- 1) им соответствуют различные значения внешнего выхода;
- 2) в множестве  $L(b_j)$  существует некоторая пара  $(b_*, x_*)$ , для которой ранее уже показано, что состояния  $(a_i, b_*, x_*)$  и  $(a'_l, b_*, x_*)$  не соизбыточны.

Любую пару состояний, не удовлетворяющую этим условиям, будем называть соизбыточной «слева».

Нетрудно доказать справедливость отношения транзитивности для состояний, соизбыточных «слева» [Л. 27]. Это отношение означает, что если состояния  $(a_i, b_j, x_k)$  и  $(a_i, b_j, x_k)$  соизбыточны и также соизбыточны состояния  $(a_l, b_j, x_k)$  и  $(a_m, b_j, x_k)$ , то состояния  $(a_i, b_j, x_k)$  и  $(a_m, b_j, x_k)$  соизбыточны «слева».

*Пример.* Возвратимся к задаче 1. В таблице 2 представлены матрицы символов и каноническая матрица элемента. В этой матрице боковые сигналы  $a_i$  и  $b_j$  поставлены во взаимооднозначное соответствие символам  $a_i$  и  $b_j$ . Элементами матрицы являются наборы значе-

ний выходных сигналов. Будем искать состояния, соизбыточные слева, используя условия 1 и 2. Из таблицы  $\varrho$  видно, что любые два состояния, имеющие общий правый символ и значение внешнего входа 0, соизбыточны. Наборы значений выходов для соизбыточных состояний обведены рамками.

Далее рассмотрим состояния  $(\alpha_0, \beta_0, 1)$  и  $(\alpha_1, \beta_0, 1)$ . Находим:

$$L(b_0) = \{(b_0, 0), (b_1, 0), (b_2, 0), (b_3, 0)\},$$

$$a'_0 = a_1, a'_1 = a_2.$$

Таким образом, состояния  $(\alpha_0, \beta_0, 1)$  и  $(\alpha_1, \beta_0, 1)$  соизбыточны, если соизбыточны следующие пары состояний:

$$(\alpha_1, \beta_0, 0) \text{ и } (\alpha_2, \beta_0, 0);$$

$$(\alpha_1, \beta_1, 0) \text{ и } (\alpha_2, \beta_1, 0);$$

$$(\alpha_1, \beta_2, 0) \text{ и } (\alpha_2, \beta_2, 0);$$

$$(\alpha_1, \beta_3, 0) \text{ и } (\alpha_2, \beta_3, 0).$$

Соизбыточность этих пар состояний уже установлена, поэтому состояния  $(\alpha_0, \beta_0, 1)$  и  $(\alpha_1, \beta_0, 1)$  также соизбыточны, так что им в тех же клетках, которым соответствуют выходы  $(\alpha_1, \beta_1, 0)$  и  $(\alpha_2, \beta_1, 0)$ , можно приписать любой из этих выходов. Аналогично соизбыточность состояний  $(\alpha_1, \beta_2, 1)$  и  $(\alpha_2, \beta_2, 1)$  вытекает из соизбыточности состояний  $(\alpha_2, \beta_1, 1)$  и  $(\alpha_3, \beta_1, 1)$ , так как  $a'_1 = a_2, a'_2 = a_3$  и  $L(b_2) = (b_1, 1)$ . В свою очередь состояния  $(\alpha_2, \beta_1, 1)$  и  $(\alpha_3, \beta_1, 1)$  соизбыточны, так как их проверка приводит к тривиальному случаю:  $(\alpha_3, \beta_1, 1)$  и  $(\alpha_3, \beta_1, 1)$  соизбыточны. Подобным образом может быть установлена соизбыточность остальных групп состояний, выделенных в табл.  $\varrho$ .

Заменив в таблице  $\varrho$  выходы, соответствующие соизбыточным состояниям, на выходы, отмеченные звездочками, заметим, что правые выходные сигналы  $a_i$  зависят от левых и правых входных сигналов, а левые выходные сигналы  $\beta_i$  не зависят от левых входных сигналов, что приводит к структуре элемента на рис. 3-3. Полученная таким образом система является регулярной.

В результате каноническая матрица элемента может быть сокращена до матрицы

	$x=0$				$x=1$			
	$\beta_0$	$\beta_1$	$\beta_2$	$\beta_3$	$\beta_0$	$\beta_1$	$\beta_2$	$\beta_3$
$\alpha_0$	$\alpha_0\beta_00$	$\alpha_0\beta_00$	$\alpha_0\beta_00$	$\alpha_0\beta_00$	$\alpha_1\beta_10$	$\alpha_1\beta_20$	$\alpha_1\beta_30$	$\alpha_3\beta_31$
$\alpha_1$	$\alpha_0\beta_00$	$\alpha_0\beta_00$	$\alpha_0\beta_00$	$\alpha_0\beta_00$	$\alpha_1\beta_10$	$\alpha_1\beta_20$	$\alpha_1\beta_30$	$\alpha_3\beta_31$
$\alpha_3$	$\alpha_0\beta_00$	$\alpha_0\beta_00$	$\alpha_0\beta_00$	$\alpha_0\beta_00$	$\alpha_3\beta_11$	$\alpha_3\beta_21$	$\alpha_3\beta_31$	$\alpha_3\beta_31$

d)

Заметим, что при сокращении матрицы устраниется только межэлементный сигнал  $\alpha_2$ , а не символ  $a_2$ . Новая матрица уже не представляет систему в канонической форме, так как нет взаимооднозначного соответствия между символами и межэлементными сигналами. Новая система эквивалентна исходной. Это можно установить посредством образования матрицы символов и внешних выходов. Одновременно обнаруживается, что состояние  $(\alpha_1, \beta_2, 1)$  не является равновесным и может встречаться только в переходном периоде. Следовательно, присвоение этому состоянию произвольных значений выходов не отразится на установившихся значениях внешних выходов. Если состоянию  $(\alpha_1, \beta_2, 1)$  присвоить значения выходов  $(\alpha_1, \beta_3, 0)$ , то регулярность системы будет сохранена, так как сохраняется независимость левого выходного сигнала от левого входного сигнала, что соответствует структуре элемента на рис. 3-4. При этом сигналы  $\alpha_0$  и  $\alpha_1$  становятся эквивалентными. Получим неизбыточную матрицу (таблица e), которая эквивалентна матрице  $g^1$ .

	$x=0$				$x=1$			
	$\beta_0$	$\beta_1$	$\beta_2$	$\beta_3$	$\beta_0$	$\beta_1$	$\beta_2$	$\beta_3$
$\alpha_0$	$\alpha_0\beta_00$	$\alpha_0\beta_00$	$\alpha_0\beta_00$	$\alpha_0\beta_00$	$\alpha_0\beta_10$	$\alpha_0\beta_20$	$\alpha_0\beta_30$	$\alpha_3\beta_31$
$\alpha_3$	$\alpha_0\beta_00$	$\alpha_0\beta_00$	$\alpha_0\beta_00$	$\alpha_0\beta_00$	$\alpha_3\beta_11$	$\alpha_3\beta_21$	$\alpha_3\beta_31$	$\alpha_3\beta_31$

e)

<sup>1</sup> Этот результат можно интерпретировать следующим образом. Часть системы, сигналы в которой распространяются справа налево, независимо «подсчитывает» число внешних входов, равных 1, в каждом блоке единиц. Когда обнаружен блок из четырех или более единиц, сигнал посыпается обратно, направо. Этот сигнал проходит через все элементы, принадлежащие блоку, так что на их внешних выходах появляется 1.

Вместо состояний, соизбыточных «слева», можно было бы искать состояния, соизбыточные «справа». В данном примере из-за симметричности исходной матрицы была бы получена матрица, аналогичная матрице  $e$ .

В некоторых случаях могут быть использованы оба типа соизбыточности состояний. Однако при этом нужно проявлять максимальную осторожность, так как в результате может получиться система, боковые выходы элемента которой зависят от всех его входов. В сетях такой системы могут возникать обратные связи, что приводит к потере регулярности. В настоящее время нет алгоритмов определения, когда одновременное использование соизбыточности состояний слева и справа не приводит к нарушению регулярности, и неизвестно, существуют ли такие алгоритмы.

Неизвестны также методы склеивания состояний вида  $(a_i, b_j, x_k)$  и  $(a_i, b_m, x_k)$ , обеспечивающие сохранение регулярности.

Следует отметить, что сокращение количества межэлементных сигналов в некоторых случаях может привести к усложнению элемента, так как один из боковых выходов элемента становится зависимым от обоих боковых входов. С другой стороны, нельзя быть уверенными, что описанным выше методом могут быть найдены все реализации, соответствующие заданному каноническому описанию системы. Это говорит о том, что сокращение межэлементных сигналов нужно рассматривать не как самоцель, а в комплексе с другими проблемами.

### 3-2. Двусторонние сети и системы из элементов с памятью

**Анализ.** Двусторонняя система из элементов с памятью может быть представлена в одном из трех видов (рис. 3-5). Каждой системе одного типа можно сопоставить эквивалентную систему любого из двух других типов.

Для этого покажем, что: а) для каждой системы типа II есть эквивалентная система типа I, б) для каждой системы типа III есть

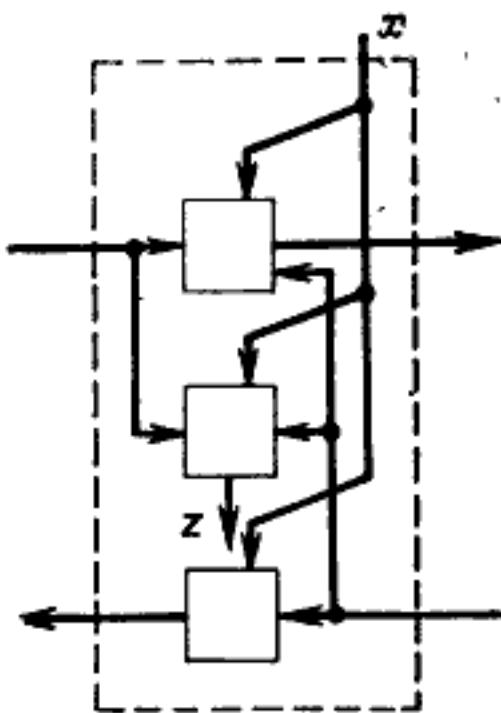


Рис. 3-4. Элемент двусторонней сети, в котором объединены левые соизбыточные состояния.

эквивалентная система типа II; в) для каждой системы типа I есть эквивалентная система типа III.

а. Множество систем типа I включает в себя множество систем типа II как частный случай, следовательно, для каждой системы типа II есть эквивалентная система типа I.

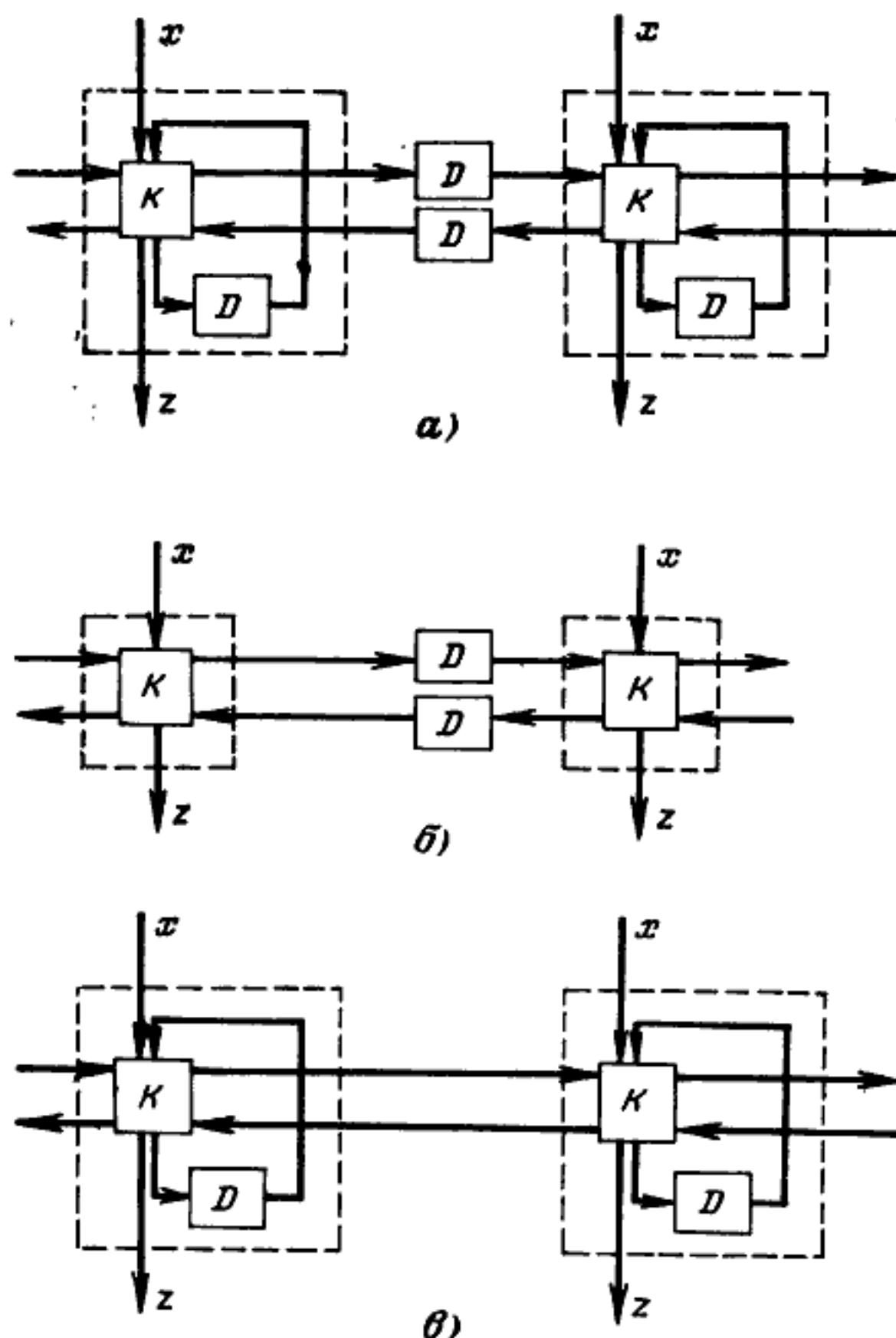


Рис. 3-5. Типы двусторонних сетей.

а — сеть типа I; б — сеть типа II; в — сеть типа III  
( $K$  — комбинационная схема,  $D$  — единичная задержка).

б. В системах типа III внутренние состояния элементов остаются постоянными до тех пор, пока боковые сигналы в сети не достигнут равновесия. Потребуем, чтобы каждая сеть системы типа III была регулярной для любого фиксированного набора внутренних состояний. Действительно, такая система может трактоваться как регулярная двусторонняя система, внешние входы которой включают в себя внутренние состояния элементов, а внешние выходы включают в себя входы во внутренние элементы задержки. Известно, что лю-

бая регулярная двусторонняя сеть может быть разделена на две односторонние сети. Сеть в такой форме с межэлементными задержками должна достигать равновесия за столько единиц времени, сколько элементов в сети. Таким образом, для того чтобы имитировать поведение системы типа III системой типа II, необходимо обеспечить возможность организации элементной памяти (петля замыкается через соседний элемент) и убедиться, что «внутренние состояния» остаются постоянными, по крайней мере  $n$  единиц времени, где  $n$  — число элементов в сети.

в. Любая система типа I может быть превращена в эквивалентную систему типа III за счет устранения межэлементных зедержек и принудительной синхронизации передачи боковых сигналов. Интервал между двумя синхронизирующими импульсами должен превышать тakt работы автомата.

Так как все три типа систем (см. рис. 3-5) эквивалентны в том смысле, что любая функция, которая может быть реализована системой одного типа, может быть также реализована системами двух других типов, то мы свободны в выборе системы, который диктуется удобством и простотой выполнения. В большинстве случаев наиболее удобен тип III.

В предыдущей главе утверждалось, что общие алгоритмы анализа односторонних одномерных систем из элементов с памятью существуют только тогда, когда фиксированы начальные состояния. Отсюда ясно, что для произвольных двусторонних одномерных систем не существует общего алгоритма анализа, т. е. проверки их на стабильность и эквивалентность.

**Функциональные возможности<sup>1</sup>.** Возникают два вопроса: 1) существует ли функция, реализуемая двусторонней одновыходной системой из элементов с памятью, но не реализуемая односторонней системой из элементов с памятью; 2) можно ли представить любую двустороннюю многовыходную систему из элементов с памятью в каноническом виде (при помощи двух односторонних систем из элементов с памятью с потоками справа налево и слева направо и выходных преобразователей). Ответы на них пока не получены. Ф. Хенни [Л. 27] предполагает, что ответ на второй вопрос отрицателен, откуда, в частности, следует, что функциональные возможности одновыходных двусторонних систем выше, чем односторонних. Однако примера, подтверждающего эту точку зрения, найти не удалось.

**Синтез.** Очень часто для синтеза нерегулярных двусторонних систем можно использовать тот же подход, что и при синтезе регулярных систем, а именно: разде-

<sup>1</sup> Одномерные двусторонние бесконечные сети из элементов с памятью обладают всеми возможностями машин Тьюринга [Л. 10]. Им можно поставить в соответствие языки типа «0» [Л. 7].

лить задачу на две такие части, чтобы каждая из них решалась односторонней нерегулярной системой. Направления потоков сигналов в этих системах должны быть противоположными. Каждый элемент полученной таким образом двусторонней сети должен содержать выходной преобразователь, комбинационный или с памятью (рис. 3-6).

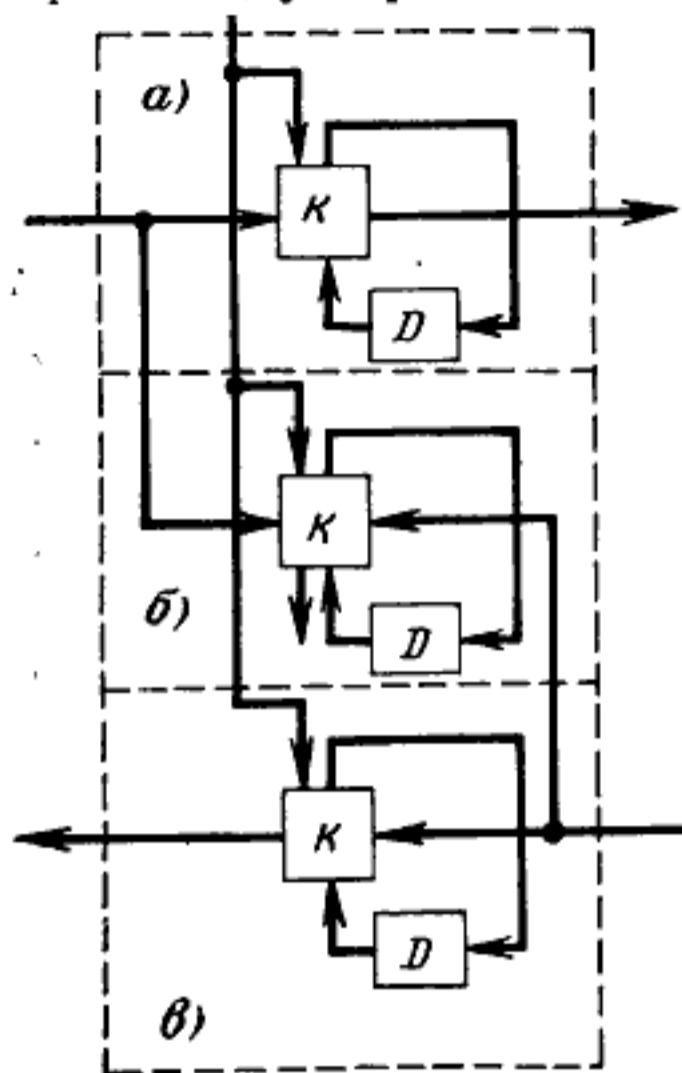


Рис. 3-6. Каноническое представление элемента двусторонней сети.

*α* — часть элемента, принадлежащая подсистеме с потоком сигналов слева направо; *β* — выходная часть элемента; *γ* — часть элемента, принадлежащая подсистеме с потоком сигналов справа налево (*K* — комбинационная схема; *D* — единичная задержка).

Можно построить две односторонние системы, которые «подсчитывают» число единиц слева и справа соответственно от данного элемента, представляя эти числа временными последовательностями межэлементных сигналов. Таблицы переходов для этих двух систем имеют вид:

$x = 0$	$x = 1$		
$A$	$B$	$A$	$B$
$S_0, A$	*	$S_1, B$	$S_0, B$
$S_1, A$	*	$S_1, A$	*

Задача 1 [Л. 27]. Построить систему, внешний выход каждого элемента которой равен 1, если в наборе внешних входов слева или справа от него больше единиц, чем нулей, и равен 0 в противном случае.

Задача может быть решена объединением двух односторонних систем задачи 2 (§ 2-1). Если внешний выход элемента первой системы  $z_1$ , а элемента второй системы  $z_2$ , то внешний выход элемента двусторонней системы  $z = z_1 \vee z_2$ .

Задача 2 [Л. 27]. Построить систему, внешний выход каждого элемента которой равен 1, если числа единиц в наборах внешних входов слева и справа от него равны, и равен 0 в противном случае.

$x = 0$	$x = 1$		
$C$	$D$	$C$	$D$
$R_0, C$	*	$R_1, D$	$R_0, D$
$R_1, C$	*	$R_1, C$	*

Если слева (справа) от данного элемента  $m$  внешних входов равны 1, то последовательность боковых сигналов, приходящих на этот элемент, имеет вид:

$$\underbrace{S_1, S_1, \dots, S_1}_{m}, S_0, S_0, \dots \quad (\underbrace{R_1, R_1, \dots, R_1}_{m}, R_0, R_0, \dots).$$

Выход преобразователя равен 1 тогда и только тогда, когда за парой сигналов  $S_1, R_1$  следует пара сигналов  $S_0, R_0$ . Выходной преобразователь задан автоматом с таблицей переходов следующего вида:

	$E$	$F$	$G$
$S_0 R_0$	$G, *$	$F, 0$	$G, 1$
$S_0 R_1$	$F, *$	$F, 0$	$G, 1$
$S_1 R_0$	$F, *$	$F, 0$	$G, 1$
$S_1 R_1$	$E, *$	$F, 0$	$G, 1$

Для понимания работы этой системы подчеркнем, что задержка боковых сигналов на каждом элементе пренебрежимо мала по сравнению с тактом работы автомата, определяемым величиной задержки (рис. 3-6).

Как уже отмечалось, представление нерегулярной двусторонней системы в виде рис. 3-6 возможно, по-видимому, не всегда. Рассмотрим пример, для которого решение в виде рис. 3-6 пока не найдено.

**Задача 3 [Л. 27].** Построить систему, внешний выход любого элемента которой равен 1, если произведение числа единиц слева от элемента на число единиц справа от него равно числу нулей слева от него, и равен 0 в противном случае.

По условию задачи необходимо найти произведение двух чисел, представленных в системе с фиксированным основанием, одно из которых поступает слева, а другое справа. Количество разрядов произведения не превышает суммы разрядов сомножителей. Поэтому трудно представить, что такая задача может быть решена двусторонней системой, представленной в каноническом виде. Однако она довольно просто решается, если допустить зависимость одной односторонней системы от другой. Обозначим через  $L_j$  и  $R_j$  число внешних входов, равных 1, слева и справа от данного элемента  $j$  соответ-

ственno. Элемент  $j$  должен вычислять произведение  $L_j R_j$ , и сравнивать его с числом нулей слева. Произведение  $L_{j+1} R_{j+1}$  может быть определено по произведению  $L_j R_j$  и значениям внешних входов  $j$ -го и  $(j+1)$ -го элементов:

$x_j$	$x_{j+1}$	$L_{j+1} R_{j+1}$
0	0	$L_j R_j$
1	0	$(L_j + 1) R_j = L_j R_j + R_j$
0	1	$L_j (R_j - 1) = L_j R_j - L_j$
1	1	$(L_j + 1) (R_j - 1) = L_j R_j + R_j - L_j - 1$

Таким образом, для нахождения нового произведения необходимо сложить не более четырех слагаемых.  $L_j$  и  $R_j$  легко представить в двоичной системе счисления с помощью двух односторонних двоичных счетчиков, из которых один передает сигналы слева направо, а другой — справа налево. При этом система с передачей сигналов справа налево независимо вычисляет  $R_j$ , а система слева направо наряду с вычислением  $L_j$  вычисляет также  $L_j R_j$  и число нулей слева от элемента  $j$ .

### 3-3. Двусторонние системы из автоматов Мура

Несколько в стороне от рассмотренных задач стоят следующие задачи, в которых строятся двусторонние системы из автоматов Мура.

**Задача 1** [Л. 4, 12, 20, 25, 45, 57, 59] (задача Дж. Майхилла о синхронизации цепи автоматов). Построить двустороннюю одномерную автономную систему из элементов с памятью, такую, что после подачи стартового сигнала на крайний элемент в момент времени  $t=0$  все элементы синхронизируются (переходят в одинаковое состояние) в некоторый момент времени  $T$ .

Формулировка этой задачи была впервые опубликована Э. Муром [Л. 45]. Задача Дж. Майхилла, поставленная в связи с необходимостью обеспечения одновременного включения всех частей самовоспроизводящейся машины, заключается в следующем. Имеется цепь стрелков, каждый из которых может обмениваться информацией только с двумя своими непосредственными соседями. Цепь состоит из конечного числа стрелков, два крайних стрелка имеют только по одному соседу. Один из крайних стрелков получает команду, после чего стрелки

должны «договориться» и одновременно произвести выстрел. Каждый стрелок, получив информацию от своих соседей, отвечает им с постоянной фиксированной задержкой, одинаковой для всех стрелков. Возникает вопрос о существовании локального алгоритма поведения отдельного стрелка (одинакового для всех стрелков, за исключением, быть может, крайних), сложность которого не зависит от числа стрелков в цепи и использование которого стрелками позволяет им произвести одновременный выстрел.

Как сообщает Э. Мур, положительный ответ на вопрос о существовании решения этой задачи был получен Дж. Мак-Карти и М. Минским, а в 1962 году Э. Гото [Л. 25] опубликовал решение с минимальным временем синхронизации  $T=2n-2$ , где  $n$  — число автоматов в цепи. Предложенный Э. Гото автомат имеет много тысяч состояний. В 1965 г. В. И. Левенштейн [Л. 12] опубликовал блестящее решение этой задачи для минимального времени синхронизации и числа внутренних состояний автомата, равного девяти. В 1966 г. А. Ваксман [Л. 59] и в 1967 г. Р. Балзер [Л. 20] опубликовали сходные решения, в которых число внутренних состояний автомата было соответственно 16 и 8. Приведенное здесь решение, являющееся модификацией решения В. И. Левенштейна, опубликовано в [Л. 4, 57].

Среди особенностей этой задачи следует отметить следующие:

1. Существенно то, что каждый элемент срабатывает с задержкой (одинаковой для всех элементов). Ранее мы также предполагали, что все элементы, в том числе комбинационные, обладают задержками. Однако здесь сама формулировка задачи требует учета задержек.

2. При решении этой задачи не удалось разбить ее на две части, каждая из которых реализовывалась бы односторонней системой.

3. Рассматриваемая система является автономной.

Система, реализующая условия задачи, состоит из автоматов Мура, внутренние состояния автоматов одновременно являются распространяющимися по цепи сигналами. В основу решения положена идея деления отрезка любой сети системы пополам. В результате деления отрезка пополам автомат (или два автомата), находящийся в середине отрезка, переходит в состояние готовности, предшествующее синхронному состоянию.

