

Министерство образования Республики Беларусь

**Учреждение образования
«Гомельский государственный технический
университет имени П. О. Сухого»**

Кафедра «Промышленная электроника»

Ю. Е. Котова

СХЕМОТЕХНИКА ЦИФРОВЫХ УСТРОЙСТВ

ПРАКТИКУМ

для студентов специальности 1-36 04 02

«Промышленная электроника»

дневной и заочной форм обучения

Электронный аналог печатного издания

Гомель 2024

УДК 621.37/39(075.8)
ББК 32я73
К73

*Рекомендовано к изданию научно-методическим советом
факультета автоматизированных и информационных систем
ГГТУ им. П. О. Сухого
(протокол № 9 от 17.05.2023 г.)*

Рецензент: доц. каф. «Информационные технологии» ГГТУ им. П. О. Сухого
канд. техн. наук, доц. В. С. Захаренко

Котова, Ю. Е.
К73 Схемотехника цифровых устройств : практикум для студентов специальности 1-36 04 02 «Промышленная электроника» днев. и заоч. форм обучения / Ю. Е. Котова. – Гомель : ГГТУ им. П. О. Сухого, 2024. – 71 с. – Систем. требования: PC не ниже Intel Celeron 300 МГц ; 32 Mb RAM ; свободное место на HDD 16 Mb ; Windows 98 и выше ; Adobe Acrobat Reader. – Режим доступа: <https://elib.gstu.by>. – Загл. с титул. экрана.
ISBN 978-985-535-536-7.

Предназначен для получения и закрепления знаний, требуемых в рамках учебной программы по предмету «Схемотехника цифровых устройств» на практических занятиях и при самостоятельной работе.

Для студентов специальности 1-36 04 02 «Промышленная электроника» дневной и заочной форм обучения.

УДК 621.37/39(075.8)
ББК 32я73

ISBN 978-985-535-536-7

© Котова Ю. Е., 2024
© Учреждение образования «Гомельский
государственный технический университет
имени П. О. Сухого», 2024

Предисловие

Данный практикум предназначен для подготовки студентов по специальности «Промышленная электроника» дневной и заочной форм обучения в рамках учебной программы по предмету «Схемотехника цифровых устройств». Структурно практикум разбит на шесть разделов, каждый из которых представляет собой отдельное практическое занятие с необходимыми теоретическими сведениями, примерами и заданиями для самостоятельного выполнения. Прежде, чем приступить к выполнению заданий, настоятельно рекомендуется изучить теоретическую часть для понимания основных принципов, и разобрать приведенные примеры.

Стоит отметить, что издание содержит достаточный, но не исчерпывающий уровень материала для подготовки по данной дисциплине. Поэтому рекомендуется для более углубленной проработки материала обратиться к использованным при составлении практикума источникам.

Практическое занятие № 1

ПЕРЕВОД ЧИСЕЛ ИЗ ОДНОЙ СИСТЕМЫ СЧИСЛЕНИЯ В ДРУГИЕ. АРИФМЕТИЧЕСКИЕ ДЕЙСТВИЯ С МНОГОРАЗРЯДНЫМИ ДВОИЧНЫМИ ЧИСЛАМИ

Теоретические сведения

Системы счисления

Все современные системы счисления (кроме некоторых римских цифр) являются позиционными, т. е. в них одна и та же цифра в разных позициях (слева, справа) имеет разное значение. Например, в десятичном числе 55 левая цифра означает 50, а правая, – только 5. В общем виде в позиционной системе счисления с основанием системы X число A можно представить в виде:

$$A = \sum_{i=1}^n a_i \cdot X,$$

где n – количество разрядов числа A ; a_i – коэффициенты каждого разряда, которые могут принимать значения от 0 до $(X - 1)$.

При необходимости основание системы счисления указывается внизу после числа в виде нижнего индекса.

Примеры позиционных систем счисления, которые используются в цифровых устройствах, можно представить в следующем виде:

- Четырехразрядное десятичное число:

$5680_{10} = 5 \cdot 10^3 + 6 \cdot 10^2 + 8 \cdot 10^1 + 5 \cdot 10^0$, где $X = 10$ – основание системы счисления; $a_0 = 5$, $a_1 = 8$, $a_2 = 6$, $a_3 = 5$ – коэффициенты в каждом разряде; $n = 4$ – количество разрядов числа A .

- Трехразрядное восьмеричное число:

$372_8 = 3 \cdot 8^2 + 7 \cdot 8^1 + 2 \cdot 8^0 = 250_{10}$, где $X = 8$ (восьмеричная система счисления); коэффициенты в разрядах числа A : $a_0 = 2$, $a_1 = 7$; $n = 3$ – количество разрядов числа A .

- Двухразрядное шестнадцатеричное число:

$4E_8 = 4 \cdot 16^1 + 14 \cdot 16^0 = 78_{10}$, где $X = 16$ – шестнадцатеричная система счисления; коэффициенты в разрядах числа A : $a_0 = E = 14_{10}$, $a_1 = 4$ (шестнадцатеричные цифры от 0 до 9 записываются так же как и соответствующие десятичные цифры, а шестнадцатеричные цифры 10, 11, 12, 13, 14, 15 записываются заглавными латинскими буквами A, B, C, D, E, F соответственно); $n = 2$ – количество разрядов числа A .

- Четырехразрядное двоичное число:

$1101_2 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 13_{10}$, где $X = 2$ – двоичная система счисления; коэффициенты в разрядах числа A : $a_0 = 1$, $a_1 = 0$, $a_2 = 1$, $a_3 = 1$; $n = 4$ – количество разрядов числа A .

Двоичный разряд, который может принимать только два значения: 0 или 1, имеет название «бит», происходящее от сокращения английских слов **BI NARY DIGI T (BIT)** – «двоичная цифра». В английском языке слово *bit* означает также «кусочек».

Преобразование чисел из одной системы счисления в другую

Преобразовать десятичное число в двоичное можно путем деления на 2: сначала самого числа, а затем каждого промежуточного частного. При этом каждый неделимый остаток дает очередную цифру соответствующего разряда искомого двоичного числа. Первый, полученный таким образом остаток, даст цифру младшего разряда, а последний – старшего разряда двоичного числа. Например, десятичное число 53_{10} преобразуем в двоичное (рис. 1.1).

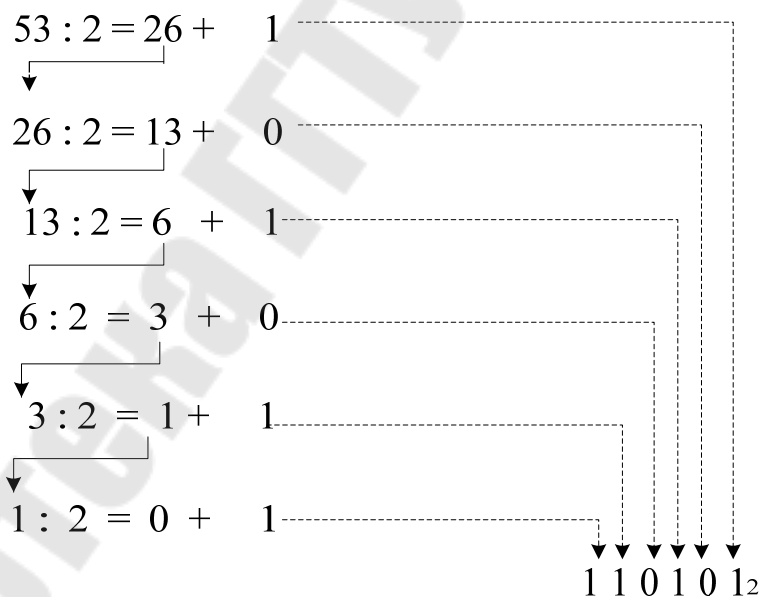


Рис. 1.1. Пример преобразования десятичного числа 53_{10} в двоичное

Аналогично можно преобразовывать числа с другими основаниями. Пример перевода десятичного числа 3480_{10} в шестнадцатеричное приведен ниже (рис. 1.2).

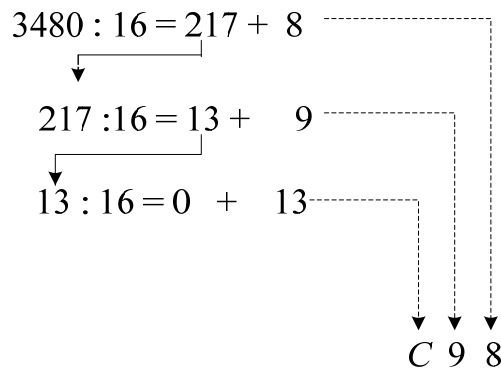


Рис. 1.2. Пример преобразования десятичного числа 3480_{10} в шестнадцатеричное

Так же как большие десятичные числа для удобства чтения разбивают при записи на тройки (к примеру, $234\ 010\ 034_{10}$), так и большие двоичные числа обычно разбивают на четверки – тетрады ($1001\ 1101\ 1000_2$).

В цифровой аппаратуре в основном при индикации показаний десятичными цифрами или при задании параметров десятичными задатчиками широко применяются различные двоично-десятичные коды.

Самый распространенный из них – BCD-код (сокращенное от *BINARY CODED DECIMAL* – двоично-кодированная десятичная цифра), который порой называют позиционным 8421-кодом или натуральным двоично-десятичным кодом. В этом коде каждая десятичная цифра представляется своей отдельной тетрадой – четверкой двоичных цифр, например:

$$496_{10} = \begin{array}{ccc} 4 & 9 & 6 \\ 0100 & 1001 & 0110_{BCD} \end{array}$$

Еще один распространенный двоично-десятичный код – код с избытком 3 (*EXCESS-3 CODE*). В нем каждая десятичная цифра кодируется двоичной тетрадой, в которой взвешенная сумма разрядов больше этой десятичной цифры на три. Так, десятичная цифра 9 записывается тетрадой 1100, для которой взвешенная сумма разрядов $8 \cdot 1 + 4 \cdot 1 + 2 \cdot 0 + 1 \cdot 0 = 12$, что на 3 больше числа девять.

В двоично-кодированных датчиках перемещения или угла поворота часто применяется код Грея (*GRAY CODE*). В этом коде комбинации двоичных цифр, представляющие числа, соседние по величине, отличаются лишь в одной кодовой позиции, т. е. при последовательном переходе от одного числа к другому всегда изменяется только один из двоичных разрядов.

Число B , записанное в двоичном коде, можно преобразовать в число G в коде Грея с помощью следующего выражения:
 $G_i = B_i \oplus B_{i+1}$.

Число G , записанное в коде Грея, можно преобразовать в число B в двоичном коде с помощью следующего выражения: $B_i = G_i \oplus B_{i+1}$.

Здесь знак \oplus означает сумму по модулю два, т. е. функцию, которая равна единице, если входные переменные различны, и – нулю в случае их равенства. Алгебраически это можно записать следующим образом:

$$\begin{aligned} 0 \oplus 0 &= 0; \\ 0 \oplus 1 &= 1; \\ 1 \oplus 0 &= 1; \\ 1 \oplus 1 &= 0. \end{aligned}$$

Число в коде Грея можно также получить из двоичного кода, пользуясь схемой, приведенной на рис. 1.3.

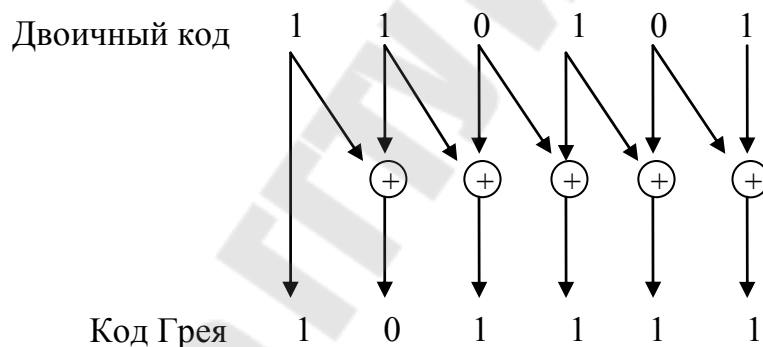


Рис. 1.3. Схема преобразования двоичного числа в код Грея

Обратное преобразование кода Грея в двоичный код производят по похожей схеме (рис. 1.4).

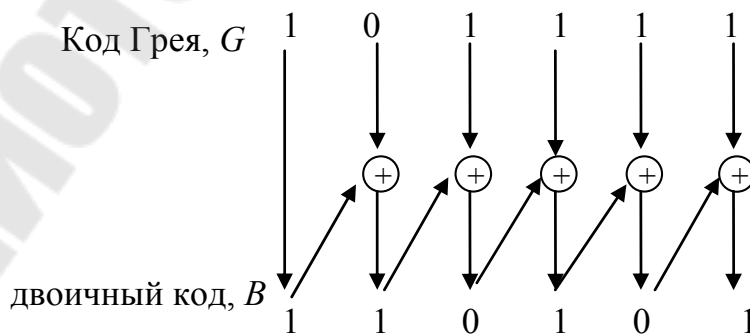


Рис. 1.4. Схема преобразования числа, представленного в коде Грея, в двоичное

Сложение многоразрядных чисел

Сложение двух двоичных чисел осуществляется точно так же как и в случае десятичных чисел. Более того, двоичное сложение даже проще, так как есть лишь несколько правил, которые необходимо будет запомнить. Рассмотрим десятичное сложение:

$$\begin{array}{r} 3 \ 7 \ 6 \\ + 4 \ 6 \ 1 \\ \hline 8 \ 3 \ 7. \end{array}$$

Сначала складываются младшие значащие разряды, которые в данном примере дают в сумме 7. После этого прибавляются разряды во второй позиции справа. Их сумма равна 13, поэтому делается перенос 1 из разряда десятков в старший разряд. Складывая ее с цифрами, которые там уже имеются, получаем 8.

Аналогичные операции нужно выполнить и в случае двоичного сложения. Однако теперь дело приходится иметь только с четырьмя возможными вариантами сложения, так как в любой позиции можно складывать лишь два двоичных числа (бита):

$$\begin{aligned} 0 + 0 &= 0; \\ 1 + 0 &= 1; \\ 1 + 1 &= 0 = 10 = 0 + \text{перенос } 1 \text{ в следующий разряд}; \\ 1 + 1 + 1 &= 11 = 1 + \text{перенос } 1 \text{ в следующий разряд}. \end{aligned}$$

Последний случай иллюстрирует ситуацию, когда складываются единицы, стоящие в одном разряде, а к ним после переноса из младшего разряда добавляется еще одна единица. Теперь рассмотрим несколько примеров сложения пары двоичных чисел (в скобках рядом приведены их десятичные эквиваленты):

$$\begin{array}{r} 0 \ 1 \ 1 \ (3) \\ + 1 \ 1 \ 1 \ (7) \\ \hline 1 \ 0 \ 0 \ 1 \ (10). \end{array} \qquad \begin{array}{r} 1 \ 0 \ 1 \ 1 \ (11) \\ + 1 \ 1 \ 1 \ 1 \ (15) \\ \hline 1 \ 1 \ 0 \ 1 \ 0 \ (26). \end{array}$$

Обратный код двоичного числа получают заменой каждого 0 на 1 и каждой 1 – на 0. Иначе говоря, каждый бит числа изменяется на противоположный (обратный). Этот процесс показан далее:

1	0	1	1	0	1	двоичное число
↓	↓	↓	↓	↓	↓	
0	1	0	0	1	0	обратный код двоичного числа.

Таким образом, обратный код числа 101101_2 будет равен 010010_2 .

Дополнительный код двоичного числа получают из обратного кода путем добавления 1 к его младшему значащему биту. Процесс преобразования числа $101101_2 = 45_{10}$ в обратный код показан далее:

1	0	1	1	0	1	двоичный эквивалент 45;	
0	1	0	0	1	0	каждый бит преобразуется в обратный код;	
+						1	к младшему разряду прибавляется 1;
0	1	0	0	1	1	дополнительный код двоичного числа 45.	

Таким образом, число 010011_2 будет дополнительным кодом числа 101101_2 .

Представление чисел со знаком в системе дополнительных кодов

Дополнительный код применяется для представления чисел со знаком. Это можно схематически представить следующим образом.

Если число положительное, то его модуль представляется в естественной форме двоичного числа, а знаковый бит содержит 0, который ставится перед старшим знаковым битом (рис. 1.5).

