

Министерство образования и науки Российской Федерации
Ярославский государственный университет им. П. Г. Демидова

К. С. Артемов, Н. Л. Солдатова

ОСНОВЫ ЦИФРОВОЙ ЭЛЕКТРОНИКИ

Учебное пособие

Рекомендовано

*Научно-методическим советом университета для студентов,
обучающихся по направлениям Радиотехника, Радиофизика*

Ярославль
ЯрГУ
2013

УДК 621.38(075.8)
ББК 385я73
А86

*Рекомендовано
Редакционно-издательским советом университета
в качестве учебного издания. План 2013 года*

Рецензенты:

Проказников А. В., доктор физико-математических наук,
ведущий научный сотрудник Ярославского филиала
Федерального государственного бюджетного учреждения науки
Физико-технологического института Российской академии наук;
научно-технический совет НПФ «Яр»

Артемов, Константин Серафимович.

А86 Основы цифровой электроники: учебное посо-
бие / К. С. Артемов, Н. Л. Солдатова ; Яросл. гос. ун-т
им. П. Г. Демидова. – Ярославль : ЯрГУ, 2013. – 100 с.
ISBN 978-5-8397-0979-9

Излагаются основы теории цифровой электроники
от простейших логических операций до синтеза триггер-
ных устройств.

Предназначено для студентов, обучающихся по на-
правлениям 210400.62 Радиотехника, 011800.62 Радио-
физика (дисциплины «Цифровая электроника», «Циф-
ровые устройства», цикл Б2), очной формы обучения,
а также могут быть полезны для студентов специаль-
ности «Радиофизика и электроника».

УДК 621.38(075.8)
ББК 385я73

ISBN 978-5-8397-0979-9

© ЯрГУ, 2013

Предисловие

Учебное пособие предполагает знание студентами физических основ электроники, принципа действия и параметров диодов и транзисторов.

Авторы не ставили своей целью охватить все разделы цифровой схемотехники. В книге достаточно подробно описаны лишь основные цифровые устройства. В заключительной части представлено введение в теорию и практику основ теории логических устройств. К каждой главе даются вопросы и задания для самоконтроля. Отдельной частью выделены лабораторные задания, которые позволят закрепить теоретические знания и дадут навыки построения и расчета схем основных цифровых устройств и схем. В большинстве случаев приводятся примеры построения устройств, что существенно облегчая освоение материала студентами, особенно обучающихся заочно.

В соответствии с программой дисциплин по основам цифровой и вычислительной техники в пособии приведены методические указания по выполнению лабораторных работ фронтальным методом на соответствующих макетах. Для студентов-заочников полезным будет компьютерное моделирование в среде Electronics WORKBENCH.

ГЛАВА I. ОСНОВЫ ТЕОРИИ ЛОГИЧЕСКИХ УСТРОЙСТВ

1.1. Простейшие логические операции

В логических устройствах сигнал на входе и выходе каскада является двоичным, бинарным. Он может принимать только два значения: логического нуля «0» или логической единицы «1». Значения «0» и «1» являются символическими (условными) и не соответствуют числовым значениям напряжения, выражаемых в вольтах. В реальных цифровых устройствах приходится придавать логическим переменным определённые значения. Например, наличие напряжения на входе или выходе устройства соответствует логической единице «1», а отсутствие – логическому нулю «0».

Проектирование логических устройств проводят с использованием алгебры логики, разработанной в середине XIX века ирландским математиком Джорджем Булем. В булевой алгебре используют двоичную переменную X , удовлетворяющую условию: $X=1$, если $X \neq 0$, и $X=0$, если $X \neq 1$. С такими переменными можно производить следующие основные логические операции.

Операция дизъюнкции

Эту операцию называют также операцией ИЛИ (операция логического сложения). Для двух переменных (X_1 и X_2) эта операция даёт такие результаты:

$$1 + 0 = 1; \quad 0 + 1 = 1; \quad 1 + 1 = 1; \quad 0 + 0 = 0.$$

Операция, например, для первого выражения произносится так: «Единица ИЛИ ноль ЕСТЬ (ЭТО) единица». Под знаком равенства понимается тождественность левой и правой частей. Переменная Y принимает единичное значение, если хотя бы одна из переменных равна единице. Результат операции дизъюнкции, как и других логических операций, удобно отражать с помощью так называемых таблиц истинности, в которых записываются все возможные значения переменных X_1 и X_2 , т. е. все возможные их сочетания. В той же таблице приводятся и значения функции Y для данной комбинации логических переменных. Значения функ-

ции Y , равные единице, называют истинными, а значения, равные нулю, – ложными. Таблица истинности для операции дизъюнкции соответствует таблице 1.1.

Операция ИЛИ

Таблица 1.1

X_1	X_2	Y
0	0	0
0	1	1
1	0	1
1	1	1

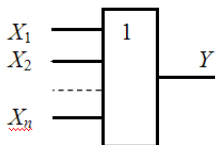


Рис. 1.1. Операция логического сложения

Аналитически операцию дизъюнкции над переменными X_1 и X_2 записывают в виде

$$Y = X_1 + X_2 \quad \text{или} \quad Y = X_1 \vee X_2.$$

Условное обозначение логического элемента, выполняющего эту операцию, показано на рис. 1.1. В цифровой электронике принято считать, что слева входы, а справа выходы устройства. Простейшее логическое устройство, выполняющее операцию дизъюнкции над логическими переменными X_1 и X_2 , выражаемыми в форме электрических напряжений, представлено на рис. 1.2.

Под единичным уровнем понимают высокий положительный потенциал. Если единичный уровень присутствует хотя бы на одном входе, то через открытый диод VD1 (VD2) это напряжение передается на выход, создавая единичный уровень напряжения.

Операция конъюнкции

Эту операцию называют также операцией И (операцией логического умножения). Аналитически операцию для двух переменных записывают в виде

$$Y = X_1 \cdot X_2 \quad \text{или} \quad Y = X_1 X_2$$

(без точки, изображающей знак логического умножения) или в виде

$$Y = X_1 \wedge X_2.$$

Рассмотрим возможные комбинации значений переменных. При операции конъюнкции

$$0 \cdot 0 = 0; 0 \cdot 1 = 0; 1 \cdot 0 = 0; 1 \cdot 1 = 1.$$

Значение функции Y истинно только в том случае, когда переменная X_1 так же, как и переменная X_2 , принимает единичное значение. Таблица 1.2 соответствует таблице истинности функции

$$Y = f(X_1, X_2) = X_1 \cdot X_2.$$

Условное обозначение логического элемента, выполняющего операцию конъюнкции, показано на рис. 1.3. Принципиально схема простейшего логического каскада И представлена на рис. 1.4. Если хотя бы на одном входе схемы имеется низкий уровень положительного напряжения, принимаемый за условный нуль, то диод, связанный через катод с этим входом, открыт и напряжение на его аноде, а следовательно, и на выходе устройств равно нулю. Если же на всех входах системы присутствует высокий (единичный) уровень напряжения, то диоды запираются и образуют для источника питания с резистором R делитель напряжения источника питания с коэффициентом передачи близким к единице. На выходе получается высокий уровень, соответствующий логической единице.

Операция И

Таблица 1.2

X_1	X_2	Y
0	0	0
0	1	0
1	0	0
1	1	1

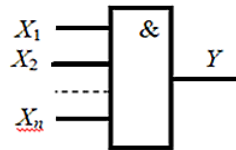


Рис. 1.2. Операция логической конъюнкции

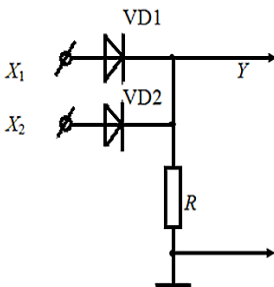


Рис. 1.3. Простейшее логическое устройство, выполняющее операцию дизъюнкции

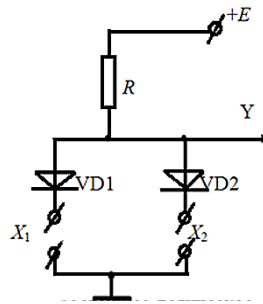


Рис. 1.4. Простейшее логическое устройство, выполняющее операцию конъюнкции

Эту операцию называют операцией НЕ (операцией логического отрицания). Операцию инверсии записывают в виде

$$Y = \bar{X}.$$

Выполняется эта операция над переменной X . Таблица истинности для этой операции соответствует таблице 1.3. Таким образом,

$$\bar{1} = 0; \quad \bar{0} = 1.$$

Условное обозначение устройства, выполняющего данную операцию, представлено на рис. 1.5. Здесь и в более сложных логических устройствах кружок в разрыве контура на выходе Y условно обозначает инверсность значения Y относительно X . Схемным примером инвертора могут служить ключевые каскады по схеме с общим эмиттером или общим истоком.

Операция НЕ

Таблица 1.3

X	Y
0	1
1	0

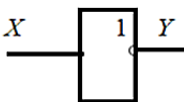


Рис. 1.5. Операция логической конъюнкции

Операции И, ИЛИ, НЕ – примеры простейших булевых функций.

Систему логических элементов И, ИЛИ, НЕ называют функционально полной системой логических элементов, так как с их помощью можно реализовать любую более сложную функцию.

1.2. Двойственность алгебры Буля

Увидеть двойственность можно просто, если проследить за алгоритмом вычисления простейших функций И и ИЛИ.

Рассмотрим функцию И. Здесь, если обе переменные равны 1, то функция $X_1 X_2 = 1$ истинна. Если же X_1 и X_2 равны нулю, то $X_1 X_2$ – ложна, но $X_1 + X_2$ тоже ложна. Таким образом, для единиц мы имеем функцию И, а для нолей – ИЛИ (рис. 1.6).

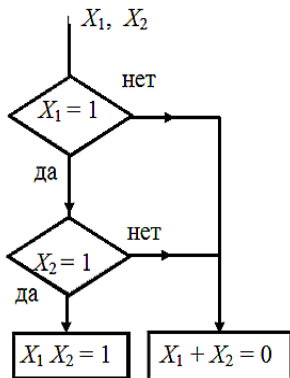


Рис. 1.6. Функция И

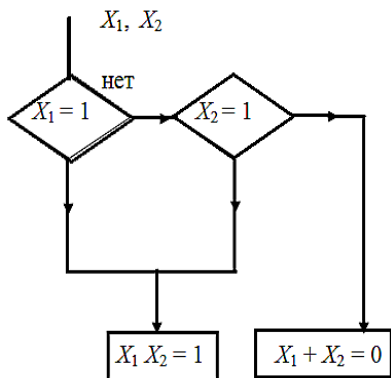


Рис. 1.7. Функция ИЛИ

Функция ИЛИ. Здесь, если $X_1=1$ или $X_2=1$ или X_1 и X_2 равны 1, то $X_1 + X_2=1$, т. е. функция истинна, но если $X_1=0$ и $X_2=0$, то справедливо и произведение $X_1X_2=0$. Таким образом, для единиц мы имеем функцию ИЛИ, а для нулей – И (рис. 1.7).

Можно продемонстрировать двойственность и словами на таком примере:

1) в комнате ТЕПЛО, если батарея ВКЛЮЧЕНА И окно ЗАКРЫТО.

2) в комнате НЕ ТЕПЛО, если батарея НЕ ВКЛЮЧЕНА ИЛИ окно НЕ ЗАКРЫТО.

Двойственность: отрицаются аргументы и сама функция, а смысл остается одним и тем же. Возвращаясь к функциям И и ИЛИ, можно сказать, что если $f = X_1 + X_2$, то двойственная к ней $fd = X_1X_2$ получается заменой «+» на «*», и наоборот: $f = X_1X_2$, $fd = X_1 + X_2$.

1.3. Аксиомы и законы алгебры

Аксиомы дизъюнкции.

1. $X+0=X$. Действительно, если $X=0$, то $0+0=0$, а если $X=1$, то $1+0=1$.

2. $X+1=1$ – при любом X все определяет 1.

3. $X+X=X$ – $0+0=0$, $1+1=1$.

4. $X + \bar{X} = 1$.

Аксиомы конъюнкции.

1. $X0=0 - 00=0, 10=0.$
2. $X1=X - 01=0, 11=1.$
3. $XX=X - 00=0, 11=1.$
4. $X\bar{X} = 0 - 01=0, 10=0.$

Аксиомы под номером 1 называют теоремой объединения, под номером 2 – пересечения, 3 – закон тавтологии, 4 – закон дополнительности.

Аксиомы инверсии.

1. $X = \bar{\bar{X}}$. Словами: « X – это НЕ НЕ X ».
2. $\overline{(X)} = \bar{X}$. Словами: «НЕ от X это НЕ X ».

Законы алгебры-логики

1. Закон коммутативности, или переместительный закон

$$X_1 + X_2 = X_2 + X_1, X_1 \cdot X_2 = X_2 X_1$$

Справедливость законов можно подтвердить путем подстановки непосредственно значений аргументов, т. е. 1 и 0, перебрав все комбинации.

С точки зрения электроники этот закон позволяет подавать сигналы на любые входы элемента.

2. Закон ассоциативности или сочетательный закон

$$X_1 + X_2 + X_3 = X_1 + (X_2 + X_3) = (X_1 + X_2) + X_3$$

3. Закон дистрибутивности или распределительный закон

$$X_1 \cdot (X_2 + X_3) = X_1 \cdot X_2 + X_1 \cdot X_3$$

$$X_1 + X_2 \cdot X_3 = (X_1 + X_2) \cdot (X_1 + X_3).$$

Последняя запись не имеет аналога в математике. Но его справедливость можно доказать. Исходя из справедливости $X_1 \cdot (X_2 + X_3) = X_1 \cdot X_2 + X_1 \cdot X_3$ для любых значений переменных X_1, X_2 и X_3 используем принцип двойственности для изменения формы записи левой и правой частей. В результате получим: $\overline{X_1 + X_2 \cdot X_3} = \overline{(X_1 + X_2) \cdot (X_1 + X_3)}$. Введем новые обозначения $\overline{X_1} = A, \overline{X_2} = B, \overline{X_3} = C$ и получим $A + B\bar{C} = (A + B)\bar{C}$ ($A + B$), что полностью совпадает со второй формой записи закона дистрибутивности.

4. Закон поглощения или избыточности

$X_1 + X_1 \cdot X_2 = X_1$, а в двойственной форме $X_1(X_1 + X_2) = X_1$.

Докажем. $X_1 + X_1 \cdot X_2 = X_1(1 + X_2)$ – по аксиоме пересечения $1 + X_2 = 1$ и $X_1 \cdot 1 = X_1$, то $X_1 + X_1 \cdot X_2 = X_1$. Соответственно для второго

$$X_1(X_1 + X_2) = X_1 \cdot X_1 + X_1 \cdot X_2 = X_1 + X_1 \cdot X_2 = X_1 \cdot (1 + X_2) = 1 + X_2 = 1 = X_1 \cdot 1 = X_1.$$

5. Закон склеивания

$$X_1 X_2 + \overline{X_1} \cdot X_2 = X_2$$

$$(X_1 + X_2) \cdot (\overline{X_1} + X_2) = X_2$$

Первому члену можно поставить в соответствие лист бумаги, на одной стороне которого написано X_1 , а на другой стороне – X_2 . Второму члену соответствует лист с надписями $\overline{X_1}$ и X_2 . При склеивании заклеивается меняющаяся величина X_1 . Докажем эти соотношения

$$X_1 \cdot X_2 + \overline{X_1} + X_2 = \overline{X_1} \quad X_2 \cdot (X_1 + \overline{X_1}) \text{ т.к. } X_1 + \overline{X_1} = 1, \text{ то}$$

Аналогично

$$(X_1 + X_2) \cdot (X_2 + \overline{X_1}) = X_1 \cdot \overline{X_1} + X_1 \cdot X_2 + X_2 \cdot \overline{X_1} + X_2 \cdot X_2 = X_1 + \overline{X_1} = 0, X_2 \cdot X_2 = X_2 \mid = X_2 + X_2 \cdot (X_1 + \overline{X_1}) = X_1 + \overline{X_1} = 1, X_2 \cdot 1 = X_2 \mid = X_2 + X_2 = X_2$$

6. Правило де Моргана

$\overline{X_1 + X_2} = \overline{X_1} \cdot \overline{X_2}$ – отрицание дизъюнкции есть конъюнкция отрицаний.

$\overline{X_1 \cdot X_2} = \overline{X_1} + \overline{X_2}$ – отрицание конъюнкции есть дизъюнкция отрицаний.

Справедливость вытекает из принципа двойственности, т. е. получая из ИЛИ И путем замены переменных на инверсные и знака (+) на (*).

Так, для нескольких переменных правило будет

$$\overline{X_1 + X_2 + X_3 + \dots + X_n} = \overline{X_1} \cdot \overline{X_2} \cdot \overline{X_3} \cdot \dots \cdot \overline{X_n}$$

$$\overline{X_1 \cdot X_2 \cdot X_3 \cdot \dots \cdot X_n} = \overline{X_1} + \overline{X_2} + \overline{X_3} \dots + \overline{X_n}$$

$$X_1 + X_2 + X_3 + \dots + X_n = \overline{\overline{X_1} \cdot \overline{X_2} \cdot \overline{X_3} \cdot \dots \cdot \overline{X_n}}$$

$$X_1 \cdot X_2 \cdot X_3 \cdot \dots \cdot X_n = \overline{\overline{X_1} + \overline{X_2} + \overline{X_3} + \dots + \overline{X_n}} \text{ или}$$

Справедливость законов можно подтвердить и путем постановки непосредственных значений аргументов, т. е. 0 или 1, перебрав все комбинации.

Положительная и отрицательная логика

В электронике в качестве общего полюса источника питания может быть как плюс, так и минус. У ключа на биполярном транзисторе $n - p - n$ в режиме отсечки на выходе большой положительный уровень, а в режиме насыщения малый. Ключ на $p - n - p$ -транзисторе в разомкнутом состоянии имеет большой отрицательный уровень, а в замкнутом состоянии низкий. Что считать единицей, а что нулём? В логических устройствах ввели понятие положительной и отрицательной логики.

В положительной логике единица – высокий уровень, ноль – низкий уровень или более отрицательный. Транзистор $n - p - n$ в режиме отсечки – на выходе единица, в режиме насыщения – ноль. Транзистор $p - n - p$ в режиме отсечки – на выходе большой минус – ноль, низкий уровень – единица. В отрицательной логике всё наоборот.

Если логический элемент даёт на выходе высокий уровень при совпадении на всех входах высоких уровней, а при низком уровне хотя бы на одном входе низкий уровень на выходе, то это элемент И в положительной логике, а в отрицательной логике это элемент ИЛИ. Изменение трактовки происходит по принципу двойственности или по правилу де Моргана.

1.4. Логические функции

Число возможных логических функций определяется формулой $N = 2^{2^n}$, где n – число аргументов. Так, если число аргументов $n = 2$, то функций будет 16. При $n = 3$ $N = 256$, при $n = 4$ $N = 65536$. На практике используется лишь малая часть возможных функций.

Существует несколько способов задания логических функций.

1. Словами. Например, для операции $Y = X_1 + X_2 + \dots + X_n$ значение функции истинно, если хотя бы один из ее аргументов принимает единичное значение.

2. Таблицей истинности. Например,

X_1	X_2	X_3	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

Функция Y соответствует так называемому мажоритарному элементу «2 из трех». Эта функция хороша тем, что по названию можно понять её логику: «Функция истинна, если, по крайней мере, два из трех аргументов равны 1». Подобные формулировки используют в кодовых замках. В таблице конкретный ряд (строка) значений аргументов, например, 0 1 0, называется набором. От табличной формы записи можно перейти к аналитической. Из таблицы видно, что значения функции истинны только для некоторых наборов значений переменных.

а) $X_1 = 0, X_2 = 0, X_3 = 1$, т. е. для сочетания $\bar{X}_1 \bar{X}_2 X_3$;

б) $X_1 = 1, X_2 = 0, X_3 = 1$, т. е. для сочетания $X_1 \bar{X}_2 X_3$;

в) $X_1 = 1, X_2 = 1, X_3 = 0$, т. е. для сочетания $X_1 X_2 \bar{X}_3$;

г) $X_1 = 1, X_2 = 1, X_3 = 1$, т. е. для сочетания $X_1 X_2 X_3$.

Подставьте в какое-либо сочетание соответствующие конкретные значения аргументов, и Вы получите конъюнкцию, равную единице.

Каждое из произведений переменных, для которых значение функции истинно, называется минтермом, или конституентом единицы, а функцию можно представить в виде суммы всех минтермов (т. к. каждый минтерм равен 1, то сумма равна 1 и функция справедлива для всех таких наборов):

$$Y = \bar{X}_1 \bar{X}_2 X_3 + X_1 \bar{X}_2 X_3 + X_1 X_2 \bar{X}_3 + X_1 X_2 X_3.$$

Если функция представлена дизъюнкцией отдельных членов, каждый из которых конъюнкция аргументов, в том числе инвер-

сий, то это – дизъюнктивная форма. Если в дизъюнктивной форме нет отрицаний над смыслами (термами), а есть только отрицания, применимые к аргументам, то это дизъюнктивная нормальная форма. Если в каждом слагаемом дизъюнктивной нормальной формы есть все переменные или их отрицания, то это совершенная дизъюнктивная нормальная форма (СДНФ).

Также можно выделить и нулевые значения функции, имеющиеся в таблице: если истинное значение – это Y , то неистинное (нулевое) – это \bar{Y} :

а) $X_1 = 0, X_2 = 0, X_3 = 0$, т. е. для сочетания $\bar{X}_1 \bar{X}_2 \bar{X}_3$;

б) $X_1 = 0, X_2 = 0, X_3 = 1$, т. е. для сочетания $\bar{X}_1 \bar{X}_2 X_3$;

в) $X_1 = 0, X_2 = 1, X_3 = 0$, т. е. для сочетания $\bar{X}_1 X_2 \bar{X}_3$;

г) $X_1 = 1, X_2 = 0, X_3 = 0$, т. е. для сочетания $X_1 \bar{X}_2 \bar{X}_3$.

Так как эти сочетания дают 0, то, сложив их, получим тоже 0:

$$\bar{Y} = \bar{X}_1 \bar{X}_2 \bar{X}_3 + \bar{X}_1 \bar{X}_2 X_3 + \bar{X}_1 X_2 \bar{X}_3 + X_1 \bar{X}_2 \bar{X}_3.$$

Используя принцип двойственности или правило де Моргана, получим

$$Y = (X_1 + X_2 + X_3)(X_1 + X_2 + \bar{X}_3)(X_1 + \bar{X}_2 + X_3)(\bar{X}_1 + X_2 + X_3).$$

Функция в этом случае задана в виде произведения (конъюнкции) сумм переменных или их отрицаний. Так как в суммы входят все переменные или их отрицания, то такая форма записи называется совершенной конъюнктивной нормальной формой (СКНФ).

Сами же суммы, для которых значение функции неистинно, называют макстермами, или конституентами нуля.

1.5. Построение логической схемы

Речь идет о том, как по аналитической форме записи осуществить реализацию функции на практике.

Рассмотрим на примере мажоритарного элемента типа «два из трех». По СДНФ имеем:

$$Y = \underbrace{\bar{X}_1 X_2 X_3}_{(1)} + \underbrace{X_1 \bar{X}_2 X_3}_{(2)} + \underbrace{X_1 X_2 \bar{X}_3}_{(3)} + \underbrace{X_1 X_2 X_3}_{(4)}.$$

Для реализации Y потребуются:

а) Инверторы (перевод X_1, X_2, X_3 в $\bar{X}_1, \bar{X}_2, \bar{X}_3$), т. к. значений таких три, то необходимо три инвертора;

б) Схемы И, реализованные, например, на диодах. Каждая схема должна иметь по 3 входа (по числу сомножителей минтерма), а общее их число равно числу минтермов в формуле;

в) Схемы ИЛИ (например, на диодах) – число входов равно числу минтермов, т. е. 4, а общее число в нашем случае – один элемент.

