

УНИВЕРСАЛЬНЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ С ЧАСТИЧНО
ПЕРЕМЕННОЙ СТРУКТУРОЙ

Э.В. Евреинов

Одним из перспективных направлений в создании универсальных вычислительных систем является разработка систем с переменной структурой [1]. С помощью таких систем можно будет программным способом изменять

- а) число элементарных машин;
- б) систему коммутаций между элементарными машинами;
- в) набор операций, разрядность слов, объем памяти в элементарной машине.

В данной работе рассматриваются некоторые теоретические основы построения вычислительных систем высокой производительности с частично переменной структурой.

§ 1. Основные определения

Пусть дан конечный автомат $\mathcal{K}$ с входными каналами $x_{i\ell}$ и выходными каналами $z_{i\ell}$ ($i=0,1,\dots,n$, $\ell=1,2,3$) (Рис. 1). Пусть каждый из каналов может пребывать в u попарно различных состояниях. Поставим в соответствие каждому такому состоянию символ из некоторого конечного множества символов A (число символов равно u). Множество A будем называть алфавитом, а отдельные сим-

воли этого множества - буквами. Словом в алфавите A будем называть всякую упорядоченную последовательность букв из множества A . Будем предполагать, что в каждый дискретный момент времени $t = 1, 2, \dots$, в каждый входной канал поступает и из каждого выходного канала выдается по одной букве. Будем обозначать через $x_{i\ell}(t)$ и $z_{i\ell}(t)$ буквы, проходящие в момент t по каналам $x_{i\ell}$, $z_{i\ell}$, соответственно.

Назовем алфавитом состояний автомата $\mathcal{K}$ алфавит $B = \{H_k C_{ij}^{(e)}\}$, где

$$k=1,2; \ell=1,2,3; i,j=0,1,\dots,n; H_k \in \{0,1\}; C_{ij}^{(e)} \in \{0,1\}.$$

Пусть даны алфавиты $A_1 = \{a_1, a_2, \dots, a_m\}$; $A_2 = \{p\}$; $A_3 = \{\alpha_1, \alpha_2, S_{ij}^{(e)}\}$, составленные из символов, поступающих во входные каналы x_{i1}, x_{i2}, x_{i3} , соответственно.

Назовем коммутационно-настроенным конечным автоматом $\mathcal{K}$, заданный системой канонических уравнений:

$$\left. \begin{aligned} z_{i\ell}(t) &= x_{i\ell}(t) \& C_{ij}^{(e)}(t) \& H_2(t), \quad i,j=0,1,\dots,n; \ell=1,2; \\ z_{i3}(t) &= x_{i3}(t) \& C_{ij}^{(3)}(t) \& H_2(t) \text{ для } x_{i3} \in \{\alpha_2, S_{ij}^{(e)}\}; \\ &\quad e=1,2,3 \\ z_{i3}(t) &= x_{i3}(t) \& C_{ij}^{(1)}(t) \& H_1(t) \text{ для } x_{i3} \in \{\alpha_1\}; \\ H_1(t+1) &= H_1(t) \vee (x_{i3}(t) = \alpha_1), \quad \text{где } H_1(t), H_2(t) \in \{0,1\}; \\ H_2(t+1) &= H_2(t) \vee (x_{i3}(t) = \alpha_2); \\ C_{ij}^{(e)}(t+1) &= x_{i3}(t) \& H_1(t) \quad \text{для } x_{i3} \in \{S_{ij}^{(e)}\}, \\ &\quad S_{ij}^{(e)} \in \{0,1\}, \\ &\quad i,j=0,1,\dots,n; \ell=1,2,3. \end{aligned} \right\} (I)$$

Назовем внешними полюсами автомата концы каналов, не примыкающие к автомату, внутренними полюсами - концы каналов, примыкающие к автомату. Будем различать соответственно входные и выходные полюсы.

Под соединением будем понимать отождествление внутренних полюсов входных каналов с внутренними полюсами выходных каналов. Введем матрицу соединений $[C_{ij}^{(e)}]$. В матрице строки соответствуют внутренним входным полюсам, а столбцы выходным. В случае отождествления $x_{i\ell}$ внутреннего входного полюса с $z_{j\ell}$ внутренним выходным полюсом $C_{ij}^{(e)} = 1$, а при отсутствии отождествления полюсов $C_{ij}^{(e)} = 0$.

Пусть дан программный автомат В.М. Глушкова [2] с универсальным набором операций (ввода и вывода, пересылки, переадресации на ± 1 и условного перехода по точному совпадению слов). Введем дополнительно входные и выходные каналы $y_{0\ell}$ и $W_{0\ell}$ ($\ell = 1, 2, 3$).

Будем полагать, что при выполнении операции условного перехода в выходной канал W_{02} выдается буква $p \in \{0, 1\}$ (при совпадении $p = 1$, при несовпадении $p = 0$). Введем также дополнительный набор операций $\mathcal{L}$. Пусть $v_{e_0+k}, v_{e_0+2k}, \dots, v_{e_0+mk}$ - номера слов, составленных из букв алфавита A_{i1} .

Назовем операцией передачи операцию π_1 , которая сводится к последовательной передаче слов с номерами $v_{e_0+k \cdot i}$ из памяти автомата в выходной канал W_{01} . $\pi_1: [v_{e_0+ki}] \Rightarrow W_{01}$, где $i = 1, 2, \dots, m; k, e_0$ - константы.

Назовем операцией приема операцию π_2 , которая сводится к последовательной передаче слов с номерами v_{e_0+ki} в память автомата по входному каналу y_{01} .

$$\pi_2: y_{01} \Rightarrow [v_{e_0+ki}].$$

Назовем обобщенной операцией условного перехода операцию $\mathcal{U}$, при которой совершается переход к очередной команде в случае выполнения условия $P_k(m)$ и к команде с номером v_e - в противном случае.

Это условие определяется по формуле:

$$\left. \begin{aligned} P_k(m) &= \text{sign} \left(\sum_{i=1}^m y_{02}(t) \cdot i - k \right) \\ P_k(m) &= \begin{cases} 1 & \text{при } \left(\sum_{i=1}^m y_{02}(t) \cdot i - k \right) \geq 0 \\ 0 & \text{при } \left(\sum_{i=1}^m y_{02}(t) \cdot i - k \right) < 0 \end{cases} \end{aligned} \right\} (2)$$

где k - устанавливаемая константа, $y_{02}(t) \in A_{i2} = \{p\}$,

$$p \in \{0, 1\}$$

$$y: \Rightarrow v_\ell, \text{ если } P_K(m)=0 \quad \ell \in \{1, 2, \dots\} \\ \Rightarrow \ell+1, \text{ если } P_K(m)=1$$

Назовем операцию настройки H операцией, при которой из памяти автомата (настроечной информации) в выходной канал W_{03} передается последовательность слов либо вида $\alpha_1 \Rightarrow W_{03}; \{S_{ij}^{(e)}\} \Rightarrow W_{03}; \alpha_2 \Rightarrow W_{03} \quad i, j=0, 1, \dots, n, \ell=1, 2, 3$, либо вида $\{S_{ij}^{(e)}\} \Rightarrow W_{03}; \alpha_2 \Rightarrow W_{03}$.

Для общности будем считать возможным прием в память автомата настроечной информации, поступающей во входной канал y_{03} .

Универсальный программный автомат с конечным объемом памяти, с дополнительными входными y_{0e} и выходными w_{0e} каналами, дополнительным набором операций $\mathcal{L} = \{\pi_1, \pi_2, y, H\}$ будем называть функциональным автоматом Φ (рис. 2).

Используя общий способ композиции автоматов [2], отождествим полюса x_{0e}, z_{0e} коммутационно-настроечного автомата K с полюсами w_{0e}, y_{0e} функционального автомата Φ , соответственно.

Схему из автоматов Φ и K , полученную в результате такого отождествления, назовем элементарной машиной (ЭМ) (рис. 3).

В элементарной машине полюса x_{ie} , где $i=1, 2, \dots, n$ будем называть входными, а полюса z_{ie} выходными.

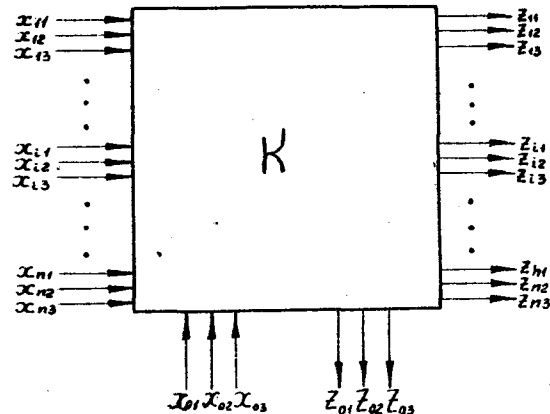


Рис. 1.

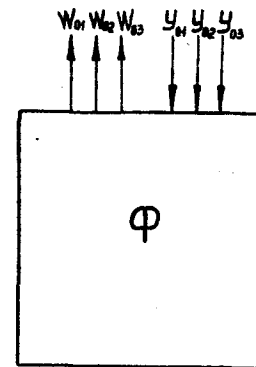


Рис. 2

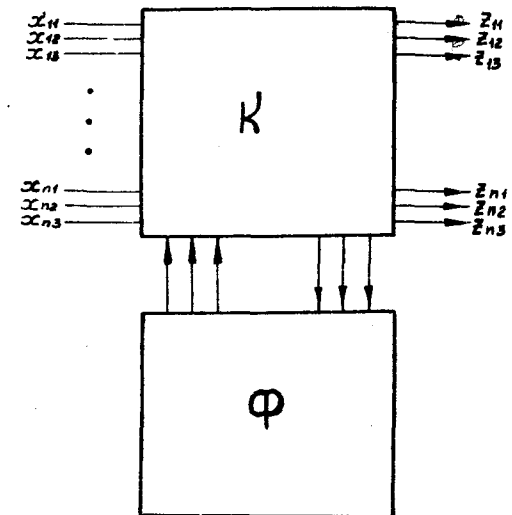


Рис. 3

Назовем сетку двумерной, если она образована пересечением двух семейств параллельных прямых:

$$v_1 = v_{10} + i h \quad (i=0, \pm 1, \pm 2, \dots),$$

$$v_2 = v_{20} + j h \quad (j=0, \pm 1, \pm 2, \dots).$$

Точки пересечения этих прямых назовем узлами.

Два узла будем называть соседними, если они удалены друг от друга в направлении оси v_1 или v_2 на расстояние шага сетки. Пусть по аналогии с двумерной сеткой дана S' -мерная сетка $(v_1, v_2, \dots, v_s)$.

Тогда универсальной вычислительной системой с частично переменной структурой будем называть S' -мерную сетку, у которой

- 1) узлами являются элементарные машины;
- 2) входные полюса x_{ie} каждой ЭМ отождествлены соответственно с выходными полюсами z_{ie} ($i=1, 2, \dots, n=2S'$) соседних ЭМ;

3) выходные полюса x_{ie} каждой ЭМ отождествлены соответственно с входными полюсами x_{ie} ($i=1,2,\dots,n=2S$) соседних ЭМ;

4) выделено подмножество входных и выходных полюсов ЭМ, которые называются входными и выходными полюсами системы.

§ 2. Основные свойства универсальных вычислительных систем с частично переменной структурой

1. Универсальность вычислительной системы.

ТЕОРЕМА I. В S -мерной сетке из элементарных машин может быть реализован всякий конечный автомат.

Доказательству теоремы I предположим следующие леммы:

ЛЕММА I. Матрица соединений $[C_{ij}^{(e)}]$ $n+1$ -го порядка автомата $\mathcal{K}$ эквивалентна (тождественна) матрице смежности $[a_{ij}^e]$ графа $G=(Y,V)$ с $n+1$ -ой вершиной.

Поставим в соответствие каждой внешней паре полюсов x_{ie} , z_{ie} вершину графа $y_i \in Y$, где $i=0,1,\dots,n$; тогда элемент матрицы смежности a_{ij}^e будет эквивалентен (равен) элементу $C_{ij}^{(e)}$ матрицы соединений. Действительно, если вершина y_i соединена с вершиной y_j дугой $v_e \in V$, идущей от y_i в y_j , то $a_{ij}^e = C_{ij}^{(e)} = 1$, что соответствует отождествлению полюса x_{ie} с полюсом z_{ie} ; в противном случае $a_{ij}^e = C_{ij}^{(e)} = 0$. Если вершина y_i соединена с вершиной y_j дугой, идущей от y_j в y_i , то $a_{ji}^e = C_{ji}^{(e)} = 1$, в противном случае $a_{ji}^e = C_{ji}^{(e)} = 0$. Следовательно, матрицы смежности и соединений эквивалентны.

СЛЕДСТВИЕ I. Матрица соединений $[C_{ij}^{(e)}]$ $n+1$ -го порядка автомата $\mathcal{K}$ отображает граф $G=(Y,V)$ с $n+1$ -ой вершиной.

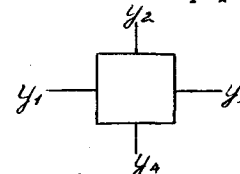
СЛЕДСТВИЕ 2. В автомате $\mathcal{K}$ может быть отображен всякий конечный граф $G=(Y,V)$ с $n+1$ -ой вершиной.

ЛЕММА 2. Всякий граф может быть отображен на S -мерной сетке с помощью всевозможных графов с $2S$ вершинами ($S \geq 2$).

Для доказательства леммы достаточно показать, что любой граф может быть изображен на двумерной сетке с помощью графов с 4-мя вершинами.

Пусть дан граф $G=(X,V)$. Поставим в соответствие вершинам графа $x_1, \dots, x_n$ строки сетки, а дугам $v_1, \dots, v_m$ столбцы сетки. Требуется соединить дугой вершину x_i с вершиной x_j . Здесь возможны три случая: $i > j$, $i = j$, $i < j$.

Пусть имеются всевозможные графы с четырьмя вершинами вида:



Поставим на пересечении i -ой строки, соответствующей x_i вершине, и столбца, соответствующего дуге (x_i, x_j) , граф, у которого вершины y_1, y_3 соединены дугами (y_1, y_3) и (y_3, y_1) ; а в случае $i > j$ вершины y_2 и y_4 соединены дугой (y_2, y_4) . На пересечении j -ой строки и столбца для дуги (x_i, x_j) построим граф, у которого вершины y_1 и y_3 соединены дугами (y_1, y_3) и (y_3, y_1) , а вершины y_2 и y_4 дугой (y_2, y_4) . На пересечении столбца, соответствующего дуге (x_i, x_j) и каждой из остальных строк, построим граф, вершины y_1 и y_3 которого соединены дугами (y_1, y_3) и (y_3, y_1) , а вершины y_2 и y_4 дугой (y_2, y_4) .

Последовательность дуг $(y_1, y_4)_i, (y_2, y_4)_{i-1}, \dots, (y_2, y_1)_j$ образует путь, который является фактически дугой (x_i, x_j) . Из этого построения видно, что только x_i и x_j вершины соединены данной дугой (x_i, x_j) . Аналогично могут быть проведены дуги для случаев $i = j$, $i < j$. В силу произвольности взятых вершин x_i, x_j аналогично могут быть проведены все остальные дуги графа. Следовательно, любой граф может быть изображен как в двумерной, так и в S -мерной сетках с помощью всевозможных графов с $2S$ вершинами.

СЛЕДСТВИЕ I. В S -мерной сетке из элементарных машин может быть отображен любой граф.

Перейдем теперь к доказательству теоремы I. Покажем, что элементарная машина образует полную систему элементарных автоматов и соединительных элементов [3]. Полнота системы из элементарных автоматов определяется наличием в элементарной машине памяти и реализацией универсального набора операций. Полнота системы соединительных элементов вытекает из лемм I и 2.

Тогда из справедливости теоремы 6 (см. [3]) следует доказательство теоремы I.

