



ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
УПРАВЛЕНИЕ ДИСТАНЦИОННОГО ОБУЧЕНИЯ И ПОВЫШЕНИЯ  
КВАЛИФИКАЦИИ

Кафедра «Программное обеспечение вычислительной тех-  
ники и автоматизированных систем»

## **Учебно-методическое пособие** по дисциплине

# **«ОСНОВЫ МИКРОПРОЦЕССОРНОЙ ТЕХНИКИ»**

Автор  
Кузин А. П.

Ростов-на-Дону, 2019



## Аннотация

Учебно-методическое пособие предназначено для студентов очной формы обучения направления 09.03.04 «Программная инженерия».

## Автор

Старший преподаватель  
Кафедры ПОВТиАС  
Кузин А.П.



## Оглавление

<b>1. Лабораторная работа №1: Знакомство со средой «NI Multisim» .....</b>	<b>6</b>
1.1. Описание окна программы Multisim.....	6
1.2. Редактор схем.....	9
1.3. Создание новой схемы.....	15
1.3.1. Размещение компонентов на странице схемы...1	<b>Ошибка! Закладка не определена.</b>
1.3.2. Соединение компонентов проводниками .....	23
1.4. Режим моделирования схемы .....	<b>2Ошибка! Закладка не определена.</b>
1.5. Задание к лабораторной работе.....	<b>2Ошибка! Закладка не определена.</b>
<b>2. Лабораторная работа №2: Исследование логических схем .....</b>	<b>3Ошибка! Закладка не определена.</b>
2.1. Цель работы.....	34
2.2. Краткие сведения из теории.....	34
2.2.1. Аксиомы алгебры логики.....	34
2.2.2. Логические выражения.....	35
2.2.3. Законы булевой алгебры.....	35
2.2.4. Логические функции.....	36
2.2.5. Логические схемы.....	36
2.2.6. Таблица истинности.....	36
2.3. Последовательность выполнения работы.....	37
2.3.1. Исследование логической функции "И".....	37
2.3.2. Исследование логической функции "И-НЕ".....	38
2.3.3. Исследование логической функции "ИЛИ".....	39
2.3.4. Исследование логической функции "ИЛИ-НЕ".	40
2.3.5. Исследование логической функции "Исклю- чающее ИЛИ".....	41
2.4. Индивидуальное задание.....	42
2.5. Содержание отчета.....	44
<b>3. Лабораторная работа №3: Исследование арифметического сумматора с использованием программного комплекса Multisim4</b>	<b>Ошибка! Закладка не определена.</b>

Основы микропроцессорной техники

3.1.	Цель работы .....	44
3.2.	Краткие сведения из теории .....	44
3.3.	Последовательность выполнения работы .....	47
3.4.	Описание используемых контрольно-измерительных приборов .....	48
3.5.	Индивидуальное задание.....	51
3.6.	Содержание отчета.....	52
<b>4.</b>	<b>Лабораторная работа №4: Исследование триггеров в Multisim.....</b>	<b>52</b>
4.1.	Цель работы .....	52
4.2.	RS-Триггеры .....	52
4.2.1.	Основные теоретические положения.....	52
4.2.2.	Синхронные триггеры.....	55
4.2.3.	Задание 1.....	55
4.3.	JK-триггеры .....	<b>50</b>
	<b>Ошибка! Закладка не определена.</b>	
4.3.1.	Основные теоретические положения.....	56
4.3.2.	Задание 2.....	58
4.4.	D-триггеры .....	59
4.4.1.	Основные теоретические положения.....	59
4.4.2.	Задание 3.....	65
<b>5.</b>	<b>Лабораторная работа №5: Исследование регистров, счетчиков и дешифраторов с использованием программного комплекса Multisim .....</b>	<b>66</b>
5.1.	Цель работы .....	66
5.2.	Краткие сведения из теории.....	66
5.2.1.	Регистры.....	66
5.2.2.	Счетчики.....	68
5.2.3.	Дешифраторы.....	69
5.3.	Последовательность выполнения работы .....	71
5.3.1.	Исследование регистра.....	71
5.4.	Описание используемых контрольно - измерительных приборов .....	72
5.4.1.	Функциональный генератор (Function Generator).....	72
5.4.2.	Логический анализатор (Logic Analyzer).....	73
5.4.3.	Исследование суммирующего счетчика.....	74
5.4.4.	Исследование вычитающего счетчика.....	75
5.4.5.	Исследование дешифратора.....	76

Основы микропроцессорной техники

5.5.	Задания для самостоятельного выполнения.....	70
5.6.	Содержание отчета.....	78
<b>6.</b>	<b>Лабораторная работа №6: Мультиплексоры и демультиплексоры .....</b>	<b>78</b>
6.1.	Мультиплексоры .....	78
6.1.1	Краткие сведения из теории .....	78
6.2.	Демультиплексоры .....	85
6.2.1.	Основные теоретические положения ....	80
	<b>Закладка не определена.</b>	
<b>7.</b>	<b>Лабораторная работа №7: Создание приложение для моделирования работы микропроцессора с заданными параметрами.....</b>	<b>92</b>
7.1.	Выполнение работы.....	92
7.2.	Здания для самостоятельного выполнения.....	92
	<b>Список литературы .....</b>	<b>99</b>

## 1. ЛАБОРАТОРНАЯ РАБОТА №1: ЗНАКОМСТВО СО СРЕДОЙ «NI MULTISIM»

### 1.1. Описание окна программы Multisim

Разработка любого радиоэлектронного устройства включает физическое или математическое моделирование. Физическое моделирование связано с большими материальными затратами, так как требует изготовления макетов и их трудоемкое исследование. Иногда чисто физическое моделирование просто невозможно из-за сложности устройства. В этом случае прибегают к математическому моделированию с использованием средств и методов вычислительной техники.

Наиболее простой и легко осваиваемой программой, содержащей блок логического моделирования цифровых устройств, является программа Electronics Workbench (EWB) канадской компании Interactive Image Technologies. Особенность программы – наличие в ней контрольно-измерительных приборов, по внешнему виду, органам управления и характеристикам максимально приближенных к их промышленным аналогам. Опыт использования программы в лабораторном практикуме по ряду предметов показывает, что для проведения лабораторных работ достаточно двух часов предварительного ознакомления с программой. Ниже приведено краткое описание программы.

Окно программы Multisim 2001 (версия 6.20) (рис. 1.1) содержит шкалу меню (вторая строка окна): *File, Edit, View, Place, Simulate, Transfer, Tools, Options, Help*. Далее описаны некоторые из них.

Меню *Simulate* содержит команды:

*Run* – запуск моделирования;

*Pause* – временный останов моделирования;

*Default Instrument Setting* – установка по умолчанию режима работы контрольно-измерительных приборов;

*Default Simulation Setting* – выбор идеального (при использовании компонентов типа Virtual) или реального режимов работы моделирования;

*Instruments* – список контрольно-измерительных приборов (линейка этих приборов расположена слева в окне);

*Analyses* – список команд моделирования.

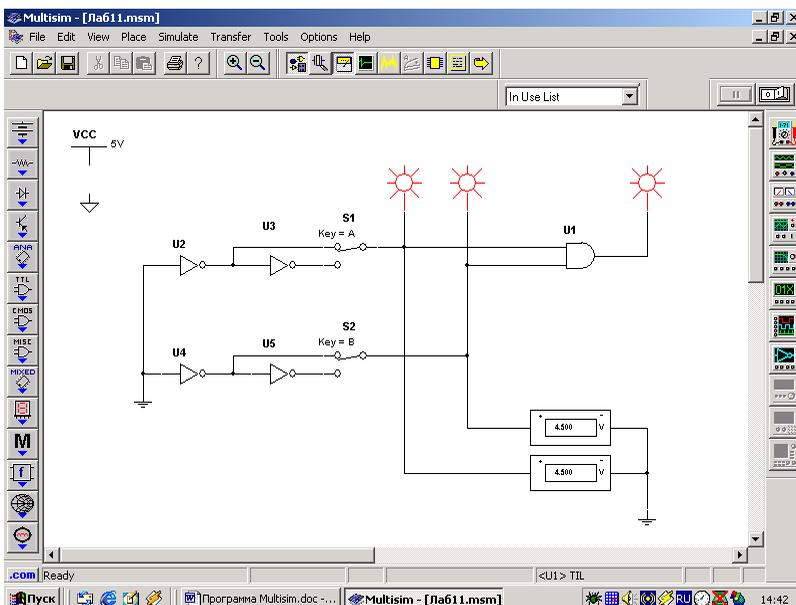


Рис. 1.1. Окно схемного редактора программы Multisim

Меню *Options* – (настройка программы), содержит команды:  
*Preferences* – пользовательские настройки (назначение команд описаны ниже);

*Modify Title Block* – внесение данных в разделы штампа;

*Global Restrictions* – установка общего пароля;

*Circuit Restrictions* – установка атрибутов только *Read-only*.

Третья строка окна содержит:

1) восемь мнемонических кнопок общесистемного характера (*System*);

2) две кнопки, с помощью которых можно увеличить или уменьшить масштаб изображения (*Zoom*);

3) девять специальных кнопок, дублирующих наиболее часто используемые команды (*Design*);

4) выпадающий список использованных в текущей схеме компонентов (*In Use List*);

5) кнопку, временно приостанавливающую процесс моделирования;

6) переключатель, запускающий и останавливающий процесс моделирования.

Слева в окне расположена вертикальная панель библиотек компонентов (базы данных *Component Bars*), справа – вертикаль-

ная панель контрольно-измерительных приборов (*Instruments*).

Кнопки можно сделать видимыми или невидимыми, если вызвать команду *View> Toolbars* и во всплывающем меню установить (или нет) соответствующие флажки: *System, Zoom, Design, In Use List*. Здесь и далее в данном тексте на первом месте указывается меню из строки меню окна редактора схем, на втором месте – команда из выпадающего меню, появляющегося при выборе кнопки конкретного меню.

Панель библиотек компонентов можно сделать видимой или невидимой, если вызвать команду *View> Component Bars* и во всплывающем меню установить или нет флажок *Multisim Database*.

Панель контрольно-измерительных приборов можно сделать видимой или невидимой, если вызвать команду *View> Toolbars* и во всплывающем меню установить или нет флажок *Instruments*.

Панель библиотек компонентов (базы данных *Component Bars*) содержит:

- источники (*Sources*);
- пассивные компоненты и коммутационные устройства (*Basic*);
- диоды (*Diodes*);
- транзисторы (*Transistors*);
- аналоговые микросхемы (*Analog*);
- цифровые микросхемы TTL серии (*TTL*);
- цифровые микросхемы КМОП серии (*CMOS*);
- одиночные цифровые схемы, АЛУ, регистры, счетчики, мультиплексоры, дешифраторы, ОЗУ и т.п. (*Misc Digital*);
- микросхемы смешанного типа (*Mixed*);
- индикаторные устройства (*Indicators*);
- компоненты смешанного типа (*Miscellaneous*);
- аналоговые вычислительные устройства (*Controls*);
- радиочастотные компоненты (*RF*);
- электромеханические элементы (*Electro\_Mechanical*).

Панель контрольно-измерительных приборов (*Instruments*) содержит:

- цифровой мультиметр (*Multimeter*);
- функциональный генератор (*Function Generator*);
- измеритель активной мощности и коэффициента мощности (*Wattmeter*);
- осциллограф (*Oscilloscope*);
- измеритель АЧХ и ФЧХ (*Bode Plotter*);

## Основы микропроцессорной техники

- генератор слова (*Word Generator*);
- логический анализатор (*Logic Analyzer*);
- логический преобразователь (*Logic Converter*);
- измеритель нелинейных искажений в диапазоне частот от 20 до 200000 Гц (*Distortion Analyzer*);
- спектральный анализатор (*Spectrum Analyzer*);
- прибор для анализа электрических цепей в обобщенном виде – в виде четырехполюсников, имеющих два входа и два выхода (четыре полюса) (*Network Analyzer*).

Последние три прибора недоступны.

### 1.2. Редактор схем

Окно редактора схем предназначено для создания и редактирования принципиальных схем устройства. По умолчанию цвет фона окна черный. Изменить установки по умолчанию можно, выбрав команду *Options>Preferences*. После выбора данной команды открывается диалоговое окно *Preferences* (рис. 1.2). Оно содержит шесть вкладок, на которых можно установить необходимые опции. При выводе этого окна активной является вкладка *Circuit*, на которой имеются две панели *Show* и *Color*.

Панель *Show* содержит окно просмотра, в котором можно увидеть установленные опции, и шесть флажков, с помощью которых устанавливаются необходимые опции.

С помощью флажка *Show component labels* выбирают видимость или невидимость позиционного обозначения компонента на схеме. Для того чтобы установить видимость позиционного обозначения компонента на схеме, нужно щелкнуть левой клавишей мыши по белому квадрату, расположенному слева от имени флажка. Если флажок выбран, то в белом квадрате появляется галочка.

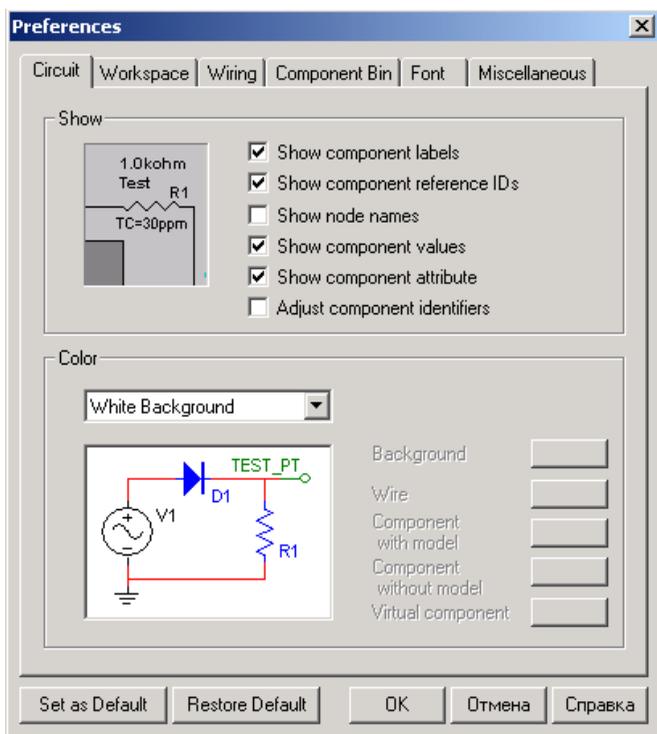


Рис. 1.2. Диалоговое окно *Preferences* с вкладкой *Circuit*

Следующими флажками устанавливают видимость на схеме:  
*Show component reference IDs* – порядкового номера компонента;

*Show node names* – имени узла;

*Show component values* – номинала компонента;

*Show component attribute* – таблицы свойств компонента.

Флажок *Adjust component identifiers* позволяет при упорядочивании схемы изменять идентификационные номера однотипных компонентов, присвоенных программой автоматически в порядке их установки.

Панель *Color* содержит выпадающий список с именами команд, устанавливающих цвет фона, надписей и компонентов, окно просмотра, в котором можно просмотреть выбранный режим установки цвета, и пять кнопок, окрашенных в различные цвета, с помощью которых можно установить цвет для фона (*Background*), проводника (*Wire*), компонента, имеющего математическую модель (*Component with model*), компонента, не имеющего математическую модель (*Component without model*), виртуального компонента (*Virtual component*).

тическую модель (*Component without model*), и виртуального компонента (*Virtual component*).

На вкладке *Workspace* (рис. 1.3) определяются настройки текущего окна.

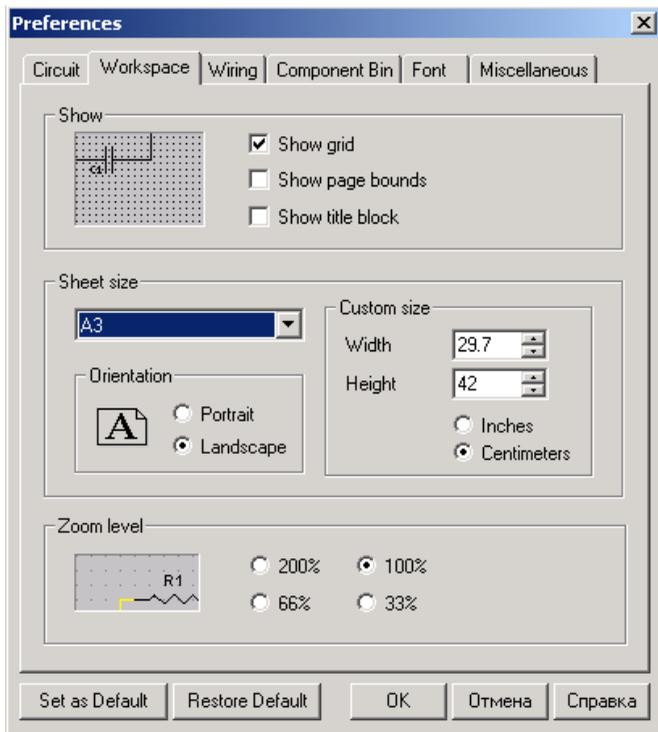


Рис. 1.3. Диалоговое окно *Preferences* с вкладкой *Workspase*

На панели *Show* в окне, расположенном слева, отображаются результаты установки опций с помощью флажков, находящихся справа. Самый верхний флажок — показывать (или нет) сетку (*Show grid*), второй сверху флажок — показывать или нет границы страницы (*Show page bounds*), третий флажок — показывать или нет штамп страницы (*Show title block*).

На панели *Sheet size* (размер страницы схемы) в выпадающем списке выбирают формат страницы (A3, A4...). На панели *Orientation* устанавливают ориентацию страницы: портрет или альбомная (*Landscape*); на панели *Custom size* — ширину и высоту страницы в дюймах или сантиметрах; на панели *Zoom level* — масштаб изображения.

На вкладке *Wiring* (рис. 1.4) устанавливают ширину проводников и степень автоматизации разводки проводников; на панели *Wire width (drawing option)* – ширину линии. Результат отображается в окне.

На панели *Autowire* устанавливается (или нет) автоматическое соединение проводников (флажок *Autowise on connection*) и автоматическое движение (флажок *Autowise on move*) проводников.

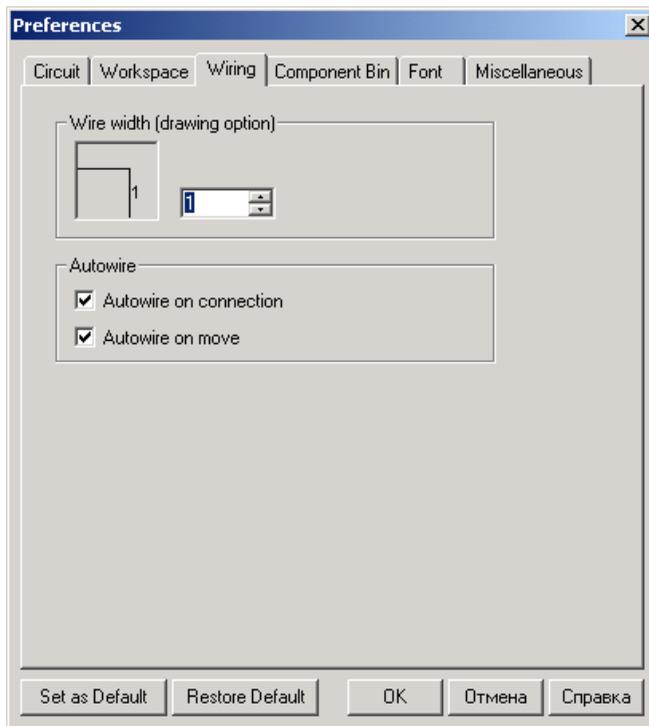


Рис. 1.4. Диалоговое окно *Preferences* с вкладкой *Wiring*

На вкладке *Component Bin* (рис. 1.2.4) на панели *Symbol standard* устанавливается стандарт изображения компонентов *ANCI* (США) или *DIN* (Европа).

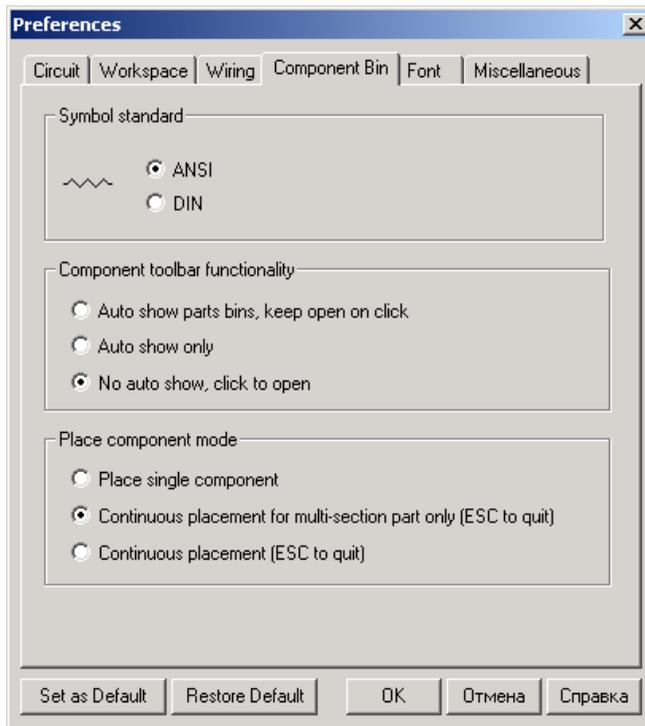


Рис. 1.5. Диалоговое окно *Preferences* с вкладкой *Component Bin*

На панели *Component toolbar functionality* указывают, изменять или нет цвет изображения компонента при его выборе.

На панели *Place component mode* выбирают режим размещения компонентов: единственный компонент (*Place single component*), непрерывное размещение многосекционного компонента (*Continuous placement for multi-section part only*), непрерывное размещение (*Continuous*).

На вкладке *Font* (шрифт) (рис. 1.2.5) выбирают шрифт и его атрибуты для выбранных компонентов схемы.

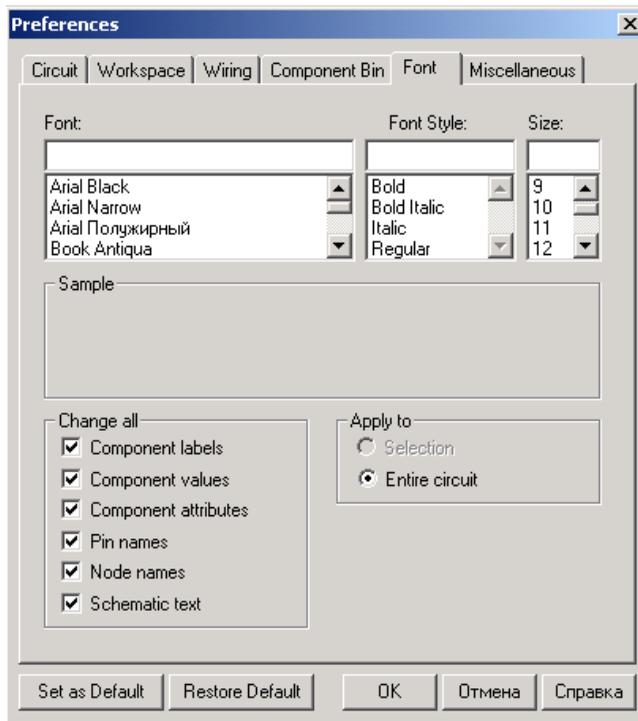


Рис. 1.6. Диалоговое окно *Preferences* с вкладкой *Font*

На вкладке *Miscellaneous* (разное) (рис. 1.2.6) устанавливаются режим автосохранения данных, пути к рабочему каталогу, идеальный или реальный режим моделирования (переключатели *Ideal* и *Real* на панели *Digital Simulation Setting*) цифровых схем, соединить или рассоединить аналоговое и цифровое заземление (флажок *Connect digital ground to analog ground* на панели *PCB Ground Option*).

Кроме того, окно *Preferences* содержит пять кнопок.

Для того чтобы записать выбранные опции для текущей схемы, нужно щелкнуть левой клавишей мыши по кнопке *OK*.

Для того чтобы записать выбранные опции, как устанавливаемые по умолчанию, нужно щелкнуть левой клавишей мыши по кнопке *Set as Default*, затем по кнопке *OK*.

Для того чтобы вернуться к предыдущим опциям, установленным по умолчанию, нужно щелкнуть по кнопке *Restore Default*. Затем можно продолжить устанавливать новые опции.

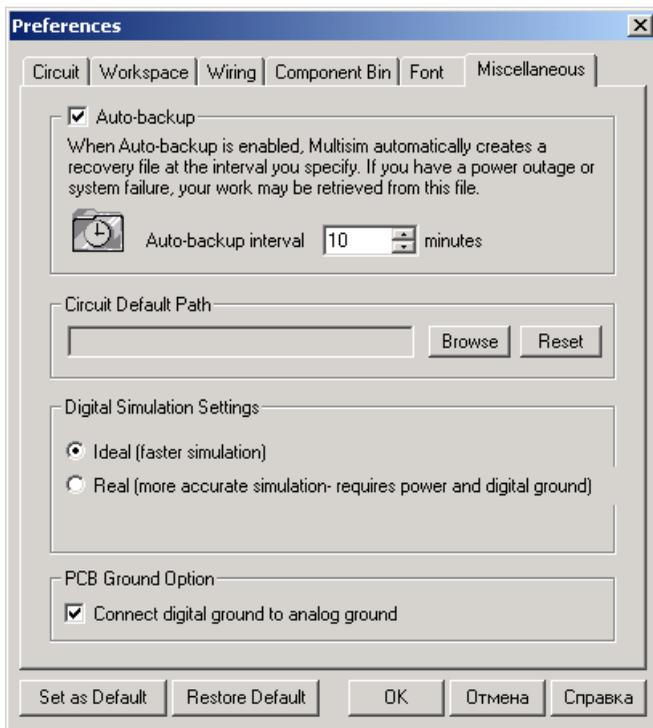


Рис. 1.7. Диалоговое окно *Preferences* с вкладкой *Miscellaneous*

Чтобы остановить изменения и закрыть диалоговое окно, нужно щелкнуть по кнопке *Cancel*.

### 1.3. Создание новой схемы

Создание новой схемы рассмотрим на примере схемы, исследующей логическую схему «И» (рис. 1.3.1).

Прежде чем начать создание схемы, нужно запустить программу *Multisim*. Для этого в окне *Windows* нажать на кнопку *Пуск*, в открывшемся меню выбрать команду *Программы*. В новом открывшемся меню выбрать строку *Multisim*. Автоматически откроется окно программы, в котором будет расположена страница новой схемы. На этой странице и будет создана схема исследования логической схемы «И».

