

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное  
учреждение высшего образования  
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ

---

## ЦИФРОВЫЕ УСТРОЙСТВА

Методические указания  
к выполнению лабораторных работ

Составители: Ю. В. Бакшеева, К. К. Томчук

Рецензент – кандидат технических наук, доцент *Н. В. Поваренкин*

Содержат краткие теоретические сведения и методические указания к выполнению лабораторных работ, предназначенных для получения базовых знаний по цифровой схемотехнике. Предназначено для студентов направлений 11.03.01 «Радиотехника», 11.03.02 «Инфокоммуникационные технологии и системы связи», 12.03.04 «Биотехнические системы и технологии» и специальности «Радиоэлектронные системы и комплексы» 11.05.01 и др., изучающих дисциплины: «Цифровые устройства», «Цифровые устройства и микропроцессоры», «Схемотехника», «Схемотехника ЭС. Цифровая схемотехника» и др.

Публикуется в авторской редакции.

Компьютерная верстка *И. Н. Мороз*

---

Сдано в набор 17.10.19. Подписано к печати 16.12.19.

Формат 60×84 1/16. Усл. печ. л. 2,73

Уч.-изд. л. 2,93. Тираж 50 экз. Заказ № 505.

---

Редакционно-издательский центр ГУАП  
190000, Санкт-Петербург, Б. Морская ул., 67

© Санкт-Петербургский государственный  
университет аэрокосмического  
приборостроения, 2019

## **ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ**

ДНФ – дизъюнктивная нормальная форма.

ИМС – интегральная микросхема.

КМОП – комплементарный металл-оксид-полупроводник.

КНФ – конъюнктивная нормальная форма.

КУ – комбинационное устройство.

ЛФ – логическая функция.

ЛЭ – логический элемент.

СДНФ – совершенная дизъюнктивная нормальная форма.

СКНФ – совершенная конъюнктивная нормальная форма.

ТИ – таблица истинности.

ТТЛ – транзисторно-транзисторная логика.

ЭСЛ – эмиттерно-связанная логика.

# 1. ЛАБОРАТОРНАЯ РАБОТА № 1

## «ИССЛЕДОВАНИЕ ЛОГИЧЕСКИХ ЭЛЕМЕНТОВ»

*Цель работы:* знакомство с элементарными логическими функциями, освоение пакета моделирования *Electronic Workbench MultiSim*, получение первичных навыков по работе с виртуальными приборами, исследование параметров логических элементов.

### 1.1. Аксиомы булевой алгебры. Понятие логического элемента

Основы *алгебры логики* были заложены в середине XIX века английским математиком Джорджем Булем, по имени которого алгебра логики называется также *булевой алгеброй*.

В алгебре логики рассматриваются переменные, которые могут принимать только два значения – 0 и 1.

В алгебре логики определены:

- отношение эквивалентности ( $=$ );

- логическая операция «отрицание» (инверсия, операция НЕ) – унарная операция (операция над одним операндом), обозначается горизонтальной чертой над переменной или значением, например,  $\bar{x}$  или  $\bar{0}$ ;

- логическая операция «конъюнкция» (операция И, логическое умножение), обозначается знаками  $\&$ ,  $\wedge$  или точкой (умножением), которую допускается опускать, например,  $x \wedge y$ ,  $x \& y$ ,  $x \cdot y$  или  $xy$ ;

- логическая операция «дизъюнкция» (операция ИЛИ, логическое сложение), чаще всего обозначается знаком  $\vee$ , например,  $x \vee y$ .

Отношение эквивалентности удовлетворяет следующим свойствам:

- рефлексивность –  $x = x$ ;

- симметричность – если  $x = y$ , то  $y = x$ ;

- транзитивность – если  $x = y$  и  $y = z$ , то  $x = z$ .

Алгебра логики оперирует системой аксиом, которые, по сути, определяют логические операции:

$$\begin{cases} x = 0, & \text{если } x \neq 1 \\ x = 1, & \text{если } x \neq 0, \end{cases} \quad (1.1)$$

$$\begin{cases} 1 \vee 1 = 1 \\ 0 \vee 0 = 0 \\ 1 \vee 0 = 0 \vee 1 = 1, \end{cases} \quad (1.2)$$

$$\begin{cases} 1 \wedge 1 = 1 \\ 0 \wedge 0 = 0 \\ 1 \wedge 0 = 0 \wedge 1 = 0, \end{cases} \quad (1.3)$$

$$\begin{cases} \bar{1} = 0 \\ \bar{0} = 1. \end{cases} \quad (1.4)$$

Аксиома (1.1) является утверждением того, что в булевой алгебре рассматриваются только двоичные переменные. Аксиомы (1.2) и (1.3) определяют операции дизъюнкции и конъюнкции, а аксиома (1.4) – операцию инверсии.

Любое логическое выражение, составленное из некоторого количества логических переменных с помощью конечного числа операций алгебры логики, называют *логической функцией*.

Логические операции {И, ИЛИ, НЕ} составляют так называемый *полный базис* – такое множество логических функций, что любую другую логическую функцию можно представить как композицию функций данного множества.

В цифровой схемотехнике широко применяют аппарат алгебры логики для анализа и синтеза цифровых схем.

Элементом цифровой схемотехники называют устройство, которое выполняет одну из задач реализации булевых функций. Все виды элементов цифровых устройств можно разделить на три группы:

- логические,
- функциональные,
- вспомогательные.

*Логическим элементом* (ЛЭ) называют устройство, которое реализует только одну булеву функцию. Такие элементы называют *однофункциональными*. Логический элемент, реализующий операцию инверсии, называют *инвертор* или *элемент НЕ*. Логический элемент, реализующий операцию дизъюнкции, называют *элемент ИЛИ* (иногда в названии уточняют количество операндов, например, 2ИЛИ – логический элемент с двумя входами, реализующий операцию  $x \vee y$ ; 3ИЛИ – логический элемент с тремя входами, реализующий операцию  $x \vee y \vee z$ ). Логический элемент, реализующий операцию конъюнкции, называют *элемент И* (иногда в названии уточняют количество операндов, например, 2И – логический элемент с двумя входами, реализующий операцию  $x \wedge y$ ; 3И – логический элемент с тремя входами, реализующий операцию  $x \wedge y \wedge z$ ).

Функциональные элементы реализуют несколько булевых функций и могут состоять из нескольких логических элементов. Их состав, как правило, отражается в их названии, например, 2ИЛИ-НЕ – функциональный элемент с двумя входами, реализующий логическую функцию  $\overline{x} \vee \overline{y}$ .

Одним из способов описания элементарных устройств цифровой схемотехники является *таблица истинности* (ТИ), которая представляет собой функциональную зависимость между входными и выходными сигналами, выраженную в табличной форме. Входная часть ТИ для устройства с  $n$  входами содержит  $n$  колонок и  $N=2^n$  строк, содержащих все возможные комбинации входных сигналов. Выходная часть заполняется для каждого выхода отдельно и описывает реакцию данного выхода устройства на каждую комбинацию входных сигналов. В таблицах 1.1–1.3 представлены таблицы истинности для элементарных логических операций. В таблицах входные переменные обозначены  $x_1$  и  $x_2$ , а выходная переменная –  $y$ .

Таблица 1.1

Таблица истинности логического элемента 2И

$x_1$	$x_2$	$y$
0	0	0
0	1	0
1	0	0
1	1	1

Таблица 1.2

Таблица истинности логического элемента 2ИЛИ

$x_1$	$x_2$	$y$
0	0	0
0	1	1
1	0	1
1	1	1

Таблица истинности логического элемента НЕ

$x1$	$y$
0	1
1	0

## 1.2. Основные характеристики логических элементов

*Передаточная характеристика* (рис. 1.1) – основная статическая характеристика ЛЭ – зависимость выходного напряжения от напряжения на одном из входов при постоянных напряжениях на остальных входах, равных напряжению пассивного логического сигнала для данного ЛЭ. Передачная характеристика содержит информацию об уровнях номинальных и пороговых входных и выходных напряжений, соответствующих логическим сигналам «0» и «1». Также используется для анализа *помехоустойчивости* ЛЭ.

*Помехоустойчивость* ЛЭ – максимально допустимые постоянные напряжения помехи положительной полярности (для напряжения логического «0» на входе) и отрицательной полярности (при напряжении логической «1» на входе).

*Входная характеристика* – зависимость входного тока от напряжения на данном входе при постоянных напряжениях на остальных входах.

*Выходная характеристика* – зависимость выходного напряжения от выходного тока при заданных постоянных напряжениях на входах.

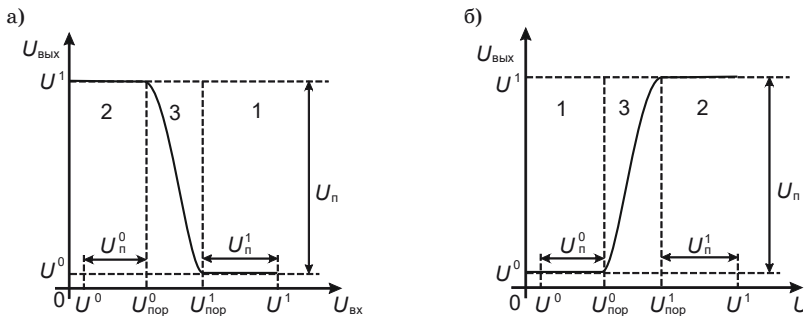


Рис. 1.1. Передачная характеристика инвертирующего (а) и неинвертирующего (б) ЛЭ

*Нагрузочная способность  $n$*  (коэффициент разветвления по выходу) – максимальное число ЛЭ, аналогичных рассматриваемому, которые одновременно можно подключать к его выходу.

*Коэффициент объединения по входу  $m$*  – число входов ЛЭ. С увеличением коэффициента  $m$  расширяются функциональные возможности ЛЭ за счет выполнения функций над большим числом логических переменных. При этом для создания сложного устройства требуется меньше ЛЭ. Однако увеличение числа входов, как правило, ухудшает другие основные параметры ЛЭ, прежде всего, быстроедействие.

*Потребляемая мощность ЛЭ* – зависит от его логического состояния, так как изменяется ток источника питания в цепи питания. Средняя потребляемая мощность в статическом режиме  $P_{\text{ср}}$ :

$$P_{\text{ср}} = 0,5U_{\text{и.п.}} \left( I_{\text{и.п.}}^0 + I_{\text{и.п.}}^1 \right), \quad (1.5)$$

где  $U_{\text{и.п.}}$  – напряжение источника питания,  $I_{\text{и.п.}}^0$  и  $I_{\text{и.п.}}^1$  – потребляемый от источника питания ток при выходных напряжениях низкого («0») и высокого («1») логических уровней соответственно.

