

Оглавление

Введение	4
Основы работы в программе Electronics Workbench и Multisim	6
Тема 1. Логические элементы	17
1.1. Основные соотношения алгебры логики (Булева алгебра)	18
1.1.1. Основные тождества алгебры логики	19
1.1.2. Основные законы алгебры логики	19
Тема 2. Арифметические устройства. Сумматоры и субтракторы	23
2.1. Арифметические сумматоры	23
2.2. Субтракторы	26
Тема 3. Цифровой компаратор	33
Тема 4. Устройство контроля четности	37
Тема 5. Шифраторы и дешифраторы	41
5.1. Шифраторы	41
5.2. Дешифраторы	42
Тема 6. Мультиплексоры и демультиплексоры	48
6.1. Мультиплексоры	48
6.2. Демультиплексоры	49
Тема 7. Арифметико-логическое устройство	51
Требования к содержанию и оформлению отчетов по лабораторным работам	59
Приложение 1	60
Приложение 2	62
Приложение 3	69
Приложение 4	70
Литература	71

ВВЕДЕНИЕ

Дисциплина «Вычислительные машины, системы и сети» позволяет усвоить принципы работы вычислительной техники, а также представлять себе тенденции ее развития. Базовыми для изучения этой дисциплины являются «Информатика» и «Программирование и основы алгоритмизации».

Сегодня вычислительная техника охватывает многие стороны жизни, причем с каждым днем она входит во все новые и новые области. Еще 30 лет назад вычислительные машины использовали только для математических операций, а сейчас уже сложно назвать ту сферу деятельности человечества, где их нет.

Развитие программного обеспечения и вычислительной техники коснулось и образовательной сферы. Если раньше лабораторный практикум по данной дисциплине проводился на реальном оборудовании, и проводилось физическое моделирование процессов в электрических цепях и устройствах, то теперь стало возможным математическое моделирование процессов с использованием виртуальных схем в программах Electronics Workbench и Multisim. Эти программы имеют достаточный набор инструментов и средств для использования в лабораторном практикуме дисциплины «Вычислительные машины, системы и сети».

Учебно-методическое пособие для лабораторного практикума включает в себя семь тем: «Логические элементы», «Арифметические устройства. Сумматоры и субтракторы», «Цифровой компаратор», «Устройство контроля четности», «Шифраторы и дешифраторы», «Мультиплексоры и демультиплексоры», «Арифметико-логическое устройство». Каждая тема содержит краткие теоретические сведения, задания и контрольные вопросы. Темы «Логические элементы» и «Арифметические устройства. Сумматоры и субтракторы» входят в состав первой лабораторной работы, темы «Цифровой компаратор» и «Устройство контроля четности» - второй, темы «Шифраторы и дешифраторы» и «Мультиплексоры и демультиплексоры» - третьей, а тема «Арифметико-логическое устройство» входит в состав четвертой лабораторной работы. Перечисленные темы лабораторного практикума содержат информацию о комбинационных устройствах (узлах).

Под комбинационными понимают узлы, не содержащие элементов памяти. В таких узлах всякое изменение состояния входных сигналов вызывает соответствующее изменение выходных сигналов с минимальной задержкой, зависящей от быстродействия данного узла. В общем случае комбинационный узел содержит несколько входов и несколько выходов. Поведение комбинационного узла описывается функциями алгебры логики (булевыми или логическими функциями). При этом для каждого выхода можно определить соответствующую булеву функцию, которая полностью определяет поведение комбинационного узла по данному выходу без учета задержки распространения сигнала, то есть для идеального узла. Время задержки сигнала по каждому выходу может быть разным. Время задержки зависит и от того, какой входной сигнал изменением своего состояния привел к соответствующему изменению данного выходного сигнала.

Поведение комбинационного узла может быть задано таблицей истинности, в которой каждой комбинации значений входных сигналов ставится в соответствие значение каждой выходной переменной. Количество комбинационных узлов, которые можно синтезировать, огромно даже при достаточно небольшом числе входных переменных. Но практика проектирования средств вычислительной техники показывает, что в большинстве случаев не требуется собственно разработки нового комбинационного узла, а достаточно ограничиться применением известных схемных решений, реализованных в виде микросхем. Достаточно лишь разумно использовать стандартные микросхемы.

Искренне надеемся, что изученный в рамках этой дисциплины материал послужит еще одним кирпичиком в Вашем жизненном фундаменте.

Основы работы в программе Electronics Workbench и Multisim

Electronics Workbench (EWB) канадской фирмы Interactive Image Technologies разработана в 1989 г. и в России известны версии 3.0, 4.0, 4.1, 5.0, 5.12 Professional Edition и Multisim. Программа непрерывно развивается, совершенствуется. Растет библиотека компонент, измерительных приборов, моделирующих функций. Во всех версиях остается неизменным (почти) дружественный интуитивный интерфейс, простой мощный графический редактор электрических схем, прекрасная интеграция с Windows системой.

Характерные особенности программы Electronics Workbench.

1) Схема изображается в графическом виде привычным образом. Из горизонтально расположенного меню выбирают библиотеку компонент, состав которой изображается слева от рабочего экрана. Движением мыши символы компонент переносят на схемы и выполняют электрические соединения. Достаточно указать начальный и конечный вывод цепи, как цепь будет проложена автоматически (правда, не всегда удачно, так что ее приходится немножко корректировать).

2) Полностью поддерживается текстовый формат программы моделирования SPICE, причем при загрузке текстового файла в формате SPICE на экране будет нарисована принципиальная схема с подключенными измерительными приборами (топология сложных схем синтезируется не вполне удачно, но моделируются такие схемы без ошибок).

3) Предусмотрен вывод списка соединений в формате программы OrCAD PCB (в файлах с расширением имени .NET) для разработки печатных плат.

4) Поддерживается стандартный набор компонент: резисторы, конденсаторы, индуктивности, управляемые линейные и нелинейные источники, линии задержки без потерь и с потерями, диоды, тиристоры, различные транзисторы, операционные усилители, цифровые интегральные схемы и др., а также светодиоды, цифровые индикаторы, резистивные матрицы, плавкие предохранители, лампочки накаливания и ключи. Имеется механизм создания макромоделей.

5) Предусмотрена возможность изменения параметров компонентов нажатием клавиш. Есть кнопочные переключатели, управляемые с клавиатуры. При этом параметры можно изменять, не прерывая моделирования. Как в реальном эксперименте.

6) Имеются следующие измерительные приборы: мультиметры (измерения постоянного и переменного напряжения и тока, сопротивления, результаты выводятся в относительных единицах и децибелах); двухлучевые осциллографы (регулируются усиления каналов, частота развертки, смещение лучей по координатам X, Y, имеются открытый и закрытый входы, предусмотрен ввод сигналов синхронизации); измерители частотных характеристик (Bode Plotter); генератор цифровых сигналов (Word Generator); цифровой логический анализатор и логический преобразователь. На схеме можно разместить только по одному из приборов каждого типа. При развертывании изображения лицевой панели прибора на весь экран с помощью двух электронных курсоров проводят точные измерения характеристик.

7) Различные цепи можно окрашивать в разные цвета для улучшения восприятия схемы. При этом временные диаграммы на экране двухлучевого осциллографа и многоканального логического анализатора окрашиваются в те же цвета.

8) Возможен ввод дискретных отсчетов сигналов из файлов.

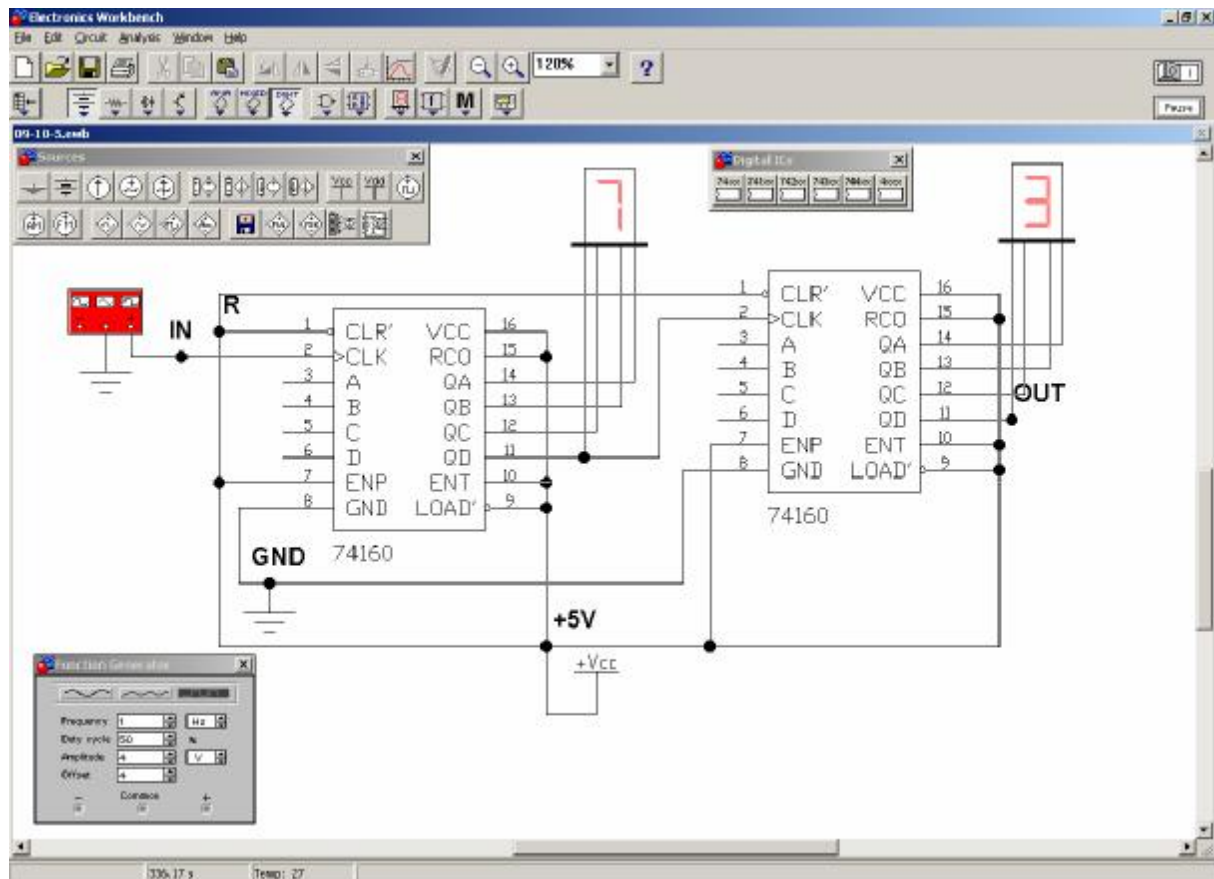
9) На периферийные устройства можно вывести принципиальную схему, ее текстовое описание, перечень компонентов.

Для измерительных инструментов рисуется лицевая панель с изображением характеристик и положением органов управления, а для осциллографа изображаются также эпюры напряжений неограниченной длины.

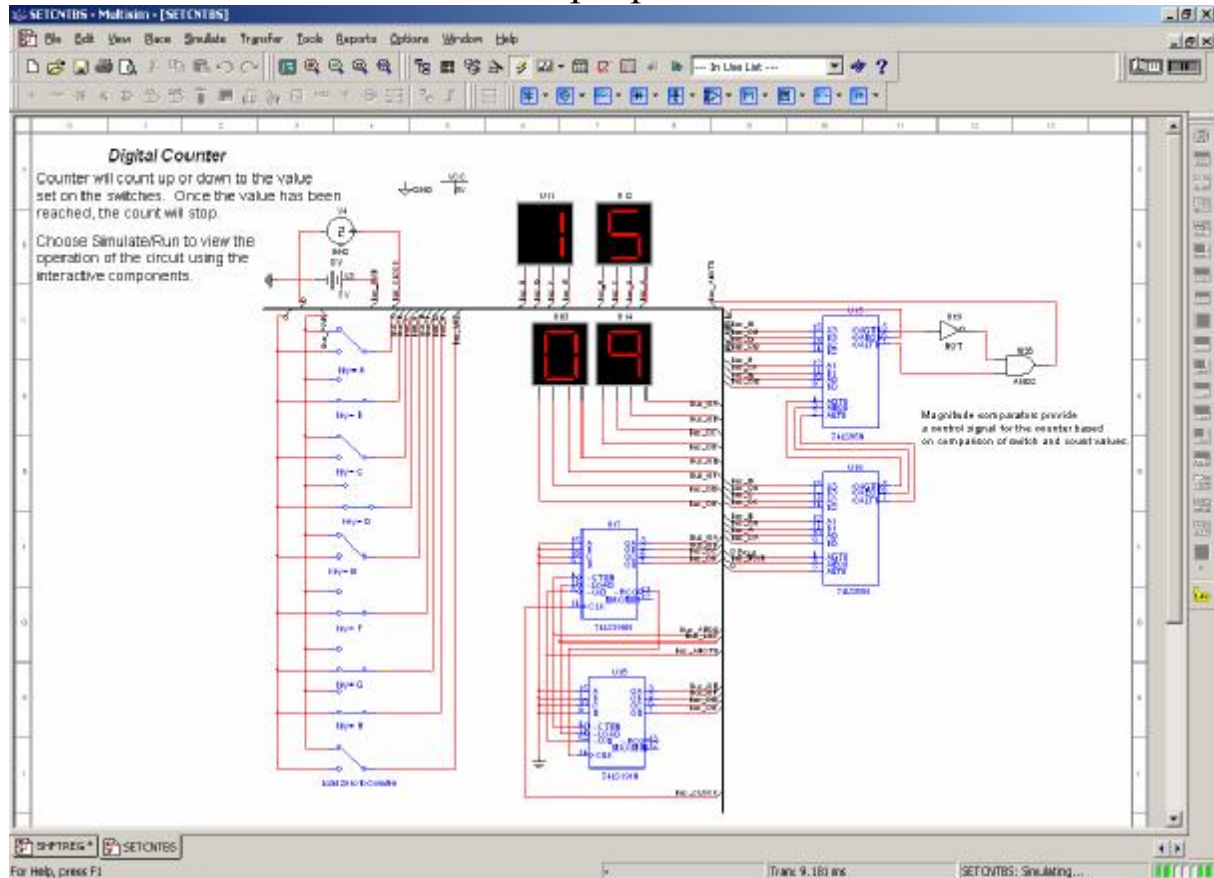
Интерфейс программы Electronics Workbench и Multisim

Главное окно программ Electronics Workbench и Multisim имеет стандартный оконный интерфейс пользователя.

Главное окно программы Electronics Workbench:



Главное окно программы Multisim:

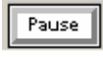


Меню Circuit программы Electronics Workbench

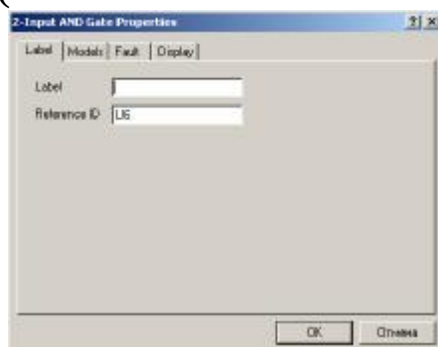
Меню Circuit используется при подготовке схем.

1. **Activate** (Ctrl + G) — запуск моделирования.

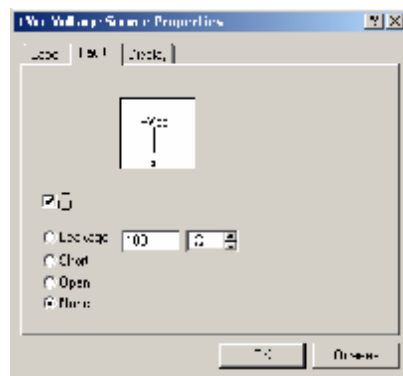
2. **Stop** (Ctrl + T) — остановка моделирования. Эта и предыдущая команды могут быть выполнены также нажатием кнопки , расположенной в правом верхнем углу экрана.

3. **Pause** (F9) — прерывание моделирования. Команда .

4. **Label** (Ctrl + L) — ввод позиционного обозначения выделенного компонента (например, R1 — для резистора, C5 — для конденсатора и т.д.) в диалоговом окне на рисунке ниже. При необходимости сдвига обозначения вправо можно слева ввести необходимое число пробелов (но не более 14 символов в строке).



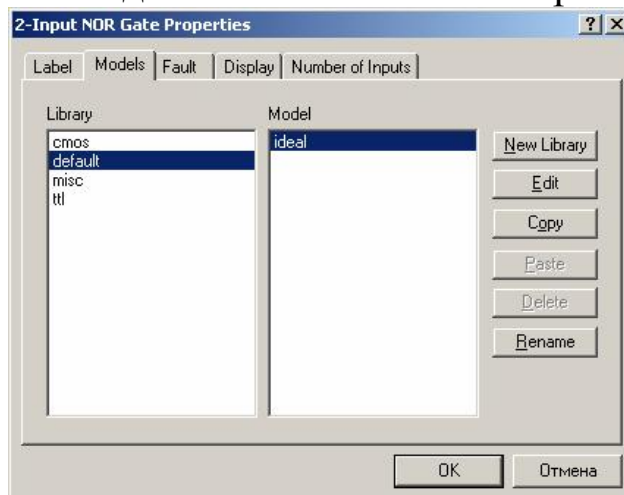
5. **Value** (Ctrl + U) — изменение номинального значения параметра компонента в диалоговом окне, например, для источника питания:



Команда выполняется также двойным щелчком по компоненту. Номинальное значение параметра вводится с клавиатуры, после чего нажатием курсором мыши на кнопки вверх-вниз выбирается множитель, кратный 1000. Например, для конденсатора задается его емкость в пикофарадах (пФ), нанофарадах (нФ), микрофарадах (мкФ) или миллифарадах (мФ).

6. **Model** (Ctrl + M) — выбор модели компонента (логического элемента, операционного усилителя, трансформатора и др.); ко-

манда выполняется также двойным щелчком по компоненту или при нажатии на пиктограмму . В меню команды выбирается: **Library** — перечень библиотек, в которых находятся компоненты выбранного типа; **Model** — перечень моделей компонентов выбранной библиотеки;



New Library — создание новой библиотеки; после внесения ее имени в вызываемом диалоговом окне и нажатия клавиши Ассерт (принять) это имя появится в колонке Library;

Edit — после нажатия этой кнопки вызывается диалоговое окно с параметрами выбранной модели. При необходимости редактирования параметров целесообразно по команде New Library создать отдельную библиотеку (чтобы не портить параметры библиотечного компонента куда переносится редактируемый компонент с помощью команд:

Copy (Ctrl + C) — копирование отмеченного в колонке Model компонента в буфер обмена;

Paste (Ctrl + V) — вставка скопированной в буфер обмена модели компонента в выбранную в колонке Library библиотеку (в том числе и вновь созданную) с последующим редактированием ее параметров без изменения характеристик компонента основной библиотеки;

Rename — переименование отмеченной модели компонента, работа с меню, как и во всех других подобных случаях, заканчивается нажатием кнопок Ассерт или Cancel — с сохранением или без сохранения введенных изменений.

При создании библиотеки моделей отечественных компонентов целесообразно действовать в следующей последовательности:

- создать библиотеку, например, под именем rus_lib;

- скопировать в эту библиотеку модель компонента, наиболее близкого по параметрам к отечественному компоненту;
- переименовать скопированную модель, присвоив ей, например, имя K140UD5 (латинская транскрипция К140УД5);
- при необходимости отредактировать значения параметров переименованной модели, используя данные каталогов отечественных микросхем или литературных источников.

7. Zoom (Ctrl + Z) — раскрытие (развертывание) выделенной подсхемы или контрольно-измерительного прибора, команда выполняется также двойным щелчком мыши по иконке компонента или прибора.

