

МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ
И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ
Северо-Кавказский филиал ордена Трудового Красного Знамени федерального
государственного бюджетного образовательного учреждения высшего
образования
«Московский технический университет связи и информатики»

Методические указания
для выполнения практических занятий

по дисциплине

ТЕОРИЯ АВТОМАТОВ

Направление подготовки 09.03.01 Информатика и вычислительная техника

Ростов-на-Дону
2022

Методические указания
для выполнения практических занятий
по дисциплине

ТЕОРИЯ АВТОМАТОВ

Составитель: В.И. Юхнов, зав. каф. «ИТСС»

Рассмотрено и одобрено
на заседании кафедры «ИТСС»
Протокол от «19» декабря 2022 г., № 5.

Введение

Настоящие методические указания содержат материал для проведения практических занятий по одному из основных разделов дисциплины “Теория автоматов” – “Логические основы цифровых автоматов”.

Целью занятий является практическое закрепление знаний о формах представления и методах преобразования логических функций, а также методике синтеза комбинационных схем.

Каждое практическое занятие включает в себя постановку цели занятия, краткий теоретический материал по теме, характерные примеры, контрольные вопросы и упражнения для самостоятельной работы.

До проведения занятия студент должен уяснить его цель и ответить на контрольные вопросы. Для самостоятельной подготовки к занятиям студентам рекомендуется литература, представленная в списке использованных источников. Во время занятия разбираются примеры и выполняются упражнения по вариантам. Контроль знаний проводится по результатам ответов на контрольные вопросы и выполнения упражнений.

1. Практическое занятие №1. Формы представления логических функций

Существует несколько форм представления логических функций (ЛФ), используемых на различных этапах проектирования комбинационных схем, в частности: словесная, табличная, аналитическая, геометрическая, кубическая.

Целью практического занятия является изучение табличной и аналитической форм представления ЛФ и алгоритмов перехода от табличного описания ЛФ к аналитическому описанию.

1.1 Табличная форма представления ЛФ

Логическая функция наиболее наглядно представляется посредством таблицы истинности (ТИ) и карты Карно.

Таблица истинности - это таблица, в которой каждому двоичному набору значений аргументов x_i ($i = 1, n$) ставится в соответствие значение функции $Y = f(x_1, x_2, \dots, x_i, \dots, x_n)$ на данном наборе. В таблице 1.1 в качестве примера представлена ТИ функции для трех аргументов. Функция Y принимает значение 1 или 0 на каждом наборе. Если значение функции не определено, то в соответствующей позиции ТИ ставится прочерк.

Таблица 1.1 – Таблица истинности логической функции

N	x_1	x_2	x_3	Y
0	0	0	0	1
1	0	0	1	1
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

Иногда используют списочную форму представления ТИ, в которой приводится список единичных и нулевых наборов. Так, рассматриваемая в примере функция в списочной форме может быть представлена в виде:

$$Y = F(x_1, x_2, x_3) = \bigvee_1(0, 1, 3, 6, 7)$$

$$Y = \bigwedge_0(2, 4, 5)$$

В скобках, приведены десятичные эквиваленты двоичных кодов наборов.

Карта Карно представляет собой прямоугольную таблицу, содержащую 2^n клеток, причем каждая клетка находится на пересечении i -ой строки и j -ого столбца, a_i и a_j являются составными элементами двоичного набора n – местной ЛФ.

Карта Карно имеет следующие особенности:

1) любая пара клеток, являющихся соседними по вертикали или горизонтали, а также любая пара клеток, расположенных симметрично карте по вертикали или горизонтали, соответствуют двоичным наборам, отличающимся значением лишь одной переменной (т.е. отличается в одном разряде);

2) в клетках карты Карно указываются значения функции на соответствующем наборе;

3) двоичные наборы аргументов, которыми отмечены строки, а также двоичные наборы аргументов, которыми отмечены столбцы, образуют код Грея, в котором любая пара соседних кодов, а также последний и первый коды, различаются лишь в одном разряде, поэтому карта Карно может быть представлена как некоторый цилиндр, образованный при склеивании левого и правого края таблицы, либо верхнего и нижнего.

На рисунке 1.1 представлены два варианта карт Карно для трех переменных, соответствующие таблице истинности 1.1.

		$X_2 X_3$			
		00	01	11	10
X_1	0	1	1	1	0
	1	0	0	1	1

		X_2			
		X_3			
X_1	0	1	1	1	0
	1	0	0	1	1

Вариант 1
Вариант 2

Рисунок 1.1 – Варианты карт Карно для трех переменных

Во втором варианте для удобства построения и использования карты строки и столбцы, соответствующие единичным значениям переменных x_j , отмечаются чертой, т.е. если столбец или строка отмечена чертой, то значение x_j в этой строке или столбце равно 1, в противном случае - 0.

На рисунке 1.2 представлены два варианта карт Карно для двух (а), четырех (б) и пяти (в) переменных:

		X_2	
		0	1
X_1	0		
	1		

Вариант 1

		X_2	
X_1			

Вариант 2

a)

		X_3X_4			
		00	01	11	10
X_1X_2	00				
	01				
	11				
	10				

Вариант 1

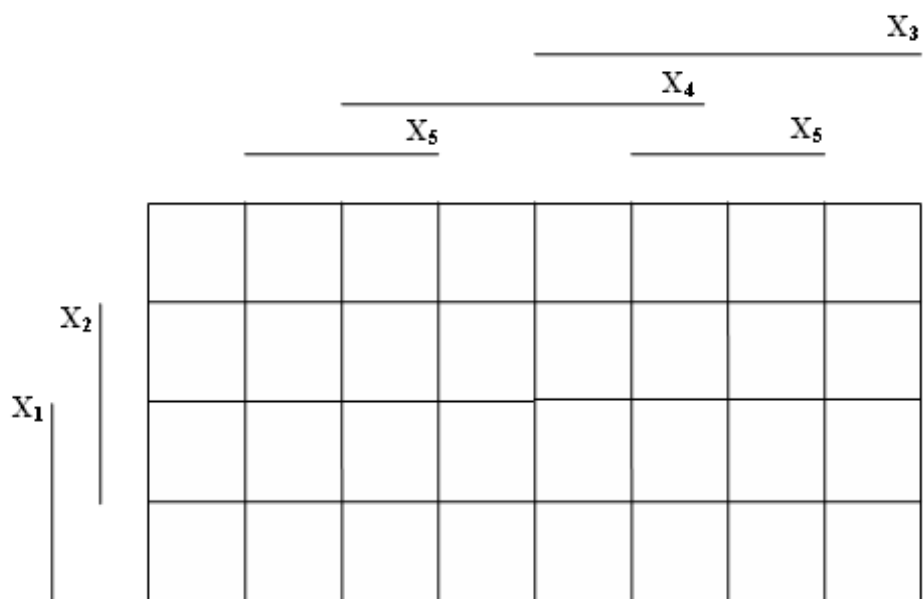
б)

		X_3			
		X_4			
X_1	X_2				

Вариант 2

		$X_3X_4X_5$							
		000	001	011	010	110	111	101	100
X_1X_2	00								
	01								
	11								
	10								

Вариант 1



Вариант 2

в)

Рисунок 1.2 - Варианты карт Карно а) для двух переменных, б) четырех переменных, в) пяти переменных

В случае шести переменных обычно используют четыре 16 - клеточные карты Карно. Для ЛФ с числом переменных $n > 6$ карты Карно становятся громоздкими и неудобными для практического применения.

