

Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Д. В. Куприянова, И. В. Лукьянова, Ю. А. Луцик

АРИФМЕТИЧЕСКИЕ И ЛОГИЧЕСКИЕ ОСНОВЫ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

*Рекомендовано УМО по образованию в области информатики
и радиоэлектроники в качестве пособия для специальности
1-40 02 01 «Вычислительные машины, системы и сети»*

Минск БГУИР 2021

УДК 004.31(075)
ББК 32.973я7
К92

Рецензенты:
кафедра робототехнических систем
Белорусского национального технического университета
(протокол №5 от 27.12.2019);

заведующий лабораторией идентификации систем
государственного научного учреждения
«Объединенный институт проблем информатики
Национальной академии наук Беларуси»
доктор технических наук, профессор А. А. Дудкин

Куприянова, Д. В.
К92 Арифметические и логические основы вычислительной техники :
пособие /Д. В. Куприянова, И. В. Лукьянова, Ю. А. Луцик. – Минск :
БГУИР, 2021. – 72 с. : ил.
ISBN 978-985-543-573-1.

Приведены контрольные вопросы для проверки знаний по теме занятия, примеры выполненных заданий и задания для выполнения на практических занятиях.

УДК 004.31(075)
ББК 32.973я7

ISBN 978-985-543-573-1

© Куприянова Д. В., Лукьянова И. В., Луцик Ю. А., 2021
© УО «Белорусский государственный университет
информатики и радиоэлектроники», 2021

Введение

В пособии к каждой изучаемой теме приводится список контрольных вопросов, владение которыми необходимо для успешного выполнения практических заданий по теме занятия. Перед выполнением заданий по теме студент может ознакомиться с выполненными примерами, аналогичными тем заданиям, которые он будет выполнять. В решении примеров приводятся пояснения и ссылки на литературу, где рассматривается теоретический материал по теме.

Данное пособие может быть использовано студентами как на занятиях, так и при подготовке к ним.

Библиотека БГУИР

Тема 1. Системы счисления и операции с числами

Контрольные вопросы

1. Что такое система счисления (с/с)?
2. Какая система счисления называется позиционной?
3. Какие категории систем счисления можно выделить?
4. Какие существуют способы перевода чисел из одной системы счисления в другую?
5. Перечислите существующие критерии выбора системы счисления.

Арифметические операции над числами в различных системах счисления

Прежде чем приступать к выполнению практических заданий, рекомендуется ознакомиться с материалом в [4, с. 10–22]. Следует помнить, что правила арифметики для всех позиционных систем счисления одинаковы.

Рассмотрим несколько примеров на сложение и вычитание чисел в различных системах счисления.

Пример 1. Выполнить сложение и вычитание чисел в двоичной системе счисления: $A_2 = 10010111$ и $B_2 = 01001101$.

$$\begin{array}{r} A_2 = 10010111 \\ \underline{B_2 = 01001101} \\ A_2 - B_2 = 01001010 \end{array} \qquad \begin{array}{r} A_2 = 10010111 \\ \underline{B_2 = 01001101} \\ A_2 + B_2 = 11100100 \end{array}$$

Пример 2. Выполнить сложение и вычитание чисел в восьмеричной системе счисления: $A_8 = 5726$ и $B_8 = 1267$.

$$\begin{array}{r} A_8 = 5726 \\ \underline{B_8 = 1267} \\ A_8 - B_8 = 4437 \end{array} \qquad \begin{array}{r} A_8 = 5726 \\ \underline{B_8 = 1267} \\ A_8 + B_8 = 7215 \end{array}$$

Пример 3. Выполнить сложение и вычитание чисел в шестнадцатеричной системе счисления: $A_{16} = A7F6$ и $B_{16} = 4D69$.

$$\begin{array}{r} A_{16} = A7F6 \\ \underline{B_{16} = 4D69} \\ A_{16} - B_{16} = 5A8D \end{array} \qquad \begin{array}{r} A_{16} = A7F6 \\ \underline{B_{16} = 4D69} \\ A_{16} + B_{16} = F55F \end{array}$$

Перевод чисел из одной системы счисления в другую

Необходимо отметить, что для перевода целых и дробных чисел используются разные способы. В примерах используем универсальные методы – метод деления на основание системы счисления, в которую выполняется перевод, для целой части числа и метод умножения на основание системы счисления, в которую выполняется перевод, для дробной части числа. Необходимо отметить,

что все действия будут производиться в той системе счисления, из которой осуществляется перевод.

Пример 4. Выполнить перевод числа $A_{10} = 713,793$ из десятичной в двоичную систему счисления.

| | | | | | | | | | | |
|-----------|-----------|-----------|----------|-----------|-----------|-----------|-----------|----------|----------|----------|
| 713 | <u>2</u> | | | | | | | | | 0,793 |
| <u>6</u> | | 356 | <u>2</u> | | | | | | | <u>2</u> |
| 11 | <u>2</u> | 178 | <u>2</u> | | | | | | | 1,586 |
| <u>10</u> | | <u>15</u> | | <u>16</u> | <u>2</u> | | | | | <u>2</u> |
| 13 | <u>14</u> | 18 | <u>2</u> | <u>8</u> | | <u>44</u> | <u>2</u> | | | 1,172 |
| <u>12</u> | <u>16</u> | <u>18</u> | | 9 | <u>44</u> | <u>22</u> | <u>2</u> | | | <u>2</u> |
| 1 | <u>16</u> | 0 | <u>2</u> | <u>8</u> | | 0 | <u>22</u> | 11 | <u>2</u> | 0,344 |
| | 0 | | | 1 | | 0 | <u>10</u> | 5 | <u>2</u> | <u>2</u> |
| | | | | | | 1 | <u>4</u> | 2 | <u>2</u> | 0,688 |
| | | | | | | | 1 | <u>2</u> | <u>2</u> | <u>2</u> |
| | | | | | | | | 0 | | 1,376 |
| | | | | | | | | | | <u>2</u> |
| | | | | | | | | | | 0,752 |
| | | | | | | | | | | <u>2</u> |
| | | | | | | | | | | 1,504 |
| | | | | | | | | | | <u>2</u> |
| | | | | | | | | | | 1,008 |
| | | | | | | | | | | <u>2</u> |
| | | | | | | | | | | 0,016 |

Старшая цифра нового целого числа определяется последним результатом деления, а дальше записываются остатки от деления, начиная с последнего. В дробной части числа старшей цифрой новой дроби (после запятой) будет целая часть от первого умножения, далее производится умножение только дробной части числа, а цифры, которые получаются в результате последующих умножений в целой части результата, будут определять последующие цифры дроби (слева направо).

В результате получено двоичное число $A_2 = 1011001001,110010110$.

Пример 5. Выполнить перевод числа $A_8 = 265$ из восьмеричной системы счисления в десятичную, используя разложение в ряд.

$$265_8 = 2 \cdot 8^2 + 6 \cdot 8^1 + 5 \cdot 8^0 = 181_{10}.$$

Пример 6. Выполнить перевод числа $A_2 = 10110101$ из двоичной системы счисления в четверичную. Перевод будет выполняться по основанию, кратному степени двойки. Исходное число разобьем на диады, после чего запишем четверичный эквивалент каждой диады. В результате получим четверичное число:

$$\underline{10} \underline{11} \underline{01} \underline{01}_2 = 2311_4.$$

Пример 7. Выполнить перевод числа $A_4 = 2311$ из четверичной системы счисления в восьмеричную, используя метод деления на основание системы счисления. Число делим на основание новой системы счисления в старой системе счисления ($8 = 20_4$).

$$\begin{array}{r}
 2311 \quad | \underline{20} \\
 \underline{20} \quad 112 \quad | \underline{20} \\
 31 \quad \underline{100} \quad 2 \\
 \underline{20} \quad 12 \quad \swarrow \\
 111 \quad \swarrow \quad 2 \\
 \underline{100} \quad 6 \\
 11 \quad \swarrow \\
 5
 \end{array}$$

В результате деления мы получили три остатка от деления, теперь запишем результат перевода числа из одной системы счисления в другую: $A_8 = 265$.

Пример 8. Выполнить перевод числа $A_{10} = 181$ в шестнадцатеричную систему счисления. Для перевода числа использовать метод деления на основание системы счисления.

$$\begin{array}{r}
 181 \quad | \underline{16} \\
 \underline{16} \quad 11 \\
 21 \quad \swarrow \\
 \underline{16} \quad B \\
 5 \quad \swarrow \\
 5
 \end{array}$$

В результате деления получилось два остатка от деления, результат перевода числа: $A_{16} = B5$.

Практические задания

1. Вычислить сумму и разность чисел в указанной системе счисления:

- а) $A_2 = 1001,0010$; $B_2 = 0101,1011$;
- б) $A_2 = 1010,10001$; $B_2 = 0101,01010$;
- в) $A_2 = 10101,0101$; $B_2 = 01101,1001$;
- г) $A_4 = 2232,13$; $B_4 = 1312,12$;
- д) $A_4 = 2312,231$; $B_4 = 1232,313$;
- е) $A_8 = 5421,72$; $B_8 = 1463,16$;
- ж) $A_8 = 4576,17$; $B_8 = 2541,75$;
- з) $A_{16} = 94A,5FE$; $B_{16} = 5C7,B45$;
- и) $A_{16} = A23F,E8$; $B_{16} = 3CA9,5B$;
- к) $A_8 = 4712,26$; $B_8 = 1561,72$.

2. Перевести числа, заданные в десятичной системе счисления, в двоичную, восьмеричную и шестнадцатеричную системы счисления методом деления и умножения на основание системы счисления, в которую выполняется пе-

ревод. Проверить правильность выполненного перевода чисел, для этого перевести полученный результат обратно в десятичную систему счисления.

а) $A_{10} = 413,718$;

е) $A_{10} = 802,299$;

б) $A_{10} = 595,397$;

ж) $A_{10} = 658,315$;

в) $A_{10} = 528,805$;

з) $A_{10} = 413,718$;

г) $A_{10} = 315,421$;

и) $A_{10} = 813,922$;

д) $A_{10} = 239,945$;

к) $A_{10} = 592,712$.

Библиотека БГУИР

Тема 2. Коды чисел

Контрольные вопросы

1. Какие виды кодов чисел существуют?
2. Как заменить операцию вычитания операцией сложения?
3. Что такое дополнение числа?
4. Как записать число из естественной формы записи в дополнительном коде?
5. Как записать число из естественной формы записи в обратном коде?

Сложение чисел с фиксированной запятой в дополнительном и обратном коде

Прежде чем приступать к выполнению практического задания, рекомендуется ознакомиться с материалом в [4, с. 22–25].

Пример 1. Представить заданные числа в прямом, обратном и дополнительном коде:

$$\begin{array}{llll} A_2 = 10110110 & A_{10} = 3759 & A_8 = 5723 & A_{16} = A75C \\ B_2 = -10110110 & B_{10} = -3759 & B_8 = -5723 & A_{16} = -A75C \end{array}$$

| Прямой код | Обратный код | Дополнительный код |
|----------------------------------|-----------------------------------|-----------------------------------|
| $[A_2]_{\text{пр}} = 0.10110110$ | $[A_2]_{\text{обр}} = 0.10110110$ | $[A_2]_{\text{доп}} = 0.10110110$ |
| $[B_2]_{\text{пр}} = 1.10110110$ | $[B_2]_{\text{обр}} = 1.01001001$ | $[B_2]_{\text{доп}} = 1.01001010$ |
| $[A_{10}]_{\text{пр}} = 0.3759$ | $[A_{10}]_{\text{обр}} = 0.3759$ | $[A_{10}]_{\text{доп}} = 0.3759$ |
| $[B_{10}]_{\text{пр}} = 9.3759$ | $[B_{10}]_{\text{обр}} = 9.6240$ | $[B_{10}]_{\text{доп}} = 9.6241$ |
| $[A_8]_{\text{пр}} = 0.5723$ | $[A_8]_{\text{обр}} = 0.5723$ | $[A_8]_{\text{доп}} = 0.5723$ |
| $[B_8]_{\text{пр}} = 7.5723$ | $[B_8]_{\text{обр}} = 7.2054$ | $[B_8]_{\text{доп}} = 7.2055$ |
| $[A_{16}]_{\text{пр}} = 0.A75C$ | $[A_{16}]_{\text{пр}} = 0.A75C$ | $[A_{16}]_{\text{пр}} = 0.A75C$ |
| $[B_{16}]_{\text{пр}} = F.A75C$ | $[B_{16}]_{\text{пр}} = F.58A3$ | $[B_{16}]_{\text{пр}} = F.58A4$ |

Пример 2. Выполнить сложение чисел $A_2 = 1001101$ и $B_2 = 0100011$ в дополнительном коде. Здесь и далее (в пределах темы 2) сложение выполнять при всех случаях сочетания знаков слагаемых:

- а) $A > 0, B > 0$; б) $A > 0, B < 0$; в) $A < 0, B > 0$; г) $A < 0, B < 0$.

$$\begin{array}{r} \text{а) } [A]_{\text{доп}} = 0.1001101 \\ \underline{[B]_{\text{доп}} = 0.0100011} \\ [A+B]_{\text{доп}} = 0.1110000 \end{array}$$

$$\begin{array}{r} \text{б) } [A]_{\text{доп}} = 0.1001101 \\ \underline{[B]_{\text{доп}} = 1.1011101} \\ [A+B]_{\text{доп}} = 10.0101010 \\ \quad \uparrow \text{отбрасывается} \end{array}$$

$$\begin{array}{r} \text{в) } [A]_{\text{доп}} = 1.0110011 \\ \underline{[B]_{\text{доп}} = 0.0100011} \\ [A+B]_{\text{доп}} = 1.1010110 \end{array}$$

$$\begin{array}{r} \text{г) } [A]_{\text{доп}} = 1.0110011 \\ \underline{[B]_{\text{доп}} = 1.1011101} \\ [A+B]_{\text{доп}} = 11.0010000 \\ \quad \uparrow \text{отбрасывается} \end{array}$$

В примерах «б» и «г» выделенный серым цветом разряд (единица) выходит за пределы разрядной сетки и отбрасывается.

Пример 3. Выполнить сложение чисел $A_2 = 0,010010$ и $B_2 = 0,101101$ в обратном коде.

$$\begin{array}{r} \text{а) } [A]_{\text{обр}} = 0.010010 \\ \underline{[B]_{\text{обр}} = 0.101101} \\ [A+B]_{\text{обр}} = 0.111111 \end{array}$$

$$\begin{array}{r} \text{б) } [A]_{\text{обр}} = 0.010010 \\ \underline{[B]_{\text{обр}} = 1.010010} \\ [A+B]_{\text{обр}} = 1.100100 \end{array}$$

$$\begin{array}{r} \text{в) } [A]_{\text{обр}} = 1.101101 \\ \underline{[B]_{\text{обр}} = 0.101101} \\ [A+B]_{\text{обр}} = 10.011010 \\ \phantom{[A+B]_{\text{обр}}} \xrightarrow{\quad} 1 \\ \hline 0.011011 \end{array}$$

$$\begin{array}{r} \text{г) } [A]_{\text{обр}} = 1.101101 \\ \underline{[B]_{\text{обр}} = 1.010010} \\ [A+B]_{\text{обр}} = 10.111111 \\ \phantom{[A+B]_{\text{обр}}} \xrightarrow{\quad} 1 \\ \hline 1.000000 \end{array}$$

В примерах «б» и «г» разряд (единица) выходит за пределы разрядной сетки и в случае обратного кода выполняется циклический перенос с добавлением единицы в младший разряд.

Пример 4. Выполнить сложение чисел $A_2 = 101101$ и $B_2 = 110001$ в обратном модифицированном коде.

$$\begin{array}{r} \text{а) } [A]_{\text{обр}}^M = 00.101101 \\ \underline{[B]_{\text{обр}}^M = 00.110001} \\ [A+B]_{\text{обр}}^M = 01.011110 \\ \text{(переполнение)} \end{array}$$

$$\begin{array}{r} \text{б) } [A]_{\text{обр}}^M = 00.101101 \\ \underline{[B]_{\text{обр}}^M = 11.001110} \\ [A+B]_{\text{обр}}^M = 11.111011 \end{array}$$

$$\begin{array}{r} \text{в) } [A]_{\text{обр}}^M = 11.010010 \\ \underline{[B]_{\text{обр}}^M = 00.110001} \\ [A+B]_{\text{обр}}^M = 100.000011 \\ \phantom{[A+B]_{\text{обр}}^M} \xrightarrow{\quad} 1 \\ \hline 00.000100 \end{array}$$

$$\begin{array}{r} \text{г) } [A]_{\text{обр}}^M = 11.010010 \\ \underline{[B]_{\text{обр}}^M = 11.001110} \\ [A+B]_{\text{обр}}^M = 110.100000 \\ \text{(переполнение)} \end{array}$$

В примерах «а» и «г» было зафиксировано переполнение разрядной сетки, признаком которого является различие значений знаковых разрядов (01 или 10).

Пример 5. Выполнить сложение чисел $A_2 = 101101$ и $B_2 = 111001$ в дополнительном модифицированном коде.

$$\begin{array}{r} \text{а) } [A]_{\text{доп}}^M = 00.101101 \\ \underline{[B]_{\text{доп}}^M = 00.111001} \\ [A+B]_{\text{доп}}^M = 01.101000 \\ \text{(переполнение)} \end{array}$$

$$\begin{array}{r} \text{б) } [A]_{\text{доп}}^M = 00.101101 \\ \underline{[B]_{\text{доп}}^M = 11.000111} \\ [A+B]_{\text{доп}}^M = 11.110100 \end{array}$$

$$\begin{array}{r} \text{в) } [A]_{\text{доп}}^M = 11.010011 \\ \underline{[B]_{\text{доп}}^M = 00.111001} \\ [A+B]_{\text{доп}}^M = 100.001100 \\ \phantom{[A+B]_{\text{доп}}^M} \uparrow \text{отбрасывается} \end{array}$$

$$\begin{array}{r} \text{г) } [A]_{\text{доп}}^M = 11.010011 \\ \underline{[B]_{\text{доп}}^M = 11.000111} \\ [A+B]_{\text{доп}}^M = 110.011010 \\ \text{(переполнение)} \end{array}$$

В примерах «а» и «г» знаковые разряды различаются (01 или 10), это свидетельствует о наличии переполнения разрядной сетки.

Практическое задание

Вычислить сумму чисел в обратном и дополнительном модифицированном коде при всех случаях сочетания знаков слагаемых:

- а) $A = 11110011, B = 11001111$;
- б) $A = 10110001, B = 10111101$;
- в) $A = 11011111, B = 01110111$;
- г) $A = 11101101, B = 00110010$;
- д) $A = 10101111, B = 00011011$;
- е) $A = 00001001, B = 11001111$;
- ж) $A = 00000111, B = 10000111$;
- з) $A = 01111101, B = 11111101$;
- и) $A = 00000001, B = 01101100$;
- к) $A = 00011110, B = 01111100$.

Тема 3. Формы представления чисел. Числа с фиксированной и плавающей запятой. Сложение чисел с плавающей запятой

Контрольные вопросы

1. Приведите общую форму записи числа с фиксированной запятой.
2. Приведите общую форму записи числа с плавающей запятой.
3. Какое число называется нормализованным?
4. Какие варианты денормализации возможны при выполнении арифметических операций над числами с плавающей запятой?
5. Приведите пример классической разрядной сетки для чисел с плавающей запятой (мантисса, порядок).
6. Приведите пример разрядной сетки используемой в ЭВМ для чисел с плавающей запятой (мантисса, характеристика).
7. Приведите последовательность действий при сложении чисел с плавающей запятой.

Сложение чисел с плавающей запятой

Пример 1. $M_A = +0,01010$, $p_A = +101$;
 $M_B = +0,10101$, $p_B = +010$.

$$[M_A]_{\text{доп}} = 0,01010.$$

$$[M_B]_{\text{доп}} = 0,10101.$$

$$\begin{aligned} \text{Разность порядков } p &= [p_A]_{\text{доп}} + [-p_B]_{\text{доп}} = 0.101 \\ &+ \underline{1.110} \\ &1\ 0.011 > 0 \end{aligned}$$

Так как разность порядков $[p_A - p_B]_{\text{доп}} > 0$, то сдвигу подвергается мантисса M_B . При каждом сдвиге мантиссы на один разряд из положительной разности порядков производим последовательное вычитание единицы до тех пор, пока в результате не будет получен нуль (т. е. порядки чисел A и B не будут выравнены):

$$\begin{array}{ll} [M_B]_{\text{доп}} = 0,10101 & 0.011 \\ & [-1]_{\text{доп}} = \underline{1.111} \\ [M_B]_{\text{доп}} = 0,010101 & 0.010 \neq 0 \\ [M_B]_{\text{доп}} = 0,0010101 & [-1]_{\text{доп}} = \underline{1.111} \\ & 0.001 \neq 0 \\ [M_B]_{\text{доп}} = 0,00010101 & [-1]_{\text{доп}} = \underline{1.111} \\ & 0.000 = 0 \end{array}$$

Порядки чисел A и B выравнены (мантисса M_B числа B сдвинута). Теперь можно выполнять сложение мантисс M_A и M_B .

$$\begin{aligned} [M_A]_{\text{доп}} &= 0,01010 \\ [M_B]_{\text{доп}} &= \underline{0,00010} 101 \\ [M_{A+B}]_{\text{доп}} &= 0,01100 101 \quad - \text{произошла денормализация результата.} \end{aligned}$$

Тема 4. Машинные методы умножения чисел

Контрольные вопросы

1. Что такое частичное произведение и частичная сумма?
2. По какому признаку разделяются четыре алгоритма умножения на группы?
3. В каких алгоритмах умножения выполняется сдвиг частичных сумм?
4. В каких алгоритмах умножения выполняется сдвиг частичных произведений?
5. Какие основные элементы образуют структурную схему устройства умножения?

Умножение чисел в прямых кодах

Прежде чем приступить к выполнению практического задания, рекомендуется ознакомиться с материалом в [4, с. 37–42].

Рассмотрим примеры умножения чисел алгоритмами умножения А, Б, В и Г.

Пример 1. Выполнить умножение двух чисел $M_H = 0,10101$ и $M_T = 0,10111$, используя алгоритм А.

| | | |
|-----------------|-----------------|---|
| $M_H = 0,10101$ | $M_T = 0,10111$ | |
| | $b_5 \quad b_1$ | |
| 0,00000 | | |
| 0,10101 | | $\Pi_1 = M_H \cdot b_1$ |
| 0,10101 | | Σ_1 |
| 0,01010 | 1 | $\Sigma_1 \cdot 2^{-1}$ |
| 0,10101 | | $\Pi_2 = M_H \cdot b_2$ |
| 0,11111 | 1 | Σ_2 |
| 0,01111 | 11 | $\Sigma_2 \cdot 2^{-1}$ |
| 0,10101 | | $\Pi_3 = M_H \cdot b_3$ |
| 1,00100 | 11 | Σ_3 |
| 0,10010 | 011 | $\Sigma_3 \cdot 2^{-1}$ |
| 0,00000 | | $\Pi_4 = M_H \cdot b_4$ |
| 0,10010 | 011 | Σ_4 |
| 0,01001 | 0011 | $\Sigma_4 \cdot 2^{-1}$ |
| 0,10101 | | $\Pi_5 = M_H \cdot b_5$ |
| 0,11110 | 0011 | Σ_5 |
| 0,01111 | 00011 | $\Sigma_5 \cdot 2^{-1} = M_H \cdot M_T$ |

В результате получаем $M_H \cdot M_T = +0,0111100011$.

Пример 2. Выполнить умножение двух чисел $M_H = 0,10101$ и $M_T = 0,10111$, используя алгоритм Б.

$$M_H = 0,10101$$

$$M_T = 0,10111$$

$$b_5 \quad b_1$$

$$\begin{array}{r} 0,0000000000 \\ 0,0000010101 \\ \hline 0,0000010101 \\ 0,0000101010 \\ \hline 0,0000111111 \\ 0,0001010100 \\ \hline 0,0010010011 \\ 0,0000000000 \\ \hline 0,0010010011 \\ 0,0101010000 \\ \hline 0,0111100011 \end{array}$$

$$\begin{array}{l} \Sigma_0 \\ \Pi_1 = M_H \cdot b_1 \cdot 2^0 \\ \Sigma_1 \\ \Pi_2 = M_H \cdot b_2 \cdot 2^1 \\ \Sigma_2 \\ \Pi_3 = M_H \cdot b_3 \cdot 2^2 \\ \Sigma_3 \\ \Pi_4 = M_H \cdot b_4 \cdot 2^3 = 0, \text{ т. к. } b_4 = 0 \\ \Sigma_4 \\ \Pi_5 = M_H \cdot b_5 \cdot 2^4 \\ \Sigma_5 \end{array}$$

В результате получаем $M_H \cdot M_T = +0,0111100011$.