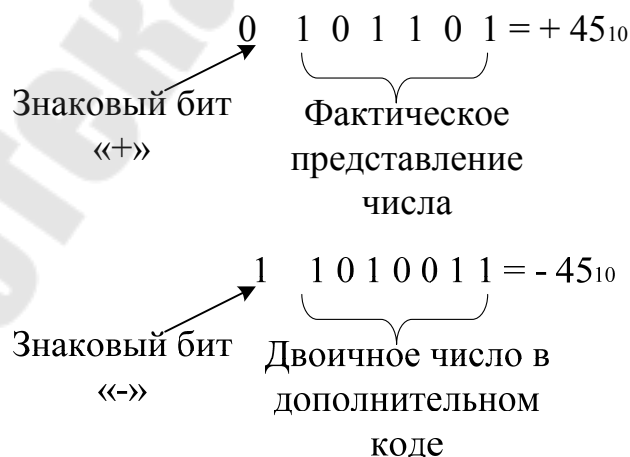


Рис. 1.5. Представление чисел со знаком в системе дополнительных кодов

Пример получения модуля отрицательного двоичного числа, представленного в дополнительном коде:

Знаковый бит	1	1	0	1	0	0	Отрицательное двоичное число -12_{10} в дополнительном коде;
	0	0	1	0	1	1	каждый бит преобразуется в обратный код;
+						1	к младшему разряду прибавляется 1;
	0	0	1	1	0	0	модуль отрицательного числа: $ -12 = 12_{10}$.

Таким образом, если знаковый бит у двоичного числа равен единице, это говорит о том, что число отрицательное и представлено в дополнительном коде. Для того чтобы узнать модуль этого числа, требуется проделать обратную операцию, т. е. проинвертировать каждый бит числа (перевести в обратный код), а затем к младшему разряду добавить единицу.

Дополнительный код используется для представления чисел со знаками, потому что этот код позволяет осуществлять операцию вычитания через сложение.

Сложение в системе дополнительных кодов

Теперь проанализируем, как осуществляются операции сложения и вычитания в цифровых машинах, которые для представления отрицательных чисел используют дополнительный код. Важно помнить, что в большинстве случаев со знаковыми битами обращаются так же как и с битами модуля:

- *Вариант I.* Два положительных числа.

Сложение двух положительных чисел выполняется побитно. Рассмотрим сложение чисел +9 и +4:

+9	→	0	1	0	0	1	первое слагаемое;	
+4	→	0	0	1	0	0	второе слагаемое;	
		0	1	1	0	1	сумма (+13).	
		↑						
		знаковые биты						

Важно отметить, что оба знаковых бита как первого, так и второго слагаемого равны 0, поэтому и знаковый бит суммы равен 0, что говорит о ее принадлежности к положительным числам. Также следует заметить, что оба слагаемых должны иметь одинаковое количество битов, так как это необходимо для использования дополнительного кода. Поэтому **всегда** нужно добавлять нули слева от значащей части числа, чтобы уравнять количество битов обоих чисел.

• *Вариант II.* Положительное число и отрицательное число, меньшее по модулю.

Рассмотрим сложение двух чисел: +9 и -4. Число -4 выражается в виде дополнительного кода. Таким образом, сначала надо перевести +4(00100) в -4(11100).

В этом случае знаковый бит второго слагаемого содержит 1. Обратите внимание, что теперь оба знаковых бита участвуют в процессе сложения. В последнем разряде модуля возникает перенос, который однако не учитывается:

		•						
		↓						
	знаковые биты							
+9 →	0	1	0	0	1	первое слагаемое;		
-4 →	1	1	1	0	0	второе слагаемое;		
	0	0	1	0	1	сумма (+5).		
±								

↑
этот перенос игнорируется

В основном при использовании дополнительного кода перенос из старшего разряда не учитывается, поэтому конечный результат равен 00101, что эквивалентно +5.

• *Вариант III.* Положительное число и отрицательное число, большее по модулю.

Рассмотрим сложение чисел -9 и +4:

		•					
-9 →	1	0	1	1	1	первое слагаемое	
+4 →	0	0	1	0	0	второе слагаемое	
	1	1	0	1	1	сумма (-5).	

↑
бит отрицательного знака

Полученная сумма имеет знаковый бит, содержащий 1, это говорит о том, что она отрицательна. Так как сумма меньше нуля, то она имеет вид дополнительного кода, т. е. только четыре младших бита (1011) представляют собой реальную величину числа (модуль). Чтобы получить абсолютное значение числа, необходимо применить операцию отрицания (получить дополнительный код числа) к величине 11011 и прибавить 1 к младшему разряду; результат будет равен $00101 = +5$. Таким образом, число 11011 с учетом знакового бита представляет собой число -5 .

• *Вариант IV.* Два отрицательных числа.

Сложим два отрицательных числа: -4 и -9 . Конечный результат будет отрицательным и представляется в виде дополнительного кода, т. е. его знаковый бит равен 1. Отрицание этого числа и добавление единицы к младшему разряду (получение его дополнительного кода) даст $01101 = +13$:

$$\begin{array}{r}
 \begin{array}{cccc}
 & & \text{знаковые биты} & \\
 & \swarrow & & \\
 -9 \rightarrow & 1 & 0 & 1 & 1 & 1 \\
 -4 \rightarrow & 1 & 1 & 1 & 0 & 0 \\
 \hline
 & 1 & 0 & 0 & 1 & 1
 \end{array} \\
 \begin{array}{l}
 \uparrow \\
 \text{этот перенос игнорируется}
 \end{array}
 \end{array}
 \quad \text{результат равен } 10011, \text{ сумма } (-13).$$

Вычитание в системе дополнительных кодов

Операция *вычитания* в системе дополнительных кодов по сути состоит из операции сложения и ничем не отличается от примеров, показанных ранее. При вычитании одного двоичного числа (*вычитаемого*) из другого двоичного числа (*уменьшаемого*) следуют такой процедуре:

- отрицание вычитаемого. Эта операция заменяет вычитаемое эквивалентной по модулю величиной, но с противоположным знаком;
- сложение полученной величины с уменьшаемым. Сумма такого сложения и будет представлять собой разность между уменьшаемым и вычитаемым.

Так же как и во всех математических операциях с применением дополнительного кода, необходимо, чтобы оба числа содержали одинаковое количество битов.

Задания для практического занятия

Номер варианта задания выбрать из табл. 1.1 согласно N – номеру студента в списке журнала группы.

Таблица 1.1

Варианты заданий

Номер варианта	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>
1	123	7145	00110111	A14	101
2	231	6567	11100110	B67	111
3	286	6012	10111010	60F	010
4	331	5567	11111000	55C	011
5	386	5012	00011111	501	100
6	441	4665	11010011	6E5	101
7	496	4600	01111010	46C	110
8	542	4515	11001110	F15	101
9	597	4450	11100101	4A5	111
10	652	4385	10111100	3AA	010
11	707	4321	10010111	AF1	011
12	174	6936	11100110	6EA	100
13	242	6456	11011100	64B	101
14	297	5901	10001111	590	110
15	342	5456	11110001	F54	101
16	397	4901	01101110	49E	111
17	452	4652	11001101	A52	010
18	507	4587	01110101	5F8	011
19	553	4502	11110001	4AB	100
20	608	4437	00111110	CF7	101
21	663	4372	10100111	BA2	110
22	192	6893	11101100	C24	101
23	253	6345	10011101	6F5	111
24	308	5890	11001011	F40	010
25	353	5345	01111001	A04	011
26	408	4890	00101111	4890	100
27	463	4639	11001101	4639	101
28	518	4574	10111001	4574	110
29	564	4489	00011111	4489	111
30	619	4424	11101010	4424	101

Необходимо выполнить следующие задания для практического занятия:

1. Перевести заданные числа A и B из десятичной системы в двоичную, восьмеричную и шестнадцатеричную системы счисления.

2. Записать числа A и B в BCD -коде и в коде с избытком 3 ($EXCESS-3 CODE$).

3. Перевести полученные двоичные числа A и B в код Грея.

4. Перевести число C , заданное в коде Грея, в двоичное, а затем – в десятичное.

5. Перевести число D , заданное в шестнадцатеричном коде, в десятичное, а затем – в двоичное и в код Грея.

6. Вычислить в двоичной системе счисления $F_1 = (A + B)$; $F_2 = (C - B)$; $F_3 = (B - D)$.

7. Вычислить в двоичной системе счисления произведения $F_4 = D \cdot E$; $F_5 = A \cdot E$ и частное $F_6 = C \div E$.

8. Для проверки правильности арифметических действий перевести результаты вычислений из двоичной системы счисления в десятичную. Сравнить полученные результаты.

Практическое занятие № 2
СОСТАВЛЕНИЕ БУЛЕВЫХ ВЫРАЖЕНИЙ
ПО ТАБЛИЦЕ ИСТИННОСТИ И ИХ МИНИМИЗАЦИЯ

Теоретические сведения

**Дизъюнктивные и конъюнктивные формы
записи функций алгебры логики**

Совершенной дизъюнктивной нормальной формой (СДНФ) называют наиболее полную форму записи логического выражения. Эта форма записи представляет собой сумму, каждое слагаемое которой является произведением всех входных аргументов или их инверсий, например:

$$F = \overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C} + A\overline{B}\overline{C} + ABC.$$

Совершенная дизъюнктивная нормальная форма является избыточной, но логические функции, записанные в СДНФ, легко сравнивать между собой, их удобно преобразовывать в таблицы истинности и составлять по ним карты Карно. Булево выражение, полученное из таблицы истинности логической функции, имеет совершенную дизъюнктивную нормальную форму.

В некоторых случаях более удобной формой записи логического выражения является **совершенная конъюнктивная нормальная форма (СКНФ)**. Это произведение сомножителей, каждый из которых является суммой всех входных аргументов или их инверсий, например:

$$F = (\overline{A} + B + \overline{C})(\overline{A} + B + C)(\overline{A} + \overline{B} + C)(A + B + C).$$

Так же как и СДНФ, СКНФ является явно избыточной.

**Преобразование таблицы истинности
в булево выражение**

Допустим, имеется логическая функция F для трех переменных A , B и C , заданная в виде таблицы истинности (табл. 2.1).

Смысл составления булева выражения по таблице истинности в том, чтобы показать, при каких сочетаниях входных переменных или их инверсий заданная функция F равна единице (при составлении выражения в СДНФ) (или нулю – при составлении выражения в СКНФ).

Запишем булево выражение в совершенной дизъюнктивной нормальной форме. Для этого выберем из таблицы истинности из

всех возможных восьми комбинаций входных переменных A , B и C те сочетания переменных, при которых функция F равна единице. В табл. 2.1 они записаны в виде *логических произведений* переменных P_0 , P_2 , P_3 и P_7 , причем переменная, имеющая значение «лог. 0», записывается с инверсией.

Таблица 2.1

Таблица истинности представления логической функции в совершенной дизъюнктивной нормальной форме

Номер	A	B	C	F	Примечания
0	0	0	0	1	$P_0 = \overline{A}\overline{B}\overline{C}$
1	0	0	1	0	
2	0	1	0	1	$P_2 = \overline{A}B\overline{C}$
3	0	1	1	1	$P_3 = \overline{A}BC$
4	1	0	0	0	
5	1	0	1	0	
6	1	1	0	0	
7	1	1	1	1	$P_7 = ABC$

Поскольку функция будет иметь такое значение при любом из наборов P_0 , P_2 , P_3 , P_7 независимо друг от друга, то их можно соединить между собой знаком ИЛИ (логическим сложением):

$$F = P_0 + P_2 + P_3 + P_7.$$

Каждый из наборов P_0 , P_2 , P_3 , P_7 является таким сочетанием входных переменных или их инверсий, которые только при совместном их воздействии обеспечивают единичное состояние выходной функции.

Следовательно, каждый такой набор состоит из всех входных переменных или их инверсий, связанных между собой функцией И, т. е. логическим умножением:

$$P_0 = \overline{A}\overline{B}\overline{C}; \quad P_2 = \overline{A}B\overline{C}; \quad P_3 = \overline{A}BC; \quad P_7 = ABC.$$

Исходя из этого, получаем результирующее выражение:

$$F = \overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C} + \overline{A}BC + ABC.$$

Как можно заметить, полученное выражение соответствует СДНФ.

Совершенная конъюнктивная нормальная форма (СКНФ) представляется логическим произведением дизъюнкций, каждая из которых содержит все переменные в прямом или инверсном виде не более одного раза. Рассмотрим табл. 2.2. Она аналогична табл. 2.1, только в ней выделены из всех комбинаций переменных A , B и C те сочетания, при которых функция F равна нулю. Эти комбинации обозначены S_1 , S_4 , S_5 , S_6 и записаны в виде *логической суммы* переменных или их инверсий, причем в этом случае инверсной считается переменная, принимающая значение «лог. 1».

Таблица 2.2

Таблица истинности представления логической функции в совершенной конъюнктивной нормальной форме

Номер	A	B	C	F	Примечания
0	0	0	0	1	
1	0	0	1	0	$S_1 = A + B + \bar{C}$
2	0	1	0	1	
3	0	1	1	1	
4	1	0	0	0	$S_4 = \bar{A} + B + C$
5	1	0	1	0	$S_5 = \bar{A} + B + \bar{C}$
6	1	1	0	0	$S_6 = \bar{A} + \bar{B} + C$
7	1	1	1	1	

Каждый из наборов S_1 , S_4 , S_5 , S_6 является таким сочетанием входных переменных и их инверсий, которые только при совместном их воздействии обеспечивает нулевое состояние выходной функции. Если записать выражение в виде *произведения* данных наборов, получим булево выражение, записанное в совершенной конъюнктивной нормальной форме:

$$F = (A + B + \bar{C})(\bar{A} + B + C)(\bar{A} + B + \bar{C})(\bar{A} + \bar{B} + C).$$

Минимизация логических функций алгебраическим способом

Основой минимизации алгебраическим способом является последовательное использование законов булевой алгебры и правил преобразований. Кроме законов, известных нам из обычной алгебры, в булевой алгебре типовыми приемами можно считать следующие:

- Многократное прибавление или умножение какого-либо переменного, или нескольких переменных, что не изменяет функцию, поскольку:

$$A + A + \dots + A; \quad ABC + ABC + \dots = ABC;$$

$$A \cdot A \cdot A \cdot \dots = A; \quad ABC \cdot ABC \cdot \dots = ABC.$$

- Умножение членов уравнения на сумму $A + \bar{A} = 1$, или сложение их с $A \cdot \bar{A} = 0$.

- Использование выражений и законов булевой алгебры:

$$A + A = A; \quad A \cdot A = A;$$

$$\overline{\overline{A}} = A; \quad A + \bar{A}B = A + B;$$

$$A + AB = A; \quad A \cdot (A + B) = A;$$

$$A + \bar{A} = 1; \quad A \cdot \bar{A} = 0;$$

$$A + (BC) = (A + B)(A + C); \quad A + 1 = 1.$$

Пример минимизации логической функции алгебраическим способом:

$$F = \bar{A}BC + A\bar{B}C + ABC\bar{C} + ABC = \bar{A}BC + A\bar{B}C + ABC\bar{C} +$$

$$+ ABC + (ABC + ABC).$$

В форму записи функции добавим слагаемое $(ABC + ABC)$, после чего перегруппируем слагаемые таким образом, чтобы вынести за скобки общий множитель:

$$F = \bar{A}BC = ABC + A\bar{B}C + ABC + ABC\bar{C} + ABC =$$

$$= (A + \bar{A})BC + (B + \bar{B})AC + (C + \bar{C})AB = AB + DC + AC.$$

Задания для практического занятия

Номер варианта задания выбрать из табл. 2.3 согласно следующим критериям:

- а) по N – номеру студента в списке журнала группы;
- б) равным $(30 - N)$, где N – номер студента в списке журнала группы.

Таблица 2.3

Варианты заданий

Номер варианта	F_1	F_2
1	00110111	0001011011110000
2	11100110	0011100000011100
3	10111010	1100001110100000
4	11111000	1110001100000111
5	00011111	0000011100110011
6	11010011	1111010000011000
7	01111010	0000111101100000
8	11001110	1100001100011100
9	11100101	1110000011000010
10	10111100	0011000111100110
11	10010111	1110000011000001
12	11100110	0001110000011101
13	11011100	1001100011100001
14	10001111	0001000111000111
15	11110001	0011110000000111
16	01101110	1111010000011000
17	11001101	0011000111100110
18	01110101	0001011011110000
19	11110001	0011000111100110
20	00111110	0001011011110000
21	10100111	1100001110100000
22	11101100	1001100011100001
23	10011101	1001100011100001
24	11001011	0001000111000111
25	01111001	0011000111100110
26	00101111	0011110000000111
27	11001101	1100001110100000
28	10111001	0001110000011101
29	00011111	0011110000000111
30	11101010	0001011011110000

Необходимо выполнить следующие задания для практического занятия:

1. В столбец F_1 таблицы истинности (табл. 2.4) вписать заданное восьмиразрядное двоичное число F_1 , взятое согласно номеру варианта из табл. 2.3.

