8.

	q_0	q_1	q_2	q_3	q_4	q_5
0	$0/q_1$	$1/q_4$	$1/q_1$	$1/q_4$	$0/q_1$	$1/q_4$
1	$0/q_2$	$0/q_5$	$0/q_2$	$1/q_5$	$0/q_2$	$1/q_5$
2	$1/q_3$	$1/q_0$	$0/q_3$	$1/q_0$	$0/q_3$	$0/q_0$

2.6 Получение минимальных ДНФ при помощи карт Карно

Итак, КА был представлен системой булевых функций, причем некоторые из них могут быть не всюду определенными, т. е. содержать в своей записи не только 0 и 1 но и $*$. Следующий шаг на пути построения реально работающего устройства — представление каждой булевой функции из этой системы в виде терма над некоторым базисным множеством B. По традиции, в качестве основного базиса будем рассматривать элементарный базис $B_0 = \{\&, \vee, -\}$, однако отметим, что в последние годы весьма эффективно используются термальные представления булевых функций над полным бинарным базисом $B_1 = \{\&, \vee, \oplus, -\}$. Следует заметить, что представление булевой функции в виде терма над заданным базисом B является одной из самых интересных и в то же время самых трудных задач. Само по себе нахождение произвольного представления функции f, например над базисом B_0 не представляет особого интереса. Имеется множество канонических форм и алгоритмы, гарантирующие их посторoение для любой булевой функции. Как правило, при реализации булевой функции f в виде терма ставится дополнительное условие: полученный терм должен быть достаточно простым. Действительно, мы уже сталкивались при построении СФЭ с тем, что одна и та же булева функция может быть представлена различными термами над базисом B_0, причем канонические формы, как правило, дают не самую простую ее реализацию.

Легко понять, что простой терм позволяет построить простую СФЭ, что ведет, в частности, к экономии материалов, объема, веса,

энергопотребления схемы и времени обработки ею сигнала. Ясно, что среди всех представлений некоторой булевой функции f в B_0 имеется и самое простое. Это представление называется минимальным для f. Известно, что нахождение минимального представления для большинства булевых функций является весьма трудоемкой задачей, связанной с полным перебором, неприемлемым с практической точки зрения. Обычно ставится менее сложная задача: для заданной функции f найти не самый простой терм вообще, а наиболее простой терм специального вида.

Далее будем рассматривать представления булевых функций специальными термами вида $\bigvee\limits_{i=1}^{m} K_i$, где K_i есть элементарная конъюнкция $x_{i_1}^{\sigma_1} \cdot x_{i_2}^{\sigma_2} \cdots x_{i_s}^{\sigma_s}$. Такой терм представляет из себя дизъюнкцию элементарных конъюнкций и называются *дизъюнктивными нормальными формами* (ДНФ). Ясно, что СДНФ является частным случаем ДНФ.

Под сложностью ДНФ $\Phi = \bigvee\limits_{i=1}^{m} K_i$ будем понимать число входящих в ее состав элементарных конъюнкций K_i. Для сложности ДНФ введем следующее обозначение: $L_{\text{ДНФ}}(\Phi) = m$.

Пример. Для ДНФ $\Phi = \overline{x}_1 x_2 \vee x_1 \overline{x}_2 x_3$ сложность $L_{\text{ДНФ}}(\Phi) = 2$.

В связи с тем что одна булева функция f может иметь много представлений в виде ДНФ, задача ее минимизации, но уже в классе ДНФ, остается актуальной. Под минимальной для функции f будем понимать такую ДНФ, которая представляет f и имеет наименьшую возможную сложность.

Для булевых функций небольшой размерности имеется простой и весьма наглядный способ получения минимальных ДНФ, основанный на использовании карт Карно. Его суть заключается в следующем: булева функция помещается в матрицу специального вида, называемую картой Карно. Приведем пустые матрицы для функций двух, трех и четырех переменных (рис. 31).

Способ заполнения матрицы значениями функции f очевиден: на пересечении строки $\tilde{\alpha}$ и столбца $\tilde{\beta}$ помещается значение $f(\tilde{\alpha}, \tilde{\beta})$.

$x_1 \backslash x_2$	0	1
0		
1		

x_2	0	0	1	1
$x_1 \backslash x_3$	0	1	1	0
0				
1				

x_3	0	0	1	1
$x_1\ x_2 \backslash x_4$	0	1	1	0
0 0				
0 1				
1 1				
1 0				

Рис. 31

Следует обратить особое внимание на тот факт, что наборы в строках и столбцах матрицы упорядочены не натурально, а рефлексивно, что приводит к замене местами некоторых значений булевой функции, заданной, как правило, таблицей с натуральным упорядочением.

