

КОММУТИРУЮЩИЕ СЕТИ ОДНОРОДНЫХ СТРУКТУР

И.П. Егоров, Т.М. Парамонова

(Москва)

Однородная перестраиваемая структура включает в себя две сети: логическую и коммутирующую. Коммутирующая сеть представляет собой коммутатор с конечной памятью. Каждому из состояний логической сети поставлено в соответствие возбуждение определенных ключей коммутирующей сети, связывающих функциональные элементы логической сети в определенную схему. Таким образом с помощью управляющего коммутирования в однородной структуре моделируются различные устройства. Настройка структуры на выполнение конкретной задачи осуществляется при поступлении в элементы памяти коммутирующей сети управляющей информации, содержащейся в программном устройстве. Для уменьшения ошибок, возникающих вследствие самопроизвольного включения и отключения элементов памяти, перепись программы производится непрерывно цикл за циклом. Значения переменных при этом должны меняться не чаще одного раза за цикл переписи.

Исполнение однородной структуры в интегральном виде существенно ограничивает число входных каналов (внешних выводов), по которым к элементам памяти сети коммутации поступают сигналы настройки из программного устройства. Необходимо также, чтобы коммутирующая сеть была однородной, т.е. состояла из одинаковых участков, повторением которых воспроизводится вся сеть.

Эти, во многом противоречивые, требования создают трудности при построении однородных коммутирующих сетей большой емкости, управляемых с помощью ограниченного числа внешних выводов. Так, например, становится невозможным использование адресного управления, когда управляющий код, состоящий из кода адреса и кода настройки, поступает в коммутирующую сеть и при совпадении адресов код настройки запоминается соответствующими элементами памяти. Способ неприемлем, поскольку адреса (дешифраторы) в этом случае различны, и, следовательно, сеть коммутации неоднородна.

В настоящей статье описываются некоторые коммутирующие сети, удовлетворяющие отмеченным выше требованиям интегральной технологии. Для каждого рассматриваемого способа управления приводятся зависимости:

$$A_1 = f(N_1), \quad T_1 = \varphi(N_1, n_1),$$

где A_1 — число управляющих входов (внешних выводов) сети коммутации;

T_1 — время настройки;

N_1 — число элементов памяти сети;

$n_1 \in N_1$ — число тех элементов памяти, которым соответствуют коммутаторы сети, подлежащие возбуждению при выполнении данной задачи.

Способы управления элементами памяти сети коммутации делятся на три группы.

1. При параллельном управлении все элементы настраиваются одновременно; этот способ здесь не рассматривается, поскольку управление памятью осуществляется с помощью большого числа внешних выводов ($A_1 = N_1$).

2. Последовательное управление допускает выборку одного элемента памяти сети за один такт настройки структуры. При этом возможны два случая: либо все элементы памяти управляются в такой последовательности, что настройка i -го элемента возможна только после настройки $i-1$ элемента, либо допускается выборка элементов памяти в произвольном порядке. В первом случае время настройки T_1 не зависит от n_1 ($T_1 = N_1$ тактам), во втором случае $T_1 = n_1$.

3. Параллельно-последовательное управление есть сочетание двух отмеченных способов. При этом, если p_1 — число элементов, настраиваемых параллельно (одновременно), то

$$T_1 = \frac{N_1}{P_1}$$

или

$$T_1 = \frac{n_1}{P_1}$$

Координатный метод управления

В общем случае координатное управление (рис. 1а) основано на принципе временного совпадения сигналов, которые поступают из программного устройства на координатные шины сети, и комбинации которых представляют собой различные неповторяющиеся сочетания. При координатном управлении

$$N_1 = [C_{m+1}^q - (C_m^q + C_1^q)] k \cdot S,$$

где q — число входов схемы совпадения "И". Выход схемы "И" управляет i -ым элементом памяти ($i = 1, 2, \dots, N$)

$k \cdot S$ — число участков сети;

C_{m+1}^q — число всевозможных сочетаний сигналов из $m+1$ по q , поступающих на координатные шины одного участка сети;

$C_m^q + C_1^q$ — число комбинаций сигналов, сочетания которых являются общими для всех участков одного столбца (строки).

При координатном управлении $A_1 = m \cdot k + 1 \cdot S + 1$ — один дополнительный вход служит для перевода всех элементов памяти в исходное состояние $T_1 = n_1$.