Таблица 2.4

Таблица истинности функции F_1

A	B	C	F_1
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

2. Представить логическую функцию, заданную таблицей истинности, в конъюнктивной совершенной нормальной форме.

3. Представить логическую функцию, заданную таблицей истинности, в дизъюнктивной совершенной нормальной форме.

4. Выполнить минимизацию функций, записанных в СДНФ, используя метод непосредственных преобразований.

5. В столбец F_2 таблицы истинности (табл. 2.5) вписать заданное шестнадцатиразрядное двоичное число F_2 .

Таблица 2.5

Таблица истинности функции F_2

A	B	C	D	F_2
0	0	0	0	
0	0	1	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	

Окончание табл. 2.5

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>F₂</i>
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

6. Представить логическую функцию, заданную таблицей истинности, в конъюнктивной совершенной нормальной форме.

7. Представить логическую функцию, заданную таблицей истинности, в дизъюнктивной совершенной нормальной форме.

8. Выполнить минимизацию функций, записанных в СДНФ, используя метод непосредственных преобразований.

Практическое занятие № 3

СИНТЕЗ КОМБИНАЦИОННЫХ СХЕМ ПО БУЛЕВЫМ ВЫРАЖЕНИЯМ

Теоретические сведения

Минимизация логических функций по картам Карно

Для минимизации логических функций очень удобно пользоваться *картами Карно* или очень схожими с ними диаграммами Вейча.

Карта Карно изображает в виде графических квадратов (клеток) все возможные комбинации переменных, причем переменные, определяющие координаты клеток карты, размещают так, чтобы при переходе из одной клетки в соседнюю, как по горизонтали, так и по вертикали, изменялась только одна переменная.

Таблица истинности для четырех переменных включает 16 строк, следовательно, карта Карно должна состоять из 16 клеток, как показано на рис. 3.1

	$\overline{A}\overline{B}$	$\overline{A}B$	AB	$A\overline{B}$
$\overline{C}\overline{D}$				
$\overline{C}D$				
CD				
$C\overline{D}$				

Рис. 3.1. Пример карты Карно для четырех переменных

Если требуется получить карту Карно для какой-либо функции, сначала надо записать эту функцию в СДНФ – в совершенной дизъюнктивно нормальной форме, или в виде таблицы истинности. Каждое слагаемое булева выражения в СДНФ, или каждая единица в столбце функции таблицы истинности задается на карте Карно единицей в соответствующей клетке. Координаты этой клетки содержат те же входные переменные и их инверсии, что и данное слагаемое СДНФ булева выражения (или данная строка таблицы истинности).

У карты Карно для четырех переменных клетки крайнего левого столбца должны рассматриваться как соседние для клеток крайнего правого столбца, а клетки верхней строки – как соседние для клеток нижней строки. Другими словами, можно сказать, что эта карта расположена на поверхности цилиндра (склеили правый край карты с ле-

вым), изогнутого и растянутого так, что его верхний срез соединяется с нижним срезом; при этом цилиндр превращается в тор.

Правила упрощения заполненной карты Карно заключаются в следующем:

- соседние две, четыре, восемь, или другое число единиц, равное степени двойки, обводят общим контуром;
- контур должен быть прямоугольным без изгибов или наклонов;
- каждый контур превращает все входящие в него единицы в одну, т. е. объединенные таким образом слагаемые СДНФ булева выражения дают одно слагаемое в упрощенном выражении;
- те входные переменные, которые входят в координаты данного контура совместно со своими инверсиями, исключаются из слагаемого, которое дает этот контур в упрощенном выражении.

Рассмотрим примеры упрощения булевых выражений с помощью карты Карно для четырех переменных.

Пример 1. Упростим при помощи карты Карно логическую функцию, заданную выражением:

$$F_1 = \bar{A}BC\bar{D}_1 + A\bar{B}C\bar{D}_2 + \bar{A}B\bar{C}D_3 + A\bar{B}\bar{C}D_4 + \bar{A}\bar{B}C\bar{D}_5 + \bar{A}B\bar{C}D_6.$$

Рассмотрим карту Карно, представленную на рис. 3.2.

	$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$	
$\bar{C}\bar{D}$		1 ₁	1 ₂		→ $B\bar{C}_{1,2,3,4}$
$\bar{C}D$		1 ₃	1 ₄		
$C\bar{D}$					
CD		1 ₅	1 ₆		→ $\bar{A}C\bar{D}_{5,6}$

Рис. 3.2. Пример минимизации булевой функции F_1 с помощью карты Карно для четырех переменных

В примере 1 минимизации булевой функции F_1 нижний контур из двух единиц 1₅ и 1₆, соответствующих пятому и шестому слагаемым в исходном булевом выражении, дает возможность опустить B и \bar{B} .

После этого в нем остается произведение $\bar{A}C\bar{D}$. В верхнем контуре из четырех единиц 1₁, 1₂, 1₃ и 1₄, соответствующих первым четырем слагаемым в исходном булевом выражении, попарно опускаются A и \bar{A} , D и \bar{D} , так что в результате этого верхний контур дает произведение $B\bar{C}$.

Полученное упрощенное выражение: $F_1 = BC + \bar{A}C\bar{D}$.

Пример 2. Упростим при помощи карты Карно логическую функцию, заданную выражением:

$$F_2 = \overline{A}\overline{B}\overline{C}\overline{D}_1 + \overline{A}B\overline{C}\overline{D}_2 + A\overline{B}\overline{C}\overline{D}_3 + \overline{A}\overline{B}C\overline{D}_4 + \overline{A}B\overline{C}D_5.$$

Во этом примере минимизации булевой функции F_2 (рис. 3.3) контур из двух единиц 1_2 и 1_3 , соответствующих второму и третьему слагаемым в исходном булевом выражении, дает возможность опустить A и \overline{A} . После этого в нем остается произведение $B\overline{C}\overline{D}$. В контуре из четырех единиц $1_1, 1_2, 1_4$ и 1_5 , соответствующих другим четырем слагаемым из исходного булева выражения, попарно опускаются B и \overline{B} , C и \overline{C} , так что в результате этого верхний контур дает произведение $\overline{A}\overline{D}$.

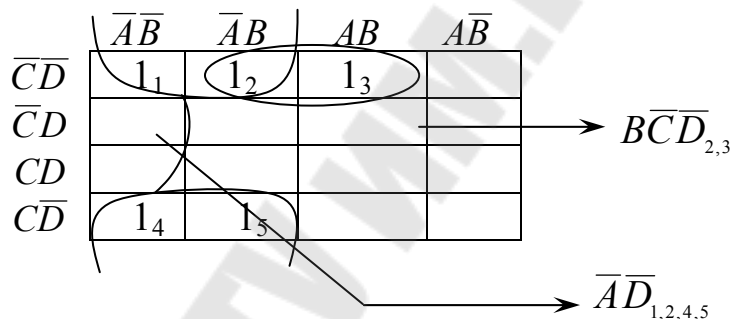


Рис. 3.3. Пример минимизации булевой функции F_2 с помощью карты Карно для четырех переменных

Полученное упрощенное выражение: $F_2 = \overline{A}\overline{D} + B\overline{C}\overline{D}$. Карта Карно представляется в данном случае свернутой в цилиндр, в котором верхний край совмещается с нижним.

Этот пример показывает также, что контуры могут накладываться друг на друга (сколько угодно раз).

Построение комбинационных логических схем по заданным булевым выражениям

Построим схему для реализации двухвходовой функции «Исключающее ИЛИ», булево выражение которого в двухвходовом случае совпадает с булевым выражением для сумматора по модулю счета 2 и имеет следующий вид: $F = A\overline{B} + \overline{A}B$.

1. *Первый этап*: выполняется логическое сложение, логическая операция ИЛИ, считая входными переменными функции $A\overline{B}$ и $\overline{A}B$ (рис. 3.4).

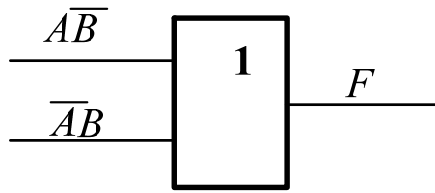


Рис. 3.4. Первый этап построения комбинационных логических схем по заданному булеву выражению

2. *Второй этап:* к входам элемента ИЛИ подключаются логические элементы И, входными переменными которых являются уже A и B и их инверсии (рис. 3.5).

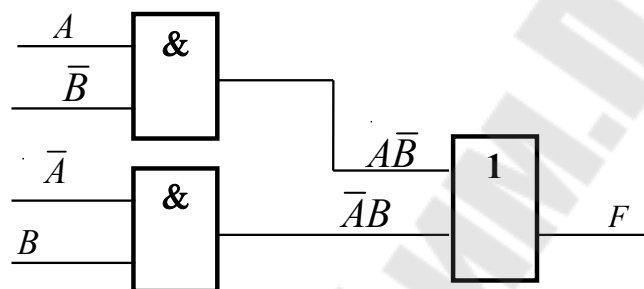


Рис. 3.5. Второй этап построения комбинационных логических схем по заданному булеву выражению

3. *Третий этап:* для получения инверсий \bar{A} и \bar{B} на соответствующих входах ставят инверторы (рис. 3.6).

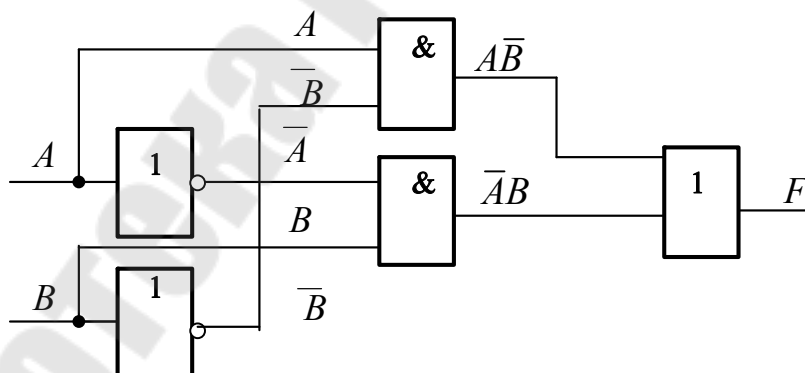


Рис. 3.6. Третий этап построения комбинационных логических схем по заданному булеву выражению

Данное построение основано на следующей особенности – поскольку значениями логических функций могут быть только нули и единицы, то любые логические функции могут быть представлены как аргументы других более сложных функций.

Таким образом, построение комбинационной логической схемы осуществляется от выхода к входу.

Особенности построения логических схем в инвертирующих базисах

Первой особенностью построения логических схем в инвертирующих базисах считается не прямая зависимость между простотой булева выражения и минимальностью соответствующей ему логической схемы. Другими словами, самое минимизированное булево выражение не всегда дает схему, минимальную по количеству инвертирующих логических элементов.

Вторая особенность базируется на принципе де Моргана о замене инверсии сложения умножением инверсий и наоборот:

$$\overline{A + B} = \overline{A} \overline{B} \quad \text{и} \quad \overline{AB} = \overline{A} + \overline{B}.$$

Исходя из этого можно утверждать:

- если в произвольной комбинационной схеме все логические элементы заменить на ЛЭ И-НЕ, то реализуемая схемой функция не изменится;

- если в произвольной комбинационной схеме все логические элементы заменить на ЛЭ ИЛИ-НЕ и при этом проинвертировать все входные и выходные сигналы, то реализуемая схемой функция не изменится.

Рассмотрим реализацию булева выражение для двухвходовой функции «Исключающее ИЛИ» в базисе И-НЕ (рис. 3.7). Заменяем в предыдущей схеме (рис. 3.6) все логические элементы на ЛЭ И-НЕ.

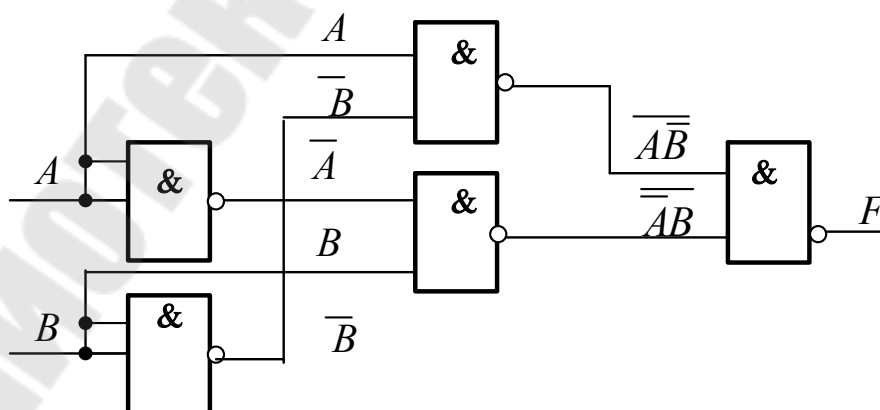


Рис. 3.7. Синтез функции «Исключающее ИЛИ» в инвертирующем базисе И-НЕ

Проверим алгебраически соответствие полученной булевой функции заданной:

$$F = \overline{\overline{A}B} \cdot \overline{A\overline{B}} = \overline{\overline{A}B + A\overline{B}} = \overline{A}B + A\overline{B}.$$

Рассмотрим пример синтеза схемы в базисе ИЛИ-НЕ двухвходовой функции «Исключающее ИЛИ» по следующему булеву выражению: $F = B\overline{A} + A\overline{B} = \overline{\overline{B}A} + \overline{\overline{A}B} = \overline{\overline{B} + A} + \overline{\overline{B} + A}$ (рис. 3.8).

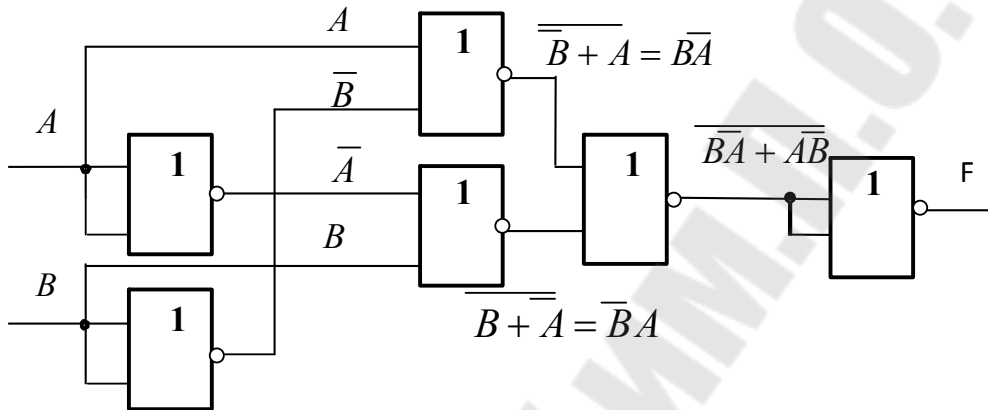


Рис. 3.8. Синтез функции «Исключающее ИЛИ» в инвертирующем базисе ИЛИ-НЕ

Задания для практического занятия

Номер варианта задания выбрать из табл. 3.3 согласно следующим критериям:

- а) по N – номеру студента в списке журнала группы;
- б) равным $(30 - N)$, где N – номер студента в списке журнала группы.