Дальнейшее решение состоит в нахождении минимального покрытия множества всех единиц в матрице фигурами следующего вида (рис. 32):

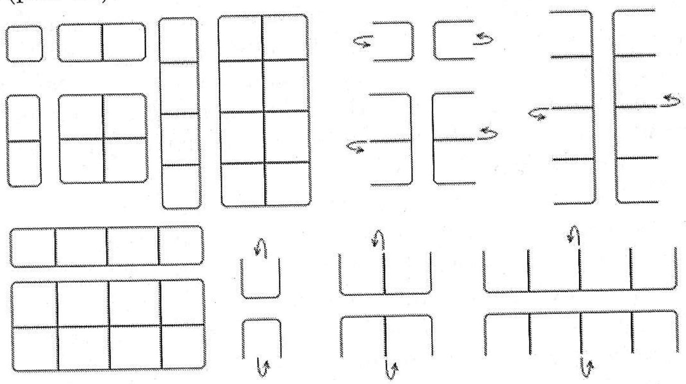
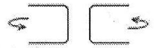
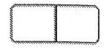
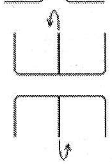
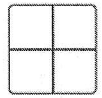


Рис. 32

Шесть фигур со стрелками нужно понимать следующим образом: матрица считается "склеенной"как по вертикальным, так и по

горизонтальным границам, т. е. фигура может начинаться, например, у ее правого края, а продолжиться на левом. Таким образом, фигура  есть частный случай фигуры ,

а фигура  — частный случай фигуры , и т. д.

При покрытии множества единиц функции фигурами следует учитывать, что каждая фигура добавляет к ДНФ одну элементарную конъюнкцию. Что касается размера, то чем фигура больше, тем меньшей конъюнкции она соответстует. Таким образом, для нахождения минимальной ДНФ необходимо искать покрытие наименьшим возможным числом как можно больших фигур. Заметим, что фигуры могут пересекаться.

Заключительным этапом является получение собственно ДНФ, соответствующей найденному минимальному покрытию. Как уже было сказано, каждой фигуре соответствует одна элементарная конъюнкция. Ее можно получить, выписывая все наборы, которые занимает данная фигура, и далее:

а) если переменная x_i на всех этих наборах равна 1, то в элементарную конъюнкцию она входит без отрицания;

б) если переменная x_i на всех этих наборах равна 0, то в элементарную конъюнкцию она входит с отрицанием;

в) если переменная x_i принимает на этих наборах различные значения, то она отсутствует в данной элементарной конъюнкции.

Особо следует рассмотреть неполностью определенные булевы функции, т. е. функции, содержащие $*$. Для них справедливы все вышеприведенные рассуждения, причем следует помнить, что $*$ может как входить в какую-либо фигуру, так и не входить ни в одну из них. В первом случае $*$ интерпретируется как 1, во втором – как 0. Наличие символов $*$ упрощает задачу построения минимального покрытия, так как позволяет варьировать расположение фигур.

Рассмотрим некоторые примеры:

1) *Построить минимальную ДНФ для функции* $f_1 = (1100\ 1010)$.

Поместим функцию f_1 в матрицу и найдем минимальное покрытие (рис. 33). Оно состоит из двух фигур. Найдем соответствующие элементарные конъюнкции:

x_1 \\ $x_2 x_3$	00	01	11	10
0	1	1	0	0
1	1	0	0	1

Рис. 33

x_1	x_2	x_3	
0	0	0	$\to \overline{x}_1\overline{x}_2$
0	0	1	

x_1	x_2	x_3	
1	0	0	$\to x_1\overline{x}_3$
1	1	0	

Таким образом, $f_1 = \overline{x}_1\overline{x}_2 \vee x_1\overline{x}_3$, и $L_{\text{ДНФ}=2}$.

2) *Построить минимальную ДНФ для функции* $f_2 = (0101\ 0011\ 0101\ 0110)$.

Поместим функцию f_2 в матрицу и найдем одно из минимальных покрытий (рис. 34). Оно состоит из четырех фигур. Найдем соответствующие элементарные конъюнкции:

$x_1 x_2$ \\ $x_3 x_4$	00	01	11	10
0 0	0	1	1	0
0 1	0	0	1	1
1 1	0	1	0	1
1 0	0	1	1	0

Рис. 34

x_1	x_2	x_3	x_4	
1	1	0	1	$\to x_1\overline{x}_3x_4$
1	0	0	1	

x_1	x_2	x_3	x_4	
0	0	0	1	$\to \overline{x}_2x_4$
0	0	1	1	
1	0	0	1	
1	0	1	1	

x_1	x_2	x_3	x_4	
0	1	1	1	$\to \overline{x}_1x_2x_3$
0	1	1	0	

x_1	x_2	x_3	x_4	
0	1	1	0	$\to x_2x_3\overline{x}_4$
1	1	1	0	

$f_2 = x_1\overline{x}_3x_4 \vee \overline{x}_2x_4 \vee \overline{x}_1x_2x_3 \vee x_2x_3\overline{x}_4$, и $L_{\text{ДНФ}=4}$.

3) *Построить минимальную ДНФ для функции* $f_3 = (0101\ *011\ 0101\ 011*)$.