Строим схему. Для упрощения соединения элементов обозначим термы и логические элементы цифрами и соединим их в соответствии с записью (рис. 1.8).

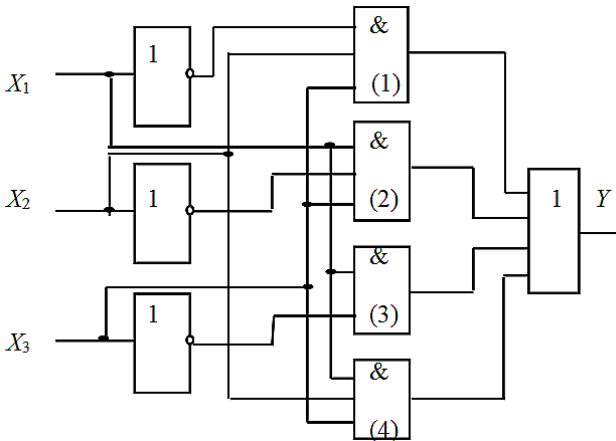


Рис. 1.8. Схемная реализация СДНФ «два из трех»

Аналогично можно реализовать и СНКФ записи. В этом случае понадобится также 3 инвертора, 4 схемы ИЛИ с тремя входами и одна четырехвходовая схема И.

1.6. Универсальные логические элементы

Разработчики электронной аппаратуры заметили наиболее часто встречающиеся логические комбинации и разработали так называемые универсальные логические элементы.

Элемент ИЛИ-НЕ

Элемент ИЛИ-НЕ реализует логическую функцию

$$Y = X_1 + X_2 .$$

Эту функцию называют функцией ИЛИ-НЕ (отрицание дизъюнкции). Иногда ее обозначают

$$Y = X_1 \downarrow X_2$$

и называют стрелкой Пирса, функцией Вебба, или NOR. Условное обозначение элемента, выполняющего данную операцию, показано на рис. 1.9.

Элемент ИЛИ-НЕ выполняет все основные логические операции:

а) операцию инверсии (рис. 1.9.1). При объединении входов логического элемента получим

$$\overline{X + X} = \bar{X},$$

т. е. сигнал \bar{X} на выходе;

б) операцию дизъюнкции (рис. 1.9.2). Последовательное включение двух элементов, второй из которых работает как инвертор, позволяет реализовать преобразование

$$Y = Y_1 = \overline{\overline{X_1 + X_2}} = X_1 + X_2;$$

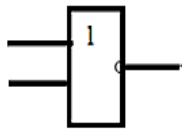


Рис. 1.9.1. Функция ИЛИ-НЕ

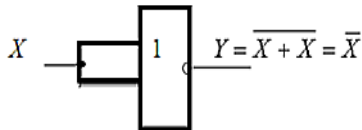


Рис. 1.9.2. Функция НЕ на элементе ИЛИ-НЕ

в) операцию конъюнкции (рис. 1.9.3). Включение логических элементов, работающих в режиме инверторов, во входные цепи третьего элемента позволяет реализовать логическую операцию

$$Y = \overline{Y_1 + Y_2} = \overline{\bar{X}_1 + \bar{X}_2} = X_1 \cdot X_2 .$$

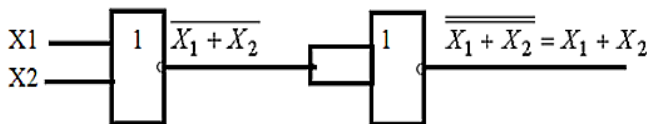


Рис. 1.9.3. Операция дизъюнкции ИЛИ-НЕ

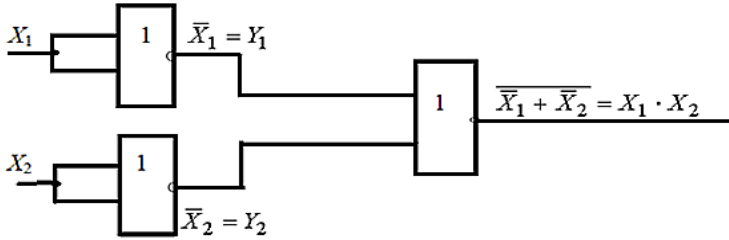


Рис. 1.9.4. Операция И на элементах ИЛИ-НЕ

Элемент И-НЕ

Элемент И-НЕ реализует логическую функцию

$$Y = \overline{X_1 X_2}$$

(отрицание конъюнкции). Эту операцию обозначают

$$Y = X_1 / X_2$$

и называют функцией Шеффера, или штрихом Шеффера, или NAND. Условное обозначение элемента И-НЕ дано на рис. 1.10.1.

Элемент И-НЕ выполняет все основные логические операции:

а) операцию инверсии (рис. 1.10.2). При объединении ходов

$$Y = \overline{X \cdot X} = \overline{X} / Y = \overline{X};$$

б) операцию конъюнкции (рис. 1.10.3). Последовательное включение двух элементов, работающих как инверторы, между точкой подключения входного сигнала и соответствующим входом третьего элемента И-НЕ позволяет реализовать функцию

$$Y = \overline{\overline{Y_1}} = \overline{\overline{X_1 \cdot X_2}} = X_1 \cdot X_2,$$

т. е. операцию И;

в) операцию дизъюнкции (рис. 1.10.4). Включение двух элементов, работающих как инверторы, между точкой подключения входного сигнала и соответствующим входом третьего элемента И-НЕ позволяет реализовать функцию

$$Y = \overline{\overline{Y_1} \cdot \overline{Y_2}} = \overline{\overline{X_1} \cdot \overline{X_2}} = X_1 + X_2,$$

т. е. функцию ИЛИ.

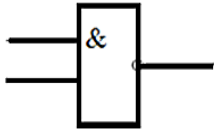


Рис. 1.10.1. Элемент И-НЕ

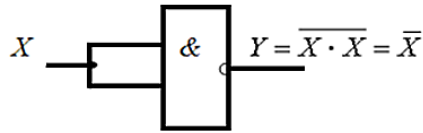


Рис. 1.10.2. Инверсия на элементе И-НЕ

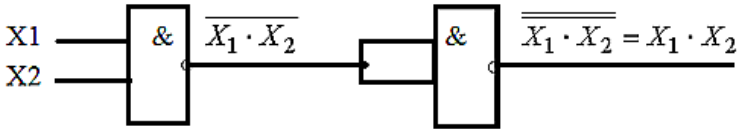


Рис. 1.10.3. Операция И на элементах И-НЕ

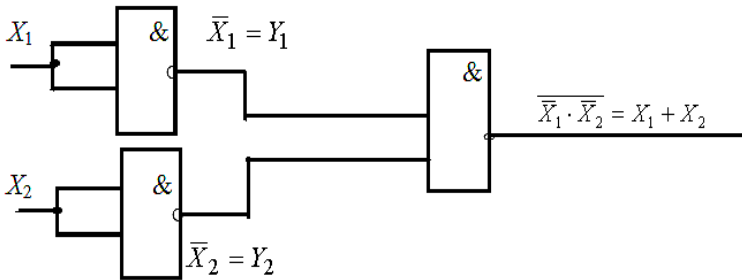


Рис. 1.10.4. Операция ИЛИ на элементах ИЛИ-НЕ

В общем случае как элементы ИЛИ-НЕ, так и элементы И-НЕ могут иметь не два, а n входов.

Для перехода в универсальный базис используют правило де Моргана. Покажем на примере «два из трех».

$$Y = \bar{X}_1 X_2 X_3 + X_1 \bar{X}_2 X_3 + X_1 X_2 \bar{X}_3 + X_1 X_2 X_3.$$

1. В базисе И-НЕ.

Используем аксиому инверсии $\bar{\bar{X}} = X$ и ставим над левой и правой частями двойное отрицание:

$$\bar{\bar{Y}} = \bar{\bar{X}_1 X_2 X_3 + X_1 \bar{X}_2 X_3 + X_1 X_2 \bar{X}_3 + X_1 X_2 X_3}.$$

Раскрываем нижнее отрицание правой части по де Моргану:

$$\bar{Y} = \overline{\bar{X}_1 X_2 X_3} \cdot \overline{X_1 \bar{X}_2 X_3} \cdot \overline{X_1 X_2 \bar{X}_3} \cdot \overline{X_1 X_2 X_3}.$$

2. В базисе ИЛИ-НЕ убираем конъюнкцию:

$$Y = \overline{\bar{X}_1 X_2 X_3} + \overline{X_1 \bar{X}_2 X_3} + \overline{X_1 X_2 \bar{X}_3} + \overline{X_1 X_2 X_3}.$$

Раскрываем нижние отрицания по де Моргану:

$$Y = \overline{X_1 + \bar{X}_2 + \bar{X}_3} + \overline{\bar{X}_1 + X_2 + \bar{X}_3} + \overline{\bar{X}_1 + \bar{X}_2 + X_3} + \overline{\bar{X}_1 + \bar{X}_2 + \bar{X}_3}.$$

Остается привести дизъюнкцию слагаемых к форме ИЛИ-НЕ

$$\bar{Y} = \overline{\overline{\bar{X}_1 + \bar{X}_2 + \bar{X}_3} + \overline{\bar{X}_1 + X_2 + \bar{X}_3} + \overline{\bar{X}_1 + \bar{X}_2 + X_3} + \overline{\bar{X}_1 + \bar{X}_2 + \bar{X}_3}}.$$

Раскрывать общую инверсию здесь не следует.

Используя условные обозначения логических элементов, попытайтесь реализовать функцию Y в базисе И-НЕ или ИЛИ-НЕ. Не забывайте вводить для облегчения работы нумерацию термов и элементов.

1.7. Функции двух переменных

Кроме основных логических операций (НЕ, ИЛИ, И), существует ряд других. Например, для двух переменных возможно 16 операций. Если каждой операции привести в соответствие функцию, то таким образом для двух переменных существует набор из 16 функций.

x_1 x_2	0	0	1	1	Условное обозначение и алгебраическое вычисление	Название функции
f_0	0	0	0	0	$f_0 = 0$	Постоянный 0
f_1	0	0	0	1	$f_1 = x_1 x_2$	Конъюнкция
f_2	0	0	1	0	$f_2 = x_1 \rightarrow x_2 = x_1 \bar{x}_2$	Запрет или словами: «И с НЕ x_2 »
f_3	0	0	1	1	$f_3 = x_1$	Тождественность x_1
f_4	0	1	0	0	$f_4 = x_2 \rightarrow x_1 = \bar{x}_1 x_2$	Запрет или словами: «И с НЕ x_1 »
f_5	0	1	0	1	$f_5 = x_2$	Тождественность x_2
f_6	0	1	1	0	$f_6 = x_1 \oplus x_2 = x_1 \bar{x}_2 + \bar{x}_1 x_2$	Исключающее ИЛИ, Исключительное ИЛИ, Неравнозначность

f_7	0	1	1	1	$f_7 = x_1 + x_2$	Дизъюнкция
f_8	1	0	0	0	$f_8 = x_1 \downarrow x_2 = \overline{x_1 + x_2}$	Стрелка Пирса (ИЛИ-НЕ)
f_9	1	0	0	1	$f_9 = x_1 x_2 + \bar{x}_1 \bar{x}_2$	Равнозначность (эквивалентность), исключающее ИЛИ-НЕ
f_{10}	1	0	1	0	$f_{10} = \bar{x}_2$	Инверсия x_2 (НЕ x_2)
f_{11}	1	0	1	1	$f_{11} = x_2 \rightarrow x_1 = x_1 + \bar{x}_2$	Импликация от x_2 к x_1 , ИЛИ с (НЕ x_2)
f_{12}	1	1	0	0	$f_{12} = \bar{x}_1$	Инверсия x_1
f_{13}	1	1	0	1	$f_{13} = x_1 \rightarrow x_2 = \bar{x}_1 + x_2$	Импликация от x_1 к x_2 , ИЛИ с (НЕ x_1)
f_{14}	1	1	1	0	$f_{14} = x_1 / x_2 = \bar{x}_1 \bar{x}_2$	Штрих Шеффера (И-НЕ)
f_{15}	1	1	1	1	$f_{15} = 1$	Постоянная 1

При внимательном рассмотрении таблицы можно понять, почему на практике используется мало функций. Например, функция «постоянный ноль» является инверсией функции «постоянная единица». Другой пример. Если проинвертировать функцию «Запрет», то получим импликацию по одному аргументу, а двойственная форма «Запрета» даёт импликацию по второму аргументу. Остановимся ещё на одной функции – «Неравнозначность». Её называют ещё суммой по модулю 2 (М2). Если складывать два одноразрядных числа, то мы получим, без учёта переноса, таблицу истинности функции «Неравнозначность». Она используется в цифровых компараторах и сумматорах. Логика этого элемента на русском языке звучит так: «Жизнь или кошелёк», или из истории древнего Рима: «Со щитом или на щите». Логический смысл импликации: «Если..., то...». Логика запрета звучит чётко: «Нет». Функция ИЛИ-НЕ: «Ни – ни». Например, ни рыба, ни мясо. Смыслового аналога функции И-НЕ в русском языке нет. По этой причине русскому человеку мыслить в базисе И-НЕ очень трудно.

1.8. Карта Карно

Под картой Карно мы будем понимать таблицу всех возможных минтермов и макстермов функции. Пример построения карты для двух переменных:

X_2	\bar{X}_2	X_2
X_1		
\bar{X}_1		
X_1		

X_2	\bar{X}_2	X_2
X_1	$\bar{X}_1\bar{X}_2$	\bar{X}_1X_2
\bar{X}_1	$X_1\bar{X}_2$	X_1X_2
X_1	$\bar{X}_1\bar{X}_2$	\bar{X}_1X_2

X_1	\bar{X}_1	X_1
X_2	$\bar{X}_1\bar{X}_2$	$X_1\bar{X}_2$
\bar{X}_2	$X_1\bar{X}_2$	X_1X_2
X_2	$\bar{X}_1\bar{X}_2$	$X_1\bar{X}_2$

X_1	X_2	\bar{X}_2
0	00	10
1	01	11

По строке меняется переменная X_1 , а по столбцу – X_2 . В клетки заносятся произведения (термы) переменных в соответствии с пересечением строк и столбцов. Строки и столбцы можно менять местами. Вместо X можно ставить конкретные значения 0 или 1.

Карта Карно для трех и четырех переменных:

X_1X_2	$\bar{X}_1\bar{X}_2$	\bar{X}_1X_2	X_1X_2	$X_1\bar{X}_2$
X_3	$\bar{X}_1\bar{X}_2\bar{X}_3$	$\bar{X}_1X_2\bar{X}_3$	$X_1X_2\bar{X}_3$	$X_1\bar{X}_2\bar{X}_3$
\bar{X}_3	$\bar{X}_1\bar{X}_2X_3$	$\bar{X}_1X_2X_3$	$X_1X_2X_3$	$X_1\bar{X}_2X_3$
X_3	$\bar{X}_1\bar{X}_2X_3$	$\bar{X}_1X_2X_3$	$X_1X_2X_3$	$X_1\bar{X}_2X_3$

X_3X_4	$\bar{X}_3\bar{X}_4$	\bar{X}_3X_4	X_3X_4	$X_3\bar{X}_4$
X_1X_2	0000	0001	0011	0010
$\bar{X}_1\bar{X}_2$	0100	0101	0111	0110
\bar{X}_1X_2	1100	1101	1111	1110
X_1X_2	1000	1001	1011	1010
$X_1\bar{X}_2$	1000	1001	1011	1010

Мы видим, что карта Карно перебирает все возможные наборы переменных. Обязательным условием карты Карно является то, что при переходе от клетки к клетке меняется **только одна** переменная. Например, изменения по верхней строке

$$\bar{X}_3\bar{X}_4 \rightarrow \bar{X}_3X_4 \rightarrow X_3X_4 \rightarrow X_3\bar{X}_4.$$

С помощью карты Карно можно задать функцию. Для этого сравнивают последовательно наборы таблиц состояний с термами карты. Если при каком-либо наборе значение функции равно 1, то в соответствующую этому набору клетку ставят 1.

	X_2		
X_1	\bar{X}_2	X_2	
\bar{X}_1	0	0	
X_1	0	1	

	X_2		
X_1	\bar{X}_2	X_2	
\bar{X}_1	0	1	
X_1	1	1	

Если набор дает нулевое значение функции, то ставят 0 (например, карта Карно для функций двух переменных).

Для построения алгебраического выражения функции, заданной картой Карно, следует просуммировать все термы, имеющие значение 1. В этом случае мы получим СДНФ функции. Например, для карты Карно функции И это всего лишь один терм $X_1 X_2$, т. е. $f = X_1 X_2$.

Для функции ИЛИ: $f = \bar{X}_1 X_2 + X_1 \bar{X}_2 + X_1 X_2$.

Если суммировать термы, дающие нулевые значения, то получим отрицание функции и, взяв инверсию, перейдем к СКНФ.

Иногда для некоторых наборов таблицы состояний значение функции может быть неизвестно. В этом случае значение функции может быть задано произвольно нулем или единицей по Вашему усмотрению. Этот терм – безразличный набор, не влияющий на значение функции. Он необязателен. Договоримся обозначать его буквой Ф.

	$X_3 X_4$			
$X_1 X_2$	$\bar{X}_3 \bar{X}_4$	$\bar{X}_3 X_4$	$X_3 X_4$	$X_3 \bar{X}_4$
$\bar{X}_1 \bar{X}_2$	0	1	3	2
$\bar{X}_1 X_2$	4	5	7	6
$X_1 X_2$	12	13	15	14
$X_1 \bar{X}_2$	8	9	11	10

Чтобы каждый раз не рисовать карту Карно, ввели нумерацию клеток. Обычно нумерация соответствует значению десятичных чисел в простом двоичном коде. Разряды двоичного числа $X_1 X_2 X_3 X_4$. Слева старший разряд. При этом условии нумерация карты для четырех переменных такова.

Действительно, цифре 3 соответствует двоичное число $X_1 X_2 X_3 X_4$ или 0011. Числу 15 – $X_1 X_2 X_3 X_4$ или 1111 и т. д.

Функцию в дизъюнктивной форме (через единицы) задают в виде суммы номеров термов, имеющих единичное значение:

$$f = \sum(1, 3, 5, 8),$$

то есть в клетках 1, 3, 5, 8 надо поставить 1, а в остальных нули.

Функцию в конъюнктивной форме задают (через нули) в виде произведения номеров термов, имеющих нулевое значение

$$f = \prod (0, 2, 7, 11, 15),$$

то есть в клетках 0, 2, 7, 11, 15 надо поставить нули, а в остальных – единицы.

Безразличные наборы (Ф) приплюсовывают к сумме или произведению под названием ТНБ (термы, не доставляющие беспокойства, безразличные наборы).

$$f = \sum (4,5,8,11) + \text{ТНБ}(7,12,14) \text{ или } f = \prod (1,3,6,14) + \text{ТНБ}(0,2,5,13).$$

1.9. Минимизация логических функций

Под минимизацией понимают приведение алгебраического выражения функции к более простому виду. В предыдущем параграфе мы это сделали на примере функции дизъюнкции двух переменных. Обобщая преобразования, можно сказать, что существуют стандартные приемы.

1. Прибавление одного или несколько однотипных членов из числа имеющихся в данной функции.

2. Умножение отдельных членов на сумму $A + \bar{A}$, где A может быть как одной из переменных, так и частью функции или даже самой функцией. При этом по аксиоме $A + \bar{A} = 1$ и аксиоме $f \cdot 1 = f$ мы не меняем значения функции.

3. Выделение слагаемого типа $A + \bar{A}$.

4. Использование законов склеивания: $X_1 X_2 + \bar{X}_1 X_2 = X_2$, $(X_1 + X_2)(X_1 + X_2) = X_1 + X_2$. Здесь пропадает переменная, имеющая свое отрицание.

5. Применение правила де Моргана.

После всевозможных преобразований получается функция, не поддающаяся дальнейшему упрощению, – тупиковая форма. Такая функция называется минимизированной. Она может быть представлена в дизъюнктивной форме (МДНФ) или конъюнктивной (МКНФ).

Алгебраический метод требует навыков в работе и интуиции.

Рассмотрим простейший случай – функцию ИЛИ.

$$f = \bar{X}_1 X_2 + X_1 \bar{X}_2 + X_1 X_2.$$

По аксиоме тавтологии добавим терм $X_1 X_2$:

$$f = \bar{X}_1 X_2 + X_1 \bar{X}_2 + X_1 X_2 + X_1 X_2.$$

По закону ассоциативности (сочетательный):

$$f = (\bar{X}_1 X_2 + X_1 X_2) + (X_1 \bar{X}_2 + X_1 X_2).$$

По закону дистрибутивности (распределительный):

$$f = X_2 (\bar{X}_1 + X_1) + X_1 (\bar{X}_2 + X_2).$$

По аксиоме дополнительности $\bar{X}_1 + X_1 = 1$ и $\bar{X}_2 + X_2 = 1$:

$$f = X_2 \cdot 1 + X_1 \cdot 1.$$

По аксиоме пересечения: $X_2 \cdot 1 = X_2$, $X_1 \cdot 1 = X_1$:

$$f = X_2 \cdot 1 + X_1 \cdot 1.$$

По закону коммутативности $f = X_1 + X_2$.

Из-за двойственности алгебры можно эту же функцию определить через макстермы, тем более что их всего один: $\bar{f} = \bar{X}_1 \bar{X}_2$.

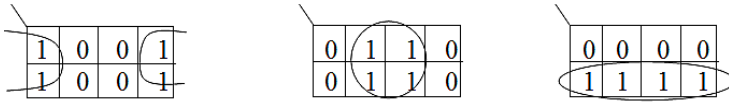
По де Моргану $\bar{\bar{f}} = \overline{\bar{X}_1 \bar{X}_2} = X_1 + X_2 = f$.

Более просто проводить минимизацию с помощью карты Карно. Требование, что при переходе от терма к терму должна меняться только одна переменная, подсказывает, что теорема склеивания применяется к соседним минтермам (функция) или к соседним макстермам (НЕ функция). Имеется в виду соседство по строкам и столбцам. Диагональное соседство не учитывается. Если внимательно посмотреть на карту, то мы увидим, что соседними являются и крайние термы таблицы по конкретной строке или столбцу, то есть – карта Карно – развертка цилиндра по горизонтальной и вертикальной оси.

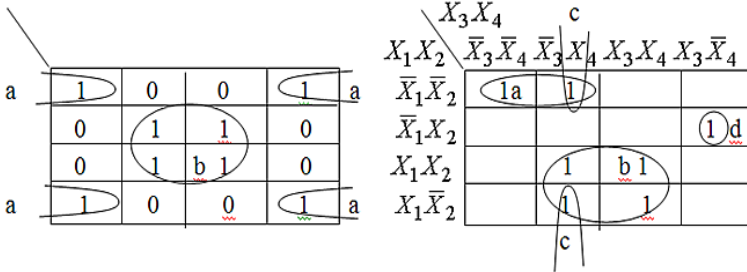
Правила склеивания:

- 1) сначала склеиваются восьмерки (если они есть);
- 2) затем склеиваются четверки (если они есть);
- 3) затем склеиваются двойки;
- 4) к полученным выражениям добавляются одиночные термы;
- 5) один и тот же терм может склеиваться сколько угодно раз (теорема тавтологии).