Автомат переходит в синхронное состояние, если он сам и два его соседа находятся в состоянии готовности.

Обратимся к рис. 3-7. Первое деление цепи автоматов пополам производится следующим образом. Стартовый сигнал переводит крайний автомат в состояние готовности, и от него по сети начинают распространяться два сигнала:  $p_1$  и  $p_3$ .

Первый сигнал распространяется со скоростью единица, а второй — со скоростью  $1/3$  (сигнал распространяется по цепи со скоростью  $1/m$ , если до перехода к следующему автомату он сохраняется  $m$  единиц времени предыдущим). Сигнал  $p_1$  доходит до конца сети, переводит крайний автомат в состояние готовности и движется обратно с той же скоростью. Встреча отраженного сигнала с сигналом  $p_3$  происходит в середине сети, и соответствующий автомат (или два автомата в случае четности числа элементов сети) переходит в состояние готовности. Если отраженный сигнал будет продолжать распространяться по сети со скоростью единица, а первый автомат испустит в начальный момент сигнал со скоростью распространения  $1/7$  (сигнал  $p_7$ ), то эти сигналы встретятся на расстоянии  $1/4$  от начала цепи. Далее, если каждый автомат, находящийся в состоянии готовности, будет испускать последовательность сигналов, распространяющихся со скоростями  $1/(2^{m+1}-1)$ , и в точках встречи сигналов автоматы будут переходить в состояние готовности, то в сети, как это видно из рис. 3-7, будет осуществляться последовательное деление пополам возникающих отрезков.

Рис. 3-7. Картина распространения сигналов в цепи автоматов при подаче стартового сигнала на крайний левый автомат (задача 1).

Основная трудность заключается в необходимости организации испускания автоматами с конечными числами состояний последовательности сигналов со скоростями распространения  $1/(2^{m+1}-1)$ .

Вообще говоря, числа состояний цепи хватает для этого, так как цепь имеет  $k^n$  состояний, если в цепи  $n$  автоматов и каждый из них имеет  $k \geq 2$  состояний. Значения  $2^{m+1}-1$  лежат в пределах  $1 \leq 2^{m+1}-1 < n-1$ . Если не накладывать ограничений на функцию переходов цепи, как одного автомата, то для организации счета до  $2^{m+1}-1$  нужно  $n-1$  состояний. Однако для цепи не все переходы из состояния в состояние возможны, так как в ней жестко заданы связи между составляющими цепь автоматами.

Задача состоит в том, чтобы при заданной структуре обеспечить выполнение алгоритма последовательного деления цепи пополам, связанного с испусканием сигналов со скоростями  $1/(2^{m+1}-1)$ .

На рис. 3-8 приведен пример синхронизации цепи из 26 таких автоматов. Правила смены состояний автомата приведены в табл. 3-1. Столбец «Правило» этой таблицы разделен на три части. В центре записывается состояние автомата, слева и справа — состояния левого и правого соседних автоматов соответственно. Подобные обозначения приняты и далее в этом параграфе.

Так же как в работе [Л. 12], введем отношение противоположности внутренних состояний и функций переходов автоматов цепи. Будем считать противоположными состояния  $\vec{D}$  и  $\tilde{D}$ .

Остальные состояния являются противоположными сами себе. Противоположными значениями функции переходов автоматов будут

$$F(x_1, \dots, x_n) \text{ и } F^*(x_{n-1}, \dots, x_1),$$

где  $x_{n-1}^*$  и  $x_n$  — противоположные состояния автоматов.

Для  $j$ -го автомата цепи выполняется соотношение

$$F_j(x_{j-1}, x_j, x_{j+1}) = F^*_{j+1}(x_{j+1}^*, x_j^*, x_{j-1}).$$

Поэтому в табл. 3-1 функция переходов задана лишь на половине наборов. Значения функции для остальных наборов могут быть получены из предыдущего соотношения.

В приведенном решении автомат имеет 8 внутренних состояний, а время синхронизации равно  $T=2n-2$ , где  $n$  — число автоматов в сети.

У читателя может возникнуть вопрос: что такое стартовый сигнал? Из сказанного ясно, что подача стартового сигнала на левый крайний элемент сети эквивалентна начальной установке сети в состояние  $RAA\dots A$ .

$t \setminus i$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	
0	●																										
1	●	○																									
2	●	△	○																								
3	●	□	△	○																							
4	●	□		△	○																						
5	●	□	△	○		△	○																				
6	●	△	○			△	○																				
7	●	□	○			△	○																				
8	●	□	○			△	○																				
9	●	□				△	○																				
10	●	□				△	○																				
11	●	□				△	○																				
12	●	□				△	○																				
13	●	□	△	○		○	○																				
14	●	△	○			○	○																				
15	●	□	○			○	○																				
16	●	□	○			○	○																				
17	●	□	○			○	○																				
18	●	□	○			○	○																				
19	●	□	○			○	○																				
20	●	□	○	△		○	○	△																			
21	●	□		○		○	○	○																			
22	●	□		○		○	○	○																			
23	●	□		○		○	○	○																			
24	●	□		○		○	○	○																			
25	●	□		○		○	○	○																			
26	●	□		○		○	○	○																			
27	●	□		○		○	○	○																			
28	●	□		○		○	○	○																			
29	●	□		○	△	○	○	○																			
30	●	△	○			○	○	○																			
31	●	□	○			○	○	○																			
32	●	□	○			○	○	○																			
33	●	□	○			○	○	○																			
34	●	□	○			○	○	○																			
35	●	□	○			○	○	○																			
36	●	□	○			○	○	○																			
37	●	□	○			○	○	○																			
38	●	□	○			○	○	○																			
39	●	□	○			○	○	○																			
40	●	□	○			○	○	○	△																		
41	●	□	○			○	○	○	△																		
42	●	□	○			○	○	○	△																		
43	●	□	○			○	○	○	△																		
44	●	□	○	△				●	△																		
45	●	□		○	●	●	△	○	○																		
46	●	□		○	●	●	△	●	○	○																	
47	●	□		●	△	○	●	●	○	●																	
48	●	□		●	●	○	●	●	○	●																	
49	●	□		●	●	●	●	●	●	●																	
50	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	

обозначение состояний

A	B	→	→	E <sub>1</sub>	E <sub>2</sub>	R	F
●	▷	◁	○	□	●	■	

Рис. 3-8. Пример синхронизации цепи из 26 автоматов (задача 1).

Таблица 3-1

Номер	Правило			Условие
1	$x$	$A$	$z$	1) $x \in \{A, \vec{D}, E_1\}; z \in \{\vec{D}, E_1\}$
		$A$		2) $x \in \{A, E_2\}; z \in \{A, E_2\}$
2	$x$	$\vec{D}$	$z$	1) $x \in \{A, B, \vec{D}, E_1\}; z = A$
		$A$		2) $x \in \{A, E_2\}; z \in \{B, \vec{D}\}$
3	$\vec{D}$	$E_1$	$z$	$z \in \{A, E_2\}$
		$A$		
4	$x$	$A$	$z$	$x \in \{B, R\}; z \in \{A, \vec{D}, E_1, E_2\}$
		$B$		
5	$x$	$\vec{D}$	$R$	$x \in \{A, E_2\}$
		$R$		
6	$\vec{D}$	$A$	$z$	$z \in \{A, E_1, E_2\}$
		$\vec{D}$		
7	$x$	$R$	$z$	1) $x = \vec{D}; z \in \{A, E_2\}$
		$\vec{D}$		2) $x \in \{A, E_1\}; z \in \{\vec{D}, R\}$
8	$x$	$\vec{D}$	$z$	1) $x \in \{\vec{D}, R\}; z = A$
		$\vec{D}$		2) $x = \vec{D}; z = \vec{D}$
9	$x$	$\vec{D}$	$\vec{D}$	$x \in \{\vec{D}, R\}$
		$\vec{D}$		
10	$x$	$E_2$	$\vec{D}$	$x \in \{A, R\}$
		$\vec{D}$		
11	$A$	$\vec{D}$	$E_2$	
		$E_1$		

Номер	Правило			Условие
12	$A$	$E_1$	$z$	$z \in \{A, \vec{D}, E_2\}$
		$E_1$		
13	$x$	$\vec{D}$	$z$	1) $x \in \{R, \overleftarrow{D}, E_1\}; z = R$ 2) $x = A; z = E_1$
		$E_2$		
14	$x$	$E_2$	$z$	1) $x \in \{A, \overleftarrow{D}, E_1\}; z = R$ 2) $x = A; z = A$
		$E_2$		
15	$A$	$y$	$R$	$y \in \{A, \vec{D}, E_2\}$
		$R$		
16	$B$	$y$	$z$	1) $y = E_1; z \in \{A, \vec{D}, E_2\}$ 2) $y = A; z = B$ 3) $y = E_2; z \in \{A, R\}$
		$R$		
17	$x$	$B$	$z$	1) $x \in \{\vec{D}, R\}; z = E_2$ 2) $x = \overleftarrow{D}; z = E_1$
		$R$		
18	$\vec{D}$	$\vec{D}$	$z$	$z \in \{E_1, E_2\}$
		$R$		
19	$x$	$R$	$z$	1) $x \in \{A, \vec{D}, \overleftarrow{D}, R\}; z \in \{A, \overleftarrow{D}\}$ 2) $x = E_2; z \in \{\vec{D}, E_2, R\}$ 3) $x = \overleftarrow{D}; z = \vec{D}$ 4) $x = B; z \in \{B, \vec{D}, R\}$
		$R$		
20	$R$	$R$	$R$	
		$F$		
21	$F$	$F$	$F$	
		$A$		

**Задача 2 [Л. 4, 56, 57].** Решить задачу 1 в предположении, что стартовый сигнал подается на произвольный элемент сети.

На рис. 3-9 изображена общая картина распространения сигналов. После подачи стартового сигнала от начального автомата  $H$  в обе стороны начинают распространяться сигналы  $p_1$  и  $p'_1$  со скоростями, равными единице. (При этом начальный автомат не переходит в состояние готовности, за исключением тех случаев, когда он является крайним.) Дойдя до краев сети, сигналы  $p_1$  и  $p'_1$  переводят крайние автоматы в состояние готовности и порождают отраженные сигналы, распространяющиеся с той же скоростью. Как было сказано выше, автомат, переходя в состояние готовности, начинает испускать последовательность сигналов, распространяющихся со скоростями  $1/(2^{m+1}-1)$ . Если бы стартовый сигнал был подан на автомат  $O$ , находящийся на ближайшем к начальному автомату конце сети, то картина распространения сигналов повторяла бы рис. 3-7 с началом в точке  $O'$ . При этом сигнал  $p'_3$ , идущий из точки  $O'$  со скоростью  $1/3$ , встретился бы с отраженным сигналом  $p''_1$  в точке  $A_1$  (в середине сети). Нетрудно видеть, что линия распространения сигнала  $p'_3$  пересекается с линией распространения отраженного сигнала  $p''_1$  в точке  $A$ , соответствующей положению начального автомата. Следовательно, для осуществления первого деления сети пополам необходимо в точке  $A$  изменить скорость распространения отраженного сигнала  $p''_1$  с единицы на  $1/3$ . Линия распространения сигнала, идущего из точки  $O'$  со скоростью  $1/(2^{m+1}-1)$ , и линия распространения сигнала, идущего из точки  $O_1$  со ско-

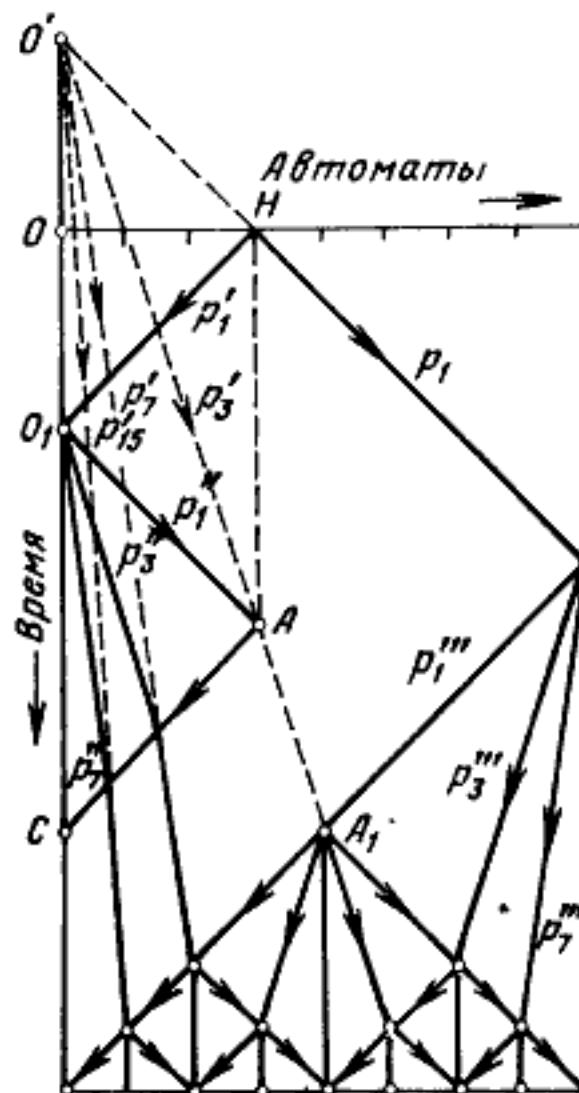


Рис. 3-9. Картина распространения сигналов в цепи автоматов при подаче стартового сигнала на произвольный элемент цепи (задача 2).

Рис. 3-9. Картина распространения сигналов в цепи автоматов при подаче стартового сигнала на произвольный элемент цепи (задача 2).

ростью  $1/(2^m-1)$ , пересекаются на прямой  $AC$ , образующей линию переключения скоростей. Для осуществления правильной последовательности делений необходимо, чтобы каждый сигнал, идущий из точки  $O_1$  со скоростью  $1/(2^m-1)$ , изменял свою скорость на  $1/(2^{m+1}-1)$ . Наклон линии переключения соответствует скорости распространения сигнала, равной единице. В остальном картина распространения сигналов на рис. 3-9 повторяет рис. 3-7. Необходимость организации переключения скоростей повлекла за собой введение двух дополнительных внутренних состояний автоматов и частичное изменение правил перехода. На рис. 3-10 приведен пример синхронизации сети из 28 автоматов при подаче стартового сигнала на десятый автомат. Правила смены состояний автомата на десять состояний приведены в табл. 3-2.

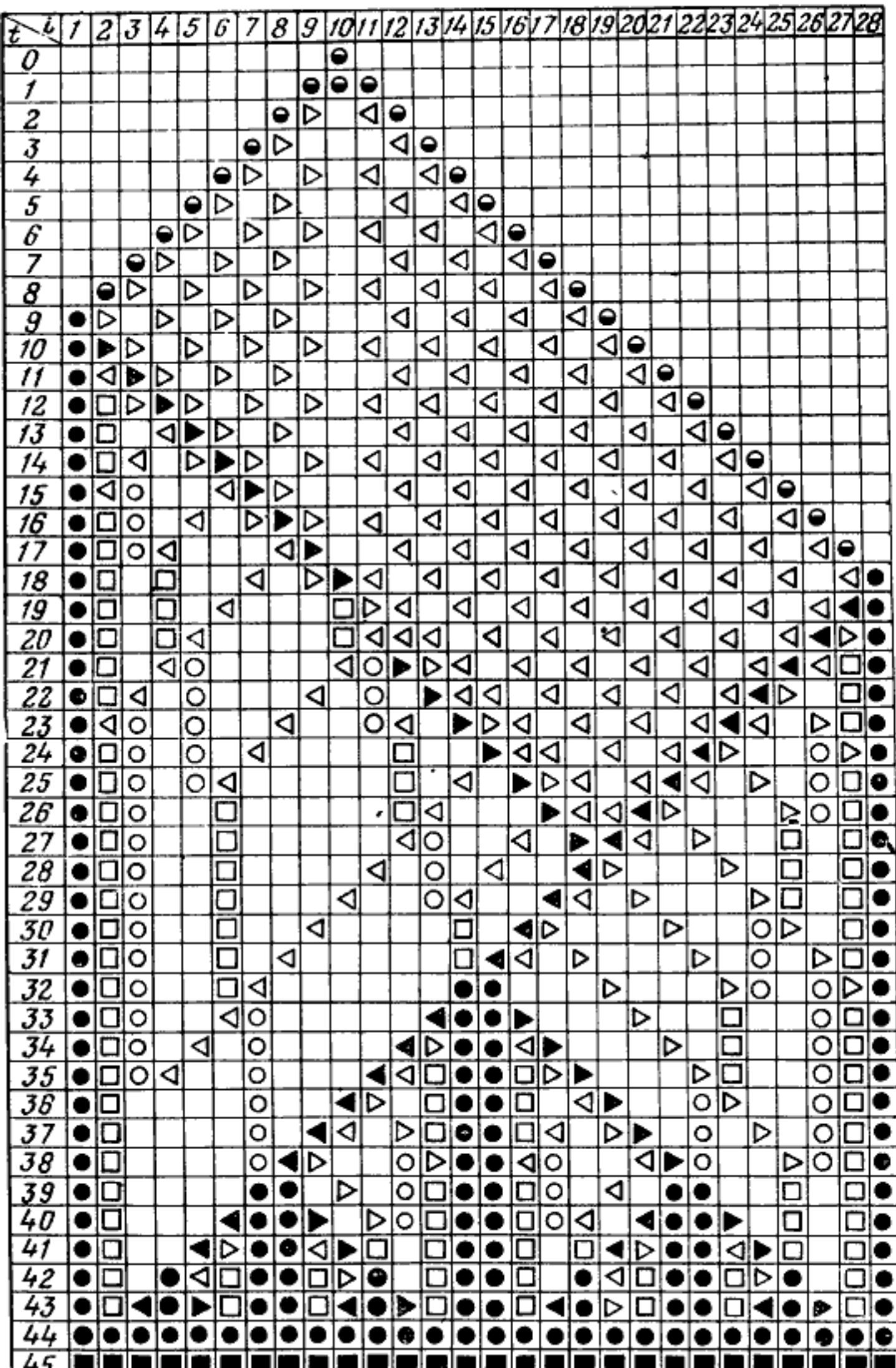
Из сравнения рис. 3-7 и 3-9 видно, что синхронизация на рис. 3-9 осуществляется быстрее ровно на то время, которое требуется для прохождения сигнала, идущего со скоростью единицы, от точки  $O$  до точки  $H$ , т. е. в случае подачи стартового сигнала на произвольный автомат цепи время синхронизации  $T = 2n - 2 - a_{\min}$ , где  $a_{\min}$  — расстояние от начального автомата до ближайшего края цепи.

**Задача 3** [Л. 3, 56, 57] («задача о голосовании»). Построить систему, реализующую  $k$ -значную функцию голосования.

Сначала определим  $k$ -значную функцию голосования. Будем говорить, что значение  $a_j (1 \leq j \leq k)$  имеет  $i$  голосов, если  $i$  переменных принимают значение  $a_j$ . Значение функции голосования совпадает со значением переменной  $a_j$ , если она имеет наибольшее число голосов. При равенстве голосов у двух значений предпочтение отдается значению с наибольшим индексом. Нетрудно заметить, что при  $k=2$  функция голосования совпадает с мажоритарной функцией (см. задачу 2 § 2-1).

Значение внешних входов элементов системы в данном случае является не двузначным, что имеет место в предыдущих задачах, а  $k$ -значным. Это можно интерпретировать как наличие  $\lceil \log_2 k \rceil$  внешних входов. Иначе говоря, внешним входом элементов является двоичный код числа  $a_j (1 \leq j \leq k)$ .

Блок-схема устройства приведена на рис. 3-11. Для простоты изложения рассматриваются автоматы Мура,



Обозначение состояний

A	B	C	$\bar{C}$	D	$\bar{D}$	E <sub>1</sub>	E <sub>2</sub>	R	F
●	►	◀	▷	◁	○	□	•	■	

Рис. 3-10. Пример синхронизации цепи из 28 автоматов при подаче стартового сигнала на 10-й автомат (задача 2).

Таблица 3-2

Номер	Правило			Условие
1	$x$	$A$	$z$	1) $x = A; z \in \{A, \vec{D}, E_1, E_2\}$ 2) $x \in \{\vec{C}, E_1\}; z = \vec{D}$ 3) $x = \vec{D}; z = \vec{D}$ 4) $x = E_1; z = E_1$ 5) $x = E_2; z = E_2$
2	$x$	$\vec{C}$	$z$	1) $x = A; z \in \{\vec{D}, \vec{D}\}$ 2) $x \in \{E_1, E_2\}; z = \vec{D}$
3	$x$	$\vec{D}$	$z$	1) $x \in \{B, \vec{C}, E_1\}; z = A$ 2) $x = E_2; z = \vec{C}$ 3) $x = A; z \in \{A, \vec{C}\}$
4	$B$	$B$	$B$	
5	$x$	$E_1$	$\vec{D}$	$x \in \{A, E_2\}$
6	$A$	$y$	$z$	1) $y = B; z = A$ 2) $y = A; z = B$
7	$x$	$A$	$z$	$x \in \{\vec{C}, R\}; z \in \{A, \vec{D}, E_1, E_2\}$
8	$\vec{C}$	$\vec{C}$	$z$	$z \in \{A, E_1, E_2\}$
9	$x$	$\vec{D}$	$z$	$x \in \{\vec{C}, R\}; z \in \{A, \vec{C}, \vec{D}, \vec{D}\}$
10	$x$	$\vec{D}$	$z$	1) $x \in \{\vec{C}, \vec{D}\}; z = \vec{D}$ 2) $x = R; z = A$

Продолжение табл. 3-2

Номер	Правило			Условие
11	$R$	$B$	$B$	
		$\vec{C}$		
12	$x$	$A$	$z$	1) $x = \vec{C}; z \in \{\vec{C}, \overleftarrow{C}\}$ 2) $x = B; z = B$
		$R$		
13	$x$	$\vec{C}$	$z$	1) $x \in [R, \vec{D}]; z = E_2$ 2) $x = \overleftarrow{D}; z = E_1$
		$R$		
14	$x$	$R$	$z$	1) $x = \overleftarrow{C}; z = \vec{C}$ 2) $x = B; z = B$ 3) $x = R; z \in \{A, \vec{C}, \overleftarrow{D}, E_2\}$ 4) $x \in \{A, \overleftarrow{D}\}; z \in \{A, \vec{D}\}$ 5) $x = \vec{D}; z \in \{A, \vec{D}, \overleftarrow{D}\}$ 6) $x = E_2; z = E_2$
		$R$		
15	$x$	$y$	$z$	1) $x = \vec{C}; y = \vec{D}; z \in \{E_1, E_2\}$ 2) $x = R; y \in \{B, \vec{D}, \overleftarrow{D}, E_2\}; z = R$
		$R$		
16	$\vec{C}$	$y$	$z$	1) $y = E_1; z \in \{A, \vec{D}, E_2\}$ 2) $y = E_2; z \in \{A, R\}$
		$R$		
17	$A$	$B$	$z$	$z \in \{B, \vec{D}\}$
		$\vec{D}$		
18	$x$	$A$	$z$	1) $x \in \{\vec{C}, \overleftarrow{D}\}; z \in \{A, E_1, E_2\}$ 2) $x = \vec{D}; z = \vec{D}$
		$\vec{D}$		
19	$x$	$\vec{C}$	$z$	$x \in \{A, \vec{C}, \overleftarrow{D}, E_1\}; z \in \{R, \vec{D}\}$
		$\overleftarrow{D}$		

Продолжение табл. 3-

Номер	Правило			Условие
20	$\overleftarrow{D}$	$\vec{C}$	$z$	$z \in \{A, \overleftarrow{C}, \vec{D}, E_2\}$
21	$x$	$\vec{D}$	$\overleftarrow{D}$	$x \in \{A, B\}$
22	$x$	$\overleftarrow{D}$	$z$	1) $x \in \{\vec{C}, \overleftarrow{D}\}; z \in \{A, B\}$ 2) $x = \vec{D}; z \in \{E_1, E_2\}$
23	$\vec{D}$	$E_2$	$z$	$z \in \{A, R\}$
24	$\overleftarrow{D}$	$\vec{C}$	$\overleftarrow{D}$	
25	$x$	$\vec{D}$	$E_2$	1) $x \in \{A, \vec{D}\}$
26	$x$	$E_1$	$z$	1) $x = A; z \in \{A, \vec{D}, E_2\}$ 2) $x = \overleftarrow{C} z = \vec{D}$
27	$x$	$\vec{D}$	$E_1$	$x \in \{A, \vec{D}\}$
28	$x$	$E_2$	$z$	1) $x = A; z \in \{A, \vec{C}, \vec{D}, R\}$ 2) $x = R; z \in \{E_1, \vec{D}\}$
29	$x$	$\vec{C}$	$\overleftarrow{D}$	$x \in \{R, \vec{D}\}$
30	$x$	$\vec{D}$	$R$	$x \in \{\vec{C}, E_1\}$
31	$R$	$R$	$R$	
32	$F$	$F$	$F$	
		$A$		

выходами которых являются номера их внутренних состояний, и каждый автомат осуществляет однотактную задержку. Опишем общий алгоритм взаимодействия автоматов в сети. В связи с тем что сложность автомата не должна зависеть от длины сети (числа переменных), мы лишены возможности осуществить непосредственный подсчет числа переменных, принимающих одинаковые значения (число голосов  $j$ -го значения). С другой стороны, если мы образуем из равных значений переменных сплошные цепочки (голосующие коалиции),

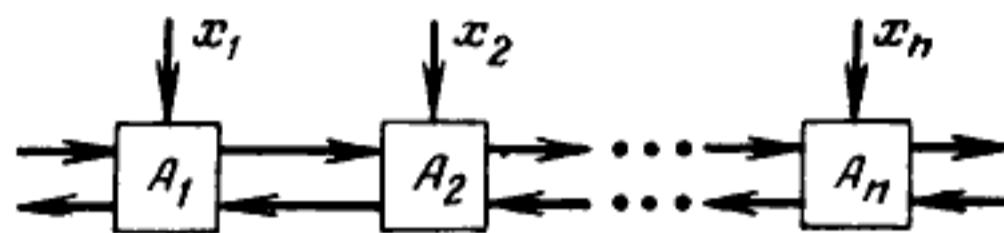


Рис. 3-11. Цепь автоматов для вычисления значения функции голосования.

то сможем сравнить времена прохождения сигналов по цепочкам автоматов, образующих голосующие коалиции. Пусть работа схемы состоит из трех фаз. Первая фаза — занесение значений переменных в автоматы (фаза считывания); вторая фаза — образование голосующих коалиций (фаза перераспределения); третья фаза — попарное сравнение числа членов в голосующих коалициях (фаза сравнения). Рассмотрим организованную таким образом систему.

Пусть в начальный момент времени все автоматы находятся в начальном состоянии  $H$ . Кроме того, множество внутренних состояний автомата имеет подмножество состояний  $E_1 = \{E_{11}, \dots, E_{1k}\}$ . Пусть левым граничным сигналом является сигнал подмножества  $E_1$ . Пусть начальным автоматом является крайний левый автомат, тогда переход из начального состояния  $H$  в состояние подмножества  $E_1$  определим следующим образом: если автомат  $A_j$  и его правый сосед находятся в состоянии  $H$ , а левый сосед — в состояние  $E_1$ , то автомат  $A_j$  переходит в состояние  $E_{1i}$ , где  $i$  — значение переменной  $x_j$ . Так осуществляется считывание переменных в цепь автоматов. Далее до окончания решения задачи состояния входов, на которые поступают переменные, не влияют на поведение автоматов. На этом заканчивается фаза считывания.

