

Печатается по решению
редакционно-издательского совета
физического факультета
Казанского государственного университета.

Насыров И. А.

Рецензент

Кандидат физ.-мат. наук, доцент Акчурина А. Д.

**КОНСПЕКТЫ ЛЕКЦИЙ
ПО ЦИФРОВОЙ ЭЛЕКТРОНИКЕ**

Учебное пособие

Насыров И. А.

Конспекты лекций по цифровой электронике. Учебное пособие. – Казань: КГУ, 2006. – 98 стр.

Изложены принципы построения и функционирования цифровых устройств. Приведены способы математического описания их работы, а также основы анализа и синтеза цифровых устройств с заданными техническими характеристиками. Рассмотрены принципы построения и функционирования базовых логических элементов, получивших наибольшее распространение при изготовлении современных цифровых интегральных схем.

Предназначено для студентов, обучающихся по направлению «010800 – Радиофизика и электроника» и специальности «010801 - Радиофизика и электроника».

КАЗАНЬ – 2006 г.

© Физический факультет, 2006

© Казанский государственный университет, 2006

© Зеленодольский филиал Казанского государственного университета, 2006

СОДЕРЖАНИЕ

I. ОБЩИЕ ПОЛОЖЕНИЯ	4
1. ПРЕДСТАВЛЕНИЕ ЧИСЕЛ И ВЫПОЛНЕНИЕ АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ	4
1.1. Системы счисления	4
1.2. Кодирование чисел в двоичной системе счисления	7
1.3. Выполнение арифметических операций в цифровых системах	9
2. ОСНОВЫ АЛГЕБРЫ ЛОГИКИ И ВЫПОЛНЕНИЕ ЛОГИЧЕСКИХ ОПЕРАЦИЙ	16
2.1. Логические константы и переменные. Операции Булевой алгебры	16
2.2. Основные аксиомы и законы алгебры-логики	17
2.3. Способы записи функций алгебры логики	19
3. ЛОГИЧЕСКИЕ ЭЛЕМЕНТЫ И СХЕМЫ	24
3.1. Принцип двойственности	24
3.2. Синтез логических схем в заданном базисе логических элементов	27
4. МИНИМИЗАЦИЯ ФУНКЦИЙ АЛГЕБРЫ ЛОГИКИ	28
4.1. Цели и общие принципы минимизации функций алгебры логики	28
4.2. Минимизация функций алгебры-логики с использованием карт Вейча	30
4.2.1. Минимизация полностью определенной функции алгебры логики	32
4.2.2. Минимизация недоопределенной функции алгебры логики	34
4.2.3. Минимизация системы функций алгебры логики	36
4.3. Минимизация функций алгебры-логики на ЭВМ методом Квайна и Мак-Класки	38
II. ЦИФРОВЫЕ ЛОГИЧЕСКИЕ УСТРОЙСТВА	42
5. ФУНКЦИОНАЛЬНЫЕ УЗЛЫ КОМБИНАЦИОННЫХ ЛОГИЧЕСКИХ УСТРОЙСТВ	42
5.1. Мультиплексоры и демультиплексоры	42
5.2. Преобразователи кодов	44
5.3. Цифровые компараторы	47
5.4. Сумматоры	49
5.5. Логические элементы, реализующие сложные функции	53
6. ТРИГГЕРНЫЕ УСТРОЙСТВА	54
6.1. Одноступенчатые триггеры	57
6.2. Двухступенчатые триггеры	62
6.3. Триггеры с динамическим управлением	65
7. ФУНКЦИОНАЛЬНЫЕ УЗЛЫ ПОСЛЕДОВАТЕЛЬНОСТНЫХ ЛОГИЧЕСКИХ УСТРОЙСТВ	67
7.1. Регистры	67
7.2. Счетчики	70
8. БАЗОВЫЕ ЛОГИЧЕСКИЕ ЭЛЕМЕНТЫ	76
8.1. Основные требования к базовым логическим элементам	78
8.2. Базовые логические элементы транзисторно-транзисторной логики (ТТЛ)	83
8.3. Базовые логические элементы эмиттерно-связанной логики (ЭСЛ)	86
8.4. Базовые логические элементы на МДП-транзисторах	90
8.5. Базовые логические элементы интегрально-инжекционной логики (И ² Л)	94
СПИСОК ЛИТЕРАТУРЫ	97

I. ОБЩИЕ ПОЛОЖЕНИЯ

1. ПРЕДСТАВЛЕНИЕ ЧИСЕЛ И ВЫПОЛНЕНИЕ АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ

1.1. Системы счисления

Существующие системы счисления подразделяются на *позиционные* и *непозиционные*. В непозиционных системах значение конкретной цифры постоянно и не зависит от ее расположения в записи числа. Примером такой системы счисления является Римская система записи числа. Например, в числе XXXVII значение X не зависит от ее местоположения в записи числа. Оно везде равно 10.

В позиционных системах счисления значимость конкретной цифры определяется ее местоположением в записи числа. Так, произвольное число A в позиционной системе счисления с основанием q в общем случае можно представить в виде полинома:

$$A_q = a_{n-1}q^{n-1} + a_{n-2}q^{n-2} + \dots + a_0q^0 + a_{-1}q^{-1} + \dots + a_{-k}q^{-k}, \quad (1.1)$$

где a_i — разрядный коэффициент ($a_i = 0 \dots q-1$); q^i — весовой коэффициент.

Число q называется основанием системы счисления. Следует отметить, что число q может быть как целым, так и дробным.

Если в выражении (1.1) отбросить весовые коэффициенты q^i и соответствующие знаки сложения, то получим сокращенную запись числа, носящую название q -ичного кода числа A_q . Номер позиции цифры a_i называют его разрядом. Разряды с положительными степенями q образуют целую часть числа A_q , с отрицательными степенями — дробную. Цифры a_{n-1} и a_{-k} соответственно являются старшим и младшим разрядами числа.

Количество различных чисел, которое может быть записано в позиционной системе счисления с основанием q при заданном числе разрядов,

$$N = q^{n+k}. \quad (1.2)$$

Количество разрядов, необходимое для записи в позиционной системе счисления с основанием q некоторого числа A , можно определить из следующих соображений. Согласно (1.2), для записи числа A в системе с основанием q должно выполняться условие $A_q \leq q^{n+k} - 1$. Тогда

$$n + m \geq \log_q (A_q + 1). \quad (1.3)$$

В цифровой технике нашли применение только позиционные системы счисления.

Для представления числа, записанного в позиционной системе счисления с выбранным основанием q , при помощи электрических сигналов необходимо иметь некоторое электронное устройство, формирующее на выходе q различных электрических сигналов, которые достаточно легко можно отличить друг от друга. При этом необходимое число таких устройств должно равняться числу разрядов целой и дробной частей записываемого числа.

Критерием выбора q является минимизация аппаратных затрат при обеспечении достаточной помехоустойчивости. Попытки чисто математического

решения поставленной задачи показали, что оптимальной при поставленных требованиях является система счисления с основанием $e=2,71\dots$. Однако практически создать такую систему сложно и технически нецелесообразно.

Широкое распространение в цифровой технике получила позиционная система счисления с основанием $q = 2$ — двоичная система счисления. По определению в такой системе фигурируют только два цифровых знака 0 и 1.

Цифровые системы оперируют действительными, целыми и дробными числами, которые могут иметь две формы представления: с *плавающей запятой*, с *фиксированной запятой*.

При использовании плавающей запятой число состоит из двух частей: мантиссы (m), содержащей значащие цифры числа, и порядка (p), показывающего степень, в которую надо возвести основание счисления (q), чтобы полученное при этом число, умноженное на мантиссу, давало истинное значение представляемого числа:

$$A_q = mq^p.$$

Обычно число дается в *нормализованном* виде, когда его мантисса является правильной дробью, причем первая значащая цифра (единица) следует непосредственно после запятой: например, $A_2 = 0,1010 \times 2^{10} = 10$, где $m = 0,1010$; $p = 10$; $q = 2$.

При использовании фиксированной запятой число представляется в виде единого целого, причем положение запятой в используемой разрядной сетке жестко фиксировано. Обычно числа с фиксированной запятой даются в виде правильной дроби. Для этого все числа умножают на масштабный коэффициент, чтобы перевести их в правильную дробь. Цифровые системы, использующие числа с плавающей запятой, сложнее, чем использующие числа с фиксированной запятой, так как при этом требуется выполнение операций как над мантиссами, так и над порядками. Однако диапазон представляемых чисел при одинаковом числе разрядов в системах с плавающей запятой значительно больше.

Для представления знака числа используется дополнительный *знаковый разряд* z , который обычно располагается перед числовыми разрядами. Для положительных чисел значение знакового разряда $z = 0$, для отрицательных чисел $z=1$. Для чисел с плавающей запятой вводятся отдельные знаковые разряды для мантиссы и для порядка чисел.

При работе с устройствами вычислительной техники приходится сталкиваться с позиционными системами счисления с основанием 2, 8, 10 и 16. Рассмотрим ряд правил, позволяющих выполнить преобразование чисел из одной системы счисления в другую.

Переход от системы счисления с меньшим основанием к системе с большим основанием осуществляется при помощи выражения (1.1), которое справедливо как для целой, так и дробной частей числа.

Пример 1.1. Преобразовать двоичное число $A_2=1011$ в десятичное A_{10} .

Решение. Согласно выражению (1.1) для $q=2$ получим

$$A_{10} = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 11$$

Переход от системы счисления с большим основанием к системе счисления с меньшим основанием выполняется с соблюдением следующих правил:

- а) целая часть исходного числа делится на основание новой системы счисления;
- б) дробная часть исходного числа умножается на основание новой системы счисления.

Пример 1.2. Преобразовать в двоичную систему счисления десятичное число 25,12.

Решение.

а) Преобразуем целую часть:

$$25:2 = 12+1 \quad (A_0=1)$$

$$12:2 = 6+0 \quad (A_1=0)$$

$$6:2 = 3+0 \quad (A_2=0)$$

$$3:2 = 1+1 \quad (A_3=1)$$

$$1:2 = 0+1 \quad (A_4=1)$$

Запись целой части двоичного числа A_2 производится с последнего результата деления, т.е. $25_{10} = 11001_2$.

б) Преобразуем дробную часть:

$$0,12 \cdot 2 = 0+0,24 \quad (A_{-1}=0)$$

$$0,24 \cdot 2 = 0+0,48 \quad (A_{-2}=0)$$

$$0,48 \cdot 2 = 0+0,96 \quad (A_{-3}=0)$$

$$0,96 \cdot 2 = 1+0,92 \quad (A_{-4}=1)$$

$$0,92 \cdot 2 = 1+0,84 \quad (A_{-5}=1)$$

Запись дробной части двоичного числа производится с первого результата умножения, т.е. $0,12_{10} = 0,0001_2$.

Таким образом, окончательно получим: $25,12_{10} \approx 11001,0001_2$.

В таблице 1.1 для примера приведен натуральный ряд чисел в различных системах счисления.

Переход из двоичной системы счисления в восьмеричную или шестнадцатеричную может быть выполнен более простым путем. Так как $8 = 2^3$, а $16 = 2^4$, то один разряд числа, записанного в восьмеричной системе счисления, преобразуется в три разряда, а один разряд в числа в шестнадцатеричной системе счисления – в четыре разряда числа двоичной системы счисления.

Таблица 1.1

Натуральный ряд чисел в различных системах счисления

Десятеричная	Шестнадцатеричная	Восьмеричная	Двоичная	Десятеричная	Шестнадцатеричная	Восьмеричная	Двоичная
0	0	0	0	10	A	12	1010
1	1	1	1	11	B	13	1011
2	2	2	10	12	C	14	1100
3	3	3	11	13	D	15	1101
4	4	4	100	14	E	16	1110
5	5	5	101	15	F	17	1111
6	6	6	110	16	10	20	10000
7	7	7	111	17	11	21	10001
8	8	10	1000	18	12	22	10010
9	9	11	1001	19	13	23	10011

1.2. Кодирование чисел в двоичной системе счисления

В двоичном счислении любое число можно представить двумя цифрами: 0 и 1. Для представления этих чисел в цифровых системах достаточно иметь электронные схемы, которые могут принимать два состояния, четко различающиеся значением какой-либо электрической величины — напряжения или тока. Одному из значений этой величины соответствует цифра 0, другому — 1. Относительная простота создания электронных схем с двумя электрическими состояниями и привела к тому, что двоичное представление чисел доминирует в современной цифровой технике.

Для упрощения выполнения арифметических операций в цифровой технике вводят специальные двоичные коды чисел. Например, операцию вычитания в цифровых системах реализуют с помощью операции сложения, представляя вычитаемое в одном из специальных кодов. Одной из разновидностей таких кодов является *обратный код* числа \bar{A}_2 , который получается заменой всех 0 в числе на 1 и наоборот (табл. 1.2). Более подробно алгоритмы выполнения арифметических операций, используемые в цифровых системах, будут рассмотрены в следующем параграфе. Еще одной разновидностью специальных кодов является *дополнительный код* числа \tilde{A}_2 , который образуется из обратного прибавлением 1 к младшему разряду (табл. 1.2). Дополнительный код \tilde{A}_2 числа A_2 называется также дополнением числа до 2, так как для цифр в каждом разряде числа $a_j + \tilde{a}_j = 10$ (число 2 в двоичной форме).

Таблица 1.2

Наиболее распространенные коды чисел от 0 до 15

Десятичное число	Форма представления															
	Двоичное счисление				Обратный код				Дополнительный код				Циклический код Грея			
A_{10}	a_3	a_2	a_1	a_0	b_3	b_2	b_1	b_0	c_3	c_2	c_1	c_0	d_3	d_2	d_1	d_0
0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
1	0	0	0	1	1	1	1	0	1	1	1	1	0	0	0	1
2	0	0	1	0	1	1	0	1	1	1	1	0	0	0	1	1
3	0	0	1	1	1	1	0	0	1	1	0	1	0	0	1	0
4	0	1	0	0	1	0	1	1	1	1	0	0	0	1	1	0
5	0	1	0	1	1	0	1	0	1	0	1	1	0	1	1	1
6	0	1	1	0	1	0	0	1	1	0	1	0	0	1	0	1
7	0	1	1	1	1	0	0	0	1	0	0	1	0	1	0	0
8	1	0	0	0	0	1	1	1	1	0	0	0	1	1	1	0
9	1	0	0	1	0	1	1	0	0	1	1	1	1	1	0	1
10	1	0	1	0	0	1	0	1	0	1	1	0	1	1	1	1
11	1	0	1	1	0	1	0	0	0	1	1	1	1	1	1	0
12	1	1	0	0	0	0	1	1	0	1	0	0	1	0	1	0
13	1	1	0	1	0	0	1	0	0	0	1	1	1	0	1	1
14	1	1	1	0	0	0	0	1	0	0	1	0	1	0	0	1
15	1	1	1	1	0	0	0	0	0	0	0	1	1	0	0	0

Помимо рассмотренных кодов в цифровых системах используются и другие способы двоичного представления чисел. В качестве примера в табл. 1.2 приведен

циклический код Грея, особенность этого кода заключается в том, что при переходе к каждому последующему числу в коде изменяется значение только одного двоичного разряда. При этом 2-разрядные числа образуют циклическую последовательность 00 – 01 – 11 – 10 – 00 (0 – 1 – 3 – 2 – 0), 3-разрядные — последовательность 000 – 001 – 011 – 010 – 110 – 111 – 101 – 100 – 000 (0 – 1 – 3 – 2 – 6 – 7 – 5 – 4 – 0) и т. д. Такая цикличность кода является весьма удобной, например, для кодирования угловых перемещений в преобразователях угла поворота в цифровой код.

Перевод десятичных чисел в двоичный код требует использования достаточно сложных схем-преобразователей и занимает относительно долгое время. Более просто и быстро осуществляется перевод десятичных чисел в двоично-десятичный код. При этом цифра в каждом разряде десятичного числа заменяется соответствующим 4-разрядным двоичным числом (тетрадой) согласно таблице 1.3. Например, число $A_{10} = 729$ в двоично-десятичном коде записывается в виде

$$A_{2/10} = \underbrace{0111}_{тетрада2} \underbrace{0010}_{тетрада1} \underbrace{1001}_{тетрада0}$$

Для выполнения арифметических операций над двоично-десятичными числами наиболее удобно использовать так называемые *самодополняющие* коды, к числу которых относятся код Айкена, код «с избытком 3» (табл. 1.3) и некоторые другие. Код Айкена отличается от обычного двоично-десятичного, имеющего весовые коэффициенты разрядов в тетрадах 8—4—2—1, другими значениями весовых коэффициентов разрядов: 2—4—2—1. Код «с избытком 3» получается из обычного двоично-десятичного арифметическим прибавлением числа 3 (двоичное число 0011).

Таблица 1.3

Наиболее распространенные виды двоично-десятичного кодирования

Десятичное число	Двоично-десятичный код (8-4-2-1)				Код Айкена (2-4-2-1)				Код «с избытком три»			
	a_3	a_2	a_1	a_0	b_3	b_2	b_1	b_0	c_3	c_2	c_1	c_0
A_{10}	a_3	a_2	a_1	a_0	b_3	b_2	b_1	b_0	c_3	c_2	c_1	c_0
0	0	0	0	0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	0	1	0	0	1	0	1
3	0	0	1	1	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	1	1	1	0	0	0
6	0	1	1	0	1	1	0	0	1	0	0	1
7	0	1	1	1	1	1	0	1	1	0	1	0
8	1	0	0	0	1	1	1	0	1	0	1	1
9	1	0	0	1	1	1	1	1	1	1	0	0

Как видно из табл. 1.3, обратный код $\bar{A}_{2/10}$ числа, представленного в каком-либо самодополняющем двоично-десятичном коде $A_{2/10}$, является его двоичным *дополнением до 9*. Например, число 5 в коде «с избытком 3» $A_{2/10} = 1000$ имеет обратный код $\bar{A}_{2/10} = 0111$, соответствующий числу 4 в коде «с избытком 3», которое «дополняет» число 5 до 9, так как $5 + 4 = 9$.

Можно использовать для кодирования одного разряда десятичных и более

четырёх двоичных разрядов. Два таких кода даны в таблице 1.4. В коде «два из пяти» для представления десятичных цифр используются только те комбинации, которые содержат две 1 и три 0. В *циклическом коде Джонсона* переход к последующему числу осуществляется путем последовательной замены 0 на 1, начиная справа, до заполнения всех разрядов единицами, а затем заменой 1 на 0 до заполнения нулями. При таком кодировании используется избыточное количество личных разрядов, т. е. их требуется больше, чем при представлении в двоичной системе исчисления.

Применение таких кодов, использующих для представления десятичных цифр избыточное число двоичных разрядов (более четырех), позволяет обнаруживать некоторые ошибки, которые могут возникнуть при передаче чисел в результате неисправностей или действия помех. Например, при использовании кода «два из пяти» легко обнаруживаются ошибки типа ложного перехода 0 в 1 и наоборот, так как при этом в двоичном 5-разрядном коде оказываются соответственно три или одна 1 вместо двух 1 при правильном функционировании. При использовании кода Джонсона ошибочными являются комбинации, при которых один или несколько 0 оказываются между 1 или наоборот.

Таблица 1.4

Примеры избыточных двоично-десятичных кодов

Десятичное число	Код «два из пяти»					Циклический код Джонсона					
	A_{10}	a_4	a_3	a_2	a_1	a_0	b_4	b_3	b_2	b_1	b_0
0	0	0	0	1	1	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	1
2	0	0	1	1	0	0	0	0	1	1	1
3	0	1	0	0	1	0	0	1	1	1	1
4	0	1	0	1	0	0	1	1	1	1	1
5	0	1	1	0	0	1	1	1	1	1	1
6	1	0	0	0	1	1	1	1	1	1	0
7	1	0	0	1	0	1	1	1	1	0	0
8	1	0	1	0	0	1	1	0	0	0	0
9	1	1	0	0	0	1	0	0	0	0	0

1.3. Выполнение арифметических операций в цифровых системах

Арифметические операции над двоичными числами могут производиться по тем же правилам, что и над десятичными. Однако с целью упрощения цифровых систем для выполнения сложения, умножения, деления обычно применяются алгоритмы, не используемые в обычной десятичной арифметике.

Выполнение операций сложения-вычитания над двоичными числами. Для выполнения операций сложения и вычитания в цифровой системе удобно представлять отрицательные числа в обратном коде (табл. 1.2), а положительные - в обычном коде. Вычитание заменяется сложением с переводом вычитаемого в обратный код, если это число положительное, и в обычный код (т. е. перевод \bar{B}_2 в обратный код), если число отрицательное. Если результат операции получается отрицательным ($z_S = 1$), то он представлен в обратном коде, если положительным ($z_S=0$) — в обычном двоичном коде.

Пример 1.3. Сложить два числа $A_{10} = 12, B_{10} = -4$.

Решение:

$$\begin{array}{r|l}
 \text{Знак} & \text{Число} \\
 z_A \rightarrow 0 & 01100 \leftarrow A_2 = A_{10} = 12 \text{ (прямой код),} \\
 + & \\
 z_B \rightarrow 0 & 11011 \leftarrow \bar{B}_2 = B_{10} = -4 \text{ (обратный код),} \\
 \hline
 & 100111 \\
 & \xrightarrow{\text{циклический перенос}} \\
 z_S \rightarrow 0 & 01000 \leftarrow S_2 = S_{10} = 8.
 \end{array}$$

Пример 1.4. Вычесть из числа $A_{10} = -21$ число $B_{10} = -14$.

Решение:

$$\begin{array}{r|l}
 \text{Знак} & \text{Число} \\
 z_A \rightarrow 1 & 01010 \leftarrow \bar{A}_2 = A_{10} = -21 \text{ (обратный код),} \\
 - & \\
 z_B \rightarrow 1 & 10001 \leftarrow \bar{B}_2 = B_{10} = -14 \text{ (обратный код),} \\
 \hline
 & 101010 \leftarrow \bar{A}_2 \\
 + & \\
 & 011110 \leftarrow B_2 \text{ (перевод } \bar{B}_2 \text{ в обратный код)} \\
 \hline
 z_S \rightarrow 1 & 11000 \leftarrow \bar{S}_2 = S_{10} = -7.
 \end{array}$$

Недостатком использования обратного кода является образование циклического переноса, который приводит к повторению операции сложения, что существенно увеличивает время выполнения действий. Поэтому во многих случаях предпочтительнее использовать дополнительный код числа \tilde{A}_2 (табл. 1.2). При использовании дополнительного кода отпадает необходимость в циклическом переносе и перенос, который может возникать в знаковом разряде числа, не учитывается. В дополнительном коде представляются все отрицательные числа. Вычитание заменяется сложением с переводом вычитаемого в дополнительный код, если это число положительное, или переводом из дополнительного кода в обычный, если вычитаемое отрицательное. Если знаковый разряд результата $z_S = 1$, то полученное число отрицательное и представлено в дополнительном коде. Если $z_S = 0$, то результат положительный и представлен в обычном коде.

Пример 1.5. Сложить два числа $A_{10} = 12, B_{10} = -4$, используя дополнительный код.

Решение:

$$\begin{array}{r|l}
 \text{Знак} & \text{Число} \\
 z_A \rightarrow 0 & 01100 \leftarrow A_2 = A_{10} = 12 \text{ (прямой код),} \\
 + & \\
 z_B \rightarrow 1 & 11100 \leftarrow \tilde{B}_2 = B_{10} = -4 \text{ (дополнительный код),} \\
 \hline
 & 101000 \\
 & \xrightarrow{\text{циклический перенос}} \\
 z_S \rightarrow 0 & 01000 \leftarrow S_2 = S_{10} = 8.
 \end{array}$$

Пример 1.6. Вычтись из числа $A_{10} = -21$ число $B_{10} = -14$, используя дополнительный код.

Решение:

Знак	Число
$z_A \rightarrow 1$	01011 $\leftarrow \tilde{A}_2 = A_{10} = -21$ (дополнительный код),
-	
$z_B \rightarrow 1$	11100 $\leftarrow \tilde{B}_2 = B_{10} = -14$ (дополнительный код),
1	01011 $\leftarrow \tilde{A}_2$ (дополнительный код)
+	
0	01110 $\leftarrow B_2$ (обычный код)
$z_S \rightarrow 1$	11001 $\leftarrow \tilde{S}_2 = S_{10} = -7$.

Сложение и вычитание двоичных чисел с применением дополнительного кода выполняются проще и быстрее, хотя преобразование чисел в дополнительный код несколько сложнее, чем в обратный.

Выполнение операций умножения-деления над двоичными числами. Для выполнения умножения и деления двоичных чисел используются различные методы, наиболее употребительные из которых описаны ниже. Умножение или деление двоичного числа на $2^k = 2, 4, 8, 16$ и т. д. производится путем сдвига значащих цифр числа соответственно влево или вправо на $k = 1, 2, 3, 4$ и т. д. разрядов.

Перемножение двоичных чисел чаще всего выполняется с помощью метода *умножения со старших разрядов*. При этом множитель B_2 предварительно сдвигается влево на n_A разрядов, где n_A — число разрядов множимого A_2 . Затем проверяется значение младшего разряда множимого a_0 . Если $a_0 = 1$, то первое частичное произведение R_1 образуется путем арифметического сложения A и $B \cdot 2^{n_A}$ (множитель, сдвинутый на n_A разрядов) и последующего сдвига, суммы вправо на один разряд. При этом крайний правый разряд суммы выходит за пределы разрядной сетки и не учитывается. Если $a_0 = 0$, то частичное произведение образуется только сдвигом вправо на один разряд с потерей крайнего правого разряда. Далее проверяется значение младшего разряда $(r_0)_1$ полученного частичного произведения и образуется второе частичное произведение R_2 суммированием R_1 и $B \cdot 2^{n_A}$ со сдвигом суммы вправо на один разряд (при $(r_0)_1 = 0$). Аналогичная процедура получения частичных произведений повторяется n_B раз, где n_B — число значащих разрядов множителя. Последнее частичное произведение является результатом умножения $S_2 = R_{n_B}$.

Таким образом, умножение двоичных чисел производится путем последовательного выполнения операций сложения и сдвига. Поэтому умножение требует существенно большего времени работы цифрового устройства, чем сложение и вычитание.

Для деления двоичных чисел в цифровых системах чаще всего используется метод *деления с восстановлением*. Если делимое A_2 больше делителя B_2 , то предварительно производится масштабирование делимого A_2 путем его сдвига вправо на l разрядов, чтобы получить $A'_2 < B_2$. Затем сдвигом A'_2 влево на один разряд образуют число $2A'_2$ и выполняют вычитание $(2A'_2 - B_2)$. Если получаемый в результате промежуточный остаток $R_1 = (2A'_2 - B_2)$ имеет отрицательный знак

($z_{R_1} = 1$), то старший разряд частного $s_{-1} = 0$, если положительный знак ($z_{R_1} = 0$), то $s_{-1} = 1$. При $s_{-1} = 1$ остаток R_1 удваивается сдвигом влево, при $s_{-1} = 0$ вместо остатка используется число $2A'_2$, которое также удваивается. Из полученного значения $2R_1$ или $4A'_2$ снова вычитается B_2 . Если результат R_2 отрицательный, то следующий разряд частного $s_{-2} = 0$, если положительный, то $s_{-2} = 1$. Вычитание выполняется с помощью одного из описанных выше методов. Для получения частного S'_2 с точностью до k -го знака данная процедура повторяется k раз. При этом анализируется знак каждого промежуточного остатка z_{R_i} и следующий остаток определяется как разность $R_{i+1} = (2X_i - B_2)$, где $X_i = R_i$ при $z_{R_i} = 0$, или $X_i = 2R_{i-1}$ при $z_{R_i} = 1$, где $i = 0, 1, \dots, k$, $X_0 = 2A_2$. В результате получается частное от деления A_2 на B_2 в виде правильной дроби $S_2 = 0, s_{-1}, s_{-2}, \dots, s_{-k}$ и остаток от деления $P'_2 = R_k \cdot 2^k$. Значения частного S_2 и остатка P_2 образуются в результате восстановления масштаба с помощью сдвига S'_2 и P'_2 влево на l разрядов.

Таким образом, деление чисел выполняется путем последовательного выполнения операции вычитания и сдвига. Знак результата при умножении и делении получается суммирование знаковых разрядов без учета переноса, получающегося при $z_A = z_B = 1$.

Пример 1.7. Умножить число $A_{10} = 13$ на число $B_{10} = 11$.

Решение:

$$\begin{array}{r}
 1101 \leftarrow A_2 = A_{10} = 13, \\
 \times \qquad \qquad \qquad n_A = n_B = 4, \\
 1011 \leftarrow B_2 = B_{10} = 11. \\
 \\
 00001101 \rightarrow a_0 = 1 \text{ (суммирование и сдвиг)} \\
 + \\
 10110000 \leftarrow B_2 \cdot 2^{n_A} \\
 \hline
 10111101 \rightarrow \text{сдвиг суммы} \\
 R_1 \rightarrow 01011110 \rightarrow (r_0)_1 = 0 \text{ (сдвиг)} \\
 R_2 \rightarrow 00101111 \rightarrow (r_0)_2 = 1 \text{ (суммирование и сдвиг)} \\
 + \\
 1011000 \leftarrow B_2 \cdot 2^{n_A} \\
 \hline
 11011111 \rightarrow \text{сдвиг суммы} \\
 R_3 \rightarrow 01101111 \rightarrow (r_0)_3 = 1 \text{ (суммирование и сдвиг)} \\
 + \\
 10110000 \leftarrow B_2 \cdot 2^{n_A} \\
 \hline
 100011111 \rightarrow \text{сдвиг суммы} \\
 R_4 \rightarrow 10001111 \rightarrow S_2 = S_{10} = 143.
 \end{array}$$

Пример 1.8. Разделить $A_2=0,0111$ на $B_2=0,1100$.

Решение: Так как $A_2 < B_2$, то масштабирование в этом случае не требуется.

$$\begin{array}{r}
 X_0 = 2A_2 \rightarrow 0,1110 \\
 - \\
 B_2 \rightarrow 0,1100 \\
 \hline
 R_1 = (2A_2 - B_2) \rightarrow +0,0010 > 0, s_1 = 1, \\
 X_1 = 2R_1 \rightarrow 0,0100 \\
 - \\
 B_2 \rightarrow 0,1100 \\
 \hline
 R_2 = (2R_1 - B_2) \rightarrow -0,1000 < 0, s_2 = 0, \\
 X_2 = 4R_1 \rightarrow 0,1000 \\
 - \\
 B_2 \rightarrow 0,1100 \\
 \hline
 R_3 = (4R_1 - B_2) \rightarrow -0,0100 < 0, s_3 = 0, \\
 X_3 = 8R_1 \rightarrow 1,0000 \\
 - \\
 B_2 \rightarrow 0,1100 \\
 \hline
 R_4 = (8R_1 - B_2) \rightarrow 0,0100 > 0, s_4 = 1.
 \end{array}$$

Получаем частное от деления $S_2=0,1001$ и остаток $P_2=0,01 \cdot 2^{-4}=0,000001$.

Описанные выше методы выполнения арифметических операций, рассмотренные для чисел с фиксированной запятой, применяются и для операций над числами с плавающей запятой. Однако при этом требуется еще выполнение определенных операций с порядками чисел. При сложении и вычитании чисел с плавающей запятой предварительно производится выравнивание порядков p_A и p_B обоих чисел путем сдвига мантиссы одного из них, например B_2 , на $(p_A - p_B)$ разрядов вправо, если $p_A > p_B$, либо влево, если $p_A < p_B$. Порядок результатов при этом будет $p_S = p_A$. При умножении или делении чисел с плавающей запятой порядок результата равен соответственно сумме или разности порядков: $p_S = p_A \pm p_B$.

Выполнение арифметических операций над двоично-десятичными числами. Как было указано в параграфе 1.2, для выполнения арифметических операций над двоично-десятичными числами наиболее удобно использовать самополняющиеся коды (табл. 1.3). Рассмотрим операции сложения-вычитания для чисел, записанных в двоично-десятичном коде.