Отметим, что f_2 является частным случаем f_3, при доопределении двух символов $*$. Таким образом, f_3 должна иметь сложность, не превышающую сложность ДНФ для f_2. Действительно, помещая f_3 в матрицу, находим лучшее чем у f_2 представление (рис. 35):

$x_1 x_2$ \\ $x_3 x_4$	00	01	11	10
0 0	0	1	1	0
0 1	*	0	1	1
1 1	0	1	*	1
1 0	0	1	1	0

Рис. 35

x_1	x_2	x_3	x_4	
0	0	0	1	$\to \overline{x}_2x_4$
0	0	1	1	
1	0	0	1	
1	0	1	1	

x_1	x_2	x_3	x_4	
1	1	0	1	$\to x_1x_4$
1	1	1	1	
1	0	0	1	
1	0	1	1	

x_1	x_2	x_3	x_4	
0	1	1	1	$\to x_2x_3$
0	1	1	0	
1	1	1	1	
1	1	1	0	

Таким образом, частичную функцию f_3 можно доопределить термом $\overline{x}_2x_4 \vee x_1x_4 \vee x_2x_3$, и $L_{\text{ДНФ}} = 3$.

4) *Построить минимальную ДНФ для функции* $Q_2(t)$ *из примера на странице 68.* $Q_2(t) = (0110\ 1010\ 0110\ ****)$.

Поместим $Q_2(t)$ в матрицу и найдем одно из минимальных покрытий (рис. 36):

$x_1 x_2$ \\ $Q_1 Q_2$	00	01	11	10
0 0	0	1	0	1
0 1	1	0	0	1
1 1	*	*	*	*
1 0	0	1	0	1

Рис. 36

x_1	x_2	Q_1	Q_2	
0	0	1	0	$\to Q_1\overline{Q}_2$
0	1	1	0	
1	1	1	0	
1	0	1	0	

$$\begin{array}{cccc|l} x_1 & x_2 & Q_1 & Q_2 & \\ 0 & 1 & 0 & 0 & \multirow{2}{*}{$\to \overline{x}_1 x_2 \overline{Q}_2$} \\ 0 & 1 & 1 & 0 & \end{array} \qquad \begin{array}{cccc|l} x_1 & x_2 & Q_1 & Q_2 & \\ 0 & 0 & 0 & 1 & \multirow{2}{*}{$\to \overline{x}_2 \overline{Q}_1 Q_2$} \\ 1 & 0 & 0 & 1 & \end{array}$$

$$Q_2(t) = Q_1\overline{Q}_2 \vee \overline{x}_1 x_2 \overline{Q}_2 \vee \overline{x}_2 \overline{Q}_1 Q_2,\ L_{\text{ДНФ}} = 3.$$

Задачи для самостоятельного решения

Для следующих функций построить минимальную ДНФ .

1. $f = (1110\ 1101)$	**11.** $f = (1011\ 1011\ 1101\ 0001)$
2. $f = (1110\ 0111)$	**12.** $f = (1101\ 0001\ 1010\ 0101)$
3. $f = (1110\ 1001)$	**13.** $f = (1111\ 1110\ 0001\ 0001)$
4. $f = (1111\ 0101)$	**14.** $f = (1010\ 1110\ 1010\ 0011)$
5. $f = (0101\ 0110)$	**15.** $f = (0101\ 1101\ 0101\ 1011)$
6. $f = (1001\ 0110)$	**16.** $f = (1110\ 0101\ 1110\ 1011)$
7. $f = (1000\ 0001)$	**17.** $f = (1011\ 0110\ 1101\ 1011)$
8. $f = (0001\ 0111)$	**18.** $f = (0010\ 0{*}{*}0\ 011{*}\ 1{*}01)$
9. $f = (0101\ 0011)$	**19.** $f = (0{*}01\ 1{*}00\ 0{*}01\ {*}{*}{*}{*})$
10. $f = (0110\ 1101)$	**20.** $f = (1{*}11\ {*}1{*}{*}\ {*}10{*}\ {*}{*}{*}0)$

2.7 Реализация КА в виде ПЛМ

Целью данного раздела, заключительного в главе, является консолидация всех изученных методов, в виде общего алгоритма реализации информационного процесса схемой, называемой *программируемой логической матрицей* (ПЛМ). ПЛМ является схемой из функциональных элементов специального вида, которая реализует термы, являющиеся нормальными формами. Нас будут интересовать ПЛМ, представляющие ДНФ. Как правило, для получения обратной связи, в состав ПЛМ вводятся элементы задержки. Схема, имеющая в своем составе элементы задержки, обладает конечной памятью, в связи с чем логические матрицы такого вида называют ПЛМ с памятью.

Пусть некоторый КА задан системой булевых функций $y_1(t), \ldots, y_p(t), Q_1(t), \ldots, Q_s(t)$, причем каждая функция из этой

системы представлена в виде ДНФ:

$$\begin{cases} y_i(t) = \bigvee\limits_{t=1}^{p_i} K_t \\ \dots\dots\dots\dots\dots \\ Q_j(t) = \bigvee\limits_{t=1}^{s_i} K_t, \end{cases} \tag{6}$$

где K_t есть элементарные конъюнкции от переменных $x_1(t), \ldots, x_k(t), Q_1(t-1), \ldots, Q_s(t-1)$. (Напомним, что переменная может как входить в элементарную конъюнкцию (с отрицанием или без), так и не входить вообще. Если элементарная конъюнкция содержит все переменные, она называется полной.) Далее, для облегчения восприятия, будем опускать зависимость переменных и функций от времени t там, где это возможно.