Теперь рассмотрим процесс упорядочения состояний автоматов, соответствующих значениям переменных. Для того чтобы образовать голосующие коалиции, расположенные по убыванию значений переменных справа налево, необходимо осуществить обмен номерами внутренних состояний между соседними автоматами каждый раз, когда номер внутреннего состояния правого автомата меньше номера внутреннего состояния левого автомата. Работа системы иллюстрируется рис. 3-12 и табл. 3-3.

Мы будем говорить, что на автомат поступает сигнал  $E_1$ , если его сосед находится в состоянии  $E_1$ . Фронт сигнала  $E_1$  распространяется по цепи автоматов, и за ним происходит упорядочение состояний. Нетрудно заметить при этом, что в процессе упорядочения путем парного обмена номерами состояний не может возникнуть логических противоречий, при которых автомат должен обмениваться номерами внутренних состояний сразу с обоими соседями. Состояние с максимальным номером, возникнув на фронте сигнала  $E_1$ , продолжает двигаться за фронтом сигнала с той же скоростью, что и фронт, а состояние с минимальным номером, возникнув на фронте сигнала  $E_1$ , движется в противоположную сторону со скоростью один автомат за один такт. Следовательно, если сигнал  $E_1$ , дойдя до конца сети,

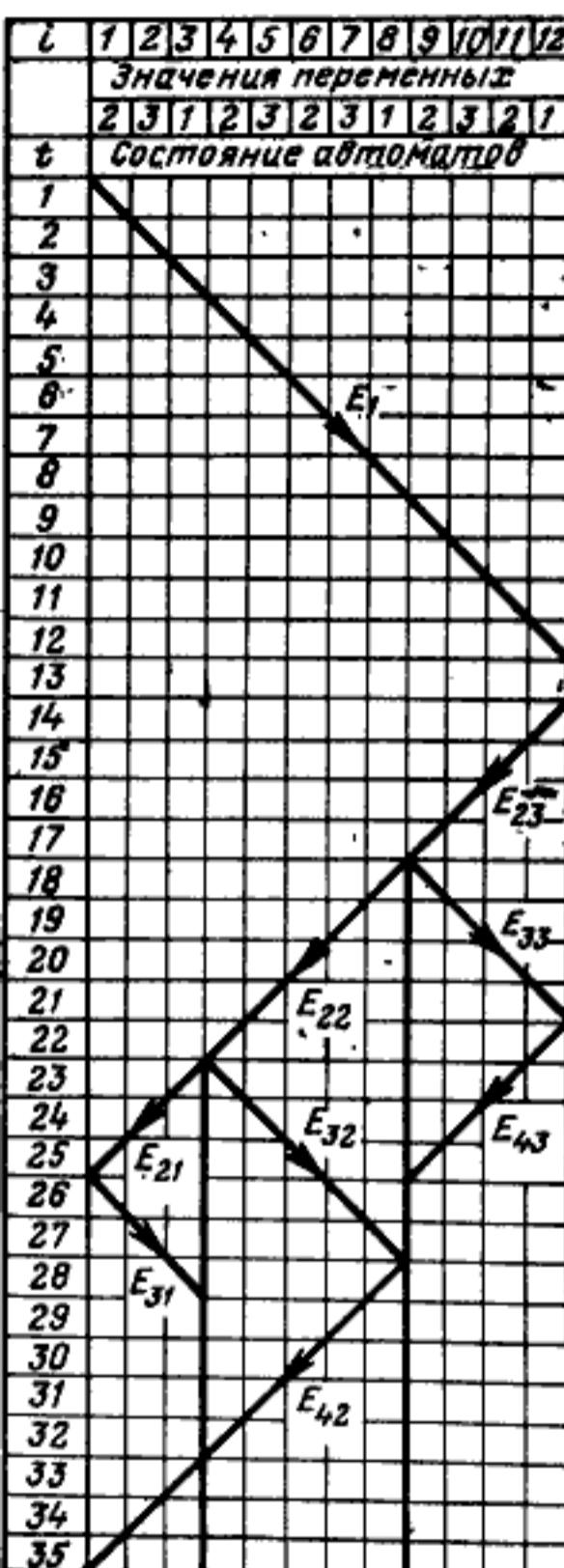


Рис. 3-12. Пример, иллюстрирующий решение задачи 3.

породит отраженный сигнал, то отраженный сигнал будет двигаться по сети автоматов с упорядоченными внутренними состояниями и может быть использован для сравнения чисел автоматов в голосующих коалициях. Сигнал  $E_1$  на краю сети порождает отраженный сигнал  $E_2$ , на фронте которого происходит переход автоматов из состояний подмножества  $E_1$  в состояния подмножества

Таблица 3-3

<i>i</i>	1	2	3	4	5	6	7	8	9	10	11	12
<i>t</i>	Значения переменных											
	2	3	1	2	3	2	3	1	2	3	2	1
1	2 <sub>1</sub>											
2	2 <sub>1</sub>	3 <sub>1</sub>										
3	2 <sub>1</sub>	3 <sub>1</sub>	1 <sub>1</sub>									
4	2 <sub>1</sub>	1 <sub>1</sub>	3 <sub>1</sub>	2 <sub>1</sub>								
5	1 <sub>1</sub>	2 <sub>1</sub>	2 <sub>1</sub>	3 <sub>1</sub>	3 <sub>1</sub>							
6	1 <sub>1</sub>	2 <sub>1</sub>	2 <sub>1</sub>	3 <sub>1</sub>	3 <sub>1</sub>	2 <sub>1</sub>						
7	1 <sub>1</sub>	2 <sub>1</sub>	2 <sub>1</sub>	3 <sub>1</sub>	2 <sub>1</sub>	3 <sub>1</sub>	3 <sub>1</sub>					
8	1 <sub>1</sub>	2 <sub>1</sub>	2 <sub>1</sub>	2 <sub>1</sub>	3 <sub>1</sub>	3 <sub>1</sub>	3 <sub>1</sub>	1 <sub>1</sub>				
9	1 <sub>1</sub>	2 <sub>1</sub>	2 <sub>1</sub>	2 <sub>1</sub>	3 <sub>1</sub>	3 <sub>1</sub>	1 <sub>1</sub>	3 <sub>1</sub>	2 <sub>1</sub>			
10	1 <sub>1</sub>	2 <sub>1</sub>	2 <sub>1</sub>	2 <sub>1</sub>	3 <sub>1</sub>	1 <sub>1</sub>	3 <sub>1</sub>	2 <sub>1</sub>	3 <sub>1</sub>	3 <sub>1</sub>		
11	1 <sub>1</sub>	2 <sub>1</sub>	2 <sub>1</sub>	2 <sub>1</sub>	1 <sub>1</sub>	3 <sub>1</sub>	2 <sub>1</sub>	3 <sub>1</sub>	3 <sub>1</sub>	3 <sub>1</sub>	2 <sub>1</sub>	
12	1 <sub>1</sub>	2 <sub>1</sub>	2 <sub>1</sub>	1 <sub>1</sub>	2 <sub>1</sub>	2 <sub>1</sub>	3 <sub>1</sub>	3 <sub>1</sub>	3 <sub>1</sub>	2 <sub>1</sub>	3 <sub>1</sub>	1 <sub>1</sub>
13	1 <sub>1</sub>	2 <sub>1</sub>	1 <sub>1</sub>	2 <sub>1</sub>	2 <sub>1</sub>	2 <sub>1</sub>	3 <sub>1</sub>	3 <sub>1</sub>	2 <sub>1</sub>	3 <sub>1</sub>	1 <sub>1</sub>	3 <sub>1</sub>
14	1 <sub>1</sub>	1 <sub>1</sub>	2 <sub>1</sub>	2 <sub>1</sub>	2 <sub>1</sub>	2 <sub>1</sub>	3 <sub>1</sub>	2 <sub>1</sub>	3 <sub>1</sub>	1 <sub>1</sub>	3 <sub>1</sub>	3 <sub>2</sub>
15	1 <sub>1</sub>	1 <sub>1</sub>	2 <sub>1</sub>	3 <sub>1</sub>	1 <sub>1</sub>	3 <sub>1</sub>	3 <sub>2</sub>	3 <sub>2</sub>				
16	1 <sub>1</sub>	1 <sub>1</sub>	2 <sub>1</sub>	1 <sub>1</sub>	3 <sub>1</sub>	3 <sub>2</sub>	3 <sub>2</sub>	3 <sub>2</sub>				
17	1 <sub>1</sub>	1 <sub>1</sub>	2 <sub>1</sub>	2 <sub>1</sub>	2 <sub>1</sub>	2 <sub>1</sub>	1 <sub>1</sub>	2 <sub>1</sub>	3 <sub>2</sub>	3 <sub>2</sub>	3 <sub>2</sub>	3 <sub>2</sub>
18	1 <sub>1</sub>	1 <sub>1</sub>	2 <sub>1</sub>	2 <sub>1</sub>	2 <sub>1</sub>	1 <sub>1</sub>	2 <sub>1</sub>	2 <sub>2</sub>	3 <sub>2</sub>	3 <sub>2</sub>	3 <sub>2</sub>	3 <sub>2</sub>
19	1 <sub>1</sub>	1 <sub>1</sub>	2 <sub>1</sub>	2 <sub>1</sub>	1 <sub>1</sub>	2 <sub>1</sub>	2 <sub>2</sub>	2 <sub>2</sub>	3 <sub>2</sub>	3 <sub>2</sub>	3 <sub>2</sub>	3 <sub>2</sub>
20	1 <sub>1</sub>	1 <sub>1</sub>	2 <sub>1</sub>	1 <sub>1</sub>	2 <sub>1</sub>	2 <sub>2</sub>	2 <sub>2</sub>	2 <sub>2</sub>	3 <sub>2</sub>	3 <sub>2</sub>	3 <sub>2</sub>	3 <sub>2</sub>
21	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	2 <sub>1</sub>	2 <sub>2</sub>	2 <sub>2</sub>	2 <sub>2</sub>	2 <sub>2</sub>	3 <sub>2</sub>	3 <sub>2</sub>	3 <sub>2</sub>	3 <sub>2</sub>
22	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	2 <sub>2</sub>	3 <sub>2</sub>	3 <sub>2</sub>	3 <sub>2</sub>	3 <sub>4</sub>				
23	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>2</sub>	2 <sub>2</sub>	3 <sub>2</sub>	3 <sub>2</sub>	3 <sub>2</sub>	3 <sub>4</sub>				
24	1 <sub>1</sub>	1 <sub>2</sub>	1 <sub>2</sub>	2 <sub>2</sub>	3 <sub>2</sub>	3 <sub>2</sub>	3 <sub>4</sub>					
25	1 <sub>2</sub>	1 <sub>2</sub>	1 <sub>2</sub>	2 <sub>2</sub>	3 <sub>4</sub>							
26	1 <sub>2</sub>	1 <sub>2</sub>	1 <sub>2</sub>	2 <sub>2</sub>								
27	1 <sub>2</sub>	1 <sub>2</sub>	1 <sub>2</sub>	2 <sub>2</sub>								
28	1 <sub>2</sub>	1 <sub>2</sub>	1 <sub>2</sub>	2 <sub>4</sub>								
29	1 <sub>2</sub>	1 <sub>2</sub>	1 <sub>2</sub>	2 <sub>2</sub>	2 <sub>2</sub>	2 <sub>2</sub>	2 <sub>4</sub>					
30	1 <sub>2</sub>	1 <sub>2</sub>	1 <sub>2</sub>	2 <sub>2</sub>	2 <sub>2</sub>	2 <sub>4</sub>						
31	1 <sub>2</sub>	1 <sub>2</sub>	1 <sub>2</sub>	2 <sub>2</sub>	2 <sub>4</sub>							
32	1 <sub>2</sub>	1 <sub>2</sub>	1 <sub>2</sub>	2 <sub>4</sub>								
33	1 <sub>2</sub>	1 <sub>2</sub>	2 <sub>4</sub>									
34	1 <sub>2</sub>	2 <sub>4</sub>										
35	2 <sub>4</sub>											

$E_2 = \{E_{21}, \dots, E_{2k}\}$ . Состояния подмножеств  $E_1$  и  $E_2$  поставлены во взаимно однозначное соответствие. Идея, использованная для сравнения размеров голосующих коалиций, ясна из рис. 3-12. Каждый раз, проходя границу голосующих коалиций, сигнал  $E_2$  порождает сигнал  $E_3$ , двигающийся в противоположную сторону. На фронте сигнала  $E_3$  происходит переход автоматов из состояний подмножества  $E_2$  в состояния  $E_3 = \{E_{31}, \dots, E_{3k}\}$ . Состояния подмножеств  $E_2$  и  $E_3$  поставлены во взаимно однозначное соответствие. Сигнал  $E_3$  распространяется только в пределах одной голосующей коалиции. На краю цепи (на крайний автомат цепи подан граничный сигнал  $K$ ) сигнал  $E_3$  порождает отраженный сигнал  $E_4 = \{E_{41}, \dots, E_{4k}\}$ . Сигнал  $E_{4j}$  распространяется по цепи, не изменяя своего индекса  $j$  до тех пор, пока он не встретится с автоматом, находящимся в состоянии, принадлежащем подмножеству  $E_2$ . В этом случае сигнал  $E_4$  прекращает распространение. Автомат находится в состоянии, принадлежащем подмножеству  $E_4$ , один такт, после чего возвращается в начальное состояние. Сигнал  $E_3$  порождает отраженный сигнал  $E_4$  каждый раз, когда он встречает автомат, находящийся в начальном состоянии. Из сказанного, рис. 3-12 и табл. 3-3 очевидна процедура сравнения размеров голосующих коалиций. Выходным сигналом на крайнем автомате является сигнал  $E_{4j}$ , индекс которого  $j$  является значением функции голосования. Правила смены состояний автомата приведены в табл. 3-4. Получившийся автомат имеет  $4k+1$  внутренних состояний. Время между подачей стартового сигнала и получением значения функции лежит в пределах

$2n+2 \left\lceil \frac{n}{k} \right\rceil \leq T \leq 4n$ . Точное значение  $T$  складывается из удвоенных времен прохождения сигналом всей цепи и голосующей коалиции максимальной длины.

Заметим, что эта задача была решена в предположении, что используется автомат Мура. Она может быть решена и на автоматах Мили способами, примененными в задаче (§ 2-1) для случая  $k=2$ . При этом число боковых каналов имеет порядок  $k$ , а число внутренних состояний  $k+1$ .

**Задача 4 [Л. 5].** Построить одномерную двустороннюю систему, реализующую произвольную симметрическую булеву функцию, заданную набором ее рабочих чисел (рис. 3-13).

Таблица 3-4

Номер	Правило			Условие
1	$x$	$H$	$z$	$x \in \{H, E_{4j}\}; z \in \{K, H\}$
		$H$		
2	$x$	$H$	$z$	$x \in \{E_{1j}\}; z \in \{K, H\}$
		$E_{1j}$		
3	$E_{1a}$	$E_{1b}$	$E_{1c}$	$b^* = \max(\text{sign}(c - b) \max(b, a),$ $\text{sign}(b - a) \min(b, c),$ $\text{sign } x = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$
		$E_{1b^*}$		
4	$E_{1a}$	$E_{1b}$	$z$	$z \in \{K, H\}, b^* = \max(a, b)$
		$E_{1l^*}$		
5	$K$	$E_{1b}$	$E_{1c}$	$b^* = \min(b, c)$
		$E_{1b^*}$		
6	$x$	$E_{1j}$	$z$	$x \in \{K, E_{1j}, E_{1i}\}; z \in \{K, E_{2j}, E_{2i}\}$ $i < j < l$
		$E_{2j}$		
7	$x$	$E_{2j}$	$z$	$x \in \{E_{1j}, E_{2j}\}; z \in \{K, H, E_{2j}, E_{3l}, E_{4l}\}$ $j < l$
		$E_{2j}$		
8	$x$	$E_{2j}$	$z$	$x \in \{K, E_{1j}, E_{1i}\}; z \in \{K, H, E_{2j}, E_{3l}, E_{4l}\}$ $i < j < l$
		$E_{3j}$		
9	$x$	$E_{3j}$	$z$	$x \in \{K, E_{2i}, E_{3i}, E_{3j}\}; z \in \{E_{2j}, E_{3j}, E_{3i}\}$ $i < j < l$
		$E_{3j}$		
10	$x$	$E_{3j}$	$z$	$x \in \{K, E_{2i}, E_{3i}, E_{3j}\}; z \in \{K, H, E_{4j}\}$ $i < j$
		$E_{4j}$		
11	$x$	$E_{3j}$	$E_{4l}$	$x \in \{K, E_{2i}, E_{3j}, E_{3i}\}$ $i < j < l$
		$E_{4l}$		
12	$x$	$E_{4j}$	$z$	$x \in \{K, E_{2i}, E_{3i}, E_{3j}\}; z \in \{K, H\}$ $i < j$
		$H$		

Будем использовать элементы, имеющие по два внешних входа, на один из которых подается переменная  $x_i$  ( $1 \leq i \leq n$ ), а на второй — константы 1 или 0 в соответствии с тем, равно ли одно из рабочих чисел  $r_i$  по-

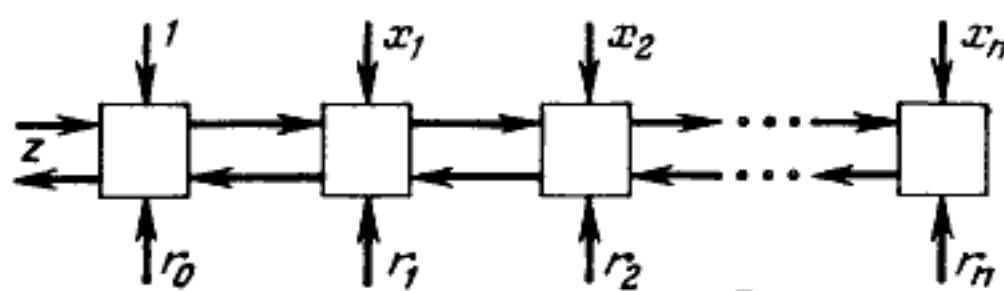


Рис. 3-13. Двусторонняя сеть, реализующая произвольную симметрическую булеву функцию.

Таблица 3-5

Номер	Правило			Условие
1	$x$	$y$ $H$	$z$	1) $x \in \{H, 0_2, 1_2, X, K\}; y = H; z \in \{H, K\}$ 2) $x \in \{1_1, K\}; y \in \{0_2, 1_2\}; z \in \{H, K\}$ 3) $x \in \{0_1, 1_1\}; y = X; z \in \{H, K\}$
2	$x$	$y$ $0_1$	$z$	1) $x \in \{0_1, 1_1\}; y = 0_1; z \in \{H, 0_1\}$ 2) $x = 0_1; y = 1_1; z \in \{H, 0_1\}$
3	$x$	$y$ $1_1$	$z$	1) $x \in \{0_1, 1_1\}; y = 0_1; z = 1_1$ 2) $x \in \{1_1, K\}; y = 1_1; z \in \{H, 0_1, 1_1\}$
4	$x$	$1_1$ $0_2$	$0_2$	$x \in \{1_1, K\}$
5	$x$	$1_1$ $1_2$	$1_2$	$x \in \{1_1, K\}$
6	$x$	$y$ $X$	$z$	1) $x \in \{0_1, 1_1\}; y = 0_1; z \in \{X, K\}$ 2) $x = 0_1; y = 1_1; z \in \{X, K\}$
7	$x$	$H$	$z$	$x \in \{0_1, 1_1\}; z \in \{H, K\}$ $t = 0_1$ , если $x = 0$ , $t = 1_1$ , если $x = 1$
8	$x$	$1_1$ $p$	$z$	$x \in \{1_1, K\}; z \in \{K, X\}$ $p = 0_2$ , если $r = 0$ , $p = 1_2$ , если $r = 1$

рядковому номеру  $i$  элемента или нет. Так как для произвольной симметрической функции количество рабочих чисел равно  $n+1$ , то для подачи на схему всех рабочих чисел и переменных достаточно иметь  $n+1$  автоматов.

Таблица 3-6

	Значения переменных													
	1	0	0	1	0	1	1	1	1	1	1	0	1	
	Состояния автомата													
1	1 <sub>1</sub>	H	H	H	H	H	H	H	H	H	H	H	H	H
2	1 <sub>1</sub>	0 <sub>1</sub>	H	H	H	H	H	H	H	H	H	H	H	H
3	1 <sub>1</sub>	0 <sub>1</sub>	0 <sub>1</sub>	H	H	H	H	H	H	H	H	H	H	H
4	1 <sub>1</sub>	0 <sub>1</sub>	0 <sub>1</sub>	1 <sub>1</sub>	H	H	H	H	H	H	H	H	H	H
5	1 <sub>1</sub>	0 <sub>1</sub>	1 <sub>1</sub>	0 <sub>1</sub>	0 <sub>1</sub>	H	H	H	H	H	H	H	H	H
6	1 <sub>1</sub>	1 <sub>1</sub>	0 <sub>1</sub>	0 <sub>1</sub>	0 <sub>1</sub>	1 <sub>1</sub>	H	H	H	H	H	H	H	H
7	1 <sub>1</sub>	1 <sub>1</sub>	0 <sub>1</sub>	0 <sub>1</sub>	1 <sub>1</sub>	0 <sub>1</sub>	1 <sub>1</sub>	H	H	H	H	H	H	H
8	1 <sub>1</sub>	1 <sub>1</sub>	0 <sub>1</sub>	1 <sub>1</sub>	0 <sub>1</sub>	1 <sub>1</sub>	0 <sub>1</sub>	1 <sub>1</sub>	H	H	H	H	H	H
9	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	0 <sub>1</sub>	1 <sub>1</sub>	0 <sub>1</sub>	1 <sub>1</sub>	0 <sub>1</sub>	1 <sub>1</sub>	H	H	H	H	H
10	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	0 <sub>1</sub>	1 <sub>1</sub>	0 <sub>1</sub>	1 <sub>1</sub>	0 <sub>1</sub>	1 <sub>1</sub>	H	H	H	H
11	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	0 <sub>1</sub>	1 <sub>1</sub>	0 <sub>1</sub>	1 <sub>1</sub>	0 <sub>1</sub>	1 <sub>1</sub>	H	H	H
12	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	0 <sub>1</sub>	1 <sub>1</sub>	0 <sub>1</sub>	1 <sub>1</sub>	0 <sub>1</sub>	0 <sub>1</sub>	H	H
13	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	0 <sub>1</sub>	1 <sub>1</sub>	0 <sub>1</sub>	0 <sub>1</sub>	0 <sub>1</sub>	0 <sub>1</sub>	1 <sub>1</sub>
14	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	0 <sub>1</sub>	0 <sub>1</sub>	0 <sub>1</sub>	0 <sub>1</sub>	1 <sub>1</sub>	X
15	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	0 <sub>1</sub>	0 <sub>1</sub>	1 <sub>1</sub>	X	H
16	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	0 <sub>1</sub>	1 <sub>1</sub>	X	H	H
17	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	X	H	H	H
18	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	0 <sub>2</sub>	H	H	H	H
19	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	0 <sub>2</sub>	H	H	H	H	H
20	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	0 <sub>2</sub>	H	H	H	H	H	H	H
21	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	0 <sub>2</sub>	H	H	H	H	H	H	H	H
22	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	0 <sub>2</sub>	H	H	H	H	H	H	H	H	H
23	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	0 <sub>2</sub>	H	H	H	H	H	H	H	H	H	H
24	1 <sub>1</sub>	1 <sub>1</sub>	0 <sub>2</sub>	H	H	H	H	H	H	H	H	H	H	H
25	1 <sub>1</sub>	0 <sub>2</sub>	H	H	H	H	H	H	H	H	H	H	H	H
26	0 <sub>2</sub>	H	H	H	H	H	H	H	H	H	H	H	H	H

Теперь работу схемы можно организовать следующим образом. После подачи на автомат  $A_0$  пускового сигнала по цепи начинает распространяться сигнал  $y_1$  (состояния  $0_1$  и  $1_1$ , правило 7 табл. 3-5), который производит считывание значений переменных в автоматы. За фронтом сигнала происходит упорядочение значений перемен-

ных, т. е. все единицы собираются к началу сети, а нули — к ее концу (правому краю в табл. 3-6). Упорядочению соответствуют правила 2 и 3 табл. 3-5. Сигнал  $Y_1$  отражается от края сети и движется обратно (состояние  $\bar{X}$ ). На границе между областью нулей и единиц формируется выходной сигнал, совпадающий со значением рабочего числа на граничном автомате (состояния  $0_2$  и  $1_2$ , правило 8 табл. 3-5). При таком алгоритме автомат имеет 6 внутренних состояний. Правила перехода для него задаются табл. 3-5. Пример реализации симметрической функции 12 переменных с рабочими числами 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0 на одном из наборов приведен в табл. 3-6.

Заметим, что некоторым изменением структуры элемента можно расширить логические возможности приведенной выше схемы. Назовем функции, полученные из симметрической инвертированием некоторых переменных, *квазисимметрическими* (они в общем случае не являются симметрическими). Тогда, для того чтобы описанная система реализовала произвольную квазисимметрическую функцию, необходимо на каждый автомат добавить один вход  $\sigma_i$  и схему  $z_i = x_i \oplus \sigma_i$ , где  $z_i$  становится внешней переменной для элемента  $A_i$ .

### 3-4. О возможностях моделирования поведения двусторонних одномерных сетей односторонними кольцами

Рассмотрим сначала двустороннюю автономную сеть из автоматов Мура (рис. 3-14). Состояние  $j$ -го автомата сети в момент времени  $t+1$  является функцией его состояния и состояний двух его соседей в предыдущий момент времени, т. е.

$$a_j(t+1) = f_1[a_{j-1}(t), a_j(t), a_{j+1}(t)], \quad 0 \leq j \leq n-1,$$

где  $n$  — число элементов сети;  $a_{-1}$  и  $a_n$  — граничные условия.

Такой сети из  $n$  элементов поставим в соответствие одностороннее кольцо из  $n$  автоматов Мура (рис. 3-15). Поведение  $j$ -го автомата кольца, очевидно, описывается выражением

$$b_j(t+1) = f_2[b_{j-1}(t), b_j(t)],$$

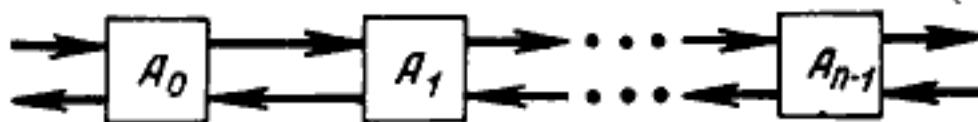


Рис. 3-14. Двусторонняя одномерная сеть из элементов с памятью.

где отчет  $j$  производится по модулю  $n$ . Пусть каждый автомат кольца состоит из двух подавтоматов (рис. 3-16) — верхнего  $U$  и нижнего  $D$ , состояния которых будем обозначать буквами  $u$  и  $d$  соответственно.