Примеры склеивания четверок:



Другие примеры:



При склеивании четверок (контуры *a* и *б*) пропадают две меняющиеся переменные. Из рисунка с четырьмя контурами:

Контур *a*: $\bar{X}_1 \bar{X}_2 \bar{X}_3 \bar{X}_4 + \bar{X}_1 \bar{X}_2 \bar{X}_3 X_4 = \bar{X}_1 \bar{X}_2 \bar{X}_3 (\bar{X}_4 + X_4) = \bar{X}_1 \bar{X}_2 \bar{X}_3 \cdot 1 = \bar{X}_1 \bar{X}_2 \bar{X}_3$ – пропала меняющаяся переменная X_4 .

Контур *b*: меняются переменные X_2 и X_3 . Следовательно, получим терм из неменяющихся переменных: $X_1 X_4$ (проверьте).

Контур *c* дает: $\bar{X}_2 \bar{X}_3 X_4$

Контур *d*: $\bar{X}_1 X_2 X_3 \bar{X}_4$.

Итак, минимизированная функция будет

$$f = \bar{X}_1 \bar{X}_2 \bar{X}_3 + X_1 X_4 + \bar{X}_2 \bar{X}_3 X_4 + \bar{X}_1 X_2 X_3 \bar{X}_4.$$

В случае, когда в карте Карно есть безразличные наборы Φ , вместо Φ ставят 0 или 1, исходя из удобства минимизации.

Например:

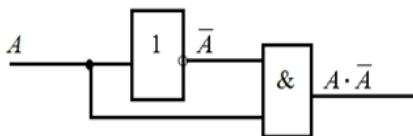
$X_1 X_2$	$\bar{X}_1 \bar{X}_2$	$\bar{X}_1 X_2$	$X_1 X_2$	$X_1 \bar{X}_2$
X_3	0	1	Φ	0
\bar{X}_3	0	1	1	0

Ясно, что выгоднее поставить 1, чтобы получить четверку, которая даст уменьшение количества переменных на 2.

С точки зрения электроники минимизация даёт более простую принципиальную электрическую схему, позволяет использовать элементы с меньшим числом входов, уменьшает количество пересечений токовых линий и габариты устройства.

1.10. Ситуация статического риска

Рассмотрим реализацию функции $f = A \cdot \bar{A}$. Для идеального элемента И это постоянный ноль.



Пусть переменная A изменяется из 0 в 1. Из-за задержки переключения инвертора на время t_3 схема даст на это время значение 1, а после переключения инвертора получается верное значение 0.

Посмотрим на осциллограммах (рис. 1.15, а):

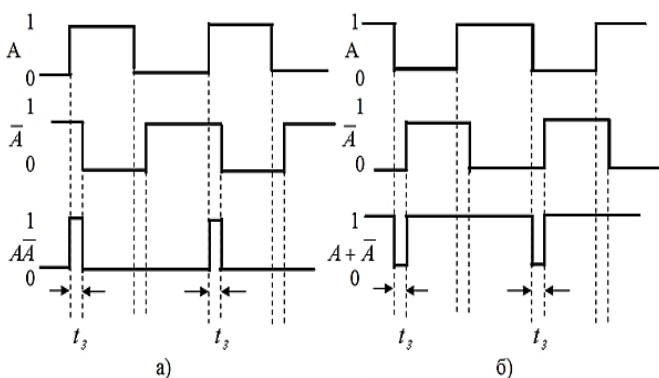
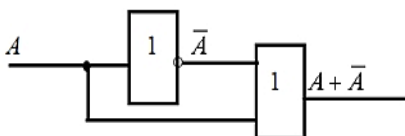


Рис. 1.15.

Мы видим, что ошибочное значение «1» получается только при изменении A из 0 в 1, а при изменении из 1 в 0 ошибки нет.

Появление ошибочного значения функции называют статическим «0-риском».

Рассмотрим реализацию функции $f = A + \bar{A}$. Это двойник предыдущей функции.



Пусть переменная A меняется из 1 в 0, а элемент НЕ имеет задержку переключения t_z . Осциллограммы процесса изображены на рис. 1.15, б.

Мы видим, что на выходе элемента ИЛИ получаются ошибочные нулевые выбросы. Это статический «1-риск».

Добавив к нашим схемам на выходе по элементу НЕ, мы получим универсальные логические элементы и проинвертированные ошибочные выбросы. Подведем итог. Во-первых:

- 1) схема И дает положительный выброс при изменении переменной из 0 в 1 – «0-риск»;
- 2) схема И-НЕ дает отрицательный выброс при изменении переменной из 0 в 1 – «1-риск»;
- 3) схема ИЛИ дает отрицательный выброс при изменении переменной из 1 в 0 – «1-риск»;
- 4) схема ИЛИ-НЕ дает положительный выброс при изменении переменной из 1 в 0 – «0-риск».

Во-вторых, если основная форма записи имеет риск, то он существует и в двойственной форме, и в дополнительной функции.

Как провести анализ функции на ситуации риска? Возьмем функцию $f = X_1 X_2 + X_1 X_3$. Мы видим, что переменная X_1 имеет отрицание, и, следовательно, возможен риск. Если принять $X_2 = 1$, $X_3 = 1$, то $f = X_1 \cdot 1 + \bar{X}_1 \cdot 1 = X_1 + \bar{X}_1$. Таким образом, если переменная X_1 будет меняться из 1 в 0, то возникнет «1-риск». Проверим по двойственной форме: $f\bar{a} = (X_1 + X_2)(\bar{X}_1 + X_3)$. Пусть $X_2 = 0$ и $X_3 = 0$: $f\bar{a} = (X_1 + 0)(\bar{X}_1 + 0)$. При изменении из 0 в 1 получим «0-риск».

Таким образом, для анализа функции на ситуацию риска можно использовать любую форму записи функции. Перебирая значения всех других переменных (0 или 1), сводят функцию к виду $A\bar{A}$ или $A + \bar{A}$.

$X_1 X_2$	$\bar{X}_1 \bar{X}_2$	$\bar{X}_1 X_2$	$X_1 \bar{X}_2$	$X_1 X_2$
X_3	0	1	1	0
\bar{X}_3	0	0	1	1
X_3	0	0	1	1

Как устранить ситуацию риска, подскажет карта Карно. Для функции $f = X_1 X_2 + \bar{X}_1 X_3$ восстановим карту Карно.

Терм $X_1 X_2$ получается при склеивании термов $X_1 X_2 X_3$ и $X_1 X_2 \bar{X}_3$ – меняющаяся переменная X_3 пропала (ставим еди-

ницы). Терм $\bar{X}_1 X_3$ получился при склеивании термов с меняющейся переменной X_2 : $\bar{X}_1 X_2 X_3$ и $\bar{X}_1 \bar{X}_2 X_3$ ставим единицы). Все остальные термы имеют нулевое значение (ставим нули). Из карты видно, что можно получить третий терм: $\bar{X}_1 X_2 X_3$, и $X_1 X_2 X_3 = X_2 X_3 (\bar{X}_1 + X_1) = X_2 X_3 \cdot 1 = X_2 X_3$. Необязательная конъюнкция $X_2 X_3$ не повлияет на значение функции, т. к. одни и те же единицы можно склеивать сколько угодно раз:
 $f = X_1 X_2 + \bar{X}_1 X_3 + X_2 X_3$.

Проверим, будет ли риск сейчас. Пусть $X_2 = X_3 = 1$. Таким образом, добавление терма $X_2 X_3$ устранило риск. Терм $X_2 X_3$ называют **термом согласования**.

Вывод. При восстановлении выражения функции по карте Карно для устранения статического риска обязательно подлежат склеиванию все соседние термы.

В литературе ситуации риска называют также гонками, гонами, опасными состязаниями.

1.11. Вопросы для самопроверки

1. Какие функции выполняют логические элементы?
2. Какие логические операции выполняют логические элементы «И», «НЕ», «ИЛИ»:
3. Какие логические функции реализуют логические элементы «И-НЕ», «ИЛИ-НЕ»?
4. Когда появляется сигнал на выходе элемента «И»: когда на каком-нибудь одном из входов есть сигнал или когда есть сигналы на всех входах?
5. Какое состояние должно быть на входе элемента «НЕ», когда выходное состояние «0»: «1» или «0»?
6. Назовите основное правило карты Карно. Что оно дает?
7. Составьте таблицу истинности и дайте названия функциям:
 $f_1 = X_1 X_2 + \bar{X}_1 \bar{X}_2$, $f_2 = \bar{X}_1 X_2 + X_1 \bar{X}_2$.
8. Функции $f_1 = X_1 \bar{X}_2$ и $f_2 = \bar{X}_1 X_2$ называются «запрет». Почему? Восстановите карты Карно этих функций.
9. Приведите примеры двойников функций двух аргументов.

10. Как алгебраически проверить функцию на ситуации риска?
11. Умеете ли вы по заданной схеме восстановить алгебраическое выражение и карту Карно функции?
12. Что такое безразличные наборы и как их используют?

1.12. Задачи для самостоятельного решения

1. Доказать аксиомы (элементарные высказывания) функции ИСКЛ ИЛИ.
2. Реализуйте функцию ИСКЛ ИЛИ в универсальных базах. Сравните.
3. Реализуйте функцию ИСКЛ ИЛИ на четырёх элементах И-НЕ.
4. Минимизируйте с помощью карты Карно функцию трёх аргументов, заданную $F = \Pi(1,2,4,6) + \text{ТНБ}(0,7)$. Постройте в базисе ИЛИ, И, НЕ принципиальную электрическую схему.
5. Минимизируйте, переведите в базис И-НЕ функцию четырёх аргументов, заданную суммой-нумерацией термов $(0,1,9,10,11)$. Постройте схему.
6. Используя карту Карно, представьте в виде разложения по минтермам (СДНФ) функцию $F(A,B,C,D) = CD + B(C + D)$.

1.13. Лабораторная работа № 1

«Синтез и анализ логической функции»

В таблице 1.4. даны варианты логических функций четырёх переменных. Вариант задаёт преподаватель.

Выпишите таблицу состояний вашей функции.

1. Запишите ее в форме СДНФ. Пусть безразличные наборы дают нулевое значение функции.
2. Получите из нее двойственную форму.
3. Запишите по таблице истинности форму СКНФ.
4. Попробуйте алгебраическими методами провести минимизацию и получите МДНФ или МКНФ по вашему усмотрению. Объясните, почему вы берете ту или иную форму и какие приёмы минимизации используете.
5. Составьте карту Карно, пронумеруйте термы, запишите функцию через сумму-нумерацию и произведение-нумерацию.

6. Задайте функцию картой Карно. Дайте значения термам ТНБ, объясните выбор.

7. Обведите и обозначьте цифрами или буквами контуры (первичные импликанты) склеиваемых термов. Обязательно укажите возможные ситуации риска, если не склеить какие-то термы. При каких значениях переменных, их изменениях и какой риск возможен? Докажите письменно. **Обязательно оставьте один риск по одной переменной.**

8. Проведите минимизацию функции с помощью карты. Сравните результат с алгебраической минимизацией.

9. Перейдите к базису И-НЕ или ИЛИ-НЕ.

10. Реализуйте полученное выражение на универсальных элементах И-НЕ или ИЛИ-НЕ по вашему выбору.

11. Прodelайте пункты 7 – 10 на ЭВМ и сверьте значения функции с таблицей истинности. Используйте программы «Синтез», «Риск», «Electronics Workbench».

12. Продемонстрируйте преподавателю ситуацию статического риска.

Таблица 1.4

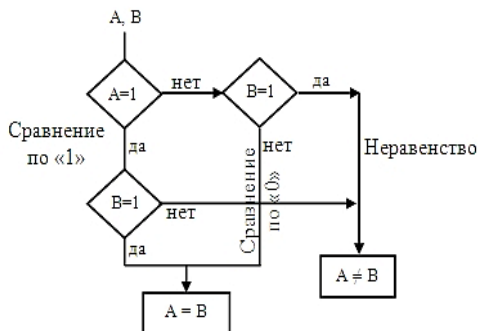
Номер набора	Значения аргументов X_1, X_2, X_3, X_4				Значения функций. Варианты: 1 2 3 4 5 6 7 8 9 10									
	0	0	0	0	0	Ф	1	0	0	1	0	1	1	1
1	0	0	0	1	0	Ф	1	0	0	1	0	1	1	1
2	0	0	1	0	1	1	Ф	1	0	0	1	1	0	1
3	0	0	1	1	1	1	1	Ф	0	1	0	1	1	0
4	0	1	0	0	0	1	0	1	Ф	0	1	0	1	1
5	0	1	0	1	1	0	1	0	0	Ф	0	1	1	1
6	0	1	1	0	0	1	0	Ф	1	0	Ф	0	Ф	0
7	0	1	1	1	1	0	Ф	1	1	1	1	Ф	1	1
8	1	0	0	0	0	Ф	0	1	Ф	0	1	Ф	Ф	0
9	1	0	0	1	Ф	0	1	0	1	0	Ф	0	0	Ф
10	1	0	1	0	0	0	1	0	0	Ф	0	0	1	0
11	1	0	1	1	1	1	1	0	Ф	1	0	1	0	0
12	1	1	0	0	0	0	0	Ф	1	0	0	1	0	1
13	1	1	0	1	0	1	Ф	1	0	1	0	1	1	1
14	1	1	1	0	0	0	1	0	1	1	1	1	1	1
15	1	1	1	1	Ф	0	0	1	1	0	1	0	1	0

ГЛАВА II. КОМБИНАЦИОННЫЕ УСТРОЙСТВА СРАВНЕНИЯ, ДЕШИФРАЦИИ И ПРЕОБРАЗОВАНИЯ КОДОВ

2.1. Компаратор

В цифровой электронике компаратор – устройство сравнения двух чисел – формирует на выходе «1», если сигналы равны, и «0», если они не равны.

Алгоритм работы компаратора можно представить в таком виде:



В связи с тем что разряд может принимать два значения, сравнение проводят по нулям и по единицам. В принципе, сравнение может проводиться не только на равенство или неравенство, но и по другим условиям. В этом смысле устройство сравнения чисел называют компаратором по условию.

Рассмотрим таблицу истинности компаратора двух одноразрядных чисел.

Числа		F1	F2	F3	F4	F5	F6
A	B	A=B	A≠B	A<B	A>B	A≤B	A≥B
0	0	1	0	0	0	1	1
0	1	0	1	1	0	1	0
1	0	0	1	0	1	0	1
1	1	1	0	0	0	1	1

Из таблицы следует:

- функция F1 это ИСКЛ ИЛИ – НЕ $F1 = \overline{AB} + AB$;
- функция F2 это ИСКЛ ИЛИ, или сумма по модулю 2:
 $F2 = \overline{AB} + \overline{AB}$;
- F3 и F4 – это ЗАПРЕТЫ: $F3 = \overline{AB}$ или $F4 = \overline{AB}$;
- F5 и F6 – ИМПЛИКАЦИИ.

Полный компаратор двух одноразрядных чисел будет представлять устройство, имеющее входы А и В и 6 выходов по числу функций F.

В случае многоразрядных чисел надо не только провести поразрядную проверку на выполнение условия сравнения, но и подвести общий итог, например, на логическом элементе И. Если в каком-либо разряде условие не выполняется, то по аксиоме конъюнкции элемент И даст на выходе ноль.

Далее будем рассматривать только условия равенства и неравенства.

$A=B$, если равны все соответствующие разряды сравниваемых чисел, и $A \neq B$, если не совпадают цифры хотя бы в одном разряде. Следовательно, функцию равнозначности (совпадения) $F_{A=B}$ и функцию неравнозначности $F_{A \neq B}$ можно представить следующим образом:

$$F_{A=B} = \bigwedge_{j=1}^n F_{jA=B}; \quad (1)$$

$$F_{A \neq B} = \bigvee_{j=1}^n F_{jA \neq B}, \quad (2)$$

где $F_{jA=B}$ – функция равнозначности j-го разряда; $F_{jA \neq B}$ – функция неравнозначности этого же разряда; символы \wedge и \vee обозначают соответственно логическое произведение и логическую сумму.

Аналогично можно записать для функции неравнозначности цифр j-го разряда (3), (4):

$$F_{jA=B} = a_j b_j + \overline{a_j} \overline{b_j}, \quad (3)$$

$$F_{jA \neq B} = \overline{a_j b_j + \overline{a_j} \overline{b_j}}, \quad (4)$$

Поскольку $F_{A=B}$ и $F_{A \neq B}$ – взаимодополняющие функции, для схемотехнической реализации может быть использована любая из них.

Рассмотрим пример реализации функций $F_{A=B}$ и $F_{A \neq B}$ на логических элементах И-НЕ. Подставив (3) в (1) и воспользовавшись двойной инверсией и правилом де Моргана, получим логическое выражение, удобное для реализации в указанном базисе:

Для реализации схемы сравнения на основании логической функции (6)

$$F_{jA \neq B} = a_j \bar{b}_j + \bar{a}_j b_j, \quad \text{либо} \quad (5)$$

$$F_{jA \neq B} = \overline{a_j b_j + \bar{a}_j \bar{b}_j}.$$

$$F_{A=B} = \bigwedge_{j=1}^n \overline{a_j b_j + \bar{a}_j \bar{b}_j}. \quad (6)$$

требуется $3n$ схем И-НЕ на два входа, одна схема И-НЕ на n входов, а также инвертор (рис. 1,а). Быстродействие такой схемы определяется суммарным временем переключения четырех последовательно соединенных элементов.

Подставим теперь (4) в (1) и перейдем к базису И-НЕ:

$$F_{A=B} = \bigwedge_{j=1}^n \overline{a_j \bar{b}_j + \bar{a}_j b_j}.$$

Для реализации этой функции требуется $2n$ схем И-НЕ на два входа, схема И-НЕ на $2n$ входов, а также инвертор (рис. 2.1,б). Быстродействие такого компаратора определяется временем переключения уже трех логических элементов.

Получим выражение для логической функции $F_{A=B}$, удобное при ее реализации на элементах И-ИЛИ-НЕ. Для этого подставим (5) в (2) и результат проинвертируем:

$$F_{A=B} = \overline{F_{A \neq B}} = \bigvee_{j=1}^n F_{jA \neq B} = \bigvee_{j=1}^n (a_j \bar{b}_j + \bar{a}_j b_j). \quad (7)$$

Чтобы выполнить функцию (7) на одном элементе И-ИЛИ-НЕ, коэффициент объединения по ИЛИ у него должен быть равен $2n$. Поэтому практически реализовать таким образом можно только схему сравнения чисел, разрядность которых не превышает

4. При большей разрядности чисел целесообразно использовать схемы И-ИЛИ-НЕ и И-НЕ. Соответствующее логическое выражение может быть получено, если к (7) применить закон инверсии.

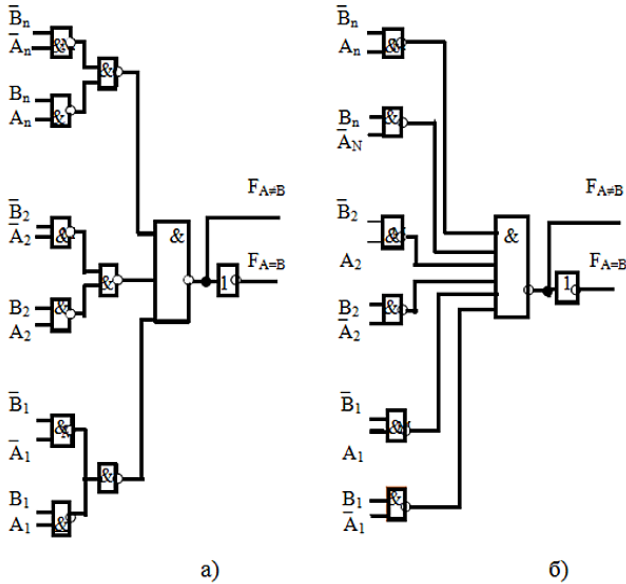


Рис. 2.1.

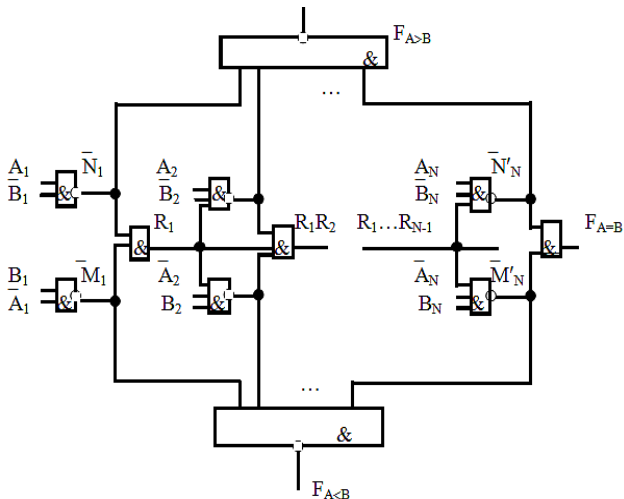


Рис. 2.2.

Здесь коэффициент объединения по ИЛИ у схемы И-ИЛИ-НЕ должен равняться двум, а коэффициент разветвления по входу у схемы И-НЕ – n .

$$F_{A=B} = \bigwedge_{j=1}^n (a_j \bar{b}_j + \bar{a}_j b_j),$$

Рассмотрим пример реализации компаратора. Пусть у нас есть два четырёхразрядных числа A и B (например, 0011 и 0101).

На чём проводить сравнение разрядов? Можно на микросхемах ИСКЛ ИЛИ-НЕ, т. е. функции равенства двух элементов. Но можно и на ИСКЛ ИЛИ, или сумма по модулю 2 ($M2$ – обозначение функции). В ТТЛ-логике К155ЛП5 содержит четыре двухвходовых элемента ИСКЛ ИЛИ.

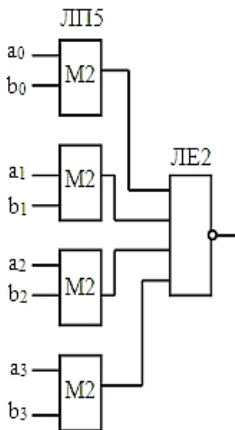


Рис. 2.3.

Но МС ЛП5 вырабатывает инверсию требуемой функции равенства (неравенство). Поэтому далее ставят 4-входовую схему ИЛИ-НЕ 155ЛЕ2 (рис. 2.3). Тогда если хотя бы в одном разряде есть неравенство, то соответствующая М2 вырабатывает «1», она поглощает в схеме ИЛИ все нули и после инверсии на выходе получим «0», т. е. компаратор покажет неравенство. На выходе будет «1» только тогда, когда все М2 выдадут нули.

Если же проинвертировать хотя бы один элемент на входе, то есть вместо A взять \bar{A} , то автоматически схема ИСКЛ ИЛИ превращается в схему равенства и теперь для сравнения разрядов надо уже использовать схему И 4-входовую. В 155 серии 4-входовой схемы И нет, но есть И-НЕ (ЛА2). К ней надо ещё добавить инвертор ЛН1 (рис. 2.4).

Если число разрядов больше 4-х (например, 8), то после ЛП5 ставят 4-входовые ЛЕ2 (4ИЛИ-НЕ) и далее К155ЛИ1 – (4 элемента 2И-НЕ). т. е., сравнив в четвёрках разрядов на элементе И, сравниваем равенство четвёрок между собой (рис. 2.5).