СЛЕДСТВИЕ 2. В УВС может быть реализована система Унгера [4], система Холланда [5], система Ли [6], система СОЛОМОН [7].

ТЕОРЕМА 2. Универсальная вычислительная система с частично переменной структурой является универсальной в смысле В.М. Глушкова, если в набор операций элементарной машины системы, кроме операций ввода и вывода, операции пересылки, переадресации на $\pm I$ условного перехода по точному совпадению слов, входят также операции настройки, приема и передачи информации.

Доказательство основывается на том факте, что каждая элементарная машина обладает универсальным набором операций, а УВС в целом имеет практически неограниченный объем памяти. Действительно, с помощью операции настройки можно подключить в УВС сколь угодно большое число элементарных машин, а с помощью операций приема и передачи информации в сочетании с операциями пересылки — реализовать и использовать неограниченную память. Наличие же универсального набора операций и неограниченной памяти позволяют свести УВС к универсальному программному автомату В.М. Глушкова, для которого доказана теорема о реализации нормального алгоритма Маркова.

ТЕОРЕМА 3. Всякий параллельный алгоритм [8] реализуется на УВС с частично переменной структурой, если в состав набора операций ЭМ, кроме операций ввода и вывода, пересылки, переадресации на $\pm I$ условного перехода по точному совпадению слов, операций настройки, приема и передачи информации, входит операция обобщенного условного перехода.

Пусть дана логическая схема параллельного алгоритма, которая представляет собой конечную строчку, составленную из символов операторов $Q_1, Q_2, \dots$, логических условий $P_1, P_2, \dots$ и правых полускобки $\downarrow, \downarrow, \dots$ и такую, что для каждой левой полускобки с индексом i в строке найдется одна и только одна правая полускобка с тем же индексом i и, обрат-

но, для каждой правой полускобки $\downarrow$ найдется одна и только одна левая с тем же индексом.

Каждый оператор Q_i состоит из совокупности M независимых операторов $Q_{i1}, Q_{i2}, \dots, Q_{iM}$ в том смысле, что независимо от порядка их выполнения результат не изменяется.

Логическое условие P_i состоит из совокупности логических условий $P_{i1}, P_{i2}, \dots, P_{iM}$.

Логическое условие P_i считается выполненным, если выполнены все условия P_{ij} .

Если логическое условие P_i выполнено, то совершается переход к выполнению очередного оператора, стоящего в строчке непосредственно справа, если не выполнено, то совершается переход к оператору, стоящему в строчке непосредственно справа от правой полускобки $\downarrow$.

В общем случае значение M может меняться при переходе от оператора к оператору в логической схеме.

Пусть дана универсальная вычислительная система с частично переменной структурой. Тогда при выполнении оператора Q_i в каждой из M элементарных машин будем выполнять соответственно операторы $Q_{i1}, \dots, Q_{iM}$. В силу независимости операторов $Q_{i1}, \dots, Q_{iM}$ их выполнение можно производить в одно и то же время.

Выполнение логического условия P_i производится следующим образом. В каждой из M элементарных машин с помощью операций условного перехода соответственно проверяются условия $P_{i1}, \dots, P_{iM}$. Проверка же того факта, что все логические условия $P_{i1}, \dots, P_{iM}$ выполнены, осуществляется с помощью операций обобщенного условного перехода. Изменение числа элементарных машин M при реализации логической схемы параллельного алгоритма может осуществляться с помощью операции настройки.

Теорема доказана.

2. Быстродействие (производительность) вычислительной системы. Скорость работы элементарной машины можно измерять числом рабочих циклов (цикл соответствует времени выполнения команды), осуществляемых ЭМ в течение одной секунды. Так как в общем случае рабочие циклы (в зависимости от команды) могут иметь различную длительность, то при определении быстродействия ЭМ подсчитывается среднее число команд в единицу времени в предположении, что ЭМ работает все время с оперативным за-

поминающим устройством, без учета ввода, вывода и работы с внешним запоминающим устройством. По аналогии с понятием номинального быстродействия автомата [2] введем понятие номинального быстродействия элементарной машины.

Номинальным быстродействием элементарной машины будем называть такое быстродействие, которое осуществляется ЭМ с оперативным запоминающим устройством и выражается средним числом операций (команд), выполняемых ЭМ в течение одной секунды.

Пусть дана универсальная вычислительная система с числом элементарных машин $M = 0, 1, 2, \dots$ и номинальным быстродействием U_H . Назовем номинальным быстродействием системы V_H произведение числа машин в системе на номинальное быстродействие ЭМ.

$$V_H = U_H \cdot M. \quad (3)$$

Номинальное быстродействие системы V_H является переменной величиной и изменяется дискретно с шагом дискретности U_H .

Если учитывать время, затрачиваемое на ввод и вывод информации, на обращение к внешним запоминающим устройствам, на обнаружение и устранение неисправности в системе, на обмен информацией между ЭМ в системе, на выполнение различных по своей сложности операций и т.д., то быстродействие ЭМ мы выразим числом выполняемых ею в единицу времени стандартных операций (через которую выражены все операции) и будем называть его эффективным быстродействием машины.

Естественно, что на эффективное быстродействие ЭМ влияют также выбранный метод решения данной задачи, качество программирования, выбранная система команд и т.д. Эффективное быстродействие зависит также от задач, которые решаются на ЭМ. В связи с этим целесообразно ввести понятие среднего эффективного быстродействия ЭМ.

Среднее эффективное быстродействие ЭМ определяется следующим образом. Выделяется конечное множество S типовых задач $S_1, S_2, \dots, S_m$, каждая из которых должна быть решена на вычислительной системе за время T . Пусть для каждой из задач общее число произведенных стандартных операций соответственно составляет $R_1, R_2, \dots, R_m$ и вероятности введения в систему каждой из данных задач будут: $p_1, p_2, \dots, p_m$. Тогда

для решения каждой из задач в заданное время T можно определить необходимое число машин M_i и эффективное быстродействие ЭМ системы $V_{\Sigma i}$ ($i = 1, \dots, m$):

$$V_{\Sigma i} = \frac{R_i}{T \cdot M_i}. \quad (4)$$

Используя вероятности $p_1, \dots, p_m$ и эффективные быстродействия $V_{\Sigma 1}, \dots, V_{\Sigma m}$ можно определить среднее эффективное быстродействие ЭМ $V_{\Sigma \text{ср}}$. Используя вероятности $p_1, \dots, p_m$ и необходимое количество элементарных машин, можно определить среднее число машин $M_{\text{ср}}$ в системе для решения данного множества задач.

Произведение среднего эффективного быстродействия ЭМ на среднее число машин будем называть эффективным быстродействием системы $V_{\Sigma \text{ср}}$:

$$V_{\Sigma \text{ср}} = V_{\Sigma \text{ср}} \cdot M_{\text{ср}}, \quad (5)$$

Для сравнения различных вычислительных систем введем критерий эффективности системы и критерий цены эффективного быстродействия системы. Под критерием эффективности системы $\mathcal{L}$ будем понимать отношение среднего эффективного быстродействия ЭМ к среднему числу машин в системе:

$$\mathcal{L} = \frac{V_{\Sigma \text{ср}}}{M_{\text{ср}}}. \quad (6)$$

Под критерием цены эффективного быстродействия системы Q будем понимать отношение среднего эффективного быстродействия к стоимости изготовления элементарной машины C_{Σ} :

$$Q = \frac{V_{\Sigma \text{ср}}}{C_{\Sigma}}. \quad (7)$$

Остановимся теперь на сравнении быстродействия УВС с быстродействием универсальной вычислительной машины, имеющей тот же набор операций и то же время их выполнения, что и в элементарной машине. При сравнении будем использовать два типа универсальных вычислительных машин (УВМ), различающихся по соотношению объемов внешней и оперативной памяти. Будем предполагать, что общий объем памяти УВМ для обоих типов равен объему памяти УВС. Для первого типа УВМ будем предполагать, что вся память является оперативной. Обозначим через $S'_{\text{по}}$ объем оперативной памяти УВМ (в двоичных разрядах), а через $S_{\Sigma \text{м}}$ - объем памяти элементарной машины. Тогда для первого

типа УВМ объем памяти $S_{MO} = S_{ЭМ} \cdot M$, а номинальное быстродействие УВМ $V_M = V_H$. Если предположить, что для широкого круга задач существуют параллельные алгоритмы, допускающие расчленение вычислений на M ветвей [9], то R - отношение быстродействия УВС к быстродействию УВМ - будет:

$$R = \frac{V_H}{V_M} = \frac{V_H \cdot M}{V_H} = M. \quad (8)$$

Для второго типа УВМ будем предполагать, что общий объем памяти равен

$$S_M = S_{MO} + S_{MB}, \quad (9)$$

где $S_{MO} = S_{ЭМ}$, а объем внешней памяти $S_{MB} = (M-1) S_{ЭМ}$. Пусть время выборки слова из внешней памяти составляет ε_B , а ε_O - время, затрачиваемое в среднем на обработку этого слова. Обозначим через λ отношение времени выборки слова из внешней памяти ко времени обработки слова

$$\lambda = \frac{\varepsilon_B}{\varepsilon_O}. \quad (10)$$

Тогда отношение быстродействия УВС к быстродействию УВМ составит:

$$R = \frac{V_H}{V_M} = (1 + \lambda) \cdot M. \quad (11)$$

3. Надежность вычислительной системы. Под надежностью системы будем понимать способность сохранять заданные свойства при заданных условиях работы в течение определенного периода времени [10].

Количественной оценкой надежности является вероятность (P) того, что система обладает заданными свойствами при заданных условиях работы в течение определенного времени (T).

Работу системы можно характеризовать также вероятностью выхода ее из строя (Q) за определенный промежуток времени (T). Так как ненадежность системы и ее надежность являются событиями несовместимыми, то

$$Q = 1 - P. \quad (12)$$

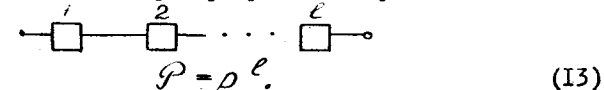
Пусть p - вероятность исправной работы элементарной машины, а $q = 1 - p$ - вероятность ее выхода из строя. В силу однородности структуры элементарных машин УВС, а также системы коммутации между ними можно считать, что выход из строя одной элементарной машины не сказывается на работе остальных ЭМ. В связи с этим будем предполагать, что выходы элементарных ма-

шин из строя являются независимыми событиями. Благодаря переменной структуре и возможности обнаружить неисправности ЭМ программным способом, процесс обнаружения неисправной ЭМ и замены ее исправной может происходить автоматически и при пренебрежимо малых затратах времени.

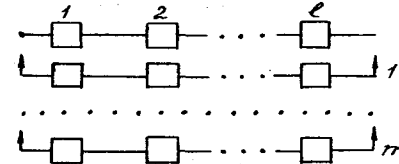
Пусть дана вычислительная система из M элементарных машин. Пусть для решения задачи требуется одновременная работа ℓ машин.

Определим надежность вычислительной системы с различными схемами резервирования (дублирования) [11], [12].

а) Надежность УВС без резервирования и ремонта:



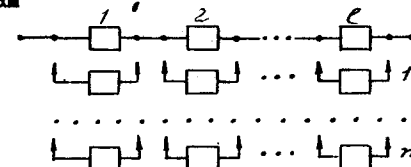
б) Надежность УВС при резервировании всей системы в целом:



Пусть применяется $m = \frac{M}{\ell} - 1$ резервных цепей для всей системы в целом. Тогда надежность системы будет:

$$P = 1 - (1 - p^{\ell})^{m+1}. \quad (14)$$

в) Надежность УВС при резервировании системы по элементарным машинам



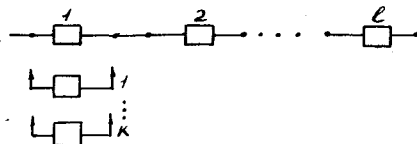
Пусть для каждой элементарной машины используется $m = \frac{M}{\ell} - 1$ резервных машин. Тогда надежность системы будет

$$P = [1 - (1 - p)^{m+1}]^{\ell}. \quad (15)$$

г) Надежность УВС при автоматическом ремонте неисправностей в системе.

При выходе из строя одной из ℓ машин неисправная ЭМ исключается из системы, к оставшимся исправным машинам добавляется одна машина из резерва. Между ЭМ системы производится пе-

пераспределение задачи. По мере выхода из строя машин в системе из ℓ машин процесс повторяется



Пусть $K = M - \ell$ машин используется в качестве резерва. Тогда надежность системы будет

$$P = \sum_{\ell=0}^{K=M} C_M^{\ell} p^{\ell} \cdot (1-p)^K. \quad (I6)$$

4. Стоимость вычислительной системы. Стоимость УВС может быть выражена либо общим количеством эквивалентных элементов оборудования, взятых за единицу измерения, либо в других единицах измерения (например рублях). Будем сравнивать стоимость C_C вычислительной системы со стоимостью C_M универсальной вычислительной машины, которая эквивалентна вычислительной системе по своему быстродействию, объему памяти и надежности.

Коэффициентом экономичности системы K_C будем называть отношение стоимости машины к стоимости системы

$$K_C = \frac{C_M}{C_C}. \quad (I7)$$

Рассмотрим два случая:

- 1) все элементы одинаковы,
- 2) элементы памяти машины являются более простыми по сравнению с остальными элементами.

Пусть все элементы, из которых выполнены система и машина, одинаковы, т.е. устройство памяти, арифметические устройства, устройства управления и т.д. составлены из одних и тех же элементов. Пусть C_n - общее число элементов в памяти элементарной машины, а C_A - число элементов во всей остальной части ЭМ. Тогда коэффициент экономичности K_C для системы из M машин будет

$$K_C = \frac{C_M}{C_C} = \frac{M \cdot C_n + C_A}{M \cdot C_n + M \cdot C_A} = \frac{C_n + \frac{C_A}{M}}{C_n + C_A} = \frac{1 + \beta/M}{1 + \beta}, \quad (I8)$$

где $\beta = \frac{C_A}{C_n}$.

Но так как в случае применения одинаковых элементов $\beta \ll 1$, то

$$K_C \approx \frac{1}{1 + \beta}. \quad (I9)$$

Обозначим отношение стоимости элемента машины к стоимости элемента системы через α .

Тогда

$$K_C = \frac{(1 + \beta/M)\alpha}{1 + \beta} \approx \frac{\alpha}{1 + \beta}, \quad (20)$$

2) Пусть элементы собственно памяти являются более простыми по сравнению с другими элементами, из которых реализуется система и машина.

Обозначим через C_0 - общее число элементов, затрачиваемых на выборку слов из памяти ЭМ, запись в память, считывание из памяти. Элементы, затрачиваемые на реализацию собственно памяти ЭМ, не будем учитывать при определении K_C . Тогда коэффициент экономичности K_C для системы из M машин будет

$$K_C = \frac{C_M}{C_C} = \frac{M \cdot C_0 + C_A}{M \cdot C_0 + M \cdot C_A} = \frac{C_0 + C_A/M}{C_0 + C_A} = \frac{1 + \gamma/M}{1 + \gamma}, \quad (21)$$

где $\gamma = \frac{C_A}{C_0}$.

Но так как, практически, $\gamma \approx 1$, то

$$K_C \approx \frac{1}{1 + \gamma} \quad (22)$$

или

$$K_C = \frac{(1 + \gamma/M)\alpha}{1 + \gamma} \approx \frac{\alpha}{1 + \gamma}. \quad (23)$$

5. Синхронизация вычислительной системы. Синхронизация работы УВС может быть осуществлена многими способами. В качестве примера приведем два из них: 1) способ командной синхронизации, 2) способ подпрограммной синхронизации.

При командной синхронизации работа системы происходит следующим образом. В каждой элементарной машине выполняется команда. После выполнения команды работа ЭМ прекращается до завершения операций во всех остальных ЭМ системы. Чтобы приступить к выполнению очередной команды, необходимо завершить выполнения предыдущей команды во всех машинах.