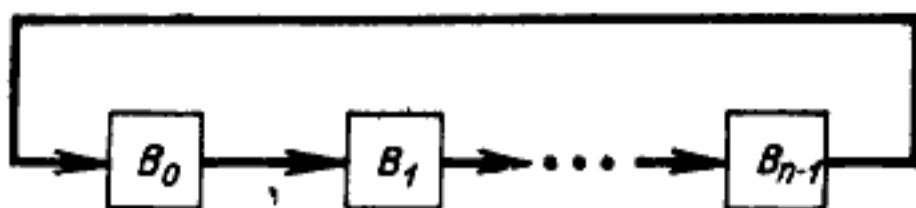


Рис. 3-15. Одностороннее кольцо из элементов с памятью.

Зададим следующие правила функционирования подавтоматов. В четные такты состояние подавтомата  $U_j$ ,  $j$ -го автомата передается подавтоматам  $D_j$  и  $U_{j+1}$ , т. е.

$$d_j(2t) = u_j(2t-1);$$

$$u_{j+1}(2t) = u_j(2t-1),$$

где  $j$  отсчитывается по модулю  $n$ ,  $t=0, 1, 2 \dots$

В нечетные такты переходы подавтоматов  $j$ -го автомата определяются функциями вида

$$d_j(2t+1) = d_0;$$

$$u_j(2t+1) = f_2[u_{j-1}(2t), u_j(2t), d_j(2t)].$$

В силу правил функционирования автомата в четные такты

$$u_j(2t+1) = f_3[u_{j-2}(2t-1), u_{j-1}(2t-1), u_j(2t-1)].$$

Сравнивая это выражение с выражением для функции перехода автомата двусторонней сети, заметим, что оба автомата обладают одинаковым функциональным поведением, если сделать следующие допущения:

$$1) f_1 = f_3,$$

2) автомат кольца работает вдвое быстрее автомата двусторонней сети, причем состояния автомата кольца наблюдаются только в нечетные моменты времени.

Таким образом, можно говорить, что кольцо из автоматов моделирует поведение двусторонней сети, если оказывается несущественным то обстоятельство, что в четные моменты времени осуществляется циклический сдвиг всех состояний подавтоматов  $U$  на один автомат. Тогда если

$$a_j(t) = u_{j(\text{mod } n)}(2t+1), \text{ то}$$

$$a_j(t+1) = u_{j+1(\text{mod } n)}(2t+3).$$

При этом предполагается, что в начальный момент времени элементы обеих сетей поставлены во взаимно однозначное соответствие (пронумерованы) и  $a_j = u_j$ .

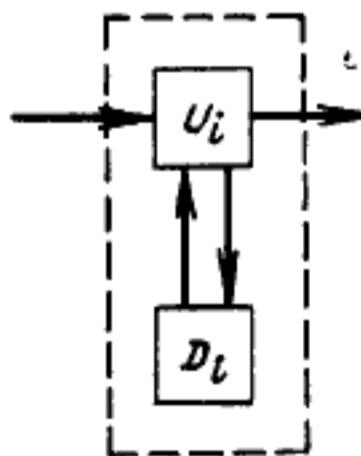


Рис. 3-16. Структура элемента кольца.

Сказанное дополним следующим замечанием. Наличие граничных условий в двусторонней сети и их отсутствие в кольце может быть учтено при моделировании сети кольцом введением дополнительных состояний элемента кольца.

Рассмотрим случай, когда двусторонняя сеть из автоматов Мура не является автономной, т. е. когда на элементы сети подаются внешние переменные. В этом случае, для того чтобы одностороннее кольцо моделировало поведение двусторонней сети, необходимо осуществить циклический сдвиг переменных на один элемент за каждые два такта. Считывание значений внешних выходов должно осуществляться с таким же циклическим сдвигом.

## Глава четвертая

# МНОГОМЕРНЫЕ СЕТИ И СИСТЕМЫ

### 4-1. Многомерные системы

**Анализ.** Относительно многомерных систем известна следующая теорема.

**Теорема [Л. 27].** Любая стабильная сеть или система из комбинационных элементов регулярна.

Покажем сначала, что любая нерегулярная сеть (система) нестабильна. Рассмотрим два случая.

1. Сеть нерегулярна потому, что для некоторой комбинации внешних входов она не имеет состояния равновесия. Очевидно, что в этом случае сеть имеет цикл и, следовательно, нестабильна.

2. Сеть нерегулярна потому, что для некоторой комбинации внешних входов она имеет по крайней мере два различных состояния равновесия  $E_1$  и  $E_2$ . Покажем, что в этом случае существует хотя бы один набор внешних входов, при подаче которого сеть входит в цикл.

Разделим элементы сети (рис. 4-1) на два множества  $P_1$  и  $P_2$ . К  $P_1$  принадлежат элементы с нечетной суммой координат, а к  $P_2$  — с четной. В сети (рис. 4-1) им соответствуют белые и заштрихованные квадраты. В предположении, что в начальный момент  $t=0$  состояния элементов можно задать произвольно, переведем элементы множества  $P_1$  в состояния, соответствующие состоянию сети  $E_1$ , а элементы множества  $P_2$  в состояния, соответствующие  $E_2$ . Считаем, что на внешние входы в момент  $t=0$  подается набор, которому соответствуют равновесные состояния  $E_1$  и  $E_2$ . Тогда в момент  $t=1$  все межэлементные связи, по которым идут сигналы на элементы множества  $P_1$ , будут нести сигналы, соответствующие состоянию равновесия  $E_2$ , поскольку они приходят от элементов множества  $P_2$ . Таким образом, в момент  $t=1$  элементы множества  $P_1$  будут находиться в состояниях, соответствующих равновесному состоянию  $E_2$ . Аналогично элементы  $P_2$  будут в  $E_1$ . В  $t=2$  ситуация снова поменяется и схема перейдет в то же состояние, в котором она была при  $t=0$ . Поскольку предполагалось, что  $E_1$  и  $E_2$  отличны друг от друга, сеть входит в цикл.

Очевидно, может быть сделано обобщение сделанного утверждения на случай более двух измерений.

Поскольку регулярная сеть всегда стабильна, то из доказанного следует справедливость теоремы.

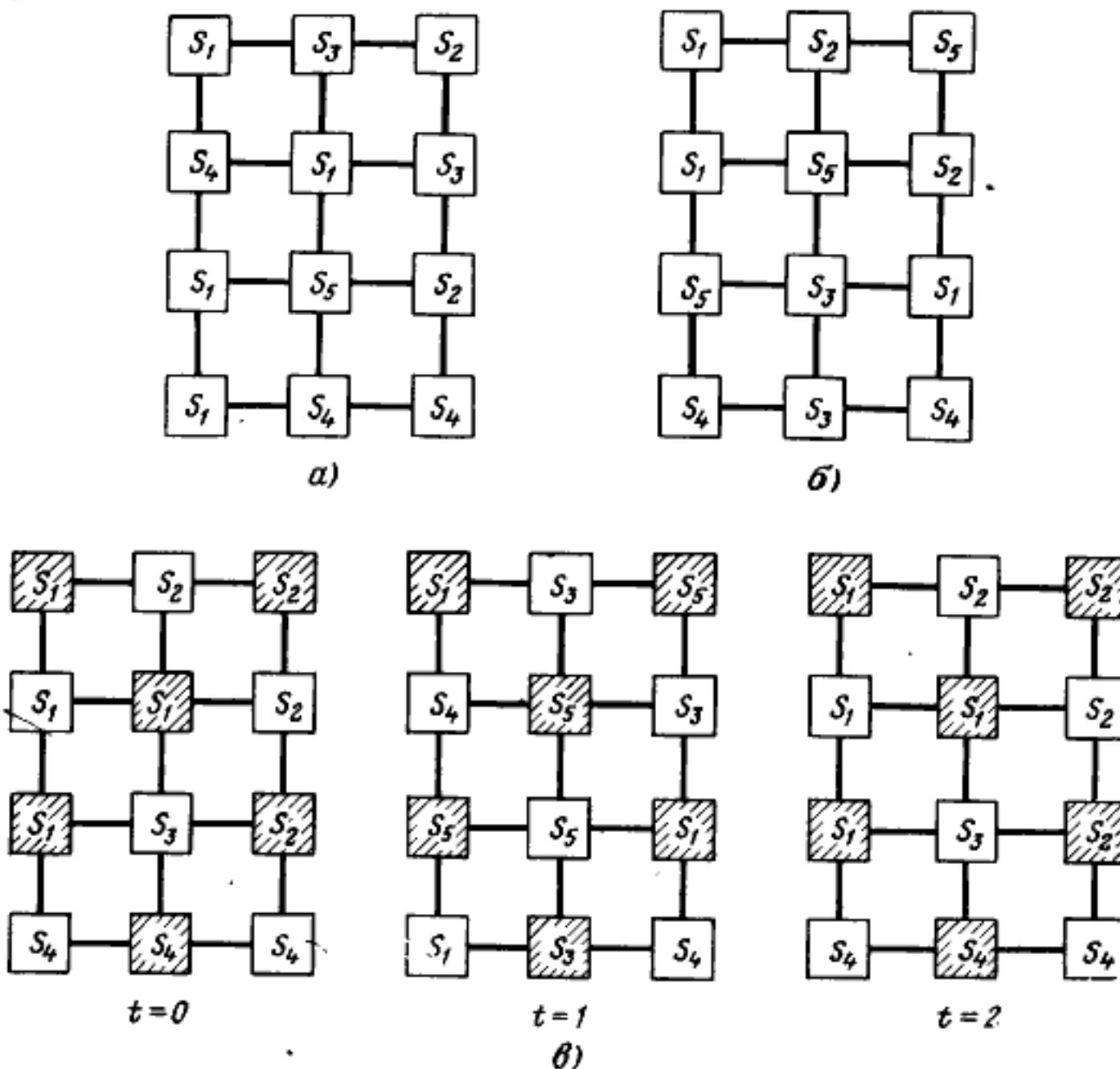


Рис. 4-1. Пример нестабильной сети с двумя состояниями равновесия.

*a* — состояние равновесия  $E_1$ , *b* — состояние равновесия  $E_2$ , *c* — цикл в сети.

Возникает вопрос, существует ли для таких систем общий тест на регулярность. Оказывается, что такой тест существует лишь для сравнительно узкого класса многомерных систем из комбинационных элементов. Определим свойства таких систем.

Взаимодействие сигналов, протекающих в различных направлениях, может быть представлено направленными графами. Поставим в соответствие каждой координатной оси вершину графа. Если сигналы, протекающие параллельно  $i$ -й оси в любом направлении, могут влиять на сигналы, протекающие параллельно  $j$ -й оси, то на графике

имеется стрелка, направленная от вершины  $i$  к вершине  $j$ . Взаимное влияние сигналов, текущих параллельно одной и той же оси, на графе не указывается.

В качестве примера на рис. 4-2 приведено два графа, соответствующих некоторым трехмерным системам.

Оказывается, что общий тест на регулярность существует лишь для систем, графы которых не содержат замкнутых путей. Доказательство этого утверждения читатель может найти в [Л. 27]. Поэтому система, кото-

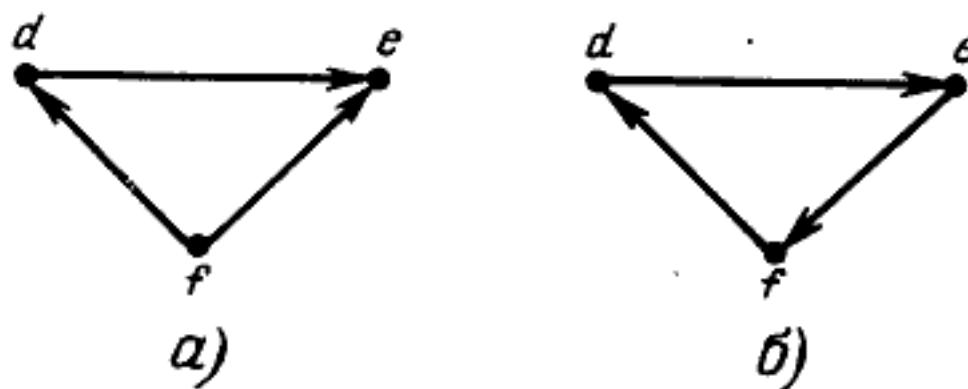


Рис. 4-2. Примеры графов, соответствующих системам, для которых тест на регулярность существует (a) и не существует (b).

рой соответствует граф на рис. 4-2, может быть проверена на регулярность, в то время как для системы на рис. 4-2 такой проверки не существует.

Вершины графа, не содержащего замкнутых путей, упорядочиваются следующим образом. Вершины без входящих стрелок называются вершинами первого порядка. К вершинам  $k$ -го порядка отнесем те из них, в которые входят стрелки от вершин не более чем  $(k-1)$ -го порядка. На рис. 4-2, а вершины имеют такие порядки:  $f-1$ ,  $d-2$ ,  $e-3$ . Упорядочения для графа на рис. 4-2, б провести нельзя, так как он содержит замкнутый путь.

В соответствии с проводимым упорядочением всю систему разобьем на подсистемы соответствующих порядков, причем каждая подсистема  $k$ -го порядка содержит одну или более подсистем меньших порядков.

Полная система может быть исследована на регулярность посредством проверки на регулярность всех ее подсистем, начиная с подсистем младшего порядка. Процедура анализа двусторонних систем была описана ранее (§ 3-1). Если же подсистема является односторонней, то необходимости в анализе ее не возникает.

Отметим, что для систем более чем второго порядка эта методика очень громоздка.

Аналогичным способом две системы из комбинационных элементов могут быть проверены на эквивалентность лишь в том случае, если соответствующие им графы состоят из одинакового числа вершин и не имеют циклов.

**Функциональные возможности двумерных систем.** В предыдущей главе было показано, что одновыходные регулярные односторонние и двусторонние системы обладают одинаковыми функциональными возможностями. Что касается двумерных систем, то регулярные одновыходные системы с тремя направлениями потока сигналов обладают большими возможностями, чем системы с двумя направлениями потока. Подтвердим это примером.

**Задача 1** [Л. 27]. Построить систему, выход каждой сети которой равен 1, если длина этой сети кратна ширине, и 0 в противном случае.

Можно показать [Л. 27], что эта задача не может быть решена регулярной двумерной системой с двумя направлениями потока. Как решать эту задачу с помощью регулярной двумерной системы с тремя направлениями потока, видно из рис. 4-3. Для решения этой задачи не нужны внешние входы, поэтому система автономна.

В [Л. 27] показано, что функциональные возможности двумерных систем с четырьмя направлениями потока больше, чем у систем с тремя направлениями.

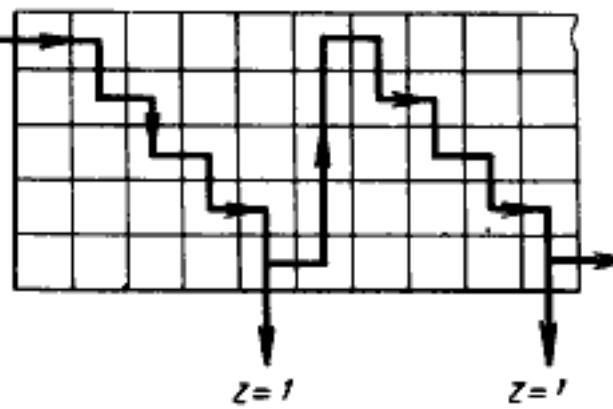


Рис. 4-3. Способ решения задачи 1.

**Синтез.** Многомерные системы обладают очень сложным поведением, поэтому подход к их синтезу очень труден. В частности, трудно свести многомерную решетку к нескольким решеткам с меньшим числом направлений сигнального потока. Поскольку в случае двумерной решетки встречаются почти все трудности, что и в решетках с большим числом измерений, рассмотрим пример синтеза двумерной системы.

**Задача 2** [Л. 27]. Построить систему, внешний выход каждого элемента которой равен 1, если он принад-

лежит к связной группе не менее чем из пяти элементов с внешними входами, равными 1, и равен 0 в противном случае.

Группа называется *связной*, если из каждого элемента группы можно попасть в любой другой, проходя в направлении сигнальных потоков только по элементам группы.

Разобьем систему на четыре более простых системы, каждая из которых имеет два ортогональных направления сигнального потока.

Представим себе каждый элемент состоящим из четырех составляющих, соответствующих этим системам, и дополнительной комбинационной схемы, необходимой для определения внешнего выходного сигнала элемента. Если каждой составляющей произвольного элемента известна информация о состояниях внешних входов элементов, принадлежащих выделенному в каждой системе квадрату размером  $5 \times 5$  (см. рис. 4-4), то по этой информации элемент может

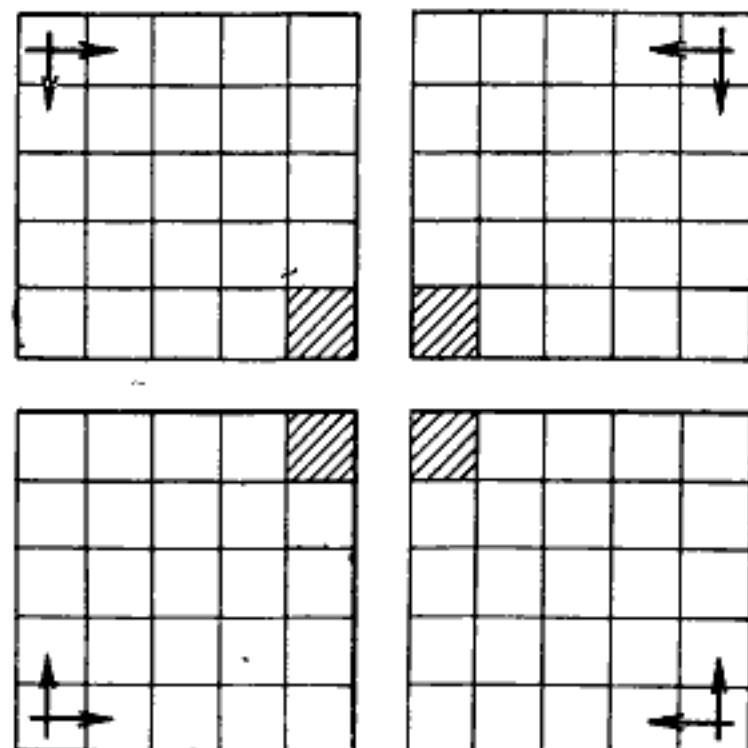


Рис. 4-4. Разбиение сети с восемью направлениями потока сигналов на четыре сети с двумя направлениями потока в каждой сети.

определить свою принадлежность к связной группе из пяти или более элементов.

Однако такой метод синтеза слишком громоздок. Кроме того, существуют двумерные системы, которые не могут быть разделены на четыре независимые системы меньшей сложности [Л. 27], т. е. этот метод синтеза не является общим. Можно сделать вывод, что расширение метода синтеза одномерных регулярных систем на двумерные регулярные системы не дает удовлетворительных результатов.

Очевидно, существуют задачи, которые не могут быть решены регулярными многомерными системами. В качестве примера приведем следующую задачу.

**Задача 3** [Л. 27]. Внешний выход элемента двумерной системы равен 1, если только этот элемент централь-

ный в сети и число элементов, расположенных между ним и каждым краем сети, нечетно (четно), и равен 0 в противном случае.

Даже в том случае, когда заранее известно, что данная задача может быть решена двумерной регулярной системой, часто легче и более экономично использовать нерегулярную систему.

Для иллюстрации этого положения вернемся к задаче 2. Одним из методов решения этой задачи является:

1. Сканирование сигнала по рядам элементов сети, как показано на рис. 4-5. Когда на пути этого сигнала встретится первый внешний вход, равный 1, заменить его на  $M$  и заменить все единицы в связной группе, в которую он входит, на  $B$ .

2. Повторить операцию сканирования. Сканирующий сигнал переводит в  $M$  первую встретившуюся единицу или  $B$ . Если первой встретится единица, то все единицы, входящие с ней в одну группу, переводятся в  $B$ . Если первой встретится  $B$ , то все  $B$ , связанные с ней, переводятся в  $C$ .

При следующих сканированиях состояние  $C$  переходит соответственно в  $M$  и  $D$ , состояние  $D$  — в  $M$  и  $E$ . Если сканирующий сигнал первым встретит элемент в состоянии  $E$ , то все элементы группы, в которую входит этот элемент, переходят в состояние  $F$  (в том числе и сам этот элемент). Если элемент находится в состоянии  $F$ , то он является членом связной группы из пяти или более элементов. При дальнейших сканированиях на элементы, находящиеся в состоянии  $F$ , сканирующий сигнал не действует. Все элементы связных групп, содержащих менее пяти элементов, перейдут в состояние  $M$ . На рис. 4-6 иллюстрируется состояние сети из 24 элементов после каждого сканирования. Нетрудно видеть, что каждый элемент сети имеет 8 состояний ( $0, 1, B, C, D, E, F, M$ ).

Рассмотрим еще одну задачу.

**Задача 4 [Л. 24].** Внешний выход элемента системы равен 1, если он является центральным элементом кон-

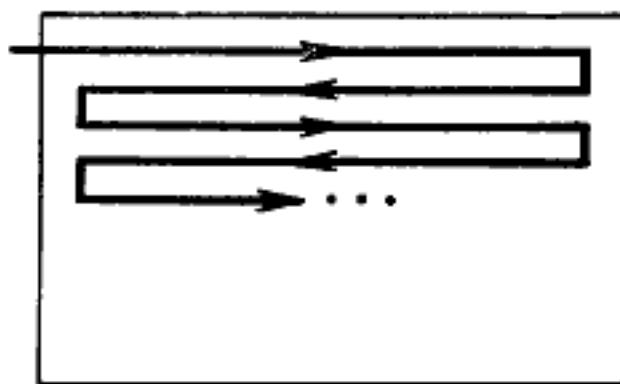


Рис. 4-5. Метод сканирования в двумерной сети.

0 1 0 1 0 0	0 М 0 В 0 0	0 М 0 М 0 0	0 М 0 М 0 0
0 1 1 1 0 1	0 В В В 0 1	0 С С С 0 1	0 С С С 0 М
0 0 1 0 1 1	0 0 В 0 1 1	0 0 С 0 1 1	0 0 С 0 В В
1 0 0 0 0 0	1 0 0 0 0 0	1 0 0 0 0 0	1 0 0 0 0 0
<i>a</i>	<i>b</i>	<i>v</i>	<i>z</i>
0 М 0 М 0 0	0 М 0 М 0 0	0 F 0 F 0 0	0 F 0 F 0 0
0 D D M 0 M	0 E M M 0 M	0 F F F 0 M	0 F F F 0 M
0 0 D 0 В В	0 0 E 0 В В	0 0 F 0 В В	0 0 F 0 M C
1 0 0 0 0 0	1 0 0 0 0 0	1 0 0 0 0 0	1 0 0 0 0 0
<i>d</i>	<i>e</i>	ж	<i>z</i>
0 F 0 F 0 0	0 F 0 F 0 0		
0 F F F 0 M	0 F F F 0 M		
0 0 F 0 M M	0 0 F 0 M M		
1 0 0 0 0 0	M 0 0 0 0 0		
<i>u</i>	<i>k</i>		

Рис. 4-6. Пример решения задачи 2 для сети из 24 элементов.

фигурации на рис. 4-7, и равен 0 в противном случае (задача нахождения центра креста).

В системе используются два сигнала  $S_0$  и  $S_1$  для каждого из четырех направлений сигнального потока. Временная последовательность межэлементных сигналов содержит столько сигналов  $S_1$ , сколько элементов

с внешним входом, равным 1, располагается подряд в данном направлении. Число сигналов в последовательности увеличивается на единицу в каждом последующем элементе с внешним входом, равным 1. Внешний выход элемента равен 1, если его внешний вход равен 1 и этот элемент получает последовательности, одинаковые по числу сигналов  $S_1$ , от своих соседей.

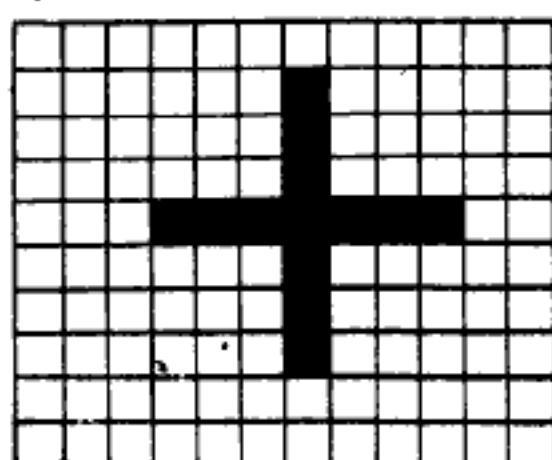


Рис. 4-7. Конфигурация, опознаваемая в задаче 4.

## 4-2. Однородные решетки из трехходовых элементов

В этом параграфе сделано допущение, что переменные на внешние входы элементов сети рис. 4-8 могут подаваться как в прямом виде, так и с инверсией. Если это по каким-либо причинам нежелательно, то к каждому элементу можно добавить еще один настроочный вход  $y_i$ , так что элемент вместо функции  $z=f(x_i, a_i, b_i)$  будет реализовать функцию  $z=f(x_i \oplus y_i, a_i, b_i)$ , так что при  $y_i=0$  на элемент подается  $x_i$  при  $y_i=1-\bar{x}_i$ . При этом, однако, увеличивается число входов.

1. Решетки из мажоритарных элементов (мажоритарный элемент с тремя входами реализует трехместную операцию  $x_1 \# x_2 \# x_3 = x_1x_2 \vee x_1x_3 \vee x_2x_3$ ). С точки зрения практической реализации схем из комбинационных элементов в рамках использования микрэлектронной технологии возникает задача синтеза однородных схем с заданной структурой связей. Как правило, методы синтеза, развитые для различных базисов, предполагают реализацию логических функций в виде пирамидальных конфигураций. Применение этих методов синтеза с учетом ограничений на топологию схем приводит к серьезным трудностям. Наиболее простым и общим выходом из этого положения является наложение пирамидальной конфигурации на заданную однородную структуру. При этом, естественно, намного увеличивается число элементов по сравнению с пирамидальной схемой, потому что часть элементов не используется совсем либо служит лишь для передачи сигналов. Наиболее полный обзор таких методов приведен в [Л. 44]. Здесь мы не можем подробно изложить все аспекты этого подхода. Приведем лишь пример.

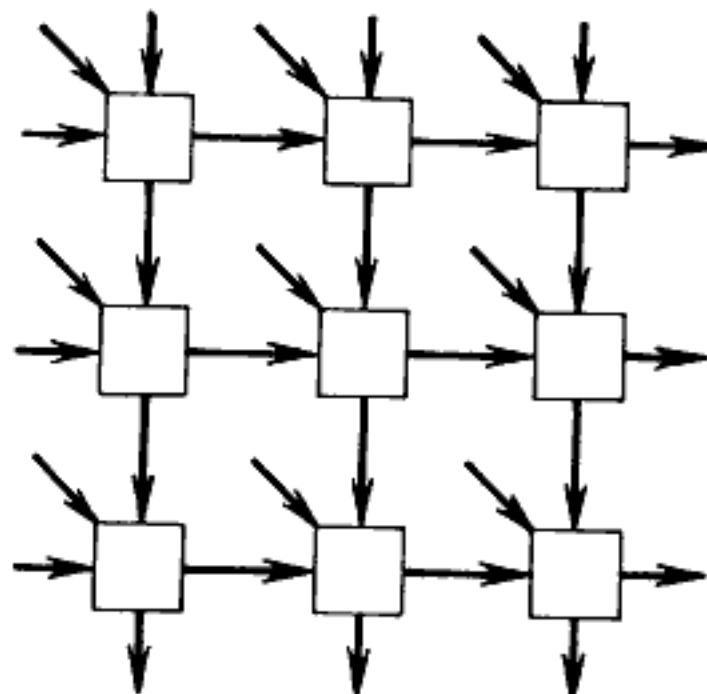


Рис. 4-8. Общий вид однородной решетки из трехходовых элементов.