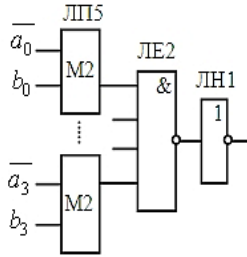


Рис. 2.4.

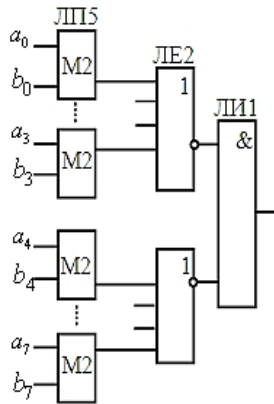


Рис. 2.5.

2.1.1. Вопросы для самопроверки

Укажите назначение цифровых компараторов.

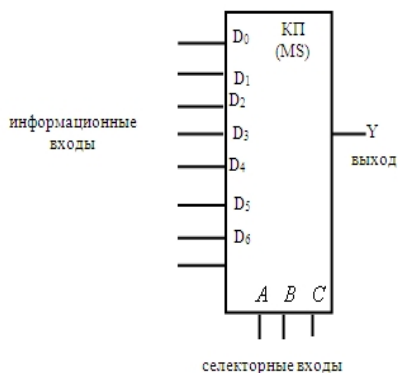
1. Постройте алгоритм работы компаратора.
2. Приведите методику (алгебру) синтеза компараторов.
3. Проведите сравнительный анализ компараторов, построенных в разных базисах по быстродействию.

2.1.2. Задачи для самостоятельного решения

1. В базисе И-НЕ синтезируйте компаратор двух трехразрядных чисел. Условие сравнения: $A < B < C$.
2. Постройте компаратор трёх двухразрядных чисел по условию.

2.2. Мультиплексоры

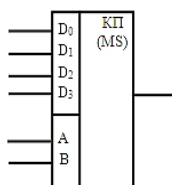
Другое название – каналные коммутаторы или селекторы. Условно мультиплексор можно изобразить следующим образом:



С помощью сигналов, подаваемых на селекторные входы (их также называют адресными), выбирается информационный вход, сигнал с которого поступает на выход. При этом передача сигналов с других входов невозможна. Адресные входы работают в параллельном коде – одновременная подача сигналов. Сигналы с каналов поступают последовательно. Это – временное мультиплексирование.

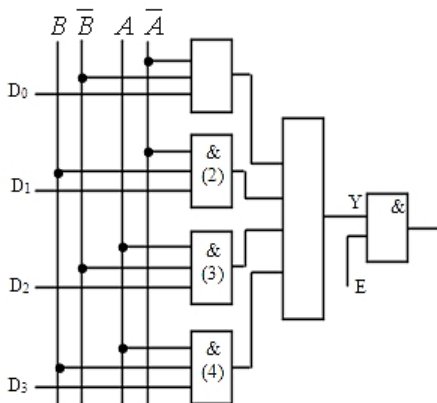
Так как код может принимать 2^n значений, то число адресных входов равно n , а число информационных входов равно 2^n , где n – количество разрядов кода.

Рассмотрим на примере мультиплексора с двумя селекторными входами ($n=2$). Он может иметь 4 информационных входа. Выбор адреса (выбор информационного канала) осуществляется подачей разрешения передачи информации от выбранного информационного входа на выход. Остальные входы не имеют разрешения. Такое можно осуществить на элементах И.



Так как переменные А и В могут давать четыре набора 00, 01, 10, 11, то адреса кодируют, например так 00 – D_0 , 01 – D_1 , 10 – D_2 , 11 – D_3 .

Рассмотрим схему мультимплексора.



Пусть $A=0, B=0$:
на 2, 3, 4 элементах И поступают «0», а на 1 – две единицы. На его выходе имеем

$$D_0 \cdot 1 \cdot 1 = D_0.$$

Если $A=0$ и $B=1$ – работает элемент 2 и т.д.

Элемент ИЛИ логически складывает сигналы. Например, для первого случая $AB=00$ имеем:

$$Y = D_0 + 0 + 0 + 0 = D_0$$

Для предотвращения ситуаций риска вводят схему разрешения. При $E=0$ («0» – активный сигнал для элемента И) выход мультимплексора закрыт.

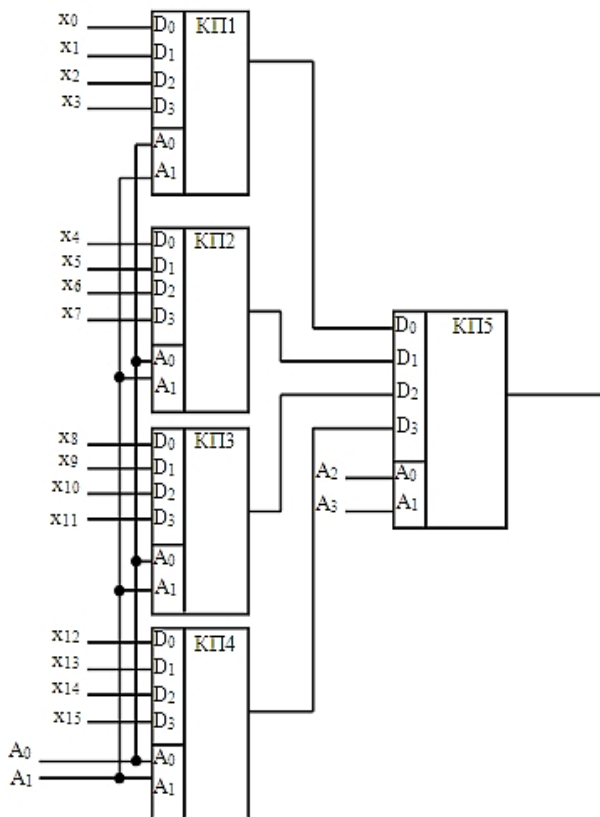
При $E=1$ – пассивный сигнал:

$$Y = D_i \cdot 1 = D_i$$

Забегая вперёд, отметим, что данный мультимплексор собран на *линейном дешифраторе* кода.

Промышленность выпускает многообразные мультимплексо-ры. Их функциональная принадлежность обозначается кодом КП или MS. Существуют мультимплексо-ры «2 в 1», «4 в 1», «8 в 1», «16 в 1». Например, в серии 555: K555КП11 – четыре КП 2-1, K555КП12 – два КП 4 – 1.

Разработаны способы увеличения числа входов мультимплек-сора, например, построением мультимплексорного дерева 16 в 1:



В данном случае мы получили двухуровневое дерево.

Примеры использования мультиплексоров

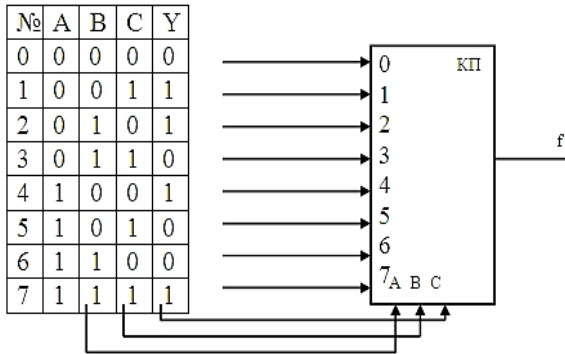
1. Генератор логической функции (последовательности).

Рассмотрим упрощённый вариант – функцию 3 переменных, заданную картой Карно:

A	$B\bar{C}$	BC		
	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
\bar{A}	0	1	0	1
A	1	0	1	0

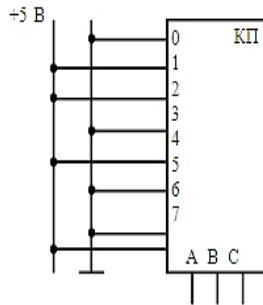
Из карты видно, что функция не минимизируется

$$f = A\bar{B}\bar{C} + \bar{A}\bar{B}C + ABC + \bar{A}B\bar{C}$$



Рисуем таблицу истинности функции, а рядом мультиплексор «8 в 1». Стрелками показываем, что куда подаётся.

Реализация – подача кода ABC на адресные входы (перебираем последовательно все возможные наборы от нулевого до седьмого), а на информационные входы подаём значения функции «1» или «0» как показано стрелками.



Итак, адресные входы используются как информационные и на них поступают наборы аргументов, а информационные входы служат для настройки селектора как настроечные.

В связи с тем, что реально существующие мультиплексоры имеют не более 16 информационных входов, спроектировать практически можно логические функции, содержащие не более четырех переменных. Поскольку никаких дополнительных логи-

ческих элементов не требуется, то мультиплексор может играть роль универсального логического элемента.

2. Рассмотрим функцию, заданную таблицей истинности:

№	A	B	C	Y
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	0

Сравнивая изменения переменных B и C, мы видим, что есть две подтаблицы, которым соответствуют $A = 0$ и $A = 1$. По схеме построения селекторов мы должны подать наборы переменных B и C на адресные входы селектора «1 из 4», а входы данных использовать как настроечные, а также для подачи значений переменной A. Следующий шаг – определение сигналов, которые должны поступать на информационные входы селектора. Назовем их D_0, D_1, D_2, D_3 . Для определения сигнала на входе D_0 таблица истинности свертывается так, чтобы под строкой № 0 оказалась строка № 4, а строки с № 1 по № 3 и с № 5 по № 7 не были видны.

№	A	B	C	Y
0	0	0	0	0
4	1	0	0	1

Ясно, что для адреса $B=C=0$, приписанного линии D_0 , значение функции совпадает с сигналом A, т.е. $D_0 = A$.

Берем следующую пару строк.

№	A	B	C	Y
1	0	0	1	1
5	1	0	1	0

Для линии D_1 ($B=0, C=1$) значение функции противоположно значению A, т.е. $D_1 = \bar{A}$.

Для третьей пары:

№	A	B	C	Y
2	0	1	0	1
6	1	1	0	1

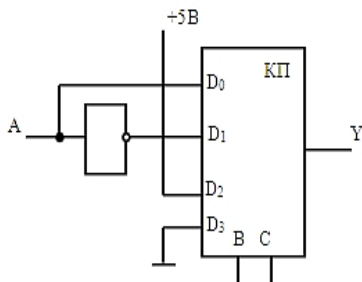
$$D_2 = 1.$$

Для четвертой:

№	A	B	C	Y
3	0	1	1	0
7	1	1	1	0

$$D_3 = 0.$$

Строим схему. Напряжению +5 Вольт соответствует логическая единица, заземлению – логический ноль.



Этот метод называется **методом свёртки**.

3. Метод остаточной (настроечной) функции

Возьмем функцию из примера 2, но заданную картой Карно.

Минимизируем ее:

BC	$\overline{B}\overline{C}$	$\overline{B}C$	$B\overline{C}$	BC
\overline{A}	0	1	0	1
A	1	0	0	1

$$Y = \overline{B}\overline{C} + \overline{A}\overline{B}C + A\overline{C}$$

Младшие переменные В и С поступают на адресные входы. Подставим наборы этих переменных в функцию и определим для каждого набора так называемую настроечную или остаточную функцию

$$Y = 1 \cdot 0 + \bar{A} \cdot 0 \cdot 1 + A \cdot 0 = 0 = D_3$$

$$Y = 0 \cdot 1 + \bar{A} \cdot 1 \cdot 0 + A \cdot 1 = A = D_0$$

$$Y = 0 \cdot 0 + \bar{A} \cdot 1 \cdot 1 + A \cdot 0 = \bar{A} = D_1$$

$$Y = 1 \cdot 1 + \bar{A} \cdot 0 \cdot 0 + A \cdot 1 = 1 = D_2$$

$$Y = 1 \cdot 0 + \bar{A} \cdot 0 \cdot 1 + A \cdot 0 = 0 = D_3$$

Составим таблицу:

Информ. вход	В	С	Настроечная функция
D_0	0	0	A
D_1	0	1	\bar{A}
D_2	1	0	1
D_3	1	1	0

Мы видим, что на входы D поступают логические константы – «1» и «0», переменная A и ее отрицание. Также видно, что эта реализация совпадает со схемой, полученной по методу свертки.

Метод применим и для функций с большим числом аргументов. Пусть $f = (A, B, C, D) = \sum (5, 9, 10, 11, 13, 14, 15)$.

CD \ AB	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

После минимизации: $f = B\bar{C}D + AC + AD$.

Находим настроечную функцию:

$$f = B \cdot 1 \cdot 0 + A \cdot 0 + A \cdot 0 = 0 = D_0$$

$$f = B \cdot 1 \cdot 1 + A \cdot 0 + A \cdot 1 = A + B = D_1$$

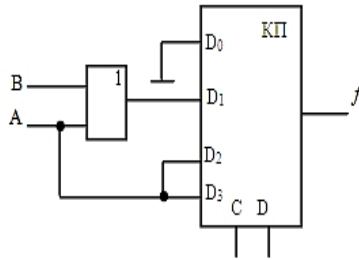
$$f = B \cdot 0 \cdot 0 + A \cdot 1 + A \cdot 0 = A = D_2$$

$$f = B \cdot 0 \cdot 1 + A \cdot 1 + A \cdot 1 = A = D_3$$

Информ. вход	C	D	Настроечная функция
D_0	0	0	0
D_1	0	1	A+B
D_2	1	0	A
D_3	1	1	A

Видно, что в этом случае на входы D поступает логическая константа «0», переменная A и некоторая функция A+B. В принципе, число аргументов может быть больше, но с увеличением их количества техническая эффективность падает.

Схема:



Приведенные примеры показывают, что такие устройства на мультиплексорах выполняют роль программируемых логических устройств.

Демультимплексор осуществляет обратную процедуру, т. е. передаёт информацию из одного канала в несколько. Например, демультимплексор «1 в 4» имеет один информационный вход, четыре выхода, два адресных входа, с помощью которых выбирается один из четырёх выходов. Такой демультимплексор можно постро-

ить на четырёх двухвходовых элементах И. На демультиплексорах для увеличения числа входов строят демультиплексорные деревья. Синтез демультиплексора рассмотреть самостоятельно.

2.2.1. Вопросы для самопроверки

1. Дайте определение мультиплексора.
2. Чем определяется количество входов мультиплексора?
3. Почему мультиплексор называют программируемым устройством?
4. В чем идея метода свёртки?
5. Что дает метод остаточной функции?
6. Какую функцию выполняет демультиплексор?

2.2.2. Задачи для самостоятельного решения

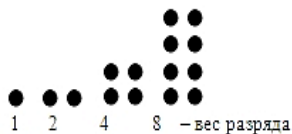
1. Постройте мультиплексорное дерево «8 в 1» на мультиплексорах «2 в 1».
2. Реализуйте мажоритарный элемент «2 из 3х» на мультиплексоре «8 в 1».
3. Выполните предыдущую задачу методом свертки.
4. Реализуйте функцию «2 из 3х» методом настроечной функции.
5. Составьте демультиплексор «1 в 4».
6. Составьте демультиплексорное дерево «1 в 8» на демультиплексорах «1 в 2».

2.3. Двоичное кодирование (коды)

Если есть одна двоичная переменная, то с ее помощью можно изобразить два числа десятичных – 0 и 1.

Если есть два двоичных числа, то, перебрав все значения, можно получить числа 0, 1, 2, 3.

Если отбросить нулевое значения, то двоичным числом X_1 можно получить 1 предмет (точка), X_1X_2 – это уже 3 предмета, т. е. добавятся 2 точки.



Числом с тремя разрядами $X_1X_2X_3$ можно получить уже 7 комбинаций (без нулевого), т. е. добавится четверка точек.

Числом $X_1X_2X_3X_4$ можно получить 15 предметов (без нулевого), т. е. добавится еще 8 точек.

Из вышесказанного получается, что каждый разряд двоичного числа имеет разные возможности или разный вес. Проставим его – напишем вес разрядов по форме записи двоичного числа: 8421. Следовательно, можно получить соответствие двоичного и десятичного числа. Это соответствие – кодирование.

Если мы будем изображать десятичные цифры в двоичном коде, то такой код можно назвать кодом 8421 или простым двоичным кодом (таблица 2.1).

Из двоичного числа легко получить десятичное, для этого нужно сложить все веса значащих разрядов:

$$\begin{array}{c} \swarrow \quad \downarrow \quad \searrow \quad \swarrow \\ 8 \quad + \quad 0 \quad + \quad 2 \quad + \quad 1 = 39 \end{array}$$

В обратную сторону:

$$39 \div 2 = 19 \text{ с остатком } 1 - \text{ разряд с весом } 1$$

$$19 \div 2 = 9 \text{ с остатком } 1 - \text{ разряд с весом } 2$$

$$9 \div 2 = 4 \text{ с остатком } 1 - \text{ разряд с весом } 4$$

$$4 \div 2 = 2 \text{ с остатком } 0 - \text{ разряд с весом } 8$$

$$2 \div 2 = 1 \text{ с остатком } 0 - \text{ разряд с весом } 16$$

$$1 \div 2 = 0 \text{ с остатком } 1 - \text{ разряд с весом } 32$$

$$\begin{array}{r} 39 = 10011 \\ \text{проверим:} \quad \begin{array}{c} \swarrow \quad \downarrow \quad \searrow \quad \swarrow \quad \downarrow \\ 32 \quad + \quad 0 \quad + \quad 0 \quad + \quad 4 \quad + \quad 2 \quad + \quad 1 = \end{array} \end{array}$$

Двоичный код *аддитивен*, т. е. сумма представлений двух чисел есть представление суммы.

Например, сложим числа 2 и 3:

$$\begin{array}{r} 0010 - 2 \\ 0011 - 3 \\ \hline 0101 - 5 \end{array}$$

Двоичный код наиболее естественный, но имеет и недостаток, связанный с переносом в следующий десятичный разряд и сложностью перехода к обратным и дополнительным кодам для десятичных чисел.

Перенос в следующий десятичный разряд.

Число «10» – это уже нули в разряде единиц и «1» в разряде десятков:

Число	разряды десятков	разряды единиц
10	0001	0000
11	0001	0001
и т. д.		

В этом отличие простого двоичного кода от двоичного представления десятичных цифр.

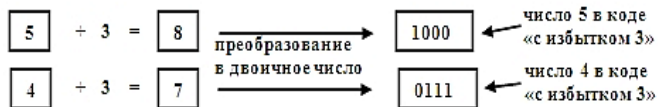
Простой двоичный код называется также *позиционным* – веса разрядов определяются занимаемыми ими местами в соответствии с ростом разрядов переменных. Позиционные коды – веса разрядов определяются занимаемыми ими местами. Непозиционные коды – веса разрядов не определяются занимаемыми ими местами. Примером такого кода является код 2421. Здесь старший разряд имеет вес 2.

Код Грея – циклический, непозиционный. Цикличность состоит в том, что все разряды, кроме самого старшего, подчиняются правилу: любая сплошная группа разрядов, считая справа, перебирает все свои возможные комбинации по некоторому закону, а затем начинает перебирать их в обратном порядке. Например: 00,01,11,10 → 10,11,01,00. Закон справедлив и для числа в один разряд: 0, 1 → 1, 0. Получается как бы зеркальное отражение. Отсюда и название – отраженный, рефлексный. Код Грея не является двоично-десятичным:

- 10: 1111
- 11: 1110
- 12: 1010
- 13: 1011
- 14: 1001
- 15: 1000
- 16: 11000
- 17: 11001
- 18: 11011

В коде «2 из 5» все комбинации содержат только две единицы. Это свойство используют для обнаружения ошибочных комбинаций.

Код «с избытком 3». В этом коде любая цифра получается добавлением к исходной цифры 3: $N + 3 = A$ и A переводится в двоичное число



Код «с избытком 3» – *самодополняющийся* код. Цифра, соответствующая любой из десятичных цифр, представляет собой инверсию кодовой комбинации, соответствующей ее дополнению до девяти. Например, в этом коде цифра «5» – это инверсия цифры «4».

Код «с избытком 3» относится к двоично-десятичным кодам, т. е. представление двузначных десятичных цифр осуществляется раздельно по сотням, десяткам и единицам. Например, цифра «15» в коде «с избытком три» – 0100 1000.

Код 2421 (код Айкена). Заполнение таблицы идет до цифры «4» как в коде 8421, но, так как вес 4-го разряда только 2, дальнейший алгоритм использовать нельзя. Вот почему идет «1» в четвертом разряде. Код имеет две формы – самодополняющуюся и несамодополняющуюся.

Таким же свойством дополнения до 9 обладает *код «3а+2»*. Кроме того, любая пара кодовых комбинаций его отличается не менее чем двумя разрядами. Отсюда применение – обнаружения ошибочных комбинаций: ошибка в одном разряде будет сразу выявлена, так как такое изменение в представлении десятичных чисел не используется.

Код 7421. Его особенность в том, что любая кодовая комбинация содержит не более двух единиц.

Все коды имеют 3 вида: прямой, дополнительный и обратный.

Дополнительный код

Он определяется выражением:

$$N^* = 2^n - N,$$

где N^* – число N в дополнительном коде, n – число цифр (разрядов) в числе N .

Например: $N_{10} = 9$
 $N_2 = 1001$
 $N^* = 2^4 - 1001 = 16 - 1001 = 10000 - 1001 = 0111$

Можно более простой алгоритм: инверсия всех разрядов числа N и добавление к младшему единицы, т. е. $0110 + 1 = 0111$. Дополнительным этот код называется из-за того, что это дополнение по основанию 2, а 2 – основание двоичной системы. Дополнительный код используют для сложения и вычитания чисел.

Обратный код определяется выражением:

$$N^{**} = (2^n - 1) - N.$$

Все обозначения как в коде N^* .

Пример:

$$N = 1011$$

$$N^{**} = (10000 - 1) - 1011 = 0100$$

Более упрощенный алгоритм – инверсия цифр всех разрядов.

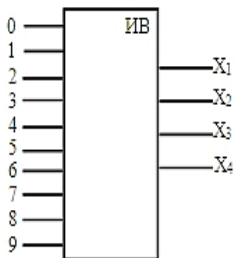
$$N = 1011$$

$$N^{**} = 0100$$

В цифровой электронике, кроме перечисленных, используется еще целый ряд специальных кодов.

Преобразователи кодов (ПК)

ПК служит для перехода из одного кода в другой. Если осуществляется перевод из десятичного кода в двоичный, то такой преобразователь кода называется шифратором.

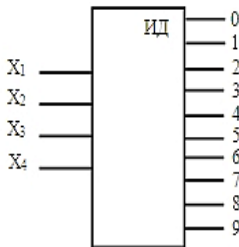


Это схематическое изображение шифратора. В логике шифраторы обозначают функциональным признаком ИВ. В шифраторе

должна быть обязательно обеспечена приоритетность входов (т. е. входы работают последовательно). Выходы работают параллельно.

Шифратор – комбинационное устройство. Так как у него 4 выхода, то он состоит из 4 функций. Их построение следует из таблиц кодов. Например, чтобы зашифровать цифру в коде 8421, необходимо реализовать 4 функции – 4 разряда. Разряд с весом 8 получается на двухвходовом элементе ИЛИ, на который подают цифры 8 и 9.

Устройство, осуществляющее обратную операцию, – дешифратор.



Обозначение микросхем – ИД. На входе дешифратора код, на выходе только одна единица, остальные нули. Например, при дешифрации цифры 7 на входе код 0111, а на выходе – единица на седьмом выходе.

Рассмотрим пример дешифратора из 4 в 10 кода 8421 в десятичный A_{10} . И сразу отметим: так как цифр десять (от 0 до 9), то число переменных на входе должно быть четыре. Составим карту Карно для четырех переменных и проставим нумерацию клеток в соответствии с простым двоичным кодом.