При подпрограммном способе синхронизации в каждой ЭМ независимо выполняется своя подпрограмма. После завершения подпрограммы во всех машинах по команде обобщенного условного перехода производится переход к новым подпрограммам.

6. Синтез вычислительной системы. Как уже описывалось выше, для построения универсальной вычислительной системы необходимо задать элементарную машину и размерность сетки. Тогда

синтез вычислительной системы можно свести к двум этапам:

1. К синтезу элементарной машины, необходимой для создания ВС с заданной размерностью сетки.

2. К синтезу универсальной ВС из элементарных машин для решения данной задачи или данного класса задач.

Вполне очевидно, что на I этапе, в силу того, что элементарная машина фактически является конечным автоматом, применимы все методы синтеза, развиваемые в теории конечных автоматов.

На II этапе для решения данной задачи или данного класса реализация ВС из ЭМ может быть осуществлена программной настройкой, поэтому на II этапе приемлемы методы синтеза, развиваемые в теории программирования.

§ 3. Схемы реализации операции настройки

Отличительной чертой УВС является возможность изменения структуры системы программным способом с помощью операции настройки. Рассмотрим некоторые варианты реализации схем настройки. В зависимости от структуры будем различать схемы с фиксированной и переменной системами настройки. При фиксированной системе настройки структура схемы, управляющей настройкой, не может программно изменяться. При переменной системе настройки можно программно перестраивать структуру схемы настройки. Будем различать схемы с произвольной системой выборки настраиваемых ЭМ (произвольная настройка) и схемы с упорядоченной выборкой ЭМ (упорядоченная настройка).

Будем также предполагать, что управление настройкой может осуществляться на каждом этапе решения задачи любой ЭМ. Возможно также управление настройкой одновременно несколькими ЭМ.

Пусть в вычислительной системе из M элементарных машин, каждая ЭМ имеет один входной канал для приема информации от других ЭМ и один выходной канал для передачи информации в другие ЭМ. Пусть в каждый данный момент времени входной канал может быть соединен только с одним выходом любой ЭМ вычислительной системы. Выход же данной ЭМ может соединяться с любым числом входных каналов других ЭМ. Таким образом, в данный момент времени в элементарную машину вводится информация только от

какой-либо одной ЭМ и выводится из данной ЭМ в любое их число.

Коммутацию в вычислительной системе будем представлять в виде квадратной матрицы:

$$\begin{bmatrix} C_{11}(t) & C_{12}(t) & \dots & C_{1\ell}(t) & \dots & C_{1M}(t) \\ C_{21}(t) & C_{22}(t) & \dots & C_{2\ell}(t) & \dots & C_{2M}(t) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ C_{\ell 1}(t) & C_{\ell 2}(t) & \dots & C_{\ell \ell}(t) & \dots & C_{\ell M}(t) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ C_{M1}(t) & C_{M2}(t) & \dots & C_{M\ell}(t) & \dots & C_{MM}(t) \end{bmatrix} \quad t=0,1,2,\dots \quad (24)$$

элементы которой $C_{\ell k}(t)$ принимают значение 0 или 1. Равенство $C_{\ell k}(t)=1$ означает, что входной канал ЭМ m_k присоединен в момент времени t к выходному m_ℓ . Матрица $[C_{\ell k}(t)]$ обладает следующими свойствами:

$$\sum_{i=1}^M C_{ki} \& C_{kj} = 0; \quad i \neq j \quad (25)$$

$$C_{\ell \ell} = 0. \quad k=\ell \quad (26)$$

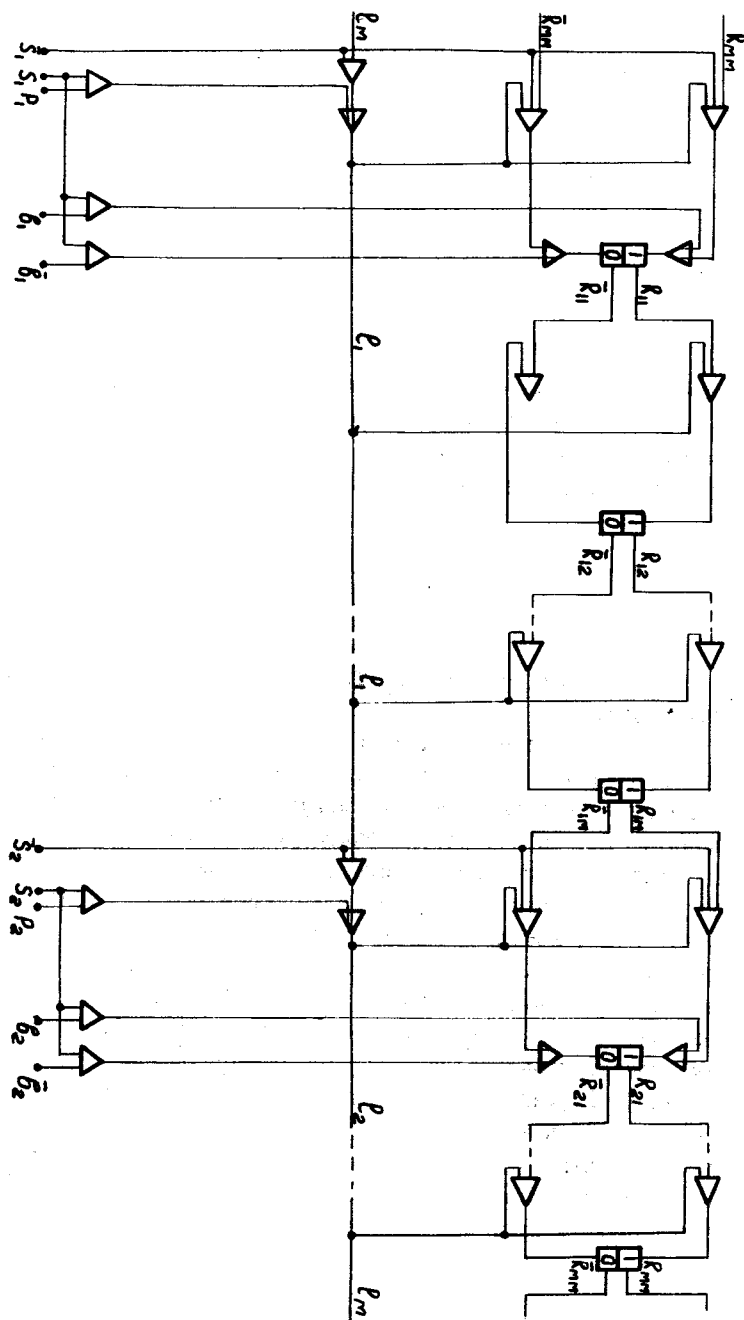
Рассмотрим несколько методов настройки УВС с фиксированной системой.

I. Настройка УВС путем поразрядного сдвига с накоплением. Поставим в соответствие каждому элементу матрицы $C_{\ell k}(t)$ двоичный элемент памяти $R_{\ell k}$ и построим логическую сеть (рис. 4), в которой элемент $R_{\ell k}$ принимает сигналы от элемента $R_{\ell, \ell-1}$ ($\ell \neq 1$) либо от элемента $R_{\ell-1, M}$ ($\ell=1$) (исключение составляет элемент R_{11} , который принимает сигналы от элемента R_{MM}). Элементам $R_{u1}, R_{u2}, \dots, R_{uM}$ ($u=1, \dots, M$) ставится в соответствие ЭМ m_u , выходы которой $S_u, S_u, \bar{S}_u, \bar{S}_u$ и P поданы на элемент R_{u1} . Данная схема предусматривает, что в каждый данный момент любая из ЭМ может стать управляющей. Для этого она должна содержать в своей памяти все M^2 элементов матрицы $[C_{\ell k}(t)]$, которые в ходе настройки вводятся в соответствующие элементы памяти $R_{\ell k}$. Производится это следующим образом. Пусть какая-либо из машин, например m_u , будет управляющей, что соответствует значениям:

$$S_u=1, \quad S_{k \neq u}=0 \quad (k=1,2,\dots,u-1,u+1,\dots,M). \quad (27)$$

В этом случае поступление сигналов из предыдущей машины m_{u-1}

Рис. 4.



в машину m_u прекращается и открываются пути для подачи через выходы $b_u, \bar{b}_u$ соответствующих значений элементов матрицы $C_{ke}(t)$ и тактовых импульсов p_u в схему настройки. На первом такте в R_{u1} поступает элемент матрицы $C_{u-1,M}$, на втором - $C_{u-1,M-1}$, на третьем - $C_{u-1,M-2}$ и т.д. и, наконец, на такте M^2 - элемент C_{u1} . Одновременно на каждом такте все R_{ke} будут передавать свое содержимое соседу справа. В результате за M^2 тактов все элементы C_{ke} будут введены в соответствующие элементы памяти R_{ke} .

Данная логическая сеть может быть описана с помощью следующей системы уравнений:

$$\left. \begin{aligned} R_{ke}(t+1) &= R_{k\ell-1}(t) \cdot \ell_k(t) & (\ell \neq 1) \\ \bar{R}_{ke}(t+1) &= \bar{R}_{k\ell-1}(t) \cdot \ell_k(t) & (\ell \neq 1) \\ R_{k1}(t+1) &= R_{k-1,M}(t) \cdot \bar{S}_k(t) \vee b_k(t) \cdot S_k(t) & (k \neq 1) \\ \bar{R}_{k1}(t+1) &= \bar{R}_{k-1,M}(t) \cdot \bar{S}_k(t) \vee \bar{b}_k(t) \cdot S_k(t) & (k \neq 1) \\ R_H(t+1) &= R_{MM}(t) \cdot \bar{S}_1(t) \vee b_1(t) \cdot S_1(t) \\ \bar{R}_H(t+1) &= \bar{R}_{MM}(t) \cdot \bar{S}_1(t) \vee \bar{b}_1(t) \cdot S_1(t) \\ \ell_k(t) &= \ell_{k-1}(t) \cdot \bar{S}_k(t) \vee p_k(t) S_k(t) & (k \neq 1) \\ \ell_1(t) &= \ell_H(t) \cdot \bar{S}_1(t) \vee p_1(t) S_1(t) \end{aligned} \right\} (28)$$

Время настройки может быть уменьшено, если управление настройкой осуществляется сразу несколькими ЭМ. В частности, если в каждой ЭМ m_k содержится k -ая строка матрицы $[C_{ke}(t)]$, то все M машин могут быть сделаны управляющими, для чего надо положить все $S_k(t) = 1$ ($k = 1, 2, \dots, M$). Тогда настройка может быть осуществлена всего за M тактов.

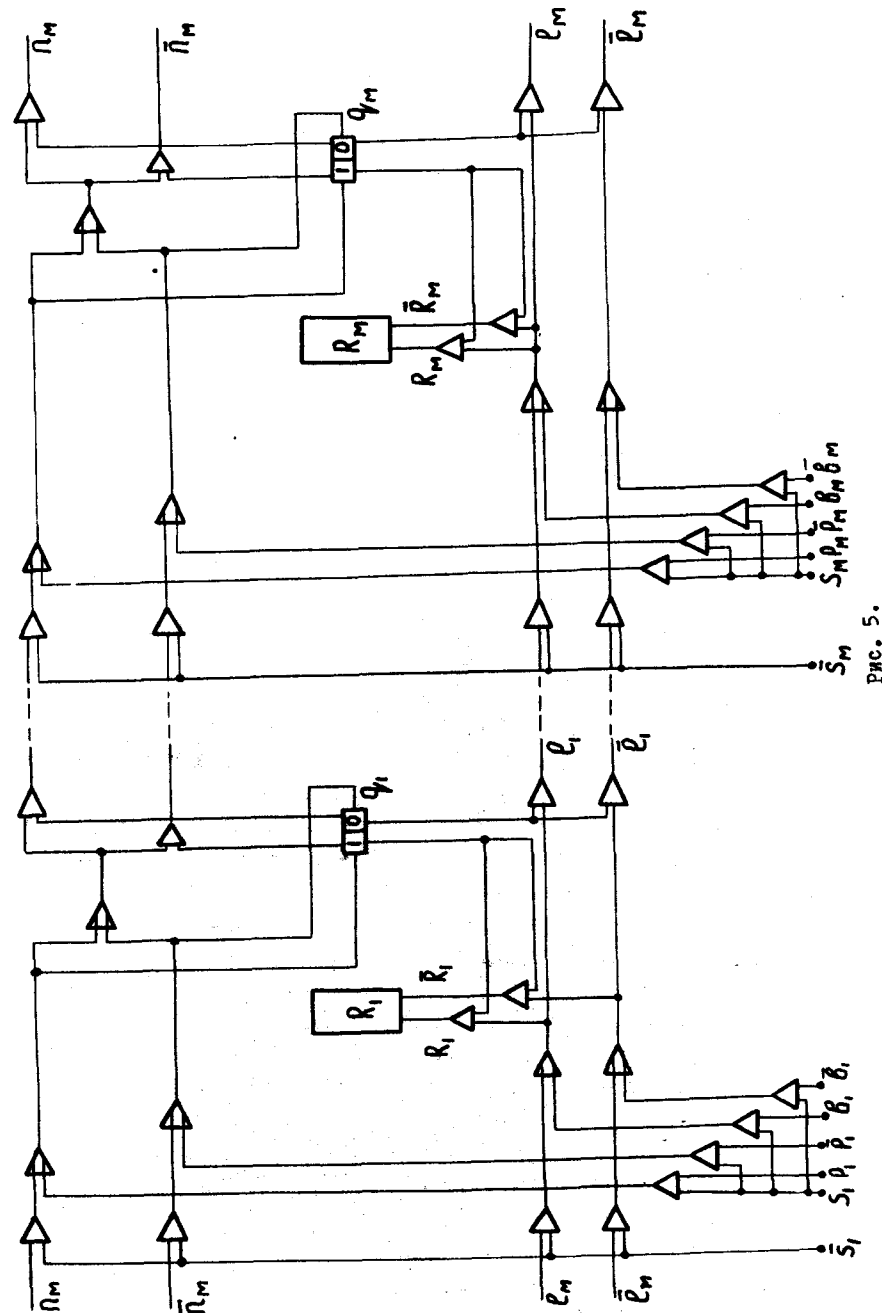
2. Пошаговая настройка. Данный метод использует свойство матрицы $[C_{ke}(t)]$ (25), согласно которому в каждой строке матрицы не может быть более одного ненулевого элемента. Это позволяет определить матрицу $[C_{ke}(t)]$ не более чем M числами, каждое из которых обозначает адрес ненулевого элемента матрицы: $a_1, a_2, \dots, a_M$. Каждый из a_k будет, очевидно, иметь $m = \log_2 M$ двоичных разрядов. Пусть в каж-

дой ЭМ m_k имеется память R_k объемом m двоичных разрядов для хранения a_k . Тогда операция настройки сведется к записке в память R_k соответствующих a_k . Это может быть реализовано, например, с помощью схемы (рис. 5), составленной для следующей системы уравнений:

$$\left. \begin{aligned} R_k(t) &= q_k(t) \cdot [e_{k-1}(t) \cdot \bar{s}_k(t) \vee b_k(t) \cdot s_k(t)] \\ \bar{R}_k(t) &= q_k(t) \cdot [\bar{e}_{k-1}(t) \cdot \bar{s}_k(t) \vee \bar{b}_k(t) \cdot s_k(t)] \\ e_k(t) &= \bar{q}_k(t) \cdot [e_{k-1}(t) \cdot \bar{s}_k(t) \vee b_k(t) \cdot s_k(t)] \\ \bar{e}_k(t) &= \bar{q}_k(t) \cdot [\bar{e}_{k-1}(t) \cdot \bar{s}_k(t) \vee \bar{b}_k(t) \cdot s_k(t)] \\ q_k(t+1) &= r_{k-1}(t) \cdot \bar{s}_k(t) \vee p_k(t) \cdot s_k(t) \vee q_k(t) \\ \bar{q}_k(t+1) &= \bar{r}_{k-1}(t) \cdot \bar{s}_k(t) \vee \bar{p}_k(t) \cdot s_k(t) \vee \bar{q}_k(t) \\ r_k(t) &= q_k(t) \cdot [r_{k-1}(t) \cdot \bar{s}_k(t) \vee p_k(t) \cdot s_k(t) \vee \bar{r}_{k-1}(t) \cdot \bar{s}_k(t) \vee \bar{p}_k(t) \cdot s_k(t)] \\ \bar{r}_k(t) &= \bar{q}_k(t) \cdot [\bar{r}_{k-1}(t) \cdot \bar{s}_k(t) \vee p_k(t) \cdot s_k(t) \vee \bar{r}_{k-1}(t) \cdot \bar{s}_k(t) \vee \bar{p}_k(t) \cdot s_k(t)] \end{aligned} \right\} (29)$$

Как и в предыдущем случае, все машины при выполнении операции настройки оказываются расположенными вдоль замкнутой линии. В каждой машине m_k имеется триггер q_k , который находится в состоянии $q_k=1$ в той ЭМ, которая в данный момент настраивается. Тогда настройка выполняется следующим образом.

