

# ЦИФРОВЫЕ УСТРОЙСТВА И МИКРОПРОЦЕССОРЫ

Учебно-методическое пособие

## ОГЛАВЛЕНИЕ

Введение .....	4
1. Выбор варианта задания.....	4
2. Контрольная работа .....	5
2.1. Задания на контрольную работу .....	5
2.2. Методические указания и пояснения к работе .....	11
2.2.1. Системы счисления .....	11
2.2.2. Перевод чисел из одной системы в другую .....	13
2.2.3. Арифметические операции в различных ССч.....	17
2.2.4. Формы представления чисел .....	18
2.2.5. Машинные коды .....	23
2.2.6. Операции над числами в машинных кодах .....	25
2.2.7. Двоично-десятичная система кодирования.....	30
2.2.8. Переполнение разрядной сетки машины.....	32
2.2.9. Двоичные счетчики .....	34
2.2.10. Счетчики с произвольным основанием .....	38
2.3. Оформление контрольной работы.....	40
3. Расчетно-графическое задание .....	40
3.1. Варианты задания конечного автомата Мили .....	40
3.2. Синтез эквивалентного автомата Мура.....	45
3.3. Методические указания и пояснения к работе .....	45
3.3.1. Конечные автоматы.....	45
3.3.2. Структурный синтез конечных автоматов .....	47
3.3.3. Пример структурного синтеза автомата Мили .....	51
3.3.4. Переход от автомата Мили к автомату Мура .....	57
3.4. Оформление РГЗ .....	60
Библиографический список .....	62
Приложение .....	63

## ВВЕДЕНИЕ

Учебно-методическое пособие состоит из двух частей: заданий для контрольной работы и расчетно-графического задания (РГЗ).

**Контрольная работа** по курсу «Цифровые устройства и микропроцессоры» направлена на приобретение практических навыков по синтезу узлов цифровых вычислительных устройств и предполагает знание базовых разделов теоретического курса, включающего в себя: арифметические и логические основы цифровой техники, запись и схемную реализацию функций алгебры логики, комбинационные и последовательностные устройства. Контрольная работа состоит из заданий на следующие темы.

1. Арифметические и логические основы цифровой техники.
2. Синтез счетчика.

**Расчетно-графическое задание** является заключительным этапом курса и предполагает знание студентами основных положений теории конечных автоматов и структуры микропроцессоров. Выполнение РГЗ способствует приобретению практических навыков по синтезу узлов электронно-вычислительной аппаратуры и изучению всех базовых разделов теоретического курса.

Каждый студент выполняет индивидуальное задание, состоящее из двух частей:

- а) синтез конечного автомата Мили;
- б) синтез эквивалентного конечного автомата Мура.

## 1. ВЫБОР ВАРИАНТА ЗАДАНИЯ

Номер варианта задания выбирается по четырем последним цифрам индивидуального шифра. Следует перемножить эти пары цифр и принять за номер варианта две последние цифры полученного результата. Например, студент, имеющий шифр 30221218, выполняет вариант 16 ( $12 \cdot 18 = 216$ ).

Контрольная работа, выполненная не по своему варианту, не засчитывается.

## 2. КОНТРОЛЬНАЯ РАБОТА

### 2.1. Задания на контрольную работу

Контрольная работа содержит три задания.

1. Выполнение арифметических преобразований в различных системах счисления.
2. Реализация логической функции четырех переменных.
3. Синтез счетчика.

*Задание 1.* Выполнить следующие арифметические операции по преобразованию систем счисления (ССч):

а) исходное десятичное число (табл. 1), содержащее целую и дробную части, заданное в форме с фиксированной запятой (точкой), перевести в двоичную, в двоично-десятичную (код 8421) и в шестнадцатеричную ССч;

б) исходное шестнадцатеричное число (табл. 2) представить в двоично-десятичной (код 8421) и в десятичной ССч;

в) исходные десятичные числа (табл. 3) перевести в двоично-десятичную систему (код 8421), сложить алгебраически, предварительно представив отрицательные числа в дополнительном коде, а результат – в десятичной ССч;

г) исходное число из табл. 1 представить в форме с плавающей запятой (точкой) с точностью до четвертого десятичного знака после запятой.

Таблица 1

#### Варианты заданий

Предпоследняя цифра шифра	Последняя цифра варианта				
	0	1	2	3	4
0	371,135	258,341	236,712	148,124	124,261
1	157,913	126,145	234,611	791,314	236,910
2	245,678	346,112	478,245	145,124	235,113

Окончание табл. 1

Предпоследняя цифра шифра	Последняя цифра варианта				
	0	1	2	3	4
3	236,711	139,123	178,124	235,132	136,891
4	214,145	124,561	148,245	367,415	368,123
5	345,810	125,131	378,121	135,810	341,213
6	235,912	345,801	145,614	461,245	246,131
7	346,721	125,314	236,811	456,714	234,232
8	361,123	134,115	279,103	568,297	689,435
9	158,214	237,125	345,281	248,135	123,101
Предпоследняя цифра шифра	Последняя цифра варианта				
	5	6	7	8	9
0	156,814	136,112	145,114	135,121	138,315
1	346,213	245,314	345,213	241,413	124,101
2	458,112	247,211	269,131	679,451	691,314
3	136,101	126,141	257,101	345,131	245,913
4	368,219	678,121	157,123	679,141	129,141
5	167,891	129,213	367,112	789,131	246,211
6	234,135	135,121	378,121	246,135	236,712
7	456,811	124,411	459,145	126,812	451,121
8	236,711	141,415	145,610	367,913	589,123
9	467,159	246,191	691,214	357,149	571,532

Таблица 2

## Варианты заданий

Пред- последняя цифра шифра	Последняя цифра варианта									
	0	1	2	3	4	5	6	7	8	9
0	2FH	2EH	2DH	3CH	3AH	3BH	41H	57H	A8H	B7H
1	9CH	9DH	9EH	9FH	A0H	A1H	A2H	A3H	A4H	A5H
2	A6H	A7H	A8H	A9H	AAH	ABH	ACH	ADH	AEH	AFH
3	B0H	B1H	B2H	B3H	B4H	B5H	B6H	B7H	B8H	B9H
4	BAH	BBH	BCH	BDH	BEH	BFH	96H	97H	98H	99H
5	4AH	5BH	6CH	7DH	8EH	9FH	77H	78H	79H	85H
6	E1H	E0H	DFH	DEH	DDH	DCH	DBH	DAH	D9H	D8H
7	EBH	EAH	E9H	E8H	E7H	E6H	E5H	E4H	E3H	E2H
8	F5H	FF4H	F3H	F2H	F1H	F0H	EFH	EEH	EDH	ECH
9	FFH	FEH	FDH	FCH	FBH	FAH	F9H	F8H	F7H	F6H

Таблица 3

## Варианты заданий

Предпослед- няя цифра шифра	Последняя цифра варианта									
	0	1	2	3	4	5	6	7	8	9
0	-26	-54	56	-60	63-	59	-67	69	-73	-33
	54	26	-28	89	28	48	28	-24	89	85
1	-28	-56	57	-61	64	51	-68	69	-75	85
	56	28	-79	88	-27	49	27	-25	85	34
2	24	91	81	92	-56	-36	39	56	81	34
	-14	-74	45	-44	48	37	-49	-37	-41	-72
3	56	49	58	64	91	45	23	25	71	99
	-91	36	-28	-32	-34	-26	-28	75	-24	-26
4	-36	34	24	84	71	91	98	97	95	92
	89	-65	-46	-28	-28	-27	-25	-23	-24	-25

Предпоследняя цифра шифра	Последняя цифра варианта									
	0	1	2	3	4	5	6	7	8	9
5	49 -75	48 -28	47 -23	45 -22	43 -21	41 20	39 -18	37 -25	38 -85	43 -84
6	45 -23	47 -25	49 -99	51 -73	53 -48	43 -29	-43 29	-53 48	-51 73	-49 99
7	-33 48	-35 49	-37 51	-39 53	63 -39	39 -53	37 -51	37 -50	35 -47	39 -23
8	63 -23	65 -25	67 -24	68 -23	73 -28	-73 28	-68 23	-67 24	-65 25	-69 23
9	-27 55	-28 56	-29 57	-30 58	-31 59	33 -60	34 -61	35 -60	37 -62	39 -64

**Задание 2.** Логическая функция четырех переменных  $x_1, x_2, x_3, x_4$  имеет истинное значение на тех наборах входных переменных, которые эквивалентны десятичным числам, указанным в табл. 4, т. е. заданы в числовом виде. Требуется:

а) построить таблицу истинности функции четырех указанных переменных и записать для нее СДНФ и СКНФ;

б) минимизировать полученную функцию графическим методом Карно–Вейча и/или аналитически, используя законы алгебры логики;

в) на основании полученной минимальной (тупиковой) формы переключательной функции построить логическую схему на микросхемах произвольной серии.

Таблица 4

## Варианты заданий

Предпоследняя цифра шифра	Последняя цифра варианта				
	0	1	2	3	4
0	0, 3, 7, 10, 11, 13, 15	2, 5, 8, 12, 13, 14, 15	0, 2, 3, 6, 7, 12, 13	1, 4, 8, 10, 11, 12, 14, 15	0, 1, 2, 4, 5, 6, 11, 14
1	0, 1, 5, 7, 9, 10, 13	1, 2, 6, 8, 10, 14, 15	2, 3, 4, 6, 10, 11, 13	0, 7, 9, 10, 13, 14, 15	2, 3, 6, 7, 9, 10, 11, 12

Пред- последняя цифра шифра	Последняя цифра варианта				
	0	1	2	3	4
2	2, 4, 5, 6, 7, 8, 15	3, 4, 6, 8, 11, 12, 14	4, 7, 8, 10, 12, 14, 15	1, 4, 5, 6, 11, 12, 14,	0, 2, 3, 5, 8, 10, 13
3	2, 3, 6, 7, 10, 11, 12, 13	1, 3, 6, 9, 11, 12, 13	0, 1, 7, 8, 10, 12, 14	2, 3, 5, 10, 11, 12, 13	0, 1, 3, 6, 8, 9, 12, 13
4	0, 1, 10, 11, 12, 14, 15	0, 1, 2, 4, 5, 6, 11, 14	1, 4, 8, 10, 11, 12, 14, 15	0, 3, 6, 7, 8, 14, 15	1, 3, 6, 8, 9, 12, 13
5	3, 4, 5, 6, 7, 8, 9, 11	1, 2, 5, 7, 10, 13, 15	0, 3, 7, 8, 11, 12, 14, 15	1, 3, 5, 6, 8, 9, 10, 13	0, 3, 4, 10, 12, 13, 14, 15
6	2, 3, 5, 7, 9, 10, 11	3, 4, 5, 6, 7, 8, 9, 11	0, 1, 4, 5, 6, 11, 14	1, 4, 6, 11, 12, 14, 15	0, 2, 4, 6, 11, 13, 15
7	0, 2, 4, 6, 7, 9, 13	0, 1, 2, 5, 7, 9, 13, 14	2, 3, 6, 7, 8, 11, 12	0, 4, 5, 6, 7, 9, 14	2, 3, 4, 5, 12, 13, 14
8	3, 6, 10, 11, 12, 13, 14, 15	1, 3, 4, 6, 7, 8, 11, 15	0, 2, 7, 8, 9, 10, 15	2, 5, 6, 8, 9, 10, 11, 12	4, 6, 8, 9, 11, 13, 15
9	1, 5, 8, 10, 11, 12, 14, 15	0, 2, 3, 7, 9, 11, 12, 15	2, 3, 4, 5, 8, 10, 11	0, 2, 4, 8, 10, 13, 15	1, 2, 3, 4, 5, 9, 10, 13

**Задание 3.** По своему номеру варианта из табл. 5 выписать исходные данные для расчета: число разрядов счетчика, модуль счета, тип триггера и серию микросхем. По последней цифре номера варианта задания из табл. 6 выписать последовательность кодовых комбинаций, начиная с номера строки начала счета. Количество кодовых комбинаций равно модулю счета (М). Эта последовательность является основой для синтеза счетчика.

Если у вас указаны два типа триггера, то два младших разряда выполняются на первом типе триггера, а остальные – на втором.

Если число разрядов счетчика равно 4, а в табл. 5 указаны пятиразрядные двоичные числа, то левый (старший) бит кода отбрасывается.

Если номер варианта превышает число 29, то из номера следует вычесть 30.

Таблица 5

## Варианты заданий для синтеза счетчика

Номер варианта задания	Число разрядов (шт.)	Модуль счета (М)	Номер строки начала счета	Тип триггера	Серия ИМС
0	4	11	0	JK, D	155
1	5	17	0	T	555
2	4	12	1	JK	1533
3	5	18	1	T	531
4	4	13	2	RS	561
5	5	19	1	D	564
6	4	14	3	RS	155
7	5	20	0	T, D	555
8	4	11	4	JK, T	1533
9	5	17	1	D, T	531
10	5	18	2	T, D	561
11	4	12	5	JK, D	564
12	5	19	1	T	155
13	4	13	6	JK	555
14	5	20	0	D	1533
15	4	14	0	T, JK	531
16	5	17	1	D	561
17	4	11	1	RS	564
18	5	18	2	T, D	155
19	4	12	2	JK, T	555
20	4	13	3	RS, T	1533
21	5	19	0	T, D	531
22	4	14	4	D, JK	561
23	5	20	0	T	155
24	4	11	5	T, RS	555
25	5	17	2	D	1533
26	4	12	6	RS	531
27	5	18	1	D	561
28	4	13	0	D, JK	555
29	5	19	1	T	1533

Таблица 6

**Варианты смены состояний счетчика**

Вариант Номер строки	Последняя цифра номера варианта задания									
	0	1	2	3	4	5	6	7	8	9
0	00000	00010	00100	00110	01000	01010	01100	01110	10000	10010
1	00001	00011	00101	00111	01001	01011	01101	01110	10001	10011
2	00010	00100	00110	01000	01010	01100	01110	10000	10010	10100
3	00011	00101	00111	01001	01011	01101	01111	10001	10011	10101
4	00100	00110	01000	01010	01100	01110	10000	10010	10100	10110
5	00101	00111	01001	01011	01101	01111	10001	10011	10101	10111
6	00110	01000	01010	01100	01110	10000	10010	10100	10110	11000
7	00111	01001	01011	01101	01111	10001	10011	10101	10111	11001
8	01000	01010	01100	01110	10000	10010	10100	10110	11000	11010
9	01001	01011	01101	01111	10001	10011	10101	10111	11001	11011
10	01010	01100	01110	10000	10010	10100	10110	11000	11010	11100
11	01011	01101	01111	10001	10011	10101	10111	11001	11011	11101
12	01100	01110	10000	10010	10100	10110	11000	11010	11100	11110
13	01101	01111	10001	10011	10101	10111	11001	11011	11101	11111
14	01110	10000	10010	10100	10110	11000	11010	11100	11110	00001
15	01111	10001	10011	10101	10111	11001	11011	11101	11111	00011
16	10000	10010	10100	10110	11000	11010	11100	11110	00010	00101
17	10001	10011	10101	10111	11001	11011	11101	11111	00100	00111
18	10010	10100	10110	11000	11010	11100	11110	00001	00110	01000
19	10011	10101	10111	11001	11011	11101	11111	00011	00111	01001

**2.2. Методические указания и пояснения к работе***2.2.1. Системы счисления*

Система счисления (ССч) – это совокупность правил записи чисел цифровыми знаками. ССч бывают *позиционными* и *непозиционными* (например, римская ССч).