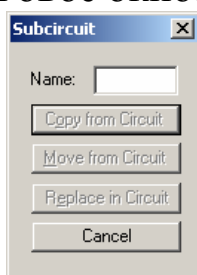
8. Rotate (Ctrl + R) — вращение выделенного компонента; большинство компонентов поворачиваются по часовой стрелке на 90° при каждом выполнении команды, для измерительных приборов (заземление, амперметр, вольтметр и др.) меняются местами клеммы подключения; команда используется при подготовке схем. В готовой схеме пользоваться командой нецелесообразно, поскольку это чаще всего приводит к путанице — в таких случаях компонент нужно сначала отключить, а затем вращать.

9. Fault (Ctrl + F) — имитация неисправности выделенного компонента путем введения: Leakage — сопротивления утечки; Short — короткого замыкания; Open — обрыва; None — неисправность отсутствует (включено по умолчанию).

10. Subcircuit (Ctrl + B) — преобразование предварительно выделенной части схемы в подсхему, можно использовать пиктограмму



. Выделяемая часть схемы должна быть расположена таким образом, чтобы в выделенную область не попали не относящиеся к ней проводники и компоненты. В результате выполнения команды вызывается диалоговое окно:



В строке Name вводится имя подсхемы, после чего возможны следующие варианты:

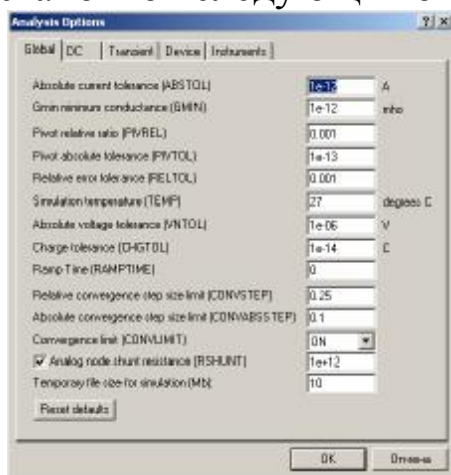
Copy from Circuit — подсхема копируется с указанным названием в библиотеку Custom без внесения изменений в исходную схему;

Move from Circuit — выделенная часть вырезается из общей схемы и в виде подсхемы с присвоенным ей именем копируется в библиотеку пользователя (Custom);

Replace in Circuit — выделенная часть заменяется в исходной схеме подсхемой присвоенным ей именем с одновременным копированием в библиотеку Custom.

Для просмотра или редактирования подсхемы нужно дважды щелкнуть мышью по ее значку. Редактирование подсхемы производится по общим правилам.

Analysis Options (Ctrl + Y) — выбор режимов моделирования в диалоговом окне с установкой следующих опций:



Меню Analysis Options (Ctrl + Y) — набор команд для установки параметров моделирования.

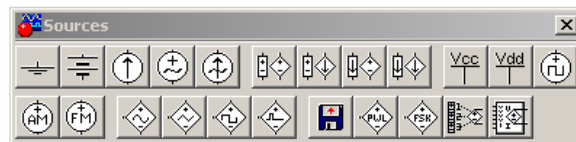
В наиболее общем случае процесс моделирования в программе EWB сводится следующему. После запуска моделирования данные моделируемой схемы считываются программой (с экрана монитора), затем компоненты заменяются их математическими моделями и составляется система линейных, нелинейных или дифференциальных уравнений по методу, аналогичному методу узловых потенциалов (отличие заключается в том, что потенциал рассчитывается для каждой точки схемы или так называемой **ноды** — точки соединения выводов двух и более компонентов, включая и **узел** — точку соединения трех и более компонентов). Далее система уравнений (матрица) преобразуется в две треугольные — нижнего (Low) и верхнего (Upper) уровней (LU-факторизация, аналогичная (напоминающая) двухходовому мето-

ду последовательного исключения переменных Гаусса), после чего для нахождения корней (потенциалов в каждой ноде) применяется метод Ньютона-Рафсона.

Библиотека компонентов EWB

Библиотека компонентов содержит следующие разделы:

1. **Favorites** – раздел, аналогичный группе Custom.
2. **Sources** – источники сигналов: источники питания и управляемые источники.



3. **Basic** – раздел, в котором собраны все пассивные компоненты и коммутационные устройства.



4. **Diodes** – диоды.



5. **Transistors** – транзисторы.



6. **Analog ICs** – аналоговые микросхемы.



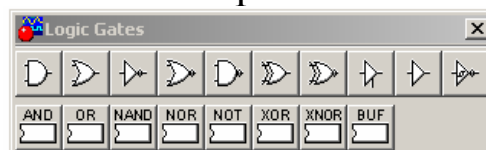
7. **Mixed ICs** – микросхемы смешанного типа.



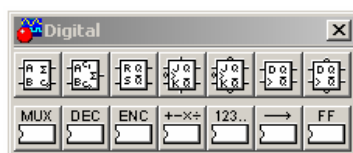
8. **Digital ICs** – цифровые микросхемы.



9. **Logic Gates** – логические микросхемы.



10. **Digital** – цифровые микросхемы.



11. Indicators – индикаторные устройства.



12. Controls – аналоговые вычислительные устройства.

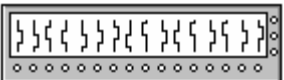


13. Miscellaneous – компоненты смешанного типа.

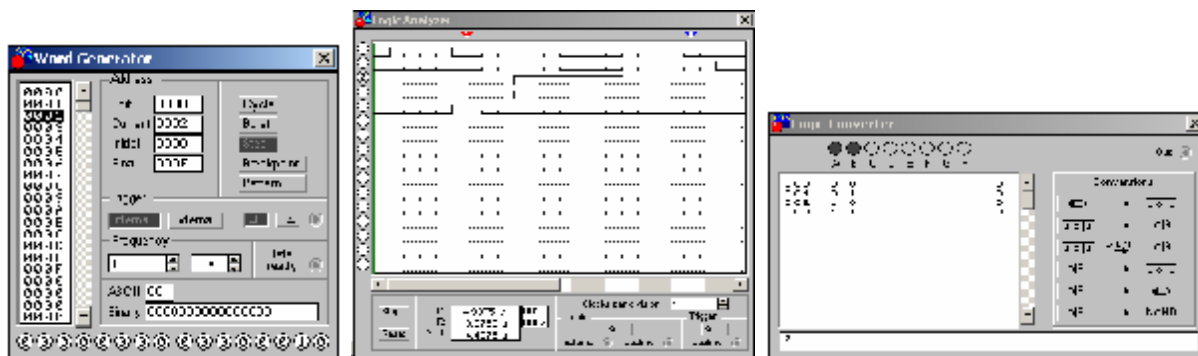


14. Instruments – контрольно-измерительные приборы.



Панель контрольно-измерительных приборов содержит цифровой мультиметр (Multimeter) , функциональный генератор (Function Generator) , двухканальный осциллограф (Oscilloscope) , измеритель амплитудно-частотных и фазо-частотных характеристик (Bode Plotter) , генератор слов (кодовый генератор – Word Generator) , 8-канальный анализатор логических сигналов (Logic Analyzer) , и логический преобразователь (Logic Converter) .

Общий порядок работы с приборами такой: иконка прибора курсором переносится на рабочее поле и подключается проводниками к исследуемой схеме. Для приведения прибора в рабочее (развернутое) состояние необходимо дважды щелкнуть курсором по его иконке. Далее представлены в развернутом виде основные контрольно-измерительные приборы, используемые в цифровой схемотехнике.



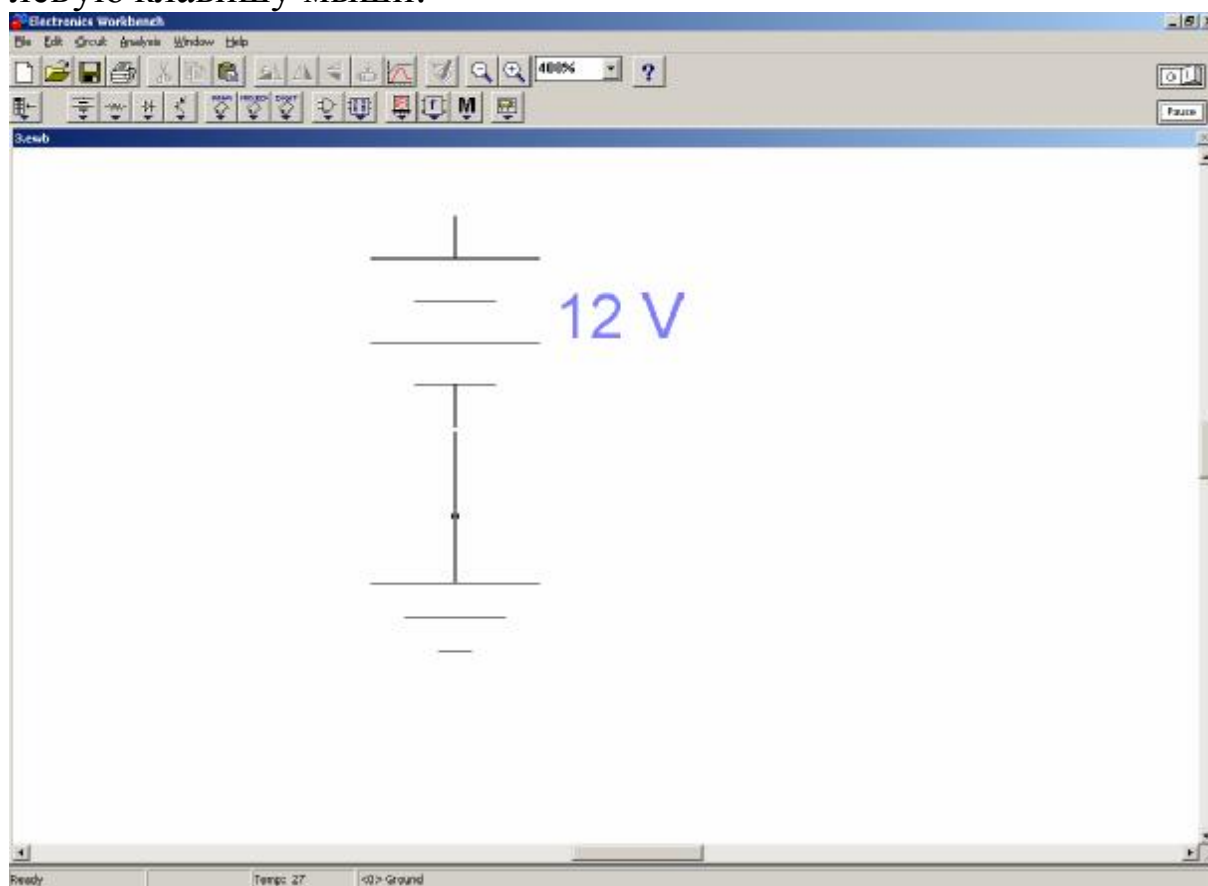
Генератор слов предназначен для создания на выходных зажимах последовательности 16-разрядных двоичных слов с заданной частотой следования (Frequency). Шестнадцатеричные значения слов устанавливаются с помощью клавиатуры в левом большом окне. В двух других, расположенных снизу меньших по размеру окнах можно установить значение слова в двоичном или ASCII-кодах. Для установки начальных (Initial) и конечных (Final) номеров слов генерируемой последовательности, а также для быстрого поиска нужного слова используется блок Address, где также можно установить адреса редактируемых (Edit) или установленных на выходных зажимах (Current) слов. Как правило, используется внутренняя (Internal) синхронизация (Trigger) генератора по переднему фронту и циклическая выдача логических сигналов (Cycle). Для синхронизации на анализируемую схему можно подать логический сигнал готовности данных (Data ready) с заданной частотой Frequency.

Анализатор логических сигналов предназначен для индикации двоичных кодов. Для правильного воспроизведения логических сигналов необходимо, нажимая кнопку Set, установить внутреннюю частоту прибора выше частоты генератора логических слов, а число импульсов на деление (Clock per division), равное для удобства наблюдения 1-3. В приборе имеются две визирные линии (красная и синяя), перемещаемые с помощью курсора.

Логический преобразователь предназначен для получения таблицы состояний комбинационной схемы, для преобразования таблицы истинности в логическую функцию и наоборот, а также для преобразования логической функции в схему устройства на логических элементах.

Для правильного соединения двух элементов между собой необходимо подвести курсор мыши так, чтобы появился черный маркер и, не отпуская левую клавишу мыши, подвести к другому

элементу и, как только там появится черный маркер, отпустить левую клавишу мыши:



Признаком, который определяет установившуюся связь между элементами, является исчезновение черных маркеров.

Принцип создания схем в программе Multisim незначительно отличается от программы Electronics Workbench. Основные отличия заключаются в том, что в Multisim добавлены компоненты в библиотеки элементов, отдельно выведены устройства TTL (ТТЛ) и CMOS (КМОП) на панели инструментов Components, а также можно одновременно открывать несколько схем в одной программе.

Тема 1 ЛОГИЧЕСКИЕ ЭЛЕМЕНТЫ

Цели работы:

1. Изучить принцип работы логических элементов.
2. Составление таблиц истинности элементов.

Необходимое оборудование в программе EWB:

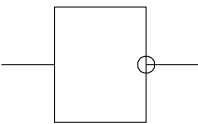
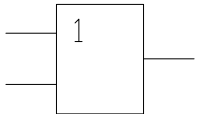
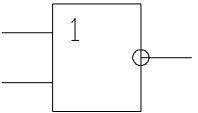
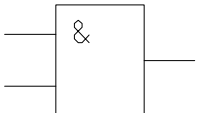
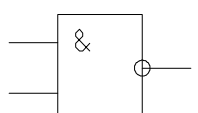
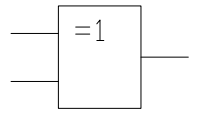
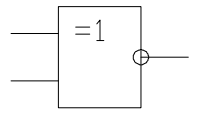
1. Модуль ключей.
2. Модуль светодиодных индикаторов.
3. Модули логических элементов.
4. Источник напряжения.

Краткие теоретические сведения.

Логические элементы на интегральных микросхемах (ИМС) – это микроэлектронные изделия, предназначенные для преобразования и обработки дискретных сигналов. К основным параметрам логических элементов относятся: число входов по И и по ИЛИ; коэффициент разветвления по выходу (характеризует нагрузочную способность логического элемента и определяется количеством входов однотипных элементов, которые можно подключить к выходу); потребляемая мощность; динамические параметры: задержка распространения сигнала и (или) максимальная частота входного сигнала. Число входов по И и по ИЛИ лежит в пределах от 2 до 8. Если имеющегося числа входов недостаточно, то для их увеличения используются интегральные схемы расширителей по ИЛИ. Из алгебры логики известно, что сложные логические функции можно выразить через совокупность конечного числа базисных логических функций. Такие совокупности образуют: базисные логические функции НЕ (инверсия), И (конъюнкция), ИЛИ (дизъюнкция). В таб. 1 приведены основные логические элементы и их функции.

В зависимости от технологии изготовления логические ИМС делятся на серии, отличающиеся набором элементов, напряжением питания, потребляемой мощностью, динамическими параметрами и др. Наибольшее применение получили серии логических ИМС, выполненные по ТТЛ (транзисторно-транзисторная логика), ЭСЛ (эмиттерно-связанная логика) и КМОП (комплементарная МОП логика) технологиям.

Таб. 1

Название	Условное графическое обозначение		Функция	
	Иностранное	Российское		
НЕ (NOT)			логическое отрицание (инверсия)	$y = \bar{x}$
ИЛИ (OR)			логическое сложение	$y = x_1 + x_2$
ИЛИ-НЕ (NOR)			логическое сложение и инвертирование результата	$y = \overline{(x_1 + x_2)}$
И (AND)			логическое умножение	$y = x_1 \cdot x_2$
И-НЕ (NAND)			логическое умножение и инвертирование результата	$y = \overline{(x_1 \cdot x_2)}$
Исключающее ИЛИ (XOR)			арифметическое сложение	$y = x_1 \oplus x_2$
Исключающее ИЛИ-НЕ (XNOR)			арифметическое сложение и инвертирование результата	$y = \overline{(x_1 \oplus x_2)}$

1.1. Основные соотношения алгебры логики (Булева алгебра)

Приводимые далее соотношения записаны в основном для частного случая двух переменных.

1.1.1. Тожества алгебры логики:

$$\begin{aligned}
 x + 0 &= x; & x + 1 &= 1; & x + x &= x; & x + \bar{x} &= 1; \\
 x \cdot 0 &= 0; & x \cdot 1 &= x; & x \cdot x &= x; & \bar{x} \cdot x &= 0; & x &= \bar{\bar{x}}.
 \end{aligned}$$

Справедливость первых двух тождеств каждой строки легко установить, имея в виду, что вторая переменная задана (0 — событие не наступило, 1 — наступление события). Поэтому правая часть, к примеру, первого тождества верхней строки, т.е. сложное событие целиком определяется значением первой переменной x . Третье выражение первой строки тождеств ($x + x = x$) говорит о том, что логическая сумма одинаковых событий равна одному из них. Последнее выражение первой строки тождеств ($x + \bar{x} = 1$) — закон исключения третьего: из двух противоположных событий (например, ключ замкнут или ключ разомкнут) одно обязательно должно быть истинным (третьего не дано), поэтому их логическая сумма истинна. Предпоследнее выражение второй строки тождеств ($\bar{x} \times x = 0$) — закон противоречия: два противоположных события (например, ключ замкнут и ключ разомкнут) не могут быть одновременно истинными.

1.1.2. Основные законы алгебры логики. При сложении и умножении переменных действуют следующие законы алгебры логики:

переместительный закон

$$\begin{aligned}x_1 + x_2 &= x_2 + x_1; \\x_1 x_2 &= x_2 x_1;\end{aligned}$$

сочетательный закон

$$\begin{aligned}(x_1 + x_2) + x_3 &= x_1 + (x_2 + x_3); \\(x_1 x_2) x_3 &= x_1 (x_2 x_3);\end{aligned}$$

распределительный закон

$$\begin{aligned}x_1 (x_2 + x_3) &= x_1 x_2 + x_1 x_3; \\x_1 + (x_2 x_3) &= (x_1 + x_2) (x_1 + x_3).\end{aligned}$$

Закон поглощения

$$\begin{aligned}x_1 + x_1 x_2 &= x_1; \\x_1 (x_1 + x_2) &= x_1.\end{aligned}$$

Закон склеивания

$$\begin{aligned}x_1 x_2 + \bar{x}_1 \bar{x}_2 &= x_1; \\(x_1 + x_2)(\bar{x}_1 + \bar{x}_2) &= x_1.\end{aligned}$$

Закон отрицания (теорема де Моргана):

дизъюнкции

$$\overline{x_1 + x_2} = \bar{x}_1 \bar{x}_2; \quad (1)$$

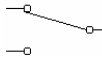
конъюнкции

$$\overline{x_1 x_2} = \overline{x_1} + \overline{x_2}. \quad (2)$$

Справедливость этих выражений следует из принципа двойственности алгебры Буля.

Приведем пример, иллюстрирующий теорему де Моргана. Пусть переменная x_1 представляет поездку на автобусе, а x_2 - поездку на трамвае. Правая часть равенства (1) выражает то, что мы не поедem на автобусе ($\overline{x_1}$) И не поедem на трамвае ($\overline{x_2}$). Поэтому высказывание «Мы поедem на автобусе ИЛИ на трамвае» - $(x_1 + x_2)$ не верно, и его надо заменить на противоположное - $(\overline{x_1} + \overline{x_2})$, чему и соответствует левая часть равенства.

Задание 1. Выполните проверку таблицы истинности элемента НЕ (NOT).

а) Соедините вход элемента НЕ  с ключом , а выход со светодиодным индикатором .

б) Задайте последовательно два возможных уровня на вход и сравните с таблицей истинности элемента НЕ.

Задание 2. Выполните проверку таблицы истинности для элемента ИЛИ (OR).