Пусть машина m_u будет управляющей и содержит в своей памяти значения всех $a_1, a_2, \dots, a_m$. И пусть, как и ранее, выполняется условие (27). Соблюдение этого условия означает, что линии схемы настройки управляющей машины заперты для внешних сигналов и через входы $b_u, \bar{b}_u$ подключены к источникам настроечной информации, а через входы $p_u, \bar{p}_u$ к источникам тактовых толкающих импульсов. Последние подаются через интервалы времени $m \cdot t$, где t - рабочий такт, необходимый для ввода одного двоичного разряда кода. Первый толкающий импульс подается на вход p_u , все $M-1$ последующих импульсов - на вход $\bar{p}_u$. Первый толкающий импульс устанавливает значение триггера $q_u=1$, тем самым открываются входы памяти R_u , и в нее за m последовательных тактов вводится код a_u . Второй толкающий импульс устанавливает значение триггера $q_{u+1}=1$ следующей машины и гасит триггер q_u . Тогда очередной код a_{u+1} воспримется



только машиной m_{u+1} . Третий толкающий импульс возмущает триггер q_{u+2} и т.д. Таким образом, через M шагов по m тактов каждый все машины будут настроены.

С помощью данной схемы можно изменять настройку только у некоторых произвольно выбранных ЭМ. Можно также одновременно настраивать несколько ЭМ с одинаковыми значениями α_k . Достигается это подбором соответствующих последовательностей толкающих импульсов. Например, если ЭМ m_k не нужно перенастраивать, то, когда подойдет ее очередь, подаются подряд два толкающих импульса. Если, например, $\alpha_{m/2+1} = \alpha_1, \alpha_{m/2+2} = \alpha_2, \dots, \alpha_M = \alpha_{m/2}$, то сначала подается серия толкающих импульсов $p(t=1)=1, p(t=2)=0, p(t=3)=0, \dots, p(t=M/2)=0, p(t=M/2+1)=1$, а затем обычным порядком вводятся коды $\alpha_1, \dots, \alpha_{m/2}$.

Данная схема, как и предыдущая, допускает также одновременную настройку с помощью нескольких управляющих машин.

3. Координатная настройка. При двух предыдущих методах порядок настройки ЭМ был задан жестко. Но возможен метод и с произвольным порядком настройки.

Известно, что в оперативных памяти ЭВМ широко применяются схемы, допускающие произвольный порядок записи и считывания информации. В этих схемах выборка необходимой ячейки памяти достигается подачей ее адреса в соответствующие координатные шины в виде определенной совокупности сигналов. Аналогичным образом может быть построена и схема с произвольным порядком выборки настраиваемых ЭМ. Для этого матрице $[C_{ke}(t)]$ ставится в соответствие матричная память объемом M^2 двоичных ячеек. В этом случае при двумерной системе координат для задания адреса любой ячейки потребуется регистр R_a объемом $2m$ разрядов, где $m = \log_2 M$ (рис. 6а). Если нужно запомнить только адреса ненулевых элементов, то можно обойтись памятью объемом $M \cdot m$ двоичных ячеек и для задания адреса достаточно будет регистра объемом m двоичных разрядов (рис. 6б). Однако в этом случае объем R_3 увеличится до m разрядов вместо одного, как в первом случае.

При координатном методе настройкой управляет одна из ЭМ, например m_u и настройка выполняется следующим образом. Из ЭМ m_u в регистр R_a подается адрес настраиваемого элемента, а в регистр R_3 — соответствующий код настройки. В каждый момент времени настраивается только один элемент матрицы. Время настройки будет при этом составлять либо M^2 (когда запомина-

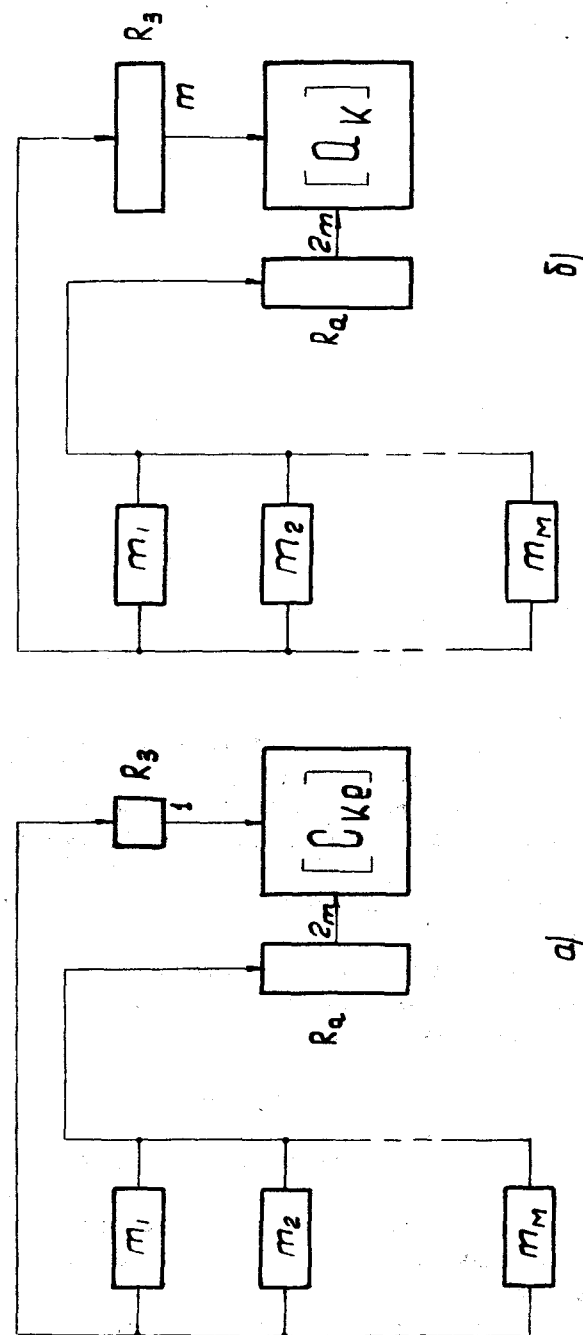


Рис. 6.

ются $C_{ке}$, либо M -тактов (когда запоминаются a_k).

4. Адресная настройка. При этом методе код настройки, сопровождаемый адресом, посылается из управляющей ЭМ в общую систему связи, соединяющую все ЭМ. Каждая ЭМ анализирует адрес, сопровождающий код настройки, и при совпадении этого адреса с ее собственным воспринимает код настройки. В результате изменяются соответствующие коммутации в блоке K (рис. 7а). Вход и выход каждой ЭМ подключены к общей шине O . К коммутатору K каждой ЭМ подходит M входных каналов. В каждый данный момент времени к ЭМ с помощью коммутатора K может быть подключено не более одного входного канала.

Схема коммутатора состоит из регистра R , управляющего коммутациями, и собственно коммутатора, в котором осуществляется подключение к входу ЭМ только одного входного канала, определяемого содержимым регистра R (рис. 7б). Информация в регистр R подается из ЭМ.

Адресная настройка осуществляется по следующей схеме.

Пусть в некоторой машине m_u хранится информация, определяющая коммутацию вычислительной системы. Процесс настройки распадается на две фазы:

1) Из ЭМ m_u по каналу O информация настройки каждой ЭМ передается в соответствующую ЭМ. Это можно осуществить с помощью специальных подпрограмм, хранящихся в каждой ЭМ. Информация настройки выделяется из общего потока по признаку её принадлежности данной ЭМ и помещается в памяти ЭМ для управления коммутацией на второй фазе. 2) Информация настройки по специальной команде подается на регистр коммутаторного устройства. В результате осуществляется подключение соответствующего входного канала к данной ЭМ.

При адресном способе настройки можно одновременно настраивать все ЭМ, если у них нужно установить одинаковую систему соединений. Возможна также одновременная настройка с помощью нескольких управляющих ЭМ. Для этого, воспользовавшись ранее установленной системой коммутации, нужно передать в различные ЭМ соответствующую настроечную информацию, а затем дать сигнал перехода к новой системе коммутации, т.е. сначала для всех ЭМ выполнить первую фазу настройки, а затем вторую.

Адресный способ отличается большой гибкостью и коммутация может быть реализована за сравнительно небольшое время. В тоже время он не требует больших дополнительных схем. Метод настройки с произвольной выборкой особенно выгоден, если изменяется структура только некоторых ЭМ.

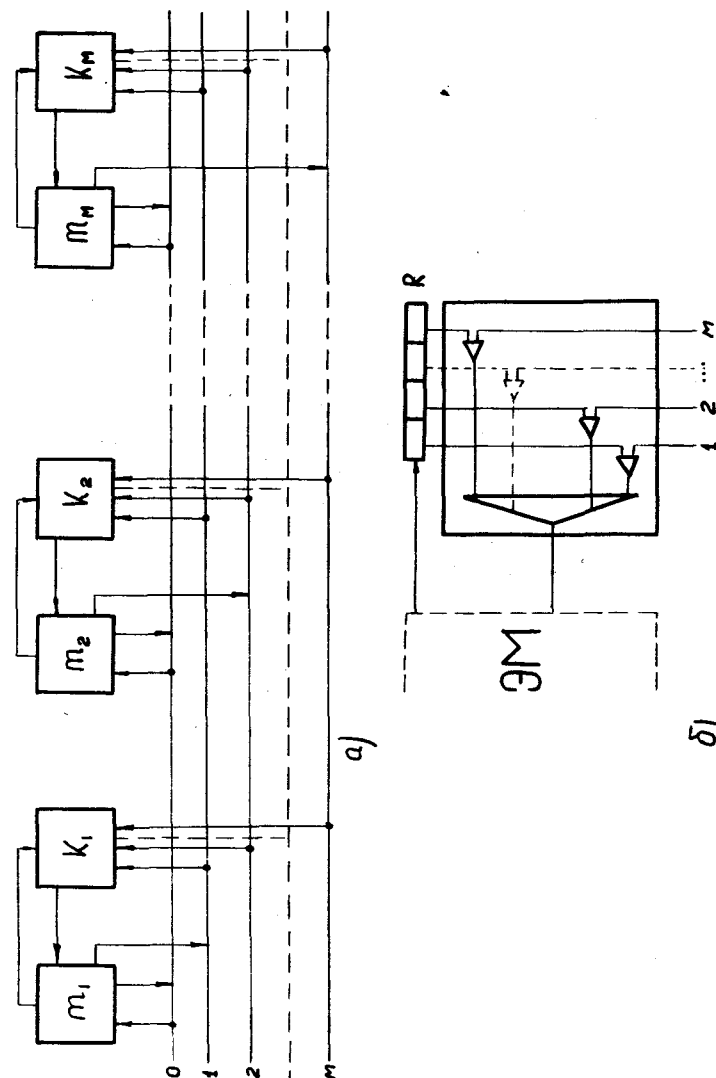


Рис. 7.

Все четыре рассмотренных метода имеют фиксированную систему настройки, что делает их недостаточно надежными. Выход из строя одной или нескольких ЭМ или локальные неисправности каналов связи могут существенно снизить производительность УВС и даже сделать ее непригодной для дальнейшего использования без соответствующего ремонта. При большом же числе ЭМ ремонт значительно сократит полезное время работы УВС. Кроме того, ремонт УВС, изготовленных, как единое целое, на основе микроинтегризации, может встретить серьезные затруднения, и от него, может быть, вообще придется отказаться. Всё это требует принятия мер по повышению надежности УВС. Безусловно, первым необходимым требованием является увеличение надежности ЭМ и каналов связи. Однако одни эти меры едва ли смогут обеспечить надежную работу таких многомашинных систем, как УВС.

Поэтому необходимо предпринять шаги, которые обеспечили бы сохранение производительности УВС на достаточно высоком уровне при выходе из строя отдельных ЭМ и каналов связи. Одной из таких мер является переход к переменной структуре схем настройки, благодаря чему неисправность локализуется в небольшой области и приводит к исключению из УВС только некоторого числа ЭМ при сохранении всех остальных в рабочем состоянии.

Рассмотрим в качестве примера двумерную УВС. Пусть каждая ЭМ с координатами (i, j) получает коды настройки от любой, но в каждый данный момент только одной соседней ЭМ, и передает коды любой соседней ЭМ с координатами $(i, j-1)$, $(i-1, j)$, $(i, j+1)$, $(i+1, j)$. В дальнейшем, для простоты, будем данную ЭМ и соответствующие ей параметры обозначать индексом 0, а ее соседей индексами 1, 2, 3, 4, соответственно.

Для задания системы коммутаций достаточно, чтобы для каждой ЭМ было указано, в какую соседнюю ЭМ она должна передавать свою информацию. Это позволяет все ЭМ системы объединить в единую цепочку передачи информации, начиная с некоторой управляющей ЭМ, либо образовать несколько таких цепочек, если имеется несколько иерархических уровней управления. Порядок следования ЭМ в цепочках должен храниться в памяти управляющих ЭМ и последовательно передаваться в другие ЭМ по мере прокладывания пути.