Наиболее простым и общим выходом из этого положения является наложение пирамидальной конфигурации на заданную однородную структуру. При этом, естественно, намного увеличивается число элементов по сравнению с пирамидальной схемой, потому что часть элементов не используется совсем либо служит лишь для передачи сигналов. Наиболее полный обзор таких методов приведен в [Л. 44]. Здесь мы не можем подробно изложить все аспекты этого подхода. Приведем лишь пример.

Известна каноническая пирамидальная схема, связанная с последовательным исключением переменных для реализации произвольной функции в мажоритарном базисе. Для случая исключения трех переменных она представлена на рис. 4-9. Результат наложения этой пирамиды на прямоугольную решетку показан на рис. 4-10, который является канонической формой реализации лю-

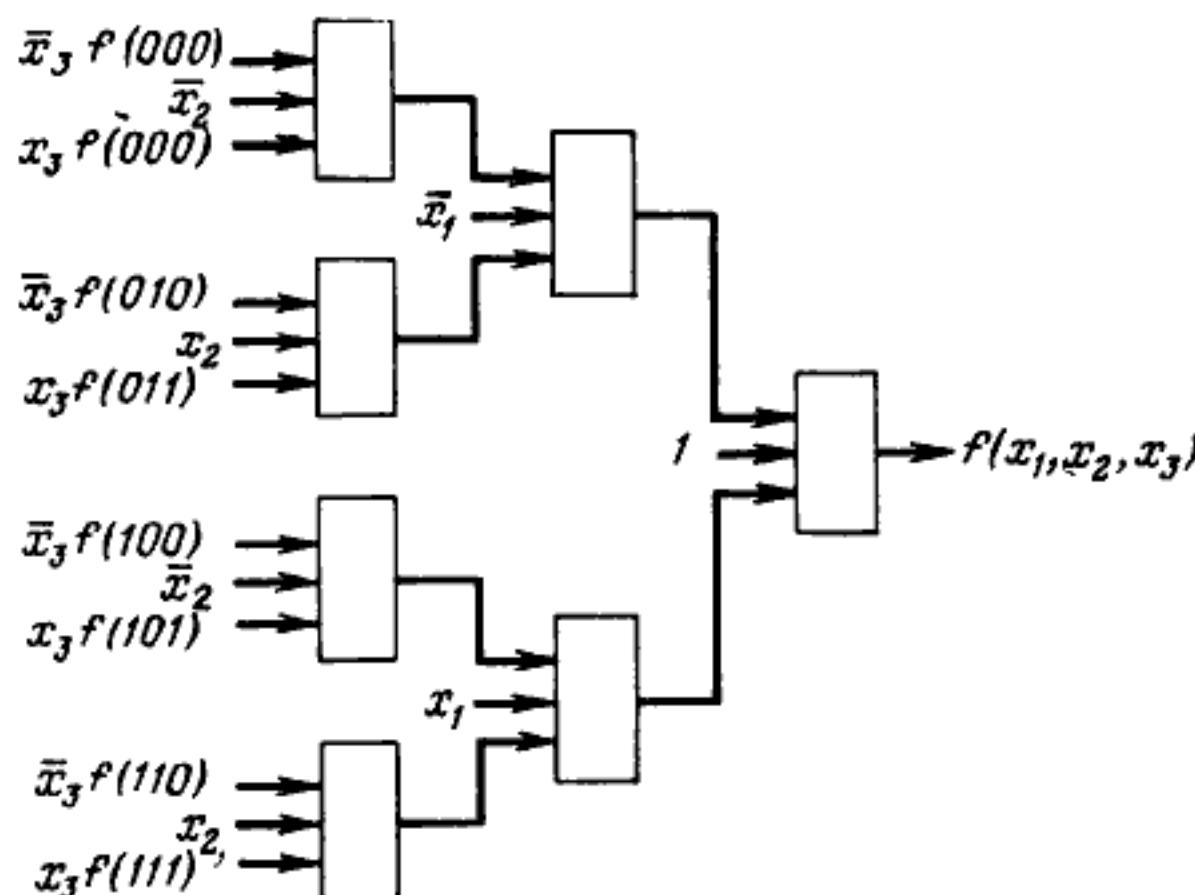


Рис. 4-9. Реализация произвольной логической функции пирамидальной схемой из мажоритарных элементов.

бой логической функции трех переменных на трехходовых мажоритарных элементах. Число требуемых элементов в этом случае равно  $3n2^n - 3 \cdot 2^{n-1} - 2n + 1$ , т. е. растет с ростом  $n$  приблизительно как  $3n2^n$ , в то время как в пирамидальной схеме число элементов равно  $2^n - 1$ .

На рис. 4-11 представлена каноническая схема реализации любой функции пяти переменных при использовании треугольной структуры.

Число требуемых элементов  $2^{2n-3} + 2^{n-2}$ .

Однако такой подход к синтезу однородных структур не экономичен. Лучшие результаты можно ожидать от применения специальных методов синтеза, ориентированных на синтез схем с заданной топологией.

В качестве примера рассмотрим метод синтеза решеток из мажоритарных элементов, предложенный в [Л. 22]. Как показывает анализ схемы, представленной на рис. 4-12, она реализует любую ло-

гическую функцию двух переменных. Действительно,

$$\begin{aligned} f = & ((1 \# \bar{x}_1 \# 0) \# 1 \# g_1) \# ((1 \# \bar{x}_1 \# 0) \# \\ & \# 0 \# g_0) \# x_1 = \bar{x}_1 g_0 \vee x_1 g_1, \\ g_0, g_1 \in & \{x_2, \bar{x}_2, 1, 0\}. \end{aligned}$$

Представленной на рисунке решетке из мажоритарных элементов можно однозначно поставить в соответствие матрицу

$$\begin{vmatrix} \bar{x}_1 & g_1 \\ g_0 & x_1 \end{vmatrix},$$

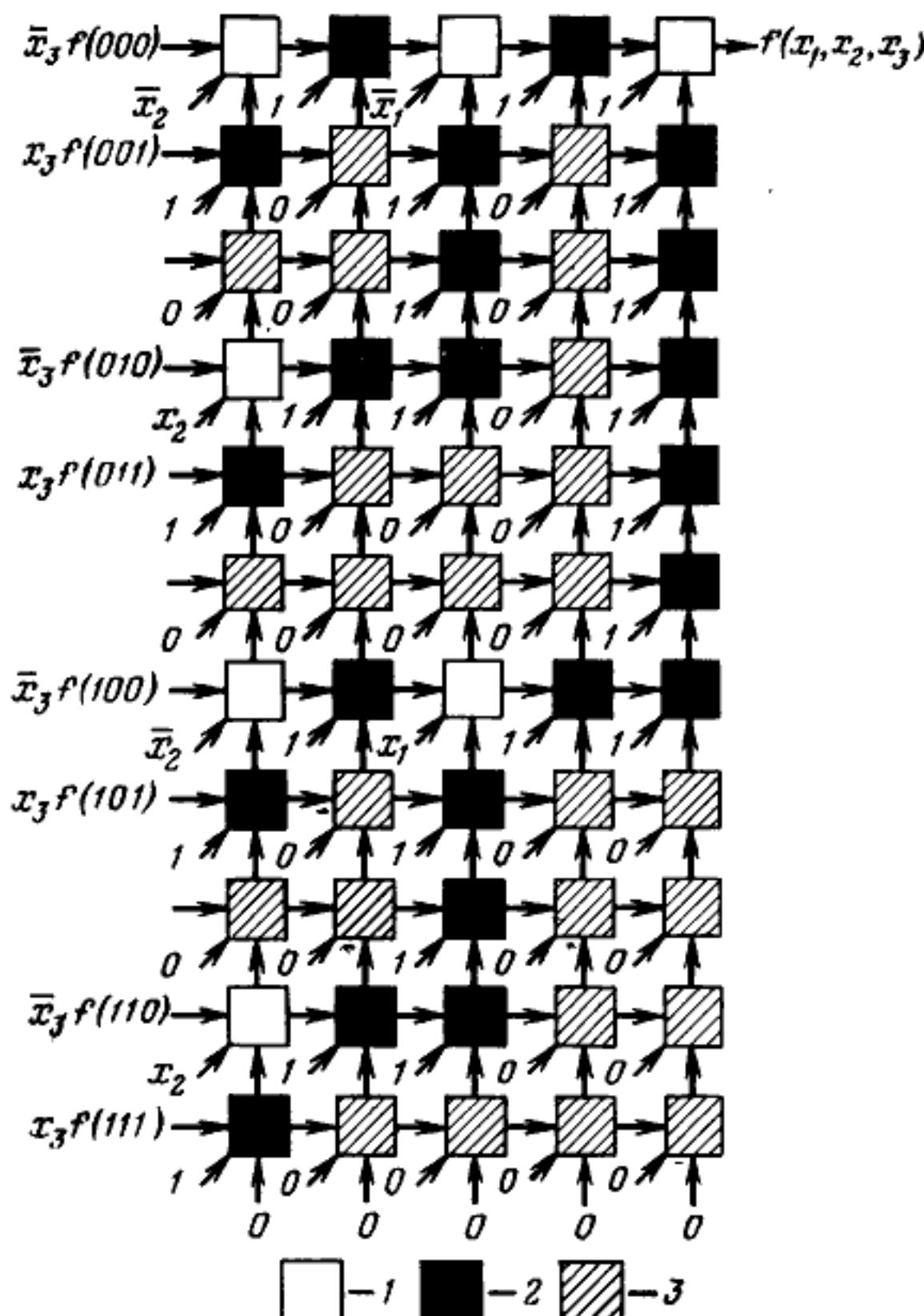


Рис. 4-10. Однородная решетка, реализующая произвольную функцию трех переменных.

1 — элемент, несущий логическую нагрузку; 2 — соединительный элемент; 3 — буферный элемент.

элементы которой соответствуют внешним входным сигналам. Решетка размером  $3 \times 3$ , которой соответствует матрица

$$\begin{vmatrix} \bar{x}_1 & g_{11} & \bar{x}_2 \\ g_{00} & x_2 & g_{10} \\ \bar{x}_2 & g_{01} & x_1 \end{vmatrix}$$

реализует любую функцию трех переменных:

$$f = \bar{x}_1 \bar{x}_2 g_{00} \vee \bar{x}_1 x_2 g_{01} \vee x_1 \bar{x}_2 g_{10} \vee x_1 x_2 g_{11},$$

$$g_{00}, g_{01}, g_{10}, g_{11} \in \{x_3, \bar{x}_3, 1, 0\}.$$

Матрицу для  $n = 4$  можно построить из двух матриц для  $n = 3$

$$\begin{vmatrix} \bar{x}_2 & g_{011} & \bar{x}_3 \\ g_{000} & x_3 & g_{010} \\ \bar{x}_3 & g_{001} & x_2 \end{vmatrix} \text{ и } \begin{vmatrix} \bar{x}_2 & g_{111} & \bar{x}_3 \\ g_{100} & x_3 & g_{110} \\ \bar{x}_3 & g_{101} & x_2 \end{vmatrix}$$

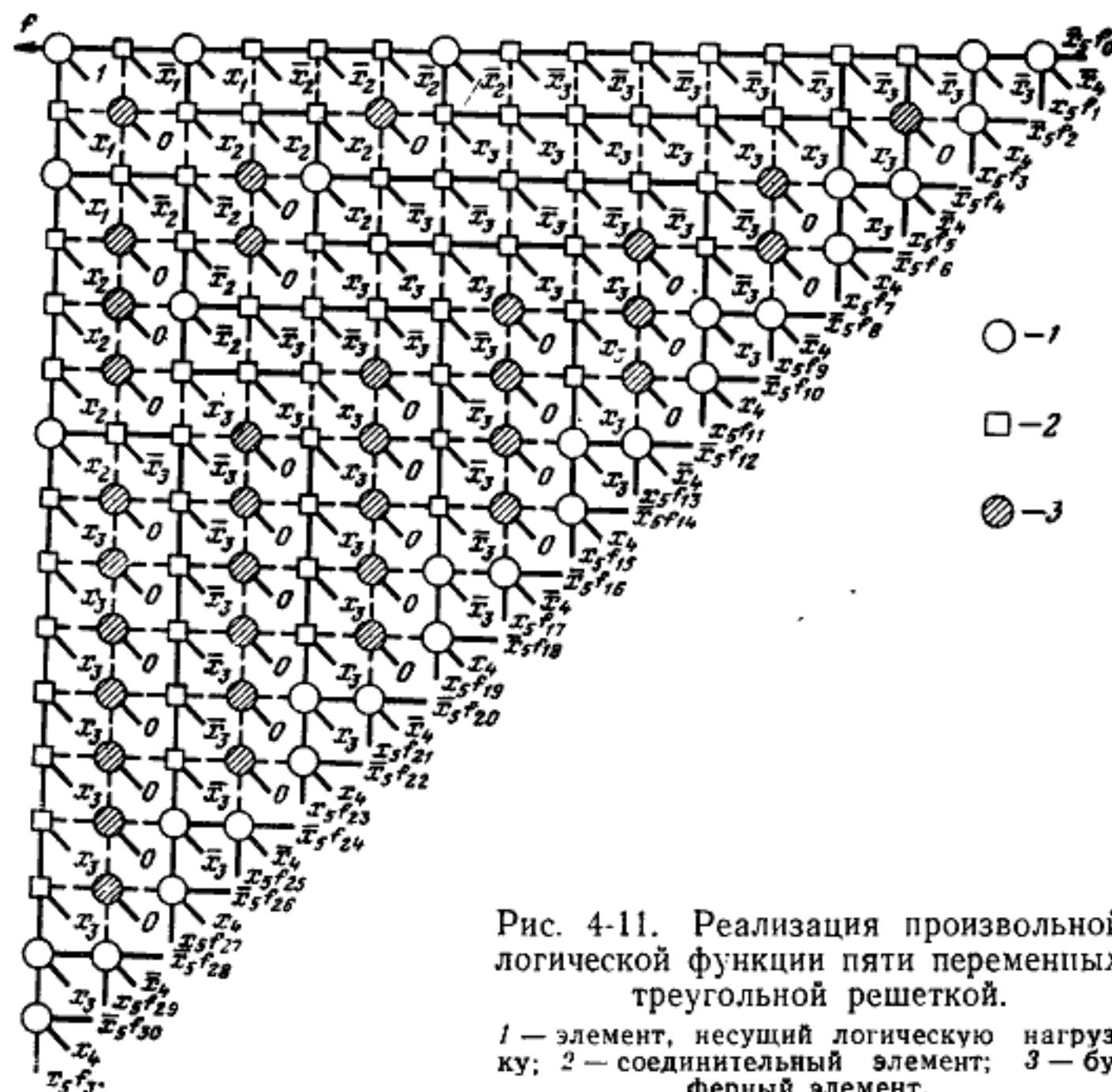


Рис. 4-11. Реализация произвольной логической функции пяти переменных треугольной решеткой.

1 — элемент, несущий логическую нагрузку; 2 — соединительный элемент; 3 — буферный элемент.

любым из следующих двух способов:

$$\left| \begin{array}{ccc|ccc} \bar{x}_1 & \bar{x}_1 & \bar{x}_1 & \bar{x}_2 & g_{111} & \bar{x}_3 \\ \bar{x}_2 & g_{011} & \bar{x}_3 & g_{100} & x_3 & g_{110} \\ g_{000} & x_3 & g_{010} & \bar{x}_3 & g_{101} & x_2 \\ \bar{x}_3 & g_{001} & x_2 & x_1 & x_1 & x_1 \end{array} \right| \text{ или}$$

$$\left| \begin{array}{c|ccc} \bar{x}_1 & \bar{x}_2 & g_{111} & \bar{x}_3 \\ \bar{x}_1 & g_{100} & x_3 & g_{110} \\ \bar{x}_1 & \bar{x}_3 & g_{101} & x_2 \\ \hline \bar{x}_2 & g_{011} & \bar{x}_3 & x_1 \\ g_{000} & x_3 & g_{010} & x_1 \\ \bar{x}_3 & g_{001} & x_2 & x_1 \end{array} \right|,$$

причем

$$f = \bigvee_{\sigma_1, \sigma_2, \sigma_3} x_1^{\sigma_1} x_2^{\sigma_2} x_3^{\sigma_3} g_{\sigma_1 \sigma_2 \sigma_3}.$$

В этих матрицах отрезками выделены исходные матрицы трех переменных. Построив решетки, соответствующие приведенным выше матрицам, можно убедиться в том, что при  $g_{000}, \dots, g_{111} \in \{x_4, \bar{x}_4, 1, 0\}$  они реализуют любые функции четырех переменных. Аналогично можно построить матрицы для  $n=k$  из двух матриц для  $n=k-1$ . При рекурсивном построении матриц для  $n>4$  можно достичь некоторого уменьшения числа элементов за счет чередования этих двух способов построения. Иначе говоря, матрица для заданного  $n$  строится таким образом, чтобы она была ближе к квадратной. Если обозначить через  $h_n$  число строк, а через  $w_n$  — число столбцов матрицы  $n$  переменных, то  $h_3=w_3=3$  и для  $n>3$ ,

$$h_n = w_{n-1} + 1,$$

$$w_n = 2h_{n-1}.$$

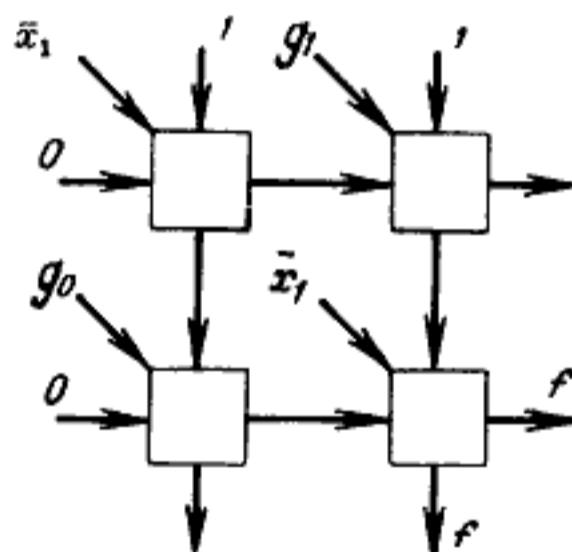


Рис. 4-12. Решетка из мажоритарных элементов, реализующая произвольную функцию двух переменных.

Размеры канонических решеток, реализующих любую логическую функцию  $n$  переменных, для малых  $n$  приведены в табл. 4-1.

«Настройка» решетки сводится к подаче вычисленных значений остаточных функций на соответствующие входы решетки, причем

$$g_i = f(x_1 = i);$$

$$g_{ij} = f(x_1 = i, x_2 = j);$$

$$g_{ijk} = f(x_1 = i, x_2 = j, x_3 = k) \text{ и т. д.}$$

Если ставится задача реализации конкретной логической функции, то в большинстве случаев решетка, соответствующая канонической матрице, не является минимальной и допускает упрощение [Л. 22].

Таблица 4-1

Число переменных	Размеры решеток	Число внешних входов <sup>1</sup>
2	2 $\times$ 2	6
3	3 $\times$ 3	10
4	6 $\times$ 4	16
5	8 $\times$ 7	26
6	14 $\times$ 9	44
7	18 $\times$ 15	78
8	30 $\times$ 19	144
9	38 $\times$ 31	274

<sup>1</sup> Указано число входов при объединении тех одноименных входов, которые являются общими при реализации всех функций данного числа переменных.

В ряде работ даны методы синтеза однородных и частично однородных структур на мажоритарных элементах: в [Л. 18, 19, 44] — синтез треугольных структур (сведением к монотонным самодвойственным функциям), в [Л. 19] — синтез прямоугольных и некоторых специальных структур, реализующих пороговые симметрические функции и функции других классов.

**2. Решетки из ненастраиваемых элементов, реализующих любую функцию двух переменных.** Рассмотренная выше решетка из мажоритарных элементов обладает тем недостатком (с точки зрения возможности ее практического использования в микроэлектронном исполнении), что число внешних входов уже при малом  $n$  достигает нескольких десятков. Поэтому предпринимались попытки в рамках той же конструкции решетки за счет увеличения логических возможностей элемента уменьшить число внешних выходов. Одна из таких попыток реализована в [Л. 9], где предлагается применить также трехходовой элемент, поведение которого описывается функцией<sup>1</sup>

$$z = x_1 g_1 \vee \bar{x}_1 g_0,$$

<sup>1</sup> В оригинале  $f = x_1 \bar{g}_1 \vee \bar{x}_1 g_0$ . Изменение позволяет обойти некоторые сложности не только в изложении, но и собственно в синтезе.

где  $x_1$  — его центральный внешний вход,  $g_1$  и  $g_0$  — соответственно левый и верхний входы (рис. 4-13, а). Элемент, следовательно, способен реализовать любую функцию двух переменных, если  $g_1, g_0 \in \{x_2, \bar{x}_2, 1, 0\}$ .

Легко показать, что решетки  $2 \times 2$  и  $4 \times 4$  из таких элементов, изображенные на рис. 4-13, а и б, могут быть

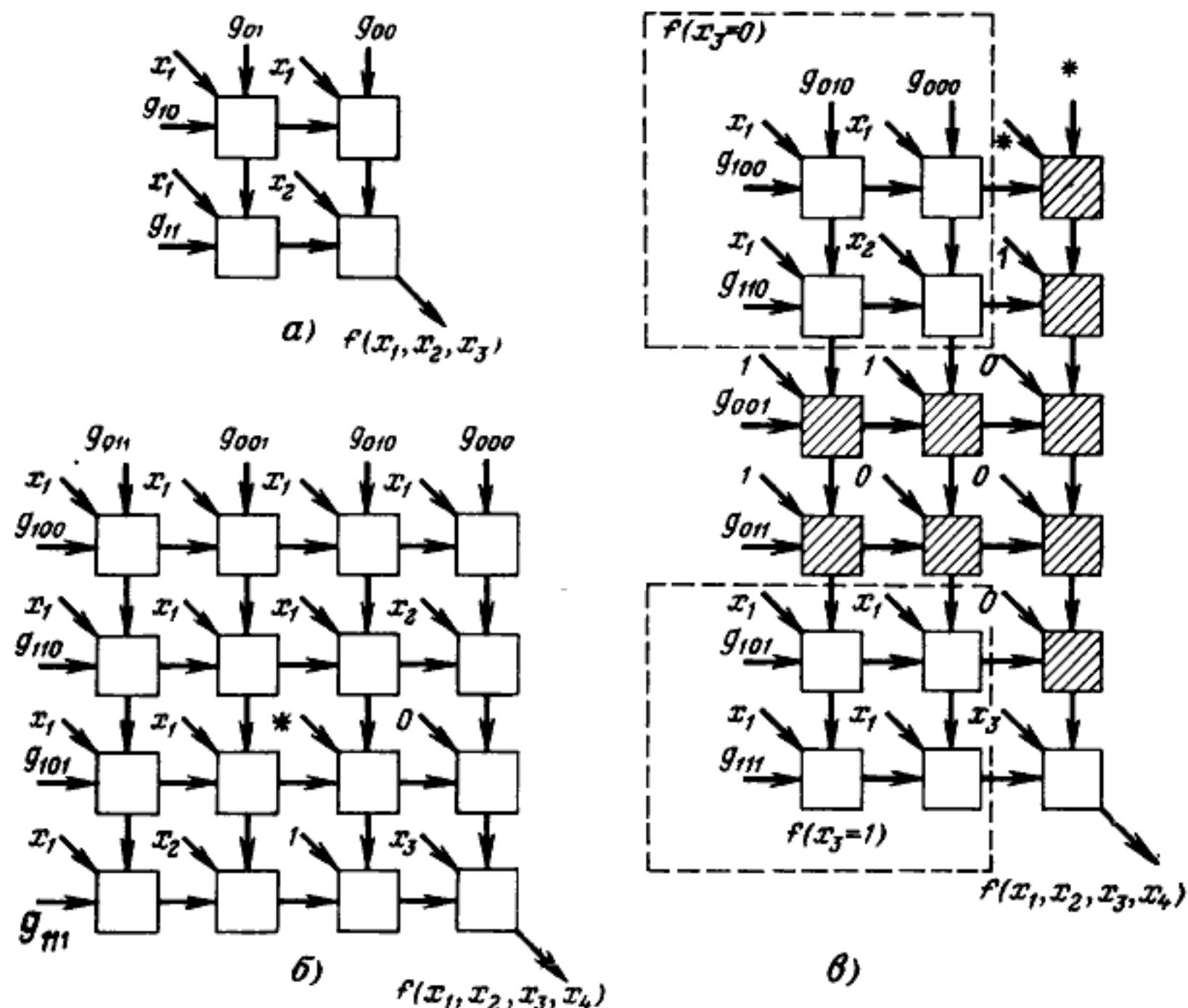


Рис. 4-13. Решетки, реализующие произвольные функции трех (а) и четырех (б, в) переменных. (Решетка в — каноническая. На входы, стмеченные звездочками, могут быть поданы любые переменные или константы.)

путем соответствующей подачи переменных и констант настроены на выполнение любой функции трех и четырех переменных:

$$f(x_1, x_2, x_3) = \bar{x}_1 \bar{x}_2 g_{00} \vee x_1 \bar{x}_2 g_{10} \vee \bar{x}_1 x_2 g_{01} \vee x_1 x_2 g_{11},$$

$$\begin{aligned} f(x_1, \dots, x_4) = & x_4 (\bar{x}_1 \bar{x}_2 g_{001} \vee x_1 \bar{x}_2 g_{101} \vee \bar{x}_1 x_2 g_{011} \vee x_1 x_2 g_{111}) \vee \\ & \vee \bar{x}_4 (\bar{x}_1 \bar{x}_2 g_{000} \vee x_1 \bar{x}_2 g_{100} \vee \bar{x}_1 x_2 g_{010} \vee x_1 x_2 g_{110}), \end{aligned}$$

где  $g_{00}, g_{10}, g_{01}, g_{11}, g_{001}, \dots, g_{111} \in \{x_3, \bar{x}_3, 1, 0\}$ .

Однако путь к канонизации синтеза указывает другой, менее экономичный способ реализации произвольной функции четырех переменных на структуре  $6 \times 3$  (рис. 4-13,в). Два куска структуры размером  $2 \times 2$  реализуют функции трех переменных  $f(x_3=1)$  и  $f(x_3=0)$ . Значительная часть элементов структуры (они заштрихованы) функциональной нагрузки не несет и используется лишь для образования соединительных каналов. Подробное описание процедуры образования решеток для  $n > 3$  разложением заданной функции по  $n-3$  переменным читатель найдет в [Л. 9]. Здесь мы приведем таблицу, в которой указано число внешних входов решетки и размеры ее для различных  $n$  (табл. 4-2).

Таблица 4-2

Число переменных $n$	Размер решетки $(2^{n-1}-4)(n-1)$ , $n > 4$	Число выходов <sup>1</sup> $2^{n-1}+n+1$ , $n > 4$
2	$1 \times 1$	3
3	$2 \times 2$	6
4	$4 \times 4$	13
5	$12 \times 4$	22
6	$28 \times 5$	39
7	$60 \times 6$	72
8	$124 \times 7$	135
9	$252 \times 8$	266

<sup>1</sup> Указано число входов при объединении тех одноименных входов, которые являются общими при реализации всех функций данного числа переменных (например, всех входов на рис. 4-13,в).