1.2 Аналитическая форма представления ЛФ

Аналитической формой представления ЛВ является формульное описание ЛФ, содержащее обозначение переменных, знаков операций и разделительных скобок. К основным операциям, используемым в формулах, относятся отрицание, конъюнкция и дизъюнкция. Причем, если в аналитическом выражении отсутствуют скобки, старшинство операций определяется следующим образом: сначала выполняется отрицание, затем конъюнкция и затем дизъюнкция.

В теории автоматов существуют следующие основные аналитические формы представления ЛФ: дизъюнктивная нормальная форма (ДНФ), конъюнктивная нормальная форма (КНФ), совершенная дизъюнктивная нормальная форма (СДНФ) и совершенная конъюнктивная нормальная форма (СКНФ).

Составным элементом перечисленных форм являются элементарные конъюнкции (ЭК) и элементарные дизъюнкции (ЭД).

Элементарной конъюнкцией (дизъюнкцией) называется конъюнкция (дизъюнкция) переменных, каждая из которых входит в нее один раз в прямом или инверсном виде.

Число переменных, составляющих ЭК или ЭД, называют ее рангом.

Например, $\overline{x_1} \cdot x_2 \cdot \overline{x_3}$ - ЭК третьего ранга; $x_1 \cdot x_2 \cdot x_3 \cdot x_4$ - четвертого ранга.

ДНФ представляет собой регулярное аналитическое выражение (формулу) ЛФ в виде дизъюнкции (суммы) элементарных конъюнкций.

Например:

$$Y = x_1 \cdot x_2 \vee \overline{x_3} \cdot x_4 \cdot x_5$$

В теории автоматов часто используют следующие виды ДНФ:

- 1) совершенную ДНФ (СДНФ);
- 2) сокращенную ДНФ (СкДНФ);
- 3) тупиковую ДНФ (ТДНФ);
- 4) минимальную ДНФ (МДНФ).

Если в каждом члене ДНФ представлены все аргументы (или их инверсии) функции, то такая форма называется СДНФ.

Например:

$$Y = x_1 x_2 x_3 \vee \overline{x_1} x_2 x_3 \vee \overline{x_1} \overline{x_2} x_3 \vee \overline{x_1} x_2 \overline{x_3}$$

СкДНФ - это ДНФ в виде дизъюнкции всех ее простых импликант.

Импликантой ЛФ $Y = f(x_1, x_2, \dots, x_n)$ называется функция $Y = f(x_1, x_2, \dots, x_n)$, которая обращается в 1 на некотором подмножестве единичных наборов функции Y .

Например, импликантами являются ЭК в ДНФ функции.

Простой импликантой функции $Y = f(x_1, x_2, \dots, x_n)$ называется импликанта, которая не поглощается никакой другой импликантой данной функции Y .

Например, функция $Y_1 = \overline{x_1} x_2 x_3$ является импликантой функции $Y = \overline{x_1} x_3 \vee \overline{x_1} x_2 x_3$, а функция $Y_2 = \overline{x_1} x_3$ является простой импликантой функции Y .

ТДНФ - это СкДНФ, из которой нельзя исключить ни одной простой импликанты без потери эквивалентности функции.

МДНФ - это ТДНФ, содержащая минимальное число букв среди возможных ТДНФ функции.

КНФ - это регулярное аналитическое выражение ЛФ в виде (произведения) элементарных дизъюнкций.

Например:

$$Y = x_1 (x_2 \vee \overline{x_3} \vee x_4) \cdot (\overline{x_2} \vee x_4)$$

СДНФ n -местной ЛФ является ДНФ, содержащая ЭК только ранга n .

СКНФ n -местной ЛФ является КНФ, содержащая ЭД только ранга n .

Каждой ЛФ соответствует одна СДНФ и одна СКНФ.

В общем случае СДНФ можно представить в следующем виде:

$$F(x_1, x_2, \dots, x_i, \dots, x_n) = \bigvee_1 (\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_i, \dots, \tilde{x}_n); \text{ где}$$

$$\tilde{x}_i = \begin{cases} x_i, & \text{если } x_i = 1 \\ \overline{x_i}, & \text{если } x_i = 0 \end{cases}$$

Например, двоичному набору 1010 соответствует ЭК $\overline{x_1} \overline{x_2} x_3 \overline{x_4}$.

СКНФ логической функции - это конъюнкция конstituент нуля, соответствующих входным наборам, для которых функция равна нулю.

В общем случае СКНФ можно представить в форме:

$$F(x_1, x_2, \dots, x_i, \dots, x_n) = \bigwedge_0 (\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_i, \dots, \tilde{x}_n); \text{ где}$$

$$\tilde{x}_i = \begin{cases} x_i, & \text{если } x_i = 0 \\ \overline{x_i}, & \text{если } x_i = 1 \end{cases}$$

Например, двоичному набору 1010 соответствует ЭД: $\overline{x_1} \vee x_2 \vee \overline{x_3} \vee x_4$.

Алгоритм перехода от таблицы истинности логической функции к ее записи в виде СДНФ имеет следующий вид:

- 1) выбрать в ТИ такие входные наборы, на которых функция обращается в единицу;
- 2) записать конstituенты единицы для выбранных входных наборов;
- 3) полученные конstituенты единиц соединить между собой знаками дизъюнкции.

Например, для логической функции, представленной в таблице истинности 1.1, СДНФ имеет следующий вид:

$$Y = \overline{x_1} \overline{x_2} \overline{x_3} \vee \overline{x_1} \overline{x_2} x_3 \vee \overline{x_1} x_2 \overline{x_3} \vee x_1 x_2 \overline{x_3} \vee x_1 x_2 x_3$$

Алгоритм перехода от табличного задания логической функции к ее записи в виде СКНФ имеет следующий вид:

- 1) выбрать в ТИ такие входные наборы, на которых функция имеет нулевые значения;
- 2) записать конstituенты нуля для выбранных входных наборов;
- 3) полученные конstituенты нуля соединить между собой знаками конъюнкции.

Например, для логической функции, представленной в таблице истинности 1.1, СКНФ имеет следующий вид:

$$Y = x_1 \overline{x_2} \overline{x_3} \cdot \overline{x_1} x_2 x_3 \cdot \overline{x_1} x_2 \overline{x_3}$$

Контрольные вопросы

- 1 Для чего нужны таблицы истинности и карта Карно?
- 2 Как составляется таблица истинности?
- 3 Как заполняется карта Карно?
- 4 Дать определение элементарной конъюнкции, элементарной СДНФ, СКНФ.
- 5 Как построить СДНФ, СКНФ?

Упражнение №1

Построить таблицу истинности, карту Карно, СДНФ и СКНФ для следующих вариантов ЛФ, заданной списком единичных наборов:

1. (0, 2, 4, 6, 8, 10, 12, 14)
2. (1, 3, 5, 7, 9, 11, 13, 15)
3. (0, 1, 4, 5, 8, 9, 12, 13)
4. (1, 2, 5, 6, 9, 10, 13, 14)
5. (1, 5, 7, 9, 10, 12, 14)
6. (0, 2, 5, 8, 10, 13, 14, 15)
7. (1, 6, 8, 9, 10, 13, 14, 15)
8. (0, 2, 3, 5, 8, 9, 11, 13, 15)
9. (0, 3, 5, 6, 8, 9, 10, 12)
10. (1, 4, 5, 7, 9, 13, 14, 15)
11. (3, 5, 6, 8, 9, 13, 14, 15)
12. (1, 2, 3, 4, 5, 11, 12, 13)
13. (0, 1, 2, 5, 6, 7, 12, 13, 14)
14. (6, 7, 8, 11, 12, 13, 15)
15. (0, 4, 8, 12, 13, 14, 15)

2. Практическое занятие №2. Алгебраическое преобразование формул логических функций

Преобразование формул ЛФ производят, как правило, для достижения следующих целей:

- 1) для получения более простого аналитического выражения, описывающего комбинационную схему (КС), вследствие чего синтезируемая КС также имеет меньшую конструктивную сложность;
- 2) для преобразования формулы ЛФ к регулярному виду, используемому для регуляризации и сравнения КС;
- 3) для получения СДНФ и СКНФ ЛФ.