*Быстродействие ЛЭ* оценивают *средним временем задержки распространения сигнала*  $t_{\text{зд.п.ср.}}$  от входа к выходу, определяющим среднее время выполнения логической операции:

$$t_{\text{зд.п.ср.}} = 0,5 \left( t_{\text{зд.п.}}^{0,1} + t_{\text{зд.п.}}^{1,0} \right), \quad (1.6)$$

где  $t_{\text{зд.п.}}^{0,1}$  и  $t_{\text{зд.п.}}^{1,0}$  – времена задержки распространения сигнала при переходе напряжения на выходе от низкого логического уровня к высокому и наоборот, соответственно.

*Работа переключения ЛЭ* – интегральный параметр, позволяющий сравнивать ЛЭ разного технологического исполнения:

$$A_{\text{пер}} = P_{\text{ср}} t_{\text{зд.п.ср.}} \quad (1.7)$$

### 1.3. Выполнение работы

#### Описание лабораторной установки

Лабораторная работа выполняется в пакете компьютерного моделирования *Electronic Workbench MultiSim*.

Запуск моделирования работы исследуемых схем выполняется с помощью кнопки **Run** («запуск») на панели инструментов. В процессе работы можно приостановить или остановить модели-



рование кнопками **Pause** («пауза») и **Stop** («остановить») соответственно.

В лабораторной работе используются, преимущественно, следующие элементы и инструменты:

Group: Sources (группа «Источники»)

Family: POWER\_SOURCES (подгруппа «Источники питания»)

Component (компонент) **"GROUND"** (земля)

**"VCC"** (источник питания напряжением +5 В)

Family: SIGNAL\_VOLTAGE\_SOURCES (подгруппа «Генераторы напряжения»)

Component (компонент) **"CLOCK\_VOLTAGE"** (генератор тактовых сигналов)

Group: Basic (группа «Пассивные элементы»)

Family: SWITCH (подгруппа «Переключатели»)

Component (компонент) **"SPDT"** (переключатель)

Family: BASIC\_VIRTUAL (подгруппа «Идеальные элементы»)

Component (компонент) **"RESISTOR\_VIRTUAL"** (сопротивление)

Group: Indicators (группа «Индикаторы»)

Family: PROBE (подгруппа «Пробники»)

Component (компонент) **"PROBE\_DIG\_RED"** (пробник логического уровня красный)

Group: TTL (группа «Цифровые ТТЛ-микросхемы»)

Family: 74LS (подгруппа «ТТЛ-микросхемы на диодах Шоттки с пониженной рассеиваемой мощностью»)

Component (компонент)

**"74LS04D"** (6 элементов «НЕ»)

**"74LS08D"** (4 2-х-входовых элемента «И»)

**"74LS00D"** (4 2-х-входовых элемента «И-НЕ»)

**"74LS32D"** (4 2-х-входовых элемента «ИЛИ»)

**"74LS02D"** (4 2-х-входовых элемента «ИЛИ-НЕ»)

**"74LS86D"** (4 2-х-входовых элемента «исключающее ИЛИ»)

**"74LS11D"** (3 3-х-входовых элемента «И»)

**"74LS10D"** (3 3-х-входовых элемента «И-НЕ»)

**"74LS27D"** (3 3-х-входовых элемента «ИЛИ-НЕ»)

**"74LS21D"** (2 4-х-входовых элемента «И»)

**"74LS20D"** (2 4-х-входовых элемента «И-НЕ»)

Group: CMOS (группа «Цифровые КМОП-микросхемы»)

Family: CMOS\_5V (подгруппа «КМОП-микросхемы с напряжением питания 5 В»)

Component (компонент)

**"4081BD\_5V"** (4 2-х-входовых элемента «И»)

Instruments (панель инструментов)

**"Functional Generator"** (генератор)

**"Oscilloscope"** (осциллограф)

### Описание осциллографа

Программа Electronics Workbench MultiSim имитирует аналог двухлучевого запоминающего осциллографа.

Для работы с осциллографом следует на панели инструментов (**Instruments**) нажать левой кнопкой мыши на кнопку **«Oscilloscope»**, поместить прибор в рабочее поле и вторично нажать левую кнопку мыши. На рабочем поле появится уменьшенное изображение осциллографа. На этом изображении имеется три входных зажима:

- боковой правый зажим – вход синхронизации;
- левый и правый нижние зажимы представляют собой соответственно вход канала А (**Channel A**) и вход канала В (**Channel B**).

Двойным щелчком мыши по уменьшенному изображению осциллографа открывается рабочее изображение передней панели осциллографа.

Для проведения измерений осциллограф нужно настроить:

- задать вид отображения сигналов. Нажать кнопку **«Y/T»**: по вертикальной оси откладывается напряжение, по горизонтальной оси – время.

- установить нужный масштаб развёртки по времени (**Timebase**) (горизонтальная развертка) – на панели управления горизонтальной разверткой (масштабом времени) **«Scale»**;

- установить нужный масштаб оси отображаемого напряжения. На поле канала А (**Channel A**) задать **«Scale»** 5В/дел (**V/div**), на поле канала В (**Channel B**) задать 2В/дел (**V/div**). При необходимости – скорректировать (изменить) масштаб отображаемого напряжения по каждому каналу;

- установить режим работы по входу для каждого канала. Кнопка **«AC»** – режим работы осциллографа с закрытым входом, кнопка **«DC»** – режим работы осциллографа с открытым входом.

- установить режим синхронизации: на панели **Trigger** нажать кнопку автоматического режима **«AUTO»** – запуск осциллограм-

мы производится автоматически при подключении осциллографа к схеме или при её включении.

### **Описание генератора**

Программа Electronics Workbench MultiSim имитирует аналог функционального генератора колебаний заданной формы положительной или отрицательной полярности:

Настройки генератора позволяют выбирать следующие параметры:

- форму колебаний – синусоидальную, треугольную (пилообразную), прямоугольную (меандр);
- частоту колебаний;
- амплитуду колебаний (**Amplitude**);
- смещение постоянной составляющей относительно нуля (**Offset**);
- симметричность полувольт (скважность).

### **Порядок выполнения работы**

#### **А) Элементарные логические функции и логические элементы.**

##### **Таблица истинности.**

Получить у преподавателя индивидуальное задание по исследованию логического элемента. Для заданного логического элемента (“И”, “ИЛИ”, “И-НЕ”, “ИЛИ-НЕ”, “исключающее ИЛИ”, “исключающее ИЛИ-НЕ”, “И-ИЛИ-НЕ” и т. д.) определить в форме таблицы выполняемую функцию (измерить таблицу истинности). Таблицу оформить аналогично табл. 1.1–1.3.

Для получения таблицы истинности собрать схему исследования своего логического элемента, аналогичную одной из изображенных на рис. 1.2.

##### *Примечания:*

1. По умолчанию переключатели на схеме рис. 1.2, а управляются клавишей «Пробел» ([Space]). Для независимого управления переключателями следует назначить разные управляющие клавиши.

2. На рис. 1.2, б изображена схема для получения таблицы истинности в динамическом режиме с использованием генераторов тактовых сигналов. В этом случае следует задать низкие частоты тактовых генераторов (порядка 0,1–1 Гц). Частоты должны отличаться в 2 раза.

3. Перед началом исследования таблицы истинности показать собранную схему преподавателю. После проверки таблицы истинности преподавателем схему удалить.

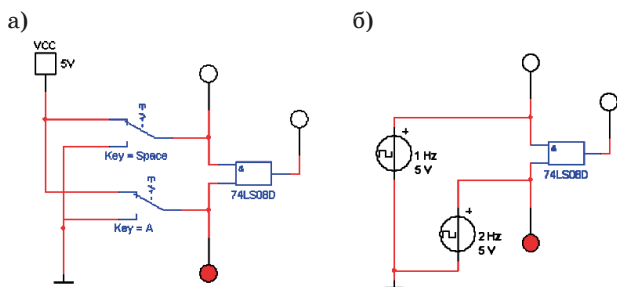


Рис. 1.2. Схема измерения таблицы истинности в статическом режиме (а) и динамическом режиме (б)

### Б) Измерение времени задержки. Влияние нагрузки на быстродействие элементов ТТЛ и КМОП.

Время задержки  $t_{\text{зд.р.ср.}}$  измеряется с использованием логического элемента 2И для двух типов микросхем – КМОП-микросхемы **4081BD\_5V** и ТТЛ-микросхемы **74LS08D**. Измерение проводится в двух режимах – без нагрузки и с нагрузкой.

Последовательно выбрать на панели элементов заданные микросхемы в группах **TTL** (подгруппа **74LS**) и **CMOS** (подгруппа **CMOS\_5V**) и собрать по очереди схемы, приведенные на рис. 1.3, для каждой микросхемы. На рис. 1.3, а приведена схема для измерений в режиме без нагрузки, на рис. 1.3, б – для измерений в режиме с нагрузкой.

При выполнении измерений установить на тактовом генераторе частоту:

1 МГц – для КМОП-микросхемы **4081BD\_5V**,

10 МГц – для ТТЛ-микросхемы **74LS08D**.

Настройки осциллографа:

- установить масштаб развертки по оси времени (**Timebase**) порядка 1–10 нс/дел;

- установить для канала А (входной сигнал) режим работы с закрытым входом (кнопка **AC**), для канала В (выходной сигнал) – режим работы с открытым входом (кнопка **DC**);

- измерения на осциллографе производить в режиме **Pause** или **Stop** с помощью разноцветных маркеров 1 и 2, расположенных по бокам экрана осциллографа.

Включить схему. Произвести измерения, как показано на рис. 1.4, где  $t_{\text{зд.р.}}^{1,0}$  время задержки при переходе выходного сигнала с высокого логического уровня на низкий;  $t_{\text{зд.р.}}^{0,1}$  – время задержки

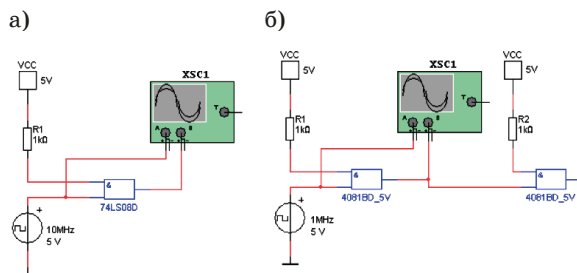


Рис. 1.3. Схема измерения быстродействия без нагрузки (а) и с нагрузкой (б)

при переходе выходного сигнала с низкого логического уровня на высокий.

Рассчитать среднее времени задержки по формуле (1.6). Результаты измерений занести в табл. 1.4.