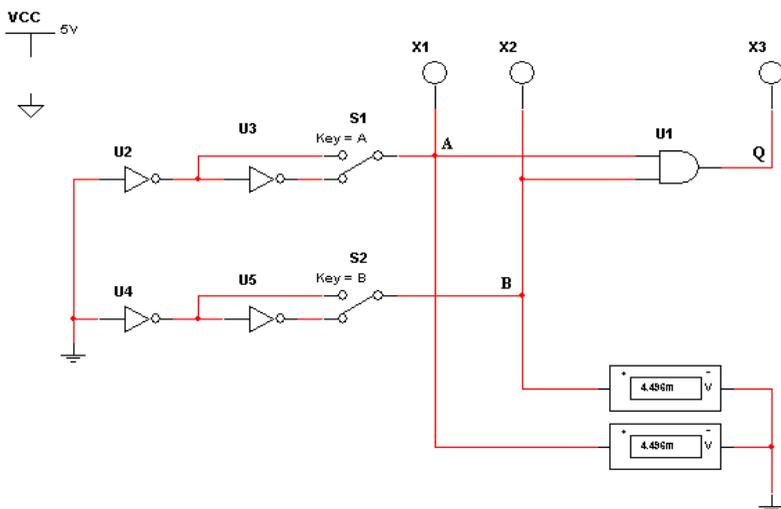


Рис. 1.8. Схема исследования логической схемы «И»

### 1.3.1. Размещение компонентов на странице схемы

Для размещения компонентов служат кнопки панели инструментов, расположенной вертикально на левой стороне окна. По умолчанию эта панель видима. Если она не видима, то надо нажать кнопку *Components* на строке инструментов *Design Bar*. Компоненты, необходимые для создания схемы, сгруппированы в логические разделы (*Parts Bin*). Каждому логическому разделу соответствует кнопка на шкале инструментов. При нажатии на одну из этих кнопок открывается соответствующая панель (*Parts Bin*), содержащая кнопки для каждого компонента, входящего в семейство компонентов. Разместить компоненты можно также с помощью команды *Place>Place Component*.

Разместите на странице схемы компонент источника постоянного напряжения +5 В. Для этого необходимо:

1. Нажать на кнопку *Sources*. Откроется панель, содержащая источники напряжения или тока (рис. 1.3.1.1). Удерживая курсор на кнопке, не щелкая по ней, можно увидеть название кнопки.

2. Щелкнуть по кнопке *Vcc*. Курсор примет форму выбранного компонента. Поместить выбранный компонент на схему и щелкнуть левой клавишей мыши по точке схемы, в которой должен быть расположен компонент, изображение компонента появится на схеме.

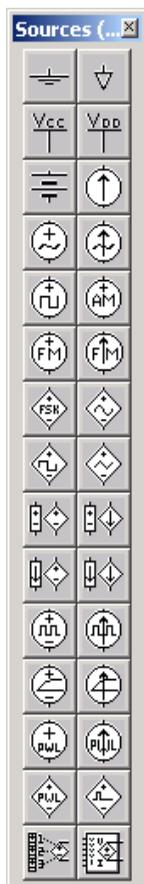


Рис. 1.9.  
Панель инстру-  
ментов *Sources*

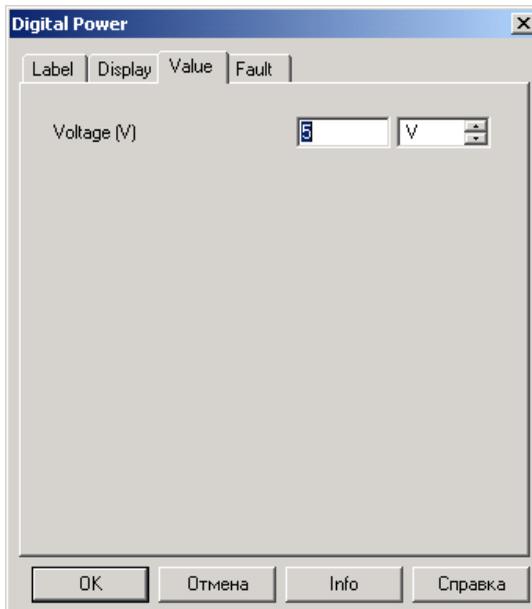


Рис. 1.10. Диалоговое окно *Digital Power*  
с вкладкой *Value*

3. Если необходимо изменить свойства компонента (порядковый номер, имя и величину значения), то дважды щелкнуть по изображению компонента. Откроется диалоговое окно *Digital Power* (рис. 1.10). Название окна зависит от назначения компонента. Это окно содержит четыре вкладки. По умолчанию открыта вкладка *Value*, на которой расположены, текстовая строка и выпадающий список.

По умолчанию значение напряжения источника равно 12 В. Напряжение питания микросхем серии 74ххXX равно 5 В. Поэтому



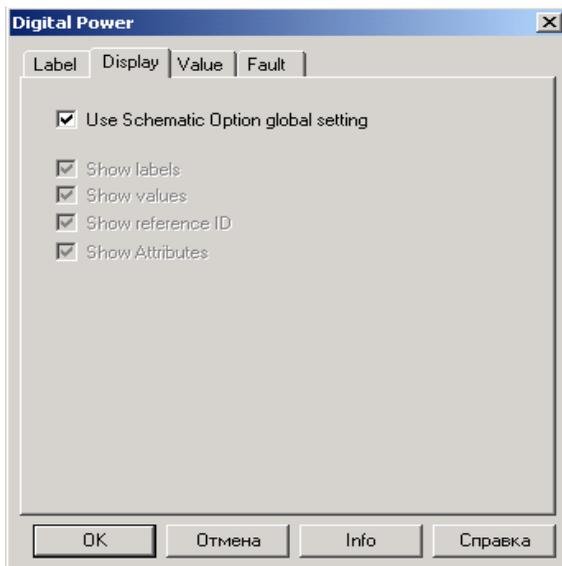


Рис. 1.12 Диалоговое окно *Digital Power* с вкладкой *Display*

На вкладке *Fault* (погрешность) (рис. 1.13) устанавливают границы погрешности.

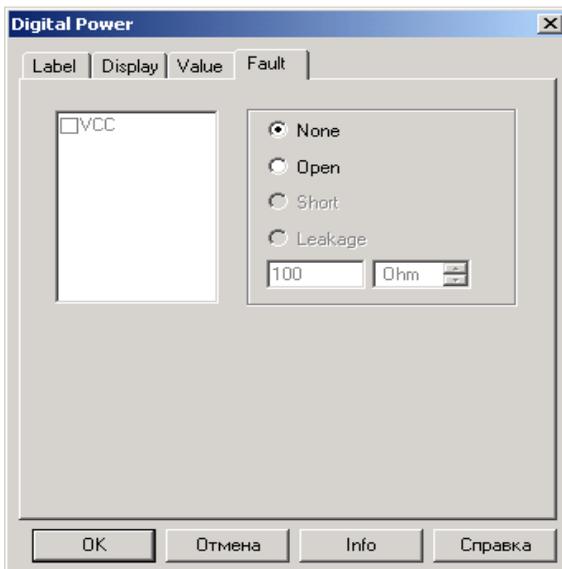


Рис. 1.13. Диалоговое окно *Digital Power* с вкладкой *Fault*

Чтобы разместить на схеме изображения четырех компонентов «НЕ» и одного компонента «И», необходимо:

1. Нажать кнопку *Misc Digital*. Откроется панель с соответствующими компонентами (рис. 1.14).

2. Нажать на кнопку с изображением логического элемента. Откроется диалоговое окно *Component Browser* (рис. 1.15), в котором имеется окно просмотра *Component Name List*, содержащее список цифровых компонентов. Во втором окне *Symbol* появляется изображение выбранного компонента в формате *ANSI* (США) или *DIN* (Европа).

3. В списке компонентов выбрать компонент *2И (AND2)* и поместить его на схему, щелкнув по кнопке *OK*. Ускорить поиск компонента можно, если ввести несколько первых символов имени компонента в строку *Component Name*.

4. В списке компонентов выбрать компонент *НЕ (NOT)* и расположить четыре образца этого компонента на схеме.



Рис.1.14. Панель инструментов

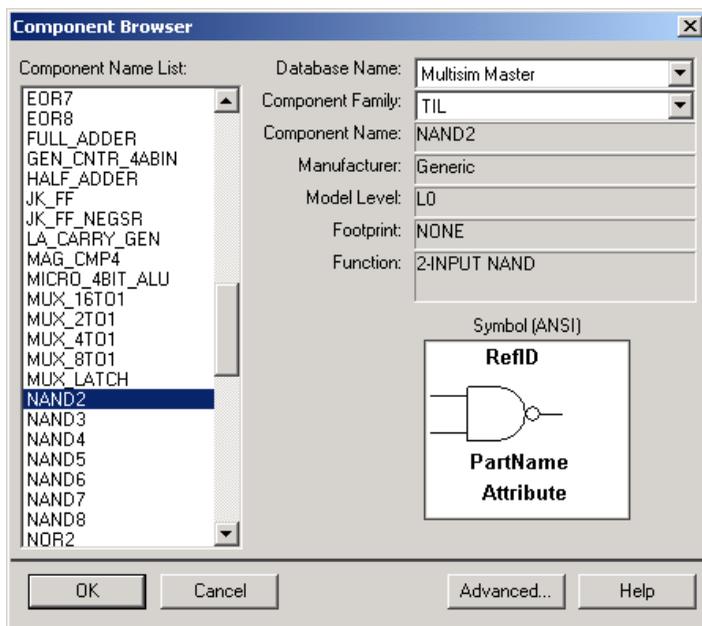


Рис. 1.15. Диалоговое окно *Component Browser*  
Поместить на схему несколько копий одного и того же ком-

понента можно двумя способами:

- повторить несколько раз действия, описанные в п. 3;
- воспользоваться списком компонентов, размещенных на схеме ранее. Для этого открыть список *In Use List*, в строке *System toolbar*, нажав на черный треугольник, расположенный в окне списка справа. Выбрать в этом списке нужный компонент и поместить его на схему.

Разместить на схеме два переключателя на два положения.

Для этого необходимо:

1. Нажать кнопку *Electro*. Откроется панель с соответствующими компонентами (рис. 1.16).

2. Выбрать кнопку *SUPPLEMENTARY\_CONTACTS*. Откроется диалоговое окно *Component Browser* (рис. 1.17).

3. В окне просмотра (рис. 1.17) *Component Name List* выбрать компонент *SPDT\_SB* и поместить два образца этого компонента на схему (как описано выше).



Рис. 1.16. Панель инструментов *Electro*

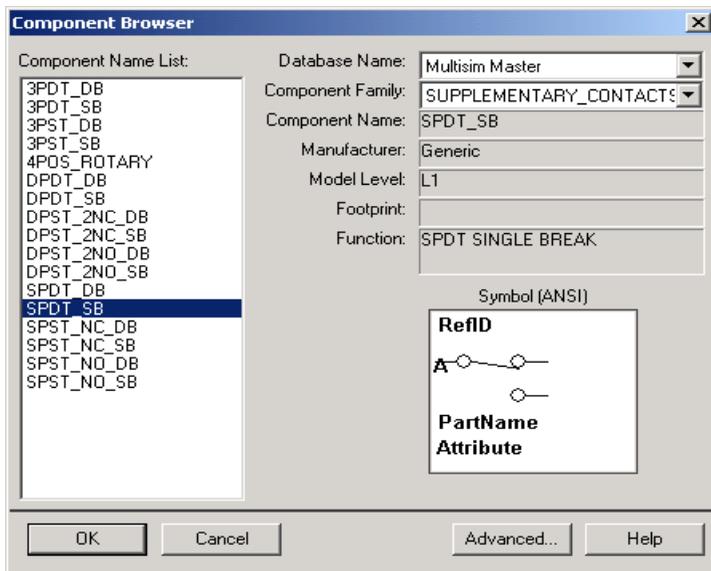


Рис. 1.17. Диалоговое окно *Component Browser*

4. При необходимости повернуть эти образцы компонента на нужный угол по горизонтали или по вертикали, или на  $90^\circ$  направо или влево.

Чтобы сделать это, необходимо:

1) щелкнуть правой клавишей мыши по компоненту. Откроется всплывающее меню (рис. 1.18);

2) в этом меню выбрать нужную строчку и щелкнуть левой клавишей мыши по этой строке.

Разместить на схеме три светодиода. Для этого необходимо:

1. Нажать кнопку *Indicator*. Откроется панель инструментов *Indicator* (рис. 1.19).

2. Щелкнуть по кнопке с изображением светодиода и поместить три образца этого компонента на схему.

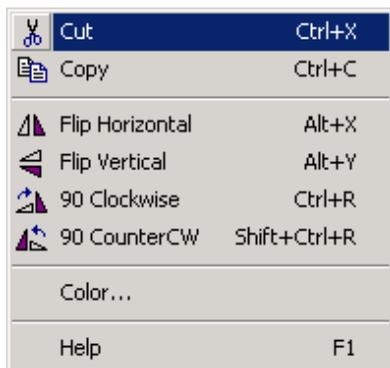


Рис. 1.18. Всплывающее меню



Рис. 1.19. Панель инструментов *Indicator*

3. При необходимости повернуть эти образцы компонента на нужный угол (как описано выше).

Схема, содержащая цифровые компоненты, должна включать источник напряжения *VCC* и символ «цифровая земля» (*Digital ground*), которые *Multisim* использует для подачи питания на цифровые компоненты.

Разместить на схеме изображение символ «земля». Для этого необходимо:

1. Нажать на кнопку *Sources*. Откроется панель, содержащая источники напряжения или тока (рис. 1.9) и символы «земля».

2. Поместить на схему символы «земля» и «цифровая земля», нажав кнопки *Ground* и *Digital\_Gnd* соответственно.

Разместить на схеме изображение вольтметра. Для этого необходимо:

1. Нажать кнопку *Indicator*. Откроется панель инструментов

Indicator (рис. 1.19).

2. Поместить на схему вольтметр, нажав на кнопку *Voltmeter*.

В результате страница схемы будет выглядеть, как показано на рис. 1.20.

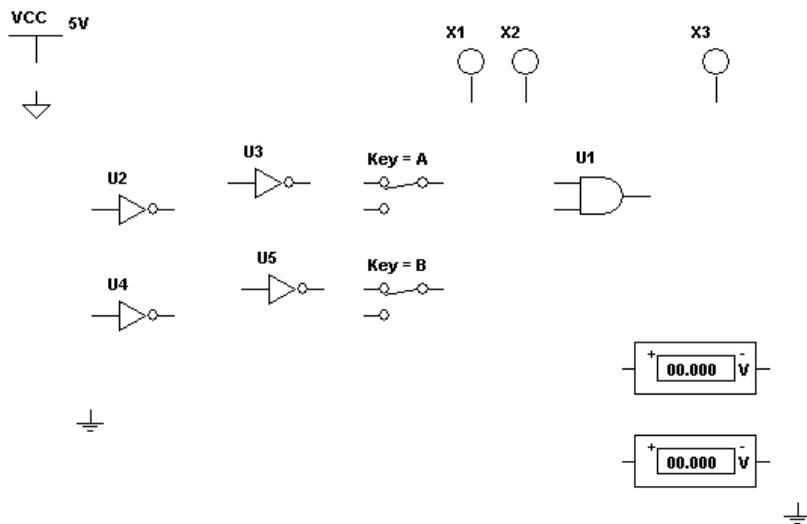


Рис. 1.20. Страница схемы с размещенными на ней компонентами

Используя команду *File> Save*, записать страницу схемы с размещенными на ней компонентами.

### 1.3.2. Соединение компонентов проводниками

После того как компоненты размещены на схеме, их необходимо соединить проводниками. Все компоненты имеют выводы, которые используются для этих целей. Соединить компоненты проводниками можно двумя способами: автоматически и вручную. При автоматическом соединении программа выбирает наилучший путь между двумя соединяемыми выводами компонентов, т.е. программа избегает размещения проводников через другие компоненты или частичного перекрытия проводников. При ручном соединении компонентов пользователь должен сам контролировать путь проводника на схеме. При разводке одного проводника можно комбинировать эти два метода, например, начать разводку проводника вручную и затем позволить программе закончить ее.

Для автоматического соединения двух компонентов необходимо:

1. Щелкнуть по выводу первого компонента, например «земля». Изображение курсора измениться на крест, показывая, что программа находится в режиме разводки.

2. Щелкнуть по первому выводу второго компонента логической схемы «HE» ( $U4$ ), к которому необходимо присоединить проводник. Соединение выводов первого и второго компонентов произойдет автоматически. Затем изображение курсора примет первоначальный вид и можно выполнять очередные команды.

3. Повторять эти шаги до тех пор, пока все выводы всех компонентов не будут соединены нужным образом.

Соединить первый вывод компонента  $U2$  и первый вывод компонента  $U1$ .

Чтобы удалить проводник, соединяющий два вывода, нужно щелкнуть по проводнику правой клавишей мыши и из всплывающего меню выбрать команду *Delete* или нажать клавишу *Delete*.

Для более точного определения пути проводника, соединяющего два вывода, используется ручная разводка. Программа предотвращает присоединение двух проводников к одному выводу, что исключает возможные ошибки соединения. Поэтому в этом случае начинать разводку проводника нужно не на выводах компонентов, а в какой-нибудь точке проводника, соединяющего эти выводы. Для этого нужно добавить точку соединения (*junction*) к данной точке, т.е. необходимо:

- 1) выбрать команду *Place>Place Junction*. В точке проводника, на которой остановлен курсор, появится легкое изображение точки;

- 2) передвинуть изображение точки в нужное место проводника и щелкнуть левой клавишей мыши. На проводнике появится точка.

Соединять выводы компонентов проводниками вручную удобнее, если в редакторе схемы видна сетка. Для того чтобы сделать сетку видимой, если она не видима, нужно щелкнуть правой клавишей мыши в какой-нибудь точке редактора схемы и из всплывающего меню выбрать команду *Show Grid*.

Для того чтобы выполнить соединение двух выводов вручную, необходимо:

1. Щелкнуть по точке соединения, только что размещенной на проводнике. Изображение курсора изменится на крест, показывая, что программа находится в режиме разводки проводников.

2. Перетащить курсор к выводу другого компонента и щелкнуть левой клавишей мыши. Это фиксирует проводник на его месте.

3. Если нужно изменить направление перемещения проводника, то щелкнуть левой клавишей мыши в точке изменения направления движения. Маленькие черные квадраты показывают точки изменения направления движения.

4. Если необходимо переместить проводник на другое место, нужно щелкнуть в любой точке проводника и передвинуть этот сегмент проводника.

### 1.4. Режим моделирования схемы

После создания принципиальной схемы исследуемого устройства необходимо запустить процесс моделирования, чтобы исследовать поведение устройства.

- Для того чтобы запустить процесс моделирования, нужно щелкнуть по кнопке *Simulate*  и из всплывающего меню выбрать команду *Run*. Слева от команды *Run* появится галочка, означающая, что процесс моделирования начался.

- Для того чтобы приостановить на время процесс моделирования, нужно щелкнуть по кнопке *Simulate*  и из всплывающего меню выбрать команду *Pause*. Для возобновления процесса моделирования нужно щелкнуть по кнопке *Simulate*  и из всплывающего меню снова выбрать команду *Pause*. Процесс моделирования возобновится с того момента, когда он был остановлен.

- Для того чтобы остановить процесс моделирования, следует щелкнуть по кнопке *Simulate*  и из всплывающего меню выбрать команду *Run*. Галочка слева от команды *Run* исчезнет, показывая, что процесс моделирования остановлен. Если после остановки процесса моделирования запустить его снова, то, в отличие от действия команды *Pause*, процесс моделирования начнется из начальной точки.

- Процесс моделирования можно запустить или приостановить с помощью команд *Simulation>Run* и *Simulation>Pause* соответственно.

- Процесс моделирования можно запускать и останавливать с помощью переключателя, который можно сделать видимым или невидимым с помощью команды *View>Show Simulation Switch*.

### 1.5. Задание к лабораторной работе

Для выполнения лабораторной работы №1 по Multisim следует самостоятельно создать схемы, приведенные ниже.

Основы микропроцессорной техники

Используйте ту же нумерацию компонентов и введите те же названия для узлов. Добавьте блок заголовка с вашим именем.

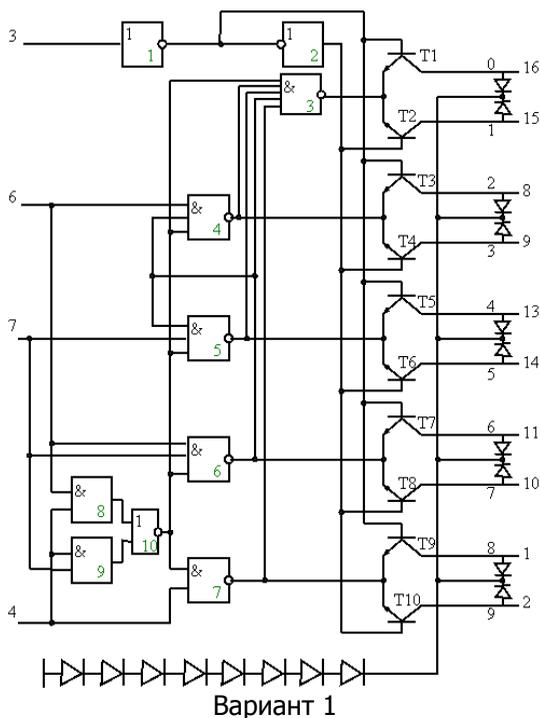
Вариант работы выбирается исходя из номера студента в списке группы

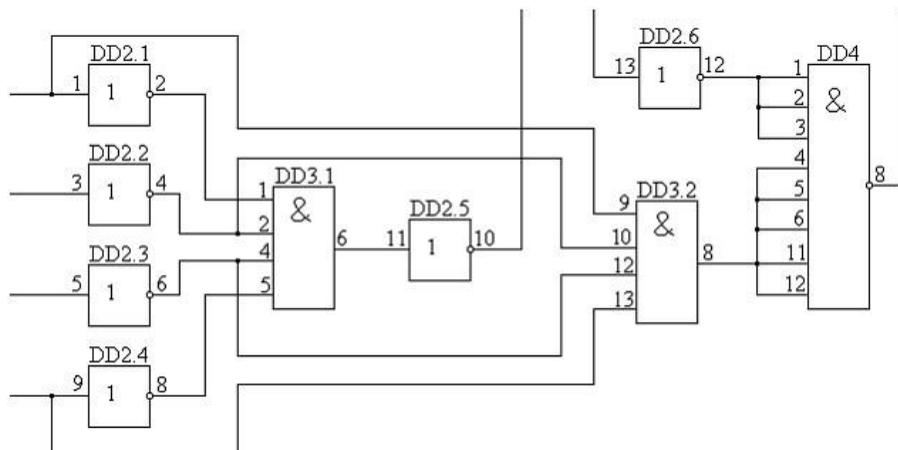
$YZ$  минус 16 умножить на  $K$ ,

где  $Y$  - предпоследняя цифра в номере списка;  $Z$  - последняя цифра в номере списка.

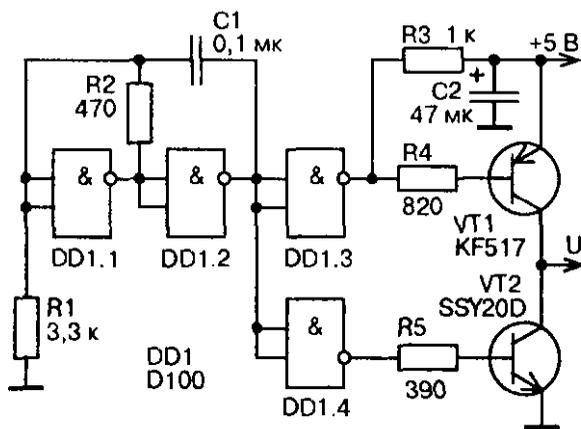
$K$  - 0 или 1. (выбирается из условия, чтобы номер варианта не был отрицательным)

Примечание: Ряд элементов схем изображен по Российской ГОСТу, который во многом может не совпадать с международным. В этом случае, если студенту неизвестно наименование элемента или его вид по международным стандартам, то необходимо самостоятельно с помощью Интернет определить его назначение и вид. Если такой элемент отсутствует в базе – выбрать другой, близкий по функционалу.