Если $m = 1 = 1$, то $N_1 = S \cdot k$, $A_1 = k + S + 1$. В этом случае, с целью сокращения T_1 , управлять элементами памяти следует параллельно-последовательным способом, как показано в работе [1]; за один цикл производится перепись содержимого программного устройства в элементы памяти сети последовательно строка за строкой, содержимое же строки переписывается одновременно (параллельно), таким образом $T_1 = k$, где k — число строк.

Двухкоординатный метод управления достаточно прост и коммутирующая сеть весьма однородна. Недостатком метода является большое число координатных шин (внешних выводов), необходимых для управления коммутирующей сетью большой емкости.

Число A_1 при двухкоординатном управлении можно сократить до $A_1 = 1$, если при допущении некоторой неоднородности (такое допущение вполне правомерно, ибо краевые области конечной сети не отвечают принципу однородности) управлять координ-

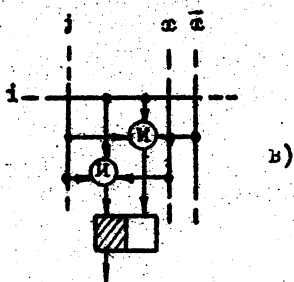
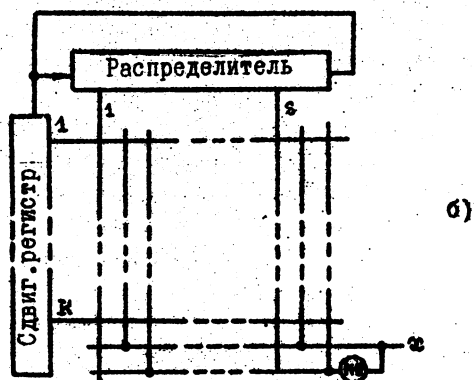
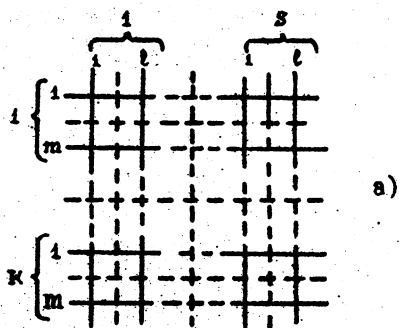


Рис. I.

натными шинами, как показано на рис. 1б. В момент, когда распределитель находится в позиции j ($j = 1, 2, \dots, S$), а регистр — в позиции i ($i = 1, 2, \dots, k$), значение сигнала настройки x фиксируется элементом памяти с координатами i, j (рис. 1в). При этом $T_1 = S \cdot k = N_1$.

Управление с помощью распределителя

Пусть коммутирующая сеть включает в себя циклический распределитель на N_1 позиций (рис. 2а). Каждой позиции распределителя соответствует свой элемент памяти. Значение управляющего сигнала x в момент, когда распределитель находится на i -ой позиции, определяет состояние i -ого элемента памяти сети. Таким образом, $A_1 = I$, $T_1 = N_1$.

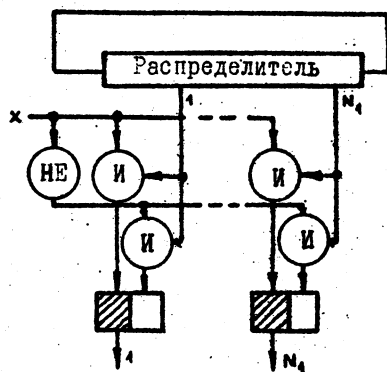
Основным достоинством схемы коммутации является минимальное число внешних выводов управляющих сетей любой емкости. Вместе с тем, методу присущи такие недостатки, как большое время настройки, относительная сложность схемы, низкая надежность (при любом отказе типа обрыва или короткого замыкания в цепи распределителя нарушается работа всей схемы).

С тем, чтобы ускорить настройку структуры, достаточно перейти к последовательно-параллельному управлению, т.е. составить сеть из p_1 параллельно работающих распределителей; при этом — $T_1 = \frac{N_1}{p_1}$, $A_1 = p_1$.

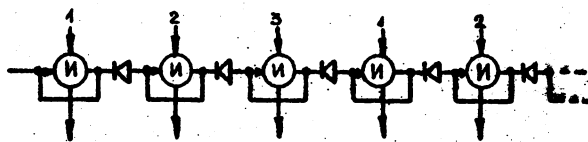
Для того, чтобы упростить конструкцию распределителя, а следовательно, и всей сети коммутации, следует дополнительно ввести три входа $1, 2, 3$ и построить распределитель, как показано на рис. 2б. Представленная цепочка выполняет функции распределителя; при подаче на вход цепочки короткого запускающего импульса, последний, синхронно с тактовыми сигналами $1, 2, 3$ (рис. 2в), перемещается по цепочке, имитируя работу распределителя.