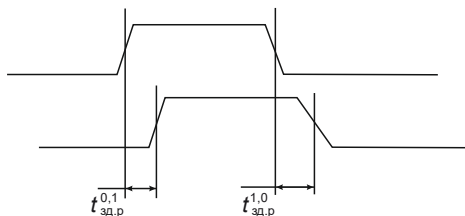


Рис. 1.4. Измерение быстродействия

Таблица 1.4

Результаты измерения быстродействия ЛЭ

ИС	Модель	$t_{зд.п.кр., НС}$	$t_{зд.п.кр._нагр., НС}$
ТТЛ	74LS08D		
КМОП	4081BD_5V		

*Примечание.* Исследуемые микросхемы содержат несколько одинаковых логических элементов. Можно исследовать любой из них.

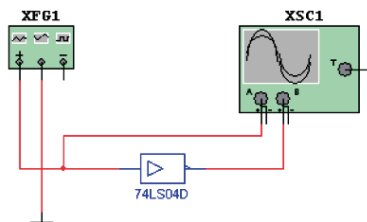


Рис. 1.5. Схема измерения помехоустойчивости

**В) Измерение пороговых уровней входных сигналов напряжения, соответствующих логической «1» и «0» ( $V_{IH}, V_{IL}$ ). Расчет помехоустойчивости.**

Собрать схему, показанную на рис. 1.5. Пороговые значения напряжений измеряются для логического элемента «НЕ», микросхема 74LS04D.

На генераторе “**Functional Generator**” сформировать симметричное пилообразное напряжение низкой частоты (0,1–0,5 Гц). Напряжение должно меняться от 0 до 5 В. Для этого задать следующие параметры: амплитуда (**Amplitude**) 2,5 В, смещение (**Offset**) 2,5 В.

На осциллографе:

- развертка по оси времени – порядка 0,5 с/дел;
- режим работы – с открытым входом (**DC**) – оба канала;
- развертка по вертикальной оси порядка 2–5 В/дел.

Включить схему. Произвести измерения  $U_{пор}^1$  и  $U_{пор}^0$ . Рассчитать помехоустойчивость по формулам:

$$U_{п}^1 = U^1 - U_{пор}^1, \quad (1.8)$$

$$U_{п}^0 = U_{пор}^0 - U^0. \quad (1.9)$$

Результаты измерений занести в табл. 1.5.

Таблица 1.5

Результаты измерения помехоустойчивости

ИС	Модель	$U_{пор}^0$ , В	$U^0$ , В	$U_{пор}^1$ , В	$U^1$ , В	$U_{п}^0$ , В	$U_{п}^1$ , В
ТТЛ	74LS04D		0,4		5		

## **Содержание отчета**

1. Цель работы;
2. Вариант задания;
3. Схемы, используемые для проведения измерений;
4. Таблицы с результатами измерений и вычислений;
5. Необходимые формулы;
6. Выводы по работе.

## **Контрольные вопросы**

1. Что такое цифровое устройство? Что такое цифровой сигнал?
2. Что такое логическая функция?
3. Перечислить логические функции 2-х переменных. Сколько логических функций можно сформировать для  $n$  переменных?
4. Способы описания логических функций. Таблица истинности. Логическое выражение.
5. Основные операции булевой алгебры. Приоритет логических операций.
6. Основные законы булевой алгебры.
7. Основные теоремы булевой алгебры.
8. Что такое логическая схема?
9. Определение базиса логической схемы. Какие базисы вы знаете?
10. Определение функционально полной системы.
11. Принципиальная схема базисного логического элемента ТТЛ. Принцип работы. Эпюры напряжений, поясняющие работу элемента. Недостатки ТТЛ.
12. Принципиальная схема базисного логического элемента КМОП. Принцип работы. Эпюры напряжений, поясняющие работу элемента. Недостатки КМОП.
13. Принципиальная схема базисного ЭСЛ-элемента. Принцип работы. Эпюры напряжений, поясняющие работу элемента.
14. Технические и эксплуатационные характеристики микросхем.
15. Сравнение технических характеристик микросхем, построенных на базе ТТЛ, КМОП, ЭСЛ.
16. Чем определяется время задержки распространения сигнала в ТТЛ-микросхемах и КМОП-микросхемах?
17. Какие технические характеристики микросхем ТТЛ и КМОП требуют улучшения? Методы улучшения технических характеристик микросхем ТТЛ и КМОП.
18. Что такое помехоустойчивость, помехозащищенность?

19. Методика измерения помехоустойчивости.

20. Сравнение теоретической помехоустойчивости ТТЛ и КМОП-микросхем.

21. Чем отличаются измерения с открытым входом и с закрытым входом на осциллографе?

22. Какие ГОСТы определяют условно-графические обозначения (УГО) логических элементов? Какие УГО логических элементов приняты по ГОСТу?



## 2. ЛАБОРАТОРНАЯ РАБОТА № 2

### «ИССЛЕДОВАНИЕ КОМБИНАЦИОННЫХ УСТРОЙСТВ»

*Цель работы:* получение первичных представлений о комбинационных устройствах и первичных навыков синтеза комбинационных устройств с заданной логикой работы.

#### 2.1. Общие сведения о комбинационных устройствах

##### Понятие комбинационного устройства

Комбинационным называют цифровое устройство, состояние выходов которого определяется исключительно состоянием его входов в данный момент времени. То есть комбинационное устройство (КУ) не обладает памятью (в отличие от последовательностных устройств, например, счетчиков).

Функция, выполняемая комбинационным устройством, описывается с помощью таблицы истинности (ТИ). Например, табл. 2.1 представляет собой ТИ полного двухразрядного дешифратора, преобразующего двоичный код на входе в активный логический уровень выхода с соответствующим этому коду номером.

*Таблица 2.1*

**Таблица истинности полного двухразрядного дешифратора**

Входы		Выходы			
$x_1$	$x_0$	$y_0$	$y_1$	$y_2$	$y_3$
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Помимо таблицы истинности, выполняемая комбинационным устройством функция может быть описана с помощью системы математических выражений из элементарных логических операторов, по одному выражению на каждый выход устройства. Каждое такое выражение представляет собой логическую функцию (ЛФ), выражающую зависимость состояния соответствующего выхода устройства от состояний входов. Например, для полного двухразрядного дешифратора можно записать следующую систему ЛФ:

$$\begin{cases} y_0(x_0, x_1) = \overline{x_0} \cdot \overline{x_1} \\ y_1(x_0, x_1) = x_0 \cdot \overline{x_1} \\ y_2(x_0, x_1) = \overline{x_0} \cdot x_1 \\ y_3(x_0, x_1) = x_0 \cdot x_1. \end{cases} \quad (2.1)$$

На основе системы ЛФ может быть построена цифровая схема из базовых логических элементов (ЛЭ), реализующая заданное КУ.

### **Дизъюнктивная и конъюнктивная нормальные формы**

Одна и та же ЛФ функция может быть записана различными логическими выражениями (и схематехнически реализована различными сочетаниями ЛЭ). С помощью дизъюнктивных и конъюнктивных нормальных форм решается задача создания универсальной формы записи математического выражения для заданной ЛФ. При этом вводится понятие термина.

Терм – это дизъюнкция (логическая сумма) или конъюнкция (логическое произведение) любого числа переменных, взятых с инверсией или без нее.

Так, дизъюнктивной нормальной формой (ДНФ) называется запись ЛФ в виде логической суммы конъюнктивных термов. В англоязычной литературе ДНФ обозначается аббревиатурой SOP («Sum Of Products», «сумма произведений»). Например, две некоторые ЛФ, записанные в ДНФ:

$$f_1(A, B, C) = A \cdot B \cdot C \vee \overline{A} \cdot B \cdot \overline{C}, \quad (2.2)$$

$$f_2(A, B, C, D) = A \cdot B \vee \overline{A} \cdot B \cdot \overline{C} \vee \overline{C} \cdot \overline{D} \vee D. \quad (2.3)$$

В свою очередь, конъюнктивной нормальной формой (КНФ) называют запись ЛФ в виде логического произведения дизъюнктивных термов:

$$f_3(A, B, C) = (A \vee B \vee C) \cdot (A \vee C) \cdot B. \quad (2.4)$$

В англоязычной литературе КНФ обозначается аббревиатурой POS («Product Of Sums», «произведение сумм»).

Стоит отметить, что в ДНФ все выражение равно логической единице, если единице равен хотя бы один терм; а в КНФ – только если единице равны все его термы.

Тем не менее, представление ЛФ в виде ДНФ или КНФ все еще не является единственным: в этих формах одна и та же функция

может быть записана разными выражениями. Для перехода к однозначной (канонической) форме записи логического выражения на ДНФ и КНФ накладываются три дополнительных ограничения:

- 1) не должно быть одинаковых термов,
- 2) в каждом терме каждый аргумент должен встречаться не более раза (не важно, с инверсией он или нет),
- 3) в каждом терме должны присутствовать все аргументы функции и всегда в одном порядке следования.

Если ДНФ или КНФ этим ограничениям удовлетворяют, то говорят, соответственно, о совершенной дизъюнктивной нормальной форме (СДНФ) и совершенной конъюнктивной нормальной форме (СКНФ). Примеры двух произвольных функций в виде СДНФ и СКНФ:

$$f_{\text{СДНФ}}(x_1, x_2, x_3) = x_1 \cdot \overline{x_2} \cdot x_3 \vee x_1 \cdot x_2 \cdot x_3 \vee x_1 \cdot \overline{x_2} \cdot x_3, \quad (2.5)$$

$$f_{\text{СКНФ}}(x_1, x_2, x_3) = (x_1 \vee \overline{x_2} \vee x_3)(\overline{x_1} \vee x_2 \vee \overline{x_3})(x_1 \vee x_2 \vee x_3). \quad (2.6)$$

Обращаясь к ранее приведенным примерам, можно сказать, что выражение (2.2) для  $f_1$  соответствует всем требованиям СДНФ, а выражение (2.3) для  $f_2$  не удовлетворяет третьему требованию, поэтому не является СДНФ.

Термы, входящие в СДНФ, называют минтермами, или конституэнтами единицы. Минтерм принимает значение единицы только при одном единственном наборе значений входящих в него переменных. Например, в выражении (2.5) минтерм  $x_1 \cdot x_2 \cdot x_3$  равен единице только если  $x_1 = 1$ ,  $x_2 = 0$ ,  $x_3 = 1$  и равен нулю во всех остальных случаях.

По аналогии, термы, входящие в СКНФ, называют макстермами, или конституэнтами нуля. Макстерм принимает значение нуля только при одном единственном наборе значений входящих в него переменных.

Любая ЛФ, не равная тождественно нулю, может быть представлена в виде СДНФ, причем такое представление единственно (изменение порядка следования термов в выражении во внимание не принимается). И любая ЛФ, не равная тождественно единице, может быть единственным образом представлена в виде СКНФ.

При проектировании цифровых устройств чаще используются СДНФ.