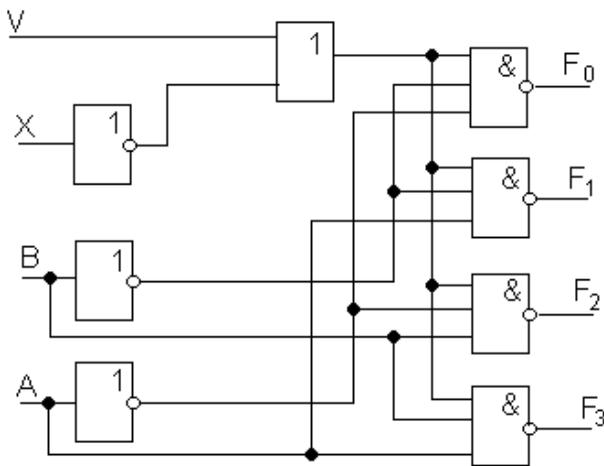




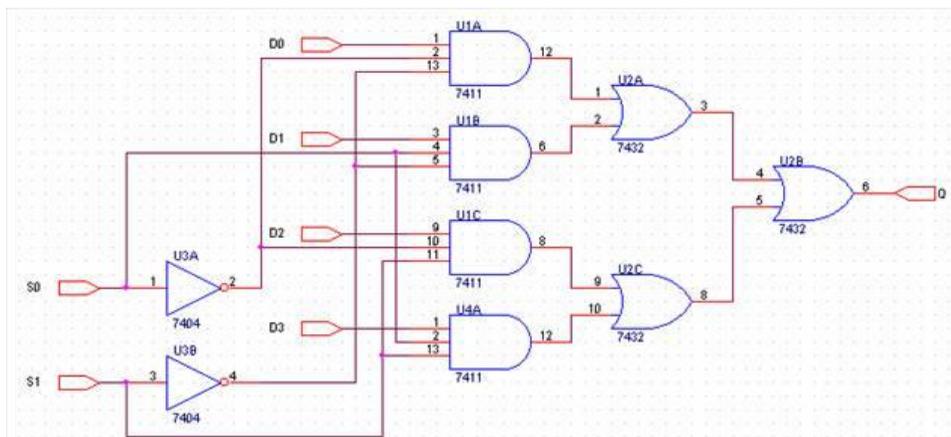
Вариант 2



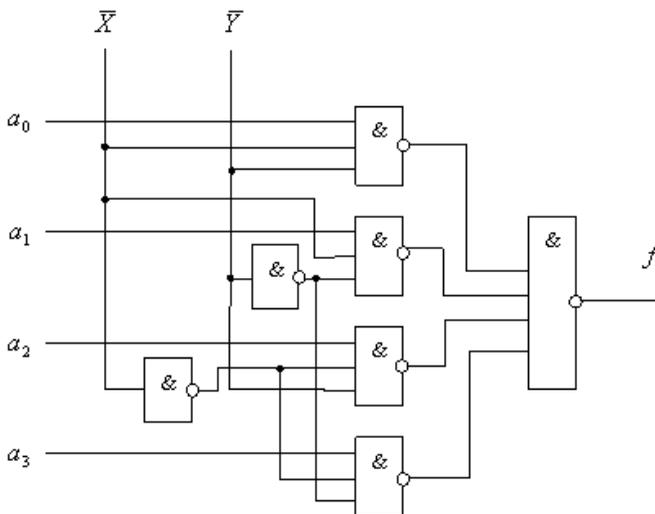
Вариант 3



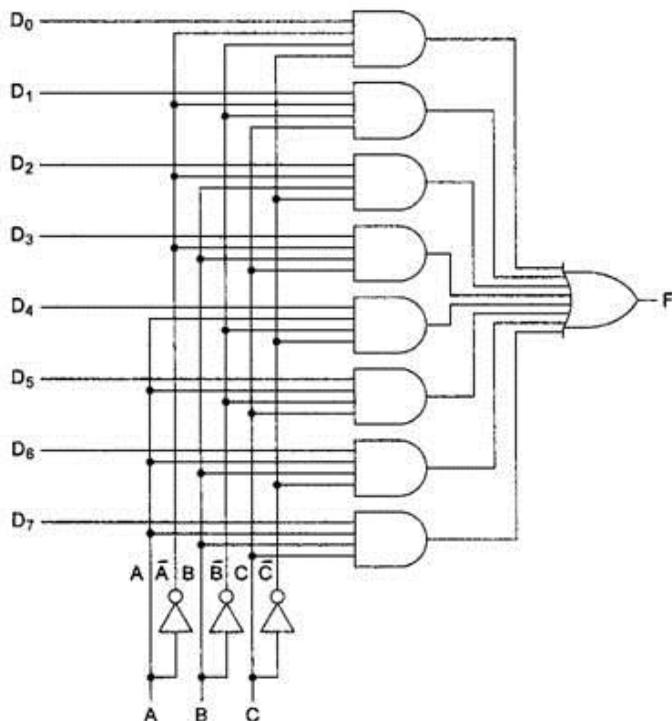
Вариант 4



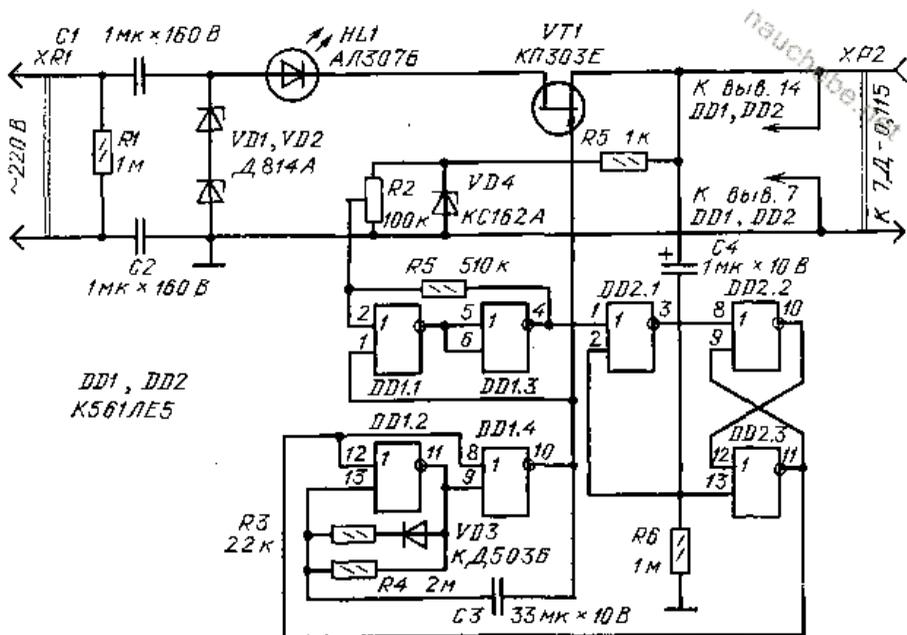
Вариант 5



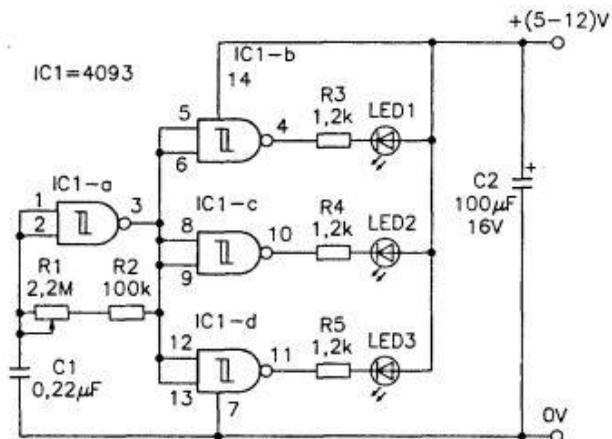
Вариант 6



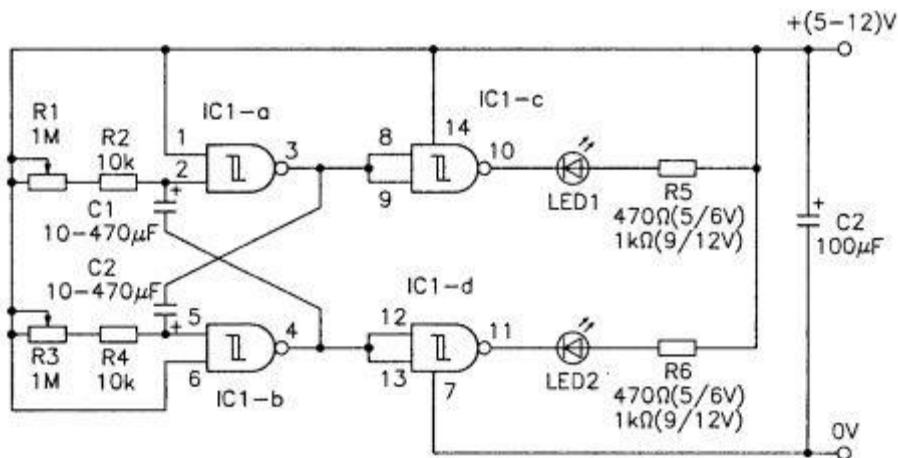
Вариант 7



Вариант 8



Вариант 9



Вариант 10

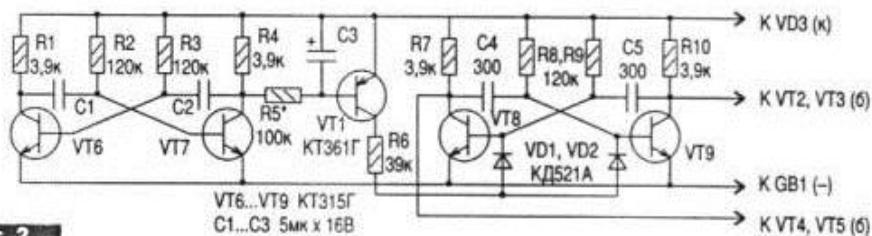
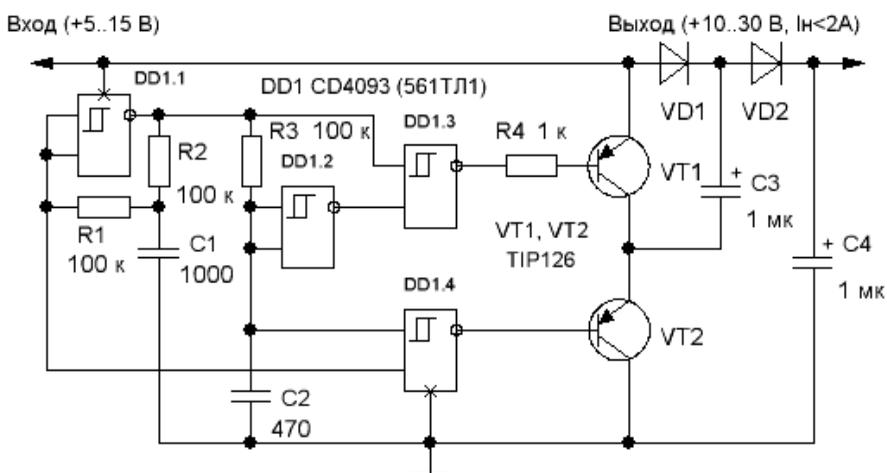
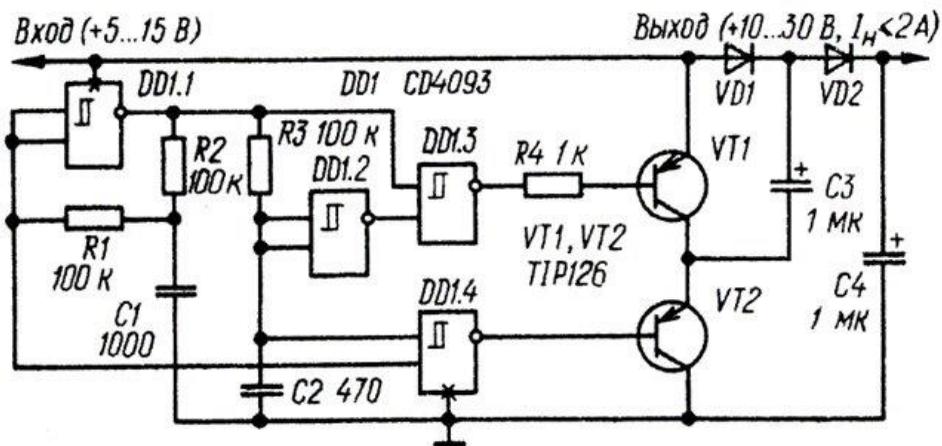


Рис. 2

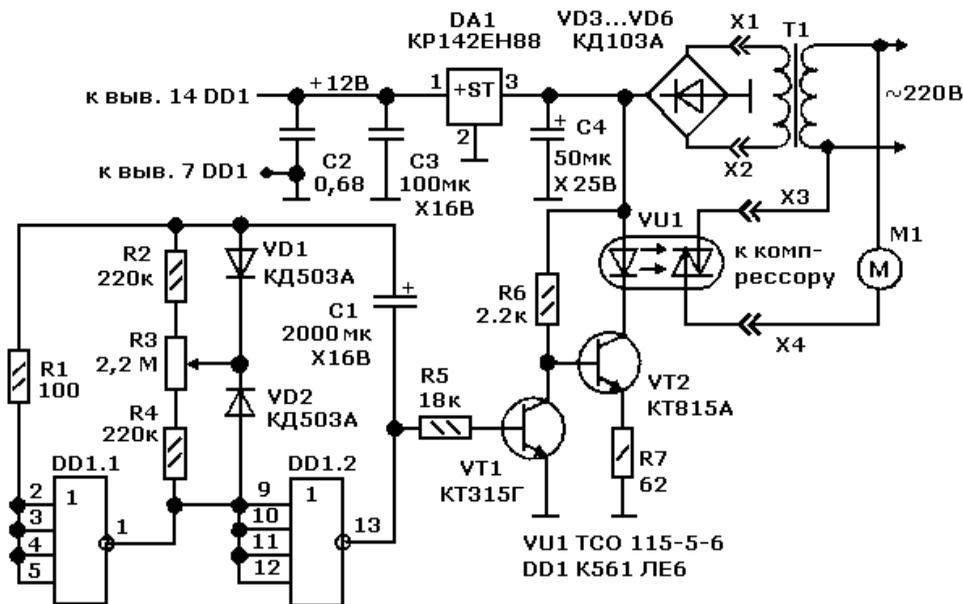
Вариант 11



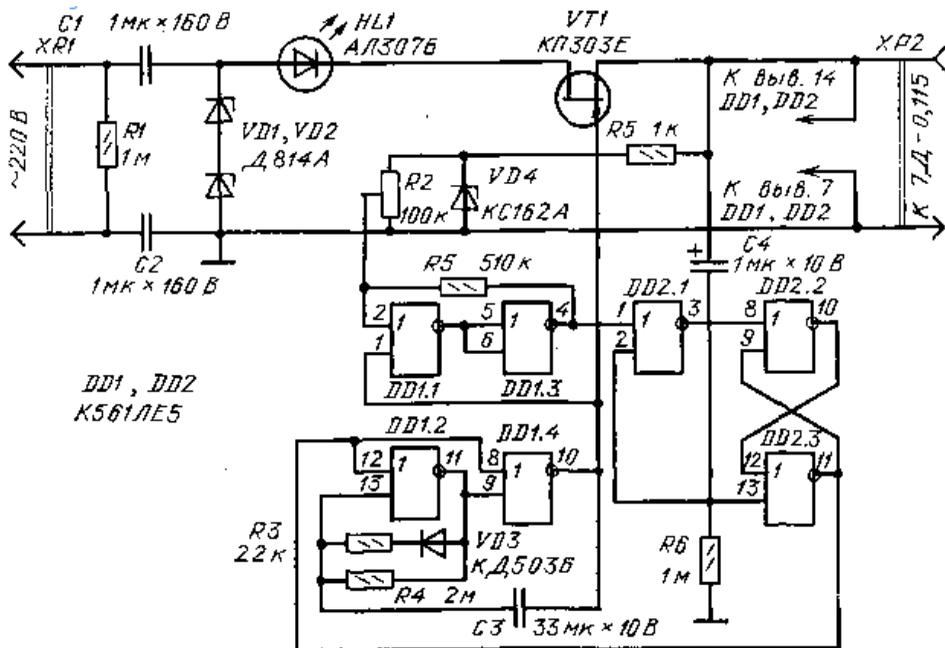
Вариант 12



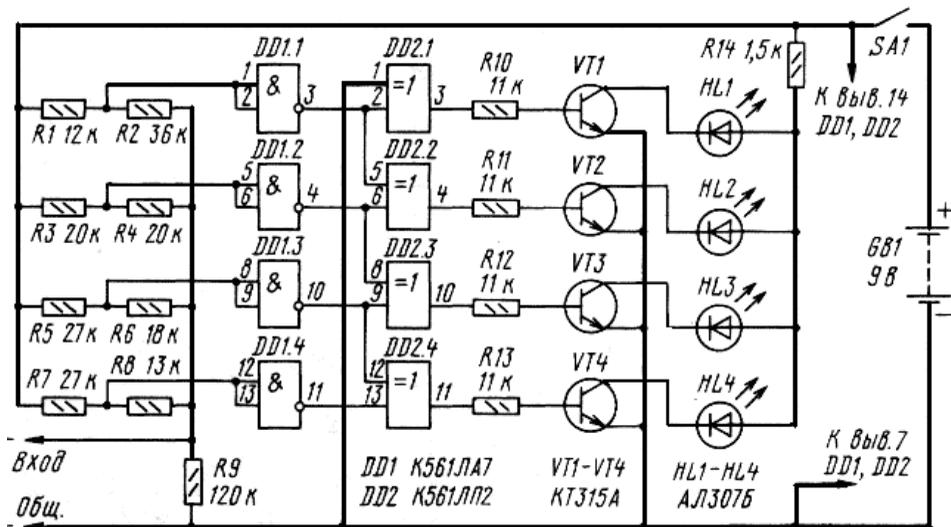
Вариант 13



Вариант 14



Вариант 15



Вариант 16

## 2. ЛАБОРАТОРНАЯ РАБОТА №2: ИССЛЕДОВАНИЕ ЛОГИЧЕСКИХ СХЕМ

### 2.1. Цель работы

Цель работы – исследование логических схем «И2», «И-НЕ», «ИЛИ», «ИЛИ-НЕ», «Исключающее ИЛИ».

### 2.2. Краткие сведения из теории

#### 2.2.1. Аксиомы алгебры логики

Переменные, рассматриваемые в алгебре логики, могут принимать только два значения ноль или единица. В алгебре логики определены:

- отношение эквивалентности, обозначаемое знаком = ;
- операция сложения (дизъюнкция), обозначаемая знаком + или  $\vee$  ;
- операция умножения (конъюнкция), обозначаемая знаком & или \* ;
- операция отрицания (или инверсия), обозначаемая знаком надчеркивания или апострофом ' .

Алгебра логики определяется следующей системой аксиом:

$$x = 0, \text{ если } x \neq 1, \quad \bar{0} = 1,$$

## Основы микропроцессорной техники

$$\begin{array}{ll}
 x = 1, \text{ если } x \neq 0, & \bar{1} = 0, \\
 1 + 1 = 1, & 0 * 0 = 0, \\
 0 + 0 = 0, & 1 * 1 = 1, \\
 0 + 1 = 1 + 0 = 1, & 1 * 0 = 0 * 1 + 0.
 \end{array}$$

**2.2.2. Логические выражения**

Логические выражения связывают значение логической функции со значениями логических переменных. Они могут записываться или в конъюнктивной или дизъюнктивной нормальных формах. В дизъюнктивной форме логические выражения записываются как логическая сумма логических произведений, в конъюнктивной – как логическое произведение логических сумм. Порядок действий в логических выражениях такой же, как и в обычных алгебраических выражениях. Логические выражения связывают значение логической функции со значениями логических переменных.

**2.2.3. Законы булевой алгебры**

Они вытекают из аксиом и имеют две формы выражения: для конъюнкции и дизъюнкции. Эти законы используются при преобразованиях логических выражений.

Переместительный закон:  $xy = yx$ ;  $x + y = y + x$ ;

сочетательный закон:  $x(yz) = (xy)z = xyz$ ;  $x + (y + z) = (x + y) + z = x + y + z$ ;

распределительный закон:  $x(y + z) = xy + xz$ ;  $x + yz = (x + y)(x + z)$ ;

закон повторения:  $x + x = x$ ;  $x * x = x$ ;

закон обращения: если  $x = y$ , то  $\bar{x} = \bar{y}$ ;

закон двойной инверсии:  $\bar{\bar{x}} = x$ ;

закон универсального множества:  $x * 1 = x$ ;  $x + 1 = 1$ ;

закон дополнительности:  $x\bar{x} = 0$ ;  $x + \bar{x} = 1$ ;

закон нулевого множества:  $x * 0 = 0$ ;  $x + 0 = x$ ;

закон поглощения:  $x + x * y = x$ ;  $xy + x\bar{y} = x$ ;

закон склеивания:  $(x + y)(x + \bar{y}) = x$ ;  $xy + x\bar{y} = x$ ;

закон инверсии (закон Де Моргана):  $x\bar{y} = \bar{x} + \bar{y}$ ;

$\bar{x} + \bar{y} = \overline{xy}$ .

### 2.2.4. Логические функции

Любое логическое выражение, составленное из  $n$  переменных  $x_n, x_{n-1}, \dots, x_1$  с помощью конечного числа операций алгебры логики, можно рассматривать как некоторую функцию  $n$  переменных, называемую логической. В соответствии с аксиомами алгебры логики функция может принимать в зависимости от значения переменных значение 0 или 1. Функция  $n$  логических переменных может быть определена для  $2^n$  значений переменных, соответствующих всем возможным значениям  $n$ -разрядных двоичных чисел.

Основной интерес представляют следующие функции двух переменных  $x$  и  $y$ :

$$f_1(x, y) = x * y \text{ – логическое умножение,}$$

$$f_2(x, y) = x + y \text{ – логическое сложение,}$$

$$f_3(x, y) = \overline{x * y} \text{ – логическое умножение с инверсией,}$$

$$f_4(x, y) = \overline{x + y} \text{ – логическое сложение с инверсией,}$$

$f_5(x, y) = x \oplus y = \overline{xy} + \overline{\overline{xy}} = \overline{xy} + xy \text{ – суммирование по модулю два или «Исключающее ИЛИ»,}$

$$f_6(x, y) = x \odot y = xy + \overline{xy} \text{ – равнозначность.}$$

### 2.2.5. Логические схемы

Физическое устройство, реализующее одну из операций алгебры логики или простейшую логическую функцию, называется логическим элементом. Схема, составленная из конечного числа логических элементов по определенным правилам, называется логической. Основным логическим функциям соответствуют выполняющие их схемные элементы. Например, функции  $f_1(x, y)$  соответствует логическая схема «И», функции  $f_2(x, y)$  – логическая схема «ИЛИ», функции  $f_3(x, y)$  – логическая схема «И-НЕ», функции  $f_4(x, y)$  – логическая схема «ИЛИ-НЕ».

### 2.2.6. Таблица истинности

Так как область определения любой функции  $n$  переменных конечна (может принимать  $2^n$  значений), то такая функция может быть задана таблицей значений  $f(x)$ , которые она принимает в точках  $x_i$ , где  $i = 0, 1, \dots, 2^n - 1$ . Такие таблицы называются таблицами истинности. В табл. 1 представлены значения функций  $f_1(x, y), \dots, f_6(x, y)$ .

Таблица 2.1 – Таблица истинности

$i$	Значения переменных		Функции					
	$X$	$Y$	$f_1(x,y)$	$f_2(x,y)$	$f_3(x,y)$	$f_4(x,y)$	$f_5(x,y)$	$f_6(x,y)$
0	0	0	0	0	1	1	0	1
1	0	1	0	1	1	0	1	0
2	1	0	0	1	1	0	1	0
3	1	1	1	1	0	0	0	1

### 2.3. Последовательность выполнения работы

#### 2.3.1. Исследование логической функции «И»

1. Нарисовать схему исследования функции «И» (рис.2.1).

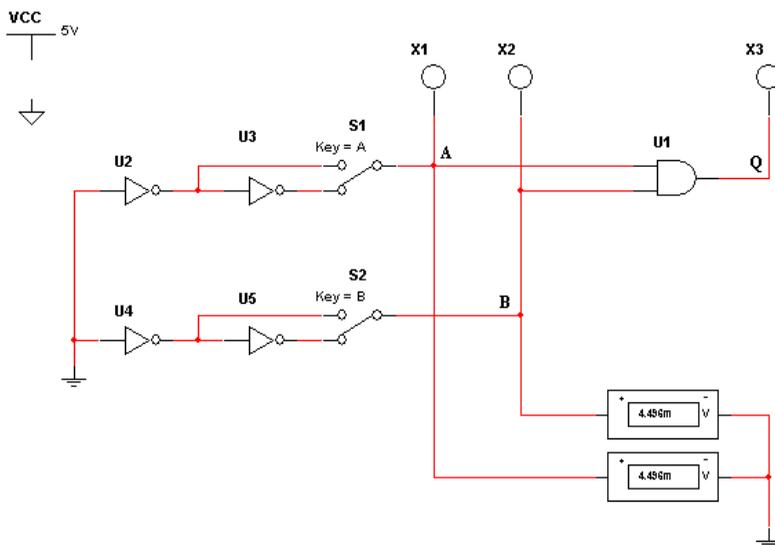


Рис. 2.1. Схема исследования функции «И»

Схема содержит исследуемую функцию «И» ( $U1$ ), два двухпозиционных переключателя ( $S1$ ,  $S2$ ), управляемые клавишами  $A$  и  $B$  (заглавные буквы латинского алфавита), источники сигналов логической единицы ( $U2, U4$ ), логического нуля ( $U3, U5$ ), три светодиода ( $X1$ ,  $X2$ ,  $X3$ ), два вольтметра и источник постоянного напряжения 5В ( $VCC$ ).



2. Запустить процесс моделирования, нажав кнопку *Run*.

3. Подать на входы схемы «И» все возможные комбинации уровней сигналов *A* и *B* с помощью переключателей *S1* и *S2*. И для каждой комбинации зафиксировать показания вольтметров и уровни входных сигналов *A* и *B* и уровень выходного сигнала *Q* (логическая единица – соответствующий светодиод *Xi* светится, логический ноль – соответствующий светодиод *Xi* не светится). Результаты измерений занести в таблицу истинности (табл. 2.2).

Таблица 2.2

Входы		Выход
A	B	Q
0	0	
0	1	
1	0	
1	1	

### 2.3.2. Исследование логической функции «И-НЕ»

1. Нарисовать схему исследования функции «И-НЕ» (рис. 2.2).

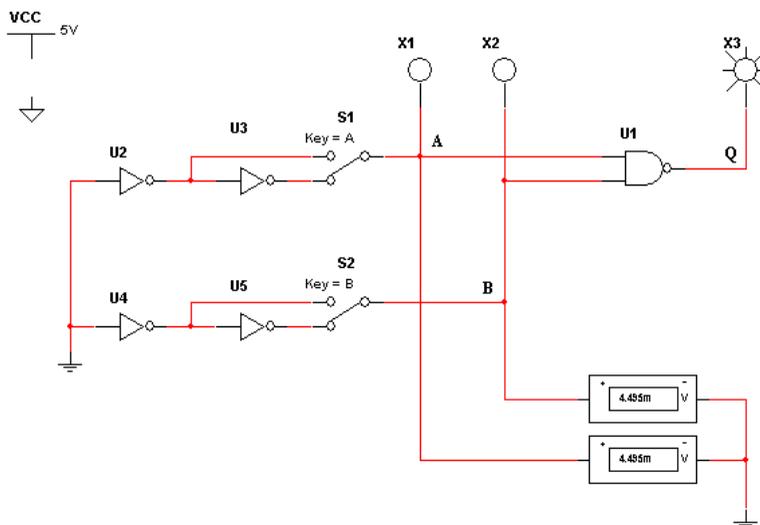


Рис. 2.2. Схема исследования функции «И-НЕ»

2. Запустить процесс моделирования, нажав кнопку  на панели инструментов, и в появившемся меню выбрать команду *Run*.

3. Подать на входы схемы «И-НЕ» все возможные комбинации уровней сигналов *A* и *B* с помощью переключателей *S1* и *S2*. И для каждой комбинации зафиксировать показания вольтметров, уровни входных сигналов *A* и *B* и уровень выходного сигнала *Q* (логическая единица – соответствующий светодиод  $X_i$  светится, логический ноль – соответствующий светодиод  $X_i$  не светится). Результаты измерений занести в таблицу истинности (табл. 2.3).

Таблица 2.3

Входы		Выход
A	B	Q
0	0	
0	1	
1	0	
1	1	

### 2.3.3. Исследование логической функции «ИЛИ»

1. Нарисовать схему исследования функции «2ИЛИ» (рис.2.3).