В вычислительной технике используются только позиционные системы счисления.

Число в любой позиционной ССч можно представить в виде последовательности цифр:

$$A = a_n a_{n-1} \dots a_1 a_0, b_{-1} b_{-2} \dots b_{-k},$$

где  $a_i, b_i$  – цифры данной системы счисления;

или в виде формулы разложения:

$$A = a_n p^n + a_{n-1} p^{n-1} + \dots + a_2 p^2 + a_1 p^1 + a_0 p^0 + \\ + b_{-1} p^{-1} + b_{-2} p^{-2} + \dots b_{-k} p^{-k},$$

где  $p$  – основание системы счисления (количество различных цифр в ССч);  $p^i$  – вес единицы данного разряда.

В ЭВМ используются ССч с основаниями  $p = 10, 2, 8, 16$ .

Рассмотрим эти системы счисления.

**Десятичная ССч.**  $p = 10$ . Разрешенные цифры (0, 1, 2, 3, 4, 5, 6, 7, 8, 9). Число можно представить так:

$$A_{10} = 247,56_{10} = 2 \cdot 10^2 + 4 \cdot 10^1 + 7 \cdot 10^0 + 5 \cdot 10^{-1} + 6 \cdot 10^{-2}.$$

Весы соседних разрядов влево и вправо от запятой различаются в десять раз ( $p = 10$ ):

$$\dots 1000 \ 100 \ 10 \ 1, \ 1/10 \ 1/100 \ 1/1000 \dots$$

**Двоичная ССч.**  $p = 2$ . Разрешенные цифры (0, 1). Число представляется так:

$$A_2 = 101110,101_2 = 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + \\ + 1 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} = 46,625_{10}.$$

Весы соседних разрядов влево и вправо от запятой различаются в два раза ( $p = 2$ ):

$$\dots 32 \ 16 \ 8 \ 4 \ 2 \ 1, \ 1/2 \ 1/4 \ 1/8 \dots$$

**Восьмеричная ССч.**  $p = 8$ . Разрешенные цифры (0, 1, 2, 3, 4, 5, 6, 7). Число представляется так:

$$\begin{aligned} A_8 = 125,46_8 &= 1 \cdot 8^2 + 2 \cdot 8^1 + 5 \cdot 8^0 + 4 \cdot 8^{-1} + 6 \cdot 8^{-2} = \\ &= 64 + 16 + 5 + 0,5 + \frac{6}{64} = 85,6_{10}. \end{aligned}$$

Веса соседних разрядов влево и вправо от запятой различаются в восемь раз ( $p = 8$ ):

$$\dots 4096 \ 512 \ 64 \ 8 \ 1, \ 1/8 \ 1/64 \ 1/512 \dots$$

**Шестнадцатеричная ССч.**  $p = 16$ . Разрешенные цифры (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F). Число представляется так:

$$\begin{aligned} A_{16} = 2AF,C_{16} &= 2 \cdot 16^2 + 10 \cdot 16^1 + 15 \cdot 16^0 + 12 \cdot 16^{-1} + 4 \cdot 16^{-2} = \\ &= 512 + 160 + 15 + \frac{12}{16} + \frac{4}{256} = 687,76_{10}. \end{aligned}$$

Веса соседних разрядов влево и вправо от запятой различаются в шестнадцать раз ( $p = 16$ ):

$$\dots 4096 \ 256 \ 16 \ 1, \ 1/16 \ 1/256 \ 1/4096 \dots$$

Вполне очевидно, что для записи одного и того же числа в разных системах счисления требуется разное количество разрядов. Основание ССч во всех системах записывается одинаково: 10.

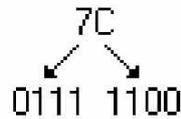
### 2.2.2. Перевод чисел из одной системы в другую

Перевод целых чисел и правильных дробей выполняется по разным правилам. В действительном числе переводят отдельно целую и дробную части.

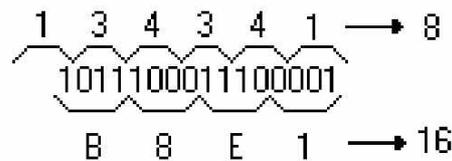
**Перевод целых чисел.** Для перевода необходимо исходное число разделить на основание новой ССч до получения целого остатка, являющегося младшим разрядом числа в новой системе счисления. Полученное частное снова делим на основание – и так до тех пор, пока частное не станет меньше нового основания. Все операции выполняются в исходной ССч.



а каждую триаду представляют восьмеричной цифрой. Перевод  $16 \rightarrow 2$  выполняют с помощью тетрад:

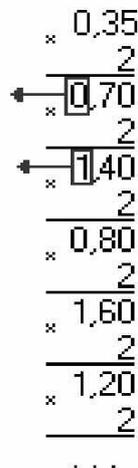


Перевод из двоичной ССч в восьмеричную и шестнадцатиричную (и обратно) выполняется так же:



**Перевод правильных дробей.** Этот перевод выполняется умножением исходного числа на основание новой ССч. Целая часть произведения является старшим разрядом числа в новой системе. Дробную часть произведения снова умножают на основание ССч и т. д. Все операции выполняют в исходной ССч.

Перевод:  $10 \rightarrow 2 \quad A = 0,35_{10} = 0,01011:$



В общем случае перевод правильных дробей бесконечен. Число разрядов в новой системе можно найти исходя из одинаковой точности представления чисел в разных ССч.

Одинаковая точность – одинаковые веса младших разрядов чисел

$$P^{-n_p} = q^{-n_q},$$

где  $p$  и  $q$  – основания старой и новой ССч соответственно.

Например, для десятичного числа  $0,35$  вес младшего разряда  $1/100 = 0,01$ , а для двоичного числа  $0,01011$  вес младшего разряда  $1/32 \cong 0,03$ .

Берем логарифм по основанию  $p$ :

$$-n_p = -n_q \log_p q, \text{ откуда находим } n_q = \frac{n_p}{\log_p q}.$$

Тогда при  $p = 10, q = 2$  получаем

$$n_2 = \frac{n_{10}}{\lg 2} = \frac{n_{10}}{0,3}.$$

Если  $n_{10} = 2$ , то в новой ССч число разрядов равно

$$n_2 = \frac{2}{0,3} = 7,$$

а при  $n_{10} = 3$ :  $n_2 = \frac{3}{0,3} = 10$ . Для восьмеричной и шестнадцатеричной ССч имеем

$$n_8 = \frac{n_{10}}{\lg 8} = \frac{n_{10}}{0,9}; \quad n_{16} = \frac{n_{10}}{1,2}.$$

Из десятичной в восьмиричную ССч переходим по тому же правилу:

$$\begin{aligned} 10 \rightarrow 8, \quad A = 0,35_{10} &= 0,263_8 = 2 \cdot 8^{-1} + 6 \cdot 8^{-2} + 3 \cdot 8^{-3} = \\ &= 0,25 + \frac{6}{64} + \frac{3}{256} \cong 0,35. \end{aligned}$$

Аналогично можно делать перевод в любую ССч. Перевод из  $8 \rightarrow 2 \rightarrow 16$  (и обратно) выполняется с помощью триад и тетрад, которые отмеряются от запятой.

### 2.2.3. Арифметические операции в различных ССч

Арифметические операции в различных ССч выполняются на основании таблиц сложения и умножения:

$$\begin{array}{ll} 0+0=0 & 0*0=0 \\ 0+1=1 & 0*1=0 \\ 1+0=1 & 1*0=0 \\ 1+1=10 & 1*1=1 \end{array}$$

**Двоичная ССч.** Возьмем два десятичных числа и выполним сложение, вычитание и умножение:

$$A = 5_{10} = 101_2;$$

$$B = 3,5_{10} = 11,1_2 = 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} = 2 + 1 + 0,5 = 3,5;$$

$$\begin{array}{l} A+B=1010 \\ \underline{11,1} \\ 1000,1 \end{array} = 1 \cdot 2^3 + 1 \cdot 2^{-1} = 8,5_{10}; \quad \begin{array}{l} A-B=1010 \\ \underline{11,1} \\ 001,1 \end{array} = 1 \cdot 2^0 + 1 \cdot 2^{-1} = 1,5_{10};$$

$$\begin{array}{l} A*B=101 \\ \underline{11,1} \\ 101 \\ 101 \\ \underline{101} \\ 10001,1 \end{array} = 1 \cdot 2^4 + 1 \cdot 2^0 + 1 \cdot 2^{-1} = 17,5_{10}.$$

**Восьмеричная ССч.** Выполним сложение, вычитание и умножение взятых нами чисел:

$$A = 5_{10} = 5_8;$$

$$B = 3,5_{10} = 3,4_8 = 3 \cdot 8^0 + 4 \cdot 8^{-1} = 3 + \frac{4}{8} = 3,5;$$

$$\begin{array}{l} A+B=5,0 \\ \underline{3,4} \\ 10,4 \end{array} = 1 \cdot 8^0 + 4 \cdot 8^{-1} = 8,5_{10}; \quad \begin{array}{l} A-B=5,0 \\ \underline{3,4} \\ 1,4 \end{array} = 1 \cdot 8^0 + 4 \cdot 8^{-1} = 1,5_{10};$$

$$\begin{array}{r}
 A \cdot B = 5,0 \\
 \underline{3,4} \\
 3,4 \\
 \underline{5} \\
 21,4
 \end{array}
 = 2 \cdot 8^1 + 1 \cdot 8^0 + 4 \cdot 8^{-1} = 17,5$$

Теперь в ССч с основанием  $P=16$ :

$$A = 5_{10} = 5_{16};$$

$$B = 3,5_{10} = 3,8_{16};$$

$$\begin{array}{r}
 A+B=5,0 \\
 \underline{3,8} \\
 8,8
 \end{array}
 = 8 \cdot 16^0 + 8 \cdot 16^{-1} = 8,5; \quad
 \begin{array}{r}
 A \cdot B = 5,0 \\
 \underline{3,8} \\
 1,8_{16} = 1,5_{10}
 \end{array};$$

$$\begin{array}{r}
 A \cdot B = 3,8 \\
 \underline{5} \\
 11,8
 \end{array}
 = 1 \cdot 16^1 + 1 \cdot 16^0 + 8 \cdot 16^{-1} = 16 + 1 + 0,7 = 17,5.$$

Надо помнить, что при переносе в старший разряд уходит число единиц, равное основанию ССч, а при заеме из старшего разряда приходит число единиц, так же равное основанию ССч.

#### 2.2.4. Формы представления чисел

Наименьшая мера информации, используемая в цифровой технике, – 1 бит (один двоичный разряд). Производными от этой единицы являются:

$$1 \text{ байт} = 8 \text{ бит};$$

$$1 \text{ К} = 2^{10} \text{ бит} = 1024 \text{ бит (кило)};$$

$$1 \text{ М} = 2^{10} \text{ К} = 1024 \text{ К (мега)};$$

$$1 \text{ Г} = 2^{10} \text{ М} = 1024 \text{ М (гига)}.$$

Наряду с этим используются и другие единицы – слово, полуслово, двойное слово (слово может равняться 1, 2, 4, 8 или иному числу байт), поле (до 256 байт).

В ЭВМ используются две формы представления чисел: с фиксированной запятой (естественная форма) и с плавающей запятой (показательная форма).

**Форма с фиксированной запятой.** Используется для записи целых чисел. Имеется две разновидности: знаковые и беззнаковые числа (рис. 2.1).



Рис. 2.1. Форматы целых чисел

Минимальное по модулю число, которое может быть представлено в этом формате: 0 и 1.

Максимальное число равно

$$A_{\max} = 2^{15} - 1 = 32767_{10} \quad (\text{со знаком});$$

$$2^{16} - 1 = 65535_{10} \quad (\text{без знака}).$$

**Форма с плавающей запятой.** Число представляют в виде

$$N = \pm m p^{\pm q},$$

где  $m$  – мантисса числа – правильная дробь;  $p$  – основание ССч;  $q$  – порядок числа.

Если мантисса находится в пределах  $\frac{1}{p} \leq m < 1$ , то говорят, что

число *нормализовано*:

$$0,26 \cdot 10^{15} \text{ – нормализованное число};$$

$$0,026 \cdot 10^{16} \text{ – не нормализованное число}.$$

Нормализацию чисел машина выполняет автоматически и работает только с нормализованными числами.

Классический формат числа с плавающей запятой представлен на рис. 2.2.



Рис. 2.2. Формат с плавающей запятой

Здесь запятая в мантиссе фиксирована перед старшим разрядом, а порядок – целое число (запятая фиксирована после младшего разряда). В разрядной сетке запятые нигде не стоят, они подразумеваются в определенном месте.

Максимальное число, которое может быть записано в этом формате:

$$A \cong 1 \cdot 2^{(2^6-1)} \cong 2^{63} \cong 10^{19}, \text{ так как } 10^x = 2^{63},$$

откуда  $X = 63 \lg 2 = 63 \cdot 0,3 \cong 19$ .