Пример 3. Выполнить умножение двух чисел $M_H = 0,10101$ и $M_T = 0,10111$, используя алгоритм В.

$$M_H = 0,10101$$

$$M_T = 0,10111$$

$$b_5 \quad b_1$$

$$\begin{array}{r} 0,0000000000 \\ 0,0000000000 \\ 0,0000010101 \\ \hline 0,0000010101 \\ 0,0000101010 \\ \hline 0,0000000000 \\ 0,0000101010 \\ 0,0001010100 \\ \hline 0,0000010101 \\ 0,0001101001 \\ 0,0011010010 \\ \hline 0,0000010101 \\ 0,0011100111 \\ 0,0111001110 \\ \hline 0,0000010101 \\ 0,0111100011 \end{array}$$

$$\begin{array}{l} \Sigma_0 \quad \text{сдвиг} \\ \Sigma_0 \cdot 2 \\ \Pi_1 = M_H \cdot b_5 \\ \Sigma_1 \\ \Sigma_1 \cdot 2 \\ \Pi_2 = M_H \cdot b_4 \\ \Sigma_2 \\ \Sigma_2 \cdot 2 \\ \Pi_3 = M_H \cdot b_3 \\ \Sigma_3 \\ \Sigma_3 \cdot 2 \\ \Pi_4 = M_H \cdot b_2 \\ \Sigma_4 \\ \Sigma_4 \cdot 2 \\ \Pi_5 = M_H \cdot b_1 \\ \Sigma_5 \end{array}$$

В результате получаем $M_H \cdot M_T = +0,0111100011$.

Пример 4. Выполнить умножение двух чисел $M_H = 0,10101$ и $M_T = 010111$, используя алгоритм Г.

$$M_H = 0,10101 \quad M_T = 0,10111$$

$$b_5 \quad b_1$$

| | |
|--------------|------------------------------------|
| 0,0000000000 | Σ_0 |
| 0,0101010000 | $P_1 = M_H \cdot b_5 \cdot 2^{-1}$ |
| 0,0101010000 | Σ_1 |
| 0,0000000000 | $P_2 = M_H \cdot b_4 \cdot 2^{-2}$ |
| 0,0101010000 | Σ_2 |
| 0,0001010100 | $P_3 = M_H \cdot b_3 \cdot 2^{-3}$ |
| 0,0110100100 | Σ_3 |
| 0,0000101010 | $P_4 = M_H \cdot b_2 \cdot 2^{-4}$ |
| 0,0111001110 | Σ_4 |
| 0,0000010101 | $P_5 = M_H \cdot b_1 \cdot 2^{-5}$ |
| 0,0111100011 | Σ_5 |

В результате получаем $M_H \cdot M_T = +0,0111100011$.

В алгоритме Г производится сдвиг частичных произведений вправо. В этом случае в освобождающийся разряд (справа от запятой) заносится значение, совпадающее со знаковым разрядом.

Практическое задание

Выполнить умножение чисел, используя алгоритмы умножения А, Б, В и Г:

- а) $M_H = 0,10001$, $M_T = 0,10111$;
- б) $M_H = 0,10101$, $M_T = 0,11001$;
- в) $M_H = 0,00110$, $M_T = 0,11101$;
- г) $M_H = 0,11111$, $M_T = 0,11011$;
- д) $M_H = 0,01010$, $M_T = 0,00110$;
- е) $M_H = 0,01110$, $M_T = 0,11111$;
- ж) $M_H = 0,11011$, $M_T = 0,01111$;
- з) $M_H = 0,11100$, $M_T = 0,11111$;
- и) $M_H = 0,00100$, $M_T = 0,10000$;
- к) $M_H = 0,01110$, $M_T = 0,01011$.

Тема 5. Умножение с хранением переносов

Контрольные вопросы по теме

1. В чем основное отличие умножения с хранением переносов от четырех рассмотренных ранее обычных методов умножения?
2. В чем состоит преимущество умножения с хранением переносов от рассмотренных ранее методов умножения?
3. Какие изменения необходимо внести в схему, чтобы она реализовывала умножение с хранением переносов?
4. Какой алгоритм умножения (А, ..., Г) используется при умножении с хранением переносов? Ответ обоснуйте.
5. Обязательно ли на последнем шаге выполнять умножение с учетом переносов?

Умножение с хранением переносов

Прежде чем приступить к выполнению практического задания, рекомендуется ознакомиться с материалом в [4, с. 42-43].

Рассмотрим пример умножения двух чисел с хранением переносов. Применяется алгоритм А.

| | | |
|-----------------|-----------------|-------------------------|
| $M_H = 0,11101$ | $M_T = 0,10111$ | |
| | $b_5 \quad b_1$ | |
| 0,00000 | | Σ_0 |
| 0,00000 | | перенос |
| 0,11101 | | $P_1 = M_H \cdot b_1$ |
| 0,11101 | | Σ_1 |
| 0,00000 | | перенос |
| 0,00110 | 1 | $\Sigma_1 \cdot 2^{-1}$ |
| 0,11101 | | $P_2 = M_H \cdot b_2$ |
| 0,11011 | 1 | Σ_2 |
| 0,00100 | | перенос |
| 0,01101 | 11 | $\Sigma_2 \cdot 2^{-1}$ |
| 0,11101 | | $P_3 = M_H \cdot b_3$ |
| 0,10100 | 11 | Σ_3 |
| 0,01101 | | перенос |
| 0,01010 | 011 | $\Sigma_3 \cdot 2^{-1}$ |
| 0,00000 | | $P_4 = M_H \cdot b_4$ |
| 0,00111 | 011 | Σ_4 |
| 0,01000 | | перенос |
| 0,00011 | 1011 | $\Sigma_4 \cdot 2^{-1}$ |
| 0,11101 | | $P_5 = M_H \cdot b_5$ |
| 1,01000 | 1011 | Σ_5 |
| 0,10100 | 01011 | $\Sigma_5 \cdot 2^{-1}$ |

В результате получаем $M_H \cdot M_T = +0,1010001011$.

При использовании этого метода, в отличие от рассмотренных в предыдущей теме, межразрядный перенос при формировании частичной суммы не выполняется, а сохраняется в специальном выделенном для этого регистре. В каждом такте при формировании очередной частичной суммы выполняется суммирование трех значений: предыдущей частичной суммы, сохраненного переноса и очередного частичного произведения. В последнем такте сложение выполняется уже с учетом межразрядного переноса.

Необходимо отметить, что данный метод может применяться только для алгоритма А (это связано с направлением сдвига частичных сумм).

Практическое задание

Перемножить числа с хранением переносов:

- а) $M_H = 0,11011$, $M_T = 0,10111$;
- б) $M_H = 0,01101$, $M_T = 0,10101$;
- в) $M_H = 0,10111$, $M_T = 0,01011$;
- г) $M_H = 0,11101$, $M_T = 0,11011$;
- д) $M_H = 0,11010$, $M_T = 0,11101$;
- е) $M_H = 0,11001$, $M_T = 0,11111$;
- ж) $M_H = 0,11101$, $M_T = 0,10111$;
- з) $M_H = 0,01110$, $M_T = 0,01101$;
- и) $M_H = 0,11011$, $M_T = 0,11111$;
- к) $M_H = 0,11010$, $M_T = 0,11011$.

Тема 6. Умножение чисел на два разряда множителя одновременно в прямом коде

Контрольные вопросы

1. Какое основное требование должно выполняться при умножении на два разряда множителя одновременно?
2. Какая из пар множителя подвергается преобразованию? Почему?
3. Можно ли умножать на 4, 8, ... разрядов множителя? Если да, то какие преимущества и проблемы будут при этом?
4. Какие проблемы могут возникнуть при умножении по алгоритму В и Г?
5. Почему при преобразовании пары вначале надо сделать перенос в нее (из младшей пары), а затем ее преобразовать?

Умножение чисел на два разряда в прямом коде

Прежде чем приступить к выполнению практического задания, рекомендуется ознакомиться с материалом в [4, с. 43–46].

Рассмотрим пример умножения чисел на два разряда множителя одновременно в прямом коде (алгоритм А): $M_N = 0101$, $M_T = 0111101101$.

Для преобразования множителя необходимо разбить его на пары. Разбиение на пары для целых чисел следует начинать с младших разрядов.

Вначале выполним преобразование множителя, начиная с младшей пары:

$$M_T = \underline{01}\underline{11}\underline{10}\underline{11}\underline{01};$$

$$M_T^p = \underline{10}\underline{000}\underline{10}\underline{101}.$$

Из преобразованного множителя видно, что в нем содержатся пары: 00, 01, $0\bar{1}$ и 10. В соответствии с этим возможны следующие частичные произведения:

$$[M_N]_{\text{доп}} = 0.0101; \quad [-M_N]_{\text{доп}} = 1.1011; \quad [2M_N]_{\text{доп}} = 0.1010.$$

Умножение выполняется по алгоритму А.

| | | |
|--------|------|--|
| 0.0000 | | Σ_0 |
| 0.0101 | | $\Pi_1 = [M_N]_{\text{доп}}$ |
| 0.0101 | | Σ_1 |
| 0.0001 | 01 | $\Sigma_1 \cdot 2^{-2}$ |
| 1.1011 | | $\Pi_2 = [-M_N]_{\text{доп}} (0\bar{1})$ |
| 1.1100 | 01 | Σ_2 |
| 1.1111 | 0001 | $\Sigma_2 \cdot 2^{-2}$ |
| 1.1011 | | $\Pi_3 = [-M_N]_{\text{доп}} (0\bar{1})$ |
| 1.1010 | 0001 | Σ_3 |

| | | | |
|---------|--|------------|-------------------------------|
| 11.1110 | | 100001 | $\Sigma_3 \cdot 2^{-2}$ |
| 11.1111 | | 10100001 | $\Sigma_4 \cdot 2^{-2}$ |
| 00.1010 | | | $\Pi_5 = [2M_H]_{\text{доп}}$ |
| 00.1001 | | 10100001 | Σ_5 |
| 00.0010 | | 0110100001 | $\Sigma_5 \cdot 2^{-2}$ |

В результате получим $M_H \cdot M_T = +00.00100110100001$.

Практическое задание

Перемножить следующие числа, используя метод умножения на два разряда множителя одновременно, для всех алгоритмов умножения (А, Б, В и Г):

- а) $M_H = 01011$, $M_T = 110011110$;
- б) $M_H = 0110$, $M_T = 011101010$;
- в) $M_H = 01100$, $M_T = 111000111$;
- г) $M_H = 01101$, $M_T = 001001001$;
- д) $M_H = 0110$, $M_T = 101010110$;
- е) $M_H = 01110$, $M_T = 110010011$;
- ж) $M_H = 01111$, $M_T = 101101101$;
- з) $M_H = 0011$, $M_T = 010010001$;
- и) $M_H = 0100$, $M_T = 000111000$;
- к) $M_H = 0010$, $M_T = 001111100$.

Тема 7. Умножение чисел в дополнительном коде

Контрольные вопросы

1. Какие алгоритмы машинного умножения можно использовать для умножения чисел в дополнительном коде?
2. Какие поправки необходимо выполнить при умножении целых чисел в дополнительном коде?
3. Какие поправки необходимо выполнить при умножении дробных чисел при умножении чисел в дополнительном коде?
4. При каком знаковочетании сомножителей поправка на требуется и почему?
5. Когда вводится поправка при умножении чисел (в начале, в конце или на любом этапе умножения)?

Умножение чисел в дополнительном коде

Прежде чем приступить к выполнению практического задания, рекомендуется ознакомиться с материалом в [4, с. 46–51].

Умножение чисел в дополнительном коде отличается от умножения в прямом коде тем, что в некоторых случаях в конце операции умножения необходимо выполнить добавление поправки. Величина поправки и ее вывод приведены в [4].

Пример 1. Выполнить умножение чисел в дополнительном коде (алгоритм В). Если $M_N < 0$, $M_T > 0$, то поправка может и не вводиться [4], но в этом случае частичное произведение берется вместе со знаком.

| | |
|--------------------------------|-----------------------------------|
| $-M_N = 0,10101$ | $+M_T = 0,10111$ |
| $[M_N]_{\text{доп}} = 1,01011$ | $b_5 \quad b_1$ |
| $0,0000000000$ | Σ_0 |
| $1,1111101011$ | $\Pi_1 = b_5 \cdot M_N \cdot 2^1$ |
| <hr style="width: 100%;"/> | Σ_1 |
| $1,1111101011$ | $\Sigma_1 \cdot 2^1$ |
| $1,1111010110$ | $\Pi_2 = b_4 \cdot M_N = 0$ |
| $0,0000000000$ | Σ_2 |
| <hr style="width: 100%;"/> | $\Sigma_2 \cdot 2^1$ |
| $1,1111010110$ | $\Pi_3 = b_3 \cdot M_N$ |
| $1,1111010110$ | Σ_3 |
| <hr style="width: 100%;"/> | $\Sigma_3 \cdot 2^1$ |
| $1,1111010111$ | $\Pi_4 = b_2 \cdot M_N$ |
| $1,1110010111$ | Σ_4 |
| $1,1100101110$ | $\Sigma_4 \cdot 2^1$ |
| <hr style="width: 100%;"/> | $\Pi_5 = b_1 \cdot M_N$ |
| $1,1111101011$ | Σ_5 |
| <hr style="width: 100%;"/> | |
| $1,1000011101$ | |

В результате получим $M_N \cdot M_T = -0,1000011101$.

Пример 2. Выполнить умножение чисел в дополнительном коде (алгоритм Г): $M_H = -0,0110$, $M_T = -0,0011$.

$$\begin{aligned} [M_H]_д &= 1,1010 \\ [M_T]_д &= 1,1101 \\ \Delta &= [-M_H]_{доп} = 0,0110 \end{aligned}$$

| | |
|-------------|--------------------------|
| 0,0000 0000 | $\sum_0^ч$ |
| 1,1101 0000 | $P_1 = M_H \cdot 2^{-1}$ |
| 1,1101 0000 | $\sum_1^ч$ |
| 1,1110 1000 | $P_2 = M_H \cdot 2^{-2}$ |
| 1,1011 1000 | $\sum_2^ч$ |
| 1,1111 1010 | $P_4 = M_H \cdot 2^{-3}$ |
| 1,1011 0010 | $\sum_4^ч$ |
| 0,0110 | $\Delta = [-M_H]_{доп}$ |
| 0,0001 0010 | $[M_H \cdot M_T]_{доп}$ |

В результате получим $M_H \cdot M_T = +0,00010010$.

Практическое задание

Перемножить числа в дополнительном коде для следующих случаев сочетания знаков сомножителей: $M_H > 0, M_T < 0$; $M_H < 0, M_T > 0$; $M_H < 0, M_T < 0$.

- а) $M_H = 0,11011, M_T = 0,00110$;
- б) $M_H = 0,01011, M_T = 0,10010$;
- в) $M_H = 0,11101, M_T = 0,00101$;
- г) $M_H = 0,11010, M_T = 0,11110$;
- д) $M_H = 0,01001, M_T = 0,01010$;
- е) $M_H = 0,00011, M_T = 0,00110$;
- ж) $M_H = 0,10101, M_T = 0,10000$;
- з) $M_H = 0,11000, M_T = 0,00111$;
- и) $M_H = 0,11011, M_T = 0,11001$;
- к) $M_H = 0,00111, M_T = 0,10101$.

Тема 8. Умножение чисел на два разряда в дополнительном коде

Контрольные вопросы

1. Какие алгоритмы умножения допускают некорректное преобразование множителя при умножении в прямых кодах на два разряда множителя?
2. В чем заключается основное отличие в преобразовании пар множителя в дополнительном коде от преобразования в прямом коде?
3. Вводятся ли поправки при умножении в дополнительном коде на два разряда множителя? Поясните.
4. Что происходит с переносом (если он возникает) из старшей пары множителя?
5. Почему при преобразовании очередной пары вначале надо сделать ее преобразование, а затем добавить в нее перенос?

Умножение чисел на два разряда в дополнительном коде

Прежде чем приступить к выполнению практического задания, необходимо ознакомиться с материалом в [4, с. 51–54].

Рассмотрим пример умножения чисел на два разряда множителя одновременно (алгоритм Г): $M_n = -0101$, $M_t = -1011111$, $[M_t]_{\text{доп}} = \underline{10100001}$.

Выполним преобразование множителя в дополнительном коде. Данный метод умножения отличается от аналогичного метода умножения в прямом коде тем, что преобразованию подвергается не только пара 11, а также пара 10. Обоснование этому приведено в [4].

$$[M_t]_{\text{доп}}^n = 0\bar{1}\bar{1}00001.$$

В результате преобразования множителя получены пары 00, 01, 0 $\bar{1}$ и $\bar{1}0$. Подготовим соответствующие этим парам множимые:

$$\begin{aligned}[-M_n] &= 0.0101; \\ [M_n] &= 1.1011; \\ [-2M_n] &= 0.1010.\end{aligned}$$

В рассматриваемом примере выделенные разряды показывают сдвиг частичного произведения.

$$\begin{array}{r} 0,0000 \\ 0,000101 \\ \hline 0,000101 \\ 0,00001010 \\ \hline 0,00011110 \\ 1,11111111011 \\ \hline 0,000111011011 \end{array}$$

$$\begin{aligned} \Sigma_0 \\ P_1 &= [-M_n] \cdot 2^{-2} \\ \Sigma_1 \\ P_2 &= [-2M_n] \cdot 2^{-4} \\ \Sigma_2 &= \Sigma_3 \\ P_4 &= [M_n] \cdot 2^{-8} \\ \Sigma_4 &= [M_n \cdot M_t]_{\text{доп}} \end{aligned}$$

В результате получим $M_H \cdot M_T = +0,000111011011$.

Практическое задание

Перемножить числа, используя метод умножения на два разряда множителя одновременно в дополнительном коде для следующих случаев сочетания знаков сомножителей: $M_H > 0, M_T < 0$; $M_H < 0, M_T > 0$; $M_H < 0, M_T < 0$.

- а) $M_H = 1010, M_T = 1011110$;
- б) $M_H = 0111, M_T = 0101010$;
- в) $M_H = 1100, M_T = 1000111$;
- г) $M_H = 1101, M_T = 0010011$;
- д) $M_H = 1011, M_T = 1010110$;
- е) $M_H = 1110, M_T = 1110011$;
- ж) $M_H = 0001, M_T = 1001101$;
- з) $M_H = 1011, M_T = 0110001$;
- и) $M_H = 0110, M_T = 0001000$;
- к) $M_H = 1010, M_T = 0011100$.

Библиотека БГУИР

Тема 9. Машинные методы деления чисел

Контрольные вопросы

1. Какие существуют подходы к выполнению операции деления?
2. Какова последовательность действий в алгоритме деления с восстановлением остатка?
3. Какова последовательность действий в алгоритме деления без восстановления остатка?
4. Какова последовательность действий в алгоритме деления в дополнительных кодах?
5. Приведите схему устройства деления.

Машинные методы деления чисел

Прежде чем приступить к выполнению практического задания, рекомендуется ознакомиться с материалом в [4, с. 56–60].

Пример 1. Выполнить деление чисел в прямом коде с восстановлением остатка.

$$\begin{array}{r}
 D_M = 0,10101 \qquad D_T = 0,11011 \\
 [-D_T]_{\text{доп}} = 1,00101 \\
 \\
 \begin{array}{r}
 0,10101 \quad D_M (A_0) \\
 + 1,00101 \quad [-D_T]_{\text{доп}} \\
 \hline
 \mathbf{1},11010 \quad A_1 < 0 \\
 + 0,11011 \quad + D_T \text{ (восстановление } A_0) \\
 \hline
 0,10101 \quad A_0 \\
 1,01010 \quad A_0 \cdot 2 \\
 + 1,00101 \quad [-D_T]_{\text{доп}} \\
 \hline
 \mathbf{0},01111 \quad A_2 > 0 \\
 0,11110 \quad A_2 \cdot 2 \\
 + 1,00101 \quad [-D_T]_{\text{доп}} \\
 \hline
 \mathbf{0},00011 \quad A_3 > 0 \\
 0,00110 \quad A_3 \cdot 2 \\
 + 1,00101 \quad [-D_T]_{\text{доп}} \\
 \hline
 \mathbf{1},01011 \quad A_4 < 0 \\
 + 0,11011 \quad + D_T \text{ (восстановление } A_0) \\
 \hline
 \mathbf{0},00110 \quad A_3 \cdot 2 \\
 0,01100 \quad (A_3 \cdot 2) \cdot 2 \\
 + 1,00101 \quad [-D_T]_{\text{доп}} \\
 \hline
 \mathbf{1},10001 \quad A_5 < 0
 \end{array}
 \end{array}$$

ЧТ = 0, 1 1 0 0 . . .

Выделенные жирным шрифтом знаковые разряды остатков определяют очередную цифру частного, и надо или не надо восстанавливать предыдущий остаток. Если знаковый разряд остатка равен «1» (остаток отрицателен), то в

очередной разряд частного заносится «0» и выполняется восстановление предыдущего остатка. Если же знаковый разряд равен «0», то в частное пишется «1» и восстанавливать остаток не надо.

В результате выполнения деления получим $ЧТ = 0,1100$.

Пример 2. Выполнить деление чисел в прямом коде без восстановления остатка.

$$ДМ = 0,10101$$

$$ДТ = 0,11011$$

| | | |
|-----------|----------------|---------------|
| | 0,10101 | ДМ (A_0) |
| | + 1,00101 | $[-ДТ]_{доп}$ |
| $A_1 < 0$ | <u>1,11010</u> | $A_1 < 0$ |
| | 1,10100 | $A_0 \cdot 2$ |
| | + 0,11011 | $[ДТ]_{доп}$ |
| $A_2 > 0$ | <u>0,01111</u> | A_2 |
| | 0,11110 | $A_2 \cdot 2$ |
| | + 1,00101 | $[-ДТ]_{доп}$ |
| $A_3 > 0$ | <u>0,00011</u> | A_3 |
| | 0,00110 | $A_3 \cdot 2$ |
| | + 1,00101 | $[-ДТ]_{доп}$ |
| $A_4 < 0$ | <u>1,01011</u> | A_4 |
| | 0,10110 | $A_3 \cdot 2$ |
| | + 0,11011 | $[ДТ]_{доп}$ |
| $A_5 < 0$ | <u>1,10001</u> | A_5 |

ЧТ = 0, 1 1 0 0 . . .

В этом случае цифра частного определяется так же, как и в предыдущем примере, кроме того, знак остатка определяет, с каким знаком добавлять делитель к остатку после его сдвига. Если знак остатка положительный, он сдвигается влево и делитель добавляется к нему со знаком минус; если же знак остатка отрицательный, он сдвигается влево и делитель добавляется к нему со знаком плюс.

В результате получим $ЧТ = 0,11000$.

Пример 3. Выполнить деление чисел в дополнительном коде.

В этом случае цифра частного определяется из сравнения знаков получаемого остатка и делителя. При совпадении знаков в очередной разряд частного заносится «1», иначе «0». Из этого же сравнения знаков определяется и то действие (добавлять или вычитать делитель), которое будет выполнено на следующем шаге деления. То есть если знаки остатка и делителя не совпадают, то к сдвинутому остатку добавляется делитель, иначе (при совпадении знаков) к сдвинутому остатку добавляется делитель с противоположным знаком.