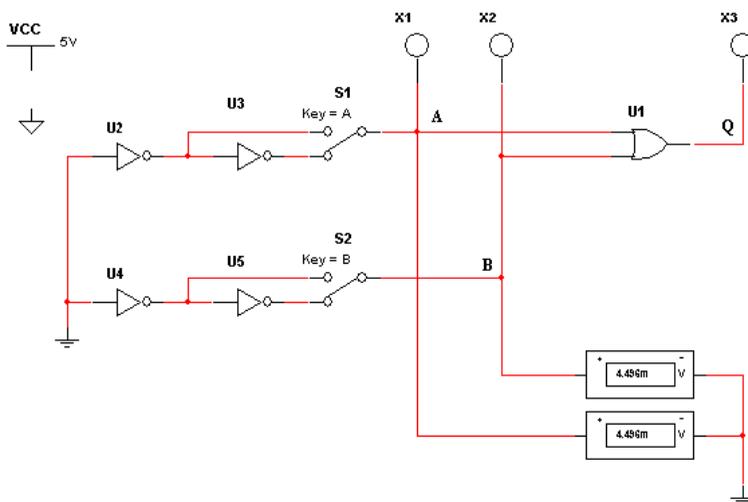


Рис.2.3. Схема исследования функции «2ИЛИ»

2. Запустить процесс моделирования, нажав кнопку  на панели инструментов, и в появившемся меню выбрать команду *Run*.

3. Подать на входы схемы «ИЛИ» все возможные комбинации уровней сигналов  $A$  и  $B$  с помощью переключателей  $S1$  и  $S2$ . И для каждой комбинации зафиксировать показания вольтметров, уровни входных сигналов  $A$  и  $B$  и уровень выходного сигнала  $Q$  (логическая единица – соответствующий светодиод  $X_i$  светится, логический ноль – соответствующий светодиод  $X_i$  не светится). Результаты измерений занести в таблицу истинности (табл. 2.4).

Таблица 2. 4

Входы		Выход
A	B	Q
0	0	
0	1	
1	0	

### 2.3.4. Исследование логической функции «ИЛИ-НЕ»

1. Нарисовать схему исследования функции «2ИЛИ-НЕ» (рис.2.4).

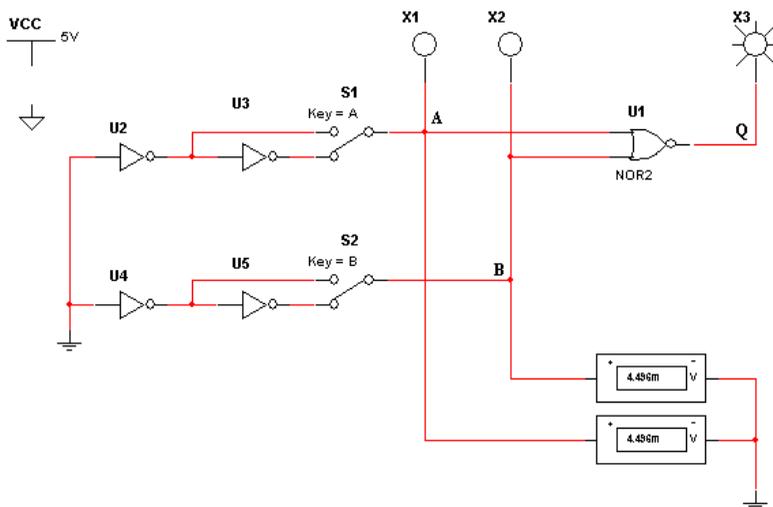


Рис.2.4. Схема исследования функции «2ИЛИ-НЕ»

2. Запустить процесс моделирования, нажав кнопку  на панели инструментов, и в появившемся меню выбрать команду *Run*.

3. Подать на входы схемы «ИЛИ-НЕ» все возможные комбинации уровней сигналов *A* и *B* с помощью переключателей *S1* и *S2*. И для каждой комбинации зафиксировать показания вольтметров, уровни входных сигналов *A* и *B* и уровень выходного сигнала *Q* (логическая единица – соответствующий светодиод *X<sub>i</sub>* светится, логический ноль – соответствующий светодиод *X<sub>i</sub>* не светится). Результаты измерений занести в таблицу истинности (табл. 2.5).

Таблица 2.5

Входы		Выход
A	B	Q
0	0	
0	1	
1	0	
1	1	

### 2.3.5. Исследование логической функции «Исключающее ИЛИ»

1. Нарисовать схему исследования функции «Исключающее ИЛИ» (рис. 2.5).

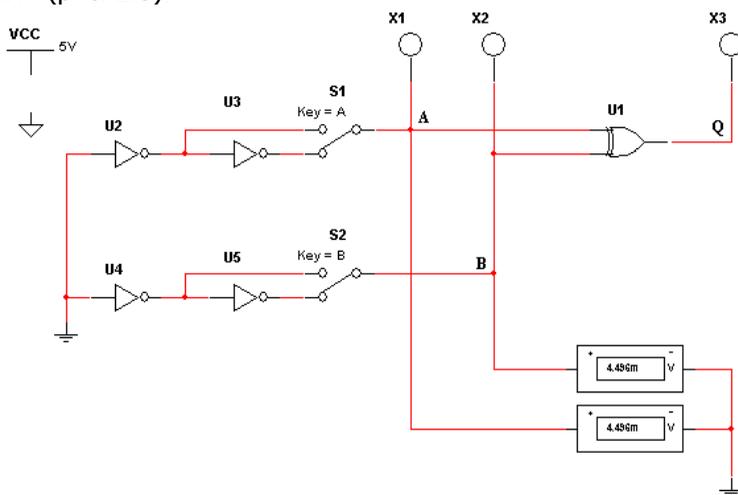


Рис. 2.5. Схема исследования функции «Исключающее ИЛИ»

2. Запустить процесс моделирования, нажав кнопку  на панели инструментов, и в появившемся меню выбрать команду *Run*.

3. Подать на входы схемы «Исключающее ИЛИ» все возможные комбинации уровней сигналов  $A$  и  $B$  с помощью переключателей  $S1$  и  $S2$ . И для каждой комбинации зафиксировать показания вольтметров, уровни входных сигналов  $A$  и  $B$  и уровень выходного сигнала  $Q$  (логическая единица – соответствующий светодиод  $X_i$  светится, логический ноль – соответствующий светодиод  $X_i$  не светится). Результаты измерений занести в таблицу истинности (табл. 2.6).

Таблица 2.6

Входы		Выход
A	B	Q
0	0	
0	1	
1	0	
1	1	

#### 2.4. Индивидуальное задание

1. Создайте в программе электронную схему, приведенную на рисунке 2.6, в соответствии с вашим вариантом, представленным в таблице 2.7. Проверьте работу схемы, подав на ее входы двоичные коды от генератора слов. Постройте таблицу истинности.

2. Используя логический конвертер, проверьте правильность полученной таблицы истинности.

3. С помощью логического конвертера получите логическое выражение для построенной схемы и в нем же упростите его.

3. Для полученного упрощенного логического выражения, с помощью логического конвертера постройте логическую схему в базе 2И-НЕ для вашего исходного варианта логической схемы.

4. Все перечисленные действия должны быть представлены в виде скриншотов.

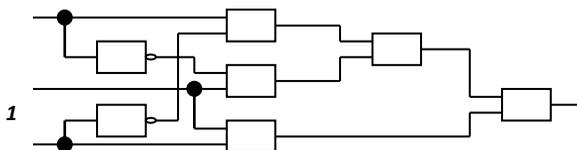


Рисунок 2.6. Пример схемы

## Основы микропроцессорной техники

Таблица 2.7 - Варианты заданий

№ варианта	ДД1	ДД2	ДД3	ДД4	ДД5	ДД6	ДД7
1	*	-	И	ИЛИ	И-НЕ	ИЛИ	И
2	-	*	ИЛИ-НЕ	И-НЕ	ИЛИ-НЕ	И	ИЛИ
3	-	*	И	ИЛИ	ИЛИ-НЕ	И	ИЛИ-НЕ
4	-	*	И-НЕ	ИЛИ-НЕ	И-НЕ	ИЛИ	И-НЕ
5	*	-	ИЛИ	И	И-НЕ	ИЛИ	И-НЕ
6	*	-	И-НЕ	И	ИЛИ-НЕ	И	ИЛИ-НЕ
7	-	*	ИЛИ-НЕ	ИЛИ	ИЛИ	И-НЕ	И
8	*	-	И	И-НЕ	ИЛИ-НЕ	И	ИЛИ-НЕ
9	-	*	И-НЕ	И	И-НЕ	ИЛИ	И-НЕ
10	*	-	ИЛИ	И	ИЛИ-НЕ	И	ИЛИ-НЕ
11	*	-	ИЛИ-НЕ	ИЛИ	И-НЕ	ИЛИ	И-НЕ
12	-	*	ИЛИ-НЕ	ИЛИ	И	ИЛИ-НЕ	ИЛИ
13	-	*	ИЛИ	И-НЕ	ИЛИ	И-НЕ	И
14	-	*	ИЛИ-НЕ	И-НЕ	ИЛИ-НЕ	И	ИЛИ-НЕ
15	*	-	И-НЕ	ИЛИ	И	ИЛИ-НЕ	ИЛИ-НЕ
16	*	-	ИЛИ-НЕ	И	ИЛИ	И-НЕ	И
17	*	-	И	ИЛИ	И-НЕ	ИЛИ	И-НЕ
18	*	-	И-НЕ	И	ИЛИ-НЕ	И	ИЛИ
19	-	*	ИЛИ-НЕ	И-НЕ	И	И	ИЛИ
20	-	*	ИЛИ-НЕ	ИЛИ	ИЛИ	И-НЕ	И-НЕ
21	*	-	И	И-НЕ	ИЛИ	И-НЕ	И-НЕ
22	*	-	И-НЕ	ИЛИ	И	ИЛИ-НЕ	ИЛИ-НЕ
23	-	*	ИЛИ-НЕ	ИЛИ	ИЛИ	И-НЕ	И
24	-	*	И-НЕ	И	И-НЕ	ИЛИ	И-НЕ
25	*	-	ИЛИ	И-НЕ	И	ИЛИ-НЕ	ИЛИ
26	*	-	ИЛИ-НЕ	ИЛИ	ИЛИ	И-НЕ	И-НЕ
27	-	*	И-НЕ	И	И	ИЛИ-НЕ	ИЛИ
28	*	-	ИЛИ	ИЛИ-НЕ	ИЛИ-НЕ	И	ИЛИ
29	-	*	И	ИЛИ	И	ИЛИ-НЕ	ИЛИ-НЕ
30	*	-	ИЛИ	И	И-НЕ	ИЛИ	И
31	*	-	ИЛИ-НЕ	ИЛИ	И	ИЛИ-НЕ	ИЛИ-НЕ
32	*	-	ИЛИ	ИЛИ-НЕ	И-НЕ	ИЛИ	И-НЕ
33	-	*	ИЛИ	И-НЕ	ИЛИ-НЕ	И	ИЛИ
34	-	*	ИЛИ-НЕ	И	ИЛИ-НЕ	ИЛИ-НЕ	ИЛИ
35	*	-	ИЛИ-НЕ	ИЛИ-НЕ	И-НЕ	ИЛИ	И-НЕ
36	*	-	И	И-НЕ	И-НЕ	ИЛИ	И-НЕ
37	-	*	И-НЕ	И	ИЛИ-НЕ	И	ИЛИ-НЕ
38	*	-	ИЛИ	ИЛИ-НЕ	ИЛИ	И-НЕ	И
39	-	*	ИЛИ	И	ИЛИ	И-НЕ	И-НЕ
40	*	-	ИЛИ	И-НЕ	И	ИЛИ-НЕ	ИЛИ-НЕ

ДД1, ДД2 – инверторы, прочерк в таблице соответствует отсутствию инвертора(заменяется проводником), знак «\*» соответствует наличию инвертора в схеме.

## 2.5. Содержание отчета

Отчет должен включать:

- 1) название пункта работы,
- 2) исследуемые схемы,
- 3) результаты моделирования,
- 4) выводы.

## 3. ЛАБОРАТОРНАЯ РАБОТА №3: ИССЛЕДОВАНИЕ АРИФМЕТИЧЕСКОГО СУММАТОРА С ИСПОЛЬЗОВАНИЕМ ПРОГРАММНОГО КОМПЛЕКСА MULTISIM

### 3.1. Цель работы

*Цель работы* – исследовать арифметический сумматор, полусумматор и полный сумматор и вычитатель.

### 3.2. Краткие сведения из теории

Арифметические сумматоры – составная часть арифметико-логических устройств (АЛУ) микропроцессоров (МП). Арифметический сумматор состоит из двух устройств: полусумматора и  $n$  полных сумматоров. Полный сумматор имеет три входа:  $A$ ,  $B$  – входы суммируемых операндов,  $C_i$  – вход переноса из предыдущего разряда сумматора и два выхода:  $S$  – выход полного сумматора и  $C_0$  – выход переноса. Полусумматор отличается от полного тем, что у него нет входа переноса из предыдущего разряда. Полусумматор используется в качестве первого разряда арифметического сумматора, а в качестве остальных разрядов – полные сумматоры (рис. 3.1). Полусумматор – одна из простейших комбинационных логических схем.

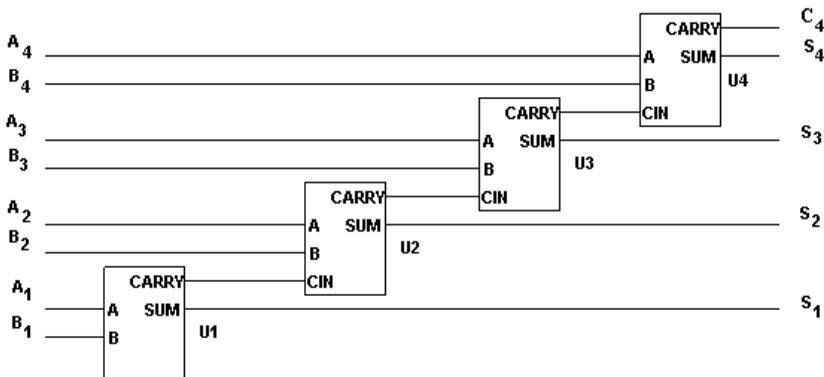


Рис. 3.1. Четырехразрядный арифметический сумматор

## Основы микропроцессорной техники

Рассматривая таблицу истинности полусумматора (табл. 3.1) можно заметить, что выход  $S$  полусумматора выполняет функции элемента «ИСКЛЮЧАЮЩЕЕ ИЛИ», а выход переноса  $C$  полусумматора – элемента «И». Таким образом, логические выражения для функций  $S$  и  $C$  равны:

$$S = AB' + A'B, \quad C = AB.$$

Таблица 3.1

Входы		Выходы	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Схема полусумматора представлена на рис. 3.2.

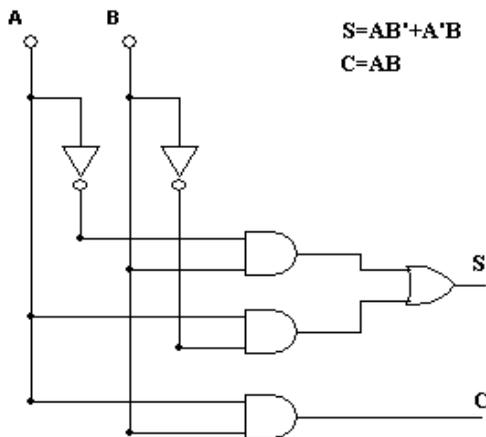


Рис. 3.2. Структура полусумматора

Из таблицы истинности полного сумматора (табл. 3.2) можно получить логические выражения для  $S$  (суммы) и  $C$  (переноса в следующий разряд). Логическое выражение для  $S$  будет иметь четыре слагаемых, соответствующих строкам таблицы, в которых выход  $S$  равен единице (строки 4, 5, 7, 10),

$$S = A'B'C_{i-1} + A'BC_{i-1}' + AB'C_{i-1}' + ABC_{i-1}.$$

Таблица 3.2

Входы			Выходы	
<i>A</i>	<i>B</i>	<i>C<sub>i-1</sub></i>	<i>S</i>	<i>C<sub>i</sub></i>
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Логическое выражение для *S* также будет иметь четыре слагаемых (строки 6, 8, 9, 10):

$$C_i = A'BC_{i-1} + A'BC_{i-1}' + ABC_{i-1}' + ABC_{i-1}$$

С помощью законов булевой алгебры (см. лаб. раб. №1) это выражение можно упростить, тогда оно будет иметь вид

$$C_i = AC_{i-1} + BC_{i-1} + AB$$

Схема полного сумматора изображена на рис. 3.3.

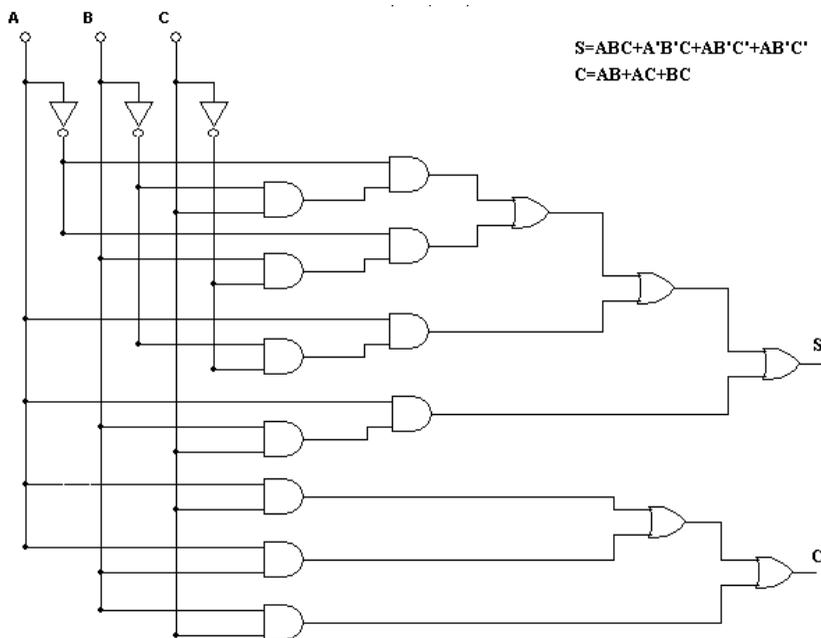


Рис. 3.3. Структура полного сумматора

### 3.3. Последовательность выполнения работы

1. Собрать (нарисовать) схему четырехразрядного арифметического сумматора (рис. 3.4). Поместить на схему три 16-ричных индикатора и генератор слова.

2. Открыть генератор слова и задать суммируемые числа. Четыре младших разряда каждого генерируемого слова составляют первое слагаемое (операнд). Следующие четыре разряда составляют второе слагаемое (операнд).

3. Запустить процесс моделирования и следить за показаниями индикаторов. Записать суммируемые числа и результат суммирования.

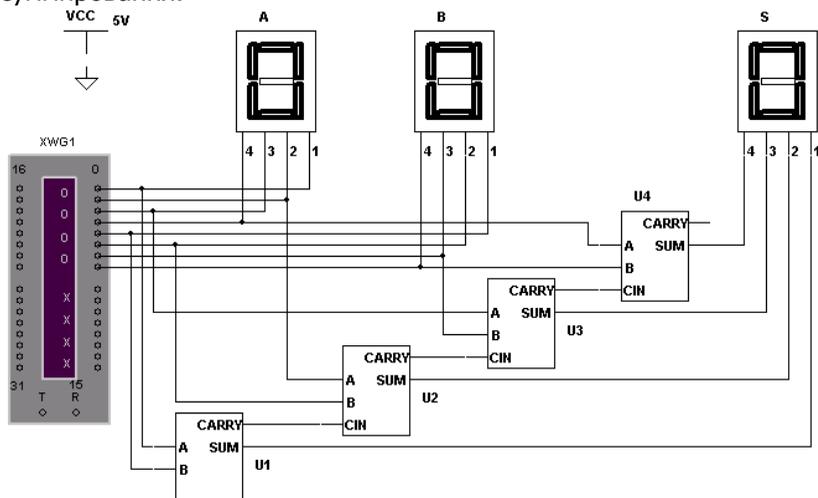
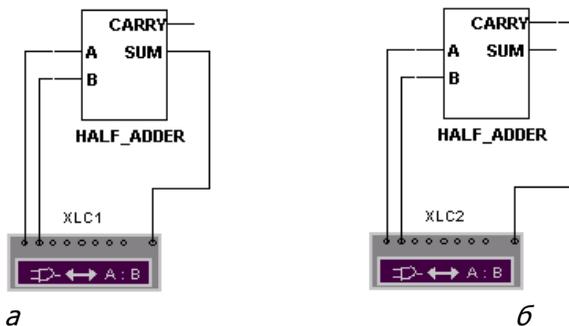


Рис. 3.4 . Схема исследования четырехразрядного сумматора

4. Собрать схему, изображенную на рис. 3.5, и с помощью логического анализатора, последовательно нажимая кнопки *Circuit to Truth Table* (таблица истинности цепи) , *Truth Table to Boolean Expression* (булево выражение по таблице истинности) , *Boolean Expression to Circuit* (создание схемы по булеву выражению) , получить:

- таблицу истинности полусумматора,
- логические выражения для выходов  $S$  и  $C$ ,
- схему реализации логических выражений для выходов  $S$  и  $C$ .

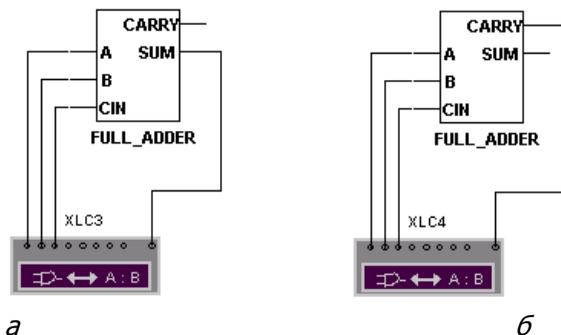


а

б

Рис. 3.4. Схема исследования полусумматора: а – выход  $S$ , б – выход  $C$

5. Собрать схему, изображенную на рис. 3.5, и с помощью логического анализатора получить таблицу истинности полного сумматора, логические выражения для выходов  $S$  и  $C$  и схемную реализацию логических выражений (см. п. 4).



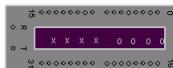
а

б

Рис. 3.5. Схема исследования полного сумматора: а – выход  $S$ , б – выход  $C$

### 3.4. Описание используемых контрольно-измерительных приборов

Генератор слова (Word Generator)



Генератор слова (или кодовый генератор) предназначен для генерации 32-разрядных двоичных слов, которые набираются пользователем в 16-ричном коде в строке *Hex* или в двоичном коде в строке *Binary* на панели *Edit* (рис. 3.6).

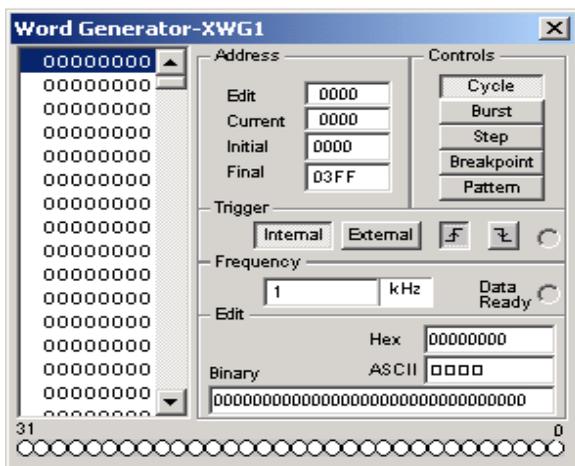


Рис. 3.6. Лицевая панель генератора слова

В окне, расположенном слева на лицевой панели генератора слова, отображаются 8-разрядные 16-ричные числа от 00000000 до FFFFFFFF (от 0 до 4294967265). Каждая горизонтальная строка представляет 32-разрядное двоичное число. Эти числа поступают в параллельном виде на выходные клеммы генератора, расположенные в нижней части лицевой панели.

Чтобы изменить значение любого бита кодового слова, надо выбрать число (щелкнуть по нему, при этом оно меняет цвет), которое необходимо изменить, и ввести новое значение в строках *HEX*, *ASCII* или *Binary* на панели *Edit*. Измененное кодовое слово отображается на выходных клеммах генератора, расположенных в нижней части лицевой панели.

На панели *Address* расположены четыре окна. Каждое кодовое слово из списка имеет адрес, выраженный 4-разрядным 16-ричным числом.

В окне *Edit* отображается адрес выбранного в таблице слова, в окне *Current* – адрес выдаваемого кодового слова.

В окне *Initial* устанавливается адрес первого кодового слова множества слов, поступающих на выход генератора, в окне *Final* – адрес последнего кодового слова множества слов, поступающих на выход генератора.

Для того чтобы создать множество кодовых слов, выдаваемых генератором слова, нужно ввести адрес первого и последнего слова в окно *Initial* и *Final* соответственно.

На панели *Controls* устанавливается режим выдачи кодовых слов.

## Основы микропроцессорной техники

Чтобы выдать 32-разрядное слово на выход прибора, надо щелкнуть по одной из кнопок *Step*, *Burst* or *Cycle*. Номер этого слова отобразится в окне *Current* на панели *Address*.

Если необходимо выдать только одно слово, следует щелкнуть по кнопке *Step*, если все кодовые слова множества, то щелкнуть по кнопке *Burst*.

Если щелкнуть по кнопке *Cycle*, то будут выдаваться все кодовые слова множества непрерывно циклически. Остановить выдачу слов можно, повторно щелкнув по кнопке *Cycle*.

Если нужно остановить и возобновить выдачу слов с определенного слова, нужно щелкнуть по кнопке *Breakpoint*.

Чтобы установить контрольную точку (*Breakpoint*), нужно выбрать в списке кодовое слово, на котором следует остановить вывод слов, и затем щелкнуть по кнопке *Breakpoint*. У этого слова появится метка в виде звездочки.

Чтобы удалить контрольную точку, нужно выбрать существующую контрольную точку, затем щелкнуть по кнопке *Breakpoint*.

Можно установить несколько контрольных точек. Контрольные точки могут использоваться как при непрерывной (*Cycle*), так и при однократной (*Burst*) выдаче множества слов.

С помощью кнопки *Pattern* можно создавать новые или использовать ранее записанные множества кодовых слов.

На панели *Triggering* расположены четыре кнопки, с помощью которых можно установить источник запускающего сигнала (внутренний (*Internal*) или внешний (*External*)) и фазу запускающего сигнала (по переднему или заднему фронту).

На панели *Frequency* устанавливается тактовая частота генератора слова в герцах, кило- или мегагерцах. Кодовые слова поступают на выход генератора с каждым тактом генератора. Рядом расположена клемма, на которую выдается сигнал готовности выдавать данные.