При использовании самополняющихся двоично-десятичных кодов вычитание заменяется сложением аналогично тому, как это производится для двоичных чисел в случае применения обратного кода. При этом все отрицательные числа представляются в обратном самополняющемся коде. При выполнении вычитания вычитаемое преобразуется в обратный код и производится его суммирование с уменьшаемым. В случае образования переноса в старшем (знаковом) разряде выполняется его прибавление к младшему разряду младшей тетрады (циклический перенос). Затем производится дополнительная операция — коррекция кода в каждой из тетрад.

Рассмотрим выполнения операций сложения-вычитания с применением кода «с избытком 3». Исходные положительные числа представляются в коде «с избытком

3», а отрицательные — в обратном коде «с избытком 3». При сложении суммируются соответствующие тетрады двоичных разрядов. При вычитании сначала вычитаемое переводится в обратный код, а затем производится суммирование. Если при суммировании в тетраде не образуется перенос из старшего разряда, то сумма, полученная в данной тетраде, превышает на 3 действительное значение результата в коде «с избытком 3». Поэтому для коррекции значения суммы, полученной в этой тетраде, необходимо вычесть 3, что эквивалентно прибавлению 1101 (3 в дополнительном коде). Если в тетраде образуется перенос, т.е. десятичное число 16, то 10 поступает в виде двоичной 1 в младший разряд следующей тетрады, а 6 необходимо добавить к сумме, полученной в данной тетраде. Но так как для коррекции кода необходимо вычесть 3, то достаточно добавить $(6-3) = 3$, т.е. 0011. Переносы, возникающие при коррекции, не учитываются. Знаковые разряды чисел также суммируются. Перенос, образуемый в знаковом разряде, циклически переносится в младший разряд младшей тетрады. Коррекция кода требуется и при использовании других двоично-десятичных кодов. Поэтому выполнение арифметических операций при двоично-десятичном представлении сложнее, чем при двоичном. Однако при обработке большого количества данных этот недостаток компенсируется простотой перевода чисел в двоично-десятичную форму.

Пример 1.9. Сложить число $A_{10}=34$ с числом $B_{10}=19$, используя двоично-десятичное кодирование.

Решение:

	знак	тетрада 1	тетрада 0	
$z_A \rightarrow$	0	0110	0111	$\leftarrow A'_{2/10} = A_{10} = 34,$
+				
$z_B \rightarrow$	0	0100	1100	$\leftarrow B'_{2/10} = B_{10} = 19,$
	0	1010	0011	
+				
		1		
	0	1011	0011	
+				
		1101	0011	\leftarrow коррекция
$z_S \rightarrow$	0	1000	0110	$\leftarrow S'_{2/10} = S_{10} = 53.$
	1			

Перенос не учитывается.

Пример 1.10. Вычесть из числа $A_{10}=34$ число $B_{10}=19$, используя двоично-десятичное кодирование.

Решение:

		знак	тетрада 1	тетрада 0		
z_A	→	0	0110	0111	←	$A'_{2/10} = A_{10} = 34,$
		-				
z_B	→	0	0100	1100	←	$B'_{2/10} = B_{10} = 19,$
z_A	→	0	0110	0111	←	$A'_{2/10}$
		+				
z_B	→	1	1011	0011	←	$\overline{B'}_{2/10}$
		1	0001	1010		
		+				
		1	0001	1010		
		10	0001	1010		
				1		циклический перенос
		0	0001	1011		
		0	0011	1101	←	коррекция
z_S	→	0	0100	1000	←	$S'_{2/10} = S_{10} = 15.$
				1		перенос не учитывается

Контрольные вопросы и упражнения к разделу 1

1. В чем отличие позиционной системы счисления от непозиционной?
2. Как осуществляется перевод числа из одной системы счисления в другую: с большим основанием; с меньшим основанием?
3. Какие двоичные коды чисел Вы знаете?
4. Приведите наиболее распространенные типы двоично-десятичного кодирования.
5. Преобразуйте число $X_2=101001$ в X_{16} .
6. Преобразуйте число $X_{10}=23$ в X_8 .
7. Опишите алгоритм выполнения сложения вычитания над двоичными числами, применяемыми в цифровой технике.
8. При помощи, каких операций выполняется операция умножения двоичных чисел в цифровой технике?
9. Приведите алгоритм умножения двоичного числа на число, отличное от 2^k , где $k=1, 2, 3, 4, \dots$

2. ОСНОВЫ АЛГЕБРЫ ЛОГИКИ И ВЫПОЛНЕНИЕ ЛОГИЧЕСКИХ ОПЕРАЦИЙ

2.1. Логические константы и переменные. Операции Булевой алгебры

Для описания алгоритмов работы цифровых устройств необходим соответствующий математический аппарат. Такой аппарат для решения задач формальной логики в середине прошлого века разработал ирландский математик Джорж Буль. По его имени математический аппарат и получил название *булевой алгебры*, или *алгебры логики*.

Булева алгебра — это математическая система, оперирующая двумя понятиями: «*событие истинно*» и «*событие ложно*». Естественно ассоциировать эти понятия с цифрами, используемыми в двоичной системе счисления. Далее будем их называть соответственно логическими единицей (лог. 1) и нулем (лог. 0).

Два элемента булевой алгебры, а именно «*событие истинно*» и «*событие ложно*», называются ее константами. Будем понимать под ними значения соответственно лог. 1 и лог. 0.

В общем случае логические выражения являются функциями *логических переменных* A, B, C , каждая из которых может иметь значения 0 или 1. Если имеется k логических переменных, то они образуют 2^k возможных логических наборов из 0 и 1. При $k=1$: $A=0$ или $A=1$; При $k=2$: $AB=00, 01, 10, 11$ и т. д. Для каждого набора переменных логическая функция F может принимать значение 0 или 1. Поэтому для k переменных можно образовать $l_k=2^{2^k}$ различных логических функций. Таким образом, при увеличении k число l растет чрезвычайно быстро: при $k=2$ получим $l_k=16$; при $k=3$ получим $l_3=256$; при $k=4$ получим $l_4=65\,536$ и т.д.

Всё возможные логические функции k переменных можно образовать с помощью трех основных операций:

- **логическое сложение.** Эту операцию называют операцией ИЛИ или *дизъюнкцией*. Постулаты логического сложения двух переменных приведены в таблице 2.1.
- **логическое умножение.** Эту операцию называют операцией И или *конъюнкцией*. Постулаты логического умножения двух переменных приведены в табл. 2.1.

Следует отметить, что данная операция также справедлива для произвольного числа переменных. Она соответствует математической операции пересечения множеств. Число переменных, над которыми выполняется операция, также обозначается цифрой. В данном случае можно сказать, что табл. 2.1 определяет операцию 2И.

- **отрицание.** Операцию отрицания называют *инверсией* или *дополнением*. Для ее обозначения используют черту над соответствующим выражением. Постулаты, определяющие операцию отрицания, приведены в таблице 2.2.

Для обозначения эквивалентности логических выражений используется знак равенства « \Leftrightarrow ».

Таблица 2.1

Операции дизъюнкции и конъюнкции.

A	B	$F = A+B$	$F = A \cdot B$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1

Таблица 2.2

Операция инверсии.

A	$F = \bar{A}$
0	1
1	0

2.2. Основные аксиомы и законы алгебры-логики

Для рассмотренных логических операций справедлив ряд аксиом (тождеств) и законов, основные из которых даны в таблице 2.3. Следует отметить, что алгебраические выражения тождеств и законов в таблице 2.3. заданы парами, и взаимной заменой операций И, ИЛИ и символов 0 и 1 из одного выражения получается другое. Используя данные тождества и законы, можно получать новые логические выражения, а также доказывать справедливость тех или иных законов на основании других.

Таблица 2.3

Основные аксиомы и законы алгебры-логики

Аксиомы (тождества)	$1+A = 1.$ $0 \cdot A = 0.$	(2.1)
	$0+A = A$ $1 \cdot A = A$	(2.2)
	$A+A = A$ $A \cdot A = A$	(2.3)
	$A+\bar{A} = 1$ $A \cdot \bar{A} = 0$	(2.4)
	$\bar{\bar{A}} = A$	(2.5)
Законы коммутативности	$A+B = B+A$ $A \cdot B = B \cdot A$	(2.6)
Законы ассоциативности	$A+B+C = A+(B+C)$ $A \cdot B \cdot C = A \cdot (B \cdot C)$	(2.7)
Законы дистрибутивности	$A(B+C) = (A \cdot B)+(A \cdot C)$ $A+(B \cdot C) = (A+B) \cdot (A+C)$	(2.8)
Законы дуальности (теоремы Де-Моргана)	$\overline{A+B} = \bar{A} \cdot \bar{B}$ $\overline{A \cdot B} = \bar{A} + \bar{B}$	(2.9)
Законы поглощения	$A+A \cdot B = A$ $A \cdot (A+B) = A.$	(2.10)

Например, с помощью второго закона дистрибутивности (2.8) и тождества (2.4) получаем соотношение:

$$A + \bar{A} \cdot B = (A + \bar{A}) \cdot (A + B) = A + B \quad (2.11)$$

Используя первый закон дистрибутивности (2.8), тождества (2.1), (2.2) и закон коммутативности (2.6), получаем доказательство справедливости второго закона поглощения (2.10)

$$A(A+B) = AA + AB = A + AB = A(1+B) = A.$$

Применение данных тождеств и законов позволяет производить упрощение логических функций, т.е. находить для них выражения, имеющие наиболее простую форму.

Используя законы ассоциативности, любую логическую функцию многих переменных ($k > 2$) можно представить в виде комбинации функции двух переменных.

Полный набор $2^{2^2} = 16$ логических функций двух переменных дан в таблице 2.4. Каждая функция обозначает одну из 16 возможных логических операций над двумя переменными A, B и имеет собственное название и условное обозначение.

Таблица 2.4

Полный набор логических функций для двух переменных

A	0	0	1	1	Условное обозначение и алгебраическое выражение	Название функции
B	0	1	0	1		
F_0	0	0	0	0	$F_0 = 0$	Постоянный ноль
F_1	0	0	0	1	$F_1 = AB$	Конъюнкция
F_2	0	0	1	0	$F_2 = A \rightarrow B = \bar{A}B$	Запрет
F_3	0	0	1	1	$F_3 = A$	Тождественность A
F_4	0	1	0	0	$F_4 = B \rightarrow A = \bar{A}B$	Запрет
F_5	0	1	0	1	$F_5 = B$	Тождественность B
F_6	0	1	1	0	$F_6 = A \oplus B = \bar{A}B + \bar{A}\bar{B}$	Исключительное ИЛИ (неравнозначность)
F_7	0	1	1	1	$F_7 = A+B$	Дизъюнкция
F_8	1	0	0	0	$F_8 = A \downarrow B = \overline{A+B}$	Стрелка Пирса (ИЛИ-НЕ)
F_9	1	0	0	1	$F_9 = \overline{A \oplus B} = AB + \bar{A}\bar{B}$	Равнозначность (исключительное ИЛИ-НЕ)
F_{10}	1	0	1	0	$F_{10} = \bar{B}$	Инверсия B
F_{11}	1	0	1	1	$F_{11} = B \rightarrow A = A + \bar{B}$	Импликация от B к A
F_{12}	1	1	0	0	$F_{12} = \bar{A}$	Инверсия A
F_{13}	1	1	0	1	$F_{13} = A \rightarrow B = \bar{A} + B$	Импликация от A к B
F_{14}	1	1	1	0	$F_{14} = A / B = \bar{A}B$	Штрих Шеффера (И-НЕ)
F_{15}	1	1	1	1	$F_{15} = 1$	Постоянная единица

2.3. Способы записи функций алгебры логики

Рассмотрим некоторое логическое устройство, на входе которого присутствует некоторый n -разрядный двоичный код $x_{n-1} \dots x_1 x_0$, на выходе соответственно m -разрядный двоичный код $z_{m-1} \dots z_1 z_0$, (рис. 2.1). Для того, чтобы описать поведение этой схемы, необходимо определить зависимость каждой из m выходных переменных z_i от входного двоичного кода $x_{n-1} \dots x_1 x_0$.

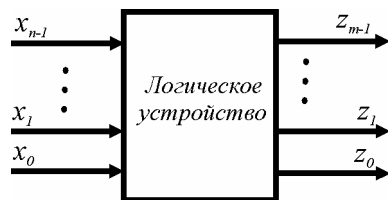


Рис. 2.1. Обобщенная схема логического устройства.

Зависимость выходных переменных z_i , выраженная через совокупность входных переменных $x_{n-1} \dots x_1 x_0$ с помощью операций алгебры логики, носит название *функции алгебры логики* (ФАЛ). Иногда данную зависимость также называют переключательной функцией. Задать ФАЛ — это значит определить значения z_i для всех возможных комбинаций переменных $x_{n-1} \dots x_1 x_0$. Очевидно, что для n -разрядного двоичного кода $x_{n-1} \dots x_1 x_0$ существует 2^n различных значений z_i .

Функция называется *полностью определенной*, если заданы 2^n ее значений. Если часть значений функции не задана, то она называется *частично определенной* или *недоопределенной*.

Иногда известно, что по условиям работы устройства появление некоторых входных кодов невозможно, и поэтому значения ФАЛ на этих кодах не задаются. При этом возникают так называемые *факультативные* или *необязательные* значения функции, которые могут задаваться произвольными значениями. Входные коды, для которых ФАЛ имеет факультативные значения, называются *запрещенными*.

Устройства, поведение которых описывается при помощи ФАЛ, называют *логическими*.

Для описания ФАЛ могут быть использованы различные способы. Основными из них являются описание функции в словесной форме, в виде таблиц истинности, алгебраических выражений, последовательностей десятичных чисел, а также кубических комплексов.

Словесное описание ФАЛ. Данный вид описания наиболее часто применяется для первоначального, исходного описания поведения логического устройства. Проиллюстрируем словесное описание ФАЛ на примере.

Пример 2.1. Логическая функция трех переменных равна единице, если хотя бы две входные переменные равны единице.

Описание ФАЛ в виде таблицы истинности. Таблица, содержащая все возможные комбинации входных переменных $x_{n-1} \dots x_1 x_0$ и соответствующие им значения выходных переменных z_i , называется таблицей истинности, или комбинационной таблицей. В общем случае таблица истинности содержит 2^n строк и $m+n$ столбцов. Проиллюстрируем построение таблицы истинности на примере.

Пример 2.2. Составить таблицу истинности для ФАЛ из примера 2.1.

Решение. Данная таблица имеет четыре столбца.

Таблица 2.5

Таблица истинности для трех переменных

x_2	x_1	x_0	y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Описание ФАЛ в виде алгебраического выражения. При описании ФАЛ алгебраическим выражением используются две стандартные формы ее представления.

1. *Дизъюнктивная нормальная форма* (ДНФ). ДНФ называется логическая сумма элементарных логических произведений, в каждый из которых аргумент или его инверсия входят один раз.

Получена ДНФ может быть из таблицы истинности с использованием следующего алгоритма:

- для каждого набора переменных, на котором ФАЛ равна единице, записываются элементарные логические произведения входных переменных, причем переменные, равные нулю, записываются с инверсией. Полученные произведения называют *конституентами единицы*, или *минтермами* (m).
- логически суммируют все конституенты единицы (минтермы).

Пример 2.3. Записать ДНФ для ФАЛ, заданной в примере 2.2.

Решение. Составим таблицу конституент единицы (минтермов) для ФАЛ, заданной в примере 2.2.

Таблица 2.6

Минтермы ФАЛ $y(x_2, x_1, x_0)$

x_2	x_1	x_0	Минтермы (m)	Значение функции
0	0	0	$m_0 = \bar{x}_2 \bar{x}_1 \bar{x}_0$	$y_0=0$
0	0	1	$m_1 = \bar{x}_2 \bar{x}_1 x_0$	$y_1=0$
0	1	0	$m_2 = \bar{x}_2 x_1 \bar{x}_0$	$y_2=0$
0	1	1	$m_3 = \bar{x}_2 x_1 x_0$	$y_3=1$
1	0	0	$m_4 = x_2 \bar{x}_1 \bar{x}_0$	$y_4=0$
1	0	1	$m_5 = x_2 \bar{x}_1 x_0$	$y_5=1$
1	1	0	$m_6 = x_2 x_1 \bar{x}_0$	$y_6=1$
1	1	1	$m_7 = x_2 x_1 x_0$	$y_7=1$

Согласно приведенному выше алгоритму, используя минтермы из табл. 2.6 и основные аксиомы (тождества) алгебры-логики (табл. 2.3), получим:

$$y(x_2, x_1, x_0) = y_0 m_0 + y_1 m_1 + y_2 m_2 + y_3 m_3 + y_4 m_4 + y_5 m_5 + y_6 m_6 + y_7 m_7 = 0 \cdot (\bar{x}_2 \bar{x}_1 \bar{x}_0) + 0 \cdot (\bar{x}_2 \bar{x}_1 x_0) + 0 \cdot (\bar{x}_2 x_1 \bar{x}_0) + 1 \cdot (\bar{x}_2 x_1 x_0) + 0 \cdot (x_2 \bar{x}_1 \bar{x}_0) + 1 \cdot (x_2 \bar{x}_1 x_0) + 1 \cdot (x_2 x_1 \bar{x}_0) + 1 \cdot (x_2 x_1 x_0) = \bar{x}_2 x_1 x_0 + x_2 \bar{x}_1 x_0 + x_2 x_1 \bar{x}_0 + x_2 x_1 x_0$$

Дизъюнктивную нормальную форму, полученную суммированием конститuent единицы (минтермов), называют *совершенной дизъюнктивной нормальной формой* (СДНФ). В общем случае СДНФ любой логической функции можно представить в следующей форме:

$$F = \sum_{i=0}^{q-1} f_i m_i \quad (2.12)$$

где f_i – значение функции (0 или 1), m_i – минтерм, соответствующие i -ому набору переменных; $q = 2^n$ – количество различных значений ФАЛ при n заданных переменных.

2. *Конъюнктивная нормальная форма* (КНФ). КНФ называется логическое произведение элементарных логических сумм, в каждую из которых аргумент или его инверсия входят один раз.

Получена КНФ может быть из таблицы истинности с использованием следующего алгоритма:

а) для каждого набора переменных, на котором ФАЛ равна нулю, записывают элементарные логические суммы входных переменных, причем переменные, значения которых равны единице, записывают с инверсией. Полученные суммы называют *конститuentой нуля*, или *макстермами*.

б) логически перемножают все конститuentы нуля (макстермы).

Пример 2.4. Записать КНФ для ФАЛ, заданной в примере 2.2.

Решение. Составим таблицу конститuent нуля (макстермов) для ФАЛ, заданной в примере 2.2.

Таблица 2.7

Макстермы ФАЛ $y(x_2, x_1, x_0)$				
x_2	x_1	x_0	Макстермы (M)	Значение функции
0	0	0	$M_0 = x_2 + x_1 + x_0$	$y_0=0$
0	0	1	$M_1 = x_2 + x_1 + \bar{x}_0$	$y_1=0$
0	1	0	$M_2 = x_2 + \bar{x}_1 + x_0$	$y_2=0$
0	1	1	$M_3 = x_2 + \bar{x}_1 + \bar{x}_0$	$y_3=1$
1	0	0	$M_4 = \bar{x}_2 + x_1 + x_0$	$y_4=0$
1	0	1	$M_5 = \bar{x}_2 + x_1 + \bar{x}_0$	$y_5=1$
1	1	0	$M_6 = \bar{x}_2 + \bar{x}_1 + x_0$	$y_6=1$
1	1	1	$M_7 = \bar{x}_2 + \bar{x}_1 + \bar{x}_0$	$y_7=1$

Согласно приведенному выше алгоритму, используя макстермы из табл. 2.7 и основные аксиомы (тождества) алгебры-логики (табл. 2.3), получим:

$$y(x_2, x_1, x_0) = (y_0 + M_0) \cdot (y_1 + M_1) \cdot (y_2 + M_2) \cdot (y_3 + M_3) \cdot (y_4 + M_4) \times (y_5 + M_5) \cdot (y_6 + M_6) \cdot (y_7 + M_7) = (x_2 + x_1 + x_0) \cdot (x_2 + x_1 + \bar{x}_0) \times (x_2 + \bar{x}_1 + x_0) \cdot (\bar{x}_2 + x_1 + x_0)$$

Конъюнктивную нормальную форму, полученную суммированием конститuent нуля (макстермов), также называют *совершенной конъюнктивной нормальной формой* (СКНФ). В общем случае СКНФ любой логической функции можно представить в следующей форме:

$$F = \prod_{i=0}^{q-1} (f_i + M_i) \quad (2.13)$$

где f_i – значение функции (0 или 1), M_i – макстерм, соответствующие i -ому набору переменных;

Рассмотренные методики позволяют получить математическую форму записи для самой функции. Иногда удобнее применять не саму ФАЛ, а ее инверсию. В этом случае при использовании вышеописанных методик для записи СДНФ необходимо выбирать нулевые, а для записи СКНФ – единичные значения функции.

Пример 2.5. Для ФАЛ из примера 2.2 записать СДНФ и СКНФ инверсной функции.

Решение. Воспользовавшись таблицей 2.5, запишем

СДНФ: $\bar{y}(x_2, x_1, x_0) = \bar{x}_2 \bar{x}_1 \bar{x}_0 + \bar{x}_2 \bar{x}_1 x_0 + \bar{x}_2 x_1 \bar{x}_0 + x_2 \bar{x}_1 \bar{x}_0$

СКНФ: $\bar{y}(x_2, x_1, x_0) = (x_2 + \bar{x}_1 + \bar{x}_0) \cdot (\bar{x}_2 + x_1 + \bar{x}_0) \cdot (\bar{x}_2 + \bar{x}_1 + x_0) \cdot (\bar{x}_2 + \bar{x}_1 + \bar{x}_0)$

Описание ФАЛ в виде последовательности десятичных чисел. Иногда для сокращения записи ФАЛ представляют в виде последовательности десятичных чисел. При этом последовательно записывают десятичные эквиваленты двоичных кодов соответствующих конститuent единицы и нуля (минтермов и макстермов).

Пример 2.6. Записать в виде последовательности десятичных чисел ФАЛ из примеров 2.3 и 2.4.

Решение. В СДНФ из примера 2.3 первая конститuent единицы (минтерм - $\bar{x}_2 x_1 x_0$)

соответствует двоичному коду 011 (табл. 2.6). Десятичный эквивалент этого кода равен 3. Аналогично записываются все остальные конститuentы:

$$y(x_2, x_1, x_0) = \sum (3, 5, 6, 7).$$

В СКНФ из примера 2.4 первая конститuent нуля (макстерм $x_2 + x_1 + x_0$)

соответствует двоичному коду 000 (табл. 2.7). Десятичный эквивалент этого кода равен 0. Аналогично записывают все остальные конститuentы:

$$y(x_2, x_1, x_0) = \prod (0, 1, 2, 4).$$

Кубические комплексы. В последнее время широкое распространение получило так называемое кубическое представление ФАЛ. Такое представление использует ограниченное число символов и поэтому применяется при автоматизации процессов логического проектирования цифровых интегральных схем (ИС).

Основной кубической формы является представление каждого набора входных переменных в качестве n -мерного вектора. Вершины этих векторов геометрически могут быть представлены как вершины n -мерного куба. Отмечая точками вершины векторов, для которых ФАЛ равна единице, получаем геометрическое представление функции куба.

Пример 2.7. Задана ФАЛ $z(x_2, x_1, x_0) = \sum(3, 4, 5, 6, 7)$. Дать геометрическое представление в виде куба.

Решение. Графическое решение задачи проиллюстрировано на рисунке 2.2.

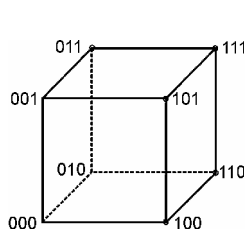


Рис. 2.2. Геометрическое представление ФАЛ.

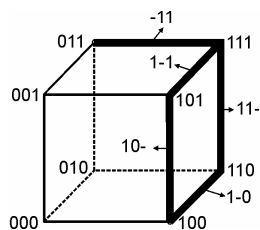


Рис. 2.3. Единичный кубический комплекс ФАЛ (см. пример 2.8).

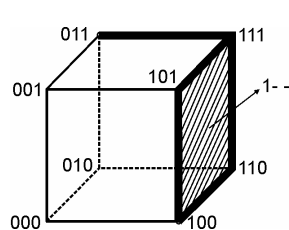


Рис. 2.4. Двоичный куб для ФАЛ (см. пример 2.8).

Очевидно, что наборы переменных, расположенные на концах ребер куба, отличаются только одной переменной. Такие наборы (коды) принято называть соседними.

Каждую функцию куба, в которой функция принимает единичное значение, называют *нулевым кубом* (0-кубом). Записывается 0-куб последовательностью образовавших его входных переменных, т.е. кодом, соответствующим конституенте единицы. Множество нулевых кубов образуют нулевой кубический комплекс K_0 ФАЛ.

Если два нулевых куба комплекса K_0 отличаются только по одной координате (переменной), т.е. два набора переменных, для которых ФАЛ равна единице, являются *соседними*, то они образуют *единичный куб* (1-куб). Геометрически это соответствует ребру исходного n -мерного куба (рис. 2.3), 1-куб записывается последовательностью общих элементов образовавших его 0-кубов с прочерком несовпадающих элементов. Множество единичных кубов образует единичный кубический комплекс K_1 .

Аналогично, если два единичных куба комплекса K_1 отличаются только по одной координате (переменной), то эти единичные кубы образуют *двоичный куб* (2-куб). Геометрически это соответствует грани исходного n -мерного куба (рис. 2.4). 2-куб также записывается последовательностью общих элементов, образовавших его 1-кубов с прочерком несовпадающих элементов, а множество двоичных кубов образуют двоичный кубический комплекс K_2 . И так далее.

Пример 2.8. Для ФАЛ из примера 2.7 записать кубические комплексы.

Решение. Нулевой кубический комплекс содержит пять членов по числу конституент единицы ФАЛ. $K_0 = (011, 100, 101, 110, 111)$.

Сравнивая записанные 0-кубы, можно увидеть, что 1-й и 5-й кубы

отличаются только первым членом. Поэтому они образуют 1-куб вида -11 . Аналогично, 2-ой и 3-й 0-кубы образуют 1-куб $10-$ и т.д. Единичный кубический комплекс заданной ФАЛ будет иметь вид: $K_1 = (-11, 10-, 11-, 1-1)$.

Аналогично может быть получен и двоичный кубический комплекс, состоящий из одного 2-куба: $K_2 = (1--)$.

Из сказанного следует, что размерность куба (его *ранг*) определяется числом несовпадающих координат, т.е. числом прочерков в его записи.

Объединение кубических комплексов K_0, K_1, \dots, K_m для ФАЛ n -переменных образует ее кубический комплекс: $K(z) = \bigcup (K_0, K_1, \dots, K_m)$.

Контрольные вопросы и упражнения к разделу 2

1. Что называется булевыми константами и переменными в алгебре логики?
2. Назовите основные операции булевой алгебры. Как они описываются с помощью таблиц истинности; с помощью алгебраических выражений.
3. Что отражают теоремы булевой алгебры? Сформулируйте теоремы Де-Моргана, законы ассоциативности, коммутативности, поглощения.
4. Какие функции алгебры логики называются полностью и частично определенными? Что такое факультативное значение функции и запрещенный код?
5. Приведите пример описания ФАЛ в словесной форме; в виде таблицы истинности; в виде алгебраического выражения; в дизъюнктивной и конъюнктивной нормальной формах; в виде последовательности чисел; в виде куба.
6. Что такое нулевой куб; единичный куб; двоичный куб; единичный и двоичный кубические комплексы; кубический комплекс?
7. Дайте определение ранга куба.

3. ЛОГИЧЕСКИЕ ЭЛЕМЕНТЫ И СХЕМЫ

3.1. Принцип двойственности

Пользуясь ФАЛ, мы до сих пор ничего не говорили о структуре логического устройства, представляя его аналогично устройству на рис. 2.1 в виде некоторого «черного ящика». Однако ФАЛ однозначно определяет и внутреннюю структуру логического устройства. Если мы располагаем элементарными узлами, реализующими основные логические операции, заданные постулатами в п. 2.1, то с их помощью можно построить логическую схему, выполняющую заданный алгоритм преобразования исходных логических переменных. В общем случае характер реальных логических переменных не имеет значения. Он может быть произвольным.

В соответствии с перечнем логических операций различают три основных логических элемента (ЛЭ): И, ИЛИ, НЕ. Условные графические обозначения этих ЛЭ показаны на рисунке 3.1.

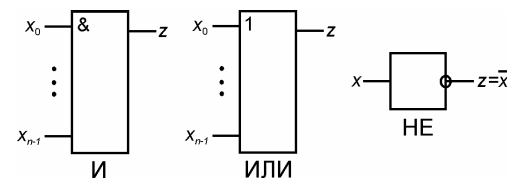


Рис. 3.1. Условные графические изображения логических элементов.

Следует еще раз подчеркнуть, что число элементов И или ИЛИ может быть произвольным. Элемент НЕ имеет всегда только один вход.

Для построения логической схемы необходимо ЛЭ, предназначенные для выполнения логических операций, указанных в ФАЛ, располагать от входа в порядке, определенном булевым выражением.

Пример 3.1. Построить структурную схему логического устройства по ФАЛ из примера 2.3 (т.е. $y(x_2, x_1, x_0) = \bar{x}_2x_1x_0 + x_2\bar{x}_1x_0 + x_2x_1\bar{x}_0 + x_2x_1x_0$).

Решение. Для реализации заданной ФАЛ в виде структурной логической схемы нам понадобится три ЛЭ, реализующих операцию НЕ, т.к. исходная ФАЛ формируется тремя переменными (x_2, x_1, x_0), которые входят в нее как в прямом, так и в инверсном виде. Операция дизъюнкции должна быть выполнена четыре раза над тремя переменными, таким образом, для ее реализации нам понадобится четыре ЛЭ, реализующих операцию 3И. Последней выполняется операция конъюнкции над четырьмя выражениями, для реализации которой потребуются ЛЭ, реализующий операцию 4ИЛИ. Пример структурной логической схемы, реализующей заданную ФАЛ, приведен на рис. 3.2.

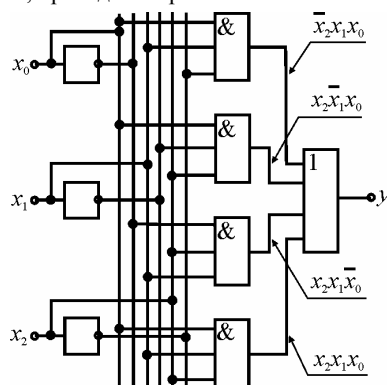


Рис. 3.2. Структурная схема логического устройства, реализующая ФАЛ вида

$$y(x_2, x_1, x_0) = \bar{x}_2x_1x_0 + x_2\bar{x}_1x_0 + x_2x_1\bar{x}_0 + x_2x_1x_0$$

При сравнении таблиц истинности для операций И и ИЛИ (см. табл. 2.1), легко заметить, что если в условиях, определяющих операцию И, значения всех переменных и самой функции заменить инверсией, а знак логического умножения – знаком логического сложения, получим постулаты, определяющие операцию ИЛИ, и наоборот (см. теоремы Де-Моргана (2.9)):

$$\text{если } x_1 \cdot x_0 = z, \text{ то } \bar{x}_1 + \bar{x}_0 = \bar{z},$$

$$\text{если } x_1 + x_0 = z, \text{ то } \bar{x}_1 \cdot \bar{x}_0 = \bar{z}.$$

Это свойство взаимного преобразования постулатов операций логического сложения и умножения носит название *принципа двойственности*.

Важным практическим следствием принципа двойственности является тот факт, что при записи логических выражений и, следовательно, построении логических схем, можно обойтись только двумя типами операций, например

операциями И и НЕ или операциями ИЛИ и НЕ.

Введем понятие функционально полной системы ЛЭ. *Функционально полной системой* называется совокупность ЛЭ, позволяющая реализовывать логическую схему произвольной сложности. Таким образом, системы двух элементов И и НЕ, а также ИЛИ и НЕ наравне с системой из трех элементов (И, ИЛИ, НЕ) являются функционально полными. На практике широкое применение получили ЛЭ, совмещающие функции элементов указанных выше функционально полных систем. Это ЛЭ, реализующие операции штрих Шеффера (табл. 2.4, операция И-НЕ) и стрелка Пирса (табл. 2.4, операция ИЛИ-НЕ). По определению каждый из этих ЛЭ также образует функционально полную систему. Их условные графические изображения приведены на рис. 3.3.