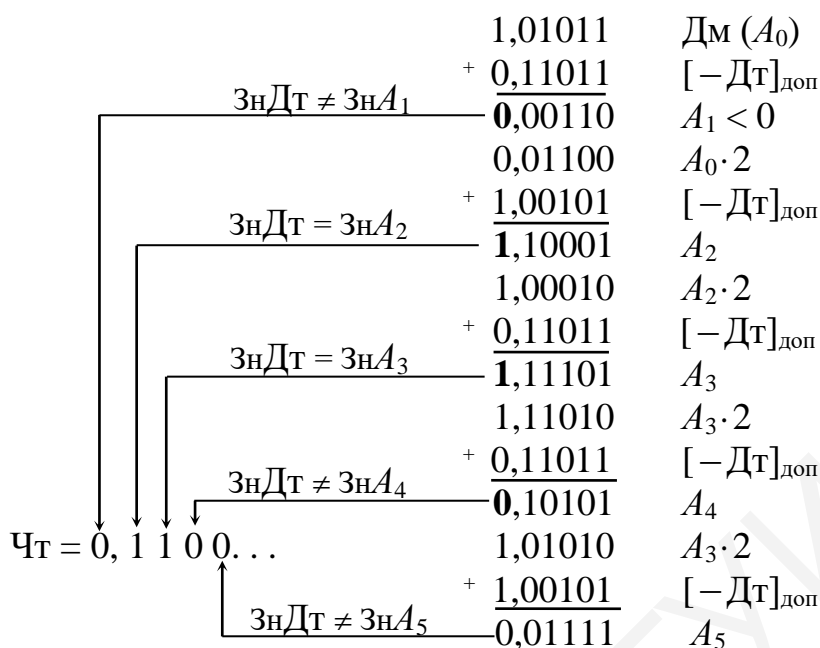
$$ДМ = -0,10101$$

$$[ДМ]_{доп} = 1,01011$$

$$ДТ = -0,11011$$

$$[ДТ]_{доп} = 1,00101$$

$$[-ДТ]_{доп} = 0,11011$$



Далее вычисляется знак полученного частного:

$$\text{ЗнЧТ} = \text{ЗнДм} \oplus \text{ЗнДТ} = 1 \oplus 1 = 0.$$

В результате получим ЧТ = +0,110001.

Практическое задание

Выполнить деление чисел в прямом коде с восстановлением остатков, без восстановления остатков и в дополнительном коде для следующих случаев сочетания знаков: Дм > 0, ДТ < 0; Дм < 0, ДТ > 0; Дм < 0, ДТ < 0.

- а) Дм = 0,11010, ДТ = 0,11110;
- б) Дм = 0,00111, ДТ = 0,01010;
- в) Дм = 0,10100, ДТ = 0,11001;
- г) Дм = 0,01101, ДТ = 0,10011;
- д) Дм = 0,10011, ДТ = 0,11101;
- е) Дм = 0,01110, ДТ = 0,10011;
- ж) Дм = 0,10001, ДТ = 0,11001;
- з) Дм = 0,01011, ДТ = 0,10101;
- и) Дм = 0,10110, ДТ = 0,11010;
- к) Дм = 0,01010, ДТ = 0,11100.

**Тема 10. Методы минимизации булевых функций. Алгоритм Квайна.
Функционально полный базис**

Контрольные вопросы

1. С какой целью выполняется минимизация булевой функции?
2. Назовите формы представления булевых функций.
3. Перечислите основные правила и законы булевой алгебры.
4. Что такое простая импликанта, сокращенная и минимальная форма функции?
5. Что такое функционально полный базис? Приведите его примеры.

Минимизация булевых функций методом Квайна

Рассмотрим этапы метода Квайна. Пусть, например, задана функция

$$f_{\text{СДНФ}}(x_1 x_2 x_3 x_4) = V(0, 2, 5, 6, 7, 8, 9, 10, 12, 13, 14),$$

$$f_{\text{СДНФ}} = \underbrace{\bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4}_1 \vee \underbrace{\bar{x}_1 \bar{x}_2 x_3 \bar{x}_4}_2 \vee \underbrace{\bar{x}_1 x_2 \bar{x}_3 x_4}_3 \vee \underbrace{\bar{x}_1 x_2 x_3 \bar{x}_4}_4 \vee \underbrace{\bar{x}_1 x_2 x_3 x_4}_5 \vee \underbrace{x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4}_6 \vee \underbrace{x_1 \bar{x}_2 \bar{x}_3 x_4}_7 \vee \underbrace{x_1 \bar{x}_2 x_3 \bar{x}_4}_8 \vee \underbrace{x_1 \bar{x}_2 x_3 x_4}_9 \vee \underbrace{x_1 x_2 \bar{x}_3 \bar{x}_4}_{10} \vee \underbrace{x_1 x_2 \bar{x}_3 x_4}_{11}.$$

На первом этапе метода Квайна выполняется переход от функции, заданной в форме СДНФ, к сокращенной ДНФ. Суть метода заключается в последовательном выполнении всех возможных склеиваний и затем всех поглощений, что приводит к сокращенной ДНФ.

В результате выполнения первого этапа склеивания, в котором участвуют все конституенты единицы, составляющие СДНФ функции, получаются конъюнкции ранга на единицу меньше. Представим первый этап склеивания в виде табл. 10.1.

Таблица 10.1

| Номера конституент, участвующих в склеивании | Переменная, по которой склеиваются конституенты | Полученная в результате склеивания импликанта | Номер полученной импликанты |
|--|---|---|-----------------------------|
| 1 | x_3 | $\bar{x}_1 \bar{x}_2 \bar{x}_4$ | 1 |
| 1-2 | x_1 | $\bar{x}_2 \bar{x}_3 \bar{x}_4$ | 2 |
| 2-4 | x_2 | $\bar{x}_1 \bar{x}_3 \bar{x}_4$ | 3 |
| 2-8 | x_1 | $\bar{x}_2 \bar{x}_3 x_4$ | 4 |
| 3-5 | x_3 | $\bar{x}_1 x_2 x_4$ | 5 |
| 3-10 | x_1 | $x_2 \bar{x}_3 x_4$ | 6 |
| 4-11 | x_1 | $x_2 x_3 \bar{x}_4$ | 7 |

| 1 | 2 | 3 | 4 |
|------|-------|-------------------------------------|----|
| 4-5 | x_4 | $\overline{x_1 x_2 x_3}$ | 8 |
| 6-7 | x_4 | $\overline{x_1 x_2} \overline{x_3}$ | 9 |
| 6-8 | x_3 | $\overline{x_1 x_2} \overline{x_4}$ | 10 |
| 6-9 | x_2 | $\overline{x_1 x_3} \overline{x_4}$ | 11 |
| 7-10 | x_2 | $\overline{x_1 x_3} x_4$ | 12 |
| 8-11 | x_2 | $\overline{x_1 x_3} \overline{x_4}$ | 13 |
| 9-10 | x_4 | $\overline{x_1 x_2} \overline{x_3}$ | 14 |
| 9-11 | x_3 | $\overline{x_1 x_2} \overline{x_4}$ | 15 |

На каждом этапе склеивания необходимо проанализировать, все ли исходные конъюнкции (на первом этапе это конститутенты единицы) приняли участие в склеивании, и если какая-то из них ни с чем не склеилась, то она становится простой импликантой. В данном примере все конъюнкции исходной функции приняли участие в склеивании, образовав новые конъюнкции, а значит, простых импликант на первом этапе не образовалось.

Полученные конъюнкции снова подвергаются склеиванию. Второй этап склеивания представлен в табл. 10.2.

Таблица 10.2

| Номера конъюнкций, участвующих в склеивании | Переменная, по которой склеиваются конъюнкции | Полученная в результате склеивания импликанта | Номер полученной импликанты |
|---|---|---|-----------------------------|
| 1-10 | x_1 | $\overline{x_2} \overline{x_4}$ | 1 |
| 2-4 | x_3 | $\overline{x_2} \overline{x_4}$ | 1 |
| 3-13 | x_1 | $\overline{x_3} \overline{x_4}$ | 2 |
| 4-7 | x_2 | $\overline{x_3} \overline{x_4}$ | 2 |
| 9-14 | x_2 | $\overline{x_1} \overline{x_3}$ | 3 |
| 10-15 | x_2 | $\overline{x_1} \overline{x_4}$ | 4 |
| 11-12 | x_4 | $\overline{x_1} \overline{x_3}$ | 3 |
| 11-13 | x_3 | $\overline{x_1} \overline{x_4}$ | 4 |

На втором этапе склеивания исходные конъюнкции с номерами 5, 6, 8 ($\overline{x_1 x_2 x_4}$, $\overline{x_2 x_3 x_4}$, $\overline{x_1 x_2 x_3}$) не приняли участия в склеивании, а значит, они включаются во множество простых импликант и войдут в состав сокращенной ДНФ. Остальные наборы в результате склеивания дали еще четыре конъюнкции низшего ранга: $\overline{x_2} \overline{x_4}$, $\overline{x_3} \overline{x_4}$, $\overline{x_1} \overline{x_3}$, $\overline{x_1} \overline{x_4}$.

Проанализировав возможность склеивания вновь полученных на третьем этапе импликант, мы видим, что дальнейшее склеивание невозможно, а значит, четыре конъюнкции, полученные на втором этапе склеивания, тоже являются простыми импликантами и войдут в состав сокращенной ДНФ:

$$f_{\text{сокр ДНФ}} = \bar{x}_1 \bar{x}_2 \bar{x}_4 \vee \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 \bar{x}_2 \bar{x}_3 \vee \bar{x}_2 \bar{x}_4 \vee \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 \bar{x}_3 \vee \bar{x}_1 \bar{x}_4.$$

Для формирования тупиковой ДНФ строится импликантная таблица, строки которой отмечаются простыми импликантами сокращенной ДНФ, а столбцы – конституентами единицы исходной СДНФ (табл. 10.3). В строке напротив каждой простой импликанты ставится метка под теми наборами (конституентами единицы), на которых она принимает значение «1». Соответствующие конституенты поглощаются (покрываются) данной простой импликантой.

Таблица 10.3

| Исходные наборы простых импликант | $\bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4$ 1 | $\bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4$ 2 | $\bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4$ 3 | $\bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4$ 4 | $\bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4$ 5 | $\bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4$ 6 |
|--|--|--|--|--|--|--|
| $\bar{x}_1 \bar{x}_2 \bar{x}_4$ | | | * | | * | |
| $\bar{x}_2 \bar{x}_3 \bar{x}_4$ | | | * | | | |
| $\bar{x}_1 \bar{x}_2 \bar{x}_3$ | | | | * | * | |
| $\bar{x}_2 \bar{x}_4$ | * | * | | | | * |
| $\bar{x}_3 \bar{x}_4$ | | * | | * | | |
| $\bar{x}_1 \bar{x}_3$ | | | | | | * |
| $\bar{x}_1 \bar{x}_4$ | | | | | | * |

Окончание табл. 10.3

| Исходные наборы простых импликант | $\bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4$ 7 | $\bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4$ 8 | $\bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4$ 9 | $\bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4$ 10 | $\bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4$ 11 |
|--|--|--|--|---|---|
| $\bar{x}_1 \bar{x}_2 \bar{x}_4$ | | | | | |
| $\bar{x}_2 \bar{x}_3 \bar{x}_4$ | | | | * | |
| $\bar{x}_1 \bar{x}_2 \bar{x}_3$ | | | | | |
| $\bar{x}_2 \bar{x}_4$ | | * | | | |
| $\bar{x}_3 \bar{x}_4$ | | * | | | * |
| $\bar{x}_1 \bar{x}_3$ | * | | * | * | |
| $\bar{x}_1 \bar{x}_4$ | | * | * | | * |

Из общего числа простых импликант необходимо отобрать их минимальное количество, исключив лишние. Формирование тупиковых форм и выбор минимального покрытия начинается с выявления обязательных простых импликант, т. е. таких, которые (и только они) покрывают некоторый исходный набор. В данном случае обязательными простыми импликантами будут $\bar{x}_2 \bar{x}_4$ и $x_1 \bar{x}_3$, т. к. только $\bar{x}_2 \bar{x}_4$ покрывает конституенту с номером 1 и только $x_1 \bar{x}_3$ покрывает конституенту с номером 7. Эти наборы обязательно войдут в минимальное покрытие. Эти обязательные простые импликанты покрывают также конституенты с номерами 2, 6, 8, 9, 10. Остались непокрытыми конституенты с номерами 3, 4, 5, 11. Для того чтобы их покрыть, по таблице выбираем простые импликанты $\bar{x}_1 x_2 x_4$, которые покроют 3 и 5 конституенты, и $x_3 \bar{x}_4$, которые покроют 4 и 11 конституенты.

Таким образом, минимальная ДНФ функции

$$f_{\text{МДНФ}} = \bar{x}_2 \bar{x}_4 \vee x_1 \bar{x}_3 \vee \bar{x}_1 x_2 x_4 \vee x_3 \bar{x}_4.$$

Практическое задание

Выполнить минимизацию булевой функции методом Квайна:

а) $f_{\text{СДНФ}} = x_1 x_2 x_3 \bar{x}_4 + \bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 + \bar{x}_1 x_2 x_3 \bar{x}_4 + x_1 x_2 \bar{x}_3 x_4 + \bar{x}_1 \bar{x}_2 x_3 x_4 + x_1 \bar{x}_2 \bar{x}_3 x_4 + \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 + x_1 \bar{x}_2 x_3 \bar{x}_4 + \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4;$

б) $f_{\text{СДНФ}} = x_1 x_2 x_3 \bar{x}_4 + \bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 + \bar{x}_1 x_2 x_3 \bar{x}_4 + \bar{x}_1 x_2 x_3 x_4 + \bar{x}_1 \bar{x}_2 x_3 x_4 + x_1 \bar{x}_2 \bar{x}_3 x_4 + x_1 x_2 x_3 x_4 + x_1 \bar{x}_2 x_3 \bar{x}_4 + \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4;$

в) $f_{\text{СДНФ}} = x_1 \bar{x}_2 x_3 x_4 + \bar{x}_1 x_2 x_3 \bar{x}_4 + \bar{x}_1 x_2 x_3 x_4 + x_1 x_2 \bar{x}_3 x_4 + \bar{x}_1 \bar{x}_2 x_3 x_4 + x_1 \bar{x}_2 \bar{x}_3 x_4 + \bar{x}_1 x_2 \bar{x}_3 x_4 + x_1 \bar{x}_2 x_3 \bar{x}_4 + \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4;$

г) $f_{\text{СДНФ}} = x_1 \bar{x}_2 x_3 x_4 + \bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 + \bar{x}_1 x_2 x_3 \bar{x}_4 + x_1 x_2 \bar{x}_3 x_4 + \bar{x}_1 \bar{x}_2 x_3 x_4 + x_1 \bar{x}_2 \bar{x}_3 x_4 + \bar{x}_1 \bar{x}_2 x_3 \bar{x}_4 + x_1 \bar{x}_2 x_3 \bar{x}_4;$

д) $f_{\text{СДНФ}} = x_1 x_2 x_3 \bar{x}_4 + x_1 x_2 \bar{x}_3 \bar{x}_4 + \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 + x_1 \bar{x}_2 x_3 x_4 + \bar{x}_1 x_2 \bar{x}_3 x_4 + x_1 \bar{x}_2 \bar{x}_3 x_4 + x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 + x_1 \bar{x}_2 x_3 \bar{x}_4 + \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4;$

е) $f_{\text{СДНФ}} = x_1 x_2 \bar{x}_3 x_4 + \bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 + \bar{x}_1 x_2 x_3 \bar{x}_4 + x_1 \bar{x}_2 x_3 x_4 + \bar{x}_1 \bar{x}_2 x_3 x_4 + x_1 \bar{x}_2 \bar{x}_3 x_4 + \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 + x_1 \bar{x}_2 x_3 \bar{x}_4 + \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4;$

ж) $f_{\text{СДНФ}} = x_1 x_2 x_3 x_4 + \bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 + \bar{x}_1 x_2 \bar{x}_3 x_4 + \bar{x}_1 x_2 x_3 x_4 + \bar{x}_1 \bar{x}_2 x_3 x_4 + x_1 \bar{x}_2 \bar{x}_3 x_4 + \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 + x_1 \bar{x}_2 x_3 \bar{x}_4;$

з) $f_{\text{СДНФ}} = \bar{x}_1 x_2 \bar{x}_3 x_4 + x_1 x_2 \bar{x}_3 \bar{x}_4 + \bar{x}_1 x_2 x_3 \bar{x}_4 + x_1 \bar{x}_2 x_3 x_4 + \bar{x}_1 \bar{x}_2 x_3 \bar{x}_4 + x_1 x_2 \bar{x}_3 x_4 + \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 + x_1 \bar{x}_2 x_3 \bar{x}_4;$

и) $f_{\text{СДНФ}} = x_1 x_2 x_3 \bar{x}_4 + \bar{x}_1 \bar{x}_2 x_3 \bar{x}_4 + \bar{x}_1 x_2 \bar{x}_3 x_4 + \bar{x}_1 x_2 x_3 x_4 + \bar{x}_1 \bar{x}_2 x_3 x_4 + x_1 \bar{x}_2 x_3 \bar{x}_4 + \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 + x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4;$

к) $f_{\text{СДНФ}} = x_1 x_2 x_3 \bar{x}_4 + \bar{x}_1 \bar{x}_2 x_3 \bar{x}_4 + \bar{x}_1 x_2 \bar{x}_3 x_4 + x_1 x_2 \bar{x}_3 \bar{x}_4 + \bar{x}_1 \bar{x}_2 x_3 x_4 + x_1 \bar{x}_2 x_3 \bar{x}_4 + \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 + x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4;$

Тема 11. Минимизация булевых функций. Карты Вейча (Карно).
Минимизация неполностью определенных булевых функций

Контрольные вопросы

1. Что такое соседние наборы?
2. В чем отличие карт Вейча от карт Карно?
3. Что такое код Грея?
4. Могут ли контуры на минимизирующей карте пересекаться?
5. Сформулируйте основные правила минимизации булевых функций с помощью карт.

Минимизация булевых функций с применением карт Вейча

Таблица 11.1

| № п/п | $x_1x_2x_3$ | f |
|----------|-------------|-----|
| 0 | 0 0 0 | 0 |
| 1 | 0 0 1 | 0 |
| 2 | 0 1 0 | 0 |
| 3 | 0 1 1 | 1 |
| 4 | 1 0 0 | 0 |
| 5 | 1 0 1 | 0 |
| 6 | 1 1 0 | 1 |
| 7 | 1 1 1 | 1 |

Более подробно теоретические основы методов минимизации булевых функций (БФ) с использованием карт Вейча (Карно) рассмотрены в [4, с. 95–103]. Остановимся на некоторых примерах минимизации функций, представленных в дизъюнктивной нормальной форме (ДНФ) и конъюнктивной нормальной форме (КНФ), а также более общими случаями их представления – совершенной ДНФ и совершенной КНФ. В качестве исходной формой записи булевой функции для ее минимизации может быть использована таблица истинности (табл. 11.1), схема, описывающая поведение БФ (рис. 11.1), аналитическая форма записи и математическая (более компактная) форма записи БФ. Примеры их приведены ниже.

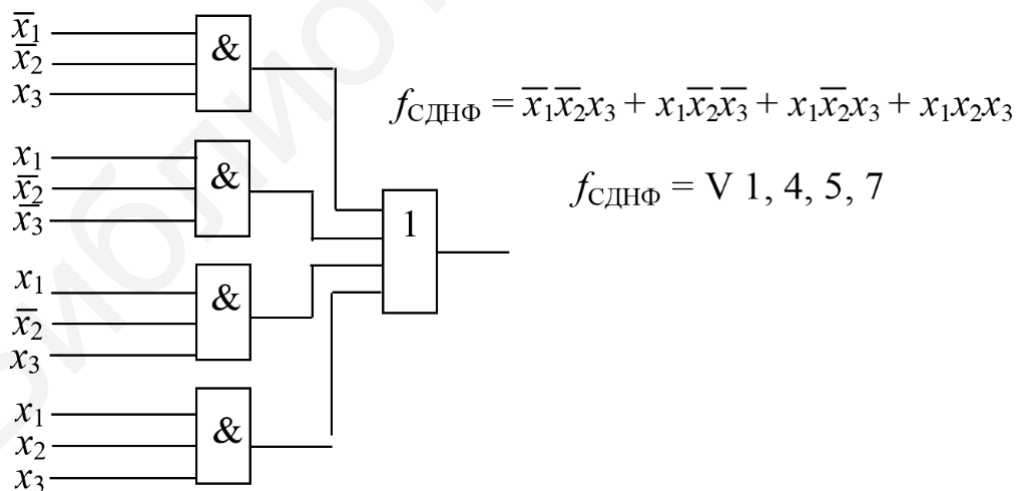


Рис. 11.1. Схемное задание БФ

Не останавливаясь на теоретических основах построения карты Вейча и минимизации БФ с ее использованием (подробно рассмотренным в [4]), приведем несколько примеров ее использования.

Пример 1. Выполнить минимизацию БФ (получить $f_{\text{МДНФ}}$ и $f_{\text{МКНФ}}$), заданной аналитически:

$$f_{\text{СДНФ}} = x_1x_2\bar{x}_3\bar{x}_4 + x_1x_2x_3\bar{x}_4 + x_1\bar{x}_2x_3\bar{x}_4 + x_1\bar{x}_2\bar{x}_3\bar{x}_4 + x_1\bar{x}_2x_3x_4 + x_1\bar{x}_2\bar{x}_3x_4 + \bar{x}_1x_2x_3x_4 + \bar{x}_1\bar{x}_2\bar{x}_3x_4.$$

В результате минимизации получаем на карте четыре контура, объединяющие все восемь единичных наборов, рис. 11.2.

| | | | | | | | |
|-------|--|-------|---|---|---|--|-------|
| | | x_2 | | | | | |
| x_1 | | 1 | 1 | 1 | 1 | | x_4 |
| | | 0 | 0 | 1 | 1 | | |
| | | 0 | 1 | 0 | 1 | | |
| | | 0 | 0 | 0 | 0 | | |
| | | x_3 | | | | | |

Рис. 11.2. Контурь на единичных наборах для примера 1

Запишем получившуюся (единственную) тупиковую форму:

$$f_{\text{МДНФ}} = x_1\bar{x}_4 + x_1\bar{x}_2 + \bar{x}_2\bar{x}_3x_4 + \bar{x}_1x_2x_3x_4.$$

В случае формирования минимальной БФ в форме КНФ контуры образуются на нулевых наборах (рис 11.3).

| | | | | | | | |
|-------|--|-------|---|---|---|--|-------|
| | | x_2 | | | | | |
| x_1 | | 1 | 1 | 1 | 1 | | x_4 |
| | | 0 | 0 | 1 | 1 | | |
| | | 0 | 1 | 0 | 1 | | |
| | | 0 | 0 | 0 | 0 | | |
| | | x_3 | | | | | |

Рис. 11.3. Контурь на нулевых наборах для примера 1

В результате минимизации также получены четыре контура, образованные на нулевых наборах. Запишем получившуюся тупиковую форму:

$$f_{\text{МКНФ}} = (\bar{x}_2+x_3+\bar{x}_4) (\bar{x}_1+\bar{x}_2+\bar{x}_4) (x_1+ x_2+ \bar{x}_3) (x_1+ x_4).$$

Пример 2. Выполнить минимизацию БФ (получить $f_{\text{МДНФ}}$ и $f_{\text{МКНФ}}$) заданной таблицей истинности (табл. 11.2).

Таблица 11.2

| № п/п | $x_1x_2x_3x_4$ | f |
|----------|----------------|-----|
| 1 | 2 | 3 |
| 0 | 0000 | 0 |
| 1 | 0001 | 1 |
| 2 | 0010 | 1 |
| 3 | 0011 | 0 |
| 4 | 0100 | 0 |
| 5 | 0101 | 0 |
| 6 | 0110 | 0 |
| 7 | 0111 | 0 |
| 8 | 1000 | 0 |
| 9 | 1001 | 1 |
| 10 | 1010 | 0 |
| 11 | 1011 | 1 |
| 12 | 1100 | 1 |
| 13 | 1101 | 0 |
| 14 | 1110 | 1 |
| 15 | 1111 | 1 |

По приведенной в таблице истинности функции заполним карту Вейча и выделим на ней контуры (рис. 11.4). Пунктиром обозначены те из них, которые покрывают общие наборы.