Программируемой логической матрицей (ПЛМ) с памятью, реализующей систему ДНФ (6), а вместе с ней и соответствующий КА, называется схема из функциональных элементов и задержек, в общем виде представленная на рис. 37.

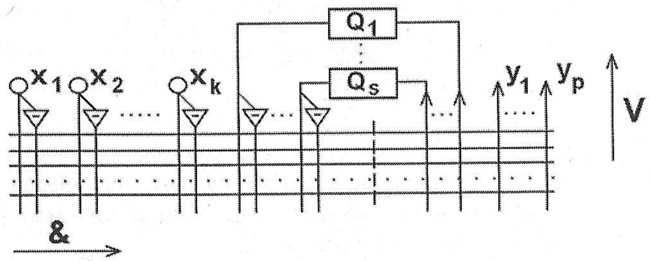


Рис. 37

При построении ПЛМ с памятью следует помнить о следующих условиях.

а) Каждой элементарной конъюнкции в ДНФ соответствует горизонтальная шина в ПЛМ. Если одна и та же элементарниая конъюнкция встречается в нескольких ДНФ системы, ее можно реализовать одной шиной.

б) Если переменная x_i (отрицание x_i) входит в некоторую элементарную конъюнкцию K_t, то на пересечении горизонтальной шины, соответствующей K_t и входа x_i (отрицания x_i) ставится контакт: , в противном случае контакта нет: .

Если некоторая элементарная конъюнкция K_t входит в ДНФ, реализующую y_i (Q_j), то на пересечении горизонтальной шины, соответствующей K_t и вертикальной шины, соответствующей y_i (Q_j), ставится такой же контакт, в противном случае контакта нет.

в) Для удобства восприятия в схеме ПЛМ не указываются конъюнкции и дизъюнкции. Считается, что все контакты, расположенные на одной горизонтальной шине соединены через многоместную конъюнкцию (на рисунке находятся слева от пунктирной линии); все контакты, расположенные на одной вертикальной выходной шине соединены через многоместную дизъюнкцию (на рисунке находятся справа от пунктирной линии);

г) Под сложностью ПЛМ понимается число ее горизонтальных шин.

Рассмотрим следующий пример.

Реализовать в виде ПЛМ с памятью следующую систему ДНФ:

$$\begin{cases} y_1 = x_1 x_2 \overline{x}_3 \vee x_1 \overline{Q}_1 \vee x_2 \overline{x}_3 Q_2 \\ y_2 = \overline{Q}_1 Q_2 \vee \overline{x}_1 \overline{x}_2 Q_2 \vee x_2 \overline{x}_3 Q_2 \\ Q_1 = x_1 x_2 \overline{x}_3 \vee Q_1 \vee \overline{x}_1 \overline{x}_2 Q_2 \\ Q_2 = x_2 \overline{x}_3 Q_2 \end{cases}$$

(Следует помнить, что задержки Q, стоящие в левой части равенств, а так же входы и выходы зависят от текущего момента времени t, тогда как задержки Q, стоящие в правой части равенств зависят от предыдущего момента $t-1$.)

Построим ПЛМ с памятью, реализующую заданную систему (рис. 38). Для удобства, слева от каждой горизонтальной шины поместим соответсвующую элементарную конъюнкцию. (В дальнейшем этого можно не делать.)

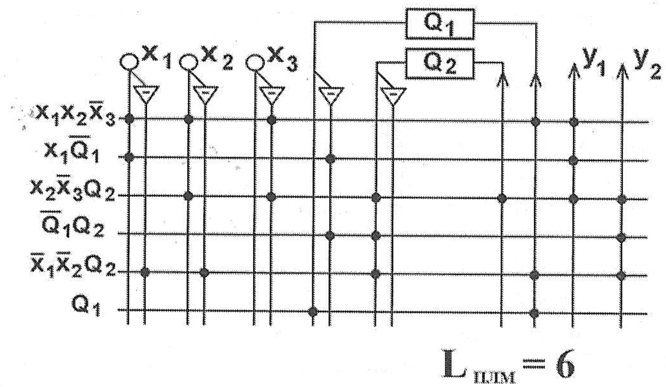


Рис. 38

Общий алгоритм реализации преобразования информации в виде ПЛМ.

1. Построить КА, осуществляющий заданное преобразование информации, в виде диаграммы или автоматной таблицы.
2. Минимизировать построенный КА.
3. Провести двоичное кодирование всех алфавитов КА.
4. Реализовать КА в виде системы булевых функций.
5. Представить каждую булеву функцию полученной системы в виде ДНФ достаточно простого вида.
6. Построить ПЛМ с памятью, реализующую множество полученных ДНФ.
7. Протестировать построенную ПЛМ.

Проведем полный цикл разработки преобразователя информации на примере автомата по продаже газированной воды.

Построить в виде ПЛМ с памятью КА, продающий газированную воду и выдающий сдачу. Автомат может принимать монеты достоинством 1 и 2 рубля, стакан газировки стоит 3 рубля. Кроме отверстий для приема монет и выдачи сдачи у автомата есть еще две кнопки "Налить"и "Сброс".