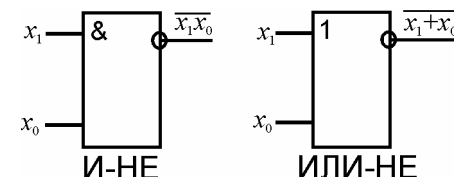


Рис. 3.3. Условные графические обозначения ЛЭ 2И-НЕ и 2ИЛИ-НЕ

В качестве примера рассмотрим реализацию логических операций И, ИЛИ и НЕ с использованием только элемента ИЛИ-НЕ.

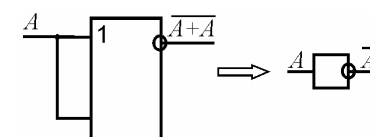


Рис. 3.4. Реализация ЛЭ НЕ на элементе ИЛИ-НЕ

Если ЛЭ 2ИЛИ-НЕ включен по схеме, показанной на рис. 3.4, то при подаче на его вход логической переменной A на его выходе получим логическое выражение вида $\overline{A+A}$, но согласно аксиоме (2.3) можем записать $\overline{A+A} = \bar{A}$. Таким образом, мы получили элемент, реализующий операцию логического отрицания (НЕ).

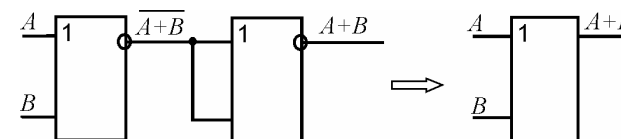


Рис. 3.5. Реализация ЛЭ ИЛИ на элементах ИЛИ-НЕ

Если на входы ЛЭ ИЛИ-НЕ поданы логические переменные A и B , тогда на его выходе получим выражение $\overline{A+B}$. Для реализации операции конъюнкции получившееся выражение необходимо инвертировать, что можно реализовать, применив к нему операцию отрицания (второй элемент ИЛИ-НЕ в структурной логической схеме на рис 3.5). Таким образом, мы получили элемент, реализующий операцию конъюнкции (ИЛИ).

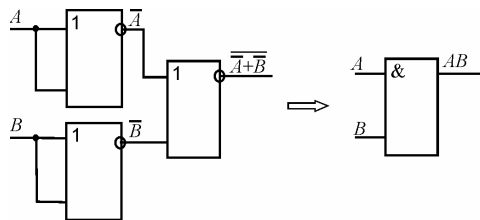


Рис. 3.6. Реализация ЛЭ И на элементах ИЛИ-НЕ

Если на входы структурной схемы, изображенной на рис. 3.6 подать логические переменные A и B , то на выходе мы будем иметь выражение $\overline{\overline{A} \cdot \overline{B}}$. Применяя теорему Де-Моргана (2.9) к этому выражению, получим - $\overline{\overline{A} \cdot \overline{B}} = \overline{\overline{A}} \cdot \overline{\overline{B}}$, далее, последовательно применяя тождество (2.5) к A и B , запишем - $\overline{\overline{A}} \cdot \overline{\overline{B}} = A \cdot B$, т.е. данная структурная схема (рис. 3.6) реализует операцию дизъюнкции (И).

На основе аналогичных рассуждений можно показать выполнение основных логических операций с использованием только элемента И-НЕ (рис. 3.7).

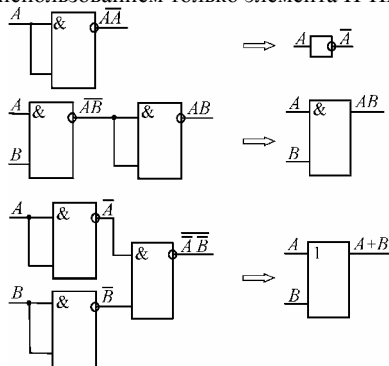


Рис. 3.7. Реализация основных логических операций на базе элементов 2И-НЕ

3.2. Синтез логических схем в заданном базисе логических элементов

При построении логических схем обычно не пользуются функционально полной системой ЛЭ, реализующих все три основные логические операции: И, ИЛИ и НЕ. На практике с целью сокращения номенклатуры элементов используются функционально полной системой элементов, включающей только два элемента, выполняющих операции И-НЕ и ИЛИ-НЕ, или даже только один из этих элементов. Причем число входов этих элементов, как правило, задано. Поэтому вопросы синтеза логических устройств в заданном базисе ЛЭ имеют большое практическое значение.

На основании примеров, рассмотренных в параграфе 3.1, любую ФАЛ можно записать в требуемом базисе ЛЭ. При этом используются два технических приема: двойное инвертирование исходного выражения или его части и применение теорем Де-Моргана.

Если требуется привести ФАЛ к базису ЛЭ И-НЕ, то указанными приемами функция преобразуется к виду, содержащему только операции логического умножения и инверсии. Далее она переписывается через условные обозначения

операции И-НЕ. Аналогично поступают при преобразовании ФАЛ к базису ЛЭ ИЛИ-НЕ. В этом случае в выражении оставляют только операции логического сложения и инверсии. Проиллюстрируем сказанное примером.

Пример 3.2. Заданна ФАЛ $z(x) = x_3x_0 + (\overline{x_3x_2x_0})(\overline{x_2} + x_1)$. Преобразовать к базисам ЛЭ И-НЕ и ИЛИ-НЕ.

Решение. Базис ЛЭ И-НЕ (штрих Шеффера):

$$z(x) = x_3x_0 + \overline{x_3x_2x_0}(\overline{x_2} + x_1) = \overline{\overline{x_3x_0} \cdot \overline{\overline{x_3x_2x_0}(\overline{x_2} + x_1)}} = \overline{x_3x_0 \cdot \overline{x_3x_2x_0}(\overline{x_2} + x_1)} = \overline{x_3x_0 \cdot \overline{x_3x_2x_0} \cdot \overline{\overline{x_2} + x_1}} = \overline{x_3x_0 \cdot \overline{x_3x_2x_0} \cdot (x_2 \downarrow \overline{x_1})}$$

Базис ЛЭ ИЛИ-НЕ (стрелка Пирса):

$$z(x) = \overline{\overline{x_3x_0} \cdot \overline{\overline{x_3x_2x_0}(\overline{x_2} + x_1)}} = \overline{\overline{x_3} + \overline{x_0} + \overline{\overline{x_3x_2x_0} + \overline{x_2} + x_1}} = \overline{\overline{x_3} + \overline{x_0} + x_3 + \overline{x_2} + x_0 + \overline{x_2} + x_1} = \overline{\overline{x_3} \downarrow \overline{x_0} \downarrow x_3 \downarrow \overline{x_2} \downarrow x_0 \downarrow \overline{x_2} \downarrow x_1}$$

Контрольные вопросы и упражнения к разделу 3

1. Приведите условные графические обозначения логических элементов И, ИЛИ и НЕ. Что такое функционально полная система логических элементов?
2. Как строится структурная схема логического устройства по ФАЛ?
3. В чем заключается принцип двойственности и каково его практическое значение для построения схем логических устройств?
4. Что такое функционально полная система и базис логических элементов?
5. Преобразовать ФАЛ вида $z(x) = \overline{x_2}x_1x_0 + x_2\overline{x_1}x_0 + x_2x_1\overline{x_0} + x_2x_1x_0$ к базисам логических элементов Штрих Шеффера и Стрелка Пирса.

4. МИНИМИЗАЦИЯ ФУНКЦИЙ АЛГЕБРЫ ЛОГИКИ

4.1. Цели и общие принципы минимизации функций алгебры логики

В предыдущем разделе было показано, что логическую схему, реализующую заданный алгоритм преобразования сигналов, можно синтезировать непосредственно по выражению, представленному в виде СДНФ или СКНФ. Однако полученная при этом схема, как правило, не оптимальна с точки зрения ее практической реализации. Поэтому исходные ФАЛ обычно минимизируют.

Целью минимизации логической функции является уменьшение стоимости ее технической реализации. Следует отметить, что сам критерий, в соответствии с которым выполняется минимизация ФАЛ, далеко не однозначен и зависит как от типа решаемой задачи, так и уровня развития технологии. Так, в те времена, когда цифровые устройства строились на дискретных элементах, минимизация числа этих элементов и числа построенных на их основе элементарных логических узлов однозначно определяла и уменьшение стоимости технической реализации. С появлением больших и сверх больших интегральных схем (БИС и СБИС), стоимость которых определяется в основном площадью схемы на кристалле и мало зависит от числа входящих в нее транзисторов и других элементов, критерии минимизации ФАЛ претерпели существенные изменения. На первое место при проектировании самих ИС выдвигается требование регулярности их внутренней структуры и минимизация

числа внешних соединений даже за счет увеличения числа элементов и внутренних соединений. Эти требования диктуются требованиями повышения надежности электронных средств.

Однако при проектировании аппаратуры с применением БИС и СБИС требование уменьшения числа корпусов ИС и количества их соединений между собой по-прежнему остается весьма важным.

Требование уменьшения числа элементарных ЛЭ, входящих в разрабатываемое устройство, в настоящее время также не потеряло своей актуальности. Объясняется это все более широким использованием при проектировании электронных средств программируемых логических СБИС широкого применения и полупроводниковых СБИС на основе базовых матричных кристаллов. Эти СБИС и БИС, как правило, содержат отдельные некоммутированные между собой элементарные ЛЭ, например 2И-НЕ или 2ИЛИ-НЕ, или просто наборы транзисторов, резисторов и диодов, которые могут быть соединены между собой в соответствии с заданным алгоритмом обработки логических сигналов. Поскольку число элементов в одной СБИС задано из технологических соображений, то минимизация ФАЛ по критерию уменьшения числа используемых элементов позволяет на одном кристалле решать более сложные задачи логической обработки сигналов, т.е. в конечном счете уменьшать число требуемых ИС и связей между ними. Это снижает стоимость и повышает надежность электронной аппаратуры.

Рассмотрим ряд методов, позволяющих провести минимизацию ФАЛ по критерию уменьшения числа элементарных ЛЭ.

Наиболее просто и наглядно задача минимизации ФАЛ решается с использованием ее кубических представлений. Ранее было показано, что любая логическая функция n -переменных характеризуется своим кубическим комплексом $K(y)$, образованным кубическими комплексами K_0, K_1, \dots, K_{n-1} . Из кубического комплекса $K(y)$ всегда можно выделить множество кубов $P(y)$ таких, что каждый член комплекса K_0 , т.е. вершина куба, будет включен по крайней мере в один куб из множества $P(y)$. Множество кубов $P(y)$ называется *покрытием* комплекса $K(y)$, или покрытием логической функции. Вполне очевидно, что для любой ФАЛ существует несколько ее покрытий. В свою очередь, каждому покрытию $P(y)$, так же как и самому комплексу, соответствует своя дизъюнктивная нормальная форма, получаемая логическим суммированием логических произведений, соответствующих выделенным кубам ФАЛ.

Сложность полученной таким образом ДНФ принято характеризовать понятием «цена покрытия» (C_P), которая равна сумме цен всех кубов, составляющих данное покрытие $P(y)$: $C_P = \sum C_K$. В свою очередь, цена одного r -куба ФАЛ n -переменных определяется как разность полного числа входных переменных и ранга соответствующего куба, т.е. равна числу переменных в соответствующей дизъюнкции: $C_K = n - r$. Так, для ФАЛ трех переменных цена 0-куба равна трем, а 2-куба — единице.

В соответствии со сказанным, задача минимизации ФАЛ сводится к поиску покрытия $P(y)$ кубического комплекса $K(y)$, имеющего минимальную цену.

Покрытие $P(y)$ комплекса $K(y)$, имеющее минимальную цену, называется *покрытием Квайна*, а соответствующая этому покрытию ДНФ — называется *минимальной ДНФ* (МДНФ).

Пример 4.1. Минимизировать ФАЛ заданную в виде словесного описания в примере 2.1. Составить структурную схему логического устройства.

Решение. Как было показано в примере 2.3, СДНФ для данной ФАЛ запишется в виде $y(x_2, x_1, x_0) = \bar{x}_2 x_1 x_0 + x_2 \bar{x}_1 x_0 + x_2 x_1 \bar{x}_0 + x_2 x_1 x_0$. Запишем ее кубический комплекс (рис. 3.1): $K(y) = (011, 101, 110, 111, 11-, 1-1, -11)$.

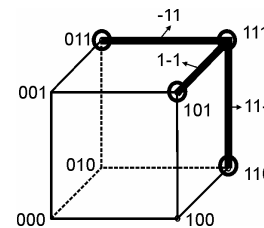


Рис. 4.1. Геометрическое представление кубического комплекса для ФАЛ из примера 3.1

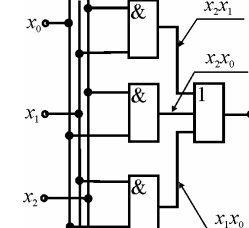


Рис. 4.2. Структурная схема логического устройства, реализующая МДНФ ФАЛ

Нулевой кубический комплекс включает все вершины куба (рис. 4.1), поэтому образует покрытие функции:

$$P_1(y) = K_0 = (011; 101; 110; 111), \text{ найдем цену покрытия } C_{P_1} = 12.$$

Все вершины куба включаются также в единичный кубический комплекс K_1 , поэтому и он образует покрытие ФАЛ:

$$P_2(y) = K_1 = (11-; 1-1; -11), \text{ цена покрытия будет } C_{P_2} = 3.$$

Перебирая сочетания кубов различных рангов, можно получить и другие покрытия ФАЛ, но здесь очевидно, что $P_2(y)$ для данной ФАЛ будет иметь минимальную цену, т.е. является покрытием Квайна. Соответствующая этому покрытию МДНФ запишется в виде: $y_2(x_2, x_1, x_0) = x_2 x_1 + x_2 x_0 + x_1 x_0$.

Анализируя получившееся алгебраическое выражение для ФАЛ, можно сказать, что для реализации этой функции в виде структурной схемы потребуется 3 логических элемента 2И и один элемент 3ИЛИ (рис. 4.2), в отличие от структурной схемы представленной на рис 3.1, не потребуется ЛЭ реализующих операцию НЕ.

Таким образом, полученная в результате минимизации функция и ее структурная схема проще. Техническая реализация такой схемы будет дешевле и надежнее.

4.2. Минимизация функций алгебры-логики с использованием карт Вейча

Данный метод базируется на табличном представлении ФАЛ. Он широко используется при ручной, без применения ЭВМ, минимизации ФАЛ, число переменных в которой обычно не превышает пяти.

Карта Вейча — это прямоугольная таблица, число клеток в которой для ФАЛ n -переменных равно 2^n , каждой из клеток поставлен в соответствие некоторый набор входных переменных, причем рядом расположенным клеткам соответствуют соседние наборы входных переменных (кодов), а в самих клетках записаны значения функций, определенные для этих кодов.

Рассмотрим построение карт Вейча для функций двух, трех и четырех переменных.

Карта Вейча функции двух переменных приведена на рис. 4.3. Она содержит четыре клетки и является плоской фигурой. Для удобства использования по краям

карты указаны значения входных переменных, которые для соответствующих строк и столбцов остаются постоянными. Набор переменных для заданной клетки таблицы определяется как совокупность аргументов, постоянных для строк и столбцов, на пересечении которых она расположена.

	x_1	\bar{x}_1
x_0	$f(x_1, x_0)$	$f(\bar{x}_1, x_0)$
\bar{x}_0	$f(x_1, \bar{x}_0)$	$f(\bar{x}_1, \bar{x}_0)$

Рис. 4.3. Карта Вейча функции двух переменных

Карта Вейча функции трех переменных приведена на рис. 4.4. Она содержит восемь клеток. Наборы входных переменных, соответствующие крайним левому и правому столбцам, являются соседними. Поэтому данную карту удобно представить как поверхность цилиндра и она, в отличие от карты двух переменных, является объемной фигурой.

	x_1	x_1	\bar{x}_1	\bar{x}_1
x_0	$f(\bar{x}_2, x_1, x_0)$	$f(x_2, x_1, x_0)$	$f(x_2, \bar{x}_1, x_0)$	$f(\bar{x}_2, \bar{x}_1, x_0)$
\bar{x}_0	$f(\bar{x}_2, x_1, \bar{x}_0)$	$f(x_2, x_1, \bar{x}_0)$	$f(x_2, \bar{x}_1, \bar{x}_0)$	$f(\bar{x}_2, \bar{x}_1, \bar{x}_0)$
	\bar{x}_2	x_2	x_2	\bar{x}_2

Рис. 4.4. Карта Вейча функции трех переменных

Карта Вейча функции четырех переменных приведена на рис. 4.5. Она содержит 16 клеток. Очевидно, что наборы входных переменных, соответствующие крайним левому и правому столбцам, как и в карте для трех переменных, являются соседними. Кроме этого соседние коды содержатся в нижней и верхней строках карты. Поэтому данная карта тоже является объемной фигурой и может быть представлена как поверхность тора.

	x_1	x_1	\bar{x}_1	\bar{x}_1	
x_0	$f(\bar{x}_3, \bar{x}_2, x_1, x_0)$	$f(x_3, \bar{x}_2, x_1, x_0)$	$f(x_3, \bar{x}_2, \bar{x}_1, x_0)$	$f(\bar{x}_3, \bar{x}_2, \bar{x}_1, x_0)$	\bar{x}_2
x_0	$f(\bar{x}_3, x_2, x_1, x_0)$	$f(x_3, x_2, x_1, x_0)$	$f(x_3, x_2, \bar{x}_1, x_0)$	$f(\bar{x}_3, x_2, \bar{x}_1, x_0)$	x_2
\bar{x}_0	$f(\bar{x}_3, x_2, x_1, \bar{x}_0)$	$f(x_3, x_2, x_1, \bar{x}_0)$	$f(x_3, x_2, \bar{x}_1, \bar{x}_0)$	$f(\bar{x}_3, x_2, \bar{x}_1, \bar{x}_0)$	x_2
\bar{x}_0	$f(\bar{x}_3, \bar{x}_2, x_1, \bar{x}_0)$	$f(x_3, \bar{x}_2, x_1, \bar{x}_0)$	$f(x_3, \bar{x}_2, \bar{x}_1, \bar{x}_0)$	$f(\bar{x}_3, \bar{x}_2, \bar{x}_1, \bar{x}_0)$	\bar{x}_2
	\bar{x}_3	x_3	x_3	\bar{x}_3	

Рис. 4.5. Карта Вейча функции четырех переменных

Еще более сложную форму имеет карта Вейча функции пяти переменных. Ее можно представить как две карты Вейча функции четырех переменных, расположенные одна над другой и отличающиеся лишь значением одной переменной.

Геометрически это можно представить как два тора, один из которых расположен в другом. Соседним кодам тут дополнительно соответствуют клетки, расположенные на разных торах одна под другой. Ввиду сложности работы с такими картами, данный способ редко используется при минимизации ФАЛ пяти переменных.

4.2.1. Минимизация полностью определенной функции алгебры логики

При минимизации ФАЛ используют либо ее нулевые, либо единичные значения. В обоих случаях получают равносильные выражения, которые, однако, могут отличаться по числу членов (т. е. цене) и выполняемым логическим операциям.

Алгоритм минимизации ФАЛ сводится к следующему.

1. На карте Вейча ФАЛ n -переменных выделяют прямоугольные области, объединяющие выбранные значения функции (лог. 0 или лог. 1). Каждая область должна содержать 2^k клеток, где k – целое число. Выделенные области могут пересекаться, т.е. одна или несколько клеток могут включаться в различные области.
2. Каждой из выделенных областей соответствует k -куб исходной ФАЛ, который представляется самостоятельным логическим произведением переменных, значения которых в рамках выделенной области остаются постоянными. Каждое произведение содержит $n-k$ переменных и носит название *импликанты*.
3. Из полученного множества выбирают минимальное число максимально больших областей, включающих все выбранные значения ФАЛ.
4. Логически суммируют импликанты, соответствующие выбранным областям. Полученная сумма образует МДНФ, т.е. является покрытием ФАЛ минимальной стоимости (покрытием Квайна).

При объединении клеток с единичными значениями ФАЛ получают МДНФ самой функции, а при объединении клеток с нулевыми значениями ФАЛ - МДНФ функции, инверсной заданной. Последнее легко объясняется при помощи тождества (2.4).

Очевидно, что если полностью определенная ФАЛ n -переменных принимает значение 1 на m наборах переменных, то на остальных $2^n - m$ входных наборах ее значение равно нулю. Следовательно, объединение 0 значений согласно правилам записи ДНФ приведет к получению функции, инверсной заданной.

Применяя к полученной инверсной минимальной форме теоремы Де-Моргана (2.9), получаем минимальную функцию, записанную в виде КНФ.

Пример. 4.2. Минимизировать ФАЛ

$$z(x) = x_2 x_1 x_0 + x_2 \bar{x}_1 x_0 + x_2 x_1 \bar{x}_0 + x_2 \bar{x}_1 \bar{x}_0.$$

Решение. 1. По заданной ФАЛ составим карту Вейча.

	x_1	x_1	\bar{x}_1	\bar{x}_1
x_0	0	1	1	0
\bar{x}_0	0	1	1	0
	\bar{x}_2	x_2	x_2	\bar{x}_2

2. Запишем кубические комплексы:

$$K_0 = (111, 101, 110, 100),$$

$$K_1 = (1-1, 1-0, 11-, 10-),$$

$$K_2 = (1--).$$

3. Выделим три покрытия и запишем соответствующие им ДНФ:

$$\Pi_1 = (1-1, 1-0) \rightarrow z(x) = x_2 x_0 + x_2 \bar{x}_0$$

$$\Pi_2 = (11-, 10-) \rightarrow z(x) = x_2 x_1 + x_2 \bar{x}_1$$

$$\Pi_3 = (1--) \rightarrow z(x) = x_2$$

Последнее покрытие является покрытием минимальной стоимости и ему соответствует МДНФ. Легко видеть, что этому покрытию соответствует область из четырех клеток, объединяющая единичные значения ФАЛ и расположенная в центре карты Вейча.

При объединении нулевых значений функции максимальная область соответствует 2-кубу (0--), для которого $\bar{z}(x) = \bar{x}_2$ или $z(x) = x_2$. В обоих случаях для минимальной ФАЛ получено одно и то же выражение.

Пример 4.3. Минимизировать ФАЛ

$$z(x) = \bar{x}_2 x_1 x_0 + x_2 x_1 x_0 + x_2 \bar{x}_1 x_0 + x_2 \bar{x}_1 \bar{x}_0 + \bar{x}_2 \bar{x}_1 x_0.$$

Решение. 1. Составим карту Вейча.

	x_1	x_1	\bar{x}_1	\bar{x}_1
x_0	1	1	1	1
\bar{x}_0	0	0	1	0
	\bar{x}_2	x_2	x_2	\bar{x}_2

2. Запишем покрытие минимальной стоимости для единичных значений ФАЛ:

$$\Pi_1 = (-1, 10-) \text{ или } z(x) = x_0 + x_2 \bar{x}_1.$$

3. Запишем покрытие минимальной стоимости для нулевых значений ФАЛ:

$$\Pi_2 = (-10, 0-0) \text{ или } \bar{z}(x) = x_1 \bar{x}_0 + \bar{x}_2 \bar{x}_0.$$

4. Воспользовавшись теоремами Де-Моргана (2.9) и тождеством (2.5), найдем:

$$\bar{\bar{z}}(x) = \overline{x_1 \bar{x}_0 + \bar{x}_2 \bar{x}_0} = \overline{x_1 \bar{x}_0} \cdot \overline{\bar{x}_2 \bar{x}_0} = (\bar{x}_1 + x_0) \cdot (\bar{x}_2 + \bar{x}_1) = (\bar{x}_1 + x_0) \cdot (x_2 + x_0).$$

Согласно второму закону дистрибутивности (2.8) получим ФАЛ вида

$$z(x) = x_0 + x_2 \bar{x}_1.$$

Очевидно, что обе функции $z(x)$ одинаковы.

Из последнего примера видно, что при минимизации по нулевым и единичным значениям функции первоначально можно определить равносильные, но не всегда одинаковые выражения. Различной будет и их техническая реализация. Используя основные тождества и теоремы алгебры логики, их можно преобразовать к единому виду. Однако такое преобразование не всегда очевидно и требует достаточного навыка. Поэтому для нахождения наиболее простого технического решения желательно проводить минимизацию как для нулевых, так и для единичных значений ФАЛ и из полученных значений выбирать простейшее.

Следует еще раз отметить, что так как карты Вейча для ФАЛ трех и четырех переменных являются объемными фигурами, то при формировании областей на таких картах могут объединяться крайние столбцы и строки, а на карте четырех переменных также четыре угловых клетки (рис. 4.5).

4.2.2. Минимизация недоопределенной функции алгебры логики

Напомним, что недоопределенной называется ФАЛ, значения которой заданы не на всех наборах входных переменных. Существуют факультативные (не обязательные) значения функции на неговоренных (запрещенных) входных кодах. Поэтому при минимизации недоопределенной ФАЛ ее факультативные значения доопределяются произвольно из условия получения на карте Вейча наименьшего числа максимально больших областей, что приводит к получению покрытия минимальной стоимости и простейшей технической реализации.

Пример 4.4. Минимизировать ФАЛ, заданную следующей таблицей истинности:

x_2	x_1	x_0	$z(x)$
0	0	0	-
0	0	1	0
0	1	0	1
0	1	1	-
1	0	0	1
1	0	1	-
1	1	0	-
1	1	1	1

Решение. 1. Составим исходную карту Вейча и выполним несколько возможных ее доопределений. Карта исходной функции имеет вид:

	x_1	x_1	\bar{x}_1	\bar{x}_1
x_0	-	1	-	0
\bar{x}_0	1	-	1	-
	\bar{x}_2	x_2	x_2	\bar{x}_2

2. Запишем выражения для единичных значений ФАЛ:

$$z(x) = \bar{x}_2 x_1 \bar{x}_0 + x_2 x_1 x_0 + x_2 \bar{x}_1 \bar{x}_0.$$

Допустим $z(000) = 1$ и $z(110) = 1$, тогда карта Вейча примет вид:

	x_1	x_1	\bar{x}_1	\bar{x}_1
x_0	-	1	-	0
\bar{x}_0	1	1	1	1
	\bar{x}_2	x_2	x_2	\bar{x}_2

$$z(x) = \bar{x}_0 + x_2 x_1$$

Допустим $z(000) = 1$, $z(110) = 1$ и $z(011) = 1$, тогда

	x_1	x_1	\bar{x}_1	\bar{x}_1
x_0	1	1	-	0
\bar{x}_0	1	1	1	1
	\bar{x}_2	x_2	x_2	\bar{x}_2

$z(x) = x_1 + \bar{x}_0$

Допустим $z(000) = 1$, $z(110) = 1$ и $z(101) = 1$, тогда:

	x_1	x_1	\bar{x}_1	\bar{x}_1
x_0	-	1	1	0
\bar{x}_0	1	1	1	1
	\bar{x}_2	x_2	x_2	\bar{x}_2

$z(x) = x_2 + \bar{x}_0$

Допустим $z(011) = 1$, $z(101) = 1$ и $z(110) = 1$, тогда:

	x_1	x_1	\bar{x}_1	\bar{x}_1
x_0	1	1	1	0
\bar{x}_0	1	1	1	-
	\bar{x}_2	x_2	x_2	\bar{x}_2

$z(x) = x_2 + x_1$

3. Минимизация ФАЛ для нулевых значений приводит к выражениям:

$$\bar{z}(x) = \bar{x}_2 \bar{x}_1 x_0 \text{ или } z(x) = x_2 + x_1 + \bar{x}_0.$$

Допустим $z(000) = 0$, тогда:

	x_1	x_1	\bar{x}_1	\bar{x}_1
x_0	-	1	-	0
\bar{x}_0	1	-	1	0
	\bar{x}_2	x_2	x_2	\bar{x}_2

$$\bar{z}(x) = \bar{x}_2 \bar{x}_1 \text{ или согласно теореме Де-Моргана } z(x) = x_2 + x_1.$$

Допустим $z(101) = 0$, тогда:

	x_1	x_1	\bar{x}_1	\bar{x}_1
x_0	-	1	0	0
\bar{x}_0	1	-	1	-
	\bar{x}_2	x_2	x_2	\bar{x}_2

$\bar{z}(x) = \bar{x}_1 x_0$ или $z(x) = x_1 + \bar{x}_0$.

Допустим $z(011) = 0$, тогда:

	x_1	x_1	\bar{x}_1	\bar{x}_1
x_0	0	1	-	0
\bar{x}_0	1	-	1	-
	\bar{x}_2	x_2	x_2	\bar{x}_2

$\bar{z}(x) = \bar{x}_2 x_0$ или $z(x) = x_2 + \bar{x}_0$

Данный пример наглядно иллюстрирует, что при различном доопределении ФАЛ можно получить различные результаты минимизации. Однако любая из полученных ФАЛ, несмотря на их различную техническую реализацию, обеспечивает выполнение алгоритма, заданного исходной таблицей истинности.

4.2.3. Минимизация системы функций алгебры логики

Если логическое устройство имеет N выходов, то его поведение описывается системой, состоящей из N ФАЛ. Минимизация структуры такого устройства может быть выполнена с использованием вышеприведенных методов при отдельной минимизации N структур, на выходе каждой из которых формируется только один выходной сигнал. Однако с точки зрения всего устройства такая структура, как правило, не будет оптимальной.

С точки зрения минимизации всей структуры необходимо, чтобы цепь формирования каждого выходного сигнала была выполнена не минимальным, а некоторым оптимальным способом, обеспечивающим, в конечном счете, минимальность общей структуры устройства. Минимизация в этом случае обеспечивается за счет использования общих целей формирования сигнала для получения нескольких выходных функций. Последнее достигается выделением на картах Вейча различных выходных функций одинаковых областей.

Пример 4.5. Минимизировать структуру устройства, алгоритм работы которого задан следующей таблицей истинности:

x_2	x_1	x_0	z_2	z_1	z_0
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	1	1	1
0	1	1	1	0	0
1	0	0	1	1	1
1	0	1	0	0	1
1	1	0	1	1	1
1	1	1	0	1	0

Решение. Нарисуем карты Вейча для каждой ФАЛ, входящей в заданную систему. Минимизируем данную систему ФАЛ по каждому выходу отдельно по известному алгоритму, как это показано на рис. 4.6.

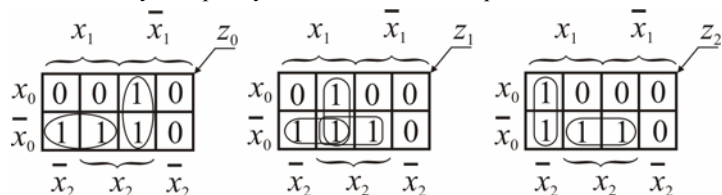


Рис. 4.6. Карты Вейча для системы ФАЛ, пример 4.5

Используя приведенные на рис. 3.6 карты Вейча, для заданной таблице истинности можно записать следующую систему минимальных ФАЛ:

$$z_0 = x_1 \bar{x}_0 + x_2 \bar{x}_1,$$

$$z_1 = x_1 \bar{x}_0 + x_2 \bar{x}_0 + x_2 x_1,$$

$$z_2 = \bar{x}_2 x_1 + x_2 \bar{x}_0.$$

Техническая реализация данной системы потребует семь элементов 2И, два элемента 2ИЛИ и один элемент 3ИЛИ, т.е. всего десять элементов.

Нетрудно заметить, что полученные выражения содержат общие члены $x_1 \bar{x}_0$ и $x_2 \bar{x}_0$. Поэтому техническую реализацию устройства можно упростить. При использовании общих для нескольких элементов выходов для реализации потребуется: пять элементов 2И, два элемента 2ИЛИ и один элемент 3ИЛИ, т.е. всего восемь элементов.