Таблица 3.1

Вариант заданий

Номер варианта	F_1	F_2
1	00110101	1111010011011000
2	11000110	0011000111110110
3	10110010	1001011011110000
4	11011000	1111000111100110
5	00011101	1001011011110011
6	11010011	1100001110111011
7	01101010	1001101111110001
8	11001100	1001100011100001
9	10100101	1101000111000111
10	10111100	1111011100011000
11	10010111	1110001111110001
12	10100110	1001110011011111
13	11011000	1111100011100001
14	10001110	0001111111000111
15	11110000	0011110000111111
16	01101100	1111010000011000
17	11001100	1001100011100001
18	01110001	1001100011100001
19	11110001	1111000111000111
20	00111110	0011110111100110
21	00100111	0011111111000111
22	10101100	1100001110100000
23	10011101	1001110000011111
24	11001010	1111011100011001
25	01101001	1110011011000001
26	00101011	0001111110011101
27	11001101	1001101111100001
28	10101001	0001000111111111
29	00011101	0011110011000111
30	10101010	1111010011111000

Необходимо выполнить следующие задания для практического занятия:

1. В последний столбец таблицы истинности (табл. 3.2) вписать заданное восьмиразрядное двоичное число F_1 , взятое согласно номеру варианта из табл. 3.1. а в последний столбец таблицы истинности (табл. 3.3) – шестнадцатиразрядное двоичное число F_2 .

Таблица 3.2

Таблица истинности функции F_1

A	B	C	F_1
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Таблица 3.3

Таблица истинности функции F_2

A	B	C	D	F_2
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

2. Представить логические функции F_1 и F_2 , заданные таблицами истинности, в совершенной дизъюнктивной нормальной форме.

3. Выполнить минимизацию функций F_1 и F_2 , записанных в СДНФ, при помощи карт Карно.

4. Построить схемы, реализующие заданные функции, согласно упрощенным по карте Карно булевым выражения в отрицательной логике.

Исходя из условия заданий реализовать: вариант а) – в базисе И-НЕ; вариант б) – в базисе ИЛИ-НЕ.

Практическое занятие № 4 СИНТЕЗ ПРЕОБРАЗОВАТЕЛЕЙ КОДОВ

Теоретические сведения

Преобразователи кодов

Под построением преобразователей кодов понимается синтез схем, реализующих преобразование n -разрядных двоичных чисел, представляющих информацию в каком-либо входном заданном коде, в m -разрядные двоичные числа, представляющие эту информацию в другом коде (выходном). Наиболее распространены следующие два подхода к построению преобразователей кодов.

Первый подход – синтез n независимых одновыходных функций по заданной таблице истинности – таблице соответствия кодов.

Допустим, имеется таблица истинности преобразователя 2-разрядного двоичного кода a_1a_0 в 3-разрядный $b_2b_1b_0$ (табл. 4.1).

Таблица 4.1

Таблица истинности преобразователя кодов

a_1	a_0	b_2	b_1	b_0
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	1	0

Считая b_0 , b_1 и b_2 независимыми одновыходными функциями, запишем для каждой из них булевы выражения:

$$b_2 = \bar{a}_1\bar{a}_0 + a_1a_0;$$

$$b_1 = a_1\bar{a}_0 + a_1a_0;$$

$$b_0 = \bar{a}_1a_0.$$

Используя приведенные булевы выражения и учитывая, что произведение $a_1\bar{a}_0$ встречается в двух выходных функциях, составляем схему преобразователя на логических элементах (рис. 4.1).

Второй способ построения преобразователей кодов – синтез схемы по принципу «дешифратор-шифратор». Для того чтобы синтезировать схему преобразователя кодов по табл. 4.1, требуется иметь представление о структуре комбинационных устройств: дешифратора и шифратора.

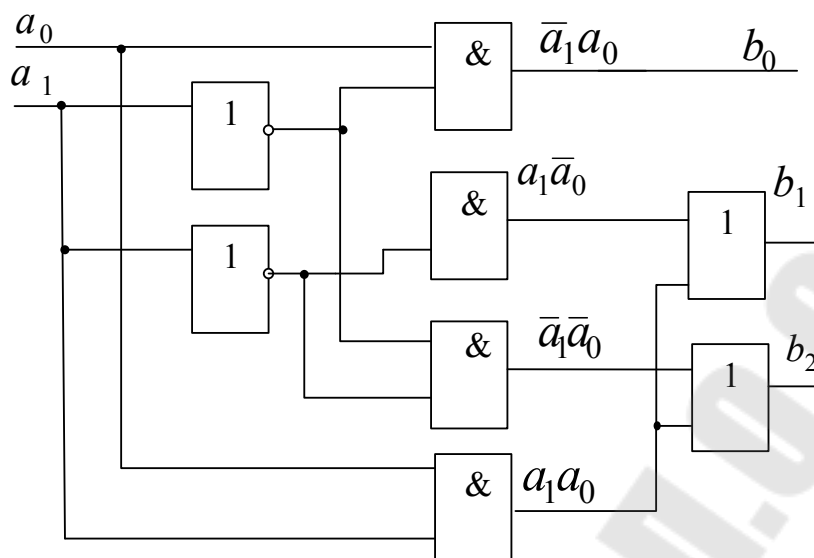


Рис. 4.1. Синтез преобразователя кодов по заданному булеву выражению

Рассмотрим, что представляет собой дешифратор.

Дешифратор – это цифровое устройство, в котором входной двоичный n -разрядный код преобразуется в выходной унарный (один из всех), т. е. активный уровень присутствует только на одном из выходов, а именно – на том, номер которого соответствует этой цифре в двоичном коде. Активным может быть как высокий уровень («H» – High, т. е. единица для положительной логики, так и низкий («L» – Low, т. е. ноль для отрицательной логики).

Число выходов дешифратора $N = 2^n$, где n – число разрядов входного кода дешифратора. Составим таблицу истинности 2-разрядного дешифратора по форме табл. 4.2.

Таблица 4.2

Таблица истинности 2-разрядного дешифратора

Входы		Выходы			
a_1	a_0	d_3	d_2	d_1	d_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

Учитывая, что активный уровень появляется на каждом из выходов дешифратора только для одной из входных комбинаций, булевы выражения выходных функций дешифратора содержат только одно произведение входных переменных или их дополнений, а в правой части таблицы истинности единица может быть только в одном столбце каждой строки.

Поэтому схема дешифратора будет содержать только логические элементы И (конъюнкторы), перемножающие входные переменные и их дополнения, реализованные по булевым выражениям, полученным из таблицы состояний: $d_0 = \bar{a}_1\bar{a}_0$, $d_1 = \bar{a}_1a_0$, $d_2 = a_1\bar{a}_0$, $d_3 = a_0a_1$.

На рис. 4.2 показана схема 2-разрядного дешифратора, построенная на логических элементах И по приведенной выше таблице истинности в форме табл. 4.2.

В случае, если в схеме многоразрядного дешифратора не используется какая-либо входная комбинация переменных, такой дешифратор называется *неполным*.

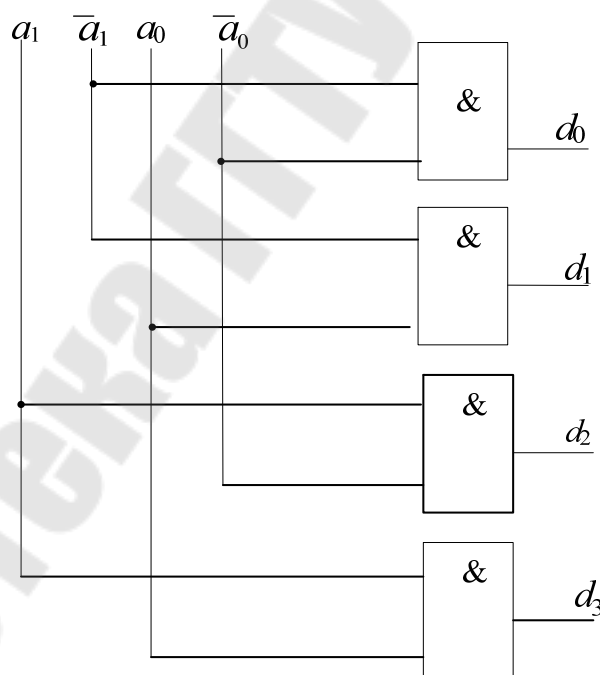


Рис. 4.2. Дешифратор 2-разрядный

На рис. 4.3 изображено условное графическое изображение дешифратора с разрядностью 3 : 8.

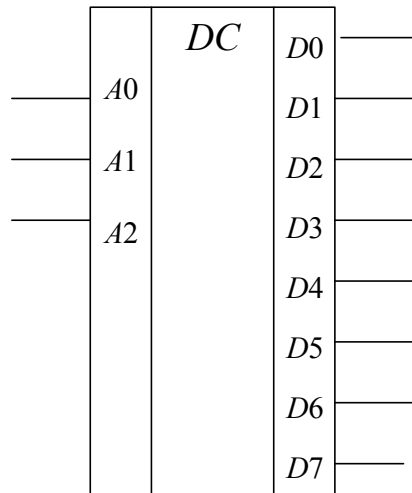


Рис. 4.3. Условное графическое изображение дешифратора с разрядностью 3 : 8

Шифратор выполняет функцию, обратную дешифратору. Только один из входов шифратора может быть активным, остальные должны оставаться пассивными, т. е. логическую единицу можно подавать только на один из входов, а на все остальные входы при этом надо подавать нули (для положительной логики, для отрицательной – наоборот).

Число входов такого преобразователя равно $M = 2^n$, где n – число разрядов выходного кода шифратора. Учитывая это обстоятельство, составили таблицу истинности для 3-разрядного шифратора (табл. 4.3).

Таблица 4.3

Таблица истинности для 3-разрядного шифратора

Входы	Выходы		
	b_2	b_1	b_0
d_0	0	0	0
d_1	0	0	1
d_2	0	1	0
d_3	0	1	1
d_4	1	0	0
d_5	1	0	1
d_6	1	1	0
d_7	1	1	1

Булевы выражения на основании табл. 4.3 выглядят следующим образом:

$$b_0 = d_1 + d_3 + d_5 + d_7; \quad b_1 = d_2 + d_3 + d_6 + d_7; \quad b_2 = d_4 + d_5 + d_6 + d_7.$$

Схема шифратора, построенная по этим выражениям, должна содержать только элементы ИЛИ (дизъюнкторы) для положительной логики, суммирующие те входные переменные, для которых есть единица в соответствующем разряде выходного кода (или только элементы И (конъюнкторы) для отрицательной логики).

На рис. 4.4 представлена схема, реализующая 3-разрядный шифратор.

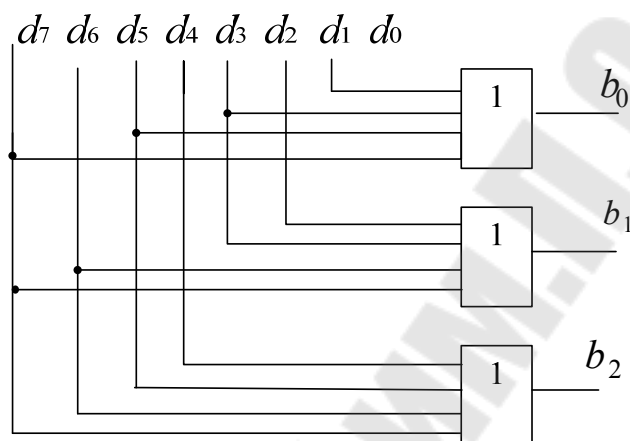


Рис. 4.4. Шифратор 3-разрядный

Если в схеме преобразователя кодов один из выходных разрядов всегда неактивный, то соответствующий элемент ИЛИ не включается в схему шифратора и такой шифратор также называется *неполным*.

На рис. 4.5 изображено условное графическое изображение дешифратора с разрядностью 8 : 3

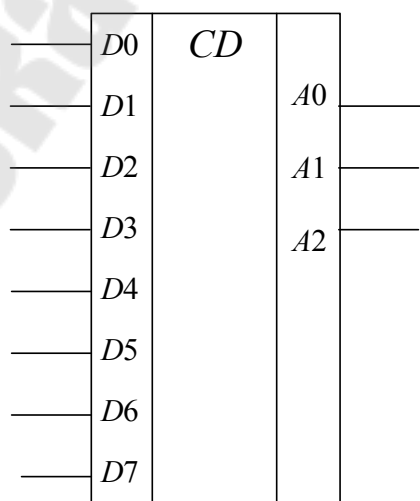


Рис. 4.5. Условное графическое изображение шифратора с разрядностью 8 : 3

Рассмотрим метод построения преобразователя кодов способом «дешифратор-шифратор».

Схема преобразователя кода образуется соединением выходов дешифратора и входов шифратора в соответствии с таблицей входных и выходных переменных преобразователя при переводе двоичного кода в десятичный. Таблица 4.1 в этом случае преобразуется в табл. 4.4.

Таблица 4.4

Таблица истинности преобразователя 2-разрядного кода в 3-разрядный

a_1	a_0	b_2	b_1	b_0	A_{10}	B_{10}
0	0	1	0	0	0	4
0	1	0	0	1	1	1
1	0	0	1	0	2	2
1	1	1	1	0	3	6

Преобразователь кодов, синтезированный вторым способом по табл. 4.4, представлен на рис. 4.6.

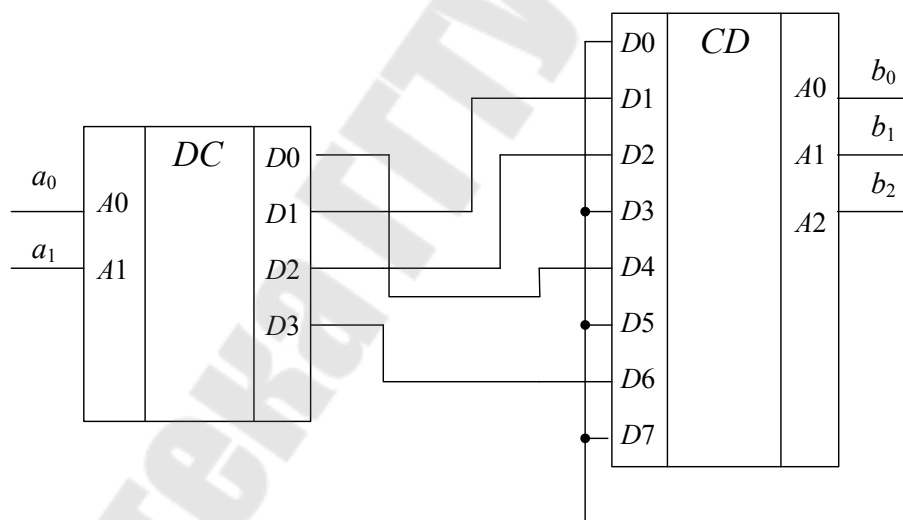


Рис. 4.6. Преобразователь кодов, синтезированный способом «дешифратор-шифратор»

Задания для практического занятия

Номер варианта задания выбрать из табл. 4.5 согласно N – номеру студента в списке журнала группы.

Таблица 4.5

Номер варианта	Y	Номер варианта	b_2	b_1	b_0
1	00110111	1, 6, 12, 5, 28	1	1	0
2	11100110		0	0	1
3	10111010		0	1	1
4	11111000		0	1	1
5	00011111		1	0	0
6	11010011	2, 7, 11, 17, 20	1	1	0
7	01111010		1	0	1
8	11001110		0	1	0
9	11100101		0	1	1
10	10111100		0	0	0
11	10010111		3, 5, 10, 14, 23	1	0
12	11100110	1		0	1
13	11011100	1		1	1
14	10001111	0		1	0
15	11110001	0		1	0
16	01101110	26, 8, 30, 16, 24	1	0	1
17	11001101		0	0	0
18	01110101		1	1	1
19	11110001		1	0	1
20	00111110		0	1	1
21	10100111	29, 13, 18, 21, 25	1	1	1
22	11101100		0	1	0
23	10011101		1	1	0
24	11001011		1	0	0
25	01111001		0	0	1
26	00101111	27, 4, 9, 19, 22	1	0	0
27	11001101		0	1	1
28	10111001		1	1	1
29	00011111		1	1	0
30	11101010		0	0	1

Необходимо выполнить следующие задания для практического занятия:

1. В последний столбец таблицы истинности (табл. 4.6) вписать 8-разрядное двоичное число Y , взятое из табл. 4.5, получить таблицу истинности дешифратора функции Y для четырех входных переменных.

Таблица 4.6

**Таблица истинности дешифратора функции Y
для четырех входных переменных**

a_3	a_2	a_1	a_0	Y
0	0	1	1	
0	1	0	1	
0	0	1	0	
0	1	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

2. Получить таблицу истинности (табл. 4.7) преобразователя 4-разрядного $(a_3 a_2 a_1 a_0)$ двоичного кода в 3-разрядный $(b_2 b_1 b_0)$. Из комбинаций входных переменных выбирать те, для которых функция Y принимает значение единицы; выходные переменные $(b_2 b_1 b_0)$ взять из табл. 4.5.