Минимальное нормализованное число

$$A_{\min} = 2^{-1} \cdot 2^{-(2^6-1)} = 2^{-1} \cdot 2^{-63} = 2^{-64} \cong 10^{-19}.$$

В этом формате есть одно замечательное число – нуль. Различают *истинный нуль* – когда во всех разрядах мантиссы стоят нули, и *машинный (нормализованный) нуль*, когда число меньше минимально представимого в данном формате, он может быть как положительным, так и отрицательным (рис. 2.3).

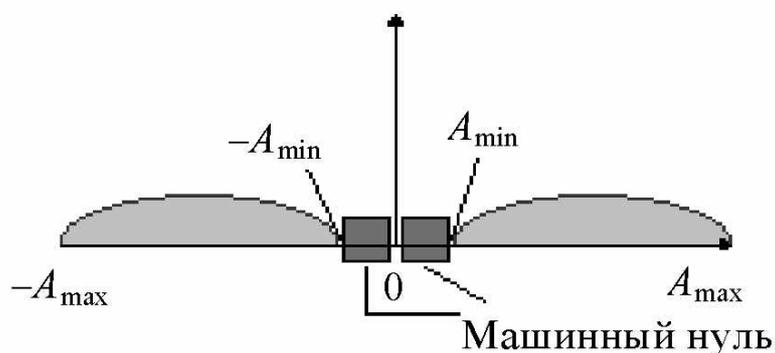


Рис. 2.3. Диапазон чисел формата с плавающей запятой

Существует много различных форматов с плавающей запятой. Это осложняет перевод программ с одной машины на другую, поэтому в 1985 году был принят международный стандарт IEEE-754, который оговаривает четыре формата с плавающей запятой.

**Базовый одинарный формат.** Слово длиной 4 байта, в котором используется смещенный порядок (характеристика) числа, дано на рис. 2.4.

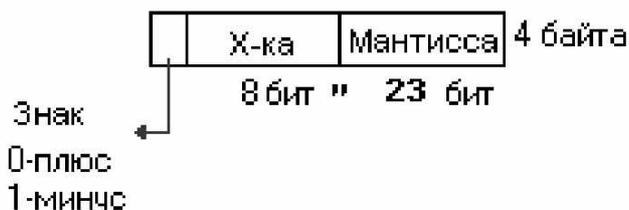


Рис. 2.4. Базовый одинарный формат с плавающей запятой

Запятые для характеристики и мантиссы фиксированы. Характеристика – целое число без знака, для нее запятая расположена после младшего разряда. Для мантиссы запятая расположена перед старшим разрядом.

$X = 128 + q$  – смещенный порядок ( $2^n - 1 = 255$  – максимальное число в 8 битах). Характеристика числа всегда положительна:

$$X \geq 0,$$

$$q = 127, X = 255,$$

$$q = -128, X = 0.$$

Это упрощает выполнение арифметических операций, так как не надо проверять знак порядка. В этом формате

$$A_{\max} \approx 1 \cdot 2^{127} \approx 10^{38},$$

$$10^x = 2^{127},$$

$$X = 127 \lg 2 = 127 \cdot 0,3 = 38.$$

Поэтому диапазон чисел такой:  $N = \pm 10^{\pm 38}$ .

По сравнению с классическим форматом диапазон чисел значительно шире, но мантисса числа на 1 бит меньше. Поэтому точность представления должна быть хуже, но так как число всегда нормализовано, то первую единицу после запятой не хранят, а подразумевают, поэтому точность чисел такая же, как в классическом формате. В этом формате используется так называемая *скрытая единица мантиссы*.

**Базовый двойной формат.** Здесь слово длиной 8 байт (рис. 2.5), смещение порядка составляет 1024.

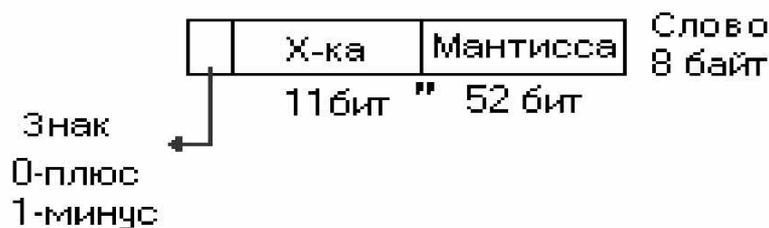


Рис. 2.5. Базовый двойной формат

Характеристика  $X = 1024 + q$ .

Порядок может находиться в пределах  $-1024 \leq q \leq 1023$ .

Диапазон чисел следующий:

$$A_{\max} = 1 \cdot 2^{1024};$$

$$10^x = 2^{1024};$$

$$X = 1024 \lg 2 = 1024 \cdot 0,3 = 308;$$

$$N \cong \pm 10^{\pm 308}.$$

В базовых форматах значение характеристики, равное нулю, соответствует нулевому числу, а значение характеристики, равное максимуму, соответствует бесконечности.

Мантисса длиной 24 бита соответствует точности представления числа 6–7 десятичных цифр. Мантисса длиной 52 бита соответствует точности представления 16–17 десятичных цифр.

Имеются также и расширенные форматы, но их мы не рассматриваем.

### 2.2.5. Машинные коды

Независимо от формы записи чисел (с фиксированной или плавающей запятой), все числа в ЭВМ представляются в виде специальных кодов – прямого, обратного или дополнительного.

Прямой код используется для хранения чисел в памяти и выполнения операции умножения.

Обратный и дополнительный коды используются для сложения положительных и отрицательных чисел.

Рассмотрим машинные коды на примере чисел с фиксированной запятой.

*Прямой код:*

$$[A]_{\text{пр}} = 0 \ a_n a_{n-1} \dots a_1 a_0, \ A \geq 0;$$

$$[A]_{\text{пр}} = 1 \ a_n a_{n-1} \dots a_1 a_0, \ A \leq 0.$$

**Знак «плюс» кодируют нулем, а знак «минус» – единицей. Знак числа обязательно должен быть в любом машинном коде, например:**

Число	Прямой код
+1101	01101
-1101	11101
+0,1101	01101
-0,1101	11101
-0,0000	10000
+0,0000	00000

Запятая в коде не пишется. Число нуль в прямом коде имеет двойное изображение – положительное и отрицательное.

*Обратный код:*

$$[A]_{\text{обр}} = 0 \ a_n a_{n-1} \dots a_1 a_0, \ A \geq 0;$$

$$[A]_{\text{обр}} = 1 \ \bar{a}_n \bar{a}_{n-1} \dots \bar{a}_1 \bar{a}_0, \ A < 0,$$

где  $\bar{a}_i = 1 - a_i$  – дополнение числа до 1 (инверсия разрядов двоичного числа), например:

Число	Обратный код
+1101	01101
-1101	10010
-0,1101	10010
+0,0000	00000

*Дополнительный код:*

$$[A]_{\text{доп}} = 0 \ a_n a_{n-1} \dots a_1 a_0, \quad A \geq 0;$$

$$[A]_{\text{доп}} = 1 \ \bar{a}_n \bar{a}_{n-1} \dots \bar{a}_1 \bar{a}_0 \ + 00 \dots 01, \quad A < 0,$$

где  $\bar{a}_i = 1 - a_i$  – дополнение числа до 1 (инверсия разрядов двоичного числа).

Дополнительный код числа – это обратный код плюс единица в младший разряд, например:

Число	Дополнительный код
+1101	01101
-1101	10011
-1100	10100

Дополнительный код правильной дроби – это дополнение числа до основания системы счисления.  $A + [A]_{\text{доп}} = 10$ , где 10 – основание ССч.

Дополнительный код  $n$ -разрядного целого отрицательного числа есть результат вычитания этого числа из единицы с  $(n + 1)$  нулями. Так, для числа  $A = -1101$  ( $n = 4$ )  $[A]_{\text{доп}} = 100000 - 1101 = 10011$ .

Для положительных чисел прямой, обратный и дополнительный коды совпадают.

## 2.2.6. Операции над числами в машинных кодах

### Операции с фиксированной запятой

**Сложение чисел.** В вычислительной технике, благодаря машинным кодам, операция вычитания заменяется операцией сложения с числом обратного знака:

$$A - B = A + (-B).$$

**Сложение в обратном коде.** Пусть даны два числа  $A$  и  $B$ . Надо найти их сумму:

$$A = -0,1101;$$

$$B = +0,0010.$$

Выполним сложение в обратном коде. При этом биты знака участвуют в сложении наравне со значащими разрядами:

$$[A]_{\text{обр}} = +10010;$$

$$[B]_{\text{обр}} = 00010;$$

$$[C]_{\text{обр}} = \overline{10100}.$$

Ответ:  $C = -0,1011$ .

Теперь сложим два других числа:

$$A = -0,1101$$

$$B = -0,0010$$

$$[A]_{\text{обр}} = +10010$$

$$[B]_{\text{обр}} = 11101$$

$$\begin{array}{r} 1 + 01111 \\ \hline \phantom{1} \phantom{+} \phantom{0} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \\ \phantom{1} \phantom{+} \phantom{0} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \end{array}$$

При сложении в обратном коде единица переноса из старшего (знакового) бита добавляется в младший разряд результата – имеет место

так называемый циклический перенос. Результат сложения  $[C]_{\text{обр}} = 10\ 000$ , а ответ  $C = -0,1111$ .

**Сложение в дополнительном коде.** Выполним сложение тех же чисел:

$$\begin{aligned}
 A &= -0,1101 \\
 B &= +0,0010 \\
 [A]_{\text{доп}} &= 10011 \\
 [B]_{\text{доп}} &= 00010 \\
 [C]_{\text{доп}} &= \underline{10101} \text{ дополнительный код результата} \\
 [C]_{\text{обр}} &= \overline{10100}
 \end{aligned}$$

Теперь возьмем другие числа:

$$\begin{aligned}
 A &= -0,1101 \\
 B &= -0,0010 \\
 [A]_{\text{доп}} &= 10011 \\
 [B]_{\text{доп}} &= 11110 \\
 [C]_{\text{доп}} &= 1\leftarrow 10001
 \end{aligned}$$

В дополнительном коде единица переноса из знакового бита отбрасывается. Тогда  $[C]_{\text{обр}} = 10\ 000$ , а ответ равен  $C = -0,1111$ .

При сложении обязательно выравнивание разрядов слагаемых (не кодов!) нулями. Для отрицательных чисел эти выравнивающие нули превращаются в единицы при инвертировании. Знаки чисел (крайние левые биты кодов) обязательно находятся один под другим.

Аналогично складывают и целые числа. Например, нужно сложить в разрядной сетке 1 байт два числа  $C = A + B$ , где  $A = +9_{10}$  и  $B = -7_{10}$ . Разместим их в фиксированной разрядной сетке – в 8 битах:

$$\begin{aligned}
 [A]_{\text{пр}} &= 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1; \\
 [B]_{\text{пр}} &= 1\ 0\ 0\ 0\ 0\ 1\ 1\ 1; \\
 [A]_{\text{доп}} &= 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1;
 \end{aligned}$$

$$[B]_{\text{доп}} = 11111001;$$

$$[C]_{\text{доп}} = 00000010;$$

$$[C]_{\text{пр}} = 00000010.$$

В результате сложения получилось положительное число, поэтому других преобразований ответ не требует.

### Умножение двоичных чисел

Умножение выполняется по тем же правилам, что и десятичное умножение, т. е. перемножаются модули чисел, а знак результата получается сложением знаков сомножителей по модулю два.

Известно, что произведение двух  $n$ -разрядных чисел есть число  $2n$ -разрядное:

$$mn = 2n.$$

Возьмем два целых, четырехразрядных двоичных числа:

$$A = -0101;$$

$$B = +0010 \quad (n = 4);$$

$$C = AB.$$

Знак произведения  $\text{sign } C = \text{sign } A \oplus \text{sign } B = 1 \oplus 0 = 1$ . Перемножаем модули

$$\begin{array}{r} [A]_{\text{пр}} = 10101 \\ [B]_{\text{пр}} = 00010 \\ \times \quad 0101 \\ \hline \quad 0000 \\ \quad 0101 \\ \quad 0000 \\ \quad 0000 \\ \hline 00001010 \\ \underbrace{\hspace{2em}}_{n=4} \quad \underbrace{\hspace{2em}}_{n=4} \end{array}$$

При умножении целых чисел в качестве результата берутся  $n$  младших разрядов. При этом в старших  $n$ -разрядах должны быть нули. Если это не так, то имеет место переполнение разрядной сетки маши-

ны. Для расширения разрядной сетки нужно добавить слева нули. Теперь умножим два дробных числа:

$$\begin{array}{l}
 A = -0,101 (n = 3) \\
 B = 0,010 \\
 \text{sign } C = \text{sign } A \oplus \text{sign } B = 1 \oplus 0 = 1
 \end{array}
 \quad
 \begin{array}{r}
 \begin{array}{r}
 \phantom{\times} 101 \\
 \times 010 \\
 \hline
 000 \\
 101 \\
 \hline
 0001010
 \end{array}
 \longrightarrow C_0 = 0,001_2 = \\
 \underbrace{\phantom{0001010}}_{n=3} \underbrace{\phantom{0001010}}_{n=3} \phantom{\longrightarrow} = 1 * 2^{-3} = 0,125
 \end{array}$$

При умножении дробных чисел в качестве результата выбираются  $n$  старших разрядов. При этом происходит округление по правилу: если в  $(n + 1)$ -разряде была единица, то к ответу добавляется единица в младший разряд, если в  $(n + 1)$ -разряде был ноль, то к ответу ничего не добавляется. В примере имеем ноль, поэтому ничего не добавляем. Проверим ответ в десятичной системе:

$$A = 1 \cdot 2^{-1} + 1 \cdot 2^{-3} = 0,625; \quad B = 1 \cdot 2^{-2} = 0,25;$$

$$C_0 = A \cdot B = 0,625 \cdot 0,25 = 0,156;$$

$$\Delta = 0,156 - 0,125 = 0,031 \text{ — абсолютная ошибка;}$$

$$\delta = \frac{\Delta}{C_0} = \frac{0,031}{0,156} = 20 \% \text{ — относительная погрешность.}$$

Округление при умножении дробных чисел вносит значительную погрешность (операции же с целыми числами выполняются абсолютно точно!).