Мы видим, что термы 1010, 1011, 1100, 1101, 1110, 1111 (цифры 10 – 15) не используются для дешифрации. Но они могут привлекаться для упрощения представления других цифр. Эти термы – ТНБ (безразличные наборы). В клетках карты Карно вместо них ставятся нули или единицы по желанию конструктора, исходя из соображений минимизации. Так как функций десять, то заполняют десять карт Карно. В каждую карту ставят только одну единицу в клетку под номером функции. Например, для функции «3» единицу ставят в клетку под этим номером, в клетки 0, 1, 4 – 9 ставят нули, а безразличные наборы используют по правилам минимизации. В итоге после реализации получим десять конъюнкторов. Функции конъюнкторов называют переключательными функциями. В нашем случае мы использовали не все возможные наборы. Часть термов использовали как безразличные наборы. В этом случае дешифратор называется неполным.

Максимальное число выходов дешифратора определяется величиной $N=2^n$, где n – число разрядов двоичного числа. Дешифратор называется неполным, если $N < 2^n$, и полным, если $N=2^n$.

Как следует из приведенного определения, дешифратор реализует следующую систему переключательных функций:

$$\begin{aligned} Y_1 &= \bar{X}_1 \bar{X}_2 \dots \bar{X}_{n-1} \bar{X}_n, \\ Y_2 &= \bar{X}_1 \bar{X}_2 \dots \bar{X}_{n-1} X_n, \\ &\dots \\ Y_{N-1} &= X_1 X_2 \dots X_{n-1} \bar{X}_n, \\ Y_N &= X_1 X_2 \dots X_{n-1} X_n, \end{aligned} \quad (11)$$

где $X_1, X_2, \dots, X_{n-1}, X_n$ – входные переменные дешифратора, $Y_1, Y_2, \dots, Y_{n-1}, Y_n$ – его выходы.

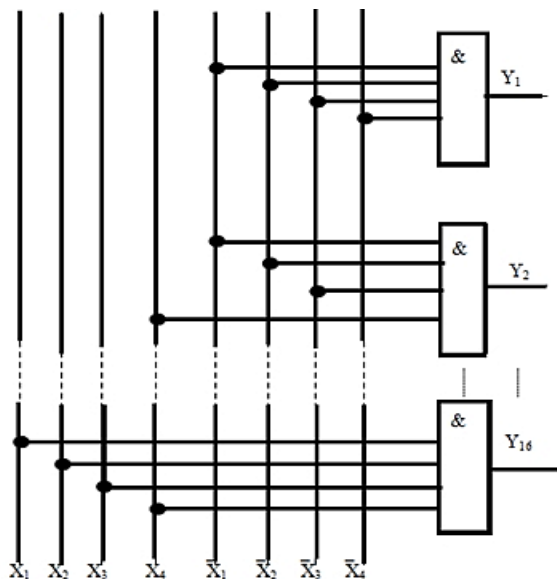
Полный дешифратор, количество входных переменных у которого равно четырем, даёт переключательные функции:

$$\begin{aligned} Y_1 &= \bar{X}_1 \bar{X}_2 \bar{X}_3 \bar{X}_4, & Y_2 &= \bar{X}_1 \bar{X}_2 \bar{X}_3 X_4, \\ Y_3 &= \bar{X}_1 \bar{X}_2 X_3 \bar{X}_4, & Y_4 &= \bar{X}_1 \bar{X}_2 X_3 X_4, \\ Y_5 &= \bar{X}_1 X_2 \bar{X}_3 \bar{X}_4, & Y_6 &= \bar{X}_1 X_2 \bar{X}_3 X_4, \\ Y_7 &= \bar{X}_1 X_2 X_3 \bar{X}_4, & Y_8 &= \bar{X}_1 X_2 X_3 X_4, \\ Y_9 &= X_1 \bar{X}_2 \bar{X}_3 \bar{X}_4, & Y_{10} &= X_1 \bar{X}_2 \bar{X}_3 X_4, \\ Y_{11} &= X_1 \bar{X}_2 X_3 \bar{X}_4, & Y_{12} &= X_1 \bar{X}_2 X_3 X_4, \\ Y_{13} &= X_1 X_2 \bar{X}_3 \bar{X}_4, & Y_{14} &= X_1 X_2 \bar{X}_3 X_4, \\ Y_{15} &= X_1 X_2 X_3 \bar{X}_4, & Y_{16} &= X_1 X_2 X_3 X_4. \end{aligned}$$

Реализация этих переключательных функций даст *линейный дешифратор*. В линейных дешифраторах каждое уравнение системы реализуется логическим элементом И.

Быстродействие линейного дешифратора определяется задержкой только одного логического элемента. Правда, предварительно потребуется ещё получить инверсию переменных.

Линейный дешифратор может быть реализован на универсальных логических элементах. В этом случае результат на выходе будет инверсным.



Прямоугольные дешифраторы. В отличие от линейных одноступенчатых дешифраторов, прямоугольные дешифраторы в простейшем случае являются двухступенчатыми, в общем – многоступенчатыми. Схема прямоугольного двухступенчатого дешифратора, реализующего переключательные функции с четырьмя входными переменными полного четырёхразрядного дешифратора, показана на рис. 2.4. На схеме все элементы типа И.

Двухступенчатый дешифратор содержит на первой ступени два линейных дешифратора (ЛД1 и ЛД2), на каждом из которых дешифруется часть кода. Во второй ступени на матричном дешифраторе (МД) из двухвходовых вентилей осуществляется совпадение выходных сигналов линейных дешифраторов. С позиции минимального количества вентилей, необходимых для реализации прямоугольного двухступенчатого дешифратора, M -разрядный входной код необходимо разбить на две равные группы ($M/2$), если M – четное, или же на две группы, отличающиеся по количеству разрядов на единицу, если M – нечетное.

Приведенный способ образования прямоугольного дешифратора не является единственным. Так, можно реализовать этот

трехступенчатый дешифратор, разбив входной код, например, на четыре группы. Каждая группа на первой ступени дешифрируется линейными дешифраторами. На второй ступени выходы линейных дешифраторов попарно объединяются по двум матричным схемам, и на третьей выходы двух матричных схем также объединяются по матричной схеме. Отметим, что увеличение количества ступеней приводит к уменьшению быстродействия дешифратора.

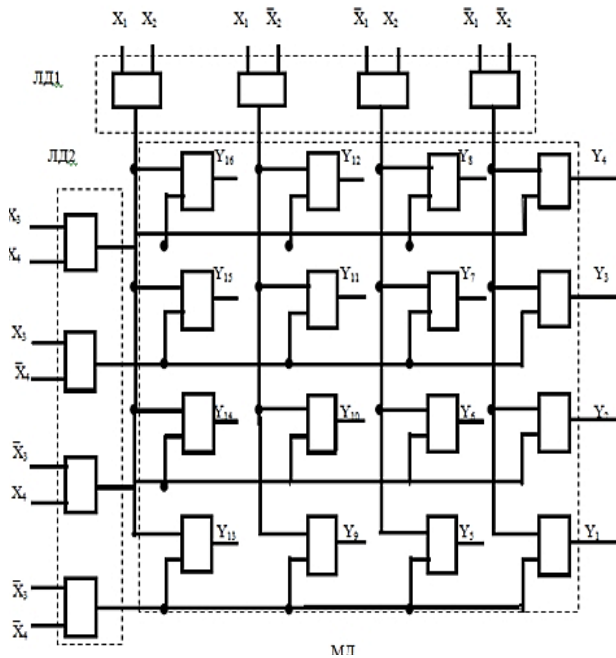


Рис. 2.4. Прямоугольный двухступенчатый дешифратор на 16 выходов

В реальной цифровой интегральной схемотехнике существуют дешифраторы с инверсными активными входами, с инверсными активными выходами, с входами разрешения, с объединением функций дешифратора и демультиплексора и другие. Ознакомиться с многочисленными устройствами дешифрации можно в справочной литературе.

Устройства, осуществляющие перевод одного двоичного кода в другой двоичный код, называются *преобразователями*

кода. Функциональный признак преобразователя код-код – ПР, двоичного кода в семисегментный – ПП (но иногда и ИД, например ИД18). Для шкальных индикаторов – ИД15, ИД16, для дискретных на фотодиодах – ИД8, ИД9.

Синтез преобразователей кодов осуществляется по обычной методике синтеза комбинационных устройств с использованием карт Карно, причем карта нумеруется во входном коде. Нумеруя таким образом клетки карты, мы осуществляем дешифрацию входного кода. Таких одинаковых карт должно быть равно числу разрядов выходного кода. Следующим шагом ставим в пронумерованные клетки нули и единицы из таблицы истинности разряда выходного кода. Этим шагом мы как бы проводим шифрацию пронумерованных клеток выходным кодом. После минимизации с использованием безразличных наборов получаем искомую функцию разряда.

2.3.1. Вопросы для самопроверки

1. Что такое «вес разряда»?
2. Перечислите свойства кодов. Приведите примеры.
3. Проиллюстрируйте принцип построения самодополняющегося кода.
4. Как строится семисегментный индикаторный код?
5. Как работает шифратор?
6. Сравните по быстродействию и аппаратурным затратам линейные, прямоугольные и пирамидальные дешифраторы.
7. В чем особенность построения прямоугольных многоступенчатых дешифраторов?
8. Каково назначение преобразователей кодов?

2.3.2. Задачи для самостоятельного решения

1. Переведите десятичное число 8 в коды 7421, 5421, 2421.
2. В коде самодополняющемся 2421 дана комбинация 1011. Какому десятичному числу она соответствует? А комбинация 0111?
3. Постройте шифратор кода Грея.
4. Постройте дешифратор кода Грея с использованием карты Карно.
5. Постройте неполный линейный дешифратор кода 7421.
6. Постройте полный линейный дешифратор кода Грея.

7. Постройте полный прямоугольный дешифратор кода 5421.
8. Синтезируйте преобразователь простого двоичного кода в семисегментный индикаторный.

2.4. Лабораторная работа № 2

1. Изучение цифрового компаратора.

1. Синтезировать компаратор двух-, трехразрядных чисел в универсальном базисе. Устройство должно показывать случаи $A > B$, $A < B$, $A = B$. Использовать ЭВМ.

2. На макете составить таблицу истинности двухразрядного компаратора. (Плата П1, карта 1-8). Объяснить принцип действий компаратора на основе функции ИСКЛ. ИЛИ.

2. Изучение мультиплексора.

С помощью макета (плата П4, карта 4-3) составить таблицу истинности, алгебраическое выражение и описать словами функцию мультиплексора. Реализовать схему на ЭВМ.

3. Изучение дешифраторов.

1. Синтезировать линейный неполный дешифратор в логическом базисе И-НЕ, осуществляющий дешифрацию состояния одного разряда десятичного счетчика. Реализовать на ЭВМ.

2. С помощью макета (Плата П4, карта 4-1) составить таблицу истинности, получить алгебраические выражения и реализовать в универсальном базисе полученное устройство. Сравнить его со стандартным дешифратором серии K155.

3. Определить функцию демультимплексора в устройстве (плата П4, карта 4-1).

4. Изучение преобразователей кода.

1. Выполнить синтез преобразователя кода 5-4-2-1 в код 8-4-2-1 в базисе И-ИЛИ-НЕ.

2. На макете (плата 4, карта 4-2) проанализировать работу преобразователя кодов на ПЗУ. (ИС K155PE3) .

5. Проделать в системе «Electronics Workbench» компьютерный эксперимент с устройствами сравнения преобразования кодов и мультиплексорами. Схемы взять из лабораторной базы данных «Electronics Workbench».

ГЛАВА III. ДВОИЧНЫЕ СУММАТОРЫ

Двоичными сумматорами называют комбинационные логические устройства, выполняющие операцию сложения двух чисел, представленных в двоичном коде. В зависимости от способа обработки чисел различают последовательные и параллельные сумматоры.

Последовательные сумматоры строятся на основе одноразрядной суммирующей схемы. В таких устройствах сложение двух чисел производится поразрядно, последовательно во времени. Параллельные сумматоры комбинационного типа представляют собой композиции одноразрядных суммирующих схем, причем обработка чисел в таких условиях осуществляется одновременно во всех разрядах. Логическое проектирование в этом случае сводится к синтезу одноразрядных сумматоров и организации цепей переноса в соответствии с заданными требованиями на быстродействие.

3.1. Одноразрядные двоичные сумматоры

Сложение основано на таблице двоичного сложения:

$$0 + 0 = 0,$$

$$0 + 1 = 1,$$

$$1 + 0 = 0,$$

$1 + 1 = 10$ – здесь происходит перенос единицы в соседний, старший разряд.

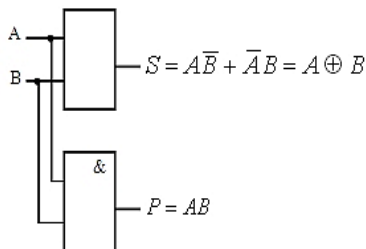
Получается сумма и перенос. Представим полученные результаты сложения в виде таблицы истинности (табл. 3.1).

На входе два одноразрядных числа A и B . Выходными сигналами такого устройства является сумма S и перенос P , которые вырабатываются согласно табл. 3.1 функционирования данного устройства.

Таблица 3.1

A	B	S	P
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Результат по S соответствует функции неравнозначности, т. е. ИСКЛ ИЛИ. Результат по переносу P – элементу И – генератор переноса. Таким образом, устройство, выполняющее операцию суммирования и определения переноса, можно реализовать так:



Или, обобщив:



Итак, полусумматор – устройство сложения в разряде единиц.

Его реализация требует наличия, во-первых, логических элементов И, ИЛИ, НЕ, во-вторых, прямых и инверсных слагаемых. В дальнейшем будем обозначать разряды слагаемых малыми буквами, имея в виду, что под заглавной буквой будет пониматься всё многоразрядное число.

Воспользовавшись правилом де Моргана можно получить выражения суммы и переноса, позволяющие построить полусумматор на универсальных логических элементах И-НЕ:

$$S = \overline{\overline{a}b\overline{a}b}, \quad P = \overline{ab}. \quad (3.1)$$

При синтезе комбинационных устройств было замечено частое повторение одних и тех же операций. Это позволило, однажды реализовав какую-то малую формулу, затем многократно её использовать, что дало возможность сократить общее количество логических элементов, необходимых для построения всего устройства. Преобразуем (3.1) таким образом, чтобы схемы полусумматора можно было построить лишь на элементах И-НЕ:

$$S = \overline{\overline{\overline{a}b} \overline{ab}}, \quad P = \overline{ab}. \quad (3.2)$$

Получив формулу для реализации полусумматора в универсальном базисе, мы всё же должны использовать как прямые, так и инверсные разряды слагаемых, т. е. парафазные сигналы. На практике такой набор кодов входных переменных приводит к увеличению аппаратных затрат и снижению быстродействия схемы.

Можно преобразовать сумму к виду, в котором присутствуют только прямые переменные a и b :

$$S = \bar{a}b + a\bar{b} = \bar{a}b + b\bar{b} + a\bar{b} + a\bar{a} = b(\bar{a} + \bar{b}) + a(\bar{a} + \bar{b}) = (a + b)(\bar{a} + \bar{b}) = (a + b)\bar{a}\bar{b} = a\bar{a}\bar{b} + b\bar{a}\bar{b}.$$

Остаётся для перехода к универсальному базису взять двойное отрицание и построить схемы суммы и переноса:

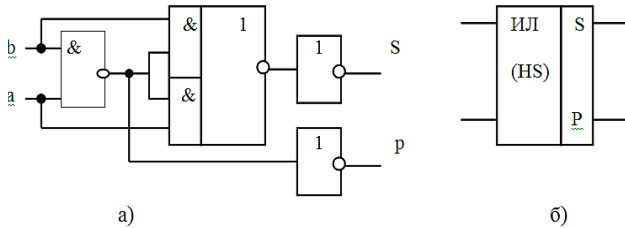


Рис. 3.1 Реализация полусумматора на элементах И-НЕ, И-ИЛИ-НЕ (а) и его условное обозначение (б)

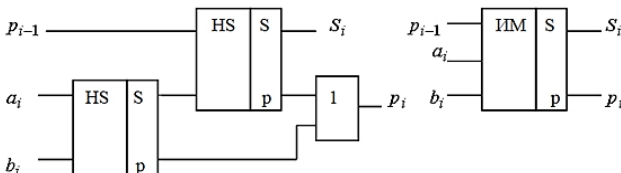
В результате реализации имеем:

$$S = a\bar{P} + b\bar{P}, \quad P = ab. \quad (3.3)$$

Соотношения (3.3) удобны при использовании базиса И-ИЛИ-НЕ, НЕ, И-НЕ (рис. 3.1). Приведем их к виду, удобному для построения полусумматора только на элементах И-НЕ:

$$S = \overline{\overline{a\bar{P}b\bar{P}}}, \quad P = \overline{\overline{ab}}. \quad (3.4)$$

Функции (3.4) реализуются на четырех двухвходовых схемах И-НЕ и одном инверторе. Одновременно отметим, что неожиданно получили устройство, которое может складывать уже не два, а три одноразрядных числа, а это уже *полный сумматор*:



Сумматор, в отличие от полусумматора, должен воспринимать не два, а три входных сигнала: два слагаемых a , b и сигнал переноса с предыдущего разряда p . Схемотехническая реализация, показанная на рис. 3.2, не является экономичной ни по числу используемых логических схем, ни по быстродействию. Поэтому схему сумматора синтезируют как единое устройство.

Составим таблицу функционирования полного одноразрядного сумматора (табл. 3.2).

Таблица 3.2

a	b	p	S	P	a	b	p	S	P
0	0	0	0	0	1	0	0	1	0
0	0	1	1	0	1	0	1	0	1
0	1	0	1	0	1	1	0	0	1
0	1	1	0	1	1	1	1	1	1

Из таблицы следует

$$\begin{aligned}
 S &= \bar{a}\bar{b}p + \bar{a}b\bar{p} + a\bar{b}\bar{p} + abp, \\
 P &= \bar{a}bp + a\bar{b}p + ab\bar{p} + abp.
 \end{aligned}
 \tag{3.5}$$

В учебниках по основам цифровой электроники, кроме карт Карно, используют другую таблицу, которую называют диаграммой Вейча. Несмотря на совершенно иной вид, диаграмма Вейча сохраняет общее с картой Карно правило: от клетки к клетке меняется только одна переменная. Это позволяет, как и в карте Карно, применять закон склеивания и упрощать алгебраическое выражение.

Нанесем минтермы логических функций на диаграммы Вейча (рис. 3.3). Макстермы (нулевые значения функции) не прописываем. Из диаграмм Вейча видно, что выражение для функции S не упрощается посредством склеивания. Существенно уменьшить количество элементов при реализации этих функций за счет выделения общих частей также нельзя, так как функции имеют только один общий минтерм abp . Упростить функцию переноса P можно, если выбрать контуры так, как указано на рис. 3.3, б.

В результате получим

$$S = abp + a\bar{b}\bar{p} + \bar{a}b\bar{p} + \bar{a}\bar{b}p, \quad P = ab + ap + bp.
 \tag{3.6}$$

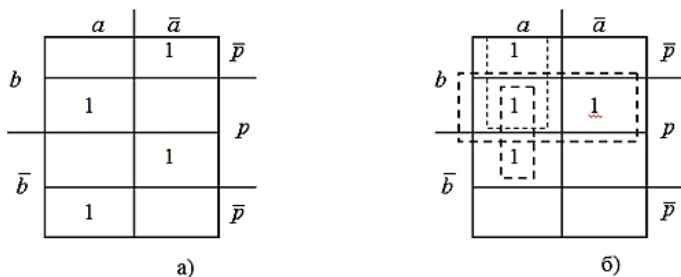


Рис. 3.3. Диаграммы Вейча для функций суммы (а), переноса (б).

Для реализации сумматора по соотношениям (3.5) либо (3.6) требуются схемы И, ИЛИ, НЕ, а также парафазные входные сигналы.

Таблица 3.3

P	a	b	p	S	P	a	b	p	S
0	0	0	0	0	1	0	0	0	x
0	0	0	1	1	1	0	0	1	x
0	0	1	0	1	1	0	1	0	x
0	0	1	1	x	1	0	1	1	0
0	1	0	0	1	1	1	0	0	x
0	1	0	1	x	1	1	0	1	0
0	1	1	0	x	1	1	1	0	0
0	1	1	1	x	1	1	1	1	1

Можно построить более экономичную схему, если преобразовать функцию суммы таким образом, чтобы функция переноса либо ее инверсия входила явно в функцию S . Этот прием упрощения предполагает, что функция P уже реализована и мы хотим добиться упрощения структуры для S , имея в качестве базовой схемы логическую схему функции P . Считая перенос P четвертой независимой переменной, построим таблицу истинности для S (табл. 3.3). В таблице избыточные комбинации, т. е. такие, которые не могут встретиться при работе, отмечены крестиками. Функция S по отношению к набору переменных a, b, p, P является не полностью определенной. Доопределим эту функцию таким образом, чтобы ее максимально упростить. Для этого нанесем минтермы функции S на диаграмму Вейча (рис. 3.4).

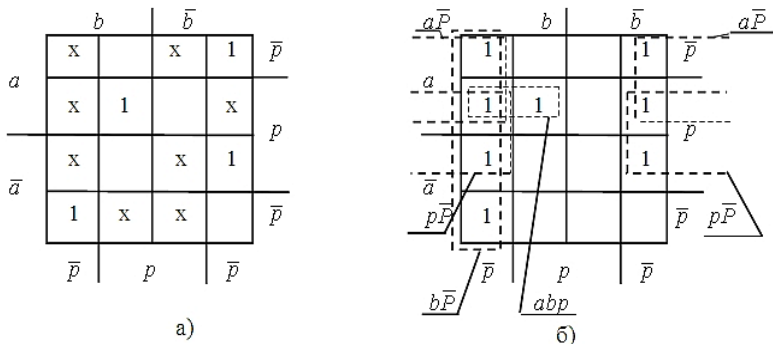


Рис. 3.4. Диаграммы Вейча недоопределенной функции сумматора (а) и вариант ее доопределения (б)

Из диаграммы Вейча следует

$$S = abp + \bar{P}(a + b + p).$$

Этот результат можно интерпретировать следующим образом. По таблице функционирования сумматора $S = 1$, если все три входные переменные равны 1 либо только одна из указанных переменных равна 1, а две другие равны 0. В последнем случае перенос $P = 0$, поэтому логическое произведение $\bar{P}(a + b + p)$ позволяет установить ситуации, когда лишь одна из входных переменных равна 1. Окончательно преобразованные логические функции S и P примут вид

$$S = abp + \bar{P}a + \bar{P}b + \bar{P}p, \quad P = ab + ap + bp. \quad (3.7)$$

Функции (3.7) удобно реализовать в базисе И-ИЛИ-НЕ (рис. 3.5, левая часть).