Целью практического занятия является изучение способов преобразования формул ЛФ.

В основе правил преобразований логических функций лежат законы, аксиомы и теоремы булевой алгебры.

2.1 Законы булевой алгебры

1 Закон коммутативности:

$$a \vee b = b \vee a;$$

$$a \cdot b = b \cdot a$$

2 Закон ассоциативности:

$$a \cdot (b \cdot c) = (a \cdot b) \cdot c;$$

$$a \vee (b \vee c) = (a \vee b) \vee c$$

3 Закон дистрибутивности (распределительный закон):

$$a \cdot (b \vee c) = a \cdot b \vee a \cdot c;$$

$$a \vee (b \cdot c) = (a \vee b) \cdot (a \vee c)$$

2.2 Аксиомы и теоремы булевой алгебры

- | | | |
|----|--|--|
| 1 | $a = 1$, если $a \neq 0$; | $a = 0$, если $a \neq 1$; |
| 2 | $0 \cdot 0 = 0$; | $1 + 1 = 1$; |
| 3 | $1 \cdot 1 = 1$; | $0 + 0 = 0$; |
| 4 | $1 \cdot 0 = 0$; | $0 + 1 = 1$; |
| 5 | $\bar{0} = 1$; | $\bar{1} = 0$; |
| 6 | $a + 0 = a$; | $a \cdot 1 = a$; |
| 7 | $a + 1 = 1$; | $a \cdot 0 = 0$; |
| 8 | $a + a = a$; | $a \cdot a = a$; |
| 9 | $\overline{(\bar{a})} = a$; | $\overline{(\bar{a})} = a$; |
| 10 | $a + \bar{a} = 1$; | $a \cdot \bar{a} = 0$; |
| 11 | $\overline{a + b + c} = \bar{a} \cdot \bar{b} \cdot \bar{c}$; | $\overline{a \cdot b \cdot c} = \bar{a} + \bar{b} + \bar{c}$ |

Для преобразования ЛФ используют следующие правила.

1). Правило склеивания для соседних элементарных конъюнкций (СЭК).

СЭК – это две ЭК одного и того же ранга, являющиеся функциями одних и тех же переменных и отличающиеся только знаком отрицания (инверсии) одной из переменных.

Например:

$$x_1 x_2 x_3 \quad \text{и} \quad x_1 \overline{x_2} x_3$$

Правило склеивания является следствием распределительного закона и имеет следующий вид:

логическую сумму двух соседних ЭК ранга r можно заметить одной элементарной конъюнкцией ранга $(r-1)$.

Например:

$$\overline{x_1} x_2 \vee x_1 x_2 = x_2 \quad \text{и} \quad \overline{x_1} \overline{x_2} x_3 \vee x_1 \overline{x_2} x_3 = \overline{x_2} x_3$$

2). Правило склеивания для соседних элементарных дизъюнкций (СЭД).

СЭД – это две элементарные дизъюнкции одного и того же ранга, являющиеся функциями одних и тех же переменных.

Например:

$$x_1 \vee x_2 \vee x_3 \quad \text{и} \quad x_1 \vee \overline{x_2} \vee x_3 .$$

Правило склеивания для СЭД имеет следующий вид: логическое произведение двух СЭД ранга $(r-1)$, является общей частью исходных конъюнкций.

Например:

$$(x_1 \vee \overline{x_2} \vee x_3 \vee \overline{x_4}) \cdot (\overline{x_1} \vee \overline{x_2} \vee x_3 \vee \overline{x_4}) = \overline{x_2} \vee x_3 \vee \overline{x_4} .$$

3). Правило поглощения для элементарных дизъюнкций: логическое произведение двух элементарных дизъюнкций разных рангов, из которых одна является общей частью другой, можно заменить дизъюнкцией имеющей меньший ранг.

Например:

$$(x_1 \vee \overline{x_2} \vee x_3 \vee \overline{x_4}) \cdot (x_1 \vee \overline{x_4}) = x_1 \vee \overline{x_4} .$$

Это правило является следствием закона дистрибутивности.

Примеры алгебраического преобразования формул логических функций:

а) упрощение формулы ЛФ

$$\begin{aligned} Y &= \overline{x_1} \overline{x_2} \overline{x_3} \vee \overline{x_1} \overline{x_2} x_3 \vee \overline{x_1} x_2 \overline{x_3} \vee \overline{x_1} x_2 x_3 = (\overline{x_1} \overline{x_2} \overline{x_3} \vee \overline{x_1} \overline{x_2} x_3) \vee (\overline{x_1} x_2 \overline{x_3} \vee \overline{x_1} x_2 x_3) = \\ &= \overline{x_1} \overline{x_2} \vee \overline{x_1} x_2 = \overline{x_1} \end{aligned}$$

б) преобразование формулы ЛФ к регулярному виду

$$\begin{aligned}
 Y &= \overline{\overline{\overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4}}} = \overline{\overline{\overline{x_1 x_2 x_3 x_4} \cdot (\overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4})}} = \\
 &\overline{\overline{\overline{x_1 x_2 x_3 x_4} \cdot \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4} \cdot \overline{x_1 x_2 x_3 x_4}}} = (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3} \vee \overline{x_4}) \cdot (\overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} \cdot \overline{x_4}) = \\
 &= \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4} = \overline{x_1 x_2 x_3 x_4}
 \end{aligned}$$

в) преобразование ДНФ ЛФ в СДНФ

$$\begin{aligned}
 Y = f(x_1, x_2, x_3) &= x_1 \vee x_2 \cdot \overline{x_3} = x_1 \cdot (x_2 \vee \overline{x_2}) \cdot (x_3 \vee \overline{x_3}) \vee x_2 \cdot \overline{x_3} \cdot (x_1 \vee \overline{x_1}) = x_1 \cdot x_2 \cdot x_3 \vee x_1 \cdot x_2 \cdot \overline{x_3} \vee \\
 &\vee x_1 \cdot \overline{x_2} \cdot x_3 \vee x_1 \cdot \overline{x_2} \cdot \overline{x_3} \vee x_1 \cdot x_2 \cdot x_3 \vee x_1 \cdot x_2 \cdot \overline{x_3}
 \end{aligned}$$

г) преобразование КНФ ЛФ в СКНФ

$$\begin{aligned}
 Y = f(x_1, x_2, x_3) &= x_1(x_2 \vee \overline{x_3}) = (x_1 \vee x_2 \cdot \overline{x_2} \vee x_3 \cdot \overline{x_3}) \cdot (x_2 \vee \overline{x_3} \vee x_1 \cdot \overline{x_1}) = (x_1 \vee x_2 \vee x_3) \cdot \\
 &\cdot (x_1 \vee x_2 \vee \overline{x_3}) \cdot (x_1 \vee \overline{x_2} \vee x_3) \cdot (x_1 \vee \overline{x_2} \vee \overline{x_3}) \cdot (x_1 \vee x_2 \vee \overline{x_3}) \cdot (\overline{x_1} \vee x_2 \vee \overline{x_3}) \cdot \\
 &(x_1 \vee x_2 \vee \overline{x_3}) \cdot (x_1 \vee \overline{x_2} \vee x_3) \cdot (x_1 \vee \overline{x_2} \vee \overline{x_3}) \cdot (\overline{x_1} \vee x_2 \vee \overline{x_3})
 \end{aligned}$$

Контрольные вопросы

- 1 Сформулировать законы и аксиомы булевой алгебры.
- 2 Сформулировать и доказать теоремы булевой алгебры.
- 3 Объяснить способы преобразования ДНФ в СДНФ и СКНФ.