### Формирование СДНФ по таблице истинности

Рассмотрим произвольную логическую функцию  $F(x_1, x_2, x_3)$ , представленную в виде ТИ (табл. 2.2), и запишем для нее выражение в виде СДНФ.

В первую очередь, необходимо для каждого набора аргументов  $x_1, x_2, x_3$ , при котором функция  $F$  принимает значение «1» (т. е. для всех строк ТИ, в которых  $F = 1$ ), составить минтерм. В минтерм входят все три аргумента  $x_1, x_2, x_3$  либо с инверсией, либо без нее так, чтобы при значениях, которые эти три аргумента принимают в текущей строке ТИ, минтерм равнялся логической единице. Для этого аргументы, равные нулю, записываются в минтерм в инверсной форме, а равные единице – в прямой.

И далее все полученные минтермы следует объединить в одну функцию операций дизъюнкции:

$$F(x_1, x_2, x_3) = x_1 \cdot \overline{x_2} \cdot \overline{x_3} \vee x_1 \cdot x_2 \cdot \overline{x_3} \vee x_1 \cdot \overline{x_2} \cdot x_3 \vee x_1 \cdot x_2 \cdot x_3. \quad (2.7)$$

Полученное выражение  $F$  принимает значение «1», если хотя бы одно из слагаемых равно «1». Четыре минтерма в выражении как раз соответствуют четырем строкам ТИ, в которых функция  $F$  принимает значение «1». Во всех остальных комбинациях входных аргументов ни один из минтермов не будет равен единице, соответственно, вся логическая сумма будет равна нулю.

### Минимизация ЛФ с помощью алгебраических преобразований

Полученное по таблице истинности выражение в СДНФ может быть упрощено, чтобы в дальнейшем синтезируемая по нему схема цифрового устройства содержала минимальное количество элементов, и, следовательно, была проще, надежнее и дешевле в реализации.

Таблица 2.2

Формирование минтермов СДНФ по таблице истинности

$x_3$	$x_2$	$x_1$	$F$	Минтерм
0	0	0	0	—
0	0	1	1	$x_1 \cdot \overline{x_2} \cdot \overline{x_3}$
0	1	0	0	—
0	1	1	1	$x_1 \cdot x_2 \cdot \overline{x_3}$
1	0	0	0	—
1	0	1	1	$x_1 \cdot \overline{x_2} \cdot x_3$
1	1	0	1	$\overline{x_1} \cdot x_2 \cdot x_3$
1	1	1	0	—

В табл. 2.3 приведены основные законы алгебры логики, позволяющие осуществить минимизацию ЛФ. В силу того, что алгебра логики оперирует всего двумя значениями переменных (логический ноль и логическая единица), законы алгебры логики могут быть легко доказаны подстановкой в выражения всех возможных комбинаций значений аргументов.

Таблица 2.3

**Основные законы алгебры логики**

Наименование закона	Математическое описание
Переместительный (коммутативный) закон	$x \cdot y = y \cdot x$ $x \vee y = y \vee x$
Сочетательный (ассоциативный) закон	$(x \cdot y) \cdot z = x \cdot (y \cdot z) = x \cdot y \cdot z$ $(x \vee y) \vee z = x \vee (y \vee z) = x \vee y \vee z$
Распределительный (дистрибутивный) закон	$x \cdot y \vee x \cdot z = x \cdot (y \vee z)$ $(x \vee y) \cdot (x \vee z) = x \vee y \cdot z$
Закон идемпотентности (повторения, тавтологии)	$x \cdot x = x$ $x \vee x = x$
Законы отрицания	$x \vee \bar{x} = 1$ (закон исключенного третьего) $x \cdot \bar{x} = 0$ (закон противоречия) $0 \vee x = x$ $1 \cdot x = x$ $1 \vee x = 1$ $0 \cdot x = 0$
Законы де Моргана (законы инверсии)	$\overline{x \vee y} = \bar{x} \cdot \bar{y}$ $\overline{x \cdot y} = \bar{x} \vee \bar{y}$
Закон двойного отрицания	$\bar{\bar{x}} = x$
Законы поглощения (абсорбция)	$x \vee x \cdot y = x$ $x \cdot (x \vee y) = x$
Операции склеивания	$x \cdot y \vee x \cdot \bar{y} = x$ $(x \vee y) \cdot (x \vee \bar{y}) = x$
Операции обобщенного склеивания	$x \vee \bar{x} \cdot y = x \vee y$ $x \cdot (\bar{x} \vee y) = x \cdot y$

Согласно закону идемпотентности повторяющиеся члены логического произведения или логической суммы можно опускать. В законе исключенного третьего из-за наличия инверсии в любом случае единице будет равно либо первое слагаемое, либо второе. В законе противоречия по этой же причине оба множителя никак не могут быть одновременно равны единице, всегда один из них равен нулю, поэтому и произведение равно нулю.

Законы де Моргана позволяют раскрывать общую для нескольких переменных инверсию, или, наоборот, заменять несколько отдельных инверсий на одну общую (рис. 2.1). Следует отметить, что законы де Моргана действительны и для большего количества аргументов. Также из этих двух законов следует, что в процессе преобразования ЛФ нельзя заменять отдельные инверсии аргументов на общие и наоборот без изменения связывающих их функций, т. е.  $x \cdot y \neq \overline{x \cdot y}$ .

Распределительные законы указывают на возможность вынесения общего множителя за скобки и на возможность факторизации выражения, т. е. разложения его на множители. Для логических выражений действуют также переместительный и сочетательный законы.

Рассмотрим пример минимизации с помощью аксиом и законов алгебры логики следующей ЛФ:

$$Y = A \cdot B \cdot C \vee A \cdot \overline{B} \cdot \overline{\overline{A} \cdot \overline{C}}. \quad (2.8)$$

В правой части имеется общая для нескольких множителей инверсия, ее можно разделить, применив закон де Моргана:  $Y = A \cdot B \cdot C \vee A \cdot \overline{B} \cdot (\overline{\overline{A} \vee \overline{C}})$ .

Возникшие над аргументами  $A$  и  $C$  двойные отрицания можно отбросить (закон двойного отрицания), а затем раскрыть скобки (распределительный закон):  $Y = A \cdot B \cdot C \vee A \cdot \overline{B} \cdot \overline{A} \vee A \cdot \overline{B} \cdot C$ .

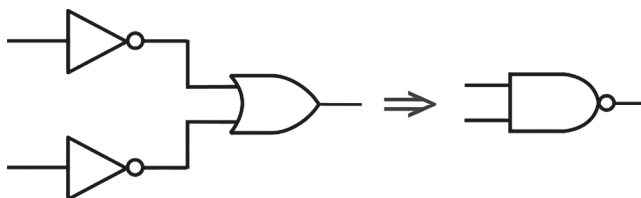


Рис. 2.1. Применение закона де Моргана

В среднем слагаемом присутствует два одинаковых множителя  $A$ , в соответствии с законом идемпотентности один из них можно опустить. В результате, у двух последних слагаемых окажется общий множитель  $A \cdot \bar{B}$ , который можно вынести за скобки (либо сразу применить одну из форм закона поглощения):  $Y = A \cdot B \cdot C \vee A \cdot \bar{B} \cdot (1 \vee C) = A \cdot B \cdot C \vee A \cdot \bar{B}$ .

Наконец, в получившемся выражении можно также вынести за скобки общий множитель  $A$  и к выражению в скобках применить одну из форм операции обобщенного склеивания:  $Y = A \cdot (B \cdot C \vee \bar{B}) = A \cdot (C \vee \bar{B})$ .

### Минимизация ЛФ с помощью карты Карно

Карта Карно, известная также как К-карта, или диаграмма Вейча, – это удобное графическое средство для формирования упрощенных выражений для логических функций, заданных таблицей истинности или СДНФ. По своей сути, карта Карно является таблицей истинности, представленной в несколько иной форме. Карта Карно представляет собой таблицу, все строки и столбцы которой подписаны аргументами функции и в прямой, и в инверсной формах, а ячейки заполнены нулями и единицами. Количество ячеек карты Карно определяется выражением  $2^N$ , где  $N$  – количество аргументов функции, и, соответственно, равно количеству строк в ТИ, определяющемуся аналогичным выражением.

Рассмотрим простой пример построения карты Карно для функции двух аргументов, ТИ и минтермы которой показаны в табл. 2.4.

Карта Карно для такой ЛФ будет содержать четыре ячейки. Строки карты обозначим прямой и инверсной формой аргумента  $A$ , а столбцы – аргумента  $B$  (можно разместить и наоборот). Каждой строке ТИ соответствует ровно одна ячейка карты Карно, координаты которой определяются значениями аргументов в этой стро-

Таблица 2.4

Таблица истинности и минтермы функции двух аргументов

$A$	$B$	$F$	Минтерм
0	0	1	$\bar{A} \cdot \bar{B}$
0	1	0	—
1	0	0	—
1	1	1	$A \cdot B$

ке ТИ: единица соответствует прямой форме, а ноль – инверсной. Содержимое ячейки карты Карно – это значение ЛФ в этой строке ТИ. Например, в первой строке ТИ оба аргумента равны нулю, т. е. соответствуют инверсным формам аргументов, поэтому соответствующая ячейка находится на пересечении строки  $\bar{A}$  и столбца  $\bar{B}$ . В эту ячейку следует записать из ТИ соответствующее значение функции, то есть единицу. Такую процедуру следует провести для всех строк ТИ, заполнив тем самым все ячейки карты (табл. 2.5).

В случае если у функции больше двух аргументов, по строкам и столбцам карты размещают обозначения сразу нескольких аргументов во всех возможных комбинациях инверсий. При этом важно, чтобы комбинации в соседних строках или столбцах отличались ровно на одну инверсию (что соответствует коду Грея). Пример для некоторой функции трех аргументов показан в табл. 2.6.

Для составления по карте Карно логического выражения в СДНФ достаточно записать логическую сумму минтермов, составляемых для всех ячеек карты со значением «1». Минтерм при этом записывается в соответствии с обозначением соответствующей

Таблица 2.5

Карта Карно функции двух аргументов

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

Таблица 2.6

Карта Карно функции трех аргументов

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



строки и столбца ячейки. Для табл. 2.6 можно записать следующее выражение:

$$f_{\text{СДНФ}}(A, B, C) = A \cdot B \cdot C \vee A \cdot \bar{B} \cdot C \vee \bar{A} \cdot \bar{B} \cdot C. \quad (2.9)$$

В то же время, карта Карно позволяет получить выражение сразу в минимизированной форме. Для этого предварительно необходимо сгруппировать ячейки со значением «1» по две/четыре/восемь штук в группе (такие группы называются, соответственно, парами, квартетами и октетами). При этом объединение возможно только в виде столбца, строки или прямоугольника (в том числе квадрата). Объединение по диагонали или с изгибами группы не допускается. При объединении ячеек в группы следует учитывать свойство цикличности карты Карно: можно считать, что первая строка карты является соседней для последней строки, аналогично с первым и последним столбцами. Также допустимо пересечение нескольких групп и наличие в карте групп разных размеров. Лучшую минимизацию дадут наиболее крупные группы. Пример группирования для карты из табл. 2.6 показан на рис. 2.2.