1. Решение начнем с представления процесса продажи газ.воды в виде диаграммы КА. На вход автомата могут поступать следующие сигналы:

"1пущена монета 1 руб.;

"2пущена монета 2 руб.;

"Сажата кнопка "Сброс";

"Нажата кнопка "Налить".

На выход автомат может выдать:

"1ыдать 1 руб.;

"2ыдать 2 руб.;

"3ыдать 3 руб.;

"*ичего не делать;

"Галить стакан газировки.

В процессе функционирования автомата могут встречаться следующие ситуации:

q_0 ("0") – первоначальное состояние, т. е. уплачено 0 руб.;

q_1 ("1") – уплачено 1 руб.;

q_2 ("2") – уплачено 2 руб.;

q_3 ("3") – уплачено 3 руб.;

Приведем часть команд, описывающих работу автомата.

В состоянии 0 опущена монета 1 руб., перейти в состояние 1, ничего не делать.

В состоянии 0 нажата кнопка Н, остаться в состоянии 0, ничего не делать.

В состоянии 1 опущена монета 2 руб., перейти в состояние 3, ничего не делать.

В состоянии 1 нажата кнопка С, перейти в состояние 0, выдать 1 руб.

В состоянии 2 опущена монета 1 руб., перейти в состояние 3, ничего не делать.

В состоянии 2 опущена монета 2 руб., перейти в состояние 3, выдать 1 руб.

В состоянии 2 нажата кнопка Н, остаться в состоянии 2, ничего не делать.

В состоянии 2 нажата кнопка С, перейти в состояние 0, выдать 2 руб.

В состоянии 3 опущена монета 1 руб., остаться в состоянии 3, выдать 1 руб.

В состоянии 3 нажата кнопка С, перейти в состояние 0, выдать 3 руб.

В состоянии 3 нажата кнопка Н, перейти в состояние 0, налить стакан газировки.

Полностью система команд данного КА описана в виде диаграммы на рис. 39.

Автоматная таблица имеет вид:

	q_0	q_1	q_2	q_3
1	$*/q_1$	$*/q_2$	$*/q_3$	$1/q_3$
2	$*/q_2$	$*/q_3$	$1/q_3$	$2/q_3$
Н	$*/q_0$	$*/q_1$	$*/q_2$	$Г/q_0$
С	$*/q_0$	$1/q_0$	$2/q_0$	$3/q_0$

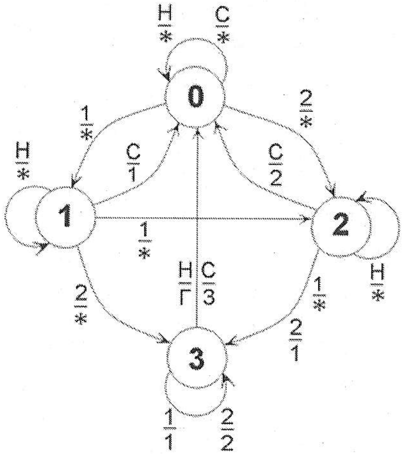


Рис. 39

2. Проведем минимизацию построенного КА на порядке входных символов 1, 2, Н, С:

$Q_0 = \{ \overset{****}{q_0} \}$, $Q_1 = \{ \overset{***1}{q_1} \}$, $Q_0 = \{ \overset{*1*2}{q_2} \}$, $Q_0 = \{ \overset{123}{q_3} \}$. Видно, что построенный первоначально КА уже был минимальным.

3. Произведем двоичное кодирование:

$A_{вх}$:	$1 = 00$ $2 = 01$ $Н = 10$ $С = 11$	$A_{вых}$:	$* = 000$ $1 = 001$ $2 = 010$ $3 = 011$ $Г = 100$	Q:	$q_0 = 00$ $q_1 = 01$ $q_2 = 10$ $q_3 = 11$

4. Представим КА в виде системы булевых функций. Эти функции будут зависеть от переменных x_1, x_2, соответствующих входному сигналу и переменных $Q_1(t-1)$, $Q_2(t-1)$, соответствующих предыдущему состоянию. Всего в системе будет пять функций: y_1, y_2, и y_3 соответствуют сигналу, подаваемому на выход, $Q_1(t)$ и $Q_2(t)$ соответствуют текущему состоянию.

x_1	x_2	Q_1	Q_2	y_1	y_2	y_3	Q_1	Q_2
0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	1	0
0	0	1	0	0	0	0	1	1
0	0	1	1	0	0	1	1	1
0	1	0	0	0	0	0	1	0
0	1	0	1	0	0	0	1	1
0	1	1	0	0	0	1	1	1
0	1	1	1	0	1	0	1	1
1	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	1
1	0	1	0	0	0	0	1	0
1	0	1	1	1	0	0	0	0
1	1	0	0	0	0	0	0	0
1	1	0	1	0	0	1	0	0
1	1	1	0	0	1	0	0	0
1	1	1	1	0	1	1	0	0

5. Представим каждую функцию системы в виде ДНФ наиболее простого вида.

Легко заметить, что $y_1 = x_1\overline{x}_2Q_1q_2$.