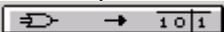
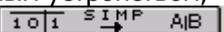
*Логический преобразователь (Logic Converter)*



На лицевой панели преобразователя (рис. 3.7) расположены клеммы-индикаторы входов *A*, *B*, *C*, ..., *N* и клемма выхода *Out*, окно для отображения таблицы истинности исследуемой схемы, строка для отображения ее булева выражения и панель *Conversions*.

На панели *Conversions* расположены шесть кнопок, исполь-

зубых для получения:

-  →  $\overline{101}$  – таблицы истинности исследуемого устройства,
-  →  $A|B$  – булева выражения, реализуемого исследуемым устройством,
-  →  $A|B$  – минимизированного булева выражения,

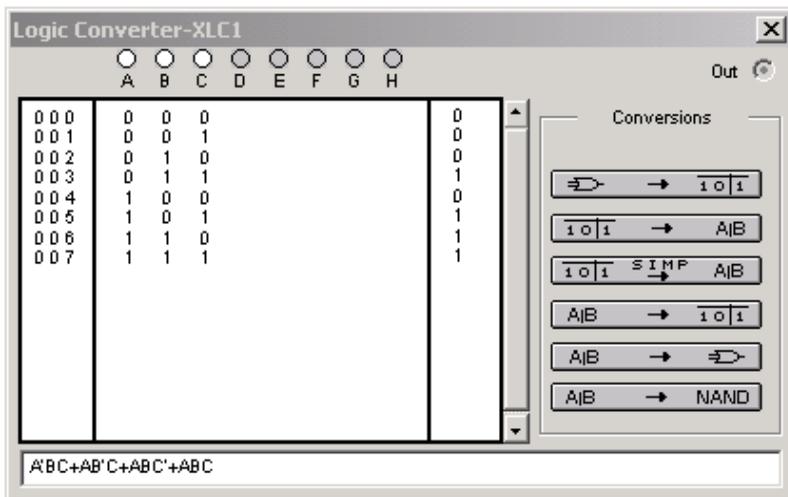
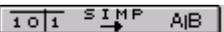
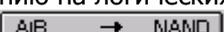


Рис. 3.7. Лицевая панель логического преобразователя

-  →  $A|B$  – таблицы истинности по булевому выражению;
-  →  $\Rightarrow$  – схемы устройства по логическому выражению на логических элементах без ограничения их типа,
-  → NAND – создания схемы устройства только на логических элементах «И-НЕ».

### 3.5. Индивидуальное задание

1. Построить схему полного сумматора и произвести, согласно вашего варианта, представленного в таблице 3.3, суммирование двух чисел с получением результата ответа. Количество разрядов сумматора будет определяться разрядностью исходных данных

2. Самостоятельно найти и построить схему полного вычитателя. С помощью логического конвертера построить таблицу истинности полного вычитателя.

Таблица 3.3

№ ва р	Слагае- мые	№ ва р	Слагае- мые	№ ва р	Слагае- мые	№ ва р	Слагае- мые
1	17 и 6	9	39 и 7	17	13 и 27	25	16 и 32
2	12 и 12	10	56 и 6	18	46 и 7	26	26 и 18
3	24 и 1	11	34 и 9	19	2 и 53	27	40 и 20
4	31 и 0	12	19 и 19	20	22 и 28	28	50 и 10
5	60 и 2	13	8 и 14	21	1 и 56	29	30 и 30
6	45 и 3	14	14 и 24	22	4 и 44	30	31 и 31
7	17 и 15	15	61 и 1	23	5 и 25	31	11 и 12
8	8 и 23	16	22 и 29	24	11 и 27	32	25 и 6

### 3.6. Содержание отчета

Отчет должен включать:

- 1) название пункта работы,
- 2) исследуемые схемы,
- 3) результаты моделирования,
- 4) краткие выводы.

## 4. ЛАБОРАТОРНАЯ РАБОТА №4: ИССЛЕДОВАНИЕ ТРИГГЕРОВ В MULTISIM

### 4.1. Цель работы

Изучение различных схем триггеров и их функционирования, приобретение навыков в определении характеристик триггеров.

### 4.2. R-S триггер

#### 4.2.1. Основные теоретические положения

Триггер (защелка) представляет собой устройство с двумя устойчивыми состояниями.

Наибольшее распространение получили RS-триггеры, построенные на логических элементах 2И-НЕ или 2ИЛИ-НЕ. На рисунке 4.1,а приведена функциональная схема RS-триггера с инверсными входами на двух логических элементах 2И-НЕ, а на рисунке 4.1,б – его условное обозначение на принципиальных схемах. RS- триггер с прямыми входами можно получить, имея в наличии два логических элемента 2ИЛИ-НЕ. Триггер получается путем соединения выхода первого логического элемента с одним

из входов второго и соединения выхода второго логического элемента с одним из входов первого. Как видно из получившейся схемы (рис. 4.2,а), по отношению к логическим элементам триггер симметричен. По этой причине не имеет принципиального значения, выход какого из элементов считать прямым выходом триггера.

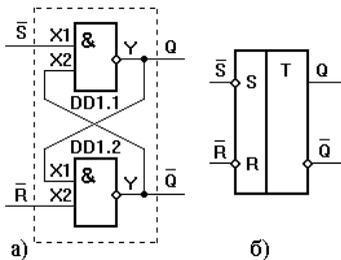


Рис. 4.1

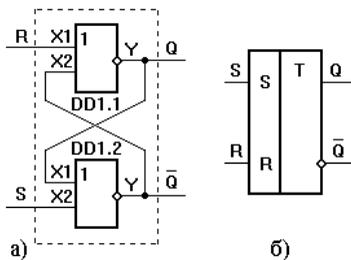


Рис. 4.2

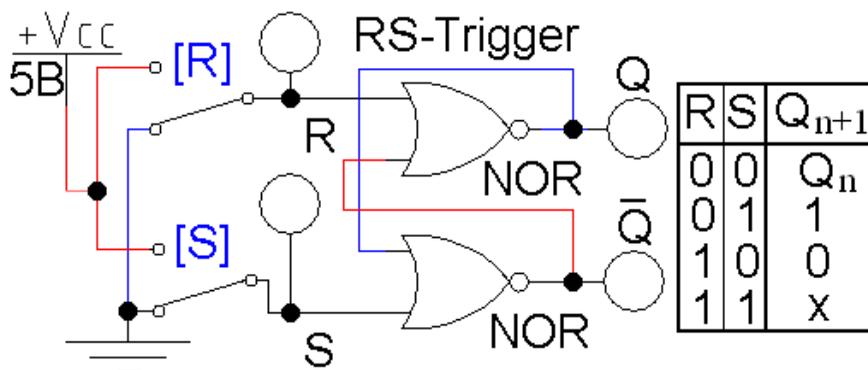
По способу записи информации различают триггеры: асинхронные; синхронные.

У асинхронных триггеров запись информации осуществляется с поступлением информационного сигнала на его вход.

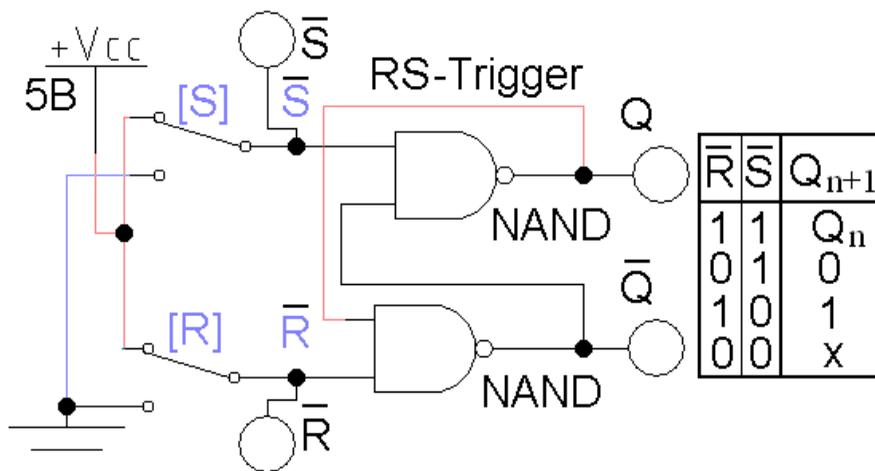
Асинхронный  $RS$ -триггер имеет два выхода: прямой  $Q$  и инверсный  $Q'$  и два входа  $S$  (set) установка ( $Q = 1$ ) и  $R$  (reset) сброс ( $Q = 0$ ).

Выход триггера переходит в состояние  $Q_{n+1} = 0$ , если  $S = 0$ ,  $R = 1$ . При  $S = 1$ ,  $R = 0$  на выходе  $Q_{n+1} = 1$ . Когда  $S = 0$ ,  $R = 0$  триггер сохраняет прежнее значение  $Q_n$ . Комбинация сигналов  $S = 1$ ,  $R = 1$  запрещена, т.к. на прямом  $Q$  и инверсном  $Q'$  выходе устанавливаются одинаковые значения, которые при переходе в режим хранения не сохраняются.

На рисунках 4.3, 4.4 изображены асинхронные  $RS$ -триггеры и их таблицы истинности.


 Рис. 4.3 Схема *RS*-триггера на элементах 2ИЛИ-НЕ

В синхронных триггерах, имеющих информационные и тактовые входы, запись осуществляется только при подаче разрешающего тактового импульса.


 Рис. 4.4 Схема *RS*-триггера на элементах 2И-НЕ

Триггеры в виде логических компонентов расположены в поле элементов *Digital* (рис. 4.5).

Использование генератора слов позволяет проанализировать работу *RS*-триггера на всех наборах переменных.

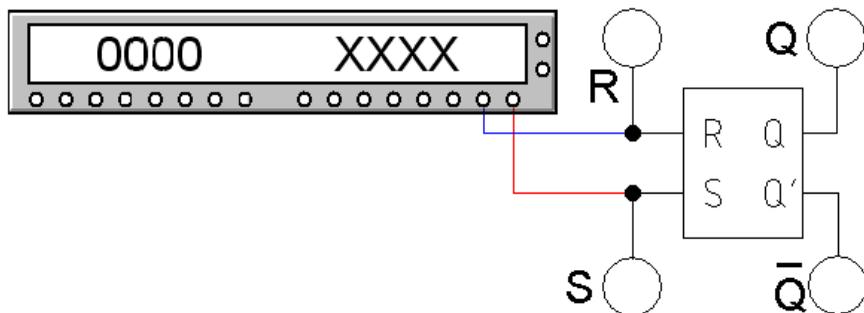


Рис. 4.5 RS-триггер из поля элементов *Digital*

### 4.2.2. Синхронные триггеры

Запись информации в триггер (рис. 4.6) происходит при наличии на входе  $C=1$ .

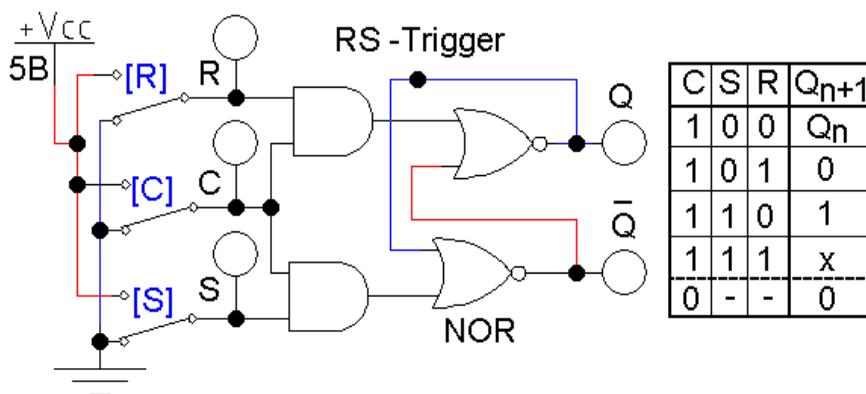


Рис. 4.6 Синхронный RS-триггер

Триггер имеет потенциальный вход  $C$ . Запись информации происходит при наличии на входе  $C$  уровня логической единицы. При этом изменение состояния триггера связано с поступлением сигналов на входы  $R, S$ .

### 4.2.3. Задание 1

1. Собрать, приведенные выше схемы и изучить их работу.
2. Выберите из базы элементов программы микросхему серии 74LS279D, которая представляет собой набор RS-триггеров. Исследуйте работу триггера с дополнительным входом установки.

### 4.3. JK-триггеры

#### 4.3.1. Основные теоретические положения

Широкое распространение получили JK-триггеры, построенные с использованием синхронных RS-триггеров с динамическим управлением. На рисунке 4.7,а приведена функциональная схема JK-триггера, переключающегося по спаду синхронизирующего импульса, а условное обозначение этого триггера на принципиальных схемах приведено на рисунке 4.7,б. Элементы DD1.1, DD1.2 образуют асинхронный RS-триггер.

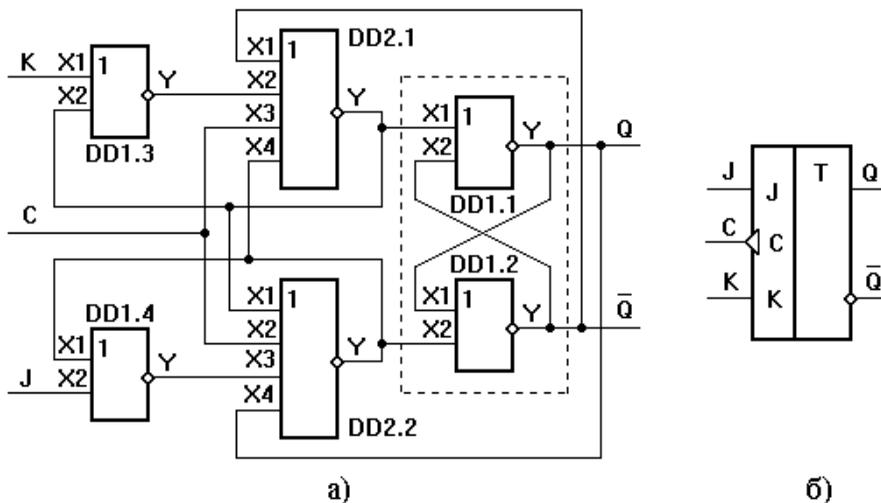


Рис. 4.7

Функциональная схема JK-триггера, переключающегося по фронту синхронизирующего импульса, показана на рисунке 8,а, а условное обозначение приведено на рисунке 8,б. При  $C=0$  на выходах элементов DD2.1 и DD2.2 логические единицы и состояние RS-триггера DD3 не изменяется. Если на инверсных входах J и K логические единицы, то переключение сигнала на входе C с логического нуля на логическую единицу не изменит состояние на выходе JK-триггера.

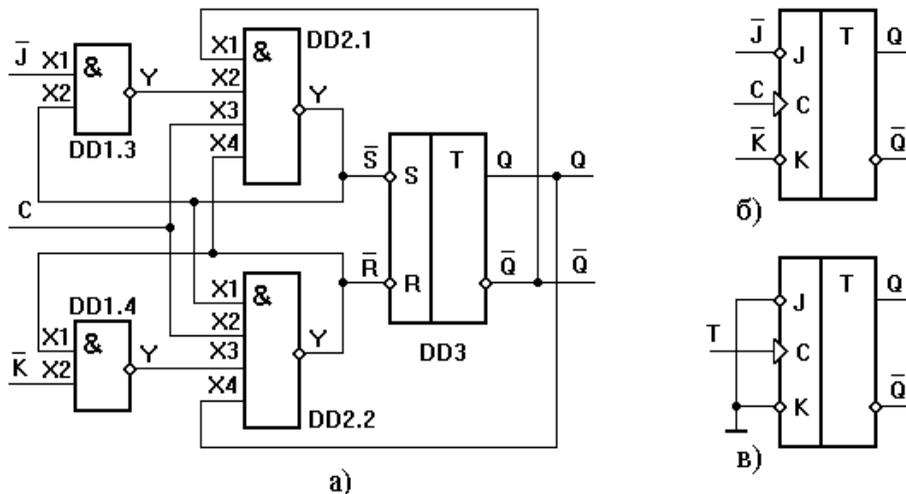


Рис. 4.8

На рисунке 4.8,в приведена схема использования JK-триггера в качестве счетного. На входы  $J$  и  $K$  подаются логические нули, а импульсы подаются на вход  $C$ . Вход  $C$  в данном случае является счетным входом  $T$ . Частота импульсов на выходе данного счетчика в два раза меньше частоты импульсов на входе. Скважность импульсов на выходе счетчика равна двум независимо от скважности импульсов на входе счетчика.

Напомним, что триггеры относятся к цифровым автоматам. Цифровые автоматы состоят из комбинационных схем. Триггеры входят в состав счетчиков электрических импульсов, регистров, запоминающих устройств.

Схема JK-триггера из набора типовых элементов программы моделирования и таблица истинности представлены на рис. 4.9.

JK-триггер имеет два информационных входа  $J$  и  $K$ , тактовый динамический вход  $C$ , и два приоритетных асинхронных входа установки  $S$  и сброса  $R$ .

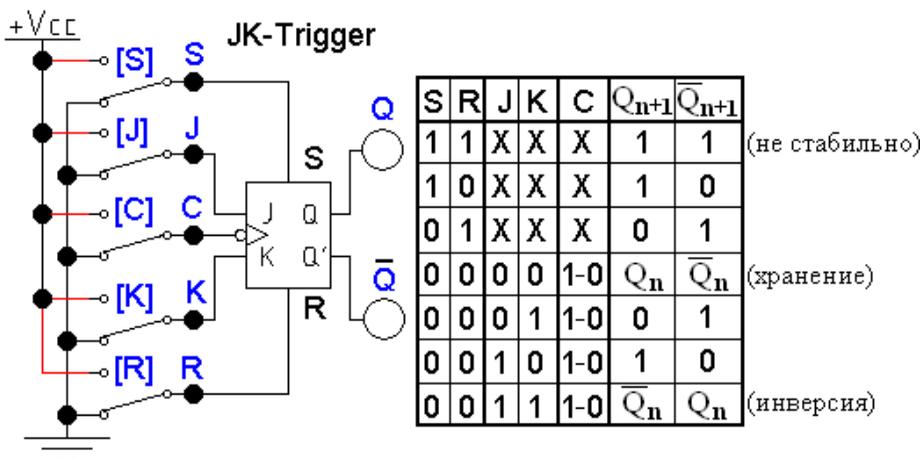


Рис. 4.9 JK-триггер из поля элементов *Digital*

Значком «X» обозначаются произвольные значения переменных  $J$ ,  $K$  и  $C$ , которые не оказывают влияния на результат вследствие приоритета входов  $R$ ,  $S$ . Этим же определяется нестабильность при  $R=S=1$ , т.к. схема работает как асинхронный  $RS$ -триггер.

триггер имеет вход установки "0" ( $R$  - вход) и вход установки "1" ( $S$  - вход).

В случае если  $R=S=0$ , асинхронный триггер переходит в режим хранения и разрешается работа синхронного динамического  $JK$ -триггера. Запись информации происходит при переходе из состояния 1 в состояние 0 сигнала на входе  $C$ .

Когда на входах  $J=K=1$ , происходит инверсия предыдущего значения на выходах триггера.

#### 4.3.2. Задание 2

1. Выполнить моделирование работы схемы, изображенной на рисунке 4.8, составить таблицу истинности.
2. Исследовать работу схемы на рисунке 4.9.
3. Выбрать из базы элементов программы моделирования микросхему серии 7473, включающую в себя два  $JK$ -триггера, смоделировать ее работу, построить таблицу истинности, как показано в таблице 4.1.
4. Сравнить таблицы истинности всех 3-х схем.

Таблица 4.1

$R'$	$J$	$K$	$C$	$Q_{n+1}$
	X	X	X	
				$Q_n$
				0
				1
				$Q_n'$

## 4.4. D-триггеры

### 4.4.1. Основные теоретические положения

В синхронных RS-триггерах со статическим управлением остается неоднозначность состояния на выходе триггера, если с входов R, S одновременно убирать активные уровни сигналов. Для устранения неоднозначности в схему синхронного RS-триггера добавляют логический элемент «НЕ». Получившийся триггер является D-триггером со статическим управлением. Функциональная схема этого триггера приведена на рисунке 4.10,а, а его условное обозначение на принципиальных схемах – на рисунке 4.10,б.

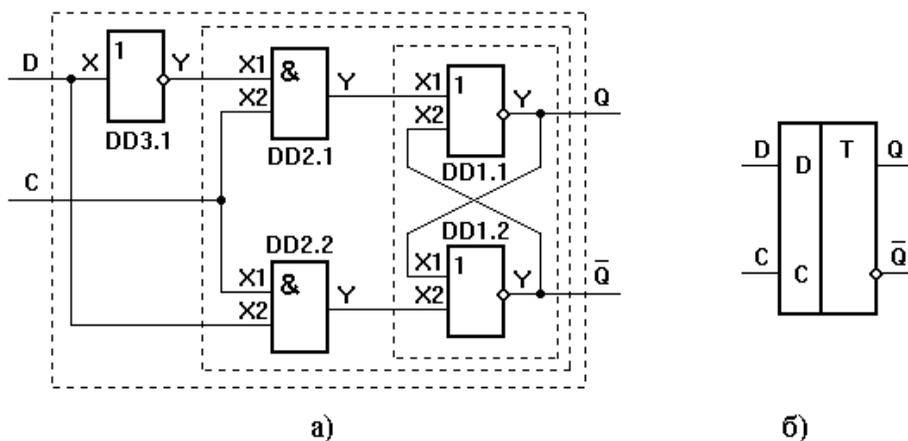


Рис. 4.10

При  $C=0$  состояние триггера изменяться не будет какой бы ни был сигнал на входе  $D$ , т.к. на выходах элементов  $DD2.1$ ,  $DD2.2$  будут сигналы логических нулей. При  $C=1$  и  $D=1$  на выходе элемента  $DD2.1$  появится сигнал логического нуля, а на прямом выходе D-триггера – сигнал логической единицы. При  $C=1$  и  $D=0$

сигнал логического нуля появится на выходе элемента DD2.2, на инверсном выходе D-триггера установится логическая единица, а на прямом выходе – логический нуль. Таким образом, D-триггер воспринимает информацию с входа D и передает ее на выход Q при C=1, и затем хранит ее сколько угодно долго (пока подключен источник питания) при C=0. Т.е. мы имеем ячейку памяти для хранения 1 бита информации.

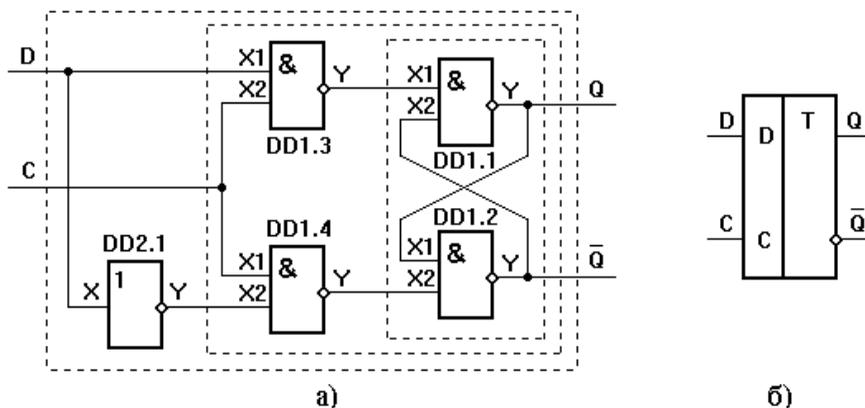


Рис. 4.11

На рисунке 4.11,а приведен еще один вариант схемы D-триггера со статическим управлением. Условное обозначение обоих этих триггеров одинаковое.

Временная диаграмма работы D-триггера со статическим управлением приведена на рисунке 4.12. Из диаграммы следует, что передача информации с входа D на выход Q осуществляется во время действия синхронизирующего импульса. Счетный триггер из данного триггера путем соединения инверсного выхода с входом D получить нельзя.

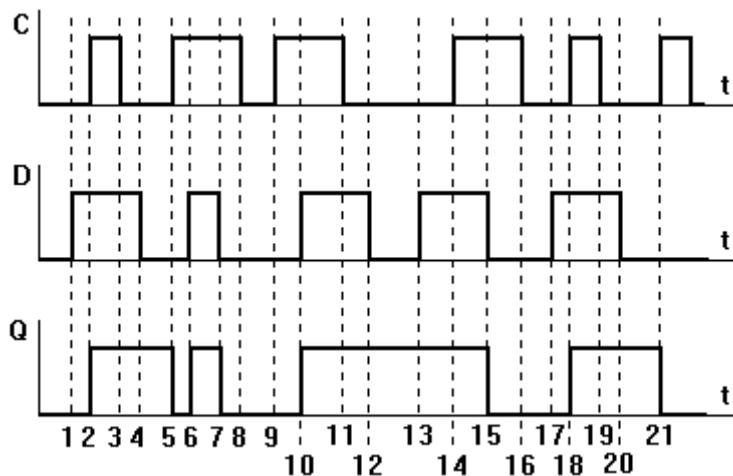


Рис. 4.12

Широко используют D-триггеры с динамическим управлением. В них передача информации с информационных входов на выходы осуществляется либо по фронту синхронизирующего импульса, либо по спаду синхронизирующего импульса. Функциональная схема D-триггера, с передачей информации с входа на выход триггера по фронту синхронизирующего импульса приведена на рисунке 4.13,а, а его условное обозначение на принципиальных схемах – на рисунке 4.13,б.

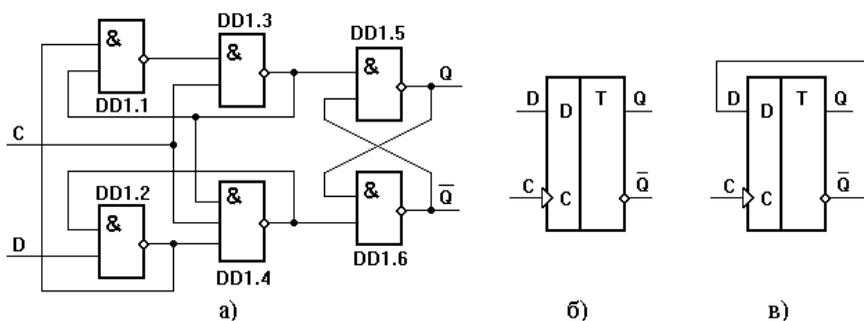


Рис. 4.13

При  $C=0$  на выходах элементов DD1.3, DD1.4 будут сигналы логических единиц и состояние на выходе триггера не изменится при любых изменениях сигнала на входе D. Установим на входе D сигнал логической единицы и изменим на входе C сигнал с логи-

ческого нуля на единицу. Перед подачей на вход С сигнала логической единицы на выходе элемента DD1.2 логический ноль, а на верхнем входе элемента DD1.3 логическая единица. При появлении на входе С логической единицы на выходе элемента DD1.3 установится логический ноль, а на прямом выходе триггера – логическая единица. Сигнал логического нуля подается с выхода элемента DD1.3 на нижний вход элемент DD1.1 и на верхний вход элемента DD1.4. Оставляя на входе С логическую единицу, изменим сигнал на входе D с логической единицы на ноль. На выходе элемента DD1.2 установится логическая единица, а сигналы на выходах элементов DD1.1, DD1.3 не изменятся, следовательно, не изменится состояние на выходе триггера.