Анализ приведенных карт Вейча показывает, что на входных кодах 010, 100 и 110 все три функции принимают единичное значение. Поэтому можно записать:

$$z_0 = x_1 \bar{x}_0 + x_2 \bar{x}_0 + x_2 \bar{x}_1 = x_0 (x_2 + x_1) + x_2 \bar{x}_1,$$

$$z_1 = x_1 \bar{x}_0 + x_2 \bar{x}_0 + x_2 x_1 = \bar{x}_0 (x_2 + x_1) + x_2 x_1,$$

$$z_2 = x_1 \bar{x}_0 + x_2 \bar{x}_0 + \bar{x}_2 x_1 = \bar{x}_0 (x_2 + x_1) + \bar{x}_2 x_1.$$

Реализация этой схемы потребует четыре элемента 2И и четыре элемента 2ИЛИ, т.е. всего также восемь элементов. Однако из схемы исключен трехвходной

элемент, что, в конечном счете, приводит к упрощению ее технической реализации.

Таким образом, выделение при минимизации системы ФАЛ общих областей на картах Вейча позволят получить наиболее простую ее техническую реализацию. При этом следует иметь в виду, что общие области могут выделяться не на всех картах, а лишь на части из них. Как правило, это приводит к упрощению технической реализации.

3.3. Минимизация функций алгебры-логики на ЭВМ методом Квайна и Мак-Класки

На практике рассмотренные ранее методы минимизации ФАЛ с использованием карт Вейча ограничивается числом переменных не более пяти. Это объясняется двумя основными причинами:

- при увеличении числа переменных метод теряет свою наглядность, что снижает эффективность его применения;
- так как выбор покрытий производится по большей части интуитивно, то конечный результат минимизации сильно зависит от индивидуального опыта разработчика, что препятствует применению для минимизации ФАЛ ЭВМ.

При увеличении числа переменных для минимизации ФАЛ используются методы, обладающие однозначностью алгоритма, что является предпосылкой применения ЭВМ. К таким методам относится метод Квайна и Мак-Класки.

Алгоритм отыскания МДНФ этим методом сводится к следующему.

1. Находят покрытие $\Pi(z)$ заданной функции. Для этого формируют кубический комплекс ФАЛ и в каждом i -ом кубическом комплексе отмечают кубы (импликанты), не образовавшие $i+1$ -й кубический комплекс. Отмеченные импликанты, называемые *простыми*, образуют покрытие заданной ФАЛ.
2. Строят таблицу покрытий матрицы Квайна. Строки указанной таблицы соответствуют простым импликантам, а столбцы – 0-кубам (конституентам единицы) ФАЛ. На пересечении i -й строки и j -го столбца ставится метка, если импликанта i покрывает конституенту j . Отметим, что импликанта i покрывает конституенту j в случае, если она отличается от нее независимыми аргументами.
3. Определяют покрытие минимальной стоимости, для этого:
 - а) выделяют ядро Квайна. Если 0-куб заданной ФАЛ покрывается только одной простой импликантой, то последняя является существенной и входит в ядро Квайна и, следовательно, в покрытие минимальной стоимости;
 - б) из таблицы вычеркивают столбцы и строки, покрытые импликантами ядра Квайна. Если в полученной после вычеркивания таблице содержатся простые импликанты, они также включаются в ядро Квайна с последующим вычеркиванием соответствующих строк и столбцов;
 - в) сжимают таблицу по столбцам, для чего из нее вычеркивают столбцы, в которые полностью входит любой из оставшихся столбцов;
 - г) сжимают таблицу по строкам, для чего из нее вычеркивают строки, которые полностью включаются в любую из оставшихся строк.

Последовательно сжимая таблицу по строкам и столбцам, получают *циклическую таблицу*, импликанты которой должны входить в покрытие ФАЛ минимальной стоимости. На пересечении i -х строк циклической таблицы и импликант, образующих ядро Квайна, получают МДНФ заданной функции.

Проиллюстрируем применение описанного алгоритма на примере.

Пример 4.6. Минимизировать ФАЛ

$$z(x) = \Sigma(0, 1, 2, 4, 5, 7, 8, 10, 12, 14, 15).$$

Решение. Цена покрытия исходной ФАЛ $C_{II} = 44$.

1. Сформируем кубический комплекс $K(z)$. Формирование кубического комплекса $K(z)$ удобно выполнять при помощи разбиения конститuent ФАЛ на группы, содержащие одинаковое количество единиц. При таком представлении кубы более высокого ранга могут образовывать только кубы, находящиеся в расположенных рядом группах. В рассматриваемом примере для ФАЛ четырех переменных можно выделить пять групп, представив их в виде таблицы.

Ранг куба	Номер группы				
	1	2	3	4	5
0-куб	0000	0001 0010 0100 1000	0101 1010 1100	0111 1110	1111
1-куб	000- 00-0 0-00 -000	0-01 -010 010- -100 10-0 1-00	01-1 1-10 11-0	-111 111-	
2-куб	0-0- 0-0- --00 0-0- -0-0 --00	1-0 1-0			

Для заполнения таблицы каждый из кубов левого столбца поочередно сравнивается с кубами правого столбца. Если сравниваемая пара образовала куб более высокого ранга, последний записывается в соответствующий столбец таблицы.

2. Кубы, не образовавшие куб более высокого ранга, являются простыми импликантами и формируют покрытие ФАЛ

$$П(z) = (01-1; -111; 111-; 0-0-; --00; -0-0; 1-0).$$

3. С использованием $П(z)$ построим таблицу покрытий Квайна.

Прост. импл.	0-кубы ФАЛ										
	0000	0001	0010	0100	0101	0111	1000	1010	1100	1110	1111
01-1					*	*					
-111						*					*
111-										*	*
0-0-	*	*		*	*						
--00	*			*			*		*		
-0-0	*		*				*	*			
1-0							*	*	*	*	

4. Согласно полученной таблице простыми импликантами являются 0-0- и -0-0, так как только первая покрывает 0-куб 0001 и только вторая покрывает 0-куб 0010.
5. В оставшейся после вычеркивания существенных импликант и покрытых ими конститuent единицы таблице больше нет существенных импликант. Поэтому произведем сжатие по столбцам и строкам. Первоначальное сжатие по столбцам не выполняется, так как в таблице отсутствуют столбцы входящие в любой из оставшихся. Таблица сжимается по строкам, так как первая строка полностью входит во вторую, а четвертая строка полностью входит в пятую. Поэтому из таблицы вычеркиваем строки с номерами один и четыре.

Простые импликанты	0-кубы ФАЛ			
	0111	1100	1110	1111
01-1	*			
-111	*			*
111-			*	*
-00		*		
1-0		*	*	

Оставшаяся таблица может быть сжата по столбцам, так как первый столбец полностью входит в четвертый, а второй столбец – в третий. На основании этого из таблицы вычеркиваются третий и четвертый столбцы.

Простые импликанты	0-кубы ФАЛ			
	0111	1100	1110	1111
-111	*			*
111-			*	*
1-0		*	*	

Полученная таблица больше не может быть сжата ни по строкам, ни по столбцам. При этом импликанта 111- является лишней, так как она не покрывает ни одну из оставшихся конститuent единицы. Полученная после ее исключения таблица и является циклической.

6. Просуммировав импликанты циклической таблицы и простые импликанты, получим ФАЛ минимальной стоимости.

$$z(x) = \bar{x}_3\bar{x}_1 + \bar{x}_2\bar{x}_0 + x_2x_1x_0 + x_3\bar{x}_0.$$

Алгоритм сжатия по строкам и столбцам можно пояснить следующим образом. Из множества импликант, полученных после исключения существенных, необходимо найти такое их минимальное подмножество, которое обеспечит покрытие всех оставшихся в таблице конститuent единицы. Поэтому, если существует i -я импликанта, которая покрывает множество 0-кубов V_i , которое включает множество V_j , покрываемое импликантой j , то импликанта j является лишней.

Описанный алгоритм без каких-либо изменений позволяет минимизировать ФАЛ любого числа переменных, в том числе и с применением ЭВМ.

Контрольные вопросы и упражнения к разделу 4

1. В чем заключается цель и принципы минимизации логических устройств, реализуемых на БИС и СБИС?
2. Чем характеризуется сложность ДНФ?
3. В чем заключается минимизация ФАЛ с помощью карт Вейча?
4. Представьте карты Вейча функции двух, трех и четырех переменных.
5. К чему сводится алгоритм минимизации ФАЛ? Что такое импликанта и покрытие Квайна?
6. В чем заключается минимизация недоопределенной ФАЛ?
7. В чем заключается минимизация системы ФАЛ?
8. К чему сводится алгоритм минимизации ФАЛ методом Квайна и Мак-Класки?
9. Минимизируйте ФАЛ вида $z(x) = \Sigma(3,5,6,7)$ при помощи карт Вейча.

II. ЦИФРОВЫЕ ЛОГИЧЕСКИЕ УСТРОЙСТВА

Цифровые логические устройства могут быть классифицированы по различным признакам. Так, в общем случае (как это показано на рис. 2.1) на входе логического устройства действуют n , а на выходе – m переменных, т.е. присутствуют соответственно n - и m -разрядные коды. Поэтому логические устройства могут быть классифицированы по способу ввода-вывода переменных (информации). С этой точки зрения они подразделяются на последовательные, параллельные и последовательно-параллельные (смешанные).

Последовательным называется устройство, в котором входные переменные подаются на вход, а выходные переменные снимаются с выхода не одновременно, а последовательно, разряд за разрядом.

Параллельным называется устройство, в котором все разряды входных переменных подаются на вход, и все разряды выходных переменных снимаются с выхода одновременно.

В *последовательно-параллельных* устройствах входные и выходные переменные представлены в различных формах. Либо на вход переменные подаются последовательно символ за символом, а с выхода они снимаются одновременно, либо наоборот.

По принципу действия все логические устройства делятся на два класса: комбинационные и последовательностные.

Комбинационными устройствами, или автоматами без памяти, называют логические устройства, выходные сигналы которых однозначно определяются только действующей в настоящий момент на входе комбинацией переменных и не зависят от значений переменных, действовавших на входе ранее.

Последовательностными устройствами, или автоматами с памятью, называют логические устройства, выходные сигналы которых определяются не только действующей в настоящий момент на входе комбинацией переменных, но и всей последовательностью входных переменных, действовавших в предыдущие моменты времени. Этот тип устройств часто называют цифровыми автоматами.

5. ТИПОВЫЕ ФУНКЦИОНАЛЬНЫЕ УЗЛЫ КОМБИНАЦИОННЫХ ЛОГИЧЕСКИХ УСТРОЙСТВ

5.1. Мультиплексоры и демультиплексоры

Мультиплексором называется комбинационное логическое устройство, предназначенное для управляемой передачи данных от нескольких источников информации в один выходной канал.

Типовое применение мультиплексора — это передача информации от нескольких разнесенных в пространстве источников (датчиков) информации на вход одного приемника. Предположим, измеряется температура окружающей среды в нескольких помещениях и результаты этих измерений должны быть введены в одно регистрирующее устройство, например, ЭВМ. При этом, так как температура изменяется медленно, для получения достаточной точности совсем не обязательно измерять ее постоянно. Достаточно иметь информацию через некоторые фиксированные промежутки времени. Главное при этом, чтобы промежуток между двумя измерениями был существенно меньше постоянной времени, характеризующей изменение температуры в контролируемом помещении. Именно эту функцию, т.е. подключение различных источников информации к одному приемнику по заданной команде, и выполняет мультиплексор. Информацию, разнесенную в пространстве, он преобразует к виду с разделением во времени.

Согласно определению, мультиплексор должен иметь один выход и две группы входов: информационные и адресные. Код, подаваемый на адресные входы, определяет, какой из информационных входов в данный момент подключен к выходному выводу. Поскольку n -разрядный двоичный код может принимать 2^n значений, то если число адресных входов мультиплексора равно n , число его информационных входов должно равняться 2^n .

Мультиплексор обычно снабжается дополнительным инверсным выходом \bar{Q} и входом разрешения работы E . Если на вход разрешения работы E подан активный сигнал ($E = 1$), выходной сигнал мультиплексора постоянен и не зависит от его входных сигналов.

Таблица истинности, отображающая работу мультиплексора с двумя адресными входами, имеет следующий вид (табл. 5.1):

Таблица 5.1

Таблица истинности для мультиплексора с двумя адресными входами

E	A_1	A_0	Q	\bar{Q}
1	-	-	0	1
0	0	0	D_0	\bar{D}_0
0	0	1	D_1	\bar{D}_1
0	1	0	D_2	\bar{D}_2
0	1	1	D_3	\bar{D}_3

Функция алгебры логики, описывающая работу мультиплексора, имеет вид

$$Q = D_0 \bar{A}_1 \bar{A}_0 \bar{E} + D_1 \bar{A}_1 A_0 \bar{E} + D_2 A_1 \bar{A}_0 \bar{E} + D_3 A_1 A_0 \bar{E} \quad (5.1)$$

Логическая схема мультиплексора, соответствующая этой ФАЛ, и условное графическое обозначение мультиплексора на примере интегральной схемы (ИС) типа 555КП7 показаны на рис. 5.1 (а, б).

При передаче информации от нескольких источников по общему каналу с разделением по времени нужны не только мультиплексоры, но и устройства обратного назначения, распределяющие информацию, полученную из одного канала между несколькими приемниками. Эту задачу решают демultipлексоры.

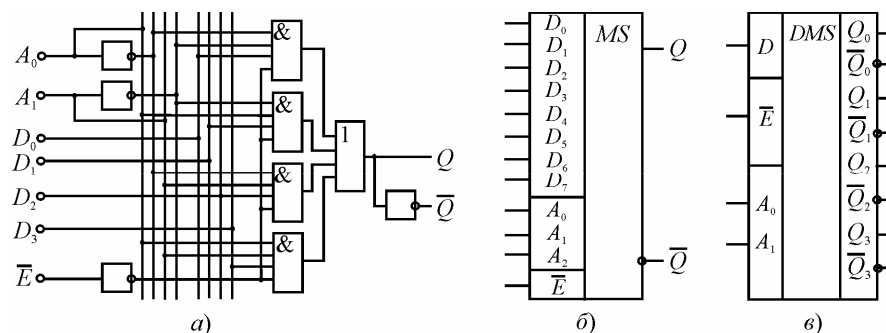


Рис. 5.1. Логическая схема мультиплексора (а) и его условное графическое обозначение (б); условное графическое изображение демultipлексора (в)

Демultipлексором называется комбинационное логическое устройство, предназначенное для управляемой передачи данных от одного источника информации в несколько выходных каналов.

Согласно данному определению, демultipлексор в общем случае имеет один информационный вход, n адресных входов и 2^n выходов. Таблица истинности, описывающая работу демultipлексора, снабженного двумя адресными входами и входом разрешения работы E , имеет вид

Таблица 5.2

Таблица истинности демultipлексора с двумя адресными входами

E	A_1	A_0	Q_0	Q_1	Q_2	Q_3
1	-	-	0	0	0	0
0	0	0	D	0	0	0
0	0	1	0	D	0	0
0	1	0	0	0	D	0
0	1	1	0	0	0	D

Данной таблице соответствует следующая система ФАЛ:

$$\begin{cases} Q_0 = D \bar{A}_1 \bar{A}_0 \bar{E} = \bar{D} \downarrow A_1 \downarrow A_0 \downarrow E, \\ Q_1 = D \bar{A}_1 A_0 \bar{E} = \bar{D} \downarrow A_1 \downarrow \bar{A}_0 \downarrow E, \\ Q_2 = D A_1 \bar{A}_0 \bar{E} = \bar{D} \downarrow \bar{A}_1 \downarrow A_0 \downarrow E, \\ Q_3 = D A_1 A_0 \bar{E} = \bar{D} \downarrow \bar{A}_1 \downarrow \bar{A}_0 \downarrow E. \end{cases} \quad (4.2)$$

Пользуясь системой уравнений (4.2), можно синтезировать логическую схему демultipлексора. На рис. 4.1 (в) показано условное графическое обозначение демultipлексора.

5.2. Преобразователи кодов

В цифровой технике применяются различные виды кодирования информации. Так, при выполнении операций в ЭВМ обычно применяют несколько разновидностей двоичного кода (прямой, обратный, дополнительный, двоично-десятичный и т.д.). При передаче информации по линиям связи удобно использовать другие виды кодов, позволяющие, например, уменьшать вероятность появления ошибки или даже

исправлять их в дальнейшем. Примерами таких кодов являются коды, построенные по принципу 2 из 5, коды с проверкой четности или нечетности, коды Хемминга. Некоторые виды этих кодов были рассмотрены в первом разделе.

В связи с этим всегда стоит задача преобразования информации одного кода в другой. Эту задачу на аппаратном уровне решают комбинационные устройства - преобразователи кодов.

Преобразователем кода называется комбинационное устройство, предназначенное для изменения вида кодирования информации.

Как и всякое комбинационное устройство, преобразователь кода характеризуется таблицей истинности, ставящей в соответствие кодам, подаваемым на вход, коды, снимаемые с выхода устройства. Следует заметить, что в этой таблице в общем случае число разрядов входного и выходного кодов может не совпадать. Главное - она должна давать однозначное соответствие различных кодов. Данная таблица является основанием для синтеза логической структуры конкретного преобразователя кодов. В качестве примера преобразователя кодов, выпускаемых в виде ИС, можно привести схемы, обеспечивающие преобразование информации из двоичного в двоично-десятичный код. Частным случаем преобразователей кода являются шифраторы и дешифраторы.

Шифраторы и дешифраторы. *Шифратором*, или кодером, называется комбинационное логическое устройство для преобразования чисел из десятичной системы счисления в двоичную. Входам шифратора последовательно присваиваются значения десятичных чисел, поэтому подача активного логического сигнала на один из входов воспринимается шифратором как подача соответствующего десятичного числа. Этот сигнал преобразуется на выходе шифратора в двоичный код. Согласно сказанному, если шифратор имеет n выходов, число его входов должно быть не более чем 2^n . Шифратор, имеющий 2^n входов и n выходов, называется *полным*. Если число входов шифратора меньше 2^n , он называется *неполным*.

Рассмотрим работу шифратора на примере преобразователя десятичных чисел от 0 до 9 в двоично-десятичный код. Таблица истинности, соответствующая этому случаю, имеет вид (табл. 5.3)

Таблица 5.3

x_9	x_8	x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	0	1	0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	0	0	1	0	1
0	0	0	1	0	0	0	0	0	0	0	1	1	0
0	0	1	0	0	0	0	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	0	1	0	0	1

Так как число входов данного устройства меньше $2^n = 16$, имеем неполный шифратор. Используя таблицу истинности (табл. 5.3) для Q_3, Q_2, Q_1 и Q_0 запишем следующую систему ФАЛ:

$$\begin{cases} Q_3 = x_8 + x_9, \\ Q_2 = x_4 + x_5 + x_6 + x_7, \\ Q_1 = x_2 + x_3 + x_6 + x_7, \\ Q_0 = x_1 + x_3 + x_5 + x_7 + x_9. \end{cases} \quad (5.3)$$

Полученная система ФАЛ характеризует работу шифратора. Логическая схема устройства, синтезированная согласно системе уравнений (5.3), приведена на рис. 5.2 (а).

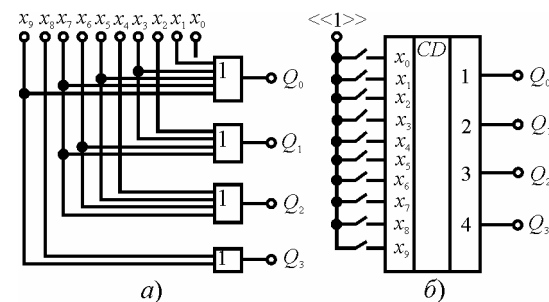


Рис. 5.2. Логическая схема шифратора десятичных чисел (а); устройство ввода информации с клавиатуры (б)

Нетрудно заметить, что в шифраторе рассматриваемого типа подаваемый на вход x_0 сигнал, не используется. Поэтому отсутствие сигнала на любом из входов x_0, \dots, x_9 трактуется схемой как наличие на входе нулевого сигнала.

Основное применение шифратора в цифровых системах — это введение первичной информации с клавиатуры. При нажатии любой клавиши на соответствующий вход шифратора подается сигнал «логическая 1», который и преобразуется на выходе в двоично-десятичный код. Вариант устройства ввода информации показан на рис 5.2. (б).

Дешифратором, или декодером, называется комбинационное логическое устройство для преобразования чисел из двоичной системы счисления в десятичную. Согласно определению дешифратор относится к классу преобразователей кодов. Здесь также понимается, что каждому входному двоичному числу ставится в соответствие сигнал, формируемый на определенном выходе устройства. Таким образом, дешифратор выполняет операцию, обратную шифратору. Если число адресных входов дешифратора n связано с числом его выходов m соотношением $m=2^n$, то дешифратор называют *полным*. В противном случае, т. е. если $m < 2^n$, дешифратор называют *неполным*.

Поведение дешифратора описывается таблицей истинности аналогичной таблице истинности шифратора (табл. 5.3), только в ней входные и выходные сигналы меняются местами. В соответствии с данной таблицей, так как выходной сигнал равен 1 только на одном единственном наборе входных переменных, т.е. для одной конституенты единицы, алгоритм работы дешифратора описывается системой уравнений вида

$$\begin{cases} x_0 = \overline{Q_3}\overline{Q_2}\overline{Q_1}\overline{Q_0}, \\ x_1 = \overline{Q_3}\overline{Q_2}\overline{Q_1}Q_0, \\ x_2 = \overline{Q_3}\overline{Q_2}Q_1\overline{Q_0}, \\ \dots \end{cases} \quad (5.4)$$

и так далее, где Q_i – значение логической переменной на i -ом входе устройства.

В общем случае эта система имеет вид:

$$x_i = (Q_3Q_2Q_1Q_0)_i, \quad (5.5)$$

где x_i – сигнал на i -м выходе дешифратора; $(Q_3Q_2Q_1Q_0)_i$ – конституента единицы, соответствующая двоичному коду i -ой десятичной цифры.

Нетрудно заметить, что ФАЛ дешифратора (5.4) отличается от ФАЛ мультиплексора (5.2) только наличием в последней дополнительного множителя, соответствующего значению сигнала на информационном входе D . Поэтому при $D=1$ демультиплексор функционирует как дешифратор. Обратное преобразование дешифратора в демультиплексор требует введения вспомогательных ЛЭ 2И, выполняющих операцию логического умножения между общим сигналом информационного входа D и соответствующим логическим произведением адресных сигналов $(Q_3Q_2Q_1Q_0)$.

Микросхемы дешифраторов часто снабжаются входом разрешения работы E (входом стробирования). Наличие этого входа позволяет на основе готовых ИС при необходимости увеличения разрядности входного кода строить структуры дешифраторного дерева.

5.3. Цифровые компараторы

Цифровым компаратором называется комбинационное логическое устройство, предназначенное для сравнения чисел, представленных в виде двоичных кодов.

Число входов компаратора определяется разрядностью сравниваемых кодов. На выходе компаратора обычно формируется три сигнала:

- $F_=$ - равенство кодов,
- $F_<$ - если числовой эквивалент первого кода больше второго,
- $F_>$ - если числовой эквивалент первого кода меньше второго.

Работу компаратора при сравнении двух одноразрядных кодов поясняет таблица истинности (табл. 5.4).

Таблица 5.4

Таблица истинности компаратора одноразрядных кодов

x_1	x_0	$F_=$	$F_>$	$F_<$
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

Анализ таблицы истинности показывает, что при любой комбинации входных сигналов на выходе компаратора может быть сформирован только один активный (единичный) логический сигнал. Поэтому, при любой разрядности входных кодов достаточно, используя входные сигналы, сформировать только любые два из выходных сигналов. Третий сигнал всегда может быть получен по двум известным.

Система ФАЛ, соответствующая приведенной таблице истинности, имеет вид:

$$\begin{aligned} F_&= \overline{x_1}\overline{x_0} + x_1x_0 = x_1 \oplus x_0 = \overline{F_<} \overline{F_>}, \\ F_&< = \overline{x_1}x_0 = \overline{F_=} \overline{F_>}, \\ F_&> = x_1\overline{x_0} = \overline{F_=} F_>. \end{aligned} \quad (5.6)$$

Анализируя приведенные выражения с точки зрения минимизации системы ФАЛ, отметим, что, используя выходные переменные, удобнее было бы получить значения $F_>$ и $F_<$, а $F_=$ - реализовать как их функцию. Однако ввиду того, что выражение для определения $F_=$ имеет в цифровой технике большое самостоятельное значение, на нем следует остановиться более подробно. Оно носит название операции исключающее ИЛИ-НЕ или инверсия от суммы по модулю два (см. Ч. I, табл. 2.3). Пример реализации этой операции с использованием элементов И, ИЛИ и НЕ, а также её условное графическое обозначение приведены на рис. 5.3. На рис. 5.4 дана структурная логическая схема, соответствующая таблице истинности цифрового компаратора.

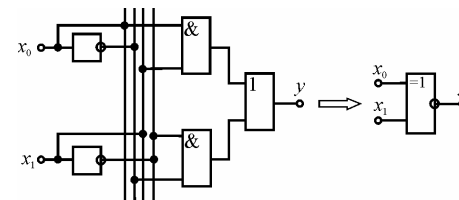


Рис. 5.3. Схемная реализация операции Исключающее ИЛИ-НЕ

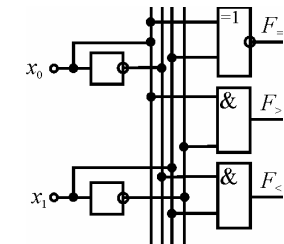


Рис. 5.4. Логическая схема одноразрядного компаратора.

На практике часто приходится сталкиваться с задачей сравнения много разрядных двоичных кодов. Записав таблицу истинности, можно синтезировать логическую схему соответствующего устройства. Однако при увеличении числа входных переменных сделать это ввиду громоздкости получаемой таблицы весьма непросто. В данной ситуации удобно воспользоваться методами так называемого блочного конструирования, или декомпозиции задачи. Суть такого подхода состоит в разбиении сложной задачи на ряд более простых, решение которых может быть выполнено доступными средствами. Далее, с использованием полученных результатов, производится решение исходной задачи. Проиллюстрируем данный подход на примере построения цифрового компаратора многоразрядных двоичных кодов. За основу примем схему компаратора одноразрядных двоичных слов. Очевидно, что результат сравнения 2-разрядных двоичных слов можно записать через результаты сравнения одноразрядных слов. Соответствующая система ФАЛ в этом случае будет иметь вид:

$$\begin{aligned} F_&= = F_{1=} F_{0=}, \\ F_&> = F_{1>} + F_{1=} F_{0=}, \\ F_&< = \overline{F_=} + \overline{F_>}. \end{aligned}$$

Аналогично для трех разрядных кодов получим:

$$F_{=} = F_{2=}F_{1=}F_{0=},$$

$$F_{>} = F_{2=} + F_{2=}F_{1>} + F_{2=}F_{1=}F_{0>},$$

$$F_{<} = \overline{F_{=} + F_{>}}.$$

В общем случае для n-разрядных двоичных кодов можно записать:

$$F_{=} = F_{n-1=}F_{n-2=} \dots F_{0=},$$

$$F_{>} = F_{n-1>} + F_{n-1=}F_{n-2>} + F_{n-1=}F_{n-2=} \dots F_{1=}F_{0>},$$

$$F_{<} = \overline{F_{=} + F_{>}}.$$

Таким образом, с использованием цифровых компараторов, имеющих ограниченную разрядность входных слов, на основании системы ФАЛ (5.7) всегда можно построить устройство требуемой разрядности.

5.4. Сумматоры

Сумматором называется комбинационное логическое устройство, предназначенное для выполнения операции арифметического сложения чисел, представленных в виде двоичных кодов.

Термин сумматор охватывает широкий спектр устройств, начиная с простейших логических схем, до сложнейших цифровых узлов. Общим для всех этих устройств является арифметическое сложение чисел, представленных в двоичной форме. Рассмотрим более подробно некоторые конкретные схемотехнические решения, предназначенные для реализации поставленной задачи.

Алгоритм двоичного сложения

Для начала получим ФАЛ, описывающие операции арифметического сложения двух одноразрядных двоичных кодов x_1 и x_0 . Алгоритм ее выполнения поясняется таблицей истинности (табл. 5.5). В графе s приведено значение результата сложения (суммы), а в графе p - полученное при этом значение переноса в старший разряд. Следует обратить внимание на отличия результатов, получаемых при арифметическом и логическом сложениях. При логическом сложении в последней строке столбца s присутствовало бы значение 1. Это отличие результатов данных операций не позволяет применить для арифметического суммирования элемент ИЛИ, а требует разработки специализированного устройства.

Значение сигнала переноса, равного единице в последней строке таблицы 5.5, говорит о том, что результат, полученный при выполнении операции арифметического сложения, в этом случае не может быть представлен двоичным кодом, разрядность которого равна разрядности слов слагаемых. Для представления результата необходимо слово, имеющее на один разряд больше, чем коды слагаемых.

Таблица 5.5.

Таблица истинности сложения двух одноразрядных двоичных кодов

x_1	x_0	s	p
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Используя приведенную таблицу, легко записать систему ФАЛ, описывающих алгоритм операции арифметического сложения

$$s = \bar{x}_1x_0 + x_1\bar{x}_0 = x_1 \oplus x_0, \quad (5.8)$$

$$p = x_1x_0. \quad (5.9)$$

Функция, описываемая выражением (5.8), очень часто встречается при разработке цифровых устройств. Ее называют функцией Исключающее ИЛИ (см. Ч. I, табл. 2.3), или *суммой по модулю два*. Таким образом, для суммирования двух двоичных одноразрядных кодов необходимо выполнить логическую операцию Исключающее ИЛИ. Обратная ей функция Исключающее ИЛИ-НЕ определяется как (см. Ч. I, табл. 2.3):

$$\bar{s} = \overline{x_1 \oplus x_0} = \bar{x}_1\bar{x}_0 + x_1x_0. \quad (5.10)$$

Логические элементы, выполняющие операции Исключающее ИЛИ и Исключающее ИЛИ-НЕ, всегда имеют только два входа, т. е. операции всегда выполняются только над двумя переменными.

Таблица 5.6

Таблица истинности сложения разрядов многоразрядных двоичных кодов

x_1	x_0	p_{-1}	s	p
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

Таблица 5.5 применима только для сложения одноразрядных двоичных кодов или младших разрядов многоразрядных слов. Таблица сложения старших разрядов многоразрядных двоичных слов должна быть дополнена переменной возможного переноса из более младшего разряда (табл. 5.6). ФАЛ, описывающие результаты сложений, в этом случае будут иметь вид

$$s = (x_1 \oplus x_0)\bar{p}_{-1} + \overline{(x_1 \oplus x_0)}p_{-1}, \quad (5.11)$$

$$p = x_1x_0 + (x_1 \oplus x_0)p_{-1}. \quad (5.12)$$

Из выражения (5.11) следует, что для получения суммы двух разрядов необходимо сначала выполнить операцию Исключающее ИЛИ над исходными слагаемыми x_1 и x_0 и затем еще одну операцию Исключающее ИЛИ над результатом первой операции Исключающее ИЛИ и сигналом переноса из предыдущего разряда. Для получения сигнала переноса также необходимо воспользоваться результатом операции Исключающее ИЛИ над слагаемыми x_1 и x_0 .

Классификация сумматоров

Классификация сумматоров может быть выполнена по различным признакам. Рассмотрим наиболее часто встречающиеся из них.

По числу выводов различают: полусумматоры, одноразрядные сумматоры, многоразрядные сумматоры.

- *Полусумматором* называется устройство, предназначенное для сложения двух одноразрядных кодов, имеющее два входа и два выхода и формирующее из сигналов входных слагаемых сигналы суммы и переноса в старший разряд.
- *Одноразрядным сумматором* называется устройство, предназначенное для сложения двух одноразрядных кодов, имеющее три входа и два выхода, и формирующее из сигналов входных слагаемых и сигнала переноса из младших разрядов сигналы суммы и переноса в старший разряд.
- *Многоразрядным сумматором* называется устройство, предназначенное для сложения двух многоразрядных кодов, формирующее на выходе код суммы и сигнал переноса в случае, если результат сложения не может быть представлен кодом, разрядность которого совпадает с разрядностью кодов слагаемых. В свою очередь, многоразрядные сумматоры подразделяются на *последовательные* и *параллельные*. В последовательных сумматорах операция сложения выполняется последовательно разряд за разрядом, начиная с младшего. В параллельных все разряды входных кодов суммируются одновременно.