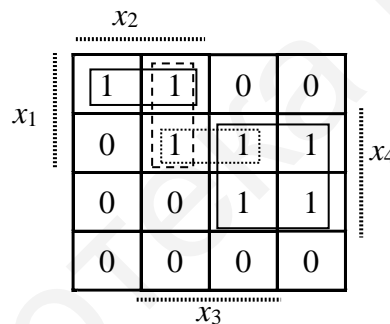


Рис. 11.4. Контуры на единичных наборах для примера 2

В результате получим две тупиковые формы БФ, одновременно и являющимися минимальными формами.

$$f^1_{\text{МДНФ}} = x_1x_2\bar{x}_4 + x_1x_3x_4 + \bar{x}_2x_4 - 1\text{-я тупиковая форма,}$$

$$f^2_{\text{МДНФ}} = x_1x_2\bar{x}_4 + x_1x_2x_3 + \bar{x}_2x_4 - 2\text{-я тупиковая форма.}$$

Пример 3. Выполнить минимизацию БФ (получить $f_{\text{МДНФ}}$ и $f_{\text{МКНФ}}$), заданной выражением

$$f_{\text{СДНФ}} = \vee 1, 2, 6, 9, 10, 11, 12, 14, 15.$$

В приведенном выражении показаны номера строк таблицы истинности, в которых значение БФ равно единице. Номера строк таблицы истинности начинаются с нуля. Построим карту Вейча для заданной БФ (рис. 11.5).

| | | | | | |
|-------|-------|-------|---|---|-------|
| | x_2 | | | | |
| x_1 | 1 | 1 | 1 | 0 | x_4 |
| | 0 | 1 | 1 | 1 | |
| | 0 | 0 | 0 | 1 | |
| | 0 | 1 | 1 | 0 | |
| | | x_3 | | | |

Рис. 11.5. Карта Вейча для ДНФ

Для выделенных на карте контуров, образованных на единичных наборах (см. рис. 11.5), запишем минимальную ДНФ:

$$f_{\text{МДНФ}} = x_1x_2\bar{x}_4 + x_3\bar{x}_4 + x_1x_3 + \bar{x}_2\bar{x}_3x_4.$$

| | | | | | |
|-------|-------|-------|---|---|-------|
| | x_2 | | | | |
| x_1 | 1 | 1 | 1 | 0 | x_4 |
| | 0 | 1 | 1 | 1 | |
| | 0 | 0 | 0 | 1 | |
| | 0 | 1 | 1 | 0 | |
| | | x_3 | | | |

Рис. 11.6. Карта Вейча для КНФ

При формировании контуров на нулевых наборах (рис. 11.6) получим минимальную КНФ:

$$f_{\text{МКНФ}} = x_2x_3x_4 + \bar{x}_2\bar{x}_3\bar{x}_4 + x_1\bar{x}_2x_3 + x_1\bar{x}_3\bar{x}_4.$$

Пример 4. Записать минимальную БФ, заданную картой Вейча (рис. 11.7).

| | | | | | | | | | |
|-------|-------|-------|---|-------|---|-------|---|---|-------|
| | x_2 | | | | | | | | |
| x_1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | x_4 |
| | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | |
| | | x_5 | | x_3 | | x_5 | | | |

| | | | | | | | | | |
|-------|-------|-------|---|-------|---|-------|---|---|-------|
| | x_2 | | | | | | | | |
| x_1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | x_4 |
| | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | |
| | | x_5 | | x_3 | | x_5 | | | |

Рис. 11.7. Карта Вейча для примера 4

$$f_{\text{МДНФ}} = x_1\bar{x}_4\bar{x}_5 + \bar{x}_2\bar{x}_3\bar{x}_4 + x_1\bar{x}_2\bar{x}_3 + x_1x_2x_4x_5,$$

$$f_{\text{МКНФ}} = (\bar{x}_2 + x_4 + \bar{x}_5)(\bar{x}_2 + \bar{x}_4 + x_5)(x_2 + \bar{x}_3 + \bar{x}_4)(x_2 + \bar{x}_3 + \bar{x}_5)(x_1 + \bar{x}_4)(x_1 + \bar{x}_2)(x_1 + \bar{x}_3).$$

Рассмотрим пример на минимизацию неполностью определенных БФ. В отличие от приведенных выше примеров такие БФ содержат наборы, на которых они не определены (не существуют). Таким наборам можно (для улучшения результата минимизации) приписывать либо единичные, либо нулевые значения.

Пример 5. Для неполностью определенной БФ (рис.11.8) получить минимальную ДНФ (КНФ).

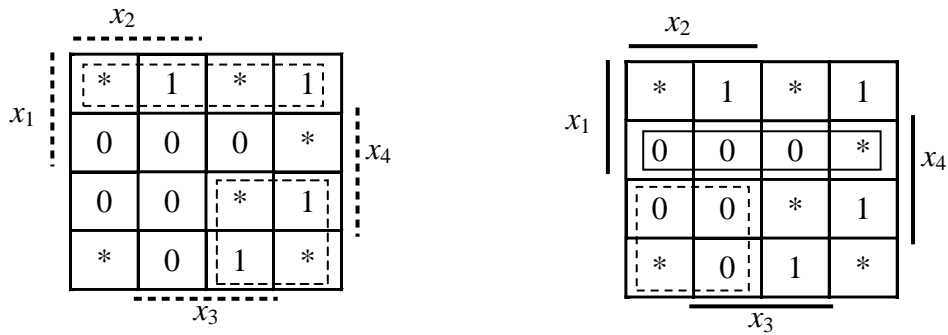


Рис. 11.8. Карта Вейча для примера 5

$$f_{\text{МДНФ}} = x_1\bar{x}_4 + \bar{x}_1\bar{x}_2,$$

$$f_{\text{МКНФ}} = (\bar{x}_1 + \bar{x}_4)(x_1 + \bar{x}_2).$$

Минимизация булевых функций с применением карт Карно

Отличие минимизирующих карт Карно от карт Вейча состоит в том, что в картах Карно строкам и столбцам ставятся в соответствие не буквы, составляющие БФ, а двоичные наборы – коды Грея [4, с. 98]. Таким образом, карта Карно наиболее эффективна по отношению к таблицам истинности. Рассматриваемые ниже примеры ориентированы на минимизацию неполностью определенных БФ, как наиболее общий случай БФ. Это же справедливо и для обычных БФ.

Пример 6. Выполнить минимизацию неполностью определенной БФ (получить $f_{\text{МДНФ}}$ и $f_{\text{МКНФ}}$), заданной таблицей истинности (табл. 11.3).

Таблица 11.3

| № п/п | $x_1x_2x_3x_4$ | f |
|----------|----------------|-----|
| 0 | 0000 | 0 |
| 1 | 0001 | 1 |
| 2 | 0010 | * |
| 3 | 0011 | 1 |
| 4 | 0100 | * |
| 5 | 0101 | 0 |
| 6 | 0110 | 0 |
| 7 | 0111 | 0 |
| 8 | 1000 | 0 |
| 9 | 1001 | 1 |
| 10 | 1010 | 0 |
| 11 | 1011 | 1 |
| 12 | 1100 | 1 |
| 13 | 1101 | * |
| 14 | 1110 | 1 |
| 15 | 1111 | 1 |

По приведенной в таблице истинности функции заполним карту Карно и выделим на ней контуры. На рис. 11.9 представлены две карты соответственно для получения минимальной БФ в форме ДНФ и КНФ.

| | | | | | |
|----------|----|----------|----|----|----|
| | | x_3x_4 | | | |
| | | 00 | 01 | 11 | 10 |
| x_1x_2 | 00 | 0 | 1 | 1 | * |
| | 01 | * | 0 | 0 | 0 |
| | 11 | 1 | * | 1 | 1 |
| | 10 | 1 | 1 | 1 | 0 |

| | | | | | |
|----------|----|----------|----|----|----|
| | | x_3x_4 | | | |
| | | 00 | 01 | 11 | 10 |
| x_1x_2 | 00 | 0 | 1 | 1 | * |
| | 01 | * | 0 | 0 | 0 |
| | 11 | 1 | * | 1 | 1 |
| | 10 | 0 | 1 | 1 | 0 |

Рис. 11.9. Карты Карно для примера 6

$$f_{\text{МДНФ}} = x_1x_2 + x_3x_4,$$

$$f_{\text{МКНФ}} = (x_1 + x_4)(x_1 + \bar{x}_2)(x_2 + \bar{x}_3 + x_4).$$

Пример 7. Выполнить минимизацию, используя карты Карно (рис. 11.10), и получить $f_{\text{МДНФ}}$ и $f_{\text{МКНФ}}$ для БФ, заданной выражением

$$f_{\text{СДНФ}} = \vee 0, 3, 8, 12, 14, 24, 17, 19, 26, 28 \text{ – единичные наборы.}$$

$$f^* = \vee 1, 5, 10, 13, 16, 18, 21, 29, 30 \text{ – наборы, на которых БФ не определена.}$$

| | | | | | | | | | |
|----------|----|-------------|-----|-----|-----|-----|-----|-----|-----|
| | | $x_3x_4x_5$ | | | | | | | |
| | | 000 | 001 | 011 | 010 | 110 | 111 | 101 | 100 |
| x_1x_2 | 00 | 1 | * | 1 | 0 | 0 | 0 | * | 0 |
| | 01 | 1 | 0 | 0 | * | 1 | 0 | * | 1 |
| | 11 | 1 | 0 | 0 | 1 | * | 0 | * | 1 |
| | 10 | * | 1 | 1 | * | 0 | 0 | * | 0 |

| | | | | | | | | | |
|----------|----|-------------|-----|-----|-----|-----|-----|-----|-----|
| | | $x_3x_4x_5$ | | | | | | | |
| | | 000 | 001 | 011 | 010 | 110 | 111 | 101 | 100 |
| x_1x_2 | 00 | 1 | * | 1 | 0 | 0 | 0 | * | 0 |
| | 01 | 1 | 0 | 0 | * | 1 | 0 | * | 1 |
| | 11 | 1 | 0 | 0 | 1 | * | 0 | * | 1 |
| | 10 | * | 1 | 1 | * | 0 | 0 | * | 0 |

Рис. 11.10. Карты Карно для примера 7

$$f_{\text{МДНФ}} = \bar{x}_2\bar{x}_3\bar{x}_4 + \bar{x}_2\bar{x}_3x_5 + x_2\bar{x}_5,$$

$$f_{\text{МКНФ}} = (x_2 + \bar{x}_4 + x_5)(x_2 + \bar{x}_3)(\bar{x}_2 + \bar{x}_5).$$

Практические задания

1. Выполнить минимизацию БФ по ее числовому заданию. В условии $f_{\text{СДНФ}}$ – единичные наборы, f^* – наборы, на которых БФ не определена.

а) $f_{\text{СДНФ}} = \vee 1, 3, 6, 8, 10, 12, 18, f^* = \vee 2, 11, 14, 15;$

б) $f_{\text{СДНФ}} = \vee 0, 2, 5, 7, 15, 18, 23, 24, 30, f^* = \vee 1, 3, 12, 14, 19;$

в) $f_{\text{СДНФ}} = \vee 1, 2, 4, 16, 17, 19, 22, 29, f^* = \vee 0, 3, 6, 14, 18;$

г) $f_{\text{СДНФ}} = \vee 0, 3, 6, 9, 13, 14, 22, f^* = \vee 1, 2, 5, 10, 23, 30;$

д) $f_{\text{СДНФ}} = \vee 1, 3, 6, 10, 11, 24, 26, 27, 31, f^* = \vee 5, 7, 12, 25, 30;$

е) $f_{\text{СДНФ}} = \vee 1, 4, 5, 7, 19, 24, 28, f^* = \vee 3, 6, 8, 12, 17;$

ж) $f_{\text{СДНФ}} = \vee 0, 2, 6, 7, 16, 20, 22, 30, f^* = \vee 1, 4, 12, 23, 25;$

- з) $f_{\text{СДНФ}} = \vee 0, 6, 9, 13, 16, 18, 24, f^* = \vee 1, 2, 8, 12, 21, 22;$
 и) $f_{\text{СДНФ}} = \vee 1, 7, 9, 12, 17, 20, 22, 28, f^* = \vee 2, 3, 5, 14, 30;$
 к) $f_{\text{СДНФ}} = \vee 0, 2, 3, 4, 5, 12, 16, 18, 24, f^* = \vee 1, 7, 8, 9, 21.$

2. Выполнить минимизацию БФ по ее заданию:

- 1) картой Вейча (записать минимальную БФ в ДНФ);
- 2) картой Вейча (записать минимальную БФ в КНФ);
- 3) картой Карно (записать минимальную БФ в ДНФ);
- 4) картой Карно (записать минимальную БФ в КНФ).

а)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | * | 1 | 0 | 0 | 1 | * | 0 |
| 1 | 1 | 0 | * | 1 | 1 | * | 1 |
| 1 | 0 | 1 | * | * | 1 | * | 1 |
| * | 1 | 1 | * | 0 | * | * | 0 |

б)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | * | 1 | 0 | 0 | 0 | * | 1 |
| * | 1 | * | * | 1 | 0 | * | 1 |
| 1 | 0 | 0 | 1 | * | 0 | * | 1 |
| * | 1 | 1 | * | 0 | 0 | * | 0 |

в)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | * | 1 | 0 | 1 | 1 | * | 0 |
| 0 | 0 | 0 | * | 1 | 1 | * | 1 |
| 1 | 1 | 1 | * | * | 1 | * | 1 |
| * | 0 | * | * | 0 | * | * | 0 |

г)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | * | 1 | 0 | 0 | 0 | * | 0 |
| * | 1 | * | * | 1 | 0 | * | 1 |
| 1 | 0 | 0 | 1 | * | 0 | 1 | 1 |
| * | 0 | 0 | * | 0 | 0 | * | 0 |

д)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | * | 1 | * | 1 | 1 | * | 0 |
| 0 | 0 | 0 | * | 1 | 1 | * | 1 |
| 0 | * | 1 | * | * | 1 | * | 0 |
| * | 0 | * | * | 0 | * | 0 | 0 |

е)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | * | 1 | 0 | 0 | 1 | * | 0 |
| 0 | 0 | 0 | * | 1 | 1 | * | 1 |
| 1 | * | 1 | * | * | 1 | * | 1 |
| * | 1 | 0 | * | 0 | 0 | * | 0 |

ж)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | * | 1 | 0 | 0 | 0 | * | 1 |
| * | 1 | * | * | 1 | 0 | * | 1 |
| 0 | * | 1 | 1 | * | 0 | * | 0 |
| * | 1 | 1 | * | 0 | 0 | * | 0 |

з)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | * | 1 | 0 | 0 | 0 | * | 1 |
| * | 1 | * | * | 1 | 0 | * | 1 |
| 0 | * | 1 | 1 | * | 0 | * | 0 |
| * | 1 | 1 | * | 0 | 0 | * | 0 |

и)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | * | 1 | 0 | 1 | 1 | * | 0 |
| 0 | 0 | 0 | * | 1 | 1 | * | 1 |
| 1 | 1 | 1 | * | * | 1 | * | 1 |
| * | 0 | * | * | 0 | * | * | 0 |

к)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | * | 1 | 0 | 0 | 0 | * | 1 |
| * | 1 | * | * | 1 | 0 | * | 1 |
| 0 | * | 1 | 1 | * | 0 | * | 0 |
| * | 1 | 1 | * | 0 | 0 | * | 0 |

Тема 12. Кубическое представление булевых функций. Алгоритм Квайна – Мак-Класки. Алгоритм извлечения (Рота)

Контрольные вопросы

1. Что такое кубическое покрытие булевой функции?
2. Что такое кодовое расстояние?
3. На использовании каких операций основан алгоритм Рота?
4. Что такое L -экстремаль?
5. В чем состоит алгоритм ветвления?

Метод Квайна – Мак-Класки

Рассмотрим пример минимизации логической функции методом Квайна – Мак-Класки. Пусть задана функция

$$f_{\text{СДНФ}}(x_1x_2x_3x_4) = \vee (0, 3, 5, 7, 8, 10, 12, 13, 14, 15).$$

Сформируем кубический комплекс K , состоящий из кубов нулевой размерности:

$$K = \{0000, 0011, 0101, 0111, 1000, 1010, 1100, 1101, 1110, 1111\}.$$

На первом этапе рассматриваемого метода все исходные n -кубы разбиваются на непересекающиеся подгруппы по количеству единиц в кубе. Так как склеиваться могут только те кубы, у которых кодовое расстояние равно единице, то имеет смысл попарное сравнение производить только с кубами из соседних групп.

Выполним разбиение комплекса K на группы, получим

$$K_0^0 = \{0000\}, K_1^0 = \{1000\}, K_2^0 = \begin{Bmatrix} 0011 \\ 0101 \\ 1010 \\ 1100 \end{Bmatrix}, K_3^0 = \begin{Bmatrix} 0111 \\ 1101 \\ 1110 \end{Bmatrix}, K_4^0 = \{1111\}.$$

Попарное сравнение проводим только между соседними по номеру группами кубов. В результате сравнения групп кубов K_0^0 и K_1^0 получим новую группу K_1^1 :
 K_0^0 и $K_1^0 \Rightarrow K_1^1 = \{x000\}$.

Аналогично анализируем кубы из других соседних групп:

$$K_1^0 \text{ и } K_2^0 \Rightarrow K_2^1 = \left\{ \begin{matrix} 10x0 \\ 1x00 \end{matrix} \right\}, K_2^0 \text{ и } K_3^0 \Rightarrow K_3^1 = \left\{ \begin{matrix} 0x11 \\ 01x1 \\ x101 \\ 1x10 \\ 11x0 \\ 110x \end{matrix} \right\}, K_3^0 \text{ и } K_4^0 \Rightarrow K_4^1 = \left\{ \begin{matrix} x111 \\ 11x1 \\ 111x \end{matrix} \right\}.$$

После выполнения первого этапа метода простых импликант не выявлено, т. к. все кубы участвовали в образовании новых кубов. Кубы второй размерности могут образоваться только при склеивании кубов, у которых свободная координата совпадает, следовательно, полученные кубы первой размерности разобьем на группы в зависимости от местоположения свободной координаты в кубе:

$$K_1^1 = \left\{ \begin{matrix} x000 \\ x101 \\ x111 \end{matrix} \right\}, K_2^1 = \left\{ \begin{matrix} 1x00 \\ 0x11 \\ 1x10 \end{matrix} \right\}, K_3^1 = \left\{ \begin{matrix} 10x0 \\ 01x1 \\ 11x0 \\ 11x1 \end{matrix} \right\}, K_4^1 = \left\{ \begin{matrix} 110x \\ 111x \end{matrix} \right\}.$$

Выполним сравнение и склеивание кубов внутри каждой из групп. В результате получим кубы второй размерности.

Из первой группы K_1^1 в результате склеивания получим куб второй размерности $x1x1$, а куб $x000$ является простой импликантой, т. к. не принял участия в склеивании кубов.

Из второй группы K_2^1 в результате склеивания получим куб второй размерности $1xx0$, а куб $0x11$ является простой импликантой.

Из третьей группы K_3^1 в результате склеивания получим кубы второй размерности $1xx0$, $x1x1$, $11xx$, простых импликант в этой группе нет, т. к. все кубы приняли участие в образовании кубов новой размерности.

Из четвертой группы K_4^1 в результате склеивания получим куб второй размерности $11xx$.

Среди вновь образованных кубов нет кубов, совпадающих по местоположению свободных координат, таким образом, формирование новых кубов больше не произойдет.

В результате получено множество простых импликант:

$$f_{\text{сокр. ДНФ}} = \{x000, 0x11, x1x1, 1xx0, 11xx, \}.$$

На втором этапе метода строится импликантная таблица (табл. 12.1). Формирование минимального покрытия сводится к выявлению обязательных простых импликант и построению на их основе тупиковых форм.

Таблица 12.1

| Простые импликанты | Минтермы | | | | | | | | | |
|-----------------------|----------|------|------|------|------|------|------|------|------|------|
| | 0000 | 0011 | 0101 | 0111 | 1000 | 1010 | 1100 | 1101 | 1110 | 1111 |
| $x000$ | * | | | | * | | | | | |
| $0x11$ | | * | | * | | | | | | |
| $x1x1$ | | | * | * | | | | * | | * |
| $1xx0$ | | | | | * | * | * | | * | |
| $11xx$ | | | | | | | * | * | * | * |

Из табл. 12.1 следует, что простые импликанты $x000$, $0x11$, $x1x1$ являются обязательными и образуют ядро функции, т. е. будут обязательно входить во все тупиковые ДНФ. Эти обязательные импликанты покроют кубы 0000, 1000, 0011, 0111, 0101, 1101, 1111. Оставшиеся две простые импликанты ($1xx0$ и $11xx$) не являются обязательными. Из исходного множества единичных кубов остались непокрытыми три набора (1010, 1100, 1110), и все они могут быть покрыты одной простой импликантой $1xx0$.

Следовательно, получена минимальная ДНФ функции:

$$f_{\text{МДНФ}} = \{x000, 0x11, x1x1, 1xx0\},$$

$$f_{\text{МДНФ}} = \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 x_3 x_4 \vee x_2 x_4 \vee x_1 \bar{x}_4.$$

Алгоритм Рота

Рассмотрим пример минимизации логической функции методом Рота. Исходное покрытие функции задано множествами кубов L (единичных) и N (безразличных):

$$L = \begin{pmatrix} 01100 \\ 01110 \\ 10001 \\ 11000 \\ 11011 \end{pmatrix} \quad N = \begin{pmatrix} 01000 \\ 10011 \\ 11001 \\ 11010 \end{pmatrix}$$

Алгоритм Рота реализуется в несколько этапов:

- нахождение множества Z простых импликант комплекса K ;
- выделение L -экстремалей на множестве Z ;
- применение алгоритма ветвления при отсутствии L -экстремалей;
- нахождение абсолютно минимального покрытия из некоторого множества избыточных покрытий.

Первый этап алгоритма Рота – нахождение множества простых импликант. Поиск простых импликант производится пошагово с использованием опе-

рации умножения кубов. Операция выполняется путем заполнения таблиц до тех пор, пока происходит образование новых кубов.

Сформируем исходное покрытие C_0 , заданное объединением множеств кубов L и N . Выполняется операция $(C_0 * C_0)$ (табл. 12.2).

Таблица 12.2

| $C_0 * C_0$ | 01100 | 01110 | 10001 | 11000 | 11011 | 01000 | 10011 | 11001 | 11010 |
|-------------|----------------|-------|----------------|-------------------------|-------------------------|-------|-------|-------|-------|
| 01100 | – | | | | | | | | |
| 01110 | 011x0 | – | | | | | | | |
| 10001 | | | – | | | | | | |
| 11000 | | | | – | | | | | |
| 11011 | | | | | – | | | | |
| 01000 | 01x00 | | | x1000 | | – | | | |
| 10011 | | | 100x1 | | 1x011 | | – | | |
| 11001 | | | 1x001 | 1100x | 110x1 | | | – | |
| 11010 | | | | 110x0 | 1101x | | | | – |
| A_1 | 011x0 01x00 | | 100x1 1x001 | x1000 1100x 110x0 | 1x011 110x1 1101x | | | | |

В результате операции формируется множество новых кубов A_1 , образовавшихся в результате склеивания кубов из исходного множества C_0 , а также кубов, которые не образуют новых кубов (включаются в множество Z_0):

$$A_1 = \begin{pmatrix} 011x0 \\ 01x00 \\ 100x1 \\ 1x001 \\ x1000 \\ 1100x \\ 110x0 \\ 1x011 \\ 110x1 \\ 1101x \end{pmatrix}, \quad Z_0 = \emptyset$$

На первом этапе в образовании новых кубов множества A_1 приняли участие все кубы исходного множества C_0 , поэтому нет кубов, которые следовало бы включить во множество Z_0 . Необходимо также сформировать множество $B_1 = C_0 - Z_0 = C_0$. Далее формируется множество $C_1 = A_1 \cup B_1$, но поскольку все кубы множества B_1 (C_0) уже покрыты кубами множества A_1 , то множество $C_1 = A_1$.

Следующий этап – выполнение операции $C_1 * C_1$ представлен в табл. 12.3.