При  $D=0$  изменим сигнал на входе С с логической единицы на ноль. На выходах элементов DD1.3, DD1.4 будут логические единицы, а на прямом выходе триггера останется сигнал логической единицы. Затем изменим сигнал на входе С с логического нуля на логическую единицу. На выходе DD1.4 установится логический ноль, на инверсном выходе триггера логическая единица, а прямом выходе – логический ноль. Из анализа работы данного триггера следует, что в нем передача информации с входа D на выход Q осуществляется по фронту синхронизирующего импульса, подаваемого на вход С.

Триггеры являются составной частью счетчиков электрических импульсов. D-триггер с динамическим управлением легко превратить в счетный триггер. С этой целью необходимо инверсный выход триггера соединить с информационным входом D, а импульсы подавать на синхронизирующий вход. Схема такого соединения приведена на рисунке 4.13,в.

Схема D-триггера имеет один информационный вход D и тактовый вход С. На рис. 4.14 изображен D-триггер из поля элементов Digital и его таблица истинности. Триггер является простейшей ячейкой памяти и применяется только как синхронный элемент.

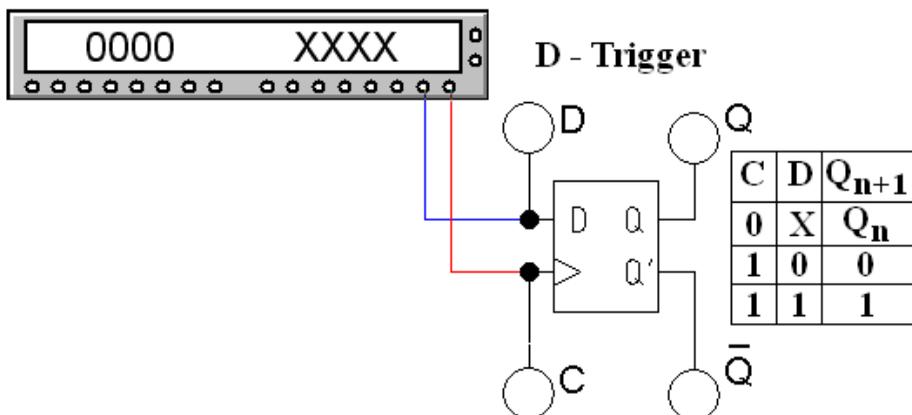


Рис. 4.14 D-триггер из поля элементов *Digital*

Значок «X» в таблице обозначает любую величину переменной, которая не оказывает влияния на результат.

На основе D-триггера можно получить другие триггеры, например, T-триггер (рис. 4.15). Данный элемент является делителем частоты входного сигнала и используется в счетчиках. Сигнал на его вход подается с источника напряжения прямоугольной формы. Имеется служебный вход EI для разрешения его работы ( $E I=1$ ).

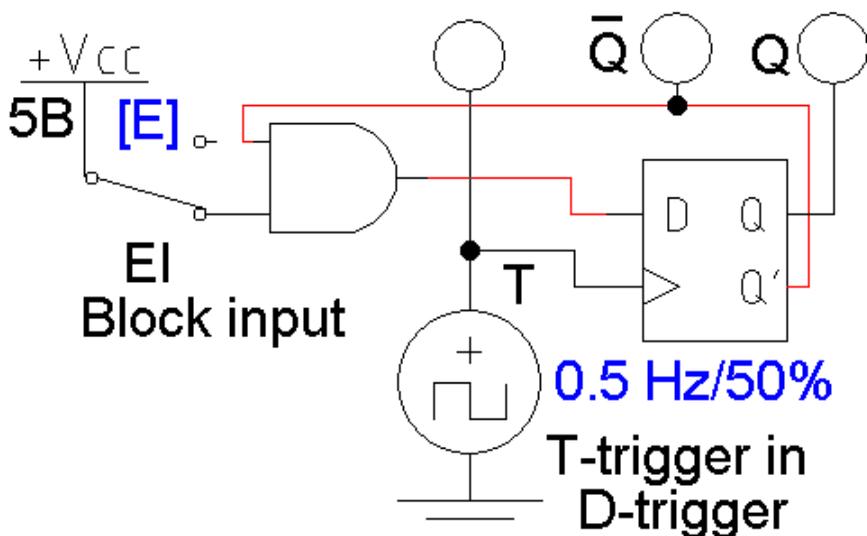


Рис. 4.15 T-триггер на основе D-триггера

Схема реализации  $D$ -триггера с помощью универсального  $JK$ -триггера представлена на рисунке 4.16.

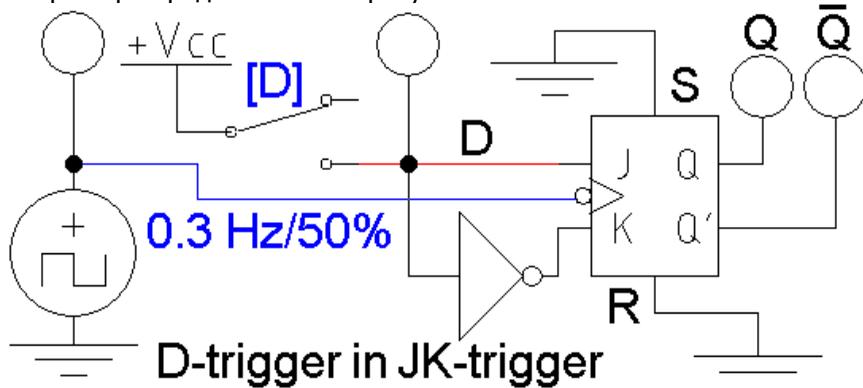


Рис. 4.16  $D$ -триггер на основе  $JK$ -триггера

### ***Принцип работы ячейки памяти***

Триггеры могут использоваться в качестве оперативной памяти. Пример ячейки памяти приведен на рис. 4.17.

Для работы с ячейкой памяти используются шины из  $n$  проводников, по которым одновременно передается  $n$  разрядов двоичного кода (параллельная передача информации).

Обычно используются:

- адресные шины, с целью указания ячейки памяти или группы ячеек;
- шины данных, по которым обмениваются информацией отдельные устройства цифрового прибора;
- шины управления, для одновременного подключения к шине данных только двух устройств, одно передающее, другое принимающее информацию.

Вначале необходимо обратиться к конкретной ячейке согласно адресу, который представляет собой номера столбца ( $X$ ) и строки ( $Y$ ). На их пересечении расположена ячейка. Данная информация в виде двоичного параллельного кода  $X=1, Y=1$  поступает по адресной шине.

Запись и считывание информации производится с шины данных ( $LD$ ). Причем, при записи выход ячейки ( $Q$ ) отключается от шины данных с помощью перевода буфера в закрытое состояние ( $F=0$ ).

Для режима записи в ячейку, выбирается значение записываемой переменной  $D$ . Эта величина считывается с шины данных, которая для упрощения схемы не соединена с линией  $LD$  и под-

ключена к источнику логической единицы через выключатель.

Только после этого, по шине управления ( $R$ ) в ячейку передается команда записывать информацию в ячейку  $R=1$  или считывать  $R=0$ .

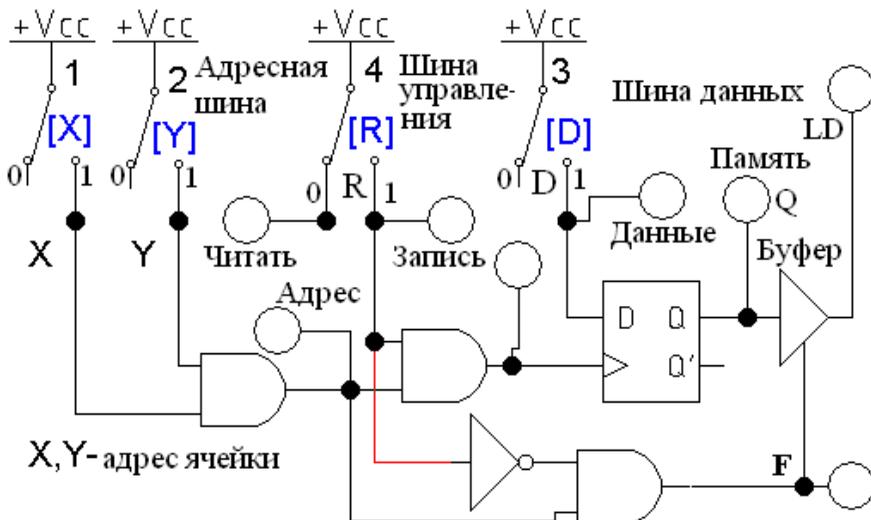


Рис. 4.17  $D$ -триггер в качестве ячейки памяти

#### 4.4.2. Задание 3

1. Построить приведенные выше схемы  $V$ -триггеров и исследовать их.

2. Заполнить таблицу истинности (табл. 4.2) для ячейки памяти, изображенной на рисунке 4.17, в соответствии с указанными операциями в столбце Примечание.

Таблица 4.2

$X$	$Y$	$R$	$D$	$Q_{n+1}$	$F$	$LD$	Примечание
							С ячейкой обмена информацией нет
							С ячейкой обмена информацией нет
							С ячейкой обмена информацией нет
							Запись в ячейку логической единицы
							Считывание из ячейки логической единицы
							Запись в ячейку логического нуля
							Считывание из ячейки логического нуля

Использовать обозначения:  $Q_n$  – предыдущее значение на выходе триггера;

$X$  – любое значение переменной, которое не оказывает влияния на результат.

3. D-триггер выпускается промышленностью в виде интегральных микросхем, например, микросхема 7474, содержащая два триггера. Найти данную микросхему в базе элементов программы и построить таблицу истинности его работы.

## 5. ЛАБОРАТОРНАЯ РАБОТА №5: ИССЛЕДОВАНИЕ РЕГИСТРОВ, СЧЕТЧИКОВ И ДЕШИФРАТОРОВ С ИСПОЛЬЗОВАНИЕМ ПРОГРАММНОГО КОМПЛЕКСА MULTISIM

### 5.1. Цель работы

**Цель работы** – исследовать регистры, счетчики и дешифраторы.

### 5.2. Краткие сведения из теории

#### 5.2.1. Регистры

Триггерным регистром называется совокупность триггеров с определенными связями между ними, при которых они действуют как единое устройство. Регистры выполняются на синхронных триггерах *JK*- или *D*-типа. В зависимости от выполняемых функций регистры делятся на накопительные (параллельные) и сдвигающие.

В последовательном регистре выход предыдущего триггера подается на вход следующего триггера, а тактовые импульсы подаются на входы *C* всех триггеров, составляющих регистр, одновременно (рис. 5.1). При этом содержимое каждого триггера записывается в последующий триггер. Такие регистры называются сдвиговыми регистрами, или регистрами сдвига.

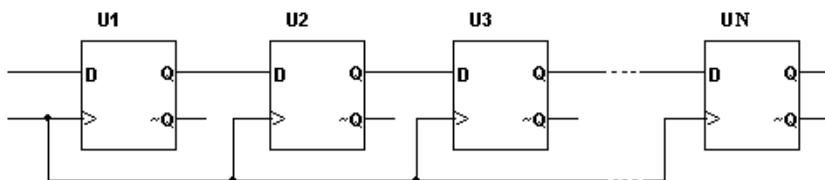


Рис. 5.1. Последовательный регистр (регистр сдвига)

Если на вход *D* регистра сдвига подать единицу, а на вход *C* тактовую частоту, то единица начнет продвигаться по регистру

сдвига, т.е. под воздействием первого тактового импульса единица запишется в первый триггер регистра. Под воздействием второго тактового импульса эта единица переписывается во второй триггер и т.д., когда под воздействием  $N$ -го тактового импульса единица не выйдет из регистра сдвига. Временная диаграмма работы четырехразрядного регистра сдвига приведена на рис. 5.2.

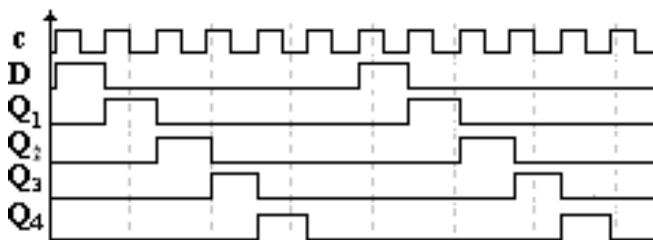


Рис. 5.2. Временная диаграмма четырехразрядного регистра сдвига

Виды регистров сдвига:

- с последовательным вводом и выводом;
- с последовательным вводом и параллельным выводом;
- с параллельным вводом и последовательным выводом;
- с переменным направлением сдвига (реверсивные регистры сдвига).

Кроме последовательных регистров сдвига существуют параллельные регистры, в которых информация подается одновременно на все  $N$  триггеров и считывается одновременно с выходов всех триггеров регистра (рис. 5.3). Тактовая частота подается одновременно на все триггеры.

Параллельные регистры используются для хранения двоичной информации небольшого объема в течение короткого промежутка времени.

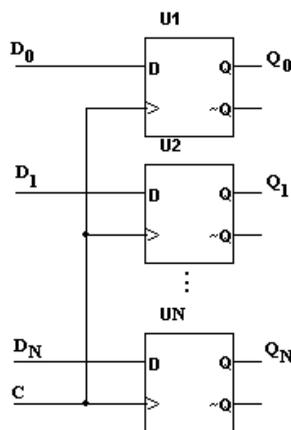


Рис.5.3. Параллельный регистр

### 5.2.2. Счетчики

Счетчиком называется устройство, подсчитывающее число входных импульсов. Число, представляемое состоянием его выходов по фронту каждого входного импульса, изменяется на единицу. Счетчик состоит из  $n$  последовательно соединенных счетных триггеров, причем выход одного счетного триггера соединен с тактовым входом следующего триггера. Счетчики бывают суммирующими (прямой счет) и вычитающими (обратный счет). В суммирующих счетчиках каждый входной импульс увеличивает число на его выходах на единицу, в вычитающих счетчиках уменьшает это число на единицу. Для того чтобы построить суммирующий счетчик, необходимо счетный вход очередного триггера подключить к инверсному выходу предыдущего (рис. 5.4).

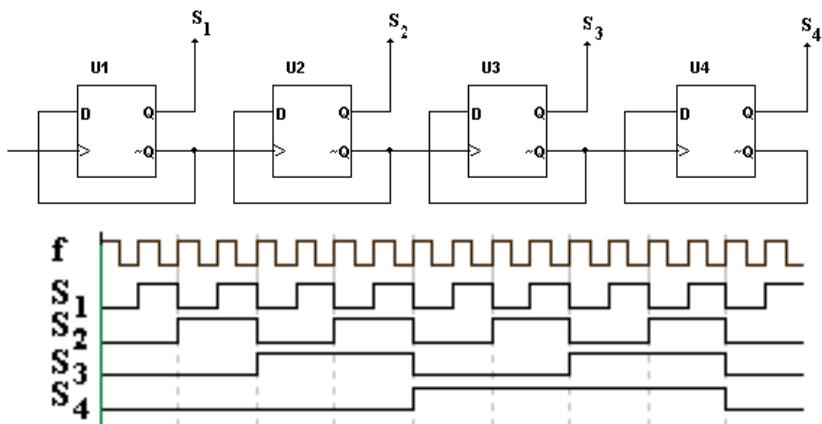


Рис. 5.4. Суммирующий счетчик и диаграмма его работы

Для того чтобы изменить направление счета (реализовать вычитающий счетчик), необходимо счетный вход очередного триггера подключить к прямому выходу предыдущего, при этом изменяется последовательность переключения триггеров (рис. 5.5).

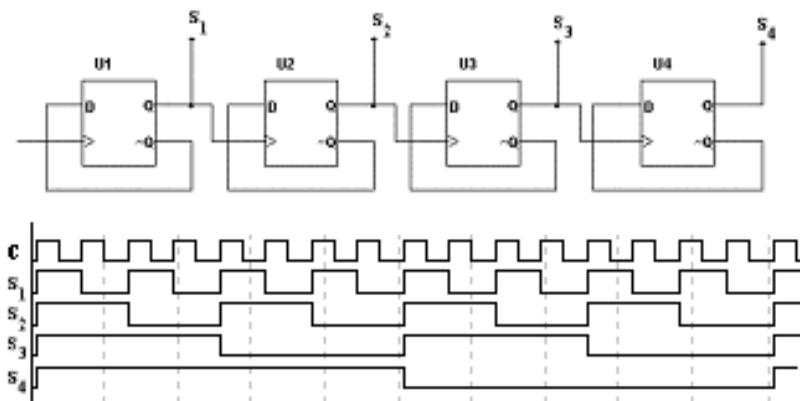


Рис. 5.5. Вычитающий счетчик и диаграмма его работы

Счетчик характеризуется числом состояний в течение одного периода (цикла). Для двоичных счетчиков полный цикл счета  $N=2^n$  от состояния 0...000 до состояния 1...11. Число состояний называется коэффициентом пересчета  $K_{сч}$ , равным отношению числа импульсов  $N_c$  на входе к числу импульсов  $N_{QCT}$  на выходе старшего разряда за период:

$$K_{сч} = \frac{N_c}{N_{QCT}}.$$

Если на вход счетчика подавать периодическую последовательность импульсов с частотой  $f_c$ , то частота  $f_Q$  на выходе старшего разряда счетчика будет меньше в  $K_{сч}$  раз:  $K_{сч} = f_c / f_Q$ . Поэтому счетчики также называют делителями частоты, а  $K_{сч}$  – коэффициентом деления. Для увеличения величины  $K_{сч}$  нужно увеличить число триггеров в цепочке. Каждый дополнительный триггер удваивает число состояний счетчика и число  $K_{сч}$ .

### 5.2.3. Дешифраторы

Комбинационной называется логическая схема, реализующая однозначное соответствие между значениями входных и выходных сигналов. Дешифратор – логическая комбинационная схема, имеющая  $n$  информационных входов и  $2^n$  выходов. Каждой комбинации логических уровней на входах будет соответствовать активный уровень на одном из  $2^n$  выходов. Как любая логическая схема, дешифратор может быть задан таблицей истинности. Таблица истинности дешифратора  $3 \times 8$  (табл. 5.1) состоит из трех столбцов, соответствующих входным сигналам  $X_0, X_1, X_2$ , и восьми

## Основы микропроцессорной техники

столбцов, соответствующих выходным сигналам  $Y_0, Y_1, Y_2, Y_3, Y_4, Y_5, Y_6, Y_7$ . В первых слева трех столбцах расположены возможные комбинации входных сигналов, а в последних восьми – соответствующие им комбинации выходных сигналов.

Таблица 5.1

$X_2$	$X_1$	$X_0$	$Y_7$	$Y_6$	$Y_5$	$Y_4$	$Y_3$	$Y_2$	$Y_1$	$Y_0$
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

Схема имеет восемь выходов, на одном из которых потенциал высокий, на остальных низкий. Номер единственного выхода, имеющего высокий потенциал, соответствует двоичному числу, формируемому состояниями входных сигналов. Этот принцип формирования выходного сигнала можно описать следующим образом:  $Y_i = 0$ , если  $i = k$ ;  $Y_i = 1$ , если  $i \neq k$ , здесь  $i$  – номер разряда;  $k = 2^2 X_2 + 2^1 X_1 + 2^0 X_0$ .

Выражения для каждого выхода дешифратора:

$$\begin{aligned} \bar{Y}_0 &= \bar{X}_2 \bar{X}_1 \bar{X}_0, & Y_4 &= X_2 \bar{X}_1 \bar{X}_0, \\ \bar{Y}_1 &= \bar{X}_2 \bar{X}_1 X_0, & Y_5 &= X_2 \bar{X}_1 X_0, \\ \bar{Y}_2 &= \bar{X}_2 X_1 \bar{X}_0, & Y_6 &= \bar{X}_2 X_1 X_0, \\ Y_3 &= X_2 X_1 \bar{X}_0, & Y_7 &= X_2 X_1 X_0, \end{aligned}$$

где « $\bar{\phantom{x}}$ » – инвертирование.

Таким образом, схема дешифратора должна содержать три схемы «НЕ» и восемь схем «ЗИ» (рис. 5.6).

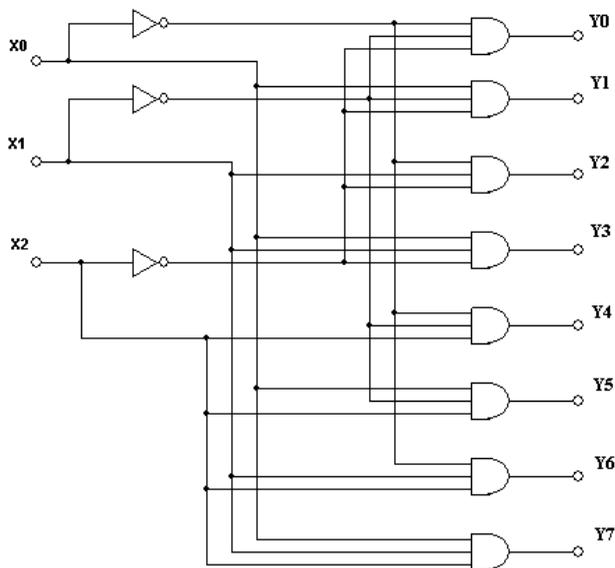


Рис. 5.6. Схема дешифратора 3x8

### 5.3. Последовательность выполнения работы

#### 5.3.1. Исследование регистра

1. Нарисовать схему четырехразрядного регистра сдвига с автоматической записью единицы в первый разряд регистра (рис. 5.7).

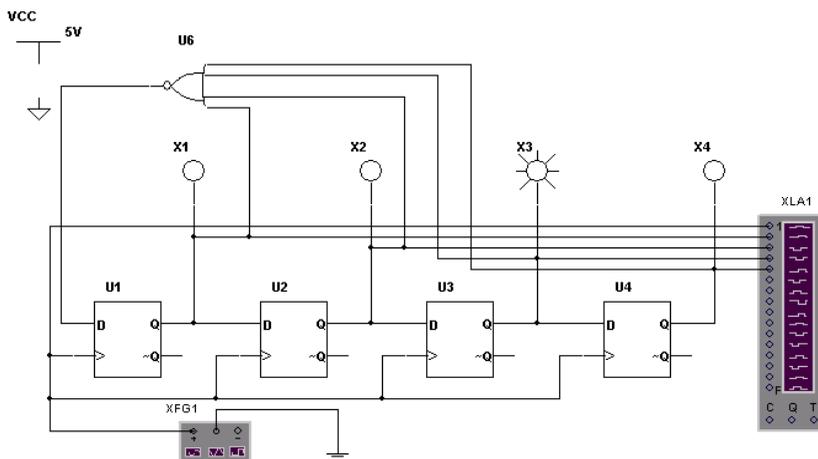


Рис. 5.7. Четырехразрядный регистр сдвига с автоматической записью единицы

Схема содержит четыре *D*-триггера, четыре светодиода, одну логическую схему «ИЛИ-НЕ», функциональный генератор и логический анализатор. Логическая схема «ИЛИ-НЕ» служит для автоматической записи единицы в регистр. На выходе этой схемы единица будет только тогда, когда все разряды регистра будут находиться в нулевом состоянии.

2. Открыть окно функционального генератора и установить вид генерируемых сигналов (прямоугольные импульсы), генерируемую частоту 1000 Гц, амплитуду генерируемых импульсов 5 В.

3. Открыть окно логического анализатора, дважды щелкнув по иконке логического анализатора.

4. Запустить процесс моделирования, нажав кнопку  на панели инструментов, и в появившемся меню выбрать команду *Run*.

5. Зарисовать диаграммы сигналов с экрана логического анализатора.

6. Зарисовать все состояния светодиодов и сравнить их с временными диаграммами.

## 5.4. Описание используемых контрольно-измерительных приборов

### 5.4.1. Функциональный генератор (Function Generator)

На лицевой панели функционального генератора (рис 5.8)

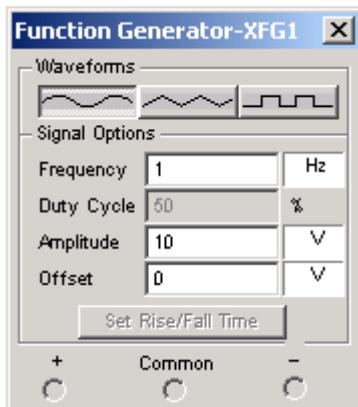


Рис. 5.8. Лицевая панель функционального генератора

расположены три кнопки

, с помощью которых выбирается форма генерируемого сигнала: синусоидальной (выбирается по умолчанию), треугольной и прямоугольной.

В четырех белых окнах устанавливают параметры выходного сигнала:

- частоты – ,
- коэффициента заполнения в процентах. Для импульсных сигналов это отношение длительности импульса к периоду повторения (величина обрат-

ная скважности), для треугольных сигналов – соотношение между длительностями переднего и заднего фронтов –

Duty Cycle  %

• амплитуды выходного сигнала –  V

• смещения (постоянной составляющей) –  V

В низу лицевой панели расположены выходные зажимы  Common ; при заземлении клеммы COM (общий) на клеммах «+» и «-» генератор выдает парафазный сигнал.

### 5.4.2. Логический анализатор (Logic Analyzer)

Логический анализатор предназначен для отображения на экране монитора 16-разрядных кодовых последовательностей одновременно в шестнадцать точек схемы, а также в виде двоичных чисел на входных клеммах-индикаторах (рис. 5.9).