Погрешность, возникающая при умножении дробных чисел, равна

$$\Delta = \pm \frac{1}{2} \cdot 2^{-n} = \pm 2^{-1} \cdot 2^{-n} = \pm 2^{-(n+1)}.$$

Чтобы не потерять точность при вычислениях, нужно расширить разрядную сетку путем добавления нулей справа.

**Операции с плавающей запятой.** Операции над числами с плавающей запятой выполняются по тем же правилам. Мантиссы отрица-

тельных чисел вступают в операцию в обратном или дополнительном коде. Перед сложением чисел сначала выравнивают порядки – приводят число к большему порядку и затем складывают мантиссы. Полученный результат нормализуют. Для умножения чисел перемножают мантиссы по правилам двоичной арифметики, а порядки складывают.

В качестве примера рассмотрим сложение двух действительных десятичных чисел  $A = 3,23$  и  $B = -14,85$  в классическом формате с плавающей запятой, используя дополнительный код.  $C = A + B = ?$

Решение начнем с перевода чисел в двоичную ССч:

$$A = 3,23_{10} = +11,0011101_2 = +0,110011101 \cdot 2^{10};$$

$$B = -14,85_{10} = -1110,1101100_2 = -0,11101101100 \cdot 2^{100}.$$

Количество разрядов после запятой в ненормализованном двоичном числе равно  $n_2 = n_{10} / 0,3 = 2 / 0,3 = 7$ .

Занесем эти числа в разрядную сетку 4 байта:

$$[A]_{\text{пр}} = 0\ 110011101000000000000000\ 0\ 000010;$$

$$[B]_{\text{пр}} = 1\ 111011011000000000000000\ 0\ 000100.$$

Приводим операнды к большему порядку ( $4_{10} = 100_2$ ):

$$[A]_{\text{пр}} = 0\ 001100111010000000000000\ 0\ 000100.$$

Записываем дополнительные коды операндов:

$$[B]_{\text{обр}} = 1\ 000100100111111111111111\ 0\ 000100;$$

$$[B]_{\text{доп}} = 1\ 000100101000000000000000;$$

$$[A]_{\text{доп}} = 0\ 001100111010000000000000;$$

$$[C]_{\text{доп}} = 1\ 010001100010000000000000 \text{ – результат сложения};$$

$$[C]_{\text{обр}} = 1\ 010001100001111111111111;$$

$$[C]_{\text{пр}} = 1\ 101110011110000000000000\ 0\ 000100.$$

Ответ в двоичной системе счисления:

$$C = -0,10111001111 \cdot 2^{100} = -1011,1001111_2 =$$

$$= -(2^3 + 2^1 + 2^0 + 2^{-1} + 2^{-4} + 2^{-5} + 2^{-6} + 2^{-7}) \approx -11,617_{10}.$$

Точный ответ:  $C_0 = -14,85 + 3,23 = -11,62$ .

Появилась погрешность за счет перевода чисел в двоичную ССс.

### 2.2.7. Двоично-десятичная система кодирования

Все операции в ЭВМ выполняются в двоичной ССс. Однако человеку удобно вводить информацию и получать результаты вычислений в десятичной ССс. Для этого используются так называемые двоично-десятичные коды. В них один десятичный разряд представляют четыре двоичными разрядами (тетрадой). При помощи 4 бит можно закодировать 16 различных символов (цифр). Существует много разных систем кодирования [10], но наиболее широко применяется код прямого замещения – код 8–4–2–1 (это веса двоичных разрядов влево от запятой). Составим таблицу соответствия двоично-десятичного кода и десятичных цифр (табл. 7).

Таблица 7

Двоично-десятичный код				Десятичная цифра
8	4	2	1	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9

Остальные комбинации двоичного кода являются лишними (запрещенными). Запишем пример двоично-десятичного кода:

$$1258 = 0001\ 0010\ 0101\ 1000;$$

$$589 = 0000\ 0101\ 1000\ 1001.$$

### Сложение двоично-десятичных чисел

Сложение двоично-десятичных чисел производится по правилам двоичной арифметики, с учетом переносов. Пусть имеем два десятичных числа  $A$  и  $B$ . Требуется найти сумму  $C = A + B$ .

В каждой тетраде выполняется сложение трех чисел – двух слагаемых и переноса из предыдущего разряда, т. е.  $a_n + b_n + p_{n-1}$ . При этом возможны следующие ситуации.

1.  $a_n + b_n + p_{n-1} < 10$ ;  $A = 14$ ;  $B = 23$ ;  $C = A + B = 37$ .

$$A = 0001\ 0100;$$

$$B = 0010\ 0011;$$

$$C = 0011\ 0111 \Rightarrow 37.$$

Ответ получился верный.

2.  $a_n + b_n + p_{n-1} > 15$ ;  $A = 47$ ;  $B = 39$ ;  $C = A + B = 86$ .

$$A = 0100\ 0111;$$

$$B = 0011\ 1001;$$

$$C = 1000 \leftarrow 0000 \Rightarrow 80.$$

Ответ получился неверный, так как был перенос из младшей тетрады в старшую. Тетрада переполняется числом 16, т. е. единица межтетрадного переноса уносит в старшую тетраду 16, а не 10 единиц, как в десятичной ССх (шесть лишних единиц!). Поэтому результат необходимо скорректировать путем добавки + 6. Выполним коррекцию:

$$C = 1000\ 0000$$

$$0000\ 0110 \text{ – коррекция (+ 6).}$$

$$\text{Ответ: } 1000\ 0110 \Rightarrow 86.$$

Ответ правильный.

3.  $10 \leq a_n + b_n + p_{n-1} \leq 15$ ;  $A = 47$ ;  $B = 36$ ;  $C = A + B = 83$ .

$$A = 0100\ 0111;$$

$$B = 0011\ 0110;$$

$C = 0111\ 1101 \Rightarrow$  ответ неверный, хотя и нет межтетрадного переноса, но имеется запрещенная комбинация. Необходимо вызвать ис-

кусственное переполнение тетрады путем добавки + 6. Выполним коррекцию:

$$C = 0111 \ 1101$$

$$0000 \ 0110 - \text{коррекция (+ 6)}.$$

Ответ:  $1000 \leftarrow 0011 \Rightarrow 83$ .

Теперь ответ правильный.

**Внимание!** Коррекция результата выполняется только один раз, поэтому межтетрадный перенос при коррекции не требует еще одной коррекции.

Таким образом, при сложении двоично-десятичных чисел выполняется коррекция результата по правилу: если был межтетрадный перенос (переполнение тетрады) или получилась запрещенная комбинация, то к этой тетраде добавляется + 6 (0110).

Еще один пример:  $A = 479$ ;  $B = 128$ ;  $C = A + B = 607$ .

$$\begin{array}{r} 0100 \quad 0111 \quad 1001 \\ + 0001 \quad 0010 \quad 1000 \\ \hline 0101 \quad 1010 \quad 0001 \end{array} - \text{выполняем коррекцию};$$

$$\begin{array}{r} 0000 \quad 0110 \quad 0110 \\ \hline 0110 \quad 0000 \quad 0111 \end{array} - \text{результат верный}.$$

В старшей тетраде коррекция 0, в средней + 6 (запрещенная комбинация), в младшей тетраде + 6 (был перенос).

### 2.2.8. Переполнение разрядной сетки машины

При сложении чисел одинакового знака может возникнуть переполнение разрядной сетки. Признаком переполнения служит отличие знака результата от знаков слагаемых. Пусть, например, требуется сложить два числа:

$$A = -1101 \quad [A]_{\text{доп}} = 10011;$$

$$B = -1001 \quad [B]_{\text{доп}} = 10111;$$

$$C = A + B \quad [C]_{\text{доп}} = 01010.$$

Складывали два отрицательных числа, а ответ – положительный. Чтобы не было переполнения, необходимо расширить разрядную сетку. Запишем числа  $A$  и  $B$  в 5 битах:

$$A = -01101 \quad [A]_{\text{доп}} = 110011;$$

$$B = -01001 \quad [B]_{\text{доп}} = 110111;$$

$$C = A + B \quad [C]_{\text{доп}} = 101010;$$

$$[C]_{\text{обр}} = 101001;$$

$$C = -10110.$$

Теперь ответ верный. В вычислителях для контроля переполнения следят за переносом в знаковый разряд и из него. Если оба переноса имеют место или оба отсутствуют, то переполнения нет. Если есть только один из переносов, то имеет место переполнение разрядной сетки.

В некоторых машинах для контроля переполнения используют так называемые модифицированные коды: прямой, обратный, дополнительный. В этих кодах под знак числа отводится по 2 бита. После сложения эти знаковые биты складывают между собой по модулю два. Если результат сложения равен 0, то нет переполнения, если результат сложения равен 1, то переполнение есть. Проверим это правило на примере обратного модифицированного кода. Сложим два числа:

$$A = -1101;$$

$$B = -1001;$$

ЗНАК	
$[A]_{\text{обр}}^{\text{М}} = 110010$	
$[B]_{\text{обр}}^{\text{М}} = 110110$	
$\begin{array}{r} 110010 \\ 110110 \\ \hline 101000 \\ \rightarrow 1 \end{array}$	
$\begin{array}{r} 101001 \\ \downarrow \\ \oplus = 1 \end{array}$	ИМЕЕТ МЕСТО ПЕРЕПОЛНЕНИЕ РАЗЯДНОЙ СЕТКИ

$[A]_{\text{обр}}^{\text{М}} = 11\ 10010$	
$[B]_{\text{обр}}^{\text{М}} = 11\ 10110$	
$\begin{array}{r} 11\ 10010 \\ 11\ 10110 \\ \hline 11\ 01000 \\ \rightarrow 1 \end{array}$	
$\begin{array}{r} 11\ 01001 \\ \downarrow \\ \oplus = 0 \end{array}$	переполнения нет

Таким образом, модифицированные коды удобны для использования, хотя и занимают на 1 бит больше памяти.

### 2.2.9. Двоичные счетчики

Счетчиком называют функциональный узел, предназначенный для подсчета числа импульсов, поступивших на его вход. Счетчики широко используются в устройствах дискретной обработки информации. Их можно классифицировать по ряду признаков:

1) по направлению переходов (суммирующие, вычитающие и реверсивные);

2) по способу организации счета (синхронные и асинхронные);

3) по модулю счета (двоичные и с произвольным модулем счета). Модуль счета  $M$  (коэффициент пересчета) равен числу устойчивых состояний счетчика;

4) по способу построения цепей переноса (с последовательным, параллельным и групповым переносом).

Двоичные счетчики – это счетчики, у которых модуль счета кратен величине  $2^n$ , где  $n$  – число разрядов ( $n = 1, 2, 3, \dots$ ).

Простейший суммирующий счетчик можно выполнить на асинхронных Т-триггерах, как показано на рис. 2.6.

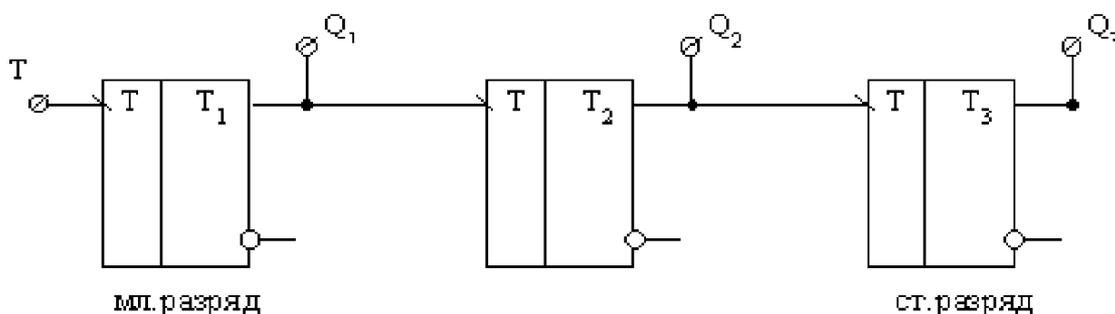


Рис. 2.6. Асинхронный суммирующий счетчик с последовательным переносом

На рис. 2.7 приведены временные диаграммы, поясняющие работу.

Очевидно, что счетчик представляет собой цепь последовательно соединенных триггеров, каждый из которых работает в счетном режиме и меняет свое состояние по заднему фронту (переходу  $1 \rightarrow 0$ ) входного сигнала. Если  $Q_3 Q_2 Q_1$  – двоичное число, записанное в счетчик, то с приходом каждого импульса на вход  $T$  содержимое счетчика увеличивается на 1. На рис. 2.7. в первом, третьем и шестом тактах указано содержимое счетчика (для положительной логики). Если принять, что

двоичное число, записанное в счетчик, есть  $\overline{Q_3}\overline{Q_2}\overline{Q_1}$ , т. е. читается на инверсных выходах триггеров, то счетчик становится вычитающим. Это видно по временной диаграмме.

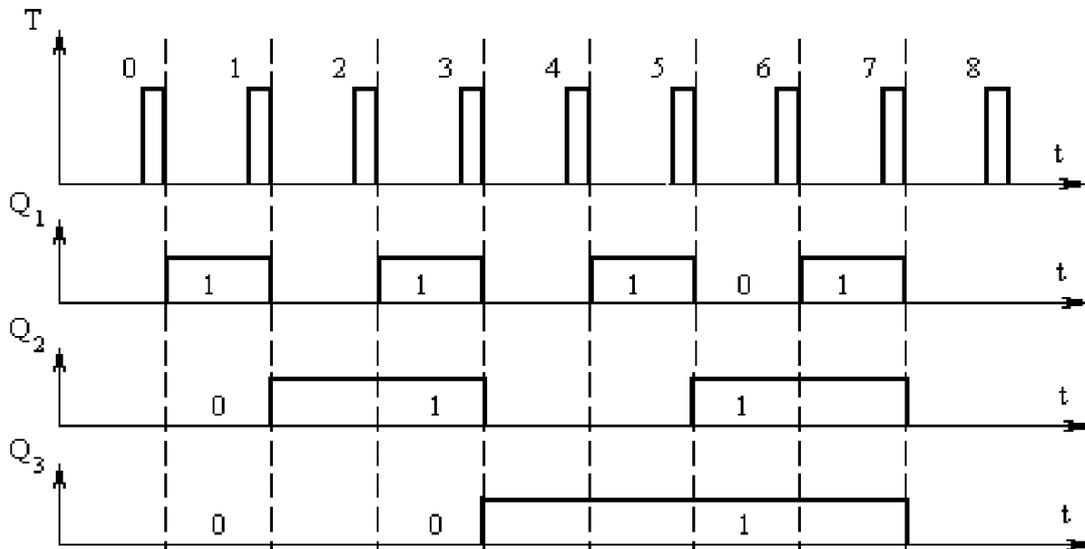


Рис. 2.7. Временные диаграммы асинхронного двоичного счетчика

Счетчики на асинхронных триггерах применяются редко из-за их малого быстродействия, определяемого временем перехода из состояния 111 в состояние 000. Чаще применяются синхронные счетчики, все разряды которых одновременно изменяют свое состояние под воздействием синхроимпульса С.