Для двухвходовых элементов (см. таб. 1) можно увеличить количество входов до восьми, открывая двойным щелчком по значку компонента диалоговое окно. По умолчанию в этом окне указано минимально возможное число входов, равное двум. При оптимизации логических устройств с целью получения наиболее простой схемы используются различные методы минимизации булевых выражений (карты Карно, диаграммы Вейча и др.), однако при числе входов более пяти этот процесс становится крайне трудоемким.

а) Соедините каждый из двух входов элемента ИЛИ со своим ключом, а выход соедините со светодиодным индикатором.

б) Задайте последовательно четыре возможных комбинаций логических сигналов на входы и запишите соответствующие значения логического уровня на выходе.

в) По полученным результатам составьте таблицу истинности.

Задание 3. Выполните проверку таблицы истинности элемента И (AND).

а) Соедините каждый из трех входов элемента И со своим ключом, а выход соедините со светодиодным индикатором.

б) Задайте последовательно восемь возможных комбинаций логических сигналов на входы и запишите соответствующие значения выхода.

в) По полученным результатам составьте таблицу истинности.

Задание 4. Составьте таблицу истинности элемента ИСКЛЮЧАЮЩЕЕ ИЛИ (XOR).

а) Соедините каждый из двух входов элемента ИСКЛЮЧАЮЩЕЕ ИЛИ со своим ключом, а выход соедините со светодиодным индикатором.

б) Задайте последовательно четыре возможных комбинаций логических сигналов на входы и запишите соответствующие значения логического уровня на выходе.

в) По результатам эксперимента составьте таблицу истинности и сравните с таблицей истинности в задании 2 (в).

Задание 5. Составьте таблицы истинности элементов ИЛИ-НЕ, И-НЕ, ИСКЛЮЧАЮЩЕЕ ИЛИ-НЕ.

а) Соедините каждый из двух входов элемента ИЛИ-НЕ, каждый из пяти входов элемента И-НЕ, каждый из двух входов элемента ИСКЛЮЧАЮЩЕЕ ИЛИ-НЕ со своим ключом, а выход соедините со светодиодным индикатором.

б) Задайте последовательно все возможные состояния на входы элементов: ИЛИ-НЕ, И-НЕ, ИСКЛЮЧАЮЩЕЕ ИЛИ-НЕ и запишите соответствующие значения выходов для каждого.

в) По полученным результатам составьте таблицы истинности.

г) Сравните полученные таблицы истинности ИЛИ-НЕ, И-НЕ, ИСКЛЮЧАЮЩЕЕ ИЛИ-НЕ с таблицами истинности элементов ИЛИ, И и ИСКЛЮЧАЮЩЕЕ ИЛИ соответственно.

Задание 6. Проведите моделирование двухвходовых логических элементов ИЛИ (ИЛИ-НЕ), И (И-НЕ), ИСКЛЮЧАЮЩЕЕ ИЛИ (ИСКЛЮЧАЮЩЕЕ ИЛИ-НЕ) с использованием логического преобразователя и установите для каждого из них соответствие таблицы истинности и булева выражения.

Задание 7. Проведите испытания ИМС логического элемента в соответствии с заданным вариантом, представленном в Приложении 1 (таб.4 или таб.5). В программе EWB5.12 используйте

библиотеку *Digital ICs*, а в программе Multisim – библиотеки *TTL* и *CMOS*.

Сделайте выводы о получившихся таблицах истинности логических элементов.

Контрольные вопросы и задания

1. Что такое бит? Какие значения он может принимать?
2. На два из трех входов элемента ИЛИ-НЕ подан уровень логической 1. Уровень на оставшемся входе неизвестен. Каков будет на выходе уровень сигнала? Ответ аргументировать.
3. У вас есть модуль с двухвходовыми и трехвходовыми элементами ИЛИ. Придумайте и нарисуйте логическую цепь, чтобы получился один шестивходовой элемент. Какую функцию будет выполнять полученный элемент?
4. Составьте схему управления светофором на элементах ИЛИ, НЕ или И, НЕ, имея два входа и 4 выходных состояния: горит зеленый свет, горит желтый свет, горит красный свет, горит желтый и красный свет.
5. Укажите логический потенциал (1 или 0), который нужно подать на один из 2-х входов элемента «Исключающее ИЛИ», чтобы по второму входу он выполнял роль инвертора.
6. Укажите логический потенциал (1 или 0), который нужно подать на один из 2-х входов элемента «Исключающее ИЛИ», чтобы по второму входу он выполнял роль повторителя.
7. Укажите логический потенциал (1 или 0), который нужно подать на один из 2-х (или n) входов элемента И-НЕ, однозначно определяющий состояние его выхода, а также значение потенциала на выходе.
8. Укажите логический потенциал (1 или 0), который нужно подать на один из 2-х (или n) входов элемента ИЛИ-НЕ, однозначно определяющий состояние его выхода, а также значение потенциала на выходе.
9. Сформулируйте основные законы булевой алгебры.
10. Спроектируйте схему, реализующую функцию ИЛИ 3-х переменных на элементах 2И-НЕ.

Тема 2

АРИФМЕТИЧЕСКИЕ УСТРОЙСТВА. СУММАТОРЫ И СУБТРАКТОРЫ

Цели работы:

1. Изучить структуру и принцип работы четверть сумматора.
2. Изучить структуру и принцип работы полусумматора.
3. Изучить структуру и принцип работы одноразрядного сумматора.
4. Изучить структуру и принцип работы субтрактора (вычитателя).

Необходимое оборудование в программе EWB:

1. Модуль ключей или генератор слова (Word Generator).
2. Модуль полусумматора.
3. Модуль одноразрядного полного сумматора.
4. Модуль светодиодного индикатора.
5. Источники напряжения.

Краткие теоретические сведения.

Основной элементарной операцией, выполняемой над кодами чисел в цифровых устройствах, является арифметическое сложение.

2.1. Арифметические сумматоры

Арифметический сумматор - логический операционный узел, выполняющий арифметическое сложение кодов двух чисел. При арифметическом сложении выполняются и другие дополнительные операции: учет знаков чисел, выравнивание порядков слагаемых. Указанные операции выполняются в арифметико-логических устройствах (АЛУ) или процессорных элементах, ядром которых являются сумматоры.

Сумматоры классифицируют по различным признакам.

В зависимости от системы счисления различают:

1. Двоичные.
2. Двоично-десятичные (в общем случае двоично-кодированные).
3. Десятичные.
4. Прочие (например, амплитудные).

По количеству одновременно обрабатываемых разрядов складываемых чисел:

1. Одноразрядные.
2. Многоразрядные.

По числу входов и выходов одноразрядных двоичных сумматоров.

1. *Четвертьсумматоры* (элементы "сумма по модулю 2"; элементы "исключающее ИЛИ"), характеризующиеся наличием двух входов, на которые подаются два одноразрядных числа, и одним выходом, на котором реализуется их арифметическая сумма.

2. *Полусумматоры*, характеризующиеся наличием двух входов, на которые подаются одноименные разряды двух чисел, и двух выходов: на одном реализуется арифметическая сумма в данном разряде, а на другом перенос в следующий (более старший разряд).

Полусумматор (рис. 1) имеет два входа a и b для двух слагаемых и два выхода: S сумма, P перенос.

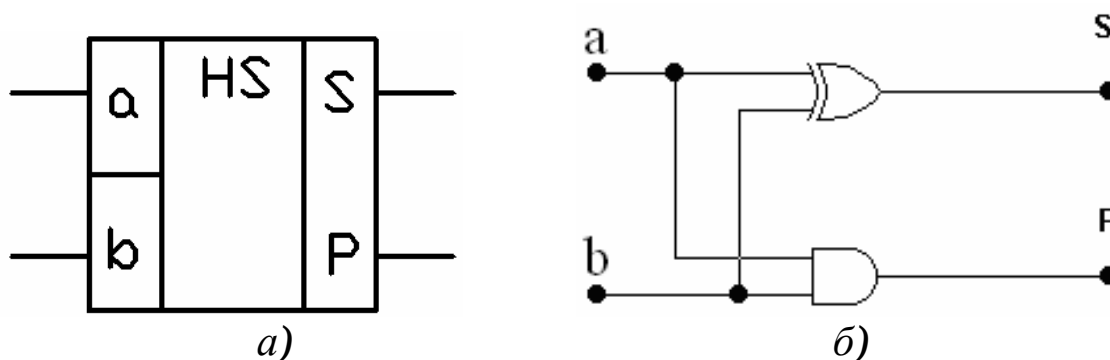


Рис. 1. Условное графическое изображение полусумматора а) и схема на логических элементах полусумматора б)

3. *Полные одноразрядные двоичные сумматоры*, характеризующиеся наличием трех входов, на которые подаются одноименные разряды двух складываемых чисел и перенос из предыдущего (более младшего) разряда, и двумя выходами: на одном реализуется арифметическая сумма в данном разряде, а на другом перенос в следующий (более старший разряд).

Полный одноразрядный двоичный сумматор (рис. 2) имеет три входа: a , b для двух слагаемых и P для переноса из предыдущего (более младшего) разряда и два выхода: S сумма, P перенос в следующий (более старший) разряд. Обозначением полного двоичного сумматора служат буквы SM. Полный одноразрядный двоичный сумматор может быть реализован на двух полусумматорах и одном двухвходовом элементе ИЛИ (рис. 2, б).

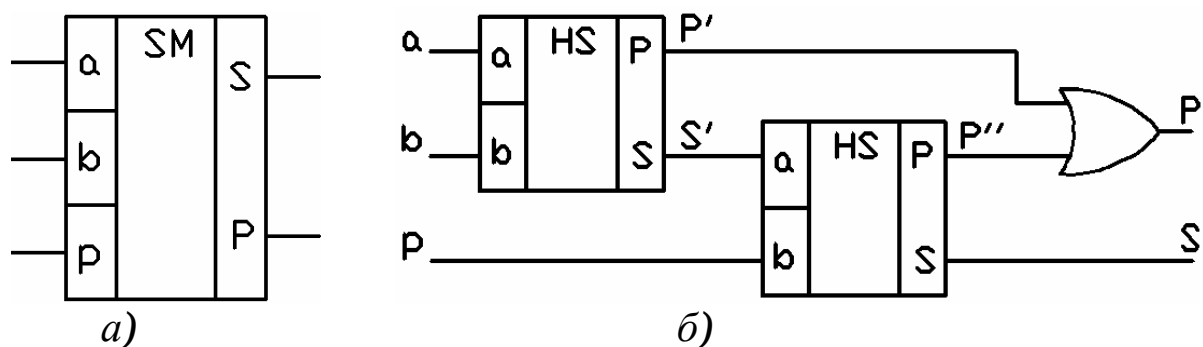


Рис. 2. Условное графическое изображение полного одноразрядного сумматора а) и его схема на двух полусумматорах и одном двухвходовом элементе ИЛИ б)

Многоразрядным сумматором называется устройство, предназначенное для сложения двух многоразрядных кодов, формирующее на выходе код суммы и сигнал переноса в случае, если результат сложения не может быть представлен кодом, разрядность которого совпадает с разрядностью кодов слагаемых.

В свою очередь, многоразрядные сумматоры подразделяются на *последовательные* и *параллельные*. В последовательных сумматорах операция сложения выполняется последовательно разряд за разрядом, начиная с младшего. В параллельных все разряды входных кодов суммируются одновременно.

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

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

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

Схемы и принцип действия многоразрядных сумматоров подробно представлены в учебном пособии «Цифровая схемотехника» Угрюмов Е.П.

2.2. Субтракторы

Субтракторы (вычитатели) выполняют операции противоположные сумматорам и их структура и условное графическое обозначение приведены на рис. 3. Условное графическое обозначение *полувычитателя* показано на рис. 3, а. Слева указаны входы a и b , справа - выходы D_i и B_0 . При вычитании многоразрядных двоичных чисел нужно принимать во внимание заем «единиц» в более старших разрядах.

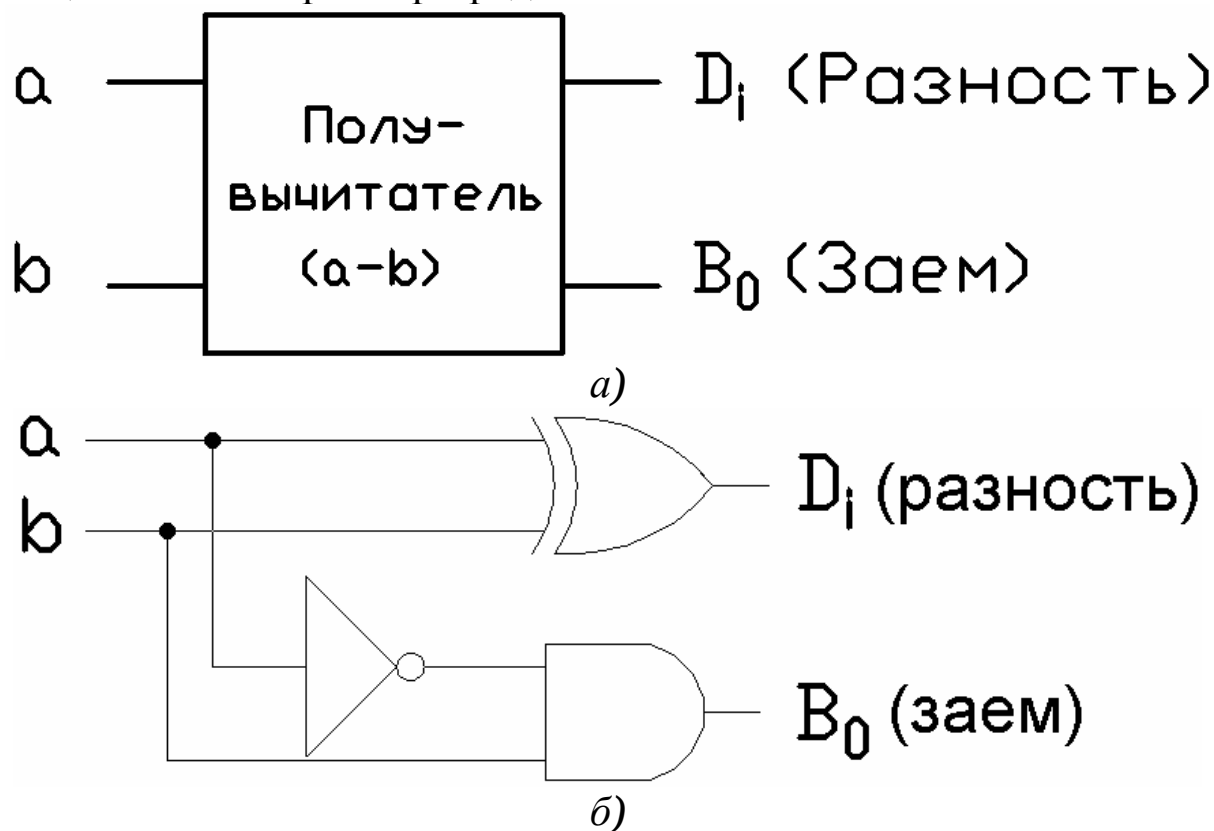
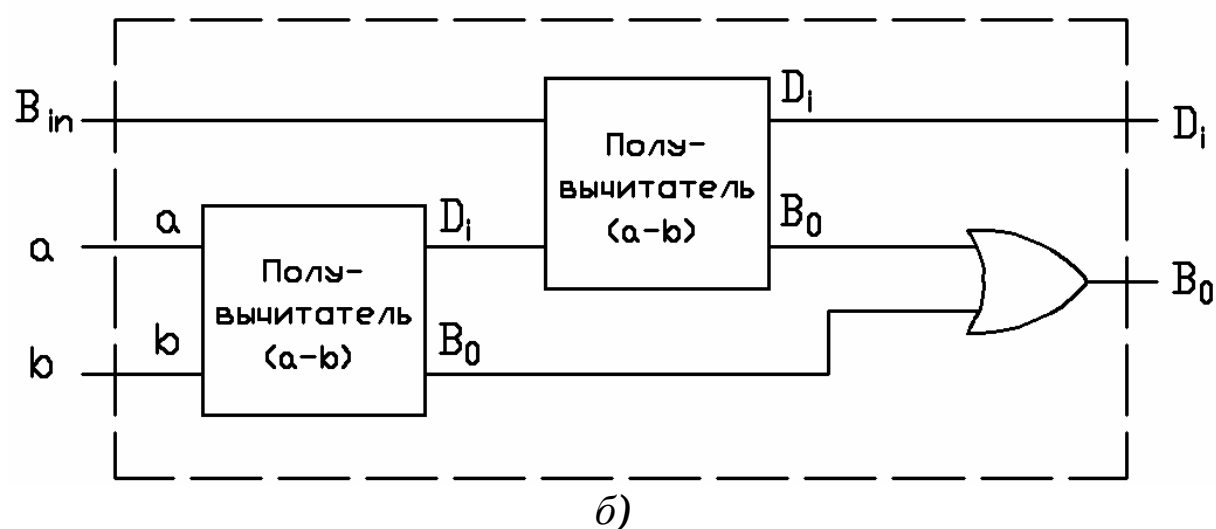


Рис. 3. Полувычитатель. а) – условное графическое обозначение; б) - принципиальная схема

Условное графическое обозначение *полного вычитателя* показано на рис. 4, а. Слева - входы a , b и B_{in} , справа - выходы D_i и B_0 . По аналогии с полным сумматором полный вычитатель можно собрать из двух полувычитателей и логического элемента ИЛИ (рис. 4, б). Принципиальная схема полного вычитателя, составленная на логических элементах приведена на рис. 4, в.



Полный вычитатель



Полувывчитатель

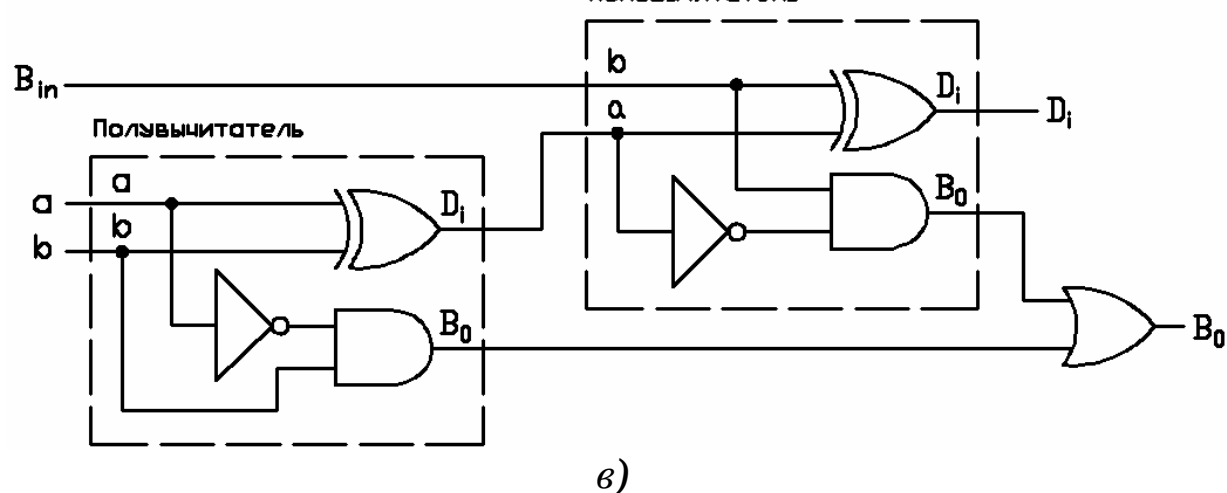


Рис. 4. Полный вычитатель.

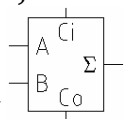
а) - Условное графическое обозначение; б) - структурная схема для случая использования двух полувывчитателей и логического элемента ИЛИ; в) - принципиальная схема.

Задание 1. Составьте булевы выражения для S и P по таблице истинности (см. таб. 2) для реализации схемы полусумматора, показанного на рис. 1, б. Соберите и проверьте работу одноразрядного полусумматора, где a и b – входы операндов, S – выход суммы, P – выход переноса.