Сравнение табл. 4-1 и 4-2 говорит о том, что при  $n > 5$  число элементов в решетках из рассматриваемых элементов больше, чем в решетках из мажоритарных элементов. К тому же недостатком рассмотренных структур является их «неквадратность». Однако число входов (с учетом объединения одноименных) с ростом  $n$  увеличивается медленнее, чем в решетках из мажоритарных элементов.

Следует заметить, что понятие функциональной полноты, обычно принятное в теории переключательных схем, в данном случае неприменимо. Дело в том, что при синтезе пирамидальных схем элементы, образующие функ-

ционально полный базис, могут соединяться каналами неограниченной длины, и выход одного элемента может подключаться к произвольному числу входов других элементов. Иначе говоря, заданная функция представляется в виде произвольной суперпозиции элементарных функций, реализуемых базисными элементами. Однородность схем накладывает ограничения на виды суперпозиции. Поэтому следует говорить о функциональной полноте (т. е. универсальности) структуры, а не только набора ее элементов. В [Л. 8] предлагается все элементы структуры разделять на функциональные (т. е. несущие логическую нагрузку) и соединительные (необходимые для передачи сигналов и организации пересечения путей их передачи) и доказывается теорема о том, что любая логическая функция может быть реализована решеткой из функционально полного набора элементов и двух типов соединительных элементов.

Далее функциональная полнота решеток рассматриваемых типов будет доказываться конструктивным способом, т. е. для каждого вида решетки будет предлагааться метод реализации заданной функции, из которого вытекает, что любая функция реализуется этим методом. Методы синтеза рассматриваемых ниже типов решеток являются по существу попытками приспособления известных способов представлений и разложений логических функций (нормальные формы, разложения по Шенону и Риду).

#### **4-3. Однородные решетки из настраиваемых элементов, реализующих некоторые функции двух переменных**

Рассмотрим двумерную решетку типа рис. 4-14, которая отличается от двумерных сетей, исследовавшихся в предыдущих параграфах, двумя особенностями:

а) она составлена из настраиваемых элементов с двумя боковыми входами и двумя выходами. Условное обозначение элемента приведено на рис. 4-15. Его таблица истинности приведена на стр. 130.

Левый боковой выход элемента повторяет значение левого бокового входа, значение второго (нижнего) выхода  $z$  определяется в соответствии со значениями входов настройки  $a, b, c, d$ , каждой из восьми комбинаций которых приписан определенный индекс. Отметим, что на-

стройка элемента по индексу 13 делает его триггером (с раздельными входами), что бывает необходимо для запоминания промежуточных результатов;

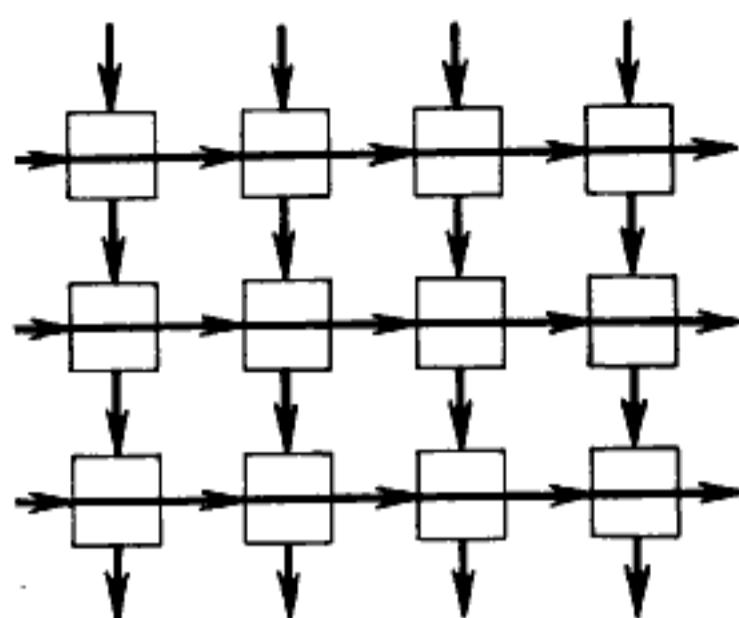


Рис. 4-14. Однородная решетка из настраиваемых элементов.

Рис. 4-14. Однородная решетка из настраиваемых элементов.

Рис. 4-15 можно установить, что один столбец матрицы, являющийся одномерной сетью, может реа-

лизовать класс функций, совпадающий с классом, реализуемым повторными одномерными сетями. Таким образом, при синтезе одномерной сети на элементах типа рис. 4-15 следует пользоваться изложенными в § 1-2,

Индекс	$a$	$b$	$c$	$d$	$z$
0	0	0	0	0	1
1	0	0	0	1	$\bar{y}$
2	0	0	1	0	$\bar{x} \vee \bar{y}$
3	0	0	1	1	$\bar{x}\bar{y}$
4	0	1	0	0	$x \vee y$
5	0	1	0	1	$x\bar{y}$
6	0	1	1	0	$x \oplus y$
7	0	1	1	1	0
13	1	1	0	1	$\bar{y}(x \vee z)$

1-3 алгоритмами и применять правило де Моргана, выбирая те варианты схем, которые позволяют подавать на сеть лишь переменные в прямом виде и использовать те режимы элементов, которые соответствуют имеющимся индексам настройки. На рис. 4-16 дан вариант решения примера на стр. 27 на элементах типа рис. 4-15.

б) на левые граничные входы подаются не константы, а переменные, которые прошивают элементы насеквоздь. Иначе говоря, значение правого бокового выхода элемента совпадает со значением левого бокового входа.

По рис. 4-15 можно установить, что один столбец матрицы, являющейся одномерной сетью, может реа-

Легко показать, что при помощи соответствующей настройки решетки типа рис. 4-14 могут реализовать любые логические функции. Под настройкой решетки

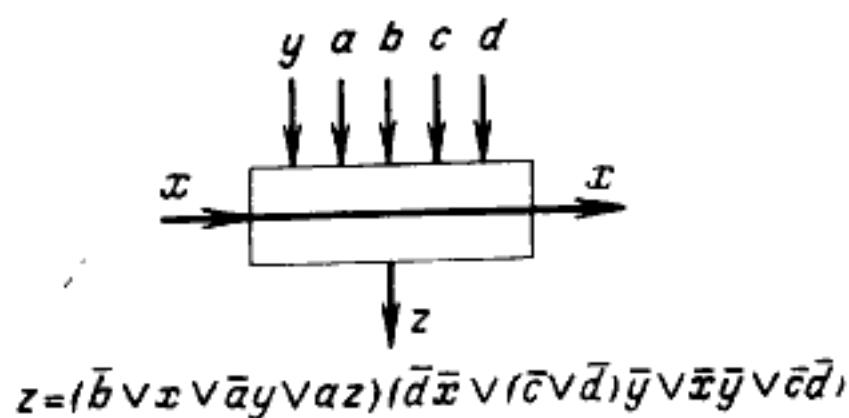


Рис. 4-15. Элемент сети типа рис. 4-14.

здесь понимается, во-первых, настройка всех ее элементов и, во-вторых, соответствующая коммутация внешних входов и выходов.

Стандартным приемом синтеза решетки по заданной функции (если она не реализуется одномерной бесповторной или повторной сетью) является последовательное исключение переменных:

$$f(x_1, \dots, x_n) = x_i f(x_i=1) \vee \bar{x}_i f(x_i=0).$$

Это разложение приводит к решетке, изображенной на рис. 4-17, а, в которой выход левого столбца соединен со входом нижнего ряда элементов. Дальнейшая задача сводится к реализации функции  $\bar{f}(x_i=1)$  и  $\bar{f}(x_i=0)$ . Эти функции представляются как одномерные бесповторные или повторные сети, если это возможно. При этом желателен такой выбор дальнейшего порядка исключения переменных, чтобы длина сетей для  $\bar{f}(x_i=1)$  и  $\bar{f}(x_i=0)$  совпадала. Иногда с целью их выравнивания бесповторную сеть следует реализовать как повторную или использовать избыточные сети. Необхо-

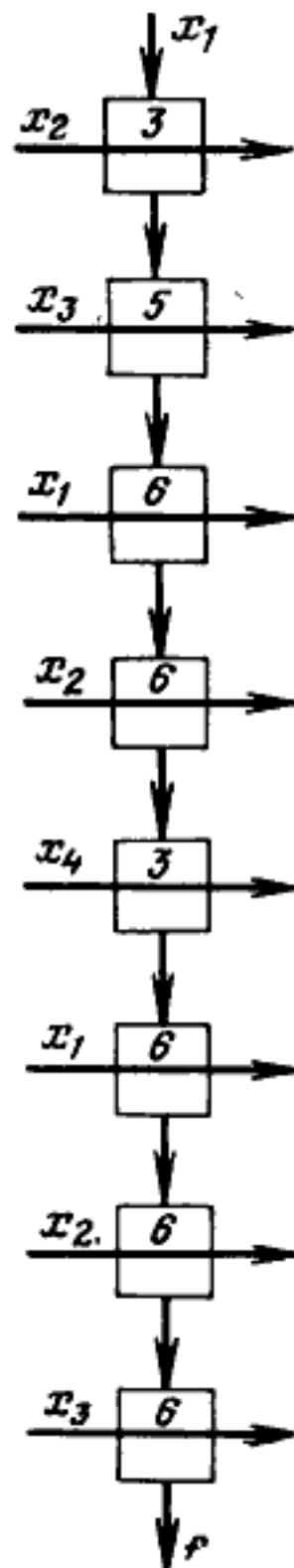


Рис. 4-16. Пример синтеза сети на элементах типа рис. 4-15.

димость таких преобразований читатель уяснит, самостоятельно прорешав следующий пример.

Дана функция

$$f(x_1, \dots, x_n) = \bar{x}_1 x_2 (x_3 \oplus \bar{x}_4) \vee \bar{x}_1 \bar{x}_2 (\bar{x}_3 \vee \bar{x}_4).$$

Построить решетку типа рис. 4-17; первой пусть исключается переменная  $x_2$ . Имеем:

$$f(x_2=1) = \bar{x}_1 (x_3 \oplus \bar{x}_4), \quad \bar{f}(x_2=1) = x_1 \vee (x_3 \oplus x_4).$$

$$f(x_2=0) = \bar{x}_1 (\bar{x}_3 \vee \bar{x}_4), \quad \bar{f}(x_2=0) = x_1 \vee x_3 x_4.$$

Остаточные функции  $\bar{f}(x_2=1)$  и  $\bar{f}(x_2=0)$  реализуются одномерными сетями. Два варианта решения представлены на рис. 4-17, б, в.

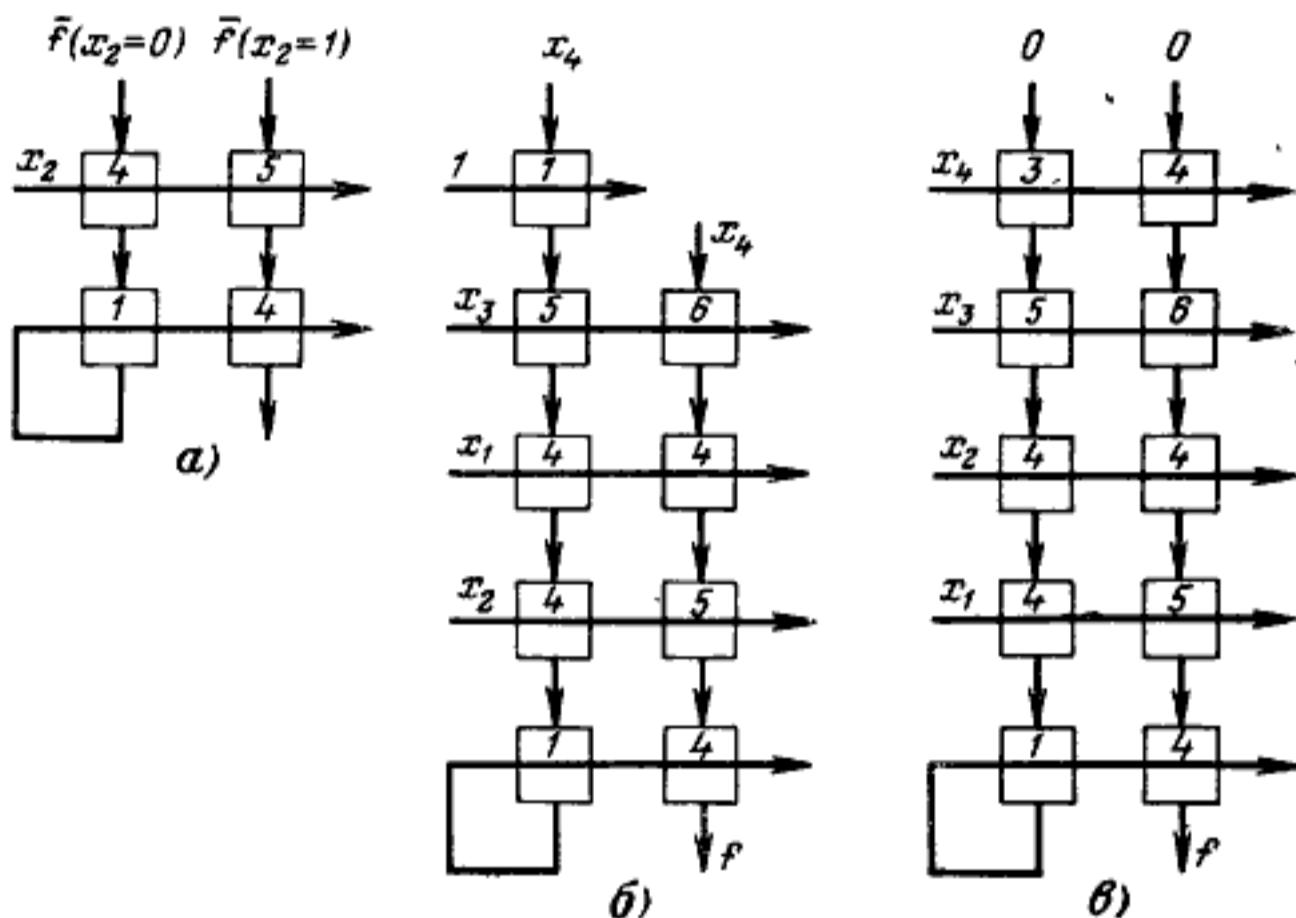


Рис. 4-17. Решение примера.

Разложение повторяется до тех пор, пока каждая из полученных остаточных функций не окажется реализуемой одномерной бесповторной или повторной сетью.

Возможная минимизация решеток по числу элементов вытекает не только из варьирования порядка исключения переменных, но и из использования свойств функциональной разделимости булевых функций.

В [Л. 42] приведена таблица оптимальных реализаций логических функций трех переменных схемами типа рис. 4-14.

Этим схемам свойствен большой расход элементов, усугубляемый тем, что часть элементов не несет логиче-

ской нагрузки вследствие необходимости введения внешних перемычек. Например, из рис. 4-17,а видно, что левый нижний элемент решетки не несет логической нагрузки. При совместной реализации нескольких функций или при вынужденном одновременном исключении нескольких переменных число таких элементов быстро растет с ростом числа переменных и числа совместно реализуемых функций.

#### 4-4. Решетки типа «паутинка»

Уже говорилось, что усложнение настраиваемого элемента, увеличение его логических возможностей, а также усложнение рисунка связей между элементами приводит к меньшему расходу элементов.

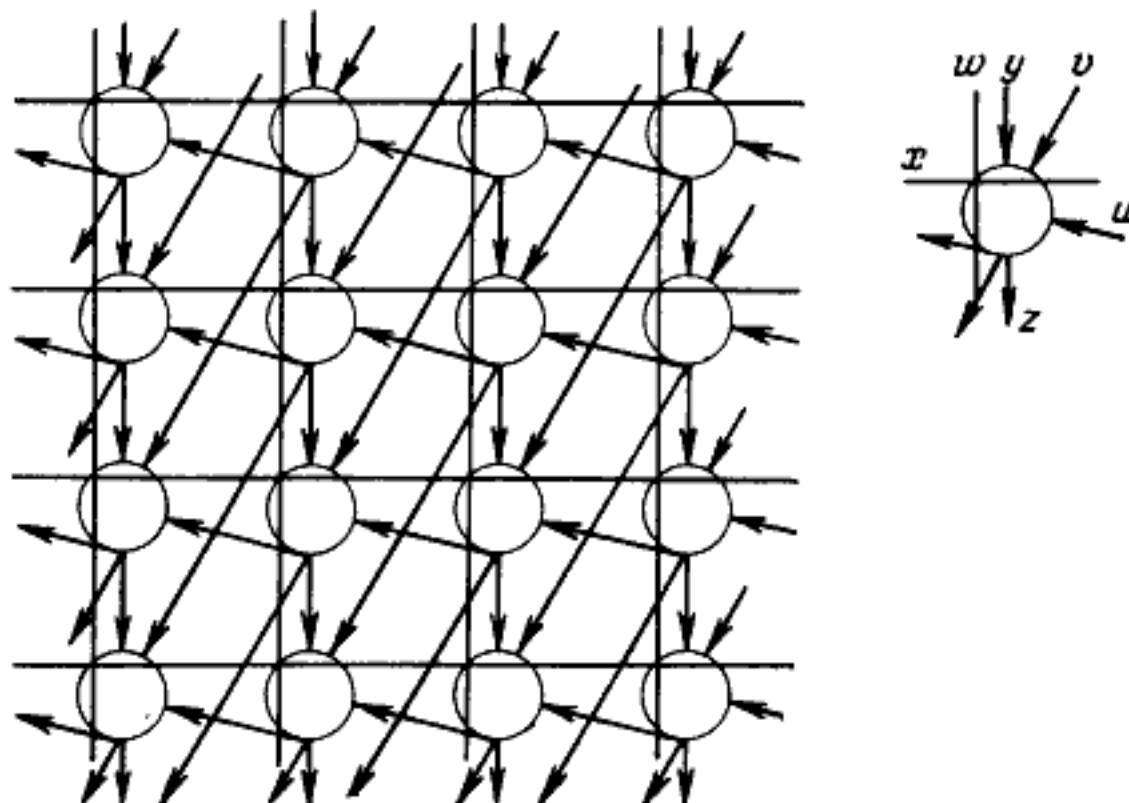


Рис. 4-18. Решетка типа «паутинка» и ее элемент.

В решетке типа «паутинка» (рис. 4-18) используются оба эти рычага. Каждый элемент решетки имеет три идентичных выхода и пять входов: два от вертикальной и горизонтальной шин и три от соседних элементов (двух непосредственных соседей сверху и справа и одного, расположенного справа вверху по ходу шахматного коня<sup>1</sup>). Помимо этих входов, каждый элемент снабжен дополнительными настроочными входами, позволяющими:

- использовать элемент в одном из режимов таблицы на стр. 130 (по отношению ко входам  $x$  и  $y$ );
- использовать его в таком режиме, что горизонтальный сигнал элемента определяется дизъюнкцией  $x \vee y$ , а вертикальный —  $u \vee v \vee w$ ;

<sup>1</sup> Наличие этой связи позволяет использовать некоторую промежуточную функцию, сформированную в части соседнего столбца решетки.

- в) коммутировать некоторые входы и выходы элемента;  
 г) изолировать элементы решетки, благодаря чему возможно проведение поэлементных испытаний.

Если заданная функция не реализуется одномерной схемой, то синтез соответствующей решетки может быть произведен исключением переменных в соответствии с одним из разложений:

$$f(x_1, \dots, x_n) = x_i f(x_i = 1) \vee \bar{x}_i f(x_i = 0) \text{ или}$$

$$f(x_1, \dots, x_n) = f(x_i = 0) \oplus x_i (f(x_i = 0) \oplus f(x_i = 1)),$$

что может быть выполнено так, как показано на рис. 4-19. Индексом  $j$  обозначено, что элемент логически не активен и используется лишь для коммутации. Кружок (треугольник) на входе означает, что данный вход является горизонтальным (вертикальным) входом элемента на рис. 4-15. Кружки на входе и выходе одного и того же элемента с индексом  $j$  означают, что вход  $y$  и выход  $z$  этого элемента замкнуты.

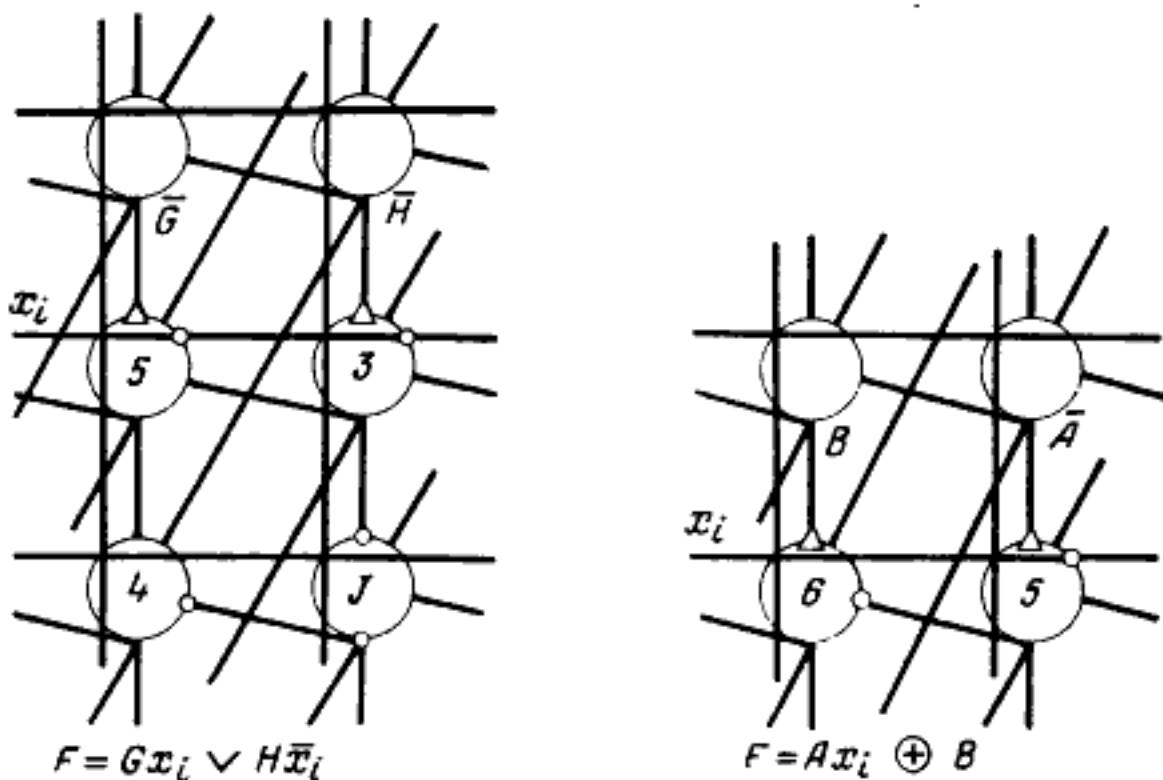


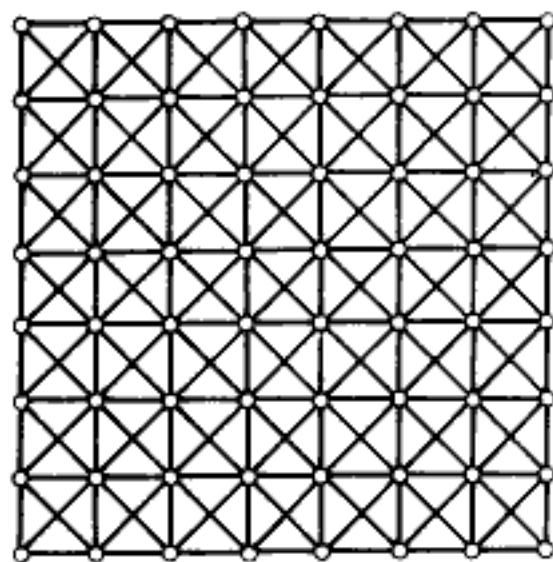
Рис. 4-19. Исключение переменной на решетке типа «паутинка».

В [Л. 41, 42] приведен ряд примеров синтеза некоторых вычислительных схем и устройств: трехразрядного параллельного сумматора, пятиразрядного регистра сдвига и множительного устройства размером  $27 \times 16$  элементов. Они демонстрируют ряд достоинств решеток типа «паутинка» по сравнению с решетками типа рис. 4-14, а именно: уменьшение числа элементов; увеличение процента элементов, несущих логическую нагрузку; отсутствие необходимости установки внешних перемычек.

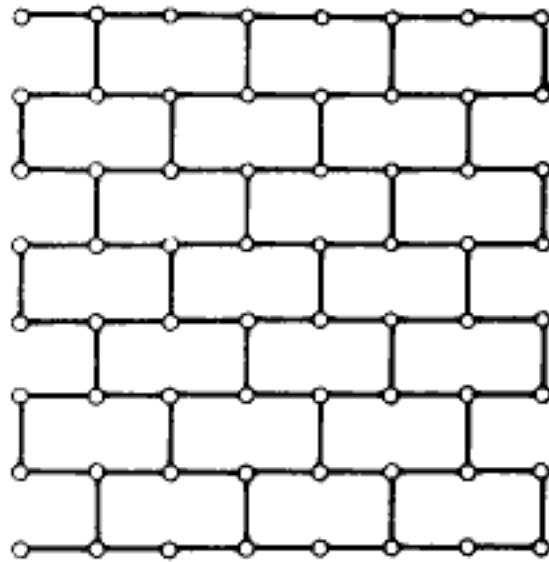
Одновременно эти сети имеют и недостатки, связанные с усложнением элемента, значительным увеличением числа выводов решетки (особенно за счет настроечных входов) и, главное, с необходимостью соответствующей настройки всех элементов решетки.