Схема, реализующая данный вариант настройки, содержит регистр R_H , в котором запоминается номер последующей ЭМ в цепочке передачи информации (рис. 8). Изменением состояния регистра R_H управляет триггер активности T_a . В состоянии $T_a=1$

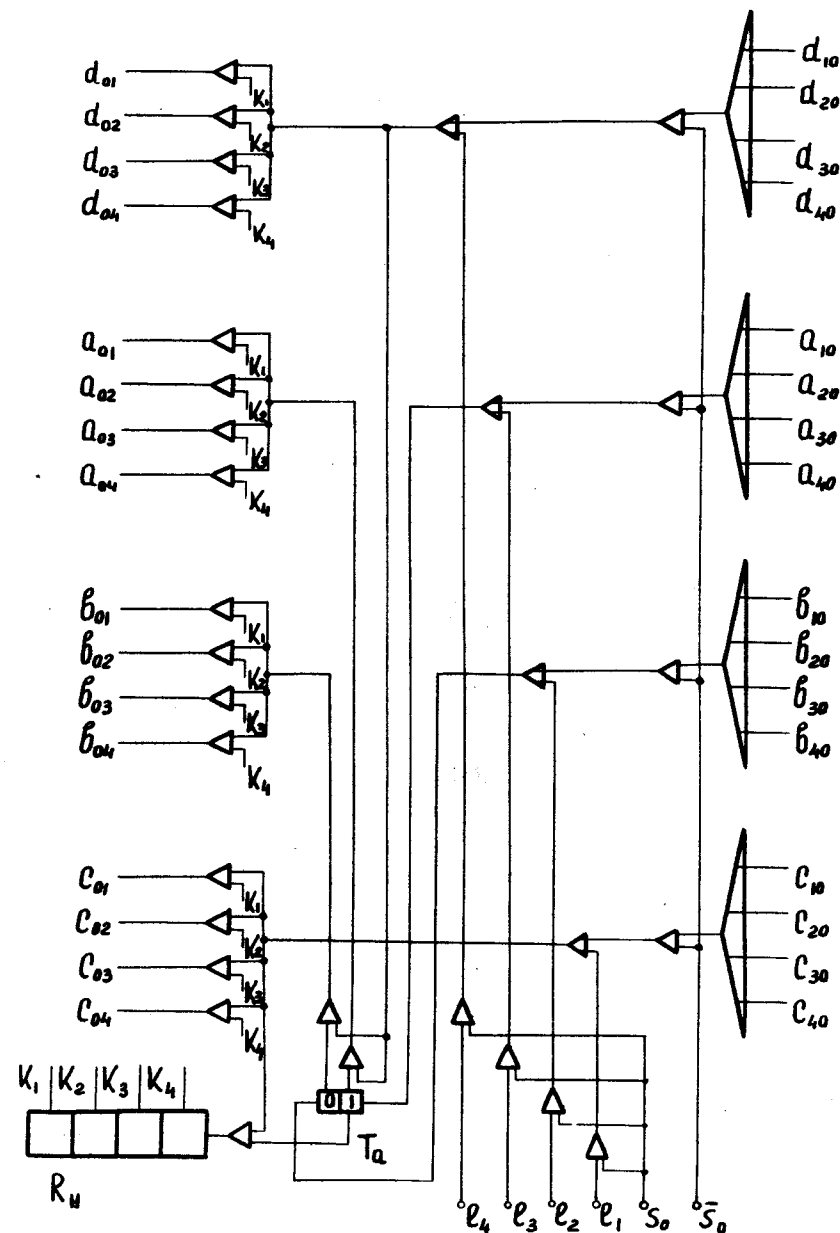


Рис. 8.

код настройки, поступающий извне через один из входов C_{01}, C_{20}, C_{30} или C_{40} , либо из памяти самой ЭМ через вход e_1 , посылается в регистр R_H , чем и производится настройка данной ЭМ.

Операция настройки начинается с настройки регистра R_H самой управляющей ЭМ. Для этого через вход e_2 подается сигнал, переводящий триггер активности в состояние $T_a = 1$. После этого в регистр R_H засылается код настройки. Последний отпирает выходы только к одной из соседних ЭМ. Подачей импульса через вход e_4 возбуждается триггер следующей ЭМ (через выходы d_{01}, d_{02}, d_{03} или d_{04}) и одновременно гасится триггер T_a данной ЭМ. Затем через выходы C_{01}, C_{02}, C_{03} или C_{04} передается код настройки следующей ЭМ и т.д., пока не будет образована цепочка из всех ЭМ, либо, если настройку желательно прекратить на какой-либо ЭМ, со входа e_3 транзитом через выходы $v_{01}, v_{02}, v_{03}, v_{04}$ всех предшествующих ЭМ посылается сигнал сброса, приводящий триггер активности данной ЭМ в состояние $T_a = 0$ и не возбуждающий, таким образом, следующей машины.

Нетрудно видеть, что данная схема позволяет реализовать переменную структуру настройки и посредством изменения порядка следования машин в цепочке изолировать неисправные ЭМ, либо поврежденные линии связи.

§ 4. Основные типы УВС

В связи с изучением УВС возникает необходимость в систематизации и классификации различных типов УВС по конструктивным признакам. В настоящее время, когда только начинается систематическое исследование вычислительных систем, дать достаточно полную классификацию УВС трудно, тем более, что подобной классификации не существует и для ЭВМ. Поэтому ограничимся некоторым, не претендующим на полноту, вариантом классификации, в основу которого положены следующие признаки: характер пространственного размещения ЭМ (система координат и состояние между ЭМ); тип коммутатора; организация схем, управляющих изменением структуры ЭМ и коммутатора (структура схемы настройки, способ выборки настраиваемых ЭМ); способ управления ходом вычислений; характер изменения структуры; система ввода и вывода данных и т.п.

По характеру пространственного размещения ЭМ будем раз-

личать одно-, двух- и многомерные УВС, в которых ЭМ можно представить в виде точек на линии, плоскости и многомерном пространстве. В первом случае для задания ЭМ достаточно одной координаты. Каждая ЭМ будет иметь двух соседей. Во втором случае положение ЭМ задается двумя координатами и каждая ЭМ имеет четырех соседей. В n -мерном случае положение ЭМ задается n координатами и каждая ЭМ имеет $2n$ соседей.

По расстоянию между ЭМ будем различать сосредоточенные и распределенные УВС. У сосредоточенных УВС время распространения сигналов между двумя наиболее удаленными друг от друга ЭМ существенно меньше времени выполнения операции элементарной машины. В этом случае запаздывание сигналов в каналах связи практически не влияет на производительность УВС. У распределенных УВС время распространения сигналов между ЭМ сравнимо, либо превышает время выполнения операции ЭМ. В этом случае запаздывание сигналов в каналах связи может существенно сказываться на производительности УВС, если не предприняты специальные меры.

По типу коммутатора будем различать УВС с коммутаторами, реализующими соединения между ЭМ по типу полного графа, когда каждая ЭМ соединена каналом связи с каждой ЭМ; и по типу R_n графа, когда каждая ЭМ соединена только со своими соседями в n -мерной системе координат (см. [13]).

В зависимости от структуры схемы настройки будем различать УВС с фиксированной и переменной системами настроек.

В зависимости от способа выборки настраиваемых ЭМ будем различать УВС с произвольной системой выборки настраиваемых ЭМ (произвольная настройка) и с упорядоченной системой выборки (упорядоченная настройка). При произвольной настройке в каждый данный момент осуществляется настройка одной ЭМ и её соединений с другими ЭМ. При упорядоченной настройке изменение структуры ЭМ осуществляется в строго заданной последовательности и обычно сразу для всех ЭМ УВС или групп ЭМ.

По способу управления ходом вычислений будем различать УВС с иерархической и однородной структурами управления. При иерархической

структуре управления программа работы УВС находится либо в одной ЭМ, которая посылает команды управления во все остальные ЭМ, либо в определенной их группе. В последнем случае устанавливаются иерархическая подчиненность и правила приоритета команд управления, поступающих от различных ЭМ. При однородной структуре управления все ЭМ равнозначны, программа решения задачи распределена между всеми ЭМ, и изменение хода вычислений, в зависимости от полученных результатов, осуществляется с помощью операции обобщенного условного перехода, учитывающей состояния всех ЭМ.

По характеру изменения структуры будем различать УВС с частично изменяемой структурой, когда изменяется только структура коммутатора, и с полностью изменяемой структурой, когда изменяются как структура коммутатора, так и структура ЭМ.

По характеру обмена информацией между ЭМ внутри УВС и между УВС и внешними объектами будем различать УВС с последовательной, параллельной и последовательно-параллельной передачей слов.

При последовательной передаче слова передаются поразрядно, т.е. информация передается разряд за разрядом от одной ЭМ к другой.

При параллельной передаче производится обмен между ЭМ в параллельном коде всех разрядов слова за один такт.

При последовательно-параллельной передаче обмен информацией между ЭМ производится частями слов. Для каждой части слова осуществляется одновременная передача всех разрядов за один такт.

Большинство приведенных выше признаков являются независимыми и могут сочетаться друг с другом в произвольных комбинациях. Хотя этими признаками классификация далеко не исчерпывается, однако их комбинации охватывают, по-видимому, все основные типы возможных вариантов УВС.

§ 5. Распределенные универсальные вычислительные системы

При построении распределенных универсальных вычислитель-

ных систем необходимо учитывать то обстоятельство, что элементарные машины расположены на значительных расстояниях друг от друга. В связи с этим приходится предпринимать специальные меры, чтобы запаздывание сигналов в каналах связи не сказывалось существенно на производительности УВС [14]. Из-за относительно высокой стоимости каналов связи приходится ограничиваться такими вариантами построения УВС, у которых число каналов связи между ЭМ минимально. С учетом вышесказанного рассмотрим некоторые варианты реализации распределенных УВС.

1. Одномерные распределенные УВС. Известный теоретический и практический интерес представляет рассмотрение одномерных УВС, в частности распределенных, образованных из обычных далеко расположенных друг от друга ЭВМ, когда существенно стоимость каналов связи.

Построение УВС из обычных ЭВМ, как уже указывалось ранее [1], может рассматриваться как первый этап создания УВС и явиться одним из способов решения сложных задач, требующих выполнения большего числа операций, чем это допускается в отдельных ЭВМ (в том числе и наиболее производительных).

Пусть имеется M вычислительных машин, произвольно расположенных относительно друг друга. Образует из них последовательность, придав машинам номера от 1 до M . Будем считать, что каждая из машин m_i соединена с каждым из двух своих соседей m_{i-1} и m_{i+1} двумя каналами связи: входным и выходным. Первая и последняя машины могут быть либо соединены друг с другом (кольцевая система), либо не соединены (линейная система). Будем также считать, что передача информации может одновременно осуществляться по обоим каналам. Такие УВС будем далее называть одномерными двусторонними и УВС.

Предположим далее, что УВС состоит из обычных ЭВМ, соединенных друг с другом программно изменяемой системой коммутаций (УВС с частично переменной структурой). Будем также исходить из того, что каждая ЭВМ может совмещать ввод и вывод информации с работой по основной программе. Иначе говоря, каждая ЭВМ может одновременно выполнять три программы: программу ввода информации, программу вывода информации и основную программу.

Будем считать, что в данной УВС применяется иерархический принцип управления вычислений и на каждом этапе вычисления одна из машин управляет работой и настройкой всех остальных

них. При этом каждая ЭВМ либо выполняет команды, поступающие от управляющей машины, либо, если их нет, — команды, хранящиеся в памяти самой ЭВМ.

Подобную УВС можно построить из обычных ЭВМ, если добавить к каждой из них блок коммутации (БК) и блок управления коммутацией (БУК) (рис. 9). В блок коммутации каждой машины m_i (рис. 10) входят три канала связи (от самой машины m_i и от машин m_{i-1} и m_{i+1}) и выходят три канала к тем же машинам, соответственно, z_0, z_1, z_2 и x_0, x_1, x_2 . Схема коммутации, выполненная, например, на шести конъюнкторах, позволяет соединить входной канал любой из этих трех машин с любым, но только одним выходным каналом другой машины

$$z_{\ell+k} \quad (k, \ell = 0, 1, 2)$$

Работа блока коммутации может быть описана системой булевых функций:

$$\begin{aligned} x_k &= \bigvee_{\ell=0}^{2n} C_{k\ell} \& z_{\ell}, \\ \bigvee_{\substack{k,j=0 \\ k \neq j}}^{2n} C_{ki} \& C_{kj} &= 0, \end{aligned} \quad (30)$$

где $2n$ — размерность пространственного расположения УВС. Для одномерных УВС $n=1$.

Функциями блока управления коммутацией является подача сигналов на входы $C_{k\ell}$, а также обмен информацией между ЭВМ и блоком коммутации. В соответствии с этим БК состоит из трёх узлов: узла управления коммутацией (УУК), узла ввода информации (УВв) и узла вывода информации (УВ) (рис. 11).

Узел управления коммутацией состоит из регистра R_5 , в котором хранится информация, управляющая блоком коммутации, и схемы управления коммутацией и синхронизацией (СУК), посредством которой осуществляется синхронизация ввода, вывода, пуск и останов ЭВМ, приоритет ввода над выводом и т.п.

Узел ввода информации состоит из командного регистра ввода R_3 , регистра R_4 , в который поступает информация из канала x_0 , и схемы управления вводом (СУВв). Информация, поступающая в регистр R_4 , имеет адресную часть, которая состоит из признака (q), указывающего, предназначена ли информация всем ЭВМ ($q=1$) или только одной ($q=0$), номер (n) которой указывается далее, а также из номера регистра, в который долж-

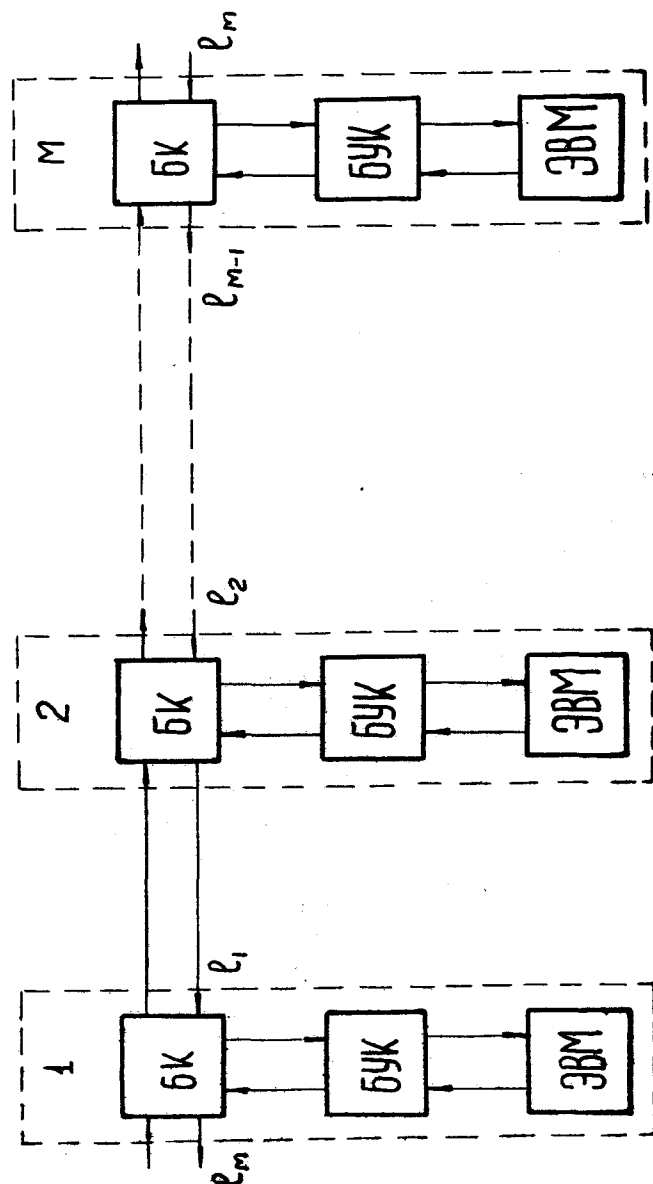


Рис. 9.

Рис. 10.