$x_1 x_2$ \ $Q_1 Q_2$	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	0	0	1	1
10	0	0	0	0

y_2:

x_1	x_2	x_3	x_4		
0	1	1	1	$\rightarrow$	$x_2Q_1Q_2$
1	1	1	1		

x_1	x_2	x_3	x_4		
1	1	1	1	$\rightarrow$	$x_1x_2Q_1$
1	1	1	0		

$y_2 = x_2Q_1Q_2 \vee x_1x_2Q_1$.

y_3:

x_1	x_2	x_3	x_4		
0	0	1	1	$\rightarrow$	$\overline{x}_1\overline{x}_2Q_1Q_2$

x_1	x_2	x_3	x_4		
0	1	1	0	$\rightarrow$	$\overline{x}_1x_2Q_1\overline{Q}_2$

x_1	x_2	x_3	x_4		
1	1	0	1	$\rightarrow$	$x_1x_2Q_2$
1	1	1	1		

$y_3 = \overline{x}_1\overline{x}_2Q_1Q_2 \vee \overline{x}_1x_2Q_1\overline{Q}_2 \vee x_1x_2Q_2$.

$x_1 x_2$ \ $Q_1 Q_2$	00	01	11	10
00	0	0	1	0
01	0	0	0	1
11	0	1	1	0
10	0	0	0	0

$x_1 x_2$ \ $Q_1 Q_2$	00	01	11	10
00	0	1	1	1
01	1	1	1	1
11	0	0	0	0
10	0	0	0	1

$Q_1(t)$:

x_1	x_2	x_3	x_4		
0	0	0	1		
0	0	1	1	$\rightarrow$	$\overline{x}_1Q_2$
0	1	0	1		
0	1	1	1		

x_1	x_2	x_3	x_4		
0	1	0	0		
0	1	0	1	$\rightarrow$	$\overline{x}_1x_2$
0	1	1	1		
0	1	1	0		

x_1	x_2	x_3	x_4		
0	0	1	0	$\rightarrow$	$\overline{x}_2Q_1\overline{Q}_2$
1	0	1	0		

$Q_1(t) = \overline{x}_1Q_2 \vee \overline{x}_1x_2 \vee \overline{x}_2Q_1\overline{Q}_2$.

$Q_2(t)$:

x_1	x_2	x_3	x_4		
0	0	1	1		
0	0	1	0	$\rightarrow$	$\overline{x}_1Q_1$
0	1	1	1		
0	1	1	0		

x_1	x_2	x_3	x_4		
0	0	0	0	$\rightarrow$	$\overline{x}_1\overline{x}_2\overline{Q}_2$
0	0	1	0		

$x_1 x_2$ \ $Q_1 Q_2$	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	0	0	0	0
10	0	1	0	0

$$\begin{array}{cccc|l} x_1 & x_2 & x_3 & x_4 & \\ 1 & 0 & 0 & 1 & \to \quad x_1\overline{x}_2\overline{Q}_1Q_2 \end{array} \qquad \begin{array}{cccc|l} x_1 & x_2 & x_3 & x_4 & \\ 0 & 1 & 0 & 1 & \\ 0 & 1 & 1 & 1 & \to \quad \overline{x}_1x_2Q_2 \end{array}$$

$$Q_2(t) = \overline{x}_1Q_1 \vee x_1\overline{x}_2\overline{Q}_1Q_2 \vee \overline{x}_1\overline{x}_2\overline{Q}_2 \vee \overline{x}_1x_2Q_2.$$

Таким образом, система булевых функций, задающая построенный КА имеет следующий вид:

$$\begin{cases} y_1 = x_1\overline{x}_2Q_1Q_2 \\ y_2 = x_2Q_1Q_2 \vee x_1x_2Q_1 \\ y_3 = \overline{x}_1\overline{x}_2Q_1Q_2 \vee \overline{x}_1x_2Q_1\overline{Q}_2 \vee x_1x_2Q_2 \\ Q_1(t) = \overline{x}_1Q_2 \vee \overline{x}_1x_2 \vee \overline{x}_2Q_1\overline{Q}_2 \\ Q_2(t) = \overline{x}_1Q_1 \vee x_1\overline{x}_2\overline{Q}_1Q_2 \vee \overline{x}_1\overline{x}_2\overline{Q}_2 \vee \overline{x}_1x_2Q_2 \end{cases} \qquad (7)$$

6. Построим ПЛМ с памятью, реализующую процесс продажи газированной воды (систему функций (7)). Если сигнал 1 подается только на выход y_3, то автомат выдает 1 руб. Если сигнал 1 подается только на выход y_2, то автомат выдает 2 руб. Если сигнал 1 подается и на y_3 и на y_2, то автомат выдает 3 руб. Если сигнал 1 подается подается на выход y_1, то автомат наливает стакан газированной воды. Первоначально в задержках содержатся нули.