Наконец, с тем, чтобы иметь возможность производить настройку структуры при неисправностях в сети коммутации типа обрыва, сеть строится следующим образом.

Пусть, например, плоским антисимметрическим графом на рис. 3а изображена некоторая однородная структура. Если каждой дуге графа поставить в соответствие элемент, приведенный на рис. 3б, то получим коммутирующую сеть этой структуры (участок сети показан на рис. 3в). Цифра, которой помечена ду-



а)

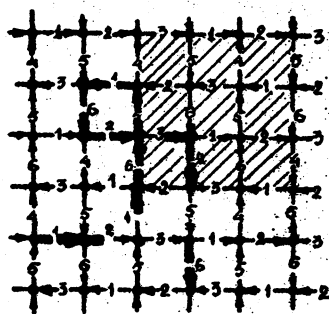


б)

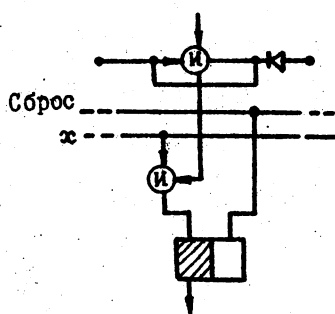


в)

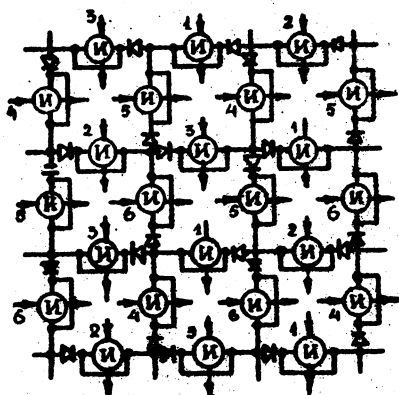
Рис. 2.



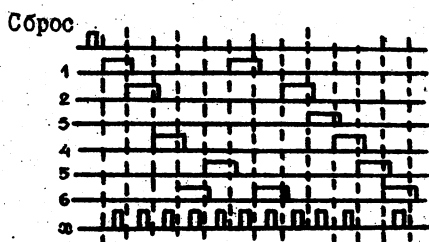
а)



б)



в)



г)

Рис. 3.

га графа, показывает, что схема совпадения элемента, изображаемого дугой, управляется настроечным входом, обозначенным этой цифрой. В зависимости от последовательности сигналов настройки, поступающих на входы $1, 2, \dots, 6$ (рис. 3г), короткий запускаящий импульс, поданный на один любой краевой вход схемы коммутации, начинает "перемещаться" от элемента к элементу, "воспроизводя" подграф определенной конфигурации на графе сети (рис. 3а, например).

Значение управляющего сигнала на входе x сети коммутации в момент, когда "возбуждена" дуга $i \in N$, определит состояние i -ого элемента памяти. Аналогично строятся коммутирующие сети однородных структур, изображаемых любыми графами.

Рассмотренный принцип построения коммутирующих сетей, очевидно, дает возможность производить настройку при обрывах, а также позволяет существенно сократить время настройки, поскольку выборку элементов можно производить в соответствии со схемой реализуемого устройства. Действительно, задача построения в однородной структуре некоторого устройства сводится к построению на графе структуры G подграфа G' , соответствующего этому устройству. Если G' связный и имеет четные степени вершин, то он обладает эйлеровой линией, т.е. линией, проходящей через каждую дугу графа G' в точности по одному разу; тогда, очевидно, для настройки структуры следует "воспроизвести" граф G' , для чего понадобится $T_1 = n_1$ тактов, где n_1 — число дуг G' . Если G' не связный граф, или связный, но не эйлеров, то в нем всегда можно выделить семейство из R подграфов, каждый из которых есть эйлеров граф; тогда для реализации G' потребуются $T_1 > n_1$, поскольку при переходе от одного подграфа из R к другому некоторое число шагов затрачивается "вхолостую", т.к. на этих шагах не производится настройка элементов памяти (на управляющий вход x не подается сигнал настройки).

Рассмотренные способы управления элементами памяти называем непосредственными. С целью повышения эффективности использования ограниченного числа внешних выводов следует перейти к косвенному управлению, при котором настройка каждого элемента памяти осуществляется в два этапа: сначала выбирается участок сети, содержащий нужный элемент памяти, затем производится настройка этого элемента. Любой из описанных выше способов можно использовать на первом этапе настройки; выход L 1-ой схемы совпадения будет сигналом выборки 1-го участка сети коммутации.