Упражнение № 2

- 1 Упростить СДНФ и СКНФ ЛФ, приведенные в упражнении № 1.
- 2 Упростить СДНФ и СКНФ инверсных функций на примерах упражнения № 1.

3. Практическое занятие №3. Метод минимизации Квайна и Мак-Класки

Метод получения тупиковых дизъюнктивных нормальных форм ЛФ, разработанный Квайном и усовершенствованный Мак-Класки, включает четыре основных этапа:

- 1) нахождение всех простых импликант;
- 2) построение таблицы покрытий матрицы Квайна;
- 3) поиск минимального покрытия функции;
- 4) отыскание минимальной формы логической функции.

3.1 Нахождение всех простых импликант

Поиск простых импликант осуществляется последовательным применением операции склеивания к элементарным конъюнкциям СДНФ с целью понижения ранга.

Для удобства склеивания обычно все элементарные конъюнкции ДНФ кодируются n – разрядным троичным кодом: если переменная входит в ЭК в прямом виде, то в коде ей соответствует 1, в инверсном – 0, если переменная отсутствует – Z (или прочерк “-”). Число единиц и нулей в коде определяет его ранг.

Каждый этап выполняется последовательно по шагам.

1. На первом шаге исходное множество ЭК разбивается на группы с одинаковым числом единиц. Нулевая группа включает код, не содержащей единиц (если такой имеется). Первая группа включает коды, содержащие по одной единице, вторая – по две и т.д.

2. Затем попарно сравниваются между собой все коды соседних групп, т.е. коды нулевой группы “склеиваются” с кодами первой группы, коды первой группы – с кодами второй группы и так далее. Операция “склеивания” возможна лишь для пары кодов, различающихся только в одном разряде: пара “склеиваемых” кодов заменяется одним кодом меньшего ранга, имеющим Z (или прочерк “-”) в этом разряде.

Например пара кодов 0 1 0 Z и 1 1 0 Z заменяется интервалом Z 1 0 Z.

3 Все коды, участвующие в операции “склеивания”, отмечаются (например звездочкой “*” или буквой n), так как они поглощаются полученным кодом. Один и тот же код можно “склеивать” несколько раз. В результаты операции “склеивания”, проведенных над кодами групп r -ого ранга получаем производные группы $(r - 1)$ - ого ранга, коды которых подвергаются аналогично операции склеивания до тех пор, пока не останутся коды, не допускающие операции “склеивания”, т.е. неотмеченные коды. Совокупность неотмеченных кодов, соответствует совокупности простых импликант, образующих сокращенную дизъюнктивную нормальную форму ЛФ.

3.2 Построение таблицы покрытий матрицы Квайна

Таблица покрытий строится следующим образом.

Строки таблицы соответствуют простым импликантам (неотмеченным кодам), полученным на первом шаге первого этапа, а столбцы – элементарным конъюнкциям из СДНФ. На пересечение i -ой строки и j -ого столбца ставится 1, если i -я импликанта покрывает j -ю ЭК из СДНФ.

3.3 Поиск минимального покрытия функции

Для нахождения минимального покрытия функции необходимо удалить из таблицы лишние простые импликанты (строки). Для этого в методе Квайна и Мак-Класски применяется следующий алгоритм:

1) если в каком-либо столбце таблицы покрытий имеется только одна единица, то импликанта, стоящая в соответствующей строке является существенной (обязательной) и должна входить в минимальное покрытие, поскольку не используя её, невозможно покрыть все конституенты;

2) поглощение лишних столбцов (сжатие по столбцам). Из таблицы удаляется тот столбец, в который полностью входит в любой другой столбец. Если в таблице покрытий имеется такая пара столбцов a_i и a_j , что $a_i \subseteq a_j$, то столбец a_j удаляется, т.к. покрытие этого столбца может производиться путём покрытия оставшегося столбца;

3) поглощение лишних строк (сжатие по строкам). Если в таблице покрытий имеется такая пара строк a_i и a_j , что $a_j \subseteq a_i$, то строка a_j удаляется, т.к. она поглощается строкой a_i ;

4) последовательное применение двух преобразований (сжатие по столбцам и строкам).

3.4 Получение минимальной формы ЛФ

Простые импликанты, соответствующие строкам, которые входят в минимальное покрытие, соединяются знаками дизъюнкции и образуют минимальную дизъюнктивную нормальную форму булевой функции.

Рассмотрим применение метода Квайна и Мак-Класски на примере. Пусть ЛФ содержит восемь элементов четвёртого ранга:

$$Y = \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3} x_4 \vee \overline{x_1 x_2} x_3 x_4 \vee \overline{x_1} x_2 x_3 x_4 \vee \overline{x_1} x_2 x_3 \overline{x_4} \vee \overline{x_1} x_2 \overline{x_3} x_4 \vee \overline{x_1} \overline{x_2} x_3 x_4 \vee \overline{x_1} \overline{x_2} \overline{x_3} x_4$$

Процесс склеивания кодов представлен в таблице 3.1

Таблица 3.1 - Процесс склеивания кодов

Коды 4 – го ранга	Коды 3 – го ранга	Коды 2 - ранга
0 0 0 0 n	0 0 Z 0	
0 0 1 0 n	Z 0 0 0	
1 0 0 0 n	0 0 1 Z	
0 0 1 1 n	0 Z 1 1 n	
0 1 0 1 n	Z 0 1 1 n	
0 1 1 1 n	0 1 Z 1	Z Z 1 1
1 0 1 1 n	Z 1 1 1 n	Z Z 1 1
1 1 1 1 n	1 Z 1 1 n	

После склеивания неотмеченными остались шесть кодов:

0 0 Z 0, Z 0 0 0, 0 0 1 Z, 0 1 Z 1, Z Z 1 1, Z Z 1 1

Два одинаковых кода 2-ого ранга заменяются одним.

В таблице 3.2 представлена импликантная таблица и показан процесс поиска минимального покрытия ЛФ.

Таблица 3.2 - Процесс поиска минимального покрытия ЛФ

N	1	2	3	4	5	6	7	8
x_1	0	0	0	0	0	1	1	1
x_2	0	0	0	1	1	0	0	1
x_3	0	1	1	0	1	0	1	1
x_4	0	0	1	1	1	0	1	1
1	00Z0	1	1					
2	Z000	1				1		
3	001Z		1	1				
4	01Z1				1	1		
5	ZZ11			1		1	1	1

В таблице 3.2 столбцы 4, 6, 7 и 8 содержат по одной единице, поэтому соответственно импликанты строк 4, 2 и 5 являются существенным (обязательным) и включаются в искомую МДНФ.

Используя правило сжатия по столбцам, удалим столбец 5, т.к. он включает в себя столбец 4 и столбец 3. Аналогично – столбец 1, т.к. он включает в себя столбец 6 и столбец 2. Столбец 8, т.к. он включает в себя столбец 7.

После этих преобразований получим следующую таблицу импликант.