Таблица 12.3

| $C_1 * C_1$ | 011x0 | 01x00 | 100x1 | 1x001 | x1000 | 1100x | 110x0 | 1x011 | 110x1 | 1101x |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 011x0 | — | | | | | | | | | |
| 01x00 | | — | | | | | | | | |
| 100x1 | | | — | | | | | | | |
| 1x001 | | | | — | | | | | | |
| x1000 | | | | | — | | | | | |
| 1100x | | | | | | — | | | | |
| 110x0 | | | | | | | — | | | |
| 1x011 | | | | 1x0x1 | | | | — | | |
| 110x1 | | | 1x0x1 | | | | 110xx | | — | |
| 1101x | | | | | | 110xx | | | | — |
| A_2 | | | 1x0x1 | 1x0x1 | | 110xx | 110xx | | | |

Из полученных новых кубов образуется множество A_2 , а из кубов, которые не участвовали в образовании новых, – множество Z_1 :

$$A_2 = \left\{ \begin{array}{l} 1x0x1 \\ 110xx \end{array} \right\}, \quad Z_1 = \left\{ \begin{array}{l} 011x0 \\ 01x00 \\ x1000 \end{array} \right\}.$$

Множество C_2 , как и на предыдущем этапе, формируется из кубов множества A_2 . Следующий этап – выполнение операции $C_2 * C_2$ представлен в табл. 12.4. Из табл. 12.4 следует, что $A_3 = \emptyset$. Таким образом, новых кубов при выполнении операции $C_2 * C_2$ не было получено:

$$Z_2 = \left\{ \begin{array}{l} 1x0x1 \\ 110xx \end{array} \right\}, \quad C_3 = A_3 = \emptyset.$$

Таблица 12.4

| $C_2 * C_2$ | 1x0x1 | 110xx |
|-------------|-------|-------|
| 110xx | — | |
| 110xx | | — |
| A_3 | | |

На этом процесс выявления простых импликант окончен. Таким образом, сформировано множество простых импликант Z :

$$Z = \bigcup_{i=0}^2 Z_i = Z_0 \cup Z_1 \cup Z_2 = \left\{ \begin{array}{l} 011x0 \\ 01x00 \\ x1000 \\ 1x0x1 \\ 110xx \end{array} \right\}.$$

Далее необходимо выяснить, не содержатся ли в этом множестве «лишние» простые импликанты. Для этого переходим к следующему этапу.

Второй этап алгоритма Рота – определение L -экстремалей. Для определения L -экстремалей выполняется операция вычитания (#) кубов, результат представлен в таблице (табл.12.5).

Таблица 12.5

| $z\#(Z - z)$ | 011x0 | 01x00 | x1000 | 1x0x1 | 110xx |
|--------------|----------------|----------------|----------------|----------------|-------------------------|
| 011x0 | – | zz0zz 01000 | 1zyzz x1000 | y0yzy 1x0x1 | yzyz1 110xx |
| 01x00 | zzz1z 01110 | – | 1zzzz 11000 | y0z1y 1x0x1 | yzz11 110xx |
| x1000 | zzyyz 01110 | zzzzz ∅ | – | z0z1y 1x0x1 | zzz11 1101x 110x1 |
| 1x0x1 | yzzyz 01110 | | zzzzy 11000 | – | zzzz0 11010 zzzzz |
| 110xx | yzyzz 01110 | | zzzzz ∅ | z0zzz 100x1 | – |
| Остаток | 01110 | ∅ | ∅ | 100x1 | 11010 |

Если после последовательного вычитания из некоторой простой импликанты всех остальных получаем в качестве остатка куб, содержащий единичный набор из множества L , то данная простая импликанта будет обязательной, или L -экстремалью. Проверим, принадлежат ли остатки множеству L , с помощью операции пересечения кубов (табл. 12.6). Из табл. 12.6 следует, что в остатках 01110 и 100x1 содержатся наборы из множества L единичных наборов функции. Это значит, что простые импликанты 011x0 и 1x0x1 являются L -экстремальями, а куб 11010 не пересекается с кубами комплекса L , и значит, соответствующая ему простая импликанта не является L -экстремалью.

Таблица 12.6

| $z\#(Z - z) \cap L$ | 01100 | 01110 | 10001 | 11000 | 11011 |
|---------------------|-------|-------|-------|-------|-------|
| 01110 | ∅ | 01110 | ∅ | ∅ | ∅ |
| 100x1 | ∅ | ∅ | 10001 | ∅ | ∅ |
| 11010 | ∅ | ∅ | ∅ | ∅ | ∅ |

Таким образом, из табл. 12.6 получено множество L -экстремалей:

$$E = \left\{ \begin{array}{l} 011x0 \\ 1x0x1 \end{array} \right\}.$$

Далее необходимо выяснить, какие из вершин комплекса L не покрываются L -экстремалиями. Для этого из каждого куба комплекса L вычтем ($\#$) элементы множества E (табл. 12.7). В результате вычитания получим $L_1 = L \# E$.

Таблица 12.7

| $L \# E$ | 01100 | 01110 | 10001 | 11000 | 11011 |
|----------|----------------------|----------------------|----------------------|----------------|----------------------|
| 011x0 | zzzzz \emptyset | zzzzz \emptyset | yyzyy 10001 | yzyzz 11000 | yzzyy 11011 |
| 1x0x1 | yzzyy 01100 | yzzyy 01110 | zzzzz \emptyset | zzzyy 11000 | zzzzz \emptyset |

Из табл. 12.7 видно, что не покрывается L -экстремалиями куб $L_1 = \{11000\}$. Этот куб необходимо покрыть какими-либо простыми импликантами, которые не стали L -экстремалиями:

$$\hat{Z} = Z \setminus E = \left\{ \begin{array}{l} 011x0 \\ 01x00 \\ x1000 \\ 1x0x1 \\ 110xx \end{array} \right\} \setminus \left\{ \begin{array}{l} 011x0 \\ 1x0x1 \end{array} \right\} = \left\{ \begin{array}{l} 01x00 \\ x1000 \\ 110xx \end{array} \right\}.$$

Теперь из полученного множества \hat{Z} надо выбрать куб с минимальной ценой (максимальной размерностью), чтобы покрыть набор $L_1 = \{11000\}$. Очевидно, что этот набор можно покрыть двумя простыми импликантами $x1000$ и $110xx$, причем второй куб ($110xx$) имеет большую размерность, а значит, именно его нужно включить в минимальное покрытие.

$$f_{\text{МДНФ}} = \{011x0, 1x0x1, 110xx\},$$

$$f_{\text{МДНФ}} = \bar{x}_1 x_2 x_3 \bar{x}_5 \vee x_1 \bar{x}_3 x_5 \vee x_1 x_2 \bar{x}_3.$$

Практические задания

1. Выполнить минимизацию булевой функции заданной таблицей истинности методом Квайна – Мак-Класки. Таблица истинности задана аналитически: цифры являются номерами строк таблицы истинности, в которых размещены наборы, на которых функция принимает истинное значение.

- $f_{\text{СДНФ}} = \vee 0, 1, 3, 4, 6, 8, 9, 12, 14, 15, 25, 31;$
- $f_{\text{СДНФ}} = \vee 2, 3, 5, 7, 8, 10, 12, 13, 15, 18, 19, 25, 27;$
- $f_{\text{СДНФ}} = \vee 0, 2, 4, 7, 8, 10, 12, 15, 18, 27, 31;$
- $f_{\text{СДНФ}} = \vee 0, 2, 4, 6, 7, 8, 10, 12, 14, 16, 17, 18, 23;$
- $f_{\text{СДНФ}} = \vee 7, 8, 9, 12, 13, 14, 15, 20, 21, 25, 28, 30, 31;$
- $f_{\text{СДНФ}} = \vee 0, 1, 3, 5, 6, 7, 8, 9, 14, 15, 16, 17, 21, 29, 31;$

ж) $f_{\text{СДНФ}} = \vee 0, 3, 4, 7, 8, 9, 11, 12, 13, 15, 24, 31$;

з) $f_{\text{СДНФ}} = \vee 0, 1, 2, 5, 7, 8, 9, 10, 13, 15, 17, 21, 29$;

и) $f_{\text{СДНФ}} = \vee 0, 2, 4, 6, 7, 10, 12, 14, 15, 17, 20, 21, 25$;

к) $f_{\text{СДНФ}} = \vee 2, 3, 4, 5, 7, 8, 12, 18, 21, 22, 23, 24, 28, 30, 31$.

2. Используя алгоритм Рота, получить минимальную форму булевой функции, исходное покрытие которой задано множествами кубов L (единичных) и N (безразличных):

$$\text{а) } L = \begin{pmatrix} 01100 \\ 01110 \\ 10001 \\ 11000 \\ 11011 \end{pmatrix}, \quad N = \begin{pmatrix} 01000 \\ 10011 \\ 11001 \\ 11010 \end{pmatrix}; \quad \text{б) } L = \begin{pmatrix} 0000 \\ 0010 \\ 0110 \\ 1101 \\ 1011 \\ 1001 \\ 1000 \end{pmatrix}, \quad N = \emptyset; \quad \text{в) } L = \begin{pmatrix} 01101 \\ 01000 \\ 10011 \\ 11001 \\ 11011 \end{pmatrix}, \quad N = \begin{pmatrix} 01001 \\ 10010 \\ 10001 \\ 11010 \end{pmatrix};$$

$$\text{г) } L = \begin{pmatrix} 0101 \\ 1101 \\ 0x10 \\ 1000 \\ 1010 \end{pmatrix}, \quad N = \begin{pmatrix} 0x00 \\ 0111 \\ 1011 \end{pmatrix}; \quad \text{д) } L = \begin{pmatrix} 0001 \\ 0100 \\ 1100 \\ 1x01 \\ x110 \\ 0010 \end{pmatrix}, \quad N = \emptyset; \quad \text{е) } L = \begin{pmatrix} 1001 \\ 010x \\ 0111 \\ 1110 \end{pmatrix}, \quad N = \begin{pmatrix} 00x0 \\ 1101 \\ 1011 \end{pmatrix};$$

$$\text{ж) } L = \begin{pmatrix} 00100 \\ 01000 \\ 01001 \\ 01100 \\ 01111 \\ 10000 \\ 10001 \end{pmatrix}, \quad N = \begin{pmatrix} 00000 \\ 01101 \\ 01110 \\ 10100 \end{pmatrix}; \quad \text{з) } L = \begin{pmatrix} 1000 \\ 010x \\ 0011 \\ 1110 \end{pmatrix}, \quad N = \begin{pmatrix} 0000 \\ 1101 \\ 10x1 \end{pmatrix}; \quad \text{и) } L = \begin{pmatrix} 0001 \\ 0010 \\ 0101 \\ 1000 \\ 1100 \\ 1101 \\ 1110 \end{pmatrix}, \quad N = \begin{pmatrix} 0000 \\ 1010 \\ 0111 \end{pmatrix};$$

$$\text{к) } L = \begin{pmatrix} 1000 \\ 0101 \\ 0011 \\ 11x0 \end{pmatrix}, \quad N = \begin{pmatrix} 0000 \\ 11x1 \\ 1001 \end{pmatrix}.$$

Тема 13. Абстрактные и структурные автоматы. Память автомата, триггеры. Канонический метод синтеза структурного автомата

Контрольные вопросы

1. Какой автомат называется абстрактным, структурным?
2. Какие существуют способы задания абстрактных автоматов?
3. В чем отличие автоматов Мили и Мура, способов их задания?
4. В чем состоит канонический метод синтеза автоматов по таблицам переходов, выходов?
5. Как определяется разрядность кода для кодирования множеств Z , W и A ?

Канонический метод структурного синтеза автомата Мили

Ниже приведен пример канонического метода синтеза структурного автомата Мили, заданного таблицами переходов и выходов, описывающих поведение абстрактного автомата. Более подробно данный метод синтеза рассмотрен в [4, с. 142–147]. Для синтеза блока памяти будем использовать, например, JK -триггеры. Исходные данные для выполнения синтеза структурной схемы приведены в таблице переходов (табл. 13.1) и таблице выходов (табл. 13.2).

Определяем вначале общее количество входов (L), выходов (N) и элементов памяти автомата (R):

$$L = \lceil \log_2 Z \rceil = \lceil \log_2 4 \rceil = 2,$$

$$N = \lceil \log_2 W \rceil = \lceil \log_2 5 \rceil = 3,$$

$$R = \lceil \log_2 |A| \rceil = \lceil \log_2 3 \rceil = 2.$$

Таблица 13.1

| δ | a_1 | a_2 | a_3 |
|----------|-------|-------|-------|
| z_1 | a_2 | a_2 | — |
| z_2 | a_3 | a_3 | a_2 |
| z_3 | a_3 | a_3 | a_3 |
| z_4 | — | a_1 | — |

Таблица 13.2

| λ | a_1 | a_2 | a_3 |
|-----------|-------|-------|-------|
| z_1 | w_1 | w_2 | — |
| z_2 | w_2 | w_2 | w_4 |
| z_3 | w_3 | w_5 | w_3 |
| z_4 | — | w_4 | — |

Структурная схема автомата, который должен быть построен в процессе выполнения канонического метода, изображена на рис. 13.1.

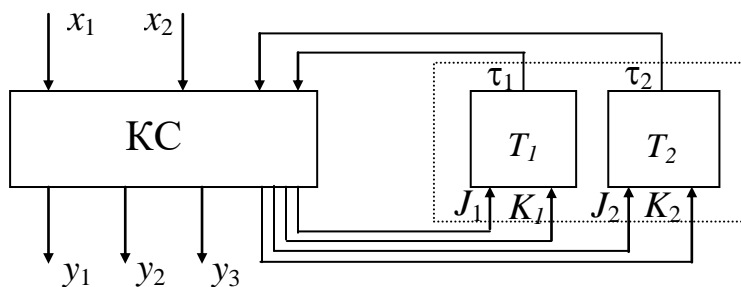


Рис. 13.1. Структурная схема автомата

На основании полученных значений L , N и R выполним кодирование входного, выходного алфавита и внутренних состояний автомата (табл. 13.3–13.5).

Таблица 13.3

| Z \ X | x_1 | x_2 |
|-------|-------|-------|
| z_1 | 0 | 0 |
| z_2 | 0 | 1 |
| z_3 | 1 | 0 |
| z_4 | 1 | 1 |

Таблица 13.4

| W \ Y | y_1 | y_2 | y_3 |
|-------|-------|-------|-------|
| w_1 | 0 | 0 | 0 |
| w_2 | 0 | 0 | 1 |
| w_3 | 0 | 1 | 0 |
| w_4 | 0 | 1 | 1 |
| w_5 | 1 | 0 | 0 |

Таблица 13.5

| A \ T | τ_1 | τ_2 |
|-------|----------|----------|
| a_1 | 0 | 0 |
| a_2 | 0 | 1 |
| a_3 | 1 | 0 |

По результатам кодирования строим таблицы переходов (табл. 13.6) и выходов (табл. 13.7) структурного автомата.

Таблица 13.6

| $\tau_1\tau_2$ | 00 | 01 | 10 |
|----------------|-------|-------|-------|
| x_1x_2 | a_1 | a_2 | a_3 |
| 00 z_1 | 01 | 01 | — |
| 01 z_2 | 10 | 10 | 01 |
| 10 z_3 | 10 | 10 | 10 |
| 11 z_4 | — | 00 | — |

Таблица 13.7

| $\tau_1\tau_2$ | 00 | 01 | 10 |
|----------------|--------------|--------------|--------------|
| x_1x_2 | a_1 | a_2 | a_3 |
| 00 z_1 | 000 w_1 | 001 w_2 | — |
| 01 z_2 | 001 w_2 | 001 w_2 | 011 w_4 |
| 10 z_3 | 010 w_3 | 100 w_5 | 010 w_3 |
| 11 z_4 | — | 011 w_4 | — |

$y_1y_2y_3$ $y_1y_2y_3$

Используя таблицу работы JK -триггера (табл. 13.8), внесем изменения в таблицу функций возбуждения элементов памяти (см. табл. 13.6), построенную для случая реализации блока памяти на основе D -триггера.

Таблица 13.8

| Q_t | JK | | | |
|-------|------|----|----|----|
| | 00 | 01 | 10 | 11 |
| 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 |

Для пояснения ниже показаны значения сигналов R и S , подаваемых на триггер для перевода его из старого состояния в новое:

$$\begin{aligned}
 0 \rightarrow 0 & - \bar{J}-, & 1 \rightarrow 0 & - -K, \\
 0 \rightarrow 1 & - J-, & 1 \rightarrow 1 & - -\bar{K}.
 \end{aligned}$$

Таблица 13.9

| $\tau_1\tau_2$ | 00 | 01 | 10 |
|----------------|-------|-------|-------|
| x_1x_2 | a_1 | a_2 | a_3 |
| 00 z_1 | 0-1- | 0--0 | - |
| 01 z_2 | 1-0- | 1--1 | -11- |
| 10 z_3 | 1-0- | 1--1 | -00- |
| 11 z_4 | - | 0--1 | - |

$\downarrow \downarrow \downarrow \downarrow$
 $J_1 K_1 J_2 K_2$

В результате этого будет получена новая таблица переходов (табл. 13.9).

На основании полученных табл. 13.7 и 13.9, которые можно рассматривать как таблицы истинности, может быть записана система булевых функций для построения комбинационной схемы автомата.

Далее для примера реализации комбинационной схемы будем рассматривать систему булевых функций, содержащую функции выходов y_1 и y_2 и функции возбуждения элементов памяти J_1 и K_1 . Для упрощения комбинационной схемы выполним минимизацию каждой из булевых функций.

Для этого используем метод минимизирующих карт Карно. Карты для данных функций приведены на рис. 13.2.

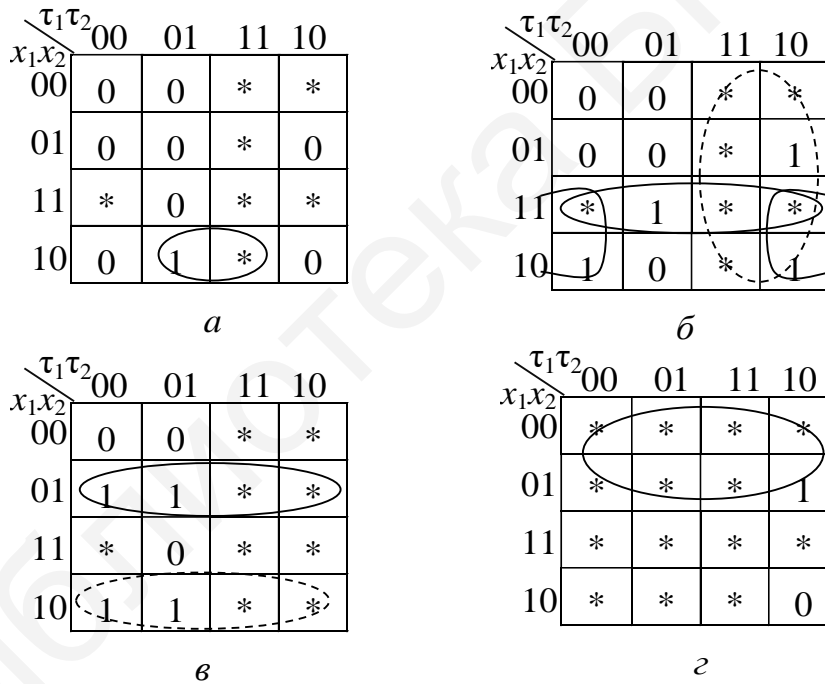


Рис. 13.2. Карты Карно для минимизации булевых функций:
 а – функция y_1 ; б – функция y_2 ; в – функция J_1 ; г – функция K_1

По результатам минимизации запишем систему минимальных функций:

$$y_1 = \tau_2 x_1 \bar{x}_2,$$

$$y_2 = \bar{\tau}_2 x_1 \vee x_1 x_2 \vee \tau_1,$$

$$J_1 = \bar{x}_1 x_2 \vee x_1 \bar{x}_2,$$

$$K_1 = \bar{x}_1.$$

На рис. 13.3 изображена логическая схема, построенная на основании полученной системы булевых функций (y_1, y_2, J_1 и K_1). Преобразования для остальных функции (y_3, J_2 и K_2) выполняются аналогично.

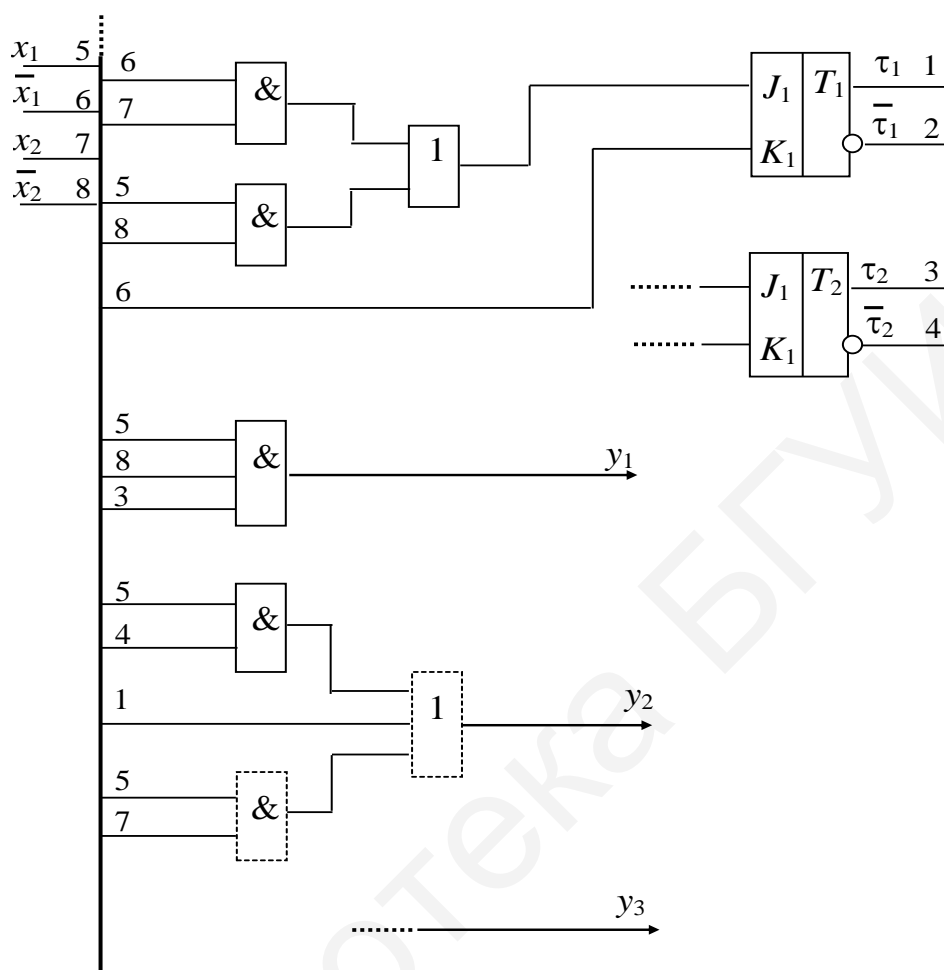


Рис. 13.3. Логическая схема автомата Мили

Канонический метод структурного синтеза автомата Мура

Далее рассмотрим канонический метод на примере синтеза структурного автомата Мура, блок памяти которого построим на RS -триггерах. Как и ранее, более подробно данный метод синтеза рассмотрен в [4, с. 147–150]. Исходные данные для выполнения синтеза структурной схемы заданы совмещенной таблицей переходов и выходов (табл. 13.10).

Таблица 13.10

| λ | w_1 | w_2 | w_4 | w_3 |
|-----------|-------|-------|-------|-------|
| δ | a_1 | a_2 | a_3 | a_4 |
| z_1 | a_2 | a_3 | a_4 | — |
| z_2 | a_3 | — | a_3 | a_2 |
| z_3 | — | a_2 | a_4 | a_1 |

Как и в предыдущем примере, вначале определим общее количество входов, выходов и элементов памяти структурного автомата: $L = 2, N = 2, R = 2$.

Структурная схема автомата изображена на рис. 13.4.

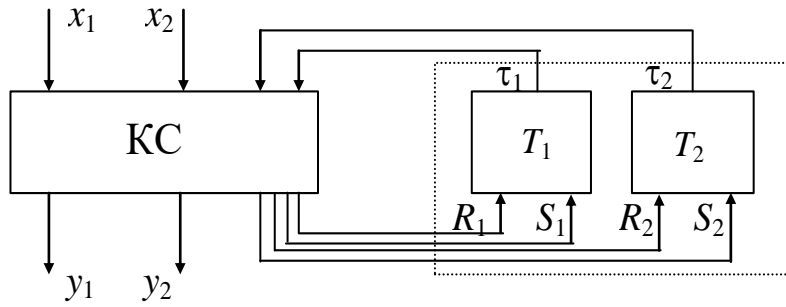


Рис. 13.4. Структурная схема автомата Мура

Выполним кодирование входного, выходного алфавита и внутренних состояний автомата (табл. 13.11–13.13).

