

ОБ ОДНОМ ПОДХОДЕ К СИНТЕЗУ ДИСКРЕТНЫХ АВТОМАТОВ

А. П. Стахов, Н. В. Алипов

Харьков

В технике ЦВМ известны примеры построения дискретных автоматов (ДА) с использованием запоминающих устройств. Суть такого метода построения ДА сводится к следующему.

Вершины a_0 графа M , задающего ДА, ставится в соответствие ячей-

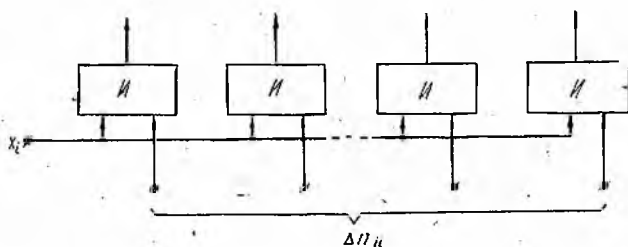


Рис. 1.

ка памяти n_i . Двоичный код состояния a_i ДА является адресом ячейки n_i , где $a_i \in A$, $A = \{a_0, a_1, \dots, a_N\}$, $n_i \in Z$, $Z = \{n_0, n_1, \dots, n_N\}$, N — количество состояний ДА. Количество разрядов ячейки памяти n_i равно количеству микроопераций, заданных на некотором операционном автомате [1]. Разряду j ячейки n_i соответствует микрооперация O_j ($j = 1, 2, \dots, r$, r — количество микроопераций). Если микрооперация O_j соответствует вершине графа, то в j -м разряде ячейки n_i стоит единица. Примером такого построения ДА может служить устройство управления ЦВМ «Мир».

Запоминающее устройство в подобных структурах играет роль комбинационной схемы выделения выходных сигналов ДА. Однако существуют такие структуры, у которых запоминающее устройство (ЗУ) используется для организации переходов дискретного автомата. Указанные структуры ДА строятся следующим образом. В ячейке памяти, кроме информации о наборе микроопераций, размещаются коды относительных адресов ячеек $n_{j1}, n_{j2}, \dots, n_{j\gamma}$, соответствующих состояниям $a_{j1}, a_{j2}, \dots, a_{j\gamma}$, в которые может переходить автомат из состояния a_j (γ — количество стрелок, выходящих из вершины a_j графа M).

Основная трудность реализации ДА по описанной схеме заключается в выделении кодов относительных адресов из ячейки n_j . Действительно, пусть в ячейке n_j записано γ кодов относительных адресов $\Delta n_{j1}, \Delta n_{j2}, \dots, \Delta n_{j\gamma}$, причем код относительного адреса соответствует одному символу входного алфавита ДА. Тогда для выделения кода относительного адреса Δn_{ji} ($i = 1, 2, \dots, \gamma$) из ячейки n_i необходимо γ комбинационных схем. Каждая подобная схема представляет собой набор схем И (рис. 1).

Количество комбинационных схем указанного вида для N ячеек памяти значительно больше. Оно определяется суммарным количеством стрелок графа, задающего ДА, и числом символов входного алфавита. Очевидно, самая простая схема ДА получается, если в каждой ячейке памяти записан только один код относительного адреса.

Выясним, нельзя ли задать ДА так, чтобы удовлетворить этому очевидному факту. В работе [2] описан способ представления деревообразных автоматов системой автоматов $A_1(m)$, $A_k(1)$, Σ . В настоящей работе приводится способ представления любого ДА системой автоматов $A_1(m)$, $A_k(1)$, Σ , ЗУ.

Введем ряд определений.

1. Вершину графа, задающего некоторый автомат M , будем считать особой, если в нее входит две или более стрелок.

2. Автомат M назовем автоматом с циклом, если среди допустимого множества входных слов [2] встречаются слова вида $x_{j1}, x_{j2}, \dots, x_{j1}$.

Можно показать истинность утверждения.

Утверждение 1. Для любого дискретного автомата без цикла существует эквивалентный деревообразный автомат.

Действительно, будем считать, что для конечных состояний ДА функции переходов не определены, а каждой последовательности состояний длины l

$$a_{r_0}, a_{r_1}, \dots, a_{r_l}, \quad (1)$$

где a_{r_0} — особая вершина, поставлены в соответствие γ последовательностей длины l вида

$$a_{ir_0}, \dots, a_{ir_l}. \quad (2)$$

Таким образом, i -я последовательность (2) соответствует i -й стрелке, входящей в особую вершину a_{r_0} (предполагается, что стрелки и последовательности (2) упорядочены), а для всех i, j, r выполняется равенство

$$\lambda(a_{rj}) = \lambda(a_{irj}),$$

где $i = 1, 2, \dots, \gamma$;

$$j = 0, 1, 2, \dots, l;$$

γ — количество стрелок, входящих в особую вершину a_{r_0} .