Таб. 2

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

Задание 2. Составьте таблицу истинности сложения двух 2-х разрядных чисел, используя модули с полными одноразрядными сумматорами



(см. таб. 3).

Таб. 3

Входное число a		Входное число b		Перенос P		Сумма по разрядам S	
a_1	a_0	b_1	b_0	P_1	P_2	S_1	S_0
...	...	...	...	...	...	...	...

Задание 3. Соберите схему, реализующую вычитание двух чисел и составьте таблицу истинности полного одноразрядного вычитателя.

Задание 4. Соберите схему многоразрядного параллельного сумматора с последовательным переносом в соответствии с заданным вариантом по его разрядности, взяв за основу 3-х разрядный (рис. 5).

Варианты:

Номер варианта	1	2	3	4	5	6
разрядность	4	5	6	7	8	9

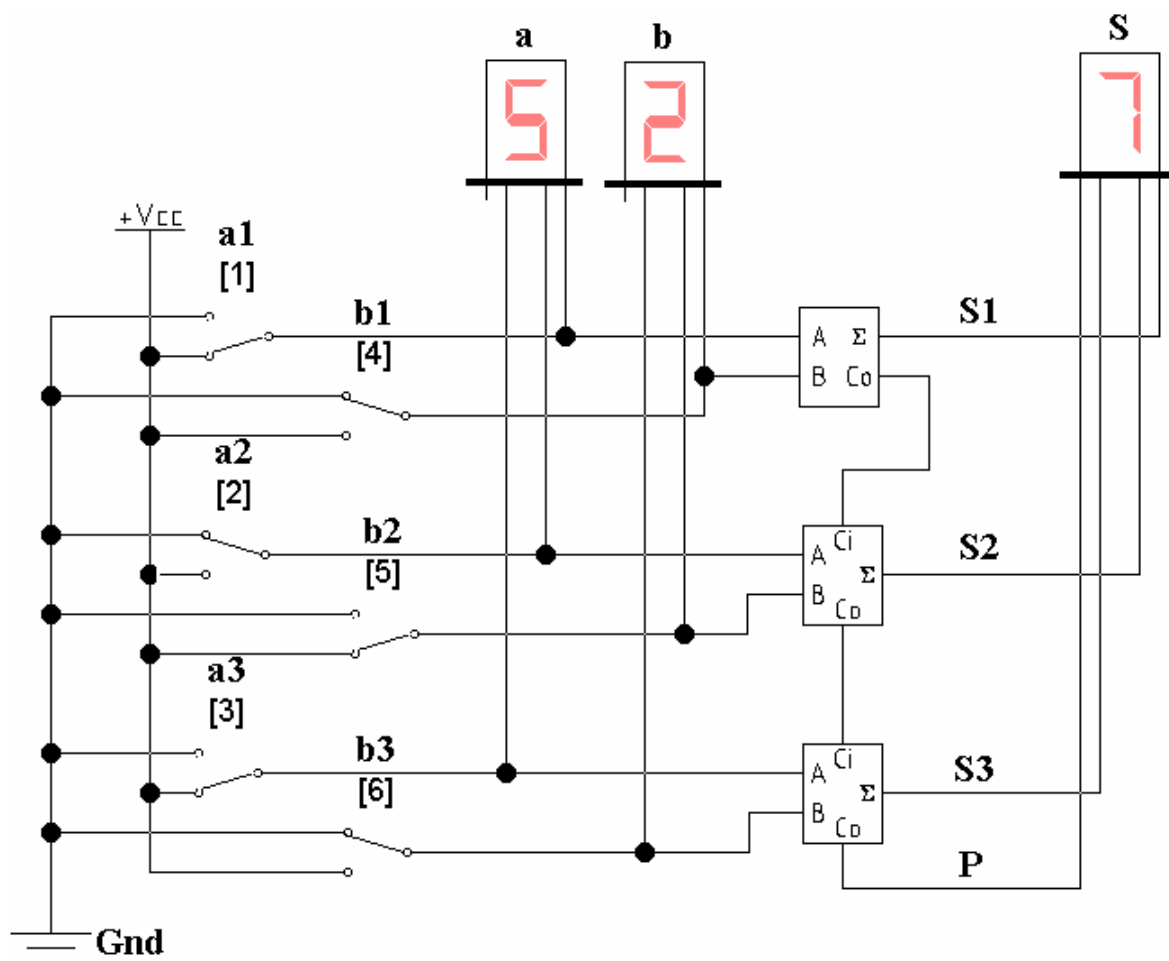


Рис. 5. Схема 3^x разрядного параллельного сумматора с последовательным переносом

Задание 5. На интегральной микросхеме (ИМС) арифметико-логического устройства (АЛУ) 74181 (Российский аналог К155ИП3) соберите многоразрядный сумматор (рис. 6) в соответствии со своим вариантом из задания 4.

Эта ИМС обеспечивает 32 режима работы АЛУ в зависимости от состояния управляющих сигналов на входах M , $SO \div S3$. На рис. 6 (б) представлена схема ИМС 74181 в программе EWB 5.12, и ее аналог в программе Multisim на рис. 6 (а) по выбору.

Возможные режимы задаются с помощью входов управления SO , $S1$, $S2$, $S3$. При подаче сигнала 0 на вход M , выполняются 16 арифметических операций (16 комбинаций сигналов $SO \div S3$) с учетом переноса по входу CN или без учета переноса (сигнал 0 на входе CN). Когда же на входе M появляется сигнал 1 выполняют-

ся 16 логических операций, задаваемых теми же входами управления $S0, S1, S2, S3$.

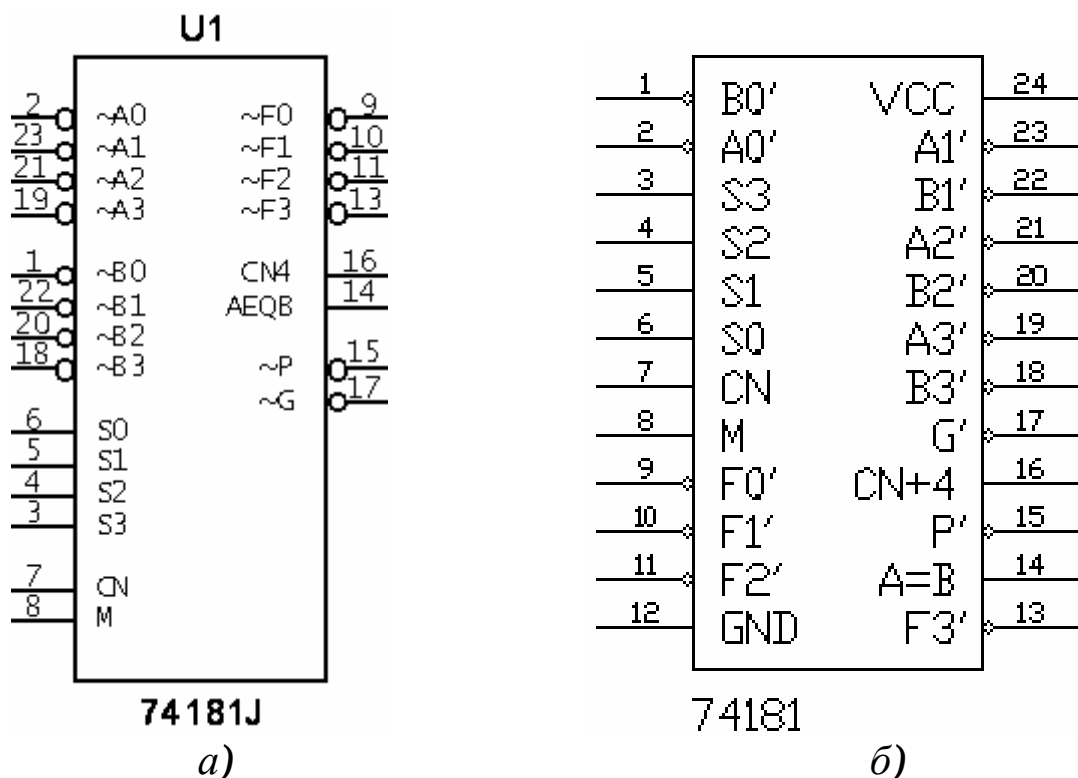


Рис. 6. ИМС 74181 в программе Multisim (а) и EWB 5.12 (б)

Значения четырехразрядных операндов A и B задаются обычно с помощью генератора слова и в шестнадцатеричном коде отображаются на алфавитно-цифровых индикаторах. На выходах $F0, F1, F2, F3$ результат суммирования высвечивается на цифровых индикаторах. При коде 1111 на этих выходах и при равенстве операндов $A=B$ выход переводится в единичное состояние.

Изменяя состояния сигналов на управляющих входах, можно промоделировать большинство функций АЛУ, используемых в микропроцессорах.

Сделайте выводы о работе сумматоров и вычитателей.

Контрольные вопросы и задачи

1. Назовите типы арифметических сумматоров.
2. Чем отличается полусумматор от полного сумматора?
3. Где используются арифметические сумматоры?
4. Аналитические выражения функций суммы и переноса арифметических сумматоров.
5. Назовите входы, выходы и назначение сумматора и полусумматора.

6. Составьте таблицы истинности сумматора и полусумматора
7. Начертите схему последовательного многоразрядного сумматора.
8. Начертите схему параллельного сумматора с последовательным переносом.
9. Начертите схему параллельного сумматора с параллельным переносом.
10. Какова структура и принцип работы сумматоров с групповой структурой (схема группового сумматора с цепным переносом, параллельным переносом, с условным переносом)?
11. В чем состоит отличие сумматора на ИМС К555ИМ3 и сумматора К555ИМ6?
12. Какова структура и принцип работы накапливающего сумматора?
13. Начертите временные диаграммы сумматора и полусумматора.
14. Составьте схему сумматора на логических элементах (И, НЕ, ИЛИ-НЕ, И-НЕ, Исключающее ИЛИ...).
15. Полусумматор на логических элементах (И, НЕ, ИЛИ-НЕ, И-НЕ, Исключающее ИЛИ...).
16. Составьте схему сумматора, способного реализовать сложение и вычитание многоразрядных чисел.
17. Составьте схему вычитателя для младших разрядов чисел.
18. Сложите два отрицательных числа с помощью сумматора схематически.
19. Сложите два числа с помощью сумматора схематически, когда одно из них отрицательное.
20. Составьте схемы полувычитателя и полного вычитателя. Опишите входы и выходы, принцип работы и их таблицы истинности.
21. Составьте схему многоразрядного вычитателя. Опишите входы и выходы, принцип работы и их таблицы истинности.

Задачи на перевод из одной системы счисления в другую, сложение (арифметическое и логическое), вычитание, умножение (арифметическое и логическое), деление чисел:

1. Переведите данное число из десятичной системы счисления в двоичную:

а) 464_{10} ; б) $380,1875_{10}$; в) $115,94_{10}$ (получить пять знаков после запятой в двоичном представлении).

б) Преобразуйте следующие десятичные числа в двоичные:

$0_{10}, 1_{10}, 18_{10}, 25_{10}, 32_{10}, 64_{10}, 69_{10}, 128_{10}, 145_{10}, 1001_{10}$.

в) Преобразуйте следующие шестнадцатеричные числа в двоичные:

$8A_{16}, B7_{16}, 6C_{16}, FF_{16}, 35_{16}, 58_{16}, 99_{16}, F2_{16}, 123_{16}, 350_{16}$.

г) Преобразуйте следующие двоичные числа в шестнадцатеричные:

$1111101_2, 111111100011_2, 101010101_2, 10000011101_2, 110101110001_2, 11001_2, 111010001_2$.

д) Преобразуйте следующие шестнадцатеричные числа в десятичные: $35_{16}, 38_{16}, 4A_{16}, A4_{16}, B5_{16}, 1B_{16}, 18_{16}, 1F_{16}, 1C_{16}$.

е) Преобразуйте следующие десятичные числа в шестнадцатеричные:

$100_{10}, 51_{10}, 45_{10}, 99_{10}, 76_{10}, 32_{10}, 48_{10}, 21_{10}, 77_{10}$.

2. Сложите числа:

а) $10000000100_2 + 111000010_2 = 10111000110_2$.

б) $223,2_8 + 427,54_8 = 652,74_8$.

в) $3B3,6_{16} + 38B,4_{16} = 73E,A_{16}$.

3. Выполните вычитание:

а) $1100000011,011_2 - 101010111,1_2 = 110101011,111_2$.

б) $1510,2_8 - 1230,54_8 = 257,44_8$.

в) $27D,D8_{16} - 191,2_{16} = EC,B8_{16}$.

4. Найдите разницу двоичных чисел a и b , десятичные эквиваленты которых соответственно равны 18 и 7, выразив их в дополнительном коде.

5. Сложите двоичные числа a и b , десятичные эквиваленты которых соответственно равны 34 и -15, выразив их в прямом коде.

6. Найдите сумму двоичных чисел a и b , десятичные эквиваленты которых соответственно равны -23 и -12, выразив их в обратном дополнительном коде.

Тема 3

ЦИФРОВОЙ КОМПАРАТОР

Цель работы:

Изучить устройство и принцип работы одноразрядного цифрового компаратора.

Необходимое оборудование в программе EWB:

1. Модуль ключей либо генератор слова (Word Generator).
2. Модуль светодиодных индикаторов.
3. Модули логических элементов.
4. Источники напряжения.

Краткие теоретические сведения.

Цифровые компараторы (от английского *compare* — сравнивать) выполняют сравнение двух чисел A и B одинаковой разрядности, заданных в двоичном или двоично-десятичном коде. В зависимости от схемного исполнения компараторы могут определять равенство $A=B$ или неравенства $A<B$, $A>B$. Результат сравнения отображается в виде логического сигнала на одноименных выходах.

Цифровые компараторы применяются для выявления нужного числа (слова) в цифровых последовательностях, для отметки времени в часовых приборах, для выполнения условных переходов в вычислительных устройствах, а также в адресных селекторах. Цифровые компараторы можно использовать в системах автоматического контроля и регулирования.

В сериях цифровых элементов обычно имеются компараторы с тремя выходами: «равно», «больше» и «меньше» (рис. 7).

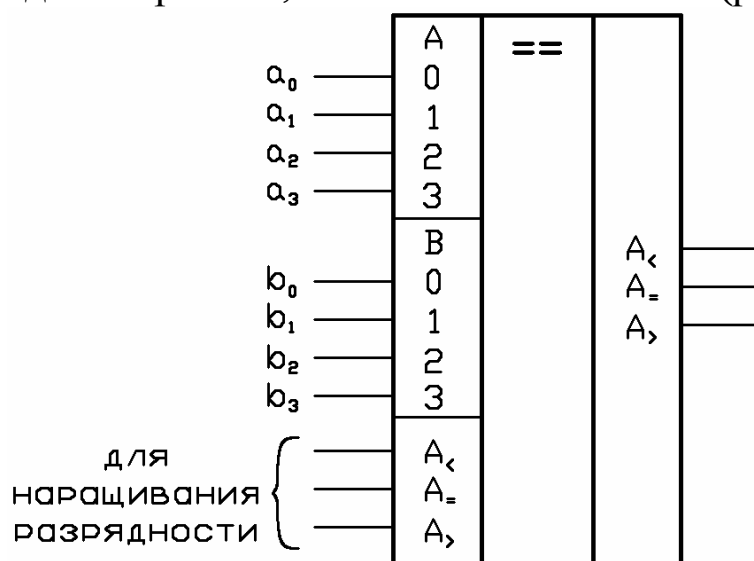


Рис. 7. Условное обозначение компаратора с тремя выходами

Схема одноразрядного компаратора приведена на рис. 8. Компаратор состоит из двух элементов НЕ, двух элементов И и одного элемента ИЛИ-НЕ.

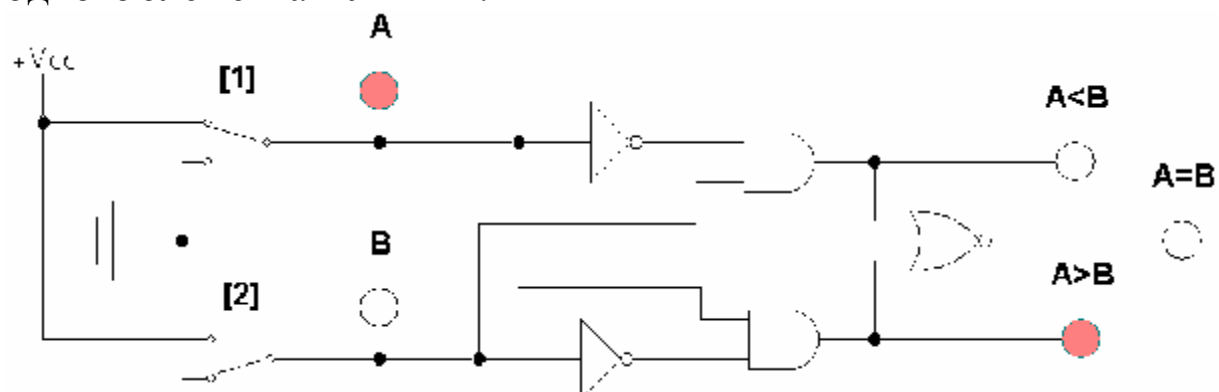


Рис. 8. Схема одноразрядного цифрового компаратора

Задание 1. Соберите схему одноразрядного компаратора как показано на рис. 8 и проверьте его работу. По результатам экспериментов составьте таблицу истинности для сравнения двух одноразрядных чисел A и B .

Задание 2. Соберите схему одноразрядного компаратора с учетом старших разрядов многоразрядного числа как показано на рис. 9.

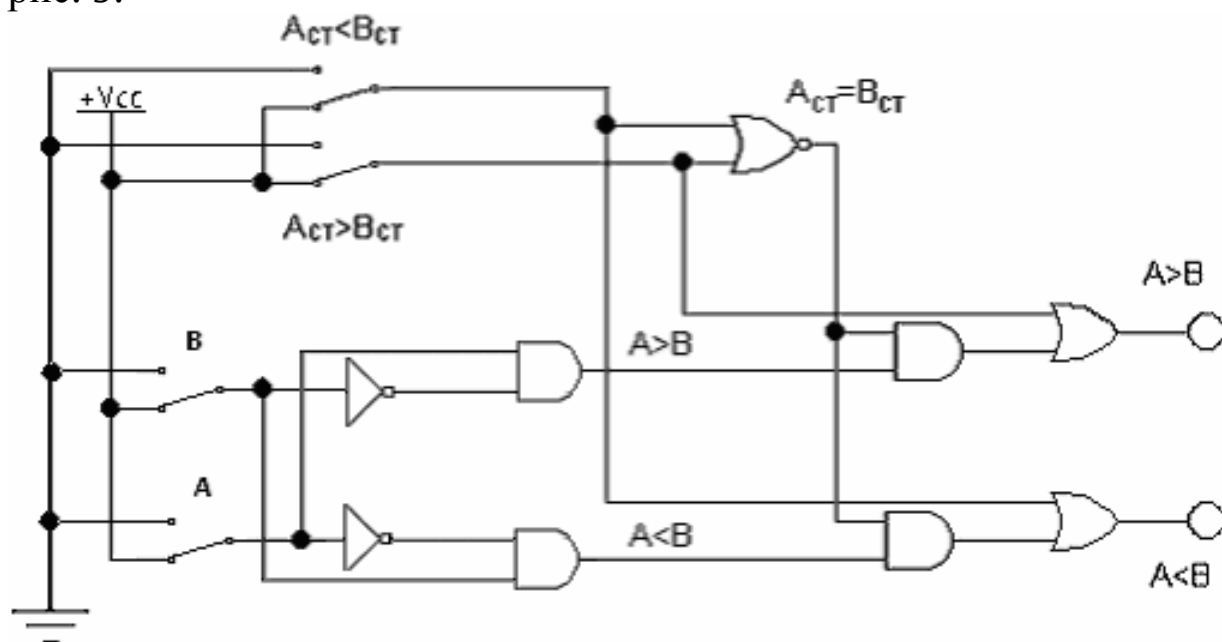


Рис. 9. Схема одноразрядного цифрового компаратора с учетом старших разрядов многоразрядного числа

Задание 3. Соберите схему многоразрядного компаратора в соответствии с заданным вариантом по его разрядности, взяв за основу 3-х разрядный цифровой компаратор на рис. 10. Для этого объедините полученную схему на рис. 9 в блок с именем **cmp**. В

программе EWB для этого используйте пиктограмму , задайте имя **cmp** в диалоговом окне *Subcircuit* и нажмите на кнопку *Copy from Circuit*. Для добавления необходимого количества блоков **cmp** используйте пиктограмму .