Таблица 4.7

**Таблица истинности преобразователя 4-разрядного кода
в 3-разрядный**

Входы				Выходы		
a_3	a_2	a_1	a_0	b_2	b_1	b_0

3. Выполнить синтез преобразователя 4-разрядного $(a_3 a_2 a_1 a_0)$ двоичного кода в 3-разрядный $(b_2 b_1 b_0)$, используя логические элементы И-НЕ.

4. Получить таблицу истинности (табл. 4.8) и выполнить синтез преобразователя 3-разрядного $(b_2 b_1 b_0)$ двоичного кода в 4-разрядный, $(a_3 a_2 a_1 a_0)$, используя логические элементы ИЛИ-НЕ.

Таблица 4.8

Таблица истинности преобразователя 3-разрядного кода в 4-разрядный

Входы			Выходы			
b_2	b_1	b_0	a_3	a_2	a_1	a_0

5. Выполнить синтез преобразователя 4-разрядного $(a_3 a_2 a_1 a_0)$ двоичного кода в 3-разрядный $(b_2 b_1 b_0)$ и преобразователя 3-разрядного $(b_2 b_1 b_0)$ двоичного кода – в 4-разрядный, $(a_3 a_2 a_1 a_0)$ методом «шифратор-дешифратор».

Практическое занятие № 5

МУЛЬТИПЛЕКСОР КАК УНИВЕРСАЛЬНЫЙ ЛОГИЧЕСКИЙ ЭЛЕМЕНТ

Теоретические сведения

Мультиплексор

Мультиплексированием (*MULTIPLEX*) называют передачу данных от нескольких источников по одному каналу поочередно.

Мультиплексор (или селектор данных) – это функциональное устройство, осуществляющее подключение (коммутацию) одного из нескольких информационных входов данных к выходу. Номер выбранного входа соответствует двоичному коду, подаваемому на адресные входы мультиплексора. В цифровой схемотехнике мультиплексор имеет один выход; m информационных входов данных $D_0, D_1, D_2, D_3, \dots, D_m$ и n адресных входов $A_1, A_2, A_3, \dots, A_n$, причем $m = 2^n$.

На рис. 5.1 приведен пример условного графического обозначения мультиплексора на схемах. Число информационных входов D_0 – D_3 , коммутируемых на выход Y , составляет $m = 4$. Такой мультиплексор имеет размерность 4 : 1.



Рис. 5.1. Условное графическое обозначение мультиплексора 4 : 1 с разрешающим входом

Булево выражение для выходной функции мультиплексора выглядит следующим образом:

$$Y = \overline{A_1}A_0D_0 + \overline{A_1}A_0D_1 + A_1A_0D_2 + A_1A_0D_3.$$

При учете разрешающего входа E , булево выражение запишется следующим образом:

$$Y = E(\overline{A1A0}D0 + \overline{A1}A0D1 + A1A0D2 + A1A0D3).$$

Представим таблицу истинности мультиплексора с двумя адресными входами в форме табл. 5.1.

Таблица 5.1

Таблица истинности мультиплексора с двумя адресными входами

E	$A1$	$A0$	Y
1	0	0	$D0$
1	0	1	$D1$
1	1	0	$D2$
1	1	1	$D3$

Наличие разрешающего входа E позволяет увеличивать число информационных входов вдвое путем последовательного соединения разрешающих входов двух мультиплексоров (наращивание разрядности).

Кроме прямого назначения мультиплексор может быть использован как универсальный логический элемент, т. е. на основе мультиплексора возможна реализация любой заданной логической функции.

Если на адресные входы мультиплексора подать входные переменные, зная, какой выходной уровень должен отвечать каждой комбинации этих переменных, то, предварительно установив на входах данных уровни напряжения «лог. 1» и «лог. 0», согласно таблице истинности, можно получить устройство, реализующее заданную логическую функцию.

Использование мультиплексоров дает следующие преимущества при синтезе комбинационных схем:

- не требуется упрощения логического выражения, составленного по таблице истинности;
- минимизируется число требуемых интегральных схем;
- соответственно, синтез комбинационных схем упрощается.

Для реализации заданной таблицей истинности логической функции с использованием мультиплексора необходимо определить десятичные номера каждого из логических выражений таблицы истинности, для которых выходной сигнал принимает значение: $Y = 1$. Входы мультиплексора, соответствующие этим номерам, соединить с «лог. 1». Все остальные входы – соединить с «лог. 0.» Входные пе-

ременные заданной функции – подать на адресные входы в соответствии с весом разрядов.

Рассмотрим пример: пусть требуется реализовать на мультиплексоре логическую функцию, заданную таблицей истинности (табл. 5.2).

Таблица 5.2

Таблица истинности логической функции

b_2	b_1	b_0	B_{10}	Y
0	0	0	0	0
0	0	1	1	1
0	1	0	2	0
0	1	1	3	1
1	0	0	4	0
1	0	1	5	0
1	1	0	6	1
1	1	1	7	1

Определим десятичные номера каждого из логических выражений таблицы истинности, для которых выходной сигнал принимает значение: $Y = 1$. Это десятичные цифры 1, 3, 6, 7. На входы данного мультиплексора, соответствующие этим номерам, подадим «лог. 1», остальные соединим с землей.

Входные переменные заданной функции ($b_2 b_1 b_0$) подадим на адресные входы мультиплексора. Реализация заданной логической функции на мультиплексоре представлена на рис. 5.2.

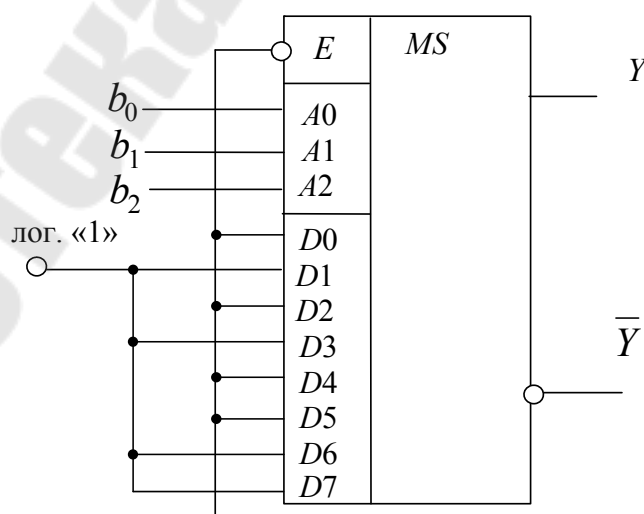


Рис. 5.2. Реализация логической функции, заданной табл. 5.2

Рассмотрим пример реализации функции $(m + 1)$ переменных на мультиплексорах с m адресными входами. Реализуем функцию четырех переменных $(b_3 b_2 b_1 b_0)$, заданную таблицей истинности (табл. 5.3) при помощи мультиплексора $8 : 1$.

Таблица 5.3

Таблица истинности логарифмической функции четырех переменных

Переменные				Значение функции
b_3	b_2	b_1	b_0	Y
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1

Функция четырех переменных может быть реализована на мультиплексоре размерностью $8 : 1$ следующим образом: переменные заданной функции $(b_2 b_1 b_0)$ подключим к адресным входам A_2, A_1, A_0 соответственно. Переменную b_3 , не подключенную к адресным входам, называют **выделенной**. Без выделенной переменной наборы переменных $b_2 b_1 b_0$ образуют несколько пар неизменных значений. В табл. 5.4 эти пары выделены пунктирными линиями.

Теперь рассмотрим соотношения между выделенной переменной b_3 и выходом Y для каждой пары.

При этом возможны четыре варианта:

- выход не зависит от переменной и равен нулю;
- выход не зависит от переменной b_3 и равен единице;
- выход Y зависит от переменной b_3 и равен ей;
- выход Y зависит от переменной b_3 и совпадает с ее инверсией $\overline{b_3}$.

Исходя из вышеизложенного, на информационные входы мультиплексора подаются «лог. 0», «лог. 1», b_0 или \bar{b}_0 . Пример реализации функции четырех переменных, заданной табл. 5.3 на базе мультиплексора с тремя адресными входами, изображен на рис. 5.3.

Таблица 5.4

Таблица истинности логарифмической функции с выделенной переменной b_0

Переменные				Значение функции	
b_3	b_2	b_1	b_0	Y	
0	0	0	0	0	Совпадает с b_0
0	0	0	1	1	
0	0	1	0	0	«лог. 0»
0	0	1	1	0	
0	1	0	0	0	«лог. 0»
0	1	0	1	0	
0	1	1	0	0	«лог. 0»
0	1	1	1	0	
1	0	0	0	1	Совпадает с \bar{b}_0
1	0	0	1	0	
1	0	1	0	1	Совпадает с \bar{b}_0
1	0	1	1	0	
1	1	0	0	1	«лог. 1»
1	1	0	1	1	

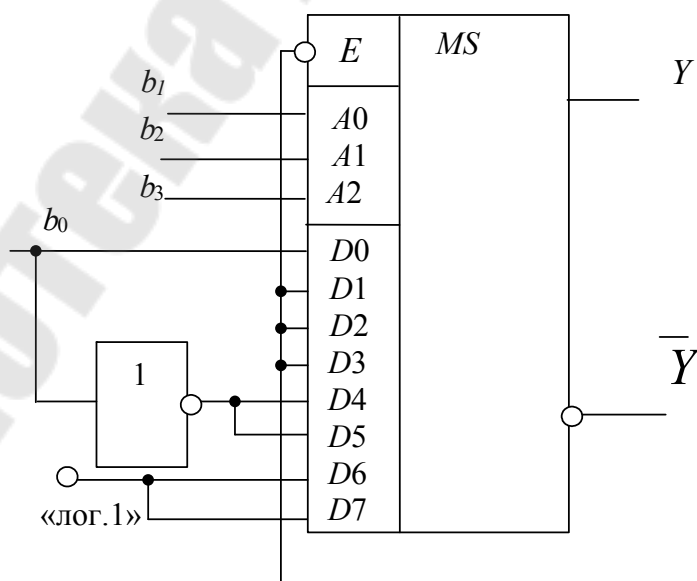


Рис. 5.3. Пример реализации на мультиплексоре 8 : 1 функции четырех переменных

Задания для практического занятия

Номер варианта задания выбрать из табл. 5.5 согласно N – номеру студента в списке журнала группы.

Таблица 5.5

Варианты заданий

Номер варианта	У
1	00110111
2	11100110
3	10111010
4	11111000
5	00011111
6	11010011
7	01111010
8	11001110
9	11100101
10	10111100
11	10010111
12	11100110
13	11011100
14	10001111
15	11110001
16	01101110
17	11001101
18	01110101
19	11110001
20	00111110
21	10100111
22	11101100
23	10011101
24	11001011
25	01111001
26	00101111
27	11001101
28	10111001
29	00011111
30	11101010

Необходимо выполнить следующие задания для практического занятия:

1. В столбец таблицы истинности (табл. 5.6) вписать заданное восьмиразрядное двоичное число Y , взятое из табл. 5.5. Получить таблицу истинности логической функции для четырех входных переменных.

Таблица 5.6

**Таблица истинности логической функции
для четырех входных переменных**

b_3	b_2	b_1	b_0	Y
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

2. Реализовать логическую функцию, заданную полученной таблицей истинности на мультиплексоре с четырьмя адресными входами (мультиплексор 16 : 1).

3. В столбец таблицы истинности (табл. 5.6) вписать восьмиразрядное двоичное число, равное инверсии заданного числа \bar{Y} . Получить таблицу для четырех переменных.

4. Реализовать логическую функцию, заданную полученной таблицей истинности на мультиплексоре с тремя адресными входами (мультиплексор 8 : 1).

Практическое занятие № 6

СИНТЕЗ ПОСЛЕДОВАТЕЛЬНОСТНЫХ ЦИФРОВЫХ УСТРОЙСТВ

Теоретические сведения

Триггеры

Триггеры и регистры являются простейшими представителями цифровых микросхем, имеющих внутреннюю память. Если выходные сигналы логических элементов и комбинационных микросхем однозначно определяются их текущими входными сигналами, то выходные сигналы микросхем с внутренней памятью зависят также еще и от того, какие входные сигналы и в какой последовательности поступали на них в прошлом, т. е. они помнят предысторию поведения схемы. Именно поэтому их применение позволяет строить гораздо более сложные и интеллектуальные цифровые устройства, чем в случае простейших микросхем без памяти. Микросхемы с внутренней памятью называются еще **последовательностными** в отличие от *комбинационных* микросхем.

Триггер можно рассматривать как одноразрядную, а регистр – как многоразрядную ячейку памяти, которая состоит из несколько триггеров, соединенных параллельно (обычный, параллельный регистр) или последовательно (сдвиговой регистр или (что то же самое) регистр сдвига).

Триггером называют логическую схему с положительной обратной связью, имеющую два устойчивых состояния, которые называют единичным и нулевым. Триггер имеет два выхода: прямой Q и инверсный \bar{Q} . Состояние триггера определяется логическим уровнем на прямом выходе. Перевод триггера в состояние $Q = 1$ называют **установкой** (*Set*) и вход, по которому это осуществляется, обозначается S . Перевод триггера в нулевое состояние $Q = 0$ называют **сбросом** (*Reset*) и вход сброса обозначают R .

Триггер является *асинхронным*, если его переключение происходит в момент поступления управляющих сигналов. Триггер является *синхронным* или *тактируемым*, если его переключение производится только при наличии разрешающего, тактирующего импульса; у таких триггеров помимо информационных входов имеется вход тактовых импульсов.

Триггеры характеризуются таблицей переключений и составленной по ней *функцией возбуждения* в виде $Q_n = f(Q_{n-1}, X_n)$, где X_n – входное воздействие, представляющее собой набор входных сигналов. Она описывает состояние триггера после воздействия и зависит как от входного воздействия, так и от его предшествующего состояния. Обычно она записывается в форме $Q_{n+1} = f(Q_n, X_n)$, где X_n представляет собой двухразрядное слово, представленное сигналами S и R . Часто состояние Q_{n+1} обозначается Q^+ .

При комбинации $S = 0, R = 0$ триггер сохраняет свое предыдущее состояние выхода Q_n , т. е. $Q_{n+1} = Q_n$, если же $S = 1, R = 0$ переходит в единичное, а когда $S = 0, R = 1$ – в нулевое. Последнее означает, что $Q_{n+1} = 0$ независимо от того, в каком состоянии находился триггер до воздействия. Данный алгоритм характерен для большинства разновидностей триггеров, которые отличаются друг от друга лишь по реакции на комбинацию сигналов $S = 1, R = 1$ (рис. 6.2). Возможны следующие варианты: $Q_{n+1} = 0, Q_{n+1} = Q_n, Q_{n+1} = \bar{Q}_n$ и, наконец, эту комбинацию можно считать запрещенной, т. е. не подавать ее на входы управления. Формально при ее наличии состояние триггера будет неопределенным: $Q_{n+1} = *$, так как не подавая данную комбинацию, нельзя ничего сказать и о выходном состоянии $Q_{n+1}^{\bar{R}S}$.

Входы		Выходы			
S	R	Q_{n+1}^{RS}	$Q_{n+1}^{\bar{R}\bar{S}}$	Q_{n+1}^E	Q_{n+1}^{JK}
0	0	Q_n	Q_n	Q_n	Q_n
0	1	0	0	0	0
1	0	1	1	1	1
1	1	*	*	Q_n	\bar{Q}_n

Рис. 6.1. Таблица функционирования различных триггеров

Первый вариант триггера (рис. 6.1) называется RS -триггером, который является простейшим в этом классе; второй – RS -триггером с инверсными входами, третий – E -триггером, а последний – JK -триггером. Функции возбуждения этих триггеров должны описываться соотношениями вида $Q^+ = f(R, S, Q)$. То, что состояние таких устройств зависит от комбинации входных сигналов R и S , следует

из заданного алгоритма работы. Однако, так как сохранение комбинации управляющих сигналов к следующему моменту времени формально может считаться новым воздействием, то при переходе $R = 0, S = 0 \rightarrow R = 0, S = 0$ состояние Q^+ останется равным Q , следовательно, в общем случае можно записать, что $Q^+ = f(Q)$. Таким образом, предложенный алгоритм описывает работу некоторого последовательностного устройства.

Исходя из этого, для синтеза RS -триггера в состав переменных необходимо ввести значения Q .