Таблица 3.3 – Таблица импликант

		N	
		2	3
	x ₁	0	0
	x ₂	0	0
	x ₃	1	0
	x ₄	0	0
1	0 0 Z 0	1	
3	0 0 1 Z	1	1

Используя правило сжатия по строкам, удалим строку 1, т.к. она входит в строку 3. При этом получим последнюю импликанту МДНФ 0 0 1 Z.

Таким образом определили состав МДНФ:

$$Y = \overline{x_2} \overline{x_3} \overline{x_4} \vee \overline{x_1} \overline{x_2} \overline{x_3} \vee \overline{x_1} \overline{x_2} x_4 \vee x_3 x_4$$

Контрольные вопросы

- 1 В чём заключается сущность метода Квайна и Мак-Класки?
- 2 Перечислить этапы минимизации ЛФ по методу Квайна и Мак-Класки.

Упражнение №3

- 1 Найти методом Квайна и Мак-Класки минимальную ДНФ ЛФ, полученную в упражнении №1.
- 2 Минимизировать СКНФ ЛФ, полученную при выполнении упражнения №1.

4. Практическое занятие №4. Минимизация логических функций по картам Карно

Минимизация ЛФ необходима для построения комбинационных схем цифровых автоматов минимальной сложности.

Минимизация ДНФ и КНФ по картам Карно представляет собой процесс, аналогичный минимизации ЛФ, представленных в аналитическом виде с использованием правил преобразования формул, рассмотренных на практическом занятии № 2, в частности, правил склеивания.

Известно, что каждая клетка карты Карно, заполненная единицей, соответствует элементарной конъюнкции r -го ранга. Двум соседним клеткам карты Карно, расположенным в одном столбце или строке, заполненными единицами, соответствуют ЭК, отличающиеся лишь в одном разряде. К этим клеткам можно применить правило склеивания. Область (покрытие), соответствующая этим двум клеткам, называется смежной и может быть аналитически описана ЭК ранга $(r-1)$, в которой отсутствует элемент (x) , меняющий своё значение при переходе из одной клетки в другую. Если покрытие включает в себя $2^1, 2^2$ или же любое другое число клеток, равное 2^m , то ЭК, описывающие это покрытие, содержит число элементов входного набора, равное $n - m$. Покрытие на карте Карно обводится контуром. Аналогично определяются контуры при минимизации КНФ.

Процесс минимизации ДНФ (КНФ) ЛФ может быть описан следующим алгоритмом:

- 1) определить множество возможных покрытий карты Карно;
- 2) из этого множества последовательно в порядке убывания m выделяем контурами покрытие до тех пор, пока не “покроем” все клетки карты Карно;
- 3) оптимизируем покрытие, руководствуясь следующим принципом: необходимо наименьшим числом контуров, каждый из которых охватывает как можно большую область смежных клеток, “покрыть” все клетки, содержащие единицы (или нули);
- 4) определим вид тупиковой ДНФ (КНФ). Для этого определяем дизъюнкцию (конъюнкцию) ЭК (ЭД) по следующему правилу: переменная x включается в ЭК (ЭД) в прямом виде, если эта переменная имеет значение 1 (0) на всех клетках области. Соответственно, переменная x включается в ЭК (ЭД) в инверсном виде, если эта переменная имеет значение 0 (1) на всех клетках области.

В результате минимизации возможно получение нескольких ТДНФ (ТКНФ) одинаковой сложности.

При минимизации частичных ЛФ на карте Карно первоначально в клетках, соответствующих неопределённым наборам ЛФ, ставится прочерк. При определении контуров покрытия в зависимости от необходимости эти клетки доопределяются единицами или нулями. В остальном процесс минимизации аналогичен процессу для полностью определенных ЛФ.

Изучение способов минимизации рассмотрим на примерах.

4.1. Построение минимальных ДНФ

1) Пусть СДНФ ЛФ имеет вид:

$$Y = \overline{x_1} \overline{x_2} \overline{x_3} \vee \overline{x_1} x_2 x_3 \vee x_1 x_2 x_3 \vee \overline{x_1} x_2 x_3$$

Составим карту Карно для данной функции.

$X_2 X_3$	00	01	11	10
X_1 0	0	0	1	0
1	1	1	1	0

Рисунок 4.1 – Карта Карно для примера 1

Тогда МДНФ данной функции будет иметь вид:

$$Y = \overline{x_1} \overline{x_2} \vee x_2 x_3$$

2) Пусть СДНФ ЛФ имеет вид:

$$Y = \overline{x_1} \overline{x_2} \overline{x_3} \overline{x_4} \vee \overline{x_1} \overline{x_2} x_3 x_4 \vee \overline{x_1} x_2 \overline{x_3} x_4 \vee \overline{x_1} x_2 x_3 \overline{x_4} \vee \overline{x_1} x_2 x_3 x_4 \vee x_1 \overline{x_2} \overline{x_3} \overline{x_4} \vee x_1 \overline{x_2} x_3 \overline{x_4} \vee x_1 \overline{x_2} x_3 x_4 \vee x_1 x_2 \overline{x_3} \overline{x_4} \vee x_1 x_2 \overline{x_3} x_4 \vee x_1 x_2 x_3 \overline{x_4} \vee x_1 x_2 x_3 x_4$$

Составим карту Карно для данной функции.

$X_3 X_4$	00	01	11	10
$X_1 X_2$ 00	1	0	1	1
01	0	1	1	1
11	1	1	0	0
10	1	0	0	1

Рисунок 4.2 – Карта Карно для примера 2

МДНФ данной функции имеет вид:

$$Y = \overline{x_2} \overline{x_4} \vee \overline{x_1} x_3 \vee \overline{x_1} x_2 x_4 \vee x_1 x_2 \overline{x_3}$$

4.2 Построение минимальных КНФ

1) Пусть СКНФ ЛФ имеет вид:

$$Y = (x_1 \vee x_2 \vee \overline{x_3}) \cdot (\overline{x_1} \vee x_2 \vee \overline{x_3}) \cdot (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3}) \cdot (\overline{x_1} \vee \overline{x_2} \vee x_3)$$

На рисунке 4.3 представлена карта Карно для данной СКНФ.

$X_2 X_3$	00	01	11	10
X_1				
0	1	0	1	1
1	1	0	0	0

Рисунок 4.3 – Карта Карно для примера 1

Минимизация СКНФ позволила получить следующую минимальную форму ЛФ:

$$Y = (\overline{x_1} \vee \overline{x_2}) \cdot (x_2 \vee \overline{x_3})$$

2) пусть СКНФ ЛФ имеет следующий вид:

$$Y = (x_1 \vee x_2 \vee x_3 \vee x_4) \cdot (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3} \vee \overline{x_4}) \cdot (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3} \vee \overline{x_4}) \cdot (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3} \vee \overline{x_4}) \cdot (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3} \vee x_4) \cdot (\overline{x_1} \vee x_2 \vee \overline{x_3} \vee \overline{x_4}) \cdot (\overline{x_1} \vee x_2 \vee \overline{x_3} \vee \overline{x_4})$$

На рисунке 4.4 представлена карта Карно для данной СКНФ.