Варианты:

Номер варианта	1	2	3	4	5	6
разрядность	4	5	6	7	8	9

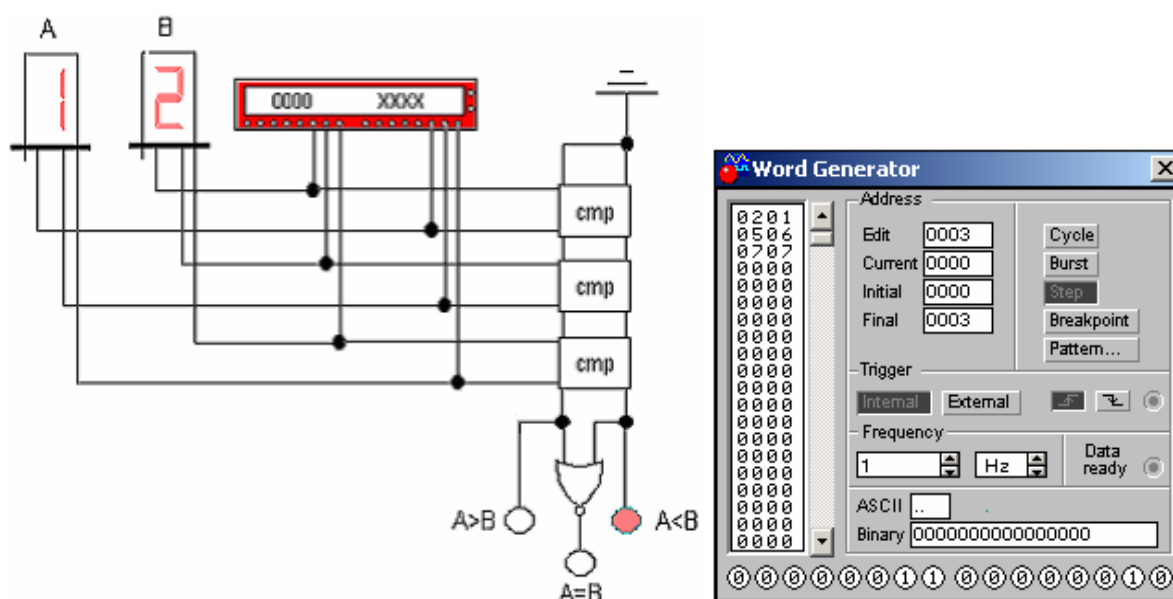


Рис. 10. Схема 3^x разрядного цифрового компаратора и генератор слова (Word Generator)

Сделайте выводы о том, с каких разрядов и почему начинается работа многоразрядных цифровых компараторов.

Контрольные вопросы и задания

1. Какие функции выполняет цифровой компаратор, в каких устройствах он может быть использован?
2. Чем обуславливается количество входов и выходов цифрового компаратора?
3. Начертите схему одноразрядного компаратора на логических элементах.
4. С помощью параллельного сумматора с последовательным переносом сравните два 4^x разрядных числа.
5. С каких разрядов начинается работа в многоразрядных компараторах?
6. Составьте таблицу истинности одноразрядного компаратора.

7. Составьте таблицу истинности для 4^x разрядного цифрового компаратора?
8. Постройте временную диаграмму работы цифрового компаратора?
9. Составьте схему цифрового компаратора для установления равенства двух двухразрядных кодов.
10. Составьте схему цифрового компаратора для определения неравенства двух двухразрядных кодов.
11. Сравните два положительных числа $7B,24_{16}$ и $173,53_8$.
12. Сравните два отрицательных числа $-A93,56_{16}$ и $-5224,56_8$.

Тема 4

УСТРОЙСТВО КОНТРОЛЯ ЧЕТНОСТИ

Цели работы:

1. Изучить принцип работы устройства контроля четности.

Необходимое оборудование:

1. Модуль ключей, генератор слова (Word Generator), логический анализатор (Logic Converter).
2. Модуль светодиодных индикаторов.
3. Модули логических элементов.
4. Источники напряжения.

Краткие теоретические сведения.

Контролем четности (parity) называется процедура, позволяющая определять ошибки или сбои вызванные, либо наводками, либо неисправностью микросхем памяти или другими причинами. Суть метода состоит в том, что к каждому байту прибавляется логический бит единицы или не прибавляется для того, чтобы количество логических единиц в байте было четным. При сбоях или наводках количество логических единиц может быть изменено и контроль четности может это выявить. В настоящее время контроль четности перестает быть актуальным, так как микросхемы памяти стали гораздо более надежными, но в случаях, когда требуется особо высокий уровень достоверности передачи информации, контроль четности по-прежнему применяется. Модули памяти с контролем четности отличаются от модулей без контроля четности наличием нечетного количества микросхем памяти.

Интегральная микросхема представляет собой восьмиразрядную схему контроля четности K155ИП2 (Зарубежные аналоги SN74180N, SN74180J) и показана на рис. 11.

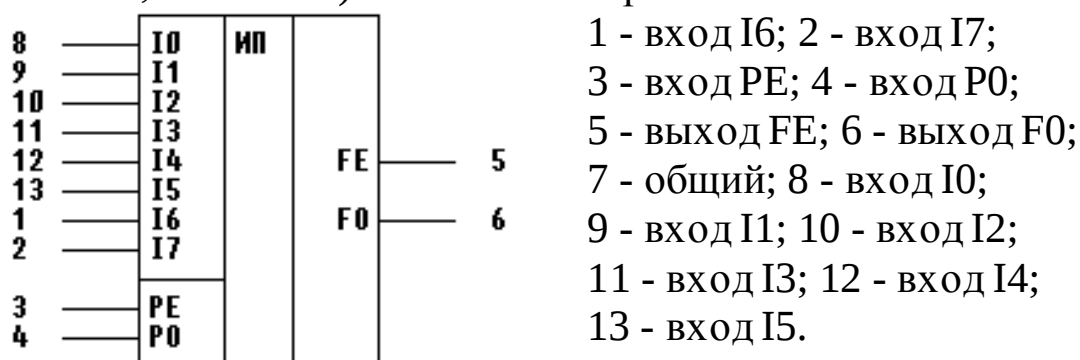


Рис. 11. Условное графическое обозначение восьмиразрядной схемы контроля четности K155ИП2

Схема формирования бита четности для четырехразрядного кода показана на рис. 12. Она содержит четыре элемента Искключающее ИЛИ, выполняющих функции сумматоров по модулю 2 (без переноса) и состоит из трех ступеней. На первой ступени попарно суммируются все биты исходного кода на входах *A*, *B*, *C*, *D*. На второй ступени анализируются сигналы первой ступени и устанавливается четность или нечетность суммы входного кода. На третьей ступени полученный результат сравнивается с контрольным сигналом на входе *E*, задающим вид используемого контроля, в результате чего на выходе *F* формируется пятый дополнительный бит четности, сопровождающий информационный сигнал в канале передачи.

Формирователь бита четности на ИМС 74280 (рис. 13 а) имеет 9 входов (*A*, *B*, ..., *I*) и два выхода (*EVEN*, *ODD*), один из которых — инверсный. Вход *I* используется для управления видом контроля (0 — контроль четности, 1 — контроль нечетности). Вывод *NC* — not connection — пустой, т.е. внутри ИМС к нему ничего не подключено.

Задание 1. Соберите схему формирователя бита четности 4^x разрядного кода, показанного на рис. 12.

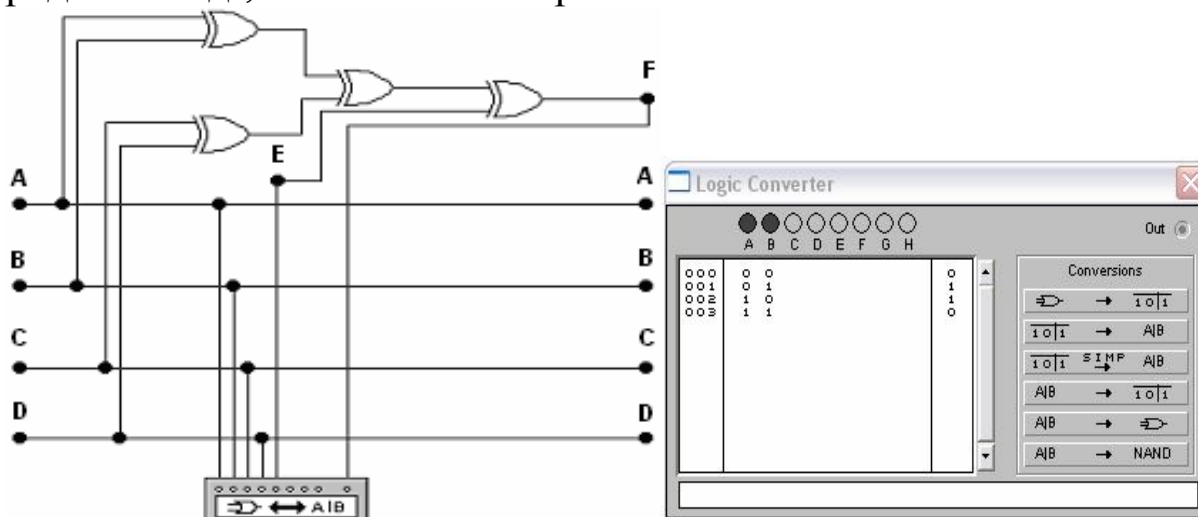
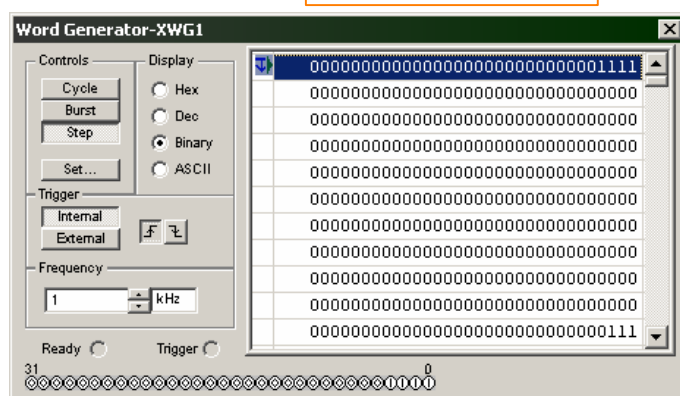
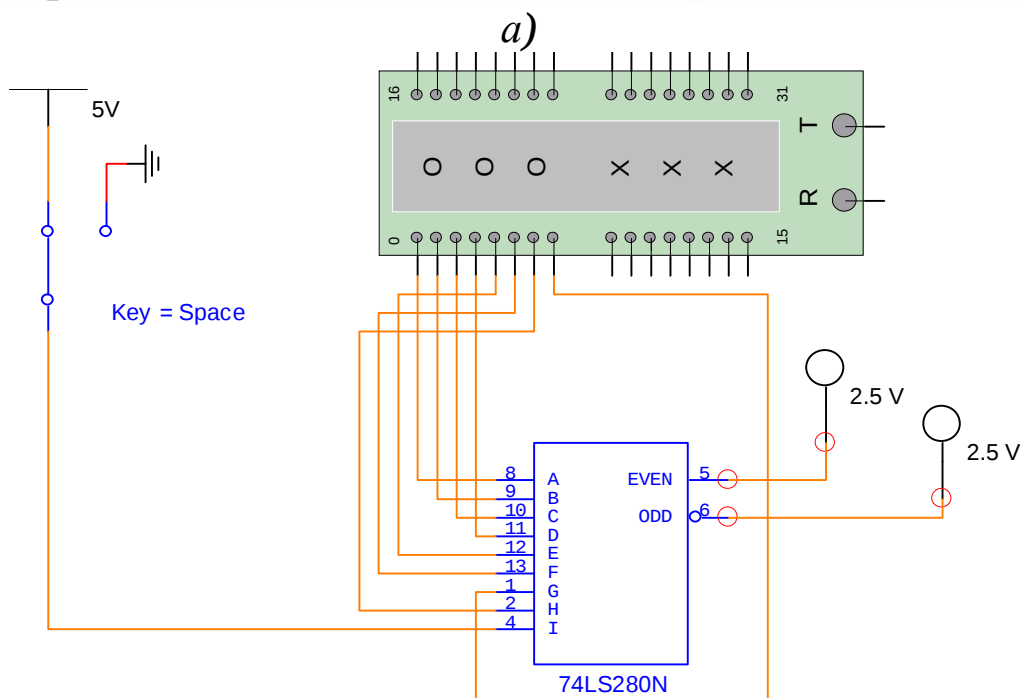
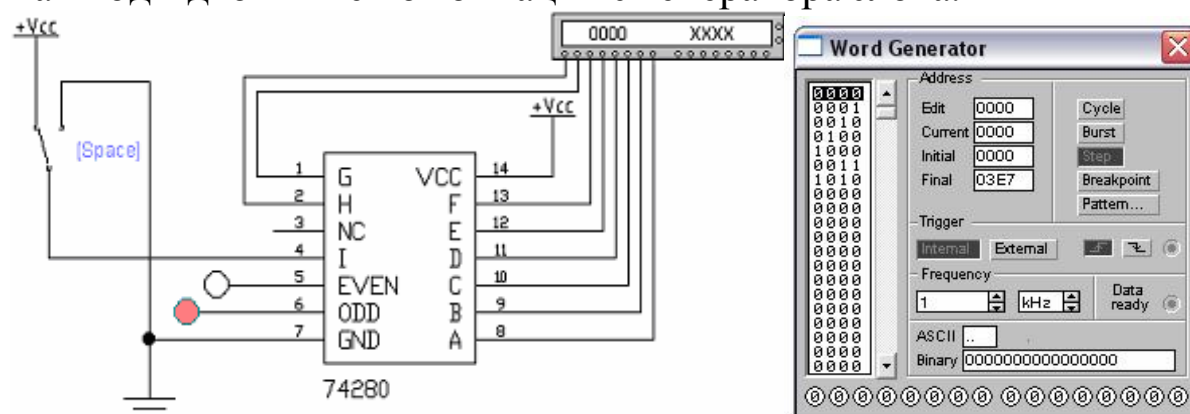


Рис. 12. Схема формирователя бита четности 4^x разрядного кода

Задание 2. Составьте таблицу истинности, для проверки работы 4^x разрядного формирователя бита четности.

Задание 3. Соберите схему включения формирователя бита четности на ИМС 74280, показанного на рис. 13 (название Sparsе позволяет переключать ключ при нажатии на клавишу пробел).

Проверьте правильность функционирования этой схемы, подавая на входы двоичные комбинации с генератора слова.



б)

Рис. 13. Схема включения ИМС 74280.

а) – в программе EWB5.12, б) – в программе Multisim

Задание 4. Проведите испытания ИМС формирователя бита четности в соответствии со своим вариантом задания (взять у преподавателя), представленном в Приложении 1 в раз-

деле *9-Bit Parity Generator* (таб. 4). В программе Multisim используйте библиотеку *TTL*.

Сделайте выводы о работе и устройстве формирователя бита четности.

Контрольные вопросы и задания

1. Что такое устройство контроля четности? Где применяется?
2. Операция контроля четности двоичных чисел позволяет повысить достоверность передачи и обработки информации. В чем заключается сущность этой операции?
3. Какое назначение имеют формирователи кода четности, где они могут быть использованы?
4. Составьте схему формирователя бита четности на логических элементах для проверки многоразрядного кода на четность или на нечетность, например, 695_{10} , $B7_{16}$, 356_8 .
5. Какая форма контроля четности чаще всего используется на практике, в частности, в Вашем компьютере, если в нем установлены модули ОЗУ с нечетным числом микросхем?
6. Объясните функционирование схемы проверки на четность и нечетность ИМС 74280 (Российский аналог К555ИП5)..
7. Начертите устройство контроля четности, используя различные логические элементы (НЕ, И, ИЛИ, И-НЕ, ИЛИ-НЕ, ИСКЛЮЧАЮЩЕЕ-ИЛИ).

Тема 5

ШИФРАТОРЫ И ДЕШИФРАТОРЫ

Цели работы:

1. Изучить устройство и принцип работы шифраторов.
2. Изучить устройство и принцип работы дешифраторов.

Необходимое оборудование:

1. Модуль ключей (с 8 ключами).
2. Модуль светодиодных индикаторов.
3. Модуль шифратора 8 разрядов в 3.
4. Модуль "дешифратора двоично-десятичного кода в 7 сегментов".
5. Модуль "7-сегментного светодиодного индикатора".
6. Источники напряжения.

Краткие теоретические сведения.

5.1. Шифраторы

Шифратор (кодер – от англ. *coder*) преобразует десятичные числа в двоичную систему счисления, причем каждому входу может быть поставлено в соответствие десятичное число, а совокупность выходных логических сигналов соответствует определенному двоичному коду. Шифраторы выпускаются приоритетными и неприоритетными. У приоритетного шифратора входы имеют разный приоритет. Возбужденный вход с большим приоритетом подавляет действие прежде возбужденного и устанавливает на выходах код, соответствующий своему значению. Шифратор, имеющий 2^n входов и n выходов, называется полным. Если число входов шифратора меньше 2^n , он называется неполным. Поведение шифратора $(8 \times 3)^1$ описывается таблицей истинности, в соответствии с которой алгоритм работы описывается системой уравнений вида:

$$Q_2 = x_4 + x_5 + x_6 + x_7,$$

$$Q_1 = x_2 + x_3 + x_6 + x_7,$$

$$Q_0 = x_1 + x_3 + x_5 + x_7,$$

где Q_0, Q_1, Q_2 – выходы шифратора, x – входы.

В шифраторе рассматриваемого типа сигнал, подаваемый на вход x_0 , не используется. Поэтому отсутствие сигнала на любом

¹ $\times$ - символ не является знаком умножения, а показывает преобразование количества входных сигналов в выходные

из входов $x_0, \dots, x_1$ трактуется схемой как наличие на входе нулевого сигнала.

На рис. 14 представлен шифратор на 4 входа.

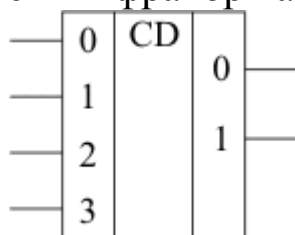


Рис. 14. Условно-графическое обозначение шифратора на 4 входа

5.2. Дешифраторы

Дешифратор (декодер – от англ. *decoder*) преобразует n -разрядный двоичный код в логический сигнал, появляющийся на том выходе, десятичный номер которого соответствует двоичному коду. Дешифраторы могут быть полными и неполными. Полные дешифраторы реагируют на все входные коды (число адресных входов n связано с числом его выходов m соотношением $m = 2^n$), неполные – на коды ($m < 2^n$), величина которых не превосходит некоторого заранее установленного значения. Выходы дешифраторов могут быть прямыми и инверсными. Поведение дешифратора описывается таблицей истинности, в соответствии с которой алгоритм работы описывается системой уравнений вида:

$$x_0 = \overline{Q_3} \overline{Q_2} \overline{Q_1} \overline{Q_0},$$

$$x_1 = \overline{Q_3} \overline{Q_2} \overline{Q_1} Q_0,$$

$$x_2 = \overline{Q_3} \overline{Q_2} Q_1 \overline{Q_0},$$

.....