Таблица 13.11

| $z \backslash X$ | x_1 | x_2 |
|------------------|-------|-------|
| z_1 | 0 | 0 |
| z_2 | 0 | 1 |
| z_3 | 1 | 0 |
| z_4 | 1 | 1 |

Таблица 13.12

| $w \backslash Y$ | y_1 | y_2 |
|------------------|-------|-------|
| w_1 | 0 | 0 |
| w_2 | 0 | 1 |
| w_3 | 1 | 0 |
| w_4 | 1 | 1 |

Таблица 13.13

| $A \backslash T$ | τ_1 | τ_2 |
|------------------|----------|----------|
| a_1 | 0 | 0 |
| a_2 | 0 | 1 |
| a_3 | 1 | 0 |
| a_4 | 1 | 1 |

Результаты кодирования занесем в исходную таблицу переходов – выходов (табл. 13.10), в результате получим табл. 13.14.

Внесем изменения в значения кодов состояний переходов в соответствии с принципом работы *RS*-триггера, приведенного в табл. 13.15.

Таблица 13.14

| $y_1 y_2$ | $y_1 y_2$ | $y_1 y_2$ | $y_1 y_2$ | $y_1 y_2$ |
|-----------------|-----------|-----------|-----------|-----------|
| $y_1 y_2$ | 00 | 01 | 11 | 10 |
| | w_1 | w_2 | w_4 | w_3 |
| $\tau_1 \tau_2$ | 00 | 01 | 10 | 11 |
| $x_1 x_2$ | a_1 | a_2 | a_3 | a_4 |
| 00 | 01 | 10 | 11 | — |
| z_1 | | | | |
| 01 | 10 | — | 10 | 01 |
| z_2 | | | | |
| 10 | — | 01 | 11 | 00 |
| z_3 | | | | |

Таблица 13.15

| Q_t | <i>RS</i> | | | |
|-------|-----------|----|----|----|
| | 00 | 01 | 10 | 11 |
| 0 | 0 | 1 | 0 | X |
| 1 | 1 | 1 | 0 | X |

Как и ранее, для пояснения ниже показаны значения сигналов *R* и *S*, подаваемых на триггер для перевода его из старого состояния в новое:

$$\begin{aligned}
 0 \rightarrow 0 & - \overline{\overline{S}}, & 1 \rightarrow 0 & - \overline{R} \overline{S}, \\
 0 \rightarrow 1 & - \overline{R} S, & 1 \rightarrow 1 & - \overline{R} \overline{S}.
 \end{aligned}$$

Используя приведенные выше пояснения работы *RS*-триггера, построим новую, перекодированную, таблицу переходов (табл. 13.16).

На основании табл. 13.16 может быть записана система булевых функций для построения комбинационной схемы автомата:

Таблица 13.16

| | | | | |
|-----------------|-------|-----------|-------|-----------|
| | | $y_1 y_2$ | | $y_1 y_2$ |
| $y_1 y_2$ | 00 | 01 | 11 | 10 |
| | w_1 | w_2 | w_4 | w_3 |
| $\tau_1 \tau_2$ | 00 | 01 | 10 | 11 |
| $x_1 x_2$ | a_1 | a_2 | a_3 | a_4 |
| 00 | -001 | 0110 | 0-01 | - |
| z_1 | | | | |
| 01 | 01-0 | - | 0--0 | 100- |
| z_2 | | | | |
| 10 | - | -00- | 0-01 | 1010 |
| z_3 | | | | |

$R_1 S_1 R_2 S_2$

$$R_1 = \tau_1 \tau_2 \bar{x}_1 x_2 \vee \tau_1 \tau_2 x_1 \bar{x}_2,$$

$$S_1 = \bar{\tau}_1 \bar{\tau}_2 \bar{x}_1 x_2 \vee \bar{\tau}_1 \bar{\tau}_2 x_1 \bar{x}_2,$$

$$R_2 = \bar{\tau}_1 \tau_2 \bar{x}_1 x_2 \vee \tau_1 \tau_2 x_1 \bar{x}_2,$$

$$S_2 = \bar{\tau}_1 \bar{\tau}_2 x_1 x_2 \vee \tau_1 \bar{\tau}_2 x_1 \bar{x}_2 \vee \tau_1 \bar{\tau}_2 \bar{x}_1 x_2,$$

$$y_1 = \bar{\tau}_1 \bar{\tau}_2 \vee \tau_1 \tau_2,$$

$$y_2 = \bar{\tau}_1 \tau_2 \vee \tau_1 \bar{\tau}_2.$$

Для упрощения комбинационной схемы выполним минимизацию каждой из булевых функций. Для этого используем метод минимизирующих карт Карно. На рис. 13.5 изображены четыре карты Карно для минимизации булевых функций R_1 , S_1 , R_2 и S_2 . Как и в примере синтеза автомата Мили, в карты Карно внесены символы «*» в клетки, соответствующие наборам входных значений, на которых функции не определены.

| | | | | |
|-----------------|----|----|----|----|
| $\tau_1 \tau_2$ | 00 | 01 | 11 | 10 |
| $x_1 x_2$ | 00 | * | 0 | * |
| | 01 | 0 | * | 1 |
| | 11 | * | * | * |
| | 10 | * | * | 1 |

a

| | | | | |
|-----------------|----|----|----|----|
| $\tau_1 \tau_2$ | 00 | 01 | 11 | 10 |
| $x_1 x_2$ | 00 | 0 | 1 | * |
| | 01 | 1 | * | 0 |
| | 11 | * | * | * |
| | 10 | * | 0 | 0 |

б

| | | | | |
|-----------------|----|----|----|----|
| $\tau_1 \tau_2$ | 00 | 01 | 11 | 10 |
| $x_1 x_2$ | 00 | 1 | 0 | * |
| | 01 | 0 | * | * |
| | 11 | * | * | * |
| | 10 | * | * | 0 |

в

| | | | | |
|-----------------|----|----|----|----|
| $\tau_1 \tau_2$ | 00 | 01 | 11 | 10 |
| $x_1 x_2$ | 00 | 0 | 1 | * |
| | 01 | * | * | 0 |
| | 11 | * | * | * |
| | 10 | * | 0 | 1 |

г

Рис. 13.5. Карты Карно для минимизации булевых функций: *a* – функция R_1 ; *б* – функция S_1 ; *в* – функция S_2 ; *г* – функция R_2

Для получения минимальной ДНФ функций y_1 и y_2 в этом примере не требуется использовать карты Карно.

По результатам минимизации запишем систему минимальных функций:

$$R_1 = \tau_2 x_1 \vee \tau_1 \tau_2 x_2,$$

$$S_1 = \bar{\tau}_1 x_2 \vee \bar{\tau}_1 \tau_2 \bar{x}_1,$$

$$R_2 = \tau_1 \tau_2 x_1 \vee \tau_1 \tau_2 x_1,$$

$$S_2 = \tau_2 \bar{x}_2,$$

$$y_1 = \tau_1,$$

$$y_2 = \bar{\tau}_1 \tau_2 \vee \tau_1 \tau_2.$$

На рис. 13.6 изображена логическая схема, построенная на основании полученной системы булевых функций. При построении схемы использованы элементы «И» и «ИЛИ».

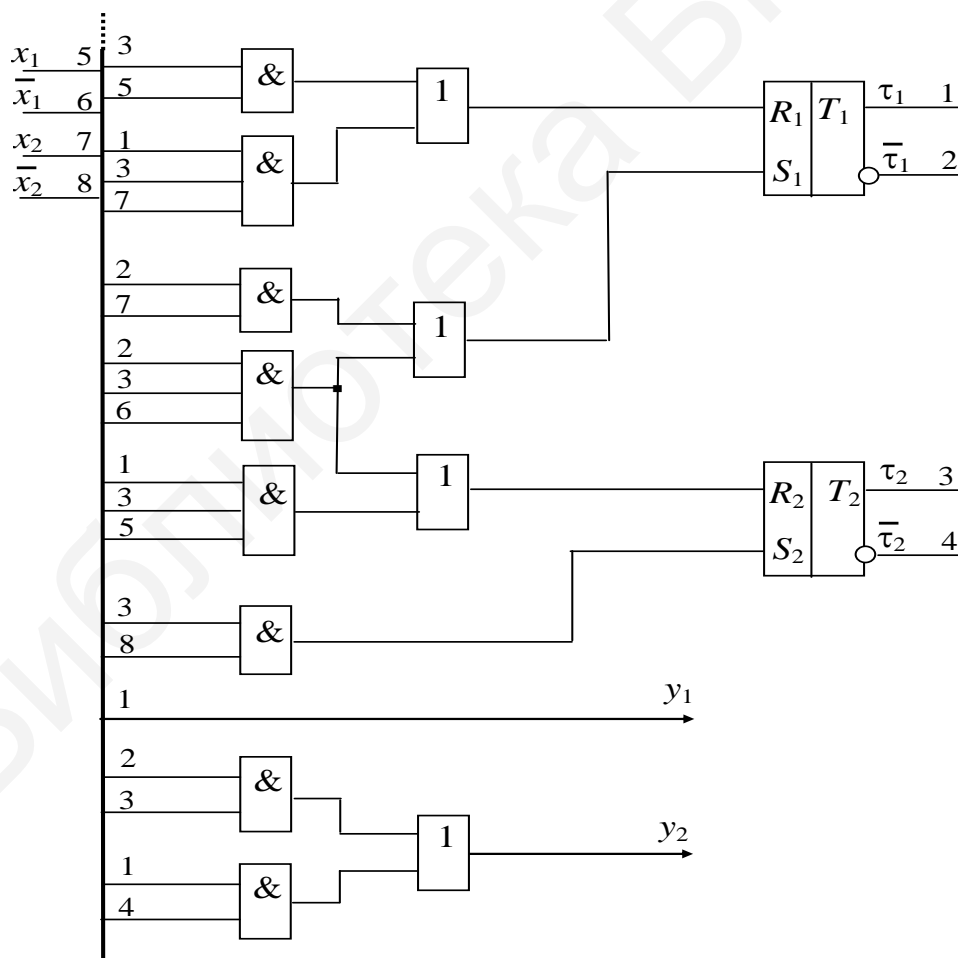


Рис. 13.6. Логическая схема автомата Мура

Практические задания

При выполнении приведенных ниже заданий блок памяти автомата реализовать на следующих триггерах:

- а) *D*-триггерах; б) *T*-триггерах;
в) *RS*-триггерах; г) *JK*-триггерах.

По результату выполненного синтеза построить логическую схему в базисе. В качестве элементного базиса использовать:

- 1) И, ИЛИ, НЕ; 2) И, НЕ; 3) ИЛИ, НЕ; 4) И-НЕ; 5) ИЛИ-НЕ.

1. Выполнить синтез структурного автомата Мили по таблицам переходов и выходов, изображенных ниже.

а)

| | | | |
|----------|-------|-------|-------|
| δ | a_1 | a_2 | a_3 |
| z_1 | — | a_2 | a_2 |
| z_2 | a_2 | a_3 | a_3 |
| z_3 | — | a_1 | — |

| | | | |
|-----------|-------|-------|-------|
| λ | a_1 | a_2 | a_3 |
| z_1 | — | w_3 | w_1 |
| z_2 | w_1 | w_2 | w_4 |
| z_3 | — | w_3 | — |

 б)

| | | | |
|----------|-------|-------|-------|
| δ | a_1 | a_2 | a_3 |
| z_1 | a_2 | a_2 | — |
| z_2 | a_1 | a_3 | a_3 |
| z_3 | — | a_2 | — |

| | | | |
|-----------|-------|-------|-------|
| λ | a_1 | a_2 | a_3 |
| z_1 | w_4 | w_3 | — |
| z_2 | w_1 | w_2 | w_3 |
| z_3 | — | w_3 | — |

в)

| | | | |
|----------|-------|-------|-------|
| δ | a_1 | a_2 | a_3 |
| z_1 | a_1 | a_2 | a_1 |
| z_2 | a_2 | a_3 | a_3 |
| z_3 | — | — | — |
| z_4 | — | a_2 | a_2 |

| | | | |
|-----------|-------|-------|-------|
| λ | a_1 | a_2 | a_3 |
| z_1 | w_5 | w_3 | w_1 |
| z_2 | w_1 | w_2 | w_4 |
| z_3 | — | — | — |
| z_4 | — | w_3 | w_2 |

 г)

| | | | |
|----------|-------|-------|-------|
| δ | a_1 | a_2 | a_3 |
| z_1 | a_2 | a_2 | — |
| z_2 | a_3 | a_3 | a_3 |
| z_3 | — | a_1 | a_2 |

| | | | |
|-----------|-------|-------|-------|
| λ | a_1 | a_2 | a_3 |
| z_1 | w_4 | w_3 | — |
| z_2 | w_1 | w_2 | w_4 |
| z_3 | — | w_3 | w_2 |

д)

| | | | | |
|----------|-------|-------|-------|-------|
| δ | a_1 | a_2 | a_3 | a_4 |
| z_1 | a_2 | a_2 | — | — |
| z_2 | a_3 | — | a_3 | a_4 |
| z_3 | — | a_1 | a_4 | a_2 |

| | | | | |
|-----------|-------|-------|-------|-------|
| λ | a_1 | a_2 | a_3 | a_4 |
| z_1 | w_4 | w_3 | — | — |
| z_2 | w_1 | — | w_3 | w_3 |
| z_3 | — | w_3 | w_4 | w_2 |

 е)

| | | | |
|----------|-------|-------|-------|
| δ | a_1 | a_2 | a_3 |
| z_1 | a_2 | a_2 | — |
| z_2 | a_3 | a_1 | a_3 |
| z_3 | — | a_2 | — |
| z_4 | a_2 | a_1 | — |

| | | | |
|-----------|-------|-------|-------|
| λ | a_1 | a_2 | a_3 |
| z_1 | w_4 | w_3 | — |
| z_2 | w_1 | w_2 | w_3 |
| z_3 | — | w_3 | — |
| z_4 | w_2 | w_4 | — |

ж)

| | | | | |
|----------|-------|-------|-------|-------|
| δ | a_1 | a_2 | a_3 | a_4 |
| z_1 | a_2 | a_2 | — | — |
| z_2 | a_3 | a_3 | a_3 | a_1 |
| z_3 | — | a_1 | a_4 | a_1 |
| z_4 | — | a_1 | a_3 | a_4 |

| | | | | |
|-----------|-------|-------|-------|-------|
| λ | a_1 | a_2 | a_3 | a_4 |
| z_1 | w_4 | w_3 | — | — |
| z_2 | w_1 | w_2 | w_3 | w_4 |
| z_3 | — | w_3 | w_4 | w_2 |
| z_4 | — | w_1 | w_4 | w_1 |

 з)

| | | | |
|----------|-------|-------|-------|
| δ | a_1 | a_2 | a_3 |
| z_1 | a_1 | a_2 | a_3 |
| z_2 | a_2 | a_3 | a_3 |
| z_3 | a_3 | a_1 | a_2 |

| | | | |
|-----------|-------|-------|-------|
| λ | a_1 | a_2 | a_3 |
| z_1 | w_4 | w_3 | w_1 |
| z_2 | w_2 | w_4 | w_2 |
| z_3 | w_3 | w_3 | w_2 |

и)

| | | | |
|----------|-------|-------|-------|
| δ | a_1 | a_2 | a_3 |
| z_1 | a_2 | a_2 | — |
| z_2 | a_3 | a_3 | a_3 |
| z_3 | — | a_1 | a_2 |

| | | | |
|-----------|-------|-------|-------|
| λ | a_1 | a_2 | a_3 |
| z_1 | w_4 | w_3 | — |
| z_2 | w_1 | w_2 | w_4 |
| z_3 | — | w_3 | w_2 |

 к)

| | | | | |
|----------|-------|-------|-------|-------|
| δ | a_1 | a_2 | a_3 | a_4 |
| z_1 | a_2 | a_4 | a_1 | a_2 |
| z_2 | a_3 | a_3 | a_3 | a_4 |
| z_3 | a_3 | a_1 | a_2 | a_3 |

| | | | | |
|-----------|-------|-------|-------|-------|
| λ | a_1 | a_2 | a_3 | a_4 |
| z_1 | w_2 | w_3 | w_3 | w_4 |
| z_2 | w_1 | w_2 | w_3 | w_5 |
| z_3 | w_3 | w_4 | w_5 | w_1 |

2. Выполнить синтез структурного автомата Мура по совмещенным таблицам переходов и выходов, изображенных ниже.

а)

| | | | |
|-----------|-------|-------|-------|
| λ | w_1 | w_2 | w_3 |
| δ | a_1 | a_2 | a_3 |
| z_1 | — | a_3 | a_2 |
| z_2 | a_2 | a_3 | a_2 |
| z_3 | — | a_1 | — |

б)

| | | | |
|-----------|-------|-------|-------|
| λ | w_1 | w_2 | w_3 |
| δ | a_1 | a_2 | a_3 |
| z_1 | a_2 | a_2 | — |
| z_2 | a_1 | a_3 | a_3 |
| z_3 | — | a_2 | — |
| z_4 | — | a_1 | a_2 |

в)

| | | | | | |
|-----------|-------|-------|-------|-------|-------|
| λ | w_1 | w_2 | w_3 | w_4 | w_5 |
| δ | a_1 | a_2 | a_3 | a_4 | a_5 |
| z_1 | a_2 | a_4 | a_2 | a_2 | — |
| z_2 | a_4 | a_3 | a_4 | a_5 | a_3 |
| z_3 | — | a_2 | a_1 | — | — |
| z_4 | — | a_1 | — | a_1 | a_5 |

г)

| | | | |
|-----------|-------|-------|-------|
| λ | w_1 | w_2 | w_3 |
| δ | a_1 | a_2 | a_3 |
| z_1 | a_2 | a_2 | — |
| z_2 | a_1 | a_3 | a_3 |
| z_3 | — | a_2 | — |
| z_4 | — | a_1 | a_2 |

д)

| | | | | | |
|-----------|-------|-------|-------|-------|-------|
| λ | w_1 | w_2 | w_3 | w_4 | w_5 |
| δ | a_1 | a_2 | a_3 | a_4 | a_5 |
| z_1 | a_2 | a_4 | a_1 | a_1 | — |
| z_2 | — | a_2 | a_4 | a_3 | a_4 |
| z_3 | a_3 | a_2 | — | — | — |
| z_4 | — | a_1 | — | a_2 | a_5 |

е)

| | | | | |
|-----------|-------|-------|-------|-------|
| λ | w_1 | w_2 | w_3 | w_4 |
| δ | a_1 | a_2 | a_3 | a_4 |
| z_1 | a_2 | a_2 | a_4 | — |
| z_2 | a_1 | a_3 | — | a_3 |
| z_3 | — | a_2 | a_4 | — |
| z_4 | — | a_1 | a_3 | a_2 |

ж)

| | | | | |
|-----------|-------|-------|-------|-------|
| λ | w_1 | w_3 | w_2 | w_3 |
| δ | a_1 | a_2 | a_3 | a_4 |
| z_1 | a_2 | a_2 | a_4 | a_3 |
| z_2 | — | — | — | a_4 |
| z_3 | a_4 | — | a_4 | — |
| z_4 | — | a_3 | a_2 | a_1 |

з)

| | | | | |
|-----------|-------|-------|-------|-------|
| λ | w_1 | w_2 | w_3 | w_4 |
| δ | a_2 | a_2 | a_2 | a_4 |
| z_1 | — | a_1 | a_4 | — |
| z_2 | a_1 | a_3 | a_2 | a_3 |
| z_3 | — | — | — | — |
| z_4 | a_4 | a_1 | a_1 | a_3 |

и)

| | | | | |
|-----------|-------|-------|-------|-------|
| λ | w_1 | w_3 | w_2 | w_3 |
| δ | a_1 | a_2 | a_3 | a_4 |
| z_1 | a_2 | a_2 | a_4 | a_3 |
| z_2 | — | — | — | a_4 |
| z_3 | a_4 | — | a_4 | — |
| z_4 | — | a_3 | a_2 | a_1 |

к)

| | | | | | |
|-----------|-------|-------|-------|-------|-------|
| λ | w_1 | w_2 | w_3 | w_4 | w_5 |
| δ | a_1 | a_2 | a_3 | a_4 | a_5 |
| z_1 | a_2 | a_4 | a_1 | a_1 | — |
| z_2 | — | a_2 | a_4 | a_3 | a_4 |
| z_3 | a_3 | a_2 | — | — | — |
| z_4 | — | a_1 | — | a_2 | a_5 |

**Тема 14. Синтез структурного автомата по граф-схеме алгоритма.
Гонки, риск сбоя**

Контрольные вопросы

1. Что такое абстрактный (структурный) автомат?
2. Какие существуют способы задания абстрактного автомата?
3. Назовите этапы синтеза структурного автомата по граф-схеме алгоритма (ГСА).
4. Что такое состояние автомата?
5. В чем состоит алгоритм разметки ГСА метками Мура (Мили)?
6. Что представляет собой прямая (обратная) таблица переходов, структурная таблица?
7. Что такое гонки и риск сбоя?

Пример синтеза МПА Мили по ГСА

Более подробно метод синтеза МПА Мили (Мура), рассмотрен в [4, с. 153–159]. Ниже приведен пример выполнения синтеза МПА Мили по ГСА, изображенной на рис. 14.1.

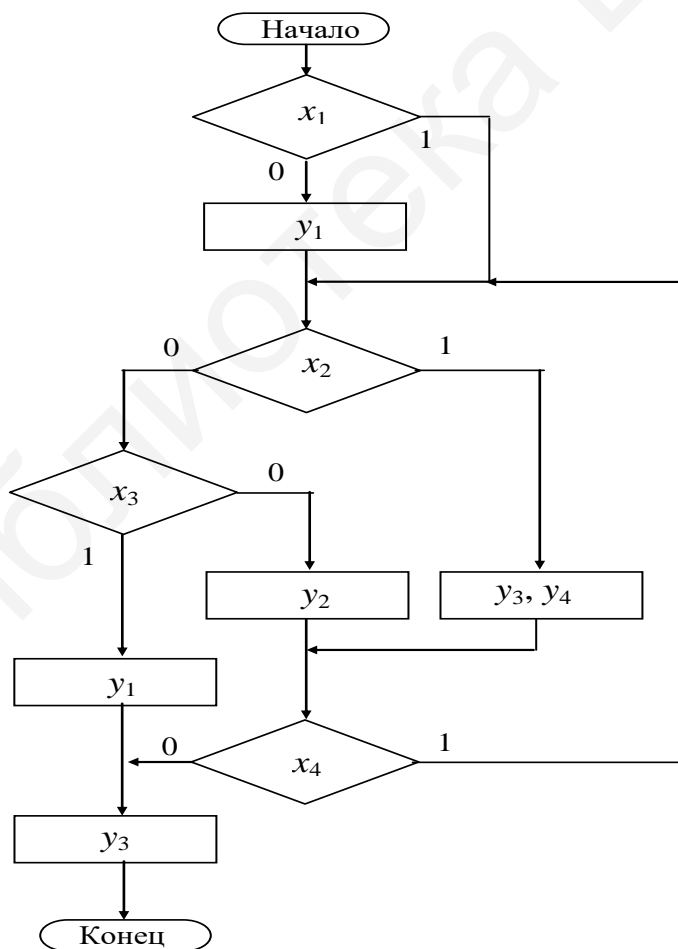


Рис. 14.1. ГСА для синтеза МПА Мили

Первым этапом синтеза МПА по ГСА является разметка ГСА метками Мили. Алгоритм разметки ГСА метками Мили приведен в [4, с. 156]. На рис. 14.2 изображена ГСА, отмеченная метками.