Различают *комбинационные сумматоры* - устройства, не имеющие собственной памяти, и *накапливающие сумматоры*, снабженные собственной внутренней памятью, в которой аккумулируются результаты выполненной операции. При этом каждое очередное слагаемое прибавляется к уже имевшемуся в устройстве значению.

По способу тактирования различают *синхронные* и *асинхронные* сумматоры. В *синхронных* сумматорах время выполнения операции арифметического суммирования двух кодов не зависит от вида самих кодов и всегда остается постоянным. В *асинхронных* сумматорах время выполнения операции зависит от вида слагаемых. Поэтому по завершении выполнения суммирования необходимо вырабатывать специальный сигнал завершения операции.

В зависимости от используемой системы счисления различают двоичные, двоично-десятичные и другие типы сумматоров.

Двоичный полусумматор

Согласно определению, выходные сигналы двоичного полусумматора должны соответствовать системе ФАЛ (5.8) и (5.9). Для ее технической реализации необходимы логические элементы И и Исключающее ИЛИ. Так как ранее элемент Исключающее ИЛИ не был описан, рассмотрим возможность его построения на уже известных элементах. Для этого преобразуем выражение (5.5) к базису И-НЕ воспользовавшись техническими приемами, описанными в параграфе 3.2 части I

$$s = x_1 \oplus x_0 = \bar{x}_1 x_0 + x_1 \bar{x}_0 = \overline{\bar{x}_1 x_0} + \overline{x_1 \bar{x}_0} = \overline{\bar{x}_1 x_0} \cdot \overline{x_1 \bar{x}_0} = (\bar{x}_1 | x_0) \cdot (x_1 | \bar{x}_0) = (\bar{x}_1 | x_0) | (x_1 | \bar{x}_0).$$

Техническая реализация полученного выражения приведена рис. 5.5. На этом же рисунке показано условное графическое обозначение элемента Исключающее ИЛИ.

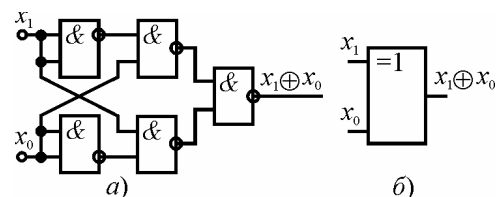


Рис. 5.5. Структурная схема реализации операции Исключающее ИЛИ (а) и ее условное обозначение (б)

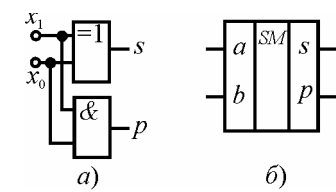


Рис. 5.6. Полусумматор (а) и его условное обозначение (б)

С использованием сказанного легко можно синтезировать логическую схему двоичного полусумматора (рис. 5.6 (а)). Условное графическое обозначение двоичного полусумматора показано на рис. 5.6 (б).

Одноразрядный сумматор

Функционирование одноразрядного сумматора определяется системой ФАЛ (5.11) и (5.12). Техническая реализация данной ФАЛ может быть выполнена на ЛЭ любого типа. Рассмотрим, например построение одноразрядного сумматора с использованием схем двоичных полусумматоров (рис. 5.7 (а)). Очевидно, что для этой цели необходимо два полусумматора и элемент ИЛИ.

Следует отметить, что если синтезировать схему одноразрядного сумматора непосредственно по табл. 5.6 относительно элементарных ЛЭ, можно получить более простое техническое решение.

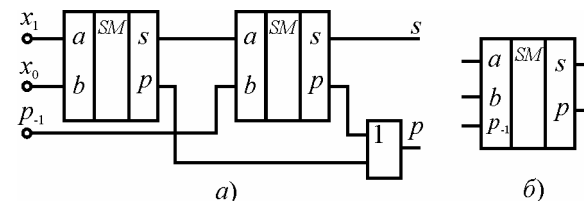


Рис. 5.7. Одноразрядный сумматор (а) и его условное обозначение (б)

Условное графическое обозначение одноразрядного сумматора приведено на рис. 5.7 (б).

Многоразрядный сумматор параллельного действия

В этом сумматоре согласно данному ранее определению операции суммирования должны выполняться одновременно по всем разрядам исходных двоичных чисел. Из этого следует, что такой сумматор должен иметь отдельные аппаратные средства для выполнения суммирования в каждом разряде.

Рассмотрим приведенную на рис. 5.8 типовую структуру 4-х разрядного сумматора, выполненного с использованием трех одноразрядных сумматоров и одного полусумматора. Разряды кодов слагаемых подаются на соответствующие входы сумматоров, выходы суммы которых подсоединяются к первым входам ЛЭ И, используемых в качестве выходных ключей, на вторые входы которых подается сигнал Z, определяющий момент считывания результата. Выход сигнала переноса сумматора нулевого разряда подается вход переноса сумматора первого разряда и т.д.

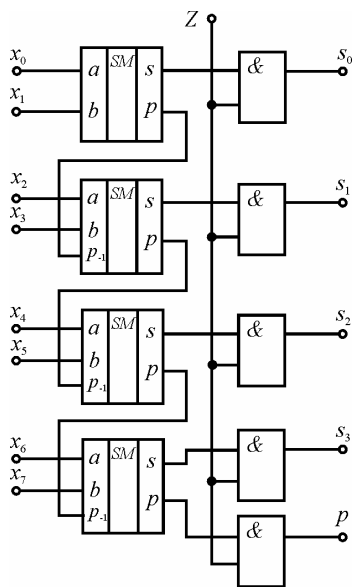


Рис. 5.8. Структурная схема параллельного многоразрядного сумматора с последовательным переносом

Из сказанного следует, что для получения на выходе сигнала, равного реальной сумме входных кодов, необходимо, чтобы сигнал переноса последовательно сформировался на выходах сумматоров всех разрядов. Следовательно, не зависимо от того, что для суммирования в каждом разряде используется отдельный сумматор, реальное время выполнения операции в данной схеме определяется последовательным переносом сигнала P из разряда в разряд. Поэтому результат, который может быть снят с выхода схемы через время, равное времени суммирования в одном разряде, не будет являться реальным значением искомой суммы.

Для исключения получения ложного результата на выходе схемы установлены элементы И. Сигнал Z на входах этих элементов должен появляться не ранее, чем после последовательной передачи сигнала переноса по всем разрядам сумматоров.

Следует отметить, что реально схемы многоразрядных сумматоров строятся только с применением одноразрядных сумматоров, что позволяет, используя их последовательное включение, увеличивать разрядность кодов слагаемых.

5.5. Логические элементы, реализующие сложные функции

При разработке сложных логических устройств часто приходится сталкиваться с последовательным выполнением однотипных операций типа И-ИЛИ-НЕ над различным числом переменных. Для упрощения технической реализации и уменьшения числа внешних соединений, что ведет к повышению надежности устройства, в серийно выпускаемых сериях ИС содержатся элементы, выполняющие подобные последовательности логических операций. В качестве примера на рис. 4.5 приведены структурная схема и условное графическое обозначение ИС типа 555ЛР2. Эта схема выполняет операцию 2-2-3-4-И-4ИЛИ-НЕ.

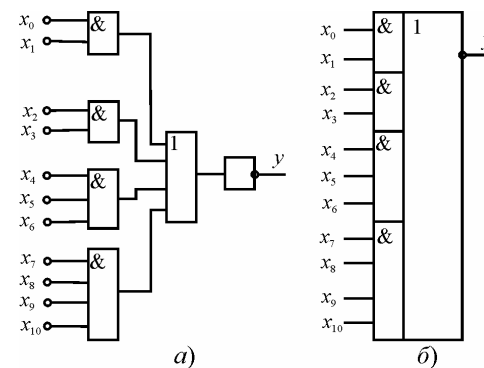


Рис. 5.9. Логическая схема 2-2-3-4-И-4ИЛИ-НЕ (а) и ее условное графическое обозначение (б)

Контрольные вопросы и упражнения к разделу 5

1. Каковы назначение и структурная схема мультиплексора?
2. Каково назначение демультиплексора? Составьте структурную схему демультиплексора.
3. Каково назначение преобразователя кодов?
4. Каковы назначение и логическая схема шифратора?
5. Приведите условную схему устройства ввода информации и клавиатуры.
6. Приведите схему двоично-десятичного дешифратора.
7. Каковы назначение и логическая схема цифрового компаратора?
8. Запишите ФАЛ, реализующие арифметическое суммирование одноразрядных двоичных кодов.
9. Составьте таблицу истинности для элемента Исключающие ИЛИ.
10. Составьте таблицу истинности для элемента Исключающие ИЛИ-НЕ.
11. Чем отличаются полусумматор и одноразрядный сумматор?
12. В чем основное отличие многоразрядных сумматоров параллельного и последовательного действий?
13. Составьте схему ЛЭ, реализующую сложную функцию четырнадцати переменных вида 2-3-4-5-И-4ИЛИ-НЕ.

6. ТРИГГЕРНЫЕ УСТРОЙСТВА

Особенностью последовательных логических устройств является зависимость выходного сигнала не только от действующих в настоящий момент времени на входе логических переменных, но и от тех значений переменных, которые действовали на входе в предыдущие моменты времени. Очевидно, что для выполнения этого условия значения переменных должны быть запомнены логическим устройством. Функцию запоминания значений логических переменных в цифровых схемах выполняют так называемые *триггеры*. Таким образом, триггерный элемент, выполняя функции памяти, является неотъемлемой частью любого последовательностного устройства.

Рассмотрим основные принципы построения и функционирования наиболее распространенных триггерных элементов.

Триггером называется устройство, способное формировать два устойчивых значения выходного сигнала и скачкообразно изменять эти значения под действием внешнего управляющего сигнала. Именно способность формировать на выходе два

устойчивых значения сигнала, которые могут поддерживаться без изменения сколь угодно длительный промежуток времени, и позволяет применять триггер в качестве элемента памяти.

В общем случае триггер содержит собственно элемент памяти и некоторую входную комбинационную схему, преобразующую входные сигналы триггера в сигналы, требуемые для управления элементом памяти. Простейший триггер состоит только из элемента памяти, который может быть реализован на основе усилителя, охваченного глубокой положительной обратной связью (ПОС). В частности, роль триггера может выполнять операционный усилитель с цепью положительной обратной связи. На рис. 6.1 приведены принципиальная схема и передаточная характеристика гистерезисного компаратора. Очевидно, что если на его входе действует напряжение $-U_{\text{вых}m} R_2 / (R_1 + R_2) < U_{\text{вх}} < +U_{\text{вых}m} R_2 / (R_1 + R_2)$, выходной сигнал компаратора неизменен и определяется только предыдущим значением входного сигнала, удовлетворяющего неравенству

$$U_{\text{вх}} > \left| U_{\text{макс}m} \right| R_2 / (R_1 + R_2).$$

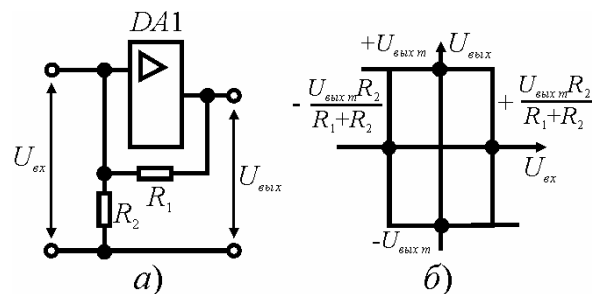


Рис. 6.1. Принципиальная схема (а) и передаточная характеристика (б) гистерезисного компаратора

В качестве базовых узлов, на основе которых может быть построен триггер, могут использоваться и элементы И-НЕ или ИЛИ-НЕ, передаточная характеристика которых подобна характеристике инвертирующего усилителя. Так как для построения триггера исходный усилитель должен быть охвачен цепью ПОС, необходимо взять два элемента. В этом случае простейший триггер, включающий только непосредственно элемент памяти, имеет вид, показанный на рис. 6.2. Такой элемент содержит два информационных входа, на которые подаются входные переменные x_1 и x_0 , и два выхода, с которых снимаются сигналы y_1 и y_0 , причем, если

$$\begin{aligned} y_0 = 1, & \text{ то } y_1 = 0, \\ y_0 = 0, & \text{ то } y_1 = 1. \end{aligned} \quad (6.1)$$

Выражения (6.1) называются постулатом триггера.

При описании триггера также пользуются понятием активного логического уровня. Для изменения выходного сигнала триггера, или, как принято говорить, для изменения его *состояния*, на вход схемы рис. 6.2 (а), выполненной на элементах И-НЕ, следует подать сигнал $x = 0$. Для переключения триггера (рис. 6.2 (б)), выполненного на элементах ИЛИ-НЕ, требуется сигнал $x = 1$. Для обоих триггеров (рис. 6.2), если $x_0 = 0$, а $x_1 = 1$, то $y_0 = 1$ и $y_1 = 0$. Однако в триггере на рис. 6.2 (а) активным определяющим сигналом будет $x_0 = 0$, а в триггере на рис. 6.2 (б) — $x_1 = 1$.

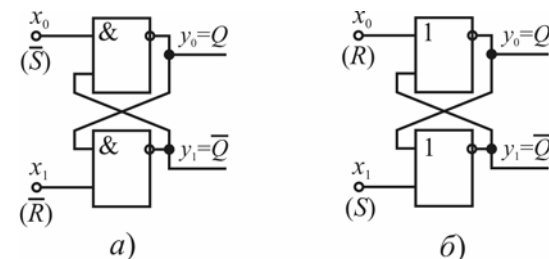


Рис. 6.2. Схемы простейших триггеров на элементах 2И-НЕ (а) и 2ИЛИ-НЕ (б)

При описании работы триггера приняты следующие соглашения:

- если $Q = 1$, а $\bar{Q} = 0$, то триггер находится в состоянии *установки*, или просто *установлен*;
- если $Q = 0$, а $\bar{Q} = 1$, то триггер находится в *сброшенном состоянии*, или просто *сброшен*.

Существующие типы триггеров могут быть классифицированы по различным признакам. Наиболее часто триггеры классифицируют по типу используемых информационных входов. Различают следующие типы основных информационных входов триггера:

- R – раздельный вход сброса триггера ($Q = 0$);
- S – раздельный вход установки триггера ($Q = 1$);
- K – вход сброса универсального триггера ($Q = 0$);
- J – вход установки универсального триггера ($Q = 1$);
- T – счетный вход триггера;
- D – информационный вход переключения триггера в состояние, соответствующее логическому уровню на этом входе;
- C – управляющий или синхронизирующий вход.

Кроме этих основных входов некоторые триггеры могут снабжаться входом V . Вход V блокирует работу триггера, и он сколь угодно долго может сохранять ранее записанную в него информацию. С точки зрения типа используемых входов различают RS -, D -, T -, JK -, VD - и VT - триггеры.

По моменту реакции на входной сигнал триггеры подразделяют на *асинхронные* и *синхронные*.

Асинхронный триггер изменяет свое состояние непосредственно в момент изменения сигнала на его информационных входах, т.е. его непосредственная реакция на изменение входного сигнала подобна реакции комбинационного элемента.

Синхронный триггер изменяет свое состояние лишь в строго определенные (тактовые) моменты времени, соответствующие действию активного сигнала на его синхронизирующем входе C , и не реагирует на любые изменения информационных сигналов при пассивном значении сигнала на входе C .

По виду активного логического сигнала, действующего на информационных входах, триггеры различают на *статические* - управляемые уровнем, и *динамические* - управляемые перепадом входного сигнала. При этом сами входы могут быть *прямыми* и *инверсными*. Для переключения триггера на его прямой вход необходимо подать сигнал x , а на инверсный — сигнал \bar{x} .

Так, определение «синхронный RS-триггер с инверсными статическими входами» означает, что рассматриваемый триггер имеет три информационных входа: вход установки S , вход сброса R и синхронизирующий вход C ; переключение триггера происходит в моменты времени, определяемые появлением активного логического сигнала на входе синхронизации ($C = 0$), причем для переключения на входы R или S необходимо подать низкий логический уровень, т. е. сигнал лог. 0 ($R = 0$ или $S = 0$). Такие входы соответственно обозначают \bar{C} , \bar{R} и \bar{S} .

Для описания работы триггера аналогично комбинационным схемам могут быть использованы словесное описание, таблицы истинности, логические выражения. Особенностью такого описания является использование в качестве дополнительной входной переменной значения сигнала Q_n , т. е. предыдущего значения выходного сигнала триггера. Однако наиболее часто для описания работы триггера используют таблицы переходов, определяющие, какие логические сигналы необходимо подать на его информационные входы для перехода из заданного состояния Q_n в заданное состояние Q_{n+1} .

6.1. Одноступенчатые триггеры

Асинхронный RS-триггер снабжен только двумя информационными входами: входом сброса R и входом установки S . По сути это простейший элемент памяти, который может быть реализован на элементах И-НЕ или ИЛИ-НЕ (рис. 6.2). В зависимости от этого данный триггер обладает либо инверсными, либо прямыми

Таблица 6.1

Таблица переходов RS-триггера на элементах И-НЕ

Q_n	Q_{n+1}	\bar{R}	\bar{S}
0	0	-	1
0	1	1	0
1	0	0	1
1	1	1	-

Таблица 6.2

Таблица переходов RS-триггера на элементах ИЛИ-НЕ

Q_n	Q_{n+1}	R	S
0	0	-	0
0	1	0	1
1	0	1	0
1	1	0	-

информационными входами. Опишем поведение асинхронных RS-триггеров рис. 6.2 с использованием таблиц переходов (табл. 6.1 и 6.2).

Из приведенных таблиц следует, что триггер на элементах И-НЕ снабжен инверсными, а на элементах ИЛИ-НЕ – прямыми информационными входами. Причем эти входы статические. Условные обозначения данных триггеров с учетом типа используемых информационных входов приведены соответственно на рис. 6.3 (а, б).

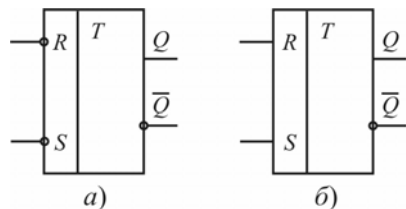


Рис. 6.3. Условное графическое изображение асинхронных RS-триггеров с инверсными (а) и прямыми (б) информационными входами

Отметим еще одну особенность рассмотренных триггеров. Допустим, что на оба информационных входа триггера подан активный логический сигнал. Например, для триггера рис. 6.3 (а) $R = S = 0$. Тогда, согласно рис. 6.2 (а), $Q = \bar{Q} = 1$, что противоречит постулатам (6.1), и при одновременном снятии информационных сигналов ($R = S = 1$) состояние триггера будет непредсказуемо. Следовательно, при использовании схем рис. 6.2 в качестве элемента памяти подача на оба информационных входа активных логических сигналов запрещена, хотя сам факт подачи такой комбинации сигналов вполне допустим.

Для получения математического выражения, описывающего работу триггера данного типа, составим карту Вейча

	R	R	\bar{R}	\bar{R}	
Q_n	0	10	1	1	Q_{n+1}
\bar{Q}_n	0	10	1	0	
	\bar{S}	S	S	\bar{S}	

Согласно определению входов, данному в начале настоящего раздела, для триггера с прямыми информационными входами при $S = 1$ имеем $Q_{n+1} = 1$. Аналогично для $R = 1$ имеем $Q_{n+1} = 0$. Если $\bar{R} = \bar{S} = 0$, т. е. на входах действуют пассивные логические уровни, состояние триггера остается неизменным: $Q_{n+1} = Q_n$.

Полученная карта подтверждает сделанный ранее вывод о недопустимости одновременной подачи на информационные входы R и S двух активных логических уровней ($R = S = 1$), так как в этом случае выходной сигнал Q_{n+1} одновременно должен принимать два взаимоисключающих значения (0 и 1).

Определяя возникшие факультативные значения выходной функции Q_{n+1} для $R = S = 1$ как сигнал лог. 1, получаем следующую ФАЛ:

$$Q_{n+1} = S + Q_n \bar{R}. \quad (6.2)$$

Для триггера с инверсными информационными входами аналогично можно получить

$$Q_{n+1} = \bar{S} + Q_n R. \quad (6.3)$$

Синхронный RS-триггер может быть получен на базе асинхронного RS-триггера введением дополнительной логической схемы, которая формировала бы на его входах активные логические уровни только при наличии дополнительного сигнала синхронизации. Синтезируем такую схему.

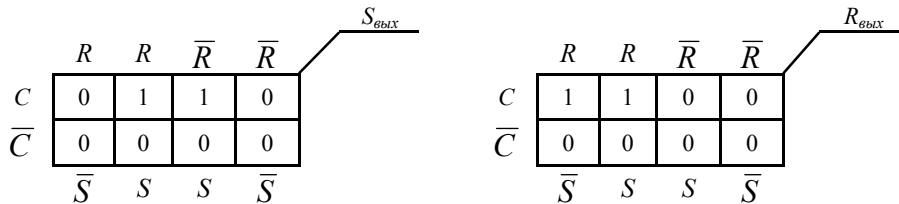
Предположим, что триггер снабжен прямым входом синхронизации. Информационные входы R - и S -триггера тоже прямые. В этом случае таблица истинности (табл. 6.3) дополнительной схемы будет иметь вид:

Таблица 6.3

Таблица истинности формирования управляющих сигналов синхронного RS-триггера

C	R _{вх}	S _{вх}	R _{вых}	S _{вых}
1	0	0	0	0
1	0	1	0	1
1	1	0	1	0
1	1	1	1	1
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0

С использованием полученной таблицы минимизируем ФАЛ для S_{вых} и R_{вых}.
Карты Вейча для этого случая имеют вид



Откуда получаем, что S_{вых} = CS, R_{вых} = CR.

Подставив полученные значения S_{вых} и R_{вых} в выражение (6.3) получим ФАЛ, описывающую работу синхронного RS-триггера

$$\begin{aligned}
 Q_{n+1} &= S + Q_n \bar{R} = CS + Q_n \bar{C} R = CS + Q_n (\bar{C} + R) = \\
 &= CS + Q_n \bar{C} + Q_n R = CS + Q_n \bar{C} + Q_n R (C + \bar{C}) = \\
 &= C(S + Q_n \bar{R}) + Q_n \bar{C} (1 + R).
 \end{aligned}$$

Окончательно будем иметь

$$Q_{n+1} = \bar{C} Q_n + C(S + Q_n \bar{R}) \quad (6.4)$$

Из анализа полученного выражения можно сделать следующие выводы:

- данная ФАЛ содержит два слагаемых;
- первое слагаемое представляет логическое произведение активного логического уровня сигнала синхронизации на ФАЛ, описывающую работу асинхронного триггера;
- второе слагаемое представляет логическое произведение пассивного логического уровня сигнала синхронизации на предыдущее состояние триггера.

Следует отметить, что аналогичную структуру имеют ФАЛ для всех синхронных триггеров.

Структурные схемы синхронных RS-триггеров, удовлетворяющих выражению (6.4) приведены на рис. 6.4.

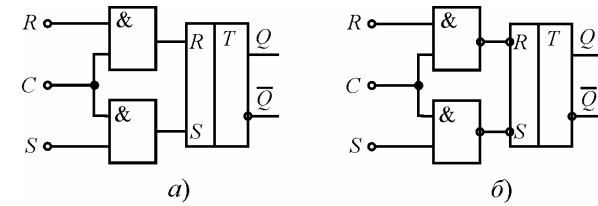


Рис. 6.4. Структурные схемы синхронных RS-триггеров на основе асинхронных триггеров с прямыми (а) и инверсными (б) входами

Синхронные RS-триггеры могут быть дополнены асинхронными установочными входами, сигналы которых подаются непосредственно на элемент памяти (рис. 6.5 (а)) и поэтому обладают более высоким приоритетом, чем сигналы синхронных входов. При использовании для построения синхронного триггера одностипных логических элементов (И-НЕ или ИЛИ-НЕ) его синхронные и асинхронные входы управляются различными активными логическими уровнями. В случае элементов И-НЕ для синхронных входов активными является сигнал лог.1, а для асинхронных входов – лог. 0. Условное графическое изображение синхронного RS-триггера с прямыми информационными и инверсными установочными входами показан на рис. 6.5 (б).

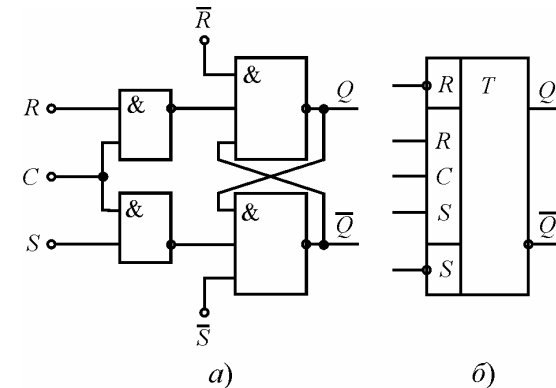


Рис. 6.5. Структурная схема (а) и условное графическое обозначение (б) синхронного RS-триггера с входами асинхронной установки

Следует отметить, что полученная структура синхронного триггера при условии C = 1 функционирует как асинхронный RS-триггер.

D-триггер обычно снабжен только одним информационным входом. Этот вход D, информация с которого по определению входа переписывается на выход триггера только по сигналу синхронизации. Из сказанного следует, что D-триггер может быть только синхронным. В соответствии со сказанным, таблица переходов для D-триггера приведена в табл. 6.4.

Таблица 6.4

Таблица переходов D-триггера

C	Q_n	Q_{n+1}	D
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1
0	0	0	0
0	0	0	1
0	1	1	0
0	1	1	1

Так как информация на выходе D -триггера остается неизменной вплоть до прихода очередного импульса синхронизации, данный триггер часто называют триггером с запоминанием информации, или триггером защелкой.

Синтезируем структуру D -триггера на основе синхронного RS -триггера. Очевидно, что для реализации алгоритма работы, описанного табл. 6.4, сигналы на его R и S входах должны быть связаны с сигналом на D -входе следующими соотношениями: $R' = \bar{D}, S' = D$.

Подставим эти соотношения в ФАЛ RS -триггера и получим выражение, описывающее работу D -триггера

$$Q_{n+1} = \bar{C}Q_n + C(S' + Q_n R') = \bar{C}Q_n + C(D + Q_n \bar{D}) = \bar{C}Q_n + CD(1 + Q_n),$$

или

$$Q_{n+1} = \bar{C}Q_n + CD. \quad (6.5)$$

Структурная схема, удовлетворяющая полученной ФАЛ, и условное графическое изображение D -триггера, приведены на рис 6.6 (а, б).

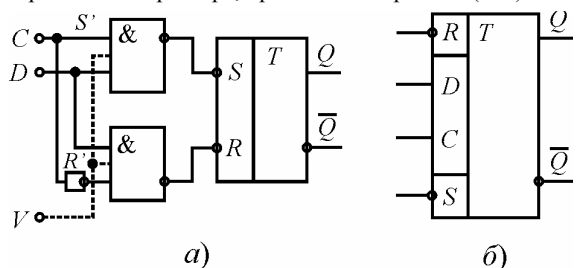


Рис. 6.6. Структурная схема (а) и условное графическое изображение (б) D -триггера

Разновидностью D -триггера является VD -триггер. Этот триггер дополнительно снабжен входом разрешения работы V , который блокирует его работу. По сути, действие данного входа аналогично действию входа C . Поэтому структурно этот вход может быть выполнен, например, как объединение дополнительных входов элементов И-НЕ входной логики триггера (рис. 6.6 (а)). При $V=1$ поведение триггера полностью соответствует выражению (6.5). При $V=0$ триггер хранит записанную ранее в него информацию: $Q_{n+1} \equiv Q_n$.

D -триггер также может быть снабжен дополнительными входами асинхронной установки. С учетом входа блокировки V и инверсных асинхронных входов установки R и S ФАЛ D -триггера имеет вид

$$Q_{n+1} = [V(CQ_n + CD) + \bar{V}Q_n + \bar{S}]R. \quad (6.6)$$

6.2. Двухступенчатые триггеры

T -триггер, или, как его называют, счетный триггер, согласно определению входов должен изменять свое состояние по каждому активному логическому сигналу, действующему на информационном входе T . Данному алгоритму работы соответствует таблица переходов (табл. 6.5), в которой активным считается сигнал лог. 1 и логическое выражение вида

$$Q_{n+1} = Q_n \bar{T} + \bar{Q}_n T. \quad (6.7)$$

Это выражение характеризует работу асинхронного T -триггера. Используя (6.7) и замечания, сделанные при рассмотрении синхронного RS -триггера, можно получить выражение для синхронного T -триггера:

$$Q_{n+1} = (Q_n \bar{T} + \bar{Q}_n T)C + \bar{C}Q_n. \quad (6.8)$$

Таблица 6.5

Таблица переходов T -триггера

Q_{n+1}	Q_n	T
0	0	0
0	1	1
1	0	1
1	1	0

Реализация данного режима работы с использованием простейшего элемента памяти требует подачи на последний для каждого последующего подключения инверсных значений входных сигналов. Такой режим обеспечивается введением в RS - или D -триггер цепей дополнительной обратной связи (рис. 6.7)

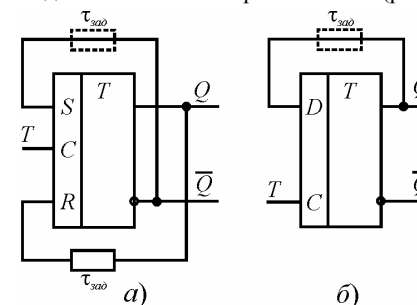


Рис. 6.7. Асинхронные T -триггеры на основе RS (а) и D -триггеров (б)

Полученные структуры соответствуют алгоритму работы T -триггера. Однако на практике они не могут обеспечить его надежное функционирование. Объясняется это тем, что в данном случае элемент памяти одновременно должен выполнять две взаимоисключающие функции. С одной стороны, он должен быть источником информации, а с другой, - ее приемником. Очевидно, что одновременное выполнение этих функций одним триггером невозможно, так как прием новой информации

автоматически означает потерю старой. Поэтому при идеальном быстродействии используемых ЛЭ приведенные структуры неработоспособны.

Неработоспособны данные структуры и при использовании элементов, обладающих собственной инерционностью, либо внешних цепей задержки (рис. 6.7). Объясняется это следующим: если к моменту изменения сигналов на входах R , S или D сигнал T еще не снят, происходит следующее переключение триггера. В результате устройство начинает функционировать как генератор незатухающих колебаний. Для их устранения необходимо либо использовать дополнительный элемент памяти, запоминающий новые значения сигналов R , S или D и подающий их на информационные входы основного элемента памяти только после снятия активного сигнала со входа T , либо искусственно ограничивать длительность сигнала T . Эти принципы реализованы в так называемых двухступенчатых триггерах и триггерах с динамическим управлением.

В двухступенчатом триггере противоречия между процессами хранения старой и приемом новой информации решены введением второго запоминающего элемента. При этом новая информация формируется сначала только во входной ступени триггера при сохранении старой информации в выходной его ступени. Когда новое состояние входной ступени сформировано и снят активный уровень сигнала со входа T , происходит ее перезапись в выходную ступень устройства. Таким образом, двухступенчатый триггер фактически состоит из двух последовательно соединенных триггеров - ведущего и ведомого (рис. 6.8 (а)). Для устранения режима автоколебаний синхронизация работы ведущего и ведомого триггеров осуществляется инверсными логическими уровнями. Часто двухступенчатый триггер называют MS -триггером (от английских слов *master* и *slave*—хозяин и раб). Следует отметить, что по структуре двухступенчатого могут быть построены любые типы триггеров.

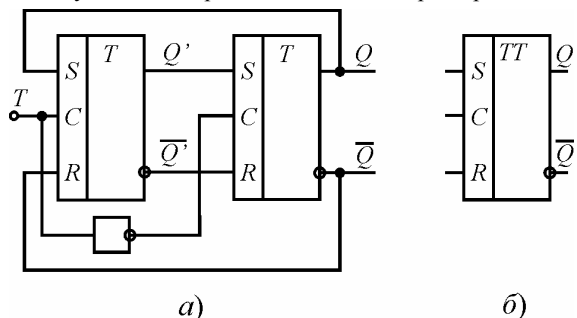


Рис. 6.8. Двухступенчатый T -триггер (а) и условное графическое обозначение двухступенчатого RS -триггера (б)

Из сказанного следует, что использование двухступенчатых триггеров, выполненных по структурной схеме рис. 6.8 (а), позволяют обеспечить высокую надежность функционирования триггеров с внутренними цепями обратной связи. На принципиальных схемах двухступенчатые триггеры обозначаются двоякой буквой (ТТ), как это показано на рис. 6.8 (б).