и так далее, где Q_i – значение логической переменной на i -м входе устройства. В общем виде эта система имеет вид $x_i = (Q_3 Q_2 Q_1 Q_0)_i$, где x_i – сигнал на i -м выходе дешифратора; $(Q_3 Q_2 Q_1 Q_0)_i$ – конstituента единицы, соответствующая двоичному коду i -го десятичного кода.

Условно-графическое обозначение дешифратора на три входа приведено на рис. 15 а и дешифратора с инверсными выходами б.

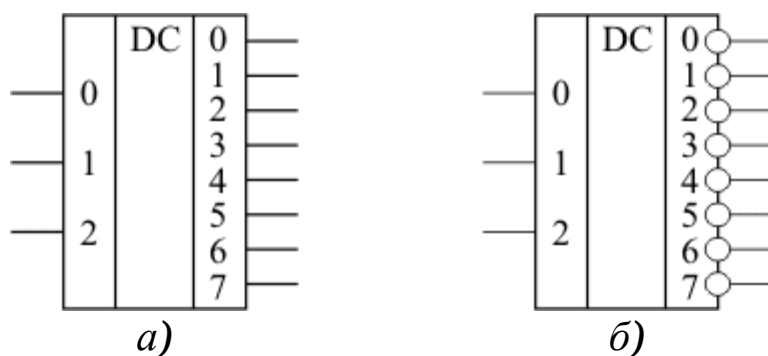


Рис. 15. Условно-графическое обозначение.

а) трехвходовой дешифратор; б) дешифратор с инверсными выходами

На рис. 15 а представлен дешифратор, номер разряда которого устанавливается "1" на выходе дешифратора, определяется кодом на его входах. На выходах дешифратора, представленного на рис. 15 б образуется унитарный код, содержащий один и только один ноль. Например, если входные сигналы имеют значение $110_2 = 6_{10}$, то выходы такого дешифратора будут находиться в состоянии 10111111, то есть выход 6 будет иметь значение, отличное от остальных выходов.

Дешифраторы широко применяются в различных устройствах компьютеров. Прежде всего, они используются для выбора ячейки запоминающего устройства, к которой производится обращение для записи или считывания информации. При этом часть разрядов адресного кода может дешифроваться дешифраторами, выполненными в виде отдельных интегральных схем, а другая часть разрядов (обычно младшая) дешифруется с помощью дешифраторов, встроенных непосредственно в БИС запоминающего устройства. Кроме того, дешифраторы находят применение в устройстве управления для определения выполняемой операции, построения распределителей импульсов и в других блоках.

Задание 1. Соберите приоритетный шифратор 8×3 на логических элементах и составьте таблицу истинности. За основу можно взять приоритетный шифратор 4×2 (рис. 16).

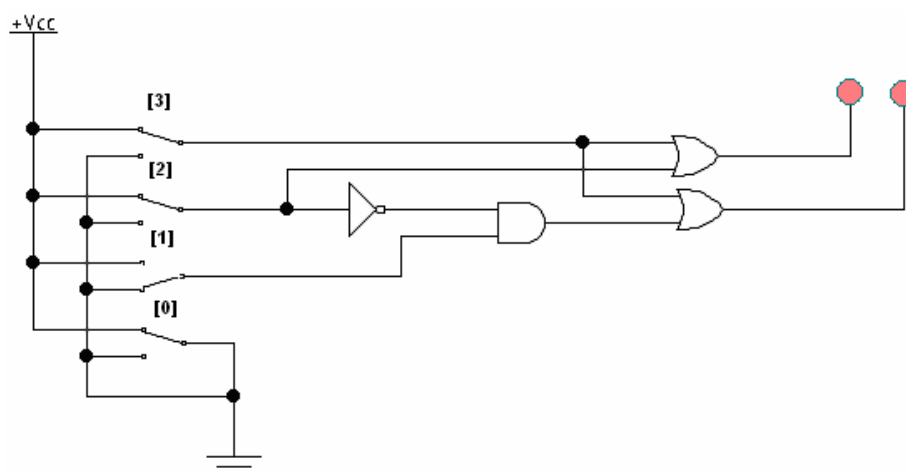


Рис. 16. Схема приоритетного шифратора 4'2 на логических элементах

Задание 2. Соберите приоритетный шифратор 8×3 на ИМС 74148 (Российский аналог К155ИБ1) как показано на рис. 17, и, меняя кодовые комбинации в генераторе слова, показать, в чем заключается смысл термина «приоритетный».

При моделировании необходимо обратить внимание на реализацию этого принципа, поскольку, например, возможно нажатие сразу нескольких клавиш 9, 5 и 2. На выходе шифратора будет генерироваться код 1001, соответствующий цифре 9 (приоритет старшего разряда). При моделировании следует учесть, что все входы и выходы — инверсные (на функциональной схеме ИМС в программе EWB они ошибочно показаны прямыми).

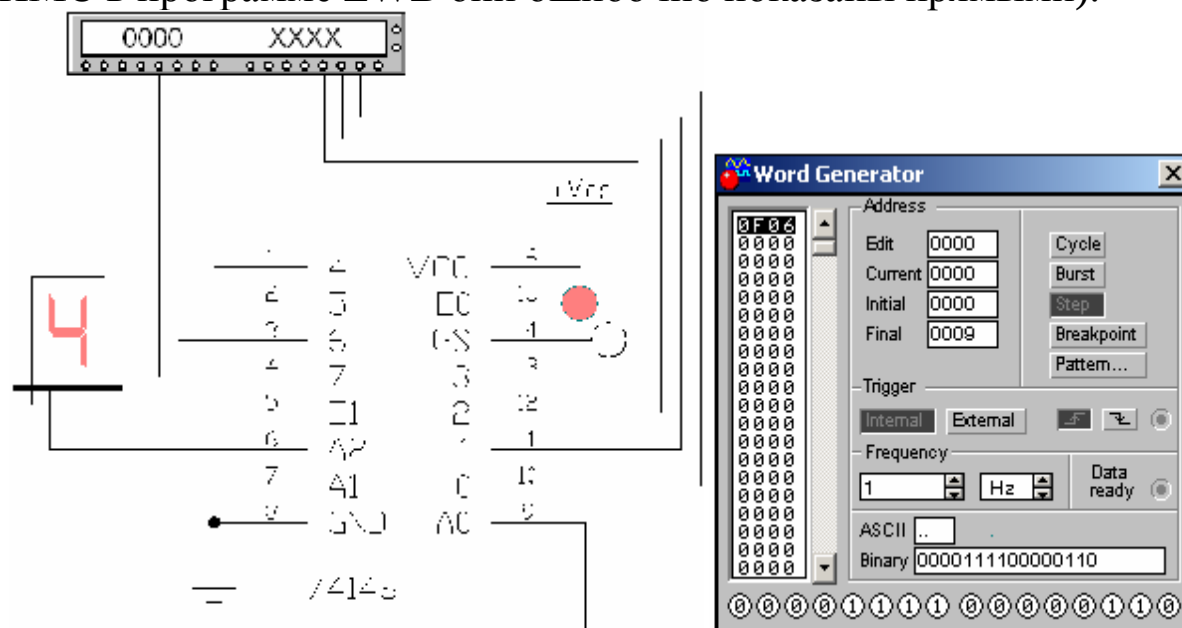


Рис. 17. Схема включения ИМС 74148 и панель генератора слова с установками для его испытания

Назначение выводов ИМС 74148: $0\div7$ — информационные входы; $A0, A1, A2$ — информационные выходы; $E1$ — вход разрешения; EO, GS — выходы для каскадирования шифраторов.

Задание 3. Соберите дешифратор 3×8 с входом разрешения на логических элементах, как показано на примере дешифратора 2×4 (рис. 18).

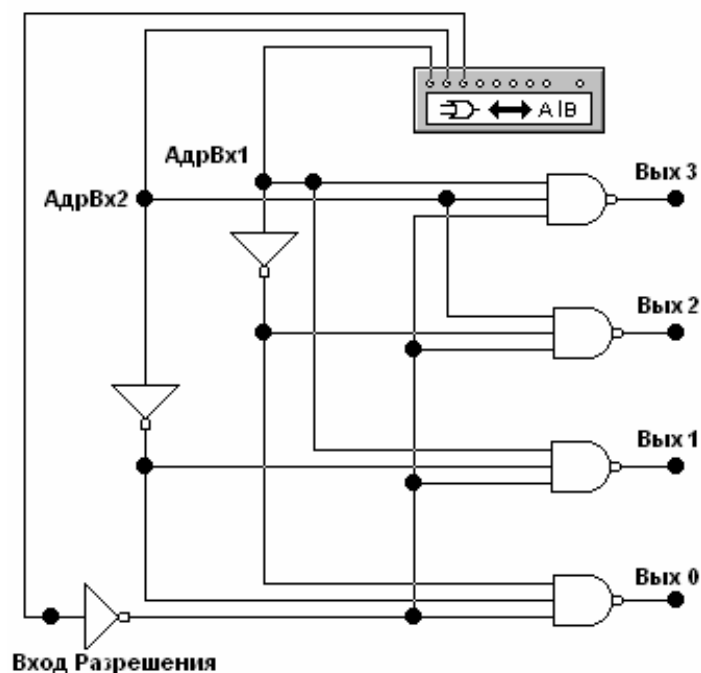


Рис. 18. Схема дешифратора 2×4 с входом разрешения на логических элементах

Задание 4. Соберите дешифратор 4×16 на ИМС 74154 как показано на рис. 19 и объясните принцип его работы, используя генератор слова, светодиодные индикаторы.

3. У словное графическое и буквенное обозначения шифраторов и дешифраторов.
4. Почему шифраторы и дешифраторы называют комбинационными устройствами?
5. Составьте таблицу истинности шифратора.
6. Составьте таблицу истинности дешифратора.
7. Как называется дешифратор, у которого используются не все выходы.
8. Какие меры предпринимаются для увеличения числа входов и выходов дешифраторов. Какие еще дополнительные входы и выходы используются в дешифраторах и шифраторах?
9. При решении каких задач цифровой техники используется дешифратор?
10. Составьте схемы дешифраторов 2×4 , 3×8 , 4×16 на логических элементах.
11. Как осуществляется наращивание размерности дешифратора и приоритетного шифратора?
12. Что обуславливает работу дешифратора в режиме демультиплексора?
13. Каскадное включение дешифраторов.
14. Составьте схемы шифраторов 4×2 , 8×3 , 16×4 на логических элементах.

Тема 6

МУЛЬТИПЛЕКСОРЫ И ДЕМУЛЬТИПЛЕКСОРЫ

Цели работы:

1. Изучить устройство и принцип работы мультиплексора
2. Изучить устройство и принцип работы демультиплексора

Необходимое оборудование:

1. Модуль ключей или генератор слова (Word Generator).
2. Модули логических элементов.
4. Модули светодиодных индикаторов.

Краткие теоретические сведения.

6.1. Мультиплексоры

Мультиплексор осуществляет подключение одного из входных каналов к выходному под управлением управляющего (адресующего) слова.

На рис. 20 приведена схема двухканального мультиплексора, состоящего из элементов ИЛИ, НЕ и двух элементов И.

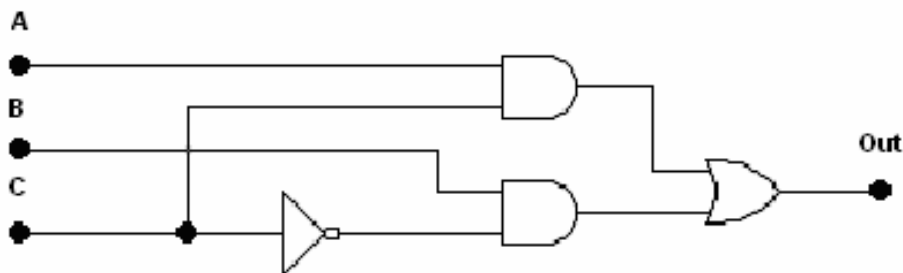


Рис. 20. Схема мультиплексора на логических элементах

Наращивание размерности мультиплексоров возможно с помощью пирамидальной структуры из нескольких мультиплексоров. При этом первый ярус схемы представляет собою столбец, содержащий столько мультиплексоров, сколько необходимо для получения нужного числа информационных входов. Все мультиплексоры столбца адресуются одним и тем же кодом, составленным из соответствующего числа младших разрядов общего адресного кода (если число информационных входов схемы равно 2^n , то общее число адресных разрядов равно n , младшее поле n_1 адресного кода используется для адресации мультиплексоров первого яруса). Старшие разряды адресного кода, число которых равно $n - n_1$, используются во втором ярусе, мультиплексор которого обеспечивает поочередную работу мультиплексоров первого яруса на общий выходной канал.

Пирамидальная схема, выполняющая функции мультиплек-

сора "32-1" и построенная на мультиплексорах меньшей размерности, показана на рис. 21. (сокращение MUX от английского MultipleXer).

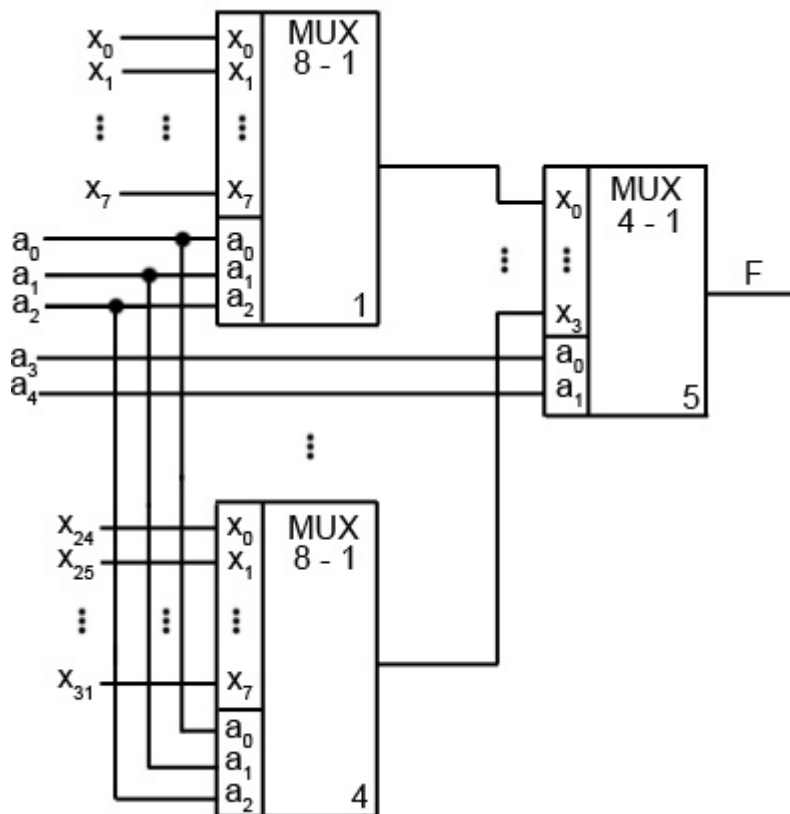


Рис. 21. Схема наращивания мультиплексоров

6.2. Демультимплексоры

Демультимплексор – это комбинационное устройство, выполняющее операцию, обратную операции мультиплексора – передают данные из одного входного канала в один из нескольких каналов-приемников.

Схема демультимплексора на рис. 22 состоит из элемента И и одного элемента НЕ, адресный вход – A , информационный вход – X .

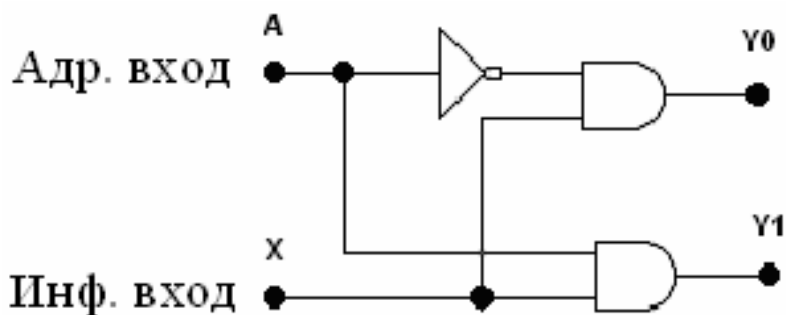


Рис. 22. Схема демультимплексора на логических элементах

Отдельные демультимплексоры могут отсутствовать, а дешифратор с входом разрешения часто называется дешифратором-

демультиплексором(рис. 23).

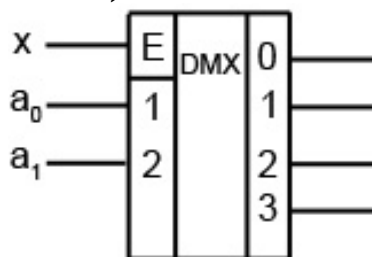


Рис. 23. Условное обозначение дешифратора-демультиплексора

Задание 1. Постройте мультиплексор в EWB или в Multisim с двумя адресными входами на логических элементах, проверяя его работу, и составьте таблицу истинности.

Задание 2. Подготовьте схему в EWB или в Multisim и проведите испытания ИМС 74151 (Российский аналог К155КП5), для чего на входы 1 и 2, 3 и 4 подайте постоянные уровни сигналов логического нуля и единицы соответственно, а на остальные информационные, адресные и вход управления – сигналы с генератора слова в такой комбинации, чтобы при каждом шаге информация передовалась последовательно через шаг с первого по восьмой каналы; для индикации состояния выходов используйте логический анализатор.

Задание 3. Постройте демультиплексор в EWB или в Multisim с четырьмя информационными выходами на логических элементах, проверяя его работу, и составьте таблицу истинности.

Задание 4. Используя методику анализа двухканального мультиплексора с помощью логического преобразователя, исследуйте внутреннюю структуру сдвоенного четырехканального мультиплексора на ИМС 74153 (Российский аналог К155КП2) .

Задание 5. В соответствии с индивидуальным вариантом (Приложение 1, таб. 4 и таб. 5) изучить работу устройства (по варианту может быть: дешифратор, шифратор, мультиплексор или демультиплексор), используя для этого ключи или генератор слова, светодиодные индикаторы.

Сделайте выводы о принципе действия мультиплексора и демультиплексора

Контрольные вопросы и задания

1. Что такое мультиплексор каково его назначение?
2. Что такое демультиплексор для решения каких задач его можно применить?
3. Как работает дешифратор в режиме демультиплексора?

Тема 7

АРИФМЕТИКО-ЛОГИЧЕСКОЕ УСТРОЙСТВО

Цель работы:

Изучить устройство и принцип работы арифметико-логического устройства (АЛУ).

Необходимое оборудование:

1. Модуль ключей или генератор слова (Word Generator).
2. Модуль светодиодных индикаторов.
3. Модули логических элементов.
4. Источники напряжения.
5. Модули светодиодных индикаторов.

Краткие теоретические сведения.

Арифметико-логическое устройство (АЛУ или англ. ALU – Arithmetic Logic Unit) является составной частью микропроцессора и служит для выполнения арифметических и логических операций над словами, называемыми в данном случае операндами.

Выполняемые АЛУ операции можно разделить на следующие группы:

- 1) операции двоичной арифметики для чисел с фиксированной запятой;
- 2) операции двоичной арифметики для чисел с плавающей запятой;
- 3) операции десятичной арифметики;
- 4) операции индексной арифметики (при модификации адресов команд);
- 5) операции специальной арифметики;
- 6) логические операции;
- 7) операции над алфавитно-цифровыми полями.

Современные ЭВМ общего назначения реализуют операции всех приведенных выше групп. Малые ЭВМ и микропроцессоры часто не имеют аппаратуры арифметики чисел с плавающей запятой, десятичной арифметики и операций над алфавитно - цифровыми полями. В этом случае эти операции выполняются специальными подпрограммами.