После выделения групп можно записать минимизированное выражение в виде ДНФ. Для единиц, которые оказалось невозможно ни с кем объединить, по-прежнему записывается слагаемое в соответствии с обозначением ячейки. Для каждой группы единиц записывается слагаемое, состоящее из неизменных для всей группы аргументов. Например, на рис. 2.2 для верхней группы (единица в левом верхнем и правом верхнем углах), аргументы  $A$  и  $C$  в одинаковой форме для обеих единиц, а аргумент  $B$  в выражение ДНФ записан не будет, так как для одной единицы он записан в прямой форме, для второй – в инверсной. В результате, для

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

Рис. 2.2. Объединение единиц в группы в карте Карно

обеих выделенных групп можно записать следующее итоговое выражение:

$$f_{\text{ДНФ}}(A, B, C) = A \cdot C \vee \bar{B} \cdot C. \quad (2.10)$$

Выражения (2.9) и (2.10) дают эквивалентный результат.

Объединение единиц в пары позволяет отбросить один аргумент (в (2.9) все слагаемые содержат по три аргумента, а в (2.10) после объединения в пары – по два). Объединение единиц в квартеты позволяет отбросить два аргумента, в октеты – три.

## 2.2. Синтез дешифратора для управления сегментом индикатора

Сформируем логическую схему, обеспечивающую работу сегмента «G» семисегментного индикатора (рис. 2.3) по входному двоичному коду шестнадцатеричного символа (рис. 2.4).

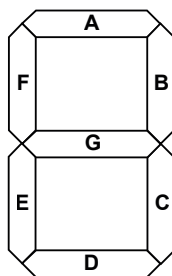


Рис. 2.3. Обозначение сегментов семисегментного индикатора



Рис. 2.4. Шестнадцатеричные символы на индикаторе

### Составление таблицы истинности

В первую очередь необходимо составить ТИ, описывающую работу требуемой схемы (табл. 2.7). Заполнение столбца «G» происходит в соответствии с отображаемым на индикаторе символом для действующего на входе двоичного кода (рис. 2.4): если сегмент светится, то в таблицу записывается единица, иначе – ноль.

Таблица истинности дешифратора сегмента «G»

Шестнадцатиричный символ на индикаторе	Двоичный код				$G$	$\overline{G}$
	$Q_3$	$Q_2$	$Q_1$	$Q_0$		
0	0	0	0	0	0	1
1	0	0	0	1	0	1
2	0	0	1	0	1	0
3	0	0	1	1	1	0
4	0	1	0	0	1	0
5	0	1	0	1	1	0
6	0	1	1	0	1	0
7	0	1	1	1	0	1
8	1	0	0	0	1	0
9	1	0	0	1	1	0
A	1	0	1	0	1	0
B	1	0	1	1	1	0
C	1	1	0	0	0	1
D	1	1	0	1	1	0
E	1	1	1	0	1	0
F	1	1	1	1	1	0

### Составление и минимизация логической функции

По составленной ТИ сформируем выражение в виде СДНФ и затем произведем его минимизацию на основе аксиом и законов алгебры логики. По табл. 2.7 можно заметить, что столбец  $G$  равен единице в большинстве строк, поэтому СДНФ будет содержать большое количество слагаемых. Для экономии времени и упрощения расчетов можно составить СДНФ для инверсной функции  $\overline{G}$ , которая будет принимать значение единицы всего в четырех строках ТИ (в строках, в которых основная функция в табл. 2.7 равна нулю). В дальнейшем для перехода от инверсной функции обратно к прямой достаточно будет применить к полученному результату операцию инверсии согласно закону двойного отрицания.

$$\overline{G} = \overline{Q_3} \cdot \overline{Q_2} \cdot \overline{Q_1} \cdot \overline{Q_0} \vee \overline{Q_3} \cdot \overline{Q_2} \cdot \overline{Q_1} \cdot Q_0 \vee \overline{Q_3} \cdot Q_2 \cdot \overline{Q_1} \cdot \overline{Q_0} \vee Q_3 \cdot Q_2 \cdot \overline{Q_1} \cdot \overline{Q_0}. \quad (2.11)$$

Выполним минимизацию полученного выражения для сокращения количества используемых ЛЭ:

$$\begin{aligned}
\overline{G} &= \overline{Q_3 \cdot Q_2 \cdot Q_1 \cdot Q_0} \vee \overline{Q_3 \cdot Q_2 \cdot Q_1 \cdot Q_0} \vee \overline{Q_3 \cdot Q_2 \cdot Q_1 \cdot Q_0} \vee \overline{Q_3 \cdot Q_2 \cdot Q_1 \cdot Q_0} = \\
&= \overline{Q_3 \cdot Q_2 \cdot Q_1 \cdot Q_0} \vee \overline{Q_3 \cdot Q_0} \cdot (\overline{Q_2 \cdot Q_1} \vee \overline{Q_2 \cdot Q_1}) \vee \overline{Q_3 \cdot Q_2 \cdot Q_1 \cdot Q_0} = \\
&= \overline{Q_3 \cdot Q_2 \cdot Q_1 \cdot Q_0} \vee \overline{Q_3 \cdot Q_0} \cdot (\overline{Q_2 \equiv Q_1}) \vee \overline{Q_3 \cdot Q_2 \cdot Q_1 \cdot Q_0} = \\
&= \overline{Q_1 \cdot Q_0} \cdot (\overline{Q_3 \cdot Q_2} + \overline{Q_3 \cdot Q_2}) \vee \overline{Q_3 \cdot Q_0} \cdot (\overline{Q_2 \equiv Q_1}) = \\
&= \overline{Q_1 \cdot Q_0} \cdot (\overline{Q_3 \equiv Q_2}) \vee \overline{Q_3 \cdot Q_0} \cdot (\overline{Q_2 \equiv Q_1}) = \\
&= \overline{Q_1 \cdot Q_0} \cdot (\overline{Q_3 \oplus Q_2}) \vee \overline{Q_3 \cdot Q_0} \cdot (\overline{Q_2 \oplus Q_1}) = \\
&= \overline{Q_1 \vee Q_0 \vee (Q_3 \oplus Q_2)} \vee \overline{Q_3 \vee Q_0 \vee (Q_2 \oplus Q_1)}.
\end{aligned} \tag{2.12}$$

Символом  $\equiv$  обозначена операция логической равнозначности (исключающее ИЛИ-НЕ, эквивалентность), символом  $\oplus$  – операция логической неравнозначности (исключающее ИЛИ, сложение по модулю 2). Переходя от инверсной функции обратно к прямой, получаем:

$$G = \overline{\overline{G}} = \overline{Q_1 \vee Q_0 \vee (Q_3 \oplus Q_2)} \vee \overline{Q_3 \vee Q_0 \vee (Q_2 \oplus Q_1)}. \tag{2.13}$$

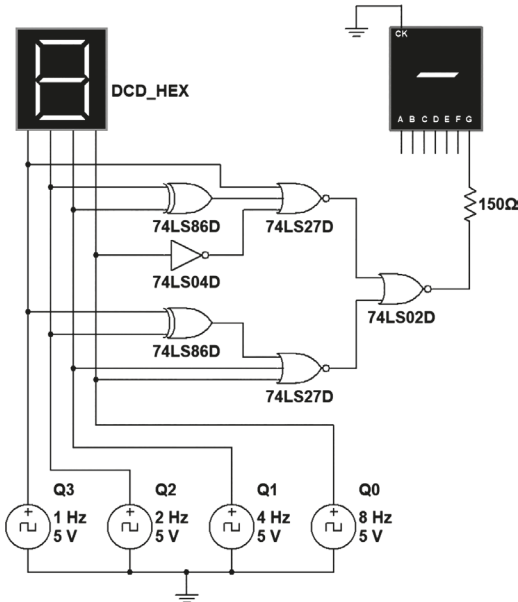


Рис.2.5. Рабочая схема управления сегментом «G» индикатора

## Реализация логической функции на основе комбинационных цифровых элементов

Для реализации рабочей схемы по полученному выше выражению (2.13) потребуются следующие элементы:

- один элемент НЕ,
- два элемента исключающее ИЛИ,
- два элемента ЗИЛИ-НЕ,
- один элемент 2ИЛИ-НЕ.

Соответствующая рабочая схема, включающая также блок генерации тестового сигнала четырехразрядного двоичного кода и эталонный семисегментный индикатор со встроенным дешифратором, показана на рис. 2.5. При включении схемы на эталонном индикаторе отображаемые символы будут последовательно меняться от «F» до «0». При правильном выполнении всех этапов синтеза на тестовом индикаторе сегмент «G» будет загораться и гаснуть синхронно с сегментом «G» эталонного индикатора. Тестовый индикатор следует выбирать с общим катодом (сверху имеется вывод «СК»), а сигнал к его входам следует подводить через токоограничивающие резисторы.

### 2.3. Выполнение лабораторной работы

#### Задание на лабораторную работу

В рамках лабораторной работы необходимо выполнить синтез цифровой комбинационной схемы, управляющей одним сегментом семисегментного или 14-сегментного индикатора (рис. 2.6).

#### Варианты заданий

Задания поделены на три группы в зависимости от типа выводимой на индикатор последовательности символов. Варианты за-

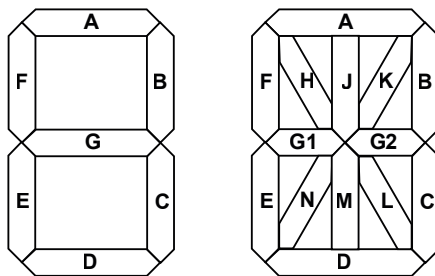


Рис. 2.6. Обозначения сегментов семисегментного и 14-сегментного индикаторов

даний приведены в табл. 2.8, пояснения по особенностям каждой группы заданий даны после таблицы.

**Группа А.** Семисегментный индикатор, шестнадцатеричный алфавит. Эталонный индикатор подключается в соответствии с рис. 2.5. Вид отображаемых на индикаторе символов показан на рис. 2.4, соответствия двоичного кода отображаемым символам – в табл. 2.9. Номер управляемого сегмента определяется по столбцу «Сегмент» табл. 2.8.