JK-триггер является наиболее универсальным триггером, так как на его основе могут быть выполнены любые из описанных ранее типов триггеров. В отличие от RS -триггера JK -триггер не имеет запрещенных комбинаций входных сигналов, и его

таблица переходов (табл. 6.6) в предположении, что его активным является сигнал лог. 1 имеет следующий вид:

Таблица 6.6
Таблица переходов JK -триггера

Q_{n+1}	Q_n	J	K
0	0	0	-
0	1	1	-
1	0	-	1
1	1	-	0

Эта таблица отражает данное ранее определение входов J и K , которые при подаче на них активного входного сигнала соответственно устанавливают и сбрасывают JK -триггер.

Составив карту Вейча для приведенной таблицы, можно получить аналитическое выражение, описывающее работу асинхронного и синхронного JK -триггеров:

$$\begin{aligned} Q_{n+1} &= J\bar{Q}_n + \bar{K}Q_n, \\ Q_{n+1} &= C(J\bar{Q}_n + \bar{K}Q_n) + \bar{C}Q_n. \end{aligned} \quad (6.9)$$

Из таблицы переходов следует, что при подаче на информационные входы J и K сразу двух активных логических уровней триггер работает как счетный. Следовательно, структурная схема данного триггера должна повторять схему T -триггера, т.е. должна содержать дополнительные цепи обратной связи. Поэтому структура JK -триггера базируется на структуре MS -триггера или триггера с динамическим входом.

Поскольку введение обратных связей решает задачу исключения запрещенных комбинаций входных сигналов, входы J и K могут функционировать как входы R и S RS -триггера.

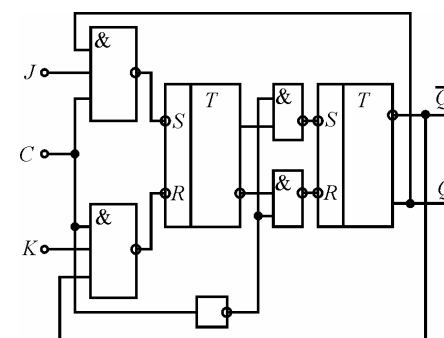


Рис. 6.9. Структурная схема JK -триггера

Выполненная с учетом сделанных замечаний структурная схема JK -триггера приведена на рис. 6.9. Очевидно, что всегда на вход одного из логических элементов 3И-НЕ с выхода триггера поступает активный логический сигнал. Вследствие этого при любых комбинациях входных сигналов непосредственно на входах первого RS -триггера не могут появиться два активных логических уровня.

Ранее уже отмечалось, JK-триггер является универсальным и позволяет построить на своей основе все рассмотренные ранее типы триггеров. Сказанное иллюстрируется на рис. 6.10. Следует отметить, что JK-триггер, аналогично ранее рассмотренным, может дополнительно снабжаться входами асинхронной установки R и S, сигналы на которых имеют приоритет над сигналами остальных входов.

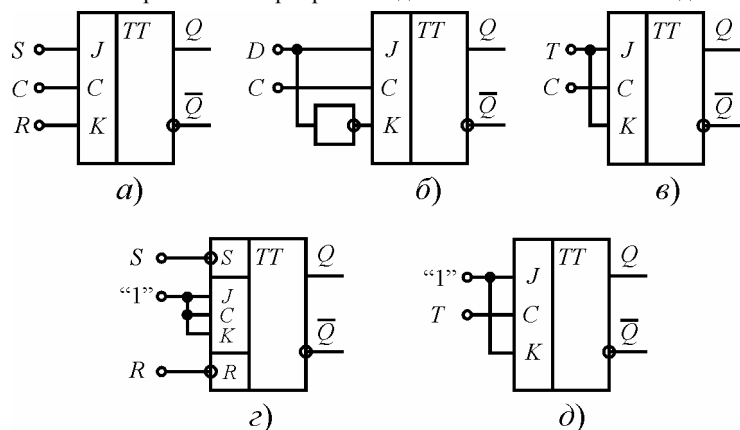


Рис. 6.10. Построение триггеров на основе JK-триггера: (а) – синхронный RS-триггер; (б) – D-триггер; (в) – синхронный T-триггер; (г) – асинхронный RS-триггер; (д) – асинхронный T-триггер

6.3. Триггеры с динамическим управлением

Все рассмотренные типы триггеров являются триггерами со статическим управлением. Это накладывает определенные требования на вид сигнала синхронизации. Так как активным логическим уровнем на входе синхронизации рассмотренных ранее триггеров является сигнал $C=1$, то на интервале действия этого сигнала они беспрепятственно воспринимают любое изменение сигналов на своих информационных входах. Поэтому для записи определенной информации сигнал на информационных входах триггера должен оставаться неизменным на всем интервале действия активного логического сигнала на входе синхронизации. На практике такое требование приводит к дополнительным трудностям при проектировании цифровых устройств и их усложнению.

От указанного недостатка свободны триггеры с так называемым динамическим управлением. Суть такого управления заключается в том, что в качестве активного логического сигнала выступает не сам статический уровень, а его изменение. Другими словами, переключение триггера происходит в течение короткого промежутка времени вблизи фронта или среза импульса синхронизации.

Если триггер реагирует на фронт импульса, т.е. на перепад сигнала от лог. 0 к лог. 1, то считают, что он снабжен *прямым динамическим* входом. Если же триггер реагирует на срез импульса, т.е. на перепад сигнала от лог. 1 к лог. 0, то считают, что он снабжен *инверсным динамическим* входом. Обозначение на принципиальных электрических схемах таких входов показано на рис. 6.11.

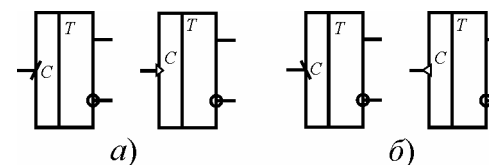


Рис. 6.11. Обозначение входов синхронизации триггеров: (а) – прямой динамический; (б) – инверсный динамический

Здесь следует отметить, что триггер с динамическим управлением не чувствителен к изменению информационных сигналов на интервалах действия сигналов $C = 1$ и $C = 0$. Переключение происходит по изменению сигнала синхронизации.

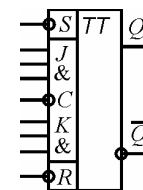


Рис. 6.12. Условное графическое обозначение JK-триггера с объединенными операциями И входами J и K

Строго говоря, переключение триггера происходит не непосредственно по фронту или срезу импульса синхронизации. Для гарантированной записи информации она должна оставаться неизменной на информационном входе на некотором интервале времени в окрестности изменения сигнала синхронизации. Продолжительность этого интервала времени определяется реальными задержками в выбранных логических элементах.

На практике ИС триггеров часто снабжаются несколькими однотипными входами, например S, R, J, K, объединенными по входу операцией И. Такое исполнение существенно упрощает схемы устройств, проектируемых с применением данных микросхем. В качестве примера на рис. 6.12 приведено условное графическое обозначение JK-триггера, имеющего по три входа J и K, объединенных логической операцией И.

Контрольные вопросы и упражнения к разделу 6

1. Каково назначение и состав триггерных устройств?
2. Приведите принципиальную схему и передаточную характеристику гистерезисного компаратора, построенного на основе операционного усилителя.
3. Опишите принцип работы гистерезисного компаратора.
4. Что называется постулатом триггера?
5. По каким признакам классифицируют триггеры?
6. Дайте классификацию триггеров по типам информационных входов.
7. Что означает определение: «синхронный RS-триггер с инверсными статическими входами»?
8. Какой тип триггера называют триггером защелкой?
9. Почему JK-триггер является наиболее универсальным?
10. Чем различаются между собой одно- и двухступенчатые триггеры различных типов?
11. Проиллюстрируйте выполнение триггеров различных типов на основе JK-триггера.
12. Каковы особенности триггеров с динамическим управлением?

7. ФУНКЦИОНАЛЬНЫЕ УЗЛЫ ПОСЛЕДОВАТЕЛЬНОСТНЫХ ЛОГИЧЕСКИХ УСТРОЙСТВ

7.1 Регистры

Регистром называется последовательное устройство, предназначенное для записи, хранения и (или) сдвига информации, представленной в виде многоразрядного двоичного кода.

В соответствии с данным определением в общем случае регистр может выполнять следующие микрооперации над кодовыми словами:

1. установка в исходное состояние (запись нулевого кода);
2. запись входной информации в последовательной форме;
3. запись входной информации в параллельной форме;
4. хранение информации;
5. сдвиг хранимой информации вправо или влево;
6. выдача хранимой информации в последовательной форме;
7. выдача хранимой информации в параллельной форме.

Любой N -разрядный регистр состоит из N однотипных ячеек - разрядных схем, выходной сигнал каждой из которых ассоциируется с весовым коэффициентом соответствующего разряда двоичного кода. При этом каждая разрядная схема, как любое последовательное устройство, состоит из триггерной подсистемы (элемента памяти) и некоторой комбинационной схемы, преобразующей входные воздействия и состояния триггерной подсистемы в выходные сигналы регистра.

Регистры могут быть классифицированы по различным признакам. Рассмотрим основные из них.

По способу приема информации регистры подразделяют на:

- *параллельные* (статические), в которые информация записывается и считывается только в параллельной форме;
- *последовательные* (сдвигающие), в которые информация записывается и считывается только в последовательной форме;
- *последовательно-параллельные*, в которые информация записывается или считывается как в параллельной, так и в последовательной формах.

По числу каналов передачи информации регистры подразделяют на:

- *парафазные*, в которых информация записывается и считывается в прямом и обратном кодах;
- *однофазные*, в которых информация записывается и считывается либо в прямом, либо в обратном коде.

По способу тактирования регистры подразделяют на:

- *однотактные*, управляемые одной управляющей последовательностью импульсов;
- *многотактные*, управляемые несколькими управляющими последовательностями импульсов.

Параллельный регистр предназначен для выполнения только 1-ой, 3-ей, 4-ой и 7-ой из перечисленных микроопераций над кодовыми словами, т.е. обрабатывает информацию только в параллельной форме. Поэтому образующие его разрядные схемы не связаны между собой.

Простейший параллельный регистр может быть выполнен на RS-триггерах (рис. 7.1).

Он содержит N триггеров, входы синхронизации которых объединены между собой. На его входы x_{N-1}, \dots, x_0 и $\bar{x}_{N-1}, \dots, \bar{x}_0$ информация подается в прямом и обратном кодах, а с выходов Q_{N-1}, \dots, Q_0 снимается только в обратном коде. Следовательно, согласно приведенной выше классификации, это однотактный регистр с парафазными входами и однофазным инверсным выходом. Запись информации в такой регистр выполняется за один такт синхронизации.

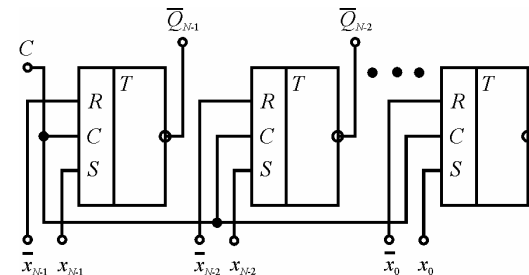


Рис. 7.1. Схема параллельного регистра

Если для записи в данном регистре используется только прямой или обратный входной код, то запись информации выполняется за два такта синхронизации. По первому такту необходимо сбросить или установить все триггеры регистра (подачей на соответствующие входы активного логического уровня), а по второму - записать в регистр новую информацию.

Если в данной структуре RS-триггеры заменить на D-триггеры, получим однотактный регистр с однофазным входом. Очевидно, что быстродействие такого регистра при использовании однофазного входного сигнала будет в два раза выше, так как для записи информации необходим только один импульс синхронизации.

Сдвигающий регистр. Параллельный (сдвигающий) регистр является, как правило, универсальным и может выполнять все доступные для регистров микрооперации. Для этого разрядные схемы, входящие в его состав, соединены между собой. Рассмотрим построение такого регистра на примере однотактного сдвигающего регистра с возможностью параллельной записи информации (рис. 7.2 (а)). Для упрощения на рис. 7.2 (а) показаны только две разрядные схемы регистра.

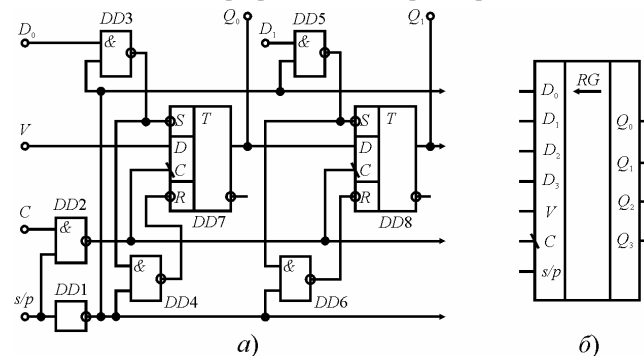


Рис. 7.2. Фрагмент структурной схемы сдвигающего регистра (а); условное графическое обозначение сдвигающего регистра (б)

Выбор режима работы регистра определяется значением сигнала входе s/p . При сигнале $s/p = 0$ элемент $DD1$ 2И-НЕ, формируя на своем выходе сигнал лог. 1, не зависимо от сигнала синхронизации, поданного на вход C , блокирует синхронное переключение триггеров разрядных схем регистра. Одновременно выходной сигнал инвертора $DD2$ преобразует элементы 2И-НЕ $DD3$ и $DD5$ в инверторы и сигналы, присутствующие на входах параллельной записи информации D , переписываются в триггеры разрядных схем. Элементы $DD4$ и $DD6$ в данном режиме также работают как инверторы, предотвращая возможность одновременной подачи на асинхронные входы R - и S -триггеров двух активных логических уровней.

Если $s/p = 1$, параллельная запись информации становится невозможной, так как логические элементы $DD3$ - $DD6$ не зависимо от сигналов на входах параллельной записи D формируют на асинхронных входах R - и S -триггеров пассивные логические сигналы. Одновременно элемент $DD1$ 2И-НЕ превращается в инвертор, и по фронту импульса синхронизации информация со входа V (вход последовательного приема информации) записывается в триггер первой разрядной схемы регистра. В триггер второй разрядной схемы переписывается информация из первой разрядной схемы и т.д. Регистр выполняет прием информации в последовательном виде и сдвиг ранее записанной в него информации влево (из младшего разряда в старший).

Условное графическое изображение 4-х разрядного регистра сдвига показано на рис. 7.2 (б).

Если регистр выполняет операцию сдвига информации из старшего разряда в младший, то на его условном графическом обозначении стрелка указывает вправо. Если стрелки показывают в обе стороны, то это значит, что регистр является реверсивным, и направление сдвига информации определяется управляющим сигналом.

Следует отметить, что в сдвигающих регистрах используются только двухступенчатые триггеры или триггеры с динамическим управлением. Это гарантирует сдвиг информации строго на один разряд по каждому импульсу синхронизации. При использовании других триггеров процесс сдвига становится неуправляемым и за один импульс синхронизации информация может быть сдвинута на несколько разрядов.

Организация межрегистровых связей. В реальных цифровых устройствах информация, подлежащая обработке, хранится в регистрах. Обработка информации предполагает передачу ее из регистра в регистр. Эта передача может быть выполнена различными способами.

В простейшем случае между всеми регистрами может быть организована индивидуальная связь. Этот способ реализует пространственное разделение цепей передачи информации, при котором обмен между несколькими регистрами может производиться одновременно, что обеспечивает получение максимально высокого быстродействия ее обработки. Однако при большом числе регистров реализация данного способа наталкивается на большие технические трудности ввиду громоздкости линий связи.

В микропроцессорных системах обмен информацией между несколькими регистрами осуществляется по последовательному принципу. При этом используется единственная линия связи (общая шина DB - Data Bus). К этой шине подключены входы и выходы всех регистров. При этом реализуется принцип разделения информации во времени, при котором в каждый конкретный момент обмен возможен

только между двумя регистрами. Путь передачи информации в каждый конкретный момент определяется значениями управляющих сигналов.

7.2. Счетчики

Счетчиком называется последовательностное устройство, предназначенное для счета входных импульсов и фиксации их числа в двоичном коде.

Счетчики так же, как и сдвиговые регистры, строятся на основе N однотипных связанных между собой разрядных схем, каждая из которых в общем случае состоит из триггера и некоторой комбинационной схемы, предназначенной для формирования сигналов управления триггером.

В цифровых схемах счетчики могут выполнять следующие микрооперации над кодовыми словами:

- 1) установка в исходное состояние (запись нулевого кода);
- 2) запись входной информации в параллельной форме;
- 3) хранение информации;
- 4) выдача хранимой информации в параллельной форме;
- 5) инкремент – увеличение хранящегося кодового слова на единицу;
- 6) декремент – уменьшение хранящегося кодового слова на единицу.

Основные параметры и классификация счетчиков. Основным статическим параметром счетчика является *модуль счета M* , который характеризует максимальное число импульсов, после прихода которого счетчик устанавливается в исходное состояние.

Основным динамическим параметром, определяющим быстродействие счетчика, является *время установления* выходного кода t_k , характеризующее временной интервал между моментом подачи входного сигнала и моментом установления нового кода на выходе.

Счетчики могут классифицироваться по многим параметрам. Рассмотрим основные из них.

По значению модуля счета счетчики подразделяют на:

- *двоичные*, модуль счета которых равен целой степени числа 2 ($M = 2^n$);
- *двоично-кодированные*, в которых модуль счета может принимать любое, не равное целой степени числа 2 значение.

По направлению счета счетчики подразделяют на:

- *суммирующие*, выполняющие микрооперацию инкремента над хранящимся кодовым словом;
- *вычитающие*, выполняющие микрооперацию декремента над хранящимся кодовым словом;
- *реверсивные*, выполняющие в зависимости от значения управляющего сигнала над хранящимся кодовым словом либо микрооперацию декремента, либо инкремента.

По способу организации межразрядных связей счетчики делятся на:

- *счетчики с последовательным переносом*, в которых переключение триггеров разрядных схем осуществляется последовательно один за другим;
- *счетчики с параллельным переносом*, в которых переключение всех триггеров разрядных схем осуществляется одновременно по сигналу синхронизации;
- *счетчики с комбинированным последовательно-параллельным переносом*, при котором используются различные комбинации способов переноса.

Двоичные счетчики. Для определения структуры двоичного счетчика рассмотрим последовательность двоичных чисел, ограничившись для простоты 3-разрядным кодом (табл. 7.1).

Таблица 7.1

Таблица 3-разрядных двоичных чисел

C	Q_0	Q_1	Q_2
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

Так как в счетчике значения Q ассоциируются с выходными сигналами соответствующих триггеров, то для получения счетчика с модулем счета $M=8$ необходимо как минимум три триггера. Рассматривая табл. 7.1, нетрудно заметить, что Q_0 , соответствующее младшему разряду двоичного числа, изменяет свое значение с приходом каждого импульса синхронизации; Q_1 - с приходом каждого второго импульса синхронизации, а Q_2 - с приходом каждого четвертого импульса. Данный алгоритм можно легко реализовать, используя асинхронные T -триггеры, причем синхронизацию каждого последующего триггера осуществляют выходным сигналом предыдущего, а переключение первого триггера, формирующего значение Q_0 , непосредственно последовательностью синхриимпульсов. Временные диаграммы, поясняющие такой алгоритм работы, приведены на рис. 7.3.

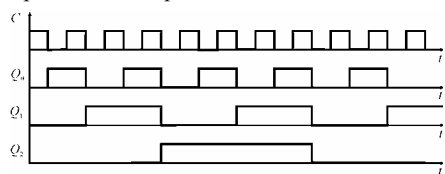


Рис. 7.3. Временные диаграммы работы суммирующего счетчика

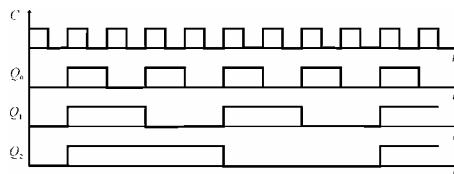


Рис. 7.4. Временные диаграммы работы вычитающего счетчика

Из таблицы и временных диаграмм также следует, что для организации микрооперации инкремента переключение каждого последующего триггера должно происходить в момент изменения выходного сигнала предыдущего триггера из 1 в 0, т.е. по срезу импульса. Следовательно, для суммирующего счетчика последовательно включенные асинхронные T -триггеры должны быть снабжены инверсными динамическими входами (рис. 7.5).

Если исходные асинхронные T -триггеры снабжены прямыми динамическими входами, то счетчик превращается в вычитающий и выполняет микрооперацию декремента. Временные диаграммы, поясняющие данный режим работы, приведены на рис. 7.4.

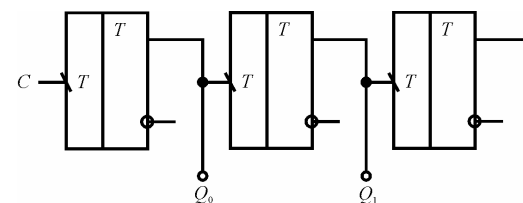


Рис. 7.5. Схема трехразрядного суммирующего счетчика с последовательным переносом

Сказанное справедливо, если для синхронизации каждого последующего асинхронного T -триггера использовать прямой выход предыдущего триггера (вывод Q). Если для этой цели использовать инверсный выход триггера, то суммирующий счетчик реализуется на T -триггерах с прямыми динамическими входами, а вычитающий - на T -триггерах с инверсными динамическими входами.

Таким образом, в счетчиках на синхронных T -триггерах направление счета зависит как от того, какой из выходов используется для синхронизации последующего триггера, так и от типа входа синхронизации. В табл. 7.2 приведены все возможные комбинации соединения триггеров с различными типами входов синхронизации и получаемые при этом виды счетчиков.

Таблица 7.2

Зависимость типа выполняемой микрооперации от вида межразрядных связей

Тип входа T	Используемый выход	
	Q	\bar{Q}
Прямой динамический	Декремент	Инкремент
Инверсный динамический	Инкремент	Декремент

Из сказанного также следует, что направление счета счетчика может изменяться путем изменения вида межразрядных связей. Последнее легко достигается включением в состав каждой разрядной схемы счетчика мультиплексора (рис. 7.6).

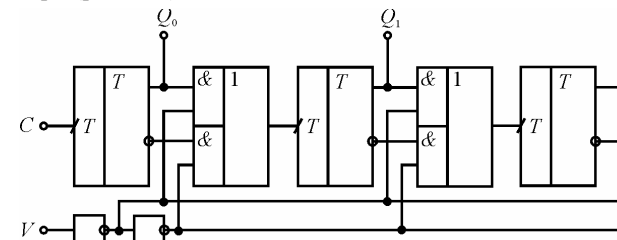


Рис. 7.6. Фрагмент структурной схемы реверсивного счетчика

Сигнал на входе V данного счетчика определяет вид межразрядных связей, а следовательно, и тип получаемого счетчика. Временные диаграммы, поясняющие работу такого счетчика при различных значениях сигнала V , приведены на рис. 7.7.

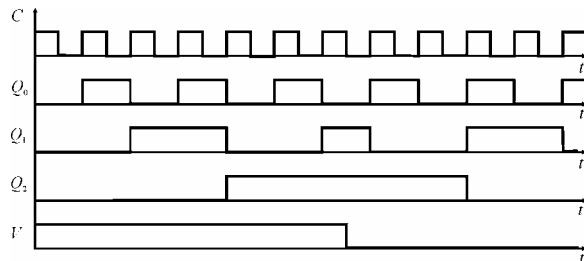


Рис. 7.7. Временные диаграммы работы реверсивного счетчика

Все рассмотренные раньше счетчики являются счетчиками с последовательным переносом, так как переключение каждого последующего триггера может произойти только после переключения предыдущего. Данный тип счетчика отличается простотой внутренней структуры. Однако следствием такой организации является большое время установления выходного кода, которое к тому же не остается постоянным в процессе работы и зависит от конкретного значения его выходного кода. Максимальное значение t_k наблюдается в случае необходимости переключения всех триггеров счетчика, например при изменении выходного кода со значения 111 ... в 000 ... или наоборот. Численно

$$t_{k \max} = N t_{k, \text{тр}}, \quad (7.1)$$

где N – число разрядов счетчика; $t_{k, \text{тр}}$ – время переключения (установления выходного кода) одного разряда счетчика.

Уменьшить время установления выходного кода счетчика можно при условии, что все триггеры его разрядных схем будут переключаться одновременно. Для этого необходимо отказаться от применения асинхронных триггеров в пользу синхронных и сформировать сигналы, регламентирующие требуемый порядок переключения триггеров разрядных схем до прихода импульса синхронизации.

Для получения алгоритма переключения триггеров разрядных схем вернемся еще раз к таблице последовательности двоичных чисел (см. табл. 7.1). Анализируя таблицу, нетрудно заметить, что переключение каждого последующего триггера при приходе очередного импульса синхронизации происходит только тогда, когда все предыдущие триггеры установлены, т. е. на выходах присутствуют единичные сигналы. Математически данный алгоритм можно записать следующей ФАЛ:

$$Q_{i, n+1} = \bar{Q}_{i, n} p_i + Q_{i, n} \bar{p}_i = Q_i \oplus p_i, \quad (7.2)$$

где $Q_{i, n+1}$ – значение i -го разряда выходного кода счетчика в $(n+1)$ -й момент времени; $Q_{i, n}$ – значение i -го разряда выходного кода счетчика в n -й момент времени; $p_i = Q_{0, n} Q_{1, n} \dots Q_{i-1, n}$ – сигнал переноса.

Итак, для одновременного переключения триггеров всех разрядных схем в счетчике необходимо сформировать сигнал переноса. Схемотехническая реализация такого алгоритма переключения приведена на рис. 7.8. Следует отметить, что в данной структуре триггер, формирующий сигнал Q_0 , по прежнему остался асинхронным. Поэтому его входной сигнал $T \equiv 1$.

Очевидно, что в данной схеме время установления выходного кода t_k max будет равно времени переключения одного триггера $t_{k, \text{тр}}$. Однако если счетчик с последовательным переносом непосредственно после установления нового значения

выходного кода готов к следующему переключению, то при реализации данного алгоритма для подготовки счетчика к следующему переключению должно пройти некоторое время $t_{\text{под}}$. Это время необходимо для формирования нового сигнала переноса и определяется временем задержки распространения логических элементов И ($t_{\text{под}} = t_{\text{эл}}$), использующихся в цепях формирования сигналов p_i . Так как это время всегда меньше времени установления выходного кода одиночного триггера, быстродействие полученного счетчика всегда выше быстродействия счетчика с последовательным переносом.

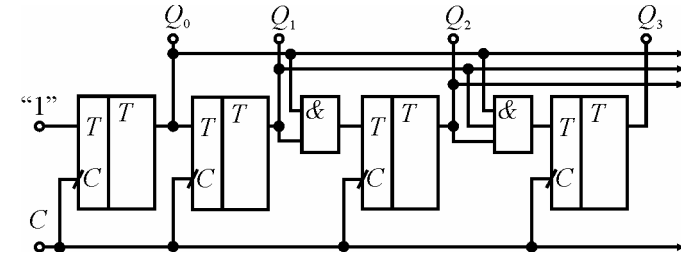


Рис. 7.8. Фрагмент схемы счетчика с параллельным переносом

Счетчики, реализующие описанный алгоритм работы, называются счетчиками с параллельным переносом.

Следует отметить, что в счетчиках с параллельным переносом направление счета не зависит от того, какой (прямой или инверсный) динамический вход имеют триггеры, составляющие его разрядные схемы. Направление счета определяется исключительно тем, какой (прямой или инверсный) выход триггера используется для формирования сигнала переноса. Так, счетчик, схема которого показана на рис. 7.8, будет суммирующим. Если же для формирования сигнала переноса будут использованы инверсные выходы триггеров разрядных схем, счетчик будет вычитающим (табл. 7.3).

Таблица 7.3

Зависимость типа операции, реализуемой счетчиком с параллельным переносом, от вида межразрядных связей

Сигнал связи	Выполняемая микрооперация
Q_i	Инкремент
\bar{Q}_i	Декремент

Следовательно, и при использовании параллельного переноса введением в разрядные схемы мультиплексоров на элементах 2Х2/И-ИЛИ можно легко построить реверсивный счетчик.

Сложность практической реализации счетчиков с параллельным переносом состоит в том, что при увеличении числа разрядов пропорционально увеличивается число входов логических элементов И, используемых в цепях формирования сигнала переноса. Поэтому при увеличении числа разрядов используют структуры счетчиков с комбинированным, как правило, последовательно-параллельным или параллельно-параллельным переносом.

Двоично-кодированные счетчики. Как уже отмечалось ранее, двоично-кодированный счетчик имеет модуль счета, отличный от целой степени числа 2.

Примером такого счетчика может служить счетчик с модулем счета 10, т. е. двоично-десятичный счетчик.

При построении таких счетчиков получили распространение в основном два метода: *метод исключения лишних состояний*; *метод управляемого сброса*.

Первый метод заключается в использовании методики синтеза последовательных устройств (который будет рассмотрен в следующем разделе). Полученная в этом случае схема однозначно определяет виды связей между ее отдельными элементами и не может быть оперативно изменена. Любое изменение требует выполнения в полном объеме всего цикла проектирования. Поэтому такой подход используется только при проектировании часто встречающихся устройств, выпускаемых большими партиями. В иных случаях его применение становится экономически нецелесообразным.

На практике, как правило, желательно иметь схему полуфабрикат, которая без особых сложностей и дополнительных элементов позволила бы гибко изменять алгоритм работы устройства. Этого легко добиться, если разрядные схемы счетчика строить с учетом возможности предварительной асинхронной установки исходного состояния. Используя счетчики с входами предварительной установки, можно построить устройство с любым наперед заданным модулем счета. Для этого используют *метод управляемого сброса*.

Идея данного метода состоит в принудительном формировании сигнала сброса триггеров разрядных схем двоичного счетчика при появлении на его выходе кода, совпадающего с требуемым модулем счета M . Проиллюстрируем сказанное на примере преобразования двоичного 4-разрядного счетчика со сквозным переносом в двоично-десятичный счетчик (рис. 7.9).

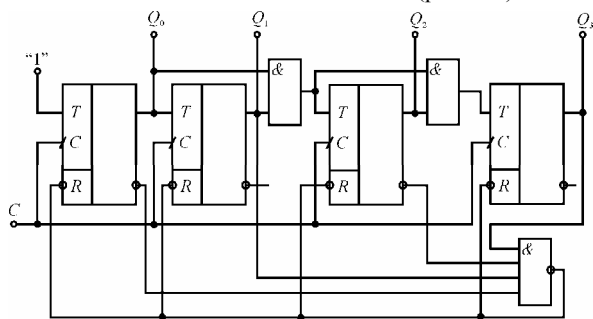


Рис. 7.9. Использование метода управляемого сброса для преобразования двоичного счетчика в двоично-десятичный

Для этого необходимо на входы дополнительного логического элемента 4И-НЕ подать комбинацию выходных сигналов разрядных схем, соответствующую коду 1010, т. е. $Q_3\bar{Q}_2Q_1\bar{Q}_0$. В этом случае при появлении на выходе одиннадцатого по счету выходного кода (1010) элемент 4И-НЕ через время, равное времени его задержки распространения, сформирует сигнал сброса и на выходе счетчика установится нулевой код. На рис. 7.10 приведены временные диаграммы, иллюстрирующие работу такого счетчика.