#### 4-5. Решетки типа «алмаз»

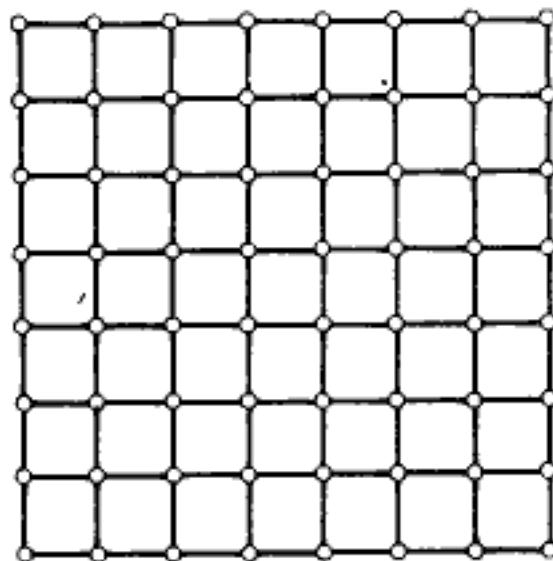
Естественно ожидать, что усложнение рисунка межэлементных связей позволит увеличить потенциальные возможности структур, поскольку каждый элемент может обмениваться информацией с большим числом со-



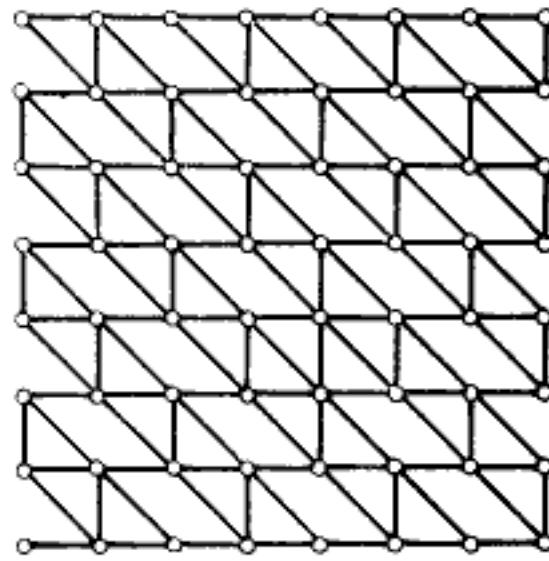
а)



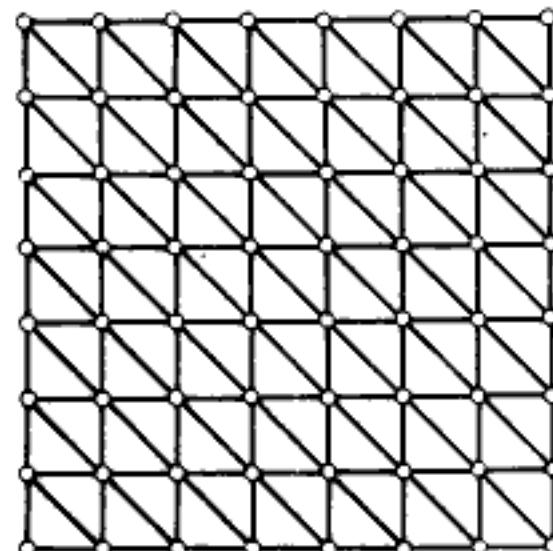
б)



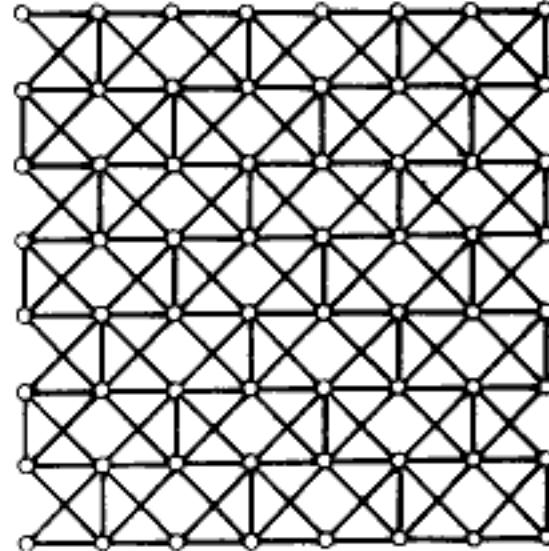
в)



г)

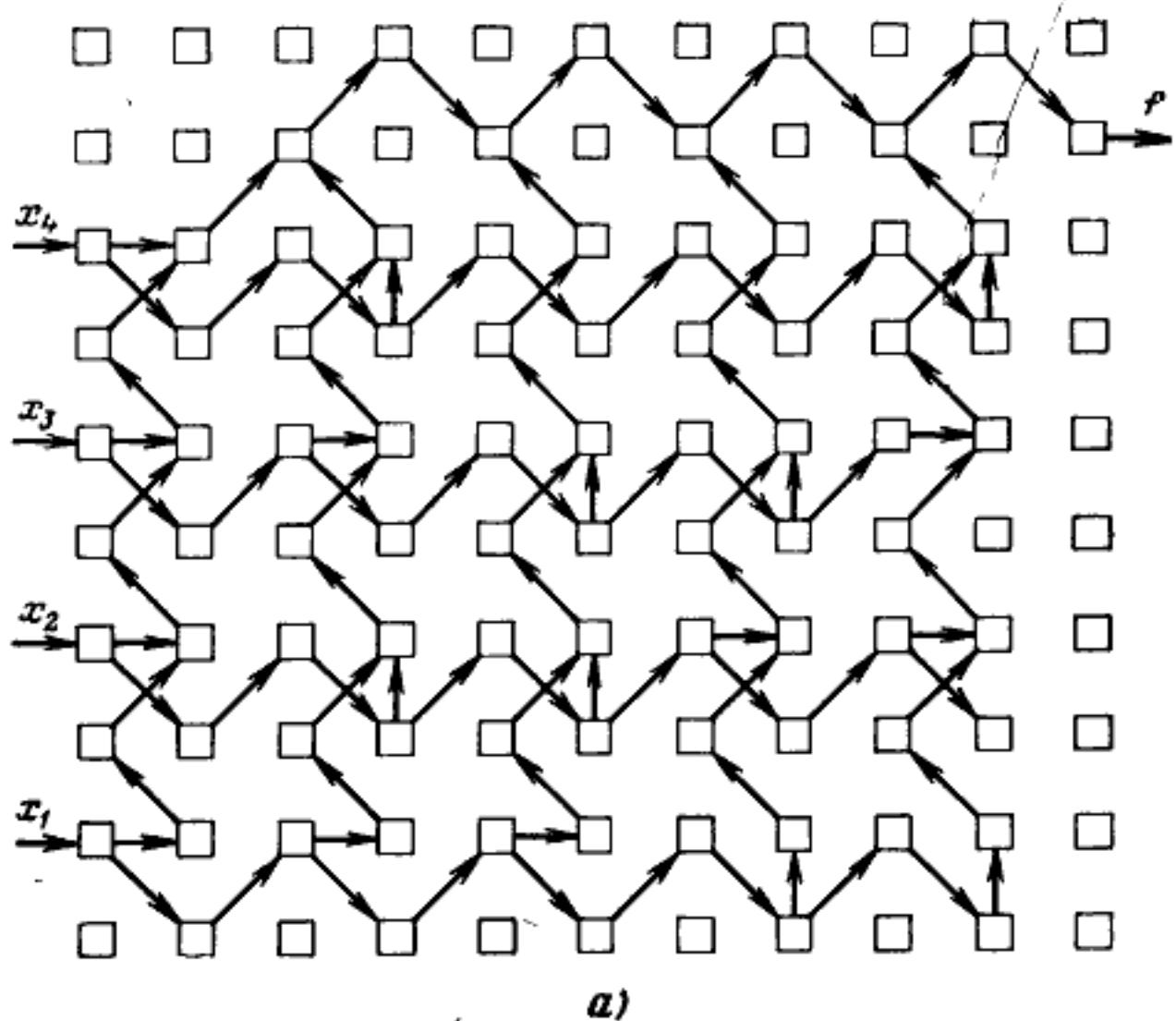


д)

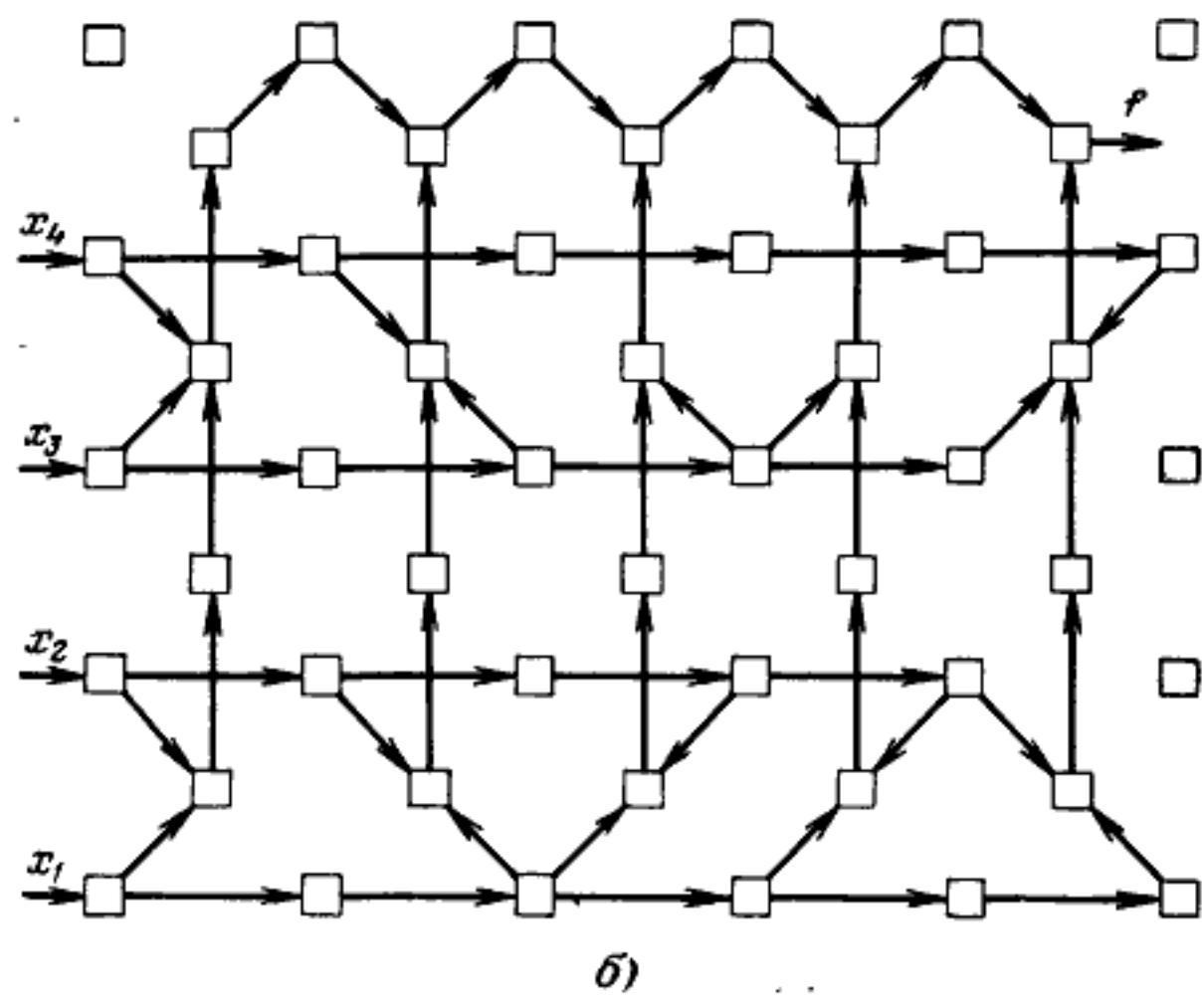


е)

Рис. 4-20. Некоторые возможные графы связей в плоской прямоугольной решетке.



*a)*



*б)*

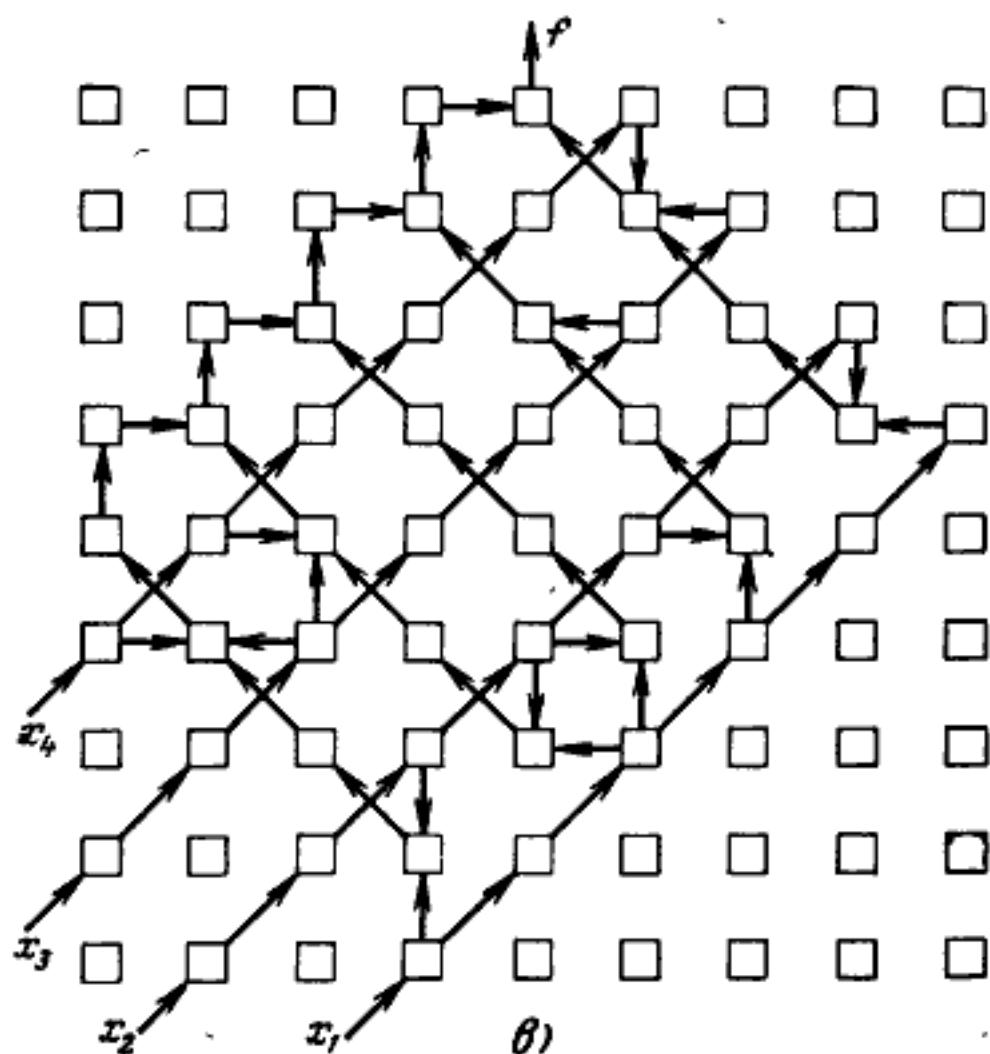
Рис. 4-21. Реализация функции четырех переменных на

седей. При этом возникает возможность передачи межэлементных сигналов по многим направлениям, часть из которых используется как соединительные (буферные), а часть — для реализации логических преобразований.

Графы связей между элементами двумерной решетки, изображенные на рис. 4-20, б—е, являются подграфами графа рис. 4-20, а. Каждый элемент решетки, соответствующий вершине графа на рис. 4-20, а, связан с восемью соседями.

Пусть в качестве элемента решетки с графиком связей типа рис. 4-20, а используется элемент в восемь выходами, причем некоторые выходы  $z_i$  осуществляют лишь инверсию входного сигнала  $x_i$  ( $z_i = \bar{x}_i$ ), в то время как другие выходы реализуют функцию от двух до восьми входных переменных. Например,  $z_j = x_{i_1} \vee x_{i_2} \vee \dots \vee x_{i_k}$  или  $z_j = \overline{x_{i_1} x_{i_2} \dots x_{i_k}}$ , где  $k = 2 \div 8$ .

Метод синтеза решеток такого типа (они получили название «решетки типа алмаз») заключается в формировании импликантов минимальной (тупиковой) нормальной формы и последующего их объединения.



решетках типа «алмаз», все элементы — типа НЕ-И.

Ниже даны примеры реализации одной и той же функции четырех переменных  $f(x_1, \dots, x_4) = \bar{x}_1\bar{x}_2x_3\bar{x}_4 \vee \bar{x}_1x_2\bar{x}_3x_4 \vee \bar{x}_1x_2x_3 \vee x_1\bar{x}_2x_3 \vee x_1\bar{x}_2\bar{x}_3x_4$  на решетках типа «алмаз», отличающихся способом выбора горизонтальных и вертикальных путей передачи переменных на соответствующие элементы решетки, а также способом выбора входов и выходов элементов, несущих логическую нагрузку (рис. 4-21). Из этих рисунков ясен способ синтеза. Остается определить максимальный расход элементов:

для рис. 4-21,*a* —  $2(n-1)(2^n-1) \approx n2^{n+1}$ ;

для рис. 4-21,*б* —  $(2n+1)2^{n-1} + n(2n+1) \approx n2^n$ ;

для рис. 4-21,*в* —  $(1,5n-1)2^{n-1} + 2^n \approx 3n2^{n-2}$ .

Для каждого из видов решеток, хотя функция элемента фиксирована, схема остается гибкой, так как можно менять связи. Часть связей в процессе настройки решетки приходится уничтожать. Хотя физически процесс настройки решетки в этом случае отличается от процесса настройки элементов, тем не менее по функциональным возможностям эти процессы аналогичны. Ведь конструкцию настраиваемых элементов можно сделать такой, чтобы выбирать лишь те связи между элементами, которые необходимы для заданной настройки элементов. Поэтому решетки типа «алмаз» следует отнести к классу однородных структур с настраиваемыми элементами. Недостатком этих решеток также является их неэкономичность.

В [Л. 8, 16] широко освещены вопросы построения однородных решеток («вычислительных сред»), являющихся по нашей классификации двумерными двусторонними сетями, с тем ограничением, что переменные подаются лишь на элементы периметра решетки. Настройка решетки на реализацию заданной вычислительной схемы производится: а) настройкой собственно элементов среды; б) выбором определенных связей между элементами; в) соответствующей коммутацией входных переменных. Выбор связей (замкнуто, разомкнуто) между элементами производится установкой четырех триггеров каждого элемента, что в принципе эквивалентно выбору типа соединения с помощью удаления лишних связей. Основная задача синтеза, которая здесь возникает, сводится к на-

ложению пирамидальной реализации на участок вычислительной среды и выработке соответствующих правил настройки.

#### 4-6. Реализация автоматов в однородных сетях<sup>1</sup>

При решении задачи о построении однородных двумерных сетей, реализующих произвольные конечные автоматы, следует сделать предположение о том, что число состояний элемента системы не должно зависеть от характеристик реализуемого автомата.

Рассмотрим пример. Пусть задан конечный автомат  $W$ , имеющий 4 состояния  $w_1, w_2, w_3, w_4$ , входной алфавит которого есть  $X = \{x_1, x_2, x_3\}$ , выходной —  $Z = \{z_1, z_2, z_3\}$ , начальное состояние —  $w_1$ , и функции переходов  $\delta$  и выходов  $\lambda$  заданы следующей таблицей:

	$x_1$	$x_2$	$x_3$		$x_1$	$x_2$	$x_3$
$w_1$	$w_4$	$w_2$	$w_4$	$w_1$	$z_1$	$z_3$	$z_2$
$w_2$	$w_1$	$w_1$	$w_2$	$w_2$	$z_2$	$z_1$	$z_3$
$w_3$	$w_4$	$w_3$	$w_1$	$w_3$	$z_1$	$z_2$	$z_1$
$w_4$	$w_3$	$w_1$	$w_2$	$w_4$	$z_3$	$z_2$	$z_1$

Построим элемент и однородную сеть из этих элементов, которая будет реализовать автомат  $W$ . Прежде всего введем новую функцию  $\delta^*$ :

	$x_1$	$x_2$	$x_3$
$w_1$	$3 \downarrow$	$1 \downarrow$	$3 \downarrow$
$w_2$	$1 \uparrow$	$1 \uparrow$	$0 \downarrow$
$w_3$	$1 \downarrow$	$0 \downarrow$	$2 \uparrow$
$w_4$	$1 \uparrow$	$3 \uparrow$	$2 \uparrow$

Если автомат  $W$  из состояния  $w_i$  под действием символа  $x_j$  переходит в состояние  $w_k$ , то на пересечении  $i$ -й строки и  $j$ -го столбца поместим абсолютную величину разности  $|k-i|$  и одну из стрелок

<sup>1</sup> Этот параграф написан А. Я. Макаревским. Соответствующие результаты более подробно освещены в следующих публикациях:

1. Макаревский А. Я. Реализация дискретных управляющих устройств в однородных средах. — Институт проблем управления (ИАТ), М., 1970.

2. Макаревский А. Я. О моделировании в однородных средах. — «Кибернетика», 3, 1970.

3. Макаревский А. Я. Реализация отображений клеточными автоматами. — «Кибернетика», 6, 1970.

4. Макаревский А. Я. О реализации автоматов в однородных средах. — «Вычислительные системы», Новосибирск, вып. 41, 1971.

$\uparrow$  или  $\downarrow$  в качестве указателя знака разности  $k-i$ , а именно ставим  $\downarrow$ , если  $k \geq i$ , и  $\uparrow$ , если  $k < i$ . Например,  $\delta(w_1, x_1) = w_4$  и  $\delta^*(w_1, x_1) = 3\downarrow$ . Таким образом, чтобы найти в таблице следующее состояние, нужно подняться или опуститься в зависимости от направления стрелки на столько строк, чему равно число в соответствующей клетке таблицы.

Возьмем сеть, состоящую из 24 элементов, расположенных в 4 строках и 6 столбцах (рис. 4-22). Элемент сети будем описывать постепенно, по мере объяснения идеи ее работы. Элемент имеет три группы состояний: состояния вида  $(+, x_i)$  и  $(-, x_i)$ ,  $x_i \in X$ , состояния вида  $(+, pr)$  и  $(-, pr)$ , где  $pr \in \{\downarrow, \uparrow\}$  и  $n$  — целое неотрица-

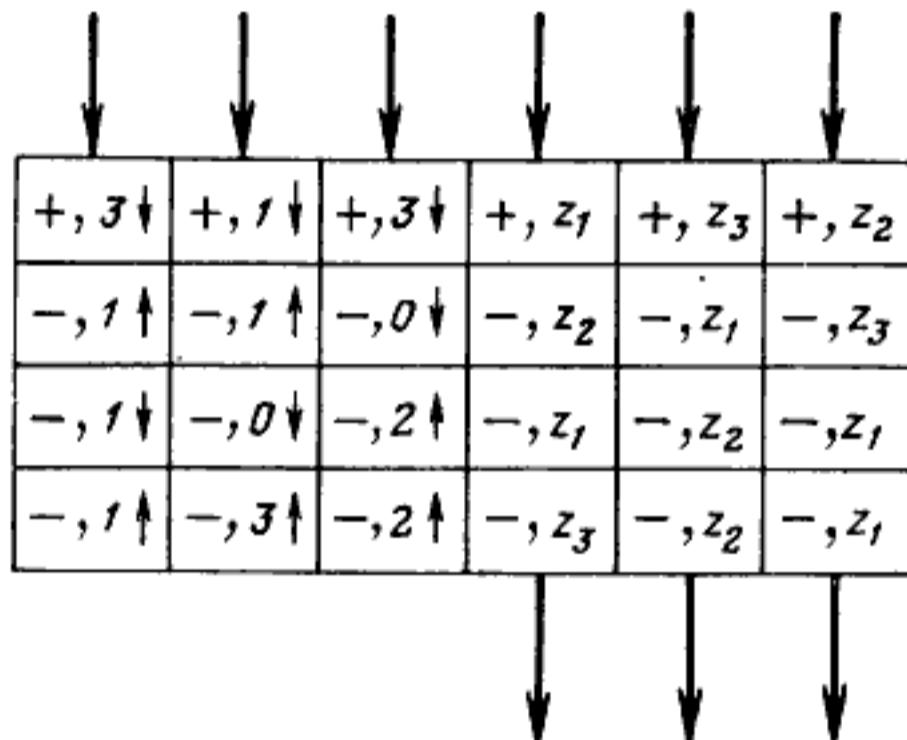


Рис. 4-22. Однородная среда.

тельное число, и состояния вида  $(mp_1, np_2)$ ,  $p_1, p_2 \in \{\downarrow, \uparrow\}$ . Состояния вида  $(+, \dots)$  будем называть активными, а состояния вида  $(-, \dots)$  — пассивными. Состояния элементов, расположенных на пересечении  $i$ -й строки и  $j$ -го столбца, в начальный момент показаны на рис. 4-22. Таким образом, в элементах, расположенных в правых трех столбцах, записана функция выходов  $\lambda$ , а в левых трех столбцах записана таблица функции  $\delta^*$ . Элементы, находящиеся в первой строке (начальные состояния  $w_1$ ), имеют активные состояния, а все остальные элементы — пассивные. Каждому состоянию автомата  $W$  соответствует строка сети, и каждому столбцу таблиц для  $\lambda$  и  $\delta^*$  соответствует столбец сети. Далее закодирует символы входного алфавита  $X$  следующим образом. Символ  $x_1$  кодируется набором  $(2, 1, 1, 2, 1, 1)$ , символ  $x_2$  — набором  $(1, 2, 1, 1, 2, 1)$ , символ  $x_3$  — набором  $(1, 1, 2, 1, 1, 2)$ . Внешними входами сети являются входы элементов, расположенных в первой строке, а внешними выходами — выходы элементов, расположенных в трех правых столбцах последней строки (см. рис. 4-22). На выходе сети будем получать наборы вида  $(z_i, 0, 0)$ ,  $(0, z_i, 0)$  и  $(0, 0, z_i)$ , которые будем кодировать как  $z_1, z_2, z_3$ .

Автомат  $W$  под действием символа  $x_1$  переходит из начального состояния  $w_1$  в состояние  $w_4$  и выдает символ  $z_1$ . Подадим на внеш-

ние входы сети код символа  $x_1$  — набор  $(2,1,1,2,1,1)$  и опишем работу сети.

Элементы  $1,2; 1,3; 1,5$  и  $1,6$  под действием сигнала  $1$  входного набора перейдут в пассивные состояния. Элемент  $1,4$  под действием символа  $2$  входного набора передает вниз сигнал  $z_1$  и переходит в пассивное состояние  $(-, z_1)$ . Сигнал  $z_1$  распространяется вниз, пока не появится на выходе элемента  $4,4$ . Состояния элементов  $2,4; 3,4$  во время передачи вниз сигнала  $z_1$  не меняются.

Элемент  $1,1$  под действием символа  $2$  входного набора выдает вниз сигнал  $2\downarrow$ , т. е. команду «сдвинуться вниз на 2 строки», и переходит в состояние  $(-, 3\downarrow)$ . Элемент  $2,1$  под воздействием сигнала  $2\downarrow$  переходит в состояние  $(1\uparrow, 2\downarrow)$ . Так как вторая координата не есть  $0\downarrow$ , то элемент  $2,1$  выдает вниз сигнал  $1\downarrow$  и возвращается в состояние  $(-, 1\uparrow)$ . Элемент  $3,1$  работает точно так же и выдает вниз сигнал  $0\downarrow$ , который переводит элемент  $4,1$  в состояние  $(1\uparrow, 0\downarrow)$ . На этот раз вторая координата состояния есть  $0\downarrow$ , и элемент  $4,1$  переходит в состояние  $(+, 1\uparrow)$  и вырабатывает сигналы, которые, распространяясь в четвертой строке, переводят все элементы в этой строке в активные состояния. Сеть готова к переработке следующего набора.

При подаче следующего набора элементы  $1,1; 1,2; \dots; 1,6; 2,1; \dots; 2,6; 3,1; \dots; 3,6$  пропускают символы  $1$  и  $2$  вниз, при этом состояния этих элементов не меняются. На этот раз в части сети, реализующей функцию переходов  $\delta^*$ , будет происходить передача сигналов вверх. Она происходит точно так же, как и описанная передача вниз.