Для синтеза синхронного счетчика оставим его таблицу истинности (табл. 8).

Таблица 8

Таблица истинности счетчика

Номер набора (сигнал счета С)	Такт $t$ Состояние счетчика			Такт $t + 1$		
	Q3	Q2	Q1	Q3	Q2	Q1
0	0	0	0	0	0	1
1	0	0	1	0	1	0
2	0	1	0	0	1	1
3	0	1	1	1	0	0
4	1	0	0	1	0	1
5	1	0	1	1	1	0
6	1	1	0	1	1	1
7	1	1	1	0	0	0

По такой таблице можно синтезировать счетчик, т. е. составить его схему на конкретных типах триггеров. Для этого необходимо иметь словарь переходов триггера, т. е. значения управляющих сигналов, которые следует подать на триггер, чтобы перевести его в требуемое состояние. Словарь переходов (табл. 9) составляется на основании таблицы истинности триггера. Если счетчик выполнять на Т-триггерах, то с учетом его словаря переходов следует составить таблицу возбуждения всех разрядов счетчика (табл. 10).

Таблица 9

**Словари переходов триггеров**

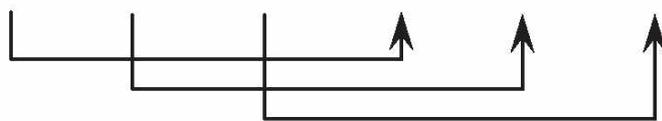
$Q(t)$	Сигналы на входах триггера для перевода его в нужное состояние						$Q(t+1)$
	JK-триггер		RS-триггер		Т-триггер	D-триггер	
	$J$	$K$	$R$	$S$			
0	0	–	–	0	0	0	0
0	1	–	0	1	1	1	1
1	–	1	1	0	1	0	0
1	–	0	0	–	0	1	1

Прочерк означает безразличное состояние входа

Таблица 10

**Таблица возбуждения триггеров**

Номер набора	Такт $t$			Такт $t+1$			Управляющий вход		
	$Q_3$	$Q_2$	$Q_1$				$T_3$	$T_2$	$T_1$
0	0	0	0	0	0	1	0	0	1
1	0	0	1	0	1	0	0	1	1
2	0	1	0	0	1	1	0	0	1
3	0	1	1	1	0	0	1	1	1
4	1	0	0	1	0	1	0	0	1
5	1	0	1	1	1	0	0	1	1
6	1	1	0	1	1	1	0	0	1
7	1	1	1	0	0	0	1	1	1



Переходы триггеров

Составим уравнения для управляющих входов по единичным значениям функций  $T_i$ :

$$T_3 = \bar{Q}_3 Q_2 Q_1 + Q_3 Q_2 Q_1,$$

$$T_2 = \bar{Q}_3 \bar{Q}_2 Q_1 + \bar{Q}_3 Q_2 Q_1 + Q_3 \bar{Q}_2 Q_1 + Q_3 Q_2 Q_1,$$

$$T_1 = 1.$$

В уравнения заносятся исходные состояния счетчика (в такте  $t$ ), являющиеся набором входных переменных. Таким образом, для счетчика на Т-триггерах с модулем  $M = 2^3 = 8$  после минимизации функции возбуждения имеют вид:

$$T_1 = 1,$$

$$T_2 = Q_1,$$

$$T_3 = Q_2 Q_1.$$

Схема счетчика, использующая полученные уравнения, показана на рис. 2.8. Это счетчик с параллельным переносом.

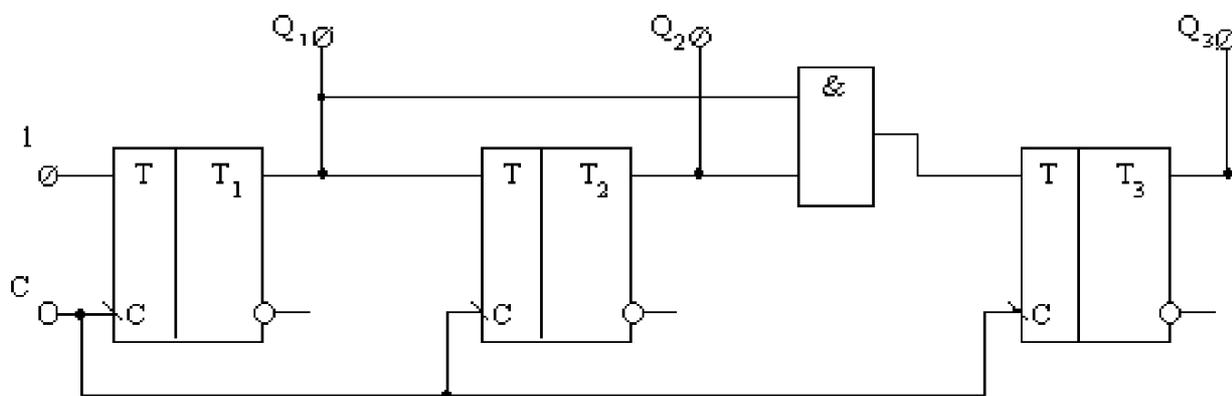


Рис. 2.8. Синхронный счетчик с параллельным переносом

Аналогично синтезируется счетчик, если выполнить часть разрядов на одних триггерах, а часть – на других. При этом точно так же составляется таблица возбуждения для каждого разряда. Уравнения минимизируются известными методами.

### 2.2.10. Счетчики с произвольным основанием

Различные области применения требуют счетчиков с модулем счета не кратным  $2^n$ . Например, для схем часов, календарей и пр. В таких счетчиках используются не все кодовые комбинации. Лишние комбинации называются запрещенными. Порядок смены кодовых комбинаций на счетчике в общем случае может быть произвольным, а соседние комбинации не обязательно отличаются на  $\pm 1$ . Такие счетчики используются в устройствах управления вычислителями. Существуют различные методы синтеза счетчиков.

Рассмотрим метод прямого синтеза по заданному основанию на конкретном примере.

Пусть требуется построить счетчик с модулем  $M = 6$ . Порядок смены состояний: 001, 010, 011, 100, 110, 111. Старший разряд выполнен на D-триггере, средний на T-триггере, а младший на RS.

Составим таблицу возбуждения всех разрядов счетчика (табл. 11).

Таблица 11

Таблица возбуждения счетчика

$Q_3$	Такт $t$		Такт $t + 1$			Управляющие входы			
	$Q_2$	$Q_1$				$D_3$	$T_2$	$R_1$	$S_1$
0	0	1	0	1	0	0	1	1	0
0	1	0	0	1	1	0	0	0	1
0	1	1	1	0	0	1	1	1	0
1	0	0	1	1	0	1	1	–	0
1	1	0	1	1	1	1	0	0	1
1	1	1	0	0	1	0	1	0	–

Минимизацию функций возбуждения выполним с помощью карт Карно (рис. 2.9). Для этого занесем в карты все состояния (нули, единицы и запрещенные комбинации). Координаты клеток в карте определяются состоянием счетчика в такте  $t$ .

D3:	Q3	$\bar{Q}_3$		
Q2	1	0	1	0
$\bar{Q}_2$	1		0	
	Q1			

T2:	Q3	$\bar{Q}_3$		
Q2	0	1	1	0
$\bar{Q}_2$	1		1	
	Q1			

R1:	Q3	$\bar{Q}_3$		
Q2	0	0	1	0
$\bar{Q}_2$	-		1	
	Q1			

S1:	Q3	$\bar{Q}_3$		
Q2	1	-	0	1
$\bar{Q}_2$	0		0	
	Q1			

Рис. 2.9. Карты Карно счетчика с модулем M = 6

Пустые клетки карт Карно соответствуют запрещенным комбинациям и их можно использовать для минимизации. Нетрудно получить:

$$D_3 = Q_3 \bar{Q}_1 + \bar{Q}_3 Q_2 Q_1; \quad \bar{T}_2 = Q_2 \bar{Q}_1;$$

$$\bar{R}_1 = Q_3 + \bar{Q}_1; \quad \bar{S}_1 = Q_1 + \bar{Q}_2.$$

По полученным минимальным формам составляем теоретическую схему счетчика (на произвольных элементах), приведенную на рис. 2.10.

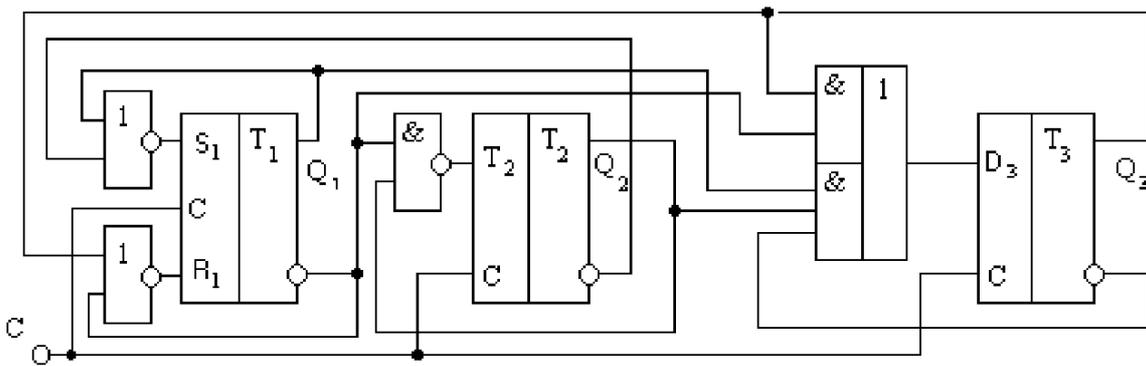


Рис. 2.10. Схема счетчика с модулем M = 6 на произвольных элементах

При реализации счетчика схему следует перевести на конкретные элементы заданной серии микросхем, выбрать триггеры и логические элементы. Составить полную принципиальную схему с перечнем элементов в соответствии с ЕСКД. В схеме предусмотреть начальную установку кода.

## 2.3. Оформление контрольной работы

Контрольная работа выполняется на стандартных листах бумаги (А4) и пишется с одной стороны. Она оформляется следующим образом.

1. Титульный лист (см. приложение) считается первым листом и не нумеруется.

2. На втором листе приводится содержание пояснительной записки. Он оформляется рамкой и содержит заполненный основной штамп по ЕСКД.

3. На третьем листе пишется задание на контрольную работу с указанием номера конкретного варианта. Третий и все последующие листы также оформляются рамкой.

4. Начиная с четвертого листа, размещается текст работы, включающий все исходные логические функции и их преобразования, расчеты с подробным пояснительным текстом; полные принципиальные схемы устройств, выполненные согласно ЕСКД; в схеме счетчика следует предусмотреть кнопку начальной установки кода; выводы по работе; список использованной литературы.

## 3. РАСЧЕТНО-ГРАФИЧЕСКОЕ ЗАДАНИЕ

### 3.1. Варианты задания конечного автомата Мили

*А.* Для своего номера варианта задания выпишите из табл. 13 восемь четверок чисел и постройте граф конечного автомата Мили.

*Б.* Определите тип и количество элементов памяти по табл. 12.

Таблица 12

Тип элемента памяти

Последняя цифра варианта задания	0	1	2	3	4
	5	6	7	8	9
Тип синхронного триггера	JK	T	RS	JK	D

*В.* Составьте таблицы переходов и выходов КА.

Г. Составьте таблицу возбуждения элементов памяти.

Д. Синтезируйте комбинационную часть КА.

Е. Составьте полную логическую схему автомата. Реализуйте КА на микросхемах одной из серий: К155, К1531, К555, К1533, К561, К564. Составьте принципиальную схему с перечнем элементов по правилам ЕСКД.