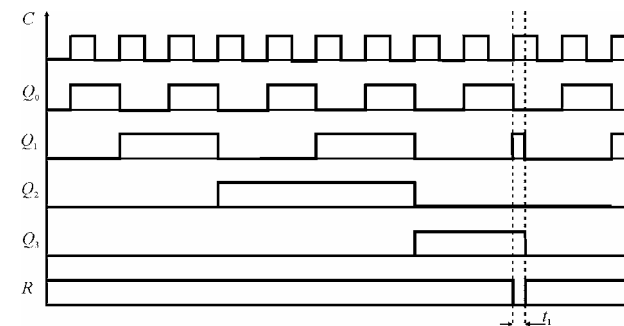


Рис. 7.10. Временные диаграммы работы двоично-десятичного счетчика, реализующего метод управляемого сброса

Следует подчеркнуть, что при использовании метода управляемого сброса на выходе счетчика на время t_1 , равное сумме времени задержки распространения сигнала в дополнительном элементе 4И-НЕ и времени установления выходного кода счетчика по входу R , устанавливается лишнее выходное состояние. Это является платой за универсальность метода. Если по условию работы счетчика даже кратковременное появление на его выходе лишнего состояния является недопустимым, при проектировании необходимо пользоваться методом исключения лишних состояний, например, по выходному коду $M-1$ формировать управляющие сигналы, обеспечивающие по следующему импульсу синхронизации сброс всех триггеров счетчика.

Контрольные вопросы и упражнения к разделу 7

1. По каким признакам можно провести классификацию регистров? Приведите классификацию регистров.
2. Приведите структурные схемы и условные обозначения параллельного, сдвигающего и реверсивного регистра. Опишите принцип их работы.
3. Назовите основные параметры и признаки классификации счетчиков.
4. Каким образом достигается повышение быстродействия счетчиков?
5. Как осуществляется предварительная установка счетчиков?
6. В чем суть метода управляемого сброса?

8. БАЗОВЫЕ ЛОГИЧЕСКИЕ ЭЛЕМЕНТЫ

Значениям логических функций и аргументов в электрических схемах могут быть поставлены в соответствие различные характеристики токов или напряжений. Это могут быть как их некоторые установившиеся величины, так и параметры электрических импульсов. В соответствии с этим различают потенциальный и импульсный способы представления логических переменных.

При *потенциальном способе представления* значениям лог. 0 и лог. 1 ставят в соответствие два различных уровня напряжения или тока. В зависимости от того, какой именно из этих уровней конкретно ставят в соответствие значениям 0 и 1, различают положительную и отрицательную логики.

В *положительной логике* значению лог. 1 ставят в соответствие большее, а значению лог. 0 - меньшее значение напряжения или тока. В *отрицательной логике* наоборот - большим значениям электрического сигнала соответствуют значения лог. 0, меньшим - значения лог. 1. Это определение справедливо для любой полярности

используемого напряжения или любых направлениях протекания токов.

Из рис. 8.1 (а) видно, что при положительной логике и условии $U_{\text{вых}} \geq 0$ значению лог. 1 соответствует большее по абсолютному значению напряжение, в то время как при $U_{\text{вых}} < 0$ большее абсолютное напряжение соответствует значению лог. 0. Однако с учетом знака уровни, показанные на рис. 8.1 (а, б), полностью соответствуют данному выше определению типов логики. Важно отметить, что если некоторое логическое устройство относительно положительной логики реализует операцию И, то относительно отрицательной логики это будет операция ИЛИ, и наоборот.

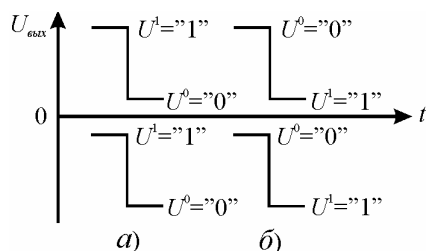


Рис. 8.1. Потенциальный способ представления лог. 0 и лог. 1:
а – положительная логика; б – отрицательная логика

При импульсном способе представления значению лог. 1, например, может быть поставлен в соответствие сам факт наличия импульса или его положительный перепад, а значению лог. 0 - отсутствие импульса или его отрицательный перепад.

Следует отметить, что при потенциальном способе представления значение логической переменной может быть определено в любой произвольный момент времени. При импульсном же представлении это может быть сделано только в строго определенных дискретные моменты времени. Другими словами, при потенциальном представлении возможен как синхронный, так и асинхронный съем информации. При импульсном же представлении возможен только синхронный съем информации. Это определяет преимущества и недостатки указанных методов представления логических переменных.

В настоящем лекционном курсе ограничимся рассмотрением базовых логических элементов, реализующих потенциальный способ представления логических переменных, т.к. он получил наиболее широкое практическое распространение.

На начальном этапе развития цифровой электроники было разработано большое число всевозможных схемотехнических решений, реализующих основные логические операции. Однако широкое практическое использование получили только некоторые из них. Эти элементы наилучшим образом сочетают ряд важных для потребителя и изготовителя характеристик.

В настоящее время при разработке ИС наибольшее распространение получили следующие базовые логические элементы (БЛЭ):

- транзисторно-транзисторная логика (ТТЛ);
- эмиттерно-связанная логика (ЭСЛ);
- интегрально-инжекционная логика (И²Л);
- логика на однотипных полевых транзисторах (n-МОП и p-МОП);
- логика на комплементарных полевых транзисторах (КМОП).

Первые три типа используют биполярные транзисторы, последние два – полевые.

Распространение нескольких типов логики, выполняющих одни и те же логические функции, объясняется различием их основных характеристик, что в зависимости от технических требований условий эксплуатации позволяет строить электронные средства с требуемыми параметрами.

Самыми распространенными на сегодняшний день являются ИС, реализующие ТТЛ и ее разновидности. Этот тип ИС, судя по прогнозам специалистов, останется наиболее массовым и в ближайшем будущем. Интегральные схемы данного типа обладают средним быстродействием ($F_{\text{max}} = 20 \dots 50$ МГц) и средней потребляемой мощностью.

Интегральные схемы, реализующие ЭСЛ, являются наиболее быстродействующими, но мощность, потребляемая этими элементами, превышает мощность элементов ТТЛ. Однако гибкость ее схемотехнических решений позволяет достаточно просто реализовывать сложные логические функции.

Базовые элементы И²Л были разработаны специально для использования в БИС. Их отличает высокая, недоступная для ТТЛ и ЭСЛ степень интеграции, пониженное напряжение питания, простота согласования с элементами ТТЛ и возможность регулировать в широких пределах быстродействие путем изменения потребляемой мощности.

Особенностью ИС, выполненных на полевых транзисторах, является малая потребляемая мощность. Однако по быстродействию эти элементы пока уступают логике на биполярных транзисторах. Данный тип логики также обладает высокой помехоустойчивостью. В сравнении с биполярными транзисторами полевые транзисторы имеют меньшие размеры и проще в изготовлении, что позволяет разместить на единице площади кристалла больше элементов. В соответствии с этими особенностями область их применения ограничивается функционально сложными устройствами невысокого быстродействия и малого токопотребления.

Прогнозы специалистов показывают, что по мере совершенствования технологии, направленной на повышение их быстродействия, этот класс ИС постепенно станет самым массовым. В первую очередь это касается ИС КМОП.

8.1. Основные требования к базовым логическим элементам

В реальных цифровых устройствах электрические сигналы, отображающие значения логических переменных, проходят через большое число последовательно соединенных однотипных электронных блоков, реализующих их преобразование в соответствии с аксиомами основных логических операций. Для надежного функционирования таких схем необходимо, чтобы каждый отдельный электронный блок, не зависимо от своего расположения в схеме и различных комбинаций входных сигналов и выходных нагрузок, безошибочно выполнял заданную логическую функцию. При этом должна гарантироваться передача неискаженного сигнала, соответствующего требуемому значению логической переменной. Возникшие в процессе преобразования искажения, во-первых, не должны влиять на работу устройства, если они лежат в пределах некоторых допустимых зон изменения этого сигнала, и, во-вторых, самоустраняться по мере дальнейшего преобразования сигнала.

Функциональная сложность реальных логических устройств, множество различных сочетаний входных сигналов и нагрузок, а также использование при их изготовлении гибридной и полупроводниковой технологий, не позволяют

рассчитывать на индивидуальную регулировку и настройку каждого отдельного элементарного электронного блока. Поэтому для надежной работы реальных цифровых устройств необходимо, чтобы все входящие в его состав элементарные электронные блоки обладали рядом фундаментальных свойств, к основным из которых относятся:

- совместимость уровней входных и выходных сигналов;
- нагрузочная способность;
- формирующее свойство (свойство квантования сигнала);
- помехоустойчивость.

Совместимость уровней входных и выходных сигналов. Для совместного использования большого числа логических элементов, выполненных, например, в виде ИС, должно гарантироваться согласование уровней сигналов, отображающих значения логических переменных. Это означает, что установившиеся отклонения выходного напряжения ЛЭ, вызванные действием различных внешних дестабилизирующих факторов и собственным разбросом параметров входящих в его состав элементов, не должны превышать некоторые заданные значения (рис. 8.2). Таким образом, должны выполняться неравенства

$$\begin{aligned} U_{\min}^1 &\leq U_{(t)}^1 \leq U_{\max}^1, \\ U_{\min}^0 &\leq U_{(t)}^0 \leq U_{\max}^0. \end{aligned} \quad (8.1)$$

где $U_{(t)}$ – текущее значение выходного напряжения ЛЭ. Следовательно, формирование на выходе сигнала,

$$U_{\max}^0 < U_{(t)} < U_{\min}^1 \quad (8.2)$$

не удовлетворяющего системе неравенств (8.1), недопустимо.

Для повышения надежности работы цифровых устройств допустимая область изменения входного сигнала ЛЭ, распознаваемая как соответствующий логический уровень, должна быть шире выходной

$$\begin{aligned} U_{\text{вых max}}^{1,0} &< U_{\text{вх max}}^{1,0}, \\ U_{\text{вых min}}^{1,0} &> U_{\text{вх min}}^{1,0}. \end{aligned} \quad (8.3)$$

Таким образом, допустимая область изменения входных сигналов ЛЭ должна включать область изменения выходных сигналов (рис. 8.2).

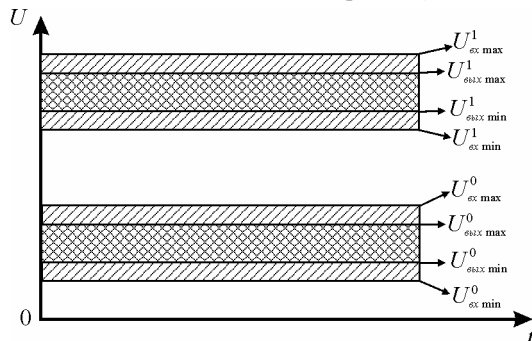


Рис. 8.2. Области отображения входных и выходных сигналов

Нагрузочная способность ЛЭ характеризует его способность получать сигнал от нескольких источников информации и одновременно быть источником информации для ряда других элементов. При этом его входные и выходные сигналы должны удовлетворять неравенствам (8.1) и (8.3). Для численной характеристики нагрузочной способности ЛЭ используются два коэффициента: коэффициент объединения по входу и коэффициент разветвления по выходу.

Коэффициент объединения по входу $K_{об}$ численно равен максимальному числу выходов одноступенчатых ЛЭ, которые могут быть подключены ко входу данного ЛЭ, не вызывая при этом искажений формы и амплитуды его сигнала, выходящих за границы зон отображения уровней лог. 0 и лог. 1.

Коэффициент разветвления по выходу $K_{раз}$ численно равен максимальному числу входов одноступенчатых ЛЭ, которые могут быть подключены к выходу данного ЛЭ, не вызывая при этом искажений формы и амплитуды его сигнала, выходящих за границы зон отображения уровней лог. 0 и лог. 1.

Типовые значения коэффициентов $K_{об} = 2 \dots 8$, $K_{раз} = 4 \dots 10$. В выпускаемых сериях ИС существуют элементы с повышенной нагрузочной способностью, для которых $K_{раз} = 20 \dots 30$.

Квантование сигнала. При прохождении электрического сигнала по цепочке последовательно включенных ЛЭ его амплитуда и форма не должны претерпевать недопустимых изменений. Для этого каждый ЛЭ должен обладать формирующими свойствами.

Сигнал, установившийся после прохождения по цепочке по последовательно включенных ЛЭ (рис. 8.3(a)), называют *стандартным*, или *асимптотическим*.

Формирующее свойство ЛЭ определяется видом его амплитудной передаточной характеристики (АПХ), под которой понимают зависимость его выходного напряжения от входного: $U_{\text{вых}} = f(U_{\text{вх}})$. Рассмотрим формирующее свойство ЛЭ с использованием амплитудной передаточной характеристики инвертирующего усилителя, приведенной на рис. 8.3 (б).

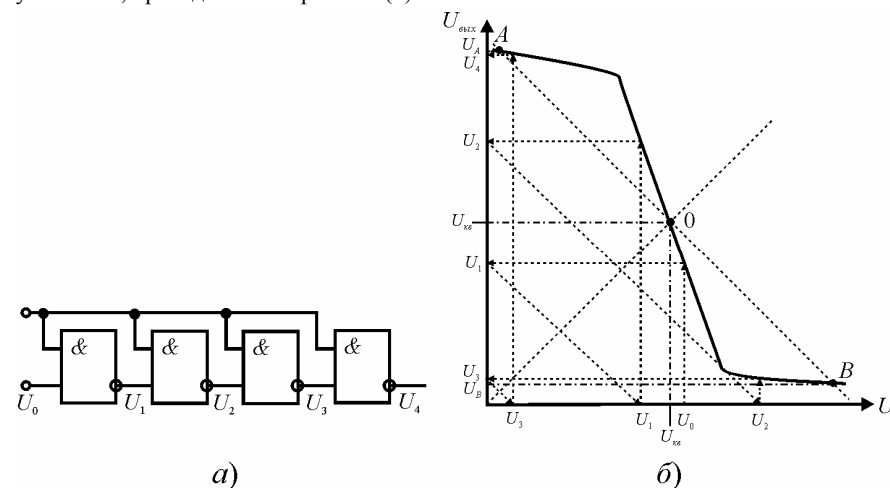


Рис. 8.3. Цепочка последовательно соединенных ЛЭ 2И-НЕ (а) и формирование сигнала на ее выходе (б)

Точка «0» данной характеристики соответствует пересечению характеристики ЛЭ с прямой $U_{\text{вых}} = U_{\text{вх}}$. Точки A и B получены как пересечения характеристики ЛЭ с прямой, перпендикулярной к зависимости $U_{\text{вых}} = U_{\text{вх}}$ в точке 0. Следует отметить, что полученные в результате такого построения значения напряжений U_A и U_B являются асимптотическими.

Предположим, что на вход первого ЛЭ цепочки (рис. 8.3(a)) подан сигнал U_0 , лежащий на участке характеристики, соответствующей работе этого элемента в качестве усилителя, т. е. в области, не удовлетворяющей неравенству (8.2). Используя приведенную на рис. 8.3(б) амплитудную передаточную характеристику, найдем значение сигнала U_4 , которое установится на выходе четвертого ЛЭ. При этом $U_4 > U_0$. Очевидно, что с увеличением числа последовательно включенных ЛЭ ($n \rightarrow \infty$) выходное напряжение U_n , снимаемое с выхода последнего в цепочке элемента, будет стремиться к значениям U_A или U_B .

Следовательно, исходный искаженный сигнал, пройдя через цепочку из последовательно включенных ЛЭ, стремится к своему асимптотическому значению.

Напряжение, соответствующее точке 0 амплитудной передаточной характеристики (АПХ) ЛЭ, называют напряжением порога квантования $U_{кв}$. Если $U = U_{кв}$, то сигнал, пройдя через цепочку последовательно включенных ЛЭ, строго говоря, не претерпит никаких изменений. Таким образом, $U_{кв}$ делит характеристику ЛЭ на две области, соответствующие зонам отображения сигналов лог. 0 и лог. 1. Если $U > U_{кв}$, то этот сигнал воспринимается как сигнал лог. 1, если $U < U_{кв}$ - то как сигнал лог. 0. Следует заметить, что точка 0 является точкой неустойчивого равновесия, и любое сколь угодно малое отклонение напряжения от значения $U_{кв}$ приведет к формированию на выходе цепочки элементов сигнала U_A или U_B .

Амплитудная передаточная характеристика реального ЛЭ существенно нелинейна, что обеспечивает быстрое формирование асимптотических значений логических сигналов.

Помехоустойчивость ЛЭ. Очевидно, что даже кратковременное искажение логических сигналов в силу формирующих свойств ЛЭ может привести к потере истинности получаемых результатов. Поэтому ЛЭ должны обладать высокой помехоустойчивостью.

Под помехоустойчивостью понимается свойство нечувствительности ЛЭ к отклонениям его входных сигналов от асимптотических значений. Эти отклонения обычно вызываются действием различных помех на логическое устройство. Степень их влияния во многом определяется типом используемых схмотехнических решений.

Помехи в цифровых устройствах носят, как правило, характер кратковременных импульсов. Принято различать внешние и внутренние помехи.

К первым относятся помехи, вызванные внешними воздействиями: электромагнитное воздействие промышленной сети электропередачи, силовых переключателей и электродвигателей, городского транспорта и т. п. Уменьшить влияние этих помех можно как конструктивными, так и схмотехническими средствами, например экранированием или соответствующим выбором уровней лог. 0 и лог. 1.

Ко вторым относятся помехи, амплитуда и длительность которых зависит от амплитуды и длительности перепадов напряжений на выходах самих ЛЭ, а также видом и длиной соединительных линий между ними.

Зона помехоустойчивости ЛЭ определяется по его АПХ. Уровень допустимой входной помехи для комбинационных устройств задается как разность между порогом квантования и соответствующим асимптотическим уровнем сигналов лог. 0 и лог. 1. В соответствии с этим различают уровни помехи по сигналам лог. 0 и лог. 1, которые находятся из соотношений:

$$\begin{aligned} U_{\text{помк}}^0 &= |U_{кв} - U_B|, \\ U_{\text{помк}}^1 &= |U_{кв} - U_A|. \end{aligned} \quad (8.4)$$

Если $U(t)_{\text{ном}} < U_{\text{номк}}$, то на выходе ЛЭ появится сигнал, амплитуда помехи в котором меньше, чем на входе, и в силу формирующих свойств при прохождении по цепочке последовательно включенных ЛЭ сигнал помехи быстро затухнет.

Последовательные устройства, как было показано ранее, содержат внутренние цепи обратной связи. Поэтому для их надежного функционирования необходимо, чтобы в процессе работы модуль коэффициента передачи разомкнутого контура не превышал единицу. В противном случае возможно лавинообразное переключение устройства в новое устойчивое состояние, что приведет к потере информации. Поэтому в таких устройствах допустимая амплитуда помехи меньше, чем в комбинационных, и определяется выражениями (рис. 8.4):

$$\begin{aligned} U_{\text{помл}}^0 &= |U_{\text{л}}^0 - U_B|, \\ U_{\text{помл}}^1 &= |U_{\text{л}}^1 - U_A|, \end{aligned} \quad (8.5)$$

где $U_{\text{л}}^0$ и $U_{\text{л}}^1$ - входные напряжения, соответствующие точкам характеристики, для которых $dU_{\text{вых}}/dU_{\text{вх}} = -1$ (см. рис. 8.4).

При определении реальной зоны помехоустойчивости необходимо учитывать существующий разброс АПХ ЛЭ. Последнее еще больше уменьшает допустимую амплитуду сигнала помехи.

Следует отметить, что допустимая амплитуда помехи также зависит от ее длительности. Эту зависимость определяет *характеристика импульсной помехоустойчивости*, связывающая допустимую амплитуду помехи с ее длительностью (рис. 8.5).

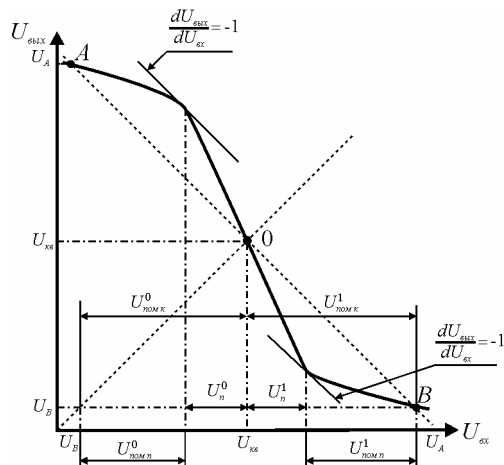


Рис. 8.4. Зона импульсной помехоустойчивости ЛЭ

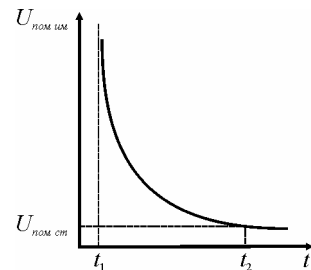


Рис. 8.5. Характеристика импульсной помехоустойчивости

Из приведенной зависимости следует, что для ЛЭ существует некоторая граничная длительность $t_{ном} = t_2$, начиная с которой допустимая амплитуда сигнала помехи начинает увеличиваться относительно уровней, рассчитанных с использованием выражений (8.4) и (8.5). При очень малых длительностях $t_{ном} < t_1$ ЛЭ перестает быть чувствительным к сигналу помехи любой амплитуды.

8.2 Базовые логические элементы транзисторно-транзисторной логики (ТТЛ)

Схемотехнически большинство ИС, входящих в состав серий ТТЛ, выполнено на основе комбинации двух базовых схем: элемента И-НЕ (Штрих Шеффера) и расширителя по ИЛИ.

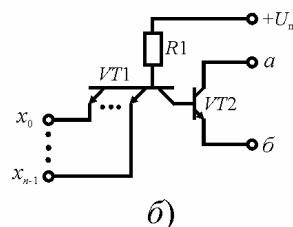
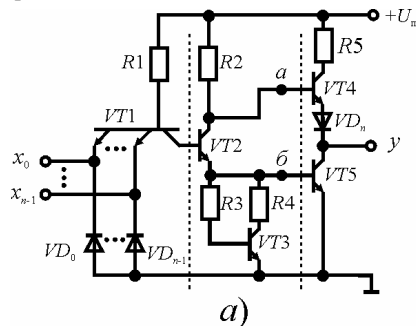


Рис. 8.6. Принципиальная электрическая схема ЛЭ И-НЕ (а), принципиальная электрическая схема расширителя по ИЛИ серии ТТЛ (б)

Элемент И-НЕ (рис. 8.6 (а)) может быть представлен последовательным соединением трех каскадов:

- входного многоэмиттерного транзистора $VT1$ с резистором $R1$ и диодами $VD_0 - VD_{n-1}$, реализующих логическую операцию И;

- фазорасщепителя на транзисторе $VT2$, резисторе $R2$ и цепи нелинейной коррекции $R3, R4$ и $VT3$;
- двухтактного выходного усилителя на транзисторах $VT4$ и $VT5$, резисторе $R5$ и диоде VD_n .

Расширитель по ИЛИ (рис. 8.6 (б)) по сути, повторяет первые два каскада элемента И-НЕ и содержит входной многоэмиттерный транзистор $VT1$ с резистором $R1$ и транзистор $VT2$ фазорасщепителя.

Объединение элементов И-НЕ и расширителя по ИЛИ путем соединения точек a и b позволяет получить ЛЭ, реализующий последовательность операций И-ИЛИ-НЕ.

Объединение элементов И-НЕ и расширителя по ИЛИ путем соединения точек a и b позволяет получить ЛЭ, реализующий последовательность операций И-ИЛИ-НЕ.

Рассмотрим работу элемента И-НЕ (рис. 8.6(а)). Предположим, что хотя бы один вход элемента x_0, \dots, x_{n-1} непосредственно подключен к общей шине, т. е. на него подано напряжение лог. 0. В этом случае многоэмиттерный транзистор $VT1$ оказывается насыщенным током, протекающим от источника питания через резистор $R1$. Напряжение на его коллекторе будет мало отличаться от нулевого. Поэтому транзистор $VT2$ фазорасщепителя оказывается запертым. Так как эмиттерный ток транзистора $VT2$ практически равен нулю, будет заперт и транзистор $VT5$. Ток, протекающий через резистор $R2$, втекает в базу транзистора $VT4$, насыщая его. Поэтому напряжение, присутствующее на выходе у ЛЭ, близко к напряжению питания и определяется выражением

$$U_{вых}^1 = U_{II} - I_{вых} R5 - U_{КЭVT4} - U_{VD_n}. \quad (8.6)$$

Таким образом, при наличии на любом из входов схемы напряжения низкого уровня напряжение на его выходе будет иметь высокий уровень.

Предположим теперь, что на все входы ЛЭ подан высокий уровень напряжения. В этом случае все эмиттерные переходы многоэмиттерного транзистора $VT1$ оказываются запертыми. При этом его коллекторный переход сместится в прямом направлении, и по цепи, состоящей из резистора $R1$, коллекторного перехода транзистора $VT1$ и последовательно соединенных эмиттерных переходов транзисторов $VT2$ и $VT5$, будет протекать ток. Этот ток насытит транзисторы $VT2$ и $VT5$, и на выходе у ЛЭ установится низкое напряжение, численно равное напряжению насыщения транзистора $VT5$:

$$U_{вых}^0 = U_{КЭVT5} = I_{вых} r_{выхVT5}. \quad (8.7)$$

Так как транзистор $VT2$ насыщен, присутствующее на его коллекторе напряжение оказывается недостаточным для прямого смещения двух последовательно включенных $p-n$ -переходов (эмиттерный переход $VT4$ и диод VD_n). Транзистор $VT4$ будет заперт.

Таким образом, если на всех входах схемы присутствует высокое напряжение, на выходе БЛЭ будем иметь напряжение низкого уровня.

Из приведенного описания следует, что в общем случае параллельное соединение нескольких выходов таких элементов недопустимо, так как формирование на них сигналов различных уровней приведет к выходу из строя транзисторов выходного усилителя мощности. Параллельно можно соединять выходы только синхронно работающих элементов.

Для положительной логики описанный алгоритм работы соответствует определению операции И-НЕ:

$$y = \overline{x_0 x_1 \dots x_{n-1}}. \quad (8.8)$$

Рассмотрим более подробно назначение отдельных элементов схемы.

Выполнение выходного каскада по двухтактной схеме позволяет решить одновременно две задачи:

- повысить быстродействие элемента. Нагрузка БЛЭ, как правило, носит емкостной характер, и применение двухтактного выходного каскада позволяет увеличить ток перезаряда емкости нагрузки;
- снизить потребление. В установленном режиме лог. 0 через выходной каскад протекает только ток нагрузки.

Во входной цепи многоэмиттерного транзистора $VT1$ включены дополнительные диоды $VD_0 - VD_{n-1}$, которые защищают элемент от появления на его входе недопустимых напряжений обратной полярности.

Нелинейная цепь коррекции $R3$, $R4$ и $VT3$ позволяет увеличить быстродействие элемента и приблизить его АПХ к прямоугольной. Последнее улучшает формирующие свойства элемента. Рассмотрим ее работу. Принцип действия данной цепи основан на зависимости ее сопротивления от состояния транзистора $VT5$. Если этот транзистор заперт, то транзистор нелинейной цепи коррекции $VT3$ также заперт. Суммарное сопротивление цепи, шунтирующей эмиттерный переход транзистора $VT5$, в основном определяется сопротивлением резистора $R3$, которое велико. Поэтому на начальном этапе формирования на выходе элемента напряжения лог. 0 весь эмиттерный ток транзистора $VT2$ втекает в базу транзистора $VT5$, что форсирует его включение. После включения $VT5$ насыщается и $VT3$, шунтируя эмиттерный переход транзистора $VT5$ низкоомным сопротивлением резистора $R4$. Это, во-первых, уменьшает степень насыщения транзистора $VT5$ и, во-вторых, при последующем выключении увеличивает ток, удаляющий из базовой области этого транзистора избыточный заряд неосновных носителей. Оба этих фактора способствуют снижению времени рассасывания транзистора $VT5$, что повышает быстродействие элемента.

Следует отметить, что в момент переключения БЛЭ в его выходной цепи протекает так называемый «сквозной ток», обусловленный тем, что в течение интервала рассасывания запираемого транзистора оба транзистора выходного двухтактного усилителя оказываются насыщенными. Это приводит к тому, что ток потребления элемента имеет явно выраженный импульсный характер. Поэтому с увеличением частоты переключения среднее значение тока, потребляемого элементом, растет. Растет и его суммарная потребляемая мощность. Кроме этого, протекание импульсов тока за счет действия индуктивности соединительных проводов ($L_{пр}$ на рис. 8.7) может привести к появлению ложных срабатываний соседних элементов (внутренние помехи).

Для ограничения величины «сквозного тока» в коллекторную цепь $VT4$ включен резистор $R5$. Однако чрезмерное увеличение сопротивления этого резистора, во-первых, увеличивает мощность, рассеиваемую в элементе, и, во-вторых, уменьшает его нагрузочную способность. Поэтому для исключения действия помех шины питания должны выполняться с малой собственной индуктивностью и по всей длине шунтироваться дополнительными конденсаторами $C_{доп}$ с малой паразитной индуктивностью (как правило, керамическими) (рис. 8.7). Использование такого технического решения позволяет свести к минимуму действие на устройство внутренних помех.

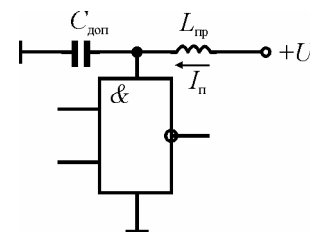


Рис. 8.7. Шунтирование шины питания БЛЭ ТТЛ дополнительными конденсаторами

Следует отметить еще одну особенность применения БЛЭ ТТЛ. Согласно приведенному алгоритму работы, если вход элемента остается неподключенным к источнику сигнала, можно считать, что на него подан сигнал лог. 1. Однако на практике неиспользуемые входы элементов ТТЛ рекомендуется не оставлять свободными, а через дополнительный резистор $R_{доп}$ подключать к выводу $+U_{п}$. В противном случае, элемент будет обладать большим входным сопротивлением в состоянии лог. 1, что резко увеличивает вероятность воздействия на него помех, и снижает надежность работы логического устройства. Обычно один резистор используется для подключения нескольких входов ЛЭ. В этом случае его сопротивление может быть найдено из условия

$$R_{доп} \leq \frac{U_{п} - U_{вх\min}}{nI_{вх\max}},$$

где n – число входов БЛЭ, подключаемых к резистору.

Интегральные параметры БЛЭ определяют уровень развития и совершенства технологии и схемотехники, а также качество ИС. Рассмотрим два из них – энергию переключения и степень (уровень) интеграции.

Энергия переключения определяется как произведение средней мощности, потребляемой ЛЭ, на среднее время задержки распространения импульса

$$W_{\tau} = P_{потр} t_{з\text{ ср}}, \quad (8.9)$$

обычно она измеряется в пикоджоулях (мВт·нс).

По мере совершенствования схемотехники и технологии энергия W_{τ} уменьшается. По имеющейся на сегодняшний день статистике уменьшение происходит со скоростью полтора порядка за десять лет.

Уровень W_{τ} свидетельствует о том, что на данном этапе развития технологии и схемотехники увеличение быстродействия ИС всегда связано с увеличением потребляемой мощности и, наоборот, снижение потребляемой мощности одновременно ведет к падению быстродействия.

Степень интеграции N характеризует число элементарных ЛЭ (обычно двухвыходовых), расположенных на одном кристалле ИС. Численное значение N определяется выражением

$$N = \lg K, \quad (8.10)$$

где K – количество элементов в ИС. В настоящий момент существуют ИС 6-й степени интеграции.

Все ИС ТТЛ, выпускаемые отечественной промышленностью в настоящий момент можно разделить на следующие группы:

- стандартные – серия 155;

- быстродействующие с диодами Шотки – серии 530, 531, 1531;
 - маломощные с диодами Шотки – серии 533, 555, 1533.
- Элементы всех этих серий практически выполнены по единой схеме. Основное различие данных серий заключается в их быстродействии и потребляемой мощности. Типовые параметры БЛЭ различных серий ТТЛ приведены в таблице 8.1.

Напряжение всех БЛЭ ТТЛ равно $+5В \pm 5\%$ или $+5В \pm 10\%$ в зависимости от типа исполнения.