Таблица переключений и условное графическое обозначение RS -триггера представлены на рис. 6.2.

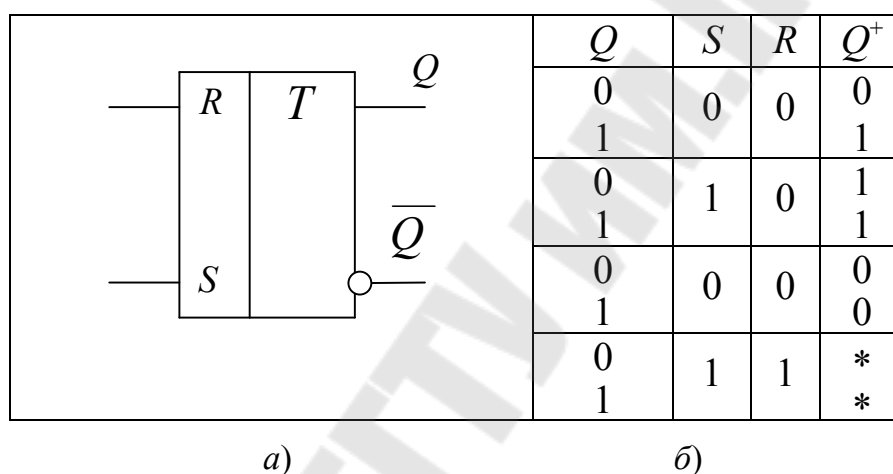


Рис. 6.2. Асинхронный RS -триггер:
 а – его условно-графическое обозначение;
 б – его таблица переключений

По данной таблице функционирования составим функцию возбуждения RS -триггера, воспользовавшись картой Карно для трех переменных для недоопределенных функций (рис. 6.3).

Такая карта строится стандартным образом, в ее клетки записываются нули, единицы и звездочки. Как и для обычной функции, единицы и звездочки можно охватить контурами склейки. Однако из-за того, что функция не определена на наборах, которые не могут реализоваться, ей в этом случае можно приписать любое значение, т. е. либо ноль, либо единицу. В принципе это ничего не меняет, так как данные наборы и приписанные им значения функции не реализуются. Отсюда следует, что для не полностью определенных функций звездочки в клетках карты Карно можно в зависимости от ситуации заменять нулями, или единицами.

После упрощения получаем выражение для схемной реализации RS -триггера в виде: $Q_{RS}^+ = S + Q\bar{R}$, для инверсного выхода: $\bar{Q}_{RS}^+ = R + \bar{Q}S$. Таким образом, для синтеза RS -триггера требуется два двухвходовых логических элемента 2ИЛИ-НЕ (рис. 6.3, б).

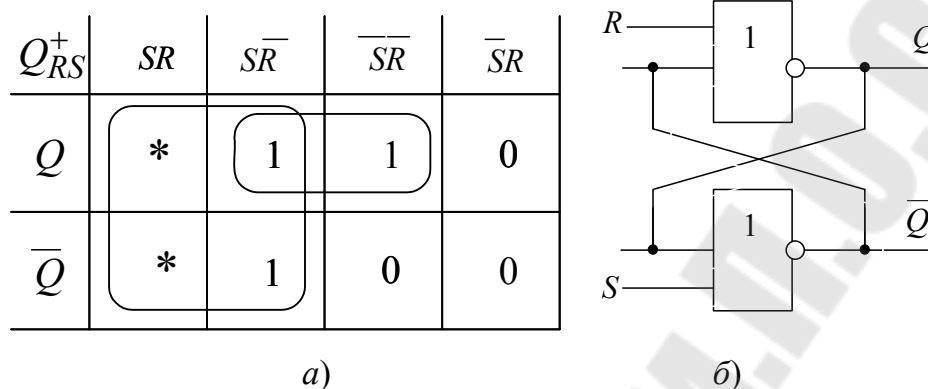


Рис. 6.3. Асинхронный RS -триггер:

а – карта Карно; б – реализация на элементах 2ИЛИ-НЕ

Особое место среди рассмотренных ранее вариантов занимает JK -триггер, который при единичных значениях управляющих сигналов меняет свое состояние на противоположное. Вход J выполняет функции входа S , а K – входа R при комбинациях, разрешенных для RS -триггера. Упрощенная и полная таблицы его функционирования приведены на рис. 6.4.

J	K	Q_{JK}^+
0	0	Q
0	1	0
1	0	1
1	1	\bar{Q}

J	K	Q	Q_{JK}^+
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Рис. 6.4. Таблицы функционирования JK -триггера:
а – упрощенная; б – полная

Полученная после минимизации по карте Карно (рис. 6.5) функция возбуждения имеет вид: $Q_{JK}^+ = Q\bar{K} + \bar{Q}J$.

Q_{JK}^+	JK	$J\bar{K}$	$\bar{J}K$	$\bar{J}\bar{K}$
Q	0	1	1	0
\bar{Q}	1	1	0	0

Рис. 6.5. Карта Карно JK-триггера

Если входы J и K рассмотренного триггера объединить, то получится счетный T -триггер, имеющий один вход и меняющий свое состояние каждый раз с приходом спада управляющего сигнала. Его функция возбуждения может быть получена из соответствующей функции JK -триггера и имеет вид: $Q_T^+ = Q\bar{T} + \bar{Q}T$. Так как его состояние меняется на противоположное после каждого воздействия, то формально функция возбуждения T -триггера может быть записана в виде: $Q_T^+ = \bar{Q}$.

Все вышерассмотренные триггеры относятся к классу *асинхронных*, т. е. они реагируют на изменения управляющих сигналов непосредственно в момент их поступления. В ряде ситуаций, особенно при построении сложных цифровых устройств, различие в моментах прихода таких сигналов на разные триггеры приведет к неодновременности их переключения, что может вызвать нарушение работы связанных с ними устройств, в частности, вследствие возникновения соствязаний фронтов.

Эта проблема может быть решена при использовании *синхронных* триггеров, реагирующих на входные воздействия лишь после прихода сигнала синхронизации. Соответствующий вход триггера обозначается буквой S . Синхронный триггер можно представить как совокупность асинхронного и некоторого устройства синхронизации, подключаемого к его входам (рис. 6.6, б).

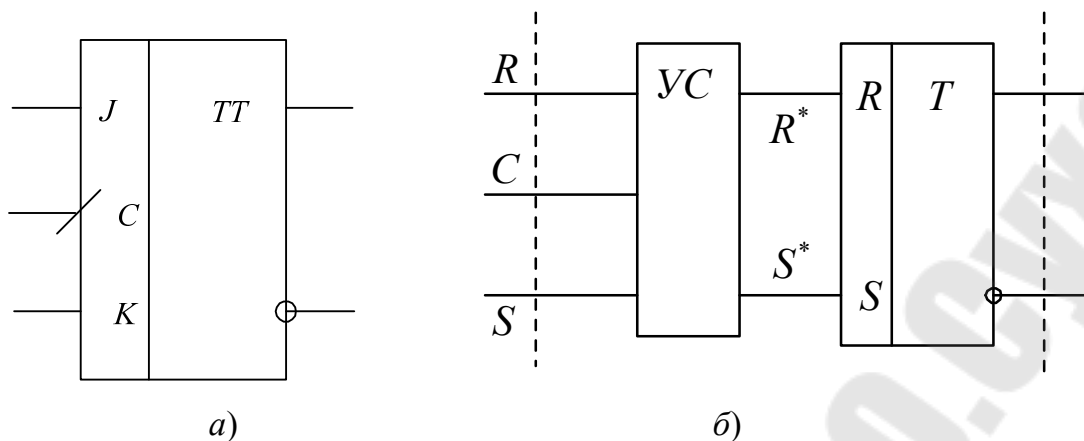


Рис. 6.6. Синхронный триггер:
a – условное графическое обозначение синхронного JK-триггера, срабатывающего по фронту синхроимпульса;
б – функциональная схема синхронного триггера

Синхронные триггеры, реагирующие на входные воздействия при поступлении определенного уровня сигнала синхронизации, называются **триггерами**, синхронизируемыми уровнем, или *триггерами-защелками*. Кроме них в классе синхронных триггеров имеется еще одна разновидность – триггеры, синхронизируемые фронтом (срабатывают по фронту синхроимпульса).

Их отличие от синхронизируемых уровнем состоит в том, что реакция на внешние сигналы управления возникает лишь при поступлении фронта (или среза) сигнала синхронизации. На принципиальных схемах такие триггеры обозначаются обычным образом, но у символа входа синхронизации вводится значок в виде наклонной черты, как показано на рис. 6.6, *a*.

Счетчики

К группе последовательностных относятся узлы, называемые **счетчиками**. Это устройства, по состоянию выходов которых можно определить количество входных воздействий (импульсов), поступивших на их входы к моменту наблюдения. Ячейкой, которая под действием внешних сигналов может менять свое состояние и сохранять его, является триггер, поэтому именно триггеры используются в качестве разрядов счетчиков.

Одним из основных параметров счетчика является *модуль счета M*, представляющий собой число устойчивых различных состояний счетчика. Когда число поступивших входных воздействий превысит модуль счета, то произойдет переполнение счетчика, он

вернется в исходное состояние и начнется новый процесс смены его состояний. Входным воздействием обычно является импульс, поступающий на специальный, так называемый счетный вход.

По значению модуля счетчики делятся на *двоичные*, *двоично-десятичные* (*десятичные*), счетчики с *произвольным* и *управляемым* модулем счета. В *двоичных* счетчиках модуль кратен степени двойки, т. е. принимает значения 2, 4, 8, 16 и т. п., в *десятичных* он кратен степени десяти. В счетчиках с *произвольным* модулем он может иметь любое фиксированное значение, а в счетчиках с *управляемым* модулем имеется возможность менять модуль счета под воздействием внешних сигналов.

Счетчики могут быть *суммирующими*, *вычитающими* и *реверсивными*. В *суммирующих* счетчиках число, соответствующее формируемому счетчиком коду, с приходом очередного счетного импульса на единицу увеличивается, в *вычитающих* – уменьшается. *Реверсивные* счетчики в зависимости от установленного режима работы могут функционировать и как суммирующие, и как вычитающие.

Двоичные счетчики имеют модуль счета $M = 2^N$, где N – количество триггеров. Однако при проектировании цифровых устройств часто возникает необходимость использовать счетчик с произвольным модулем счета: $M \neq 2^N$. Принцип построения таких счетчиков состоит в исключении избыточных состояний либо с помощью обратных связей внутри счетчика, либо методом управляемого сброса его в ноль при достижении определенного состояния.

Таким образом, если требуется получить счетчик с произвольным модулем счета, то число необходимых при этом триггеров определяется исходя из условия $M \leq 2^N$.

Так, при $N = 4$ может быть получен любой модуль счета в пределах от 9 до 16. Например, при построении счетчика с модулем счета $M = 10$, шесть его состояний не используется. При этом, если счет осуществляется в натуральном *BCD*-коде, то счетчик последовательно проходит состояния от 0000 до 1001 (код десятичного числа – 9), подобно двоичному счетчику, только после этого следующим состоянием счетчика опять будет 0000 (код десятичного числа – 0).

Синтез десятичного асинхронного счетчика, или любого другого асинхронного счетчика с $M \neq 2^N$, или счетчика, работающего не в прямом двоичном коде, является сложной проблемой, поскольку не существует прямой процедуры синтеза.

На выходах суммирующего двоичного счетчика с модулем $M = 2^N$ формируются N -разрядные кодовые комбинации, порядок смены которых соответствует изменению двоичного кода, описывающего состояния счетчика от 0 до $M - 1$.

Работа счетчика может быть описана тремя способами:

- в виде таблицы состояний (рис. 6.7, а);
- посредством *графа переходов* (рис. 6.7, б);
- с помощью временных диаграмм, отражающих состояния его разрядов после поступления очередного импульса на счетный вход в виде таблицы состояний (рис. 6.7, в).

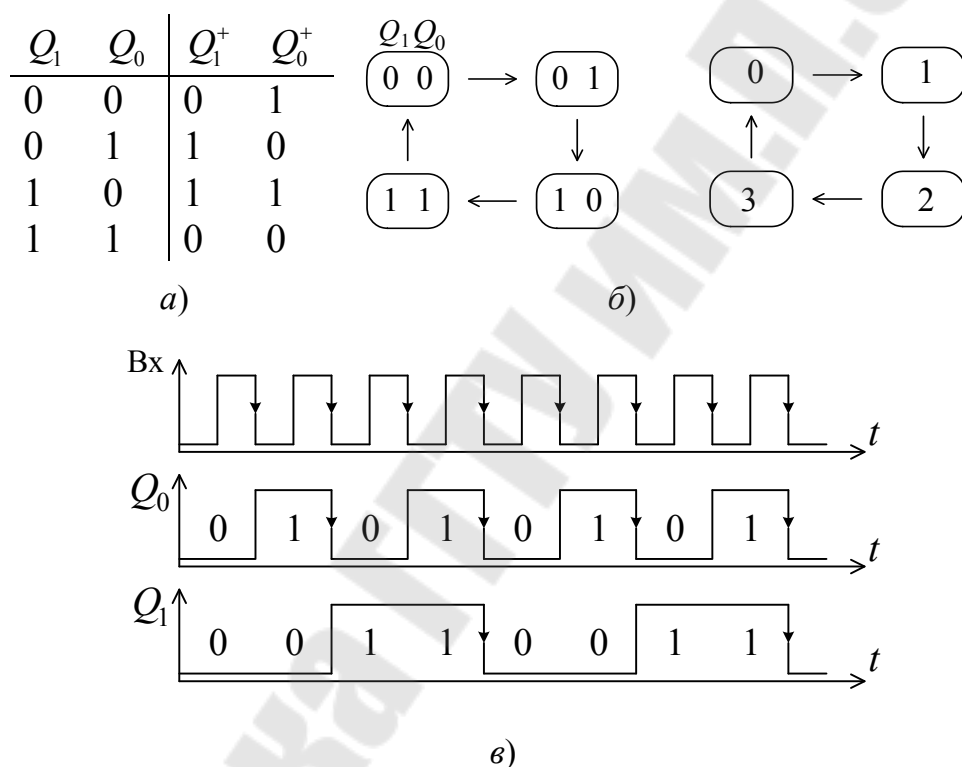


Рис. 6.7. Способы описания работы счетчика с модулем счета 4:
 а – таблица состояний; б – графы переходов;
 в – временные диаграммы

Граф переходов представляет собой символическую запись процедуры смены состояний счетчика и может описывать порядок их смены как в виде комбинаций значений разрядов, так и в виде десятичных чисел. В этом случае состояние всех разрядов счетчика в текущий момент времени интерпретируется как двоичная кодовая комбинация, которая переводится в десятичную систему счисления. При этом i -му разряду приписывается соответствующий вес, равный 2^i , где i – номер разряда. Нумерация разрядов начинается с нуля.

Таблица переходов отображает процедуру смены состояний разрядов счетчика. В левой части строки записывается состояние счетчика до поступления входного воздействия, а в правой – после него.

К примеру, для построения счетчика с модулем счета $M = 4$ потребуется система из двух триггеров, для которых возможны четыре состояния. У суммирующего счетчика число, описывающее его состояние и равное количеству поступивших импульсов, должно увеличиваться с приходом каждого из них на единицу. Младший разряд счетчика Q_0 должен иметь вес 2^0 , а старший $Q_1 - 2^1$. Способы описания его работы приведены на рис. 6.7.

Синтез счетчиков

В счетчиках можно использовать как асинхронные, так и синхронизируемые фронтом JK -триггеры. Синтез синхронных счетчиков на синхронных триггерах удобно начинать, определив граф переходов, описывающий его функционирование. Так, для суммирующего счетчика с модулем 8 он имеет вид, представленный на рис 6.8, а).

На основании этого графа составляется таблица переходов, в левой части которой указываются состояния разрядов счетчика до, а в правой – после переключения (рис. 6.8, б).