Положительным свойством такой реализации сумматора, кроме экономичности, является его высокое быстродействие, так как $t_{3S} = 2t_3$, $t_{3P} = 2t_3$, а также наличие только однофазных входов. Именно поэтому он находит широкое применение в интегральной схемотехнике, например в сумматоре ИМ2. В схеме рис. 3.5 правая часть не имеет отношения к уравнениям 3.7. Микросхема ИМ2 – сумматор двухразрядных чисел. В нем второй разряд организован так, чтобы задержки формирования суммы и переноса у обоих разрядов были одинаковы $2t_3$. Логические функции вто-

рого разряда получаются, если предварительно проинвертировать входные сигналы, получить \bar{a} , \bar{b} , \bar{p} :

$$S = \bar{b}P + \bar{p}P + \bar{a}P + \bar{a}\bar{b}\bar{p}, \quad P = \bar{a}p + \bar{a}\bar{b} + \bar{b}\bar{p}.$$

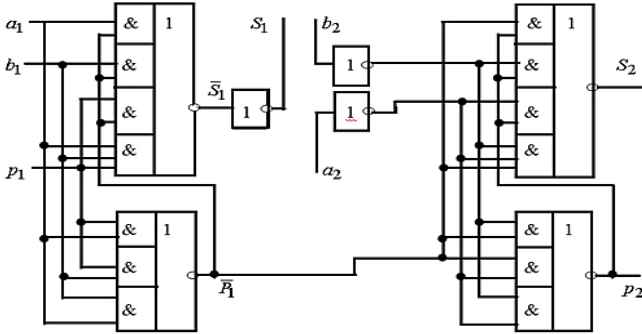


Рис. 3.5. Двухразрядный сумматор ИМ2

На основании соотношений (3.7) реализована и схема одноразрядного сумматора ИМ1 серий ТТЛ (рис. 3.6). Однако в нем предусмотрена возможность работы с прямыми, инвертированными и парафазными сигналами. Кроме того, предусмотрена возможность тактирования, блокировки и т. д. На рис. 3.6. a_1 , a_2 и b_1 , b_2 – входы, на которые могут подаваться прямые сигналы, а также осуществляться тактирование или разрешение; a_4 и b_4 – входы, на которые могут подаваться инверсные сигналы. Такая универсальность сумматора привела к снижению его быстродействия и в случае прямых входных сигналов составляет $4t_3$.

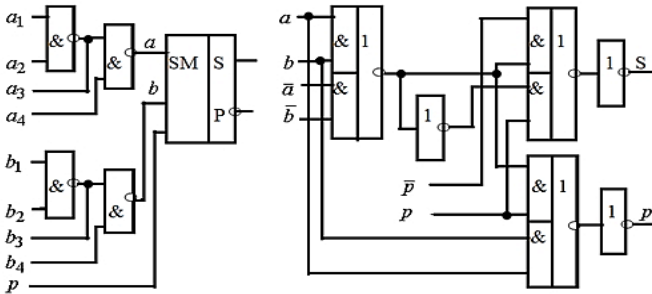


Рис. 3.6. Одноразрядный сумматор ИМ1 (слева) и одноразрядный сумматор с парафазными входными сигналами (справа)

Рассмотрим некоторые реализации схем сумматоров в случае парафазных входных сигналов. Для получения экономичной схемы преобразуем функции (3.6) таким образом, чтобы максимально выделить общие члены:

$$S = abp + a\bar{b}\bar{p} + \bar{a}b\bar{p} + \bar{a}\bar{b}p = (ab + \bar{a}\bar{b})p + (a\bar{b} + \bar{a}b)\bar{p} = (ab + \bar{a}\bar{b})p + \overline{(a\bar{b} + \bar{a}b)}\bar{p},$$

$$P = ab + ap + bp = ab + p(a + b) = ab + p\overline{\bar{a}\bar{b}} = ab + \overline{(ab + \bar{a}\bar{b})}p. \quad (3.8)$$

Схема, в основу которой положены логические функции (3.8), приведена на правой части рис. 3.6.

Преобразуем функции (3.8) к виду, удобному для реализации в логическом базисе И-НЕ, воспользовавшись двойной инверсией и правилом де Моргана:

$$S = \overline{\overline{ab\bar{a}\bar{b}} \overline{pab\bar{a}\bar{b}} \bar{p}}, \quad P = \overline{\overline{ab\bar{a}\bar{b}} \overline{pab}}. \quad (3.9)$$

Для построения сумматора на основании соотношений (3.9) требуются две схемы И-НЕ на три входа и шесть таких схем на два входа. Быстродействие сумматора определяется задержкой четырех логических элементов.

Функции (3.9) можно преобразовать в формулы без инверсии входных сигналов, а также для реализации в логическом базисе И-ИЛИ-НЕ, распространенном в интегральной схемотехнике.

Как и все комбинационные схемы, сумматоры для запоминания результатов сложения на выходе требуют установки элементов памяти, например регистров.

3.2. Многоразрядные сумматоры

Сумматоры такого типа предназначены для суммирования нескольких многоразрядных чисел. Они различаются способами передачи сигналов переноса от младших разрядов сумматора к старшим. Различают комбинационные сумматоры с последовательным, одновременным и комбинированным переносом. Тип переноса между разрядами суммирующего устройства определяет его быстродействие.

Для сложения многоразрядных чисел необходимо брать такое количество сумматоров, сколько разрядов в числе, и учесть еще возможный перенос в старшем разряде.

На примере трехразрядного сумматора:

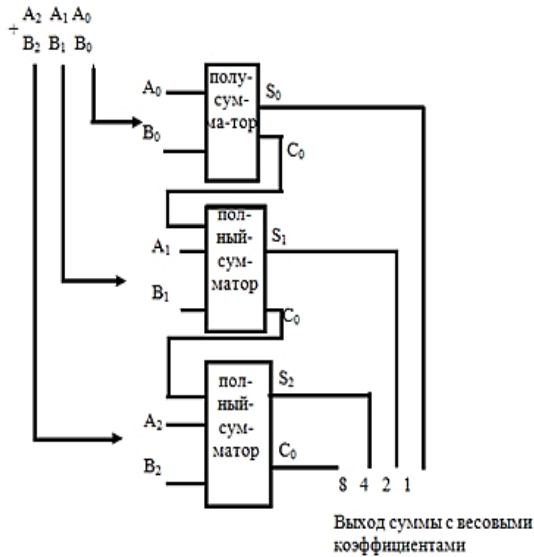


Рис. 3.7. Простейший параллельный сумматор

Здесь сигналы на входы поступают одновременно – это *параллельный* сумматор. В нем осуществляется *последовательный* перенос. Это простейший из параллельных сумматоров. На входы каждой одноразрядной схемы сумматора поступают два слагаемых и перенос из предыдущего разряда. Сигнал переноса, образованный в младшем разряде, распространяется последовательно по цепям переноса к старшим разрядам. Время распространения переноса определяется суммарной задержкой этих цепей.

Сумматор n двухразрядных чисел показан на рис. 3.8.

Быстродействие сумматора при сложении двух n -разрядных чисел характеризуется временем суммирования, которое в наилучшем случае равно

$$t_{\Sigma} = (n - 1)t_p + t_{1\Sigma},$$

где $t_{1\Sigma}$, t_p – задержки формирования одноразрядным сумматором суммы и переноса соответственно. Следовательно, сумматоры с последовательным переносом обладают низким быстродействием.

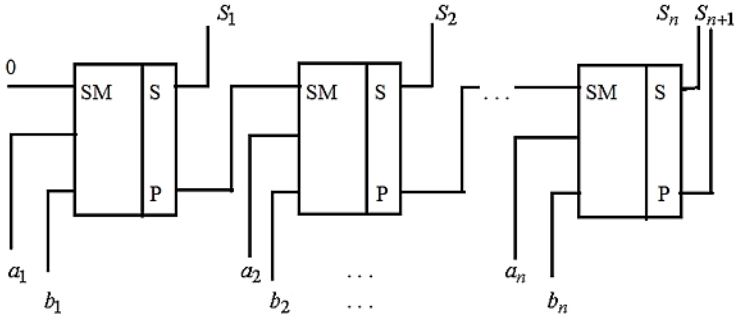


Рис. 3.8. Комбинационный n-разрядный сумматор с последовательным переносом

С целью повышения быстродействия применяются сумматоры с *одновременным переносом*. Одновременным называется такой перенос, при осуществлении которого рассматриваются цифры слагаемых, поступающих в младший разряд сумматора; на основании анализа этих цифр одновременно вырабатываются переносы во все разряды сумматора. Переносы поступают на соответствующие суммирующие схемы, которые дают правильную сумму. Быстродействие такого сумматора определяется следующей величиной: $t_{\Sigma} = t_{1\Sigma} + t_{0P}$, где t_{0P} , $t_{1\Sigma}$ – задержки формирования переносов и суммы одноразрядным сумматором соответственно.

Этот метод реализован на практике в четырехразрядном сумматоре ИП4 (можно посмотреть в соответствующих справочниках). Обозначение микросхемы ИП говорит о том, что она относится к арифметико-логическим устройствам (АЛУ).

Серьезным недостатком сумматоров с одновременным переносом является значительный рост аппаратных затрат при построении схем межразрядных переносов. Поэтому на практике применяют метод построения сумматоров с *комбинированным переносом*.

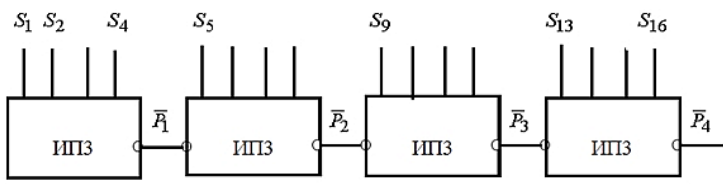


Рис. 3.9. 16-разрядный сумматор с комбинированным переносом

В сумматорах с комбинированным переносом полные одно-разрядные сумматоры объединяются в группы. Внутри группы осуществляется одновременный перенос, а между группами может быть как последовательный, так и одновременный. На рис. 3.9 приведена схема 16-разрядного двоичного сумматора с комбинированным переносом на базе четырехразрядных схем сумматоров ИПЗ.

Для удобства построения сумматоров с одновременным переносом в ряде серий микросхем логических функций, формирующих переносы, реализуются в виде отдельной микросхемы. В ТТЛ логике такой микросхемой является ИП4 – блок ускоренного переноса для арифметического узла на четыре разряда.

В схеме рис. 3.10 одновременный перенос осуществляется как внутри каждой группы, так и между отдельными группами.

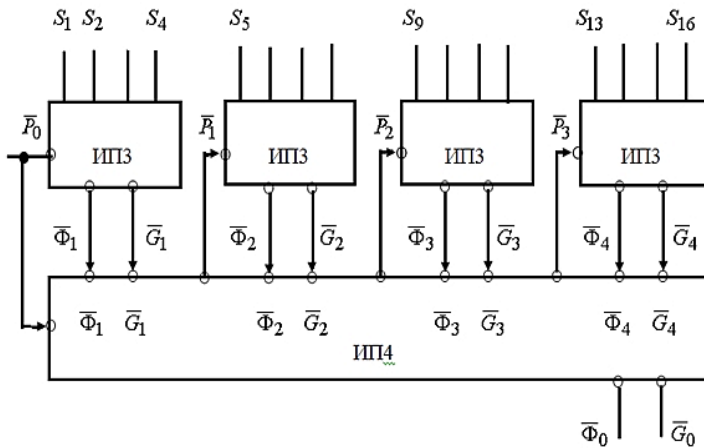


Рис. 3.10. Схема 16-разрядного сумматора на основе микросхем ИПЗ, ИП4

В данной схеме

$$\bar{G}_i = \overline{a_2 b_2 + a_1 b_1 (a_2 + b_2)} = \overline{a_2 b_2 + a_1 a_2 b_1 + a_1 b_1 b_2},$$

$$\bar{\Phi}_i = \overline{(a_1 + b_1)(a_2 + b_2)} = \overline{a_1 a_2 + a_1 b_2 + b_1 a_2 + b_1 b_2}.$$

На основе микросхемы ИП4 можно строить сумматоры с одновременным переносом и на большее количество разрядов

(рис. 3.11). Использование комбинированного переноса в много-разрядных сумматорах позволяет проектировать схемы, наиболее полно отвечающие предъявляемым требованиям к быстродействию и приемлемые в отношении аппаратных затрат.

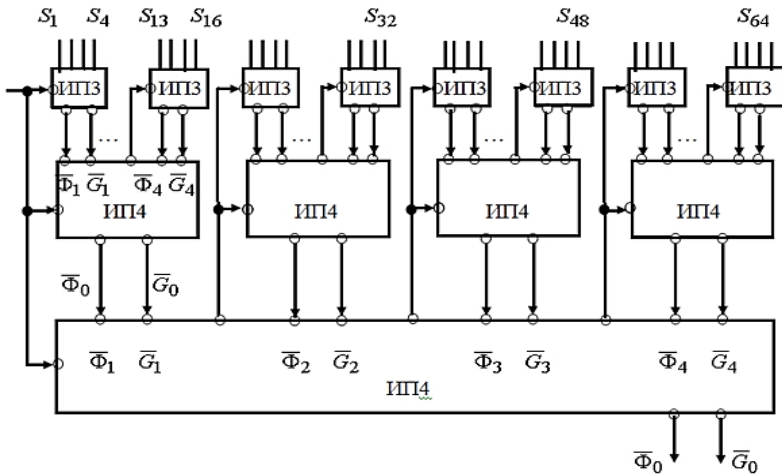


Рис. 3.10. 64-разрядный двоичный сумматор с одновременным переносом

3.3. Вопросы для самопроверки

1. Дайте определение полусумматора, запишите его таблицу функционирования.
2. Приведите реализации полусумматоров в различных логических базисах.
3. Запишите таблицу функционирования полного одноразрядного сумматора.
4. Назовите различные приемы преобразования логических функций с целью получения экономичных схем полных одноразрядных сумматоров.
5. Сравните логические базисы И-НЕ, ИЛИ-НЕ, И-ИЛИ-НЕ с позиций быстродействия реализованных на их основе двоичных сумматоров.
6. Приведите различные схмотехнические реализации полных одноразрядных сумматоров.

7. Перечислите способы организации переноса в параллельных многоразрядных сумматорах.

8. Определите понятия выработанного и сквозного переносов.

9. Укажите принцип построения сумматоров с последовательным переносом, оцените их быстродействие.

10. Охарактеризуйте способ реализации сумматоров с одновременным переносом, перечислите преимущества и недостатки таких сумматоров.

11. Каким образом строятся сумматоры с комбинированным переносом?

12. Постройте сумматоры с комбинированным переносом на основе микросхем одно-, двух- либо четырехразрядных сумматоров и микросхемы ускоренного переноса ИП4.

3.4. Лабораторная работа № 3

1. Изучение полусумматора.

Постройте идеализированные временные диаграммы полусумматора, реализованного в базисе И-НЕ, если на входы подаются сигналы, приведенные на рис. 3.14,а.

2. Изучение одноразрядного сумматора.

Используйте на универсальном стенде плату П1, карту I-9. Схема сумматора указана на карте. Базис ИЛИ-НЕ.

а) Получите таблицу функционирования сумматора.

б) Постройте идеализированные временные диаграммы, если на входы подаются сигналы, приведенные на рис. 3.14, б.

3. Используя плату П3 и карту Ш-3, исследуйте на макете четырехразрядный параллельный сумматор. Проведите суммирование 4-5 пар четырехразрядных двоичных чисел (операндов). Формирование и промежуточное хранение операндов осуществляется с помощью регистра RG и тумблеров SA4-SA1. Суммирование производить с учетом сигнала переноса, который задается тумблером SB3. Правильность операций проверяйте переводом значений операндов в десятичный код.

4. Определите, какое максимальное число можно получить в результате сложения на параллельном четырехразрядном сумматоре.

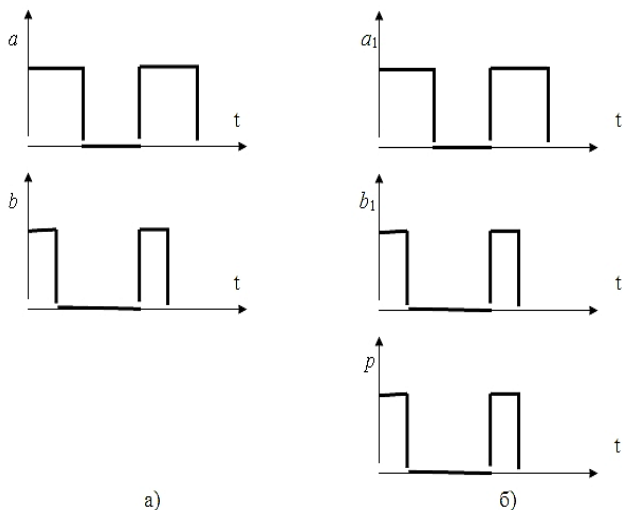


Рис. 3.14. Входные сигналы для полусумматоров (а) и сумматоров (б)

5. Проведите вычитание 3-4 пар четырехразрядных операндов, используя предварительно перевод вычитаемого в дополнительный код по формуле $N_d = 2^n - N$, где N_d – число N в дополнительном коде, n – число цифр (разрядов) в числе N . Можно воспользоваться также простым алгоритмом:

- 1) перевод числа N в обратный код (инверсия всех разрядов);
- 2) добавление к младшему разряду единицы.

Просмотр состояний входов и выходов проводить с помощью устройства цифровой индикации в режимах представления информации в двоичном и шестнадцатеричном кодах.

Примечание

Карта Ш-3 предназначена для исследования работы сумматора двух четырехразрядных двоичных чисел на ИМС типа К155ИМ3 (D2). Для задания двух четырехразрядных слагаемых используются тумблеры SA1-SA4 (слагаемое A) и регистр D1 (слагаемое B). Ввод слагаемого B в регистр может осуществляться как в параллельном (с тумблеров SA1-SA4), так и в последовательном (с кнопки SB2) режимах.

Последовательный режим осуществляется при сигнале управления $V=0$ (SA5 в положении «0»). Входная информация поступает в регистр без индикации кнопкой SB2. Кнопка «отпущена» $D=1$, кнопка «нажата» $D=0$. Выходная информация выводится на светодиоды HL6-HL9.

Параллельный режим загрузки регистра осуществляется сигналом $V=1$ (тумблер SA5 в положении «1»). В этом случае входная информация индицируется светодиодами HL2-HL5, а выходная – HL6-HL9.

Внимание! Регистр имеет два тактовых входа G_1 и G_2 . От любого из пяти информационных входов (SA1-SA4, SB2) код поступает на выходы регистра синхронно с отрицательным перепадом, вырабатываемым с помощью кнопки SB1.

Сигнал начального переноса P_0 подается с кнопки SB3 через инвертор D2.3, имеющийся в основном блоке. Индикация входных операндов А и В осуществляется двоичным восьмиразрядным индикатором HL2-HL9. Сигнал переноса в старший (пятый) разряд подается на индикатор HL1. Сумма в четырех первых разрядах дешифруется на блоке индикации и высвечивается в виде шестнадцатеричного числа на семисегментном индикаторе.

ГЛАВА IV. ТРИГГЕРЫ

4.1. Общие положения

Триггер как устройство относится к генераторам прямоугольных импульсов. Импульс на выходе таких генераторов выглядит так, как показано на рис. 4.1. Здесь τ_1 – длительность высокого уровня τ_2 – длительность низкого уровня.

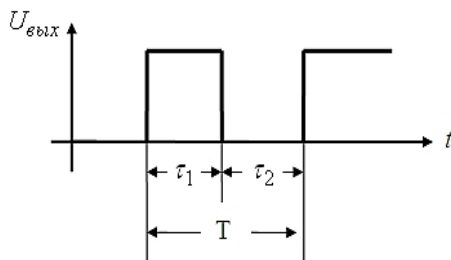


Рис. 4.1.

Говоря о цифровых устройствах, можно высокому уровню поставить в соответствие логическую единицу, а низкому – логический ноль. Для триггера состояние «1» или «0» на выходе является устойчивым, стабильным. Отсюда название «бистабильный генератор». Если устойчивым является только одно состояние, а другое – временным, то это моностабильный генератор. Устойчивое состояние будет всегда, если нет внешних воздействий. При внешнем воздействии генератор переходит в другое состояние, сохраняющееся только во время внешнего воздействия. Такие генераторы называют еще ждущими или заторможенными. Генераторы такого типа нашли применение в осциллографах при работе с внешней синхронизацией.

Третий тип генераторов прямоугольных импульсов – это астабильный генератор, или автогенератор. Их называют также мультивибраторами. У мультивибраторов устойчивые состояния отсутствуют.

Возвращаемся к триггеру. С точки зрения выходного импульса мы имеем дело с устройством, обеспечивающим скачкоо-

бразный переход из одного состояния в другое. Отсюда и название – триггер, что дословно в переводе с английского означает «спусковой крючок огнестрельного оружия».

Триггер как логическое устройство из-за стабильных, устойчивых состояний может служить элементом памяти и, таким образом, является последовательной схемой.

Основу триггера составляет так называемая бистабильная ячейка **БЯ** (англ. «*flip-flop*»). В качестве выходов берутся коллекторы транзисторов (рис. 4.2). Схема электрически симметричная. В ней также действуют взаимно-перекрестные обратные связи. Однако полная симметрия невозможна из-за различия параметров схемы при их изготовлении. Рассмотрим принцип действия **БЯ**. При включении питания пусть T_1 начинает приоткрываться чуть быстрее T_2 . Напряжение $U_{кэ_1}$ уменьшается быстрее, чем $U_{кэ_2}$. Так как T_2 управляется уменьшающимся напряжением $U_{кэ_1}$, то он начинает приоткрываться. Его напряжение $U_{кэ_1}$ растет, а так как оно поступает на базу T_1 , то T_1 еще быстрее начинает открываться, а T_2 закрывается. В результате T_1 входит в режим насыщения, т. е. полностью открывается, а T_2 в режим отсечки, т. е. закрывается.

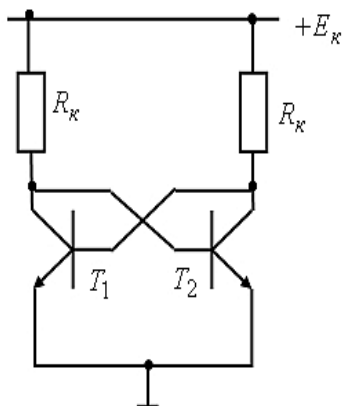


Рис. 4.2.

Лавинообразное нарастание тока в одной половине **БЯ** и уменьшение в другой называется регенерацией. Мы исходили

из того, что T_1 открывается чуть больше T_2 , но это не обязательно. В **БЯ** любое исходное состояние является равновероятным.

Так как T_1 и T_2 работают в ключевом режиме, имеют схему включения с общим эмиттером, то они могут рассматриваться как инверторы. Поставив в соответствие высокому уровню логическую «1», а низкому – логический «0», мы получим **БЯ** на логических элементах – инверторах (рис. 4.3, а) или в двойственной форме (рис. 4.3, б).

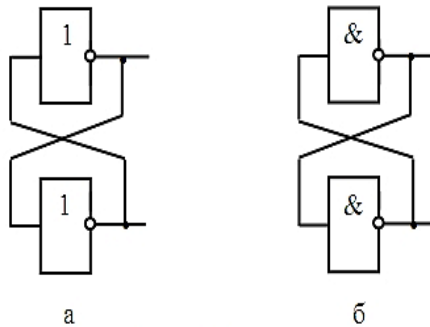


Рис. 4.3.

С помощью внешних сигналов можно управлять состоянием бистабильной ячейки. Различают два способа управления: режим отдельных входов и режим общего входа.

Режим отдельных входов

В этом режиме параллельно каждому транзистору подключают еще по одному транзисторному ключу (рис. 4.4).

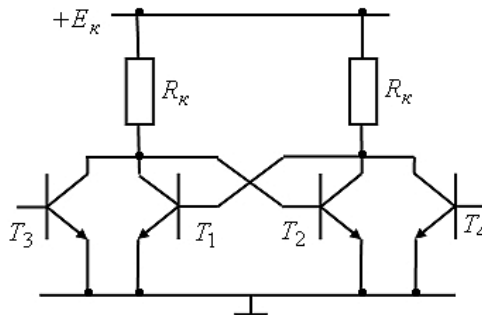


Рис. 4.4.