Таким образом, при косвенном управлении N_1 - число участков сети коммутации. Если N_2 - число элементов памяти одного участка, то общий объем памяти сети $N = N_1 \cdot N_2$. Если A_2 - число управляющих входов (общих для всей сети), дополнительно вводимых для выборки элементов памяти внутри участков, то общее число управляющих вводов (внешних выводов) при косвенном управлении $A = A_1 + A_2$. Если T_2 - время настройки элементов памяти одного участка, то $T = N_1 \cdot T_2$ - есть время настройки структуры.

Схема участка сети в общем случае представлена на рис. 4а. Очевидно:

$$N_2 = C_{A_2}^0 + C_{A_2}^1 + C_{A_2}^2 + \dots + C_{A_2}^{A_2} = 2^{A_2};$$

$$T_2 = n_2.$$

В частном случае участок сети (рис. 4б) содержит N_2 элементов памяти, каждый из которых управляется собственной схемой совпадения. Схемы "И" имеют $q+1$ входов (на дополнительный $(q+1)$ -ый вход поступает сигнал L выборки участка). В этом случае:

$$N_2 = C_{A_2}^q; \quad T_2 = n_2.$$

В другом частном случае (рис. 4в) число элементов памяти участка составляет

$$N_2 = C_{A_2}^2 - (C_K^2 + C_B^2), \quad \text{где } A_2 = k + s;$$

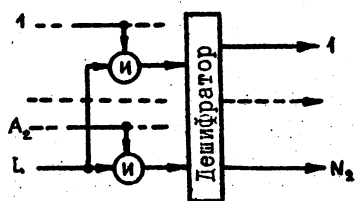
$$T_2 = \frac{n_2}{k^2}.$$

Наконец, можно воспользоваться двумя простыми приемами для увеличения объема памяти сети коммутации. Так, для всех рассмотренных случаев непосредственного и косвенного управления объем памяти можно увеличить в два раза (введя дополнительно одну линию управления M), если сеть строить как показано на рис. 4г для косвенного управления. Можно также, введя дополнительно C входов, увеличить объем памяти в C раз, если построить сеть как показано на рис. 4д для косвенного управления.

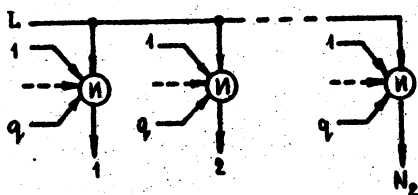
В общем случае, для рассмотренных способов управления элементами памяти коммутирующих сетей

$$A = A_1 + A_2 + C + 1; \quad N = 2 \cdot C \cdot N_1 \cdot N_2.$$

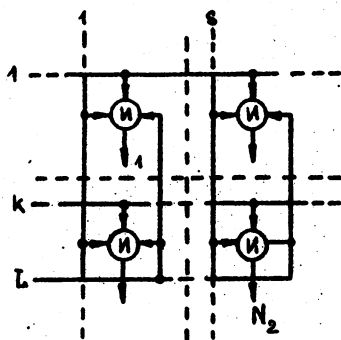
Описанные способы управления и приведенные соотношения позволяют выбрать приемлемую схему сети коммутации для задан-



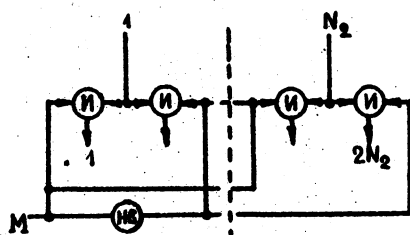
а)



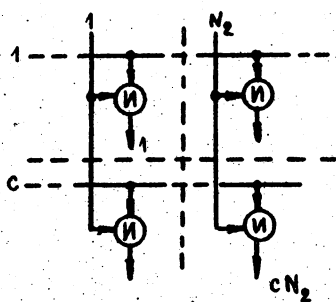
б)



в)



г)



д)

Рис. 4.

ных А и Н. В дальнейшем предполагается для различных способов управления выработать комплексные оценки, учитывающие также сложность и надежность коммутирующих сетей.

Л и т е р а т у р а

1. И.П. Егоров, И.В. Прангивили, М.А. Ускач. Однородная микроэлектронная структура и реализация в ней логических функций. — Автоматика и телемеханика, 1967, № 7.