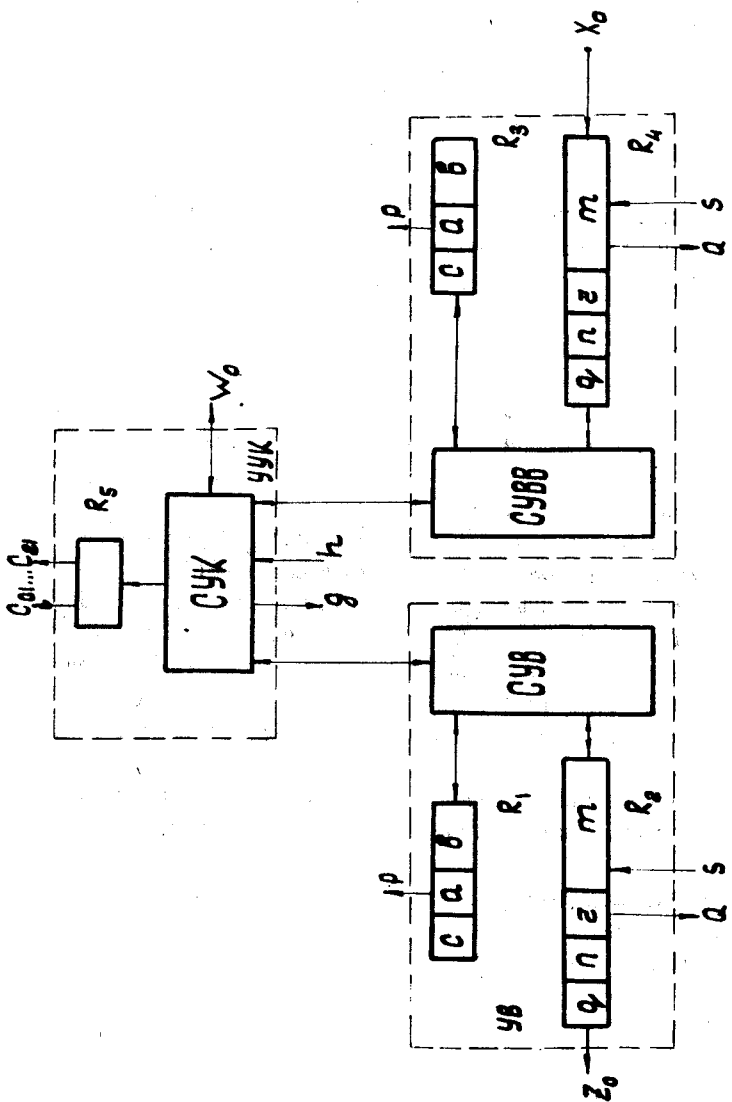
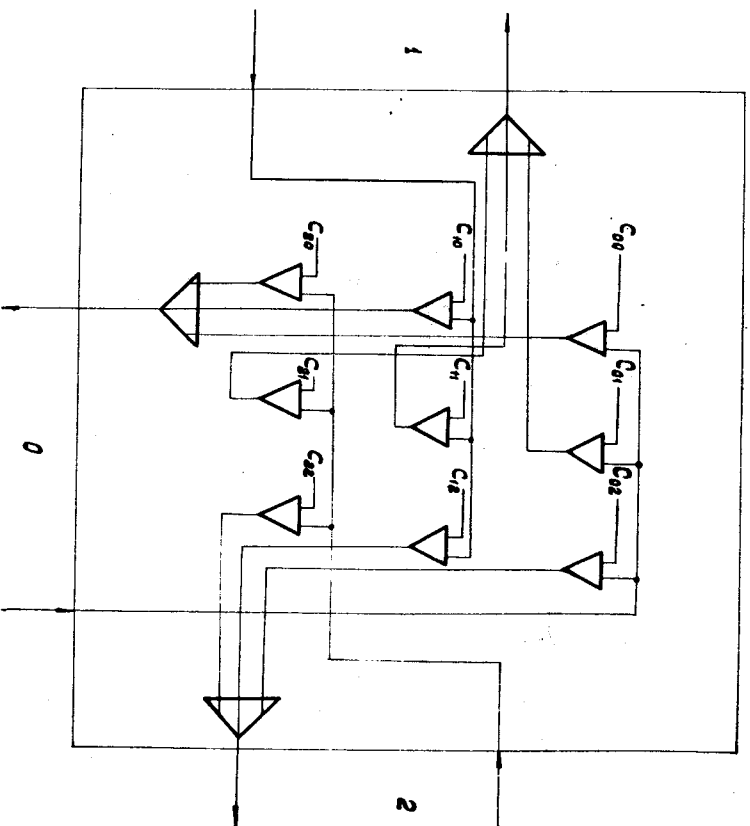


Рис. 11.

на быть передана данная информация (τ). Схема управления вводом анализирует адрес информации и в тех случаях, когда полученная информация предназначена для данной машины, открывает пути ее передачи в соответствующие регистры.

Узел вывода информации состоит из командного регистра вывода R_1 , в котором хранится очередная команда вывода, регистра R_2 , в который из оперативной памяти данной ЭВМ поступает информация, подлежащая выводу через канал Z_0 , и схемы управления выводом (СУВ), которая анализирует информацию, поступившую в R_2 , и в зависимости от ее адреса открывает пути для ее передачи либо в канал Z_0 , либо в один из регистров данной ЭВМ.

Возможны и другие варианты схемы блока управления коммутацией. В частности, возможен вариант, в котором одна и та же сравнительно простая схема осуществляет и ввод и вывод информации.

Блок-схема ЭВМ (рис. 12) является типовой и состоит из устройства управления (УУ) с регистром команд R_6 и счетчиком команд R_7 ; арифметического устройства (АУ); оперативной памяти (ОП) с регистром информации R_8 , регистром адреса R_9 и самой памятью (F); блока ввода-вывода (БВВ). Для определенности будем предполагать, что машина является обычной серийной ЭВМ, в схему которой внесены небольшие изменения, связанные с подключением блока управления коммутацией.

Многие современные ЭВМ рассчитаны на работу с каналами связи и на режимы совмещения ввода и вывода с работой основной программы, что существенно упрощает построение одномерной двусторонней УВС и фактически сводит его лишь к добавлению относительно простых блоков коммутации и подключению ЭВМ к каналам связи.

Рассмотрим теперь выполнение операций, специфичных для УВС: настройки, ввода, вывода и обобщенного условного перехода.

Состояние коммутатора каждой машины задается командами настройки, поступающими на регистр R_5 1) либо из данной ЭВМ, 2) либо из ЭВМ, управляющей ходом вычислений. В первом случае на регистре команд R_6 устанавливается команда передачи содержимого ячейки оперативной памяти f , хранящей соответствующую команду настройки, в регистр R_5 . Эта команда выполняется в тот момент, когда нет приема или передачи информации. Во втором случае машина, выполняющая функции управления, транзитом

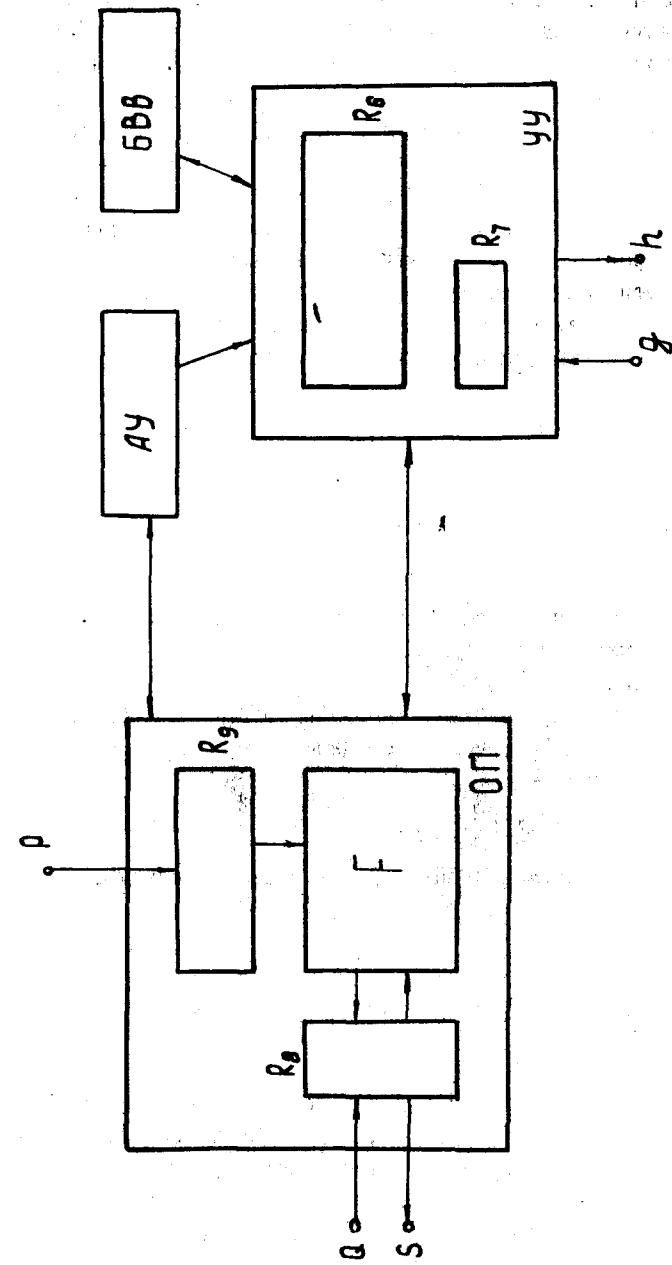


Рис. 12.

через все машины посылает команду настройки, сопровождаемую или предваряемую специальным сигналом настройки ω , который включает регистры R_4 всех ЭВМ на прием команды настройки. Этот сигнал может передаваться по особому каналу, проходящему через все ЭВМ, как показано на рис. IО и II, либо по тем же каналам, что и остальная информация в виде особого сигнала. Поступившая в R_4 информация анализируется схемами СУВВ и СУК и если удовлетворяются соответствующие условия, то информация передается в регистр R_5 . В результате изменяется набор сигналов $c_{ке}$, отпирающих соответствующие вентили блока коммутации, и устанавливается требуемая система коммутаций.

Для описания выполнения операций настройки введем функцию:

$$\psi_H = \{ \psi \vee \psi([R_4(n)], i) \} \& \psi([R_4(z)], z_s), \quad (31)$$

где

$$\psi(x, y) = \begin{cases} 1 & \text{при } x = y, \\ 0 & \text{при } x \neq y. \end{cases} \quad (32)$$

$[R_j(k)]$ — содержимое части k регистра с номером j ;

z_e — номер регистра R_e ;

i — номер данной ЭВМ.

Тогда первый случай будет иметь место при $\psi_H = 0$ и при наличии в регистре R_6 команды передачи содержимого ячейки памяти f в регистр R_5 :

$$[R_6] = \Pi_f R_5 \quad (33)$$

При этом содержимое ячейки f замещает содержимое регистра R_5 :

$$[f] = : R_5 \quad (34)$$

Второй случай имеет место, когда $\psi = 1$. При этом

$$[R_4(m)] = : R_5. \quad (35)$$

Для выполнения операции ввода информации на регистре R_3 устанавливается команда ввода (c), в которой указывается номер ячейки, начиная с которой осуществляется размещение в памяти кодов вводимой информации (a), и число вводимых кодов (b). Ввод осуществляется порциями, размер которых определяется емкостью регистра R_4 . После того, как регистр R_4 полностью заполняется поступающей через вход x_0 информацией, его содер-

жимое передается в оперативную память по адресу, содержащемуся в R_3 . На это время оперативная память прерывает свою работу по основной программе. Затем регистр R_4 начинает принимать новую порцию информации, а ЭВМ переключается на продолжение основной программы и т.д., пока не будет принята вся информация. После завершения данной операции ввода ЭВМ приступает к выполнению следующей команды ввода, если таковая имеется.

Команда ввода может быть установлена в R_3 либо программным способом из данной ЭВМ, либо путем передачи соответствующей команды из ЭВМ, управляющей вычислительным процессом.

Операция ввода информации может быть описана с помощью функции:

$$\psi_{BB} = \{ \psi \vee \psi([R_4(n)], i) \} \& \psi([R_4(z)], z_s), \quad (36)$$

в которой приняты те же обозначения, что и в (31).

Тогда при $\psi_{BB} = 0$ и

$$[R_6] = \Pi_f R_3 \quad (37)$$

выполняется пересылка информации

$$[f] = : R_3, \quad (38)$$

а при $\psi_{BB} = 1$

$$[R_4(m)] = : R_3. \quad (39)$$

Если в регистре R_3 оказывается команда ввода, то регистр R_4 принимает информацию, поступающую из канала x_0 , и порциями, равными емкости регистра R_4 , отправляет ее в ОП

$$[x_0] = : R_4, \quad (40)$$

$$[R_4] = : a + \kappa, \quad (41)$$

где $\kappa = 0, 1, \dots, b-1$.

При $\kappa = b-1$ процесс ввода оканчивается.

Операция вывода информации осуществляется аналогично операции ввода. Команда вывода устанавливается в регистре R_3 либо из самой ЭВМ, либо из ЭВМ, управляющей ходом вычислений, через регистр R_4 . В команде вывода информации указывается номер первой ячейки участка памяти (a), хранящего выводимую информацию, и общее число считываемых кодов (b). После того, как все содержимое регистра R_2 передано в канал связи z_0 , в ре-

гистр R_2 передается следующая порция информации. На этот период оперативная память выключается из основной программы. После завершения данной операции вывода ЭВМ переходит к выполнению следующей команды вывода, если таковая имеется.

Операция вывода информации может быть описана с помощью функции:

$$\psi_8 = \{q \vee \psi([R_4(n)], i) \& \psi([R_4(z)], z_1), \quad (42)$$

где обозначения те же, что и в (36).

Тогда при $\psi_8 = 0$ и

$$[R_6] = \Pi_f R_i \quad (43)$$

в регистр R_1 засылается команда, содержащаяся в ячейке оперативной памяти

$$[f] = : R_1, \quad (44)$$

а при $\psi_6 = 1$ - команда, переданная по каналам связи

$$[R_4(m)] = : R_1. \quad (45)$$

После этого начинается вывод информации в канал Z_0 порциями, равными емкости регистра R_2 :

$$[\alpha + \kappa] = : R_2, \quad (46)$$

$$[R_2] = : Z_0, \kappa = 0, 1, \dots, \beta - 1. \quad (47)$$

При $\kappa = \beta - 1$ процесс вывода заканчивается.

Работа каждой ЭВМ УВС по основной программе также может быть описана с помощью аналогичных формул:

$$\psi_0 = \{q \vee \psi([R_4(n)], i) \& \psi([R_4(z)], z_0). \quad (48)$$

Тогда при $\psi_0 = 0$ выполняется очередная команда, содержащаяся в памяти самой ЭВМ, а при $\psi_0 = 1$ - команда, поступившая из канала X_0

$$[R_4(m)] = : R_6. \quad (49)$$

Выполнение операции обобщенного условного перехода при иерархической системе управления можно представить себе следующим образом. Когда наступает время выполнения команды обобщенного условного перехода, содержащегося в программе машины, которая управляет работой системы, то из этой машины во все

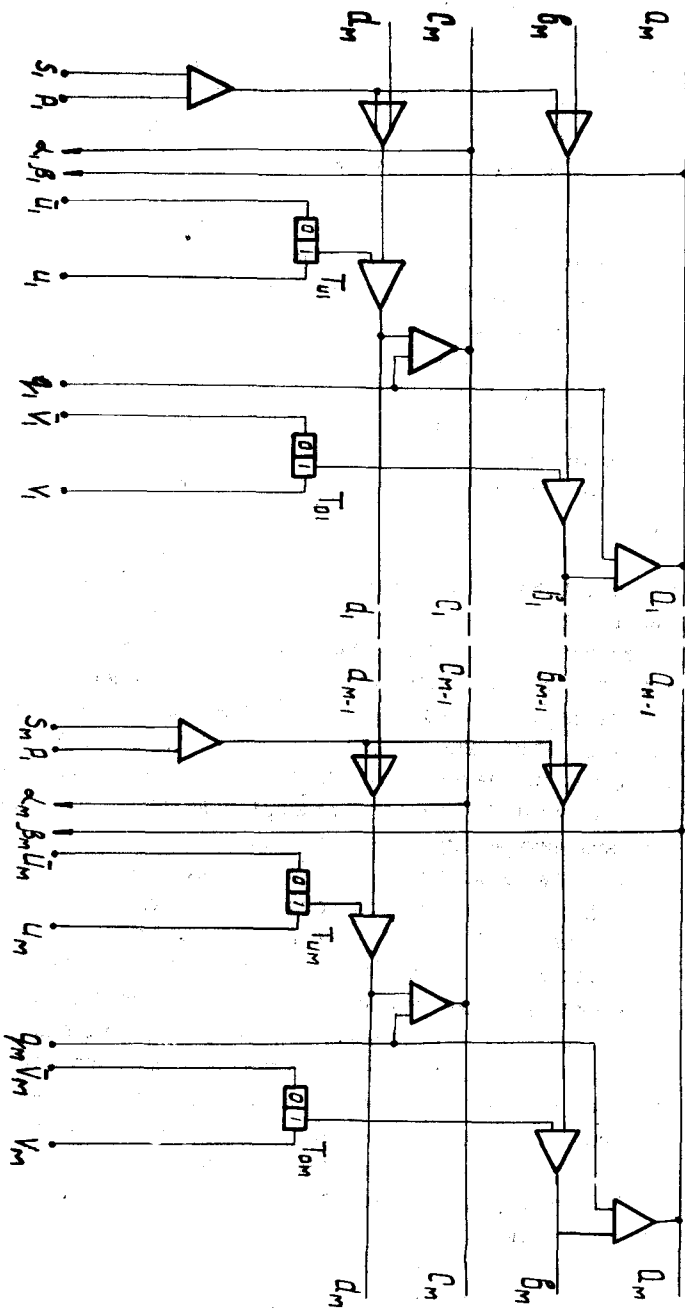
остальные поступает специальная команда. Эта команда может передаваться либо по специальному каналу, либо по обычным каналам в виде особой команды. По этой команде каждая машина посылает в управляющую информацию о состоянии, в котором она находится. Управляющая машина анализирует информацию о состоянии всех машин и в зависимости от этого изменяет ход выполнения программы.