Таблица 13

**Варианты задания конечного автомата**

Вершина графа	$a_1$		$a_2$		$a_3$		$a_4$	
	$Z_i$	$W_i$	$Z_i$	$W_i$	$Z_i$	$W_i$	$Z_i$	$W_i$
Сигнал								
Номер выходящей из вершины ветви	1234	1234	1234	1234	1234	1234	1234	1234
Вариант	Индексы сигналов							
1	0241	0343	3201	1203	3102	4403	0100	0400
2	3241	2131	3240	3120	0321	0323	0023	0044
3	2340	3320	0012	0041	0300	0400	4002	3004
4	2431	2212	1342	1111	4000	3000	0430	0330
5	3210	1410	0010	0030	0312	0134	4200	2300
6	2300	3200	0240	0130	4020	2030	0124	0131
7	0320	0240	1234	2321	4032	3022	0100	0100
8	0012	0031	3410	1410	3412	1431	0300	0200
9	0210	0430	4300	2200	0342	0414	1042	1033
10	0340	0130	4310	1110	1340	3420	0001	0003
11	3000	4000	4023	2014	0300	0400	0132	0334
12	0034	0022	1030	4030	0002	0004	4102	1403
13	4001	1004	3200	2100	0041	0014	2400	1400
14	0213	0314	3042	4022	0014	0012	0020	0010
15	0102	0201	2040	4030	3410	4240	0304	0102
16	4003	4001	0020	0020	0201	0202	0102	0103
17	4100	2100	0130	0430	0024	0034	0001	0003
18	0210	0440	1002	4002	0201	0301	2004	2004
19	0200	0200	0310	0440	0021	0044	3000	3000

Продолжение табл. 13

Вершина графа	$a_1$		$a_2$		$a_3$		$a_4$	
	$Z_i$	$W_j$	$Z_i$	$W_j$	$Z_i$	$W_j$	$Z_i$	$W_j$
Сигнал								
Номер выходящей из вершины ветви	1234	1234	1234	1234	1234	1234	1234	1234
Вариант	Индексы сигналов							
20	2104	1103	4320	1420	0402	0404	4201	1101
21	2004	2002	0100	0300	3000	4000	2410	1230
22	0032	0034	1342	3324	0043	0012	2000	2000
23	0423	0124	1403	1301	1302	4102	0423	0121
24	4302	1202	1432	2213	2304	4404	3000	2000
25	0132	0114	0140	0220	3041	4021	3024	1041
26	2314	3114	1200	2200	0432	0414	0010	0020
27	1020	4010	0403	0404	4320	4240	2034	3032
28	0024	0034	2014	2013	0032	0042	0400	0100
29	3240	3430	1324	2331	1034	3023	4010	1040
30	2004	1003	3241	3211	0014	0022	3120	1330
31	1024	2043	0410	0120	3041	3021	3000	3000
32	0103	0302	0014	0032	3042	2011	0001	0003
33	3124	3214	0231	0111	3100	3100	1034	3013
34	4010	4040	1043	1044	0002	0003	4320	3430
35	3010	1010	1020	4030	0013	0013	1023	4022
36	0301	0302	3001	1004	2010	3020	2004	4002
37	0231	0324	3140	4140	0200	0400	0234	0141
38	4021	4013	3042	4034	0004	0002	2040	4010
39	0200	0200	3204	3303	3100	3300	0413	0242
40	4301	1202	0102	0204	0003	0004	0300	0100
41	2030	4030	0143	0411	0312	0313	0003	0001
42	0043	0032	0423	0323	0100	0400	0423	0424
43	4310	2240	0431	0143	1432	4114	2300	4400
44	2000	1000	4213	1334	0021	0042	2000	1000
45	0041	0034	1402	3204	0314	0411	1002	2002

Продолжение табл. 13

Вершина графа	$a_1$		$a_2$		$a_3$		$a_4$	
	$Z_i$	$W_j$	$Z_i$	$W_j$	$Z_i$	$W_j$	$Z_i$	$W_j$
Сигнал								
Номер выходящей из вершины ветви	1234	1234	1234	1234	1234	1234	1234	1234
Вариант	Индексы сигналов							
46	0403	0204	2400	2200	0040	0030	2040	4010
47	0120	0110	0020	0020	4001	1001	1400	2300
48	0200	0300	3420	2440	0421	0332	0104	0104
49	0400	0400	0312	0114	1024	2031	3004	4002
50	3200	2400	3402	4404	0213	0413	1300	4400
51	0104	0304	0210	0140	0431	0113	0030	0010
52	0002	0001	1200	2100	4123	1143	1420	4240
53	0132	0131	2043	4024	0030	0020	2003	4003
54	0021	0023	0002	0004	0401	0204	2300	1400
55	0400	0300	0020	0010	2010	2010	0203	0401
56	1040	1030	1200	1200	0320	0310	0210	0310
57	0100	0100	0100	0100	4210	4440	1020	3010
58	0040	0020	1204	3202	3400	1200	0324	0424
59	0100	0300	0103	0302	3401	4304	3010	2010
60	0420	0330	1032	1012	0200	0200	0001	0002
61	0314	0412	1432	4133	2004	3004	0300	0400
62	2041	4034	0431	0241	4103	4202	0413	0333
63	0030	0040	0010	0010	0124	0343	0020	0020
64	0021	0032	1040	1040	0040	0040	0321	0234
65	0304	0102	4301	1103	2013	1014	1230	4340
66	0402	0104	0201	0303	2140	1140	2030	2030
67	0400	0200	1402	2203	3120	4230	2040	4020
68	4130	2340	0423	0322	0040	0030	1320	2110
69	4210	3340	0243	0423	3040	4040	0201	0102
70	4321	4311	0341	0122	0023	0014	0200	0200
71	1003	3003	1300	2300	2034	2022	3020	3040

Продолжение табл. 13

Вершина графа	$a_1$		$a_2$		$a_3$		$a_4$	
	$Z_i$	$W_i$	$Z_i$	$W_i$	$Z_i$	$W_i$	$Z_i$	$W_i$
Сигнал								
Номер выходящей из вершины ветви	1234	1234	1234	1234	1234	1234	1234	1234
Вариант	Индексы сигналов							
72	0403	0402	1234	2332	0410	0430	0100	0100
73	0204	0104	0304	0202	2310	4440	4310	4220
74	2143	4334	0102	0403	4203	3103	2000	3000
75	0002	0003	0431	0231	2400	4200	1032	4033
76	3204	1303	0400	0200	1020	1040	0301	0103
77	0234	0234	1020	4040	0132	0134	1300	2200
78	1024	1011	4030	4040	0423	0241	2430	2330
79	4003	1002	0031	0013	0024	0034	0013	0042
80	0321	0122	1020	4040	3021	3042	4100	3300
81	0004	0001	1003	2003	0401	0401	0341	0234
82	4001	2001	3200	3200	4021	3044	4302	1104
83	1240	2110	4001	1003	4203	3204	4203	2304
84	3014	4042	0423	0434	0020	0030	2410	2410
85	4010	3030	3214	4122	1030	4030	3042	1031
86	0403	0203	0140	0120	0042	0021	2340	3210
87	1300	2200	0302	0103	0320	0240	0201	0204
88	2013	2024	0100	0300	1020	2010	3410	2410
89	0203	0302	0143	0322	0040	0030	3401	1202
90	4010	2020	0200	0100	3410	4410	0423	0233
91	0302	0104	2303	3200,	0214	0141	0010	0020
92	0021	0022	4000	1001	4021	2031	2040	1040
93	3010	1010	4120	4310	0041	0014	0103	0301
94	2004	4002	2003	3004	0013	0011	4003	2001
95	2003	3001	1204	3103	0402	0404	0300	0200
96	0104	0402	0041	0033	3400	4300	0001	0003
97	2143	3313	2104	1303	1342	3233	4230	1210

Вершина графа	$a_1$		$a_2$		$a_3$		$a_4$	
	$Z_i$	$W_j$	$Z_i$	$W_j$	$Z_i$	$W_j$	$Z_i$	$W_j$
Номер выходящей из вершины ветви	1234	1234	1234	1234	1234	1234	1234	1234
Вариант	Индексы сигналов							
98	0032	0024	4002	1003	4003	2004	0321	0224
99	1320	4330	0031	0011	1324	3211	1230	2330
100	0030	0020	1003	2004	2401	1104	2400	4400

### 3.2. Синтез эквивалентного автомата Мура

**А.** Путем эквивалентного преобразования исходного автомата Мили в автомат Мура, постройте граф и таблицу переходов эквивалентного автомата Мура.

**Б.** Составьте полную логическую схему автомата. Реализуйте КА на микросхемах заданной серии, составьте принципиальную схему с перечнем элементов по правилам ЕСКД.

### 3.3. Методические указания и пояснения к работе

#### 3.3.1. Конечные автоматы

Граф синтезируемого автомата Мили для каждого варианта получается путем исключения некоторых ветвей обобщенного графа автомата, имеющего 4 внутренних состояния (рис. 3.1). У такого графа из каждой вершины выходят 4 ветви (и столько же входят). Каждая ветвь символизирует переход автомата в другое внутреннее состояние  $a_k$  ( $k=1, 2, 3, 4$ ) при совместном действии входного сигнала  $Z_i$ , выходного сигнала  $W_j$  и обозначается их комбинацией  $Z_iW_j$  для конкретного значения индексов. Эти индексы берутся из табл. 13 в строке, номер которой совпадает с номером варианта задания. Здесь каждой вершине графа  $a_k$  поставлены в соответствие два набора индексов по 4 цифры:

для  $i$  и  $j$  соответственно. При построении графа следует для каждой ветви, выходящей из каждой вершины, сформировать комбинацию  $Z_iW_j$  и указать ее на графе в соответствии с порядковой нумерацией выходящих ветвей. Этот процесс показан на рис. 3.2.

Порядковая нумерация выходящих ветвей для каждой вершины указана на рис. 3.1.

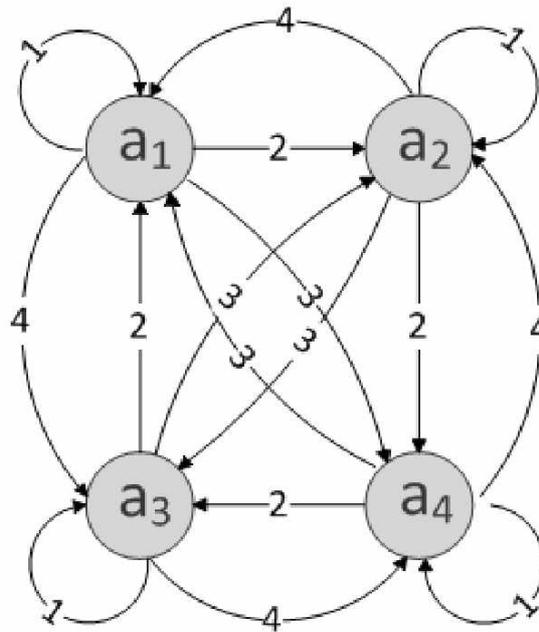


Рис. 3.1. Обобщенный граф автомата с четырьмя внутренними состояниями

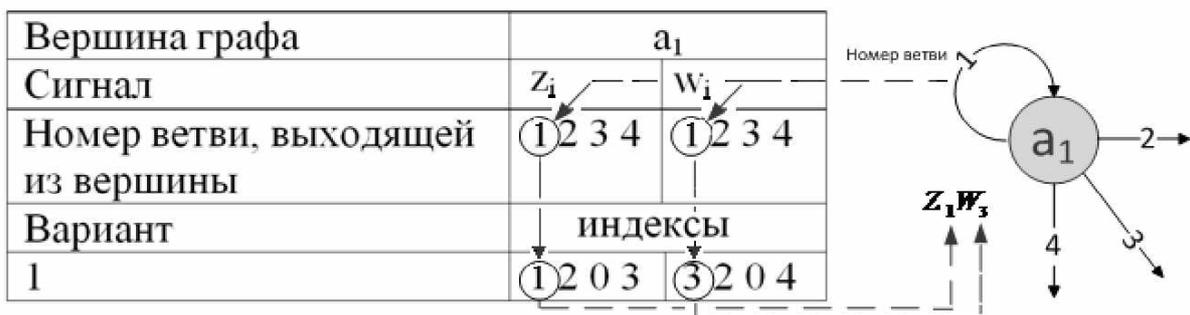


Рис. 3.2. Схема нумерации внутренних состояний автомата

Пример конкретного варианта графа для следующей кодировки индексов сигналов: 1300 2100 0210 0330 0123 0311 0003 0002, приведен на рис. 3.3.

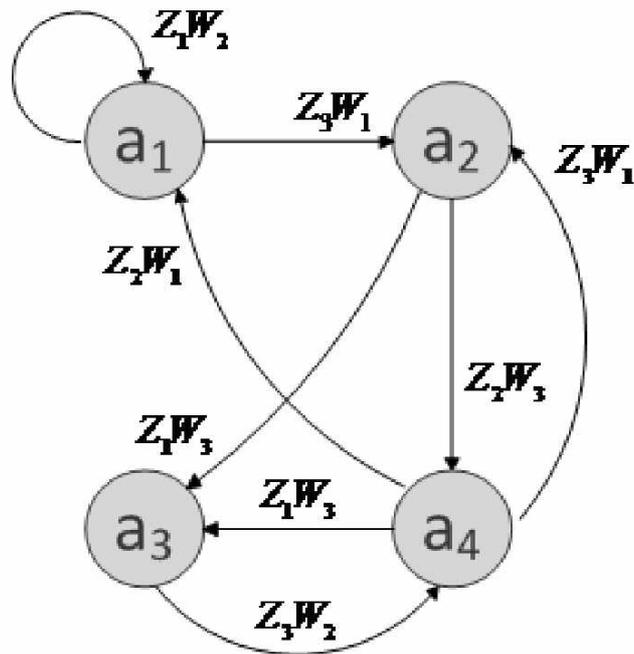


Рис. 3.3. Пример составления графа с учетом варианта задания

Порядковая нумерация ветвей графа опущена, так как каждая ветвь задается определенной комбинацией сигналов. По такому графу легко записать таблицы переходов и выходов, необходимые для проведения структурного синтеза КА. За исходное состояние автомата принимается состояние  $a_1$ .

### 3.3.2. Структурный синтез конечных автоматов

Под цифровым конечным автоматом понимают дискретный преобразователь информации, состоящий из комбинационной схемы и элементов памяти. Комбинационная схема строится из логических элементов, память – из элементарных автоматов, а именно – синхронных триггеров различных типов.

Работа подобных цифровых устройств описывается набором входных сигналов  $Z = \{Z_1, Z_2, \dots, Z_p\}$ , набором выходных сигналов  $W = \{W_1, Z_2, \dots, Z_s\}$ , множеством внутренних состояний (выходных сигналов элементов памяти)  $A = \{a_1, a_2, \dots, a_k\}$ , двумя характеристическими функциями: функцией переходов  $\delta$  и функцией выходов  $\lambda$ , а также начальным состоянием автомата  $a_1 \in A$ .

Различают два типа автоматов: автоматы Мили и Мура. Для каждого из них внутреннее состояние в последующем такте полностью определяется входным сигналом и внутренним состоянием в данном такте, что отражается в записи функций переходов

$$a(t+1) = \delta[z(t), a(t)],$$

где  $t$  – автоматное время.

Отличаются автоматы Мили и Мура лишь записью функции выходов. Если в автомате Мили выходной сигнал зависит как от внутреннего состояния автомата, так и от входного сигнала в том же такте

$$W(t) = \lambda[z(t), a(t)],$$

то в автомате Мура выходной сигнал определяется только внутренним состоянием в данном такте

$$W(t) = \lambda[a(t)],$$

В частном случае возможно и  $W(t) = a(t)$ , т. е. выходной сигнал автомата одновременно является выходным сигналом памяти.

Практически указанное отличие проявляется в том, что при смене входного сигнала в пределах такта в автомате Мили состояние выхода изменяется, а в автомате Мура – сохраняется неизменным, так как изменение состояния элементов памяти происходит только в момент действия импульсов синхронизации.