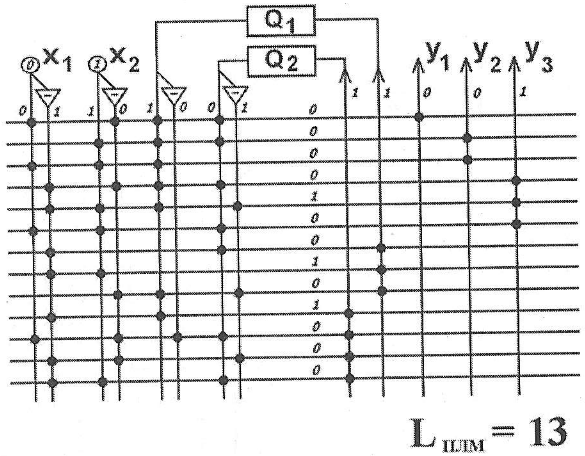


Рис. 40

На рис. 40 приведена данная ПЛМ с тестом на наборе 0110.

(Этот набор соответствует ситуации, когда в автомат опущено 2 руб., причем до этого уже было опущено столько же. Он должен выдать сдачу 1 руб. и перейти в состояние 3, т. е. 001 на выход и переход в состояние 11.)

7. Полное тестирование проводится для всех наборов длины четыре по аналогии с приведенным на рис. 40 для набора 0110. Оставляем его в качестве упражнения.

Задачи для самостоятельного решения

1. Построить ПЛМ с памятью, реализующую КА, заданный системой булевых функций из примера на стр. 68. Провести его полное тестирование.

Во всех следующих задачах построить ПЛМ с памятью, реализующую описанное преобразование.

2. Пусть $A_{вх} = \{3, 5\}$, $A_{вых} \subseteq \mathbb{Z}_7$, $Q = \mathbb{Z}_7$. Положим, что $\rho(a_j, q_i) = (a_j \cdot q_i)_7$, $\lambda(a_j, q_i) = (a_j + q_i)_7$.
Найти $A_{вых}$, построить ПЛМ.

3. Построить автомат, продающий лотерейные билеты. Автомат вмещает 7 билетов. Один билет стоит 1 руб. Автомат может принимать монеты в 1 и 2 руб., выдавая, соответственно 1 или 2 билета. Если в нем нет нужного числа билетов, то автомат недостающее выдает в виде сдачи.

4. Регулировщик Свистулькин управляет движением на перекрестке двух важнейших правительственных трасс, по каждой из которых могут проезжать только Чрезвычайные и Полномочные Послы. Каждую минуту к перекрестку подъезжает как минимум одна машина с юга или с запада. Свистулькину дано два указания:

а) из стратегических соображений пропускать через перекресток не более одной машины в минуту;

б) Чрезвычайные Послы важнее Полномочных, в связи с чем отдавать Чрезвычайным приоритет при пересечении перекрестка.

Конфликтные ситуации, когда одновременно к перекрестку подъезжают два равных по рангу Посла, Свистулькин разрешает следующим образом: дав в одной такой ситуации дорогу западному Послу, в следующей конфликтной ситуации он пропускает Посла с юга и наоборот.

В ходе важной правительственной реформы было решено заменить Свистулькина современной системой управления движением, способной различать ранги Послов. Система должна полностью дублировать действия Свистулькина. Вам, как главному инженеру, поручается разработка ее важнейшего узла — ПЛМ с памятью.

2.8 Задача, которую нельзя решить при помощи КА

При переходе от СФЭ к КА была рассмотрена задача сложения двух двоичных чисел произвольного, заранее неизвестного размера. Мы показали, что для ее решения устройство должно обладать некоторой памятью, всвязи с чем никакая СФЭ не сможет просуммировать два произвольных натуральных числа, хотя при ограничении их размера такая схема может быть построена (например двухразрядный параллельный сумматор).

Далее было проведено построение последовательного двоичного сумматора, складывающего числа произвольной длины. Данное устройство является КА и обладает конечной памятью. Факт ограниченности памяти заложен во внутреннее строение КА, и для того, чтобы увеличить размер его памяти, нужно внести в его устройство коренные изменения, по сути создать новый КА. Таким образом, главным свойством КА (и это отражено в его названии) является ограниченная или конечная память. Для решения некоторых задач этого ресурса вполне достаточно, но существует множество задач, которые не могут быть решены ни одним КА. Рассмотрим одну из них.

Следующей в ряду задач после нахождения суммы чисел произвольной длины логично рассмотреть задачу умножения двух двоичных чисел произвольной длины. Покажем, что никакой КА не сможет решить ее в общем виде. Сформулируем это утверждение

в следующем виде.

Теорема. Для любого КА найдется натуральное число, такое, что данный КА не сможет возвести его в двоичной записи в квадрат.

Доказательство. Будем говорить, что КА работает в автономном режиме, если на его вход подается один и тот же символ α. Покажем, что при работе КА в автономном режиме, на его выходе получается переодическая последовательность с длиной периода m. Эта последовательность устанавливается через k тактов работы в автономном режиме, причем выполняется условие $m + k \leq s$, где s — число состояний КА. Действительно, при подаче сигнала α КА из некоторого произвольного состояния q_i перейдет в предписанное состояние q_j, причем, попав в состояние q_i когда-либо в дальнейшем, КА снова перейдет в следующем такте в q_j.