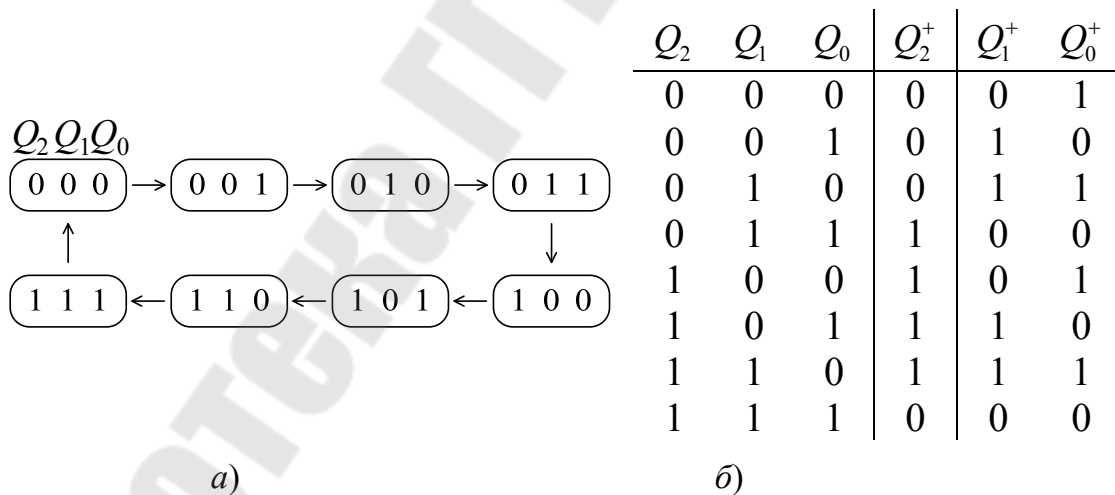


Рис. 6.8. Описание работы суммирующего счетчика с модулем счета 8: а – при помощи графа; б – при помощи таблицы переходов

Состояния разрядов Q_i^+ можно считать некоторыми функциями предшествующих, т. е. $Q_i^+ = F_i(Q_0, Q_1 \dots Q_{n-1})$. Конкретный вид соответствующей функции зависит от задаваемого алгоритма работы счетчика.

Используя для минимизации карты Карно, функции можно привести к виду, показанному на рис. 6.9.

Далее, задавшись типом конкретного триггера, требуется в соответствии с его функцией возбуждения подобрать такие наборы управляющих сигналов, чтобы переключение триггера происходило по законам, определяемым полученными функциями. В этом случае, триггер, функционируя в соответствии со своим алгоритмом работы, будет выполнять функции разряда счетчика.

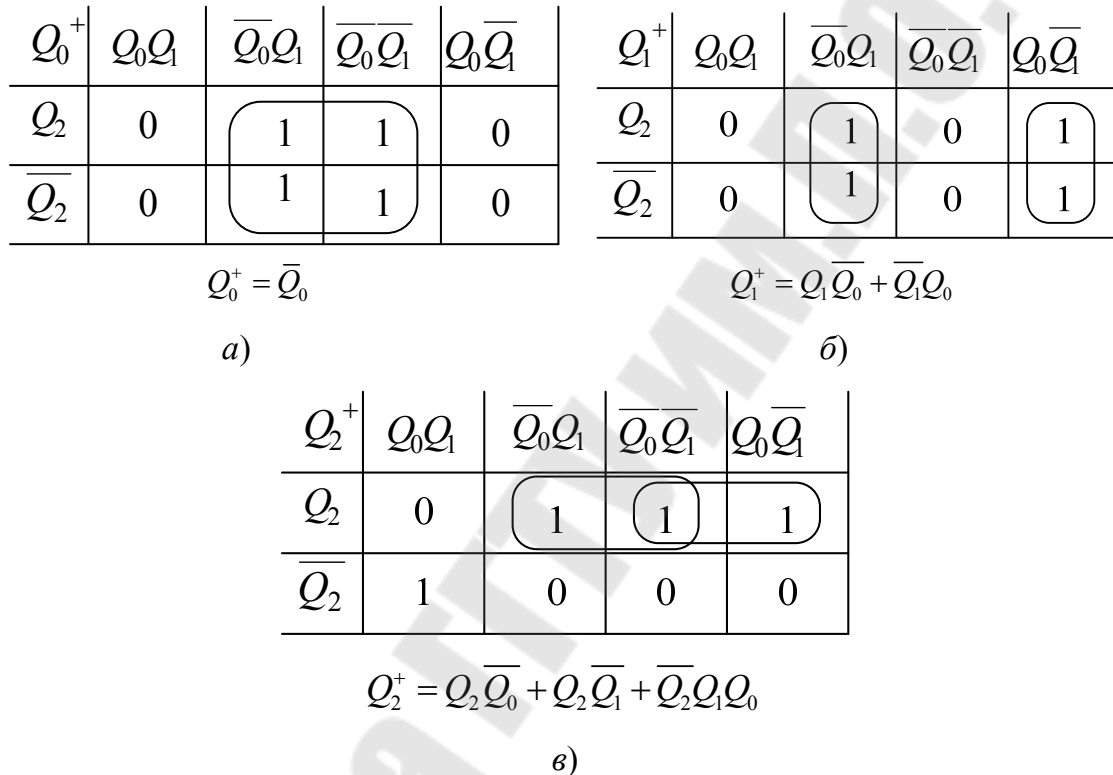


Рис. 6.9. Карты Карно и упрощенные выражения для суммирующего счетчика с модулем счета 8:

а – для нулевого разряда счетчика; б – для первого разряда счетчика; в – для второго разряда счетчика

Функция возбуждения JK-триггера выглядит следующим образом: $Q_{JK}^+ = Q\bar{K} + \bar{Q}J$. Для нулевого разряда счетчика она будет совпадать с найденной, если $K_0 = J_0 = 1$. Аналогичный триггер будет функционировать как первый разряд при $K_1 = J_1 = Q_0$. Для подбора управляющих сигналов триггера, выполняющего функцию второго разряда, полученную функцию необходимо преобразовать к подобному виду $Q_2^+ = Q_2\bar{K} + \bar{Q}_2J$:

$$Q_2^+ = Q_2 \bar{Q}_0 + Q_2 \bar{Q}_1 + \bar{Q}_2 Q_1 Q_0 = Q_2 (\bar{Q}_0 + \bar{Q}_1) + \bar{Q}_2 Q_1 Q_0 = Q_2 \bar{Q}_1 \bar{Q}_0 + \bar{Q}_2 Q_1 Q_0.$$

Отсюда следует, что $K_2 = J_2 = Q_1 Q_0$.

Получаем выражения для построения выходных разрядов счетчика на JK -триггерах:

– для нулевого разряда – $Q_0^+ = Q_0 \bar{K}_0 + \bar{Q}_0 J_0$,

т. е. $Q_0^+ = \bar{Q}_0 = Q_0 \cdot 0 + \bar{Q}_0 \cdot 1$;

– для первого разряда – $Q_1^+ = Q_1 \bar{K}_1 + \bar{Q}_1 J_1$, т. е. $Q_1^+ = Q_1 \bar{Q}_0 + \bar{Q}_1 Q_0$;

– для второго разряда – $Q_2^+ = Q_2 \bar{K}_2 + \bar{Q}_2 J_2$,

т. е. $Q_2^+ = Q_2 \bar{Q}_0 \bar{Q}_1 + \bar{Q}_2 Q_0 Q_1$.

Если данные наборы сигналов подать на соответствующие входы триггеров, то они будут выполнять функции разрядов суммирующего синхронного счетчика с модулем 8, принципиальная схема которого показана на рис. 6.10.

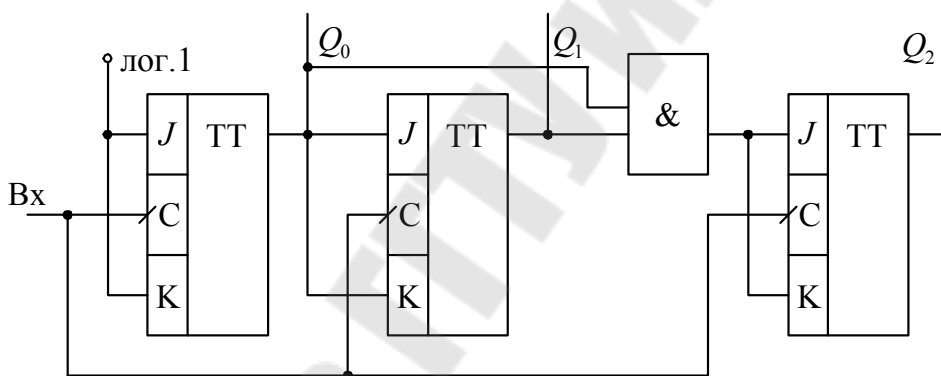


Рис. 6.10. Принципиальная схема суммирующего синхронного счетчика с модулем счета 8

Узлы для формирования управляющих сигналов при использовании JK -триггеров получаются проще, чем для триггеров других типов, что обуславливает их широкое применение для построения счетчиковых структур.

Пусть имеется двоичный суммирующий счетчик с модулем M . Последовательность смены его состояний представлена на рис. 6.11. Из состояния, соответствующего коду числа $(M - 1)$, счетчик будет переходить в исходное, нулевое.

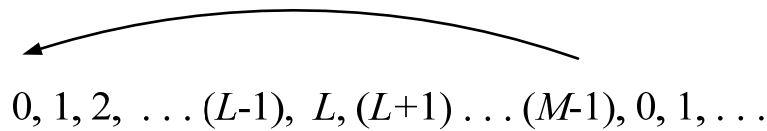


Рис. 6.11. Последовательность смены состояний двоичного суммирующего счетчика с модулем счета M

В счетчиках с недвоичным модулем количество рабочих состояний L отличается от 2^n . Для построения таких устройств можно использовать двоичные счетчики с $M > L$, у которых часть состояний, а именно $(M - L)$ исключается из числа рабочих.

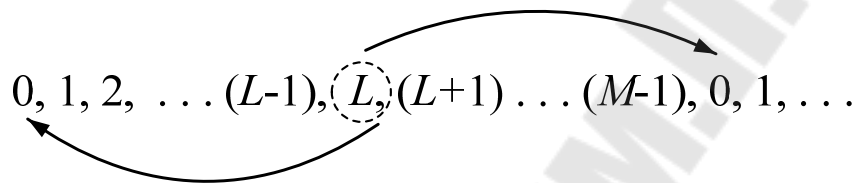


Рис. 6.12. Последовательность смены состояний недвоичного суммирующего счетчика с модулем счета L

Для реализации счетчика с модулем L необходимо из рабочего цикла исходного двоичного исключить $(M - L)$ состояний (рис. 6.12). Эти состояния могут исключаться произвольным образом, т. е. $(M - L)$ первых, последних, либо выбранных состояний в соответствии с заданными требованиями. Такие состояния будут нерабочими.

Количество счетчиков с модулем L , которые можно реализовать на базе двоичного с модулем M , определяется соотношением

$$N = \frac{M!}{(M - L)!},$$

где восклицательный знак обозначает факториал, т. е.

результат произведения чисел от 1 до аргумента данной функции.

При использовании счетчика с модулем $M = 4$ в качестве исходного, число вариантов счетчиков с модулем 3 будет равно

$$N = \frac{4!}{(4 - 3)!} = \frac{1 \cdot 2 \cdot 3 \cdot 4}{1} = 24.$$

Если граф переходов двухразрядного двоичного счетчика представить, как показано на рис. 6.13, а, то условные графы переходов, описывающих порядок смены состояний 24 вариантов счетчиков с модулем 3, будут иметь вид, представленный на том же рис. 6.13, в.

Записи, объединенные в группы, описывают работу одного и того же счетчика с различными начальными состояниями. Количество

счетчиков с несовпадающими графами переходов будет равно 8, общая формула в этом случае выглядит следующим образом:

$$N = \frac{M!}{(M-L)! \cdot L}, \text{ а графы их переходов приведены на рис. 6.13, б.}$$

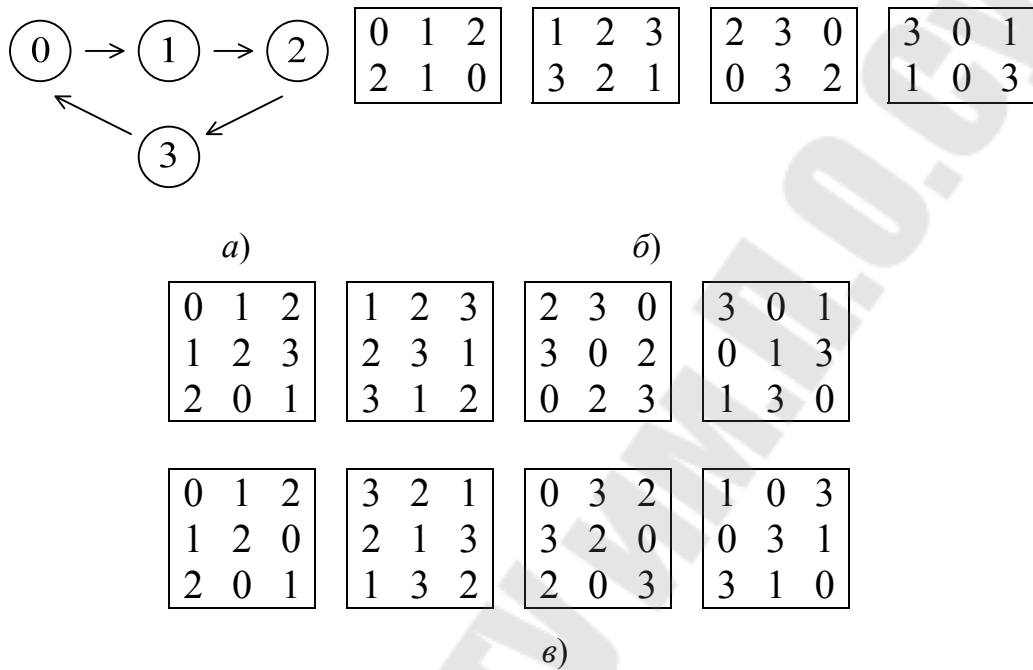


Рис. 6.13. Описание работы суммирующего счетчика с модулем счета 3:

- а – граф переходов двухразрядного двоичного счетчика;
- б – несовпадающие графы переходов счетчиков с модулем счета 3;
- в – условные графы переходов, описывающих порядок смены состояний 24 вариантов счетчиков с модулем счета 3

Для синтеза таких счетчиков можно использовать методику, описанную при построении двоичных счетчиков на синхронных триггерах.

Рассмотрим вариант построения счетчика с модулем счета 3, граф переходов которого и таблица переключений изображены на рис. 6.14.

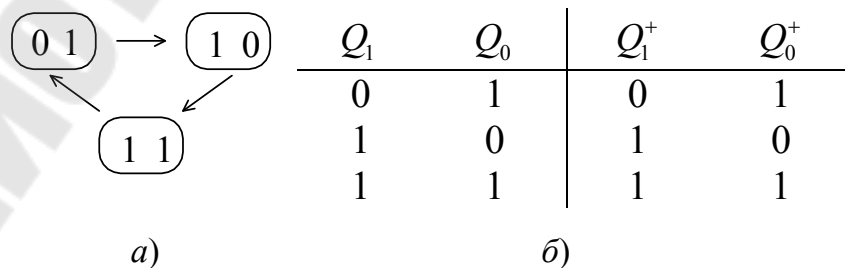


Рис. 6.14. Описание работы счетчика с модулем счета 3: а – граф переходов; б – таблица переключений

Функции, описывающие состояния его разрядов, могут быть представлены как: $Q_0^+ = Q_0Q_1 + \overline{Q_0}Q_1$, $Q_1^+ = Q_1\overline{Q_0} + \overline{Q_1}Q_0$, откуда и $K_1 = Q_0$, $J_1 = Q_0$. Схема такого счетчика, построенного на JK -триггерах, приведена на рис. 6.15.

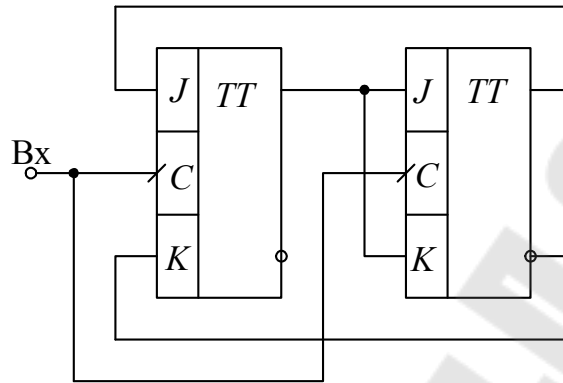


Рис. 6.15. Схема счетчика с модулем счета 3, построенного по таблице переключений и графу, представленным на рис. 6.14

Для рассматриваемого счетчика $Q_0^+ = Q_1^+ = 0$, т. е. счетчик в этом состоянии заикнется, и его полный граф переходов будет иметь вид как на рис. 6.16.