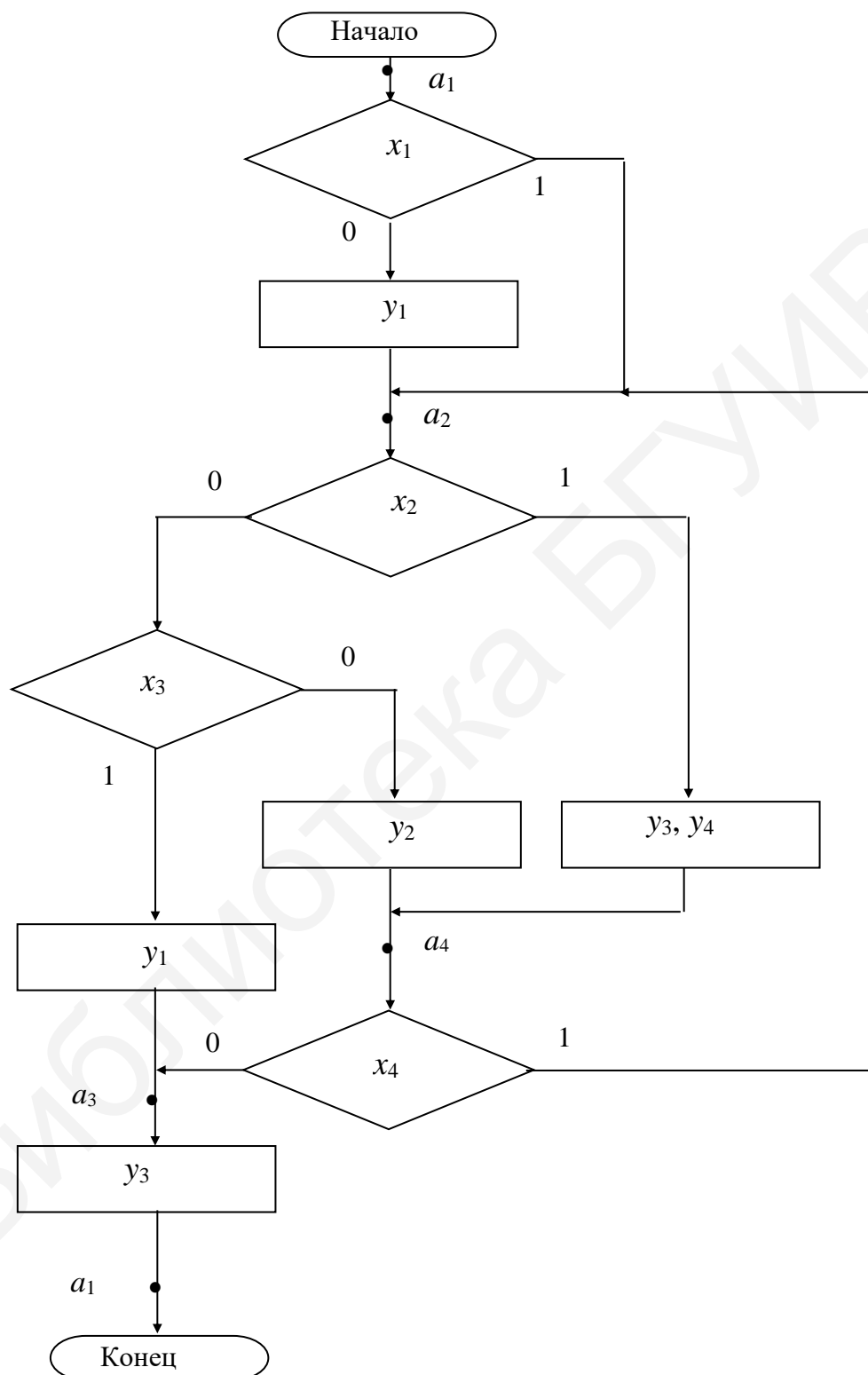


Рис. 14.2. Отмеченная ГСА при синтезе МПА Мили

По размеченной ГСА строится структурная таблица (прямая или обратная). В *прямой* таблице вначале записываются все переходы из состояния a_1 , затем из состояния a_2 и т. д. В *обратной* таблице сначала записываются все переходы в состояние a_1 , затем в a_2 и т. д.

Для реализации блока памяти синтезируемого МПА Мили использованы T -триггеры. В последнем столбце $F(a_m, a_s)$ структурной таблицы отмечены функции возбуждения, которые приводят к изменению содержимого каждого из элементов памяти на соответствующем переходе. В рассматриваемом примере построена прямая структурная таблица (табл. 14.1).

Таблица 14.1

| Исходное состояние | Код исходного состояния | Состояние перехода | Код состояния перехода | Входной сигнал | Выходной сигнал | Функции возбуждения |
|--------------------|-------------------------|--------------------|------------------------|---------------------------|-----------------|-----------------------|
| a_m | $K(a_m)$ | a_s | $K(a_s)$ | $X(a_m, a_s)$ | $Y(a_m, a_s)$ | $F(a_m, a_s)$ |
| a_1 | 00 | a_2 | 01 | \bar{x}_1 | y_1 | $\bar{T}_1 T_2$ |
| | | a_3 | 10 | $x_1 \bar{x}_2 x_3$ | y_1 | $T_1 \bar{T}_2$ |
| | | a_4 | 11 | $x_1 \bar{x}_2 \bar{x}_3$ | y_2 | $T_1 T_2$ |
| | | a_4 | 11 | $x_1 x_2$ | $y_3 y_4$ | $T_1 T_2$ |
| a_2 | 01 | a_3 | 10 | $\bar{x}_2 x_3$ | y_1 | $T_1 T_2$ |
| | | a_4 | 11 | $\bar{x}_2 \bar{x}_3$ | y_2 | $T_1 \bar{T}_2$ |
| | | a_4 | 11 | x_2 | $y_3 y_4$ | $T_1 \bar{T}_2$ |
| a_3 | 10 | a_1 | 00 | 1 | y_3 | $T_1 \bar{T}_2$ |
| a_4 | 11 | a_1 | 00 | \bar{x}_4 | y_3 | $T_1 T_2$ |
| | | a_4 | 11 | $x_4 x_2$ | $y_3 y_4$ | $\bar{T}_1 \bar{T}_2$ |
| | | a_4 | 11 | $x_4 \bar{x}_2 \bar{x}_3$ | y_2 | $\bar{T}_1 \bar{T}_2$ |
| | | a_3 | 10 | $x_4 \bar{x}_2 x_3$ | y_1 | $\bar{T}_1 T_2$ |

Синтез МПА может быть выполнен и по обратной структурной таблице (табл. 14.2).

Таблица 14.2

| Исходное состояние | Код исходного состояния | Состояние перехода | Код состояния перехода | Входной сигнал | Выходной сигнал | Функции возбуждения |
|--------------------|-------------------------|--------------------|------------------------|---------------------------|-----------------|-----------------------|
| a_m | $K(a_m)$ | a_s | $K(a_s)$ | $X(a_m, a_s)$ | $Y(a_m, a_s)$ | $F(a_m, a_s)$ |
| a_3 | 10 | a_1 | 00 | 1 | y_3 | $T_1 \bar{T}_2$ |
| a_4 | 11 | | | \bar{x}_4 | y_3 | $T_1 T_2$ |
| a_1 | 00 | a_2 | 01 | \bar{x}_1 | y_1 | $\bar{T}_1 T_2$ |
| a_1 | 00 | a_3 | 10 | $x_1 \bar{x}_2 x_3$ | y_1 | $T_1 \bar{T}_2$ |
| a_2 | 01 | | | $\bar{x}_2 x_3$ | y_1 | $T_1 T_2$ |
| a_4 | 11 | | | $x_4 \bar{x}_2 x_3$ | y_1 | $\bar{T}_1 T_2$ |
| a_1 | 00 | a_4 | 11 | $x_1 x_2$ | $y_3 y_4$ | $T_1 T_2$ |
| a_1 | 00 | | | $x_1 \bar{x}_2 \bar{x}_3$ | y_2 | $T_1 T_2$ |
| a_2 | 01 | | | x_2 | $y_3 y_4$ | $T_1 \bar{T}_2$ |
| a_2 | 01 | | | $\bar{x}_2 \bar{x}_3$ | y_2 | $T_1 \bar{T}_2$ |
| a_4 | 11 | | | x_2 | $y_3 y_4$ | $\bar{T}_1 \bar{T}_2$ |
| a_4 | 11 | | | $\bar{x}_2 \bar{x}_3$ | y_2 | $\bar{T}_1 \bar{T}_2$ |

Информация в столбце $F(a_m, a_s)$ структурной таблицы получена с помощью таблицы работы T -триггера [4] (табл. 14.3).

Таблица 14.3

| Q_t | T | |
|-------|-----|---|
| | 0 | 1 |
| 0 | 0 | 1 |
| 1 | 1 | 0 |

В рассматриваемом примере синтеза МПА воспользуемся прямой таблицей. Для построения схемы синтезируемого МПА Мили удобно результаты, приведенные в структурной таблице (см. табл. 14.2), представить в виде таблицы истинности (табл. 14.4).

Таблица 14.4

| № п/п | $x_1x_2x_3x_4$ | $\tau_1\tau_2$ | $y_1 y_2 y_3 y_4$ | $T_1 T_2$ |
|-------|----------------|----------------|-------------------|-----------|
| 0 | 0--- | 00 | 1 0 0 0 | 0 1 |
| 1 | 1 0 1 - | 00 | 1 0 0 0 | 1 0 |
| 2 | 1 0 0 - | 00 | 0 1 0 0 | 1 1 |
| 3 | 1 1 - - | 00 | 0 0 1 1 | 1 1 |
| 4 | - 0 1 - | 01 | 1 0 0 0 | 1 1 |
| 5 | - 0 0 - | 01 | 0 1 0 0 | 1 0 |
| 6 | - 1 - - | 01 | 0 0 1 1 | 1 0 |
| 7 | - - - - | 10 | 0 0 1 0 | 1 0 |
| 8 | - - - 0 | 11 | 0 0 1 0 | 1 1 |
| 9 | - 1 - 1 | 11 | 0 0 1 1 | 0 0 |
| 10 | - 0 0 1 | 11 | 0 1 0 0 | 0 0 |
| 11 | - 0 1 1 | 11 | 1 0 0 0 | 0 1 |

Для примера реализации логической схемы синтезируемого МПА Мили рассмотрим реализацию функций y_1, y_2, T_1 и T_2 :

$$y_1 = \bar{x}_1 \bar{\tau}_1 \bar{\tau}_2 \vee x_1 \bar{x}_2 \bar{x}_3 \bar{\tau}_1 \bar{\tau}_2 \vee \bar{x}_2 \bar{x}_3 \bar{\tau}_1 \bar{\tau}_2 \vee \bar{x}_2 \bar{x}_3 x_4 \tau_1 \tau_2,$$

$$y_2 = x_1 \bar{x}_2 \bar{x}_3 \bar{\tau}_1 \bar{\tau}_2 \vee \bar{x}_2 \bar{x}_3 \bar{\tau}_1 \bar{\tau}_2 \vee \bar{x}_2 \bar{x}_3 x_4 \tau_1 \tau_2,$$

$$T_1 = \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{\tau}_1 \bar{\tau}_2 \vee x_1 \bar{x}_2 \bar{x}_3 \bar{\tau}_1 \bar{\tau}_2 \vee x_1 x_2 \bar{\tau}_1 \bar{\tau}_2 \vee \bar{x}_2 \bar{x}_3 \bar{\tau}_1 \bar{\tau}_2 \vee \bar{x}_2 \bar{x}_3 \bar{\tau}_1 \bar{\tau}_2 \vee x_1 x_2 \bar{\tau}_1 \bar{\tau}_2 \vee \tau_1 \bar{\tau}_2 \vee \bar{x}_4 \tau_1 \tau_2 = x_1 \bar{\tau}_1 \bar{\tau}_2 \vee \bar{\tau}_1 \bar{\tau}_2 \vee \tau_1 \bar{\tau}_2 \vee \bar{x}_4 \tau_1 \tau_2,$$

$$T_2 = \bar{x}_1 \bar{\tau}_1 \bar{\tau}_2 \vee x_1 \bar{x}_2 \bar{x}_3 \bar{\tau}_1 \bar{\tau}_2 \vee x_1 x_2 \bar{\tau}_1 \bar{\tau}_2 \vee \bar{x}_2 \bar{x}_3 \bar{\tau}_1 \bar{\tau}_2 \vee \bar{x}_4 \tau_1 \tau_2 \vee \bar{x}_2 \bar{x}_3 x_4 \tau_1 \tau_2.$$

По полученной системе булевых функций построим схему (рис. 14.3).

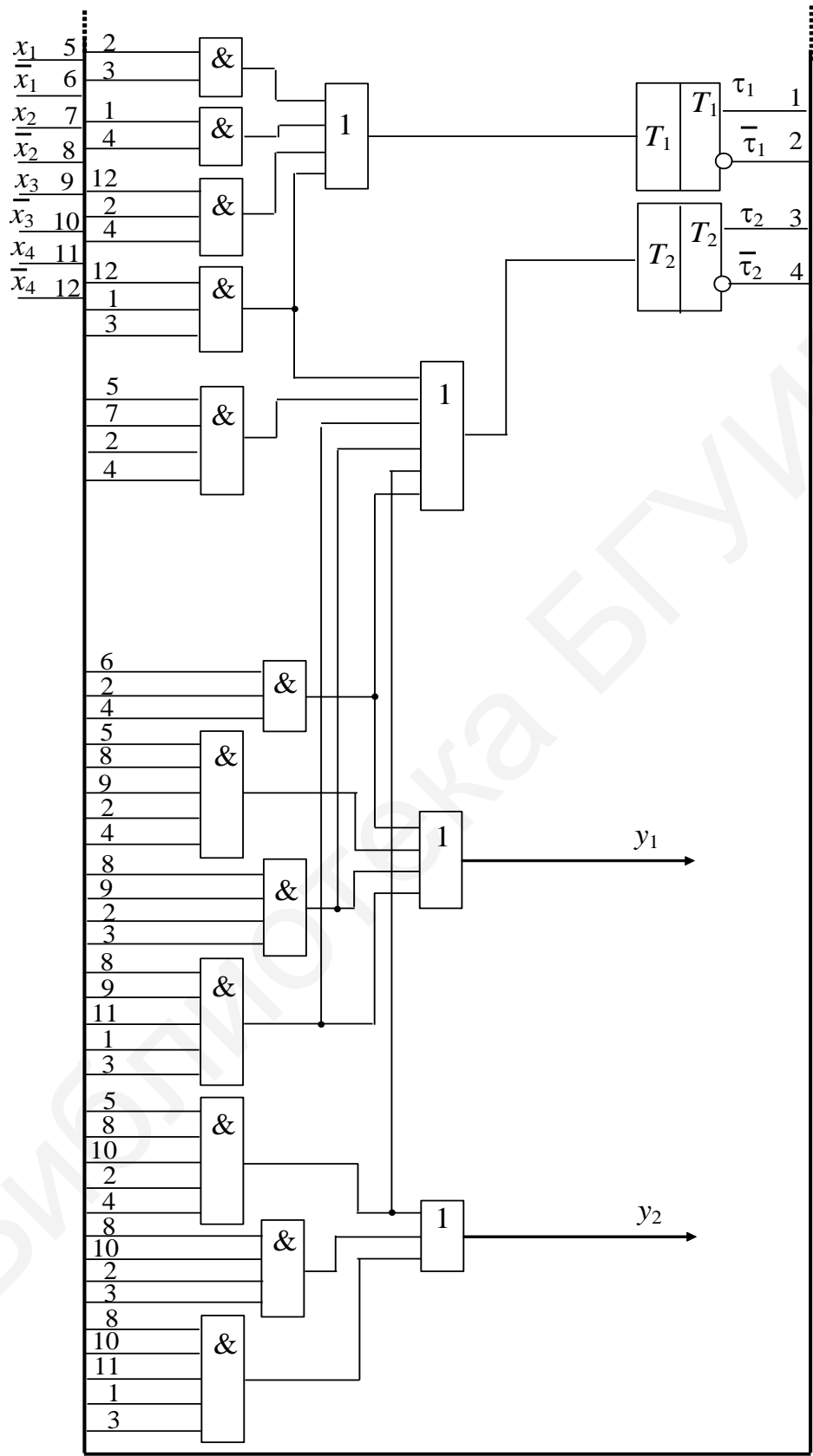


Рис. 14.3. Логическая схема автомата Мили

Пример синтеза МПА Мура по ГСА

Для наглядного сравнения двух методов синтеза (автоматов Мили и Мура) синтез автомата Мура выполним для алгоритма, представленного на ГСА, изображенной на рис. 14.1. Более подробно метод рассмотрен в [4, с. 160–164]. Как и ранее, для получения графа автомата исходная ГСА отмечается метками. Алгоритм разметки ГСА метками Мура приведен в [4, с. 160]. На рис. 14.4 изображена отмеченная ГСА.

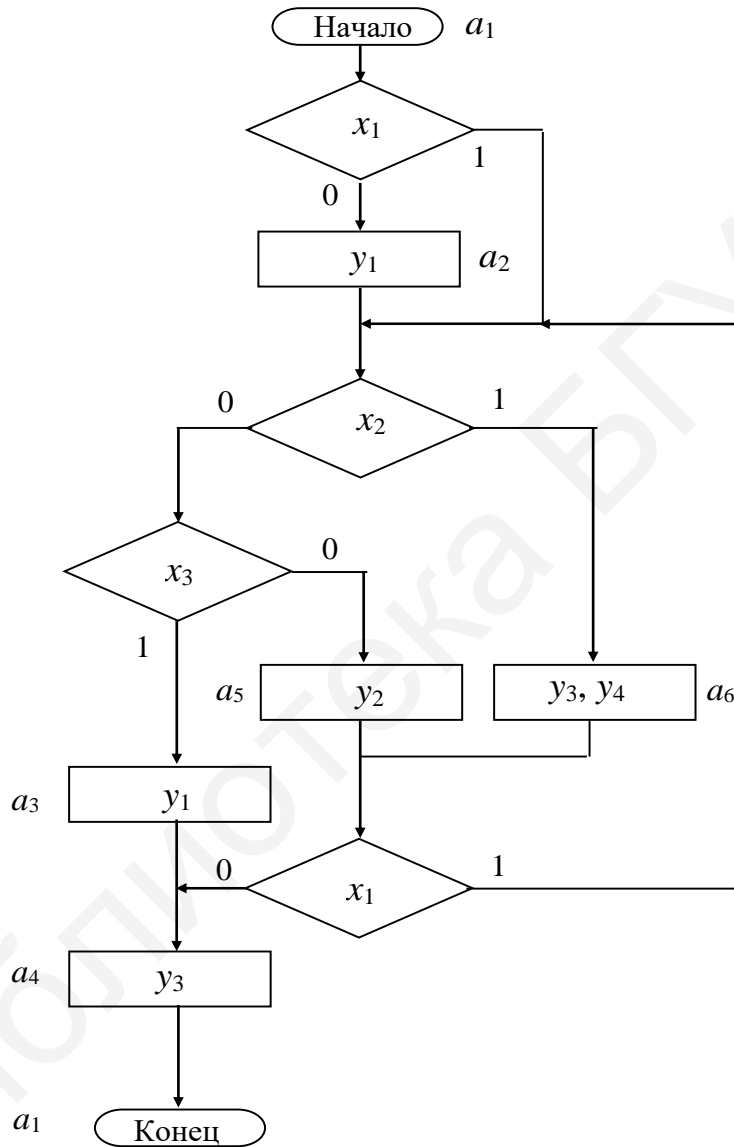


Рис. 14.4. Отмеченная ГСА при синтезе МПА Мура

Кодирование состояний автомата Мура выполним, как и ранее в случае синтеза автомата Мили, поставив каждому состоянию в соответствие двоичный эквивалент номера состояния.

Как и в предыдущем примере синтеза автомата Мили, при синтезе автомата Мура для представления автомата используем прямую структурную таблицу переходов (табл. 14.5). В структурной таблице автомата Мура совмещены

столбцы a_m и $Y(a_m)$. В столбце $F(a_m, a_s)$ отмечены функции возбуждения, изменяющие содержимое соответствующего элемента памяти на некотором переходе.

Таблица 14.5

| Исходное состояние | Код исходного состояния | Состояние перехода | Код состояния перехода | Входной сигнал | Функции возбуждения |
|--------------------|-------------------------|--------------------|------------------------|---------------------------|---|
| $a_m, Y(a_m)$ | $K(a_m)$ | a_s | $K(a_s)$ | $X(a_m, a_s)$ | $F(a_m, a_s)$ |
| $a_1, -$ | 000 | a_2 | 001 | \bar{x}_1 | $-\bar{S}_1 - \bar{S}_2 \bar{R}_3 S_3$ |
| | | a_6 | 101 | $x_1 x_2$ | $\bar{R}_1 S_1 - \bar{S}_2 \bar{R}_3 S_3$ |
| | | a_3 | 010 | $x_1 \bar{x}_2 x_3$ | $-\bar{S}_1 \bar{R}_2 S_2 - \bar{S}_3$ |
| | | a_5 | 100 | $x_1 \bar{x}_2 \bar{x}_3$ | $\bar{R}_1 S_1 - \bar{S}_2 - \bar{S}_3$ |
| a_2, y_1 | 001 | a_3 | 010 | $\bar{x}_2 x_3$ | $-\bar{S}_1 \bar{R}_2 S_2 R_3 \bar{S}_3$ |
| | | a_5 | 100 | $\bar{x}_2 \bar{x}_3$ | $\bar{R}_1 S_1 - \bar{S}_2 R_3 \bar{S}_3$ |
| | | a_6 | 101 | x_2 | $\bar{R}_1 S_1 - \bar{S}_2 \bar{R}_3 -$ |
| a_3, y_1 | 010 | a_4 | 011 | 1 | $-\bar{S}_1 \bar{R}_2 - \bar{R}_3 S_3$ |
| a_4, y_3 | 011 | a_1 | 000 | 1 | $-\bar{S}_1 R_2 \bar{S}_2 R_3 \bar{S}_3$ |
| a_5, y_2 | 100 | a_4 | 011 | \bar{x}_1 | $R_1 \bar{S}_1 \bar{R}_2 S_2 \bar{R}_3 S_3$ |
| | | a_3 | 010 | $x_1 \bar{x}_2 x_3$ | $R_1 \bar{S}_1 \bar{R}_2 S_2 - \bar{S}_3$ |
| | | a_5 | 100 | $x_1 \bar{x}_2 \bar{x}_3$ | $\bar{R}_1 S_1 - \bar{S}_2 - \bar{S}_3$ |
| | | a_6 | 101 | $x_1 x_2$ | $\bar{R}_1 - - \bar{S}_2 \bar{R}_3 S_3$ |
| $a_6, y_3 y_4$ | 101 | a_4 | 011 | \bar{x}_1 | $R_1 \bar{S}_1 \bar{R}_2 S_2 \bar{R}_3 -$ |
| | | a_6 | 101 | $x_1 x_2$ | $\bar{R}_1 - - \bar{S}_2 \bar{R}_3 -$ |
| | | a_5 | 100 | $x_1 \bar{x}_2 \bar{x}_3$ | $\bar{R}_1 - - \bar{S}_2 R_3 \bar{S}_3$ |
| | | a_3 | 010 | $x_1 \bar{x}_2 x_3$ | $-\bar{S}_1 \bar{R}_2 S_2 R_3 \bar{S}_3$ |

Блок памяти МПА Мура реализуем с использованием, например, RS-триггеров. Информация в столбце $F(a_m, a_s)$ структурной таблицы получена из таблицы работы RS-триггера [4] (табл. 14.6).

Таблица 14.6

| Q_t | RS | | | |
|-------|----|----|----|-----|
| | 00 | 01 | 10 | 11 |
| 0 | 0 | 1 | 0 | x |
| 1 | 1 | 1 | 0 | x |

Для пояснения ниже показаны значения сигналов R и S , подаваемых на триггер для перевода его из старого состояния в новое:

$$\begin{aligned} 0 \rightarrow 0 & - \bar{S}, & 1 \rightarrow 0 & - R\bar{S}, \\ 0 \rightarrow 1 & - \bar{R}S, & 1 \rightarrow 1 & - \bar{R}. \end{aligned}$$

Для удобства построения схемы, реализующей синтезируемый МПА Мура, результаты, приведенные в структурной таблице (см. табл. 14.5), как и ранее при синтезе автомата Мили, представим в виде таблицы истинности (табл. 14.7).