*Таблица 2.8*

**Варианты заданий**

Вариант	Сегмент	Описание
1	A	Группа заданий А: семисегментный индикатор, символы шестнадцатеричного алфавита
2	B	
3	C	
4	D	
5	E	
6	F	
7	A	Группа заданий Б: 14-сегментный индикатор, первые 16 символов русского алфавита
8	B	
9	C	
10	D	
11	E	
12	F	
13	G1	
14	G2	
15	K	
16	A	Группа заданий В: 14-сегментный индикатор, первые 16 символов современного латинского алфавита
17	B	
18	C	
19	D	
20	G1	
21	G2	

**Группа Б.** 14-сегментный индикатор, первые 16 символов русского алфавита. Эталонный индикатор подключается с помощью иерархического блока G16RUSHB в соответствии с рис. 2.7. Добавление иерархического блока на схему осуществляется одновременным нажатием клавиш <Ctrl + H>, в открывшемся окне проводника следует выбрать файл «.../Multisim/G16RUSHB». Вид отображаемых на индикаторе символов показан на рис. 2.8, соответствия двоичного кода отображаемым символам – в табл. 2.9. Номер управляемого сегмента определяется по столбцу «Сегмент» табл. 2.8.

**Группа В.** 14-сегментный индикатор, первые 16 символов современного латинского алфавита. Эталонный индикатор подключается с помощью иерархического блока G16ENGHB по аналогии с рис. 2.7. Добавление иерархического блока на схему осуществляется одновременным нажатием клавиш <Ctrl + H>, в открывшемся окне проводника следует выбрать файл «.../Multisim/G16ENGHB». Вид отображаемых на индикаторе символов показан на рис. 2.9, соответствия двоичного кода отображаемым символам – в табл. 2.9. Номер управляемого сегмента определяется по столбцу «Сегмент» табл. 2.8.

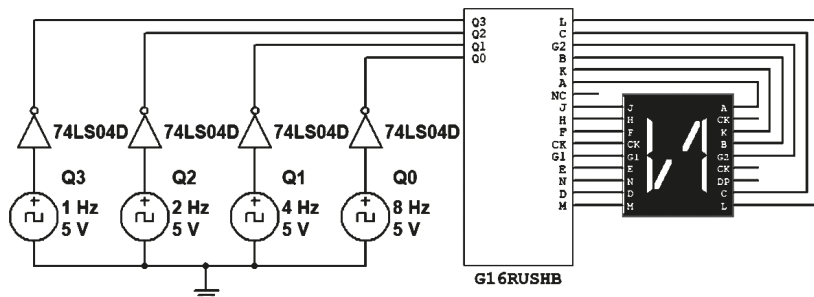


Рис. 2.7. Подключение иерархического блока

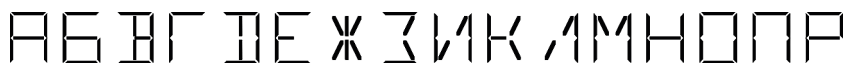


Рис. 2.8. Символы русского алфавита на индикаторе



Рис. 2.9. Символы латинского алфавита на индикаторе

Таблица 2.9

Соответствия двоичного кода отображаемым на индикаторе символам

Двоичный код символа				Символ		
Q3	Q2	Q1	Q0	Группа А	Группа Б	Группа В
0	0	0	0	0	А	А
0	0	0	1	1	Б	В
0	0	1	0	2	В	С
0	0	1	1	3	Г	Д
0	1	0	0	4	Д	Е
0	1	0	1	5	Е	Ф
0	1	1	0	6	Ж	Г
0	1	1	1	7	З	И
1	0	0	0	8	И	Й
1	0	0	1	9	К	Л
1	0	1	0	А	Л	М
1	0	1	1	В	М	Н
1	1	0	0	С	Н	О
1	1	0	1	Д	О	П
1	1	1	0	Е	П	Р
1	1	1	1	Ф	Р	С

**Порядок выполнения работы**

- 1) Изучить методические указания.
- 2) Составить ТИ для своего варианта задания по примеру табл. 2.7.
- 3) Записать аналитическое выражение ЛФ и произвести его минимизацию с помощью аксиом и законов алгебры логики.
- 4) Составить по ТИ карту Карно и записать по ней минимизированное логическое выражение.
- 5) Используя полученное на шаге 3) или 4) выражение, смоделировать в программе Multisim рабочую цифровую схему на ТТЛ-элементах серии 74LS и проверить корректность ее работы.
- 6) Для каждого вида использованных в схеме ТТЛ-элементов подобрать по справочнику аналог среди отечественных ИМС серии К555.
- 7) Сделать выводы по проделанной работе.



## **Содержание отчета**

1. Цель работы.
2. Вариант задания.
3. Таблица истинности, аналитические выражения ЛФ.
4. Карта Карно, минимизированное аналитическое выражение ЛФ.
5. Результат синтеза рабочей цифровой схемы в программе Multisim.
6. Схема электрическая принципиальная синтезированного устройства, выполненная в соответствии с ГОСТ.
7. Отечественные аналоги ИМС в виде таблицы со столбцами:
  - наименование ИМС 74LS,
  - наименование ИМС К555,
  - функциональное назначение,
  - количество вентилях в схеме,
  - количество вентилях в одной ИМС,
  - требуемое количество ИМС.
8. Спецификация на выбранные отечественные микросхемы.
9. Выводы по работе.

## **Контрольные вопросы**

1. Правила перевода чисел из одной системы счисления в другую.
2. Двоичная арифметика: правила сложения, вычитания, умножения, деления двоичных чисел.
3. Формы представления чисел в ЭВМ.
4. Что такое прямой, обратный, дополнительный код? Их назначение. Правила формирования обратного и дополнительного кода.
5. Что такое КНФ, ДНФ, СКНФ, СДНФ.
6. Алгоритм формирования СКНФ, СДНФ.
7. Алгоритм преобразования КНФ в СКНФ, ДНФ в СДНФ.
8. Для чего нужна минимизация КНФ, ДНФ?
9. Алгоритм минимизации СДНФ с помощью карты Карно.
10. Как выглядит карта Карно для 3-х, 4-х, 5-ти, 6-ти переменных?
11. Что такое комбинационное устройство?
12. Алгоритм (этапы) синтеза комбинационного устройства.
13. Какие типы комбинационных устройств вы знаете? Чем они функционально отличаются друг от друга?
14. Приведите примеры маркировки цифровых микросхем отечественного производства, содержащих логические элементы «И», «ИЛИ», «НЕ», «И-НЕ», «ИЛИ-НЕ», а также отдельные комбинационные устройства. Каковы их технические и эксплуатационные характеристики. Чем могут различаться микросхемы одинакового функционального назначения?

### 3. ЛАБОРАТОРНАЯ РАБОТА № 3

#### «ИССЛЕДОВАНИЕ ПОСЛЕДОВАТЕЛЬНОСТНЫХ УСТРОЙСТВ»

*Цель работы:* получение первичных представлений об элементарных последовательностных устройствах, счетчиках, видах синхронизации и способах описания последовательностных устройств. Приобретение навыков синтеза счетчиков с заданным модулем счета.

#### 3.1. Триггер как элементарное последовательностное устройство

*Последовательностным* называют цифровое устройство, обладающее памятью. Состояние последовательностного устройства определяется не только действующими в данный момент сигналами на его входах (как это происходит у комбинационных устройств), но также состоянием входов и выходов этого устройства в предыдущие моменты времени.

Основу последовательностных устройств составляют *триггеры* – элементарные ячейки полупроводниковой памяти, способные принимать два устойчивых состояния, то есть хранить один бит информации. На триггерах строится весь широкий спектр последовательностных устройств: счетчики, регистры, запоминающие устройства.

Триггеры бывают *асинхронные* (изменяют свое состояние сразу после установки сигналов на его входах) и *синхронные* (моменты изменения состояния триггера определяются специальным тактовым сигналом, или синхросигналом). В свою очередь, синхронные триггеры бывают со *статическим управлением* (изменяют свое состояние только при подаче на вход тактового сигнала логической единицы; гораздо реже – логического нуля) и синхронные с *динамическим управлением* (изменяют свое состояние в моменты перехода тактового сигнала из логического нуля в логическую единицу, или наоборот).

Триггеры со статическим и с динамическим управлением можно различить по способу изображения тактового входа на их условно-графическом обозначении (рис. 3.1). В случае динамического управления изображение тактового входа дополняется обозначениями, указывающими направление перехода тактового сигнала, на которое реагирует триггер. Если триггер переключается в моменты перехода тактового сигнала из «нуля» в «единицу», то говорят, что

он реагирует на *передний*, или нарастающий, *фронт* тактового сигнала, а если в моменты перехода из «единицы» в «ноль», то говорят, что триггер реагирует на задний фронт (или *срез, спад*).

Для построения счетчиков импульсов используются синхронные триггеры с динамическим управлением.

Существуют четыре основных типа синхронных триггеров, их типовые обозначения приведены на рис. 3.2:

- RS-триггер (от англ. Reset и Set – сбросить и установить),
- JK-триггер (по имени одного из его изобретателей – Jack Kilby; также в качестве мнемоник используют англ. слова Jump и Kill – резко установить и резко сбросить),
- D-триггер (от англ. Data/Delay – данные, задержка),
- T-триггер (от англ. Toggle – переключать; также называется счетным триггером).

Работа триггеров RS и JK удобно представляется таблицей состояний: таблицы 3.1 и 3.2. Эти триггеры относятся к триггерам

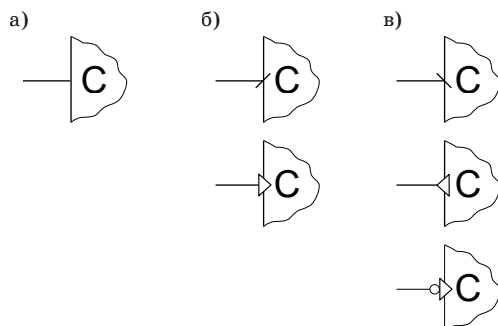


Рис. 3.1. Обозначения входа тактового сигнала синхронных триггеров: со статическим управлением (а); с динамическим управлением по переднему фронту (б) и по заднему фронту (в)

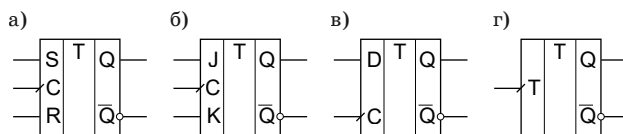


Рис. 3.2. Обозначения триггеров: RS-триггер (а), JK-триггер (б), D-триггер (в), T-триггер (г)

с отдельной установкой состояний, так как для записи в триггер логической единицы (то есть установки триггера) и логического нуля (сброса триггера) используются два отдельных входа. При подаче неактивных логических уровней на оба входа такие триггеры будут работать в режиме хранения, то есть ранее установленное состояние изменяться не будет. Различие в работе RS- и JK-триггеров проявляется только при одновременной подаче на оба входа активных сигналов: для RS-триггера такая входная комбинация является запрещенной и не используется, а JK-триггер в этом случае работает в счетном режиме, то есть на каждом импульсе тактового сигнала меняет хранимое значение на противоположное.