Таким образом, можно однозначно построить цепочку состояний, в которые будет переходить КА при неограниченной подаче сигналов α. Так как число состояний КА ограничено, а сигналы можно подавать сколь угодно долго, то состояния в цепочке обязательно начнут повторяться. Пусть $q_{i_{k+1}}$ — первое состояние в этой цепочке, встретившееся два раза, и цепочка имеет вид:

$$q_{i_1} q_{i_2} \cdots q_{i_k} \underbrace{q_{i_{k+1}} q_{i_{k+2}} \cdots q_{i_p}} \underbrace{q_{i_{k+1}} q_{i_{k+2}} \cdots q_{i_p}} q_{i_{k+1}} \cdots$$

Все состояния в цепочке до второго появления $q_{i_{k+1}}$ различны. Очевидно, что их число не превышает s. Состояния $q_{i_1} \cdots q_{i_k}$ встречаются в цепочке один раз, после чего начинается период $q_{i_{k+1}} \cdots q_{i_p}$ (его длина по условию – m тактов). Отсюда видно, что $m + k \leq s$.

Отметим, что если при подаче s раз на вход сигнала α КА выдает на выход один и тот же сигнал β, то он будет выдавать β и далее до конца автономного режима. Это следует из того, что после s тактов автономной работы заведомо устанавливается периодичная работа КА, причем в течение всего периода автомат подает на выход сигнал β.

Перейдем теперь к доказательству утверждения теоремы. Пусть некоторый КА имеет s состояний. Покажем, что он не сможет возвести в квадрат число 2^{s+1}, двоичная запись которого имеет вид

$1\underbrace{0\ldots0}_{s+1}$. Его квадрат ($2^{2s+2}$) имеет вид $1\underbrace{0\ldots0}_{2s+2}$. Таким образом, автомат должен работать по схеме:

$$\left.\begin{array}{l} 0\overbrace{0\ldots0}^{s}1\overbrace{0\ldots0}^{s+1} \\ 0\underbrace{0\ldots0}_{s}1\underbrace{0\ldots0}_{s+1}\end{array}\right] \longrightarrow 1\underbrace{\overbrace{0\ldots0}^{s}0\overbrace{0\ldots0}^{s+1}}_{2s+2}.$$

Видно, что через $s+2$ тактов работы КА переходит в автономный режим и, работая в нем s тактов, выдает на выход одни нули. По доказанному, он будет выдавать нули и далее при работе в автономном режиме, т. е. не сможет выдать на $(2s+3)$-м такте единицу, что и требовалось доказать.

Здесь было показано, что никакой КА не сможет перемножить даже два одинаковых числа заранее неизвестной длины, что влечет отсутствие КА, который мог бы правильно перемножать произвольные числа заранее неизвестной длины.

3 Машины Тьюринга

3.1 Основные понятия и определения

3.1.1 Машина Тьюринга

Невозможность построения КА для решения проблемы умножения двух натуральных чисел является следствием того, что объем информации, которую должен запомнить такой автомат, неограниченно растет с ростом значений исходных чисел (при сложении такого не происходит). Отсюда напрямую следует, что для решения подобных проблем нужно устройство с бесконечным объемом памяти. На практике их можно решать при помощи ДПИ с потенциально бесконечной памятью. Последнее значит, что в каждый момент времени память ДПИ ограничена, но она может быть увеличена до необходимых для решения задачи размеров без радикального изменения самого преобразователя.

Преобразователь с бесконечным объемом памяти является, очевидно, умозрительным, т. е. может быть построен лишь мысленно,

тем не менее он может служить общей моделью для вполне реальных устройств. Ярким примером ДПИ с бесконечной памятью является *машина Тьюринга* (МТ).

Эта мысленная машина была предложена в середине 30-х годов XX в. английским математиком Аланом Тьюрингом как инструмент для изучения мощности алгоритмических процессов.

Машина Тьюринга состоит из

а) бесконечной в обе стороны ленты, разбитой на ячейки, в каждый момент времени в каждой ячейке может храниться ровно один символ из конечного внешнего алфавита $A = \{a_0, a_1, \ldots, a_n\}$;

б) *считывающе-записывающего устройства* (СЗУ), которое в каждый момент времени находится в одном состоянии из конечного множества состояний $Q = \{q_0, q_1, \ldots, q_s\}$;

в) множества команд, описывающих работу МТ.

Должны выполняться следующие условия.

Внешний алфавит A обязательно содержит специальный символ Λ, соответствующий пустой ячейке. В каждый момент на ленте МТ записано конечное число непустых символов.

За один такт СЗУ может выполнить следующие операции:

– воспринять символ из ячейки;
– записать в ячейку вместо одного символа другой (возможно тот же самый);
– изменить свое состояние (возможно на то же что и было);
– передвинуться на одну ячейку влево или вправо или остаться на месте.

Далее передвижение СЗУ на одну ячейку влево будем обозначать через L (left), передвижение на одну ячейку вправо будем обозначать через R (right), если СЗУ должно остаться на месте, обозначаем S.

Перед началом работы СЗУ воспринимает самый левый непустой символ на ленте и находится в состоянии q_1. Будем говорить, что работа МТ завершена корректно, если ее СЗУ находится в состоянии q_0 и воспринимает снова самый левый непустой символ на ленте.

Принцип работы МТ весьма схож с принципом работы КА. Действие, выполняемое в следующий такт зависит исключительно