К арифметическим операциям относятся: сложение, вычитание, умножение и деление. Группу логических операций состав-

ляют операции: дизъюнкции и конъюнкции над многоразрядными двоичными словами, а также сравнение кодов на равенство.

Специальные арифметические операции включают в себя: нормализацию, арифметический сдвиг и другие.

По способу действия над операндами АЛУ делятся на последовательные и параллельные. В последовательных АЛУ операнды представляются в последовательном коде, а операции производятся последовательно во времени над их раздельными разрядами. В параллельных АЛУ операнды представляются параллельным кодом и операции совершаются параллельно во времени над всеми разрядами операндов.

По способу представления чисел различают:

1. АЛУ с фиксированной запятой,
2. АЛУ для чисел с плавающей запятой,
3. АЛУ для десятичных чисел.

По характеру использования элементов и узлов АЛУ делятся на:

- блочные;
- многофункциональные.

К блочным АЛУ относятся операции с фиксированной и плавающей запятой, выполняющиеся в отдельных блоках. При этом повышается скорость обработки, т.к. блоки могут параллельно выполнять соответствующие операции, но значительно возрастают затраты оборудования.

В многофункциональных АЛУ операции для всех форм представления чисел выполняются одними и теми же схемами, которые коммутируются (соединяются) нужным способом, в зависимости от требуемого режима работы.

По своим функциям АЛУ является операционным блоком, выполняющим микрооперации: прием из других устройств операндов, преобразование и выдача результатов преобразования в другие устройства. Управляются АЛУ устройством управления, генерирующим сигналы, которые инициируют в АЛУ определенные микрооперации, генерируемые устройством управления.

Последовательность сигналов определяется кодом операции команды и оповещающими сигналами. Обычно в АЛУ операции алгебраического сложения сводятся к арифметическому сложению кодов чисел путем применения инверсных кодов. Инверсные коды используются для представления отрицательных чисел. Об-

ратный код имеет 2 представления нуля, что затрудняет анализ результата операции. Чаще используется дополнительный код. Алгоритм выполнения АЛУ арифметических операций зависит от того, в каком виде хранятся в памяти отрицательные числа, а именно: в прямом или дополнительном коде. В последнем случае сокращается время выполнения операций за счет исключения операции преобразования, полученного в АЛУ дополнительного кода отрицательного результата в прямой код.

Схемотехнически АЛУ представляется так, как показано на рис. 24.

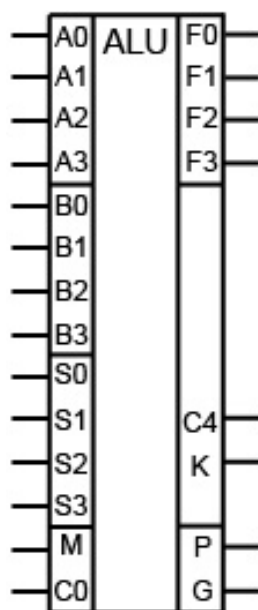


Рис. 24. Условное графическое обозначение арифметико-логического устройства

АЛУ имеет входы чисел $A0-A3$ и $B0-B3$, входы управления $S0-S3$, M , вход переноса $C0$, выход результата $F0-F3$, выход переноса $C4$, выход равенства кодов K , выходы P и G для схемы быстрого переноса. Вход M определяет вид выполняемых операций (при $M = 1$ над A и B выполняется 16 логических операций, при $M = 0$ выполняются арифметические операции).

АЛУ состоит из регистров, сумматора с соответствующими логическими схемами и элемента управления выполняемым процессом. Устройство работает в соответствии с сообщаемыми ему именами (кодами) операций, которые при пересылке данных нужно выполнить над переменными, помещаемыми в регистры.

Арифметико-логическое устройство функционально можно разделить на две части: а) микропрограммное устройство (устройство управления), задающее последовательность микро-

манд (команд); б) операционное устройство (АЛУ), в котором реализуется заданная последовательность микрокоманд (команд).

Структурная схема АЛУ и его связь с другими блоками машины показаны на рис. 25. В состав АЛУ входят регистры $P_21 - P_27$, в которых обрабатывается информация, поступающая из оперативной или пассивной памяти $N_1, N_2, \dots, N_S$; логические схемы, реализующие обработку слов по микрокомандам, поступающим из устройства управления.

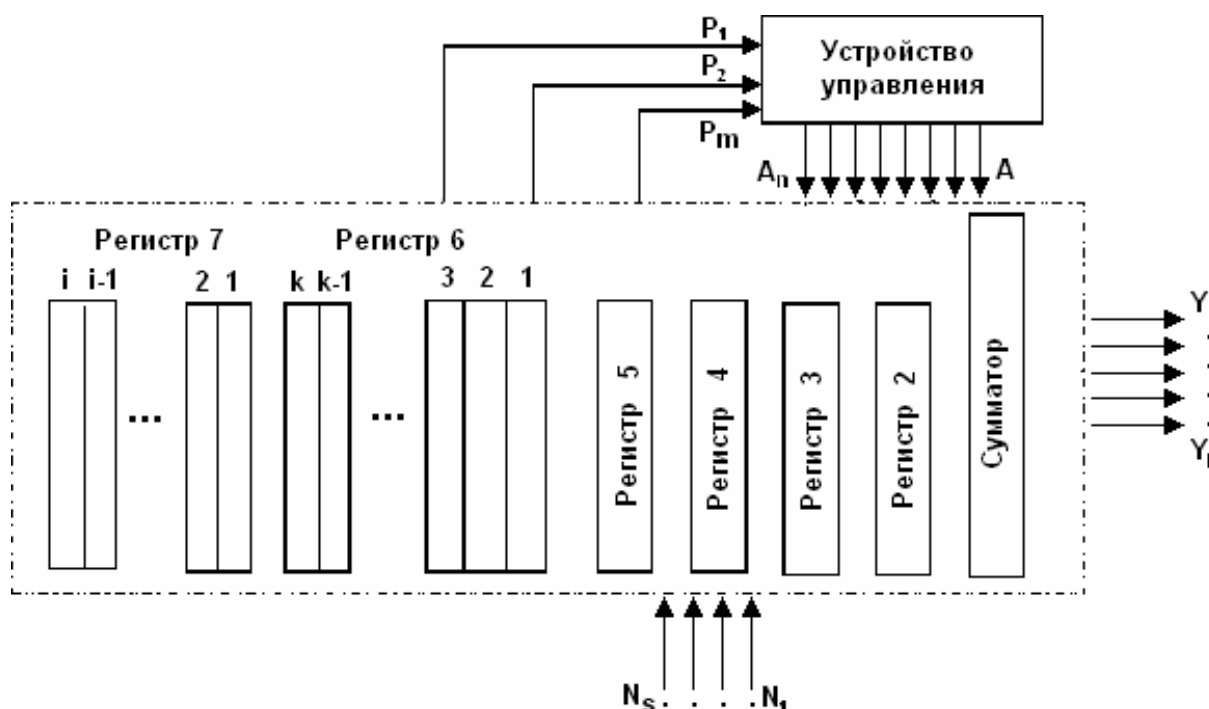


Рис. 25. Структурная схема АЛУ

Закон переработки информации задает микропрограмма, которая записывается в виде последовательности микрокоманд $A_1, A_2, \dots, A_{n-1}, A_n$. При этом различают два вида микрокоманд: внешние, поступающие в АЛУ от внешних источников и вызывающие в нем те или иные преобразования информации (на рис. 25 микрокоманды $A_1, A_2, \dots, A_n$), и внутренние, которые генерируются в АЛУ и воздействуют на микропрограммное устройство, изменяя естественный порядок следования микрокоманд. Например, АЛУ может генерировать признаки в зависимости от результата вычислений: признак переполнения, признак отрицательного числа, признак равенства 0 всех разрядов числа др. На рис. 25 эти микрокоманды обозначены $P_1, P_2, \dots, P_m$.

Результаты вычислений из АЛУ передаются по кодовым шинам записи $Y_1, Y_2, \dots, Y_r$, в ОЗУ.

Функции регистров, входящих в АЛУ: P_21 - сумматор (или сумматоры) - основной регистр АЛУ, в котором образуется результат вычислений; P_22 , P_23 - регистры слагаемых, сомножителей, делимого или делителя (в зависимости от выполняемой операции); P_24 - адресный регистр (или адресные регистры), предназначен для запоминания (иногда и формирования) адреса операндов и результата; P_26 - k индексных регистров, содержимое которых используется для формирования адресов; P_27 - i вспомогательных регистров, которые по желанию программиста могут быть аккумуляторами, индексными регистрами или использоваться для запоминания промежуточных результатов

Часть операционных регистров является программно-доступной, то есть они могут быть адресованы в команде для выполнения операций с их содержимым. К ним относятся: сумматор, индексные регистры, некоторые вспомогательные регистры. Операционные устройства можно классифицировать по виду обрабатываемой информации, по способу обработки информации и логической структуре.

Остальные регистры программно-недоступны, так как они не могут быть адресованы в программе.

АЛУ может оперировать четырьмя типами информационных объектов: булевскими (1 бит), цифровыми (4 бита), байтными (8 бит) и адресными (16 бит). В АЛУ выполняется 51 различная операция пересылки или преобразования этих данных. Так как используется 11 режимов адресации (7 для данных и 4 для адресов), то путем комбинирования "операция / режим адресации" базовое число команд 111 расширяется до 255 из 256 возможных при однобайтном коде операции.

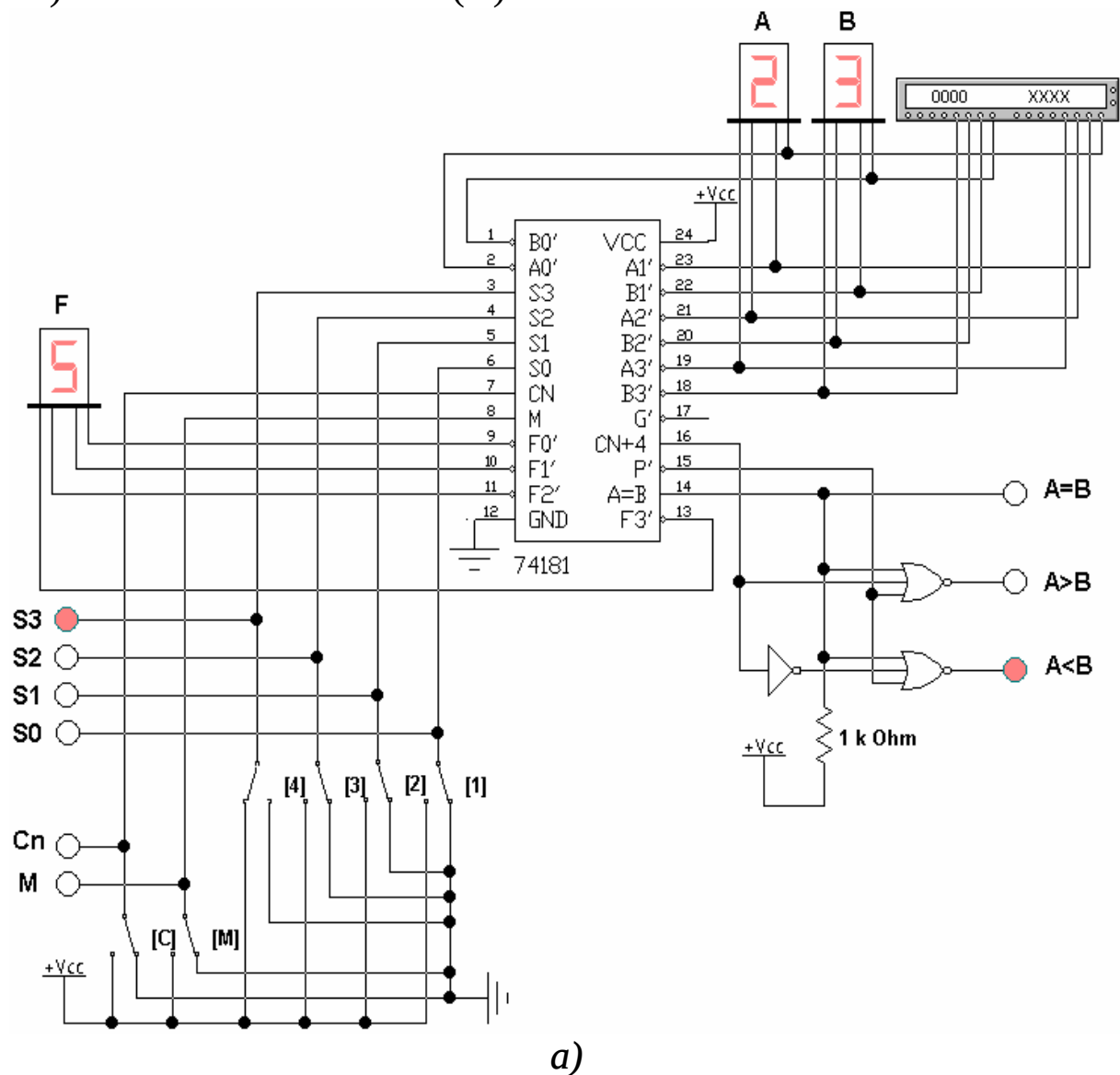
Задание 1. Проведите моделирование в EWB или в Multisim перечисленных ниже режимов АЛУ (рис. 26), предварительно составив неповторяющиеся комбинации на выходе генератора слова (согласно своему варианту):

- 1) 0000 — A^2 — передача на выход инвертированного операнда A ;
- 2) 0001 — $(A \vee B)'$ — поразрядная операция ИЛИ с инверсией над операндами A и B ;

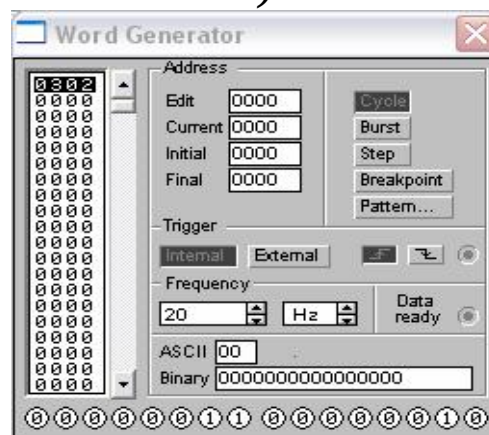
² — символ указывает на инвертирование сигнала и аналогичен символу верхнего надчеркивания, например, $\overline{F}$

- 3) 0010 — $A' \wedge B$ — операция И инвертированного операнда A и операнда B;
 - 4) 0011 — 0 — нет операции;
 - 5) 0100 — $(A \wedge B)'$ — операция И с инверсией;
 - 6) 0101 — B' — инверсия операнда B;
 - 7) 0110 — $A \oplus B$ — операция ИСКЛЮЧАЮЩЕЕ ИЛИ, команда XRA;
 - 8) 0111 — $A \wedge B'$ — операция И над операндами A и инверсией B;
 - 9) 1000 — $A' \vee B$ — операция ИЛИ над инверсией A и операндом B;
 - 10) 1001 — $(A \oplus B)'$ — операция ИЛИ с инверсией;
 - 11) 1010 — B — передача на выход операнда B;
 - 12) 1011 — $A \wedge B$ — операция И, команда ANA;
 - 13) 1100 — 1;
 - 14) 1101 — $A \vee B'$ — операция ИЛИ над инверсией B и операндом A;
 - 15) 1110 — $A \vee B$ — операция ИЛИ, команда ORA;
 - 16) 1111 — A — передача на выход операнда A.
- Арифметические операции ($M=0$) без переноса ($C_n=1$) и с переносом ($C_n=0$, данные приводятся в круглых скобках);
- 17) 0000 — A — передача на выход операнда ($A+1$ — суммирование операнда с 1 переноса, команда инкремента);
 - 18) 0001 — $A \vee B$ — операция суммирования без учета переноса, команда ADD ($(A+B)+1$ — суммирование с учетом переноса, команда ADC);
 - 19) 0010 — $A \vee B'$ — операция суммирования операнда A с инверсией операнда B без учета переноса ($(A+B') + 1$ — то же, но с учетом переноса);
 - 20) 0011 — -1 (0);
 - 21) 0100 — $A \oplus A \wedge B'$ ($A \oplus (A \wedge B)' \oplus 1$);
 - 22) 0101 — $(A \vee B) \oplus A \wedge B'$ ($((A \vee B) \oplus A \wedge B' \oplus 1)$);
 - 23) 0110 — $A - B - 1$, команда SBB ($A - B$, команда SUB);
 - 24) 0111 — $A \wedge B' - 1$ ($((A \wedge B)')$);
 - 25) 1000 — $A \oplus A \wedge B$ ($A \oplus B \oplus 1$);
 - 26) 1001 — $A \oplus B$, команда ADD ($A \oplus B \oplus 1$);
 - 27) 1010 — $(A \vee B') \oplus A \wedge B$ ($((A \vee B') \oplus A \wedge B \oplus 1)$);
 - 28) 1011 — $A \wedge B - 1$ (AB);

- 29) 1100 — $A \oplus A$ (A+A+1);
 30) 1101 — $(A \vee B) \oplus A$ $((A \vee B) \oplus A \oplus 1)$;
 31) 1110 — $(A \vee B') \oplus A$ $((A \vee B') \oplus A \oplus 1)$;
 32) 1111 — $A - 1$ (A).



a)



б)

Рис. 26. Схема АЛУ на ИМС 74181 (а) и генератор слова (б)

Задание 2. Составьте схему многоразрядного параллельного сумматора с последовательным переносом на базе ИМС 74181 согласно индивидуальному заданию.

Сделайте выводы о принципе действия АЛУ и об использовании его в структурах многоразрядных сумматоров различного типа.

Контрольные вопросы

1. Что такое арифметико-логическое устройство? Для решения каких задач его можно применить?
2. Что лежит в основе АЛУ?
3. Опишите все входы и выходы АЛУ.
4. Какие функции и какое их количество выполняют АЛУ?
5. Какие способы переносов существуют для соединения АЛУ друг с другом, их специфика.
6. Каково назначение входов и выходов АЛУ, условное графическое обозначение которого дано на рис. 27?

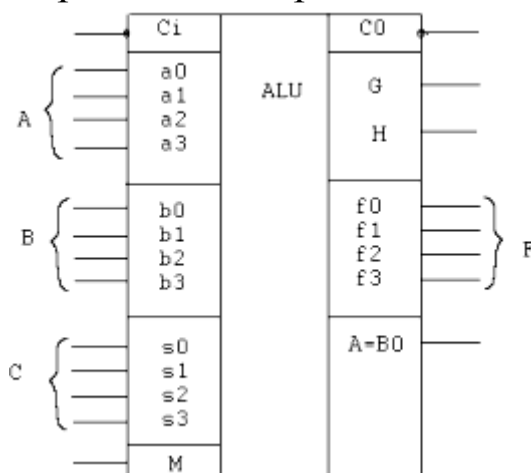


Рис. 27. Схема условного обозначения АЛУ

Требования к содержанию и оформлению отчетов по лабораторным работам

1. Название работы.
2. Цель работы.
3. Задачи работы.
4. Краткие теоретические сведения.
5. Исследуемые схемы устройств.
6. Полученные таблицы истинности.
7. Выводы, сделанные в лабораторной работе.