Как легко видеть, с учетом этих предположений можно любому ДА без цикла поставить в соответствие некоторый эквивалентный ему деревообразный автомат.

Пример. Пусть ДА без цикла задан графом рис. 2, а. Эквивалентный ему деревообразный автомат изображен на рис. 2, б.

Следствием 1 может быть следующее утверждение.

Утверждение II. Любой ДА без цикла можно однозначно задать системой автоматов $A_1(m)$, $A_k(1)$, Σ .

Однако такое эквивалентное преобразование ДА без цикла неприменимо в ряде случаев из-за того, что растет общее число состояний автомата, а это приводит к увеличению ячеек памяти в случае, когда для выделения выходных сигналов используется ЗУ. Для сокращения объема ЗУ предположим, что некоторый ДА, не имеющий цикла, задан системой автоматов $A_1(m)$, $A_k(1)$, Σ , ЗУ без учета I, II. Тогда согласно работе [2] для однозначного задания особых вершин необходимо выполнение таких условий:

а) равноценность путей, входящих в особую вершину a_{r_0} ;

2—2. Считать некоторый символ x_{k+1} входным символом ДА, который принимает значения из некоторого множества

$$Z_1 = \{0, 1, \dots, T\}.$$

2—3. Выбрать максимальное значение весовой функции ($Q_{s_{\max}}$) состояния a_{r_0} , которое соответствует одному какому-то пути, входящему в особую вершину a_{r_0} .

2—4. Каждой стрелке, входящей в особую вершину a_{r_0} , поставить в соответствие символ x_{k+1} значение которого равно

$$Q_{s_{\max}} - C'_s, \quad (2^*)$$

где Q'_s — весовая функция i -го входного слова из множества R_{r_0} .

2—5. Идентификатор символа x_{k+1} и значение символа x_{k+1} поместить в ячейку n_{r_i} , где n_{r_i} — ячейка памяти, соответствующая последнему состоянию последовательности состояний i -го пути.

Легко убедиться, что, выполнив указанные преобразования, можно любой ДА без цикла задать системой автоматов $A_1(m)$, $A_k(1)$, Σ , $\Sigma\cup$. При этом коды относительных адресов будут размещаться в тех ячейках, которые используются для организации переходов ДА в «особые» состояния. В каждой такой ячейке будет размещаться только один код относительного адреса.

Другими словами, сконструированная система автоматов $A_1(m)$, $A_k(1)$, Σ , $\Sigma\cup$ приводит к самой простой схеме ДА.

При задании ДА с циклом системой автоматов $A_1(m)$, $A_k(1)$, Σ , $\Sigma\cup$ для удовлетворения условиям а), б) необходимо, кроме преобразований 1—1, 1—2, 2—1, 2—2, 2—3, 2—4, 2—5, выполнить следующее.

3—1. Символ x_{k+2} считать входным символом ДА, который принимает значения из множества $Z_1 = \{0, 1, \dots, T_1\}$.

3—2. Если, кроме входного слова вида

$$x_{r_0}, x_{r_1}, \dots, x_{r_p}, x_{r_0}, \quad (3)$$

возможно входное слово

$$x_{r_0}, x_{r_1}, \dots, x_{r_p}, x_{r_{p+1}}, \quad (4)$$

удлинить последовательность состояний

$$a_{r_0}, a_{r_1}, \dots, a_{r_p}, a_{r_0} \quad (5)$$

на одно состояние a^* :

$$a_{r_0}, a_{r_1}, a_{r_2}, \dots, a_{r_p}, a^*, a_{r_0}. \quad (6)$$

3—3. Символу входного алфавита x_{k+2} придать значение

$$Q_s^{r_0} - Q_s^{r_p}, \quad (7)$$

где $Q_s^{r_0}$, $Q_s^{r_p}$ — весовые функции, поставленные в соответствие вершинам a_{r_0} , a_{r_p} .

3—4. Идентификатор символа x_{k+2} и его значение поместить в ячейку n_{r_p} , если в последовательность (5) не введено состояние a^* , или в ячейку n_{r^*} , если оно введено. В этом случае n_{r_p} , n_{r^*} — ячейки, адреса которых соответствуют вершинам a_{r_p} , a^* .

3—5. Автомат $A_1(m)$ представить системой автоматов Σ^1 , $A_k(1)$, считая некоторые символы x_{k+3} , x_{k+4} входными символами автомата $A_1(m)$, символу x_{k+3} приписать вес, равный единице, а символу x_{k+4} — вес