$X_3 X_4$	00	01	11	10
$X_1 X_2$				
00	0	1	1	1
01	0	1	1	0
11	1	1	0	0
10	1	0	0	1

Рисунок 4.4 – Карта Карно для примера 2

Минимизация СКНФ позволила получить следующую минимальную форму ЛФ:

$$Y = (x_1 \vee x_3 \vee x_4) \cdot (\overline{x_2} \vee \overline{x_3} \vee x_4) \cdot (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3}) \cdot (\overline{x_1} \vee x_2 \vee \overline{x_4})$$

4.3 Минимизация не полностью определенных ЛФ

Пусть ЛФ задана списками единичных и нулевых наборов соответственно:

$$Y = \underset{\vee}{1} (4, 6, 7, 8, 13, 14),$$

$$Y = \underset{\wedge}{0} (0, 3, 5, 11, 15)$$

Покрывание карты Карно для построения МДНФ и МКНФ представлены на рисунке 4.5 “а” и “б” соответственно.

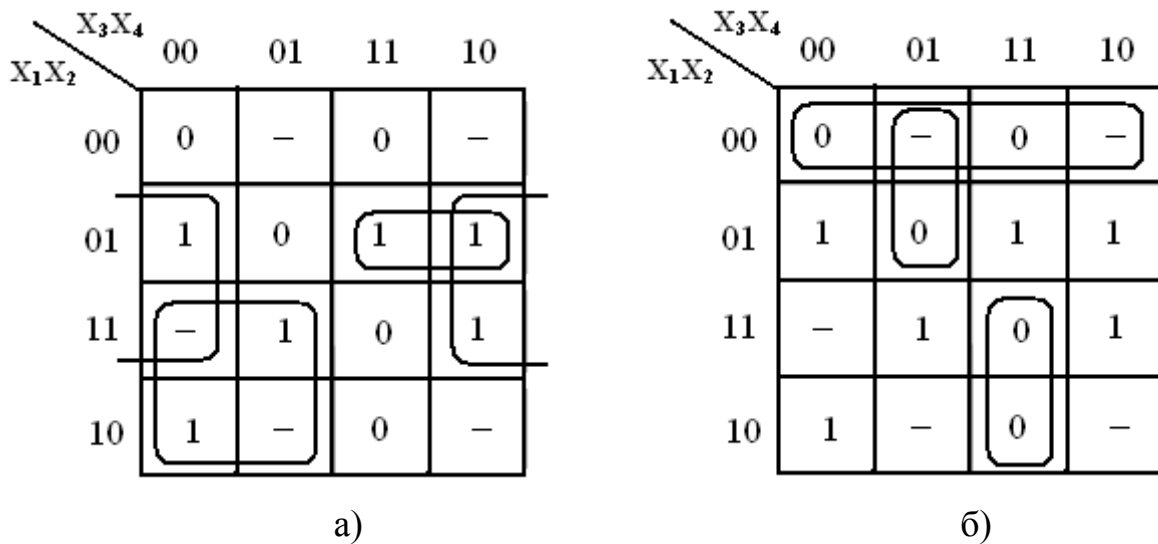


Рисунок 4.5 – Покрывание карты Карно

МДНФ и МКНФ ЛФ имеют соответственно следующий вид:

$$Y = x_2 \overline{x_4} \vee x_1 \overline{x_3} \vee \overline{x_1} x_2 x_3;$$

$$Y = (x_1 \vee x_2) \cdot (x_1 \vee x_3 \vee \overline{x_4}) \cdot (\overline{x_1} \vee \overline{x_3} \vee \overline{x_4})$$

Контрольные вопросы

1. Изложите основные правила минимизации ЛФ с применением карты Карно.
2. Укажите особенности минимизации не полностью определенных ЛФ.

Упражнение № 4

1. С помощью карты Карно минимизировать ЛФ, заданные в упражнении №1.
2. Минимизировать с применением карты Карно следующие пары ЛФ, заданные соответственно списками единичных и нулевых наборов:

1. (2, 5, 8, 10, 12), (1, 3, 6, 11, 15);
2. (0, 4, 8, 13, 14), (2, 3, 5, 7, 9, 11, 15);
3. (1, 2, 3, 7, 8, 9), (4, 5, 6, 10, 11, 12, 15);
4. (0, 5, 10, 15), (2, 7, 8, 13);
5. (0, 1, 5, 7, 8, 9, 14, 15), (4, 6, 12, 14);
6. (0, 2, 8, 10), (1, 3, 4, 6, 9, 12, 13, 14);
7. (3, 5, 7, 9, 11, 14), (0, 2, 4, 6, 12, 15);
8. (1, 2, 4, 5, 8, 9), (6, 7, 10, 12, 14, 15);
9. (2, 3, 11, 13, 15), (4, 5, 8, 10, 12, 14, 15);
10. (1, 3, 7, 8, 9, 11, 13), (4, 5, 6, 10, 12, 15);
11. (3, 7, 8, 9, 13, 14), (4, 5, 6, 10, 11, 12, 15);
12. (4, 5, 6, 10, 11, 12, 15), (1, 2, 3, 7, 8, 9);
13. (2, 3, 5, 7, 9, 11, 15), (0, 4, 8, 13, 14);
14. (4, 6, 12, 14), (0, 1, 5, 7, 8, 9, 14, 15);
15. (4, 5, 8, 10, 12, 14, 15), (2, 3, 11, 13, 15).

5 Практическое занятие №5. Формы задания конечных автоматов

Цифровой автомат (ЦА) - это устройство для преобразования дискретных данных в результате перехода из одного состояния в другое под воздействием входных сигналов и сохраняющее свое состояние при отсутствии последних.

Как правило, ЦА описывается следующим кортежем: $M = \{X, Y, S, \delta, \lambda, s_0\}$, где X, Y, S – соответственно множества входных, выходных значений и внутренних состояний ЦА.

$$X = \{x_1, x_2, x_3, \dots, x_n\}; Y = \{y_1, y_2, y_3, \dots, y_m\}; S = \{s_1, s_2, s_3, \dots, s_k\}; \quad (5.1)$$

где m, n, k – числа элементов множеств;

δ, λ – соответственно функция перехода из одного состояния в другое и функция выхода ЦА;

s_0 – начальное состояние ЦА.

Задать автомат – это значит определить все элементы выражения (5.1).

Если m, n, k конечны, то автомат называют конечным. Состояние ЦА определяется состоянием элементов памяти.

По закону функционирования или по виду выходной функции ЦА делятся на: автоматы первого рода (автоматы Мили) и автоматы второго рода.

Модель функционирования автомата Мили имеет следующий вид:

$$s(t) = \delta[s(t-1), x(t)], \quad y(t) = \lambda[s(t-1), x(t)], \quad (5.2)$$

где: $s(t)$ - состояние автомата в настоящий момент дискретного времени;
 $s(t-1)$ - состояние автомата в предыдущий момент. Если $t=0$, то $s(t-1)=s_0$;
 $x(t)$ - входной сигнал в текущий момент;
 δ - оператор (функция) формирования состояния;
 λ - оператор (функция) формирования выходного сигнала.

Таким образом, функционирование ЦА представляется совокупностью двух функций: функции перехода δ и функции выхода λ . При этом состояние $s(t)$ зависит от предыдущего состояния $s(t-1)$ и входного сигнала в данный момент времени, выходной сигнал в данный момент времени так же определяется предыдущим состоянием и входным сигналом в данный момент времени.

Закон функционирования ЦА 2-го описывается следующим образом:

$$s(t) = \delta[s(t-1), x(t)], \quad y(t) = \lambda[s(t), x(t)]. \quad (5.3)$$

Если функцию выхода в последнем выражении представить как:

$$y(t) = \lambda[s(t)], \quad (5.4)$$

то получим модель работы автомата Мура, являющегося одним из представителей ЦА 2-го рода.