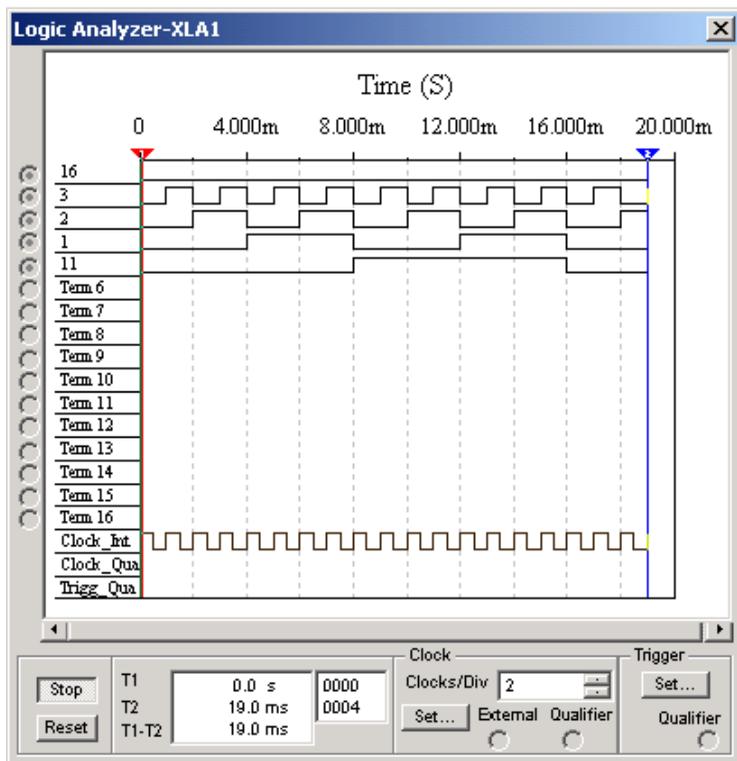


Рис. 5.9. Лицевая панель логического анализатора

Анализатор снабжен двумя визирными линейками для точных отсчетов временных интервалах  $T_1$ ,  $T_2$  и  $T_2 - T_1$  (окно в поле  $T_1$ ,  $T_2$ ,  $T_2 - T_1$ ), а также линейкой прокрутки по горизонтали для просмотра осциллограмм на большом временном интервале.

На панели *Clock* в строке *Clocks/Div* задается длительность развертки. В этом блоке имеются клеммы для подключения как обычного (*Extend*), так и избирательного (*Qualifier*) источника запускающих сигналов, параметры которых устанавливаются с помощью меню, вызываемого кнопкой *Set*.

В окне *Trigger* можно задать логические сигналы 1, 0 и  $X$ , при наличии которых производится запуск анализатора.

Кнопки *Stop* и *Reset* предназначены для остановки и перезапуска режима моделирования.

### 5.4.3. Исследование суммирующего счетчика

1. Нарисовать схему четырехразрядного счетчика, считающего в прямом направлении (рис. 5.10).

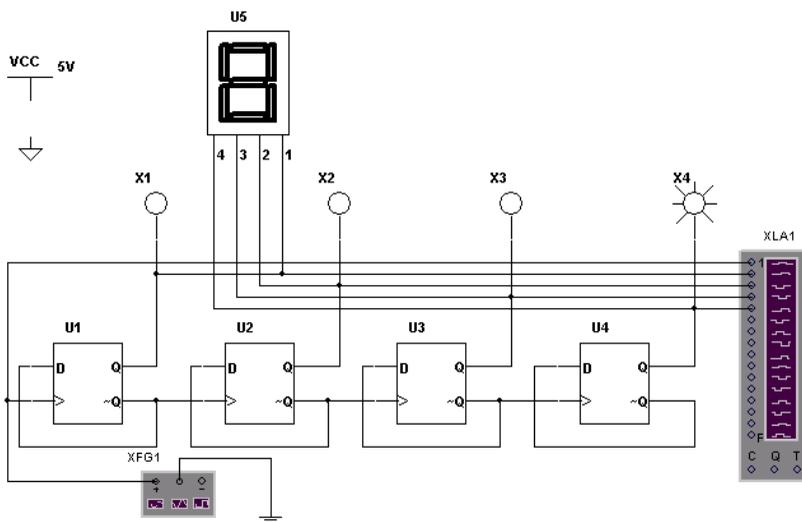


Рис. 5.10. Четырехразрядный суммирующий счетчик

Поместить на схему четыре *D*-триггера, четыре светодиода, функциональный генератор, логический анализатор и 16-ричный индикатор.

2. Открыть окно логического анализатора, щелкнув по иконке логического анализатора.

3. Запустить процесс моделирования, нажав кнопку



на панели инструментов, и в появившемся меню выбрать команду *Run*.

4. Зарисовать диаграммы сигналов с экрана логического анализатора.

5. Зарисовать все состояния светодиодов и сравнить их с временными диаграммами.

6. Наблюдать за показаниями 16-ричного индикатора и сравнить его показания с соответствующими состояниями светодиодов.

#### 5.4.4. Исследование вычитающего счетчика

1. Нарисовать схему четырехразрядного счетчика, считающего в обратном направлении (рис. 5.11).

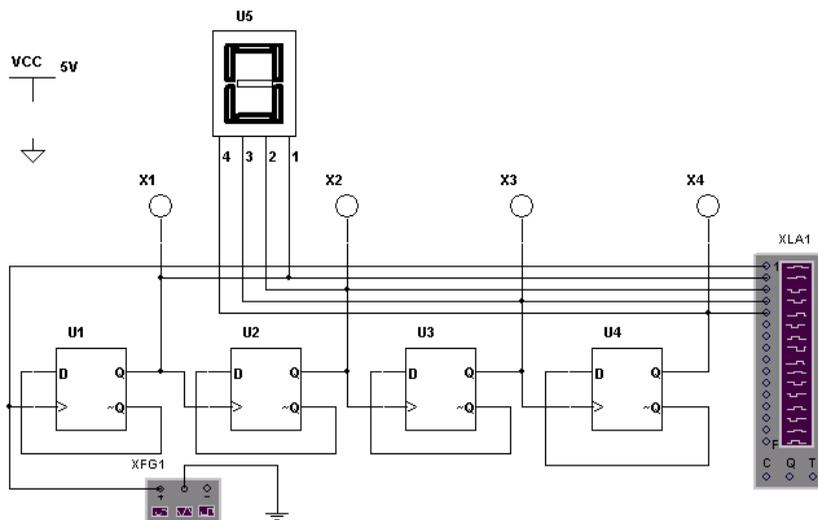


Рис. 5.11. Четырехразрядный вычитающий счетчик

2. Открыть окно логического анализатора, щелкнув по иконке логического анализатора.

3. Запустить процесс моделирования, нажав кнопку  на панели инструментов, и в появившемся меню выбрать команду *Run*.

4. Зарисовать диаграммы сигналов с экрана логического анализатора.

5. Зарисовать все состояния светодиодов и сравнить их с

временными диаграммами.

6. Наблюдать за показаниями 16-ричного индикатора и сравнить его показания с соответствующими состояниями светодиодов.

### 5.4.5. Исследование дешифратора

1. Нарисовать схему дешифратора 3x8 (рис.5.12). Для этого поместить на схему восемь логических схем «ЗИ», восемь светодиодов, пять логических схем «НЕ», три переключателя на два направления и три вольтметра.

2. Присвоить переключателям управляющие клавиши.

3. С помощью переключателей подать на вход дешифратора все возможные комбинации сигналов и записать для каждого входного сигнала выходной сигнал (восьмиразрядную комбинацию), т.е. заполнить таблицу истинности.

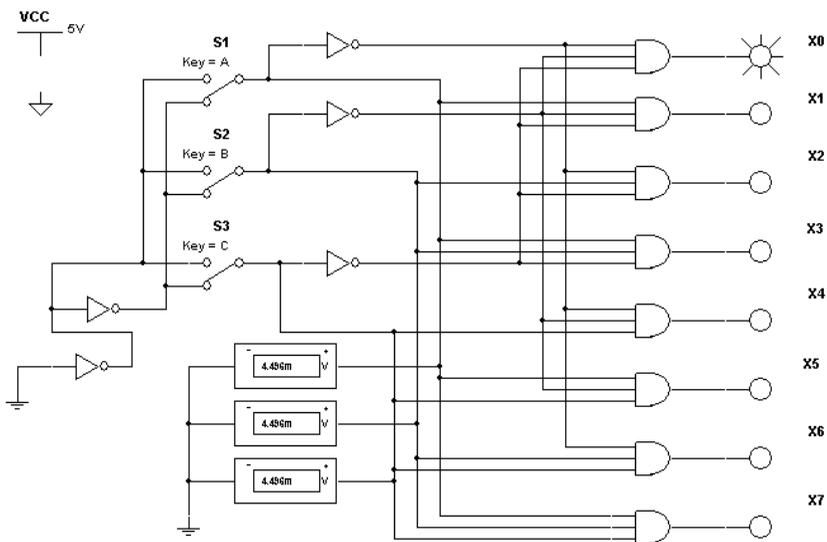


Рис. 5.12. Дешифратор 3x8

### 5.5. Задания для самостоятельного выполнения

Изобразить в программе схему и исследовать работу следующих систем (по вариантам):

## Основы микропроцессорной техники

Таблица 5.1 – Варианты заданий

№ варианта	Регистр	Счетчик	Дешифра-тор	Шифра-тор
1	Параллельный 3- разр.	74LS90 (IE2)	4 на 16	8 в 3
2	74LS165 (IP9)	до 8	4 на 10	9 в 4
3	Параллельный 4- разр.	до 9	74LS141 (ИД1)	10 в 4
4	Параллельный 5- разр.	74LS92 (IE4)	3 на 8	16 в 4
5	74LS95 (IP1)	до10	5 на 24	20 в 5
6	74LS194 (IP11)	до 11	5 на 28	24 в 5
7	Параллельный 6- разр.	74LS168 (IE16)	5 на 20	28 в 5
8	Параллельный 7- разр.	до 12	74LS154 (ИД3)	32 в 5
9	Параллельный 8- разр.	74LS93 (IE5)	6 на 42	40 в 6
10	Последовательн. 6- разр.	до 13	3 на 6	74LS148 (ИБ1)
11	Последовательн. 7- разр.	74LS169 (IE12)	6 на 40	42 в 6
12	Последовательн. 8- разр.	до 14	74LS138 (ИД7)	44 в 6
13	74LS164 (IP8)	до 15	6 на 34	34 в 6
14	Последовательн. 5- разр.	74LS192 (IE6)	4 на 12	36 в 6
15	Последовательн. 4- разр.	до 16	6 на 36	74LS147 (ИБ3)
16	Кольцевой 6- разр.	до 17	74LS145 (ИД10)	12 в 4
17	Кольцевой 7- разр.	74LS169 (IE17)	5 на 32	6 в 3
18	Кольцевой 8- разр.	до 18	6 на 64	48 в 6
19	74LS195 (IP12)	до 19	4 на 9	50 в 6
20	Кольцевой 4- разр.	до 20	74LS139 (ИД14)	54 в 6
21	Кольцевой 5- разр.	74LS193 (IE7)	3 на 8	64 в 6
22	Последовательн. 9- разр.	до 21	6 на 50	8 в 3
23	Последовател. 11- разр.	до 22	6 на 48	9 в 4
24	Последовател. 10- разр.	до 23	72LS42 (ИД6)	10 в 4
25	Кольцевой 9- разр.	74LS97 (IE8)	6 на 40	16 в 4
26	Кольцевой 10- разр.	до 24	6 на 42	20 в 5
27	74LS198 (IP13)	до 25	6 на 44	24 в 5
28	74LS295 (IP16)	до 26	5 на 32	28 в 5
29	Параллельный 11- разр.	74LS160 (IE9)	5 на 24	32 в 5
30	Параллельный 9- разр.	до 27	5 на 20	40 в 6
31	74LS173 (IP15)	до 28	5 на 18	42 в 6
32	Параллельный 10- разр.	74LS161 (IE10)	4 на 14	44 в 6

Примечание:

1. Не важно, с использованием каких элементов будет реализована схема (от логических до отдельных типовых схем)

2. Если в базе программы отсутствует данная микросхема, необходимо найти ее аналог из других серий микросхем

3. Раздел, посвященный шифраторам изучается самостоятельно.
4. Большинство шифраторов и дешифраторов будут не полными.

### 5.6. Содержание отчета

Отчет должен включать:

- 1) название пункта работы,
- 2) исследуемые схемы,
- 3) результаты моделирования,
- 4) краткие выводы по каждому пункту.

## 6. ЛАБОРАТОРНАЯ РАБОТА №6: МУЛЬТИПЛЕКСОРЫ И ДЕМУЛЬТИПЛЕКСОРЫ

### 6.1. Мультиплексоры

#### 6.1.1. Краткие сведения из теории

В цифровой технике широко применяются мультиплексоры и демультиплексоры. Мультиплексор это устройство, обеспечивающее соединение одного из информационных входов с выходом. Номер информационного входа, который соединяется с выходом, задается в двоичном коде на адресных входах. Если мультиплексор имеет  $n$  адресных входов, то в нем может быть  $2^n$  информационных входов. Демультиплексор это устройство, обеспечивающее соединение одного из информационных выходов с одним входом. Номер информационного выхода, который соединяется с входом, задается в двоичном коде на адресных входах. Если демультиплексор имеет  $n$  адресных входов, то в нем может быть  $2^n$  информационных выходов.

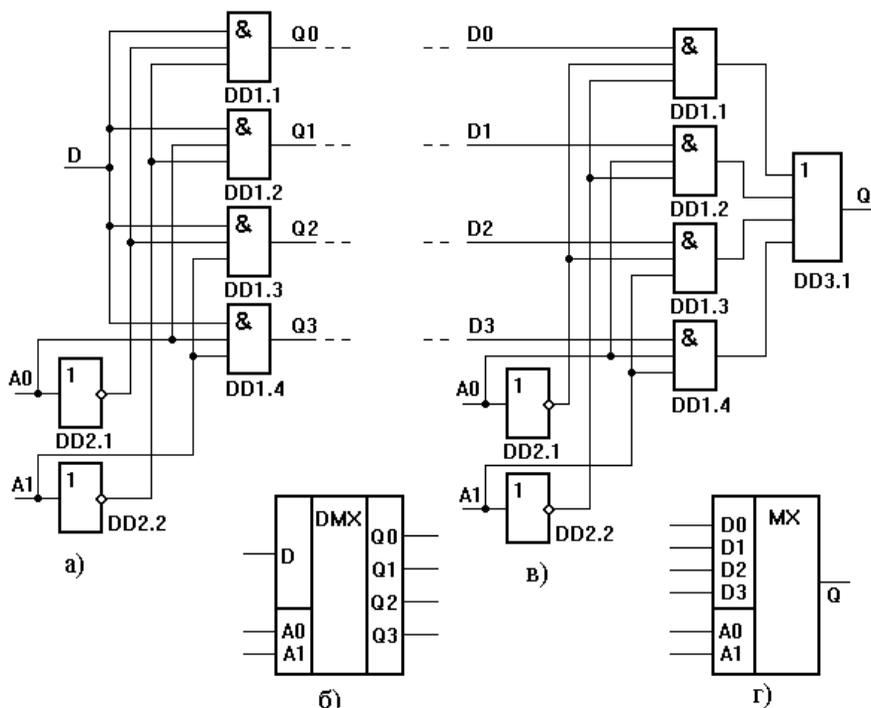


Рисунок 6.1

Функциональная схема демультиплексора, имеющего четыре выхода, приведена на рисунке 6.1,а, а его условное обозначение на принципиальных схемах – на рисунке 6.1,б.

Функциональная схема мультиплексора, имеющего четыре входа, приведена на рисунке 6.1,в, а его условное обозначение на принципиальных схемах – на рисунке 6.1,г. Мультиплексоры могут снабжаться дополнительным входом – входом разрешения передачи информации с входов на выход.

Мультиплексор – комбинационное цифровое устройство для управляемой передачи данных от нескольких источников информации  $X$  в один выход  $F$  согласно коду адреса  $A$ .

Мультиплексор имеет  $n$  адресных шин,  $m = 2^n$  – входов и один информационный выход  $F$ . Обозначение  $MUX(m-1)$ .

Например,  $MUX(4-1)$ : в зависимости от кода, подаваемого в адресные шины  $A_0, A_1$  один из информационных входов  $X_0...X_3$  подключается к выходному каналу  $F$  (рис.6.2).

Основы микропроцессорной техники

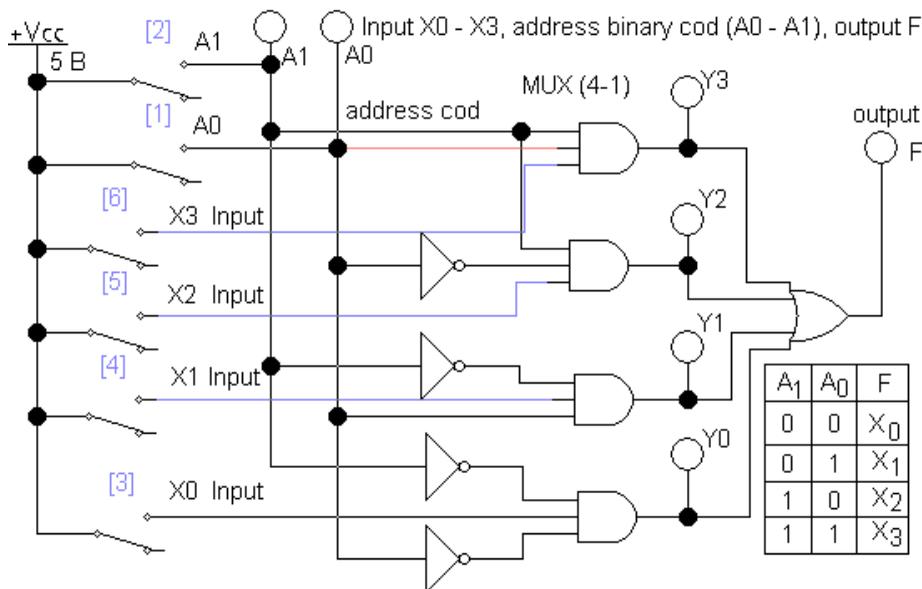


Рис. 6.2 Мультиплексор  $MUX(4-1)$

На рис. 6.3 представлен  $MUX(8-1)$ , входы  $X_0, X_1 \dots X_7$ , адреса  $A_0, A_1, A_2$ , выход  $F$ .

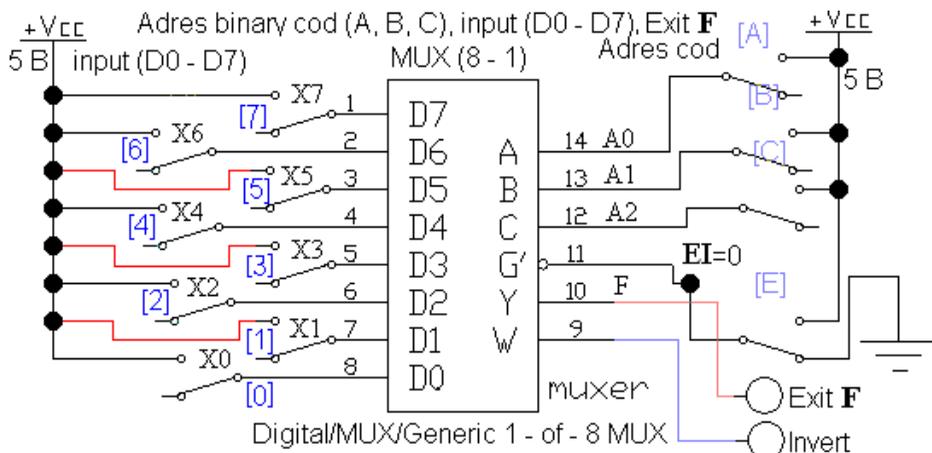


Рис. 6.3 Мультиплексор  $MUX(8-1)$

Функция алгебры логики, описывающая работу мультиплексора (рис. 2), имеет вид:

$$F = \overline{A_1} \cdot \overline{A_0} \cdot X_0 + \overline{A_1} A_0 X_1 + A_1 \overline{A_0} X_2 + A_1 A_0 X_3.$$

Мультиплексоры имеют разрешающий вход  $EI$ , если на него подается логическая единица, то происходит передача информации. На рисунке 6.3 этот вход инверсный.

На мультиплексорах можно реализовать любую логическую функцию. Например, необходимо получить логический элемент ЗИ (рис. 6.4). Для этого составляется таблица истинности (табл. 6.1).

Таблица 6.1 - Таблица истинности элемента «И»

№ входа $X$	$A_2$	$A_1$	$A_0$	$F$
$X_0$	0	0	0	0
$X_1$	0	0	1	0
$X_2$	0	1	0	0
$X_3$	0	1	1	0
$X_4$	1	0	0	0
$X_5$	1	0	1	0
$X_6$	1	1	0	0
$X_7$	1	1	1	1

В соответствии с адресом, выход  $F$  подключается к соответствующему входу  $X$ . Если на эти входы подать логические сигналы согласно столбцу  $F$ , то получится требуемый элемент (рис. 6.4).

Используя табл. 6.1, согласно стандартным процедурам, можно записать аналитическое выражение работы мультиплексора:

$$F = \overline{A_2} \cdot \overline{A_1} \cdot \overline{A_0} \cdot X_0 + \overline{A_2} \cdot \overline{A_1} A_0 X_1 + \overline{A_2} A_1 \overline{A_0} X_2 + \overline{A_2} A_1 A_0 X_3 + A_2 \overline{A_1} \cdot \overline{A_0} X_4 + A_2 \overline{A_1} A_0 X_5 + A_2 A_1 \overline{A_0} X_6 + A_2 A_1 A_0 X_7.$$

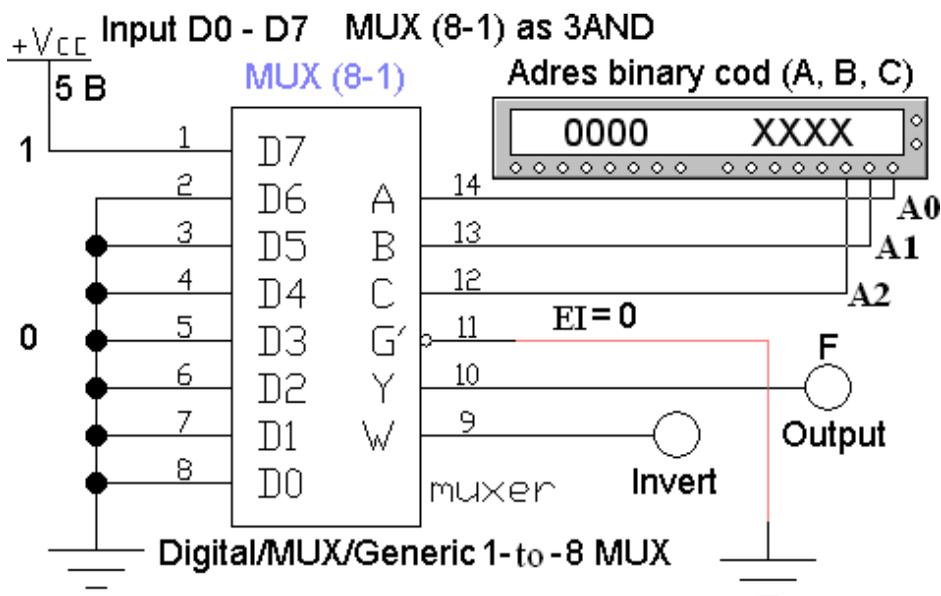


Рис. 6.4 Логический элемент 3И на мультиплексоре  $MUX(8-1)$

**Пример.** Реализовать функцию  $F$  на  $MUX(4-1)$ :

$$F = D_0 D_1 D_2 + D_0 \overline{D_1} \cdot \overline{D_2} + \overline{D_0} D_1 \overline{D_2} + \overline{D_0} \cdot \overline{D_1} \cdot \overline{D_2}$$

Последовательность операций:

Адресных входов меньше, чем переменных  $X$ . Поскольку  $D_0$  и  $D_1$  входят во все слагаемые группы сомножителей в прямом и инверсном виде, примем:  $D_0 = A_0$ ;  $D_1 = A_1$ .

Составим карту Карно (табл. 6.2), используя аналитическое выражение для определения значений функции  $F$  в клетках данной таблицы. Наборам адресов  $A_0$ ;  $A_1$  соответствуют входы мультиплексора  $X$ , которые подключаются к выходу  $F$  (таблицы 6.2, 6.3).

Из таблицы 6.2 следует, что на первых трех наборах адресов, функция  $F$  равна инверсному значению  $D_2$ . Для четвертого набора  $F = D_2$ . Схема устройства представлена на рис. 6.5. Выход микросхемы 74352 инверсный, поэтому к нему подключен инвертор.

Таблица 2 – Карта Карно

$D_2D_0$	00	01	11	10
$D_1$				
0	1	1	0	1
1	0	0	1	0
Информационные Входы	$X_0$	$X_1$	$X_3$	$X_2$

Таблица 3 – Таблица истинности

№	$D$	$A_1$	$A_0$	$F$
1	$D_2'$	0	0	$D_0$
2	$D_2'$	0	1	$D_1$
3	$D_2'$	1	0	$D_2$
4	$D_2$	1	1	$D_3$

$X_2'$  – инверсное значение  $X_2$

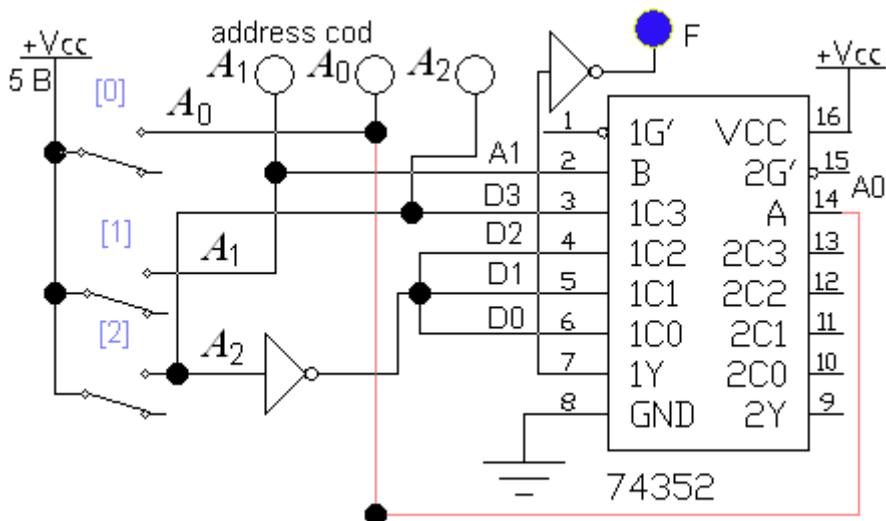


Рис. 6.5 Реализация логической функции на мультиплексоре  $MUX(4-1)$

Для увеличения разрядности мультиплексора используются разрешающие входы  $EI$ .

На рис. 6.6 представлен  $MUX(8-1)$  на двух  $MUX(4-1)$  и одном  $MUX(2-1)$ .

Мультиплексоры расположены в модулях 1, 2, 3, которые раскрываются двойным щелчком мыши.

Переключение между мультиплексорами (4-1) осуществляется с помощью старшего адресного разряда  $A_2$ , который управляет работой  $MUX(2-1)$ : первые четыре набора для  $X_0...X_3$  считываются с  $MUX1$ , когда  $A_2=0$ , наборы  $X_4...X_7$  с  $MUX2$ , при  $A_2=1$ .