Таблица 3.1

Таблица состояний RS-триггера

R	S	Q	Режим
0	0	$Q_{-1}$	хранение
0	1	1	установка
1	0	0	сброс
1	1	—	запрещенная комбинация

Таблица 3.2

Таблица состояний JK-триггера

R	S	Q	Режим
0	0	$Q_{-1}$	хранение
0	1	1	установка
1	0	0	сброс
1	1	$\overline{Q_{-1}}$	счетный

D-триггер имеет всего один информационный вход. Действующее на этом входе логическое значение записывается в триггер на каждом импульсе тактового сигнала.

T-триггер не имеет информационных входов и постоянно работает в счетном режиме, то есть на каждом импульсе тактового сигнала меняет свое состояние на противоположное.

Любые типы триггеров, как правило, кроме основного выхода  $Q$  имеют также инверсный  $\bar{Q}$ : состояния этих двух выходов в любой момент времени имеют противоположные значения.

Для реализации счетчиков импульсов необходимы триггеры, работающие в счетном режиме. Ранее уже было показано, что такие режимы имеются у JK- и T-триггеров. D-триггер можно перевести в счетный режим, соединив между собой его инверсный выход со входом данных: в этом случае, действительно, на каждом тактовом импульсе триггер будет принимать состояние, противоположное предыдущему. RS-триггер тоже можно перевести в счетный режим, однако для этого потребуются применение дополнительных элементарных комбинационных элементов. На практике для построения счетчиков чаще всего используются JK- и D-триггеры, так как они присутствуют в номенклатуре большинства серий цифровых микросхем и не требуют дополнительных элементов для перевода в счетный режим работы.

## 3.2. Счетчики

### Счетчик импульсов с последовательным переносом

Рассмотрим соединение нескольких триггеров, работающих в счетном режиме, последовательно, как показано на рис. 3.3.

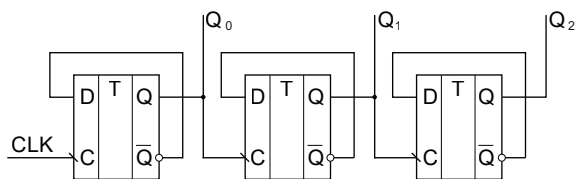


Рис. 3.3. Трехразрядный счетчик с последовательным переносом на D-триггерах

Временные диаграммы работы такой схемы приведены на рис. 3.4. Каждый триггер переключается по заднему фронту сигнала  $CLK$ , поступающего на его тактовый вход  $C$ . Стоит отметить, что переключение триггера происходит не моментально, а с небольшой задержкой. Поэтому после прихода фронта тактового сигнала на выходе триггера еще некоторое время действует предыдущее состояние. Так как триггеры соединены последовательно, и каждый из них вносит дополнительную задержку, переключение состояний

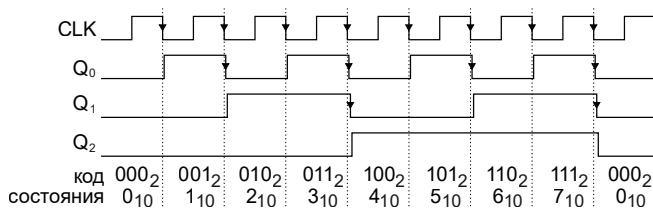


Рис. 3.4. Временные диаграммы работы трехразрядного счетчика

триггеров в такой схеме происходит не одновременно, и наибольшая задержка переключения наблюдается у последнего триггера.

Если представить выходы триггеров как разряды двоичного числа (левый триггер при этом соответствует младшему разряду), то можно заметить, что формируемые при этом двоичные числа на каждом такте последовательно будут меняться от  $000_2$  до  $111_2$ , после чего все сигналы начнут повторяться заново. Таким образом можно сделать вывод, что на рис. 3.3 изображен трехразрядный *суммирующий* или *прямого счета* (то есть считающий от меньшего к большему) счетчик с периодом повторения состояний, равным восьми тактам синхросигнала.

Количество состояний, которое может принимать счетчик, является его важнейшей характеристикой и называется *модулем счета*, или *коэффициентом пересчета*. Модуль счета  $M$  связан с количеством триггеров  $N$ , использованных при построении счетчика, выражением  $M = 2^N$ .

В рассмотренной схеме можно заменить триггеры, реагирующие на срез синхросигнала, на триггеры, реагирующие на передний фронт. Либо тактовый вход каждого следующего триггера подключать не к прямому выходу предыдущего, а к инверсному. Путем анализа временных диаграмм можно будет увидеть, что состояния счетчика при любом из двух предложенных изменений схемы будут следовать в обратном направлении: от  $111_2$  до  $000_2$ . Такой счетчик будет *вычитающим*, или *счетчиком обратного счета*.

В случае применения сразу обоих изменений, то есть при использовании триггеров, реагирующих на передний фронт синхросигнала и при соединении триггеров через инверсные выходы, счетчик получится снова суммирующий.

Также можно заметить, что эффекта вычитающего счетчика можно достигнуть на суммирующем счетчике, если снимать выходные сигналы не с прямых выходов триггеров, а с инверсных.

Достоинством счетчика с последовательным переносом является его простота, а недостатком – неодновременность переключения триггеров и связанная с этим общая задержка в установке нового состояния счетчика, увеличивающаяся с ростом количества триггеров в схеме.

### Счетчик импульсов с параллельным переносом

В счетчике с параллельным переносом все триггеры имеют общий тактовый сигнал, и поэтому переключаются одновременно. В то же время, чтобы триггеры при таком подключении работали как счетчик, необходимо обеспечить своевременное переключение каждого из них. Если посмотреть на временные диаграммы на рис. 3.4, можно заметить, что каждый триггер меняет свое состояние на противоположное всякий раз, когда все триггеры левее его к моменту прихода очередного активного фронта синхросигнала находились в состоянии логической единицы. Эта особенность и положена в основу управления триггерами в счетчике с параллельным переносом: триггер должен находиться в счетном режиме, только если выходы всех предыдущих триггеров равны логической единице; во все остальные интервалы времени триггер должен находиться в режиме хранения, не реагируя на синхросигнал.

Для реализации возможности переключения между счетным режимом и режимом хранения при построении счетчика удобно использовать JK-триггеры (см. табл. 3.2). Схема трехразрядного суммирующего счетчика с параллельным переносом показана на рис. 3.5.

Первый триггер постоянно находится в счетном режиме и меняет свое состояние на противоположное по каждому срезу синхросигнала. Второй триггер будет находиться в счетном режиме

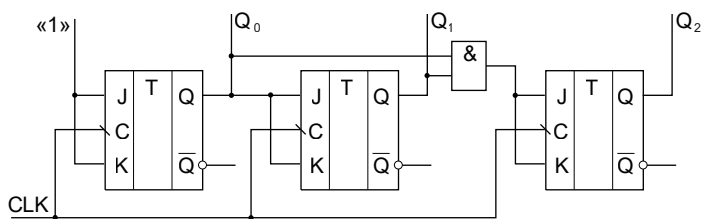


Рис. 3.5. Трехразрядный суммирующий счетчик с параллельным переносом на JK-триггерах

и менять свое состояние, только если в момент прихода среза синхросигнала на выходе первого триггера действует значение логической единицы. И так далее... Временные диаграммы идентичны диаграммам счетчика на рис. 3.4 за исключением того, что выходы всех трех триггеров в данном случае имеют одинаковую малую задержку относительно синхросигнала.

### Асинхронные входы установки и сброса триггера

Синхронный триггер с динамическим управлением может менять свое состояние только в моменты перехода тактового сигнала из низкого логического уровня в высокий или наоборот. На интервалах между такими активными фронтами тактового сигнала триггер никак не реагирует на изменения сигналов на его информационных входах.

Часто возникает необходимость осуществить установку или сброс триггера в произвольный момент времени вне зависимости от тактового сигнала, то есть *асинхронно*. Для этого в триггер добавляют специальные асинхронные входы, имеющие приоритет перед синхронными. Для асинхронных входов активным логическим уровнем, как правило, является низкий, что отображается с помощью инверсии на условном обозначении (рис. 3.6).

При подаче на оба асинхронных входа  $\bar{S}$  и  $\bar{R}$  высокого логического уровня, они никак не влияют на работу D-триггера, триггер в этом случае функционирует как триггер, показанный на рис. 3.2, в. Если же, к примеру, на вход асинхронной установки  $\bar{S}$  подать низкий логический уровень, триггер сразу будет установлен в единичное состояние независимо от сигналов на входах данных и синхросигнала. И в установленном состоянии триггер будет находиться как минимум до момента снятия низкого логического уровня со входа  $\bar{S}$ . Аналогично, триггер сбросится (установится в состояние логического «0»), если на вход асинхронного сброса  $\bar{R}$  подать

низкий логический уровень. Нужно отметить, что одновременная подача активных логических уровней на оба асинхронных входа недопустима и является запрещенной комбинацией, как и у RS-триггера.

Триггеры с асинхронными входами позволяют синтезировать счетчики импульсов с требуемым коэффициентом пересчета за счет возможности сброса триггеров в любой момент времени.

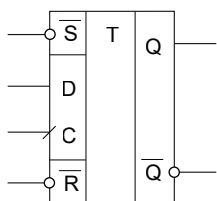


Рис. 3.6. D-триггер с асинхронными входами



### 3.3. Синтез двоичного счетчика с требуемым модулем счета

Рассмотрим пример синтеза суммирующего счетчика, принимающего состояния от  $0_{16}$  до  $B_{16}$  (от  $0_{10}$  до  $11_{10}$ ). Модуль счета такого счетчика  $M = 12$  по определению.

На первом шаге синтеза необходимо определить требуемое количество триггеров  $N$ :

$$N = \lceil \log_2 M \rceil, \quad (3.1)$$

где скобками показана операция округления вверх до целого. При  $M = 12$  логарифм по основанию 2 будет находиться в интервале от 3 до 4 (так как  $2^3 < M < 2^4$ ), соответственно, для реализации счетчика потребуется 4 триггера.

По умолчанию такой счетчик будет считать от  $0000_2$  до  $1111_2$ , поэтому потребуется дополнительная схема сброса триггеров, которая не будет позволять счетчику принимать состояния, большие, чем максимальное допустимое состояние  $11_{10} = 1011_2$ .

Составим таблицу состояний счетчика, в которой обозначим направление счета, разрешенные и запрещенные состояния (табл. 3.3).