Каждое состояние входа и выхода любого автомата однозначно задается комбинацией двоичных сигналов на соответствующих  $n$ -разрядных шинах, поэтому условимся каждое отдельное состояние кодировать набором двоичных символов (битов):

$Z_i \sim x(t) = \{x_1, x_2, x_n\}$  – входной сигнал;

$W_i \sim y(t) = \{y_1, y_2, y_m\}$  – выходной сигнал;

$a_i \sim Q(t) = \{Q_1, Q_2, Q_r\}$  – сигнал внутреннего состояния (состояния триггеров памяти);

$U(t) = \{u_1, u_2, \dots, u_k\}$  – входной сигнал памяти.

В такой записи символическим обозначениям  $Z_i$ ,  $W_i$  и  $a_i$  соответствуют двоичные кодовые комбинации  $x(t)$ ,  $y(t)$  и  $Q(t)$ , где  $x_j$ ,  $y_j$ ,  $Q_j$ , и  $u_j$  – отдельные биты кода, каждый из которых передается по своей информационной шине. Структурные схемы автоматов

Мили и Мура показаны на рис. 3.4 и 3.5. Исходными данными для их синтеза являются таблицы переходов и выходов.

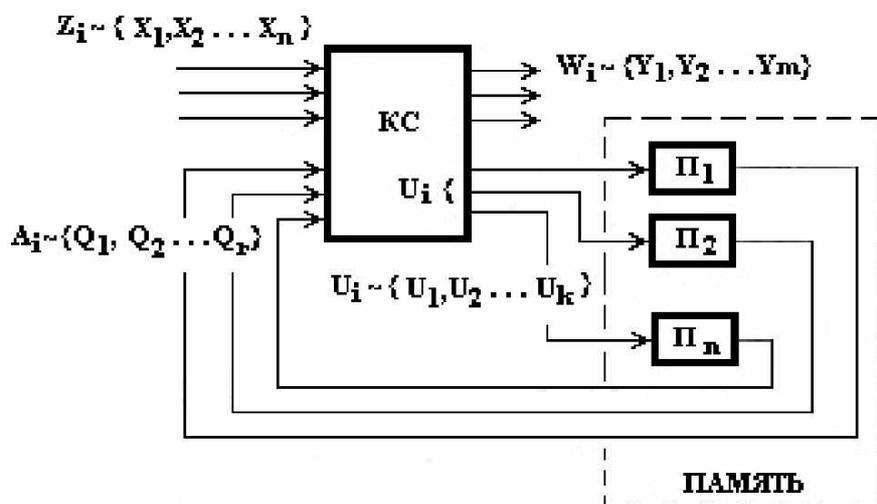


Рис. 3.4. Структурная схема КА Мили

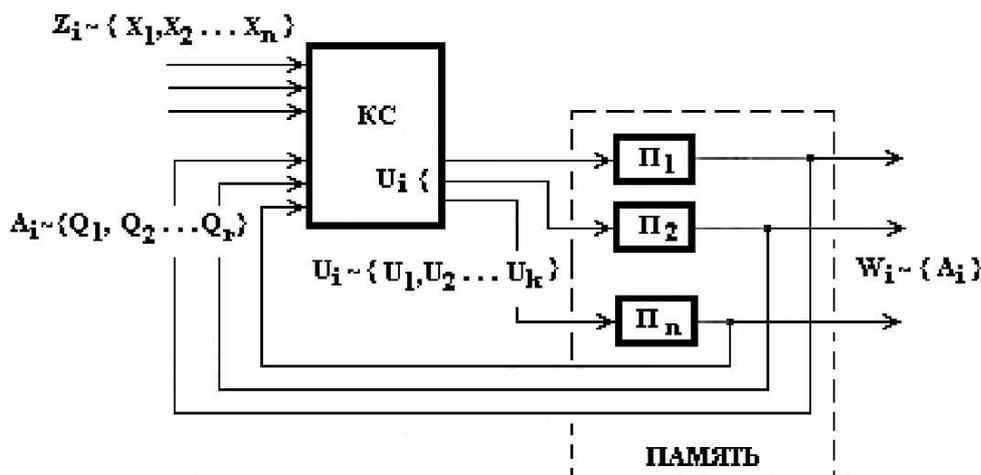


Рис. 3.5. Структурная схема КА Мура

Структурный синтез КА Мили предполагает выполнение ряда этапов.

1. Выбирают систему логических элементов и элементов памяти, на которых будет строиться КА.

2. Определяют недостающие входные данные:

- число элементов памяти  $r \geq \log_2 k$ ;
- число разрядов входной шины  $n \geq \log_2 p$ ;

– число разрядов выходной шины  $m \geq \log_2 s$ ,  
 где  $k, p, s$  – количество внутренних состояний, входных  $Z_i$  и выходных  $W_i$  сигналов соответственно. Числа  $r, n, m$  – целые.

3. Кодируют автомат. Это означает, что каждому входному и выходному сигналу и внутреннему состоянию ставят в соответствие определенный двоичный код.

4. Переводят исходные таблицы выходов и переходов из символического алфавита в двоичный.

5. По таблице выходов составляют систему логических уравнений, связывающих выходные, сигналы КА  $y_j$  с внутренними состояниями и входными сигналами  $x_j$ . Общий вид такого уравнения

$$Y_j = f_j(x_1, x_2, \dots, x_n, Q_1, Q_2, \dots, Q_r).$$

6. По таблице переходов составляют таблицу возбуждения памяти, т. е. двоичных сигналов, которые следует подавать на входы триггеров памяти для перевода их в требуемые состояния. Словари переходов наиболее часто применяемых триггеров приведены в табл. 14.

Таблица 14

### Словари переходов триггеров

$Q(t)$	Сигналы на входах триггера для перевода его в нужное состояние					$Q(t+1)$	
	JK-триггер		RS-триггер		Т-триггер		D-триггер
	J	K	R	S			
0	0	–	–	0	0	0	
0	1	–	0	1	1	1	
1	–	1	1	0	1	0	
1	–	0	0	–	0	1	

Прочерк означает безразличное состояние входа

Для синтезируемого КА с помощью словаря следует преобразовать его таблицу переходов в таблицу возбуждения памяти (процесс преобразования поясняется ниже на конкретном примере).

7. По таблице возбуждения памяти составляют систему логических уравнений, связывающих входные сигналы памяти  $U_j$  с внутренними состояниями  $Q_j$  и входными сигналами  $x_j$ :

$$U_j = f_j(x_1, x_2, \dots, x_n, Q_1, Q_2, \dots, Q_r).$$

8. Все уравнения, полученные в пп. 5 и 7, минимизируют при помощи карт Карно.

9. По минимальным формам составляют полную логическую схему КА на микросхемах заданной серии.

### 3.3.3. Пример структурного синтеза автомата Мили

Зададим автомат Мили таблицами переходов и выходов (рис. 3.6).

$a(t+1) = \delta[a(t), Z(t)]$					$W(t) = \lambda[a(t), Z(t)]$				
Сост. входа	$a_1$	$a_2$	$a_3$	$a_4$	Сост. входа	$a_1$	$a_2$	$a_3$	$a_4$
$Z_1$	$a_1$	$a_4$	$a_4$	-	$Z_1$	$W_2$	$W_3$	$W_3$	-
$Z_2$		$a_3$	$a_1$	-	$Z_2$	-	$W_3$	$W_1$	-
$Z_3$	$a_2$	-	$a_2$	$a_3$	$Z_3$	$W_1$	-	$W_1$	$W_2$

Рис. 3.6. Исходный КА Мили

1. Выбираем в качестве элементов памяти JK-триггеры. Базис логических элементов – произвольный.

2. Для данного примера очевидно:

$k = 4$  – число внутренних состояний ( $a_k$ );

$p = 3$  – число входных сигналов ( $Z_i$ );

$s = 3$  – число выходных сигналов ( $W_I$ ).

3. Находим:

– число элементов памяти

$$r > \log_2 k = \log_2 4 = 2;$$

– число разрядов входной шины

$$n > \log_2 p = \log_2 3 = 2;$$

– число разрядов выходной шины

$$m > \log_2 s = \log_2 3 = 2.$$

4. Кодировем автомат, ставя в соответствие каждому символическому сигналу произвольный двоичный код (число разрядов кода соответствует найденным  $r, n, m$ ).

Таблица 15

**Кодировка автомата Мили**

Входные сигналы			Выходные сигналы			Сигналы памяти		
Состояние входа	Биты кода		Состояние выхода	Биты кода		Внутренние состояние	Биты кода	
	$x_1$	$x_2$		$y_1$	$y_2$		$Q_1$	$Q_2$
$Z_1$	0	0	$W_1$	0	0	$a_1$	0	0
$Z_2$	0	1	$W_2$	0	1	$a_2$	0	1
$Z_3$	1	1	$W_3$	0	0	$a_3$	1	0
						$a_4$	1	1

С учетом введенных кодов переводим таблицы переходов и выходов в двоичный алфавит (рис. 3.7).

Таблица переходов ( $\delta$ )

$x_1x_2$	$Q_1 Q_2$			
	00	01	10	11
00	00	11	11	-
01	-	10	00	-
11	01	-	01	10

Таблица выходов ( $\lambda$ )

$x_1x_2$	$Q_1 Q_2$			
	00	01	10	11
00	01	10	10	-
01	-	10	00	-
11	00	-	00	01

Рис. 3.7. Двоично-кодированные таблицы КА Мили

5. По таблице выходов  $\lambda$  составляем логические уравнения для выходных сигналов  $y_1$  и  $y_2$ . Учтем, что в каждой клетке таблицы левый бит характеризует сигнал  $y_1$ , правый –  $y_2$ . Записывая уравнения «по единицам», получаем СДНФ:

$$y_1 = \bar{x}_1\bar{x}_2\bar{Q}_1Q_2 + \bar{x}_1x_2\bar{Q}_1Q_2 + \bar{x}_1\bar{x}_2Q_1\bar{Q}_2;$$

$$y_2 = \bar{x}_1\bar{x}_2\bar{Q}_1\bar{Q}_2 + x_1x_2Q_1Q_2.$$

6. Преобразуем таблицу переходов автомата в таблицу возбуждения памяти. Для обеспечения каждого отдельного перехода из исходного состояния памяти в последующее нужно подать на входы элементов памяти (синхронных триггеров) определенные сигналы. Именно эти сигналы и заносятся в соответствующие клетки таблицы возбуждения памяти.

На рис. 3.8 для примера рассмотрены возможные переходы в первом столбце таблицы переходов (состояния входов  $x_1x_2$  на данном этапе несущественны).

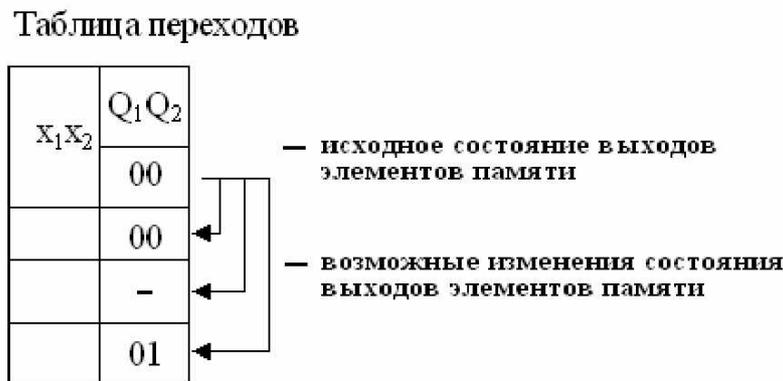


Рис. 3.8. Пример переходов КА Мили

Выделим первое возможное изменение состояния  $Q_1Q_2$ :  $00 \rightarrow 00$ . Поскольку каждый бит характеризует состояние выхода отдельного JK-триггера, то согласно таблице возбуждения такого триггера для перевода его выхода из состояния 0 в состояние 0 (т. е. сохранение состояния) необходимо подать на вход  $J$  сигнал 0, а на вход  $K$  – все равно ноль или единицу. Безразличное состояние входа изображается прочерком в таблице возбуждения памяти. Это иллюстрируется рис. 3.9.



Рис. 3.9. Заполнение таблицы возбуждения памяти

Ясно, что переход второго триггера должен быть отображен аналогичной комбинацией его входных сигналов (на рисунке не показано).

При переходе памяти в следующее возможное состояние процесс показан на рис. 3.10.

Таблица переходов (б)

	Q <sub>1</sub> Q <sub>2</sub>	
	0	0
Переход первого триггера	↓	↓
	0	1

Переход второго триггера

Таблица возбуждения памяти

	Q <sub>1</sub> Q <sub>2</sub>			
	0		0	
	J <sub>1</sub>	K <sub>1</sub>	J <sub>2</sub>	K <sub>2</sub>
	0	-	1	-

Сигналы на входах первого триггера
Сигналы на входах второго триггера

Рис. 3.10. Переход памяти в следующее состояние

Окончательный вид таблицы возбуждения памяти автомата после рассмотрения переходов по всем столбцам приведен в табл. 16.

Таблица 16

Таблица возбуждения памяти, выполненной на JK-триггерах

X <sub>1</sub> X <sub>2</sub>	Q <sub>1</sub> Q <sub>2</sub>							
	0	0	0	1	1	0	1	1
	J <sub>1</sub> K <sub>1</sub>	J <sub>2</sub> K <sub>2</sub>	J <sub>1</sub> K <sub>1</sub>	J <sub>2</sub> K <sub>2</sub>	J <sub>1</sub> K <sub>1</sub>	J <sub>2</sub> K <sub>2</sub>	J <sub>1</sub> K <sub>1</sub>	J <sub>2</sub> K <sub>2</sub>
00	0-	0-	1-	-0	-0	1-	—	—
01	—	—	1-	-1	-1	0-	—	—
11	0-	1-	—	—	-1	1-	-0	-1

Если в качестве элементов памяти использовать Т-триггеры, то таблица возбуждения памяти будет иметь более простой вид, так как Т-триггер имеет один информационный вход (табл. 17).

Таблица возбуждения памяти, выполненной на Т-триггерах

$X_1X_2$	$Q_1Q_2$			
	00	01	10	11
00	00	10	01	–
01	–	11	10	–
11	01	–	11	01

В случае использования D-триггеров таблица возбуждения памяти повторяет таблицу переходов.