Основы микропроцессорной техники

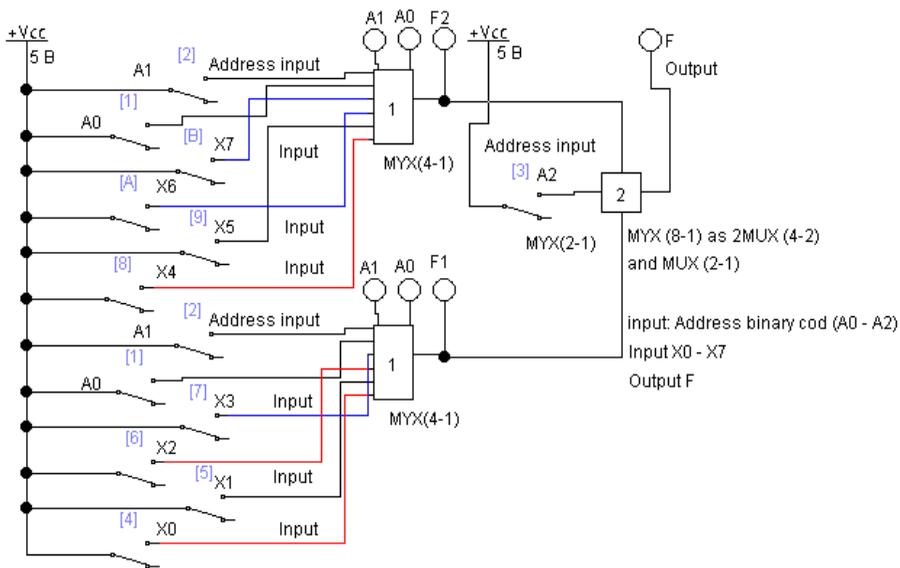


Рис. 6.6 Мультиплексор  $MUX(8-1)$  на двух  $MUX(4-1)$  и одном  $MUX(2-1)$

**Задание 1.** Задавая комбинации входных сигналов, изучить работу мультиплексоров (рис. 6.2...6.6).

**Задание 2.** Составить таблицу истинности (табл. 6.4)  $MUX(8-1)$ , рис. 6.3, где  $A_0$  управляется кнопкой А,  $A_1$  – кнопкой В,  $A_2$  – кнопкой С.

Таблица 6.4 - Таблица истинности  $MUX(8-1)$

$EI$	$A_2$	$A_1$	$A_0$	Номера входов X на выходе F
	0	0	0	
	0	0	1	
	0	1	0	
	0	1	1	
	1	0	0	
	1	0	1	
	1	1	0	
	1	1	1	
	1	1	1	

## 6.2. Демультимплексоры

### 6.2.1. Основные теоретические положения

Демультимплексором называется комбинационное логическое устройство, предназначенное для управляемой передачи данных от одного источника  $X$  информации в несколько выходных каналов выходов  $Y$  согласно коду адреса  $A$ . Демультимплексор имеет один информационный вход,  $n$  адресных шин и  $m = 2^n$  – выходов. Обозначение  $CDX(1- m)$ .

Для схемы  $DMX(1-2)$ , рис. 6.7:

$$Y_0 = \overline{A_0} \cdot X; \quad Y_1 = A_0 \cdot X.$$

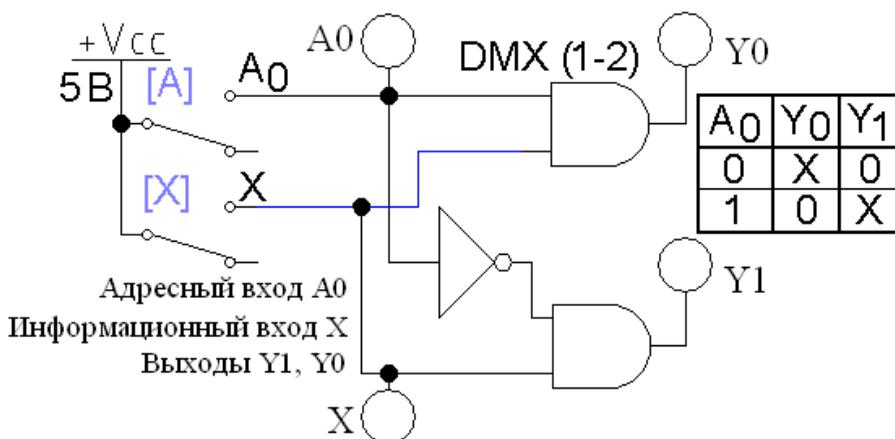


Рис. 6.7 Демультимплексор  $DMX(1-2)$

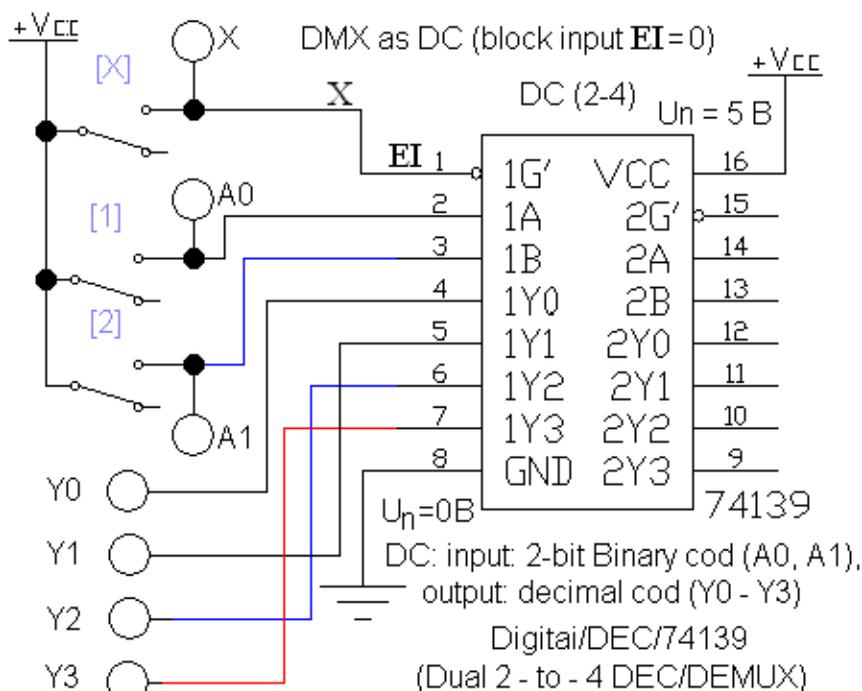


Рис. 6.8 Дешифратор – демультиплексор (2-4)

Дешифратор – демультиплексор (рис. 6.8) в случае, если выключатель  $X$  постоянно подключен к логическому нулю, работает как дешифратор. Когда с помощью данного выключателя подается двоичный код на информационный вход  $X$ , устройство работает как демультиплексор. В качестве входа  $X$  используется служебный вход  $EI$ .

Применение демультиплексора как дешифратора представлено на рис. 6.9.

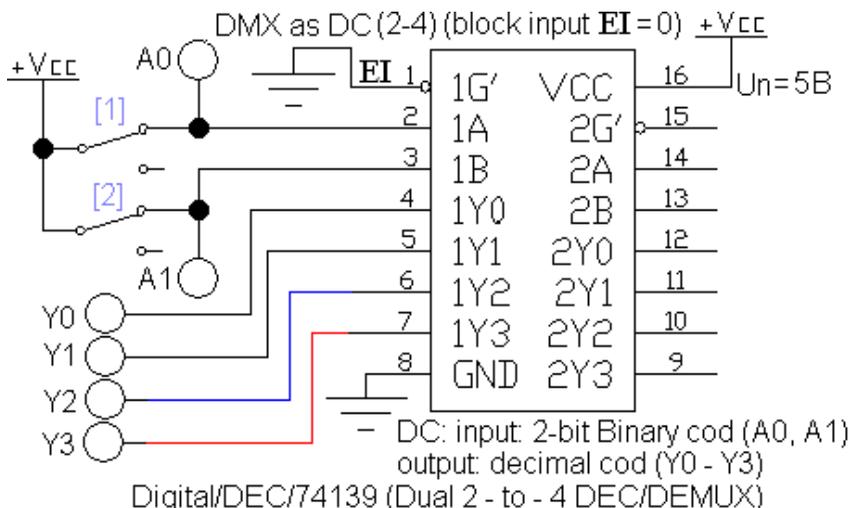


Рис. 6.9 Демультимплексор  $DMX(1-4)$  в качестве дешифратора  $DC(2-4)$

Увеличение разрядности демультиплексоров осуществляется аналогично дешифраторам за счет переключения двух  $DMX$  старшим разрядом входного адресного двоичного кода (рис. 4). В данной схеме  $DMX(1-8)$  используется два  $DMX(1-4)$  одной микросхемы. В качестве информационного входа использованы служебные входы  $EI$ . Адресные входы:  $A_2$ ,  $A_1$ ,  $A_0$ , выходы:  $Y_0$ ,  $Y_1 \dots Y_7$ . Демультимплексор может использоваться как дешифратор  $DC(3-8)$ , если вход  $EI$  постоянно подключен к нулю.

Основы микропроцессорной техники

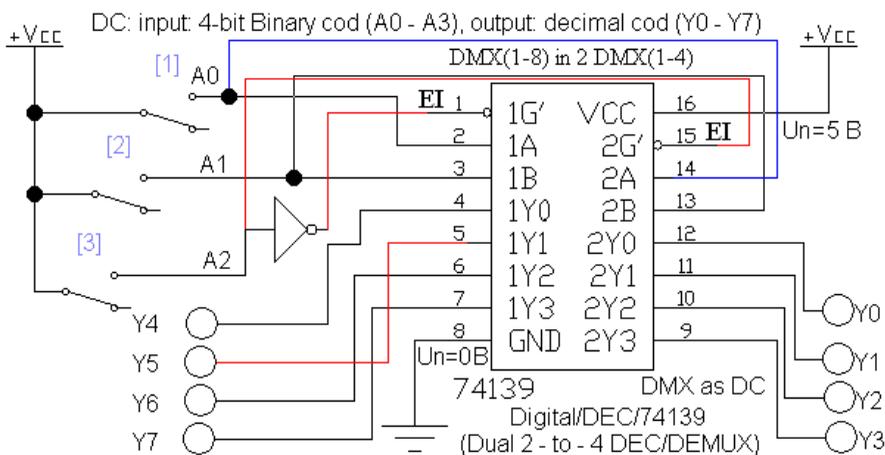


Рис. 6.10 Демультимплексор  $DMX(1-8)$  на двух  $DMX(1-4)$

Аналогичное техническое решение использовано для создания  $DMX(1-16)$  на двух  $DMX(1-8)$ , рис. 6.11.

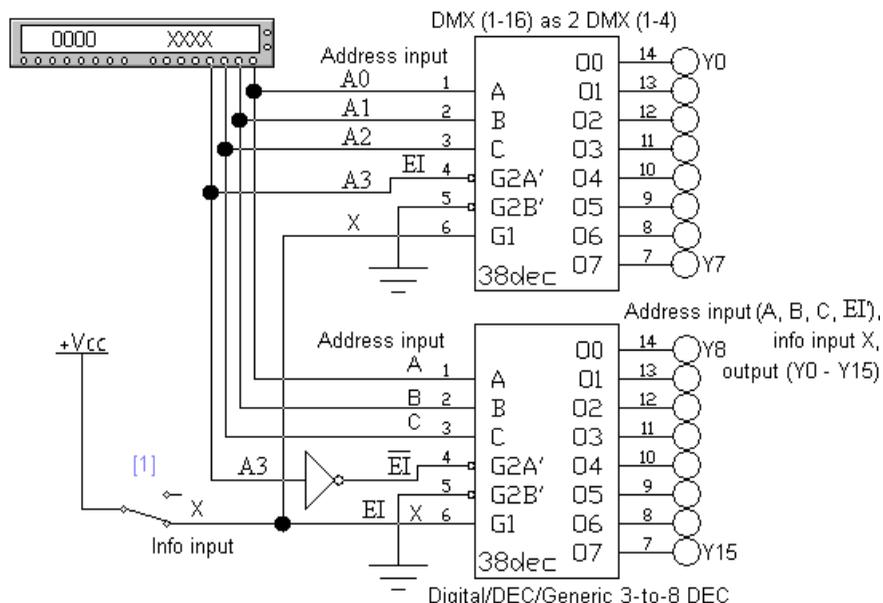


Рис. 6.11 Демультимплексор  $DMX(1-16)$  на двух  $DMX(1-8)$

Для передачи данных по одному общему каналу с разделением во времени используются демультиплексоры и мультиплек-

сору (рис. 6.12). Эти устройства выполняют по отношению друг к другу обратную функцию.

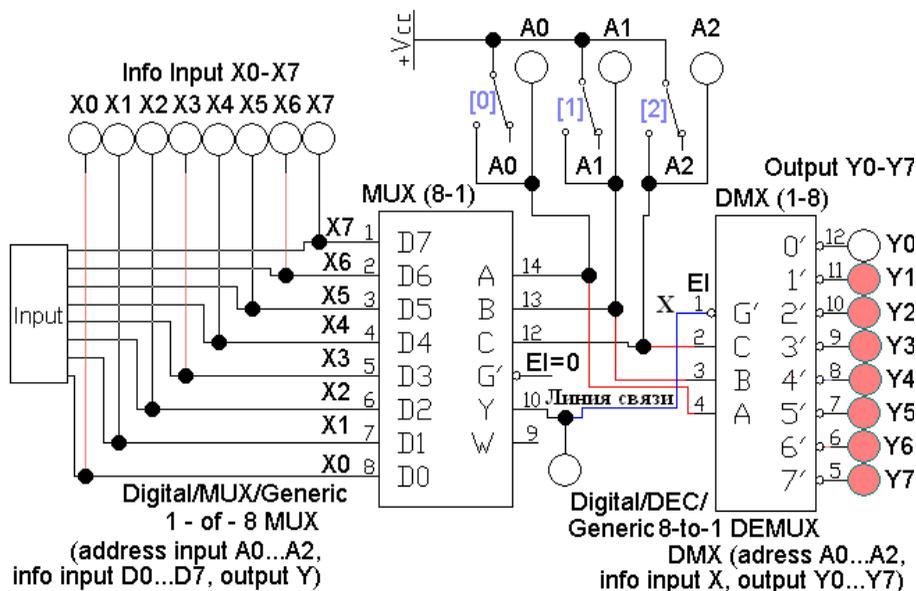


Рис. 6.12 Связь с временным разделением восьми каналов в последовательном коде на  $DMX(1-8)$  и мультиплексоре  $MUX(1-8)$

Система (рис. 6.12) содержит: набор источников информации в двоичном параллельном коде (*input*), передатчик, преобразующий параллельный код в последовательный – мультиплексор  $MUX(8-1)$ , на приемной части линии связи демультиплексор  $DMX(1-8)$  той же разрядности, преобразующий последовательный код в параллельный. Адресные коды устройств согласованы ( $A_0...A_2$ ) и, соответственно, приемники ( $Y_0...Y_7$ ) с передатчиками ( $X_0...X_7$ ) информации. Пример использования: Интернет, сотовая связь и т.д.

### Задание 1

Задавая комбинации входных сигналов, изучить работу демультиплексоров (рис. 6.7...6.12).

### Задание 2

Для дешифратора – демультиплексора (2-4), рис. 6.8 составить таблицу истинности (табл. 6.5) его работы как демультиплексора, а также в качестве дешифратора, задав соответствующую

щее значение переменной  $X$  (табл. 6.6).

Таблица 6.5 - Таблица истинности  $DMX$

Входы		Выходы			
$A_1$	$A_0$	$Y_3$	$Y_2$	$Y_1$	$Y_0$
0	0				
0	1				
1	0				
1	1				

Таблица 6.6 - Таблица истинности  $DC$

Входы			Выходы			
	$A_1$	$A_0$	$Y_3$	$Y_2$	$Y_1$	$Y_0$
	0	0				
	0	1				
	1	0				
	1	1				
	X	X				

Символ «X» обозначает любое значение переменной, которое не оказывает влияния на результат.

### Задание 3

Для модели системы связи (рис. 6.12), заполнить табл. 6.7. Требуется записать номера соединяемых линий  $X_0...X_7$  (параллельный код) с линиями  $Y_0...Y_7$  в соответствии с указанным адресным кодом  $A$  (одна линия  $X$  с одной линией  $Y$ ).

Таблица 6.7 - Таблица истинности системы связи

№	$A_2$	$A_1$	$A_0$	Номер $X$	Номер $Y$
1	0	0	0		
2	0	0	1		
3	0	1	0		
4	0	1	1		
5	1	0	0		
6	1	0	1		
7	1	1	0		
8	1	1	1		

**Задание 4.**

Найти в базе элементов мультиплексор в соответствии со своим вариантом. Получить таблицу истинности. Сделать выводы.

№ Варианта			
<b>Мультиплексоры</b>			
1	74LS150	КП1	16-входовый мультиплексор
2	74LS257	КП11	4 2-входовых мультиплексора
3	74LS253	КП12	двухканальный мультиплексор
4	74LS298	КП13	двухканальный мультиплексор и 4-разрядный регистр
5	74LS258	КП14	4 2-входовых мультиплексора
6	74LS251	КП15	мультиплексор
7	74LS153	КП2	2 4-входовых мультиплексора
8	74LS152	КП5	селектор-мультиплексор
9	74LS151	КП7	селектор-мультиплексор
<b>Демультимплексоры(дешифраторы)</b>			
10	74LS141	ИД1	дешифратор для управления газоразрядными индикаторами
11	74LS145	ИД10	двоично-десятичный дешифратор
12	74LS139	ИД14	двойной высокоскоростной дешифратор
13	74LS154	ИД3	дешифратор
14	74LS155	ИД4	2 дешифратора
15	74LS42	ИД6	двоично-десятичный дешифратор
16	74LS138	ИД7	высокоскоростной дешифратор-демультимплексор

*Вариант работы выбирается исходя из номера студента в списке группы*

*YZ минус 16 умножить на K,*

*где Y - предпоследняя цифра в номере списка; Z - последняя цифра в номере списка.*

*K - 0 или 1. (выбирается из условия, чтобы номер варианта не был отрицательным)*

## 7. ЛАБОРАТОРНАЯ РАБОТА №7: СОЗДАНИЕ ПРИЛОЖЕНИЯ ДЛЯ МОДЕЛИРОВАНИЯ РАБОТЫ МИКРОПРОЦЕССОРА С ЗАДАННЫМИ ПАРАМЕТРАМИ

### 7.1. Выполнение работы

Необходимо смоделировать работу микропроцессора с заданными параметрами. Для сдачи лабораторной работы необходимо реализовать:

1. возможность просмотра содержимого сегментов памяти;
2. просмотр содержимого регистров;
3. размещение в памяти набора команд;
4. выполнение набора команд с отслеживанием их выполнения;
5. все данные хранятся в двоичном виде.
6. 1 регистр данных и 1 регистр команд.

### 7.2. Задания для самостоятельного выполнения

#### *Вариант №1*

<b><i>Количество регистров</i></b>	8
<b><i>Размер слова в битах (один регистр может содержать не более 1 слова)</i></b>	16
<b><i>Тип регистра</i></b>	Общий
<b><i>Список базовых команд</i></b>	+, отрицание, перемещение в регистр, запись в память, обращение к биту регистра[.].
<b><i>Тип команд</i></b>	Префиксный (+ a b)
<b><i>Максимальный размер сегмента памяти</i></b>	32 слова
<b><i>Тип команд</i></b>	Адресный
<b><i>Реализовать на основе базовых команд операцию</i></b>	- a b (a и b вещественные числа)

**Вариант №2**

<b>Количество регистров</b>	8
<b>Размер слова в битах (один регистр может содержать не более 1 слова)</b>	16
<b>Тип регистра</b>	Общий
<b>Список базовых команд</b>	+, отрицание, перемещение в регистр, запись в память, обращение к биту регистра[.].
<b>Тип команд</b>	Инфиксный ( a + b)
<b>Максимальный размер сегмента памяти</b>	32 слова
<b>Тип команд</b>	Регистровый
<b>Реализовать на основе базовых команд операцию</b>	$A * B$

**Вариант №3**

<b>Количество регистров</b>	8
<b>Размер слова в битах (один регистр может содержать не более 1 слова)</b>	16
<b>Тип регистра</b>	Общий
<b>Список базовых команд</b>	+, отрицание, перемещение в регистр, запись в память, обращение к биту регистра[.].
<b>Тип команд</b>	Постпрефиксный ( a b +)
<b>Максимальный размер сегмента памяти</b>	32 слова
<b>Тип команд</b>	Регистровый
<b>Реализовать на основе базовых команд операцию</b>	$a \ b /$ (с остатком и без)

**Вариант №4**

<b>Количество регистров</b>	8
<b>Размер слова в битах (один регистр может содержать не более 1 слова)</b>	16
<b>Тип регистра</b>	Старшие и младшие
<b>Список базовых команд</b>	+ , -, отрицание, перемещение в регистр, запись в память, обращение к биту регистра[.].
<b>Тип команд</b>	Префиксный (+ a b)
<b>Максимальный размер сегмента памяти</b>	32 слова
<b>Тип команд</b>	Регистровый
<b>Реализовать на основе базовых команд операцию</b>	Размещение и суммирование элементов массива (размер элемента массива 1 слово)

**Вариант №5**

<b>Количество регистров</b>	8
<b>Размер слова в битах (один регистр может содержать не более 1 слова)</b>	16
<b>Тип регистра</b>	Общий
<b>Список базовых команд</b>	+ , отрицание, перемещение в регистр, запись в память, обращение к биту регистра[.].
<b>Тип команд</b>	Инфиксный ( a + b)
<b>Максимальный размер сегмента памяти</b>	32 слова
<b>Тип команд</b>	Адресный
<b>Реализовать на основе базовых команд операцию</b>	$A * B$

**Вариант №6**

<b>Количество регистров</b>	8
<b>Размер слова в битах (один регистр может содержать не более 1 слова)</b>	16
<b>Тип регистра</b>	Общий
<b>Список базовых команд</b>	+, -, отрицание, перемещение в регистр, запись в память, обращение к биту регистра[.].
<b>Тип команд</b>	префиксный (+ a b)
<b>Максимальный размер сегмента памяти</b>	32 слова
<b>Тип команд</b>	Регистровый
<b>Реализовать на основе базовых команд операцию</b>	Цикл for

**Вариант №7**

<b>Количество регистров</b>	8
<b>Размер слова в битах (один регистр может содержать не более 1 слова)</b>	16
<b>Тип регистра</b>	Общий
<b>Список базовых команд</b>	+, -, отрицание, перемещение в регистр, запись в память, обращение к биту регистра[.].
<b>Тип команд</b>	префиксный (+ a b)
<b>Максимальный размер сегмента памяти</b>	32 слова
<b>Тип команд</b>	Регистровый
<b>Реализовать на основе базовых команд операцию</b>	Цикл while

**Вариант №8**

<b>Количество регистров</b>	8
<b>Размер слова в битах (один регистр может содержать не более 1 слова)</b>	16
<b>Тип регистра</b>	Общий
<b>Список базовых команд</b>	+, -, отрицание, перемещение в регистр, запись в память, обращение к биту регистра[.].
<b>Тип команд</b>	префиксный (+ a b)
<b>Максимальный размер сегмента памяти</b>	32 слова
<b>Тип команд</b>	адресный
<b>Реализовать на основе базовых команд операцию</b>	Цикл repeat

**Вариант №9**

<b>Количество регистров</b>	8
<b>Размер слова в битах (один регистр может содержать не более 1 слова)</b>	16
<b>Тип регистра</b>	Общий
<b>Список базовых команд</b>	+, отрицание, перемещение в регистр, запись в память, обращение к биту регистра[.].
<b>Тип команд</b>	префиксный (+ a b)
<b>Максимальный размер сегмента памяти</b>	32 слова
<b>Тип команд</b>	адресный
<b>Реализовать на основе базовых команд операцию</b>	Поиск в памяти по значению

**Вариант №10**

<b>Количество регистров</b>	8
<b>Размер слова в битах (один регистр может содержать не более 1 слова)</b>	16
<b>Тип регистра</b>	Общий
<b>Список базовых команд</b>	+, отрицание, перемещение в регистр, запись в память, обращение к биту регистра[.].
<b>Тип команд</b>	префиксный (+ a b)
<b>Максимальный размер сегмента памяти</b>	32 слова
<b>Тип команд</b>	регистровый
<b>Реализовать на основе базовых команд операцию</b>	Очищение блока памяти

**Вариант №11**

<b>Количество регистров</b>	8
<b>Размер слова в битах (один регистр может содержать не более 1 слова)</b>	16
<b>Тип регистра</b>	Общий
<b>Список базовых команд</b>	+, отрицание, перемещение в регистр, запись в память, обращение к биту регистра[.].
<b>Тип команд</b>	префиксный (+ a b)
<b>Максимальный размер сегмента памяти</b>	32 слова
<b>Тип команд</b>	адресный
<b>Реализовать на основе базовых команд операцию</b>	Вывод списка свободных блоков памяти

**Вариант №12**

<b>Количество регистров</b>	8
<b>Размер слова в битах (один регистр может содержать не более 1 слова)</b>	16
<b>Тип регистра</b>	Общий
<b>Список базовых команд</b>	+, отрицание, перемещение в регистр, запись в память, обращение к биту регистра[.].
<b>Тип команд</b>	префиксный (+ a b)
<b>Максимальный размер сегмента памяти</b>	32 слова
<b>Тип команд</b>	регистровый
<b>Реализовать на основе базовых команд операцию</b>	Самый большой свободный блок памяти

**Вариант №13**

<b>Количество регистров</b>	8
<b>Размер слова в битах (один регистр может содержать не более 1 слова)</b>	16
<b>Тип регистра</b>	Общий
<b>Список базовых команд</b>	+, -, отрицание, перемещение в регистр, запись в память, обращение к биту регистра[.].
<b>Тип команд</b>	префиксный (+ a b)
<b>Максимальный размер сегмента памяти</b>	32 слова
<b>Тип команд</b>	регистровый
<b>Реализовать на основе базовых команд операцию</b>	If (переход к новой команде)

## СПИСОК ЛИТЕРАТУРЫ

1. Хернитер Марк Е., "Электронное моделирование в Multisim", М.: ДМК Пресс, 2010.-501с.
2. Инженерная графика «Схемы электрические»: Учебное пособие / СибГУТИ. Новосибирск, 2013 74 с.
3. Р.Токхейм Основы цифровой электроники — М.: Мир, 1988. — 392 с. (Глава 4, страницы 88—95)
4. Тюрин С.Ф., Аляев Ю.А. Дискретная математика:
5. Практическая дискретная математика и математическая логика: учеб. пособие.- Издательство: Финансы и статистика, 2010.