Эти ключи управляются внешним сигналом «1», задающим $I_b > 0$ или «0» $I_b = 0$. Пусть в исходном состоянии T_1 заперт, а T_2 открыт и насыщен. Оба ключа T_3 и T_4 заперты. Если отпереть T_3 , то U_k и U_{b2} падают до нуля, а, значит, T_2 запирается. При этом благодаря регенерации T_1 отпирается до насыщения. После запуска процесса регенерации ключ T_3 теряет свое управляющее действие и его «замыкание» или «размыкание» не изменит состояния **БЯ**.

Чтобы вернуть **БЯ** в исходное состояние, необходимо подать «1» (т.е. $+I_b$) на T_4 .

Если подать управляющий сигнал сразу на два входа, то на базе T_1 и на базе T_2 будет нулевой потенциал и T_1 и T_2 будут заперты. Это состояние электрической симметрии временное, и из него **БЯ** с равной вероятностью может перейти в любое устойчивое состояние. Результат действия управляющих сигналов оказывается заранее непредсказуемым. Следовательно, подача управляющего импульса сразу на два входа недопустима.

Такой вариант бистабильной ячейки получил название **RS-триггера** (**S** от set – устанавливать; **R** от reset – переключение). Сигнал на входе **S** устанавливает на выходе единицу, а сигнал на входе **R** меняет состояние выхода на нулевое.

Введем обозначения выходов. Пусть **S** вход – вход транзистора T_3 . Тогда если на него подать единицу (рис. 4.5), то на выходе $T_2 T_4$ установится высокий уровень. Поэтому назовем этот выход **Q**. На выходе $T_1 T_3$ при этом будет низкий уровень, т. е. \bar{Q} .

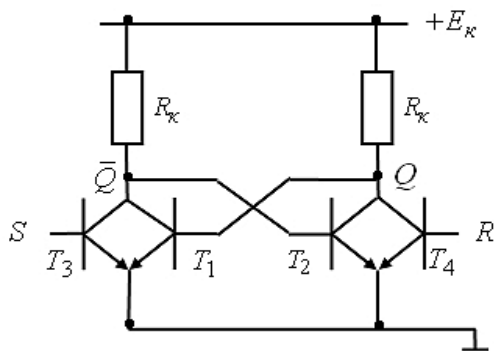


Рис. 4.5.

Можно изобразить алгоритм работы **RS**-триггера, как показано на рис. 4.6.

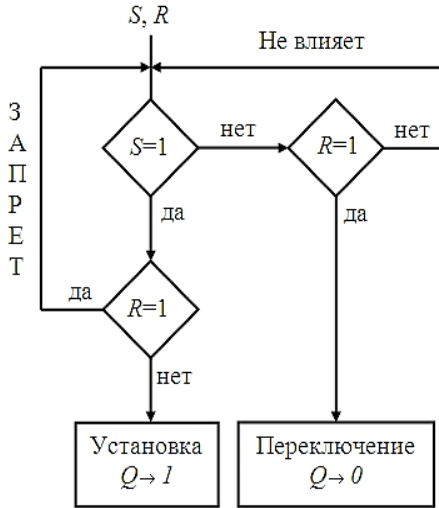


Рис. 4.6.

Мы видим, что $R = S = 1$ должно быть исключено, т. к. схема переходит в чистую бистабильную ячейку.

$S = 1, R = 0$ устанавливает $Q = 1$, а $\bar{Q} = 0$;

$S = 0, R = 1$ устанавливает $Q = 0$, а $\bar{Q} = 1$;

$S = 0, R = 0$ не влияет, т. е. не меняет исходного состояния.

Отсюда можно записать таблицу переключений (*таблицу состояний*) – табл. 4.1.

Таблица 4.1.

S	R	Q	\bar{Q}
1	0	1	0
0	1	0	1
0	0	предыдуш	
1	1	?	?

запрещено

Следует отметить, что при описании последовательностных устройств из-за того, что существуют исходное и последующее состояния, принято называть таблицу не таблицей истинности, а таблицей переключений. **RS**-триггер по этой же причине является функцией трёх аргументов R , S и Q .

Режим общего входа

Мы получили, что подача управляющего сигнала на оба входа сразу недопустима. Однако это верно лишь в том случае, если у нас простая **БЯ**. Если же **БЯ** имеет внутреннюю память, т. е. состояние **БЯ** сохраняется во время действия управляющего сигнала и лишь при его прекращении **БЯ** переходит в новое устойчивое состояние, то **БЯ** будет работать в режиме общего (объединенного в один) входа.

В качестве запоминающих устройств используют емкости. Схема такого триггера показана на рис. 4.7.

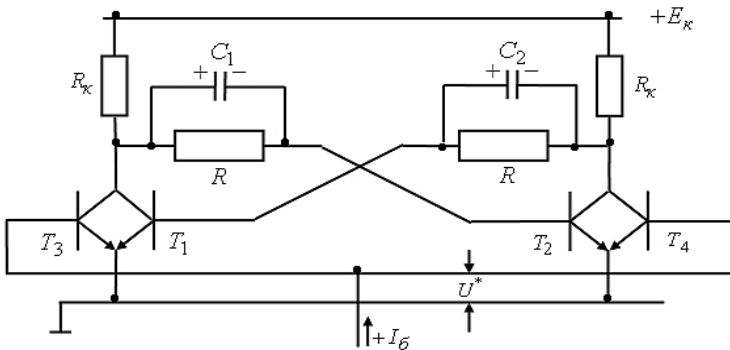


Рис. 4.7.

Принцип работы схемы.

Пусть в исходном состоянии T_1 заперт, а T_2 насыщен. Тогда $I_b = 0$, $U_{c_2} = U_{b_2} \cdot R = 0$, а C_1 заряжен и имеет полярность напряжения такую, как показано на рис. 4.7.

При поступлении короткого положительного управляющего сигнала T_3 и T_4 открываются до насыщения. Тогда потенциал коллектора первого транзистора стремится к нулю, и, пока действует входной импульс, C_1 разряжается через R , то есть медленно, и T_2 остается в открытом состоянии, открыт и T_1 . При прекращении

входного импульса C_1 оказывается приложенным к базе T_2 . Одновременно закрывается T_4 и, следовательно, закрывается T_2 , а, значит, T_1 остается в открытом состоянии. При подаче следующего импульса вновь оба транзистора входят в режим насыщения. При прекращении же второго импульса работает конденсатор C_2 и запирается T_1 . Таким образом, мы получили второе устойчивое состояние. Следовательно, поочередно открываются и запираются транзисторы T_1 и T_2 , а конденсаторы, запоминая (сохраняя какое-то время) напряжения, свойственные предыдущему устойчивому состоянию, обеспечивают однозначную искусственную асимметрию. Входной импульс нужен лишь для запуска процесса регенерации и по этой причине должен быть коротким:

- 1) $t_{вх} \ll \tau_c$;
- 2) $T > 3\tau_c$, где $\tau = RC$.

Второе условие ограничивает быстродействие схемы.

Такие **БЯ** назвали *T-триггерами*. *T*-триггер, как и *RS*-триггер, относится к простейшим триггерам.

Возвращаясь к выходному импульсу, можно отметить, что меняется потенциал от высокого уровня до низкого или наоборот. Отсюда название – *потенциальные* или *статические* триггеры.

Для удобства, чтобы не раскрывать внутренней схемы триггера, ввели условные обозначения (рис. 4.8). На правом рисунке выделено поле входов.

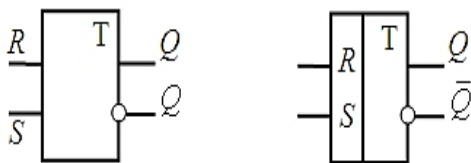


Рис. 4.8.

Потенциальные триггеры могут управляться потенциалом или динамически. В динамическом управлении это может быть фронт или срез. Обозначения динамического управления приведены на рис. 4.9. На рис. 4.9,а приведено обозначение управления фронтом, а на рис. 4.9,б – управления срезом.

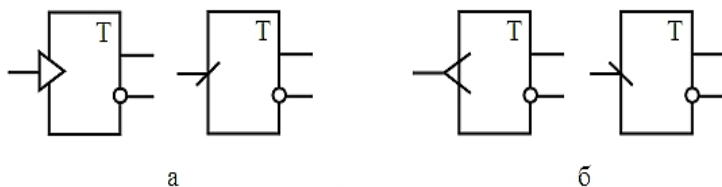


Рис. 4.9.

Чтобы получить триггер на логических элементах, необходимо организовать управление бистабильной ячейкой на логических элементах, как, например, это показано на рис. 4.10.

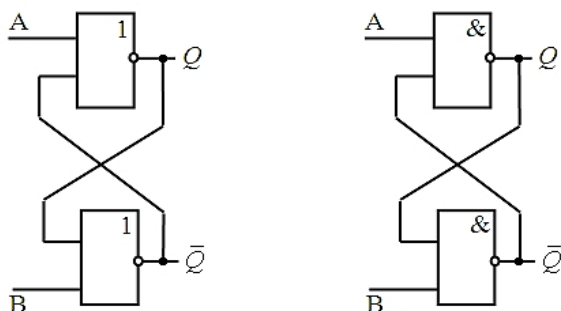


Рис. 4.10.

При этом получается какой-то триггер.

Возможность выполнения триггерных схем на основе стандартных логических элементов позволила разработчикам электронной аппаратуры процесс проектирования триггеров свести к разработке схем соединения логических элементов и к обеспечению их управления.

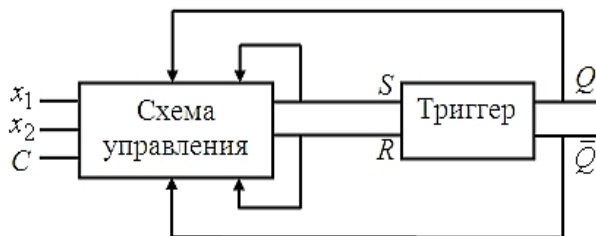


Рис. 4.11. Обобщенная схема триггерного устройства

Если схема управления строится как комбинационная, то мы имеем дело с простейшими триггерами. Если же схема управления содержит в себе простейшие триггеры, то это уже *триггерная система*.

Из рис. 4.11 видны обратные связи: вход триггера – схема управления; выход триггера – схема управления.

Триггеры по своему назначению называют также *конечными автоматами*, или однобитовыми элементами памяти.

Существуют классификации триггеров – по способу записи информации и по функциональному признаку.

По способу записи информации триггеры делятся на асинхронные (несинхронизируемые), и синхронные (синхронизируемые). В синхронных триггерах информация на выходе зависит не только от информационных сигналов (например, R и S), но и от сигнала разрешения или синхросигнала (C – от слова «clock» – часы, задающие время). Синхронный триггер обозначают, как показано на рис. 4.12.

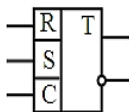


Рис. 4.12.

Синхронизация может быть как потенциальной, так и динамической.

По функциональному признаку разделение ведут по состоянию входов и выходов триггера в момент времени до срабатывания (в исходном состоянии) и после срабатывания. По этому признаку триггеры подразделяют на **RS**-, **D**-, **T**-, **TV**-, **DV**-, **E**-, **S**-, **R**-, **JK**-типа и т. д.

Многообразие типов обусловлено многообразием схем связи выхода триггера и схем управления.

В принципе, число триггеров зависит от количества входов и состояний выхода: $N = 5^{2^x}$, где 5 – число возможных состояний выхода (0, 1, Q, Q̄, X); x – число входов.

Значения выходов сигнала трактуются так:

«0» – Q = 0 независимо от сигнала на входе;

«1» – $Q = 1$ независимо от сигнала на входе;

Q – состояние не меняется с подачей сигналов на вход, т. е.

$Q = 1$ или $Q = 0$;

\bar{Q} – при изменении информации на входе состояние меняется на противоположное, т. е. 0 на 1, 1 на 0;

X – неопределенное состояние, т. е. $Q = \bar{Q} = 1$ или $Q = \bar{Q} = 0$ с равной вероятностью.

Если $x = 1$, то $N = 25$; если $x = 2$, то $N = 625$.

Однако число реально реализуемых триггеров значительно меньше. Так, при $x = 1$ триггеров всего 2 (**T-**, **D-**), а при $x = 2$ триггеров 24. Остальные либо тривиальны и из-за отсутствия процесса регенерации не имеют практического значения, либо дуальны (двойственные формы).

4.2. Синтез триггерных устройств

Составим таблицу переключений RS-триггера – табл. 4.2.

Таблица 4.2

S_n	R_n	Q_n	Q_{n+1}	\bar{Q}_{n+1}
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	x	x
1	1	1	x	x

Таблицу составляют, исходя из словесного описания.

1. Если на входах нет сигналов, т.е. $S_n = 0$ и $R_n = 0$, то состояние триггера не изменяется, т.е. если $Q_n = 0$, то $Q_{n+1} = 0$, а если $Q_n = 1$, то и $Q_{n+1} = 1$.

2. Сигнал $S_n = 1$ устанавливает на выходе сигнал $Q_n = 1$. Поэтому, если $S_n = 1$ и $R_n = 0$, то $Q_{n+1} = 1$ независимо от состояния $Q_n = 0$ до действия $S_n = 1$, т. е. если $Q_n = 0$, то $Q_{n+1} = 1$ и если $Q_n = 1$, то и $Q_{n+1} = 1$.

То есть уже из двух примеров мы можем обобщить схему и получить обобщённую (сокращённую) таблицу переключений (табл. 4.3).

Таблица 4.3

S_n	R_n	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	X

3. Сигнал $R_n = 1$ устанавливает на выходе Q уровень логического «0». Поэтому при $S_n = 0, R_n = 1, Q_{n+1} = 0$. Если в исходном состоянии $Q_n = 1$, то происходит переключение, а если $Q_n = 0$, то сохраняется исходное состояние.

4. Совокупность $S_n = 1$ и $R_n = 1$, т. е. управляющие сигналы сразу на обоих входах, то в этом случае состояние Q_{n+1} триггера считается неопределённым. Состояние неопределённости обозначается по-разному. Например, X или н/о. Конкретное состояние может быть определено, если известна схема триггера и его соединения. В отличие от комбинационных устройств термы, дающие неопределённые значения функции, не являются безразличными наборами.

Таблица переключения может быть представлена в виде карты Карно или диаграммы Вейча. В ячейки карты ставят значения Q_{n+1} .

Таким образом, для Q_{n+1} получаем следующую карту.

S_n, R_n	00	01	11	10
Q_n				
0	0	0	X	1
1	1	0	X	1

$$Q_{n+1} = \overline{\overline{S_n + Q_n R_n}} = \overline{\overline{S_n} \overline{Q_n} \overline{R_n}}$$

Для \bar{Q}_{n+1} карта представляет собой инверсию карты для Q_{n+1} .

S_n, R_n	00	01	11	10
Q_n				
0	1	1	X	0
1	0	1	X	0

$$\bar{Q}_{n+1} = \overline{R_n + Q_n \bar{S}_n} = \overline{R_n} \overline{Q_n} \overline{S}_n$$

В этих картах контуры склеивания проведены как пример для случая, когда X-наборы дают единичные значения функции.

4.3. RS-триггер с инверсными входами

Раскроем столбец неопределенности таблицы переключений и карту Карно двумя единицами. $S_n R_n Q_n Q_{n+1} \bar{Q}_{n+1}$

S_n	R_n	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	X
1	1	1	X

После минимизации и перевода в базис И-НЕ получим:

$$Q_{n+1} = \overline{S_n + Q_n \bar{R}} = \overline{S_n} \overline{Q_n} \overline{R}$$

Реализуем принципиальную электрическую схему (рис. 4.13).

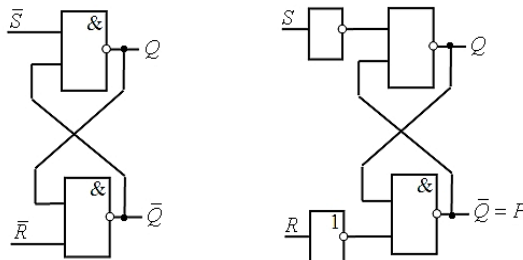


Рис. 4.13.

Учитывая, что устойчивость триггера получается из-за перекрёстных обратных связей, запишем сигналы обратной связи (уравнения возбуждения), но, чтобы не путать сигнал инверсного выхода с инверсией Q , переобозначим инверсный выход как P (рис. 4.13).

$$Q = \overline{\overline{S} \cdot P} = S + \overline{P},$$

$$P = \overline{\overline{R} \cdot Q} = R + \overline{Q}.$$

Составим так называемую обобщенную карту Карно, в которой по строкам меняются QP , а по столбцам SR . В клетки карты внесём первой цифрой значения Q , получающиеся по уравнению возбуждения для Q , а второй цифрой значения P по уравнению возбуждения для P .

Следующим шагом сравним исходные QP карты и значения в клетках. Совпадающие цифры заключим в квадратные скобки. Совпадение означает, что если возьмем какое-то исходное состояние и подадим на вход триггера соответствующий сигнал, то исходное состояние сохранится, будет устойчивым.

$SR \backslash QP$	00	01	11	10
00	11	11	11	11
01	[01]	[01]	11	11
11	00	01	[11]	10
10	[10]	11	11	[10]

Например, при исходном $QP=01$ при подаче на вход $SR=00$ или $SR=01$ состояние триггера оказывается устойчивым. Одна скобка в этой же строке говорит о том, что не меняется значение только одного выхода. Значения столбца $SR=00$ показывают, что триггер при отсутствии сигналов на входах (подаче двух нулей) может иметь два исходных состояния $QP=01$ или $QP=10$. Подачей сигналов $S=0, R=1$ триггер устанавливается в одно устойчивое состояние $Q=0, P=1$ (установка в ноль). Сигналами $S=1, R=0$ триггер устанавливается только в одно устойчивое состояние $Q=1, P=0$ (установка в единицу). Все эти результаты соответству-

ют словесной формулировке триггера. При исходном состоянии $QP=11$ и подаче на вход сигналов $SR=11$ триггер устанавливается в устойчивое состояние $QP=11$, что не соответствует словесной формулировке триггера о противофазности его выходов.

Таким образом, получаем

$S = 0, R = 0$ [01] или [10] – хранение информации

$S = 1, R = 0$ [10] – установка в «1»

$S = 0, R = 1$ [01] – установка в «0»

$S = 1, R = 1$ [11] – «запрет» (подача двух единиц на вход триггера с инверсными входами запрещается).

Математически условие запрета $SR=0$.

4.4. Синтез JK-триггера

Синтез триггерных устройств – задача сложная. Сложность объясняется тем, что реализация триггеров таким образом, как мы проделали на **RS**-триггере, не учитывает состязаний сигналов, т. е. может привести последовательное устройство к нестабильному состоянию.

Во-первых, многообразие триггерных устройств исключает возможность простого подхода, так как обычно триггерное устройство состоит не из одной ячейки памяти, а во-вторых, возможно динамическое управление.

Рассмотрим вопрос на примере **JK**-триггера. **JK**-триггер – универсальный триггер, который может работать в режиме раздельных входов как **RS**-триггер и в счетном режиме как **T**-триггер, где **J** и **K** – входные сигналы.

№ набора	J	K	Q_n	Q_{n+1}
0	0	0	0	(0)
1	0	0	1	(1)
2	0	1	0	(0)
3	0	1	1	0↑
4	1	0	0	1↓
5	1	0	1	(1)
6	1	1	0	1
7	1	1	1	0

Столбец неопределенности для $J = K = 1$ раскрыт как $1/0$.

Выделим устойчивые состояния скобками, т. е. состояния, когда $Q_n = Q_{n+1}$. Неустойчивые состояния разрешим в устойчивые стрелкой. То есть если в паре одинаковых входных сигналов есть одно устойчивое и одно неустойчивое, но такое же по значению состояние, то неустойчивое переходит в устойчивое. Например, для 2 и 3 неустойчивый «0» переходит в устойчивый «0».

Состояния 6 и 7 – неустойчивые, соответствуют колебательному режиму. Поэтому *построение JK-триггера на одной ячейке памяти* или же на одном **RS**-триггере без специальных мер *дает кольцевой генератор*. Для получения из **JK**-триггера счетного триггера необходимо ограничивать время действия входных сигналов, чтобы оно было меньше времени процесса регенерации (или переходного процесса возбуждения).

Из таблиц переключений можно получить сокращенную таблицу

№	J	K	Q_{n+1}
1	0	0	Q_n
2	0	1	0
3	1	0	1
4	1	1	\bar{Q}_n

Введем понятия запираания и отпираания триггера. Будем считать, что переход триггера из состояния «1» в состояние «0» – запираание, а из «0» в «1» – отпираание. Выделим на основании таблицы наборы: отпирающие – 4 и 6 и запирающие – 3 и 7.

Отпираанию можно поставить в соответствие некоторый сигнал:

$$S = JK\bar{Q}_n + JK\bar{Q}_n = J\bar{Q}_n = \overline{\overline{JQ}_n}.$$

Запираанию можно поставить в соответствие некоторый сигнал:

$$R = \bar{J}KQ_n + JKQ_n = KQ_n = \overline{\overline{KQ}_n},$$

то есть мы имеем сигналы управления. Выберем для реализации **JK**-триггера в качестве элементарной ячейки памяти **RS**-триггер с инверсными входами и реализуем сигнал управления S и R (рис. 4.14).

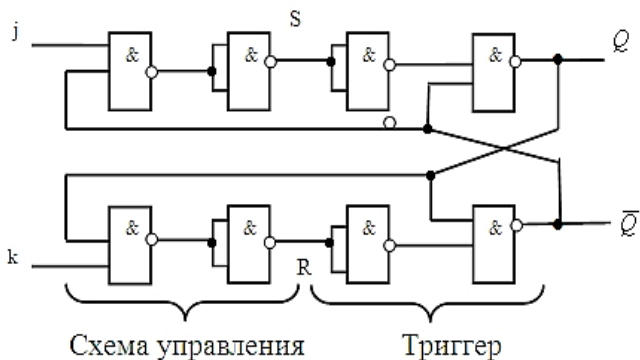


Рис. 4.14.

Так как 2 инвертора не меняют функций, то итоговая схема выглядит, как показано на рис. 4.15:

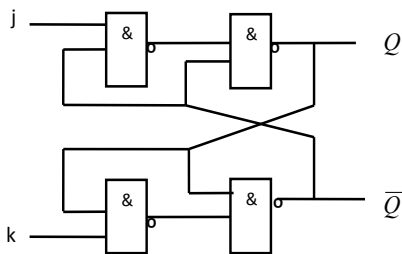


Рис. 4.15.

Можно сделать следующие выводы:

- В данном примере мы сделали попытку реализации триггера по схеме управления – ячейка памяти с реализацией обратных связей: выход – вход ячейки, выход – вход схемы управления. **RS**- и **D**-триггеры не имеют обратных связей типа ячейка – схема управления, а **JK**- и **T**-триггеры имеют;

- Несмотря на наличие взаимных перекрестных связей, такая реализация **JK**-триггера далека от оптимальной с точки зрения устойчивости триггера и адекватности его работы по таблице переключений. Она справедлива для управления потенциалом;

- Понятно, что входные сигналы **J** и **K** есть внутренние **R** и **S**.