Очевидно можно построить элемент, работающий описанным выше образом. Однако в сетях из таких элементов можно будет реализовать лишь автоматы с ограниченным числом состояний и фиксированными входным и выходным алфавитами. Поэтому закодируем числа и символы выходного алфавита  $Z$  автомата  $W$  двоичными наборами соответствующей длины (в данном случае достаточно наборов длины 2):

	$x_1$	$x_2$	$x_3$		$x_1$	$x_2$	$x_3$
$w_1$	$11\downarrow$	$01\downarrow$	$11\downarrow$	$w_1$	$01$	$11$	$10$
$w_2$	$01\uparrow$	$01\uparrow$	$00\downarrow$	$w_2$	$10$	$01$	$11$
$w_3$	$01\downarrow$	$00\downarrow$	$10\uparrow$	$w_3$	$01$	$10$	$01$
$w_4$	$01\uparrow$	$11\uparrow$	$01\uparrow$	$w_4$	$11$	$10$	$01$

Построим новый элемент. Он имеет три группы состояний: состояния вида  $(+, p)$ ,  $(-, p)$ ,  $p \in \{0,1\}$  для реализации функций выходов  $\lambda$ ; состояния  $(+, *)$ ,  $(-, *)$  и состояния вида  $(p)$ , где  $p \in \{0,1,\uparrow,\downarrow\}$ , служащие для записи функции  $\delta^*$ , и вспомогательные состояния вида  $(p_1, p_2)$ , где  $p_1, p_2 \in \{0,1,\uparrow,\downarrow\}$ . Теперь заменим каждый из элементов (рис. 4-22) одномерной цепочкой из новых элементов (рис. 4-23). Элементы с координатами  $i, j$ ,  $1 \leq i \leq 4$ ,  $4 \leq j \leq 6$  заменяются цепочками из двух элементов (для записи кодов символов  $z_i$ ), а элементы с координатами  $i, j$ ,  $1 \leq i \leq 4$ ,  $1 \leq j \leq 3$  — цепочками из четырех элементов. При этом самый левый в цепочке элемент используется для записи  $*$ , определяющей нач-

ло цепочки, второй и третий элементы используются для записи двоичного кода соответствующего числа в таблице для  $\delta^*$ . а в четвертом элементе запоминается одна из стрелок  $\downarrow$ ,  $\uparrow$ . Состояния элементов сети в начальный момент показаны на рис. 4-23. Символы  $x_1, x_2, x_3$  кодируются следующим образом:  $x_1$  — набором  $(2,1,1,2,2,1,1,1,1)$ ,  $x_2$  — набором  $(1,2,1,1,2,2,1,1)$ ,  $x_3$  — набором  $(1,1,2,1,1,1,2,2)$ .

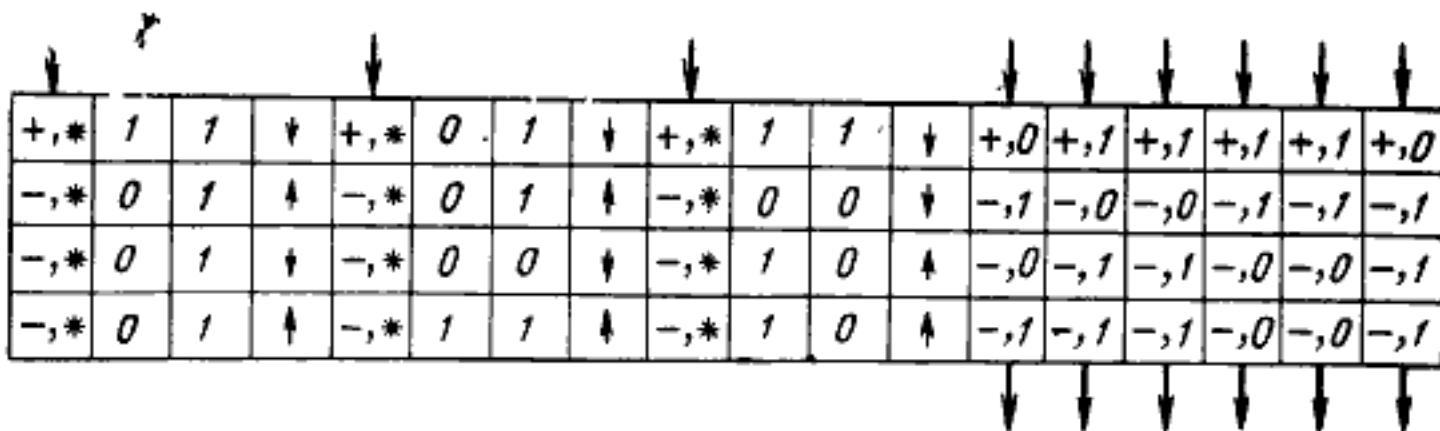


Рис. 4-23. Однородная среда из универсальных элементов.

ром  $(1,1,2,1,1,1,2,2)$ . Внешние входные и выходные каналы показаны на рис. 4-23. На выходе сети появляются наборы  $(a_1, a_2, a_3, a_4, a_5, a_6)$ , такие, что самое большое одна из пар  $(a_1, a_2)$ ,  $(a_3, a_4)$ ,  $(a_5, a_6)$  не имеет вид  $(0,0)$ . Образуем пару  $(a_1 + a_3 + a_5, a_2 + a_4 + a_6)$  и набор

$(a_1, a_2, a_3, a_4, a_5, a_6)$  будем декодировать символом  $z_i$ , кодом которого является пара  $(a_1 + a_3 + a_5, a_2 + a_4 + a_6)$ .

Рассмотрим работу сети при подаче входного набора  $(2,1,1,2,2,1,1,1,1)$ . Элементы с координатами  $i, j$ ,  $1 \leq i \leq 4$ ,  $13 \leq j \leq 18$ , реализующие функцию  $\lambda$ , работают точно так же, как и в сети на рис. 4-22. Элементы 1,5 и 1,9 перейдут в пассивные состояния  $(-, *)$ . Элемент 1,1 под действием входного символа 2 вырабатывает сигнал, который, распространяясь слева направо, проверяет, не записан ли в цепочке нулевой набор. Если бы был записан нулевой набор, то элемент 1,4 выдал бы сигналы, которые перевели бы все

элементы в первой строке в активные состояния. Посланный элементом 1,1 сигнал, отражаясь от элемента 1,4, проходит справа налево и вызывает передачу вниз набора  $(1,0,\downarrow)$ , который записывается в элементах 2,2; 2,3; 2,4, а именно эти элементы переходят в состояния  $(0,1)$ ,  $(1,0)$  и  $(\uparrow, \downarrow)$  соответственно. Элемент 1,1 передает вниз сигнал, приводящий в действие элемент 2,1. Далее происходит аналогичная проверка равенства нулю набора, записанного во вторых координатах состояний элементов 2,2 и 2,3. Элементы 2,2; 2,3 и 2,4 передают вниз набор  $(0,1,\downarrow)$  и возвращаются в исходные состояния. Наконец, вторые координаты состояний элементов 4,2 и 4,3 будут равны нулю, и элемент 4,4 выдаст сигналы, которые, распространяясь по четвертой строке, восстановят прежние состояния элементов 4,2 и 4,3 и переведут в активные состояния элементы 4,1; 4,5; 4,9; 4,13; ...; 4,18. В остальном работа сети на рис. 4-23 аналогична работе сети на рис. 4-22.

Нетрудно видеть, что можно построить элемент, работающий списанным образом, причем однородная двумерная сеть из таких элементов будет реализовать произвольный конечный автомат. Приведем один из вариантов такого универсального элемента. Элемент имеет четыре группы входных каналов —  $x_1, x_2, x_3, x_4$  и четыре группы выходных каналов —  $y_1, y_2, y_3, y_4$  (рис. 4-24). Поведение элемента описывается табл. 4-4, в которой при указании входных и выходных комбинаций опущены нулевые значения, а именно запись  $x_1=1$  следует читать как  $x_1=1, x_2=x_3=x_4=0$  и  $y_1=2$  — как  $y_1=2, y_2=y_3=y_4=0$ .

Нетрудно видеть, что некоторые состояния элементов избыточны. Проведя минимизацию, число состояний элемента можно довести до 9.

Интересно оценить сложность (число элементов) сети, реализующей заданный автомат. Пусть  $\mathcal{E}$  — элемент, такой, что однородные двумерные сети из этих элементов реализуют всевозможные конечные автоматы.

Через  $L_{\mathcal{E}}(W)$  обозначим наименьшую сложность сети из элементов  $\mathcal{E}$ , реализующей автомат  $W$ , а через  $L_{\mathcal{E}}(m, q, r)$  обозначим  $\max L_{\mathcal{E}}(W)$ , где максимум берется по всем автоматам  $W$  с входным алфавитом из  $m$  букв, выходным алфавитом из  $q$  букв и  $r$  состояниями.

*Теорема.* Существует элемент  $\mathcal{E}$ , такой, что

$$L_{\mathcal{E}}(m, q, r) \lesssim mr \log mr.$$

Доказательство по существу состоит в формализации изложенного выше метода.

Можно показать, что не существует элемента, который позволял бы получить существенно менее сложную реализацию. Очевидно, что если не учитывать число состояний элемента, то оценка сложности сети может быть сделана только с точностью до порядка, так как можно объединить в один  $n$  элементов и тем самым получить сеть, реализующую то же самое отображение и имеющую в  $n$  раз меньшую сложность.

Справедлива следующая теорема.

*Теорема.* Для любого элемента  $\mathcal{E}$

$$L_{\mathcal{E}}(m, q, r) \gtrsim mr \log qr$$

( $\varphi(x) > \psi(x)$ , если существует константа  $C$ , такая, что  $\psi(x) < C\varphi(x)$ ).

*Замечание.* Понятно, что можно построить различные универсальные элементы. Представляет интерес простейшая в некотором смысле реализация универсального элемента. Оказывается, что за счет усложнения рисунка межэлементных связей число состояний универсального элемента можно сократить до двух.

В заключение отметим следующие особенности предложенного метода реализации произвольных автоматов однородными двумерными сетями. Во-первых, сеть строится непосредственно по таблицам, задающим конечный автомат, при этом исключается этап структурного синтеза. Во-вторых, память элемента используется не только для настройки, но и в процессе работы сети.

Таблица 4-3

	$x_1 = 1$	$x_1 = 2$	$x_1 = 3$	$x_1 = 4$	$x_3 = 1$	$x_3 = 2$	$x_3 = 3$	$x_3 = 4$	$x_3 = 5$	$x_3 = 1$	$x_3 = 2$	$x_3 = 3$	$x_4 = 1$	$x_4 = 2$	$x_4 = 3$
(+, 0)	(-, 0)	(-, 0)	(-, 0)	(-, 0)	$y_1 = 0$										
(+, 1)	(-, 1)	(-, 1)	(-, 1)	(-, 1)	$y_1 = 0$										
(-, 0)	(-, 0)	(-, 0)	(-, 0)	(-, 0)	$y_1 = 1$										
(-, 1)	(-, 1)	(-, 1)	(-, 1)	(-, 1)	$y_1 = 1$										
(+, *)	(-, *)	(+, *)	(+, *)	(-, *)	$y_1 = 0$										
(-, *)	(-, *)	(+, *)	(+, *)	(+, *)	$y_1 = 1$										
(0)	(0, 1)	(0, 0)	(0, 0)	(0, 0)	$y_1 = 0$										

	$x_1 = 1$	$x_1 = 2$	$x_1 = 3$	$x_1 = 4$	$x_2 = 1$	$x_2 = 2$	$x_2 = 3$	$x_2 = 4$	$x_3 = 5$	$x_3 = 1$	$x_3 = 2$	$x_4 = 1$	$x_4 = 2$	$x_4 = 3$
--	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

(1)	(1, 1)	(1, 0)	(1)	(1)	(1)	(1)	(1)	(1)	(1, 1)	(1, 0)	(1)	(1)	(1)	(1)	
	$y_1 = 0$	$y_1 = 0$	$y_1 = 4$	$y_1 = 3$	$y_2 = 3$	$y_2 = 5$	$y_2 = 5$	$y_2 = 3$	$y_3 = 0$	$y_3 = 0$	$y_4 = 2$	$y_4 = 2$	$y_4 = 2$	$y_4 = 2$	
( $\uparrow$ )	( $\uparrow$ , $\downarrow$ )				( $\uparrow$ )				( $\uparrow$ , $\uparrow$ )	( $\uparrow$ )					
		$y_1 = 0$				$y_2 = 3$				$y_3 = 0$		$y_2 = 3$	$y_2 = 3$	$y_2 = 3$	$y_2 = 3$
( $\downarrow$ )	( $\downarrow$ , $\downarrow$ )				( $\downarrow$ )				( $\downarrow$ , $\uparrow$ )	( $\downarrow$ )					
		$y_1 = 0$				$y_2 = 3$				$y_3 = 0$		$y_2 = 3$	$y_2 = 3$	$y_2 = 3$	$y_2 = 3$
(0, 0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)
	$y_1 = 3$	$y_1 = 4$	$y_1 = 3$	$y_1 = 4$	$y_2 = 3$	$y_2 = 4$	$y_2 = 4$	$y_2 = 5$	$y_3 = 1$	$y_3 = 2$	$y_4 = 1$	$y_4 = 2$	$y_4 = 1$	$y_4 = 2$	$y_4 = 2$
(0, 1)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0, 1)	(0, 1)	(0, 1)	(0, 1)
	$y_1 = 4$	$y_1 = 3$	$y_1 = 2$	$y_1 = 2$	$y_2 = 3$	$y_2 = 2$	$y_2 = 2$	$y_2 = 5$	$y_3 = 5$	$y_3 = 2$	$y_4 = 2$	$y_4 = 2$	$y_4 = 1$	$y_4 = 2$	$y_4 = 2$

$x_1 = 1$	$x_1 = 2$	$x_1 = 3$	$x_1 = 4$	$x_2 = 1$	$x_2 = 2$	$x_2 = 3$	$x_2 = 4$	$x_3 = 5$	$x_3 = 1$	$x_3 = 2$	$x_4 = 1$	$x_4 = 2$	$x_4 = 3$
-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

(1, 1)

(1)

(1)

(1)

(1)

(1)

(1)

(1, 1)

(1, 1)

(1, 1)

(1, 1)

(1, 1)

(1, 1)

(1, 1)

(1, 1)

(1, 1)

(1, 1)

$y_1 = 4$

$y_1 = 3$

$y_2 = 5$

$y_2 = 5$

$y_2 = 5$

$y_2 = 5$

$y_3 = 2$

( $\uparrow$ ,  $\downarrow$ )

( $\uparrow$ ,  $\uparrow$ )

( $\uparrow$ ,  $\uparrow$ )

( $\downarrow$ ,  $\downarrow$ )

( $\downarrow$ ,  $\uparrow$ )

$y_1 = 4$

$y_1 = 3$

$y_2 = 2$

$y_1 = 4$

$y_1 = 3$

$y_2 = 2$

## ЛИТЕРАТУРА

1. Блох А. Ш., Ладес В. И. Синтез однотактных схем, поведение которых описывается линейными неравенствами. — «Известия АН БССР», сер. физико-технических наук, 1966, № 3.
2. Блох А. Ш., Ладес В. И. Метод синтеза принципиальных схем логических устройств. «Самонастраивающиеся системы. Распознавание образов. Конечные автоматы и релейные устройства», 1965, М., «Наука», 1967.
3. Варшавский В. И., Маражовский В. Б., Песчанский В. А. О задаче голосования в цепи автоматов. — «Известия АН СССР, Техническая кибернетика», 1968, № 4.
4. Варшавский В. И., Маражовский В. Б., Песчанский В. А. Некоторые варианты задачи о синхронизации цепи автоматов. — «Проблемы передачи информации», 1968, т. 4, вып. 3.
5. Варшавский В. И., Маражовский В. Б., Песчанский В. А. Реализация инвариантных функций линейными однородными схемами. — «Известия АН СССР. Техническая кибернетика», 1969, № 4.
6. Гершкович Ю. Б., Полтерович В. М. О бесповторных суперпозициях функций алгебры логики от двух переменных. — «Автоматика и телемеханика», 1967, № 5.
7. Гусев Л. А., Смирнова И. М. Языки, грамматика и абстрактные автоматные модели (обзор). — «Автоматика и телемеханика», 1968, № 4, № 5.
8. Евреинов Э. В., Косарев Ю. Г. Однородные универсальные вычислительные системы высокой производительности. Новосибирск, «Наука», 1966.
9. Игнатющенко В. В. Синтез переключательных функций в некоторых типах однородных структур — «Автоматика и телемеханика», 1968, № 5.
10. Кратко М. И. Регулярные и стабильные итеративные системы. — Сб. «Проблемы кибернетики», М., «Наука», 1967, вып. 19.
11. Кузнецов О. П. Вычисления в одномерных итеративных сетях. — «Конференция по теории автоматов и искусенному мышлению». Аннотации докладов, Ташкент, 27—31 мая 1968 г., ВЦ АН СССР, М., 1968.
12. Левенштейн В. И. Об одном методе решения задачи синхронизации цепи автоматов за минимальное время — «Проблемы передачи информации», 1965, т. 1, вып. 4.
13. Левенштейн В. И. О синхронизации двусторонних сетей автоматов. — «Проблемы передачи информации», 1968, т. 4, вып. 4.

14. Макаревский А. Я. Реализация отображений клеточными автоматами. — «Конференция по теории автоматов и искусственному мышлению». Ташкент, 27—31 мая 1968 г. Аннотации докладов. ВЦ АН СССР, М., 1968.
15. Маражовский В. Б., Песчанский В. А., Розенблум Л. Я. Логические возможности некоторых однородных схем. — Конференция по теории автоматов и искусственному мышлению. Ташкент 27—31 мая 1968 г. Аннотации докладов. ВЦ АН СССР, М., 1968.
16. Прангвили И. В., Абрамова Н. А., Бабичева Е. В., Игнатушенко В. В. Микроэлектроника и однородные структуры для построения логических и вычислительных устройств. М., «Наука», 1967.
17. Прангвили И. В. Итеративные и однородные планарные структуры и соответствующие им графы. — «Автоматика и телемеханика», 1968, № 5.
18. Amarel S., Cooke G., Winder R. O. Majority gate networks. IEEE Trans., EC-13, 1, 1965.
19. Akers S. B. Synthesis of combinational logic using three-input majority gates. Switching circuit and logical design. Oct. 7—12, 1962, AIEE publ. S-141, 1962.
20. Balzer R. An 8-state minimal time solution to the firing squad synchronization problem. Inf. and Control, vol. 10, I, 1967.
21. Caldwell S. H. Switching circuits and logical design. J. Wiley and Sons, New York, 1958. (С. Колдуэлл, Логический синтез релейных устройств. Изд-во иностр. лит., М., 1962.)
22. Canaday R. H. Two-dimensional iterative logic. Proc. Fall Joint Computer Conference, part I, 1965.
23. Dvorak A. Two rail cascade synthesis of Boolean functions. IEEE Trans., EC-17, 6, 1968.
24. Fisher P. C. Generation of primes by a one-dimensional real-time iterative array. J. Association for Computing Machinery, vol. 12, 3, 1965.
25. Goto E. A minimum time solution of the firing squad problem. Dittoed course notes for applied mathematics 298, Harvard University, May 1962.
26. Hennie F. S. Analysis of bilateral iterative networks. IRE Trans., CT-6, March 1959.
27. Hennie F. S. Iterative arrays of logical circuits. N. Y., Wiley, 1961.
28. Keister W., Ritchie A. E., Washburn S. H. The design of switching circuits. D. Van Nostrand Co., Princeton, N. J., 1951.
29. Kilmer W. L. Transient behaviour in iterative combinational switching networks. Switching Circuit Theory and Logical Design. Proc. of the Second Annual Symposium, Detroit, Mich., Oct. 17—20, 1961, Published by the AIEE, Sept. 1961.
30. Kilmer W. L. Iterative switching networks composed of combinational cells. IRE Trans., EC-11, 2, 1962.
31. Kilmer W. L. On dynamical switching in one-dimensional iterative logic networks. Inf. and Control, vol. 6, 4, 1963.
32. Ledley R., Wilson J. Integrated circuitry implication in logical design. Symp. Microelectronics and Large System, Washington, 1964. (Русск. перевод в сб. «Микроэлектроника и большие системы», М., «Мир», 1967.)

33. Lendaris G. G., Stanley G. L. On the structure-dependent properties of adaptive logic networks. G. M. Defense Research Labs., Santa Barbara, Calif., TR 63-219, July, 1963.
34. Levien R. E. The synthesis of cascade switching circuits. Switching Circuits Theory and Logical Design. Proc. of the 3-rd Annual Symp., 1962, AIEE publ. S141, 1962.
35. Levien R. E. Determining the best ordering of variables in cascade switching circuits. Switching Circuits Theory and Logical Design. Proc. of the 4-th Annual Symp., 1963, AIEE publ., S156, 1963.
36. Levy S. Y., Winder R. O., Mott T. H. A note on tributary switching networks. IEEE Trans., EC-13, 2, 1964.
37. Maitra K. K. Cascaded switching networks of two-input flexible cells. IRE Trans., EC-11, 2, 1962.
38. McCluskey E. J. Iterative combinational switching networks. General Design Considerations. IRE Trans., EC-7, 4, 1958.
39. McCluskey E. J. A comparison of sequential and iterative circuits. Communication and Electronics, 46, Jan. 1960.
40. McCluskey E. J. Reduction of feedback loops in sequential circuits and carry leads in iterative networks. «Switching Circuit Theory and Logical Design». Proc. of the 3<sup>rd</sup> Annual Symp., 1962, AIEE publ. S-141, 1962.
41. Minnic R. C. Cutpoint cellular logic. IEEE Trans., EC-15, 6, 1964.
42. Minnic R. C. Application of cellular logic to the design of monolithic digital systems. «Microelectronics and Large Systems». Spartan Books, Washington, 1965. (Русск. перевод в сб. «Микроэлектроника и большие системы», М., «Мир», 1967.)
43. Minnic R. C. Cobweb cellular arrays. Proc. 1965 Fall Joint Computer Conf., part II. (Русск. перевод в сб. «Микроэлектроника и большие системы», «Мир», М., 1967.)
44. Minnic R. C. A survey of microcellular research. J. of ACM, vol. 14, 2, 1967.
45. Moore E. F. The firing squad synchronization problem. «Sequential machines» reading. Mass-Palo Alto-London, Addison-Wesley Publ. Co., Inc., 1964 (Мур Э. Задача о синхронизации цепи стрелков. «Кибернетический сборник», вып. 1, «Мир», 1965).
46. Okada S., Liss D., Rajappan K. P., Young K. P. Boolean two-terminal analysis and synthesis. AIEE publ. S-134, 1961.
47. Plisch D. C., Scidmore A. K. The number of equivalence classes of functions realizable by tributary networks. IEEE Trans., EC-15, 2, 1966.
48. Short R. A. Two-rail cellular cascades. AFIPS Conference Proc., vol. 27, part I, Washington, Spartan Books, 1965. (Русск. перевод в сб. «Микроэлектроника и большие системы», М., «Мир», 1967.)
49. Sklansky J. General synthesis of tributary switching networks. IEEE Trans., EC-12, 5, 1963.
50. Sklansky J., Korenjak A. J., Stone H. S. Canonical tributary networks. IEEE Trans., EC-14, 6, 1965.
51. Stone H. S. On the number of equivalence classes of functions realizable by cellular cascades. Proc. Nat. Symp. Impact Batch Fabric. Future Computers, Los Angeles, Calif., 1965.
52. Stone H. S. Canonical form synthesis of cellular cascades. IEEE Trans., EC-14, 6, 1965.

53. Stone H. S., Korenjak A. J. Canonical form and synthesis of cellular cascades. IEEE Trans., EC-14, 6, 1965.
54. Unger S. H. A computer oriented toward spatial problems. Proc. IRE, 46, 10, 1958.
55. Unger S. H. Pattern recognition using two-dimensional, bilateral, iterative, combinational switching circuits. Proc. Polytechnic Inst. of Brooklyn Symp. on Mathematical Theory of Automata. New York, Polytechnik Press Brooklyn, 1963.
56. Varshavsky V. I. Collective behaviour and control problems. Machine Intelligence III, Ed. D. Michie, Edinburgh University Press., 1968.
57. Varshavsky V. I. The organization of interaction in collectives of automata. Machine Intelligence IV, Ed. B. Meltzer, D. Michie, Edinburgh University Press, 1969.
58. Waite W. M. The production of completion signal by asynchronous iterative networks. IEEE Trans., EC-13, 2, 1964.
59. Waksman A. An optimum solution to the firing squad synchronization problem. Inf. and Control, vol. 9, 1, 1966
60. Yang C. C., Yan S. S. A cutpoint cellular associative memory. IEEE Trans., EC-15, 4, 1966.
61. Yoeli M. A group-theoretical approach to the two-rail cascades. IEEE Trans., EC-14, 6, 1965.
62. Yoeli M. Generalized Maitra cascades. IEEE Trans., EC-14, 6, 1965.
63. Yoeli M. Ternary cellular cascades. IEEE Trans., C-17, 1, 1968.

## ОГЛАВЛЕНИЕ

Предисловие . . . . .	3
Введение . . . . .	4
<b>Глава первая. Односторонние одномерные сети и системы из комбинационных элементов . . . . .</b>	<b>13</b>
1-1. Одноканальные сети и системы из ненастраиваемых элементов . . . . .	15
1-2. Одноканальные бесповторные сети из настраиваемых элементов . . . . .	19
1-3. Одноканальные повторные сети из настраиваемых элементов . . . . .	25
1-4. Двухканальные сети . . . . .	29
1-5. Трехканальные сети . . . . .	34
1-6. Об эквивалентности произвольных односторонних сетей и сетей из элементов с одним внешним входом . . . . .	35
1-7. Системы из комбинационных элементов . . . . .	38
<b>Глава вторая. Одномерные односторонние сети и системы из элементов с памятью . . . . .</b>	<b>50</b>
2-1. Стабильные сети и системы . . . . .	50
2-2. Нестабильные сети и системы . . . . .	65
2-3. Реализация булевых функций одномерными сетями из элементов с памятью . . . . .	70
<b>Глава третья. Двусторонние одномерные сети и системы . . . . .</b>	<b>73</b>
3-1. Двусторонние одномерные системы из комбинационных элементов . . . . .	73
3-2. Двусторонние сети и системы из элементов с памятью . . . . .	87
3-3. Двусторонние системы из автоматов Мура . . . . .	92
3-4. О возможности моделирования поведения двусторонних одномерных сетей односторонними кольцами . . . . .	112
<b>Глава четвертая. Многомерные сети и системы . . . . .</b>	<b>114</b>
4-1. Многомерные системы . . . . .	114
4-2. Однородные решетки из трехходовых элементов . . . . .	121
4-3. Однородные решетки из настраиваемых элементов, реализующих некоторые функции двух переменных . . . . .	129
4-4. Решетки типа «паутинка» . . . . .	133
4-5. Решетки типа «алмаз» . . . . .	135
4-6. Реализация автоматов в однородных сетях . . . . .	139
<b>Литература . . . . .</b>	<b>147</b>
	151

*Виктор Ильич Варшавский  
Вячеслав Борисович Мараховский  
Валерий Анатольевич Песчанский  
Леонид Яковлевич Розенблум*

**Однородные структуры (Анализ. Синтез.  
Поведение)**

Редактор *Н. И. Глазунов*

Редактор издательства *Т. В. Жукова*

Обложка художника *А. М. Кувшинникова*

Технический редактор *О. Д. Кузнецова*

Корректор *Г. Г. Желтова*

---

Сдано в набор 21/XII 1971 г.

Подписано к печати 30/XI 1972 г.

Т-19639 Формат 84×108<sup>1</sup>/<sub>32</sub> Бумага типографская № 2

Усл. печ. л. 7,98

Уч.-изд. л. 7,85

Тираж 8 000 экз.

Зак. 1004

Цена 40 коп.

---

Издательство „Энергия“. Москва, М-114,  
Шлюзовая наб., 10.

Московская типография № 10 Главполиграфпрома  
Комитета по печати при Совете Министров СССР.  
Шлюзовая наб., 10.