Как видно из описаний автоматов, у ЦА Мили выходной сигнал имеется только тогда, когда есть входной сигнал, а у ЦА Мура выходной сигнал на текущий момент не зависит от входного сигнала.

Большинство аппаратно-программных средств вычислительной техники описывается автоматами Мили и Мура.

Для описания (задания) ЦА используются разнообразные средства, называемые языками, которые делятся на начальные и автоматные языки.

Среди начальных языков основными являются: язык логических

выражений, язык логических схем алгоритмов, язык граф–схем алгоритмов (ГСА).

В автоматных языках поведение ЦА задается путем явного описания функции переходов и выходов, в частности с использованием уточненных графов переходов и выходов, таблиц переходов и выходов, матриц переходов и выходов.

Рассмотрим задачи формирования табличной и графической форм задания конечных автоматов. В качестве исходных данных будем использовать ГСА микропрограмм.

По ГСА можно построить графы и таблицы переходов-выходов микропрограммного автомата, соответствующие автомату Мура или Мили. Для этого на ГСА нужно отметить (указать) символы состояний автомата. Существует несколько способов разметки ГСА ЦА. Рассмотрим один из них. Будем полагать, что автомат начинает работу с состояния s_0 , в котором он не вырабатывает никаких выходных сигналов и после выполнения микропрограммы снова оказывается в этом же состоянии. Затем автомат переходит в состояния, предписанные законом функционирования, и формирует микрокоманды из заданного множества Y . Момент окончания выполнения микропрограммы отмечается возвратом автомата в начальное состояние s_0 .

Отметка состояний на ГСА должна соответствовать закону функционирования автомата Мура или Мили, то есть выполняется различным образом.

Для автомата Мура выходные сигналы связаны только с состоянием автомата, поэтому каждой операторной вершине ГСА нужно поставить в соответствие одно из состояний автомата. Правило разметки состояний автомата на ГСА выглядит следующим образом:

- символом s_0 отмечаются начальная и конечная вершины ГСА, при этом символ состояния ставится рядом с вершиной;
- каждая операторная вершина отмечается единственным символом из списка: $s_1, s_2, s_3, s_4, s_5, \dots$; две и более различные операторные вершины не могут быть отмечены одинаковыми символами.

На рисунке 5.1 представлена ГСА, размеченная для автоматов Мура и Мили. В каждом такте автомат Мура, интерпретирующий данную микропрограмму, переходит из одного состояния в другое и выдаёт соответствующие выходные управляющие сигналы y_i . Так, при наличии входного сигнала $x_1 = 0$ автомат из состояния s_0 перейдет в состояние s_1 и выдаст выходной сигнал y_1 . В следующем такте работы под воздействием входного сигнала $x_2 = 1$ автомат из состояния s_1 перейдет в состояние s_3 с выдачей выходных сигналов y_2 и y_3 .

Если для реализации ГСА используется автомат Мили, то разметка граф-схемы алгоритма производится в следующем порядке:

- символом s_0 отмечается выход начальной и вход конечной вершины;
- символами: s_1, s_2, \dots - отмечаются входы вершин, следующие за операторными вершинами;
- входы двух различных вершин не могут быть отмечены одинаковыми символами;

входы вершины могут отмечаться только одним символом состояния.

На рисунке 5.1 разметка ГСА для автомата Мили представлена состояниями в скобках.

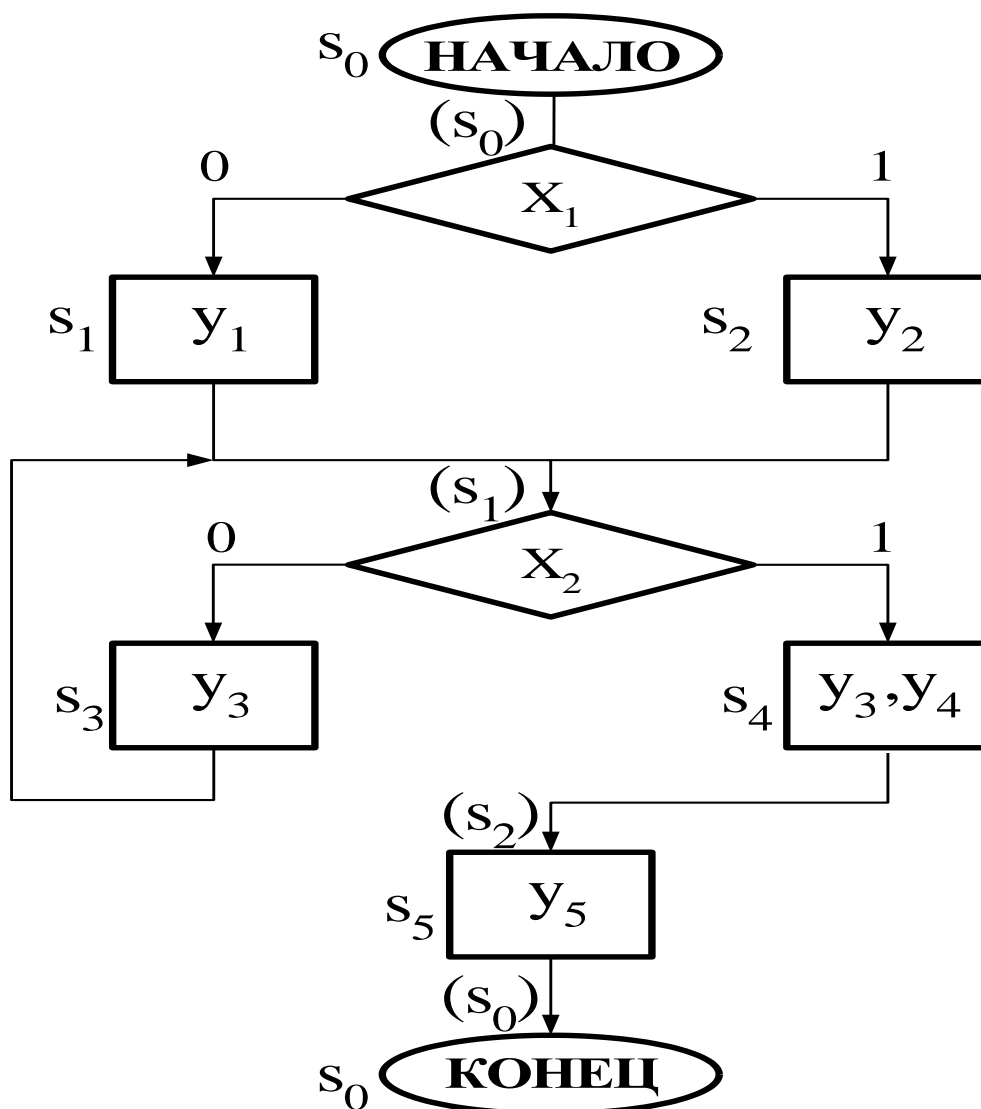


Рисунок 5.1 – Размеченная граф-схема алгоритма

Графы переходов состояний ЦА строятся с использованием размеченной ГСА. При этом состояния ЦА представляются вершинами графа. Переходы от одного состояния к другому изображаются направленными дугами. Значение входного сигнала, вызывающего этот переход из текущего состояния $s(t)$ в последующее $s(t+1)$, приписывается соответствующей дуге. Для автомата Мура значения выходных сигналов зависят только от состояния и поэтому приписываются соответствующей вершине. Таким образом, на графах отображаются обе характеристические функции автомата. Граф автомата Мура, построенный по ГСА, представлен на рисунке 5.2 “а”, а для автомата Мили – на рисунке 5.2 “б”.