Таблица 14.7

| № п/п | $x_1x_2x_3x_4$ | $\tau_1\tau_2\tau_3$ | $y_1y_2y_3y_4$ | $R_1S_1R_2S_2R_3S_3$ |
|-------|----------------|----------------------|----------------|----------------------|
| 0 | 0---- | 000 | 0000 | *0*001 |
| 1 | 11--- | 000 | 0000 | 01*001 |
| 2 | 101- | 000 | 0000 | *001*0 |
| 3 | 100- | 000 | 0000 | 01*0*0 |
| 4 | -01- | 001 | 1000 | *00110 |
| 5 | -00- | 001 | 1000 | 01*010 |
| 6 | -1--- | 001 | 1000 | 01*00* |
| 7 | ----- | 010 | 1000 | *00*01 |
| 8 | ----- | 011 | 0010 | *01010 |
| 9 | 0---- | 100 | 0100 | 100101 |
| 10 | 101- | 100 | 0100 | 1001*0 |
| 11 | 100- | 100 | 0100 | 01*0*0 |
| 12 | 11--- | 100 | 0100 | 0**001 |
| 13 | 0--1 | 101 | 0011 | 10010* |
| 14 | 11--- | 101 | 0011 | 0**00* |
| 15 | 100- | 101 | 0011 | 0**010 |
| 16 | 101- | 101 | 0011 | *00110 |

Для примера реализации логической схемы синтезируемого МПА Мура рассмотрим реализацию функций y_1, y_2, R_3 и S_3 :

$$y_1 = \bar{x}_2\bar{x}_3\bar{\tau}_1\bar{\tau}_2\bar{\tau}_3 \vee \bar{x}_2\bar{x}_3\bar{\tau}_1\bar{\tau}_2\tau_3 \vee \bar{x}_2\bar{\tau}_1\bar{\tau}_2\bar{\tau}_3 \vee \bar{\tau}_1\bar{\tau}_2\bar{\tau}_3 = \bar{\tau}_1\bar{\tau}_2\bar{\tau}_3 \vee \bar{\tau}_1\bar{\tau}_2\tau_3,$$

$$y_2 = \bar{x}_1\bar{\tau}_1\bar{\tau}_2\bar{\tau}_3 \vee \bar{x}_1\bar{x}_2\bar{x}_3\bar{\tau}_1\bar{\tau}_2\bar{\tau}_3 \vee \bar{x}_1\bar{x}_2\bar{x}_3\bar{\tau}_1\bar{\tau}_2\tau_3 \vee \bar{x}_1\bar{x}_2\bar{\tau}_1\bar{\tau}_2\bar{\tau}_3 = \bar{\tau}_1\bar{\tau}_2\bar{\tau}_3,$$

$$\begin{aligned} R_3 &= \bar{x}_2\bar{x}_3\bar{\tau}_1\bar{\tau}_2\bar{\tau}_3 \vee \bar{x}_2\bar{x}_3\bar{\tau}_1\bar{\tau}_2\tau_3 \vee \bar{\tau}_1\bar{\tau}_2\bar{\tau}_3 \vee \bar{x}_1\bar{x}_2\bar{x}_3\bar{\tau}_1\bar{\tau}_2\bar{\tau}_3 \vee \bar{x}_1\bar{x}_2\bar{x}_3\bar{\tau}_1\bar{\tau}_2\tau_3 = \\ &= \bar{x}_2\bar{\tau}_1\bar{\tau}_2\bar{\tau}_3 \vee \bar{\tau}_1\bar{\tau}_2\bar{\tau}_3 \vee \bar{x}_1\bar{x}_2\bar{\tau}_1\bar{\tau}_2\bar{\tau}_3. \end{aligned}$$

Использование наборов, не содержащихся в таблице истинности, т. е. неопределенных наборов, позволяет выполнить дальнейшее упрощение функции R_3 :

$$R_3 = \bar{x}_2 \bar{\tau}_1 \bar{\tau}_2 \tau_3 \vee \tau_2 \tau_3 \vee x_1 x_2 \tau_1 \tau_3,$$

$$\begin{aligned} S_3 &= \bar{x}_1 \bar{\tau}_1 \bar{\tau}_2 \tau_3 \vee x_1 x_2 \bar{\tau}_1 \bar{\tau}_2 \tau_3 \vee \bar{\tau}_1 \tau_2 \bar{\tau}_3 \vee x_1 \tau_1 \bar{\tau}_2 \tau_3 \vee x_1 x_2 \tau_1 \bar{\tau}_2 \tau_3 = \\ &= \bar{x}_1 \bar{\tau}_2 \tau_3 \vee x_1 x_2 \bar{\tau}_2 \tau_3 \vee \bar{\tau}_1 \tau_2 \tau_3. \end{aligned}$$

Аналогично использование неопределенных наборов таблицы истинности позволяет выполнить дальнейшее упрощение функции S_3 :

$$S_3 = \bar{x}_1 \bar{\tau}_2 \tau_3 \vee x_1 x_2 \bar{\tau}_2 \tau_3 \vee \tau_2 \tau_3.$$

По полученной системе булевых функций построим схему (рис. 14.5).

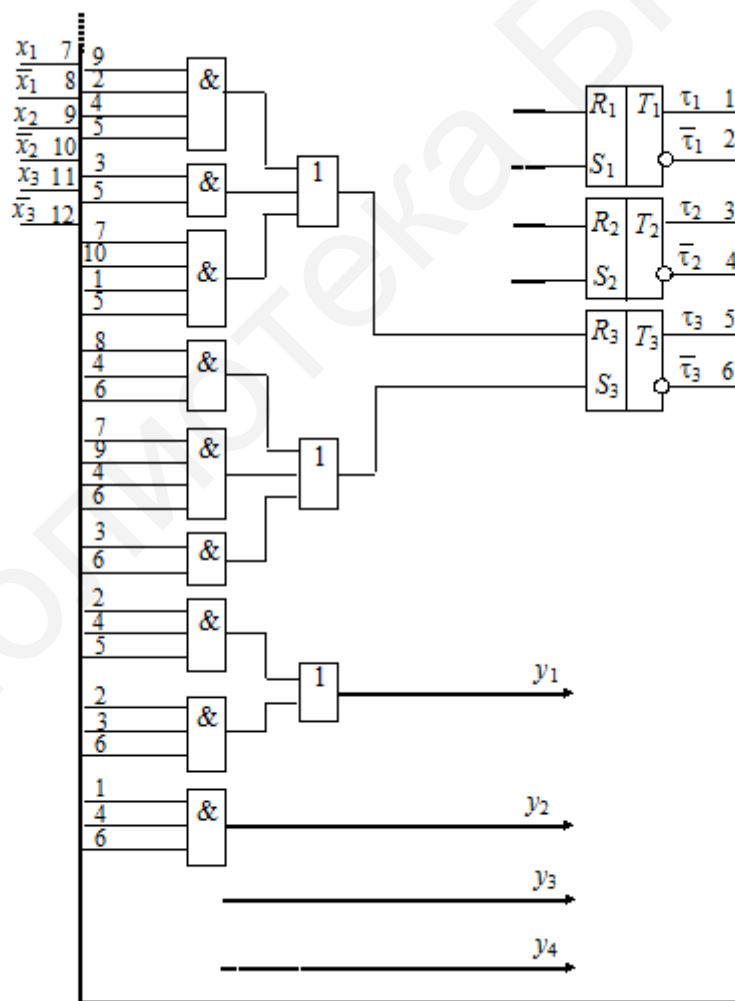


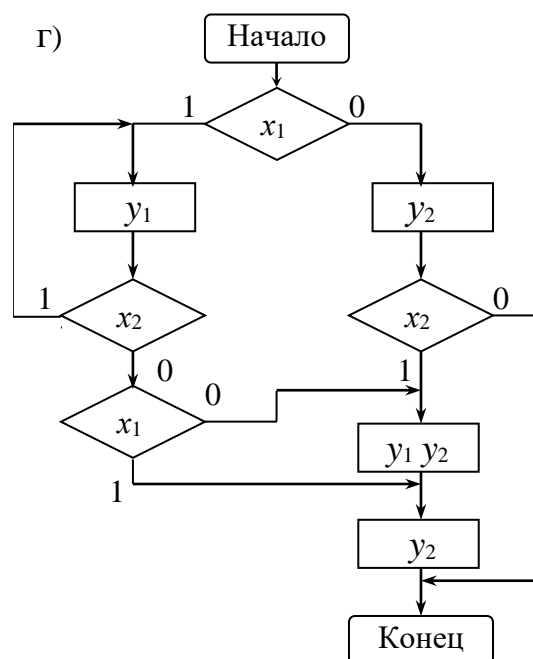
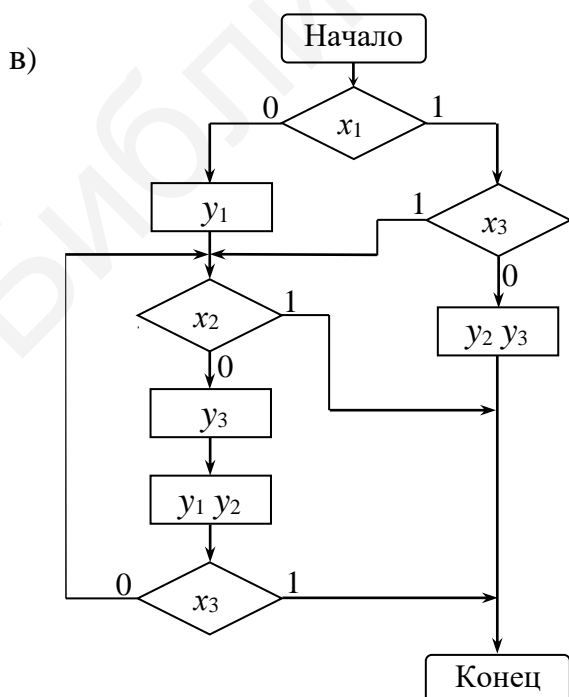
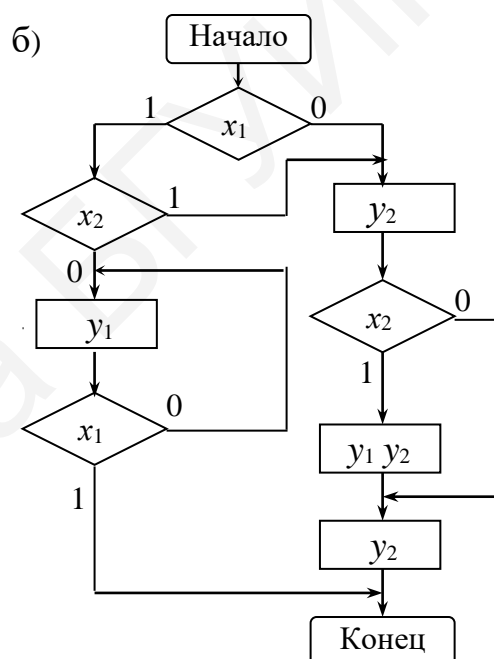
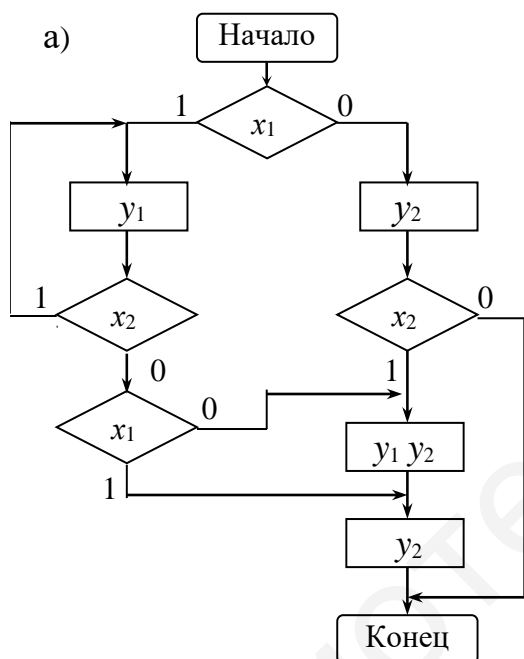
Рис. 14.5. Логическая схема автомата Мура

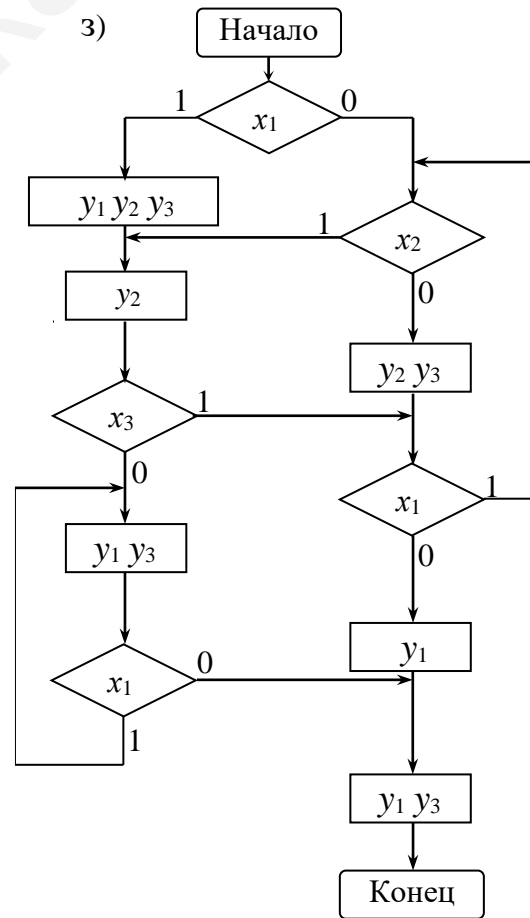
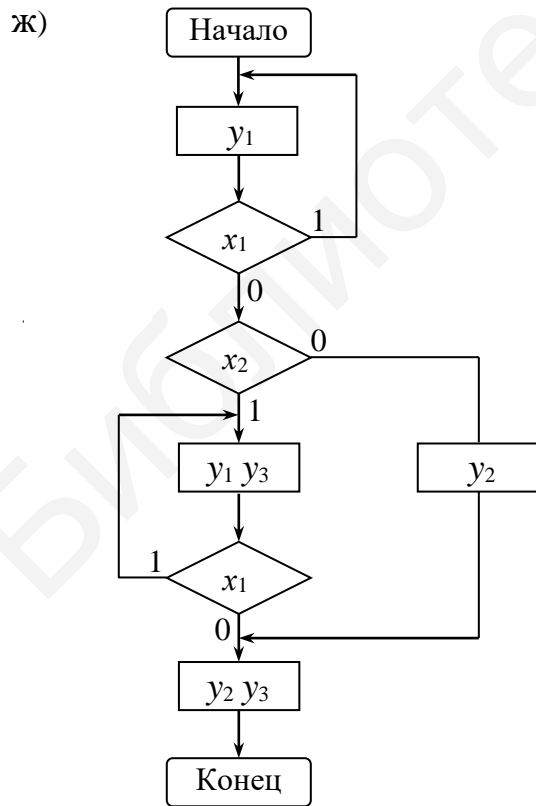
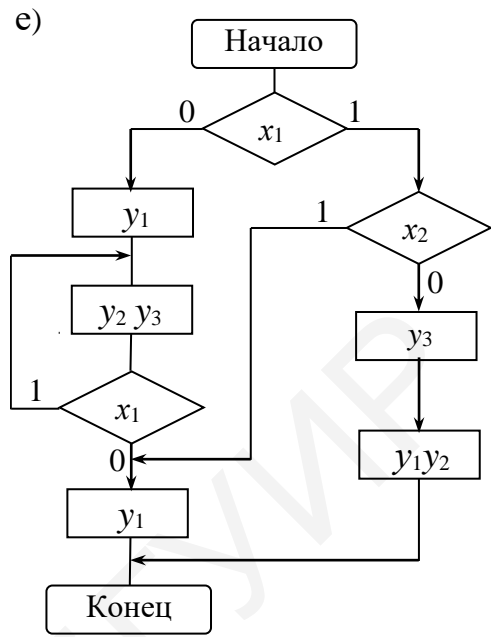
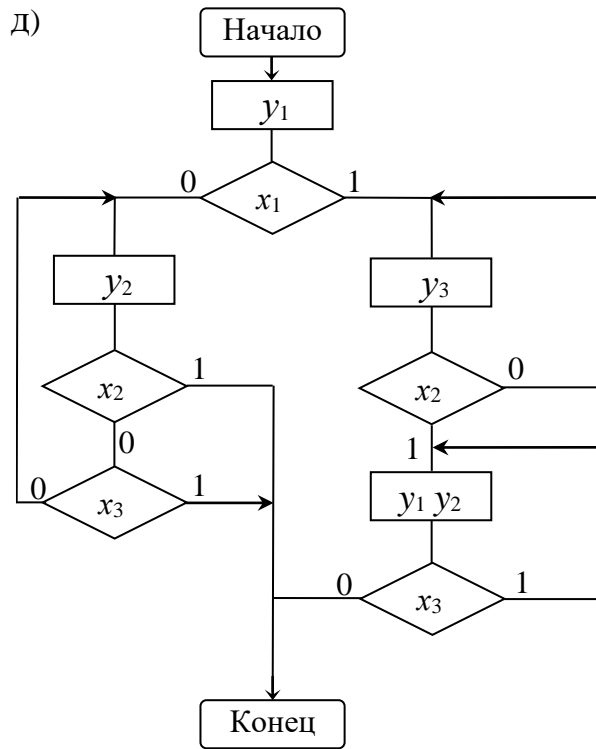
В заключение отметим, что, как видно из примеров, синтезированный МПА Мили работает не медленнее, чем МПА Мура для одного и того же алгоритма. Это определяется в общем случае количеством внутренних состояний автомата (числом меток на ГСА). Для рассмотренных примеров МПА Мили его блок памяти оказался также более экономичным (на один триггер меньше).

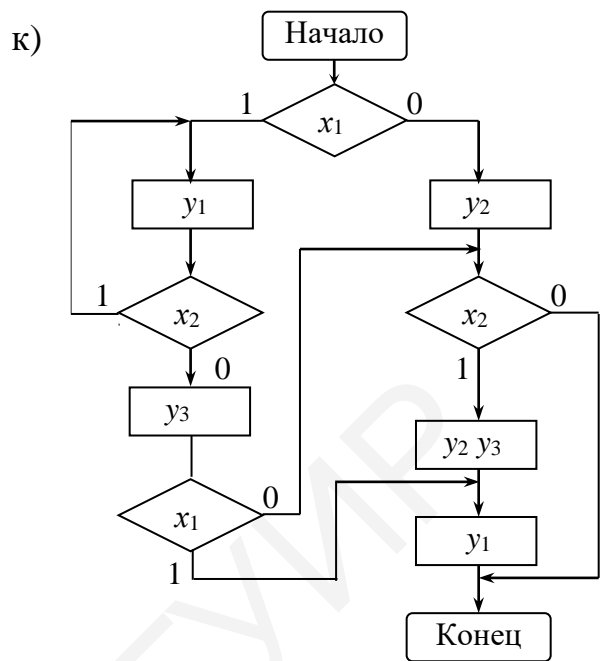
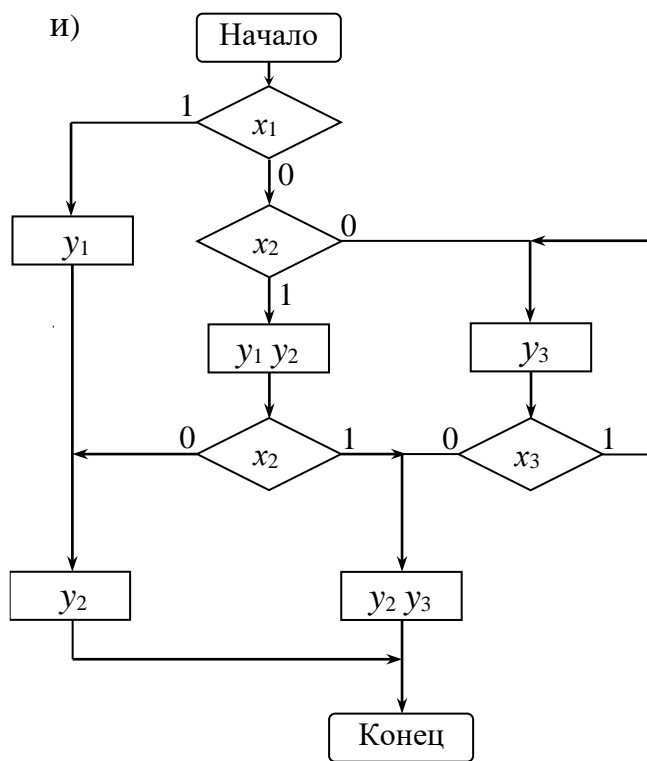
Практическое задание

Выполнить синтез структурного автомата по ГСА. Синтезировать МПА Мили (Мура). Память автомата реализовать на следующих триггерах:

- 1) *D*-триггерах; 2) *T*-триггерах; 3) *RS*-триггерах; 4) *JK*-триггерах.







Библиотека БГУ

Список использованных источников

1. Савельев, А. Я. Прикладная теория цифровых автоматов : учебник для вузов по спец. ЭВМ / А. Я. Савельев. – М. : Высш. шк., 1987.
2. Лысиков, Б. Г. Арифметические и логические основы цифровых автоматов : учебник для вузов по спец. «Электронные вычислительные машины» / Б. Г. Лысиков. – 2-е изд., перераб. и доп. – Минск : Выш. шк., 1980.
3. Лысиков, Б. Г. Цифровая и вычислительная техника : учебник / Б. Г. Лысиков. – Минск : УП Экоперспектива, 2003.
4. Луцик, Ю. А. Арифметические и логические основы вычислительной техники : метод. пособие / Ю. А. Луцик, И. В. Лукьянова. – Минск : БГУИР, 2004.
5. Мельников, В. П. Информационные технологии : учебник для студентов вузов / В. П. Мельников. – 2-е изд., стер. – М. : Академия, 2009.
6. Синаторов, С. В. Информационные технологии : учеб. пособие / С. В. Синаторов. – М. : Альфа-М : ИНФРА-М, 2009.
7. Жмакин, А. П. Архитектура ЭВМ / А. П. Жмакин. – СПб. : БХВ-Петербург, 2006.
8. Потапов, И. В. Прикладная теория цифровых автоматов : учеб. пособие / И. В. Потапов. – Омск : Изд-во ОмГТУ, 2014.
9. Постников, А. И. Прикладная теория цифровых автоматов : учеб. пособие / А. И. Постников, О. В. Непомнящий, Л. В. Макуха. – Красноярск : Сиб. федер. ун-т, 2017.
10. Савельев, А. Я. Основы информатики : учебник для вузов / А. Я. Савельев. – М. : Изд-во МГТУ им. Н. Э. Баумана, 2001. – 328 с.

Содержание

| | |
|---|----|
| Введение | 3 |
| Тема 1. Системы счисления и операции с числами. | 4 |
| Тема 2. Коды чисел..... | 8 |
| Тема 3. Формы представления чисел. Числа с фиксированной и плавающей запятой. Сложение чисел с плавающей запятой. | 11 |
| Тема 4. Машинные методы умножения чисел. | 16 |
| Тема 5. Умножение с хранением переносов. | 19 |
| Тема 6. Умножение чисел на два разряда множителя одновременно в прямом коде..... | 21 |
| Тема 7. Умножение чисел в дополнительном коде..... | 23 |
| Тема 8. Умножение чисел на два разряда в дополнительном коде..... | 25 |
| Тема 9. Машинные методы деления чисел. | 27 |
| Тема 10. Методы минимизации булевых функций. Алгоритм Квайна. Функционально полный базис. | 30 |
| Тема 11. Минимизация булевых функций. Карты Вейча (Карно). Минимизация неполностью определенных булевых функций. | 34 |
| Тема 12. Кубическое представление булевых функций. Алгоритм Квайна – Мак-Класки. Алгоритм извлечения (Рота). | 41 |
| Тема 13. Абстрактные и структурные автоматы. Память автомата, триггеры. Канонический метод синтеза структурного автомата. | 49 |
| Тема 14. Синтез структурного автомата по граф-схеме алгоритма. Гонки, риск сбоя..... | 58 |
| Список использованных источников..... | 71 |