Таблица 8.1

Типовые параметры БЛЭ различных серий ТТЛ

Серия ИС	t_3 ср (нс)	$P_{\text{потр}}$ (мВт)	$U_{\text{вых}}^1$ (В)	$U_{\text{вых}}^0$ (В)	$I_{\text{вых}}^1$ (мА)	$I_{\text{вых}}^0$ (мА)
K155	10	10	2,4	0,4	-0,4	16
K531	3	20	2,7	0,5	-1,0	20
K555	10	2,0	2,7	0,5	-0,4	8,0
Серия ИС	$U_{\text{ex max}}^0$ (В)	$U_{\text{ex min}}^1$ (В)	$I_{\text{вых}}^0$ (мА)	$I_{\text{вых}}^1$ (мА)	$K_{\text{раз}}$	f_{max} (МГц)
K155	0,8	2,0	-1,6	0,04	10	10
K531	0,8	2,0	-2,0	0,05	10	50
K555	0,8	2,0	-0,36	0,02	10	10

8.3. Базовые логические элементы эмиттерно-связанной логики (ЭСЛ)

Причиной появления БЛЭ ЭСЛ явилось желание повысить быстродействие цифровых устройств. Это желание привело к использованию в них совершенно отличного от ТТЛ схемотехнического решения. Как было показано выше, основными причинами инерционности ключей, выполненных на биполярных транзисторах, являются время рассасывания неосновных носителей из его базовой области и постоянная времени перезаряда выходной емкости. Если время рассасывания транзистора при работе последнего в активной области может быть полностью исключено, то от влияния постоянной времени перезаряда выходной емкости транзистора полностью избавиться не представляется возможным. Это влияние лишь можно уменьшить путем увеличения коллекторного тока транзистора, как это сделано в БЛЭ ТТЛ серии 531. При неизменном постоянном токе перезаряда выходной емкости транзистора длительность его перехода из состояния, классифицируемого как лог. 0, в состояние, классифицируемое как лог. 1, и обратно может быть уменьшена только за счет уменьшения логического перепада. Такое решение позволяет повысить быстродействие. Однако дается оно за счет снижения помехоустойчивости БЛЭ, что требует создания схем, при прочих равных условиях, менее подверженных воздействию помех. Этот принцип и используется при построении БЛЭ ЭСЛ.

Основой БЛЭ ЭСЛ является токовый ключ, выполненный на транзисторах (рис. 8.8). На базу одного из них, например VT_2 , подано некоторое постоянное опорное напряжение $U_{\text{оп}}$. Изменение напряжения, подаваемого на вход x_0 ниже или выше $U_{\text{оп}}$, приводит к перераспределению постоянного тока I_3 , заданного токостабилизирующим резистором R_3 , между транзисторами VT_1 и VT_2 . При этом транзисторы не попадают в режим насыщения и, следовательно, в ключе принципиально отсутствует интервал рассасывания их неосновных носителей. Таким образом, особенностью БЛЭ ЭСЛ является постоянство потребляемого тока не зависимо от значения выходного сигнала ключа.

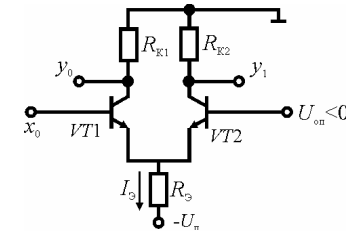


Рис. 8.8. Принципиальная электрическая схема токового ключа БЛЭ ЭСЛ

Эта особенность выгодно отличает БЛЭ ЭСЛ от БЛЭ ТТЛ, в котором в момент переключения ток, потребляемый элементом, резко возрастает, создавая внутренние помехи, ухудшающие помехозащищенность цифрового устройства.

Нетрудно заметить, что общей шиной для элемента является шина $+U_{\text{п}}$, в результате чего все потенциалы точек схемы отрицательны относительно общей шины. Однако в схеме токового ключа так же, так и в схемах ТТЛ, реализован принцип положительной логики, при котором большему выходному напряжению соответствует сигнал лог. 1, а меньшему - сигнал лог. 0 (см рис. 8.1).

Быстродействие токового ключа весьма велико, так как, во-первых, транзисторы принципиально не заходят в область насыщения и, во-вторых, мал логический перепад напряжений между значениями лог. 0 и лог. 1. Последнее реализовано выбором малых сопротивлений резисторов $R_{к1}$ и $R_{к2}$ схемы, что крайне полезно с точки зрения уменьшения постоянной времени перезаряда выходной емкости транзистора.

С токового ключа снимаются одновременно два сигнала - прямой и инверсный, связанные с сигналом x_0 на входе схемы соотношениями:

$$\begin{aligned} y_1 &= x_0, \\ y_0 &= \bar{x}_0. \end{aligned} \quad (8.11)$$

Следует отметить, что схемотехнически токовый ключ на рис. 8.8 повторяет схему дифференциального усилителя постоянного тока.

Выходное напряжение, снимаемое с выходов y_1 и y_0 , всегда больше $U_{\text{оп}}$, так как транзисторы VT_1 и VT_2 всегда работают в ненасыщенном режиме. Поэтому непосредственное последовательное включение нескольких таких ключей невозможно. Для этого необходим согласующий каскад. В качестве такого согласующего каскада используются схемы эмиттерных повторителей, включенных между выходами токового ключа и выходами элемента.

Полная схема БЛЭ, выполненного на основе токового ключа, приведена на рис. 8.9 (элемент серии 500). Базовый элемент получен путем замены входного транзистора VT_1 токового ключа группой параллельно включенных транзисторов VT_1-VT_n .

Функционально схему БЛЭ можно разбить на три узла:

- токовый ключ на транзисторе VT_1-VT_{n+1} и резисторе R_{n+2} ;
- источник эталонного напряжения, включающий параметрический стабилизатор на элементах R_{n+4} , $VD1$, $VD2$, R_{n+5} и эмиттерный повторитель на VT_{n+2} и R_{n+3} ;
- выходные эмиттерные повторители на транзисторах VT_{n+3} и VT_{n+4} .

Цепь нагрузок транзисторов VT_{n+3} и VT_{n+4} вынесена из ИС, что способствует снижению рассеиваемой в ней мощности и расширению функциональных

возможностей. Диоды $VD1, VD2$ в задающей цепи источника эталонного напряжения служат для термокомпенсации напряжения $U_{оп}$.

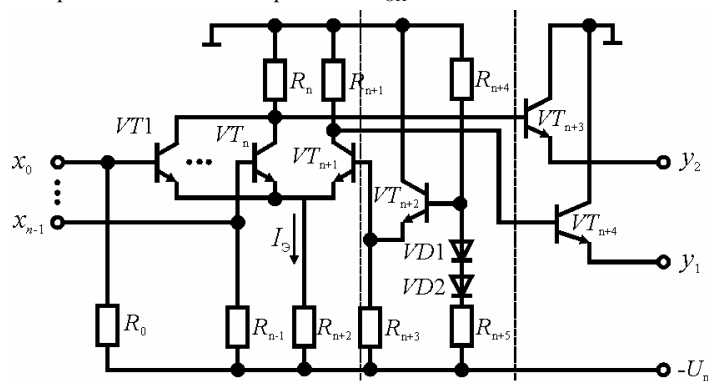


Рис. 8.9. Принципиальная электрическая схема БЛЭ ЭСЛ ИС серии 500

Принцип действия и функциональные возможности БЛЭ ЭСЛ. Предположим, что на все входы схемы рис. 8.9 x_0, \dots, x_{n-1} подано напряжение, близкое к $-U_n$. Тогда транзисторы $VT1 - VT_n$ будут заперты. Весь ток резистора R_{n+2} протекает через транзистор VT_{n+1} , к выводу базы которого приложено напряжение $U_{оп}$. Этот транзистор поддерживается в активном режиме работы за счет действия глубокой последовательной отрицательной обратной связи по току. Если не учитывать обратные токи коллекторных переходов транзисторов $VT1 - VT_n$, через резистор R_n протекает только базовый ток транзистора VT_{n+2} выходного эмиттерного повторителя. Следовательно, напряжение на выходе близко к нулевому

$$U_{y2} = -I_{БVT_{n+3}} R_n - U_{БЭVT_{n+3}}. \quad (8.12)$$

Численно напряжение U_{y2} примерно равно $-0,9$ В.

Через резистор R_{n+1} кроме базового тока транзистора VT_{n+4} протекает ток $I_{КVT_{n+1}}$, примерно равный I_3 . Эти токи создают на резисторе R_{n+1} падение напряжения, равное

$$U_{R_{n+1}} = (I_{К_{n+1}} + I_{Б_{n+1}}) \cdot R_{n+1} = \left[\frac{(I_3 h_{21Э})}{(h_{21Э} + 1)} + I_{Б_{n+4}} \right] \cdot R_{n+1} \approx I_3 R_{n+1}.$$

Это напряжение преобразуется выходным эмиттерным повторителем на транзисторе VT_{n+4} в выходное напряжение U_{y1} , определяемое выражением

$$U_{y1} \approx -U_{БЭVT_{n+4}} - I_3 R_{n+1}. \quad (8.13)$$

Численно напряжение U_{y1} примерно равно $-1,7$ В.

Если хотя бы на один из входов схемы x_0, \dots, x_{n-1} подано напряжение, превышающее по уровню $U_{оп}$ ($-1,3$ В), соответствующий транзистор VT перейдет в активный режим работы. Его ток будет равен току I_3 , что приведет к смене уровней выходного напряжения

$$\begin{aligned} U_{y2} &= -U_{БЭVT_{n+3}} - I_3 R_n, \\ U_{y1} &= -I_{БVT_{n+4}} R_{n+4} - U_{БЭVT_{n+4}}. \end{aligned} \quad (8.14)$$

Из сказанного следует, что рассмотренная схема реализует по выводу y_2 операцию ИЛИ-НЕ, а по выводу y_1 – операцию ИЛИ

$$\begin{aligned} y_1 &= (x_0 + x_1 + \dots + x_{n-1}), \\ y_2 &= \overline{(x_0 + x_1 + \dots + x_{n-1})}. \end{aligned} \quad (8.15)$$

Резисторы $R_0 - R_{n-1}$, включенные между базами транзисторов $VT1 - VT_n$ и выводом $-U_n$, обеспечивают запертое состояние этих транзисторов при отсутствии входного сигнала. Это позволяет не беспокоиться о подключении неиспользуемых входов ИС к выводам источника питания.

Особенностью схемотехнического построения элементов ЭСЛ является использование для подключения общей шины собственно токового переключателя и выходных эмиттерных повторителей различных выводов ИС. Потребляемый ток, протекающий в этих цепях, имеет качественно различный характер. Как было отмечено ранее, в принципе работы токового ключа заложено потребление принципиально постоянного тока, так как его работа связана с перераспределением тока эмиттерного резистора R_3 (см. рис. 8.8). Эмиттерные же повторители потребляют импульсный ток. К тому же для улучшения частотных свойств сопротивления резисторов, подключаемых к выводам y_1 и y_2 ИС, выбираются весьма малыми ($R_{внеш} = 75 \dots 100$ Ом). Поэтому совместное питание этих цепей из-за малой величины логического перепада может приводить к ложному срабатыванию соседних элементов, т. е. к появлению сбоев при обработке информации (внутренних помех). Разделение цепей питания позволяет устранить этот недостаток.

Раздельное питание токовых ключей и выходных эмиттерных повторителей позволяет дополнительно решить задачу снижения мощности, рассеиваемой в реальной аппаратуре. Так как выходное напряжение элемента лежит в диапазоне $-0,9 \dots -1,7$ В, то для питания внешних резисторов может быть использовано напряжение, не превышающее 2 В. Такое решение при малых сопротивлениях $R_{внеш}$ позволяет значительно уменьшить бесполезные потери мощности.

Способы повышения быстродействия БЛЭ ЭСЛ. Инерционность, обусловленная временем рассасывания биполярного транзистора, в БЛЭ ЭСЛ отсутствует, поэтому повышение быстродействия может быть достигнуто уменьшением входной емкости и суммарной емкости коллекторного узла схемы. Данный способ реализован в так называемых элементах Э²СЛ. Идея их построения состоит в отказе от применения эмиттерных повторителей в выходной цепи элемента и переносе их в его входную цепь. При этом емкость нагрузки эмиттерного повторителя, т.е. собственно входная емкость токового переключателя при пересчете в его входную цепь, уменьшается в $h_{21Э}$ раз ($C_{вх} = C_n/h_{21Э}$). Элемент такого типа обладает уменьшенной энергией переключения.

Таблица 8.2

Основные параметры БЛЭ ЭСЛ

Тип элемента	Серия	$t_{зр\ ср}$ (нс)	$P_{потр}$ (мВт)	$U_{пом}$ (мВ)	$U_{лп}$ (В)	U_n (В)
ЭСЛ	500	2	25	130...160	0,8	5,2
	1500					
Э ² СЛ	-	0,5...0,8	5	130...160	0,6	5,2
МЭСЛ	-	0,5...0,8	5	40...50	0,4	2...3

Другим способом повышения быстродействия является уменьшение уровня логического перепада и напряжения питания. Это решение использовано при разработке так называемой малосигнальной ЭСЛ (МЭСЛ). Однако так как при этом сильно падает помехоустойчивость БЛЭ, данное решение находит применение только при разработке СБИС. При этом входные и выходные сигналы самой ИС находятся на стандартном уровне ТТЛ. Типовые характеристики БЛЭ ЭСЛ различных типов приведены в табл. 8.2.

8.4. Базовые логические элементы на МДП-транзисторах

В 1962 г. с использованием планарной технологии был разработан МДП-транзистор с изолированным затвором. Это положило начало новому схмотехническому направлению – разработке ИС МДП. В первую очередь это были логические ИС и ИС запоминающих устройств.

Широкому распространению МДП-схмотехники способствовали следующие ее особенности:

- более простая технология изготовления (короче технологический цикл изготовления), что способствует повышению процента выхода годных изделий;
- меньшие геометрические размеры прибора и более простые технические решения, что при одинаковой с биполярной ИС площади кристалла позволяет разместить на нем более сложную по выполняемым функциям схему или при одинаковой функциональной сложности получить меньшую площадь кристалла, что также способствует повышению выхода годных изделий (меньше влияние дефектов исходного полупроводника).

Вместе с тем при разработке ИС МДП приходится учитывать следующие особенности данного класса приборов:

1. МДП-транзистор обладает худшими ключевыми свойствами по сравнению с биполярным транзистором. Это выражается в большем выходном сопротивлении и, следовательно, большем остаточном напряжении во включенном состоянии;
2. производная выходной характеристики МДП-транзистора dI_C/dU_{CH} при малых напряжениях U_C имеет явно выраженную зависимость от напряжения $U_{ЗИ}$ (рис. 8.10 (а)), в то время как аналогичная производная для выходных характеристик биполярного транзистора практически постоянна (рис. 8.10 (б)). Вследствие этого остаточное напряжение на включенном МДП-транзисторе сильно зависит от управляющего напряжения. Применительно к логическим схемам это означает сильную зависимость напряжения лог. 0 от напряжения лог. 1. Ослабить эту зависимость можно уменьшением абсолютного тока стока во включенном состоянии, что требует использования в ключе на МДП-транзисторе высокоомной нагрузки;
3. необходимость применения собственной высокоомной нагрузки при фиксированной емкости (входной емкости аналогичного элемента) увеличивает постоянную времени, определяющую длительности фронта и спада выходного напряжения ЛЭ. Это приводит к падению его быстродействия;
4. в силу рассмотренных выше особенностей (нестабильность выходных логических уровней) для обеспечения достаточной помехоустойчивости ЛЭ на МДП-транзисторах должны работать при больших величинах логического перепада. Это также способствует падению быстродействия элемента.

Следует отметить, что необходимость применения высокоомной нагрузки имеет и положительное свойство, выражающееся в уменьшении рассеиваемой в ключе на МДП-транзисторе мощности.

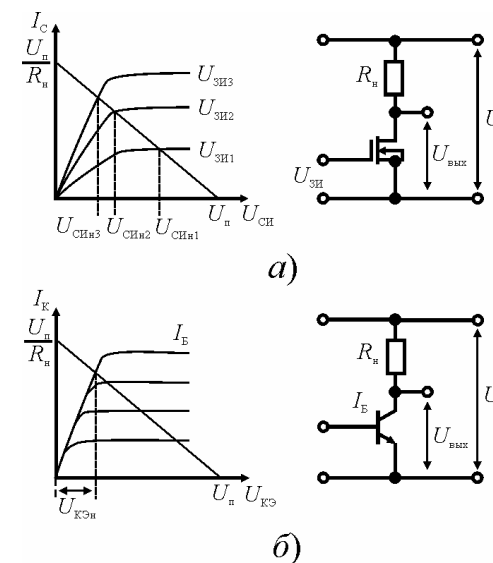


Рис. 8.10. Статические характеристики МДП (а) и биполярного (б) транзисторных ключей

Схмотехника базовых логических элементов nМОП и pМОП типов. Схмотехнические решения, используемые при построении ИС МДП, направлены на устранение вышеописанных недостатков элементарного ключа. Поэтому при построении ИС схема ключа с нагрузочным резистором не используется. Широкое применение нашла схема ключа с нагрузочным МДП-транзистором. Это дополнительно позволяет упростить технологию изготовления ИС, так как из схемы исключаются все пассивные элементы (резисторы) и она строится только на однотипных элементах - МДП-транзисторах.

В зависимости от типа используемого транзистора различают ИС nМОП и pМОП типов. Рассмотрим построение БЛЭ с использованием nМОП-транзисторов. На рис. 8.11 (а, б) приведены принципиальные электрические схемы двухвходовых БЛЭ, реализующих операции 2И-НЕ и 2ИЛИ-НЕ.

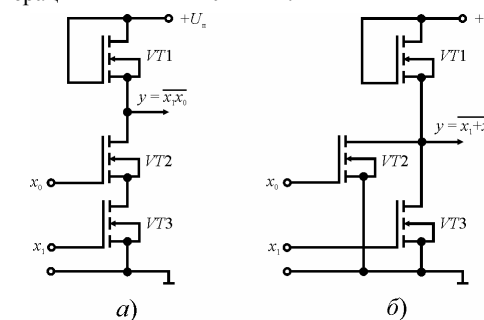


Рис. 8.11. Принципиальные электрические схемы БЛЭ на МДП-транзисторах, реализующих операции 2И-НЕ (а) и 2ИЛИ-НЕ (б)

Обе схемы содержат по три транзистора, из которых $VT1$ выполняет роль активной нагрузки, а $VT2$ и $VT3$ являются собственно транзисторными ключами, реализующими логические операции. В схеме рис. 8.11(а), реализующей логическую операцию И-НЕ, транзисторы $VT2$ и $VT3$ включены последовательно. Поэтому для появления на выходе схемы низкого напряжения на затворы обоих транзисторов необходимо подать высокое напряжение, достаточное для протекания тока активной нагрузки. В схеме рис. 8.11(б) транзисторы $VT2$ и $VT3$ включены параллельно. Поэтому при подаче на затвор любого из них высокого напряжения на выходе будет сформировано напряжение низкого уровня.

Увеличение числа входных переменных элемента требует увеличения количества последовательно или параллельно включенных транзисторов. Поэтому, используя описанный принцип с минимальными схмотехническими затратами, можно легко построить логический элемент с требуемым числом входов.

Схмотехника базовых логических элементов КМОП-типа. Увеличение быстродействия ИС МДП требует увеличения токов перезаряда емкостей нагрузки. Однако это ограничивается ростом потребляемой мощности и увеличением нестабильности выходных логических уровней. Преодолеть указанное противоречие можно либо технологическим путем, создавая транзисторы с меньшей входной емкостью, либо схмотехническим путем, применяя схему ключа на транзисторах с каналами различного типа (комплементарные транзисторы). Подобные ключи, с одной стороны, позволяют значительно увеличить токи перезаряда емкости нагрузки, а с другой - максимально уменьшить мощность, рассеиваемую в элементе. Заметим, что ключ на комплементарных транзисторах при правильном выборе параметров входящих в него элементов в статическом режиме работы практически не потребляет мощность от источника питания.

Потребляемая элементом мощность в статическом режиме тождественно равна мощности, отдаваемой им в нагрузку. А так как нагрузкой элемента являются входные цепи аналогичных элементов, носящие чисто емкостный характер, то мощность, отбираемая от источника питания, расходуется только в динамическом режиме на перезаряд этой емкости, т. е. имеет минимально возможное значение.

Схмотехнически БЛЭ КМОП-типа повторяют схемы элементов n МОП и p МОП-типов. Отличие состоит в том, что всегда используются пары транзисторов. При этом если для реализации заданной логической функции транзисторы с каналом n -типа включаются последовательно, то парные им транзисторы p -типа включаются параллельно, и наоборот. В качестве примера на рис. 8.12 приведены принципиальные электрические схемы, реализующие логические операции 2И-НЕ и 2ИЛИ-НЕ. Для упрощения на приведенных схемах не показаны элементы входных и выходных цепей ключа.

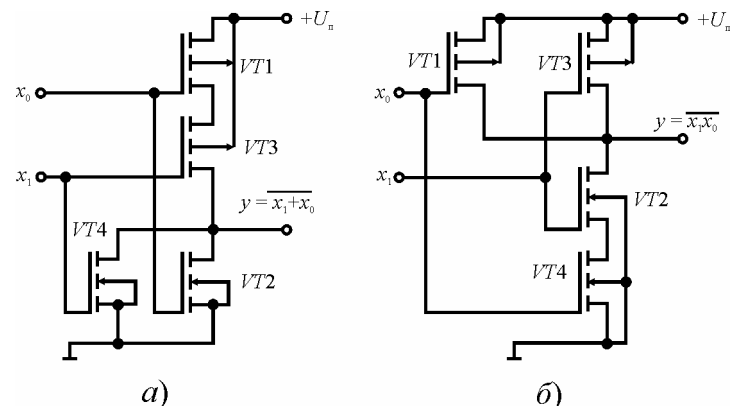


Рис. 8.12. Принципиальные электрические схемы БЛЭ КМОП-типа, реализующие функции 2И-НЕ (а) и 2ИЛИ-НЕ (б)

К особенностям схем БЛЭ следует также отнести отсутствие дополнительного нагрузочного транзистора. Его роль выполняет один из транзисторов ключа.

Анализ схем позволяет сделать важный практический вывод о том, что аналогично БЛЭ ТТЛ для БЛЭ КМОП параллельное включение нескольких их выходов запрещено.

Следует также отметить, что КМОП-элементы обладают высокой помехоустойчивостью — до 40% напряжения питания.

В таблице 8.3 приведены наиболее важные параметры БЛЭ КМОП.

Таблица 8.3

Основные параметры БЛЭ КМОП							
$U_{\text{вых min}}^1$ (В)	$U_{\text{вых min}}^0$ (В)	$t_{\text{зр ср}}$ (нс)	F_{max} (МГц)	$I_{\text{потр}}$ (мкА)	$U_{\text{п}}$ (В)	$K_{\text{раз}}$	$C_{\text{вх}}$ (пФ)
8	0,3	30 ($C_{\text{н}}=15$ пФ) 100 ($C_{\text{н}}=100$ пФ)	1	0,1	3 ... 15 (9)	50	5

8.5. Базовые логические элементы интегрально-инжекционной логики (И²Л)

Для повышения технологичности изготовления желательно при разработке ИС применять схмотехнические решения, использующие только однотипные элементы, например, транзисторы. Этот путь, как было показано ранее, реализован в ИС МДП, что наряду с другими достоинствами является причиной их широкого распространения. Однако, как уже отмечалось, ключ на биполярных транзисторах на сегодняшний день обладает лучшими как ключевыми, так и частотными свойствами. Это является предпосылкой к постоянному поиску новых схмотехнических решений для реализации биполярных ИС. Такой поиск привел к почти одновременной разработке фирмами Philips и IBM элемента интегральной инжекционной логики (И²Л). Срез топологии и соответствующая ему принципиальная электрическая схема БЛЭ И²Л приведены рис. 8.13 (а, б).

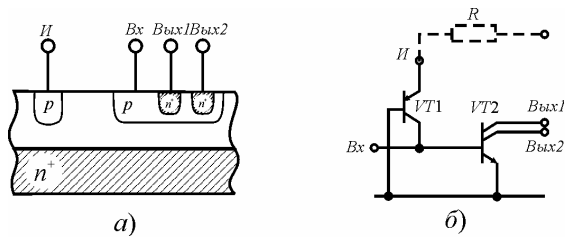


Рис. 8.13. Срез топологии (а) и принципиальная электрическая схема БЛЭ (б) I^2L

Особенностью элементов I^2L является:

- 1) отсутствие резисторов, что резко упрощает технологию производства ИС;
- 2) использование токового принципа питания, при котором в ИС задается не напряжение, а ток, который непосредственно инжектируется в область полупроводника, образующего структуру одного из транзисторов;
- 3) пространственное совмещение в кристалле полупроводника областей, функционально принадлежащих различным транзисторам. При этом структура располагается как по горизонтали (планарно), так и по вертикали. Такое решение позволяет отказаться от применения специальных решений для отделения областей, принадлежащих различным элементам, как это необходимо делать в элементах ТТЛ и ЭСЛ;
- 4) малое значение логического перепада, что позволяет максимально увеличить быстродействие элемента.

В приведенной схеме (рис. 8.13(б)) многоколлекторный транзистор $VT2$ выполняет функцию инвертирования входного сигнала, а транзистор $VT1$ - генератора (инжектора) базового тока транзистора $VT2$. К особенностям элемента следует отнести и постоянство тока инжектора во всех режимах работы элемента. Ток инжектора задается резистором R , который, как правило, выполняется общим на группу элементов.

Из рис. 8.13(а) видно, что транзистор $VT1$ образован планарной структурой, а многоколлекторный транзистор $VT2$ - вертикальной структурой. Причем, так как площадь каждого коллектора транзистора $VT2$ меньше площади его эмиттера, этот транзистор, по сути, работает в инверсном режиме, что способствует уменьшению его напряжения насыщения. Все сказанное позволило разместить весь элемент I^2L на площади, занимаемой в схеме ТТЛ одним многоэмиттерным транзистором.

Важной особенностью элемента I^2L является возможность, варьируя ток инжектора в широких пределах, изменять его быстродействие. Реально ток инжектора может изменяться от 1 нА до 1 мА, т. е. на 6 порядков. А поскольку при заданной схемотехнике энергия переключения элемента - величина постоянная, в таких же пределах может изменяться и быстродействие элемента. Важно, что для этого не требуется никаких схемотехнических изменений в элементе.

Принцип действия схемы I^2L заключается в следующем. Допустим, внешний сигнал на входе элемента (база транзистора $VT2$) отсутствует, что соответствует сигналу лог. 1. В этом случае ток инжектора, втекая в базу транзистора $VT2$, насыщает его. На его коллекторах, а следовательно, и на выходных выводах элемента присутствует напряжение низкого уровня, равное напряжению насыщения транзистора $VT2$. Реально это 0,1 В ... 0,2 В.

Если база транзистора $VT2$ непосредственно или через насыщенный транзистор подключена к общей шине, то выполняется условие $U_{вх} < U_{БЭ0}$ и

транзистор $VT2$ заперт, так как ток инжектора замыкается на общую шину, минуя его эмиттерный переход. В этом случае напряжение на его коллекторах определяется внешними цепями. При последовательном включении нескольких инверторов это напряжение равно напряжению эмиттерного перехода последующего транзистора. Таким образом, для БЛЭ I^2L справедливы следующие соотношения:

$$U^0 = 0,1 \dots 0,1 \text{ В}; U^1 = 0,6 \dots 0,7 \text{ В}.$$

Из приведенных соотношений следует, что логический перепад для БЛЭ I^2L составляет 0,4 ... 0,6 В.

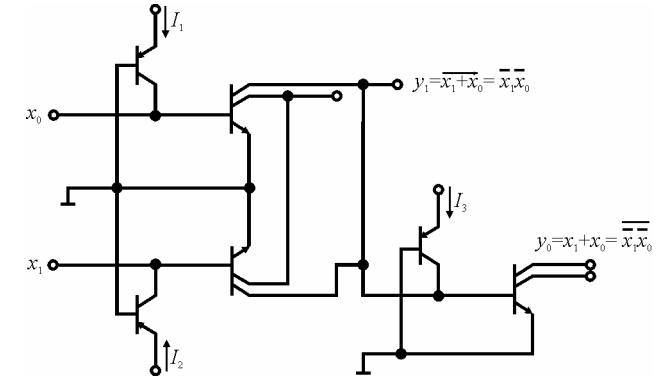


Рис. 8.14. Реализация логических операций 2И-НЕ и 2ИЛИ-НЕ на БЛЭ I^2L

С использованием приведенной схемы могут быть реализованы основные логические операции И-НЕ и ИЛИ-НЕ. На рис. 8.14 показана логическая схема, построенная на трех инверторах I^2L .

Особенностью элементов I^2L является возможность параллельного включения нескольких их выходов. Из приведенной схемы следует, что при параллельном включении нескольких выходов в общей точке относительно входных переменных реализуется логическая операция ИЛИ-НЕ. Относительно же выходных сигналов элементов реализуется логическая операция И. Таким образом, если не требуется гальваническое разделение между входными и выходными сигналами, то логическая операция И выполняется без каких-либо дополнительных схемотехнических затрат простым объединением соответствующих выходов БЛЭ. После инвертирования результата выполненной операции ИЛИ-НЕ дополнительным элементом относительно исходных входных переменных реализуется логическая операция ИЛИ, а относительно выходных сигналов первых элементов - операция И-НЕ.

Таким образом, БЛЭ I^2L позволяет максимально унифицировать структуру ИС, снизив площадь ее кристалла, и либо уменьшить ее потребление, либо повысить быстродействие.

Типовое время задержки распространения БЛЭ I^2L при токе инжектора 0,1 мкА составляет 10 нс. При этом энергия переключения для этого элемента на несколько порядков меньше, чем для элемента ТТЛ.

Ввиду небольшой помехоустойчивости, обусловленной малым логическим перепадом, БЛЭ I^2L используются исключительно в составе БИС и СБИС и как отдельные ИС малой степени интеграции не выпускаются. При этом входные и выходные цепи ИС, выполненных по технологии I^2L , делаются совместимыми по логическим уровням с сигналами ТТЛ.

Контрольные вопросы и упражнения к разделу 8

1. Что общего и каковы отличия потенциального и импульсного способов представления лог. 0 и лог. 1?
2. Назовите условия совместимости уровней входных и выходных сигналов логических элементов.
3. Что характеризуют коэффициенты объединения по входу и разветвления по выходу и каковы их типовые значения?
4. Изобразите амплитудную передаточную характеристику инвертирующего усилителя. В чем заключается формирующее свойство ЛЭ?
5. Чем определяется зона импульсной помехоустойчивости ЛЭ?
6. Назовите основные схемотехнические решения, применяемые при создании базовых ЛЭ разных типов; сравните их по техническим параметрам.
7. Приведите схему базового элемента ЗИ-НЕ ТТЛ и опишите ее функционирование.
8. Перечислите динамические параметры ЛЭ.
9. Сравните основные параметры ИС ТТЛ различных серий. Объясните причины существующих отличий.
10. Приведите схему токового ключа как основы БЛЭ ИС ЭСЛ.
11. Опишите функционирование схемы БЛЭ ЭСЛ.
12. В чем заключаются схемотехнические способы повышения быстродействия БЛЭ ЭСЛ?
13. Какие особенности характеризуют схемы на базе МДП-транзисторов?
14. Приведите схему ключа с нагрузочным МДП-транзистором и опишите ее функционирование.
15. Приведите схемы БЛЭ на МДП-транзисторах, реализующих операции ЗИ-НЕ и ЗИЛИ-НЕ, и опишите их функционирование.
16. Приведите схему и статическую характеристику ключа на КМОП-транзисторах.
17. Приведите схемы БЛЭ КМОП, выполняющие операции ЗИ-НЕ и 3 ИЛИ-НЕ.
18. В чем заключаются особенности ЛЭ И²Л. Приведите принципиальную схему базового логического элемента И²Л.

СПИСОК ЛИТЕРАТУРЫ

1. Аналоговая и цифровая электроника. (Полный курс): Учебник для вузов /Ю.Ф.Опадчий, О.П. Глудкин, А.И. Гуров; Под ред. О. П. Глудкина. – М.: Горячая линия – Телеком, 2002. – 768 стр.
2. Фрике К. Вводный курс цифровой электроники: Пер. с немецкого. (изд. 2-ое) – М.: Техносфера, 2004 г. – 432 стр.
3. Токхейм Р. Основы цифровой электроники: Пер. с англ. – М.: Мир, 1988. – 392стр.
4. Шило В. Л. Популярные цифровые микросхемы: Справочник. – М: Радио и связь, 1987. – 352 стр.