7. По таблице возбуждения памяти (для JK-триггеров) составляем логические уравнения сигналов на каждом информационном входе каждого триггера. Записывая их «по единицам», получаем следующие СДНФ:

$$J_1 = \bar{x}_1\bar{x}_2\bar{Q}_1Q_2 + \bar{x}_1x_2\bar{Q}_1Q_2;$$

$$K_1 = \bar{x}_1x_2Q_1\bar{Q}_2 + x_1x_2Q_1\bar{Q}_2;$$

$$J_2 = x_1x_2\bar{Q}_1\bar{Q}_2 + \bar{x}_1\bar{x}_2Q_1\bar{Q}_2 + x_1x_2Q_1\bar{Q}_2;$$

$$K_2 = \bar{x}_1x_2\bar{Q}_1Q_2 + x_1x_2Q_1Q_2.$$

8. Минимизируем уравнения, полученные в пп. 5 и 7, при помощи карт Карно. Так как функции переходов и выходов не определены на некоторых наборах аргументов, доопределяем карты Карно на этих наборах единицами или нулями с целью проведения контуров наиболее высокого ранга (положение этих единиц отмечено на картах символом \*). Так, для  $y_1$  и  $y_2$  карты Карно имеют вид, представленный на рис. 3.11.

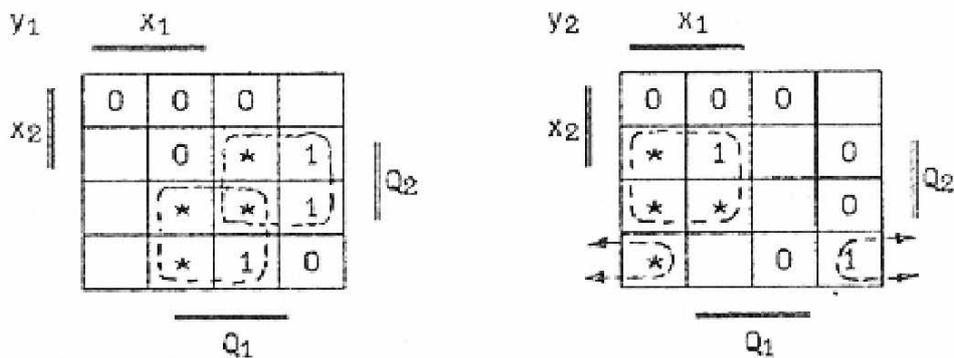


Рис. 3.11. Карты Карно для выходных сигналов

Записываем минимальные ДНФ:

$$y_1 = \bar{x}_2 Q_1 + \bar{x}_1 Q_2;$$

$$y_2 = x_1 Q_2 + \bar{x}_2 \bar{Q}_1 \bar{Q}_2.$$

Проводя минимизацию остальных функций, получаем следующие функции:

$$J_1 = Q_2;$$

$$K_1 = x_2 \bar{Q}_2;$$

$$J_2 = Q_1 \bar{Q}_2 + x_2 \bar{Q}_2;$$

$$K_2 = x_2.$$

9. По полученным минимальным формам составляем логическую схему автомата на микросхемах выбранной серии. Этот процесс сложности не представляет, поэтому приведем только функциональную схему автомата Мили (рис. 3.12).

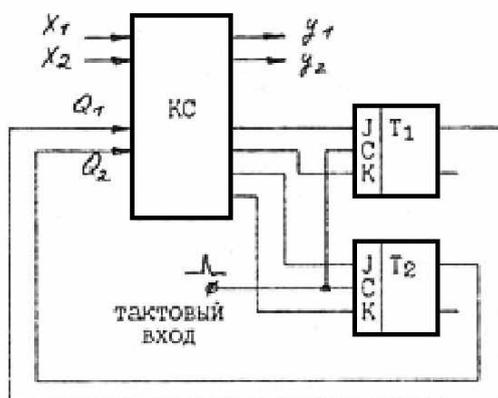


Рис. 3.12. Функциональная схема КА Мили

### 3.3.4. Переход от автомата Мили к автомату Мура

Обычно число внутренних состояний автомата Мура больше или равно числу внутренних состояний автомата Мили. Такое увеличение иллюстрируется рис. 3.13, где показаны фрагменты графов автомата Мили и Мура.

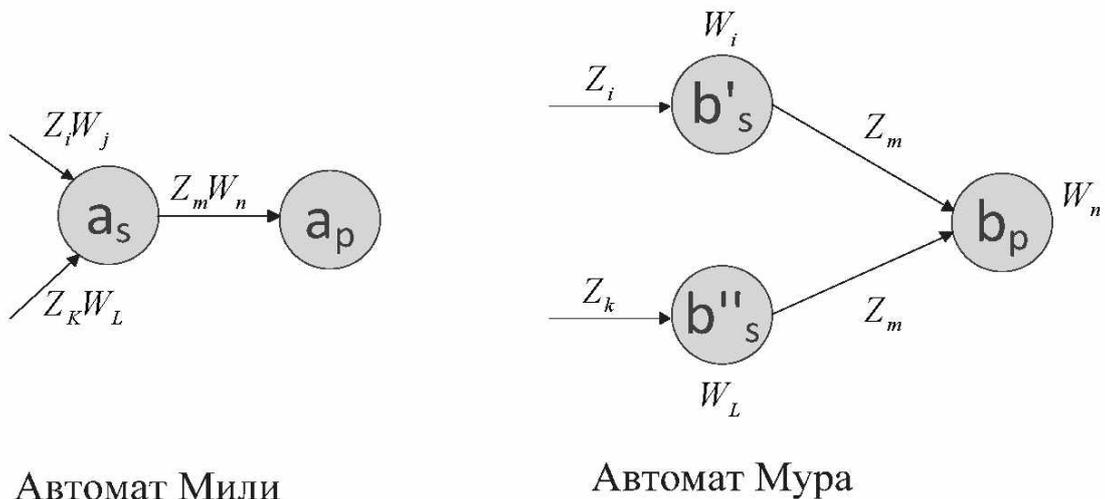


Рис. 3.13. Переход от автомата Мили к автомату Мура

Для лучшего понимания процесса перехода рассмотрим его на примере. Пусть задан автомат Мили совмещенной таблицей переходов (табл. 18), которой соответствует граф, изображенный на рис. 3.13.

Таблица 18

Совмещенная таблица переходов КА Мили

$Z_j$	$a_i$	$a_1$	$a_2$	$a_3$	$a_4$
$Z_1$	$a_1$	$a_4$	$a_4$	$a_4$	—
$Z_2$	—	$a_3$	$a_1$	$a_1$	—
$Z_3$	$a_2$	—	$a_2$	$a_3$	$a_3$

Имеем алфавиты:  $Z = \{Z_1, Z_2, Z_3\}$ ;  $A = \{a_1, a_2, a_3, a_4\}$ ;  $W = \{W_1, W_2, W_3\}$ .

Переход к автомату Мура осуществляется в следующем порядке.

1. Находим множества  $B_s$ , определяемые числом различных выходных сигналов на дугах, входящих в данное состояние (см. рис. 3.13).

$$B1 = \{a_1W_1, a_1W_2\} = \{b_1, b_2\};$$

$$B2 = \{a_2W_1\} = \{b_3\};$$

$$B3 = \{a_3W_2, a_3W_3\} = \{b_4, b_5\};$$

$$B4 = \{a_4W_3\} = \{b_6\}.$$

2. Составляем таблицу переходов автомата Мура на основании таблицы переходов автомата Мили и состояний  $B_s$  ( $s = 1, 2, 3, 4$ ).

Таблица 19

Таблица переходов КА Мура

	$a_1$		$a_2$	$a_3$		$a_4$
$b_i$ $Z_i$	$b_1/W_1$	$b_2/W_2$	$b_3/W_1$	$b_4/W_2$	$b_5/W_3$	$b_6/W_3$
$Z_1$	$b_2$	$b_2$	$b_6$	$b_6$	$b_6$	—
$Z_2$	—	—	$b_5$	$b_1$	$b_1$	—
$Z_3$	$b_3$	$b_3$	—	$b_3$	$b_3$	$b_4$

Для полученного автомата Мура несложно составить граф, понимая, что его выходные сигналы  $W_1, W_2, W_3$  определяются внутренними состояниями  $b_1, \dots, b_6$ . Этот граф изображен на рис. 3.14.

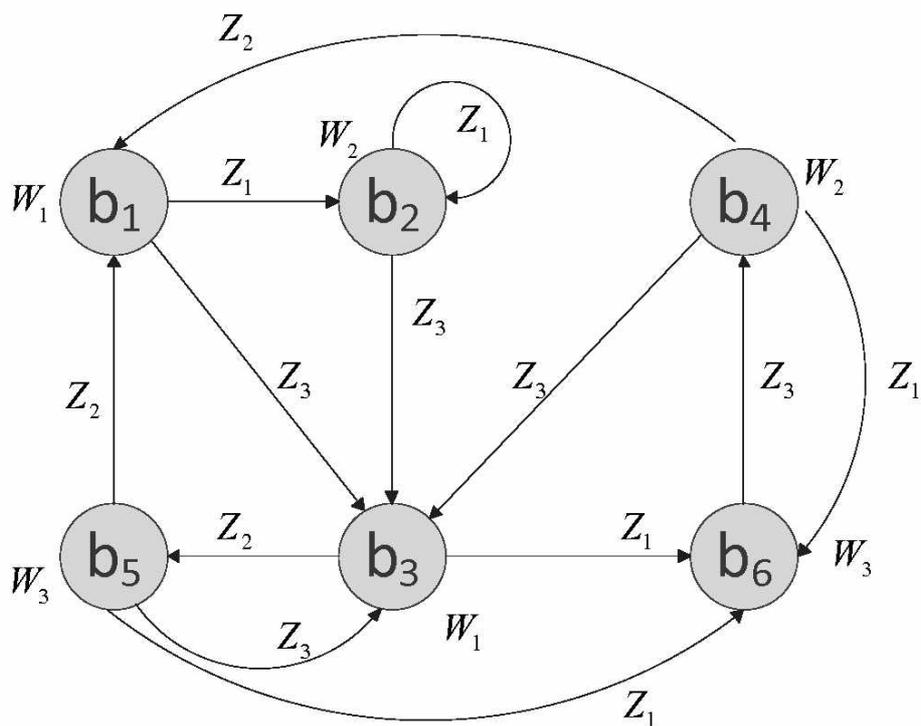


Рис. 3.14. Граф автомата Мура эквивалентного автомату Мили

Пусть автоматы Мили и Мура находятся в начальных состояниях  $a_1$  и  $b_2$  соответственно. Убедиться в эквивалентности преобразования можно путем подачи на входы исходного автомата Мили и полученного автомата Мура некоторой последовательности букв входного алфавита, например такой:

$$Z = \{Z_1, Z_1, Z_3, Z_2, Z_1 \dots \}.$$

Выходная последовательность обоих автоматов будет следующей:

$$W = \{W_2, W_2, W_1, W_3, W_3, \dots \}.$$

Значит, абстрактные автоматы Мили и Мура эквивалентны. При этом выходной алфавит автомата Мура может отличаться от выходного алфавита исходного автомата Мили, поскольку количества внутренних состояний автоматов различно. На этапе структурного синтеза это приводит к тому, что кодировка выходных сигналов автоматов также может отличаться.

### 3.4. Оформление РГЗ

Расчетно-графическое задание выполняется на стандартных листах бумаги (А4) с одной стороны и оформляется следующим образом.

1. Титульный лист (см. приложение) считается первым и не нумеруется.

2. На втором листе приводится содержание РГЗ. Он оформляется рамкой и содержит заполненный основной штамп по ЕСКД.

3. На третьем листе пишется задание на курсовую работу с указанием номера конкретного варианта. Третий и все последующие листы также оформляются рамкой.

4. Начиная с четвертого листа размещается текст пояснительной записки, включающий следующие разделы:

а) общая характеристика синтезируемого устройства и принципы его работы, граф исходного автомата Мили;

б) все исходные логические функции и их преобразования, граф и схема автомата Мура, расчеты с подробным пояснительным текстом;

в) полные принципиальные схемы устройств, выполненные согласно ЕСКД (предусмотреть во всех схемах кнопки начальной установки кода);

г) выводы по работе;

д) список использованной литературы.

Иллюстрации нумеруются арабскими цифрами, применяется сквозная нумерация или нумерация по главам. На все рисунки в тексте делаются ссылки, например (*рисунок 5*) или (*рисунок 1.3*). Если рисунок в работе один, то его не нумеруют, а ссылку в тексте делают так: (*см. рисунок*).

Иллюстрации при необходимости могут иметь наименование и пояснительные данные.

Уравнения и формулы выделяются из текста в отдельную строку. Выше и ниже каждой формулы или уравнения оставляется не менее одной свободной строки. Если уравнение (или формула) не умещается в одну строку, то оно переносится после знака равенства (=) или знаков плюс (+), минус (-), умножения (x), деления (:), причем знак в начале следующей строки повторяется. На нумерованные формулы в тексте должны быть ссылки. Например:

Система (5) решается при начальных условиях (1) и (2).

Порядковые номера формул обозначают арабскими цифрами и помещают в круглых скобках у правого края полосы. Номер для многострочной формулы ставится против последней ее строки. Сквозная нумерация формул применяется в небольших работах, где нумеруется ограниченное число наиболее важных формул.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Смирнов Ю.А.* Основы микроэлектроники и микропроцессорной техники / Ю.А. Смирнов, С.В. Соколов, Е.В. Титов. – СПб.: Лань, 2013. – 496 с.
2. *Микушин А.В.* Цифровые устройства и микропроцессоры / А.В. Микушин, А.М. Сажнев, В.И. Сединин. – СПб: БХВ-Петербург, 2010. – 832 с.
3. *Безуглов Д.А.* Цифровые устройства и микропроцессоры: учебное пособие / Д.А. Безуглов, И.В. Калиенко. – Р-Д., 2008. – 469 с.
4. *Сажнев А.М.* Цифровые устройства и микропроцессоры: конспект лекций / А.М. Сажнев. – Новосибирск: Изд-во НГТУ, 2007. – Ч. 1. – 116 с.
5. *Пухальский Г.Я.* Цифровые устройства: учебное пособие для вузов. – СПб.: Политехника, 2006. – 885 с.