При однородной системе управления выполнение операции обобщенного условного перехода можно представить следующим образом. Пусть требуется осуществить операцию обобщенного условного перехода при выполнении какого-либо условия каждой из ЭМ системы, которые, вообще говоря, могут быть различными для различных ЭМ. В программе каждой ЭМ m_i эти условия могут выполняться в виде обычной операции условного перехода, снабженной особым признаком, который указывает, что эта операция участвует в образовании операции обобщенного условного перехода.

Рассмотрим некоторые возможные модификации операции обобщенного условного перехода, отличающиеся условиями перехода УВС в новое состояние. Пусть этими условиями будут: 1) выполнение всеми ЭМ операции условного перехода, отмеченной признаком (все машины, которые завершили эту операцию раньше, ждут последнюю); 2) выполнение всеми машинами первого условия и некоторого дополнительного; 3) выполнение операции условного перехода с признаком какой-нибудь одной ЭМ; 4) выполнение какой-нибудь одной ЭМ третьего условия и некоторого дополнительного. Все эти четыре случая могут быть реализованы на сравнительно простой схеме (рис. 13), приданной каждой ЭМ.

Все ЭМ объединяются четырьмя общими каналами связи a, b, c, d . Схема каждой ЭМ m_i содержит два триггера: T_{ui} , который приходит в состояние $T_{ui} = 1$, когда наступает момент выполнения операции условного перехода с признаком, и T_{oi} , который приходит в состояние $T_{oi} = 1$, когда выполняется некоторое условие, содержащееся в команде условного перехода с признаком. На входы S_i и q_i подаются потенциальные сигналы, открывающие соответствующие вентили. На входы P_i подаются через определенные промежутки времени импульсы.

Первое условие реализуется подачей сигнала S_i у первой ЭМ m_1 и сигнала q_n - у последней ЭМ m_n (и только у них), тогда, после того, как все триггеры T_{ui} через входы u_i будут приведены в состояние $T_{ui} = 1$, импульс со входа P_1 поступит



в шину d , пройдет через все вентили и попадет в шину c , откуда в виде сигнала на входы α_i поступит во все ЭМ, которые его воспримут как сигнал выполнения обобщенного условного перехода.

Второе условие отличается от первого только тем, что сигналом обобщенного условного перехода будет одновременное поступление сигналов на входы α_i и β_i всех ЭМ. Поступление сигнала только на входы α_i означает, что условие, необходимое для выполнения операции обобщенного условного перехода, не осуществилось.

В третьем условии на входы S_i и q_i всех ЭМ подаются потенциальные сигналы. При поступлении сигнала на вход u_i любой из машин триггер T_{ui} переходит в состояние $T_{ui}=1$, и в шине c появляется сигнал, попадающий на входы α_i всех машин.

Четвертое условие отличается от третьего только тем, что сигналом обобщенного условного перехода служит импульс, поступающий в ЭМ через входы β_i .

Соответствующие значения S_i и q_i устанавливаются с помощью операций настройки. Заметим, что не обязательно все ЭМ выполняют операцию обобщенного условного перехода или получают его сигналы. У ЭМ, подлежащих исключению из участия в операции обобщенного условного перехода, триггеры T_{ui} и T_{oi} с помощью операций настройки блокируются в двух первых и двух последних случаях в состояниях 1 и 0, соответственно, а входы α_i и β_i , через которые поступают сигналы, управляющие переходом, могут закрываться. В первых двух случаях роль первой и последней ЭМ могут выполнять либо две соседние ЭМ, либо такие, между которыми находятся ЭМ, не участвующие в образовании обобщенного перехода.

Отметим некоторые общие свойства одномерных двусторонних систем рассматриваемого типа.

Все ЭМ системы равноправны. Любая из них или все вместе могут управлять ходом вычислительного процесса. При этом не обязательно, чтобы одна и та же машина управляла решением данной задачи от начала до конца. Процесс решения можно разделить на этапы, каждым из которых управляет своя ЭМ.

С помощью операций настройки можно разбить УВС на подсистемы, в каждой из которых находятся ЭМ с номерами, совпадающими с отрезком натурального ряда чисел в одномерных системах. В каждой такой подсистеме в любой момент времени вывести информацию в канал связи может только одна ЭМ. Все ос-

тальные машины подсистемы могут либо принимать, либо не принимать данную информацию. ЭВМ, выдающая информацию, может одновременно принимать информацию от какой-либо ЭВМ, не входящей в данную подсистему.

Рассмотрим теперь другие возможные варианты структуры распределенных вычислительных систем.

2. К о л ь ц е в а я р а с п р е д е л е н н а я У В С. Кольцевая УВС (рис. 9) наиболее проста по структуре. Она состоит из ВС, соединенных двухсторонними каналами связи. В каждую ВС входят блок коммутатора (БК) и блок управления коммутацией (БУК), которые обеспечивают варьирование коммутаций при вводе или выводе информации. При этом возможна передача данных от одной ВС к любой другой или ко всем остальным, либо одновременная передача данных всеми ВС своим ближайшим соседям. Предусматривается также возможность управления данной ВС всеми остальными ВС кольцевой ВС.

Кольцевая ВС по сравнению с одномерной обладает повышенной надежностью работы, так как при выходе из строя одной ВС или линии связи на одном из участков между двумя соседними ВС работа ВС в целом не нарушается.

3. С е т е в а я р а с п р е д е л е н н а я У В С. По аналогии с единой информационной сетью можно создать сеть из ВС, образующую сетевую распределенную УВС (рис. 14). Последняя может состоять из трех объединенных каналами связи иерархических уровней: главных вычислительных систем (ГВС), областных вычислительных систем (ОВС), местных вычислительных систем (МВС).

ГВС представляют собой высокопроизводительные вычислительные системы, соединенные друг с другом каналами связи по принципу полного графа. ОВС соединяются с ГВС, и все ОВС, относящиеся к одной и той же ГВС, образуют между собой кольцевую вычислительную систему. Наконец, МВС соединяются с ОВС и друг с другом, образуя одномерную вычислительную систему.

Чем выше уровень, тем выше пропускная способность каналов, и надежность системы обеспечивается переходом от одномерных ВС на двух нижних уровнях к соединению по принципу полного графа на верхнем уровне. Производительность и надежность сетевой распределенной УВС могут быть весьма высокими. Как и в случае с кольцевой системой, здесь возможно управление из любой ВС всеми остальными.

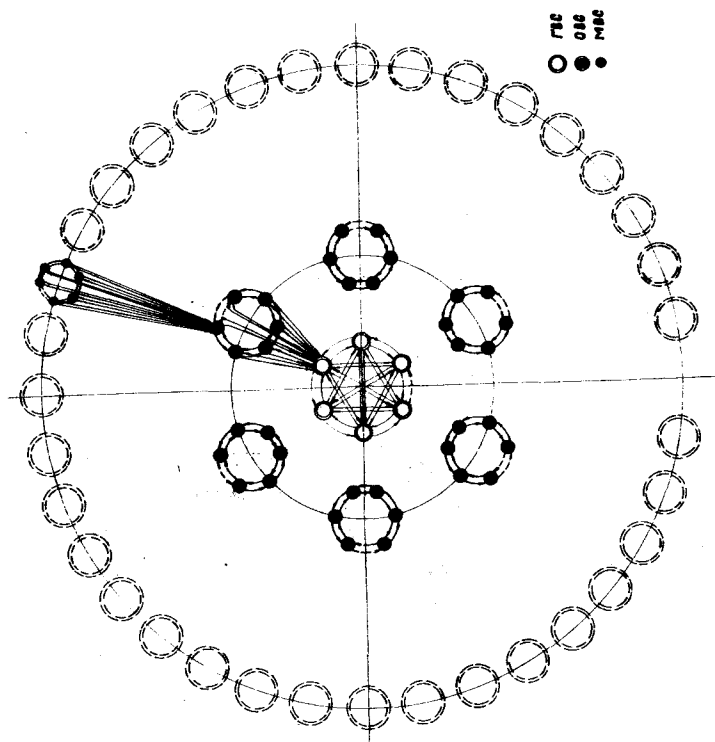
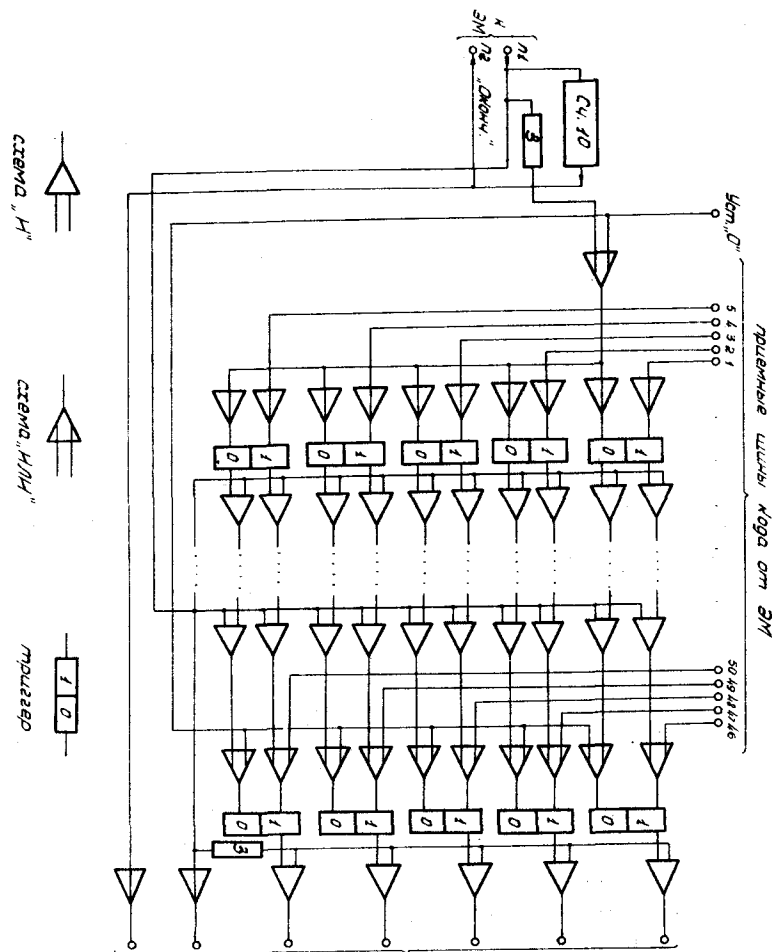


Рис. 14.

Если на регистре команд машины m_i установлена команда ОУП2 и условие w_i не выполняется, то осуществляется переход к очередной команде. Если условие w_i выполняется, то содержимое по адресу "а" заносится в некоторую фиксированную ячейку "б" и возбуждается триггер T_{wsi} , сигнализирующий о вы-





на входы Э
коммутатора Н

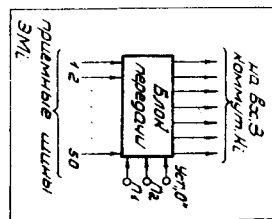


Рис. 17.

полнении условия w_i в машине m_i . Затем также производится переход к очередной команде. При возбуждении триггеров T_{wci} во всех машинах, за исключением заблокированных, выдается сигнал обобщенного условного перехода w_c . По сигналу w_c , при возбужденных триггерах T_{wci} , T_{wc} , содержимое ячейки "В" трактуется как код команды и передается в регистр команд каждой ЭМ.

Схема блока условных обобщенных операций дана на рис.18.

При настройке системы путем подачи разрешения на управляющие входы $\Gamma, K, \bar{K}, w_g, \bar{w}_g$ все ЭМ системы (или некоторой независимой её части) делятся на

- начальную (1-ю в системе),
- конечную (последнюю в системе),
- промежуточные (между первой и последней) и
- блокированные (промежуточные, не участвующие в выполнении ОУП1 и ОУП2).

В начальной машине сигнал разрешения подается на вход Γ , подключая генератор "Ген". В конечной ЭМ сигналы разрешения и запрещения на K подаются соответственно. В заблокированной ЭМ сигналы разрешения и запрещения подаются, соответственно, на входы w_g

При операции ОУП1 (в случае выполнения условия w) по сигналу "ОУП1", возникшему в какой-нибудь ЭМ, возбуждаются триггеры T_{wc} во всех машинах, за исключением заблокированных.

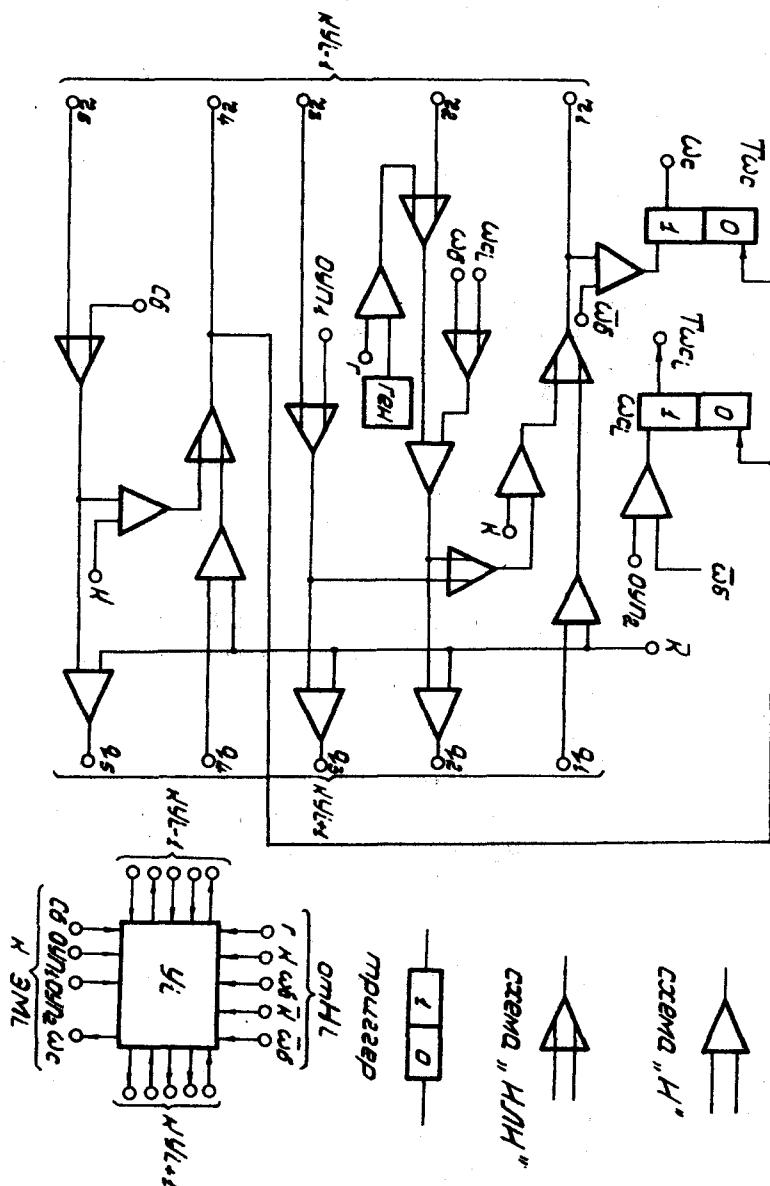
При операции ОУП2 (в случае выполнения условия w) по сигналу "ОУП2" в неблокированной ЭМ возбуждается триггер T_{wci} . После возбуждения всех T_{wci} (в системе изолированной ее части) сигналами от генератора "Ген" возбуждаются и все триггеры T_{wc} . После выполнения очередной операции триггеры T_{wc} и T_{wci} сбрасываются по сигналу сброса "сб".