$$b_{k+4} = m_{r_0} - m_{r_p},$$

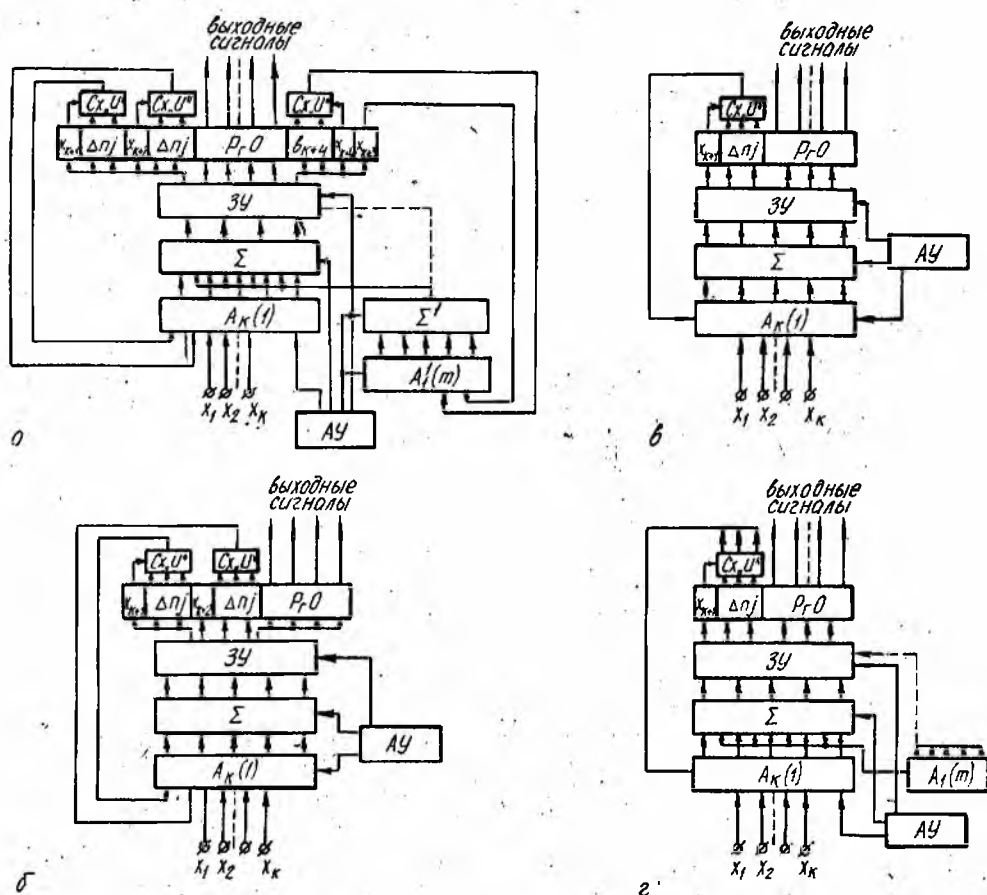


Рис. 3.

если в последовательность состояний (5) не введено состояние a^* . В противном случае

$$b_{k+4} = m_{r_0} - m_{r^*},$$

где $b_{k+4} \in Z_2$, $Z_2 = \{0, 1, 2, \dots, T_2\}$;

m_{r_0} , m_{r_p} , m_{r^*} — состояния автомата $A_1(m)$, соответствующие вершинам a_{r_0} , a_{r_p} , a^* .

Выполнив указанные преобразования, можно задать любой ДА системой автоматов $A_1(m)$, $A_k(1)$, Σ , $ЗУ$. Структуры ДА с циклом приведены на рис. 3, а, б, структуры ДА без цикла приведены на рис. 3, в, г. Здесь PzO — регистр микроопераций, $CX \ll H \gg$ — комбинационная схема.

Введение в систему автоматов автомата $ЗУ$ требует изменения алгоритма работы автомата управления,

Приведем алгоритм работы АУ для системы автоматов $A_1(m)$, $A_k(m)$, Σ и ЗУ.

1. Если П, то 2, иначе 1.
2. Установить автоматы $A_1(m)$ $A_k(1)$ и Σ в начальное состояние, перейти на 3.
3. Распознать входной символ. Вычислить значение весовой функции входного слова согласно (3) [2], перейти на 4.
4. Организовать переход автомата $A_1(m)$ из состояния s в состояние $s + 1$, перейти на 5.
5. Выбрать необходимое слово из памяти. Если на вход системы автоматов поступил конечный символ входного слова, то 1, иначе 3.

Система автоматов $A_1(m)$, $A_k(1)$, Σ и ЗУ обладает некоторой универсальностью: изменяя содержимое ячеек памяти, можно задать любую последовательность микроопераций и, следовательно, реализовать некоторое множество алгоритмов. Это особенно важно при реализации ДА на интегральных подсистемах [3].

ЛИТЕРАТУРА

1. В. М. Глушков. Теория автоматов и формальные преобразования микропрограмм. «Кибернетика», 1965, № 5.
2. А. П. Стахов, Н. В. Алипов. Метод структурного синтеза деревообразных автоматов. Сб. «Приборы и системы автоматики», вып. 12. Изд-во Харьковск. ун-та, 1969.
3. В. Долкерт и др. Микроминиатюрные аэрокосмические цифровые вычислительные машины. Изд-во «Советское радио», 1967.