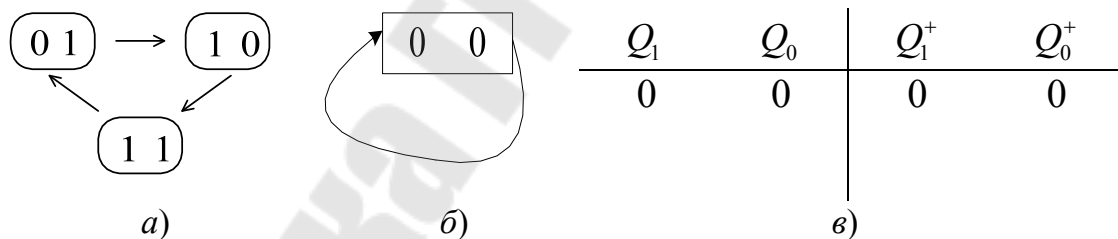


Рис. 6.16. Описание работы счетчика с модулем счета 3:
a – граф переходов; *б* – граф заикнутой комбинации;
в – таблица переключений в случае сбоя

Нерабочим состоянием здесь является комбинация 0-0. Несмотря на то, что этой комбинации на выходах счетчика не должно быть, после включения питания или под воздействием помех разряды счетчика могут перейти в это состояние. Такая ситуация называется **сбоем**. Некоторые варианты счетчиков через несколько тактов могут самостоятельно выйти из сбоя, т. е. восстановить рабочий цикл. Однако возможна ситуация, когда сбой будет сменяться сбоем и работоспособность счетчика не восстановится. Для проверки необ-

ходимо провести анализ развития событий после возникновения сбоя. С этой целью значения разрядов счетчика, соответствующие сбою, подставляются в формулы, описывающие их состояния после переключения.

В некоторых случаях использование стандартных процедур минимизации функций, описывающих состояния разрядов счетчика после переключения, приводит к выражениям, которые не позволяют подобрать требуемые комбинации управляющих сигналов на входах *JK*-триггеров. Это может потребовать отказа от полной минимизации с целью получения необходимой структуры функций.

Задания для практического занятия

Номер варианта задания выбрать из табл. 6.1 согласно N – номеру студента в списке журнала группы.

Необходимо выполнить следующие задания для практического занятия:

1. В соответствии с вариантом задания разработать принципиальную схему счетчика с заданным графом переходов и проанализировать ситуации при возникновении сбоев в его работе.

Для этого последовательно сделать:

- по заданному графу составить таблицу переходов;
- определить функции, описывающие состояния разрядов счетчика после переключения;
- подобрать совокупности управляющих сигналов на входах триггеров для его реализации;
- осуществить анализ работы счетчика при возникновении сбоев;
- синтезировать принципиальную схему разработанного устройства;
- произвести построение временных диаграмм работы счетчика.

Таблица 6.1

Варианты заданий

Номер варианта	Граф переходов	Номер варианта	Граф переходов
1		16	
2		17	
3		18	
4		19	

Продолжение табл. 6.1

Номер варианта	Граф переходов	Номер варианта	Граф переходов
5		20	
6		21	
7		22	
8		23	
9		24	
10		25	
11		26	
12		27	
13		28	

Номер варианта	Граф переходов	Номер варианта	Граф переходов
14		29	
15		30	

Рассмотрим пример синтеза трехразрядного счетчика с модулем 6 на синхронных *JK*-триггерах с графом переходов, приведенным на рис. 6.17.

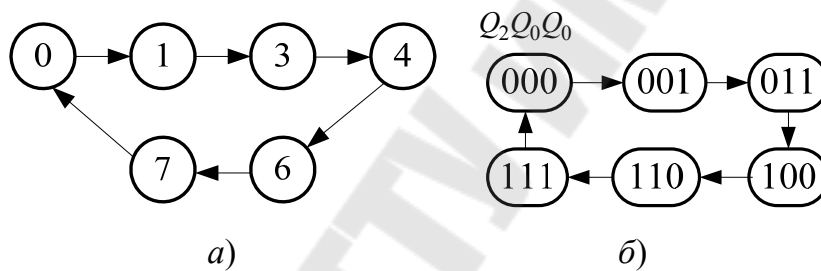


Рис. 6.17. Граф переходов трехразрядного счетчика с модулем счета 6:

а – представленный десятичными числами;

б – представленный двоичными числами

В соответствии с графом получаем следующую таблицу переходов, показанную на рис. 6.18.

Q_2	Q_1	Q_0	Q_2^+	Q_1^+	Q_0^+
0	0	0	0	0	1
0	0	1	0	1	1
0	1	1	1	0	0
1	0	0	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

Рис. 6.18. Таблица переходов трехразрядного счетчика с модулем счета 6

Карты Карно для функций, описывающих состояния разрядов Q_0^+ , Q_1^+ , Q_2^+ счетчика после переключения, имеют вид, изображенный на рис. 6.19.

Контуры склейки выбираются таким образом, чтобы в выражения для соответствующих функций Q_i^+ входили прямые и инверсные значения функций Q_i .

Q_0^+	Q_1Q_0	$Q_1\overline{Q_0}$	$\overline{Q_1}Q_0$	$\overline{Q_1}\overline{Q_0}$	Q_1^+	Q_1Q_0	$Q_1\overline{Q_0}$	$\overline{Q_1}Q_0$	$\overline{Q_1}\overline{Q_0}$	Q_2^+	Q_1Q_0	$Q_1\overline{Q_0}$	$\overline{Q_1}Q_0$	$\overline{Q_1}\overline{Q_0}$
Q_2	0	1	0	*	Q_2	0	1	1	*	Q_2	0	1	1	*
$\overline{Q_2}$	0	*	1	1	$\overline{Q_2}$	0	*	0	1	$\overline{Q_2}$	1	*	0	0

Рис. 6.19. Карты Карно для функций, описывающих состояния разрядов счетчика после переключения

Это необходимо для подбора управляющих сигналов на J -, K -входах соответствующих триггеров, функции возбуждения которых имеют вид: $Q_i^+ = Q_i\overline{K_i} + \overline{Q_i}J_i$.

Функция Q_0^+ , описывающая состояние младшего (нулевого) разряда счетчика, имеет вид: $Q_0^+ = Q_0\overline{Q_1} + \overline{Q_0}Q_1 + \overline{Q_0}Q_2$. Для подбора управляющих сигналов она должна быть преобразована к двухкомпонентной форме: $Q_0^+ = Q_0\overline{K_0} + \overline{Q_0}J_0$.

После преобразования получаем:

$$Q_0^+ = Q_0\overline{Q_1} + \overline{Q_0}Q_1 + \overline{Q_0}Q_2 = Q_0\overline{Q_1} + \overline{Q_0}(Q_1 + Q_2) = Q_0\overline{Q_1} + \overline{Q_0}\overline{Q_1}Q_2.$$

Отсюда следует, что на входы управления триггера младшего разряда счетчика должны подаваться сигналы:

$$K_0 = Q_1; \quad J_0 = \overline{Q_1}Q_2.$$

Функция Q_1^+ , описывающая состояние первого разряда счетчика, имеет вид: $Q_1^+ = Q_1\overline{Q_0} + \overline{Q_1}Q_2 + \overline{Q_1}Q_0$. Для подбора управляющих сигналов она должна быть преобразована к двухкомпонентной форме: $Q_1^+ = Q_1\overline{K_1} + \overline{Q_1}J_1$. После преобразования получаем:

$$Q_1^+ = Q_1\overline{Q_0} + \overline{Q_1}Q_2 + \overline{Q_1}Q_0 = Q_1\overline{Q_0} + \overline{Q_1}(Q_2 + Q_0) = Q_1\overline{Q_0} + \overline{Q_1}\overline{Q_2}\overline{Q_0}.$$

Отсюда следует, что на входы управления триггера разряда Q_1 счетчика должны подаваться сигналы:

$$K_1 = Q_0; \quad J_1 = \overline{\overline{Q_2 Q_0}}.$$

Функция Q_2^+ , описывающая состояние старшего (второго) разряда счетчика, имеет вид: $Q_2^+ = Q_2 \overline{Q_0} + \overline{Q_2} Q_1$. Она уже представлена в форме: $Q_2^+ = Q_2 \overline{K_2} + \overline{Q_2} J_2$. На входы управления триггера старшего разряда счетчика должны подаваться сигналы:

$$K_2 = Q_0; \quad J_2 = Q_1.$$

Состояниями сбоя для проектируемого счетчика являются комбинации сигналов 010 и 101 на его выходах Q_2, Q_1, Q_0 . Для анализа ситуации после сбоя подставим значения данных сигналов в формулы, описывающие состояния разрядов счетчика после переключения.

Первый сбой: $Q_2 = 0, Q_1 = 1, Q_0 = 0$;

$$Q_0^+ = Q_0 \overline{Q_1} + \overline{Q_0} Q_1 + \overline{Q_0} \overline{Q_2} = 0 \cdot 0 + 1 \cdot 1 + 1 \cdot 1 = 1;$$

$$Q_1^+ = Q_1 \overline{Q_0} + \overline{Q_1} Q_2 + \overline{Q_1} Q_0 = 1 \cdot 1 + 0 \cdot 0 + 0 \cdot 0 = 1;$$

$$Q_2^+ = Q_2 \overline{Q_0} + \overline{Q_2} Q_1 = 0 \cdot 1 + 1 \cdot 1 = 1.$$

Второй сбой: $Q_2 = 1, Q_1 = 0, Q_0 = 1$;

$$Q_0^+ = Q_0 \overline{Q_1} + \overline{Q_0} Q_1 + \overline{Q_0} \overline{Q_2} = 1 \cdot 1 + 0 \cdot 0 + 0 \cdot 0 = 1;$$

$$Q_1^+ = Q_1 \overline{Q_0} + \overline{Q_1} Q_2 + \overline{Q_1} Q_0 = 0 \cdot 1 + 1 \cdot 1 + 1 \cdot 1 = 1;$$

$$Q_2^+ = Q_2 \overline{Q_0} + \overline{Q_2} Q_1 = 1 \cdot 0 + 0 \cdot 0 = 0.$$

Таблица переходов в случае сбоев имеет вид, представленный на рис. 6.20.

Q_2	Q_1	Q_0	Q_2^+	Q_1^+	Q_0^+
0	1	0	1	1	1
1	0	1	0	1	1

Рис. 6.20. Таблица переходов трехразрядного счетчика с модулем счета 6 в случае сбоев

Из рис. 6.20 следует, что в случае возникновения сбоя счетчик в следующем такте перейдет в рабочее состояние, т. е. он обладает свойствами самовосстановления.

Принципиальная схема разработанного варианта счетчика с модулем счета 6 изображена на рис. 6.21.

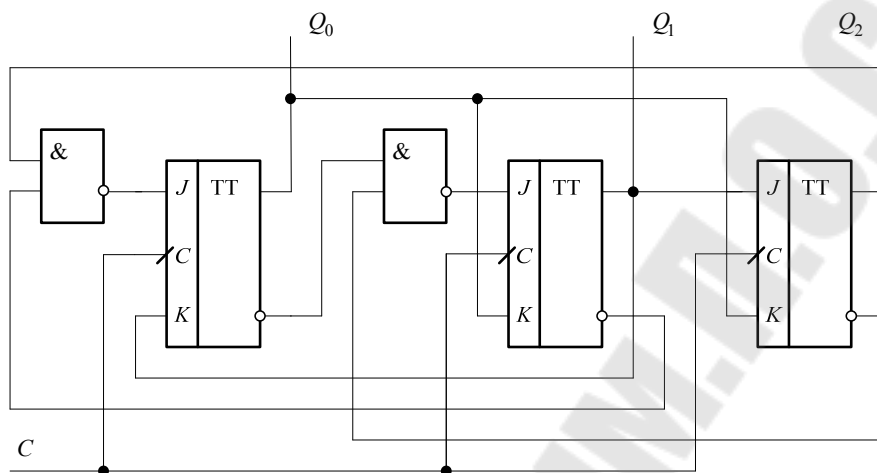


Рис. 6.21. Принципиальная схема разработанного счетчика с модулем счета 6

Временные диаграммы функционирования счетчика 6 изображены на рис. 6.22.

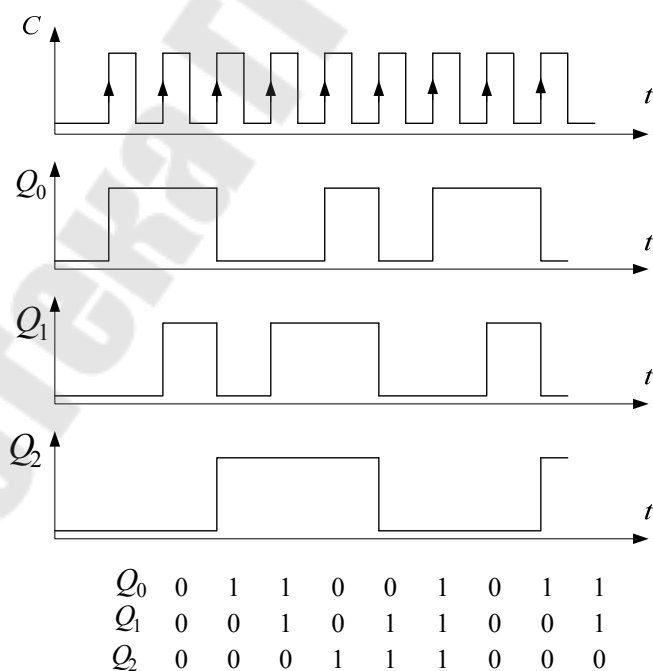


Рис. 6.22. Временные диаграммы функционирования разработанного счетчика с модулем счета 6

Литература

1. Храбров, Е. А. Цифровая электроника : учеб. пособие для вузов / Е. А. Храбров, Ю. Е. Котова. – Гомель : ГГТУ им. П. О. Сухого, 2013. – 271 с.
2. Белоус, А. И. Основы схемотехники микроэлектронных устройств / А. И. Белоус, В. А. Емельянов, А. С. Турцевич. – М. : Техносфера, 2012. – 472 с.
3. Безуглов, Д. А. Цифровые устройства и микропроцессоры : учеб. пособие для вузов / Д. А. Безуглов, И. В. Калиенко. – Ростов н/Д : Феникс, 2006. – 400 с.
4. Пухальский, Г. И. Проектирование дискретных устройств на интегральных микросхемах : справочник / Г. И. Пухальский, Т. Я. Новосельцева. – М. : Радио и связь, 1990. – 304 с.
5. Новиков, Ю. В. Введение в цифровую схемотехнику : учеб. пособие / Ю. В. Новиков. – М. : ИНТУИТ : БИНОМ, 2007. – 343 с.

Содержание

Предисловие.....	3
<i>Практическое занятие № 1. Перевод чисел из одной системы счисления в другие. Арифметические действия с многоразрядными двоичными числами</i>	4
<i>Практическое занятие № 2. Составление булевых выражений по таблице истинности и их минимизация</i>	17
<i>Практическое занятие № 3. Синтез комбинационных схем по булевым выражениям</i>	23
<i>Практическое занятие № 4. Синтез преобразователей кодов</i>	33
<i>Практическое занятие № 5. Мультиплексор как универсальный логический элемент</i>	42
<i>Практическое занятие № 6. Синтез последовательностных цифровых устройств</i>	49
Литература	70

Учебное электронное издание комбинированного распространения

Учебное издание

Котова Юлия Евгеньевна

СХЕМОТЕХНИКА ЦИФРОВЫХ УСТРОЙСТВ

Практикум

для студентов специальности 1-36 04 02

«Промышленная электроника»

дневной и заочной форм обучения

Электронный аналог печатного издания

Редактор

Т. Н. Мисюрова

Компьютерная верстка

Н. Б. Козловская

Подписано в печать 21.03.24.

Формат 60x84/16. Бумага офсетная. Гарнитура «Таймс».

Цифровая печать. Усл. печ. л. 4,18. Уч.-изд. л. 4,50.

Изд. № 17.

<http://www.gstu.by>

Издатель и полиграфическое исполнение

Гомельский государственный

технический университет имени П. О. Сухого.

Свидетельство о гос. регистрации в качестве издателя

печатных изданий за № 1/273 от 04.04.2014 г.

пр. Октября, 48, 246746, г. Гомель