Таблица 3.3

Таблица состояний счетчика от  $0_{16}$  до  $B_{16}$

Направление счета	Двоичный код				Шестнадцатиричный код	Примечание
	$Q_3$	$Q_2$	$Q_1$	$Q_0$		
↓	0	0	0	0	0	Разрешенные состояния
	0	0	0	1	1	
	0	0	1	0	2	
	0	0	1	1	3	
	0	1	0	0	4	
	0	1	0	1	5	
	0	1	1	0	6	
	0	1	1	1	7	
	1	0	0	0	8	
	1	0	0	1	9	
	1	0	1	0	A	
	1	0	1	1	B	
×	1	1	0	0	C	Запрещенные состояния
	1	1	0	1	D	
	1	1	1	0	E	
	1	1	1	1	F	

Счетчик будет последовательно принимать состояния от  $0_{16}$  до  $B_{16}$ , и как только он перейдет в первое запрещенное состояние  $C_{16} = 1100_2$ , необходимо сразу же асинхронно сбросить все четыре триггера. В результате счетчик сразу перейдет в состояние  $0000_2$  и начнет работу заново.

Для реализации такого поведения необходимо разработать комбинационную схему, управляющую сбросом триггеров – *дешифратор первого запрещенного состояния*. Такой дешифратор в данном примере будет описываться выражением:

$$\overline{R} = \overline{Q_3 \cdot Q_2 \cdot Q_1 \cdot Q_0}. \quad (3.2)$$

Очевидно, что выражение (3.2) будет равно логической единице для всех состояний счетчика, кроме первого запрещенного.

Рабочая схема синтезированного счетчика показана на рис. 3.7.

Для удобного подсоединения разрядов двоичного кода к разным частям схемы в Multisim использована четырехразрядная шина, изображенная на схеме жирной линией. Для вставки на схему новой шины необходимо нажать на клавиатуре комбинацию <Ctrl + U>, после чего курсор изменит свой вид, и можно, кликая по экрану, проводить шину. Завершается создание шины двойным кликом по последней точке. При подключении проводников к шине необходимо следить за соответствием нужной жиле шины.

Также можно отметить, что благодаря наличию у триггеров асинхронных входов не только сброса, но и установки, можно при наступлении первой запрещенной комбинации переводить счетчик не только в нулевое состояние, но и в любое другое, создавая тем самым счетчики, считающие не с нуля.

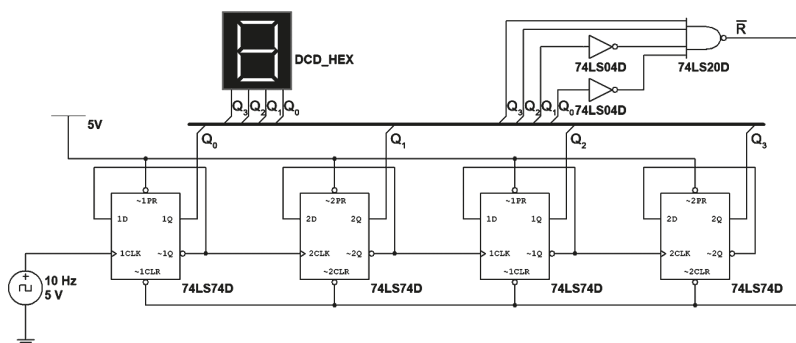


Рис. 3.7. Рабочая схема суммирующего счетчика от  $0_{16}$  до  $B_{16}$

### 3.4. Выполнение лабораторной работы

#### Задание на лабораторную работу

В рамках лабораторной работы необходимо выполнить синтез счетчика импульсов в соответствии с полученным вариантом задания.

#### Варианты заданий

Варианты заданий приведены в табл. 3.4, пояснения по заданиям даны после таблицы.

При синтезе вычитающего счетчика следует иметь в виду, что при использовании последовательной схемы соединения триггеров

Таблица 3.4

Варианты заданий

Вариант	Описание счетчика
1	от $0_{10}$ до $99_{10}$ (двоично-десятичный)
2	от $0_{10}$ до $59_{10}$ (двоично-десятичный)
3	от $0_{10}$ до $55_{10}$ (двоично-десятичный)
4	от $0_{10}$ до $34_{10}$ (двоично-десятичный)
5	от $79_{10}$ до $0_{10}$ (двоично-десятичный вычитающий)
6	от $59_{10}$ до $0_{10}$ (двоично-десятичный вычитающий)
7	от $55_{10}$ до $0_{10}$ (двоично-десятичный вычитающий)
8	от $34_{10}$ до $0_{10}$ (двоично-десятичный вычитающий)
9	от $1_{10}$ до $99_{10}$ (двоично-десятичный)
10	от $1_{10}$ до $30_{10}$ (двоично-десятичный)
11	от $10_{10}$ до $59_{10}$ (двоично-десятичный)
12	от $40_{10}$ до $59_{10}$ (двоично-десятичный)
13	от $0_{16}$ до $2A_{16}$ (двоичный)
14	от $0_{16}$ до $55_{16}$ (двоичный)
15	от $0_{16}$ до $5F_{16}$ (двоичный)
16	от $2A_{16}$ до $0_{16}$ (двоичный вычитающий)
17	от $55_{16}$ до $0_{16}$ (двоичный вычитающий)
18	от $5F_{16}$ до $0_{16}$ (двоичный вычитающий)
19	от $1_{16}$ до $2A_{16}$ (двоичный)
20	от $1_{16}$ до $55_{16}$ (двоичный)
21	от $1_{16}$ до $5F_{16}$ (двоичный)

возможны ложные срабатывания дешифратора запрещенного состояния из-за одновременного переключения состояний триггеров. Поэтому для вычитающих счетчиков целесообразно использовать схему счетчика с параллельным переносом либо предусматривать защиту от ложных срабатываний дешифратора.

При синтезе двоично-десятичного счетчика работа делится на две части: синтез счетчика единиц и синтез счетчика десятков (так называемые *декадные счетчики*). Например, для счетчика от  $0_{10}$  до  $79_{10}$  счетчик единиц будет иметь коэффициент пересчета десять, а счетчик десятков – восемь. Для отображения текущего состояния каждого разряда десятичного числа используется отдельный семисегментный индикатор. Вся схема должна работать от одного генератора тактовых импульсов.

При синтезе двоичного счетчика отображение его текущего состояния осуществляется в шестнадцатеричном коде с помощью двух семисегментных индикаторов.

### **Порядок выполнения работы**

Изучить методические указания.

Составить таблицы состояний счетчика для своего варианта задания (см. табл. 3.3).

Записать аналитические выражения дешифраторов запрещенных состояний.

Смоделировать в программе Multisim рабочую цифровую схему на D-триггерах 74LS74 либо JK-триггерах 74LS76 и других ТТЛ-элементах серии 74LS.

Проверить корректность работы схемы.

Для каждого вида использованных в схеме ТТЛ-элементов подобрать по справочнику аналог среди отечественных ИМС серии К5555.

Сделать выводы по проделанной работе.

### **Содержание отчета**

1. Цель работы.
2. Вариант задания.
3. Таблицы состояний, аналитические выражения дешифраторов запрещенных состояний.
4. Результат синтеза рабочей цифровой схемы в программе Multisim.
5. Схема электрическая принципиальная синтезированного устройства, выполненная в соответствии с ГОСТ.

6. Отечественные аналоги ИМС в виде таблицы со столбцами:
  - наименование ИМС 74LS,
  - наименование ИМС К555,
  - функциональное назначение,
  - количество цифровых элементов в схеме,
  - количество цифровых элементов в одной ИМС,
  - требуемое количество ИМС.
7. Спецификация на выбранные отечественные микросхемы.
8. Выводы по работе.

### **Контрольные вопросы**

1. В чем состоит принципиальное отличие последовательностных устройств от комбинационных.
2. Приведите примеры последовательностных устройств.
3. Перечислите основные типы триггеров.
4. Какие существуют способы тактирования триггеров.
5. В чем сходство и различие функционала RS-триггера и JK-триггера.
6. Как с помощью JK-триггера получить функционал T-триггера.
7. Входы и выходы типового D-триггера.
8. Входы и выходы типового T-триггера.
9. Что такое асинхронные входы синхронного триггера.
10. Приведите примеры классификаций счетчиков импульсов.
11. Как связан модуль счета счетчика и необходимое для построения такого счетчика количество триггеров.
12. В чем отличие двоично-десятичного счетчика от двоичного.
13. Имея два параллельных счетчика, возможно ли соединить их последовательно для увеличения модуля счета.
14. Имея два параллельных счетчика, как соединить их для увеличения модуля счета, чтобы все разряды получившегося счетчика переключались синхронно.
15. Как реализуются счетчики с модулем счета, не равным степени двойки.
16. Что такое реверсивный счетчик импульсов.
17. Какие типы триггеров могут быть использованы в качестве основы для построения счетчиков.
18. Какие типы триггеров могут быть использованы в качестве основы для построения регистров.
19. Приведите классификацию регистров по способу ввода-вывода информации.
20. Что такое реверсивный сдвиговый регистр.

## СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Схемотехника электронных систем. Цифровые устройства / Бойко В. И., Гуржий А. Н., Жуйков В. Я. и др. СПб.: БХВ-Петербург, 2004. 512 с.
2. *Угрюмов Е. П.* Цифровая схемотехника. 2 изд. перераб. и доп. СПб.: БХВ-Петербург, 2007. 782 с.
3. *Бирюков С. А.* Применение цифровых микросхем серий ТТЛ и КМОП. М.: ДМК Пресс, 2000. 240 с.
4. Цифровые интегральные микросхемы / Богданович М. И., Грель И. Н., Прохоренко В. А. и др. Минск: Беларусь, 1991. 493 с.
5. *Точки Р. Дж., Уидмер Н. С.* Цифровые системы. Теория и практика. М.: Вильямс, 2004. 1024 с.

## СОДЕРЖАНИЕ

Перечень сокращений и обозначений .....	3
1. Лабораторная работа № 1. «Исследование логических элементов» .....	4
1.1. Аксиомы булевой алгебры. Понятие логического элемента .....	4
1.2. Основные характеристики логических элементов .....	7
1.3. Выполнение работы.....	8
2. Лабораторная работа № 2. «Исследование комбинационных устройств» .....	17
2.1. Общие сведения о комбинационных устройствах .....	17
2.2. Синтез дешифратора для управления сегментом индикатора .....	26
2.3. Выполнение лабораторной работы .....	29
3. Лабораторная работа № 3. «Исследование последовательностных устройств» .....	34
3.1. Триггер как элементарное последовательностное устройство .....	34
3.2. Счетчики .....	37
3.3. Синтез двоичного счетчика с требуемым модулем счета.....	41
3.4. Выполнение лабораторной работы .....	43
Список рекомендуемой литературы.....	46