Таблица кодов элементов программы EWB 5.12
по вариантам заданий

NAND		NOR	Inverter	AND	OR
1) 7400	11) 7440	21) 7402	30) 7404	40) 7408	48) 7432
2) 7403	12) 74133	22) 7425	31) 7405	41) 7409	49) 4071
3) 7410	13) 74134	23) 7427	32) 7406	42) 7411	50) 4072
4) 7412	14) 74132	24) 7428	33) 7414	43) 7415	51) 4075
5) 7420	15) 4011	25) 7433	34) 7416	44) 7421	
6) 7422	16) 4012	26) 4001	35) 4009	45) 4073	
7) 7426	17) 4023	27) 4002	36) 4049	46) 4081	
8) 7430	18) 4068	28) 4025	37) 4069	47) 4082	
9) 7437	19) 4093	29) 4078	38) 40106		
10) 7438	20) 7439		39) 4502		
XOR		Encoder	Decoder	Demux	Mux
52) 7486		1) 74147Π	4) 74138	13) 74139	19) 74150
53) 74266 (XNOR)		2) 74148Π	5) 74139	14) 74154	20) 74151
54) 40303		3) 4532	6) 74145	15) 74155	21) 74153
55) 4070			7) 74154	16) 74156	22) 74157
56) 4077 (XNOR)			8) 74155	17) 74159	23) 74158
			9) 74156	18) 74238	24) 74251
			10) 74159		25) 74257
			11) 74238		26) 74258
			12) 4556		27) 74352
					28) 74353
				29) 4019	
				30) 4559	

Таблица кодов элементов программы Multisim
по вариантам заданий

NAND		NOR	Inverter	AND	OR
1) 7400n	15) 74LS12	28) 7402n	38) 7404n	42) 7408n	53) 7432n
2) 7403n	16) 74LS20	29) 7427n	39) 7405n	43) 7409n	54) 74LS32
3) 7410n	17) 74LS22	30) 7428n	40) 74LS04	44) 74LS08	55) 74HC32
4) 7412n	18) 74LS26	31) 7433n	41) 74LS05	45) 74LS09	
5) 7420n	19) 74LS30	32) 74LS02		46) 74LS11	
6) 7422n	20) 74LS37	33) 74LS27		47) 74LS15	
7) 7426n	21) 74LS38	34) 74LS28		48) 74LS21	
8) 7430n	22) 74LS40	35) 74LS33		49) 74HC08	
9) 7437n	23) 74HC00	36) 74HC02		50) 74HC09	
10) 7438n	24) 74HC10	37) 74HC27		51) 74HC11	
11) 7439n	25) 74HC20			52) 74HC21	
12) 7440n	26) 74HC30				
13) 74LS00	27) 74LS10				
14) 74LS03					
XOR		Encoder	Decoder	Demux	Mux
56) 7486n		1) 74LS147	7) 74154n	24) 74154n	32) 74151n
57) 74LS266 (XNOR)		2) 74LS148	8) 74155n	25) 74155n	33) 74153n
58) 74LS86		3) 74LS348	9) 74156n	26) 74156n	34) 74LS151
59) 74HC266		4) 4532B	10) 74159n	27) 74159n	35) 74LS153
60) 74HC86		5)74HC147	11) 7445n	28) 4556B	36) 74LS251
		6)74HC148	12) 74LS138	29)74HC137	37) 74LS253
9-Bit Parity Generator			13) 74LS42	30)74HC139	38) 74LS399
1) 74LS280D			14) 74LS445	31)74HC154	39) 4019B
2) 74LS280N			15) 4028B		40) 4512B
3) 74F280D			16) 4514B		41)74HC298
4) 74F280N			17) 4515B		
5) 74AS280M			18) 4556B		
6) 74AS280N			19)74HC137		
7) 74180N			20)74HC138		
8) 74S280D			21)74HC139		
9) 74S280N			22)74HC154		
		23)74HC42			

Представление двоичных чисел различными формами кодов

Система счисления - это код, в котором используют специальные символы для обозначения количества каких-либо объектов.

Десятичная система имеет символы 0, 1, 2, 3, ..., 9 всего их 10, поэтому ее иногда называют *системой счисления с основанием 10*.

Двоичная система счисления имеет только 2 символа 0 и 1, поэтому ее называют *системой счисления с основанием 2*. Символы десятичной системы счисления могут быть записаны в двоичной системе следующим образом:

Таблица 6

Десятичный символ	0	1	2	3	4	5	6	7	8	9
Двоичное число	0	1	10	11	100	101	110	111	1000	1001

Заметим, что символы 0 и 1 в обеих системах совпадают. Рассмотрим число 648 в десятичной системе - его иногда записывают так: 648_{10} . В этом числе:

Цифра 6 обозначает 600, так как занимает 3 разряда слева от десятичной точки ($6 \cdot 10^2$),

Цифра 4 обозначает 40, так как занимает второй разряд от десятичной точки ($4 \cdot 10^1$),

Цифра 8 представляет число 8 ($8 \cdot 10^0$), т.к. находится в первом разряде слева от десятичной точки, таким образом, все число есть сумма:

$648 = 600 + 40 + 8 = 6 \cdot 10^2 + 4 \cdot 10^1 + 8 \cdot 10^0$, где цифры 648 образуют код, а 10^{n-1} определяет вес n -го разряда, ($\cdot$) символ операции умножения. Этот пример иллюстрирует понятие «вес разряда». Аналогичное понятие «вес разряда» используется и в двоичной системе.

Например, число $1011_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$

Общее правило:

Вес следующего разряда = весу предыдущего разряда умноженному ($\cdot$) на основание системы счисления

Используя это правило, запишем веса десяти первых разрядов двоичной системы:

Таблица 7

2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
512	256	128	64	32	16	8	4	2	1

Преобразование двоичных чисел в десятичные

Пример 1

Пусть двоичное число 110011 необходимо преобразовать в десятичное.

1. Подписываем под таблицей 8 преобразуемое число:

Таблица 8

512	256	128	64	32	16	8	4	2	1
				1	1	0	0	1	1

2. Искомое десятичное число будет равно сумме произведений соответствующих разрядов двоичного числа и их «весов» из таблицы 2:

$$110011_2 = 1 \cdot 32 + 1 \cdot 16 + 0 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 = 51_{10}.$$

Пример 2

Пусть двоичное число 1111101000₂ необходимо преобразовать в десятичное. Решение представлено ниже:

$$1111101000_2 = 1 \cdot 512 + 1 \cdot 256 + 1 \cdot 128 + 1 \cdot 64 + 1 \cdot 32 + 0 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 0 \cdot 2 + 0 \cdot 1 = 1000_{10}.$$

Преобразование десятичных чисел в двоичные

Пусть 13₁₀ нужно перевести в двоичную систему счисления. Перевод в этом случае осуществляется делением числа 13₁₀ на основание системы счисления в которую мы преобразуем. В целых числах с выписыванием остатков деления, по следующей схеме:

Пример 3

$$13 : 2 = 6 \quad \text{остаток } 1 \quad \text{младший (нулевой) разряд весом } 1$$

$$6 : 2 = 3 \quad \text{остаток } 0 \quad \text{первый разряд весом } 2$$

$$3 : 2 = 1 \quad \text{остаток } 1 \quad \text{второй разряд весом } 4$$

$$1 : 2 = 0 \quad \text{остаток } 1 \quad \text{третий разряд весом } 8$$

Помня о том, что самый младший разряд всегда занимает крайнее правое место в записанном числе в любой системе счисления, выписываем результат:

$$13_{10} = 1101_2.$$

Процесс перевода заканчивается в тот момент, когда очередной результат деления даст ноль (0) целых.

Вывод: остатки, от деления, выписанные в соответствии с весами разрядов, дадут искомое число.

Переведем еще одно число 37_{10} в двоичную систему счисления:

Пример 4

$$37:2 = 18 \text{ остаток } 1$$

$$18:2 = 9 \text{ остаток } 0$$

$$9:2 = 4 \text{ остаток } 1$$

$$4:2 = 2 \text{ остаток } 0$$

$$2:2 = 1 \text{ остаток } 0$$

$$1:2 = 0 \text{ остаток } 1$$

Отсюда $37_{10} = 100101_2$.

Шестнадцатеричные числа

В шестнадцатеричной системе счисления, согласно определению, должно быть 16 различных символов перечислим их 0,1,2,3,4,5,6,7,8,9,A, B, C, D, E, F.

Буква A обозначает число 10

B обозначает число 11

и т.д.

Преимущество шестнадцатеричной системы состоит в том, что она позволяет реализовывать переход от шестнадцатеричной к двоичной системе счисления достаточно просто, используя тетрады (tetra в переводе с греческого означает четыре) двоичных символов.

Например: символ F в шестнадцатеричной системе соответствует четырехразрядному числу 1111_2 . Двухразрядное двоичное число $A6_{16} = 10100110_2$. Для перехода от двоичной к шестнадцатеричной системе счисления используют следующую таблицу соответствия:

Таблица 9

Десятичное число	Двоичное число	Шестнадцатеричное число
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7

8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Переход от шестнадцатеричной системы к двоичной

Пусть мы хотим перейти от шестнадцатеричного числа C_{16} к его двоичному эквиваленту. Для этого необходимо по таблице 3 найти тетраду для цифры C_{16} , затем по той же таблице найти тетраду для цифры 3_{16} и после этого записать найденные тетрады в том порядке в каком стояли цифры C_{16} и 3_{16} в исходном шестнадцатеричном числе.

$$C_{16} - 1100_2$$

$$3_{16} - 0011_2$$

$$C 3_{16} = 11000011_2$$

Замечание: при использовании таблицы 3 для перехода необходимо всегда помнить, что работа идет только с тетрадами, то есть нельзя брать вместо одной шестнадцатеричной цифры двоичный эквивалент меньше четырех двоичных знаков.

Пример 5

$$29_{16} = 00101001_2.$$

Переход от двоичной к шестнадцатеричной системе

Этот переход производится по той же таблице 3, только предварительно необходимо разбить исходное двоичное число на тетрады начиная с крайнего правого разряда. После этого тетрады в исходном числе заменяют соответствующими шестнадцатеричными символами. Например:

Пример 6

$$11101010_2 = X_{16}, \text{ где } X - \text{искемое выражение}$$

$$1110_2 = E_{16}.$$

$$1010_2 = A_{16}. \text{ Поэтому } 11101010_2 = EA_{16}.$$

Переход от шестнадцатеричной системы к десятичной

Переход от шестнадцатеричной системы к двоичной осуществляется по тому же алгоритму, что и переход от двоичной сис-

темы к десятичной. Разница состоит в том, что вес разряда в шестнадцатеричной системе представляет собой натуральную степень числа 16, так как 16 в данном случае является основанием системы счисления. Сетка младших четырех разрядов шестнадцатеричной системы представлена в таблице 10.

Таблица 10

16^3	16^2	16^1	16^0
4096	256	16	1

Пример 7

$2DB_{16} = X_{10}$, где X – искомое выражение

$$2DB_{16} = 2 \cdot 16^2 + D \cdot 16^1 + B \cdot 16^0 = 2 \cdot 256 + 13(D_{16}) \cdot 16 + 11(B_{16}) \cdot 1 = 731_{10}.$$

Пример 8

$$2C6E_{16} = 2 \cdot 16^3 + C \cdot 16^2 + 6 \cdot 16^1 + E \cdot 16^0 = 2 \cdot 4096 + 12 \cdot 256 + 6 \cdot 16 + 14 \cdot 1 = 11374_{10}.$$

Переход от десятичной системы счисления к шестнадцатеричной

Переход в этом случае осуществляется делением исходного числа, на основание системы счисления в которую происходит переход, в целых числах с выписыванием остатков деления, по схеме, которую уже использовали в пункте 2 этой темы.

Пример 9

$$47_{10} = X_{16}.$$

$$47_{10} : 16 = 2 \text{ остаток } 15 - F \text{ разряд весом } 1. (16^0)$$

$$2_{10} : 16 = 0 \text{ остаток } 2 - 2 \text{ разряд весом } 16. (16^1)$$

$$\text{Поэтому } 47_{10} = 2F_{16}.$$

Остатки представленные в шестнадцатеричном виде с учетом веса разряда и дают искомое шестнадцатеричное число.

Основы двоичной арифметики

1. **Сложение:**
- $$\begin{aligned} 0 + 0 &= 0 \\ 0 + 1 &= 1 \\ 1 + 0 &= 1 \\ 1 + 1 &= 0 \rightarrow 1 \text{ (перенос в старший разряд)} \end{aligned}$$

Пример сложения многоразрядных чисел. Сложение чисел в двоичной системе счисления.

Запись:

2	10	0000000000000010
+3	+11	+ 0000000000000011
5	101	0000000000000101

Требуется сложить два числа 18_{10} и 23_{10}

$$\begin{array}{r}
 \overset{1}{\curvearrowright} \overset{1}{\curvearrowright} \overset{1}{\curvearrowright} \text{--перенос 1 в старшие разряды} \\
 18 = 010010 \\
 + 23 = 010111 \\
 \hline
 41 = 101001
 \end{array}$$

При сложении чисел надо обязательно проговаривать алгоритм сложения. Следует учитывать, что сложение, как обычно, нужно начинать с младшего разряда. Если сумма единиц разряда окажется равной или большей основания системы счисления, то возникает перенос единицы в старший разряд. После сложения следует обязательно сделать обратный перевод и убедиться, что число 101 — действительно число 5 в десятичной системе счисления.

2. **Вычитание:**

$$\begin{array}{l}
 0 - 0 = 0 \\
 1 - 1 = 0 \\
 1 - 0 = 1 \\
 0 - 1 = 1 \rightarrow \text{заем из старшего разряда}
 \end{array}$$

Вычитание двоичных чисел. В компьютере вычитание заменяется сложением с отрицательным числом, представленным в дополнительном коде.

Запись:

$$\begin{array}{r}
 14 \quad 1110 \quad 0000000000001110 \\
 -9 \quad +1001 \quad +1111111111110111 \\
 \hline
 5 \quad 101 \quad 1000000000000101
 \end{array}$$

Так как под целое число отводится 16 разрядов, то старшая единица теряется. Ответ получается 101.

3. **Умножение:**

$$\begin{array}{l}
 0 \times 0 = 0 \\
 1 \times 0 = 0 \\
 0 \times 1 = 0 \\
 1 \times 1 = 1
 \end{array}$$

Умножение чисел в двоичной системе счисления.

$$\begin{array}{r}
 110 \\
 \times 101 \\
 \hline
 110 \\
 + 000 \\
 110 \\
 \hline
 11110_2 = 1E_{16}.
 \end{array}$$

При умножении $1 \cdot 1 = 1$ и $1 \cdot 0 = 0$. Анализируя примеры умножения в двоичной системе счисления, необходимо обратить внимание на важную особенность выполнения этой операции в данной системе счисления. Так как очередная цифра множителя может быть только 1 или 0, то промежуточное произведение равно либо множимому, либо 0. Таким образом, операция умножения в двоичной системе фактически не производится: в качестве промежуточного произведения записывается либо множимое, либо 0, а затем промежуточные произведения суммируются. Иначе говоря, операция умножения заменяется последовательным сложением.

Сложение в дополнительном коде

Обратный код получается путем замены всех “0” на “1” и всех “1” на “0” прямого кода (двоичного числа со знаком). Причем, знаковый разряд при этом остается неизменным.

$\boxed{1}0101110$ - прямой код отрицательного числа 46
 $\boxed{1}1010001$ - обратный код числа 46

Замена “0” на единицу (“1”) называется *инвертированием* (также и замена “1” на “0”).

Обратный код, дополненный единицей в младшем разряде, называется *дополнительным кодом*. Последовательность действий при получении дополнительного кода:

$\boxed{1}0101110$ - прямой код
 { инвертирование всех разрядов кроме знакового
 $\boxed{1}1010001$ - обратный код
 + 1 - дополнение 1 к младшему разряду
 $\boxed{1}1010010$ - дополнительный код.

Обратный и дополнительный коды позволяют использовать вместо вычитания операцию сложения, при этом результат автоматически получается нужный.

**Федеральное агентство по образованию
Государственное образовательное учреждение
высшего профессионального образования
Московский Государственный Индустриальный Университет
(ГОУ МГИУ)**

Факультет “Прикладная математика и техническая физика”
Кафедра “Автоматика, информатика и системы управления”

Тема № ____
Название лабораторной работы
по курсу “Вычислительные машины системы и сети”

Группа: ____
Студент: ____
Преподаватель: ____
Вариант: ____
Оценка: ____
Дата ____

Москва 200_

Приложение 4

Кратные и дольные единицы

Приставка	Множитель	Сокращенное русское обозначение	Сокращенное зарубежное обозначение	Примеры
йотта	10^{24}	Й	Y	5 Йбит. 5 йотта-бит. Пять септиллионов бит.
зетта	10^{21}	З	Z	4 Збайт. 4 зеттабайт. Четыре секстиллиона байт.
экса	10^{18}	Э	E	3 Эбайт. 3 экзabayт. Три квинтиллиона байт.
пета	10^{15}	П	P	2 Пбит. 2 петабит. Два квадриллиона бит.
тера	10^{12}	Т	T	1 ТВт. 1 тераватт. Один триллион ватт.
гига	10^9	Г	G	2 ГГц. 2 гигагерца. Два миллиарда герц.
мега	10^6	М	MEG(M)	3 Мом. 3 мегаом. Три миллиона Ом.
кило	10^3	к	K	4 кВ. 4 киловольт. Четыре тысячи вольт.
милли	10^{-3}	м	m	1 мА. 1 миллиампер. Одна тысячная ампера.
микро	10^{-6}	мк	μ	2 мкс. 2 микросекунды. Две миллионных секунды.
нано	10^{-9}	н	n	3 нГн. 3 наногенри.
пико	10^{-12}	п	p	4пФ. 4 пикофарады.
фемто	10^{-15}	ф	f	5 фс. 5 фемтосекунд.
атто	10^{-18}	а	a	9 ас. 9 аттосекунд.

Литература

1. Мелехин В.Ф., Павловский Е.Г. Вычислительные машины, системы и сети: Учебник для студ. вузов. – М.: Академия, 2006. – 506 с.
2. Угрюмов Е.П. Цифровая схемотехника: Учеб. пособие для вузов. – 2-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2004. – 800 с.: ил.
3. Опадчий Ю.Ф., Глудкин О.П., Гуров А.И. Аналоговая и цифровая электроника (полный курс): Учебник для вузов. Под ред. О.П. Глудкина – М.: Горячая линия-Телеком, 2005. – 768 с.: ил.
4. Семененко В.А., Скуратович Э.К. Арифметико-логические основы компьютерной схемотехники: Учебное пособие для высшей школы. – М.: Академический Проект, 2004. – 144 с. – («Gaudeamus»).
5. Браммер Ю.А., Пащук И.Н. Цифровые устройства: Учебное пособие для вузов. – М.: Высш. шк., 2004. – 229 с.: ил.
6. Карлащук В.И. Электронная лаборатория на IBM PC. Программа Electronics Workbench и ее применение. Издание 5-е, дополненное и переработанное. М.: Солон-Р, 2004.
7. Лачин В.И., Савелов Н.С. Электроника: Учебное пособие. – 5-е изд., перераб. и доп. – Ростов н/Д: Феникс, 2005. – 704 с. – (Высшее образование).
8. Интегральные микросхемы и их зарубежные аналоги: Справочник. Том 2./А.В. Нефедов. - М.:ИП РадиоСофт, 1998г. - 640с.: ил.
9. Бирюков С.А. Применение цифровых микросхем серий ТТЛ и КМОП. – 2-е изд., стер. – М.: ДМК, 2000. – 240 с.
10. Отечественные микросхемы и зарубежные аналоги Справочник. Перельман Б.Л. Шевелев В.И. "НТЦ Микротех", 1998г., 376 с.
11. Прянишников В.А. Электроника: Курс лекций. – 2-е изд. исп. и под. – СПб.: КОРОНА принт, 2000. – 416 с.
12. Гук М.Ю. Аппаратные средства IBM PC: Энциклопедия. – 2-е изд. – СПб.: Питер, 2003. – 928 с.
13. Токхайм Р. Микропроцессоры: Курс и упражнения / Пер. с. англ. под .ед. В. Н. Грасевича. – М.: Энергоатомиздат, 1987. – 336 с.

14. Новиков Ю.В. Основы цифровой схемотехники. – М.: Мир, 2001. – 379 с.
15. Хоровиц П., Хилл У. Искусство схемотехники: Пер. с англ. – 6-е изд. – М.: Мир, 2001. – 830 с.