6. 0 п е р а ц и я н а с т р о й к и Н

Н	α_1, α_2	$\beta_1 - \beta_6$	β_7, β_8	β_9, β_{10}
---	----------------------	---------------------	--------------------	-----------------------

При этой операции между машинами задается коммутация каналов передачи и приема рабочей информации, каналов настроечной информации и каналов обобщенного условного перехода. Имеется возможность из данной ЭМ проводить настройку любой другой ЭМ. Настройка осуществляется как пошаговая операция, при которой в каждый данный момент настраивается только одна машина, после настройки данной машины настраивается соседняя, справа или слева от нее. При настройке данной ЭМ из нее же в разряде

Рис. 18.



$\alpha_1=1$ команды настройки дается разрешение на настройку, в разрядах $\beta_7-\beta_6$ записывается информация о коммутации каналов приема и передачи рабочей информации, в разрядах $\beta_7 \beta_8$ - информация о коммутации каналов обобщенного условного перехода, в разрядах $\beta_9 \beta_{10}$ - информация о направлении передачи информации настройки для последующего шага. При настройке ЭМ, находящейся справа или слева от машины, задающей настройку последующих ЭМ, в команде настройки в разряде α_2 записывается "1", а в остальных разрядах - информация в зависимости от требуемой коммутации.

Таким образом, операция настройки машин всей системы из данной ЭМ может быть задана последовательностями команд 1) для настройки машин вправо от данной ЭМ; 2) для настройки машин влево от неё. Каждая из последовательностей начинается с команды настройки данной ЭМ. Эта команда имеет разряд $\alpha_1=1$, остальные команды имеют разряды $\alpha_2=1$.

Команда настройки H выполняется в два этапа:

- 1) передается разрешение на настройку от предыдущего шага (от предыдущей ЭМ),
- 2) передается собственно информация настройки.

Перейдем теперь к описанию схем коммутации и настройки.

Схема коммутатора K приведена на рис. 19. Коммутатор K предназначен для реализации коммутаций между входными и выходными каналами рабочей информации. Как уже указывалось выше, рабочая информация передается и принимается последовательно по 5-ти двоичным разрядам. Кроме того, используются два разряда для передачи служебных сигналов, связанных с передачей рабочей информации. Таким образом, каждый из входов и выходов коммутации является фактически семиканальным. В связи с принятым вариантом ВС коммутатор имеет 3 входа и 3 выхода. Входы предназначаются для приема информации от коммутатора слева, от коммутатора справа и от элементарной машины. Выходы предназначаются для передачи информации правому коммутатору, левому коммутатору и элементарной машине. Матрица соединений коммутатора имеет вид:

	1'	2'	3'
1	α_{11}	0	α_{13}
2	0	α_{22}	α_{23}
3	α_{31}	α_{32}	0

$\alpha_{ij} \in \{0,1\}$

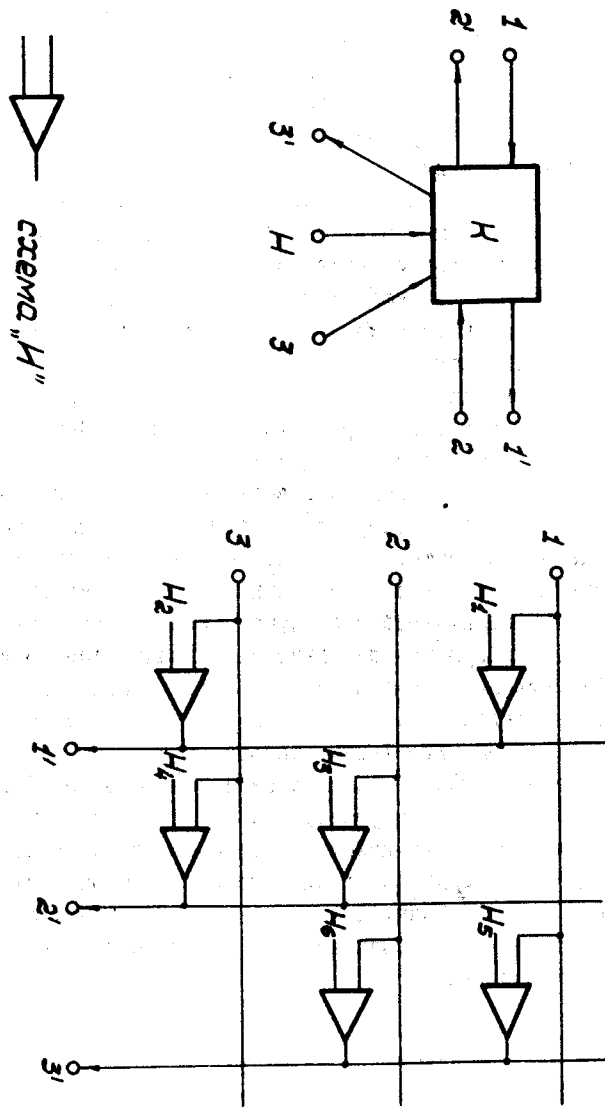


Рис. 19.

Значения α_{ij} в матрице соединений могут быть заданы с помощью посылки соответствующих сигналов разрешения на управляющие входы $H_1 - H_6$.

Схема блока настройки приведена на рис. 20.

На первом этапе по команде настройки дается разрешение на настройку. При настройке из самой элементарной машины сигнал разрешения на настройку при $\alpha_1 = 1$ поступает на вход ℓ_1 , при настройке от машины слева - на вход S_1 , от машины справа - на вход t_3 . Сигналом разрешения настройки возбуждается триггер T_1 .

На втором этапе информация настройки поступает на входы ℓ_3, ℓ_4 или S_5, S_6 , или t_7, t_8 и передается в накопительный регистр P_H .

Разряды регистра P_H с первого по шестой используются для задания разрешения на управляющие входы $H_1 - H_6$ коммутатора K . Разряды P_{H7}, P_{H8} используются для задания разрешений на управляющие входы $\omega_8, \bar{\omega}_8, \Gamma, \bar{\Gamma}, \bar{K}$ блока Y . Разряды P_{H9}, P_{H10} служат для управления направлением передачи настроечной информации вправо и влево, соответственно.

Таким образом, после настройки данной ЭМ будут заданы коммутации в коммутаторе K и в блоке Y , а также определено направление передачи настроечной информации в последующую машину.

Пусть для определенности установлено направление передачи информации настройки вправо (возбужден триггер разряда P_{H9}). Тогда по очередной команде настройки будет настраиваться машина, расположенная справа от данной. По сигналу, поступающему на вход ℓ_2 (при $\alpha_2 = 1$ в команде настройки) или на S_2 , триггер T_1 данной ЭМ переходит в состояние "0", а триггер T_1 машины, расположенной справа от данной, - в состояние "1". Информация настройки, поступающая на входы ℓ_3, ℓ_4 , передается на выходы t_5, t_6 для настройки соседней справа машины, где поступает на входы S_5, S_6 . Аналогично производится настройка машин, расположенных слева от данной.

П. Двумерные УВС с коммутациями по типу P_n -графа. Пусть все ЭМ расположены в узлах однородной квадратной сетки. Положение каждой ЭМ будем характеризовать двумя координатами i, j . Для каждой ЭМ m_{ij} будем называть соседними ЭМ $m_{i,j-1}, m_{i-1,j}, m_{i,j+1}, m_{i+1,j}$. Условимся далее при рассмотрении данной ЭМ приписывать ей ин-

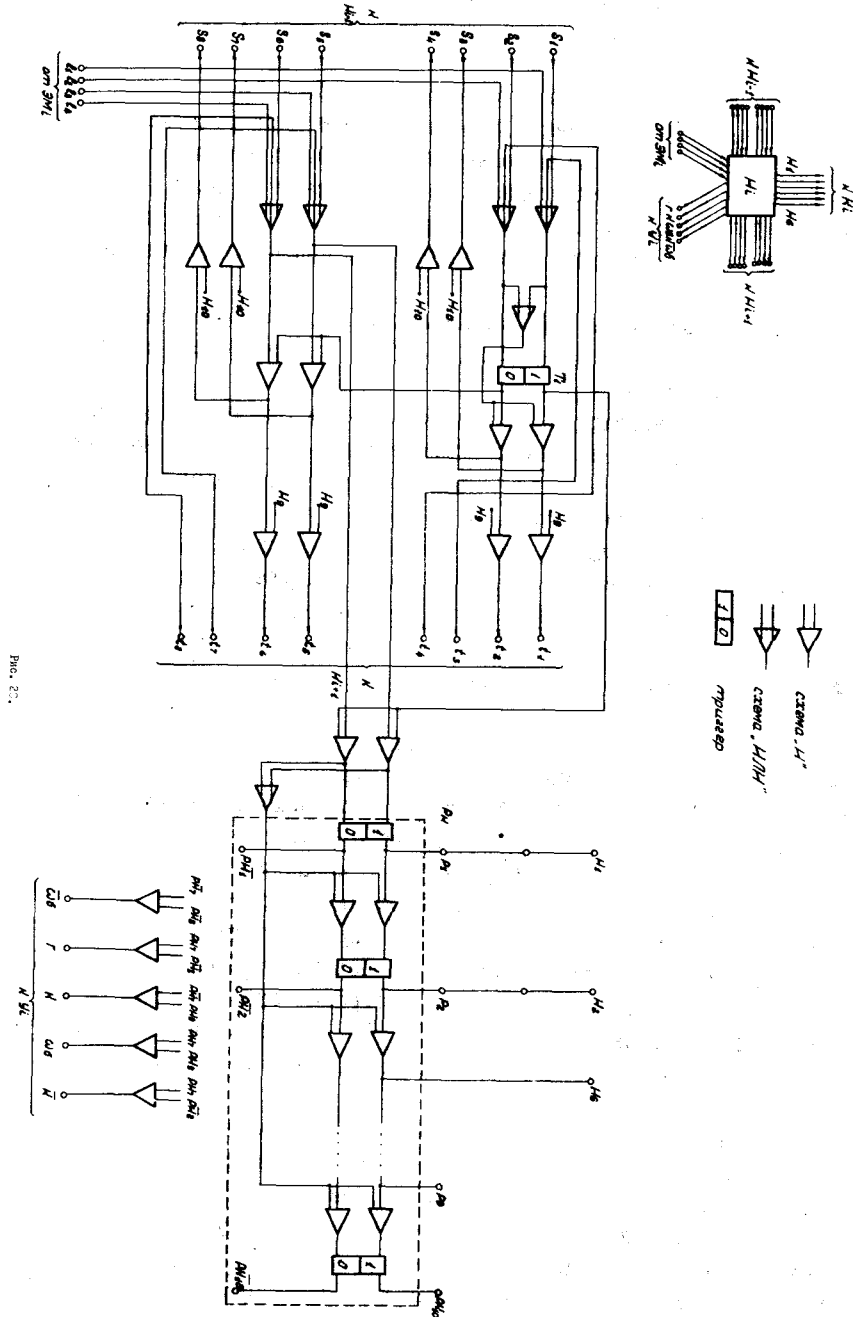


Рис. 20.

декс 0, а ее соседям—1,2,3,4, соответственно. Как и ранее, будем считать, что каждая ЭМ имеет блок коммутации, к которому от каждой соседней ЭМ подходят двусторонние каналы связи. Отличие этого блока от соответствующего блока одномерных УВС сводится к тому, что коммутатор осуществляет любое соединение между пятью парами входов и выходов, а не тремя. Работа блока коммутации может быть описана аналогично, как и в случае одномерных ВС.

При двумерном расположении ЭМ блок настройки будет отличаться от показанного на рис. 19 тем, что с его помощью можно производить настройку не в двух направлениях, а в четырех (см. § 3). Выполнение операции ввода и вывода, настройки, обобщенного условного перехода может быть таким же, как и у одномерных УВС.

У двумерных УВС появляются некоторые свойства, которых нет у одномерных, например, возможность передавать информацию от одной ЭМ m_i к любой другой m_j многими путями, определяемыми выбором значений соответствующих элементов матрицы соединений $[C_{ke}]$. Это свойство значительно повышает гибкость использования УВС и, что особенно важно, её надежность (благодаря возможности выбирать обходные пути для передачи информации в случае выхода из строя отдельных ЭМ или каналов связи (см. § 3)).

Аналогичным образом могут быть построены и трехмерные УВС. В этом случае каждая ЭМ будет соединена с шестью соседними, что вызывает соответствующие изменения в блоках коммутации и управления коммутацией, а также каналах связи.

Выводы

1. На основе использования принципов однородности и перемешивания структуры дано определение универсальной вычислительной системы с частично переменной структурой.
2. Исследованы основные свойства универсальных вычислительных систем с частично переменной структурой: универсальность, производительность, надежность, стоимость.
3. Дана классификация универсальных вычислительных систем с частично переменной структурой.
4. Рассмотрены примеры реализации различных типов универсальных вычислительных систем с частично переменной структурой.

Поступила в редакцию
18.IX.1964 г.

ЛИТЕРАТУРА

1. Евреинов Э.В., Косарев Ю.Г. О вычислительных системах высокой производительности. Техническая кибернетика, 1963, № 4, 3-25.
2. Глушков В.М. Синтез цифровых автоматов. М., Физматгиз, 1962.
3. Евреинов Э.В. Теоретические основы построения вычислительных сред. В сб.: "Вычислительные системы", 1965, вып. 16, 3-72 (Сиб. отд. АН СССР, Ин-т математики).
4. Unger S. Pattern recognition and detection Proc. IRE. Oct. 1959, 47, p. 1737-1752.
5. Holland J.H. Iterative circuits computers Proc. EJCC, 1960, p. 259-265.
6. Lee C.J. A content addressable distributed logic memory with applications to information retrieval Proc. IRE, June 1963, p.924-932.
7. Slotnick D.L. and oth. The Solomon Computer. Proc. EJCC. 1962, p. 97-107.
8. Косарев Ю.Г. О методике решения задач на универсальных вычислительных системах. Данный сборник, стр.61-99.
9. Евреинов Э.В., Косарев Ю.Г. О решении задач на универсальных вычислительных системах. Данный сборник, стр.106-164.
10. Сифоров В.И. О методах расчета надежности работы систем, содержащих большое число элементов. Изв. АН СССР, ОТН, 1954, № 6.
11. Сеница М.А. К вопросам резервирования радиоэлектронных устройств. В сб.: "Надежность радиоэлектронной аппаратуры". Советское радио. М, 1958, стр. 40-74.
12. Гнеденко Б.В. Курс теории вероятностей. Изд. Ф.М., М., 1961.
13. Решетняк Ю.Г. О задаче соединения элементов вычислительной системы. В сб.: "Вычислительные системы", Новосибирск, 1962, вып. 3, 17-30 (Сиб. отд. АН СССР Ин-т математики).
14. Евреинов Э.В. О возможности построения вычислительных систем в условиях запаздывания сигналов. В сб.: "Выч. системы", Новосибирск, 1962, вып. 3, 3-16 (Сиб. отд. АН СССР, Ин-т математики).