При формировании графа для автомата Мили необходимо учитывать, что значения выходных сигналов $Y(t)$, определяемые значениями текущего состояния $s(t)$ и входных сигналов $x(t)$, ставятся в соответствие самой дуге. Если смена

состояний ЦА происходит без наличия входных сигналов (без условий), а по тактовому синхросигналу, то на соответствующей дуге перехода ставится прочерк.

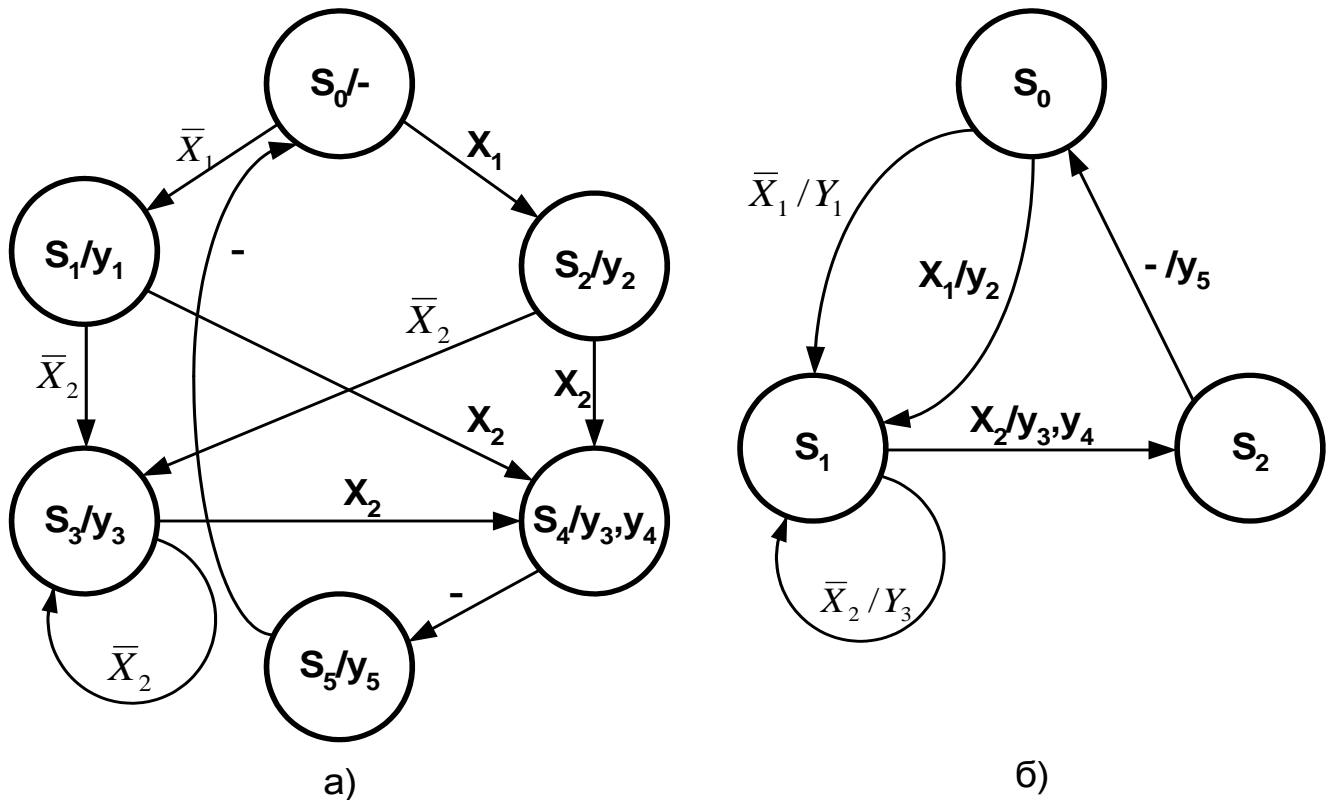


Рисунок 5.2 – Графы переходов: а) для автомата Мура, б) для автомата Мили

Таблицы переходов состояний и выходов ЦА строятся с учетом размеченной ГСА и графов переходов. Существует несколько форм табличного представления ЦА, представленные в таблицах 5.1 – 5.3.

В таблице 5.1 представлена совмещенная таблица переходов и выходов ЦА Мура и Мили, построенная в соответствии с выражениями (5.1)-(5.4).

Прочерки в таблице для входных сигналов обозначают безразличную реакцию ЦА на значение входного сигнала в данный момент времени. Прочерки для выходных сигналов означают смену перехода состояния ЦА без выдачи команды. Таблица 5.1 обычно используется в процессе синтеза ЦА и имеет детализированную структуру.

На этапе задания ЦА часто используется более компактная форма описания ЦА, представленная в таблицах 5.2 – 5.3.

Для автомата Мура в ячейках таблицы переходов-выходов для значений аргументов $x(t)$, и состояния $s(t-1)$ проставляются коды состояний $s(t)$. Код выходных сигналов $Y(t)$ представляются в отдельном столбце.

Для автомата Мили в ячейках таблицы переходов-выходов для каждой пары аргументов и предшествующего состояния ЦА проставляются коды текущих состояний и выходных сигналов.

Таблица 5.1 - Таблица переходов и выходов для автоматов Мура и Мили

X(t)		S(t-1)	S(t)	Y(t)
X ₁	X ₂			
Автомат Мура				
0	-	S ₀	S ₁	Y ₁
1	-	S ₀	S ₂	Y ₂
-	0	S ₁	S ₃	Y ₃
-	1	S ₁	S ₄	Y ₄
-	0	S ₂	S ₃	Y ₃
-	1	S ₂	S ₄	Y ₃ , Y ₄
-	1	S ₃	S ₄	Y ₃ , Y ₄
-	0	S ₃	S ₃	Y ₃
-	-	S ₄	S ₅	Y ₅
-	-	S ₅	S ₀	-
Автомат Мили				
0	-	S ₀	S ₁	Y ₁
1	-	S ₀	S ₁	Y ₂
-	0	S ₁	S ₁	Y ₃
-	1	S ₁	S ₂	Y ₃ , Y ₄
-	-	S ₂	S ₀	Y ₅

Таблица 2 – Совмещенная таблица переходов и выходов автомата Мура

Y_j	S_i	X_1X_2			
		00	01	10	11
-	S_0	S_1	S_1	S_2	S_2
Y_1	S_1	S_3	S_4	S_3	S_4
Y_2	S_2	S_3	S_4	S_3	S_4
Y_3	S_3	S_3	S_4	S_3	S_4
Y_3, Y_4	S_4	S_5	S_5	S_5	S_5
Y_5	S_5	S_0	S_0	S_0	S_0

Таблица 3 – Совмещенная таблица переходов и выходов автомата Мили

S_i	X_1X_2			
	00	01	10	11
S_0	S_1/Y_1	S_1/Y_1	S_1/Y_2	S_1/Y_2
S_1	S_1/Y_3	S_2/Y_3Y_4	S_1/Y_3	S_2/Y_3Y_4
S_2	S_0/Y_5	S_0/Y_5	S_0/Y_5	S_0/Y_5

Контрольные вопросы

- 1 Сформулируйте правила разметки ГСА для автоматов Мура и Мили.
- 2 Сформулируйте правила построения графов переходов для автоматов Мура и Мили.
- 3 Сформулируйте правила построения таблиц переходов-выходов для автоматов Мура и Мили.

Упражнение № 5

Для варианта задания по ГСА микропрограммы построить графы и таблицы переходов-выходов автоматов Мура и Мили.