4.5. D-триггер

D-триггер имеет один вход, обычно обозначаемый буквой **D**. Рассмотрим таблицу переключений:

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

Характеристическое уравнение триггера может быть получено из таблицы истинности путем сложения сочетаний $D_n Q_n$, дающих единичные значения Q_{n+1} :

$$Q_{n+1} = \underbrace{D_n \bar{Q}_n + D_n Q_n}_{\text{СДНФ}} = D_n$$

Из данного выражения следует, что реализация «в лоб» дает тривиальную схему (рис. 4.16), т. е. нет регенерации, а есть просто задержка сигнала.

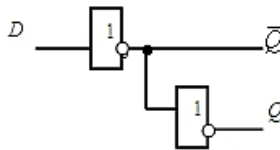


Рис. 4.16.

Но если строить триггер по схеме ячейка памяти – схема управления, то $S^* = D$, $R^* = \bar{D}$.

Возьмем за основу **RS**-триггер с прямыми входами (базис ИЛИ-НЕ) (рис. 4.17).

Таким образом, значение сигнала на выходе **D** в момент времени t_{n+1} всегда равно значению сигнала t_n . Из таблицы функционирования **D**-триггера видно, что включающий сигнал (3-я строка) $S^* = D$; и выключающий сигнал (2-я строка) $R^* = \bar{D}$.

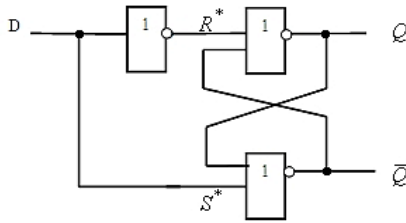
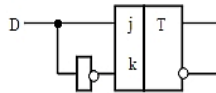


Рис. 4.17.

Получить **D**-триггер можно, используя **JK**-триггер. В этом случае таблице состояний **D**-триггера будут соответствовать строки 2-5 таблицы **JK**-триггера, т. е. $J=D=\bar{k}$. В этом случае $S = D = \bar{k}$.



Кроме обычного **D**-триггера, существует также запирающий **D**-триггер. Он получается из синхронного **RS**-триггера с инверсными входами. Этот **RS**-триггер при $C=1$ реагирует на все сочетания сигналов **R** и **S**, кроме запрещенных ($R=S=1$). Если $C=0$, то триггер не реагирует на входные сигналы. Он запирается сигналом $C=0$. Превратим его в **D**-триггер. Работу можно проследить на временной диаграмме (рис. 4.18), из которой видно, что триггер фиксирует на выходе состояние, которое было на входе **D** до переключения **C**. Поэтому он называется также «защелкой», фиксатором.

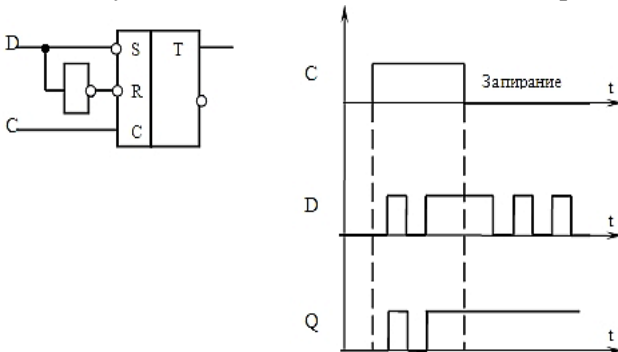


Рис. 4.18.

До появления сигнала $C=1$ он задерживает поступающий по входу D сигнал. Откуда и пошло название D -триггера (delay – задержка). Другое назначение D -триггера – сохранить данные (data), поступившие однажды по входу D . В этом случае сигнал C означает команду «записать в триггер».

Таким образом, D -триггер гораздо лучше подходит для записи и хранения информации, чем RS -триггер, требующий двух последовательных сигналов гашения (R) и записи (S).

Если установить на входе C постоянный уровень 1, то свойство защелки теряется. Она /защелка/ становится прозрачной и пропускает информацию. В этом смысле триггер-защелку использует как буферный усилитель мощности в линии передачи данных. Примеры триггеров-защелок: K155TM5, K155TM7, K561TM3. Они содержат 4 триггера с входом C . Иногда в CD -триггер вводят еще один вход – вход установки в ноль (R).

Существуют D -триггеры, в которых параллельно синхронизации вводят вход разрешения. Этот вход называют V -входом (от «valve» – клапан) и получают DV -триггер. Подробно этот триггер рассматривается в лабораторном практикуме.

4.6. T-триггер

T -триггер – счетный триггер. Для него имеем таблицу переключений:

N	T	Q_n	Q_{n+1}
0	0	0	(0)
1	0	1	(1)
2	1	0	1
3	1	1	0

Карта Карно выглядит так:

$Q_n \backslash T$	\bar{T}	T
\bar{Q}_n	0	1
Q_n	1	0

Из нее можно записать следующее уравнение:

$$Q_{n+1} = Q_n \bar{T} + \bar{Q}_n T$$

Из таблицы следует неустойчивость триггера. Пусть $Q_n = 0$. Подаем $T=1$, $Q_{n+1} = 1$, и если еще держим $T=1$, то получаем $Q_{n+1} = 0$. Таким образом мы получили колебательный режим.

T-триггер можно получить из синхронного **RS**-триггера (рис. 4.19,а) или **D**-триггера (рис. 4.19,б).

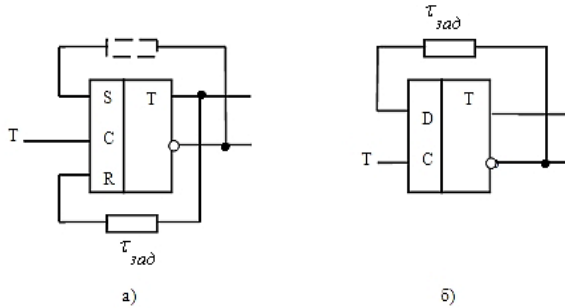


Рис. 4.19.

Линия задержки $\tau_{зad}$ устраняет колебательный режим. Однако 100 % устойчивости получить нельзя, приходится следить за длительностью входного сигнала.

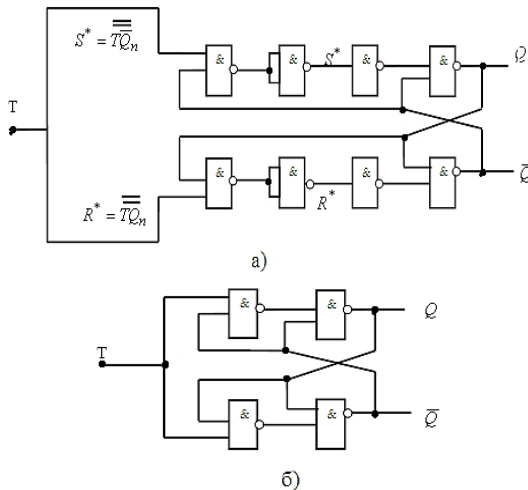


Рис. 4.20.

На рис. 4.20, а показано построение Т-триггера по схеме ячейка памяти – схема управления. Если исключить двойное инвертирование, то получим триггер, показанный на рис. 4.20, б.

4.7. Синхронные триггеры

Обычно триггер работает в условиях системы. Возможны гонки сигналов. Поэтому предусматривается схема синхронизации работы триггера или строятся синхронные триггеры. Схема синхронного RS-триггера с инверсными входами показана на рис. 4.21.

Принцип синхронизации очевиден. На входе стоят элементы И-НЕ, т. е. для операции И активным является нулевой сигнал. Если такой ноль С подать сразу на ячейки R и S, то это равносильно: $S=R=0$, т. е. условию хранения информации. Если сигнал $C=1$, то сигнал становится неактивным и все определяют R и S (по аксиоме $1 \cdot X = X$).

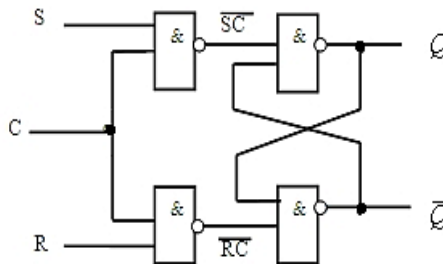


Рис. 4.21.

Аналитически можно написать:

без синхронизации $Q_{n+1} = S_n + \bar{R}_n Q_n = \bar{S}_n \cdot \overline{\bar{R}_n Q_n}$,
с синхронизацией .

$$Q_{n+1} = \overline{\overline{S_n C} \cdot \overline{Q_n R_n} \cdot C}$$

Т. е. входные сигналы S и R просто подали на схему И с C и проинвертировали.

Если триггер представляет собой триггерную систему, то вопрос с синхронизацией осложняется. Во-первых, сколько ячеек

синхронизировать – одну или все? Во-вторых, синхронизация бывает динамической с управлением фронтом или срезом.

Если синтезировать синхронный триггер, то в таблицу функционирования С вводится как самая старшая переменная. Ясно, что в этом случае потребуется карта Карно 4 переменных. Реализацию можно посмотреть в справочниках по ЦИС. Например, микросхемы ТТЛ: ТВ6; ТВ9; ТВ1; ТВ10.

Выводы

1. С точки зрения обобщенной схемы триггеров триггеры **RS** и **D** не имеют обратных связей типа триггер – схема управления, а триггеры **JK** и **T** используют выходные сигналы для воздействия на свои входы.

2. Из таблиц переходов триггеров **JK** и **T** следует колебательный режим. Для его устранения можно наложить ограничения на длительность входного сигнала. Но можно пойти другим путем – построить *двухступенчатый триггер*. Двухступенчатый триггер делает триггер непрозрачным, т. е. устраняется колебательный режим. Схема двухступенчатого триггера показана на рис. 4.22.

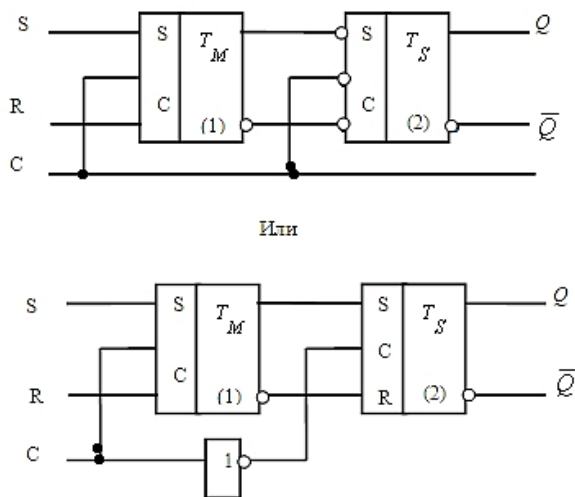


Рис. 4.22.

При $C=1$ разрешается срабатывание первого триггера (1) и запрещается работа второго (2). При $C=0$: информация передается последовательно из (1) во (2). Вторая схема может дать ситуацию риска, так как входной триггер T_M состязается с инвертором. Процесс регенерации в триггере T_M опережает срабатывание инвертора.

3. Асинхронные входы предустановки в «0» или в «1» являются доминирующими перед информационными входами.

4. Двухступенчатые триггеры называют **MS**-триггерами (**M** – master – хозяин, **S** – slave – слуга [(1) – мастер, (2) – слуга]).

4.8. Триггерные схемы со сложной входной логикой

Возьмем схему RS-триггера с инверсными входами

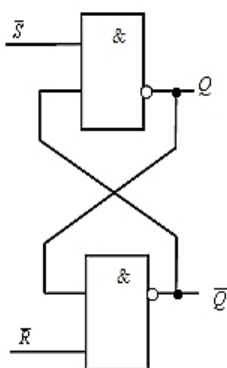


Рис.4.23.

Если $\bar{S} = \bar{R} = 1$, т. е. $S=R=1$, то мы имеем элемент памяти – хранение информации 10 или 01. Если $\bar{S} = 0$, $\bar{R} = 1$ мы имеем $S = 1$, $R = 0$ и получим 1 на выходе Q .

Теперь рассмотрим многовходовую схему.

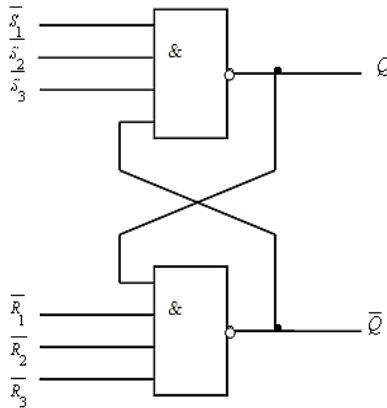


Рис. 4.24.

Чтобы схема находилась в режиме хранения, как в предыдущем случае, необходимо, чтобы все входные переменные $\overline{S_1}, \overline{S_2}, \overline{S_3}$ и $\overline{R_1}, \overline{R_2}, \overline{R_3}$ одновременно были равны 1. Для установки триггера в состояние $Q=1$ достаточно подать нулевой сигнал, по крайней мере, хотя бы на один из входов $\overline{S_i}$, а для установки в ноль подать ноль хотя бы на один из входов $\overline{R_i}$.

При этом функции установки в 1 и 0 можно записать в виде: $S = S_1 + S_2 + S_3$, $R = R_1 + R_2 + R_3$ из $\overline{S} = \overline{S_1} \cdot \overline{S_2} \cdot \overline{S_3} = S = S_1 + S_2 + S_3$.

Применение этого принципа позволяет проводить предварительную *очистку* или *установку* триггера или всех его ячеек памяти в нужное состояние перед подачей сигналов на информационные входы.

Мы рассмотрели вопрос на примере RS-триггера с инверсными входами, но важен принцип *расширения входов логических элементов ячейки памяти*.

4.9. Вопросы для самопроверки

1. Как классифицируются триггерные устройства по функциональному признаку?

2. Объясните назначение R-, S-, D-, C-, T-входов триггеров. Что такое активный уровень? Почему RS-триггер на элементах И-НЕ называют триггером с инверсными входами?

3. Чем отличаются синхронные и асинхронные триггеры? Зачем в последовательностных схемах необходимо тактирование?
4. Приведите схемы RS -триггеров в логических базисах И-НЕ и ИЛИ-НЕ.
5. Какие требования предъявляются к длительности тактовых импульсов RS -триггера?
6. Укажите отличительную особенность JK -триггера.
7. Почему простейшие JK -триггеры являются функционально ненадежными?
8. Запишите таблицу переключений D -триггера и его характеристическое уравнение.
9. Чем определяется задержка в тактируемых и асинхронных D -триггерах?
10. Объясните принцип работы и назначение триггеров различных типов (RS , T , JK , D).
11. Покажите, в чем суть методов минимизации характеристических уравнений.
12. Приведите схемы D -триггеров на элементах И-НЕ, ИЛИ-НЕ.

4.10. Задачи для самостоятельного решения

1. Синтезируйте триггер в базисе ИЛИ-НЕ. Для этого замените столбец неопределенности двумя нолями. С помощью обобщенной карты Карно проверьте триггер на разрешенные и запрещенные сигналы. Организуйте синхронизацию.
2. Синтезируйте DV -триггер. Вход V – вход разрешения – доминирующий. Базис И-НЕ.
3. Синтезируйте E -триггер, для чего раскройте столбец неопределенности как $0/1$. Базис И-НЕ. Перед переводом в универсальный базис функцию надо преобразовать.
4. По схеме ячейка памяти – схема управления синтезируйте R -триггер, у которого при подаче на вход $R=1$ и $S=1$ предпочтение отдается сигналу R .
5. По схеме ячейка памяти – схема управления синтезируйте S -триггер, у которого при подаче на вход $R=1$ и $S=1$ предпочтение отдается сигналу S .

6. Нарисуйте схемы преобразования JK-триггера в D- и T-триггеры.

7. Можно ли построить D- и T-триггеры на базе RS-триггеров?

8. Постройте идеализированную временную диаграмму тактируемого уровнем RS-триггера на элементах И-НЕ.

9. Постройте идеализированную временную диаграмму тактируемого уровнем DV-триггера на элементах И-НЕ.

4.11. Лабораторная работа № 4

4.11.1. Выполнение работы на универсальном стенде

Лабораторная работа выполняется на «Универсальном стенде по основам информатики и электронно-вычислительной техники».

Триггеры, выполненные на элементах «И-НЕ», «ИЛИ-НЕ», а также в интегральном исполнении на ИМС K155TM2, составляют плату II (римская цифра два). Перед выполнением работы закрепить плату II на разъеме основного блока. На технологических картах II-1, II-2, II-3, II-4, II-7 можно выполнить следующие задания.

1. **Карта II-1.** Позволяет исследовать RS-триггер с прямыми входами (триггер на элементах ИЛИ-НЕ). Активный уровень для данной схемы – логическая единица. Управляющие R и S входы X_1, X_2 , что соответствует кнопкам SB_2 и SB_3 на макете. Индикаторы управляющих сигналов HL_1, HL_2 . Если один из индикаторов горит, значит, на данном входе присутствует логическая единица. Главный и инверсный выходы Y_1, Y_2 . Индикаторы выходных сигналов HL_3, HL_4 . Составьте для данного триггера таблицу переключений. По данной таблице переключений нарисуйте идеализированную временную диаграмму RS-триггера с прямыми входами.

2. **Карта II-2.** Исследуется RS-триггер с инверсными входами (триггер на элементах И-НЕ). Активный уровень для этой схемы – логический ноль. Управляющие сигналы передаются с кнопок SB_2, SB_3 на входы триггера X_1, X_2 . Уровни входных сигналов индицируются в инверсном коде на индикаторах HL_1, HL_2 , т. е. когда на входе триггера логический ноль, то светодиод, соответствующий этому входу, горит. Выходные сигналы индицируются светодиодами HL_5, HL_6 в прямом коде. Для данного триггера сделайте то же задание, что и для RS-триггера с прямыми входами.

3. **Карта П-3.** Исследуется *RS*-триггер с инверсными входами в интегральном исполнении на ИМС типа К155ТМ2. Управляющие сигналы те же, что и в предыдущем случае. Выходные сигналы индицируются светодиодами HL_9 (прямой выход) и HL_8 (инверсный выход). Постройте таблицу переключений и покажите идентичность работы триггера на элементах И-НЕ и К155ТМ2, для которого «И-НЕ» – базовый логический элемент.

4. **Карта П-4.** Исследуется тактируемый *D*-триггер. Уровень сигнала на управляющем входе *D*-триггера определяется по положению тумблера SA_1 на лабораторном стенде. Верхнее положение тумблера соответствует логической единице, а нижнее – логическому нулю. Выходные уровни индицируются светодиодами HL_9 (прямой выход) и HL_8 (инверсный выход). Составьте таблицу истинности и таблицу переключений. По полученной таблице переключений составить временную диаграмму напряжений на главном выходе Q по сигналам на входах *D* и *C*. Подключите осциллограф к макету, как показано на карте П-4. На осциллограмме наблюдайте синхроимпульс. Покажите, что запись информации в *D*-триггер происходит по фронту синхроимпульса.

5. **Карта П-7.** Позволяет исследовать счетный *T*-триггер. Поставьте перемычку между контактами Y_1 и X_2 платы П2. Активный уровень в данной схеме – логическая единица. Индикация выходного сигнала главного выхода осуществляется индикатором HL_7 . Последовательно подавая на вход *T*-триггера импульсы, составить таблицу переключений аналогично предыдущим заданиям.

4.11.2. Выполнение работы на ПЭВМ

Лабораторный практикум выполняется во время самостоятельной работы по предмету. Схемы триггерных устройств заданы в базе системы компьютерного моделирования EWB в папке «lib». Методические указания по выполнению компьютерных лабораторных работ в электронном виде получить у преподавателя.

СПИСОК ЛИТЕРАТУРЫ

1. Опадчий, Ю. Ф. Аналоговая и цифровая электроника / Ю. Ф. Опадчий, О. П. Глудкин, А. И. Гуров. – М.: Горячая Линия-Телеком, 2000. – 768 с.
2. Угрюмов, Е. П. Цифровая схемотехника / Е. П. Угрюмов. – СПб., 2000. – 528 с.
3. Алексенко, А. Г. Основы микросхемотехники / А. Г. Алексенко. – М.: ЮНИМЕДИАСТАЙЛ, 2002. – 448 с.
4. Бабич, Н. П. Основы цифровой схемотехники: учебное пособие / Н. П. Бабич, И. А. Жуков. – М.: Додэка-XXI; К.: МК-Пресс, 2007. – 480 с.
5. Браммер, Ю. А. Цифровые устройства: учебное пособие для вузов / Ю. А. Браммер, И. Н. Пашук. – М.: Высшая Школа, 2004. – 229 с.
6. Фрике, К. Вводный курс цифровой электроники / К. Фрике. – М.: Техносфера, 2003. – 432 с.
7. Кардашев, Г. А. Цифровая электроника на персональном компьютере. Electronics Work bench и Mikro-Cap / Г. А. Кардашев. – М.: Горячая линия-Телеком, 2003. – 311 с.
8. Шило, В. Л. Популярныe цифровые микросхемы: справочник / В. Л. Шило. – М.: Металлургия, 1988 и др. года издания.

Оглавление

Предисловие.....	3
Глава I. ОСНОВЫ ТЕОРИИ ЛОГИЧЕСКИХ УСТРОЙСТВ.....	4
1.1. Простейшие логические операции.....	4
1.2. Двойственность алгебры Буля.....	7
1.3. Аксиомы и законы алгебры.....	8
1.4. Логические функции.....	11
1.5. Построение логической схемы.....	13
1.6. Универсальные логические элементы.....	14
1.7. Функции двух переменных.....	18
1.8. Карта Карно.....	20
1.9. Минимизация логических функций.....	22
1.10. Ситуация статического риска.....	25
1.11. Вопросы для самопроверки.....	27
1.12. Задачи для самостоятельного решения.....	28
1.13. Лабораторная работа № 1 «Синтез и анализ логической функции».....	28
Глава II. КОМБИНАЦИОННЫЕ УСТРОЙСТВА СРАВНЕНИЯ, ДЕШИФРАЦИИ И ПРЕОБРАЗОВАНИЯ КОДОВ.....	30
2.1. Компаратор.....	30
2.2. Мультиплексоры.....	36
2.3. Двоичное кодирование (коды).....	44
2.4. Лабораторная работа № 2.....	54
Глава III. ДВОИЧНЫЕ СУММАТОРЫ.....	55
3.1. Одноразрядные двоичные сумматоры.....	55
3.2. Многоразрядные сумматоры.....	62

3.3. Вопросы для самопроверки.....	66
3.4. Лабораторная работа № 3.....	67
Глава IV. ТРИГГЕРЫ.....	70
4.1. Общие положения.....	70
4.2. Синтез триггерных устройств.....	79
4.3. RS-триггер с инверсными входами.....	81
4.4. Синтез JK-триггера.....	83
4.5. D-триггер.....	86
4.6. T-триггер.....	88
4.7. Синхронные триггеры.....	90
4.8. Триггерные схемы со сложной входной логикой.....	92
4.9. Вопросы для самопроверки.....	93
4.10. Задачи для самостоятельного решения.....	94
4.11. Лабораторная работа № 4.....	95
Список литературы.....	97

Учебное издание

Артемов Константин Серафимович
Солдатова Нина Львовна

ОСНОВЫ **ЦИФРОВОЙ ЭЛЕКТРОНИКИ**

Учебное пособие

Редактор, корректор М. В. Никулина
Правка, верстка Е. Б. Половкова

Подписано в печать 05.12.2013. Формат 60×84^{1/16}.

Усл. печ. л. 5,81. Уч.-изд. л. 5,24.

Тираж 100 экз. Заказ .

Оригинал-макет подготовлен
в редакционно-издательском отделе ЯрГУ.

Ярославский государственный университет
им. П. Г. Демидова.
150000, Ярославль, ул. Советская, 14.